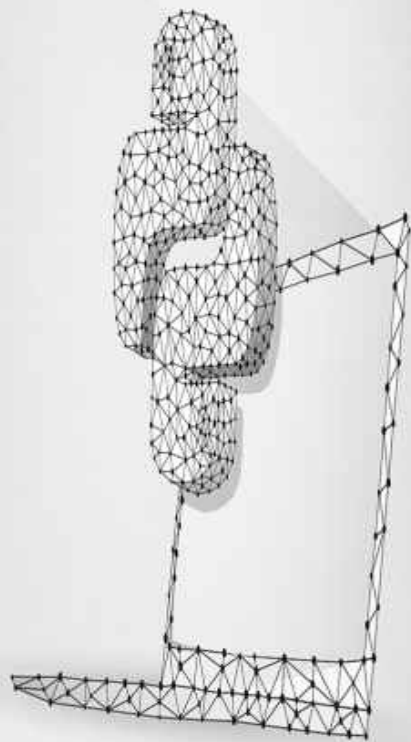


PYTHON

— DATA SCIENCE —

HANDS ON LEARNING FOR BEGINNERS

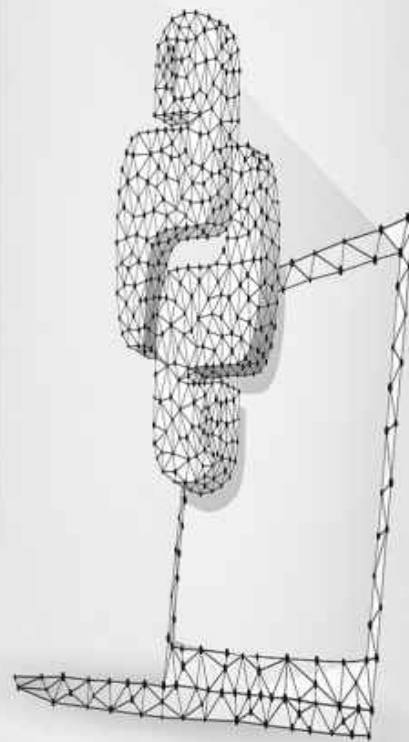


TRAVIS BOOTH

PYTHON

— DATA SCIENCE —

A HANDS ON GUIDE BEYOND THE BASICS

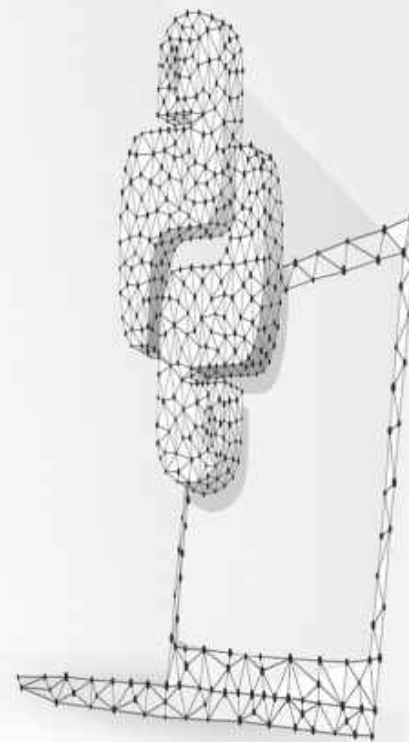


TRAVIS BOOTH

PYTHON

— DATA SCIENCE —

A HANDS ON GUIDE FOR EXPERTS



TRAVIS BOOTH

PYTHON DATA ANALYTICS

Travis Booth

© Copyright 2019 - All rights reserved.

The content contained within this book may not be reproduced, duplicated or transmitted without direct written permission from the author or the publisher.

Under no circumstances will any blame or legal responsibility be held against the publisher, or author, for any damages, reparation, or monetary loss due to the information contained within this book. Either directly or indirectly.

Legal Notice:

This book is copyright protected. This book is only for personal use. You cannot amend, distribute, sell, use, quote or paraphrase any part, or the content within this book, without the consent of the author or publisher.

Disclaimer Notice:

Please note the information contained within this document is for educational and entertainment purposes only. All effort has been executed to present accurate, up to date, and reliable, complete information. No warranties of any kind are declared or implied. Readers acknowledge that the author is not engaging in the rendering of legal, financial, medical or professional advice. The content within this book has been derived from various sources. Please consult a licensed professional before attempting any techniques outlined in this book.

By reading this document, the reader agrees that under no circumstances is the author responsible for any losses, direct or indirect, which are incurred as a result of the use of information contained within this document, including, but not limited to, — errors, omissions, or inaccuracies.

TABLE OF CONTENTS

PYTHON DATA ANALYTICS

The Beginner's Real-World Crash Course

Introduction

Chapter 1: Introduction to Data Analytics

Difference Between Data Analytics and Data Analysis

Necessary Skills for Becoming a Data Scientist

Python Libraries for Data Analysis

Chapter 2: About NumPy Arrays and Vectorized Computation

Creating ndarrays

Table for Array Creation Functions

Chapter 3: Improve the System

Quality in Software Development

Getting It Right the First Time

Cycles in Learning

5 Tips You Can Apply to Amplify Learning

Continuous Improvement

Chapter 4: Create Quality

Pair Programming

Test Driven Development

Regular Feedback to Inspect and Adapt

Reduce Time Between Stages

Automation

Constant Integration

Controlling Trade-Offs

Build Quality in the Software Delivery

Chapter 5: Decide as Late as Possible

[Concurrent Software Development](#)

[Rise in Cost](#)

[Problem Solving Strategies](#)

[Simple Rules in Software Development](#)

Chapter 6: Fast Delivery

[Why Should You Deliver Fast?](#)

[Software Development Schedules](#)

[Information Producers](#)

[Cycle Time](#)

[Slack](#)

[To Deliver Fast, You Must Think Small](#)

Chapter 7: Trust and Team Empowerment

[Team Empowerment](#)

[Motivation and Purpose](#)

[The Foundations of Motivation](#)

Chapter 8: Integrity

[The Goal to the Integrity](#)

[Perceived Integrity](#)

[Create a Model Approach Kind of Design](#)

[How to Maintain the Perceived Integrity?](#)

[Conceptual Integrity](#)

[How to Maintain Conceptual Integrity](#)

Chapter 9: Optimize the Whole

[System Thinking](#)

[System Measurements](#)

Chapter 10: Go Lean in Your Organization

[Learn JIT Coaching](#)

Chapter 11: The Relationship Between Lean and Agile Development

[The Connection Between Lean and Agile Principles](#)

[Iterative Approach](#)

[Connection with Lean: Deliver Fast and Delay Commitment](#)

[Disciplined Project Management Process](#)

[Connection with Lean: Develop Quality](#)

[Short Feedback Loops](#)

[Connection with Lean: Remove waste](#)

[Lean and Agile Development](#)

Chapter 12: Pros and Cons of Lean Software Development

Conclusion

PYTHON DATA ANALYTICS

A Hands-on Guide Beyond The Basics

Introduction

Chapter One: An Introduction to Data Science And Data Analytics

[What Exactly is Data Science?](#)

[What Information is Necessary for a Data Scientist to Know?](#)

[Who is a Data Analyst?](#)

[Do Data Analytics and Data Science Intersect?](#)

[Understanding Machine Learning](#)

[Do Machine Learning and Data Science Intersect?](#)

[Evolution of Data Science](#)

[Data Science: Art and Science](#)

[Five Important Considerations in Data Science](#)

[Ten Platforms to be Used in the Field of Data Science](#)

Chapter Two: Types of Data Analytics

[Descriptive Analytics](#)

[Prescriptive Analytics](#)

[Predictive Analytics](#)

[Data Science Techniques that an Aspiring Data Scientist Should Know](#)

[Classification Analysis](#)

Chapter Three: Data Types and Variables

[Choosing the Right Identifier](#)

[Python Keywords](#)

[Understanding the Naming Convention](#)

[Creating and Assigning Values to Variables](#)

[Recognizing Different Types of Variables](#)

[Working with Dynamic Typing](#)

[The None Variable](#)

[Using Quotes](#)

[How to use Whitespace Characters](#)

[How to Create a Text Application](#)

[Working with Numbers](#)

[Converting Data Types](#)

Chapter Four: Conditional Statements

[How to Compare Variables](#)

[Manipulating Boolean Variables](#)

[Combine Conditional Expressions](#)

[How to Control the Process](#)

[Nesting Loops](#)

[For](#)

Chapter Five: Data Structures

[Items in Sequences](#)

[Tuples](#)

[List](#)

[Stacks and Queues](#)

[Dictionaries](#)

Chapter Six: Working with Strings

[Splitting Strings](#)

[Concatenation and Joining Strings](#)

[Editing Strings](#)

[Creating a Regular Expression Object](#)

Chapter Seven: How to Use Files

[How to Open Files](#)

[Modes and Buffers](#)

[Reading and Writing](#)

[Closing Files](#)

Chapter Eight: Working with Functions

[Defining a Function](#)

[Defining Parameters](#)

[Documenting your Function](#)

[Working with Scope](#)

[Understanding Scope](#)

[Manipulating Dictionaries and Lists](#)

[Abstraction](#)

Chapter Nine: Data Visualization

[Know your Audience](#)

[Set your Goals](#)

[Choose the Right Type of Charts](#)

[Number Charts](#)

[Maps](#)

[Pie Charts](#)

[Gauge Charts](#)

[The Color Theory Advantage](#)

[Handling Big Data](#)

[Prioritize using Ordering, Layout and Hierarchy](#)

[Utilization of Network Diagrams and Word Clouds](#)

[Comparisons](#)

[Telling a Story](#)

Chapter Ten: Visualization Tools for the Digital Age

[7 Best Data Visualization Tools](#)

[10 Useful Python Data Visualization Libraries](#)

Chapter Eleven: An Introduction To Outlier

Detection In Python

[What is an Outlier?](#)

[Why Do We Need To Detect Outliers?](#)

[Why Should We Use PyOD For Outlier Detection?](#)

[Outlier Detection Algorithms Used In PyOD](#)

[Implementation of PyOD](#)

Chapter Twelve: An Introduction To Regression Analysis

[Linear Regression Analysis](#)

[Multiple Regression Analysis](#)

Chapter Thirteen: Classification Algorithm

[Advantages Of Decision Trees](#)

[Disadvantages Of Decision Trees](#)

Chapter Fourteen: Clustering Algorithms

[K-Means Clustering Algorithm](#)

[Code for Hierarchical Clustering Algorithm](#)

Conclusion

References

PYTHON DATA ANALYTICS

The Expert's Guide to Real-World Solutions

Introduction

Chapter 1: Conceptual Approach to Data Analysis

[Techniques Used in Data Analysis](#)

[Data Analysis Procedure](#)

[Methods Used in Data Analysis](#)

[Types of Data Analysis](#)

[Tools Used in Data Analysis](#)

[Benefits of Data Analysis in Python over Excel](#)

Possible Shortcomings of Analyzing Data in Python

Chapter 2: Data Analysis in Python

Python Libraries for Data Analysis

Installation Guide for Windows

Installation Guide for Linux

Installing IPython

Building Python Libraries from Source

Chapter 3: Statistics in Python - NumPy

Generating Arrays

Slicing and Indexing

Importance of NumPy Mastery

Chapter 4: Data Manipulation in Pandas

Installing Pandas

Fundamentals of Pandas

Building DataFrames

Loading Data into DataFrames

Obtaining Data from SQL Databases

Extracting Information from Data

Dealing with Duplicates

Cleaning Data in a Column

Computation with Missing Values

Data Imputation

Describing Variables

Data Manipulation

Chapter 5: Data Cleaning

Possible Causes of Unclean Data

How to Identify Inaccurate Data

How to Clean Data

How to Avoid Data Contamination

Chapter 6: Data Visualization with Matplotlib in Python

[Fundamentals of Matplotlib](#)
[Basic Matplotlib Functions](#)
[Plotting Function Inputs](#)
[Basic Matplotlib Plots](#)
[Logarithmic Plots \(Log Plots\)](#)
[Scatter Plots](#)
[Display Tools in Matplotlib](#)
[How to Create a Chart](#)
[Using Multiple Axes and Figures](#)
[Introducing New Elements to Your Plot](#)

Chapter 7: Testing Hypotheses with SciPy

[Fundamentals of Hypothesis Testing](#)
[Hypothesis Testing Procedure](#)
[One-Sample T-Test](#)
[Two-Sample T-Test](#)
[Paired T-Test](#)
[Using SciPy](#)
[Installing SciPy](#)
[SciPy Modules](#)
[Integration](#)

Chapter 8: Data Mining in Python

[Methods of Data Mining](#)
[Building a Regression Model](#)
[Building Clustering Models](#)

Conclusion

Other Books by Travis Booth

Machine Learning Series

[Machine Learning With Python:
Hands-On Learning for Beginners](#)

[Machine Learning With Python:
An In Depth Guide Beyond the Basics](#)

Deep Learning Series

[Deep Learning With Python:
A Hands-On Guide for Beginners](#)

[Deep Learning With Python:
A Comprehensive Guide Beyond the Basics](#)

Python Data Science Series

[Python Data Science:
Hands-On Learning for Beginners](#)

[Python Data Science:
A Hands-On Guide Beyond the Basics](#)

Bonus Offer:

***Get the Ebook absolutely free when you purchase the paperback
via Kindle Matchbook!***

PYTHON DATA ANALYTICS

The Beginner's Real-World Crash Course

Travis Booth

Introduction

Although Python is more known as a programming language, it has become a consistently popular tool for data analytics. In the recent years, several libraries have reached maturity thereby permitting Stata and R users to take advantage of the flexibility, performance, and beauty of Python without having to sacrifice the functionalities gathered by the older programs over the years.

In this book, we will take a look at introducing the social science and data analysis applications of Python. This book is particularly tailored for those users that have little or no programming experience of note. It will be especially useful for these programmers who wish to get things done and have a lot of experience in programs such as Stata and R.

The greatest reason for learning Python also happens to be the hardest to explain to someone who is just beginning his work in Python. Python is superbly designed in terms of structure and syntax; it is intuitive; however, very powerful general-purpose programming language.

Python's main advantage lies in how easy it is. For these kinds of things, you need an easy language. A harder one will generally take quite a large toll on your ability to program and analyze data.

Because of this, taking a difficult language will show off your programming skills, but not your data analytics skills. In the end, there's a reason many of the top companies in the world use python as their main programming

language. It lets the coders focus on what's really important instead of looking up syntax every 20 minutes.

The programming language was explicitly designed; therefore, the code written in the language is simple for humans to read and reduces the amount of time needed for writing the code. Actually, its ease of use is the main reason why most of the top CS programs in the US use Python to introduce computer science in their classes according to a recent study.

Having said all that, Python is very real and is a general-purpose programming language. You will find major companies such as Dropbox and Google using Python for their core applications. This sets the programming language apart from other domain-specific languages such as R which are highly tuned to cater to a specific purpose such as statistics and they work for specific audiences. R was created by John Chambers with the target of making a language that even the non-programmers can learn to use quickly, and it could also be utilized by the power users. He succeeded in his endeavor to a large degree as can be seen from the uptake of R. However, in the process of making the programming language more accessible to the non-programmers, some compromises had to be made in the language. R just serves one purpose, and that is statistical analysis, and its syntax contains all kinds of peculiarities and moles that come with the original bargain.

Python, on the other hand, needs some training to get started, although not a great deal more. However, there are no limits to what you can do by using Python; when you are learning Python, you are learning a full programming language. It means if you have to work in another programming language such as C or Java for some reason or have to understand pieces of code written by somebody else or in some cases have to deal with programming problems, this learning background in programming will provide a solid

conceptual foundation for anything you will come across. This is the reason why all the top CS programs teach Python.

There are many reasons for choosing Python as your tool, but we have not touched on the most compelling reason for them all. Python will set you up for understanding and operating in the broad programming world. In case you are interested in performing computational social science and building general programming skills, Python gives you more flexibility. If you are looking to run just the regressions R is great or if you are doing things which fit the mold perfectly because someone has created the molds by using R functions. However, social scientists will find new data sources such as text and find newer ways of analyzing it. So the better you are at a general programming language, the more prepared you are for stealing the tools from other disciplines and write newer tools by yourself.

Most experienced programmers will find the idea of using a single programming language extremely appealing; it allows you to unify your workflow. For everyone, one of the best things about Python is that you can pretty much do anything you wish by using this programming language. However, everyone doesn't feel that way. There are a lot of people who use Python with other tools such as R and move back and forth depending on the application at hand. However, even in case you are planning to do this mix and match, the great thing about Python is that due to its generality several people have suggested that becoming better at Python has turned them into better programmers. Not only in Python but also on Stata and R.

Performance is not a criterion that comes into play for the majority of social science applications. Therefore, it is not the top reason for selecting Python. But in case you find yourself in a situation where performance does matter, Python has some significant advantages over all the other high-level

languages including R and Matlab, both in terms of memory use and computation speeds. R is notorious for being a memory hog. More significantly, there are new tools available in Python which make it possible for writing Python code which runs at the same speed as that of FORTRAN or C. Sometimes even faster than native Python or R. Although this is a lower-level consideration in most cases, it is an example of the advantages of using Python giving you options which will hold no matter what the future will bring.

Chapter 1

Introduction to Data Analytics

Difference Between Data Analytics and Data Analysis

Data analytics and data analysis are many times treated as interchangeable terms. However, they are slightly different in their meanings. Data analysis is one of the practices employed by Data analytics, and it includes the use of data analytic tools and techniques for achieving business objectives. Data analytics is a broad term, and it refers to the process of compilation and analysis of data in order to present their findings to the management and helping them make decisions. Data analysis is a subcomponent of the bigger picture called data analytics, which uses the technical tools along with data analysis techniques.

The data analytics tools that are used by the data analysts are Tableau Public, KNIME, OpenRefine, RapidMiner, NodeXL, Google Fusion Tables, Wolfram Alpha, and Google Search Operators. Data analysis is a process in which we examine, transform, and arrange raw data in some specific ways for generating useful information out of it. Basically, data analysis permits evaluation of data via logical and analytical reasoning, and it leads to some outcome or conclusion in a context. It's a multi-faceted procedure and contains many steps, approaches, and a range of techniques. The approach you will take to data analysis will mainly depend on the data available to you for analysis and the reason for performing the analysis.

Data analytics is the conventional way of doing analytics and is used in

several industries such as health, business, insurance, and telecom for making decisions from data and performing necessary actions on the data. Data analysis is used in organizations and various domains for analyzing data and get useful insight out of the data.

In data analytics, you collect the data and inspect it in general. It has more than a single use while data analysis means the definition of data, investigation, cleaning, and removing the Na values or other outliers in the data and thereby transforming it for producing a meaningful outcome. For performing the data analytics, you need to learn several tools to be able to take necessary actions on the data. You need to be aware of Python, R, SAS, Apache Spark, Tableau Public, Excel, etc.. You need hands-on tools for data analysis such as KNIME, Open Refine mentioned above.

The life cycle of data analytics is like this,

- Business Case Evaluation.
- Data Extraction.
- Data Acquisition and Filtering.
- Data Identification.
- Data Validation and Cleansing.
- Data Analysis.
- Data Aggregation and Representation.
- Data Visualization.
- Utilization of Analysis Results.

We are aware that data analysis is the sub-component of the broader term data analytics. Therefore, data analysis life cycle also comes under data

analytics. This involves,

- Data gathering.
- Data scrubbing.
- Analysis of data.
- Data Interpretation.

The last step enables you with what the data wishes you to say.

When you are trying to find out what will happen next, especially in terms of marketing, we go to data analytics as data analytics helps in predicting future figures. On the other hand, in the case of data analysis, the analysis is performed on the last set of data to understand what took place prior. Both data analytics and data analysis are necessary to understand the data and are helpful in understanding the future demands. They are helpful in performing the analysis of the data and taking a look at the past.

Interpolation and extrapolation are two especially important principles in analyzing data. Interpolation means to take a look at the data, and based on current data, determine what the data was in the past.

This can be especially useful when performing market analysis and the like. Extrapolation is the opposite; you take the current data and make an educated guess as to what it's going to be in the future. Generally, it is the much more used of the two, as companies are largely more concerned with the future than they are with the past.

These two techniques will be scattered around your career, so make sure you know what both of them are.

The data usage has increased rapidly in the recent past, and a large amount of

data gets collected in the companies. The data can be related to business, customers, users of applications, stakeholders, or visitors. The data is divided and processed to understand, find, and analyze patterns. Data analytics refers to different skills and tools that involve quantitative and qualitative methods for collecting the data to have an outcome that can be utilized for improving efficiency, reducing risks, increasing productivity, and raise business gains. These techniques change from company to company as per their demands.

Data analysis is a specialized decision taking a tool that uses various technologies such as tableau public, KNIME, Open Refine, Rapid Miner, etc. These tools are useful for performing exploratory analysis, and they give some insight from the available data by using, cleaning, modeling, transforming, and visualizing the data and provide the outcome.

Necessary Skills for Becoming a Data Scientist

Here are some skills required for becoming a data scientist.

1. Education: Almost all data scientists are well educated. 88% have a Master's degree at least, and 46% have PhDs although there are some notable exceptions to this rule a strong academic background is required for developing the in-depth knowledge needed for becoming a data scientist. In order to become a data scientist, you may opt for a Bachelor's degree in social sciences, computer sciences, statistics, and physical sciences. Some of the common fields of study are Math and Statistics (32%), computer science (19%), and engineering (16%). A degree in these courses can get you the necessary skills for processing and analyzing a large amount of data.

But, after completing the degree course, do not consider that you are done. Remember, most data scientists have a master's degree or Ph.D. They also undergo online training for learning specific skills such as how to use Big

Data querying or Hadoop. You may enroll for the master's degree course in the data science field or in the field of Astrophysics, Mathematics, or other related areas. The various skills you have learned while doing the degree program enable you to transit toward data science. Apart from the information you have learned in your classroom, you can practice by building an app or start a blog or explore data analysis, which will help you to learn more.

Keep in mind that this is a developing field; it's important to keep on top of new developments. A single research paper can easily flip the whole world of data science upside down. Because of this, make sure to keep in touch with the data analytics world consistently. This will let you perform way better because you'll know about the principles and techniques your colleagues won't.

2. R Programming: An in-depth knowledge of at least one of the analytical tools is required. R is used a lot for data science as it is specifically designed for the data science requirements. R can be used for solving any problems you will face in data science. A lot of data scientists use R for solving the statistical problems. But R has a steep learning curve, and it is difficult to learn, especially if you have mastered some programming language but, there are resources available on the internet to get you started with R.

The thing with R is that it's generally less useful than other languages in general-purpose programming situations. Sure, it's great for data analytics, but it's not good for anything else.

This makes it not ideal for a first language, as you'll find that even if you're able to analyze the data, making programs that are able to actually use it will be nigh-impossible. In the end, you want to focus on making programs that are able to use your data, as well as thinking of the implications of your data.

3. Python Coding: Python is the most commonly used programming language used in various requirements of data science. It is a great choice for the data scientists, along with Perl, Java, C, and C++. A survey conducted by O'Reilly indicates that 40% of the users prefer Python as their primary programming language option. Due to the versatility of Python, you can use it for all the steps related to data science processes. The languages can make use of different data formats, and it is easily possible to import the SQL tables inside the code. It permits you to create data sets. You can literally find all types of data sets needed on Google.

4. Using Hadoop Platform: Hadoop platform is not a necessity, but it is preferred in most cases. It is also a strong selling point to have experience in Pig or Hive. You are also better off being familiar with various cloud tools such as Amazon S3. CrowdFlower conducted a study on 3490 jobs related to data science, and it ranked Apache Hadoop as the 2nd most significant skill for data scientists with a 49% rating. While working as a data scientist, you will encounter situations where the data volume will exceed the memory of your system, or you will be required to send data to various servers, and this is where Hadoop comes into the picture. Hadoop can be used to quickly convey data to different points of a system, and that is not all. You may use Hadoop for data filtration, data exploration, data sampling, and summary.

5. SQL DB/Coding: Although Hadoop and NoSQL have become significant components of data science, you may still expect that the candidates are able to write and execute difficult queries in SQL. The Structured Query Language known popularly as SQL is a programming language which is useful for carrying out various operations such as addition, deletion, and extraction of data from the DB. It may also aid you in carrying out analytical functions and change the DB structures.

You need to be good in SQL for being a data scientist. It is because SQL is particularly developed to aid you to communicate, access, and work on the data. It will provide insight if you utilize it for querying a DB. It comes with concise commands which will help you save time and reduce the amount of programming needed for performing complex queries. Learning SQL aids you in understanding the relational database better and will boost your profile for being a data scientist.

6. Apache Spark: It is one of the big data technologies available and is one of the most popular ones. Apache Spark is a big data computation framework similar to Hadoop. The advantage of using Apache Spark is that it is faster than Hadoop. This is because Hadoop will read and write to a disk and that makes it slower while Apache Spark caches the computations in the memory. Apache Spark is particularly designed for data science applications and helps in running complicated algorithms faster. It also helps in the dismantling of data processing when you are dealing with a sea of data and saving time in the process. This also aids the data scientists to manage unstructured and complex data sets. You may use it on a single machine or a cluster of machines.

The Apache Spark enables the data scientists to prevent loss of data. The advantage of using Apache Spark is in its speed and platform as it makes things easy for carrying out data science-related projects. By using Apache Spark, you can carry out data analytics from the intake of data to distributing the computing.

7. AI and Machine Learning: There are many data scientists out there who are not proficient in the area of machine learning and related techniques. These include reinforcement learning, neural networks, and adversarial learning, etc. If you are looking to stand out from the other data scientists,

you will have to understand the machine learning techniques like supervised machine learning, logistic regression, and decision trees, etc. These skills can aid you in solving various data science related problems which are based on predictions made by major organizational outcomes.

The data science requires skill application in several areas of machine learning. In one of their surveys, Kaggle revealed that just a small percentage of data professionals would be competent in the advanced ML techniques like supervised and unsupervised machine learning, natural language processing, time series, computer vision, outlier detection, survival analysis, recommended engines, adversarial learning, and reinforcement learning. In data science, you will be working with a large number of data sets. So it will be a good idea to be familiar with machine learning.

8. Data Visualization: There is a large amount of data generated in the world of business and frequently. All this data must be translated into some format which will be legible. Normally, people understand pictures in the form of graphs and charts far more than raw data. There is an old idiom which says, "A picture is worth a thousand words." As a data scientist, you need to be capable of visualizing the data by using data visualization tools like ggplot, d3.js, Tableau, and Matplotlib. All these tools will aid you in converting the complex results of the projects into a format which will be simple to understand. The problem is that many people do not understand p values or serial correlation. It is required to be shown visually what the terms represent in the results. Data visualization provides an opportunity for organizations to work directly with data. They are able to grasp insights quickly, which will help them act on newer business opportunities and stay ahead of the competition.

9. Unstructured Data: It is important that a data scientist is able to work by

using unstructured data. The unstructured data is basically all the undefined content which doesn't fit into DB tables. Examples of it include blog posts, social media posts, audio, video feeds, customer reviews, and videos. These are heavy chunks of text grouped together. Sorting out this kind of data is tough as it is not streamlined. Many people refer to the unstructured data as dark analytics due to the complexity.

If you are working with unstructured data, it will aid you in unraveling insights, which can be useful in decision making. As data scientists, you need to have the capability to understand and manage unstructured data from a range of platforms.

10. Intellectually Curious Nature: Albert Einstein used to say, "I have no special talent. I am only passionately curious."^[1] You must have seen the phrase everywhere lately, especially because it is related to data science. Curiosity can be defined as a desire to acquire more and more knowledge. As data scientists, you must be able to ask questions related to data, as the scientists spend much of their time finding and preparing data. It is because the data science field is evolving very quickly, and you need to learn a lot for keeping up with the pace.

You are regularly required to update your knowledge by reading content on the internet and also by reading relevant books on the trends involved in data science. Do not get overwhelmed by the sheer quantity of data which is floating around the internet. You need to know how you will make sense out of it all.

One of the main skills required for this is a curiosity for succeeding as a data scientist. For instance, you might not see anything relevant in the data you have collected initially. Curiosity enables you to work through the data and find answers, along with insights.

11. Business Understanding: For becoming a data scientist, you are required to have a great understanding of the industry you are working in and be aware of the business problems your organization is facing. In relation to data science being able to discern what problems are significant and need to be resolved immediately to reduce the impact on the business. It is why you must know how businesses work so that you may direct the efforts in the right direction.

12. Communication Skills: Organizations searching for good data scientists are looking for people that can fluently and clearly translate the technical findings to non-technical teams such as sales or marketing departments. The data scientist has to empower the business to make decisions by providing them with quantifiable insights. This is in addition to understanding the requirements of the non-technical colleagues for wrangling out the data properly. There are surveys available online that provide information about the communications skills required by quantitative professionals.

Apart from speaking the same language, the organization understands you will communicate by using the storytelling technique employed on data. As a data scientist, you must create a storyline surrounding the data for making it easy for anybody to understand. For example, presenting the data in a tabular form is not as effective as sharing insight from the data in a storytelling format. Using the storytelling format will aid you in properly communicating the findings to the employees.

You need to pay attention to the results while you are communicating and values which are embedded inside the data you have analyzed. Most business owners do not wish to know what has been analyzed; they are only interested in how it will affect their business positively. So you must learn to focus on delivering value and building long-lasting relationships via communication.

13. Teamwork: The data scientist cannot work single-handedly. There is a need to work with other company executives for developing strategies. You will work with product designers and managers for creating better products. You can work with the company marketers for launching campaigns that will convert better. You may work with the client and server software developers for creating data pipelines and improving workflow. You will need to literally work with everyone in the company, including the clients.

Typically, you will collaborate with the team members for developing use cases to understand the business goals better and develop the data necessary for solving the problems. You must be aware of the correct approach for addressing use cases. Data will be needed for resolving the issues and translating and presenting the results into what may easily be understood by the people involved.

Python Libraries for Data Analysis

The Python programming language is assisting the developers for creating standalone PC games, mobiles, and other similar enterprise applications. Python has in excess of 1, 37,000 libraries which help in many ways. In this data-centric world, most consumers demand relevant information during their buying process. The companies also need data scientists for achieving deep insights by processing the big data.

This info will guide the data scientists while making critical decisions regarding streamlining business operations and several other related tasks that need valuable information for accomplishment efficiently. Therefore, with the rise in demand for data scientists, beginners and pros are looking to reach resources for learning this art of analysis and representation of data. There are some certifications programs available online which can be helpful for training. You can find blogs, videos, and other resources online as well.

Once you have understood dealing with unstructured info, they are good for several good opportunities available out there. Let's have a look at some of the Python libraries that are helpful for all these data science-related operations.

1. NumPy: It is among the first choice for the data scientists and developers who know their technologies dealing in data related things. This is a Python package and is available for performing scientific computations. The package is registered under the BSD license. By using NumPy, you may leverage the n-dimensional array objects, C, C++, FORTRAN programs based on integration tools, functions for difficult mathematical operations such as Fourier transformations, linear algebra, and random numbers. The NumPy might also be utilized as a multi-dimensional container for treating generic data. Therefore you may effectively integrate the DB by selecting a variety of operations for performing.

NumPy gets installed under TensorFlow and other such machine learning platforms, thereby internally providing strength to their operations. As this is an array interface, it will allow multiple options for reshaping large data sets. NumPy may be used for treating images, sound wave representations, and other binary operations. In case you have just arrived in the field of data science and machine learning, you must acquire a good understanding of NumPy for processing the real-world data sets.

2. Theano: Another useful Python library is Theano, which assists the data scientists to create big multi-dimensional arrays which are related to computing operations. This is similar to TensorFlow; however, the only difference being it is not very efficient. It involves getting used to parallel and distributed computing-related tasks. By using this, you may optimize, evaluate, or express the data enabled mathematical operations. The library is

tightly joined with NumPy and is powered by the implemented numpy.nd array functions.

Due to its GPU based infrastructure, the library has the capability of processing the operations in quicker ways than compared to the CPU. The library stands fit for stability and speed optimization and delivering you the expected outcome. For quicker evaluation, the C code generator used is dynamic and is extremely popular among data scientists. They can do unit testing here for identifying the flaws in the model.

3. Keras: One of the most powerful Python libraries is Keras that permits higher-level neural network APIs for integration. The APIs will execute over the top of TensorFlow, CNTK, and Theano. Keras was developed for decreasing the challenges faced in difficult researches permitting them to compute quicker. For someone using the deep learning libraries for their work, Keras will be their best option. Keras permits quicker prototyping and supports recurrent and convoluted networks independently. It also allows various blends and execution over CPU and GPU.

Keras give you a user-friendly environment, thereby decreasing the efforts required for cognitive loads by using simple APIs and so providing necessary results. Because of the modular nature of Keras, you may use a range of modules from optimizers, neural layers, and activation functions, etc. for preparing newer models. Keras is an open source library and is written in Python. It is a particularly good option for the data scientists who are having trouble in adding newer models as they may easily add newer modules as functions and classes.

4. PyTorch: It is one of the largest machine learning libraries available for data scientists and researchers. The library aids them with dynamic computational graph designs; quick tensor computation accelerated via GPU

and other complicated tasks. In the case of neural network algorithms, the PyTorch APIs will play an effective role.

This crossbreed front-end platform is simple to use and allows transitioning into a graphical mode for optimization. In order to get precise results in the asynchronous collective operations and for the establishment of peer to peer communication, the library gives native support to its users. By using ONNX (Open Neural Network Exchange), you may export models for leveraging the visualizers, run times, platforms, and many other resources. The greatest part of PyTorch is that it enables a cloud-based environment for simple scaling of resources utilized for deployment testing.

PyTorch is developed on a similar concept to another machine learning library called Torch. During the last few years, Python has gradually become more popular with the data scientists because of the trending data-centric demands.

5. SciPy: This is a Python library used by the researchers, data scientists, and developers alike. However, do not confuse SciPy stack with the library. SciPy gives you optimizations, integration, statistics, and linear algebra packages for the computations. The SciPy is based on NumPy concept for dealing with difficult mathematical problems. SciPy gives numerical routines that can be used for integration and optimization. SciPy will inherit a range of sub-modules to select from. In the event that you have recently started your career in data science, SciPy will be quite helpful for guiding you through the whole numerical computation.

We have seen thus far how the Python programming can assist the data scientists in analyzing and crunching big and unstructured data sets. There are other libraries such as Scikit-Learn, TensorFlow, and Eli5 available for assistance through this journey.

6. Pandas: The Python Data Analysis Library is called PANDAS. It is an open source library in Python for availing the analysis tools and high-performance data structures. PANDAS is developed on the NumPy package, and its main data structure is DataFrame. By using DataFrame, you can manage and store data from the tables by doing manipulation of rows and columns.

Methods such as square bracket notation decrease the personal effort involved in data analysis tasks such as square bracket notation. In this case, you will have the tools for accessing the data in the memory data structures and perform read and write tasks even though they are in multiple formats like SQL, CSV, Excel, or HDFS, etc.

7. PyBrain: This is a powerful modular machine learning library which is available in Python. The long form of PyBrain goes like Python Based Reinforcement Learning Artificial Intelligence and Neural Network Library. For the entry-level data scientists, this offers flexible algorithms and modules for advanced research. It has a range of algorithms available for evolution, supervised and unsupervised learning, and neural network. For the real-life tasks PyBrain has emerged as a great tool, and it is developed across a neural network in the kernel.

8. SciKit-Learn: This is a simple tool used for data analysis and data mining related tasks. It is licensed under BSD and is an open source tool. It can be reused or accessed by anyone in different contexts. The SciKit is developed over NumPy, Matplotlib, and SciPy. The tool is utilized for regression, classification, and clustering or managing spam, image recognition, stock pricing, drug response, and customer segmentation, etc. SciKit-Learn allows for dimensionality reduction, pre-processing, and model selection.

9. Matplotlib: This library of Python is used for 2D plotting and is quite

popular among data scientists for designing different figures in multiple formats across the respective platforms. It can be easily used in the Python code, Jupyter notebook, or IPython shells application servers. By using the Matplotlib, you will be able to make histograms, bar charts, plots, and scatter plots, etc.

10. TensorFlow: TensorFlow is an open source library designed by Google for computing the data flow graphs by using empowered ML algorithms. The library was designed for fulfilling the high demands for training for neural network work. TensorFlow is not only limited to scientific computations conducted by a Google raster. It is used extensively for the popular real-world applications. Because of the flexible and high-performance architecture, you can easily deploy it for all GPUs, CPUs, or TPUs and you can perform the PC server clustering for the edge devices.

11. Seaborn: It was designed for visualizing complex statistical models. Seaborn comes with the potential of delivering accurate graphs like heat maps. Seaborn is developed on the Matplotlib concept, and as a result, it is highly dependent on it. Even the minor data distributions can be seen by using this library, which is the reason why the library has become popular with the developers and data scientists.

12. Bokeh: It is one of the more visualization purpose libraries used for the design of interactive plots. Similar to the library described above, this one is also developed on Matplotlib. Because of the support of used data-driven components such as D3.js this library can present interactive designs in your web browser.

13. Plotly: Now, let's see the description of Plotly, which happens to be one of the most popular web-based frameworks used by the data scientists. The toolbox offers the design of visualization models by using a range of API

varieties supported by the multiple programming languages which include Python. Interactive graphics can be easily used along with numerous robust accessories via the main site plot.ly. For utilizing Plotly in the working model, you will have to set up the available API keys correctly. The graphics are processed on the server side, and once they are successfully executed, they will start appearing on the browser screen.

14. NLTK: The long form of NLTK is Natural Language ToolKit. As indicated by its name, the library is useful in accomplishing natural language processing tasks. In the beginning, it was created for promoting teaching models along with other NLP enabled research like the cognitive theory used in AI and linguistic models. It has been a successful resource in its area and drives real-world innovations of artificial intelligence.

By using NLTK you can perform operations such as stemming, text tagging, regression, corpus tree creation, semantic reasoning, named entities recognition, tokenization, classifications and a range of other difficult AI related tasks. Now challenging work will need large building blocks such as semantic analysis, summarization, and automation. But this work has become easier and can be easily accomplished by using NLTK.

15. Gensim: It is a Python-based open source library that permits topic modeling and space vector computation by using an implemented range of tools. It is compatible with the big test and makes for efficient operation and in-memory processing. It utilizes SciPy and NumPy modules to provide easy and efficient handling of the environment. Gensim utilizes unstructured digital text and processes it by using in-built algorithms such as word2vec, Latent Dirichlet Allocation (LDA), Hierarchical Dirichlet Processes (HDP), and Latent Semantic Analysis (LSA).

16. Scrapy: It is also known as spider bots. Scrapy is a library responsible for

crawling the programs and retrieving structured data out of web applications. Scrapy is a Python written open source library. As the name suggests, Scrapy was designed for scraping. This happens to be a complete framework having the potential to collect data via APIs and acts as a crawler. You can write codes by using Scrapy, re-utilize universal programs, and develop scalable crawlers for the applications. It is created across a spider class that contains instructions for the crawler.

17. Statsmodels: Statsmodels is another Python library, and it is responsible for giving exploration modules by using multiple methods for performing assertions and statistical analysis. It uses robust linear models, time series, analysis models, regression techniques, and discrete choice models, thereby making it prominent among similar data science libraries. It comes with a plotting function for the statistical analysis for achieving high-performance outcomes during the processing of the large statistical data sets.

18. Kivy: This is another open source Python library providing a natural user interface that may be accessed easily over Linux, Windows, or Android. The open source library is licensed under MIT, and it is quite helpful in the building of mobile apps along with multi-touch applications. In the beginning, the library was developed for the Kivy iOS and came with features such as graphics library. Extensive support is provided to the hardware with a keyboard, mouse, and a range of widgets. You can also use Kivy for creating custom widgets by applying it as an intermediate language.

19. PyQt: Another Python binding toolkit for being used as a cross-platform GUI is PyQt. PyQt is being implemented as the Python plugin. It is a free application licensed under the General Public License (GNU). It comes with around 440 classes and in excess of 6000 functions in order to make the user experience simpler. PyQt has classes to access SQL databases, active X

controller classes, an XML parser, SVG support, and several other useful resources for reducing user challenges.

20. OpenCV: This library is designed for driving the growth of real-time computation application development. The library is created by Intel, and the open source platform is licensed with BSD. It is free for use by anyone. OpenCV comes with 2D and 3D feature toolkits, mobile robotics, gesture recognition, SFM, Naive Bayes classifier, gradient boosting trees, AR boosting, motion tracking, segmentation, face recognition and object identification algorithms. Although OpenCV is written by using C++, it will provide binding with Python, Octave, and Java. This application is supported on FreeBSD, iOS, Windows, and Linux.

Chapter 2

About NumPy Arrays and Vectorized Computation

NumPy is short for Numerical Python, and it is one of the more significant foundational packages used for numerical computing of Python. Almost all computational packages provide scientific functionality by using NumPy array objects like lingua franca used for data exchange. Here are the things you can use with NumPy.

- ndarray which is an efficient multidimensional array which provides a fast array oriented arithmetic operation with flexible broadcasting capabilities.
- Math functions used for quick operations on the entire array of data without the requirement for writing loops.
- Tools for reading and writing the array data to a disk and working by using memory and mapped files.
- Capabilities are provided for random number generation, linear algebra, and Fourier transformation.
- A C API used for connecting the NumPy with libraries developed by using C, C++, and FORTRON.

A simple to use C API is provided by NumPy, and so it is straightforward to pass the data to external libraries written in lower level languages. It also enables the external libraries to return data as NumPy arrays to Python. This

feature makes Python the language of choice for wrapping of legacy C/C++/FORTRAN code bases and providing them a simple, dynamic, and easy-to-use interface.

Although NumPy does not give scientific or modeling functionalities, being blessed with the understanding of NumPy arrays along with the array-oriented computing will aid you in using tools having array-oriented semantics such as Pandas a lot more effectively. As NumPy is a big topic, we will cover the fundamental features here. We will see the main functional areas related to data analytics in this chapter. These include

- Quick Vectorized array operation used for data munging and cleaning, transformation, subsetting and filtering, and other types of computations.
- Usual array algorithms used such as unique, sorting, and set operations.
- Efficient descriptive aggregating and statistics and summarizing the data.
- Relational data manipulations and data alignment for joining and merging the heterogeneous datasets together.
- Expressing of conditional logic as array expressions rather than loops having if-elif-else branches.
- Group-wise data manipulation such as transformation, aggregation, and function application.

Although NumPy provides a computational foundation for normal numerical data processing, several users wish to use pandas as their basis for almost all kinds of analytics and statistics, especially in the case of tabular data. Pandas

will also provide greater domain specific functionalities such as time series manipulation that is not there in NumPy.

Remember that the Python array-oriented computing goes way back to 1995 when the Numeric library was created by Jim Hugunin. Over the period of next 10 years, several scientific programming communities started doing array programming by using Python. However, the library ecosystem had become fragmented during the early 2000s. In the year 2005, a person by the name of Travis Oliphant was successful in forging the NumPy from the then Numeric and Numarray projects for bringing together the community around one array computing network.

One of the main reasons why the NumPy is so significant towards the numerical computations of Python is that it is designed for efficiency regarding the large data arrays. You can find a number of reasons for it such as these,

- There is a contiguous block of memory available in NumPy where it stores the memory independent of other Python objects that are available built-in. The NumPy library containing algorithms is written in C and can operate on this particular memory with no type checking or other overheads. The NumPy arrays utilize a lot less memory than compared to the built-in Python sequences.
- Complex computations can be performed by NumPy on the entire arrays without needing Python for the loops.

For providing an idea regarding the difference in performance, you can consider the NumPy array with a million integers and an equivalent Python list.

```
In [7]: import numpy as np
```

```
In [8]: our_arr = np.arange(1000000)
```

```
In [9]: my_list = list(range(1000000))
```

Now, let's multiple every sequence by 2.

```
In [10]: %time for _ in range(10): our_array2 = our_arr * 2
```

CPU times: user 20 ms, sys: 10 ms, total: 30 ms

Wall time: 31.3 ms

```
In [11]: %time for _ in range(10): my_list2 = [x * 2 for x in my_list]
```

CPU times: user 680 ms, sys: 180 ms, total: 860 ms

Wall time: 861 ms^[2]

The NumPy-based algorithms are normally ten to a hundred times quicker than their Python counterparts, and they use substantially less memory.

ndarray of NumPy! The multidimensional array object!

NumPy has many features, and one of the key ones is the ndarray (N-dimensional array object) ndarray. It is a quick and flexible container for the large datasets of Python. The arrays enable you to do mathematical operations on whole sets of data by using a syntax similar to the equivalent operations of scalar elements. For providing you an idea about how the NumPy facilitates batch computations by using similar syntax to that of the scalar values of built-in Python objects, you will first have to import NumPy and develop a small array containing random data.

```
In [12]: import numpy as np
```

```
# Generate some random data
```

```
In [13]: data = np.random.randn(2, 3)
```

```
In [14]: data
```


Out[14]:

```
array([[ -0.2047,  0.4789, -0.5194],  
       [-0.5557,  1.9658,  1.3934]])[3]
```

Then we will write the mathematical operations of the data,

In [15]: data * 10

Out[15]:

```
array([[ -2.0471,  4.7894, -5.1944],  
       [-5.5573, 19.6578, 13.9341]])
```

In [16]: data + data

Out[16]:

```
array([[ -0.4094,  0.9579, -1.0389],  
       [-1.1115,  3.9316,  2.7868]])[4]
```

In your first instance, all the code elements are multiplied by 10. And in the 2nd, the corresponding values of every cell of the array are added to each other. Remember, in this chapter the standard NumPy convention is used as `np` for import numpy. The reader is welcome to use `from numpy import *` for your code to avoid writing “np.” However, you are advised against making a habit of it. The namespace numpy is big and consists of a number of functions having names that conflict with the built-in Python functions such as `min` and `max`.

The `ndarray` is a generic and multidimensional homogeneous data container. It means that all its elements have to be of the same type. Each array comes with a `shape` and a tuple, which indicates the size of every dimension. It also

comes with a dtype, and an object describing the data type of an array.

```
In [17]: data.shape
```

```
Out[17]: (2, 3)
```

```
In [18]: data.dtype
```

```
Out[18]: dtype('float64')
```

We will see some fundamentals for using the NumPy arrays, and that should be sufficient to go along with the remaining part of the book. Although it is not necessary to have an in-depth understanding of NumPy for the data analytical applications, it will certainly help to become proficient in array-oriented programming and thought process. It is one of the important aspects of becoming a scientific Python guru. If you see an array, numPy array or an ndarray in your text, they are all referring to the same thing barring a few exceptions which are ndarray object

Creating ndarrays

The simplest way of creating an array by using the array function. It accepts all sequence like objects including other arrays and produces new NumPy arrays containing passed data. For instance, the list is a good candidate for converting.

```
In [19]: data1 = [6, 7.5, 8, 0, 1]
```

```
In [20]: array1 = np.array(data1)
```

```
In [21]: array1
```

```
Out[21]: array([ 6. ,  7.5,  8. ,  0. ,  1. ])[5]
```

Nested sequences such as a list containing other equal-length lists can be

converted into multidimensional arrays.

```
In [22]: data2 = [[1, 2, 3, 4], [5, 6, 7, 8]]
```

```
In [23]: array2 = np.array(data2)
```

```
In [24]: array2
```

```
Out[24]:
```

```
array([[1, 2, 3, 4],
```

```
[5, 6, 7, 8]])[6]
```

Here as `data2` was a list containing other lists, NumPy array `array2` has 2 dimensions with the shape inferred from data. You can confirm it by inspecting `ndim` and `shape` attributes.

```
In [25]: array2.ndim
```

```
Out[25]: 2
```

```
In [26]: array2.shape
```

```
Out[26]: (2, 4)
```

Unless it is explicitly specified the `np.array` will try to infer a good data type for an array which it creates. A data type is stored in special `dtype` metadata object for instance. For instance in the examples specified above we have,

```
In [27]: array1.dtype
```

```
Out[27]: dtype('float64')
```

```
In [28]: array2.dtype
```

```
Out[28]: dtype('int64')
```

In addition to the `np.array` you can find a number of other functions for the creation of new arrays. For example, `zeros` and `ones` can create arrays of 0s and 1s respectively with provided shape or length. An "empty" will create an array without initializing a value to a specific value. For creating a high dimensional array by using these methods you can pass a tuple for a shape,

```
In [29]: np.zeros(10)
```

```
Out[29]: array([ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.])
```

```
In [30]: np.zeros((3, 6))
```

```
Out[30]:
```

```
array([[ 0.,  0.,  0.,  0.,  0.,  0.],
```

```
[ 0.,  0.,  0.,  0.,  0.,  0.],
```

```
[ 0.,  0.,  0.,  0.,  0.,  0.]])
```

```
In [31]: np.empty((2, 3, 2))
```

```
Out[31]:
```

```
array([[[ 0.,  0.],
```

```
[ 0.,  0.],
```

```
[ 0.,  0.]],
```

```
[[ 0.,  0.],
```

```
[ 0.,  0.],
```

```
[ 0.,  0.]])[7]
```

It is not safe to assume here that the `np.empty` can return an array containing all zeros. In many cases it will return uninitialized garbage values. An

“arrange” happens to be the array valued version of a built-in “range” function of Python.

```
In [32]: np.arange(15)
```

```
Out[32]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```

You can see the standard array creation functions in the table below. As NumPy concentrates on numerical computing its data type in case it is not specified will be float64 (floating point) in many cases.

Table for Array Creation Functions

Function	Description
----------	-------------

array	Converts the input data (tuple, list, or an array, or another sequence type) to a ndarray either by surmising a dtype or explicitly stating a dtype; also by default copies the input data.
-------	---

asarray	Converts the input to ndarray, however, does not copy in case the input happens to be an ndarray.
---------	---

arange	Similar to a built-in range, however, it returns a ndarray rather than a list.
--------	--

ones,

ones_like	Produces an array of all 1s having a given shape and dtype; ones_like will take another array and will produce a ones array having the same shape and dtype.
-----------	--

zeros	Similar to the ones and ones_like, however, will only produce arrays of 0s instead.
-------	---

zeros_like

empty, Will create new arrays with an allocation of new memory, but does not populate with any values such as ones and zeros.

empty_like

full,

full_like Produces an array of a given shape and dtype having all values set to the indicated “fill value.” The full_like will take another array and will produce a filled array of the same shape and dtype.

eye, identity Will create a square $N \times N$ identity matrix with 1s on the diagonal and 0s everywhere else. [\[8\]](#)

Data types required for ndarrays

A dtype or data type is a special object which contains information or metadata. This is required by the ndarray for interpreting memory chunks as a specific kind of data.

```
In [33]: array1 = np.array([1, 2, 3], dtype=np.float64)
```

```
In [34]: array2 = np.array([1, 2, 3], dtype=np.int32)
```

```
In [35]: array1.dtype
```

```
Out[35]: dtype('float64')
```

```
In [36]: array2.dtype
```

```
Out[36]: dtype('int32')\[9\]
```

Dtypes happen to be the source for the NumPy flexibility while interacting with the data that comes from other systems. In almost all cases, they will provide mapping directly on the underlying disks or memory representation. It makes things easy for reading and writing binary streams of data to the disk

and also for connecting to the code written in any low-level language such as FORTRON or C. Numeric dtypes are named in a similar manner. They have a type name such as an int or a float which is followed by a number which indicates the number of bits per every element. Any standard double precision float value of the kind used under the Python hood of the floating object takes up 8 or 64 bits. Therefore the type is called as float64 in NumPy. Look at the table below for the full listing of data types supported by NumPy. But do not bother memorizing the NumPy dtypes especially in case you are a new user. Often it is necessary to take care of the general data you are handling whether it is complex, Boolean, floating point, integer, string, or another general Python object. In case you want to control how data is put into storage, then you can pick the storage type for it with ease in python.

Chapter 3

Improve the System



The core foundation of Lean is in production. However, Lean principles can also be applied to other disciplines. Lean practices have specific guidelines on what needs to be done, and it cannot be directly transferred from a manufacturing plant to a software development industry. Most efforts carried out to transfer Lean production to software development have failed because good software does not assume a production process. A good software depends on the development process.

Software development is different from production. You can consider development as a means of preparing a recipe while production as the process of adhering to a recipe. In other words, they are different activities and must be conducted using different models. Preparing a recipe is a trial and error process. Therefore, you can look at it as part of a learning process. The table below shows the difference between production and development.

Development	Production
<i>Designs the Recipe</i>	<i>Produces the Dish</i>
<ul style="list-style-type: none">• Quality is fitness for use• Variable results are good• Iteration generates value	<ul style="list-style-type: none">• Quality is conformance to requirements• Variable results are bad• Iteration generates waste (called rework)

Quality in Software Development

When it comes to quality in software development, the final result should enhance system integrity. System integrity involves both perceived integrity and conceptual integrity. Perceived integrity describes a product that

accomplishes usability, function balance, reliability, and economy that enhances customers. On the other hand, conceptual integrity implies that the key concepts of a system work together in a cohesive way.

Clients of a software system will look at perceived integrity as an easy solution to a majority of their problems. No matter whether the problem changes over time or is dependent on external factors, a system that has perceived integrity continues to provide solutions to problems in a cost-effective way. Therefore, improvements in the design imply a ready to use instead of something that conforms to the requirements.

Getting It Right the First Time

To solve problems that are yet to be solved, it is important to produce information. With problems that are complicated, the best approach is to apply a scientific method. This method involves the creation of a hypothesis, observation, and development of an experiment to test the hypothesis. The best features of a scientific method are that the hypothesis will always be right, and you will not learn more about it.

There are two kinds of concepts when it comes to creating a software. The first encourages developers to be confident that every design and code segment is correct the first time. The other concept encourages small, try-it and fix-it cycles to ensure that the design and code are correct the first time. In the first concept, there is little room for knowledge production. One of the reasons for this is because the generation of knowledge has to happen through reviews and deliberation.

Let your major objective be to create a balance between deliberation and review. To achieve this, consider how you can produce the most information. For example, when the cost of testing increases, you may want to generate

more knowledge via a review and deliberation. Or if experimentation is cheap and produces better knowledge, then this could be the right technique to use. Often, a combination of peer review, iteration, and experimentation have always the best results.

Cycles in Learning

Most of the time, the main problem that has to be solved needs to be understood by the right people in the business. This means that it is correct to have such people in the focus groups. It is also critical that you speak with the business people using a representation that they can understand quickly. There are different ways that you can demonstrate a system. You can choose to use models or prototypes; the most important thing is to use a demo that has the most knowledge. Many people understand something fast by seeing a working demo than just reading a document. Therefore, having a working software will result in more knowledge.

Using iteration and refactoring while you design is important. In fact, this has been found to be one of the most effective means to produce knowledge and discover solutions to problems early. Additionally, it helps one build a system that has integrity because the model produces knowledge that is appropriate for problems not defined well. In general, the right option is to ensure that you have a lot of short learning cycles.

5 Tips You Can Apply to Amplify Learning

Amplify learning is a major principle of Lean software development. Software development is an advanced process. Every day, there are new things that arise which have to be fixed. Some of these things could be changes in the market, user behavior, and the type of technology used. Certain occasions arise when one feels like it is hard to keep up. This is the

reason why enhancing an organization and teams is a strategic advantage.

Learning in software development can be broken down into three different meanings.

1. Learn a new technology. This means you reach that point where you can use a new technology to develop a product.
2. Learn what users want. In this case, you understand and adopt the changes in behavior, which are not directly involved in software.
3. Learn a new skill. This implies that you reach a point where you are able to apply a new skill when you are under pressure.

Different mechanisms support faster learning and can be used for various kinds of learning.

1. **Make use of social learning**

Learning is the default option for most developers. Learning through social gatherings include:

- **Receiving feedback via code review.** A shorter feedback loop will allow faster learning. The easiest way to achieve this is by writing an application in the new technology and sending it for review by other people who can be experts in that technology. The final result of the discussion will increase the level of understanding. Pair programming could be a better option because it is a real-time code review.
- **Look for a mentor.** There are certain occasions when developers are asked to learn a new technology. The end result is that developers have a person they can get back to ask questions when they get stuck. In addition, they have someone present to review the code when it starts. This kind of association is the best to accept by a lot of

programmers. It is better than having to display your code to strangers in the social community.

- **Use community networks.** After every two weeks, the AgileWorks community has an event called “Code and Beer.” The purpose of the event is to allow software developers to learn whatever they want from any person who attends.

2. Show the technology to an audience

Nothing can quicken learning than the chance to teach or explain a new technology to an audience. This often causes the mind to remain focused and scan through dozens of articles. An individual will further develop a few code samples that they can use to demonstrate their idea. Explanations become clear, and the presenter is likely to request for help. Obviously, the audience should be friendly so that the presenter is not overcome by the fear of public speaking.

To learn a new skill, you need to ensure that after some years you become a master. When you master a skill, you are able to apply it even when under pressure or emergency. In the software development industry, the use of a new skill is quite different from the use of a new technology. As a developer, when you find yourself under pressure, you have no other option but to write the code. However, you can avoid writing a unit test. This key difference implies that there are different strategies that should be used to master a new skill.

3. Intentional practice

Your deliberate effort is the difference between becoming an expert performer and normal performer. Expert performers combine the following aspects.

- Repetition

- Intention
- Apply current skill level
- Aim at enhancing performance
- Combine immediate feedback

4. Find a coach or take part in sessions

Constant practice is difficult to design. The only way is to have it created for you and facilitate someone who can ensure that the experience remains seamless to all attendees who can send immediate feedback.

In the brutal refactoring approach, a facilitator will review a code written by attendees. Once the facilitator can identify a problem, they pin a sticky note on their desk with additional information. The rules of this game are that developers aren't permitted to generate any additional code, but they are free to fix the problem. The coach can choose to work with a team or a developer based on the needs at hand.

5. Experiment

The Lean Startup movement always develops small experiments, measure results and respond to learnings. Designing and implementing an experiment requires one to have specific skills. This means that a team has to get ready beforehand.

Continuous Improvement

This is the major key for one to stay relevant and competitive in the market. It involves regular improvements and experimenting on all levels. Developing the ability to learn faster is a key advantage.

However, a lot of managers struggle so much. The major challenge to continuous learning is mastering the value.

Chapter 4

Create Quality

Quality leads to seeing all kinds of waste. A waste in testing a code more than once. A waste of logging issues as well as a waste in fixing the problem. Therefore, Lean principles seek to solve these issues.

If you take time to familiarize yourself with Agile methodologies such as Extreme Programming and Scrum, there are a few things that can help you accomplish quality.

First, there are several quality assurance approaches that can help one deal with issues related to quality. Example of these approaches includes Test Driven Development and Pair Programming.

Pair Programming

This addresses issues related to quality by taking advantage of two developers to deal with a single task. This means there is an overall benefit from a collective combined experience of two developers rather than one. Usually, the end result is increased productivity because two developer minds can see solutions that they can't see on their own. Pair programming further results in enhanced quality because one individual could be ahead in thinking of the other person.

Test Driven Development

It solves issues related to quality by writing tests before the code is written. You can think about a test analyst who writes down a test condition for every

feature before it is created. If the developer knows how the testing is done, then they are likely to write a code that should be tested. The code shall address all cases. In its complex nature, Extreme Programming stabs the code out and writes automated unit tests for every test condition before the code is written. The developer finally writes the code that can pass the test.

All these two practices originate from Extreme Programming and aim to prevent problems that arise due to quality.

Regular Feedback to Inspect and Adapt

Scrum and XP create quality using different processes. Choosing to develop using small incremental steps and collaboration, it creates an opportunity for a two-way feedback between a team and product owner. This kind of feedback is helpful when you inspect a product every day to ensure that there is a great level of quality as well as the right quality.

Reduce Time Between Stages

This is another amazing technique to help create quality in the development process. It involves reducing the time between testing, development and fixing a bug. Instead of logging bugs, it is better to deal with the bugs immediately. Logging bugs simply add waste into the system. If you have the tester ready to test the code and the developer ready to fix the bug, there is no value in logging a bug. Don't forget that presence of a long interval between creating a code, testing it, and fixing it results in loss of continuity. This translates to delays from switching between tasks, the absence of focus and knowledge gaps.

Automation

In the Agile development practices, one is encouraged to apply automated regression testing. This type of practice is rare in Agile development; despite

this, it is a good means to reduce effort related to discovering issues related to quality before they can show up.

It is through automation that both XP and Scrum have managed to achieve Lean principles in the software development. In addition, this is how quality is generated in the process.

Constant Integration

A lot of the Agile practices encourage the adoption of regular builds most of the time. It can be on a daily basis or an hourly basis. Extreme Programming encourages repeated integration. The code that is integrated into the whole system is then built and automatically tested once it is checked in. Reducing this gap between builds helps eliminate another type of waste. With complex Waterfall projects, both the integration and regression testing stages of the project can be very long. However, constant integration eliminates this problem.

Controlling Trade-Offs

It is important to note that quality is just a single dimension in a project. Other factors include cost, time, and scope. There are cases when a commercial reason to trade off quality against other factors exists. In this case, be careful to watch out for instances where a focus on cost is more than issues that you want to avoid.

An example of a situation where Agile adopts this principle is in the acceptance of refactoring due to a detailed spec. Traditionally, the methodology practices had been designed to enhance quality early in the life cycle of the project. But as years passed, these methods were found not to be that productive and hence Agile methods were invented.

In the same way, if you have a fairly low-complex project that has a low

impact, it may be better not to spend a lot of time in building quality because there is a low chance of issues related to quality to arise in this kind of project. This is left for you to decide because it can be very difficult to tell.

Building quality in a software project is necessary, or you may end up creating several wastes. Build quality as early as possible in your project to prevent a ripple effect of issues related to quality from affecting your project. In addition, make it a practice to build quality in the whole development process.

Build Quality in the Software Delivery

Make Use of Scenarios

To figure out how a software can solve a given problem, it is important to have a detailed user scenario created for the developer. Developers should write down short notes about this and then presented in a formal way. One of the major reasons why big tech giants such as Microsoft and Apple have one of the best software is because they are mainly concerned about user scenarios and finding solutions to user problems.

Create Regression Tests

Developers know that better software must be tested continuously and improved. It is possible to create multiple tests while you progress in the development phase. And that is the reason why having an automated regression testing is critical to improving the quality of a software. The process is easy and simple. It involves testing and retesting a software until you reach the final phase of the product. Just make sure that the entire process is free of errors and seamless. If an individual is building a software, perform multiple regression tests by automation. Remember, many programming errors develop because developers are on a different page from

testers. The coding could be poorly done without adhering to the rules of coding.

Design

New beginners in the field of software development make big mistakes of picking easy designs to solve any problem that pops up in the development. However, most of these simple solutions don't work well in the real-world scenario.

Once a comprehensive user scenario is provided, a senior developer can tell how to apply a strategy and fill in the best practices by gathering data from users. Difficult aspects such as comparison of data would be clear in this phase.

When you plan to develop a high-quality software that is reliable, make sure that you have a strong design that can accommodate changes in future. Change might appear difficult to make, but it has far many benefits.

Edit a Poor Design

Just remember that poor designs shall arise even in a large software developed by many developers. Therefore, you should be ready to refactor the design to make it perfect.

At the start, you might be afraid to change the code even though the design is poor. What you must recall is that poorly designed things can operate in the short run, and it can do a lot of harm in the lifespan of the project. That is the reason why it is advised not to think that making edits in a code is a wasteful effort. There is a great possibility that the change you make may make the software better for users. Lastly, adopt quality in every process of your software development. Repairing defects has no value to clients but can reduce the value when it shows up in the production.

Chapter 5

Decide as Late as Possible

In the process of transforming sheet metal into a car body, a huge machine has to press the metal into shape. This machine contains a massive metal die that gets in contact with the sheet metal and compresses it into a shape of a door or fender. The process of designing and cutting the dies so that it develops the right shape is about half the capital investment of a new car development program and powers the vital path. When a mistake destroys a die, the whole development program is affected. If there is something that automakers always want to do right is the cutting and die design.

The issue is while the car development continues, engineers continue to make new car changes. And these changes finally find their way to the die design. Therefore, regardless of how long engineers can attempt to decompress the design, they cannot manage.

The strategy that the U.S. applied in creating a die was to wait until that point when the design specs were frozen, then they could send the final design to the tool and die maker. This then activated the process of requesting for the block of steel and cutting it. However, small a change was it had to go through a rigorous change of approval. This usually lasted for two years after the ordering of steel before the die is applied in production.

But in Japan, this process started immediately. It was better referred to as concurrent development. It worked because die engineers in Japan were expected to be aware of a die for a front door panel will consist, and they had

to be in constant communication with the body engineer. They have to anticipate the end result and be skilled in techniques to create changes in the development. In most cases, die engineers have the ability to deal with the engineering design as it changes. In case of a mistake, a new die has to be developed quickly because the entire process is restructured.

The Japanese automakers don't eliminate design points until when it is very late in the development process. This provides an opportunity for a lot of changes to take place. If it is compared to the previous design freeze actions carried out by the United States in the early '80s. The Japanese die makers spent more money on the changes and created better die designs.

As a result of the many benefits and advantages that the Japanese automakers experienced from their manufacturing process, the U.S. automotive companies had to shift to concurrent development practices in the '90s. This led to the reduction of the product development performance gap by a huge figure.

Concurrent Software Development

When you look at the whole process of die cutting, that is how programming seems to be. There is a lot of input and effort set, and mistakes can be very costly. As a way to prevent mistakes, requirements are identified before development starts. This is also called sequential development. It has both its negatives and positives. One of the negatives of this form of development is that designers have to take a depth-first instead of a breadth-first type of design. The depth-first requires one to make low-level decisions before they can experience the results of a high-level decision. Costly mistakes occur when one ignores or forgets to take into consideration something vital. One

way of finding yourself making this mistake is rushing down to details fast. Remember, once the details are identified, it is very difficult to go back. Therefore, when a huge mistake is done, it is better to review the landscape and take time before making a comprehensive decision.

Concurrent software development assumes an iterative development. It is the right approach to take when you have a critical situation. With concurrent development, it gives you the freedom to apply a breadth-first approach and realize the costly problems before things get worse. The process of shifting from sequential development to concurrent development implies that one has to start by programming advanced features once a high-level conceptual design has been designed. This can appear as a counterintuitive, but you can look at it as an explanatory mechanism which allows an individual to learn by testing different options.

Besides delivery of insurance against critical and costly errors, concurrent development is the correct way to handle dynamic requirements. And the reason for this is that both big and smaller decisions are deferred. In case change becomes investable, concurrent development will cut down on the delivery time and the entire cost. At the same time, the performance of the final product is going to improve.

For concurrent development to be successful, there are a few skills that need to be considered. In the sequential development, the U.S. automakers alienated die engineers from the automotive engineers, similar to programmers in a sequential development process usually have little contact with the customers and users. Concurrent development in the die cutting prompted the U.S. automakers to develop two major changes. The engineer had to have the expertise to postulate what the emerging design required in the cut steel as well as to collaborate with the body engineer.

In the same manner, concurrent software development calls for developers that have the right expertise to predict the place where the new design may end. In addition, it requires a close interaction with the analysts and customers that design how a system can operate or solve a business problem.

Rise in Cost

The software is not the same as other products that have to be redefined or upgraded every time. In most cases, more than half of the development work happens in a software system after it has been sold in the production. Apart from the internal changes, a software system is subject to a changing surrounding. Virtually all software is designed to change in the course of its lifetime. Additionally, once a software upgrade stops, it is then about to reach the end of its life. This leads to a new type of waste created by software, and it is hard to change.

Initially, there used to be a rise in cost in the product development. It had been assumed that a change in the production would result in a thousand more times when a change is caused in the previous design. That belief that cost rises when the development increases contribute highly to standardizing the sequential development process. Nobody identified that the sequential process may be the result of the increased cost escalation. But once concurrent development replaced sequential development in the United States, cost escalation talks came to an end. It was not again about how a change may impact a development, the discussion focused on the way to reduce the need for change via concurrent engineering.

Some changes aren't the same. There are some architectural decisions that one requires to have at the start of the development because it fixes the problems in the system. A few examples include architectural layering decisions, choice of the language, and many more. Since most of these

decisions are important, one should concentrate on the number of high-stakes cases. You also need to assume a breadth-first approach to most of the high-stakes decisions.

The greatest change in a system doesn't need to have a high-cost escalation factor. Instead, the sequential approach causes a majority of the cost in a system to change abnormally as one goes through the development. With sequential development, it emphasizes finding all the decisions made quickly, this means that the cost of all changes remains the same.

Lean software development takes time before it can freeze all design decisions. And the reason for this is because it is easy to change a decision that is not yet made. It focuses on creating a strong design that accepts changes easily. The design has to be adapted to most of the existing changes.

The major reason why software changes in the entire lifecycle are because business processes change every time. There are specific domains that change faster compared to others, while other domains might be stable. It is hard to create flexibility that can handle all the arbitrary changes. The right idea is to create tolerance to deal with changes into the system along the domain dimensions that might want to change. Identifying where changes take place in the iterative development process sends a good signal to determine if a system requires flexibility in the future. In case there are frequent changes in the development, one should expect these changes not to end once the product is produced. The thing is to learn more about the domain to ensure that there is flexibility as well as prevent letting things be complex.

When a system is created after the emergence of a design via iterations, the design shall remain robust, this will make it adapt quickly to the changes that happen during the development. In fact, the ability to adapt will remain on the system so that multiple changes can take place after it has been released.

And this also allows easy incorporation. Conversely, a system that is created with the intention to ensure everything is right so that it can cut down on the costs, chances are that the design may be brittle and not easy to accept the changes. The worst scenario is that chances of making critical mistakes in the main decisions increase with a depth-first instead of a breadth-first approach.

Concurrent software development implies beginning the process of development with just partial requirements. In addition, it involves developing a short iteration that delivers feedback that causes a system to emerge. With concurrent development, it is possible for one to delay commitment until that required moment. This is the moment when one fails to make a decision removes an important option. Therefore, when commitments are delayed beyond the required moment, then decisions become a default option. Overall, this is never the right path.

Remember, procrastinating is different from making decisions at the critical moment. As a matter of fact, taking time before making a decision is a hard work. Below are important tactics to help make decisions at the most critical times.

- **Arrange for a face to face employee collaboration.**

Releasing incomplete information early implies that the design must be redefined while development continues. This needs people who can understand the little details that the system should do. To ensure that you deliver value, good communication is important. Make it a practice to communicate face-to-face with people so that you can develop an understanding of how the code operates.

- **Share partial information about the design information.**

The idea that a design must be complete before it is made available is the

greatest enemy of concurrent development. The need for a complete information before the release of a design increases the period of the feedback and may result in irreversible decisions made early. The right design encompasses a journey of discovery that goes through brief and repeated cycles.

- **Create a sense of the way you can absorb changes.**

When it comes to delaying a commitment, the distinction between experts and amateurs is that experts are aware of how they can delay commitments as well as how they can hide their errors for as long as they want. Most of the time, experts fix any error before it results in a problem while amateurs will work hard to get everything right the first time. This has a tendency to overload the problem-solving ability. As a result, they commit a lot of mistakes and make wrong decisions. Below are some tips one can use to delay commitment in the software development industry.

- **Create modules.** Object-oriented focuses on hiding information. In this case, you have to delay commitment associated with the internal design of a module until that point when client requirements stabilize.
- **Develop parameters.** Create magic numbers. These are like constants that have a meaning. So you should create a magic function such as third-party middleware and databases into parameters. Passing capabilities into modules bound in a simple interface, your reliance on specific implementation are removed and testing becomes easy.
- **Don't apply sequential programming.** It is important to apply a declarative programming technique rather than a procedural

programming. You have to go for flexibility instead of performance. Algorithms have to be defined such that they don't rely on a specific style of execution.

- **Create abstraction.** This is very popular in the object-oriented concepts. Both abstraction and commitment are inverse processes. It is good to defer commitments to certain representations as long as the abstract will create an immediate design.
- **Be on the lookout for custom tool building.** By investing in frameworks and many other tools calls for an early commitment to the implementation details. This usually ends up complicating things. It is important to use frameworks from a collection of successful implementations.
- **Separate concerns.** To use this tactic, make sure that every module has a single and well-defined role. In this case, a class has just one reason to change.
- **Don't repeat.** It is a principle that is very important. Popularly referred to as don't repeat yourself. It allows for only one place to change in case there are some changes that may be required in the future. This helps prevent inconsistency in the code.
- **Encapsulate variation.** The thing that is most likely to change should remain inside. The changes should not extend to other modules. In this tactic, one has to have an in-depth knowledge of the domain so that they can predict stable and variable aspects.
- **Don't adopt extra features.** If you decide not to add additional features that you might need, then it means you are not ready to use those features. Remember, extra features come with an extra burden

to test a code and maintain a code. Additional features bring more complexity rather than flexibility.

- **Avoid future implementations.** Choose to implement only the simplest code that will fulfill your immediate needs. Don't opt to implement future implementations. Wait until that moment come before you can do so. In fact, when that time comes, you will know better.
- **Develop a feeling of when these decisions have to be made.** You don't want to make a default decision or even delay decisions. However, there are specific architectural ideas such as packaging a component, layering, and usability design that have to be created early to ensure the rest of the design expands. Ending up with a late commitment should not result in not making a commitment. It is important to create a sensible feeling associated with the timing as well as a mechanism that will trigger decisions to happen when that time comes.
- **Try to figure out what is important in the domain.** The greatest fear that comes with sequential development is remembering important features of a system when it is very late. Therefore, if you have the response time or security as key features of the system. These concepts have to be addressed from the start. Ignoring them until it is too late makes everything costly. The perception about sequential development as the best means to identify these features is false. Practically, early commitments may not determine these vital elements than late commitments.
- **Create a quick response mechanism.** The rate at which you respond to an event or situation is important. If you respond slowly,

you will have enough time to make decisions. For example, the Dell company can assemble computers in a week. This means that they have less than a week to decide what to develop before they can ship. But other computer manufacturers take time to assemble computers, and therefore, they have a short time to decide what to make. Therefore, if you are able to change your software quickly, you can just wait to learn what your customers want and then make that change.

Problem Solving Strategies

Depth First vs. Breadth-First

Depth-first and breadth-first are the two strategies to use in problem-solving. You can assume breadth-first as a funnel while depth-first as a tunnel. Delaying commitments is a key feature with breadth-first while early commitments are the major feature of depth-first. People are born differently. There are those who prefer the breadth-first while others prefer the other approach. Besides that, when faced with new problems, most people go with the depth-first because this technique reduces the problem complexity. The design is an example of a new problem, and as such, novice designers select a depth-first approach.

The drawback of the depth-first approach is that the domain under consideration gets narrow fast, especially when the people involved in making commitments aren't experts in that domain. So if a change is a must, the work of exploring the details disappears. In other words, this technique comes at a large cost.

You should also note that both depth-first and breadth-first technique call for some expertise in the domain. A depth-first technique works when there is a

perfectly selected part for one to focus. To get the selection correctly calls for two things. An individual with the right experience to make early decisions and a great assurance that no changes will arise to eliminate the decisions.

A breadth-first approach needs a person who can understand how details will arise and be able to identify the time to make commitments. A breadth-first technique does not require the presence of a stable domain.

Simple Rules in Software Development

Simple rules allow people to make decisions. These aren't instructions to dictate to people what they need to do. One can look at it as a set of principles that can be applied differently in different domains. Experienced people use simple rules to make decisions. Everyone has a limit to what he or she has to consider first before making a decision. Therefore, simple rules should be based on a few major principles.

Chapter 6

Fast Delivery

Fast delivery does not mean that you rush at a breaking speed. This is basically an operational practice which will offer a competitive advantage. Customers love fast delivery. This means that when a company improves its delivery, its competitors will definitely do the same.

Why Should You Deliver Fast?

Every customer loves fast delivery. For this reason, instant shipping is the standard for online products. Fast delivery gives time to customers before they make decisions. To some, there is that feeling of satisfaction that comes up. In the software development, rapid delivery results in an increase in the business flexibility.

Fast delivery does not benefit customers but also businesses. Fast delivery means that a company delivers a product faster before a customer can make their decision. It means that companies have fewer resources as a work in progress.

When there is a massive work in progress, it creates more risk. Whether a problem is small or large, it will be considered as work that is half done. If developers write a code that they haven't tested, defects continue to increase. A written code that is not integrated results in a high-risk effort. To reduce these risks, an individual is supposed to cut down on the value stream.

Therefore, you can see that the principle of fast delivery complements

decision making. The quicker you are at deliveries, the longer you can take before making a decision. For example, when you know that you can make changes in a software in just one week, then you have no hurry to choose what you want to do. You can wait until that moment arrives before you can make any change. Conversely, if you can spend a whole month making changes, then it is good to decide on the changes a month before. Fast delivery is a friendly approach to software development. Your options remain open while you cut down on the uncertainty and make informed decisions.

It is important to note that fast delivery cannot occur as an accident. When people report to work, they have to know how they are going to spend their whole day. Every person has to know what he or she can do to improve the business. If an individual doesn't know what to do, productivity is affected, time is lost, and fast delivery is not realized.

If you want to make sure that your employees remain effective in the use of their time. You can decide to tell them what they need to do or prepare things such that they can make decisions by themselves. In a fast environment, the second option is the right one. People who are used to handle emergency or fluid incidences, don't wait for the leader to tell them what they need to do. They simply make decisions on their own depending on the situation at hand.

When incidences occur, there is little time for information to flow down the chain of command and return as a directive. This means that there should be methods set to coordinate the work. One way to achieve this is by allowing customers to do the work instead of having a work plan to push the work.

Software Development Schedules

In the current world of advanced software development, the question is always how one can ensure that the people who report to work use their time

in the most effective way. Failing to have better procedures to help one understand this for themselves forces project managers to depend on schedules. Additionally, they change it depending on their knowledge and tell developers what they would want to do.

However, the main problem is that a project schedule can never be perfect. It will continue to be unreliable like the way a manufacturing schedule is. In addition, instructing developers on what they need to do cannot lead to more motivation.

Certain times you may hear complaints related to micromanagement in the software industry. Managers may choose to deliver a comprehensive instruction to developers when the work is not organized. In a complicated system, when resources are inadequate and deadlines approach, then everyone should be productive. So the big question is: how can people stay productive without being reminded?

Even a schedule is not effective in defining complex assignments. Waiting for a computerized schedule to generate assignments or tell developers what to do is not the right way to deal with complex situations. The most effective way is to choose to apply a pull system which can develop the correct signal and commitment. This should allow team members to determine for themselves the most productive way they can use their time.

In the software pull system, software development features short iterations which depend on the input of the customer at the start of each stage. You can assume that at the start of the iteration, a customer specifies or writes down the descriptions they want in an index card. Then developers estimate the time required to implement. Finally, the customers prioritize the cards. In this case, the cards reveal to the developers what they should achieve in a given period. The point here is to ensure that the work is self-directing. This means

that no card is assigned to developers, but it is the developers who select the cards they want.

The cards could just be posted in a specific area where developers can walk in to read what needs to be done. Developers working on the tasks transfer the cards to a checked area alongside the name of the card. Once an implementation passes a test, the card is transferred to a different part.

As you can see, the cards notify the developer about what to do, but they aren't enough to achieve this task. Planning for a regular meeting each day is also a great idea to ensure that the iteration remains self-directing. This daily meeting should be less than fifteen minutes and include team members alone. Every member of the team has to be present. Active participation in this meeting should be left to the team members.

During this meeting, team members deliver a summary of what they have done in the previous day and what they plan to do on that day. In addition, if there is an area that you need help, this is the right time to speak. In case there are problems that cannot be dealt with in this fifteen-minute meeting, they can be discussed later with the interested party.

Information Producers

Visual control is an important feature of a pull system. For work to be self-directing, it is important for every person to see what is happening, what needs to be done, existing problems and progress being made. It is difficult for workers to be self-directing when simple visual control isn't in place.

Cycle Time

There is a lot of time spent in queues. Whether it is in traffic jams, long lines in a store, or waiting for a tax refund. There is a queueing theory that aims to reduce the wait time as possible. The theory helps calculate the number of

servers one should have in a computer room.

Cycle time is the main measurement of a queue. This is the average time for something to move from one end to another. This time begins when something enters a queue and continues to count as it waits in the queue. Remember, any time you are in a queue, your aim is to have a short cycle time. No one wants to stand on the queue for a long time because there is something that he or she needs to achieve. The only hindrance to achieving your goal is because the resources necessary are limited.

There are two ways one can reduce the cycle time. The first one is by examining how the work arrives, and the other one involves determining how to process the work. In some cases, it is not easy to affect the rate of arrival at work.

One way that one can control the rate of arrival of work is by developing small work packages. If you may need to wait for a large list of work to arrive before you can begin to process, the queue will be at least as long as the entire batch. When the same work is produced in small batches, chances are that the queue will be smaller.

Can you take, for instance, the bottleneck in the testing department? You need somebody who can run acceptance tests for a project every day instead of a suite of tests each month. Are you able to negotiate the same amount of hours over a month and ensure that there is a continuous rate of testing to be done?

Most software firms control work using a review process. In this process, priorities are defined, and a project is selected. Therefore, in a yearly event that is bounded by the budget, a year's worth of work arrives once. This creates long queues. While most managers still think it is a good thing to

compile projects into a single priority-setting process, queueing theory recommends releasing a project either weekly or monthly.

When you eliminate variability from work that arrives in a queue, the next thing is to exclude variability from the processing time since a smaller package contains few things which can go wrong. However, small work package can cause a lot of problems when it comes to figuring out the amount of time each work package can take. The correct way to solve this problem is by increasing the number of servers that can compile work in a single queue. Both the airport and banks have no mechanism to determine the type of customer that may take a lot of time. In this case, they choose to reduce variability.

When there is a process that has multiple steps, then it will affect the processing time and rate at which work flows in other stations. A big processing variation in the upstream workstations cascades the system. This means it is a great thing to transfer variability downstream.

The seriousness of upstream variation becomes clear in an iterative development. For instance, if you have a problem with the acceptance testing. This is the last step before deployment; this problem won't slow down any previous work. However, if you are going to perform iterations, acceptance testing is never going to be the last step. It is an important aspect of each iteration and has to be done before moving to the next iteration. Skipping this step will make you not receive feedback, and that is a major objective of iteration. So in iterative development, acceptance testing shifts upstream, and delays are reflected in other iterations. Thus, one should ensure that there is no bottleneck during testing.

Slack

The right way that one can deal with the cycle time is by having enough capacity to process the work. Short cycle times aren't possible when resources are used up. It is obvious that traffic slows down when all roads are used up.

To Deliver Fast, You Must Think Small

To ensure that you deliver fast in software development, it is important to get your team to think and deliver small. Thinking small borrows a few Agile development concepts.

1. Smaller work units

Ensure that your work units are small and can be accomplished in a single sprint. When you have a long story, try and break it up. There are a lot of ways that you can divide a story. Some of those techniques include:

- **CRUD.** It involves create, read, update and delete. These are the four main operations that are carried out on the data object.
- **Acceptance criteria.** When a story has a lot of acceptance criteria, use those criteria to split the story.
- **Multiple data objects.** When a story has the same operation to many data objects, it is possible to split the story for every data object.
- **Steps of a workflow.** A story that has more than one step in a workflow is possible to break that story into individual steps.
- **Supported technology platforms.** In case the function of your story requires to run on different target platforms, try and divide the story by platform. Ensure you target the most used platform.

Note:

Sometimes, it is tempting to split a story by task, architectural layer,

dev story, or design story. It is not advised to apply these approaches. The reason is that splitting stories using this method doesn't deliver a complete production slice of a usable software. Therefore, these approaches don't create any value for each story. There is never any business value in a design or UI because it is difficult for a user to interact with it or deliver feedback.

2. Little work in progress

Another wonderful technique that can help one deliver fast is to reduce the work of the team. Teams struggle a lot to get stories finished at the end of each sprint, and most of the time these stories remain in progress. When there is so much work in progress, the progress slows down because most of the team members have to multitask.

Instead of having a lot of work in progress, stories can be moved from one sprint to another. Concentrate on a few in-progress stories and get them done. There are several techniques that one can apply to help solve this:

1. If you plan a sprint, remember to organize the order by which stories are going to be implemented and delivered in every sprint.
2. Define work-in-progress limits and don't begin on a new story until all in-progress stories are done.

Applying either of these will help your team to collaborate and get the work done fast.

3. Maximize the throughput.

When you optimize the work in progress, you will get a lower utilization of individuals. That is a good thing. The goal is to ensure that the work product is faster.

Chapter 7

Trust and Team Empowerment

Trust and empowerment are the major aspects that power technology teams. Companies such as Amazon, Spotify, and Facebook have discovered this. They apply Lean software development principles in each stage of development. These two elements are a cutting edge. It allows a company to realize a continuous framework of delivery and produce quality software every time.

Well, what do you think is the relation between trust and empowerment and Lean software delivery? The truth is that there is no way you can implement a Lean software development framework without trusting and empowering teams. Lean allows a management to tell a team when a problem occurs. However, it also allows the team to come up with a solution. Lean allows a manager to implement a software for production. This implies that there is no delay or waiting for a given date to release or let the management deploy a software.

With Lean, once a software has passed testing, it is deployed immediately for production. That sounds great. Big companies such as Spotify have done this for many years. When you want to remain competitive, you must aim to apply Lean delivery practices.

But most companies do the opposite. They have long release dates and cycles. You will find that they take more than a month to deliver a new software to the customer. They stick to a fixed release management schedule

that has lots of controls. The entire process is occupied with waste which lags behind the delivery of a functioning software.

The basic principle behind Lean software development is to remove waste. Waste refers to anything that has no value to the customer. When you take a look at the software development system in an organization, chances are that you will encounter a lot of waste. Waste slows down the process of delivery of a working software.

This is one of the reasons why management has to take charge. It needs to trust and empower their teams. The major strength of a high-performing software company originates from teams. According to Mary Poppendieck, an author and expert on Lean software development, “Top-notch execution lies in getting the details right, and no one understands the details better than the people who actually do the work.”

Trust and empowerment are the major keys of a great team. It is the main factor which drives a team forward. It supports Lean software development in an Agile framework. The final result of Lean software development is customers who are satisfied. A satisfied customer is an important element in the software development process. When you have customers that are happy with the services, then you know you have achieved quality.

Team Empowerment

Traditionally, teamwork requires a project manager to control everything. The manager has total control over all decisions and plans. It is where a single person is accorded all the power. However, when there is a hardworking team, it becomes difficult and seems to prevent a team from achieving its full potential. So it is important not to spoil things with a single authority. Make necessary changes by shifting the power from one person

and empower the whole team.

Who is going to be accountable? Who is going to be responsible? Who is the leader? Everyone. The most critical thing on any project is having the right leadership. The last and not so important is who is leading. Leadership and management are very different. They aren't the same thing even though it might look similar. When it comes to leadership, you identify a problem and get people together to determine the solution. Any member of a team with the ability to identify a problem has the ability to lead people to find a solution. And that is your responsibility.

It might appear counter intuitive, even so, a calm and confident team can be very decisive without the presence of a boss. There are so many jokes said about a committee finding it difficult to create a good design. Bad designs can occur to any team when the vision of a product is not strong and well defined.

When you are a customer of a Lean project, it is not easy because in most cases, you will find yourself in leadership roles. You have to participate actively in most discussions so that you can ensure that your vision is implemented. You will always find yourself in uncomfortable situations that require you to calm a hotly debated discussion. You have to decide the time when the team's code is ready for deployment. The most challenging part comes when you need to set priorities and make hard decisions of what has to be done and what won't. This is different from appending your signature at the front page of a requirement spec.

When a customer requests to meet the software team, it may be that they are building a wrong software. Find out from the customer the most important thing that prompts him or her to meet the team.

The duty of a developer or programmer is to program, test, and design a system. The responsibilities of a team include getting organized and deciding on what has to be done next. As team members, it is good to accept to lead and search for possible problems while eliminating risks. This is really different from just writing code depending on the instruction stated.

As a manager, there will be a lot of responsibilities. One of the most critical roles is to watch as the team interacts. When quarrels arise then the manager has to fix it. When a customer is not using a consistent product vision, the manager has to step in and act. A wrong customer can take a team in circles and have nothing done. A developer who interferes with the flow of the whole process can bring down the whole project. To replace a customer or developer is not an easy thing. One must do it with a lot of caution.

People develop an emotional link with their projects. Both Agile and Lean projects are critical because team members work tightly. There are people who may be scared of getting hurt when a project fails and attempt to protect their emotions before they get hurt. It may appear as if people want to make a project fail. Managers may not know why team members would want to interfere with a project and ensure that it does not succeed. In this case, they may decide to replace the project. However, it is advised for managers to ask themselves why people are scared of a given project, and this could help fix a problem.

One of the duties of management that may not change is obstacle removal. The only thing that changes is that you can't decide what an obstacle is and what must be acknowledged. A team will raise different issues that you need to act on best on your abilities.

Furthermore, the manager has to make a decision about whether a team will continue to remain focused during an iteration. The manager has the role to

schedule meetings and has to do so wisely. However, it is important to realize that so many planning meetings can result in chaos, and nothing is going to get done.

It is critical for each team member to be familiar with a distributed leadership. Most people who make decisions do it when facing it. Evidence from Lean development indicates that delaying a decision until the last moment when you are fully aware of a given impact is the best idea.

A highly organized team is not easy to find, and it calls for discipline from every member. A good leader does not choose to do whatever they want when they want. A team member who is not disciplined has the highest control by not adhering to the process. You need to define and fulfill the expectations of a team. Other team members also need to show trust in you and know what they expect from you.

There are decisions that are going to be made, and it might not go well. Decision made by who reaches first has huge consequences. One has to accept that decisions have to be made. And not all decisions will be made by you. Your role as an empowered team member is to know the kind of discussions that go on. You have the responsibility to participate and contribute your ideas.

The right thing that will enhance team empowerment is to avoid a blame. There are some team members that will fear to make their points because they think someone is going to bully them. So it is good to avoid blame by making it known that no one is interested in listening to it.

The greatest misunderstandings of team empowerment are that teams become empowered once agility is triggered. The team begins to adhere to a set of expectations among team members. Many people hate rules because rules

inhibit an individual. Therefore, you should define simple rules that focus on interfaces between team members. For example, the daily meetings. You can set a rule to have the meeting in the same place and time for not more than fifteen minutes.

Finally, you will start to experience the ups and down of leadership. You will learn to trust what reluctant leaders say because you know they see something you undermined. You remain organized around problems without any prompt. You will develop a comprehensive knowledge about team empowerment to the point where you can build your process to mirror your company and project very easily.

In case it appears like problems may arise, then know that you are right unless you change your mode of communication. If you have a single control point, you might just need a single person who is aware of everything. It is possible for team members to stay for months without communicating with each other as long as there is one person who knows everything and can control everything. However, this is not always the case. In very small projects, this type of command easily breaks down. This means that one should eliminate barriers for the project to work.

Motivation and Purpose

In most cases, you will find a group of people who join hands to accomplish something. There is excitement in the air since people are ready to conquer the impossible. Every member is fully involved in the task and committed to the purpose. Passion creates the right atmosphere that anything is possible.

Kenneth Thomas, author of *Intrinsic Motivation at Work* discovered that people are hardwired to focus a lot on purpose. According to Thomas, there is a huge evidence that shows people go through lots of pain when they don't

have a purpose. Intrinsic motivation originates from the work performed and the feeling of assisting a customer. The purpose then makes work more engaging and energizing.

People need something more than just a collection of tasks. If their work is to deliver intrinsic motivation, then they must know as well as commit to the purpose of the work. This type of motivation is powerful when a team commits together to achieve a given purpose that they care about. Below are some things that one can do to help a team gain and develop a sense of purpose:

1. Begin with a clear and more critical purpose

A successful team should have a champion ready to channel the vision of a new product so that they can get stakeholders. Team members that are dedicated to a critical purpose can collaborate with a passion to table their product.

2. Provide customers with access to a team

When you speak to real customers, it is a great thing among team members. This creates a chance for the team members to understand the purpose of what they are doing. It becomes very important when they are able to see how the software should simplify life for real people. This further makes team members to develop insight into how their individual work relates to the big picture.

3. Be confident that the purpose is achieved

The basic rule of empowerment is to make sure that a team has the ability to achieve a given purpose in its work. If a team is dedicated to fulfilling a business objective, it must have access to the resources required to achieve that goal.

4. Allow a team to make its own commitment

At the start of an iteration, the team should discuss with customers so that they can understand their priorities and choose the next iteration. No one should make an assumption of how much work has to be done to finish. A team has to make a call and when the members remain dedicated to a set of features, it is equal to being committed.

5. Avoid skeptics

There is nothing worse than having a person who believes that nothing can be done and they have a lot of good reasons to say. The team does not require to listen to this kind of person.

The Foundations of Motivation

Intrinsic motivation is created by a self-determination and a sense of purpose. But it cannot progress in a hostile environment. According to research, intrinsic motivation requires a sense of competence, a feeling of safety, and a sense of progress. Let's look at each in detail.

Safety

The quickest way to kill motivation is the zero-defects mentality. This is the kind of environment that doesn't allow even a single mistake. Perfection is an important element even in the smallest detail. The zero defects mentality is viewed as a serious problem in leadership because it destroys the energy required to create success. In the software development, every team member has delegated some responsibilities to do. This calls for a lot of tolerance because there are certain situations when things just don't work as it is expected. Mistakes will happen. However, when a person is right, the mistakes they make may not be that serious in the long run compared to mistakes made by management when it decides to dictate those in authority

on how they have to accomplish the job.

Competence

People usually believe that they can do a great job. So they want to participate in something that they know can work. It is a great thing to belong to a winning team. If you are in an environment that does not generate a sense of freedom, a sense of doom develops. A software development environment should be disciplined so that work can continue well. Great practices such as coding in the right code repository and an automated testing are important for a rapid development. Another important thing is the mechanism to share ideas and enhance designs.

A feeling of competence originates from skill and knowledge, better standards, positive feedback, and interaction with a difficult challenge.

Belong Somewhere

The modern work environment allows a team to accomplish a lot of things. On a stable and healthy team, everyone knows what the goal of a project is and is dedicated to ensuring it succeeds. Team members show respect to each other and remain honest. Lastly, the team must win or lose as a group. Showering individuals credit instead of a team is a sure way to kill team motivation.

Milestone Achieved

In a motivated team, it won't take long before members achieve something. This reinforces the purpose and keeps everyone fired up. Once a team attains a specific objective, that is the time for them to celebrate. Team members are always happy with little achievements. They congratulate their colleagues as they have some fan.

Chapter 8

Integrity

Integrity in software development exists of two types:

1. Perceived integrity
2. Conceptual integrity

Perceived integrity describes a product that achieves a balance of function, reliability, usability, and economy that impresses customers. On the other hand, conceptual integrity refers to how the major concepts of a system work to achieve cohesiveness.

Perceived integrity depends mostly on the experience of the customer when he or she uses the entire system. This could be how the system is advertised, accessed, installed, and how it can handle changes in the domain among many other factors.

One can measure perceived integrity by determining the market share of the application. To help you understand perceived integrity, consider this case.

If you were to uninstall all your computer application, which one will you install first?

That represents something which you perceive to be necessary for your life.

Conceptual integrity is a requirement of perceived integrity. A system without a uniform design has issues to do with usability. Users always want a system with a consistent design idea.

Conceptual design pops up as the system development continues. Even

though conceptual integrity is a necessity for perceived integrity, it is not enough. When the best design fails to meet the users' needs, it becomes difficult for users to identify the conceptual integrity. This is the reason why the design and architecture of a system have to change. New features must be added into the system to ensure that perceived integrity remains. While new features are added into the system, it is also important to add new features that support the cohesive nature of the system.

The Goal to the Integrity

Integrity is realized through excellence and comprehensive flow of information. Perceived integrity refers to the flow of information both from customers and developers. On the other hand, conceptual integrity refers to the underlying flow of technical information.

The method of building a system that combines great conceptual integrity and perceived integrity is to ensure that information flows are excellent. The flow of information has to be good both for the customer to the development team. In addition, all upstream and downstream processes must also be fluent. The flow of information has to include the present and possible application of the system.

Below are three major requirements that must be present to enhance the development performance.

- Accepting change as a normal process and the ability to sustain emergence design decisions.
- An environment that allows communication to support information, tools, and people.
- A boost in the use of application knowledge across all software development members.

Perceived Integrity

In general, most decisions that are made daily affect perceived integrity. Organizations that successfully realize perceived integrity have the means to ensure that the customer values remain key before the technical team that is vested with making design decisions. This is mostly the duty of the chief engineer in many organizations. These are people who have that special skills to listen and understand their prospective customers. Besides that, a chief engineer should possess the right leadership skills to channel the vision of the company to people who have to make decisions and tradeoffs.

The procedural software development tries to transfer the concept of perceived integrity to developers using various processes. The first thing is that the requirements have to be collected from customers and put down. The same requirements are analyzed by people apart from those who collected the requirements. Analysis helps one understand the technical terms, what the requirements mean by using different models. In the traditional model, the analysis does not cover implementation details. Instead, it just improves the requirements further. After analysis, a design is created that describes how the software should be implemented. The role of creating a design is done by a separate group of people. Once the design is over, it is transferred to the programmer to write the code.

As you can see, sequential software development has some challenges even if the customer can describe the problem and then a different person document the requirements. Requirements must be put down and handed to analysts. The analysts then hand another document to designers to design the software. Once the software design is created, it is handed over to the programmer. The programmer is tasked with making the final decision about how the code has to be written. This means that there are about three documents that are

handed over before the programmer finally gets the last document. So you can be sure that at every stage of the transition, some considerable information is lost or misinterpreted.

In this case, there is no chief engineer or master developer who has a good understanding of what the customer values and the types of decisions that the programmer has to make. No one can update the programmer about the changes the customer wants to be made and help them make certain key decisions that will contribute to the general integrity of the system. In this case, a large percentage of this system will lack perceived integrity because there is no detailed flow of information from the customer to the developer. What could be the solution to this? Well, there are different techniques that one can apply to achieve first-class customer developer information flow.

- Perform customer test to acquire customer communication
- Let smaller systems be created by a single team that has a direct communication with the people to assess the integrity of the system. The team must apply short iterations which have an extensive category of people skilled in different types of integrity.
- When building a large system, it should be represented using models that both the customer and programmer can understand.
- Large systems should again have a master developer with an extensive knowledge about how a customer understands a system and excellent technical details. The role of the master is to represent the customer and provide directions related to the design.

These techniques can benefit top master developers the same way chief engineers gain from regular prototypes. The most important thing to ensure is that all customer tests must be created and should exactly describe the working of the system. These tests are key to customers because they will

help them understand how the system can work.

Create a Model Approach Kind of Design

In this kind of design, a software model is created to help in the implementation. It is important to ensure that domain models can be understood by customers and developers who write the code. Both developers and customers should aim to use words that mean the same things. By doing this way, both the customer and developer will understand each other. This type of model is the best to use for complex and large projects because it gives everyone the freedom to speak their language.

Models demonstrate how a system will appear to users and how it can handle important concepts. The right type of model depends on the domain and how the details are abstracted into a correct format.

Below are some models that you can use to represent the flow of customer information to the developer.

- **Use case model.** Both the glossary and domain model are static views. With a use case model, it shows a dynamic view of the system and how important it is at illustrating usability. It showcases the customer goals and sub-goals and how they interact with the domain model.
- **Glossary.** This describes terms contained in the domain model. The main purpose is to ensure that both the developer and the customer can speak a similar language. This document further contains policies, semantics, and rules in the domain model.
- **Conceptual domain model.** This kind of model may contain basic entities in the system. Entities in a system could range from events, transactions, documents, and many more. It could be anything as long

as it has the major concepts from user's model and the relation between among these models. This model does not need to be very detailed but should have the major ideas and concepts. Its purpose is to show how users understand the system.

Those developers tasked with the role to write business layer and presentation layer can use any of the above models without the need to change anything. When they are discussing the same concept, both the customer and developer must learn to use words originating from the domain. When a model is altered or translated, there is a lot of information that is lost in the process. Furthermore, building a software that reflects the domain model is going to be stronger. This software is able to withstand the changing business needs.

One way that you can predict if a model is useful is by looking to determine whether it is up to date. It is believed that you have to update a model every time so that it can continue to be used. If a model is not used, it is no longer updated. This means that you can develop models that are useful for a short time before they are outdated. However, it is a waste if you are going to create a model and update it because it is a great thing. But you will know that you have developed a good model when most people can reference it.

While using a model, make sure to look at it from a detailed perspective where it interacts with the customer. The best thing to do is by beginning with an advanced level of abstraction and adding more detail when you want to implement a given area.

Besides that, it is good to realize that people can only handle a few concepts at a time. Therefore, minimize communication to only a handful of concepts each time. The secret to communication in complicated systems is to hide details in the abstraction. Models are important when you want to build abstractions and allow communication in an extensive topic. Iterations are

very important to activate the movement from abstractions to implementation details.

How to Maintain the Perceived Integrity?

A good customer developer information flow is not enough to showcase the need for the application to change. Majority of the software programs are dynamic. It is important for systems to adapt to both technical change and business change in an economic style.

The correct way to ensure that change exists in a system is to ensure that the development process accepts change. The only fear with the application of iterative development is that later iterations may bring additions that might need a change in the design. But when a system is developed with the notion that each aspect has to be known before an effective design is realized, then it will be ready for a change in the future. If you apply a change tolerant process, then you can be sure to have a system that is acceptable to change.

Ensuring that institutional memory continues to exist is the secret to having long-term memory. Some companies and organizations have tried to apply documentation that was used in the design. However, this has one disadvantage that the design documentation cannot reflect the system as it was originally built. As a result, it is ignored by many programmers. To ensure that institutional memory remains, let developers be accountable for regular updates. Conversely, both the developer and programmer can work together to transfer knowledge.

Conceptual Integrity

In this type of design, the major concepts of a system have to work in a cohesive manner. The components should match and work together. The architecture has to have an effective balance between efficiency, flexibility,

maintainability, and responsiveness. The architecture of a system describes how a system is developed to deliver the required capabilities and features. The right architecture presents a system with a correct conceptual integrity.

How to Maintain Conceptual Integrity

1. Clarity

It is important for the code to be easy to understand by all those who are going to interact with it. Each element has to communicate and describe itself without the need for comments. Use naming conventions that can be understood by everybody. Have code that is clear to be understood easily.

2. Suitability to use

Each design must fulfill its function. A fork that creates a lot of problems when you want to eat is not well designed. An interface that is not intuitive is not great to be used in a consumer website. Any time tests indicate that performance has reduced to an unacceptable level, the problem must be addressed quickly even if it requires the design to change.

3. Simplicity

In each field, a simple working design is the right one to use. Experienced developers know the way to expand a complicated code, and many software development patterns are aimed at finding complexity to a very difficult problem.

4. No repetition

An identical code should never be present in two or more places. Repetition represents an emerging pattern and must send a notification to request for changes in the design. When changes are done in more than one area, the possibility for error increases. This means that duplication can be a bigger

enemy to flexibility.

5. **No additional features**

When the function of a code expires, the waste that originates from it is very large. That waste needs to be kept, integrated, compiled tested every time the code is applied. The same applies to features that may be used in future. Most of the time when you anticipate for the future, it fails. It is possible to take an option on the future by choosing to delay decisions.

Great design will always change in the course of a system. However, this may not happen accidentally. A poor code cannot get better when it is ignored. If a developer finds a wrong thing in the code, he or she should end the line. The team should spend some time to find and fix the source of the problem before moving on with the development.

An important refactoring calls for a sensible design. Teams that are inexperienced are known to repeatedly change the code without the need to enhance the design.

Chapter 9

Optimize the Whole



The best software project does not just depend on the proper functioning of every part of the life cycle. This includes coding, testing, analysis, and deployment. The success of a software project depends on how all these pieces fit together.

Optimizing the whole refers to how you can ensure that everything measures up. Enhancing your business is part of a sub-optimization. You can spend all energy improving the unit testing experience but never come to tell that it's your interactions that lag down the project. Or you might struggle to improve the delivery process of a team while the rest of the organization continues to work in a wasteful manner. In both cases, it is advised to take time to review the whole situation. If you want to optimize the unit tests using Groovy, you must take time to figure out how you can enhance the entire project. If you have a team that you are working to create a better life cycle, you have to step back and find out how you will improve the lifecycle of the team.

Writing a code involves many things besides creating unit tests. Even more important, you have to be aware of the type of tests you need to write as well as what the customer wants. This calls for developers to work closely with the customers. In most cases, it is the customer who documents how the system should behave, developers then re-document it using functional tests. Story cards are generated as well as user input tests produced.

Simple stories assume three states: pending, failing, and passing. Failing and

passing must be obvious. Pending implies that a story has to be written but not worked on.

Apart from collaboration, a common issue in optimization is the need to have developers build a perfect environment to code. This is always another challenge when it comes to collaboration that seems to affect most teams.

Fortunately, one can choose to use Gradle. This is an enterprise build tool that is created around the philosophy to develop simple things as well as complex things. Gradle has a convention that is built by default.

System Thinking

This considers an organization as a system. It determines how the different parts of an organization relate and the way an organization as a whole performs. Once this analysis is over, then it is referred to as system dynamics. Software development is a dynamic process. This process is often rigorous and assumes a sequential processing. Documents call for a written customer approval, changes in control as well as tracing every requirement in the code. In case an organization does not have a basic development discipline, the application of an in-depth sequential process might improve the situation. In system thinking, simply because everything looks better does not mean that whatever you are doing is correct. Don't forget that delaying effects that come with a sequential process may later start to show up, this is the time when you will find it difficult to maintain the system with the current needs of the customer. At this point, if you push for a more rigorous procedural process, it may result in a downward spiral.

The most basic pattern present in system thinking is the limits to growth. While a process creates a given result, it develops a secondary effect which balances and lags behind success. By continuing to push for the same

process, it will expand the secondary effect and result in a downward spiral. Searching and removing growth limits is the key concept in the theory of constraints. The point is to identify and eliminate the present constraint to growth. You have to realize that the current constraint to growth will shift to a different place once it is addressed.

Another pattern present in system thinking is the shift in burden. This pattern consists of an underlying problem that results in symptoms that are difficult to ignore. Since the problem is hard to address, people try to solve the symptoms rather than the source of the problem. However, these quick fixes make the underlying problem to become worse.

System Measurements

System measurements help one track the software development progress. For instance, the number of defects in a system is important in measuring the readiness of a software before it is released. But it is important to use information measurements rather than performance measurements. You can acquire information measurements by combining data to hide the individual performance. A defect measurement system may result in a performance measurement system if the defects are attributed to the individual. It only turns out to be an informational system in case the defects affect the features.

Well, why is it important to monitor defects in a system? First, it helps developers improve the performance of the system. The challenge with attributing defects to developers is that an assumption is made that individuals are the root cause of the defects. However, this is not true because the source of the defects exists in the software development and procedures. This means that pointing the defects to individual developers is equivalent to shifting the burden. Attributing defects to individuals means that it is difficult to identify the root of the problem. The correct way to find the root cause of a

problem is by empowering the whole development team to search for it. Inferring defects in individuals discourages these collaborations.

A Lean organization always aims at optimizing the entire value stream but not just a few functions. Big delays in a project and process are common. Communication issues also are a common thing in projects. Misunderstanding will exist as a result of handoffs between team members and organizations. The point is that crossing boundaries in an organization are very easy.

The best principle of Agile methodologies that arises from this experience is the concept that to organize teams to achieve a complete, co-located product team with skills and roles, it is a must to deliver a request from start to finish without referencing any teams.

This is not easy to achieve especially when you don't have the authority to restructure your organization. This is the reason why Agile has to be driven from the top. Despite this, the fact is many problems experienced in the traditional IT departments is a result of structuring teams around skills instead of products.

A team that is organized by product has certain unique advantages. Besides maximizing the workflow of a team and preventing issues from arising, a team that is organized in this form tends to have a better ownership of the product. In addition, it is responsible for creating better quality, innovation, and commitment. There is also a better team spirit and amazing cooperation between members of a team.

Chapter 10

Go Lean in Your Organization

It is important for an organization to adapt to change quickly while ensuring all personnel remains focused and productive. The same way that you can apply Lean processes in developing successful programs and products, it is the same way you can implement Lean to help manage people. This is referred to as Agile performance management.

Performance management has not experienced a lot of changes. Most companies such as Google embrace and shape new technologies that are appropriate to the changing world. Below are four tips that you can use to ensure that Lean performance management practices succeed in your organization.

- 1. Develop a monthly goal setting to allow all your employees to stay focused.**

In the current world, long-term goals are irrelevant because goals lose their relevancy as the environment changes. Conversely, if it is done correctly, goal setting can bring a lot of changes as well as provide new directions. It can ensure an organization remains focused and enhance results.

How to achieve this:

- The organization should have quarterly goals.
- The goals of the employees must be aligned with those of the organization.

- Metrics are present to track results.
- All goals should be measurable and impact-oriented.

In the implementation:

- Set aside extra time and effort to allow adoption in the first three months.
- Concentrate on achieving high-quality goals.
- Get the top management to buy in before you can implement anything.
- Use a software to monitor goals. Currently, there are several systems that one can use to effectively track organizational goals. However, don't use performance management systems created to support the traditional management process.

2. Develop an accountability mechanism that can enhance improvement

Better accountability systems are the key to high employee autonomy as well as positive performance. It is also the secret to reduced management overhead.

How to implement this:

You should monitor your monthly goals by applying measurable monthly goals discussed in the first tip. Measure it on a scale of 1-100 depending on the percentage of the monthly impact target that employee set for themselves. Most importantly, avoid communicating performance scores as a tool of evaluation but let it be a mechanism to monitor and support constant improvement. The major focus of performance goals should be to increase performance growth.

3. Enhance self-organization. Make use of just-in-time support systems that optimize self-development.

Most organizations don't have enough time allocated to formal employee training and development. Therefore, management has to build support systems and environment that can allow self-development to grow among employees.

How?

This is achieved by ensuring that there is trust on all levels. Resolve any issues that result in low trust among the leaders.

Learn JIT Coaching

- Offer immediate coaching as well as organizational support to underperformers
- Make sure that all employees have a good support network by doing three of these things:
 1. Peer coaches in the organization
 2. Good coaching relationships with the manager
 3. Have coaches and mentors outside the organization
 4. Practice semi-annual employee "retrospectives" but not "reviews"

There are several strong reasons that one can do to avoid traditional performance reviews.

1. Two-thirds of appraisals have negative effects on performance after the feedback meeting.
2. 87 percent of the employees and managers have this feeling that performance reviews aren't useful.

3. HR executives often measure their performance review system at “C” or below.

In short, you should never do anything that has proved not to work. This might have more negative effects. About 30 percent of performance reviews lower employee performance. However, there are many better alternatives.

How to achieve this:

Both the managers and direct report should have a meeting once a year to determine how to improve performance.

- Get feedback both to and from the customer.
- The aim of the retrospective meeting is to communicate performance levels.

Ideas for successful Lean implementation

1. Begin with action from the technical system.

To ensure that a strategic Lean approach works, process operators must work in the process associated with teams instead of current functional ones. It is important for teams to become self-directed and let problems select the people to solve them within the teams instead of the management selecting problems and allocating it to people to solve. This implies that you must begin to understand that Lean involves a change in the mindset. Many reasons why Lean implementation fails is because of not changing the management style.

2. Have members of a team trained properly?

This eliminates conflict and creates a management group that can allow change to take place. In addition, it will remove waste in an organization.

3. Make use of Kaizen workshops to educate and perform rapid changes

Get a talented and experienced facilitator who has an in-depth knowledge of Lean tools and philosophy. Ensure that this problem addresses a specific problem. This will make the training relevant to the real world scenario.

4. Make arrangements around value streams

In many organizations, management is designed based on the processes and functions. This means that managers own specific steps in the process but not the whole value stream.

5. Create communication and feedback channels

This will allow one to receive support via people participation at different levels by sharing concepts that boost the synergy to enhance a positive Lean development journey.

6. Get ready to handle middle management resistance in the implementation

Middle management has been identified as the number one cause of resistance to change in the Lean implementation.

7. Always start with a value stream pilot to showcase Lean as a system and deliver a model

The most important Lean tools are the “Value Stream Mapping”. If this tool is correctly used, it can let one create a map of both waste and value in a specific process. You can then use this map to understand the causes of waste before you can eliminate the waste to allow value to flow without any obstruction.

When you want to build the current state map, future state map, an action plan to apply in the implementation, search for a cross-functional group that has managers to authorize resources. In addition, look for doers to be part of the process being mapped. It is good for the value stream mapping to be used on a specific product of families that can be transformed immediately.

8. Establish a positive environment

You need to remain tolerant of mistakes that may happen in the process of Lean implementation. Develop a supportive learning attitude to ensure that development continues. Learn to be patient with progress since this will be important in ensuring that you get the results you want. Develop the courage to take risks at critical stages and do everything you can to ensure resources attain the correct plan.

Chapter 11

The Relationship Between Lean and Agile Development

Both Agile and Lean are popular practices in the software development industry. Lean and Agile help teams deliver fast and more productive results. Most teams in the software developers have little knowledge about the difference between Lean and Agile. Usually, the terms are used synonymously to explain or refer to a given set of practices. So, do you think you are Lean or Agile? Is it possible to be both Lean and Agile?

First, let's provide you with a brief answer: Agile development is a methodology that facilitates rapid delivery of a software, and it applies a lot of Lean principles.

The Connection Between Lean and Agile Principles

Agile development is associated with any development method that relates to the concepts in the Agile Manifesto. This is a foundational document written by fourteen top software experts. The Agile Manifesto provides directions in the way Agile software development should be implemented. It has three key concepts: an iterative approach to development, disciplined project management process, and short feedback loops. Below is a description in the way all the three concepts have a connection with Lean principles.

Iterative Approach

When you examine the Agile software development, teams use an iterative

approach to manage software projects. A working software is produced as fast as possible instead of waiting for large batches. Constant code deployment provides an opportunity for teams to practice Agile quickly and receive feedback from customers and stakeholders. This feedback is important because it determines how the future product will appear. Therefore, teams can apply late changes in the development process.

Connection with Lean: Deliver Fast and Delay Commitment

Iterative development is similar to the Lean principles of delivering fast and delays commitment. Lean advocates for teams to deliver fast by taking control of the flow and reducing the work in progress. By reducing work in progress, it helps limit context switching and enhance focus. Agile teams control the flow by working in collaboration to generate one iteration at a time.

The principle of Lean to delay commitment encourages Lean organizations to function like just-in-time systems, it waits until the last moment to make a decision. This allows Lean organizations to develop the agility to make important decisions using relevant and up-to-date information.

Disciplined Project Management Process

Agile depends on a disciplined project management process that advocates constant review and adaptation. This type of approach facilitates software development teams to concentrate on completing high-quality as well as high-value work fast. As a result, valuable insight is generated after each release. Towards the end of the iteration, teams take time to review opportunities for improvements based on the feedback from stakeholders.

Connection with Lean: Develop Quality

When you have a disciplined process, teams can practice the Lean principle

of developing quality. This concept is very simple. It involves automation and standardizing any tiresome and repeatable process. Or any process that is vulnerable to human mistakes. This principle makes Lean teams reduce errors in most of their processes. Therefore, teams are able to concentrate their effort and energy on creating value for their customers.

Short Feedback Loops

A short feedback loop makes teams concentrate on work that fulfills the most up-to-date business requirements. A popular principle in the Agile manifesto requires close interaction between customers, stakeholders, and developers. This allows Agile teams to take into consideration and complete a task depending on the goals of the company, and remove anything that is not important to the customer.

Connection with Lean: Remove waste

The concept of Lean advocates for this concept. If a customer is not going to pay for something, then it is a waste. A short feedback loop between Agile developers and their stakeholders makes teams develop a formula for eliminating processes, products, and activities that cannot generate customer value.

Lean and Agile Development

Agile will provide a chance for a software development team to deliver high-quality work, move faster and remain aligned with the business stakeholders. There are different ways that you can apply Agile methodologies such as XP, Scrum, and Kanban.

No matter the methodology your team selects, it is critical to understand the principles behind the method so that you can achieve sustainable and disciplined practice. In case you have a team that is implementing Agile but

not familiar with Lean principles, take some effort to find a way to enlighten them.

Chapter 12

Pros and Cons of Lean Software Development



There has been a significant change in software development in the last decade. New methods have been invented to help reduce the development time and handle costs. These new methods include rapid application development, spiral model, dynamic system development, and Agile development.

Lean is a subset of Agile software development. Its main focus is to increase the development cycle by adopting different principles. The first two important principles include waste elimination and improve learning. All of these principles have been discussed in detail in the different chapters. No need to mention each principle. A seamless design allows one to resolve any issues. All the seven principles of Lean discussed in this book sound pretty good. The principles improve development and speed of delivery while ensuring that fixing problems becomes easy.

Unfortunately, Lean software development is not different from other methodologies. It has both advantages and disadvantages.

Among the advantages of Lean software development include the removal of waste that helps save money and time. Furthermore, it supports additional functionality that shortens the period of delivery. In addition, it empowers the development team in making decisions related to processes. The end result is that there is an increase in motivation among team members.

The Lean methodology is scalable; this makes it a good option to apply in conventional software development methods which are mainly designed for large projects. Besides that, Lean works well with Agile because it fits across different teams as well as it integrates teams and promotes cooperation.

Although Lean seems to be an amazing software development approach. It has its own drawbacks. For example, it mainly depends on the team. This means that one should always have a team that is well trained with the right skills. Given that the whole team has many different responsibilities divided into smaller sub-teams, there is a chance to lose focus. In addition, Lean development calls for quality documentation especially when the development contains business requirements to be fulfilled. So, any area that is documented poorly can result in a poor system.

All in all, the advantages of Lean software development surpass the disadvantages. This is very true when the time for upgrade and addition of features comes. Therefore, make sure that you have the right team and let them adopt Lean.

Conclusion

Thank for making it through to the end of *Lean Software Development: Avoiding Project Mishaps*. Let's hope it was informative and able to provide you with all of the tools you need to achieve your goals whatever they may be.

The use of Lean principles in software development process calls for interpretation, and there are several ways that interpret Lean software development. There are ways which to concentrate on Lean principles applied in popular development practices while others depend on the workflow management. The principles of Lean thinking are a relevant abstract compared to the approaches used in Toyota cars.

Since software development is not assembling cars, it requires a means of interpretation to make sense about the Lean principles. The first school of thought that tries to describe Lean principles relies on native software development approach. If you can highlight methods that are analogous to concepts related to Lean, then this can be important in delivering value that Lean promises.

Lean software development process is used across the world. One of the major reasons why it is very popular is because it eliminates waste in the software development process, and it is a rapid means of developing software. If you focus mostly on the seven Lean software development principles, you can be sure to eliminate waste in the system and enhance the performance of the system. Lean software development advocates for rapid

detection of waste as well as fast removal of waste and its causes. Causes can include work that is done partially, extra code, extra processes, and many other features.

Now that you have learned and understood Lean software development, the next thing is to start implementing the above principles and practices in building software products. That is the best way to develop software products that address the needs of customers.

PYTHON DATA ANALYTICS

A Hands-on Guide Beyond The Basics

Travis Booth

Introduction


The most valuable and expensive entity in the 21st century is not Gold or Oil or Diamonds, but Data. The science revolving around information, more commonly known as Data Science, Analytics around data and Machine learning is growing and evolving at an exponential rate. Professionals who can see through this vast sea of data and help organize it in a manner that can be beneficial to a company are considered as the biggest assets to an organization. The farm of data if harvested efficiently, can help reap profits of the highest order for an organization. IBM predicts that the number of jobs for data scientists in the United States alone will increase to 2.7 million openings by 2020. It is for this reason that it is important to understand more about how you can work with and analyze data.

This book will take you through different aspects of data analytics. You will gather information about what data analytics is, and the different techniques one can implement for analytics. You will also learn more about Python and understand the basics of the language. It is important to have this information so you can build different models and learn more about how you can tweak the existing models.

Over the course of the book, you will also gather information on different data visualization techniques and tools. The book also leaves you with information about different algorithms you can use to perform data analytics.

Chapter One

An Introduction to Data Science And Data Analytics



Let us quickly brush through what we learned about Data Science in the first book of this series.

What Exactly is Data Science?

For over a decade, humans have been trying to best define data science. Hugh Conway created a Venn diagram in 2010, consisting of 3 circles that helps understand data science in the best possible manner. The three circles represent the following fields of knowledge.

- Math and Statistics
- Subject Knowledge (which is knowledge of the domain under observation)
- Hacking skills

The intersection of these 3 circles lead to the zone that accurately represents the field of data science. If an individual has expertise of all these 3 skills, it can be concluded that they are highly proficient in data science.

Data Science is a process where a huge amount of data is sorted by cleaning it, organizing it and then finally analyzing it to see if it can be useful. Data is available from various sources and what a data scientist does is they collect it from the available sources and then apply several external factors to it such as

predictive analysis, machine learning, sentiment analysis, etc. to retrieve data from these data sets which is of critical importance. The next objective of a data scientist is to understand this extracted data from the point of view of the business requirement and convert it into accurate insights and predictions that can eventually be used to power the decisions that are to be taken for the given business.

What Information is Necessary for a Data Scientist to Know?

Ideally, any individual who is looking to build a career in the field of data science should be efficient with skills and tools that will help them in the following three departments.

- Domain knowledge
- Analytics
- Programming

This is a broad classification of what is required of a data scientist. If you dive another level deeper, the skills listed down below will carve out the essentials of a data scientist in an individual.

- Very Good knowledge of programming languages like Scala, R, Python, SAS
- Proficiency and hands on experience in SQL databases
- Ability to collect and sort data from unstructured and unorganized sources like digital media and social media
- Understanding of various analytical functions
- Knowledge and curiosity about machine learning

Who is a Data Analyst?

A data analyst can be defined as an individual who can provide statistics that are basic and descriptive in nature, visualize and interpret data, and then convert it into data points so that a conclusion of that data can be drawn. It is assumed and expected that a data analyst has an understanding of statistics, has some or very good knowledge about databases, can create new views, and has the perception that is required to visualize data. It can be accepted that data analytics is the primary form of data science which is considered to be much deeper and more evolved.

Skills required to become a Data Analyst

What do we expect from a data analyst? It is imperative that a data analyst has the ability to take a particular topic or a particular question and should be able to put forward the raw data in a format that can be understood comfortably by the stakeholders in a given company. The four key skills listed below are essential if you are looking to jump into the path that leads to becoming a data analyst.

- Thorough knowledge of mathematical statistics
- Fluency in understanding programming languages such as R and Python
- Understanding of PIG/HIVE
- Data Wrangling

Do Data Analytics and Data Science Intersect?

Data science is a superset that contains the small subsets of data mining, data analytics, machine learning and other disciplines related to it. A data analyst primarily extracts data from all available sources of information while a data scientist has the skills that help him or her understand the patterns in the

extracted data and forecast and predict behaviors based on the past patterns in a data set. A data scientist therefore, creates queries while a data analyst helps find answers to those queries.

Understanding Machine Learning

We can define machine learning as the practice of creating and implementing algorithms to use the data that we have at hand, learn from that data and forecast and predict future trends for a given topic. Machine learning traditionally revolves around statistical analysis coupled with predictive analysis to understand patterns in data sets that can help us identify insights that are usually hidden in the data that is collected. Let us try and understand this in simple terms with the help of an example.

Let us understand how machine learning has been implemented in the most popular social media website in the world today, Facebook. Facebook has machine learning algorithms which constantly study a user's behavior on the website. Based on the past behavior, the algorithm understands the nature of a user, which helps it know about a user's interests. It studies a user's past behavior based on what the user has liked in the past, which pages the user has followed, and then predicts what other articles of similar interest would be relevant to the user and displays it in from of them on their news feed. This is similar to Amazon where in when a user purchases a product, Amazon's algorithms quickly suggests other relevant products that the user may want to buy. Another good example on machine learning is when you watch Netflix and based on the kind of movies a user has watched in the past, Netflix starts suggest relevant movies belonging to the same genre on the user's home page.

Skills needed to become a Machine Learning Expert

Machine learning is just a digital approach to statistics. To carve out a career

in the Machine Learning domain, the following skills are considered to be essential.

- Knowledge of computer fundamentals
- Expertise in programming skills
- Experience in probability and statistics
- Evaluation skills and data modeling

Do Machine Learning and Data Science Intersect?

We have already established that data science is a superset, which consists of various disciplines, and we can say that even machine learning is a subset of data science. Processes such as clustering and regression are used in the field of machine learning. While on the other hand, it is not necessary that the data in the data science requires backing from a machine or a process that is mechanical. The main point of differentiating between the two is that data science is a broader aspect, which not only looks at statistics and algorithms, but also looks at the complete method of processing the data.

We can therefore say and believe that data science is an amalgamation of several disciplines, which include software engineering, machine learning, data analysis, data engineering, business analytics, predictive analysis, and more. The world of data science includes the processes of extraction, collection, analysis, and transformation of huge data, which can be otherwise called as Big Data. This big data is organized and given a proper structure using data science. Data science helps understand and establish patterns in large sets of data that eventually help stakeholders to make appropriate decisions for a business. Data science consumes numerous disciplines for its use, and data science and machine learning are two such examples.

Data analytics, data science and machine learning are domains that have a very high demand in the industry today. A combination of skill sets associated with these three domains will help you pursue a strong career in the twenty first century.

Evolution of Data Science

It is a well-established fact based on survey that 89 percent of data scientists love their job. 49 percent of data scientists receive a connect from recruiters every week asking them if they are looking for a job. Machine learning jobs are also on the constant increase in the world and the increase in data is what is making these jobs even more demanding. If you open up LinkedIn, you are likely to find employment that is related to machine learning and data science. The internet as a whole creates 2.3 quintillion bytes of data and information every single day. Until a few years ago, very few people knew about data science but now it has become very popular on account of its demand in the industry. Through the years, the field of data science has gained so much popularity that we are now looking at business analysts who are beginners and students who are working on building their own set of strategies and analysis programs. Information development has pushed methods that were traditional to extremes resulting into the reformation of software engineering in the domain of deep learning.

The meaning of what it means to be a data scientist has been overlooked in recent times though. As per DJ Patil, who is an American computer scientist and mathematician, “A data scientist is that one of a kind mix of abilities that can both open the experiences of data and recount an awesome story by means of the data.” Yet the hype, energy and passion around data science has become so intense that it has pressurized working individuals in the field of data to hurry themselves into learning all sorts of skills associated with data

such as computer vision, machine learning, text mining, etc.

Observation shows that over three decades, we have seen moves and patterns, an evolution of sorts, in the deep understanding of this profession with respect to its improvement and application. Let us go through the associated resolutions that display the advancement in the field of data science over recent decades.

Data science is more applied today than any time in the past known to humanity

We can see that data science is very relevant today to businesses today than ever given the large sets of data that giants like Facebook, Google, etc. deal with on a day to day basis in the present era. Therefore, we cannot build solutions that are just to be used for demonstration purposes. If a model-based solution goes unused today, it is safe to conclude that it will not be of any use at all. Models that are built today are to be built with the motive of their use in mind and not just for research and development purposes.

Challenges faced in dealing with noisy Datasets

The aim of organizations and enterprises today is using big data for analytics development to reach solutions that will end up satisfying the goals of a client. This being said, the general purpose and substance of the work that is done by data scientists still remains equivocal. For example, research shows that professionals are sought who can work with datasets that are noisy, expansive and heterogeneous in nature. Most new entrants in the field of data science have zero knowledge with respect to the latest and cutting-edge strategies in the field of data science. They therefore, need to look beyond their current skills and aim at learning the disciplines and latest methodologies in the field of data science.

Applied Science knowledge makes you a winner

Understanding what happens inside of a black box does not really concern you unless you have made the box yourself. The number of data scientists with deep knowledge of statistics who work in labs integrating the secret elements to convert them into working tools is very less. This is surprising to some extent for data scientists who have been experts over a long period of time with respect to their understanding and thorough statistical foundation. This will help to scale endeavors for modeling with the information volume we have at hand, questions related to business, and complexities that should be addressed.

The Data-Poor to Data-Rich Transition

A wide experience and deep foundation in the field of pure sciences and data science is both required as organizations are progressing from the data-poor to data-rich enterprises. The supply gap from institutes to the industry will diminish at an exponential rate when institutes change curriculum at a hurried rate to suit the current needs and requirements of the industry. Individuals who are in their 20s, 30s or even 40s who are hoping to turn to data science as their profession, need to expand essentially on important, applied learning and get hands on experience and understanding of the field. You cannot become a data analyst with just one skill or an accreditation in one online course. You need to have a deep understanding of a statistics program that is applied and solid in its sense. The most troublesome problems of data science can be tackled if you have hands-on experience.

Data Science: Art and Science

With data science and machine learning, what you are aiming to learn is the relationship and significance between the human-machine mix. This helps you further understand the correlation between the two and how the decision-making ideas from both human and a machine have advanced so much more

today into our comprehensive field.

Interconnection between Statistics and Data Science

The job of a data scientist will evolve as the field expands too. One of the many definitions of a data scientist states that data scientists are experts with respect to statistics. Although if we look at the current scenario, it may not be the case as the current profile of data scientists has derived itself from the field of engineering. Many people still believe that statistics is far more superior than data science. The current problem is that data is expanding at an exponential rate and both experts from the field of statistics and data science must work together to contain this data in a structured format so that it does not blow out of proportion. In today's world, individuals from completely different backgrounds such as economics also claim to be data scientists. Truth be told, further research shows that there have been instances where it has been established that a huge chunk of the tasks performed by data scientists can be completely automated, robotized and tuned. The tasks revolving around model approval and highlight building, machine learning and comprehension of area, will be the ones that will take center stage in the future.

The current data analytics which works with spreadsheets will eventually move to programming languages such as R and Python and the spotlight will therefore move to code that is parallel and dispersed.

Five Important Considerations in Data Science

Data science is growing very prominently as a tool of strategic business. It has been found out in a study by Deloitte that many organizations are looking to triple their team of data scientists within the next 24 months. With the current implementations of GDPR and privacy being under scrutiny more than ever, data scientists have made it important to model and process data in

a responsible manner. The following five considerations are what data scientists are looking at in the months to come.

- Explainability and Transparency
- Version Control
- Data as the new IP
- Data Bias
- Data Aggregation

Explainability and transparency

May 2016 saw the introduction of General Data Protection Regulation (GDPR), which paved a change in the manner global organizations would collect, manage and process data of people belonging to the European Union. This resulted in an impact in the field of data science as it made it important to think about what kind of data can be used by data science for its modeling purposes and how transparent would the models need to be. As per GDPR compliance an organization should be able to justify how they arrived at a decision that was based on data models. This implies that any organization must secure all the data that they have on a customer and should have sufficient consent from the customer if they want to use that data. It is also expected that in the coming years, regulations around ePrivacy could get a lot harsher which will have an impact on how data can be used. Data architecture will be the next real challenge for data scientists such that it stays in compliance with the regulations made by the law makers.

Version Control

Version control for data is closely associated with GDPR and ePrivacy. Changes being made to software and data by you or other people working on the project is very critical to the project. Why is this important? Because as a

data scientist, when you are explaining the outcome that led to a conclusion based on a data model in a given point in time, you may sometimes need to refer to an earlier iteration of the data. This is really important if you are building models that goes through build change frequently or partially until it reaches the latest build, it is important to store both historic and current builds of the data in the event of an audit.

This holds true in the case where you are running frequent iterations on development of models. Model development is a process that goes through iterations, wherein new packages and techniques are being made available with each iteration. Business should be attentive to their complete suite of models and not just the new models. Versioning should be given importance and implemented so as to be in compliance at all given times. Whether you are a person who makes changes and maintains them manually, or you use version control software like Git, or you are outsourcing version control, you need to ensure that version control is your priority as a data scientist. Failing to do so will put you and your work at risk and can result in the wrath of an Information Commissioner who may even fine you heavily.

Data as the new IP

There is a theory that data is becoming the new IP because along with the code in a software, data is as important now while creating models that are proprietary. The standard of using open source software is growing and computer resources are therefore becoming more affordable. This means that many more enterprises can now build software without a very high budget. The availability of quality and volume in training data is what differentiates models. This holds true for both industries which are just adapting to the new market and are generally slow and static where the data is sparse, and fast-moving industries where data is retrained frequently. If you look at data giants like Google and Amazon, you will understand that training data is

quickly becoming an intellectual property and something that gives one company a competitive advantage over another.

Data bias

Model retraining using automation is all well and good. There is a problem, however, which is that of human bias, a problem that is supposed to be eliminated using algorithms and machine learning. Human bias can be passed to a machine when a machine is being trained in the event that the data being fed to the machine contains traces of a bias. For example, consider a finance industry. If the data being fed is biased, the results may end up offending the fair lending act known as the Equal Credit Opportunity Act. As we have learned from the GDPR act, a customer has the right to know about how a decision was reached; if a loan was rejected, and if this decision was reached due to a bias of data, the case could become difficult to be justified to the customer. We have seen a number of data sets where speech recognition models could not recognize regional accents and image recognition models returned results that were racist in nature, all because the data used to train the models was skewed and biased in nature.

Data Aggregation

GDPR states that anonymity should be ensured by aggregating customer data to a group size that is specific in nature. This does feel like something that would put restrictions on maintaining data but we could also look at it as an opportunity to put more creativity into the thought process that goes into building models and how they would be of benefit to the consumer. Innovation in techniques in clustering and feature generation of data would mean that we will be able to understand and recognize patterns in data and information that were not seen previously. Instead of just trying to comply with GDPR, we could use this as an opportunity to create new models and techniques that will be more customer centric.

Data science has reached a state of its development cycle that is very interesting. There is something new happening every day and newer possibilities are being introduced that can be afforded by the discipline. We should also focus on how we can appreciate privacy of data and that it is the responsibility of data scientists to train machines to respect the data of consumers whose data is being used.

Ten Platforms to be Used in the Field of Data Science

The reaction between the huge volume of data that is available to organizations and how they can put it to use in their decision making such that the organization benefits from this data is what resulted in the need for data science in today's world. The absence of proper tools is all that stands as an obstacle to determine the value that all this information and data has in determining the economic and social value of this data for an organization. Data science came into existence to fill this void of tools to analyze and use this huge set of data. For a business to grow at a constant and good rate, for it to develop, inputs are required which will allow the business to manufacture and produce a product that is required by the consumer. Data science teams come into the picture to develop these specific needs of a growing business. When the general population gives intimate feedback to the models built by a data science team, you can say that their purpose is achieved. If the population were not to work with the models that a data science team creates, the business may not be able to tackle the issue of growth that the special data science team is battling with.

There are a few platforms available that serve the needs of developing models in data science and allow integration of coding languages to serve the same purpose. Models developed for data science are usually unpredictable and require proper coding knowledge and hardware that support it. To use the

data models in data science, data scientists usually deploy multiple machines that are powerful enough and the data processing is simultaneously distributed between all these machines. The platforms do not support programming languages such that one can code in the platform but allow the models to be passed as inputs in python, R, or SAS which can execute the data model code. They, therefore work as a system together with the data models by creating a group in data science. Let us have a look at all the platforms that are available to the field of data science that support the use of analytics code and are universally accepted across the world.

Matlab

When it comes to analytics of entities such as cloud processing, machine learning, neural systems, image processing and so on, MATLAB is the go-to software for many data scientists which is a platform that is very simple and easy to understand and get a grasp on. Huge amounts of data coming from multiple sources can be analyzed with the help of MATLAB. The versatile nature of MATLAB gives it a range from telematics, sensor analytics, all the way to predictive analysis. With the help of MATLAB, data from various sources such as web content, video, images, sound, record frameworks, IoT gadgets, etc. can all be analyzed. MATLAB offers a 1-month free trial and provides annual licenses beginning from USD 820 per year.

TIBCO Statistica

Multiple enterprises deploy TIBO Statistica to understand and solve their numerous issues that are unpredictable in nature. The platform allows users to assemble different models built by them allowing for refreshed learning, analytical procedures, artificial intelligence, and so on. With the help of TIBCO Statistica, one can create complex level of algorithms such as clustering, neural systems, machine learning, all which are accessible via a few nodes.

Alteryx Analytics

A California based software company is the creator of Alteryx Analytics. Business intelligence and predictive analytics products are the primary offerings of this software company that are used for processes related to data science and analytics. The annual membership starts from USD 3995.00 per year and their cloud-based software suite starts at a pricing of USD 1950.00 per year. Data giants like Amazon Web Services, Microsoft, Tableau and Qlik are partners with Asteryx Analytics.

RapidMiner Studio

RapidMiner Studio is a software that is considered to be a visual workflow designer. A tool that helps with processes such as data preparation, machine learning, text mining and predictive analytics, it was specifically developed to help make the lives of data scientists easy. Using the RapidMiner Turbo prep, data scientists can do pivots, take charge of transforming data as well as blend data collected from various sources. All these transactions surprisingly can be processes using a minimal number of clicks.

Databricks Unified Analytics Platform

The creators of Apache Spark created the Databricks Unified Analytics Platform. It provides shared notepads and an environment where users can coordinate to tackle and work with a majority of tasks that fall under analytical procedures. Data scientists can create applications working on artificial intelligence and also create new models consistently. The software is available as a trial for a 14-day period.

Anaconda

With a total number of over seven million users all over the world, Anaconda is a software that is free and open source. Anaconda Distribution and Anaconda Enterprise are the most popular products available from this open

source package. The Anaconda distribution empowers data scientists with a platform and environment that supports around 2000 data bundles in the programming suite of Python and R language for Data Science.

H2O

Used by industries such as finance, healthcare, retail, manufacturing, telco, etc. H2O brags of a user base of 155,000 users in over 14000 organizations worldwide. Driverless AI, which is one of the tools offered by H2O, made it to the winner's list of the 2018 InfoWorld Technology Awards. Multi million-dollar organizations such as PayPal, Dun and Bradstreet, Cisco, and a few more businesses working in assembly use H2O packages very prominently.

KNIME Analytics Platform

The KNIME Analytics Platform is a software that is open source again. Machine learning algorithms and advanced predictive algorithms that use end-to-end workflow in data science are powered by the KNIME Analytics Platform. This software makes it convenient to retrieve data from sources such as Google, Azure, Twitter and Amazon Web Services S3 buckets.

R-Studio

R programming language clients utilize the R-Studio tool as an Integrated Development Environment. The R-Studio platform is very intelligent and furthermore contains bundles that are built-in, which can be used for graphics and computing data that is statistical in nature. The R-Studio platform is supported by all major operating systems such as Windows, Linux and MAC.

Cloudera Data Science Workbench

Among all the platforms that are available to data scientists, software engineers and programming specialists in the world today, Cloudera Data Science Workbench is the most loved platform by all. The tool contains the

latest and most updated libraries scripted in languages such as Python, Scala and R, which can be utilized by end users and data scientists. Data scientists and users have the liberty to develop and create their machine learning models with just a few clicks and hauls which is very comfortable and convenient as compared to all other available platforms.

Chapter Two

Types of Data Analytics

The amount of data that is being collected is increasing every second in the world that we are a part of. It becomes really important to have tools that will help us with the amount of data that is being generated. If data is in the raw format, it is unstructured and often not very useful to anyone. There is a lot of significant information that can be derived by structuring raw data that is where data analysts come into the picture. This is also where different types of data analytics come into the picture. Businesses can drive new initiatives when they have insights available that are driven by data.

The required analysis and the workflow give rise to the 4 most important types of data analytics. They are as follows.

- Descriptive analytics
- Prescriptive analytics
- Diagnostic analytics
- Predictive analytics

Let us try to understand each of these one by one and when exactly these are employed.

Descriptive Analytics

Simply put and understood by the name itself, descriptive analytics is a process where raw data extracted from various sources is converted into a

summarized form which can be understood by humans easily. The process results into describing an event from the past in great detail. Descriptive analytics can help derive patterns from past events and also help draw interpretations from those events eventually helping an organization to frame and create better strategies for the future. It is the most commonly employed analytics across most organizations. Measures and key metrics can be revealed with the help of descriptive analytics in almost any kind of a business.

Prescriptive Analytics

The process of breaking down data step by step in a given situation is what is known as prescriptive analytics. For example, consider that you have booked a cab on Uber. The Uber driver is on his way to pick you up but the regular route has a lot of traffic on it. He then gets an alternate route shown to him on Google Maps. This is a part of prescriptive analytics. Google Maps analyzed the current situation and suggested an alternate route to the Uber driver so that he reaches you for a pick up as soon as possible and time is not wasted. This leads to a better customer experience as well.

Diagnostic Analytics

Diagnostic analytics is known as the successor to descriptive analytics. With the help of diagnostic analytics, data scientists are able to dig deeper into a problem and eventually reach the source of that problem. The tools used for descriptive analytics and diagnostics analytics usually go hand in hand in any business environment.

Predictive Analytics

It is very important for a business to have foresight and vision if it wants to succeed. Predictive analytics helps businesses to forecast patterns and trends

by analyzing present day events. From predicting the probability of events that might take place in the future or even trying to estimate the exact time that the event will take place, it can all be forecasted using predictive analytics. Predictive analytics makes use of variables that are co-dependent to create a pattern and understand the ongoing trend. For example, if you look at the healthcare domain, based on an individual's current lifestyle which consists of his/her eating habits, exercise, travel time, etc. you can predict the kind of illnesses they are likely to contract in the future. Therefore, it can be said that predictive analytic models are the most important as they can be employed across all fields of life.

Data Science Techniques that an Aspiring Data Scientist Should Know

There are many types of analysis that are available to any business, which can be used to extract and retrieve data. The result or the outcome of every project that is based on data science will be different and will be varied. What kind of data science technique is to be used depends on what kind of business you are applying it to and what kind of business problem you are trying to solve? There are various data science techniques that could be employed for a business and therefore, the outcome of each of these techniques could be different resulting into different insights for the given business. This could be confusing but what one needs to understand as a data scientist is that they need to observe information that is relevant to the business which can be figured out easily by recognizing patterns in datasets that are huge.

Let's go through the most common techniques that are practiced in data science for businesses today.

Anomaly Detection

When you go through a dataset that is exhibiting an expected pattern, but

then all of a sudden there is some part in it that doesn't fit the expected pattern, it is termed as an anomaly and the process to find this anomaly is called anomaly detection. Anomalies are also known by other terms such as outliers, exceptions, contaminants, or surprises and their presence often offers valuable insight to the data. Outliers are odd objects that deviate from the standard of a given set of data or deviate from the general average pattern of a dataset. With respect to numbers, an outlier suggests that it is different from all the other data in the dataset, which leads to an understanding that there is something wrong or incorrect about it and requires more analysis.

Anomaly detection is of great interest to data analysts and data scientists as it helps them to understand if there is any kind of fraud or risk involved in a process, which also helps them decide if there is any kind of advanced analysis that would be required on the available data. Thus, the process of anomaly detection helps a business identify if there is any flaw in their process, fraud or areas of business where the existing strategies are failing.

As a data scientist, it is important to accept that small set of anomalies are possible when you are dealing with huge datasets. Anomalies show deviation from standard data but it can also be caused by something that is very random or it may also end up being something that is very interesting statistically. More analysis is needed when such situations arise.

Clustering Analysis

The process of identifying data sets that exhibit attributes that are similar in nature and understanding their similarities and differences is known as clustering analysis. Clusters display specific traits that have common attributes, which if materialized on, can help optimize algorithms that can result into better targeting. For example, consider clusters of data which show the purchasing behavior of customers; this information can be used to target a

specific set of customers with products that could fall in their purchasing power and lead to better rate of conversion.

One of the many outcomes of clustering analysis is customer persona development. This basically refers to fictional characters created to represent types of customers within a demographic region. A particular customer persona defines various attributes of a customer such as their purchasing power, their salary range, their regular purchases, etc. which helps all customers who exhibit the same attributes to be clubbed together and eventually helps a business to target them with the right products. We have learnt about software platform earlier that can be used by the given business to integrate with their cluster analysis.

Association Analysis

If there is a large-scale database at hand, a business can identify and understand the relevant association of various sets of data and its variables with the help of association analysis. Using the technique of association analysis, data scientists can find valuable information in a dataset that is often covert in nature. This will help to detect hidden variables inside a dataset and also let us know if there are co-occurrences happening for variables in a dataset that exists at frequencies that are different.

The technique of association of analysis is helpful to find patterns inside datasets from a point of sales view, and therefore, is used extensively by retail stores. Using this technique, retail stores can recommend new products to customers based on their history of purchase of previous products or the kind of products a customer usually bundles together on a monthly basis. If used efficiently, association analysis can help a business grow and multiply its conversion rates.

Let's look at an example. Using data mining techniques in 2005, Walmart

studied historic data of customers buying products from their store and learnt that every time a hurricane was approaching, the sales of strawberry pops would increase seven times than regular sales. To capitalize on this, every time a hurricane was expected to strike a particular area, Walmart strategically placed strawberry pops at the checkout counter to increase their sales even further.

Regression Analysis

If you want to learn about the dependency between attributes of a dataset, regression analysis is what will help you achieve it. It is assumed that one attribute has an effect that is single-way in nature on the response of another attribute.

Attributes that are independent could be affected by each other's presence. This does not mean that the dependency is mutual between them. Regression analysis helps a business detect if one variable in a dataset is dependent on another variable but not vice versa.

Regression analysis can also be used by a business to understand client satisfaction levels and if customer loyalty can be affected by an attribute and if it may end up affecting the service levels as well, for instance, the current weather.

Another great application of regression analysis is dating websites and dating apps which use regression analysis to improve the services offered to the users. Regression analysis checks the attributes of users of a dating application and try to match two users based on those attributes to create a match that is bet for the users who are participating.

Data science helps achieve businesses focus on information that is important and relevant from the point of view of growth for the business. Therefore,

eventually data science helps establish business models that can help a business that can predict the behavior of its customers and helps the business get better conversion rates.

Gathering more information would help to build better models which can be used effectively by applying processes of data science to the information, which will increase the value of the business gradually.

Classification Analysis

The approach to gather information that is relevant and crucial about data in a systematic manner is known as classification analysis. When you have a lot of data, classification analysis techniques help a business identify which data can be used for further and deeper analysis. Given that classification of data is usually a prerequisite before you start clustering data, classification analysis goes hand in hand with cluster analysis. The biggest users of classification analysis are Email providers. A user receives a lot of emails on a daily basis, some of which is useful and the rest is spam. Email providers have algorithms in place that help classify email as genuine or spam. This is done based on the metadata of the Email that is contained in the headers of the Email such as from address, reply-to address, etc. or the content that is in the actual body of the Email message.

Chapter Three

Data Types and Variables

The previous book shed some light on what Python is and how you can use Python for analytics. You learnt more about the different techniques of data analytics, like data visualization. The next few chapters will help you understand the basics of Python so you can work on developing your very own models.

This chapter will introduce you to the different types of variables that you can use when writing a program in Python. You will also learn how these variables can be used to convert your designs into working codes using Python. This is when you begin real programming. Over the course of this chapter, we will work on two programs – one where we will learn to format and manipulate text strings and another to perform a simple mathematical calculation.

The programs mentioned above can be written easily using different variables. When you use variables, you can specify a function, method of calculation that must be used to obtain a solution without the knowledge of the type of value that the variable must refer to in advance. Every piece of information that must be put into a system needs to be converted into a variable before it can be used in a function. The output of the program is received only when the contents of these variables are put through all the functions written in the program.

Choosing the Right Identifier

Every section of your code is identified using an identifier. The compiler or editor in Python will consider any word that is delimited by quotation marks, has not been commented out, or has escaped in a way by which it cannot be considered or marked as an identifier. Since an identifier is only a name label, it could refer to just about anything, therefore, it makes sense to have names that can be understood by the language. You have to ensure that you do not choose a name that has already been used in the current code to identify any new variable.

If you choose a name that is the same as the older name, the original variable becomes inaccessible. This can be a bad idea if the name chosen is an essential part of your program. Luckily, when you write a code in Python, it does not let you name a variable with a name used already. The next section of this chapter lists out the important words, also called keywords, in Python, which will help you avoid the problem.

Python Keywords

The following words, also called keywords, are the base of the Python language. You cannot use these words to name an identifier or a variable in your program since these words are considered the core words of the language. These words cannot be misspelt and must be written in the same way for the interpreter to understand what you want the system to do. Some of the words listed below have a different meaning, which will be covered in later chapters.

- False
- None
- assert
- True

- as
- break
- continue
- def
- import
- in
- is
- and
- class
- del
- for
- from
- global
- raise
- return
- else
- elif
- not
- or
- pass
- except
- try
- while

- with
- finally
- if
- lambda
- nonlocal
- yield

Understanding the Naming Convention

Let us talk about the words that you can use and those you cannot use. Every variable name must always begin with an underscore or a letter. Some variables can contain numbers, but they cannot start with one. If the interpreter comes across a set of variables that begin with a number instead of quotation marks or a letter, it will only consider that variable as a number. You should never use anything other than an underscore, number or letter to identify a variable in your code. You must also remember that Python is a case-sensitive language, therefore false and False are two different entities. The same can be said for vvariable, Vvariable and VVariable. As a beginner, you must make a note of all the variables you use in your code. This will also help you find something easier in your code.

Creating and Assigning Values to Variables

Every variable is created in two stages – the first is to initialize the variable and the second is to assign a value to that variable. In the first step, you must create a variable and name it appropriately to stick a label on it and in the second step, you must put a value in the variable. These steps are performed using a single command in Python using the equal to sign. When you must

assign a value, you should write the following code:

```
Variable = value
```

Every section of the code that performs some function, like an assignment, is called a statement. The part of the code that can be evaluated to obtain a value is called an expression. Let us take a look at the following example:

```
Length = 14
```

```
Breadth = 10
```

```
Height = 10
```

```
Area_Triangle = Length * Breadth * Height
```

Any variable can be assigned a value or an expression, like the assignment made to `Area_Triangle` in the example above.

Every statement must be written in a separate line. If you write the statements down the way you would write down a shopping list, you are going the right way. Every recipe begins in the same way with a list of ingredients and the proportions along with the equipment that you would need to use to complete your dish. The same happens when you write a Python code – you first define the variables you want to use and then create functions and methods to use on those variables.

Recognizing Different Types of Variables

The interpreter in python recognizes different types of variables – sequences or lists, numbers, words or string literals, Booleans and mappings. These variables are often used in Python programs. A variable `None` has a type of its own called `NoneType`. Before we look at how words and numbers can be used in Python, we must first look at the dynamic typing features in Python.

Working with Dynamic Typing

When you assign a value to a variable, the interpreter will choose to decide the type of value the variable is which is called dynamic typing. This type of typing does not have anything to do with how fast you can type on the keyboard. Unlike the other languages, Python does not require that the user declare the types of the variables being used in the program. This can be considered both a blessing and a curse. The advantage is that you do not have to worry about the variable type when you write the code, and you only need to worry about the way the variable behaves.

Dynamic Typing in Python makes it easier for the interpreter to handle user input that is unpredictable. The interpreter for Python accepts different forms of user input to which it assigns a dynamic type which means that a single statement can be used to deal with numbers, words, or other data types, and the user does not have to always know what data type the variable must be. Since you do not have to declare a variable in Python before you use it, you may be tempted to introduce a new variable somewhere in the code. You must remember that you will never receive an error from Python until you use a variable that does not have a value assigned to it. That being said, it is very easy for a programmer to lose track of the variables being used and where the variables are set up in the script. You can choose to perform two different functions if you want to avoid these issues. You will need to use these techniques especially when you begin to create numerous variables in your script. The first option is to bunch all the variables at the start of the code. You can also assign some default values to these variables. The next option is to always maintain a record of the different variables you are creating and maintain a data table in your comments or in the document that you write for each program.

Python will always need to keep track of the variables that you include in the script. The first is that the machine will need to save some memory to store the value in the variable. It is important to remember that every data type takes up different amounts of space. The second is that when you keep track of the different variables, you can avoid making errors in your code. Python will flag an error called `TypeError` if you perform an operation on a variable that does not support that operation. This may seem irritating at first, but this is one of the most useful features of the language. Let us look at the example below:

```
>>> b = 3
```

```
>>> c = 'word'
```

```
>>> trace = False
```

```
>>>
```

```
b + c
```

```
Traceback (most recent call last):
```

```
File "", line 1, in <module>
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
>>> c - trace
```

```
Traceback (most recent call last):
```

```
File "", line 1, in <module>
```

```
TypeError: unsupported operand type(s) for -: 'str' and 'bool'
```

The program above tries to perform operation on data types that are incompatible. You cannot remove the boolean answer yes/no or add any

number to a text variable. You must always convert the data type before you try to process it. It is important to convert the data type to another type that is compatible for the operation. You can combine words or numbers, like you would normally, but you cannot perform an arithmetic operation on a text data type. Python will throw an alert, called the `TypeError`, which will help you trace the error in the script that you have written. The error will tell you where the error is in the code, and will point you to the exact line. You can then work on giving the code clear instructions so you can get the required value from the equation.

A data type is used to help you represent any information that can be found in the real world. What I mean by the real world is the world that exists outside the computer. In the previous examples, we used the data types `int` and `str`. You will soon learn that these data types can only be used to indicate the simplest information.

You can combine these data types to develop some complex data types. We will cover this a little later in the book. You will first need to learn more about the building blocks that you can use to define the data and also identify the set of actions that you would like to perform to manipulate the values held by these variables.

The None Variable

A predefined variable called `None` is a special value in Python. This variable has a type of its own and is useful when you need to create a variable but not define or specify a value to that variable. When you assign values such as `""` and `0`, the interpreter will define the variable as the `str` or `int` variable.

```
Information = None
```

A variable can be assigned the value `None` using the statement above. The

next few examples will use real-world information that will be modeled into a virtual form using some fantasy characters. This example uses some statistics to represent some attributes of the characters to provide data for the combat system. You can use this example to automate your database and your accounts. So, let us take a look at some of the characters in the example.

In the program, `hello_world.py`, you saw how you can get a basic output using the `print ()` function. This function can be used to print out the value of the variable and a literal string of characters. Often, each print statement must start off on a new line, but several values can be printed on a single line by using a comma to separate them; `print ()` can then be used to concatenate all the variables into a single line only separated by spaces.

```
>>> Race = "Goblin"

>>> Gender = "Female"

>>> print (Gender, Race)

Female Goblin
```

Different segments of information can be combined into a single line using multiple methods. Some of these methods are more efficient when compared to others. Adjacent strings that are not separated will be concatenated automatically, but this is not a function that works for most variables.

```
>>> print ("Male" "Elf")
```

The expression above will give you the following output – “MaleElf”

When you enter the following code,

```
>>> print (“Male” Race)
```

You will receive the following error:

```
File "<stdin>", line 1
```

```
print ("Male" Race)
```

```
^
```

```
SyntaxError: invalid syntax
```

This approach cannot be used since you cannot write a string function as a variable and a string together since this is just a way of writing a single line string.

Using Quotes

In Python, a character is used to describe a single number, punctuation mark, or a single letter. A string of characters used to display some text are called strings or string literals. If you need to tell the interpreter that you want a block of text to be displayed as text, you must enclose those characters in quotation marks. This syntax can take multiple forms –

‘A text string enclosed in single quotation marks.’

“A text string enclosed in double quotation marks.”

““A text sting enclosed in triple quotation marks.””

If text is enclosed in quotes, it is considered the type str (string).

Nesting Quotes

There are times when you may want to include literal quotation marks in your code. Python allows you to include a set of quotation marks inside another set of quotation marks, if you use a different type of quotation mark.

```
>>>text= “You are learning ‘how to’ use nested quotes in Python”
```

In the example above, the interpreter will assume that it has reached the end

of the string when it reaches end of the text at second set of double quotes in the string above. Therefore, the substring 'how to' is considered a part of the main string including the quotes. In this way, you can have at least one level of nested quotes. The easiest way to learn how to work with nested quotes is by experimenting with different types of strings.

```
>>> boilerplate = """
#====(")====#====(*)====#====(")====#

Egregious Response Generator

Version '0.1'

"FiliBuster" technologies inc.

#====(")====#====(*)====#====(")====#

"""

>>> print(boilerplate) #==== (“) ====#==== (*) ====#==== (“) ====#

Egregious Response Generator

Version '0.1'

"FiliBuster" technologies inc.

#====(")====#====(*)====#====(")====#
```

This is a useful trick to use if you want to format a whole block of text or a whole page.

How to use Whitespace Characters

Whitespace characters are can often be specified if the sequence of characters begin with a backslash. '\n' produces a linefeed character that is different

from the ‘\r’ character. In the output window, the former would shift the output to a new line, while the latter would shift the output to a new paragraph. You must understand the difference between how different operating systems use to translate the text.

The usage and meaning of some of the sequences are lost on most occasions. You may often want to use \n to shift to a new line. Another sequence that is useful is \t, which can be used for the indentation of text by producing a tab character. Most of the other whitespace characters are used only in specialized situations.

Sequence	Meaning
\n	New line
\r	Carriage Return
\t	Tab
\v	Vertical Tab
\e	Escape Character
\f	Formfeed
\b	Backspace
\a	Bell

You can use the example below to format the output for your screen:

```
>>> print (“Characters\n\nDescription\nChoose your character\n \n\n\tDobby\n\tElf\n\tMale\nDon’t forget to escape ‘\\’.”)
```

)

Characters

Description

Choose your character

Dobby

Elf

Male

Don't forget to escape '\

You must remember that strings are immutable which means that they cannot be changed. It is possible to use simple functions to create new strings with different values.

How to Create a Text Application

All the information mentioned in this chapter can be used to write the code for our role-playing game. Strings are often simple to use since you must only ensure that you enclose the strings in matching quotes. The script to design the character-description is simple.

Prompt the user for some user-defined information

Output the character description

You may want to include the following information for the character:

- Name
- Gender
- Race

- Description of the character

For this information, you can create the following variables – Name, Gender, Race and Description. These values can be printed using the following code:

```
"""
```

```
    chargen.py
```

Problem: Generate a description for a fantasy role-playing character.

Target Users: Me and my friends

Target System: GNU/Linux

Interface: Command-line

Functional Requirements: Print out the character sheet

User must be able to input the character's name, description, gender and race

```
    Testing: Simple run test
```

```
    Maintainer: maintainer@website.com
```

```
"""
```

```
__version__ = 0.1
```

```
Name = ""
```

```
Desc = ""
```

```
Gender = ""
```

```
Race = ""
```

```
# Prompt user for user-defined information
```

```
Name = input('What is your Name? ')
```

```

Desc = input ('Describe yourself: ')

Gender = input ('What Gender are you? (male / female / unsure): ')

Race = input ('What fantasy Race are you? - (Pixie / Vulcan /
Gelfling / Troll/ Elf/ Goblin): ')

# Output the character sheet

character_line = "<~==|#|!!+*@\/*+~==|#|+~>"

print ("\n", character_line)

print ("\t", Name)

print ("\t", Race, Gender)

print ("\t", Desc)

print (fancy_line, "\n")

```

The program above is a smarter version of the hello_world program written above. In this program, there is a new line added `_version_ = 0.1` at the start of the program. This is a predefined variable that has a special meaning in Python's documentation. This is the number we will continue to use to record the above example. As we go along, we will continue to increment this number when we make any changes or refine the program. Now, we will need to obtain some numerical information about the characters that will interact in the game.

Working with Numbers

You can assign any value to a variable. For example:

```
Muscle = 8
```

```
Brains = 13
```

As mentioned earlier, if there is any variable that has a mix of numbers and characters, the interpreter will assume that the variable beginning with a number will be considered a number and not a character. If you want the interpreter to look at the variable as a character, you should start the number with a quotation mark. It is for this reason that you should avoid beginning a variable with a number. There are a few things you need to consider when you work with numbers in Python.

Computers only count to one

All the information in the computer can only be stored in zeros and ones. Every computer stores and processes any volume of data using tiny switches that can either be on (1) or off (0).

Using Boolean

As mentioned earlier, a computer can only register two values – True (value = 1) and False (value = 0). These values are known as Boolean operators and can be manipulated using operators like OR, NOT and AND. These operators are explained in further detail in the following chapter. Boolean values can be assigned as follows:

Mirage = False

Intelligence = True

Using Whole Numbers

Whole numbers, also called integers, do not have decimal points and can be zero, positive and negative. These numbers are used to refer to different things like the recipe example mentioned above.

Performing Basic Mathematical Operations

Now that you know how to store data in a variable, let us take a look at how

to manipulate that data. Basic mathematical operations can be performed using operators like +, - and *. These operators create an expression that must be evaluated before you can obtain a value. The following statements can be used to perform these operations.

```
>>>muscle = 2 + 3
```

```
>>>brains = 7+4
```

```
>>> speed = 5 * 6
```

```
>>> weirdness = muscle * brains + speed
```

```
>>> weirdness
```

All these operations work using the BODMAS mathematical algorithm.

Working with Floats and Fractions

Most fractions are often expressed using the float type where decimal points can be used. These numbers, like integers, can be both positive and negative. You do not have to assign a variable to the data type float. Python automatically converts a variable into the float type if it is assigned a decimal number.

```
Muscle = 2.8
```

```
Brains = 4.6
```

```
Speed = 6.8
```

Even if the number before and after the decimal point is 0, it is still considered a fraction. This data type can be manipulated using the same mathematical operations mentioned above.

Converting Data Types

There are different built-in functions that are used in Python to convert a value from one data type to another. The data types often used are:

- `int (x)` – used to convert any number into an integer
- `float (x)` – used to convert a number to a float data type
- `str (object)` – convert any type into a string that can be used to print

```
>>> float (23)
```

```
23.0
```

```
>>> int (23.5)
```

```
23
```

```
>>> float (int (23.5))
```

```
23
```

Chapter Four

Conditional Statements

In the last few chapters, you have learned how to use Python to manipulate strings and to make simple calculations. More importantly, you have learnt how to design your software. Now, it is time to learn how to refine your code. Therefore, pull out your old scripts and find an effective way to obtain your output.

How to Compare Variables

To generate more accurate answers, you must know how to compare the values and specify what the interpreter must do based on the obtained result. Python allows you to use conditional statements to allow you to make these decisions. A conditional statement can transform the code or script from just being a list of instructions to a code that can be used by the user to make their own decisions. It would be useful to tell the interpreter to perform a different action as per the decisions made by the user. You can write a pseudocode like:

if a certain condition is true:

 then the following actions must be performed;

if another condition is true:

 then these actions must be performed.

Each pair in the example above is a conditional statement, but before we learn more about these statements, let us take a look at how to specify these

conditions. Different values can be compared using the following operators:

- <: Less than
- >: Greater than
- <=: Less than equal to
- >=: Greater than equal to
- ==: Equal to
- !=: Not equal to

These operators affect data types in different ways and give the user answers in the form of the Boolean operators. The data bits on either side of the operator are called operands and these are the variables that are compared. The comparative operator and the operands together form the conditional expression. It is important to check the conditional statements or expressions you are using since you may obtain an error if you compare incomparable data types. The results obtained by comparing these numbers are self-explanatory.

```
>>> -2 < 5
```

```
True
```

```
>>> 49 > 37
```

```
True
```

```
>>> 7.65 != 6.0
```

```
True
```

```
>>> -5 <= -2
```

```
True
```

```
>>> 7 < -7
```

```
False
```

```
>>> 23.5 > 37.75
```

```
False
```

```
>>> -5 >= 5
```

```
False
```

```
>>> 3.2 != 3.2
```

```
False
```

Variables can also be used in conditional expressions.

```
>>> variable = 3.0
```

```
>>> variable == 3
```

```
True
```

Manipulating Boolean Variables

Before you move onto the different conditional structures used in Python, you must learn how to manipulate the Boolean values True and False. You can use these values to understand the characteristics of any variable. These operators are often used with the terms AND, OR and NOT. The statements below represent some bits of information.

```
>>> a = True
```

```
>>> b = False
```

```
>>> c = True
```

```
>>> d = True
```

```
>>> e = False
```


Let us take a look at how AND, OR and NOT can be used.

```
>>> a or b
```

This operator returns the value True, since for the OR operator either one of the values needs to be true.

```
>>> c and e
```

This operator returns the value False, since for the AND operator both values must be the same.

```
>>> not d
```

This operator returns the value False, since the NOT operator provides the opposite of the value.

Combine Conditional Expressions

Conditional expressions can be combined to produce complex conditions that use the logical operators AND and OR. Let us take a look at the following conditions:

```
(a < 6) AND (b > 7)
```

This statement will only return True if the value of a is less than 6 and the value of b is greater than 7.

The Assignment Operator

Since you are familiar with the assignment operator (=) which you use to put a value into a variable, let us take a look at how you can use this operator to assign values to variables. This assignment operator can be used to unpack sequences.

```
>>> char1, char2, char3 = 'cat'
```

```
>>> char1
```

```
'c'
```

```
>>> char2
```

```
'a'
```

```
>>> char3
```

```
't'
```

The assignment operator can also be used to assign different variables with the same value.

```
a = b = c = 1
```

The assignment operator can also be used along with mathematical operators.

```
counter += 1
```

The statement above is interpreted as `counter = counter + 1`. Other operators also can be used to either increment or decrement the value of the variable.

How to Control the Process

You have the liberty to decide what happens next in the program you have written using a control flow statement. The results of the comparison statements can be used to create conditional statements that allow the interpreter to provide the output that is based on whether the predefined conditions hold true. Conditional statements can be constructed using the keywords `if`, `elif` and `else`. Unlike other languages, Python does not use the keyword `then`. The syntax is very specific therefore you must pay close attention to the layout and punctuation.

```
if condition:
```

```
# Perform some actions
```

```
    print "Condition is True"

    elif condition != True:

# Perform some other actions

        print "Condition is not True"

    else:

# Perform default or fall-through actions

        print "Anomaly: Condition is neither True nor False"
```

In the syntax above, the first line begins with the word `if`, which must be followed by a conditional statement that gives a `True` or `False` output followed by the colon. This colon means yes. The statements that follow must always start on a new line. You can leave as many spaces as you would like in a line of code, but it is important to ensure that the code that is written after the colon follows the same rules of indentation. It is always a good idea to use the right number of spaces across the code since it will help you control the flow of the program throughout your code. The group of statements that are written after the colon constitute a suite.

You can also include some conditional sections to your code using the `elif` keyword. This keyword is the abbreviation of the conditional statement `else if` which cannot be used in Python. You must remember that the statements under the `elif` section are evaluated only if the condition in the previous section fails. We will learn more about this later in the book.

You are also allowed to include a final `else` statement which will then look at any value for which the condition did not hold true. This section does not take any statements or conditions. You can use this to specify the default set of actions that you can perform. In the previous example, there would have

been an error if the conditions in the if and elif statements were not clearly defined.

You can nest statements if you wish to include more possibilities, and you can ignore the usage of the elif statement entirely. You should do this only when you do not want any action to be performed if the condition held true. In simple words, there are times when you want some action to be performed only when the condition holds true, but not when the condition is false.

Make sure that the indentation goes back to the same level once you have written the final statement in your code. This will let the interpreter know that the conditional block of code has ended. The interpreter can only know if a block of code has ended based on the indentation, and it cannot use punctuation marks like other languages to mark the block of code. This makes it important for you to maintain your indentation across the script. The interpreter will throw an error if you have not maintained the indentation across your script.

```
>>> if c:
    print(c)
    c += 1
    indent = "bad"
File "<stdin>", line 4
    indent = "bad"
^
```

IndentationError: unindent does not match any outer indentation level

A conditional statement always gives the user the ability to check or validate

the data that was used as the input. Validation is often performed when the data is first fed into the computer and also when the information is written out on a database record or file.

How to Deal with Logical Errors

You will need to develop formal ways to test your script as the applications you are developing become more complex. You can construct a trace table to make it easier for you to do this. You must trace the values of all the variables and the conditional expressions over the course of the execution of the program.

You should perform a trace with different sets of data to ensure that any alternative is tested across the entire script. The errors in a program will never occur if the values that are being tested will lie within some range. The errors do occur for critical or unusual values. A critical value is any value in the script that will always lie outside of the tolerance of the interpreter or the program. For example, the program may not have the ability to work with a specific number. It is important for us to work these out earlier during the design process to ensure that a program can be tested properly. In the calculation of the area of the triangle, the value that most needs taking into account is that of the breadth, which has been set at 14 cm. Allowing 8 cm means that the maximum breadth of the triangle can only be 8 cm.

Using the conditional code

You can now apply your understanding of different conditional statements to measure the material that you have in your data set. If the breadth of the triangle were too much, it would become a different type of triangle. Therefore, you need to identify the right code which reflects the right conditions. The first step would be to translate your trace values into a pseudocode. The following example is about measuring the length of a

curtain.

```
if curtain width < roll width:  
total_length = curtain width  
else:  
total_length = curtain length  
if (curtain width > roll width) and (curtain length > roll width):  
if extra material < (roll width / 2):  
width +=1  
if extra material > (roll width / 2):  
width +=2
```

Loops

While Statement

```
result = 1  
while result < 1000:  
result *= 2  
print result
```

You can control the number of times that you process a loop by using conditional statements in the loop. The loop will continue to run if the conditional statement written at the start of the loop will hold true at the beginning. In the previous example, the result of our conditional statement is greater than 1000. The loop will continue to process as long as the value of the variable is less than 1000. If the result reaches 1024 (2¹⁰), the loop will stop functioning and will end. The variables used in the condition of the loop are expendable in the sense that they do not have to be used anywhere else in

the code. You can name the integer counter as i or j instead of assigning some names to the variables.

There are two things that you will need to remember when you are constructing any type of code. A variable that is used in a conditional statement should always be initialized before the loop is executed. There should also be a way to update the expressions in the conditional statement. The loop will go around and round forever, which is called an infinite loop.

You can use different types of variables in the conditional expression that you write. Let us look at a problem where you are required to calculate the average of numerous inputs made by the user. The issue with this is that you never know how many numbers can be used as an input in the statement. The only solution here is to use a sentinel value, which will help you control the loop. Instead of using a counter, you can instruct the interpreter to look at the value that has been entered by the user. If the number entered is positive, the loop will continue to process, but the loop will be broken if the value entered is negative.

Let us take a look at the following example:

```
counter = 0
```

```
total = 0
```

```
number = 0
```

```
while number >= 0:
```

```
number = int (input ("Enter a positive number\nor a negative to exit:  
"))
```

```
total += number
```

```
counter += 1
```

```
average = total / counter  
print(average)
```

There are numerous ways in which you can exit a loop cleanly. You can use the continue and break keywords for the same purpose. If you want to exit a loop and stop executing any statements in the body of the loop, you should use the break keyword. If you want to iterate a specific part of the loop, you should use the continue statement. This will help you execute that section of the loop that you want to execute.

There are times when you will need the interpreter to recognize the condition and not perform any other action. You can use the pass keyword in this instance. This keyword will create a null instance that will

There are times when you want to instruct the interpreter to do nothing if a condition holds true. In this instance, you can use the pass keyword. This keyword will create a null statement that will instruct the interpreter to only move to the next instruction in the code.

Nesting Loops

It is easy to nest conditional statements and loops in Python, and you can create an infinite loop, but it is important to remember to keep the number of levels to a minimum. It is easy to get confused about the option that the interpreter is currently using. It also makes it difficult for people to read the code since there will be multiple indentations in your code. It is possible that the nesting also slows down the execution of the program. In simple words, this is a bad way to write your program.

If you write a code that has over three layers of looping, you should definitely redesign the code so you can avoid making too many errors.

For

You should also have a good understanding of the 'for' control flow statement. This statement is written in the same way as the if and the while statements. The syntax is written as the for keyword followed by a suite of instructions that have been indented well. The loop variable element will contain the first element in the sequence during the first iteration of the loop. This variable can now be used by the statements within the suite. During the second iteration, the variable takes the second element, and so on.

If you want to learn more about this statement, you should understand sequences. A simple example of a sequence in Python is a string. A string is a sequence of characters that include punctuation and spaces. Tuples and lists are other types of sequences that can be used in Python. A tuple and list are a sequence of items, and as mentioned earlier a list can be edited once created while a tuple cannot. You can construct them in a for statement in the following manner:

```
# tuple
```

```
sequence1 = (1, 2, 3)
```

```
# list
```

```
sequence2 = [1, 2, 3]
```

Chapter Five

Data Structures

In the last few chapters, you learned how you can work with individual pieces of data to obtain some simple results. Real world data is usually available in groups or lumps, and it is easier to work with groups since it makes it easier for us to eliminate repetitive code. There are numerous data types in Python that will make it easier for you to handle large groups of data.

Programmers often use strings, lists, dictionaries and tuples when they write a script in Python. These data types are known as data structures. Strings are pieces of text that are grouped together, while tuples and lists are groups of individual data items that have been grouped together. A dictionary is a group of pairs that have the highest considerations. The different methods that are used to access the data in these structures is the same. This will be covered in detail in later parts of the chapter.

You can also look at these data types in a different way depending on whether the values that the variable holds can be modified. This is called the mutability of the data type. A string and tuple cannot be modified, but they can be used to create new tuples and strings. A list is mutable which means that you can either remove or add items to it.

Items in Sequences

You can fetch individual items in a sequence using an index. This index will indicate the position of the element. The index is often an integer that is written in square brackets immediately after the name of the variable. So, you

can obtain the variable in a list by specifying the name of the list followed by the index. You can also access a single character in a string.

```
>>> vegetable = 'pumpkin'
```

```
>>> vegetable [0]
```

```
'p'
```

Or an item in a list:

```
>>>vegetable = ['pumpkins', 'potatoes', 'onions', 'eggplant']
```

```
>>>vegetable [1]
```

```
'pumpkins'
```

You will notice that indexing in Python is zero-based. This means that you can only start counting the variables at zero. An index with the number 3 in the square brackets will look at the fourth item in the list since the first item will be indexed as zero. So, you can use any number of integers beginning from zero to index the variables in your data set. A negative index will count the list from the end to the beginning:

```
>>>vegetable [-1]
```

'eggplant'

Slices can be used to grab the different sections in any sequence. This method is used to fetch many items in a sequence. A slice is written using the same notation as an index. The only difference is that the integers are separated by a colon. The first value is the starting point, and this value is included. The second number in the notation is the end point of the slice, and it is exclusive. If you look at s[0:2], the compiler will slice the list from the variable with the index zero and stop exactly before the variable with the index two.

You do not necessarily have to use the third value, and this is an additional

step. This can be negative; therefore, you can retrieve all the other items instead of picking this item from the sequential list. Alternatively, you can retrieve items backward as well. So, `s [i: j: step]` will give you the slice that begins from the variable `i`, but will not include the variable `j`. Here, `s` is the sequence.

If you ignore the initial point, the slice will always start at the beginning of the sequence. If you forget the end, the slice will continue to the end of the original or main sequence.

Slicing and indexing do not change the original sequence. They will develop a new sequence. The actual data items in the sequence will be the same. So, if you want to modify an individual item in the sequence, you will see that the item has changed in the slice as well.

Tuples

Tuples are a group of items or elements that are ordered and immutable. You should think of a tuple as a sealed packet of information.

A tuple is specified as a comma-separated list of values. These values can be enclosed within parentheses if necessary. In some cases, these parentheses are required, so always use them regardless of whether or not you think they are necessary. The values in the tuple do not necessarily have to be of the same data type. Some values can also be other tuples.

Creating a Tuple

Tuples can be created with no items in it using the round brackets `()`.

```
>>>empty_tuple= ()
```

If you do not want more than one item in the tuple, you should enter the first item followed by a comma.

```
>>>one_item = ('blue',)
```

Changing Values in a Tuple

The values in a tuple cannot be changed. These tuples are sealed packets of information that are often used in situations where a set of values need to be passed on from one location to another. If you wish to change the sequence of the data, you should use a list.

List

A list is a comma-separated and ordered list of items that are enclosed within square brackets. The items within the list do not have to be of the same data type. You can also include a list within a list.

A list can be concatenated, indexed and sliced just like any other sequence you can use in Python. You can change some items within a list when compared to a tuple or string. Lists are very flexible when compared to tuples. You can either clear a list or change the list completely by slicing the list and assigning the data to other variables.

Creating a List

It is easy to create a list.

```
>>> shopping_list = ['detergent', 'deodorant', 'shampoo', 'body  
wash']
```

Modifying a List

A new value can be added to a list using the assignment operator.

```
>>> shopping_list [1] = 'candles'
```

```
>>> shopping_list
```

```
['detergent', 'candles', 'deodorant', 'shampoo', 'body wash']
```

Stacks and Queues

You can use lists to store and retrieve data or variables in a specific order since lists are ordered data types. The main models that one can use to do this are by using stacks and queues. A stack uses the last in first out (LIFO) approach. A real-world example of this approach is how the discard pile is used in a card game. You add cards to the top of the pile and remove the card from the top. You can include items into a stack using the `list.append()` function and remove the items from a stack using the `pop()` function. There are no additional index arguments that you will need to include when you use these functions, so the last item in the list is removed.

```
>>> shopping_list.append('brush')
>>> shopping_list.pop()
'candles'
>>> shopping_list
['detergent', 'deodorant', 'shampoo', 'body wash']
```

The second approach is to create the first in first out (FIFO) structure. A queue uses this type of an approach. This method works like a pipe where the first item is pushed out of the pipe before the remaining items. You can use the same functions, `append()` and `pop()`, to either push items into the queue or remove them from the queue. You will, however, need to use the index zero to indicate that the items should be popped from the start of the list.

```
>>> shopping_list.append('brush')
>>> shopping_list.pop(0)
'detergent'
```

```
>>> shopping_list  
['deodorant', 'shampoo', 'body wash', 'brush']
```

Dictionaries

A dictionary is much like an address book. If you know the name of the person you wish to contact, you can obtain the details of that person. The name of the person is the key while the details of the person are the value.

The key that you use in a dictionary should be an immutable data type, that is it can be a number, tuple or string. The value can be anything. A dictionary is a mutable data type, and it is for this reason that you can add, modify or remove any pairs from the dictionary. The keys are mapped to an object and it is for this reason that a dictionary is also known as mappings. This will show you that a dictionary behaves differently when compared to a sequence.

A dictionary can be used anywhere you want to store a value or attribute that will describe an entity or a concept. For instance, you can use a dictionary to count the number of instances of a specific state or object. Since every key has a unique identifier, you cannot have duplicate values for the same key. Therefore, the key can be used to store the items in the input data and the values can store the result of the calculation.

Chapter Six

Working with Strings

Commands used in Python 3 work in the same way as commands in Python 2. There are a few important changes that you need to keep in mind. The most important change is how the string data type can be used. In earlier versions of Python, the string data type was coded as a single sequence of bytes using the ASCII character set. This set was used to represent the text.

To go along with the string type changes, the print statement in Python 2.x has been replaced with the print() function which is a built-in function in version 3.0. this function replaces most of the earlier syntax with keyword arguments. If you want to balance this, you should replace the input() function using the raw_input() function. You should also use the function eval(input()) in the same way you would use the old input() function.

Splitting Strings

Since strings cannot be change, that is they are immutable, you may want to split them into smaller variables or lists to make it easier for you to manipulate the content. It is important to remember that a delimiter is a string of characters or a character that is used to separate a unit of data or words. The list can be split numerous times using the maxsplit() function, and you will end up with maxsplit+1 lists. If you do not specify a separator, the string will only be split using whitespaces.

```
>>>sentence = 'This is a long sentence'
```



```
>>> sentence.rstrip('sentence').split()

['This', 'is', 'a', 'long']
```

You can split a string using the `string.partition(sep)` function that will return the tuple (head, sep, tail). When you use this method, the interpreter identifies the separator within the string and then returns the section before the separator the separator and the part of the string that is separated from the string. If the interpreter cannot find the separator, the method will return two empty strings and the original string.

Concatenation and Joining Strings

You can use the plus operator if you want to combine strings, but this is a very inefficient way of doing it. When you combine the plus operator with different print functions, it will slow the execution of your program. Python is not slow, and it is often better to manipulate the list of words in a statement and then use the function `string.join(sequence)` to return the value of a string, which is a combination of the strings present in a sequence. This method is the exact opposite of the `string.split()` function. The data that you wish to manipulate is present in the sequence of the argument, and the string that you wish to use is the string of characters you want to use to separate the items in the string. The value could either be an empty string or a space.

```
>>> s1="example"

>>> s2 = "text"

>>> s3 = " "

>>> s3.join([s1,s2])

'example text'
```

You must remember that the function `string.join()` always expects a sequence of strings as the argument.

```
>>> s3 = "-"  
  
>>> s3.join('castle')  
  
'c-a-s-t-l-e'
```

You also may have the need to convert different data types into strings by using a sub list.

Editing Strings

You cannot edit a string in a few places alone, but there are some methods that you can use to edit strings. These methods will return new versions of the string.

There are times when you will need to remove the whitespaces at the beginning or the end of the string. This will need to be done if you are trying to work on comparing the user input with any other value that is stored in the system. You can do this by using the `string.strip([chars])` method. The method will return the copy of the string by removing all the characters at the beginning and the end of the sequence. If there are no arguments given to the string, the function `string.strip()` can be used to remove these whitespaces.

```
>>> sentence = 'This is a long sentence'  
  
>>> sentence.strip('A')  
  
' This is long sentence'
```

How to Match Patterns

There are times when you cannot use basic string methods. For instance, you may have to retrieve the values that are present in a regular pattern in a block of text, but you never know what these values will be. This is when you will need to use a regular expression. A regular expression or regex for short is a pattern that can be used to match some text in your code. In the simplest form, a regular expression is a plain string of characters that match itself. A regular expression will use a syntax that has some special characters. These characters can be used to recognize a wide range of possibilities that can be matched. You can also use these expressions in search and replace operations and also split the text up in numerous ways using the `string.split()` function.

A regular expression is complex and powerful, and is often difficult to read. You can manage without using these expressions most of the time, but these expressions come in handy when you deal with some structured and complex pieces of text. It is always a good idea to take a regular expression slightly slowly, and learn them one at a time. When you try to learn the entire expression in one go, it can be pretty overwhelming. A regular expression matching operation is provided by the module 're.' This module is not a default module, and you will need to import it before you can use it.

```
>>> import re
```

The module supports both 8-bit and Unicode strings, so it should be possible to recognize any characters that you can type in from the keyboard or read from a file.

Next, you need to construct a regular expression string to represent the pattern you want to catch. Let's use the rather colorful string from earlier in the chapter again.

Creating a Regular Expression Object

You can compile a regular expression pattern into a regular expression object using `re.compile(pattern[, flags])`. This returns a pattern object on which you can call all the previous methods, but you no longer need to provide the pattern as an argument. You might want to do this if you need to use this pattern to perform lots of matches all at once; compiling the pattern into an object speeds up the pattern comparison process, because the pattern does not have to then be compiled each time it is used. You might also need to use this approach where a function specifically requires a regular expression object to be passed to it rather than just a pattern string.

Chapter Seven

How to Use Files

You would have noticed in the previous chapters that the data has either been written into the program by itself or has been received using the `input()` function and printed using the `print()` function. When the program has finished its execution, the data that is stored in the temporary memory is lost. If you want an application to always use a specific value, you must have the ability to store that information so it can be retrieved when the program is run again. Most of the information on the computer is stored on the hard drive or any other similar medium. It can also be transferred using a file-like object. A file-like object will share a few similar properties with the files and can be treated in the same manner.

How to Open Files

A file object can be created in Python using the built-in `open()` function. This function will take the parameters `filename[, mode[, and buffering]]`.

Built-in functions and methods also return file objects. Let us open a plain text file in the same directory where we started the interpreter.

```
>>> open('python.txt')
```

```
<io.TextIOWrapper object at 0xb7ba990c>
```

In the above example, we are using another Python object that states that the object is an `io.TextIOWrapper`. This means that the file is an object. If this file does not exist, you will receive an error. Before you start using different

file methods, you should understand how you can create a file object and modify it using Python.

Modes and Buffers

When you open a file by using the `open()` function and pass the name of the file, you will create a read-only object. If you want to make changes to the file, you must set the optional mode. This is an argument that can be a single character - `r`(read), `w`(write) or `a`(append). These characters can be followed by `+`(read/write) and `b`(binary). If you do not want to provide a mode argument, the `r` mode will be assumed as a default option by Python.

```
>>> open('python.txt', 'rb')
```

```
<io.BufferedReader object at 0xb7ba990c>
```

Python will return a different type of object when you use the `b` mode. This object will contain the same information that you want to use in the byte format. You can use this if you wish to handle the audio or image data. The write mode (`w`) will let you change the content of the file fully. You should append the mode `'a'` when you want to add any information to the end. The last option is used to create a log. The buffering argument can either be 0 or 1. If it is the former, the data is written directly onto the hard drive. If it is the latter, Python will create a temporary file that you can use to store the text before it is written out. The data is only written to the disk if you explicitly mention that it should by using the functions `file.flush()` or `file.close()`. This option does not need to be used immediately.

Reading and Writing

One of the most basic methods that one can use to access the contents of a file is to use the function `file.read([size])`. This function will read the size of the file in bytes. The complete file is read as one string if there is no size

argument provided or if it is negative. Unless the file is opened as a binary object, the bytes are returned as string objects. In such cases, you will only have raw bytes as output. If you are reading any plain text file that contains ordinary characters, you will notice that there is not a great deal of difference.

```
>>> text = open('python.txt')
```

```
>>> text.read()
```

```
'Are you keen to learn the python language
```

```
[... pages more text here ...]
```

```
Thank you for purchasing the book.\n'
```

In the above example, we are dealing with a large body of code. If you do not want to work on large blocks of code, you can break the code down into smaller chunks. You can do this by using the function `file.readline([size])`, which will read a single line from the file. An incomplete line may be returned in an iteration. The size argument is defined as the number of bytes that the interpreter must read. The bytes also include the trailing newline. If the interpreter reaches the end of the file, an empty string is returned.

```
>>> text = open('python.txt')
```

```
>>> text.readline()
```

```
“Are you keen to learn more about the python language. Thank you for purchasing the book. I hope you gather all the information you were looking for.”
```

A file is its own iterator. So, it makes it easier for you to iterate through the different lines of code in a file using the for loop. The same result as the `file.readline()` method is returned at each iteration and the loop only ends

when the method returns a null or empty string.

Try the following code out:

```
>>> for line in text:  
    print (line)
```

Closing Files

You should always close the file when you are done using it using the function `file.close()`. The interpreter will notice that the file is not being used, and will clear the memory space if you do not close the file. That being said, it is always better to close the file when you finish using it. The data that is present in the `file.write()` operation will be stored in a buffer until you close the file. If you want to ensure that the data is not written in the file, use the `file.flush()` function. A file once closed cannot be further read or written. You are allowed to call `close()` more than once.

Chapter Eight

Working with Functions

X

When you create a new function, you will need to understand the type of input that the function will need and what information it will return. It is also important to identify the structure and the type of the data that you will be feeding into the function. The data that is given to a function is termed as the parameter and the information returned by a function is known as the output or the result. The initial specification for any function design should include the general description of the specific purpose of the function.

Defining a Function

A function is always defined using the `def` statement. The word `def` is followed by the function name, an optional list of parameters and the line ends with a colon which indicates that the subsequent lines should be indented as a suite or block of instructions. Let's start with a function that takes no parameters:

```
>>> def generate_rpc():  
    """Role-Playing Character generator  
    """  
    profile = {}  
    print "New Character"  
    return profile
```

This block of instructions proceeds in exactly the same way as a complete

script. So, in essence, we give a name to each piece of functionality, and they are called functions.

If you want to throw some light on the purpose of the function and what it does, you can do this in the docstring. This should be the first thing you look at when you write a function. The docstring will be followed by some statements that will explain the core functionality of the function, and will be followed by the lines of code. You can use the function to return some value using the return statement.

In the above example, the last line of the function is used to specify the variables that will be returned to the main program. If you do not have to return anything, you can avoid using the return statement, and Python will assume that nothing should be returned to the main program. The block of code that you have written in your function has not been run yet, but has been assigned to the definition of the function. You will need to call the function in the main program to run the code.

Since we have given names to functions, we can call those functions any number of times:

```
>>> generate_rpc()
```

New Character

```
{}
```

In the above example, we have not specified any parameters, and it is for this reason that the parentheses after the function name are empty.

Defining Parameters

Most functions always work on some data that has been provided to them using the main program. If you want the function to receive the data, you will

need to set up some containers that can hold the data. These containers will become variables that are always unique to the function, and are known as formal parameters. It is always a good idea to ensure that these variables do not have the same name as other variables in the program. The formal parameters that you specify in the program will need to be mentioned in parentheses once you define the name of the function in the first line of the definition.

```
import random

def roll(sides, dice):

    result = 0
    for rolls in range(0,dice):

        result += random.randint(1,sides)

    return result
```

You can call this function from the main program using the following line of code:

```
muscle = roll(33,3)
```

The values in the parentheses are called arguments, and these arguments correspond to the formal parameters that can be found in the definition of the function. In the example above, the first argument being used is 33 and this argument is bound to the parameter sides. The second is 3 that is bound to the parameter dice. This will create two variables that can be used within the function. If you send values like this to the function, you will need to ensure that you have the same number of parameters and arguments. You can substitute the actual values that are present within the function using variables that you want. You must only remember that the function will refer to the value using the name of the parameter that obtains the value. The

original variable will not be affected, and only its value is passed on to the parameter.

Documenting your Function

When you have written a function fully, and the script in the function will pass the tests that you have created for the code, you should begin editing the docstring to explain the use of the function.

You are required to follow some conventions when you are writing a docstring. The first line of the docstring should be very short, and should describe the function that is being used. This statement should make sense by itself. The second line of the docstring should always be left blank. The body of the docstring should contain an explanation of the parameters within the function, some details about the function, an explanation of the algorithm used and an example of how you can use the function including some information about keyword arguments, optional variables and the values that the function will return.

You can also choose to include some information about some errors or exceptions one may encounter. You can also talk about some of the restrictions of the function. In short, the information that the programmer will need to know to understand the function better should be present in the docstring. You must remember to update the comments and the docstring every time you make a change to the code.

Working with Scope

It is easier to look at a function as a black box that takes in some data, processes that data and returns the required values to the user. The code written in the main section of the program will send the source data and receive the results. This code is known as the calling procedure. This

procedure does not need to know the data that is present in the black box, as long as the source data and the results have been identified clearly.

Understanding Scope

You never have to worry about naming a function when you are writing your program. We have the concept of scope only to cater to this purpose.

When you run a program in Python, the interpreter will create a list of the names being used in the program, and will keep a track of those names. The names are placed in a table called the symbol table and will be used by the interpreter as the dictionary. The variables that are created in the symbol table of the program are known as global variables since these variables can be accessed by any part of the program. These variables can be viewed using the `globals()` function. The result of running `globals()` is the same as the result of running `vars()` without including any arguments. The scripts that we have looked at in the book only use global variables.

A variable that is created within a function will be stored in the symbol table. These variables are unique to the function alone. The data is known as local data and can only be accessed using the `locals()` function. The main body of the program will not allow you to access variables that are set in the function. The function will still be able to access the variables in the main program. This means that you have two ways to process data in a function. The easiest way to do this is to take the data that you want to use as a parameter for your function. The function will process that data and return the required result. This result will then be used by the main program. The other way is to allow the function to access any global data and process it as required. It is best not to use this method since the function will only work with specific variables names, and cannot be used anywhere else.

Manipulating Dictionaries and Lists

Every parameter to a function is passed using a value. In the case of an object, which is different from a string or an integer, any changes made to the object are reflected outside of the function as well. This means that a function can always be used to modify mutable data types like lists or dictionaries. A pure function is often not used to perform any modifications to the data since the function will not return any value or use any input parameters.

If you want to avoid any effects of using a mutable function, it is always a good idea to only use or return immutable values in a function. You should always keep the modification procedures separate. If you send a dictionary or a list to any other function, you are only sending the value to a pointer that will have the same global instance for the object as if you had written `local_list = global_list`.

Abstraction

You will identify some new issues when you work on this aspect of testing. If you want to keep track of what is happening with your code, you must write the minimum necessary code to ensure that the code passes the test. If the new code will fail, you must roll all the changes back to the point where the code will pass the test. If you use a function that does not work, you should not worry too much about it. All you need to do is reverse the changes that you have made and move to the next section of your code. There is no rule that states that everything that you write in a program should be placed within a function.

Abstraction is the process of shifting chunks of code into a function and turning the code that deals with a general idea into a smaller section of code that can be used anywhere within the script. Once you create functions, you can move them outside the code and call upon them whenever necessary, and

it is for this reason that abstraction is a good method to use in your code. The only rule that you need to keep in mind is that you should always write the test code first and ensure that it passes through the interpreter before you refactor it. This approach will seem laborious, but it is the best way to develop a new code with no errors. You can focus on writing newer pieces of code instead of worrying about the details of the code.

Chapter Nine

Data Visualization

“By visualizing information, we turn it into a landscape that you can explore with your eyes. A sort of information map. And when you’re lost in information, an information map is kind of useful.” – David McCandless

Here is a fun fact to start this chapter with. 90 percent of the information that is transmitted to the brain is visual in nature.

We have learnt how data science is an essential platform that helps growth, development and evolution of a business by helping it form and implement strategies using insights that are driven completely by data. Data that is digital in nature not only helps perceive important insights to a business, but if this data is presented in a manner that is digestible, inspiring and a logical format, it is like telling a story to everyone in an organization and getting them onboard with your vision as a data scientist.

The part where we represent this data such that everyone in an organization who is not really tech savvy can understand too, is the part where visualization of data comes into the picture. Visualization has a big part to play in data analytics and refers to creation of representations that are graphical in nature. This whole process of data visualization helps interpret patterns in data by having a quick look at it and helps structure data in real time while still preserving the backend data that is complex in nature owing to its factual and numerical figures.

Data interpretation is the biggest challenge in organizations that have huge

sets of data readily available for analysis. Therefore, the aspect of data interpretation is very critical if we are looking at the goals, aims and long term objectives of an organization.

This is where data visualization comes into the picture.

The human brain can remember visuals way more comfortably as compared to numbers and letters. Therefore, representation of data that is huge and complex in nature in the form of graphs or charts is more convenient as compared to reports or spreadsheets.

Critical aspects and concepts of data can be conveyed in a simple, intuitive and swift manner with the help of visualization techniques. Visualization also help data scientists to make experiments with data based on different scenarios by allowing to make tiny adjustments.

Visualization of data has proved to be very beneficial to organizations. It has been observed that business meeting durations can be lowered by 24 percent if data is represented in a visual format as compared to raw data. Another study shows that with the use of visualization techniques, return on investments for a business could be increased to USD 13.00 for every dollar that is spent.

It can therefore be concluded that business success rates can improve tremendously with the aid of visualization techniques and a business can yield a value which is optimum by using this technique that has already been tried and tested to achieve results. Let us go through the 10 most essential techniques that are available for data visualization in the industry today.

Know your Audience

The concept of knowing your audience is the most overlooked of all and yet is one of the most vital concepts in data visualization.

We can safely assume that the world wide web, the internet, and information technology as a whole are still in its infant stages. Furthermore, we can safely assume that data visualization is even a far younger concept in comparison. Even the most established entrepreneurs of the 21st century sometime find it difficult to understand one single pie chart, or a neatly presented set of visuals, or mostly do not have the time to sit through and deep dive into the data available even via graphical representations. Therefore, it is very important that the data that you are converting into a visual format is interesting and tailored to suit the audience that you are presenting it to. This is why it is very important to know your audience before you put forth a set of visual data in front of them.

Set your Goals

As with many businesses, right from the story telling of your brand to selling your products digitally and beyond, visualization of data and your efforts will only yield if the strategy behind your business is concrete. At the time when you are creating a visualization of data for your business, it is important that the visuals show a narrative that is logical and that show insights that are most relevant to your business. By creating a goal for your pursuits or campaigns, you should sit down with all the stakeholders and explain your goals to them until they are as invested as you are in your goals and dreams. One of the ways of achieving this is by setting KPIs for your business that are predetermined, and use these KPIs as an input to your visualizations.

Choose the Right Type of Charts

Selecting the right kind of charts to represent your data plays a very important role. Therefore, it is very critical to select the right type of charts to represent your data effectively, all while keeping in mind the project in concern, the purpose of the project and the audience.

For example, if the project is showing changes that happened over various time periods for a business and shows only a few insights, using a simple line graph or a bar graph would be the most optimized techniques of representing data visually.

Let us go through the most popular chart types that are used to represent data visually.

Number Charts

Number charts are very efficient and effective when the data is supposed to show an indicator that is of key performance such as site visits for a website, likes on a picture on instagram, or even sales KPIs of a company.

Maps

The biggest advantage of using maps is that they are fun to look at which mean that the audience that the map is being presented to

(such as a board panel or presentation) will be highly engaged. The second advantage is representing data using maps is easy and quick and large sets of complex data on information of geography or other things can be digested easily when shown using maps.

Pie Charts

Pie charts have been considered to be the most traditional way to represent data and have received a lot of negative feedback in the recent years. We still feel that pie charts are still a great tool for visualization of data and are easy to follow.

Gauge Charts

Data that has single values or data points can be efficiently represented using gauge charts. Gauge charts are one of the best visual representations to

display instant indication of trends, whether it be dashboards used in financial organizations or for executive dashboard reports.

The Color Theory Advantage

This is the most straightforward and basic technique, which is to be taken care of during data visualization - selection of a color scheme that is appropriate and relevant to the data such that it significantly enhances your efforts.

The color theory plays a very important part in making your visualization model a success or a failure. Consistency is the key and you should always maintain a scheme that is consistent across your models. You should distinguish elements in your models by using clear contrasting color schemes. (example: negative trends in red and positive trends in green).

Handling Big Data

It is estimated that by 2020, there will be 1.7 megabytes of data that will be generated every second for every human that exists on the planet. This can be overwhelming and will deliver great insights in the digital world that we are marching towards. It can be a real challenge to handle this data, interpret it and present it whenever and wherever required. The following tips will help you learn how to manage the big data that is going to be generated.

There will be a lot of data available and you will need to decide how much of this data holds true value for you or your organization.

To ensure that data is managed smoothly across all departments, you need to ensure that all your colleagues and other people working on your project know your sources of data.

Always protect your data and keep your data handling systems simple such

that they can be converted into visuals comfortably and everyone finds it easy to understand.

Business dashboards should be easy to access and must show all the valuable insights of your projects.

Prioritize using Ordering, Layout and Hierarchy

Following up on our discussion of the previous topic, after you have categorized your data based on how much of it is valuable to your organization, the next step should be to dig further and creating a hierarchy in your data by labelling it. You should prioritize your data by using something like a color code based on how important a set of data is. This will help you to assign a visualization model to every data set based on its importance and that will bring out the best out of every visualization model.

Utilization of Network Diagrams and Word Clouds

Network Diagrams or Word Clouds come handy when you are dealing with visualization of data that is unstructured or semi-structured.

When you need to draw the graphical chart of a network, a network diagram is used. This technique is usually used by designers, network engineers, data analysts, etc. when they need to compile a network documentation in comprehensive formats.

On the other hand, complex data consisting of unstructured information can be presented in an efficient manner using word clouds. In contrast with a network diagram, a word cloud is an image that is created using words that are used for a particular subject or text. The importance and frequency of each word is represented by the size of that word.

Comparisons

This is a data visualization technique that is very brief in nature but it is still important according to us. Comparison as many as possible should be put forward whenever you are presenting your insights and information. You can show the same information over two different timeframes and draw comparisons between them using two or more graphs. This helps drill down the information deep into the brains of the audience that you have and they will remember it.

Telling a Story

As one may see in content marketing, even when it comes to presenting data in front of an audience, you should make it feel like telling a story of how the data originated and then evolved further and how it will eventually prove to be beneficial to the organization. Observation shows that an audience stays more focused and engaged when a presentation is done in the form of a story.

Chapter Ten

Visualization Tools for the Digital Age

We have come far away from the yesteryears where we would use a pen and paper or even do a copy paste of sorts. Therefore, it is very crucial that you materialize on all the digital tools available today to make your visualization of data a success.

A dashboard tool that is interactive and task-specific offers a simple and comprehensive means of retrieving, extracting, collating, organizing and presenting data with a lot of comfort. This ensures that with minimal amount of time taken, the impact is great.

7 Best Data Visualization Tools

It is fortunate that data science and visualization techniques are evolving at par with the rest of the technology stack.

Let us go through a few of the best, innovative and most popular tools available in the data visualization domain that are available today. All these tools are paid tools although they do offer trial periods and licenses for personal use.

Tableau

Tableau is popularly known as the grand master of software in the data visualization domain and it has its reasons for being called so. With 57000 plus users, it is a software used across industries because of its simplicity to

use and because in comparison with regular business intelligence solutions, tableau provides visualizations that are far more interactive. Big Data operations, which deal with huge datasets, can be processed using tableau. It also supports integration of machine learning and artificial intelligence applications since it works with the latest database solutions such as My SQL, Amazon AWS, SAP, Teradata and Hadoop. There is a lot of research that has gone into the development of tableau to make it efficient for graphics and visualization, and to make the whole process simple and easy for humans.

Qlikview

Qlikview is a tool developed by Qlik and is a major player in the data science space and the biggest competitor of Tableau. The software has a customer base of over 40,000 spread over 100 countries. The frequent users have praised its setup, which is highly customizable with a wide range of features. This could therefore mean that it takes time to get a grasp of this tool and use it in its complete potential. In addition to visualization of data, Qlikview is known for its solutions for analytics, business intelligence, and reporting capabilities, and users like it particularly for its interface that is neat and clutter-free. It is used in tandem with its sister package, QlikSense, which works very well with discovery and exploration of data. The community for Qlikview is very strong and there are a lot of resources available online which are maintained by third parties to help new users get comfortable with the tool.

FusionCharts

Based on JavaScript, FusionCharts is a widely used tool for visualization and charting and it has taken as one of the leaders in the paid market. With the ability to produce over 90 chart types, it is known for its flexibility to integrate seamlessly with popular frameworks and platforms. FusionCharts is

also popular because it allows users to use existing templates for visualization instead of starting their own charts from scratch.

Highcharts

Highcharts is like FusionCharts in the sense that it requires a paid license to be used commercially, but it can be used for free as a trial, and also can be used non-commercially for personal use. Highcharts boasts that it is used of 72 percent of the 100 largest companies in the world and it is often the first choice of users when a flexible and fast solution is needed, with minimal training required before it can be used. Cross-browser support has been its key to success, powering anyone to run and view its visualizations, which is not easy with other available platforms.

Datawrapper

Media organizations that use a lot of data to show charts and make presentations based on statistics are using Datawrapper on a large-scale basis. Simplicity of the interface and the ease of uploading data in CSV format resulting into maps, straightforward charts, etc which can be embedded quickly into reports is what makes it a popular choice.

Plotly

Plotly is a tool that supports visualizations that are sophisticated and complex in nature, given its easy integration with popular programming languages such as R, Python and Matlab. Build on the foundations of open source libraries for JavaScript such as d3.js, Plotly is a paid package and has non-commercial licenses for personal use.

Sisense

Sisense is a platform that is full stack and provides us with a visualization interface that uses drag and drop capabilities. This facilitates complex graphics, charts and interactive visualizations, to be created at the few clicks

of the mouse. It provides a repository where you can collect data from multiple sources and then allows you to query the repository to access any data, even if the set of data is huge. It allows dashboard sharing across organizations making sure that even people who are not technically very sound can get all the answers to their data problems.

10 Useful Python Data Visualization Libraries

If you scroll through the package index available in Python, you will find python libraries that will meet the needs of almost all your visualization requirements.

Let us go through the 10 most popular and useful python libraries to better your visualization techniques.

Matplotlib

matplotlib is the first libraries available in python for data visualization. Developed over a decade ago, it is still one of the most preferred python libraries by data scientists. It was built on the foundation of MATLAB, which was developed in the 1980s. Given that matplotlib was the first library built in python for data visualization, many other libraries were built further on the foundations of matplotlib such that they could run in tandem with matplotlib during the analysis process.

Seaborn

The abilities of matplotlib are harnessed by seaborn, which creates charts that are beautiful by using just a few lines of code. Therefore, in order to be able to tweak the default settings of seaborn, it is very important to have knowledge of matplotlib.

ggplot

Based on ggplot2, ggplot is a plotting system in R. The operation of ggplot is

different from that of matplotlib, in the sense that it creates a complete plot using layered components.

pygal

pygal provides integration plots that can be embedded and merged in web browsers. The main advantage of pygal is that it renders output charts in Scalable Vector Graphics(SVG) format. SVG is a good format to work with smaller data sets. Using this format with huge data sets will render the charts sluggish.

Plotly

Plotly is popular for data visualization as an online platform, but very few know that it allows you to access it from Python notebook. Plotly's forte is creating interactive plots and also that it offers charts that are not available in other libraries such as dendrograms, contour plots, etc.

geoplotlib

geoplotlib as the name goes is a tool used to plot geographical data by creating maps. It can be used to create maps of different types such as heatmaps, dot density maps and choropleths. Pyglet should be installed as a prerequisite to be able to use geoplotlib.

Gleam

Gleam is a python library, which is inspired by the Shiny package library, is available in R. You can turn your analysis into web apps using scripts in Python, which makes it okay if you don't know technologies such as HTML, CSS and javaScript.

missingno

It can be very challenging and painful to deal with missing data. Missingno helps you fill in the gaps with visual summary in the event of missing data.

Leather

Christopher Groskopf, creator of Leather describes it as “Leather is the Python charting library for those who need charts now and don’t care if they’re perfect.” Designed such that it can work with data of all types, leather renders charts in SVG and hence they are scalable without loss of quality

Chapter Eleven

An Introduction To Outlier Detection In Python

The previous chapter shed some light on anomaly detection and other techniques that are used to analyze data. This chapter will explain how outlier detection works in Python. We will also look at an example to understand this better.

What is an Outlier?

In every data set that you use, you will see some data points that vastly differ from other observations within the data set. Let us look at some examples to help us detect outliers in a data set.

- When a student scores 90% in an examination while the rest of the class can only score 70%, you can call the student an outlier.
- When you analyze a customer's purchasing patterns, it will turn out that there are some entries that have a very high value. There could be some transactions for \$1,000, while there could be one for \$10,000. This could be because they purchased an electronic or any other reason. This data point is an outlier in the data set.
- Usain Bolt is a perfect example. His record breaking sprints are outliers when you look at the time taken by most athletes to complete a run.

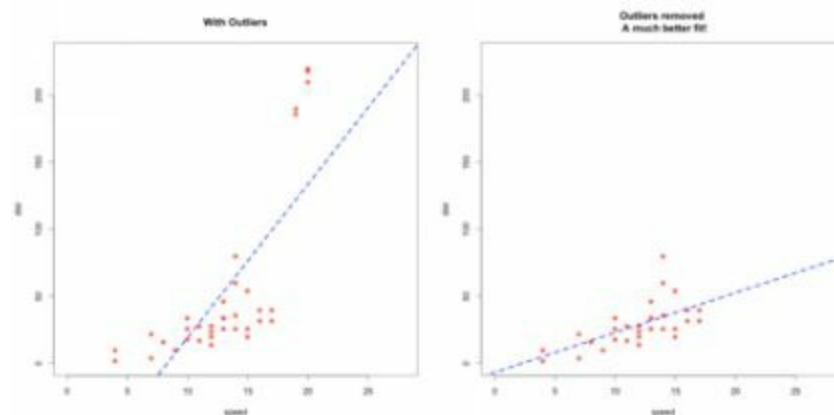
There are numerous reasons why there are outliers in the data set. The

machine may have made an error in measurement, the outlier could have intentionally been placed in the data set or the analyst made an error while making an entry. There are some people who add false information to the data set because they do not want to disclose any information.

There are two types of outliers – univariate and multivariate. If the data set has extreme values only for one variable, it is called a univariate outlier. If the data set has two variables that have an unusual score when combined, it is called a multivariate outlier. If you have three variables X, Y and Z and you plot a graph for these variables in a 3-D space, you will see a cloud. The data points that lie outside the cloud are called the multivariate outliers.

Why Do We Need To Detect Outliers?

An outlier can impact the results of any statistical modeling or analysis in a drastic way. The image below will show you how outliers present in the data will affect the analysis, and what happens when you remove the outliers from the data set.



Here is something that you must understand – an outlier is not a bad thing. It is important for you to understand this. You cannot remove an outlier from your data set without taking into account how this will affect the other points in the data set.

“Outliers are not necessarily a bad thing. These are just observations that are not following the same pattern as the other ones. But it can be the case that an outlier is very interesting. For example, if in a biological experiment, a rat is not dead whereas all others are, then it would be very interesting to understand why. This could lead to new scientific discoveries. So, it is important to detect outliers.”

– Pierre Lafaye de Micheaux, Author and Statistician

Most people use methods like scatter plots, histograms and box plots to identify the outliers in the data set since these methods are straightforward. That being said, it is extremely important to use outlier detection algorithms in some fields where you need to process large volumes of data. These algorithms will use pattern recognitions and other techniques to identify the outliers in large data sets.

Applications like intrusion detection in network security and fraud detection in finance require some accurate and intense techniques to detect the outliers in the data set. How embarrassed would you be if you considered a data point to be an outlier, when in reality it is a data point that you need to consider?

This gap can be bridged by using the PyOD library. Let us now understand what this library is all about.

Why Should We Use PyOD For Outlier Detection?

There are numerous programming languages that have outlier detection packages that can be used on any data set. That being said, there is a lack of different models for outlier detection in Python. It is surprising that this can happen, isn't it? There are some implementations like PyNomaly, which cannot be used for outlier detection, since these implementations were not designed for those alone. It was only to fill the gap that the library PyOD was

developed. This library is a scalable Python toolkit that will allow you to detect outliers in any data set. This library contains close to twenty algorithms that can be used for outlier detection.

Features of PyOD

There are numerous advantages to using PyOD because of the many features. Let us look at some of these features:

- This library is an open-source application, and the documentation for this library is detailed. There are numerous examples across different algorithms.
- This library supports advanced models including deep learning, outlier ensembles and neural networks.
- This library is compatible with every version of Python.
- This library optimizes the performance of the model using joblib and numba.

Installing PyOD in Python

Let us now power up the notebooks on Python and install PyOD on the machine.

```
pip install pyod
```

```
pip install --upgrade pyod # to make sure that the latest version is installed!
```

It is that simple.

Outlier Detection Algorithms Used In PyOD

Let us now look at the different algorithms that are used by PyOD to detect outliers in the data set. It is important to understand how PyOD works in

Python since it will give you more flexibility to understand what method you are using on your data set. You will notice the terms outlying score across this chapter. This means that every model that is used in PyOD assigns every data point in the data set a score. It will then use that score and compare it against a benchmark to see whether the point is an outlier or not.

Angle-Based Outlier Detection (ABOD)

- This method uses the relationship that exists between every point in the data set and the neighbors. This method does not look at the relationship or the similarities between the neighbors. The outlying score is calculated as the variance of the weighted cosines of the scores of each neighbor.
- This method can be used on multi-dimensional data.
- There are two versions that are present in PyOD for ABOD:
 - **Original ABOD:** Considers all training points with high-time complexity
 - **Fast ABOD:** This method approximates using the k-nearest neighbors algorithm

k-Nearest Neighbors Detector

- In this method, the distance between the data point and the nearest kth neighbor is used to determine the outlying score.
- ✓ **PyOD supports three [kNN](#) detectors:**
 - **Mean:** This method calculates the outlier score as the average of the k nearest neighbors
 - **Largest:** This method calculates the outlier score as the distance of the kth nearest neighbor

- **Median:** This method calculates the outlier score based on the median of the distance between the point and the k neighbors

Isolation Forest

- The isolation forest method uses the scikit-learn library. In this method, the data is split using decision trees. The isolation score for every data point is provided based on how far the point is from the structure. The score will be used to identify those points that can be considered as outliers.
- This method can be used on multi-dimensional data.

Histogram-based Outlier Detection

- This method is one of the most efficient methods that is used to identify outliers in a data set. This method uses the independence of features and will calculate the outlier score using histograms.
- This method is less precise, but provides results faster when compared to other methods.

Local Correlation Integral (LOCI)

- LOCI is an extremely effective method to use to detect any outliers or a group of outliers in the data set. Python will use the data points to calculate the LOCI. This point will provide a lot of information about the data set. It will also help the model determine the clusters, the diameters of clusters, micro-clusters and the distance between the clusters.
- You cannot compare this method to any other method since the output provided by this method is only one number for every data point.

Feature Bagging

- Most often we choose to break the data set into smaller sets or samples. You can dit the number of base detectors in each of these samples using a feature bagging detector. This detector will use the average or combine different methods to improve the prediction accuracy.
- Python uses the Local Outlier Factor (LOF) as the base estimator. You can also choose to the ABOD or KNN as the base estimator if you do not want to use the default.
- In feature bagging, Python will first construct the samples using the data set, and randomly select a subset of the attributes or features in that sample. This will help you identify the diversity of the estimators. The model will then calculate the prediction score by calculating the average or using the maximum of the base detectors.

Clustering Based Local Outlier Factor

- The data us broken into large and small clusters. Python then calculates the anomaly score based on the distance between the point and the nearest cluster which is fairly larger than the cluster it is in and based on the size of the cluster that is allotted to.

Extra Utilities Provided by PyOD

- You can generate some random data that has outliers using the function `generate_data`. A multivariate Gaussian distribution is used to generate the list of inliers in the data set. The outliers are often generated using a uniform distribution.
- You are allowed to choose the fraction of outliers you want to include in your data set. You can also choose the number of samples that you want to include in the data set. This utility function will help you

create the data set that you want to use when you implement the models.

Implementation of PyOD

Now that we have gathered a good understanding of what an outlier is and how it can be detected, let us work on some examples and algorithms. In this section, we will be working on implementing a PyOD library to detect the outliers. We will be using the following approaches to identify outliers:

- Using the big mart sales challenge data set
- Using a simulated data set

PyOD on a Simulated Dataset

Before you look for the outliers, you should import the different libraries that you will be using.

```
import numpy as np

from scipy import stats

import matplotlib.pyplot as plt

%matplotlib inline

import matplotlib.font_manager
```

The next step is to detect the outliers that are present in the data set. We need to do this by using different models. For the purpose of this example, we will be using the ABOD (Angle Based Outlier Detector) and KNN (K Nearest Neighbors) algorithms:

```
from pyod.models.abod import ABOD

from pyod.models.knn import KNN
```

Let us now create a random data set that has outliers in it. Once we are happy with the data set, we will plot it.

```
from pyod.utils.data import generate_data, get_outliers_inliers

#generate random data with two features

X_train, Y_train = generate_data(n_train=200,train_only=True,
n_features=2)

# by default the outlier fraction is 0.1 in generate data function

outlier_fraction = 0.1

# store outliers and inliers in different numpy arrays

x_outliers, x_inliers = get_outliers_inliers(X_train,Y_train)

n_inliers = len(x_inliers)

n_outliers = len(x_outliers)

#separate the two features and use it to plot the data

F1 = X_train[:,[0]].reshape(-1,1)

F2 = X_train[:,[1]].reshape(-1,1)

# create a meshgrid

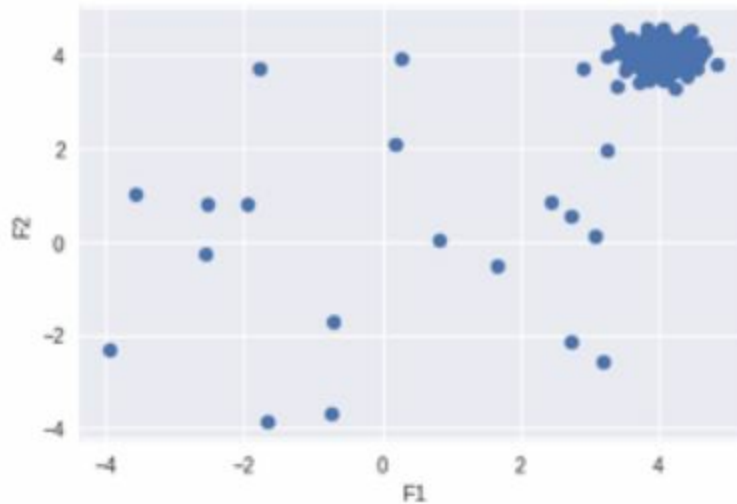
xx , yy = np.meshgrid(np.linspace(-10, 10, 200), np.linspace(-10, 10,
200))

# scatter plot

plt.scatter(F1,F2)

plt.xlabel('F1')
```

```
plt.ylabel('F2')
```



Let us now create a dictionary that we will be using to detect the outliers. We will add the models that we want to use to the dictionary.

```
classifiers = {  
    'Angle-based Outlier Detector (ABOD)' :  
    ABOD(contamination=outlier_fraction),  
    'K Nearest Neighbors (KNN)' :  
    KNN(contamination=outlier_fraction)  
}
```

Every model that is present in the dictionary should now be provided with the data. Ensure that the data is fit accurately to each model. You should now see how the model is identifying the outliers.

```
#set the figure size
```

```
plt.figure(figsize=(10, 10))
```

```
for i, (clf_name,clf) in enumerate(classifiers.items()) :
```

```

# fit the dataset to the model
    clf.fit(X_train)

# predict raw anomaly score
    scores_pred = clf.decision_function(X_train)*-1

# prediction of a datapoint category outlier or inlier
    y_pred = clf.predict(X_train)

# no of errors in prediction
    n_errors = (y_pred != Y_train).sum()
    print('No of Errors : ',clf_name, n_errors)

# rest of the code is to create the visualization

# threshold value to consider a datapoint inlier or outlier
    threshold = stats.scoreatpercentile(scores_pred,100 *outlier_fraction)

# decision function calculates the raw anomaly score for every point
    Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()]) * -1
    Z = Z.reshape(xx.shape)

    subplot = plt.subplot(1, 2, i + 1)

# fill blue colormap from minimum anomaly score to threshold value
    subplot.contourf(xx, yy, Z, levels = np.linspace(Z.min(), threshold,
    10),cmap=plt.cm.Blues_r)

# draw red contour line where anomaly score is equal to threshold

```

```

a = subplot.contour(xx, yy, Z, levels=[threshold],linewidths=2,
                    colors='red')

# fill orange contour lines where range of anomaly score is from threshold to
# maximum anomaly score

subplot.contourf(xx, yy, Z, levels=[threshold,
                                   Z.max()],colors='orange')

# scatter plot of inliers with white dots

b = subplot.scatter(X_train[:-n_outliers, 0], X_train[:-n_outliers, 1],
                    c='white',s=20, edgecolor='k')

# scatter plot of outliers with black dots

c = subplot.scatter(X_train[-n_outliers:, 0], X_train[-n_outliers:, 1],
                    c='black',s=20, edgecolor='k')

subplot.axis('tight')

subplot.legend(
[a.collections[0], b, c],
['learned decision function', 'true inliers', 'true outliers'],
prop=matplotlib.font_manager.FontProperties(size=10),
loc='lower right')

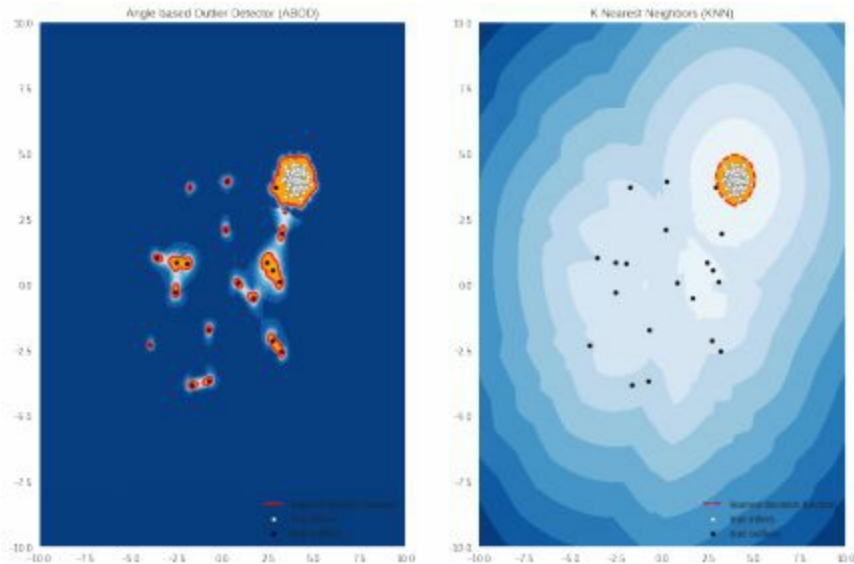
subplot.set_title(clf_name)

subplot.set_xlim((-10, 10))

subplot.set_ylim((-10, 10))

plt.show()

```

This looks perfect.

PyOD on the Big Mart Sales Problem

To better understand the Big Mart Sales Problem, we will now use the PyOD function. To understand the problem better, please check the following link: https://datahack.analyticsvidhya.com/contest/practice-problem-big-mart-sales-iii/?utm_source=outlierdetectionpyod&utm_medium=blog. Download the data set we will be using for this example from the above link. We will first need to import the libraries that we will be using to solve this problem, and then load the data set.

```
import pandas as pd

import numpy as np

# Import models

from pyod.models.abod import ABOD

from pyod.models.cblof import CBLOF

from pyod.models.feature_bagging import FeatureBagging
```

```
from pyod.models.hbos import HBOS

from pyod.models.ifeorest import IForest

from pyod.models.knn import KNN

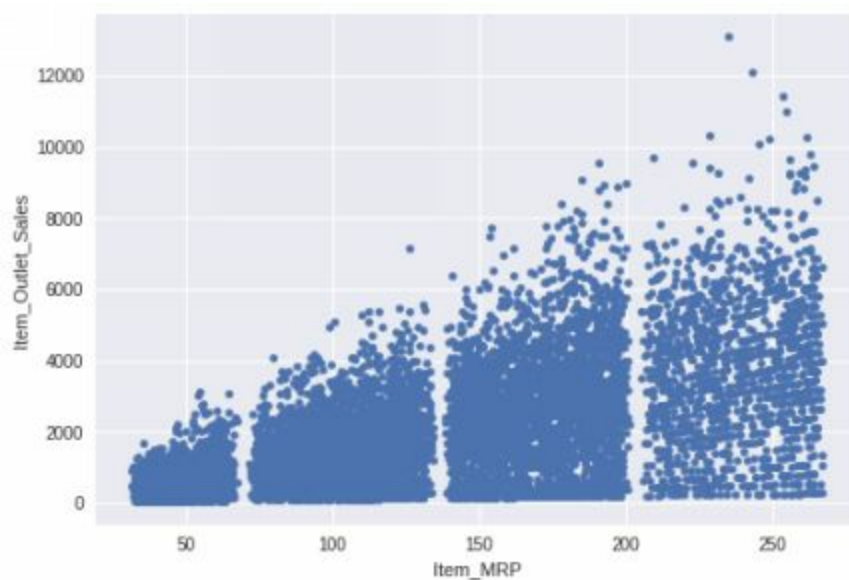
from pyod.models.lof import LOF

# reading the big mart sales training data

df = pd.read_csv("train.csv")
```

To understand the data better, we should plot the values Item MRP and Item Outlet Sales.

```
df.plot.scatter('Item_MRP', 'Item_Outlet_Sales')
```



The value of Item MRP is between zero and 250 while the range of Item Outlet Sales is between zero and 12000. For the purpose of our problem, we will need to scale the range between zero and one. The range will need to be scaled if you wish to create a visualization that is easy to explain. The graph will stretch across a vast range otherwise. For the data that we are using in this example, we will be using the exact same approach to develop that

visualization. If in other problems, you do not necessarily have to use a visualization to predict the outliers, but can choose to use the same scale.

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature_range=(0, 1))

df[['Item_MRP','Item_Outlet_Sales']] =
scaler.fit_transform(df[['Item_MRP','Item_Outlet_Sales']])

df[['Item_MRP','Item_Outlet_Sales']].head()
```

Since we do want to use these values in the later models, you should store them in the NumPy array.

```
X1 = df['Item_MRP'].values.reshape(-1,1)

X2 = df['Item_Outlet_Sales'].values.reshape(-1,1)

X = np.concatenate((X1,X2),axis=1)
```

We will now prepare a dictionary, but will now include some models to that dictionary. We will then need to see how these models will predict an outlier. Based on your understanding of the data and the problem that you are trying to solve, you can set the value of the fraction that you will use to identify the outlier. In the example below, we are instructing the model to look for at least five percent of the observations that are not like the other data. Therefore, we will need to set the value of the fraction as 0.05.

```
random_state = np.random.RandomState(42)

outliers_fraction = 0.05

# Define seven outlier detection tools to be compared

classifiers = {
```

```

'Angle-based Outlier Detector (ABOD)':
ABOD(contamination=outliers_fraction),

'Cluster-based Local Outlier Factor
(CBLOF)':CBLOF(contamination=outliers_fraction,check_estimator=
random_state=random_state),

'Feature
Bagging':FeatureBagging(LOF(n_neighbors=35),contamination=outli

'Histogram-base Outlier Detection (HBOS)':
HBOS(contamination=outliers_fraction),

'Isolation Forest':
IForest(contamination=outliers_fraction,random_state=random_state)

'K Nearest Neighbors (KNN)':
KNN(contamination=outliers_fraction),

'Average KNN':
KNN(method='mean',contamination=outliers_fraction)

}

```

Now, we will fit the data to each model one by one and see how differently each model predicts the outliers.

```

xx , yy = np.meshgrid(np.linspace(0,1 , 200), np.linspace(0, 1, 200))

for i, (clf_name, clf) in enumerate(classifiers.items()):

    clf.fit(X)

# predict raw anomaly score

scores_pred = clf.decision_function(X) * -1

```

```

# prediction of a datapoint category outlier or inlier

y_pred = clf.predict(X)

n_inliers = len(y_pred) - np.count_nonzero(y_pred)

n_outliers = np.count_nonzero(y_pred == 1)

plt.figure(figsize=(10, 10))

# copy of dataframe

dfx = df

dfx['outlier'] = y_pred.tolist()

# IX1 - inlier feature 1, IX2 - inlier feature 2

IX1 = np.array(dfx['Item_MRP'][dfx['outlier'] == 0]).reshape(-1,1)

IX2 = np.array(dfx['Item_Outlet_Sales'][dfx['outlier'] ==
0]).reshape(-1,1)

# OX1 - outlier feature 1, OX2 - outlier feature 2

OX1 = dfx['Item_MRP'][dfx['outlier'] == 1].values.reshape(-1,1)

OX2 = dfx['Item_Outlet_Sales'][dfx['outlier'] ==
1].values.reshape(-1,1)

print('OUTLIERS : ',n_outliers,'INLIERS : ',n_inliers, clf_name)

# threshold value to consider a datapoint inlier or outlier

threshold = stats.scoreatpercentile(scores_pred,100 *
outliers_fraction)

```

```

# decision function calculates the raw anomaly score for every point
Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()]) * -1
Z = Z.reshape(xx.shape)

# fill blue map colormap from minimum anomaly score to threshold value
plt.contourf(xx, yy, Z, levels=np.linspace(Z.min(), threshold,
7),cmap=plt.cm.Blues_r)

# draw red contour line where anomaly score is equal to threshold
a = plt.contour(xx, yy, Z, levels=[threshold],linewidths=2,
colors='red')

# fill orange contour lines where range of anomaly score is from threshold to
maximum anomaly score

plt.contourf(xx, yy, Z, levels=[threshold, Z.max()],colors='orange')
b = plt.scatter(IX1,IX2, c='white',s=20, edgecolor='k')

c = plt.scatter(OX1,OX2, c='black',s=20, edgecolor='k')

plt.axis('tight')

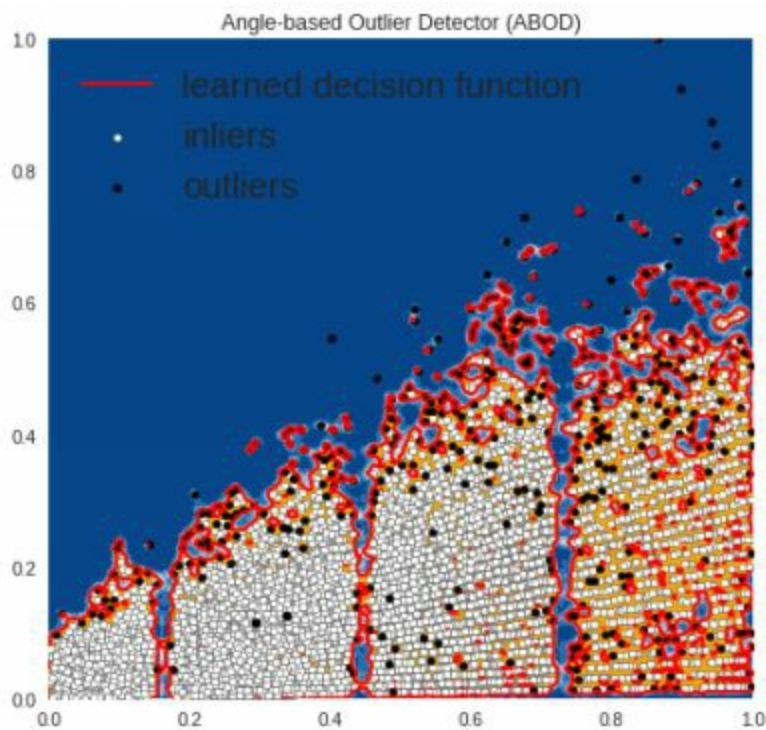
# loc=2 is used for the top left corner
plt.legend(
[a.collections[0], b,c],
['learned decision function', 'inliers','outliers'],
prop=matplotlib.font_manager.FontProperties(size=20),

```

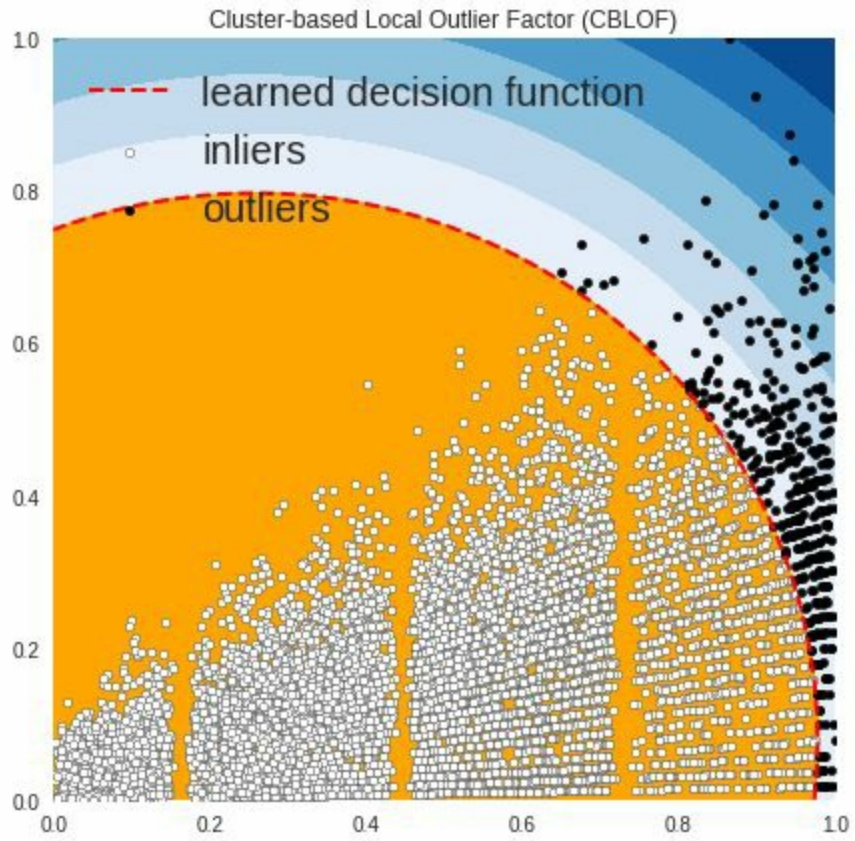
```
loc=2)
plt.xlim((0, 1))
plt.ylim((0, 1))
plt.title(clf_name)
plt.show()
```

OUTPUT

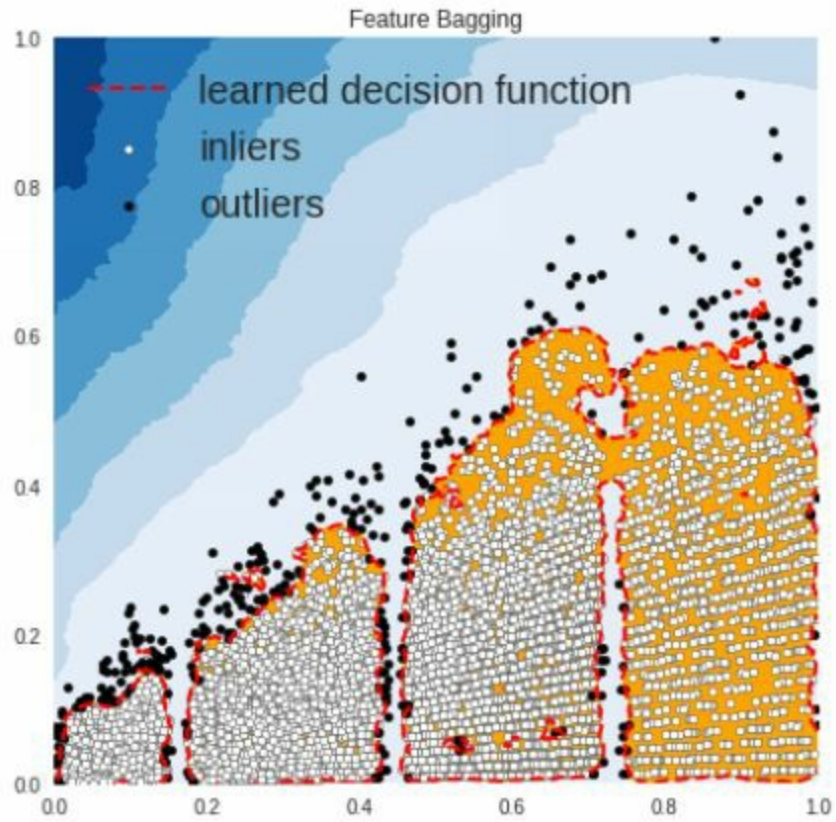
OUTLIERS : 447 INLIERS : 8076 Angle-based Outlier Detector (ABOD)



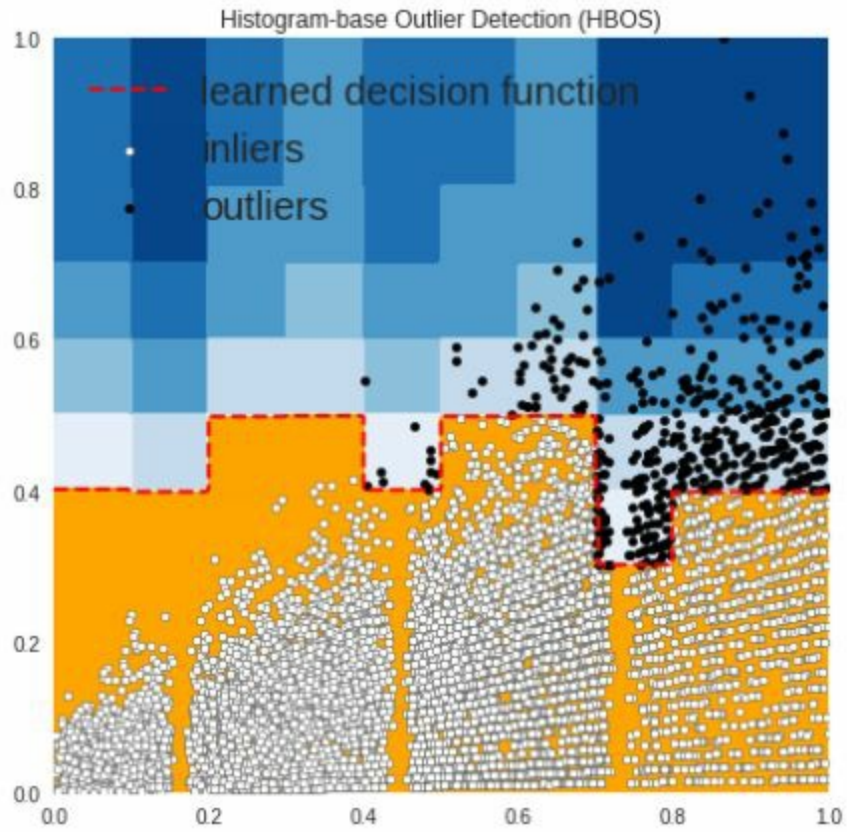
OUTLIERS : 427 INLIERS : 8096 Cluster-based Local Outlier Factor (CBLOF)



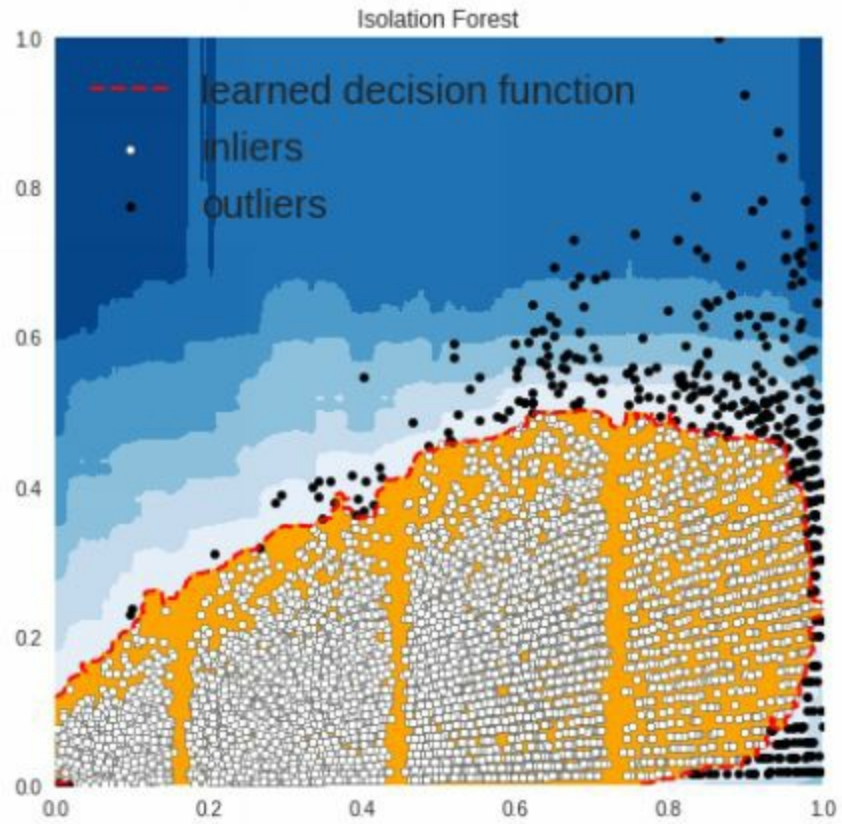
OUTLIERS : 386 INLIERS : 8137 Feature Bagging



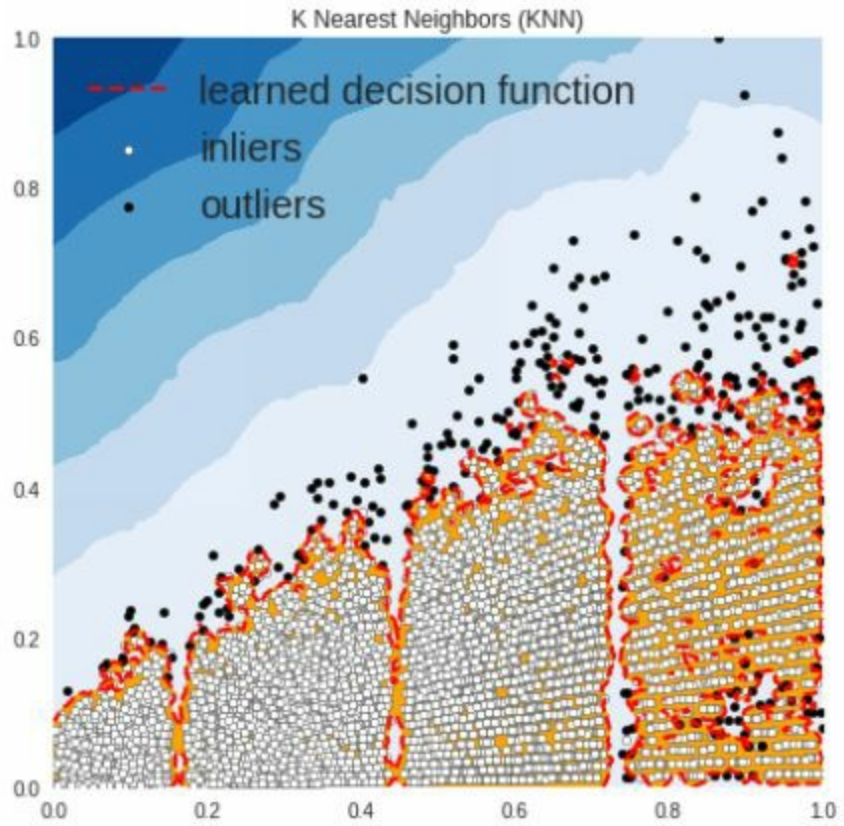
OUTLIERS : 501 INLIERS : 8022 Histogram-base Outlier Detection (HBOS)



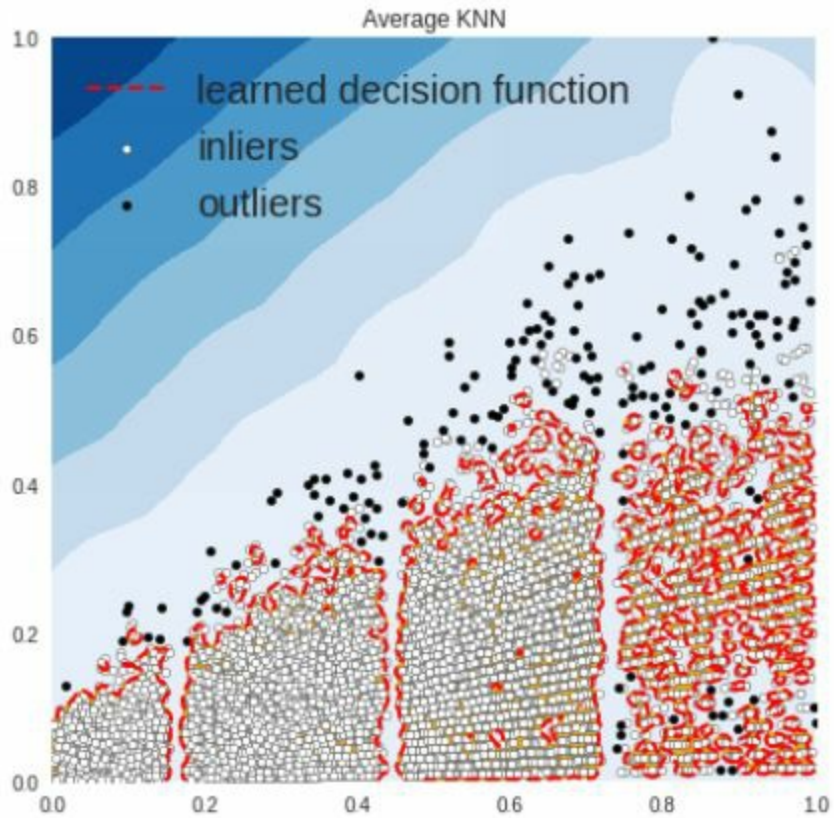
OUTLIERS : 427 INLIERS : 8096 Isolation Forest



OUTLIERS : 311 INLIERS : 8212 K Nearest Neighbors (KNN)



OUTLIERS : 176 INLIERS : 8347 Average KNN



In the above plot, the inliers are the white points that are enclosed by red lines and the outliers are the black points that are present in the blue area in the graph.

Chapter Twelve

An Introduction To Regression Analysis



As discussed in the second chapter, regression analysis is a very simple technique that most data analysts use to derive information from the data set. Let us look at another very simple linear regression model example to understand this very clearly. Regardless of the type of analysis you choose to perform, you should try to let the machine learn on its own. To reiterate the basic idea of any algorithm is that the system is exposed to a very large number of training data set and also shown a large sample of outputs expected from them. Based on these training data, the machine learns to figure out the relationship between the input and output and based on this learns to predict for new inputs give.

Below is a very primitive example to explain the same where the system needs to suggest whether the user needs to take an umbrella or not depending on the weather of the day. Let us say, the following table contains a sample set of training data.

Outside Temperature	Wear a umbrella
30°C	No
25°C	No

20°C		No	
15°C		Yes	
10°C		Yes	

+-----+-----+

As an average human being, our mind is trained to look at the input temperature and determine the output

- The decision to take an umbrella or not. Let us now try to model this decision making process into a

Algebraic equation so that a machine can also be trained to take such decision when given this data set.

For this, we will need the use of the trusted Python Library for machine learning implementations: ScikitLearn. And consider the following sample data set.

x1 x2 x3 y

1 2 3 14

4 5 6 32

11 12 13 74

21 22 23 134

5 5 5 30

Looking at the table one can infer the mathematical model or algebraic equation for getting the output $y =$

$$(x1 + 2*x2 + 3*x3).$$

To generate the training data set

```
from random import randint

TRAIN_SET_LIMIT = 1000

TRAIN_SET_COUNT = 100

TRAIN_INPUT = list()

TRAIN_OUTPUT = list()

for i in range(TRAIN_SET_COUNT):

    a = randint(0, TRAIN_SET_LIMIT)

    b = randint(0, TRAIN_SET_LIMIT)

    c = randint(0, TRAIN_SET_LIMIT)

    op = a + (2*b) + (3*c)

    TRAIN_INPUT.append([a, b, c])

    TRAIN_OUTPUT.append(op)
```

Train the model:

```
from sklearn.linear_model import LinearRegression

predictor = LinearRegression(n_jobs=-1)

predictor.fit(X=TRAIN_INPUT, y=TRAIN_OUTPUT)
```

Once the system is ready, pass a sample test data in the following format of a tuple [[10, 20, 30]] and

observe the output. This must be $10+20*2+30*2$ and output must be 140

```
X_TEST = [[10, 20, 30]]
```



```
outcome = predictor.predict(X=X_TEST)

coefficients = predictor.coef_

print('Outcome : {}\nCoefficients : {}'.format(outcome, coefficients))

Output

Outcome = [140]

Coefficients = [1.2.3]
```

We have successfully implemented a model, trained and seen in predicting the output for new input based on a mathematical linear equation.

Linear Regression Analysis

Regression is a simple data analysis model and in this model, when given an input, we get an output that is normally a numeric value, our interest here is not to learn the class but the function, the numeric function that best describes the data. The objective then is to use this model to generate estimations.

One of the most common and simplest models is the linear regression model. It is the most preferred model to find a predictive function when we are having a correlation coefficient that indicates a data predicts upcoming events.

The function is typically used to create a scatter plot of data based on the input given and usually create a straight line, this linear regression is also the preferred method tell the linear relation between two variables.

Like the common and simple mathematical formula to calculate the slope of a line

$$Y = mx + c$$

This is a common and largely popular simple algebraic equation, which can be used to explain the linear regression concept in machine learning as well. Basically we have a variable that is dependent, a function of another variable, and another variable that is an independent variable.

The objective is therefore to find this function that will help us determine how two given variables are related. In a dataset, we are normally given a list of values in a row and column format that can then be filled as the X and Y-axis values.

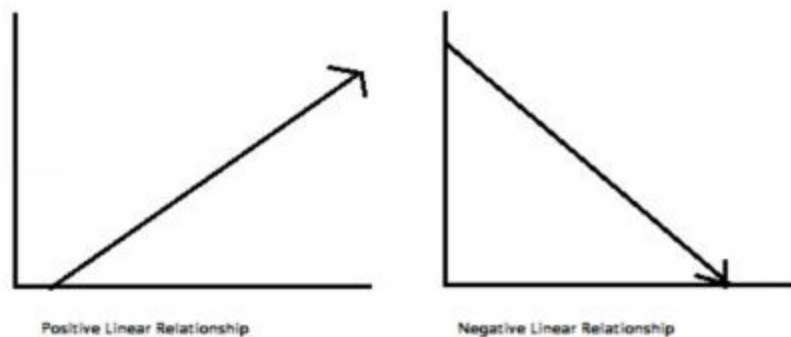


Image: Illustration to depict positive and negative linear regression

Linear regression or relationship is basically to observe that when one of more independent variables increases or decreases, the corresponding dependent variable also increases or decreases in tandem resulting in a slope as seen above.

As seen in the picture above, a linear relationship can thus be negative or positive.

Positive slope - when independent variable value goes up, the dependent variable also goes up and vice versa - almost like a direct proportion.

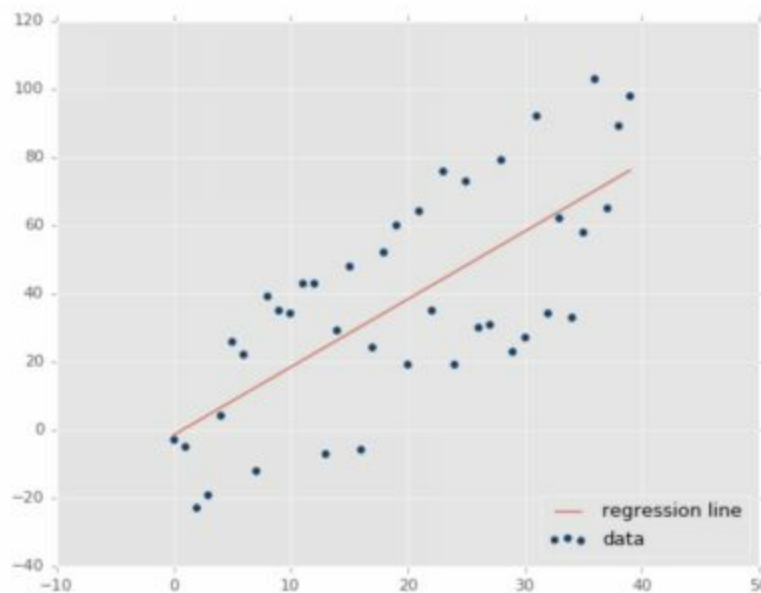
Negative Slope - this is the case when the independent variable's value goes up, the corresponding dependent variable value goes down, like in the case of

inverse proportion.

Now, let us go back to the algebraic equation of a simple slope to understand how to implement regression models in python.

We had seen that X and Y will have a relationship, however in real life this need not necessarily be true, in case of Simple Linear regression on SLR, we build a model that is based on data - the slope and the Y - axis derive from the data, also going further, we don't need the relationship between x and y to be exactly linear, it could include errors in data also called as residuals.

The objective is to simply take continuous data, find an equation that best fits the data and to be able to extrapolate and forecast or predict a specific value in the future. In the case of SLR or simple linear regression we are doing exactly that by creating a best fit line as shown in the scatter graph below



One of the popular applications of regression models is to predict stock prices or real estate market prices. There are several data sets available; we can pick the example of stock data set to see a sample python example.

For this one needs to install and import the quandl package.

The code to pull stock data set is as below, please create a python script and the following code and execute from terminal.

```
import pandas as pd  
import Quandl  
df = Quandl.get("WIKI/GOOGL")  
print(df.head())
```

Please note to check case of the library “quandl” or “Quandl” depending on the version of python you are using.

Sample Output

```
Open High Low Close Volume Ex-Dividend \  
Date  
2004-08-19 100.00 104.06 95.96 100.34 44659000 0  
2004-08-20 101.01 109.08 100.50 108.31 22834300 0  
2004-08-23 110.75 113.48 109.05 109.40 18256100 0  
2004-08-24 111.24 111.60 103.57 104.87 15247300 0  
2004-08-25 104.96 108.00 103.88 106.00 9188600 0
```

```
Split Ratio Adj. Open Adj. High Adj. Low Adj. Close \  
Date  
2004-08-19 1 50.000 52.03 47.980 50.170  
2004-08-20 1 50.505 54.54 50.250 54.155  
2004-08-23 1 55.375 56.74 54.525 54.700  
2004-08-24 1 55.620 55.80 51.785 52.435  
2004-08-25 1 52.480 54.00 51.940 53.000
```

```
Adj. Volume  
Date
```

2004-08-19 44659000

2004-08-20 22834300

2004-08-23 18256100

2004-08-24 15247300

2004-08-25 9188600

Is the sample data pulled in from the internet.

Remember the lesson from the previous chapter? Yes, the first step is scrub, clean and prepare the data set. One can notice, there are some redundancies and discrepancies in the pulled data set. The same can be rectified by adding the following line of code to the python script:

```
df = df[['Adj. Open', 'Adj. High', 'Adj. Low', 'Adj. Close', 'Adj.  
Volume']]
```

And if we were to apply a bit of common sense, one can understand that not all of this data is useful and the cleaned dataset can further be transformed for better result using the piece of code given below

```
df['HL_PCT'] = (df['Adj. High'] - df['Adj. Low']) / df['Adj. Close'] *  
100.0
```

And the following piece of code defines data frames and interprets the data output

```
df['PCT_change'] = (df['Adj. Close'] - df['Adj. Open']) / df['Adj.  
Open'] * 100.0  
  
df = df[['Adj. Close', 'HL_PCT', 'PCT_change', 'Adj. Volume']]  
print(df.head())
```

The output will look like this

```
Adj. Close HL_PCT PCT_change Adj. Volume
Date
2004-08-19 50.170 8.072553 0.340000 44659000
2004-08-20 54.155 7.921706 7.227007 22834300
2004-08-23 54.700 4.049360 -1.218962 18256100
2004-08-24 52.435 7.657099 -5.726357 15247300
2004-08-25 53.000 3.886792 0.990854 9188600
```

With this, we have our data set ready, which we will now have to convert to array format that will be understandable by the SciKit library, which we will be using to perform actual regression functions.

For us to proceed further, add the following lines of code to the python script file, these lines essentially import these libraries, which will be required for further functionalities.

```
import Quandl, math
import numpy as np
import pandas as pd
from sklearn import preprocessing, cross_validation, svm
from sklearn.linear_model import LinearRegression
```

At this point, the python script file must look something like this:

```
import Quandl, math
import numpy as np
import pandas as pd
from sklearn import preprocessing, cross_validation, svm
from sklearn.linear_model import LinearRegression
df = Quandl.get("WIKI/GOOGL")
df = df[['Adj. Open', 'Adj. High', 'Adj. Low', 'Adj. Close', 'Adj. Volume']]
```

```

df['HL_PCT'] = (df['Adj. High'] - df['Adj. Low']) / df['Adj. Close'] * 100.0
df['PCT_change'] = (df['Adj. Close'] - df['Adj. Open']) / df['Adj. Open'] *
100.0
df = df[['Adj. Close', 'HL_PCT', 'PCT_change', 'Adj. Volume']]
print(df.head())

```

Now, if we recollect, we are the stage to cross validate our cleaned and prepared data, for which we need to add the following lines, which will feed the data as a feature and label tuple to the classifier machine learning model. The feature can be defined as descriptive attributes and labels are the values that we are looking to predict with our machine learning models.

```

forecast_col = 'Adj. Close'
df.fillna(value=-99999, inplace=True)
forecast_out = int(math.ceil(0.01 * len(df)))

df['label'] = df[forecast_col].shift(-forecast_out)

```

With the above few lines of code we've defined what we want to forecast. The next steps are the train and test our model.

At this point we can use to dropna function and then proceed to converting the data to numpy array format, which is the expected data format by SciKit library functions that we will be subsequently using.

```

df.dropna(inplace=True)

X = np.array(df.drop(['label'], 1))
y = np.array(df['label'])

X = preprocessing.scale(X)

y = np.array(df['label'])

```

We have now created the label, array and preprocessed the dataset. We will now use the svm model and classifier model clf available in the SciKit toolkit to predict and print how robust the model is - the accuracy and reliability of it using the confidence functions.

```
X_train, X_test, y_train, y_test = cross_validation.train_test_split(X,
y, test_size=0.2)

clf = svm.SVR()

clf.fit(X_train, y_train)

confidence = clf.score(X_test, y_test)

print(confidence)
```

One can rerun the same script using the linear regression classifier instead of svm as follows

```
clf = LinearRegression()
```

The next steps to forecast and predict, for which the following lines of code need to be added to existing script

```
forecast_set = clf.predict(X_lately)

print(forecast_set, confidence, forecast_out)
```

The output:

```
[ 745.67829395 737.55633261 736.32921413 717.03929303 718.59047951
731.26376715 737.84381394 751.28161162 756.31775293 756.76751056
763.20185946 764.52651181 760.91320031 768.0072636 766.67038016
763.83749414 761.36173409 760.08514166 770.61581391 774.13939706
768.78733341 775.04458624 771.10782342 765.13955723 773.93369548
766.05507556 765.4984563 763.59630529 770.0057166 777.60915879]
```


0.956987938167 30

The next step is to import and Matplotlib to plot the scatter graph, which is beyond the scope of this book.

Thus linear regression is used in varied applications and domains ranging from economics to biology to predicting trendlines of oil prices, GDP, house prices, how much a country should spend on imports etc.

Correlation formulas can be used to predict how close to reality the prediction obtained from linear regression models is.

Let us look at another very simple linear regression model example to understand this very clearly.

To reiterate the basic idea of any machine learning algorithm is that the system is exposed to a very large number of training data set and also shown a large sample of outputs expected from them. Based on these training data, the machine learns to figure out the relationship between the input and output and based on this learns to predict for new inputs give.

Below is a very primitive example to explain the same where the system needs to suggest whether the user needs to take an umbrella or not depending on the weather of the day. Let us say, the following table contains a sample set of training data.

<i>Outside Temperature</i>	<i>Wear a umbrella</i>
30°C	No
25°C	No

| 20°C | No |

| 15°C | Yes |

| 10°C | Yes |

+-----+-----+

As an average human being, our mind is trained to look at the input temperature and determine the output - the decision to take an umbrella or not. So assume the temperature is 10°C, you might want to carry an umbrella expecting a snow storm or something. Let us now try to model this decision making process into an algebraic equation so that a machine can also be trained to take such a decision when given this data set.

For this, we will need the use of the trusted Python Library for machine learning implementations: sci kit learn. And consider the following sample data set.

x1	x2	x3	y
1	2	3	14
4	5	6	32
11	12	13	74
21	22	23	134
5	5	5	30

Looking at the table one can infer the mathematical model or algebraic equation to get the output $y = (x1 + 2*x2 + 3*x3)$.

To generate the training data set

```
from random import randint
TRAIN_SET_LIMIT = 1000
TRAIN_SET_COUNT = 100
TRAIN_INPUT = list()
TRAIN_OUTPUT = list()
for i in range(TRAIN_SET_COUNT):
    a = randint(0, TRAIN_SET_LIMIT)
    b = randint(0, TRAIN_SET_LIMIT)
    c = randint(0, TRAIN_SET_LIMIT)
    op = a + (2*b) + (3*c)
    TRAIN_INPUT.append([a, b, c])
    TRAIN_OUTPUT.append(op)
```

Train the model:

```
from sklearn.linear_model import LinearRegression
predictor = LinearRegression(n_jobs=-1)
predictor.fit(X=TRAIN_INPUT, y=TRAIN_OUTPUT)
```

Once the system is ready, pass a sample test data in the following format of a tuple [[10, 20, 30]] and observe the output. According to our algebraic equation this must be $10+20*2+30*2$ and output must be 140

```
X_TEST = [[10, 20, 30]]
outcome = predictor.predict(X=X_TEST)
coefficients = predictor.coef_
print('Outcome: {} \n Coefficients: {}'.format(outcome, coefficients))
```

Output

```
Outcome = [140]
```

Coefficients = [1.2.3]

We have successfully implemented a model, trained and seen in predicting the output for new input based on the mathematical linear equation.

Multiple Regression Analysis

In the previous section, we looked at regression modeling using simple linear regression where we considered a single predictor variable and a single response variable. The only interest that data miners have is on the relationship that exists between a set of predictor variables and a target variable. Most applications built for data mining have a lot of data, with some sets including thousands or millions of variables, of which most have a linear relationship with the response or target variable. That is where a data miner would prefer to use a multiple linear regression model. These models provide improved accuracy and precision of prediction and estimation, similar to the improved accuracy of regression estimates over bivariate or univariate estimates.

Multiple linear regression models use linear surfaces like hyperplanes or planes to determine the relationship between a set of predictor variables and one continuous target or response variable. Predictor variables are often continuous, but there could be categorical predictor variables included in the model through the use of dummy or indicator variables. In a simple linear regression model, a straight line of dimension one is used to estimate the relationship between one predictor and the response variable. If we were to evaluate the relationship between two predictor variables and one response variable, we would have to use a plane to estimate it because a plane is a linear surface in two dimensions.

Data miners need to guard against multicollinearity, a condition where some

of the predictor variables are correlated with each other. Multicollinearity leads to instability in the solution space, leading to possible incoherent results. For example, in a data set with severe multicollinearity, it is possible for the F-test for the overall regression to be significant, whereas none of the t-tests for the individual predictors are significant. This situation is analogous to enjoying the whole pizza while not enjoying any of the slices.

The high variability associated with the estimates for different regression coefficients means that different samples may produce coefficient estimates with widely different values. For example, one sample may provide a positive coefficient estimate for x_1 , whereas a second sample may produce a negative coefficient estimate. This situation is unacceptable when the analytic task calls for an explanation of the relationship between the response and the predictors individually. If there was a chance to avoid such instability when variables that are highly correlated are included, those variables tend to emphasize a particular component of the model being used because these elements are being counted twice. To avoid multicollinearity, the analyst should investigate the correlation structure among the predictor variables (ignoring the target variable for the moment).

Suppose that we did not check for the presence of correlation among our predictors but went ahead and performed the regression anyway. Is there some way that the regression results can warn us of the presence of multicollinearity? The answer is yes; we may ask for the variance inflation factors (VIFs) to be reported. Note that we need to standardize the variables involved in the composite to avoid the possibility that the greater variability of one of the variables will overwhelm that of the other variable.

Consider a dataset with p features (or independent variables) and one response (or dependent variable).

Also, the dataset contains n rows/observations. We define:

X (feature matrix) = a matrix of size $n \times p$ where x_{ij} denotes the values of j th feature for i th observation. So,

```
import matplotlib.pyplot as plt

import numpy as np

from sklearn import datasets, linear_model, metrics

# load the boston dataset

boston = datasets.load_boston(return_X_y=False)

# defining feature matrix(X) and response vector(y)

X = boston.data

y = boston.target

# splitting X and y into training and testing sets

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4,
random_state=1)

# create linear regression object

reg = linear_model.LinearRegression()

# train the model using the training sets

reg.fit(X_train, y_train)

# regression coefficients

print('Coefficients: \n', reg.coef_)
```

```
# variance score: 1 means perfect prediction

    print('Variance score: {}'.format(reg.score(X_test, y_test)))

# plot for residual error

## setting plot style

    plt.style.use('fivethirtyeight')

## plotting residual errors in training data

    plt.scatter(reg.predict(X_train), reg.predict(X_train) - y_train, color
                = "green", s = 10, label = 'Train data')

## plotting residual errors in test data

    plt.scatter(reg.predict(X_test), reg.predict(X_test) - y_test, color =
                "blue", s = 10, label = 'Test data')

## plotting line for zero residual error

    plt.hlines(y = 0, xmin = 0, xmax = 50, linewidth = 2)

## plotting legend

    plt.legend(loc = 'upper right')

## plot title

    plt.title("Residual errors")

## function to show plot

    plt.show()
```

Add the above code to a.py file and execute to script from the terminal, while having the downloaded dataset in the same working folder to see the desired output, including scattered plot generation.

Chapter Thirteen

Classification Algorithm



We will be looking at one of the most common classification algorithms – decision trees. Decision trees are one of the most ubiquitous and powerful classification algorithms available. This algorithm can be used for both continuous as well as categorical output variables. As we know classification algorithms are that category of algorithms used to predict the category of the given input data.

The objective is to create a model that predicts the value of a target based on simple decision rules that are inferred from the data features.

In simple words, it is very similar to the common -if.. then.. else.. conditional statement that is commonly used as part of programming languages. This is more like a flow chart and is like a branch based decision system. This algorithm is something that even a very average person can understand - something like looking at an incoming email and classifying it as personal or work or spam email based on certain pre defined rules can be given as a very simple use case of decision trees.

A decision tree is literally a tree where one can take the route of either of the branches based on the answer to the conditional question at each node, each branch represents a possible course action. Below is a simple decision tree example from real life, where the tree can be used to determine if a person is fit or not

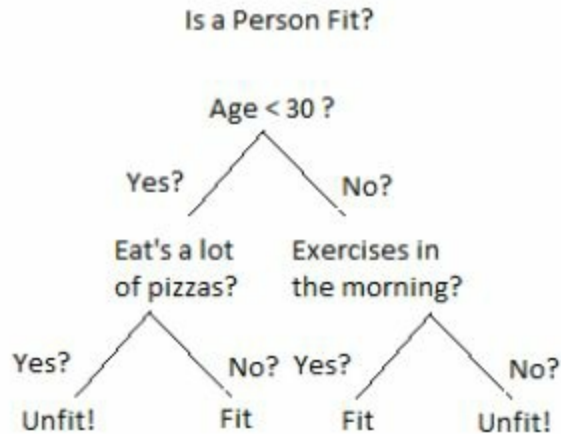


Image: A simple illustration of a Decision tree.

Advantages Of Decision Trees

Some key advantages of using decision trees are

Simple to understand and interpret and easy to visualize

This model requires little to no data preparation while other techniques require various steps such normalizing the input data set, dummy variables to be created, blank or null values have to be cleaned up and various such scrubbing and reducing activity which are normally done are not required. This significantly reduced the time and cost of execution of using this machine learning model. It is however important to remember, this model will require missing data to be filled up, else will thrown an error and not proceed with further computation.

The number of data points determines the cost of execution. The relationship of cost to number of data is logarithmic to train the data model.

The other unique strength of this data model is it works well with both numerical as well as categorical data while several other techniques are usually specially designed to work with one format of the data or the other.

Ability to handle multi output problems

It used the white box mode that is if a given solution is observable in a model, the explanation for the condition can be easily explained using a Boolean logic equation. Whereas in case of algorithms based on black box model – like a synthetic neural network and the outcome could be tough to understand.

The other advantage is the ability to evaluate the model by using statistical trials, thus making the model more reliable than the others.

This model also performs well if the assumption on data sets is slightly violated when applying to the actual model, thus ensuring flexibility and accurate results irrespective of variance.

Disadvantages Of Decision Trees

Some of the disadvantages of using a decision trees include:

1. Decision tree can create over complex trees that do not generalize data too well - basically the problem of overfitting. This can be overcome using techniques like pruning (literally like pruning the branch of tree, but is currently not supported in python libraries) - the task is to set up few samples needed at a leaf node or setting the highest depth a tree can grow to, thus limiting the problem of overfitting.
2. The trees become unstable because of small variations in the input data, resulting in an entirely different tree getting generated. This issue can be overcome by using a decision tree within an ensemble.
3. NP-Complete problem - a very common theoretical computer science problem can be a hindrance in the attempt to design the most optimal decision tree, because under several aspects, the optimality can

become affected. As a result, heuristic algorithms such as greedy algorithms where local optimal decisions at each node may become an issue. Teaching multiple trees to a collaborative learner can again reduce the effect of this issue and the feature and samples can be randomly sampled with replacements.

4. Concepts such as XOR, parity or multiplexer issues can be difficult to compute and express with decision trees.
5. In case of domination of classes, the tree learners end up creating biased learners. It is therefore important to balance the data set prior to fitting with the decision tree.
6. It is important to be conscious of the factor that decision trees tend to over fit the data with a large number of features and getting the right sample to number the features has to be taken care to not become too highly dimensional.
7. Decision tree is extensively used in designing intelligent home automation systems - say, if the current temperature, humidity and other factors are at a certain level, then control the home cooling system temperature accordingly type systems are largely designed based on decision trees. Things like an average human deciding to go out and play a game of golf can be a series of decisions and can be modeled around a decision tree.

There are two key factors

1. Entropy - measure of randomness or impurity of the sample set - this must be low!
2. Information Gain - also called as entropy reduction is the measure of how much the entropy has changed after splitting the dataset - the

value of this must be high.

Some key concepts one needs to learn when modeling a decision tree python are

Importing a csv file into the script using pandas,

```
from __future__ import print_function
import os
import subprocess
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier, export_graphviz
```

Visually Drawing the tree

```
def visualize_tree(tree, feature_names):
    """Create tree png using graphviz.
    Args
    ----
    tree -- SciKit-learn DecsisionTree.
    feature_names -- list of feature names.
    """
    with open("dt.dot", 'w') as f:
        export_graphviz(tree, out_file=f,
            feature_names=feature_names)
        command = ["dot", "-Tpng", "dt.dot", "-o", "dt.png"]
        try:
            subprocess.check_call(command)
        except:
            exit("Could not run dot, ie graphviz, to "
```

"produce visualization")

Using panda's library to prep the data set for the SciKit-learn decision tree code.

```
def get_iris_data():
    """Get the iris data, from local csv or pandas repo."""
    if os.path.exists("iris.csv"):
        print("-- iris.csv found locally")
        df = pd.read_csv("iris.csv", index_col=0)
    else:
        print(" downloading rom github")
        fn = "https://raw.githubusercontent.com/pydata/pandas/" + \
            "master/pandas/tests/data/iris.csv"
        try:
            df = pd.read_csv(fn)
        except:
            exit("Unable to download iris.csv")

    with open("iris.csv", 'w') as f:
        print("writing to local iris.csv file")
        df.to_csv(f)
    return df
```

Producing pseudocode that represents the tree.

```
def get_code(tree, feature_names, target_names,
            spacer_base=" "):
    """Produce psuedo-code for decision tree.
    Args
    ----
```

tree -- SciKit-learn DecisionTree.
feature_names -- list of feature names.
target_names -- list of target (class) names.
spacer_base -- used for spacing code (default: " ").

Notes

based on <http://stackoverflow.com/a/30104792>.

"""

```
left = tree.tree_.children_left
right = tree.tree_.children_right
threshold = tree.tree_.threshold
features = [feature_names[i] for i in tree.tree_.feature]
value = tree.tree_.value
def recurse(left, right, threshold, features, node, depth):
    spacer = spacer_base * depth
    if (threshold[node] != -2):
        print(spacer + "if (" + features[node] + " <= " + \
            str(threshold[node]) + ") {"")
        if left[node] != -1:
            recurse(left, right, threshold, features,
                left[node], depth+1)
        print(spacer + "}\n" + spacer + "else {"")
        if right[node] != -1:
            recurse(left, right, threshold, features,
                right[node], depth+1)
        print(spacer + "}")
    else:
        target = value[node]
```

```
for i, v in zip(np.nonzero(target)[1],
target[np.nonzero(target)]):
target_name = target_names[i]
target_count = int(v)
print(spacer + "return " + str(target_name) + \
" (" + str(target_count) + " examples)")
recurse(left, right, threshold, features, 0, 0)
```

Fitting the decision tree using sci kit learn library

```
y = df2["Target"]
X = df2[features]
dt = DecisionTreeClassifier(min_samples_split=20, random_state=99)
dt.fit(X, y)
```

A sample decision tree classifier program in python using the SciKit learn package.

```
>>> from sklearn import tree
>>> X = [[0, 0], [1, 1]]
>>> Y = [0, 1]
>>> clf = tree.DecisionTreeClassifier()
>>> clf = clf.fit(X, Y)

>>> clf.predict([[2., 2.]])
array([1])

>>> from sklearn.datasets import load_iris
>>> from sklearn import tree
>>> iris = load_iris()
>>> clf = tree.DecisionTreeClassifier()
>>> clf = clf.fit(iris.data, iris.target)
```

```
>>> import graphviz
>>> dot_data = tree.export_graphviz(clf, out_file=None)
>>> graph = graphviz.Source(dot_data)
>>> graph.render("iris")

>>> dot_data = tree.export_graphviz(clf, out_file=None,
... feature_names=iris.feature_names,
... class_names=iris.target_names,
... filled=True, rounded=True,
... special_characters=True)
>>> graph = graphviz.Source(dot_data)
>>> graph
```

k-Nearest neighbor algorithm - is another commonly used classification algorithm

Following is a sample python source code of k-nearest algorithm that is used as a classifying algorithm.

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
import warnings
from collections import Counter
#dont forget this
import pandas as pd
import random
style.use('fivethirtyeight')
```



```

def k_nearest_neighbors(data, predict, k=3):
    if len(data) >= k:
        warnings.warn('K is set to a value less than total voting groups!')
        distances = []
        for group in data:
            for features in data[group]:
                euclidean_distance = np.linalg.norm(np.array(features)-np.array(predict))
                distances.append([euclidean_distance,group])
        votes = [i[1] for i in sorted(distances)[:k]]
        vote_result = Counter(votes).most_common(1)[0][0]
        return vote_result

df = pd.read_csv('breast-cancer-wisconsin.data.txt')
df.replace('?',-99999, inplace=True)

df.drop(['id'], 1, inplace=True)
full_data = df.astype(float).values.tolist()

random.shuffle(full_data)

test_size = 0.2
train_set = {2:[], 4:[]}
test_set = {2:[], 4:[]}
train_data = full_data[:-int(test_size*len(full_data))]
test_data = full_data[-int(test_size*len(full_data)):]

for i in train_data:
    train_set[i[-1]].append(i[:-1])

```

```
for i in test_data:
    test_set[i[-1]].append(i[:-1])

correct = 0
total = 0

for group in test_set:
    for data in test_set[group]:
        vote = k_nearest_neighbors(train_set, data, k=5)
        if group == vote:
            correct += 1
            total += 1
print('Accuracy:', correct/total)
```

Chapter Fourteen

Clustering Algorithms

The last two chapters discussed some of the techniques that were covered in the second chapter. We looked at outlier detection, regression and classification analysis. This section throws some light on clustering analysis. We will also look at how to fit the right data to perform this analysis in Python.

K-Means Clustering Algorithm

This is one of the most popular algorithms; the k in the name refers to how many different unique clusters one wishes to generate from the given data set. Clustering is sometimes called unsupervised classification.

The value of k is defined by the user and each cluster formed has a centroid. The centroid is the central point of all the points that form any given cluster and the number of centroid is equal to the k - the number of clusters. This algorithm primarily works with numeric value and all other symbolic data type are excluded.

The key advantage and disadvantage of this clustering algorithm are as follows. The major advantage and reason for the algorithms popularity is how simple, easy it is understand and implement and how quick it is in its execution. The input data are automatically assigned into clusters and therefore very beginner friendly.

If we have to look at disadvantages or limitations of picking k means, it has

to be the fact that the k value has to be pre entered before the beginning of execution thus limiting dynamic adjustments if required.

Also the output can be significantly influenced by the seed data that is initially fed to the model. The algorithm tends to converge to local minima and therefore it is recommended to reset and rerun the algorithm with different random seeds to ensure minimal error.

It is not the most efficient when working with large data sets, and is therefore not scalable for large real time applications. Sampling can at times help to make the runs quicker on large data sets.

The other major disadvantage is how sensitive the algorithm is to outliers. A single outlier data could significantly alter the clustering efficiency of the model.

The mean value tends to get skewed if the input data set contains unusually large values (outliers), a suggested solution to address this issue is to use the 'median' value instead of the 'mean'

.K-Means Steps of the Algorithm

Randomize and choose the value of k, which is the number of centers of clusters.

Function to assign every data point to the closest cluster center. This is done by using methods such as using Euclidean distance to find the closest center of centroid in the clusters.

The position of centroids to be updated by taking the average after all the points have been assigned.

- Repeat steps 2 and 3 until arriving at a point of convergence - that is the point when the cluster assignment has reached its optimal

threshold it is point when subsequent reruns do not affect cluster positions much.

Now let us see how to implement the algorithm in python.

The following piece of code will help creating cluster blobs in python using the `make_blobs` function that is available in the sci kit learn library. The following piece of code will create random clusters on a graph.

```
# import statements
```

```
from sklearn.datasets import make_blobs
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
# create blobs
```

```
data = make_blobs(n_samples=200, n_features=2, centers=4,  
cluster_std=1.6, random_state=50)
```

```
# create np array for data points
```

```
points = data[0]
```

```
# create scatter plot
```

```
plt.scatter(data[0][:,0], data[0][:,1], c=data[1], cmap='viridis')
```

```
plt.xlim(-15,15)
```

```
plt.ylim(-15,15)
```

The above piece of code will create 4 different colored blobs on the scattered plot, two on top and two at the bottom, with each of these two slightly overlapping with each other.

Notice, how 4 clusters are created because we had defined our value of k to be 4 in the code. Next step is to measure euclidean distance, which can be achieved using the norm function in numpy package in python.

After that iteration, the new plot will show the cluster centroid in new updated positions. Repeat this iteration until optimal cluster position is reached.

```
# import K Means

    from sklearn.cluster import K Means

# import statements

    from sklearn.datasets import make_blobs

    import numpy as np

    import matplotlib.pyplot as plt

# create blobs

    data = make_blobs(n_samples=200, n_features=2, centers=4,
    cluster_std=1.6, random_state=50)

# create np array for data points

    points = data[0]

# create scatter plot

    plt.scatter(data[0][:,0], data[0][:,1], c=data[1], cmap='viridis')

    plt.xlim(-15,15)

    plt.ylim(-15,15)

# create kmeans object
```

```

    kmeans = KMeans(n_clusters=4)

# fit kmeans object to data

    kmeans.fit(points)

# print location of clusters learned by kmeans object

    print(kmeans.cluster_centers_)

# save new clusters for chart

    y_km = kmeans.fit_predict(points)

    plt.scatter(points[y_km ==0,0], points[y_km == 0,1], s=100,
    c='red')

    plt.scatter(points[y_km ==1,0], points[y_km == 1,1], s=100,
    c='black')

    plt.scatter(points[y_km ==2,0], points[y_km == 2,1], s=100,
    c='blue')

    plt.scatter(points[y_km ==3,0], points[y_km == 3,1], s=100,
    c='cyan')

```

The statement `from sklearn.cluster import K Mean` is a reference to the actual k means algorithm.

The k means algorithm is actually based on the Lloyds algorithm, which used to have clusters called as cells or voronoi cells. One of the key concepts to remember in clustering algorithms is to understand the concept of boundaries that define the range of data thereby helping to spot outliers.

After importing the sci kit learn library inbuilt k mean algorithm, next we simply plot a set of data using the following lines of code

```
x = [1, 5, 1.5, 8, 1, 9]
y = [2, 8, 1.8, 8, 0.6, 11]
plt.scatter(x,y)
plt.show()
```

Once the graph is plotted, the following set of lines can be written to convert the data into array - which is the acceptable input data structure format for scipy library.

```
X = np.array([[1, 2],
[5, 8],
[1.5, 1.8],
[8, 8],
[1, 0.6],
[9, 11]])
```

The next part of code is to initialize the k value and to in fact map the k means algorithm to k means type cluster only.

```
kmeans = KMeans(n_clusters=2)
kmeans.fit(X)
centroids = kmeans.cluster_centers_
labels = kmeans.labels_
print(centroids)
print(labels)
```

```
colors = ["g.", "r.", "c.", "y."]
for i in range(len(X)):
print("coordinate:", X[i], "label:", labels[i])
plt.plot(X[i][0], X[i][1], colors[labels[i]], markersize = 10)
plt.scatter(centroids[:, 0], centroids[:, 1], marker = "x", s=150, linewidths =
```



```
5, zorder = 10)
plt.show()
```

K-means ++ is the algorithm to implement if one needs to seed the model for better accuracy.

Code for Hierarchical Clustering Algorithm

This algorithm assumes each data point in the training data is a cluster and go on to form a cluster by finding two closest points and with subsequent iterations create what is called a dendrogram, which plots out each cluster and distance and which in turn used to determine the number of clusters. Thus the agglomerative hierarchical algorithm tends to produce more accurate results than k-means where the value of k is user defined.

```
# import hierarchical clustering libraries

import scipy.cluster.hierarchy as sch

from sklearn.cluster import AgglomerativeClustering

# create dendrogram

dendrogram = sch.dendrogram(sch.linkage(points, method='ward'))

# create clusters

hc = Agglomerative Clustering(n_clusters=4, affinity = 'euclidean',
linkage = 'ward')

# save clusters for chart

y_hc = hc.fit_predict(points)

plt.scatter(points[y_hc ==0,0], points[y_hc == 0,1], s=100, c='red')

plt.scatter(points[y_hc==1,0], points[y_hc == 1,1], s=100, c='black')
```

```
plt.scatter(points[y_hc == 2,0], points[y_hc == 2,1], s=100, c='blue')
```

```
plt.scatter(points[y_hc == 3,0], points[y_hc == 3,1], s=100, c='cyan')
```

You can use these algorithms with other algorithms to better analyze your data.

Conclusion

We cannot deny that data science is very important in today's world. It seems like only a matter of time when data science will be used for its application in almost all organizations and even administration of a nation for its productivity and efficiency. This makes it very important for data science professionals to keep up with the latest trends so that they are ready when data science is flourishing at an exponential rate. More knowledge means more opportunities and it will give you a competitive advantage over your colleagues.

We believe that data scientists are the backbone of an organization in this world full of data. Data scientists are supposed to extract, collect and analyze data and help organizations arrive at a better decision making process. Eventually the biggest goal of a data scientist is the growth of an organization. With insights provided by data scientists, organizations can implement better strategies and ultimately lead to customers having a better experience with the whole process.

Even though Data Science is in its infancy and just beginning to develop, it is evolving at an exponential rate demanding professionals from the industry who are multi-skilled possessing skills that are associated with statistics, computer science and business intelligence, all at the same time. It is evident that the demand for data scientists from the industry will shape the undergraduate students to be ready for this exploding and self-evolving discipline. It is almost known and acceptable now that data science studies

will become a staple curriculum of computer science courses.

Do let me know what you think of this book and whether it helped you understand and gain a deeper insight into what you were looking for. Stay on the lookout for my next book in this series where I'll be going into more detail on this topic! Meanwhile, do refer to the previous book in this series if there was some topic you weren't entirely clear about.

References

<https://www.simplilearn.com/data-science-vs-data-analytics-vs-machine-learning-article>

<https://www.analyticsinsight.net/five-ways-data-science-has-evolved/>

<https://www.dezyre.com/article/difference-between-data-analyst-and-data-scientist/332>

<https://www.scnsoft.com/blog/4-types-of-data-analytics>

<https://acadgild.com/blog/different-types-of-data-analytics>

<https://www.edvancer.in/common-types-data-science-techniques-must-know/>

<https://www.datapine.com/blog/data-visualization-techniques-concepts-and-methods/>

<https://visme.co/blog/examples-data-visualizations/>

<https://www.forbes.com/sites/bernardmarr/2017/07/20/the-7-best-data-visualization-tools-in-2017/#41ee204f6c30>

<https://mode.com/blog/python-data-visualization-libraries>

PYTHON DATA ANALYTICS

*The Expert's Guide
to Real-World Solutions*

Travis Booth

Introduction

Everyone talks about data today. You have probably come across the term “data” more times than you can remember in one day. Data as a concept is so wide. There is so much about data that we might never fully understand, at least not in our lifetime. One thing that is true about data is that it can be used to tell a story. The story could be anything from explaining an event to predicting the future.

Data is the future. Businesses, governments, organizations, criminals—everyone needs data for some reason. Entities are investing in different data approaches to help them understand their current situation, and use it to prepare for the unknown. The world of technology as we know it is evolving towards an open-source platform where people share ideas freely. This is seen as the first step towards decentralization of ideas, and eliminating unnecessary monopolies. Therefore, the data, tools, and techniques used in the analysis are easily available for anyone to interpret data sets and get relevant explanations.

There are many tools that can be used for data analysis. For this reason, the ultimate choice often becomes a challenge for most people. To set you on the right path, the first step is to decide which language you want to learn, then build from there. Beginner programmers struggle with this a lot, as had been explained in the earlier books in this series. However, as an expert data analyst, you have your path figured out already. That being said, there is no harm in learning something new. In the world of tech, you never know when

it will come in handy.

For most people, Python is currently one of the first programming languages they learn instead of the older languages like C. The dynamic shift in Python's popularity comes down to its intuitiveness, simplicity, and the fact that it is a high-level programming language. A high-level language means that it is as close to normal human languages as possible. From your experience with Python syntax and functions over the years, this is one of the things you probably appreciate about Python. More importantly, there is a budding community of developers, data scientists, and other experts who are constantly working to improve the Python language, and assist one another where necessary.

Python is used widely in several environments. However, our emphasis is on data analysis. Data scientists have come to appreciate Python more over the years, given its effectiveness in investigating and understanding big data. As a result, we have experts coming up with unique libraries specifically for data handling, which can be used in Python. These libraries come with amazing tools that help in data processing and analysis. Such is the development of data science that tech giants like Microsoft and Google are heavily invested in supporting the open-source projects and efforts in data science.

One of the most important concepts in data analysis that is a big deal in Python is simplicity. Python is a simple language, and this is one of the factors that set it apart from the other languages. Clarity in the definition is a common phenomenon. Other developers who come across your work don't need to struggle to understand it. This way, it is easier for them to implement it in whichever projects they are working on. Anyone reading your code should never have to struggle with comprehension.

Owing to the simplicity and ease of code flow in Python, the emphasis is

usually on how memory is consumed, other than how scripts perform their roles. As a result, this further makes data analysis easier. To use Python for data analysis, you need access to several tools specifically built for scientific, numerical, and visual computation and representation. After all, this is the crux of understanding data.

As an expert data analyst, your mastery of Python libraries will be useful from time to time. NumPy, for example, will be useful when working with or implementing linear algebra, working with vectors, random variables, and matrices. With Matplotlib, you can create and visualize data in different ways, making it easier for anyone using the data to understand it in-depth. Pandas offer reliable, fast, and easy to understand data structures that help in data manipulation and computations.

To make your work easier, IPython notebooks are an incredible Anaconda environment that allow you to work with Python code without struggling to write the code. The visuals used with the notebooks already have Python code, so you can see the results of your work instantly. All these are useful tools in data analysis that will help you along the way.

The thing about data analysis and data science is that it is an evolutionary field. Everything you learn manifests into something bigger and better. From the basics of data analysis, you can advance into machine learning. In fact, data analysis in Python helps you set the foundation for machine learning. Mastery of logistic and linear regression and learning how to use the Scikit-learn library in Python are the first steps towards advancing into machine learning and prediction science.

One of the most important lessons you learn from experience in using Python for data analysis is that analytics hardly ever exists in isolation. With this in mind, you will have to learn how to use other languages. The good thing is

that knowledge of Python is applicable in many programming environments.

Python is, and remains, the best choice for anyone who is fascinated by data. Anything from retrieval, scraping, processing, or data analysis is made easier through Python. It is an accessible language with so many tools that give you endless possibilities in terms of what you can do with data.

Chapter 1

Conceptual Approach to Data Analysis

Data is all around us. You interact with data at different times during the day. Everyone leaves behind fragments of data on the devices they use all the time. Individuals and companies need this data for decision making. How do these parties convert raw data into useful information that can help them make credible business decisions? This is where data analysis comes in. Data analysis is an elaborate process where the analyst uses statistical and analytical tools to make useful deductions from a given set of data. There are several analytical techniques that data analysts use for this purpose, including data visualization, business intelligence, and data mining.

Techniques Used in Data Analysis

As has been mentioned above and in the earlier books in this series, data analysis is an elaborate process. Below is an overview of some of the techniques you will come across in data analysis:

- ✓ **Data visualization**

Data visualization is about presentation. You are already aware of most of the tools that are used in data visualizations, such as pivot tables, pie charts, and other statistical tools. Other than presentability, data visualization makes large sets of data easy to understand. Instead of reading tables, for example, you can see the data transposed onto a color-coded pie chart.

We are visual creatures. Visual optics last longer in our minds than information we read. At a glance, you can understand what the information is

about. Summaries are faster and easier through data visualization than reading raw data. One of the strengths of data visualization is that it helps in speeding up the decision-making process.

- ✓ **Business intelligence**

Business intelligence is a process where data is converted into actionable information in accordance with the end user's strategic objectives. While most of the raw data might be difficult to understand or work with, through business intelligence, this data eventually makes sense. Business intelligence techniques help in determining trends, examining them and deducing useful insights.

Many companies use this to help in making decisions about their pricing and product placement strategies. This data is also helpful in identifying new markets for their products and services, and analyzing the sustainability of the said markets. In the long run, this information helps the company come up with specific strategies that help them thrive in each market segment.

- ✓ **Data mining**

Data mining involves studying large sets of data to determine the occurrence of patterns. Patterns help analysts identify trends, and make decisions based on their discoveries. Some of the methods used in data mining include machine learning, artificial intelligence, using databases and statistical computations.

The end result in data mining is the transformation of primitive raw data into credible information that can be used to make informed business decisions. Other than decision making, data mining can also help in finding out the existence and nature of dependency or abnormalities across different sets of data. It is also useful in cluster analysis, a procedure where the analyst studies a given set of data to identify the presence of specific data groups.

Data mining can be used alongside machine learning to help in understanding consumer behavior. Consumer tastes and preferences are traditionally dynamic. Because of this, changes take place randomly. Given the popularity of e-commerce today, the dynamic shift in consumer tastes and preferences is more volatile than ever.

Through data mining, analysts can collect lots of information about consumer actions on their websites, and make an accurate or near-accurate prediction of the purchase traits and frequencies. Such information is useful to marketing departments and other allied sectors in the business, to help them create appropriate promotional content to attract and retain more customers.

Marketing savvy experts usually create niches out of a larger market demographic. The same concept applies to data mining. Through data mining, it is possible to identify groups of data that were previously unidentified. Studying such data groups is important because it allows the analyst to experiment with undefined stimuli and in the process, probably discover new frontiers for the marketing departments.

Other than previously unidentified data, data mining is useful when dealing with data sets that are clearly defined. This also involves some element of machine learning. One of the best examples of this is the modern email system. Each mail provider has systems in place that determine spam and non-spam messages. They are then filtered to the right inboxes.

- ✓ **Text analysis**

Most people are unaware of text analysis, especially since it is often viewed as a sub-group of other data analysis methods. Text analysis is basically reading messages to determine useful information from the content available. Beyond reading texts, the information is processed and passed through specific algorithms to help in decision making.

The nature and process of text analysis depends on the organization and their needs assessment. Information is obtained from different databases or file systems and processed through linguistic analysts. From there, it is easier to determine patterns in the information available, by looking at the frequencies of specific keywords. Pattern recognition algorithms usually look for specific targets like email addresses, street names, geographical locations, or phone numbers.

Text analysis is commonly applied in marketing, when companies crawl the websites of their competitors to understand how they run their business. They look for specific target words to help them understand why the competitor is performing better or worse than they are. This method can deliver competitor keywords and phrases, which the analyst can use to deduce a counter-mechanism for their company.

Data Analysis Procedure

The data analysis methods discussed above might be different in their approaches, but the end result is almost always the same. Their core objective is to support decision-making in the organization at different levels. The following are some of the steps that you will follow during data analysis:

- ✓ **Define the objectives**

The objectives behind your study must be clearly outlined. This is the foundation of your study. Everything that you do from here onwards depends on how clearly the objectives of your study were stated. Objectives guide you on how to proceed, the kind of data to look for, and what the data will be used for.

- ✓ **Ask the right questions**

In order to meet the objectives outlined in the first step, you must seek

answers to specific questions. This narrows down your focus to the things that matter, instead of going on a wild goose chase with data. Remember that by the time you collect data, the procedure in place should be effective so that you do not end up with a lot of worthless data.

- ✓ **Collect data**

Set up appropriate data collection points. Make sure you use the best statistical method or data collection approach to help you get the correct data for your analysis. You can collect data in different forms, especially for raw data. Once you have the data you need, the hard work begins. Sift the data to weed out inaccurate or irrelevant entries. Use appropriate tools to import and analyze data.

- ✓ **Analyze data**

In this stage, you aggregate and clean data into the different tools you use. From here, you can study the data to determine and define patterns and trends. This is also the stage where most if not all of your questions are answered. You will conduct “what if” analysis in this stage.

- ✓ **Interpretation and predictive analysis**

Having obtained the necessary information from your analysis, the final stage is to infer conclusions from the data. A predictive analysis involves making informed decisions based on the data you have, and leveraging it against some other supporting information. The data from your analysis might be quantitative. To make a correct decision, for example, you have to consider some qualitative elements, too. You might have the prerequisite numbers, but the general feeling in the market about your business is unfavorable. Making predictions, therefore, is not just about relying on the data you collect and analyze, but an aggregate of other decision processes that are not directly related to the data.

In this stage, you will also look back to the objectives outlined earlier on. Does the data you collect sufficiently answer the questions posed earlier? Suppose there are some objections, do you feel the data available can help you convincingly challenge the objections? Is there something you intentionally ignored, or a limitation to your conclusions? What happens if you introduce an alien factor into the question? Does it affect the output? If so, how?

Methods Used in Data Analysis

Everyone has access to overwhelming data today. What matters is the use to which you put your data. Data analysts are exposed to lots of data from time to time. The challenge is sifting through voluminous data to interpret the ramifications. There are several tools and methods that are used, especially in statistical data analysis.

In a world where big data is coming full circle, there are several tools that can help you reduce your workload, while at the same time improving your efficiency and reliability of the data you use. The methods discussed herein are the foundation of data analysis. Once you master them, it is easier to graduate into sophisticated methods and techniques:

- ✓ **Standard deviation**

Standard deviation is an expression of how far data spreads from the arithmetic mean. Standard deviation in data analysis is about data point dispersion from the mean. A high value shows a large spread from the mean, while a low value means that most of the data in use is close to the mean.

Always use standard deviation alongside other techniques to derive conclusive results from your study. Without this, especially with data sets that contain many outliers, standard deviation is not a good value

determinant.

- ✓ **Averages**

This refers to the arithmetic mean. You arrive at this by dividing the sum of (n) items on your list by the number of (n) items on the list. Averages help you understand the general trend in a specific data set. Calculating averages is very easy, and from this information, you can tell so much about a given data set at a glance.

Even as you use averages, you must be careful not to use them in isolation. Independent of other methods, averages can be misconstrued for the same information available from median and mode. If you are working with data that has a skewed distribution, averages are not the best option because you don't get information accurate enough to support your decision-making needs.

- ✓ **Regression analysis**

Regression analysis is about identifying the relationship between different variables. From these relationships, you will then establish the dependency between the variables. This analysis helps you identify whether relationships between variables are weak or strong.

Regression analysis is usually a good option when you need to forecast decision making. Since they consider the relationship between dependent and independent variables, you can look at many variables that affect your business in one way or the other. The dependent variable in your study refers to the variable you need to understand. The independent variables are endless, and could represent any factors you are looking at, which might affect the dependent variable in some way.

- ✓ **Hypothesis testing**

This method is also referred to as *t* testing. In hypothesis testing, the goal is to test a given assertion to determine whether it is true or not for your study population. This method is popular in so many areas that are reliant on data, like economics and scientific and business research purposes.

There are several errors that you must be aware of if your hypothesis study is to be a success. One common error in hypothesis testing is the Hawthorne effect, also known as the observer effect. In this case, the results of the study do not reflect the true picture because the participants are aware they are under observation. As a result, the results are often skewed and unreliable.

Hypothesis testing helps you make decisions after comparing data against hypothetical scenarios concerning your operations. From these decisions, you can tell how some changes will affect your operation. It is about the correlation between variables.

✓ **Determining sample sizes**

You need to learn how to select the right sample size for your studies. It is not feasible to collect information from everyone in the study area. Careful selection of your sample size should help you conduct the study effectively.

One of the challenges you might experience when choosing the sample size is accuracy. While you are not going to study the entire population of interest, your sample must be randomly selected in a manner that will allow you to get accurate results, without bias.

Types of Data Analysis

Before you delve into data analytics, you should understand the basic concepts you will perform from time to time. Different terminologies are used in data analytics depending on the type of analytics. There is so much data that can be extracted from different sources today. Understanding raw

data is quite a challenge given the unpredictable nature of some forms of data. This is where data analysis comes in. Analysis deals with refining raw data into an understandable and actionable form. Here are some of the types of data analysis you will encounter:

- ✓ **Descriptive Analysis**

Descriptive analysis is about summaries. From the data available, you should be able to find summary answers to pertinent issues in the organization, events, or activities. Some of the tools you will use in descriptive analysis include generated narratives, pie charts, bar charts, and line graphs. At a glance, someone should get a summary of the information you present before them.

- ✓ **Diagnostic Analysis**

Think about diagnostic analysis in the same way you see a doctor to provide a diagnosis about your health. More often, you are only aware of the symptoms you are feeling. It is up to the doctor to run tests and rule out possibilities, then narrow down a list of possibilities and tell you what you are suffering from.

In a diagnostic analysis, the goal is to use data to explain the unknown. Assuming you are looking at your marketing campaigns on social media, for example, there are so many things you can look at, from mentions, to reviews, to the number of followers and likes. These are features that indicate some activity about your brand. However, it is only through a diagnostic analysis that you can go deeper and unearth what the numbers mean in as far as engagement goes.

- ✓ **Predictive Analysis**

Predictive analysis is one of the common types of analysis in use in organizations today. It uses a combination of statistical algorithms and

machine learning to understand data and use this to extrapolate future possibilities from historical data. For accurate predictions, the historical data must be accurate, or the predictions might be flawed.

Predictive analysis is entirely about planning for the future. You use present and historical data to determine what might happen in the future, especially when you alter a few variables that you can control. These studies focus on creating predictive models for new data.

- ✓ **Exploratory Analysis**

Exploratory analysis is about determining trends in your data, and from there explaining some features that you might not have been able to determine through other analytical methods. The emphasis is on identifying outliers to understand why and where they occur, and the variables that are affected by the outliers in as far as decision-making is concerned.

- ✓ **Prescriptive Analysis**

Many of the forms of analysis you use will give you a general view of your data. A general analysis cannot give you the kind of information you need. Prescriptive analysis is about precision. The answers you get from this analysis are specific. It is like getting prescription medicine – the doctor recommends specific drugs, which should be taken under specific instructions.

Assuming you are looking at data about recent road accidents, through prescriptive analysis, you can narrow it down to accidents as a result of drunk driving, poor road signage, roadworthiness of the vehicles, or careless driving.

Tools Used in Data Analysis

There are several tools you need to learn about to help you in your career as a

data analyst. At the basic level, you should at least have a working knowledge of web development, SQL, math, and Microsoft Excel. It also follows that you should be good at PHP, HTML, JavaScript, and know how to work with basic programming commands, libraries, and syntaxes.

As an advanced user, you should also be adept in the following fields:

- ✓ **R Programming**

One of the challenges many data analysts experience is choosing the right programming language. Essentially, it is wise to learn as many languages as you can, because you never know what the next project you work on will demand. You might not fully understand all the programming languages, but having working knowledge is a great idea.

While there are lots of programming languages you can choose from, R programming is one that any data analyst should master. It is preferred because it is unique and versatile, particularly when dealing with statistical data. Since R is an open-source platform, you have access to several data analysts who can help you.

R is a simple, yet articulately developed program. In R programming, you will use recursive functions, loops, conditionals, and support for I/O features. R also has storage features, which is good for data handling as you proceed with your tasks. You will also find the GUI effective, which is ideal for data display.

- ✓ **Python**

The basics of Python programming have been discussed in the earlier books in this series. However, we can recap by highlighting the power behind this open-source programming language. Python is simple, yet it packs quite the punch in as far as other programming languages are concerned.

Programmers and developers alike enjoy coding in Python because of the wide library support, which helps you in data management, manipulation, and analysis. It is one of the easiest languages to learn, especially if you have experience with other languages. The list of projects you can build in Python is endless, especially because there are many new projects that are still being built today, which we are yet to experience. In terms of existing projects that were built through Python, think about YouTube.

- ✓ **Database management**

You will be working with lots of data, so data management is a skill you should master or polish up. Some of the tools you must learn include MySQL, MongoDB, MS Access, and SQL Server. These tools are mandatory for data collection, processing, and storage. More importantly, you should understand how to use commands like *order by*, *having*, *group by*, *where*, *from*, and *select*.

- ✓ **MatLab**

MatLab is another simple, flexible, and powerful programming language that is necessary for data analysis. Through MatLab, you can manipulate and analyze data using the native libraries. Given that the MatLab syntax is almost similar to C++ and C, prior knowledge of these programming languages will help you progress faster in MatLab.

Over the years, the use of data analysis has become important in different environments. Companies and organizations use data to gain insight into their business performance by studying how their customers interact with their brands at different data collection points. Having understood the basics of data analysis, let's move on to data analysis with one of the most amazing programming languages, Python.

Benefits of Data Analysis in Python over Excel

At this point in time, you must have tried data analysis with different tools and applications. For most analysts, you start with Excel then advance into Python and other languages.

In the business world, Microsoft Excel is one of the most important programs, especially when it comes to collecting data. You can use it for data analysis, but there are challenges you might experience, which necessitates the move to Python programming for data analysis.

While Excel is a great tool, it has some unique challenges that you can overcome by learning Python. A bit of Python programming could really change your life and make data analysis easier for you in data science.

- ✓ **Expert data handling**

One of the first things you will enjoy in Python that sets it apart from Excel and other basic data analysis tools is the administrative privileges you enjoy when handling data. This is everything from importing data to manipulation.

You can upload any data file in Python, something that you cannot enjoy in Excel. There are some data formats that you generally cannot read or functionally work with in Excel, which impedes your ability to go about your work. This becomes a problem in many situations. You can also come across data files that are unreadable, but can still work. Python generally allows you more control over data handling. Therefore, you can easily scrape data from different databases and proceed to analyze it and draw conclusions.

Granted that you can still perform a lot of tasks on your data in Excel, you might have some restrictions. These are not there in Python. You can carry out all manner of manipulation on the data you use. Think about recording, merging, and even cleaning data. Through Python libraries like Pandas, you can view and clean some data to ensure it is suitable for the purpose you

intended the analysis. To do this in Excel, you would have to spend more time than necessary, and probably never get it done properly. Therefore, other than the value in terms of utility, Python also offers you the benefit of time consciousness.

- ✓ **Automated data management**

Excel is an awesome program. Microsoft has spent years developing Excel into an amazing tool for data management. This we can see in the GUI. It is an easy tool for anyone to use, especially someone who lacks programming knowledge. However, in data analysis, you need to go beyond the ordinary if you are to get the best results.

More often Excel will be useful up until the moment you need to automate some processes. This is where your problems begin. Other than process automation, it is also not easy to perform an analytical process across different Excel sheets or repeat a process several times.

Programming in Python takes away these problems. Assuming you need to execute some code to analyze recurrent data, you only need to write a script that would import the new data whenever it is available, parse it, and deliver an analytical report on time. On the other hand, in Excel, you would have to manually create a new file, then key in the desired formulas and functions before proceeding with the analysis.

More importantly, in Excel you would save the data format only in the supported Excel formats. However, in Python you can save the output file in whichever database file format works for you. This means you do not have to spend more time on file conversion which in most cases interferes with the outcome.

- ✓ **Economies of scale**

Spare some time and study the organization of data in Excel. One feature that strikes out clearly is that data is organized in tabs and sheets. This is a prominent feature in Excel, and it works well for processes that are completely reliant on Excel. However, the problem comes in when you have a gigantic database to work with. You might be looking at Excel data sheets with lots of entries per sheet, or a database that has too many Excel sheets.

Processing such database files will take a lot of time. This creates unnecessary lag in data analysis. Many are the times when your machine will crash, unable to process Excel sheets as fast as you need them to. In such a scenario, your only solution is to be patient and process the files one at a time.

This is a challenge that you don't have to worry about in programming. Languages like Python were specifically built to mitigate such issues. You can process large files in Python faster and more efficiently than you would in Excel. Besides, it is highly unlikely that your device will give up on you as it would when processing datasets in Excel.

- ✓ **Ability to regenerate data**

In your role as a data analyst, you will need to explain your work to more people than you can imagine. Once you are done with the analysis, you might be asked to prepare a report on your findings, which another department will use to meet their objectives. Beyond that, you might also be required to present the outcome in person, and explain to a panel the decisions you make, and your recommendations.

To meet the objectives outlined above, your data must be reproducible. People who were not part of the analytical process should be able to access the data and understand it just as you do. Here's where the problem arises when using Excel. First of all, it is generally impossible for you to provide an

elaborate illustration of the procedure and processes leading up to your recommendations. The only way you can walk anyone through your analysis is to get the original file and take them through each step.

Given the haste in which you might have done your work, this might be a challenge. Programming in Python, on the other hand, makes your work easier if you ever need to share it with someone. In some cases, all you need to do is press the OK or Enter button and the analysis will be executed as many times as you need it to. Besides, when analyzing data in Python, you can easily explain each step and have your audience follow through, executing code and seeing the results immediately.

✓ **Debugging**

If you are analyzing data in Excel, you will have a difficult time identifying errors. In fact, you have to manually look for the errors. Given a dataset with thousands of cells, this could prove to be a problem. Debugging in Excel is therefore a challenge any data analyst would not wish to deal with.

Programming languages like Python make debugging a lot easier. By design, if you enter the wrong syntax you get an error message instead of the expected output. Another good reason for analyzing data in Python is because you can trace the errors in each step. Whenever you key in the wrong functions or syntax, the program will return an error, prompting you to check and sort it out.

In Excel you would probably not know whether you have an error or not, and figuring out the genesis of the problem might force you to start from the beginning, which is more than you could have bargained for.

Since you can include comments in your code, it is easier to trace problems and sort them out. Even if you are not working with data you prepared, you

can still read the comments and understand what another programmer did.

At the same time, this should not be taken as an assertion that you will fix all the errors you encounter right away. Some errors might take you longer to identify and solve. However, the fact remains that analyzing data in Python gives you an easier and better chance at debugging errors than in Excel.

- ✓ **Open-source programming**

Everything about Excel is in the hands and control of Microsoft. If the program is buggy, you must depend on Microsoft to release patches for bugs. Feature support is also a challenge because unless Microsoft updates their releases, you will have to contend with what is available.

One of the perks of programming in Python is that you are free to enjoy the benefits of open-source programming. You have access to a large community of programmers who are always willing to assist you with any concerns.

As you work with some Python code for data analysis, you can improve any of the functions by altering the code accordingly, and share it with the rest of the Python community. There are so many developers who have created or updated some of the packages they use, in the process improving the functionality of the programming language. This has also resulted in better visualizations.

- ✓ **Advanced operation support**

When using Excel, you will struggle when it comes to machine learning and the associated features. This is because Excel was not built for these functionalities. You need advanced programming languages to help you in this regard, hence the need for Python.

In Python, you should also be able to build unique machine learning models. These can be integrated into your code through some of the popular Python

frameworks like TensorFlow and Scikit-Learn, thereby enhancing your capabilities when analyzing data.

- ✓ **Data visualization**

You need to see what you are working on. Visualization serves different purposes in data analysis. From the perspective of the analyst, the moment you come across some data, you should easily guess the kind of plot you will use for it. Someone might quip in at this juncture that Excel does offer visualization features. Well, that might be true, but visualization in Excel can be very limited.

Python offers you so much more in visualization, especially when you need advanced visualizations. In a business environment, you are called upon to make presentations all the time. Your presentation should be attention-grabbing if it is to convince someone to come onboard.

Each time you are tasked with presenting your report before a panel, remember that most of the people you engage might have no knowledge of data analysis. Therefore, it is impossible for them to read statistical data with the same precision you would. The best way of assisting such individuals would be by plotting some amazing visualizations. A good plot should be one that the audience can make sense of without straining, even if they have no knowledge of statistical computations or data analytics.

It is important to mention that this should not mean you abandon Excel altogether. Excel as a Microsoft Suite has its unique features that will come in handy in data handling and management. However, when compared against Python and other programming languages, it still has a long way to go in terms of data analysis. Perhaps one of the perks of Excel is that you can manually enter data into your database. This comes down to the GUI. If you are working with a small set of data, you can still scan through it instantly

through Excel.

Generally, Excel is ideal for the basic data analyst. As you advance in the field, however, you should think outside the box. Advance into Python programming so you can learn to perform better, accurate, and complex data analysis without the encumbrances of Excel.

Possible Shortcomings of Analyzing Data in Python

It is common knowledge that a lot of people use Python for programming and data analysis. Python is one of the easiest languages to learn, and since you do not need to write very long lines of code, many people love it for making programming easier. One of the highlights of Python programming is the easily readable syntax which makes programming simpler compared to other languages like C. Python further boasts a dynamic library which programmers can use comfortably to perform a lot of tasks like managing system interfaces and handling string operations without necessarily having to write more code.

The mention of data analysis has many people excited about their data and the work that is done to it. Today all companies that you engage from time to time will need access to some of your data. This enables them to understand your behavior and use that knowledge to improve their service delivery to you. Beyond the profit motive, businesses need to ensure you are happy and satisfied. From data analysis, they are able to determine what influences your purchase decisions, and how to appeal to your needs better.

Given all the buzz about data and data analysis, it might come as a surprise to a lot of people, but data analysis does have unique challenges that are impeding the expected deliverables. One of the biggest challenges that data analysts have to work through is the fact that most of the data they rely on are

user-level based. Because of this reason, there is room for a lot of errors which eventually affects the credibility of the data and reports obtained therefrom.

Whether in marketing or any other department in the business that relies on data, the unpredictability of user-level data means some data will be relevant to some applications and projects, but not all the time. This brings about the challenge of using and discarding data, or alternatively keeping the data and updating it over time.

While Python offers these benefits, it is also important to be aware of some of the challenges and limitations you might experience when programming in Python. This way, you know what you are getting into, and more importantly, you come prepared. Below we will discuss some of the challenges that arise for data analysts when they have to work with this kind of data.

- ✓ **Input bias**

One of a data analyst's biggest concerns revolves around the reliability of the data at their behest. Most of the data they have access to, especially at the data collection points like online ads from the company, are not 100% reliable. Contrary to what is expected, this input does not usually present the true picture of events concerning the interaction between customers and the brand.

Today there are several ways data analysts can try to obtain credible and accurate information about customers. One of these is through cookies that can be tracked. Cookies might present some data, but the accuracy of the data will always come under scrutiny.

Think about a common scenario where you have different devices, each of which you can use to go online and check some information about your

favorite brand. From this example, it is not easy to determine the point at which the sale was made. All the devices belong to you, but you could have made a purchase decision from one of them, but used another to proceed with the purchase. This level of fragmentation makes it difficult to effectively track customer data. It is likely that data obtained from customers who own different devices will not be accurate. Because of this reason, there is always the risk of using inaccurate data.

- ✓ **Speed**

By now, it is no secret that Python is relatively slower compared to majority of the programming languages. Take C++ for example, which executes code faster than Python. Because of this reason, you might need to supplement the speed of your applications. Many developers introduce a custom runtime for their applications which is more efficient than the conventional Python runtime.

In data analysis, speed is something that you cannot take for granted, especially if you are working with a lot of time-sensitive data. Awareness of the speed challenges you might encounter in Python programming should help you plan your work accordingly, and set realistic deliverables.

- ✓ **Version compatibility**

If there is a mundane challenge that you will experience in Python it is version compatibility. Many programmers consider this a mundane issue, but the ramifications are extensive. For beginner data analysts, one of the challenges is settling on the right Python version to learn. It is not an easy experience, especially when you know there is something better already.

By default, programmers consider Python version 2 as the base version. In case you need to advance to futuristic data analysis, Python version 3 is your best bet. Generally, you will receive updates to either of the versions

whenever they are available. However, when it comes to computations and executing code, some challenges might arise. A lot of programmers and data analysts still prefer the second version over the current one. This is because some of the common libraries and framework packages only support the second version.

- ✓ **Porting applications**

For a high-level programming language, you must use interpreters to help you convert written code into instructions that the operating system can understand better. In order to do this, you will often need to install the correct interpreter version in your computer to help in handling the applications. This can be a problem where you are unable to port one application to a different platform. Even if you do, the porting process hardly ever goes smoothly.

- ✓ **Lack of independence**

For all the good that can be done with Python, it is not an independent programming language. Python depends on third-party libraries, packages, and frameworks to enable you to analyze data accordingly.

Other programming languages that are available in the market today come with most of the features bundled in already, unlike Python. Any programmer interested in analyzing data in Python must make peace with the fact that they will have to use additional libraries and frameworks. This comes with unique challenges, because the only way out is to bring in open-source dependencies. Without that, legacy dependencies would consume a lot of resources, increasing the cost of the analysis project.

- ✓ **Algorithm-based analysis**

There are two acceptable methods that data analysts use to study and interpret a given set of data. The first method is to analyze a sample population, and draw conclusive remarks from the assessment of the sample about the

population. Given that the approach covers only a sample, it is possible that the data might not be a true representation of what the greater population is about. Samples can easily be biased, which makes it difficult to get the true version of events.

The second approach is to use algorithms to understand information about the population. Running algorithms is a better method because you can study an entire population based on the algorithm syntax. However, algorithms do not always provide the most important contextual answers.

Either of the methods above will easily present actionable recommendations. However, they cannot give you answers to why customers behave a certain way. Data without this contextual approach can be unreliable because it could mean any of a number of possibilities. For the average user, reports from algorithms will hardly answer their most pressing questions.

- ✓ **Runtime errors**

One of the reasons why Python is popular is because of its dynamism. This is a language that is as close to normal human syntax as possible. As a result, you will not necessarily have to define a variable before you call it in your code. You will, therefore, write code without struggling as you would in other languages like C#.

However, even as you enjoy easy coding in Python, you might come across some errors when compiling your code. The problem here arises because Python does not have stringent rules for defining variables. With this in mind, you must run a series of tests whenever you are coding to identify errors and fix them at runtime. This is a process that will cost you a lot of time and financial resources.

- ✓ **Outlier risks**

In data analysis, you will come across outliers from time to time. Outliers will have you questioning the credibility of your data, especially when you are using raw user data. If a single outlier can cast doubt on the viability of the dataset, imagine the effect of several outliers.

More often you will come across instances where you have weird outliers. It is not easy to interpret them. For example, you might be looking at data about your website, only to realize that for some reason, there was a spike in views during a two-hour period, which cannot be explained. If something like this happens, how can you tell whether the spike represents individual hits on your website or whether it was just one user whose system was perhaps hacked, or experienced an unprecedented traffic spike?

Certainly, such data will affect your results if you use them in your analysis. The important question is, how do you incorporate this data into your analysis? How can you tell the cause behind the spike in traffic? Is this a random event, or is it a trend you have observed over time? You might have to clean the data to ensure you capture correct results. However, what if by cleaning the data, you assume that the spike was an erroneous outlier, when in real sense the data you are ignoring was legitimate?

✓ **Data transfer restrictions**

Data is at the heart of everything today. For this reason, it is imperative that companies do all they can to protect the data at their disposal. If you factor in the stringent data protection laws like the GDPR, protecting databases is a core concern in any organization.

Data analysts might need to share data or discuss some data with their peers, but it is impossible to do this. Access to specific data must be protected. It is therefore impossible to share data across servers or from one device to another. If you delve further into big data, most organizations do not have

employees with the prerequisite skills to handle such data efficiently. As a result, data administrators must restrict the number of people who can interact with such data.

In light of these restrictions, most of the work done in analysis and recommendations thereof is the prerogative of the data analyst. There is hardly ever a second opinion because very few people have access to the database or data with similar rights. This also creates a problem where users or members of the organization are unable to provide a follow-up opinion on the data. They do not know the procedures or assumptions the analyst used to arrive at their conclusions. In the long run, data can only be validated by one or a few people in the organization who took part in the analysis. This kills the collaborative approach where it should have been allowed to thrive.

Chapter 2

Data Analysis in Python

Why should you use Python for data analysis? A lot of people use Microsoft Excel for analysis. This is one of the rudimentary analytical tools you can use today. Python offers more than what you can get from Excel. Python is an easy language to learn, and since its introduction in 1991, it has become one of the most prolific programming languages worldwide.

Python has enjoyed an amazing library support, with Pandas standing tall among the main libraries. Python does not just offer a means of data analysis, but over and above what you can do with Excel, you can manipulate and clean data better. For applications that rely on data, Python is the best option, especially since it is a strong multiple purpose language.

Python Libraries for Data Analysis

Before you proceed, you must choose the ideal development environment you will work on. Most people choose from the following three:

- iPython notebook
- IDLE
- Terminal

While the choice of development environment is entirely based on your preferences, most developers choose iPython because of the amazing features built into it, which make your work easier. Through iPython, you can execute

your code in blocks instead of running each line individually during testing. Before you get deep into data analysis, it is important to highlight some of the important points in Python. Most of these were discussed in-depth in the earlier books in this series, so this is a recap to refresh your memory.

Lists in Python are enclosed in square brackets, with each item on the list separated from the next by a comma. For example, below is a list of square numbers:

```
squares_list = [0, 1, 4, 9, 16, 25, 36]
```

```
squares_list
```

Output

```
[0, 1, 4, 9, 16, 25, 36]
```

Strings in Python are always defined with inverted commas. For example, below is a string:

```
1 = "Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua."
```

```
print(1)
```

Output

```
Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt
```

ut labore et dolore magna aliqua.

Lists and strings are very important in Python. Most of the work you do in data analysis will involve them, hence a reminder is always great. Assume you are asked to perform a mathematical operation or create graphs from a given data set in Python. You would have to write code specifically to address each of the tasks you are given. This can be a challenge for most people, and you might even lose enthusiasm over Python in a short while.

Instead of going through all that, Python has unique libraries with predefined instructions and functions which you can import into your development environment and solve the tasks handed to you. Python libraries are a lifesaver.

You have been introduced to the fundamentals of Python programming in the earlier books in this series. At this juncture, we will focus on the Python libraries used in data analysis. To help you learn faster, we will still enforce some of the key concepts learned in the earlier books where necessary.

Among other reasons, data scientists prefer Python over most programming languages because it is easy to learn and open source. It is also a high-performance language, which makes work easier for developers when working on object-oriented projects. Perhaps the standout reason why Python is quite popular is the large endowment of libraries. Each library is unique, yet extensive enough to enable programmers to solve many data problems every day.

The following are some of the top libraries used in data science:

- ✓ **NumPy**

For numerical computations, you need Numerical Python (NumPy). NumPy is considered the foundation of numerical computations in Python. It is a

general-purpose array processor that uses N-dimensional array objects.

NumPy is an efficient library given that when using multidimensional arrays, you have operators and functions that work with multidimensional arrays, thereby eliminating the slowness challenge during numerical computations. NumPy functions are precompiled, helping you complete numerical routines faster than other libraries.

Through NumPy's approach, you can perform computations faster and efficiently, especially when using vectors. NumPy is a mainstay in data analysis when you need powerful N-dimensional arrays. Libraries like Scikit-learn and SciPy have NumPy as their foundation, and you can also use NumPy in place of MATLAB if you are working with Matplotlib and SciPy.

✓ **TensorFlow**

If you are working on a high-performance computation project, TensorFlow is your best bet. There are thousands of contributors working on this library, which is a good resource pool whenever you are struggling with something.

Through TensorFlow, data scientists are able to define and run computations with tensors. A tensor is a computational object that can be manipulated to derive values. In this library, you can expect high-quality graphical visualizations, which makes it easier for you to present projects to an audience.

In neural machine learning, TensorFlow is preferred by developers because it helps them reduce errors in computations by up to 60%. This further allows them to perform parallel computing. Through parallel computing, developers can then build complex projects and execute them in a fairly simple manner.

Another benefit of using the TensorFlow library is that it enjoys support from Google. This partnership comes in handy especially in library management,

as the tech giant allows a seamless support framework when using the library. Besides that, you will always have some of the latest features when using TensorFlow because the development team behind it release updates frequently, and you can install them faster than most libraries.

Given all these benefits, you will find TensorFlow coming in handy when working on video detection projects, time series analysis, text applications, and image or speech recognition projects.

✓ **Matplotlib**

For data visualizations, Matplotlib provides some of the most amazing results in data science. It is by far the best plotting library you will use in Python. Matplotlib is essentially a data visualization library, hence the wide range of plots and graphs. To extend its utility further, Matplotlib also comes with an object-oriented API through which you can add the visualizations created into different apps.

If you have been working with MATLAB in the past, Matplotlib is a better alternative. Being an open-source library, usage is free, and you have access to a large pool of experts who can assist you in so many ways.

When using Matplotlib, you are not restricted in terms of the operating system. You can work with lots of output types and backends, thereby allowing you to create visualizations in any format you desire.

Perhaps one of the best things about using Matplotlib is its behavior in use. It is very easy on memory consumption compared to other libraries. Because of the efficient memory consumption, you should expect a smooth experience at runtime, too.

Matplotlib visualizations are useful when analyzing the correlation between different variables. It presents each variable in a unique way, making it easier

to spot the similarities and differences between them. You can also use it to detect outliers in a scatter plot or identify uniqueness in data distribution, helping you get a better insight into the data you are studying.

- ✓ **Pandas**

Python Data Analysis (Pandas) is another important library that you cannot miss in data science. Together with Matplotlib and NumPy, this library comes in handy, especially for cleaning data. Data structures in Pandas are flexible and efficient, allowing you an intuitive and easy way to program structured data.

Concerning the need to clean or wrangle data, Pandas comes second to none. Many data analysts store data in CSV files and other database files. Pandas has exceptional support especially for CSV files, allowing you to access data frames and perform transformations like extract, transform, and load on the data sets in question.

The Pandas syntax is elaborate with incredible functions to enable you to produce amazing results even if your data set is missing some fragments of data. Through Pandas, you can build unique functions and test them on different sets of data.

Pandas helps data scientists in many commercial, financial, and academic fields, especially when dealing with statistical data analysis. It is also a good library for financial computations and has recently been introduced into neuroscience.

- ✓ **SciPy**

For high-level computations in data science, you need Scientific Python (SciPy). It is an open-source library with thousands of members in the contributor community. SciPy is an extension of NumPy, therefore you can

expect the same efficiency in NumPy when you are working on technical and scientific computations. It makes the scientific calculation more user-friendly due to the fact that its functions and algorithms are an extension of NumPy.

You will find SciPy easier to work with if you are ever working on differential problems because its functions are built into the library. This, coupled with the ndimage submodule helps in processing multidimensional images faster. The high speed is another reason why SciPy is a reliable library for data visualization and manipulation.

Where is SciPy applicable? As a data scientist, you will need SciPy if your work involves linear algebra, working with optimization algorithms, Fourier transform or any differential equations, and operations that involve multidimensional images.

These are the main libraries you will use for data.

Most operating systems already have Python installed. However, it is wise to cross-check to ensure you have the correct version.

Installation Guide for Windows

If you are working on a Windows machine, installation should be straightforward. Windows allows you to download an installer package, and an installation wizard will help you until the final step. In the following example, we install NumPy. The procedure is the same for the other Python libraries, too.

Step 1:

Go online and download the Windows installer package that suits your setup.

Some library links are as follows:

- NumPy <https://pypi.org/project/numpy/>
- SciPy <https://scipy.org/scipylib/>
- Matplotlib <https://matplotlib.org>
- IPython <https://ipython.org>

Once you have the installer package in your device (*installer.exe*), double click on it and follow the installer wizard prompts. In case you already have Python installed, the wizard will detect this and advise you accordingly.

Installation Guide for Linux

The Linux installation method depends on the kind of Linux distribution you are running. Most of the Linux distributions have NumPy preinstalled.

Installing IPython

This installation assumes you are using version 6.0 and above. For a quick installation, enter the following if you have already installed pip:

```
$ pip install ipython
```

This will install IPython and any of the dependencies that you will need going forward. The easiest way to install IPython and the majority of the dependencies is to use pip. To use IPython effectively, you will need to install some other packages, too. To ensure that you install the correct packages, it is wise to use conda or pip.

While you can install IPython on its own without other dependencies, it is not advisable because the process is lengthy and you could encounter several issues that might affect your productivity. This is why it is best to use the Python package manager, pip, as shown:

```
$ pip install ipython
```

Before you install any Python packages, always make sure you have the correct version and are running Python from the command line. To check the version available, run the code below:

```
python --version
```

The output from the code above should be something like this:

```
Python 3.6.0
```

In case you do not have Python installed, visit www.python.org to find the latest version.

Building Python Libraries from Source

You can install Python libraries from source, especially if you need to use the latest version available. This is usually a straightforward process, but you might have some challenges unique to your operating system.

The following steps will guide you on how to retrieve NumPy from GitHub:

```
$ git clone git://github.com/numpy/numpy.git numpy
```

You will use the same commands for the other libraries, too.

Run the following command to unpack the installer package:

```
$ tar -xzf ipython.tar.gz
```

The following command will install the library to your preferred destination:

```
$ python setup.py build
```

```
$ sudo python setup.py install --prefix=/usr/local
```

In case you are using pip, you can install the directories through the following commands:

```
$ pip install numpy
```

```
$ pip install scipy
```

```
$ pip install matplotlib
```

```
$ pip install ipython
```

If you are using `setuptools`, you can install the directories through the following commands:

```
$ easy_install numpy
```

```
$ easy_install scipy
```

```
$ easy_install matplotlib
```

```
$ easy_install ipython
```

If you do not have administrator rights to the device you are using, prepend *sudo* to the commands outlined above, so that you can install the libraries with super-user rights.

Chapter 3

Statistics in Python - NumPy

In Python, there are lots of libraries you will come across. One of these is NumPy. For data analysis, your understanding of NumPy will help in scientific computation. Knowledge of this library is a fundamental step in data analysis mastery. Once you understand NumPy, you can then build on to other libraries like Pandas.

Once you learn the basics of NumPy, you can then advance into data analytics, using linear algebra and other statistical approaches to analyze data. These are two of the most important mathematical aspects that any data analyst should know about. During data analysis, you will often be required to make predictions based on some raw data at your disposal. For example, you might be asked to present the standard deviation or arithmetic mean of some data for analysis.

In linear algebra, the emphasis is on using linear equations to solve problems through NumPy and SciPy. Mastery of the NumPy basics will help you build on the knowledge you have gained over the years, and perform complex operations in Python.

In NumPy, one of the things you should remember is file I/O. All the data you access is retrieved from files. Therefore, it is important that you learn the basic read and write operations to the said files. In the example below, we will generate an identity matrix and save the contents to a file.

One of the benefits of using the NumPy library is that you are always aware

that all the items contained in any array share the same type. Because of this reason, you can easily determine the size of storage needed for the array.

Your Python distribution should have NumPy as a basic bundle. However, in case you don't have it installed, you can install it using the following commands:

If you are running a Linux system:

```
sudo apt-get install python-numpy
```

If you are running a Windows system, you must have Anaconda running:

```
conda install numpy
```

Once you have it installed, import the NumPy package into a new Python session as follows:

```
>>> import numpy as np
```

As you work on NumPy, you will realize that most of the work you do is built around the N-dimensional array, commonly identified as *ndarray*. The *ndarray* refers to a multidimensional array which could hold as many items as defined. The *ndarray* is also homogenous, meaning that all the items that are present in the array are of the same size and type.

Each object within the array is also defined by its unique data type, (*dtype*). With this in mind, each *ndarray* is always linked with one *dtype*.

Each array holds a given number of items. The items are available in different dimensions. The dimensions and items within the array define the *shape* of the array. These dimensions are referred to as the *axes* and as they compound, they form a *rank*.

When starting a new array, use the *array()* function to introduce all the

elements in a Python list as shown below:

```
>>> x = np.array([5, 7, 9])  
  
>>> x  
  
array([5, 7, 9])
```

To determine whether the object you just created is indeed an *ndarray*, you can introduce the *type()* function as shown below:

```
>>> type(x)  
  
<type 'numpy.ndarray'>
```

The *dtype* created might be associated with the *ndarray*. To identify this data type, you introduce the following function:

```
>>> x.dtype  
  
dtype('int32')
```

The array above only has one axis. As a result, its rank is 1. The shape of the array above is (3,1). How do you determine these values from the array? We introduce the attribute *ndim* to give us the number of axes, the *size* to tell us the length of the array, and finally the *shape* attribute to determine the shape of the array as shown below:

```
>>> x.ndim  
  
1  
  
>>> x.size  
  
3  
  
>>> x.shape
```


(3L,)

In the examples we have extrapolated above, we have been working with an array in one dimension. As you proceed in data analysis, you will come across arrays that have more than one dimension. Let's use an example where you have two dimensions below to explain this further.

```
>>> y = np.array([[12.3, 22.4],[20.3, 24.1]])
```

```
>>> y.dtype
```

```
dtype('float64')
```

```
>>> y.ndim
```

```
2
```

```
>>> y.size
```

```
4
```

```
>>> y.shape
```

```
(2L, 2L)
```

This array contains two axes, hence its rank is 2. The length of each of the axes is 2. The *itemsizes* attribute is commonly used in arrays to tell us the size of every item within the array in bytes as shown in the example below:

```
>>> y.itemsize
```

```
8
```

```
>>> y.data
```

```
<read-write buffer for 0x0000000003D44DF0, size 32, offset 0 at  
0x0000000003D5FEA0>
```

Generating Arrays

There are different ways of creating arrays. The examples above illustrate the simplest, by creating a sequence or a list in the form of an argument with the `array()` function. Below is an example:

```
>>> x = np.array([[5, 7, 9],[6, 8, 10]])  
  
>>> x  
  
array([[5, 7, 9],  
       [6, 8, 10]])
```

Other than the lists created, you can also create one or more tuples in the same manner as shown below using the `array()` function:

```
>>> x = np.array(((5, 7, 9),(6, 8, 10)))  
  
>>> x  
  
array([[5, 7, 9],  
       [6, 8, 10]])
```

Alternatively, you can also use the same procedure to create more than one tuple as shown below:

```
>>> x = np.array([(1, 4, 9), [2, 4, 6], (3, 6, 9)])  
  
>>> x  
  
array([[1, 4, 9],  
       [2, 4, 6],  
       [3, 6, 9]])
```

As you work with `ndarrays`, you will come across different types of data.

Generally, you will be dealing with numerical values a lot, especially float and integer values. However, the NumPy library is built to support more than those two. The following are other data types that you will use in NumPy:

- `bool_`
- `int_`
- `intc`, `intp`, `int8`, `int16`
- `uint8`, `uint16`, `uint32`, `uint64`
- `float_`, `float16`, `float32`, `float64`
- `complex64`, `complex128`

Each of the NumPy numerical types mentioned above has a unique function used to call its value as shown below:

Input

```
float64(52)
```

Output

```
52.0
```

Input

```
int8(52.0)
```

Output

```
52
```

Input

```
bool(52)
```

Output

```
True
```

Input

```
bool(0)
```

Output

False

Input

```
bool(52.0)
```

Output

True

Input

```
float(True)
```

Output

1.0

Input

```
float(False)
```

Output

0.0

Some of the functions might need a data type to complete the argument as shown below:

Input:

```
arrange (6, dtype=uint16)
```

Output:

```
array ([0, 1, 2, 3, 4, 5], dtype=uint16)
```

Before you create a multidimensional array, you must know how to create a

vector as shown below:

```
a = arange(4)
```

```
a.dtype
```

Output

```
dtype('int64')
```

```
a
```

Output

```
array([0, 1, 2, 3])
```

```
a.shape
```

Output

```
(4,)
```

The vector outlined above has only four components. The value of the components is between 0 and 3.

To create a multidimensional array, you must know the shape of the array as shown below:

```
x = array([arange(2), arange(2)])
```

```
x
```

Output

```
array([[0, 1],
```

```
[0, 1]])
```

To determine the shape of the array, use the following function:

```
x.shape
```

```
Output
```

```
(2, 2)
```

The *arrange()* function has been used to build a 2 x 2 array.

You will come across situations where you need to choose only one aspect of an array and ignore the rest. Before you begin, create a 2 x 2 matrix as shown below:

```
a = array([[10,20],[30,40]])
```

```
a
```

```
Output
```

```
array([[10, 20],  
       [30, 40]])
```

From the array above, we are going to select an item. Keep in mind that the index numbers in NumPy always start from 0.

```
Input: a (0, 0)
```

```
Output:
```

```
10
```

```
Input: a (0, 10)
```

```
Output
```

```
20
```

```
Input: a (10, 0)
```

Output

30

Input: a (10, 10)

Output

40

From the example above, you can see how easy it is to select specific elements from an array. Given an array a, as above, we have the notation a(x, y) where x and y represent the indices of each object within the array, a.

From time to time you might come across character codes. It is important to know the data types associated with them as follows:

Character code	Data type
b	bool
d	double precision float
D	complex
f	single precision float
i	integer
S	string
u	unsigned integer
U	unicode
V	void

For example, a single precision floats array can be identified as shown below:

Input:

```
arrange (5, dtype='f')
```

Output:

```
array ([0, 1, 2, 3, 4], dtype=f;pat32)
```

Slicing and Indexing

You learned about slicing standard Python lists in the previous books in this series. The same knowledge applies when slicing one-dimensional NumPy arrays. You will also learn how to flatten arrays. Flattening arrays simply means converting a multidimensional array into a one-dimensional array.

The *ravel()* function can manipulate the shape of an array as follows:

Input

```
b
```

Output

```
array([[[ 0, 1, 2, 3],  
       [ 4, 5, 6, 7]])
```

Input

```
b.ravel ()
```

Output

```
array([ 0, 1, 2, 3, 4, 5, 6, 7])
```

The *flatten()* function performs the same task as *ravel()*. However, the difference is that in the *flatten* function, the array is allocated new memory.

It is possible to set the shape of a tuple without using the *reshape()* function.

This is done as follows:

Input

```
b.shape = (3,4)
```

Input

```
b
```

Output

```
array
```

```
(([ 0, 1, 2, 3],
```

```
[ 4, 5, 6, 7],
```

```
[ 8, 9, 10, 11])
```

Transposition is a common procedure in linear algebra where you convert the rows into columns and columns into rows. Using the example above, we will have the following output:

Input

```
b.transpose = ()
```

Output:

```
array
```

```
(([ 0, 4, 8],
```

```
[ 1, 5, 9],
```

```
[ 2, 6, 10],
```

```
[ 3, 7, 11])
```

It is possible to stack an array by the depth, vertical alignment, or horizontal alignment. For this purpose, you will use the following functions:

- `hstack()`
- `dstack()`
- `vstack()`

For a horizontal stack, the *ndarray* tuple is as shown below:

Input:

```
hstack((a, b))
```

For a vertical stack, the *ndarray* tuple is as shown below:

Input:

```
vstack((a, b))
```

For a depth stack, the *ndarray* tuple is as shown below:

Input:

```
dstack((a, b))
```

Importance of NumPy Mastery

As a data analyst, you will come across a lot of packages that can help you go about your work. NumPy is one such package that will be useful in data analysis. There are several reasons why this open-source Python library is an important package you should master. The following are some of the top reasons why learning about NumPy will help you going forward:

- ✓ **Operation speed**

You might not know about this, but NumPy is written in one of the oldest

programming languages, C. One of the properties you benefit from is that it can execute faster than other packages. This makes a lot of sense when you think about Python as a whole being a dynamic language that needs interpretation. Before interpretation, Python code has to be converted to bytes. A compiled C code will definitely perform faster than the average Python code.

There are specific Python versions that are faster than others. For example, programs written in Python 2 are relatively faster than those written in Python 3. The efficiency is between 5 and 14%, so most people will never notice the performance lag, unless you are very keen.

NumPy arrays are stored in blocks of the same type and size. Because of this reason, they are easier to access and execute where necessary. On the other hand, Python uses lists for most tasks. A single list could contain different types of objects, and as a result, rendering a Python code is relatively slower than C loops, hence NumPy is a very fast package.

- ✓ **Support for other libraries**

One of the reasons why NumPy is an important language to learn is because it supports most of the Python libraries. Through NumPy, you can use Pandas, SciPy, SymPy and many others. In fact, SciPy and NumPy pretty much work hand in hand.

In NumPy, you should also be able to perform lots of linear algebra functions. This is an important part of data analysis, which also hinges on SciPy. Most of the time, you will need to install NumPy and SciPy together to enhance your performance in data analysis or scientific computing.

- ✓ **Matrix computations**

Through the *ndarray* functions, you can perform a lot of computations

involving matrices in NumPy. There are so many matrix computations that you can perform through this package, including raising matrices to specific powers and deriving the product of two matrices.

A lot of the work required in data analysis involves algebraic equations and computations. Performing these in NumPy makes your work easier and enhances your ability to deliver the best outcome.

- ✓ **Functional package**

If there is one reason why using NumPy will be a good idea for you, it is the fact that it supports many functions. Most of the functions built to support different packages are already built into NumPy, so you don't need to download them independently.

From math computations, to linear algebra, indices, random samples, statistics and polynomials, you will never run out of supporting options when working in NumPy. This further enhances your ability to analyze different types of data and draw conclusive remarks from them.

- ✓ **Universal support**

NumPy uses universal functions, referred to as *ufuncs*. These are functions that apply to each element in an array input. Owing to their universal nature, the outcome in the output array is stored in the same file size as the input.

Beyond this, you will also find the array broadcasting feature coming in handy, especially when working with different arrays. By default, arrays are available in unique sizes and shapes, and they can all be used within the same function. Because of the universality of NumPy, your system will automatically adjust the shapes to ensure they match the shape and size of the largest array in your code.

NumPy is one of the first Python libraries you should master. Knowledge of NumPy will help you advance into other libraries like SciPy which are equally important, and will form a great part of your data analysis journey.

Chapter 4

Data Manipulation in Pandas

Pandas is one of the most important Python packages you should know about, especially if you are a data analyst or data scientist. It offers amazing visualization tools that will not just help you get the attention of your audience, but will also help them understand your work faster. There are several uses of Pandas that you will come across in data analysis and beyond.

Through this library, you will learn how to analyze, transform and clean data, and present it in a manner that makes sense to your audience. Most people have data stored in Excel files. You can import this data to Pandas and convert it automatically into data frames. Data frames are simply tables, but with more privileges than the regular Excel tables.

From the data frames, you can perform statistical calculations and get answers to important questions about the data, like correlation analysis, media, max and min estimations for each column, or determine the distribution patterns for your data.

Many times, you come across data that is so jumbled up you need to spend more time cleaning it before you can make sense of it. Pandas allows you to clean such data by using specific criteria to filter the data, eliminating inaccurate data or missing values from your final data.

Beyond this, you can also use different features in Pandas to visualize your data and have your audience appreciate the appeal. You can do this through plot lines, bubbles, histograms, and bars from Matplotlib.

The data you are working on will always be useful in the future. For this reason, Pandas allows you to save the data once it has been cleaned and processed into an Excel sheet, or any other file system or database you prefer.

Pandas is not just an important library for data analysis. It is part of many other libraries that you will use from time to time. Knowledge of Pandas will help you in working with NumPy, performing statistical analytics in SciPy, working with machine learning algorithms in Scikit-learn, and using plotting functions in Matplotlib.

Before you get started with Pandas, you must have a working knowledge of Python. You do not necessarily need to be an expert at Python, but some credible knowledge will help you, especially about the basics like iterations, functions, dictionaries, and lists. Other than the fundamentals of Python, you should also learn a bit about NumPy because it shares a lot of similarities with Pandas.

Installing Pandas

Installing this library is straightforward. You will use your command line for Windows users, or Terminal if you are using a Mac as follows:

For Macs:

```
pip install pandas
```

For Windows:

```
conda install pandas
```

In case you are using a Jupyter notebook, you can install Pandas as follows:

```
!pip install pandas
```

Why is it important to use (!) in the notebook? It instructs your system to run

the code as if you were using terminal or command line.

Fundamentals of Pandas

There are two important components of Pandas: *DataFrames* and *Series*. Series refers to a column of data, while a group of Series constitutes a DataFrame.

Below is a series example for Toyota vehicles:

	Toyota
0	3
1	4
2	1
3	5

Below is a series example for BMW vehicles:

	BMW
0	4
1	5
2	2
3	9

Below is a DataFrame for the two series examples above:

	Toyota	BMW
0	3	4
1	4	5

2	1	2
3	5	9

From the information above, we can deduce that Series and DataFrames share a lot of similarities. For this reason, most of the operations that you can perform with one of them can be performed on the other, too.

Building DataFrames

Learning how to build unique DataFrames in Python is a fundamental skill that will help you when testing functions and new methods in Pandas. There are several ways of creating DataFrames. The best method is always the simplest: using *dict*.

Using the data above, you can use Pandas to determine the department sales in a car dealership. You need to create a column for each model and a row for customer purchases. In order to have this organized as a Pandas dictionary, you will have the following code:

```
data = {  
    'Toyota': [3, 4, 1, 5],  
    'BMW': [4, 5, 2, 9]  
}
```

The information above is then passed to the DataFrame constructor in Pandas as follows:

```
sales = pd.DataFrame (data)  
sales
```

Output

	Toyota	BMW
0	3	4
1	4	5
2	1	2
3	5	9

How do we arrive at this output? Every data item (*key, value*) represents a column within the DataFrame. When you create the DataFrame, the index is determined as 0-3. Indices are determined when you create the DataFrame. Alternatively, you can also create your own indices as shown:

```
sales = pd.DataFrame (data, index=['Hatchback', 'SUV', 'Sedan',  
                                'Convertible'])
```

```
sales
```

Your output will be as follows:

	Toyota	BMW
Hatchback	3	4
SUV	4	5
Sedan	1	2
Convertible	5	9

From this information, you can determine the number of orders made for each vehicle type using the name. You do this by using *loc* (from locate) as shown below:

```
sales.loc['Hatchback']
```

Output:

```
Toyota      3
```

```
BMW         4
```

```
Model: Hatchback, dtype: int64
```

You can use this knowledge to create DataFrames for different data models you are working on.

Loading Data into DataFrames

You will be working with different types and sources of data, hence you must learn how to load this into your DataFrame. We will maintain the same example above, but from different sources:

- CSV Files

For CSV files, load data using the following command:

```
df = pd.read_csv('sales.csv')
```

```
df
```

You will have the following output:

	Unnamed:0	Toyota	BMW
0	Hatchback	3	4
1	SUV	4	5
2	Sedan	1	2
3	Convertible	5	9

Remember that CSV files do not index files the way DataFrames do. Therefore, you will have to use the *index_col* designation to read the files as shown below:

```
df = pd.read_csv('sales.csv', index_col=0)
```

```
df
```

You will have the following output:

	Toyota	BMW
Hatchback	3	4
SUV	4	5
Sedan	1	2
Convertible	5	9

From the example above, the index is set to column zero. However, you will notice that when using CSV files, most of them lack an index column. For this reason, you can easily skip this step without any repercussions.

- JSON Files

JSON files are compatible with Python, so reading them should be easy as follows:

```
df = pd.read_json('sales.json')
```

```
df
```

You will have the following output:

	Toyota	BMW

Convertible	5	9
Hatchback	3	4
Sedan	1	2
SUV	4	5

In this case, the index is correct because Pandas uses the JSON indices. You can study this further by looking at the *data_file.json* file in your text editor.

Obtaining Data from SQL Databases

Before you begin, check to ensure you have a connection with the Python library in question. Once the connection is established, you can then push a query to Pandas. For this example, we will use SQLite:

Install `pysqlite3` through your terminal as follows:

```
pip install pysqlite3
```

Or run this code in your notebook

```
!pip install pysqlite3
```

You need SQLite to establish a connection with your database, from where you will then create a DataFrame using the *SELECT* query as follows:

```
import sqlite3  
  
con = sqlite3.connect("database.db")
```

Using our car dealership example above, the SQL database will have a table denoted as *sales*, and the index. We can read from the database using the command below:

```
df = pd.read_sql_query("SELECT * FROM sales", con)
```

df

You will have the following output:

	index	Toyota	BMW
0	Hatchback	3	4
1	SUV	4	5
2	Sedan	1	2
3	Convertible	5	9

Just as we did with the CSV files, you can also bypass the index as follows:

```
df = df.set_index('index')
```

df

You will have the output below:

	Toyota	BMW
index		
Hatchback	3	4
SUV	4	5
Sedan	1	2
Convertible	5	9

Once you are done with your data, you need to save it in a file system that is relevant to your needs. In Pandas, you can convert files to and from any of the file formats discussed above in the same way that you read the data files, when storing them as shown below:

```
df.to_csv('new_sales.csv')
```

```
df.to_sql('new_sales', con)
```

```
df.to_json('new_sales.json')
```

In data analysis, there are lots of methods that you can employ when using DataFrames, all of which are important to your analysis. Some operations are useful in performing simple data transformations, while others are necessary for complex statistical approaches.

In the examples below, we will use an example of a dataset from the English Premier League below:

```
squad_df = pd.read_csv("EPL-Data.csv", index_col="Teams")
```

As we load this dataset from the CSV file, we will use teams as our index.

To view the data, you must first open a new dataset by printing out rows as follows:

```
squad_df.head()
```

You will have the following Output:

	Position	Designation
Teams		
Manchester United	1	Champions League
Arsenal	2	Champions League
Chelsea	3	Champions League
Liverpool	4	Champions League

		Qualifiers
--	--	------------

`.head()` will by default print the first five rows of your DataFrame. However, if you need more rows displayed, you can input a specific number to be printed as follows:

```
squad_df.head(7)
```

This will output the top seven rows as shown below:

	Position	Designation
Teams		
Manchester United	1	Champions League
Arsenal	2	Champions League
Chelsea	3	Champions League
Liverpool	4	Champions League Qualifiers
Tottenham	5	Europa League
Everton	6	Europa League

In case you need to display only the last rows, use the `.tail()` syntax. You can also input a specific number. Assuming we want to determine the last three teams, we will use the syntax below:

```
squad_df.tail(3)
```

Our output will be as follows:

	Position	Designation
Teams		
Newcastle	18	Relegated
Watford	19	Relegated
Swansea	20	Relegated

Generally, whenever you access any dataset, you will often access the first five rows to determine whether you are looking at the correct data set. From the display, you can see the index, column names, and the preset values. You will notice from the example above that the index for our DataFrame is the *Teams* column.

Extracting Information from Data

The `.info()` command will help you derive information from your data sets. The syntax is as follows:

```
squad_df.info()
```

You will have the following output:

```
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 20 entries, Manchester United to Swansea
```

```
Data Columns (total 2 columns):
```

```
Position      20 non-null int64
```

```
Designation   20 non-null object
```

```
dtypes: int64 (1), object (1)
```

memory usage: 35.7+ KB

The `.info()` command will deliver all the important information you need about the dataset, including how many non-null values are available, the number of columns and rows, memory used by the DataFrame, and the type of data available in every column.

The dataset you are using might contain missing values in some columns. You will need to learn how to address these, to help in cleaning the data for final presentation.

Why do you need to determine the datatype? Without this, you might struggle to interpret data correctly. If, for example, you are using a JSON file but the integers are stored as strings, most of your operations will not work. This is because it is impossible to perform mathematical computations with strings. This is why the `.info()` is useful. You know the kind of content present in every column.

The `.shape` attribute can also help you because it delivers the tuple of rows and columns in the dataset. In the example above, you can have it as follows:

```
squad_df.shape
```

Your output will be as follows:

```
(20, 2)
```

It is also important to remember that there are no parentheses used in the `.shape` attribute. It basically returns the tuple format for rows and columns. In the example above, we have 20 rows and 2 columns in the `squad` DataFrame. As you work with different sets of data, you will use the `.shape` attribute a lot to transform and clean data.

Dealing with Duplicates

While the example we used above does not have any duplicate rows, you need to learn how to identify duplicates to ensure that you perform accurate computations. In the example above, we can append the squad DataFrame to itself and double it as shown:

```
temp_df = squad_df.append(squad_df)
```

```
temp_df.shape
```

Our output will be as follows:

```
(40, 2)
```

The *append()* attribute copies the data without altering the initial DataFrame. The example above does not use the real data, hence display in *temp*. In order to do away with the duplicates, we can use the following attribute:

```
temp_df = temp_df.drop_duplicates()
```

```
temp_df.shape
```

Our output will be as follows:

```
(20, 2)
```

The *drop_duplicates()* attribute works in the same manner that the *append()* attribute does. However, instead of doubling the DataFrame, it results in a fresh copy without duplicates. In the same example, *.shape* helps to confirm whether the dataset we are using has 20 rows as was present in the original file.

In Pandas, the keyword *inplace* is used to alter the DataFrame objects as shown below:

```
temp_df.drop_duplicates(inplace=True)
```

The syntax above will change your data automatically. The `drop_duplicates()` argument is further complemented with the `keep` argument in the following ways:

- *False* – This argument will eliminate all duplicates
- *Last* – This argument will eliminate all duplicates other than the last one.
- *First* – This argument will eliminate all duplicates other than the first one.

In the examples we used above, the `keep` argument has not been defined. Any argument that is not defined will always default to *first*. What this means is that if you have two duplicate rows, Pandas will maintain the first one but do away with the second.

If you use *last*, Pandas will drop the first row but maintain the second one.

Using *keep*, however, will eliminate all the duplicates. Assuming that both rows are similar, *keep* will eliminate both of them. Let's look at an example using `temp_df` below:

```
temp_df = squad_df.append(squad_df) # generate a fresh copy
temp_df.drop_duplicates(inplace=True, keep=False)
temp_df.shape
```

We will have the output below:

```
(0, 2)
```

In the above example, we appended the squad list, generating new duplicate rows. As a result, `keep=False` eliminated all the rows, leaving us with zero rows. This might sound absurd, but it is actually a useful technique that will

help you determine all the duplicates present in the dataset you are working on.

Cleaning Data in a Column

You will often come across datasets that have varying names for their columns. You will encounter typos, spaces, and a mixture of upper and lower-case words. Cleaning up these columns will make it easier for you to choose the correct column for your computations.

In the example above, the syntax below will help us print the column names:

```
squad_df.columns
```

You will have the following output:

```
Index (['Position', 'Designation'])
```

Once you have this information, you can use a simple command `.rename()` to rename some or all the columns in your data. Since we do not need to use any parentheses, we will rename the content as follows:

Assuming the Designation Column was named Designation (Next Season), you would have it renamed as follows

```
squad_df.rename(columns={  
    'Designation (Next Season)': 'Designation_next_season',  
}, inplace=True)  
squad_df.columns
```

Our output would look like this:

```
Index (['Position', 'Designation_next_season'])
```

You can also use the same process to change the column content from upper to lower case without having to enter all the connotations individually. A list comprehension will help you instead of manually changing the name of each item on the column list as shown below:

```
squad_df.columns = [col.lower() for col in squad_df]
```

```
squad_df.columns
```

You will have the following output:

```
Index (['position', 'designation_next_season'])
```

Over time, you will use a lot of *dict* and *list* attributes in Pandas. To make your work easier, it is advisable to do away with special characters and use lower case connotations instead. You should also use underscores instead of spaces.

Computation with Missing Values

One thing you can be certain about as a data analyst is that you will not always come across complete sets of data. Since data is collected by different people, they might not use the same conventions you prefer. Therefore, you can always expect to bump into some challenges with missing values in datasets.

In Python, you will encounter *None* or *np.nan* in NumPy whenever you come across such types of data. Since you must proceed with your work, you must learn how to handle such scenarios. You have two options: either replace the null values with non-null values or eliminate all the columns and rows that have null values.

First, you must determine the number of null values present in each column within your dataset. You can do this in the syntax below:

```
squad_df.isnull()
```

The result is a DataFrame that has True or False in each cell, in relation to the null status of the cell in question. From here, you can also determine the number of null returns in every column through an aggregate summation function as shown below:

```
squad_df.isnull().sum()
```

The result will list all the columns, and the number of null values in each.

To eliminate null values from your data, you have to be careful. It is only advisable to eliminate such data if you have deep knowledge of the explanation behind the null values. Besides, it is only advisable to eliminate null data if you are missing a small amount. This should not have a noteworthy effect on the data. The following syntax will help you eliminate null data from your work:

```
squad_df.dropna()
```

The syntax above eliminates all rows with at least one null value from your dataset. However, this syntax will also bring forth a new DataFrame without changing the original DataFrame you have been using.

The problem with this operation is that it will eliminate data from the rows with null values. However, some of the columns might still contain some useful information in the eliminated rows. To circumvent this challenge, we must learn how to perform imputation on such datasets.

Instead of eliminating rows, you can choose to eliminate columns that contain null values too. This is performed with the syntax below:

```
axis=1
```

For example,

```
squad_df.dropna(axis=1)
```

What is the explanation behind the *axis=1* attribute? Why does it have to be 1 in order to work for columns? To understand this, we take a closer look at the *.shape* output discussed earlier.

```
squad_df.shape
```

Output

```
(20,2)
```

In the example above, the syntax returns the DataFrame in the form of a tuple of 20 rows and 2 columns. In this tuple, rows are represented as index zero, while columns are represented as index one. From this explanation, therefore, *axis=1* will work on columns.

Data Imputation

Imputation is a cleaning process that allows you to maintain valuable data in your DataFrames, even if they have null values. This is important in situations where eliminating rows that contain null values might eliminate a lot of data from your dataset. Instead of losing all values, you can use the median or mean of the column in place of the null value.

Using the example above, and assuming a new column for earnings from gate receipts earned by the clubs over the season. Some values are missing in that revenue column. To begin, you must extract the revenue column and use it as a variable. This is done as shown below:

```
earnings = squad_df['earnings_billions']
```


Take note that when you are selecting columns to use from a DataFrame, you must enclose them with square brackets as shown above.

To handle the missing values, we can use the mean as follows:

```
earnings_mean = earnings.mean()

earnings_mean
```

The output should deliver the mean of all the values in the specified cells. Once you have this, you replace it in the null values using the following syntax: *fillna()* as shown below:

```
earnings.fillna(earnings_mean, inplace=True)
```

This will replace all the null values in the earnings column with the mean of that column. The syntax *inplace=True* changes the original *squad_df*.

Describing Variables

There is so much more information you can get from your DataFrames. A summary of the continuous variables can be arrived at using the following syntax:

```
squad_df.describe()
```

This will return information about continuous numbers. This information is useful when you are uncertain about the kind of plot diagram to use for visual representation. *.describe()* is a useful attribute because it returns the number of rows, categories, and frequency of the top category about a specific column.

```
squad_df['position'].describe()
```

The syntax above will return an output in the following format:

count	xx
unique	xx
top	xx
freq	xx

Name: genre, dtype: object

What we can deduce from this output is that the selected column contains xx number of unique values, the top value in that column, and the fact that the top column shows up xx number of times (*freq*). To determine the frequency of all the values in the position column, you use the syntax below:

```
squad_df['position'].value_counts().head(10)
```

You can also find out the relationship between different continuous variables using the *.corr()* syntax as shown below:

```
squad_df.corr()
```

The output is a correlation table that represents different relationships in your dataset. You will notice positive and negative values in the output table. Positive results show a positive correlation between the variables. This means that one variable rises as the other rises and vice versa. Negative results show an inverse correlation between the variables. This means that one variable will rise as the other falls. A perfect correlation is represented by 1.0. A perfect correlation is obvious for each column with itself.

Data Manipulation

By this point, you are aware of how to draw summaries from the data in your possession. Beyond this, you should learn how to slice, select, and extract data from your DataFrame. We mentioned earlier that DataFrames and Series

share many similarities, especially in the methods used on them. However, their attributes are not similar. Therefore you must be keen to make sure you are using the right attributes, or you will end up with attribute errors.

To extract a column, you use square brackets as shown below:

```
position_col = squad_df['position']  
  
type(position_col)
```

You will get the output below:

```
pandas.core.series.Series
```

The result is a Series. However, if you need to return the column as a Dataframe, you must use column names as shown below:

```
position_col = squad_df[['position']]  
  
type(position_col)
```

You will get the output below:

```
pandas.core.frame.DataFrame
```

What you have now is a simple list. Onto this list, you can add a new column as follows:

```
subset = squad_df[['position', 'earnings']]  
  
subset.head()
```

You should get the output below:

	Position	Earnings
Teams		
Manchester	1	xx

United		
Arsenal	2	xx
Chelsea	3	xx
Liverpool	4	xx
Tottenham	5	xx
Everton	6	xx

Next, we will look at how to call data from your DataFrame using rows. You can do this using any of the following means:

- Locating the name (*.loc*)
- Locating the numerical index (*.iloc*)

Since we will still be indexed using the Teams, we must use *.loc* and assign it the name of the team as shown below:

```
eve = squad_df.loc["Everton"]
```

```
eve
```

Another option is to use *.iloc* for the numerical index of Everton as shown below:

```
eve = squad_df.iloc[1]
```

The *.iloc* slice works in the same way that you slice lists in Python. Therefore, the item found in the index section at the end is omitted.

Chapter 5

Data Cleaning

Data cleaning is one of the most important procedures you should learn in data analysis. You will constantly be working with different sets of data and the accuracy or completeness of the same is never guaranteed. Because of this reason, you should learn how to handle such data and make sure the incompleteness or errors present do not affect the final outcome.

Why should you clean data, especially if you did not produce it in the first place? Using unclean data is a sure way to get poor results. You might be using a very powerful computer capable of performing calculations at a very high speed, but what they lack is intuition. Without this, you must make a judgement call each time you go through a set of data.

In data analysis, your final presentation should be a reflection of the reality in the data you use. For this reason, you must eliminate any erroneous entries.

Possible Causes of Unclean Data

One of the most expensive overheads in many organizations is data cleaning. Unclean data is present in different forms. Your company might suffer in the form of omissions and errors present in the master data you need for analytical purposes. Since this data is used in important decision-making processes, the effects are costly. By understanding the different ways dirty data finds its way into your organization, you can find ways of preventing it, thereby improving the quality of data you use.

In most instances, automation is applied in data collection. Because of this, you might experience some challenges with the quality of data collected or consistency of the same. Since some data is obtained from different sources, they must be collated into one file before processing. It is during this process that concerns as to the integrity of the data might arise. The following are some explanations as to why you have unclean data:

- ✓ **Incomplete data**

The problem of incomplete data is very common in most organizations. When using incomplete data, you end up with many important parts of the data blank. For example, if you are yet to categorize your customers according to the target industry, it is impossible to create a segment in your sales report according to industry classification. This is an important part of your data analysis that will be missing, hence your efforts will be futile, or expensive in terms of time and resources invested before you get the complete and appropriate data.

- ✓ **Errors at input**

Most of the mistakes that lead to erroneous data happen at data entry points. The individual in charge might enter the wrong data, use the wrong formula, misread the data, or innocently mistype the wrong data. In the case of an open-ended report like questionnaires, the respondents might input data with typos or use words and phrases that computers cannot decipher appropriately. Human error at input points is always the biggest challenge in data accuracy.

- ✓ **Data inaccuracies**

Inaccurate data is in most cases a matter of context. You could have the correct data, but for the wrong purpose. Using such data can have far-reaching effects, most of which are very costly in the long run. Think about the example of a data analyst preparing a delivery schedule for clients, but the

addresses are inaccurate. The company could end up delivering products to their customers, but with the wrong address details. As a matter of context, the company does have the correct addresses for their clients, but they are not matched correctly.

- ✓ **Duplicate data**

In cases where you collect data from different sources, there is always a high chance of data duplication. You must have a lot of checks in place to ensure that duplicates are identified. For example, one report might list student scores under Results, while another will have them under Performance. The data under these tags will be similar, but your sensors will consider them as two independent entities.

- ✓ **Problematic sensors**

Unless you are using a machine that periodically checks for errors and corrects them or alerts you, it is possible to encounter errors as a result of problematic sensors. Machines can be faulty or breakdown too, which increases the likelihood of a problematic data entry.

- ✓ **Incorrect data entries**

An incorrect entry will always deliver the wrong result. Incorrect entry happens when your dataset includes entries that are not within the acceptable range. For example, data for the month of February should range from 1 to 28 or 29. If you have data for February ranging up to 31, there is definitely an error in your entries.

- ✓ **Data mungling**

If at your data entry point you use a machine with problematic sensors, it is possible to record erroneous values. You might be recording people's ages, and the machine inputs a negative figure. In some cases, the machine could actually record correct data, but between the input point and the data

collection point, the data might be mungled, hence the erroneous results. If you are accessing data from a public internet connection, a network outage during data transmission might also affect the integrity of the data.

- ✓ **Standardization concerns**

For data obtained from different sources, one of the concerns is often how to standardize the data. You should have a system or method in place to identify similar data and represent them accordingly. Unfortunately, it is not easy to manage this level of standardization. As a result, you end up with erroneous entries. Apart from data obtained from multiple sources, you can also experience challenges dealing with data obtained from the same source. Everyone inputs data uniquely, and this might pose a challenge at data analysis.

How to Identify Inaccurate Data

More often, you need to make a judgement call to determine whether the data you are accessing is accurate or not. As you go through data, you must make a logical decision based on what you see. The following are some factors you should think about:

- ✓ **Study the range**

First, check the range of data. This is usually one of the easiest problems to identify. Let's say you are working on data for primary school kids. You know the definitive age bracket for the students. If you identify age entries that are either too young or too old for primary school kids whose data you have, you need to investigate further.

Essentially what you are doing here is an overview of a max-min approach. With these ranges in mind, you can skim through data and identify erroneous entries. Skimming through is easy if you are working with a few entries. If

you have thousands or millions of data entries, a max-min function code can help you identify the wrong entries in an instant. You can also plot the data on a graph and visually detect the values that don't fall within the required distribution pattern.

- ✓ **Investigate the categories**

How many categories of data do you expect? This is another important factor that will help you determine whether your data is accurate or not. If you expect a dataset with nine categories, anything less is acceptable, but not more. If you have more than nine categories, you should investigate to determine the legitimacy of the additional categories. Say you are working with data on marital status, and your expected options are single, married, divorced, or widowed. If the data has six categories, you should investigate to determine why there are two more.

- ✓ **Data consistency**

Look at the data in question and ensure all entries are consistent. In some cases, inaccuracies appear as a result of inconsistency. This is common when working with percentages. Percentages can either be fed into data sets as basis points or decimal points. If you have data that has both sets of entries, they might be incompatible.

- ✓ **Inaccuracies across multiple fields**

This is perhaps one of the most difficult challenges you will overcome when cleaning inaccurate data. The following entries, for example, are valid individually. A 4-year old girl is a valid age entry. 5 children is also a valid entry. However, a datapoint that depicts Grace as a 4-year old girl with 5 children is absurd. You would need to check for inconsistencies and inaccuracies in several rows and columns.

- ✓ **Data visualization**

Plotting data in visual form is one of the easiest ways of identifying abnormal distributions or any other errors in the data. Say you are working with data whose visualization should result in a bimodal distribution, but when you plot the data you end up with a normal distribution. This would immediately alert you that something is not right, and you need to check your data for accuracy.

- ✓ **Number of errors in your data set**

Having identified the unique errors in the data set, you must enumerate them. Enumeration will help you make a final decision on how and whether to use the data. How many errors are there? If you have more than half of the data as inaccurate, it is obvious that your presentation would be greatly flawed. You must then follow up with the individuals who prepared the data for clarification or find an alternative.

- ✓ **Missing entries**

A common data concern that data analysts deal with is working with datasets missing some entries. Missing entries is relative. If you are missing two or three entries, this should not be a big issue. However, if your data set is missing many entries, you have to find out the reason behind this.

Missing entries usually happen when you are collating data from multiple sources, and in the process some of the data is either deleted, overwritten, or skipped. You must investigate the missing entries because the answer might help you determine whether you are missing only a few entries that might be insignificant going forward, or important entries whose absence affects the outcome.

How to Clean Data

Having gone through the procedures described above and identified unclean data, your next challenge is how to clean it and use accurate data for analysis.

You have five possible alternatives for handling such a situation:

- ✓ **Data imputation**

If you are unable to find the necessary values, you can impute them by filling in the gaps for the inaccurate values. The closest explanation for imputation is that it is a clever way of guessing the missing values, but through a data-driven scientific procedure. Some of the techniques you can use to impute missing data include stratification and statistical indicators like mode, mean and median.

If you have studied the data and identified unique patterns, you can stratify the missing values based on the trend identified. For example, men are generally taller than women. You can use this presumption to fill in missing values based on the data you already have.

The most important thing, however, is to try and seek a second opinion on the data before imputing your new values. Some datasets are very critical, and imputing might introduce a personal bias which eventually affects the outcome.

- ✓ **Data scaling**

Data scaling is a process where you change the data range so that you have a reasonable range. Without this, some values that might appear larger than others might be given prominence by some algorithms. For example, the age of a sample population generally exists within a smaller range compared to the average population of a city. Some algorithms will give the population priority over age, and might ignore the age variable altogether.

By scaling such entries, you maintain a proportional relationship between different variables, ensuring that they are within a similar range. A simple way of doing this is to use a baseline for the large values, or use percentage

values for the variables.

- ✓ **Correcting data**

Correcting data is a far better alternative than removing data. This involves intuition and clarification. If you are concerned about the accuracy of some data, getting clarification can help allay your fears. With the new information, you can fix the problems you identified and use data you are confident about in your analysis.

- ✓ **Data removal**

One of the first things you could think about is to eliminate the missing entries from your dataset. Before you do this, it is advisable that you investigate to determine why the entries are missing. In some cases, the best option is to remove the data from your analysis altogether. If, for example, more than 80% of entries in a row is missing and you cannot replace them from any other source, that row will not be useful to your analysis. It makes sense to remove it.

Data removal comes with caveats. If you have to eliminate any data from your analysis, you must give a reason for this decision in a report accompanying your analysis. This is important so as to safeguard yourself from claims of data manipulation or doctoring data to suit a narrative.

Some types of data are irreplaceable, so you must consult experts in the associated fields before you remove them. Most of the time, data removal is applied when you identify duplicates in the data, especially if removing the duplicates does not affect the outcome of your analysis.

- ✓ **Flagging data**

There are situations where you have columns missing some values, but you cannot afford to eliminate all of them. If you are working with numeric data,

a reprieve would be to introduce a new column where you indicate all the missing values. The algorithm you are using should identify these values as such. In case the flagged values are necessary in your analysis, you can impute them or find a better way to correct them then use them in your analysis. In case this is not possible, make sure you highlight this in your report.

Cleaning erroneous data can be a difficult process. A lot of data scientists generally hope to avoid it, especially since it is time-consuming. However, it is a necessary process that will bring you closer to using appropriate data for your analysis. Remember that the main objective is to use clean data that will give you the closest reflection of the true picture of events.

How to Avoid Data Contamination

From empty data fields to data duplication and invalid addresses, there are so many ways you can end up with contaminated data. Having looked at possible causes and methods of cleaning data, it is important for an expert in your capacity to put measures in place to prevent data contamination in the future. The challenges you experienced in cleaning data could easily be avoided, especially if the data collection processes are within your control.

Looking back to the losses your business suffers in dealing with contaminated data and the resource wastage in terms of time, you can take significant measures to reduce inefficiencies, which will eventually have an impact on your customers and their level of satisfaction.

One of the most important steps today is to invest in the appropriate CRM programs to help in data handling. Having data in one place makes it easier to verify the credibility and integrity of data within your database. The following are some simple methods you can employ in your organization to

prevent data contamination, and ensure you are using quality data for decision-making.

- ✓ **Proper configurations**

Irrespective of the data handling programs you use, one of the most important things is to make sure you configure applications properly. Your company could be using CRM programs or simple Excel sheets. Whichever the case, it is important to configure your programs properly.

Start with the critical information. Make sure the entries are accurate and complete. One of the challenges of incomplete data is that there is always the possibility that someone could complete them with inaccurate data to make them presentable, when this is not the real picture.

Data integrity is just as important, so make sure you have the appropriate data privileges in place for anyone who has to access critical information. Set the correct range for your data entries. This way, anyone keying in data will be unable to enter incorrect data not within the appropriate range. Where possible, set your system up such that you can receive notifications whenever someone enters the wrong range, or is struggling, so that you can follow up later on and ensure you captured the correct data.

- ✓ **Proper training**

Human error is one of a data analyst's worst nightmares when trying to prevent data contamination. Other than innocent mistakes, many errors from human entry are usually about context. It is important that you train everyone handling data on how to go about it. This is a good way to improve accuracy and data integrity from the foundation - data entry.

Your team must also understand the challenges you experience when using contaminated data, and more importantly why they need to be keen at data

entry. If you are using CRM programs, make sure they understand different functionality levels so they know the type of data they should enter.

Another issue is how to find the data they need. When under duress, most people key in random or inaccurate data to get some work done or bypass some restrictions. By training them on how to search for specific data, it is easier to avoid unnecessary challenges with erroneous entries. This is usually a problem when you have new members joining your team. Ensure you train them accordingly, and encourage them to ask for help whenever they are unsure of anything.

- ✓ **Entry formats**

The data format is equally important as the desired level of accuracy. Think about this from a logical perspective. If someone sends you a text message written in all capital letters, you will probably disregard it or be offended by the tone of the message. However, if the same message is sent with proper formatting, your response is more positive.

The same applies to data entry. Try and make sure that everyone who participates in data handling is careful enough to enter data using the correct format. Ensure the formats are easy to understand, and remind the team to update data they come across if they realize it is not in the correct format. Such changes will go a long way in making your work easier during analysis.

- ✓ **Empower data handlers**

Beyond training your team, you also need to make sure they are empowered and aware of their roles in data handling. One of the best ways of doing this is to assign someone the data advocacy role.

A data advocate is someone whose role is to ensure and champion consistency in data handling. Such a person will essentially be your data

administrator. Their role is usually important, especially when implementing new systems. They come up with a plan to ensure data is cleaned and organized. One of their deliverables should include proper data collection procedures to help you improve the results obtained from using the data in question.

- ✓ **Overcoming data duplication**

Data duplication happens in so many organizations because the same data is processed at different levels. Duplication might eventually see you discard important and accurate data accidentally, affecting any results derived from the said data.

For example, ensure your team searches for specific items before they create new ones. Provide an in-depth search process that increases the search results and reduces the possibility of data duplication. For example, beyond looking for a customer's name, the entry should also include contact information.

Provide as many relevant fields that can be searched into, thereby increasing the possibility of arresting and avoiding duplicates. You can find data for a customer named Charles McCarthy in different databases labeled as Charles MacCarthy or Charles Mc Carthy. The moment you come across such duplicates, the last thing you want to do is to eliminate them from the database. Instead, investigate further to ascertain the similarities and differences between the entries.

Consult, verify, and update the correct entry accordingly. Alternatively, you can escalate such issues to your data advocate for further action. At the same time, put measures in place that scans your database to warn users whenever they are about to create a duplicate entry.

- ✓ **Data filtration**

Perhaps one of the best solutions would be cleaning data before it gets into your database. A good way of doing this would be creating clear outlines on the correct data format to use. With such procedures in place, you have an easier time handling data. If all the conditions are met, you will probably handle data cleaning at the entry point instead of once the data is in your database, making your work easier.

Create filters to determine the right data to collect and the data that can be updated later. It doesn't make sense to collect a lot of information to give you the illusion of a complete and elaborate database, when in a real sense very little of what you have is relevant to your cause.

The misinformation that arises from inaccurate data can be avoided if you take the right precautionary measures in data handling. Data security is also important, especially if you are using data sources where lots of other users have access. Restrict access to data where possible, and make sure you create different access privileges for all users.

Chapter 6

Data Visualization with Matplotlib in Python

Data visualization is one of the first things you have to perform before you analyze data. The moment you have a glance at some data, your mind creates a rough idea of the way you want it to look when you map it on a graph.

Matplotlib might seem rather complex at first, but with basic coding knowledge, it should be easier for you. Many of the beginner concepts were addressed in the earlier books in this series. However, to refresh your memory we will highlight some of the important concepts that will guide your work going forward.

Plotting data for visualization will need you to work with different data ranges. You might need to work with general or specific data ranges. The whole point behind Matplotlib is to help you work with data with as minimal challenges as possible. As a data analyst, you are in full control over the data you use, hence you must also understand the necessary commands to alter the same.

Remember that the machine learning environment in Matplotlib is almost similar to MATLAB. Therefore, if you have some experience with MATLAB, you should find things easier here. All the work you do in Matplotlib is built in a hierarchical manner. At the highest point, you have a state-machine environment, while at the lowest level you have the object-oriented interfaces where pyplot only performs a limited number of functions.

At this level, it is up to you to build figures, and from them you can create axes. The axes will help in all, if not most of your plotting needs.

To install Matplotlib on your machine, run the following Python command:

```
python -m pip install -U pip
```

```
python -m pip install -U matplotlib
```

To set you off, install Matplotlib on your device using the following commands:

```
pip install matplotlib
```

```
xcode-select -install (if you are working on a Mac)
```

There are several dependencies that you might need to install with Matplotlib, including NumPy and Python if it is not already installed on your device. To further enhance your interface output, you might also need to install other packages like Tornado and pycairo.

If you are going to work on animations from time to time, you might need to install ImageMagick or any other packages that could assist you like LaTeX.

Fundamentals of Matplotlib

Below are some of the important concepts that you shall come across and use in Matplotlib, and their meanings or roles:

- **Axis** – This represents a number line, and is used to determine the graph limits.
- **Axes** – These represent what we construe as plots. A single figure can hold as many axes as possible. In the event of a 3D object, you can have two or three objects. Take note that for all axes, you must have an x and y label.

- Artist – Refers to everything that you can see on your figure, for example *collection* objects, *Line2D* objects and *Text* objects. You will notice that most of the Artists are on the Axes.
- Figure – Refers to the entire figure you are working on. It might include more than one plots or axes.

Pyplot is a Matplotlib module that allows you to work with simple functions, in the process adding elements like text, images, and lines within the figure you are working on. A simple plot can be created in the following manner:

```
import matplotlib.pyplot as plt  
  
import numpy as np
```

Basic Matplotlib Functions

There are lots of command functions that you can use to help you work with Matplotlib in the same way you would use MATLAB. Each of these pyplot functions changes figures in one way or the other when executed. The following is a list of the plots you will use in Matplotlib:

- Quiver – Used to create 2D arrow fields
- Step – Used to create a step plot
- Stem – Used to build a stem plot
- Scatter – Creates a scatter plot of x against y
- Stackplot – Used to create a plot for a stacked area
- Plot – Creates markers or plot lines to your axes
- Polar – Creates a polar plot
- Pie – Creates a pie chart

- Barh - Creates a horizontal bar plot
- Bar – Creates a bar plot
- Boxplot – Creates a whisker and box plot
- Hist – Used to create a histogram
- Hist2d – Used to create a histogram plot in 2D

Given that you might be working with images from time to time during data analysis, you will frequently use the following image functions:

- Imshow – Used to show images on your axes
- Imsave – Used to save arrays in the form of an image file
- Imread – Used to read files from images into arrays

Plotting Function Inputs

You must first import the Pyplot module from your Matplotlib package before you can create a plot. This is done as shown below:

```
import matplotlib.pyplot as plt
```

After importing the module, you introduce arrays into the plot. The NumPy library has predefined array functions that you will use going forward. These are imported as follows:

```
import numpy as np
```

With this done, proceed to introduce objects into the plot using the NumPy library's *arange()* function as shown below:

```
x = np.arange(0, math.pi*2, 0.05)
```

With this data, you can then proceed to specify the x and y axis labels, and

the plot title as shown:

```
plt.xlabel("angle")
plt.ylabel("sine")
plt.title('sine wave')
```

To view the window, use the `show()` function below:

```
plt.show()
```

At this juncture, your program should look like this:

```
from matplotlib import pyplot as plt
import numpy as np
import math #will help in defining pi
x = np.arange(0, math.pi*2, 0.05)
y = np.sin(x)
plt.plot(x,y)
plt.xlabel("angle")
plt.ylabel("sine")
plt.title('sine wave')
plt.show()
```

Basic Matplotlib Plots

Before you plot on matplotlib, you must have a `plot ()` function within the `matplotlib.pyplot` subpackage. This is to give you the basic plot with x-axis and y-axis variables.

Alternatively, you can also use format parameters to represent the line style you are using. To determine the format parameters and options used, the following commands apply:

```
$ ipython -pylab
```

```
In [1] : help(plot)
```

In the example above, you are creating two unique lines. The first one, which will act as the default line, is the solid line style, while the second one will have a dashed line. Study the code snippet below. We will use it to describe the procedure on how to create a simple plot.

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
x = np.linspace(0, 20)
```

```
plt.plot(x, .5 + x)
```

```
plt.plot(x, 1 + 2 * x, '--')
```

```
plt.show()
```

Use the following procedure to plot the lines described above:

Step 1:

Determine the x coordinates using *linspace ()*, a NumPy function. The x coordinates start at 0 and end at 20, hence you should have the following function:

```
x = np.linspace(0, 20)
```

Step 2:

Plot the lines on your axis in the following order:

```
plt.plot(x, .5 + x)
```

```
plt.plot(x, 1 + 2 * x, '--')
```

Step 3:

At this point, you have two options. You can save the plot or view it on a screen. The *savefig()* function is used to save the file. If you have to view it, the *show ()* function is used. To view the function on the screen, use the following plotting function:

```
plt.show()
```

Logarithmic Plots (Log Plots)

A logarithmic plot is essentially a basic plot, but it is set on a logarithmic scale. The difference between this and a normal linear scale is that the intervals are set in order of their magnitude. We have two different types of log plots: the log-log plot and semi-log plot.

The log-log plot has logarithm scales on both the x and y axis. In matplotlib, this plot is identified by the following function: *matplotlib.pyplot.loglog()*.

The semi-log plot, on the other hand, uses two different scales. It has a logarithmic scale on one axis and a linear scale on the other. They are identified by the following functions: *semilogx()* for the x axis, and *semilogy()* for the y axis. Straight lines in such plots are used to identify exponential laws.

The code below represents data on transistor counts within a given range of years. We will use it to study the procedure for creating logarithmic plots:


```
import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

df = pd.read_csv('transcount.csv')

df = df.groupby('year').aggregate(np.mean)

years = df.index.values

counts = df['trans_count'].values

poly = np.polyfit(years, np.log(counts), deg=1)

print "Poly", poly

plt.semilogy(years, counts, 'o')

plt.semilogy(years, np.exp(np.polyval(poly, years)))

plt.show()
```

Step 1:

Build the data using the following functions:

```
poly = np.polyfit(years, np.log(counts), deg=1)

print "Poly", poly
```

Step 2:

From the data fit above, you should have a polynomial object. Based on the data available, you should have the polynomial coefficients arranged in descending order.

Step 3:

To study the polynomial created, use the NumPy function *polyval()*. Plot data and use the y axis semi-log function as shown:

```
plt.semilogy(years, counts, 'o')  
  
plt.semilogy(years, np.exp(np.polyval(poly, years)))
```

Scatter Plots

The role of a scatter plot is to identify the relationship between a couple of variables displayed in a coordinate system. Each data point is identified according to the variable values. From the scatter graph, you can tell whether there is a relationship between the variables or not.

When studying a scatter plot diagram, the direction of the trend tells you the nature of correlation. A positive correlation, for example, is represented by an upward pattern. A scatter plot can also be used alongside a bubble chart. Bubble charts introduce a third variable beyond the two identified in the scatter plot. The size of the bubble around the data points is used to determine the value of the third variable.

In matplotlib, scatter plots are called through the *scatter ()* function. The following commands are used to access the scatter function's documentation:

```
$ ipython -pylab
```

```
In [1] : help(scatter)
```

In the example below, we introduce three parameters, *s* to represent the size of the bubble chart, *alpha* to represent the transparency of the bubbles when plotted on the chart, and *c* to represent the colors. The alpha variable values are in the range of 0 - completely transparent, and 1 - completely opaque. You will have a scatter chart with the following coordinates:

```
plt.scatter(years, cnt_log, c= 200 * years, s=20 + 200 *  
gpu_counts/gpu_counts.max(), alpha=0.5)
```

You should have the following code:

```
import matplotlib.pyplot as plt  
  
import numpy as np  
  
import pandas as pd  
  
df = pd.read_csv('transcount.csv')  
  
df = df.groupby('year').aggregate(np.mean)  
  
gpu = pd.read_csv('gpu_transcount.csv')  
  
gpu = gpu.groupby('year').aggregate(np.mean)  
  
df = pd.merge(df, gpu, how='outer', left_index=True,  
right_index=True)  
  
df = df.replace(np.nan, 0)  
  
print df  
  
years = df.index.values  
  
counts = df['trans_count'].values  
  
gpu_counts = df['gpu_trans_count'].values  
  
cnt_log = np.log(counts)  
  
plt.scatter(years, cnt_log, c= 200 * years, s=20 + 200 * gpu_counts/  
gpu_counts.max(), alpha=0.5)  
  
plt.show()
```

Display Tools in Matplotlib

There are different display tools you can use to help you understand a plot the first time you see it. Legends and annotations serve this purpose. Legends identify different series of data within your plot. To access it, you call the matplotlib function *legend ()*.

Annotations, on the other hand, help in identifying the important points in the plot. Annotations are called using the matplotlib function *annotate()*. An annotation must always have an arrow and a label, each of which could be described by different parameters. Because of this reason, you can use the *help (annotate)* function to get the best explanation.

Other display tools include labels, grids, and titles. A label will be present on both axes, but you can call them using the functions *xlabel ()* and *ylabel ()* for the x and y axis respectively. The title of your plot can be identified using the *title ()* function, while the grid is identified using the *grid ()* function. It is wise to note that you can turn the grid plot on or off where necessary.

In Matplotlib, you will be working with a lot of tools and functions that enhance manipulation and representation of the objects you work with, alongside any internal objects that might be present. By design, matplotlib is built into three layers as shown below:

- ✓ **The scripting layer**

This layer is also referred to as the *pyplot*. This is where functions and artist classes operate. The *pyplot* is an interface used in data visualization and analysis.

- ✓ **The artist layer**

This is an intermediate Matplotlib layer. All the elements in this layer are used in building charts, and include things like markers, titles, and labels

assigned to the x and y axis.

- ✓ **The backend layer**

This is the lowest level in Matplotlib. All the APIs are found in this layer. At this point, graphic element implementation takes place, albeit at the lowest possible level.

Each of these layers can only share communication with the layer beneath it, but not the one above it, hence the nature of communication in Matplotlib is unidirectional.

Having mentioned *pyplot*, you should also learn about *pylab*. *PyLab* is a unique module that is installed together with Matplotlib, while *pyplot* on the other hand runs as an internal package in Matplotlib. Your installation code for these two will look like this:

```
from pylab import *  
  
and  
  
import matplotlib.pyplot as plt  
  
import numpy as np
```

PyLab allows you to enjoy the benefits of using *pyplot* and NumPy within the same namespace, without necessarily having to import NumPy as a separate package. If you already have *pylab* imported, you will not need to call the NumPy and *pyplot* functions because they are automatically called, in a process similar to what you experience in MATLAB as shown below:

Instead of having

```
plt.plot()  
  
np.array([1,2,3,4])
```

You will have

```
plot(x,y)
```

```
array([1,2,3,4])
```

Essentially, the role of the *pyplot* package is to enable you to program in Python through the matplotlib library.

How to Create a Chart

Before you begin, import *pyplot* to your programming environment and set the name as *plt* as shown below:

Input

```
import matplotlib.pyplot as plt
```

Input

```
plt.plot([1,2,3,4])
```

Output

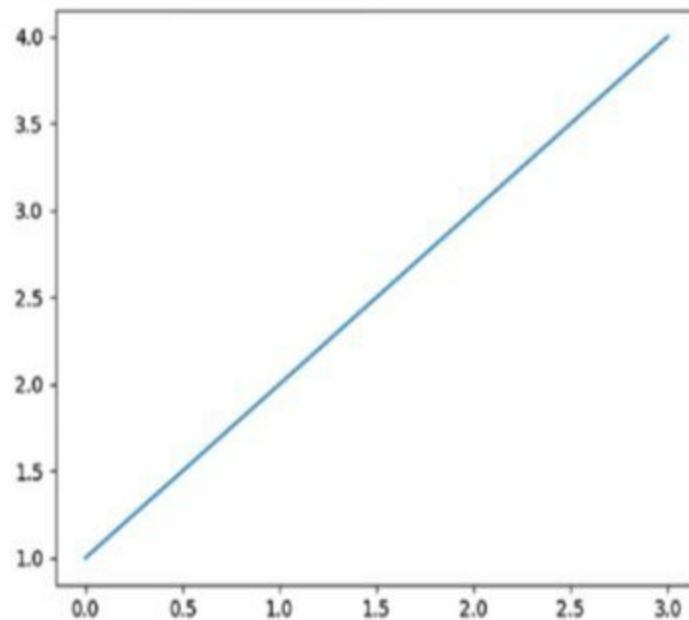
```
[<matplotlib.lines.Line2D at 0xa3eb438>]
```

When you enter this code, you will have created a Line2D object. An object in this case is a linear representation of the trends you will plot within a given chart. To view the plot, you will use the function below:

Input

```
plt.show()
```

The result should be a plotting window similar to the one below:



Depending on the platform you are using, in some cases your chart will display without necessarily calling the `show()` function, especially if you are using iPython QtConsole. Once this plot is prepared you must provide a definition for the two arrays on the x and y axis. The blue line in the example above represents all the points in your plot. This is the default configuration when your data does not have a legend, axis labels, or a title.

Using Multiple Axes and Figures

Beyond using pyplot commands for single figures, you can work with lots of figures at the same time in Matplotlib. You can take things further and introduce new plots within each figure. Other than using multiple subplots, you can also use the `subplot()` function to create multiple drawing areas in the main figure.

The `subplot()` function also helps you choose the subplot to focus your work on. Once selected, any commands passed will be called on the current subplot. A careful look at the `subplot()` function reveals three integers, each

of which serves a unique role.

The first integer outlines the number of vertical divisions available in the figure. The second integer outlines the number of horizontal divisions available in the figure. The third integer outlines the subplot where your commands are directed.

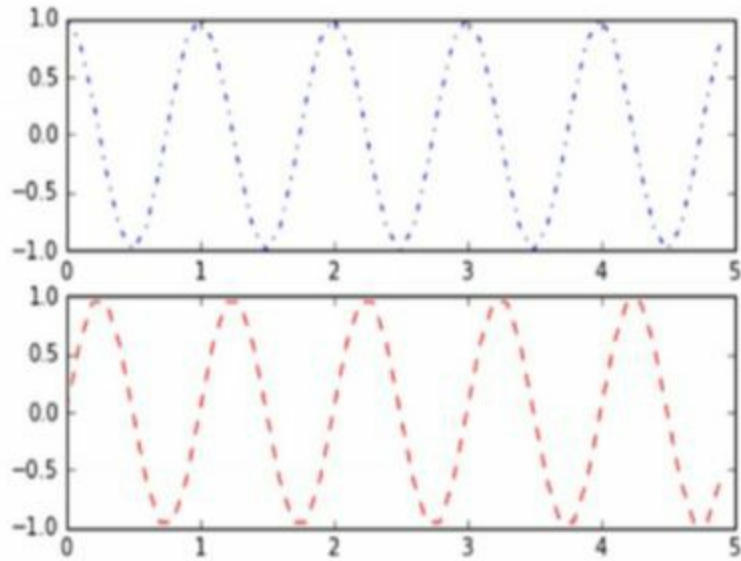
Input

```
t = np.arange(0,5,0.1)
: y1 = np.sin(2*np.pi*t)
: y2 = np.sin(2*np.pi*t)
```

Input

```
plt.subplot(211)
: plt.plot(t,y1,'b-.')
: plt.subplot(212)
: plt.plot(t,y2,'r--')
```

You should have the following plot:



In the next example, we will create vertical divisions from the plots above using the code below:

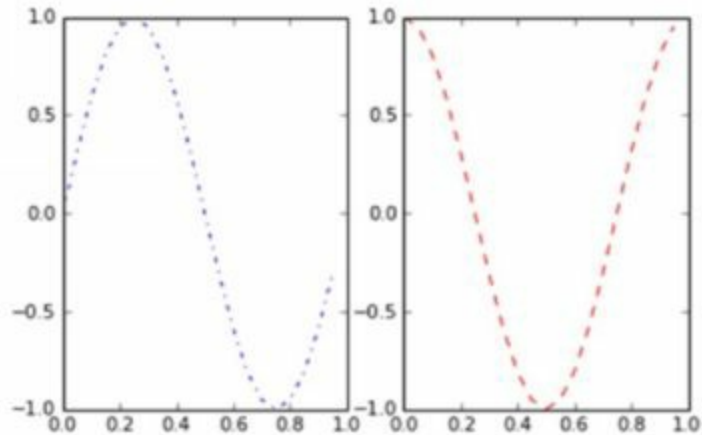
Input

```
t = np.arange(0.,1.,0.05)  
: y1 = np.sin(2*np.pi*t)  
: y2 = np.cos(2*np.pi*t)
```

Input

```
plt.subplot(121)  
: plt.plot(t,y1,'b-.')  
: plt.subplot(122)  
: plt.plot(t,y2,'r--')
```

You should have the plot below:



Introducing New Elements to Your Plot

Charts are supposed to make your data visually appealing. To do this, it is important to ensure you use the correct chart to represent the data you need, because not all charts are suitable for any kind of data. The basic lines and markers will not be sufficient in making the charts appealing. You should think of getting additional elements into the chart for this purpose.

How to add text to a chart

With the *title()* function, you can introduce an elaborate title into the chart as we have seen earlier. Beyond that, you should also be able to introduce the *axis* label. This is done with the *xlabel()* and *ylabel()* functions.

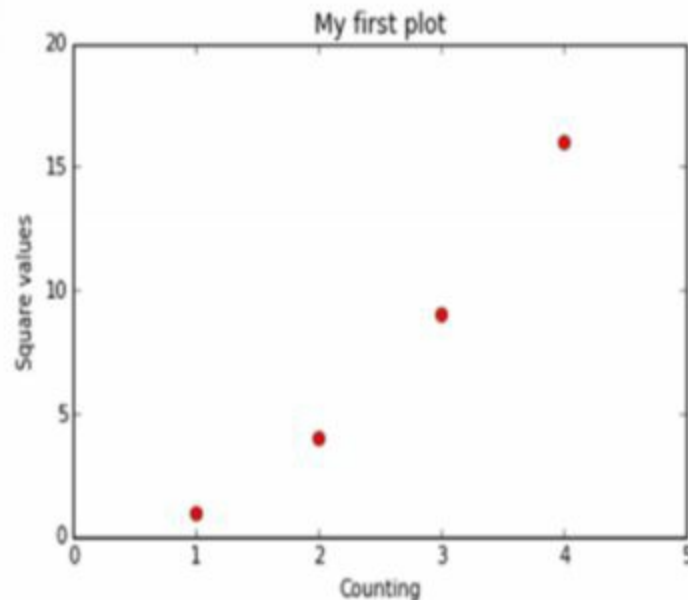
Remember that when you introduce a new function like the axis label functions, they create an argument within the string of code you are working with. We want to introduce the axis labels to a chart. This is the first step because they help you identify the values that will be assigned to every axis as you plot data. Your illustration should follow the code below:

Input

```
plt.axis([0,5,0,20])
```

```
: plt.title('My first plot')  
:  
: plt.xlabel('Counting')  
:  
: plt.ylabel('Square values')  
:  
: plt.plot([1,2,3,4], [1,4,9,16], 'ro')
```

You should have the following plot:



You can perform basic editing for all the text you have entered that describe the plot. Basic editing includes altering the font and font size, colors, or any other tweaks that you might need for the plot to be appealing.

Following the example above, we can further tweak the title as follows:

Input

```
plt.axis([0,5,0,20])  
:  
: plt.title('My first plot',fontsize=18,fontname='Comic Sans MS')  
:  
: plt.xlabel('Counting',color='black')
```

```
: plt.ylabel('Square values',color='black')  
  
: plt.plot([1,2,3,4],[1,4,9,16],'ro')
```

The Matplotlib functionality allows you to perform more edits to the chart. For example, you can introduce new text into the chart using the `text ()` function, `text(x,y,s, fontdict=None, **kwargs)`.

In the function outlined above, the coordinates `x` and `y` represent the location of the text you are introducing into the chart. `s`, represents the string of text you are adding to the chart at the specified location. The `fontdict()` function represents the font you use for the new text. However, this function is optional. Once you have these figured out, you can then introduce keywords into the code. Let's have a look at the example below to illustrate this:

Input

```
plt.axis([0,5,0,20])  
  
: plt.title('My first plot',fontsize=20,fontname='Times New Roman')  
...: plt.xlabel('Counting',color='gray')  
  
: plt.ylabel('Square values',color='gray')  
  
: plt.text(1,1.4,'First')  
  
: plt.text(2,4.4,'Second')  
  
: plt.text(3,9.4,'Third')  
  
: plt.text(4,16.4,'Fourth')  
  
: plt.plot([1,2,3,4], [1,4,9,16],'ro')
```

Matplotlib is specifically built to help you introduce mathematical expressions into your work using the LaTeX expressions. When keyed in

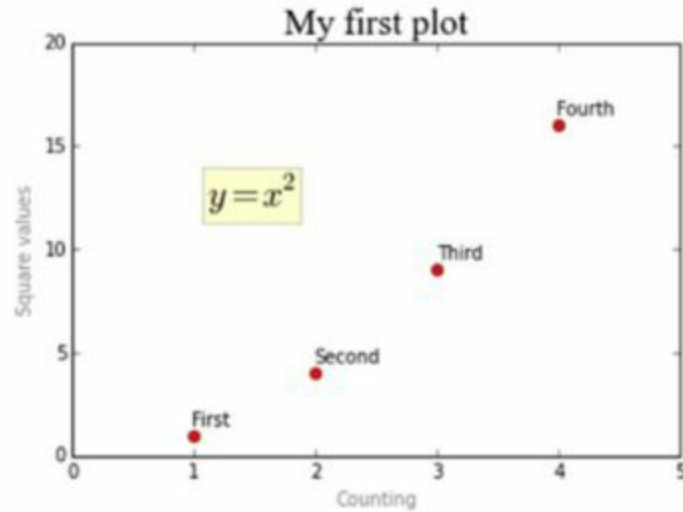
correctly, the interpreter will recognize the expressions and aptly convert them into the necessary expression graphic. This is how to introduce formula, expressions, or other unique characters into your plot.

When writing LaTeX expressions, remember to use an *r* before the expression so that the interpreter can read it as raw text.

Input

```
plt.axis([0,5,0,20])
: plt.title('My first plot',fontsize=20,fontname='Times New Roman')
...: plt.xlabel('Counting',color='gray')
: plt.ylabel('Square values',color='gray')
: plt.text(1,1.4,'First')
: plt.text(2,4.4,'Second')
: plt.text(3,9.4,'Third')
: plt.text(4,16.4,'Fourth')
: plt.text(1.1,12,r'$y = x^2$',fontsize=20,bbox={'facecolor':'yellow',
'alpha':0.2})
: plt.plot([1,2,3,4], [1,4,9,16],'ro')
```

Your plot should have a $y=x^2$ expression in a yellow background as shown below:



How to add a grid to a chart

More often, you can go online and create charts that allow you to automatically add or remove grids. You can do this in Python, too. A grid is important in your work because it shows you the position of all the points plotted on the chart. To add a grid, introduce the `grid()` function as shown below, passing it as `true`.

Input

```
plt.axis([0,5,0,20])
```

```
: plt.title('My first plot',fontsize=20,fontname='Times New Roman')
```

```
...: plt.xlabel('Counting',color='gray')
```

```
: plt.ylabel('Square values',color='gray')
```

```
: plt.text(1,1.4,'First')
```

```
: plt.text(2,4.4,'Second')
```

```
: plt.text(3,9.4,'Third')
```

```
: plt.text(4,16.4,'Fourth')
```

```
: plt.text(1.1,12,r'$y = x^2$',fontsize=20,bbbox={'facecolor':'yellow',  
'alpha':0.2})  
: plt.grid(True)  
: plt.plot([1,2,3,4],[1,4,9,16],'ro')
```

If you want to do away with the grid, you should plot the condition as false as shown below:

Input

```
plt.axis([0,5,0,20])  
: plt.title('My first plot',fontsize=20,fontname='Times New Roman')  
...: plt.xlabel('Counting',color='gray')  
: plt.ylabel('Square values',color='gray')  
: plt.text(1,1.4,'First')  
: plt.text(2,4.4,'Second')  
: plt.text(3,9.4,'Third')  
: plt.text(4,16.4,'Fourth')  
: plt.text(1.1,12,r'$y = x^2$',fontsize=20,bbbox={'facecolor':'yellow',  
'alpha':0.2})  
: plt.grid(False)  
: plt.plot([1,2,3,4],[1,4,9,16],'ro')
```

Chapter 7

Testing Hypotheses with SciPy

Hypothesis testing is one of the statistical methods that can be used in data analysis, in the process helping the analyst make useful and statistical decisions about the datasets they are using. In hypothesis testing, the concept is to make an assumption about something, then use data to determine whether this assumption is true or false. For example, you could have a premise that the average age for students in your class is 25 years old. From this premise, you use data to ascertain whether the assumption is true or not.

Hypothetical assumptions are theoretical in nature, but you must prove them using some statistical information. Proof of true or false will depend on the result of some mathematical computation.

Fundamentals of Hypothesis Testing

Hypothesis testing is one of the most important statistical methods you can use when analyzing data. This process estimates mutually exclusive events concerning a given population under study, then determines which of the statements is actually backed by the data available to the analyst.

Based on the analysis, you can then conclude that a given finding is significant statistically, after passing your hypothesis test. Hypothesis tests are built around standard normalization and normalization. These are the core concepts that any hypothesis will be built around.

Normalization in statistics refers to the process of analyzing and adjusting

values under observation to ensure that they are within a common scale before you can apply other statistical measures to the data, like averaging.

In a normal distribution, variables have the shape of a normal curve. A graph representing a normal distribution is referred to as a normal curve. In a normal curve, the following three parameters must be equal: mode, median and mean.

The formula for a normal distribution is as follows:

$$x_{\text{new}} = x - x_{\text{min}} / (x_{\text{max}} - x_{\text{min}})$$

In a standard normal distribution, you have a normal distribution, but the standard deviation = 1, while the mean = 0.

Null Hypothesis

During hypothesis testing, you will come across a null hypothesis. This refers to the default position which represents no relationship between the variables in question. It could also mean there is no association between the two groups. A null hypothesis therefore, is an assumption that you make out of basic knowledge of the issue at hand, and is not backed by any statistical data. For example, you could assume that a car dealership sells 20 units per month, without any credible data to support this claim.

Alternative Hypothesis

An alternative hypothesis is the statement you will use to challenge the null hypothesis. An alternative hypothesis usually contradicts the null hypothesis presented. From this statement, you have to choose whether to accept the null hypothesis as true or not, from the likelihood of the alternative hypothesis being true.

Significance level is the degree to which you will accept or reject the position of a null hypothesis. Logically, it is impossible to reject or accept any

hypothesis with 100% certainty. For this reason, the level of significance can be set at 5%. This is represented by the alpha symbol (α) and is often calculated as 5% or 0.05. From this assertion, therefore, the output you are working with should give you a 95% confidence level for you to consider it.

Errors

There are two types of errors you might encounter during hypothesis testing: type I and type II errors. If you encounter a type I error, you have to reject the null hypothesis even if it is true. This error is also represented by an alpha (α). If you are working on a normal curve, this critical region is usually referred to as the alpha region.

If you encounter a type II error, you have to accept the null hypothesis even if it is not true. This type of error is represented by a beta (β) sign. This acceptance region in a normal curve is referred to as the beta region.

The results you obtain from a hypothesis test and the decision you make whether to reject or accept the results are not forged in stone. You have to make a conscious decision in light of the results. The hypothesis test will only give you proof that the null hypothesis holds or not, and it is from that evidence that you can then make a decision.

It is important to remind you, however, that the evidence from hypothesis testing might not always be strong enough to help you make the correct decision. In light of this, you might end up with one of the errors mentioned above. We can illustrate this by plotting a diagram as shown below:

Input

```
plt.figure(figsize=(11,9))
```

```
plt.fill_between(x=np.arange(-3.9,-2,0.01),
```

```
                 y1= stats.norm.pdf(np.arange(-3.9,-2,0.01)) ,
```

```
        facecolor='red',
        alpha=0.36)
plt.fill_between(x=np.arange(-2.1,2,0.01),
                y1= stats.norm.pdf(np.arange(-2.1,2,0.01)) ,
                facecolor='white',
                alpha=0.36)
plt.fill_between(x=np.arange(2.1,4,0.01),
                y1= stats.norm.pdf(np.arange(2.1,4,0.01)) ,
                facecolor='red',
                alpha=0.49)
plt.fill_between(x=np.arange(-3.9,-2,0.01),
                y1= stats.norm.pdf(np.arange(-3.9,-2,0.01),loc=3, scale=2) ,
                facecolor='white',
                alpha=0.36)
plt.fill_between(x=np.arange(-2.1,2,0.01),
                y1= stats.norm.pdf(np.arange(-2.1,2,0.01),loc=3, scale=2) ,
                facecolor='blue',
                alpha=0.36)
plt.fill_between(x=np.arange(2.1,10,0.01),
                y1= stats.norm.pdf(np.arange(2.1,10,0.01),loc=3, scale=2),
                facecolor='white',
                alpha=0.36)
```

```
plt.text(x=-0.79, y=0.14, s= "Null Hypothesis")
```

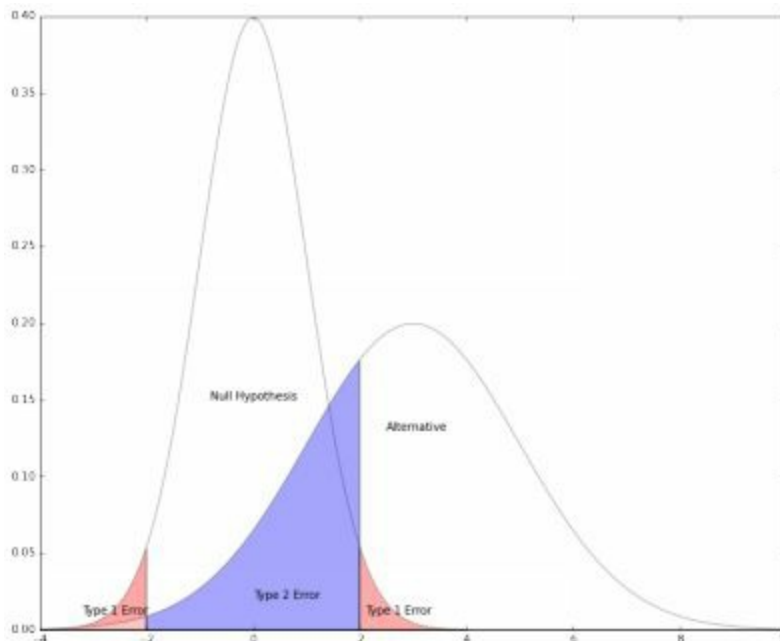
```
plt.text(x=2.4, y=0.12, s= "Alternative")
```

```
plt.text(x=2.0, y=0.01, s= "Type 1 Error")
```

```
plt.text(x=-3.1, y=0.01, s= "Type 1 Error")
```

```
plt.text(x=0, y=0.02, s= "Type 2 Error")
```

Your output should look like this



From our illustration, type I errors are represented by the red sections, proposing that the alternative hypothesis is similar to the null hypothesis if we use a two-sided test for this data, at a 95% confidence level.

The blue section of this plot shows a type II error, in the event that the null hypothesis is not similar to the alternative hypothesis. A t-test is one of the most important tools you will use in hypothesis testing to determine the difference between population and sample averages.

Hypothesis Testing Procedure

Before you begin hypothesis testing, there are specific procedures that you must follow in order to ensure that your results will be reliable.

First, you must come up with a null and alternate hypothesis. These are the statements whose validity you will be trying to prove. Remember that the main reason for testing the null hypothesis is to ascertain whether it is wrong. In the alternative hypothesis, you present a statement that indicates what you feel is untrue about the null hypothesis. The null hypothesis will in most cases be represented as H_0 , while the alternative hypothesis is represented as H_1 .

Second, you have to determine a suitable criteria for decision-making. This is based on the significance level you choose. It is based on this criteria that you will determine whether the null or alternate hypothesis holds. It is possible to have a 4% probability that the null hypothesis holds based on a 2% significance level, but at the same time, the same hypothesis is rejected based on a 5% significance level. By outlining this criteria ahead of time, you set the precedence for your work ahead of time.

Next, you have to determine the probabilities you might encounter when working on the data available. This is performed with a probability test statistic. The role of this statistic is to help you establish the likelihood of some event occurring. A higher probability means there is a high possibility that the null hypothesis will hold, based on the evidence available at your disposal.

Finally, you have to make a decision based on the results obtained. In decision-making, you will compare the results available based on the accepted significance level. In case the null hypothesis is lower than the significance level, you reject it.

It is important to take note of the possibility of accepting the wrong result when working with a sample population. This is because the sample only represents a random segment of the population. If you input data for the entire population, the results might be skewed greatly, resulting in the alternate possibility being true.

Based on the assessment above, you have four options in decision-making when it comes to evaluating the prospects of your null hypothesis. These are as follows:

- You might be right in maintaining the null hypothesis
- You might encounter a Type II error, by maintaining an incorrect null hypothesis
- You might be right in rejecting the null hypothesis
- You might encounter a Type I error, by rejecting an incorrect null hypothesis

Statistical hypothesis testing is an important part of data analysis since you will use it to determine whether the data you are analyzing conforms to a predetermined norm or not. The nature of the deviation can tell you so much about the data.

In hypothesis testing, all assumptions start from the null hypothesis. The null hypothesis predicates that there is no relationship between the variables under study. The nature of the null hypothesis depends on the kind of test you are performing. Assuming you are performing a test to determine if two groups are not similar, the null hypothesis will state that the two groups are similar.

The whole point behind a hypothesis test is to ascertain whether the null hypothesis will hold when we study a given set of data. If , after your

analysis, you determine that there is little to no evidence to refute the null hypothesis, then you would have to accept it. Just as the null hypothesis, the alternative hypothesis will also depend on the type of data you are working with.

With the alternative and null hypothesis determined, you can then set a significance level, a probability threshold that will tell you when you can accept or reject the results.

One-Sample T-Test

In this test, the goal is to determine whether the mean of a sample population is similar to the mean of the general population in the data you are studying. In the following example, we will attempt to elaborate this with fictional age data about a population of registered voters in country x, and a sample of registered voters within county y. We will test whether there is a difference between the average age of voters within one county and that of the voters within the other country.

Run the code below:

```
Input
```

```
%matplotlib inline
```

```
Input
```

```
import numpy as np
```

```
import pandas as pd
```

```
import scipy.stats as stats
```

```
import matplotlib.pyplot as plt
```

```
import math
```

Input

```
np.random.seed(6)

x_age1 = stats.poisson.rvs(loc=18, mu=35, size=150000)
x_age2 = stats.poisson.rvs(loc=18, mu=10, size=100000)
x_age = np.concatenate((x_age1, x_age2))

y_age1 = stats.poisson.rvs(loc=18, mu=30, size=30)
y_age2 = stats.poisson.rvs(loc=18, mu=10, size=20)
y_age = np.concatenate((y_age1, y_age2))

print( x_ages.mean() )

print( y_ages.mean() )

43.0

39.3
```

From the distribution above, we can perform a t-test to determine the validity of this hypothesis, given a 95% confidence level.

For this we will introduce the *stats.ttest_1samp()* function as shown below:

Input

```
stats.ttest_1samp(a= y_age,          # This is the sample population
data popmean= x_ages.mean()) # This is the main population data
```

Output

```
Ttest_1sampResult(statistic=-2.5742, pvalue=0.0132)
```

From the result above, we can tell that the value of $t = -2.5742$. This

represents how far the mean is deviating from the null hypothesis applied. If from your estimates, you notice that the t-test result is not within the t-distribution quantile associated with your confidence level, the sensible option is to reject the hypothesis altogether.

To determine the quantile, we use the *stats.t.ppf()* function as shown below:

Input

```
stats.t.ppf(q=0.025, # Check this quantile  
            df=49) # The degree of freedom
```

Output

-2.0096

Input

```
stats.t.ppf(q=0.975, # Check this quantile  
            df=49) # The degree of freedom
```

Output

-2.0096

From the illustration above, we can estimate an extreme chance of getting similar results to the results we obtained in the p-value earlier on. To do this, we must first use the t-statistic as shown below:

Input

```
stats.t.cdf(x= -2.5742, # statistic for t-test  
            df= 49) * 2 # for a two-tailed test, we multiply by 2
```

Output

0.0131

At this juncture, refer to the earlier alternative hypothesis concerning this study, whether the sample mean is not similar to the population mean. In the two-tailed test above, there is a possibility that the sample might have a negative or positive directional difference from the population mean, hence we multiply by two.

With a p-value of 0.0131, you can look forward to working with extreme data because the null hypothesis is 1.3% true. Since the significance level is higher than the p-value, the best option is to reject the null hypothesis.

With a 95% confidence level, we would not be able to attain the average population of 43.

Input

```
sigma = y_age.std()/math.sqrt(50) # sample standard deviation
stats.t.interval(0.95,           # desired confidence level
df = 49,                        # degree of freedom
loc = y_age.mean(), # sample mean
scale= sigma)                 # estimated standard deviation
```

Output

(36.3697)

From the analysis above, given a 1.3% possibility of an extreme result, at 99% confidence level this result is insignificant. This result would definitely deliver the population mean if the estimates were performed at 99% confidence level as shown below:

Input

```
stats.t.interval(alpha = 0.99,          # desired confidence level
df = 49,                                # degrees of freedom
loc = y_age.mean(), # sample mean
scale= sigma)                            # estimated standard deviation
```

Output

35.4055

If we raise the confidence level higher, we create a higher confidence interval, thereby increasing the possibility of arriving at the real mean for the population. As a result, it is highly unlikely that the null hypothesis will be rejected. From this analogy, the significance level of 1% is lower than the p-value of 1.3%, hence the null hypothesis is accepted.

Two-Sample T-Test

In a two-sample t-test, we will try to determine whether two data samples that are independent of one another are similar or not. In this test, our null hypothesis proposes that the means of the two sample groups are similar.

How is this different from a one-sample t-test? In the one-sample t-test, our study involves comparing a sample against the entire population. In the two-sample t-test, however, we are comparing one sample against another, instead of the entire population.

In this test, will use the *stats.ttest_ind()* function. We will generate another set of sample data for county m, which we shall compare against sample voter registration data for county y as outlined in the one-sample t-tests above.

Input

```
np.random.seed(12)

m_age1 = stats.poisson.rvs(loc=18, mu=33, size=30)
m_age2 = stats.poisson.rvs(loc=18, mu=13, size=20)
m_age = np.concatenate((m_age1, m_age2))

print( m_age.mean() )
```

Output

42.8

Input

```
stats.ttest_ind(a= y_age,
b= m_age,
equal_var=False) # samples share similar variance
```

Output

```
Ttest_indResult(statistic=-1.7084, pvalue=0.0907)
```

Based on the p-value derived above, there is only a 9% probability that if we compare the population means of the two samples, we will tell whether they are similar or not. In the event that we use a 95% confidence interval, the null hypothesis will hold because the 5% significance level is greater than the p-value for this data.

Paired T-Test

The tests above considered data from two population sample groups that are independent of one another. There comes a time when you have to analyze

sample data of the same group, but at different time intervals. This is to help you understand the changes that take place in the study group.

Teachers might, for example, want to tell whether students have improved their knowledge of a topic by checking their performance before and after an exercise. In such a scenario, it would be wise to use a paired t-test to determine whether the sample performance data in the same group at different intervals is similar or not.

For this study, we will use the scipy function `stats.ttest_rel()`. First, let's populate some sample performance results that we will use for this test

Input

```
np.random.seed(11)

before_exercise= stats.norm.rvs(scale=30, loc=250, size=100)

after_exercise = before_exercise + stats.norm.rvs(scale=5, loc=-1.25,
size=100)

performance_df =
pd.DataFrame({"performance_before":before_exercise,
"Performance_after":after_exercise,
"weight_change":after-before})

performance_df.describe()
```

Output

	performance_after	performance_before	performance_change
count	100.000000	100.000000	100.000000

mean	249.115171	250.345546	-1.230375
std	28.422183	28.132539	4.783696
min	165.913930	170.400443	-11.495286
25%	229.148236	230.421042	-4.046211
50%	251.134089	250.830805	-1.413463
75%	268.927258	270.637145	1.738673
max	316.720357	314.700233	9.759282

Based on the summary above, we can tell that students lost on average 1.23 points after the exercise. We can then perform a paired t-test to determine whether at 95% confidence level, this information is significant or not as shown below:

Input

stats.ttest_rel(a = before,

b = after)

Output

Ttest_relResult(statistic=2.5720, pvalue=0.01160)

From the p-value results above, we can tell that there is only a 1% chance of getting a large difference between the two population samples.

Using SciPy

You will come across a lot of data about different studies in the course of your work as a data analyst. Some of this data could be technical, others scientific depending on the objective behind the data collection methods and procedures used. Given the varied nature of data, you might also struggle to manage the data adequately. It is not very easy to perform mathematical computations on large sums of data. This is why today we have large supercomputers specifically for this task.

One of the easiest ways of handling technical data is through SciPy, one of the Python libraries that was specifically designed to handle such forms of data. Perhaps one of the best things about SciPy is that it is an open-source platform, so you can enjoy using it without paying a penny. There are several features relevant to data science that you will enjoy using in SciPy.

Installing SciPy

Installation instructions depend on the operating system you are using. The instructions below will assist you with installing SciPy in Python according to your specific device requirements.

You can install SciPy with pip. Pip is the basic package handler that is recognized in most of the common operating systems. Before installation with pip, you must make sure you have Python installed on your computer. Once this is done, run the following command:

```
Python - pip install -user numpy scipy
```

This will work for Windows operating systems. Alternatively, you can also install the SciPy package to a specific user directory instead of using the predetermined system directories. This is why you use the *-user* connotation.

If you are using a Mac system, the following commands will apply:

```
sudo port install py35-scipy py35-numpy
```

Sudo allows you to install and run programs with a different user's privileges, usually higher level privileges.

If you are using a Linux operating system, you can install SciPy using the following commands:

```
sudo apt-get install python-scipy python-numpy
```

Once you have SciPy installed, we can now move on to the next step.

SciPy Modules

You will come across different programs and tools that can handle scientific and mathematical operations in Python over the course of your data analysis career. One of the best things about SciPy is that you have access to so many modules which can help you perform anything from simple to complex operations. Here are some of the packages that you will be using going forward:

Package	Function
• Special function	scipy.special
• Spatial data structures and algorithms	scipy.spatial
• Sparse	scipy.sparse
• Statistics	scipy.stats
• Signal processing	scipy.signal
• Integration	scipy.integrate
• Interpolation	scipy.interpolate

- Input/output `scipy.io`
- Multidimensional image processing `scipy.ndimage`
- Linear algebra `scipy.linalg`
- Optimization `scipy.optimize`
- Fast fourier transformation `scipy.fftpack`

To begin working with SciPy, you have to import it. The procedure for importing SciPy is the same for all sub-packages. You simply replace the package `signal` with any of the other modules that you need to use. The import instructions are as shown below:

```
import numpy as np
```

```
from scipy import signal
```

Integration

The `scipy.integrate` sub-package is critical in performing numerical integration computations. The functions used in this package have been outlined in the section above.

A general-purpose integration, also known as a single integral, refers to a situation where your dataset only has one variable available between two study points. For this reason, we will use the `quad` function.

If, for example, you have the following information about some data integrals:

A function of $12x$ lying between two points, 0 and 1, a single integration will be represented as follows:

```
import scipy.integrate
```

```
f= lambda x: 12*x  
  
i = scipy.integrate.quad(f, 0, 1)  
  
print (i)
```

Your output will look like this:

```
(6.0, 6.661338147750939e-14)
```

Why have we used the lambda function in this statement? Its presence allows us freedom to choose whichever arguments to apply in the situation. However, take note that we can only have a single expression. In this case, the single expression we have is $12x$.

In the case of a double integral, you have data whose descriptive function has more than one variable, where the x argument always follows the y argument. Let's look at an example below:

```
dblquad(func, d, e, sfun, tfun[, args, ...])
```

Going by our explanation, the function above is a *dblquad* function. In this example, the y argument comes first, hence its value is between the limits d and e . On the other hand, the x argument always comes after the y argument, hence its value can be traced between the s and t limits. We can conclude, therefore, that the *dblquad* function above has defined two variables.

A double integral function, therefore, would look like this in your code:

```
import scipy.integrate  
  
f = lambda x, y : 12*x  
  
g = lambda x : 0  
  
h = lambda y : 1
```

```
i = scipy.integrate.dblquad(f, 0, 0.5, s, t)
print(i)
```

Your output will look like this:

```
(3.0, 6.661338147750939e-14)
```

From the example above, we can tell that f is the function under study, $12x$, 0 and 0.5 are the integrals that represent the y function. On the other hand, s and t are the integral representatives of the x function.

What happens when you have data that contains three variables? In this case, you will have three different functions, x , y and z . From these functions, you will have the following command:

```
tplquad(func, a, b, gfun, hfun, qfun, rfun)
```

The *tplquad* function comes in handy when you need to use three different integrals. From our understanding of the earlier examples, we can derive the following integral groups from the *tplquad* function:

- a and b
- g and h
- q and r

A triple integration in SciPy is as shown in the example below:

```
from scipy import integrate
```

```
f = lambda z, y, x: 12*x
```

```
integrate.tplquad(f, 0, 0.5, lambda x: 0, lambda x: 1, lambda x, y: 0, lambda x, y: 3)
```

Your output will look like this

(9.0, 3.988124156968869e-13)

Following on from the earlier examples, this code shows us that the value of the x function is between 0 and 1, the y function between 0 and 0.5, while the z function is between 0 and 3.

Chapter 8

Data Mining in Python

X

There are a lot of analytical decisions you can make from any given dataset. The process of obtaining predictive information from these datasets is called data mining. It is one of the most challenging tasks any data scientist can sit through, but a necessary one. Data in its raw form can be interpreted in different ways. Getting the right deductions from the data is the most important thing. Each set of data contains different fragments of information, which with the right analytical process, can be used to make the right predictive decisions.

In Python, you have a lot of tools you can use for data mining purposes. This is an elaborate process that involves, among others, cleaning and organizing data. We have discussed data cleaning earlier in this book, so you can understand why it is important to ascertain the credibility and integrity of data before use.

The concept of data mining is built around studying data and using it to build a model upon which accurate generalizations about a subject can be made. Data mining is linked to a lot of other machine learning processes that rely on predictive analysis. Based on previous data, the data models can respond to a new entry and act accordingly. For example, if a system stores financial information about your accounts in the US, any transaction outside that jurisdiction would be treated with contempt, and flagged for further investigation and analysis.

There are so many instances where data mining comes in handy. From social

media studies to businesses studying and analyzing consumer preferences, data mining provides important information that can help in making major decisions.

Methods of Data Mining

It is possible to build a predictive model from different sets of data. Each dataset is created using a unique technique. The following are some of the common methods used for data mining:

- ✓ **Regression analysis**

This is the process of studying and evaluating the nature of relationships between different variables. The emphasis of such studies is usually to gauge the relationships while at the same time accounting for error reduction.

- ✓ **Cluster analysis**

In this process, the analyst studies different groups of data and tries to understand them from their unique characteristics. Each cluster is built according to specific features. Members of a cluster, therefore, are expected to have similar behaviors.

- ✓ **Data classification**

Data classification is about narrowing down data groups into unique categories. The categories must first be built according to specific instructions, then any data that meets the said instructions are moved into their respective classification. A good example of this is spam mail.

- ✓ **Analyzing outliers**

This is a process where you study the outliers to determine why they exist. Outliers usually appear in a dataset when some data goes against an established pattern. This is data that does not align along a determined plane as expected.

✓ **Correlation and association analysis**

You can also study the data to determine whether there is a relationship between different variables. In particular, you should be looking at data on variables whose relationship might not be explicit. A good example is the Walmart beer-diaper case study. They realized a correlation between beer and diaper sales on Friday evenings. These are two products that should ideally not share any relationship. However, upon prodding further, it emerged that the purchases were related because most of the men who purchased diapers were young or new fathers. While picking up the diapers, they figured they might as well grab a few beers to enjoy at home.

Building a Regression Model

The first thing you think about before building a regression model is the type of solution you are looking for. Each problem is different, hence a different approach is necessary. In the examples below, we will be using [King County House Sales dataset](#). From this data, we will attempt to identify the relationship between different variables presented in the dataset. The information available online includes data on houses in King County, including their unique features and prices.

From the data available, we will try to determine the relationship between the price of a house and the unique features like the size.

Before you begin, install Jupyter on your device. Jupyter is an iPython processor. You will need the Anaconda distribution to make work easier, because it includes Jupyter, Python, and a lot of other libraries that will be useful in data analysis and scientific computing.

Download the latest Anaconda version for Python and follow the installation instructions. Once you are done, run the following:

```
jupyter notebook
```

This will initiate the notebook server. You will see some information about the server in the terminal you are using. The default web application URL should be <http://localhost:8888>

The notebook dashboard should show information on the sub-directories, files, and notebooks where your server is running.

For an experienced Python programmer, you can install Jupyter through pip instead of using Anaconda. Before you proceed, make sure you are running the current pip version. This is important because earlier versions might struggle to process some of the dependencies you will be using going forward.

```
pip3 install --upgrade pip
```

To Install the notebook:

```
pip3 install jupyter
```

You will need Pandas to help in data restructuring and cleaning. Pandas is ideal for this task because you can use it to import data from different file formats, organize, and manipulate it to suit your needs.

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
import scipy.stats as stats
```

```
import seaborn as sns
```

```
from matplotlib import rcParams
```



```
%matplotlib inline
```

```
%pylab inline
```

From the code above, we have imported Matplotlib, NumPy and SciPy, tools that will be useful in data visualization, scientific computation, and statistical computations in data analysis going forward.

One of the first things you have to do for any dataset is to study it and determine whether it needs cleaning and to what extent.

```
df = pd.read_csv('/Users/Admin/Desktop/kc_house_data.csv')  
  
df.head()
```

You should have the following output:

	id	date	price	bedrooms	bathrooms	sqft_
0	7129300520	20141013T000000	221900.0	3	1.00	1180
1	6414100192	20141209T000000	538000.0	3	2.25	2570
2	5631500400	20150225T000000	180000.0	2	1.00	770
3	2487200875	20141209T000000	604000.0	4	3.00	1960
4	1954400510	20150218T000000	510000.0	3	2.00	1680

Next, we need to determine whether we have any null values in the data as follows:

```
df.isnull().any()
```

Output

```
Id          False
Date        False
Price       False
bedrooms    False
bathrooms   False
sqft_living  False
sqft_lot    False
```

```
dtype: bool
```

The next step is to determine the types of data available in each variable. This is important so that you know whether you are working with numerical data or not.

```
df.dtypes
```

Output

```
Id          int64
Date        object
Price       float64
Bedrooms    int64
Bathrooms   float64
```

sqft_living int64

sqft_lot int64

...

dtype: object

One of the most important steps is to ensure your data is processed properly in Pandas regardless of the data file you are using. Since we are performing a regression analysis, it is important to ensure the data you have is applicable, hence the function *df.isnull().any()*.

This is important, because most of the time you will come across columns that contain different types of data from strings to integers. For regression purposes, all the data in a column should be appropriate. You can expect to come across data that is not properly organized, so understanding these functions is very important.

Next, you need to study the shape of data. From a glance, you can tell whether the distribution is credible or not. Some data might be corrupted, so it is wise to check and make sure your data is actionable.

To see all the variables you are working with, use the *df.describe()* function. After that, use the *plt.pyplot.hist()* function to see all the variables on a histogram.

`df.describe()`

Output

	price	bedrooms	bathrooms	sqft_living
count	21613	21613	21613	21613

mean	540088.10	3.37	2.11	2079.90
std	367127.20	0.93	0.77	918.44
min	75000.00	0.00	0.00	290.00
25%	321950.00	3.00	1.75	1427.00
50%	450000.00	3.00	2.25	1910.00
75%	645000.00	4.00	2.50	2550.00
max	770000.00	33.00	8.00	13540.00

From this dataset, we can tell that we are working with 21,613 observation points. You can also tell that the prices of houses are as follows:

- Mean \$540,000
- Median \$450,000

The size of each house is roughly 2080 square feet.

To determine the distribution in terms of the price of the houses and the size in square feet, we can plot the data above in a histogram using the code below:

```
fig = plt.figure(figsize=(12, 6))
sqft = fig.add_subplot(121)
cost = fig.add_subplot(122)
sqft.hist(df.sqft_living, bins=80)
sqft.set_xlabel('Ft^2')
sqft.set_title("Histogram of House Square Footage")
```

```
cost.hist(df.price, bins=80)

cost.set_xlabel('Price ($)')

cost.set_title("Histogram of Housing Prices")

plt.show()
```

You should have two histogram distributions, one for the housing prices and another for the square footage of the houses.

The data distribution is skewed to the right. From here, we can perform a regression analysis because we already have an idea of what the data should look like.

You will then import statsmodels to help you determine the estimator function using least squares as shown below:

```
import statsmodels.api as sm

from statsmodels.formula.api import ols
```

To perform a linear regression analysis in a case where you only have two variables, the following function applies:

```
Reg = ols('Dependent variable ~ independent variable(s),
dataframe).fit()

print(Reg.summary())
```

In our example and dataset above, we will have the following function:

```
m = ols('price ~ sqft_living',df).fit()

print (m.summary())
```

This model returns a summary with all the important information you need about the data, such as standard error, correlation coefficients and t-statistics.

In our example above, you will realize a significant relationship between the two variables, because of a high t-value (144.920). Another important point in the result is the $P > |t|$ value that returns 0%. From these two results, we can deduce that there is a near-zero possibility that the relationship between the two variables is as a result of chance or statistical variation.

When we look at the magnitude of the relationship between the two variables, you can also see that on average, house prices are quoted at \$28,000 more for each 1000 square-feet. Now that we have basic knowledge of the dataset, we can introduce other independent variables into the formula as follows:

```
Reg = ols('Dependent variable ~ivar1 + ivar2 + ivar3... + ivarN,  
dataframe).fit()  
  
print(Reg.summary())
```

The new summary function will be as follows:

```
m = ols('price ~ sqft_living + bedrooms + grade + condition',df).fit()  
  
print (m.summary())
```

The R-squared value following the addition of extra variables increases from 0.493 to 0.555. This is proof that when we introduce more variables, we can get a better perspective of the data.

A summary of the regression data is useful because through it, you can confirm how accurate your regression model and the data are.

Building Clustering Models

Just like we started with the regression model, the first step is to determine

the problem you need to solve by building a cluster model. Clustering is about grouping some sets of data according to predefined rules.

Within the dataset, it might not have been very clear the nature of data or objects. Therefore, it is up to you to analyze the data and create groups that share common features.

For this analysis, we will use the [Old Faithful geyser](#) data available on GitHub. This dataset only has two variables: the duration of a geyser eruption in minutes, and the interval in minutes between each eruption. When using datasets with only two variables, it is best to use a k-means cluster.

For this analysis, you need to install Scik-kit Learn. This is one of the best modules for data mining and machine learning in Python. Import the necessary modules into your notebook as follows:

```
import pandas as pd

import numpy as np

import matplotlib

import matplotlib.pyplot as plt

import sklearn

from sklearn import cluster

%matplotlib inline

faithful = pd.read_csv('/Users/Admin/Desktop/faithful.csv')

faithful.head()
```

Output

	eruptions	waiting
--	-----------	---------

0	3.600	79
1	1.800	54
2	3.333	74
3	2.283	62
4	4.533	85

This is data stored on your desktop. Next, you will check if the data is missing any values, then clean it accordingly. However, the data we are using has all the values, so there is no need for that.

```
faithful.columns = ['eruptions', 'waiting']  
plt.scatter(faithful.eruptions, faithful.waiting)  
plt.title('Old Faithful Data Scatterplot')  
plt.xlabel('Length of eruption (minutes)')  
plt.ylabel('Time between eruptions (minutes)')
```

From here, your data plot should reveal two clusters representing the two variables. However, it is not easy to tell them apart. To do this, you must introduce visualization functions to tell them apart as follows:

```
faith = np.array(faithful)  
k = 2  
kmeans = cluster.KMeans(n_clusters=k)  
kmeans.fit(faith)  
labels = kmeans.labels_
```



```
centroids = kmeans.cluster_centers_
```

Since we only have two variables, we will use $k=2$. *kmeans* in this function represents an output from your cluster module. To differentiate the two scatter variables, we need to use two different scatter plot colors as shown in the code below:

```
# only choose observations of cluster label == i
ds = faith[np.where(labels==i)]

# plotting observations
plt.plot(ds[:,0],ds[:,1], 'o', markersize=7)

# plotting centroids
lines = plt.plot(centroids[i,0],centroids[i,1], 'kx')

# enlarge the centroid x's
plt.setp(lines,ms=15.0)
plt.setp(lines,mew=4.0)

plt.show()
```

Your result should have a clear distinction of the clusters in two separate colors. You can introduce different colors for different clusters if you add more variables to your dataset.

There are many data mining techniques that you can learn which will come in handy when analyzing different kinds of data. You should know the right analysis method to use for each type of data, because some data is very specific on the analytical methods you can use.

Conclusion

Data analysis plays an important role in many aspects of life today. From the moment you wake up, you interact with data at different levels. A lot of important decisions are made based on data analytics. Companies need this data to help them meet many of their goals. As the population of the world keeps growing, their customer base keeps expanding. In light of this, it is important that they find ways of keeping their customers happy while at the same time meeting their business goals.

Given the nature of competition in the business world, it is not easy to keep customers happy. Competitors keep preying on each other's customers, and those who win have another challenge ahead - how to maintain the customers lest they slide back to their former business partners. This is one area where data analysis comes in handy.

In order to understand their customers better, companies rely on data. They collect all manner of data at each point of interaction with their customers. This data is useful in several ways. The companies learn more about their customers, thereafter clustering them according to their specific needs. Through such segmentation, the company can attend to the customers' needs better, and hope to keep them satisfied for longer.

But, data analytics is not just about customers and the profit motive. It is also about governance. Governments are the biggest data consumers all over the world. They collect data about citizens, businesses, and every other entity that they interact with at any given point. This is important information

because it helps in a lot of instances.

For planning purposes, governments need accurate data on their population so that funds can be allocated accordingly. Equitable distribution of resources is something that cannot be achieved without proper data analysis. Other than planning, there is also the security angle. To protect the country, the government must maintain different databases for different reasons. You have high profile individuals who must be accorded special security detail, top threats who must be monitored at all times, and so forth. To meet the security objective, the government has to obtain and maintain updated data on the persons of interest at all times.

There is so much more to data analysis than the corporate and government decisions. As a programmer, you are venturing into an industry that is challenging and exciting at the same time. Data doesn't lie, unless of course it is manipulated to, in which case you need insane data analysis and handling skills. As a data analyst, you will come across many challenges and problems that need solutions which can only be handled through data analysis. The way you interact with data can make a big difference, bigger than you can imagine.

There are several tools you can use for data analysis. Many people use Microsoft Excel for data analysis and it works well for them. However, there are limitations of using Excel which you can overcome through Python. Learning Python is a good initiative, given that it is one of the easiest programming languages. It is a high-level programming language because its syntax is so close to the normal language we use. This makes it easier for you to master Python concepts.

For expert programmers, you have gone beyond learning about the basics of Python and graduated into using Python to solve real-world problems. There

are many problems that can be solved through data analysis. The first challenge is usually understanding the issue at hand, then working on a data solution for it.

This book follows a series of elaborate books that introduced you to data analysis using Python. There are some important concepts that have been reiterated since the beginning of the series to help you remember the fundamentals. Knowledge of Python libraries is indeed important. It is by understanding these libraries that you can go on to become an expert data analyst with Python.

As you interact with data, you do understand the importance of cleaning data to ensure the outcome of your analysis is not flawed. You will learn how to go about this, and build on that to make sure your work is perfect. Another challenge that many organizations have is protecting the integrity of data. You should try and protect your organization from using contaminated data. There are procedures you can put in place to make sure that you use clean data all the time.

We live in a world where data is at the center of many things we do. Data is produced and stored in large amounts daily from automated systems. Learning data analysis through Python should help you process and extract information from data and make meaningful conclusions from them. One area where these skills will come in handy is forecasting. Through data analysis, you can create predictive models that should help your organization meet its objectives.

A good predictive model is only as good as the quality of data introduced into it, the data modeling methods, and more importantly the dataset used for the analysis. Beyond data handling and processing, one other important aspect of data analysis is visualization. Visualization is about presentation. Your data

model should be good enough for an audience to read and understand it at the first point of contact. Apart from the audience, you should also learn how to plot data on different visualizations to help you get a rough idea of the nature of data you are working with.

When you are done with data analysis, you should have a data model complete with visual concepts that will help in predicting outcomes and responses before you can proceed to the testing phase. Data analysis is a study that is currently in high demand in different fields. Knowing what to do, as well as when and how to handle data, is an important skill that you should not take for granted. Through this, you can build and test a hypothesis and go on to understand systems better.

[1]

[2]

[3]

[4]

[5]

[6]

[7]

[8] Safari books online. NumPy Basics: Arrays and Vectorized Computation. (2019). <https://www.oreilly.com/library/view/python-for-data/9781491957653/ch04.html>

[9] Safari books online. NumPy Basics: Arrays and Vectorized Computation. (2019). <https://www.oreilly.com/library/view/python-for-data/9781491957653/ch04.html>