INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

Lecture Notes:

Statistical Foundations of Data Science(ACAC07)

Drafted by :
Ms. K. Anjali (IARE 10959)
Assistant Professor

Department Of Computer Science and Engineering (AI & ML)
Institute of Aeronautical Engineering

September 27, 2023

Contents

List of Figures

Abbreviations

TVC	Thrust Vector Control
LOX	Liquid OXygen
LVDT	Liquid Propellant Rocket Engine
RC	Reinforced Concrete

Symbols

D^{el}	Elasticity tensor
σ	Stress tensor
ε	Strain tensor
V_{eq}	Equivalent velocity
$\dot{m}$	Mass flow rate
I_{sp}	Specific Impulse
c	Effective exhaust velocity
I_t	Total impulse
υ	Exhaust velocity
m_p	Propellant mass
m_e	Empty mass
A_{ex}	Exit Area
p_{ex}	Exhaust pressure
P_{SL-a}	Ambient pressure at sea level
F_{SL-a}	Sea level thrust of the rocket
$\dot{W_{sp}}$	Specific propellant consumption rate
C_w	Weight flow coefficient
C_f	Thrust coefficient

Chapter 1

Flavors of Data

Course Outcomes

After successful completion of this module, students should be able to:

CO 1	Recall the categories and levels of data using steps involved in data science.	Remember

1.1 What is Data science?

Data is a collection of facts, such as numbers, words, measurements, observations or just descriptions of things. Data science is the art and science of acquiring knowledge through data. Data science is the field of applying advanced analytics techniques and scientific principles to extract valuable information from data for business decision-making, strategic planning and other uses. Data science is all about how we take data, use it to acquire knowledge, and then use that knowledge to do the following:

- Make decisions
- Predict the future
- Understand the past/present
- Create new industries/products.

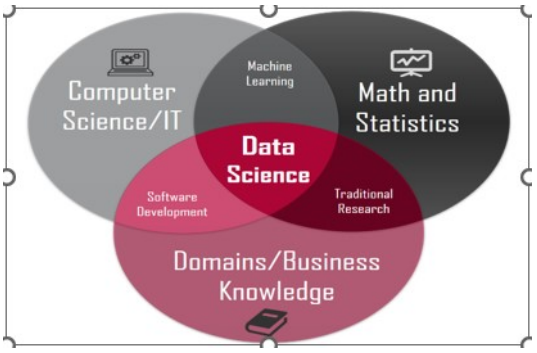


Figure 1.1: Structure of Data Science

The figure 1 shows the structure of data science with the combination of software development, business analytics, business, machine learning, computer science and statistical mathematics. Every part has their own role.

1.2 Why Data Science?

In this data age, it's clear that we have a surplus of data. But why should that necessitate an entire new set of vocabulary? What was wrong with our previous forms of analysis? For one, the sheer volume of data makes it literally impossible for a human to parse it in a reasonable time. Data is collected in various forms and from different sources, and often comes in very unorganized. Data can be missing, incomplete, or just flat out wrong. Often, we have data on very different scales and that makes it tough to compare it.
Consider that we are looking at data in relation to pricing used cars. One characteristic of a car being the year it was made and another might be the number of miles on that car. Once we clean our data (which we spend a great deal of time looking at in this book), the relationships between the data become more obvious, and the knowledge that was once buried deep in millions of rows of data simply pops out. One of the main goals of data science is to make explicit practices and procedures to discover and apply these relationships in the data. Earlier, we looked at data science in a more historical perspective, but let's take a minute to discuss its role in business today, through a very simple example.

1.3 The Data Science Venn Diagram

It is a common misconception that only those with a PhD or geniuses can understand the math/programming behind data science. This is absolutely false. Understanding data science begins with three basic areas:

1. Math/statistics: This is the use of equations and formulas to perform analysis
2. Computer programming: This is the ability to use code to create outcomes on the computer
3. Domain knowledge: This refers to understanding the problem domain (medicine, finance, social science, and so on).

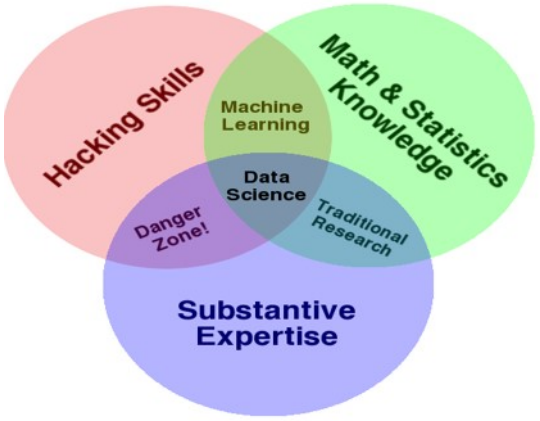


Figure 1.2: Venn Diagram of Data Science

The figure 1.2 provides a visual representation of how the three areas of data science intersect: The Venn diagram of data science. Those with hacking skills can conceptualize and program complicated algorithms using computer languages. Having a Math & Statistics Knowledge base allows you to theorize and evaluate algorithms and tweak the existing procedures to fit specific situations. Having Substantive Expertise (domain expertise) allows you to apply concepts and results in a meaningful and effective way. While having only two of these three qualities can make you intelligent, it will also leave a gap. Consider that you are very skilled in coding and have formal training in day trading.

You might create an automated system to trade in your place but lack the math skills to evaluate your algorithms and, therefore, end up losing money in the long run. It is

only when you can boast skills in coding, math, and domain knowledge that you can truly perform data science. The one that was probably a surprise for you was Domain Knowledge. It is really just knowledge of the area you are working in. If a financial analyst started analyzing data about heart attacks, they might need the help of a cardiologist to make sense of a lot of the numbers. Data Science is the intersection of the three key areas mentioned earlier.

In order to gain knowledge from data, we must be able to utilize computer programming to access the data, understand the mathematics behind the models we derive, and above all, understand our analyses' place in the domain we are in. This includes the presentation of data. If we are creating a model to predict heart attacks in patients, is it better to create a PDF of information or an app where you can type in numbers and get a quick prediction? All these decisions must be made by the data scientist.

1.4 Flavors of data

In the field, it is important to understand the different flavors of data for several reasons. Not only will the type of data dictate the methods used to analyze and extract results, knowing whether the data is unstructured or perhaps quantitative can also tell you a lot about the real-world phenomenon being measured.

We will look at the three basic classifications of data:

1. Structured vs unstructured (sometimes called organized or unorganized)
2. Quantitative vs qualitative
3. The four levels of data

1.4.1 Structured versus Unstructured Data

The distinction between structured and unstructured data is usually the first question you want to ask yourself about the entire dataset. The answer to this question can mean the difference between needing three days or three weeks of time to perform a proper analysis. The basic breakdown is as follows (this is a rehashed definition of organized and unorganized data in the first chapter):

1. Structured (organized) data: This is data that can be thought of as observations and characteristics. It is usually organized using a table method (rows and columns).

2. Unstructured (unorganized) data: This data exists as a free entity and does not follow any standard organization hierarchy.

Here are a few examples that could help you differentiate between the two:
• Most data that exist in text form, including server logs and Facebook posts, is unstructured
• Scientific observations, as recorded by careful scientists, are kept in a very neat and organized (structured) format
• A genetic sequence of chemical nucleotides (for example, ACGTATTGCA) is unstructured even if the order of the nucleotides matters as we cannot form descriptors of the sequence using a row/column format without taking a further look
Structured data is generally thought of as being much easier to work with and analyze. Most statistical and machine learning models were built with structured data in mind and cannot work on the loose interpretation of unstructured data. The natural row and column structure is easy to digest for human and machine eyes. So why even talk about unstructured data? Because it is so common! Most estimates place unstructured data as 80-90% of the world's data. This data exists in many forms and for the most part, goes unnoticed by humans as a potential source of data. Tweets, e-mails, literature, and server logs are generally unstructured forms of data.

1.4.2 Quantitative versus Qualitative Data

When you ask a data scientist, "what type of data is this?", they will usually assume that you are asking them whether or not it is mostly quantitative or qualitative. It is likely the most common way of describing the specific characteristics of a dataset.
For the most part, when talking about quantitative data, you are usually (not always) talking about a structured dataset with a strict row/column structure (because we don't assume unstructured data even has any characteristics). All the more reason why the pre-processing step is so important.
These two data types can be defined as follows:

1. Quantitative data: This data can be described using numbers, and basic mathematical procedures, including addition, are possible on the set.

2. Qualitative data: This data cannot be described using numbers and basic mathematics. This data is generally thought of as being described using "natural" categories and language.

   Example – coffee shop data

   Say that we were processing observations of coffee shops in a major city using the following five descriptors (characteristics):

   Data: Coffee Shop

   - Name of coffee shop
   - Revenue (in thousands of dollars)
   - Zip code
   - Average monthly customers
   - Country of coffee origin

   Each of these characteristics can be classified as either quantitative or qualitative, and that simple distinction can change everything. Let's take a look at each one:

   - Name of coffee shop – Qualitative

   The name of a coffee shop is not expressed as a number and we cannot perform math on the name of the shop.

   - Revenue – Quantitative

   How much money a cafe brings in can definitely be described using a number. Also, we can do basic operations such as adding up the revenue for 12 months to get a year's worth of revenue.

   - Zip code – Qualitative

   This one is tricky. A zip code is always represented using numbers, but what makes it qualitative is that it does not fit the second part of the definition of quantitative—we cannot perform basic mathematical operations on a zip code. If we add together two zip codes, it is a nonsensical measurement. We don't necessarily get a new zip code and we definitely don't get "double the zip code".

   - Average monthly customers – Quantitative

   Again, describing this factor using numbers and addition makes sense. Add up all of your monthly customers and you get your yearly customers.

- Country of coffee origin – Qualitative

We will assume this is a very small café with coffee from a single origin. This country is described using a name (Ethiopian, Colombian), and not numbers

1.4.3 The four levels of data

It is generally understood that a specific characteristic (feature/column) of structured data can be broken down into one of four levels of data. The levels are:

(a) The Nominal Level

(b) The Ordinal Level

(c) The Interval Level

(d) The Ratio Level

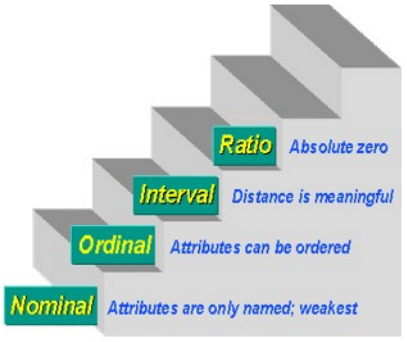


Figure 1.3: Levels of Data

As we move down the list, we gain more structure and, therefore, more returns from our analysis. Each level comes with its own accepted practice in measuring the center of the data. We usually think of the mean/average as being an acceptable form of center, however, this is only true for a specific type of data.

1.4.3.1 The Nominal Level:

The first level of data, the nominal level, (which also sounds like the word name) consists of data that is described purely by name or category. Basic examples include gender, nationality, species, or yeast strain in a beer. They are not described by numbers and are therefore qualitative. The following are some examples:

• A type of animal is on the nominal level of data. We may also say that if you are a chimpanzee, then you belong to the mammalian class as well.
• A part of speech is also considered on the nominal level of data.
Mathematical operations allowed: We cannot perform mathematics on the nominal level of data except the basic equality and set membership functions, as shown in the following two examples: • Being a tech entrepreneur is the same as being in the tech industry, but not vice versa • A figure described as a square falls under the description of being a rectangle, but not vice versa.
Measures of center: A measure of center is a number that describes what the data tends to. It is sometimes referred to as the balance point of the data. Common examples include the mean, median, and mode. In order to find the center of nominal data, we generally turn to the mode (the most common element) of the dataset. For example, look back at the WHO alcohol consumption data. The most common continent surveyed was Africa, making that a possible choice for the center of the continent column. Measures of center such as the mean and median do not make sense at this level as we cannot order the observations or even add them together.
What data is like at the nominal level: Data at the nominal level is mostly categorical in nature. Because we generally can only use words to describe the data, it can be lost in translation among countries, or can even be misspelled.
While data at this level can certainly be useful, we must be careful about what insights we may draw from them. With only the mode as a basic measure of center, we are unable to draw conclusions about an average observation. This concept does not exist at this level. It is only at the next level that we may begin to perform true mathematics on our observations.

1.4.3.2 Ordinal Level:

The nominal level did not provide us with much flexibility in terms of mathematical operations due to one seemingly unimportant fact—we could not order the observations in any natural way. Data in the ordinal level provides us with a rank order, or the means to place one observation before the other; however, it does not provide us with relative differences between observations, meaning that while we may order the observations from

first to last, we cannot add or subtract them to get any real meaning.
Examples, The Likert is among the most common ordinal level scales. Whenever you are given a survey asking you to rate your satisfaction on a scale from 1 to 10, you are providing data at the ordinal level. Your answer, which must fall between 1 and 10, can be ordered: eight is better than seven while three is worse than nine. However, differences between the numbers do not make much sense. The difference between a seven and a six might be different than the difference between a two and a one.

1.4.3.3 Interval Level:

Now we are getting somewhere interesting. At the interval level, we are beginning to look at data that can be expressed through very quantifiable means, and where much more complicated mathematical formulas are allowed. The basic difference between the ordinal level and the interval level is, well, just that—difference. Data at the interval level allows meaningful subtraction between data points. Example Temperature is a great example of data at the interval level. If it is 100 degrees Fahrenheit in Texas and 80 degrees Fahrenheit in Istanbul, Turkey, then Texas is 20 degrees warmer than Istanbul. This simple example allows for so much more manipulation at this level than previous examples.

1.4.3.4 The Ratio Level:

Finally, we will take a look at the ratio level. After moving through three different levels with differing levels of allowed mathematical operations, the ratio level proves to be the strongest of the four. Not only can we define order and difference, the ratio level allows us to multiply and divide as well. This might seem like not much to make a fuss over but it changes almost everything about the way we view data at this level. Examples, While Fahrenheit and Celsius are stuck in the interval level, the Kelvin scale of temperature boasts a natural zero. A measurement zero Kelvin literally means the absence of heat. It is a non-arbitrary starting zero. We can actually scientifically say that 200 Kelvin is twice as much heat as 100 Kelvin.

1.5 The five steps of data science

The five essential steps to perform data science are as follows:

1. Asking an interesting question
2. Obtaining the data
3. Exploring the data
4. Modelling the data
5. Communicating and visualizing the result

1.5.1 Ask an interesting question

This is probably my favorite step. As an entrepreneur, I ask myself (and others) interesting questions every day. I would treat this step as you would treat a brainstorming session. Start writing down questions regardless of whether or not you think the data to answer these questions even exists. The reason for this is twofold. First off, you don't want to start biasing yourself even before searching for data. Secondly, obtaining data might involve searching in both public and private locations and, therefore, might not be very straightforward. You might ask a question and immediately tell yourself "Oh, but I bet there's no data out there that can help me," and cross it off your list. Don't do that! Leave it on your list.

1.5.2 Obtain the data

Once you have selected the question you want to focus on, it is time to scour the world for the data that might be able to answer that question. As mentioned before, the data can come from a variety of sources; so, this step can be very creative.

1.5.3 Explore the data

Once we have the data, we use the lessons learned in Chapter 2, Types of Data, of this book and begin to break down the types of data that we are dealing with. This is a pivotal step in the process. Once this step is completed, the analyst generally has spent several

hours learning about the domain, using code or other tools to manipulate and explore the data, and has a very good sense of what the data might be trying to tell them.

1.5.4 Model the data

This step involves the use of statistical and machine learning models. In this step, we are not only fitting and choosing models, we are implanting mathematical validation metrics in order to quantify the models and their effectiveness.

1.5.5 Communicate and visualize the results

This is arguably the most important step. While it might seem obvious and simple, the ability to conclude your results in a digestible format is much more difficult than it seems. We will look at different examples of cases when results were communicated poorly and when they were displayed very well.

1.6 Explore the data

The process of exploring data is not defined simply. It involves the ability to recognize the different types of data, transform data types, and use code to systemically improve the quality of the entire dataset to prepare it for the modelling stage. In order to best represent and teach the art of exploration, I will present several different datasets and use the python package pandas to explore the data. Along the way, we will run into different tips and tricks for how to handle data. There are three basic questions we should ask ourselves when dealing with a new dataset that we may not have seen before. Keep in mind that these questions are not the beginning and the end of data science; they are some guidelines that should be followed when exploring a newly obtained set of data.

Chapter 2

DATA PRE-PROCESSING AND FEATURE SELECTION

Course Outcomes

After successful completion of this module, students should be able to:

CO 2	Demonstrate the data pre-processing terms for improving the quality of dataset using processes such as feature generation and feature selection	Under-stand

2.1 Introduction

As we know that, Data Science is the discipline of study which involves extracting insights from huge amounts of data by the use of various scientific methods, algorithms, and processes. To extract useful knowledge from data, Data Scientists need raw data. This Raw data is a collection of information from various outlines sources and an essential raw material of Data Scientists. It is additionally known as primary or source data. It consists of garbage, irregular and inconsistent values which lead to many difficulties. When using data, the insights and analysis extracted are only as good as the data we are using. Essentially, when garbage data is in, then garbage analysis comes out. Here Data cleaning comes into the picture, Data cleansing is an essential part of data science. Data cleaning

is the process of removing incorrect, corrupted, garbage, incorrectly formatted, duplicate, or incomplete data within a dataset. Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data pre-processing is a proven method of resolving such issues.

2.2 Data Cleaning

When working with multiple data sources, there are many chances for data to be incorrect, duplicated, or mislabeled. If data is wrong, outcomes and algorithms are unreliable, even though they may look correct. Data cleaning is the process of changing or eliminating garbage, incorrect, duplicate, corrupted, or incomplete data in a dataset. There's no such absolute way to describe the precise steps in the data cleaning process because the processes may vary from dataset to dataset.

Data cleansing, data cleansing, or data scrub is that the initiative among the general data preparation process. Data cleaning plays an important part in developing reliable answers and within the analytical process and is observed to be a basic feature of the info science basics. The motive of data cleaning services is to construct uniform and standardized data sets that enable data analytical tools and business intelligence easy access and perceive accurate data for each problem.

Figure 2.1: Process of Data Cleaning

2.2.1 Removal of unwanted observation:

This includes deleting duplicate/ redundant or irrelevant values from your dataset. Duplicate observations most frequently arise during data collection and Irrelevant observations are those that don't actually fit the specific problem that you're trying to solve.

• Redundant observations alter the efficiency by a great extent as the data repeats and may add towards the correct side or towards the incorrect side, thereby producing unfaithful results.

• Irrelevant observations are any type of data that is of no use to us and can be removed directly.

2.2.2 Fixing Structural errors :

The errors that arise during measurement, transfer of data, or other similar situations are called structural errors. Structural errors include typos in the name of features, the same attribute with a different name, mislabeled classes, i.e. separate classes that should really be the same, or inconsistent capitalization.

• For example, the model will treat America and America as different classes or values, though they represent the same value or red, yellow, and red-yellow as different classes or attributes, though one class can be included in the other two classes. So, these are some structural errors that make our model inefficient and give poor quality results.

2.2.3 Managing Unwanted outliers :

Missing data is a deceptively tricky issue in machine learning. We cannot just ignore or remove the missing observation. They must be handled carefully as they can be an indication of something important. The two most common ways to deal with missing data are:

**** Dropping observations with missing values.

• The fact that the value was missing may be informative in itself.

• Plus, in the real world, you often need to make predictions on new data even if some of the features are missing!

******Imputing the missing values from past observations.

- Again, "missingness" is almost always informative in itself, and you should tell your algorithm if a value was missing.
- Even if you build a model to impute your values, you're not adding any real information. You're just reinforcing the patterns already provided by other features.

Missing data is like missing a puzzle piece. If you drop it, that's like pretending the puzzle slot isn't there. If you impute it, that's like trying to squeeze in a piece from somewhere else in the puzzle. So, missing data is always an informative and an indication of something important. And we must be aware of our algorithm of missing data by flagging it. By using this technique of flagging and filling, you are essentially allowing the algorithm to estimate the optimal constant for missingness, instead of just filling it in with the mean.

2.3 Why data cleaning is essential?

Data cleaning is the most important task that should be done as a data science professional. Having wrong or bad quality data can be detrimental to processes and analysis. Having clean data will ultimately increase overall productivity and permit the very best quality information in your decision-making. With most industries relying on data today for their business growth, especially data-intensive industries like banking, insurance, retail, telecoms among others, managing data to be error-free becomes important. It is known that one way of achieving maximum efficiency is to reduce all kinds of data errors and inconsistencies.

If the company aims to optimize its working and increase their profits by using data, then data quality is of utmost importance. Old and inaccurate data can have an impact on results. Data quality problems can occur anywhere in information systems. These problems can be solved by using various data cleaning techniques. Data cleaning is a process used to determine inaccurate, incomplete, or unreasonable data and then improve quality by correcting detected errors and omissions.

2.4 Data Integration

Data integration is the process of combining data from different sources into a single and unified view. Integration begins with the ingestion process and uses the ETL approach that extracts, transforms, and loads data from different sources into a unified view. Data integration ultimately enables analytic tools to produce actionable, effective business intelligence. For instance, customer data integration involves extracting information about

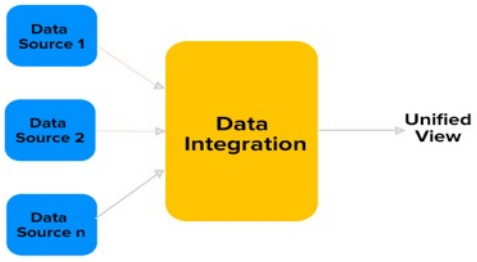


Figure 2.2: Overview of data integration

each customer from disparate business systems such as sales, marketing, and accounts, which is then combined into a single view of the customer to be used for reporting, analysis, and customer service. Data integration can consolidate all kinds of data – structured, unstructured, batch, and streaming – to do everything from basic querying of inventory databases to complex predictive analytics. Now that we know what is data integration, let's get into the benefits of using data integration in your business.

2.5 Why is data integration important?

The demand for big data integration is continually growing. As reported by Global News Wire, the data integration market is estimated to reach USD 19.6 billion by 2026 from USD 11.6 billion in 2021, growing at a CAGR of 11%. Whether an organization is looking to merge databases between partners or a government department is looking to eliminate data silos between departments, data integration systems can play an essential role in mitigating tedious data manipulation methods. So, on that note, here are a few advantages of using data integration solutions for your business.

Figure 2.3: Importance of Data integration

1. Boosts efficiency and saves time: When a company integrates techniques like data integration into their business processes, it significantly cuts down the time it takes to prepare and analyze that data. The automation of unified views also eliminates the need for gathering data manually. The time saved on these tasks can be used for analysis and execution to make an organization more productive and competitive.

2. Reduces errors: There's a lot to keep up with when it comes to a company's data resources. To gather the data manually, employees must know every location and account they might want to explore. Plus, they must install all the necessary software before they even begin to ensure their datasets will be accurate and complete. In case, a data repository is added, and that employee is unaware, they will have an incomplete dataset. When data integration is in place, your employees have access to the most accurate and real-time data whenever needed. It also reduces tedious manual work and data manipulation, thereby minimizing errors.

3. Helps in smarter business decisions: Data integration allows transparent business processes within the enterprise. Data integration systems give the flexibility to use data in different internal systems in whatever way the businesses want. This allows them to understand data clearly. Therefore, any decision regarding business processes can be made quickly and smartly.

4. Improves collaboration: Employees in every department – and sometimes in disparate physical locations – need access to the company's data for business projects. Additionally, employees in almost every department are improving and generating data that the rest of the business needs. By integrating data, everyone can access a unified view of the data

from the entire organization. This improves collaboration and unification across the organization.

5. Delivers more valuable data: Data integration process improves the value of a business' data over time. As data is integrated into a centralized repository system, quality issues are identified and improvements are implemented, resulting in more accurate data — the foundation for quality analysis.

2.5.1 Techniques used for data integration

The need for big data integration arises when data is coming in from external as well as internal sources. This is achieved using different types of data integration techniques, depending on the complexity, disparity, and volume of the data sources.

Let's get into the different types of data integration techniques one by one and see how they can help improve business intelligence processes.

1. Data consolidation: Data consolidation combines data from different sources to create a centralized data repository. This centralized repository is then used for various purposes, such as data analysis and reporting.

A key factor that differentiates data consolidation from other data integration techniques is data latency. Data latency is defined as the total time taken to retrieve data from different sources to transfer it to the data store.

A shorter latency period means fresher data is available for business intelligence and analysis in the data store.

Depending on the technologies used for data integration and the needs of your business, this latency can range from a few seconds to hours, or even more. However, with advancements in data technologies, it is possible to consolidate data and transfer changes to the destination in near real-time or real-time.

2. Data propagation: Data propagation uses applications to copy data from one location to another on an event-driven basis. Enterprise data replication (EDR) and enterprise application integration (EAI) can be used for data propagation. While EDR is more frequently used to transfer data between two databases, EAI provides a link for purposes such as business transaction processing between two systems.

3. Data federation: This technique creates a virtual database that consolidates data from

various sources. Users then use the virtual database as a single source of truth for all the data in the organization.

In this technique, data abstraction is done to create a uniform user interface for data retrieval and access. As a result, whenever an application or user queries the virtual database, the query is sent to the relevant underlying data source. In other words, the data is served on an on-demand basis in the data federation.

4. Data warehousing: This type of data integration involves using a common storage area to cleanse, format, and store data. In this technique, data from all applications across an organization is copied to the data warehouse. From here, it can be queried by data analysts.

Querying data on the warehouse instead of the source applications means that analysts don't have to worry about impacting the performance of the app or software.

Plus, data analysts can view all of the data in a central, single location, which means they can check for data's accuracy and consistency.

5. Middleware data integration: Middleware data integration involves using a middleware application between source systems and a central data repository. The middleware helps to validate data before sending it to the repository, which could be a cloud data warehouse or a database.

2.5.2 Data integration uses cases in real-world

Integrating data is the first step to uncover its true potential. When companies have all their information in one place, it becomes possible to find the most accurate and important insights within it. Data integration software is now being used by a majority of industries, majorly including healthcare, telecommunications, and retail. These industries are adopting data integration techniques to improve their business processes and this is what gives them a competitive advantage.

Here are some real use-cases of data integration that show how this technique can help businesses from different sectors and industries. 1. Healthcare: Treating patients requires utmost care and access to their treatment history. Previously, patient data used to be spread across systems, which ultimately compromised care. However, with data

Figure 2.4: Real world scenario of Data Integration

integration, hospitals have started integrating as much data as possible into a single comprehensive record. This, in turn, has resulted in improving patient outcomes, elevating health and wellness, and reducing costs.

2. Retail: Retailers deal with tons of data every single day. So, their performance tracking entirely depends on having all the relevant data in one place. Data integration empowers retailers to manage sales, inventory, and other vital metrics within their different outlets and channels smoothly.

3. Finance: The financial industry has started embracing data integration for fraud prevention and detection, measuring credit risk, maximizing cross-sell/up-sell opportunities, and retaining valuable customers.

4. Marketing: Marketing is another area where data integration has become important. It involves sending messages to the right audience at the intended time. If things are not finely tuned, marketing campaigns can go south in no time.

Managing information on thousands or potentially millions of consumers is impossible without proper integration channels and tools for data integration. It could lead to disappointing campaigns and wasted marketing budgets. Integrating data is the only way to keep it up-to-date and organized.

5. Telecommunications: Quality customer service is crucial in telecommunications, and for this data integration is important. Integrating data from a variety of sources provides a 360-degree view of company and customer relationships. Issues leading to low customer satisfaction and more customer service requests can be identified and corrected.

2.6 Data Reduction

Data mining is applied to the selected data in a large amount database. When data analysis and mining is done on a huge amount of data, then it takes a very long time to process, making it impractical and infeasible.

Data reduction techniques ensure the integrity of data while reducing the data. Data reduction is a process that reduces the volume of original data and represents it in a much smaller volume. Data reduction techniques are used to obtain a reduced representation of the dataset that is much smaller in volume by maintaining the integrity of the original data. By reducing the data, the efficiency of the data mining process is improved, which produces the same analytical results.

Data reduction does not affect the result obtained from data mining. That means the result obtained from data mining before and after data reduction is the same or almost the same.

Data reduction aims to define it more compactly. When the data size is smaller, it is simpler to apply sophisticated and computationally high-priced algorithms. The reduction of the data may be in terms of the number of rows (records) or terms of the number of columns (dimensions).

There are different steps of data reduction techniques: Dimensionally reduction, Numerosity reduction, Data cube Aggregation, Data Compression, Discretization operation.

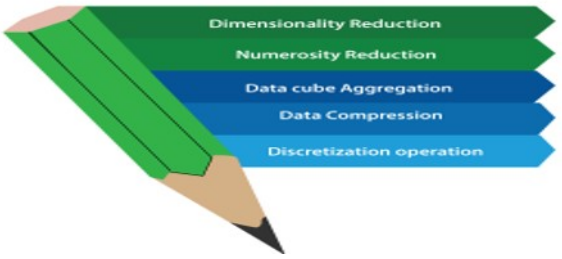


Figure 2.5: Data reduction Techniques

2.6.1 Dimensionality Reduction

Whenever we encounter weakly important data, we use the attribute required for our analysis. Dimensionality reduction eliminates the attributes from the data set under consideration, thereby reducing the volume of original data. It reduces data size as it eliminates outdated or redundant features.

2.6.2 Numerosity Reduction

The numerosity reduction reduces the original data volume and represents it in a much smaller form.

2.6.3 Data Cube Aggregation

This technique is used to aggregate data in a simpler form. Data Cube Aggregation is a multidimensional aggregation that uses aggregation at various levels of a data cube to represent the original data set, thus achieving data reduction.
For example, suppose you have the data of All Electronics sales per quarter for the year 2018 to the year 2022. If you want to get the annual sale per year, you just have to aggregate the sales per quarter for each year. In this way, aggregation provides you with the required data, which is much smaller in size, and thereby we achieve data reduction even without losing any data.

2.6.4 Data Compression

Data compression employs modification, encoding, or converting the structure of data in a way that consumes less space. Data compression involves building a compact representation of information by removing redundancy and representing data in binary form.

Data that can be restored successfully from its compressed form is called Lossless compression. In contrast, the opposite where it is not possible to restore the original form from the compressed form is Lossy compression. Dimensionality and numerosity reduction method are also used for data compression. This technique reduces the size of the files using different encoding mechanisms, such as Huffman Encoding and run-length Encoding.

2.6.5 Discretization Operation

The data discretization technique is used to divide the attributes of the continuous nature into data with intervals. We replace many constant values of the attributes with labels of small intervals. This means that mining results are shown in a concise and easily understandable way.

2.7 Data Transformation

Data transformation is the process of converting data from one format to another. The most common data transformations are converting raw data into a clean and usable form, converting data types, removing duplicate data, and enriching the data to benefit an organization. During the process of data transformation, an analyst will determine the structure, perform data mapping, extract the data from the original source, execute the transformation, and finally store the data in an appropriate database.
Transformed data is usable, accessible, and secure to benefit a variety of purposes. Organizations may transform data to make it compatible with other types of data, move it into the appropriate database, or combine it with other crucial information. Organizations benefit from transforming data by gaining insights into vital operational and informational internal and external functions. In addition, data transformation makes it possible for organizations to transform data from a storage database to the cloud to keep information moving.

2.7.1 Benefits and challenges of data transformation

Transforming data yields several benefits:

• Data is transformed to make it better-organized. Transformed data may be easier for both humans and computers to use.

• Properly formatted and validated data improves data quality and protects applications from potential landmines such as null values, unexpected duplicates, incorrect indexing, and incompatible formats.

• Data transformation facilitates compatibility between applications, systems, and types of data. Data used for multiple purposes may need to be transformed in different ways.

However, there are challenges to transforming data effectively:

• Data transformation can be expensive. The cost is dependent on the specific infrastructure, software, and tools used to process data. Expenses may include those related to licensing, computing resources, and hiring necessary personnel.

• Data transformation processes can be resource-intensive. Performing transformations in an on-premises data warehouse after loading, or transforming data before feeding it into applications, can create a computational burden that slows down other operations. If you use a cloud-based data warehouse, you can do the transformations after loading because the platform can scale up to meet demand.

• Lack of expertise and carelessness can introduce problems during transformation. Data analysts without appropriate subject matter expertise are less likely to notice typos or incorrect data because they are less familiar with the range of accurate and permissible values. For example, someone working on medical data who is unfamiliar with relevant terms might fail to flag disease names that should be mapped to a singular value or notice misspellings.

• Enterprises can perform transformations that don't suit their needs. A business might change information to a specific format for one application only to then revert the information back to its prior format for a different application.

2.7.2 How to transform data

Data transformation can increase the efficiency of analytic and business processes and enable better data-driven decision-making. The first phase of data transformations should include things like data type conversion and flattening of hierarchical data. These operations shape data to increase compatibility with analytics systems. Data analysts and data scientists can implement further transformations additively as necessary as individual layers of processing. Each layer of processing should be designed to perform a specific set of tasks that meet a known business or technical requirement.

2.7.3 Extraction and parsing

In the modern ELT process, data ingestion begins with extracting information from a data source, followed by copying the data to its destination. Initial transformations are focused on shaping the format and structure of data to ensure its compatibility with both the destination system and the data already there. Parsing fields out of comma-delimited log data for loading to a relational database is an example of this type of data transformation.

2.7.4 Translation and mapping

Some of the most basic data transformations involve the mapping and translation of data. For example, a column containing integers representing error codes can be mapped to the relevant error descriptions, making that column easier to understand and more useful for display in a customer-facing application.
Translation converts data from formats used in one system to formats appropriate for a different system. Even after parsing, web data might arrive in the form of hierarchical JSON or XML files, but need to be translated into row and column data for inclusion in a relational database.

2.7.5 Filtering, aggregation, and summarization

Data transformation is often concerned with whittling data down and making it more manageable. Data may be consolidated by filtering out unnecessary fields, columns, and records. Omitted data might include numerical indexes in data intended for graphs and dashboards or records from business regions that aren't of interest in a particular study. Data might also be aggregated or summarized. by, for instance, transforming a time series of customer transactions to hourly or daily sales counts.
BI tools can do this filtering and aggregation, but it can be more efficient to do the transformations before a reporting tool accesses the data.

2.7.6 Enrichment and imputation

Data from different sources can be merged to create denormalized, enriched information. A customer's transactions can be rolled up into a grand total and added into a customer information table for quicker reference or for use by customer analytics systems. Long or freeform fields may be split into multiple columns, and missing values can be imputed or corrupted data replaced as a result of these kinds of transformations.

2.7.7 Indexing and ordering

Data can be transformed so that it's ordered logically or to suit a data storage scheme. In relational database management systems, for example, creating indexes can improve performance or improve the management of relationships between different tables.

2.7.8 Anonymization and encryption

Data containing personally identifiable information, or other information that could compromise privacy or security, should be anonymized before propagation. Encryption of private data is a requirement in many industries, and systems can perform encryption at multiple levels, from individual database cells to entire records or fields.

2.7.9 Modeling, typecasting, formatting, and renaming

Finally, a whole set of transformations can reshape data without changing content. This includes casting and converting data types for compatibility, adjusting dates and times with offsets and format localization, and renaming schemas, tables, and columns for clarity.

2.7.10 Refining the data transformation process

Before your enterprise can run analytics, and even before you transform the data, you must replicate it to a data warehouse architected for analytics. Most organizations today choose a cloud data warehouse, allowing them to take full advantage of ELT. Stitch can load all of your data to your preferred data warehouse in a raw state, ready for transformation.

2.8 Data Discretization

The data discretization techniques can be used to reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals. Interval labels can be used to restore actual data values. It can be restoring multiple values of a continuous attribute with a small number of interval labels therefore decrease and simplifies the original information.
This leads to a concise, easy-to-use, knowledge-level representation of mining results. Discretization techniques can be categorized depends on how the discretization is implemented, such as whether it uses class data or which direction it proceeds (i.e., top-down vs. bottom-up). If the discretization process uses class data, then it can say it is supervised discretization. Therefore, it is unsupervised. If the process begins by first discovering one or a few points (known as split points or cut points) to split the whole attribute range, and then continue this recursively on the resulting intervals, it is known as top-down discretization or splitting.

In bottom-up discretization or merging, it can start by considering all of the continuous values as potential split-points, removes some by merging neighbourhood values to form intervals, and then recursively applies this process to the resulting intervals. Discretization can be implemented recursively on an attribute to support a hierarchical or multi-resolution partitioning of the attribute values, referred to as a concept hierarchy.
Concept hierarchies are useful for mining at multiple levels of abstraction. A concept hierarchy for a given numerical attribute represents a discretization of the attribute. Concept hierarchies can be used to decrease the data by collecting and restoring low-level concepts (including numerical values for the attribute age) with higher-level concepts (including youth, middle-aged, or senior). Although detail is hidden by such data generalization, the generalized data can be more meaningful and simpler to execute.
This provides a consistent description of data mining results among several mining tasks, which is a common requirement. Also, mining on a reduced data set needed fewer input/output operations and is more able than mining on a higher, ungeneralized data set. Due to these advantages, discretization techniques and concept hierarchies are generally used before data mining as a pre-processing step, rather than during mining.

2.8.0.1 What are some famous techniques of data discretization?

1. Histogram analysis: Histogram is a plot used to present the underlying frequency distribution of a set of continuous data. The histogram helps the inspection of the data for the distribution of the data. For example normal distribution representation, outliers, and skewness representation, etc.
2. Binning: Binning is a data smoothing technique and its helps to group a huge number of continuous values into a smaller number of bins. For example, if we have data about a group of students, and we want to arrange their marks into a smaller number of marks intervals by making the bins of grades. One bin for grade A, one for grade B, one for C, one for D, and one for F Grade.
3. Correlation analysis: Cluster analysis is commonly known as clustering. Clustering is the task of grouping similar objects in one group, commonly called clusters. All different objects are placed in different clusters.

4. Clustering analysis

2.9 Feature Generation and Feature Selection

In our everyday life we are faced with decisions. One of the reasons why we struggle to take a decision is because, most of the time, it involves more than one objective. For instance, when buying a car, it isn't just about buying the best car; but about buying a car that you can afford, is the right size for you, the right colour, doesn't consume too much, it's environmentally friendly etc. But each time you find a car that fulfils some of these criteria, it seems to lack on the other ones.

2.9.0.1 Feature Generation

Before we get into the details let's review what a feature is. A feature (or column) represents a measurable piece of data like name, age or gender.It is the basic building block of a dataset. The quality of a feature can vary significantly and has an immense effect on model performance. We can improve the quality of a dataset's features in the pre-processing stage using processes like Feature Generation and Feature Selection.
Feature Generation (also known as feature construction, feature extraction or feature engineering) is the process of transforming features into new features that better relate to the target. This can involve mapping a feature into a new feature using a function like log, or creating a new feature from one or multiple features using multiplication or addition.
Feature Generation can improve model performance when there is a feature interaction. Two or more features interact if the combined effect is (greater or less) than the sum of their individual effects. It is possible to make interactions with three or more features, but this tends to result in diminishing returns.
Feature Generation is often overlooked as it is assumed that the model will learn any relevant relationships between features to predict the target variable. However, the generation of new flexible features is important as it allows us to use less complex models that are faster to run and easier to understand and maintain.

2.9.0.2 Feature Selection

In fact, not all features generated are relevant. Moreover, too many features may adversely affect the model performance. This is because as the number of features increases, it becomes more difficult for the model to learn mappings between features and target (this is known as the curse of dimensionality). Thus, it is important to select the most useful features through Feature Selection, which we will further introduce in our next blog.

Examples of Feature Generation techniques

A transformation is a mapping that is used to transform a feature into a new feature. The right transformation depends on the type and structure of the data, data size and the goal. This can involve transforming single feature into a new feature using standard operators like log, square, power, exponential, reciprocal, addition, division, multiplication etc.

Often the relationship between dependent and independent variables are assumed linear, but this is not always the case. There are feature combinations that cannot be represented by a linear system. A new feature can be created based on a polynomial combination of numeric features in a dataset. Moreover, new features can be created using trigonometric combinations.

2.10 Filter Methods

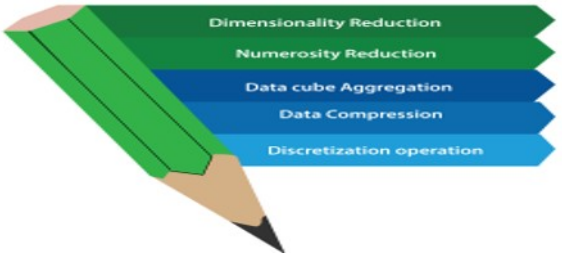


Figure 2.6: Data reduction Techniques

Filter methods are generally used as a pre-processing step. The selection of features is independent of any machine learning algorithms. Instead, features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable. The correlation is a subjective term here. For basic guidance, you can refer to the following table for defining correlation coefficients.

Feature or Response	Continuous	Categorial
Continuous	Pearsons Correlation	LDA
Categorical	Anova	Chi-Square

2.10.1 Pearsons Correlation

It is used as a measure for quantifying linear dependence between two continuous variables X and Y. Its value varies from -1 to +1. Pearson's correlation is given as:

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$$

2.10.2 LDA:

Linear discriminant analysis is used to find a linear combination of features that characterizes or separates two or more classes (or levels) of a categorical variable.

2.10.3 ANOVA:

ANOVA stands for Analysis of variance. It is similar to LDA except for the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not.

2.10.4 Chi-Square:

It is a is a statistical test applied to the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution.
One thing that should be kept in mind is that filter methods do not remove multicollinearity. So, you must deal with multicollinearity of features as well before training models for your data.

2.11 Wrapper Methods

In wrapper methods, we try to use a subset of features and train a model using them. Based on the inferences that we draw from the previous model, we decide to add or remove features from your subset. The problem is essentially reduced to a search problem. These methods are usually computationally very expensive. Some common examples of

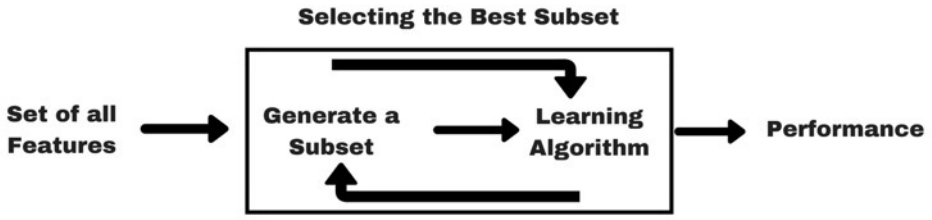


Figure 2.7: Wrapper Method

wrapper methods are forward feature selection, backward feature elimination, recursive feature elimination, etc.

• Forward Selection: Forward selection is an iterative method in which we start with having no feature in the model. In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model.

• Backward Elimination: In backward elimination, we start with all the features and removes the least significant feature at each iteration which improves the performance of the model. We repeat this until no improvement is observed on removal of features.

• Recursive Feature elimination: It is a greedy optimization algorithm which aims to find the best performing feature subset. It repeatedly creates models and keeps aside the best or the worst performing feature at each iteration. It constructs the next model with the left features until all the features are exhausted. It then ranks the features based on the order of their elimination.

One of the best ways for implementing feature selection with wrapper methods is to use Boruta package that finds the importance of a feature by creating shadow features.

1. Firstly, it adds randomness to the given data set by creating shuffled copies of all features (which are called shadow features).

2. Then, it trains a random forest classifier on the extended data set and applies a feature

importance measure (the default is Mean Decrease Accuracy) to evaluate the importance of each feature where higher means more important.

3. At every iteration, it checks whether a real feature has a higher importance than the best of its shadow features (i.e. whether the feature has a higher Z-score than the maximum Z-score of its shadow features) and constantly removes features which are deemed highly unimportant.

4. Finally, the algorithm stops either when all features get confirmed or rejected or it reaches a specified limit of random forest runs.

2.12 Decision Tree Algorithm

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data). In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

Types of Decision Tree

Types of decision trees are based on the type of target variable we have. It can be of two types:

1. Categorical Variable Decision Tree: Decision Tree which has a categorical target variable then it called a Categorical variable decision tree.

2. Continuous Variable Decision Tree: Decision Tree has a continuous target variable then it is called Continuous Variable Decision Tree.

Example:- Let's say we have a problem to predict whether a customer will pay his renewal premium with an insurance company (yes/ no). Here we know that the income of customers is a significant variable but the insurance company does not have income details for all customers. Now, as we know this is an important variable, then we can build a decision tree to predict customer income based on occupation, product, and various other variables. In this case, we are predicting values for the continuous variables.

Important Terminology related to Decision Trees

1. Root Node: It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
2. Splitting: It is a process of dividing a node into two or more sub-nodes.
3. Decision Node: When a sub-node splits into further sub-nodes, then it is called the decision node.
4. Leaf / Terminal Node: Nodes do not split is called Leaf or Terminal node.
5. Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
6. Branch / Sub-Tree: A subsection of the entire tree is called branch or sub-tree.
7. Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

Decision trees classify the examples by sorting them down the tree from the root to

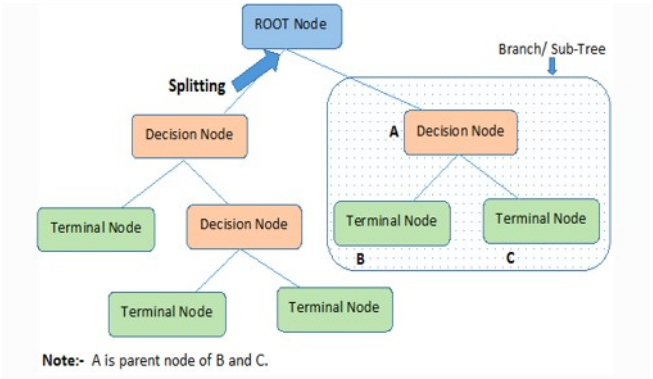


Figure 2.8: Decision tree

some leaf/terminal node, with the leaf/terminal node providing the classification of the example.

Each node in the tree acts as a test case for some attribute, and each edge descending from the node corresponds to the possible answers to the test case. This process is recursive in nature and is repeated for every subtree rooted at the new node.

Assumptions while creating Decision Tree

Below are some of the assumptions we make while using Decision tree:

- In the beginning, the whole training set is considered as the root.
- Feature values are preferred to be categorical. If the values are continuous then they

are discretized prior to building the model.

- Records are distributed recursively on the basis of attribute values.
- Order to placing attributes as root or internal node of the tree is done by using some statistical approach.

Decision Trees follow Sum of Product (SOP) representation. The Sum of product (SOP) is also known as Disjunctive Normal Form. For a class, every branch from the root of the tree to a leaf node having the same class is conjunction (product) of values, different branches ending in that class form a disjunction (sum).

The primary challenge in the decision tree implementation is to identify which attributes do we need to consider as the root node and each level. Handling this is to know as the attributes selection. We have different attributes selection measures to identify the attribute which can be considered as the root note at each level.

2.12.1 How do Decision Tree work?

The decision of making strategic splits heavily affects a tree's accuracy. The decision criteria are different for classification and regression trees.

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes. The algorithm selection is also based on the type of target variables. Let us look at some algorithms used in Decision Trees:

ID3 → (extension of D3)

C4.5 → (successor of ID3)

CART → (Classification And Regression Tree)

CHAID → (Chi-square automatic interaction detection Performs multi-level splits when computing classification trees)

MARS → (multivariate adaptive regression splines)

The ID3 algorithm builds decision trees using a top-down greedy search approach through the space of possible branches with no backtracking. A greedy algorithm, as the name

suggests, always makes the choice that seems to be the best at that moment.
Steps in ID3 algorithm:

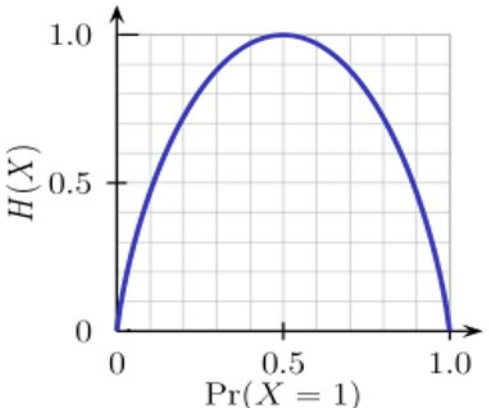
1. It begins with the original set S as the root node.
2. On each iteration of the algorithm, it iterates through the very unused attribute of the set S and calculates Entropy(H) and Information gain(IG) of this attribute.
3. It then selects the attribute which has the smallest Entropy or Largest Information gain.
4. The set S is then split by the selected attribute to produce a subset of the data.
5. The algorithm continues to recur on each subset, considering only attributes never selected before.

2.12.2 Attribute Selection Measures

If the dataset consists of N attributes then deciding which attribute to place at the root or at different levels of the tree as internal nodes is a complicated step. By just randomly selecting any node to be the root can't solve the issue. If we follow a random approach, it may give us bad results with low accuracy. For solving this attribute selection problem, researchers worked and devised some solutions. They suggested using some criteria like: Entropy, Information gain, Gini index, Gain Ratio, Reduction in Variance Chi-Square
These criteria will calculate values for every attribute. The values are sorted, and attributes are placed in the tree by following the order i.e, the attribute with a high value (in case of information gain) is placed at the root. While using Information Gain as a criterion, we assume attributes to be categorical, and for the Gini index, attributes are assumed to be continuous.

2.12.2.1 Entropy

Entropy is a measure of the randomness in the information being processed. The higher the entropy, the harder it is to draw any conclusions from that information. Flipping a coin is an example of an action that provides information that is random. From the above graph, it is quite evident that the entropy H(X) is zero when the probability is either 0



or 1. The Entropy is maximum when the probability is 0.5 because it projects perfect randomness in the data and there is no chance if perfectly determining the outcome.

ID3 follows the rule — A branch with an entropy of zero is a leaf node and A brach with entropy more than zero needs further splitting.

Mathematically Entropy for 1 attribute is represented as:

Where S → Current state, and Pi → Probability of an event i of state S or Percentage of

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

class i in a node of state S.

Mathematically Entropy for multiple attributes is represented as:

$$E(T, X) = \sum_{c \in X} P(c)E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)*__E__(3,2) + **P**(Overcast)*__E__(4,0) + **P**(Rainy)*__E__(2,3)
= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971
= 0.693

where T→ Current state and X → Selected attribute

2.12.2.2 Information Gain

Information gain or IG is a statistical property that measures how well a given attribute separates the training examples according to their target classification. Constructing a decision tree is all about finding an attribute that returns the highest information gain and the smallest entropy.

Information gain is a decrease in entropy. It computes the difference between entropy

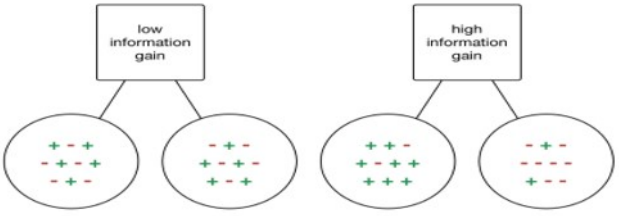


Figure 2.9: Information Gain

before split and average entropy after split of the dataset based on given attribute values. ID3 (Iterative Dichotomiser) decision tree algorithm uses information gain. Mathematically, IG is represented as:

$$\text{Information Gain}(T,X) = \text{Entropy}(T) - \text{Entropy}(T, X)$$

IG(PlayGolf, Outlook) = E(PlayGolf) - E(PlayGolf, Outlook)
= 0.940 - 0.693
= 0.247

In a much simpler way, we can conclude that:

$$Information\ Gain = Entropy(before) - \sum_{j=1}^{K} Entropy(j, after)$$

Where “before” is the dataset before the split, K is the number of subsets generated by the split, and (j, after) is subset j after the split.

2.12.2.3 Gini Index

You can understand the Gini index as a cost function used to evaluate splits in the dataset. It is calculated by subtracting the sum of the squared probabilities of each class from one. It favors larger partitions and easy to implement whereas information gain favors smaller partitions with distinct values.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Gini Index works with the categorical target variable "Success" or "Failure". It performs only Binary splits.

Higher value of Gini index implies higher inequality, higher heterogeneity.

Steps to Calculate Gini index for a split

1. Calculate Gini for sub-nodes, using the above formula for success(p) and failure(q) (p^2+q^2).

2. Calculate the Gini index for split using the weighted Gini score of each node of that split.

CART (Classification and Regression Tree) uses the Gini index method to create split points.

2.12.2.4 Gain ratio

Information gain is biased towards choosing attributes with a large number of values as root nodes. It means it prefers the attribute with a large number of distinct values.

C4.5, an improvement of ID3, uses Gain ratio which is a modification of Information gain that reduces its bias and is usually the best option. Gain ratio overcomes the problem with information gain by taking into account the number of branches that would result before making the split. It corrects information gain by taking the intrinsic information of a split into account.

Let us consider if we have a dataset that has users and their movie genre preferences based on variables like gender, group of age, rating, blah, blah. With the help of information

gain, you split at 'Gender' (assuming it has the highest information gain) and now the variables 'Group of Age' and 'Rating' could be equally important and with the help of gain ratio, it will penalize a variable with more distinct values which will help us decide the split at the next level.

$$Gain\ Ratio = \frac{Information\ Gain}{SplitInfo} = \frac{Entropy\ (before) - \sum_{j=1}^{K} Entropy(j, after)}{\sum_{j=1}^{K} w_j \log_2 w_j}$$

Where "before" is the dataset before the split, K is the number of subsets generated by the split, and (j, after) is subset j after the split.

2.12.2.5 Reduction in Variance

Reduction in variance is an algorithm used for continuous target variables (regression problems). This algorithm uses the standard formula of variance to choose the best split. The split with lower variance is selected as the criteria to split the population:

$$Variance = \frac{\Sigma(X - \overline{X})^2}{n}$$

Above X-bar is the mean of the values, X is actual and n is the number of values. Steps to calculate Variance:

1. Calculate variance for each node.
2. Calculate variance for each split as the weighted average of each node variance.

2.12.2.6 Chi-Square

The acronym CHAID stands for Chi-squared Automatic Interaction Detector. It is one of the oldest tree classification methods. It finds out the statistical significance between the differences between sub-nodes and parent node. We measure it by the sum of squares of standardized differences between observed and expected frequencies of the target variable. It works with the categorical target variable "Success" or "Failure". It can perform two or more splits. Higher the value of Chi-Square higher the statistical significance of differences between sub-node and Parent node.

It generates a tree called CHAID (Chi-square Automatic Interaction Detector). Mathematically, Chi-squared is represented as:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

Where:

χ^2 = Chi Square obtained
$\sum$ = the sum of
O = observed score
E = expected score

Steps to Calculate Chi-square for a split:

1. Calculate Chi-square for an individual node by calculating the deviation for Success and Failure both

2. Calculated Chi-square of Split using Sum of all Chi-square of success and Failure of each node of the split

2.13 How to avoid/counter Overfitting in Decision Trees?

The common problem with Decision trees, especially having a table full of columns, they fit a lot. Sometimes it looks like the tree memorized the training data set. If there is no limit set on a decision tree, it will give you 100% accuracy on the training data set because in the worst case it will end up making 1 leaf for each observation. Thus, this affects the accuracy when predicting samples that are not part of the training set. Here are two ways to remove overfitting:

1. Pruning Decision Trees.

2. Random Forest

2.13.1 Pruning Decision Trees

The splitting process results in fully grown trees until the stopping criteria are reached. But, the fully grown tree is likely to overfit the data, leading to poor accuracy on unseen data. In pruning, you trim off the branches of the tree, i.e., remove the decision nodes

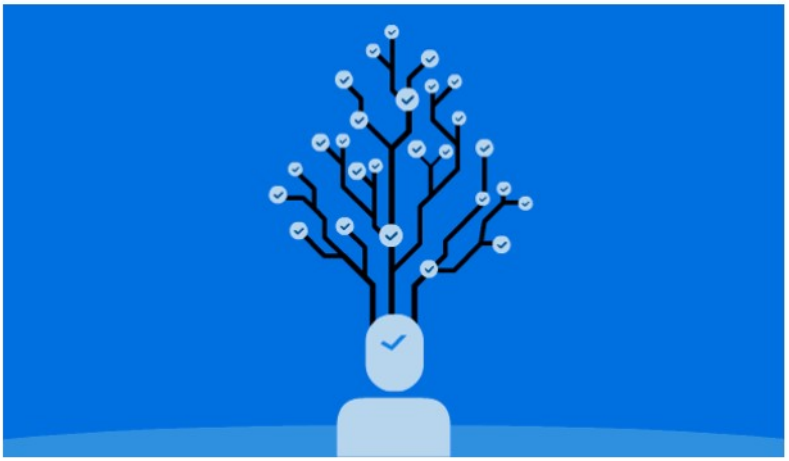


Figure 2.10: Pruning in action

starting from the leaf node such that the overall accuracy is not disturbed. This is done by segregating the actual training set into two sets: training data set, D and validation data set, V. Prepare the decision tree using the segregated training data set, D. Then continue trimming the tree accordingly to optimize the accuracy of the validation data set, V. In

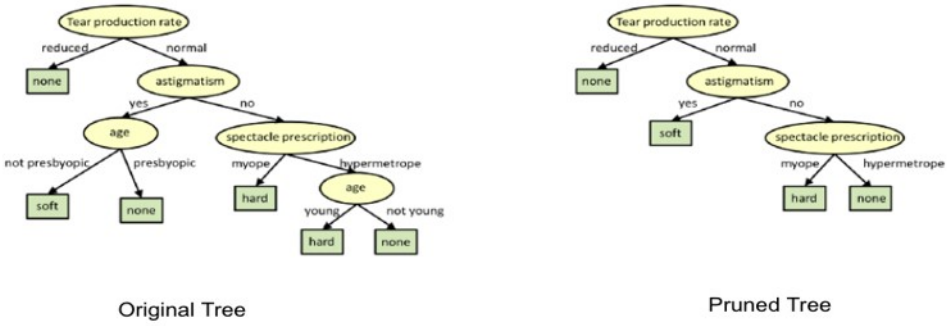


Figure 2.11: Pruning

the above diagram, the 'Age' attribute in the left-hand side of the tree has been pruned as it has more importance on the right-hand side of the tree, hence removing overfitting.

2.13.2 Random Forest

Random Forest is an example of ensemble learning, in which we combine multiple machine learning algorithms to obtain better predictive performance. Why the name "Random"? Two key concepts that give it the name random:

1. A random sampling of training data set when building trees.
2. Random subsets of features considered when splitting nodes.

A technique known as bagging is used to create an ensemble of trees where multiple training sets are generated with replacement.

In the bagging technique, a data set is divided into N samples using randomized sampling. Then, using a single learning algorithm a model is built on all samples. Later, the resultant predictions are combined using voting or averaging in parallel.

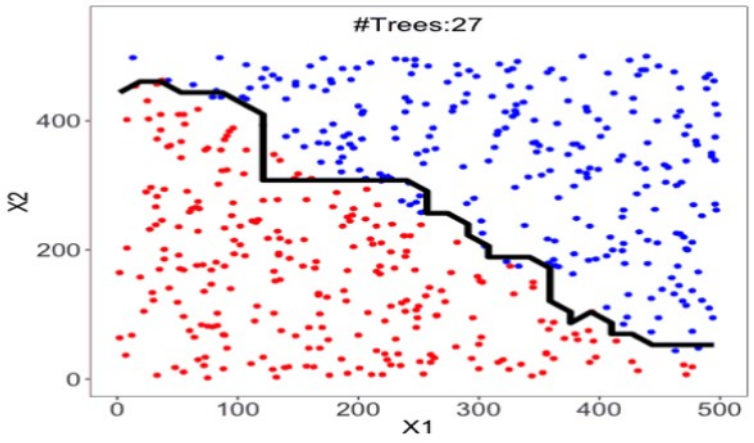


Figure 2.12: Random Forest in action

Chapter 3

BASIC MATHEMATICS AND PROBABILITY FOR DATA SCIENCE

Course Outcomes

After successful completion of this module, students should be able to:

CO 3	Solve mathematical problems using various arithmetic and more challenging forms of math.	Apply
CO 4	Apply probability theorems and approaches for calculating the number of outcomes of the events.	Apply

3.1 Basic Mathematics

It's time to start looking at some basic mathematic principles that are handy when dealing with data science. The word math tends to strike fear in the hearts of many, but I aim to make this as enjoyable as possible. In this chapter, we will go over the basics of the following topics:

- Basic symbols/terminology
- Logarithms/exponents
- The set theory
- Calculus

- Matrix (linear) algebra

We will also cover other fields of mathematics. Moreover, we will see how to apply each of these to various aspects of data science as well as other scientific endeavors.

3.2 Basic symbols and terminology

First, let's take a look at the most basic symbols that are used in the mathematical process as well as some more subtle notations used by data scientists.

3.2.1 Vectors and matrices

A vector is defined as an object with both magnitude and direction. This definition, however, is a bit complicated for our use. For our purpose, a vector is simply a 1-dimensional array representing a series of numbers. Put in another way, a vector is a list of numbers. It is generally represented using an arrow or bold font, as shown: Vectors are broken

$$\vec{x} \quad or \quad x$$

into components, which are individual members of the vector. We use index notations to denote the element that we are referring to, as illustrated:

$$\text{If } \vec{x} = \begin{pmatrix} 3 \\ 6 \\ 8 \end{pmatrix} \text{ then } x_1 = 3$$

In Python, we can represent arrays in many ways. We could simply use a Python list to represent the preceding array: x = [3, 6, 8] However, it is better to use the numpy array type to represent arrays, as shown, because it gives us much more utility when performing vector operations: import numpy as np x = np.array([3, 6, 8]) Regardless of the Python representation, vectors give us a simple way of storing multiple dimensions of a single data point/observation. Consider that we measure the average satisfaction rating (0-100) of employees for three departments of a company as being 57 for HR, 89 for engineering,

and 94 for management. We can represent this as a vector with the following formula: This vector holds three different bits of information about our data. This is the perfect

$$x=\begin{pmatrix} x1 \\ x2 \\ x3 \end{pmatrix} = \begin{pmatrix} 57 \\ 89 \\ 94 \end{pmatrix}$$

use of a vector in data science. You can also think of a vector as being the theoretical generalization of Panda's Series object. So, naturally, we need something to represent the Dataframe. We can extend our notion of an array to move beyond a single dimension and represent data in multiple dimensions. A matrix is a 2-dimensional representation of arrays of numbers. Matrices (plural) have two main characteristics that we need to be aware of. The dimension of the matrix, denoted as n x m (n by m), tells us that the matrix has n rows and m columns. Matrices are generally denoted using a capital, bold-faced letter, such as X.

Consider the following example: This is a 3 x 2 (3 by 2) matrix because it has three rows

$$\begin{pmatrix} 3 & 4 \\ 8 & 55 \\ 5 & 9 \end{pmatrix}$$

and two columns.

The matrix is our generalization of the Pandas Dataframe. It is arguably one of the most important mathematical objects in our toolkit. It is used to hold organized information, in our case, data. Revisiting our previous example, let's say we have three offices in different locations, each with the same three departments: HR, engineering, and management. We could make three different vectors, each holding a different office's satisfaction scores, as shown: However, this is not only cumbersome, but also unscalable. What if you have 100

$$x=\begin{pmatrix} 57 \\ 89 \\ 94 \end{pmatrix}, y=\begin{pmatrix} 67 \\ 87 \\ 84 \end{pmatrix}, z=\begin{pmatrix} 65 \\ 98 \\ 60 \end{pmatrix}$$

different offices? Then we would need to have 100 different 1-dimensional arrays to hold this information. This is where a matrix alleviates this problem. Let's make a matrix where each row represents a different department and each column represents a different office, as shown:

This is much more natural. Now, let's strip away the labels, and we are left with a matrix!

	Office 1	Office 2	Office 3
Hr	57	67	65
Engineering	89	87	98
Management	94	84	60

$$X=\begin{pmatrix}57 & 67 & 65\\ 89 & 87 & 98\\ 94 & 84 & 60\end{pmatrix}$$

3.3 Arithmetic symbols

In this section, we will go over some symbols associated with basic arithmetic. Summation The uppercase sigma symbol is a universal symbol for addition. Whatever is to the right of the sigma symbol is usually something iterable, meaning that we can go over it one by one (for example, a vector). For example, let's create the representation of a vector:
X = [1, 2, 3, 4, 5] To find the sum of the content, we can use the following formula:
xi = 15 In Python, we can use the following formula: For example, the formula for

$$\text{sum(x)} \# == 1$$

calculating the mean of a series of numbers is quite common. If we have a vector (x) of length n, the mean of the vector can be calculated as follows: This means that we will add

$$\text{Mean}= 1/n \sum x_i$$

up each element of x, denoted by, and then multiply the sum by 1/n, otherwise known as dividing by n, the length of the vector.

3.4 Proportional

The lowercase alpha symbol represents values that are proportional to each other. This means that as one value changes, so does the other. The direction in which the values move depends on how the values are proportional. Values can either vary directly or indirectly. If values vary directly, they both move in the same direction (as one goes up, so does the other). If they vary indirectly, they move in opposite directions (if one goes down, the other goes up). Consider the following examples:

- The sales of a company vary directly with the number of customers. This can be written

as

Sales αCustomers

- Gas prices vary (usually) indirectly with oil availability, meaning that as the availability of oil goes down (it's more scarce), gas prices will go up. This can be denoted as

Gas αOil Availability

Later on, we will see a very important formula called the Bayes formula, which includes a variation symbol.

3.5 Dot product

The dot product is an operator like addition and multiplication. It is used to combine two vectors, as shown:

So, what does this mean? Let's say we have a vector that represents a customer's senti-

$$\begin{bmatrix}3\\7\end{bmatrix}.\begin{bmatrix}9\\5\end{bmatrix}=3*9+7*5=62$$

ments towards three genres of movies—comedy, romantic, and action.

Consider that, on a scale of 1-5, a customer loves comedies, hates romantic movies, and is alright with action movies. We might represent this as follows: Here:

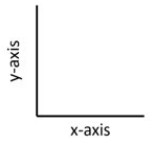
$$\begin{pmatrix}5\\1\\3\end{pmatrix}$$

- 5 denotes the love for comedies
- 1 is the hatred for romantic
- 3 is the indifference of action

3.6 Graphs

No doubt you have encountered dozens, if not hundreds, of graphs in your life so far. I'd like to mostly talk about conventions with regard to graphs and notations.

This is a basic Cartesian graph (x and y coordinate). The x and y notation are very



standard but sometimes do not entirely explain the big picture. We sometimes refer to the x variable as being the independent variable and the y as the dependent variable. This is because when we write functions, we tend to speak about them as being y is a function of x, meaning that the value of y is dependent on the value of x. This is what a graph is trying to show.

Suppose we have two points on a graph, as shown:

3.7 Linear algebra

Remember the movie recommendation engine we looked at earlier? What if we had 10,000 movies to recommend and we had to choose only 10 to give to the user? We'd have to take a dot product between the user profile and each of the 10,000 movies. Linear algebra provides the tools to make these calculations much more efficient.

It is an area of mathematics that deals with the math of matrices and vectors. It has the aim of breaking down these objects and reconstructing them in order to provide practical applications. Let's look at a few linear algebra rules before proceeding.

3.7.1 Matrix multiplication

Like numbers, we can multiple matrices together. Multiplying matrices is, in essence, a mass produced way of taking several dot products at once. Let's, for example, try to multiple the following matrices: A couple of things:

- Unlike numbers, multiplication is not commutative, meaning that the order in which you multiply matrices matters a great deal.

$$\begin{pmatrix} 1 & 5 \\ 5 & 8 \\ 7 & 8 \end{pmatrix} \cdot \begin{pmatrix} 3 & 4 \\ 2 & 5 \end{pmatrix}$$

- In order to multiply matrices, their dimensions...

A Gentle Introduction to Probability Over the next few chapters, we will explore both probability and statistics as methods of examining both data-driven situations and real-world scenarios. The rules of probability govern the basics of prediction. We use probability to define the chances of the occurrence of an event.

Probability will help us model real-life events that include a sense of randomness and chance. Over the next two chapters, we will look at the terminology behind probability theorems and how to apply them to model situations that can appear unexpectedly.

3.8 Probabbility

One of the most basic concepts of probability is the concept of a procedure. A procedure is an act that leads to a result. For example, throwing a dice or visiting a website.

An event is a collection of the outcomes of a procedure, such as getting a heads on a coin flip or leaving a website after only 4 seconds. A simple event is an outcome/event of a procedure that cannot be broken down further. For example, rolling two dice can be broken down into two simple events: rolling die 1 and rolling die 2.

The sample space of a procedure is the set of all possible simple events. For example, an experiment is performed, in which a coin is flipped three times in succession. What is the size of the sample space for this experiment?

The answer is eight, because the results could be any one of the possibilities in the following sample space—

HHH, HHT, HTT, HTH, TTT, TTH, THH, or THT. The probability of an event represents the frequency, or chance, that the event will happen.

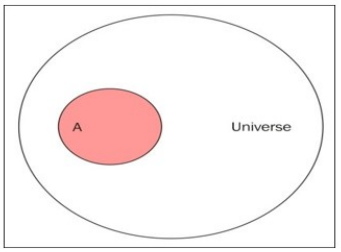
For notation, if A is an event, P(A) is the probability of the occurrence of the event.

We can define the actual probability of an event, A, as follows: Here, A is the event in question. Think of an entire universe of events where anything is possible, and let's

$$P(A) = \frac{\text{number of ways A occur}}{\text{size of sample space}}$$

represent it as a circle. We can think of a single event, A, as being a smaller circle within that larger universe, as shown in the following diagram:

Let's now pretend that our universe involves a research study on humans, and the A



event is people in that study who have cancer.

If our study has 100 people and A has 25 people, the probability of A or P(A) is 25/100. The maximum probability of any event is 1. This can be understood as the red circle grows so large that it is the size of the universe (the larger circle).

The most basic examples (I promise they will get more interesting) are coin flips. Let's say we have two coins and we want the probability that we will roll two heads. We can very easily count the number of ways two coins could end up being two heads.

There's only one! Both coins have to be heads. But how many options are there? It could either be two heads, two tails, or a heads/tails combination. First, let's define A. It is the event in which two heads occur. The number of ways that A can occur is 1.

The sample space of the experiment is HH, HT, TH, TT, where each two letter word indicates the outcome of the first and second coin simultaneously. The size of the sample space is four. So, P(getting two heads) = 1/4. Let's refer to a quick visual table to prove it. The following table denotes the options for coin 1 as the columns and the options for coin 2 as the rows. In each cell, there is either a True or a False. A True value indicates that it satisfies the condition (both heads) and False indicates otherwise. Coin 1 is Heads Coin 1 is Tails Coin 2 is Heads True False Coin 2 is Tails False False So, we have one out of a total of four possible outcomes.

3.9 Bayesian versus Frequentist

The preceding example was almost too easy. In practice, we can hardly ever truly count the number of ways something can happen. For example, let's say that we want to know the probability of a random person smoking cigarettes at least once a day. If we wanted to approach this problem using the classical way (the previous formula), we would need to figure out how many different ways a person is a smoker—someone who smokes at least once a day—which is not possible!

When faced with such a problem, two main schools of thought are considered when it comes to calculating probabilities in practice: the Frequentist approach and the Bayesian approach. This chapter will focus heavily on the Frequentist approach while the subsequent chapter will dive into the Bayesian analysis.

3.9.1 Frequentist approach

In a Frequentist approach, the probability of an event is calculated through experimentation. It uses the past in order to predict the future chance of an event. The basic formula is as follows:

Basically, we observe several instances of the event and count the number of times A was

$$P(A) = \frac{\text{number of times A occured}}{number\ times\ the\ procedure\ was\ repeated}$$

satisfied. The division of these numbers is an approximation of the probability.

The Bayesian approach differs by dictating that probabilities must be discerned using theoretical means. Using the Bayes approach, we would have to think a bit more critically about events and why they occur. Neither methodology is 100% the correct answer all the time. Usually, it comes down to the problem and the difficulty of using either approach.

The crux of the Frequentist approach is the relative frequency.

The relative frequency of an event is how often an event occurs divided by the total number of observations.

Example – marketing stats

Let's say that you are interested in ascertaining how often a person who visits your website is likely to return on a later date. This is sometimes called the rate of repeat visitors. In the previous definition, we would define our A event as being a visitor coming back to the site. We would then have to calculate the number of ways a person can come back, which doesn't really make sense at all! In this case, many people would turn to a Bayesian approach; however, we can calculate what is known as relative frequency.

So, in this case, we can take the visitor logs and calculate the relative frequency of event A (repeat visitors). Let's say, of the 1,458 unique visitors in the past week, 452 were repeat visitors. We can calculate this as follows:

So, about 31% of your visitors are repeat visitors.

$$P(A)RF(A) = \frac{452}{1458}$$

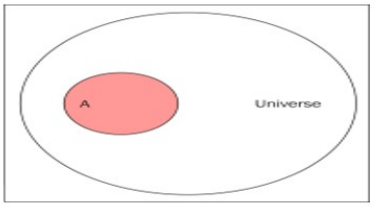
3.9.2 Compound events

Sometimes, we need to deal with two or more events. These are called compound events. A compound event is any event that combines two or more simple events. When this happens, we need some special notation.

Given events A and B:

- The probability that A and B occur is P(A ∩B) = P(A and B)
- The probability that either A or B occurs is P(A ∪B) = P(A or B)

Understanding why we use set notation for these compound events is very important. Remember how we represented events in a universe using circles earlier? Let's say that our Universe is 100 people who showed up for an experiment, in which a new test for cancer is being developed: In the preceding diagram, the red circle, A, represents 25 people who

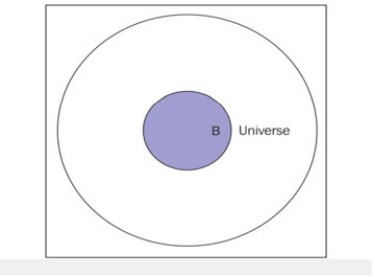


actually have cancer. Using the relative frequency approach, we can say that P(A) =

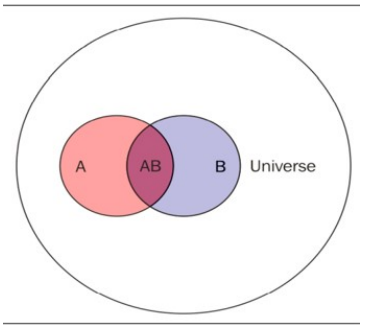
number of people with cancer/number of people in study, that is, 25/100 = ¼ = .25. This means that there is a 25% chance that someone has cancer.

Let's introduce a second event, called B, as shown, which contains people for whom the test was positive (it claimed that they had cancer). Let's say that this is for 30 people. So, P(B) = 30/100 = 3/10 = .3. This means that there is a 30% chance that the test said positive for any given person.

These are two separate events, but they interact with each other. Namely, they might



intersect or have people in common, as shown here: Anyone in the space that both A and

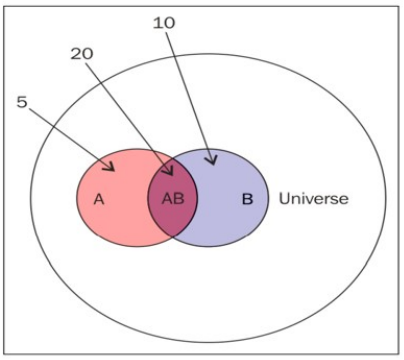


B occupy, otherwise known as A intersect B or A B, are people for whom the test claimed they were positive for cancer (A) and they actually do have cancer. Let's say that's 20 people. The test said positive for 20 people, that is, they have cancer, as shown here:

This means that P(A and B) = 20/100 = 1/5 = .2 = 20%.

If we want to say that someone has cancer or the test came back positive. This would be the total sum (or union) of the two events, namely, the sum of 5, 20, and 10, which is 35. So, 35/100 people either have cancer or had a positive test outcome. That means, P(A or B) = 35/100 = .35 = 35%. All in all, we have people in the following four different classes:

- Pink: This refers to the people who have cancer and had a negative test outcome
- Purple (A intersect B): These people have cancer and had a positive test outcome
- Blue: This refers to the people with no cancer and a positive test outcome



- White: This refers to the people with no cancer and a negative test outcome.

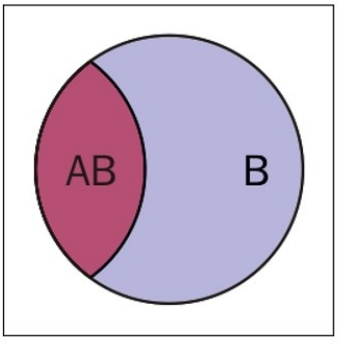
So, effectively, the only times the test was accurate was in the white and purple regions. In the blue and pink regions, the test was incorrect.

3.9.3 Conditional probability

Let's pick an arbitrary person from this study of 100 people. Let's also assume that you are told that their test result was positive. What is the probability of them actually having cancer? So, we are told that event B has already taken place, and that their test came back positive. The question now is: what is the probability that they have cancer, that is P(A)? This is called a conditional probability of A given B or P(A|B). Effectively, it is asking you to calculate the probability of an event given that another event has already happened.

You can think of conditional probability as changing the relevant universe. P(A|B) (called the probability of A given B) is a way of saying, given that my entire universe is now B, what is the probability of A? This is also known as transforming the sample space.

Zooming in on our previous diagram, our universe is now B, and we are concerned with



AB (A and B) inside of B

The formula can be given as follows:

P(A|B) = P(A and B) / P(B) = (20/100) / (30/100) = 20/30 = .66 = 66%

There is a 66% chance that if a test result came back positive, that person had cancer. In reality, this is the main probability that the experimenters want. They want to know how good the test is at predicting cancer.

3.9.4 The rules of probability

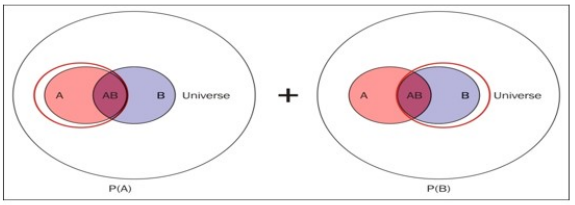
In probability, we have some rules that become very useful when visualization gets too cumbersome. These rules help us calculate compound probabilities with ease.

3.9.4.1 The addition rule

The addition rule is used to calculate the probability of either or events. To calculate P(A ∪B) = P(A or B), we use the following formula:

P(A ∪B) = P(A) + P(B) − P(A ∩B)

The first part of the formula (P(A) + P(B)) makes complete sense. To get the union of the two events, we have to add together the area of the circles in the universe. But why the subtraction of P(A and B)? This is because when we add the two circles, we are adding the area of intersection twice, as shown in the following diagram: See how both the red



circles include the intersection of A and B? So, when we add them, we need to subtract just one of them to account for this, leaving us with our formula.

Recall that we wanted the number of people who either had cancer or had a positive test result? If A is the event that someone has cancer, and B is that the test result was positive, we have: P(A or B) = P(A) + P(B) – P(A and B) = .25 + .30 - .2 = .35 This was calculated before visually in the diagram A.

3.9.4.2 Mutual exclusivity

We say that two events are mutually exclusive if they cannot occur at the same time. This means that A B= or just that the intersection of the events is the empty set. When this happens, P(A ∩B) = P(A and B) = 0.

If two events are mutually exclusive, then:

P(A ∪B) = P(A or B) = P(A) + P(B) − P(A ∩B) = P(A) + P(B)

This makes the addition rule much easier. Some examples of mutually exclusive events include the following:

- A customer seeing your site for the first time on both Twitter and Facebook.
- Today is Saturday and today is Wednesday.
- I failed Econ 101 and I passed Econ 101 None of these events can occur simultaneously.

3.9.4.3 The multiplication rule

The multiplication rule is used to calculate the probability of and events.

To calculate P(A B) = P(A and B), we use the following formula:

P(A ∩B) = P(A and B) = P(A) · P(B|A) Why do we use B|A instead of B? This is because it is possible that B depends on A. If this is the case, then just multiplying P(A) and P(B) does not give us the whole picture. In our cancer trial example, let's find P(A and B). To do this, let's redefine A to be the event that the trial is positive and B to be the person having cancer (because it doesn't matter what we call the events). The equation will be as follows: P(A ∩B) = P(A and B) = P(A) · P(B|A) = .3 * .6666 = .2 = 20% This was calculated before visually. It's difficult to see the true necessity of using the conditional probability, so, let's try another, more difficult problem. For example, of a randomly selected set of 10 people, 6 have iPhones and 4 have Androids. What is the probability that if I randomly select two people, they both will have iPhones?

This example can be retold using event spaces, as follows: I have the following two events:

- A: This event shows the probability that I choose a person with an iPhone first.
- B: This event shows the probability that I choose a person with an iPhone second.

So, basically, I want the following:

- P(A and B): P(I choose a person with an iPhone and a person with an iPhone) So, I

can use my P(A and B) = P(A) · P(B|A) formula. P(A) is simple, right?
People with iPhones are 6 out of 10, so, I have a 6/10 = 3/5 = 0.6 chance of A.
This means P(A) = 0.6. So, if I have a 0.6 chance of choosing someone with an iPhone, the probability of choosing two should just be 0.6 * 0.6, right? But wait! We only have 9 people left to choose our second person from, because one was taken away. So in our new transformed sample space, we have 9 people in total, 5 with iPhones and 4 with droids, making P(B) = 5/9 = .555. So, the probability of choosing two people with iPhones is 0.6 * 0.555 = 0.333 = 33%. I have a 1/3 chance of choosing two people with iPhones out of 10. The conditional probability is very important in the multiplication rule as it can drastically alter your answer.

3.9.4.4 Independence

Two events are independent if one event does not affect the outcome of the other, that is P(B|A) = P(B) and P(A|B) = P(A).
If two events are independent, then: P(A ∩B) = P(A) · P(B|A) = P(A) · P(B)
Some examples of independent events are as follows:

- It was raining in San Francisco, and a puppy was born in India.
- Flip a coin and get heads and flip another coin and get tails None of these pairs of events affect each other.

3.9.4.5 Bit deeper

Without getting too deep into the machine learning terminology, this test is what is known as a binary classifier, which means that it is trying to predict from only two options: have cancer or no cancer. When we are dealing with binary classifiers, we can draw what are called confusion matrices, which are 2 x 2 matrices that house all the four possible outcomes of our experiment.
Let's try some different numbers. Let's say 165 people walked in for the study. So, our n (sample size) is 165 people. All 165 people are given the test and asked if they have cancer (provided through various other means). The following confusion matrix shows us

the results of this experiment:

The matrix shows that 50 people were predicted to have no cancer and did not have it,

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

100 people were predicted to have cancer and actually did have it, and so on. We have the following four classes, again, all with different names:

The matrix shows that 50 people were predicted to have no cancer and did not have it, 100 people were predicted to have cancer and actually did have it, and so on. We have the following four classes, again, all with different names:

- The true positives are the tests correctly predicting positive (cancer) == 100
- The true negatives are the tests correctly predicting negative (no cancer) == 50
- The false positives are the tests incorrectly predicting positive (cancer) == 10
- The false negatives are the tests incorrectly predicting negative (no cancer) ==5

3.9.5 Collectively exhaustive events

When given a set of two or more events, if at least one of the events must occur, then such a set of events is said to be collectively exhaustive.

Consider the following examples:

- Given a set of events temperature < 60, temperature > 90, these events are not collectively exhaustive because there is a third option that is not given in this set of events: The temperature could be between 60 and 90. However, they are mutually exhaustive because both cannot happen at the same time.
- In a dice roll, the set of events of rolling a 1, 2, 3, 4, 5, or 6 are collectively exhaustive because these are the only possible events, and at least one of them must happen.

3.10 Bayes Theorem

The Bayes Theorem or often called Bayes Law is arguably the most powerful rule of probability and statistics, named after famous English statistician and philosopher, Thomas Bayes.

Bayes theorem is a powerful probability law that brings the concept of subjectivity into the world of Statistics and Mathematics where everything is about facts. It describes the probability of an event, based on the prior information of conditions that might be related to that event. For instance, if the risk of getting Coronavirus or Covid-19 is known to increase with age, then Bayes Theorem allows the risk to an individual of a known age to be determined more accurately by conditioning it on the age than simply assuming that this individual is common to the population as a whole.

The concept of conditional probability, which plays a central role in Bayes theory, is a measure of the probability of an event happening, given that another event has already occurred. Bayes theorem can be described by the following expression where the X and Y stand for events X and Y, respectively:

- Pr (X|Y): the probability of event X occurring given that event or condition Y has

$$Pr\,(\mathrm{X} \mid \mathrm{Y}) = \frac{Pr\,(\mathrm{Y} \mid \mathrm{X})\,Pr\,(\mathrm{X})}{Pr\,(\mathrm{Y})}$$

occurred or is true.

- Pr (Y|X): the probability of event Y occurring given that event or condition X has occurred or is true.
- Pr (X) & Pr (Y): the probabilities of observing events X and Y, respectively.

In the case of the earlier example, the probability of getting Coronavirus (event X) conditional on being at a certain age is Pr (X|Y), which is equal to the probability of being at a certain age given one got a Coronavirus, Pr (Y|X), multiplied with the probability of getting a Coronavirus, Pr (X), divided to the probability of being at a certain age., Pr (Y).

3.10.1 Random variables

A random variable uses real numerical values to describe a probabilistic event. In our previous work with variables (both in math and programming), we were used to the fact that a variable takes on a certain value. For example, we might have a triangle in which we are given a variable h for the hypotenuse, and we must figure out the length of the hypotenuse. We also might have, in Python:

x = 5

Both of these variables are equal to one value at a time. In a random variable, we are subject to randomness, which means that our variables' values are, well just that, variable! They might take on multiple values depending on the environment.

A random variable still, as shown previously, holds a value. The main distinction between variables as we have seen them and a random variable is the fact that a random variable's value may change depending on the situation.

However, if a random variable can have many values, how do we keep track of them all? Each value that a random variable might take on is associated with a percentage. For every value that a random variable might take on, there is a single probability that the variable will be this value.

With a random variable, we can also obtain our probability distribution of a random variable, which gives the variable's possible values and their probabilities.

Written out, we generally use single capital letters (mostly the specific letter X) to denote random variables.

For example, we might have:

- X = the outcome of a dice roll
- Y = the revenue earned by a company this year
- Z = the score of an applicant on an interview coding quiz (0-100%).

Effectively, a random variable is a function that maps values from the sample space of an event (the set of all possible outcomes) to a probability value (between 0 and 1).

Chapter 4

STATISTICS FOR DATA SCIENCE

Course Outcomes

After successful completion of this module, students should be able to:

CO 5	Illustrate the obtaining and sampling data in statistics to quantify and visualize our data.	Understand

This chapter will focus on the statistics required by any aspiring data scientist.
We will explore ways of sampling and obtaining data without being affected by bias and then use measures of statistics to quantify and visualize our data. Using the z-score and the Empirical rule, we will see how we can standardize data for the purpose of both graphing and interpretability.

- How to obtain and sample data
- The measures of center, variance, and relative standing
- Normalization of data using the z-score
- The Empirical rule

4.1 What are statistics?

This might seem like an odd question to ask, but I am frequently surprised by the number of people who cannot answer this simple and yet powerful question: what are statistics? Statistics are the numbers you always see on the news and in the paper. Statistics are useful when trying to prove a point or trying to scare you, but what are they?

To answer this question, we need to back up for a minute and talk about why we even measure them in the first place. The goal of this field is to try to explain and model the world around us. To do that, we have to take a look at the population.

We can define a population as the entire pool of subjects of an experiment or a model. Essentially, your population is who you care about. Who are you trying to talk about? If you are trying to test if smoking leads to heart disease, your population would be the smokers of the world. If you are trying to study teenage drinking problems, your population would be all teenagers.

Now, consider that you want to ask a question about your population, for example, if your population is all of your employees (assume that you have over 1,000 employees), perhaps you want to know what percentage of them use illicit drugs. The question is called a parameter.

We can define a parameter as a numerical measurement describing a characteristic of a population.

For example, if you ask all 1,000 employees and 100 of them are using drugs, the rate of drug use is 10%. The parameter here is 10%.

However, let's get real, you probably can't ask every single employee whether they are using drugs. What if you have over 10,000 employees? It would be very difficult to track everyone down in order to get your answer. When this happens, it's impossible to figure out this parameter. In this case, we can estimate the parameter. First, we will take a sample of the population.

We can define a sample of a population as a subset (random not required) of the population.

So, we perhaps ask 200 of the 1,000 employees you have. Of these 200, suppose 26 use

drugs, making the drug use rate 13%. Here, 13% is not a parameter because we didn't get a chance to ask everyone. This 13% is an estimate of a parameter. Do you know what that's called?
That's right, a statistic!
We can define a statistic as a numerical measurement describing a characteristic of a sample of a population. A statistic is just an estimation of a parameter. It is a number that attempts to describe an entire population by describing a subset of that population. This is necessary because you can never hope to give a survey to every single teenager or to every single smoker in the world. That's what the field of statistics is all about— taking samples of populations and running tests on these samples.
So, the next time you are given a statistic, just remember, that number only represents a sample of that population, not the entire pool of subjects.

4.1.1 How do we obtain and sample data?

If statistics is about taking samples of populations, it must be very important to know how we obtain these samples, and you'd be correct. Let's focus on just a few of the many ways of obtaining and sampling data.

4.2 Obtaining data

There are two main ways of collecting data for our analysis: observational and experimentation. Both these ways have their pros and cons, of course. They each produce different types of behavior and, therefore, warrant different types of analysis.

4.2.1 Observational

We might obtain data through observational means, which consists of measuring specific characteristics but not attempting to modify the subjects being studied. For example,

you have a tracking software on your website that observes users' behavior on the website, such as length of time spent on certain pages and the rate of clicking on ads, all the while not affecting the user's experience, then that would be an observational study.
This is one of the most common ways to get data because it's just plain easy. All you have to do is observe and collect data. Observational studies are also limited in the types of data you may collect. This is because the observer (you) is not in control of the environment. You may only watch and collect natural behavior. If you are looking to induce a certain type of behavior, an observational study would not be useful.

4.2.2 Experimental

An experiment consists of a treatment and the observation of its effect on the subjects. Subjects in an experiment are called experimental units. This is usually how most scientific labs collect data. They will put people into two or more groups (usually just two) and call them the control and the experimental group.
The control group is exposed to a certain environment and then observed. The experimental group is then exposed to a different environment and then observed. The experimenter then aggregates data from both the groups and makes a decision about which environment was more favorable (favorable is a quality that the experimenter gets to decide).
In a marketing example, consider that we expose half of our users to a certain landing page with certain images and a certain style (website A), and we measure whether or not they sign up for the service. Then, we expose the other half to a different landing page, different images, and different styles (website B) and again measure whether or not they sign up. We can then decide which of the two sites performed better and should be used going further. This, specifically, is called an A/B test. Let's see an example in Python!
Let's suppose we run the preceding test and obtain the following results as a list of lists:
results = [['A', 1], ['B', 1], ['A', 0], ['A', 0] …]
Here, each object in the list result represents a subject (person). Each person then has the following two attributes:

- Which website they were exposed to, represented by a single character
- Whether or not they converted (0 for no and 1 for yes)

We can then aggregate and come up with the following results table:

users_exposed_to_A = []

users_exposed_to_B = []

create two lists to hold the results of each individual website

Once we create these two lists that will eventually hold each individual conversion Boolean (0 or 1), we will iterate all of our results of the test and add them to the appropriate list, as shown:

for website, converted in results: # iterate through the results

will look something like website == 'A' and converted == 0

if website == 'A':

users_exposed_to_A.append(converted)

elif website == 'B':

users_exposed_to_B.append(converted)

Now, each list contains a series of 1s and 0s.

To get the total number of people exposed to website A, we can use the len() feature in Python, as illustrated:

len(users_exposed_to_A) == 188 #number of people exposed to website A

len(users_exposed_to_B) == 158 #number of people exposed to website B

To count the number of people who converted, we can use the sum() of the list, as shown:

sum(users_exposed_to_A) == 54 # people converted from website A

sum(users_exposed_to_B) == 48 # people converted from website B

If we subtract the length of the lists and the sum of the list, we are left with the number of people who did not convert for each site, as illustrated: len(users_exposed_to_A) - sum(users_exposed_to_A) == 134 # did not convert from website A

len(users_exposed_to_B) - sum(users_exposed_to_B) == 110 # did not convert from website B

We can aggregate and summarize our results in the following table that represents our experiment of website conversion testing:

	Did not sign up	Signed up
Website A	134	54
Website B	110	48

We can quickly drum up some descriptive statistics. We can say that the website conversion rates for the two websites are as follows:

Not much difference, but different nonetheless. Even though B has the higher conversion

- Conversion for website A: $\frac{54}{134+54} = .288$
- Conversion for website B: $\frac{48}{110+48} = .3$

rate, can we really say that the version B significantly converts better? Not yet. To test the statistical significance of such a result, a hypothesis test should be used. These tests will be covered in depth in the next chapter, where we will revisit this exact same example and finish it using the proper statistical test.

4.3 Sampling Data

Remember how statistics are the result of measuring a sample of a population. Well, we should talk about two very common ways to decide who gets the honor of being in the sample that we measure. We will discuss the main type of sampling, called random sampling, which is the most common way to decide our sample sizes and our sample members.

4.3.1 Probability sampling

Probability sampling is a way of sampling from a population, in which every person has a known probability of being chosen but that number might be a different probability than another user. The simplest (and probably the most common) probability sampling method is random sampling.

4.3.2 Random sampling

Suppose that we are running an A/B test and we need to figure out who will be in group A and who will be in group B. There are the following three suggestions from your data

team:

- Separate users based on location: Users on the west coast are placed in group A, while users on the east coast are placed in group B.
- Separate users based on the time of day they visit the site: Users who visit between 7 p.m. and 4 a.m. get site A, while the rest are placed in group B.
- Make it completely random: Every new user has a 50/50 chance of being placed in either group.

The first two are valid options for choosing samples and are fairly simple to implement, but they both have one fundamental flaw: they are both at risk of introducing a sampling bias.

A sampling bias occurs when the way the sample is obtained systemically favors some outcome over the target outcome.

It is not difficult to see why choosing option 1 or option 2 might introduce bias. If we chose our groups based on where they live or what time they log in, we are priming our experiment incorrectly and, now, we have much less control over the results. Specifically, we are at risk of introducing a confounding factor into our analysis, which is bad news.

A confounding factor is a variable that we are not directly measuring but connects the variables that are being measured.

Basically, a confounding factor is like the missing element in our analysis that is invisible but affects our results.

In this case, option 1 is not taking into account the potential confounding factor of geographical taste. For example, if website A is unappealing, in general, to the west coast users, it will affect your results drastically.

Similarly, option 2 might introduce a temporal (time-based) confounding factor. What if website B is better viewed in a nighttime environment (which was reserved for A), and users are turned off to the style purely because of what time it is. These are both factors that we want to avoid, so, we should go with option 3, which is a random sample.

A random sample is chosen such that every single member of a population has an equal chance of being chosen as any other member.

This is probably one of the easiest and most convenient ways to decide who will be a part of your sample. Everyone has the exact same chance of being in any particular group. Random sampling is an effective way of reducing the impact of confounding factors.

4.3.3 Unequal probability sampling

Recall that I previously said that a probability sampling might have different probabilities for different potential sample members. But what if this actually introduced problems? Suppose we are interested in measuring the happiness level of our employees. We already know that we can't ask every single person on the staff because that would be silly and exhausting. So, we need to take a sample. Our data team suggests random sampling and at first everyone high fives because they feel very smart and statistical. But then someone asks a seemingly harmless question— does anyone know the percentage of men/women who work here?

The high fives stop and the room goes silent.

This question is extremely important because sex is likely to be a confounding factor. The team looks into it and discovers a split of 75% men and 25% women in the company.

This means that if we introduce a random sample, our sample will likely have a similar split and, thus, favor the results for men and not women. To combat this, we can favor including more women than men in our survey in order to make the split of our sample less favored for men.

At first glance, introducing a favoring system in our random sampling seems like a bad idea, however, alleviating unequal sampling and, therefore, working to remove systematic bias among gender, race, disability, and so on is much more pertinent. A simple random sample, where everyone has the same chance as everyone else, is very likely to drown out the voices and opinions of minority population members. Therefore, it can be okay to introduce such a favoring system in your sampling techniques.

4.3.3.1 How do we measure statistics?

Once we have our sample, it's time to quantify our results. Suppose we wish to generalize the happiness of our employees or we want to figure out whether salaries in the company are very different from person to person.

These are some common ways of measuring our results.

4.3.3.2 Measures of center

Measures of center are how we define the middle, or center, of a dataset. We do this because sometimes we wish to make generalizations about data values. For example, perhaps we're curious about what the average rainfall in Seattle is or what the median height for European males is. It's a way to generalize a large set of data so that it's easier to convey to someone.

A measure of center is a value in the "middle" of a dataset.

However, this can mean different things to different people. Who's to say where the middle of a dataset is? There are so many different ways of defining the center of data. Let's take a look at a few.

The arithmetic mean of a dataset is found by adding up all of the values and then dividing it by the number of data values.

This is likely the most common way to define the center of data, but can be flawed! Suppose we wish to find the mean of the following numbers:

```
import numpy as np
np.mean([11, 15, 17, 14]) == 14.25
```

Simple enough, our average is 14.25 and all of our values are fairly close to it. But what if we introduce a new value: 31?

```
np.mean([11, 15, 17, 14, 31]) == 17.6
```

This greatly affects the mean because the arithmetic mean is sensitive to outliers. The new value, 31, is almost twice as large as the rest of the numbers and, therefore, skews the mean. Another, and sometimes better, measure of center is the median.

The median is the number found in the middle of the dataset when it is sorted in order, as shown:

```
np.median([11, 15, 17, 14]) == 14.5
np.median([11, 15, 17, 14, 31]) == 15
```

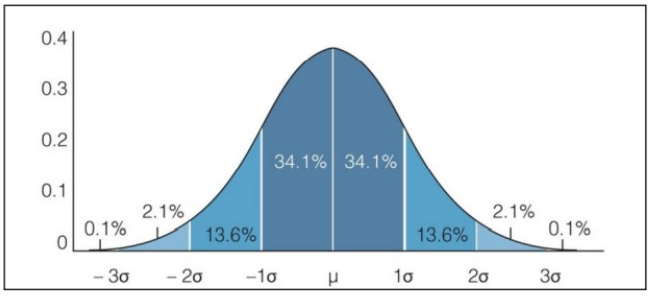
Note how the introduction of 31 using the median did not affect the median of the dataset greatly. This is because the median is less sensitive to outliers.

When working with datasets with many outliers, it is sometimes more useful to use the median of the dataset, while if your data does not have many outliers and the data points are mostly close to one another, then the mean is likely a better option.

But how can we tell if the data is spread out? Well, we will have to introduce a new type of statistic.

4.4 The Empirical rule

Recall that a normal distribution is defined as having a specific probability distribution that resembles a bell curve. In statistics, we love it when our data behaves normally. For example, if we have data that resembles a normal distribution, like so: The Empirical



rule states that we can expect a certain amount of data to live between sets of standard deviations. Specifically, the Empirical rule states for data that is distributed normally:

- about 68% of the data fall within 1 standard deviation
- about 95% of the data fall within 2 standard deviations
- about 99.7% of the data fall within 3 standard deviations

For example, let's see if our Facebook friends' data holds up to this. Let's use our Dataframe to find the percentage of people that fall within 1, 2, and 3 standard deviations of the mean, as shown:

# finding the percentage of people within one standard deviation of the mean
within_1_std = df_scaled[(df_scaled['friends_scaled'] <= 1) & (df_ scaled['friends_scaled'] >= -1)].shape[0]
within_1_std / float(df_scaled.shape[0]) # 0.75 # finding the percentage of people within two standard deviations of the mean within_2_std = df_scaled[(df_scaled['friends_scaled'] <= 2) & (df_ scaled['friends_scaled'] >= -2)].shape[0]
within_2_std / float(df_scaled.shape[0])
0.916

```
# finding the percentage of people within three standard deviations of the mean
within_3_std = df_scaled[(df_scaled['friends_scaled'] <= 3) & (df_scaled['friends_scaled']
>= -3)].shape[0]
within_3_std / float(df_scaled.shape[0])
# 1.0
```

We can see that our data does seem to follow the Empirical rule. About 75% of the people are within a single standard deviation of the mean. About 92% of the people are within two standard deviations, and all of them are within three standard deviations.

4.5 Point Estimates

Recall that, in the previous chapter, we mentioned how difficult it was to obtain a population parameter; so, we had to use sample data to calculate a statistic that was an estimate of a parameter. When we make these estimates, we call them point estimates.

A point estimate is an estimate of a population parameter based on sample data.

We use point estimates to estimate population means, variances, and other statistics. To obtain these estimates, we simply apply the function that we wish to measure for our population to a sample of the data. For example, suppose there is a company of 9,000 employees and we are interested in ascertaining the average length of breaks taken by employees in a single day. As we probably cannot ask every single person, we will take a sample of the 9,000 people and take a mean of the sample. This sample mean will be our point estimate.

The following code is broken into three parts:

- We will use the probability distribution, known as the Poisson distribution, to randomly generate 9,000 answers to the question: for how many minutes in a day do you usually take breaks? This will represent our ”population”. Remember, from Chapter 6, Advanced Probability, that the Poisson random variable is used when we know the average value of an event and wish to model a distribution around it.
- We will take a sample of 100 employees (using the Python random sample method) and find a point estimate of a mean (called a sample mean).
- Compare our sample mean (the mean of the sample of 100 employees) to our population mean.

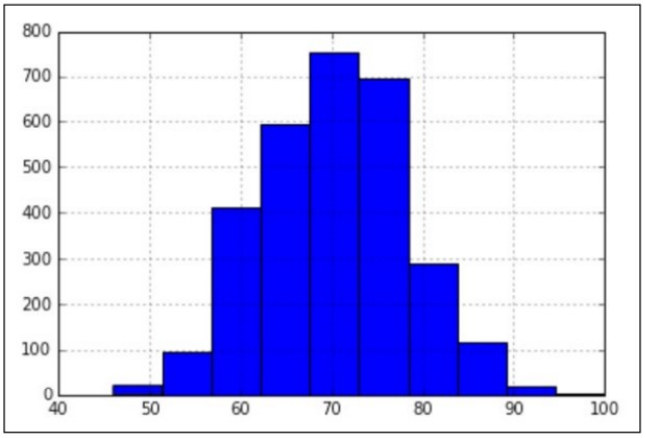
Let's take a look at the following code: np.random.seed(1234)

long_breaks = stats.poisson.rvs(loc=10, mu=60, size=3000)

represents 3000 people who take about a 60 minute break

The long_breaks variable represents 3000 answers to the question: how many minutes on an average do you take breaks for?, and these answers will be on the longer side. Let's see a visualization of this distribution, shown as follows: pd.Series(long_breaks).hist()

We see that our average of 60 minutes is to the left of the distribution. Also, because



we only sampled 3000 people, our bins are at their highest around 700-800 people. Now, let's model 6000 people who take, on an average, about 15 minutes' worth of breaks. Let's again use the Poisson distribution to simulate 6000 people, as shown:

short_breaks = stats.poisson.rvs(loc=10, mu=15, size=6000)

represents 6000 people who take about a 15 minute break

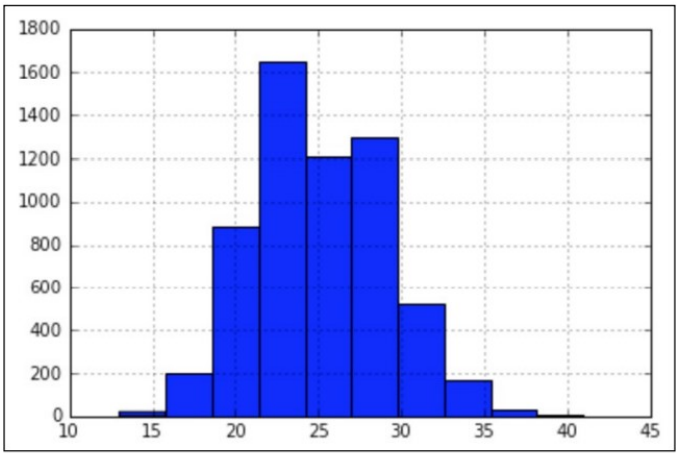
pd.Series(short_breaks).hist()

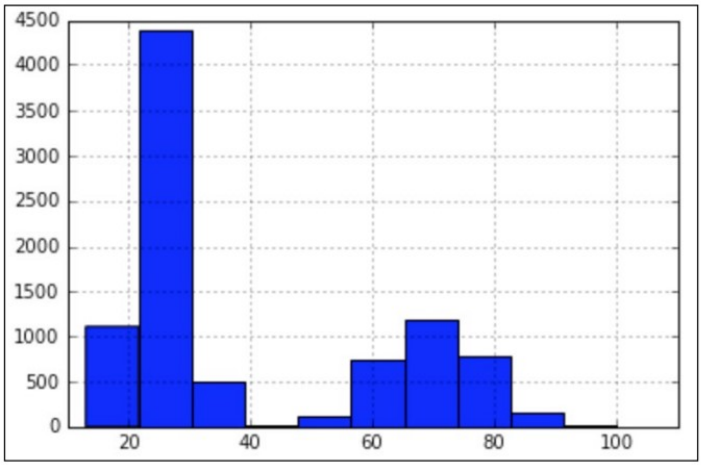
Okay, so, we have a distribution for the people who take longer breaks and a distribution for the people who take shorter breaks. Again, note how our average break length of 15 minutes falls to the left-hand side of the distribution, and note that the tallest bar is about 1600 people.

breaks = np.concatenate((long_breaks, short_breaks))

put the two arrays together to get our "population" of 9000 people The breaks variable is the amalgamation of all the 9000 employees, both long and short break takers. Let's see the entire distribution of people in a single visualization: pd.Series(breaks).hist()

We see how we have two humps. On the left, we have our larger hump of people who take about a 15 minute break, and on the right, we have a smaller hump of people who





take longer breaks. Later on, we will investigate this graph further. We can find the total average break length by running the following code: breaks.mean()

39.99 minutes is our parameter. Our average company break length is about 40 minutes. Remember that our population is the entire company's employee size of 9,000 people, and our parameter is 40 minutes. In the real world, our goal would be to estimate the population parameter because we would not have the resources to ask every single employee in a survey their average break length for many reasons. Instead, we will use a point estimate. So, to make our point, we want to simulate a world where we ask 100 random people about the length of their breaks. To do this, let's take a random sample of 100 employees out of the 9,000 employees we simulated, as shown:

sample_breaks = np.random.choice(a = breaks, size=100) # taking a sample of 100 employees Now, let's take the mean of the sample and subtract it from the population mean and see how far off we were: breaks.mean() - sample_breaks.mean() # difference between

means is 4.09 minutes, not bad! This is extremely interesting, because with only about 1% of our population (100 out of 9,000), we were able to get within 4 minutes of our population parameter and get a very accurate estimate of our population mean. Not bad! Here, we calculated a point estimate for the mean, but we can also do this for proportion parameters. By proportion, I am referring to a ratio of two quantitative values. Let's suppose that in a company of 10,000 people, our employees are 20% white, 10% black, 10% Hispanic, 30% Asian, and 30% identify as other. We will take a sample of 1,000 employees and see if their race proportions are similar.

```
employee_races = (["white"]*2000) + (["black"]*1000) +
(["hispanic"]*1000) + (["asian"]*3000) +
(["other"]*3000)
```

employee_races represents our employee population. For example, in our company of 10,000 people, 2,000 people are white (20%) and 3,000 people are Asian (30%).

Let's take a random sample of 1,000 people, as shown:

```
demo_sample = random.sample(employee_races, 1000) # Sample 1000
values
for race in set(demo_sample):
print( race + " proportion estimate:" )
print( demo_sample.count(race)/1000. )
```

The output obtained would be as follows:

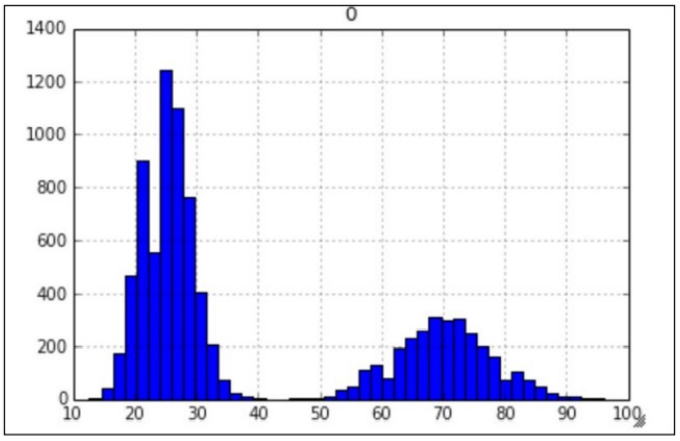
```
hispanic proportion estimate:
0.103
white proportion estimate:
0.192
other proportion estimate:
0.288
black proportion estimate:
0.1
asian proportion estimate:
0.317
```

We can see that the race proportion estimates are very close to the underlying population's

proportions. For example, we got 10.3% for Hispanic in our sample and the population proportion for Hispanic was 10%.

4.6 Sampling Distributions

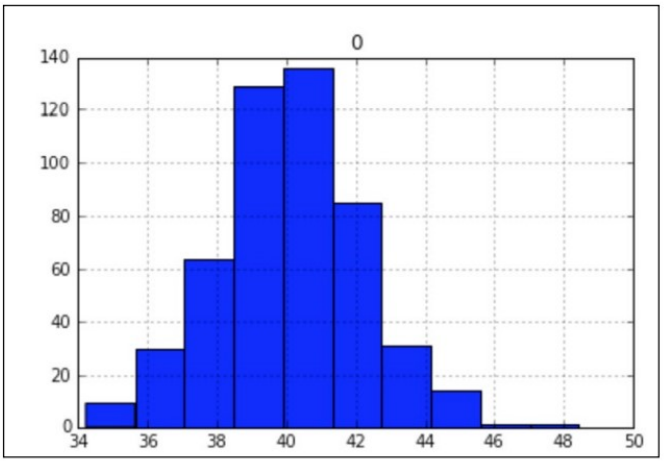
Take our employee break data for example, you might think I was just being fancy creating data using the Poisson distribution, but I had a reason for this—I specifically wanted non-normal data, as shown:
pd.DataFrame(breaks).hist(bins=50,range=(5,100)) As you can see, our data is definitely



not following a normal distribution, it appears to be bi-modal, which means that there are two peaks of break times, at around 25 and 70 minutes. As our data is not normal, many of the most popular statistics tests may not apply, however, if we follow the given procedure, we can create normal data! Think I'm crazy? Well, see for yourself.
First off, we will need to utilize what is known as a sampling distribution, which is a distribution of point estimates of several samples of the same size. Our procedure for creating a sampling distribution will be the following:
1. Take 500 different samples of the break times of size 100 each.
2. Take a histogram of these 500 different point estimates (revealing their distribution).
The number of elements in the sample (100) was arbitrary, but large enough to be a representative sample of the population. The number of samples I took (500) was also arbitrary, but large enough to ensure that our data would converge to a normal distribution:
point_estimates = []

```
for x in range(500): # Generate 500 samples
sample = np.random.choice(a= breaks, size=100)
#take a sample of 100 points
point_estimates.append( sample.mean() )
# add the sample mean to our list of point estimates
pd.DataFrame(point_estimates).hist()
# look at the distribution of our sample means
```

Behold! The sampling distribution of the sample mean appears to be normal even though



we took data from an underlying bimodal population distribution. It is important to note that the bars in this histogram represent the average break length of 500 samples of employees, where each sample has 100 people in it. In other words, a sampling distribution is a distribution of several point estimates.

Our data converged to a normal distribution because of something called the central limit theorem, which states that the sampling distribution (the distribution of point estimates) will approach a normal distribution as we increase the number of samples taken.

What's more, as we take more and more samples, the mean of the sampling distribution will approach the true population mean, as shown:

```
breaks.mean() - np.array(point_estimates).mean()
# .047 minutes difference
```

This is actually a very exciting result because it means that we can get even closer than a single point estimate by taking multiple point estimates and utilizing the central limit theorem!

4.7 Confidence intervals

While point estimates are okay estimates of a population parameter and sampling distributions are even better, there are the following two main issues with these approaches:

- Single point estimates are very prone to error (due to sampling bias among other things)
- Taking multiple samples of a certain size for sampling distributions might not be feasible, and may sometimes be even more infeasible than actually finding the population parameter.

For these reasons and more, we may turn to a concept, known as confidence interval, to find statistics.

A confidence interval is a range of values based on a point estimate that contains the true population parameter at some confidence level.

Confidence is an important concept in advanced statistics. Its meaning is sometimes misconstrued. Informally, a confidence level does not represent a "probability of being correct"; instead, it represents the frequency that the obtained answer will be accurate. For example, if you want to have a 95% chance of capturing the true population parameter using only a single point estimate, we would have to set our confidence level to 95%.

Calculating a confidence interval involves finding a point estimate, and then, incorporating a margin of error to create a range. The margin of error is a value that represents our certainty that our point estimate is accurate and is based on our desired confidence level, the variance of the data, and how big your sample is. There are many ways to calculate confidence intervals; for the purpose of brevity and simplicity, we will look at a single way of taking the confidence interval of a population mean. For this confidence interval, we need the following:

- A point estimate. For this, we will take our sample mean of break lengths from our previous example.
- An estimate of the population standard deviation, which represents the variance in the data.
  - This is calculated by taking the sample standard deviation (the standard deviation of the sample data) and dividing that number by the square root of the population size.
- The degrees of freedom (which is the -1 sample size).

Obtaining these numbers might seem arbitrary but, trust me, there is a reason for all of

them. However, again for simplicity, I will use prebuilt Python modules, as shown, to calculate our confidence interval and, then, demonstrate its value:

```
sample_size = 100
# the size of the sample we wish to take
sample = np.random.choice(a= breaks, size = sample_size)
# a sample of sample_size taken from the 9,000 breaks population from before
sample_mean = sample.mean()
# the sample mean of the break lengths sample
sample_stdev = sample.std()
# sample standard deviation
sigma = sample_stdev/math.sqrt(sample_size)
# population standard deviation estimate
stats.t.interval(alpha = 0.95, # Confidence level 95%
df= sample_size - 1, # Degrees of freedom
loc = sample_mean, # Sample mean
scale = sigma) # Standard deviation
# (36.36, 45.44)
```

To reiterate, this range of values (from 36.36 to 45.44) represents a confidence interval for the average break time with a 95% confidence.

We already know that our population parameter is 39,99, and note that the interval includes the population mean of 39.99.

I mentioned earlier that the confidence level was not a percentage of accuracy of our interval but the percent chance that the interval would even contain the population parameter at all.

To better understand the confidence level, let's take 10,000 confidence intervals and see how often our population mean falls in the interval. First, let's make a function, as illustrated, that makes a single confidence interval from our breaks data: # function to make confidence interval

```
def makeConfidenceInterval():
sample_size = 100
sample = np.random.choice(a= breaks, size = sample_size)
sample_mean = sample.mean()
```

sample mean

sample_stdev = sample.std()

sample standard deviation

sigma = sample_stdev/math.sqrt(sample_size)

population Standard deviation estimate

return stats.t.interval(alpha = 0.95, df= sample_size - 1, loc = sample_mean, scale = sigma)

Now that we have a function that will create a single confidence interval, let's create a procedure that will test the probability that a single confidence interval will contain the true population parameter, 39.99:

1. Take 10,000 confidence intervals of the sample mean.

2. Count the number of times that the population parameter falls into our confidence intervals.

3. Output the ratio of the number of times the parameter fell into the interval by 10,000:

times_in_interval = 0.

for i in range(10000):

interval = makeConfidenceInterval()

if 39.99 >= interval[0] and 39.99 <= interval[1]:

if 39.99 falls in the interval

times_in_interval += 1

print times_in_interval / 10000

0.9455 Success! We see that about 95% of our confidence intervals contained our actual population mean. Estimating population parameters through point estimates and confidence intervals is a relatively simple and powerful form of statistical inference. Let's also take a quick look at how the size of confidence intervals changes as we change our confidence level. Let's calculate confidence intervals for multiple confidence levels and look at how large the intervals are by looking at the difference between the two numbers. Our hypothesis will be that as we make our confidence level larger, we will likely see larger confidence intervals to be surer that we catch the true population parameter:

for confidence in (.5, .8, .85, .9, .95, .99):

confidence_interval = stats.t.interval(alpha = confidence, df= sample_size – 1, loc = sample_mean, scale = sigma)

length_of_interval = round(confidence_interval[1] - confidence_ interval[0], 2)
# the length of the confidence interval

print ”confidence {0} has a interval of size {1}”.
confidence 0.5 has an interval of size 2.56
confidence 0.8 has an interval of size 4.88
confidence 0.85 has an interval of size 5.49
confidence 0.9 has an interval of size 6.29
confidence 0.95 has an interval of size 7.51
confidence 0.99 has an interval of size 9.94
We can see that as we wish to be ”more confident” in our interval, our interval expands in order to compensate.
Next, we will take our concept of confidence levels and look at statistical hypothesis testing in order to both expand on these topics and also create (usually) even more powerful statistical inferences.
format(confidence, length_of_interval)

4.8 Hypothesis tests

Hypothesis tests are one of the most widely used tests in statistics. They come in many forms; however, all of them have the same basic purpose.
A hypothesis test is a statistical test that is used to ascertain whether we are allowed to assume that a certain condition is true for the entire population, given a data sample. Basically, a hypothesis test is a test for a certain hypothesis that we have about an entire population. The result of the test then tells us whether we should believe the hypothesis or reject it for an alternative one.
You can think of the hypothesis tests' framework to determine whether the observed sample data deviates from what was to be expected from the population itself. Now this sounds like a difficult task but, luckily, Python comes to the rescue and includes built-in libraries to conduct these tests easily.

A hypothesis test generally looks at two opposing hypotheses about a population. We call them the null hypothesis and the alternative hypothesis. The null hypothesis is the statement being tested and is the default correct answer; it is our starting point and our original hypothesis. The alternative hypothesis is the statement that opposes the null hypothesis. Our test will tell us which hypothesis we should trust and which we should reject.

Based on sample data from a population, a hypothesis test determines whether or not to reject the null hypothesis. We usually use a p-value (which is based on our significance level) to make this conclusion.

The following are some examples of questions you can answer with a hypothesis test:

• Does the mean break time of employees differ from 40 minutes?

• Is there a difference between people who interacted with website A and people who interacted with website B (A/B testing)?

• Does a sample of coffee beans vary significantly in taste from the entire population of beans?

4.8.1 Conducting a hypothesis test

There are multiple types of hypothesis tests out there, and among them are dozens of different procedures and metrics. Nonetheless, there are five basic steps that most hypothesis tests follow, which are as follows:

1. Specify the hypotheses:

° Here, we formulate our two hypotheses: the null and the alternative.

° We usually use the notation of H0 to represent the null hypothesis and Ha to represent our alternative hypothesis

2. Determine the sample size for the test sample:

° This calculation depends on the chosen test. Usually, we have to determine a proper sample size in order to utilize theorems, such as the central limit theorem, and assume the normality of data.

3. Choose a significance level (usually called alpha or):

° A significance level of 0.05 is common

4. Collect the data:

° They collect a sample of data to conduct the test

5. Decide whether to reject or fail to reject the null hypothesis:

° This step changes slightly based on the type of test being used.

The final result will either yield in rejecting the null hypothesis in favor of the alternative or failing to reject the null hypothesis.

In this chapter, we will look at the following three types of hypothesis tests:

- One-sample t-tests
- Chi-square goodness of fit
- Chi-square test for association/independence

There are many more tests. However, these three are a great combination of distinct, simple, and powerful tests. One of the biggest things to consider when choosing which test we should implement is the type of data we are working with, specifically, are we dealing with continuous or categorical data. In order to truly see the effects of a hypothesis, I suggest we dive right into an example. First, let's look at the use of a t-tests to deal with continuous data.

4.8.2 One sample t-tests

The one sample t-test is a statistical test used to determine whether a quantitative (numerical) data sample differs significantly from another dataset (the population or another sample). Suppose, in our previous employee break time example, we look, specifically, at the engineering department's break times, as shown:

long_breaks_in_engineering = stats.poisson.rvs(loc=10, mu=55, size=100)

short_breaks_in_engineering = stats.poisson.rvs(loc=10, mu=15, size=300)

engineering_breaks = np.concatenate((long_breaks_in_engineering, short_breaks_in_engineering))

print breaks.mean()

39.99

print engineering_breaks.mean()

34.825

Note that I took the same approach as making the original break times, but with the following two differences:

• I took a smaller sample from the Poisson distribution (to simulate that we took a sample of 400 people from the engineering department)
• Instead of using a mu of 60 as before, I used 55 to simulate the fact that the engineering department's break behavior isn't exactly the same as the company's behavior as a whole.
It is easy to see that there seems to be a difference (of over 5 minutes) between the engineering department and the company as a whole. We usually don't have the entire population and the population parameters at our disposal, but I have them simulated in order to see the example work. So, even though we (the omniscient readers) can see a difference, we will assume that we know nothing of these population parameters and, instead, rely on a statistical test in order to ascertain these differences.

4.8.2.1 Example of a one sample t-tests

Our objective here is to ascertain whether there is a difference between the overall population's (company employees) break times and break times of employees in the engineering department.
Let us now conduct a t-test at a 95% confidence level in order to find a difference (or not!). Technically speaking, this test will tell us if the sample comes from the same distribution as the population.

4.8.2.2 Assumptions of the one sample t-tests

Before diving into the five steps, we must first acknowledge that t-tests must satisfy the following two conditions to work properly:
• The population distribution should be normal, or the sample should be large (n 30).
• In order to make the assumption that the sample is independently randomly sampled, it is sufficient to enforce that the population size should be at least 10 times larger than the sample size (10n < N).
Note that our test requires that either the underlying data be normal (which we know is not true for us), or that the sample size be at least 30 points large. For the t-test, this condition is sufficient to assume normality. This test also requires independence, which is satisfied by taking a sufficiently small sample. Sounds weird, right? The basic idea is that

our sample must be large enough to assume normality (through conclusions similar to the central limit theorem) but small enough as to be independent from the population.

Now, let's follow our five steps:

1. Specify the hypotheses.

We will let H0 = the engineering department takes breaks the same as the company as a whole

If we let this be the company average, we may write: H0 :

Note how this is our null, or default, hypothesis. It is what we would assume, given no data. What we would like to show is the alternative hypothesis. Now that we actually have some options for our alternative, we could either say that the engineering mean (let's call it that) is lower than the company average, higher than the company average, or just flat out different (higher or lower) than the company average:

° If we wish to answer the question, is the sample mean different from the company average, then this is called a two-tailed test and our alternative hypothesis would be as follows:

Ha:

° If we want to answer either is the sample mean lower than the company average or is the sample mean higher than the company average, then we are dealing with a one-tailed test and our alternative hypothesis would be one or the other of the following hypotheses:

Ha:(engineering takes longer breaks)

Ha:(engineering takes shorter breaks)

The difference between one and two tails is the difference of dividing a number later on by 2 or not. The process remains completely unchanged for both. For this example, let's choose the two-tailed test. So, we are testing for whether or not this sample of the engineering department's average break times is different from the company average.

2. Determine the sample size for the test sample.

As mentioned earlier, most tests (including this one) make the assumption that either the underlying data is normal or that our sample is in the right range.

° The sample is at least 30 points (it is 400)

° The sample is less than 10% of the population (which would be 900 people)

3. Choose a significance level (usually called alpha or).

We will choose a 95% significance level, which means that our alpha would actually be 1

- .95 = .05

4. Collect the data.

Done! This was generated through the two Poisson distributions.

5. Decide whether to reject or fail to reject the null hypothesis.

As mentioned before, this step varies based on the test used. For a one sample t-test, we must calculate two numbers: the test statistic and our p value. Luckily, we can do this in one line in Python.

A test statistic is a value that is derived from sample data during a type of hypothesis test. They are used to determine whether or not to reject the null hypothesis.

The test statistic is used to compare the observed data with what is expected under the null hypothesis. The test statistic is used in conjunction with the p-value.

The p-value is the probability that the observed data occurred this way by chance. When the data is showing very strong evidence against the null hypothesis, the test statistic becomes large (either positive or negative) and the p-value usually becomes very small, which means that our test is showing powerful results and what is happening is, probably, not happening by chance.

In the case of a t-test, a t value is our test statistic, as shown:

t_statistic, p_value = stats.ttest_1samp(a= engineering_breaks, popmean= breaks.mean())

We input the engineering_breaks variable (which holds 400 break times) and the population mean, and we obtain the following numbers:

t_statistic == -5.742

p_value == .00000018

The test result shows that the t value is -5.742. This is a standardized metric that reveals the deviation of the sample mean from the null hypothesis. The p value is what gives us our final answer. Our p-value is telling us how often our result would appear by chance. So, for example, if our p-value was .06, then that would mean we should expect to observe this data by chance about 6% of the time. This means that about 6% of samples would yield results like this.

We are interested in how our p-value compares to our significance level:

- If the p-value is less than the significance level, then we can reject the null hypothesis
- If the p-value is greater than the significance level, then we failed to reject the null hypothesis

Our p value is way lower than .05 (our chosen significance level), which means that we may reject our null hypothesis in favor for the alternative. This means that the engineering department seems to take different break lengths than the company as a whole!
There are many other types of t-tests, including one-tailed tests (mentioned before) and paired tests as well as two sample t-tests (both not mentioned yet). These procedures can be readily found in statistics literature; however, we should look at something very important—what happens when we get it wrong.

4.8.2.3 Type I and type II errors

We've mentioned both the type I and type II errors in a previous chapter about probability in the examples of a binary classifier, but they also apply to hypothesis tests. A type I error occurs if we reject the null hypothesis when it is actually true. This is also known as a false positive. The type I error rate is equal to the significance level , which means that if we set a higher confidence level, for example, a significance level of 99%, our is .01, and therefore our false positive rate is 1%.
A type II error occurs if we fail to reject the null hypothesis when it is actually false. This is also known as a false negative. The higher we set our confidence level, the more likely we are to actually see a type II error.

4.8.2.4 Hypothesis test for categorical variables

T-tests (among other tests) are hypothesis tests that work to compare and contrast quantitative variables and underlying population distributions. In this section, we will explore two new tests, both of which serve to explore qualitative data. They both are a form of test called chi-square tests. These two tests will perform the following two tasks for us:

- Determine whether a sample of categorical variables is taken from a specific population (similar to the t-test)
- Determine whether two variables affect each other and are associated to each other.

4.8.2.5 Chi-square goodness of fit test

The one-sample t-test was used to check whether a sample mean differed from the population mean. The chi-square goodness of fit test is very similar to the one sample t-test in that it tests whether the distribution of the sample data matches an expected distribution, while the big difference is that it is testing for categorical variables. For example, a chi-square goodness of fit test would be used to see if the race demographics of your company match that of the entire city of the U.S. population. It can also be used to see if users of your website show similar characteristics to average Internet users.

As we are working with categorical data, we have to be careful because categories like "male", "female," or "other" don't have any mathematical meaning. Therefore, we must consider counts of the variables rather than the actual variables themselves. In general, we use the chi-square goodness of fit test in the following cases:

- We want to analyze one categorical variable from one population
- We want to determine if a variable fits a specified or expected distribution

In a chi-square test, we compare what is observed to what we expect.

4.8.2.6 Assumptions of the chi-square goodness of fit test

There are two usual assumptions of this test, as follows:

- All the expected counts are at least 5
- Individual observations are independent and the population should be at least 10 times as large as the sample, ($10n < N$)

The second assumption should look familiar to the t-test; however, the first assumption should look foreign. Expected counts are something we haven't talked about yet but are about to!

When formulating our null and alternative hypotheses for this test, we consider a default distribution of categorical variables. For example, if we have a die and we are testing whether or not the outcomes are coming from a fair die, our hypothesis might look as follows:

H0 : The specified distribution of the categorical variable is correct.

$p1 = 1/6$, $p2 = 1/6$, $p3 = 1/6$, $p4 = 1/6$, $p5 = 1/6$, $p6 = 1/6$

Our alternative hypothesis is quite simple, as shown:

Ha : The specified distribution of the categorical variable is not correct. At least one of the pi values is not correct.

In the t-test, we used our test statistic (the t value) to find our p-value. In a chi-square test, our test statistic is, well, chi-square.

Test Statistic: χ^2 = over k categories

Degrees of Freedom = k − 1

A critical value is when we use χ^2 as well as our degrees of freedom and our significance level, and then reject the null hypothesis if the p-value is below our significance level (the same as before).

Let's see an example to understand further.

4.8.2.7 Example of a chi-square test for goodness of fit

The CDC categorizes adult BMIs into four classes: Under/Normal, Over weight, Obesity, and Extreme Obesity. A 2009 survey showed the distribution for adults in the U.S. to be 31.2%, 33.1%, 29.4%, and 6.3% respectively. A total of 500 adults are randomly sampled and their BMI categories are recorded. Is there evidence to suggest that BMI trends have changed since 2009? Test at the 0.05 significance level. First, let's calculate our expected

	Under/Normal	Over	Obesity	Extreme Obesity	Total
Observed	102	178	186	34	500

values. In a sample of 500, we expect 156 to be Under/Normal (that's 31.2% of 500), and we fill in the remaining boxes in the same way. First, check the conditions:

	Under/Normal	Over	Obesity	Extreme Obesity	Total
Observed	102	178	186	34	500
Expected	156	165.5	147	31.5	500

- All of the expected counts are greater than 5
- Each observation is independent and our population is very large (much more than 10

times of 500 people)

Next, carry out a goodness of fit test. We will set our null and alternative hypotheses:

- H0 : The 2009 BMI distribution is still correct.
- Ha : The 2009 BMI distribution is no longer correct (at least one of the proportions is different now). We can calculate our test statistic by hand:

Alternatively, we can use our handy dandy Python skills, as shown:

$$\text{Test Statistic: } \chi^2 = \sum \frac{(Observed - Expected)^2}{Expected} \text{ for } df = 3$$
$$= \frac{(102-156)^2}{156} + \frac{(178-165.5)^2}{165.5} + \frac{(186-147)^2}{147} + \frac{(34-31.5)^2}{31.5} = 30.18$$

observed = [102, 178, 186, 34]

expected = [156, 165.5, 147, 31.5]

chi_squared, p_value = stats.chisquare(f_obs= observed, f_exp= expected) chi_squared, p_value

#(30.1817679275599, 1.26374310311106e-06)

Our p-value is lower than .05; therefore, we may reject the null hypothesis in favor of the fact that the BMI trends today are different from what they were in 2009.

4.8.2.8 Chi-square test for association/independence

Independence as a concept in probability is when knowing the value of one variable tells you nothing about the value of another. For example, we might expect that the country and the month you were born in are independent. However, knowing which type of phone you use might indicate your creativity levels. Those variables might not be independent. The chi-square test for association/independence helps us ascertain whether two categorical variables are independent of one another. The test for independence is commonly used to determine whether variables like education levels or tax brackets vary based on demographic factors, such as gender, race, and religion. Let's look back at an example posed in the preceding chapter, the A/B split test.

Recall that we ran a test and exposed half of our users to a certain landing page (Website A), exposed the other half to a different landing page (Website B), and then, measured the sign up rates for both sites. We obtained the following results: We calculated website

	Did not sign up	**Signed up**
Website A	134	54
Website B	110	48

Results of our A/B test

conversions but what we really want to know is whether there is a difference between the two variables: which website was the user exposed to? and did the user sign up?. For this, we will use our chi-square test.

4.8.2.9 Assumptions of the chi-square independence test

There are the following two assumptions of this test:

- All expected counts are at least 5
- Individual observations are independent and the population should be at least 10 times as large as the sample, (10n < N)

Note that they are exactly the same as the last chi-square test. Let's set up our hypotheses:

- H0 : There is no association between two categorical variables in the population of interest
- H0 : Two categorical variables are independent in the population of interest
- Ha : There is an association between two categorical variables in the population of interest
- Ha : Two categorical variables are not independent in the population of interest

You might notice that we are missing something important here. Where are the expected counts? Earlier, we had a prior distribution to compare our observed results to but now we do not. For this reason, we will have to create some. We can use the following formula to calculate the expected values for each value. In each cell of the table, we can use:
Expected Count = to calculate our chi-square test statistic and our degrees of freedom Here, r is the number of rows and c is the number of columns. Of course, as before, when we calculate our p-value, we will reject the null if that p-value is less than the significance level. Let's use some built-in Python methods, as shown, in order to quickly get our results:
observed = np.array([[134, 54],[110, 48]])

$$\text{Test Statistic: } \chi^2 = \sum \frac{(Observed_{r,c} - Expected_{r,c})^2}{Expected_{r,c}}$$
$$\text{over } r \text{ rows and } c \text{ columns}$$
$$\text{Degrees of Freedom} = (r-1)\cdot(c-1)$$

built a 2x2 matrix as seen in the table above

chi_squared, p_value, degrees_of_freedom, matrix = stats.chi2_ contingency(observed= observed) chi_squared, p_value

(0.04762692369491045, 0.82724528704422262)

We can see that our p-value is quite large; therefore, we fail to reject our null hypothesis and we cannot say for sure that seeing a particular website has any effect on a user's sign up. There is no association between these variables.

Chapter 5

COMMUNICATING DATA

This chapter deals with the different ways of communicating results from our analysis. Here, we will look at different presentation styles as well as visualization techniques. The point of this chapter is to take our results and be able to explain them in a coherent, intelligible way so that anyone, whether they are data savvy or not, may understand and use our results.

Much of what we will discuss will be how to create effective graphs through labels, keys, colors, and more. We will also look at more advanced visualization techniques, such as parallel coordinate plots.

In this chapter, we will look into the following topics:

- Identifying effective and ineffective visualizations
- Recognizing when charts are attempting to ”trick” the audience
- Being able to identify causation versus correlation
- Constructing appealing visuals that offer valuable insight

5.1 Why does communication matter?

Being able to conduct experiments and manipulate data in a coding language is not enough to conduct practical and applied data science. This is because data science is, generally, only as good as how it is used in practice. For instance, a medical data scientist might be

able to predict the chance of a tourist contracting Malaria in developing countries with >98% accuracy, however, if these results are published in a poorly marketed journal and online mentions of the study are minimal, their groundbreaking results that could potentially prevent deaths would never see the true light of day.

For this reason, communication of results is arguably as important as the results themselves. A famous example of poor management of distribution of results is the case of Gregor Mendel. Mendel is widely recognized as one of the founders of modern genetics. However, his results (including data and charts) were not well adopted until after his death. Mendel even sent them to Charles Darwin, who largely ignored Mendel's papers, which were written in unknown Moravian journals.

Generally, there are two ways of presenting results: verbal and visual. Of course, both the verbal and visual forms of communication can be broken down into dozens of subcategories, including slide decks, charts, journal papers, and even university lectures. However, we can find common elements of data presentation that can make anyone in the field more aware and effective in their communication skills.

Let's dive right into effective (and ineffective) forms of communication, starting with visuals.

5.2 Identifying effective and ineffective visualizations

The main goal of data visualization is to have the reader quickly digest the data, including possible trends, relationships, and more. Ideally, a reader will not have to spend more than 5-6 seconds digesting a single visualization. For this reason, we must make visuals very seriously and ensure that we are making a visual as effective as possible. Let's look at four basic types of graphs: scatter plots, line graphs, bar charts, histograms, and box plots.

5.2.1 Scatter plots

A scatter plot is probably one of the simplest graphs to create. It is made by creating two quantitative axes and using data points to represent observations. The main goal of a scatter plot is to highlight relationships between two variables and, if possible, reveal

a correlation. For example, we can look at two variables: average hours of TV watched in a day and a 0-100 scale of work performance (0 being very poor performance and 100 being excellent performance). The goal here is to find a relationship (if it exists) between watching TV and average work performance.

The following code simulates a survey of a few people, in which they revealed the amount of television they watched, on an average, in a day against a company standard work performance metric:

import pandas as pd hours_tv_watched = [0, 0, 0, 1, 1.3, 1.4, 2, 2.1, 2.6, 3.2, 4.1, 4.4, 4.4, 5]

This line of code is creating 14 sample survey results of people answering the question of how many hours of TV they watch in a day.

work_performance = [87, 89, 92, 90, 82, 80, 77, 80, 76, 85, 80, 75, 73, 72] This line of code is creating 14 new sample survey results of the same people being rated on their work performance on a scale from 0 to 100.

For example, the first person watched 0 hours of TV a day and was rated 87/100 on their work, while the last person watched, on an average, 5 hours of TV a day and was rated 72/100:

df = pd.DataFrame({'hours_tv_watched':hours_tv_watched, 'work_ performance':work_performance})

Here, we are creating a Dataframe in order to ease our exploratory data analysis and make it easier to make a scatter plot:

df.plot(x='hours_tv_watched', y='work_performance', kind='scatter')

Now, we are actually making our scatter plot. In the following plot, we can see that our axes represent the number of hours of TV watched in a day and the person's work performance metric:

Each point on a scatter plot represents a single observation (in this case a person) and its location is a result of where the observation stands on each variable. This scatter plot does seem to show a relationship, which implies that as we watch more TV in the day, it seems to affect our work performance.

Of course, as we are now experts in statistics from the last two chapters, we know that

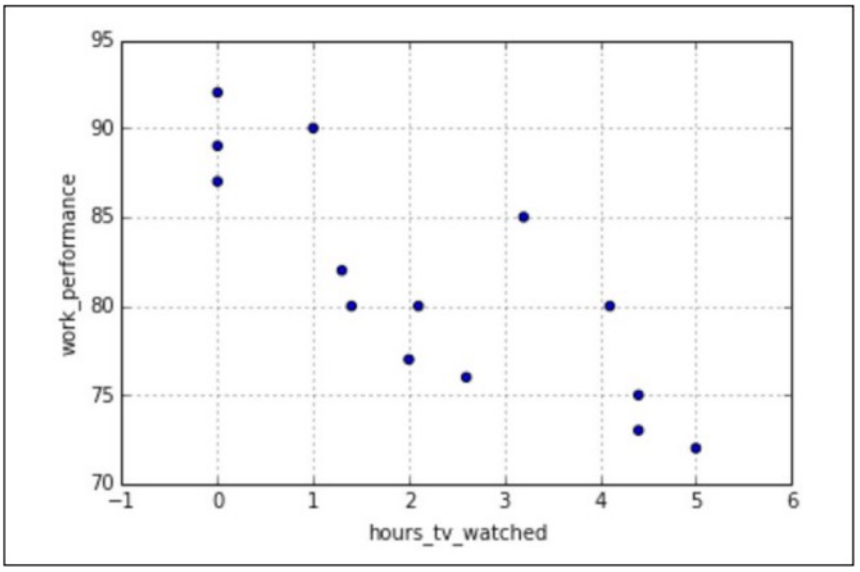


Figure 5.1: Scatter plot

this might not be causational. A scatter plot may only work to reveal a correlation or an association between but not a causation.

5.2.2 Line Graphs

Line graphs are, perhaps, one of the most widely used graphs in data communication. A line graph simply uses lines to connect data points and usually represents time on the x axis. Line graphs are a popular way to show changes in variables over time. The line graph, like the scatter plot, is used to plot quantitative variables.

As a great example, many of us wonder about the possible links between what we see on TV and our behavior in the world. A friend of mine once took this thought to an extreme—he wondered if he could find a relationship between the TV show, The X-Files, and the amount of UFO sightings in the U.S.. He then found the number of sightings of UFOs per year and plotted them over time. He then added a quick graphic to ensure that readers would be able to identify the point in time when the X-files were released:

It appears to be clear that right after 1993, the year of the X-Files premier, the number of UFO sightings started to climb drastically.

This graphic, albeit light-hearted, is an excellent example of a simple line graph. We are told what each axis measures, we can quickly see a general trend in the data, and we can

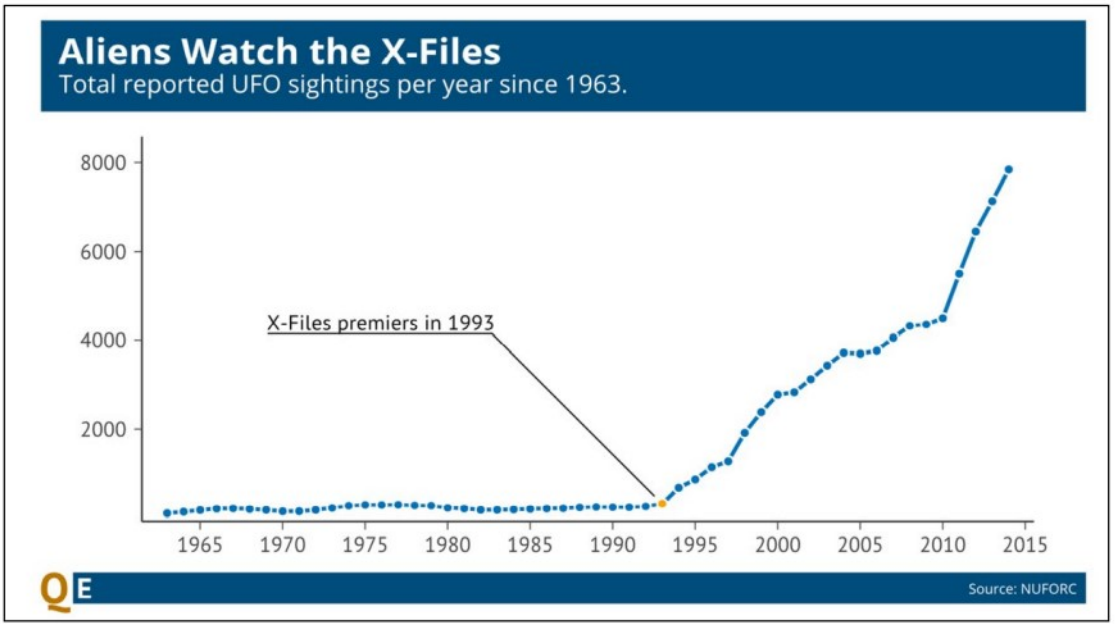


Figure 5.2: Line graph of Aliens watch the X-files

identify with the author's intent, which is to show a relationship between the number of UFO sightings and the X-files premier.

On the other hand, the following is a less impressive line chart:

This line graph attempts to highlight the change in the price of gas by plotting three

Figure 5.3: Line chart for Cost of Gas

points in time. At first glance, it is not much different than the previous graph—we have

time on the bottom x axis and a quantitative value on the vertical y axis. The (not so) subtle difference here is that the three points are equally spaced out on the x axis; however, if we read their actual time indications, they are not equally spaced out in time. A year separates the first two points whereas a mere 7 days separates the last two points.

5.2.3 Bar Charts

We generally turn to bar charts when trying to compare variables across different groups. For example, we can plot the number of countries per continent using a bar chart. Note how the x axis does not represent a quantitative variable, in fact, when using a bar chart, the x axis is generally a categorical variable, while the y axis is quantitative.
Note that, for this code, I am using the World Health Organization's report on alcohol consumption around the world by country:
drinks = pd.read_csv('data/drinks.csv')
drinks.continent.value_counts().plot(kind='bar', title='Countries per Continent')
plt.xlabel('Continent')
plt.ylabel('Count')
The following graph shows us a count of the number of countries in each continent. We can see the continent code at the bottom of the bars and the bar height represents the number of countries we have in each continent. For example, we see that Africa has the most countries represented in our survey, while South America has the least:

In addition to the count of countries, we can also plot the average beer servings per continent using a bar chart, as shown:
drinks.groupby('continent').beer_servings.mean().plot(kind='bar') Note how a scatter plot or a line graph would not be able to support this data because they can only handle quantitative variables; bar graphs have the ability to demonstrate categorical values.
We can also use bar charts to graph variables that change over time, like a line graph.

5.2.4 Histograms

Histograms show the frequency distribution of a single quantitative variable by splitting up the data, by range, into equidistant bins and plotting the raw count of observations in

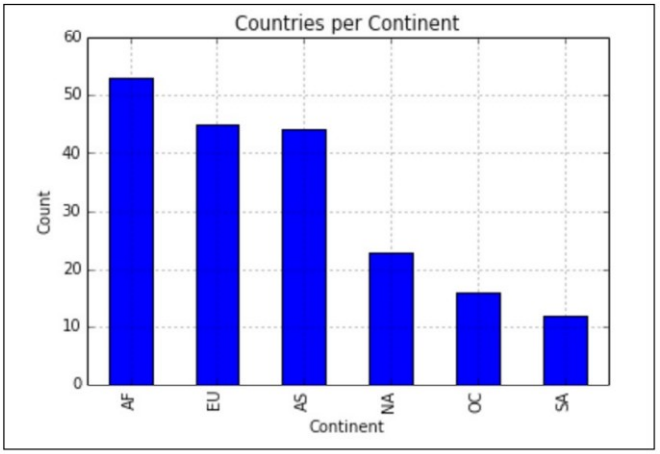


Figure 5.4: Bar chart 1

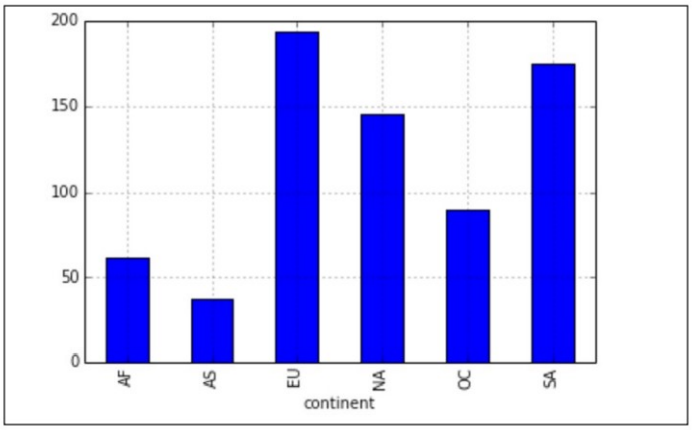


Figure 5.5: Bar chart 2

each bin. A histogram is effectively a bar chart where the x axis is a bin (subrange) of values and the y axis is a count. As an example, I will import a store's daily number of unique customers, as shown:

rossmann_sales = pd.read_csv('data/rossmann.csv')

rossmann_sales.head() Note how we have multiple store data (by the first Store column). Let's subset this data for only the first store, as shown:

first_rossmann_sales = rossmann_sales[rossmann_sales['Store']==1]

first_rossmann_sales = rossmann_sales[rossmann_sales['Store']==1]

first_rossmann_sales['Customers'].hist(bins=20)

plt.xlabel('Customer Bins')

plt.ylabel('Count')

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
0	1	5	2015-07-31	5263	555	1	1	0	1
1	2	5	2015-07-31	6064	625	1	1	0	1
2	3	5	2015-07-31	8314	821	1	1	0	1
3	4	5	2015-07-31	13995	1498	1	1	0	1
4	5	5	2015-07-31	4822	559	1	1	0	1

Figure 5.6: Table for count

The x axis is now categorical in that each category is a selected range of values, for

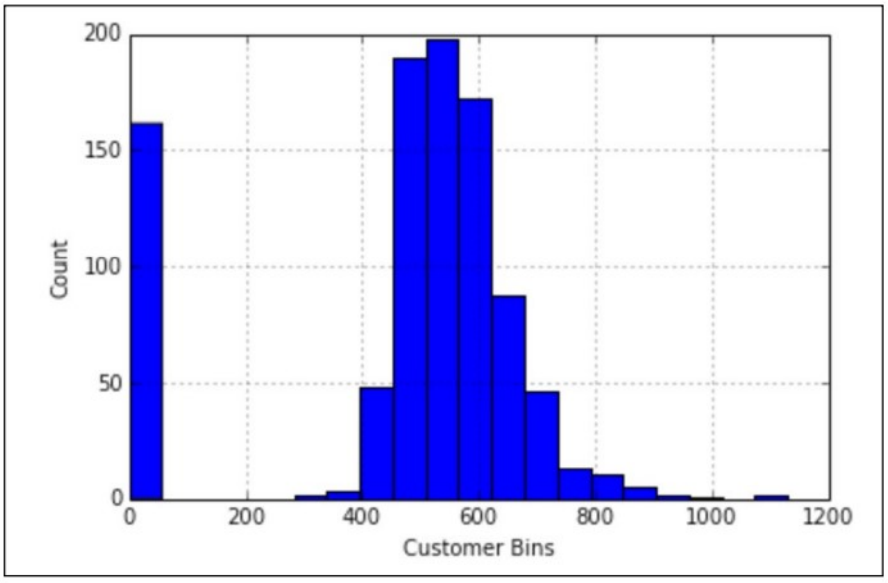


Figure 5.7: Histogram

example, 600-620 customers would potentially be a category. The y axis, like a bar chart, is plotting the number of observations in each category. In this graph, for example, one might take away the fact that most of the time, the number of customers on any given day will fall between 500 and 700.

Altogether, histograms are used to visualize the distribution of values that a quantitative variable can take on.

5.2.5 Box Plots

Box plots are also used to show a distribution of values. They are created by plotting the five number summary, as follows:

• The minimum value • The first quartile (the number that separates the 25% lowest values from the rest)

• The median

• The third quartile (the number that separates the 25% highest values from the rest)

• The maximum value

In Pandas, when we create box plots, the red line denotes the median, the top of the box (or the right if it is horizontal) is the third quartile, and the bottom (left) part of the box is the first quartile.

The following is a series of box plots showing the distribution of beer consumption according to continents:

drinks.boxplot(column='beer_servings', by='continent') Now, we can clearly see the dis-

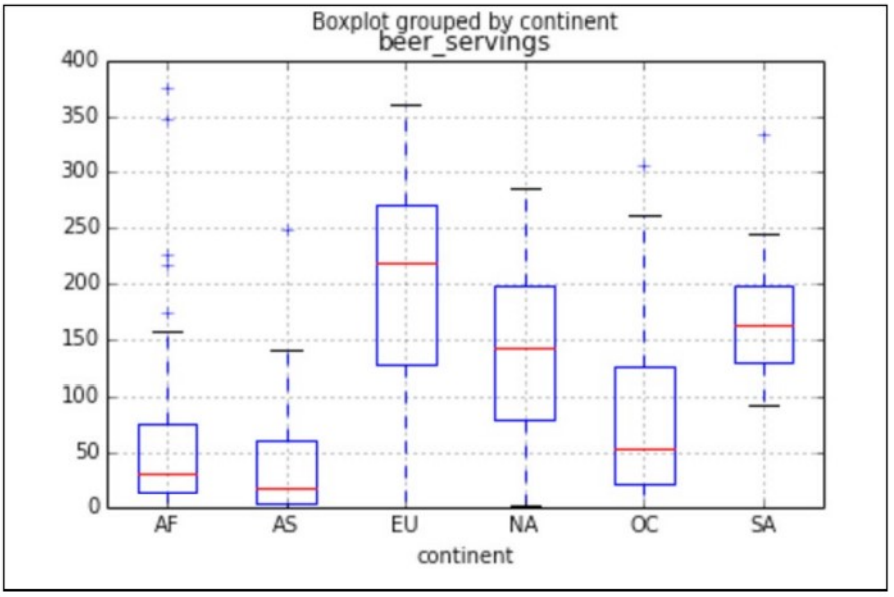


Figure 5.8: Box Plots 1

tribution of beer consumption across the seven continents and how they differ. Africa and Asia have a much lower median of beer consumption than Europe or North America.

Box plots also have the added bonus of being able to show outliers much better than a

histogram. This is because the minimum and maximum are parts of the box plot.

Getting back to the customer data, let's look at the same store customer numbers, but using a box plot:

first_rossmann_sales.boxplot(column='Customers', vert=False)

This is the exact same data as plotted earlier in the histogram; however, now it is shown

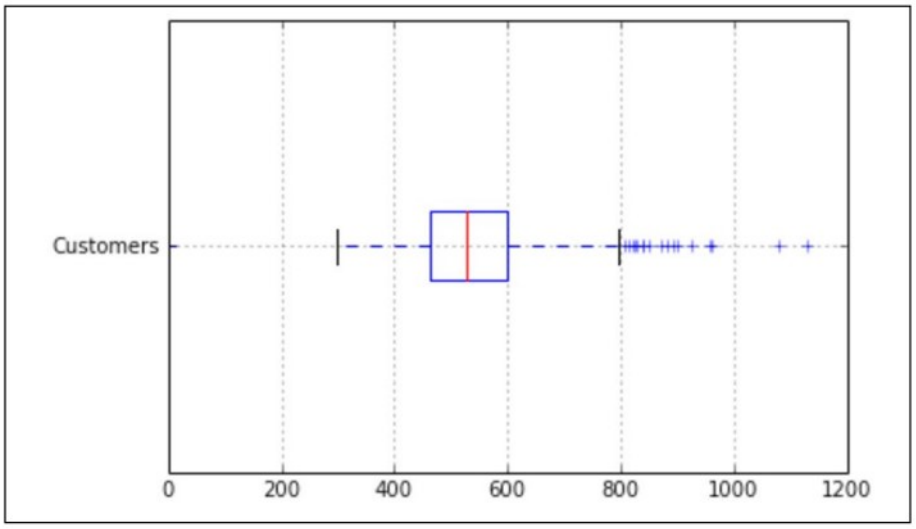


Figure 5.9: Box Plots 2

as a box plot. For the purpose of comparison, I will show you both the graphs one after the other: Note how the x axis for each graph are the same, ranging from 0 to 1,200. The box plot is much quicker at giving us a center of the data, the red line is the median, while the histogram works much better in showing us how spread out the data is and where people's biggest bins are. For example, the histogram reveals that there is a very large bin of zero people. This means that for a little over 150 days of data, there were zero customers.

Note that we can get the exact numbers to construct a box plot using the describe feature in Pandas, as shown:

first_rossmann_sales['Customers'].describe()

min 0.000000 25% 463.000000

50% 529.000000

75% 598.750000

max 1130.000000

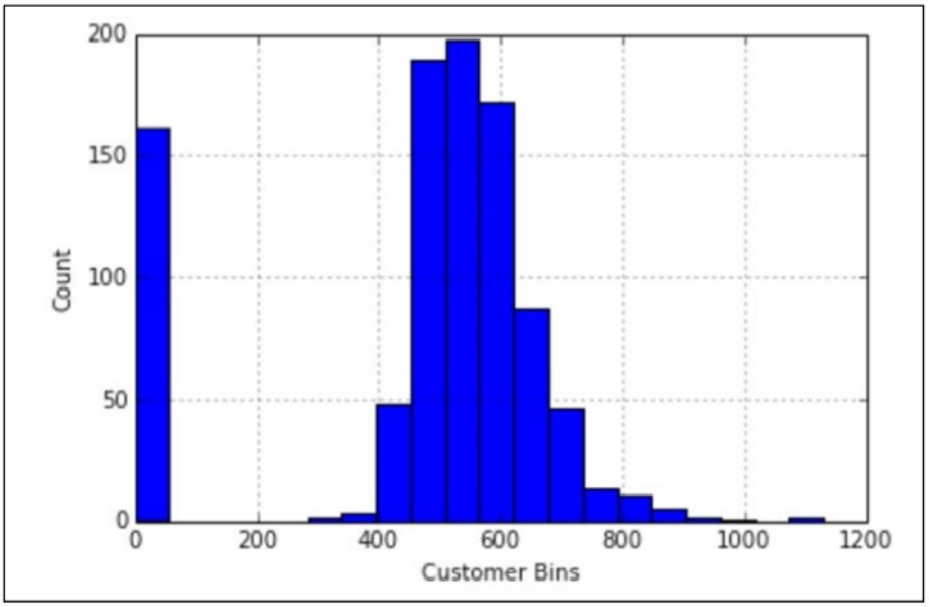


Figure 5.10: Chart for count and customer Bins

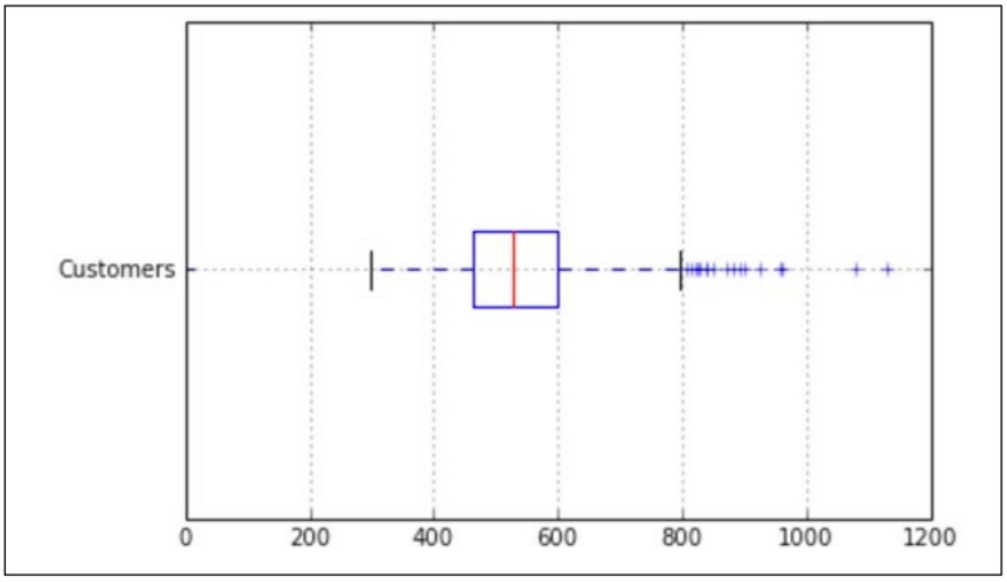


Figure 5.11: Box Plots 3

5.3 When graphs and statistics lie

I should be clear, statistics don't lie, people lie. One of the easiest ways to trick your audience is to confuse correlation with causation.

5.3.1 Correlation versus causation

I don't think I would be allowed to publish this book without taking a deeper dive into the differences between correlation and causation. For this example, I will continue to use my data of TV consumption and work performance.

Correlation is a quantitative metric between -1 and 1 that measures how two variables move with each other. If two variables have a correlation close to -1, it means that as one variable increases, the other decreases, and if two variables have a correlation close to +1, it means that those variables move together in the same direction—as one increases, so does the other, and vice versa.

Causation is the idea that one variable affects another.

For example, we can look at two variables: the average hours of TV watched in a day and a 0-100 scale of work performance (0 being very poor performance and 100 being excellent performance). One might expect that these two factors are negatively correlated, which means that as the number of hours of TV watched increases in a 24 hour day, your overall work performance goes down. Recall the code from earlier, which is as follows:

import pandas as pd

hours_tv_watched = [0, 0, 0, 1, 1.3, 1.4, 2, 2.1, 2.6, 3.2, 4.1, 4.4, 4.4, 5]

Here, I am looking at the same sample of 14 people as before and their answers to the question, how many hours of TV do you watch on average per night:

work_performance = [87, 89, 92, 90, 82, 80, 77, 80, 76, 85, 80, 75, 73, 72]

These are the same 14 people as mentioned earlier, in the same order, but now, instead of the number of hours they watched TV, we have their work performance as graded by the company or a third-party system:

df = pd.DataFrame({'hours_tv_watched':hours_tv_watched, 'work_ performance':work_performance}) df = pd.DataFrame({'hours_tv_watched':hours_tv_watched, 'work_ performance':work_performance}) Now we can introduce a new line of code that shows us the correlation between these two variables:

df.corr() # -0.824

Recall that a correlation, if close to -1, implies a strong negative correlation, while a correlation close to +1 implies a strong positive correlation.

This number helps support the hypothesis because a correlation coefficient close to -1 implies not only a negative correlation, but a strong one at that. Again, we can see this via a scatter plot between the two variables. So, both our visual and our numbers are aligned with each other. This is an important concept that should be true when communicating results. If your visuals and your numbers are off, people are less likely to take your analysis seriously: I cannot stress enough that correlation and causation are different from

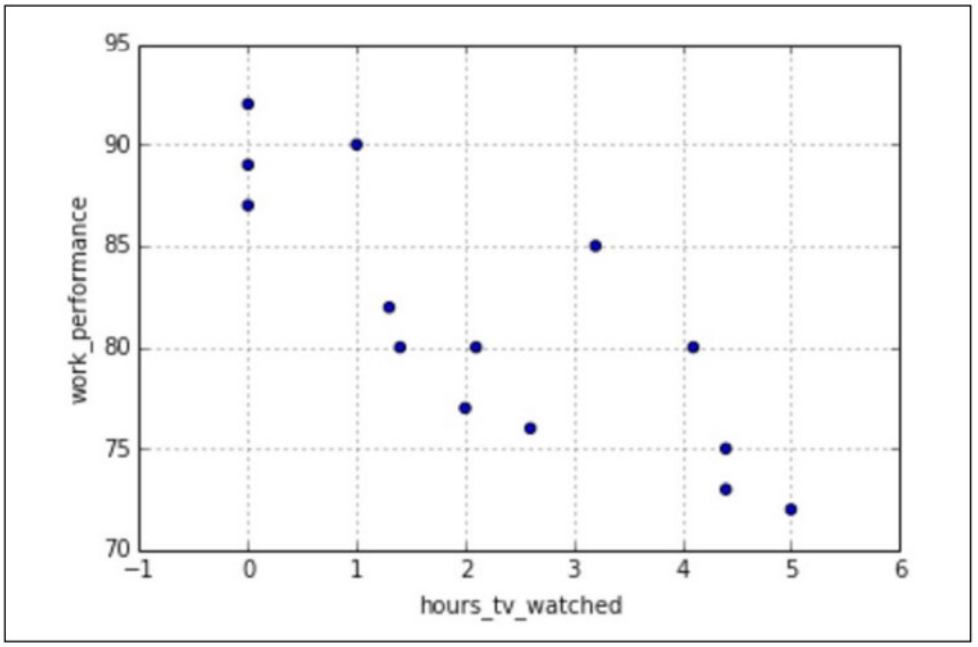


Figure 5.12: Correlation

each other. Correlation simply quantifies the degree to which variables change together, whereas causation is the idea that one variable actually determines the value of another. If you wish to share the results of your findings of your correlational work, you might be met with challengers in the audience asking for more work to be done. What is more terrifying is that no one might know that the analysis is incomplete and you may make actionable decisions based on simple correlational work.

It is very often the case that two variables might be correlated to each other but they do not have any causation between them. This can be for a variety of reasons, some of which are as follows:

- There might be a confounding factor between them. This means that there is a third lurking variable that is not being factored and that acts as a bridge between the two

variables. For example, previously, we showed that you might find that the amount of TV you watch is negatively correlated with work performance, that is, as the number of hours of TV you watch increases, your overall work performance may decrease. That is a correlation. It doesn't seem quite right to suggest that watching TV is the actual cause of a decrease in the quality of work performance. It might seem more plausible to suggest that there is a third factor, perhaps hours of sleep every night, that might answer this question. Perhaps, watching more TV decreases the amount of time you have for sleep, which in turn limits your work performance. The number of hours of sleep per night is the confounding factor.

- They might not have anything to do with each other! It might simply be a coincidence. There are many variables that are correlated but simply do not cause each other. Consider the following example: It is much more likely that these two variables only happen to

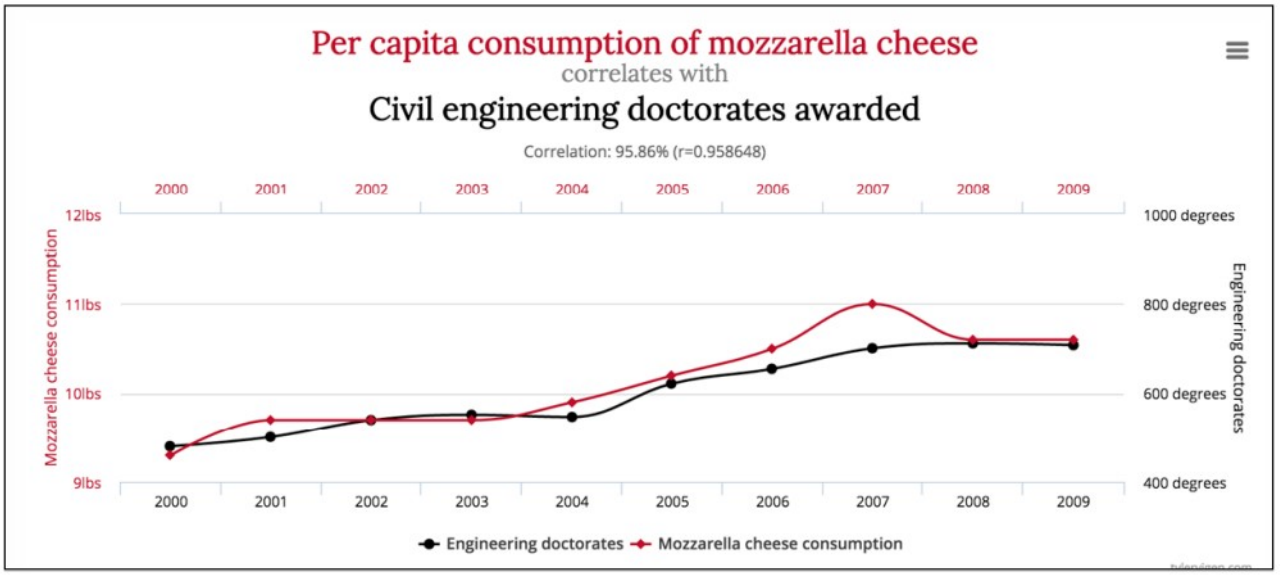


Figure 5.13: Correlation but causation

correlate (more strongly than our previous example, I may add) that cheese consumption determines the number of civil engineering doctorates in the world.

You have likely heard the statement correlation does not imply causation and the last graph is exactly the reason why data scientists must believe that. Just because there exists a mathematical correlation between variables does not mean they have causation between them. There might be confounding factors between them or they just might not

have anything to do with each other! Let's see what happens when we ignore confounding variables and correlations become extremely misleading.

5.4 Simpson's paradox

Simpson's paradox is a formal reason for why we need to take confounding variables seriously. The paradox states that a correlation between two variables can be completely reversed when we take different factors into account. This means that even if a graph might show a positive correlation, these variables can become anti-correlated when another factor (most likely a confounding one) is taken into consideration. This can be very troublesome to statisticians.

Si refers to a situation where you believe you understand the direction of a relationship between two variables, but when you consider an additional variable, that direction appears to reverse. Suppose we wish to explore the relationship between two different splash pages (recall our previous A/B testing in Chapter 7, Basic Statistics). We will call these pages page A and page B once again. We have two splash pages that we wish to compare and contrast and our main metric for choosing will be in our conversion rates, just as earlier. Suppose we run a preliminary test and find the following conversion results: This means

Page A	**Page B**
75% (263/350)	83% (248/300)

that page B has almost a 10% higher conversion rate than page A. So, right off the bat, it seems like page B is the better choice because it has a higher rate of conversion. If we were going to communicate this data to our colleagues, it would seem that we are in the clear!

However, let's see what happens when we also take into account the coast that the user was closer to, as shown: Thus the paradox! When we break the sample down by location, it seems that Page A was better in both categories but was worse overall. That's the beauty and, also, the horrifying nature of the paradox. This happens because of the unbalanced classes between the four groups.

The Page A / East Coast group as well as the Page B / West Coast group are providing

	Page A	**Page B**
West Coast	**95% (76 / 80)**	93% (231/250)
East Coast	**72% (193/270)**	34% (17/50)
Both	75% (263/350)	**83% (248/300)**

most of the people in the sample, therefore skewing the results to be non expected. The confounding variable here might be the fact that the pages were given at different hours of the day and the west coast people were more likely to see page B, while the East coast people were more likely to see page A.

There is a resolution to Simpson's paradox (and therefore an answer), however, the proof lies in a complex system of Bayesian networks and is a bit out of the scope of this book. The main takeaway from Simpson's paradox is that we should not unduly give causational power to correlated variables. There might be confounding variables that have to be examined. Therefore, if you are able to reveal a correlation between two variables (such as website category and conversation rate or TV consumption and work performance), then you should absolutely try to isolate as many variables as possible that might be the reason for the correlation or can at least help explain your case further.

5.4.1 If correlation doesn't imply causation, then what does?

As a data scientist, it is often quite frustrating to work with correlations and not be able to draw conclusive causality. The best way to confidently obtain causality is, usually, through randomized experiments in Advanced Statistics. One would have to split up the population groups into randomly sampled groups and run hypothesis tests to conclude, with a degree of certainty, that there is a true causation between variables.

5.5 Verbal communication

Apart from visual demonstrations of data, verbal communication is just as important when presenting results. If you are not merely uploading results or publishing, you are usually presenting data to a room of data scientists, executives, or to a conference hall.

In any case, there are key areas to focus on when giving a verbal presentation, especially when the presentation is regarding findings about data.
There are generally two styles of oral presentations: one meant for more professional settings, including corporate offices where the problem at hand is usually tied directly to company performance or some other KPI (key performance indicator), and another meant more for a room of your peers where the key idea is to motivate the audience to care about your work.

5.5.1 It's about telling a story

Whether it is a formal or casual presentation, people like to hear stories. When you are presenting results, you are not just spitting out facts and metrics, you are attempting to frame the minds of your audience to believe in and care about what you have to say.
When giving a presentation, always be aware of your audience and try to gauge their reactions/interest in what you are saying. If they seem unengaged, try to relate the problem to them:
"Just think, when popular TV shows like Game of Thrones come back, your employees will all spend more time watching TV and therefore will have a lower work performance."
Now you have their attention. It's about relating to your audience, whether it's your boss or your mom's friend; you have to find a way to make it relevant.

5.6 On the more formal side of things

When presenting data findings to a more formal audience, I like to stick to the following six steps:
1. Outline the state of the problem.
In this step, we go over the current state of the problem, including what the problem is and how the problem came to the attention of the team of data scientists.
2. Define the nature of the data.
Here, we go more in depth about who this problem affects, how the solution would change the situation, and previous work done on the problem, if any.

3. Divulge an initial hypothesis.

Here, we state what we believed to be the solution before doing any work. This might seem like a more novice approach to presentations; however, this can be a good time to outline not just your initial hypothesis, but, perhaps, the hypothesis of the entire company. For example, "we took a poll and 61% of the company believes there is no correlation between hours of TV watched and work performance".

4. Describe the solution and, possibly, the tools that led to the solution.

Get into how you solved the problem, any statistical tests used, and any assumptions that were made during the course of the problem.

5. Share the impact that your solution will have on the problem.

Talk about whether your solution was different from the initial hypothesis. What will this mean for the future? How can we take action on this solution to improve ourselves and our company?

6. Future steps.

Share what future steps can be taken with the problem, such as how to implement the said solution and what further work this research sparked.

By following these steps, we can hit on all of the major areas of the data scientific method. The first thing you want to hit on during a formal presentation is action. You want your words and solutions to be actionable. There must be a clear path to take upon the completion of the project and the future steps should be defined.

5.7 The why/how/what strategy of presenting

When speaking on a less formal level, the why/how/what strategy is a quick and easy way to create a presentation worthy of praise. It is quite simple, as shown:

1. Tell your audience why this question is important without really getting into what you are actually doing.

2. Then, get into how you tackled this problem, using data mining, data cleaning, hypothesis testing, and so on.

3. Finally, tell them what your outcomes mean for the audience.

This model is borrowed from famous advertisements. The kind where they would not even tell you what the product was until 3 seconds left. They want to catch your attention and

then, finally, reveal what it was that was so exciting. Consider the following example: "Hello everyone, I am here to tell you about why we seem to have a hard time focusing on our job when the Olympics are being aired. After mining survey results and merging this data with company-standard work performance data, I was able to find a correlation between the number of hours of TV watched per day and average work performance. Knowing this, we can all be a bit more aware of our TV watching habits and make sure we don't let it affect our work. Thank you."

Bibliography

[1] Sinan Ozdemir (2016). Principles of Data Science: Learn the techniques and math you need to start making sense of your data. Packt publishing.

[2] Jianqing Fan, Runze Li, Cun-Hui Zhang, Hui Zou (2020). Statistical Foundations of Data Science. Chapman and Hall / CRC Press.

[3] Cathy O Neil, Rachel Schutt (2014). Doing Data Science: Straight talk from the frontline. O Reilly.

[4] James G, Witten D, Hastie T, Tibshirani R (2013). An Introduction to Statistical Learning with applications in R Springer.

[5] Hastie Trevor, Tibshirani Robert, Friedman Jerome (2009). The Elements of Statistical Learning Data Mining, Inference and Prediction. 2nd edition.

[6] A.L. BLUM (2020). Foundations Of Data Science. HINDUSTAN.

[7] Chandrika Jaini Vedam(2022). Data Science. BPB .

[8] Thomas Nield (2022). Essential Math for Data Science: Take Control of Your Data with Fundamental Linear Algebra, Probability, and Statistics (Grayscale Indian Edition). O'REILLY.

[9] Alex Campbell. (2021). Data Science for Beginners: Comprehensive Guide to Most Important Basics in Data Science. Alex Published.

[10] S. Tsumoto; T.Y. Lin; J.F. Peters (December, 2002). Foundations of data mining via granular and rough computing. IEEE.

[11] Thomas Thiele, Thorsten Sommer, Sebastian Stiehm, Sabina Jeschke, Anja Richert. (2016). Exploring Research Networks with Data Science A Data-Driven Microservice Architecture for Synergy Detection. DOI 10.1109/W-FiCloud.2016.58, IEEE COMPUTER SOCIETY.

[12] Walter V. Piegorsch. (2015). Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery. John Wiley & Sons.

[13] Loyiso G. Nongxa. (2020). Mathematical and statistical foundations and challenges of (big) data sciences. South African Journal of Science, Volume 113.

[14] Göran Kauermann, Helmut Küchenhoff, Christian Heumann. (2021). Statistical Foundations, Reasoning and Inference: For Science and Data Science. Springer Nature.

[15] William S. Cleveland. (2007). Data Science: an Action Plan for Expanding the Technical Areas of the Field of Statistics. ISI (INTERNATIOAL STATISTICAL REVIEW).

[16] D.G. Rees. (1987). Foundations of Statistics. Chapman and Hall/CRC .

[17] Jacco Thijssen (2014). A Concise Introduction to Statistical Inference. Chapman and Hall/CRC.

[18] Sudipto Banerjee, Anindya Roy (2014). Linear Algebra and Matrix Analysis for Statistics. Chapman and Hall/CRC, ISBN 9781138061163.

[19] Richard McElreath (2020). Statistical Rethinking A Bayesian Course with Examples in R and STAN. Chapman and Hall/CRC, ISBN 9780367139919.

[20] AjayKulkarni,DeriChong,Feras A.Batarseh (2020). Foundations of data imbalance and solutions for a data democracy. Academic Press.