ND111 - Data Science II - Notebook

Contents

Course Info

Tags

- Author : AH Uyekita
- Dedication : 10 hours/week (suggested)
- Start : 14/12/2018
- End (Planned): 28/12/2018
- Title : Data Science II - Foundations Nanodegree Program
  - COD : ND111

Related Courses

- ND110 - Data Science I - Nanodegree Foundations

Bookdown and Projects

- Project 01
- Project 02
- Project 03
- Project 04

Objectives

I want to finish this course in two weeks. It includes the Optional videos and chapters.

Syllabus

- Chapter 01 - Welcome
  - Lesson 01 - Instructions
  - Lesson 02 - Tips
- Chapter 02 - SQL for Data Analysis
  - Lesson 01 - Basic SQL
  - Lesson 02 - SQL Joins
  - Lesson 03 - SQL Aggregations
  - Lesson 04 - (Optional) SQL Subqueries & Temporary Tables (Advanced)
  - Lesson 05 - (Optional) SQL Data Cleaning (Advanced)
  - Project 01 - Query a Digital Music Store Database
- Chapter 03 - Data Wrangling

  - Lesson 01 - Introduction to Data Wrangling
  - Lesson 02 - Gathering
  - Lesson 03 - Assessing Data
  - Lesson 04 - Cleaning Data
  - Project 02 - Wrangle and Analyze Data

- Chapter 04 - Advanced Statistics

  - Lesson 01 - Descriptive Statistics - Part 1
  - Lesson 02 - Descriptive Statistics - Part 2
  - Lesson 03 - Admissions Case Study
  - Lesson 04 - Probability
  - Lesson 05 - Binomial Distribution
  - Lesson 06 - Conditional Probability
  - Lesson 07 - Bayes Rule
  - Lesson 08 - Python Probability Practice
  - Lesson 09 - Normal Distribution Theory
  - Lesson 10 - Sampling Distributions and the Central Limit Theorem
  - Lesson 11 - Confidence Intervals
  - Lesson 12 - Hypothesis Testing
  - Lesson 13 - Case Study: A/B Tests
  - Lesson 14 - Regression
  - Lesson 15 - Multiple Linear Regression
  - Lesson 16 - Logistic Regression
  - Project 03 - Analyze A/B Test Results

- Chapter 05 - Intro to Machine Learning

  - Lesson 01 - Welcome to Machine Learning
  - Lesson 02 - Naive Bayes
  - Lesson 03 - SVM
  - Lesson 04 - Decision Trees
  - Lesson 05 - Choose Your Own Algorithm
  - Lesson 06 - Datasets and Questions
  - Lesson 07 - Regressions
  - Lesson 08 - Outliers
  - Lesson 09 - Clustering
  - Lesson 10 - Feature Scaling
  - Lesson 11 - Text Learning
  - Lesson 12 - Feature Selection
  - Lesson 13 - PCA
  - Lesson 14 - Validation
  - Lesson 15 - Evaluation Metrics
  - Lesson 16 - Tying It All Together
  - Project 04 - Identify Fraud from Enron Email

- Chapter 06 - (Optional) Data Visualization

  - Lesson 01 - Introduction to Data Visualization
  - Lesson 02 - Design
  - Lesson 03 - Data Visualization in Tableau
  - Lesson 04 - Making Dashboard & Stories in Tableau

Repository Structure

This is the structure of this repository, each course's chapters (or parts) will be stored in different folders.

```
ND111_data_science_foundation_02
|
+--  01-Chapter_01
|           |
|           +--  README.md                        # General information
|
+--  02-Chapter_02
|           |
|           +--  README.md                        # General information
|           +--  00-Project_01                    # Project 01
|           +--  01-Lesson_01                     # Files from Lesson 01
|           |        +--  README.md               # Notes from Lesson 01 from Chapter 02
|           +--  02-Lesson_02                     # Files from Lesson 02
|           |        +--  README.md               # Notes from Lesson 02 from Chapter 02
|           .
|
+--  03-Chapter_03
|           |
|           +--  README.md                        # General information
|           +--  00-Project_02                    # Project 02
|           +--  01-Lesson_01                     # Files from Lesson 01
|           |        +--  README.md               # Notes from Lesson 01 from Chapter 02
|           +--  02-Lesson_02                     # Files from Lesson 02
|           |        +--  README.md               # Notes from Lesson 02 from Chapter 02
|           .
```

Best practice

- Add all *deliverables* in the `GitKraken Glo`;
- Take notes using the `Markdown`.

Chapter 1

Welcome

This chapter is about the General aspects of the Udacity platform study.

Instructions

General information about the course.

- Projects Deadline
- Projects Review
- Mentoring

Tips

- Asking Help
- Keep in contact with the Slack Community
- Student Manual

Chapter 2

SQL for Data Analysis

Tags

- Author : AH Uyekita
- Title : *SQL for Data Analysis*
- Date : 14/12/2018
- Course : Data Science II - Foundations Nanodegree
  - COD : ND111
  - **Instructor:** Derek Steer

Notebooks in ModeAnalytics

- Lesson 01
- Lesson 02
- Lesson 03 - Part 01
- Lesson 03 - Part 02
- Lesson 04
- Lesson 05

Software/Tool

Although this is not a requirement, I have created an account at Mode Analytics, it is the same tool used by the Derek (instructor) to teach SQL. Some good points in this tool:

- You can find the *Parch and Posey* database available in this tool;
- You can create very good documentation about your queries;
  - There is an option to create Jupyter Notebook or a complete Report;
- Far better tool from the Udacity embedded tool;
- It is free.

2.1 SQL Basics

2.1.1 Entity Relationship Diagrams (ERD)

This is a way to see (visualize) the relationship between different spreadsheets, in other words, how is structure a database. In a database, there are several tables, and each table has your own attributes, based on the cardinality they could interact with each other.

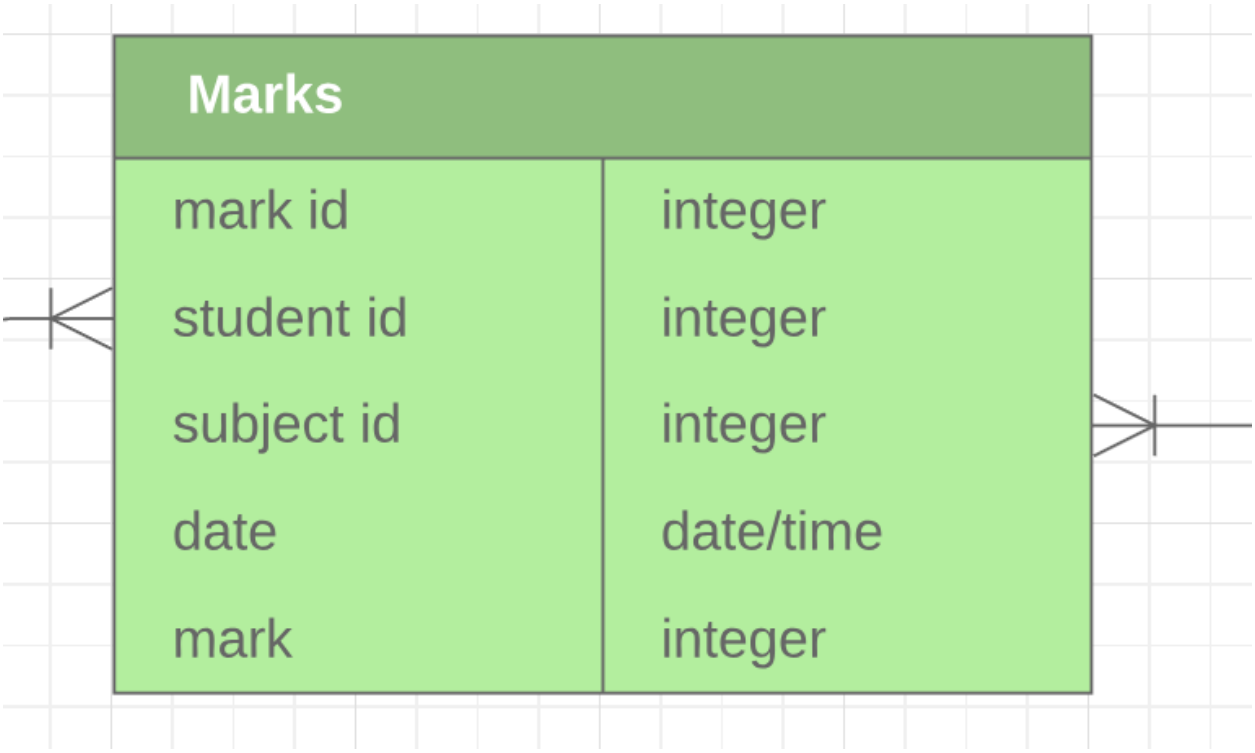

Figure 2.1: This is a entity.

2.1.1.1 Entities

This is a simple spreadsheet with information about anything you want, but keep in mind to: store new observations by rows and features/variables by column.

My example is a table called `Marks`, which has `mark id`, `student id`, `subject id`, `date` and `mark` as attributes. The other column is the variable's type.

2.1.1.2 Atributte

An attribute is a feature we want to keep track.

2.1.1.3 Relationship

Is a way to connect two tables.

Remember, this line has some properties, that is named as cardinality.

2.1.1.4 Cardinality

Cardinality represents a notation of how the information between tables will interact with each other.

Additional videos with good content.

Video 1 - Lucidchart Vídeo 2 - Lucidchart

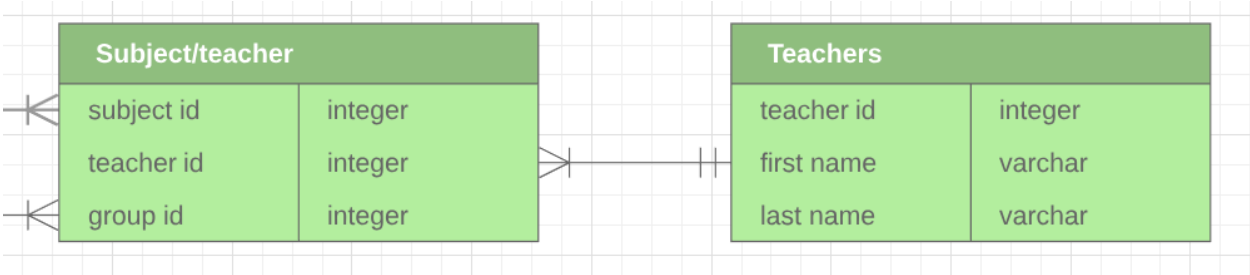

Figure 2.2: The line connecting two tables is a relationship.

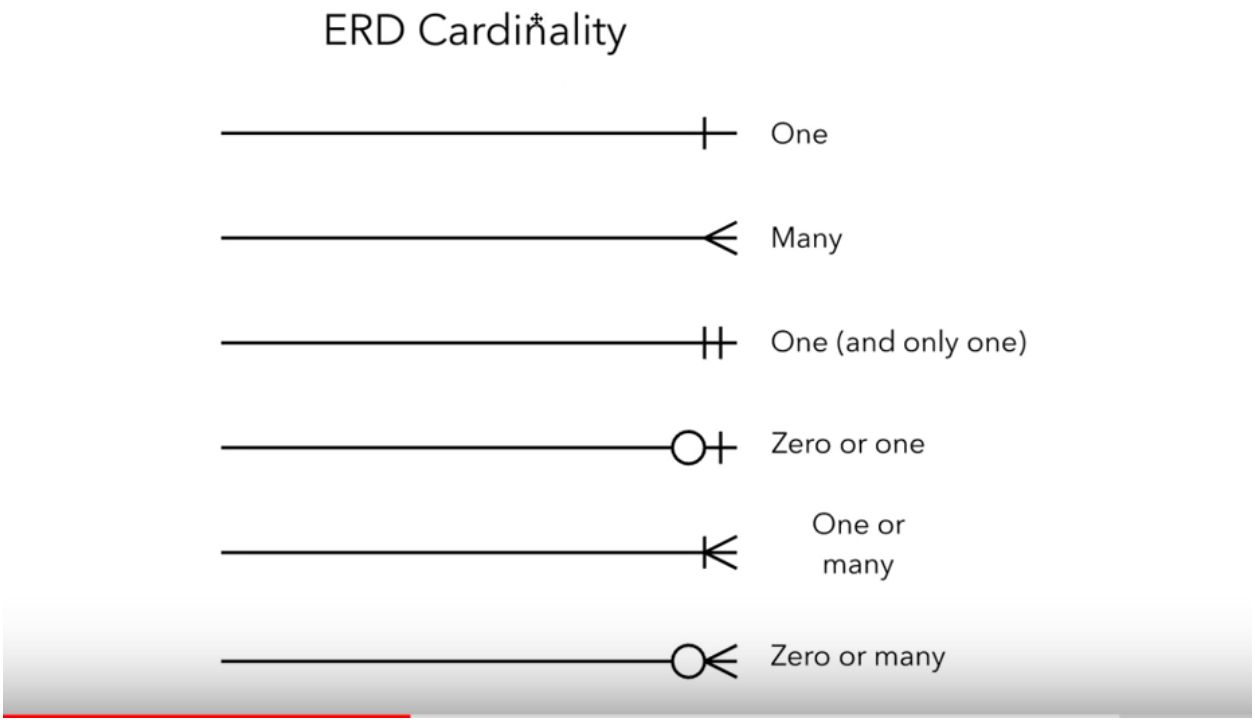

Figure 2.3: In a nutshell of Cardinality - Extracted from the Lucidchart Video.

2.1.2 SQL Introduction

SQL is a Language used to manage this interactions between tables, allowing us to access the stored database. The meaning of SQL is:

> Structured Query Language

It is very popular in Data Analysis because:

- Easy to understand
- Easy to learn
- Used to access very large datasets directly where is stored
- Easy to audit and replicate
- It is possible to run multiple queries at onde
- Almost do not have a limit of rows/observations
- Ensure the data Integrity, it is not possible to register a half child if you have defined this field as an integer
- SQL is very fast
- Database provide the data sharing, everybody could access the data simultaneously, which is good due to a standardization of database

SQL provides also functions such as:

- Summation
- Count
- Max and min
- Mean, etc.

Have in mind, probably we are going to manipulate data, and rarely updating or change values.

SQL is not case sensitive, so the best practices is to write the clauses/staments in upper case.

Best practices

```
SELECT first_column
  FROM my_table
```

Bad one

```
SelecT first_column
from my_table
```

Bear in mind, the indentation is not a requirements but helps a lot to understand your code.

2.1.2.1 SQL vs. NoSQL

Extracted from the class notes.

> You may have heard of NoSQL, which stands for not only SQL. Databases using NoSQL allow for you to write code that interacts with the data a bit differently than what we will do in this course. These NoSQL environments tend to be particularly popular for web based data, but less popular for data that lives in spreadsheets the way we have been analyzing data up to this point. One of the most popular NoSQL languages is called MongoDB. Udacity has a full course on MongoDB that you can take for free here, but these will not be a focus of this program. NoSQL is not a focus of analyzing data in this Nanodegree program, but you might see it referenced outside this course!

2.1.3 Clauses

Tell the database what to do.

2.1.3.1 DROP TABLE

Remove a table from the database.

2.1.3.2 CREATE TABLE

Create a new table.

2.1.3.3 SELECT

Is also know as query, is used to create a new table with the selected variables. You can use * if you want to select all columns.

```
SELECT first_column, second_column, last_column
  FROM first_table;
```

2.1.3.4 LIMIT

This is the same of `.head()` but this could only load a few lines to analyses the table.

```
SELECT first_column
  FROM my_table
LIMIT 1000            /* Will load the firs 1000 lines*/
```

2.1.3.5 ORDER BY

It is possible to order by in ascendant and descendent way.

ascendant

```
SELECT first_column, second_column, last_column
  FROM my_table
ORDER BY last_column /*ascendanting*/
LIMIT 1000
```

descendent

```
SELECT first_column, second_column, last_column
  FROM my_table
ORDER BY last_column DESC, second_column /*descending for last_column*/
LIMIT 1000
```

This last query will returns:

- Last_column ordered by the highest to lowest;
- The second_column will be the lowest to highest.

2.1.3.6 `WHERE`

Apply a filter to find a specific customer or anything else.

```
SELECT first_column, second_column, last_column
  FROM my_table
WHERE first_column = 100
ORDER BY second_column
LIMIT 100
```

All staments possible to use. * `>` (greater than) * `<` (less than) * `>=` (greater than or equal to) * `<=` (less than or equal to) * `=` (equal to) * `!=` (not equal to)

If the argument of the WHERE clause is not a number, you must use single quotes.

```
SELECT first_column, second_column, last_column
  FROM my_table
WHERE first_column = 'Hello World!'
ORDER BY second_column
LIMIT 100
```

2.1.4 Derived Columns

Is a new column created from the query. It is similar to the `mutate` function from R.

This is the operator to create a derived column:

- `*` (Multiplication)
- `+` (Addition)
- `-` (Subtraction)
- `/` (Division)

```
SELECT id, (standard_amt_usd/total_amt_usd)*100
FROM orders
LIMIT 10;
```

Will display without a specific name (?column?).

2.1.4.1 `AS`

If you use the `AS` the derived column will be name as you define (in other words "alias").

```
SELECT id, (standard_amt_usd/total_amt_usd)*100 AS std_percent, total_amt_usd
FROM orders
LIMIT 10;
```

Best pratices: No capital letters, descriptive names, etc.

2.1.5 Introduction to "Logical Operators"

In the next concepts, you will be learning about Logical Operators. Logical Operators include:

2.1.5.1 LIKE

Using with WHERE clause could search some patterns.

```
SELECT first_column, second_column, last_column
  FROM my_table
WHERE last_column LIKE '%ello%'
```

The % is called wild-card.

2.1.5.2 IN

It is the same in Python or R. `IN` will be used to filter the dataset based on a list.

```
SELECT first_column, second_column, last_column
  FROM my_table
WHERE last_column IN (100, 200)
```

This example will filter the rows of last_column with values of 100 or 200.

2.1.5.3 NOT

`NOT` return the reverse/opposite.

```
SELECT first_column, second_column, last_column
  FROM my_table
WHERE last_column NOT IN (100, 200)
```

This example will remove all observations equals to 100 or 200.

Possible uses:

- NOT IN
- NOT LIKE

2.1.5.4 AND

Logical statment usually to make some filtration.

```
SELECT *
FROM orders
WHERE standard_qty > 1000 AND poster_qty = 0 AND gloss_qty = 0;
```

2.1.5.5 BETWEEN

Sometimes AND statment could be replaced by BETWEEN, this is much clearly to understand. BUT the BETWEEN is inclusive, which means the endpoints will be included in the filter.

```
SELECT name
FROM accounts
WHERE name NOT LIKE 'C%' AND name LIKE '%s';
```

2.1.5.6 OR

Well, this is a logical operator.

```
SELECT id
FROM orders
WHERE gloss_qty > 4000 OR poster_qty > 4000;
```

2.2 SQL Joins

2.2.1 Joins

When a table is splited the performance to update or just to make a query is better than a big one. The reason is the quantity of data to read. This is one of the reason to split dataset in several tables, even more, sometimes in convinient to split because the type of data stored.

The reason of JOIN is to "bind" two datasets into one. Here we need to use the period `.` (table.colums) to reference which column/variable we want to select.

```
SELECT accounts.name, orders.occurred_at
  FROM orders
JOIN accounts
ON orders.account_id = accounts.id;
```

The result of this query is two columns (`name` and `occured_at`), and to linked by the `account_id` and `id`.

2.2.1.1 Primary Key (PK)

Is a columns with unique values used to map a variable.

2.2.1.2 Foreign Key (FK)

Is a Primary Key from the other table. We use the PK and FK to link the tables.

Based on the new information about `PK` and `FK`. Let's insert a picture to visualize the database.

I want to Join these tables. My query:

```
SELECT orders.*
FROM orders
JOIN accounts
ON orders.account_id = accounts.id;
```

What I need to realize:

- `PK` and `FK` **always** will be allocated in `ON`.
- `FROM` and `JOIN` each one with one table.

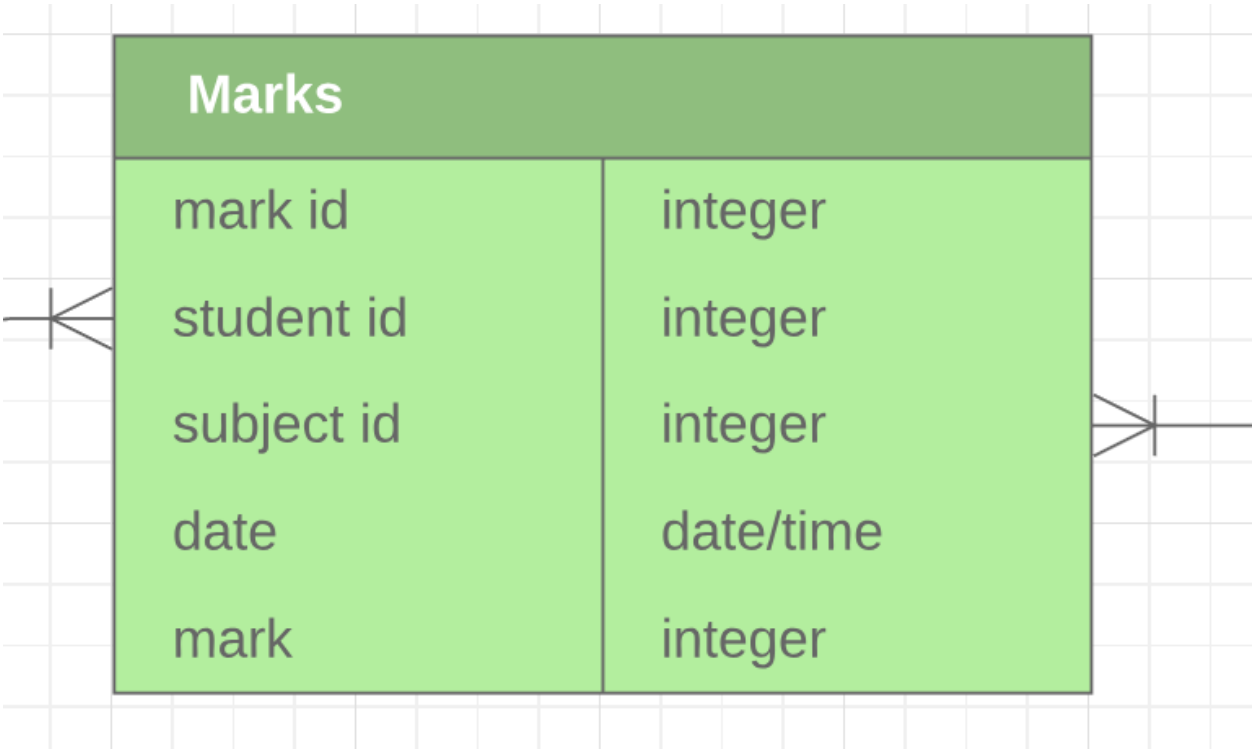

Figure 2.4: Example of Join

2.2.1.3 Binding three tables

It is possible to "chaining" three tables.

```
SELECT *
FROM web_events
JOIN accounts
ON web_events.account_id = accounts.id
JOIN orders
ON accounts.id = orders.account_id
```

In this case, I will import all columns, but I may want few columns.

```
SELECT web_events.channel, accounts.name, orders.total
FROM web_events
JOIN accounts
ON web_events.account_id = accounts.id
JOIN orders
ON accounts.id = orders.account_id
```

2.2.1.4 Alias

Alias is a form to "short" the name of columns, the first method is using `AS`, but it could be simplified by only a space.

- Example 1

```
Select t1.column1 aliasname, t2.column2 aliasname2
FROM tablename AS t1
JOIN tablename2 AS t2
```

or

```
Select t1.column1 aliasname, t2.column2 aliasname2
FROM tablename t1
JOIN tablename2 t2
```

- Example 2

```
SELECT col1 + col2 AS total, col3
```

or

```
SELECT col1 + col2 total, col3
```

or

2.2.1.5 INNER JOIN

Returns rows which appears in both tables.

```
SELECT table_1.id, table_1.name, table_2.total
  FROM table_2
    JOIN table_1
      ON table_2.account_id = table_1.id
```

These last examples are all `INNER JOINS`, and will return a new dataframe (intersection between two dataframes).

2.2.1.6 OUTER JOIN

There are two kinds of OUTER JOINs

- Left outer JOIN, and;
- Right outer JOIN.

This two new JOINs has a property to pull rows that only exist in one table, it means some rows might have NULL values. The standard for this course will be to use only the left outer join.

2.3 SQL Aggregations

2.3.1 Aggregations Functions

This is functions return a single row with the aggregated value.

- sum;
- min;
- max;
- mean, etc.

2.3.1.1 NULL

NULL is no a value, it is different from ZERO or a space, for this reason you can not use equal (=) to find it, for do so you must use `IS`. The NULL is ignored in all aggregatins functions, and it is defined as a property of the data.

For the *Parch and Posey* dataset, NULL is equal to zero.

```
WHERE something IS NULL
WHERE something IS NOT NULL
```

2.3.1.1.1 NULLs - Expert Tip

There are two common ways in which you are likely to encounter NULLs:

- NULLs frequently occur when performing a LEFT or RIGHT JOIN. You saw in the last lesson - when some rows in the left table of a left join are not matched with rows in the right table, those rows will contain some NULL values in the result set.
- NULLs can also occur from simply missing data in our database.

2.3.2 Functions

2.3.2.1 COUNT()

Count the number of rows. If the entire line has only NULLs, this line will be noted counted.

Simple Example:

```
SELECT COUNT(*)
FROM accounts;
```

Example with filter

```
SELECT COUNT(*) AS order_count
FROM some_table
WHERE any_column > 100 AND any_column < 200;
```

Example with column selection

```
SELECT COUNT(account.id)
FROM accounts;
```

2.3.2.2 SUM()

Perform the summation among rows. You must define which columns will be applied the sum function.

```
SELECT SUM(poster_qty)
FROM demo.orders;
```

2.3.2.3 MAX() and MIN()

Return a rows with the minimun or maximum of a given column.

```
SELECT MAX(poster_qty) AS max_poster_qty,
       MIN(standard_qty) AS min_standard_qty
FROM demo.orders;
```

2.3.2.4 GROUP BY

Divide the non-grouped column into groups, which means the aggregated function will be calculated by group.

- The GROUP BY always goes between WHERE and ORDER BY.

Example 1:

```
SELECT a.name, o.occurred_at
FROM accounts a
JOIN orders o
ON a.id = o.account_id
ORDER BY o.occurred_at
LIMIT 1;
```

Same example but indexing by number:

```
SELECT a.name, o.occurred_at
FROM accounts a
JOIN orders o
ON a.id = o.account_id
ORDER BY 2
LIMIT 1;
```

OBS.: The index used in ORDER BY clause is to refence o.occurred_at.

2.3.2.5 DISTINCT

DISTINCT is always used in SELECT statements, and it provides the unique rows for all columns written in the SELECT statement. Therefore, you only use DISTINCT once in any particular SELECT statement.

```
SELECT DISTINCT column1, column2, column3
FROM table1;
```

2.3.2.6 HAVING

> HAVING is the "clean" way to filter a query that has been aggregated, but this is also commonly done using a subquery. Essentially, any time you want to perform a WHERE on an element of your query that was created by an aggregate, you need to use HAVING instead.

Note extracted from the class notes.

```sql
SELECT s.id, s.name, COUNT(*) num_accounts
FROM accounts a
JOIN sales_reps s
ON s.id = a.sales_rep_id
GROUP BY s.id, s.name
HAVING COUNT(*) > 5
ORDER BY num_accounts;
```

2.3.3 DATE

To GROUP BY a date is quite complicated because each time is (obviously) different, for this, reason is necessary to "round" the time/date to group them.

2.3.3.1 DATE_TRUNC

Common trunctions are:

- day;
- month, and;
- year.

Sintaxe:

DATE_TRUNC('[interval]', time_column)

Where:

- microsecond
- millisecond
- second
- minute
- hour
- day
- week
- month
- quarter
- year
- century
- decade
- millenium

For further explanaition about date

```sql
SELECT demo.accounts.name,
       DATE_TRUNC('month', demo.orders.occurred_at) AS year_month,
       SUM(demo.orders.gloss_amt_usd) AS sum_gloss_usd
FROM demo.orders
JOIN demo.accounts
ON demo.orders.account_id = demo.accounts.id
WHERE demo.accounts.name = 'Walmart'
GROUP BY year_month, demo.accounts.name
ORDER BY sum_gloss_usd DESC
LIMIT 1;
```

2.3.3.2 DATE PART

Extract part of the date

2.3.4 CASE

Create a new column, derivate column, with a kind classification (assign a value into this new column according to the statment).

```
SELECT account_id,
       occurred_at,
       total,
       CASE WHEN total > 500 THEN 'Over 500'
            WHEN total > 300 AND total <= 500 THEN '301 - 500'
            WHEN total > 100 AND total <= 300 THEN '101 - 300'
            ELSE '100 or under' END AS total_group
FROM demo.orders
LIMIT 10;
```

Creates the total_group column.

2.3.4.1 With AGGREGATION

Combining the CASE clause with aggregations function could be a power tool, because the WHERE clause only evaluate one statement, using WHEN CASE it is possible to evaluate several staments.

```
SELECT demo.orders.account_id,
       demo.orders.total_amt_usd,
       CASE WHEN demo.orders.total_amt_usd >= 3000 THEN 'Large'
            ELSE 'Small' END AS level
FROM demo.orders
LIMIT 10;
```

2.4 SQL Subqueries & Temporary Tables (Advanced)

2.4.1 Subqueries

This is a way to nest queries, it means: The result of one query will be used as FROM to the next query.

```
SELECT *
FROM(SELECT something
     FROM   interesting) AS table_1
```

In the example above, I have one query nested to another. Bear in mind, I must give a alias to the nested query.

If the result of the subquery is a single value, you are allowed to insert this subquery wherever you want.

2.4.2 WITH

Also known as *Common Table Expression* (CTE), is a kinf of subquery but could be more helpful if someone is going to read the code. Due to the possibility to write the code in fragments an assign name, this is very handy.

Example

```
WITH my_with_example AS (SELECT ... MY CODE)

SELECT something
FROM my_with_example
```

As you can see it provide a better way to code because the code became more readable.

2.5 Data Cleaning (Advanced)

2.5.1 Data Cleaning

2.5.1.1 LEFT and RIGHT

It is the same of Excel functions.

```
SELECT LEFT(2, something) AS lefty_part_of_simething
FROM interesting
```

The example above will create a new column with the first two, from the left to right, character of something.

```
SELECT RIGHT(2, something) AS lefty_part_of_simething
FROM interesting
```

Almost the same, but start from the right to the left.

2.5.1.2 LEN

Returns the string length.

```
SELECT LEN(something)
FROM interesting
```

2.5.1.3 POSITION and STRPOS

POSITION will find a pattern in the string and will return the position (from the left to the right).

```
SELECT POSITION(',', something) /*Looking for a coma*/
FROM interesting
```

The STRPOS has the same use and same results.

```
SELECT STRPOS(something, ',') /*Looking for a coma*/
FROM interesting
```

Both functions are case sensitive.

2.5.1.4 LOWER and UPPER

Converts string into all lower or all upper cases.

```
SELECT LOWER(something)
FROM interesting
```

2.5.1.5 CONCAT

Bind/Combine/Concatenate strings (in different) columns into a new column.

Example 1

```
SELECT CONCAT(first_name, ' ',last_name) AS complete_name /* The ' ' is the space between strings*/
FROM interesting
```

You can use ||.

Example 2

```
SELECT first_name || ' ' || last_name AS complete_name /* The ' ' is the space between strings*/
FROM interesting
```

2.5.1.6 CAST

CAST allow to convert one type to another.

Example 1

```
SELECT CAST(year || month || day AS date) AS formatted_date
FROM interesting
```

The same of Example 1, but with a different notation to CAST clause.

Example 2:

```
SELECT (year || month || day AS date)::date AS formatted_date
FROM interesting
```

CAST is useful to converter strings into numbers or dates.

2.5.1.7 COALESCE

Converts NULL fields into Zero.

2.6 Project 01 - Chinook

Questions

All exercises of this chapter I have stored in the Mode Analytics platform.

Optional Questions

Project Submitted

I have written all the project in Mode Analytics because is a better place to coding.

- I can perform SQL queries;
- I can create graphics;
- An opportunity to get knowledge in a new tool.

Project 01 in Mode Analytic

2.6.1 Project Submission

To submit your project, please do the following:

- Review your project against the project Rubric. Reviewers will use this to evaluate your work.
- Create your slides with whatever presentation software you'd like (e.g. Google Slides, PowerPoint, Keynote, etc.).

In order to review your presentation, you will need to save your slides as a PDF. You can do this from within Google Slides by selecting File > Download as > PDF Document.

Chapter 3

Data Wrangling

Tags

- Author : AH Uyekita
- Title : *Data Wrangling*
- Date : 18/12/2018
- Course : Data Science II - Foundations Nanodegree
- COD : ND111

Jupyter Notebooks

- Lesson 01
- Lesson 02
- Lesson 03

Outline

- [x] Lesson 01 - Introduction to Data Wrangling
- [x] Lesson 02 - Gathering
- [x] Lesson 03 - Assessing Data
- [x] Lesson 04 - Cleaning Data
- [x] Project 02 - Wrangle and Analyze Data

3.1 Introduction to Data Wrangling

There are roughly three steps in the Data Wrangling.

- Gathering;
- Assessing, and;
- Cleaning.

This is a iterate process between these three steps.

> Data wrangling is about gathering the right pieces of data, assessing your data's quality and structure, then modifying your data to make it clean. But the assessments you make and convert to cleaning operations won't make your analysis, viz, or model better, though. The goal is to just make them possible, i.e., functional.

> EDA is about exploring your data to later augment it to maximize the potential of our analyses, visualizations, and models. When exploring, simple visualizations are often used to summarize your data's main characteristics. From there you can do things like remove outliers and create new and more descriptive features from existing data, also known as feature engineering. Or detect and remove outliers so your model's fit is better.
>
> ETL: You also may have heard of the extract-transform-load process also known as ETL. ETL differs from data wrangling in three main ways: * The users are different * The data is different * The use cases are different This article (Data Wrangling Versus ETL: What's the Difference?) by Wei Zhang explains these three differences well.

All text extracted from the class notes.

3.1.1 Gathering

Gathering is the first step of a Data Wrangling, is also known as Collecting or Acquiring. The Armenian Online Job Post has 19,000 jobs postings from 2004 to 2015.

> Best Practice: Downloading Files Programmatically

This is the reasons:

- Scalability: This automation will save time, and prevents erros;
- Reproducibility: Key point to any research. Anyone could reproduce your work and check it.

3.1.2 Assessing

The assessing in divided into two mains aspects:

- Quality of the dataset
- Tidiness of the dataset

3.1.2.1 Quality

Low quality dataset is related to a dirty dataset, which means the content quality of data.

Commom issues:

- Missing values
- Non standard units (km, meters, inches, etc. all mixed)
- Innacurate data, invalid data, inconsistent data, etc.

> One dataset may be high enough quality for one application but not for another.

3.1.2.2 Tidiness

Untidy data or *messy* data, is about the structure of the dataset.

- Each obsevation by rows, and;
- Each variable/features by column.

This is the Hadley Wickham definition of tidy data.

3.1.3 Assessing the data

There are two ways to assess the data.

- Visual, and;
- Programmatic.

3.1.3.1 Visual Assessment

Using regular tools, such as Graphics, Excel, tables, etc. It means, there is a human assessing the data.

3.1.3.2 Programmatic Assessment

Using automation to dataset evaluation is scalable, and allows you to handle a very huge quantity of data.

Examples of "Programmatic Assessment": Analysing the data using `.info()`, `.head()`, `.describe()`, plotting graphics (`.plot()`), etc..

Bear in mind, in this step we do not use "verbs" to describe any erros/problem, because the "verbs" will be actions to the next step.

3.1.4 Cleaning

Improving the quality of a dataset or cleaning the dataset do not means: Changing the data (because it could be **data fraud**).

The meaning of Cleaning is correcting the data or removing the data.

- Innacurate, wrong or irrelevant data.
- Replacing or filling (NULL or NA values) data.
- Combining/Merging datasets.

Improving the tidiness is transform the dataset to follow:

- each observation = row
- each variable = column

There are two ways to cleaning the data: manually and programmatic.

3.1.4.1 Manually

To be avoided.

3.1.4.2 Programmatic

There are three steps:

1. Define
2. Code
3. Test

Defining means defining a data cleaning plan in writing, where we turn our assessments into defined cleaning tasks. This plan will also serve as an instruction list so others (or us in the future) can look at our work and reproduce it.

Coding means translating these definitions to code and executing that code.

Testing means testing our dataset, often using code, to make sure our cleaning operations worked.

Text from the class notes.

3.2 Data Gathering

This is the first step of any Data Wrangling, sometimes this process is a bit complicated because you need to find these data (probably from different sources and then merge).

3.2.1 Flat File

This is the way to store data into a single text file, usually, this file has another extension (.csv, tsv, etc.), each one of this extension has your own characteristic.

- Each variable/features is separated by a comma and each row is an observation;
- Each variable/features is separated by a tab and each row is an observation.

There are some **advantages** for using the flat files.

- Anyone could read, even a human;
- Is lightweight;
- You do not need to install a specific software;
- Simple to understand (each variable is delimited by a coma/tab);
- Any software could open it;
- Very good to small dataset.

But has disadvantages also:

- Do not have standard;
- Do not have data integrity checks;
  - Duplicated rows;
  - You can record any value in any field;
- Not great to large datasets.

3.2.1.0.1 Importing the tsv file

I have used the `read_csv` to load the data, but I have set the sep argument as `\t`, which means tabular. Sometimes, the flat files use `;` or `,`, so it is necessary to define what is the delimiter.

Example:

```
import pandas as pd
df = pd.read_cvs('bestofrt.tsv', sep= '\t')
```

3.2.2 Web Scraping

This is terminology is used to say the data extracted from a website (usually using code to do it). Due to this code depends on the HTML file, if any change of the website happens, all the code used to web scrapping could stopping to working properly, which requires an adjustments. For this reason, web scraping is not a definitive solution.

3.2.3 HTTP Request

This is useful to access archives from the internet, combining with the OS package, it is possible to download and store locally the file.

3.2.4 Encoding and Character Set

This explanation is based on this Stack Overflow thread.

Encoding: Is a proccess to convert a something into bytes.

- Audio is encoded into MP3, WAV, etc.
- Images is encoded into PNG, JPG, TIFF, etc.
- Text files is enconded into ASCII, UTF-8, etc.

The Character Set is as the name, is a set of charaters which I can use to write a phrase, each character has a code which represents the letter/character. There are several character set such as ASCII and UTF-8.

3.2.5 Application Programming Interfaces - API

The API let you access the data from the internet in a resonable easy manner.

There are several API available in the internet for many social media:

- Facebook;
- Instagram;
- Twitter, etc.;

This lesson will use the Mediawiki, which is a Open Source API to Wikipedia.

Most of the file from the API are formated as JSON or XML.

3.2.6 JSON and XML

JSON stands for Javascript Object Notation and XML for Extensible Markup Language.

Sometimes the regular tabular way to structure the data is not a good solution, and for this reason, there are other forms to store data as JSON and XML.

They use a kind of “dictionary” to store data, which allows storing more than one information per variable.

There are some similarities in JSON and Python:

- JSON Array = Python list
- JSON Object = Python dictonary

3.2.7 Methods in this Lesson

3.2.7.1 `.find()`

This method is used to find tags and containers.

Example:

```
soup.find('title')
```

This code above will find the tag title, and return the content.

3.2.7.2 `.find_all()`

It is almost the same of `.find()`, but will find in all document the given pattern.

Example:

```
something.find_all('div')
```

This code will return all `div` in the document. It could be used with `limit = 1`, which will return the first `div`.

3.2.7.3 `.contents`

The `.contents` get the elements from the `find` and `find_all`.You are capable to select, which element you want (indexing).

```
something.find_all('div')[1].contents[2]
```

In this fragment of code, I am selecting only the third element of `something.find_all('div')[1]`.

3.2.7.4 `os.listdir()`

This function list all files inside a given folder/directory.

```
os.listdir(my_path)
```

3.2.7.5 `.glob()`

This method is a part of the glob package.

If you are familiar with Linux CLI, you have alread used the globbing to find a file in a folder.

```
import glob
glob.glob('any_folder/*.txt')
```

The result of the `.glob()` will be a list with all files which matches the `.txt`.

3.2.7.6 .read()

Convert the file into a in memory variable.

```
my_new_variable = file.read() # my_new_variable is a variable which contains the file.
```

3.2.7.7 .readline()

Read line by line every instance which is used this method.

```
file.readline() # Read the first line of the document file
file.readline() # Read the second line of the document file
file.read()     # Read the rest of the content.
```

3.2.7.8 .DataFrame()

This method from the pandas package converts a simple dictonary to a Pandas DataFrame.

```
pd.DataFrame(my_dataframe, columns = ['column_1', 'column_2', 'column_3'])
```

3.2.7.9 .page()

The .page() method from the wptools package converts a Wikipedia page into a object.

```
any_website = wptools.page('E.T._the_Extra-Terrestrial')
```

3.2.7.10 .get()

The .get() method from the wptools package extract all info from the wptools object.

```
any_website = wptools.page('E.T._the_Extra-Terrestrial').get()
```

3.3 Assessing Data

This is the second step of the Data Wrangling, and the aims of this lesson is to explain some details. There are two kind of *unclean* data:

- Quality issues: Dirty data;
  - Missing, duplicated, or incorrect data;
- Lack of tidiness: Also known as messy data.
  - Strucutural issues

There are two ways to assess:

- Visual: Plotting a simple graphic or visualizing the table (rows and columns);
- Programmatic: Using code to summarize the data frame using .info(), .describe(), average, summation, max, min, etc..

3.3.0.1 Dirty Data

Is related to the content issues, as known as low quality data.

- Innacurated data: Typos, corrupted, and duplicated data;

3.3.0.2 Messy Data

Messy data is related to structural issues, as known as untidy data.

- Each observation is a row;
- Each variable/features is a column;
- Based on the Hadley Wickham principles of tidy data.

3.3.1 Assess Proccess

In both cases (visual or programmatic), we could be divided into two main steps:

- Detect;
- Document.

3.3.2 Data Quality Dimensions

> Data quality dimensions help guide your thought process while assessing and also cleaning. The four main data quality dimensions are:

- Completeness: Missing values;
- Validity: Invalid value (like negative height or weight, zip code with only 4 digits, etc.);
- Accuracy: Wrong data which is valid (like the typo in the height);
- Consistency: Data without a standard notation (New York and NY, Colorado and CO, same information but differents notations).

The severity of this problem is decreasing order: Completeness, Validity, Accuracy, and Consistency.

3.4 Cleaning Data

Always opt to clean the data using the Programmatic way because manually it is more error prone.

This is the steps of Data Cleaning:

- Define: Defining a Data Cleaning Plan (usually writting down);
- Code: Converts the Data Cleaning Plan into code;
- Test: Evaluates the outuput of the code.

3.4.1 Tidiness

It is the standard preconized by Hadley Wickham.

Usually, the tidiness issues is the first to be solved.

3.4.2 Quality

After fixing tidiness issues, the quality issues could be fixed.

3.4.3 Methods

3.4.3.1 `.melt()`

Convert a wide format to a long format. It is the same of gather and spread functions from tidyr R package.

Good Video - Explaning the melt

3.5 Project 02 - Wrangle and Analyze Data

Project Submitted

Please, find below the URL to redirect to the project Jupyter Notebook.

Project 02 - WeRateDogs™ - Wrangle and Analyze Data

3.5.1 Project Submission

In this project, you'll gather, assess, and clean data then act on it through analysis, visualization and/or modeling.

Before you submit:

1. Ensure you meet specifications for all items in the Project Rubric. Your project "meets specifications" only if it meets specifications for all of the criteria.

2. Ensure you have not included your API keys, secrets, and tokens in your project files.

3. If you completed your project in the Project Workspace, ensure the following files are present in your workspace, then click "Submit Project" in the bottom righthand corner of the Project Workspace page:

- `wrangle_act.ipynb`: code for gathering, assessing, cleaning, analyzing, and visualizing data
- `wrangle_report.pdf` or wrangle_report.html: documentation for data wrangling steps: gather, assess, and clean
- `act_report.pdf` or `act_report.html`: documentation of analysis and insights into final data
- `twitter_archive_enhanced.csv`: file as given
- `image_predictions.tsv`: file downloaded programmatically
- `tweet_json.txt`: file constructed via API
- `twitter_archive_master.csv`: combined and cleaned data
- any additional files (e.g. files for additional pieces of gathered data or a database file for your stored clean data)

4. If you completed your project outside of the Udacity Classroom, package the above listed files into a zip archive or push them from a GitHub repo, then click the "Submit Project" button on this page.

As stated in point 4 above, you can submit your files as a zip archive or you can link to a GitHub repository containing your project files. If you go with GitHub, note that your submission will be a snapshot of the linked repository at time of submission. It is recommended that you keep each project in a separate repository to avoid any potential confusion: if a reviewer gets multiple folders representing multiple projects, there might be confusion regarding what project is to be evaluated.

It can take us up to a week to grade the project, but in most cases it is much faster. You will get an email once your submission has been reviewed. If you are having any problems submitting your project or wish to check on the status of your submission, please email us at review-support@udacity.com. In the meantime, you should feel free to proceed with your learning journey by continuing on to the next module in the program.

Chapter 4

Advanced Statistics

Tags

- Author : AH Uyekita
- Title : *Advanced Statistics*
- Date : 02/01/2019
- Course : Data Science II - Foundations Nanodegree
- COD : ND111

Related Courses

- UD170 - Introduction to Inferential Statistics
- UD170 - Introduction to Descriptive Statistics

Outline

- [x] Lesson 01 - Descriptive Statistics - Part 1
- [x] Lesson 02 - Descriptive Statistics - Part 2
- [x] Lesson 03 - Admissions Case Study
- [x] Lesson 04 - Probability
- [x] Lesson 05 - Binomial Distribution
- [x] Lesson 06 - Conditional Probability
- [x] Lesson 07 - Bayes Rule
- [x] Lesson 08 - Python Probability Practice
- [x] Lesson 09 - Normal Distribution Theory
- [x] Lesson 10 - Sampling Distributions and the Central Limit Theorem
- [x] Lesson 11 - Confidence Intervals
- [x] Lesson 12 - Hypothesis Testing
- [x] Lesson 13 - Case Study: A/B Tests
- [x] Lesson 14 - Regression
- [x] Lesson 15 - Multiple Linear Regression
- [x] Lesson 16 - Logistic Regression
- [x] Project 03 - Analyze A/B Test Results

4.1 Descriptive Statistics `Lesson 01`

This lesson is a kind of review.

4.1.1 What is data?

Data could be whatever thing: Text, Spreadsheets, video, images, database, etc.

4.1.2 Data types

- Quantitative Data: Allow us to perform mathematical operations with data (1, 2, 3, 4, etc.);
  - Continuous: Could be any real number (Age);
  - Discrete: Only integer number (Number of persons);
- Categorical Data: Used to label a group or a set of items (blue, yellow, red, etc.);
  - Ordinal: There are a way to put the categories in a scale (Very good, so-so, very poor);
  - Nominal: It is impossible to put the categories in order (blue, yellow, orange, etc.);

4.1.3 Measures of Center

4.1.3.1 Categorical

The categorical data is analyzed doing a simple summary to count the total of each category has.

4.1.3.2 Quantitative

Four main aspects when analysing **quantitative** data:

- Measures of `Center`
  - Mean
  - Median: The median splits our data so that 50% of our values are lower and 50% are higher.
    - Even number of elements: single values.
    - Odd number of elements: average between two "center" values.
  - Mode: The mode is the most frequently observed value in our dataset.
    - No mode: If all observations in our dataset are observed with the same frequency, there is no mode.
    - Many modes: If two (or more) numbers share the maximum value, then there is more than one mode.
- Measures of `Spread`
- The `Shape` of the Data
- `Outliers`

4.2 Quantitative Data `Lesson 02`

4.2.1 Measures of Spread

> How far are points from one another.

Common values of spread:

- Range;
- Interquartile range (IQR);
- Standard Deviation, and;
- Variance.

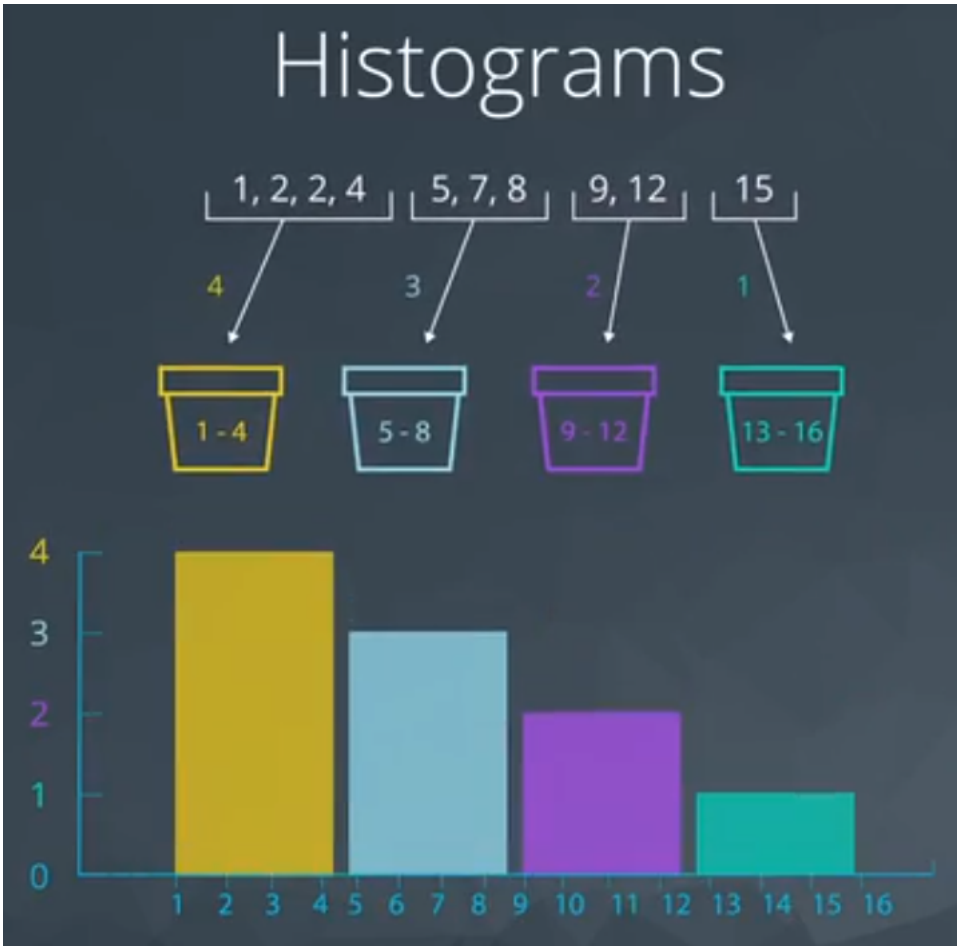

Figure 4.1:

4.2.1.1 Histogram

Figure 1 shows an example of histogram.

This is a way to visualize the quantitative data.

4.2.2 Five Number Summary

These are the number:

- Maximum;
- Third quartile or Q3 (75%);
- Second quartile (it is the same of mean) or Q2 (or 50%);
- First quartile or Q1 (25%), and;
- Minimum.

First step to do is order the values, as you can see in Figure 2 (odd set of values).

As you can see, Q1 and Q3 are the median of the data on either sides of Q2.

The range is defined as:

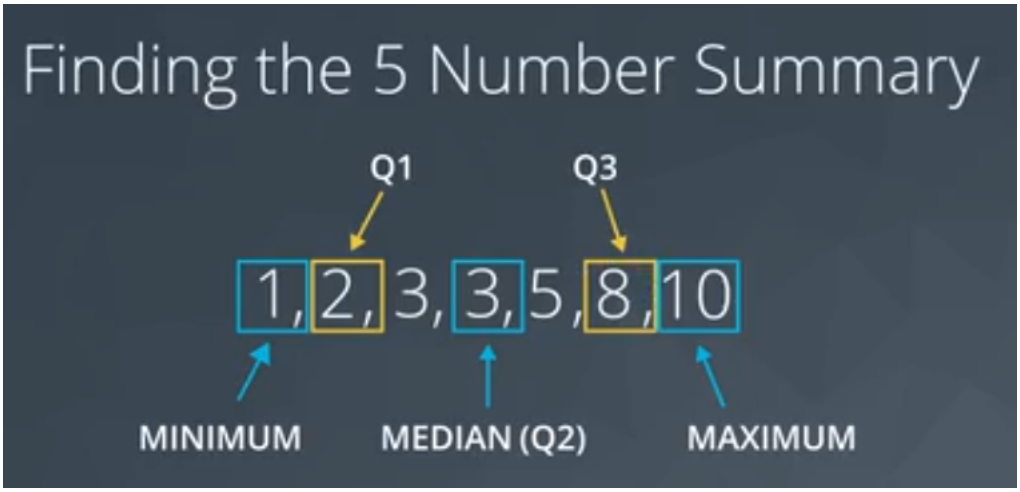

Figure 4.2:

$$\text{Range} = maximum - minimum \tag{1}$$

The Interquartile is define as:

$$\text{Interquartile} = Q3 - Q1 \tag{2}$$

For a even set of values I need to calculate the "average" of two values.

4.2.2.1 Boxplot

The boxplot graphic is a way to visualize the spread of the data.

It could be useful for quickly comparing the spread of two data sets.

Based on the Figure 4, the graphic on the right shows that in the weekends the number of dogs varies much more than on weekdays (looking to the range).

4.2.3 Standard Deviation

Meaning: On average, how much each point varies from the mean of the points.

First, I need to define the "distance" between mean and each observation. "Distance" could be interpreted as the difference of these two values. The issue observed in this difference are positive and negative values. For this reason, the square is used to turn everything positive (because later I can square root).

- Standard Deviation is frequently used to compare spread of different groups.
- Having higher standard deviation is associated with having higher risk.

$$\text{Standard Deviation} = \frac{1}{n}\sum_{i=1}^{n}(\bar{x} - x_i)^2 \tag{3}$$

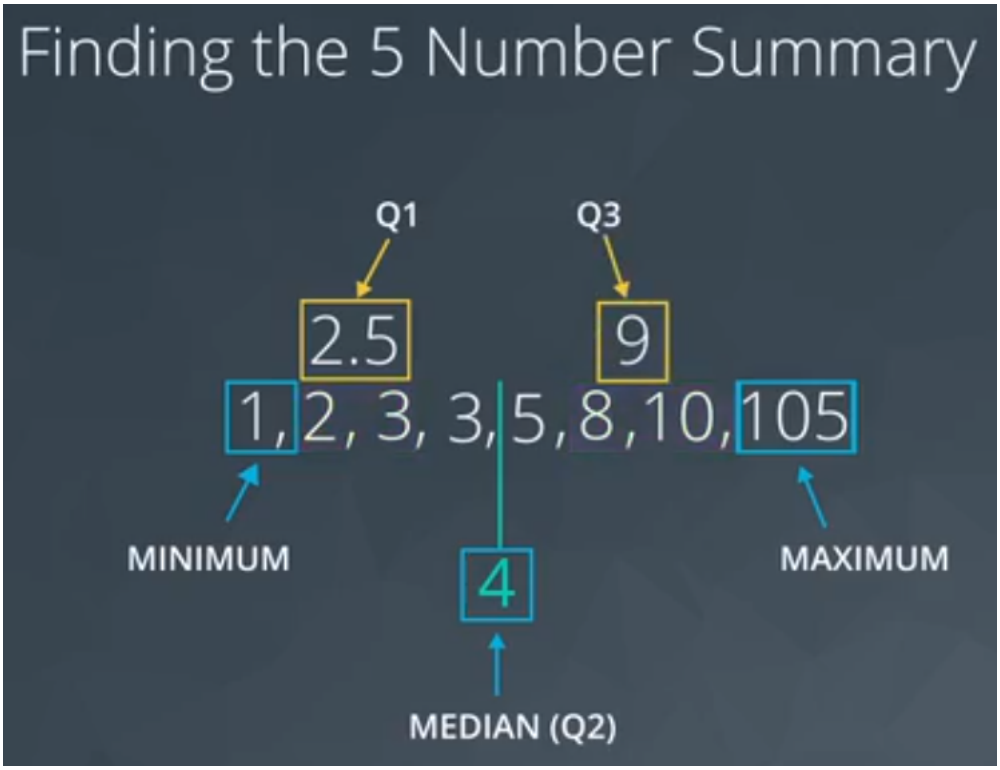

Figure 4.3:

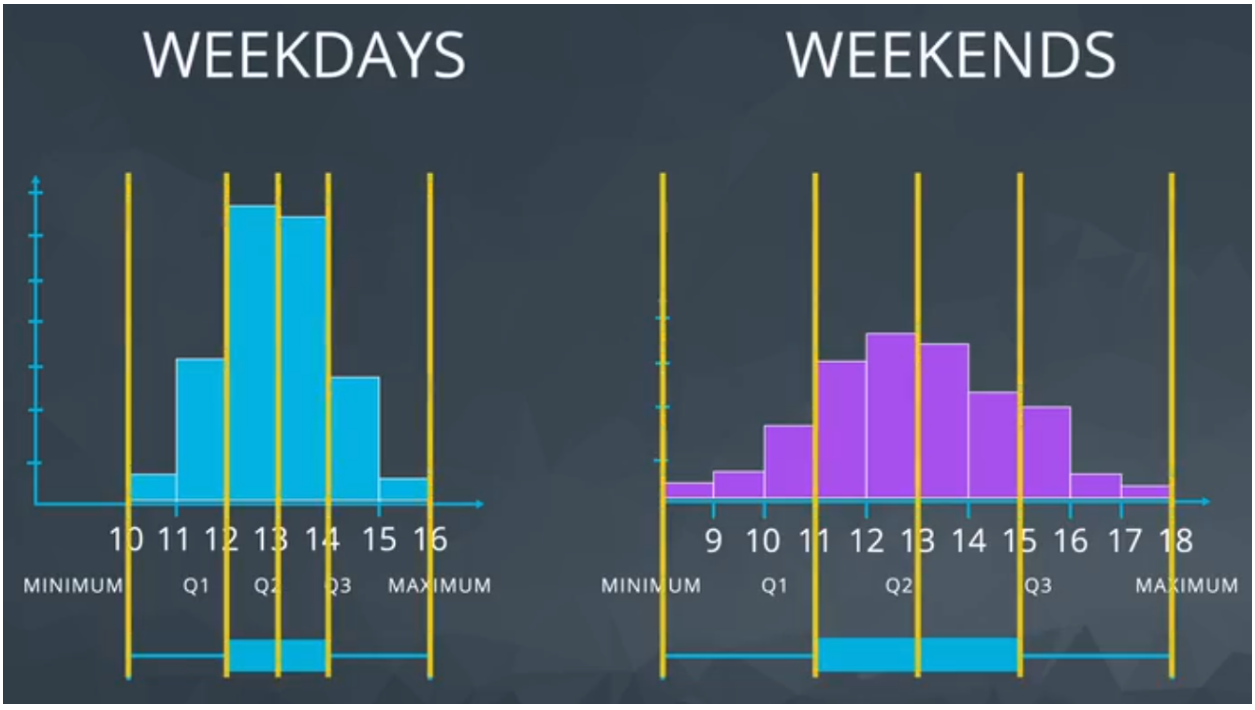

Figure 4.4:

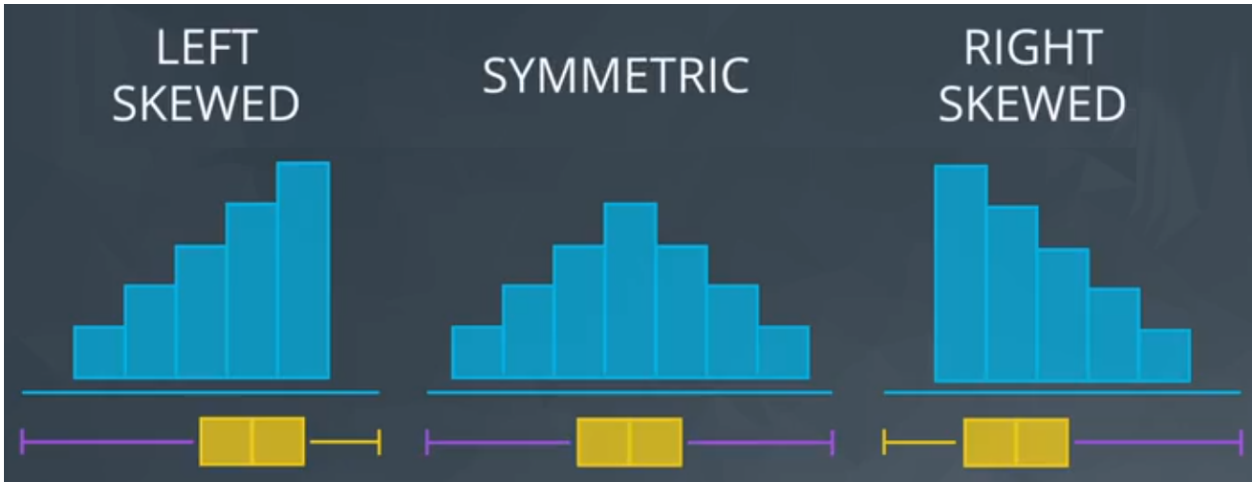

Figure 4.5:

4.2.4 Shape

The shape is related to the histogram form, Figure 5 shows an example.

- Left Skewed
  - is pulled to the "begining"
  - median stays close to the mode
  - GPA, Age of death, Asset price changes
- Simmetric (example: Normal distribution or bell curve)
  - mean = median = mode;
  - Examples: heights, weights, scores, precipitation, etc.
- Right Skewed
  - mean is pulled to the tail
  - median stays close to the mode
  - Amount of drug left in your bloodstream over time, distribution of wealth, human athletic abilities.

Side note: If you aren't sure if your data are normally distributed, there are plots called normal quantile plots and statistical methods like the Kolmogorov-Smirnov test that are aimed to help you understand whether or not your data are normally distributed. Implementing this test is beyond the scope of this class, but can be used as a fun fact.

4.2.5 Outliers

Data points thah fall very far from the rest of the values in our dataset.

The "very far" is quite generic and could be interpreted in many forms. One way to visualize it is plotting a histogram, as you can see in Figure 6.

1. Note they exist and the impact on summary Statistics
2. If typo, remove or fix it.
3. Understand why they exist, and the impact on questions we are trying to answer
4. Reporting the 5 number summary is better than mean and standard deviation when outliers are present
5. Be careful in reporting know how to ask the right questions

Figure 4.6:

4.2.6 Descriptive vs Inferential

Descriptive Statistics: Describing Collected Data Inferential Statistics: Drawing conclusions about a population based on data collected from sample of individuals from that population.

4.3 Admissions Case Study Lesson 03

4.3.1 Simpson's Paradox

In this example lesson, you learned about Simpson's Paradox, and you had the opportunity to apply it to a small example with Sebastian, as well as work through similar example in Python.

In the lessons ahead, you will be learning a lot by following along with Sebastian, but it is really important to put these ideas to practice using data and computing, because that is how you will apply these skills in a day to day environment as a Data Analyst or Data Scientist.

It is so easy to get caught up in looking at full aggregates of your data. Hopefully, the examples here serve as a reminder to look at your data multiple ways.

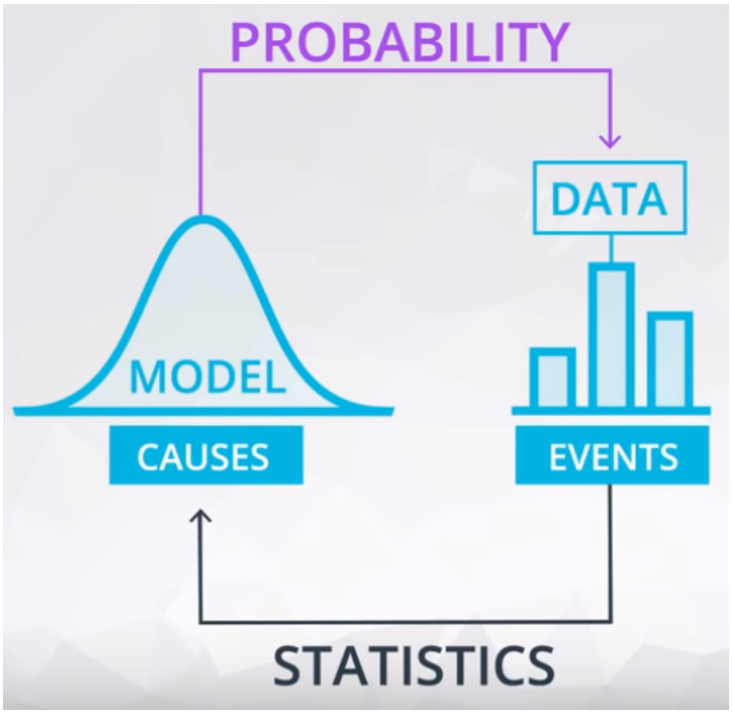

Figure 4.7:

4.3.2 Case Study

4.4 Probability `Lesson 04`

4.4.1 Introduction to Probability

Do not confound Statistics and Probability.

- Probability: Make preditcions about the future events based on models, and;
  - Here I want to predict data!
- Statistics: Analyze data from past events to infer what those models or causes could be.
  - Here I use data to preditc!

Figure 1 shows the relation between these two subjects.

4.4.1.1 Fair Coin

The probability notation is based on the 0 to 1 scale, where 0 means zero percentage and 1 means 100 percentage. The example below is a 50%.

$$P(HEADS) = 0.5$$

To be a fair coin the tail probability it is the same of heads.

$$P(TAILS) = 0.5$$

4.4.1.2 Loaded Coin

Its occurs when the $P(HEADS)$ is different of $P(TAILS)$. Bear in mind, in the equation 1.

$$P(HEADS) + P(TAILS) = 1 \tag{1}$$

Example 1: {HEADS, HEADS} = $P(H, H)$ for a fair coin.

$P(H) = P(T) = 0.5$

To ilustrate this solution, let's draw the Truth Table (Table 1)

Table 1 - Truth Table for a Fair Coin

Flip 1	Flip 2	Probability
H	H	$ 0.5 * 0.5 = 0.25 $
H	T	$ 0.5 * 0.5 = 0.25 $
T	H	$ 0.5 * 0.5 = 0.25 $
T	T	$ 0.5 * 0.5 = 0.25 $
		$ SUM = 1.0 $

The probability of P(H, H) is 0.25.

Example 2: {HEADS, HEADS} = $P(H, H)$ for a loaded coin.

$ P(H) = 0.6 $ $ P(T) = 0.4 $

To ilustrate this solution, let's draw the Truth Table (Table 2)

Table 2 - Truth Table for a Loaded Coin

Flip 1	Flip 2	Probability
H	H	$ 0.6 * 0.6 = 0.36 $
H	T	$ 0.6 * 0.4 = 0.24 $
T	H	$ 0.4 * 0.6 = 0.24 $
T	T	$ 0.4 * 0.4 = 0.16 $
		$ SUM = 1.0 $

The probability of P(H, H) is 0.36.

Example 3: Three coins flipped. What is the probability of only one heads in three coins flipped. Adopting a fair coin ($ P(H) = 0.5 $).

$P_1(OnlyoneH)$

Table 3 - Truth Table for a Loaded Coin

Flip 1	Flip 2	Flip 3	Probability	Has only one heads?	P_1
H	H	H	$ 0.5 * 0.5 * 0.5 = 0.125 $	No	0
H	H	T	$ 0.5 * 0.5 * 0.5 = 0.125 $	No	0
H	T	H	$ 0.5 * 0.5 * 0.5 = 0.125 $	No	0
H	T	T	$ 0.5 * 0.5 * 0.5 = 0.125 $	Yes	0.125
T	H	H	$ 0.5 * 0.5 * 0.5 = 0.125 $	No	0
T	H	T	$ 0.5 * 0.5 * 0.5 = 0.125 $	Yes	0.125
T	T	H	$ 0.5 * 0.5 * 0.5 = 0.125 $	Yes	0.125

Flip 1	Flip 2	Flip 3	Probability	Has only one heads?	P_1
T	T	T	$ 0.5 * 0.5 * 0.5 = 0.125 $	No	0
			$ SUM = 1.0 $	$ SUM = 3 cases $	$ SUM = 0.375 $

The $ P_1 $ is 0.375.

Example 4: Three coins flipped. What is the probability of only one heads in three coins flipped. Adopting a loaded coin ($ P(H) = 0.6 $).

$P_2(OnlyoneH)$

Table 3 - Truth Table for a Loaded Coin

Flip 1	Flip 2	Flip 3	Probability	Has only one heads?	P_2
H	H	H	$ 0.6 * 0.6 * 0.6 = 0.216 $	No	0
H	H	T	$ 0.6 * 0.6 * 0.4 = 0.144 $	No	0
H	T	H	$ 0.6 * 0.4 * 0.6 = 0.144 $	No	0
H	T	T	$ 0.6 * 0.4 * 0.4 = 0.096 $	Yes	0.096
T	H	H	$ 0.4 * 0.6 * 0.6 = 0.144 $	No	0
T	H	T	$ 0.4 * 0.6 * 0.4 = 0.096 $	Yes	0.096
T	T	H	$ 0.4 * 0.4 * 0.6 = 0.096 $	Yes	0.096
T	T	T	$ 0.4 * 0.4 * 0.4 = 0.064 $	No	0
			$ SUM = 1.0 $	$ SUM = 3 cases $	$ SUM = 0.288 $

The $ P_2 $ is 0.288.

4.4.2 Bernoulli Distribution

Founded on this introduction, let's generalize this concept using the Bernoulli Distribution.

> In probability theory and statistics, the Bernoulli distribution, named after Swiss mathematician Jacob Bernoulli, is the discrete probability distribution of a random variable which takes the value 1 with probability p and the value 0 with probability $q = 1 - p$, that is, the probability distribution of any single experiment that asks a yes–no question; the question results in a boolean-valued outcome, a single bit of information whose value is success/yes/true/one with probability p and failure/no/false/zero with probability q. It can be used to represent a (possibly biased) coin toss where 1 and 0 would represent "heads" and "tails" (or vice versa), respectively, and p would be the probability of the coin landing on heads or tails, respectively. In particular, unfair coins would have $p \neq 1/2$.
>
> The Bernoulli distribution is a special case of the binomial distribution where a single trial is conducted (so n would be 1 for such a binomial distribution). It is also a special case of the two-point distribution, for which the possible outcomes need not be 0 and 1. – Wikipedia

Rede more in wolfram.

4.4.2.1 Summary

> Here you learned some fundamental rules of probability. Using notation, we could say that the outcome of a coin flip could either be T or H for the event that the coin flips tails or heads, respectively.

Then the following rules are true:

- Probability of a Event >

$$\mathbf{P}(\mathbf{H}) = 0.5$$

- Probability of opposite event >

$$\mathbf{1} - \mathbf{P}(\mathbf{H}) = \mathbf{P}(\text{not H}) = 0.5$$

where not H is the event of anything other than heads. Since, there are only two possible outcomes, we have that $\mathbf{P}(\text{not H}) = \mathbf{P}(\mathbf{T}) = 0.5$. In later concepts, you will see this with the following notation: $\neg\mathbf{H}$.

- Probability of composite event

$$P * P * P * \cdots * P$$

It is only true because the events are independent of one another, which means the outcome of one does not affect the outcome of another.

- Across multiple coin flips, we have the probability of seeing n heads as $\mathbf{P}(\mathbf{H})^{\mathbf{n}}$. This is because these events are independent.

We can get two generic rules from this:

1. The probability of any event must be between 0 and 1, inclusive.

2. The probability of the compliment event is 1 minus the probability of an event. That is the probability of all other possible events is 1 minus the probability an event itself. Therefore, the sum of all possible events is equal to 1.

3. If our events are independent, then the probability of the string of possible events is the product of those events. That is the probability of one event AND the next AND the next event, is the product of those events.

4.4.2.2 Looking Ahead

You will be working with the Binomial Distribution, which creates a function for working with coin flip events like the first events in this lesson. These events are independent, and the above rules will hold. from Text: Recap + Next Steps

4.4.3 Conditional Probability

Here the first event will affect the second one. Figure 1 shows an example of it.

Figure 1 - Example of conditional probability.

The first event is to determine the bird type, and the second event the probability to run on the morning. Have in mind, these two birds has different probability to run on the morning.

- The early bird has 0.02;
- The night owl has 0.00.

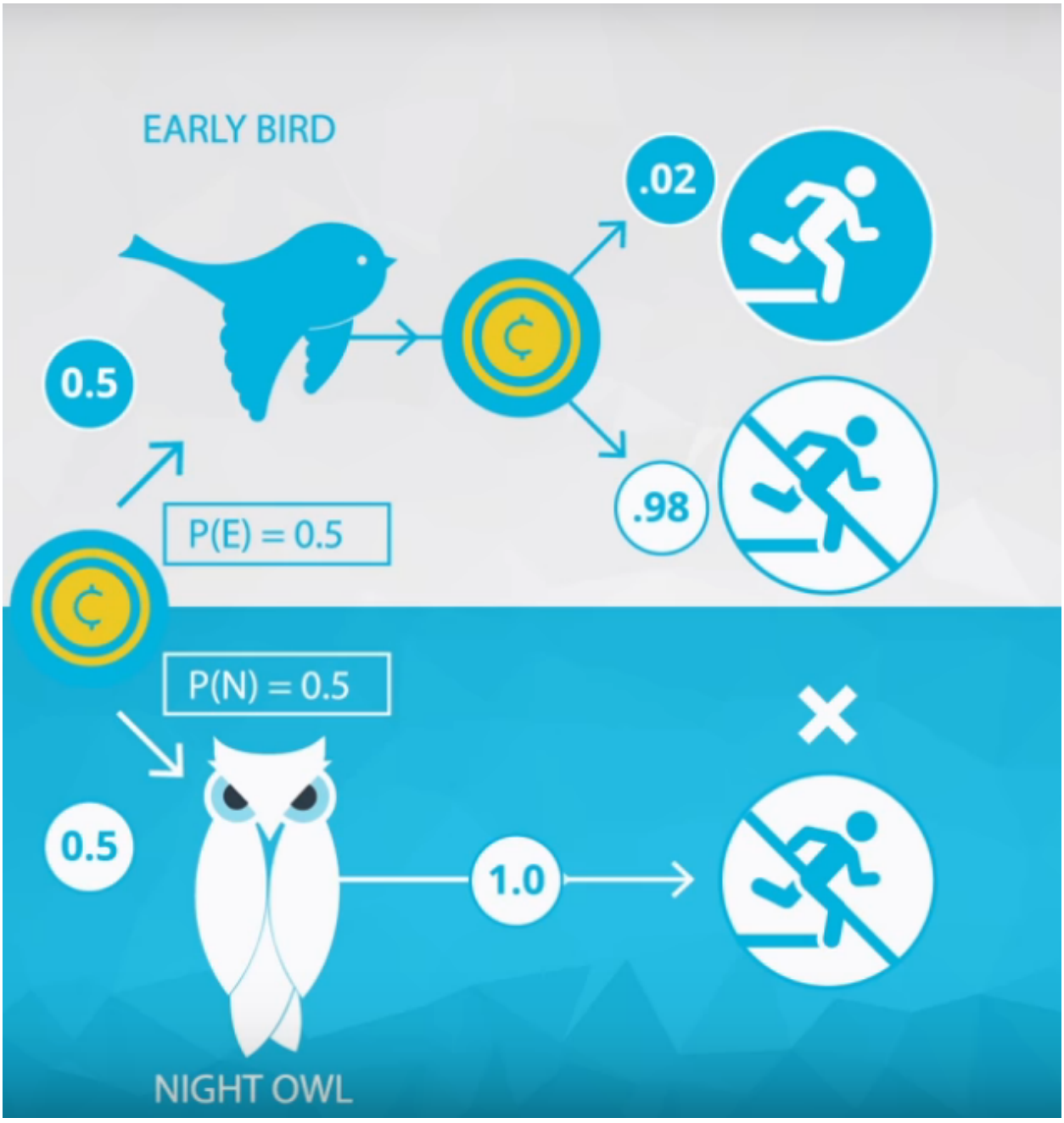

Figure 4.8: Figure 1

4.4.3.1 Medical Example

Supose a patient with a disease, the probability of this patient has cancer is 0.9 and to be free cancer is 0.1.

$$P(cancer) = 0.1 P(\neg\ cancer) = 0.9 \tag{1}$$

To be honest, we do not know if this patient has cancer, so it is necessary to apply a test. This test is not perfect, it means, there are a probability to indicates a false positive and a false negative.

For this reason, I introduce the conditional probability.

$$P(Positive|cancer) = 0.9 \tag{2}$$

What is the meaning of this notation?

Given the patient has cancer, the probability of this test indicates positive is 0.9. Thus, given the patient has cancer and the test indicates negative is 0.1, as shown in equation (3).

$$P(Negative|cancer) = 0.1 \tag{3}$$

Analogous to the case of the patient do not has cancer.

$$P(Positive|\neg\ cancer) = 0.2 P(Negative|\neg\ cancer) = 0.8 \tag{2}$$

Table 1 shows a representation in a tabular way.

Table 1 - Truth Table for Medical Example

Disease	Test	$P_{disease}$	P_{test}	P	Q1: Test Positive?	Q1: Answer
No	Negative	$P(\neg\ cancer) = 0.9$	$P(Negative\|\neg\ cancer) = 0.8$	$P(Negative\|\neg\ cancer) * P(\neg\ cancer) = 0.72$	No	0
No	Positive	$P(\neg\ cancer) = 0.9$	$P(Positive\|\neg\ cancer) = 0.2$	$P(Positive\|\neg\ cancer) * P(\neg\ cancer) = 0.18$	Yes	0.18
Yes	Negative	$P(cancer) = 0.1$	$P(Negative\|cancer) = 0.1$	$P(Negative\|cancer) * P(cancer) = 0.01$	No	0
Yes	Positive	$P(cancer) = 0.1$	$P(Positive\|cancer) = 0.9$	$P(Positive\|cancer) * P(cancer) = 0.09$	Yes	0.09
				$SUM = 1$		$SUM = 0.27$

What is the probability the test is positive?

Q1: 0.27

Coin flip example

Two coins, one fair and other loaded.

- Coin 1: $P_1(HEADS) = P_1(TAILS) = 0.5$;

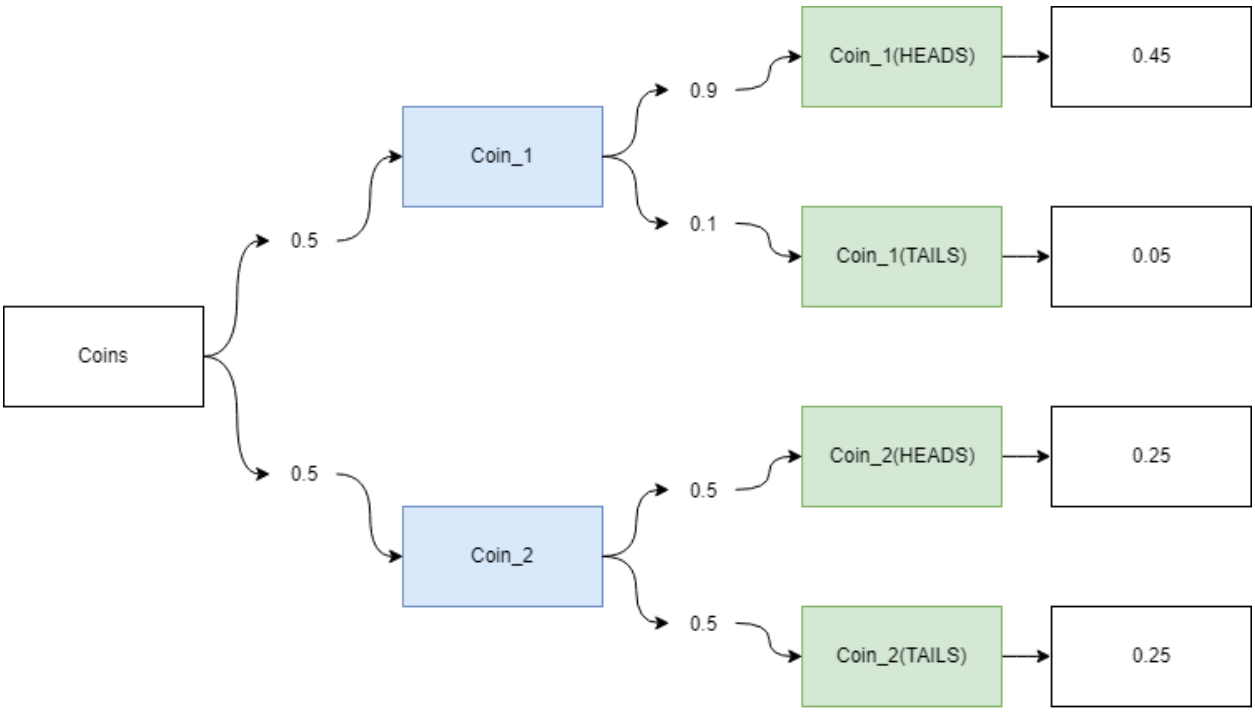

Figure 4.9: Figure 2

- Coin 2: $P_2(HEADS) = 0.9$ and $P_2(TAILS) = 0.1$.

Figure 2 - Coin Example of conditional probability.

What is the probability of this sequence HEADS and TAILS?

Coin	Flip 1	Flip 2	P_{coin}	P_{Flip1}	P_{Flip2}	P	Q2: HEADS then TAILS?	Q2: answer
1	H	H	0.5	0.9	0.9	0.405	No	0
1	H	T	0.5	0.9	0.1	0.045	Yes	0.045
1	T	H	0.5	0.1	0.9	0.045	No	0
1	T	T	0.5	0.1	0.1	0.005	No	0
2	H	H	0.5	0.5	0.5	0.125	No	0
2	H	T	0.5	0.5	0.5	0.125	Yes	0.125
2	T	H	0.5	0.5	0.5	0.125	No	0
2	T	T	0.5	0.5	0.5	0.125	No	0
						$SUM =$ 1		$SUM =$ 0.170

4.4.3.2 Summary

In this lesson you learned about conditional probability. Often events are not independent like with coin flips and dice rolling. Instead, the outcome of one event depends on an earlier event.

For example, the probability of obtaining a positive test result is dependent on whether or not you have a particular condition. If you have a condition, it is more likely that a test result is positive. We can formulate conditional probabilities for any two events in the following way:

- $P(A|B) = \frac{P(A \cap B)}{P(B)}$
- $P(AB)$

In this case, we could have this as:

Example 1: 5 coin flips. P(two HEADS).

- The order of the HEADS do not matter.

What's the number of combinations?

Flip 1	Flip 2	Flip 3	Flip 4	Flip 5
H	?	?	?	?
H	H	?	?	?
5	4	1	1	1

You can place H in 5 places, after place the first H you only have 4 places. For this reason, there are 20 posibilities.

$$P_1 = 5 * 4 = 20$$

Bear in mind, the H's are equal and you can swap each one.

H_1	H_2
?	?
2	1

So there are two possible H's to insert in the H_1

$$P_2 = 2 * 1 = 2$$

What this P_2 means?

> You have two equals instances so the P_1 has double entries. The good part is the P_2 is used to "fix" it.

$$P = \frac{P_1}{P_2} = \frac{20}{2} = 10$$

Example 2: 10 coin flips. P(four HEADS).

- The order of the HEADS do not matter.

What's the number of combinations?

Flip 1	Flip 2	Flip 3	Flip 4	Flip 5	Flip 6	Flip 7	Flip 8	Flip 9	Flip 10
H	?	?	?	?	?	?	?	?	?
H	H	H	T	H	T	T	T	T	T
10	9	8	1	7	1	1	1	1	1

You can place H in 10 places, after place the first H you only have 9 places. For this reason, there are

$(10 * 9 * 8 * 7)$ posibilities.

$$P_1 = 10 * 9 * 8 * 7 = 5,040$$

Bear in mind, the H's are equal and you can swap each one.

H_1	H_2	H_3	H_4
?	?	?	?
4	3	2	1

So there are two possible H's to insert in the H_1

$$P_2 = 4 * 3 * 2 * 1 = 24$$

What this P_2 means?

> You have two or more equals instances so the P_1 has "double" entries. The good part is the P_2 is used to "fix" it.

$$P = \frac{P_1}{P_2} = \frac{5,040}{24} = 210$$

4.5.1 Equation

Founded on the examples above, it is possible to write a equation, given 10 flips (k) and an expected 3 heads (n).

$$P = \frac{P_1}{P_2} = \frac{10 * 9 * 8}{\underbrace{3 * 2 * 1}_{3!}}$$

Let's multiply by $(7 * 6 * 5 * 4 * 3 * 2 * 1)$ or simply by 7!.

$$P = \frac{P_1}{P_2} = \frac{10 * 9 * 8 * 7!}{3! * 7!} = \frac{\overbrace{10!}^{k!}}{\underbrace{3!}_{n!} * \underbrace{7!}_{(k-n)!}} = \frac{k!}{n!(k-n)!} \tag{1}$$

Equation (1) is also noted as:

$$C_{n,k} = \binom{n}{k} \tag{2}$$

Equation (2) will only calculate the number of combinations. We can aggregate the probability.

- $P(H)$: for heads;
- $P(T)$: for tails;

Given the 10 coins flips, the probability for a single instance, no matter the order:

Obs.: Do not confound with permutation notation.

$$P_{k,n} = P(H)^n * P(T)^{k-n}$$
$$P_{10,3} = P(H)^3 * P(T)^7 \tag{3}$$

For a fair coin.

$$P_{10,3} = 0.5^3 * 0.5^7 = 0.000976563 \tag{4}$$

The value of $P(10, 3)$ is for a single time, I know there are many instances where could happen 3 heads, for this reason I use the combination.

$$C_{10,3} = \binom{10}{3} = \frac{10!}{7! * 3!} = 120$$

The probability to happen 3 heads in 10 flips coins is:

$$P(10|3) = C_{10,3} * P_{10,3} = 120 * 0.000976563 = 0.1171875$$

Expanding this concept to a all around equation:

$$P(k|n) = \underbrace{C_{k,n}}_{\text{note 1}} * \underbrace{P_{k,n}}_{\text{note 2}} \tag{5}$$

- note 1: Probability to occur the given *combination* (3 heads and 7 tails) over the all combinations ($10^2 = 1,024$);
- note 2: Probability based on the coins (heads and tails probabilities).

4.5.1.1 Additional Info

Do not confound Permutation with Combination.

- Combination: When the order does not matter;
- Permutation: When the order is important.

Read more in mathplanet.

4.6 Conditional Probability `Lesson 06`

Here the first event will affect the second one. Figure 1 shows an example of it.

Figure 1 - Example of conditional probability.

The first event is to determine the bird type, and the second event the probability to run on the morning. Have in mind, these two birds has different probability to run on the morning.

- The early bird has 0.02;
- The night owl has 0.00.

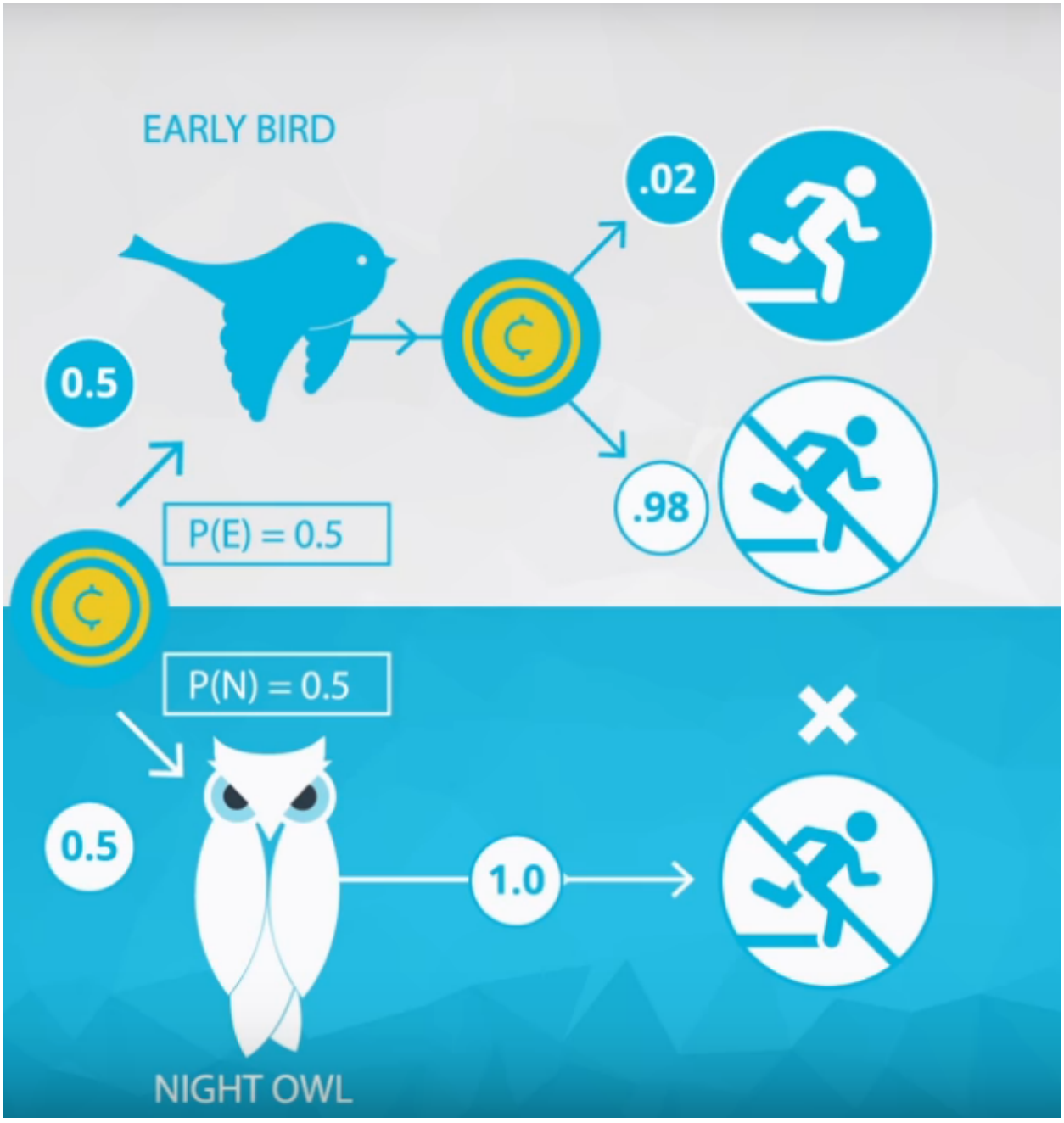

Figure 4.10: Figure 1

4.6.1 Medical Example

Supose a patient with a disease, the probability of this patient has cancer is 0.9 and to be free cancer is 0.1.

$$P(cancer) = 0.1 P(\neg\ cancer) = 0.9 \tag{1}$$

To be honest, we do not know if this patient has cancer, so it is necessary to apply a test. This test is not perfect, it means, there are a probability to indicates a false positive and a false negative.

For this reason, I introduce the conditional probability.

$$P(Positive|cancer) = 0.9 \tag{2}$$

What is the meaning of this notation?

Given the patient has cancer, the probability of this test indicates positive is 0.9. Thus, given the patient has cancer and the test indicates negative is 0.1, as shown in equation (3).

$$P(Negative|cancer) = 0.1 \tag{3}$$

Analogous to the case of the patient do not has cancer.

$$P(Positive|\neg\ cancer) = 0.2 P(Negative|\neg\ cancer) = 0.8 \tag{2}$$

Table 1 shows a representation in a tabular way.

Table 1 - Truth Table for Medical Example

Disease	Test	$P_{disease}$	P_{test}	P	Q1: Test Positive?	Q1: Answer
No	Negative	$P(\neg\ cancer) = 0.9$	$P(Negative\|\neg\ cancer) = 0.8$	0.72	No	0
No	Positive	$P(\neg\ cancer) = 0.9$	$P(Positive\|\neg\ cancer) = 0.2$	0.18	Yes	0.18
Yes	Negative	$P(cancer) = 0.1$	$P(Negative\|cancer) = 0.1$	0.01	No	0
Yes	Positive	$P(cancer) = 0.1$	$P(Positive\|cancer) = 0.9$	0.09	Yes	0.09
				$\sum = 1$		$\sum = 0.27$

What is the probability the test is positive?

Q1: 0.27

4.6.1.1 Coin flip example

Two coins, one fair and other loaded.

- Coin 1: $P_1(HEADS) = P_1(TAILS) = 0.5$;

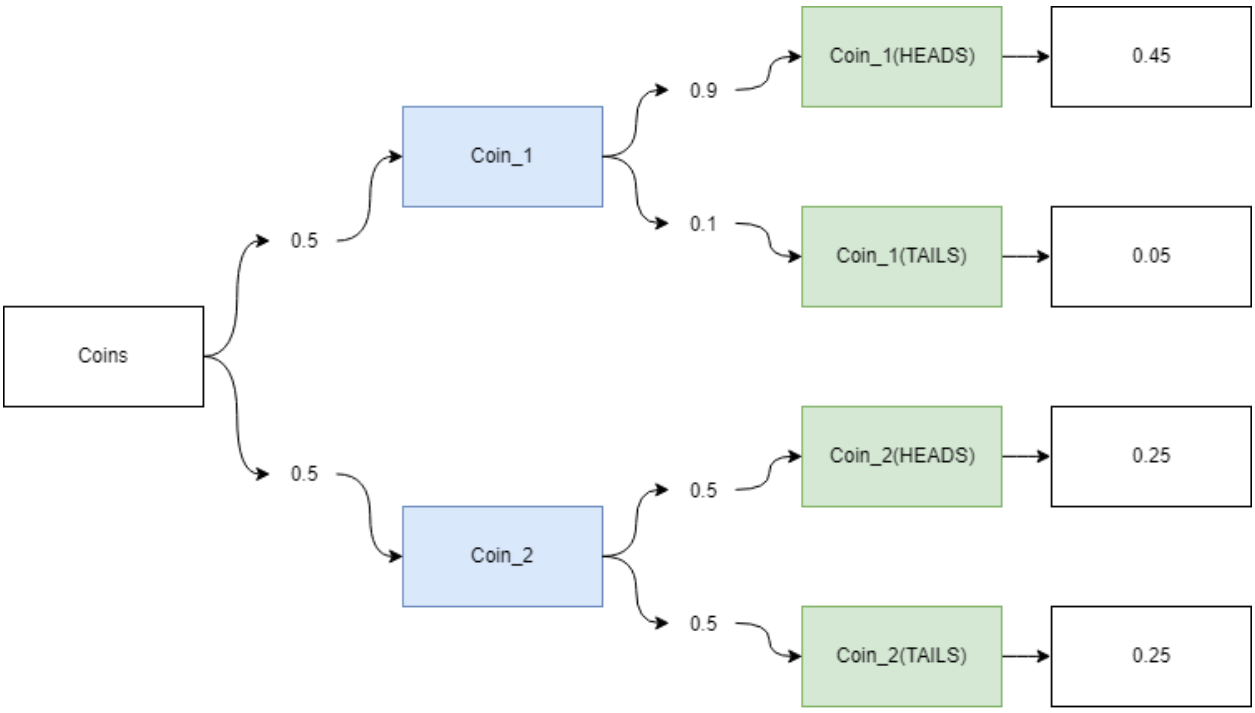

Figure 4.11: Figure 2

- Coin 2: $P_2(HEADS) = 0.9$ and $P_2(TAILS) = 0.1$.

Figure 2 - Coin Example of conditional probability.

What is the probability of this sequence HEADS and TAILS?

Coin	Flip 1	Flip 2	P_{coin}	P_{Flip1}	P_{Flip2}	P	Q2: HEADS then TAILS?	Q2: answer
1	H	H	0.5	0.9	0.9	0.405	No	0
1	H	T	0.5	0.9	0.1	0.045	Yes	0.045
1	T	H	0.5	0.1	0.9	0.045	No	0
1	T	T	0.5	0.1	0.1	0.005	No	0
2	H	H	0.5	0.5	0.5	0.125	No	0
2	H	T	0.5	0.5	0.5	0.125	Yes	0.125
2	T	H	0.5	0.5	0.5	0.125	No	0
2	T	T	0.5	0.5	0.5	0.125	No	0
						$\sum = 1$		$\sum = 0.170$

4.6.1.2 Summary

In this lesson you learned about conditional probability. Often events are not independent like with coin flips and dice rolling. Instead, the outcome of one event depends on an earlier event.

For example, the probability of obtaining a positive test result is dependent on whether or not you have a particular condition. If you have a condition, it is more likely that a test result is positive. We can formulate conditional probabilities for any two events in the following way:

- $P(A|B) = \frac{P(A \cap B)}{P(B)}$
- $P(AB)$

In this case, we could have this as:

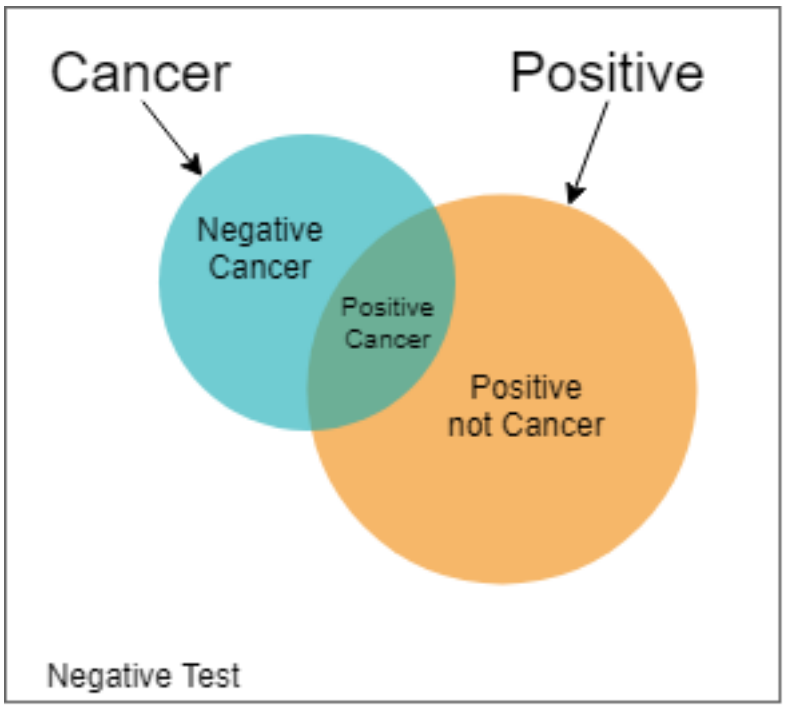

Figure 4.12:

4.7.1 Cancer Example

The probability to a person has cancer is 1% and the probability to the test gives positive is 90%. If a person do not has cancer the probability of the test gives negative is 90%.

What is the probability of a given positive test the person has cancer?

Disease	Test	$P_{disease}$	P_{test}	P	Q1: Test positive?	Q1: answer
No	Negative	0.99	0.90	0.891	No	0
No	Positive	0.99	0.10	0.099	Yes	0.099
Yes	Negative	0.01	0.10	0.001	No	0
Yes	Positive	0.01	0.90	0.009	Yes	0.009
						SUM = 0.108

Bear in mind, the probability of a false positive is 0.099, which is 11 times bigger than the a truth valeu of 0.009.

Figure 1 ilustrate this situation.

Given the test is positive, the probability of this person has cancer is:

$$P(C\|Positive) = \frac{P(C, Positive)}{P(C, Positive) + P(\neg\, C, Positive)} = \frac{0.009}{0.009 + 0.099} = 0.08333$$

4.7.2 Bayes Rule

From the example above, I can point out some definitions:

Prior:

This is a information before the test.

$$P(C) = 0.01 P(\neg C) = 0.99 \tag{1}$$

Joint:

Now, I will apply the test for a given positive result.

$$P(C, Positive) = P(C) * \underbrace{P(Positive \| C)}_{Sensibility} P(\neg C, Positive) = P(\neg\ C) * \underbrace{P(Positive \| \neg\ C)}_{Sensibility} \tag{2}$$

For a negative result.

$$P(C, Negative) = P(C) * \underbrace{P(Negative \| C)}_{Specitivity} P(\neg\ C, Negative) = P(\neg\ C) * \underbrace{P(Negative \| \neg\ C)}_{Specitivity} \tag{3}$$

Normalization:

This is performed for each result (Positive and Negative).

$$P(Positive) = P(C, Positive) + P(\neg\ C, Positive) \tag{4}$$

Posterior:

Divide the $P(C, Positive)$ and $P(\neg C, Positive)$ by $P(Positive)$.

$$P(C|Positive) = \frac{P(C, Positive)}{P(Positive)} \tag{5}$$

$$P(\neg C|Positive) = \frac{P(\neg C, Positive)}{P(Positive)} \tag{6}$$

Finally, the Bayes equation could be generalized as:

$$P(A|B) = \frac{P(A) * P(B|A)}{P(B)} \tag{7}$$

4.8 Python Probability Practice `Lesson 08`

Some methods interesting to take notes.

4.8.0.1 `.random.randint()`

Generates a random value.

```
np.random.randint(0, 2, size=10000)
```

In the example above, the line code will generate a sample of 10,000 number from 0 to 1 (the 2 is not inclusive).

4.8.0.2 .random.choice()

Generates a radom value with different loads.

```
np.random.choice([0, 1], size=10000, p=[0.8, 0.2]))
```

In the example the loads are 0.8 and 0.2.

4.8.0.3 random.binomial()

This method is an other way to simulate a coin flip.

```
np.random.binomial(10, 0.5, 1000000)
```

The example above will flip 10 coins (with a fair rate due to the 0.5), with a sample of 1 million.

The result of this methods is a "aggregation" of the success events, which means the output varies from 0 to 10.

4.9 Normal Distribution Theory Lesson 09

Comparison between a simple probability, binomial distribution, and normal distribution.

Type	Quantity
Bernoulli	1
Binomial	10
Normal	1000

Along this chapter I have seen the evolution from the simple probability (Bernoulli), to a Binomial, and finally a Normal distribution.

The difference is the size of the "sample".

4.9.1 Equations

- Bernoulli

$$P(HEADS) = P(HEADS)^n \tag{1}$$

- Binomial

$$P(n,k) = \frac{n!}{(n-k)!k!} p^k * (1-p)^{n-k} \tag{2}$$

- Normal (or Gaussian or Gauss or Laplace–Gauss) distribution

$$N(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \tag{3}$$

μ: mean; σ^2: variance.

4.10 Sampling distributions and Central Limit Theorem `Lesson 10`

Recap of Inferential Statistics.

4.10.0.1 Inferential Statistics

In order to assure you are comfortable with the ideas surrounding inferential statistics. The next 4 concepts are aimed at the difference between Descriptive and Inferential Statistics. Actually applying inferential statistics techniques to data will be taught in later lessons.

4.10.0.2 Probability to Statistics

This begins a set of lessons that will be more data oriented in the way that you are applying ideas, and less probability oriented.

Often we use statistics to verify conclusions of probability through simulation. Therefore, simulation (similar to the python lesson you completed earlier) will be a large part of the way we show mathematical ideas moving forward.

4.10.0.2.1 Solutions

It is in your best interest to work through the solution notebooks on your own before looking at the solutions available for this course. However, if you get stuck or would like to double check your solutions, notice all of the solutions and data are available in the resources tab of this course. This is true for all of the following lessons as well. — Class notes

4.10.0.3 Comparison Descriptive and Inferential Statistics

In this section, we learned about how Inferential Statistics differs from Descriptive Statistics.

- `Descriptive statistics` is about describing our collected data using the measures discussed throughout this lesson: measures of center, measures of spread, shape of our distribution, and outliers. We can also use plots of our data to gain a better understanding.

- `Inferential Statistics` is about using our collected data to draw conclusions to a larger population. Performing inferential statistics well requires that we take a sample that accurately represents our population of interest.

A common way to collect data is via a survey. However, surveys may be extremely biased depending on the types of questions that are asked, and the way the questions are asked. This is a topic you should think about when tackling the the first project.

We looked at specific examples that allowed us to identify the

- Population - our entire group of interest.
- Parameter - numeric summary about a population
- Sample - subset of the population
- Statistic numeric summary about a sample

4.10.1 Sampling distribution

A sampling distribution is the distribution of a statistic. Here we looked the distribution of the proportion for samples of 5 students. This is key to the ideas covered not only in this lesson, but in future lessons.

4.10.2 Notation

Figure 1

As you saw in this video, we commonly use Greek symbols as parameters and lowercase letters as the corresponding statistics. Sometimes in the literature, you might also see the same Greek symbols with a "hat" to represent that this is an estimate of the corresponding parameter.

4.10.3 Other Sampling Distribution

It is possible to use other statistics.

- Standard Deviation
- Variance
- Difference in Means

The difference between parameters and statistics is the last one is varies and the first is fixed.

4.10.4 Law of Large Number

This theorem preconizes the greater the number of the samples/trials the average of this sample will be close to the population mean. This is the reason we have simulated samples sizes of 10,000.

- Increasing the size of the sample the mean of this sample will be closer to the population mean.

Read more in Wikipedia and Investopedia

4.10.5 Central Limit Theorem

Quite similar with the LLN theorem, but this is related to the shape of sample. It is necessary to plot a histogram to see the miracle.

- Increasing the size of the sample the shape of the sample will be closer to a normal distribution.

Figure 2

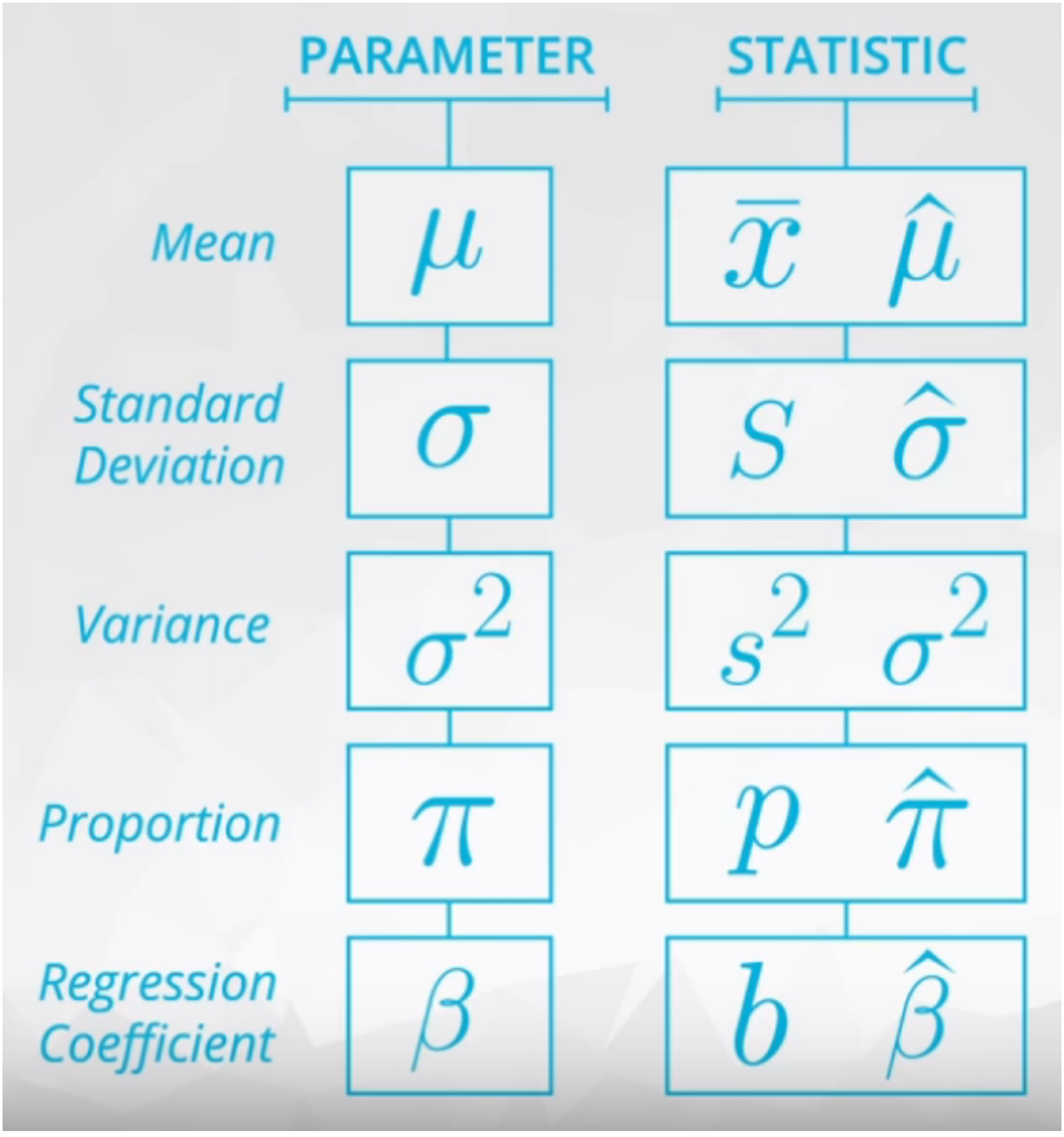
PARAMETER
STATISTIC
Mean
μ
$\bar{x}$ $\hat{\mu}$
Standard Deviation
σ
S $\hat{\sigma}$
Variance
σ^2
s^2 σ^2
Proportion
π
p $\hat{\pi}$
Regression Coefficient
β
b $\hat{\beta}$

Figure 4.13:

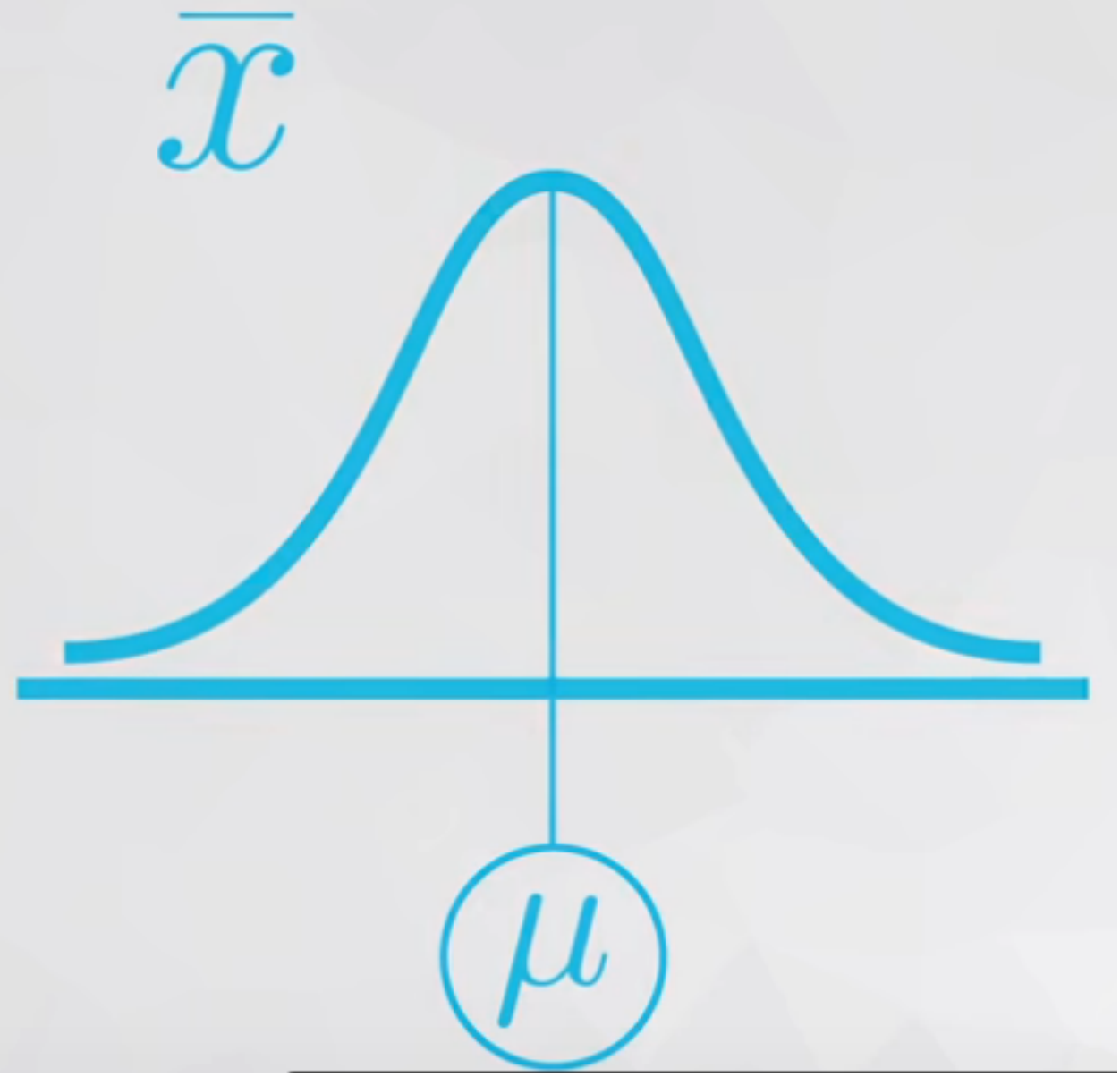

Figure 4.14:

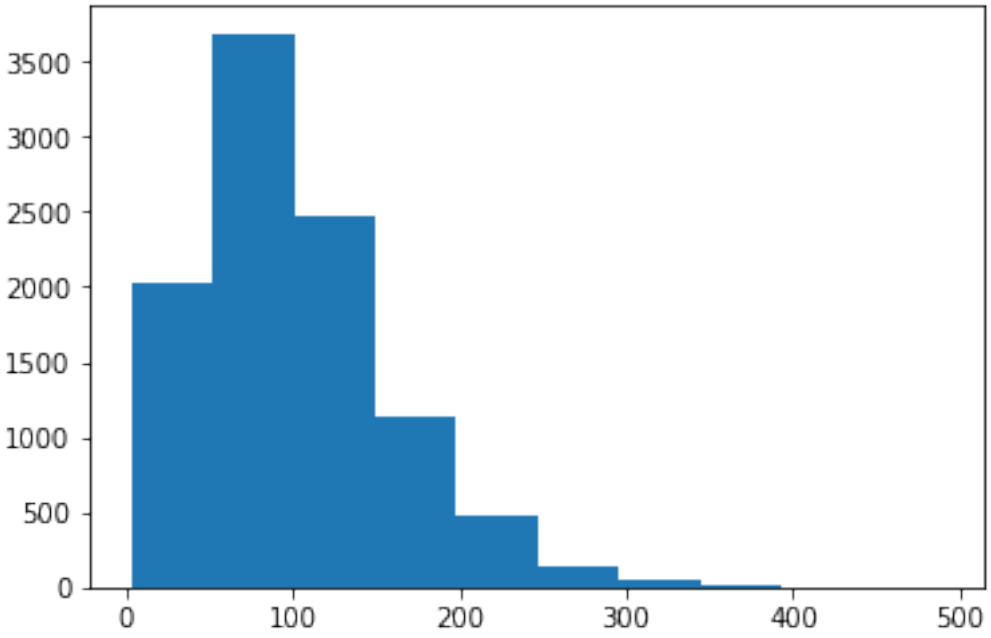

Figure 4.15:

The Central Limit Theorem states that with a large enough sample size the sampling distribution of the mean will be normally distributed.

The Central Limit Theorem actually applies for these well known statistics:

1. Sample means ($\bar{x}$)
2. Sample proportions (p)
3. Difference in sample means ($\bar{x}_1 - \bar{x}_2$)
4. Difference in sample propostions ($p_1 - p_2$)

But is not applied to:

1. Variance or Standard Deviation
2. Correlation coeficient
3. Maximum value

And it applies for additional statistics, **but it doesn't apply for all statistics!**. You will see more on this towards the end of this lesson.

Examples CLT in Figures 3 and 4.

Figure 3

Figure 4

Varying the value of the sample from 3 to 100, the bell shape could be identified.

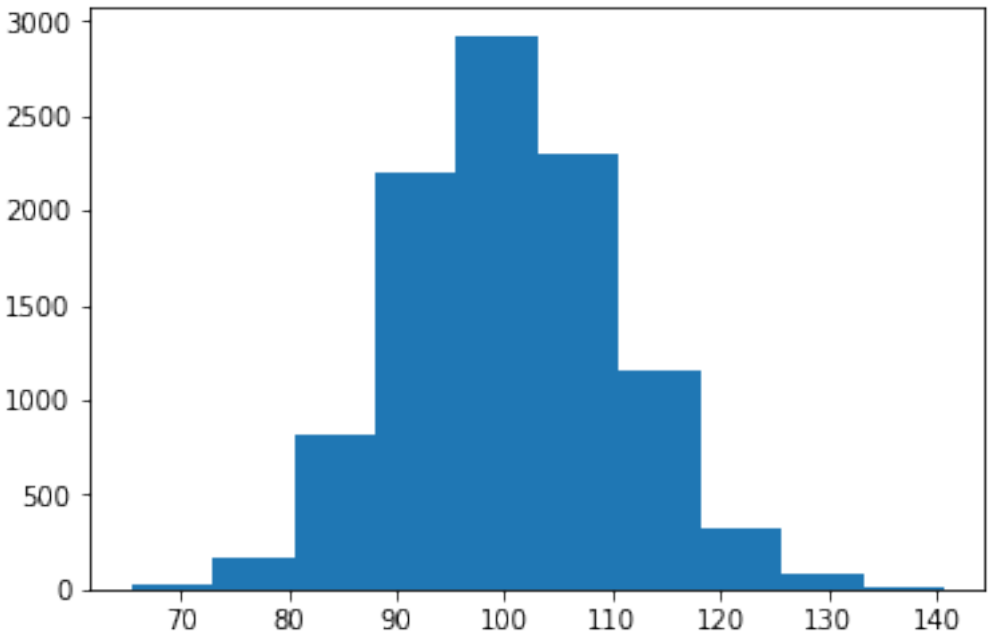

Figure 4.16:

4.10.6 Bootstrapping

Bootstrapping is a technique where we sample from a group with replacement.

We can use bootstrapping to simulate the creation of sampling distribution, which you did many times in this lesson.

By bootstrapping and then calculating repeated values of our statistics, we can gain an understanding of the sampling distribution of our statistics.

4.11 Confidence Intervals `Lesson 11`

Have in mind this is a part of Inference Statistical.

In this lesson we are going to use the sampling method to create our interval of confidence. The steps is very simple:

- From a population gather a sample;
- Based on this sample draw many new samples (e.g. 10,000);
  - The size of these 10,000 samples should be "large" (e.g. 200);
  - You can use the `replace = True` to work with Bootstrapping;
- Calculate some statistics (of this 200 elements): mean or difference of means;
- Plot a histogram of this statistics (should have 10,000 elements to plot).

Plotting this histogram you can take some conclusions about the data. For example:

- If you decided to use difference of means, you can check if the two "samples" are equals, or greater or less than.

  Assuming you control all other items of your analysis:

  1. Increasing your sample size will decrease the width of your 2. confidence interval.
  2. Increasing your confidence level (say 95% to 99%) will increase the width of your confidence interval.

  You saw that you can compute:

  1. The confidence interval **width** (Confidence Interval Width) as the difference between your upper and lower bounds of your confidence interval.
  2. The **margin of error** (MOE) is half the confidence interval width, and the value that you add and subtract from your sample estimate to achieve your confidence interval final results.
     — Udacity class notes

4.11.1 Practical and Statistical Significance

- Confidence Intervals will provide a Statistical Significance to be used as bedrock of a conclusion.
- Practical Significance is something from outside of the Confidence Interval and could decide if this solution is good or not. The Practical Significance takes account other variables (costs, lack of employee, time, etc.).

4.11.2 Traditional Methods vs Bootstrapping Confidence Intervals

The Bootstrapping Methods can perform all traditional methods of Confidence Intervals with minors differences when the sample is a true representation of the population.

> With large sample sizes, these end up looking very similar. With smaller sample sizes, using a traditional methods likely has assumptions that are not true of your interval. Small sample sizes are not ideal for bootstrapping methods though either, as they can lead to misleading results simply due to not accurately representing your entire population well.

Traditional Methods of Confidence Intervals:

- T-Test: Population mean;
- Two sample T-test: Comparing two means;
- Z-Test;
- Chi-squared test;
- F-test.

4.11.3 Misinterpretation of Confidence Intervals

The aim of a Confidence Interval is to calculate parameter, a single value of a entire population, which could be:

- mean
- standard deviation

- difference between two means (two populations means)
- Any other numeric summary in the population

Confidence Interval **do not** allow us to tell about any specific individual of this population.

> Confidence intervals take an aggregate approach towards the conclusions made based on data, as these tests are aimed at understanding population parameters (which are aggregate population values).
>
> Alternatively, machine learning techniques take an individual approach towards making conclusions, as they attempt to predict an outcome for each specific data point.
>
> In the final lessons of this class, you will learn about two of the most fundamental machine learning approaches used in practice: linear and logistic regression. — Udacity Class notes

4.11.4 New Methods

4.11.4.1 `.sample()`

Will sample a data frame, you can use `replace = True` if you are wondering perform a bootstrap.

```
my_population.sample(100, replace = True)
```

For above example the sample size is 100 and there is replacement (which denotes it is a booostrapping process).

4.11.4.2 `.percentile()`

This is a numpy method, and gives the percentile from a given value.

```
np.percentile(df, 0.5), np.percentile(df, 99.5)
```

There are two `.percentile()` representing a two tailed confidence interval of 99%.

4.12 Hypothesis Testing `Lesson 12`

Is a way to answer a given question. First step is convert/transform this question in hypothesis. Then, gather data to answer/justify which hypothesis are likely to be true.

Example: Hypothesis Examples.

My question:

What is the most favorite ice cream flavor?

I have pose two hypothesis about ice cream.

H_0 : Chocolate is most favorite.H_1 : Vanilla is most favorite.

After pose hypothesis, I need to collect data to support my hypothesis (if is `True` or `False`).

Have in mind, Confidence Intervals and Hypoteshis Testing allow us to draw conclusions about the **population** only using **sample data**.

H_0 - Null hypotheses

The H_0 hypothesis is also known as *Null hypothesis* and this hypothesis stand to be true **before** collect any data.

Commonly, the Null hypothesis is a statement of two groups being equal or "zero" effect.

Usually, the Null hypothesis tend to hold these mathematical operators:

1. $=$
2. $\leq$
3. $\geq$

H_1 - Alternative hypotheses

The H_1 hypothesis must be competing (in comparison with the H_0) and non-overlaping hypothesis.

This hypothesis is always associate with what we want or what we want to prove is `True`.

Usually, the Alternative hypothesis tend to hold these mathematical operators:

1. $\neq$
2. $>$
3. $<$

Example: Innocent until proven guilty.

My question:

Guilty or innocent?

Which could be translated as:

H_0 : We believe everyone to be innocent initially.H_1 : Guilty.

Before any collect data the H_0 hypothesis is `True`, and we will collect data/evidence to test which hypothesis is supported.

Example: New website's version.

My question:

Which version of website has more traffic?

In terms of hypothesis:

H_0 : The newer version has less traffic than the older version.H_1 : The newer version has more traffic than the older version.

Using mathematical notation and based on the average traffic (μ).

$$H_0 : \mu_{new} \leq \mu_{old} H_1 : \mu_{new} > \mu_{old}$$

Bear in mind, the H_1 hypothesis is our expectation (the new website has a better performance than the older version), and now we need to collect/gather data to analyze which hypothesis is supported.

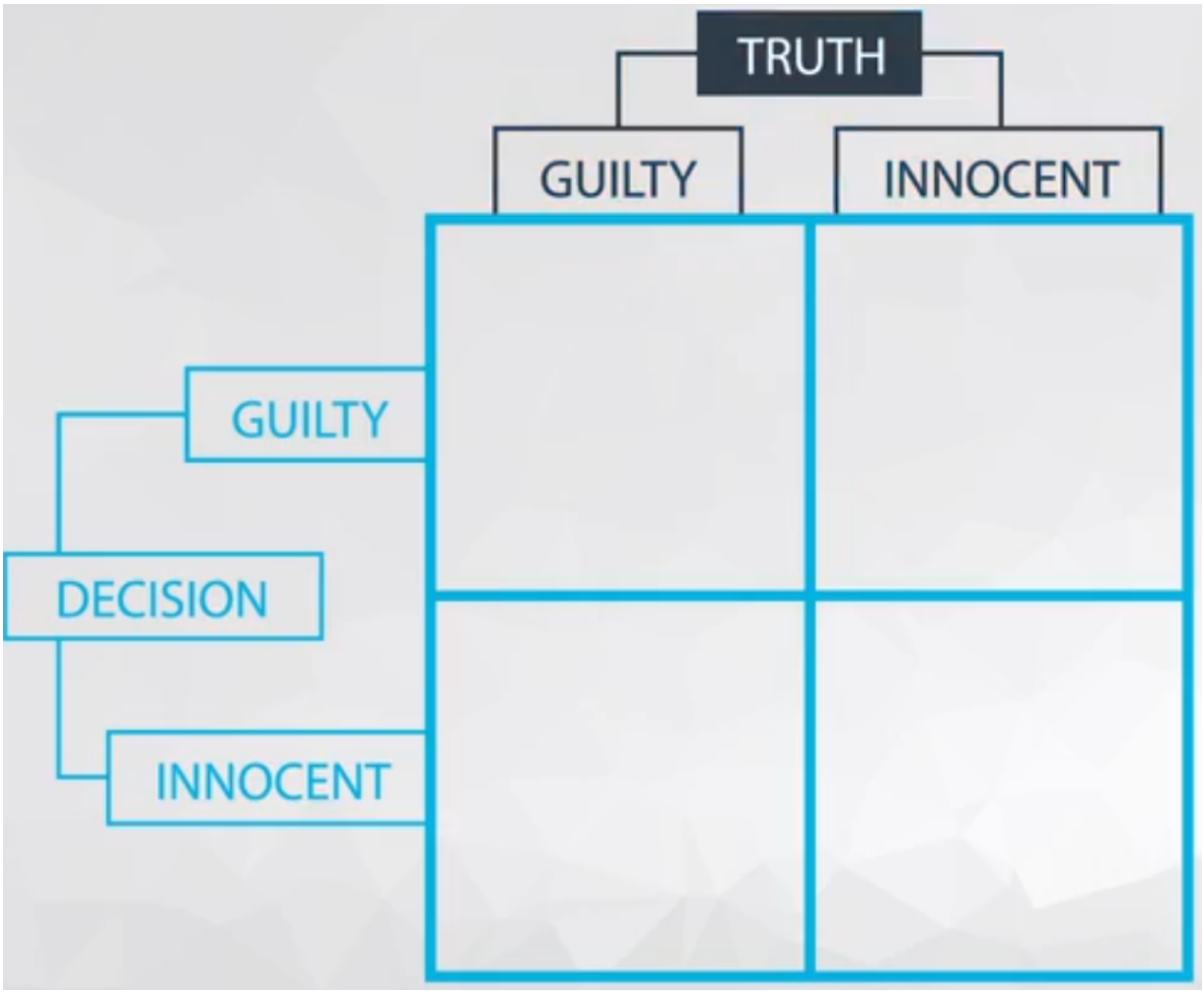

Figure 4.17:

4.12.1 Type of Errors

Before any explanation about erros, let's ilustrate the four (4) potential outcome of a given H_0 and H_1 in Figure 1, based on the judicial example.

Figure 1 - Four potential outcome from a Hypothesis Testing.

We can classify each of this outcome as ilustraded in Table 1.

Table 1 - Decision Classification.

GuiltyTruth

InnocentTruth

GuiltyJury

correct decision 1

mistake 1 - False Positive

InnocentJury

mistake 2 - False Negative

correct decision 2

- correct decision 1: Innocent person judged as innocent :+1: ;
- correct decision 2: Guilty person judged as guilty :+1: ;
- mistake 1: Innocent person judged as guilty :-1: ;
  - So-called **Type 1 Errors**;
- mistake 2: Guilty person judged as innocent :-1: ;
  - So-called **Type 2 Errors**.

4.12.1.1 Type 1 Errors (α)

This is the mistake 1 from **Table 1**.

The worse of the two types of errors.

Put a innocent person in jail.

This happens when the H_1 (alternative hypothesis) in chosen, but actually the H_0 is `True`.

False Positive

4.12.1.2 Type 2 Errors (β)

This is the mistake 2 from **Table 2**.

Let a guilty person free.

This happens when the H_0 (null alternative) in chosen, but actually the H_1 is `True`.

False Negative

4.12.1.3 Meaning of α

What is the meaning of α?

This is a threshold of how many of this kind of error (the worst one!) we are allowed to commit, leting the rest of the erros in type 2 errors. Generally, this α is quite low value, and varies according to the field.

- α
  - Medical: 0.01
  - Business and Research: 0.05

Example: Sky diving using Parachute.

In this job you are in charge of check the parachutes.You need to decide if the parachute is well prepare to a sky dive. There are 4 possible outcomes from your work.

- Accept a well prepare parachute;
  - This is a correct decision;

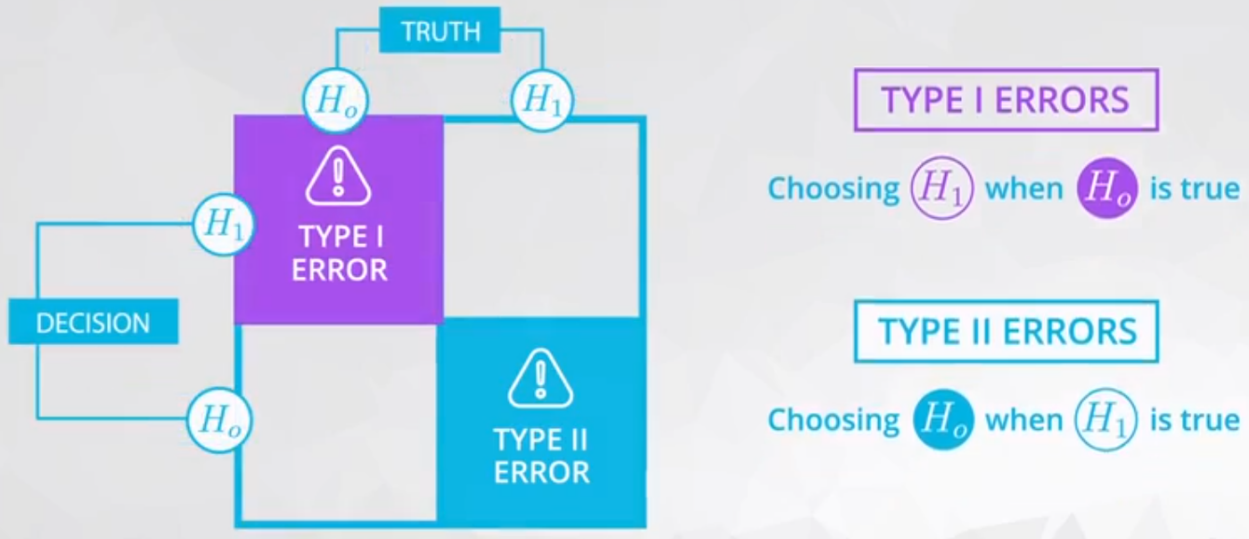

Figure 4.18:

- Accept a shortcoming parachute;
  - This is an error type 1 the worse outcome because it will kill the skydiver;
- Reject a well prepare parachute;
  - This is an error type 2 because the parachute is OK but you wrongly reject it;
- Reject a shortcoming parachute;
  - This is a correct decision because the parachute is not well prepare.

Figure 2 ilustrate it.

Figure 2 - Four potential outcome from a Hypothesis Testing of a Skydiver.

Bear in mind, for this example an α of 0.01 is still very high.

4.12.1.4 Common Hypothesis Testing

- T-test: population mean
- Two sample t-test: difference in means
- Paired t-test: Comparing after and before of a same individual
- One sample z-test: population proportion
- Two sample z-test: difference between population proportion

4.12.2 Selecting a hypothesis

Which hypothesis is more likely to be True?

There are two ways to select these hypothesis.

1. Using Confidence Intervals: Sampling distribution of our statistics.
2. Simulating what we believe to be true under the null hypothesis, and than seeing if our data is actually consistent with that

4.12.2.1 Using Confidence Intervals

Create the Confidence Intervals and check where is it.

In the example of coffee drinkers, the interval was entirely below 70, which would suggest the null (the population mean is less than 70) is actually true.

1. Bootstrapping a sample
2. Calculate the statistics
3. Plot the histogram to visualize
4. Calculte the upper and lower bounds from the Confidence Intervals
5. Check if the H_0 or H_1 is in this Confidence Interval.

4.12.2.2 Traditional way

1. We assume the H_0 is `True`.
2. We know the sampling distribution is normal
3. We will use the closest value of the H_0, which is almost 70.
4. Based on the standard deviation of the sampling distribution we could estimate the distribution from the H_0.
5. Plot the histogram
6. Check the statistics and the hypothesis H_0.
7. Decide to reject H_0 or not.

4.12.3 P-value

Based on the Bootstrapping process the P-value could be interpreted as: a statistics of how many samples will be higher/lower than the threshold defined in the hypothesis.

In the exercises the P-value is an "average" of all 10,000 samples if it is higher, lower, or in the tails of a specific parameter. This "average" (in fact is a "vector" of zero or one due to the comparison) is the probability of a sample has higher/lower values from the parameter.

Bear in mind, if 100 samples say to reject the H_0 but the others 9,900 say the opposite there are a probability of 100/10,000 to incurr in error Type 1. In other words, the 100/10,000 is the so-called **p-value**.

The relationship between **p-value** and α:

- **p-value** $< \alpha$ (or small p-value): Reject H_0
- **p-value** $\geq \alpha$ (or Large p-value): Fail to reject H_0

Reference

For a small α (less than 10/10,000, for instance) probably you will accept the H_0, in the case of:

$$H_0 : \mu \leq 0 H_1 : \mu > 0$$

There are three forms to allocate the α, as you can see in Figure 3.

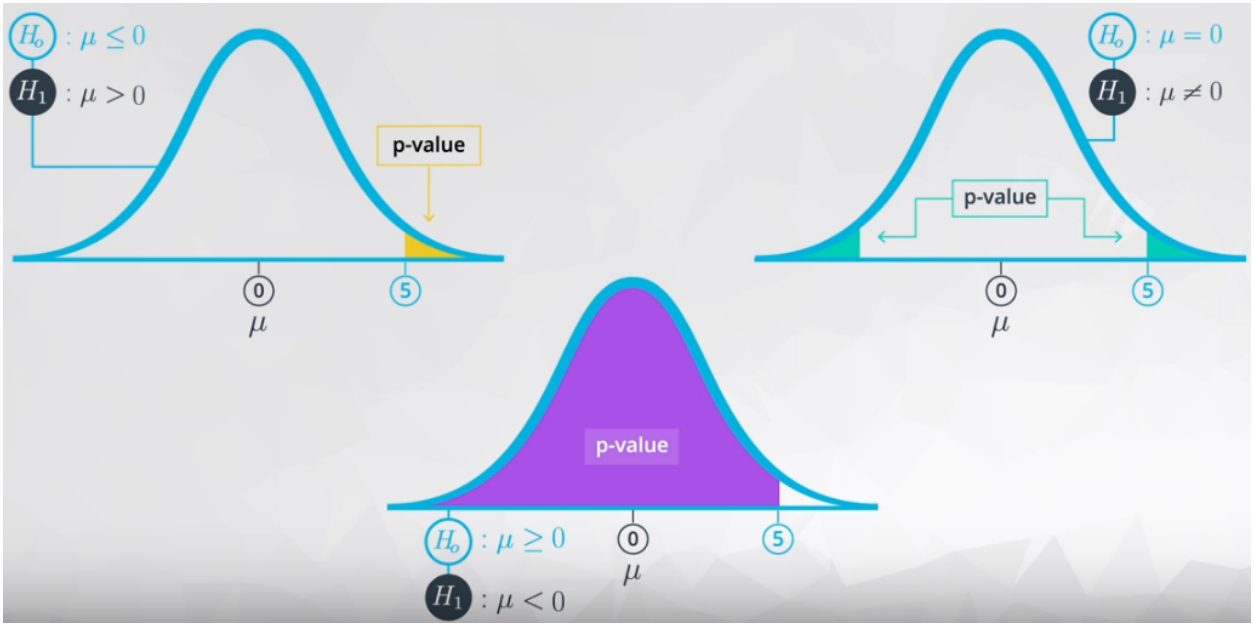

Figure 4.19:

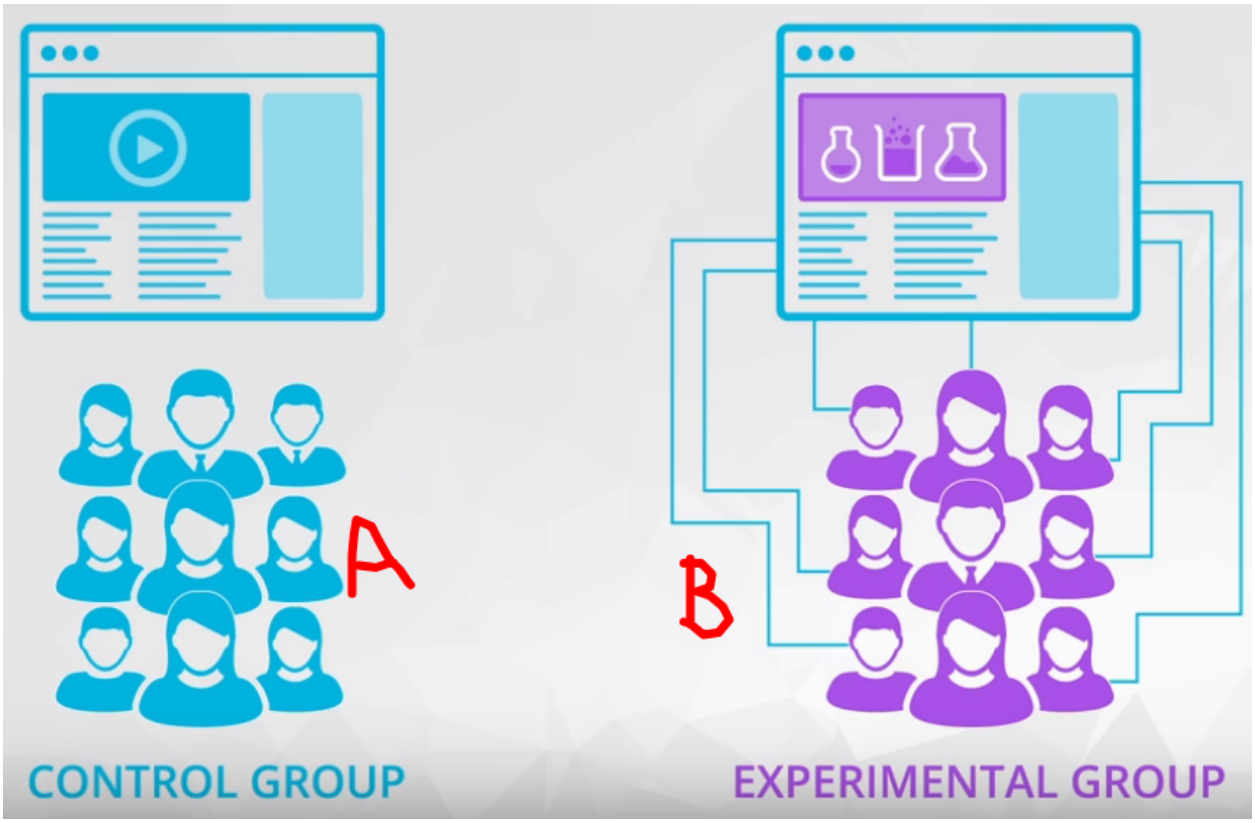

Figure 4.20:

4.13 A/B Testing `Lesson 13`

This is an application of Confidence Intervals and Hypotheses Testing.

The A/B Testing is a comparison between two groups (A and B), as ilustrated in Figure 1.

Figure 1 - Two group to be tested.

> A/B tests are used to test changes on a web page by running an experiment where a control group sees the old version, while the experiment group sees the new version. A metric is then chosen to measure the level of engagement from users in each group. These results are then used to judge whether one version is more effective than the other. A/B testing is very much like hypothesis testing with the following hypotheses:

H_0 : The new version is equal or worse than the older version.H_1 : The new version is better than the older version.

Decision:

- If we fail to reject the null hypothesis, the results would suggest keeping the old version, or;
- If we reject the null hypothesis, the results would suggest launching the change.

Drawbacks

> It can help you compare two options, but it can't tell you about an option you haven't considered. It can also produce bias results when tested on existing users, due to factors like change aversion and novelty effect.
>
> - Change Aversion: Existing users may give an unfair advantage to the old version, simply because they are unhappy with change, even if it's ultimately for the better.
> - Novelty Effect: Existing users may give an unfair advantage to the new version, because they're excited or drawn to the change, even if it isn't any better in the long run.

4.13.1 Example: New Homepage

The Audacity company want to perform an A/B Testing of two versions of a new homepage.

$$H_0 : CRT_{new} - CTR_{old} \leq 0 H_1 : CRT_{new} - CTR_{old} > 0$$

Where CRT stands to Click Through Rate.

There are two version: `control` and `experiment`.

The difference between the CRT is about 0.03.

Bootstrapping

Example: New version of Home page.

The bootstrapping provide a histogram presented in Figure 2.

The null hypothesis histogram is showed in Figure 3.

Finally, Figure 4 ilustrate both histogram.

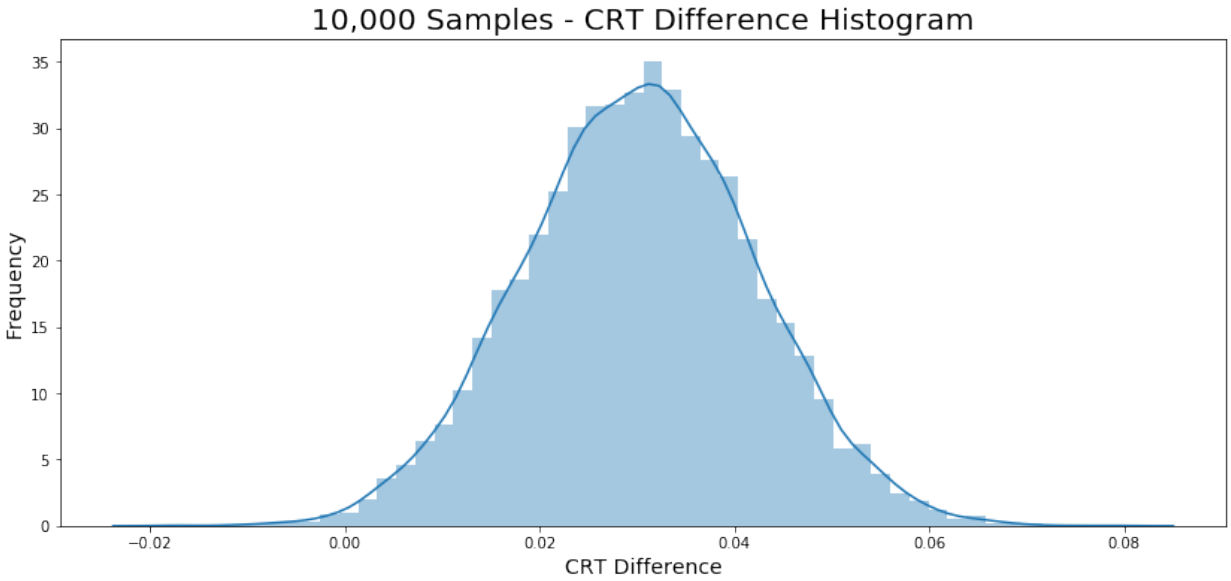

Figure 4.21:

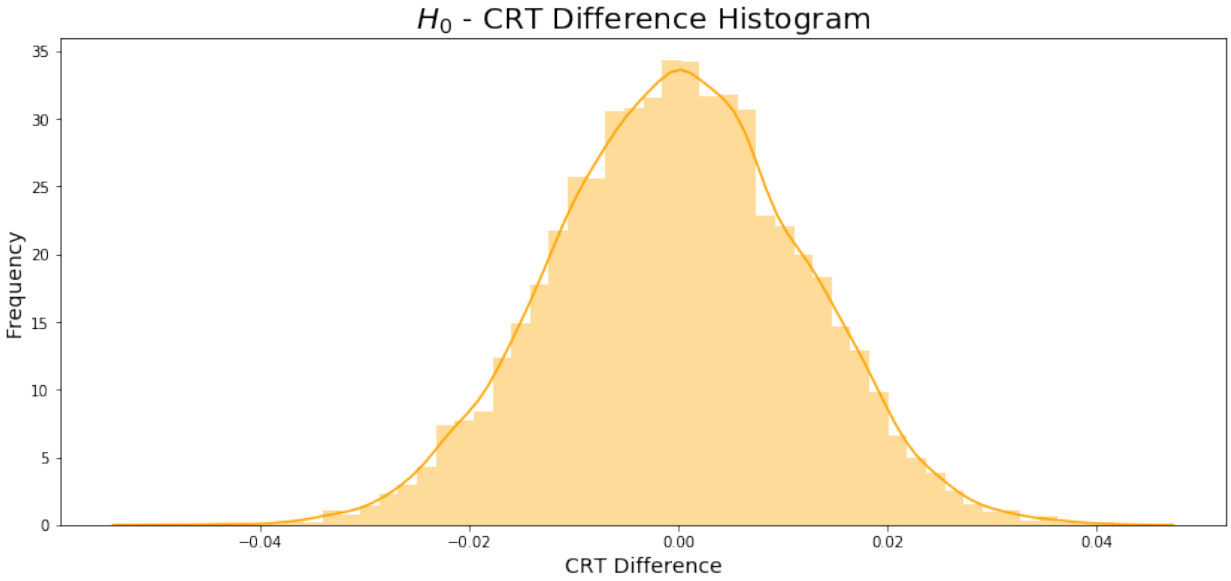

Figure 4.22:

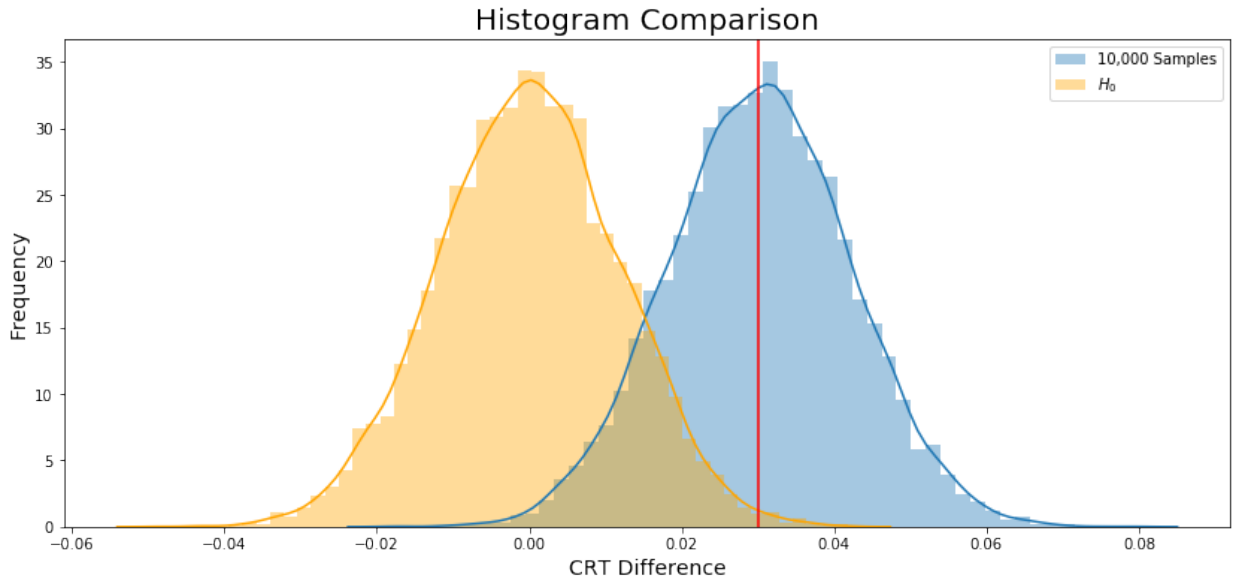

Figure 4.23:

P-value

Founded on the entire population (excepting the duplicated user id, etc.), I have calculated the `diff`.

`diff = experiment_crt - control_crt = 0.030034443684015644`

The `diff` could be interpreted as a threshold which I will use as delimiter, to do it I will calculate the proportion of H_0 (orange graph) that has a difference between CRT's higher than `diff`.

For this reason, I will calculate the average of a list of `bool`, which will return the proportion I want.

Based on the `p_value` of 0.5% we reject the H_0.

> **Conclusion:** Audacity should launch the new version of the home page.

4.13.2 Example: Average Reading Time

Same idea, two version of a website, one `control` and other `experiment`.

- Average Reading time of control: 115.38637100678429
- Average Reading time of experiment: 131.3208410471793
- Difference observed: 15.9

On average, visitor using the experiment version of website spent almost 16 more seconds.

Hypotheses posed:

$$H_0 : ART_{new} - ART_{old} \leq 0 H_1 : ART_{new} - ART_{old} > 0$$

Where ART stands to Average Reading Time.

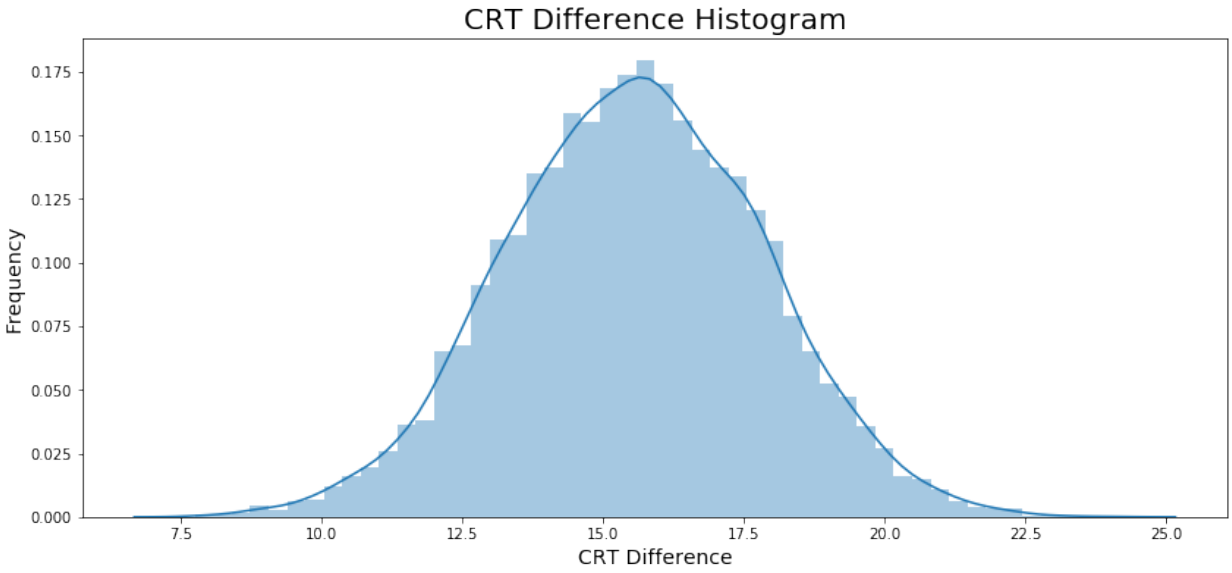

Figure 4.24:

Bootstrapping

Let's apply the bootstrapping, and plot a histogram in Figure 5.

The null hypothesis histogram is showed in Figure 6.

Finally, Figure 7 ilustrate both histogram.

P-value

The `p_value` is zero.

> **Conclusion:** Reject the H_0 because p_value $< \alpha$

Where α is 0.05.

4.13.3 Example: Enrollment Rate

This is an example to show a case where the H_0 is failed to reject.

I will use the same principle of CRT to evalute the Enrollment rate.

- Enrollment rate control: 0.23452157598499063
- Enrollment rate experiment: 0.2642986152919928
- Difference observed: 0.02977703930700215

Hypotheses posed:

$$H_0 : ER_{new} - ER_{old} \leq 0 H_1 : ER_{new} - ER_{old} > 0$$

Where ER stands to Enrollment Rate.

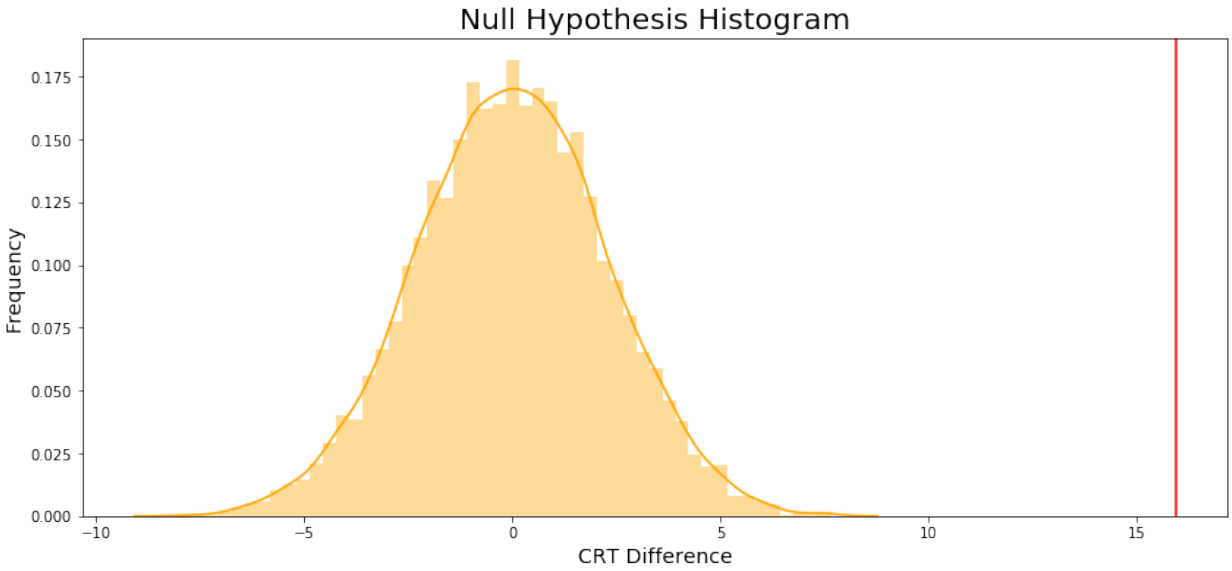

Figure 4.25:

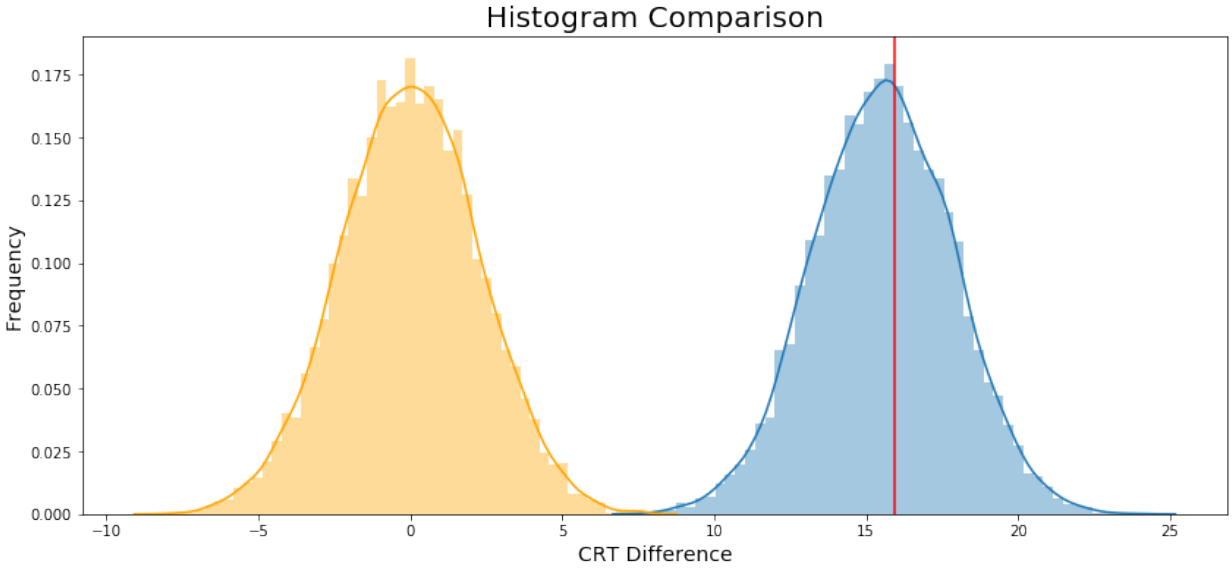

Figure 4.26:

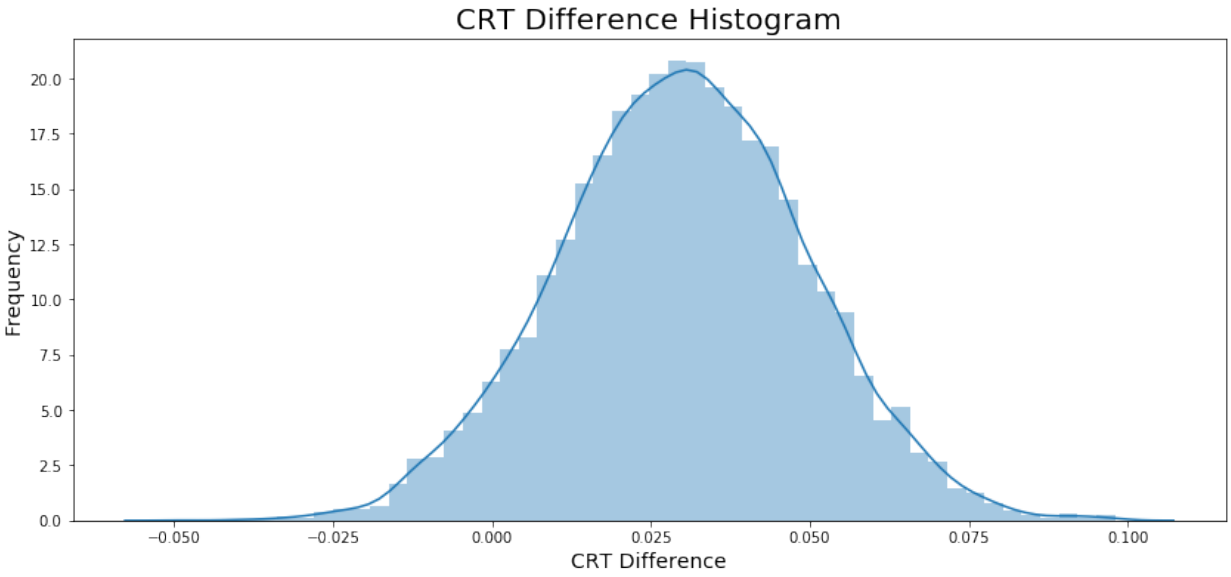

Figure 4.27:

Bootstrapping

Let's apply the bootstrapping, and plot a histogram in Figure 8.

The null hypothesis histogram is showed in Figure 9.

Finally, Figure 10 ilustrate both histogram.

P-value

The `p_value` is 0.0624.

> **Conclusion:** Due to p_value $> \alpha$ we fail to reject H_0.

Where α is 0.05.

4.13.4 Bonferroni Correction

> If you remember from the previous lesson, the Bonferroni Correction is one way we could handle experiments with multiple tests, or metrics in this case. To compute the new bonferroni correct alpha value, we need to divide the original alpha value by the number of tests.

The new α will be:

$$\alpha_{adjusted} = \frac{\alpha}{4} = \frac{0.05}{4} = 0.0125$$

Based on the several test we have done:

- Enrollment Rate: 0.0624 (Read the Jupyther Notebook)
- Average Reading Duration: 0 (Read the Jupyther Notebook)

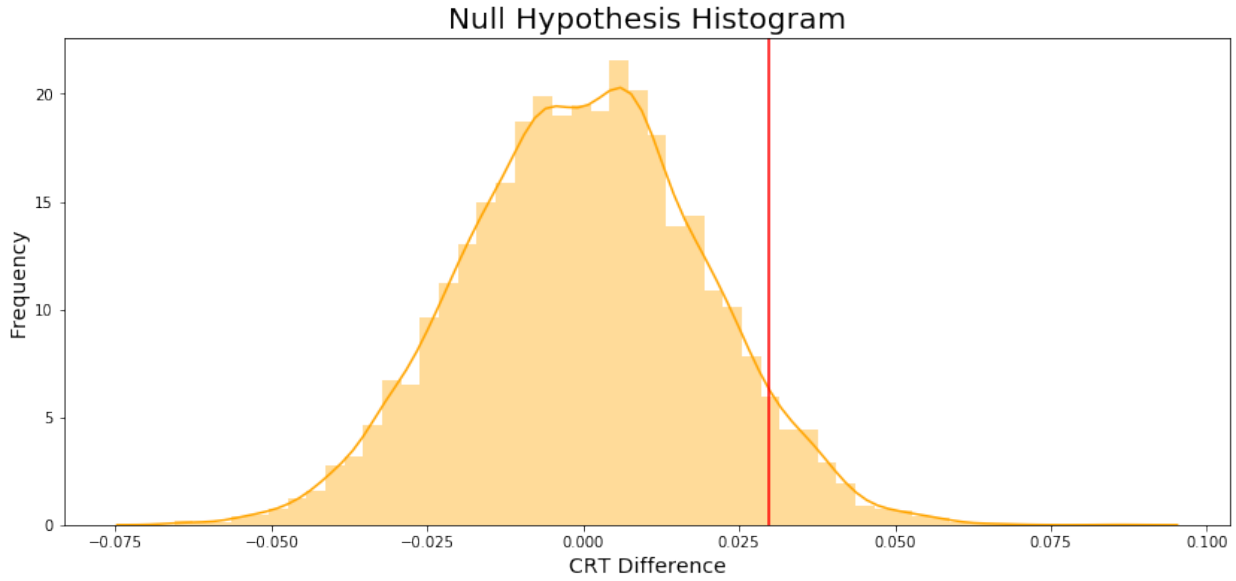

Figure 4.28:

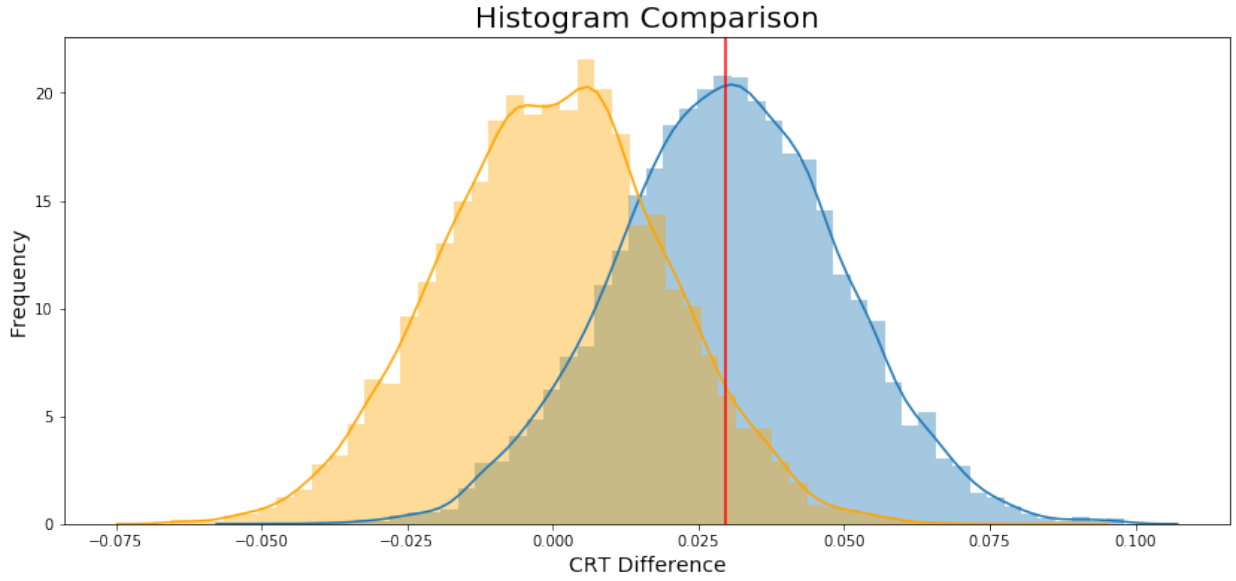

Figure 4.29:

- Average Classroom Time: 0.0384 (Read the Jupyther Notebook)
- Completion Rate: 0.0846 (Read the Jupyther Notebook)

This new α will generate only **one** A/B Testing statistical significant.

New Feature	p value	$\alpha_{adjusted}$	Result	H_0
Enrollment Rate	0.0624	0.0125	$>$	Fail to reject H_0
Average Reading Duration	0	0.0125	$<$	Reject H_0
Average Classroom Time	0.0384	0.0125	$>$	Fail to reject H_0
Completion Ratee	0.0846	0.0125	$>$	Fail to reject H_0

This is the reason the Bonferroni method is considered conservative.

4.13.5 Difficulties in A/B Testing

As you saw in the scenarios above, there are many factors to consider when designing an A/B test and drawing conclusions based on its results. To conclude, here are some common ones to consider.

- Novelty effect and change aversion when existing users first experience a change
- Sufficient traffic and conversions to have significant and repeatable results
- Best metric choice for making the ultimate decision (eg. measuring revenue vs. clicks)
- Long enough run time for the experiment to account for changes in behavior based on time of day/week or seasonal events.
- Practical significance of a conversion rate (the cost of launching a new feature vs. the gain from the increase in conversion)
- Consistency among test subjects in the control and experiment group (imbalance in the population represented in each group can lead to situations like Simpson's Paradox) — Udacity notebook

4.14 Regression Lesson 14

Bear in mind, regression is a subject of the Supervised branch of Machine Learning. Figure 1 shows a simple big picture.

Figure 1 - Machine Learning Branches.

Some hot points in these two branches (there are other branches, but these two are the most know).

- Supervised: A machine learning technique where we are attempting to predict a label based on inputs.
  - Predict fraudulent transactions
  - Predict chance of default on a loan
  - Predict home prices
- Unsupervised: A machine learning technique where we are attempting to group together unlabeled data based on similar characteristics.
  - Customer segmentation
  - Group document that cover similar topics

The Linear and Logistic Regression fall into the Supervised Machine Learning branch.

Figure 4.30:

4.14.1 Introduction to Linear Regression

- Response variable or dependent (y): The variable you are interested in predicting, and;
- Explanatory variable or independent (x): The variable used to predicted the response.

> A common way to visualize the relationship between two variables in linear regression is using a scatterplot. You will see more on this in the concepts ahead. — Udacity notebook

Figure 2 shows an example of a scatter plot.

Figure 2 - Hours studying vs Test grades.

4.14.1.1 Scatter Plot

> Scatter plots are a common visual for comparing two quantitative variables. A common summary statistic that relates to a scatter plot is the **correlation coefficient** commonly denoted by `r`.
>
> Though there are a few different ways to measure correlation between two variables, the most common way is with Pearson's correlation coefficient. Pearson's correlation coefficient provides the:
>
> 1. Strength
> 2. Direction
>
> of a linear relationship. Spearman's Correlation Coefficient does not measure linear relationships specifically, and it might be more appropriate for certain cases of associating two variables. — Udacity notebook

Figure 3 shows an example of strong positive relationship.

Figure 3 - Strong and Positive Relantionship.

When x increase y also increase, and the points is very close from each other. Figure 4 shows the opposite of positive direction.

Figure 4 - Moderate and Negative Relantionship.

When x increase y decrease, and the points is a bit sparse. Figure 5 shows an example of scatter plot with weak strength.

Figure 5 - Weak and Negative (??) Relantionship.

Both, strength and direction is capture by the correlation (`r`).

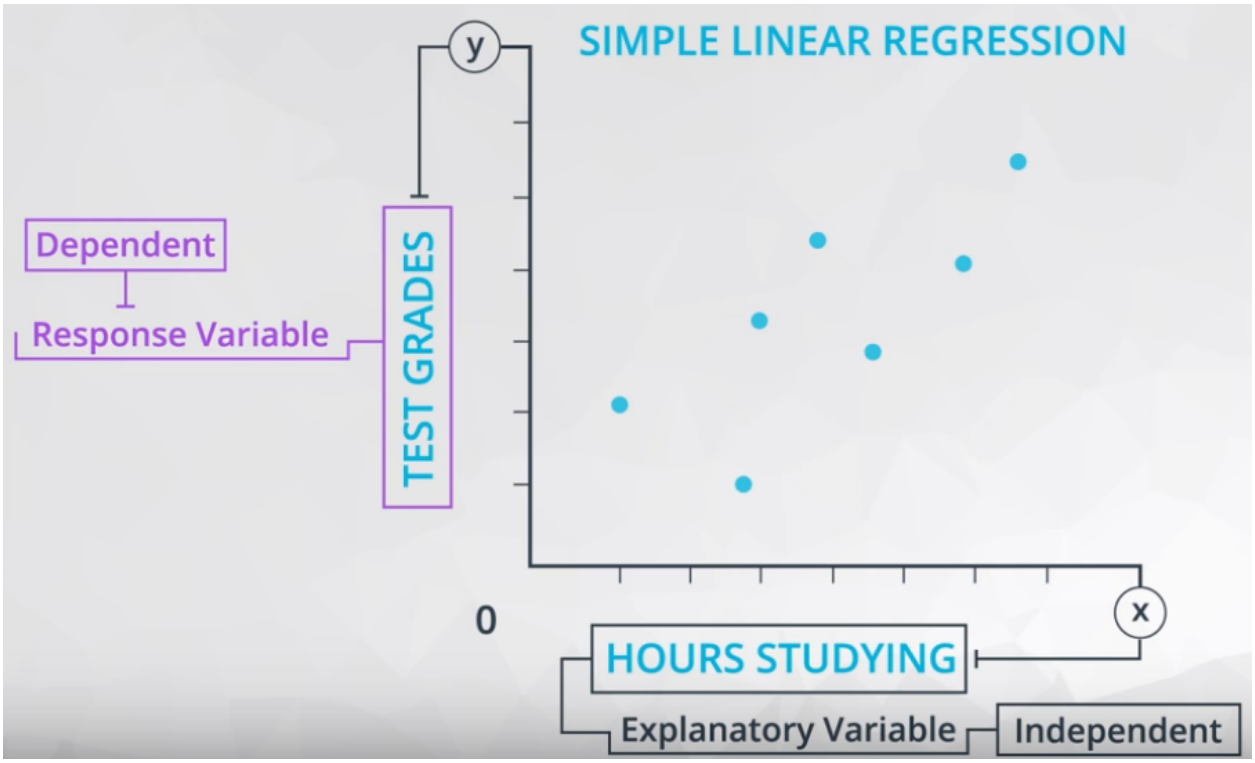

Figure 4.31:

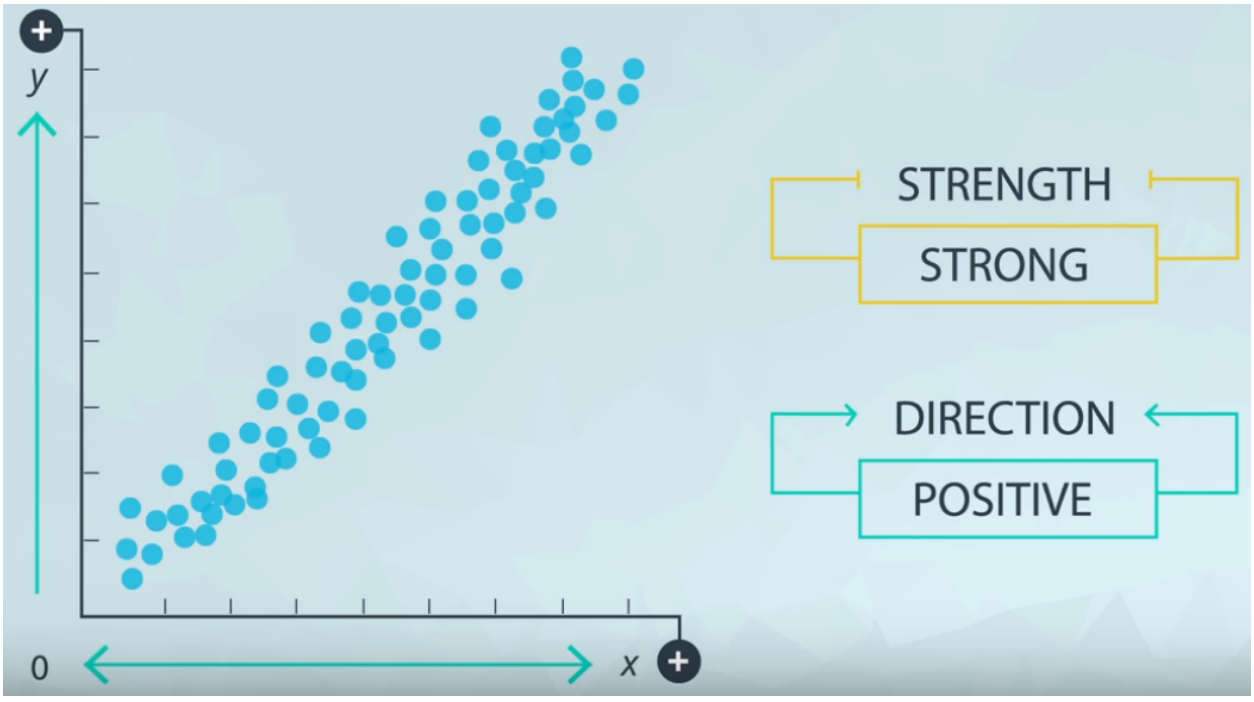

Figure 4.32:

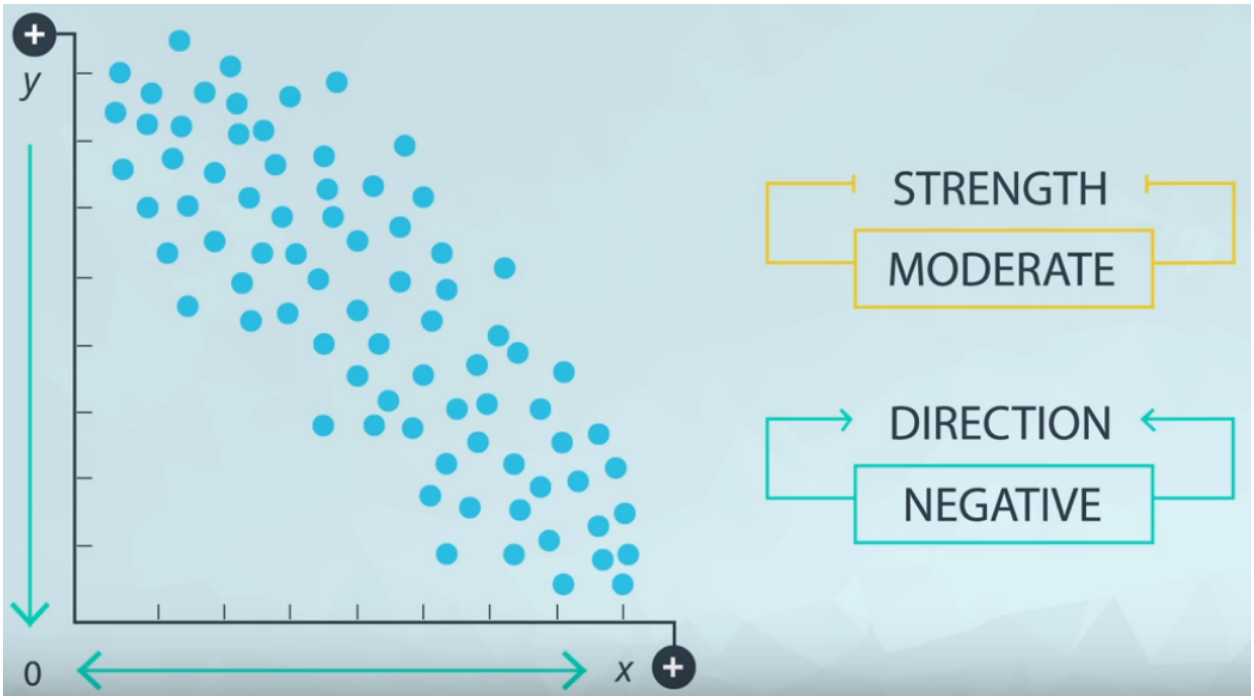

Figure 4.33:

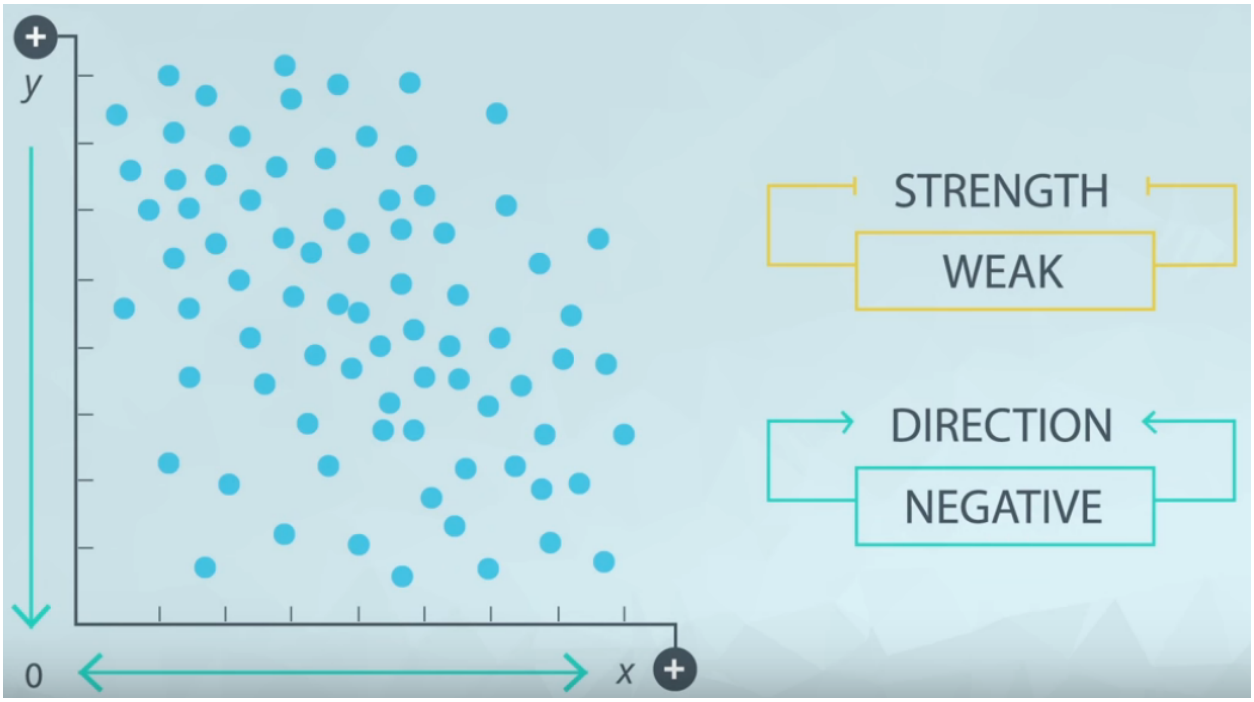

Figure 4.34:

Figure 4.35:

- Correlation
  - Varies from -1 to +1;
  - Values close to -1 and +1 are very strong;
  - The signal (positive or negative) means the direction.

Figure 6 - Example of Correlation.

4.14.2 Correlation Coefficients

This is highly field-dependent measure and these values are a general rule of thumbs.

Have in mind, in social sciences is very difficult to find a strong correlation (probably because human are very complex and hard to understand).

- Strong: $0.7 \leq |r| \leq 1.0$
- Moderate: $0.3 \leq |r| \leq .7$
- Weak: $0.0 \leq |r| \leq 0.3$

Sometimes a plot could help a lot, Figure 7 shows an example of two graphs with same correlation coefficients.

Figure 7 - Two Graphics with same Correlation Coefficients.

This problem presented in Figure 6 is part of the Anscombe's Quartet images.

4.14.3 Coefficients

A Linear Regression is a way to estimate the values of some coefficients:

- Intercept: The expected value of the response when the explanatory variable is 0 (zero);
  - b_0 : statistic value (sample)
  - β_0 : parameter value (population)

Figure 4.36:

- Slope: The expected change in the response for each 1 unit increase in the explanatory variable.
  - b_1 : statistic value (sample)
  - β_1 : parameter value (population)

Based on the Intercept and Slope, the Linear Regression equation is presented in equation (1).

$$\hat{y} = b_0 + b_1 x$$

Where: * $\hat{y}$: is the predicted value of the response from the line. * y : is an actual response value for a data point in our dataset (not a prediction from our line). * b_0 : is the intercept. * b_1 : is the slope. * x_1 : is the explanatory variable.

Figure 8 ilustrate this equation in a picture.

Figure 8 - Linear Model and Equation.

4.14.4 Least-squares

4.15 Multilinear Regression `Lesson 15`

This is a generalization of the simple Linear Model, which allows us to use several explanatory variables, as you can see an example of Multiple Linear Regression in equation (1).

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + b_4 x_4 \tag{1}$$

Figure 1 ilustrate the variables which could be used to predic the house's price.

Figure 1 - House Price Dataset.

Have in mind, in this table above there are quantitative and categorical variables.

- Quantitative:
  - Price, Area, Bedrooms, and Bathrooms;
- Categorical:
  - Neighborhood and Style.

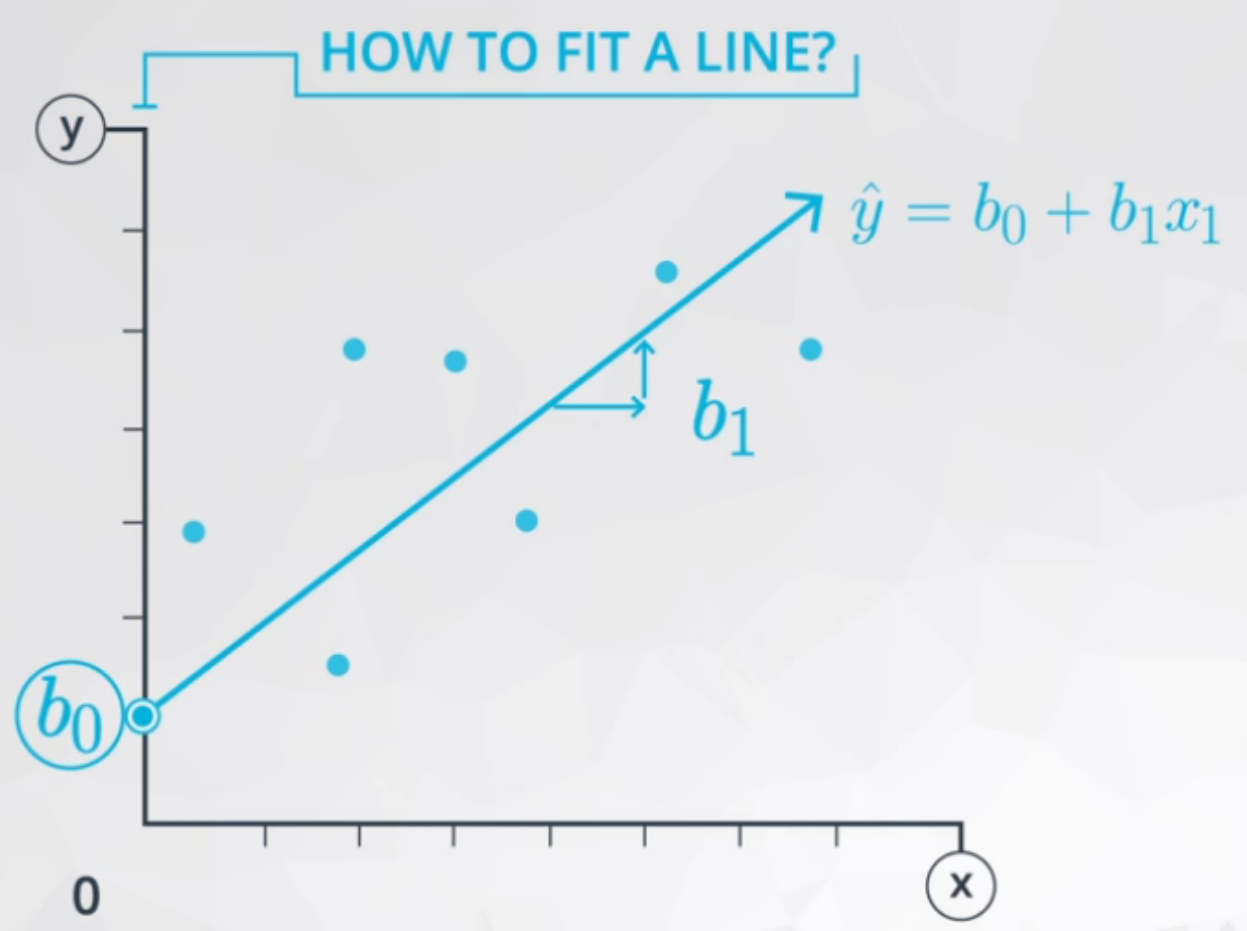

Figure 4.37:

PRICE	NEIGHBORHOOD	AREA	BEDROOMS	BATHROOMS	STYLE
$634K	A	1600	4	2	ranch
$120K	B	3200	7	3	victorian
$920K	C	1200	2	2	ranch
$124K	B	700	1	1	lodge

Figure 4.38:

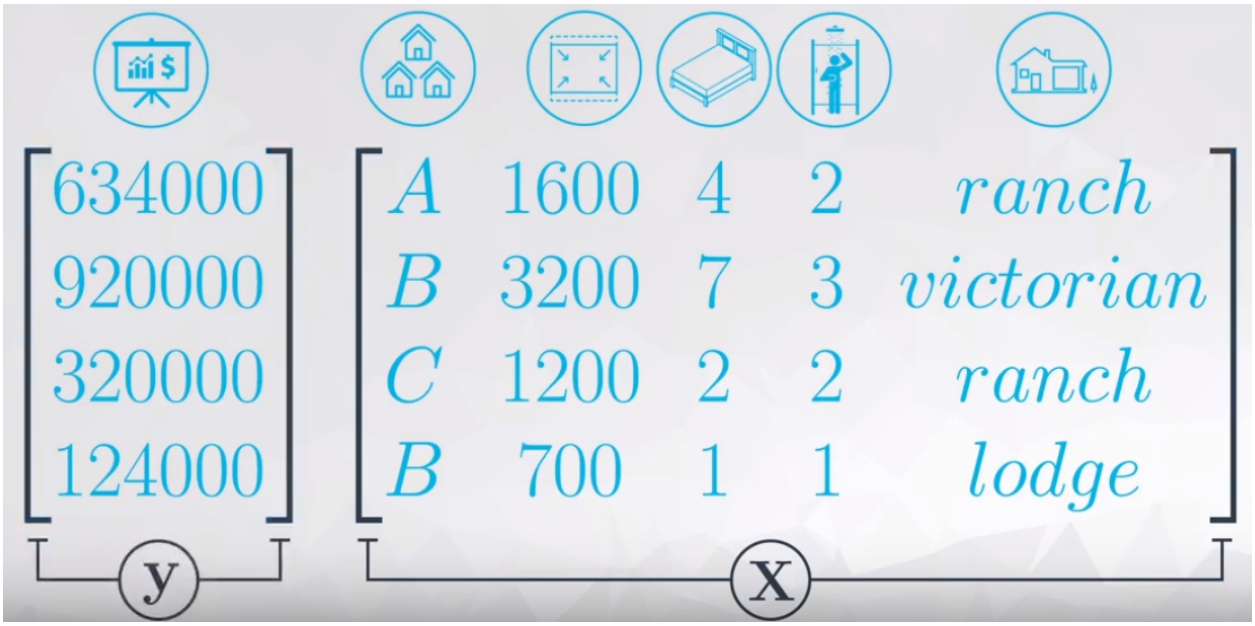

Figure 4.39:

All these variables will be used to predict the house's price, to do it so it is necessary a little of linear algebra, as you can see in the Figure 2.

Figure 2 - Matrix notation.

Where:

- **X** : Matrix of inputs;
- **y** : Vector of response that we want to predict.

After many steps of linear algebra, equation (2) resume the β calculation.

$$\beta = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}\mathbf{y} \tag{2}$$

Have in mind, the coefficients calculated in β are the same shown in Figure 3 in column `coef`.

Interpretation

Figure 3 presents outputs from the OLS of a generic dataset (house_prices.csv).

Figure 3 - Coefficients, Standard Errors, etc.

> In this video, the coefficients were all positive. Therefore, we can interpret each coefficient as the **predicted increase in the response for every one unit increase in the explanatory variable, holding all other variables in the model constant.**

This principles is the so-called *ceteris paribus*.

> *Ceteris paribus or caeteris paribus is a Latin phrase meaning "other things equal". English translations of the phrase include "all other things being equal" or "other things held constant"*

	coef	std err	t	P>|t|	[0.025	0.975]
intercept	1.007e+04	1.04e+04	0.972	0.331	-1.02e+04	3.04e+04
area	345.9110	7.227	47.863	0.000	331.743	360.079
bathrooms	7345.3917	1.43e+04	0.515	0.607	-2.06e+04	3.53e+04
bedrooms	-2925.8063	1.03e+04	-0.285	0.775	-2.3e+04	1.72e+04

Figure 4.40:

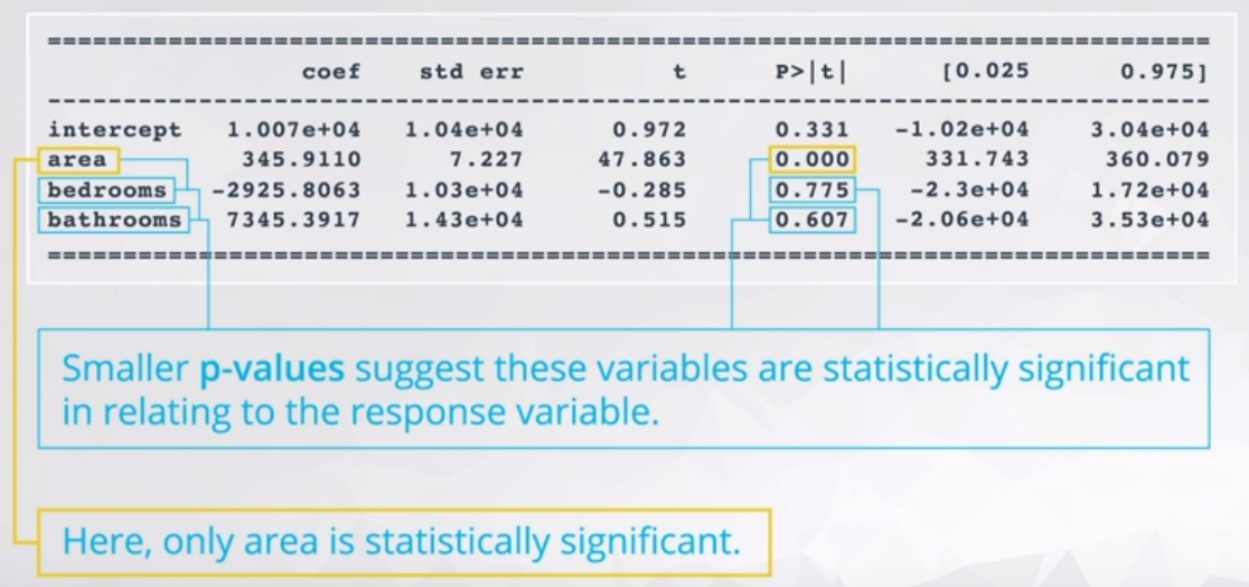

Figure 4.41:

> *or "all else unchanged". A prediction or a statement about a causal, empirical, or logical relation between two states of affairs is ceteris paribus if it is acknowledged that the prediction, although usually accurate in expected conditions, can fail or the relation can be abolished by intervening factors.* — Wikipedia

4.15.0.1 Statistical Significance

Analogous to the simple Linear Model, it is important to analyse the `p-values` of each variable. Figure 4 shows the `p-values`.

Figure 4 - Statistical Significance.

As you can see, most of the variable do not have lower `p-values`, which suggest these variables are not statistically signigicant to the response variable. In this example, only `area` reject H_0 the others variables failed to reject H_0.

Signigicant bivariate relationships are **not** always signigicant in Multiple Linear Regression.

4.15.1 Dummy Variable

This is a method to insert categorical variables in Multiple Linear Regressions. We will use 0 and 1 to encode this Dummy Variable, which will work creating new columns for each category of the variable. Figure 5 shows an example.

Figure 5 - Converting Categorical Variable to Dummy Variable.

Remember, the C columns is not necessary because is a complement of the other two columns (A and B). The dropped columns (in this case the C column) is the so-called **baseline**. What you need to understand about the **baseline** is:

> The coefficients you obtain from the output of your multiple linear regression models are then an indication of how the encoded levels compare to the baseline level (the dropped level). — Udacity Notebook

The mathematical reason to drop one columns is to garantee the $\mathbf{X}$ matrix is invertible.

Interpretation

The baseline will be used to calculate the real coefficients of each Dummy variable. Figure 6 shows an example of output of a Multiple Linear Regression.

Figure 6 - Example of Coefficients from a Dummies Variables.

The intercept is the value of the baseline (in other words it is the coefficient of A). To calculate the other coefficients you need to subtract the baseline from each coefficient.

- Coefficient A: 5.411e+05
- Coefficient B: 5.411e+05 - 5.295e+05 = 1070600.0
- Coefficient C: 5.411e+05 - 332.36 = 540767.6406

Coef C < Coef A < Coef B

Each of the coefficients provided by the OLS is a comparison with the baseline.

In a more complex example as presented in Figure 7.

Figure 7 - Dummies Variables from two categories.

There are "two" **baseline** variables: `A` and `ranch`.

- Coefficient A: -383300.0
- Coefficient B: -383300.0 + 5.229e5 = 139600.0
- Coefficient C: -383300.0 - 7168.63 = -390500.0
- ranch: -383300.0
- lodge: -383300.0 + 1.685e5 = -214800.0
- victorian: -383300.0 + 7.056e4 = -312770.0

4.15.2 Problems in Multiple Linear Regression

Problems is sensible to the application of this Multiple Linear Regression.

- What is your model for?
  - Understand how is related x and y

  - Best predict the response variable
  - Which variable is really useful in predicting your response

Founded on the objectivies of the model, there are many possibilities of problems, below there are some common problems.

- A linear relationship does not exist
- Correlated errors
- Non-constant variance
- Outliers
- Multicollinearity

Multicollinearity

Generally, we assume the explanatory variables are all uncorrelated with one another, but we want these explanatory variables are correlated with the responer variable.

An example of possible multicollinearity:

> We expected a bigger number of bedrooms and bathrooms when the house area increases.

There are two ways to identify the multicollinearity:

- Plotting the scatterplot matrix, or;
- Calculating the Variance Inflation Factors (VIFs)

Scatterplot

Figure 8 shows the scatter plot of area, bedrooms, and bathrooms.

Figure 8 - Scatterplot of area, bedrooms, and bathrooms.

All three variables have strong and positive correlation.

OK! Let's calculate the Multiple Linear Regression using `area`, `bedrooms`, and `bathrooms` as explanatory variables to predict the `price`. Figure 9 shows the output of the `OLS` method.

Figure 9 - Coefficients of a Multiple Linear Regression based on area, bedrooms, and bathrooms to predict price.

Remember, we expected positive values of coefficients, but the bedrooms coefficient is negative. This is one of the side effect of multicollinearity, it could flip the signal, and produce a weird result.

Calculating the VIF

> The Variance Inflation Factor (VIF) is a measure of colinearity among predictor variables within a multiple regression. It is calculated by taking the the ratio of the variance of all a given model's betas divide by the variane of a single beta if it were fit alone. — Ernest Tavares Website

In other words, this means to calculate for each explanatory variable the R-squared adopting it as response variable, and later calculate the VIF according to the equation (3).

$$VIF_i = \frac{1}{1 - R_i^2} \tag{3}$$

Where:

- i : intercept, area, bedrooms, and bathrooms.

This is my original equation:

$$price = intercept + area * b_1 + bedrooms * b_2 + bathrooms * b_3$$

Where:

- Explanatory variables: intercept, area, bedrooms, and bathrooms.
- Response variable: price

I need to calculate the R^2 for each of these equations:

$$intercept = area*b_1+bedrooms*b_2+bathrooms*b_3 bedrooms = intercept+area*b_1+bathrooms*b_3 bathrooms = intercept+ar$$

The result of the R^2 is presented in Table 1.

Table 1 - R-Squared used to Calculate the VIF.

Response Variable	R^2
intercept	0.8635203885901309
area	0.8167890866692036
bedrooms	0.952048682065964
bathrooms	0.9473873926707678

Table 2 shows the VIFs for each value of R^2.

Table 2 - VIFs.

Response Variable	VIF
intercept	7.3271017529266529
area	5.458190136274525
bedrooms	20.854484153608585
bathrooms	19.006851223744377

> When VIFs are greater than 10, this suggests that multicollinearity is certainly a problem in your model. Some experts even suggest VIFs of greater than 5 can be problematic. In most cases, not just one VIF is high, rather many VIFs are high, as these are measures of how related variables are with one another. — Udacity notebook.

In this case I will remove the bathrooms variable, and update my equation:

$$price = intercept + area * b_1 + bedrooms * b_2 \tag{4}$$

Following the same process, I will calculate the new R^2 and later the VIF's.

Table 3 - R-Squared used to Calculate the VIF.

Response Variable	R^2	VIF
intercept	0.8350894909730524	6.063894932472694
area	0.8129232492956318	5.345399662089865
bedrooms	0.8129232492956316	5.3453996620898625

There is an other way to calculate this VIF's using the `variance_inflation_factor` from `statsmodels.stats.outliers_inf`
It is much simpler to apply this methods, but it is quite hard to understand the hidden subject behind the scenes.

4.15.3 Higher Order Terms

Adding higher order terms is a way to add non-linearity to explain the response variable.

- Higher Order Terms:
  - Interactions: $x_1 x_2$
  - Quadratics: x^2
  - Cubics: x^3
  - Other higher orders terms: x^n

Adding these terms could improve the response results, but it makes harder to understand and explain the relationship between this higher orders terms.

For instance, y_1 do not have higher orders terms and y_2 has a interaction. Both equations are quite similar, except from the term cx_1x_2.

$$y_1 = ax_1 + bx_2 + dy_2 = ax_1 + bx_2 + cx_1x_2 + d$$

The derivate from each equation will return the slope coefficient.

$$\frac{\partial y_1}{\partial x_1} = a\frac{\partial y_2}{\partial x_1} = a + c \cdot x_2$$

In $\partial y_1/\partial x_1$ the slope is constant (a), whereas in $\partial y_2/\partial x_1$ the slope will varies according to the $c \cdot x_2$.

4.15.4 New Method

In this lesson the pandas package method called `.get_dummies` will introduced.

4.15.4.1 `.get_dummies()`

This method returns a Data Frame with each category of the categorical variable as a new column.

```
pd.get_dummies(df['categorical_variable'])
```

Remember, this method do not remove the baseline.

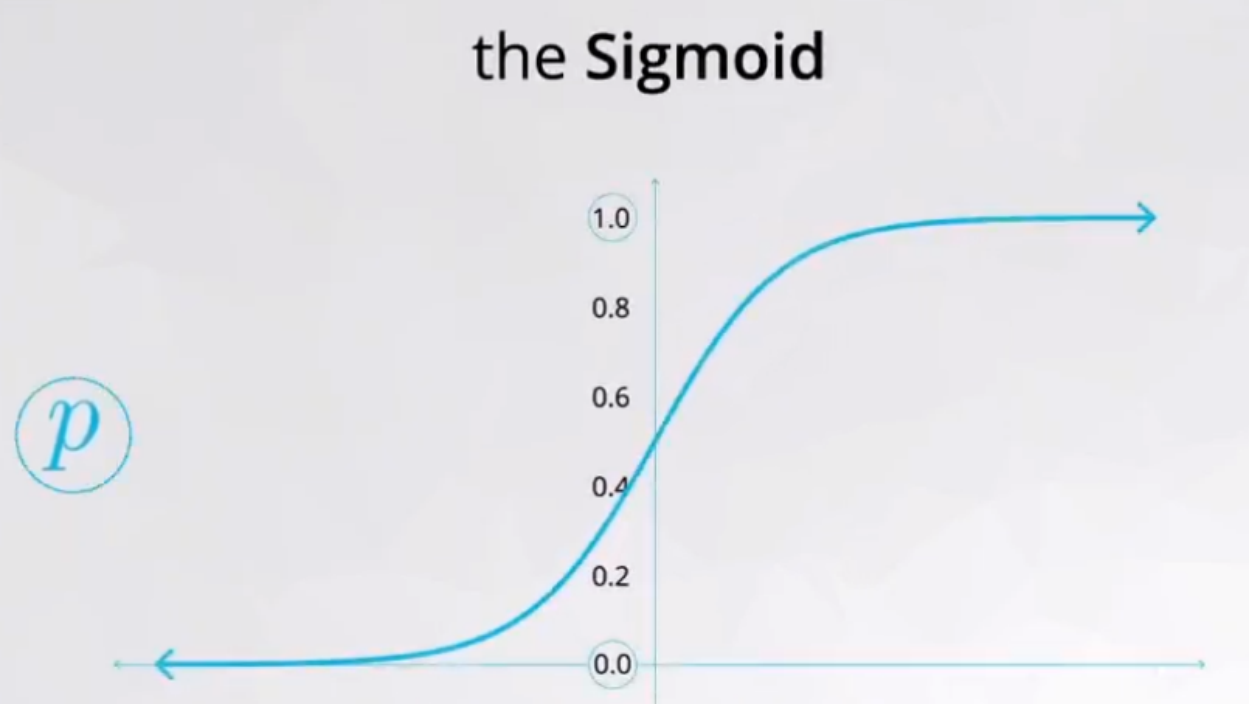

Figure 4.42:

4.16 Logistic Regression `Lesson 16`

Is used to predict a categorical response (yes or no, A or B, etc.).

Examples:

- Card transactions: Fraud or not?
- Click in the link: Yes or not?
- In a loan: default or not?

Anything else with only two outcomes. If the categorical has more than two it is classified as Multiclass Logistic Regression.

4.16.1 Sigmoid function

The sigmoid function (also known as logistic function) is used to classify the categories in two values based in your probability `p` (later it will be discussed in detail).

Figure 1 - Sigmoid function.

Observe the two extreme values, 0 and 1, these two values will be used to describe the two categories of the variable you want to predict. Figure 2 shows an example.

Figure 2 - How to interpret the categories of Transactions column.

In other words, the outcome would be:

$$y \in \{0, 1\}$$

Where:

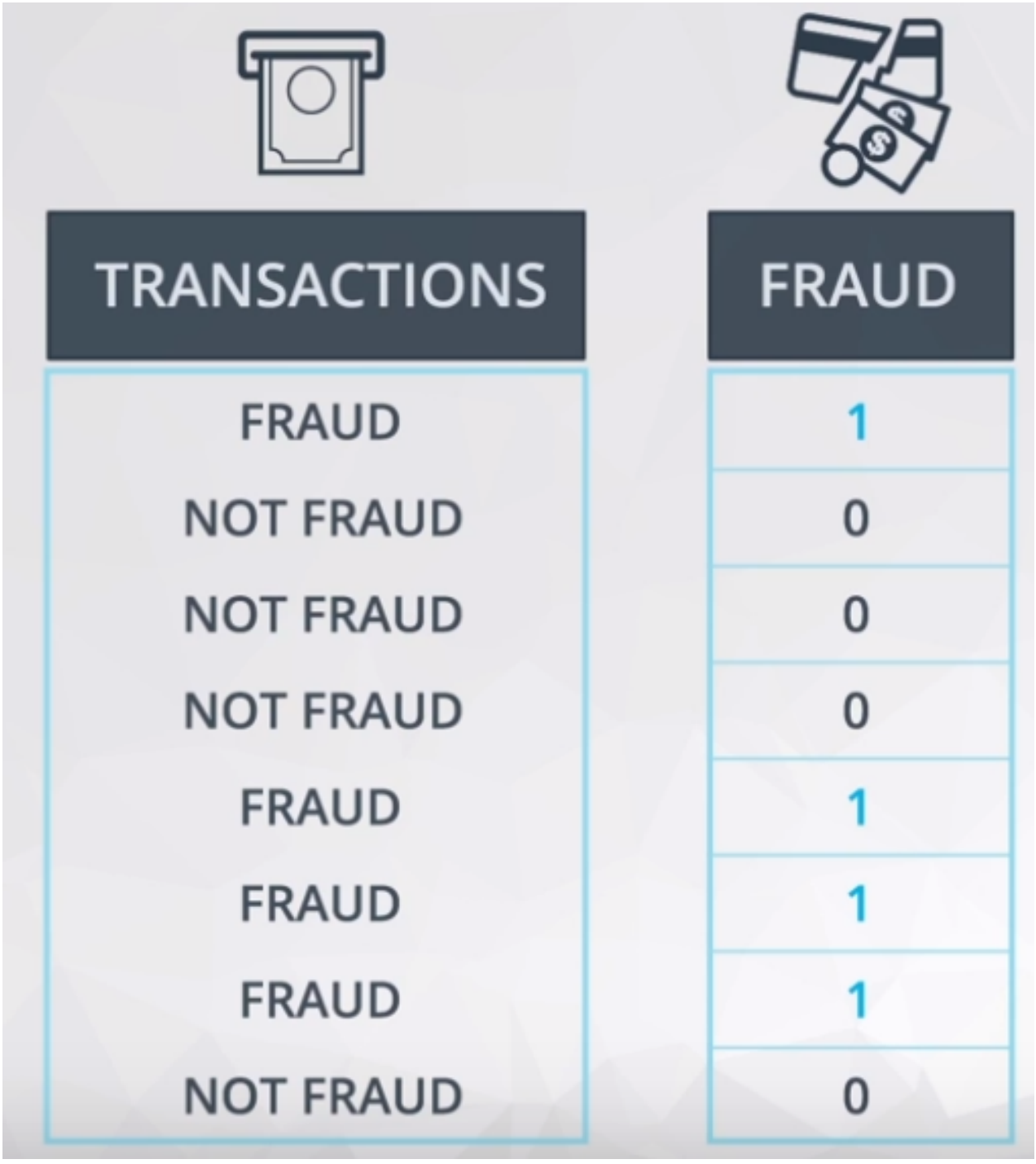
TRANSACTIONS
FRAUD
FRAUD
1
NOT FRAUD
0
NOT FRAUD
0
NOT FRAUD
0
FRAUD
1
FRAUD
1
FRAUD
1
NOT FRAUD
0

Figure 4.43:

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	9.8709	1.944	5.078	0.000	6.061	13.681
duration	-1.4637	0.290	-5.039	0.000	-2.033	-0.894
weekday	2.5465	0.904	2.816	0.005	0.774	4.319

Figure 4.44:

- 0: Positive class (Yes, Spam)
- 1: Negative class (No, Not spam)

The equation used in a regular Linear Regression is stated in equation (1).

$$y = a + b \cdot x_1 + c \cdot x_2 \tag{1}$$

Which could be noted as equation (2):

$$\mathbf{y} = \underbrace{[a\ b\ c]}_{\theta^T} \cdot \underbrace{\begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}}_{x} = \theta^T \cdot \mathbf{x} \tag{2}$$

The outcome $\mathbf{y}$ could have any real number and this is the reason to apply it to the sigmoid function, this action will delimited the outcome between 0 and 1. Equantion (3) shows how is the sigmoid function.

$$h_\theta(\mathbf{y}) = \frac{1}{1 + e^{-\mathbf{y}}} = \frac{1}{1 + e^{-\theta^T \mathbf{x}}}$$

Where:

- θ^T : Coefficients of a linear regression;
- $\mathbf{x}$:Training set.

Have in mind, the $\mathbf{y}$ will be defined as:

- $0.5 < \mathbf{y} \leq 1.0$: round to 1;
- $0.0 \leq \mathbf{y} \leq 0.5$: round to 0.

Interpretation

The way to interpret the `coef` from the `.Logit()` is quite different from the `.OLS()`. Figure 3 shows an example of outcome from the Logistic Regression.

Figure 3 - Example of Logistic Regression output.

Equally to the Multiple Linear Regression the baseline still in the intercept, but the comparison is made by "times".

Recall, in this example `weekend` is the `baseline` .

For instance:

$$e^{2.5465} = 12.76 \text{ times}$$

This means:

> On Weekdays the chance of fraud is 12.76 times more likely than on weekends. Have in mind, the weekend is our baseline.

It is not necessary to do math (add or subtract from the baseline).

$$e^{-1.4637} = 0.23 \text{ times}$$

This means:

> For each minute spent in transaction the fraud is 0.23 times more likely than on weekends.

In this case, this kind of conclusion is quite wierd, and for this reason we rephrase it.

$$1/e^{-1.4637} = 4.32 \text{ times}$$

> For each minute less spent on the transaction, the chance of fraud is 4.32 times more likely holding the day of the week constant.

4.16.2 Model Diagnostic

The model diagnostic will be performed by the Confusion Matrix. Figure 4, 5, and 6 shows examples of this kind of matrix with differents values.

Figure 4 - Confusion Matrix - Collin Powell Example.

Figure 5 - Confusion Matrix - George Bush Example.

Figure 6 - Confusion Matrix - Donald Rumsfeld - Example.

Based on these example, recall and precision could be defined as:

- **Recall:** Out of all the items that are truly positive, how many were correctly classified as positive. Or simply, how many positive items were 'recalled' from the dataset.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

- **Precision:** Out of all the items labeled as positive, how many truly belong to the positive class.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

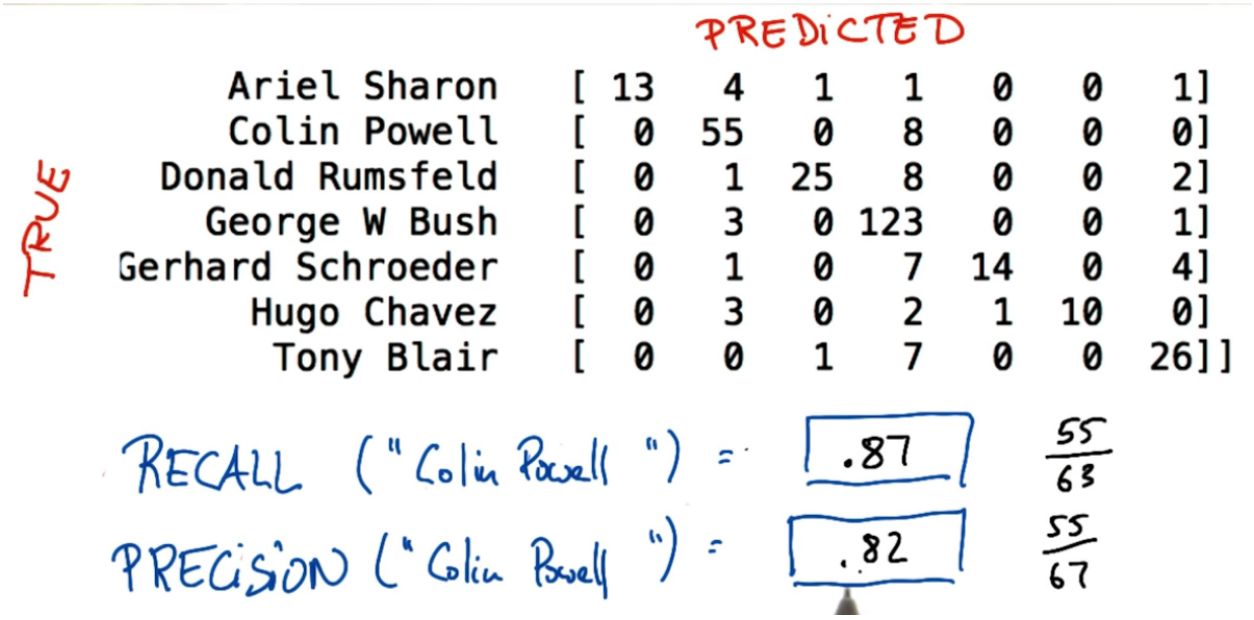

Figure 4.45:

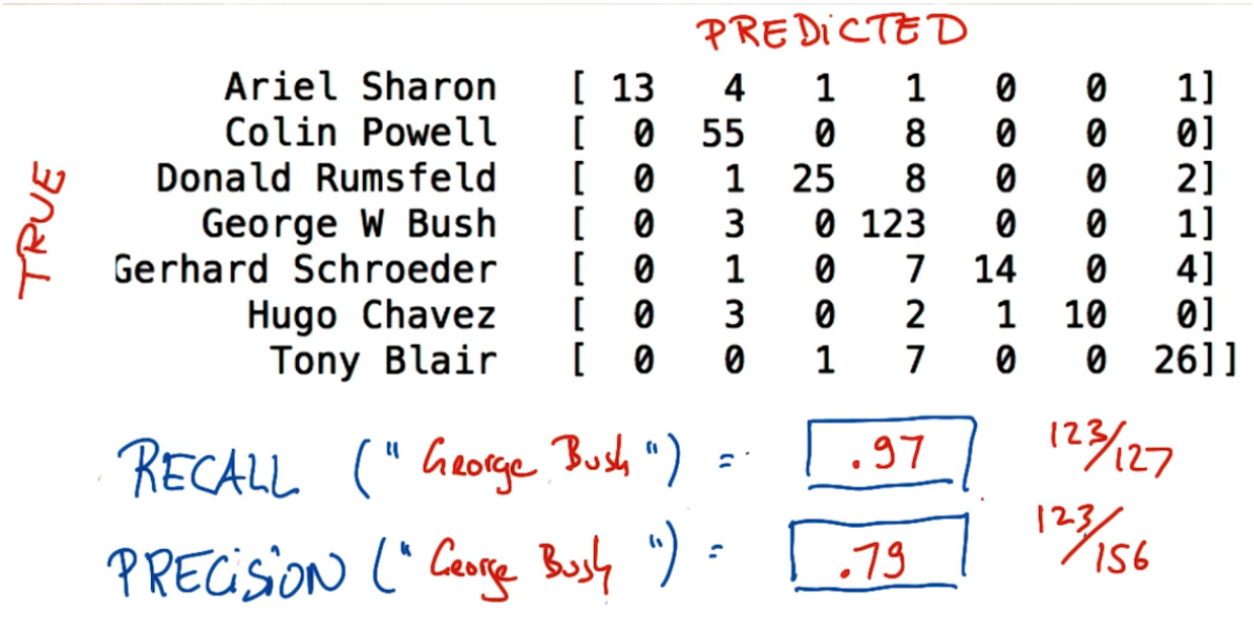

Figure 4.46:

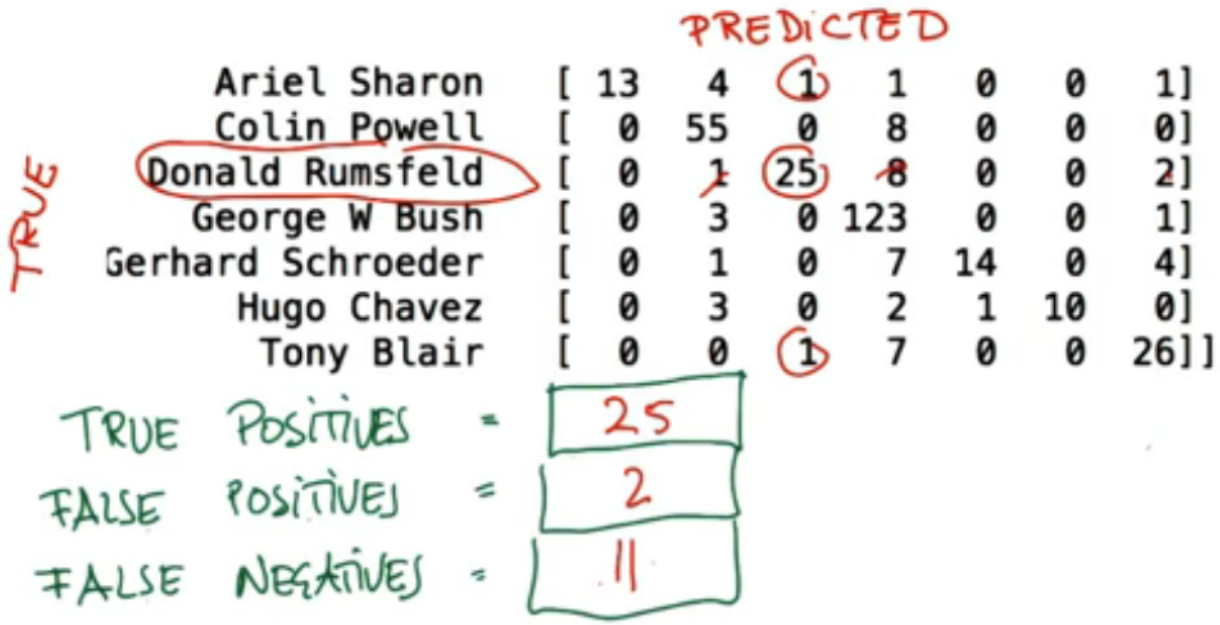

Figure 4.47:

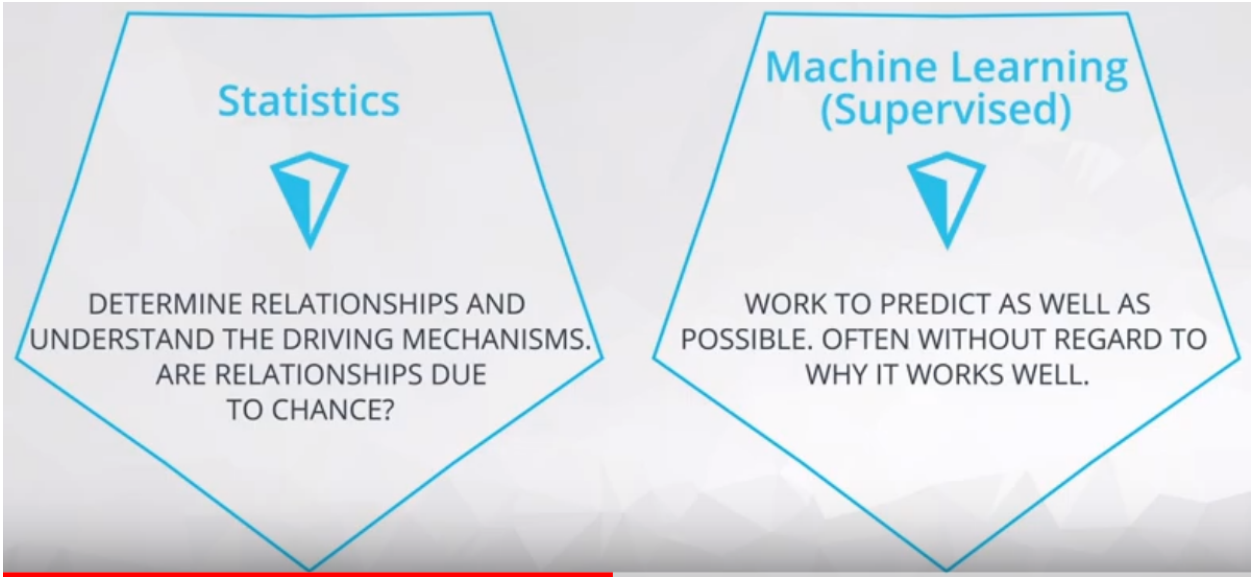

Figure 4.48:

4.16.3 Statistics vs Machine Learning

Figure 7 - Comparison between Statistics and Machine Learning.

4.16.4 New Methods

4.16.4.1 `.drop()`

This method drop a column, is a part of pandas.

```
pd.drop('column_to_be_dropped', axis = 1)
```

4.16.4.2 `.Logit()`

This method is anologous to the OLS and performs the Logistic Regression.

```
import statsmodels as sm

sm.Logit('y_variable', ['intercept','x1_variable','x2_variable'])
```

Chapter 5

Intro to Machine Learning

5.1 Naïve Bayes

Before dive in Naive Bayes. Let's talk a little bit about the thwo main groups in Machine Learning.

Supervised Learning

> **In supervised learning, we are given a data set and already know what our correct output should look like**, having the idea that there is a relationship between the input and the output.
>
> Supervised learning problems are categorized into "regression" and "classification" problems. In a regression problem, we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function. In a classification problem, we are instead trying to predict results in a discrete output. In other words, we are trying to map input variables into discrete categories.
>
> Example 1:
>
> Given data about the size of houses on the real estate market, try to predict their price. Price as a function of size is a continuous output, so this is a regression problem.
>
> We could turn this example into a classification problem by instead making our output about whether the house "sells for more or less than the asking price." Here we are classifying the houses based on price into two discrete categories.
>
> Example 2:
>
> - Regression - Given a picture of a person, we have to predict their age on the basis of the given picture
> - Classification - Given a patient with a tumor, we have to predict whether the tumor is malignant or benign. — Coursera Machine Learning Notebook

Unsupervised Learning

> Unsupervised learning allows us to approach problems with little or no idea what our results should look like. We can derive structure from data where we don't necessarily know the effect of the variables.
>
> We can derive this structure by clustering the data based on relationships among the variables in the data.

> With unsupervised learning there is no feedback based on the prediction results.
>
> Example:
>
> Clustering: Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables, such as lifespan, location, roles, and so on.
>
> Non-clustering: The "Cocktail Party Algorithm", allows you to find structure in a chaotic environment. (i.e. identifying individual voices and music from a mesh of sounds at a cocktail party). — Coursera Machine Learning Notebook

5.1.1 Gaussian - Naive Bayes

> In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features. — Wikipedia

The Naive Bayes was one of the topics covered in Advanced Statistics.

5.1.2 Scikit Learn

In this lesson we are going to use the scikit learn package to perform the Gaussian Naive Bayes.

```
# Importing the library.
from sklearn.naive_bayes import GaussianNB
```

```
GaussianNB()
```

Creating the classifier.

```
# Creating the Classifier.
clf = GaussianNB()
```

```
.fit()
```

Bear in mind, the term *fitting* or *training* will be used interchangeably along the course.

The `.fit()` method will be used to fit/train the classifier based on two inputs:

- **X:** Coordinates/features of the variable to be classified;
- **Y:** Results of already classified outputs.

Recall, this is a supervised algorithm, which means we have already some results (that I do not care its origins) and what we are aiming is a generalization of this classification using a algorithm to calculate our own coefficients based on a training data.

An example of X and Y from Scikit Learning website:

```
### Inputs.

# This is the coordinates of each point.
X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])

# This is the outcome of each classification.
Y = np.array([1, 1, 1, 2, 2, 2])
```

Plotting the X dataframe using the Y vector as categorical variable to classify each point of X.

Figure 1 - X and Y plotted.

Now, I want to train my `clf` (classifier) based on X and Y to classify any point.

```
# fit the classifier on the training features and labels
clf.fit(X, Y)
```

> What is the classification for (-0.8, -1)?

Let's take a look where is this point.

Figure 2 - New point in green.

Using the `.predict()` method it is possible to predict the classification of this point.

```
clf.predict([[-0.8, -1]])
```

> **The result is Type 1, as we expected to be.**

Have in mind, generally we use two datasets:

- Training dataset: Used to train the model;
- Test dataset: Used to test.

Training and Test datasests are complete different and it is necessary to avoid overfitting. Recall, you will use the test dataset to calculate the accuracy of your model.

5.1.2.1 `accuracy_score()`

This method is used to calculate the model accuracy.

```
accuracy_score(y_true, y_pred)
```

Where:

- `y_true`: True values;
- `y_pred`: Values predicted from the model, need to be check using the `y_true`.

5.1.3 Text Learning

This is an application of the same package Scikit Learn, but this time the problem is a bit more complicated.

Figure 3 shows, very simplified, the probability of each word given the email writer.

Figure 3 - Probability diagram of Chris and Sara.

Suppose an email like this.

$$\text{Love life!} \tag{1}$$

What is the person more likely to be written this email?

Given the probability of Chris and Sara is equal to 50%.

- $P(Chris) = P(Sara) = 0.5$

This is the *a priori* probabilities.

Based on the probabilities in Figure 3, let's calculate the *joint* probabilities.

- $P_{chris,\text{Love life}} = 0.1 \cdot 0.1 \cdot 0.5 = 0.005$
- $P_{sara,\text{Love life}} = 0.5 \cdot 0.3 \cdot 0.5 = 0.075$

A new email like this:

$$\text{Life deal} \tag{2}$$

- $P_{chris,\text{Life deal}} = 0.1 \cdot 0.8 \cdot 0.5 = 0.004$
- $P_{sara,\text{Life deal}} = 0.3 \cdot 0.2 \cdot 0.5 = 0.003$

Founded on the *joint* probabilities, we can calculate the *normalizator* (:)) of probabilities.

- For Love life!;

$$P(\text{Love life}) = P(chris) \cdot P_{chris,\text{Love life}} + P(sara) \cdot P_{sara,\text{Love life}} = P(chris) \cdot 0.08$$

- For Life deal;

$$P(\text{Life deal}) = P(chris) \cdot P_{chris,\text{Life deal}} + P(sara) \cdot P_{sara,\text{Life deal}} = 0.007$$

Finally, using *normalizator* and *joint* probabilities we can calculate the *a posteriori* probabilities.

- For Love life!;

Using the $P(\text{Love life})$ to normalize $P_{chris,\text{Love life}}$ and $P_{sara,\text{Love life}}$:

$$P(\text{Chris}|\text{"Love life"}) = P(chris) \cdot \frac{P_{chris,\text{Love life}}}{P(chris) \cdot P(\text{Love life})} = \frac{0.005}{0.080} = 0.0625$$

$$P(\text{Sara}|\text{"Love life"}) = P(sara) \cdot \frac{P_{sara,\text{Love life}}}{P(sara) \cdot P(\text{Love life})} = \frac{0.075}{0.080} = 0.9375$$

- For Life deal;

Using the $P(\text{Life deal})$ to normalize $P_{chris,\text{Life deal}}$ and $P_{sara,\text{Life deal}}$:

$$P(\text{Chris}|\text{"Life deal"}) = P(chris) \cdot \frac{P_{chris,\text{Life deal}}}{P(chris) \cdot P(\text{Life deal})} = \frac{0.004}{0.007} = 0.5714$$

$$P(\text{Sara}|\text{"Life deal"}) = P(sara) \cdot \frac{P_{sara,\text{Life deal}}}{P(sara) \cdot P(\text{Life deal})} = \frac{0.003}{0.007} = 0.4286$$

5.2 Support Vector Machine (SVM)

The Support Vector Machine was an algorithm invented by Hava Siegelmann and Vladimir Vapnik. The objective of this algorithm is to fit a line to divide a group of point/features in two or more classes.

Figure 1 show the possibilities the line to segregate the two types of points.

Figure 1 - Possible lines to divide the points in two groups.

All the three lines could be a solution, but the SMV algorithm aims to maximaze the distance to the nearest point, which is the so-called **margin**.

Have in mind, the first objective of SMV is to classify correctly, later maximize the **margin**.

Additional notebooks

For a better understanding about SVM, read the bookdown of Machine Learning offered by Stanford and taught by Andrew Ng.

This course is focused in application and do not talk much about the theoretical aspects.

5.2.1 Scikit Learn

The Scikit Learn also has methods to calculate the SVM.

5.2.1.1 `SVC()`

This function creates an object classifier.

```
clf = SVC(kernel = 'linear')
```

Remember, the `kernel` used is linar, which means the SVM will plot a straight line, and also could be lon-linear.

5.2.1.2 Non linear SVM

The non-linear SVM occurs when you adopt a non-linear kernel, as you can see in Figure 2.

Figure 2 - Example of Non-linear SVM.

There are several non-linear kernels:

- Poly;
- rbf (gives the curvy answer);
- Sigmoid, etc.

The uses of a polynomial gives the output of Figure 3.

Figure 3 - Interpretation of $z = x^2 + y^2$.

5.2.1.3 Parameters

This is characteristics that you define when you creating the classifier, so before you fitting the data. There are three:

- Kernel;
- Gamma, and;
- C.

Figure 4 shows two examples of output with different kernel.

Figure 4 - Comparison between linear (on the left side) and rbf (on the right side).

For this reason, the kernel could change completely the boundary.

```
clf = sm.SVC(C = 1, gamma = 1, kernel = "rbf")
```

5.2.1.4 C

Controls trafeoff between smooth decision boundary and classifying training points correctly.

> A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly.

5.2.1.5 Gamma

Defines how far the influence of a single training example reaches.

> gamma defines how much influence a single training example has. The larger gamma is, the closer other examples must be to be affected.

5.2.1.6 Overfitting

Avoide the overfitting, in SVM there are three parameters which could lead overfitting.

- Kernel;
- C
- gamma

5.2.1.7 `.fit()`

Similarly to the `.fit()` from GaussianNB.

```
# Fitting a model.
clf.fit(X_train,Y_train)
```

Where:

- X_train: is the "coordinates" of the points to train the data;
- Y_train: the answer of each pair of coordinates.

5.2.1.8 .pred()

Based on the `clf` it is possible to predict the values of a test dataframe.

```
# After fitting the clf you can use to predict.
clf.pred(X_test)
```

5.3 Decision Trees

> A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.
>
> Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning. — Wikipedia

In Scikit Learn you can also find the Decision Trees.

5.3.1 Scikit Learn Decision Tree

This classifier has the same workflow from the other two (GaussianNB and SVM).

```
# Import decision tree from sklearn.
from sklearn import tree

# Create the classifier using tree.
clf = tree.DecisionTreeClassifier()

# Fitting/Training the model.
clf.fit(df_train, labels_train)

# Predicting
pred = clf.predict(df_test)
```

5.3.2 Parameters

The `.DecisionTreeClassifier()` has a lot of parameters to set up.

5.3.2.1 `min_samples_split`

This parameter set a limit of how deep the decision tree will go. Figure 1 shows an example of the extension of the Tree.

Figure 1 - Example of leafs and nodes.

As you can see the `min_samples_split` will go further for each node until reach the node with two leafs.

The effect of going further and further, it means, a very low number of `min_samples_split` is more likely prone to overfitting. You can see an example of overfitting in Figure 2.

Figure 2 - Overfitting.

5.3.2.2 Entropy

Bear in mind, the default function to measure the quality of a split in `DecisionTreeClassifier()` is the `gini`index instead of the `entropy`.

The entropy controls how the decision tree decides where to split the data.

$$entropy = -\sum_{i}^{n} p_i \cdot (\ln(p_i))$$

When entropy is 1 there is no pattern, the data was spread equally on the "plot" and you can not realize where to split. On the other hand, when entropy is 0 there are a clearly areas to split.

Though, the entropy is applied to parents and child of the tree in a way that you will "compare" entropies from parents and child.

5.3.2.3 Gain

Founded on the entropies calculated from the parent and child, it is possible to calculate the gain.

$$\text{Information Gain} = entropy(parent) - [\text{weighted average}] \cdot entropy(children)$$

Figure 3 - Information Gain.

The object function of a decision tree is to maximize the information gain.

5.3.2.4 Example

Figure 4 shows a dataframe to be splited.

Figure 4 - How to split using Decision Tree?.

Bear in mind, the parent entropy of the all dataset is 1.

Now, let's use the **gain information** to decide which variable to use.

Grade

Ilustrating the decision tree using a tree, Figure 5 shows a summary.

Figure 5 - The grade splitting.

Splitting by grade flat has only fast, whereas steep has two slow and 1 gast. Let's calculate the probabilities/proportion in Table 1:

Table 1 - Table of quantity and proportion for Grade

	Steep	Flat	psteep	pflat
Slow	2	1	2/3	1/3
Fast	0	1	0/1	1/1

Based on these probabilities, I can calculate the entropy of each category (steep and flat).

- Entropy calculation:

$$entropy(flat) = -(p_{flat,slow} \cdot \ln(p_{flat,slow}) + p_{flat,fast} \cdot \ln(p_{flat,fast})) = -(1 \cdot \ln(1) + 0 \cdot \ln(0)) = 0$$

$$entropy(steep) = -(p_{steep,slow} \cdot \ln(p_{steep,slow}) + p_{steep,fast} \cdot \ln(p_{steep,fast})) = -(\frac{2}{3} \cdot \ln(\frac{2}{3}) + \frac{1}{3} \cdot \ln(\frac{1}{3})) = 0.9183$$

- Information Gain:

$$\text{Information Gain} = 1 - (0.9182 \cdot \frac{3}{4} + 0 \cdot \frac{1}{4}) = 0.3112$$

The Gain of Information is 0.3112

Table 2 - Table of quantity and proportion for Bumpiness

	Steep	Flat	psteep	pflat
Bumpy	1	1	1/2	1/2
Smooth	1	1	1/2	1/2

- Entropy calculation:

$$entropy(bumpy) = -(p_{bumpy,slow} \cdot \ln(p_{bumpy,slow}) + p_{bumpy,fast} \cdot \ln(p_{bumpy,fast})) = -(0.5 \cdot \ln(0.5) + 0.5 \cdot \ln(0.5)) = 1$$

$$entropy(smooth) = -(p_{smooth,slow} \cdot \ln(p_{smooth,slow}) + p_{smooth,fast} \cdot \ln(p_{smooth,fast})) = -(0.5 \cdot \ln(0.5) + 0.5 \cdot \ln(0.5)) = 1$$

- Information Gain:

$$\text{Information Gain} = 1 - (1 \cdot \frac{2}{4} + 1 \cdot \frac{2}{4}) = 0$$

For this variable there is no Gain of Information, which means we do not learn anything splitting by bumpiness.

Table 3 - Table of quantity and proportion for Speed Limit

	Steep	Flat	psteep	pflat
yes	2	0	2/2	0/2
no	0	2	0/2	2/2

- Entropy Calculation:

$$entropy(yes) = -(p_{yes,slow} \cdot \ln(p_{yes,slow}) + p_{yes,fast} \cdot \ln(p_{yes,fast})) = -(1 \cdot \ln(1) + 0 \cdot \ln(0)) = 0$$

$$entropy(no) = -(p_{no,slow} \cdot \ln(p_{no,slow}) + p_{no,fast} \cdot \ln(p_{no,fast})) = -(0 \cdot \ln(0) + 1 \cdot \ln(1)) = 0$$

- Information Gain:

$$\text{Information Gain} = 1 - (1 \cdot 0 + 1 \cdot 0) = 1$$

Best information gain possible. You should split using this feature.

Figure 6 - Speed Limit Decision Tree.

5.3.2.5 Bias

Bias in machine learning, means an algorithm that almost "ignores" the data, it has almost no capacity to learn anything.

On the other hand, a machine learning very perceptive to data, will only replicate what it has seen before, it has no capacity to "generalize" a new situation. It is called algorithm with high variance.

What we are looking for is something in the middle of bias and high variance. This is the trade-off of bias-variance.

5.4 Decision Trees

> A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.
>
> Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning. — Wikipedia

In Scikit Learn you can also find the Decision Trees.

5.4.1 Scikit Learn Decision Tree

This classifier has the same workflow from the other two (GaussianNB and SVM).

```
# Import decision tree from sklearn.
from sklearn import tree

# Create the classifier using tree.
clf = tree.DecisionTreeClassifier()

# Fitting/Training the model.
clf.fit(df_train, labels_train)

# Predicting
pred = clf.predict(df_test)
```

5.4.2 Parameters

The `.DecisionTreeClassifier()` has a lot of parameters to set up.

5.4.2.1 `min_samples_split`

This parameter set a limit of how deep the decision tree will go. Figure 1 shows an example of the extension of the Tree.

Figure 1 - Example of leafs and nodes.

As you can see the `min_samples_split` will go further for each node until reach the node with two leafs.

The effect of going further and further, it means, a very low number of `min_samples_split` is more likely prone to overfitting. You can see an example of overfitting in Figure 2.

Figure 2 - Overfitting.

5.4.2.2 Entropy

Bear in mind, the default function to measure the quality of a split in `DecisionTreeClassifier()` is the `gini`index instead of the `entropy`.

The entropy controls how the decision tree decides where to split the data.

$$entropy = -\sum_{i}^{n} p_i \cdot (\ln(p_i))$$

When entropy is 1 there is no pattern, the data was spread equally on the "plot" and you can not realize where to split. On the other hand, when entropy is 0 there are a clearly areas to split.

Though, the entropy is applied to parents and child of the tree in a way that you will "compare" entropies from parents and child.

5.4.2.3 Gain

Founded on the entropies calculated from the parent and child, it is possible to calculate the gain.

$$\text{Information Gain} = entropy(parent) - [\text{weighted average}] \cdot entropy(children)$$

Figure 3 - Information Gain.

The object function of a decision tree is to maximize the information gain.

5.4.2.4 Example

Figure 4 shows a dataframe to be splited.

Figure 4 - How to split using Decision Tree?.

Bear in mind, the parent entropy of the all dataset is 1.

Now, let's use the **gain information** to decide which variable to use.

Grade

Ilustrating the decision tree using a tree, Figure 5 shows a summary.

Figure 5 - The grade splitting.

Splitting by grade flat has only fast, whereas steep has two slow and 1 gast. Let's calculate the probabilities/proportion in Table 1:

Table 1 - Table of quantity and proportion for Grade

	Steep	Flat	psteep	pflat
Slow	2	1	2/3	1/3
Fast	0	1	0/1	1/1

Based on these probabilities, I can calculate the entropy of each category (steep and flat).

- Entropy calculation:

$$entropy(flat) = -(p_{flat,slow} \cdot \ln(p_{flat,slow}) + p_{flat,fast} \cdot \ln(p_{flat,fast})) = -(1 \cdot \ln(1) + 0 \cdot \ln(0)) = 0$$

$$entropy(steep) = -(p_{steep,slow} \cdot \ln(p_{steep,slow}) + p_{steep,fast} \cdot \ln(p_{steep,fast})) = -(\frac{2}{3} \cdot \ln(\frac{2}{3}) + \frac{1}{3} \cdot \ln(\frac{1}{3})) = 0.9183$$

- Information Gain:

$$\text{Information Gain} = 1 - (0.9182 \cdot \frac{3}{4} + 0 \cdot \frac{1}{4}) = 0.3112$$

The Gain of Information is 0.3112

Table 2 - Table of quantity and proportion for Bumpiness

	Steep	Flat	psteep	pflat
Bumpy	1	1	1/2	1/2
Smooth	1	1	1/2	1/2

- Entropy calculation:

$$entropy(bumpy) = -(p_{bumpy,slow} \cdot \ln(p_{bumpy,slow}) + p_{bumpy,fast} \cdot \ln(p_{bumpy,fast})) = -(0.5 \cdot \ln(0.5) + 0.5 \cdot \ln(0.5)) = 1$$

$$entropy(smooth) = -(p_{smooth,slow} \cdot \ln(p_{smooth,slow}) + p_{smooth,fast} \cdot \ln(p_{smooth,fast})) = -(0.5 \cdot \ln(0.5) + 0.5 \cdot \ln(0.5)) = 1$$

- Information Gain:

$$\text{Information Gain} = 1 - (1 \cdot \frac{2}{4} + 1 \cdot \frac{2}{4}) = 0$$

  For this variable there is no Gain of Information, which means we do not learn anything splitting by bumpiness.

Table 3 - Table of quantity and proportion for Speed Limit

	Steep	Flat	psteep	pflat
yes	2	0	2/2	0/2
no	0	2	0/2	2/2

- Entropy Calculation:

$$entropy(yes) = -(p_{yes,slow} \cdot \ln(p_{yes,slow}) + p_{yes,fast} \cdot \ln(p_{yes,fast})) = -(1 \cdot \ln(1) + 0 \cdot \ln(0)) = 0$$

$$entropy(no) = -(p_{no,slow} \cdot \ln(p_{no,slow}) + p_{no,fast} \cdot \ln(p_{no,fast})) = -(0 \cdot \ln(0) + 1 \cdot \ln(1)) = 0$$

- Information Gain:

$$\text{Information Gain} = 1 - (1 \cdot 0 + 1 \cdot 0) = 1$$

  Best information gain possible. You should split using this feature.

Figure 6 - Speed Limit Decision Tree.

5.4.2.5 Bias

Bias in machine learning, means an algorithm that almost "ignores" the data, it has almost no capacity to learn anything.

On the other hand, a machine learning very perceptive to data, will only replicate what it has seen before, it has no capacity to "generalize" a new situation. It is called algorithm with high variance.

What we are looking for is something in the middle of bias and high variance. This is the trade-off of bias-variance.

Chapter 6

Data Visualization (Optional)

We have finished a nice book.