



**SIX WEEKS SUMMER TRAINING  
REPORT**

On

**Basic of Data Science - LPU Summer Training Program**

Submitted by

Divyanshu

Singh

Registration No. 12010303

Program Name: Bachelor of Technology in Computer Science

Under the Guidance of

**Mr. Amit Mahensaria**

School of Computer Science & Engineering

Lovely professional University, Phagwara

(May-July, 2022)

## DECLARATION

I hereby declare that I have completed my six weeks summer training at Basic of Data Science - LPU Summer Training Program from 25<sup>th</sup> May 2022 to 10<sup>th</sup> July 2022 under the guidance of Mr. Amit Mahensaria. I have declare that I have worked with full dedication during these six weeks of training and my learning outcomes fulfil the requirements of training for the award of degree of Bachelor of Technology, Lovely Professional university, Phagwara.

(Signature of student)

Name of Student: Divyanshu

Singh

Registration no: 12010303

Date: 24th Sept, 2022

## **Acknowledgement**

It is with sense of gratitude; I acknowledge the efforts of entire hosts of well-wishers who have in some way or other contributed in their own special ways to the success and completion of the Summer Training.

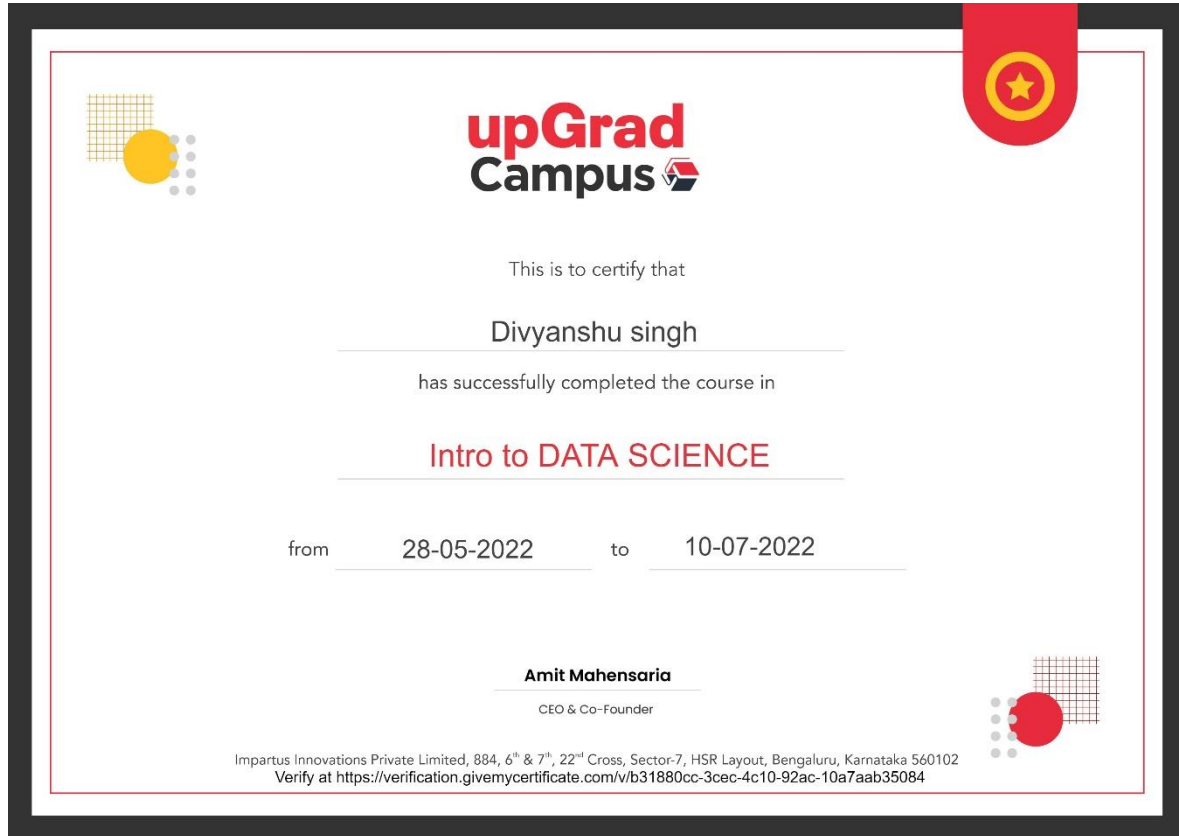
Successfully completion of any type technology requires helps from a number of people. I have also taken help from different people for the preparation of the report. Now, there is little efforts to show my deep gratitude to those helpful people.

I would like to also thank my own college Lovely Professional University for offering such a course which not only improve my programming skill but also taught me other new technology.

Then I would like to thank my parents and friends who have helped me with their valuable suggestions and guidance for choosing this course.

Last but not least I would like to thank my all classmates who have helped me a lot.

## Training certificate from organization



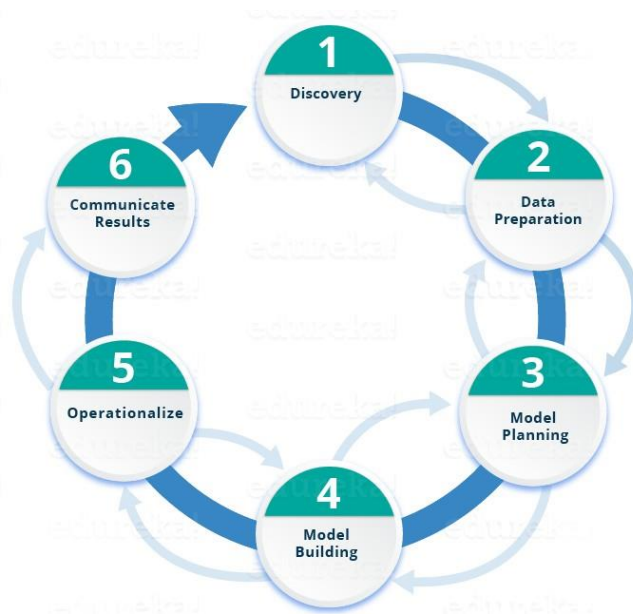
## Table Of Contents

1. Introduction
2. Technology Learnt
3. Reason for choosing this technology.
4. Profile of the Problem
5. Existing System
6. Problem Analysis
  - Product definition
  - Feasibility Analysis
7. Software Requirement Analysis
8. Design
  - Tables and their relationships
  - Flowcharts/Pseudo code
9. Implementation
10. Learning Outcome from training/technology learnt
11. Gantt chart
12. Project Legacy
  - Technical and Managerial learnt.
13. Bibliography

## Introduction

Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data.

In that field work in a variety of fields. Each is crucial to finding solutions to problems and requires specific knowledge. These fields include data acquisition, preparation, mining and modelling, and model maintenance. Data scientists take raw data, turn it into a goldmine of information with the help of machine learning algorithms that answer questions for businesses seeking solutions to their queries.



**Data Acquisition:** Here, data scientists take data from all its raw sources, such as databases and flat-files. Then, they integrate and transform it into a homogenous format, collecting it into what is known as a “data warehouse,” a system by which the data can be used to extract information from easily. Also known as ETL, this step can be done with some tools, such as Talend Studio,

DataStage and Informatica.

**Data Preparation:** This is the most important stage, wherein 60 percent of a data scientist’s time is spent because often data is “dirty” or unfit for use and must be scalable, productive and meaningful. In fact, five sub-steps exist here:

**Data Cleaning:** Important because bad data can lead to bad models, this step handles missing values and null or void values that might cause the models to fail. Ultimately, it improves business decisions and productivity.

**Data Transformation:** Takes raw data and turns it into desired outputs by normalizing it. This step can use, for example, min-max normalization or z-score normalization.

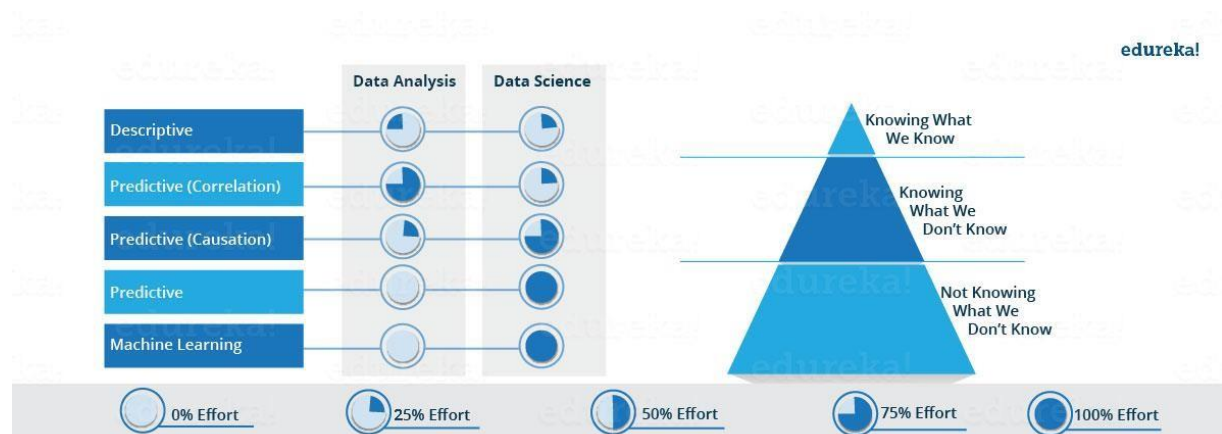
**Handling Outliers:** This happens when some data falls outside the scope of the realm of the rest of the data. Using exploratory analysis, a data scientist quickly uses plots and graphs to determine what to do with the outliers and see why they're there. Often, outliers are used for fraud detection.

**Data Integration:** Here, the data scientist ensures the data is accurate and reliable.

**Data Reduction:** This compiles multiple sources of data into one, increases storage capabilities, reduces costs and eliminates duplicate, redundant data.

**Data Mining:** Here, data scientists uncover the data patterns and relationships to take better business decisions. It's a discovery process to get hidden and useful knowledge, commonly known as exploratory data analysis. Data mining is useful for predicting future trends, recognizing customer patterns, helping to make decisions, quickly detecting fraud and choosing the correct algorithms. Tableau works nicely for data mining.

**Model Building:** This goes further than simple data mining and requires building a machine learning model. The model is built by selecting a machine learning algorithm that suits the data, problem statement and available resources.

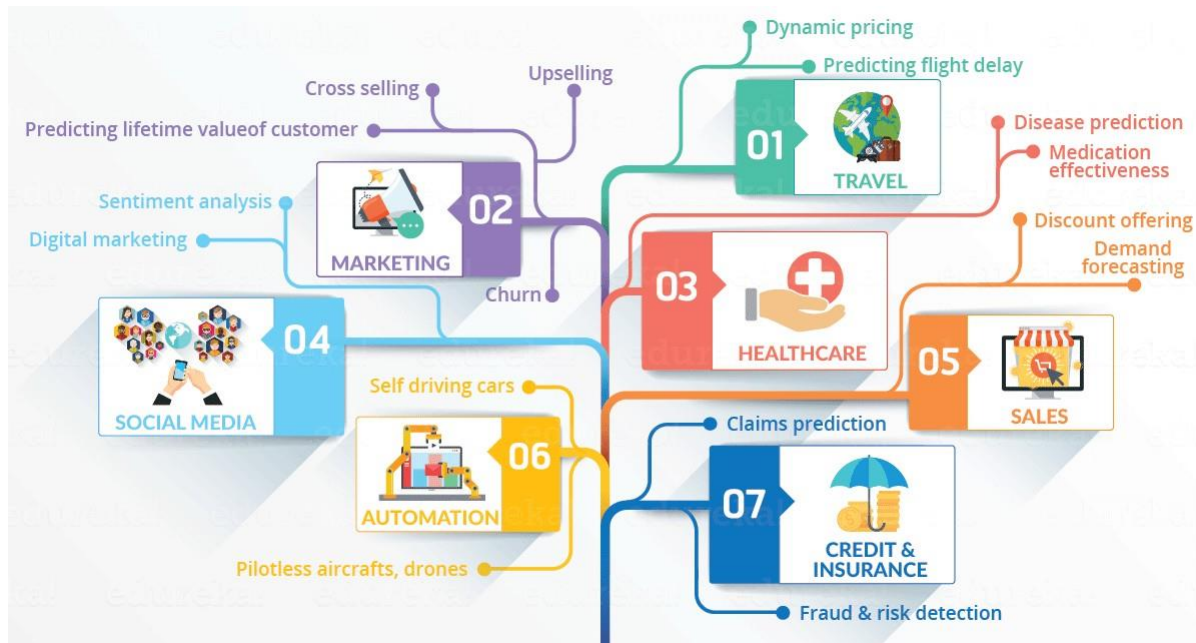


Traditionally, the data that we had was mostly structured and small in size, which could be analysed by using simple BI tools. Unlike data in the traditional systems which were mostly structured, today most of the data is unstructured or semi-structured. Let's have a look at the data trends in the image given below which shows that by 2020, more than 80 % of the data will be unstructured.

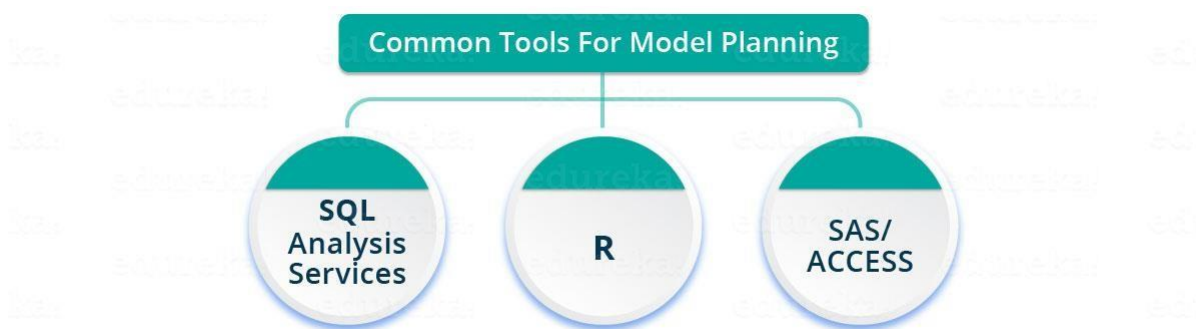
This data is generated from different sources like financial logs, text files, multimedia forms, sensors, and instruments. Simple BI tools are not capable of processing this huge

volume and variety of data. This is why we need more complex and advanced analytical tools and algorithms for processing, analysing and drawing meaningful insights out of it.

Let's have a look at the below infographic to see all the domains where Data Science is creating its impression.



Let's have a look at various model planning tools.



1. **R** has a complete set of modelling capabilities and provides a good environment for building interpretive models.
2. **SQL** Analysis services can perform in-database analytics using common data mining functions and basic predictive models.
3. **SAS/ACCESS** can be used to access data from Hadoop and is used for creating repeatable and reusable model flow diagrams.



## Technology Learnt

### 1. Data Analysis in Excel

1. Introduction
2. Understanding the Excel Interface
3. Slicing and Dicing Data - Sort and Filter
4. Basic Formatting, Conditional Formatting, Advanced Formatting
5. Printing and Page Layout
6. Passwords and Naming Files
7. Delimited Files
8. Discovering Shortcuts
9. Introduction to Formulae
10. Complex Functions
11. Cell Referencing and Text Functions
12. Logical Formulae
13. Anand's Anecdotes
14. Creating and Formatting Charts
15. Types of Charts
16. Creating a Pivot Table
17. Analysing Data in a Pivot Table
18. Filtering Data in a Pivot Table
19. Anand's Anecdotes - Pivot Tables
20. VLOOKUP - Linking Data from multiple files & tables
21. Anand's Anecdotes – VLOOKUP
22. Common Errors in Excel

### 2. Fundamentals of Python and Libraries

1. Python
  - Data Structures in Python
    - Lists
    - Tuples
    - Dictionaries
    - Sets

- Control Structures
  - If-Elif-Else
  - Loops
  - Comprehensions
- Functions
  - Lambda, Map, Filter, and Reduce
  - Map
  - Reduce
  - Filter

## 2. NumPy

- NumPy Basics
- Creating NumPy Arrays
- Structure and Content of Arrays
- Subset, Slice, Index and Iterate through Arrays
- Multidimensional Arrays
- Computation Times in NumPy and Standard Python Lists
- Basic Operations
- Operations on Arrays
- Basic Linear Algebra Operations

## 3. Pandas – Python for Data Science

- Pandas Basics
- Indexing and Selecting Data
- Merge and Append
- Grouping and Summarizing Dataframes
- Lambda function & Pivot tables

## 3. Data Visualization Using Tableau & SQL

### 1. Data Analysis using SQL

- Basics of SQL
- SQL Clauses
- Adding and Deleting Columns
- Changing Column Name and Data Type

- Changing Constraints (Primary key), Changing Constraints (Foreign key)
- String Manipulation, Date Manipulation
- Introduction to Windowing Functions
- Frames
- Named Windows
- Introduction To User-Defined Functions
- User-Defined Functions (Application)
- Introduction To Stored Procedures
- Stored Procedures (Application)
- Optimisation in Select Clause, Where Clause, Group by and Order by, Joins, Window Function
- Advanced SQL
  - Defining Data Warehouse
  - Structure of Data Warehouse
  - OLAP vs. OLTP
  - Star Schema

## 2. Visualisation using Tableau

- Histograms
- Area Maps
- Data Formats and Tableau Interface
- Connecting to the Data
- Data Preparation in Tableau
- Hierarchies and Drill Down
- Hierarchies and Drill Down
- Scatter Plots and Pie Charts
- Tree Maps
- Dual Axes Charts
- Box Plots
- Calculations in Tableau
- Dashboard and Stories

## 4. Statistics for Data Science

1. Exploratory Data Analysis

- Data Sourcing
- Univariate Analysis
- Segmented Univariate
- Bivariate Analysis
- Derived Metrics

2. Inferential Statistics

- Basics of Probability
- Discrete Probability Distributions
- Continuous Probability Distributions
- Central Limit Theorem
- Applications of Sampling Methods
- 

3. Hypothesis Testing

- Null and Alternate Hypotheses
- Critical Value Method
- Making a Decision
- p-value Method
- Types of Errors
- T Distribution
- Two-Sample Mean Test
- Two-Sample Proportion Test
- A/B Testing Demonstration
- Hypothesis testing in Python
- Z-test
- T-Test
- Chi-Square Test
- F-Test

## Reason for choosing this technology

Data science is a relatively new and emerging field. As it's not a well-known study area

Data science is the study of information – where it comes from, what it tells us and how to convert it into a useful resource which can help businesses make decisions, solve complex problems and create strategies to improve results and performance. It's an interdisciplinary field which mixes technical ability with business insights to drive change based on intelligence.

**Future-Oriented Role:** Data is the driving force behind industries in the 21st Century. Anyone planning to develop their knowledge of the data science field is placing themselves in a strong position for a successful future career.

Forward-looking enterprises who understand that data fuels the future are hiring data scientists now. It's the career of tomorrow and promises to deliver a future-proofed career for anyone who equips themselves with knowledge of technologies such as machine learning and artificial intelligence which will help them become a truly valuable asset to their employer.

As a data scientist stand a chance to contribute to making the world a better place for the present and future generations to live in. Not only will you be earning as a professional, but you will also be contributing to social wellbeing in the philanthropy and non-profit world.

For instance, there is a high demand for data scientists to help curb the refugee crisis, which has led to the death and displacement of many people. Using skills and expertise, you will be helping governments and non-profit organizations make better decision in providing aid to the most vulnerable societies based on scientific data.

# Excel

Excel is a spreadsheet program from Microsoft and a component of its Office product group for business applications. Microsoft Excel enables users to format, organize and calculate data in a spreadsheet.

Excel is a part of the Microsoft Office and Office 365 suites and is compatible with other applications in the Office suite. The spreadsheet software is available for Windows, macOS, Android and iOS platforms.

Excel is most commonly used in business settings. For example, it is used in business analysis, human resource management, operations management and performance reporting. Excel uses a large collection of cells formatted to organize and manipulate data and solve mathematical functions. Users can arrange data in the spreadsheet using graphing tools, pivot tables and formulas. The spreadsheet application also has a macro programming language called Visual Basic for Applications.

Organizations use Microsoft Excel for the following:

- collection and verification of business data;
- business analysis;
- data entry and storage;
- data analysis;
- performance reporting;
- strategic analysis;
- accounting and budgeting;
- administrative and managerial management;
- account management;
- project management; and
- office administration.


## Excel and XLS files

An XLS file is a spreadsheet file that can be created by Excel or other spreadsheet programs. The file type represents an Excel Binary File format. An XLS file stores data as binary streams - a compound file. Streams and substreams in the file contain information about the content and structure of an Excel workbook.

Versions of Excel after Excel 2007 use XLSX files by default, since it is a more open and structured format.

## Excel competitors

At a glance: Google Sheets vs. Microsoft Excel		
	Google Sheets	Microsoft Excel
PRICE	Free	Requires Office 365 subscription
TYPE OF APPLICATION	Cloud-based	Full-featured application is not cloud-based
COLLABORATION	Preferred for collaboration	Less favorable for collaboration
DATA PROCESSING	Weaker; storage is limited to 5 million cells	Stronger; storage is limited to 17 million cells
FEATURES	Basic spreadsheet features	Larger offering of advanced features
INTEGRATION	Integrates with Google apps and Microsoft files	Integrates with Microsoft apps
SUPPORT	Help articles and an interactive community	Community help forum and an Excel learning hub

©2021 TECHTARGET. ALL RIGHTS RESERVED. 

# Excel Functions and Formulas

In Microsoft Excel, a formula is an expression that operates on values in a range of cells. These formulas return a result, even when it is an error. Excel formulas enable you to perform calculations such as addition, subtraction, multiplication, and division. In addition to these, you can find out averages and [calculate percentages in excel](#) for a range of cells, manipulate date and time values, and do a lot more

## Formulas in Excel: an overview

- Choose a cell.
- To enter an equal sign, click the cell and type =.
- Enter the address of a cell in the selected cell or select a cell from the list.
- You need to enter an operator.
- Enter the address of the next cell in the selected cell.
- Press Enter.

There is another term that is very familiar to Excel formulas, and that is "[function](#)". The two words, "formulas" and "functions" are sometimes interchangeable. They are closely related, but yet different. A formula begins with an equal sign. Meanwhile, functions are used to perform complex calculations that cannot be done manually. Functions in excel have names that reflect their intended use.

Excel formulas and functions help you perform your tasks efficiently, and it's time-saving.



## Excel Formulas and Functions Examples

There are plenty of Excel formulas and functions depending on what kind of operation you want to perform on the dataset. We will look into the formulas and functions on mathematical operations, character-text functions, data and time, sumif-countif, and few lookup functions.

### 1. SUM

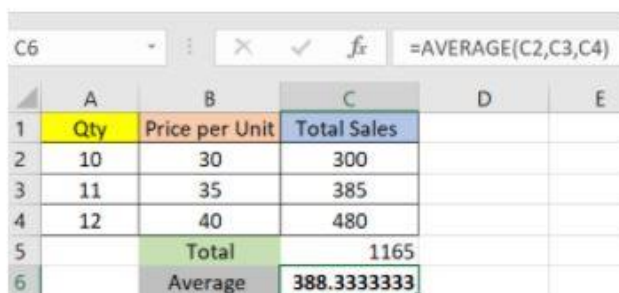
The SUM() function, as the name suggests, gives the total of the selected range of cell values. It performs the mathematical operation which is addition. Here's an example of it below:



	A	B	C	D
1	Qty	Price per Unit	Total Sales	
2	10	30	300	
3	11	35	385	
4	12	40	480	
5		Total	1165	

### 2. AVERAGE

The AVERAGE() function focuses on calculating the average of the selected range of cell values. As seen from the below example, to find the avg of the total sales, you have to simply type in “AVERAGE(C2, C3, C4)”.



	A	B	C	D	E
1	Qty	Price per Unit	Total Sales		
2	10	30	300		
3	11	35	385		
4	12	40	480		
5		Total	1165		
6		Average	388.3333333		

### 3. COUNT

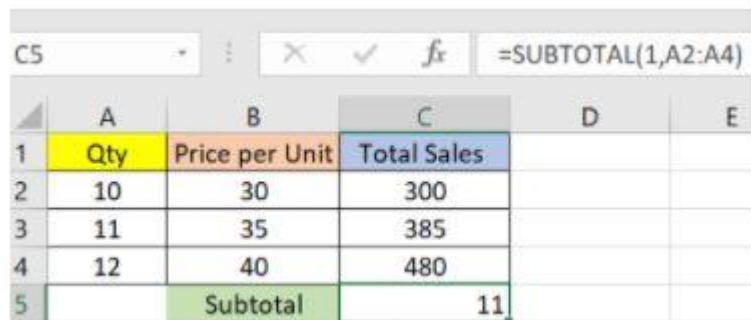
The function **COUNT()** counts the total number of cells in a range that contains a number. It does not include the cell, which is blank, and the ones that hold data in any other format apart from numeric.



	A	B	C	D
1	Qty	Price per Unit	Total Sales	
2	10	30	300	
3	11	35	385	
4	12	40	480	
5		Count	3	

### 4. SUBTOTAL

Moving ahead, let's now understand how the subtotal function works. The **SUBTOTAL()** function returns the subtotal in a database. Depending on what you want, you can select either average, count, sum, min, max, min, and others. Let's have a look at two such examples.



	A	B	C	D	E
1	Qty	Price per Unit	Total Sales		
2	10	30	300		
3	11	35	385		
4	12	40	480		
5		Subtotal	11		

### 5. MODULUS

The **MOD()** function works on returning the remainder when a particular number is divided by a divisor. Let's now have a look at the examples below for better understanding.

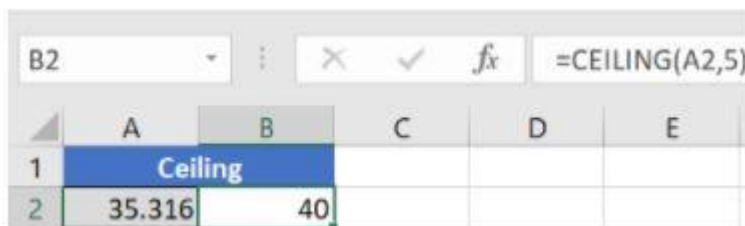
- In this example, we have divided 10 by 3. The remainder is calculated using the function “=MOD(A2,3)”. The result is stored in B2. We can also directly type “=MOD(10,3)” as it will give the same answer.

## 6. POWER

The function “Power()” returns the result of a number raised to a certain power. Let’s have a look at the examples shown below:

## 7. CEILING

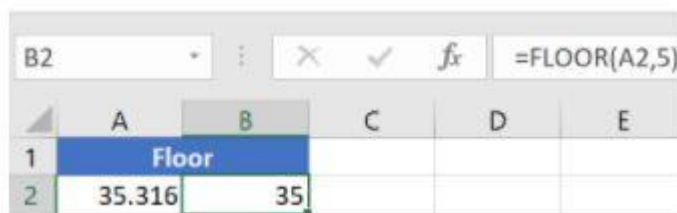
Next, we have the ceiling function. The CEILING() function rounds a number up to its nearest multiple of significance.



	A	B	C	D	E
1	Ceiling				
2	35.316	40			

## 8. FLOOR

Contrary to the Ceiling function, the floor function rounds a number down to the nearest multiple of significance.



	A	B	C	D	E
1	Floor				
2	35.316	35			

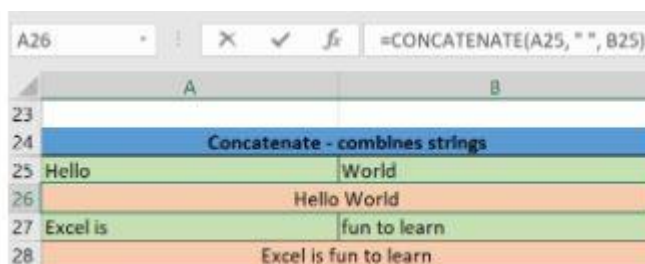
## 9. LEN

The function LEN() returns the total number of characters in a string. So, it will count the overall characters, including spaces and special characters. Given below is an example of the Len function.

## 10. CONCATENATE

This function merges or joins several text strings into one text string. Given below are the different ways to perform this function.

- In this example, we have operated with the syntax `=CONCATENATE(A25, " ", B25)`

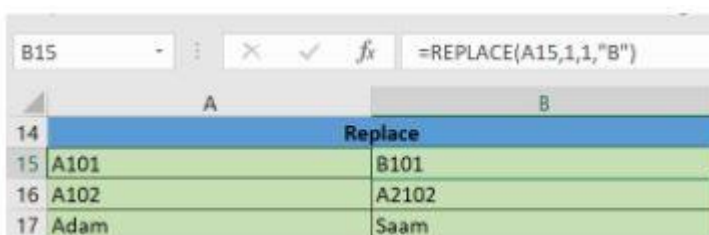


	A	B
23		
24	Concatenate - combines strings	
25	Hello	World
26	Hello World	
27	Excel is	fun to learn
28	Excel is fun to learn	

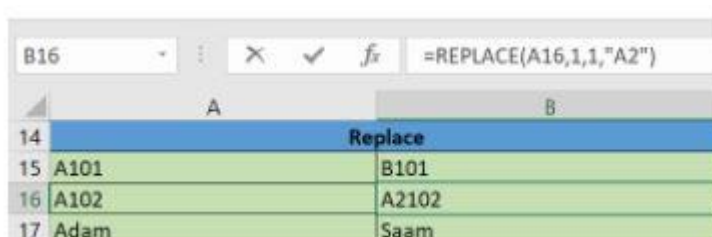
## 11. REPLACE

As the name suggests, the REPLACE() function works on replacing the part of a text string with a different text string.

The syntax is “=REPLACE(old\_text, start\_num, num\_chars, new\_text)”. Here, start\_num refers to the index position you want to start replacing the characters with. Next, num\_chars indicate the number of characters you want to replace.



	A	B
14	Replace	
15	A101	B101
16	A102	A2102
17	Adam	Saam



	A	B
14	Replace	
15	A101	B101
16	A102	A2102
17	Adam	Saam

## More function

## Excel String (Text) Functions

- **FIND Function:** Returns the starting position of a text string in another text string (case sensitive).
- **LEFT Function:** Returns the string from another string starting from the left.
- **LEN Function:** Counts the number of characters from the value supplied.
- **LOWER Function:** Converts a text into lower case.
- **MID Function:** Returns a substring from a string using a specific position and number of characters.
- **PROPER Function:** Convert a text to a proper case text.
- **REPT Function:** Repeats a value several times.
- **RIGHT Function:** Returns the string from another string starting from the right.
- **SEARCH Function:** Returns the starting position of a text string in another text string (case sensitive).
- **UPPER Function:** Convert a text into an upper case text.

## Excel Date Functions

- **DATE Function:** Returns a valid date using the day, month, and year supplied.
- **DATEDIF Function:** Calculates the difference between two dates.
- **DATEVALUE Function:** Converts a date that is formatted as text into an actual date.
- **DAY Function:** Returns the day from the date supplied.
- **DAYS Function:** Returns the count of days between two dates.
- **EDATE Function:** Returns a date after adding/subtracting months from the supplied date.
- **EOMONTH Function:** Returns the end of the month date from a future month or a past month.
- **MONTH Function:** Returns the month from the date supplied.
- **NETWORKDAYS Function:** Count of days between the start date and end date, excluding weekends and holidays.
- **NETWORKDAYS.INTL Function:** Count of days between the start date and end date, excluding weekends (Custom), and holidays.

## **Excel Time Functions**

- **HOUR Function:** *Returns the hours from the time supplied.*
- **MINUTE Function:** *Returns the minutes from the time supplied.*
- **NOW Function:** *Returns the current date and time.*
- **SECOND Function:** *Returns the seconds from the time supplied.*
- **TIME Function:** *Returns a valid time using the hours, minutes, and seconds supplied.*
- **TIMEVALUE Function:** *Convert a time value that is stored as text into actual time.*

## **Excel Math Functions**

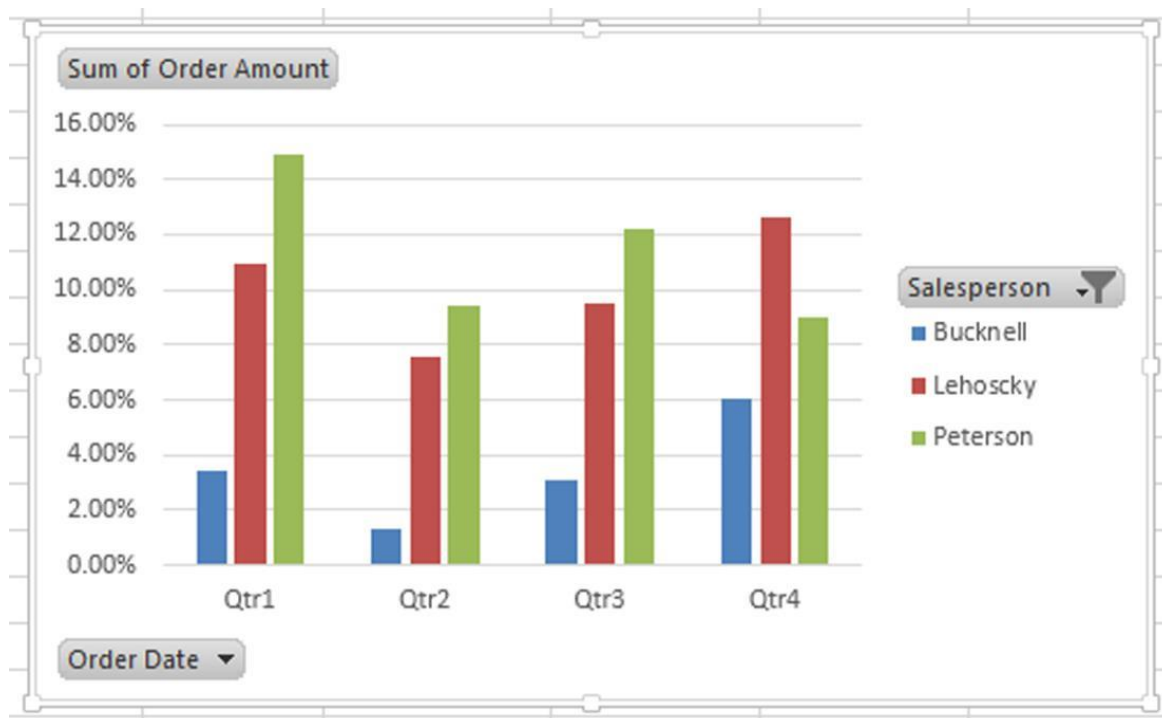
- **ABS Function:** *Converts a number into an absolute number.*
- **EVEN Function:** *Rounds a number to the nearest even number.*
- **INT Function:** *Returns the integer part from the value supplied.*
- **MOD Function:** *Returns the remainder value after dividing a number with a divisor.*
- **MROUND Function:** *Rounds a number to a given multiple.*
- **RAND Function:** *Returns a random number ranging from 0 to 1.*
- **SUM Function:** *Sum the value supplied.*
- **SUMIF Function:** *Sum the value supplied using the condition specified.*
- **SUMIFS Function:** *Sum the value supplied using the multiple conditions specified.*
- **SUMPRODUCT Function:** *Multiply and sum the array values.*
- **TRUNC Function:** *Returns a number after truncating the original number.*

# PivotTable

A PivotTable is an interactive way to quickly summarize large amounts of data. You can use a PivotTable to analyze numerical data in detail, and answer unanticipated questions about your data. A PivotTable is especially designed for: Querying large amounts of data in many user-friendly ways.

## Create a PivotTable in Excel for Windows

1. Select the cells you want to create a PivotTable from. ...
2. Select Insert > PivotTable.
3. This will create a PivotTable based on an existing table or range. ...
4. Choose where you want the PivotTable report to be placed. ...
5. Click OK.

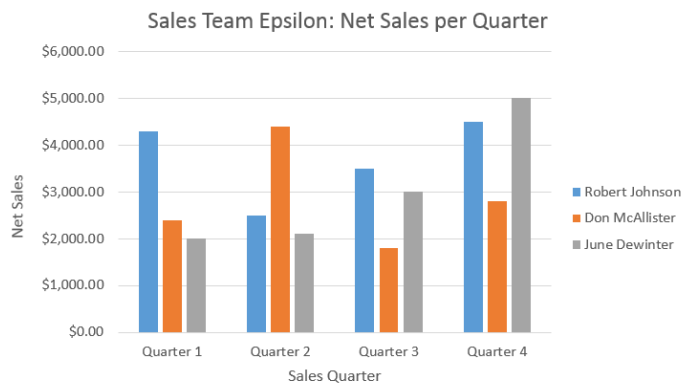


A PivotTable is a powerful tool to calculate, summarize, and analyze data that lets you see comparisons, patterns, and trends in your data. PivotTables work a little bit differently depending on what platform you are using to run Excel.

# Understanding charts

Excel has several different **types of charts**, allowing you to choose the one that best fits your data. In order to use charts effectively, you'll need to understand how different charts are used.

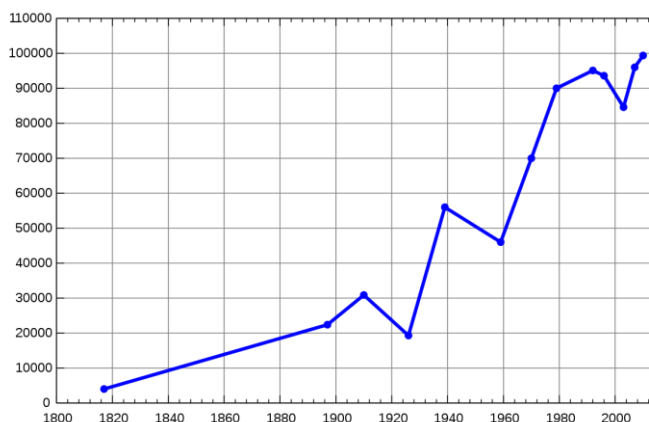
## Column Chart



A column chart is a method of displaying data with categories represented by a rectangle—sometimes called vertical bar charts. They allow easy comparisons among a number of items and trends analysis. In general, statistics and figures are difficult to understand when presented in tables or written format.

## Line Chart

A line chart is a graphical representation of an asset's historical price action that connects a series of data points with a continuous line. This is the most basic type of chart used in finance, and it typically only depicts a security's closing prices over time

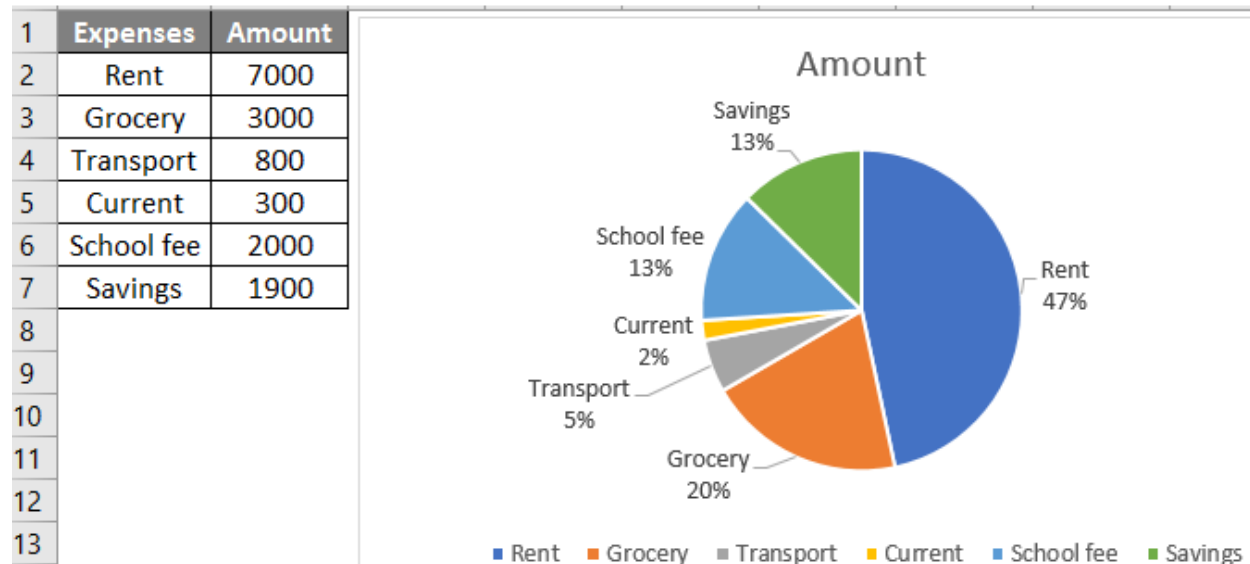


## Pie Chart



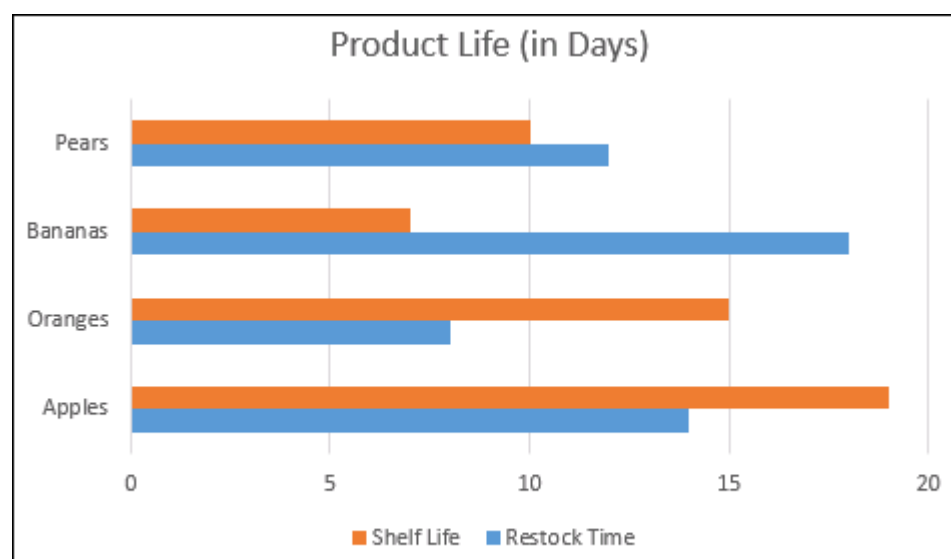
The Pie Chart is a primary chart type in Excel. Pie charts are meant to express a "part to whole" relationship, where all pieces together represent 100%. Pie charts work best to display data with a small number of categories (2-5).

## Pie Chart Examples



## Bars Chart

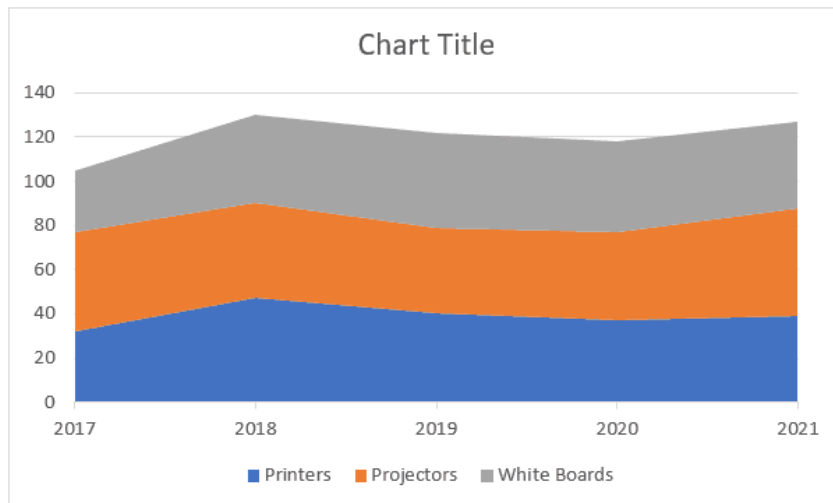
A bar chart (also called a bar graph) is a great way to visually display certain types of information, such as changes over time or differences in size, volume, or amount. Bar charts can be horizontal or vertical; in Excel, the vertical version is referred to as column chart



## Area Chart

An area chart is a primary Excel chart type, with data series plotted using lines with a filled area below. Area charts are a good way to show change over time with one data series. They offer a simple presentation that is easy to interpret at a glance.

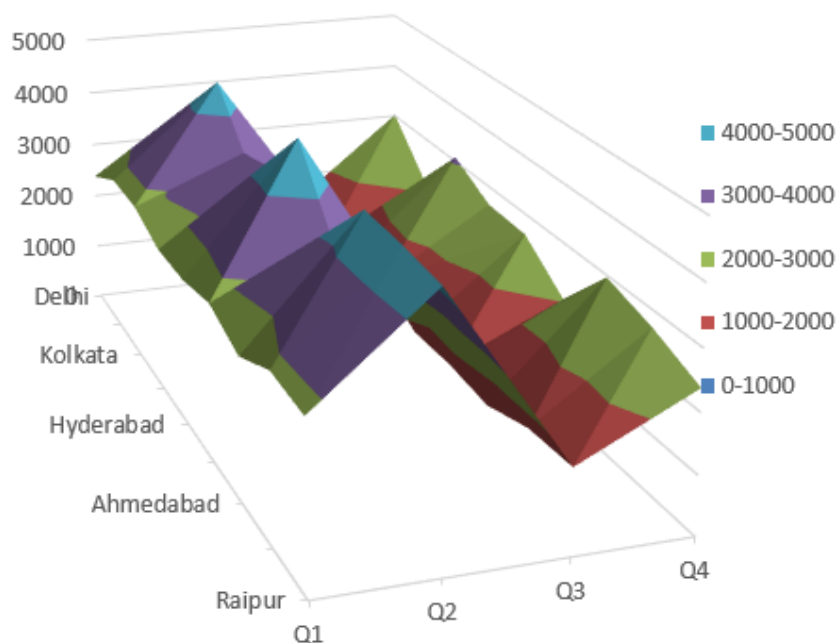
To create an area chart using the above data, highlight the data range (cells A1:B28 in the example above) and select Insert > Charts, select the Line Chart group drop-down menu and then select the second 2-D Area chart option. The following area chart is created from the selected data.



## Surface Chart

Data that is arranged in columns or rows on an Excel sheet can be plotted in a surface chart. As in a topographic map, colors and patterns indicate areas that are in the same range of values. A surface chart is useful when you want to find optimal combinations between two sets of data.

# Surface Charts in Excel



## Profile of the Problem

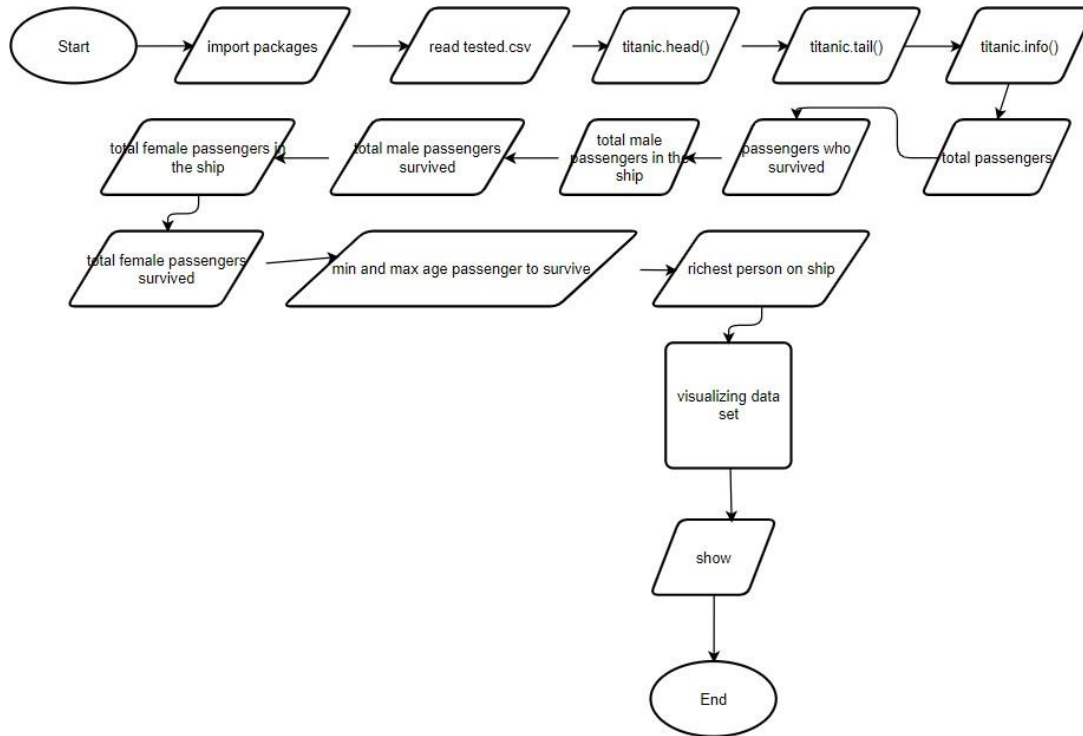
**Titanic Analysis:** I took the titanic test file and the gender submission and put them together in excel to make a csv. This is great for making charts to help you visualize. This also will help you know who died or survived. At least 70% right,.

## Software Requirement Analysis

1. Python
  - Numpy
  - Pandas
  - Networkx
  - Matplotlib
  - Scipy
2. IDE
3. Operating System
  - 4 GB Ram
  - Graphic Card

## Design

- **Flowcharts/Pseudo code**



# Implementation

EXPLORER

EXPLORATORY DATA ANAL...

tested.csv

titanicanalysis.ipynb

train.csv

titanicanalysis.ipynb

import numpy as np  
import pandas as pd

[1] ✓ 3.6s Python

titanic=pd.read\_csv('tested.csv')

[2] ✓ 0.1s Python

titanic.head()

[3] ✓ 0.1s Python

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

titanic.tail()

[4] ✓ 0.1s Python

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S

OUTLINE

TIMELINE

Jupyter Server: Local Cell 1 of 56 Go Live Prettier

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

titanic.info()

[5] ✓ 0.1s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   PassengerId  418 non-null    int64
1   Survived     418 non-null    int64
2   Pclass       418 non-null    int64
3   Name         418 non-null    object
4   Sex          418 non-null    object
5   Age          332 non-null    float64
6   SibSp        418 non-null    int64
7   Parch        418 non-null    int64
8   Ticket       418 non-null    object
9   Fare         417 non-null    float64
10  Cabin        91 non-null     object
11  Embarked     418 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

```
# total passengers
titanic['PassengerId'].nunique()
```

[6] ✓ 0.1s Python

418

```

# passengers who survived
titanic[titanic['Survived']==1][['PassengerId', 'Name', 'Sex', 'Age', 'Pclass']]

```

[7] ✓ 0.1s Python

...

	PassengerId	Name	Sex	Age	Pclass
1	893	Wilkes, Mrs. James (Ellen Needs)	female	47.0	3
4	896	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	3
6	898	Connolly, Miss. Kate	female	30.0	3
8	900	Abraham, Mrs. Joseph (Sophie Halaut Easu)	female	18.0	3
12	904	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female	23.0	1
...	...	...	...	...	...
409	1301	Peacock, Miss. Treasteall	female	3.0	3
410	1302	Naughton, Miss. Hannah	female	NaN	3
411	1303	Minahan, Mrs. William Edward (Lillian E Thorpe)	female	37.0	1
412	1304	Henriksson, Miss. Jenny Lovisa	female	28.0	3
414	1306	Oliva y Ocana, Dona. Fermina	female	39.0	1

152 rows × 5 columns

```

# total passengers survived
titanic[titanic['Survived']==1]['PassengerId'].nunique()

```

[8] ✓ 0.8s Python

... 152

```

# total male passengers in the ship
titanic[titanic['Sex']=='male']['PassengerId'].nunique()

```

[9] ✓ 0.8s Python

```

# total male passengers survived
titanic[(titanic['Sex']=='male')&(titanic['Survived']==1)]['PassengerId'].nunique()

```

[10] ✓ 0.7s Python

... 0

```

# total female passengers in the ship
titanic[titanic['Sex']=='female']['PassengerId'].nunique()

```

[11] ✓ 0.6s Python

... 152

```

# total female passengers survived
titanic[(titanic['Sex']=='female')&(titanic['Survived']==1)]['PassengerId'].nunique()

```

[12] ✓ 0.8s Python

... 152

```

# min age passenger to survive
titanic[(titanic['Survived']==1)&(titanic['Age']==titanic['Age'].min())][['PassengerId', 'Name', 'Age', 'Sex']]

```

[13] ✓ 0.6s Python

...

	PassengerId	Name	Age	Sex
354	1246	Dean, Miss. Elizabeth Gladys Millvina"	0.17	female

```

# max age passenger to survive
titanic[(titanic['Survived']==1)&(titanic['Age']==titanic['Age'].max())][['PassengerId', 'Name', 'Age', 'Sex']]

```

[14] ✓ 0.7s Python

...

	PassengerId	Name	Age	Sex
--	-------------	------	-----	-----

96 988 Cavendish, Mrs. Tyrell William (Julia Florence... 76.0 female

```
# richest person on ship
titanic[titanic['Fare']==titanic['Fare'].max()][['PassengerId', 'Name', 'Sex', 'Age', 'Fare', 'Survived', 'Pclass']]
```

[15] ✓ 0.7s Python

PassengerId	Name	Sex	Age	Fare	Survived	Pclass
343	Cardeza, Mrs. James Warburton Martinez (Charlo...	female	58.0	512.3292	1	1

```
titanic.head()
```

[16] ✓ 0.1s Python

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

```
survived_map = {0: 'No', 1: 'Yes'}
titanic['Survived'] = titanic['Survived'].map(survived_map)

# Pclass map
pclass_map = {1: 'Upper Class', 2: 'Middle Class', 3: 'Lower Class'}
titanic['Pclass'] = titanic['Pclass'].map(pclass_map)

# Embarkation port map
port_map = {'S': 'Southampton', 'C': 'Cherbourg', 'Q': 'Queenstown'}
titanic['Embarked'] = titanic['Embarked'].map(port_map)
```

[17] ✓ 0.6s Python

```
titanic.head()
```

[18] ✓ 0.9s Python

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
0	892	No	Lower Class	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Queenstown
1	893	Yes	Lower Class	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	Southampton
2	894	No	Middle Class	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Queenstown
3	895	No	Lower Class	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	Southampton
4	896	Yes	Lower Class	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	Southampton

```
# remove ticket and cabin columns
titanic.drop(['Ticket', 'Cabin'], axis=1, inplace=True)
```

[19] ✓ 0.7s Python

```
titanic.head()
```

[20] ✓ 0.8s Python

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	
0	892	No	Lower Class	Kelly, Mr. James	male	34.5	0	0	7.8292	Queenstown
1	893	Yes	Lower Class	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	7.0000	Southampton
2	894	No	Middle Class	Myles, Mr. Thomas Francis	male	62.0	0	0	9.6875	Queenstown
3	895	No	Lower Class	Wirz, Mr. Albert	male	27.0	0	0	8.6625	Southampton



```

4      896      Yes  Lower Class  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female  22.0      1      1  12.2875  Southampton

```

```

indexNames = titanic[ titanic['Survived'] == 'No' ].index
# Delete these row indexes from dataframe
titanic.drop(indexNames,inplace=True)

```

[21] ✓ 0.7s Python

```

titanic.head()

```

[22] ✓ 0.7s Python

```

...

```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked
1	893	Yes	Lower Class	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	7.0000	Southampton
4	896	Yes	Lower Class	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	12.2875	Southampton
6	898	Yes	Lower Class	Connolly, Miss. Kate	female	30.0	0	0	7.6292	Queenstown
8	900	Yes	Lower Class	Abraham, Mrs. Joseph (Sophie Halaut Easu)	female	18.0	0	0	7.2292	Cherbourg
12	904	Yes	Upper Class	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female	23.0	1	0	82.2667	Southampton

```

# family size
titanic['FamilySize'] = titanic['SibSp'] + titanic['Parch']

```

[23] ✓ 0.7s Python

```

titanic.head()

```

[24] ✓ 0.8s Python

```

...

```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	FamilySize
1	893	Yes	Lower Class	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	7.0000	Southampton	1
4	896	Yes	Lower Class	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	12.2875	Southampton	2
6	898	Yes	Lower Class	Connolly, Miss. Kate	female	30.0	0	0	7.6292	Queenstown	0
8	900	Yes	Lower Class	Abraham, Mrs. Joseph (Sophie Halaut Easu)	female	18.0	0	0	7.2292	Cherbourg	0
12	904	Yes	Upper Class	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female	23.0	1	0	82.2667	Southampton	1

```

# grouping age
age_labels = ['0-9', '10-19', '20-29', '30-39', '40-49', '50-59', '60-69', '70-79']
titanic['age_group'] = pd.cut(titanic.Age, range(0, 81, 10), right=False, labels=age_labels)

```

[25] ✓ 0.5s Python

```

titanic.head()

```

[26] ✓ 0.8s Python

```

...

```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	FamilySize	age_group
1	893	Yes	Lower Class	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	7.0000	Southampton	1	40-49
4	896	Yes	Lower Class	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	12.2875	Southampton	2	20-29
6	898	Yes	Lower Class	Connolly, Miss. Kate	female	30.0	0	0	7.6292	Queenstown	0	30-39
8	900	Yes	Lower Class	Abraham, Mrs. Joseph (Sophie Halaut Easu)	female	18.0	0	0	7.2292	Cherbourg	0	10-19
12	904	Yes	Upper Class	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female	23.0	1	0	82.2667	Southampton	1	20-29

```
titanic.drop(['SibSp', 'Parch'], axis=1, inplace=True)
```

[27] ✓ 0.6s Python

```
titanic.head()
```

[28] ✓ 0.8s Python

```
...
```

	PassengerId	Survived	Pclass	Name	Sex	Age	Fare	Embarked	FamilySize	age_group
1	893	Yes	Lower Class	Wilkes, Mrs. James (Ellen Needs)	female	47.0	7.0000	Southampton	1	40-49
4	896	Yes	Lower Class	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	12.2875	Southampton	2	20-29
6	898	Yes	Lower Class	Connolly, Miss. Kate	female	30.0	7.6292	Queenstown	0	30-39
8	900	Yes	Lower Class	Abraham, Mrs. Joseph (Sophie Halaut Easu)	female	18.0	7.2292	Cherbourg	0	10-19
12	904	Yes	Upper Class	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female	23.0	82.2667	Southampton	1	20-29

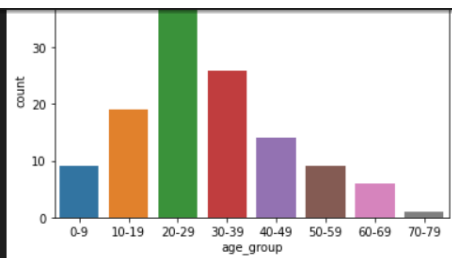
```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

[29] ✓ 4.6s Python

```
sns.countplot(x='age_group', data=titanic)
```

[30] ✓ 0.2s Python

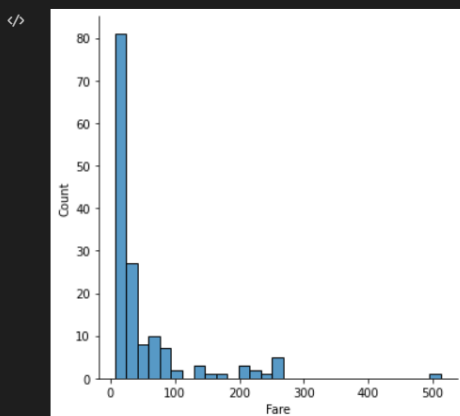
```
<AxesSubplot: xlabel='age_group', ylabel='count'>
```



```
sns.displot(titanic['Fare'], kde=False)
```

[56] ✓ 0.2s Python

```
<seaborn.axisgrid.FacetGrid at 0x2750f2d89d0>
```

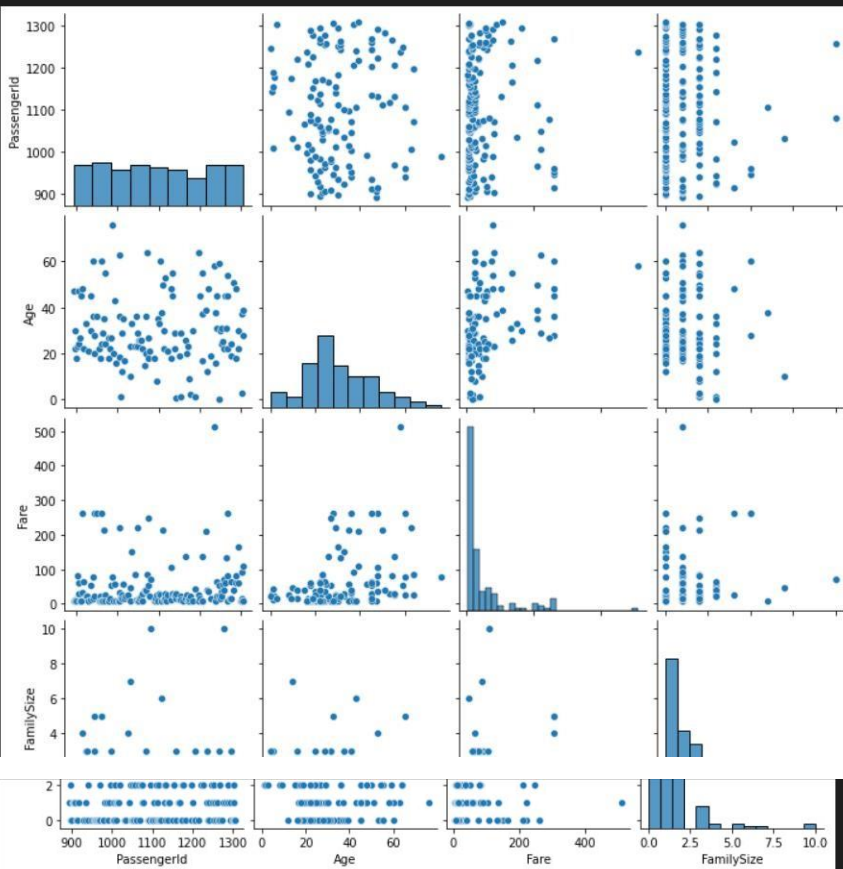


```
sns.pairplot(titanic)
```

[32] ✓ 2.1s Python

... <seaborn.axisgrid.PairGrid at 0x27509d51e40>

</>

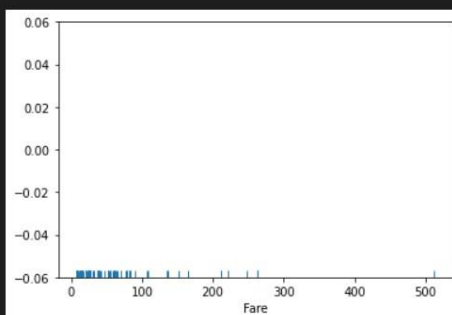


```
sns.rugplot(titanic['Fare'])
```

[33] ✓ 0.1s Python

... <AxesSubplot:xlabel='Fare'>

</>



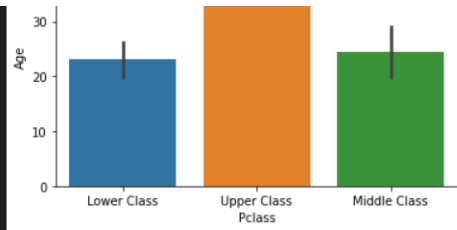
```
sns.barplot(x='Pclass', y='Age', data=titanic)
```

[34] ✓ 0.2s Python

... <AxesSubplot:xlabel='Pclass', ylabel='Age'>

</>





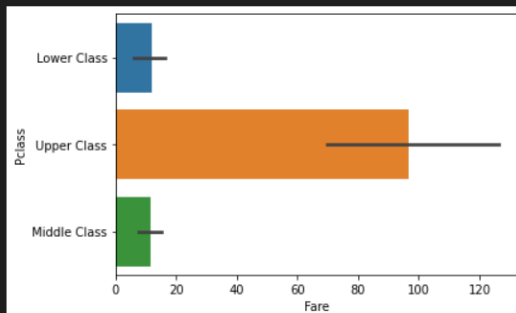
```
sns.barplot(y='Pclass',x='Fare',data=titanic,estimator=np.std)
```

[35] ✓ 0.2s

Python

```
<AxesSubplot:xlabel='Fare', ylabel='Pclass'>
```

```
</>
```



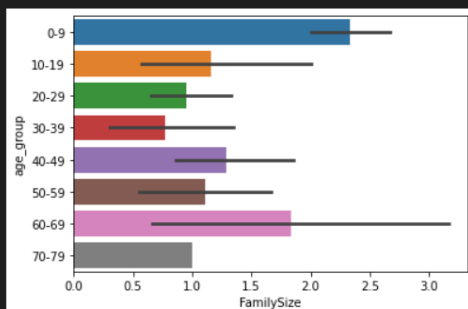
```
sns.barplot(x='FamilySize',y='age_group',data=titanic)
```

[36] ✓ 0.3s

Python

```
<AxesSubplot:xlabel='FamilySize', ylabel='age_group'>
```

```
</>
```



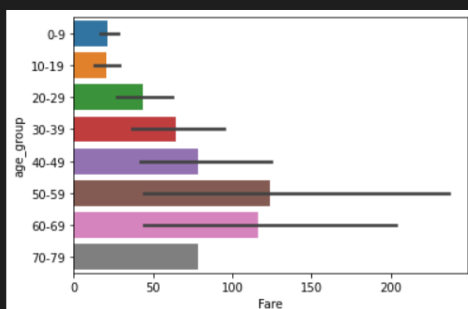
```
sns.barplot(x='Fare',y='age_group',data=titanic)
```

[37] ✓ 0.2s

Python

```
<AxesSubplot:xlabel='Fare', ylabel='age_group'>
```

```
</>
```



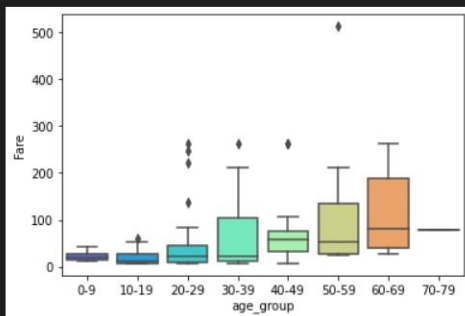
```
sns.boxplot(x="age_group", y="Fare", data=titanic,palette='rainbow')
```

[38] ✓ 0.1s

Python

... <AxesSubplot:xlabel='age\_group', ylabel='Fare'>

</>



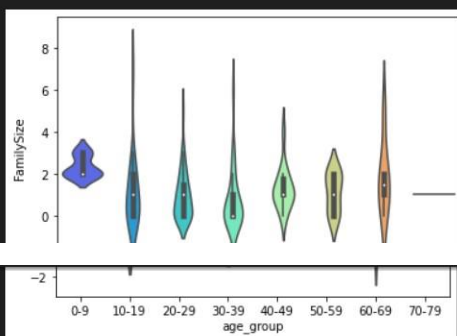
```
sns.violinplot(x="age_group", y="FamilySize", data=titanic,palette='rainbow')
```

[39] ✓ 0.2s

Python

... <AxesSubplot:xlabel='age\_group', ylabel='FamilySize'>

</>



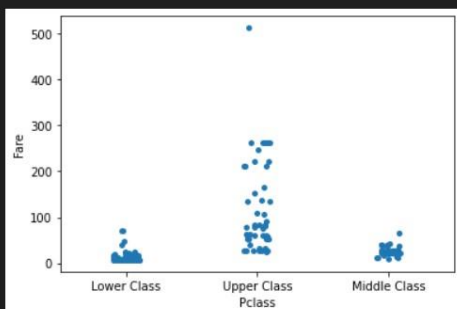
```
sns.stripplot(x="Pclass", y="Fare", data=titanic)
```

[40] ✓ 0.1s

Python

... <AxesSubplot:xlabel='Pclass', ylabel='Fare'>

</>



```
sns.stripplot(x="Pclass", y="Fare", data=titanic,hue='Embarked')
```

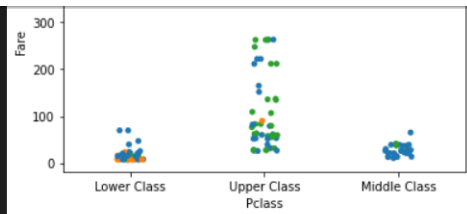
[57] ✓ 0.2s

Python

... <AxesSubplot:xlabel='Pclass', ylabel='Fare'>

</>



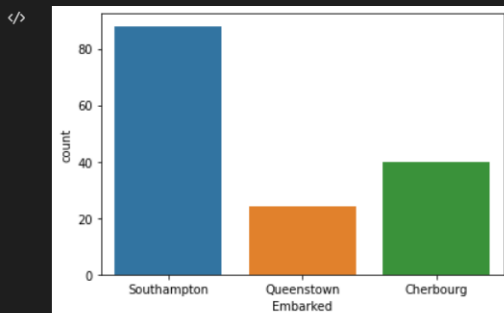


```
# which area has most number of survivors
sns.countplot(x='Embarked',data=titanic)
```

[42] ✓ 0.1s

Python

... <AxesSubplot:xlabel='Embarked', ylabel='count'>



```
titanic['Embarked'].value_counts()
```

[43] ✓ 0.3s

Python

```
... Southampton    88
   Cherbourg      40
   Queenstown     24
   Name: Embarked, dtype: int64
```

```
sns.heatmap(titanic.corr(),annot=True)
```

[44] ✓ 0.2s

Python

... <AxesSubplot:>



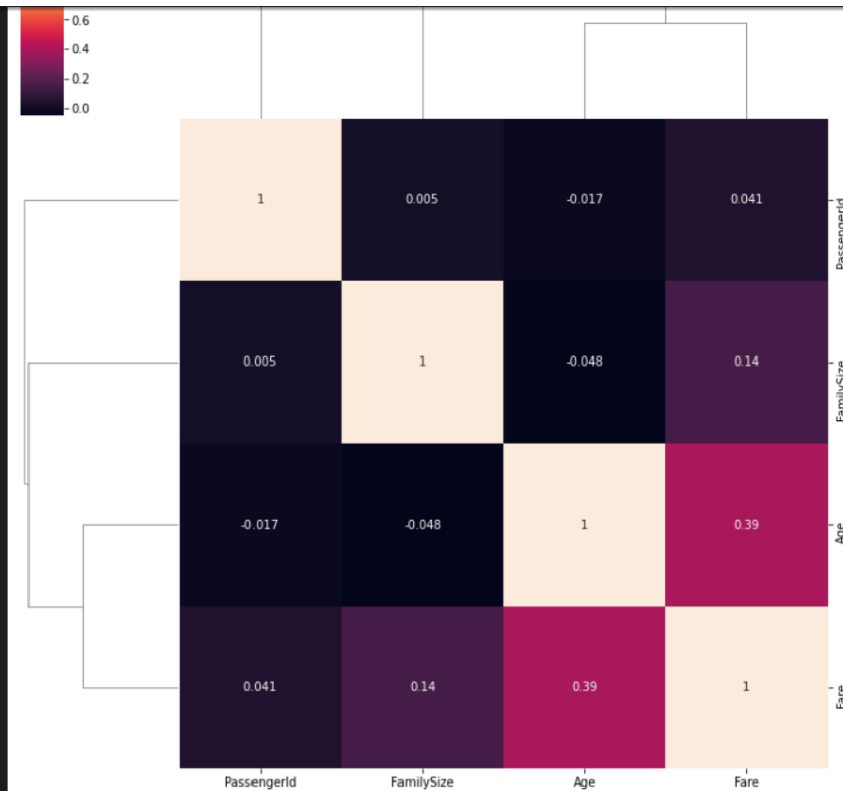
```
sns.clustermap(titanic.corr(),annot=True)
```

[45] ✓ 0.3s

Python

... <seaborn.matrix.ClusterGrid at 0x2750de1b2e0>





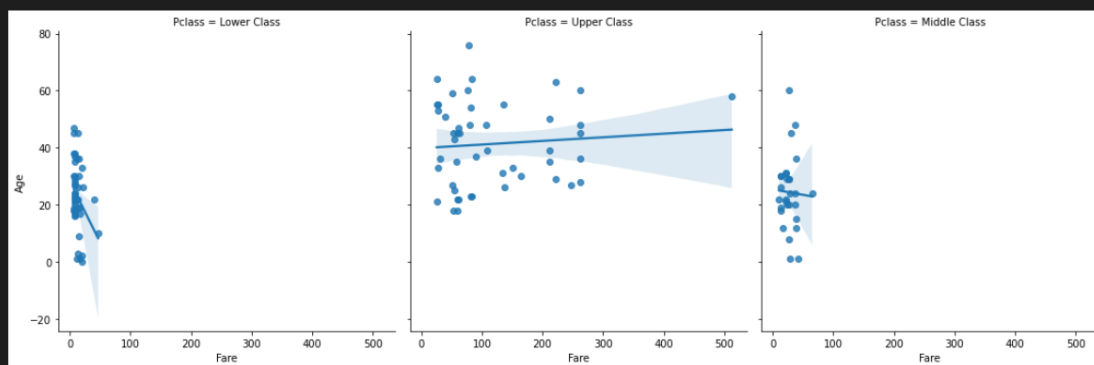
```
sns.lmplot(x='Fare',y='Age',data=titanic,col='Pclass',palette='magma')
```

[46] ✓ 0.7s

Python

```
<seaborn.axisgrid.FacetGrid at 0x2750e03c250>
```

</>



```
titanic.head()
```

[47] ✓ 0.5s

Python

```
...
```

	PassengerId	Survived	Pclass	Name	Sex	Age	Fare	Embarked	FamilySize	age_group
1	893	Yes	Lower Class	Wilkes, Mrs. James (Ellen Needs)	female	47.0	7.0000	Southampton	1	40-49
4	896	Yes	Lower Class	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	12.2875	Southampton	2	20-29
6	898	Yes	Lower Class	Connolly, Miss. Kate	female	30.0	7.6292	Queenstown	0	30-39

8	900	Yes	Lower Class	Abraham, Mrs. Joseph (Sophie Halaut Easu)	female	18.0	7.2292	Cherbourg	0	10-19
12	904	Yes	Upper Class	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female	23.0	82.2667	Southampton	1	20-29

```

from plotly import __version__
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot

print(__version__)

```

[48] ✓ 0.5s Python

... 5.10.0

```

import cufflinks as cf

```

[49] ✓ 0.4s Python

```

init_notebook_mode(connected=True)

```

[50] ✓ 0.2s Python

...

```

cf.go_offline()

```

[51] ✓ 0.3s Python

...

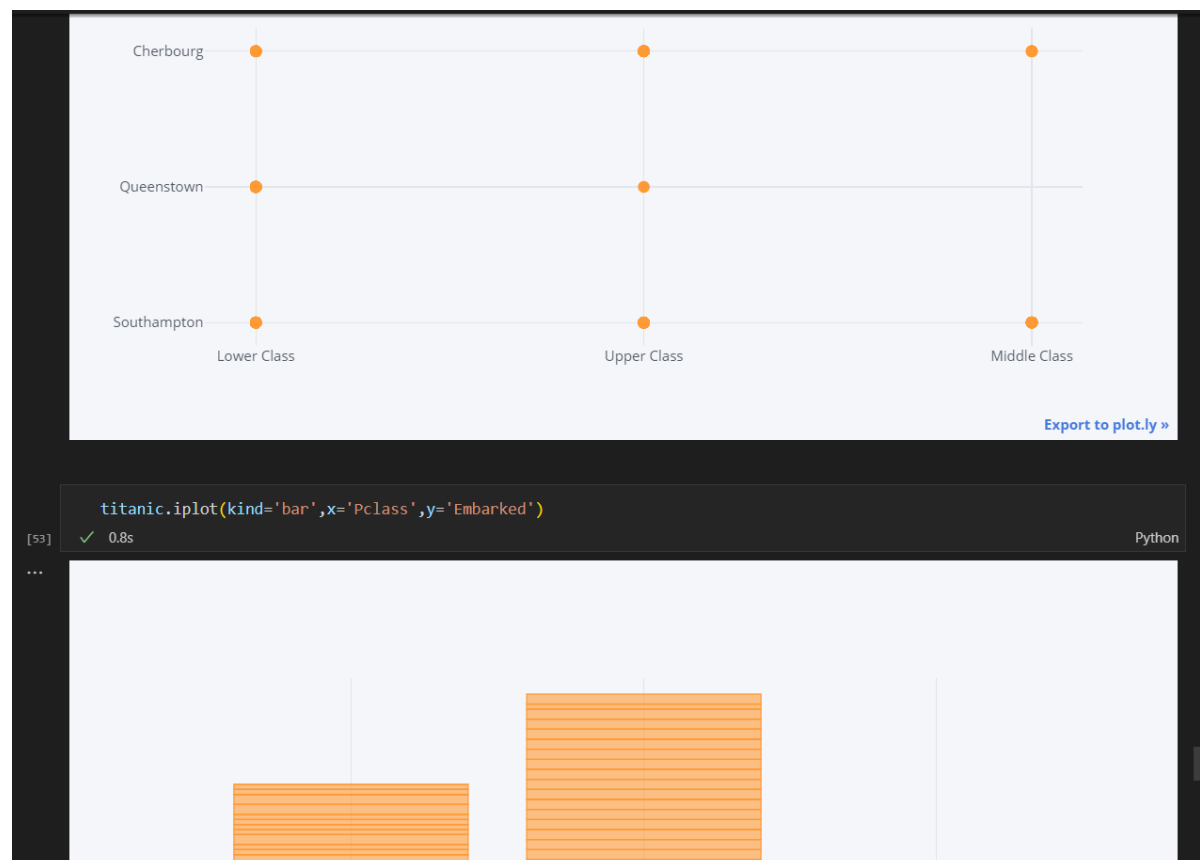
```

titanic.iplot(kind='scatter',x='Pclass',y='Embarked',mode='markers',size=10)

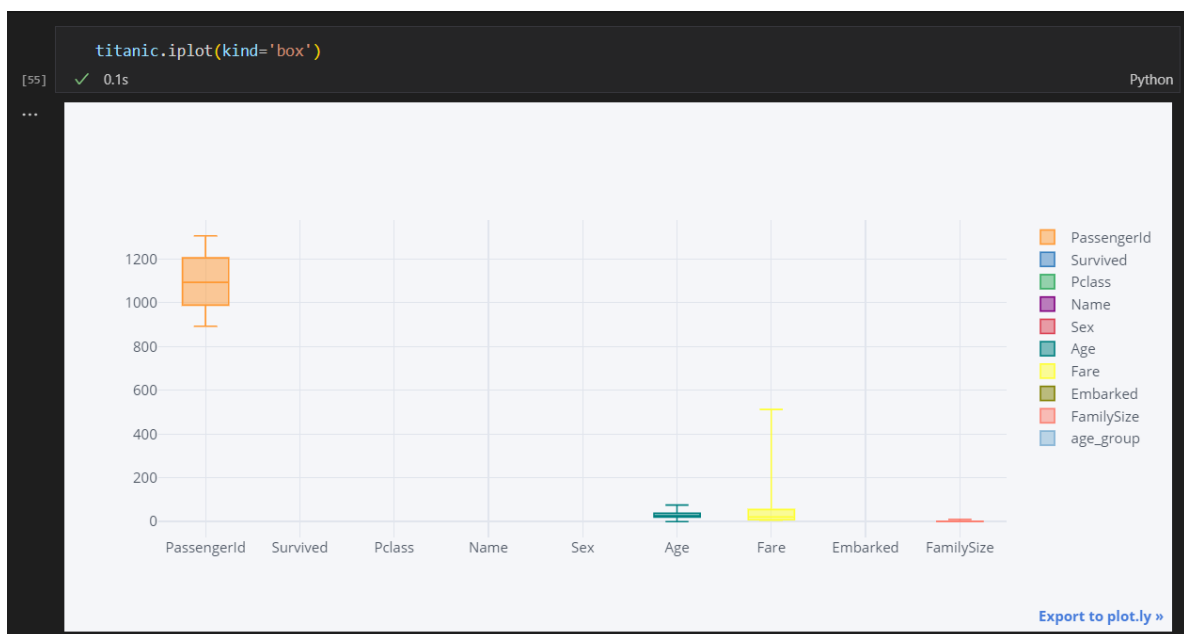
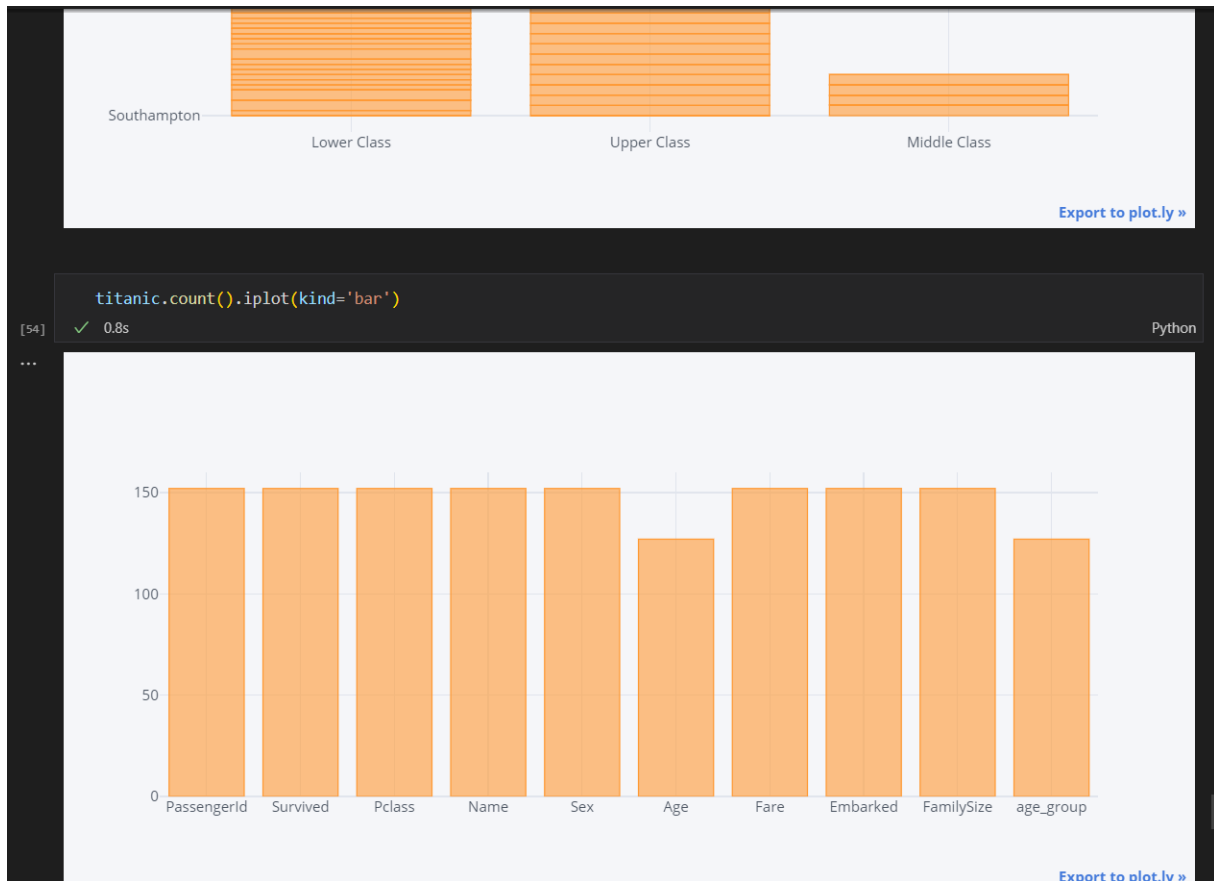
```

[52] ✓ 1.1s Python

...







## Learning Outcomes

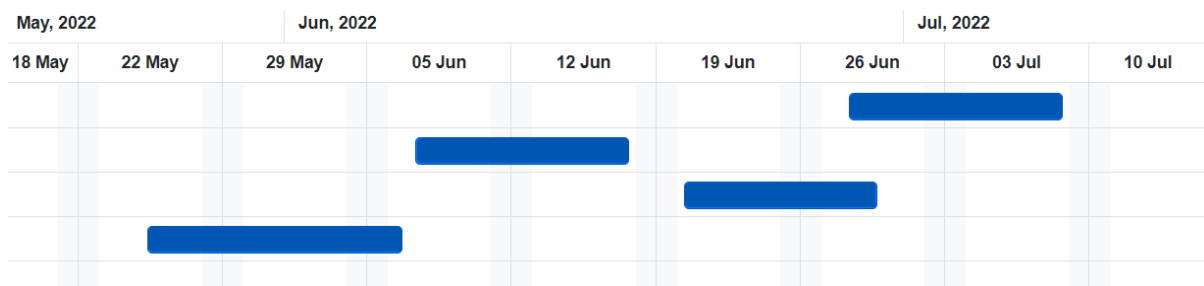
I'm capable of creating some quality projects now that I've finished my data science course from upgrad. I'm grateful to Upgrad for offering such a good course. The course and the method of instruction have made me extremely satisfied. I'm grateful to Upgrad for offering such a good course. The course and the method of instruction have made me extremely satisfied.

Data Science is the study of data. It is a multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structural and unstructured data. Data science is related to data mining, machine learning and big data.

Data mining is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems. Data mining is an interdisciplinary subfield of computer science and statistics with an overall goal to extract information (with intelligent methods) from a data set and transform the information into a comprehensible structure for further use.

## Gantt Chart

Name	↓ ⋮	Start Date	⋮	End Date	⋮	Duration	⋮	Progress %
Statistics for Data Science		Jun 28, 2022		Jul 08, 2022		9 days		100
Fundamentals of Python and Libraries		Jun 07, 2022		Jun 17, 2022		9 days		100
Data Visualization Using Tableau & SQL		Jun 20, 2022		Jun 29, 2022		8 days		100
Data Analysis in Excel		May 25, 2022		Jun 06, 2022		9 days		100



## Bibliography

- <https://www.kaggle.com/datasets/brendan45774/test-file>
- <https://numpy.org/numpy-tutorials/content/tutorial-svd.html>
- <https://networkx.org/>
- <https://pandas.pydata.org/docs/index.html>