

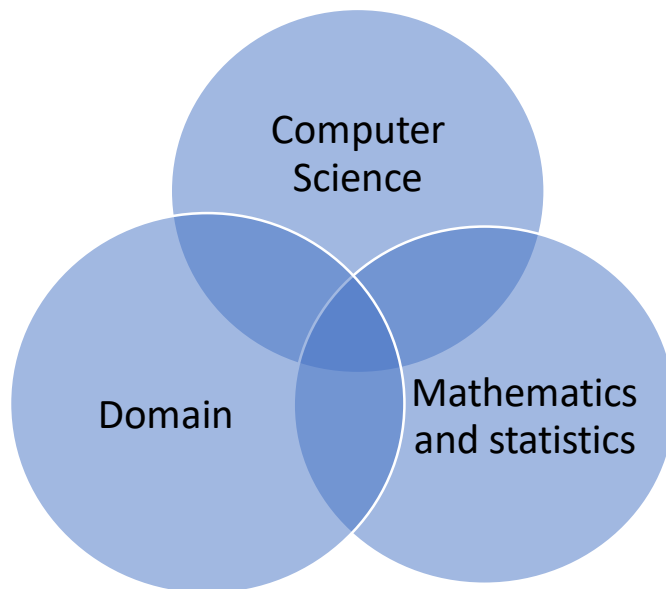
Contents

Data science	2
Skills Required	3
Data preparation	3
Introduction to colab and Python	5
Part 3: EDA/Data wrangling/Feature engineering	8
Introduction to PCA	12
Calculate covariance:.....	14
Calculate Eigen Value	15
Calculate Eigen Vectors	16
Calculate the new features	17
Plot the new dataset	18
Tabloid	19
Feature Engineering	21
Data Exploration Techniques	25
Relationship Between Two Variables	29
Violin Plot.....	31
Data Science Terminologies	34
SweetViz	36
Installing Sweetviz	36
Tableau	42
How To Frame right question for your data	43
Descriptive Data Analysis:.....	44
Predictive Data Analysis.....	44
Regression.....	44
Gretl:.....	49

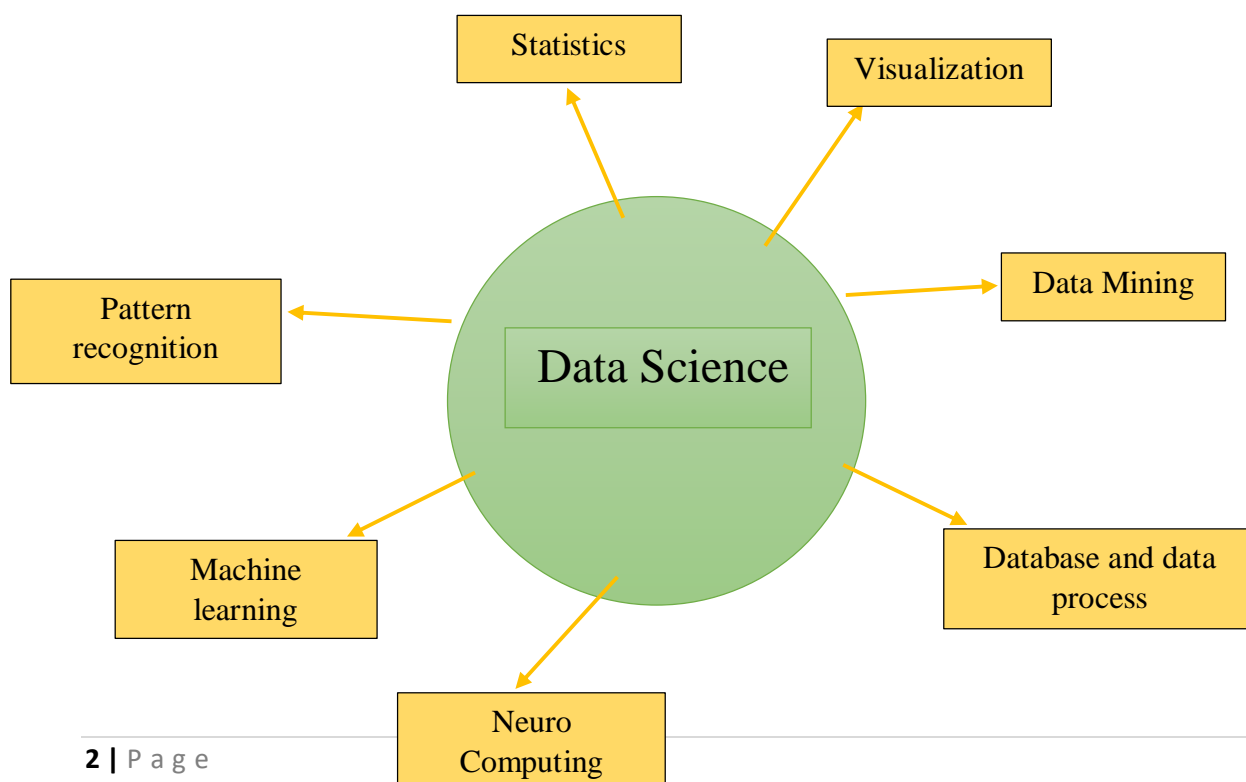
Data science

Data science is the study of data. It involves developing methods of recording, storing, and analysing data to effectively extract useful information. The goal of data science is to gain insights and knowledge from any type of data. Data science is more closely related to the mathematics field of Statistics, which includes the collection, organization, analysis, and presentation of data.

The three major components



Path way to achieve the goals



Skills Required

Communication, presentation, domain knowledge, real life practise, programming and creativity

How to go about

Steps involved in Data Analytics is

1. Data preparation
2. Visualization
3. Modelling
4. Communication, Presentation

Data preparation

Pine Line

Raw data => ETL => Database => Analyse => Modelling => visualization
=> insights

Data preparation is an important aspect of data processing. The analyst spends about 80% of their time gathering and preparing the data rather than analysing it or developing machine learning models.

Before stepping up in google colab

Some of the basic terms to be known to understand the data using google colab and to plot the basic graphs.

Importing the libraries

1. Panda as pd

It is used for data manipulation and data preparation. It can present data in a way that is suitable for data analysis. Pandas provide extremely streamlined forms of data representation. This helps to analyze and understand data better. Simpler data representation facilitates better results for data science projects.

2. Numpy as np

NumPy is a Python library used for working with arrays(array manipulation). It also has functions for working in domain of linear algebra and matrices. Mathematical operations on Numpy n-Dimension Arrays

3. Seaborn as sns

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

4. %matplotlib inline

%matplotlib inline sets the backend of matplotlib to the 'inline' backend: With this backend, the output of plotting commands is displayed inline within frontends like the Jupyter notebook, directly below the code cell that produced it. The resulting plots will then also be stored in the notebook document.

Count plot

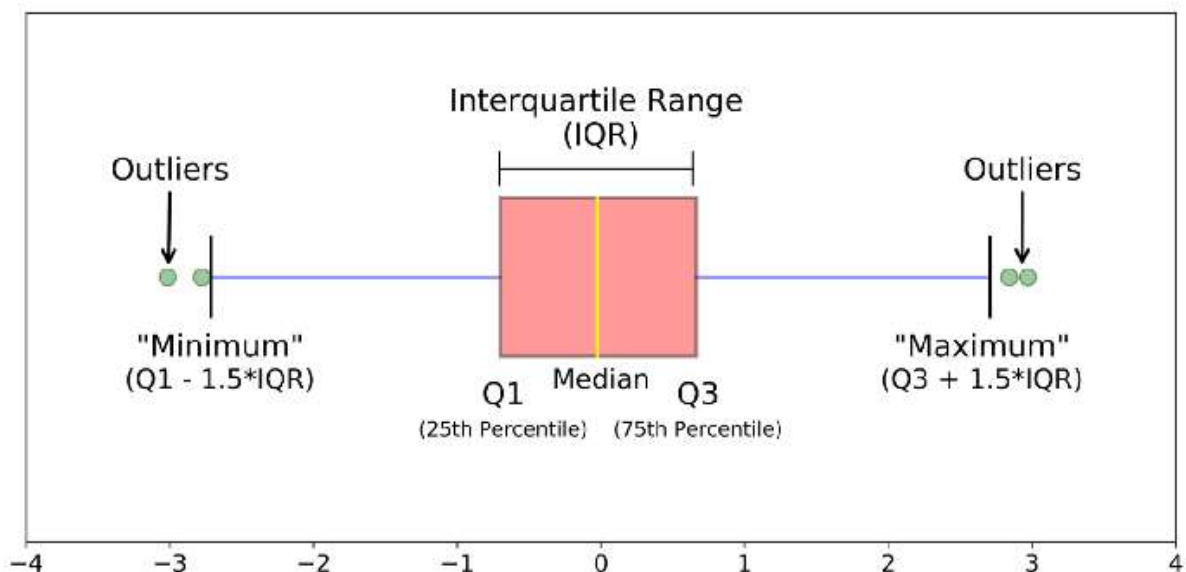
we use seaborn count plot. It is a graphical display to show the number of occurrences or frequency for each categorical data using bars.

Histogram

It is used when You want to see the shape of the data's distribution, especially when determining whether the output of a process is distributed approximately normally

BoxPlot

Boxplots are a standardized way of displaying the distribution of data based on a five number summary ("minimum", first quartile (Q1), median, third quartile (Q3), and "maximum").



median (Q2/50th Percentile):
the middle value of the dataset.

first quartile (Q1/25th Percentile):
the middle number between the smallest number (not the "minimum") and the median of the dataset.

third quartile (Q3/75th Percentile):

middle value between the median and the highest value (not the “maximum”) of the dataset.

interquartile range (IQR):

25th to the 75th percentile.

whiskers (shown in blue)

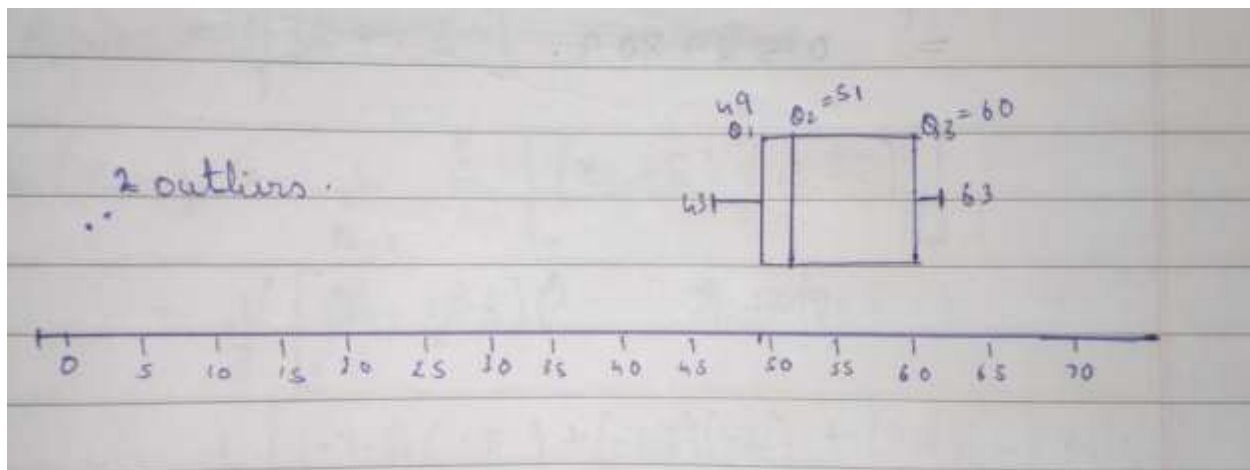
outliers (shown as green circles)

“maximum”: $Q3 + 1.5 \cdot IQR$

“minimum”: $Q1 - 1.5 \cdot IQR$

Using these above steps we can plot a boxplot

Example: dataset 2,43,49,50,51,51,53,54,60,62,63



Introduction to colab and Python

Colab notebooks allow you to combine executable code and rich text in a single document, along with images, graphs and more.

To be precise, **Colab** is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup.

Exercise 1: Titanic data

Part 1.2: To collect data and analyse data

1. Importing libraries from python



2. To read the file and display first 5 records

```
titanic_data=pd.read_csv('/content/Titanic.csv')
Titanic_data.head(5)
```

```
[ ] titanic_data=pd.read_csv('/content/Titanic.csv')
print("# of passenger in origial data: "+str(len(titanic_data.index)))

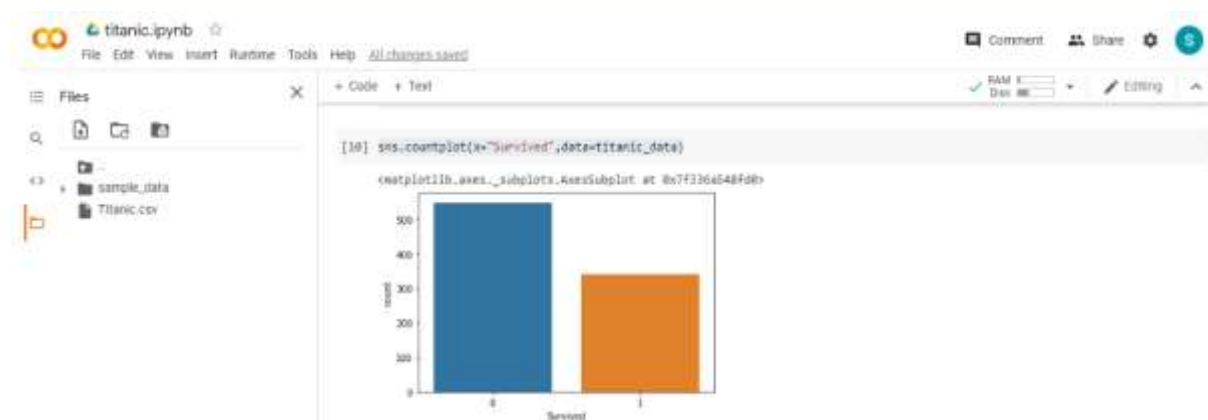
# of passenger in origial data: 891

[ ] titanic_data.head(5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

3. To display how many people survived

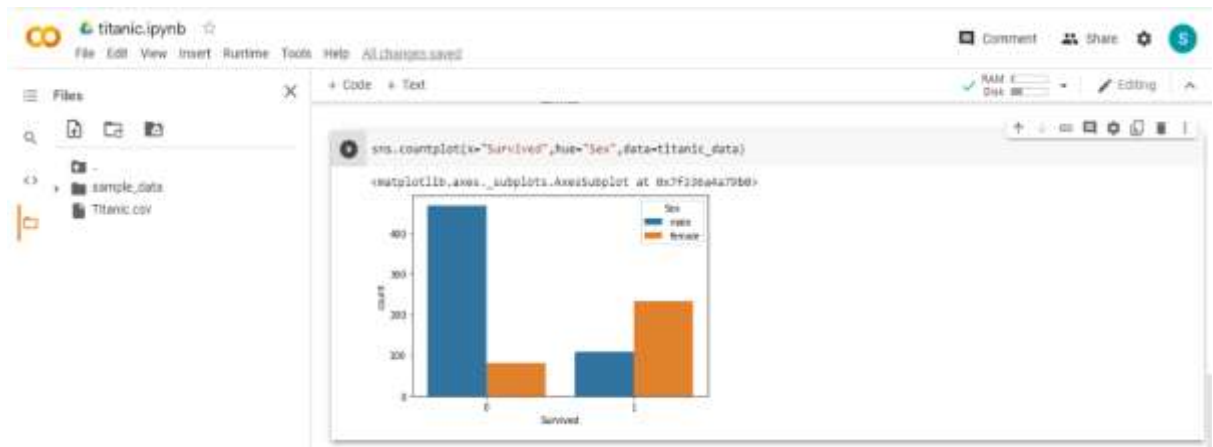
```
sns.countplot(x="Survived",data=titanic_data)
```



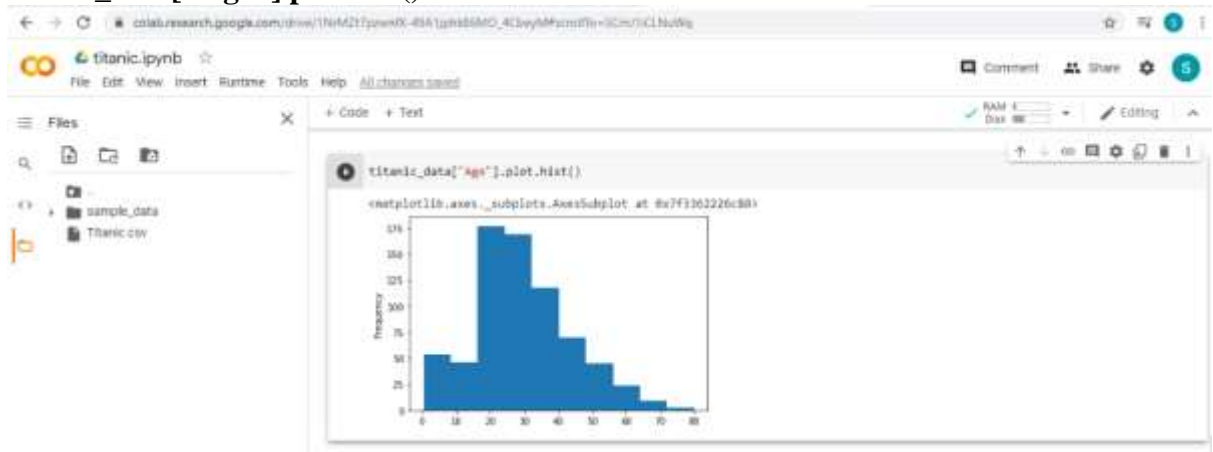
4. To plot number of male and female who survived

```
sns.countplot(x="Survived",hue="Sex",data=titanic_data)
```

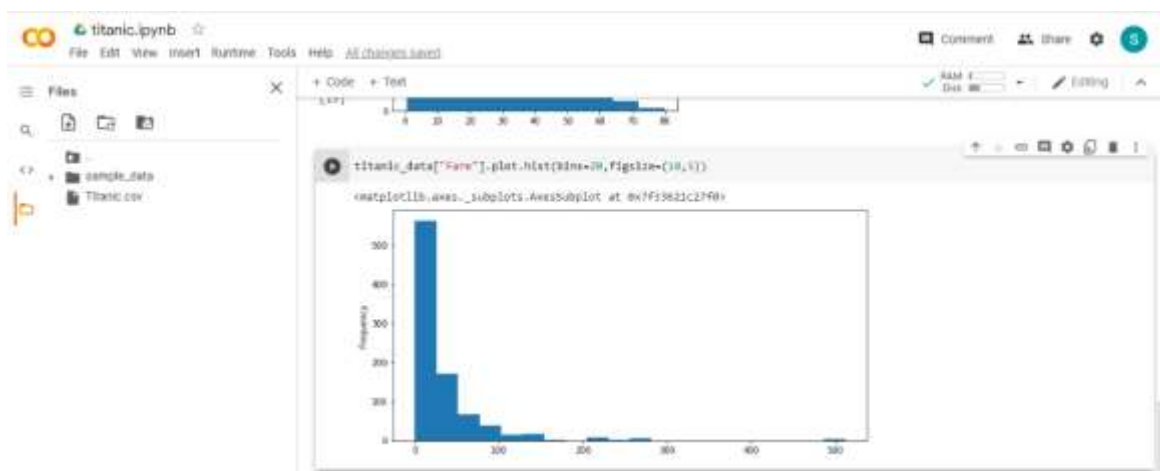
hue will take different value to sex



5. To display age to know age distribution
`titanic_data["Age"].plot.hist()`



6. To display flight fare.
`titanic_data["Fare"].plot.hist(bins=20,figsize=(10,5))`



7. To display information about titanic data
`titanic_data.info()`

```

titanic_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  --   ---
 0   PassengerId      891 non-null    int64  
 1   Survived         891 non-null    int64  
 2   Pclass           891 non-null    int64  
 3   Name             891 non-null    object  
 4   Sex              891 non-null    object  
 5   Age              714 non-null    float64 
 6   SibSp            891 non-null    int64  
 7   Parch           891 non-null    int64  
 8   Ticket           891 non-null    object  
 9   Fare             891 non-null    float64 
10   Cabin           204 non-null    object  
11   Embarked         890 non-null    object  
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB

```

Part 3: EDA/Data wrangling/Feature engineering

Exploratory data analysis (EDA) is used to analyse and investigate data sets.

EDA is used to summarize their main characteristics, maximize insight into a data set, uncover underlying structure, extract important features, detect outliers, check and remove all null values

1. Checking null values and the sum titanic_data.isnull()

```

titanic_data.isnull()

```

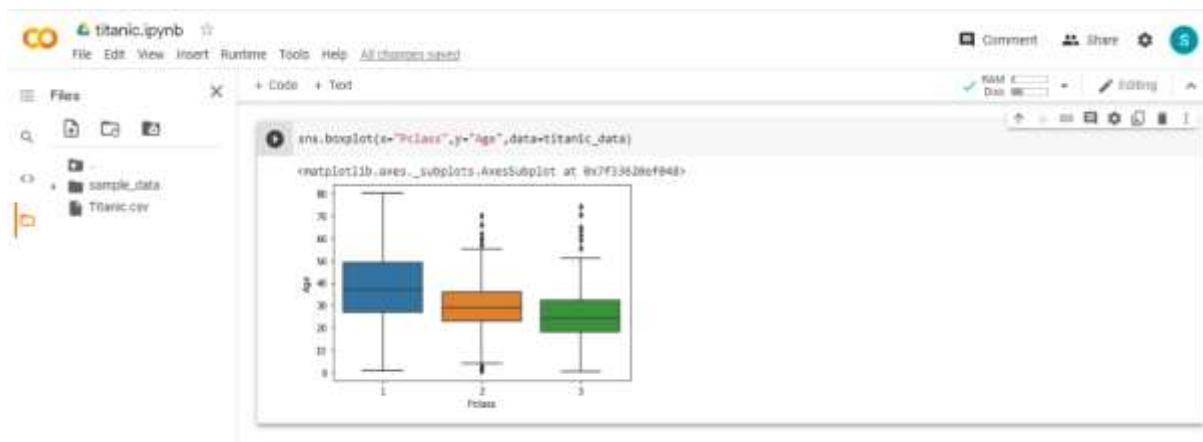
PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	True	False
...
886	False	False	False	False	False	False	False	False	False	True	False
887	False	False	False	False	False	False	False	False	False	False	False
888	False	False	False	False	True	False	False	False	False	True	False
889	False	False	False	False	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False	False	False	True	False

891 rows x 12 columns

Titanic_data.isnull().sum()



2. To display box plot based on age and passenger class
sns.boxplot(x="Pclass",y="Age",data=titanic_data)



- The age of people who travelled in first class is 0-80, median is 40, first quartile is 30-40 and third quartile is 40-50.
- first quartile and third quartile is know as inter quartile distance.

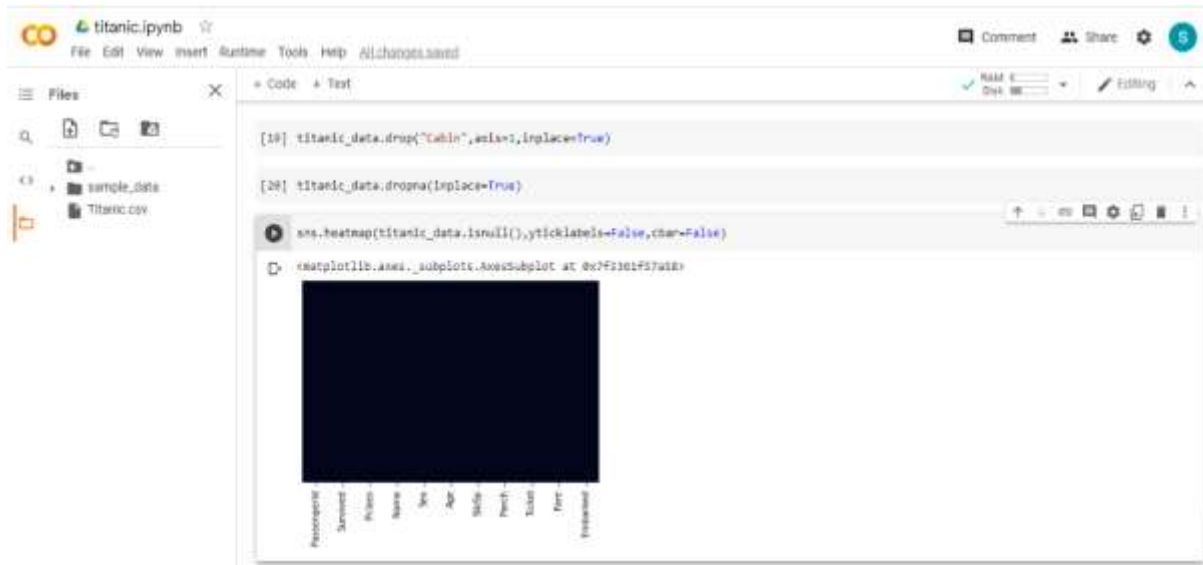
3. To drop unwanted column cabin.

To drop null values

[illegible]

titanic_data.dropna(inplace=True) => inplace saves the result

```
sns.heatmap(titanic_data.isnull(),yticklabels=False,cbar=False)
```



- If you get complete black it means no null values. So all the null values are removed.

4. To convert string to numeric values

For example sex female = 0 and male = 1 here we use dummies from pd

- **sex :**
`sex=pd.get_dummies(titanic_data["Sex"])`
`sex=pd.get_dummies(titanic_data["Sex"],drop_first=True)`



- embarked

The screenshot shows a Jupyter Notebook with the following code and output:

```
[18] embarked=pd.get_dummies(titanic_data["Embarked"])
```

```
[19] embarked.head(3)
```

	C	Q	S
0	0	0	1
1	1	0	0
2	0	0	1

```
[20] embarked=pd.get_dummies(titanic_data["Embarked"],drop_first=True)
```

```
[21] embarked.head(3)
```

	Q	S
0	0	1
1	0	0
2	0	1

- Passenger class

The screenshot shows a Jupyter Notebook with the following code and output:

```
[33] pclass=pd.get_dummies(titanic_data["Pclass"])
```

```
[34] pclass.head(3)
```

	1	2	3
0	0	0	1
1	1	0	0
2	0	0	1

```
[35] pclass=pd.get_dummies(titanic_data["Pclass"],drop_first=True)
```

```
[36] pclass.head(3)
```

	2	3
0	0	1
1	0	0
2	0	1

12. To concatenate the changed column to the original file and to drop unwanted column

```
titanic_data=pd.concat([titanic_data,sex,Embarked,pclass],axis=1)
titanic_data.drop(['Sex','Embarked','PassengerId','Name','Ticket','Pclass'],axis=1,
inplace=True)
```

```

[37] titanic_data=pd.concat([titanic_data,sex,embarked,pclass],axis=1)

[38] titanic_data.head(3)

   PassengerId  Survived  Pclass    Name  Sex  Age  SibSp  Parch    ticket   Fare  Embarked male  q  s  2  3
0         0         1         3  Braund, Mr. Owen Harris   male  22.0      1      0   A/5 21171   7.2500      S      1  0  1  0  1
1         1         0         1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1      0   PC 17599   71.2833      C      0  0  0  0  0
2         2         1         3  Heikinen, Miss. Laina   female  26.0      0      0   STON/O2  7.9200      S      0  0  1  0  1

[39] titanic_data.drop(['sex', 'embarked', 'PassengerId', 'Name', 'Ticket', 'Pclass'],axis=1,inplace=True)

titanic_data.head(3)

   survived  Age  SibSp  Parch    fare  male  q  s  2  3
0         0  22.0      1      0   7.2500      1  0  1  0  1
1         0  38.0      1      0  71.2833      0  0  0  0  0
2         1  26.0      0      0   7.9200      0  0  1  0  1

```

Introduction to PCA

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

The main feature of PCA is managing over fitting and underfitting

If we don't have right variance it can lead to overfitting an underfitting.

Overfitting:

In *overfitting*, a statistical model describes random error or noise instead of the underlying relationship. Overfitting occurs when a model is excessively complex, such as having too many parameters relative to the number of observations. A model that has been overfit has poor predictive performance, as it overreacts to minor fluctuations in the training data.

Low bias and high variance.

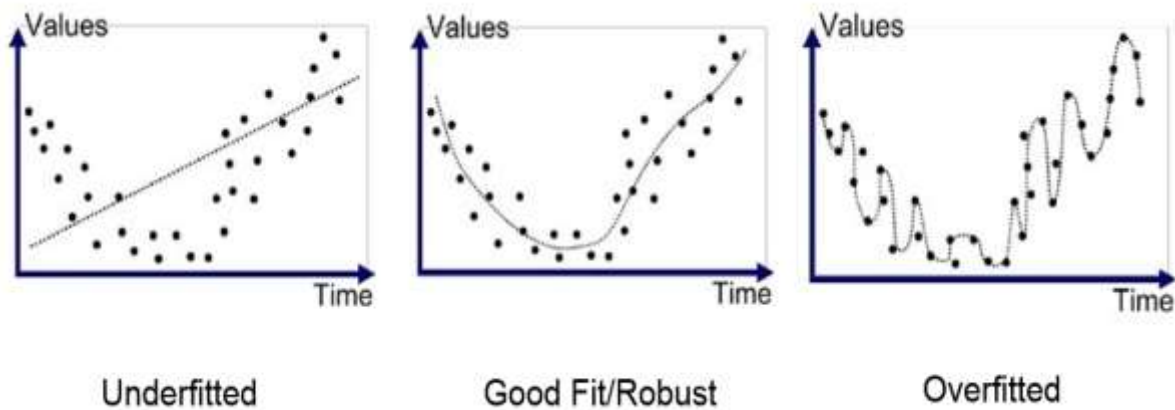
Solution for overfitting is we can drop some features, but it is not the right correct way, without removing any feature we can retain all of them so the method used here is called PCA (By finding linear combination features.

Underfitting:

Underfitting occurs when a statistical model or machine learning algorithm cannot capture the underlying trend of the data. Underfitting would occur, for example, when fitting a linear model to non-linear data. Such a model too would have poor predictive performance.

High bias and low variance

Solution for underfitting is we can add feature if it doesn't fit data well.



Why PCA is used

- Removes Correlated Features
- Improves Algorithm Performance
- Reduces Overfitting
- Improves Visualization

Algorithms used to eliminate features

- Singular value decomposition (SVD)
- Eigen value decomposition
- Independent component analysis

Let's take an example of real data to perform PCA

x	4	8	13	7
y	11	4	5	14

Find Mean of x and y

$\bar{x} = \frac{32}{4}$	$\bar{y} = \frac{34}{4}$
$= 8$	$= 8.5$

Calculate covariance:

Find Covariance

$$\begin{aligned} \text{Var}(x) &= \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \\ &= \frac{1}{3} \left[(4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2 \right] \\ &= \frac{1}{3} \left[(-4)^2 + (0)^2 + (5)^2 + (-1)^2 \right] \\ &= \frac{1}{3} \left[16 + 0 + 25 + 1 \right] \\ &= 14 \end{aligned}$$

$$\begin{aligned} \text{Cov}(y, y) &= \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2 \\ &= 23 \end{aligned}$$

$$\begin{aligned} \text{Cov}(x, y) &= \frac{1}{n-1} \sum_{i=1}^n \left[(x_i - \bar{x})(y_i - \bar{y}) \right] \\ &= \frac{1}{3} \left[(4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) \right. \\ &\quad \left. + (7-8)(14-8.5) \right] \\ &= -11 \end{aligned}$$

$$\begin{aligned} \text{Cov}(y, x) &= \text{same answer as } \text{Cov}(x, y) \\ &= -11 \end{aligned}$$

Calculate Eigen Value

To find eigen value

$$\det (S - \lambda I) = 0 \quad \Rightarrow \text{Formula}$$

$$\begin{vmatrix} 14 - \lambda & -11 \\ -11 & 23 - \lambda \end{vmatrix} = 0$$

$$\lambda^2 - 37\lambda + 201 = 0$$

Use $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ formula to find eigen value

$$\therefore \text{Eigen value } \lambda_1 = 30.3849$$

$$\text{Eigen value } \lambda_2 = 6.61$$

Calculate Eigen Vectors

To find eigen vector of λ_1

$$(S - \lambda_1, I) u_1 = 0 \Rightarrow \text{Formula}$$

$$\begin{bmatrix} 14 - \lambda_1 & -11 \\ -11 & 23 - \lambda_1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} (14 - \lambda_1)u_1 & -11u_2 \\ -11u_1 & (23 - \lambda_1)u_2 \end{bmatrix} = 0$$

$$\begin{aligned} (14 - \lambda_1)u_1 - 11u_2 &= 0 \\ -11u_1 + (23 - \lambda_1)u_2 &= 0 \end{aligned}$$

$$\frac{u_1}{11} = \frac{u_2}{14 - \lambda_1} \Rightarrow t$$

$$u_1 = 11$$

$$u_2 = 14 - \lambda_1$$

$$u \text{ of } \lambda_1 = \begin{bmatrix} 11 \\ 14 - \lambda_1 \end{bmatrix} \therefore \text{Eigen vector of } \lambda_1$$

So now we need to normalise

$$\therefore \begin{bmatrix} 11 \\ -16.3849 \end{bmatrix}$$

$$e_1 = \begin{bmatrix} 11 / \sqrt{11^2 + (-16.3849)^2} \\ -16.3849 / \sqrt{11^2 + (-16.3849)^2} \end{bmatrix}$$

$$= \begin{bmatrix} 0.5574 \\ -0.838 \end{bmatrix} \therefore \text{Eigen vector of } \lambda_1$$

We have to follow the same procedure to find λ_2 eigen vector.

$$\therefore e_2 = \begin{bmatrix} 0.833 \\ 0.5574 \end{bmatrix}$$

Calculate the new features

Derive the new data set

	E_1	E_2	E_3	E_4
PCA1	P_{11}	P_{12}	P_{13}	P_{14}

Formula to find new data sets

$$P = e_i^T \begin{bmatrix} x - \bar{x} \\ y - \bar{y} \end{bmatrix}$$

$$P_{11} = e_1^T \begin{bmatrix} 4 - 8 \\ 11 - 8.5 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5574 & -0.8353 \end{bmatrix} \begin{bmatrix} -4 \\ 2.5 \end{bmatrix}$$

$$= \begin{bmatrix} (0.5574)(-4) + (-0.8353)(2.5) \end{bmatrix}$$

$$= -4.3052$$

Same way calculate $P1_2, P1_3, P1_4$

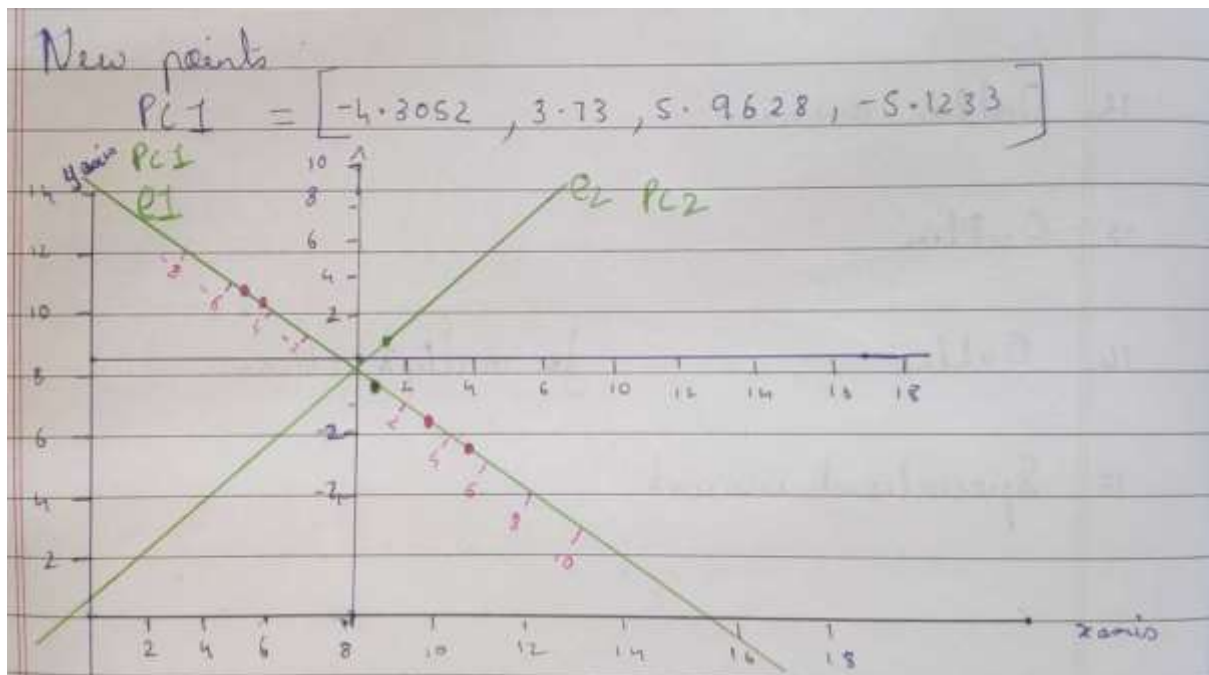
$$\therefore P1_2 = e_1^T \begin{bmatrix} 8 & -8 \\ 4 & -8.5 \end{bmatrix}$$

$$= 3.73$$

$$P1_3 = 5.6928$$

$$P1_4 = -5.1233$$

Plot the new dataset



Tabloid

Tableau is a visual analytics platform transforming the way we use data to solve problems, empowering people and organizations to make the most of their data. Tableau helps persons and organizations be more data-driven.

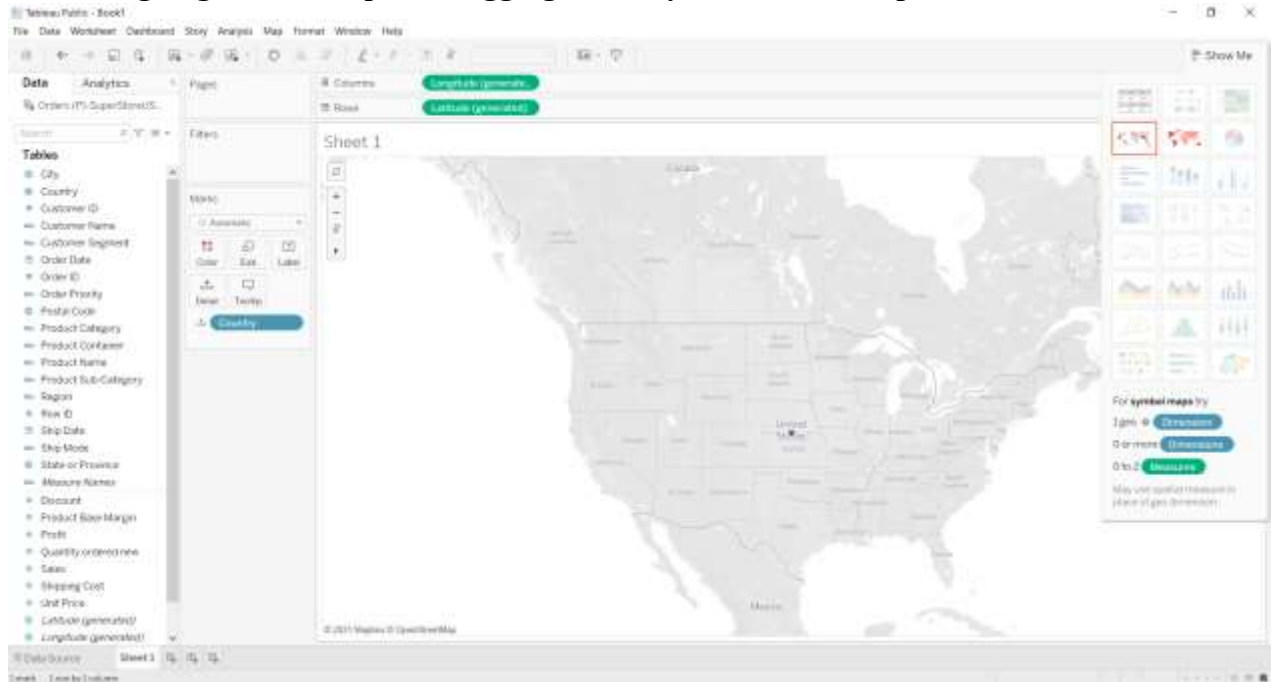
Performing on tableau operations on SuperStoreUS-2015 dataset:

1. Dragging 'orders' into workspace

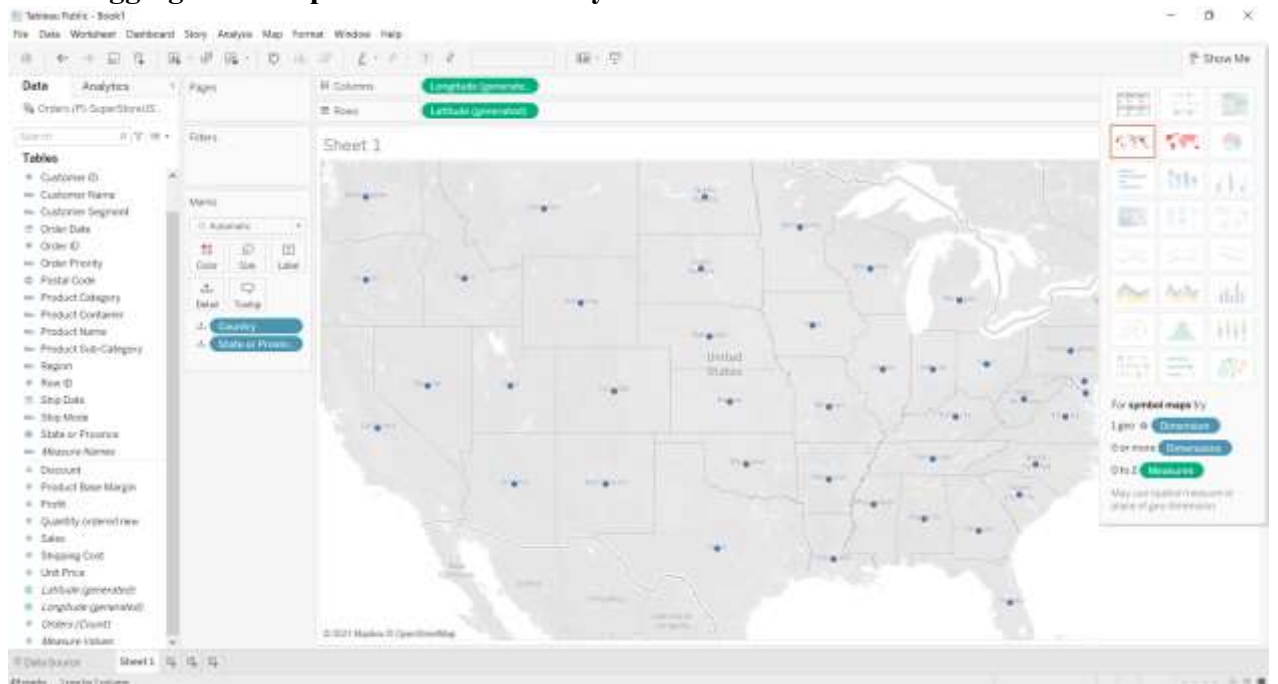
The screenshot shows the Tableau Public interface with the 'Orders' table from the 'P1-SuperStoreUS-2015' dataset loaded into the workspace. The interface includes a sidebar with 'Connections' and 'Sheets' sections. The main workspace displays a table view of the 'Orders' data with columns like Row ID, Order Priority, Discount, Unit Price, Shipping Cost, Customer ID, Customer Name, Ship Mode, Customer Segment, Product Category, Product Subcategory, and Product Name. A 'Need more data?' prompt is visible above the table.

Row ID	Order Priority	Discount	Unit Price	Shipping Cost	Customer ID	Customer Name	Ship Mode	Customer Segment	Product Category	Product Subcategory	Product Name
20947	High	0.000000	2.84	0.900	1	Bonnie Potter	Express Air	Corporate	Office Supplies	Pens & Art Supply	Washers
25228	Not Specified	0.000000	800.98	26.000	5	Ronnie Proctor	Delivery Truck	Home Office	Furniture	Chairs & Chairs	Armchairs
21776	Critical	0.060000	9.48	7.280	22	Merius Banks	Regular Air	Home Office	Furniture	Office Furnishings	Small
24344	Medium	0.080000	79.00	15.990	24	Swendolyn F Ty	Regular Air	Small Business	Furniture	Office Furnishings	Small
24345	Medium	0.080000	2.29	2.210	24	Swendolyn F Ty	Regular Air	Small Business	Office Supplies	Pens & Art Supply	Washers
24347	Medium	0.060000	3.28	4.390	24	Swendolyn F Ty	Regular Air	Small Business	Office Supplies	Pens & Art Supply	Washers
24348	Medium	0.060000	3.98	1.800	24	Swendolyn F Ty	Regular Air	Small Business	Office Supplies	Rubber Bands	Washers
18181	Critical	0.000000	4.42	4.990	25	Timothy Reese	Regular Air	Small Business	Office Supplies	Envelopes	Small
20925	Medium	0.030000	33.94	6.800	25	Timothy Reese	Regular Air	Small Business	Office Supplies	Envelopes	Small
26267	High	0.040000	2.98	1.580	26	Sarah Ramsey	Regular Air	Small Business	Office Supplies	Rubber Bands	Washers
26268	High	0.060000	115.99	2.500	26	Sarah Ramsey	Regular Air	Small Business	Technology	Telephones and C	Small

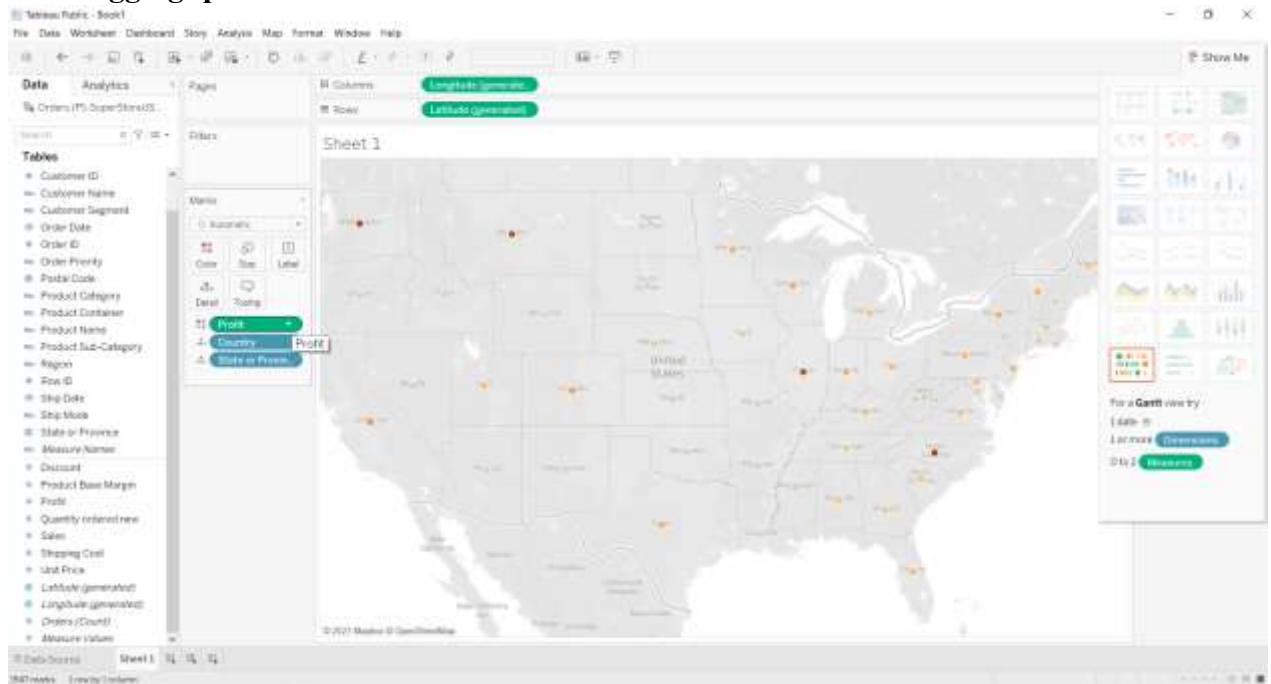
1. After going into workspace dragging 'country' into the workspace



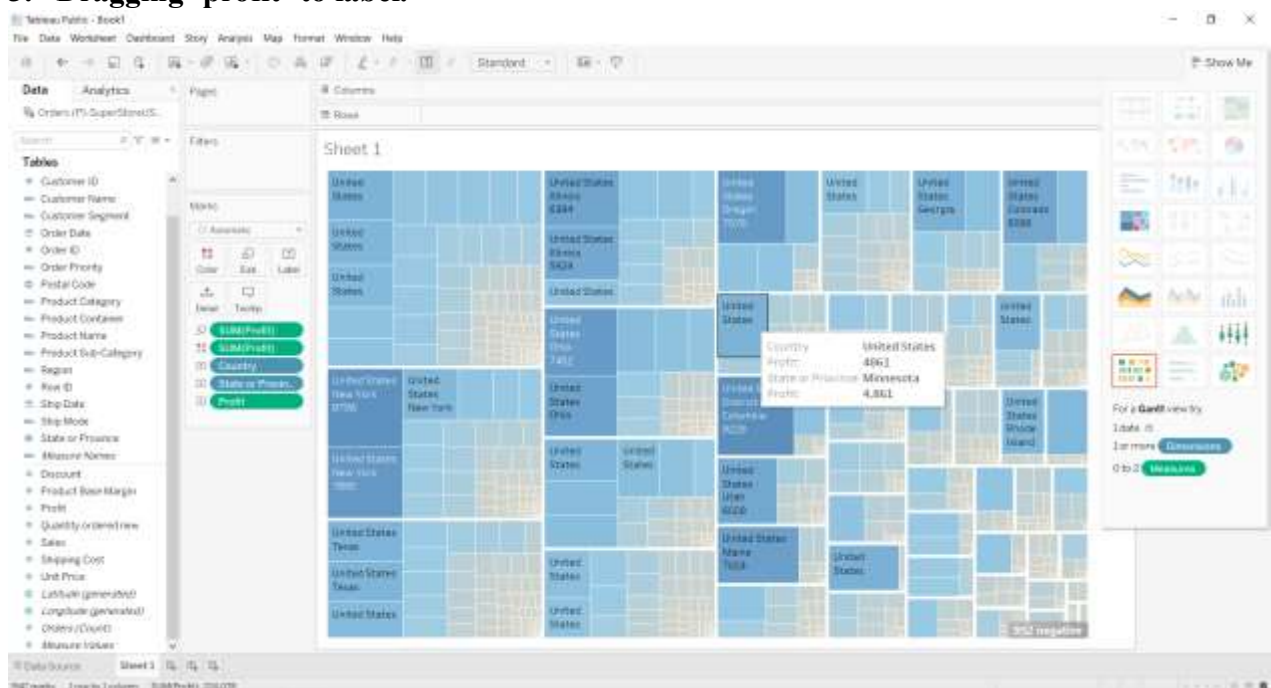
1. Dragging 'state & province' to 'country'.



2. Dragging 'profit' to colour.



3. Dragging 'profit' to label.



Feature Engineering

It is the process of using domain knowledge to extract features from raw data.

A **feature** is an attribute or property shared by all of the independent units on which analysis or prediction is to be done.

Why do we need the engineer features at all?

We know that algorithms use some input data to produce results. But quite often, the data we've been given might not be enough for designing a learning model. That's where the power of feature engineering comes into play.

Feature engineering has two goals primarily:

1. Preparing the proper input dataset, compatible with the learning algorithm requirements.
2. Improving the performance of learning models.

The feature engineering process:

1. Brainstorming or testing features;
2. Deciding what features to create;
3. Creating features;
4. Checking how the features work with your model;
5. Improving your features if needed;
6. Go back to brainstorming/creating more features until the work is done.

ETL:

Is a type of data integration that refers to the three steps (**extract, transform, load**) used to blend data from multiple sources?

ETL includes:

1. Null value elimination using east null method.
2. Handling missing values.
3. Need to normalise data.
4. Elimination of some features as too many features can lead to over fitting.
 - a. It includes bias & variance.
5. Converting string data to numeric data.

Singular Value Decomposition (SVD)

SVD is a factorization of a real or complex matrix that generalizes the eigen decomposition of a square normal matrix to any matrix via an extension of the polar decomposition.

- ❖ SVD is the decomposition of a matrix A into 3 matrices – U, S, and V.
- ❖ S is the diagonal matrix of singular values. Think of singular values as the importance values of different features in the matrix.
- ❖ The rank of a matrix is a measure of the unique information stored in a matrix. Higher the rank, more the information.

- ❖ Eigenvectors of a matrix are directions of maximum spread or variance of data.

What are Eigenvalues & Eigenvectors?

Eigenvectors are the vectors which when multiplied by a matrix (linear combination or transformation) results in another **vector having same direction** but scaled (hence scalar multiple) in forward or reverse direction by a **magnitude of the scalar multiple** which can be termed as **Eigenvalue**.

In simpler words, **eigenvalue** can be seen as the **scaling factor** for **eigenvectors**.

Here is the formula for what is called *eigenequation*: $Ax = \lambda x$

In the above equation, the **matrix A** acts on the **vector x** and the outcome is another **vector Ax** having **same direction as original vector x** but scaled / shrunk in forward or reverse direction by a magnitude of scalar multiple, λ . The **vector x** is called as **eigenvector of A** and λ is called its **eigenvalue**.

Let's understand what pictorially what happens when a matrix A acts on a vector x.
(Note that the new vector Ax has different direction than vector x.)

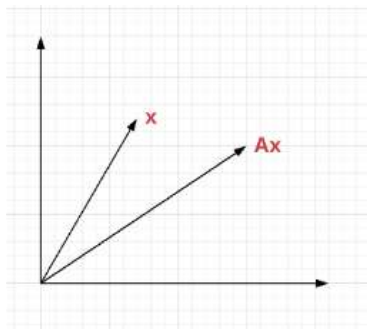


Fig 1. Matrix A acts on x resulting in another vector Ax

When the **matrix multiplication with vector** results in **another vector in the same / opposite direction** but scaled in forward / reverse direction by a magnitude of **scalar multiple** or **eigenvalue** (λ), then the vector is called as **eigenvector** of that matrix.

Here is the diagram representing the eigenvector x of matrix A because the vector Ax is in the same / opposite direction of x:

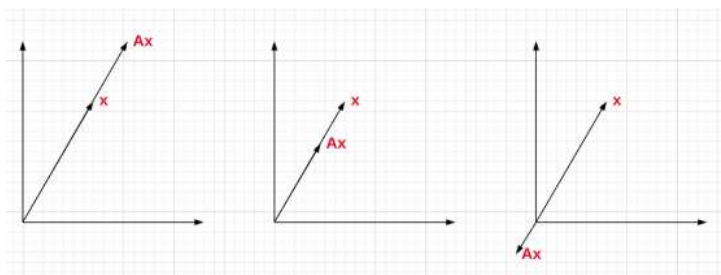


Fig 2. x is eigenvector of A

Here is some information on the value of eigenvalues:

- $\lambda \in \mathbf{R}, \lambda > 0$: v and Av point in same direction
- $\lambda \in \mathbf{R}, \lambda < 0$: v and Av point in opposite directions
- $\lambda \in \mathbf{R}, |\lambda| < 1$: Av smaller than v
- $\lambda \in \mathbf{R}, |\lambda| > 1$: Av larger than v

How to Calculate Eigenvector & Eigenvalue?

Steps to calculate the eigenvalue and eigenvector of any matrix A:

1. Calculate one or more eigenvalues depending upon number of dimensions of square matrix.
2. Determine the corresponding eigenvectors.

For calculating the eigenvalues, one needs to solve the following equation:

$$Ax = \lambda x \quad Ax - \lambda x = 0 \quad (A - \lambda I)x = 0 \quad Ax = \lambda x \quad Ax - \lambda x = 0 \quad (A - \lambda I)x = 0$$

For non-zero eigenvector, the eigenvalues can be determined by solving the following equation:

$$A - \lambda I = 0 \quad A - \lambda I = 0$$

In above equation, I is the identity matrix and λ is eigenvalue.

Once eigenvalues are determined, eigenvectors are determined by solving the equation $(A - \lambda I)x = 0$

When to use Eigenvalues & Eigenvectors?

Whenever there is a complex system having large number of dimensions with a large number of data, eigenvectors and eigenvalues concepts help in transforming the data in a set of most important dimensions (principal components). This will result in processing the data in a faster manner.

SVD and its applications in PCA (Principal Component Analysis):

In the SVD ($A = U\Sigma V^T$), we know that V is the eigenvector of the Covariance Matrix while the eigenvalues of it (λ) are hidden in Singular Values (σ).

$$A^T A = (n - 1)S = V\Sigma^2 V^T$$

$$S = V \left(\frac{\Sigma^2}{n-1} \right) V^T$$

$$\lambda = \frac{\sigma^2}{n-1}$$

The relationship between the Singular values of A and the eigenvalues of the covariance matrix of A

Since n is constant over both the cases, the Principal Components of Data Matrix is the right singular vectors (V) of the given matrix in the order of the Singular Values.

For a matrix X , the k^{th} Principal Component is the right singular vector of the covariance matrix of X corresponding to the k^{th} largest singular value.

Thus, we have come across a more generic way of representing any matrices, Singular Value Decomposition and its contribution in moulding up Principal Component Analysis which is a sophisticated method of extracting important features.

Eigen Value Decomposition

Eigen decomposition or sometimes spectral decomposition is the factorization of a matrix into a canonical form, whereby the matrix is represented in terms of its eigenvalues and eigenvectors.

Only diagonalizable matrices can be factorized in this way.

Comparison between SVD & Eigen Value Decomposition:

SVD	Eigen Value Decomposition
The vectors in the matrices are orthonormal, so they do represent rotations (and possibly flips).	Matrix are not necessarily orthogonal, so the change of basis isn't a simple rotation
The nondiagonal matrices are not necessarily the inverse of one another. They are usually not related to each other at all.	The nondiagonal matrices are inverses of each other.
The entries in the diagonal matrix are all real and nonnegative.	The entries can be any complex number, negative, positive, imaginary, whatever.
The SVD always exists for any sort of rectangular or square matrix.	Can only exists for square matrices, and even among square matrices sometimes it doesn't exist.

Data Exploration Techniques

Data exploration:

It refers to the initial step in data analysis in which data analysts use data visualization and statistical techniques to describe dataset characterizations, such as size, quantity, and accuracy, in order to better understand the nature of the data.

Data exploration techniques:

Below are a few data exploration techniques:

1. *Frequency Count*

Frequency count is finding how frequent individual values occur in column.

2. *Unique value count*

One of the first things which can be useful during data exploration is to see how many unique values are there in categorical columns. This gives an idea of what is the data about.

3. *Pareto Analysis*

Pareto analysis is a creative way of focusing on what is important.

Pareto 80–20 rule can be effectively used in data exploration.

4. *Variance*

Variance gives a good indication how the values are spread. When it comes to analysing numeric values, some basic information such as minimum, maximum and variance are very useful.

5. *Histogram*

It gives information on the range of values in which most of the values fall. It also gives information on whether there is any skew in data.

6. *Correlation Heat-map between all numeric columns*

The term correlation refers to a mutual relationship or association between two things. It is useful to express something in terms of its relationship with others. Finding correlation is very useful in data exploration, as it gives an idea on how the columns are related to each other.

Basic objectives for Exploratory Data Analysis:

1. Identifying type of data.
2. Selecting appropriate descriptive statistics.
3. Recognizing when to use robust statistics.
4. Choosing the correct type of plot.
5. Explaining the limitations of each type of plot.

Types of data:

1. *Categorical data*

Categorical data represents characteristics.

Therefore, it can represent things like a person's gender, language etc.

Categorical data can also take on numerical values.

Example: 1 for female and 0 for male.

a. Nominal data

Nominal values represent discrete units and are used to label variables, that have no quantitative value. Nominal data that has no order.

Example:

Are you married?	What languages do you speak?
<input type="radio"/> Yes	<input type="radio"/> English
<input type="radio"/> No	<input type="radio"/> French
	<input type="radio"/> German
	<input type="radio"/> Spanish

b. Ordinal data

Ordinal values represent discrete and ordered units. It is therefore nearly the same as nominal data, except that it's ordering matters.

Example:

What Is Your Educational Background?

☐ 1 - Elementary

☐ 2 - High School

☐ 3 - Undergraduate

☐ 4 - Graduate

2. Continuous data

Continuous Data represents measurements and therefore their values **can't be counted but they can be measured**.

Example:

The height of a person, which you can describe by using intervals on the real number line.

a. Interval data

Interval values represent **ordered units that have the same difference**. Therefore, we speak of interval data when we have a variable that contains numeric values that are ordered and where we know the exact differences between the values.

Example:

A feature that contains temperature of a given place like you can see below:

Temperature?

☐ -10

☐ -5

☐ 0

☐ +5

☐ +10

☐ +15

b. Ratio data

Ratio values are also ordered units that have the same difference. Ratio values are the same as interval values, with the difference that they do have an absolute zero.

Example:

Height, weight, length etc.

Length (inch)?

☐ 0

☒ 5

☐ 10

☐ 15

Why Data Types are important?

Datatypes are an important concept because statistical methods can only be used with certain data types. We have to analyse continuous data differently than categorical data otherwise it would result in a wrong analysis. Therefore, knowing the types of data you are dealing with, enables you to choose the correct method of analysis.

Descriptive statistics:

1. Measures of Central Tendency

- a. Mean
- b. Median
- c. Mode

Locates the distribution by various *points*.

Use this when you want to show how an average or most commonly indicate response.

1. Measures of Variation

- a. Range
- b. Variance
- c. Standard Deviation

Identifies the spread of scores by stating intervals.

Range = High/Low points

Variance or Standard Deviation = difference between observed score and mean.

Use this when you want to show how "spread out" the data is.

It is helpful to know when your data are so spread out that it affects the mean.

Relationship Between Two Variables

Statistical relationships between variables rely on notions of **correlation** and **regression**.

1. Correlation:

The tests are used to determine how strongly the scores of two variables are associated or correlated with each other.

Correlation is measured using values between +1.0 and -1.0.

Correlations close to 0 indicate little or no relationship between two variables, while correlations close to +1.0 (or -1.0) indicate strong positive (or negative) relationships.

Correlation denotes positive or negative association between variables.

Two variables are ***positively associated*** when larger values of one tend to be accompanied by larger values of the other.

The variables are ***negatively associated*** when larger values of one tend to be accompanied by smaller values of the other

Example of a strong positive correlation would be the correlation between age and job experience. Typically, the longer people are alive, the more job experience they might have.

Example of a strong negative correlation might occur between the strength of people's party affiliations and their willingness to vote for a candidate from different parties. In many elections, Democrats are unlikely to vote for Republicans, and vice versa.

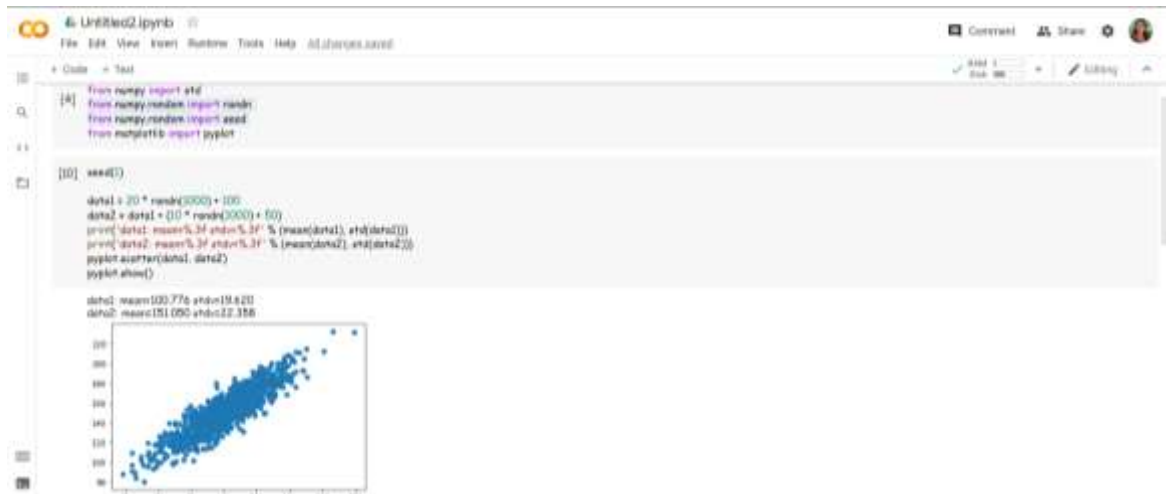
1. Pearson Correlation: It evaluates the linear relationship between two continuous variables.

2. Spearman correlation: It evaluates the monotonic relationship. The Spearman correlation coefficient is based on the ranked values for each variable rather than the raw data.

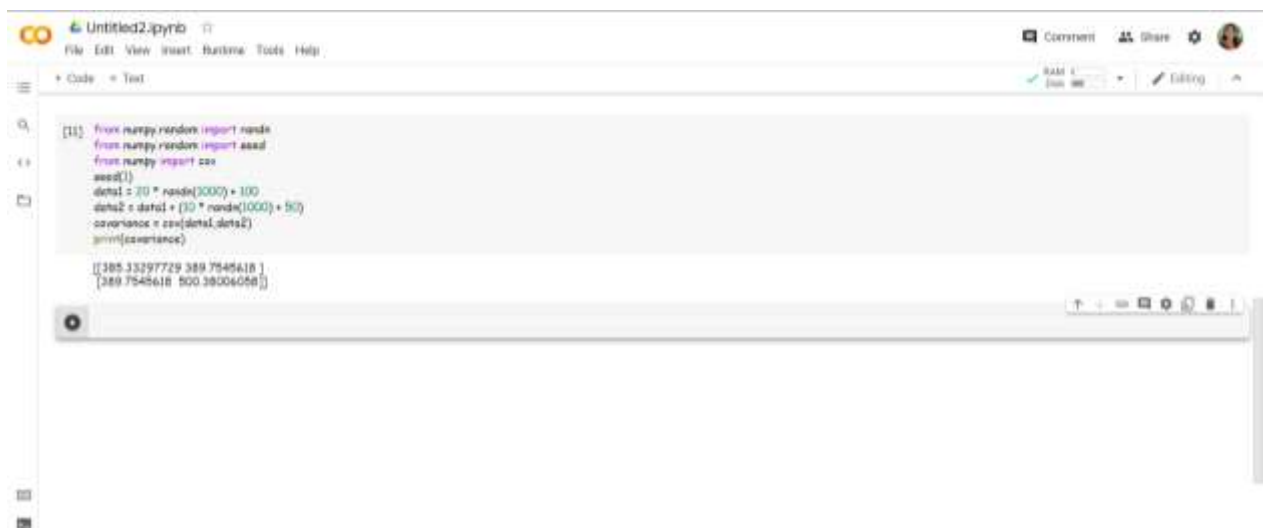
Both Pearson and Spearman are used for measuring the correlation but the difference between them lies in the kind of analysis we want!

Examples:

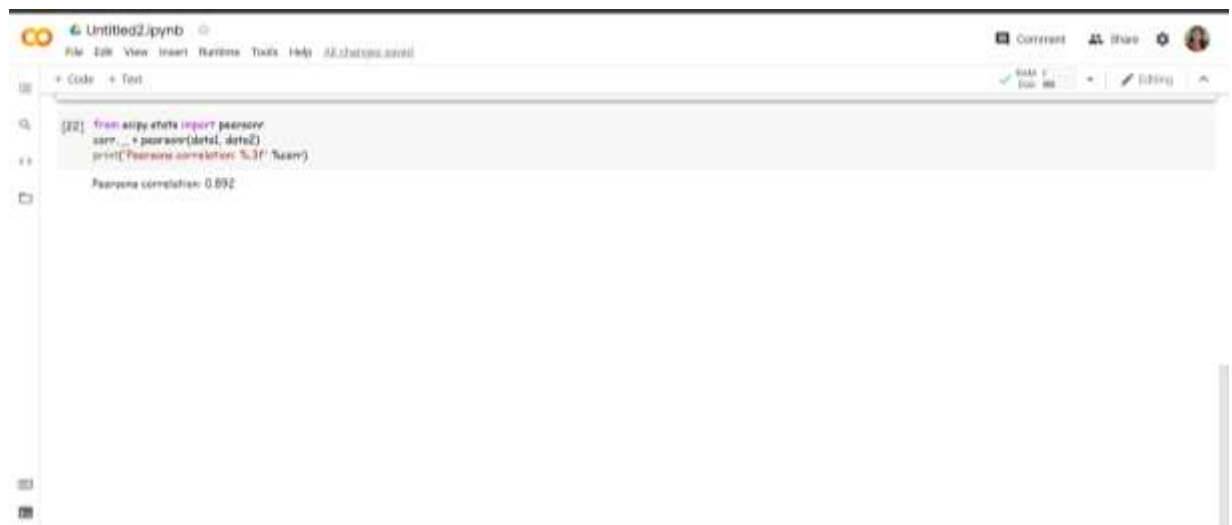
1. Finding mean, std deviation & plotting scatter plot



2. Calculate covariance



3. Calculate Pearson's correlation coefficient



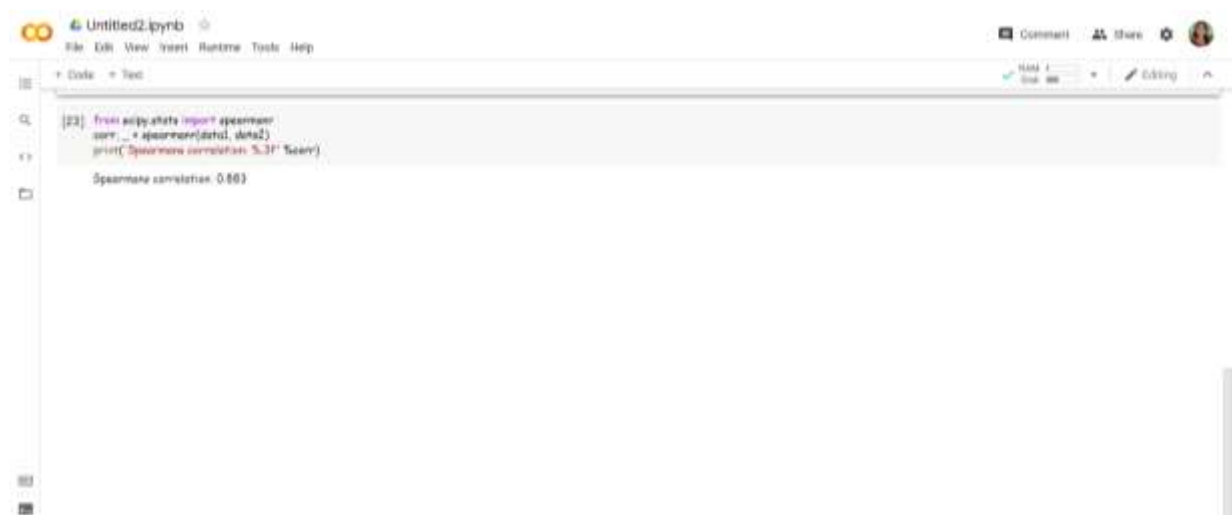
The screenshot shows a Jupyter Notebook interface with a file named 'Untitled2.ipynb'. The code cell contains the following Python code:

```
[22]: from scipy.stats import pearsonr
corr, _ = pearsonr(data1, data2)
print('Pearsons correlation: %.3f' % corr)
```

The output of the code is displayed below the code cell:

```
Pearsons correlation: 0.892
```

4. Calculate Spearman's correlation coefficient



The screenshot shows a Jupyter Notebook interface with a file named 'Untitled2.ipynb'. The code cell contains the following Python code:

```
[23]: from scipy.stats import spearmanr
corr, _ = spearmanr(data1, data2)
print('Spearman correlation: %.3f' % corr)
```

The output of the code is displayed below the code cell:

```
Spearman correlation: 0.863
```

Violin Plot

Is a method of plotting numeric data and can be considered a combination of the box plot with a kernel density plot.

In the violin plot, we can find the same information as in the box plots like:

1. Median (a white dot on the violin plot)
2. Interquartile range (the black bar at the mid of violin)
3. The lower/upper adjacent values (the black lines stretched from the bar); defined as first quartile + 1.5 IQR and third quartile + 1.5 IQR respectively.

Example:

```
Untitled2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
RAM 1 (64. MB) Editing

[27] import matplotlib.pyplot as plt
import seaborn as sns

[28] cars=sns.load_dataset('mpg')
cars.shape

(392, 9)

[29] sns.set_style('whitegrid')

[30] cars.cylinders.value_counts()

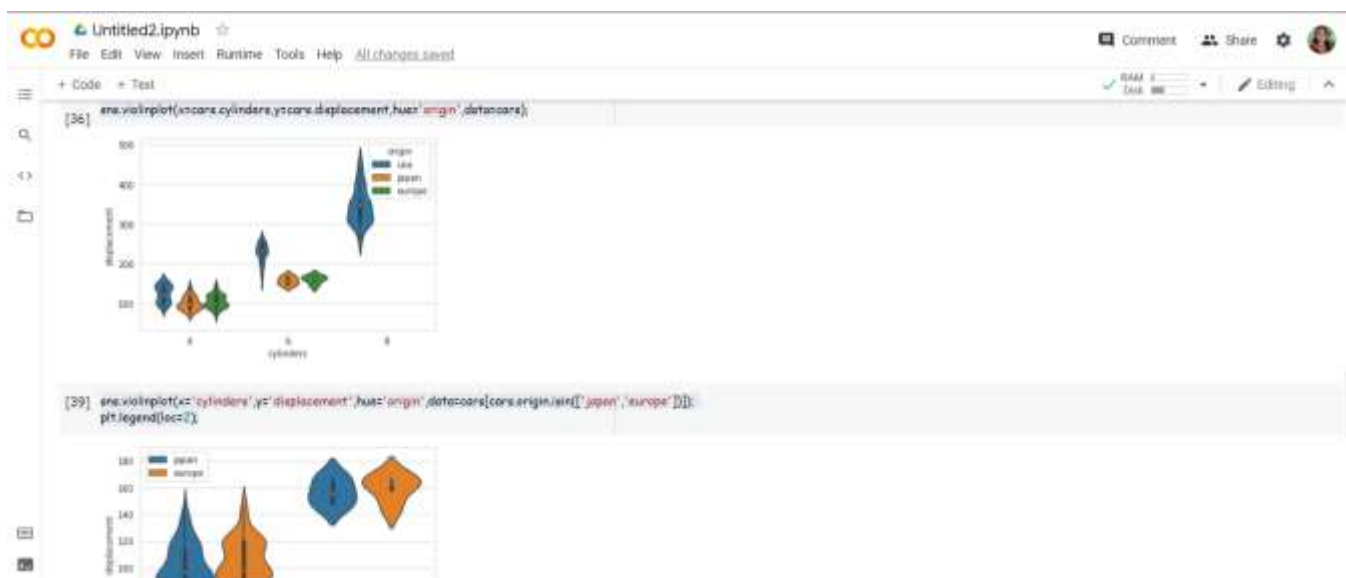
4    199
8    103
6     83
3     4
5     3
Name: cylinders, dtype: int64

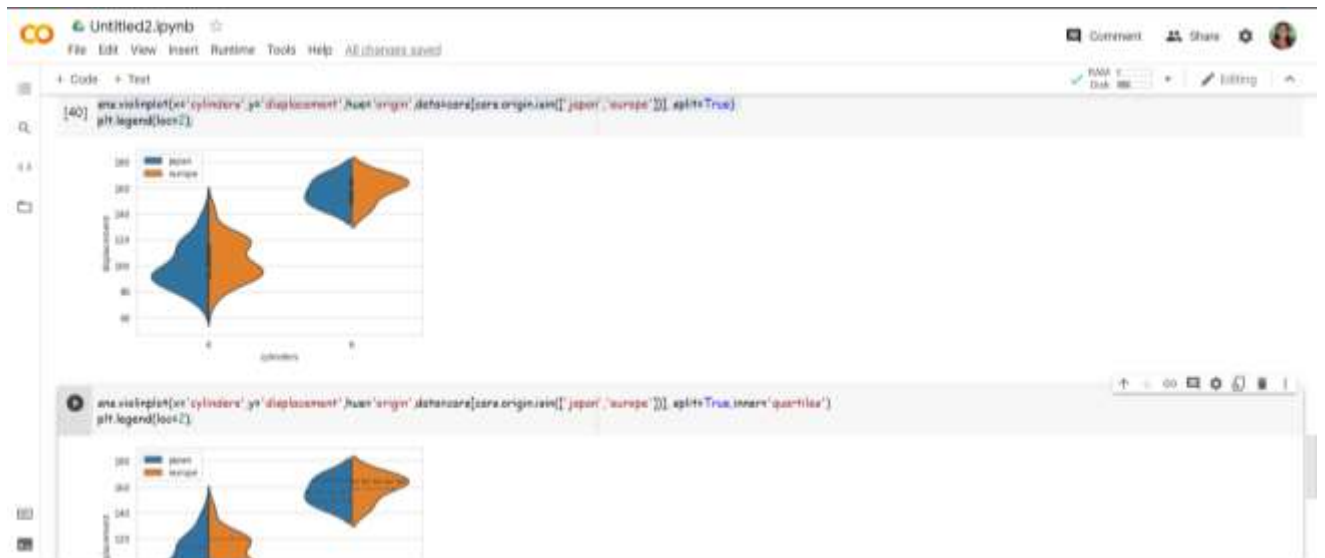
[33] cars=cars[cars.cylinders.isin([4,6,8])]

[34] sns.violinplot(cars.displacement)
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be 'data', and passing other variables as positional arguments is deprecated.







Data Science Terminologies

Below are a few Data Science Terminologies:

1. Autoregressive

Statistical model is autoregressive if it predicts future values based on past values. For example, an autoregressive model might seek to predict a stock's future prices based on its past performance.

2. Covariance

Is a statistical technique used for determining the relationship between the movement of two random variables. In short, how much two random variables change together. Positive covariance indicates that higher than average values of one variable tend to get paired with higher than the average values of the other variable.

3. Correlation

Is also a statistical technique that determines how the change of one variable related to another variable affects the relationship. In short, it defines the degree of relation between two variables. There exist three types of correlations - positive and negative, and zero correlations. A positive correlation is a relationship between the variables, where two variables move in the same direction. If one variable increases, the other also increases. If one variable decreases, the other also decreases. In a negative correlation, when one variable value decreases, the other variable value increases and vice

4. Auto correlation

Is a characteristic of data which shows the degree of similarity between the values of the same variables over successive time intervals. This post explains what autocorrelation is, types of autocorrelation - positive and negative autocorrelation, as well as how to diagnose and test for auto correlation. Autocorrelation, also known as serial correlation, is the correlation of a signal with a delayed copy of itself as a function of delay. It is often used in signal processing for analysing functions or series of values, such as time domain signals.

5. Bias

Can come from human sources because they use unrepresentative data sets, leading questions in surveys and biased reporting and measurements. Often bias goes unnoticed until you've made some decision based on your data, such as building a predictive model that turns out to be wrong.

6. Variance

Is a numerical value that shows how widely the individual figures in a set of data distribute themselves about the mean and hence describes the difference of each value in the dataset from the mean value. So, if we have zero variance in a dataset, we can state that all the values in it are identical.

7. Bagging

Is an ensemble technique used when our goal is to reduce the variance in a decision tree model. The concept behind bagging is to create multiple subsets of data from a training sample that is randomly selected and replaced, each dataset in the subset is used to prepare its decision trees, so we get a different set of models.

8. Boosting

Is also an ensemble technique for creating a set of predictors, if a given input dataset is misclassified, then its weight increases so that a future hypothesis is more likely to classify it correctly, consolidating the whole set and ultimately, we can say making weak model to more effective model.

9. Multicollinearity

Multicollinearity occurs when two or more independent variables are highly correlated with one another in a regression model. This means that an independent variable can be predicted from another independent variable in a regression model.

10. Outliers

Are data points that don't belong to a certain population, an abnormal observation that lies far away from other data values.

11. Detection techniques are:

1. Using standard Deviation
2. Using Boxplots
3. Using Violin Plots
4. Using Scatter Plots

12. Box plots

Box plots are a graphical depiction of numerical data through their quantiles, lower and upper whiskers as the boundaries of the data distribution.

13. Scatter Plot

Is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data, data is displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis.

SweetViz

Installing Sweetviz

Like any other python library, we can install Sweetviz by using the pip install command given below.

```
pip install sweetviz  
import sweetviz
```

we studied EDA to explore the data and to find missing values, correlations but there are some libraries that help us do EDA faster and in easier mode
one of them is Sweetviz(open source Python library)

Purpose of Sweetviz

- **finding out data types**
- **missing information**
- **distribution of values**
- **correlations**

How it works:

It takes pandas dataframes and generates HTML report.

Also Sweetviz guesses the datatype of each column like numerical or categorical
we are taking titanic dataset as an example and we explore on that using sweetviz:

Analyzing Dataset

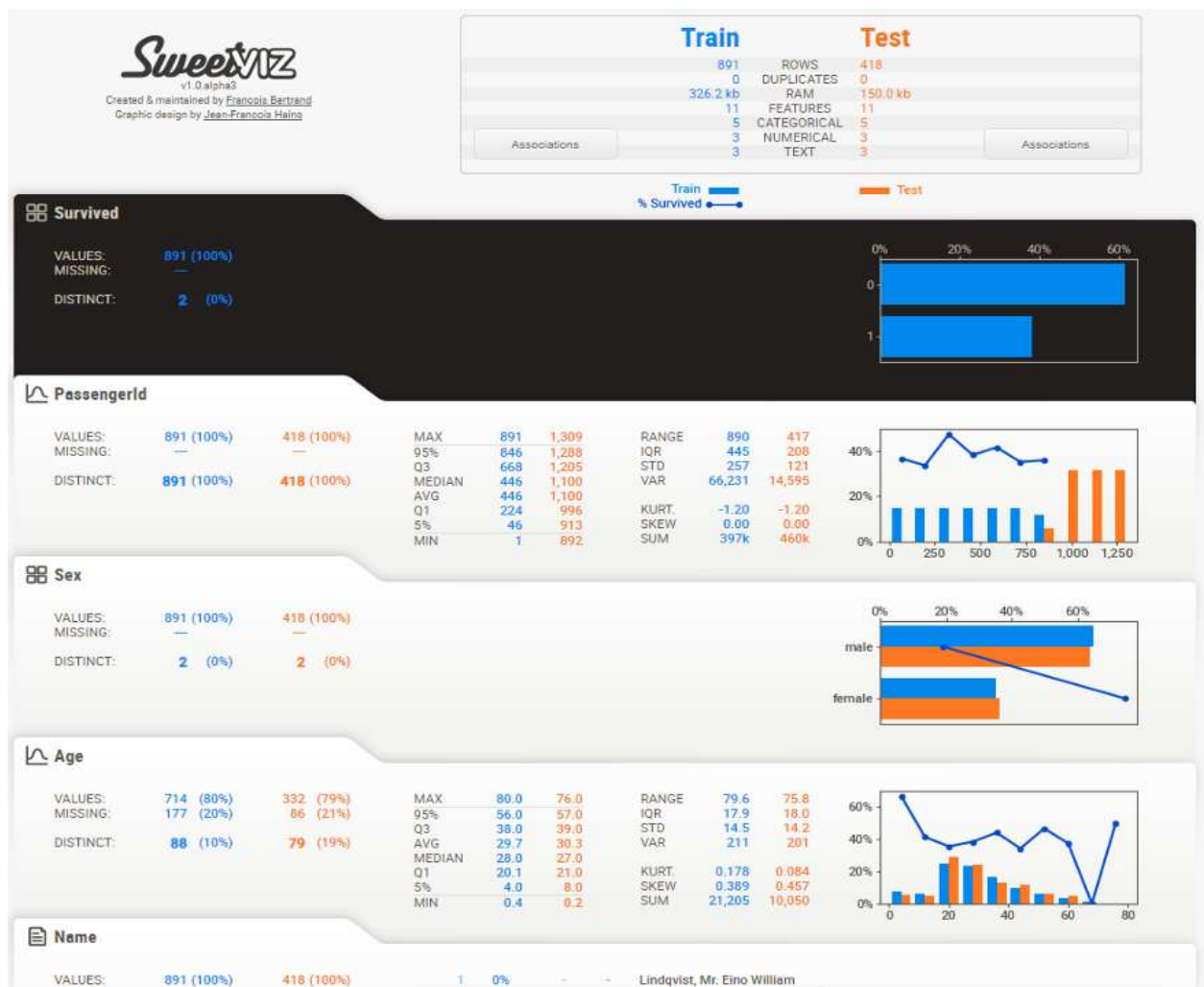
In this article, we use titanic dataset used. First, we need to load the using pandas.

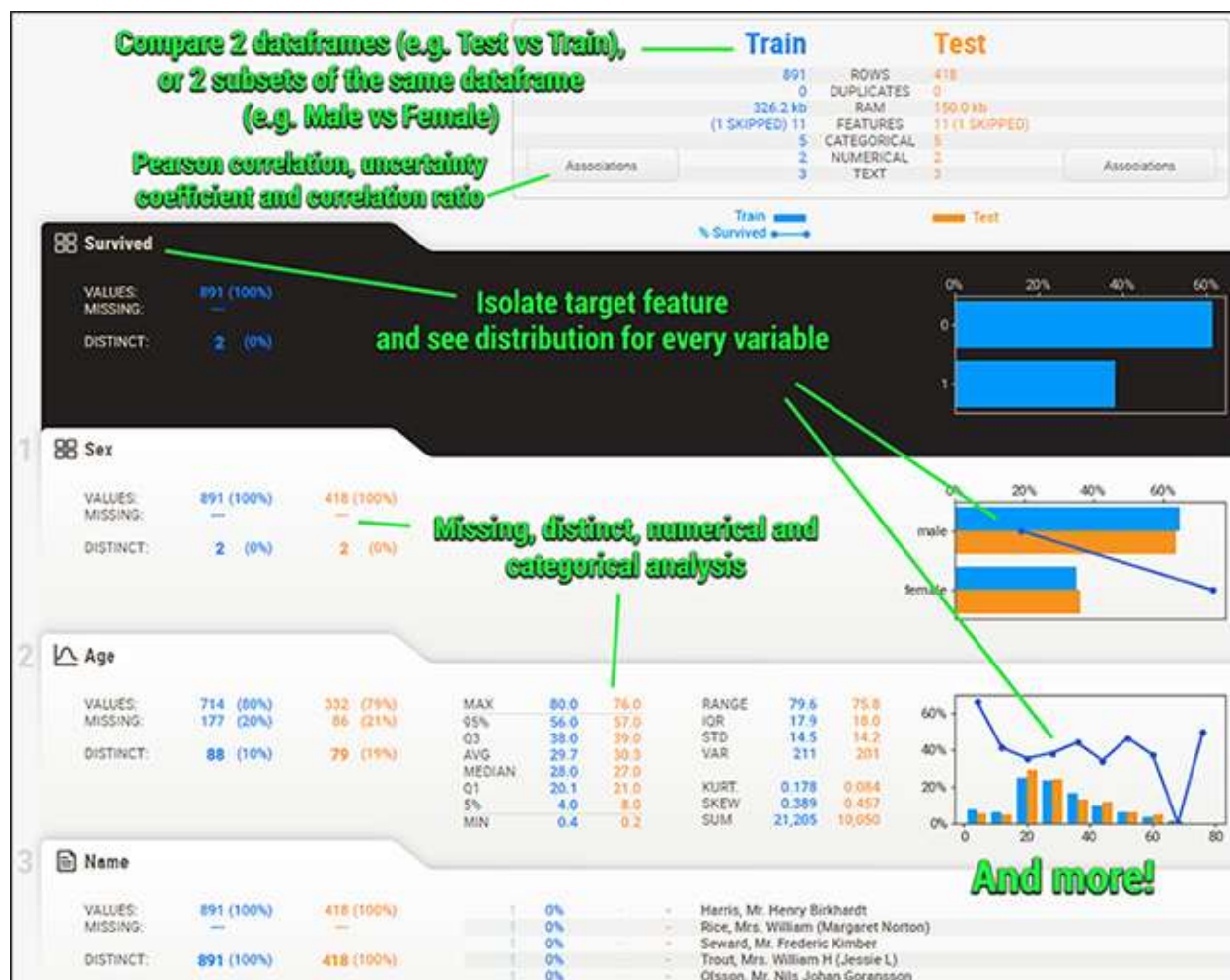
import pandas as pd

Analyzing the Titanic dataset

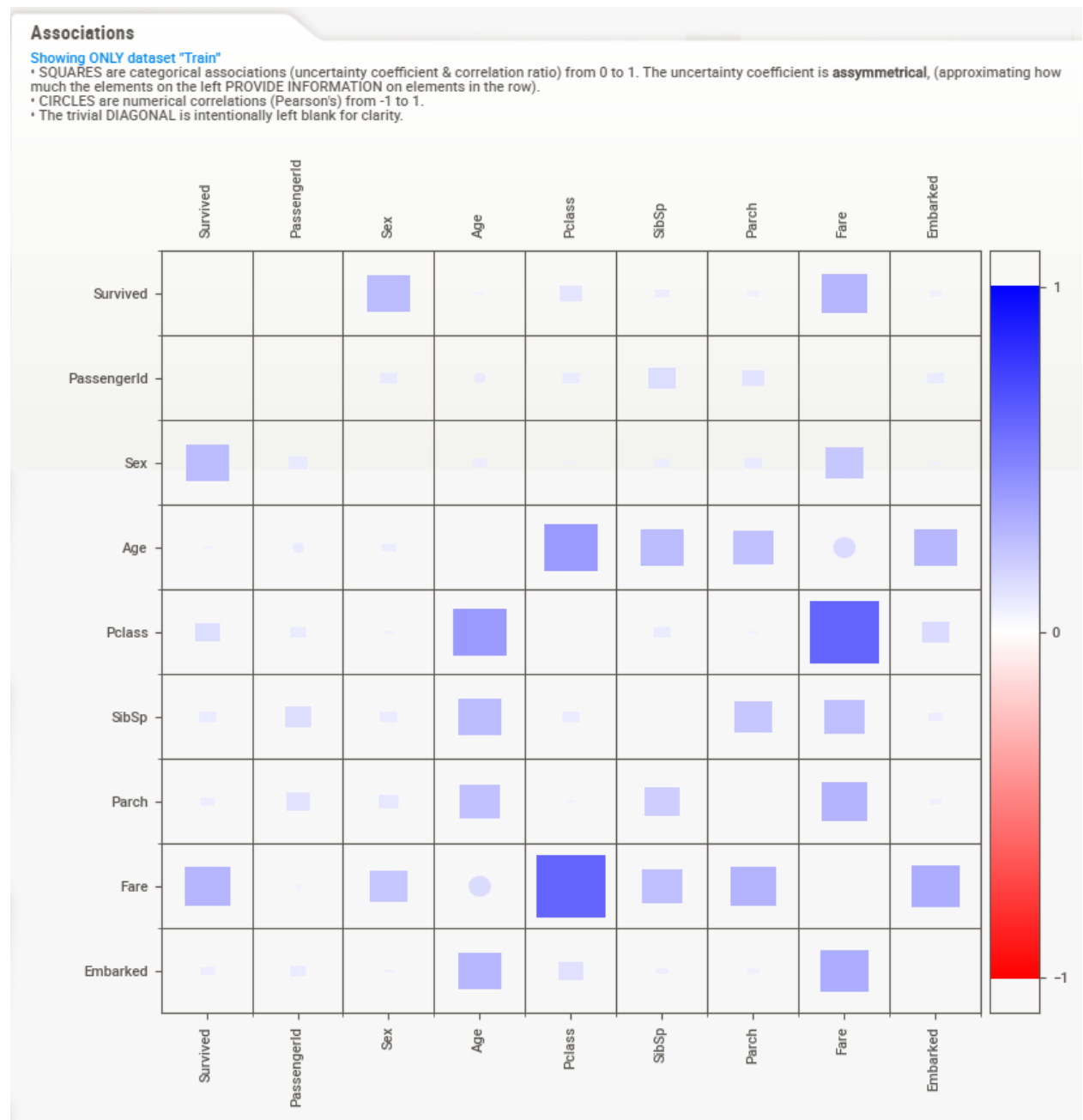
```
import sweetviz
import pandas as pd
train = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")
```

**We have 2 dataframes (train and test), and we would like to analyze the target value
“Survived”**





If we hover on association button it shows this result for that dataframe:



Sweetviz has a function named :

1. **Analyze()**
2. **compare()**
3. **intra-compare()**

Let's Analyze our dataset using the command compare given below:

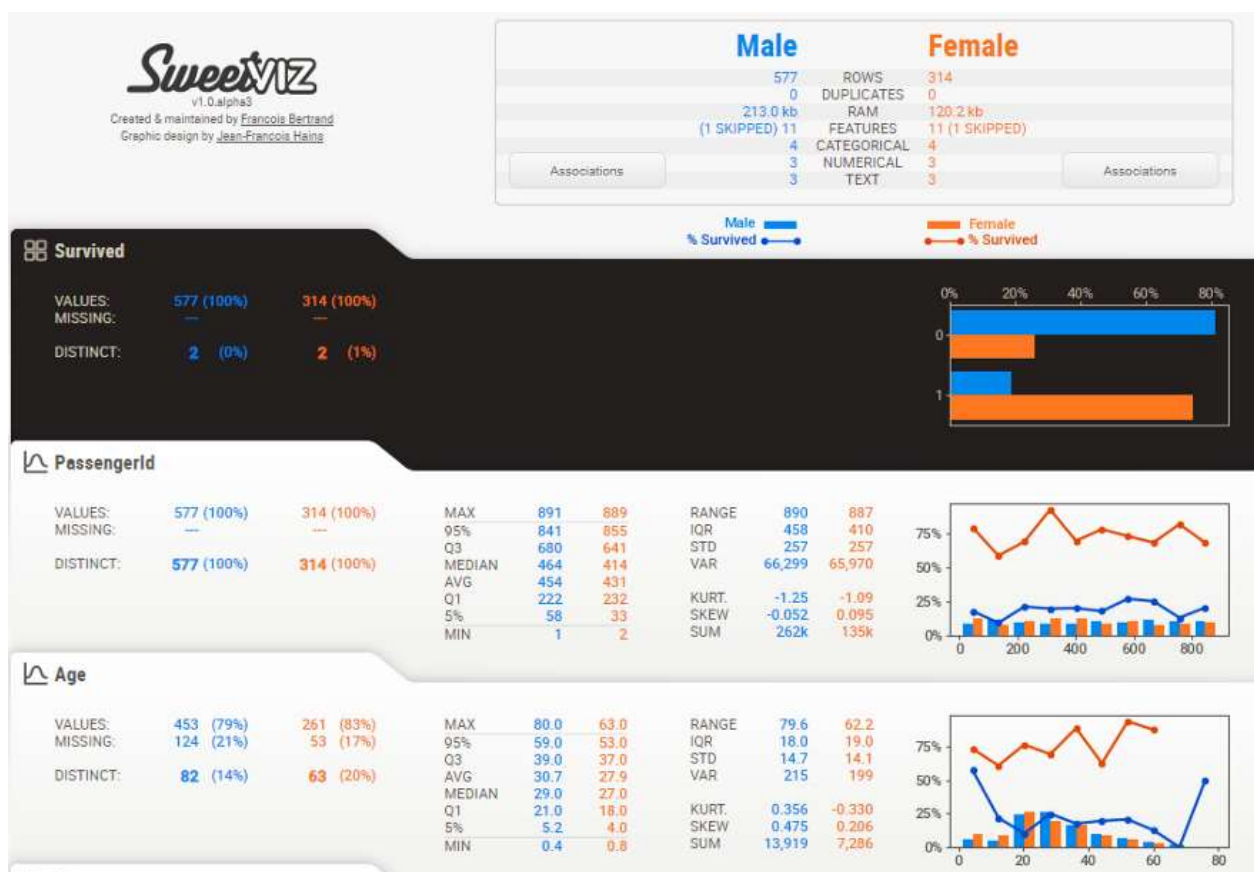
Compare() function of Sweetviz is used for comparison of the dataset. The commands given below will create and compare our test and train dataset.

We can generate a report with this line of code:

```
my_report = sweetviz.compare([train, "Train"], [test, "Test"], "Survived")
```

To get the output, we type the show_html() command:

```
my_report.show_html("Report.html")
```



Tableau

Tableau is best suited for scenarios where the data has been already prepared for visualization--in other words, where Tableau is used for actually building the visualization assets, and not for pre-processing the data. On the other hand, Tableau does not seem to shine in situations where complex data pre-processing is needed--think, for instance, of complex joins of large tables.

Tableau Public products (for free) work well during the product evaluation phase, as well as for sharing data with partners and the community for no charge. A key question is what security the customer wants with the data, and how frequently the underlying data changes. If the data does not change frequently, and the user is not concerned about securing the data, then Public is a good choice.

Tableau Public consists of a *free* downloadable version of Tableau desktop to explore and visualize data, and a *free* cloud platform to host, share and embed interactive visualizations.

Advantages and disadvantages of Tableau:

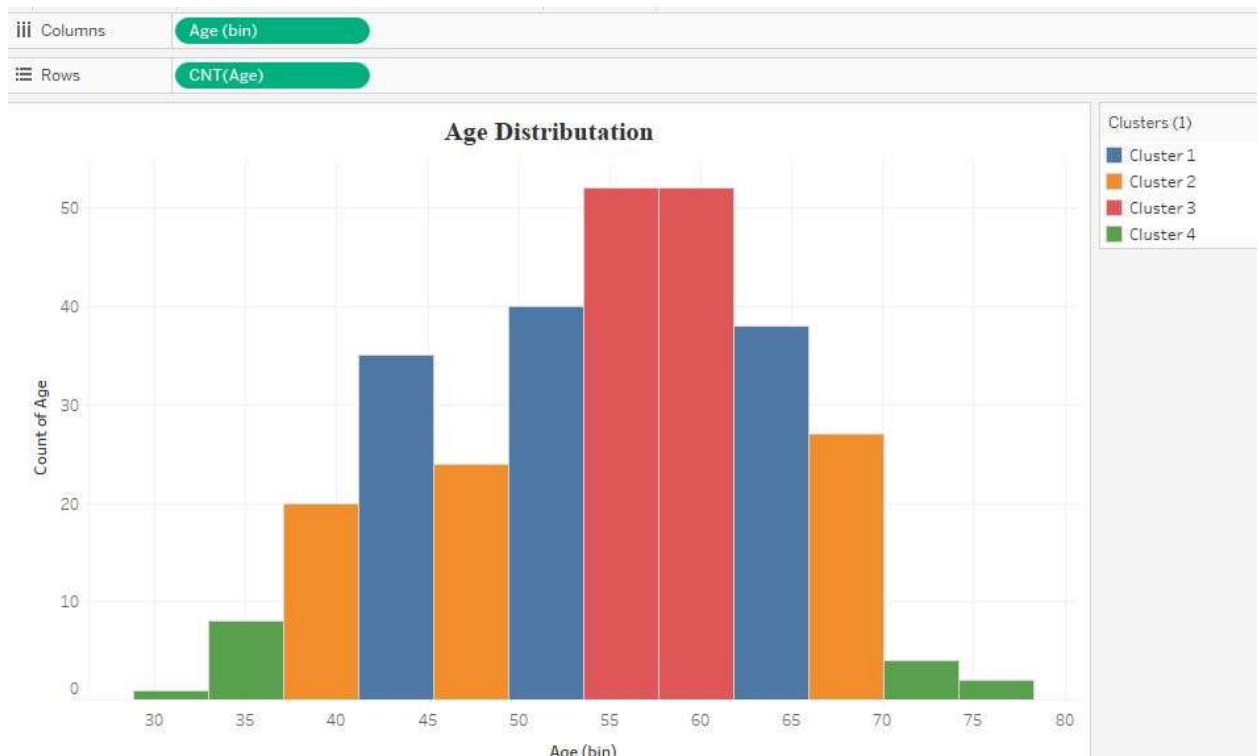
- Connectivity: TDE, Access, Excel, Text File (you can not connect Tableau Public to any of the databases)
- Distribution: Cloud (you can not save Tableau Public workbooks locally)
- Automation: Not available (you can not schedule Tableau Public reports on server)
- Security: None – (your data is accessible by anyone on the Internet. A dashboard on Tableau public will not ask for any username or password, hence we can not have user-level security on Tableau Public Server)
- Data Limit: Tableau Public has a limit of 10 million rows of data that is allowed in any single connection.

We use tableau for making charts and dashboards

We can use different sheet for making charts by using bar plot, box plot, simple or auto chart,

It shows the comparison of data like we take one column with sum or count of other column and shows the result by using Tableau application:

As an example if we consider heart dataset by using tableau we can analyze the heart rate is high in which range so basically it takes the count of every range and compares.



How To Frame right question for your data

Before going to any project we have to understand our data first and from knowing our data we need to know right question for it:

to generate meaningful questions that could be answered using data

we have to follow these tips:

1. Exploring the data

before analysing the data we should understand our data what is the data,

what is our scope, what is the type of our data (numerical or categorical)

and what is the relation between data.

2. we should determine the type of problem :

according to our dataset we should decide the category for solving our problem

like it should solve by using (descriptive analytics, predictive analytics, or prescriptive analytics)

3. We have to know what is our limitation in this problem:

We should decide to work with which team like if its related to health we should contact with doctore not engineer or

any business man ,to guide us for collecting accurate data.

Descriptive Data Analysis:

In descriptive analytics, we study the relationships between features of dataset. It could be a scatter plot, barplot, line graph,density plot, heat map, etc.

1: we use Barplot for comparison

2: Heatmap plot to study and quantify correlations

Predictive Data Analysis

we use predictive data analysis to build a model using available dataset for predictions.

and type of model to build is depend on type of target column.

If the target variable is continuous, then we use linear regression, and If the target variable is discrete, then classification will be used.

for predicitive analysis we will go through 4 stages:

1. Problem Framing

in this stage we decide on type of problem like predicting wether someone has heart disease or not.

2. Data Analysis

in this we are dong data analysis like finding missing values, finding null values, normalization, balancing of data.

3. Model Building

in this step we decide to use logistic regression or linear regression or other model building.

4. Application

its the last stage that we put machine learning model in to production.

Regression

Regression is a statistical method that attempts to determine the strength and character of the relationship between one dependent variable and a series of other variables. It certainly plays an essential role in statistics and thus in data science. In here we will go through two types of regression linear regression and logistic regression:

Linear Regression

Linear regression is a basic and commonly used type of predictive analysis. The overall idea of regression is to examine two things:

- (1) does a set of predictor variables do a good job in predicting an outcome (dependent) variable?
- (2) Which variables in particular are significant predictors of the outcome variable, and the outcome variable?

This linear estimations are used to explain the relationship between one dependent variable and one or more independent variables.

Usage of Linear Regression

Three major uses for regression analysis are

- (1) determining the strength of predictors,
- (2) forecasting an effect
- (3) trend forecasting.

First, the regression might be used to identify the strength of the effect that the independent variable(s) have on a dependent variable.

For example what is the strength of relationship between dose and effect, sales and marketing spending, or age and income.

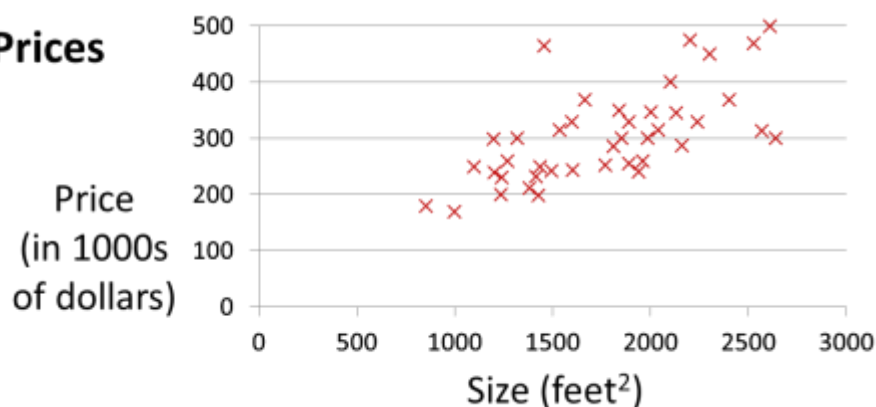
Second, it can be used to forecast effects or impact of changes. That is, the regression analysis helps us to understand how much the dependent variable changes with a change in one or more independent variables. For example , “how much additional sales income do I get for each additional \$1000 spent on marketing?”

Third, regression analysis predicts trends and future values. The regression analysis can be used to get point estimates. For example, “what will the price of gold be in 6 months?”

Example:

Linear regression shows the relationship between two variables by fitting a linear equation to observed data. One of the variables is an independent variable, while the other is a dependent variable. For example, you might want to relate the housing price with size of house in feet

Housing Prices



Types of Linear Regression

There are different types of linear regression we consider some of them here:

1. Simple linear regression

1 dependent variable and 1 independent variable Like a correlation, it determines the extent to which one independent variable predicts a dependent variable.

2. Multiple linear regression

1 dependent variable and 2 or more independent variables , It allows one to determine how well multiple independent variables predict the value of a dependent variable.

3. Logistic regression

1 dependent variable and 2 or more independent variable(s)

Linear regression with one variable or Simple Linear Regression:

The simplest form of the regression equation with one dependent and one independent variable is defined by the formula $y = c + m \cdot x$, where y = estimated dependent variable score, c = constant, m = regression coefficient, and x = score on the independent variable.

Cost function

Cost Function quantifies the error between predicted values and expected values and presents it in the form of a single real number.

Depending on the problem Cost Function can be formed in many different ways.

The purpose of Cost Function is to be either:

- **Minimized** - then returned value is usually called **cost**, **loss** or **error**. When cost function return as small number as possible.
- **Maximized** – its named **reward**. to find values of model to returned number as large as possible.

LOGISTIC REGRESSION

Currently the world seems data problem!! Why not everywhere around us you will find various data from which you can create data sets and perform analysis yes?

Lately, we can see there is lot of opportunities lying in front for data analyzers, data scientists etc. You got to dig into data and it will definitely present your results before you, but having data only is sufficient for you for predicting?? Yes? NO obviously there are lot of steps and procedure involved to make predictions one need to clean, analyze, visualize, test the data, wellbeing data scientist or analyzer is not easy you see!!.

That's not all to make prediction one must also require to plot graphs ,check what data is telling you is it enough to achieve what you want, also know about methods and yeah also the algorithms.

For example :

The data sets I have used is about Heart Disease.

```

File Edit View Run Kernel Help
logistic-regression.py
Python 3

[7]: import numpy as np
import pandas as pd
dataset = read_csv('heart.csv')
data = dataset

[8]: age sex chest pain type resting bps cholesterol fasting blood sugar resting eeg max heart rate exercise angina st slope target
0 40 1 0 140 200 0 0 170 0 0.0 1 0
1 40 0 0 160 160 0 0 150 0 1.0 2 1
2 37 1 0 130 200 0 1 90 0 0.0 1 0
3 40 0 4 130 210 0 0 100 1 1.5 2 1
4 34 1 1 130 160 0 0 120 0 0.0 1 0

[9]: data.shape
[9]: (1199, 12)


[10]: xpd.DataFrame(data.iloc[:,1:11])
ypd.DataFrame(data.iloc[:,11])

[11]: data['target'].value_counts()

[12]: 1    629
     0    570
     Name: target, dtype: int64

[13]: import matplotlib.pyplot as plt
import seaborn as sns
sns.countplot(x='target', data=data, palette='b2b')
plt.show()

```



```

File Edit View Run Kernel Help
logistic-regression.py
Python 3

[13]: import matplotlib.pyplot as plt
import seaborn as sns
sns.countplot(x='target', data=data, palette='b2b')
plt.show()

[14]: xpd.DataFrame(data.iloc[:,1:11])
ypd.DataFrame(data.iloc[:,11])

[15]: from sklearn.model_selection import train_test_split

[16]: X_train, X_test, y_train, y_test = train_test_split(
    data, data['target'], test_size=0.3, random_state=1)

[17]: from sklearn.linear_model import LogisticRegression

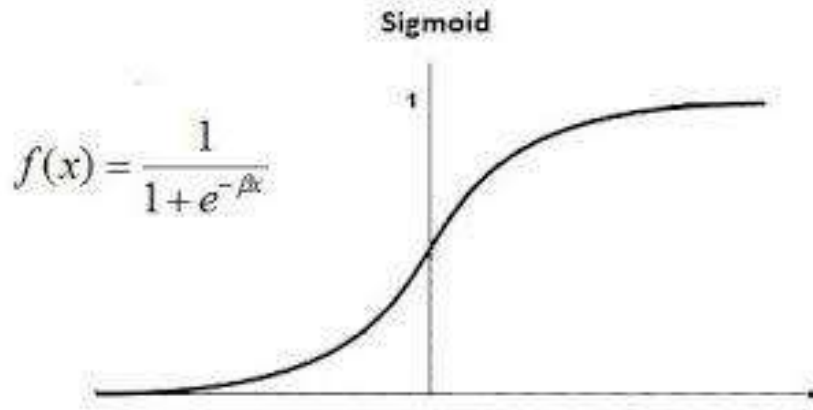
[18]: logisticRegression = LogisticRegression()

[19]: logisticRegression.fit(X_train, y_train)

```

Sigmoid Function

Also it learns a linear relationship from the data sets and then introduces a non linearity in the form of Sigmoid function (the function maps real value into another value between 0 and 1)



Real world use case of logistic regression is used

- Text editing
- Hotel booking
- Credit scoring
- Medical field.

Buliding logistic regression in python step by step:

1. Decide the dataset you want to work with.
2. Import libraries and dataset.
3. Understand your data.
4. Perform Exploratory Data analysis
5. Prepare data
6. Build logistic regression model
7. Make prediction on data set

Also Logistic regression assumes that the observations in the dataset are independent of each other. i.e the observations should not be related to each other in any way.

So I hope this was informative in knowing about logical regression.

Logistic regression assumes that the observations in the dataset are independent of each other. That is, the observations should not come from repeated measurements of the same individual or be related to each other in any way.

Gretl:

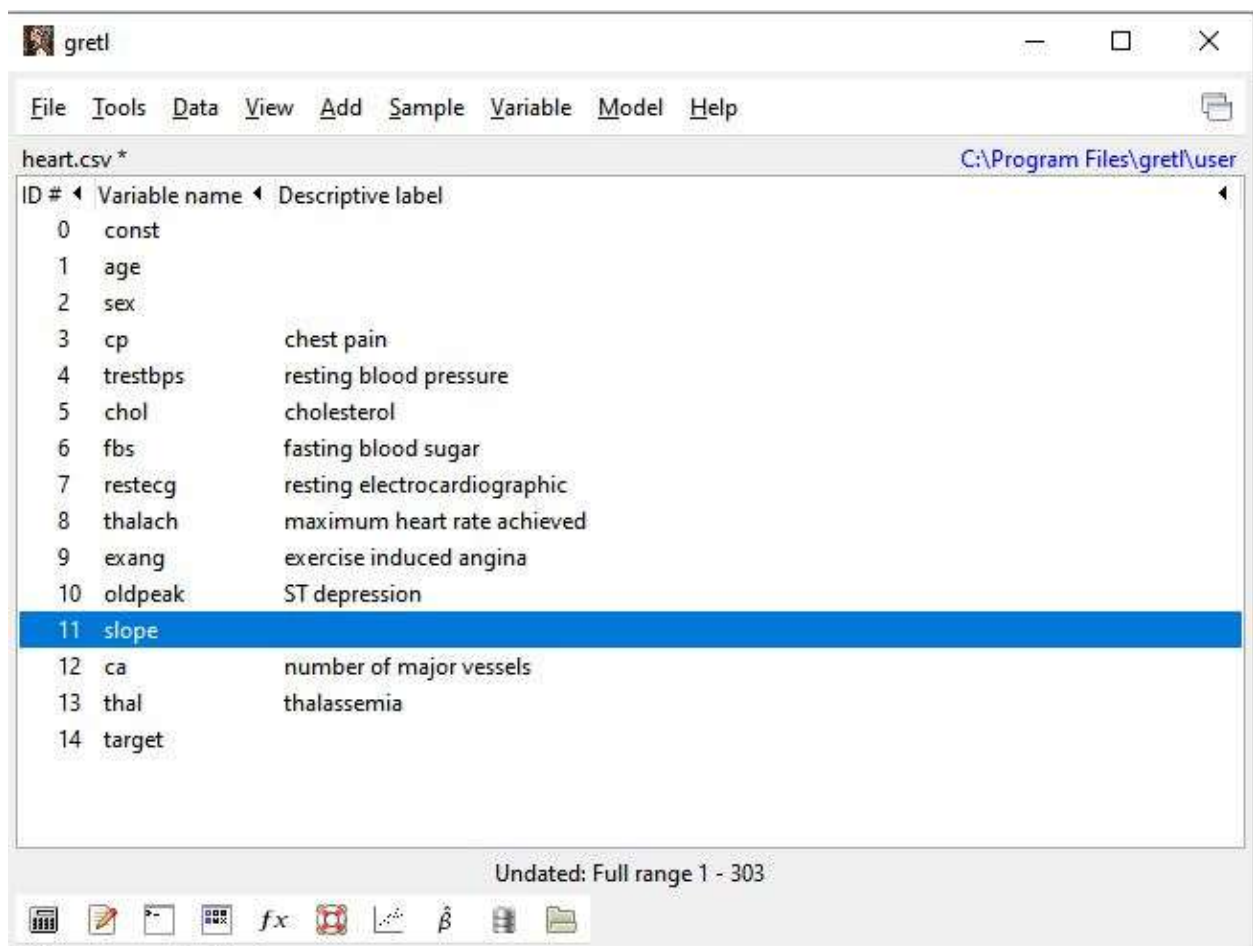
gretl is an acronym for Gnu Regression Econometrics and Time-series Library it is free econometrics software it has an easy Graphical User Interface (GUI) it runs least-squares, maximum-likelihood, systems estimators. it admits scripts (sequence of commands saved in a file)

Features of gretl

- Importing an Excel file
- Describing a variable in a dataset
- Editing a variable in a dataset
- Model building
- Robust estimation

Edit variables with gretl:

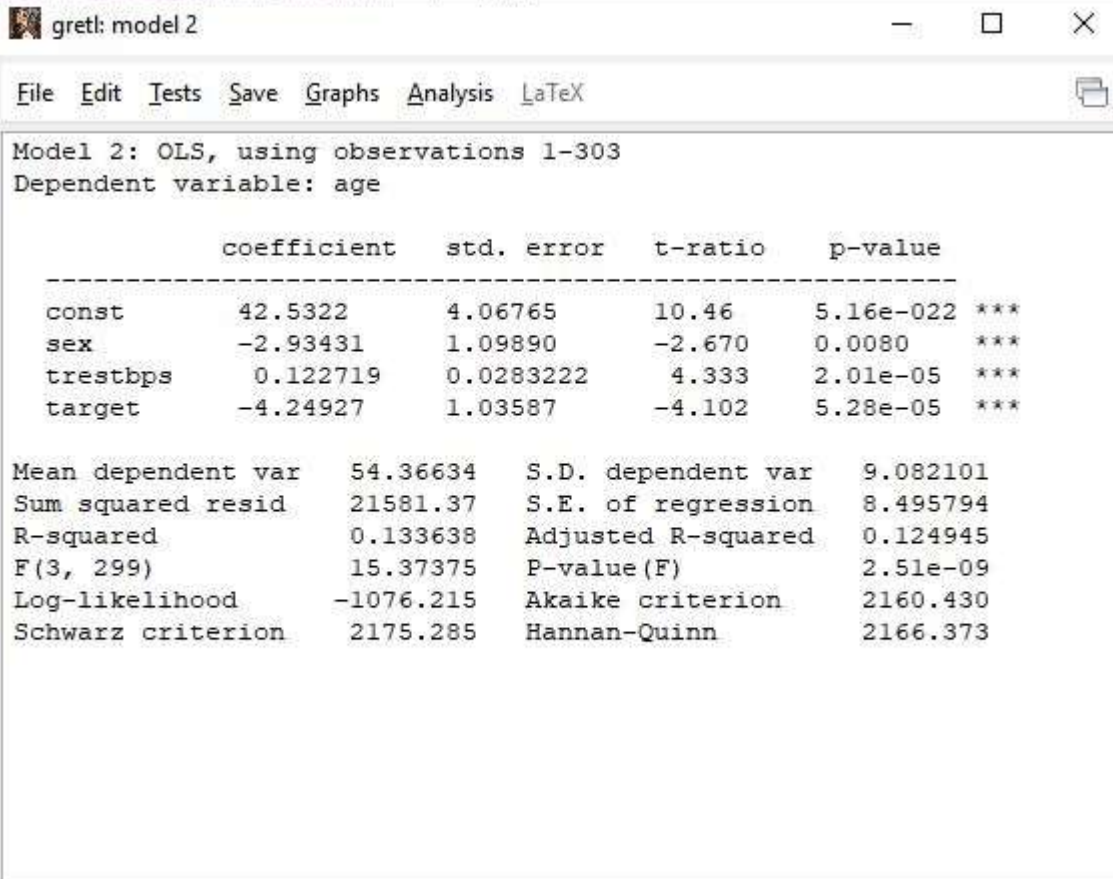
For model building First we add dataset into gretl then we add description for attributes with abbreviated column names:



Model Building with Gretl

1. we use ordinary least square for modeling :

As an example we take heart data set and try to build a model for it we take age as dependent variable to find the(coefficient, standard- error, t-ratio and p-value) with sex ,resting blood pressure and target) and also to get sum and mean



	coefficient	std. error	t-ratio	p-value	
const	42.5322	4.06765	10.46	5.16e-022	***
sex	-2.93431	1.09890	-2.670	0.0080	***
trestbps	0.122719	0.0283222	4.333	2.01e-05	***
target	-4.24927	1.03587	-4.102	5.28e-05	***

Mean dependent var	54.36634	S.D. dependent var	9.082101
Sum squared resid	21581.37	S.E. of regression	8.495794
R-squared	0.133638	Adjusted R-squared	0.124945
F(3, 299)	15.37375	P-value(F)	2.51e-09
Log-likelihood	-1076.215	Akaike criterion	2160.430
Schwarz criterion	2175.285	Hannan-Quinn	2166.373

OR we take target as dependent variable to find the(coefficient, standard- error, t-ration and p-value) with (age ,sex) and also to get sum and mean

gretl: model 4

File Edit Tests Save Graphs Analysis LaTeX

Model 4: OLS, using observations 1-303
Dependent variable: target

	coefficient	std. error	t-ratio	p-value	
age	0.0113648	0.000917151	12.39	8.99e-029	***
sex	-0.161680	0.0611596	-2.644	0.0086	***
Mean dependent var	0.544554	S.D. dependent var	0.498835		
Sum squared resid	81.59411	S.E. of regression	0.520650		
Uncentered R-squared	0.505490	Centered R-squared	-0.085771		
F(2, 301)	153.8418	P-value(F)	9.40e-47		
Log-likelihood	-231.1740	Akaike criterion	466.3481		
Schwarz criterion	473.7756	Hannan-Quinn	469.3196		

2: Robust estimation

1. we take target as dependent variable to find the(coefficient, standard- error, t-ration and p-value) with (age ,sex) and also to get sum and median

gretl: model 7

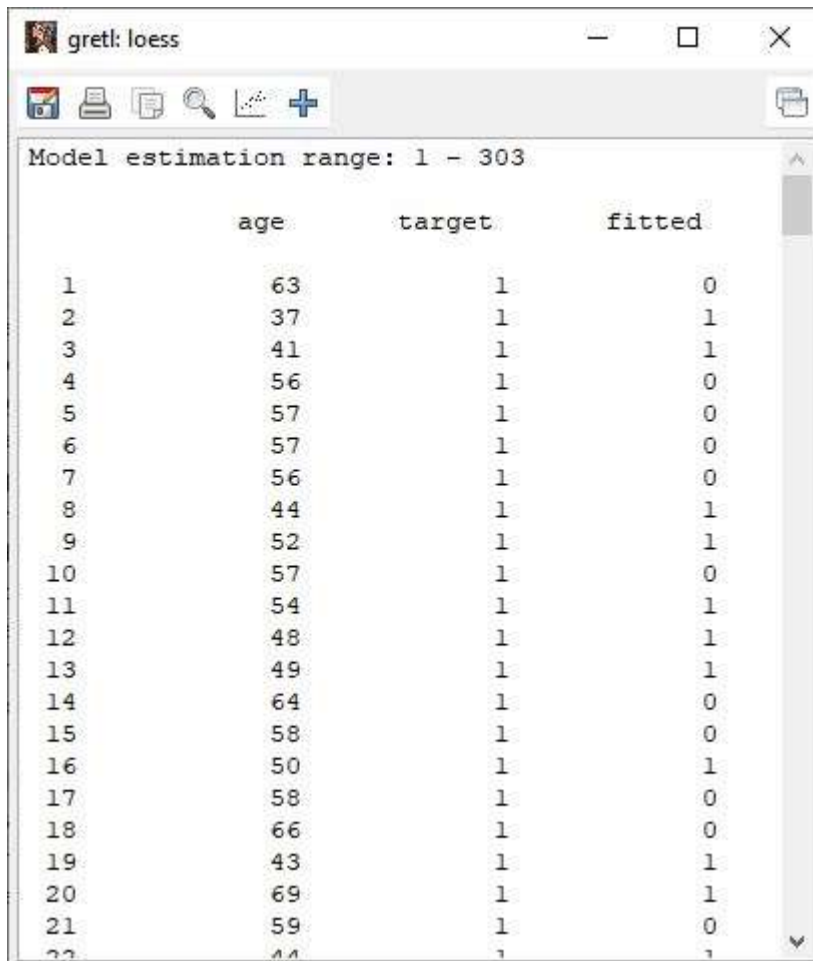
File Edit Tests Save Graphs Analysis LaTeX

Model 7: LAD, using observations 1-303
Dependent variable: target

	coefficient	std. error	t-ratio	p-value	
const	1.00000	0.496341	2.015	0.0448	**
age	0.000000	0.0101871	0.0000	1.0000	
sex	-1.00000	0.279844	-3.573	0.0004	***
Median depend. var	1.000000	S.D. dependent var	0.498835		
Sum absolute resid	117.0000	Sum squared resid	117.0000		
Log-likelihood	-224.7013	Akaike criterion	455.4025		
Schwarz criterion	466.5437	Hannan-Quinn	459.8598		

Using loess method:

we take target as dependent variable with age column for robust estimation



The screenshot shows a window titled "gretl: loess" with a toolbar containing icons for file operations, search, and plotting. Below the toolbar, it says "Model estimation range: 1 - 303". The main area displays a table with three columns: "age", "target", and "fitted". The table contains 22 rows of data, with the first row being the header and the subsequent rows numbered 1 through 22. The "age" column contains values ranging from 37 to 69. The "target" column contains values of 1 or 0. The "fitted" column contains values of 1 or 0, which appear to be the result of a loess regression fit.

	age	target	fitted
1	63	1	0
2	37	1	1
3	41	1	1
4	56	1	0
5	57	1	0
6	57	1	0
7	56	1	0
8	44	1	1
9	52	1	1
10	57	1	0
11	54	1	1
12	48	1	1
13	49	1	1
14	64	1	0
15	58	1	0
16	50	1	1
17	58	1	0
18	66	1	0
19	43	1	1
20	69	1	1
21	59	1	0
22	44	1	1

4. Graph for target vs age

