Big Data Analysis

Exploratory Data Analysis (PDF, CDF, Univariate analysis)

- Teaching Assistant
  Yash More

INDEX :

- Exploratory Data Analysis (EDA) introduction
- Key components of EDA
- Steps to perform exploratory data analysis
- EDA - Exploratory Data Analysis: Using Python Functions
  - Load the Data
  - Basic information about data - EDA
  - Duplicate values
  - Unique values in the data
  - Visualize the Unique counts
  - Find the Null values
  - Replace the Null values
  - Know the datatypes
  - Filter the Data
  - A quick box plot
  - Correlation Plot - EDA
- Ending Note - EDA

Exploratory Data Analysis (EDA) introduction :

- Exploratory Data Analysis (EDA) is a critical step in the data analysis process where we examine, summarize, and visualize the data to gain insights and identify patterns.
- Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

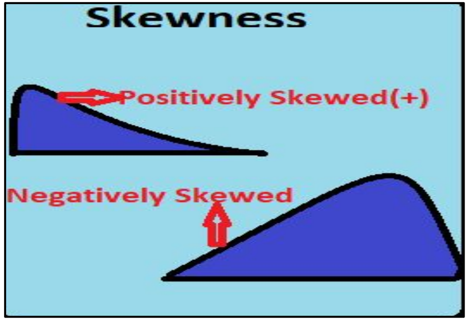
Exploratory Data Analysis (EDA) introduction :

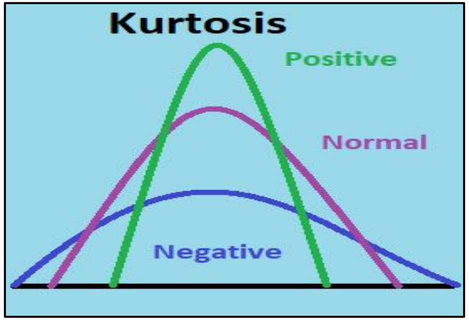
- Exploratory Data Analysis (EDA) is a critical step in the data analysis process where we examine, summarize, and visualize the data to gain insights and identify patterns.
- Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

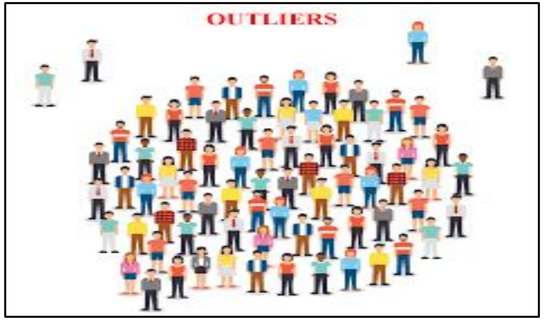
- EDA explained using sample Data set:
    - Link : https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15

Key components of EDA :

- **PDF (Probability Density Function) :** The PDF is a representation of the distribution of a continuous random variable. It shows the likelihood of the occurrence of a particular value. In EDA, PDF is used to visualize the distribution of the data and identify skewness, kurtosis, and outliers.

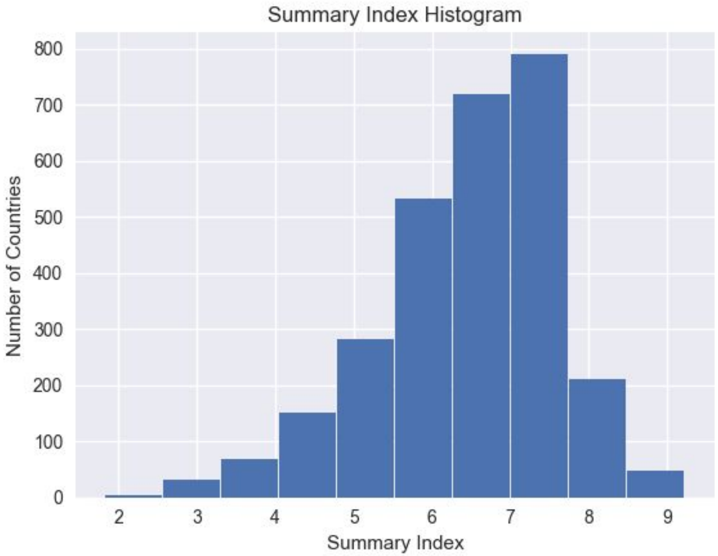






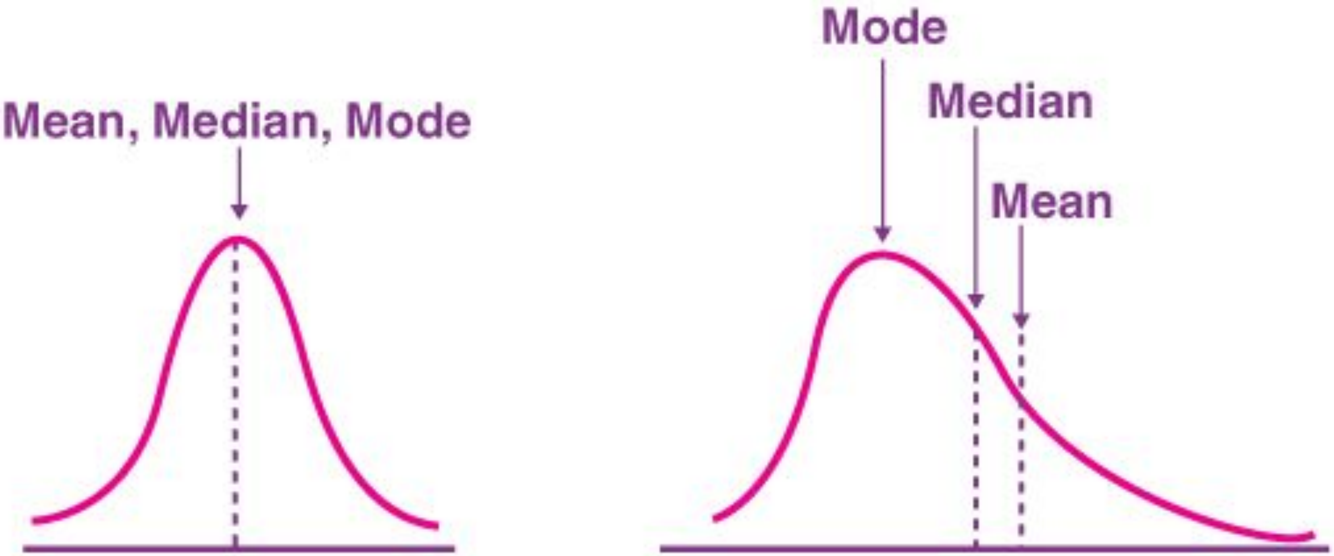
Key components of EDA :

- **CDF (Cumulative Distribution Function) :** The CDF is a function that gives the probability that a random variable X is less than or equal to a particular value. CDF can be used to summarize the distribution of the data and calculate the percentile values.



Key components of EDA :

- **Univariate Analysis :** Univariate analysis is a statistical method that focuses on one variable at a time. In EDA, univariate analysis is used to gain insights into the distribution, central tendency, and variability of the data. This includes calculating measures such as mean, median, mode, variance, and standard deviation.



Steps to perform exploratory data analysis :

Steps to perform exploratory data analysis :

To perform an exploratory data analysis, we follow the following steps:

1. **Load the data**
2. **Summary statistics**
3. **Visualization**
4. **Data Cleaning**
5. **Further Analysis**

Steps to perform exploratory data analysis :

1. **Load the data:** Load the data into a data frame or a matrix, depending on the size and structure of the data.
2. **Summary statistics**
3. **Visualization**
4. **Data Cleaning**
5. **Further Analysis**

Steps to perform exploratory data analysis :

1. **Load the data:** Load the data into a data frame or a matrix, depending on the size and structure of the data.
2. **Summary statistics:** Calculate summary statistics, such as mean, median, mode, variance, and standard deviation, to get a general idea of the distribution of the data.
3. **Visualization**
4. **Data Cleaning**
5. **Further Analysis**

Steps to perform exploratory data analysis :

1. **Load the data:** Load the data into a data frame or a matrix, depending on the size and structure of the data.
2. **Summary statistics:** Calculate summary statistics, such as mean, median, mode, variance, and standard deviation, to get a general idea of the distribution of the data.
3. **Visualization:** Plot histograms, density plots, box plots, and scatter plots to visualize the distribution of the data and identify outliers.
4. **Data Cleaning**
5. **Further Analysis**

Steps to perform exploratory data analysis :

1. **Load the data:** Load the data into a data frame or a matrix, depending on the size and structure of the data.
2. **Summary statistics:** Calculate summary statistics, such as mean, median, mode, variance, and standard deviation, to get a general idea of the distribution of the data.
3. **Visualization:** Plot histograms, density plots, box plots, and scatter plots to visualize the distribution of the data and identify outliers.
4. **Data Cleaning:** Clean the data by removing missing values, incorrect values, and outliers.
5. **Further Analysis**

Steps to perform exploratory data analysis :

1. **Load the data:** Load the data into a data frame or a matrix, depending on the size and structure of the data.
2. **Summary statistics:** Calculate summary statistics, such as mean, median, mode, variance, and standard deviation, to get a general idea of the distribution of the data.
3. **Visualization:** Plot histograms, density plots, box plots, and scatter plots to visualize the distribution of the data and identify outliers.
4. **Data Cleaning:** Clean the data by removing missing values, incorrect values, and outliers.
5. **Further Analysis:** Conduct further analysis if needed, such as bi-variate analysis, multivariate analysis, and regression analysis.

Steps to perform exploratory data analysis :

1. **Load the data:** Load the data into a data frame or a matrix, depending on the size and structure of the data.
2. **Summary statistics:** Calculate summary statistics, such as mean, median, mode, variance, and standard deviation, to get a general idea of the distribution of the data.
3. **Visualization:** Plot histograms, density plots, box plots, and scatter plots to visualize the distribution of the data and identify outliers.
4. **Data Cleaning:** Clean the data by removing missing values, incorrect values, and outliers.
5. **Further Analysis:** Conduct further analysis if needed, such as bi-variate analysis, multivariate analysis, and regression analysis.

Conclusion :

Steps to perform exploratory data analysis :

1. **Load the data:** Load the data into a data frame or a matrix, depending on the size and structure of the data.
2. **Summary statistics:** Calculate summary statistics, such as mean, median, mode, variance, and standard deviation, to get a general idea of the distribution of the data.
3. **Visualization:** Plot histograms, density plots, box plots, and scatter plots to visualize the distribution of the data and identify outliers.
4. **Data Cleaning:** Clean the data by removing missing values, incorrect values, and outliers.
5. **Further Analysis:** Conduct further analysis if needed, such as bi-variate analysis, multivariate analysis, and regression analysis.

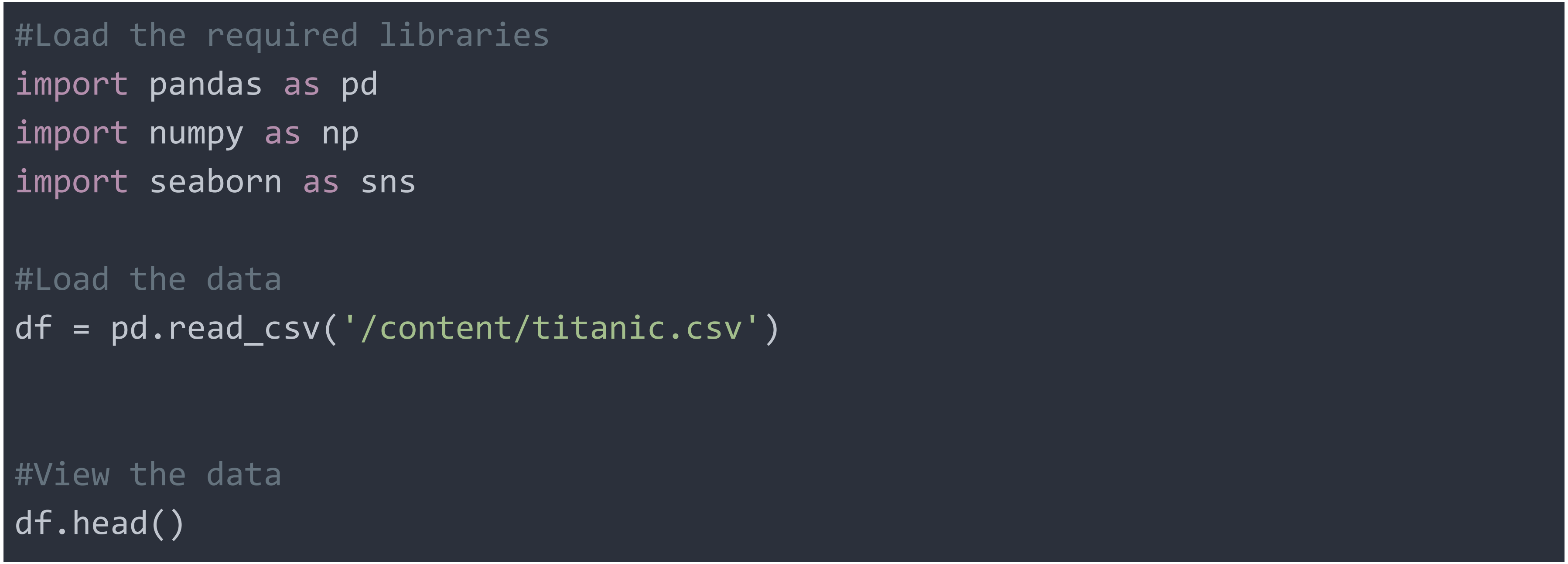
Conclusion :

Exploratory data analysis is a crucial step in the data analysis process that helps us gain insights into the data, identify patterns and relationships, and prepare the data for further analysis.

EDA - Exploratory Data Analysis: Using Python Functions :

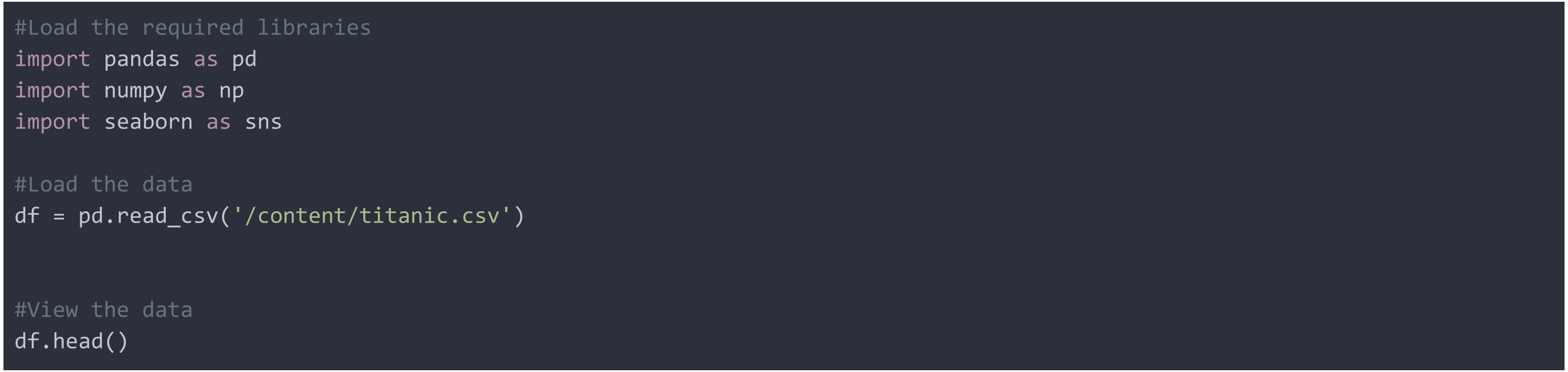
- EDA is applied to investigate the data and summarize the key insights.
- It will give you the basic understanding of your data, it’s distribution, null values and much more.
- You can either explore data using graphs or through some python functions.
- There will be two type of analysis. Univariate and Bivariate. In the univariate, you will be analyzing a single attribute. But in the bivariate, you will be analyzing an attribute with the target attribute.
- In the non-graphical approach, you will be using functions such as shape, summary, describe, isnull, info, datatypes and more.
- In the graphical approach, you will be using plots such as scatter, box, bar, density and correlation plots.

Load the Data :

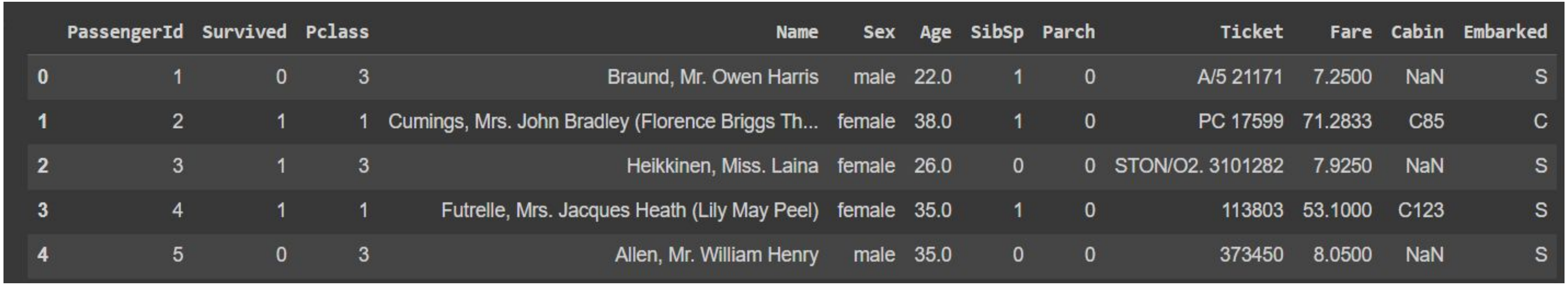


```
#Load the required libraries
import pandas as pd
import numpy as np
import seaborn as sns

#Load the data
df = pd.read_csv('/content/titanic.csv')


#View the data
df.head()
```

Load the Data :



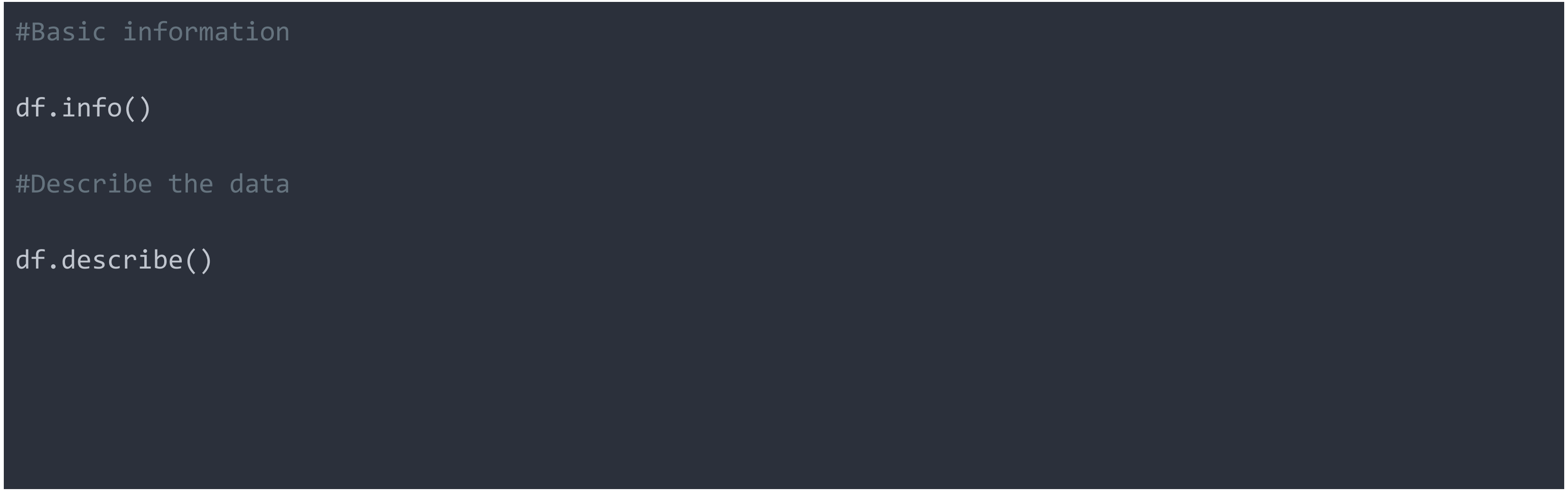
```
#Load the required libraries
import pandas as pd
import numpy as np
import seaborn as sns

#Load the data
df = pd.read_csv('/content/titanic.csv')


#View the data
df.head()
```



	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

Basic information about data - EDA :

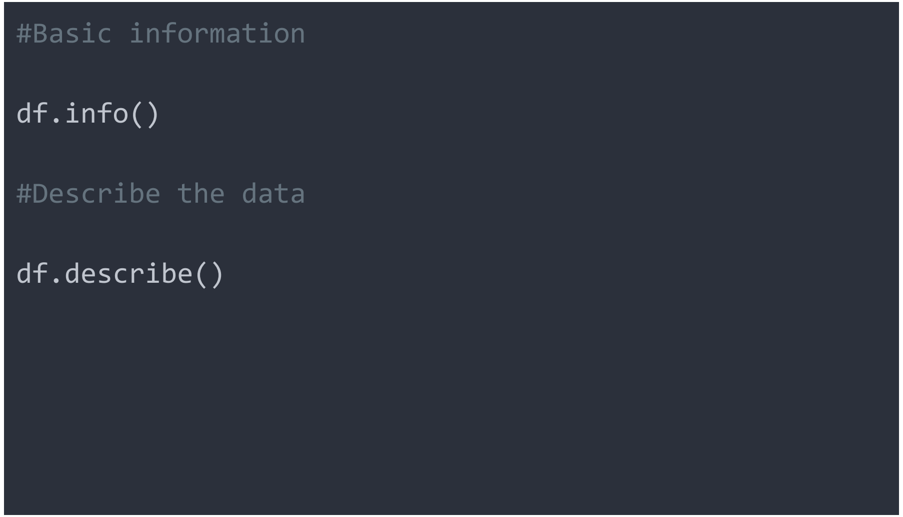
- The ***df.info()*** function will give us the basic information about the dataset. For any data, it is good to start by knowing its information. Let’s see how it works with our data.



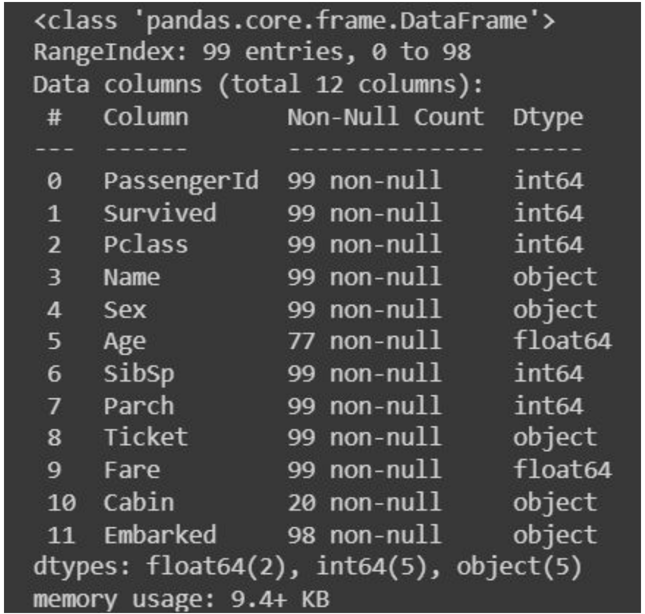
```
#Basic information

df.info()

#Describe the data

df.describe()
```

Basic information about data - EDA :

- The ***df.info()*** function will give us the basic information about the dataset. For any data, it is good to start by knowing its information. Let’s see how it works with our data.



```
#Basic information

df.info()

#Describe the data

df.describe()
```

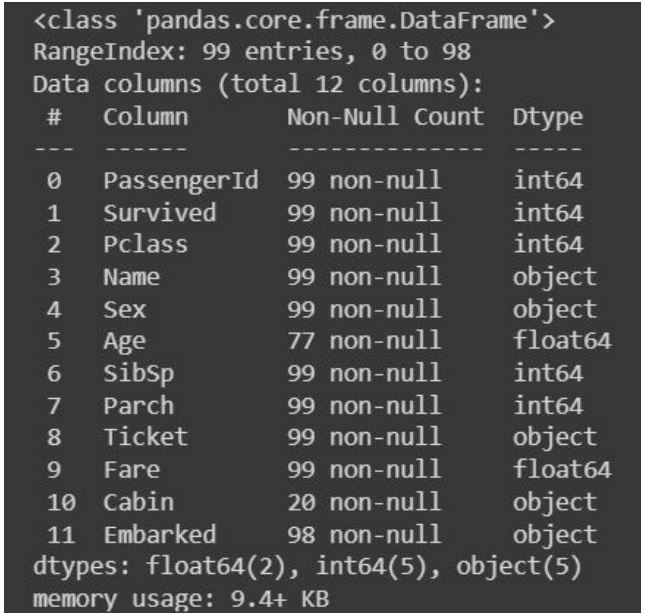


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99 entries, 0 to 98
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  99 non-null     int64  
 1   Survived     99 non-null     int64  
 2   Pclass       99 non-null     int64  
 3   Name         99 non-null     object 
 4   Sex          99 non-null     object 
 5   Age          77 non-null     float64
 6   SibSp        99 non-null     int64  
 7   Parch        99 non-null     int64  
 8   Ticket       99 non-null     object 
 9   Fare         99 non-null     float64
 10  Cabin        20 non-null     object 
 11  Embarked     98 non-null     object 
dtypes: float64(2), int64(5), object(5)
memory usage: 9.4+ KB
```

Basic information about data - EDA :

- The ***df.info()*** function will give us the basic information about the dataset. For any data, it is good to start by knowing its information. Let’s see how it works with our data.

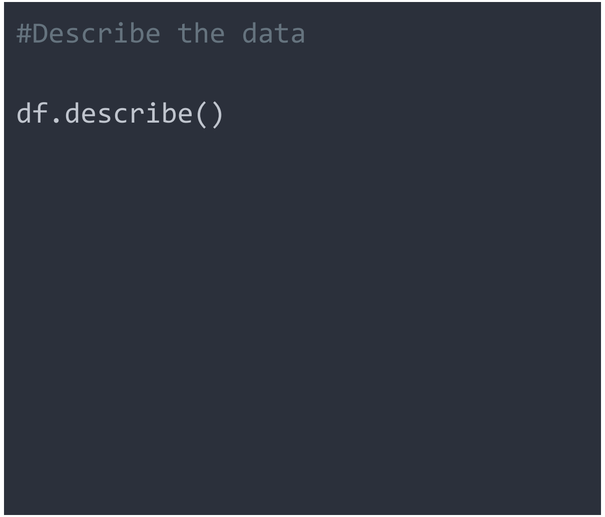
- Using this function, you can see the number of null values, datatypes, and memory usage as shown in the above outputs along with descriptive statistics.



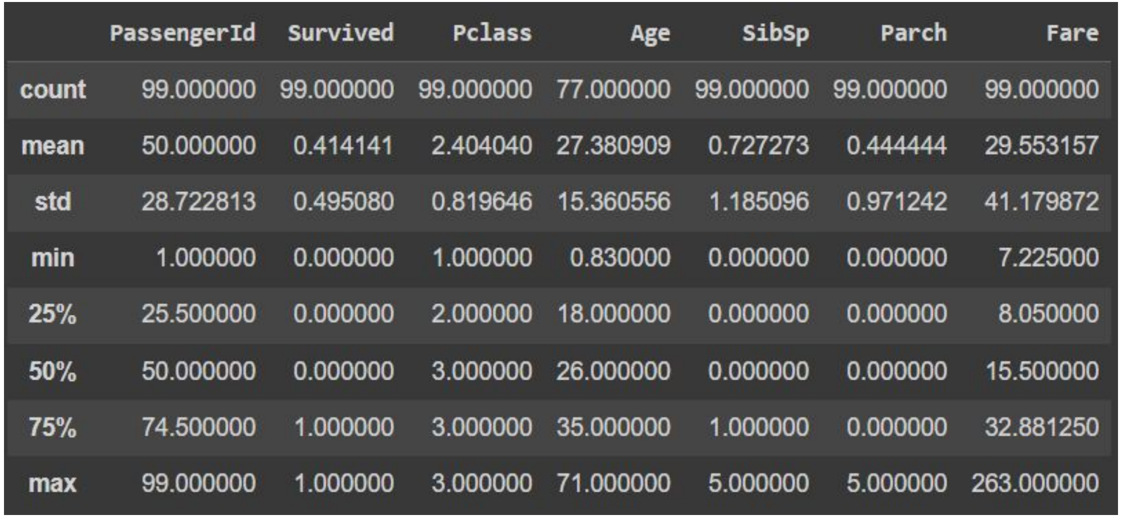
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99 entries, 0 to 98
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  99 non-null     int64
 1   Survived     99 non-null     int64
 2   Pclass       99 non-null     int64
 3   Name         99 non-null     object
 4   Sex          99 non-null     object
 5   Age          77 non-null     float64
 6   SibSp        99 non-null     int64
 7   Parch        99 non-null     int64
 8   Ticket       99 non-null     object
 9   Fare         99 non-null     float64
 10  Cabin        20 non-null     object
 11  Embarked     98 non-null     object
dtypes: float64(2), int64(5), object(5)
memory usage: 9.4+ KB
```

Basic information about data - EDA :

- The ***df.info()*** function will give us the basic information about the dataset. For any data, it is good to start by knowing its information. Let’s see how it works with our data.



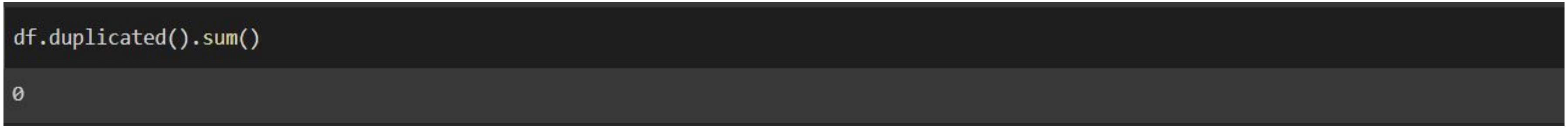
```
#Describe the data

df.describe()
```



	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	99.000000	99.000000	99.000000	77.000000	99.000000	99.000000	99.000000
mean	50.000000	0.414141	2.404040	27.380909	0.727273	0.444444	29.553157
std	28.722813	0.495080	0.819646	15.360556	1.185096	0.971242	41.179872
min	1.000000	0.000000	1.000000	0.830000	0.000000	0.000000	7.225000
25%	25.500000	0.000000	2.000000	18.000000	0.000000	0.000000	8.050000
50%	50.000000	0.000000	3.000000	26.000000	0.000000	0.000000	15.500000
75%	74.500000	1.000000	3.000000	35.000000	1.000000	0.000000	32.881250
max	99.000000	1.000000	3.000000	71.000000	5.000000	5.000000	263.000000

Duplicate values :

- You can use the df.duplicate.sum() function to the sum of duplicate value present if any. It will show the number of duplicate values if they are present in the data.



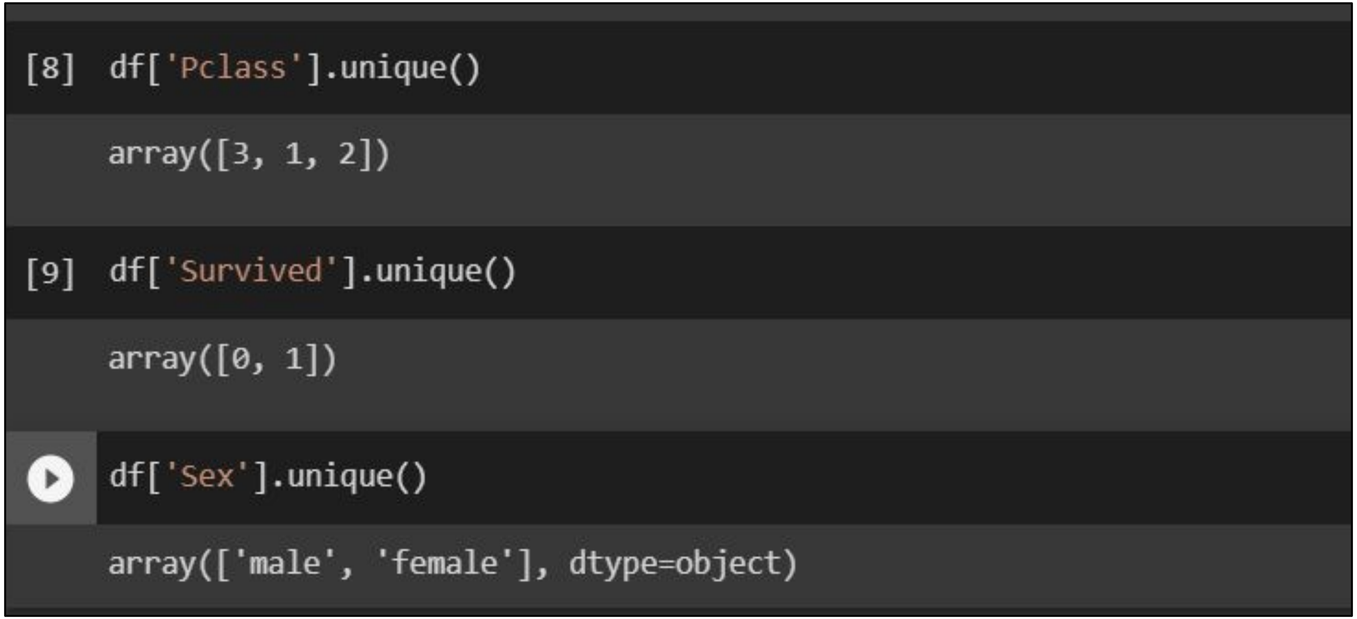
```
df.duplicated().sum()
```

```
0
```

- Well, the function returned ‘0’. This means, there is not a single duplicate value present in our dataset and it is a very good thing to know.

Unique values in the data :

- You can find the number of unique values in the particular column using unique() function in python.

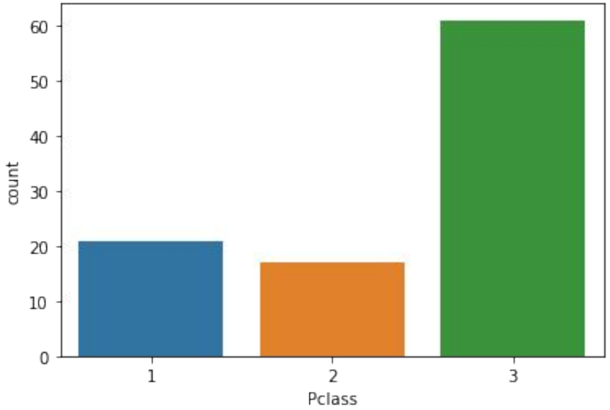


```
[8]  df['Pclass'].unique()

     array([3, 1, 2])

[9]  df['Survived'].unique()

     array([0, 1])

     df['Sex'].unique()

     array(['male', 'female'], dtype=object)
```

Visualize the Unique counts :

- Yes, you can visualize the unique values present in the data. For this, we will be using the seaborn library. You have to call the sns.countlot() function and specify the variable to plot the count plot.

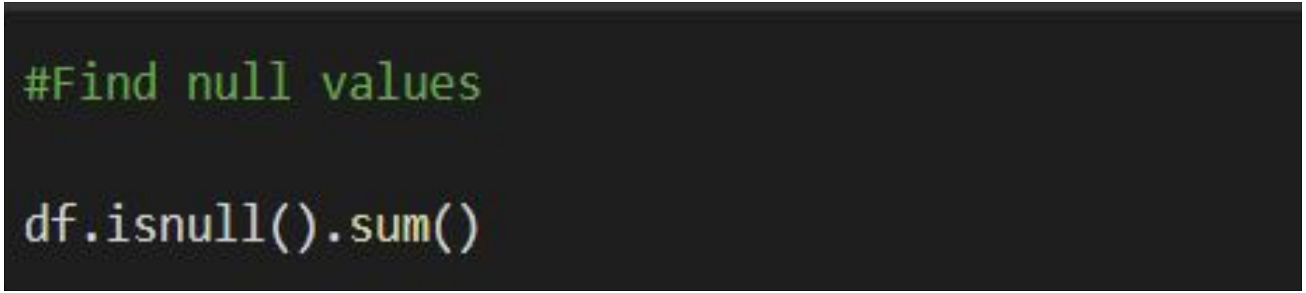
```
sns.countplot(df['Pclass'])
```



- That's great! You are doing good. It is as simple as that. Though EDA has two approaches, a blend of graphical and non-graphical will give you the bigger picture altogether.

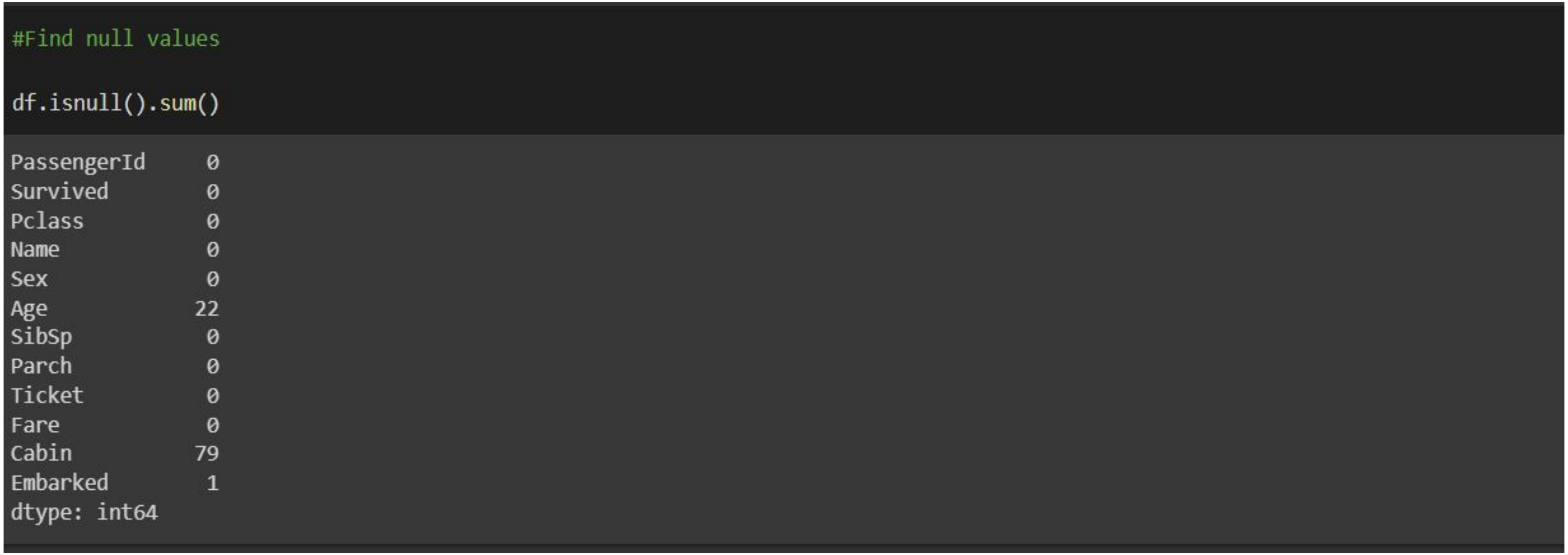
Find the Null values :

- Finding the null values is the most important step in the EDA.
- Ensuring the quality of data is paramount. So, let’s see how we can find the null values.



```
#Find null values

df.isnull().sum()
```

Find the Null values :



```
#Find null values

df.isnull().sum()
```

```
PassengerId     0
Survived        0
Pclass          0
Name            0
Sex             0
Age            22
SibSp           0
Parch           0
Ticket          0
Fare            0
Cabin          79
Embarked        1
dtype: int64
```

- Oh no, we have some null values in the ***‘Age’*** and ***‘Cabin’*** variables. But, don’t worry. We will find a way to deal with them soon.

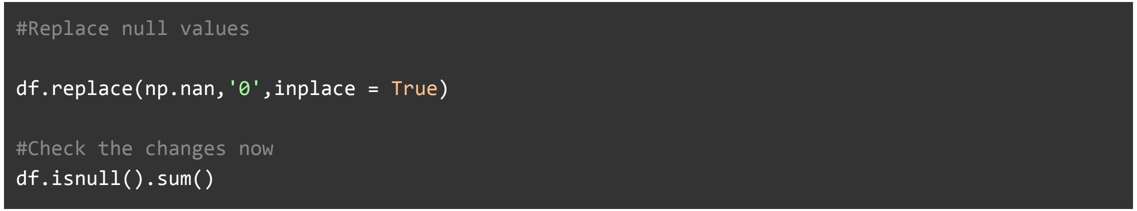
Replace the Null values :

- Hey, we got a replace() function to replace all the null values with a specific data.

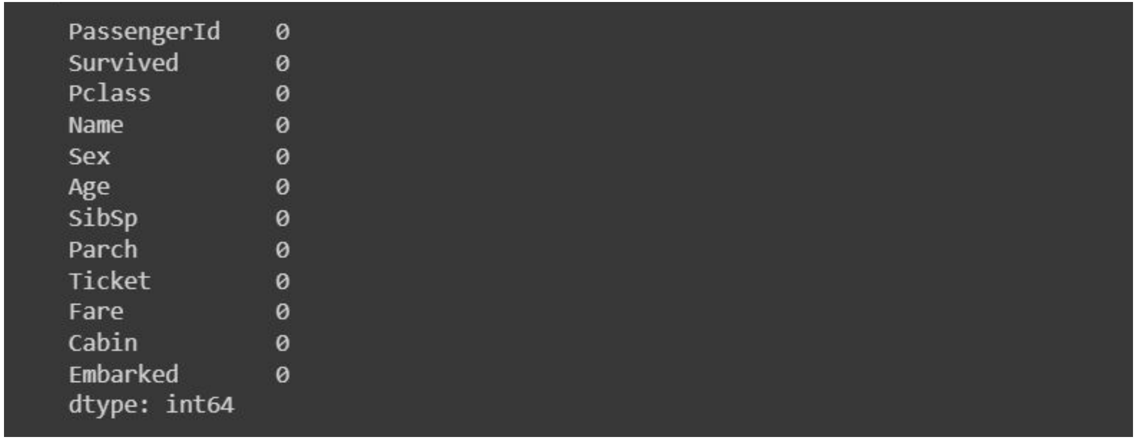
```
#Replace null values

df.replace(np.nan,'0',inplace = True)

#Check the changes now
df.isnull().sum()
```

Replace the Null values :

- Hey, we got a replace() function to replace all the null values with a specific data.



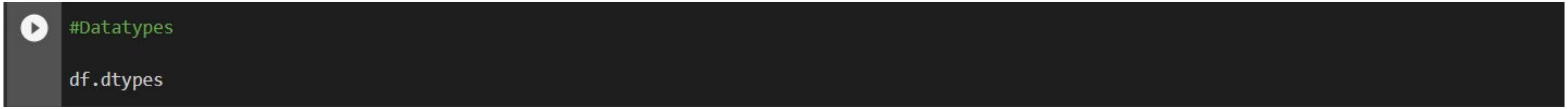
```
#Replace null values

df.replace(np.nan,'0',inplace = True)

#Check the changes now
df.isnull().sum()
```



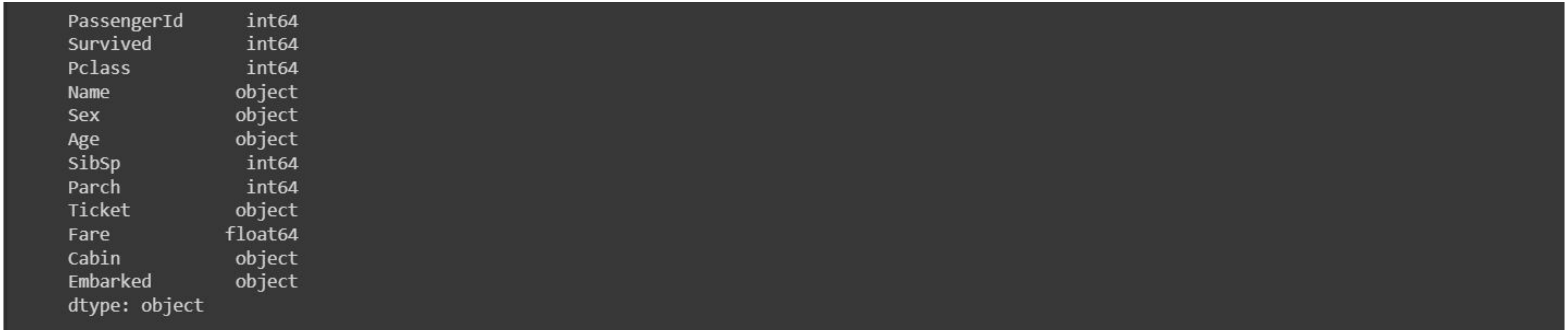
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

Know the datatypes :

- Knowing the datatypes which you are exploring is very important and an easy process too. Let’s see how it works.



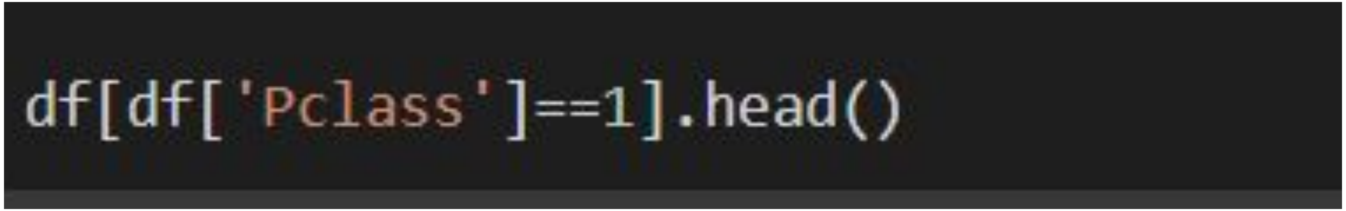
```
#Datatypes

df.dtypes
```



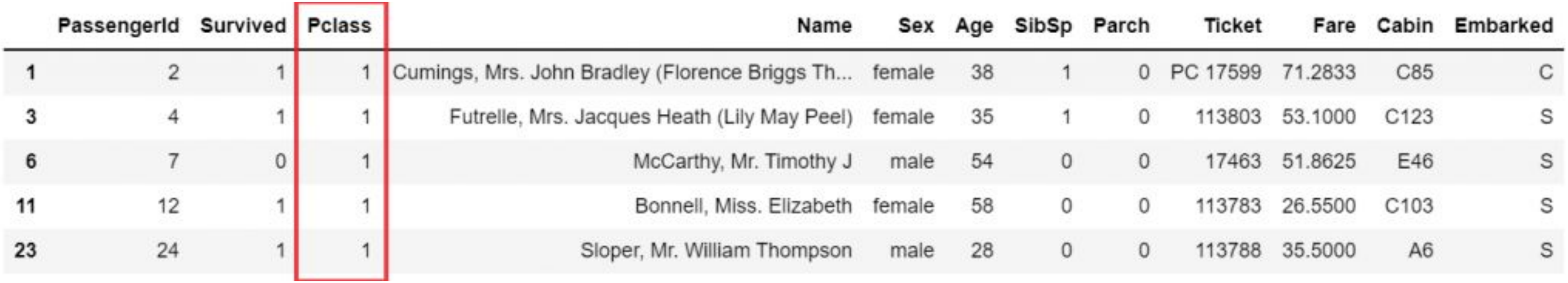
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age             object
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

Filter the Data :

- Yes, you can filter the data based on some logic.



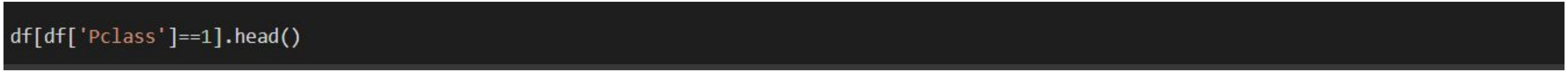
```
df[df['Pclass']==1].head()
```



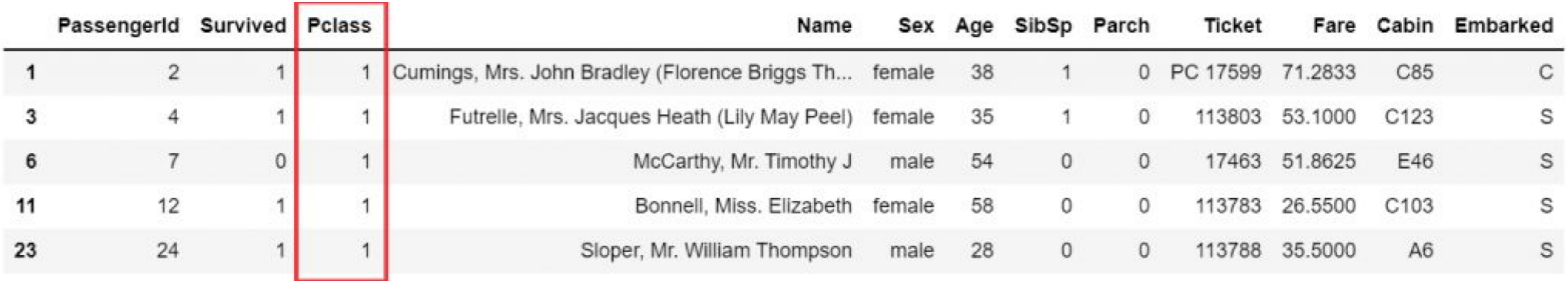
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	0	PC 17599	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.5500	C103	S
23	24	1	1	Sloper, Mr. William Thompson	male	28	0	0	113788	35.5000	A6	S

Filter the Data :

- Yes, you can filter the data based on some logic.



```
df[df['Pclass']==1].head()
```

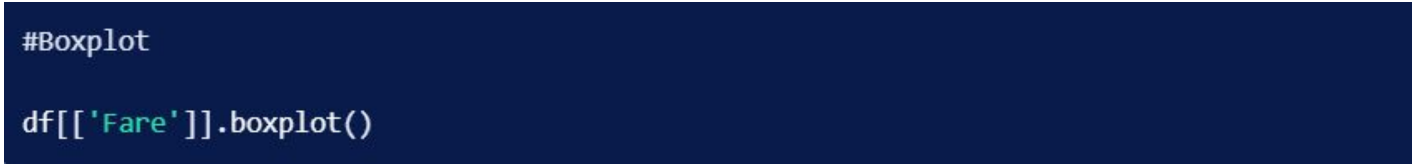


	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	0	PC 17599	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.5500	C103	S
23	24	1	1	Sloper, Mr. William Thompson	male	28	0	0	113788	35.5000	A6	S

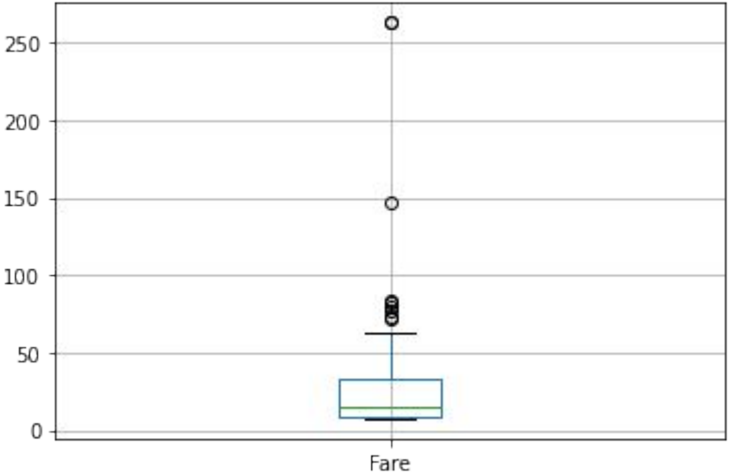
- You can see that the above code has returned only data values that belong to class 1.

A quick box plot :

- You can create a box plot for any numerical column using a single line of code.

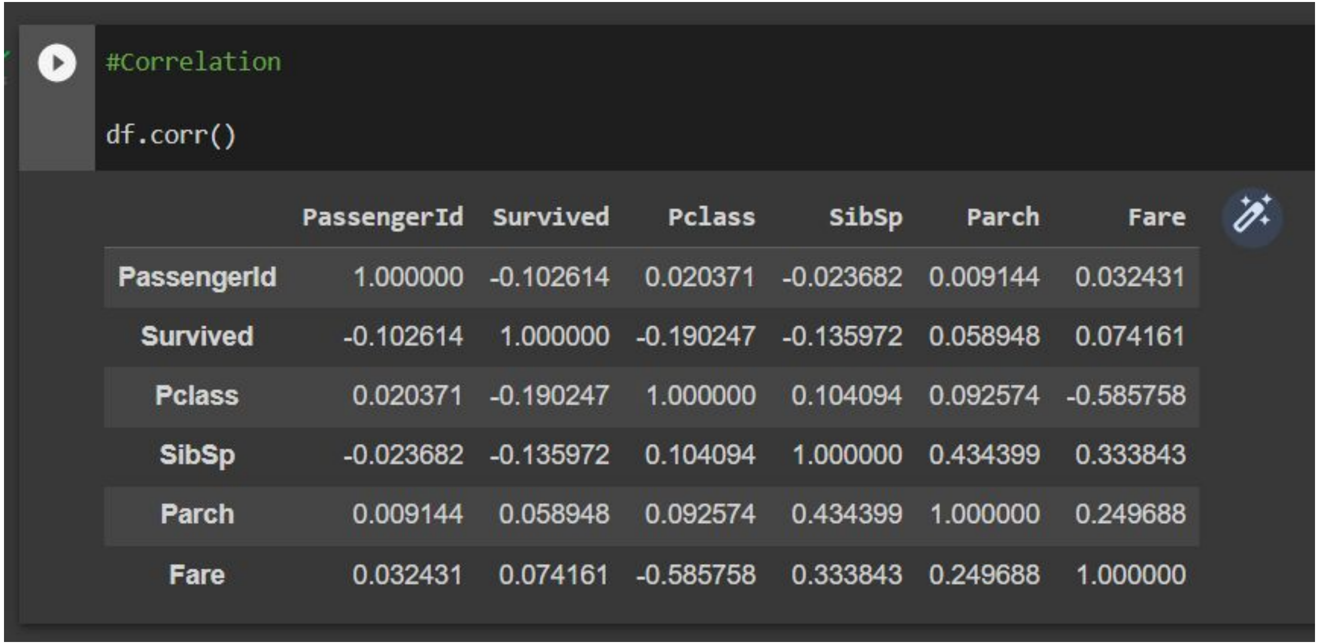


```
#Boxplot

df[['Fare']].boxplot()
```



Correlation Plot - EDA :

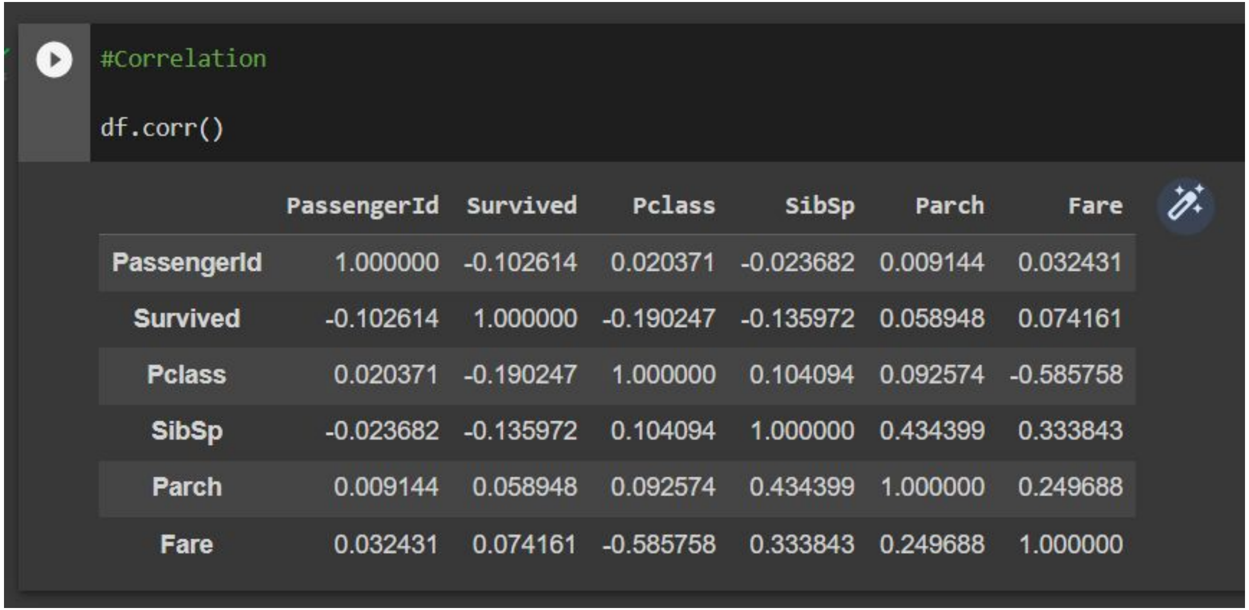
- Finally, to find the correlation among the variables, we can make use of the correlation function. This will give you a fair idea of the correlation strength between different variables.



```
#Correlation

df.corr()
```

```
             PassengerId  Survived    Pclass     SibSp     Parch      Fare
PassengerId     1.000000 -0.102614  0.020371 -0.023682  0.009144  0.032431
   Survived    -0.102614  1.000000 -0.190247 -0.135972  0.058948  0.074161
     Pclass     0.020371 -0.190247  1.000000  0.104094  0.092574 -0.585758
      SibSp    -0.023682 -0.135972  0.104094  1.000000  0.434399  0.333843
      Parch     0.009144  0.058948  0.092574  0.434399  1.000000  0.249688
       Fare     0.032431  0.074161 -0.585758  0.333843  0.249688  1.000000
```

Correlation Plot - EDA :

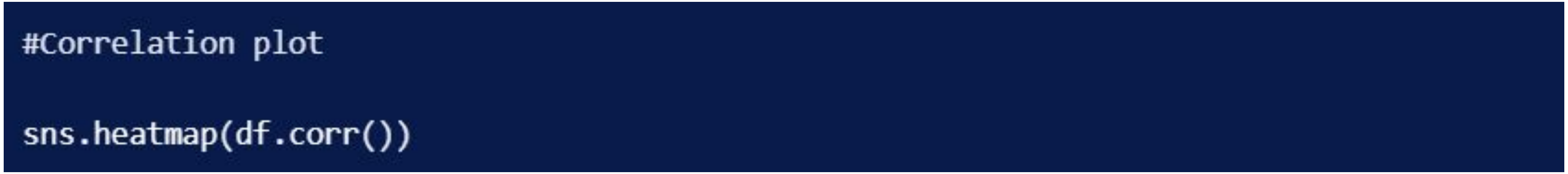


```
#Correlation

df.corr()
```

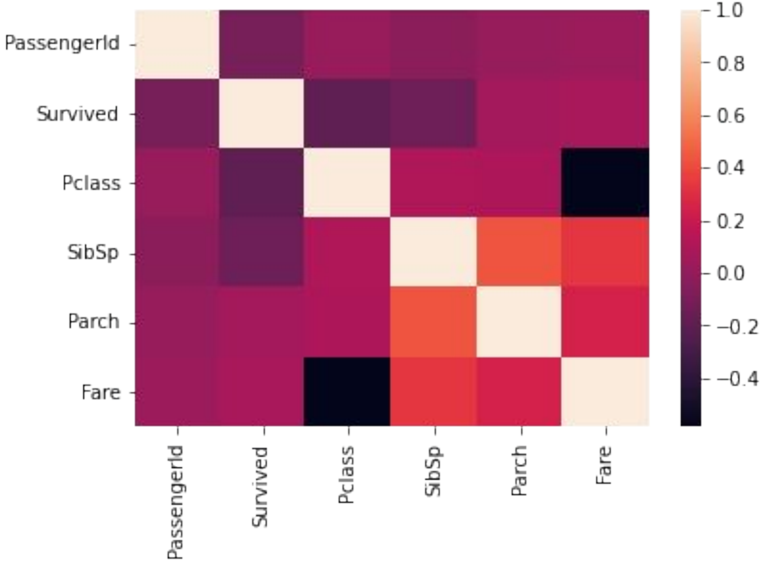
	PassengerId	Survived	Pclass	SibSp	Parch	Fare
PassengerId	1.000000	-0.102614	0.020371	-0.023682	0.009144	0.032431
Survived	-0.102614	1.000000	-0.190247	-0.135972	0.058948	0.074161
Pclass	0.020371	-0.190247	1.000000	0.104094	0.092574	-0.585758
SibSp	-0.023682	-0.135972	0.104094	1.000000	0.434399	0.333843
Parch	0.009144	0.058948	0.092574	0.434399	1.000000	0.249688
Fare	0.032431	0.074161	-0.585758	0.333843	0.249688	1.000000

- This is the correlation matrix with the range from +1 to -1 where +1 is highly and positively correlated and -1 will be highly negatively correlated.
- You can even visualize the correlation matrix using seaborn library as shown below.

Correlation Plot - EDA :



```
#Correlation plot

sns.heatmap(df.corr())
```



Ending Note - EDA :

- Exploratory data analysis is a crucial step in the data analysis process that helps us gain insights into the data, identify patterns and relationships, and prepare the data for further analysis.

Reference :

- Link : https://www.digitalocean.com/community/tutorials/exploratory-data-analysis-python#10-correlation-plot-eda

Thank You!!!