Exploratory Data Analysis (EDA)

Importing Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
```

```
df=pd.read_csv("/content/Titanic-Dataset.csv")
```

Head is used to Fetch top 5 Rows

```
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

Next steps: Generate code with df View recommended plots New interactive sheet

Tail() is used to Fetch bottom 5 Rows

```
df.tail()
```

What can I help you build?

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

info() is used to get the information about the dataset

```
df.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     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

shape is used to get the rows,coulmns of the dataset

```
df.shape
```

```
(891, 12)
```

Intial inspection

```
df.describe(include='all')
```

ssengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
509.000000	509.000000	509.000000	509.000000	509.000000	509.000000	509.000000	509.0	509.000000	509.000000	5
440.176817	0.243615	2.675835	445.658153	0.762279	22.273084	0.184676	0.0	324.471513	11.620169	
258.270734	0.429685	0.560358	252.347656	0.426106	16.629031	0.436162	0.0	186.704053	7.302116	
1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.0	2.000000	0.000000	
215.000000	0.000000	2.000000	243.000000	1.000000	0.000000	0.000000	0.0	169.000000	7.750000	
434.000000	0.000000	3.000000	441.000000	1.000000	23.000000	0.000000	0.0	317.000000	8.050000	
659.000000	0.000000	3.000000	669.000000	1.000000	32.000000	0.000000	0.0	457.000000	13.000000	
891.000000	1.000000	3.000000	887.000000	1.000000	74.000000	2.000000	0.0	680.000000	52.000000	

dtype is used to know datatype of each colounm

```
df.dtypes
```

	0
PassengerId	int64
Survived	int64
Pclass	int64
Name	object
Sex	object
Age	float64
SibSp	int64
Parch	int64
Ticket	object
Fare	float64
Cabin	object
Embarked	object

dtype: object

isnull().sum() is used to know sum of null values in each row

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

```
                 0
PassengerId      0
   Survived      0
     Pclass      0
       Name      0
        Sex      0
        Age    177
      SibSp      0
      Parch      0
     Ticket      0
       Fare      0
      Cabin    687
   Embarked      2

dtype: int64
```

df.isnull().sum().sum() is used to get total null values in dataset

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

```
np.int64(866)
```

fillna(np.mean) is used to fill the null values with mean value

```
df=df.fillna(np.mean)
```

duplicated().sum() is used to check duplicated values

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

```
np.int64(0)
```

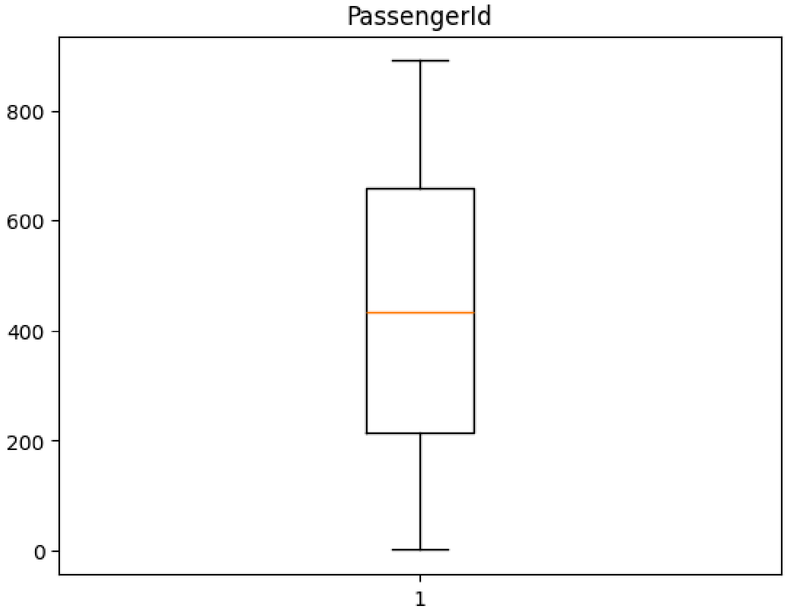
drop_duplicates(inplace=True) is used to drop duplicates

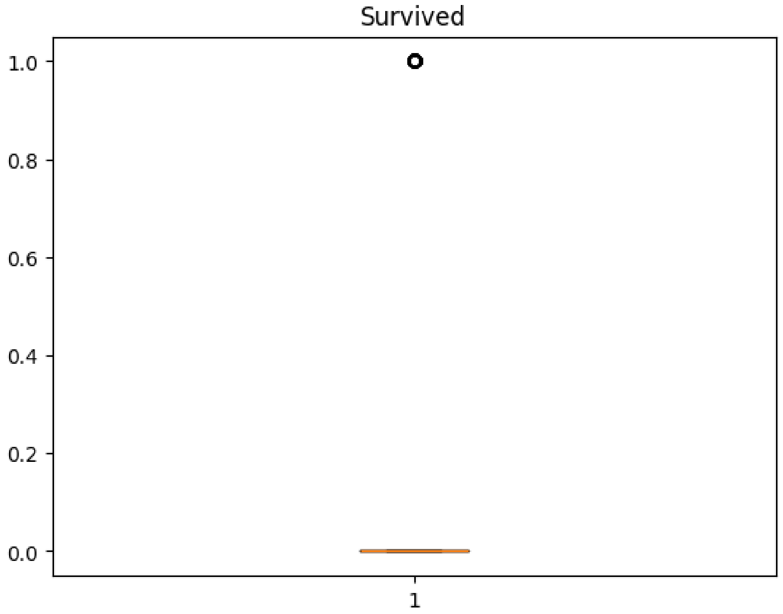
```
df.drop_duplicates(inplace=True)
```

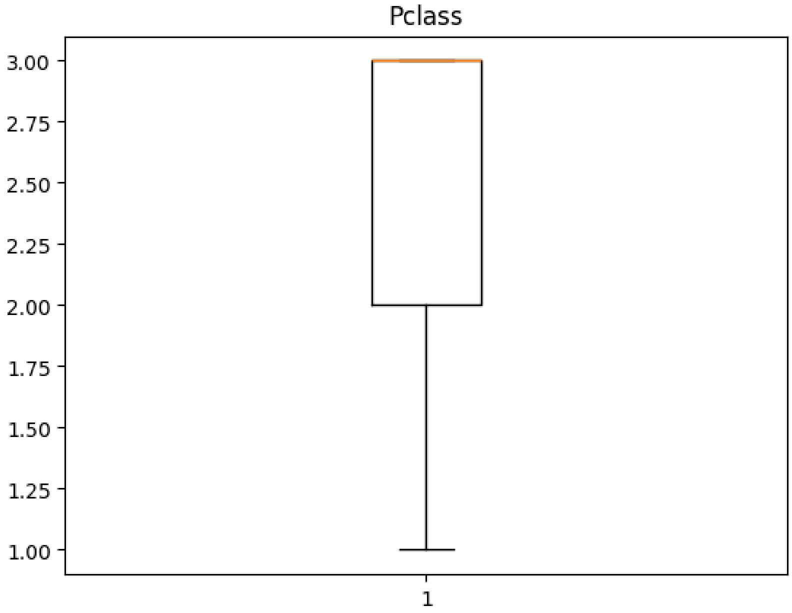
Bar Plot is used to check outliers in a dataset

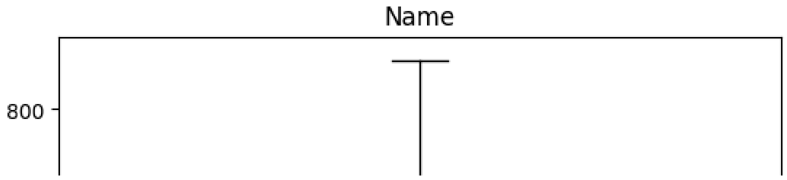
```
obj_col=df.select_dtypes('object').columns
```

```
for i in col:
  if(df[i].dtype !='object'):
    plt.boxplot(df[i])
    plt.title(i)
    plt.show()
```


PassengerId
800
600
400
200
0
1


Survived
1.0
0.8
0.6
0.4
0.2
0.0
1


Pclass
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
1


Name
800