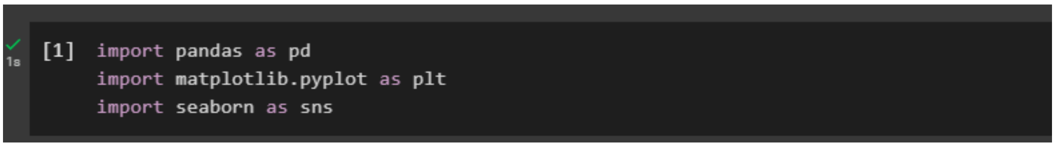
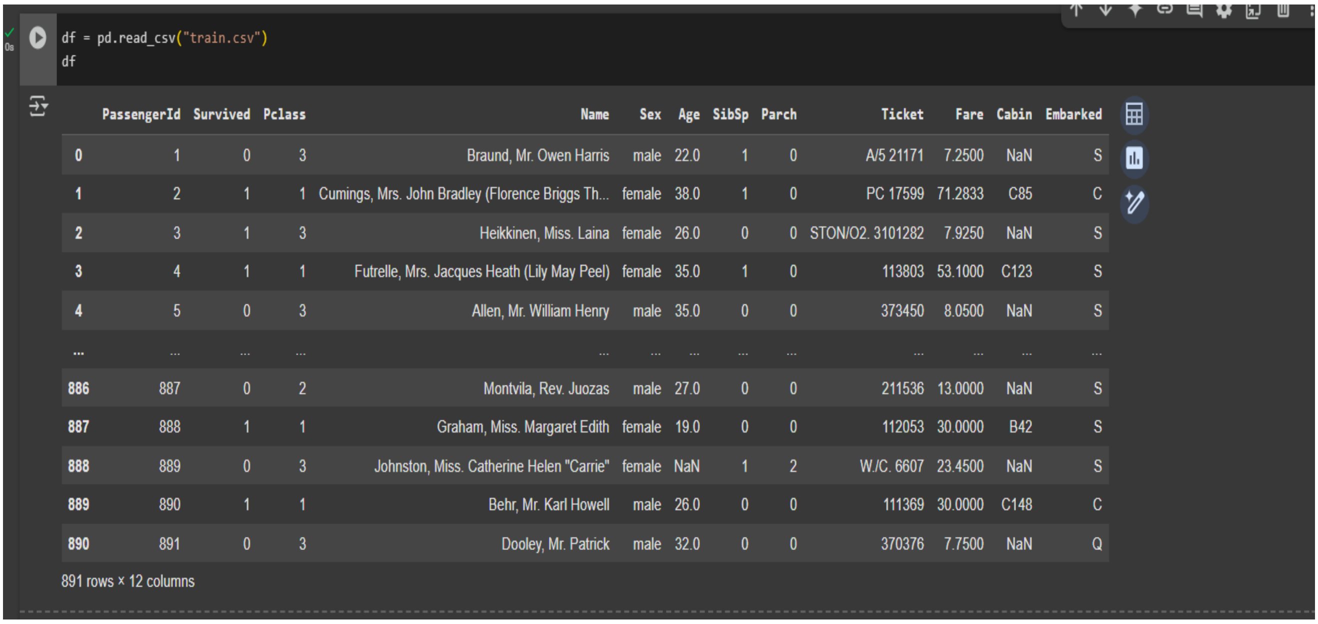
Import Pandas , Matplotlib , Seaborn



```
[1] import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
```

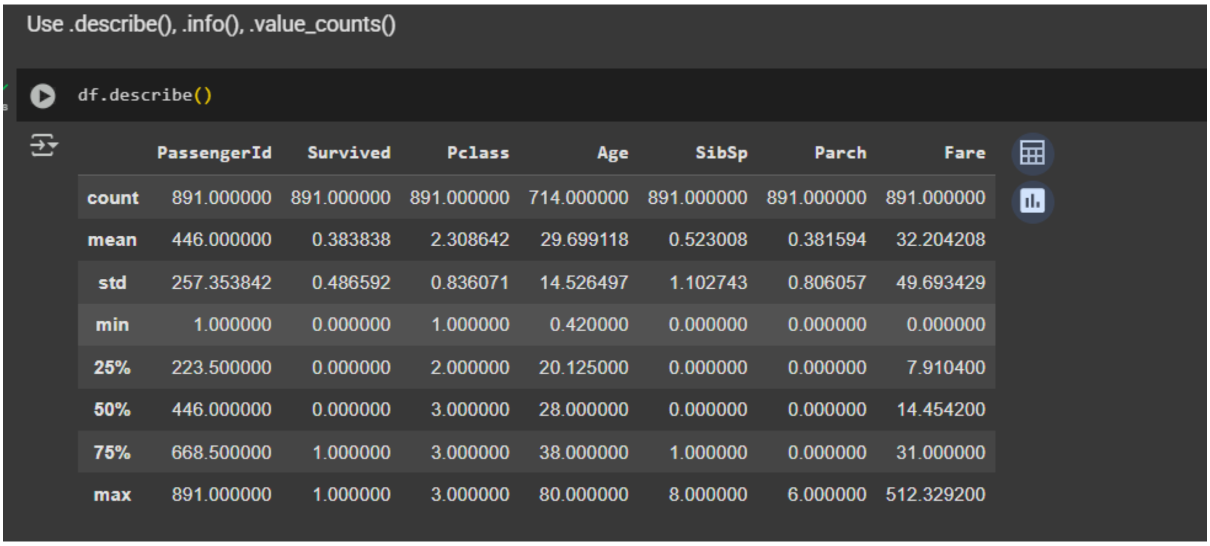
Load data



```
df = pd.read_csv("train.csv")
df
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

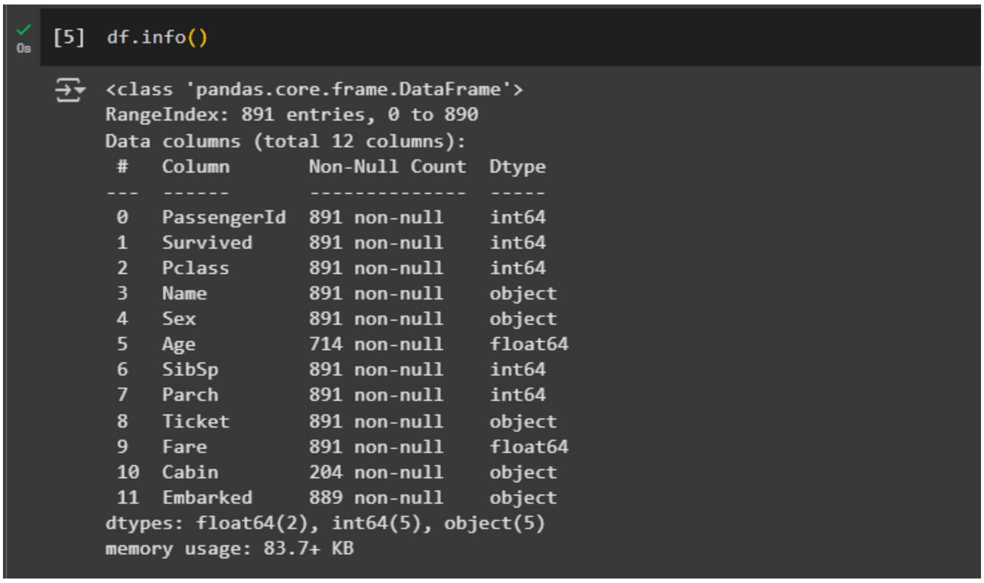
891 rows × 12 columns



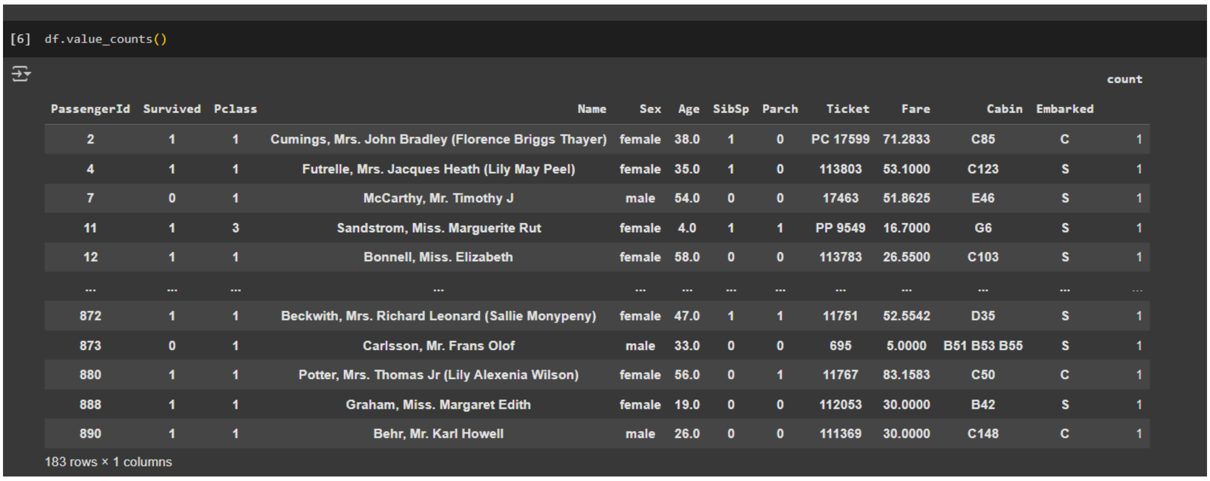
Use .describe(), .info(), .value_counts()

```
df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200



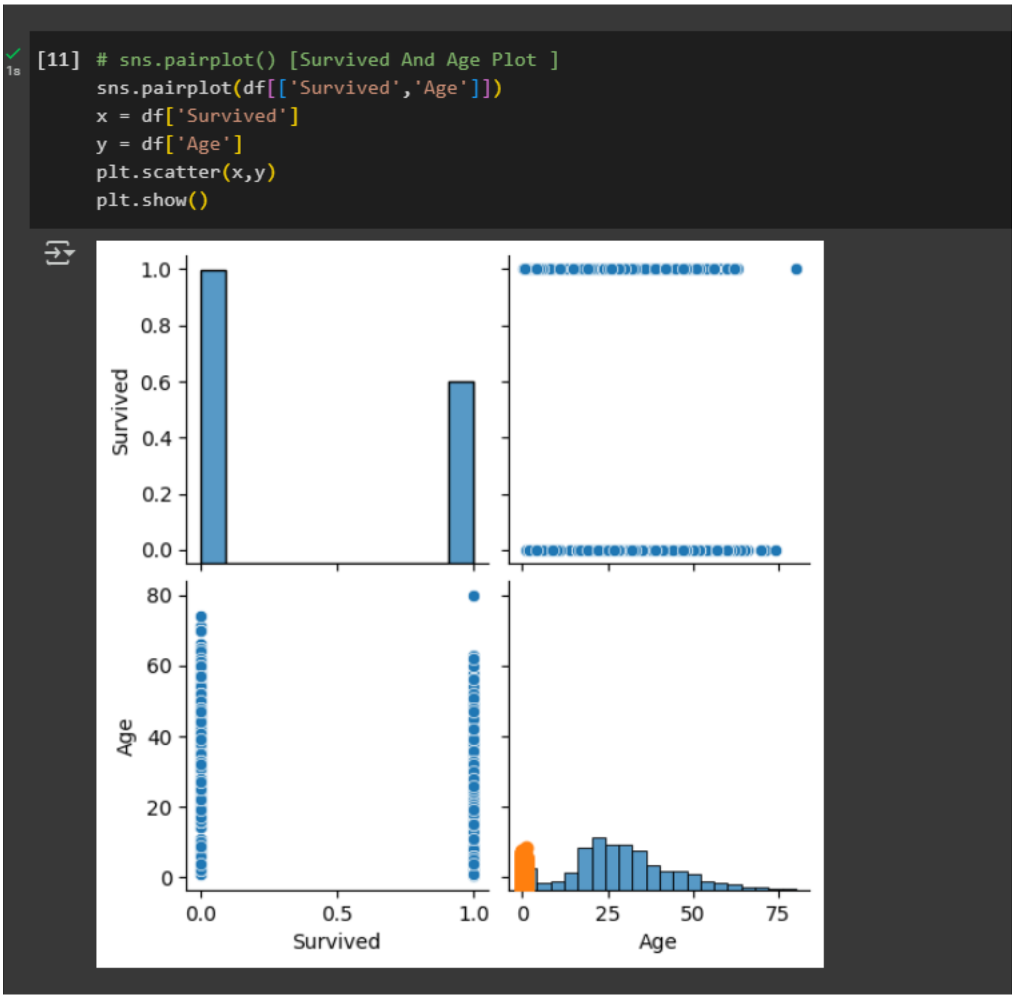
```
[5]  df.info()

<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
```



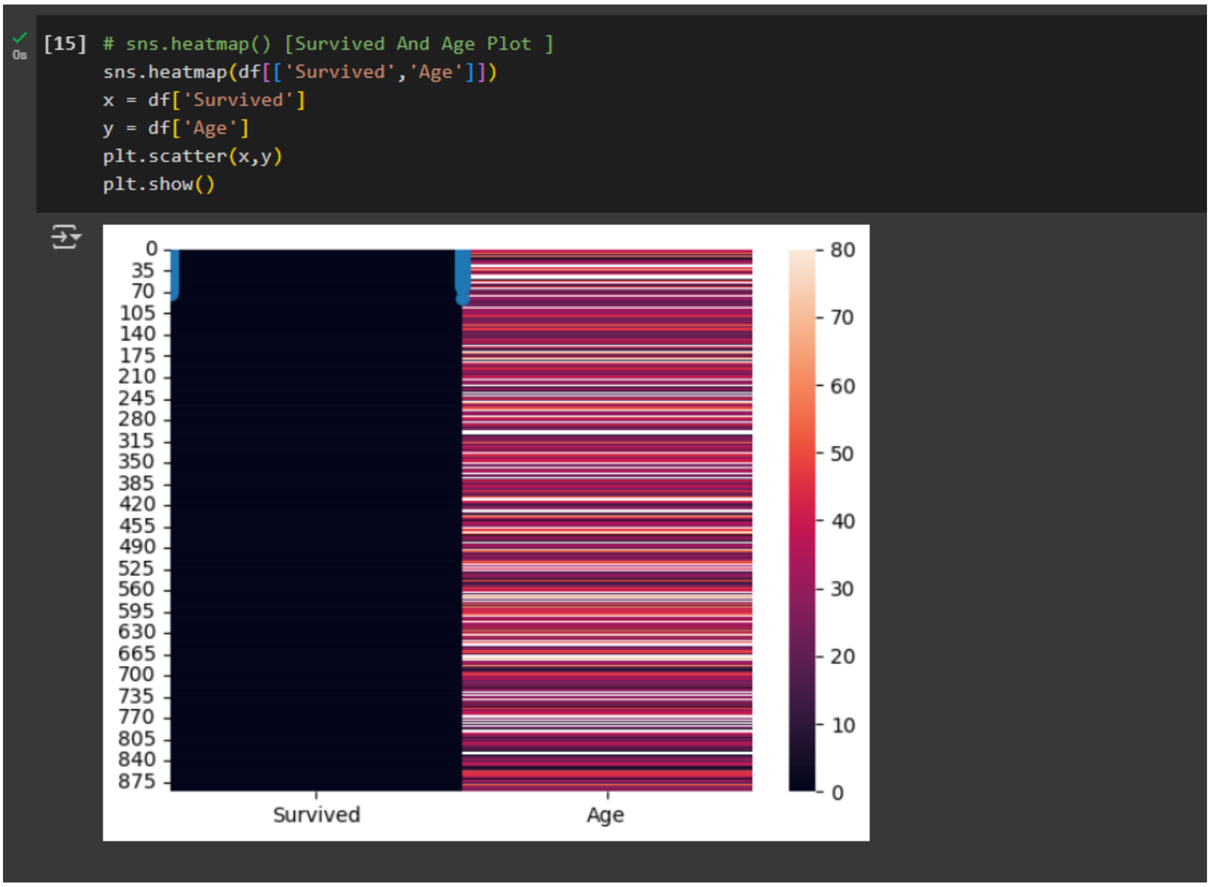
```
[6]  df.value_counts()
```

												count
PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.0	1	0	PC 17599	71.2833	C85	C	1
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	1
7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S	1
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S	1
12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S	1
...	...	...	...	...	...	...	...	...	...	...	...	...
872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S	1
873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0000	B51 B53 B55	S	1
880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C	1
888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S	1
890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C	1

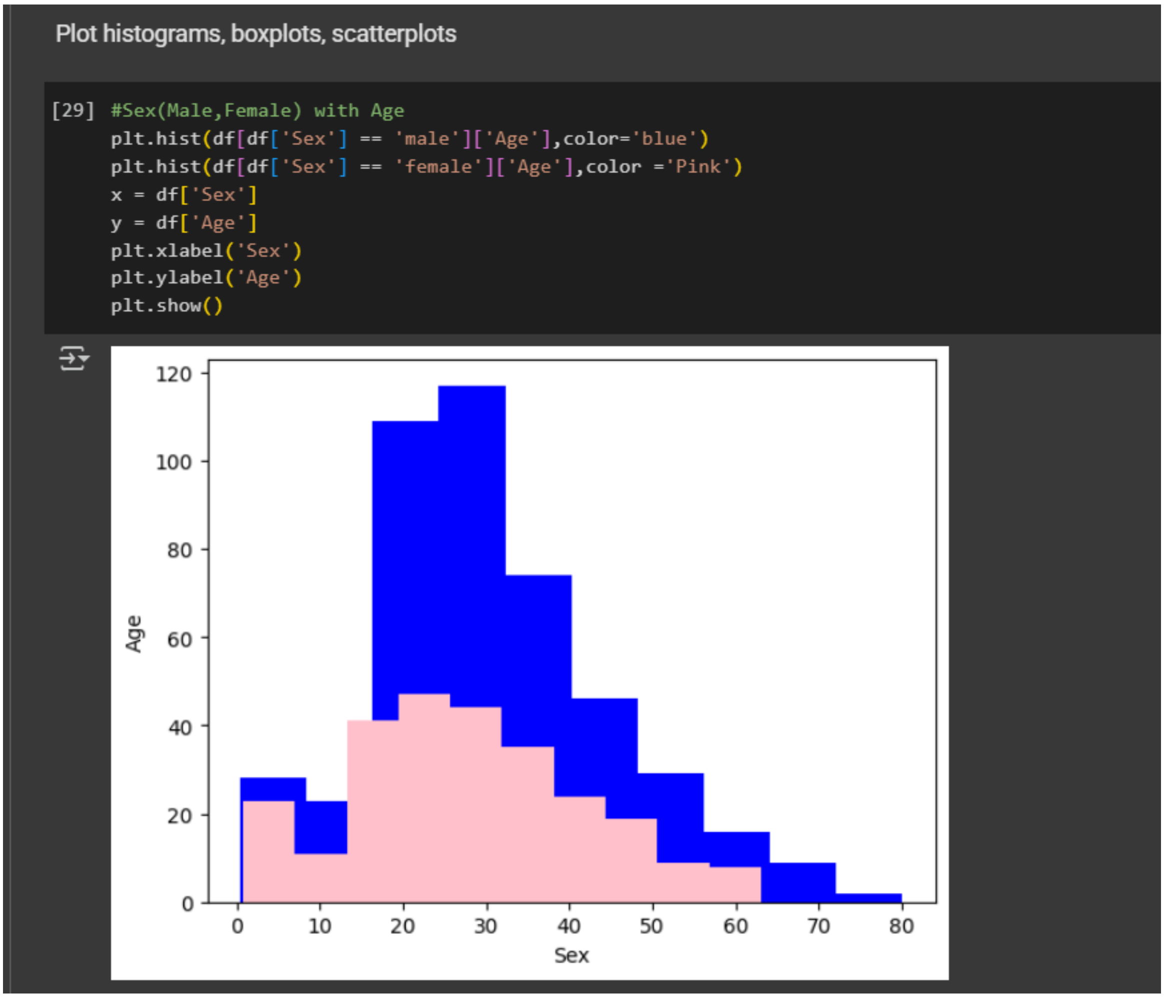
183 rows × 1 columns



```
[11] # sns.pairplot() [Survived And Age Plot ]
     sns.pairplot(df[['Survived','Age']])
     x = df['Survived']
     y = df['Age']
     plt.scatter(x,y)
     plt.show()
```

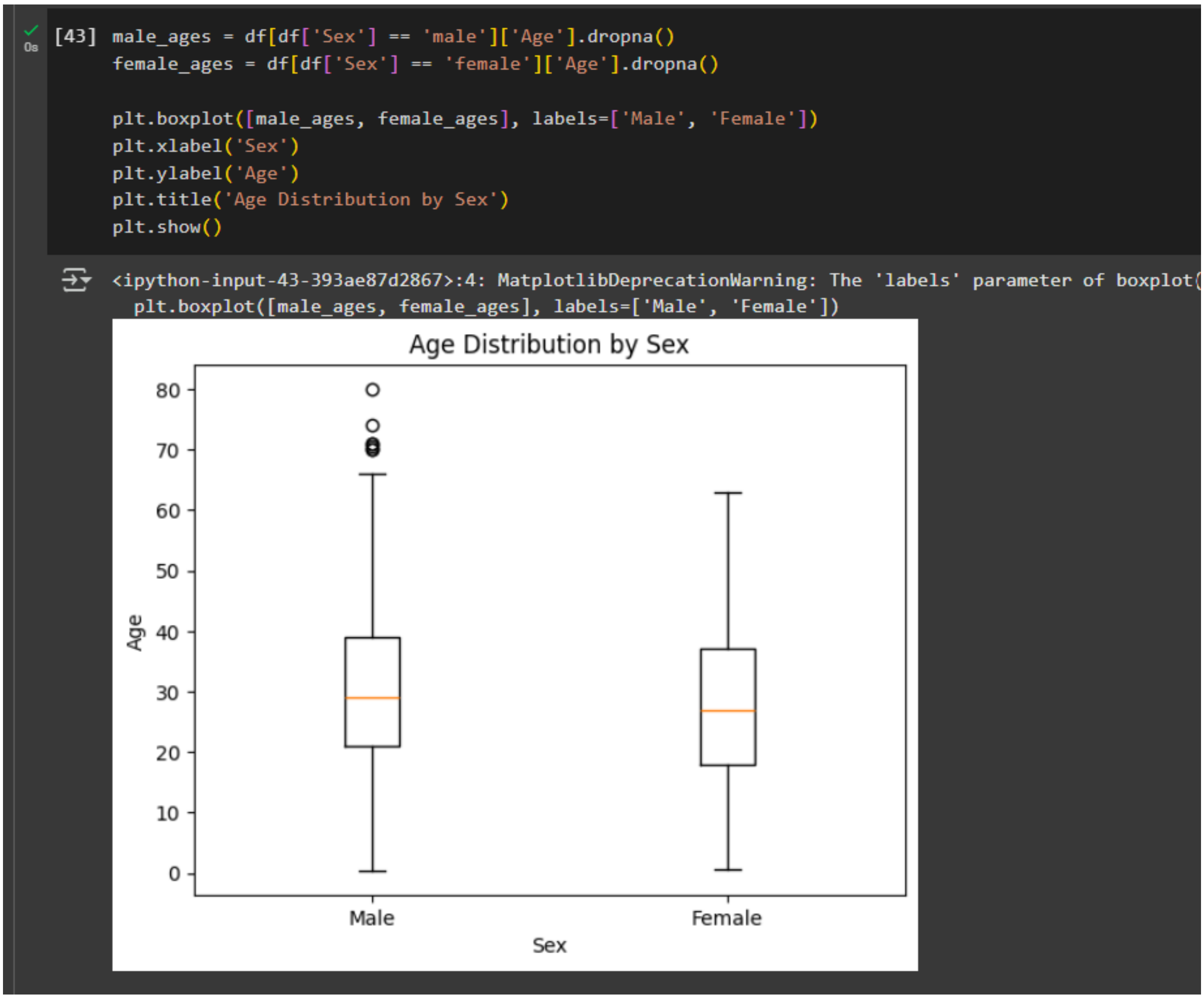


```
# sns.heatmap() [Survived And Age Plot ]
sns.heatmap(df[['Survived','Age']])
x = df['Survived']
y = df['Age']
plt.scatter(x,y)
plt.show()
```



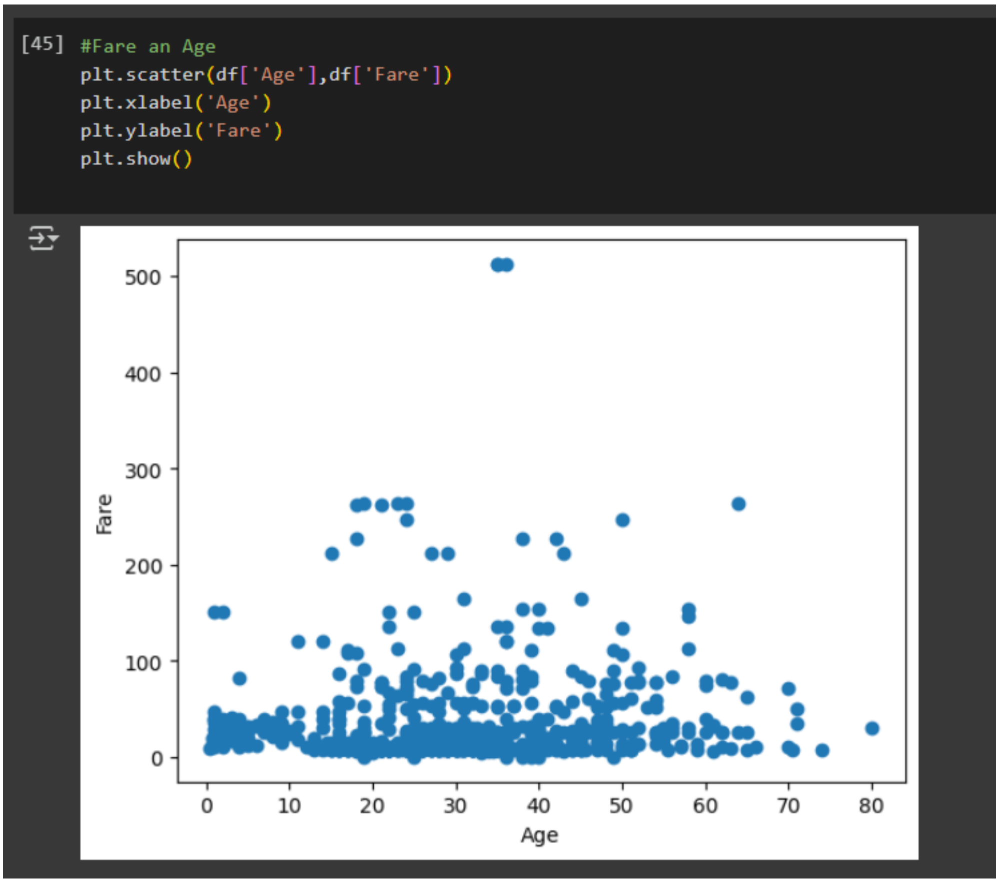
Plot histograms, boxplots, scatterplots

```
[29] #Sex(Male,Female) with Age
     plt.hist(df[df['Sex'] == 'male']['Age'],color='blue')
     plt.hist(df[df['Sex'] == 'female']['Age'],color ='Pink')
     x = df['Sex']
     y = df['Age']
     plt.xlabel('Sex')
     plt.ylabel('Age')
     plt.show()
```



```
[43] male_ages = df[df['Sex'] == 'male']['Age'].dropna()
     female_ages = df[df['Sex'] == 'female']['Age'].dropna()

     plt.boxplot([male_ages, female_ages], labels=['Male', 'Female'])
     plt.xlabel('Sex')
     plt.ylabel('Age')
     plt.title('Age Distribution by Sex')
     plt.show()
```

```
<ipython-input-43-393ae87d2867>:4: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot(
  plt.boxplot([male_ages, female_ages], labels=['Male', 'Female'])
```



```
[45] #Fare an Age
     plt.scatter(df['Age'],df['Fare'])
     plt.xlabel('Age')
     plt.ylabel('Fare')
     plt.show()
```

Based on the analysis performed in notebook, here is a summary of the findings:

1.Data Overview: The dataset contains information about Titanic passengers. The describe() and info() methods provided initial insights into the numerical and categorical features, including counts, means, standard deviations, and data types. value_counts() helped in understanding the distribution of unique values in the dataset.

2.Survival and Age:

The pairplot focusing on 'Survived' and 'Age' shows the relationship between these two variables.

The scatter plot of 'Survived' and 'Age' also visualizes this relationship.

The heatmap of 'Survived' and 'Age' provides a correlation perspective.

3.Age Distribution by Sex:

The histograms of 'Age' for 'male' and 'female' passengers show the distribution of ages for each sex.

The boxplot further clarifies the age distribution for males and females, highlighting the median, quartiles, and potential outliers.

4.Fare and Age:

The scatter plot of 'Age' and 'Fare' shows the relationship between the passenger's age and the fare they paid.