

un all ▼

X

PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked

→ 1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,,S

◆ First 5 Rows of the Dataset:

PassengerId	Survived	Pclass	\
0	1	0	3
1	2	1	1
2	3	1	3
3	4	1	1
4	5	0	3

Name	Sex	Age	SibSp	\
------	-----	-----	-------	---



		Name	Sex	Age	SibSp	\
0		Braund, Mr. Owen Harris	male	22.0	1	
1	Cummings, Mrs. John Bradley (Florence Briggs Th... <td></td> <td>female</td> <td>38.0</td> <td>1</td> <td></td>		female	38.0	1	
2		Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)		female	35.0	1	
4		Allen, Mr. William Henry	male	35.0	0	

	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
2	0	STON/O2. 3101282	7.9250	NaN	S
3	0		53.1000	C123	S
4	0	373450	8.0500	NaN	S

• Dataset 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)

In [1]:

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

dtypes: float64(2), int64(5), object(5)

memory usage: 83.7+ KB

None

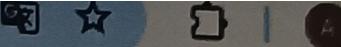
◆ Missing Values:

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

◆ Statistical Summary:

	PassengerId	Survived	Pclass	Age	SibSp	\
count	891.000000	891.000000	891.000000	714.000000	891.000000	
mean	446.000000	0.383838	2.308642	29.699118	0.523008	
std	257.353842	0.486592	0.836071	14.526497	1.102743	
min	1.000000	0.000000	1.000000	0.420000	0.000000	
25%	223.500000	0.000000	2.000000	20.125000	0.000000	
50%	446.000000	0.000000	3.000000	28.000000	0.000000	
75%	668.500000	1.000000	3.000000	38.000000	1.000000	
max	891.000000	1.000000	3.000000	80.000000	8.000000	



help



Share

Gemini

	Parch	Fare
count	891.000000	891.000000
mean	0.381594	32.204208
std	0.806057	49.693429
min	0.000000	0.000000
25%	0.000000	7.910400
50%	0.000000	14.454200
75%	0.000000	31.000000
max	6.000000	512.329200

/tmp/ipython-input-4151661480.py:20: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values a

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method

```
df['Age'].fillna(df['Age'].mean(), inplace=True)
```

/tmp/ipython-input-4151661480.py:21: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values a

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method

```
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
```

to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
~ work because the intermediate object on which we are setting values always behaves as a copy.

ing 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

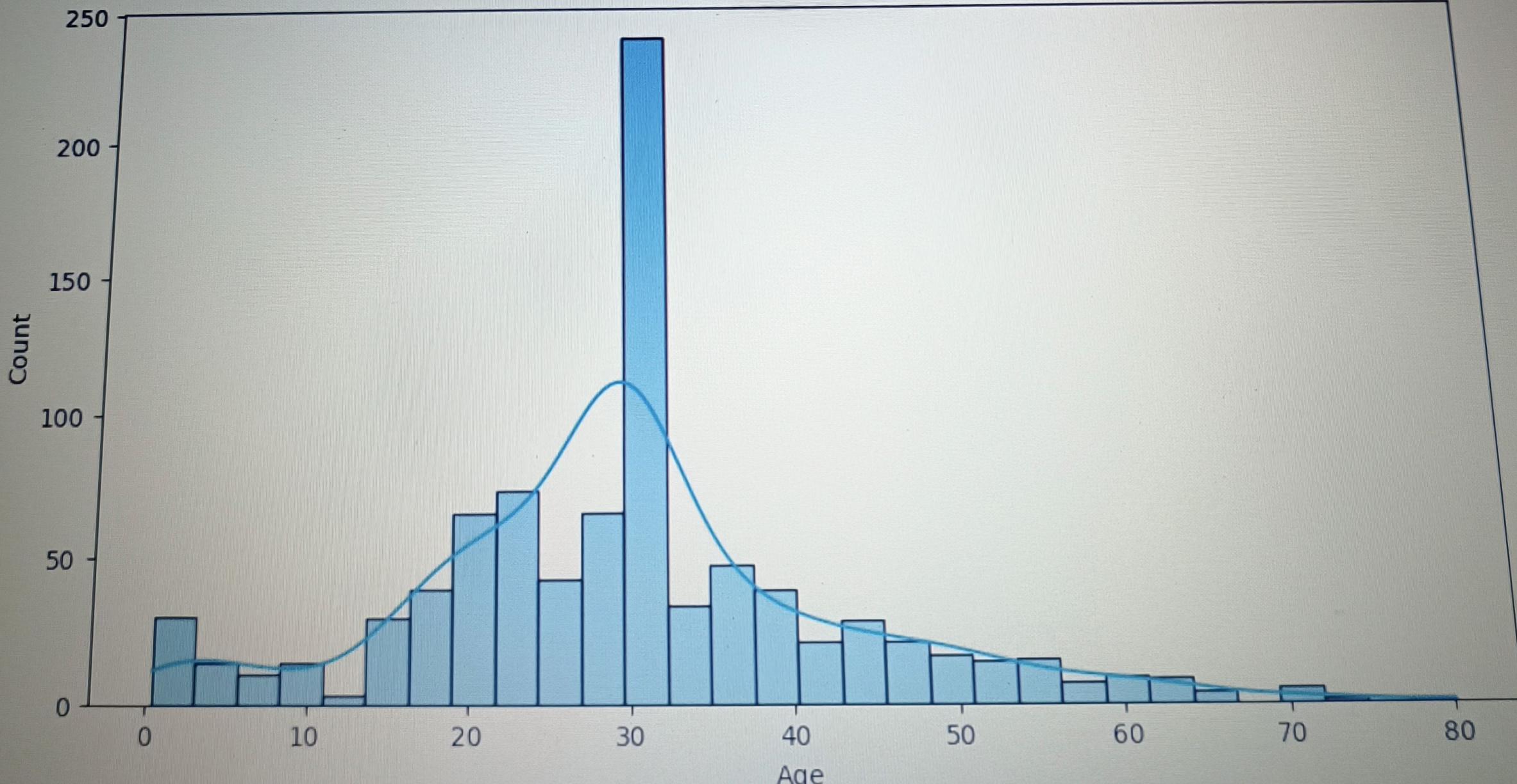
to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
~ work because the intermediate object on which we are setting values always behaves as a copy.

ing 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)'.

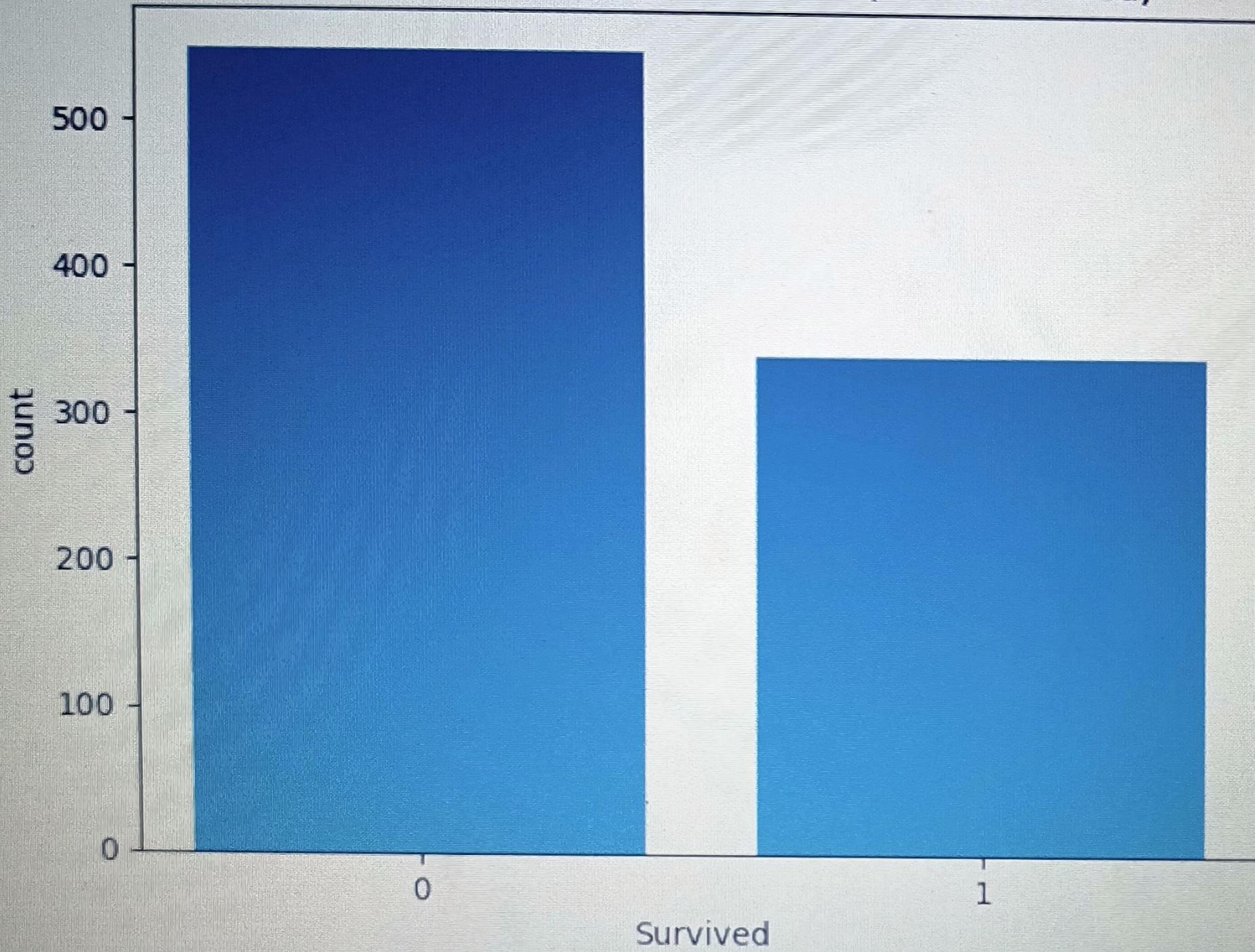
```
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
```

Distribution of Age

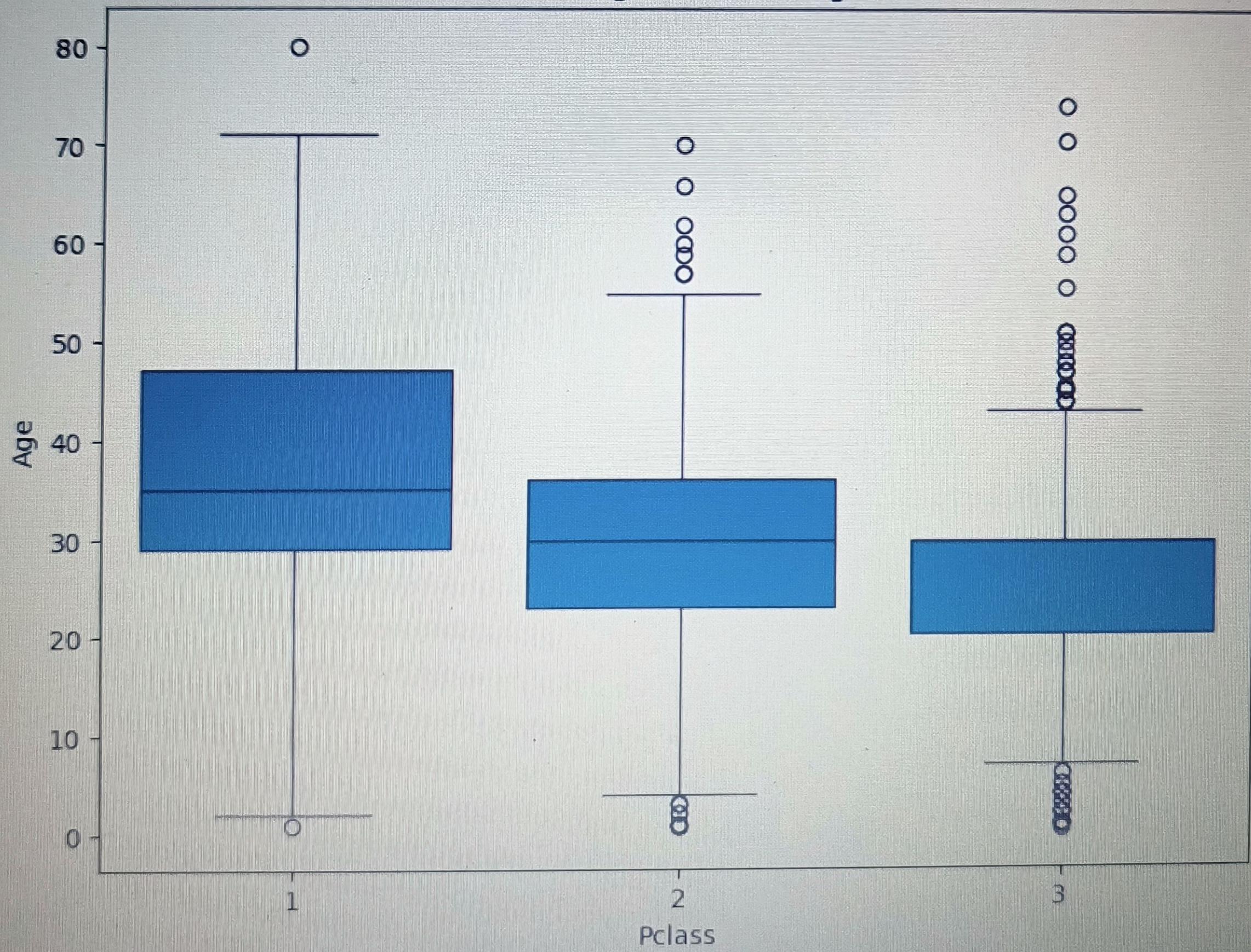


Age

Survival Count (0 = Not Survived, 1 = Survived)



Passenger Class vs Age



Correlation Heatmap





◆ Survival Rate by Passenger Class:

Pclass

1 0.629630
2 0.472826
3 0.242363

Name: Survived, dtype: float64

◆ Survival Rate by Gender:

Sex

female 0.742038
male 0.188908

Name: Survived, dtype: float64