In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
titanic_data = pd.read_csv(r"C:\Users\hp\Downloads\titanic (1).csv")
```

In [3]:
```python
titanic_data
```

Out[3]:

	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

In [4]:
```python
titanic_data.head()
```

Out[4]:

	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

In [5]:
```python
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     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [6]:
```python
#Handle missing values using mean mode median imputation
# Calculate Mean
mean = titanic_data["Age"].mean()
mean
```

Out[6]:
```
29.69911764705882
```

In [7]:
```python
#Calculate Median
median = titanic_data['Age'].median()
median
```

Out[7]:
```
28.0
```

In [8]:
```python
#Calculate mode
mode=titanic_data['Age'].mode()[0]
mode
```

Out[8]:
```
24.0
```

In [9]:
```python
titanic_data.describe()
```

Out[9]:

	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

In [10]:
```python
#Random Sample Imputation
df = pd.read_csv(r'C:\Users\hp\Downloads\titanic (1).csv',usecols=['Age','Fare','Survived'])
df.head()
```

Out[10]:

	Survived	Age	Fare
0	0	22.0	7.2500
1	1	38.0	71.2833
2	1	26.0	7.9250
3	1	35.0	53.1000
4	0	35.0	8.0500

In [11]:
```python
#Check null values
titanic_data.isnull().sum()
```

Out[11]:
```
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
```

In [12]:
```python
#There is 177 null values are present in Age feature.
#Now, we replace this NaN by using Random Sample Imputation
titanic_data['Age'].dropna().sample()
```

Out[12]:
```
115    21.0
Name: Age, dtype: float64
```

In [13]:
```python
#- where .dropna() function drop all nan values in that feature and .sample() function return any one random value
titanic_data['Age'].dropna().sample(df['Age'].isnull().sum())
```

Out[13]:
```
889    26.0
172     1.0
230    35.0
106    21.0
588    22.0
       ...
90     29.0
16      2.0
382    32.0
12     20.0
726    30.0
Name: Age, Length: 177, dtype: float64
```

In [14]:
```python
#- This function check where is null values are present and replace Nan with random sample
#- We use random_state because it replace NaN with specific value only ( if we not use random_state then values change evrytime when we run .)
titanic_data['Age'].dropna().sample(titanic_data['Age'].isnull().sum(),random_state=0)
```

Out[14]:
```
423    28.00
177    50.00
305     0.92
292    36.00
889    26.00
       ...
539    22.00
267    25.00
352    15.00
99     34.00
689    15.00
Name: Age, Length: 177, dtype: float64
```

In [15]:
```python
#rows and columns
titanic_data.shape
```

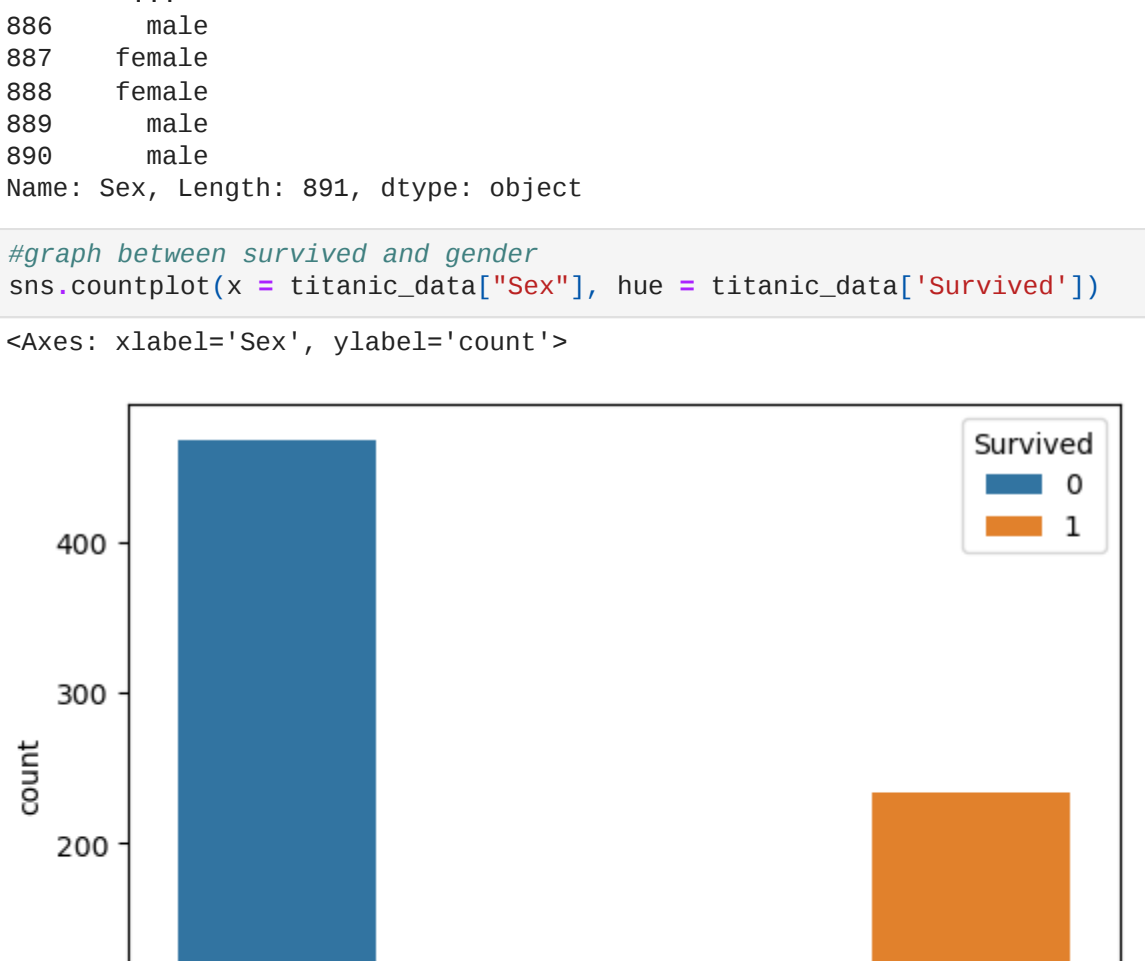
Out[15]:
```
(891, 12)
```

In [16]:
```python
#count of males and females survived
titanic_data['Survived'].value_counts()
```

Out[16]:
```
Survived
0    549
1    342
Name: count, dtype: int64
```

In [17]:
```python
#graph between survived and pclass
sns.countplot(x = titanic_data['Survived'], hue = titanic_data['Pclass'])
```

Out[17]:
```
<Axes: xlabel='Survived', ylabel='count'>
```

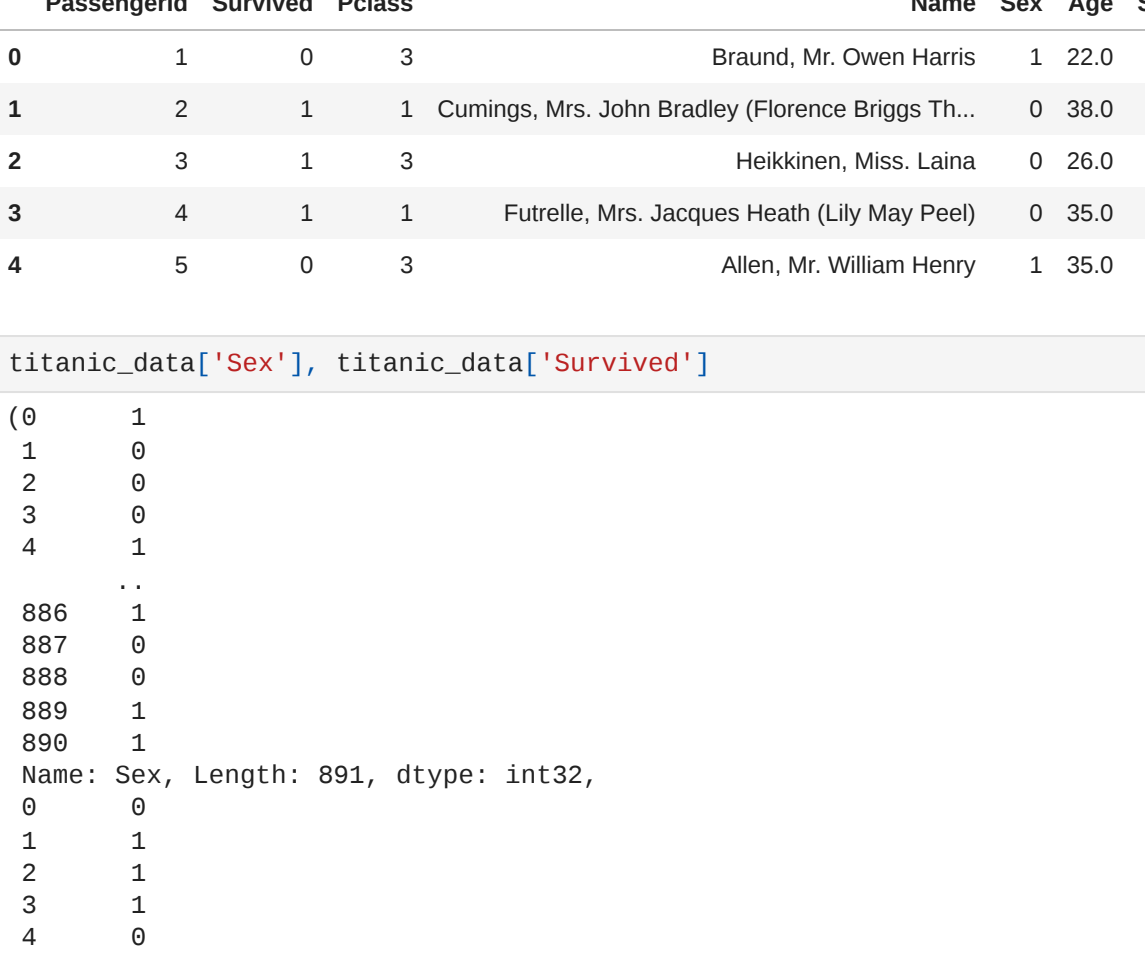


In [18]:
```python
titanic_data["Sex"]
```

Out[18]:
```
0        male
1      female
2      female
3      female
4        male
        ...
886      male
887    female
888    female
889      male
890      male
Name: Sex, Length: 891, dtype: object
```

In [19]:
```python
#graph between survived and gender
sns.countplot(x = titanic_data["Sex"], hue = titanic_data['Survived'])
```

Out[19]:
```
<Axes: xlabel='Sex', ylabel='count'>
```



In [20]:
```python
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()

titanic_data['Sex'] = labelencoder.fit_transform(titanic_data['Sex'])

titanic_data.head()
```

Out[20]:

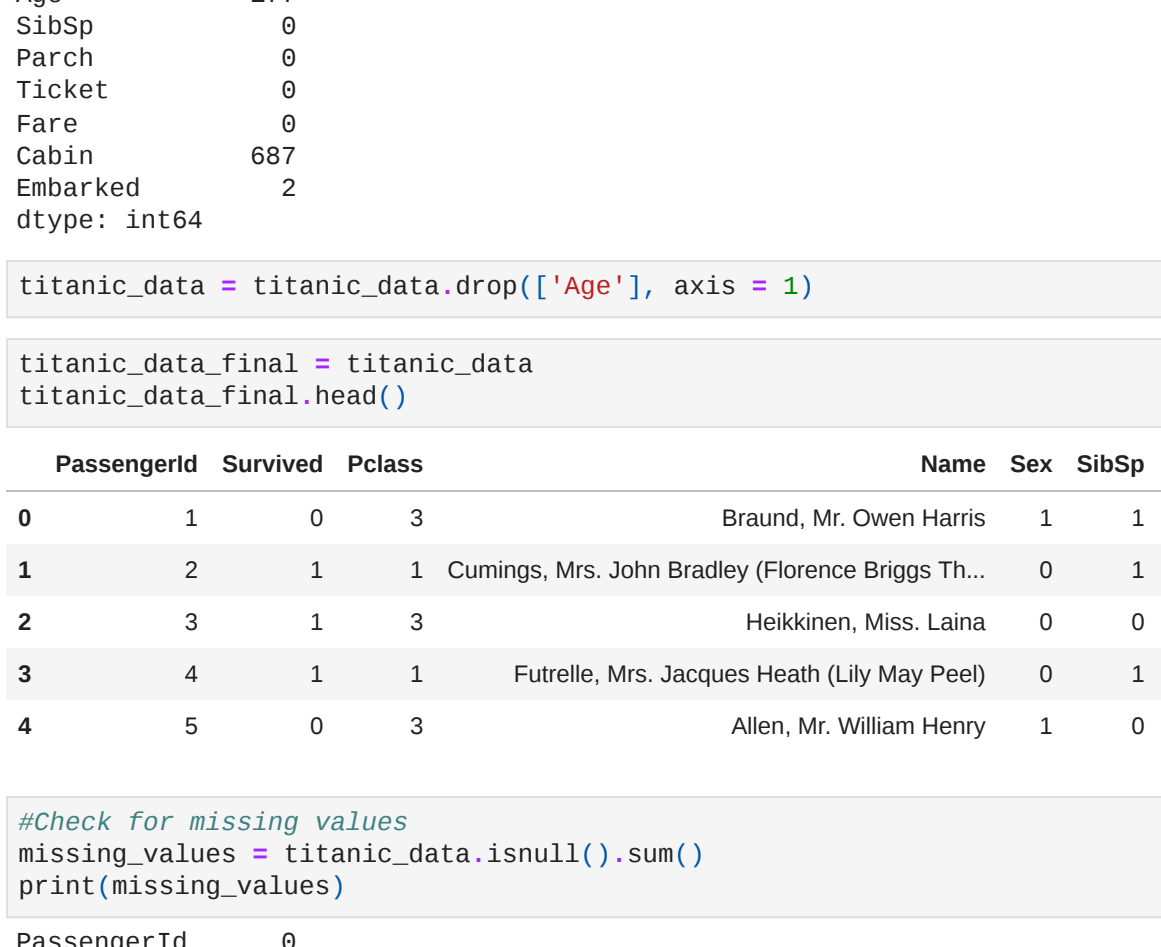
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.9250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

In [21]:
```python
titanic_data['Sex'], titanic_data['Survived']
```

Out[21]:
```
(0      1
 1      0
 2      0
 3      0
 4      1
       ..
 886    1
 887    0
 888    0
 889    1
 890    1
 Name: Sex, Length: 891, dtype: int32,
 0      0
 1      1
 2      1
 3      1
 4      0
       ..
 886    0
 887    1
 888    0
 889    1
 890    0
 Name: Survived, Length: 891, dtype: int64)
```

In [22]:
```python
sns.countplot(x = titanic_data['Sex'], hue = titanic_data['Survived'])
```

Out[22]:
```
<Axes: xlabel='Sex', ylabel='count'>
```



In [23]:
```python
titanic_data.isna().sum()
```

Out[23]:
```
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
```

In [24]:
```python
titanic_data = titanic_data.drop(['Age'], axis = 1)
```

In [25]:
```python
titanic_data_final = titanic_data
titanic_data_final.head()
```

Out[25]:

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

In [26]:
```python
#Check for missing values
missing_values = titanic_data.isnull().sum()
print(missing_values)
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [27]:
```python
#Percentage of missing values
missing_percentage = (titanic_data.isnull().sum() / len(titanic_data)) * 100
print(missing_percentage)
```

```
PassengerId     0.000000
Survived        0.000000
Pclass          0.000000
Name            0.000000
Sex             0.000000
SibSp           0.000000
Parch           0.000000
Ticket          0.000000
Fare            0.000000
Cabin          77.104377
Embarked        0.224467
dtype: float64
```

In [28]:
```python
#Visualize the missing data
sns.heatmap(titanic_data.isnull(), cmap = 'viridis', cbar = False)
plt.show()
```

In [29]:
```python
#Handling the missing values data
#Here, for numerical value fill it with mean and for categorical value fill it with 'unknown'
numerical_columns = titanic_data.select_dtypes(include = ['int64', 'float64']).columns
titanic_data[numerical_columns]= titanic_data[numerical_columns].fillna(titanic_data[numerical_columns].mean())
```

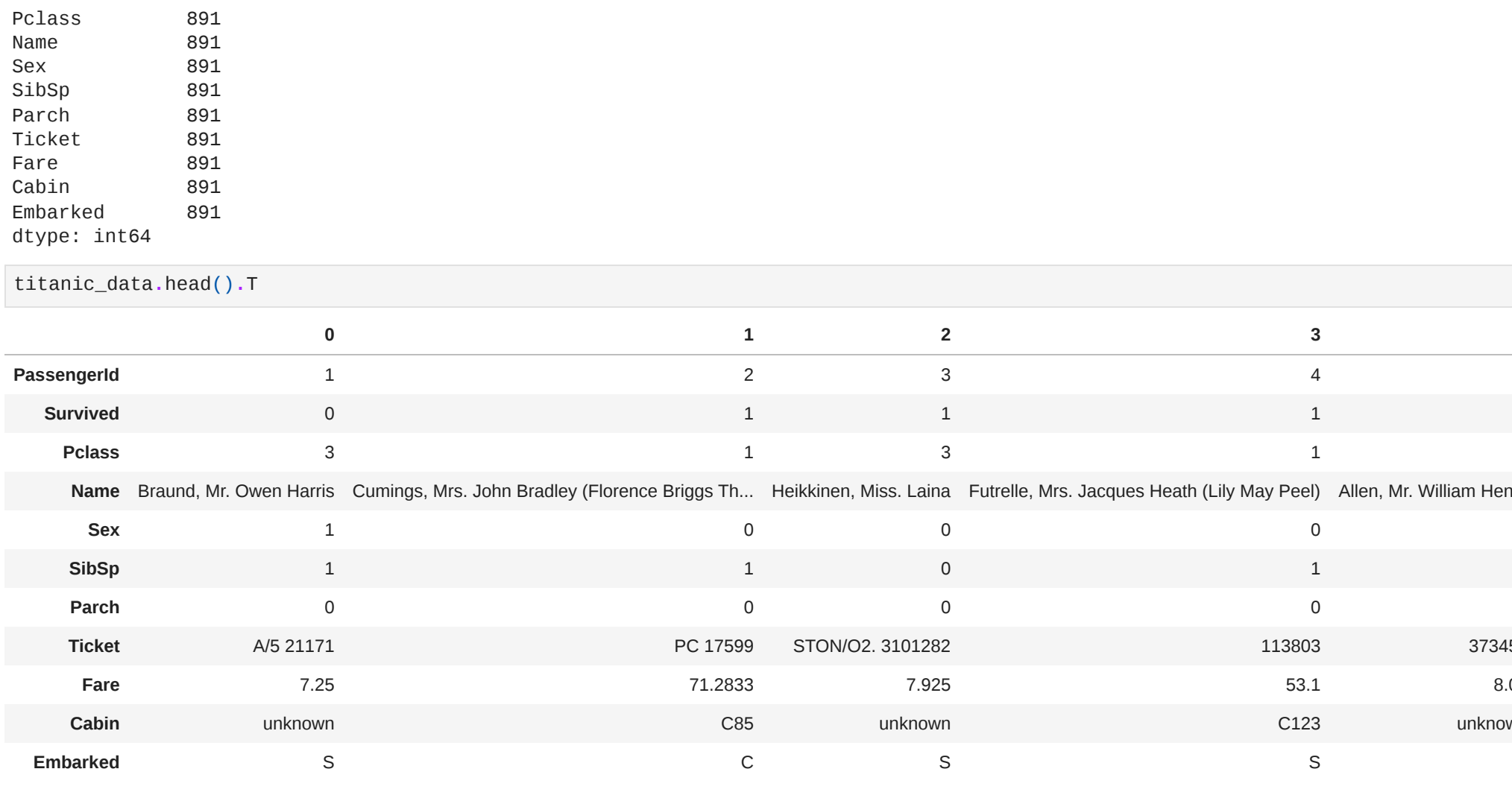
In [30]:
```python
categorical_columns = titanic_data.select_dtypes(include = 'object').columns
titanic_data[categorical_columns] = titanic_data[categorical_columns].fillna('unknown')
```

In [31]:
```python
print(titanic_data.isnull().sum())
```

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

In [32]:
```python
plt.figure(figsize=(12,6))
titanic_data.isnull().sum().plot(color='r')
```

Out[32]:
```
<Axes: >
```



In [33]:
```python
titanic_data.count() #as now number of elements in each column are equal that means there is no missing value
```

Out[33]:
```
PassengerId    891
Survived       891
Pclass         891
Name           891
Sex            891
SibSp          891
Parch          891
Ticket         891
Fare           891
Cabin          891
Embarked       891
dtype: int64
```

In [34]:
```python
titanic_data.head().T
```

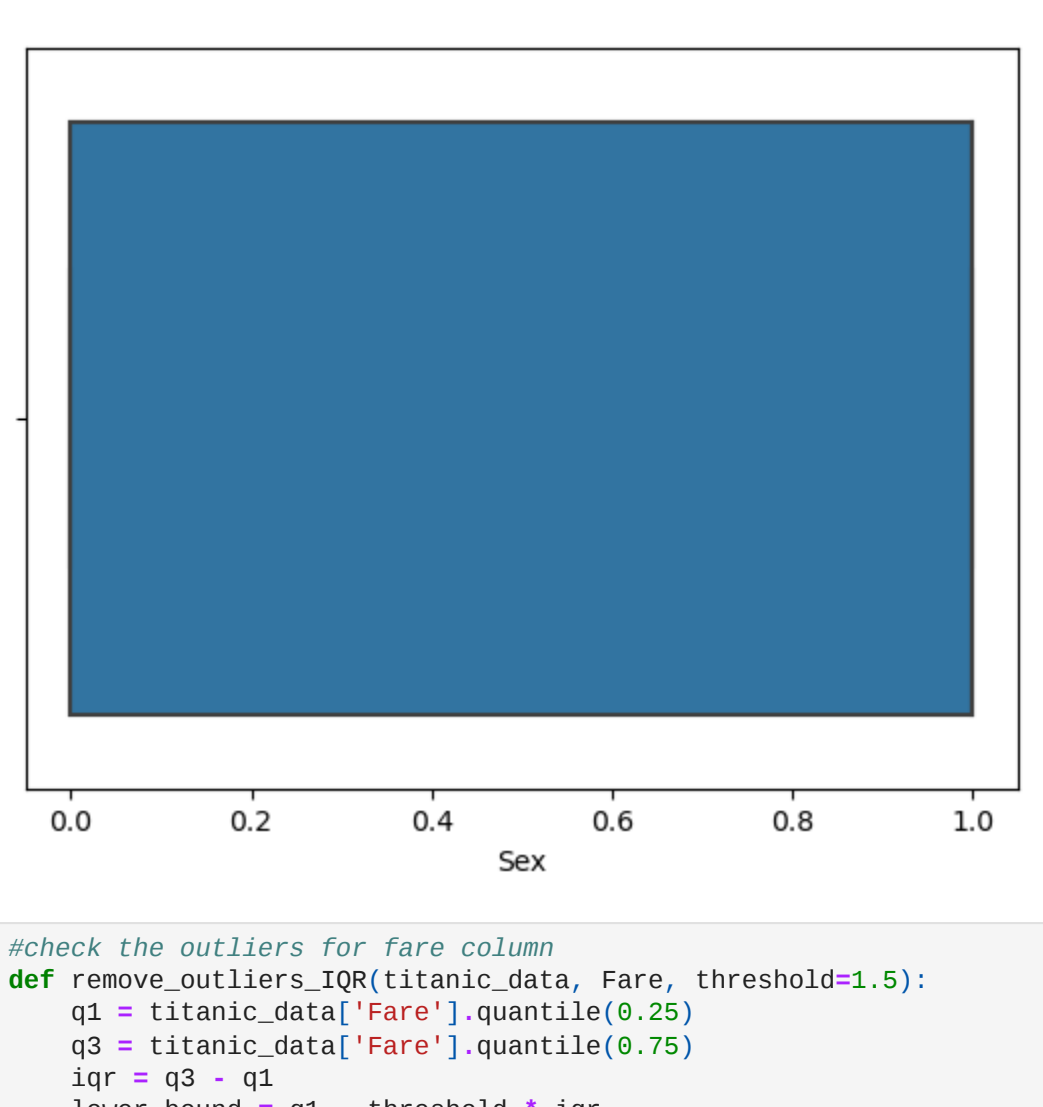
Out[34]:

	0	1	2	3	4
PassengerId	1	2	3	4	5
Survived	0	1	1	1	0
Pclass	3	1	3	1	3
Name	Braund, Mr. Owen Harris	Cumings, Mrs. John Bradley (Florence Briggs Th...	Heikkinen, Miss. Laina	Futrelle, Mrs. Jacques Heath (Lily May Peel)	Allen, Mr. William Henry
Sex	1	0	0	0	1
SibSp	1	1	0	1	0
Parch	0	0	0	0	0
Ticket	A/5 21171	PC 17599	STON/O2. 3101282	113803	373450
Fare	7.25	71.2833	7.925	53.1	8.05
Cabin	unknown	C85	unknown	C123	unknown
Embarked	S	C	S	S	S

In [35]:
```python
#save the dataset
titanic_data.to_csv('usedcleaned_dataset_csv', index = False)
```

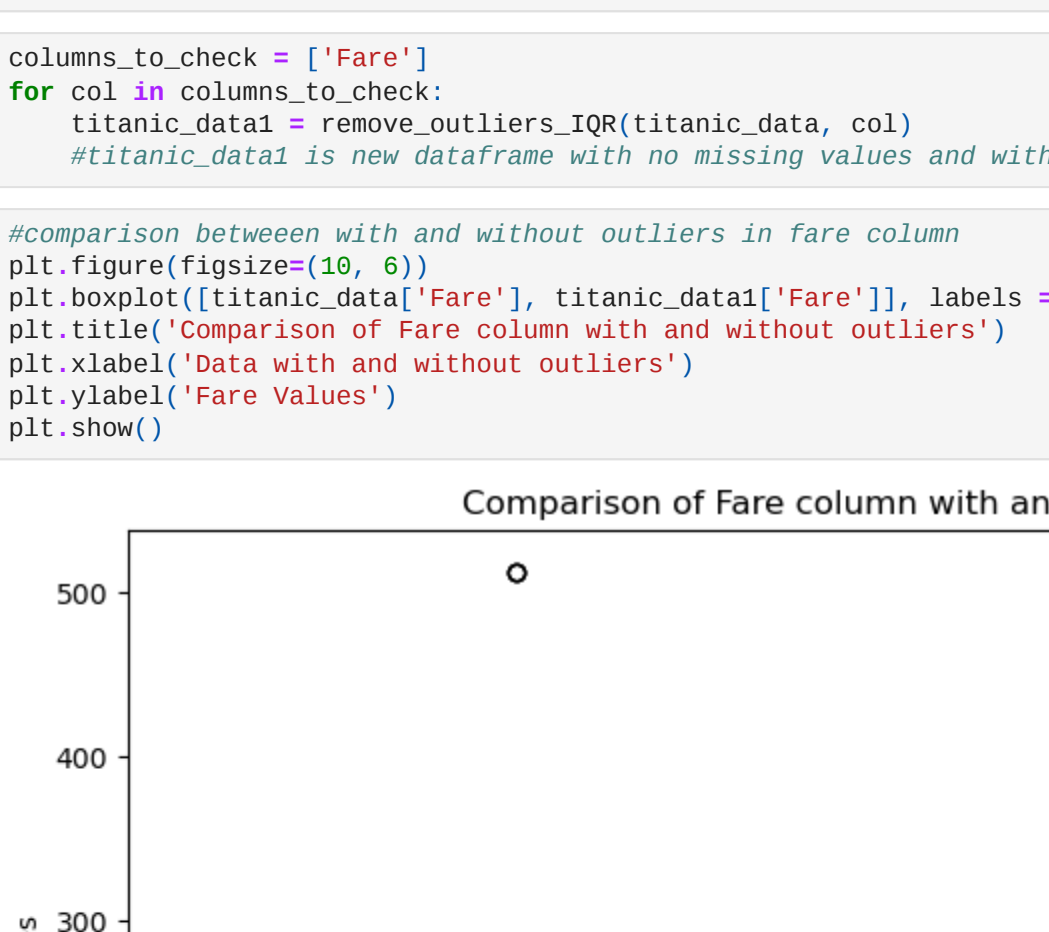
In [36]:
```python
#visualization of outliers in dataset
sns.boxplot(x  =titanic_data.Fare, showfliers = False)
```

Out[36]:
```
<Axes: xlabel='Fare'>
```



In [37]:
```python
sns.boxplot(x=titanic_data.Sex, showfliers=False)
```

Out[37]:
```
<Axes: xlabel='Sex'>
```



In [38]:
```python
#check the outliers for fare column
def remove_outliers_IQR(titanic_data, Fare, threshold=1.5):
    q1 = titanic_data['Fare'].quantile(0.25)
    q3 = titanic_data['Fare'].quantile(0.75)
    iqr = q3 - q1
    lower_bound = q1 - threshold * iqr
    upper_bound = q3 + threshold * iqr
    return titanic_data[(titanic_data['Fare'] >= lower_bound) & (titanic_data['Fare'] <= upper_bound)]
```

In [39]:
```python
columns_to_check = ['Fare']
for col in columns_to_check:
    titanic_data1 = remove_outliers_IQR(titanic_data, col)
    #titanic_data1 is new dataframe with no missing values and without outliers in fare column
```

In [40]:
```python
#comparison betweeen with and without outliers in fare column
plt.figure(figsize=(10, 6))
plt.boxplot([titanic_data['Fare'], titanic_data1['Fare']], labels = ['With Outliers', 'Without Outliers'])
plt.title('Comparison of Fare column with and without outliers')
plt.xlabel('Data with and without outliers')
plt.ylabel('Fare Values')
plt.show()
```

Model Training

In [41]:
```python
X = titanic_data[['Pclass', 'Sex']]
Y = titanic_data['Survived']
```

In [42]:
```python
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 0)
```

In [43]:
```python
from sklearn.linear_model import LogisticRegression

log = LogisticRegression(random_state = 0)
log.fit(X_train, Y_train)
```

Out[43]:

```
LogisticRegression
LogisticRegression(random_state=0)
```

Model Prediction

In [44]:
```python
pred = print(log.predict(X_test))
```

```
[0 0 0 1 1 0 1 1 0 1 0 1 0 1 1 1 0 0 0 0 0 1 0 0 1 1 0 1 1 1 0 1 0 0 0 0 0
 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 1 0 1 0 1 1 1 0 0 0
 0 1 0 0 0 0 0 1 0 0 1 1 1 1 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 1 1 1 1 0 1 0
 1 0 1 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 1 1 1 0 1
 1 0 0 1 1 0 1 0 1 0 1 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0]
```

In [45]:
```python
print(Y_test)
```

```
495    0
648    0
278    0
31     1
255    1
      ..
780    1
837    0
215    1
833    0
372    0
Name: Survived, Length: 179, dtype: int64
```

In [46]:
```python
import warnings
warnings.filterwarnings("ignore")

res = log.predict([[2,0]])

if(res==0):
    print("So Sorry! Not Survived")
else:
    print("Survived")
```

```
Survived
```

In [ ]: