Data_analysis (copy)

June 14, 2024

1 Titanic Dataset Analysis

Your task is to apply all the methods (including visualization, preprocessing, dropping the columns, etc). The purpose of this project is use various visualizations which can help us to understand the dataset and find the some hidden information

Hint:

You can drop the name, passengerID, etc which are general information

Maybe check how many survived and how many died

Check for null values and maybe handle if any

Check how dies most, male or female

What aged people died the most

And other usefull information

[1]:
```
import pandas as pd

data=pd.read_csv("titanic.csv")

data.head()
```

[1]:
```
   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  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  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            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

[3]:
```
# it gives the name of each columns of th dataset
data.columns
```

[3]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

[4]:
```
# find the shape of the data frame

data.shape
```

[4]:
```
(891, 12)
```

[5]:
```
#check for missing values in the dataset
data.isnull().sum()
```

[5]:
```
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
```

[6]:
```
# axis=1 means colum
# axis = 0 meand row
data.drop("Cabin", axis=1, inplace=True)
```

[7]:
```
data.isnull().sum()
```

[7]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
```

```
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Embarked         2
dtype: int64
```

[8]:
```
# fill the missing valus in age with the mean of the dataset
```

[9]:
```
data['Age'].fillna(data['Age'].mean(), inplace=True)
```

[10]:
```
data.isnull().sum()
```

[10]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       2
dtype: int64
```

[11]:
```
len(data)
```

[11]:
```
891
```

[12]:
```
data.dropna(inplace=True)
```

[13]:
```
len(data)
```

[13]:
```
889
```

[ ]:
```

```

[14]:
```
# find the correlation matrix
```

[15]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 889 entries, 0 to 890
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
```

```
---  ------       --------------  -----
 0   PassengerId  889 non-null    int64
 1   Survived     889 non-null    int64
 2   Pclass       889 non-null    int64
 3   Name         889 non-null    object
 4   Sex          889 non-null    object
 5   Age          889 non-null    float64
 6   SibSp        889 non-null    int64
 7   Parch        889 non-null    int64
 8   Ticket       889 non-null    object
 9   Fare         889 non-null    float64
 10  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(4)
memory usage: 83.3+ KB
```

[16]:
```
numeric_columns = data.select_dtypes(exclude=['object']).columns

data_num = data[numeric_columns]
```

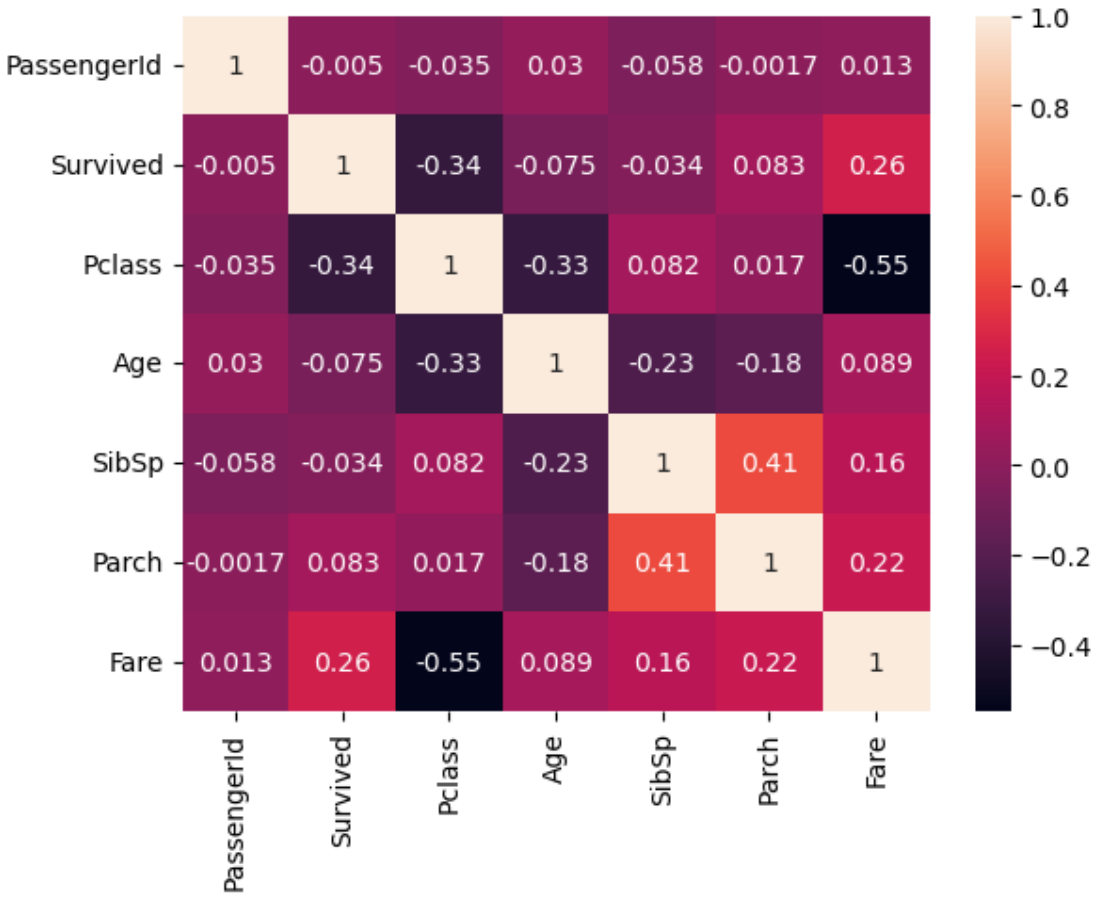
[ ]:
```

```

[17]:
```
import seaborn as sns

sns.heatmap(data_num.corr(), annot=True)
```

[17]: <AxesSubplot:>



[18]:
```
# Analysis
##### Categorical data ---
##### Analyse the regression data ---
```

[19]:
```
data.head()
```

[19]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th…	female	38.0	1
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1
4	5	0	3				

```
4                            Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Embarked
0      0         A/5 21171   7.2500        S
1      0          PC 17599  71.2833        C
2      0  STON/O2. 3101282   7.9250        S
3      0            113803  53.1000        S
4      0            373450   8.0500        S
```

[20]:
```
# Categorical data , one column at a time
```

[21]:
```
data['Survived'].unique()
```

[21]:
```
array([0, 1])
```

[22]:
```
len(data)
```

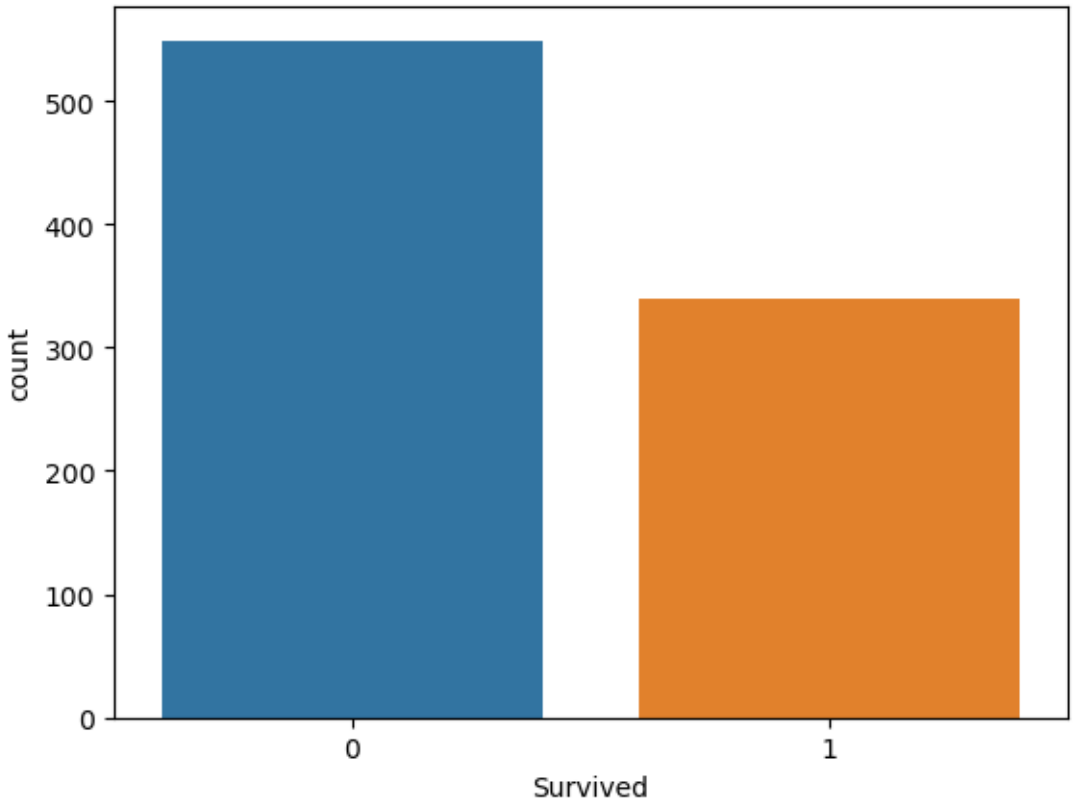
[22]:
```
889
```

[23]:
```
data['Survived'].value_counts()
```

[23]:
```
0    549
1    340
Name: Survived, dtype: int64
```

[24]:
```
sns.countplot(data, x='Survived')
```

[24]:
```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```



[25]:
```
# Pclass
data['Pclass'].unique()
```

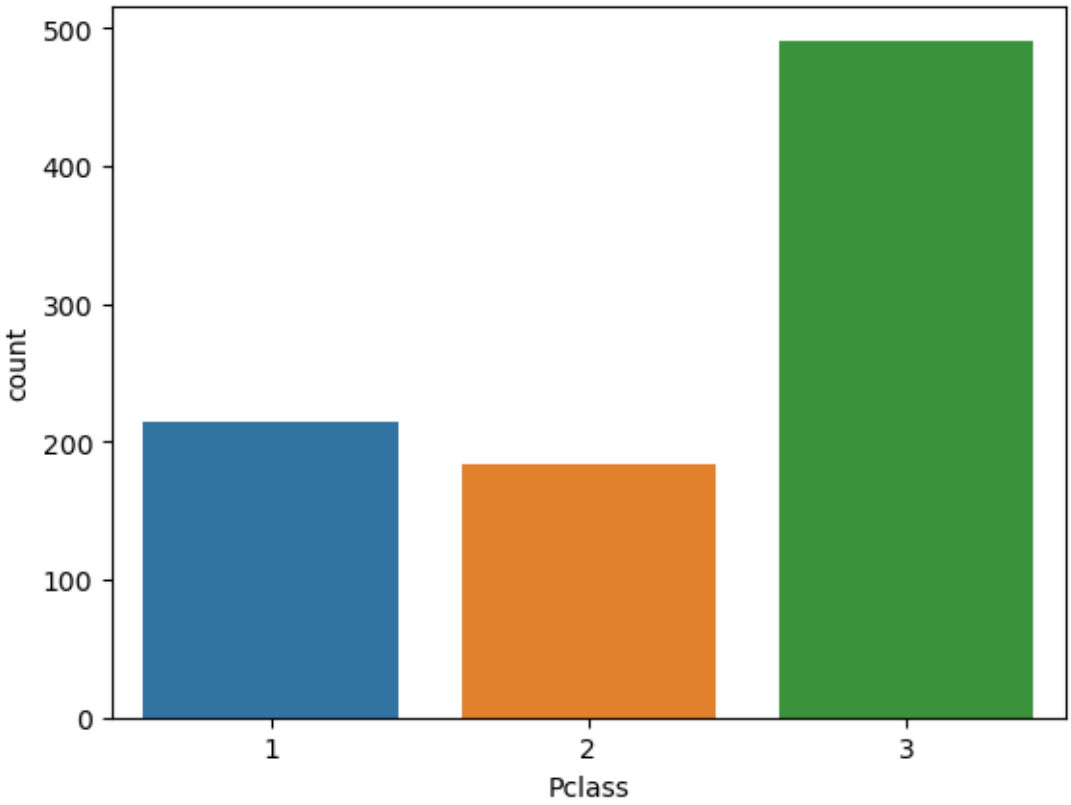
[25]:
```
array([3, 1, 2])
```

[26]:
```
data['Pclass'].value_counts()
```

[26]:
```
3    491
1    214
2    184
Name: Pclass, dtype: int64
```

[27]:
```
sns.countplot(data, x='Pclass')
```

[27]:
```
<AxesSubplot:xlabel='Pclass', ylabel='count'>
```



[28]:
```
# Sex
```

[29]:
```
data['Sex'].unique()
```

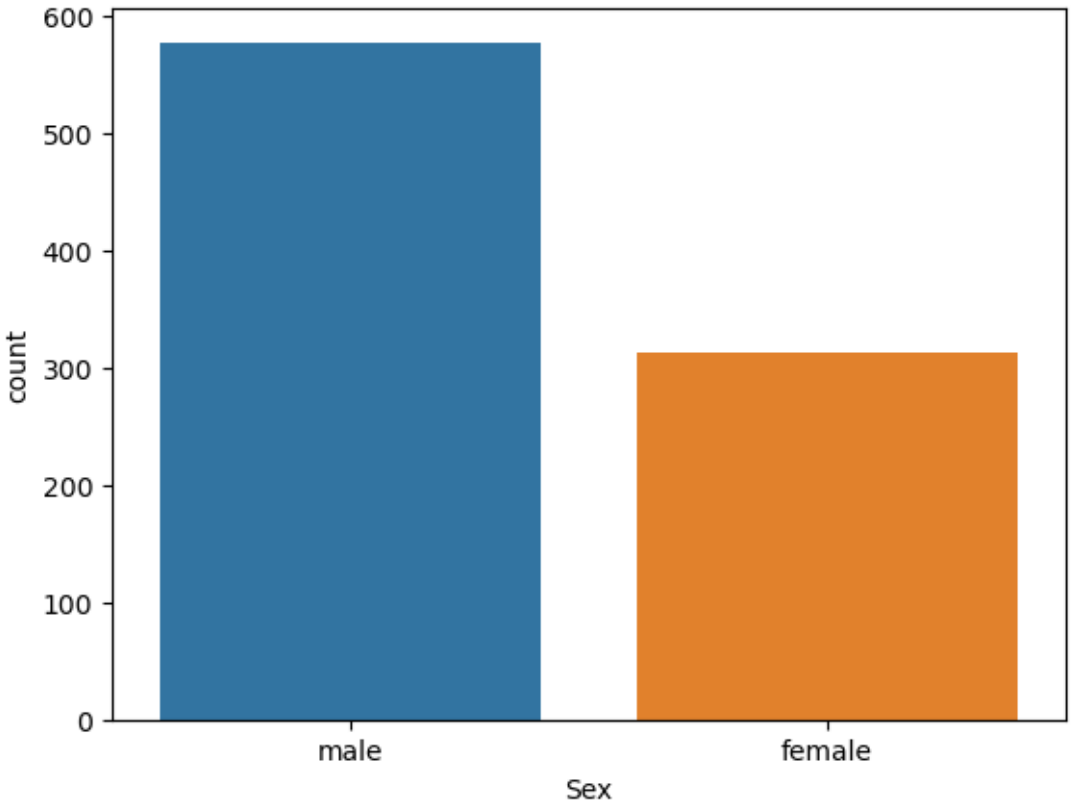
[29]:
```
array(['male', 'female'], dtype=object)
```

[30]:
```
data['Sex'].value_counts()
```

[30]:
```
male      577
female    312
Name: Sex, dtype: int64
```

[31]:
```
sns.countplot(data, x='Sex')
```

[31]:
```
<AxesSubplot:xlabel='Sex', ylabel='count'>
```



[32]:
```
# two columns comparision
```

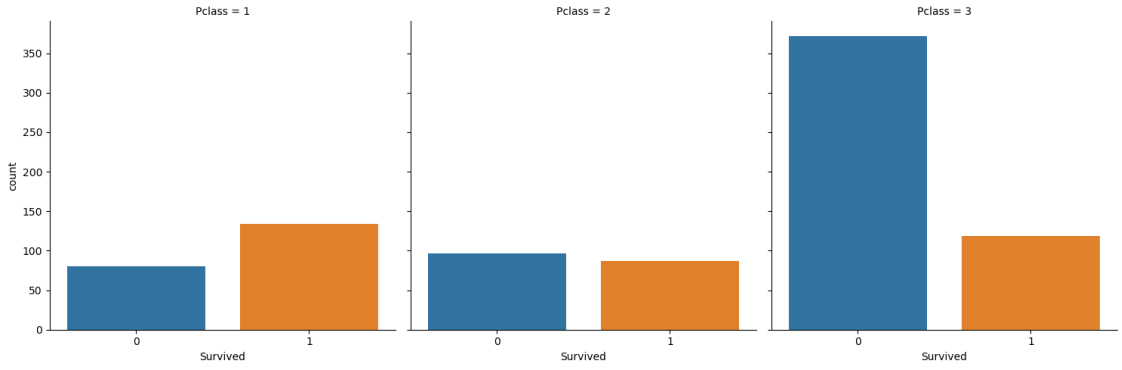
[33]:
```
# campre surivived with pclass
```

[ ]:
```

```

[34]:
```
sns.catplot(data, x='Survived', col='Pclass', kind='count')
```

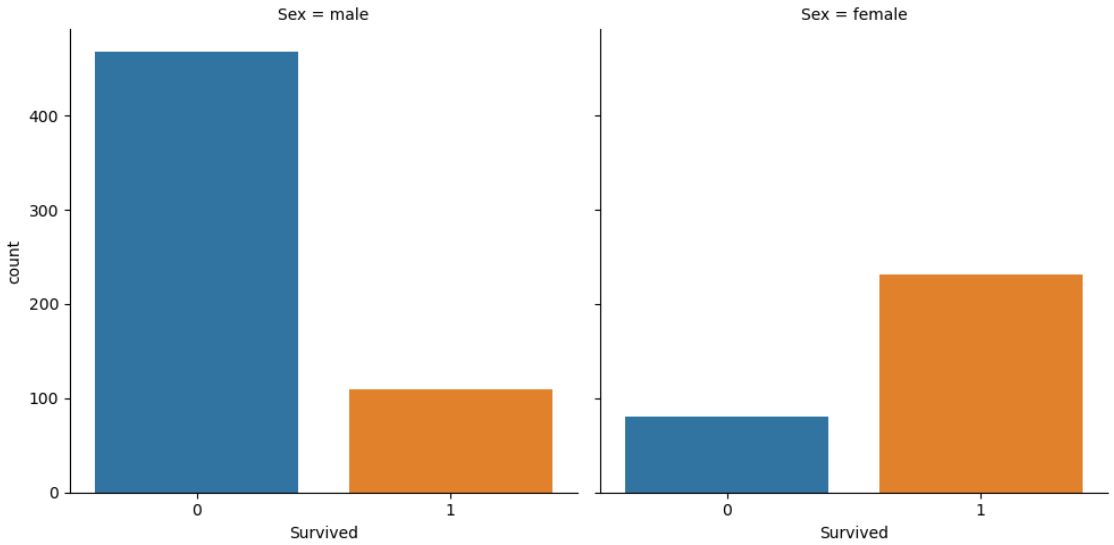
[34]: <seaborn.axisgrid.FacetGrid at 0x79ac66c13100>



[35]:
```
# survived with sex
```

[36]:
```
sns.catplot(data, x='Survived', col='Sex', kind='count')
```

[36]:
```
<seaborn.axisgrid.FacetGrid at 0x79ac6529f610>
```



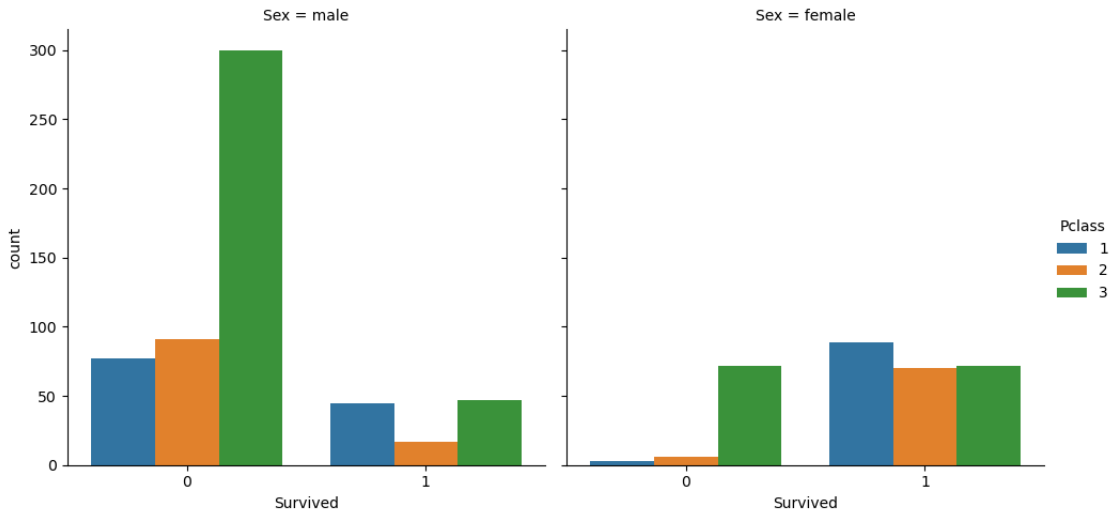
[ ]:
```

```

[37]:
```
# Three column comparision
# survived, sex and plac
```

[38]:
```
sns.catplot(data, x='Survived', col='Sex', kind='count', hue='Pclass')
```

[38]:
```
<seaborn.axisgrid.FacetGrid at 0x79ac68d9bd30>
```

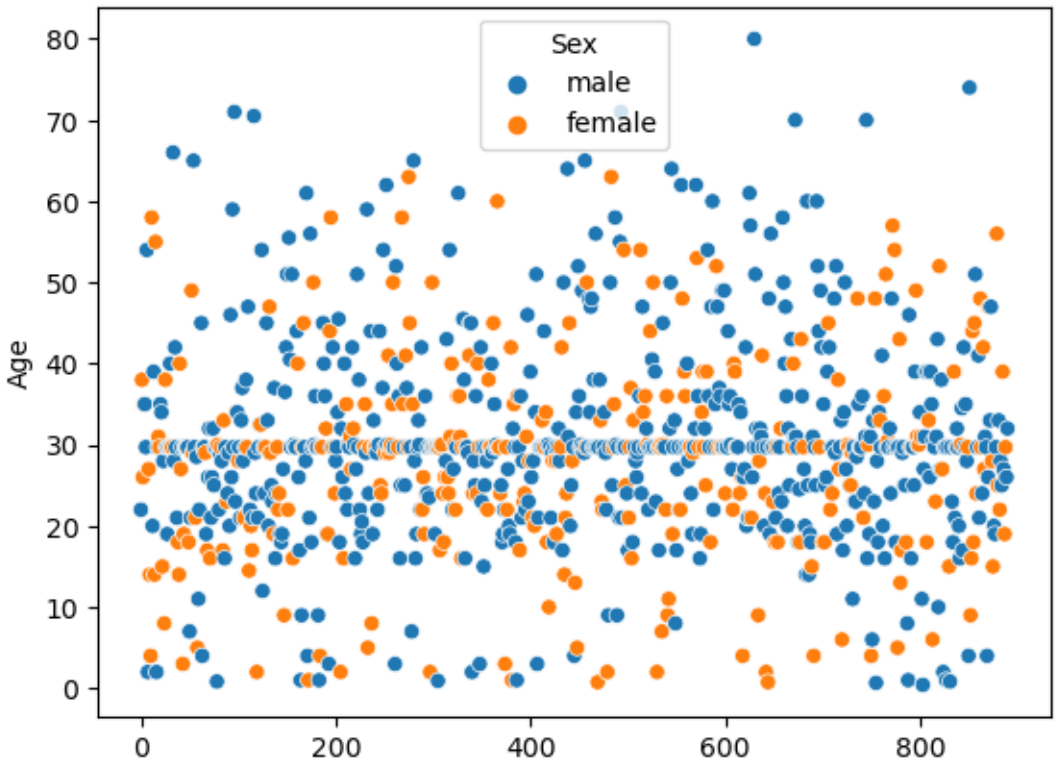


[ ]:

[39]:
```
# scatter chart for the age
```

[40]:
```
sns.scatterplot(data, x=data.index, y='Age', hue='Sex')
```

[40]: <AxesSubplot:ylabel='Age'>



[41]:
```
data.head()
```

[41]:

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

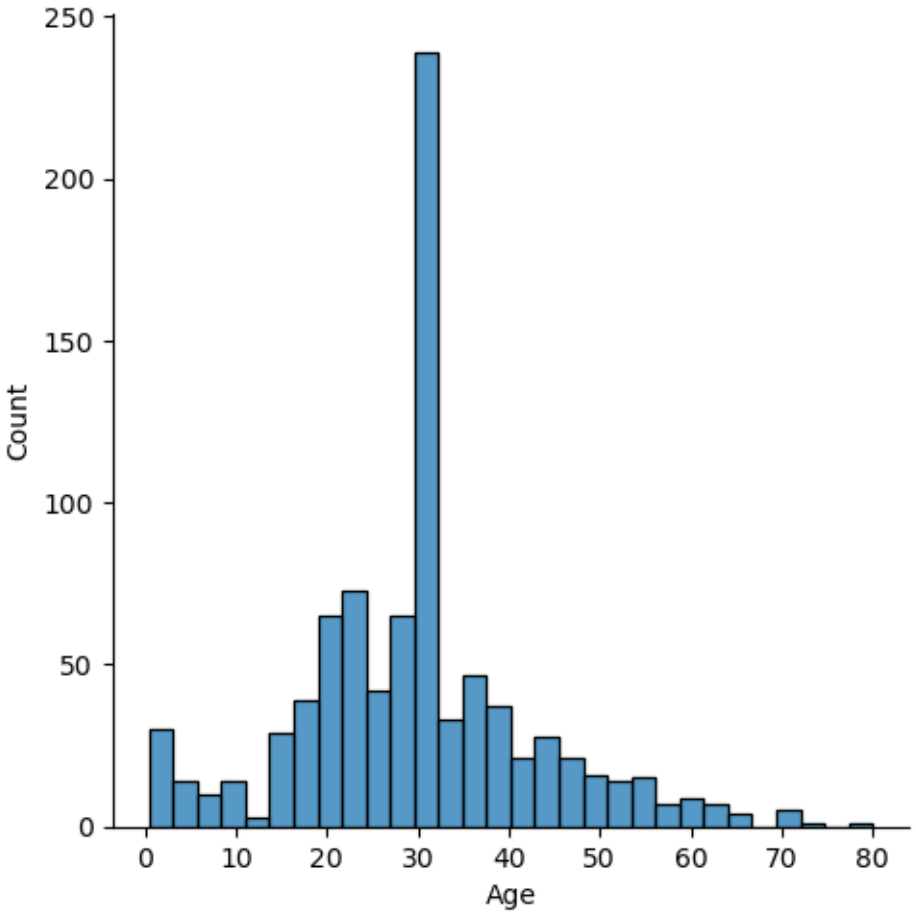
[42]:
```
# how to analyse the regression columns.....
```

[ ]:
```
# Age, Fare ---- Survived
```

[43]:
```
# The distribution of Age
```

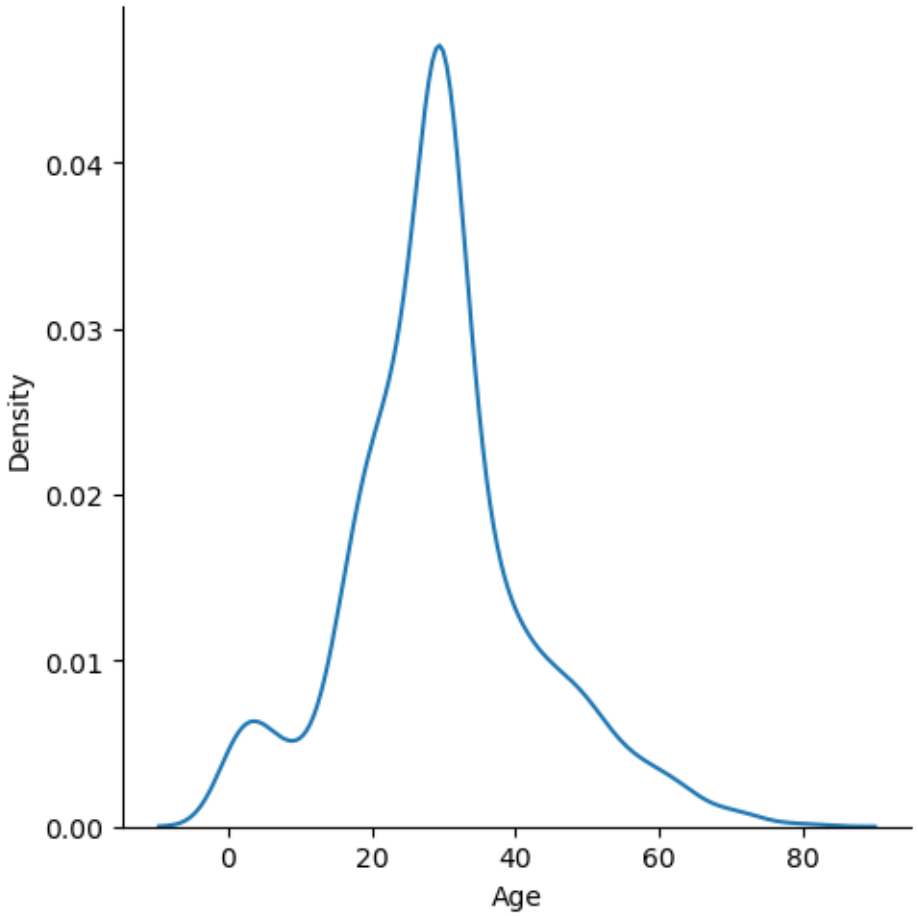
[45]:
```
sns.displot(data, x='Age')
```

[45]:
```
<seaborn.axisgrid.FacetGrid at 0x79ac651fce50>
```



[46]:
```
sns.displot(data, x='Age', kind='kde')
```

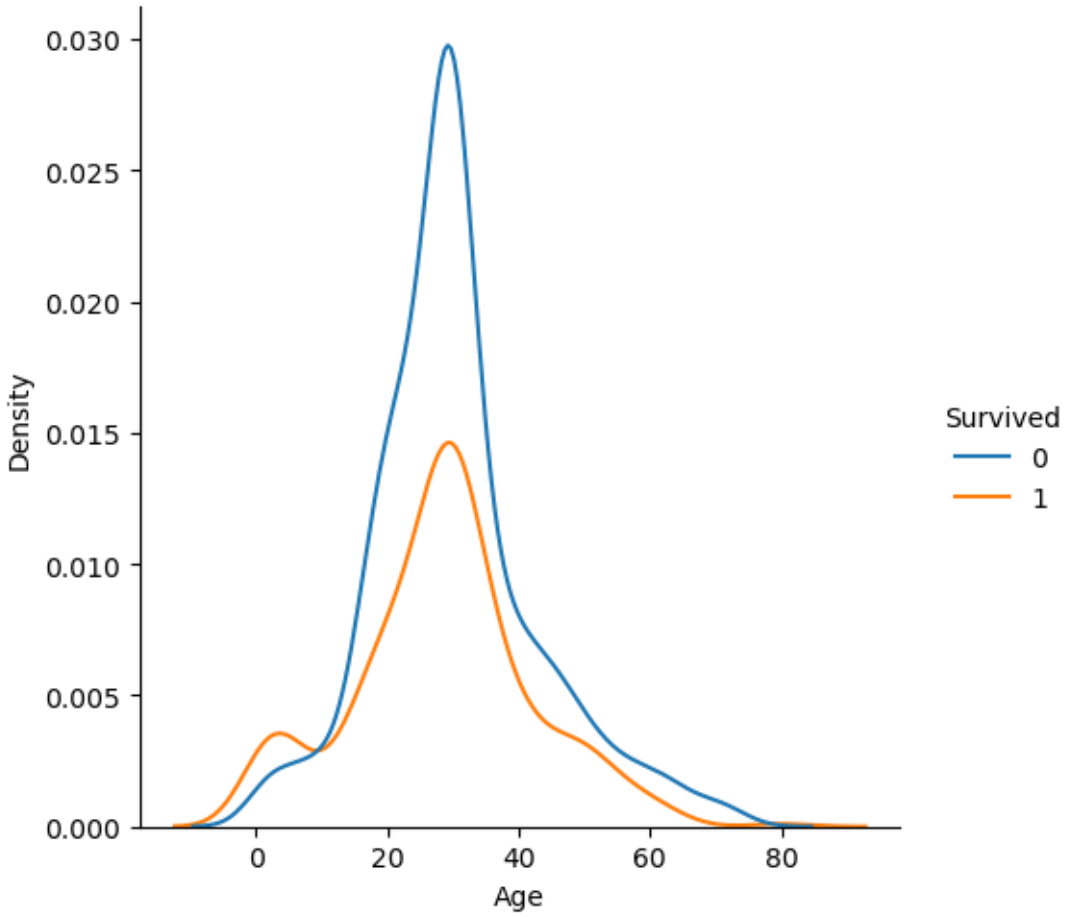
[46]: <seaborn.axisgrid.FacetGrid at 0x79ac64ac7220>



[47]:
```
# relation between age and survived
```

[48]:
```
sns.displot(data, x='Age', kind='kde', hue='Survived')
```

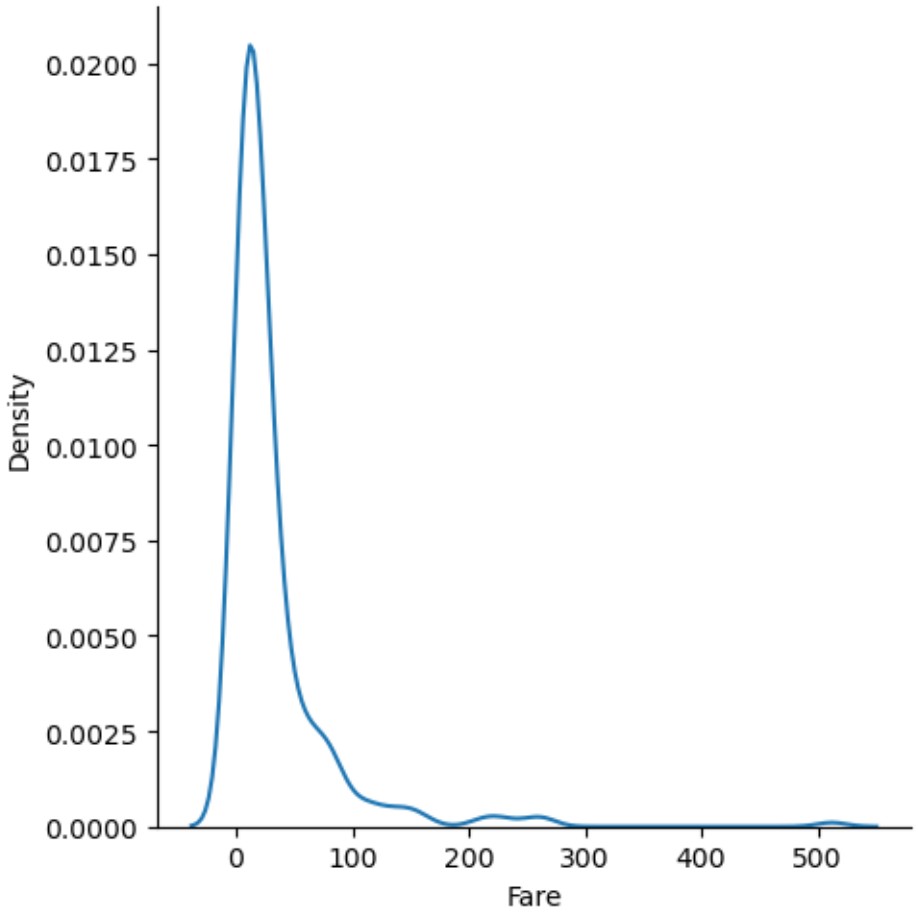
[48]:
```
<seaborn.axisgrid.FacetGrid at 0x79ac66bf0640>
```



[49]:
```
# Fare
```

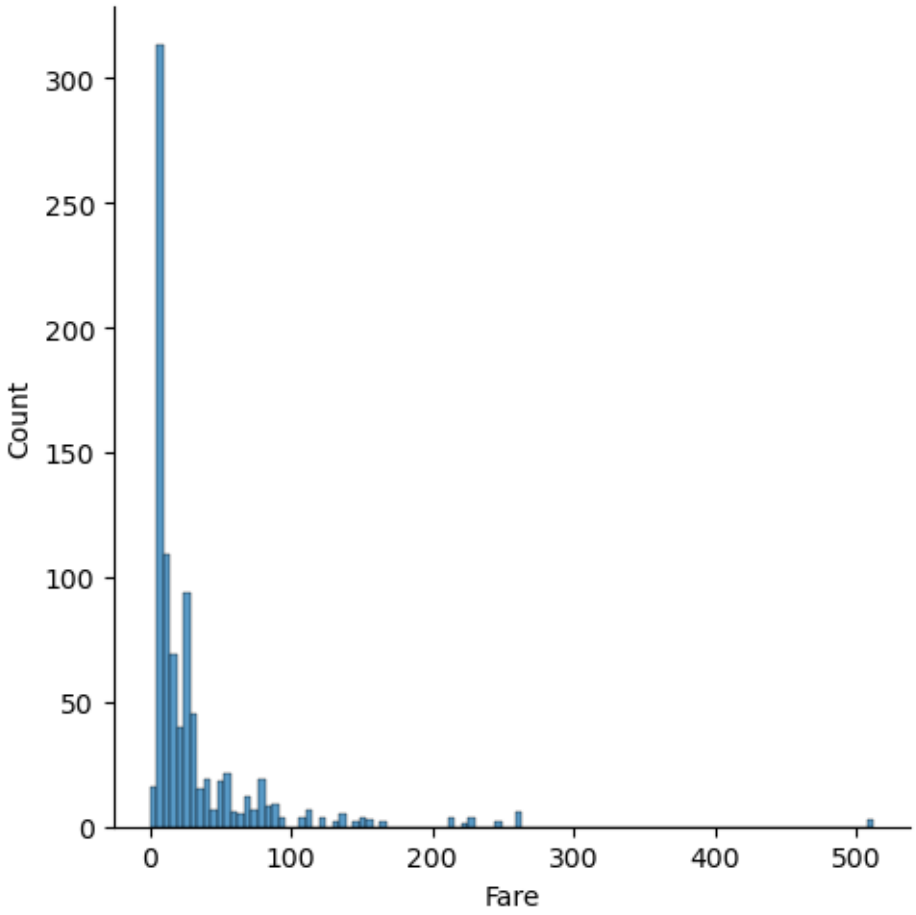
[50]:
```
sns.displot(data, x='Fare', kind='kde')
```

[50]:
```
<seaborn.axisgrid.FacetGrid at 0x79ac64a67190>
```



[53]:
```
sns.displot(data, x='Fare')
```

[53]: <seaborn.axisgrid.FacetGrid at 0x79ac64714d30>

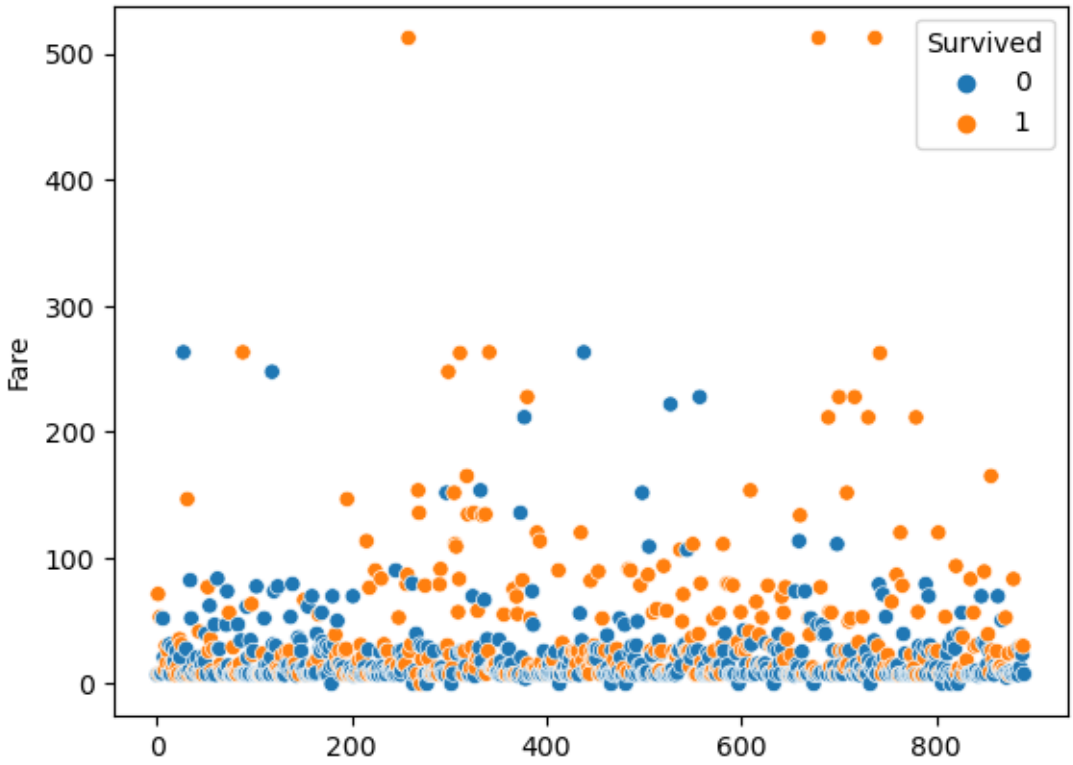


[ ]:

[54]:
```
# scatter chart
```

[56]:
```
sns.scatterplot(data, x=data.index, y='Fare', hue='Survived')
```

[56]: <AxesSubplot:ylabel='Fare'>



[57]:
```
# Let us find what is the average price of ticket for the people who survived
# and what is the averge price for the ticket for people who dies
```

[60]:
```
survived = data[data['Survived']==1]
died = data[data['Survived']==0]
```

[62]:
```
survived.head()
```

[62]:

	PassengerId	Survived	Pclass	\
1	2	1	1	
2	3	1	3	
3	4	1	1	
8	9	1	3	
9	10	1	2	

	Name	Sex	Age	SibSp	\
1	Cumings, Mrs. John Bradley (Florence Briggs Th…	female	38.0	1	
2	Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
8	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	
9	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	

```
   Parch            Ticket     Fare Embarked
1      0          PC 17599  71.2833        C
2      0  STON/O2. 3101282   7.9250        S
3      0            113803  53.1000        S
8      2            347742  11.1333        S
9      0            237736  30.0708        C
```

[63]:
```
died.head()
```

[63]:
```
   PassengerId  Survived  Pclass                                Name   Sex  \
0            1         0       3             Braund, Mr. Owen Harris  male
4            5         0       3            Allen, Mr. William Henry  male
5            6         0       3                    Moran, Mr. James  male
6            7         0       1             McCarthy, Mr. Timothy J  male
7            8         0       3      Palsson, Master. Gosta Leonard  male

         Age  SibSp  Parch     Ticket     Fare Embarked
0  22.000000      1      0  A/5 21171   7.2500        S
4  35.000000      0      0     373450   8.0500        S
5  29.699118      0      0     330877   8.4583        Q
6  54.000000      0      0      17463  51.8625        S
7   2.000000      3      1     349909  21.0750        S
```

[ ]:
```

```

[64]:
```
import numpy as np

mean_survived = np.mean(survived['Fare'])

mean_died = np.mean(died['Fare'])
```

[65]:
```
mean_survived
```

[65]:
```
48.20949823529412
```

[66]:
```
mean_died
```

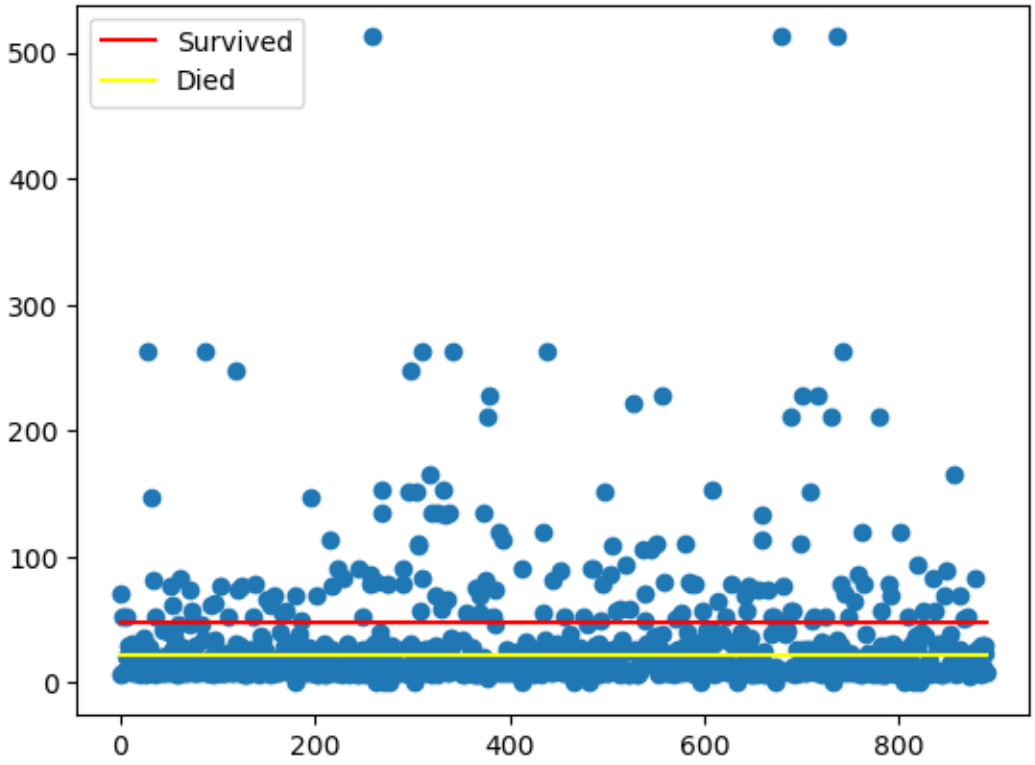
[66]:
```
22.117886885245902
```

[67]:
```
# Try to plot both means on the graph with the actual scattr plot for the fare
```

[71]:
```
import matplotlib.pyplot as plt
plt.scatter(data.index, data['Fare'])
# to plot survived mean
plt.plot(data.index, [mean_survived for i in range(len(data))], color='red',␣
 ↪label='Survived')
```

```
plt.plot(data.index, [mean_died for i in range(len(data))], color='yellow',␣
 ↪label='Died')


plt.legend()
plt.show()
```



[ ]:

[ ]: