

## Q1. Download the Titanic dataset and perform the Exploratory data analysis using pandas.

1. Read the dataset (df= pd.read\_csv(r'.....\Titanic.csv')
2. Display the first and last 10 instances from the dataset
3. Acquire the necessary information using the df.info() and df. Describe()
4. Retrieve the number of columns and rows. (using shape)

---

```
In [ ]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
```

```
In [ ]: import pandas as pd
df= pd.read_csv('Titanic - Titanic.csv')
```

```
In [ ]: df.head(10)
```

Out [ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 2117
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17596
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736

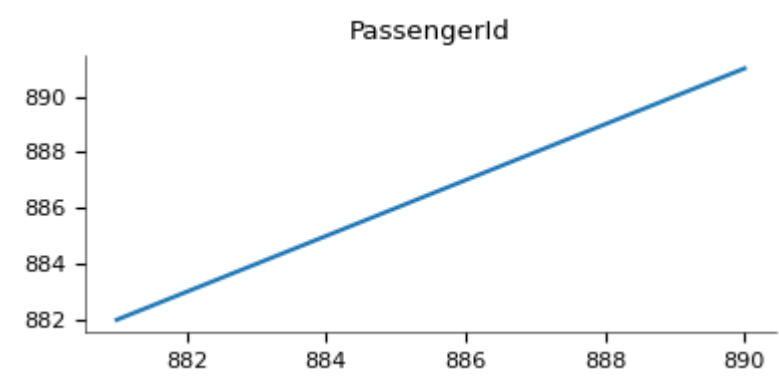
In [ ]:

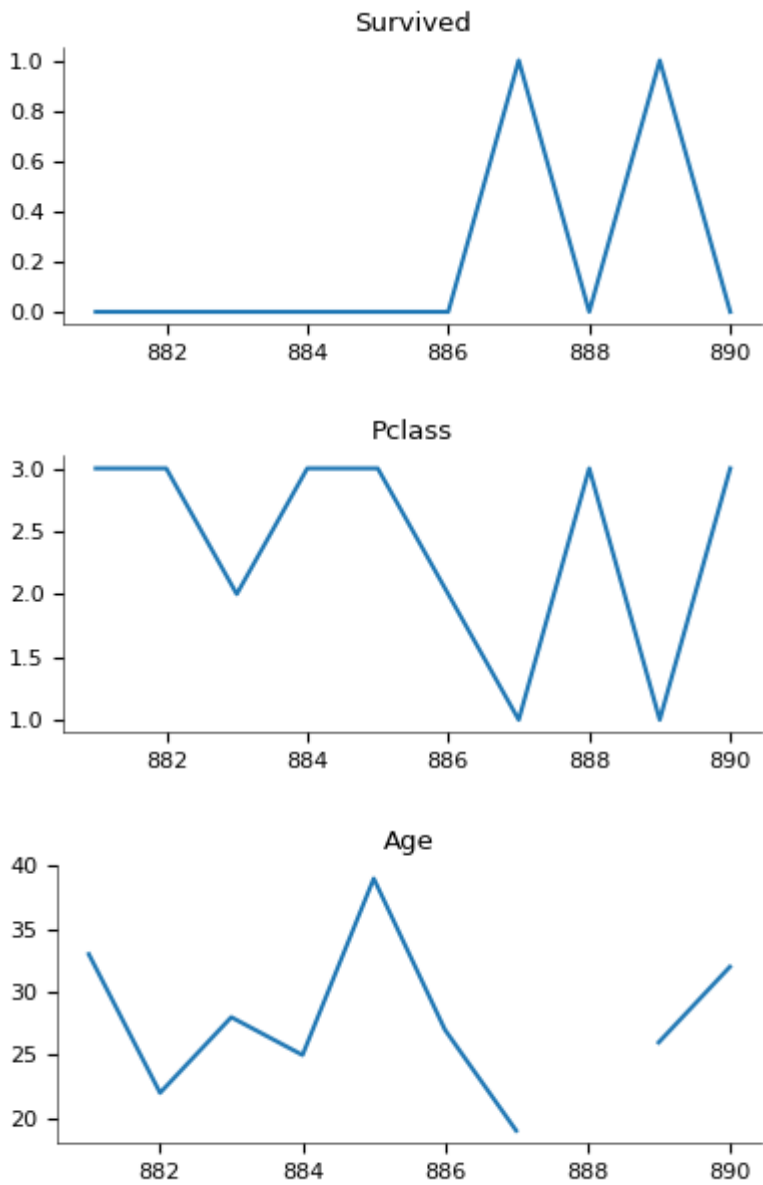
df.tail(10)

Out[ ]:

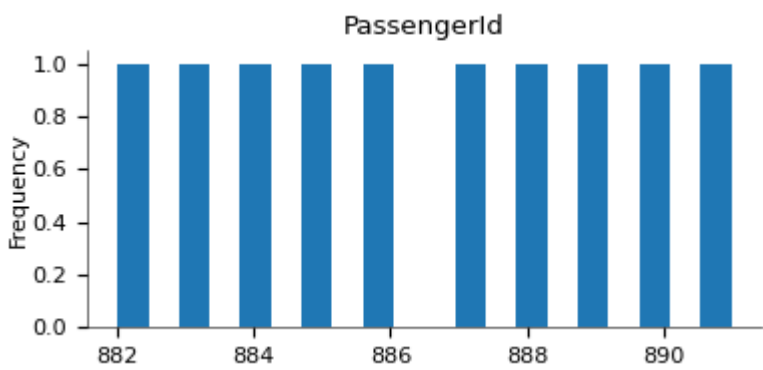
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	T
881	882	0	3	Markun, Mr. Johann	male	33.0	0	0	34
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22.0	0	0	
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0	0	C.A./SC 3.
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTOI 39
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	38
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	21
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	11
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C.
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	11
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	37

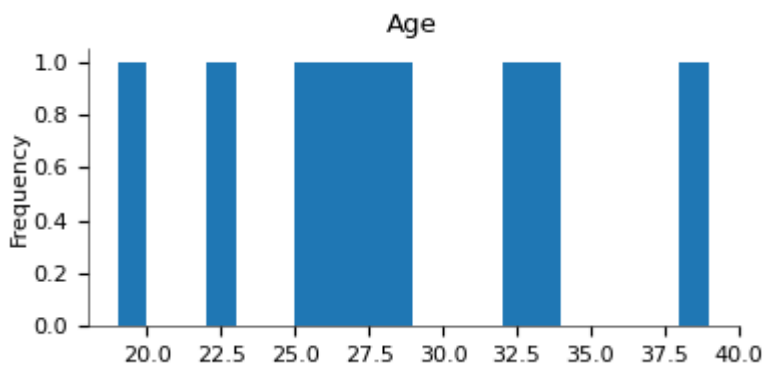
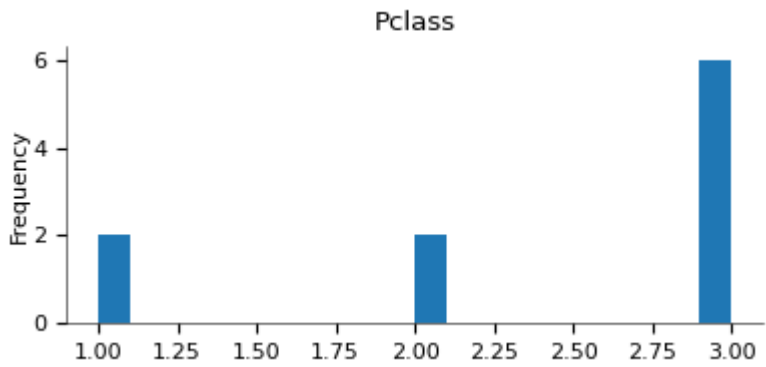
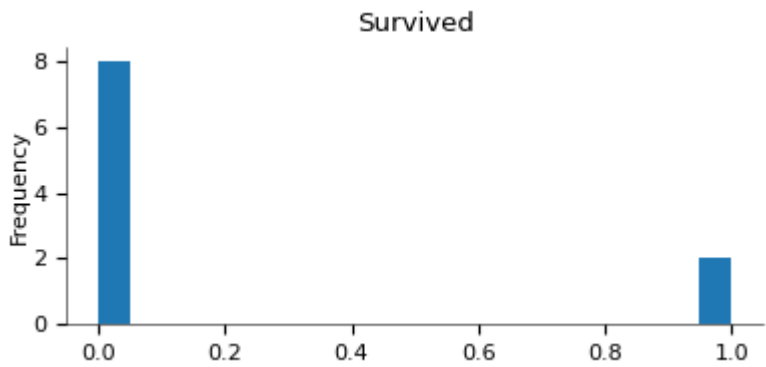
Values



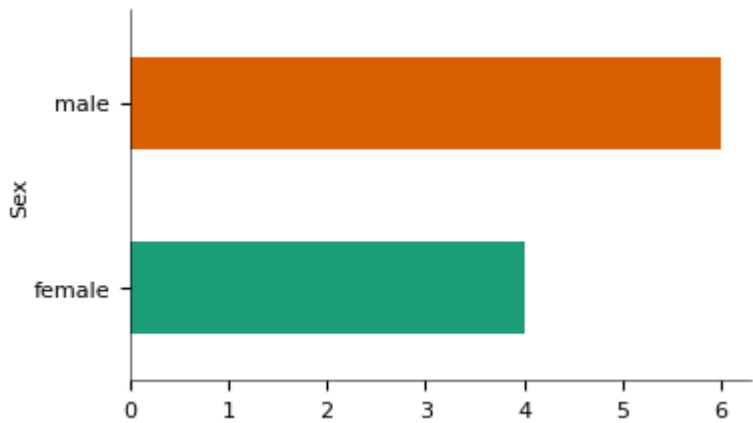


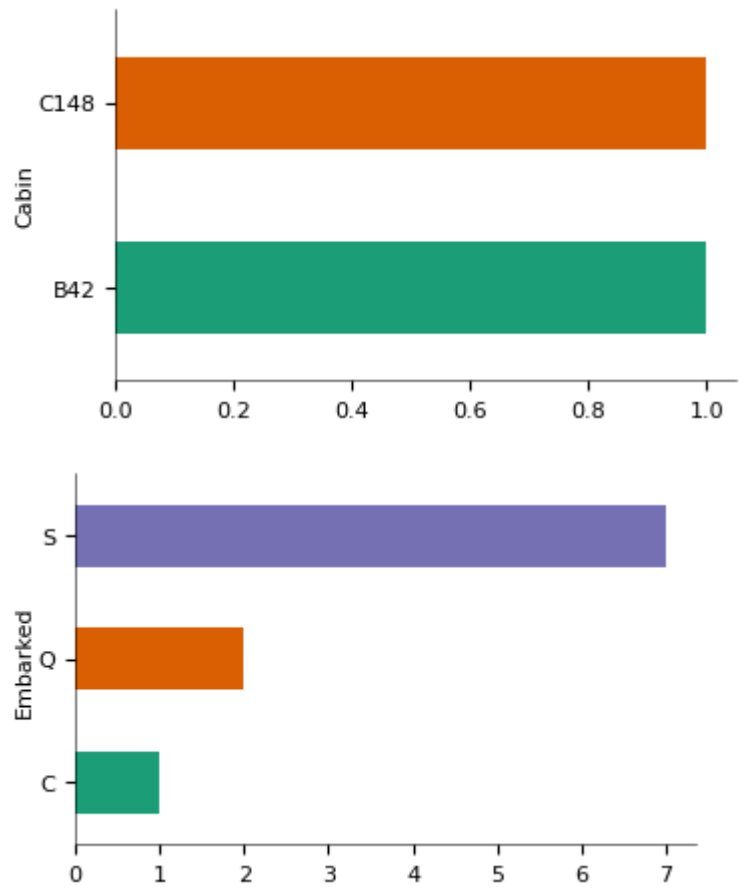
Distributions



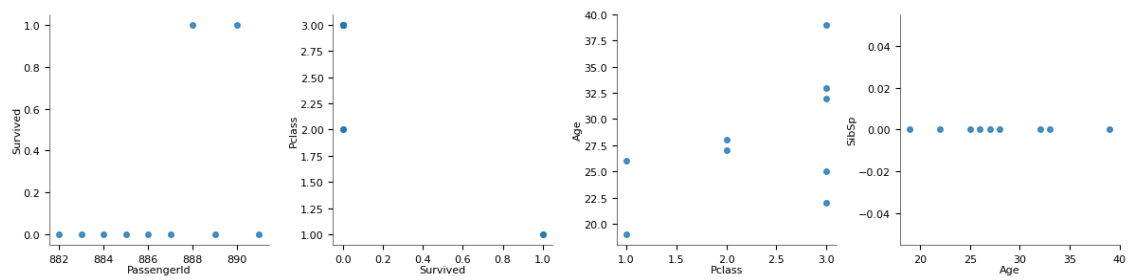


Categorical distributions

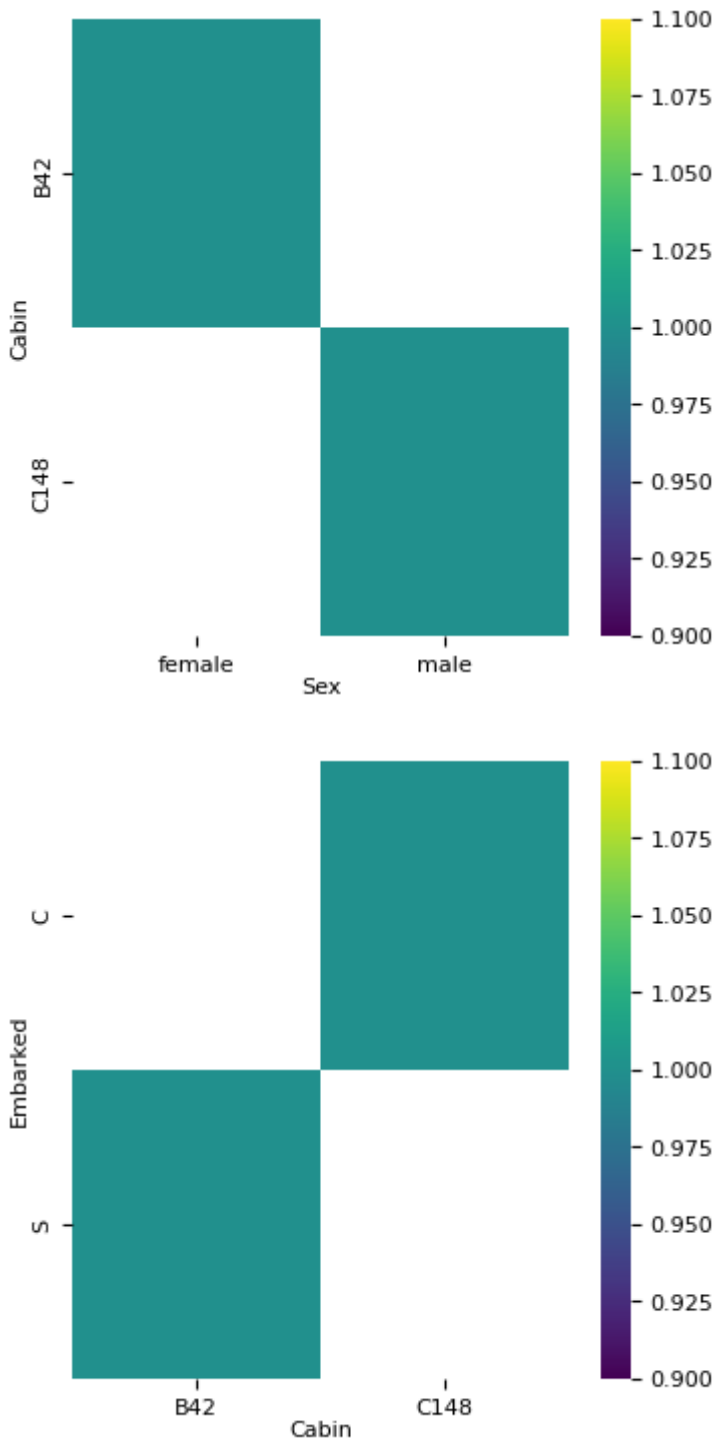




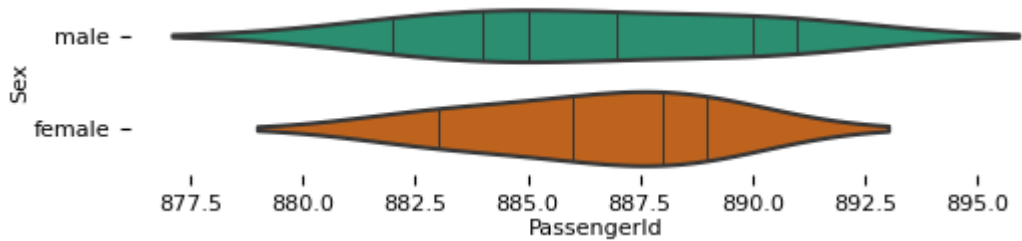
2-d distributions

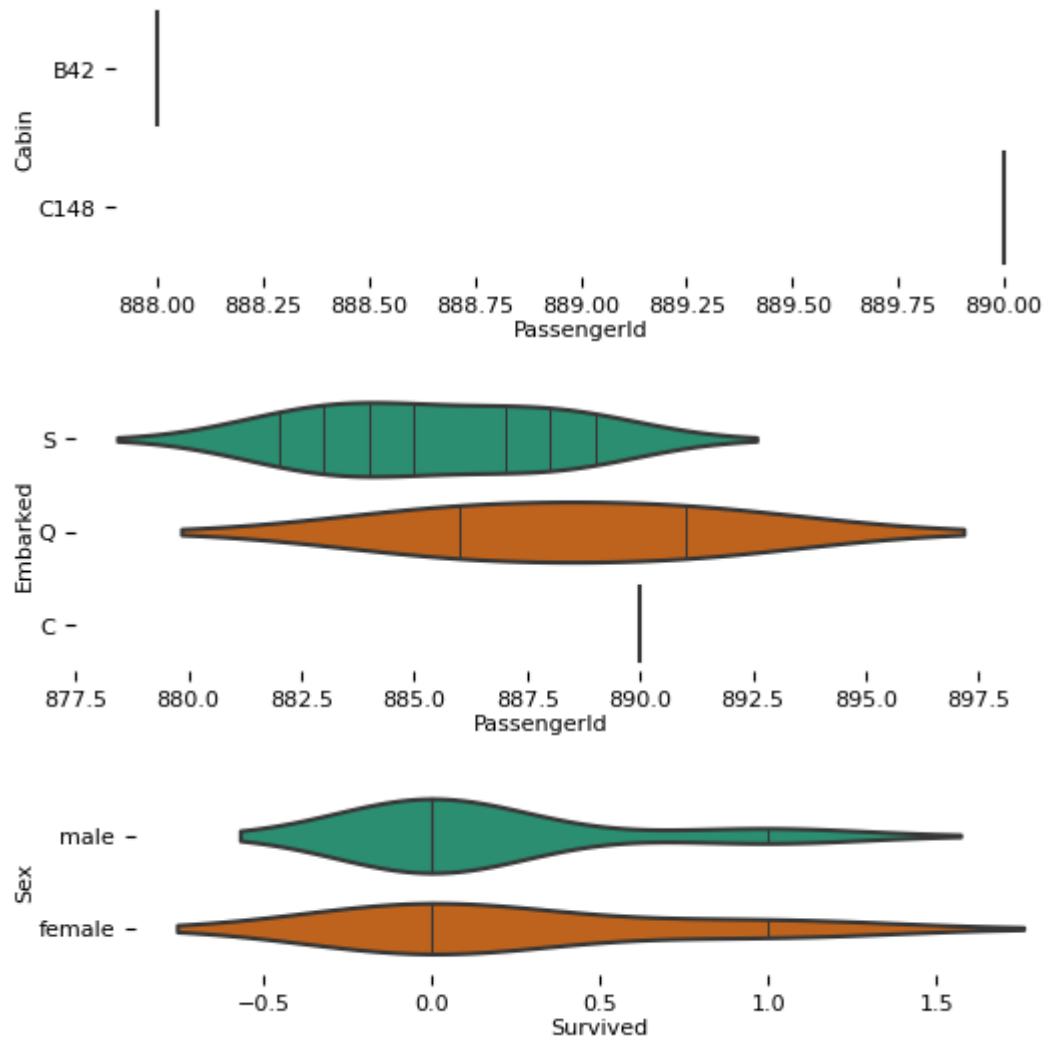


2-d categorical distributions

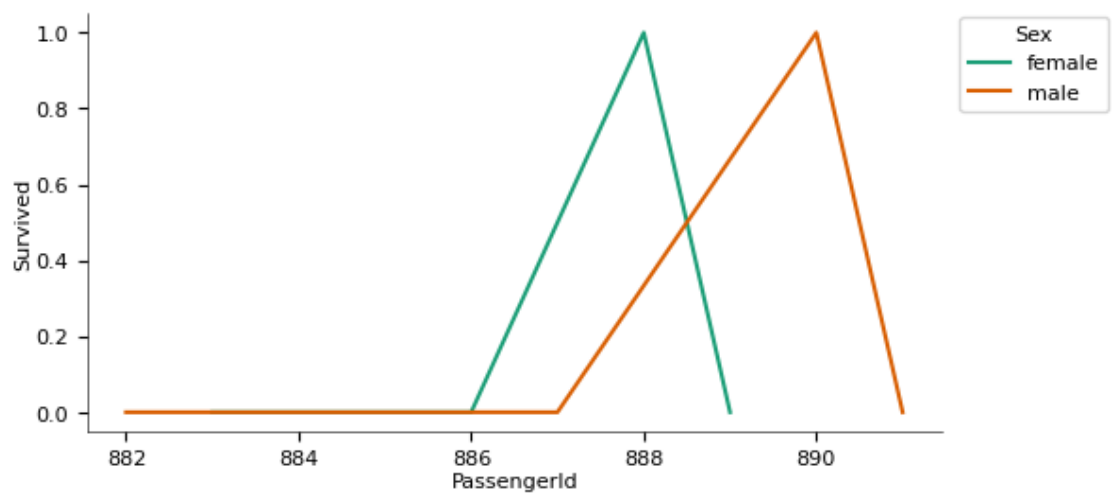


Faceted distributions

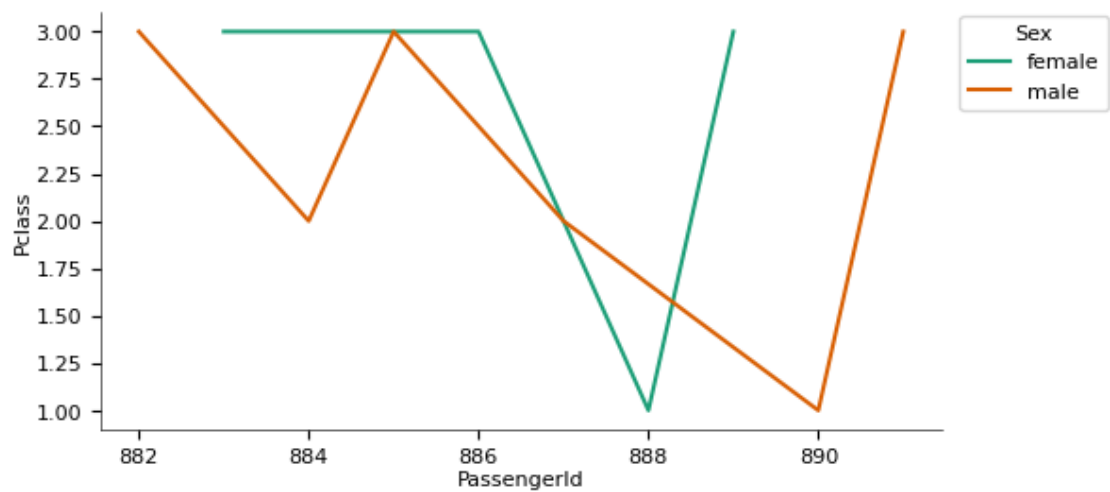
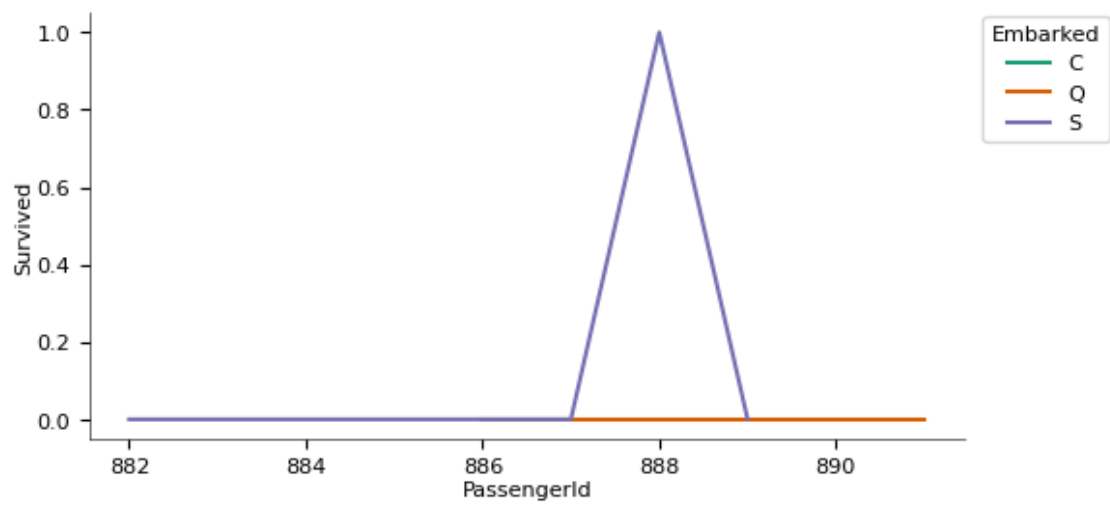
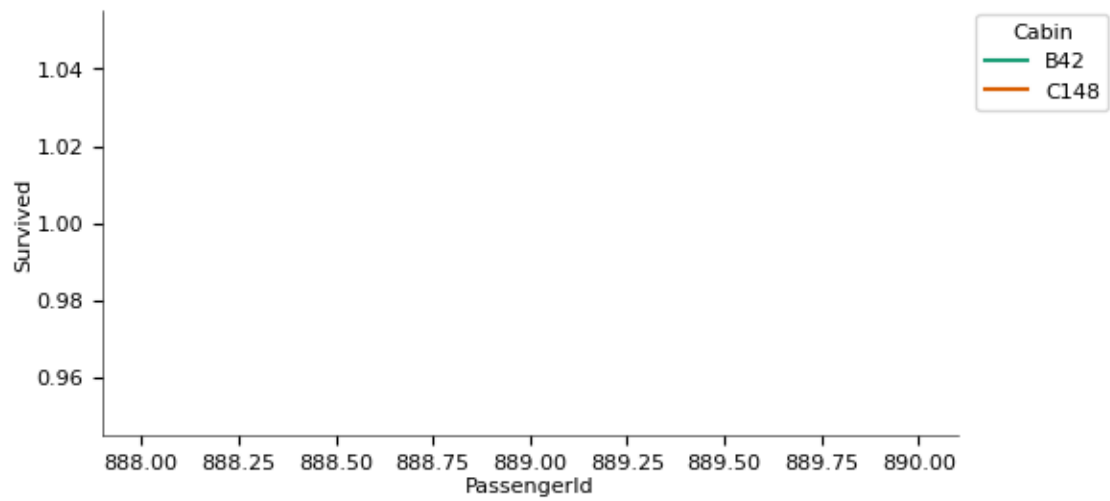




Time series







```
In [ ]: df.info(10)
```

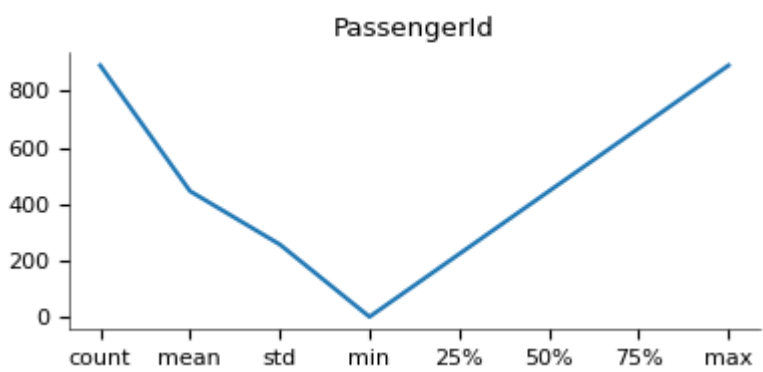
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  -
0   PassengerId  891 non-null    int64
1   Survived     891 non-null    int64
2   Pclass       891 non-null    int64
3   Name         891 non-null    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
```

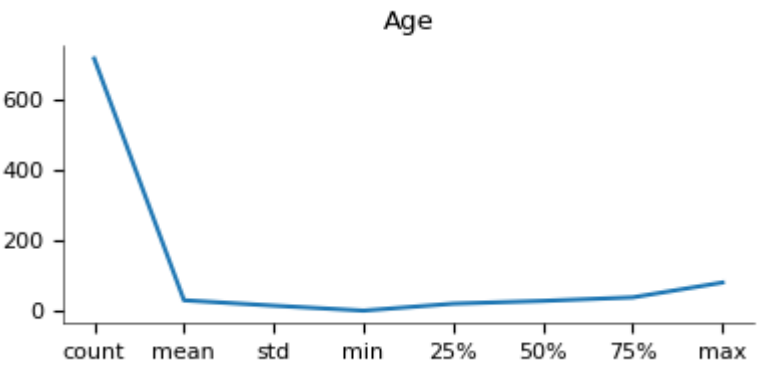
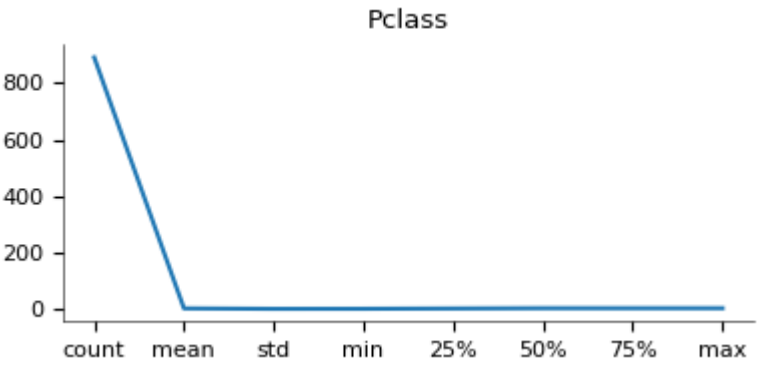
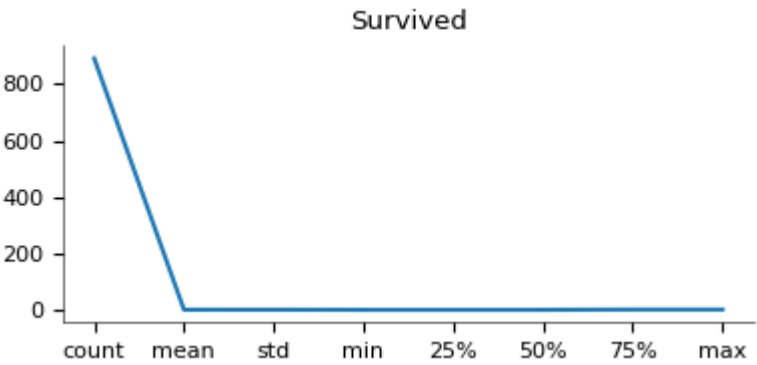
```
In [ ]: df.describe()
```

Out [ ]:

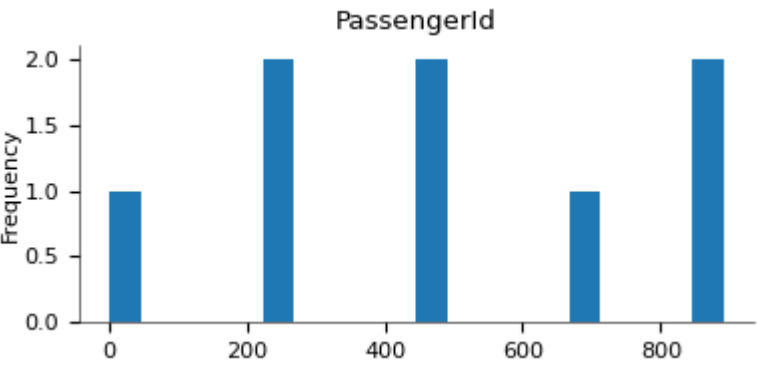
	PassengerId	Survived	Pclass	Age	SibSp	Parch
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000
25%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

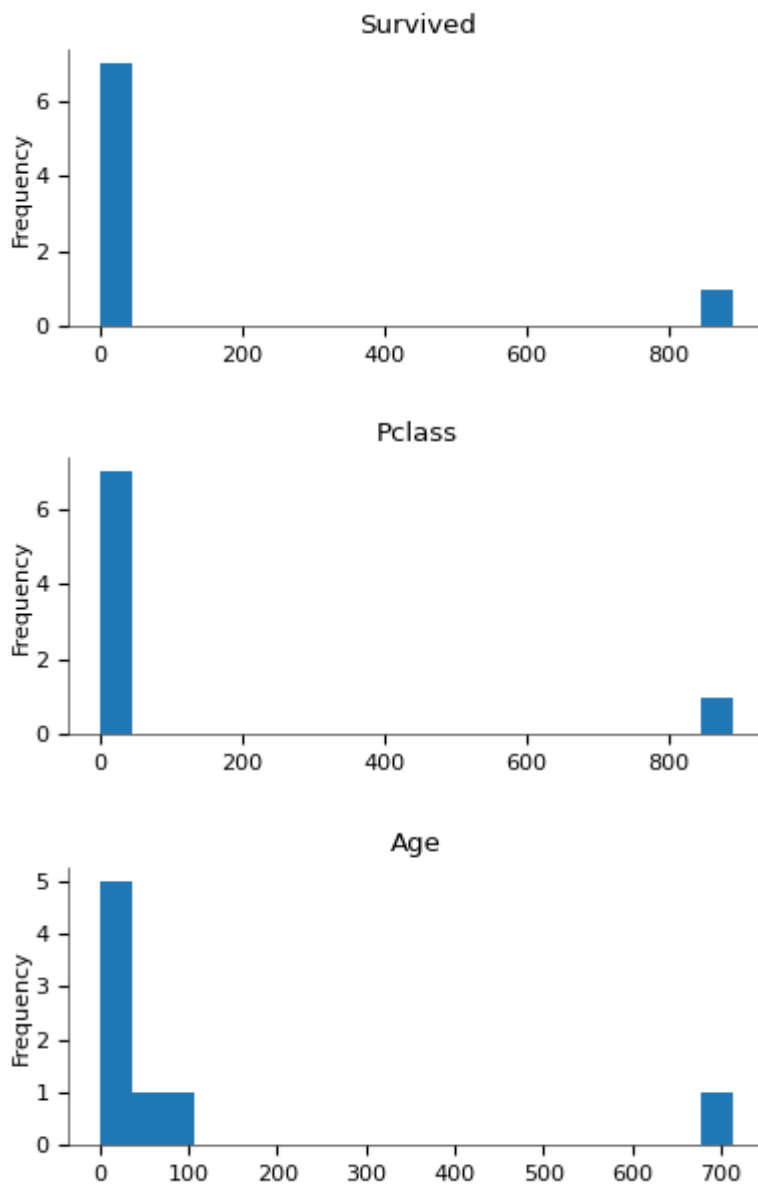
Values



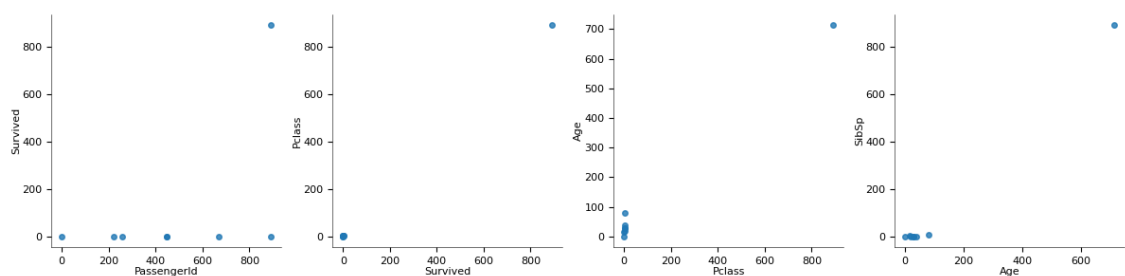


Distributions





## 2-d distributions



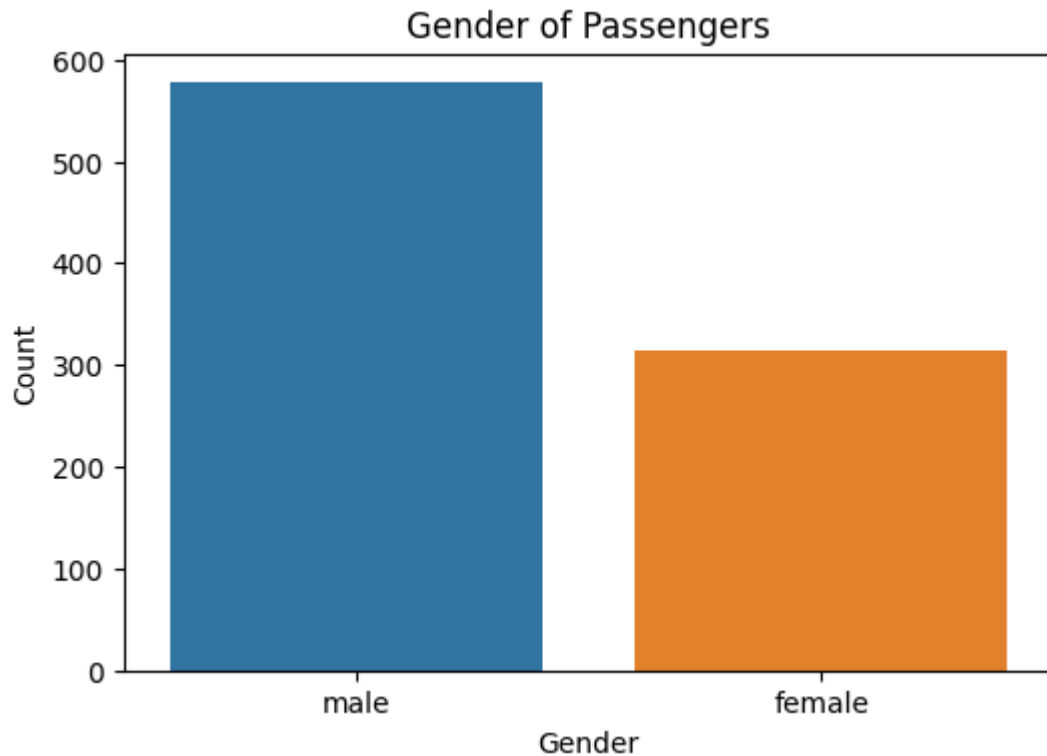
```
In [ ]: df.shape
```

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

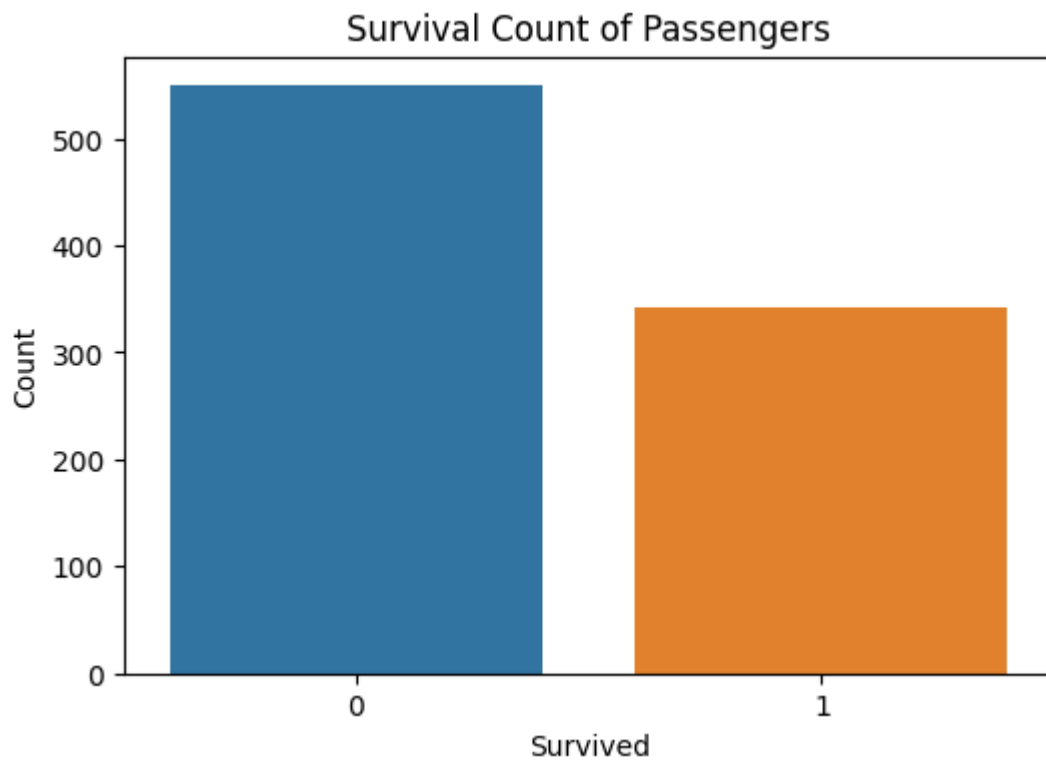
## Q2. Create the data visualization using the matplotlib.

1. Visualize the Gender of Passengers using the Bar graph.
2. Visualize the Survival Count of Passengers using the Bar graph.
3. Visualize the Age of Passengers using the Bar/Histogram graph.
4. Visualize the comparison of Age and Fare of Passengers using the Scatterplot.

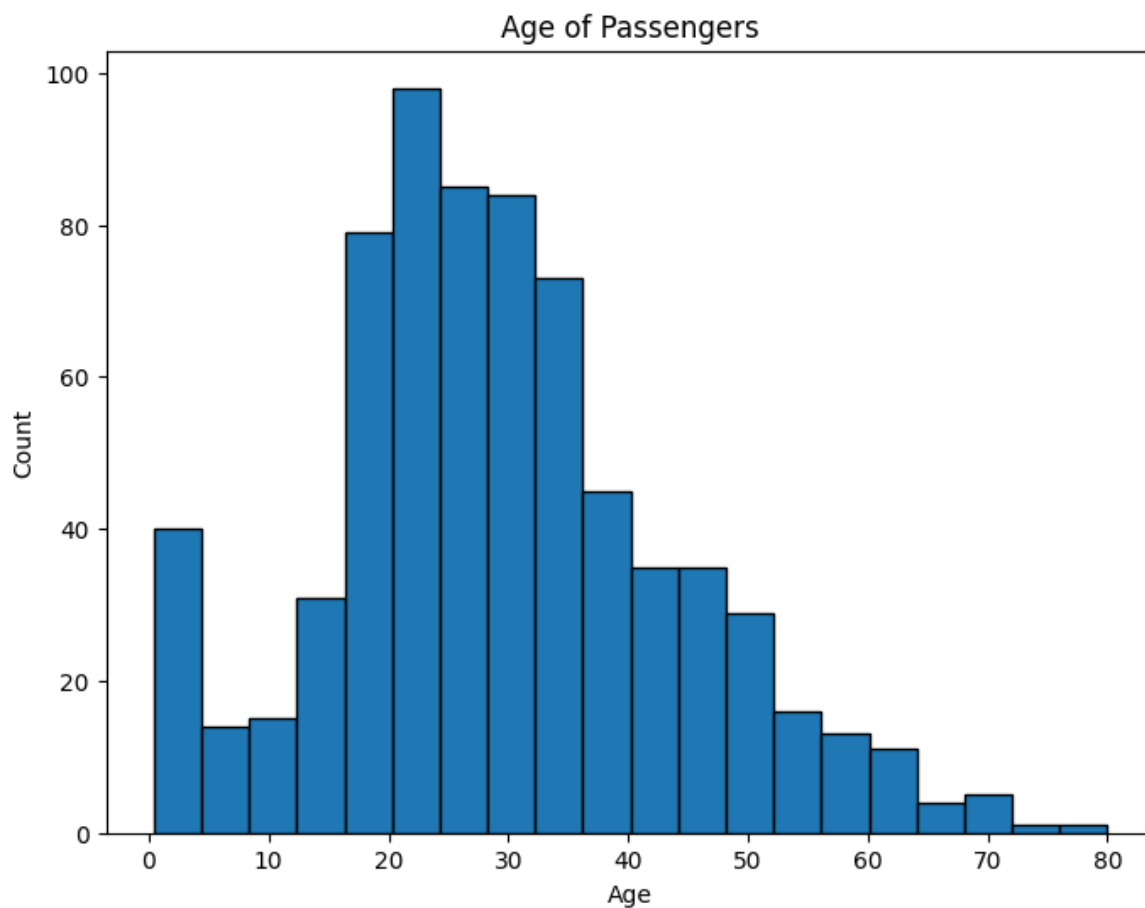
```
In [ ]: gender_counts = df['Sex'].value_counts()
plt.figure(figsize=(6, 4))
sns.barplot(x=gender_counts.index, y=gender_counts.values)
plt.title('Gender of Passengers')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```



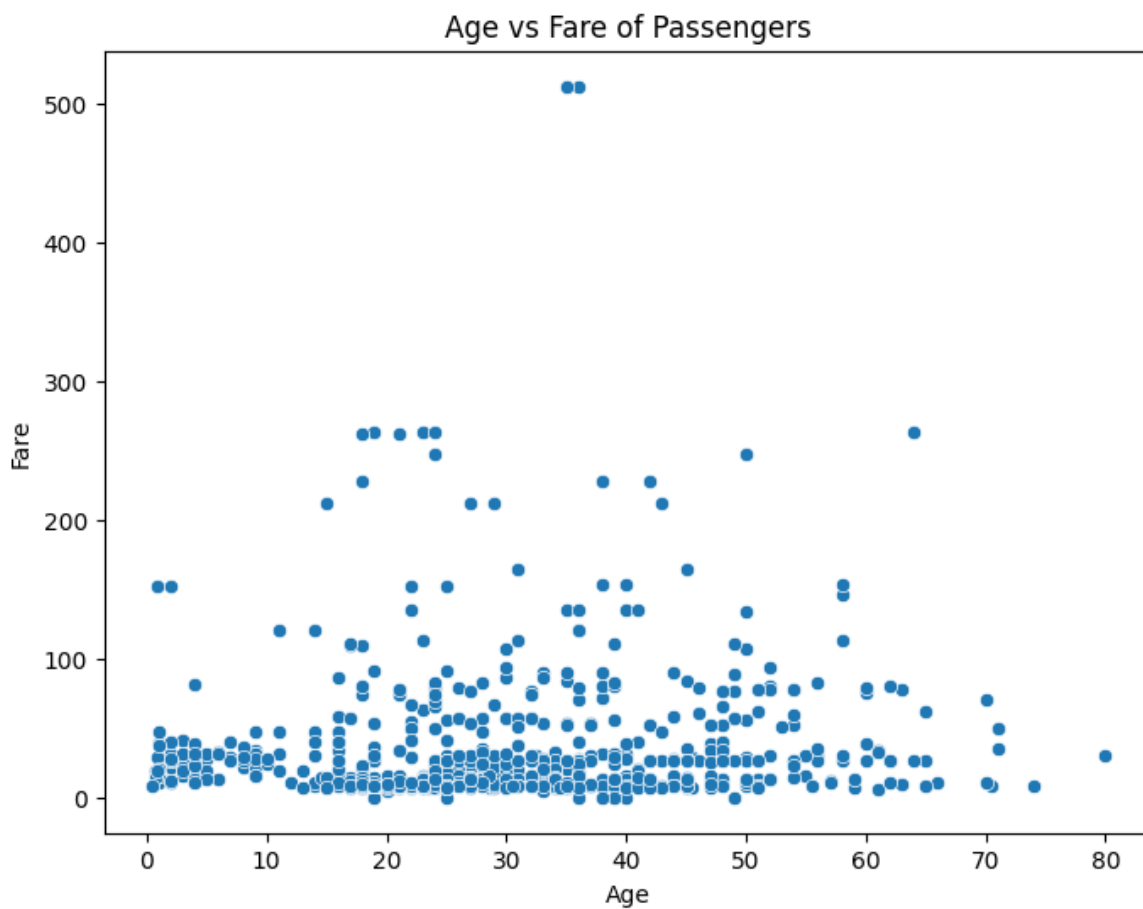
```
In [ ]: survival_counts = df['Survived'].value_counts()
plt.figure(figsize=(6, 4))
sns.barplot(x=survival_counts.index, y=survival_counts.values)
plt.title('Survival Count of Passengers')
plt.xlabel('Survived')
plt.ylabel('Count')
plt.show()
```



```
In [ ]: plt.figure(figsize=(8, 6))
plt.hist(df['Age'], bins=20, edgecolor='black')
plt.title('Age of Passengers')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```



```
In [ ]: plt.figure(figsize=(8, 6))
sns.scatterplot(x='Age', y='Fare', data=df)
plt.title('Age vs Fare of Passengers')
plt.xlabel('Age')
plt.ylabel('Fare')
plt.show()
```



```
In [ ]:
```