

```
[69]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plot
import seaborn as sns
```

```
[28]: df=pd.read_csv("titanic.csv")
```

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

	PassengerId	Survived	Passenger_class	Age	No_of_sibling_spouses	No_of_parents_childrens	Fare
count	889.000000	889.000000	889.000000	889.000000	889.000000	889.000000	889.000000
mean	446.000000	0.382452	2.311586	29.315152	0.524184	0.382452	32.096681
std	256.998173	0.486260	0.834700	12.984932	1.103705	0.806761	49.697504
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	224.000000	0.000000	2.000000	22.000000	0.000000	0.000000	7.895800
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.000000	1.000000	3.000000	35.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

[29]: df.head()

	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

[30]: df.tail()

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

```
[5]: df.shape
```

```
[5]: (891, 12)
```

```
[31]: df["Name"].duplicated()
```

```
[31]: 0      False
      1      False
      2      False
      3      False
      4      False
      ...
      886    False
      887    False
      888    False
      889    False
      890    False
      Name: Name, Length: 891, dtype: bool
```

```
[11]: df.isnull().sum()
```

```
[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
```

```
[32]: df.rename(columns={"Pclass": "Passenger_class"}, inplace=True)
df.rename(columns={"SibSp": "No_of_sibling_spouses"}, inplace=True)
df.rename(columns={"Parch": "No_of_parents_childrens"}, inplace=True)
```

```
[33]: df.head()
```

[33]:	PassengerId	Survived	Passenger_class	Name	Sex	Age	No_of_sibling_spouses	No_of_parents_childrens	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

```
[64]: df = df.dropna(subset=['Embarked'])
```

```
[65]: df.shape
```

```
[65]: (889, 12)
```

```
[66]: port_map = {'C': 'Cherbourg (France)', 'Q': 'Queenstown (Ireland)', 'S': 'Southampton (England)'}
df['Embarked'] = df['Embarked'].map(port_map)
```

```
[67]: df.tail()
```

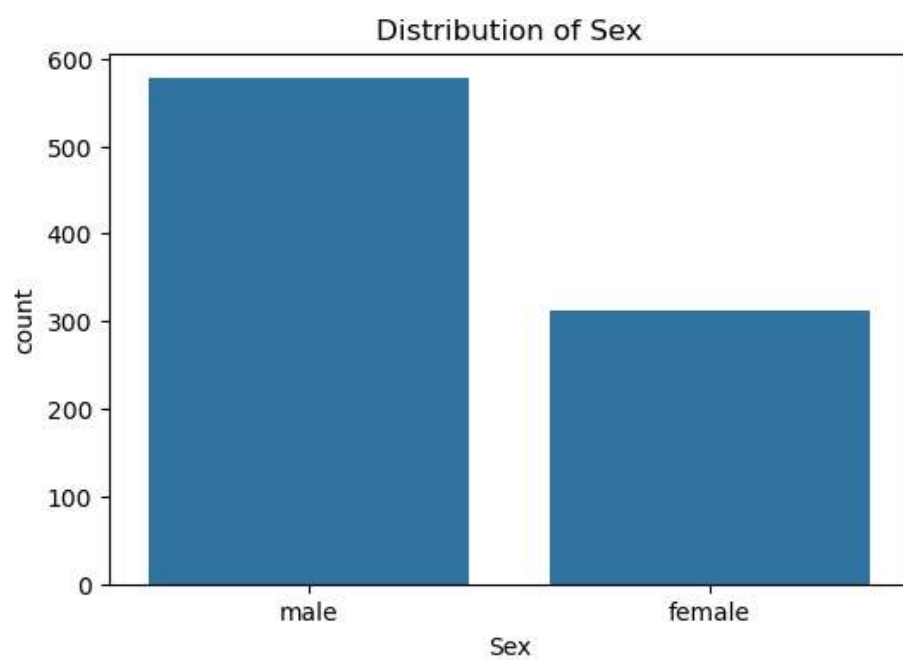
[67]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	No_of_sibling_spouses	No_of_parents_childrens	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	Southampton (England)
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	Southampton (England)
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	28.0	1	2	W./C. 6607	23.45	NaN	Southampton (England)
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	Cherbourg (France)
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Queenstown (Ireland)

```
[70]: # For each categorical column in the list, create a count plot
# to visualize how many passengers belong to each category, e.g. how many males vs females.

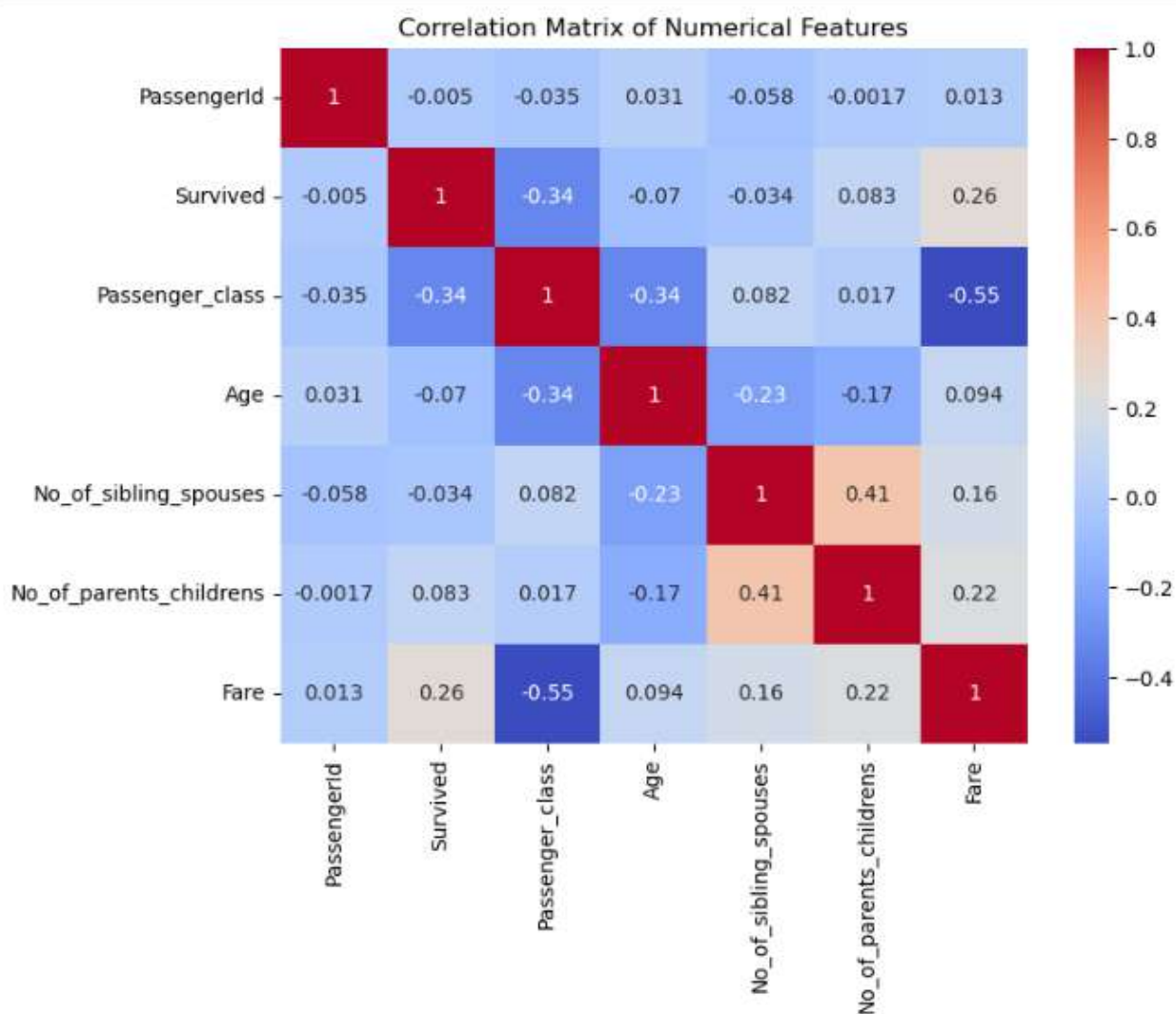
cat_cols = ['Sex', 'Pclass', 'Embarked']

for col in cat_cols:
    plot.figure(figsize=(6,4))
    sns.countplot(data=df, x=col)
    plot.title(f'Distribution of {col}')
    plot.show()
```



```
[74]: # Plot a heatmap to illustrate the correlation coefficients
# between the numerical variables in the dataset,
# helping to see relationships like age vs fare or class vs survival.
numeric_df = df.select_dtypes(include=['number'])

plot.figure(figsize=(8,6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plot.title('Correlation Matrix of Numerical Features')
plot.show()
```



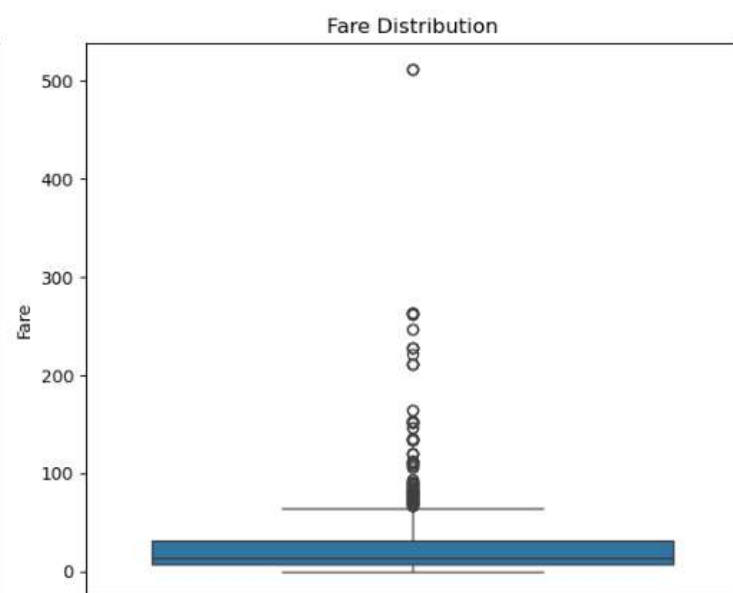
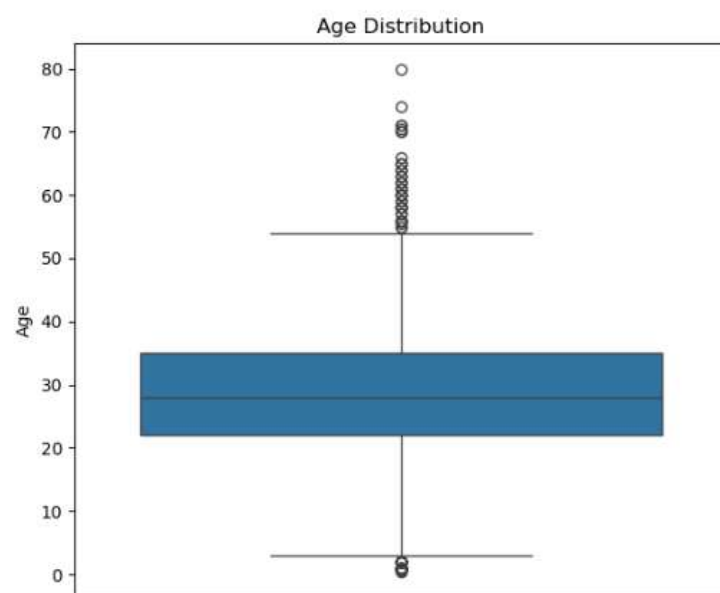

```

plot.figure(figsize=(12,5))
plot.subplot(1,2,1)
sns.boxplot(df['Age'])
plot.title('Age Distribution')

plot.subplot(1,2,2)
sns.boxplot(df['Fare'])
plot.title('Fare Distribution')

plot.tight_layout()
plot.show()

```



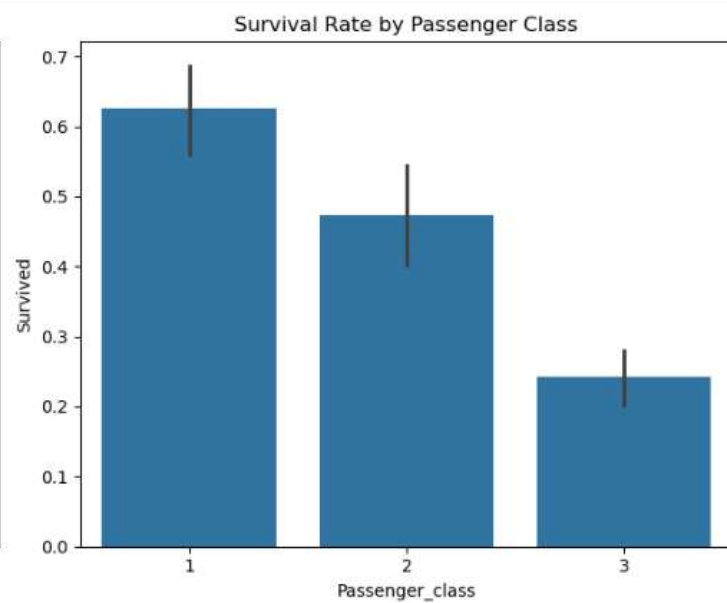
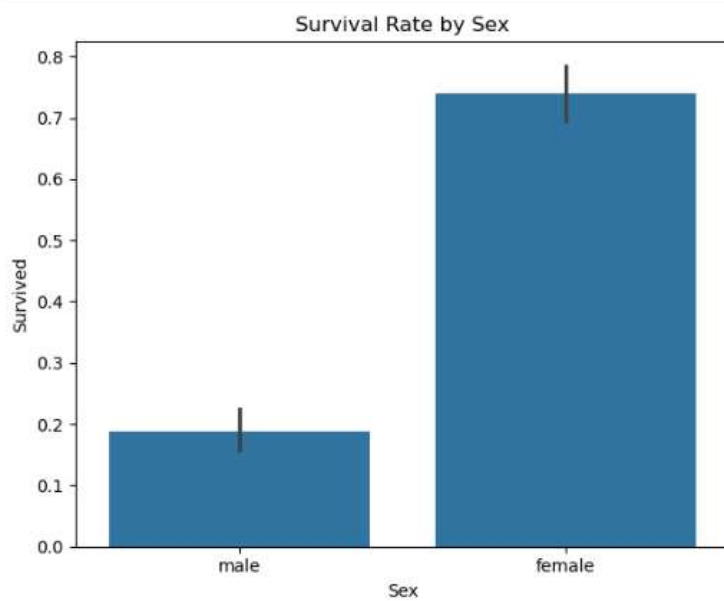
```
[79]: # Create bar plots to visualize the average survival rate
# grouped by passenger sex and class, showing how these factors impact survival chances.

plot.figure(figsize=(12,5))

plot.subplot(1,2,1)
sns.barplot(x='Sex', y='Survived', data=df)
plot.title('Survival Rate by Sex')

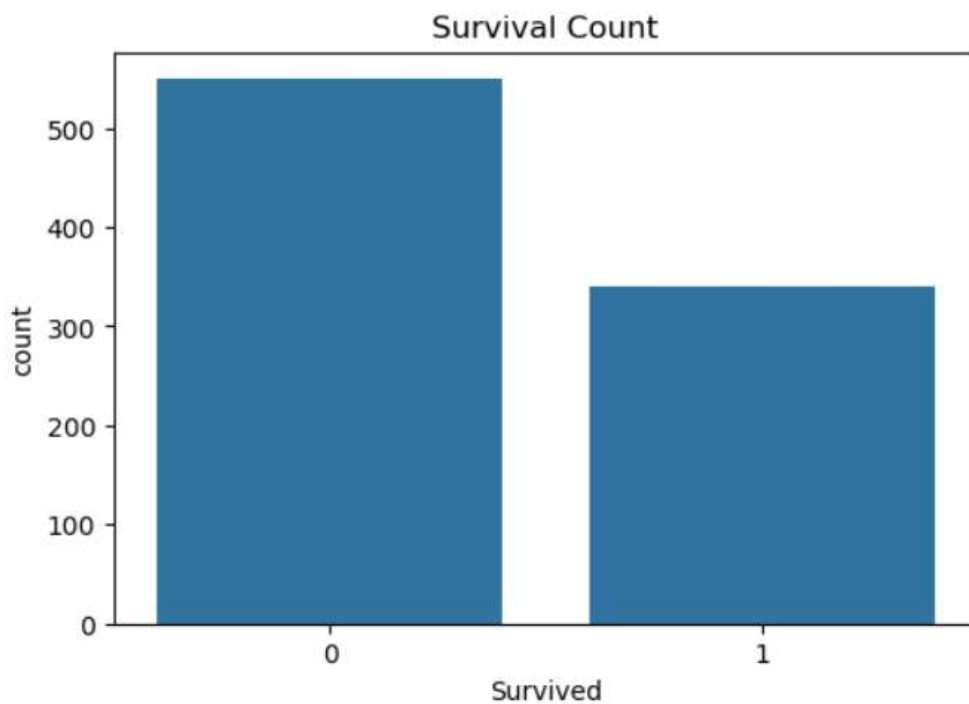
plot.subplot(1,2,2)
sns.barplot(x='Passenger_class', y='Survived', data=df)
plot.title('Survival Rate by Passenger Class')

plot.tight_layout()
plot.show()
```



```
[80]: # Plot a count plot to show the total number of passengers who survived vs those who did not.
```

```
plot.figure(figsize=(6,4))  
sns.countplot(data=df, x='Survived')  
plot.title('Survival Count')  
plot.show()
```



```
[82]: # Scatterplot: Age vs Fare, colored by Survival status
plot.figure(figsize=(8,6))
sns.scatterplot(data=df, x='Age', y='Fare', hue='Survived', palette='coolwarm')
plot.title('Age vs Fare by Survival Status')
plot.show()

# Scatterplot: Age vs Fare, colored by Passenger Class
plot.figure(figsize=(8,6))
sns.scatterplot(data=df, x='Age', y='Fare', hue='Passenger_class', palette='Set2')
plot.title('Age vs Fare by Passenger Class')
plot.show()
```

