```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns


df=pd.read_csv('/content/Titanic-Dataset.csv')


df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8
...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7

891 rows × 12 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.tail()
```

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

```
df.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
```

```
df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fa
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000(
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204;
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693-
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000(
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910-
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454;
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000(
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329;

```
df.shape
```

```
(891, 12)
```

```
df.isnull().sum().sum()
```

```
np.int64(866)
```

```
df=df.fillna(np.mean)
```

```
df.isnull().sum().sum()
```

```
np.int64(0)
```

Select only numeric columns

```
numeric_cols = df.select_dtypes(include=['int64', 'float64']).columns
```

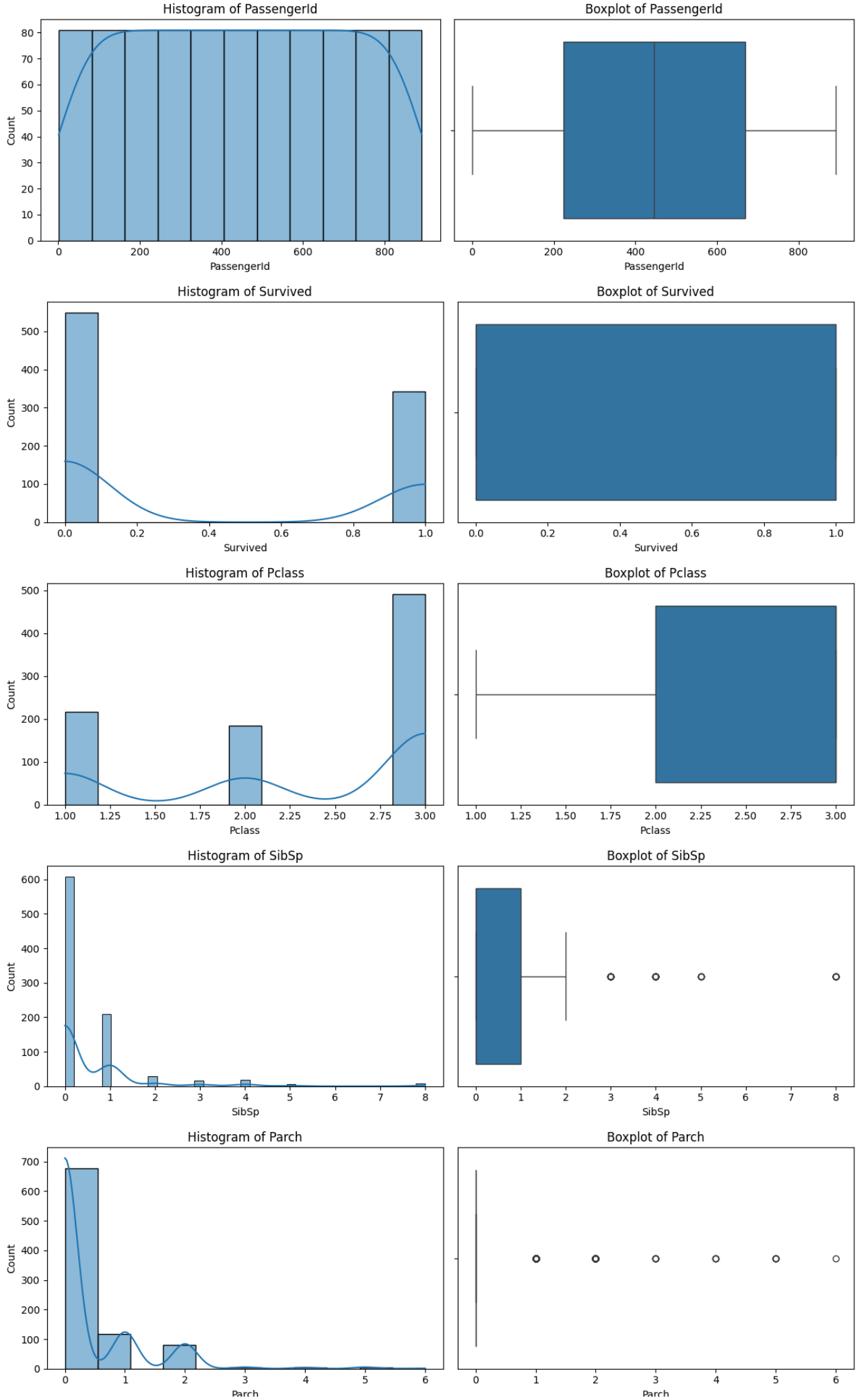
Summary Statistics

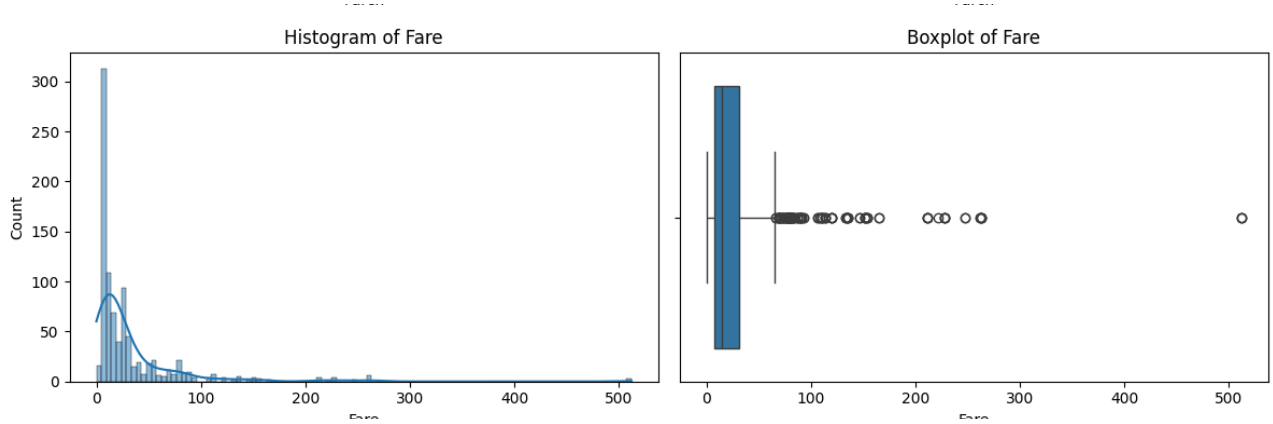
```
summary = df[numeric_cols].agg(['mean', 'median', 'std', 'min', 'max'])
print("Summary Statistics:\n", summary)
```

```
Summary Statistics:
         PassengerId  Survived    Pclass     SibSp     Parch        Fare
mean     446.000000  0.383838  2.308642  0.523008  0.381594   32.204208
median   446.000000  0.000000  3.000000  0.000000  0.000000   14.454200
std      257.353842  0.486592  0.836071  1.102743  0.806057   49.693429
min        1.000000  0.000000  1.000000  0.000000  0.000000    0.000000
max      891.000000  1.000000  3.000000  8.000000  6.000000  512.329200
```

Histograms & Boxplots

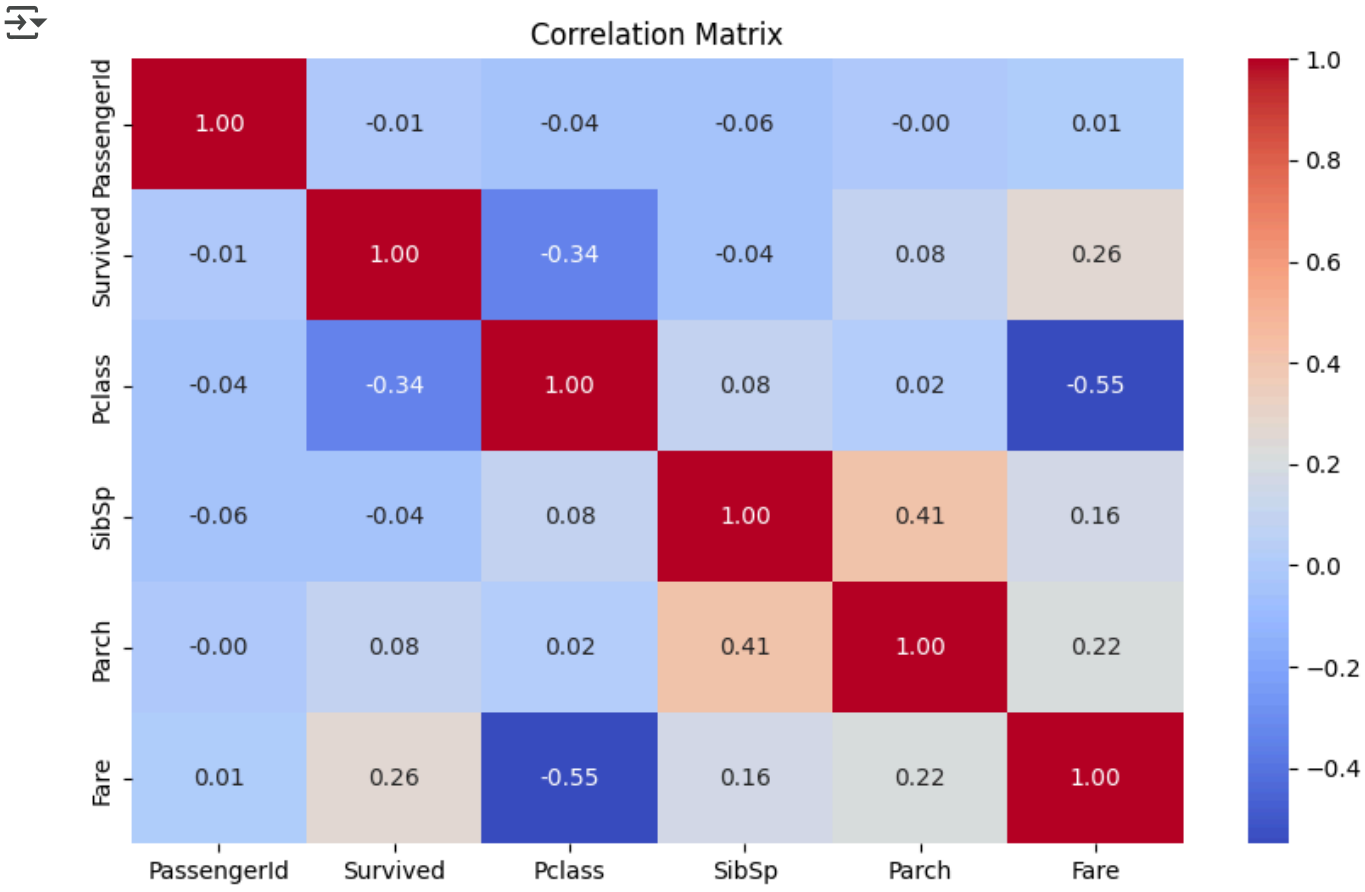
```
for col in numeric_cols:
    plt.figure(figsize=(12, 4))
    plt.subplot(1, 2, 1)
    sns.histplot(df[col].dropna(), kde=True)
    plt.title(f'Histogram of {col}')
    plt.subplot(1, 2, 2)
    sns.boxplot(x=df[col].dropna())
    plt.title(f'Boxplot of {col}')
    plt.tight_layout()
    plt.show()
```


Histogram of PassengerId
Count
PassengerId
Boxplot of PassengerId
PassengerId
Histogram of Survived
Count
Survived
Boxplot of Survived
Survived
Histogram of Pclass
Count
Pclass
Boxplot of Pclass
Pclass
Histogram of SibSp
Count
SibSp
Boxplot of SibSp
SibSp
Histogram of Parch
Count
Parch
Boxplot of Parch
Parch



Correlation Matrix

```
plt.figure(figsize=(10, 6))
sns.heatmap(df[numeric_cols].corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Matrix")
plt.show()
```



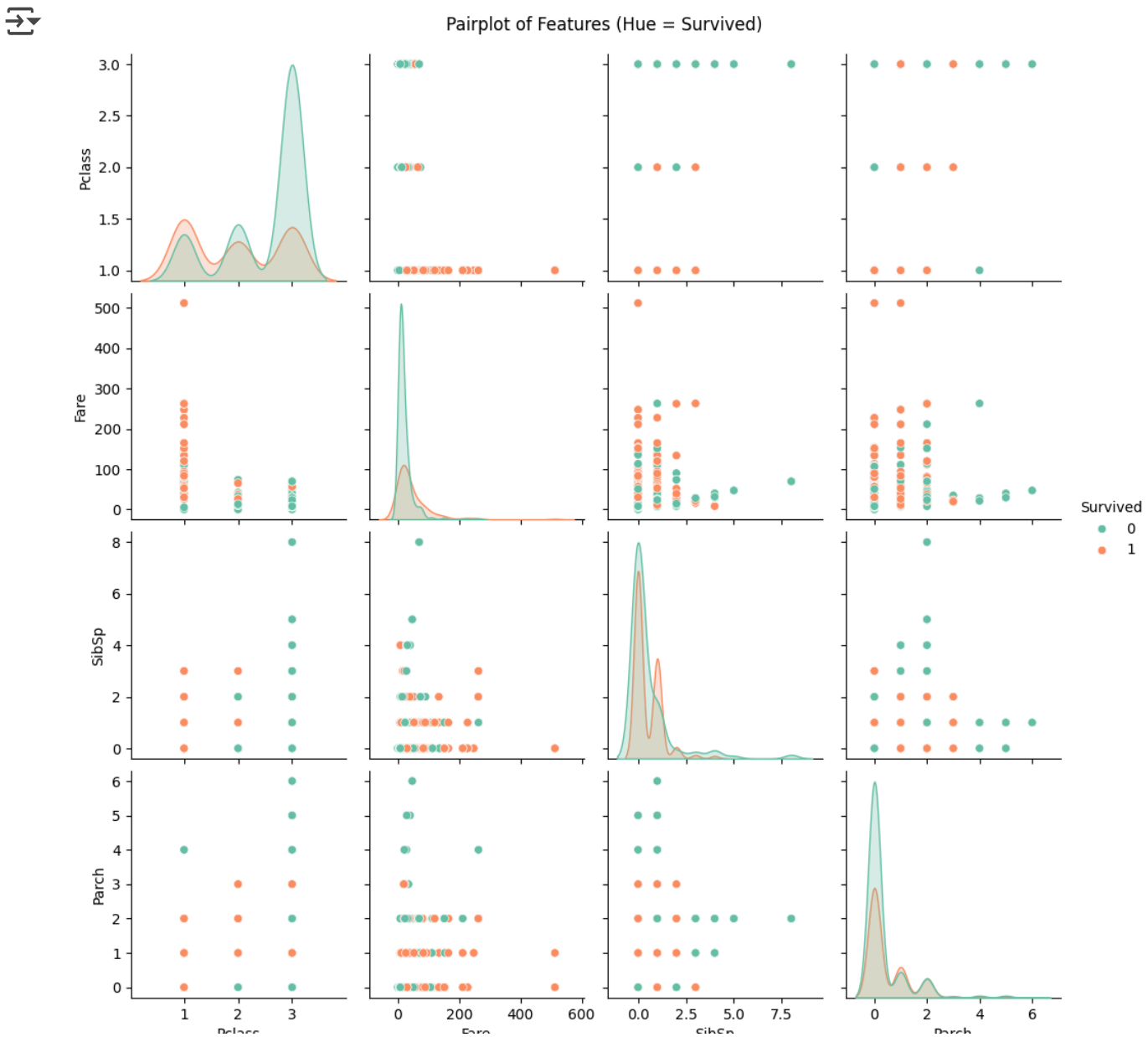
Pairplot

```
selected = ['Survived', 'Pclass', 'Age', 'Fare', 'SibSp', 'Parch']
pairplot_data = df[selected].dropna()
```

Convert all to numeric explicitly

```
pairplot_data['Survived'] = pairplot_data['Survived'].astype(str)
sns.pairplot(pairplot_data, hue='Survived', palette='Set2')
```

```
sns.pairplot(pairplot_data, hue='Survived', palette='Set2')
plt.suptitle("Pairplot of Features (Hue = Survived)", y=1.02)
plt.show()
```



Titanic EDA Program for Patterns, Trends, Anomalies

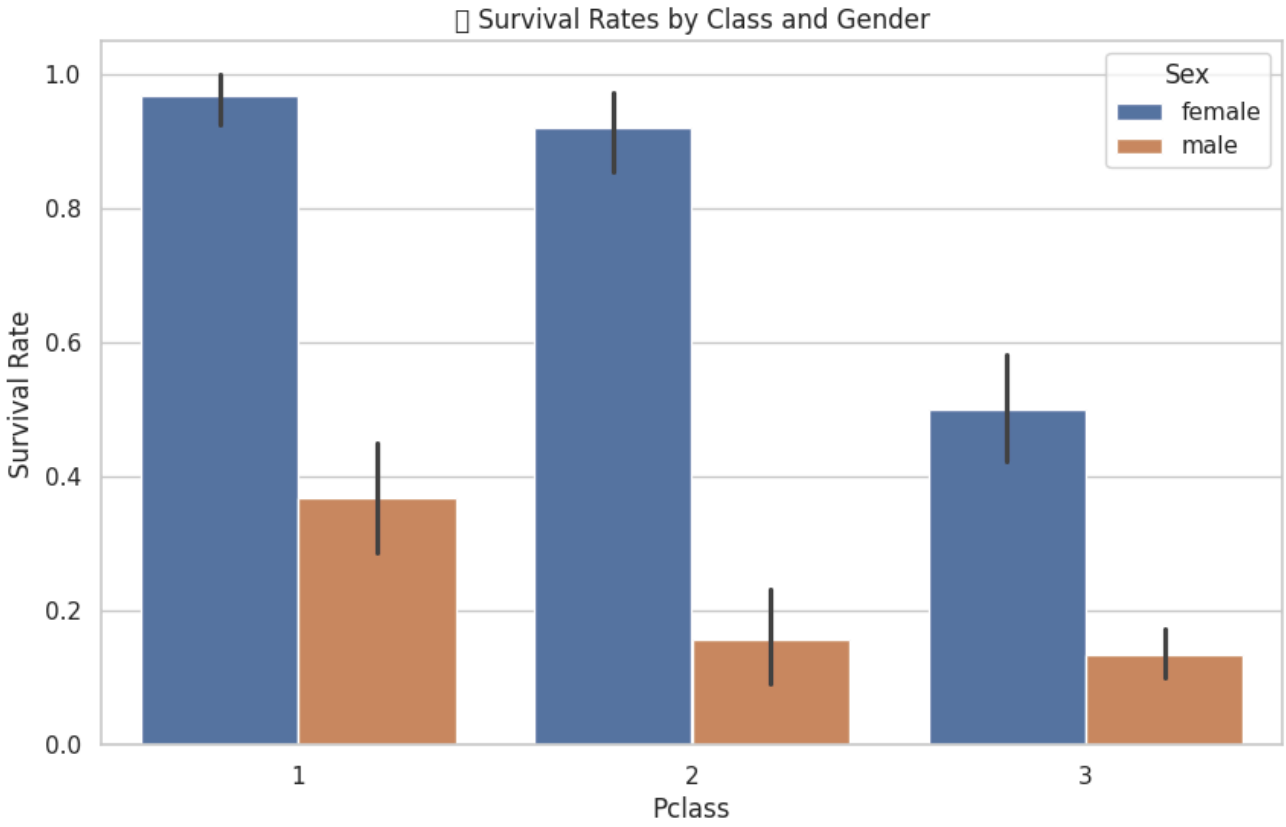
```
df['FamilySize'] = df['SibSp'] + df['Parch'] + 1
df['HasCabin'] = df['Cabin'].notnull().astype(int)
```

```
sns.set(style="whitegrid")
```

Survival Rates by Class and Gender

```
plt.figure(figsize=(10, 6))
sns.barplot(x="Pclass", y="Survived", hue="Sex", data=df)
plt.title("Survival Rates by Class and Gender")
plt.ylabel("Survival Rate")
plt.show()
```

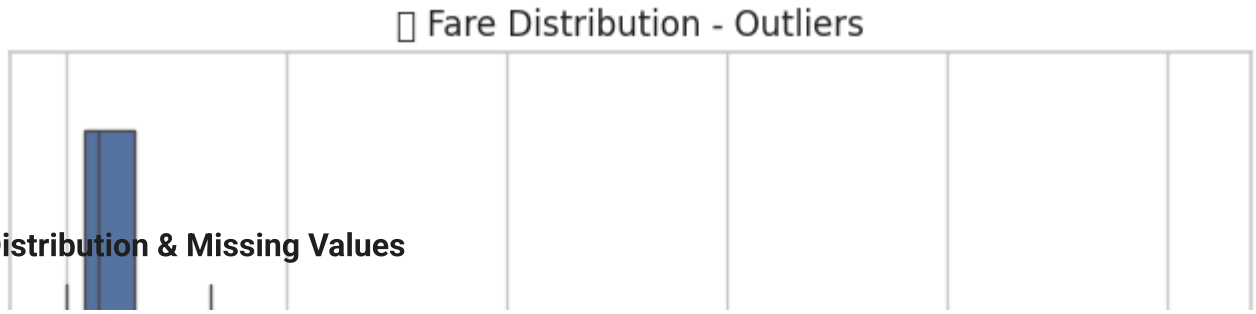
```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```



Fare Distribution (Boxplot for anomalies)

```
plt.figure(figsize=(8, 5))
sns.boxplot(x="Fare", data=df)
plt.title("Fare Distribution - Outliers")
plt.show()
```

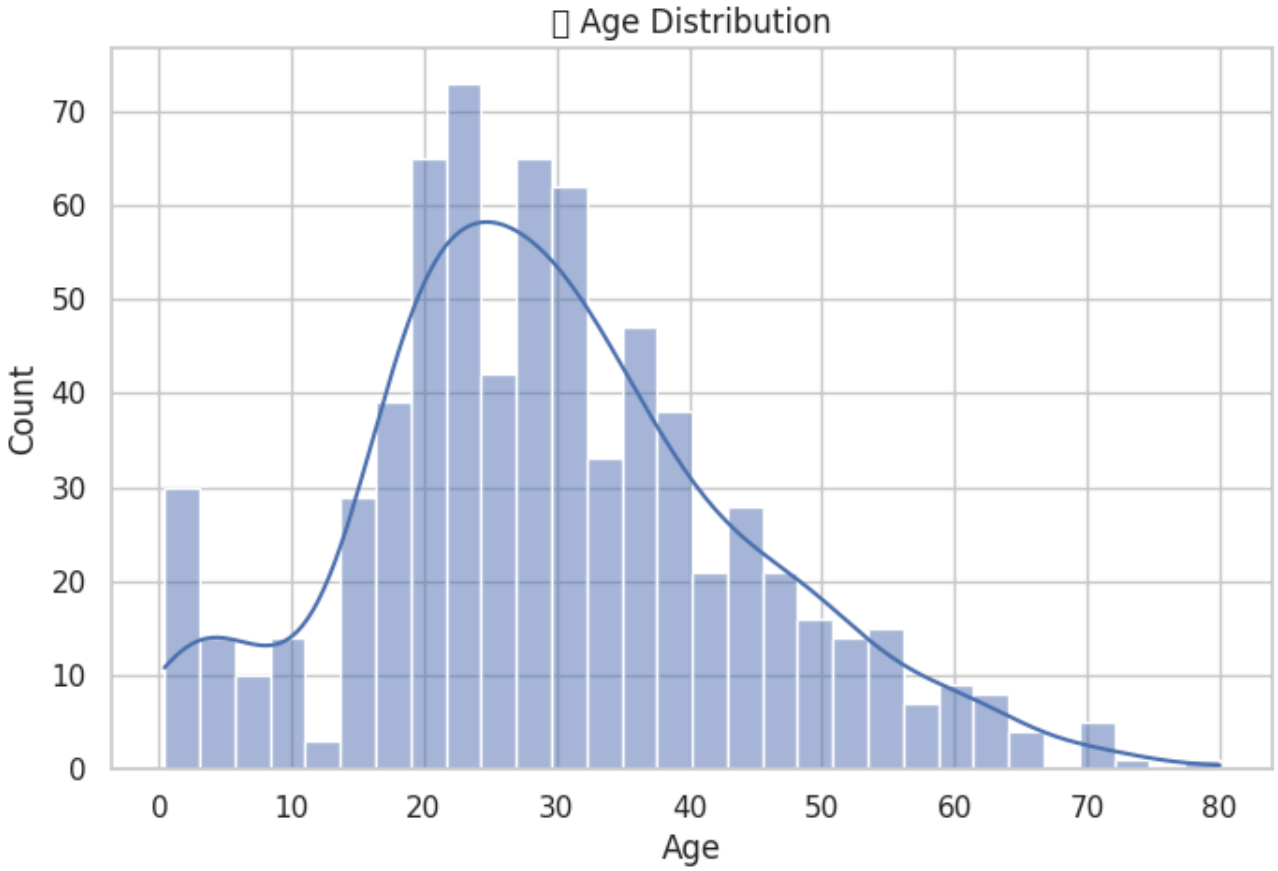
```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: (
  fig.canvas.print_figure(bytes_io, **kw)
```



Age Distribution & Missing Values

```
df["Age"] = pd.to_numeric(df["Age"], errors="coerce")
plt.figure(figsize=(8, 5))
sns.histplot(df["Age"].dropna(), kde=True, bins=30)
plt.title("Age Distribution")
plt.xlabel("Age")
plt.show()
```

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```



Family Size (SibSp + Parch + 1)

```
plt.figure(figsize=(8, 5))
sns.countplot(x="FamilySize", data=df)
plt.title("Family Size Distribution")
plt.xlabel("Family Size")
plt.ylabel("Number of Passengers")
plt.show()
```