Exploratory Data Analysis on the Titanic Dataset

This notebook performs EDA to uncover patterns, trends, and anomalies in the Titanic dataset using Python.

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

sns.set(style="whitegrid")

df = pd.read_csv (r"C:\Users\dell\OneDrive\Desktop\Titanic-
Dataset.csv") # Adjust if using another dataset

df.head()
```

```
   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
```

```
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  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

            Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

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

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
```

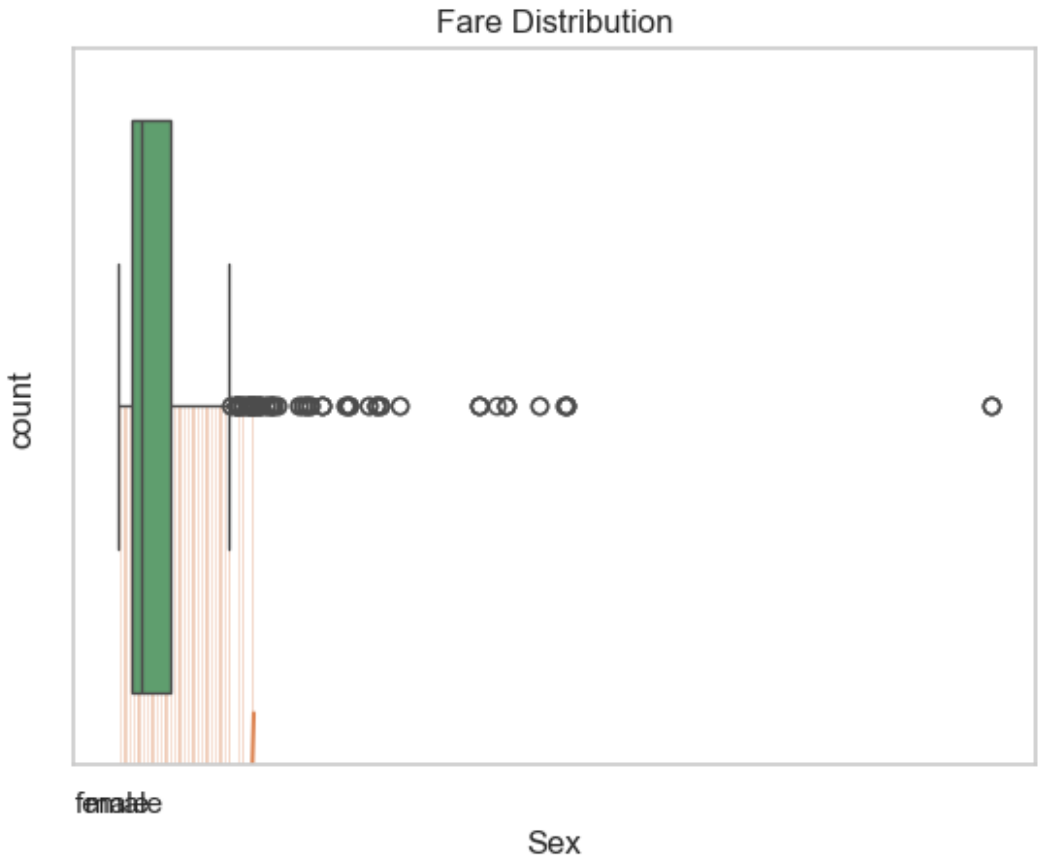
```
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
df.duplicated().sum()
```

```
0
```

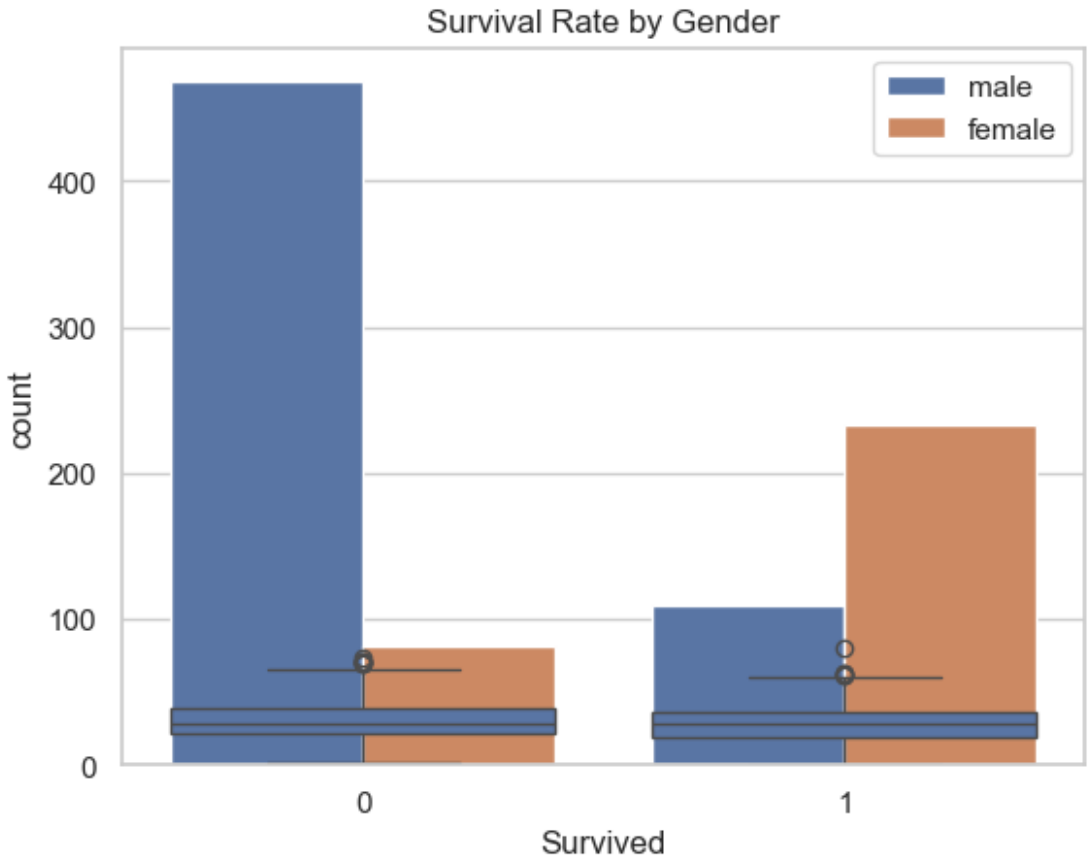
```
# Categorical
sns.countplot(x='Sex', data=df)
plt.title('Gender Distribution')

# Numerical
sns.histplot(df['Age'], bins=30, kde=True)
plt.title('Age Distribution')

sns.boxplot(x=df['Fare'])
plt.title('Fare Distribution')
```

```
Text(0.5, 1.0, 'Fare Distribution')
```



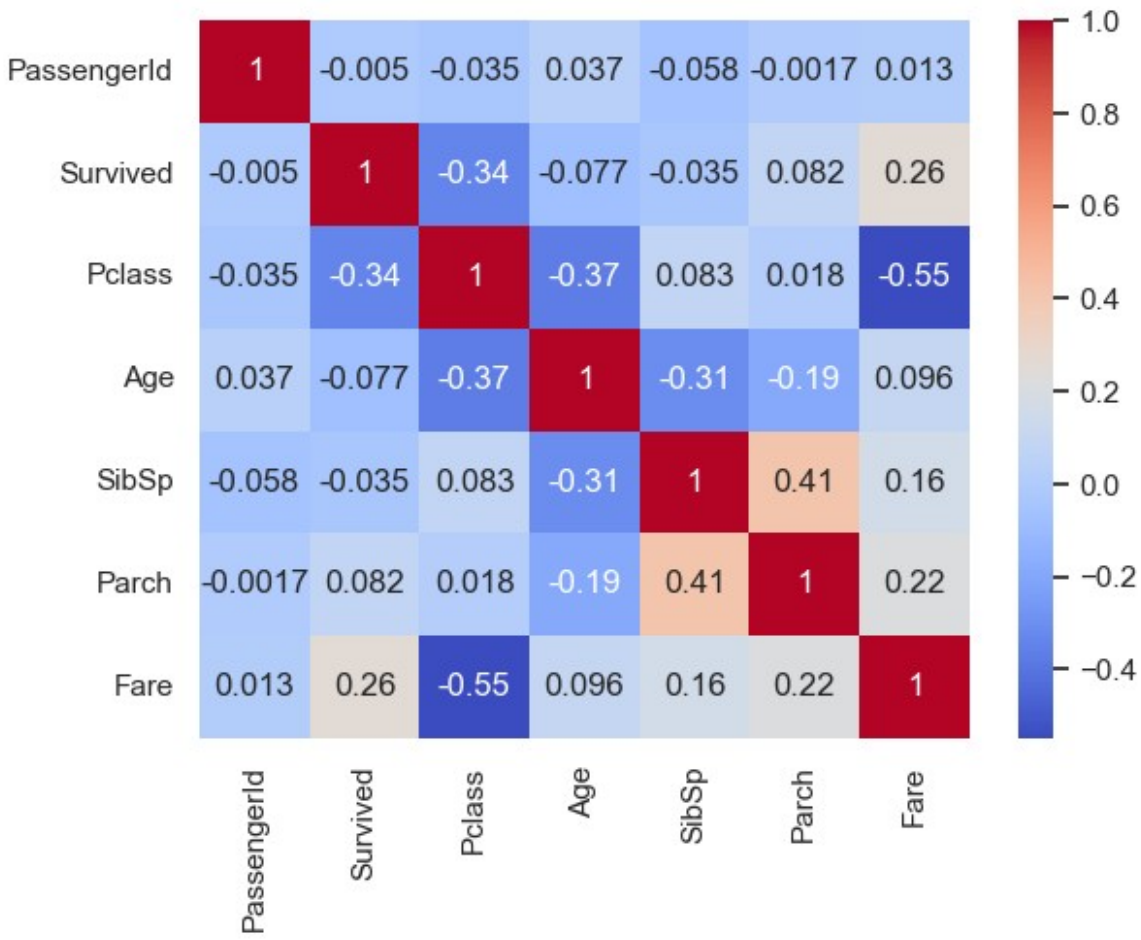
```
# Categorical vs Target
sns.countplot(x='Survived', hue='Sex', data=df)
plt.title('Survival Rate by Gender')

# Numerical vs Target
sns.boxplot(x='Survived', y='Age', data=df)
```

```
<Axes: title={'center': 'Survival Rate by Gender'}, xlabel='Survived',
ylabel='count'>
```



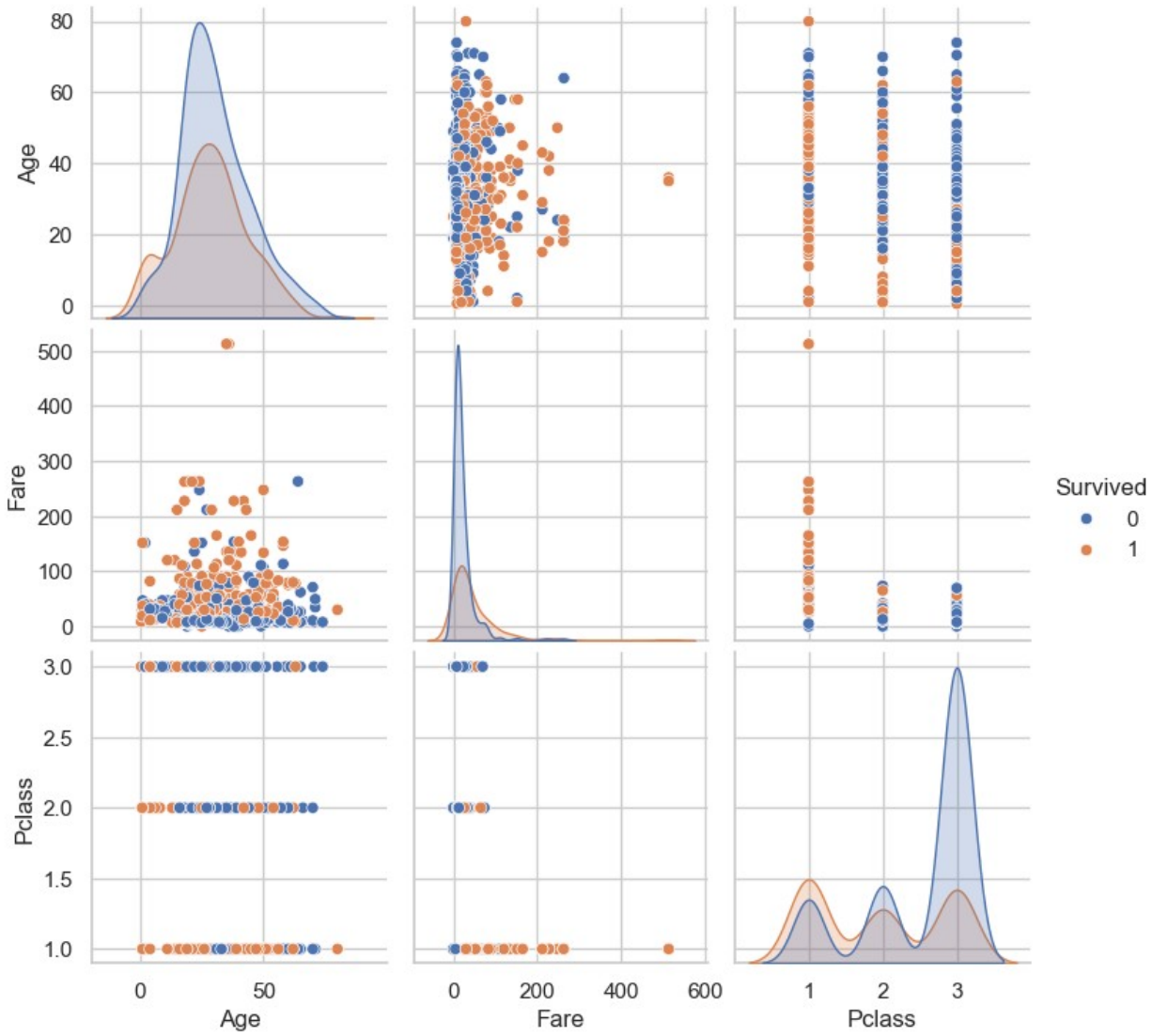

```
corr = df.corr(numeric_only=True)
sns.heatmap(corr, annot=True, cmap='coolwarm')
```

```
<Axes: >
```




```
sns.pairplot(df[['Age', 'Fare', 'Pclass', 'Survived']],
hue='Survived')
```

```
<seaborn.axisgrid.PairGrid at 0x221d8135310>
```




```
# Fill or drop based on context
df['Age'].fillna(df['Age'].median(), inplace=True)
df.drop(columns=['Cabin'], inplace=True)
```

```
C:\Users\dell\AppData\Local\Temp\ipykernel_876\68804755.py:2:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.
```

```
df['Age'].fillna(df['Age'].median(), inplace=True)
```

Summary of Findings

- Most passengers were in 3rd class and had lower survival rates.
- Females had a significantly higher survival rate.
- Older passengers tended to survive less than younger ones.
- Fare and Pclass had a correlation with survival.