Task 5: Exploratory Data Analysis (EDA)

Objective: Extract insights using visual and statistical exploration.

Tools: Python (Pandas, Matplotlib, Seaborn)

1. Importing Libraries

In [1]:

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

%matplotlib inline
sns.set(style="whitegrid")
```

2. Load Dataset

In [2]:

```
url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic
df = pd.read_csv(url)
df.head()
```

Out[2]:

	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	7.2
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

In [3]:

```
df.info()
df.describe()
df.isnull().sum()
df['Survived'].value_counts()
df['Sex'].value_counts()
```

```
<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
```

Out[3]:

```
Sex
male      577
female    314
Name: count, dtype: int64
```

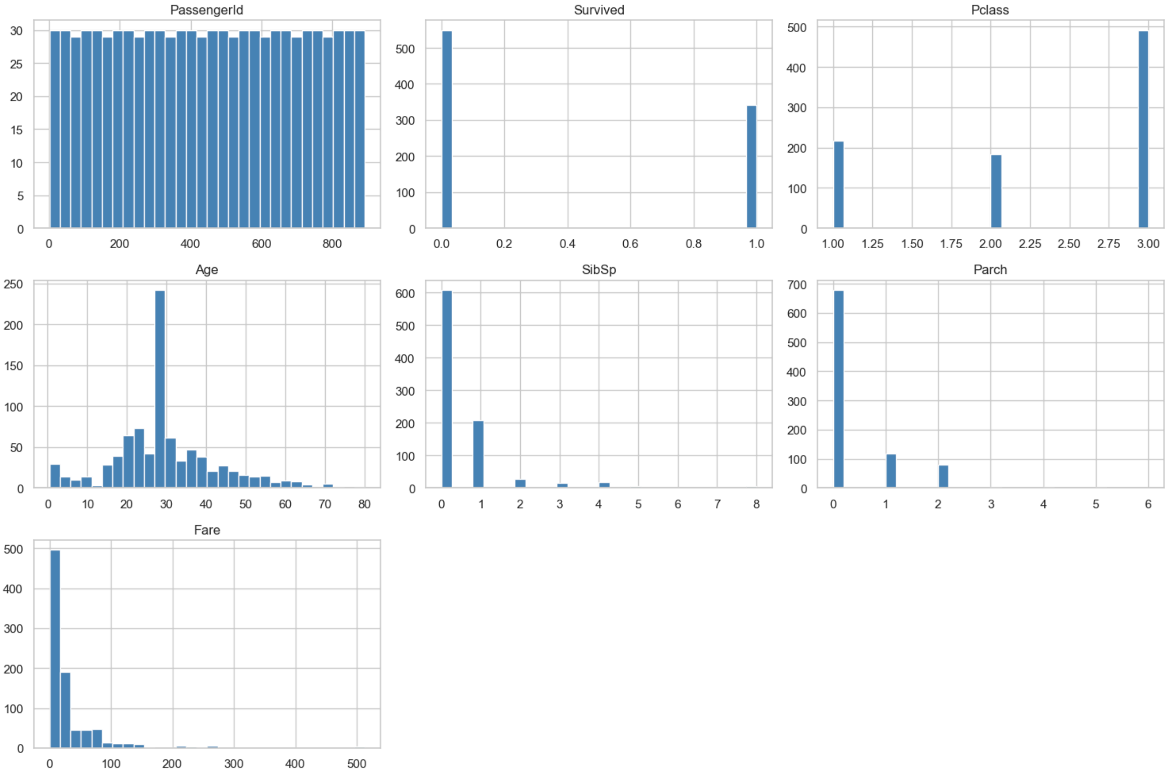
3. Handling Missing Values

In [4]:
```
df['Age'] = df['Age'].fillna(df['Age'].median())
df['Embarked'] = df['Embarked'].fillna(df['Embarked'].mode()[0])
```

4. Univariate Analysis

In [5]:
```
# Histograms
df.hist(bins=30, figsize=(15, 10), color='steelblue')
plt.tight_layout()

# Boxplot of Age
sns.boxplot(x=df['Age'])
```

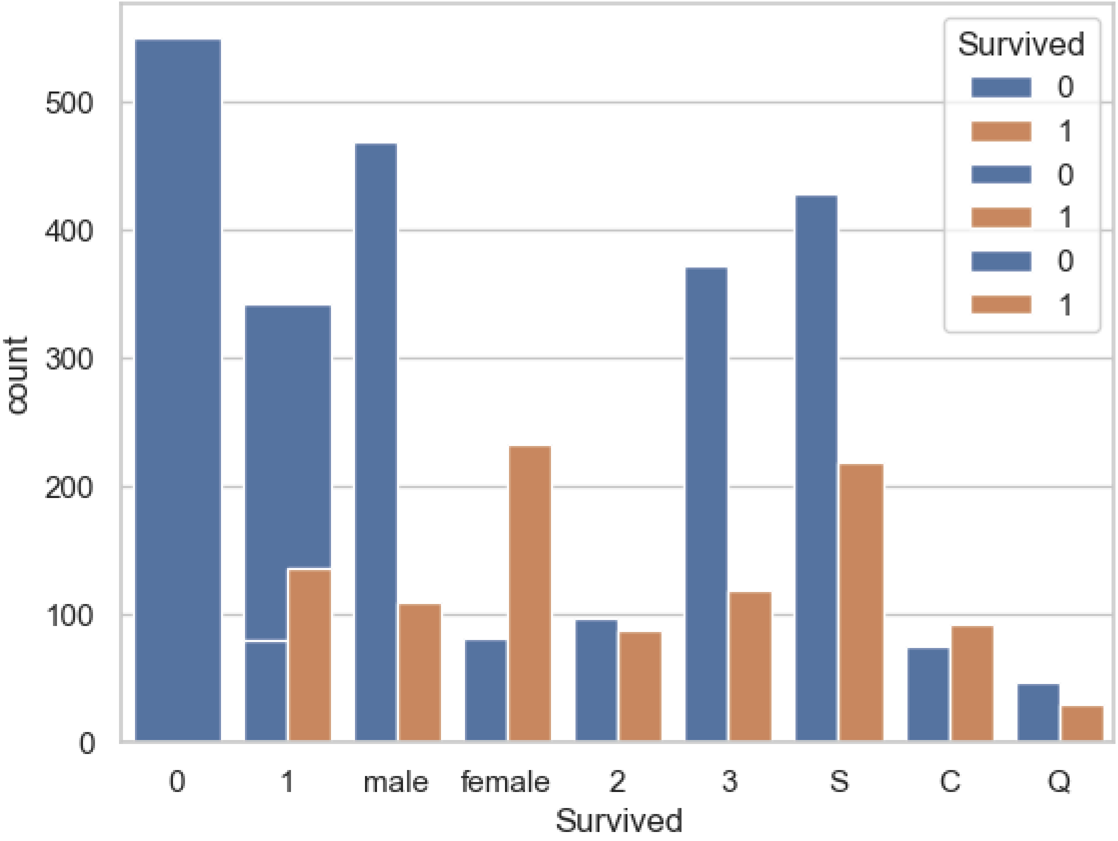
Out[5]:
```
<Axes: xlabel='Age'>
```



5. Categorical Analysis

In [6]:
```
# Countplots
sns.countplot(data=df, x='Survived')
sns.countplot(data=df, x='Sex', hue='Survived')
sns.countplot(data=df, x='Pclass', hue='Survived')
sns.countplot(data=df, x='Embarked', hue='Survived')
```

Out[6]:
```
<Axes: xlabel='Survived', ylabel='count'>
```



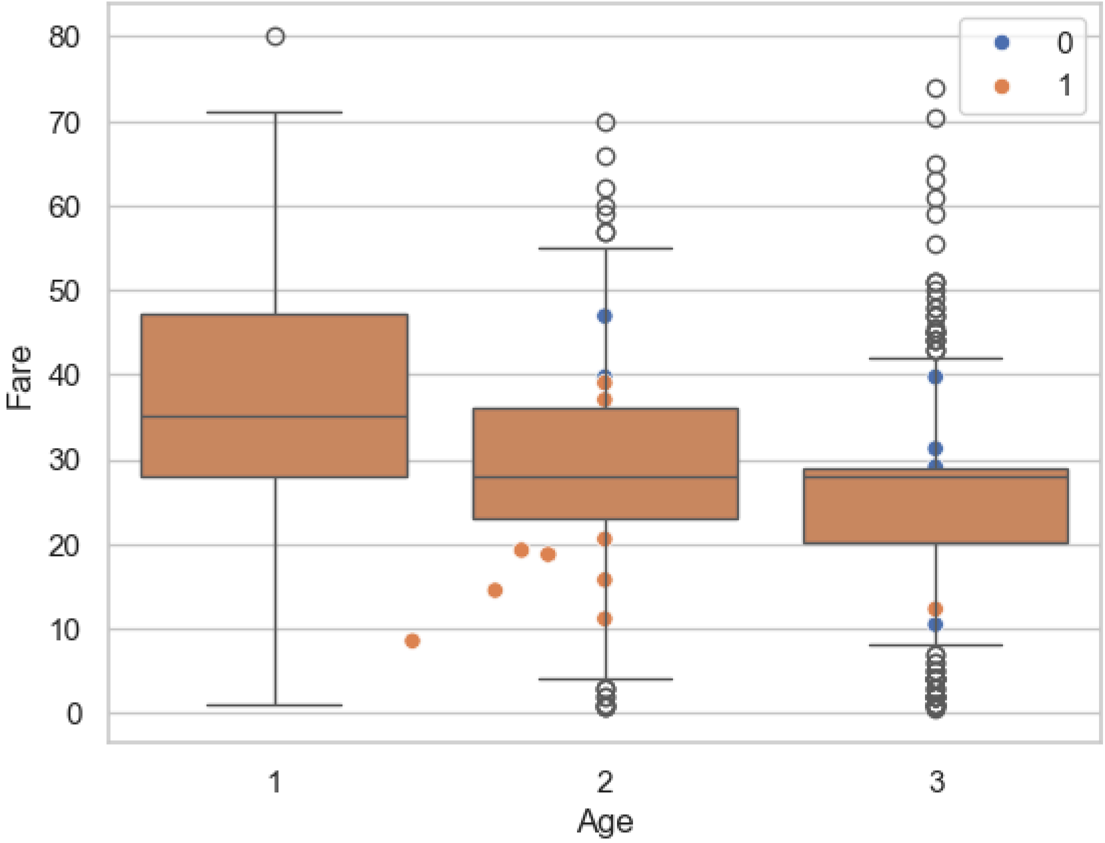
6. Bivariate Analysis

In [7]:

```
# Scatterplot of Age vs Fare
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=df)

# Boxplot of Age by Pclass
sns.boxplot(x='Pclass', y='Age', data=df)
```

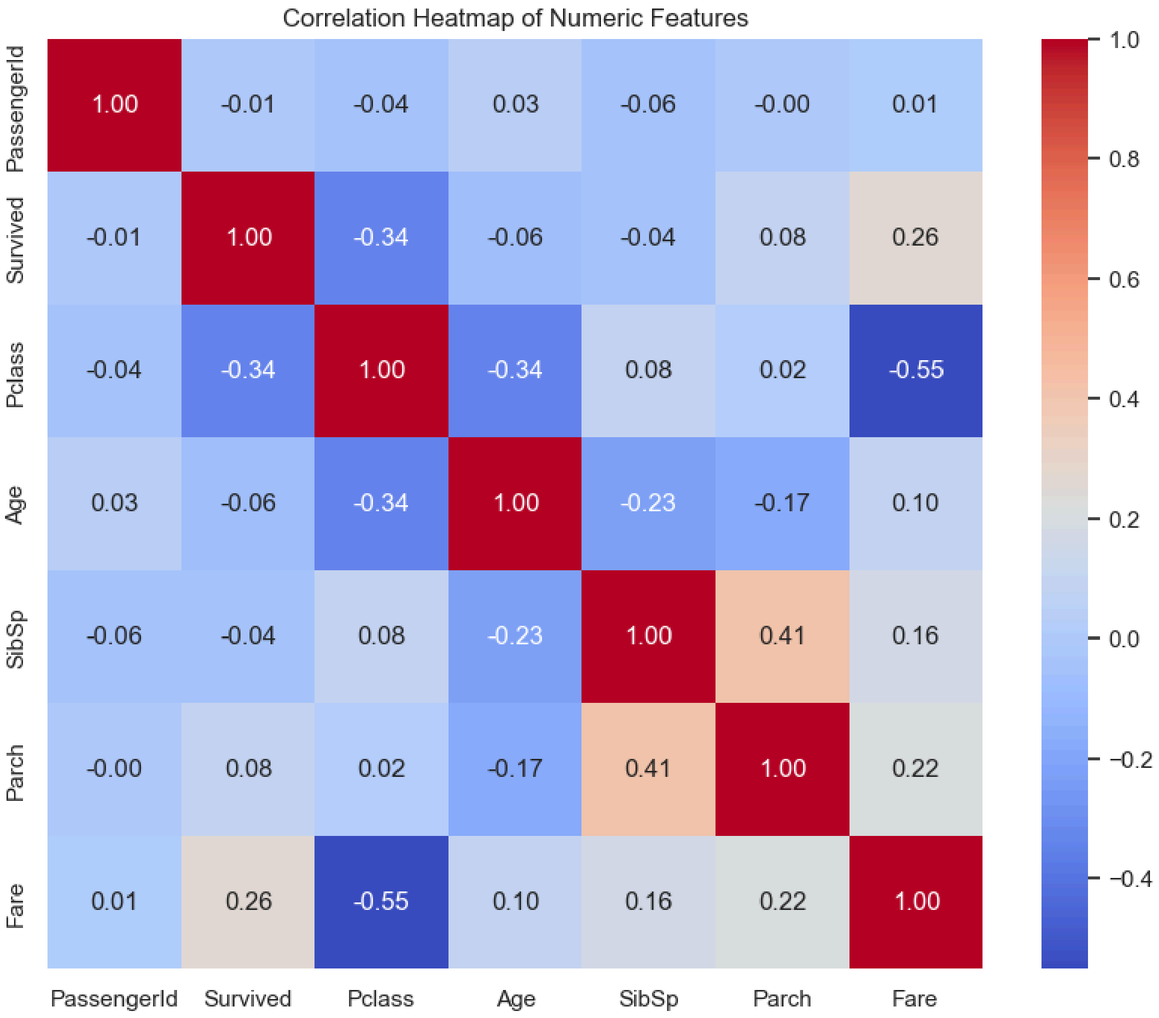
Out[7]:

```
<Axes: xlabel='Age', ylabel='Fare'>
```



7. Correlation Analysis

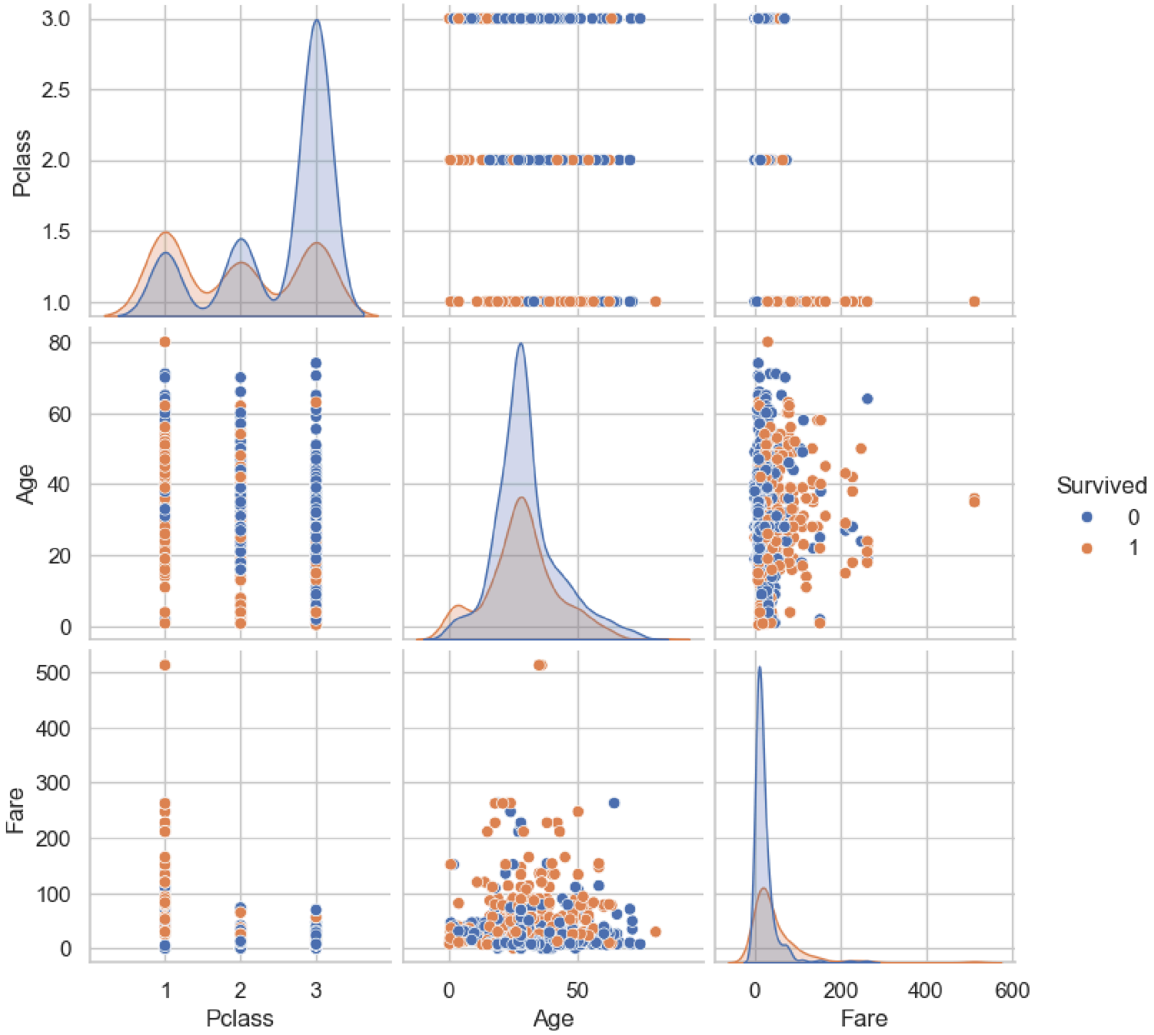
In [8]:
```
# Select only numeric columns to avoid string conversion issues
numeric_df = df.select_dtypes(include='number')

# Plot the correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title("Correlation Heatmap of Numeric Features")
plt.show()
```



8. Pairplot

```
In [9]: sns.pairplot(df[['Survived', 'Pclass', 'Sex', 'Age', 'Fare']], hue='Survived')
```

```
Out[9]: <seaborn.axisgrid.PairGrid at 0x23989f73d10>
```


Pclass
Age
Fare
3.0
2.5
2.0
1.5
1.0
80
60
40
20
0
500
400
300
200
100
0
1
2
3
0
50
0
200
400
600
Pclass
Age
Fare
Survived
0
1