

```
In [1]: #Import Packages  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns
```

```
In [2]: # Load the dataset  
df = pd.read_csv(r"C:\Users\HP\OneDrive\Desktop\ElevateLabs\Datasets\train_5
```

```
In [3]: # Dataset info  
print(df.info())  
  
# Statistical description  
print(df.describe())  
  
# Count of unique values in target column  
print(df['Survived'].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
None

```

	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

Survived

0 549

1 342

Name: count, dtype: int64

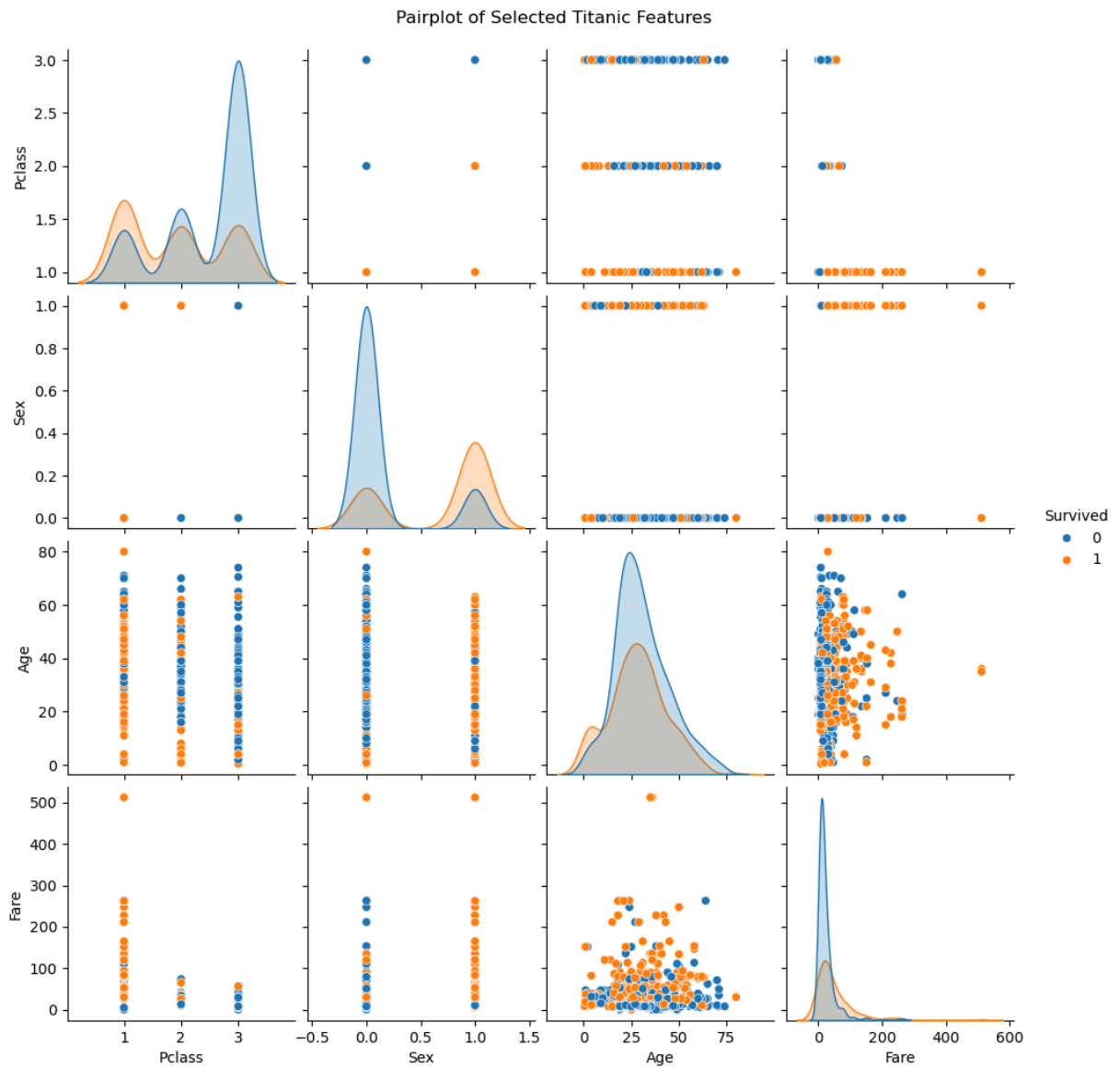
```

In [4]: # Select a few important features
selected_cols = ['Survived', 'Pclass', 'Sex', 'Age', 'Fare']
df_viz = df[selected_cols].dropna()

# Convert 'Sex' to numeric for pairplot
df_viz['Sex'] = df_viz['Sex'].map({'male': 0, 'female': 1})

# Plot pairplot
sns.pairplot(df_viz, hue='Survived')
plt.suptitle('Pairplot of Selected Titanic Features', y=1.02)
plt.show()

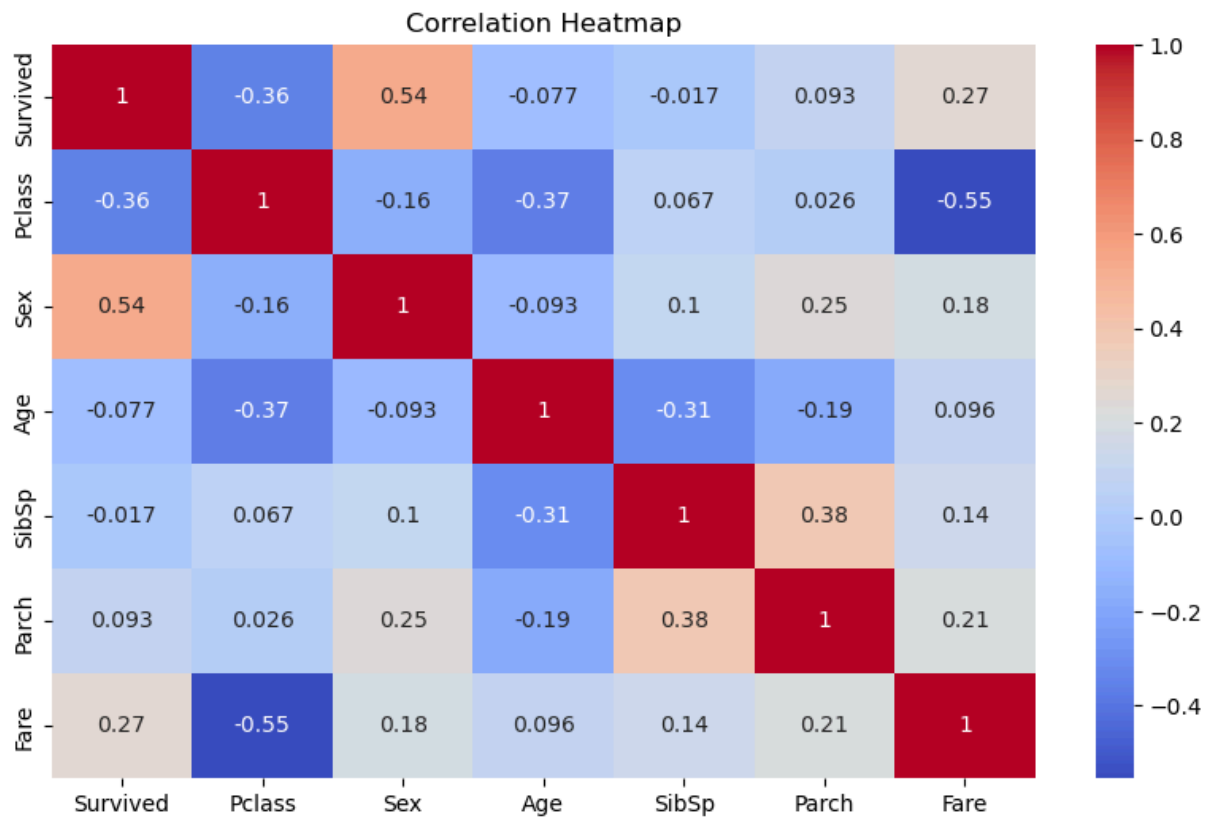
```



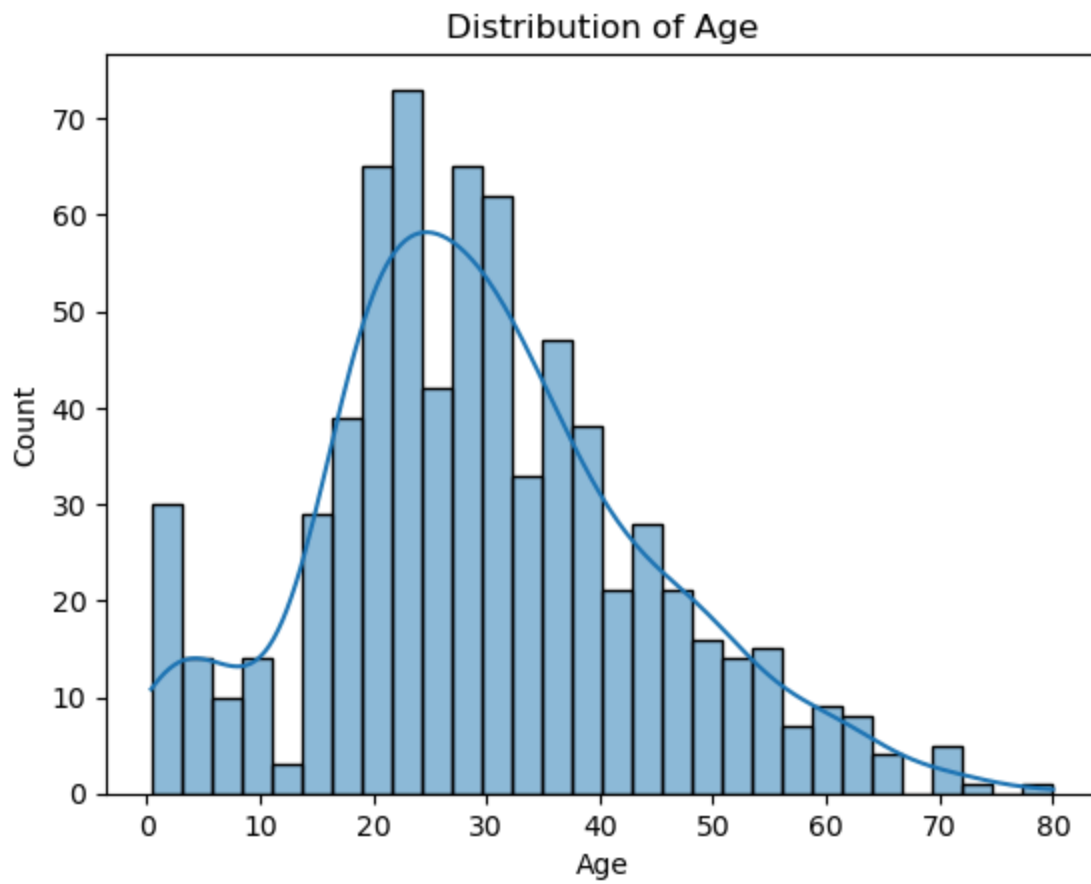
```
In [5]: # Convert categorical columns for correlation analysis
df_corr = df[['Survived', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare']]
df_corr['Sex'] = df_corr['Sex'].map({'male': 0, 'female': 1})

# Drop missing values
df_corr.dropna(inplace=True)

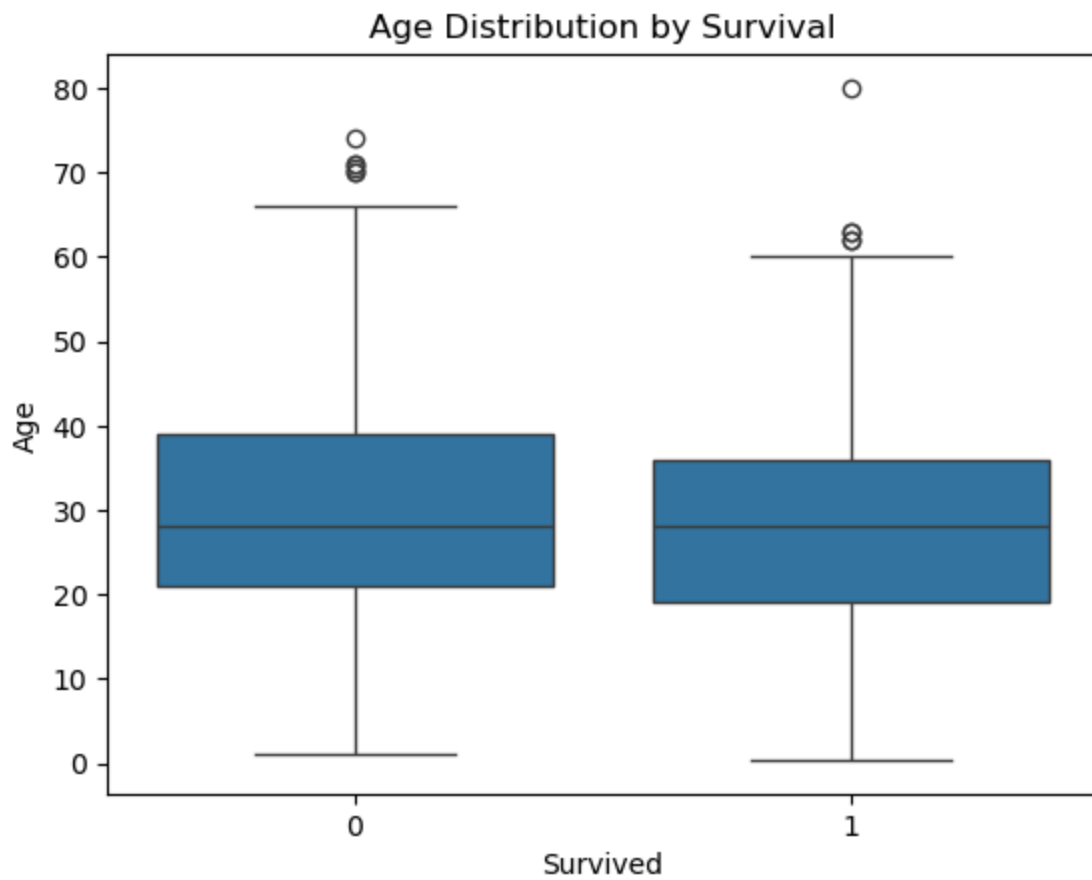
# Correlation heatmap
plt.figure(figsize=(10,6))
sns.heatmap(df_corr.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



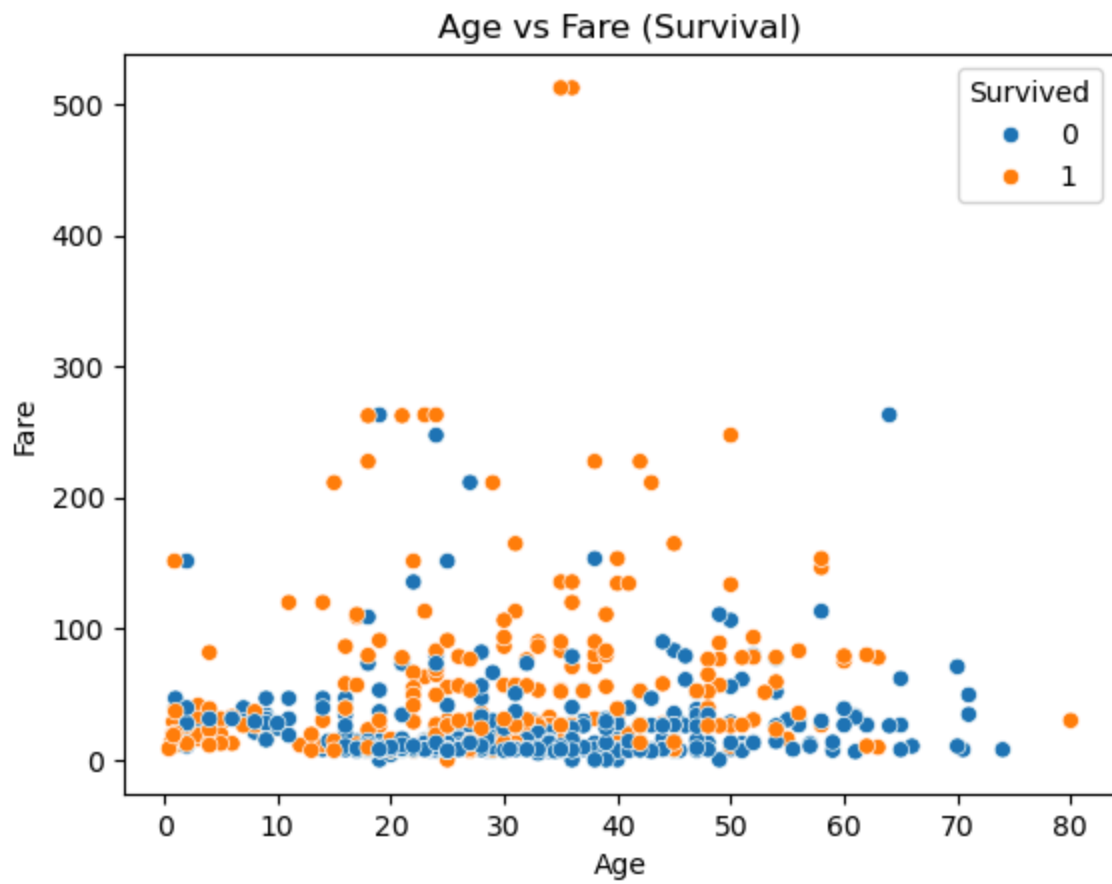
```
In [6]: # Histogram of Age
sns.histplot(df['Age'].dropna(), bins=30, kde=True)
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.show()
```



```
In [7]: # Boxplot of Age by Survival
sns.boxplot(x='Survived', y='Age', data=df)
plt.title('Age Distribution by Survival')
plt.show()
```



```
In [8]: # Scatterplot: Age vs Fare with survival hue
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=df)
plt.title('Age vs Fare (Survival)')
plt.show()
```



In [ ]: