# task2prodigy

February 4, 2025

```
[2]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.preprocessing import LabelEncoder
```

```
[3]: df = pd.read_csv("/content/sample_data/Titanic-Dataset.csv")
```

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

```
[4]:    Unnamed: 0  PassengerId  Sex  Age  Fare  Cabin  Embarked  Title  \
     0           0          892    0    2     0    2.0         2      0
     1           1          893    1    3     0    2.0         0      2
     2           2          894    0    3     0    2.0         2      0
     3           3          895    0    2     0    2.0         0      0
     4           4          896    1    1     0    2.0         0      2

        FamilySize  Survived
     0         0.0         0
     1         0.4         1
     2         0.0         0
     3         0.0         0
     4         0.8         1
```

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

```
[5]: Unnamed: 0     0
     PassengerId    0
     Sex            0
     Age            0
     Fare           0
     Cabin          0
     Embarked       0
     Title          0
     FamilySize     0
     Survived       0
     dtype: int64
```

```
[7]: df.shape
```

```
[7]: (418, 10)
```

```
[8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Unnamed: 0   418 non-null    int64
 1   PassengerId  418 non-null    int64
 2   Sex          418 non-null    int64
 3   Age          418 non-null    int64
 4   Fare         418 non-null    int64
 5   Cabin        418 non-null    float64
 6   Embarked     418 non-null    int64
 7   Title        418 non-null    int64
 8   FamilySize   418 non-null    float64
 9   Survived     418 non-null    int64
dtypes: float64(2), int64(8)
memory usage: 32.8 KB
```

```
[12]: df['Age'].fillna(df['Age'].median(), inplace=True)
      df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
      df.isnull().sum()
```

```
<ipython-input-12-ed6a0c6ccfed>:1: 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)
<ipython-input-12-ed6a0c6ccfed>: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['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
```

[12]:

```
Unnamed: 0     0
PassengerId    0
Sex            0
Age            0
Fare           0
Cabin          0
Embarked       0
Title          0
FamilySize     0
Survived       0
dtype: int64
```

[13]:

```
label_encoder = LabelEncoder()
df['Sex'] = label_encoder.fit_transform(df['Sex'])
df['Embarked'] = label_encoder.fit_transform(df['Embarked'])
df.head()
```

[13]:

```
   Unnamed: 0  PassengerId  Sex  Age  Fare  Cabin  Embarked  Title  \
0           0          892    0    2     0    2.0         2      0
1           1          893    1    3     0    2.0         0      2
2           2          894    0    3     0    2.0         2      0
3           3          895    0    2     0    2.0         0      0
4           4          896    1    1     0    2.0         0      2

   FamilySize  Survived
0         0.0         0
1         0.4         1
2         0.0         0
3         0.0         0
4         0.8         1
```

[14]:

```
df.describe()
```

[14]:

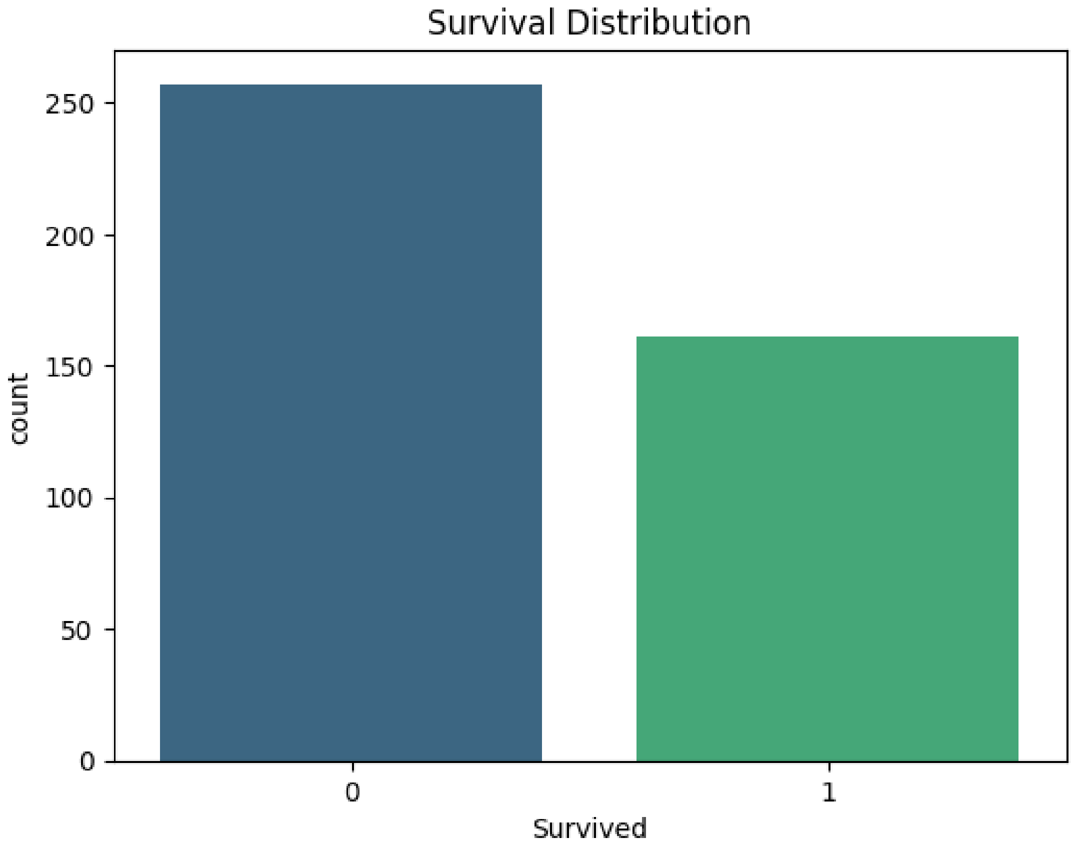
```
       Unnamed: 0  PassengerId         Sex         Age        Fare  \
count  418.000000   418.000000  418.000000  418.000000  418.000000
mean   208.500000  1100.500000    0.363636    1.763158    0.770335
std    120.810458   120.810458    0.481622    0.971479    0.998743
min      0.000000   892.000000    0.000000    0.000000    0.000000
25%    104.250000   996.250000    0.000000    1.000000    0.000000
50%    208.500000  1100.500000    0.000000    2.000000    0.000000
```

```
75%    312.750000  1204.750000    1.000000    3.000000    2.000000
max    417.000000  1309.000000    1.000000    4.000000    3.000000

            Cabin    Embarked       Title  FamilySize    Survived
count  418.000000  418.000000  418.000000  418.000000  418.000000
mean     1.687081    0.464115    0.732057    0.335885    0.385167
std      0.563371    0.685516    0.972019    0.607629    0.487218
min      0.000000    0.000000    0.000000    0.000000    0.000000
25%      1.600000    0.000000    0.000000    0.000000    0.000000
50%      2.000000    0.000000    0.000000    0.000000    0.000000
75%      2.000000    1.000000    1.000000    0.400000    1.000000
max      2.400000    2.000000    3.000000    4.000000    1.000000
```

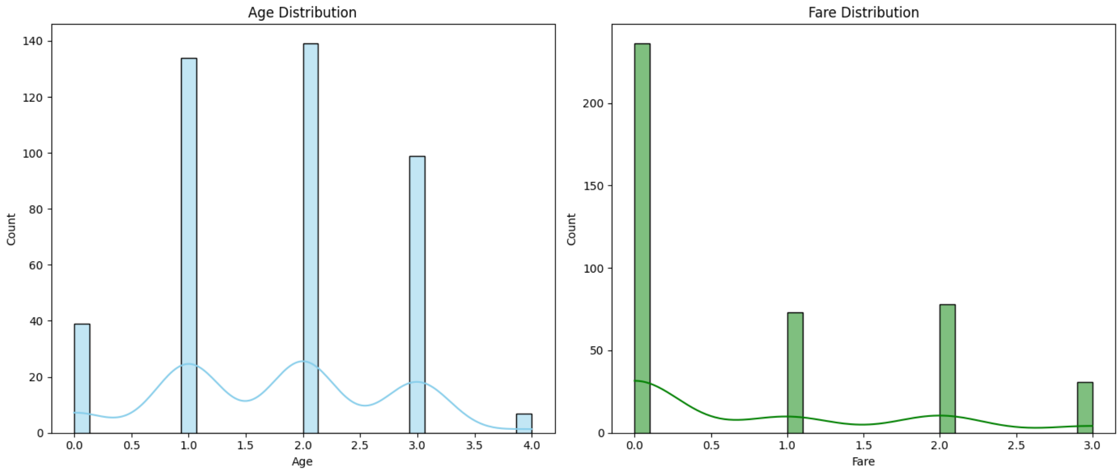
[15]:

```python
sns.countplot(data=df, x='Survived', palette='viridis')
plt.title('Survival Distribution')
plt.show()
```

```
<ipython-input-15-6f68ad396434>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(data=df, x='Survived', palette='viridis')
```

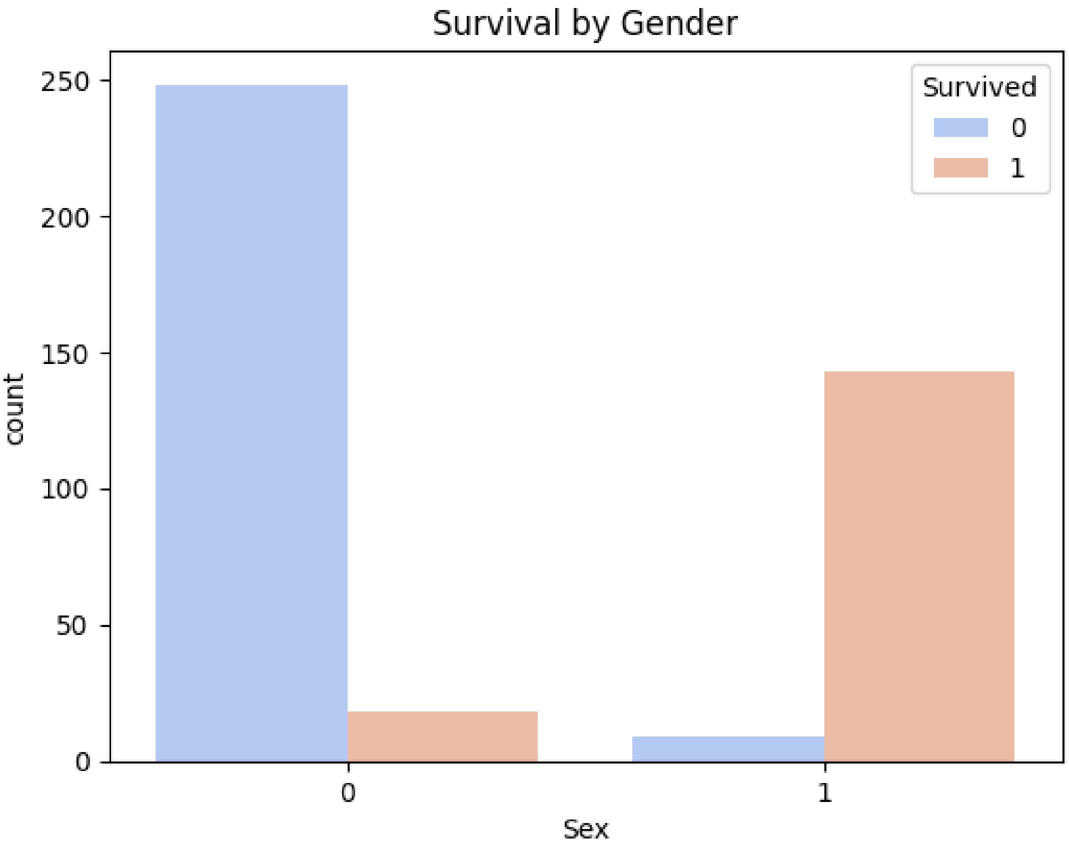


[16]:
```
fig, axes = plt.subplots(1, 2, figsize=(14, 6))
sns.histplot(df['Age'], kde=True, bins=30, color='skyblue', ax=axes[0])
axes[0].set_title('Age Distribution')
sns.histplot(df['Fare'], kde=True, bins=30, color='green', ax=axes[1])
axes[1].set_title('Fare Distribution')
plt.tight_layout()
plt.show()
```



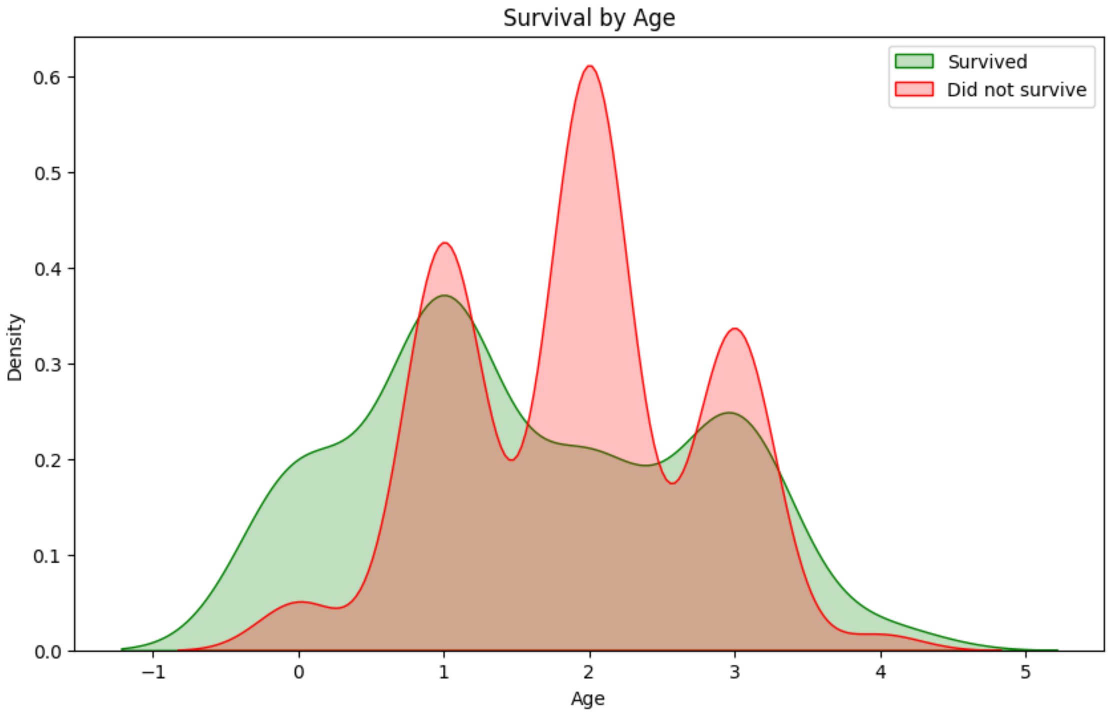
[20]:

```
sns.countplot(data=df, x='Sex', hue='Survived', palette='coolwarm')
plt.title('Survival by Gender')
plt.show()
```



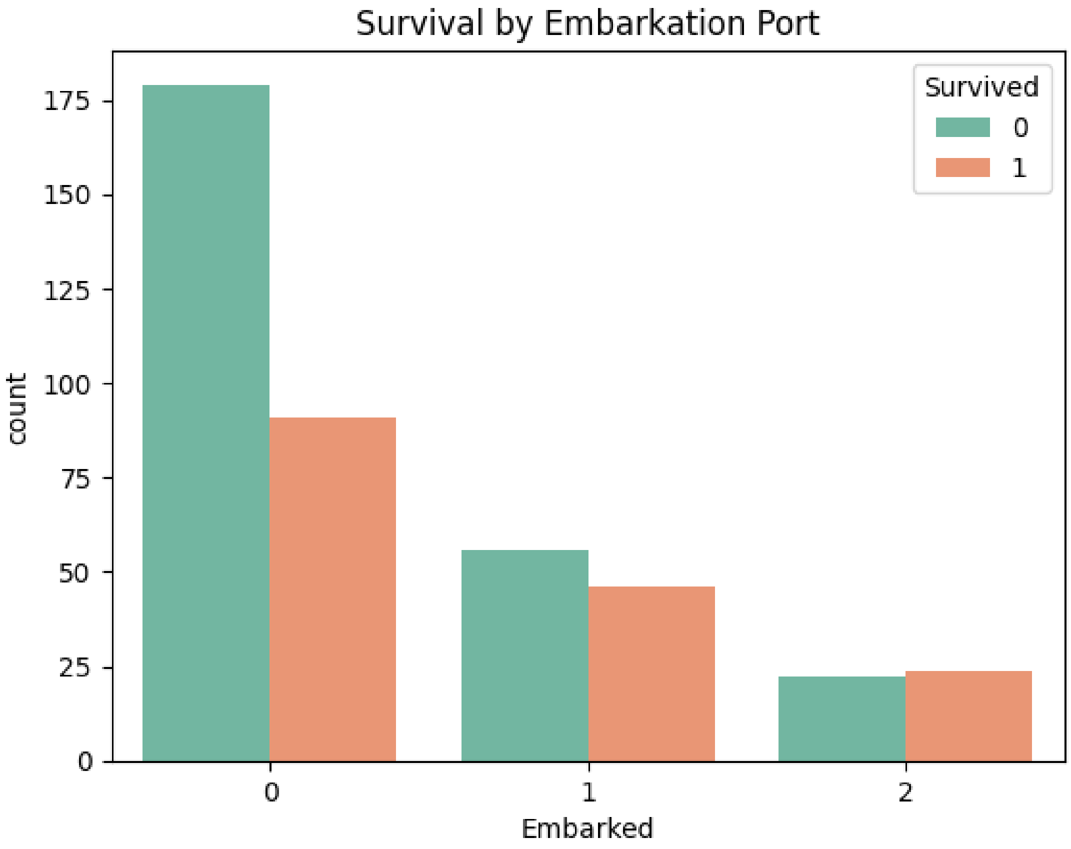
```
[21]: plt.figure(figsize=(10, 6))
      sns.kdeplot(data=df[df['Survived'] == 1], x='Age', shade=True, color='green',␣
       ↪label='Survived')
      sns.kdeplot(data=df[df['Survived'] == 0], x='Age', shade=True, color='red',␣
       ↪label='Did not survive')
      plt.title('Survival by Age')
      plt.legend()
      plt.show()
```

```
<ipython-input-21-14af344b8e7c>:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data=df[df['Survived'] == 1], x='Age', shade=True, color='green',
label='Survived')
<ipython-input-21-14af344b8e7c>:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data=df[df['Survived'] == 0], x='Age', shade=True, color='red',
label='Did not survive')
```



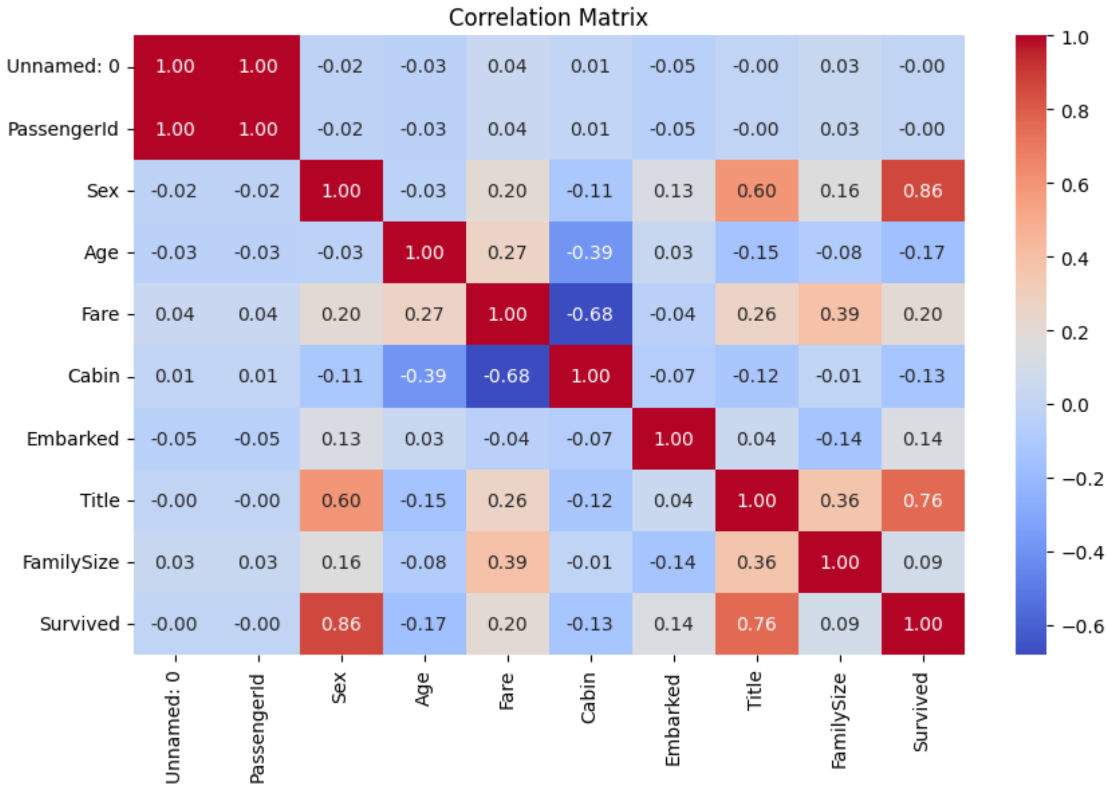
[22]:

```
sns.countplot(data=df, x='Embarked', hue='Survived', palette='Set2')
plt.title('Survival by Embarkation Port')
plt.show()
```

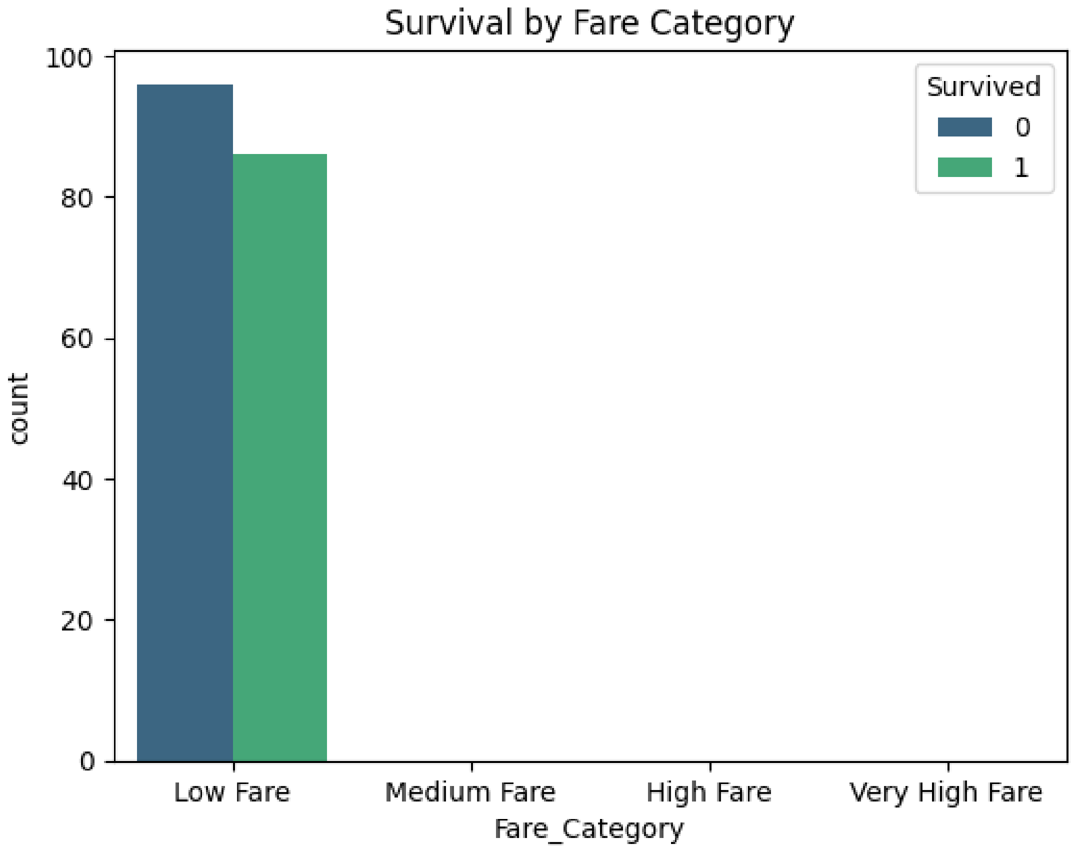


[23]:

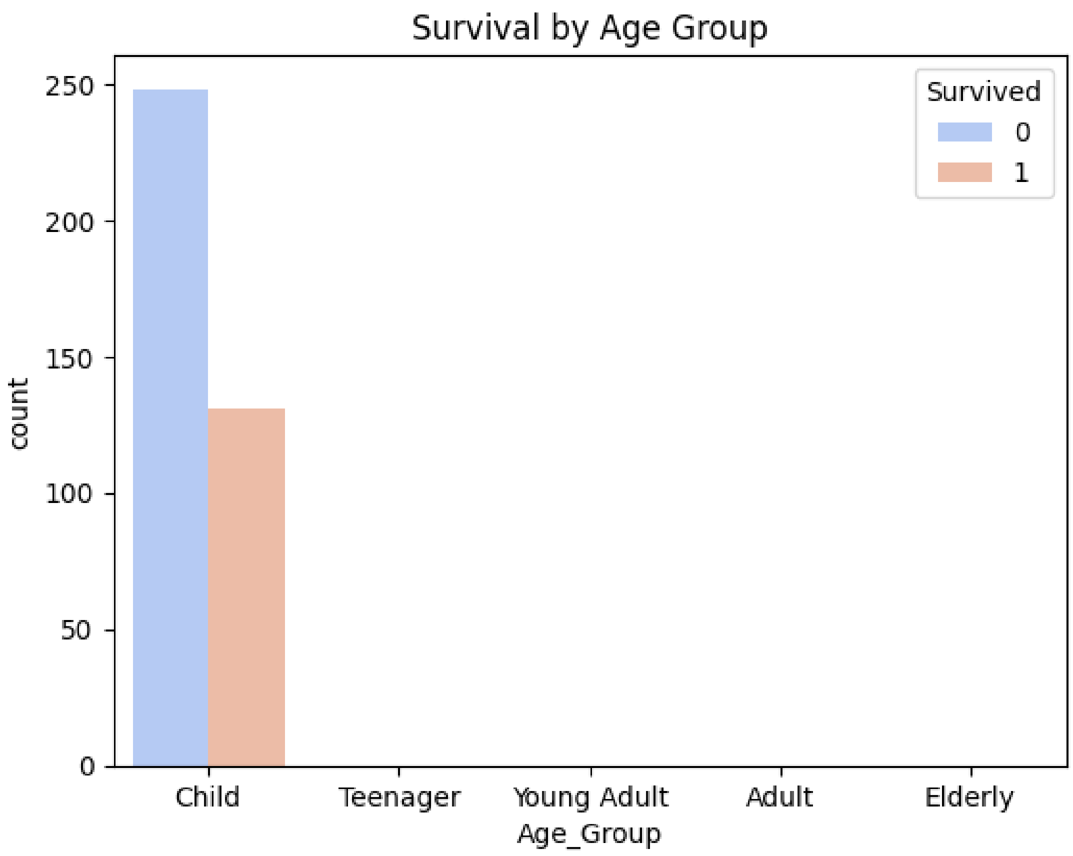
```
correlation_matrix = df.corr()
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
plt.show()
```



[26]:
```
bins = [0, 10, 50, 100, 500]
labels = ['Low Fare', 'Medium Fare', 'High Fare', 'Very High Fare']
df['Fare_Category'] = pd.cut(df['Fare'], bins=bins, labels=labels)
sns.countplot(data=df, x='Fare_Category', hue='Survived', palette='viridis')
plt.title('Survival by Fare Category')
plt.show()
```



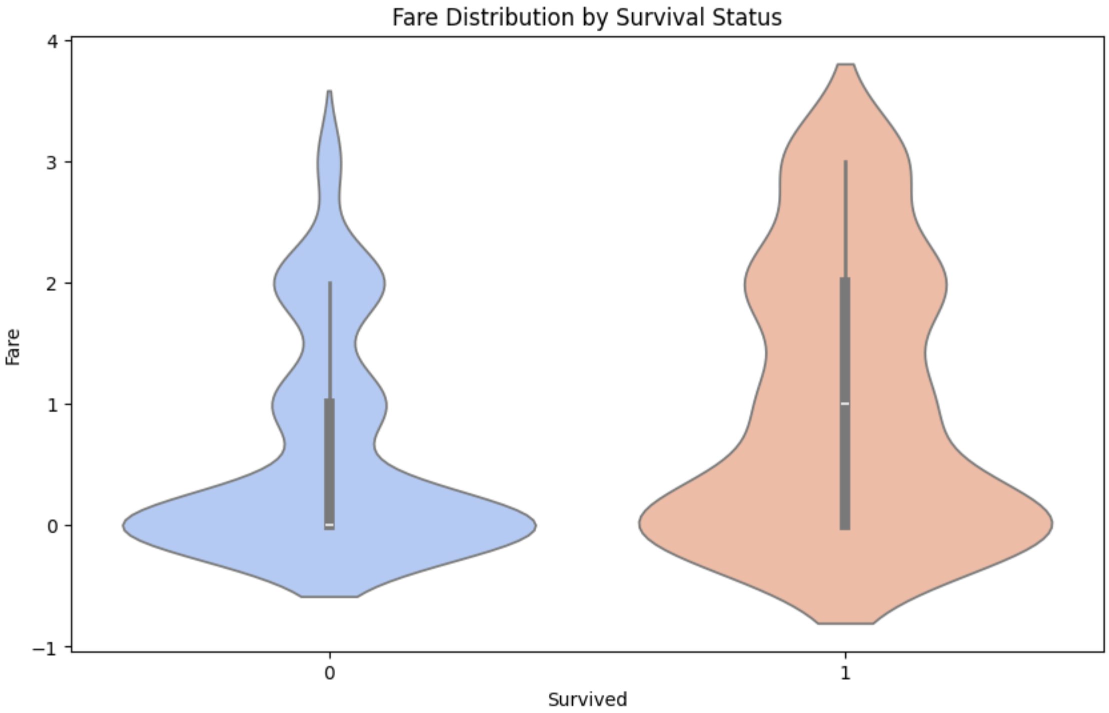
[27]:
```
bins = [0, 12, 18, 30, 50, 80]
labels = ['Child', 'Teenager', 'Young Adult', 'Adult', 'Elderly']
df['Age_Group'] = pd.cut(df['Age'], bins=bins, labels=labels)

sns.countplot(data=df, x='Age_Group', hue='Survived', palette='coolwarm')
plt.title('Survival by Age Group')
plt.show()
```



[29]:

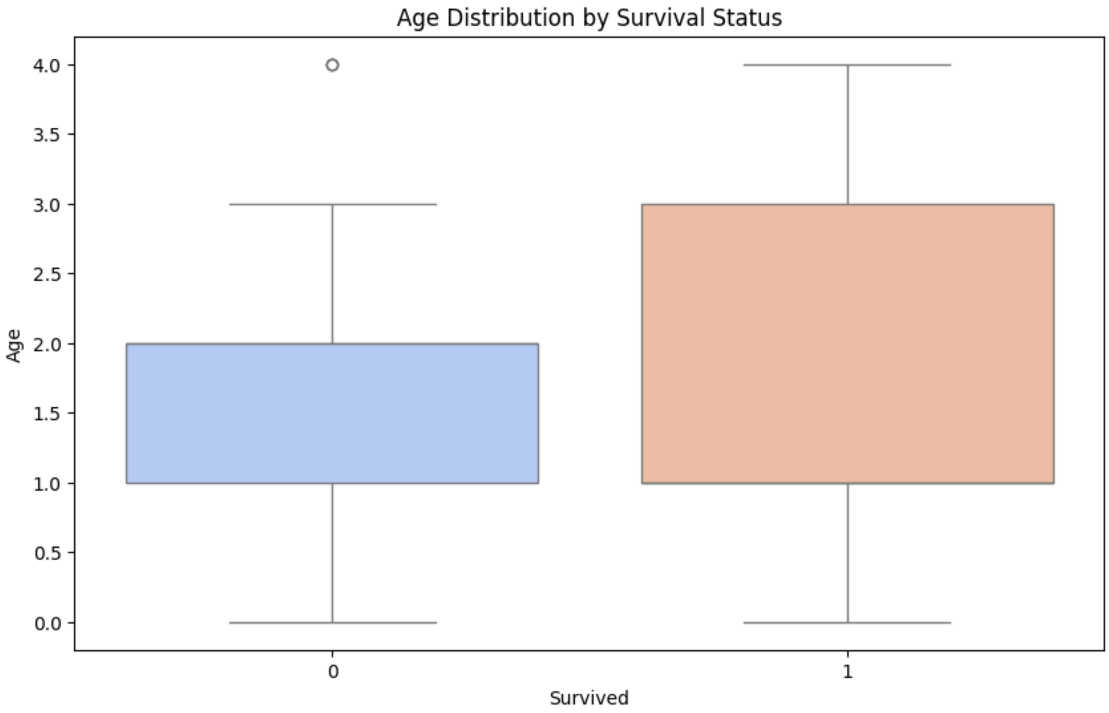
```
plt.figure(figsize=(10, 6))
sns.violinplot(data=df, x='Survived', y='Fare', palette='coolwarm')
plt.title('Fare Distribution by Survival Status')
plt.show()
```

```
<ipython-input-29-6936d498f861>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.violinplot(data=df, x='Survived', y='Fare', palette='coolwarm')
```



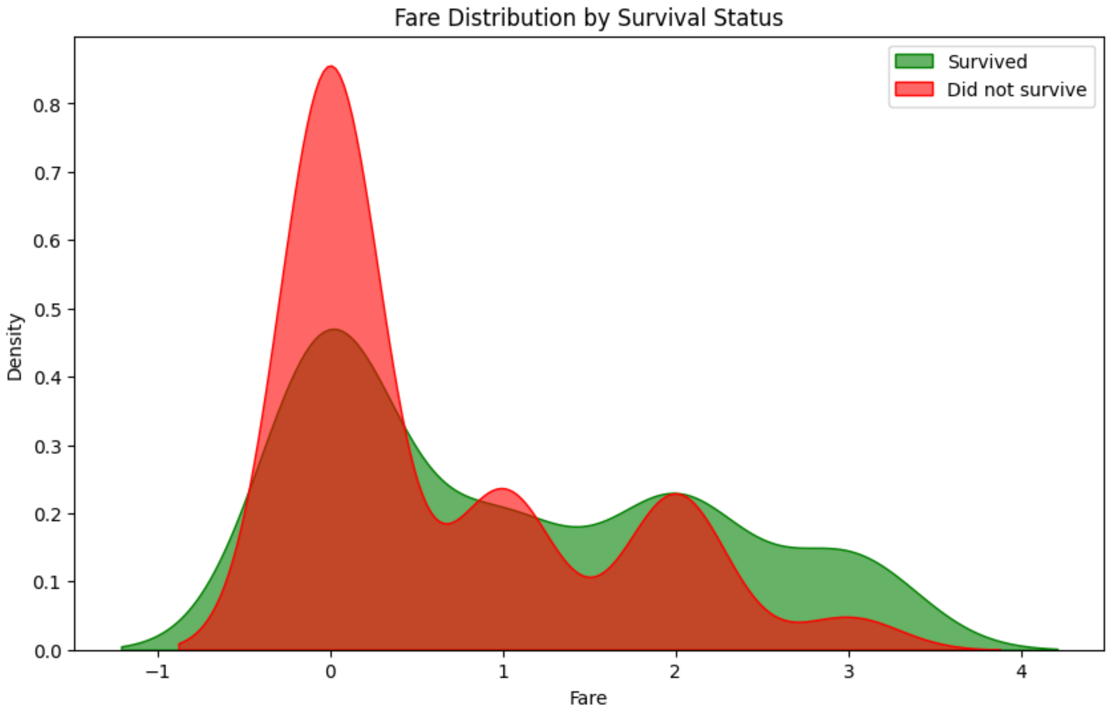
[32]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='Survived', y='Age', palette='coolwarm')
plt.title('Age Distribution by Survival Status')
plt.show()
```

```
<ipython-input-32-655ba0b8b119>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(data=df, x='Survived', y='Age', palette='coolwarm')
```



[36]:
```
plt.figure(figsize=(10, 6))
sns.kdeplot(data=df[df['Survived'] == 1], x='Fare', shade=True, color='green',␣
  ↪label='Survived', alpha=0.6)
sns.kdeplot(data=df[df['Survived'] == 0], x='Fare', shade=True, color='red',␣
  ↪label='Did not survive', alpha=0.6)
plt.title('Fare Distribution by Survival Status')
plt.legend()
plt.show()
```

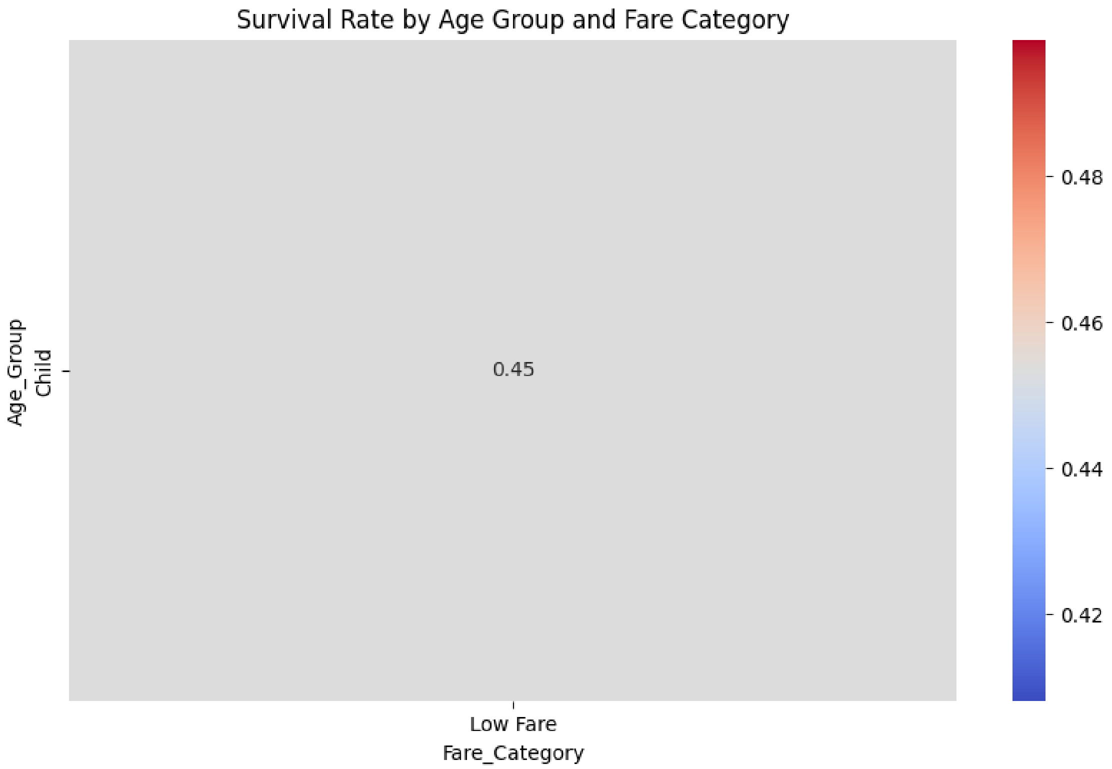
```
<ipython-input-36-7302e318dd1e>:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data=df[df['Survived'] == 1], x='Fare', shade=True, color='green',
label='Survived', alpha=0.6)
<ipython-input-36-7302e318dd1e>:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data=df[df['Survived'] == 0], x='Fare', shade=True, color='red',
label='Did not survive', alpha=0.6)
```



[40]:

```
age_bins = [0, 12, 18, 30, 50, 80]
age_labels = ['Child', 'Teenager', 'Young Adult', 'Adult', 'Elderly']
df['Age_Group'] = pd.cut(df['Age'], bins=age_bins, labels=age_labels)

fare_bins = [0, 10, 50, 100, 500]
fare_labels = ['Low Fare', 'Medium Fare', 'High Fare', 'Very High Fare']
df['Fare_Category'] = pd.cut(df['Fare'], bins=fare_bins, labels=fare_labels)

# Create a pivot table for Age Group and Fare Category vs Survived
pivot_table = df.pivot_table(index='Age_Group', columns='Fare_Category',␣
 ↪values='Survived', aggfunc='mean')

# Plot the heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(pivot_table, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Survival Rate by Age Group and Fare Category')
plt.show()
```

```
<ipython-input-40-9011431de304>:10: FutureWarning: The default value of
observed=False is deprecated and will change to observed=True in a future
version of pandas. Specify observed=False to silence this warning and retain the
current behavior
  pivot_table = df.pivot_table(index='Age_Group', columns='Fare_Category',
values='Survived', aggfunc='mean')
```


Survival Rate by Age Group and Fare Category
Age_Group
Child
0.45
Low Fare
Fare_Category
0.48
0.46
0.44
0.42