

```
# Data Handling
import pandas as pd
import numpy as np

# Visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Set default theme
sns.set(style="whitegrid")

# To display plots inside the notebook
%matplotlib inline
```

```
# Load the Titanic dataset
titanic = sns.load_dataset('titanic')

# Display the first 5 rows
titanic.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

Next steps:

[Generate code with titanic](#)[View recommended plots](#)[New interactive sheet](#)

```
# Basic info of dataset
titanic.info()

# Summary statistics for numeric and object columns
titanic.describe(include='all')

# Check for missing values
titanic.isnull().sum()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
#   Column             Non-Null Count  Dtype
---  -
0   survived            891 non-null    int64
1   pclass              891 non-null    int64
2   sex                 891 non-null    object
3   age                 714 non-null    float64
4   sibsp               891 non-null    int64
5   parch               891 non-null    int64
6   fare                891 non-null    float64
7   embarked            889 non-null    object
8   class               891 non-null    category
9   who                 891 non-null    object
10  adult_male          891 non-null    bool
11  deck                203 non-null    category
12  embark_town         889 non-null    object
13  alive               891 non-null    object
14  alone               891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB

```

	0
<b>survived</b>	0
<b>pclass</b>	0
<b>sex</b>	0
<b>age</b>	177
<b>sibsp</b>	0
<b>parch</b>	0
<b>fare</b>	0
<b>embarked</b>	2
<b>class</b>	0
<b>who</b>	0
<b>adult_male</b>	0
<b>deck</b>	688
<b>embark_town</b>	2
<b>alive</b>	0
<b>alone</b>	0

dtypes: int64

```

# Check distribution of categorical columns
print(titanic['sex'].value_counts())
print(titanic['class'].value_counts())
print(titanic['embarked'].value_counts())

```

```

sex
male      577
female    314
Name: count, dtype: int64
class
Third     491
First     216
Second    184
Name: count, dtype: int64
embarked
S         644
C         168
Q          77
Name: count, dtype: int64

```

```

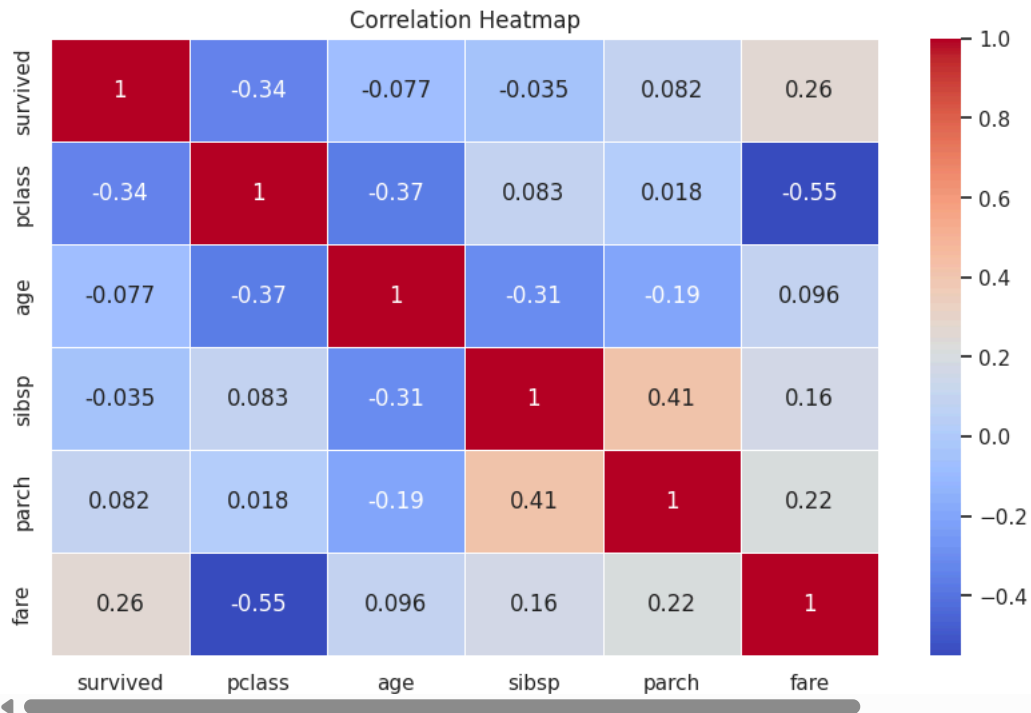
# Select only numeric columns for correlation
numeric_data = titanic.select_dtypes(include=['float64', 'int64'])

```

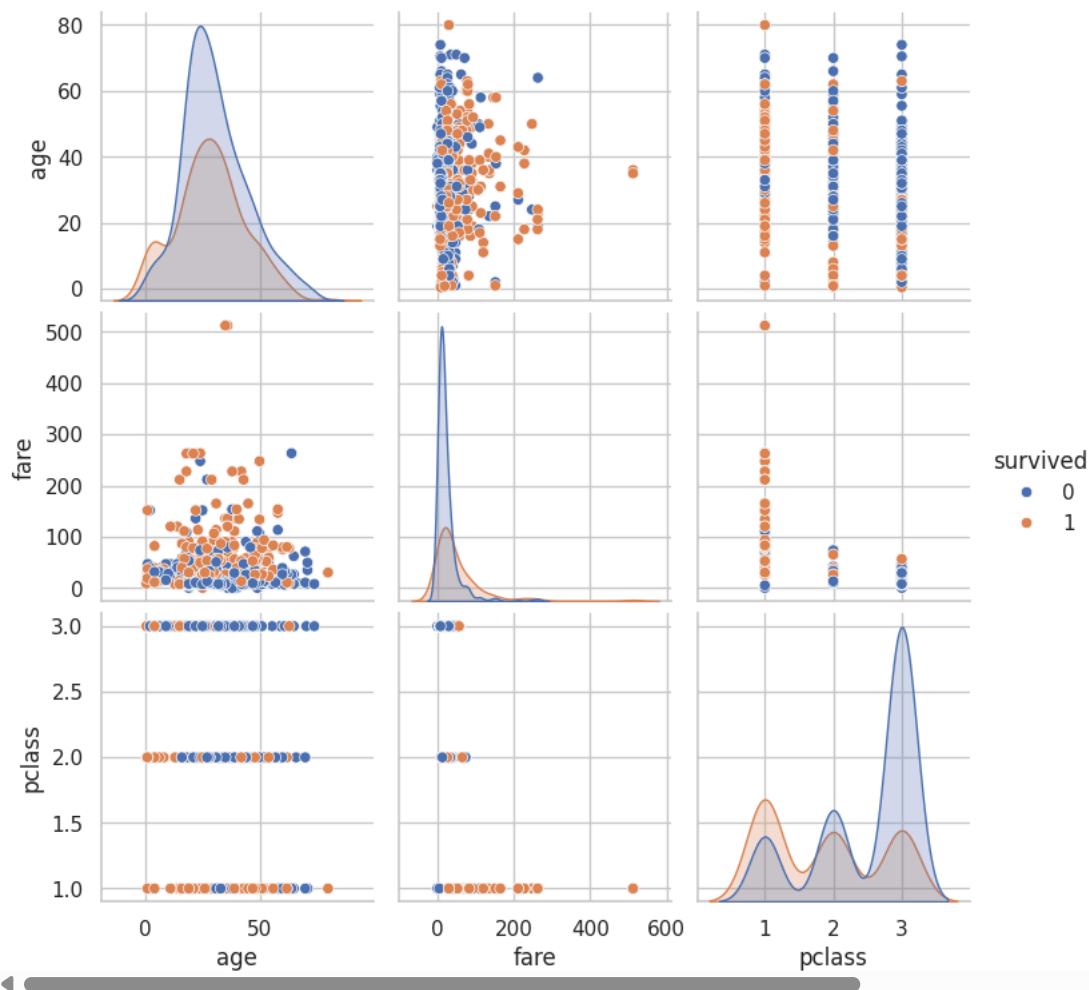
```

# Plot heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(numeric_data.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()

```



```
sns.pairplot(titanic[['age', 'fare', 'pclass', 'survived']].dropna(), hue='survived')
plt.show()
```



```
sns.histplot(titanic['age'].dropna(), kde=True)
plt.title("Age Distribution")
plt.xlabel("Age")
plt.ylabel("Count")
plt.show()
```

```
sns.boxplot(x='class', y='fare', data=titanic)
plt.title("Fare by Class")
```

```
plt.show()
```

```
sns.countplot(x='sex', hue='survived', data=titanic)
plt.title("Survival by Gender")
plt.show()
```

```
sns.scatterplot(x='age', y='fare', hue='survived', data=titanic)
plt.title("Age vs Fare by Survival")
plt.show()
```

