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

sns.set(style='whitegrid')
plt.rcParams['figure.figsize'] = (8,5)
```

```
# Load training dataset
df = pd.read_csv("train.csv")
df.head()
```

	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
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8

Next steps: Generate code with `df` New interactive sheet

```
print("Shape:", df.shape)
print("\nInfo:")
print(df.info())

print("\nSummary Statistics:")
df.describe(include='all').T

print("\nMissing values:")
print(df.isnull().sum())
```

```
print("\nDuplicate rows:", df.duplicated().sum())
```

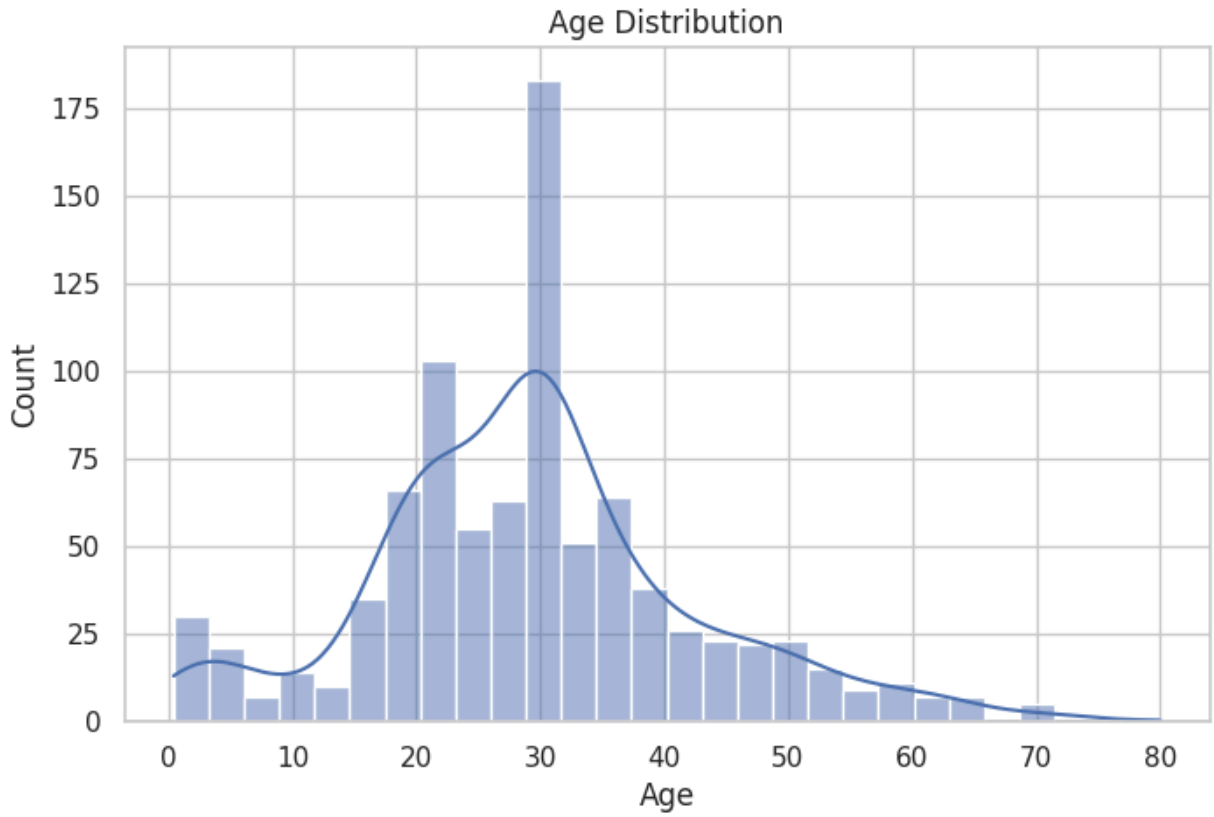
```
Shape: (891, 17)

Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 17 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          891 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     891 non-null    object
 12  Has_Cabin    891 non-null    int64
 13  Title        891 non-null    object
 14  FamilySize   891 non-null    int64
 15  IsAlone      891 non-null    int64
 16  FareBand     891 non-null    int64
dtypes: float64(2), int64(9), object(6)
memory usage: 118.5+ KB
None

Summary Statistics:

Missing values:
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         0
Has_Cabin        0
Title            0
FamilySize       0
IsAlone          0
FareBand         0
dtype: int64

Duplicate rows: 0
```

Double-click (or enter) to edit

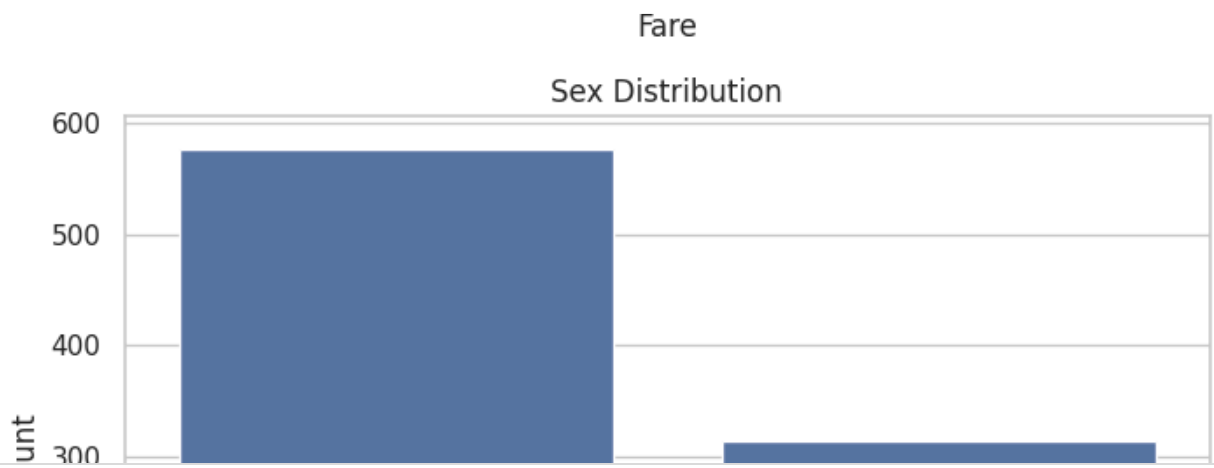
```
# Fill Embarked with mode\
df['Embarked'] = df['Embarked'].fillna(df['Embarked'].mode()[0])

# Fill Fare with median
df['Fare'] = df['Fare'].fillna(df['Fare'].median())

# Create Cabin flag
df['Has_Cabin'] = df['Cabin'].notnull().astype(int)

# Extract Title from Name
df['Title'] = df['Name'].str.extract(r',\s*([^\.]+)\.', expand=False)
df['Title'] = df['Title'].replace({'Mlle':'Miss','Ms':'Miss','Mme':'Mrs'})
common_titles = ['Mr','Mrs','Miss','Master']
df['Title'] = df['Title'].apply(lambda x: x if x in common_titles else 'Rare')

# Impute Age by Title median
df['Age'] = df.groupby('Title')['Age'].transform(lambda x: x.fillna(x.median()))
```

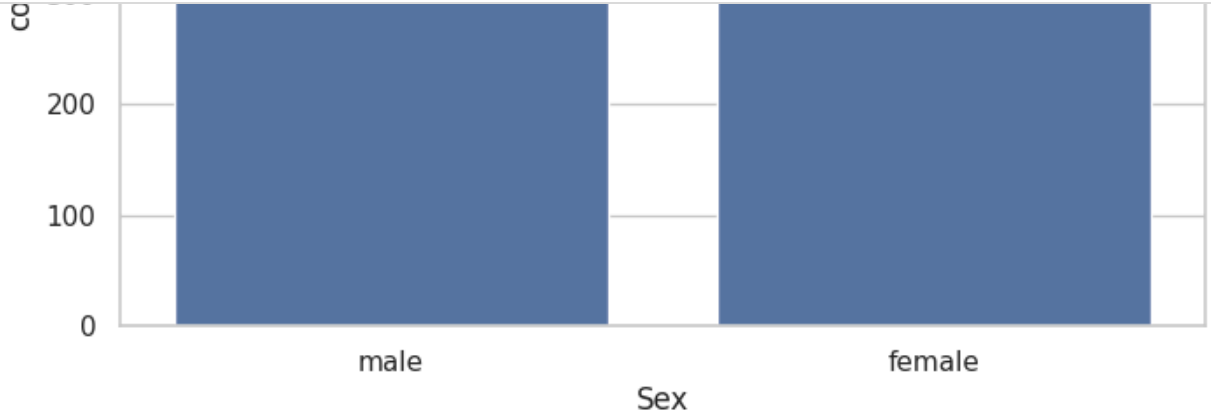
```
# Family size
df['FamilySize'] = df['SibSp'] + df['Parch'] + 1
df['IsAlone'] = (df['FamilySize']==1).astype(int)

# Fare band
df['FareBand'] = pd.qcut(df['Fare'], 4, labels=False)
```

```
# Age distribution
sns.histplot(df['Age'], kde=True)
plt.title("Age Distribution")
plt.show()

# Fare distribution
sns.histplot(df['Fare'], kde=True)
plt.title("Fare Distribution")
plt.show()

# Categorical counts
sns.countplot(x='Sex', data=df)
plt.title("Sex Distribution")
plt.show()

sns.countplot(x='Pclass', data=df)
plt.title("Pclass Distribution")
plt.show()
```




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

sns.countplot(x='Pclass', hue='Survived', data=df)
plt.title("Survival by Pclass")
plt.show()

sns.boxplot(x='Survived', y='Age', data=df)
plt.title("Age vs Survival")
plt.show()
```

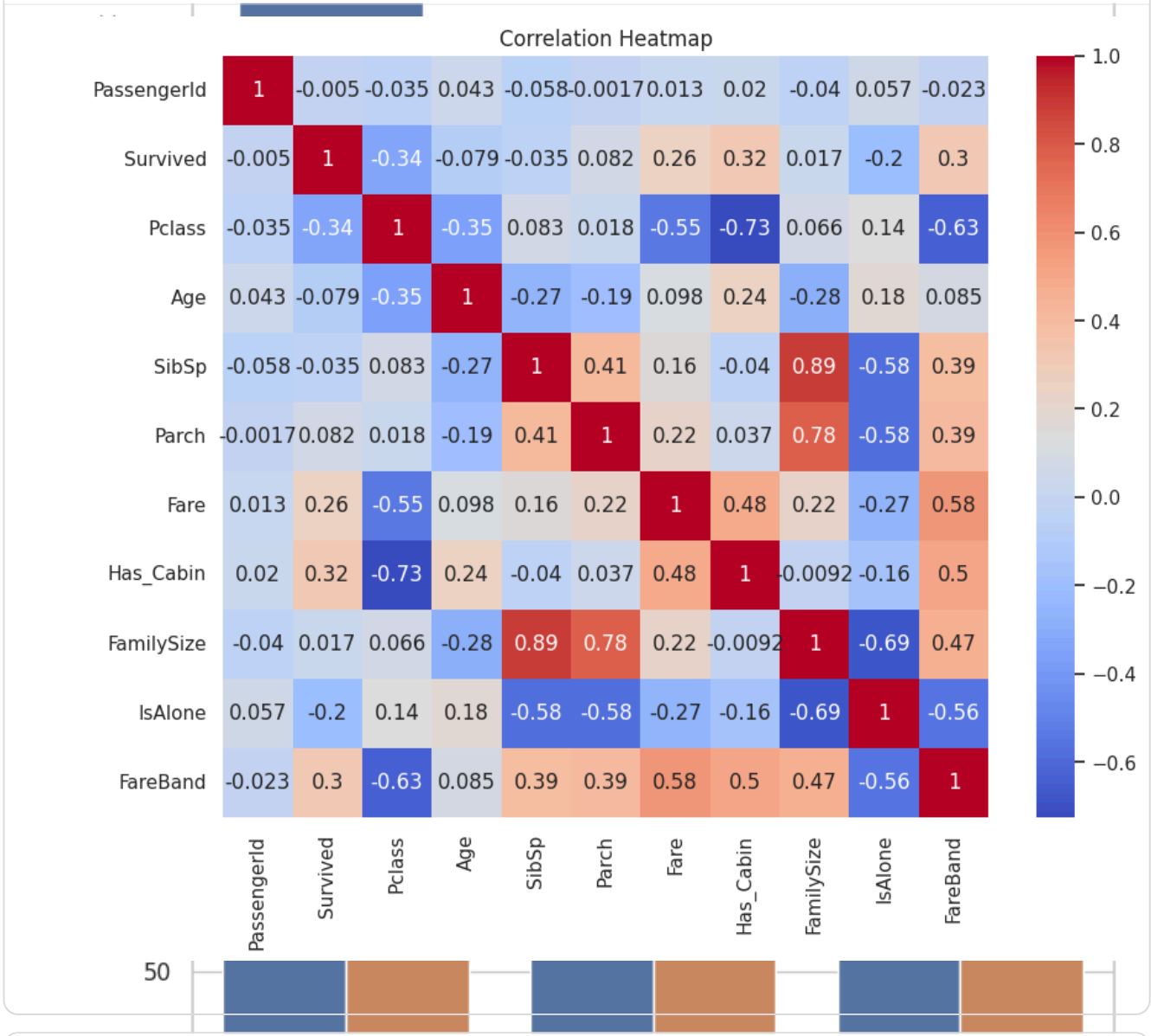



200
100
0
male
female
Sex


Pclass Distribution
500
400
300
200
100
0
count
1
2
3
Pclass

Survival by Sex

```
plt.figure(figsize=(10,8))
# Drop non-numeric columns before calculating correlation
numeric_df = df.drop(['Name', 'Sex', 'Ticket', 'Cabin', 'Embarked', 'Title'], axis
sns.heatmap(numeric_df.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```




```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

X = df[['Pclass','Sex','Age','Fare','IsAlone','Has_Cabin']].copy()
X = pd.get_dummies(X, drop_first=True)
y = df['Survived']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_st

model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
preds = model.predict(X_test)
```