Load and Preprocess the Data

```
import pandas as pd
import numpy as np
```

```
#load the dataset
df=pd.read_csv("train.csv")
```

```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
				Futrelle, Mrs. Jacques Heath								

Next steps: Generate code with df View recommended plots New interactive sheet

```
# data information
df.info()
```

```
<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
```

```
# check the null/missing value is present or not
df.isna().sum()
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2

dtype: int64

```
# replace null value with median in age column
df['Age'].fillna(df['Age'].median(),inplace=True)
```

```
# remove/drop unnessecery column
df.drop(columns=['Cabin'],inplace=True)


#  Fill missing Embarked with mode
df['Embarked'].fillna(df['Embarked'].mode()[0],inplace=True)

# One-hot encode 'Embarked'
df = pd.get_dummies(df, columns=['Embarked'], drop_first=True)


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


df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked_Q	Embarked_S
0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	False	True
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0	1	0	PC 17599	71.2833	False	False
2	3	1	3	Heikkinen, Miss. Laina	1	26.0	0	0	STON/O2. 3101282	7.9250	False	True

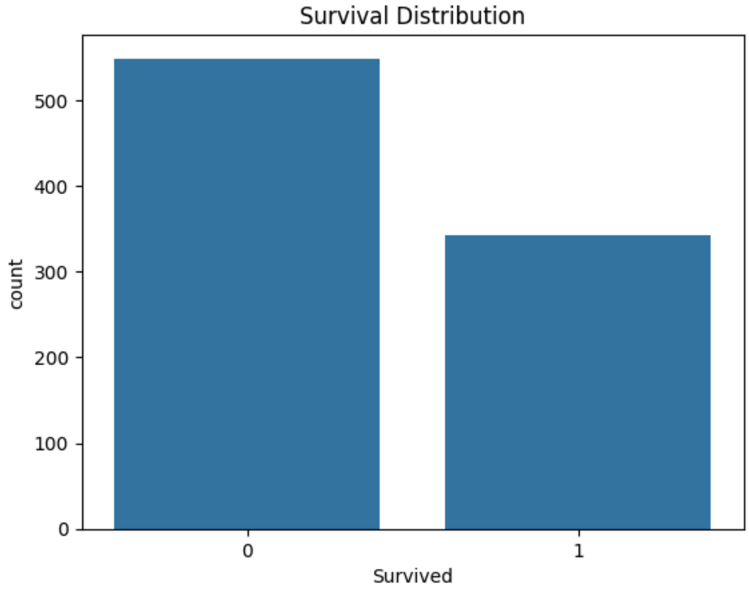
Next steps: Generate code with df View recommended plots New interactive sheet

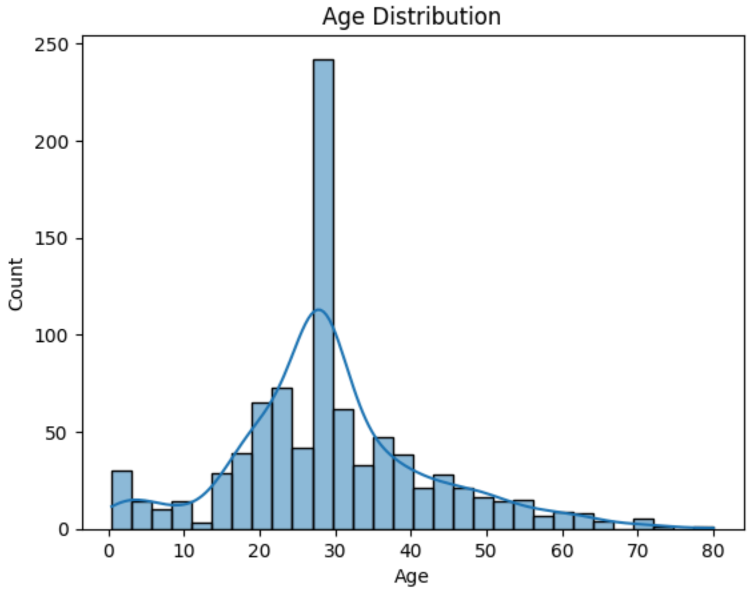
```
df.describe()
```

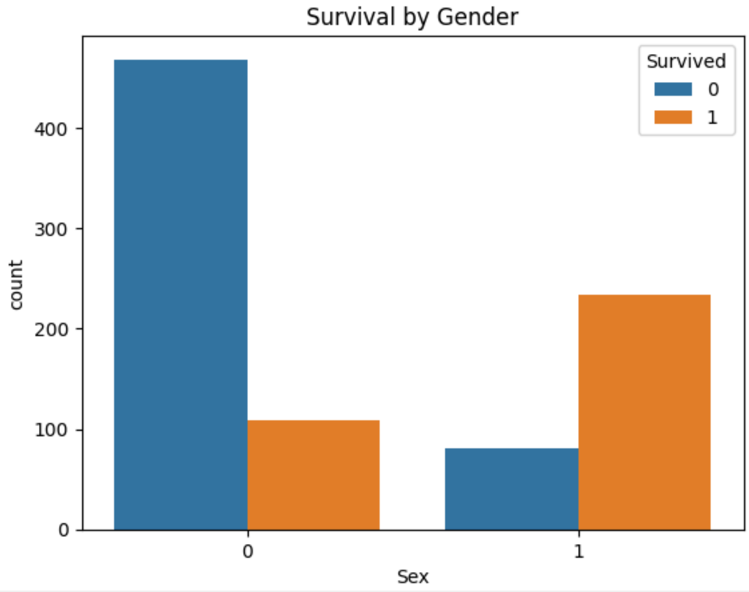
	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	0.352413	29.361582	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	0.477990	13.019697	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	0.000000	22.000000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	0.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	1.000000	35.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	1.000000	80.000000	8.000000	6.000000	512.329200

Ploting / Visualisation

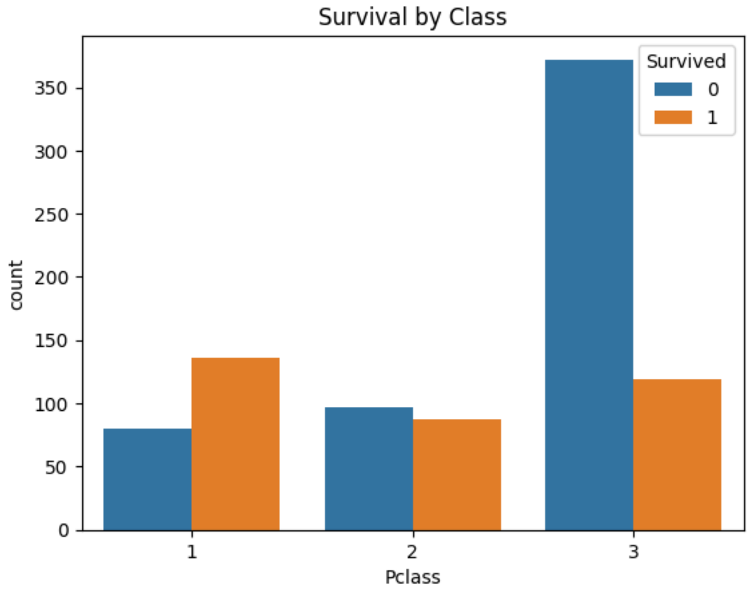
```
# import library
import seaborn as sns
import matplotlib.pyplot as plt



# Survival rate
sns.countplot(x='Survived', data=df)
plt.title('Survival Distribution')
plt.show()

# Age distribution
sns.histplot(df['Age'], bins=30, kde=True)
plt.title('Age Distribution')
plt.show()

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

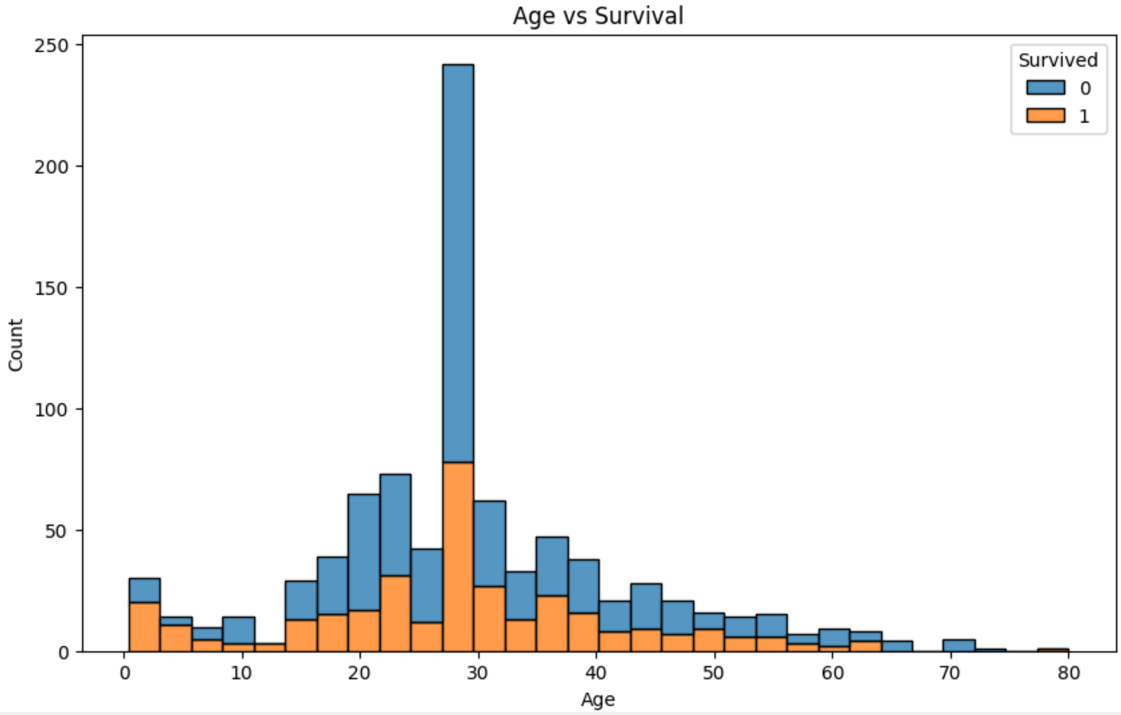

Survival Distribution
count
0
100
200
300
400
500
0
1
Survived


Age Distribution
Count
0
50
100
150
200
250
0
10
20
30
40
50
60
70
80
Age

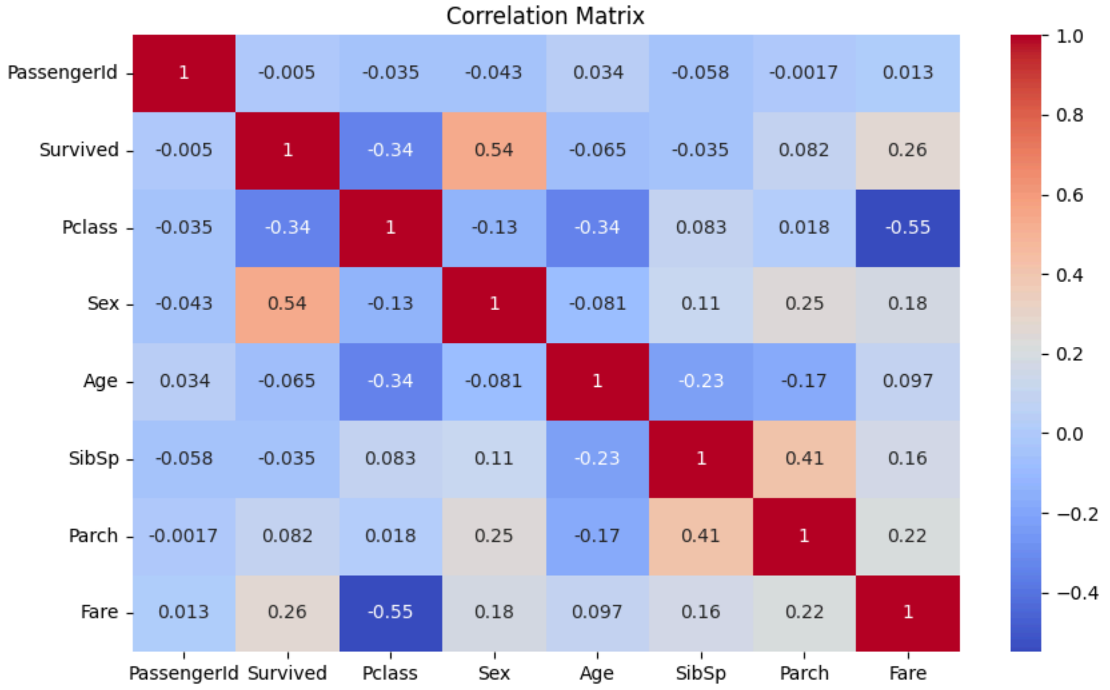

Survival by Gender
count
0
100
200
300
400
Survived
0
1
0
1
Sex

```
# Survival by class
sns.countplot(x='Pclass', hue='Survived', data=df)
plt.title('Survival by Class')
plt.show()

# Age vs Survival
plt.figure(figsize=(10, 6))
sns.histplot(data=df, x='Age', hue='Survived', multiple='stack')
plt.title('Age vs Survival')
plt.show()
```



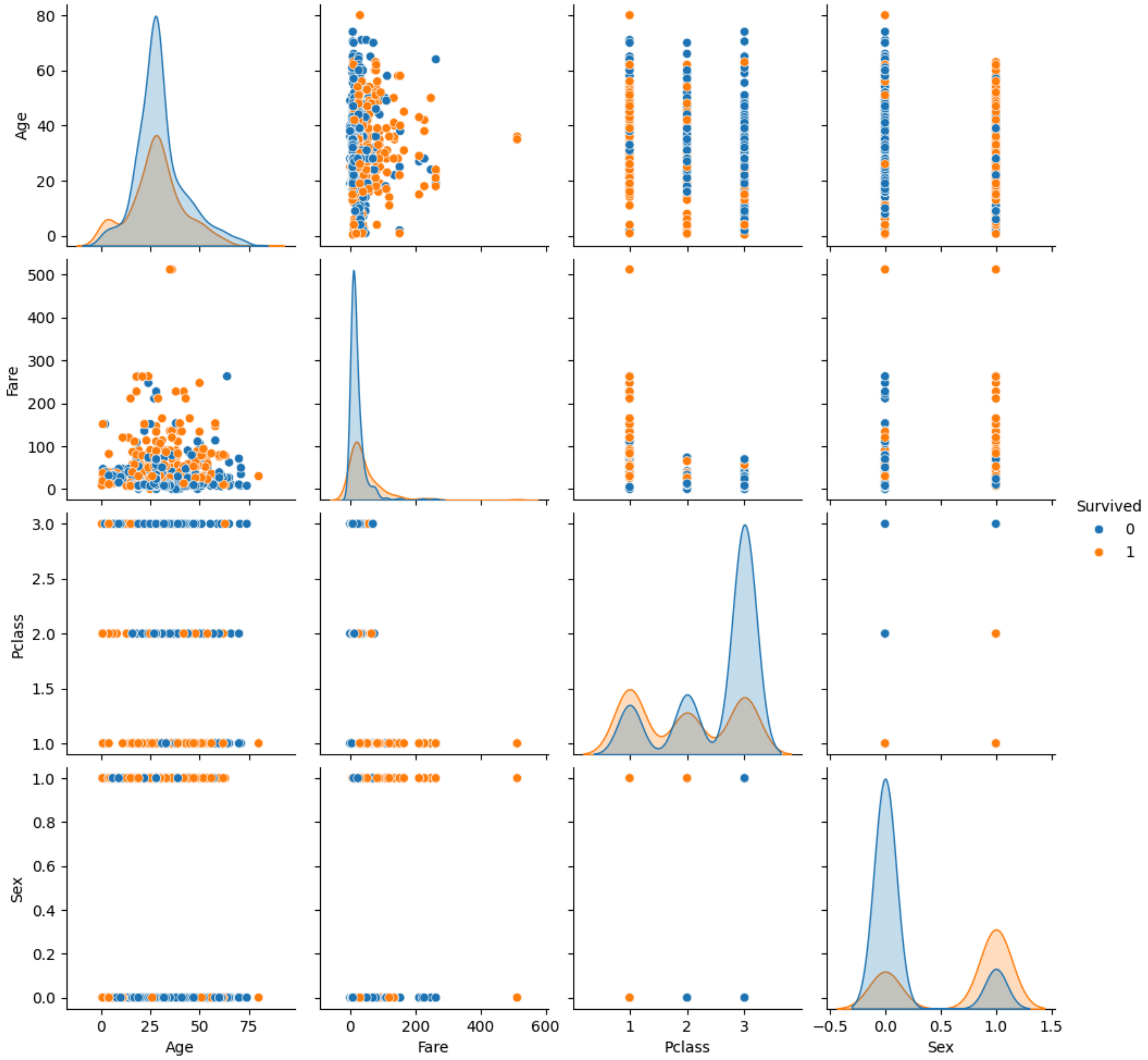


```
# Drop non-numeric columns
numeric_df = df.select_dtypes(include=[np.number])

# Correlation matrix
plt.figure(figsize=(10, 6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()

# Pairplot
sns.pairplot(df, hue='Survived', vars=['Age', 'Fare', 'Pclass', 'Sex'])
plt.show()
```



Correlation Matrix

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
PassengerId	1	-0.005	-0.035	-0.043	0.034	-0.058	-0.0017	0.013
Survived	-0.005	1	-0.34	0.54	-0.065	-0.035	0.082	0.26
Pclass	-0.035	-0.34	1	-0.13	-0.34	0.083	0.018	-0.55
Sex	-0.043	0.54	-0.13	1	-0.081	0.11	0.25	0.18
Age	0.034	-0.065	-0.34	-0.081	1	-0.23	-0.17	0.097
SibSp	-0.058	-0.035	0.083	0.11	-0.23	1	0.41	0.16
Parch	-0.0017	0.082	0.018	0.25	-0.17	0.41	1	0.22
Fare	0.013	0.26	-0.55	0.18	0.097	0.16	0.22	1



Start coding or generate with AI.