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

```
from google.colab import files
uploaded = files.upload()
```

Choose Files Titanic-Dataset.csv

- **Titanic-Dataset.csv**(text/csv) - 61194 bytes, last modified: 9/29/2025 - 100% done

```
Saving Titanic-Dataset.csv to Titanic-Dataset.csv
```

```
import pandas as pd

df = pd.read_csv("Titanic-Dataset.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

```
df.info()
df.describe()
```

```
<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
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.0
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.2
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.6
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.0
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.9
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.4
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.0
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.3

```
df.isnull().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

```
df["Age"].fillna(df["Age"].median(),inplace=True)
```

```
/usr/local/lib/python3.12/dist-packages/numpy/lib/_nanfunctions_impl.py:1231: Runti
  return np.nanmean(a, axis, out=out, keepdims=keepdims)
/tmp/ipython-input-1164800110.py:1: FutureWarning: A value is trying to be set on a
The behavior will change in pandas 3.0. This inplace method will never work because

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method


  df["Age"].fillna(df["Age"].median(),inplace=True)
/tmp/ipython-input-1164800110.py:1: FutureWarning: Downcasting object dtype arrays
  df["Age"].fillna(df["Age"].median(),inplace=True)
```

```
df.drop("Cabin", axis=1, inplace=True)
df["Embarked"].fillna(df["Embarked"].mode()[0], inplace=True)
```

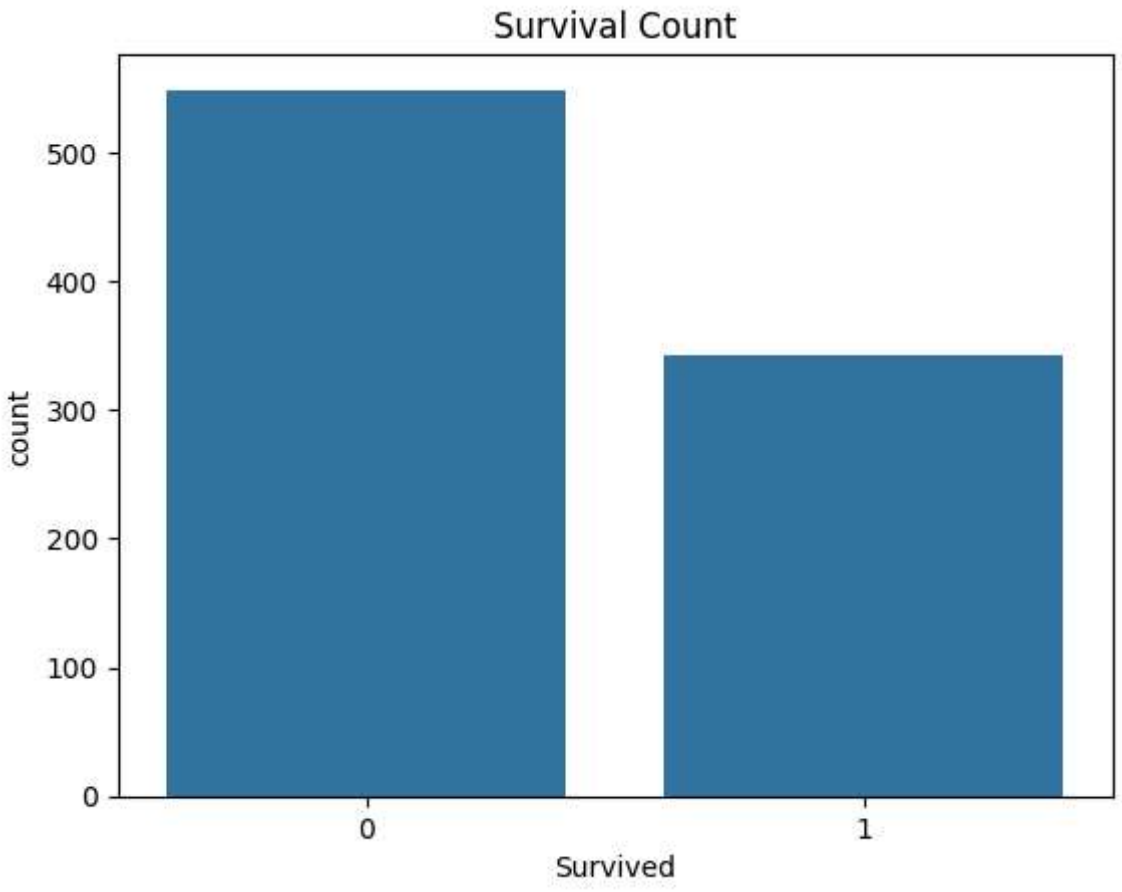
```
/tmp/ipython-input-411768894.py:2: FutureWarning: A value is trying to be set on a
The behavior will change in pandas 3.0. This inplace method will never work because

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method


  df["Embarked"].fillna(df["Embarked"].mode()[0], inplace=True)
```

```
df.isnull().sum()
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	891
SibSp	0
Parch	0
Ticket	0
Fare	0
Embarked	0

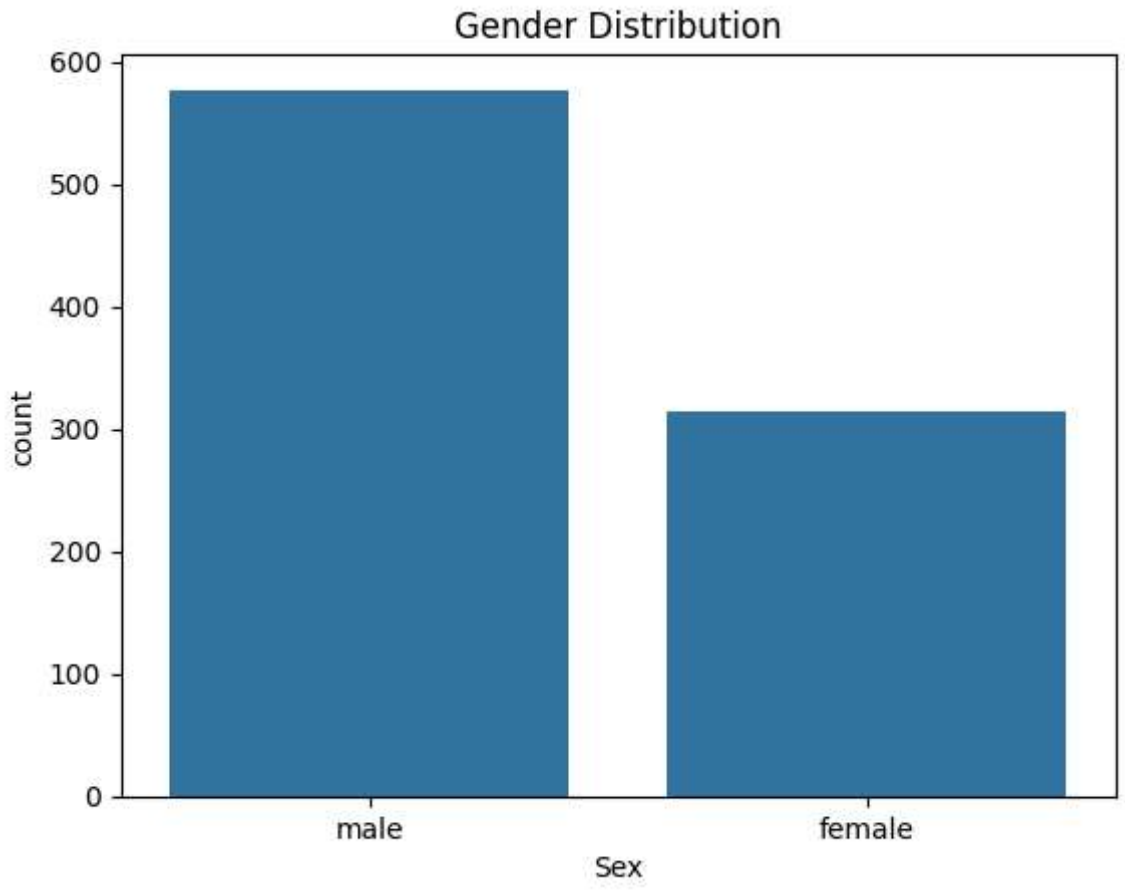
dtype: int64

```
sns.countplot(x="Survived", data=df)
plt.title("Survival Count")
plt.show()
```



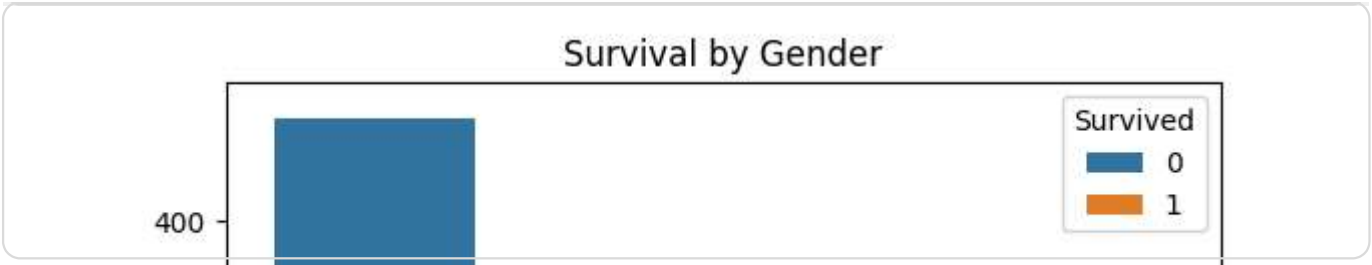
More passengers died 0 than survived 1.

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



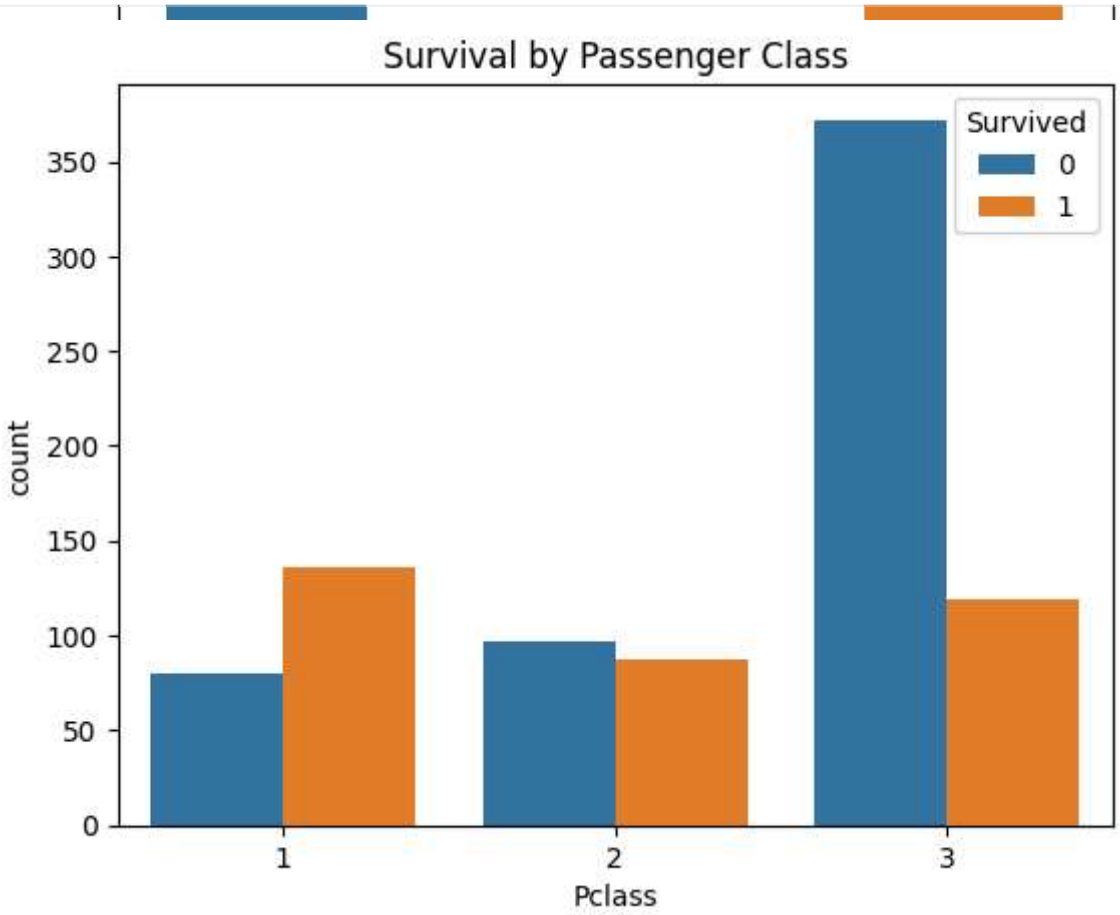
More males were on board than females.

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



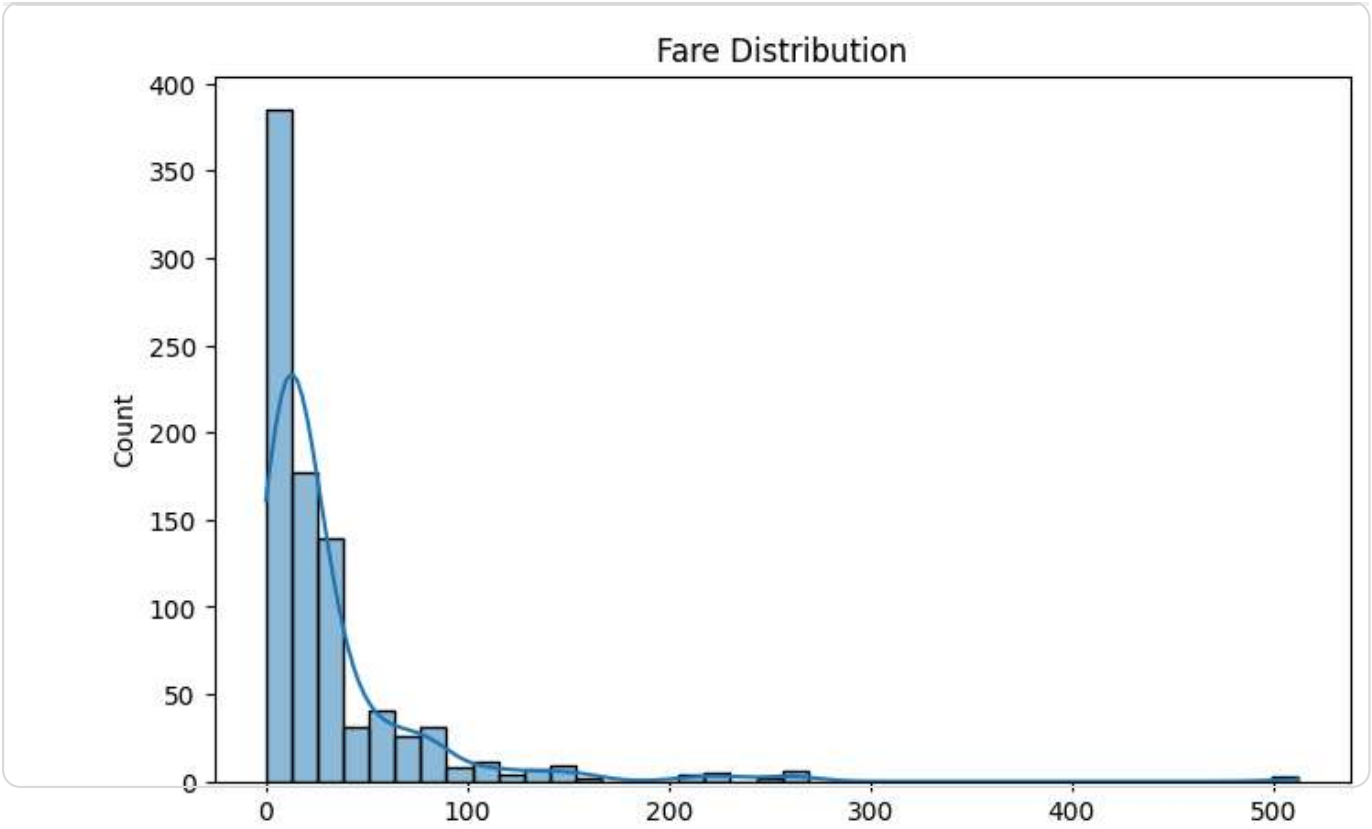
Females had much higher survival rates than males.

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



Passengers in 1st class survived more than 3rd class.

```
plt.figure(figsize=(8,5))
sns.histplot(df["Fare"], bins=40, kde=True)
plt.title("Fare Distribution")
plt.show()
```



Most fares were low <50, but a few passengers paid very high fares outliers

```
plt.figure(figsize=(8,6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```