1 - Importing the Dependencies

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

2 - Data Collection & Processing

```
titanic_data = pd.read_csv('/content/train.csv')
```

```
titanic_data.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

Next steps: Generate code with titanic_data | New interactive sheet

```
titanic_data.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
```

```
titanic_data.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

3 - Handling the Missing values

```
titanic_data = titanic_data.drop(columns='Cabin', axis=1)
```

```
titanic_data['Age'].fillna(titanic_data['Age'].mean(), inplace=True)
```

```
/tmp/ipython-input-3516126430.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chain
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are set

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


  titanic_data['Age'].fillna(titanic_data['Age'].mean(), inplace=True)
```

```
print(titanic_data['Embarked'].mode())
```

```
0    S
Name: Embarked, dtype: object
```

```
print(titanic_data['Embarked'].mode()[0])
```

```
S
```

```
titanic_data['Embarked'].fillna(titanic_data['Embarked'].mode()[0], inplace=True)
```

```
/tmp/ipython-input-3993763136.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chain
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are set

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


  titanic_data['Embarked'].fillna(titanic_data['Embarked'].mode()[0], inplace=True)
```

```
titanic_data.isnull().sum()
```

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

dtype: int64

4 - Data Analysis

```
titanic_data.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	13.002015	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	22.000000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	29.699118	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	35.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
titanic_data['Survived'].value_counts()
```

Survived	count
0	549
1	342

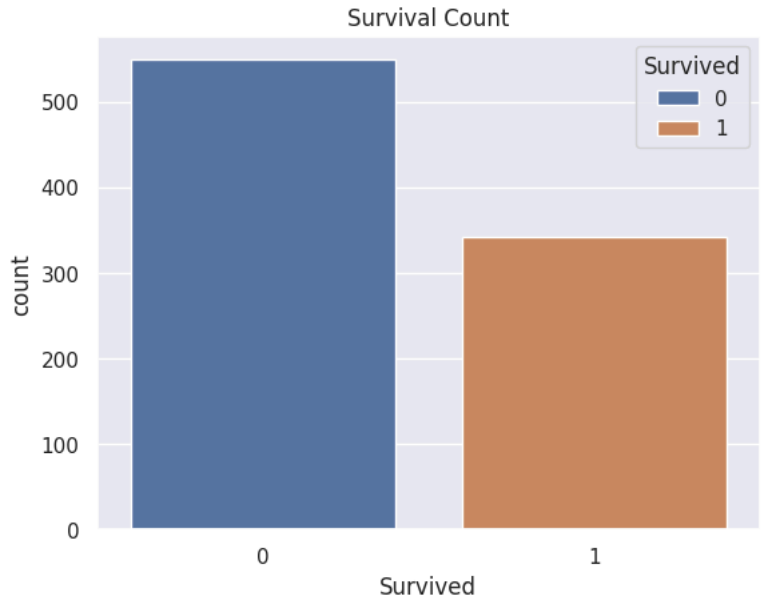
dtype: int64

5 - Data Visualization

```
sns.set()
```

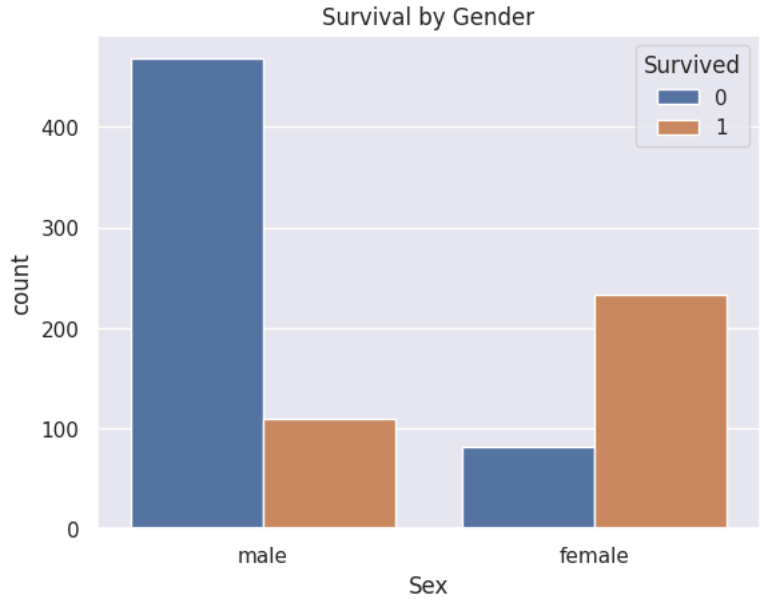
```
sns.countplot(x= 'Survived',hue='Survived', data=titanic_data)
plt.title('Survival Count')
```

```
Text(0.5, 1.0, 'Survival Count')
```



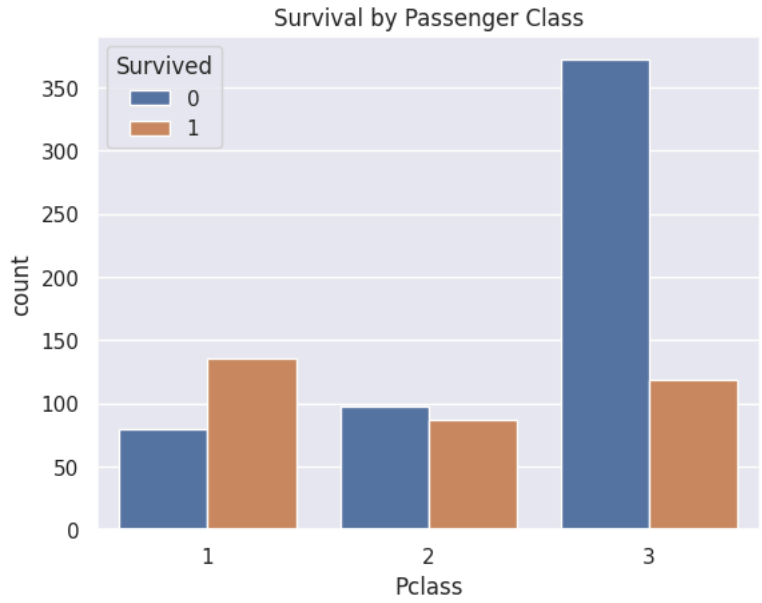

```
sns.countplot(x = 'Sex', hue='Survived', data=titanic_data)
plt.title('Survival by Gender')
```

```
Text(0.5, 1.0, 'Survival by Gender')
```




```
sns.countplot(x='Pclass', hue='Survived', data=titanic_data)
plt.title('Survival by Passenger Class')
```

```
Text(0.5, 1.0, 'Survival by Passenger Class')
```

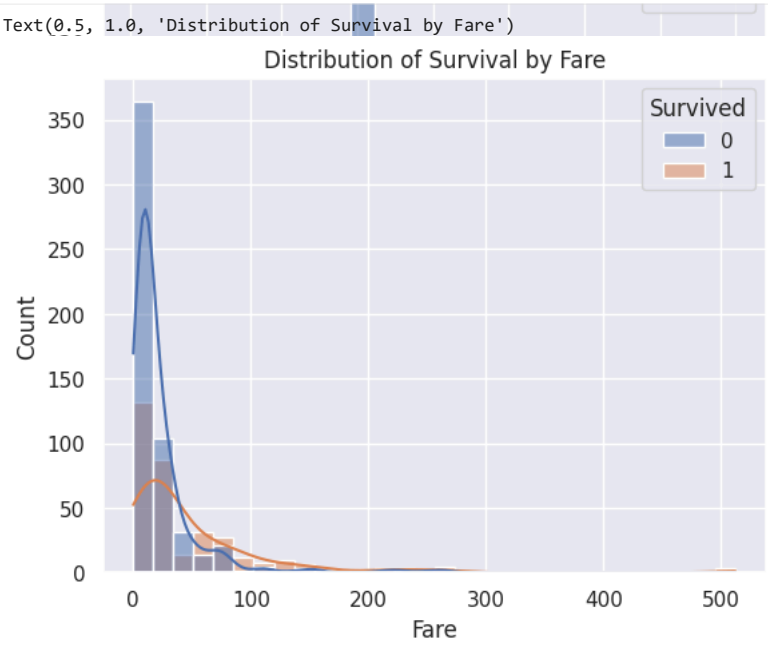



```
sns.histplot(x='Age', bins=25, kde= True, hue='Survived', data=titanic_data)
plt.title('Age Distribution of Passengers')
```

```
Text(0.5, 1.0, 'Age Distribution of Passengers')
```


```
sns.histplot(x='Fare', bins=30, kde= True, hue='Survived', data=titanic_data)
plt.title('Distribution of Survival by Fare')
```

```
Text(0.5, 1.0, 'Distribution of Survival by Fare')
```




```
numeric_titanic_data = titanic_data.drop(columns=['Name', 'Sex', 'Ticket', 'Embarked'])
sns.heatmap(numeric_titanic_data.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
```

```
Text(0.5, 1.0, 'Correlation Heatmap')
```