```
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
```

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

```
titanic_data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence	female	38.0	1	0	PC 17599	71.2

```
titanic_data.shape
```

```
(891, 12)
```

```
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()
```

```
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
```

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

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

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

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

```
Start coding or generate with AI.
```

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

```
'S'
```

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

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

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

```
titanic_data.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fa
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000(
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204
std	257.353842	0.486592	0.836071	13.002015	1.102743	0.806057	49.693
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000(
25%	223.500000	0.000000	2.000000	22.000000	0.000000	0.000000	7.910
50%	446.000000	0.000000	3.000000	29.699118	0.000000	0.000000	14.454
75%	668.500000	1.000000	3.000000	35.000000	1.000000	0.000000	31.000(
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329

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

```
Survived
0    549
1    342
Name: count, dtype: int64
```

```
sns.set()
```

```
sns.countplot(titanic_data['Survived'])
```

```
<Axes: ylabel='count'>
```

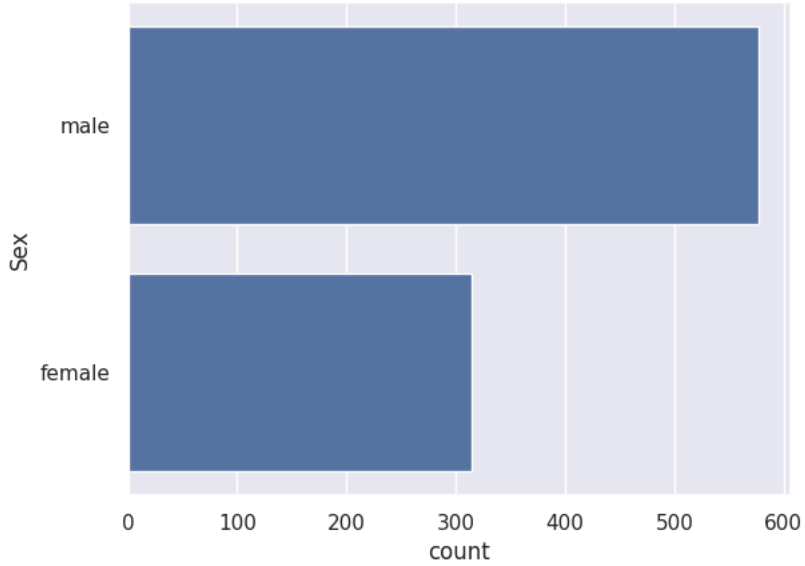
Data Collection/loading and processing

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

```
Sex
male      577
female    314
Name: count, dtype: int64
```

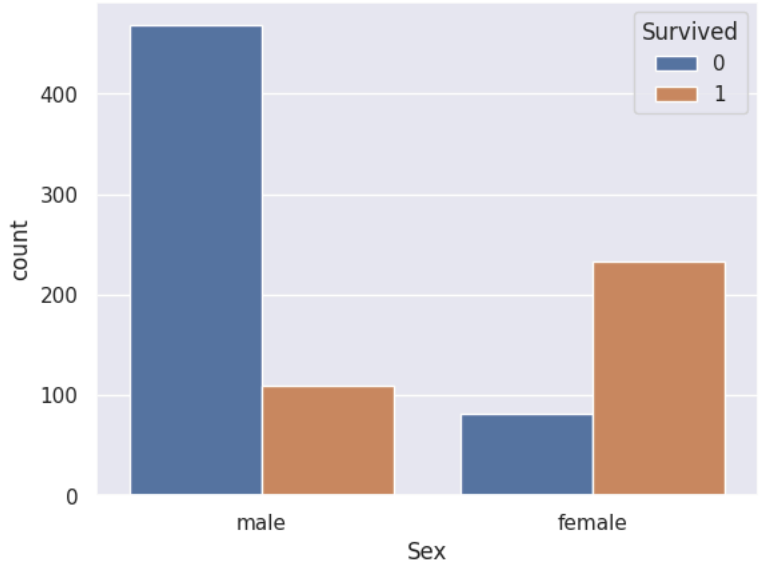
```
sns.countplot(titanic_data['Sex'])
```

```
<Axes: xlabel='count', ylabel='Sex'>
```



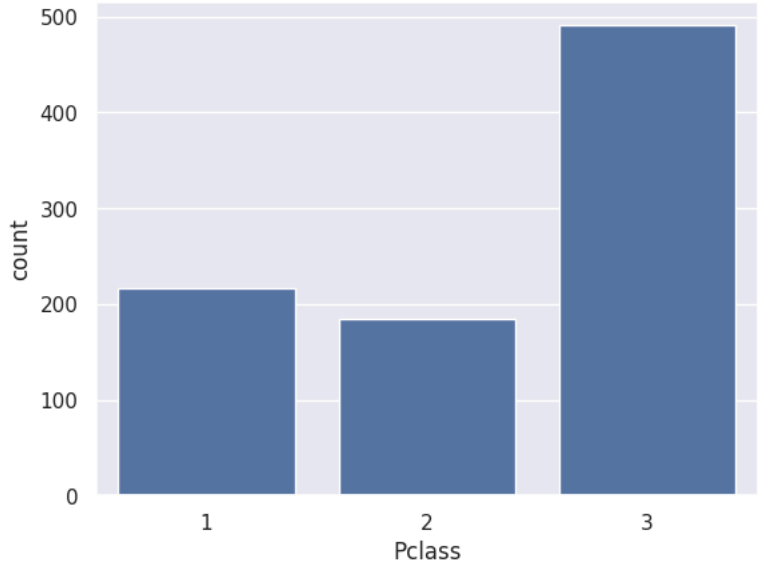

```
sns.countplot(x='Sex', hue='Survived', data=titanic_data)
```

```
<Axes: xlabel='Sex', ylabel='count'>
```



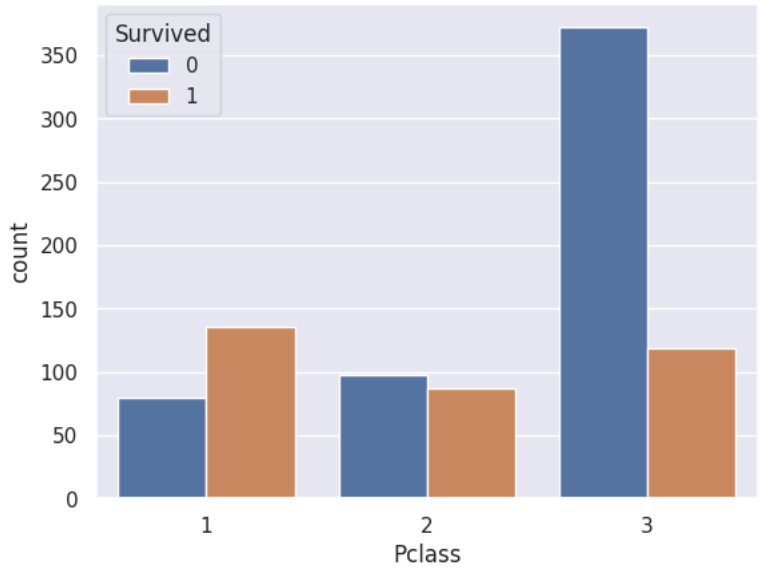

```
sns.countplot(x= 'Pclass', data=titanic_data)
```

```
<Axes: xlabel='Pclass', ylabel='count'>
```




```
sns.countplot(x='Pclass', hue='Survived', data=titanic_data)
```

```
<Axes: xlabel='Pclass', ylabel='count'>
```




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

```
Sex
male      577
female    314
Name: count, dtype: int64
```

```
Start coding or generate with AI.
```

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

```
Embarked
S    646
C    168
Q     77
Name: count, dtype: int64
```

```
titanic_data.replace({'Sex':{'male':0,'female':1}, 'Embarked':{'S':0,'C':1,'Q':2}})
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	0	22.000000	1	0	A/5 21171	7.2500	0
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.000000	1	0	PC 17599	71.2833	1
2	3	1	3	Heikkinen, Miss. Laina	1	26.000000	0	0	STON/O2. 3101282	7.9250	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.000000	1	0	113803	53.1000	0
4	5	0	3	Allen, Mr. William Henry	0	35.000000	0	0	373450	8.0500	0
...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	0	27.000000	0	0	211536	13.0000	0
887	888	1	1	Graham, Miss. Margaret Edith	1	19.000000	0	0	112053	30.0000	0
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	1	29.699118	1	2	W./C. 6607	23.4500	0
889	890	1	1	Behr, Mr. Karl Howell	0	26.000000	0	0	111369	30.0000	1

Double-click (or enter) to edit

```
X = titanic_data.drop(columns = ['PassengerId','Name','Ticket','Survived'],axis=1)
Y = titanic_data['Survived']
```

```
print(X)
```

```
     Pclass     Sex        Age  SibSp  Parch     Fare Embarked
0         3    male  22.000000      1      0   7.2500        S
1         1  female  38.000000      1      0  71.2833        C
2         3  female  26.000000      0      0   7.9250        S
3         1  female  35.000000      1      0  53.1000        S
4         3    male  35.000000      0      0   8.0500        S
..      ...     ...        ...    ...    ...      ...      ...
886       2    male  27.000000      0      0  13.0000        S
887       1  female  19.000000      0      0  30.0000        S
888       3  female  29.699118      1      2  23.4500        S
889       1    male  26.000000      0      0  30.0000        C
890       3    male  32.000000      0      0   7.7500        Q

[891 rows x 7 columns]
```

```
print(Y)
```

```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.2, random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(891, 7) (712, 7) (179, 7)
```

```
model = LogisticRegression()
```

```
model.fit(X_train,Y_train)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-103-ffa49499a3bf> in <cell line: 1>()
----> 1 model.fit(X_train,Y_train)

                                 5 frames
/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py in __array__(self, dtype)
   1996     def __array__(self, dtype: npt.DTypeLike | None = None) -> np.ndarray:
   1997         values = self._values
-> 1998         arr = np.asarray(values, dtype=dtype)
   1999         if (
   2000             astype_is_view(values.dtype, arr.dtype)

ValueError: could not convert string to float: 'male'
```