In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [75]:
```python
df= pd.read_csv("tested.csv")
```

In [76]:
```python
df.head(10)
```

Out[76]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875
5	897	0	3	Svensson, Mr. Johan Cervin	male	14.0	0	0	7538	9.2250
6	898	1	3	Connolly, Miss. Kate	female	30.0	0	0	330972	7.6292
7	899	0	2	Caldwell, Mr. Albert Francis	male	26.0	1	1	248738	29.0000
8	900	1	3	Abrahim, Mrs. Joseph (Sophie Halaut Easu)	female	18.0	0	0	2657	7.2292
9	901	0	3	Davies, Mr. John Samuel	male	21.0	2	0	A/4 48871	24.1500

In [78]:
```python
df.shape
```

Out[78]: (418, 12)

In [79]: ▶| `df.describe()`

Out[79]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	0.363636	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.481622	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	0.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	0.000000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	0.000000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	1.000000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	1.000000	3.000000	76.000000	8.000000	9.000000	512.329200

In [81]: ▶| `df['Survived'].value_counts()`

Out[81]:
```
0    266
1    152
Name: Survived, dtype: int64
```

In [84]: ▶| `sns.countplot(x=df['Survived'],hue=df['Pclass'])`

Out[84]: `<Axes: xlabel='Survived', ylabel='count'>`
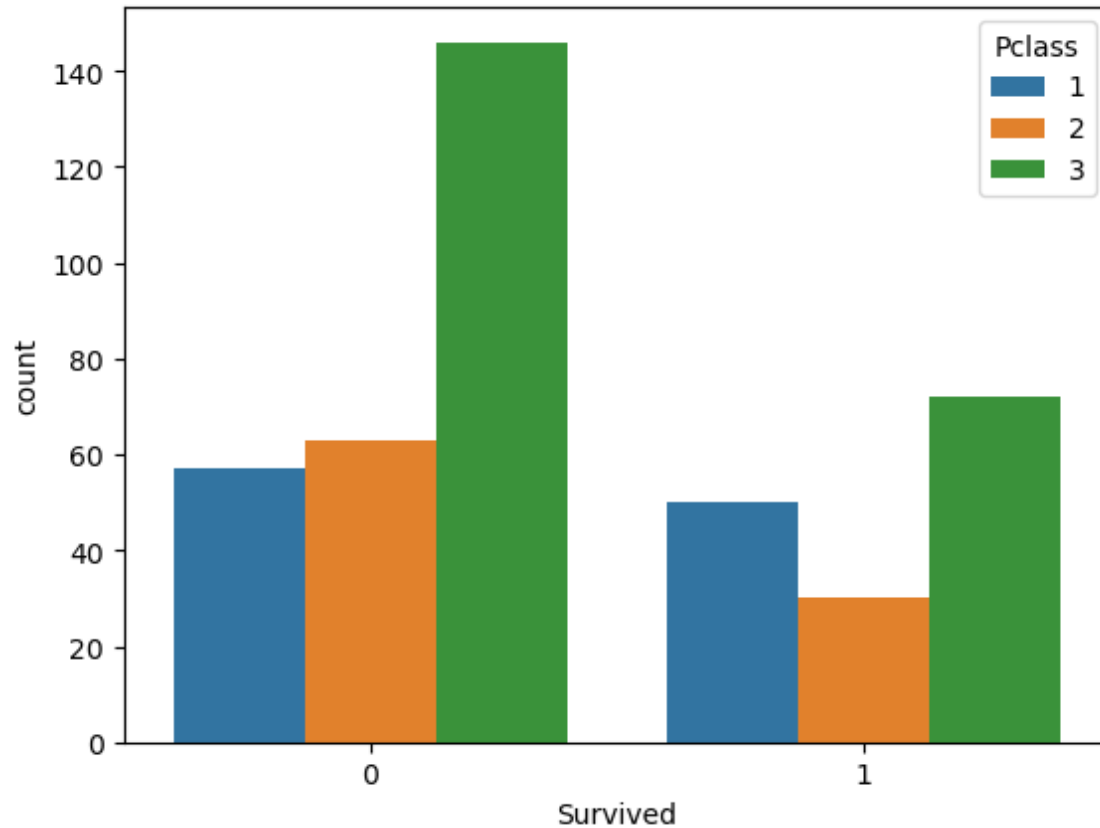
In [85]: ▶| `df['Sex']`
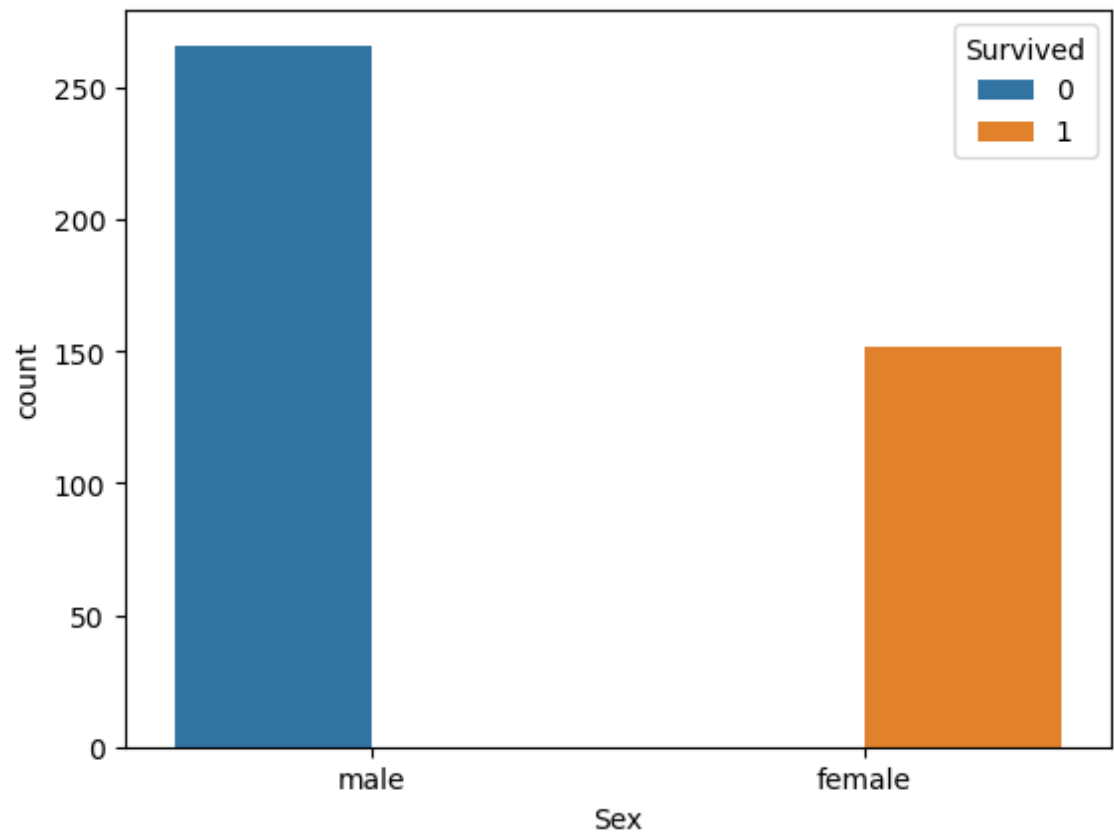
Out[85]:
```
0         male
1       female
2         male
3         male
4       female
         ...
413       male
414     female
415       male
416       male
417       male
Name: Sex, Length: 418, dtype: object
```

In [86]: ▶| `sns.countplot(x=df['Sex'],hue=df['Survived'])`

Out[86]: `<Axes: xlabel='Sex', ylabel='count'>`

In [87]: ▶| `df.groupby('Sex')[['Survived']].mean()`

Out[87]:

	Survived
Sex	
female	1.0
male	0.0

In [88]: ▶| `df['Sex'].unique()`

Out[88]: `array(['male', 'female'], dtype=object)`

In [89]: ▶|
```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
df['Sex'] = labelencoder.fit_transform(df['Sex'])
df.head()
```

Out[89]:

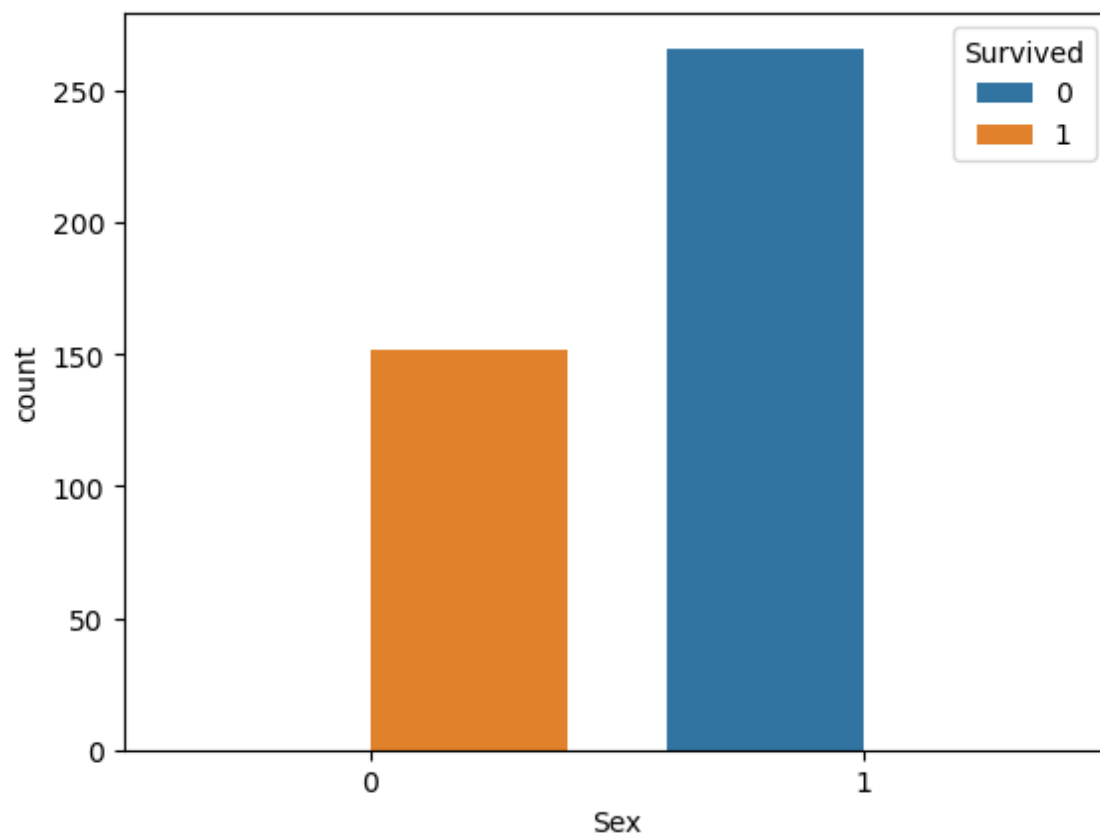|	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
0	892	0	3	Kelly, Mr. James	1	34.5	0	0	330911	7.8292	N
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	0	47.0	1	0	363272	7.0000	N
2	894	0	2	Myles, Mr. Thomas Francis	1	62.0	0	0	240276	9.6875	N
3	895	0	3	Wirz, Mr. Albert	1	27.0	0	0	315154	8.6625	N
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	0	22.0	1	1	3101298	12.2875	N

In [90]: ▶| `df['Sex'] , df['Survived']`

Out[90]: (0   1
          1   0
          2   1
          3   1
          4   0
                  ..
          413   1
          414   0
          415   1
          416   1
          417   1
          Name: Sex, Length: 418, dtype: int32,
          0   0
          1   1
          2   0
          3   0
          4   1
                  ..
          413   0
          414   1
          415   0
          416   0
          417   0
          Name: Survived, Length: 418, dtype: int64)

In [91]: ▶| `sns.countplot(x=df['Sex'] ,hue= df['Survived'])`

Out[91]: <Axes: xlabel='Sex', ylabel='count'>

In [92]: ▶| `df.isna().sum()`

Out[92]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [93]: ▶| `df = df.drop(['Age'] , axis = 1)`

In [94]: ▶|
```python
df_final = df
df_final.head(10)
```

Out[94]:

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin
0	892	0	3	Kelly, Mr. James	1	0	0	330911	7.8292	NaN
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	0	1	0	363272	7.0000	NaN
2	894	0	2	Myles, Mr. Thomas Francis	1	0	0	240276	9.6875	NaN
3	895	0	3	Wirz, Mr. Albert	1	0	0	315154	8.6625	NaN
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	0	1	1	3101298	12.2875	NaN
5	897	0	3	Svensson, Mr. Johan Cervin	1	0	0	7538	9.2250	NaN
6	898	1	3	Connolly, Miss. Kate	0	0	0	330972	7.6292	NaN
7	899	0	2	Caldwell, Mr. Albert Francis	1	1	1	248738	29.0000	NaN
8	900	1	3	Abrahim, Mrs. Joseph (Sophie Halaut Easu)	0	0	0	2657	7.2292	NaN
9	901	0	3	Davies, Mr. John Samuel	1	2	0	A/4 48871	24.1500	NaN

In [95]: ▶|
```python
X = df[['Pclass' ,'Sex']]
Y = df['Survived']
```

In [96]: ▶|
```python
from sklearn.model_selection import train_test_split
X_train , X_test ,Y_train , Y_test = train_test_split(X , Y , test_size =0.
```

In [97]: ▶|
```python
from sklearn.linear_model import LogisticRegression
log = LogisticRegression(random_state = 0)
log.fit(X_train, Y_train)
```

Out[97]: LogisticRegression(random_state=0)

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [98]: ▶|
```python
pred = print(log.predict(X_test))
```

```
[0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 0 0 0 1 0 0 0 1 1 1 1 1 0
0
 1 1 1 1 0 1 1 0 1 0 0 0 0 0 1 1 0 0 1 0 1 0 0 0 1 1 0 0 1 1 1 1 0 0 1 1
1
 1 0 0 1 0 1 0 1 0 0]
```

In [99]: ▶|
```python
print(Y_test)
```

```
360    0
170    0
224    1
358    0
309    1
      ..
100    1
7      0
22     1
68     0
328    0
Name: Survived, Length: 84, dtype: int64
```

In [102]: ▶|
```python
import warnings
warnings.filterwarnings("ignore")
res = log.predict([[2,0]])
if(res==0):
    print("Not survived")
else:
        print("survived")
```

```
survived
```

In [ ]: ▶|