

```
In [2]: import pandas as pd
import numpy as np
import matplotlib as plt

df = pd.read_csv("Downloads/tested.csv")
```

```
In [4]: df
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
...
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

418 rows × 12 columns

```
In [6]: df.describe()
```

Out[6]:

	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 [8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  -
0   PassengerId  418 non-null    int64
1   Survived     418 non-null    int64
2   Pclass       418 non-null    int64
3   Name         418 non-null    object
4   Sex          418 non-null    object
5   Age          332 non-null    float64
6   SibSp        418 non-null    int64
7   Parch        418 non-null    int64
8   Ticket       418 non-null    object
9   Fare         417 non-null    float64
10  Cabin        91 non-null     object
11  Embarked     418 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

```
In [10]: df.shape
```

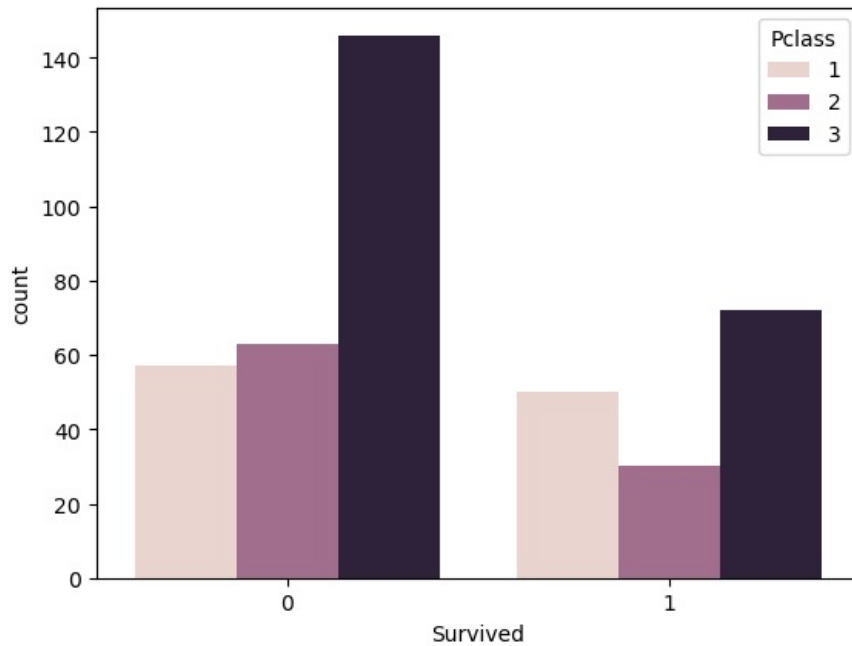
Out[10]: (418, 12)

```
In [12]: df['Survived'].value_counts()
```

```
Out[12]: Survived  
0      266  
1      152  
Name: count, dtype: int64
```

```
In [14]: #visualizing the count of survivals wrt to Pclass  
import seaborn as sns  
sns.countplot(x=df['Survived'], hue=df['Pclass'])
```

```
Out[14]: <Axes: xlabel='Survived', ylabel='count'>
```

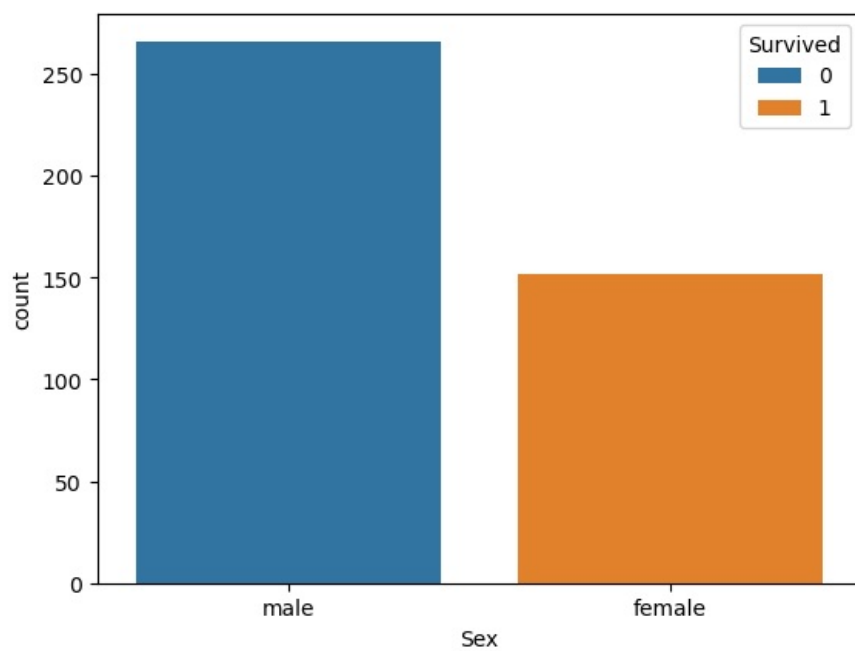


```
In [16]: df["Sex"]
```

```
Out[16]: 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 [22]: #visualizing count of survival wrt gender  
sns.countplot(x=df['Sex'], hue=df['Survived'])
```

```
Out[22]: <Axes: xlabel='Sex', ylabel='count'>
```



```
In [24]: #Survival rate by sex  
df.groupby('Sex')[['Survived']].mean()
```

```
Out[24]:
```

Survived	
Sex	
female	1.0
male	0.0

```
In [26]: df['Sex'].unique()
```

```
Out[26]: array(['male', 'female'], dtype=object)
```

```
In [28]: from sklearn.preprocessing import LabelEncoder  
labelencoder = LabelEncoder()  
df['Sex'] = labelencoder.fit_transform(df['Sex'])  
df.head()
```

Out[28]:

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

In [30]:

```
df['Sex'], df['Survived']
```

Out[30]:

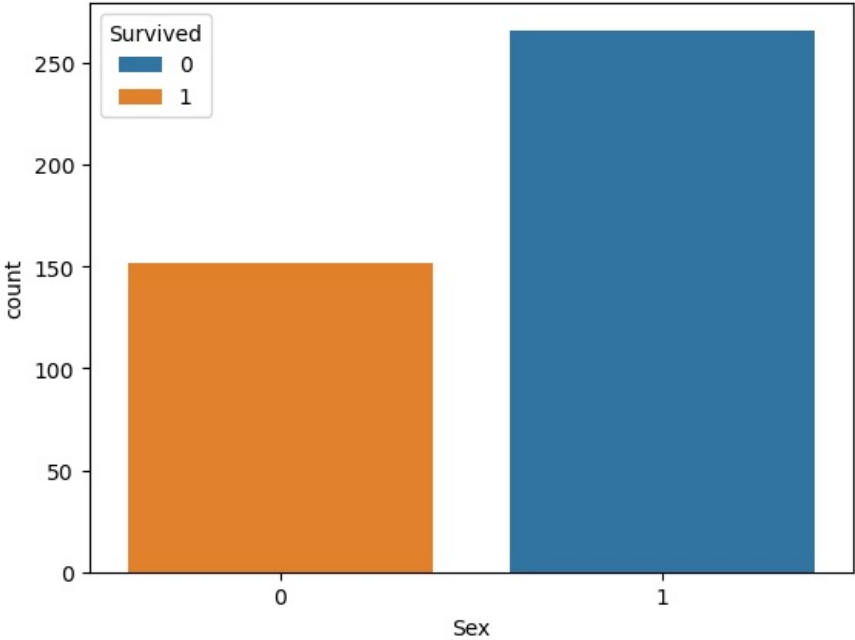
```
(0      1
1      0
2      1
3      1
4      0
..
413    1
414    0
415    1
416    1
417    1
Name: Sex, Length: 418, dtype: int32,
0      0
1      1
2      0
3      0
4      1
..
413    0
414    1
415    0
416    0
417    0
Name: Survived, Length: 418, dtype: int64)
```

In [32]:

```
sns.countplot(x=df['Sex'], hue=df['Survived'])
```

Out[32]:

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

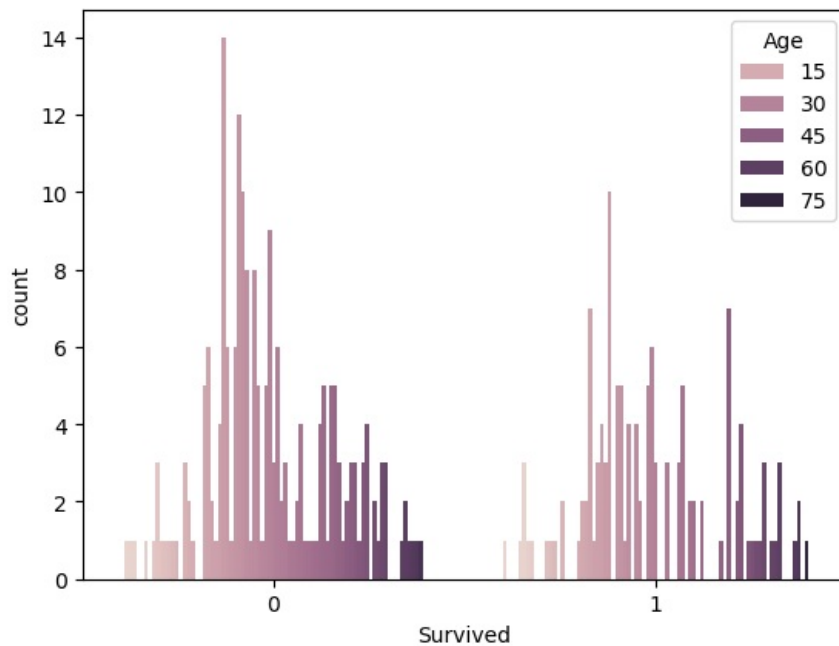


In [36]:

```
sns.countplot(x=df['Survived'], hue=df['Age'])
```

Out[36]:

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



```
In [38]: df.isna().sum()
```

```
Out[38]: 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 [40]: #Dropping non required Age column
df = df.drop(['Age'], axis=1)
```

```
In [42]: df_final = df
df_final.head(10)
```

```
Out[42]:
```

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

```
In [44]: X = df[['Pclass', 'Sex']]
Y = df['Survived']
```

```
In [46]: from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 0)
```

```
In [50]: from sklearn.linear_model import LogisticRegression

log = LogisticRegression(random_state = 0)
log.fit(X_train, Y_train)
```

```
Out[50]: LogisticRegression
LogisticRegression(random_state=0)
```

```
In [54]: 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 [56]: 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 [84]: import warnings
warnings.filterwarnings("ignore")

res= log.predict([[2,0]])

if(res==0):
    print("Not Survived")
else:
    print("Survived")
```

Survived

```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js