

```
In [1]: 1 import pandas as pd
        2 import numpy as np
        3 import seaborn as sns
        4 import matplotlib.pyplot as plt
        5 import math
```

C:\Users\admin\anaconda3\lib\site-packages\scipy__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.23.5
 warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}")

```
In [36]: 1 titanic = pd.read_csv('Titanic.csv')
        2 titanic.head()
```

Out[36]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

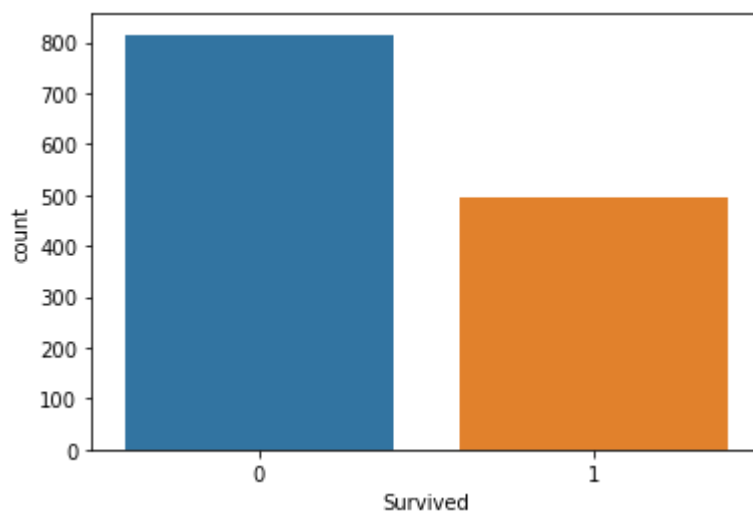
```
In [3]: 1 titanic.tail()
```

```
Out[3]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
1304	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8
1305	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108
1306	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7
1307	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8
1308	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22

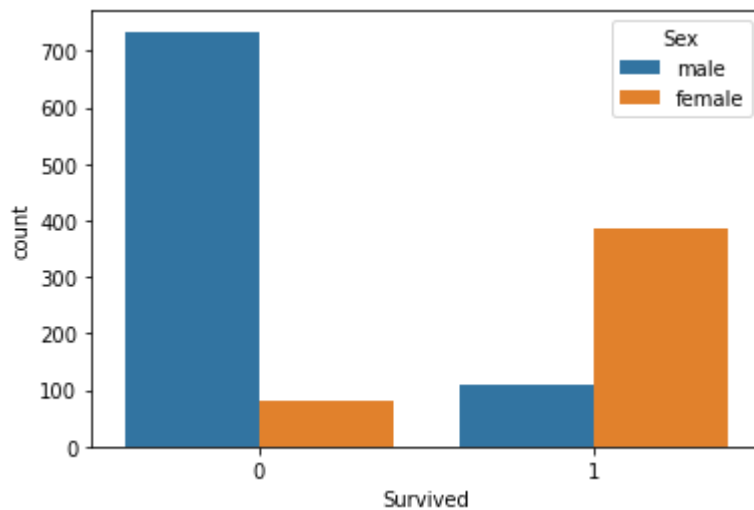
```
In [37]: 1 sns.countplot(x = "Survived", data = titanic)
```

```
Out[37]: <AxesSubplot:xlabel='Survived', ylabel='count'>
```



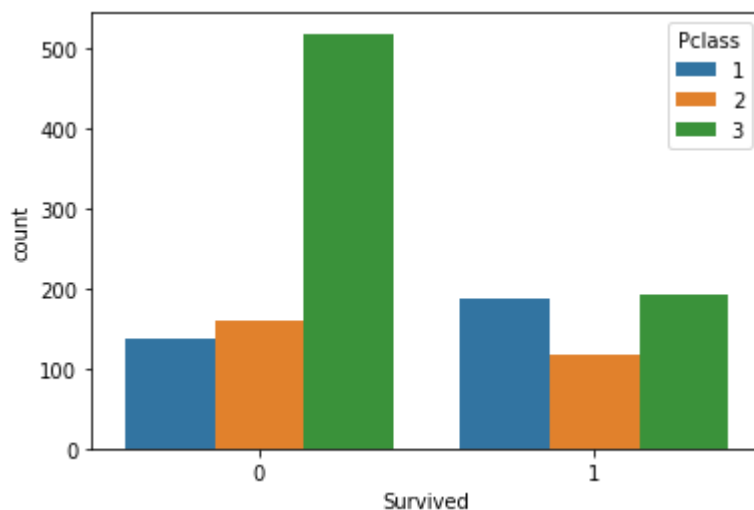
```
In [38]: 1 sns.countplot(x = "Survived", hue = "Sex", data = titanic)
```

```
Out[38]: <AxesSubplot:xlabel='Survived', ylabel='count'>
```



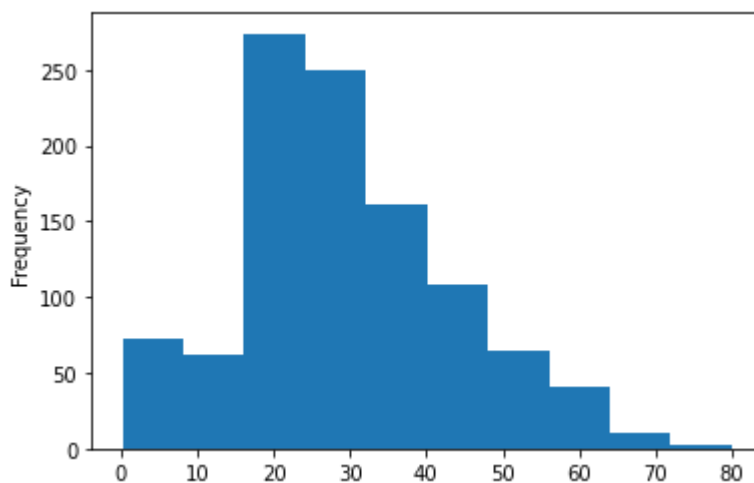
```
In [39]: 1 sns.countplot(x = "Survived", hue = "Pclass", data = titanic)
```

```
Out[39]: <AxesSubplot:xlabel='Survived', ylabel='count'>
```



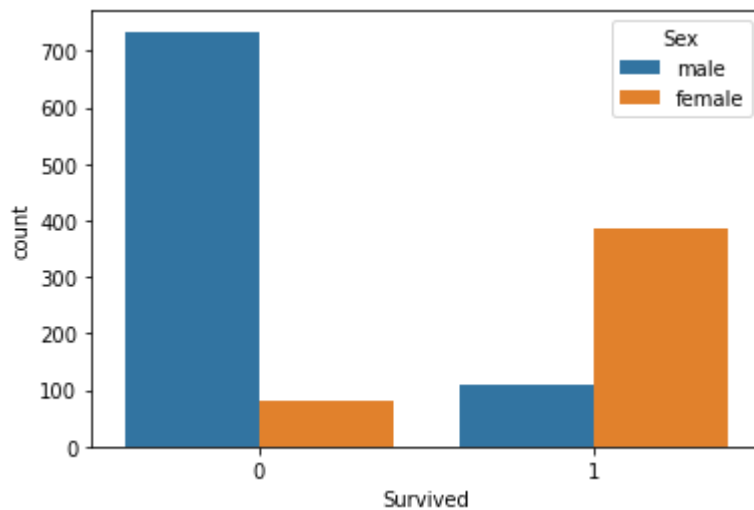
```
In [7]: 1 titanic['Age'].plot.hist()
```

```
Out[7]: <AxesSubplot:ylabel='Frequency'>
```



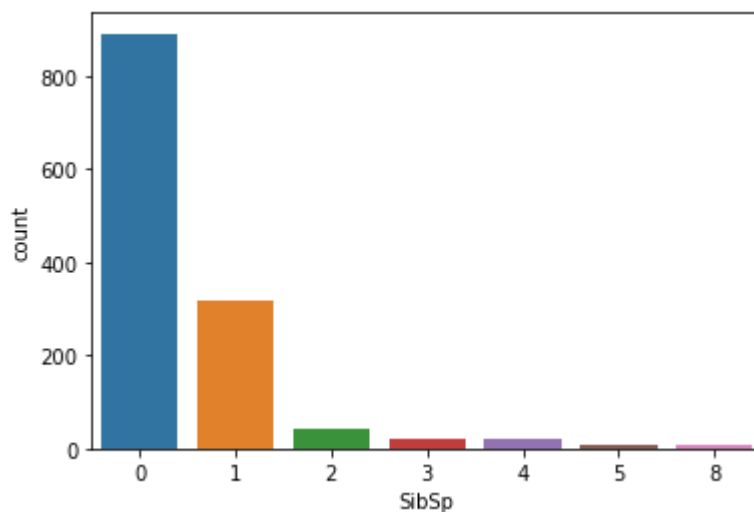
```
In [40]: 1 sns.countplot(x = "Survived", hue = "Sex", data = titanic)
```

```
Out[40]: <AxesSubplot:xlabel='Survived', ylabel='count'>
```



```
In [41]: 1 sns.countplot(x = 'SibSp', data = titanic)
```

```
Out[41]: <AxesSubplot:xlabel='SibSp', ylabel='count'>
```

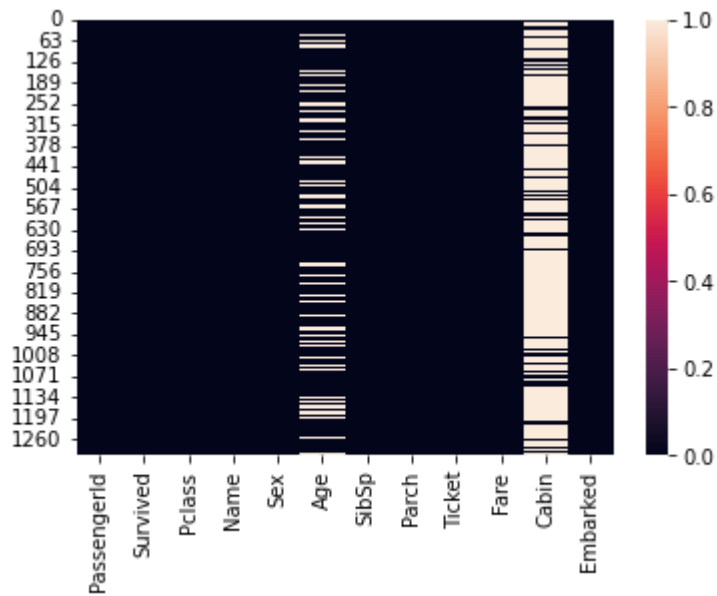


```
In [10]: 1 titanic.isnull().sum()
```

```
Out[10]: PassengerId    0
Survived              0
Pclass               0
Name                 0
Sex                  0
Age                 263
SibSp                0
Parch                0
Ticket              0
Fare                 1
Cabin              1014
Embarked             2
dtype: int64
```

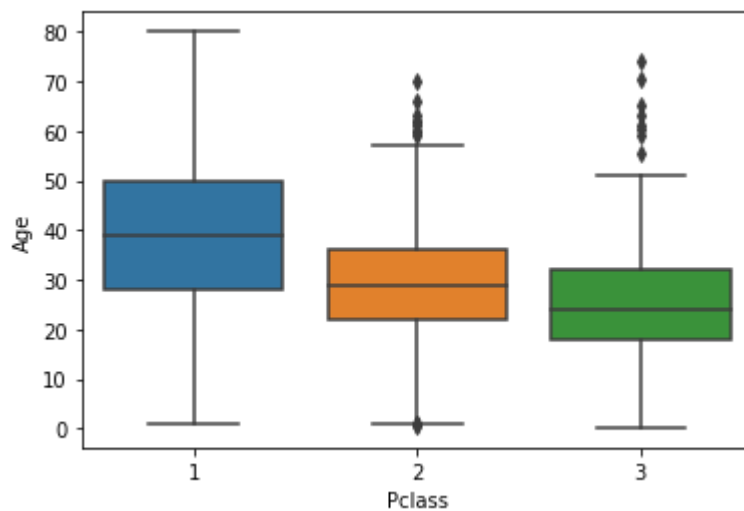
```
In [11]: 1 sns.heatmap(titanic.isnull())
```

```
Out[11]: <AxesSubplot:>
```



```
In [42]: 1 sns.boxplot(x = "Pclass", y = "Age", data = titanic)
```

```
Out[42]: <AxesSubplot:xlabel='Pclass', ylabel='Age'>
```

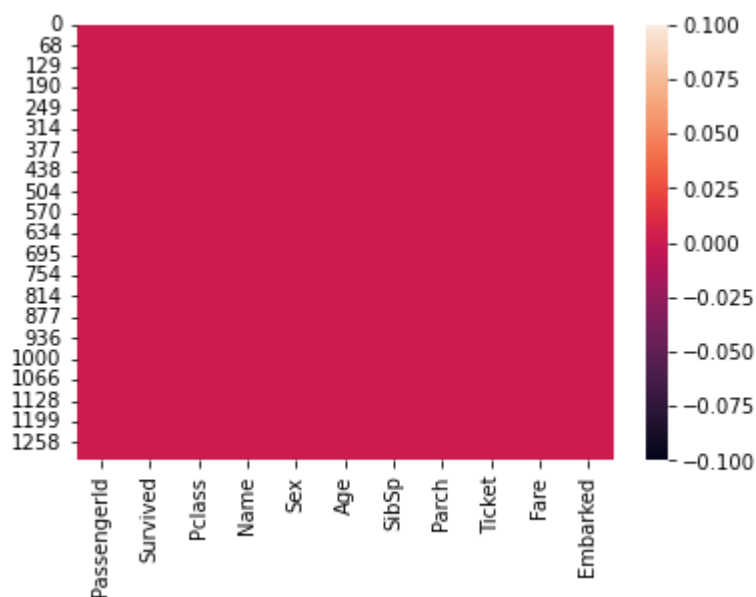


```
In [43]: 1 titanic.drop('Cabin', axis = 1, inplace = True)
```

```
In [44]: 1 titanic.dropna(inplace = True)
```

```
In [15]: 1 sns.heatmap(titanic.isnull())
```

```
Out[15]: <AxesSubplot:>
```



```
In [16]: 1 titanic.isnull().sum()
```

```
Out[16]: PassengerId    0
Survived              0
Pclass               0
Name                 0
Sex                  0
Age                  0
SibSp                0
Parch                0
Ticket               0
Fare                 0
Embarked             0
dtype: int64
```

```
In [17]: 1 pd.get_dummies(titanic['Sex'])
```

```
Out[17]:
```

	female	male
0	0	1
1	1	0
2	1	0
3	1	0
4	0	1
...
1300	1	0
1302	1	0
1303	1	0
1305	1	0
1306	0	1

1043 rows × 2 columns

```
In [45]: 1 gender = pd.get_dummies(titanic['Sex'], drop_first = True)
```

```
In [19]: 1 gender.head()
```

```
Out[19]:
```

	male
0	1
1	0
2	0
3	0
4	1

```
In [46]: 1 Embarked = pd.get_dummies(titanic['Embarked'], drop_first = True)
2 Embarked.head()
```

```
Out[46]:
```

	Q	S
0	0	1
1	0	0
2	0	1
3	0	1
4	0	1

```
In [47]: 1 titanic = pd.concat([titanic, gender, Embarked], axis = 1)
```

```
In [48]: 1 titanic.drop(['Sex', 'Embarked', 'PassengerId', 'Name', 'Ticket'], axis
```

```
In [49]: 1 titanic.drop('Pclass', axis = 1, inplace = True)
         2
         3 titanic.head()
```

```
Out[49]:
```

	Survived	Age	SibSp	Parch	Fare	male	Q	S
0	0	22.0	1	0	7.2500	1	0	1
1	1	38.0	1	0	71.2833	0	0	0
2	1	26.0	0	0	7.9250	0	0	1
3	1	35.0	1	0	53.1000	0	0	1
4	0	35.0	0	0	8.0500	1	0	1

```
In [50]: 1 x = titanic.drop('Survived', axis = 1)
         2 y = titanic['Survived']
```

```
In [51]: 1 from sklearn.model_selection import train_test_split
         2 x_train,x_test,y_train,y_test = train_test_split(x, y, test_size = 0.3,
```

```
In [52]: 1 from sklearn.linear_model import LogisticRegression
         2 logmodel = LogisticRegression()
```

```
In [54]: 1 logmodel.fit(x_train,y_train)
```

C:\Users\admin\anaconda3\lib\site-packages\sklearn\linear_model_logistic.py:763: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html> (<https://scikit-learn.org/stable/modules/preprocessing.html>)

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(
```

```
Out[54]: LogisticRegression()
```

```
In [55]: 1 y_predict = logmodel.predict(x_test)
```

```
In [56]: 1 from sklearn.metrics import classification_report
         2 classification_report(y_test,y_predict)
```

```
Out[56]:
```

		precision	recall	f1-score	support		
0.83	0.87	0.85	180	1	0.81	0.75	
0.78	133		accuracy		0.82	313	
macro avg	0.82	0.81	0.81	313	nweighted avg	0.8	
2	0.82	0.82	313				


```
In [57]: 1 from sklearn.metrics import confusion_matrix  
2         confusion_matrix(y_test,y_predict)
```

```
Out[57]: array([[157,  23],  
               [ 33, 100]], dtype=int64)
```

```
In [58]: 1 from sklearn.metrics import accuracy_score  
2         accuracy_score(y_test,y_predict)
```

```
Out[58]: 0.8210862619808307
```