

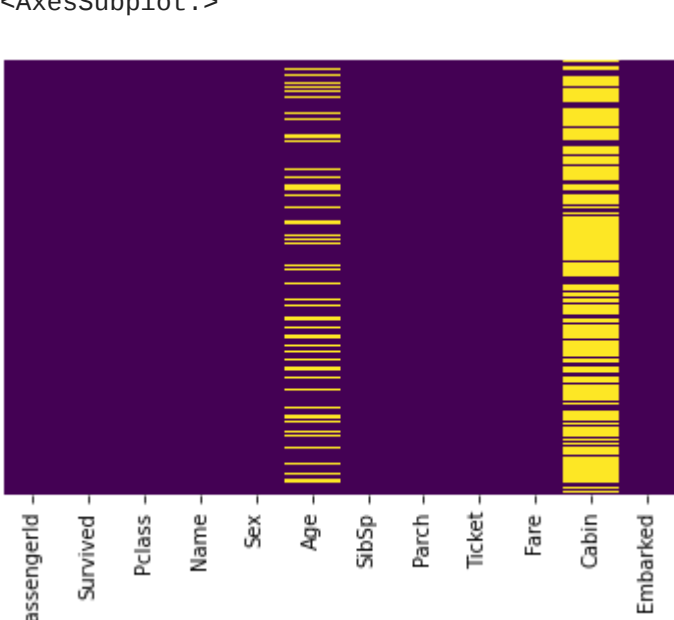
# LOGISTIC REGRESSION - TITANIC SURVIVAL

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

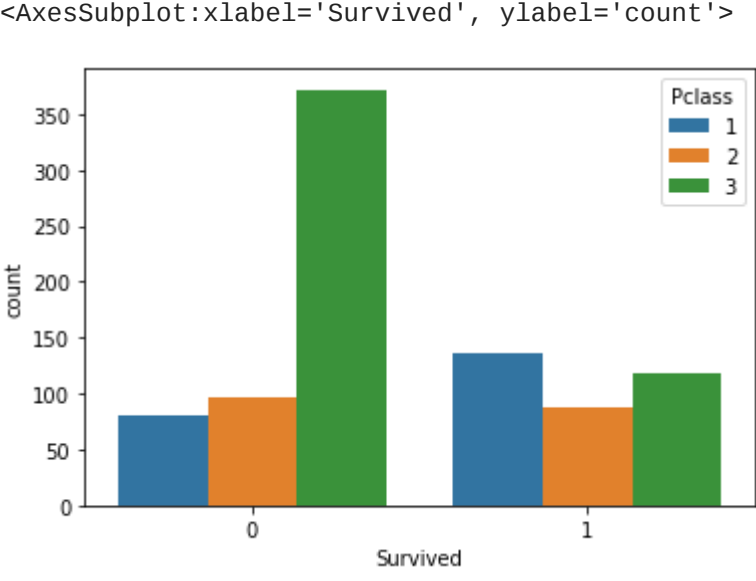
```
In [2]: train = pd.read_csv('titanic_train.csv')
train.head(2)
```

	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	Cummings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C

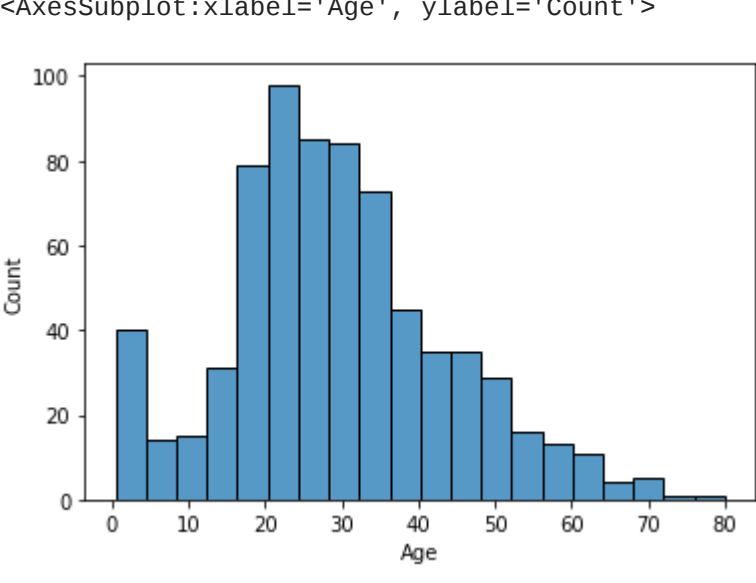
```
In [3]: #Plot to check the null values in the dataset
sns.heatmap(train.isnull(), yticklabels=False, cbar=False, cmap = 'viridis')
```



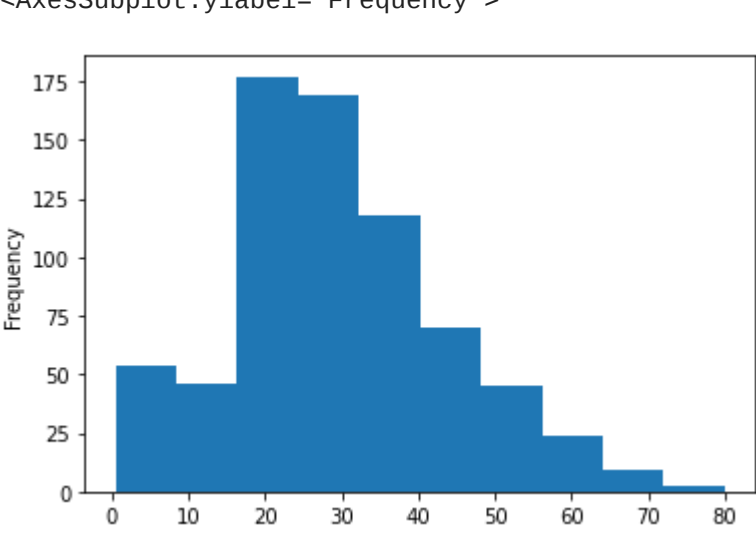
```
In [5]: sns.countplot(x='Survived', data = train, hue = 'Pclass')
```



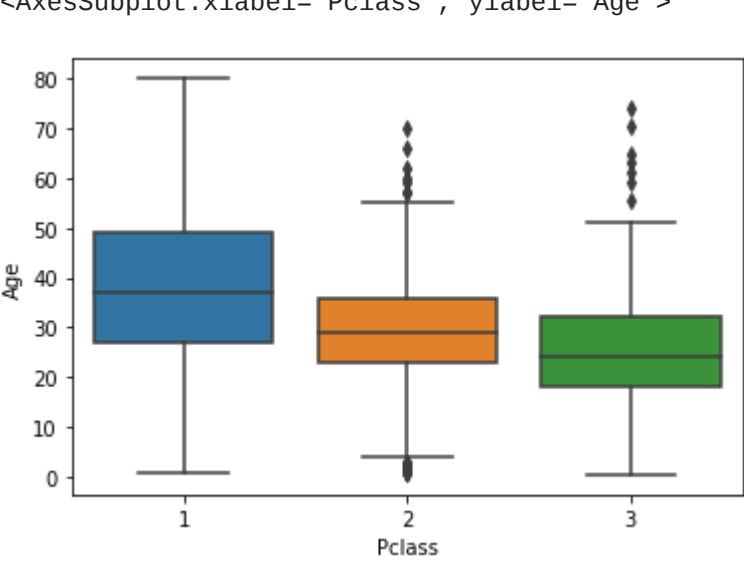
```
In [7]: sns.histplot(train['Age'].dropna(), kde=False)
```



```
In [9]: train['Age'].plot.hist()
```



```
In [10]: sns.boxplot(x='Pclass', y = 'Age', data = train)
```

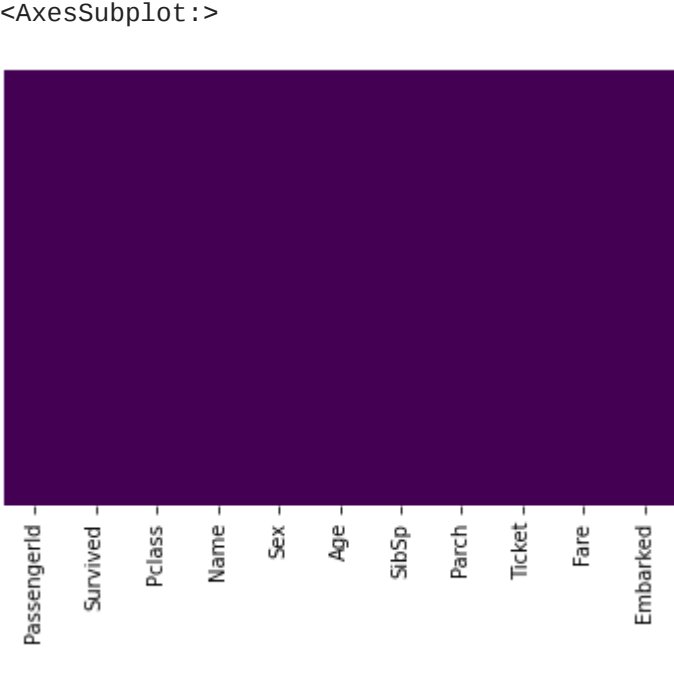


```
In [13]: #For the null values in age , we will impute the mean age according to the Passenger class
def impute_age(cols):
    Age = cols[0]
    Pclass = cols[1]

    if pd.isnull(Age):
        if Pclass == 1:
            return 37
        elif Pclass == 2:
            return 29
        else:
            return 24
    else:
        return Age
```

```
In [14]: train['Age']= train[['Age','Pclass']].apply(impute_age , axis = 1)
```

```
In [19]: sns.heatmap(train.isnull(), yticklabels=False, cbar = False, cmap = 'viridis')
```



```
In [17]: #Due to many null values for cabin , we will drop the column
train.drop('Cabin', axis =1, inplace = True)
```

```
In [20]: train.dropna(inplace = True)
```

```
In [21]: #Conversion of categorical features into numerical features using dummy variables
sex = pd.get_dummies(train['Sex'], drop_first = True)
```

```
In [22]: sex.head()
```

	male
0	1
1	0
2	0
3	0
4	1

```
In [26]: embark = pd.get_dummies(train['Embarked'], drop_first = True)
```

```
In [27]: embark
```

	Q	S
0	0	1
1	0	0
2	0	1
3	0	1
4	0	1
...	...	...
886	0	1
887	0	1
888	0	1
889	0	0
890	1	0

889 rows × 2 columns

```
In [29]: train = pd.concat([train, sex, embark ], axis = 1)
```

```
In [30]: train.head(2)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	male	Q	S
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S	1	0	1
1	2	1	1	Cummings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C	0	0	0

```
In [31]: #Drop categorical Columns with strings
train.drop(['Sex', 'Embarked', 'Name', 'PassengerId','Ticket'], axis=1, inplace=True)
```

```
In [32]: train.head(2)
```

	Survived	Pclass	Age	SibSp	Parch	Fare	male	Q	S
0	0	3	22.0	1	0	7.2500	1	0	1
1	1	1	38.0	1	0	71.2833	0	0	0

```
In [33]: #Initialize Input and output
X = train.drop('Survived', axis=1)
y = train['Survived']
```

```
In [34]: from sklearn.model_selection import train_test_split
```

```
In [35]: X_train,X_test ,y_train,y_test = train_test_split(X,y,test_size=0.3)
```

```
In [36]: from sklearn.linear_model import LogisticRegression
```

```
In [37]: log = LogisticRegression()
```

```
In [38]: log.fit(X_train,y_train)
```

```
C:\Users\siddh\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: 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
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
n_iter_i = _check_optimize_result(
LogisticRegression()
```

```
In [39]: p = log.predict(X_test)
```

```
In [41]: from sklearn.metrics import classification_report , confusion_matrix
```

```
In [42]: print(classification_report(y_test,p))
print('\n')
```

	precision	recall	f1-score	support
0	0.82	0.90	0.86	164
1	0.82	0.69	0.75	103
accuracy			0.82	267
macro avg	0.82	0.80	0.80	267
weighted avg	0.82	0.82	0.82	267

```
[[148 16]
 [ 32 71]]
```

In [ ] :