Exploratory Data Analysis (Titanic DataSet)

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

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

In [3]:
```
train.head()
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN

Finding the Missing Data

In [4]:
```
train.isnull()
```

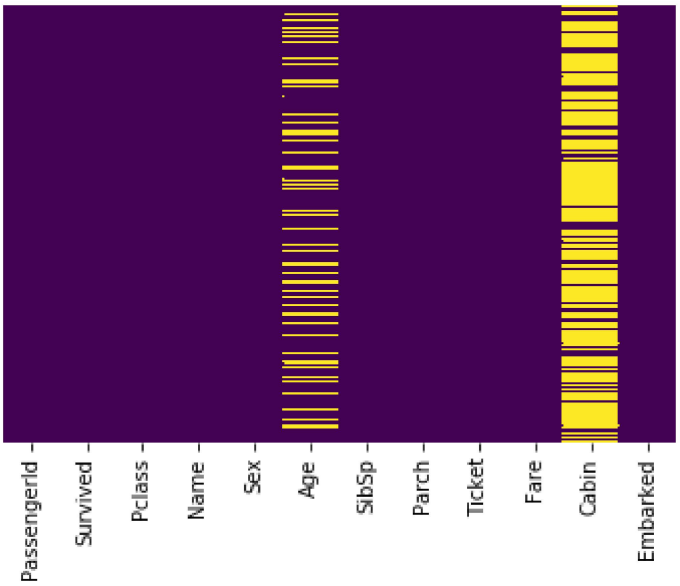
Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embar
0	False	False	False	False	False	False	False	False	False	False	True	F
1	False	False	False	False	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	False	False	False	True	F
3	False	False	False	False	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	False	False	False	True	F
5	False	False	False	False	False	True	False	False	False	False	True	F
6	False	False	False	False	False	False	False	False	False	False	False	F
7	False	False	False	False	False	False	False	False	False	False	True	F
8	False	False	False	False	False	False	False	False	False	False	True	F
9	False	False	False	False	False	False	False	False	False	False	True	F
10	False	False	False	False	False	False	False	False	False	False	False	F
11	False	False	False	False	False	False	False	False	False	False	False	F
12	False	False	False	False	False	False	False	False	False	False	True	F
13	False	False	False	False	False	False	False	False	False	False	True	F
14	False	False	False	False	False	False	False	False	False	False	True	F
15	False	False	False	False	False	False	False	False	False	False	True	F
16	False	False	False	False	False	False	False	False	False	False	True	F
17	False	False	False	False	False	True	False	False	False	False	True	F
18	False	False	False	False	False	False	False	False	False	False	True	F
19	False	False	False	False	False	True	False	False	False	False	True	F
20	False	False	False	False	False	False	False	False	False	False	True	F
21	False	False	False	False	False	False	False	False	False	False	False	F
22	False	False	False	False	False	False	False	False	False	False	True	F
23	False	False	False	False	False	False	False	False	False	False	False	F
24	False	False	False	False	False	False	False	False	False	False	True	F
25	False	False	False	False	False	False	False	False	False	False	True	F
26	False	False	False	False	False	True	False	False	False	False	True	F
27	False	False	False	False	False	False	False	False	False	False	False	F
28	False	False	False	False	False	True	False	False	False	False	True	F
29	False	False	False	False	False	True	False	False	False	False	True	F
...	...	...	...	...	...	...	...	...	...	...	...	
861	False	False	False	False	False	False	False	False	False	False	True	F
862	False	False	False	False	False	False	False	False	False	False	False	F
863	False	False	False	False	False	True	False	False	False	False	True	F
864	False	False	False	False	False	False	False	False	False	False	True	F
865	False	False	False	False	False	False	False	False	False	False	True	F

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embar
866	False	False	False	False	False	False	False	False	False	False	True	F
867	False	False	False	False	False	False	False	False	False	False	False	F
868	False	False	False	False	False	True	False	False	False	False	True	F
869	False	False	False	False	False	False	False	False	False	False	True	F
870	False	False	False	False	False	False	False	False	False	False	True	F
871	False	False	False	False	False	False	False	False	False	False	False	F
872	False	False	False	False	False	False	False	False	False	False	False	F
873	False	False	False	False	False	False	False	False	False	False	True	F
874	False	False	False	False	False	False	False	False	False	False	True	F
875	False	False	False	False	False	False	False	False	False	False	True	F
876	False	False	False	False	False	False	False	False	False	False	True	F
877	False	False	False	False	False	False	False	False	False	False	True	F
878	False	False	False	False	False	True	False	False	False	False	True	F
879	False	False	False	False	False	False	False	False	False	False	False	F
880	False	False	False	False	False	False	False	False	False	False	True	F
881	False	False	False	False	False	False	False	False	False	False	True	F
882	False	False	False	False	False	False	False	False	False	False	True	F
883	False	False	False	False	False	False	False	False	False	False	True	F
884	False	False	False	False	False	False	False	False	False	False	True	F
885	False	False	False	False	False	False	False	False	False	False	True	F
886	False	False	False	False	False	False	False	False	False	False	True	F
887	False	False	False	False	False	False	False	False	False	False	False	F
888	False	False	False	False	False	True	False	False	False	False	True	F
889	False	False	False	False	False	False	False	False	False	False	False	F
890	False	False	False	False	False	False	False	False	False	False	True	F

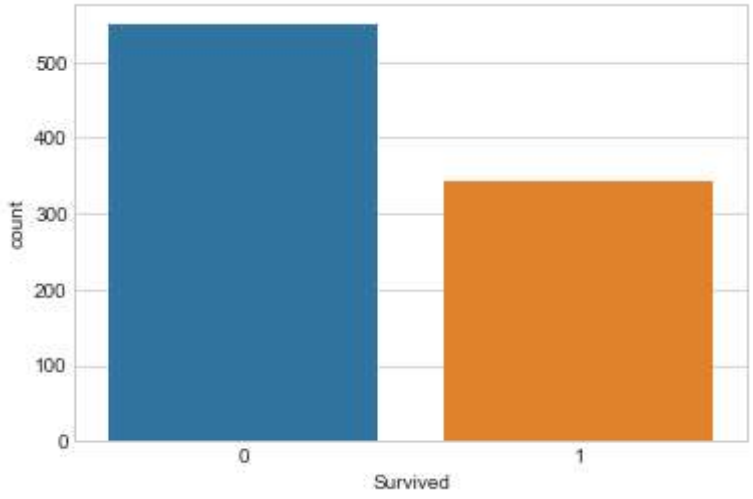
In [10]:
```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[10]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xc732278>
```



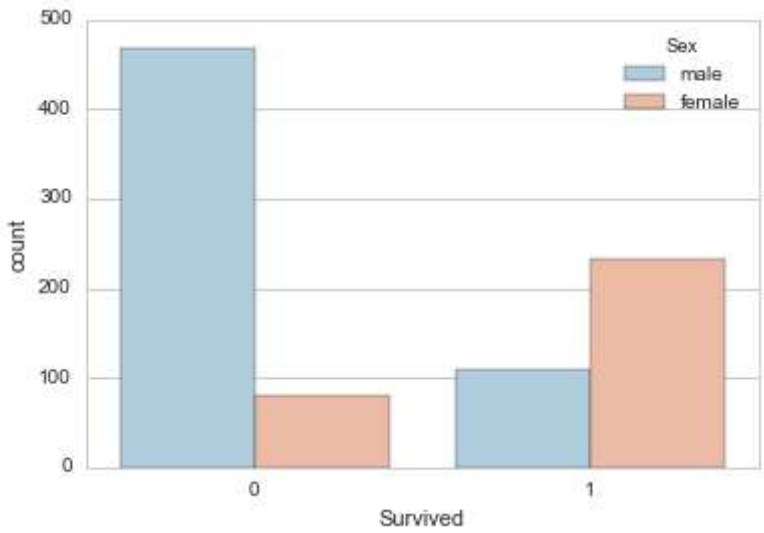
In [11]:
```
sns.set_style('whitegrid')
sns.countplot(x='Survived',data=train)
```

Out[11]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xc6ebf98>
```



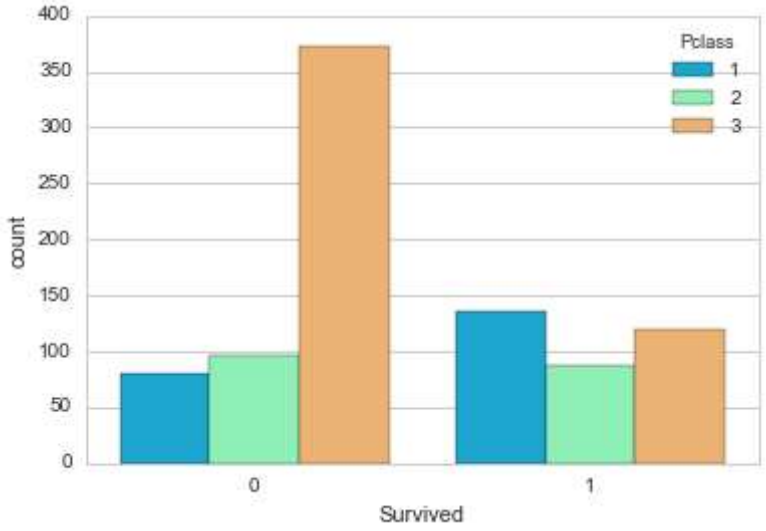
In [78]:
```
sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Sex',data=train,palette='RdBu_r')
```

Out[78]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x11b004a20>
```



In [79]:
```
sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Pclass',data=train,palette='rainbow')
```

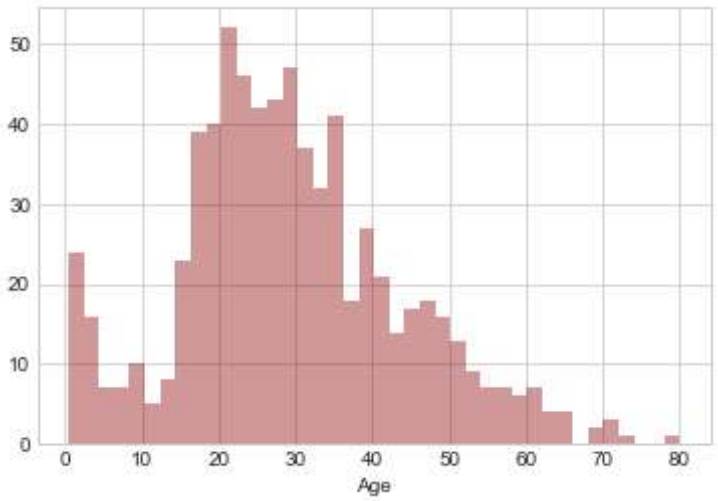
Out[79]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x11b130f28>
```



In [22]:
```
sns.distplot(train['Age'].dropna(),kde=False,color='darkred',bins=40)
```

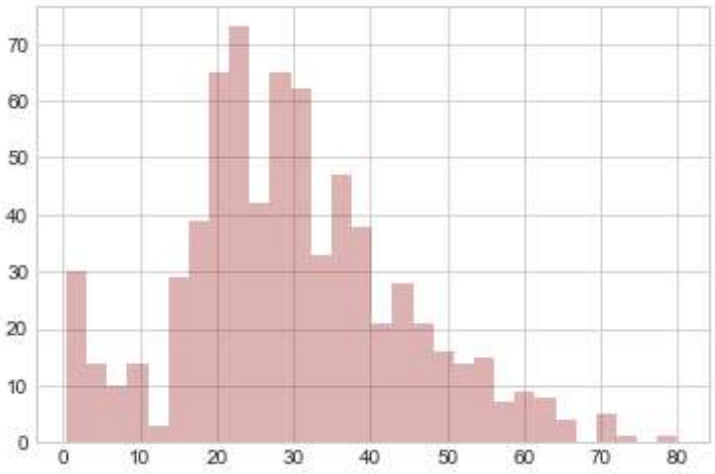
```
D:\anaconda\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'nor
med' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

Out[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xe0dea58>
```



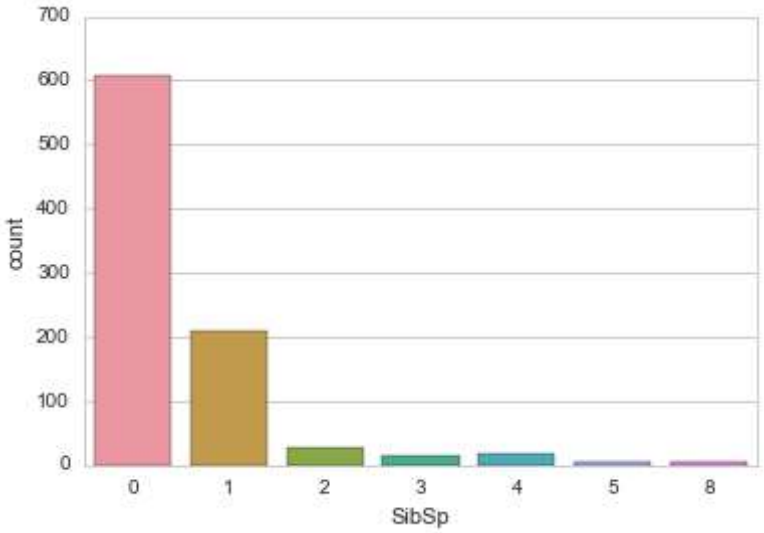
In [25]:
```
train['Age'].hist(bins=30,color='darkred',alpha=0.3)
```

Out[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xe2d8978>
```



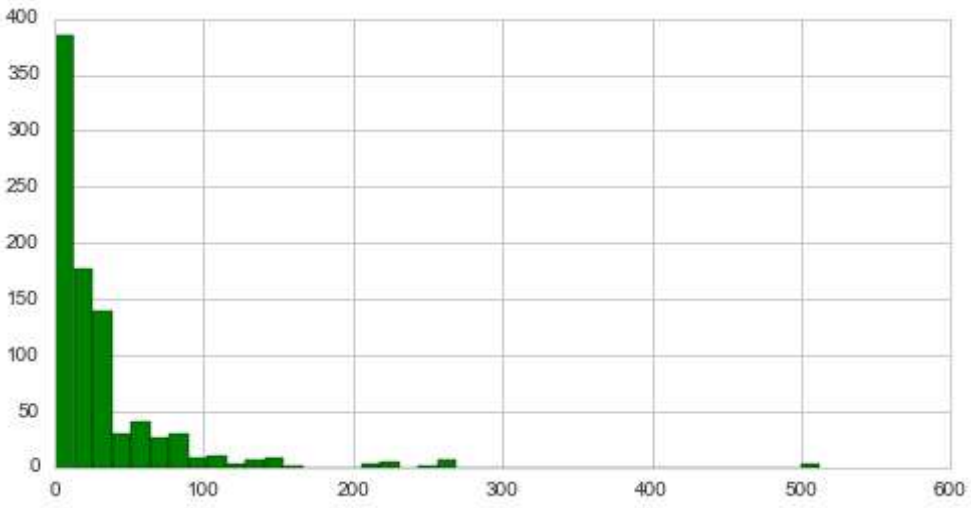
In [82]:
```
sns.countplot(x='SibSp',data=train)
```

Out[82]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x11c4139e8>
```



In [83]:
```
train['Fare'].hist(color='green',bins=40,figsize=(8,4))
```

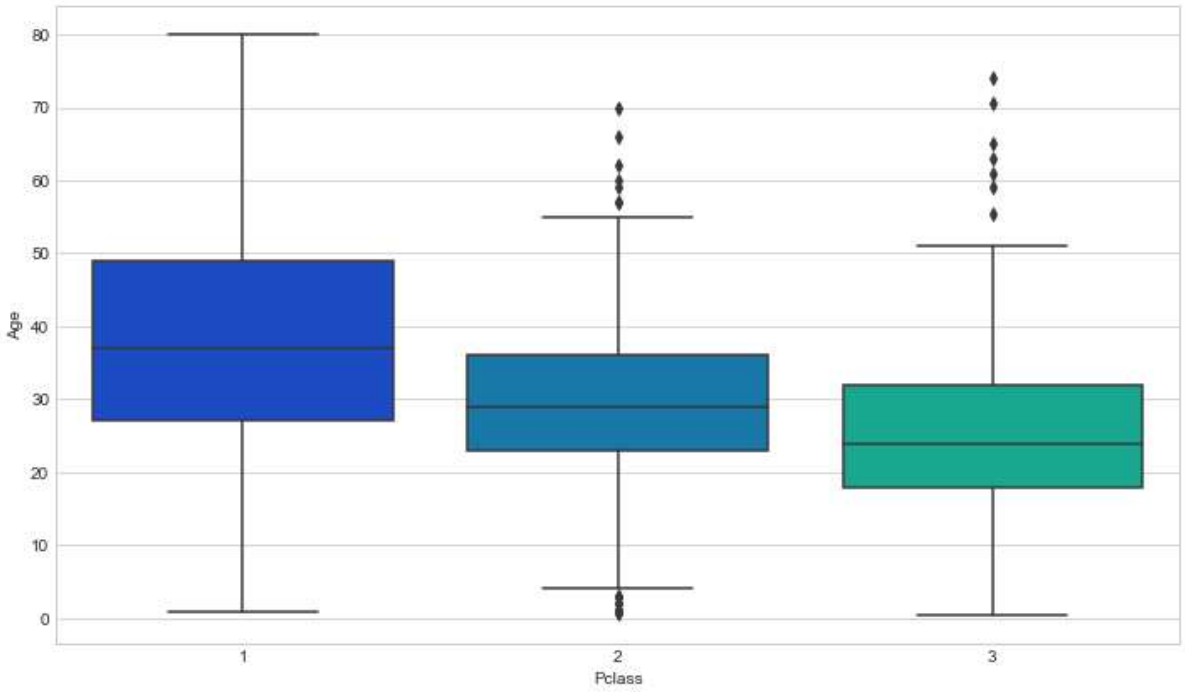
Out[83]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x113893048>
```



Data Cleaning

In [26]:
```
plt.figure(figsize=(12, 7))
sns.boxplot(x='Pclass',y='Age',data=train,palette='winter')
```

Out[26]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xe27f780>
```



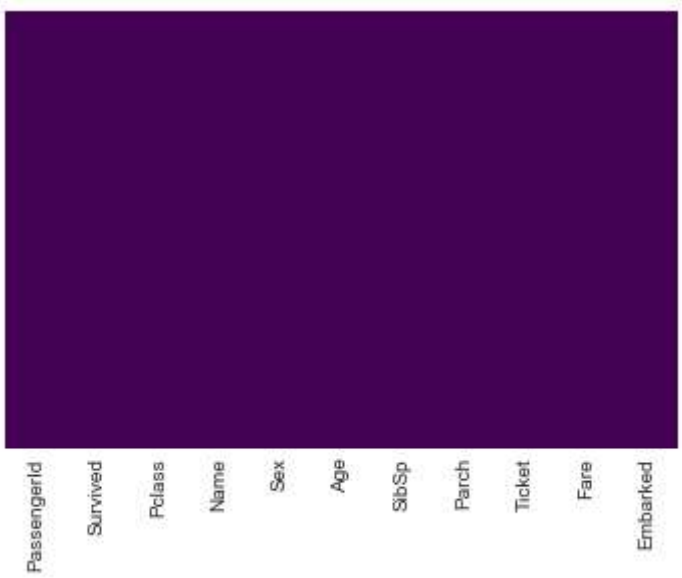
Creating a function to fill missing values of Age Column

In [27]:
```
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 [28]:
```
train['Age'] = train[['Age','Pclass']].apply(impute_age,axis=1)
```

In [33]:
```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[33]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xe4c27b8>
```



In [30]:
```
train.drop('Cabin',axis=1,inplace=True)
```

In [91]:
```
train.head()
```

Out[91]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Emba
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 [32]:
```
train.dropna(inplace=True)
```

Converting Categorical Features

In [93]:
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 889 entries, 0 to 890
Data columns (total 11 columns):
PassengerId    889 non-null int64
Survived       889 non-null int64
Pclass         889 non-null int64
Name           889 non-null object
Sex            889 non-null object
Age            889 non-null float64
SibSp          889 non-null int64
Parch          889 non-null int64
Ticket         889 non-null object
Fare           889 non-null float64
Embarked       889 non-null object
dtypes: float64(2), int64(5), object(4)
memory usage: 83.3+ KB
```

In [37]:
```
pd.get_dummies(train['Embarked'],drop_first=True).head()
```

Out[37]:

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

In [38]:
```
sex = pd.get_dummies(train['Sex'],drop_first=True)
embark = pd.get_dummies(train['Embarked'],drop_first=True)
```

In [39]:
```
train.drop(['Sex','Embarked','Name','Ticket'],axis=1,inplace=True)
```

In [40]:
```
train.head()
```

Out[40]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	1	0	3	22.0	1	0	7.2500
1	2	1	1	38.0	1	0	71.2833
2	3	1	3	26.0	0	0	7.9250
3	4	1	1	35.0	1	0	53.1000
4	5	0	3	35.0	0	0	8.0500

In [96]:
```
train = pd.concat([train,sex,embark],axis=1)
```

In [97]:
```
train.head()
```

Out[97]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	male	Q	S
0	1	0	3	22.0	1	0	7.2500	1.0	0.0	1.0
1	2	1	1	38.0	1	0	71.2833	0.0	0.0	0.0
2	3	1	3	26.0	0	0	7.9250	0.0	0.0	1.0
3	4	1	1	35.0	1	0	53.1000	0.0	0.0	1.0
4	5	0	3	35.0	0	0	8.0500	1.0	0.0	1.0

Building a Logistic Regression model

Train Test Split

In [42]:
```
train.drop('Survived',axis=1).head()
```

Out[42]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare
0	1	3	22.0	1	0	7.2500
1	2	1	38.0	1	0	71.2833
2	3	3	26.0	0	0	7.9250
3	4	1	35.0	1	0	53.1000
4	5	3	35.0	0	0	8.0500

In [43]:
```
train['Survived'].head()
```

Out[43]:
```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

In [45]:
```
from sklearn.model_selection import train_test_split
```

In [46]:
```
X_train, X_test, y_train, y_test = train_test_split(train.drop('Survived',axis=1),
                                                    train['Survived'], test_size=0
                                                    random_state=101)
```

Training and Predicting

In [47]:
```
from sklearn.linear_model import LogisticRegression
```

In [48]:
```
logmodel = LogisticRegression()
logmodel.fit(X_train,y_train)
```

Out[48]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

In [49]:
```
predictions = logmodel.predict(X_test)
```

In [52]:
```
from sklearn.metrics import confusion_matrix
```

In [54]:
```
accuracy=confusion_matrix(y_test,predictions)
```

In [55]:
```
accuracy
```

Out[55]:
```
array([[144,  19],
       [ 56,  48]], dtype=int64)
```

In [57]:
```
from sklearn.metrics import accuracy_score
```

In [58]:
```
accuracy=accuracy_score(y_test,predictions)
accuracy
```

Out[58]:
```
0.7191011235955056
```

In [50]:
```
predictions
```

Out[50]:
```
array([0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0,
       0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1,
       0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1,
       1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 1, 0], dtype=int64)
```

Evaluation

In [104…
```
from sklearn.metrics import classification_report
```

In [105…
```
print(classification_report(y_test,predictions))
```

```
             precision    recall  f1-score   support

          0       0.81      0.93      0.86       163
          1       0.85      0.65      0.74       104

avg / total       0.82      0.82      0.81       267
```