In [45]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [46]:
```
train = pd.read_csv('titanic_train.csv')
```

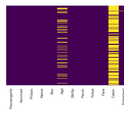
In [47]:
```
train.head()
```

Out[47]:

	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	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

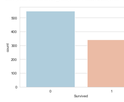
In [48]:
```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[48]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1fa363fab00>
```



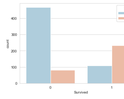
In [49]:
```
sns.set_style('whitegrid')
sns.countplot(x='Survived',data=train,palette='RdBu_r')
```

Out[49]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1fa3619ada0>
```



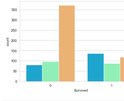
In [50]:
```
sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Sex',data=train,palette='RdBu_r')
```

Out[50]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1fa36498588>
```



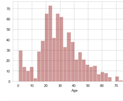
In [51]:
```
sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Pclass',data=train,palette='rainbow')
```

Out[51]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1fa364e4160>
```



In [52]:
```
sns.distplot(train['Age'].dropna(),kde=False,color='darkred',bins=30)
```

Out[52]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1fa3651ddd0>
```

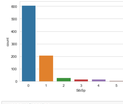


In [53]:
```
train['Age'].hist(bins=30,color='darkred',alpha=0.7)
```

Out[53]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1fa35d4fe48>
```

In [54]:
```
sns.countplot(x='SibSp',data=train)
```

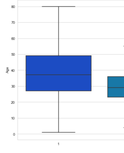
Out[54]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1fa366c7320>
```



In [55]:
```
##Data Cleaning
```

In [56]:
```
plt.figure(figsize=(12, 7))
sns.boxplot(x='Pclass',y='Age',data=train,palette='winter')
```

Out[56]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1fa366efeb8>
```

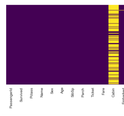


In [57]:
```
def impute_age(cols):
    Age = cols[0]
    Pclass = cols[1]

    if pd.isnull(Age):

        if Pclass == 1:
            return 37

        elif Pclass == 2:
            return 29

        else:
            return 24

    else:
        return Age
```

In [58]:
```
#apply that function
train['Age'] = train[['Age','Pclass']].apply(impute_age,axis=1)
```

In [59]:
```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

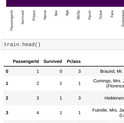
Out[59]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1fa37787b8>
```



In [60]:
```
train.drop('Cabin',axis=1,inplace=True)
```

In [61]:
```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[61]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1fa3798d2a8>
```



In [62]:
```
train.head()
```

Out[62]:

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

In [63]:
```
##Converting Categorical Features
#Need to convert categorical features to dummy variables using pandas!
```

In [64]:
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Name           891 non-null object
Sex            891 non-null object
Age            891 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Embarked       889 non-null object
dtypes: float64(2), int64(5), object(4)
memory usage: 76.6+ KB
```

In [67]:
```
#pd.get_dummies(train.Sex)
```

In [68]:
```
sex = pd.get_dummies(train['Sex'],drop_first=True)
embark = pd.get_dummies(train['Embarked'],drop_first=True)
```

In [69]:
```
train.drop(['Sex', 'Embarked','Name','Ticket'],axis=1,inplace=True)
```

In [70]:
```
train = pd.concat([train,sex,embark],axis=1)
```

In [80]:
```
train.drop('PassengerId',inplace=True,axis=1 )
```

In [81]:
```
train.head()
```

Out[81]:

	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
2	1	3	26.0	0	0	7.9250	0	0	1
3	1	1	35.0	1	0	53.1000	0	0	1
4	0	3	35.0	0	0	8.0500	1	0	1

In [ ]:
```
##Building a Logistic Regression model
```

In [72]:
```
from sklearn.model_selection import train_test_split
```

In [73]:
```
X_train, X_test, y_train, y_test = train_test_split(train.drop('Survived',axis=1),
                                                    train['Survived'], test_size=0.30,
                                                    random_state=101)
```

In [74]:
```
from sklearn.linear_model import LogisticRegression
```

In [75]:
```
logmodel = LogisticRegression()
logmodel.fit(X_train,y_train)
```

```
C:\Users\Shilpa\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
```

Out[75]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='warn',
                   n_jobs=None, penalty='l2', random_state=None, solver='warn',
                   tol=0.0001, verbose=0, warm_start=False)
```

In [76]:
```
predictions = logmodel.predict(X_test)
```

In [77]:
```
from sklearn.metrics import classification_report
```

In [78]:
```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.77      0.88      0.82       154
           1       0.79      0.64      0.71       114

    accuracy                           0.78       268
   macro avg       0.78      0.76      0.76       268
weighted avg       0.78      0.78      0.77       268
```

In [82]:
```
from sklearn import metrics
cm = metrics.confusion_matrix(y_test,predictions)
print(cm)
```

```
[[135  19]
 [ 41  73]]
```

In [ ]:
```
```

In [ ]:
```
```