Karanjot singh

In [106]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#Importing Libraries
```

In [107]:

```
titanic  = pd.read_csv('train.csv')
titanic.head()
#Reading data from train.csv, and printing the top 5 values
```

Out[107]:

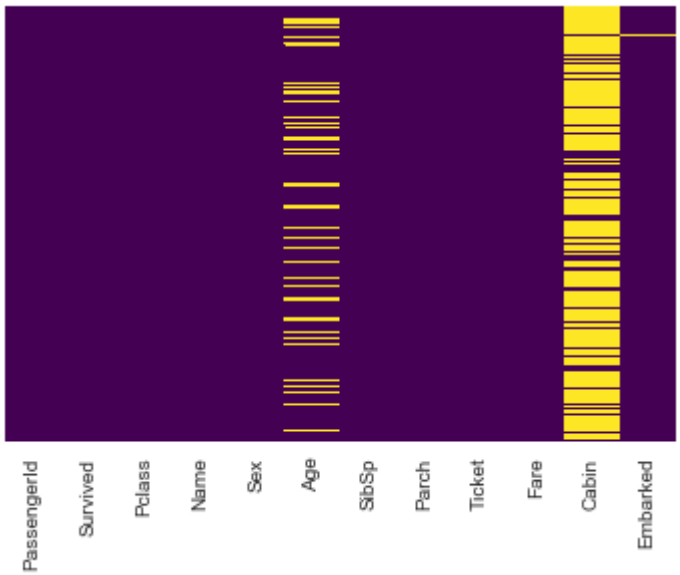
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, 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 [108]:

```
sns.heatmap(titanic.isnull(), yticklabels = False, cbar = False, cmap='viridis')
# Shows the null values in titanic data set with respect to column
```

Out[108]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2459ec3f348>
```

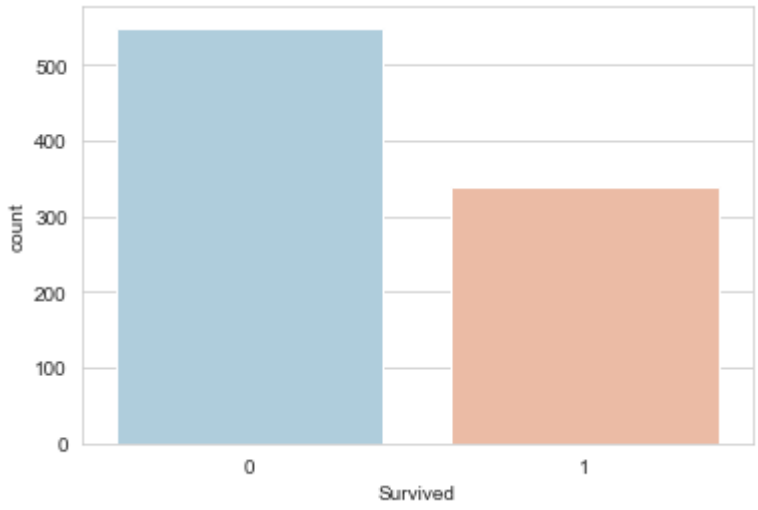


In [128]:

```
sns.set_style('whitegrid')
sns.countplot(x='Survived',data=titanic,palette='RdBu_r')
# we use countplot to Show the counts of observations in each categorical bin using bar
s .
#Using count plot to display number of survived (0 for Dead and 1 for Survived).
```

Out[128]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2459f2a0cc8>
```

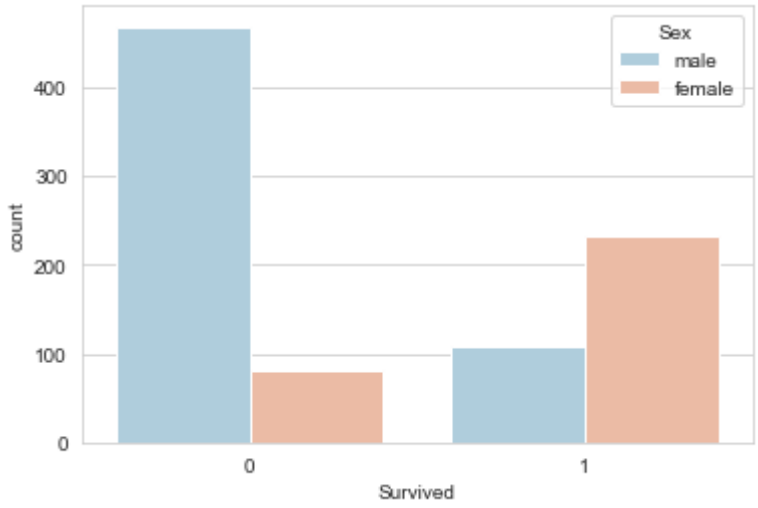


In [110]:

```
sns.set_style('whitegrid')
sns.countplot(x = 'Survived',hue = 'Sex',data = titanic,palette='RdBu_r')
# we use countplot to Show the counts of observations in each categorical bin using bar
s .
# Using countplot to display number of survived (0 for Dead and 1 for Survived) by gend
er.
```

Out[110]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2459eccff88>
```

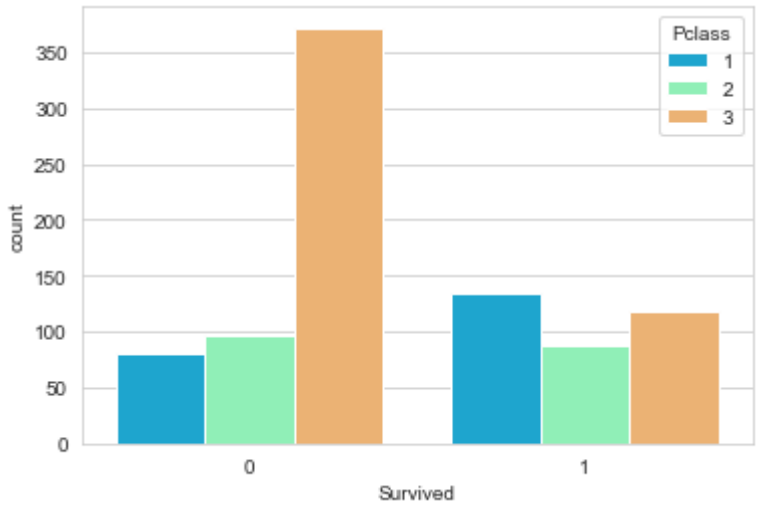


In [129]:

```
sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Pclass',data=titanic,palette='rainbow')
# we use countplot to Show the counts of observations in each categorical bin using bar
s .
# Using countplot to display number of survived (0 for Dead and 1 for Survived) by Pcla
ss.
```

Out[129]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2459f2fd6c8>
```

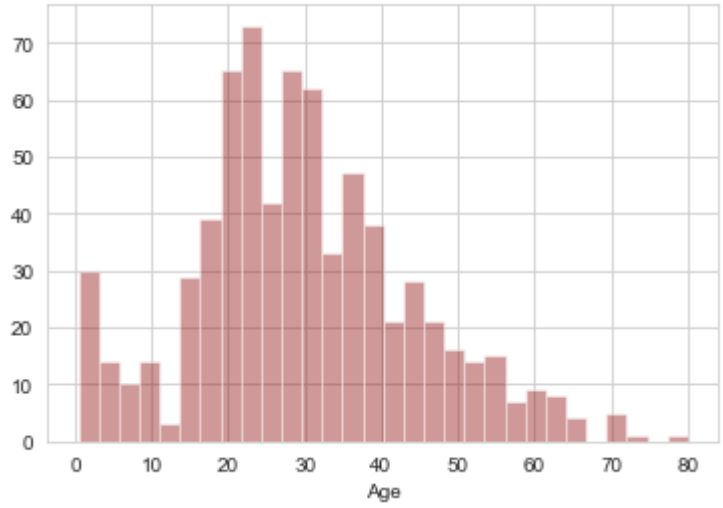


In [112]:

```
sns.distplot(titanic['Age'].dropna(),kde=False,color='darkred',bins=30)
#it shows the frequency of people's age in ship.
```

Out[112]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2459ef0e348>
```

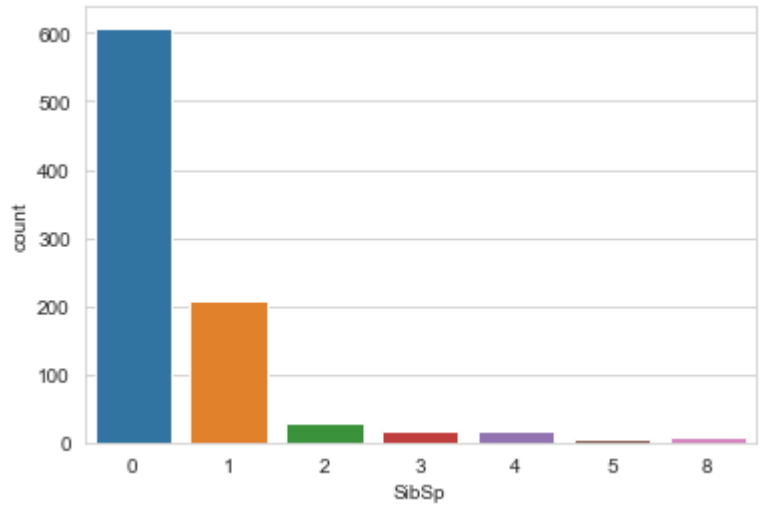


In [113]:

```
sns.countplot(x='SibSp',data=titanic)
#it shows the number of siblings (0 means no ,1 means 1 and so on..).
```

Out[113]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2459efb2d48>
```

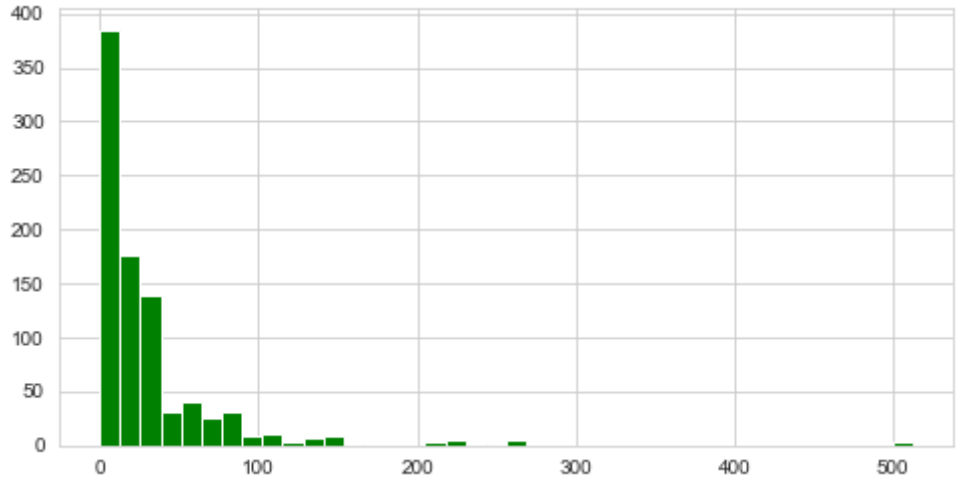


In [114]:

```
titanic['Fare'].hist(color='green',bins=40,figsize=(8,4))
# it shows the number of people paid for the ticket(cost).
```

Out[114]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2459f042bc8>
```

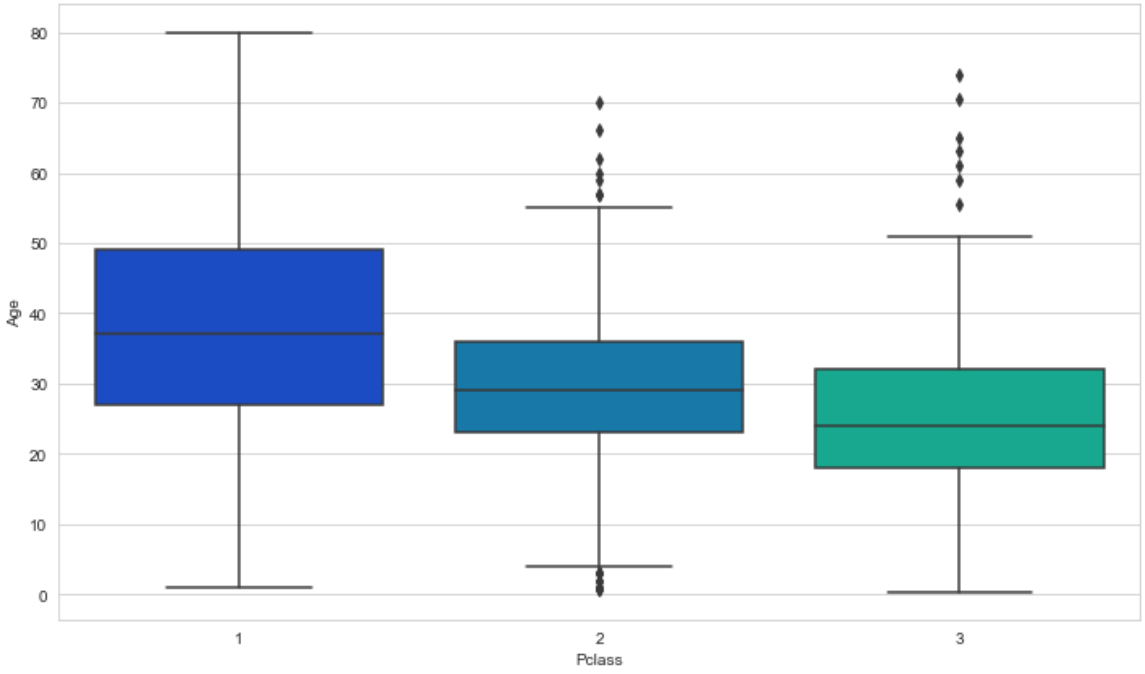


In [115]:

```
plt.figure(figsize=(12, 7))
sns.boxplot(x='Pclass',y='Age',data=titanic,palette='winter')
#boxplot shows the number of people buyed the ticket with respect to Pclass.
```

Out[115]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2459f056448>
```



In [116]:

```
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

#Data cleaning
#handling null values in Age with respect to Pclass eg: Pclass=1 age:37, Pclass2 age:2
9, Pclass=3 age:24.
```

In [117]:

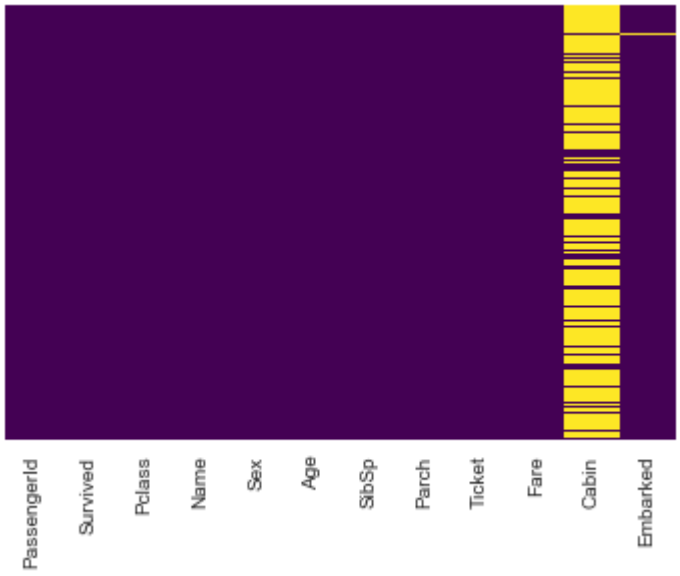
```
titanic['Age'] = titanic[['Age','Pclass']].apply(impute_age,axis=1)
```

In [118]:

```
sns.heatmap(titanic.isnull(),yticklabels=False,cbar=False,cmap='viridis')
#showing the null age replaced with values.
```

Out[118]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2459f1befc8>
```



In [119]:

```
titanic.drop('Cabin',axis=1,inplace=True)
titanic.head()
#droping the colum Cabin fron data set.
#print the first 5 values.
```

Out[119]:

	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 [130]:

```
titanic.dropna(inplace = True)
titanic.info()
#Droping blank values from dataset
#showing info of dataset
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 889 entries, 0 to 890
Data columns (total 8 columns):
PassengerId    889 non-null int64
Survived       889 non-null int64
Pclass         889 non-null int64
Sex            889 non-null int32
Age            889 non-null float64
SibSp          889 non-null int64
Parch          889 non-null int64
Fare           889 non-null float64
dtypes: float64(2), int32(1), int64(5)
memory usage: 59.0 KB
```

In [121]:

```
titanic.drop('Name',axis=1,inplace=True)
titanic.drop('Ticket',axis=1,inplace=True)
titanic.drop('Embarked',axis=1,inplace=True)
#droping String Values Name , Ticket , Embarked from dataset.
```

In [122]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
number = LabelEncoder()
titanic["Sex"] = number.fit_transform(titanic['Sex'].astype(str))
X_train, X_test, y_train, y_test = train_test_split(titanic.drop('Survived',axis=1),tit
anic['Survived'], test_size=0.2,random_state=101)
#Importing Libraries
#using LabelEncoder to code 'Sex' in binary (0 and 1).
#splitting the Dataset into 80,20 in train and test set.
```

In [123]:

```
logmodel = LogisticRegression()
logmodel.fit(X_train,y_train)
#LogisticRegression is used on X_train and y_test
#Fit function is generic term which is used to best match the curvature of given data p
oints.
```

```
C:\Users\Karan Singh\Anaconda3\lib\site-packages\sklearn\linear_model\logi
stic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in
0.22. Specify a solver to silence this warning.
  FutureWarning)
```

Out[123]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=Tru
e,
                   intercept_scaling=1, l1_ratio=None, 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 [131]:

```
predictions = logmodel.predict(X_test)
print(predictions)
#predict function is used to predict the values in (0 and 1)
#predict given a trained model, predict the label of a new set of data.
#This method accepts one argument
#the new data X_new , and returns the learned label for each object in the array.
```

```
[0 0 1 1 0 0 0 0 0 1 1 1 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1
 0 0 0 1 0 0 1 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1
 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 0 0 0 0
 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
 0 1 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0]
```

In [125]:

```
from sklearn.metrics import classification_report
print(classification_report(y_test,predictions))
# shows the report in which precision ,recall ,f1-score , support .
# compare the predicted X_test with y_test to find precision ,recall ,f1-score , suppor
t.
```

```
              precision    recall  f1-score   support

           0       0.79      0.93      0.86       107
           1       0.87      0.63      0.73        71

    accuracy                           0.81       178
   macro avg       0.83      0.78      0.80       178
weighted avg       0.82      0.81      0.81       178
```