I chose titanic_data to analyze.

In [27]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import seaborn as sns
#run it to make plots visible here.
%pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [3]:
```
titanic_df = pd.read_csv('titanic_data.csv') # uploaded the dataset to dataframe
```

In [4]:
```
titanic_df.describe()
```

Out[4]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

The question that I am interested in, is if there is a relationship between the age of people and their survival status.

In [5]:
```
titanic_df['Age'].isnull().values.any() # checking if there is a NaN value in "Age" variable
```

Out[5]:
```
True
```

In [6]:
```
titanic_df['Age'].isnull().sum().sum() # Wanted to know how many NaNs are there.
```

Out[6]:
```
177
```

In [7]:
```
Age = titanic_df['Age'].dropna() #Missing values were omitted
```

In [8]:
```
Age.isnull().sum().sum()
```

Out[8]:
```
0
```

In [9]:
```
Age.describe() # wanted to have a general idea about "Age" variable.
```

Out[9]:
```
count    714.000000
mean      29.699118
std       14.526497
min        0.420000
25%       20.125000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
```

In [10]:
```
# Missing values were omitted from these variables as well.
Pclass = titanic_df['Pclass'].dropna()
Survival = titanic_df['Survived'].dropna()
```

In [11]:
```
def correlation(a, b):
    a = (a-a.mean())/a.std(ddof=0)
    b = (b-b.mean())/b.std(ddof=0)
    return (a*b).mean()
```

In [12]:
```
correlation(Age, Survival)
```

Out[12]:
```
-0.077982678413863
```

Mild negative correlation, there was such a mild trend that, the older a passenger is, the less probability that he or she survived.

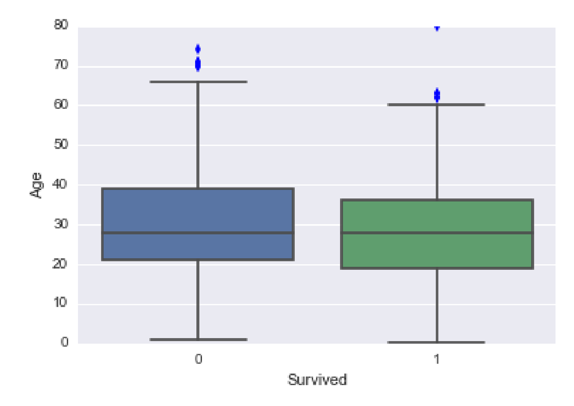
In [13]:
```
grouped_data_by_age = titanic_df.groupby('Survived')
```

In [14]:
```
grouped_data_by_age['Age'].describe()
```

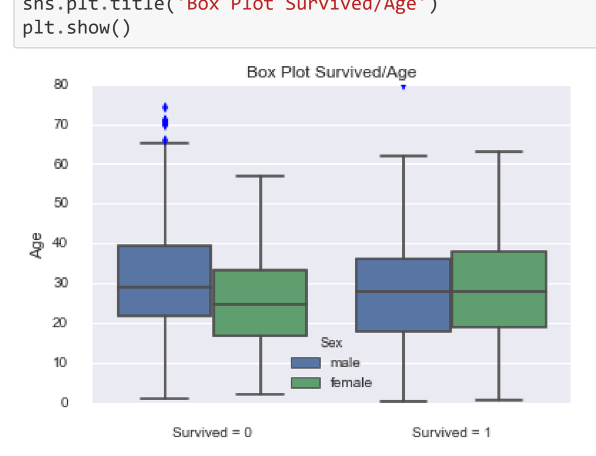
Out[14]:
```
Survived
0         count    424.000000
          mean      30.626179
          std       14.172110
          min        1.000000
          25%       21.000000
          50%       28.000000
          75%       39.000000
          max       74.000000
1         count    290.000000
          mean      28.343690
          std       14.950952
          min        0.420000
          25%       19.000000
          50%       28.000000
          75%       36.000000
          max       80.000000
dtype: float64
```

In [28]:
```
sns.boxplot(x="Survived", y="Age", data=titanic_df)
```

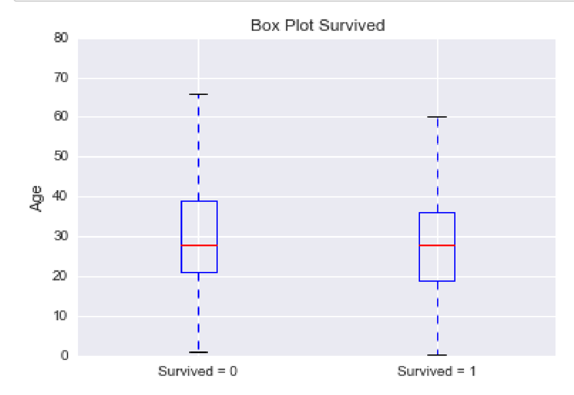
Out[28]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x20994c50>
```



In [29]:
```
fig = plt.figure()
g = sns.boxplot(x="Survived", y="Age", hue="Sex", data=titanic_df, order=[0,1])
sns.despine(offset=10, trim=True)

labels = ['Survived = 0','Survived = 1']
plt.subplots_adjust(top=0.9)
g.set_ylabel('Age')
g.set_xlabel(' ')
g.set_xticklabels(labels)
sns.plt.title('Box Plot Survived/Age')
plt.show()
```



In [30]:
```
data = titanic_df[['Age','Survived']].dropna()

fig = plt.figure()
ax = fig.add_subplot(111)

x1 = data[data['Survived'] == 0]['Age']
x2 = data[data['Survived'] == 1]['Age']

labels = ['Survived = 0','Survived = 1']

ax.boxplot([x1,x2], labels = labels)

ax.set_ylabel('Age')
ax.set_title('Box Plot Survived')
plt.legend()
plt.show()
```



There is a slight difference between age distributions based on survival status. More people of ages of 30-40 died, rather than survived. Both plots have outliers, but among people who survived there is a extreme outlier, a person at the age of 80.

As mean values show, the age played no role or very little role in the survival of passengers.

In [17]:
```
children = titanic_df[titanic_df['Age'] < 18]
```

In [18]:
```
children.describe()
```

Out[18]:

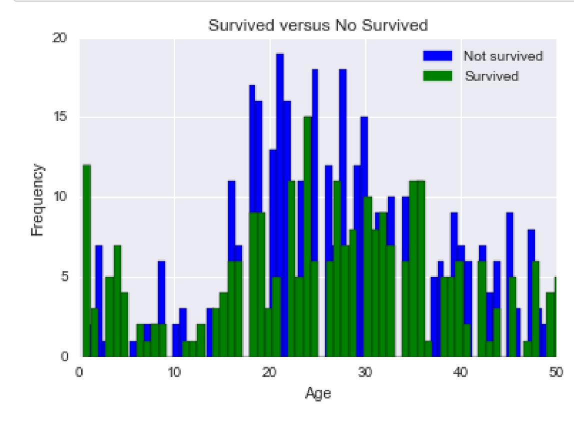
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	113.000000	113.000000	113.000000	113.000000	113.000000	113.000000	113.000000
mean	429.212389	0.539823	2.584071	9.041327	1.460177	1.053097	31.220798
std	281.743819	0.500632	0.677781	6.030408	1.625881	0.800008	32.538092
min	8.000000	0.000000	1.000000	0.420000	0.000000	0.000000	7.054200
25%	172.000000	0.000000	2.000000	3.000000	0.000000	0.000000	12.287500
50%	420.000000	1.000000	3.000000	9.000000	1.000000	1.000000	23.000000
75%	721.000000	1.000000	3.000000	16.000000	3.000000	2.000000	34.375000
max	876.000000	1.000000	3.000000	17.000000	5.000000	3.000000	211.337500

Children were distributed mainly in the second and third classes. The probability of their survival was 0.54.

I decided to use histograms to compare the age distribution among people who survived and who did not.

In [19]:
```
survived = titanic_df[titanic_df['Survived'] == 1]
no_survived = titanic_df[titanic_df['Survived'] == 0]
```

In [20]:
```
plt.figure()
plt.xlim(0, 50)
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Survived versus No Survived')
hist_no_survived = no_survived['Age'].hist(bins=100, color='b', label='no_survived')
hist_survived = survived['Age'].hist(bins=100, color='g', label='survived')
blue_patch = mpatches.Patch(color='blue', label='Not survived')
green_patch = mpatches.Patch(color='green', label='Survived')
plt.legend(handles=[blue_patch, green_patch])

plt.show()
```



Based on the histogram I can say that more people of age 20-30 died, but it does not mean that the age played a role in that. Statistical tools are needed to jump a final conclusion.

Another question that drew my attention is if the passenger class played a role in the survival of people.

In [21]:
```
grouped_data_by_pclass = titanic_df.groupby(['Pclass']) #grouped the data by the Passenger class
```

In [22]:
```
grouped_data_by_pclass['Survived'].describe() # wanted to get a general idea about the relationship between variables
#"pclass" and "survived"
```

Out[22]:
```
Pclass
1       count    216.000000
        mean       0.629630
        std        0.484026
        min        0.000000
        25%        0.000000
        50%        1.000000
        75%        1.000000
        max        1.000000
2       count    184.000000
        mean       0.472826
        std        0.500623
        min        0.000000
        25%        0.000000
        50%        0.000000
        75%        1.000000
        max        1.000000
3       count    491.000000
        mean       0.242363
        std        0.428949
        min        0.000000
        25%        0.000000
        50%        0.000000
        75%        0.000000
        max        1.000000
dtype: float64
```

This summary shows that more people from the first class survived compared to other two classes, as the mean is 0.63. The deadliest class was the third class.

In [23]:
```
grouped_data_by_pclass['Age'].describe()
```

Out[23]:
```
Pclass
1       count    186.000000
        mean      38.233441
        std       14.802856
        min        0.920000
        25%       27.000000
        50%       37.000000
        75%       49.000000
        max       80.000000
2       count    173.000000
        mean      29.877630
        std       14.001077
        min        0.670000
        25%       23.000000
        50%       29.000000
        75%       36.000000
        max       70.000000
3       count    355.000000
        mean      25.140620
        std       12.495398
        min        0.420000
        25%       18.000000
        50%       24.000000
        75%       32.000000
        max       74.000000
dtype: float64
```

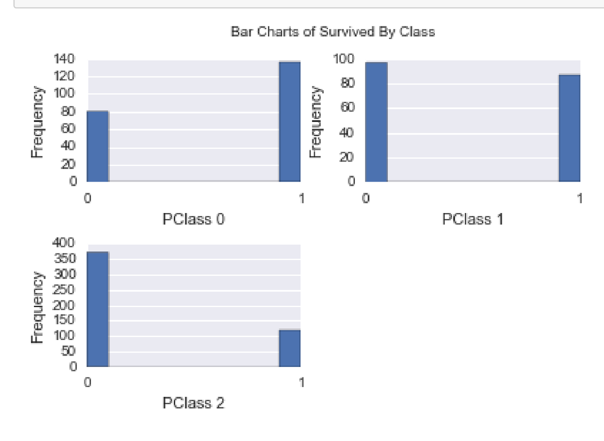
Most of people were travelling in third class and most of them were young people. Most of old passengers were travelling in first class.

In [24]:
```
correlation(Pclass, Survival)
```

Out[24]:
```
-0.33848103596101325
```

Here negative correlation is much stronger. The ranking of class played a role in survival of passengers.

In [31]:
```
axes = titanic_df['Survived'].hist(by=titanic_df['Pclass'])
xlabels = ["0","1"]

axes[0][0].set_xticks([0,1])
axes[0][0].set_xticklabels(xlabels, rotation=0)
axes[0][0].set_title(' ')
axes[0][0].set_ylabel('Frequency')
axes[0][1].set_xticks([0,1])
axes[0][1].set_xticklabels(xlabels, rotation=0)
axes[0][1].set_title(' ')
axes[0][1].set_ylabel('Frequency')
axes[1][0].set_xticks([0,1])
axes[1][0].set_xticklabels(xlabels, rotation=0)
axes[1][0].set_title(' ')
axes[1][0].set_ylabel('Frequency')
axes[1][1].set_xticks([0,1])
axes[1][1].set_xticklabels(xlabels, rotation=0)
axes[1][1].set_title(' ')
axes[1][1].set_ylabel('Frequency')
axes[0][0].set_xlabel('PClass 0')
axes[0][1].set_xlabel('PClass 1')
axes[1][0].set_xlabel('PClass 2')
axes[1][1].set_xlabel('PClass 3')

plt.suptitle("Bar Charts of Survived By Class")

plt.show()
```



Obviously, more people from fisrt class survived. People of third class were not that lucky, may be because of the location of their rooms in the ship.

Handling of missing values present limitations to the analysis depending on what is choosen. Based on biased assumptions I did not choose other variables to investigate the surviability. I thought sex, fare, cabin could not play a role in surviability. Lack of other variables also produces limitations.

In [ ]: