In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
import sklearn
from pandas import Series, DataFrame
from pylab import rcParams
from sklearn import preprocessing
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
```

In [3]:

```
url= "https://raw.githubusercontent.com/BigDataGal/Python-for-Data-Scie
nce/master/titanic-train.csv"
titanic = pd.read_csv(url)
titanic.columns = ['PassengerId','Survived','Pclass','Name','Sex','Age'
,'SibSp','Parch','Ticket','Fare','Cabin','Embarked']
```

In [4]:

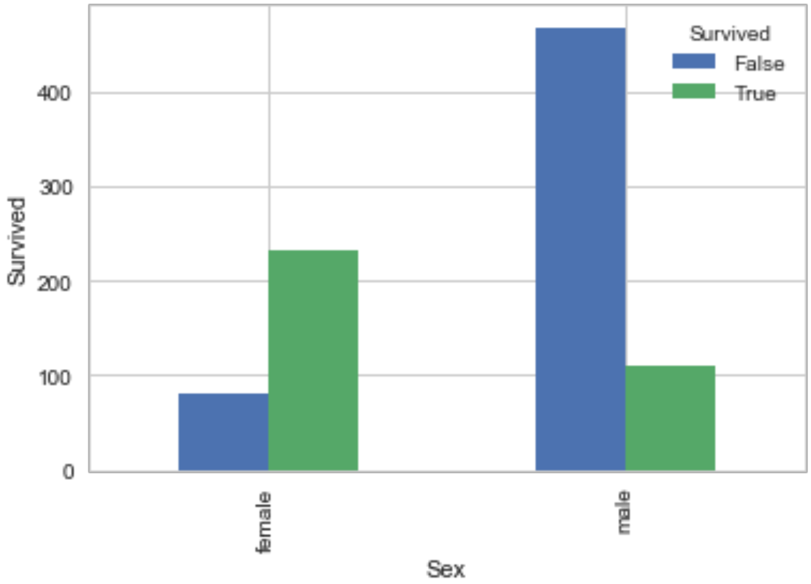
```
titanic.head()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05

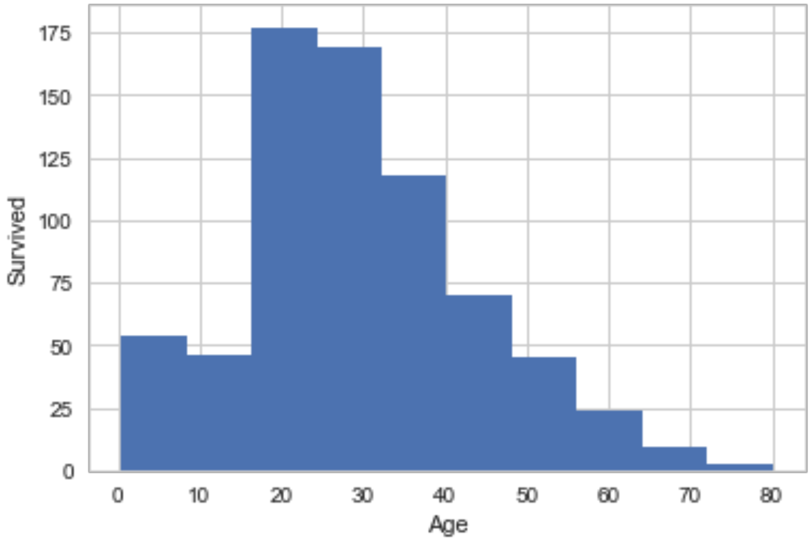
```
In [5]: import matplotlib.pyplot as plt
        import seaborn as sns
        sns.set_style('whitegrid')
        # show plots in the notebook
        %matplotlib inline


        pd.crosstab(titanic.Sex, titanic.Survived.astype(bool)).plot(kind='bar'
        )
        plt.xlabel('Sex')
        plt.ylabel('Survived')
```

```
Out[5]: <matplotlib.text.Text at 0x111a0cc88>
```



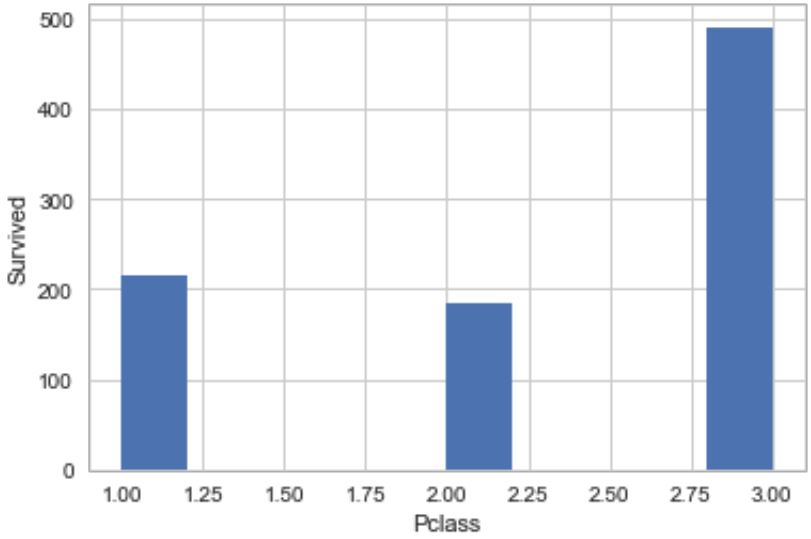
```
In [6]: titanic.Age.hist()
        plt.xlabel('Age')
        plt.ylabel('Survived')
```

```
Out[6]: <matplotlib.text.Text at 0x1109bf5c0>
```



```
In [7]: titanic.Pclass.hist()
        plt.xlabel('Pclass')
        plt.ylabel('Survived')
```

```
Out[7]: <matplotlib.text.Text at 0x114d77dd8>
```

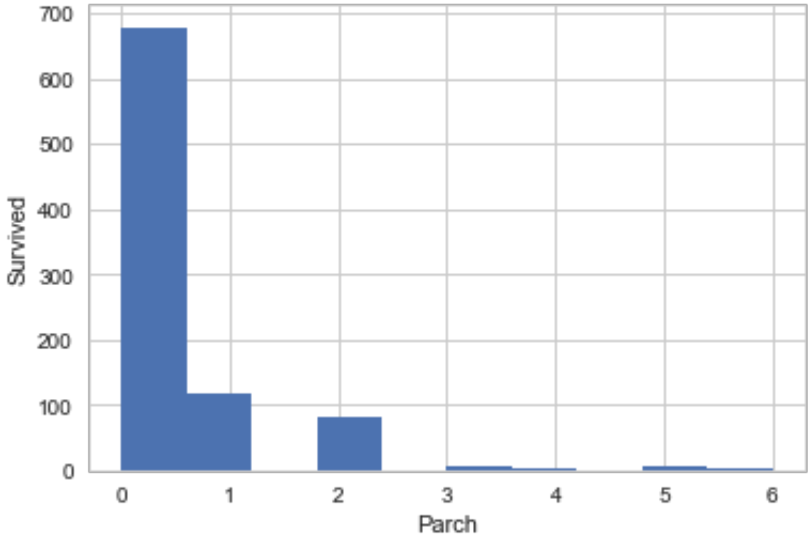


In [8]:

```
titanic.Parch.hist()
plt.xlabel('Parch')
plt.ylabel('Survived')
```

Out[8]:

```
<matplotlib.text.Text at 0x114e33ac8>
```

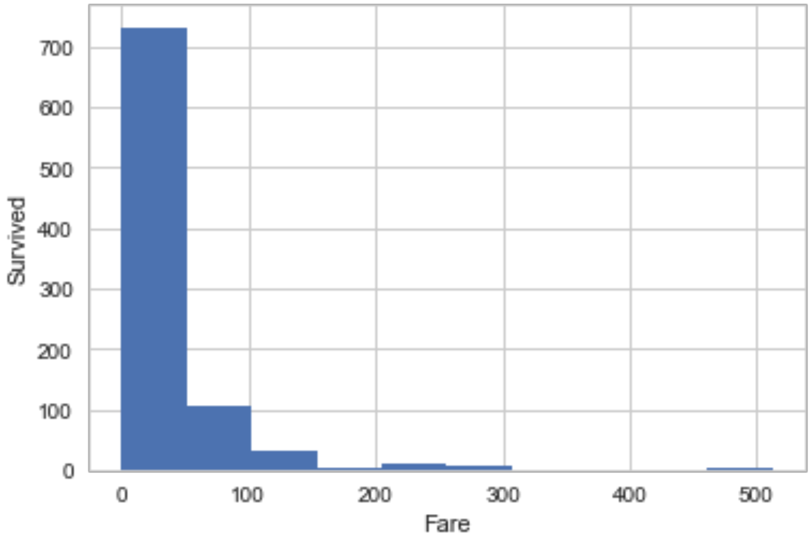


In [9]:

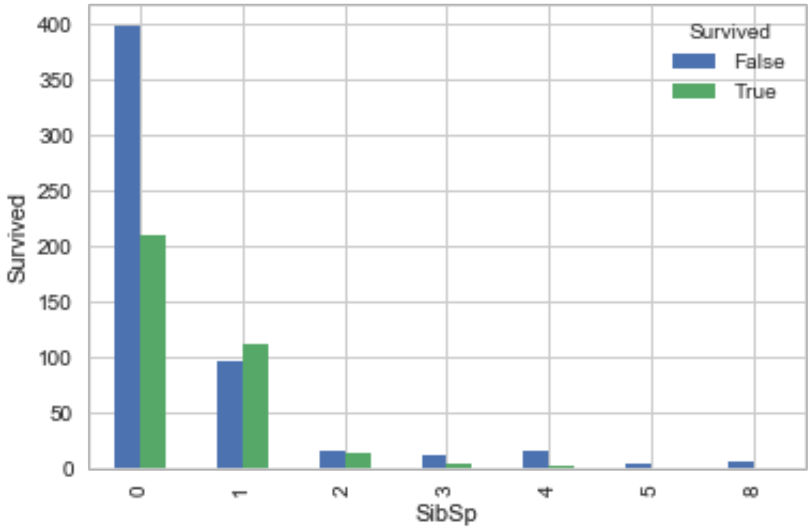
```
titanic.Fare.hist()
plt.xlabel('Fare')
plt.ylabel('Survived')
```

Out[9]:

```
<matplotlib.text.Text at 0x114f5ff60>
```



In [10]:
```
pd.crosstab(titanic.SibSp, titanic.Survived.astype(bool)).plot(kind='ba
r')
plt.xlabel('SibSp')
plt.ylabel('Survived')
plt.show()
```



In [11]:
```
titanic_data = pd.get_dummies(data= titanic, columns=['Sex'])
titanic_data.head()
```

Out[11]:

	PassengerId	Survived	Pclass	Name	Age	SibSp	Parch	Ticket	Fare	Ca
0	1	0	3	Braund, Mr. Owen Harris	22.0	1	0	A/5 21171	7.2500	Na
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1	0	PC 17599	71.2833	C8
2	3	1	3	Heikkinen, Miss. Laina	26.0	0	0	STON/O2. 3101282	7.9250	Na
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	0	113803	53.1000	C1
4	5	0	3	Allen, Mr. William Henry	35.0	0	0	373450	8.0500	Na

In [13]:

```
DT = DecisionTreeClassifier()
```

In [14]:

```
X = titanic_data[['Pclass','Sex_female','Sex_male', 'Age', 'SibSp', 'Pa
rch', 'Fare']]
Y = titanic_data.Survived
```

In [15]:

```
titanic_data.fillna('0', inplace=True)
```

In [21]:

```
DT.fit(X, Y)
```

In [22]:

```
scoring = "accuracy", # Scoring metric
Y_pred = DT.predict(X)
```

In [23]:

```
Y_pred[:5]
```

In [19]:

```
from sklearn.metrics import accuracy_score
```

In [27]:
```
accuracy_score(Y, Y_pred)
```

In [24]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3
,random_state =30)
```

In [26]:
```
DT.fit(X_train, Y_train)
```

In [ ]:
```
pred_DT_train = DT.predict(X_train)
pred_DT_test = DT.predict(X_test)
```

In [ ]:
```
pred_DT_test[:5]
```

In [ ]:
```
print(accuracy_score(Y_train, pred_DT_train))
print(accuracy_score(Y_test, pred_DT_test))
```

In [ ]:
```
LR = LogisticRegression()
LR.fit(X_train, Y_train)
```

In [ ]:
```
pred_LR_train = LR.predict(X_train)
pred_LR_test = LR.predict(X_test)
```

In [ ]:
```
print(accuracy_score(Y_train, pred_LR_train))
print(accuracy_score(Y_test, pred_LR_test))
```

In [ ]:
```
from sklearn.metrics import confusion_matrix
print(confusion_matrix(Y_test, pred_DT_test))
print('**************************************')
print(confusion_matrix(Y_test, pred_LR_test))
```

In [ ]:
```
print("Report with Decision Tree")
print(classification_report(Y_test, pred_DT_test))
print("******************************************")
print("Report with Logistic Regrssion")
print(classification_report(Y_test, pred_LR_test))
```

In [ ]:
```
pd.crosstab(Y_test, pred_DT_test)
```

In [ ]:
```
from sklearn.cross_validation import cross_val_score
```

In [ ]:

```
scores = cross_val_score(estimator= DT, # Model to test
                         X= X, y = Y, # Target variable
                         scoring = "accuracy", # Scoring metric
                         cv=10) # Cross validation folds
print("Accuracy per fold: ")
print(scores)
print("Average accuracy: ", scores.mean())
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: