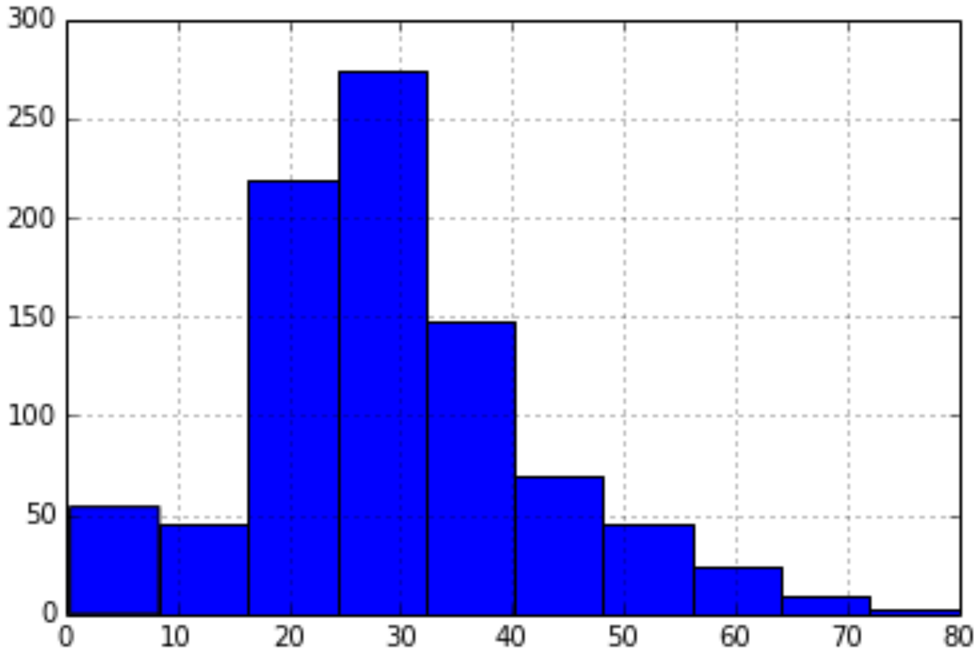
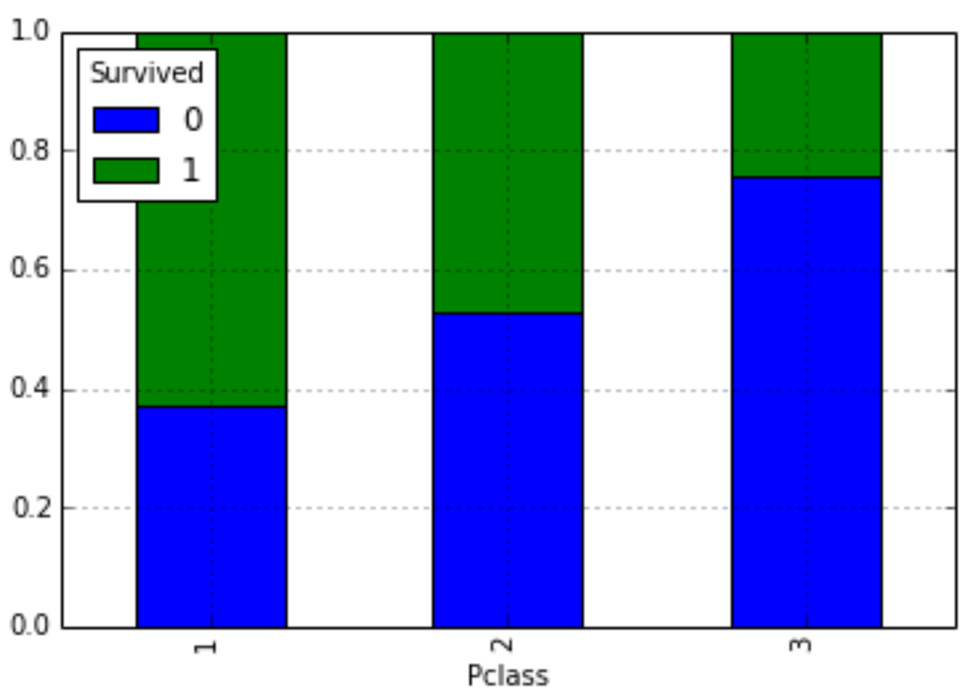
```
# -*- coding: utf-8 -*-
"""
Created on 06/17/2015

@author: Wenjun Song

The cleaning data for Titanic Project for MML2015
"""

## Here in this project, we'll use pandas, numpy, matplotlib
from pandas import Series,DataFrame
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


## First load all the data in
trainDf=pd.read_csv('train.csv',sep=',')

## Create a value absed Gender column
trainDf['Gender']=1
trainDf['Gender'] = trainDf['Sex'].map( {'female': 0, 'male': 1} ).astype(int)

## Fill missing values in age column
## Calculate the median age for each gender and each class
trainDf['AgeFilled']=trainDf['Age']
median_ages = np.zeros((2,3))
for i in range(0, 2):
    for j in range(0, 3):
        median_ages[i,j] = trainDf[(trainDf['Gender'] == i) & \
                           (trainDf['Pclass'] == j+1)]['Age'].dropna().median()
## Fill the missing data with the corresponding median we calculated
for i in range(0, 2):
    for j in range(0, 3):
        trainDf.loc[ (trainDf.Age.isnull()) & (trainDf.Gender == i) & (trainDf.Pclass == j+1),\
                'AgeFilled'] = median_ages[i,j]
trainDf['AgeFilled'].hist()
plt.show()
```



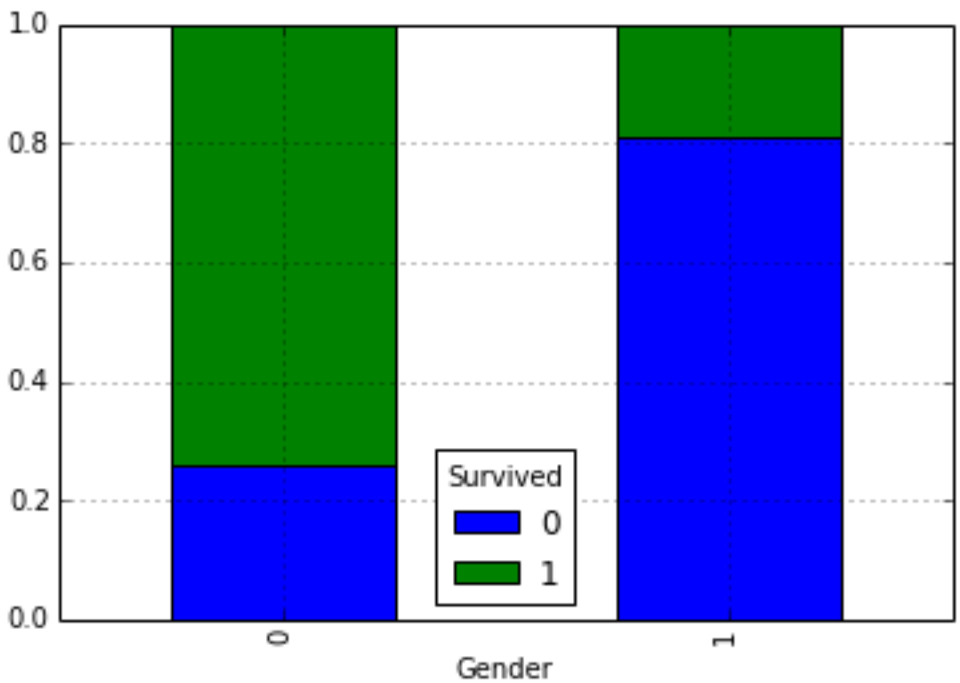

```
## Create a column combine the product of age and class
trainDf['Age*Class'] = trainDf.AgeFilled * trainDf.Pclass

## Now we start to identify important factors
## 1. Class
class_counts = pd.crosstab(trainDf['Pclass'],trainDf['Survived'])
class_pcts=class_counts.div(class_counts.sum(1).astype(float),axis=0)
class_pcts.plot(kind='bar',stacked=True)
plt.show()
```



We can see a clear trend that the higher class tend to have more chance to survive than the low class

2. Sex

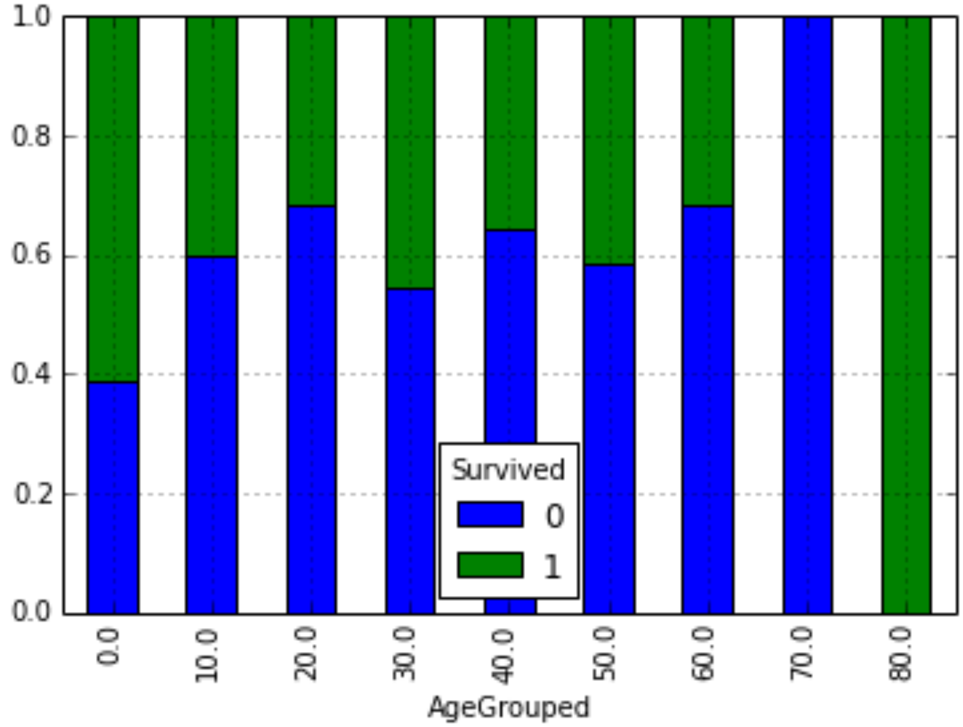
```
gender_counts = pd.crosstab(trainDf['Gender'],trainDf['Survived'])
gender_pcts=gender_counts.div(gender_counts.sum(1).astype(float),axis=0)
gender_pcts.plot(kind='bar',stacked=True)
plt.show()
```



We can see a clear trend that the women tend to have much more chance to survive than the men

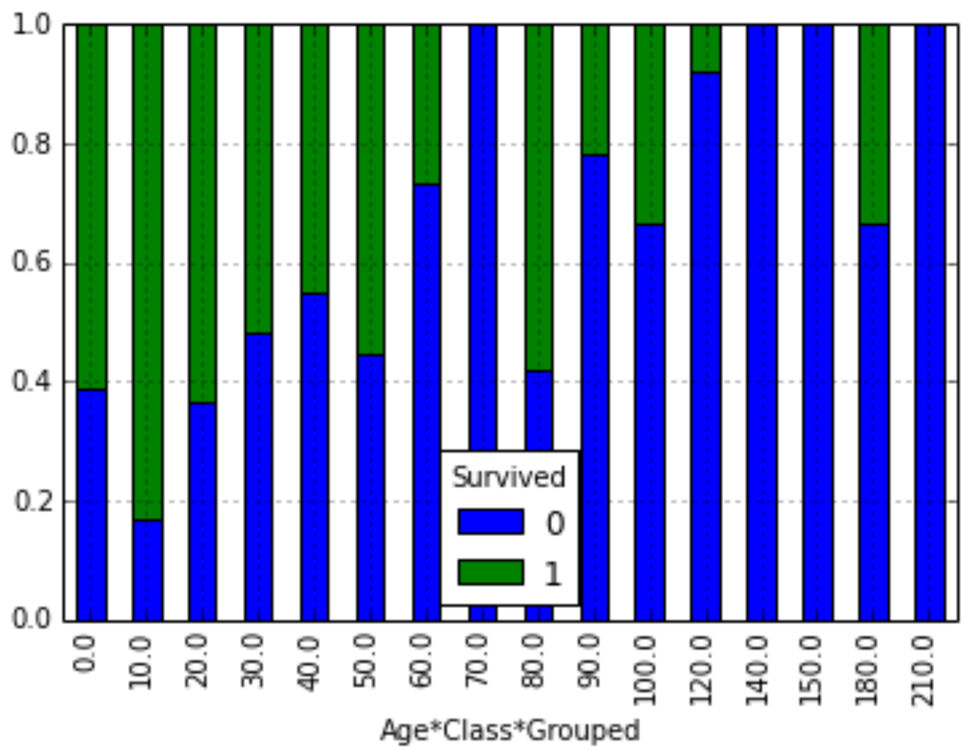
3. Age

```
bucket_size=10
trainDf['AgeGrouped']=np.floor(trainDf['AgeFilled']/bucket_size)*bucket_size
age_counts = pd.crosstab(trainDf['AgeGrouped'],trainDf['Survived'])
age_pcts=age_counts.div(age_counts.sum(1).astype(float),axis=0)
age_pcts.plot(kind='bar',stacked=True)
plt.show()
max_age=Series(trainDf[‘AgeFilled']).max(axis=1)
```



According to the graph, I don't see a clear trend that age is directly related to the survival chance.
But there might be some noice due to filling method of the missing data

4. Therefore, let's look at 'Age*Class' to see if this can give us anything
trainDf['Age*Class*Grouped'] = trainDf.AgeGrouped * trainDf.Pclass
age_m_class_counts = pd.crosstab(trainDf['Age*Class*Grouped'],trainDf['Survived'])
age_m_class_pcts=age_m_class_counts.div(age_m_class_counts.sum(1).astype(float),axis=0)
age_m_class_pcts.plot(kind=‘bar',stacked=True)



There's a rough trend that with small Age*Class tend to have high survival probability, but it is not a
absolute trend.

Save all the cleaned files to pickle files
trainDf.to_pickle('train_pickle')

In conclusion, we can see a clear trend of relationship between class,sex and survival probability.
No clear trend is available for other factors, but we can still dig deeper, and might find some other factors make sense.