Investigate a Dataset (Titanic Data)

Project 2: Data Analyst Nanodegree

Jorge Santamaría Cruzado

December 1, 2015

1 Project goals

In this project, I've chosen to analyze a dataset containing passenger information on board the Titanic. By doing that, I expect to answer this questions:

1. What were the demographics of the passengers?
2. How much did the passengers pay for their ticket?
3. Which were factors that make the passenger survive or not the accident?

2 Data overview

There are 891 entries on the csv file. Knowing the Titanic had 2224 passengers on board the day it sink, it looks like this dataset is only a small extract of the full data.

After loading it on a pandas dataframe, I see it has twelve columns:

1. **PassengerId:** The unique ID number of the passenger
2. **Survived**: 1 if the passenger survived the accident, 0 otherwise
3. **Pclass**: Class where the passenger was traveling, values 1, 2 and 3
4. **Name**: Name of the passenger
5. **Sex**: Sex of the passenger
6. **Age**: Age of the passenger
7. **SibSp**: Number of siblings/spouses for the passenger
8. **Parch**: Parent/children of the passenger
9. **Ticket**: Ticked number of the passenger
10. **Fare**: Fare paid by the passenger
11. **Cabin**: Assigned cabin
12. **Embarked**: Port where the passenger embarked

First thing I will do is a fast cleaning of the data for better handling, I don't see the point on having an index starting by 0 on the dataframe, because PassengerId can be used for the same thing and it looks more useful, so I will replace the DataFrame index for the PassengerId column.

I don't like either the Survived column being an integer, although in python booleans and integers can work the same way, I think it's a cleaner approach to transform this column into boolean values.

Finally, I will replace the embarked port abbreviations for the full names.

```
def fix_port(name):
    ports = {'C': 'Cherbourg', 'Q': 'Queenstown',
             'S': 'Southhampton'}
    if name in ports.keys():
        return ports[name]
    return name

> df = df.set_index('PassengerId')
> df['Survived'] = df['Survived'].apply(lambda x: True if x else False)
> df['Embarked'] = df['Embarked'].apply(fix_port)
```

This will make the dataframe a little easier to work with and I will avoid some extra code on the plotting functions.

It is also useful to check for columns with missing values:

```
>>> df.isnull().any()
Survived    False
Pclass      False
Name        False
Sex         False
Age          True
SibSp       False
Parch       False
Ticket      False
Fare        False
Cabin        True
Embarked     True
dtype: bool

>>> df = df.dropna(subset=['Age'])
```

Looks like columns Age, Cabin and Embarked all have some missing values.

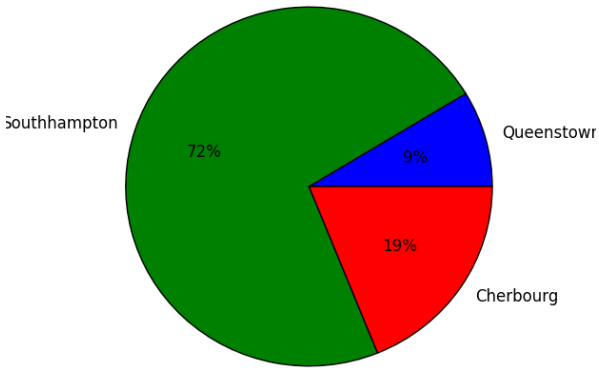
While I don't care about Cabin column in this anaylisis, I will remove rows with a missing Age value, and handle Embarked missing values later, when ploting the data.

3 Answers

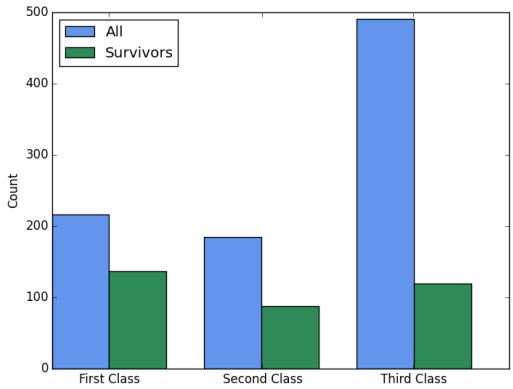
3.1 Demographics

First thing I'm curious about is where did the passengers came from, I draw a pie plot of the ports where they embarked and it shows that most of the passengers came from Southhampton. 1a

Another interesting plot will be to draw the histogram of the passenger classes and then compare it to the classes of the ones that survived. Looking at the plot, 1b it also looks that there is a relationship between the class of the passenger and his chances to survive.



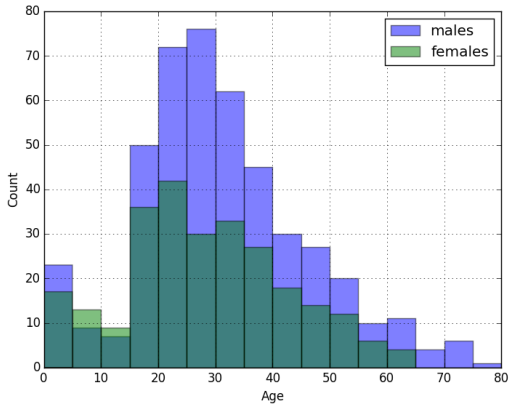
(a) Embarked port



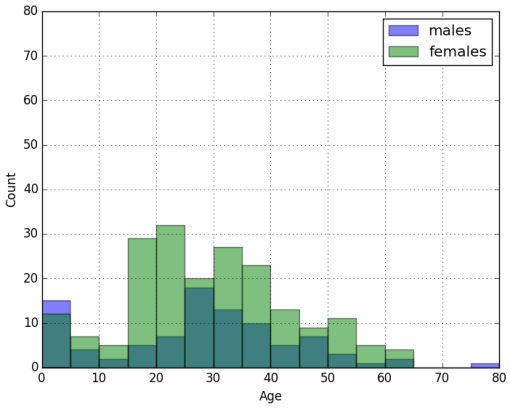
(b) Class

Then I'm drawing a histogram of the passenger ages and then compare it with the histogram of ages of the passengers that survived the accident. 2a

By looking at it, it looks like the females had a much higher chance to survive the accident, while only a few males survived. 2b



(a) Passenger Ages



(b) Survivor Ages

As shown on figure 2a, it looks like most of the passengers were in the 15 - 50 age range. Also, there were more males than females, although after the accident, most of the survivors were females.

Getting some fast statistics with pandas:

```
>>> df[['Age', 'SibSp', 'Parch', 'Fare']].describe()
              Age       SibSp       Parch        Fare
count  714.000000  714.000000  714.000000  714.000000
mean    29.699118    0.512605    0.431373   34.694514
std     14.526497    0.929783    0.853289   52.918930
min      0.420000    0.000000    0.000000    0.000000
25%     20.125000    0.000000    0.000000    8.050000
50%     28.000000    0.000000    0.000000   15.741700
75%     38.000000    1.000000    1.000000   33.375000
max     80.000000    5.000000    6.000000  512.329200

>>> len(df[df['Sex']=='male'])
453

>>> len(df[df['Sex']=='female'])
261
```

The average age was 29.7 years with an standard deviation of 14.5 years and there are 453 males and 261 females on my dataset.

3.2 Fares

The average fare paid was 34.7$ but there was a big standard deviation of 52.9$ so the prices paid differ greatly between passengers. Plotting a box plot and some histograms of the fares between classes shows even more information. 3

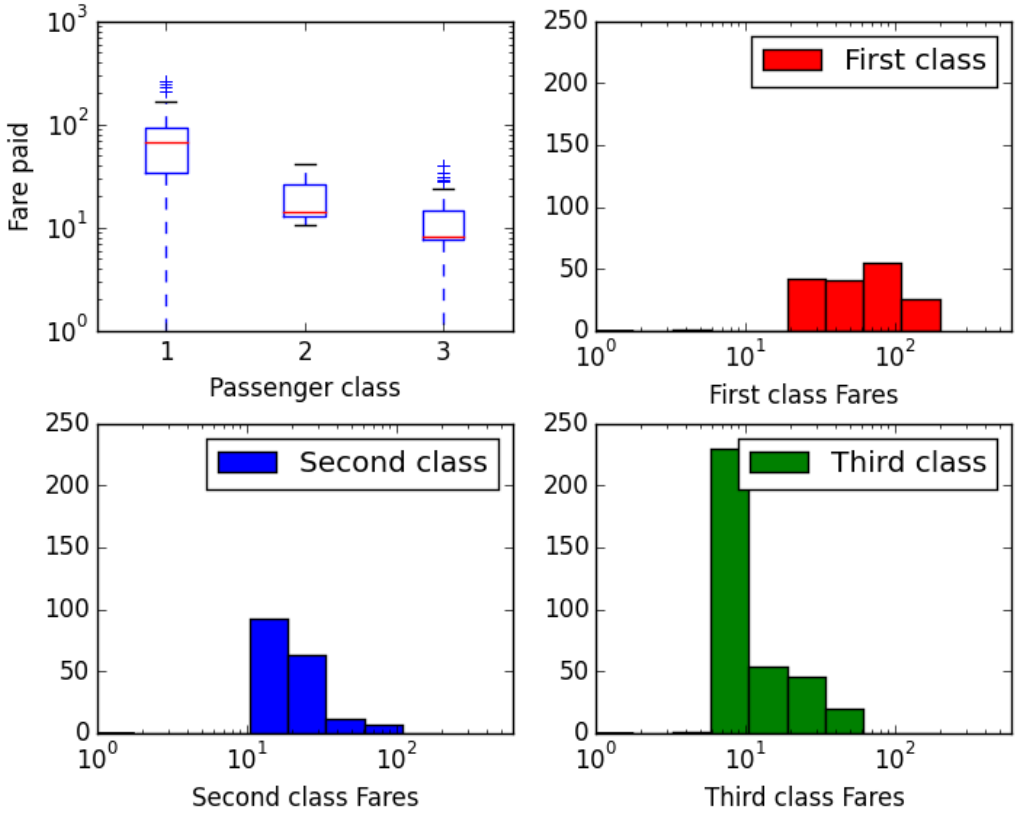


Figure 3: Fares histogram

It's surprising the prices paid for second and third class are almost the same and there's also some passengers on first class that payed similar quatities than a few on third or second class.

Searching for some info on the internet about this, it looks like classes were assigned not based on the ticket price, but based on the job or prestige of the passenger.

A simple way to check if the fare paid was related to the survival rate is to calculate Pearson's r between the two series:

```
>>> df['Fare'].corr(df['Survived'], method='pearson')
0.26818861687447715
```

It looks like there is a significant correlation between the two variables, but I will get more into it in the next subsection.

To dig a little more into the wide difference of fares between passengers on the same class I will draw a scatterplot of the fare paid against the age of the passenger to see if there is some relation. 4

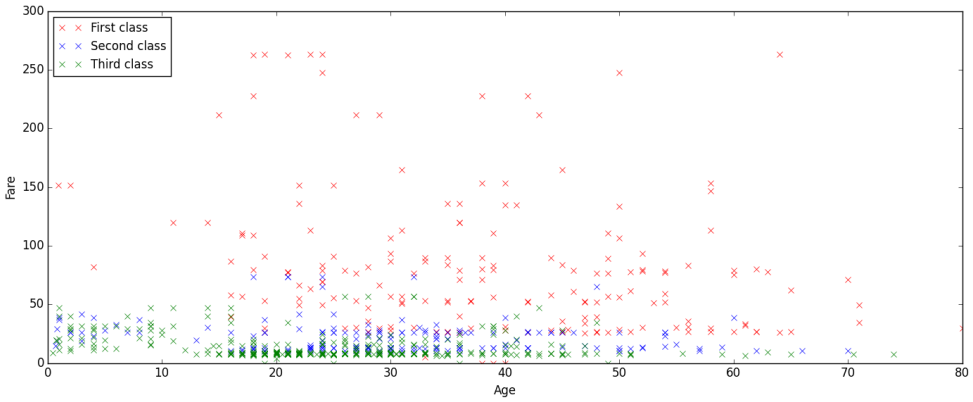


Figure 4: Fare vs Age by Pclass

To the naked eye it doesn't look like there is any relation there. I will calculate the correlation numericaly to check it out:

```
# First class
>>> df[df['Pclass']==1].corr()['Age']['Fare']
-0.2186107964549317

# Second class
>>> df[df['Pclass']==2].corr()['Age']['Fare']
-0.19703835771598235

# Third class
>>> df[df['Pclass']==3].corr()['Age']['Fare']
-0.26031469825905318
```

Surprisingly around 20% of the variability on the First class and Second class fares can be explained by the age of the passenger, while arount 25% of the variability on the Third class fares are explained by that column.

3.3 Survival factors

Although it looks clear comparing figures 2a and 2b that females had a higher chance of surviving the accident, I will now calculate the proportion numerically.

```
# male survival proportion
    >>> (df[df['Sex'] == 'male']['Survived'].sum() /
            len(df[df['Sex'] == 'male']))
    0.20529801324503311

# female survival proportion
    >>> (df[df['Sex'] == 'female']['Survived'].sum() /
            len(df[df['Sex'] == 'female']))
    0.75478927203065138

# overall survival proportion
    >>> df['Survived'].sum() / len(df)
    0.4061624649859944
```

It is simpler to look at the survival rates in a pie plot 5

Running a simple χ^2 test with the observed frequencies of the surviving females and males. The null hypothesis will be that there is no significant difference between the survival frequencies of males and females. It yields:

$$\chi^2 = 207, p \simeq 0$$

It is clear that the observed values don't fit into the expected values if the null hypothesis were correct, so there is a significant relation between the gender of the passenger and his chances of survival.

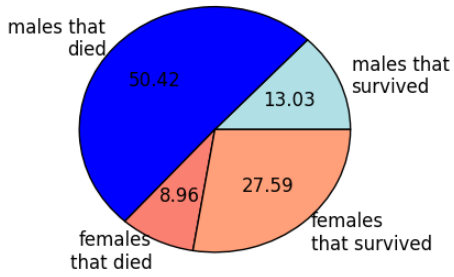


Figure 5: Survival rates

By doing the same thing comparing the classes of the passengers instead of their gender, I got:

$$\chi^2 = 93, p \simeq 0$$

So there is also a significant relation between the class the passenger was and his chances of survival, as if was shown on the histogram 1b.

Anyway, the χ^2 test doesn't give much information about the strenghts of the relationships between the variables (this numbers are related to the sample sizes, it just shows there is a relation) so I will complement it calculating Pearson's r, which will give me a lot more information.

To discover what are the most important variables related to passenger survival I will simply check the correlation with this columns as I did on the Fares subsection, calculating Pearson's r.

To do this, first I will convert the sex of the passenger to a numeric value so Pearson's r can be calculated.

```
>>> from scipy.stats import pearsonr
>>> df2 = df[['Sex', 'Age', 'Pclass', 'Fare', 'SibSp', 'Parch']]
>>> df2['Sex'] = df2['Sex'].apply(lambda x: 1 if x == 'male' else 0)
>>> df2.apply(lambda x: pearsonr(x, df['Survived']))
Sex         (-0.538825593015, 5.22470992681e-55)
Age          (-0.0772210945722, 0.0391246540135)
Pclass      (-0.359652682087, 3.16210354167e-23)
Fare         (0.268188616874, 3.15599457049e-13)
SibSp         (-0.0173583604795, 0.643327731112)
Parch         (0.0933170077422, 0.0126106500391)
dtype: object
```

With an α level of 0.05 all this factors except SibSp are significantly correlated to the passenger survival. Lowering the α level to 0.01, the most significant factors are Sex, Pclass and Fare.