

P2 - Investigate a Dataset

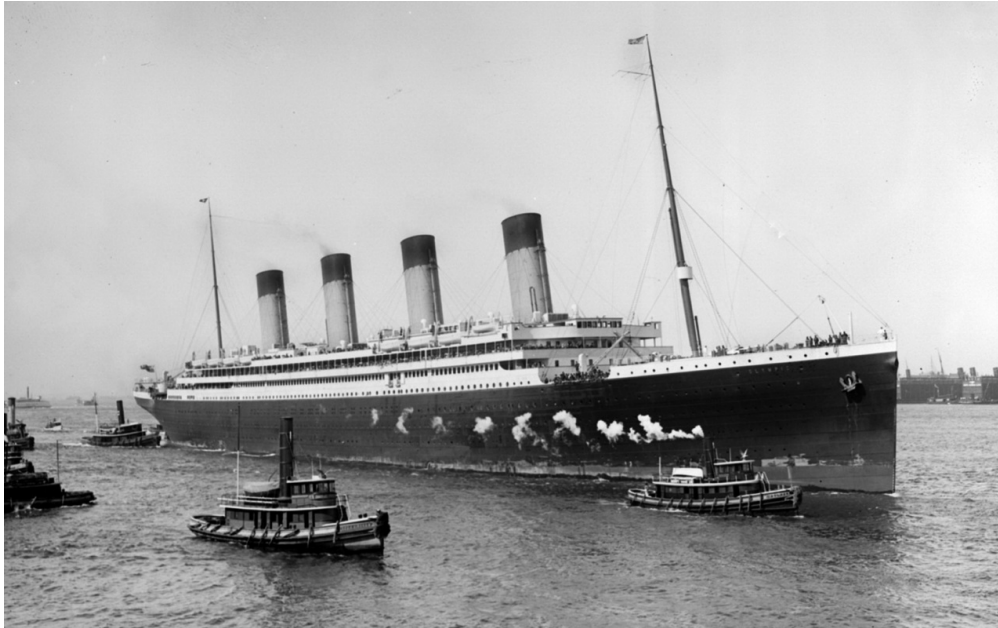
As part of P2-Investigate a Dataset, I chose to investigate **Titanic dataset**.

Index

- Section 1: Overview
 - Brief History
 - Dataset Information
- Section 2: Questions
- Section 3: Data Wrangling
 - Understanding the data
 - Data cleaning
- Section 4: Data Exploration
 - Analyzing Relationships
 - Investigating Survival Factors
- Section 5 : Limitation of the analysis
- Section 6 : Statistical Tests
- Section 7 : Conclusion
- Section 8 : References

SECTION 1 : OVERVIEW

Brief History



Titanic was a British passenger liner that sank in the North Atlantic Ocean in the early morning of 15 April 1912, after colliding with an iceberg during her maiden voyage from Southampton to New York City. Of the 2,224 passengers and crew aboard, more than 1,500 died, making it one of the deadliest commercial peacetime maritime disasters in modern history. The British registered ocean liner was built in 1912 for the transatlantic passenger and mail service between Southampton and New York. Constructed at the Harland and Wolff shipyard in Belfast, Ireland the Titanic was, on her maiden voyage, the largest vessel afloat. Thomas Andrews, her architect, died in the disaster.

Although Titanic had advanced safety features such as watertight compartments and remotely activated watertight doors, there were not enough lifeboats to accommodate all of those aboard due to outdated maritime safety regulations. Titanic only carried enough lifeboats for 1,178 people—slightly more than half of the number on board, and one third of her total capacity. After leaving Southampton on 10 April 1912, the new liner sailed from Southampton, England and on 14 April, four days into the crossing and about 375 miles (600 km) south of Newfoundland, she hit an iceberg at 11:40 p.m. ship's time. The collision caused the ship's hull plates to buckle inwards along her starboard side and opened five of her sixteen watertight compartments to the sea; the ship gradually filled with water. Meanwhile, passengers and some crew members were evacuated in lifeboats, many of which were launched only partially loaded. A disproportionate number of men were left aboard because of a "women and children first" protocol for loading lifeboats.[3] At 2:20 a.m., she broke apart and foundered—with well over one thousand people still aboard. Just under two hours after Titanic sank, the Cunard liner RMS Carpathia arrived at the scene, where she brought aboard an estimated 705 survivors.

It has rightly been called as the greatest shipwreck drama of the 20th century, a catastrophic event which has been greeted with worldwide shock and outrage at the huge loss of life and the regulatory and operational failures that had led to it.

Dataset Information

The dataset contains demographics and passenger information from **891** of the **2224 passengers and crew on board the Titanic**.

VARIABLE DESCRIPTIONS:

survival	Survival (0 = No; 1 = Yes)
pclass	Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
name	Name
sex	Sex
age	Age
sibsp	Number of Siblings/Spouses Aboard
parch	Number of Parents/Children Aboard
ticket	Ticket Number
cabin	Cabin
embarked	Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

SPECIAL NOTES:

- Pclass is a proxy for socio-economic status (SES)
1st ~ Upper; 2nd ~ Middle; 3rd ~ Lower
- Age is in Years; Fractional if Age less than One (1)
If the Age is Estimated, it is in the form xx.5
- With respect to the family relation variables (i.e. sibsp and parch) some relations were ignored. The following are the definitions used for sibsp and parch.

Sibling: Brother, Sister, Stepbrother, or Stepsister of Passenger Aboard Titanic

Spouse: Husband or Wife of Passenger Aboard Titanic (Mistresses and Fiances Ignored)

Parent: Mother or Father of Passenger Aboard Titanic

Child: Son, Daughter, Stepson, or Stepdaughter of Passenger Aboard Titanic

Other family relatives excluded from this study include cousins, nephews/nieces, aunts/uncles, and in-laws. Some children travelled only with a nanny, therefore parch=0 for them. As well, some travelled with very close friends or neighbors in a village, however, the definitions do not support such relations.

SECTION 2: QUESTIONS

Questions I hope to investigate in this report

Q1. Do individual passenger traits like

- age
- gender
- age group

impact their survival?

Q2. Did the passenger travelling condition namely

- class
- port of embarkation

had an impact on their survival? Did individual passenger traits had any relationship with this?

SECTION 3: DATA WRANGLING

Understanding the data set

Imports

```
In [1]: %pylab inline
import pandas as pd
import seaborn as sns
import numpy as np
```

Populating the interactive namespace from numpy and matplotlib

We read the titanic-data.csv into panda data frame(**titanic_df**) and print the top 5 head records to have a overview of the data.

```
In [15]: titanic_df = pd.read_csv('titanic-data.csv')
titanic_df.head()
```

Out[15]:

	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	7.25
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

When I dived deeper into this data to understand the data formats and for the purposes of cleaning this data, I found that out of **891 passenger records** available:

- Age of 177 passengers is missing
- Cabin details of only 204 passengers are available and of 607 are missing
- Port of embarkation of 2 passengers is missing

Below are the some of the general statistics about the data set

```
In [16]: titanic_df.info()
titanic_df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId      891 non-null int64
Survived         891 non-null int64
Pclass           891 non-null int64
Name             891 non-null object
Sex              891 non-null object
Age             714 non-null float64
SibSp            891 non-null int64
Parch           891 non-null int64
Ticket           891 non-null object
Fare            891 non-null float64
Cabin           204 non-null object
Embarked         889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.6+ KB
```

Out[16]:

	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	32.200000
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.690000
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	7.910000
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.450000
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	512.000000

Data Cleaning

To handle and clean the missing data, I would be refining the data set as follows:

- I would not be using **cabin** details of the passengers for my analysis, so I chose to drop this column from my data frame.
- As for the missing **port of embarkation**, I chose to ignore these passengers as they are just a 0.224 proportion of the data set.
- Since I would be investigating the passengers based on their **Age** as well, I chose to fill 0 for the missing values so that any wrong ages do not impact my analysis. Also 0 is chosen as existing passenger ages do not have passengers with age=0

```
In [17]: # drop column for 'Cabin' from the data set
titanic_data_frame= titanic_df.drop('Cabin',1)
```

```
In [18]: #Passengers with age=0 in existing data set
titanic_df[titanic_df['Age']==0]['PassengerId'].count()
```

Out[18]: 0

```
In [19]: #Replace NA values for age with 0
titanic_data_frame['Age'].fillna(value=0,inplace=True)
titanic_data_frame.head()
```

Out[19]:

	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	7.25
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

The United Nations Convention on the Rights of the Child (UNRC) defines children, for the purposes of the Convention, as persons under the age 18, unless domestic legislation provides otherwise. In that spirit, this timeline includes as children all those below the UK age of majority, which was 21 until 1970, when it was reduced to 18[1]

So following this convention I would consider passengers with age between 0.1-21 years, adults with age between 18-59 and elderly with age greater than 60.

I would add like to add another column for identifying passengers based on their ages in the following format where 0 is customly added by me above:

Age	Age classification
0	NA
Between 0.1 and 21	Children
Greater than equal to 21 but less than 60	Adult
Greater than equal to 60	Elderly

Implementing this age classification on the current titanic data set available:

```
In [20]: def ageClassification(passenger):  
    ##Ignore passengers with age equal to 0  
    if passenger['Age']==0:  
        return 'NA'  
    if passenger['Age']>=0.1 and passenger['Age']<21:  
        return "Child"  
    elif passenger['Age']>=21 and passenger['Age']<60:  
        return "Adult"  
    else:  
        return "Elderly"
```



```
In [21]: titanic_data_frame['AgeGroup']=titanic_data_frame.apply(ageClassification,axis
)
titanic_data_frame.head()
```

Out[21]:

	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	7.25
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

Woman and Child First

Since there was capacity to fit in 1,178 passengers in the lifeboat from around 2,228 people onboard the Titanic[3], the Second Officer suggested to Captain Smith, "Hadn't we better get the women and children into the boats, sir?", to which the captain responded: "women and children in and lower away". The First (Officer Murdoch) and Second (Officer Lightoller) officers interpreted the evacuation order differently; one took it to mean women and children first, while the other took it to mean women and children only. Thus one of the officers lowered lifeboats with empty seats if there were no women and children waiting to board, while the other allowed a limited number of men to board if all the nearby women and children had embarked.[4]

So I want to add an additional column to the data frame for identifying whether the passenger was Woman/Child. So I would add another column to the data frame named WomanChild with value "Yes" for female passengers or child passengers and for male passengers, I would put the value "No" in this column

```
In [22]: # Group passengers on basis on Woman/Child
def groupWomanChild(data):
    if data['AgeGroup']=='Child' or data['Sex']=='female':
        woman_child='Yes'
    else:
        woman_child='No'
    return woman_child;
```

```
In [23]: #Apply groupWomanChild function to the titanic data frame
titanic_data_frame['WomanChild'] = titanic_data_frame.apply(groupWomanChild, a
xis=1)
titanic_data_frame.head()
```

Out[23]:

	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	7.25
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

For the data set I would like to create more **descriptive labels** which would help me later in the visualization.

I would like to add a Survival Status column with values **Survived/Died** corresponding to "1/0" values for Survived column for making my visualizations more readable.

```
In [24]: # Create Survived Label Column
titanic_data_frame['Survival_Status'] = titanic_data_frame.Survived.map({0 :
'Died', 1 : 'Survived'})
titanic_data_frame.Survival_Status.head()
```

```
Out[24]: 0      Died
1     Survived
2     Survived
3     Survived
4      Died
Name: Survival_Status, dtype: object
```

I would like to add descriptive column for passenger class with values **First Class**, **Second Class** and **Third Class** for passenger travel corresponding to Pclass with values 1,2,3 respectively

```
In [25]: # Create Pclass Label Column
titanic_data_frame['Class'] = titanic_data_frame.Pclass.map({1 : 'First
Class', 2 : 'Second Class', 3 : 'Third Class'})
titanic_data_frame.Class.head()
```

```
Out[25]: 0    Third Class
1    First Class
2    Third Class
3    First Class
4    Third Class
Name: Class, dtype: object
```

Also I would like to add descriptive label for the Embarked port of the passengers i.e. **Cherbourg** for C, **Queenstown** for Q and **Southampton** for S

```
In [26]: # Create Embarked Labels Column
titanic_data_frame['Ports'] = titanic_data_frame.Embarked.map({'C' : 'Cherbour
g', 'Q' : 'Queenstown', 'S' : 'Southampton'})
```

In [27]: `titanic_data_frame.head()`

Out[27]:

	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	7.25
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

SECTION 4: DATA EXPLORATION

Analyzing Relationships

To better understand the data, I would like to have a better understanding of the passenger demographics. I would like to explore the data set on the basis of the following dimensions:

- gender
- age group
- class
- port of embarkation
- other observations

Based on Gender

I would like to explore how many male/female passengers were onboard

```
In [28]: ##Find passengers data based on gender
female_passengers_df=titanic_data_frame[titanic_data_frame['Sex']=='female']
male_passengers_df=titanic_data_frame[titanic_data_frame['Sex']=='male']
print "Total Count based on gender:"
print ""
print "Male: %d" %male_passengers_df['PassengerId'].count()
print "Female: %d" %female_passengers_df['PassengerId'].count()
```

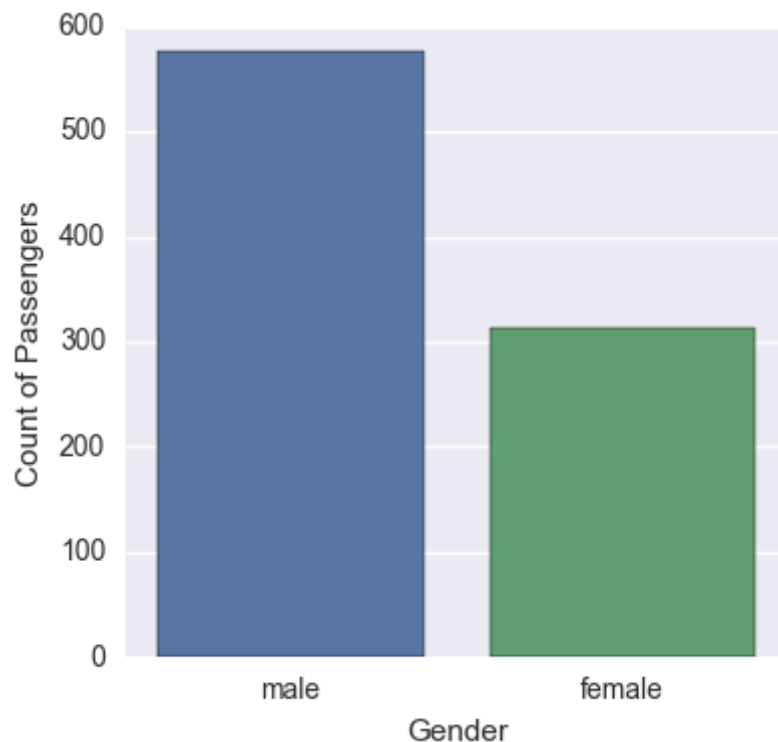
Total Count based on gender:

Male: 577

Female: 314

```
In [29]: gender_grid = sns.factorplot("Sex",data=titanic_data_frame,kind="count")
## Modify facet grid for x/y axis text
(gender_grid.set_axis_labels("Gender", "Count of Passengers")
 .despine(left=True))
```

Out[29]: <seaborn.axisgrid.FacetGrid at 0x9303f98>



From this data set it appears more male passengers were on board rather than female passengers.

Based on age group

I want to determine what was the average age of passengers on board the Titanic. Probably later on in my analysis, I would help me to identify whether ages of the passenger had an impact on their survival.

```
In [30]: print "Count of ages in each age group"
print titanic_data_frame.groupby("AgeGroup")['Age'].count()
print ""
print "Mean of ages in each age group"
titanic_age_groups=titanic_data_frame.groupby("AgeGroup")
print titanic_age_groups['Age'].mean()
print ""
print "Maximum age of passenger onboard %f" %titanic_data_frame["Age"].max()
print ""
minimum_age=titanic_data_frame.groupby("AgeGroup").get_group('Child')['Age'].min()
print "Minimum age of passenger onboard %f" %minimum_age
```

Count of ages in each age group

AgeGroup

Adult 508

Child 180

Elderly 26

NA 177

Name: Age, dtype: int64

Mean of ages in each age group

AgeGroup

Adult 33.912402

Child 12.695389

Elderly 65.096154

NA 0.000000

Name: Age, dtype: float64

Maximum age of passenger onboard 80.000000

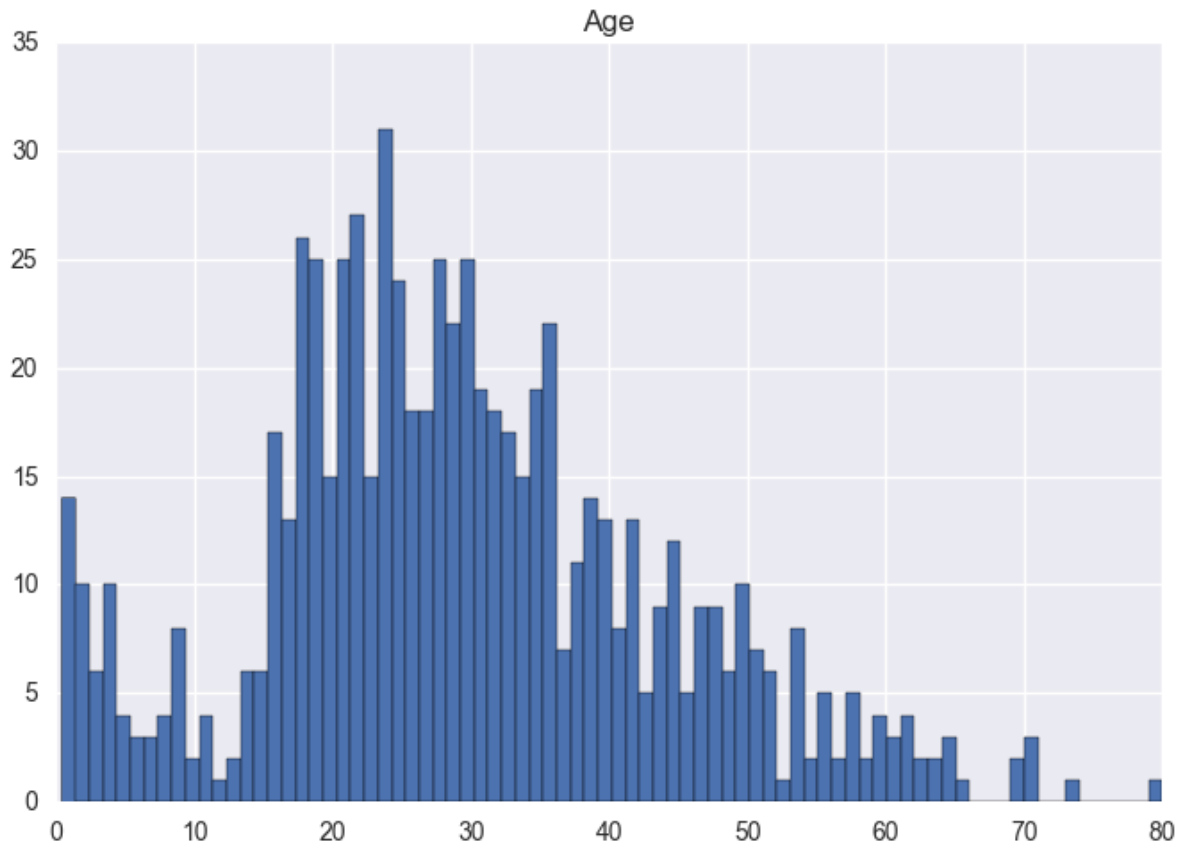
Minimum age of passenger onboard 0.420000

From this data set it appears more adults are on board as compared to children, elderly, people with no ages given(NA). Average age of passengers in the case of adults is 33.91 years, for children is 12.70 years and for elderly it is 65.10 years. Oldest passenger on board(ignoring the missing ages) had age of 80 years and the youngest passenger(ignoring the missing ages) was of 0.42 years old.

To plot the distribution of ages as a histogram of the passengers onboard, I will set the range of the plot to 0.42(minimum age) and 80(maximum age) with 80 bins

```
In [31]: titanic_data_frame.hist("Age",range=[0.42, 80],bins=80)
```

```
Out[31]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000000009303AC8>]], dtype=object)
```



From this visualization it appears, the maximum passengers onboard were between the age group 20-35.

For my further analysis, I would like to group passengers into various groups using a generic function, which I could use for grouping passengers based on age group as well

```
In [32]: def group_passengers(data,criteria):
          return data.get_group(criteria)
```

```
In [33]: #Group children, adult, elderly passengers
children_passengers=group_passengers(titanic_age_groups,"Child")
adult_passengers=group_passengers(titanic_age_groups,"Adult")
elderly_passengers=group_passengers(titanic_age_groups,"Elderly")
```

I would also like to explore the relationship between age group and the gender of the passengers from this dataset and determine any relationships present if any.

```
In [34]: def group_age_gender_passengers(male_passenger,female_passenger):
          total_count = male_passenger["PassengerId"].count()+female_passenger["PassengerId"].count()
          print "Male:%f" %((float)(male_passenger["PassengerId"].count())/total_count)
          print "Female:%f" %((float)(female_passenger["PassengerId"].count())/total_count)
```

```
In [35]: male_child_passengers=group_passengers(children_passengers.groupby("Sex"),"male")
          female_child_passengers=group_passengers(children_passengers.groupby("Sex"),"female")

          print "Children Gender proportion"
          group_age_gender_passengers(male_child_passengers,female_child_passengers)

          adult_male_passengers=group_passengers(adult_passengers.groupby("Sex"),"male")
          adult_female_passengers=group_passengers(adult_passengers.groupby("Sex"),"female")

          print ""
          print "Adult Gender proportion"
          group_age_gender_passengers(adult_male_passengers,adult_female_passengers)

          elderly_male_passengers=group_passengers(elderly_passengers.groupby("Sex"),"male")
          elderly_female_passengers=group_passengers(elderly_passengers.groupby("Sex"),"female")

          print ""
          print "Elderly Gender proportion"
          group_age_gender_passengers(elderly_male_passengers,elderly_female_passengers)
```

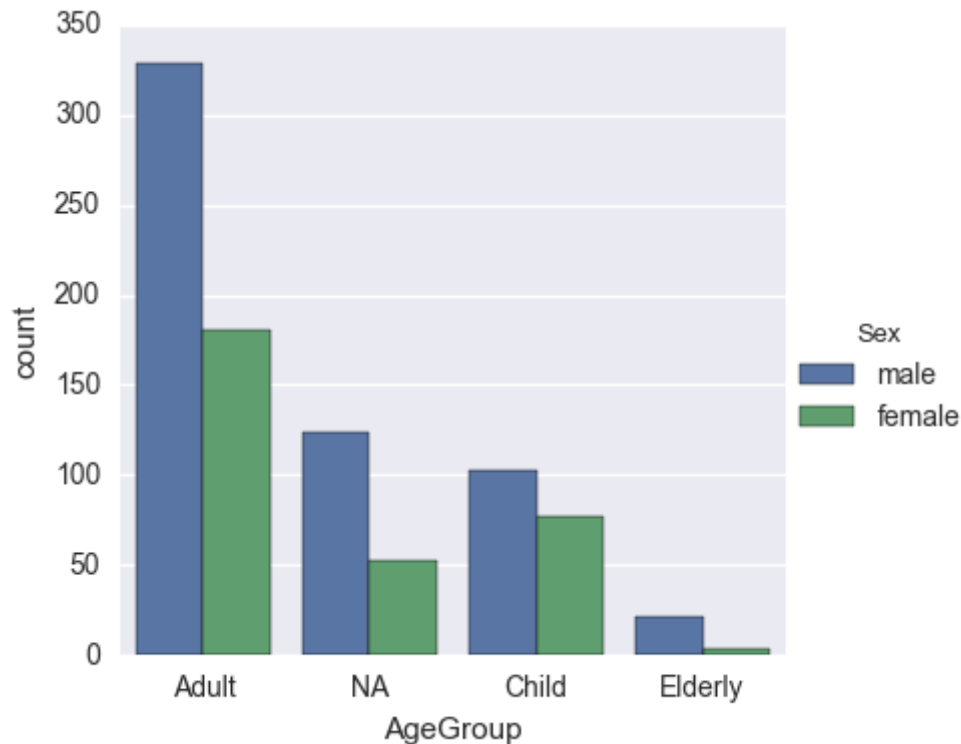
```
Children Gender proportion
Male:0.572222
Female:0.427778
```

```
Adult Gender proportion
Male:0.645669
Female:0.354331
```

```
Elderly Gender proportion
Male:0.846154
Female:0.153846
```

It is evident from these statistics, that count of male passengers of each age group had been more than female passengers onboard.


```
In [36]: age_gender_grid = sns.factorplot("AgeGroup",
hue="Sex",data=titanic_data_frame,kind="count")
```



Based on class

The Titanic's passengers were divided into three separate classes, determined not only by the price of their ticket but by wealth and social class: those travelling in first class, most of them the wealthiest passengers on board, included prominent members of the upper class, businessmen, politicians, high-ranking military personnel, industrialists, bankers, entertainers, socialites, and professional athletes. Second class passengers were middle class travellers and included professors, authors, clergymen, and tourists. Third class or steerage passengers were primarily emigrants moving to the United States and Canada.[5]

```
In [37]: print "Count of passengers in each class"
print titanic_data_frame.groupby("Class")['PassengerId'].count()
```

```
Count of passengers in each class
Class
First Class      216
Second Class     184
Third Class      491
Name: PassengerId, dtype: int64
```

```
In [39]: #Define grouping criteria on the basis of "Pclass"
passenger_class_group=titanic_data_frame.groupby("Pclass")
#Get class 1 passengers
passenger_class1= group_passengers(passenger_class_group,1)
#Get class 2 passengers
passenger_class2= group_passengers(passenger_class_group,2)
#Get class 3 passengers
passenger_class3= group_passengers(passenger_class_group,3)
```

```
In [40]: def compute_mean(data_set,total_data,criteria):
        return ((float)(data_set[criteria].count())/total_data[criteria].count())
```

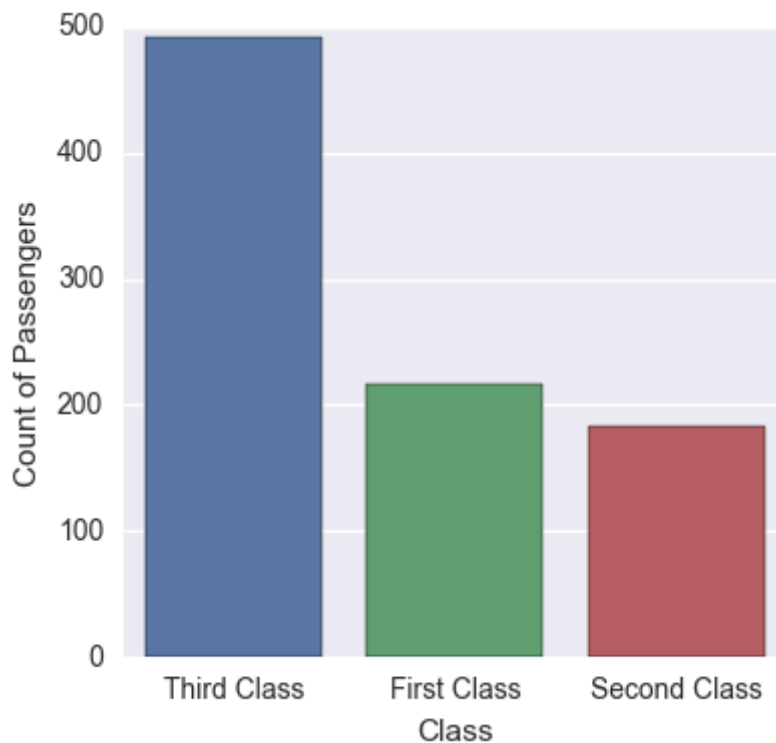
```
In [41]: print "Mean count of passengers per Pclass"
print "First Class: %f" %compute_mean(passenger_class1,titanic_data_frame,"PassengerId")
print "Second Class: %f" %compute_mean(passenger_class2,titanic_data_frame,"PassengerId")
print "Third Class: %f" %compute_mean(passenger_class3,titanic_data_frame,"PassengerId")
```

```
Mean count of passengers per Pclass
First Class: 0.242424
Second Class: 0.206510
Third Class: 0.551066
```

From this analysis, it is clear that the maximum number of passengers that were onboard on Titanic, were travelling in Class 3. Corresponding visualization for the same:

```
In [42]: class_grid = sns.factorplot("Class",data=titanic_data_frame,kind="count")
        ## Modify facet grid for x/y axis text
        (class_grid.set_axis_labels("Class", "Count of Passengers")
         .despine(left=True))
```

Out[42]: <seaborn.axisgrid.FacetGrid at 0xbde7d68>



From these statistics/visualization, it appears maximum passengers were travelling Class 3 which were for the passengers having below the normal socio-economic conditions.

Here I would like to calculate the mean count of passengers based on class and the age group

```
In [43]: def age_class_grouping_passengers(child_pclass_passengers,adult_pclass_passengers,elderly_pclass_passengers):
        print "Children Passengers: %f" %compute_mean(child_pclass_passengers,children_passengers,"PassengerId")
        print "Adult Passengers: %f" %compute_mean(adult_pclass_passengers,adult_passengers,"PassengerId")
        print "Elderly Passengers: %f" %compute_mean(elderly_pclass_passengers,elderly_passengers,"PassengerId")
```

```

In [44]: #Define grouping criteria on the basis of "AgeGroup" for passenger in each class
passenger_class1_age_group=passenger_class1.groupby("AgeGroup")
adult_class1_passengers=group_passengers(passenger_class1_age_group,"Adult")
children_class1_passengers=group_passengers(passenger_class1_age_group,"Child")

elderly_class1_passengers=group_passengers(passenger_class1_age_group,"Elderly")
print "Mean count per age group for Class 1"
age_class_grouping_passengers(children_class1_passengers,adult_class1_passengers,elderly_class1_passengers)
print ""

passenger_class2_age_group=passenger_class2.groupby("AgeGroup")
adult_class2_passengers=group_passengers(passenger_class2_age_group,"Adult")
children_class2_passengers=group_passengers(passenger_class2_age_group,"Child")

elderly_class2_passengers=group_passengers(passenger_class2_age_group,"Elderly")

print "Mean count per age group for Class 2"
age_class_grouping_passengers(children_class2_passengers,adult_class2_passengers,elderly_class2_passengers)
print ""

passenger_class3_age_group=passenger_class3.groupby("AgeGroup")
adult_class3_passengers=group_passengers(passenger_class3_age_group,"Adult")
children_class3_passengers=group_passengers(passenger_class3_age_group,"Child")

elderly_class3_passengers=group_passengers(passenger_class3_age_group,"Elderly")

print "Mean count per age group for Class 3"
age_class_grouping_passengers(children_class3_passengers,adult_class3_passengers,elderly_class3_passengers)

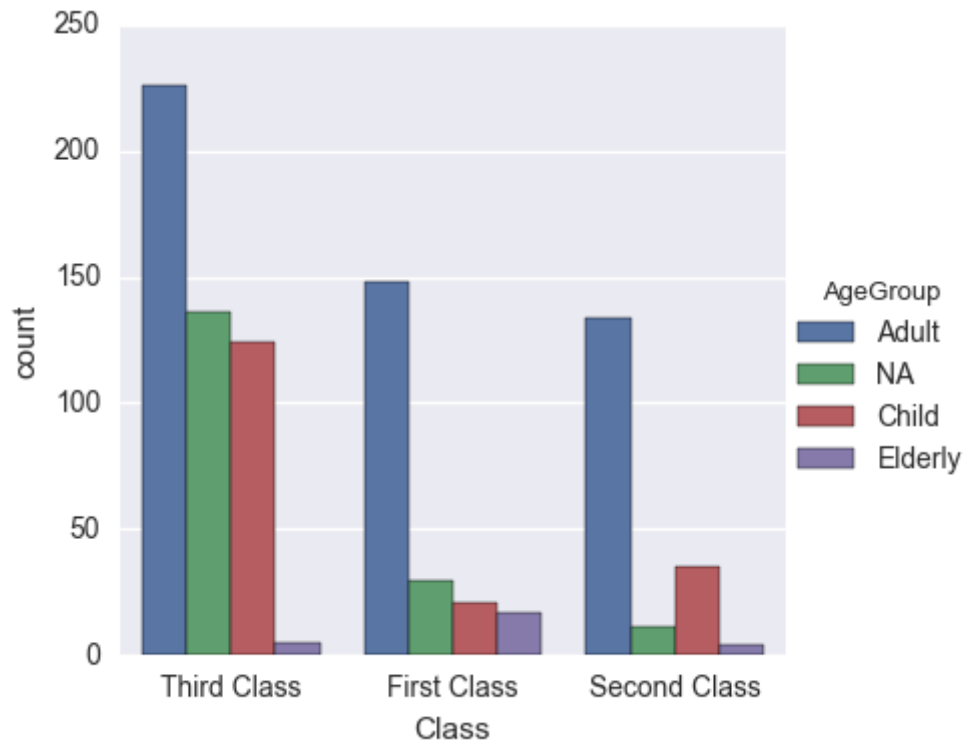
```

Mean count per age group for Class 1
 Children Passengers: 0.116667
 Adult Passengers: 0.291339
 Elderly Passengers: 0.653846

Mean count per age group for Class 2
 Children Passengers: 0.194444
 Adult Passengers: 0.263780
 Elderly Passengers: 0.153846

Mean count per age group for Class 3
 Children Passengers: 0.688889
 Adult Passengers: 0.444882
 Elderly Passengers: 0.192308

```
In [45]: class_age_grid = sns.factorplot("Class", hue="AgeGroup", data=titanic_data_frame, kind="count")
```



From this data it is evident that the maximum number of passengers in the age group "Adult" were in Class 3 (with average 0.444882), "Child" were also in Class 3 (with average 0.688889) and elderly were in Class 1 (with average 0.033465)

I would also like to explore the proportion of female passengers versus the male passengers in each class.

```
In [46]: def calculate_male_female_proportion_class(data):
class_gender_array=data.groupby("Sex")["PassengerId"].count()
male_count= class_gender_array.get("male")
female_count=class_gender_array.get("female")
print "Male: %f" %((float(male_count))/(male_count+female_count))
print "Female: %f" %((float(female_count))/(male_count+female_count))
```

```
In [47]: print "Gender proportions per class"
print "First Class"
calculate_male_female_proportion_class(passenger_class1)
print ""
print "Second Class"
calculate_male_female_proportion_class(passenger_class2)
print ""
print "Third Class"
calculate_male_female_proportion_class(passenger_class3)
```

Gender proportions per class

First Class

Male: 0.564815

Female: 0.435185

Second Class

Male: 0.586957

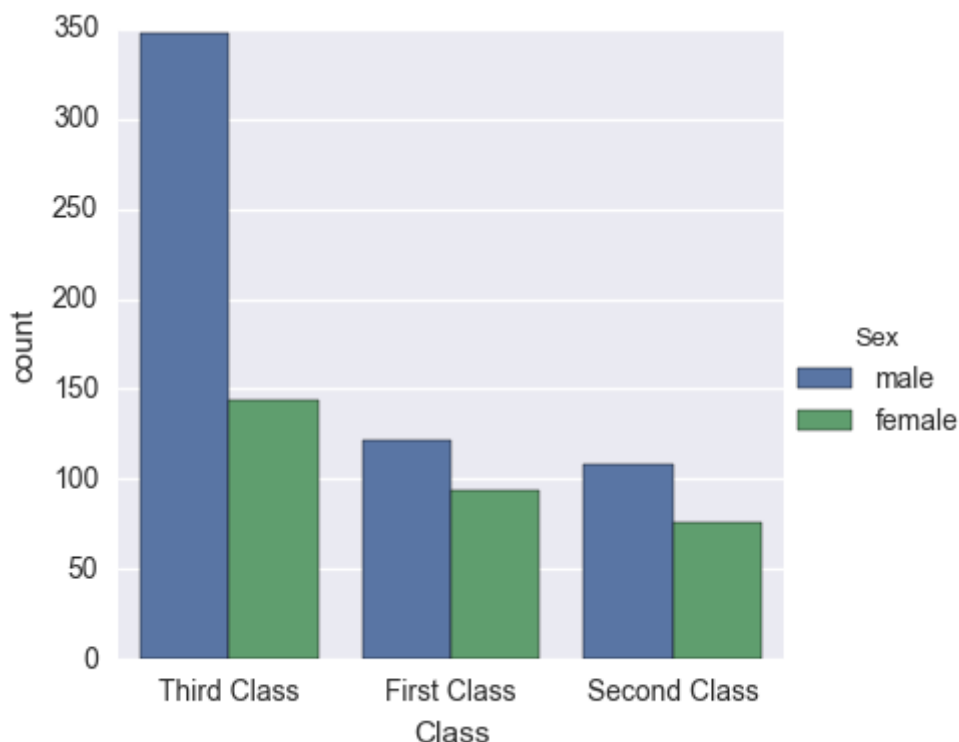
Female: 0.413043

Third Class

Male: 0.706721

Female: 0.293279

```
In [48]: class_age_grid = sns.factorplot("Class", hue="Sex", data=titanic_data_frame, kind="count")
```



As it is already evident that the maximum number of passengers in the data set are male, from this visualization it is clear as well that the proportion of male passengers in each class had been significantly higher than the female passengers. The maximum difference in proportions had been in Class 3 with 0.706721 males versus 0.293279 female passengers

Based on Port of Embarkation

For my analysis I would like to identify if the boarding point (or the port of embarkation) had any impact on the passenger's survival. So I would like to group passengers on the basis of their port of embarkation as well.

```
In [50]: #Group passengers based on port of embarkation
port_passengers=titanic_data_frame.groupby("Ports")
print "Passengers count for each port"
print port_passengers["PassengerId"].count()
```

```
Passengers count for each port
Ports
Cherbourg      168
Queenstown     77
Southampton    644
Name: PassengerId, dtype: int64
```

Reusing the above defined function for grouping passengers based on criteria and computing mean, I can do the same for classifying passengers based on their port of Embarkation i.e. Cherbourg, Queenstown, Southampton

```
In [51]: #Separate dataframes for each port
port_s_passengers=group_passengers(port_passengers,"Southampton")
port_q_passengers=group_passengers(port_passengers,"Queenstown")
port_c_passengers=group_passengers(port_passengers,"Cherbourg")
```

Here I reuse the above function to calculate mean probability of passengers embarking from each port

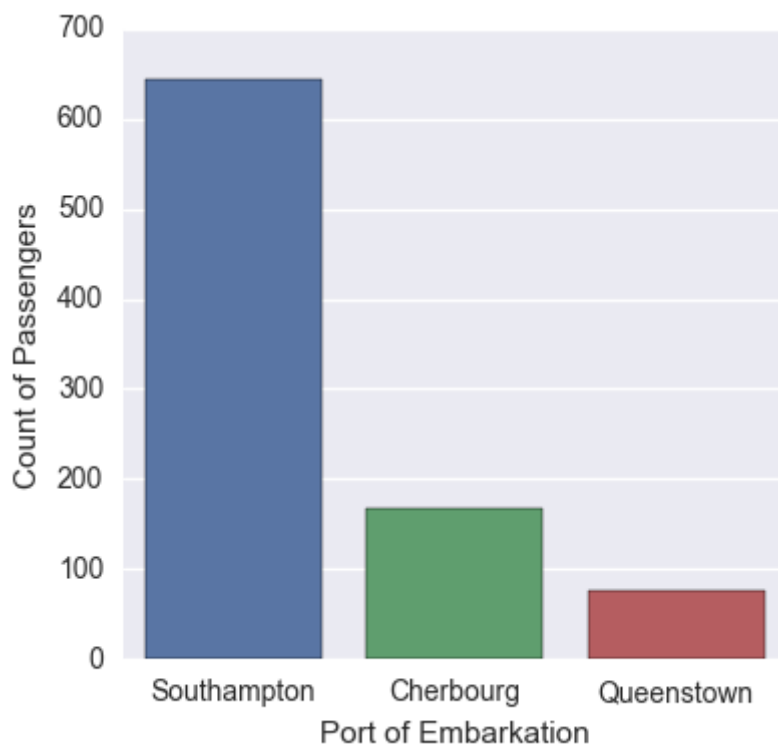
```
In [52]: print "Average number of passengers for each port"
print "Cherbourg %f" %compute_mean(port_c_passengers,titanic_data_frame,"PassengerId")
print "Queenstown %f" %compute_mean(port_q_passengers,titanic_data_frame,"PassengerId")
print "Southampton %f" %compute_mean(port_s_passengers,titanic_data_frame,"PassengerId")
```

```
Average number of passengers for each port
Cherbourg 0.188552
Queenstown 0.086420
Southampton 0.722783
```

Corresponding visualization for the same

```
In [53]: port_embark_fg = sns.factorplot("Ports",data=titanic_data_frame,kind="count")  
        ## Modify facet grid(port_embark_fg) for x/y axis text  
        (port_embark_fg.set_axis_labels("Port of Embarkation", "Count of Passengers")  
         .despine(left=True))
```

Out[53]: <seaborn.axisgrid.FacetGrid at 0xcd336a0>



From this analysis it is clear that the maximum number of passengers who embarked in Titanic were from Southampton.

Based on Survival

For my further analysis and for answering the above mentioned questions, I would like to keep data sets of surviving and non surviving passengers separately.


```
In [54]: ## split data frames into groups of passengers who survived
surviving_passengers_df =titanic_data_frame[titanic_data_frame['Survival_Status']=='Survived']
## split data frames into groups of passengers who did not survive
non_surviving_passengers_df =titanic_data_frame[titanic_data_frame['Survival_Status']=='Died']
surviving_passengers_df.head()
```

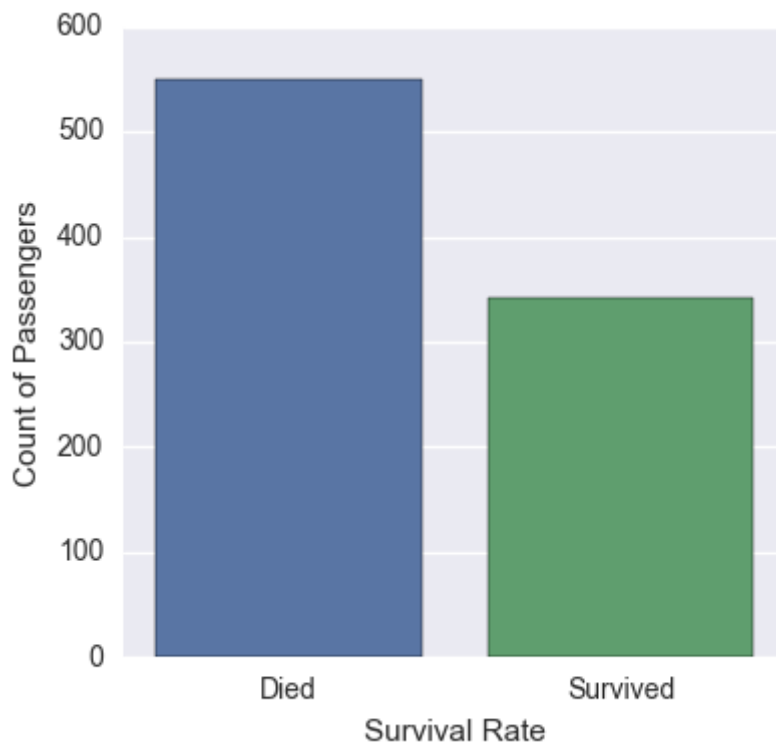
Out[54]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
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
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.13
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0

Visualization of passengers how many passengers survived and how many did not

```
In [55]: ## Factor plot to show count of passengers who survived
survival_fg =
sns.factorplot(x="Survival_Status",data=titanic_data_frame,kind="count")
## Modify facet grid for x/y axis text
(survival_fg.set_axis_labels("Survival Rate", "Count of Passengers")
.despine(left=True))
```

Out[55]: <seaborn.axisgrid.FacetGrid at 0xc4a9cf8>



```
In [57]: ##Calculate count of passengers who survived and who did not
surviving_passengers_count = surviving_passengers_df['PassengerId'].count()
print "Survival count"
print ""
print "Survived: %d" %surviving_passengers_count
non_surviving_passengers_count = non_surviving_passengers_df['PassengerId'].count()
print "Died: %d" %non_surviving_passengers_count

print "Survival proportion:%f" %(float(surviving_passengers_count)/(non_surviving_passengers_count+surviving_passengers_count))
```

Survival count

Survived: 342

Died: 549

Survival proportion:0.383838

From these statistics, it is visible that from the total count of passengers i.e. 891 in the dataset, only a small proportion of 0.383838 passengers had chances of survival.

Other Observations

Based on Ticket Number

After viewing the entire Titanic data set, I realized that there are multiple passengers having the same ticket number. It could be that these passengers may be family members. But there are instances when the SibSp(sibling and spouse) column and Parch(parent and child) columns have 0 values for passengers having same ticket number. But there may be more than passenger having more than 1 same ticket number.

So it could be friends, girlfriends, cousins, servants, step family members, nannies etc that may be travelling with the main passenger and since they are not immediate family member so the Parch and SibSp have 0 values. Since it may not be necessary that servants etc would be travelling in the same cabin as their master, I want to analyze how many people who were travelling as a group/with the same ticket number that had lost more than 1 member in that group.

```
In [58]: #Group passengers based on the same ticket number
same_ticket_passengers=titanic_data_frame.groupby('Ticket')
#calculate passenger group count
same_ticket_passengers_total= same_ticket_passengers.ngroups
print "Passenger groups count %d" %same_ticket_passengers_total
#count number of passengers in each group
ticket_total_passenger_count=same_ticket_passengers['PassengerId'].count()
passenger_group_non_surviving_count=0
#Iterate passenger group to calculate count of passengers if survival count in
each group < 1
for ticket_number,data_point in same_ticket_passengers:
    passengers_count=ticket_total_passenger_count.get(ticket_number)
    surviving_ticket_passengers_count=
data_point[data_point['Survival_Status']=='Survived']['PassengerId'].count()
    survival_proportion = surviving_ticket_passengers_count/passengers_count;
    if survival_proportion<1:
        passenger_group_non_surviving_count+=1
print "Families impacted by the disaster %d" %passenger_group_non_surviving_co
unt
print ""
print "Proportion of passenger groups travelling together who:"
print ""
print "Lost member(s) in disaster: %f" %((float)(passenger_group_non_surviving
_count)/same_ticket_passengers_total)
print "Did not loose member(s) in disaster: %f" %((float)(same_ticket_passeng
ers_total-passenger_group_non_surviving_count)/same_ticket_passengers_total)
```

Passenger groups count 681

Families impacted by the disaster 470

Proportion of passenger groups travelling together who:

Lost member(s) in disaster: 0.690162

Did not loose member(s) in disaster: 0.309838

From these statistics it is evident, that about 0.690162 proportion of families/passenger groups were impacted by the disaster as they had lost one or more than one member of the same ticket group in the disaster which explains the calamity of the disaster. This shows at least 69% of passenger groups travelling in the Titanic had lost a loved one/person they knew in the ship wreck.

Women and Children first

I want to calculate proportion for woman/child passengers vs the male passengers to understand if it had any impact

```
In [59]: #Group womanchild passengers and men passengers
woman_child_passengers=group_passengers(titanic_data_frame.groupby("WomanChild"),"Yes")
men_passengers=group_passengers(titanic_data_frame.groupby("WomanChild"),"No")
print "Woman Child passengers"
print "Count: %d" %woman_child_passengers['PassengerId'].count()
print "Proportion %f" %compute_mean(woman_child_passengers,titanic_data_frame,"PassengerId")
print ""
print "Male passengers"
print "Count: %d" %men_passengers['PassengerId'].count()
print "Proportion %f" %compute_mean(men_passengers,titanic_data_frame,"PassengerId")
```

Woman Child passengers
Count: 417
Proportion 0.468013

Male passengers
Count: 474
Proportion 0.531987

From these statistics it appears that around 47% of passengers onboard the Titanic were women or children and around 53% were men.

Investigating survival factors

Here I would like to answer the questions I put up in **Section 2: Questions** of this report which probably governed passenger's survival namely :

- individual passenger traits
- passenger's travelling conditions
- passenger's company

Let us start with exploring the answers to these questions

1. Individual Passenger Traits

Based on Gender

I want to investigate whether the gender of the passengers played any part in survival of the passengers. Can I reach a conclusion on the basis of this data set whether males or females had more chances of survival in the disaster

Comparison count of males/females who survived

Calculate count of passengers on board who survived and who did not based on gender:

```
In [61]: #Function to group surviving/non surviving passengers based on gender  
def passenger_survival(data,factor,value):  
    return data[data[factor]==value]
```

```
In [63]: ##Find surviving passengers data frame based on gender
male_surviving_passengers_df = passenger_survival(surviving_passengers_df,'Sex','male')
female_surviving_passengers_df = passenger_survival(surviving_passengers_df,'Sex','female')
print "Survival count based on gender:"
print ""
print "Survived:"
print "Male: %d" %male_surviving_passengers_df['PassengerId'].count()
print "Female: %d" %female_surviving_passengers_df['PassengerId'].count()

##Find non-surviving passengers df based on gender
male_non_surviving_passengers_df = passenger_survival(non_surviving_passengers_df,'Sex','male')
female_non_surviving_passengers_df = passenger_survival(non_surviving_passengers_df,'Sex','female')
print ""
print "Died:"
print "Male: %d" %male_non_surviving_passengers_df['PassengerId'].count()
print "Female: %d" %female_non_surviving_passengers_df['PassengerId'].count()
```

Survival count based on gender:

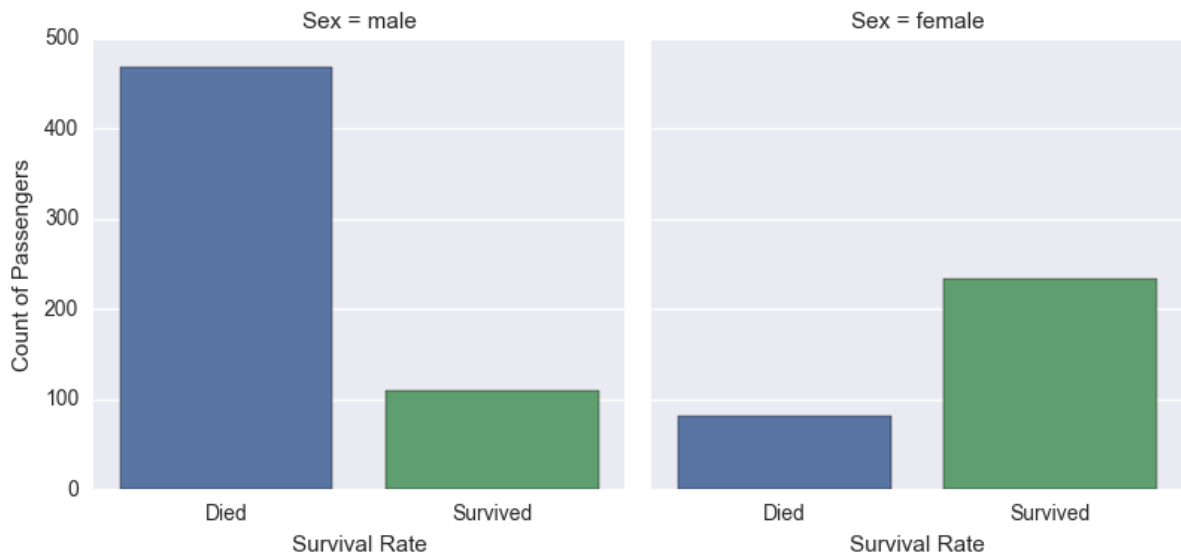
Survived:
Male: 109
Female: 233

Died:
Male: 468
Female: 81

Visualization

```
In [64]: gender_fg = sns.factorplot(x="Survival_Status",col="Sex",data=titanic_data_frame,kind="count")
          ## Modify facet grid for x/y axis text
          (gender_fg.set_axis_labels("Survival Rate", "Count of Passengers")
           .despine(left=True))
```

Out[64]: <seaborn.axisgrid.FacetGrid at 0xcdbd0f0>



From these statistics we could make there were more male passengers onboard then female passengers. But more female passengers had survived the passengers.

The rate of survival for male vs female passengers wrt. to their total count:

```
In [65]: print "Proportion of passengers who survived"
          print "Male: %f" %compute_mean(male_surviving_passengers_df,male_passengers_df,"PassengerId")
          print "Female: %f" %compute_mean(female_surviving_passengers_df,female_passengers_df,"PassengerId")
```

```
Proportion of passengers who survived
Male: 0.188908
Female: 0.742038
```

From this we can make out that the **female passengers had more chances of surviving from the male passengers** of about 0.742 in the presented data set.

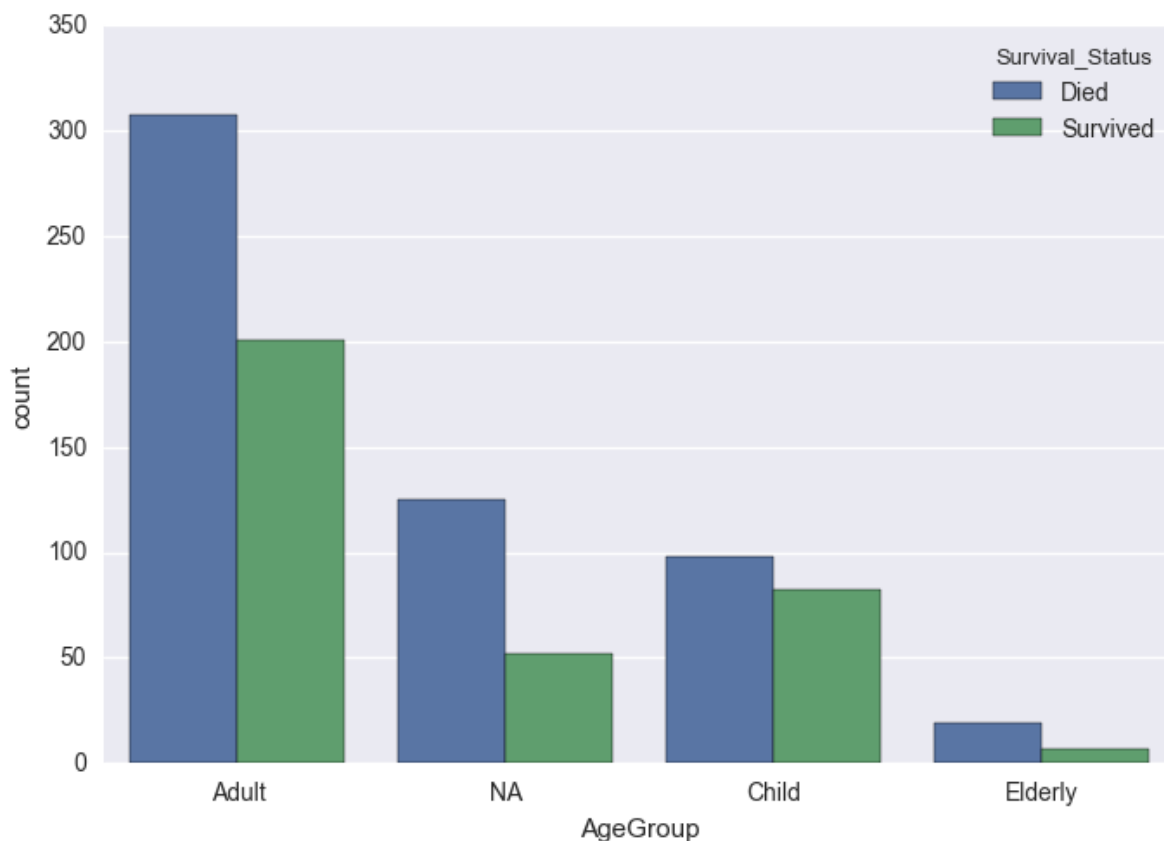
Based on age

From the surviving passenger, I want to determine whether the age group they belonged to played a role in their survival. To analyze this I can count the number of passengers in each age group.

```
In [66]: surviving_passengers_df.groupby("AgeGroup")['PassengerId'].count()
```

```
Out[66]: AgeGroup
Adult      201
Child       82
Elderly     7
NA          52
Name: PassengerId, dtype: int64
```

```
In [67]: ax = sns.countplot(x="AgeGroup", hue="Survival_Status", data=titanic_data_frame)
```



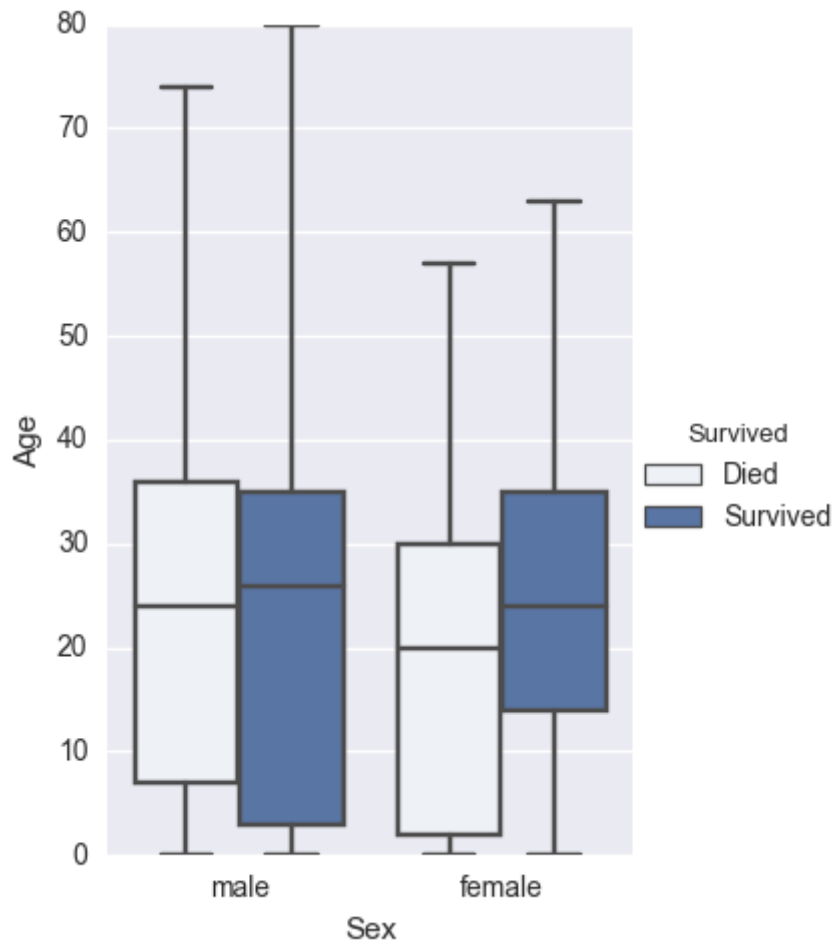
From this visualization it is visible that maximum number of adult passengers had died in the disaster and child passengers had survived more in the disaster.

Based on Age and sex

We can also determine relationship between passenger survival based on age and sex


```
In [68]: titanic_fg = sns.FacetGrid(titanic_data_frame,size=5, aspect=.7)
(titanic_fg.map(sns.boxplot, "Sex", "Age", "Survival_Status")
.despine(left=True)
.add_legend(title="Survived"))
```

Out[68]: <seaborn.axisgrid.FacetGrid at 0xd674240>



From this we can make out, **Female passengers had more chances of surviving**. Male passengers who survived had an average age around 23 and female passengers who survived had an average age around 25.

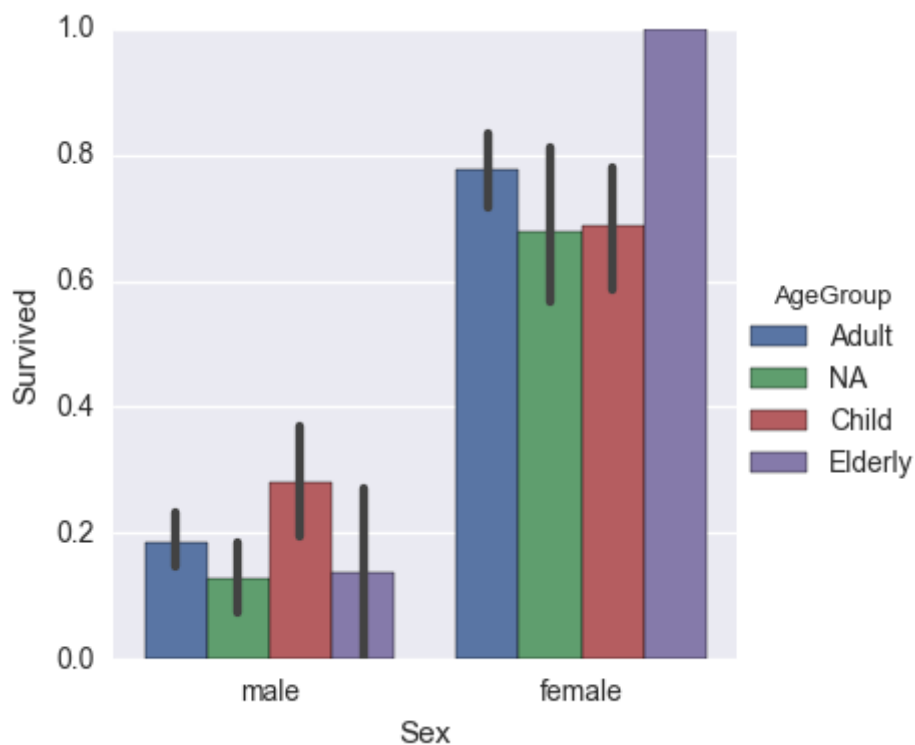
```
In [69]: print "Male"
print "Survival rate %f" %male_surviving_passengers_df['Age'].mean()
print "Oldest %f" %male_surviving_passengers_df['Age'].max()
print ""
print "Female"
print "Survival rate %f" %female_surviving_passengers_df['Age'].mean()
print "Oldest %f" %female_surviving_passengers_df['Age'].max()
```

```
Male
Survival rate 23.272202
Oldest 80.000000
```

```
Female
Survival rate 24.390558
Oldest 63.000000
```

We can also check the relationship between the age group of passengers, sex and how much it impacted their survival rate

```
In [70]: sex_age_group_axis = sns.factorplot(x="Sex", y="Survived", hue="AgeGroup", data=titanic_data_frame, kind="bar")
```



From this graph we can clearly make out that elderly woman had 100% survival rate. Female passengers in general had the highest survival rate. The statistics corresponding to this:

```
In [71]: #Group surviving female passengers based on age group and count surviving pass
sengers
surviving_female_ages=female_surviving_passengers_df.groupby('AgeGroup')['Pass
engerId'].count()
#Group all female passengers based on age group and count passsengers
all_female_ages=female_passengers_df.groupby('AgeGroup')
['PassengerId'].count()
print "Surviving Female age groups average:"
print surviving_female_ages/all_female_ages

#Group surviving male passengers based on age group and count surviving passse
ngers
surviving_male_ages=male_surviving_passengers_df.groupby('AgeGroup')['Passenge
rId'].count()
#Group all male passengers based on age group and count passsengers
all_male_ages=male_passengers_df.groupby('AgeGroup')['PassengerId'].count()
print ""
print "Surviving Male age groups average"
print surviving_male_ages/all_male_ages
```

Surviving Female age groups average:

```
AgeGroup
Adult      0.777778
Child      0.688312
Elderly    1.000000
NA         0.679245
Name: PassengerId, dtype: float64
```

Surviving Male age groups average

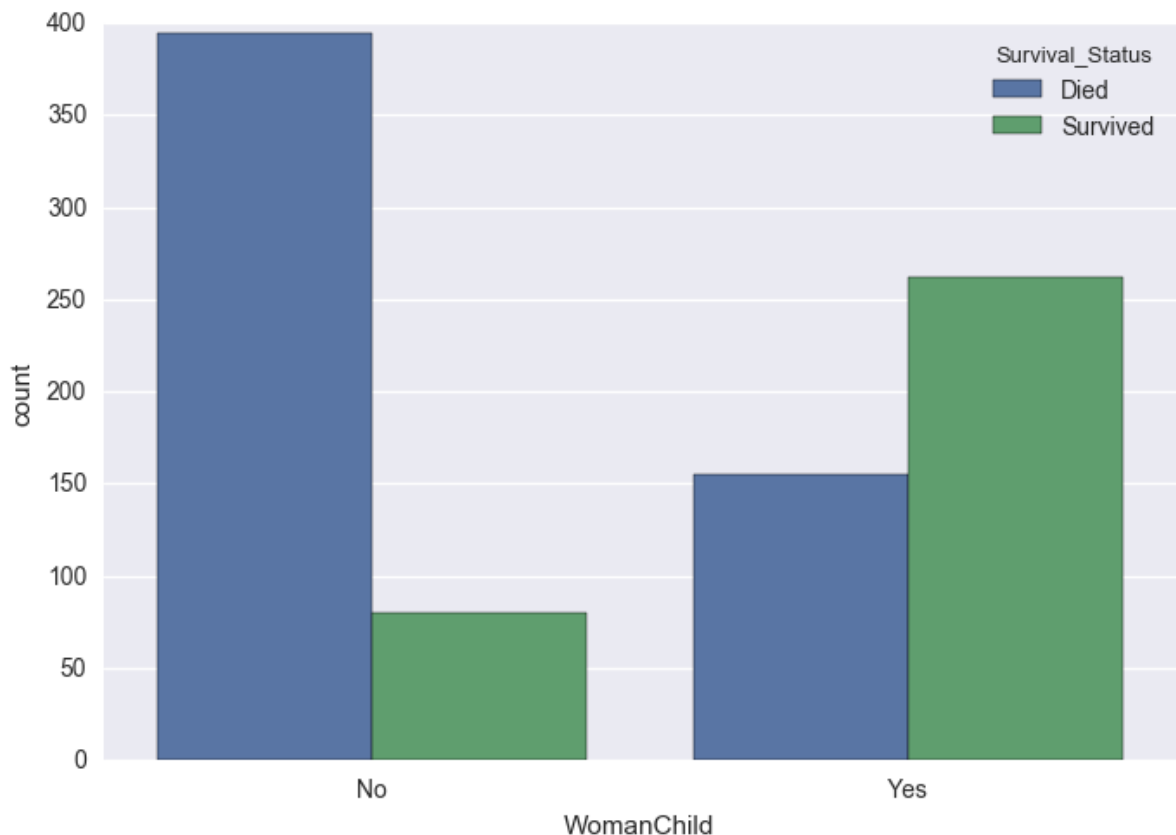
```
AgeGroup
Adult      0.185976
Child      0.281553
Elderly    0.136364
NA         0.129032
Name: PassengerId, dtype: float64
```

```
In [72]: surviving_children_passengers=passenger_survival(surviving_passengers_df,"AgeG
roup","Child")
surviving_adult_passengers=passenger_survival(surviving_passengers_df,"AgeGrou
p","Adult")
surviving_elderly_passengers=passenger_survival(surviving_passengers_df,"AgeGr
oup","Elderly")
```

Based on Woman/Child passengers

I would like to determine did being in the passenger group of woman/child helped in the survival of the passengers.

```
In [73]: womanchild_survival_axis = sns.countplot(x="WomanChild",
hue="Survival_Status", data=titanic_data_frame)
```



From this visualization it is evident that the woman/child group had better survival count compared to men passengers.

```
In [74]: print "Survival Proportion:"
woman_child_survivors = woman_child_passengers[woman_child_passengers['Survived']==1]
men_survivors = men_passengers[men_passengers['Survived']==1]
print "Woman Child: %f" %compute_mean(woman_child_survivors,woman_child_passengers,"PassengerId")
print "Men: %f" %compute_mean(men_survivors,men_passengers,"PassengerId")
```

```
Survival Proportion:
Woman Child: 0.628297
Men: 0.168776
```

Also the above statistics confirm that woman child passengers had about 62.8% chances of survival from this data set as again men passengers with about 16.8% chances.

2. Passenger's travelling conditions

Based on Pclass

I want to investigate whether the passenger class played any part in survival of the passengers. So I would group surviving passengers into their Class i.e. First, Second and Third class and determine whether it helped in their survival.

```
In [75]: def compute_counts(data,criteria):
         return data[criteria].count()
```

```
In [77]: #Group pass
surviving_class1_passengers=passenger_survival(surviving_passengers_df,'Class'
first Class')
print "In class 1"
print "%d survived out of %d" %(compute_counts(surviving_class1_passengers,'Pa
ssengerId'),compute_counts(passenger_class1,'PassengerId'))
print "Survival Proportion: %f" %compute_mean(surviving_class1_passengers,pass
enger_class1,'PassengerId')

print ""
surviving_class2_passengers = passenger_survival(surviving_passengers_df,'Clas
s','Second Class')
print "In class 2"
print "%d survived out of %d" %(compute_counts(surviving_class2_passengers,'Pa
ssengerId'),compute_counts(passenger_class2,'PassengerId'))
print "Survival Proportion: %f" %compute_mean(surviving_class2_passengers,pass
enger_class2,'PassengerId')

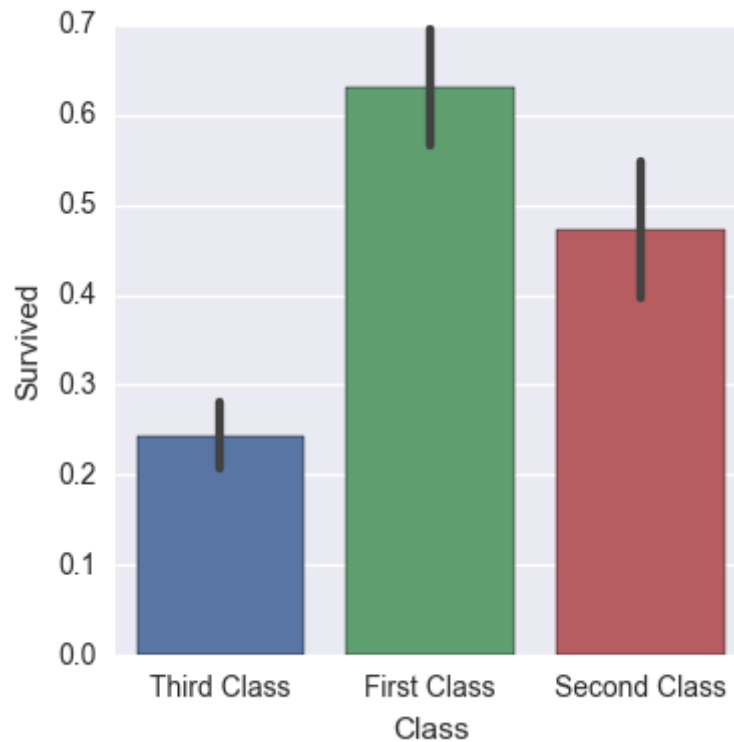
surviving_class3_passengers = passenger_survival(surviving_passengers_df,'Clas
s','Third Class')
print ""
print "In class 3"
print "%d survived out of %d" %(compute_counts(surviving_class3_passengers,'Pa
ssengerId'),compute_counts(passenger_class3,'PassengerId'))
print "Survival Proportion: %f" %compute_mean(surviving_class3_passengers,pass
enger_class3,'PassengerId')
```

```
In class 1
136 survived out of 216
Survival Proportion: 0.629630
```

```
In class 2
87 survived out of 184
Survival Proportion: 0.472826
```

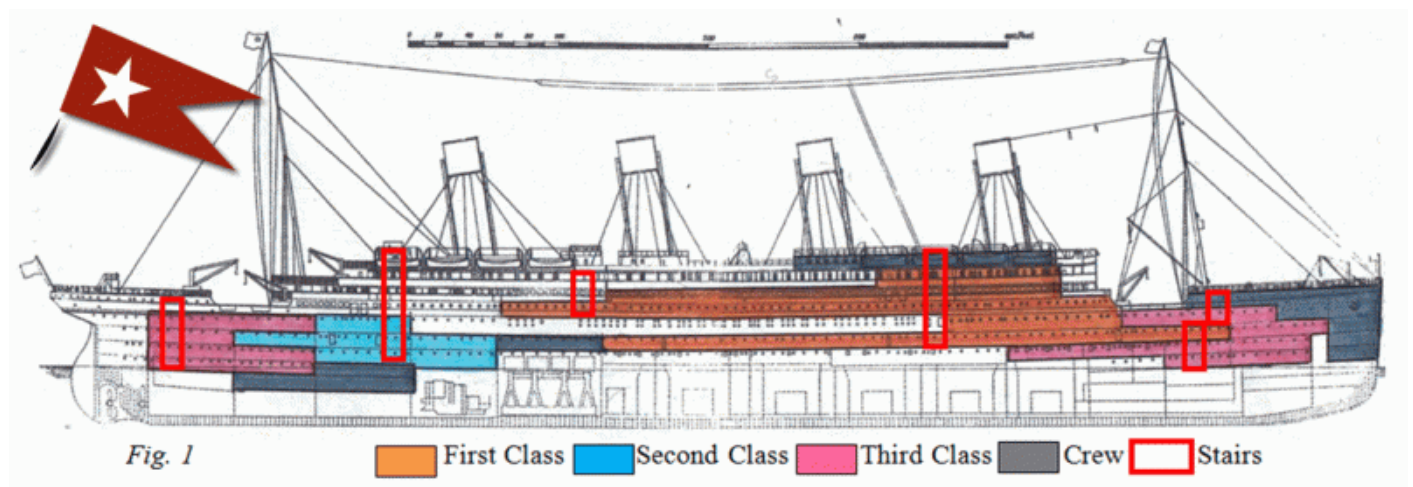
```
In class 3
119 survived out of 491
Survival Proportion: 0.242363
```

```
In [78]: class_grid = sns.factorplot("Class","Survived",
data=titanic_data_frame,kind="bar")
```



From these statistics and visualizations it appears that the **passengers in Pclass 1** had higher chances of survival of about 62.9% and **class 3 had lower chances of survival** of about 24.2%.

On researching this further I found the below analysis:



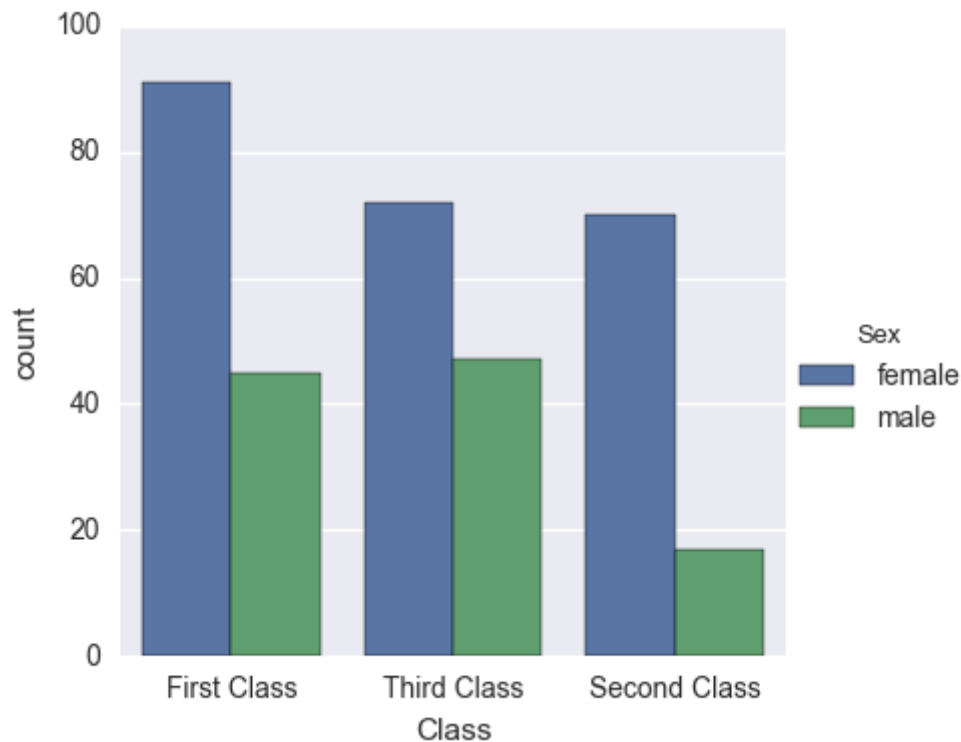
The above figure illustrates the division of class that existed on-board the Titanic, we see First Class passengers enjoying the upper and most central spaces while Third Class and Crew occupy the lowest extremities of the ship. In fact, this segregation extended to advertisement and embarkation with each class having its own booking office and entrance to the ship. So probably this fact alone facilitated the great loss of life among the Third Class.[6]

We can also check from surviving passengers passengers of which gender has the highest probability of surviving in each class

Based on Pclass and Sex

I want to investigate whether the gender also played a part in survival of passengers across various passenger classes.

```
In [79]: surviving_gender_class_grid = sns.factorplot("Class",hue="Sex",kind="count",data=surviving_passengers_df)
```



```
In [80]: print "Survival status for gender/class"
print "First Class"
print passenger_class1.groupby('Sex').Survived.mean()
print ""
print "Second Class"
print passenger_class2.groupby('Sex').Survived.mean()
print ""
print "Third Class"
print passenger_class3.groupby('Sex').Survived.mean()
```

Survival status for gender/class

First Class

Sex

female 0.968085

male 0.368852

Name: Survived, dtype: float64

Second Class

Sex

female 0.921053

male 0.157407

Name: Survived, dtype: float64

Third Class

Sex

female 0.500000

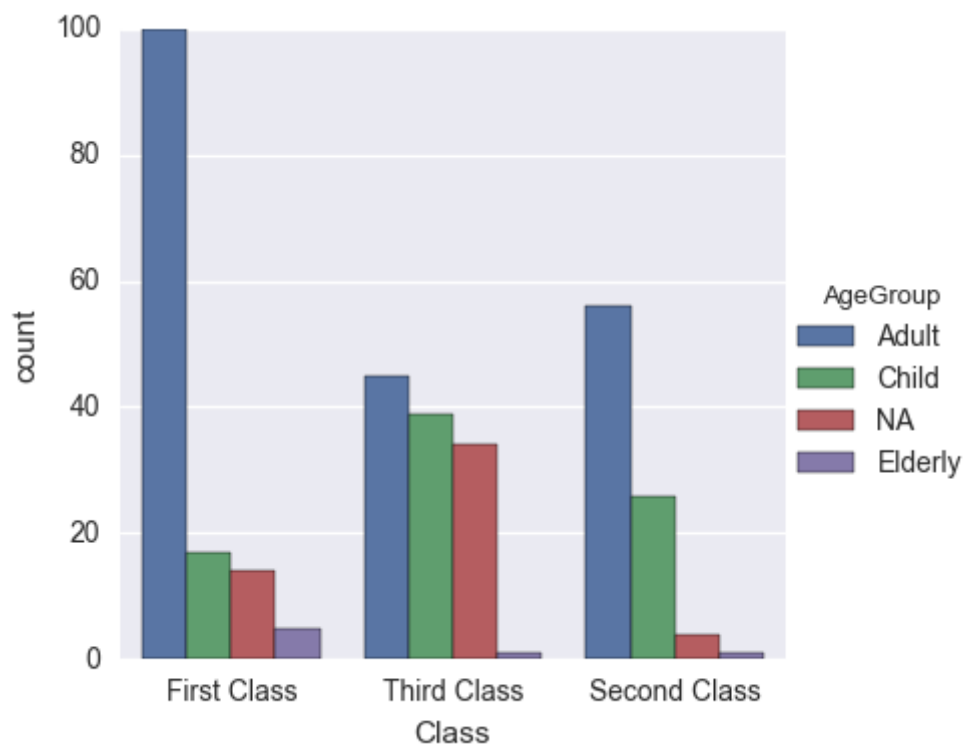
male 0.135447

Name: Survived, dtype: float64

From this visualization it is clear that **female passengers had survival rate irrespective of the class** they were travelling in. In class 1 it is 96.8%, class 2 it is 92.1% and class 3 it is around 50%

Based on Pclass and AgeGroup


```
In [83]: surviving_class_age_grid =  
sns.factorplot("Class",hue="AgeGroup",kind="count",data=surviving_passengers_d  
f)
```



The statistics corresponding to this are:

```
In [302]: print "Mean survival per age group"
print "Class 1"
print "Children: %f" %children_class1_passengers.Survived.mean()
print "Adult: %f" %adult_class1_passengers.Survived.mean()
print "Elderly: %f" %elderly_class1_passengers.Survived.mean()

print ""
print "Class 2"
print "Children: %f" %children_class2_passengers.Survived.mean()
print "Adult: %f" %adult_class2_passengers.Survived.mean()
print "Elderly: %f" %elderly_class2_passengers.Survived.mean()

print ""
print "Class 3"
print "Children: %f" %children_class3_passengers.Survived.mean()
print "Adult: %f" %adult_class3_passengers.Survived.mean()
print "Elderly: %f" %elderly_class3_passengers.Survived.mean()
```

Mean survival per age group

Class 1

Children: 0.809524

Adult: 0.675676

Elderly: 0.294118

Class 2

Children: 0.742857

Adult: 0.417910

Elderly: 0.250000

Class 3

Children: 0.314516

Adult: 0.199115

Elderly: 0.200000

From this visualization, it appears that **adults**(between age 21-59) **had the highest overall rate of survival in all the classes**, highest being in Class 1. **Children and elderly have the highest rate of survival in Class 1.**

From the above statistic, it appears about 80.9% of children, 67.5% of adults, 29.41% of elderly in survived in class 1. In class 2 about 74.2% of children, 41.7% of adults, 25% of elderly survived. Class 3 had the lowest proportion of survival with children being 31.4%, adults with 19.9% and elderly with about 20%.

Based on Woman/Child Passengers

It has been earlier stated that the lifeboat preference was given to women/children. I wanted to analyze whether being a woman/child affect survivability of the passengers.

```
In [84]: class1_woman_child_passengers=passenger_class1[passenger_class1['WomanChild']!=
='Yes']
class2_woman_child_passengers=passenger_class2[passenger_class2['WomanChild']!=
='Yes']
class3_woman_child_passengers=passenger_class3[passenger_class3['WomanChild']!=
='Yes']

print "Woman/Child survival per class"
print "Class 1: %f" %class1_woman_child_passengers.Survived.mean()
print "Class 2: %f" %class2_woman_child_passengers.Survived.mean()
print "Class 2: %f" %class3_woman_child_passengers.Survived.mean()
```

Woman/Child survival per class

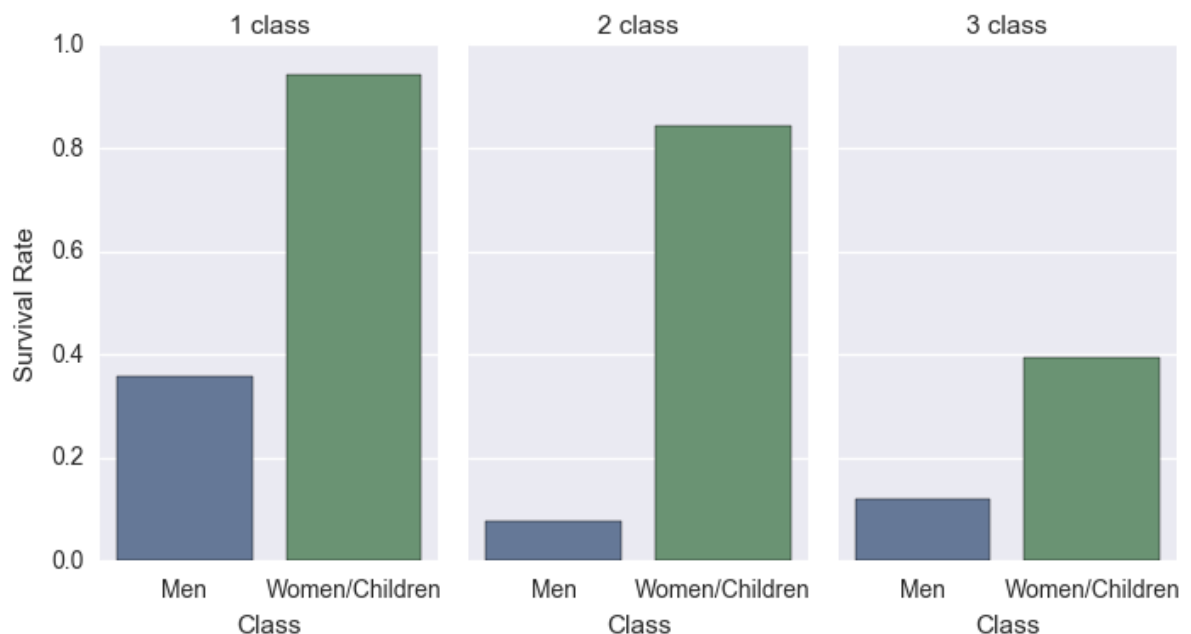
Class 1: 0.940594

Class 2: 0.842105

Class 2: 0.393665

```
In [85]: woman_child_survival_fg = sns.factorplot(x="WomanChild", y="Survived", col="Pc
lass",
          data=titanic_data_frame, saturation=.5,
          kind="bar", ci=None, aspect=.6)
(woman_child_survival_fg.set_axis_labels("Class", "Survival Rate")
.set_xticklabels(["Men", "Women/Children"]))
.set_titles("{col_name} class")
.set(ylim=(0, 1))
.despine(left=True))
```

Out[85]: <seaborn.axisgrid.FacetGrid at 0xe968438>



This visualization represents the survival rate per class for men vs women/child. And it clearly states that **women/child passengers had highest rate of survival per passenger class.**

Based on port of embarkation

I want to calculate survival rate of passengers based on the port of embarkation. I want to determine whether the boarding point of passenger i.e. Cherbourg, Queenstown, Southampton had any impact on their survival.

```
In [86]: ## Statistics based on port of embarkation
print 'Count per embarked port'
print surviving_passengers_df.groupby('Ports')['PassengerId'].count()
print 'Mean per embarked port'
print titanic_data_frame.groupby('Ports').Survived.mean()
print 'Standard Deviation per embarked port'
print titanic_data_frame.groupby('Ports').Survived.std()
```

Count per embarked port

Ports

Cherbourg	93
-----------	----

Queenstown	30
------------	----

Southampton	217
-------------	-----

Name: PassengerId, dtype: int64

Mean per embarked port

Ports

Cherbourg	0.553571
-----------	----------

Queenstown	0.389610
------------	----------

Southampton	0.336957
-------------	----------

Name: Survived, dtype: float64

Standard Deviation per embarked port

Ports

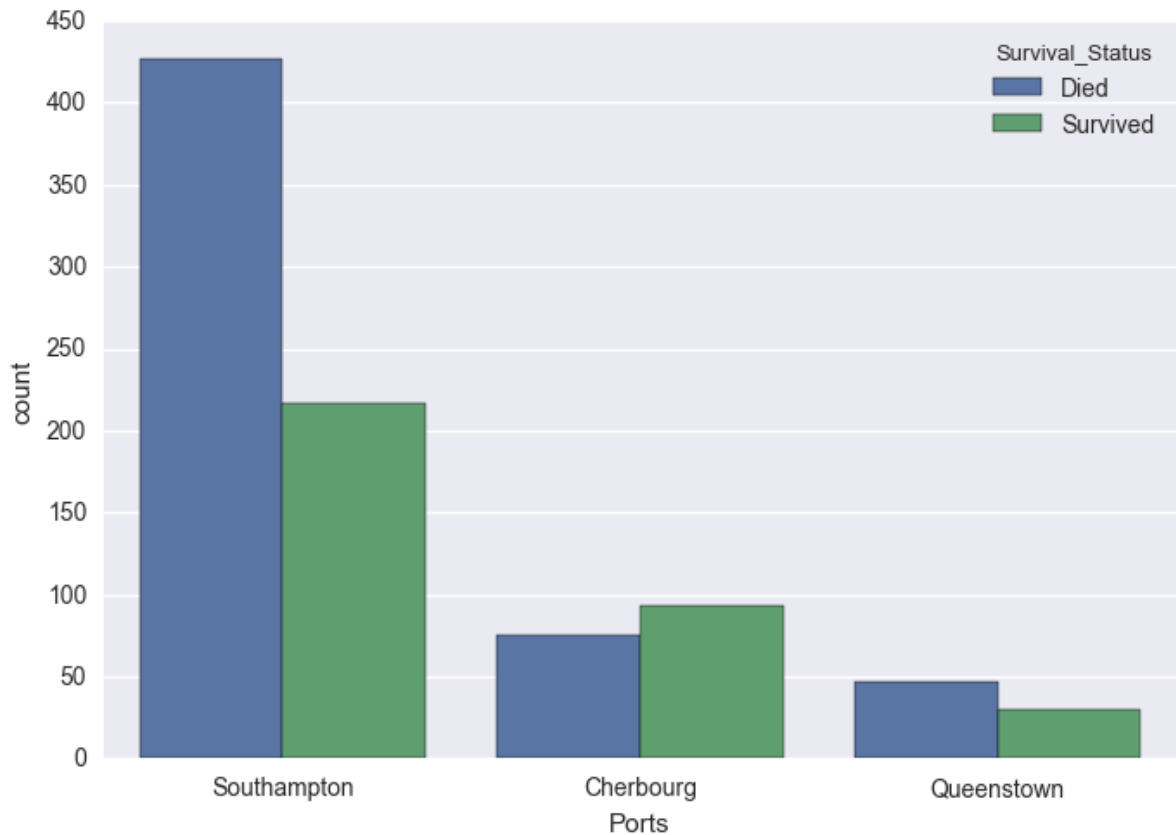
Cherbourg	0.498608
-----------	----------

Queenstown	0.490860
------------	----------

Southampton	0.473037
-------------	----------

Name: Survived, dtype: float64

```
In [87]: embarkation_axes = sns.countplot(x="Ports", hue="Survival_Status", data=titanic_data_frame)
```



Based on the above visualization it is clear that the **passengers boarding from Cherbourg had the highest survival rate** of about 55.35%

SECTION 5: LIMITATION OF THE ANALYSIS

Since the data set is for only 891 passengers onboard from the 2224 passengers on board the titanic we cannot reach a definite conclusion from the analysis. We can just analyze the sample data set and find patterns in our investigation which do form the basis in my report.

And as stated earlier that there were couple of passenger details missing in the data set like age, cabin details, port of embarkation. I chose to ignore the cabin and port details in my analysis but for missing ages I replaced them with 0 which might have had any impact in my statistical calculations and representations.

Besides the basic factors presented in the data set like passenger demographics, there might have been other factors which might have influenced passenger survival like the deck position, health conditions of the passengers etc. which might have influenced the survivability of the passengers.

SECTION 6: STATISTICAL TESTS

A statistical test provides a mechanism for making quantitative decisions about a process or processes. The intent is to determine whether there is enough evidence to "reject" a conjecture or hypothesis about the process.

The chi-squared test of independence tests whether two categorical variables are independent. The test of independence is commonly used to determine whether variables like education, political views and other preferences vary based on demographic factors like gender, race and religion. So we can apply this chi-squared test to two determine male and female passengers survivability on the Titanic.

We can make the following hypothesis:

Null Hypothesis, H_0 : Gender and Survivability are independent

Alternative Hypothesis, H_1 : Gender and Survivability are not dependent

```
In [88]: import scipy as sp

surviving_male_passenger_count=male_surviving_passengers_df['PassengerId'].count()
dead_male_passenger_count=male_passengers_df['PassengerId'].count()-surviving_male_passenger_count
surviving_female_passenger_count=female_surviving_passengers_df['PassengerId'].count()
dead_female_passenger_count=female_passengers_df['PassengerId'].count()-surviving_female_passenger_count
male_female_survival_array=np.array([[surviving_male_passenger_count,dead_male_passenger_count],
                                     [surviving_female_passenger_count,dead_female_passenger_count]])
print male_female_survival_array
sp.stats.chi2_contingency(male_female_survival_array)
```

```
[[109 468]
 [233  81]]
```

```
Out[88]: (260.71702016732104,
 1.1973570627755645e-58,
 1L,
 array([[ 221.47474747,  355.52525253],
        [ 120.52525253,  193.47474747]]))
```

The output by the `chi2_contingency` function gives us the results chi-square independence test:

- The first value indicates the chi-square value: 260.71702016732104
- The Second value is the p-value: 1.1973570627755645e-58
- The Third value is the degrees of freedom: 1L
- The fourth value is expected frequencies table:

Given the p-value is $1.1973570627755645e-58$ (due to chi-square value of 260.71702016732104) $< .00001$, the result is significant at p-value at $p < .05$ indicating there is an association between gender and survivability on the Titanic using the Titanic dataset given. With such a low p-value we can reject the null hypothesis which is that gender and survivalbility are independant and accept the alternative hypothesis which is gender and survivalbility are dependant of each other.

SECTION 7:CONCLUSIONS

Conclusions I drew from my analysis were:

- female passengers had more chances of surviving from the male passengers(proved from Statistical Test in Section 6)
- maximum number of adult passengers had died in the disaster and child passengers had survived more in the disaster
- elderly woman had 100% survival rate.
- woman/ child passengers had higher chances of survival.
- passengers in First class had higher chances of survival and passenger in Third class had lower chances of survival.
- female passengers had higher survival rate irrespective of the class they were travelling in
- adults had the highest overall rate of survival in all the classes, highest being in Class 1. Children and elderly have the highest rate of survival in Class 1
- passengers boarding from Cherbourg had the highest survival rate.

SECTION 8: REFERENCES

- [1] https://en.wikipedia.org/wiki/Timeline_of_young_people's_rights_in_the_United_Kingdom
(https://en.wikipedia.org/wiki/Timeline_of_young_people's_rights_in_the_United_Kingdom)
- [2] <http://seaborn.pydata.org/index.html> (<http://seaborn.pydata.org/index.html>)
- [3] <http://www.premierexhibitions.com/exhibitions/3/3/titanic-artifact-exhibition/blog/truth-behind-women-and-children-first> (<http://www.premierexhibitions.com/exhibitions/3/3/titanic-artifact-exhibition/blog/truth-behind-women-and-children-first>)
- [4]https://en.wikipedia.org/wiki/Women_and_children_first
(https://en.wikipedia.org/wiki/Women_and_children_first)
- [5]https://en.wikipedia.org/wiki/Passengers_of_the_RMS_Titanic
(https://en.wikipedia.org/wiki/Passengers_of_the_RMS_Titanic)
- [6]<https://www.encyclopedia-titanica.org/class-gender-titanic-disaster-1912~chapter-2.html>
(<https://www.encyclopedia-titanica.org/class-gender-titanic-disaster-1912~chapter-2.html>)

