Data Analysis of Titanic Event

by Grit on Oct 13, 2017

1. Question Posed

2. Handling the missing/NA values

3. Exploring Data and Visualization

3.1 Age, Sex, Pclass and Survival Rate

3.2 Other Factors and Survival Rate

3.3 What patterns can be found on the ticket class?

4. Drawing Conclusions

5. Limitations of Analysis

---

1. Question Posed

As we all konw, Titanic event is one of the most famous disasters throughout the world. More than 1500 people total died, including passengers and crew, and there were an estimated 2224 people aboard the ship. I'm curious about the survival rate patterns over age, sex and social-economic statu, e.g.

- During all the rescues, was the priority given to the children? What about the females?
- Were people with low social-economic status treated equally during the rescues?

Thus the follwing questions are my concern:

- **What variables are related to survival rate?**
  - What gender has a better survival rate?
  - Among the dead, how old is the oldest person？ And the youngest?
  - What are the survival rate patterns over age?
  - Are those passengers having higher ticket class likely to survive?
  - How does the family relationships (*siblings/spouses* and *parents/children*) affect the survival rate?
- **What patterns can be found on the ticket class?**
  - What's the distribution of ages and genders on each class?
  - What's the consumption situation of each class?
  - What ports did they embark from?

DataSet Source: https://www.kaggle.com/c/titanic/data (https://www.kaggle.com/c/titanic/data)

2. Handling the missing/NA values

Before investigating the data, we need to check the **missing/NA values** and fix them.

In [1]:

```
# First of all, We need to load datas from the csv file.
# Import required modules
%matplotlib inline
import numpy as np
import pandas as pd
from pandas import Series, DataFrame
import matplotlib.pyplot as plt
import seaborn as sns

# load the csv file
data = pd.read_csv('titanic-data.csv')
# Show the last five rows of data
data.tail()
```

Out[1]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7

In [2]:

```
# Identify the variables containing missing/NA values and the amount of surprising poin
ts
data.isnull().sum()
```

Out[2]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

As described in the table above, a total of three columns contain NA values, which are Age, Cabin and Embarked respectively.

- Among them, Age is our focus on the study, which can be handled with the mean age grouped by Pclass and Sex.
- Cabin can be ignored, because of little account.
- The two rows, where the values of Embarked are missing, will be deleted for the ignored count.

In [3]:

```
# Fill the NA values of data['Age'] with the mean age grouped by Pclass and Sex
data['Age'] = data.groupby(['Pclass', 'Sex'])['Age'].transform(lambda x: x.fillna(x.mea
n()))

# Go over the last five rows of data
data.tail()
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
886	887	0	2	Montvila, Rev. Juozas	male	27.00	0	0	211536	1
887	888	1	1	Graham, Miss. Margaret Edith	female	19.00	0	0	112053	3
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	21.75	1	2	W./C. 6607	2
889	890	1	1	Behr, Mr. Karl Howell	male	26.00	0	0	111369	3
890	891	0	3	Dooley, Mr. Patrick	male	32.00	0	0	370376	7

As we can see, the age of the passenger whose id is 889 has been fixed now, which was NA value before.

In [4]:

```
# Save the rows where data['Embarked'] is not null
data = data[data['Embarked'].notnull()]
```

3. Exploring Data and Visualization

3.1 Age, Sex, Pclass and Survival Rate

In [5]:

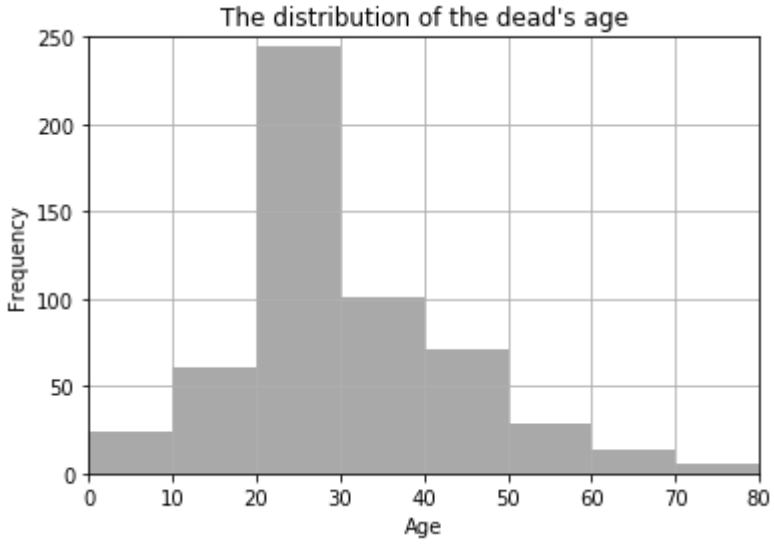
```
# Set the fixed bins
bin_values = np.arange(start=0, stop=90, step=10)
bin_values
```

Out[5]:

```
array([ 0, 10, 20, 30, 40, 50, 60, 70, 80])
```

In [6]:

```
# Select the data of the dead
dead = data[data['Survived']==0]

# Plot the distribution of the dead's age by given bins
dead['Age'].plot.hist(color='darkgray', bins=bin_values)
plt.title('The distribution of the dead\'s age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.xlim(0, 80)
plt.ylim(0, 250)
plt.grid()
plt.show()
```



The dead are mainly young people, especially those aged 20-30. The vast majority of the dead are less than 50 years old.

In [7]:

```
# Get the descriptive statistics for the age of the dead
dead['Age'].describe()
```

Out[7]:

```
count    549.000000
mean      30.025651
std       12.769635
min        1.000000
25%       22.000000
50%       26.507589
75%       37.000000
max       74.000000
Name: Age, dtype: float64
```

The average age of the dead is about 30, and 75% of them aren't more than 37 years old. The oldest deceased is as much as 74 years old, and the youngest is only 1 years old. We may go through their profiles by the follwing way.

In [8]:

```
# Get the profile of the youngest deceased
dead.loc[dead['Age'].argmin()]
```

Out[8]:

```
PassengerId                              165
Survived                                   0
Pclass                                     3
Name           Panula, Master. Eino Viljami
Sex                                     male
Age                                        1
SibSp                                      4
Parch                                      1
Ticket                               3101295
Fare                                 39.6875
Cabin                                    NaN
Embarked                                   S
Name: 164, dtype: object
```

We can see the youngest deceased is a baby boy with the lowest ticket class.

In [9]:

```
# Get the profile of the oldest deceased
dead.loc[dead['Age'].argmax()]
```

Out[9]:

```
PassengerId                    852
Survived                         0
Pclass                           3
Name           Svensson, Mr. Johan
Sex                           male
Age                             74
SibSp                            0
Parch                            0
Ticket                      347060
Fare                         7.775
Cabin                          NaN
Embarked                         S
Name: 851, dtype: object
```

The oldest deceased is a old man and he also has the lowest class of ticket.

In [10]:

```
# Calculate the count of the dead and survivors
count_survivors = data['Survived'].value_counts()
count_survivors
```

Out[10]:

```
0    549
1    340
Name: Survived, dtype: int64
```

In [11]:

```
# Get the overall survival rate and round the number
overall_survivalRate = count_survivors[1] / count_survivors.sum().astype(float)
overall_survivalRate = round(overall_survivalRate, 3)
print("The overall survival rate of the disaster is %s." % overall_survivalRate)
```

```
The overall survival rate of the disaster is 0.382.
```

In [12]:

```
# Define a function to group the data to see the distribution of survivors
# and get the survival rate of each group
def group_survivalDistr(dataset, var):
    grouped_survived = dataset.groupby([var, 'Survived']).size().unstack()
    grouped_survived.fillna(0, inplace=True)
    grouped_survived['ratio_survived'] = (grouped_survived[1] /
                                          (grouped_survived[0] +
                                           grouped_survived[1])).round(3)
    return grouped_survived
```

In [13]:

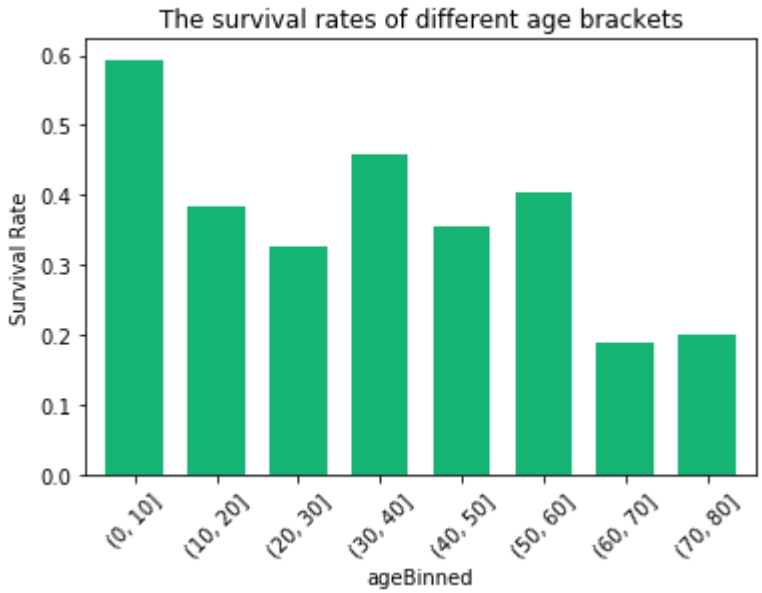
```
# Cut the age by the given bins and add ageBinned column
data['ageBinned'] = pd.cut(data['Age'], bins=bin_values)

# Apply the survival distribution function for ageBinned
survivalRate_by_ageBinned = group_survivalDistr(data, 'ageBinned')
survivalRate_by_ageBinned
```

Out[13]:

Survived	0	1	ratio_survived
ageBinned			
(0, 10]	26	38	0.594
(10, 20]	71	44	0.383
(20, 30]	248	120	0.326
(30, 40]	93	79	0.459
(40, 50]	69	38	0.355
(50, 60]	25	17	0.405
(60, 70]	13	3	0.188
(70, 80]	4	1	0.200

In [14]:

```
# Plot the survival rate of each group
survivalRate_by_ageBinned['ratio_survived'].plot.bar(color='#15B374', rot=45, width=0.7
)
# title of the plot
plt.title('The survival rates of different age brackets', fontsize=12)
# ylabel of the plot
plt.ylabel('Survival Rate')

plt.show()
```



The first bar of the two charts above shows the suvival rate of **infant children** is nearly 0.6, higher than other groups and the overall survival rate(0.384), which proves the special care for children in rescues.

I also want to konw what **gender** has a better survival rate. Does the conclusion apply equally to the children?

In [15]:

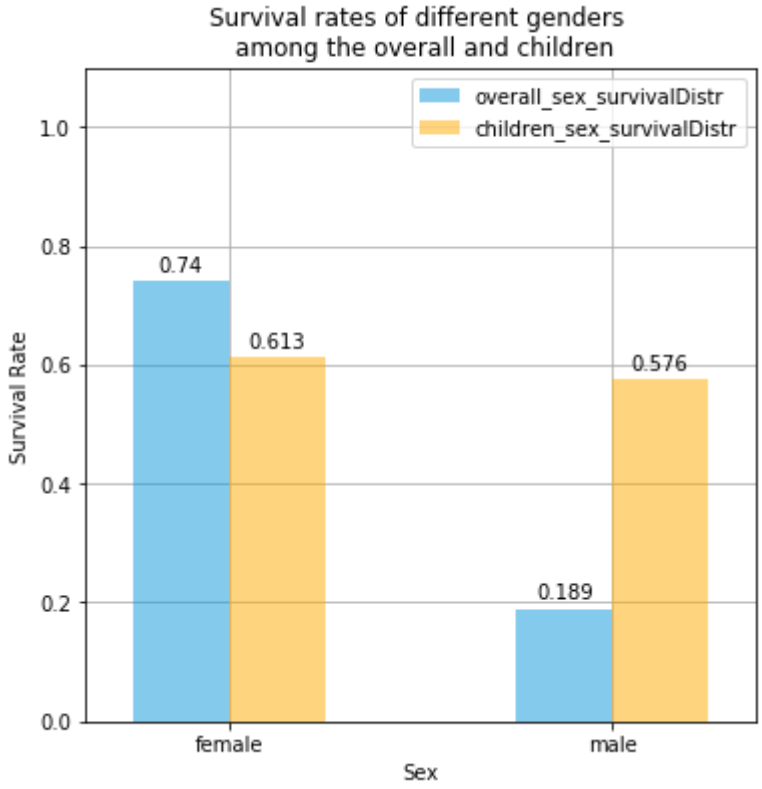
```
# Get the survival distributions of differrent genders and specify the dataset name
overall_sex_survivalDistr = group_survivalDistr(data, 'Sex')
overall_sex_survivalDistr.name = 'overall_sex_survivalDistr'

# Select the sub-dataset of the children with the highest survival rate
children = data[data['Age'] <= 10]

# Get the survival distributions of boys and girls and specify the dataset name
children_sex_survivalDistr = group_survivalDistr(children, 'Sex')
children_sex_survivalDistr.name = 'children_sex_survivalDistr'
```

In [16]:

```
# Create a function to compare the survival rates of two datasets
# grouped by the same two fileds, such as sex and survival rate
# and visualize the result

def compare_survivalDistr(dataset1, dataset1_name, dataset2, dataset2_name, title):

    # Set the indent and width for the bars and chart
    N = len(dataset1['ratio_survived'])
    ind = np.arange(N)
    width = 0.25
    graph_width = max(ind) * 3 + 3

    # Plotting the bars
    fig, ax = plt.subplots(figsize=[graph_width, 6])

    # Create a bar with the dataset
    # in indent ind + some width buffer
    rects1 = plt.bar(ind,
                     dataset1['ratio_survived'],
                     width,
                     color='#0897DC',
                     alpha=0.5,
                     label=dataset1.index[0])

    rects2 = plt.bar([i + width for i in ind],
                     dataset2['ratio_survived'],
                     width,
                     color='#FFAA00',
                     alpha=0.5,
                     label=dataset1.index[1])

    ax.set_ylabel('Survival Rate') # set the y-axis label
    ax.set_xlabel(dataset1.index.name) # set the x-axis label
    ax.set_xticks([i + width / 2 for i in ind]) # set the indent of the x ticks
    ax.set_xticklabels(dataset1.index) # set the labels for x ticks
    ax.set_axisbelow(True)

    # Create a function to attach a text label above each bar displaying its height
    def autolabel(rects):
        for rect in rects:
            height = rect.get_height()
            ax.text(rect.get_x() + rect.get_width()/2., height + 0.01, float(height),
                    ha='center', va='bottom')

    # Apply the function above to attch labels
    autolabel(rects1)
    autolabel(rects2)

    plt.xlim(-width, max(ind) + width * 2) # set the x-axis limit
    plt.ylim(0, 1.1) # set the y-axis limit
    plt.title(title, fontsize=12) # set the chart's title
    plt.legend([dataset1_name, dataset2_name], loc='upper right') # add the legend at t
he upper right
    plt.grid() # show the grid
    plt.show()
```

In [17]:

```
title = 'Survival rates of different genders \n among the overall and children' # title
 of plot
compare_survivalDistr(overall_sex_survivalDistr, overall_sex_survivalDistr.name,
                      children_sex_survivalDistr, children_sex_survivalDistr.name,
                      title) # compare the survival rates
```

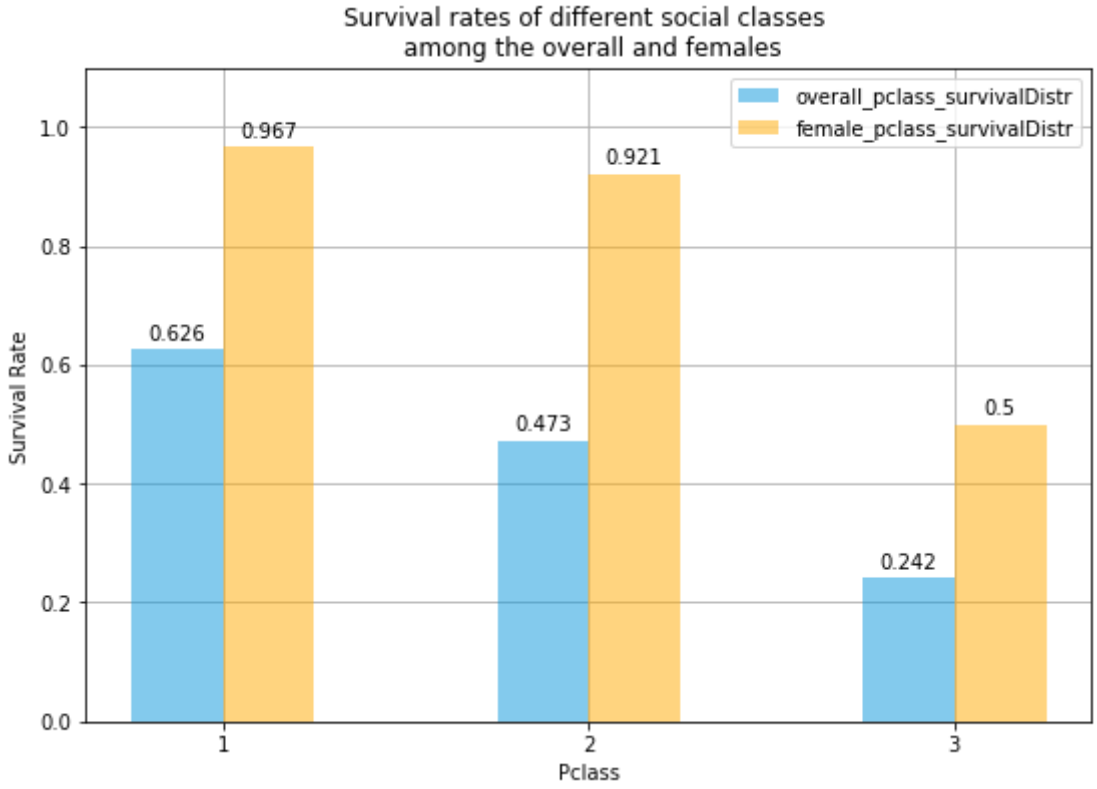


- As shown in the blue bar graphs, the overall survival rate of females (0.742) is more than 4 times that of males(0.189). It seems that **women** have acquired enough care in rescues.
- However, the preferential treatment for females doesn't apply to children, whose data is displayed in yellow. Among the children, the survival rates of boys(0.576) and girls(0.613) are **almost the same**. Interestingly, the girls' survival rate(0.613) is lower than the overall survival rate of females(0.742).

Still a small number of women are excluded from the list of survivors. So what is the common characteristic of those **servived women**? Does the characteristic include the **ticket class**, which represents the social-economic status?

In [18]:

```
# Get the survival distributions of differrent social-economic status and specify the d
ataset name
overall_pclass_survivalDistr = group_survivalDistr(data, 'Pclass')
overall_pclass_survivalDistr.name = 'overall_pclass_survivalDistr'

# Select the sub-dataset of females
females = data[data['Sex']=='female']

# Get the survival distributions of boys and girls and specify the dataset name
female_pclass_survivalDistr = group_survivalDistr(females, 'Pclass')
female_pclass_survivalDistr.name = 'female_pclass_survivalDistr'

# Set the title of the plot
title = 'Survival rates of different social classes \n among the overall and females'

# Compare the survival rates of all passengers and females in each social class
compare_survivalDistr(overall_pclass_survivalDistr, overall_pclass_survivalDistr.name,
                      female_pclass_survivalDistr, female_pclass_survivalDistr.name,
                      title)
```



- As shown the blue bar graphs, the survival rate of the passengers with ticket class 2 (0.473) is almost twice that of those people with ticket class 3 (0.242). The rate derived from ticket class 1 (0.630) is even higher. That means **the higher social-economic class a passenger belongs, the more likely it is for him/her to be saved from the disaster**, which we call it ***law of social-economic class and survival rate*** for the time being. Next, we'll realize the **universality** of the law.
- The orange bar graphs reveal that the survival rate of females is always higher than the overall rate on each social class, which illustrates a situation that **females have received priority** in rescues.
- Although females have got prior care in rescues, the degrees of care for **females of different social classes** are not the same. The survival rate of females with first two ticket classes are more than 0.9, while **only half** of females with the lowest ticket class can be rescued, which just coincides with the law above.

What about the care for **children with different ticket classes**?

In [19]:

```
# Get the survival distributions of children of differrent social classes
group_survivalDistr(children, 'Pclass')
```

Out[19]:

Survived	0	1	ratio_survived
Pclass			
1	1.0	2.0	0.667
2	0.0	17.0	1.000
3	25.0	19.0	0.432

The number of childrewn with ticket class 1 is too small to be considered. All of the children with ticket class 2 are saved, while the survival rate of children with ticket class 3 is only less than half. That is to say, the ***law of social-economic class and survival rate*** can apply to women and children.

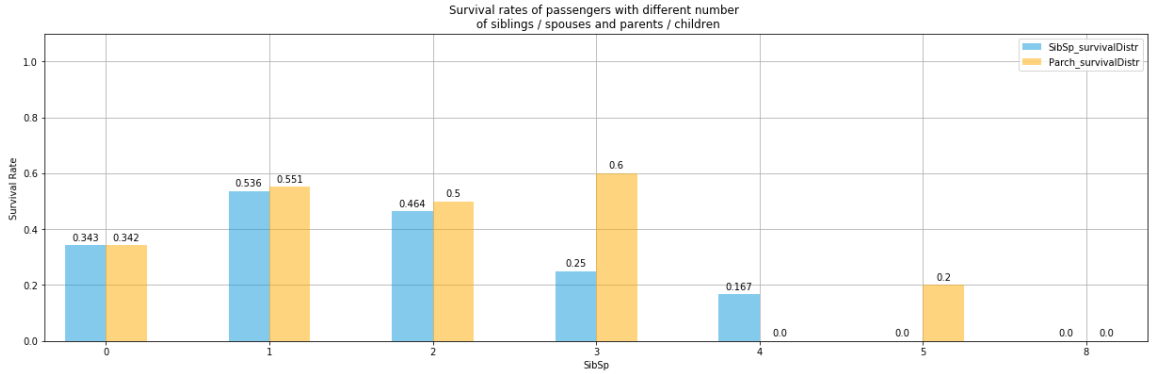
3.2 Other Factors and Survival Rate

In [20]:

```
# Get the survival distributions of differrent # of siblings / spouses
SibSp_survivalDistr = group_survivalDistr(data, 'SibSp')
SibSp_survivalDistr.name = 'SibSp_survivalDistr'

# Get the survival distributions of differrent # of parents / children
Parch_survivalDistr = group_survivalDistr(data, 'Parch')
Parch_survivalDistr.name = 'Parch_survivalDistr'

title = 'Survival rates of passengers with different number \n of siblings / spouses an
d parents / children'
```

In [21]:

```
# Compare the survival rates of siblings / spouses and parents / children
compare_survivalDistr(SibSp_survivalDistr, SibSp_survivalDistr.name,
                      Parch_survivalDistr, Parch_survivalDistr.name,
                      title)
```



As shown in the figure, those passengers with **more than 3 family members** have a lower chance of survival, less than the overall survival rate(0.384). I thought people in a big family may support each other better, so what's the reason for the lower survival chance. Is this related to their social classes or genders?

In [22]:

```
# Select the sub-dataset of passengers with more than 3 family members
familyRelat = data[(data['SibSp']>3) | (data['Parch']>3)]

# Group them by sex and pclass and get the count
grouped_familyRelat = familyRelat.groupby(['Sex', 'Pclass']).size().unstack()
grouped_familyRelat
```

Out[22]:

Pclass	**1**	**3**
Sex		
female	NaN	17.0
male	1.0	22.0

We can see almost all the passengers with more than 3 family members are **at the bottom of the socity**. That may be the reason for low survival chance.

In [23]:

```
# Get the survival distributions of differrent social classes of them
group_survivalDistr(familyRelat, 'Pclass')
```

Out[23]:

Survived	0	1	ratio_survived
Pclass			
1	1.0	0.0	0.000
3	35.0	4.0	0.103

These people with ticket class 3 have a proporlity of 0.103 to survive, obviously less than the overall 0.384.

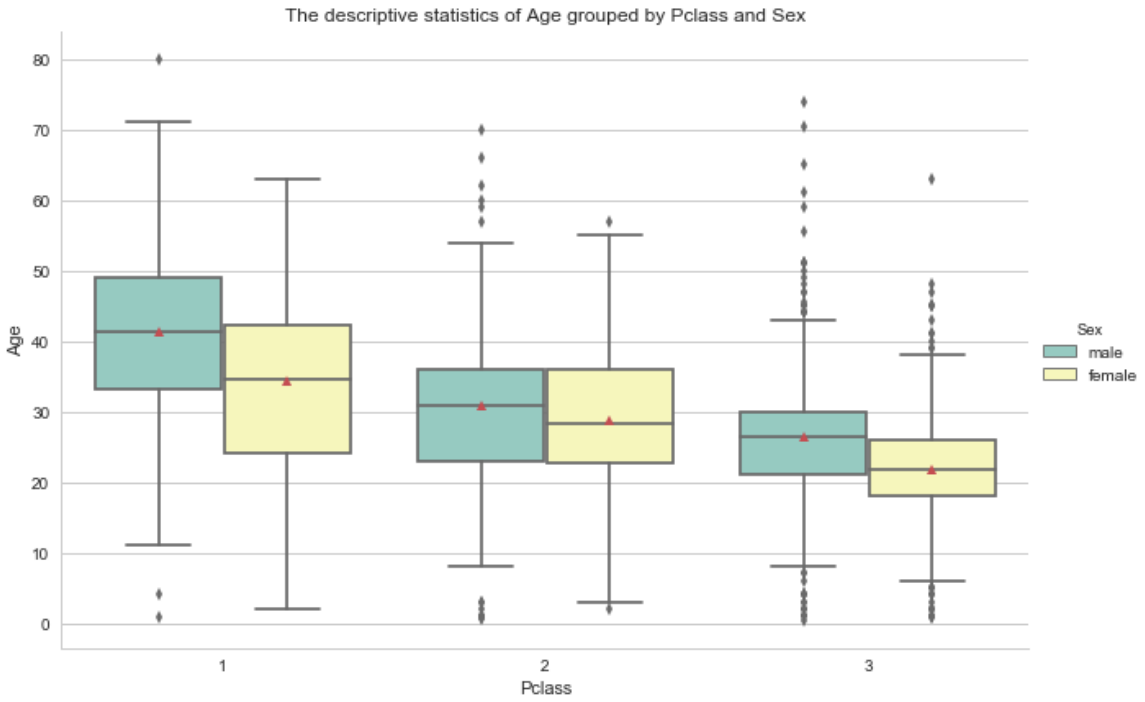
3.3 What patterns can be found on the ticket class?

In [24]:

```
# set the theme of boxplots
sns.set(style="whitegrid", color_codes=True)
```

In [25]:

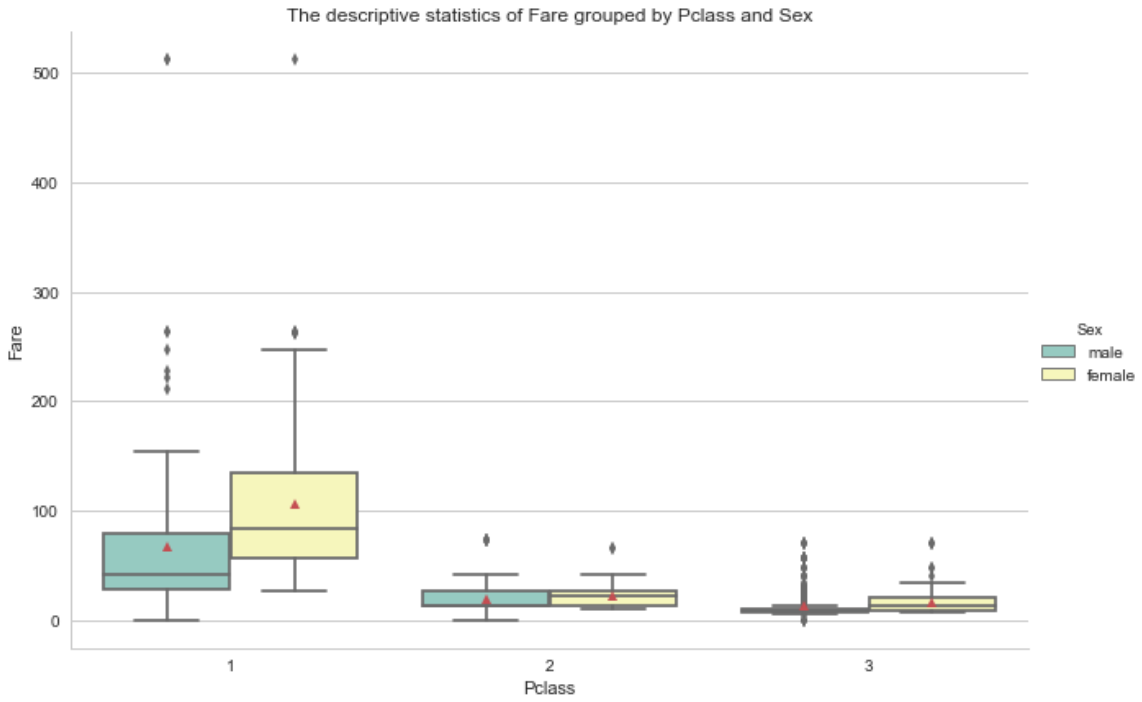
```
# Create boxplots
sns.factorplot(x='Pclass',
               y='Age',
               hue='Sex',
               kind='box',
               data=data,
               showmeans=True, # show the mean ages
               palette='Set3',
               size=6,
               aspect=1.5)

# Set title with matplotlib
plt.title('The descriptive statistics of Age grouped by Pclass and Sex')
plt.show()
```



- **The higher the social class, the greater the average age**. For males, he averege age of the lowest social class is about 27 years old, wihle that of the highest social class is about 42. It's well understood that people around the age of 40 are more likely to gain success in their careers after years of occupation and wealth accumulation.
- **The higher the social class, the larger dispersion of age the class has**. It's derived from the maximum standard deviation of the highest class.
- The mean age of females is greater than that of males on each social class. It's normal that a husband is a bit older than his wife.

In [26]:

```
# Create boxplots
sns.factorplot(x='Pclass',
               y='Fare',
               hue='Sex',
               kind='box',
               data=data,
               showmeans=True, # show the mean fares
               palette='Set3',
               size=6,
               aspect=1.5)

# Set title with matplotlib
plt.title('The descriptive statistics of Fare grouped by Pclass and Sex')
plt.show()
```



- On average, females spend more than males, which applys to each social class. Nevertheless, it is **only in the highest class** that the average fare of females is significantly higher than males! This is consistent with our consensus that women love consumption than men, enlarged in the rich.
- Also, it is **only in the highest class** that the fare has a obviously wide dispersion. Several fares are even more than $500, which is about 5 times the average fare of this group.

In [27]:

```
# the profiles of those who spend most
data[data['Fare']>500]
```

Out[27]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.

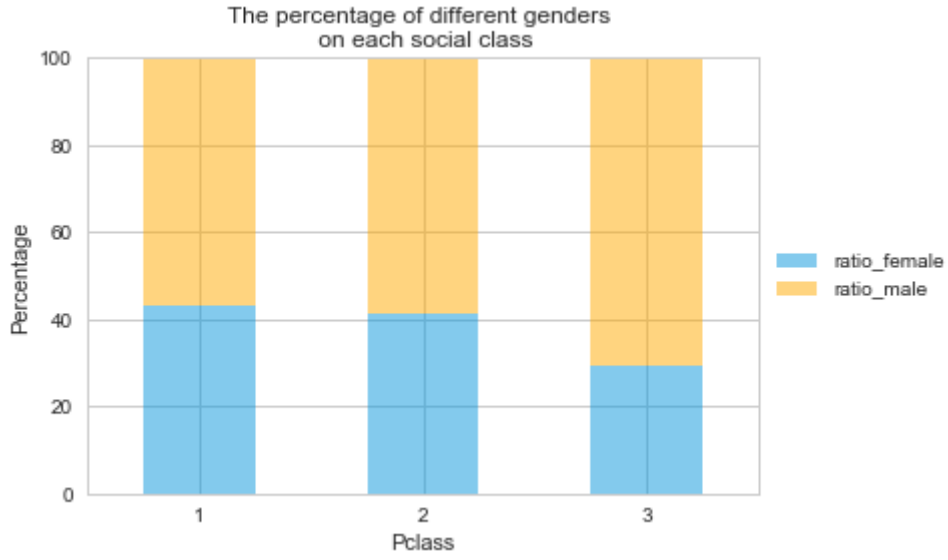
All those who spend most have been survived from the disaster.

In [30]:

```
# Create a function to plot the stacked percentage bars grouped by Pclass
def get_composed_percentage(var, figTitle):

    # Group the data by pclass and var and get the count
    count_grouped = data.groupby(['Pclass', var]).size().unstack()

    # Transform the frequency to the percentage along the columns
    count_grouped = count_grouped.apply(lambda x: x / x.sum() * 100, axis=1)

    # Prefix the keys
    count_grouped = count_grouped.add_prefix('ratio_')

    f, ax = plt.subplots(1)

    ind = np.arange(3)
    bar_width = 0.5
    tick_ind = [i for i in ind]

    ratio1 = count_grouped.iloc[:, 0]
    ax.bar(tick_ind, ratio1, bar_width,
           color='#0897DC',
           alpha=0.5,
           label=count_grouped.keys()[0])

    ratio2 = count_grouped.iloc[:, 1]
    ax.bar(tick_ind, ratio2, bar_width, bottom=ratio1,
           color='#FFAA00',
           alpha=0.5,
           label=count_grouped.keys()[1])

    # Check if the grouped DataFrame has 3 columns,
    # if so, add the 3rd stacked bars
    if len(count_grouped.columns) == 3:
        ratio3 = count_grouped.iloc[:, 2]
        ax.bar(tick_ind,
               ratio3,
               bottom=ratio1+ratio2,
               width=bar_width,
               color='r',
               alpha=0.5,
               label=count_grouped.columns[2])

    # Put a legend to the right of the current axis
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
    ax.set_xlabel('Pclass')
    ax.set_ylabel('Percentage')
    ax.set_title(figTitle)
    ax.set_axisbelow(True)

    plt.xticks(tick_ind, count_grouped.index)
    plt.xlim([min(tick_ind)-bar_width, max(tick_ind)+bar_width])
    plt.ylim(0, 100)

    plt.show()
```

In [33]:

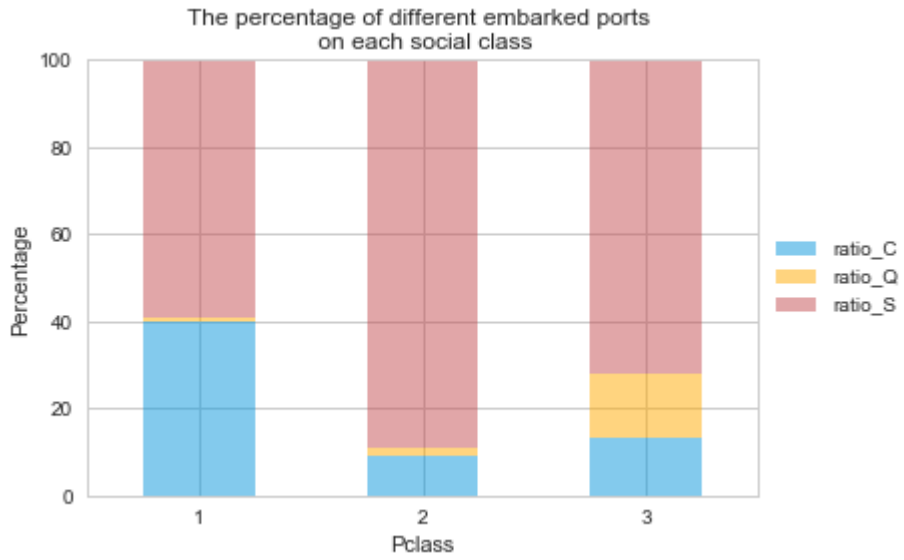
```
figTitle1 = 'The percentage of different genders \n on each social class'
get_composed_percentage('Sex', figTitle1)
```



Although the three groups of passengers are mainly male, the third class account for the highest proportion of male.

In [34]:

```
figTitle2 = 'The percentage of different embarked ports \n on each social class'
get_composed_percentage('Embarked', figTitle2)
```



Despite of the majority of passengers on each class embarking from Southampton(S) port, most people with ticket class 1 come from Cherbourg(C), and people with ticket class 3 are mainly from Queenstown(S) port. That reprents there are more rich in Cherbourg than in Queenstown.