Investigate Titanic Data Set

In this project we are going to work with the Titanic passenger data. We are interested to find out if there is any correlation between any of the data items. It makes me curious to know if we can make any conclusion about how the survival rate could be increased.

The steps I am going to follow are:

```
(i) visually inspect the data and find out if the dataset has any missing pieces,

(ii) gather various data propoerties, plot the extracted information visually as appli
cable to present the findings.
```

In [19]:
```
import numpy as np
import pandas as pd
%pylab inline
import matplotlib.pyplot as plt
import seaborn as sns
```

```
Populating the interactive namespace from numpy and matplotlib
```

At the beginning I am going to examine a few rows in the dataset to see how the data has been collected/stored, if there are missing data, etc.

In [20]:
```
titanic_df = pd.read_csv('titanic-data.csv')
"""
  Inspect the first few rows of the dataframe
"""
print titanic_df.head(10)
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3
5            6         0       3
6            7         0       1
7            8         0       3
8            9         1       3
9           10         1       2

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0
5                                   Moran, Mr. James    male   NaN      0
6                            McCarthy, Mr. Timothy J    male  54.0      0
7                     Palsson, Master. Gosta Leonard    male   2.0      3
8  Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)  female  27.0      0
9                Nasser, Mrs. Nicholas (Adele Achem)  female  14.0      1

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500              S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
5      0            330877   8.4583   NaN        Q
6      0             17463  51.8625   E46        S
7      1            349909  21.0750   NaN        S
8      2            347742  11.1333   NaN        S
9      0            237736  30.0708   NaN        C
```

As we see, there are NaN in the 'Age' and 'Cabin' data. This poses some limitation on how accurately we can analyze this data and try to find correlation between various variables. However, we will try to do our best with such limiation by eliminating these NaNs before being analysed.

Gather preliminary statistics

As a next step I am going to gather some statistics about the passengers:

• Total number of passengers

• Total number of female and male passengers

• Total number of people survived

• Number of female and male survivors

• Number of female and male deaths

In [21]:
```
no_of_passengers = titanic_df['PassengerId'].count()
print 'Total number of passengers = {}'.format(no_of_passengers)

no_of_females = (titanic_df[titanic_df['Sex'] == 'female'])['Sex'].count()
print 'Total number of females = {}'.format(no_of_females)

no_of_males = (titanic_df[titanic_df['Sex'] == 'male'])['Sex'].count()
print 'Total number of males = {}'.format(no_of_males)

no_of_survivors = (titanic_df[titanic_df['Survived'] == 1])['Survived'].count()
print 'Total number of people survived = {}'.format(no_of_survivors)

no_of_non_survivors = (titanic_df[titanic_df['Survived'] == 0])['Survived'].count()
print 'Total number of people died = {}'.format(no_of_non_survivors)

female_survivors = titanic_df[(titanic_df['Sex'] == 'female') & (titanic_df['Surviv
ed'] == 1)]
no_of_female_survivors = female_survivors['PassengerId'].count()
print 'Total number of female survivors = {}'.format(no_of_female_survivors)

female_deaths = titanic_df[(titanic_df['Sex'] == 'female') & (titanic_df['Survived'
] == 0)]
no_of_female_deaths = female_deaths['PassengerId'].count()
print 'Total number of female deaths = {}'.format(no_of_female_deaths)

male_survivors = titanic_df[(titanic_df['Sex'] == 'male') & (titanic_df['Survived']
 == 1)]
no_of_male_survivors = male_survivors['PassengerId'].count()
print 'Total number of male survivors = {}'.format(no_of_male_survivors)

male_deaths = titanic_df[(titanic_df['Sex'] == 'male') & (titanic_df['Survived'] ==
 0)]
no_of_male_deaths = male_deaths['PassengerId'].count()
print 'Total number of male deaths = {}'.format(no_of_male_deaths)
```

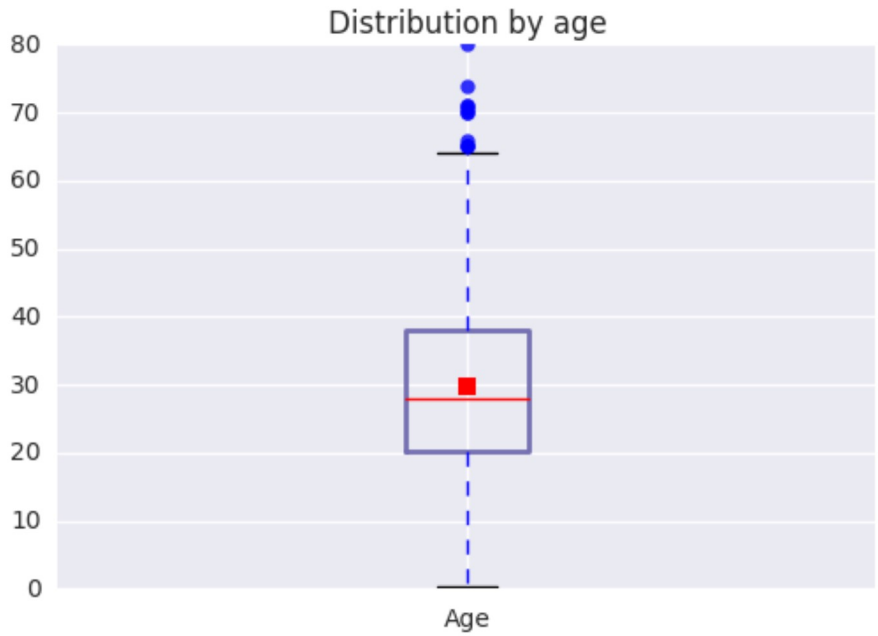
```
Total number of passengers = 891
Total number of females = 314
Total number of males = 577
Total number of people survived = 342
Total number of people died = 549
Total number of female survivors = 233
Total number of female deaths = 81
Total number of male survivors = 109
Total number of male deaths = 468
```

Investigate passengers by Age

I was curious to know the age group of the passengers. So I wanted to focus on this data and understand its distribution. As noted earlier, there are some missing data. Hence, I needed to eliminate such records before processing.

In [22]:
```
""" Look at the age distribution of the passengers """
""" How many 1-year old children were there """
print (titanic_df[titanic_df['Age'] == 1.0])['Age'].count()
""" Remove the NaNs from the age data """
age_series = (titanic_df[np.isfinite(titanic_df['Age'])])['Age']
print "Minimum age of the passengers = {}".format(age_series.min())
print "Maximum age of the passengers = {}".format(age_series.max())
print "Average age of the passengers = {}".format(age_series.mean())
bp = plt.boxplot(age_series, showmeans=True, labels = ['Age'])
plt.title("Distribution by age")
for flier in bp['fliers']:
    flier.set(marker='o', color='#e7298a', alpha=0.8)
for box in bp['boxes']:
    box.set(color='#7570b3', linewidth=2)
```

```
7
Minimum age of the passengers = 0.42
Maximum age of the passengers = 80.0
Average age of the passengers = 29.6991176471
```



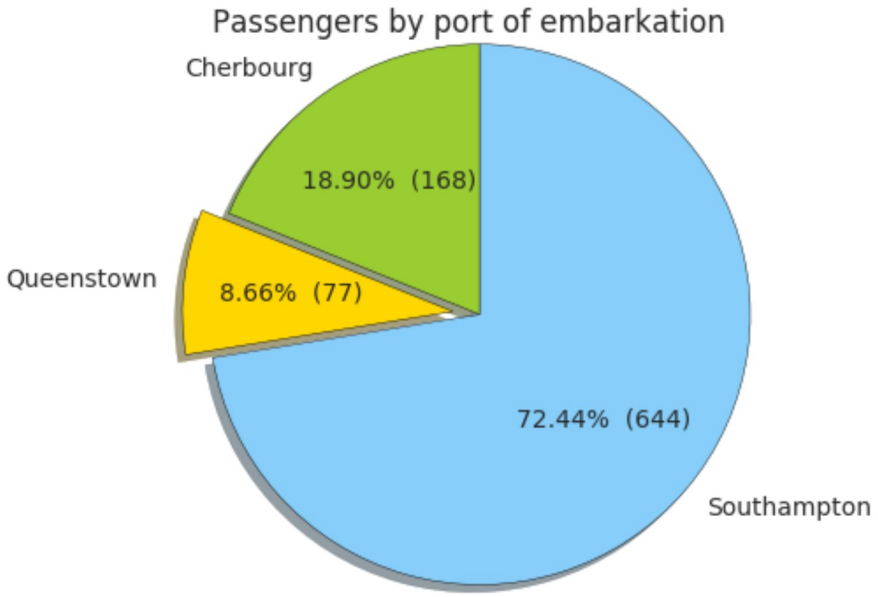
It looks like the passengers are fairly young (average and median age of the passengers are below or around 30!). There are also months old and very senior passengers travelling.

Now trying to find out where the passengers embarked the Titanic.

Investigate passengers by their port of embarkation

In [23]:
```
""" Passengers embarking from various ports """
titanic_df_embarked = titanic_df.dropna(subset = ['Embarked'])
passengers_C = (titanic_df_embarked[titanic_df_embarked['Embarked'] == 'C'])['Passe
ngerId'].count()
passengers_Q = (titanic_df_embarked[titanic_df_embarked['Embarked'] == 'Q'])['Passe
ngerId'].count()
passengers_S = (titanic_df_embarked[titanic_df_embarked['Embarked'] == 'S'])['Passe
ngerId'].count()
print "Passengers embarked at Cherbourg = {}".format(passengers_C)
print "Passengers embarked at Queenstown = {}".format(passengers_Q)
print "Passengers embarked at Southampton = {}".format(passengers_S)

""" Draw a pie-chart with the port of embarkation statistics.
    matplotlib provides a convenient way to denote the percentage number.
    However, the following function is written to customize the reporting of the
    percentage number to included the actual figure (Courtesey: Suggestion by the r
eviewer)
"""
def make_autopct(values):
  def my_autopct(pct):
      total = sum(values)
      val = int(round(pct*total/100.0))
      return '{p:.2f}%  ({v:d})'.format(p=pct,v=val)
  return my_autopct

labels = 'Cherbourg', 'Queenstown', 'Southampton'
sizes = [passengers_C, passengers_Q, passengers_S]
colors = ['yellowgreen', 'gold', 'lightskyblue']
""" Only "explode" the 2nd slice, i.e. 'Queenstown' passengers """
explode = (0, 0.1, 0)
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct=make_autopct([passengers_C, passengers_Q, passengers_S]), shadow=Tr
ue, startangle=90)
plt.title('Passengers by port of embarkation')
""" Set aspect ratio to be equal so that pie is drawn as a circle """
plt.axis('equal')
plt.show()
```

```
Passengers embarked at Cherbourg = 168
Passengers embarked at Queenstown = 77
Passengers embarked at Southampton = 644
```



Passengers are not all travelling alone. They have family members with them - their siblings, parents and children. It is a perfect time for all to spend time together for days-long journey on-board. They were looking forward to a very pleasant and safe sea voyage!

Passengers traveling with family members

In [24]:
```
""" Passengers travelling with family members """
titanic_df_families = titanic_df.dropna(subset = ['SibSp', 'Parch'])
titanic_df_families['Family'] = titanic_df_families['SibSp'] + titanic_df_families[
'Parch']
print titanic_df_families['Family'].describe()
""" The largest family is of size 10. Lets find out more about these passengers """
print titanic_df_families[titanic_df_families['Family']==10]
```

```
count    891.000000
mean       0.904602
std        1.613459
min        0.000000
25%        0.000000
50%        0.000000
75%        1.000000
max       10.000000
Name: Family, dtype: float64
     PassengerId  Survived  Pclass                                Name     Sex  \
159          160         0       3         Sage, Master. Thomas Henry    male
180          181         0       3       Sage, Miss. Constance Gladys  female
201          202         0       3                Sage, Mr. Frederick    male
324          325         0       3           Sage, Mr. George John Jr    male
792          793         0       3            Sage, Miss. Stella Anna  female
846          847         0       3           Sage, Mr. Douglas Bullen    male
863          864         0       3  Sage, Miss. Dorothy Edith "Dolly"  female

     Age  SibSp  Parch    Ticket   Fare Cabin Embarked  Family
159  NaN      8      2  CA. 2343  69.55   NaN        S      10
180  NaN      8      2  CA. 2343  69.55   NaN        S      10
201  NaN      8      2  CA. 2343  69.55   NaN        S      10
324  NaN      8      2  CA. 2343  69.55   NaN        S      10
792  NaN      8      2  CA. 2343  69.55   NaN        S      10
846  NaN      8      2  CA. 2343  69.55   NaN        S      10
863  NaN      8      2  CA. 2343  69.55   NaN        S      10
```

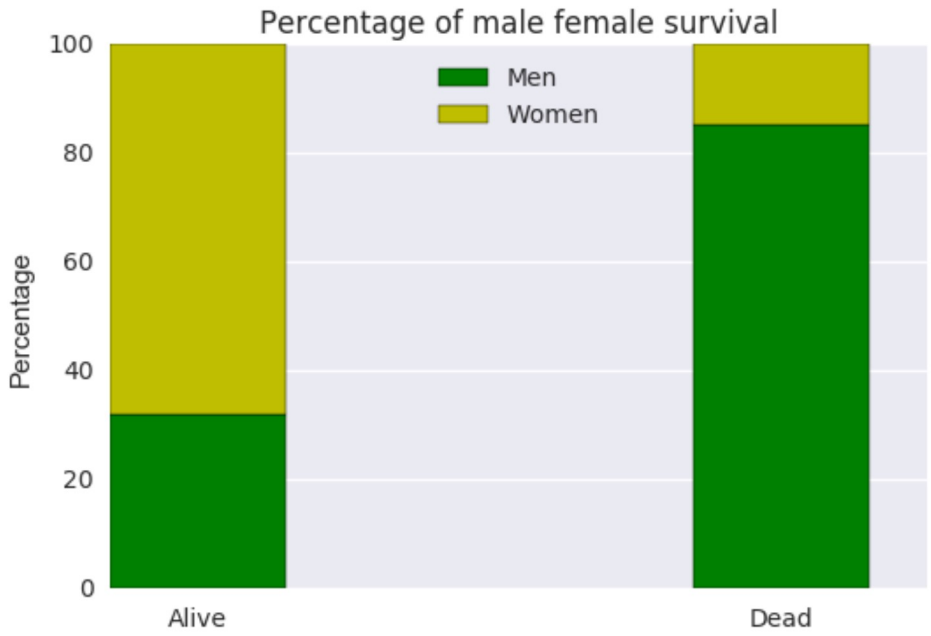
The largest family was travelling with ten passengers together. Unfortunately, the data shows that they did not make it to the destination!

Next, I am going to examine the survival data and find out if luck favored any gender.

Survival information of passengers by gender

In [28]:
```
""" Let us explore the passengers that survived or not """
survived_or_not = (titanic_df.groupby('Survived'))['PassengerId'].count()
survived = survived_or_not.loc[1]
not_survived = survived_or_not.loc[0]
print 'Total no. of people survived = {}'.format(survived)
print 'Total no. of people died = {}'.format(not_survived)
print ""
print ""

""" Create a dataframe for survival figures based on passenger gender """
survived_or_not_by_gender = titanic_df.groupby(['Survived', 'Sex'])
survived_or_not_by_gender_df = survived_or_not_by_gender['PassengerId'].count()
print "Dataframe of survival figures based on gender"
print survived_or_not_by_gender_df
print ""
print ""

""" Create a new simplified dataframe with gender-based survival information """
alive = pd.Series(survived_or_not_by_gender_df.loc[1])
dead = pd.Series(survived_or_not_by_gender_df.loc[0])
survival_df = pd.DataFrame({'Alive':alive, 'Dead':dead})
print "Dataframe Alive/dead"
print survival_df
print ""

""" Convert this dataframe to reflect the percentage information """
survival_df_percent = (survival_df/survival_df.sum())*100
print "Dataframe: Percentage of passengers by gender"
print survival_df_percent
print ""

""" Create slices of male and female survivals for plotting """
survival_df_percent_male = survival_df_percent.loc['male']
survival_df_percent_female = survival_df_percent.loc['female']

""" Plot a stacked bar for the passengers """
X = np.arange(2)
width = 0.3

p1 = plt.bar(X, survival_df_percent_male , width, color = 'g')
p2 = plt.bar(X, survival_df_percent_female, width, color = 'y', bottom = survival_d
f_percent_male)

plt.ylabel('Percentage')
plt.title("Percentage of male female survival")
plt.legend((p1[0], p2[0]), ('Men', 'Women'), loc = 'upper center')
plt.xticks(X + width/2., ('Alive', 'Dead'))
plt.show()
```

```
Total no. of people survived = 342
Total no. of people died = 549


Dataframe of survival figures based on gender
Survived  Sex
0         female     81
          male      468
1         female    233
          male      109
Name: PassengerId, dtype: int64


Datafram Alive/dead
        Alive  Dead
Sex
female    233    81
male      109   468

Dataframe: Percentage of passengers by gender
            Alive       Dead
Sex
female  68.128655  14.754098
male    31.871345  85.245902
```



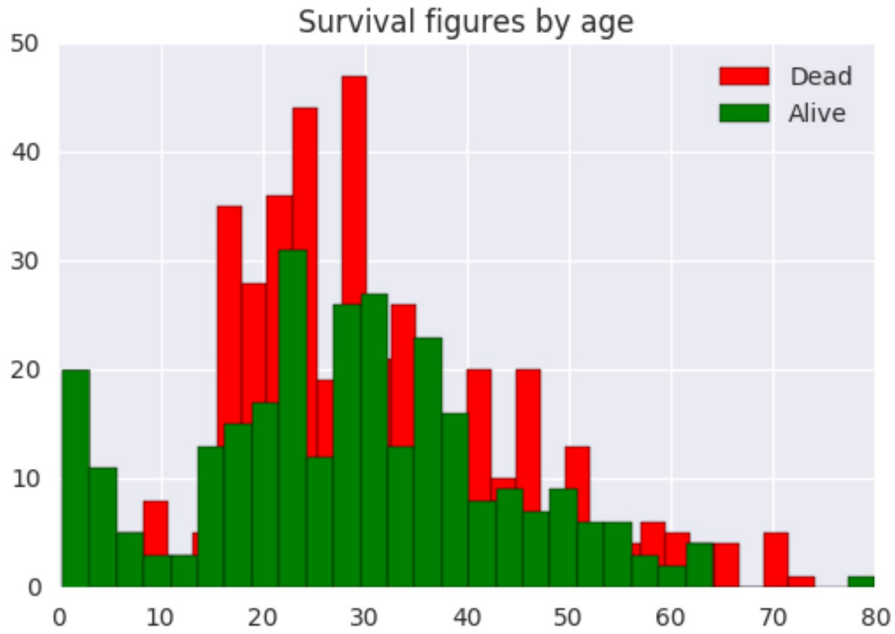
We notice that the female survival was more that that of male.

Next we are going to find out if there is any correlation between the survival and the passengers' age.

Survival figures by age

In [29]:
```
titanic_df_by_age = titanic_df.dropna(subset = ['Age'])
""" Create a series with age for those who survived """
survived_by_age = (titanic_df_by_age[titanic_df_by_age['Survived'] == 1])['Age']
""" Create a series with age for those who did not survive """
survived_not_by_age = (titanic_df_by_age[titanic_df_by_age['Survived'] == 0])['Age'
]
""" See how they comapre with histograms """
plt.hist(survived_not_by_age, bins = 30, color = 'r', label = 'Dead')
plt.hist(survived_by_age, bins = 30, color = 'g', label = 'Alive')
plt.legend(loc = 'upper right')
plt.title("Survival figures by age")
```

Out[29]: <matplotlib.text.Text at 0xa1b6b38>



The plot shows that very young (less than 10) and very aged (~80) passengers survived. Maybe they were given priorites to take up the boats. A large number of young passengers did not survive, maybe because they gave away to younger/older people to be rescued first.

Conclusion

The Titanic dataset has some limitations. The missing values in some cases, e.g. age, cabin etc. likely makes the analysis incomplete, to some extent. Based on the analysis, one would tend to infer a correlation between age and the rate of survival. However, based on some missing data, drawing such conclusions could be wrong. The statistical population (all Titanic passengers) parameters are known in this case. A z-test could possibly be conducted to establish if there is really a relationship between the the age and the survival rate.

References

i. Titanic dataset (Kaggle): https://www.kaggle.com/c/titanic/data (https://www.kaggle.com/c/titanic/data)

ii. How to drop rows of Pandas dataframe whose value of certain column is NaN http://stackoverflow.com/questions/13413590/how-to-drop-rows-of-pandas-dataframe-whose-value-of-certain-column-is-nan (http://stackoverflow.com/questions/13413590/how-to-drop-rows-of-pandas-dataframe-whose-value-of-certain-column-is-nan)

iii. Stacked bar plot: http://matplotlib.org/examples/pylab_examples/bar_stacked.html?highlight=stacked%20bar (http://matplotlib.org/examples/pylab_examples/bar_stacked.html?highlight=stacked%20bar)

iv. Pie chart plotting: http://matplotlib.org/examples/pie_and_polar_charts/pie_demo_features.html?highlight=pie%20chart (http://matplotlib.org/examples/pie_and_polar_charts/pie_demo_features.html?highlight=pie%20chart)

v. Reviewer's help with customizing pie and box plots.

In [ ]: