DATA ANALYSIS OF SURVIVORS OF TITANIC SHIPWRECK

This is the analysis of the survivors of the titanic ship wreck. The data for this analysis is being obtained from the link given below:

Sorry, Image can't be displayed

To download the data, Click Here!

Importing Packages

So in the above set of codes, we have imported three packages:

- numpy NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation

- matplotlib matplotlib is a another primarily used package in Data Sciences to plot the trends in form of different graphical representations, for example, histogram, pie chart, bar graph, etc.

- pandas pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way toward this goal. We deal with all dataframes and series in pandas.

- seaborn It is another package which works as a complimentary package with matplotlib. It beautifies and details the graphs.

- collections It is used to sort dictionaries.

How to download these packages, click here

In [976]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import collections
%matplotlib inline
sns.set()
```

Reading the excel or csv file:

In the above lines of code, initially **pd.read_csv('Name_of_the_file.csv')** is used in order to read the csv file from the path mentioned and store it in form of a dataframe in a variable named **df**

Picking up of necessary data:

x is another variable that stores the dataframe which are needed to carry out simple data analysis that we have conducted. there are many other columns in the csv file, like name, etc. which we do not need currently for our data analysis. Hence, we have filtered the data with only the columns we need.

<font color = blue> Further, we have printed the number of passengers on board, so we can display the *Sample Size* of the Data Analysis.

In [977]:
```
df=pd.read_csv('train.csv')

x=df[['PassengerId','Survived','Pclass','Sex','Age']]
total_rows=len(x.axes[0])
print("The analysis is carried out on " + str(total_rows+1) + " passengers travelling in titanic")
```

```
The analysis is carried out on 892 passengers travelling in titanic
```

Analysing the total no. of passengers who survived (Entire sample size)

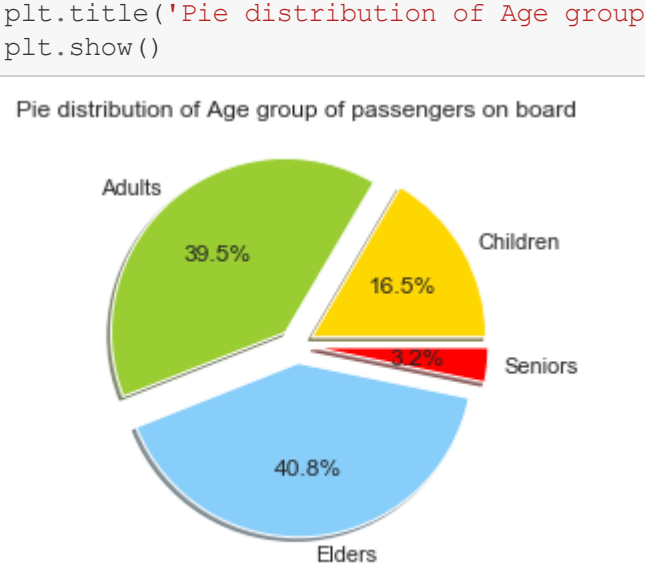
Divising of the Columns on the basis of constraints:

We have classified the data present in the **Age** Column into 4 main categories to find the age groups i,e, Children, Adults, Elders, Senior Citizens.

Further, the passengers are filtered according to their classess i.e. Class 1, Class 2, Class 3.

In [978]:
```
Children=x['Age']<18
Adults= (x['Age']>18) & (x['Age']<=30)
Elders=(x['Age']>30) & (x['Age']<60)
SeniorCitizens= x['Age']>60

class1= x['Pclass']==1
class2= x['Pclass'] == 2
class3= x['Pclass'] == 3
survived=x['Survived']==1
```

In [979]:
```
labels = 'Children','Adults','Elders','Seniors'
sizes = [len(x[Children].axes[0]),len(x[Adults].axes[0]),len(x[Elders].axes[0]),len(x[SeniorCitizens
].axes[0])]

plt.pie(sizes,labels=labels,shadow=True,colors = ['gold', 'yellowgreen', 'lightskyblue','red'],explo
de=(0.1,0.1,0.1,0.1),autopct='%1.1f%%')
plt.title('Pie distribution of Age group of passengers on board')
plt.show()
```



Pie Chart of Age groups on board

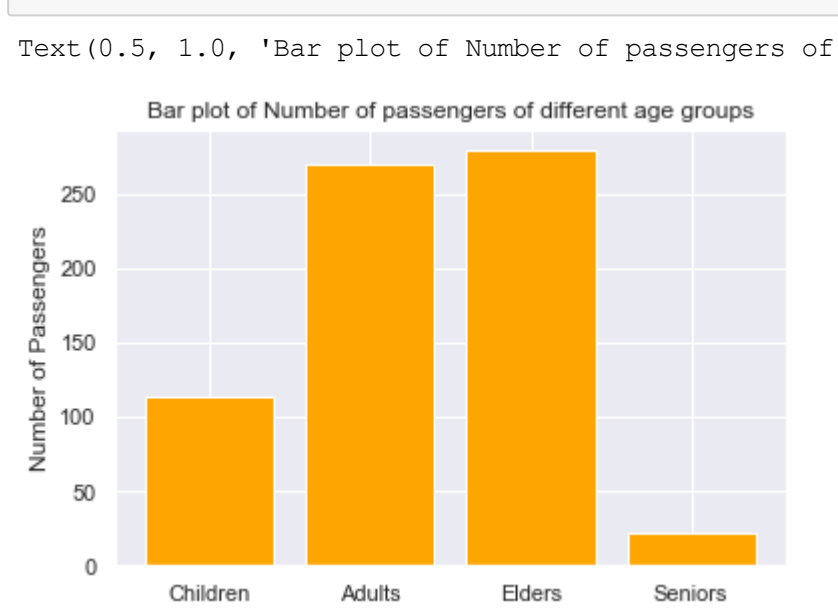
In the above pie chart, we can observe the distribution of total number of passengers on board. This is a representation of how the Main Sample is divided on the basis of age group.

- 40.8% of the passengers were Elders(i.e. 35 years to 60 years).
- 39.2% of the passengers were Adults(i.e. 18 years to 35 years).
- 16.5% of the passengers were Children(i.e. below 18 years).
- 3.2% of the passengerd were Seniors (i.e. above 60 years)

Bar graph showing different ages of passengers travelling onboard

In [980]:
```
objects = ['Children','Adults','Elders','Seniors']
y_pos = np.arange(len(objects))

plt.bar(y_pos, sizes, align='center', alpha=1, color = 'orange')
plt.xticks(y_pos, objects)
plt.ylabel('Number of Passengers')
plt.title('Bar plot of Number of passengers of different age groups')
```

Out[980]:
```
Text(0.5, 1.0, 'Bar plot of Number of passengers of different age groups')
```



About the bar graph:

The bar graph given above represents the division of the onboard passengers on the basis of their ages, just like the way we have done in the pie chart. This is similar to the pie chart shown above, rather than giving a percentage distribution it gives a quantitative distrubution

Number of passengers in each category can be shown with the help of the table given below:

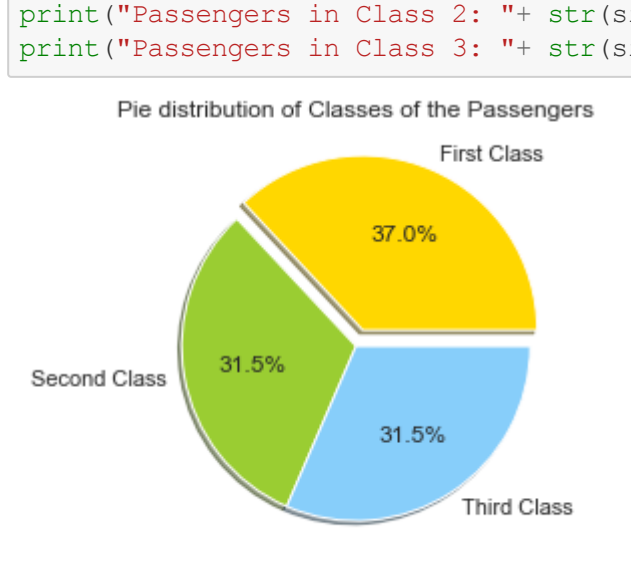
Table of number of passengers onboard

In [981]:
```
print("%-15s %s" %("Age group","Number of Passengers on board"))
for i in range(0,4):
    print("%-15s %s" %(objects[i],str(sizes[i])) )
```

```
Age group       Number of Passengers on board
Children        113
Adults          270
Elders          279
Seniors         22
```

Different Classes of Passengers on board

In [985]:
```
size_class= [len(x[class1].axes[0]),len(x[class2].axes[0]),len(x[class2].axes[0])]
plt.title('Pie distribution of Classes of the Passengers')
plt.pie(size_class, labels= ['First Class','Second Class','Third Class'],autopct='%1.1f%%', shadow=T
rue,colors = ['gold', 'yellowgreen', 'lightskyblue'],explode=(0.1,0,0))
plt.show()

print("Passengers in Class 1: "+ str(size_class[0]))
print("Passengers in Class 2: "+ str(size_class[1]))
print("Passengers in Class 3: "+ str(size_class[2]))
```




```
Passengers in Class 1: 216
Passengers in Class 2: 184
Passengers in Class 3: 184
```

Analysis of Passengers who survived

Above, we have analysed the datasets for all the passengers on board. Now, we will be analysing the passengers who **survived.**

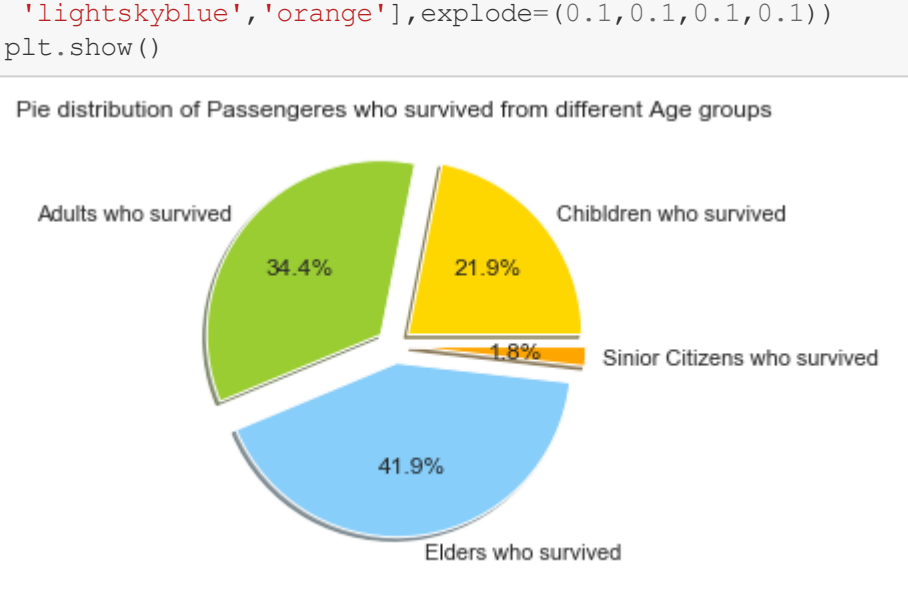
In the Analysis done below the following analysis have been conducted:

- Pie distribution of the percentage survivors from different *Age Categories*
- A double bar graph representing the number of passengers on board alongside the number of passengers who survived
- Survivor ratio chart i.e. a bar chart between the ratio of survivors:onboard passengers on y axis and different age categories as the pillars.
- Pie distribution of the percentage of survivors from different travel *Classes*
- Line of Age variation vs frequency of survivors
- Histogram of age variations with the survivor frequency

In [984]:
```
Children_survived=x[(Children) & (survived)]
Adults_survived=x[(Adults) & (survived)]
Elders_survived=x[(Elders) & (survived)]
Seniors_survived=x[(SeniorCitizens) & (survived)]
```

In the above lines of code, we have filtered the passengers who have survived according to different age groups which which we used earlier

In [982]:
```
size_age_survived= [len(Children_survived.axes[0]),len(Adults_survived.axes[0]),len(Elders_survived.
axes[0]),len(Seniors_survived.axes[0])]
plt.title('Pie distribution of Passengeres who survived from different Age groups')
plt.pie(size_age_survived, labels= ['Chibldren who survived','Adults who survived','Elders who survi
ved','Sinior Citizens who survived'],autopct='%1.1f%%', shadow=True,colors = ['gold', 'yellowgreen',
 'lightskyblue','orange'],explode=(0.1,0.1,0.1,0.1))
plt.show()
```



Understanding the Pie Chart given above:

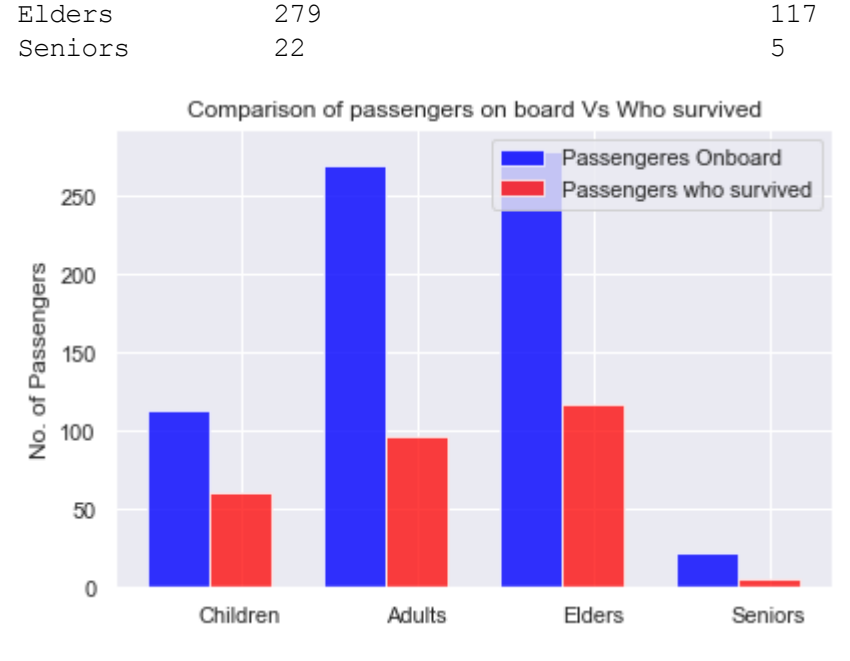
We can conclude from the above pie chhart that maximum passengers who survived were Elders i,e, 41.9 % of the total survived population.

Plotting Double Bar Chart

A double bar chart is the one which can represent the change a population sample before adn after by plotting it simultaneously. Here we can study the downfall in the populations in different age group after the shipwreck.

In [983]:
```
n_groups = 4
initial = sizes
final = size_age_survived

print("%-15s %-30s %-32s %s" %("Age group","No. of Passengers onboard","No. of passengers who surviv
ed","Total Deaths"))
for i in range(0,4):
    print("%-15s %-30s %-32s %s" %(objects[i],str(initial[i]),str(final[i]), str(initial[i]-final[i
])))
# create plot
fig, ax = plt.subplots()
index = np.arange(n_groups)
bar_width = 0.35
opacity = 0.9

rects1 = plt.bar(index, initial, bar_width,
                 alpha=0.8,
                 color='blue',
                 label='Passengeres Onboard')

rects2 = plt.bar(index + bar_width, final , bar_width,
                 alpha=0.75,
                 color='red',
                 label='Passengers who survived')

plt.ylabel('No. of Passengers')
plt.title('Comparison of passengers on board Vs Who survived')
plt.xticks(index + bar_width, ('Children','Adults','Elders','Seniors'))
plt.legend()

plt.tight_layout()

plt.figure()
plt.show()
```

```
Age group       No. of Passengers onboard      No. of passengers who survived   Total Deaths
Children        113                            61                               52
Adults          270                            96                               174
Elders          279                            117                              162
Seniors         22                             5                                17
```




```
<Figure size 432x288 with 0 Axes>
```

Understanding the chart:

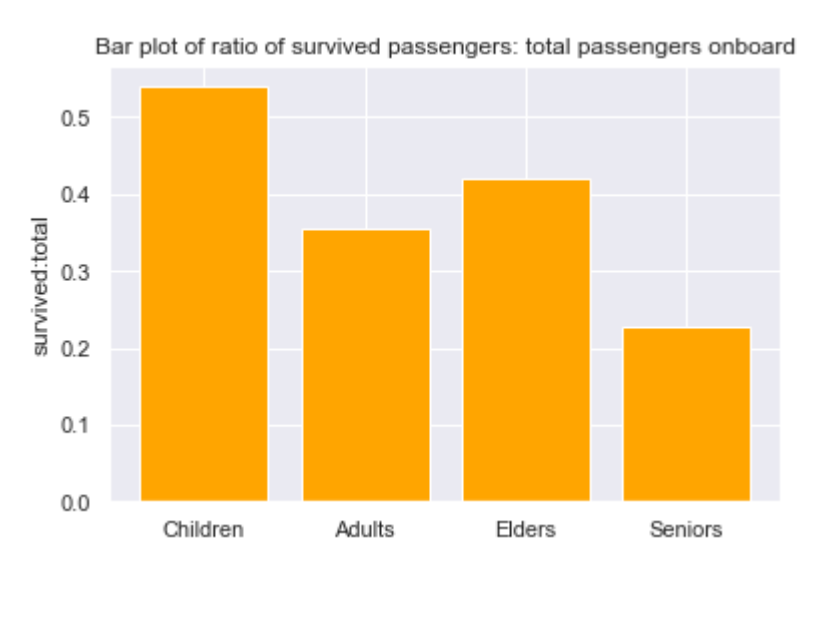
The double bar plot shows the difference or the downfall in the passenger number in each category after the shipwreck. we can see that how every age group has a downfall in the number of passengers. Seniors have a major downfall while the Children have marginal downfall.

Survivor ratio chart

(i.e. a bar chart between the ratio of survivors:onboard passengers on y axis and different age categories as the pillars.)

In [986]:
```
ratio=[]
for j in range(0,4):
    r=final[j]/initial[j]
    ratio.append(r)

plt.bar(y_pos, ratio, align='center', alpha=1, color = 'orange')
plt.xticks(y_pos, objects)
plt.ylabel('survived:total')
plt.title('Bar plot of ratio of survived passengers: total passengers onboard')
```

Out[986]:
```
Text(0.5, 1.0, 'Bar plot of ratio of survived passengers: total passengers onboard')
```

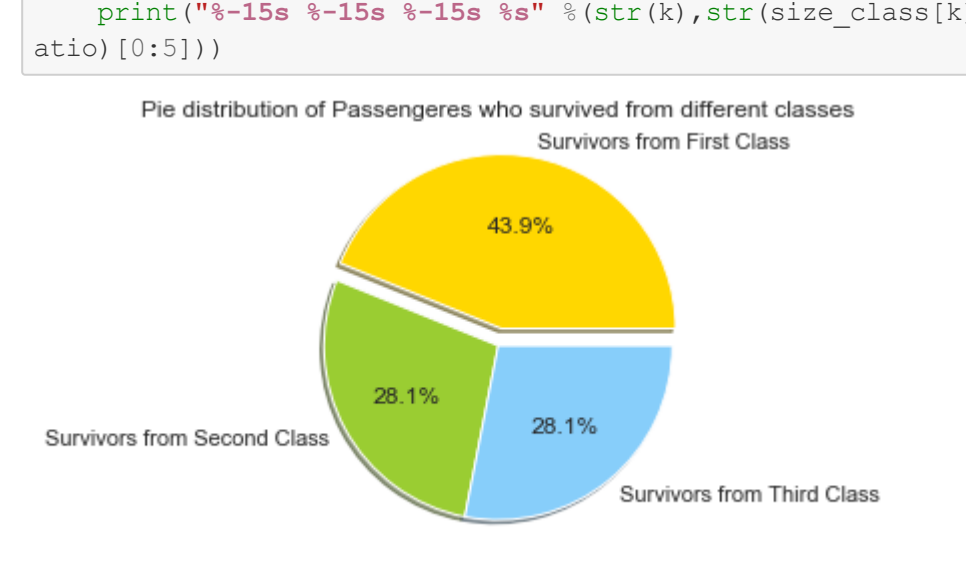


Conclusion about survival ratio

- Hence we can strongly conclude that maximum survival ratio or maximum number of survivals with respect to initial population of the category were seen in children.
- Another conclusion can be made is that the seniors or the passengers above 60 years of age had least survival ratio.

In [987]:
```
class1_survived= x[(class1) & (survived)]
class2_survived = x[(class2) & (survived)]
class3_survived = x[(class3) & (survived)]
```

In [988]:
```
size_class_survived= [len(class1_survived.axes[0]),len(class2_survived.axes[0]),len(class2_survived.
axes[0])]
plt.title('Pie distribution of Passengeres who survived from different classes')
plt.pie(size_class_survived, labels= ['Survivors from First Class','Survivors from Second Class','Su
rvivors from Third Class'],autopct='%1.1f%%', shadow=True,colors = ['gold', 'yellowgreen', 'lightsky
blue'],explode=(0.1,0,0))
plt.show()
print("%-15s %-15s %-15s %s" %("Class","Onboard","Survived","Survived:Onboard"))
for k in range(0,3):
    class_ratio=size_class_survived[k]/size_class[k]
    print("%-15s %-15s %-15s %s" %(str(k),str(size_class[k]),str(size_class_survived[k]),str(class_r
atio)[0:5]))
```




```
Class           Onboard         Survived        Survived:Onboard
0               216             136             0.629
1               184             87              0.472
2               184             87              0.472
```

Given above is a tabular representation of survivals on the basis of Classes

Conclusion:

- Maximum survival ratio is observed in Class 1. It means that the class 1 passengers survived majorly with respect to their initial population
- There was equal survival ratio in class 2 and class 3

A function that counts the frequency of the data in the given column and stores it in form of dictionary

In [989]:
```
def count_elements(seq) -> dict:
     """Tally elements from `seq`."""
     hist = {}
     for i in seq:
         hist[i] = hist.get(i, 0) + 1
     return hist
```

Here we can store all the data of the survivors in a variable called survived_filters

We have printed the number of passengers whoo survived dropna() function is used to drop the missing values of age in order to get a continuous line in the graph

In [992]:
```
survived_filters=x[survived]
print("No. of passengers who survived = " + str(len(survived_filters.axes[0]) + 1 ))
survived_age= survived_filters['Age']
ar = np.array(survived_age.dropna())
age_fr=count_elements(list(ar))
```

```
No. of passengers who survived = 343
```

Arranging the dictionary or sorting it, we need to use following commands:

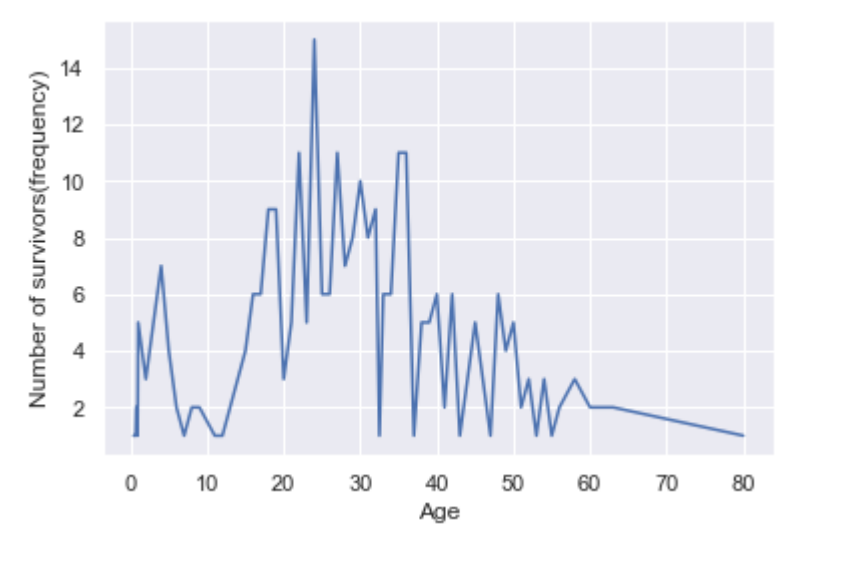
collections is a package that is being imported in the very first lines of code. We have used it here in order to sort the dictionary on the basis of items.

Plotting the Line Graph

A line graph is being plotted with Age on X Axis and No. of survivors acting as the frequency corresponding to that age. In this line graph, for every age an approximate frequency can be seen.

(Note: For many such age values which do not exist in the data, number of survivors are still shown)

In [991]:
```
o_age_fr= collections.OrderedDict(sorted(age_fr.items()))
X_axis= list(o_age_fr.keys())
Y_axis= list(o_age_fr.values())
plt.xlabel('Age')
plt.ylabel('Number of survivors(frequency)')
plt.plot(X_axis,Y_axis)
plt.show()
```



Histogram Distribution of Number of Survivors according to ages

Histogram is being plotted with 10 year taken as a bin on x axis, hence we get 10 different intervals, 0-10,10-20,20-30,... and so on

Corresponding to every age interval, a frequency is being given on the Y axis.

In [990]:
```
plt.xlabel('Age Group(bins)')
plt.ylabel('Number of survivors(frequency)')
plt.hist(ar)
```