IC 152

Computing & Data Science

Lab 6

Mohit Verma

B20215

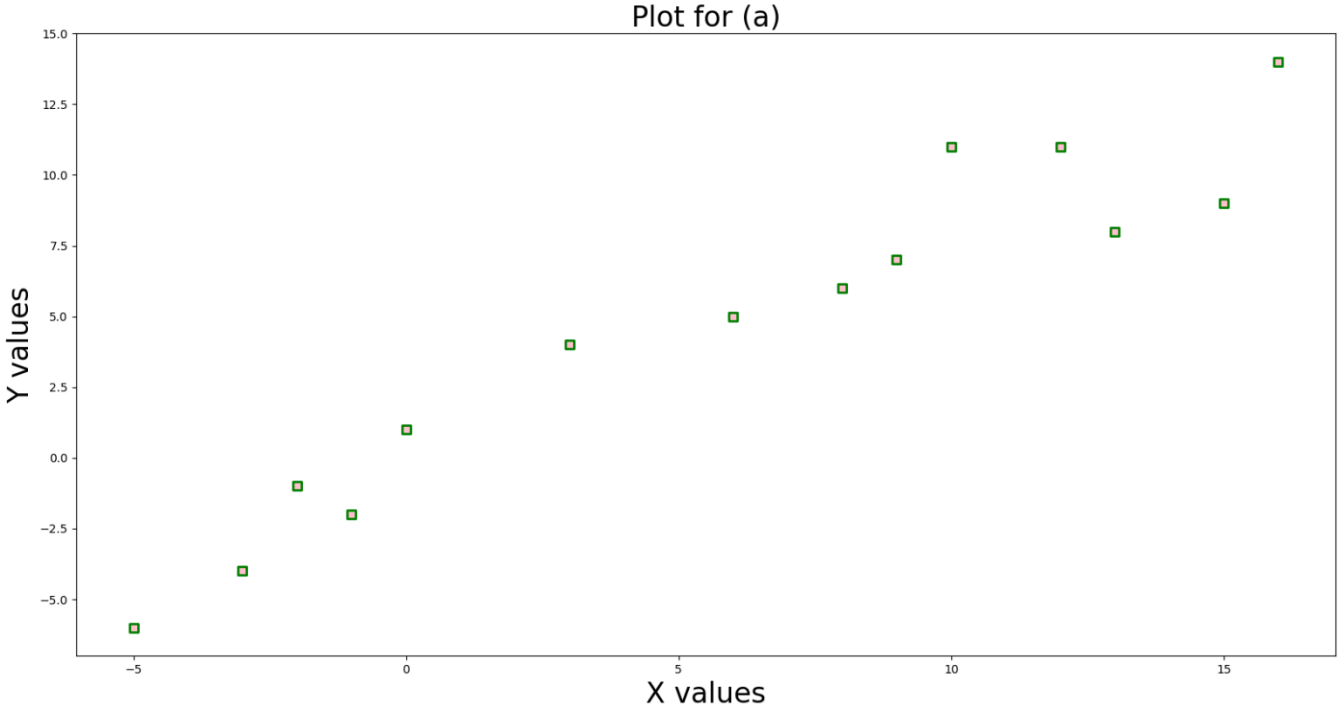
Instructor- Dr.Padmanabhan Rajan

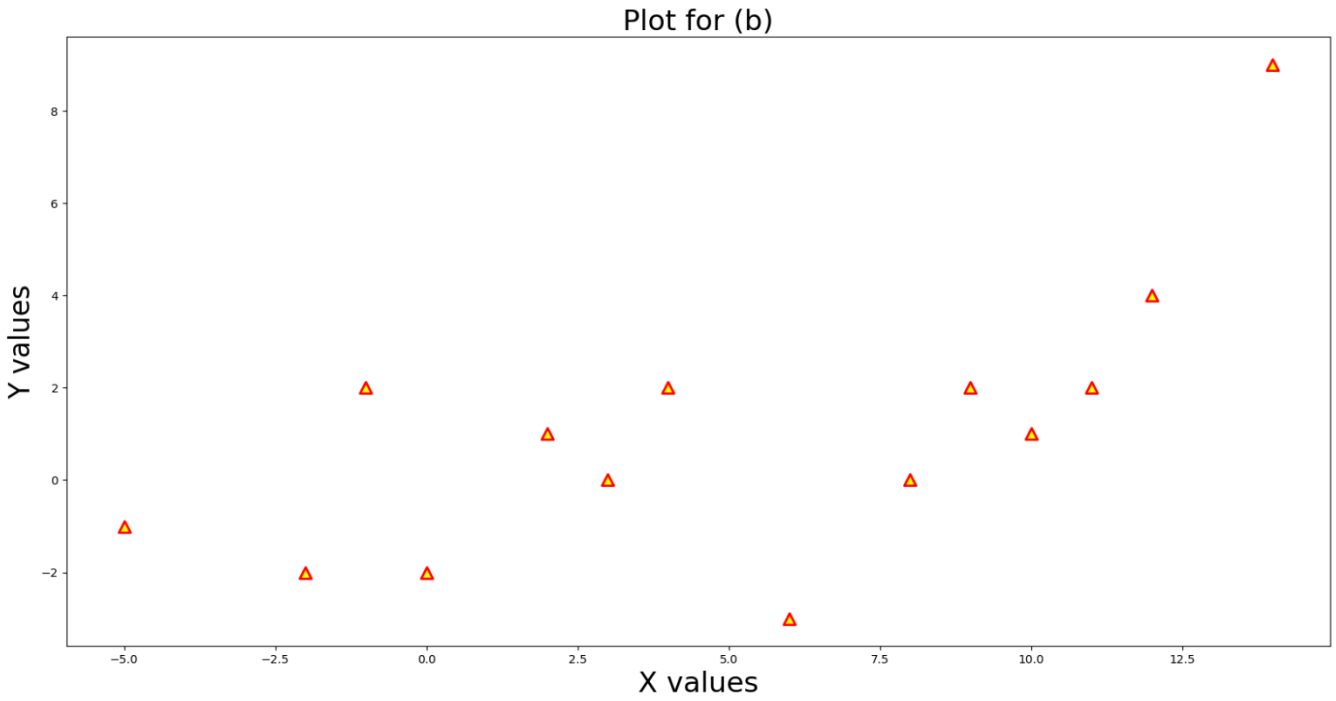
Questions & Answers

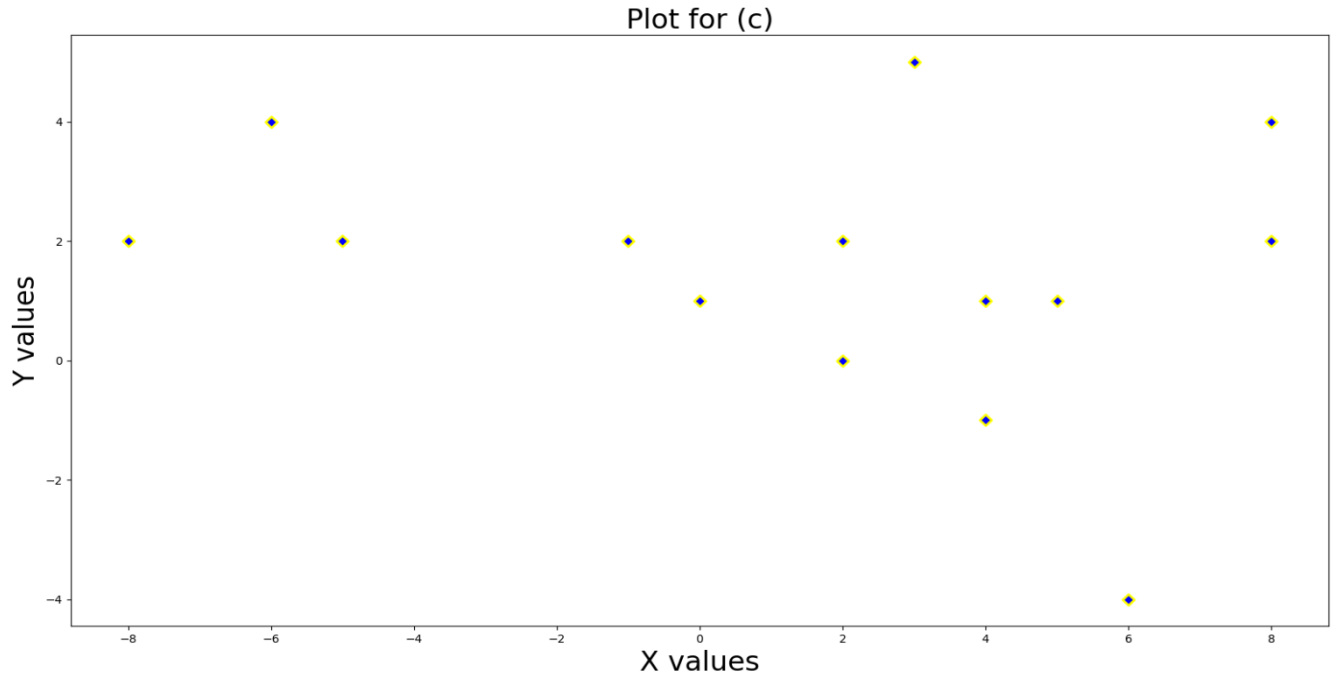
Q1 (i) Plot the variables (x and y) as scatter plots given below.

(ii) Find out the correlation between the different cases of two variables (x and y) as given below and use the equation (mentioned above). Analyse the numerical value and scatter plots of the variables (i.e., if correlation is positive, y should increase with increase in x).

Ans 1 (a) Scatter plots







Ans 1 (ii)

```
In [6]: runfile('C:/Users/verma/OneDrive/Desktop/LAB 6/untitled1.py', wdir='C:/Users/verma/OneDrive/Desktop/LAB 6')
Reloaded modules: jupyter_client.session, zmq.eventloop, zmq.eventloop.ioloop, tornado.platform,
tornado.platform.asyncio, tornado.gen, zmq.eventloop.zmqstream, jupyter_client.jsonutil, jupyter_client.adapter,
spyder, spyder.pil_patch, PIL, PIL._version, PIL.Image, PIL.ImageMode, PIL.TiffTags, PIL._binary, PIL._util,
PIL._imaging, cffi, cffi.api, cffi.lock, cffi.error, cffi.model, IPython.lib.guisupport, IPython.external,
IPython.external.qt_for_kernel, IPython.utils.version, IPython.external.qt_loaders
Correlation for (a) 0.9591483545329047
Correlation for (b) 0.6470857165183262
Correlation for (c) -0.2518839194843887

In [7]:
```

We observe that Correlation for part(a) is 0.9591 which is positive and very high(close to 1) which shows that the value of y increases with x and we have a strong relation between x and y values.

For part(b) Correlation is 0.6470 which is positive which shows that value of y increases as x increases.

For part (c) the Correlation -0.2518 which is negative which indicates that the value of y decreases with increase in x. It has weak relation between x and y.

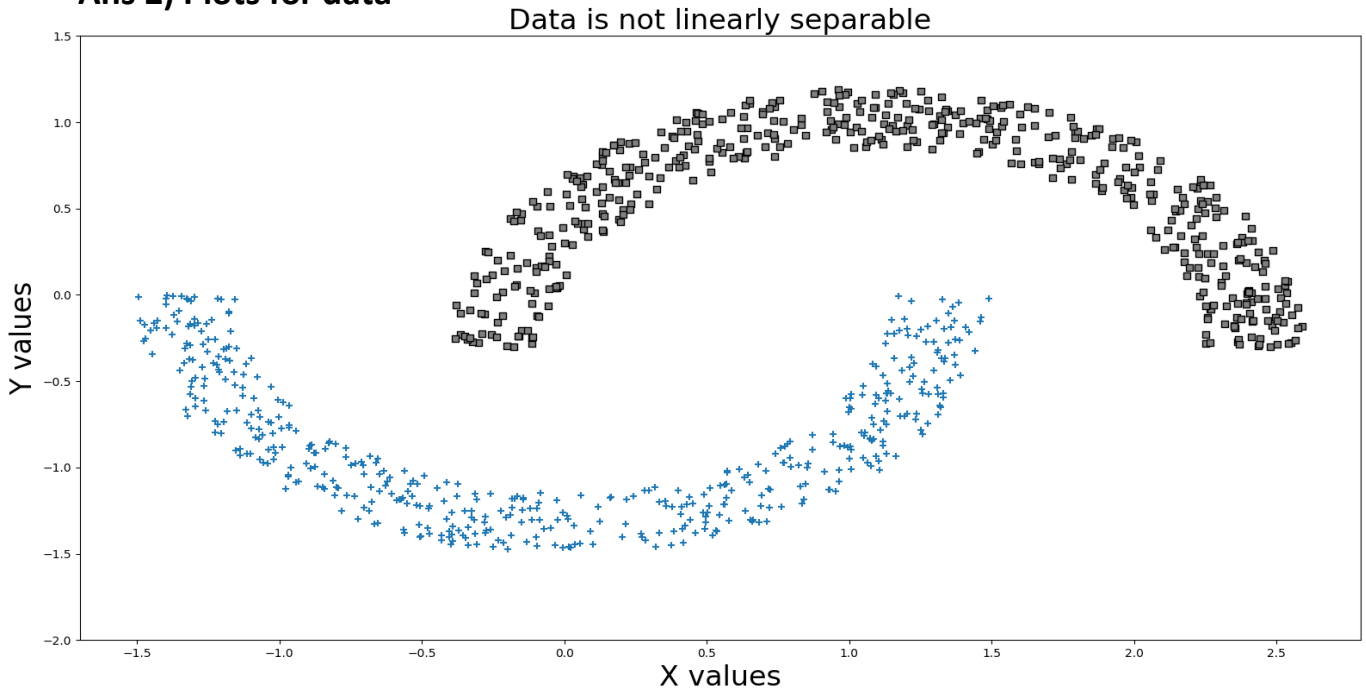
Code for 1(i) and 1(ii) is shown –

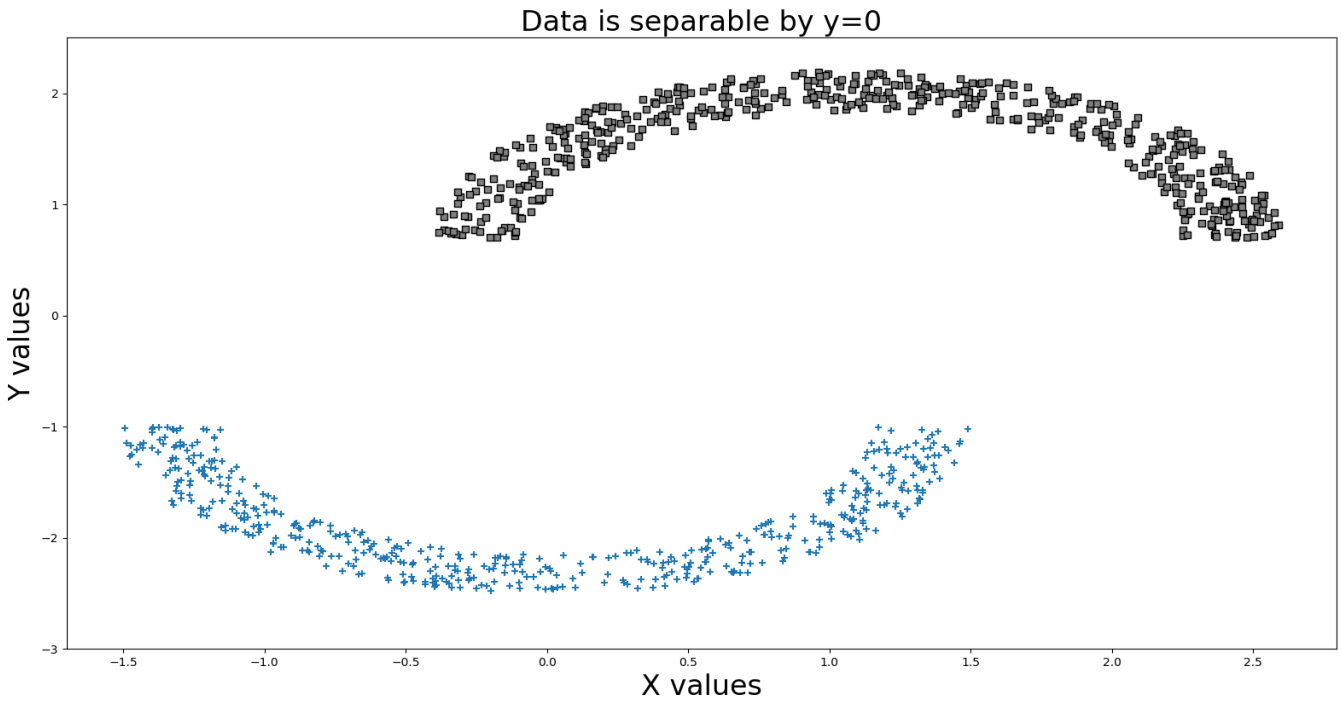
```
# -*- coding: utf-8 -*-
"""
Created on Thu Apr  8 20:19:11 2021

@author: verma
"""

# Q1
import matplotlib.pyplot as plt
def correlation(a,b):
    prod_xy=0
    sum_x2=0
    n=len(a)
    sum_y2=0
    sum_x=0
    sum_y=0
    for x,y in zip(a,b):
        prod_xy+=(x*y)
        sum_x2+=(x**2)
        sum_y2+=(y**2)
        sum_x+=x
        sum_y+=y
    p=((n*prod_xy)-(sum_x*sum_y))/((((n*sum_x2)-(sum_x)**2)**0.5)*((n*sum_y2-(sum_y)**2)**0.5))
    return p


#Q1 (A)
x1=[3,-1,6,12,8,10,9,13,15,-3,0,-5,-2,16]
x2=[3,-1,6,12,8,10,9,-5,-2,4,2,0,14,11]
x3=[3,4,2,5,8,-8,-6,-1,0,-5,4,2,6,8]
y1=[4,-2,5,11,6,11,7,8,9,-4,1,-6,-1,14]
y2=[0,2,-3,4,0,1,2,-1,-2,2,1,-2,9,2]
y3=[5,-1,0,1,2,2,4,2,1,2,1,2,-4,4]
plt.scatter(x1,y1,c="pink",linewidths=2,marker="s",edgecolor="green",s=50)
plt.title("Plot for (a)",fontsize=24)
plt.xlabel("X values",fontsize=24)
plt.ylabel("Y values",fontsize=24)
plt.show()
plt.figure()
plt.xlabel("X values",fontsize=24)
plt.ylabel("Y values",fontsize=24)
```

```
y2=[0,2,-3,4,0,1,2,-1,-2,2,1,-2,9,2]
y3=[5,-1,0,1,2,2,4,2,1,2,1,2,-4,4]
plt.scatter(x1,y1,c="pink",linewidths=2,marker="s",edgecolor="green",s=50)
plt.title("Plot for (a)",fontsize=24)
plt.xlabel("X values",fontsize=24)
plt.ylabel("Y values",fontsize=24)
plt.show()
plt.figure()
plt.xlabel("X values",fontsize=24)
plt.ylabel("Y values",fontsize=24)
plt.title("Plot for (b)",fontsize=24)
plt.scatter(x2,y2,c="yellow",linewidths=2,marker="^",edgecolor="red",s=100)
plt.show()
plt.figure()
plt.scatter(x3,y3,c="blue",linewidths=2,marker="D",edgecolor="yellow",s=50,label="plot for (c)")
plt.xlabel("X values",fontsize=24)
plt.title("Plot for (c)",fontsize=24)
plt.ylabel("Y values",fontsize=24)
plt.show()

#Q1(B) Correlation values

print("Correlation for (a)",correlation(x1, y1))
print("Correlation for (b)",correlation(x2,y2))
print("Correlation for (c)",correlation(x3,y3))
```

Q2) Consider samples belonging to the data of two categories (two different classes) as given below (Each sample is 2D and each class is having 500 samples). Plot this 2D data as a scatter plot, representing the data for each category in a different colour. See if the data for two categories is linearly separated (two classes are linearly separated if a single line can separate those two classes). If not, then perform the following operation on the given samples. Calculate the mean of the individual classes. If the mean of a class lies in the first quadrant (i.e., x and y of the mean are positive) then add +k in y - dimension of each sample. If the mean of a class lies in the third quadrant, then add -k in the y-dimension of each sample. Decide k yourself. Again, plot this 2-class data and comment about the separability of the 2 classes.

Ans 2) Plots for data-





Code for Q2 is shown-

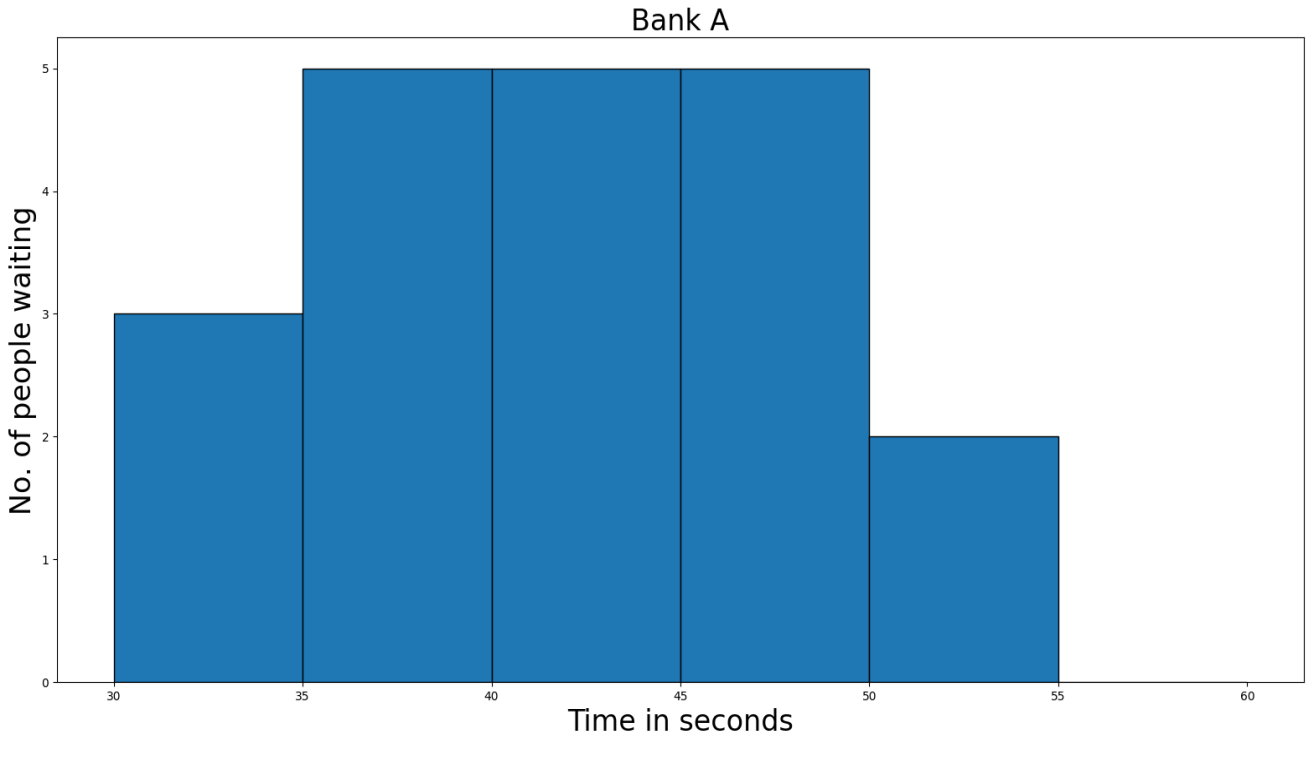
```python
# -*- coding: utf-8 -*-
"""
Created on Thu Apr  8 22:06:33 2021

@author: verma
"""
import pandas as pd
import matplotlib.pyplot as plt


# Q2
# Reading Data
foo=pd.read_csv("Class1.txt",sep=" ",header=None)
foo.columns=["x","y"]
class2=pd.read_csv("Class2.txt",sep=" ",header=None)
class2.columns=["x1","y1"]
x1=list(foo.x)
y1=list(foo.y)
x2=list(class2.x1)
y2=list(class2.y1)
plt.ylim(-2.0,1.5)
plt.scatter(x1,y1,marker="s",c="grey",edgecolor='black')
plt.title("Data is not linearly separable",fontsize=24)
plt.xlabel("X values",fontsize=24)
plt.ylabel("Y values",fontsize=24)
plt.scatter(x2,y2,marker="+")
mean_x1=sum(x1)/len(x1)
mean_y1=sum(y1)/len(y1)
mean_x2=sum(x2)/len(x2)
mean_y2=sum(y2)/len(y2)
k=1
if mean_x1>0 and mean_y1>0 :
    for i in range(0,500):
        y1[i]+=k
elif mean_x1<0 and mean_y1<0 :
     for j in range(0,500):
        y1[j]-=k
if mean_x2>0 and mean_y2>0 :
    for l in range(0,500):
     y2[l]+=k
elif mean_x2<0 and mean_y2<0 :
```

```
mean_y1=sum(y1)/len(y1)
mean_x2=sum(x2)/len(x2)
mean_y2=sum(y2)/len(y2)
k=1
if mean_x1>0 and mean_y1>0 :
    for i in range(0,500):
        y1[i]+=k
elif mean_x1<0 and mean_y1<0 :
     for j in range(0,500):
        y1[j]-=k
if mean_x2>0 and mean_y2>0 :
    for l in range(0,500):
     y2[l]+=k
elif mean_x2<0 and mean_y2<0 :
    for m in range(0,500):
     y2[m]-=k
plt.figure()
plt.xlabel("X values",fontsize=24)
plt.ylabel("Y values",fontsize=24)
plt.ylim(-3.0,2.5)
plt.title("Data is separable by y=0",fontsize=24)
plt.scatter(x1,y1,marker="s",c="grey",edgecolor='black')
plt.scatter(x2,y2,marker="+")
```

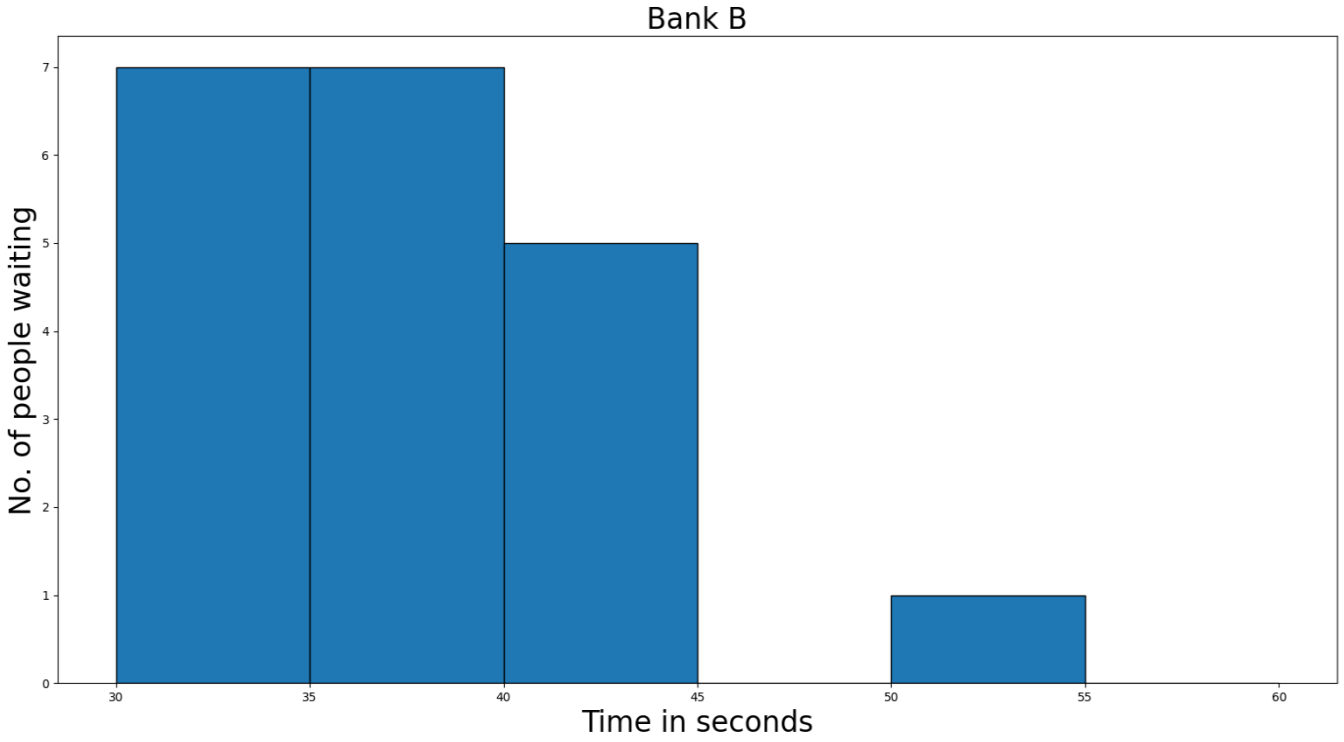
We have taken k=1 to make data separable.

Q3) A survey has been conducted by a private firm in between customer care services of two banks. The bank in which the waiting time for a client to be served by a customer service representative is less will be declared as a better bank. The data for 20 customers for the waiting time (in seconds) in bank A & bank B is given. Plot the histogram of the above data with 5-second bins (5-second intervals). Analyse the histogram and argue which bank is giving better services to its customer.

Ans 3) Histogram for Bank A is shown -



Histogram for Bank B-



From both the histogram we can infer that it is the Bank B, which is better than Bank A, since the waiting time of majority of the people is less in comparison to that of Bank A. Hence, Bank B is providing better services to its customer.

Code for Histograms-

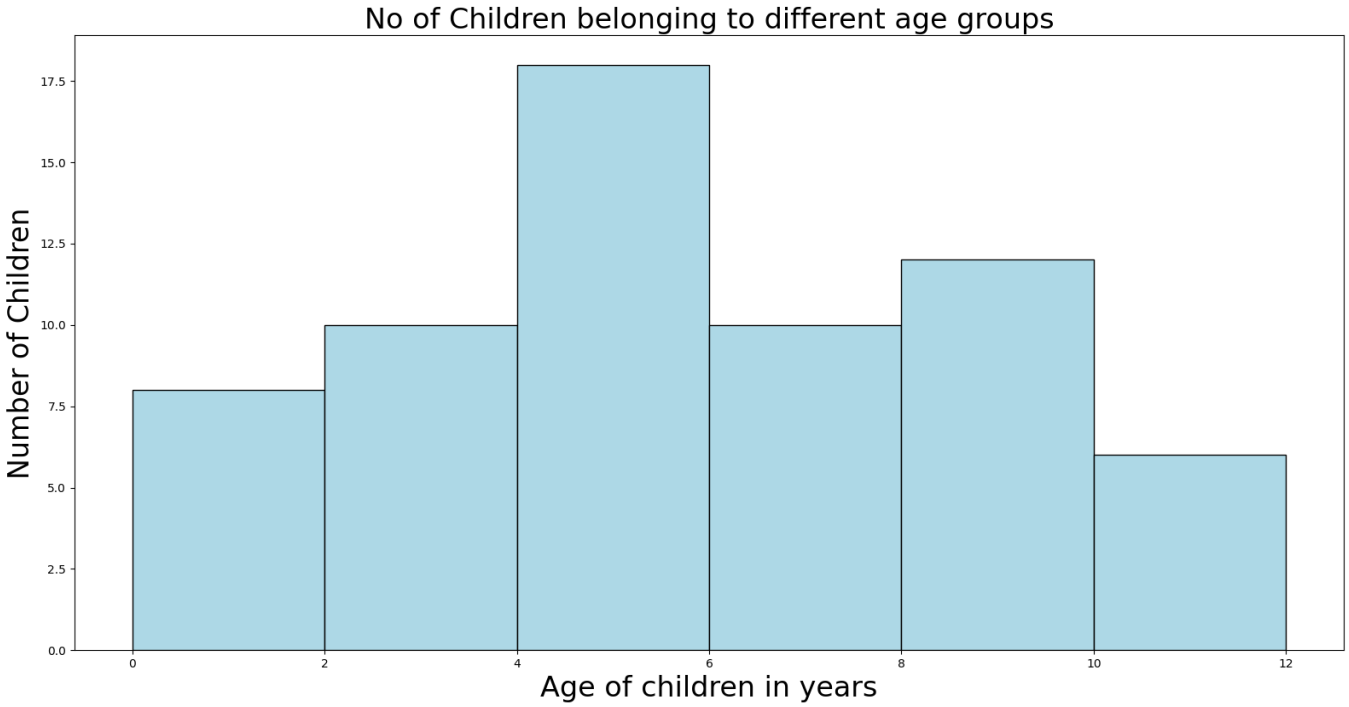
```
# -*- coding: utf-8 -*-
"""
Created on Fri Apr  9 09:42:54 2021

@author: verma
"""
# Q3
import matplotlib.pyplot as plt

Bank_A=[43.1,45.3,47.3,30.3,54.1,
        35.6,43.5,31.2,31.4,45.6,
        37.6,40.3,42.2,35.6,36.5,
        36.5,50.2,45.5,45.2,43.1]
Bank_B=[33.1,35.3,37.3,32.3,44.1,
        31.6,33.5,35.2,30.4,35.6,
        32.6,42.3,40.2,36.6,38.5,
        36.5,30.2,42.5,50.2,40.1]
plt.figure()
plt.hist(Bank_A,bins=[30,35,40,45,50,55,60],edgecolor="black")
plt.title("Bank A",fontsize=24)
plt.xlabel("Time in seconds",fontsize=24)
plt.ylabel("No. of people waiting",fontsize=24)
plt.show()
plt.figure()
plt.title("Bank B",fontsize=24)
plt.xlabel("Time in seconds",fontsize=24)
plt.ylabel("No. of people waiting",fontsize=24)
plt.hist(Bank_B,bins=[30,35,40,45,50,55,60],edgecolor="black")
plt.show()
```

Q4) The information about the number of children belonging to different age groups tabulated in the table given below.

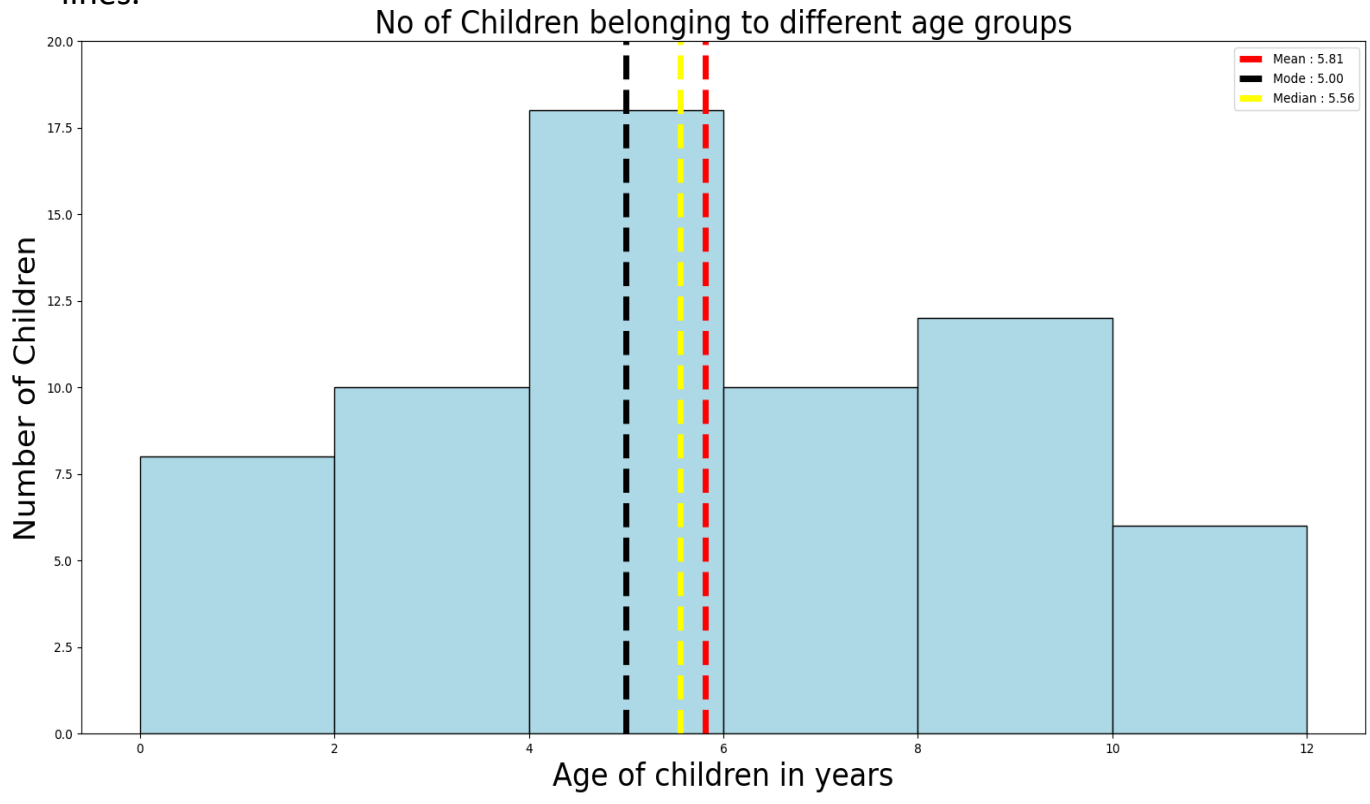
a) Plot this table as a graph, which is essentially the histogram representing the data.

b) Mark the points of mean, mode and median on the histogram, by vertical lines.

Ans 4) a) Histogram for table data –



b) Histogram marked with mean , mode & median on the histogram by vertical lines.



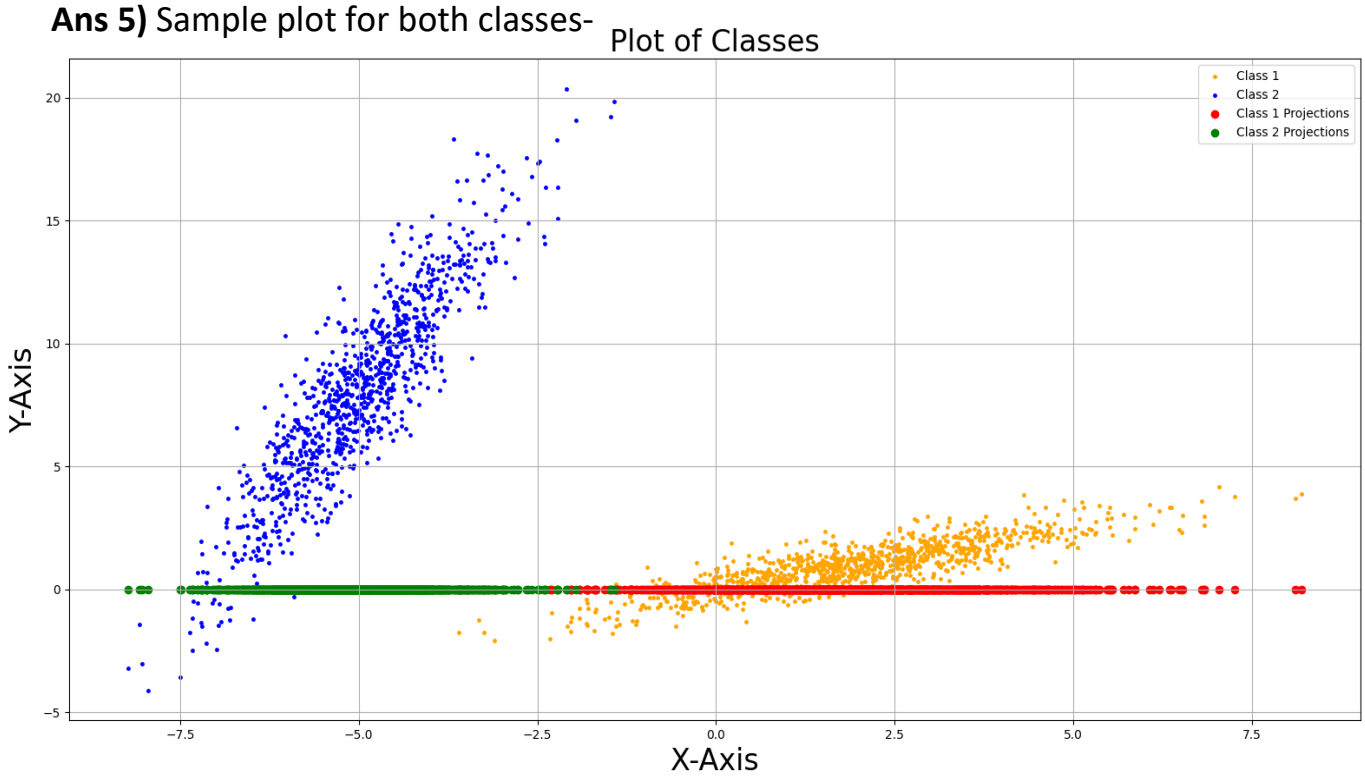
Code for Histograms-

```python
# -*- coding: utf-8 -*-
"""
Created on Fri Apr  9 10:07:26 2021

@author: verma
"""
import matplotlib.pyplot as plt

def mean(x,y):
    k=0
    for i,j in zip(x,y):
            k+=i*j
    mn=k/(sum(y)*1.0)
    return mn
def mode(x,y):
    u=max(y)
    s=y.index(u)
    c=x[s]-1
    f= c+ ((y[s]-y[s-1])/(2*y[s]-y[s-1]-y[s+1]))*2
    return f
def median(x,y):
    n=sum(y)
    k=int(n/2)
    a=0
    for i in range(0,len(y)):
        a+=y[i]
        v=i
        if a>k:break
    med=(x[v]-1)+ (((n/2)-(a-y[v]))*2/y[v])
    return med

# Q4(a)
Age=[1,3,5,7,9,11]
no_child=[8,10,18,10,12,6]
plt.bar(Age,no_child,width=2,edgecolor="black",color='lightblue',align='center')
plt.xlabel("Age of children in years",fontsize=24)
plt.ylabel("Number of Children",fontsize=24)
plt.title("No of Children belonging to different age groups",fontsize=24)
plt.show()
```

```python
        v=i
        if a>k:break
    med=(x[v]-1)+ (((n/2)-(a-y[v]))*2/y[v])
    return med

# Q4(a)
Age=[1,3,5,7,9,11]
no_child=[8,10,18,10,12,6]
plt.bar(Age,no_child,width=2,edgecolor="black",color='lightblue',align='center')
plt.xlabel("Age of children in years",fontsize=24)
plt.ylabel("Number of Children",fontsize=24)
plt.title("No of Children belonging to different age groups",fontsize=24)
plt.show()

# Q4 (b)
plt.axvline(mean(Age,no_child),color='red',linestyle='dashed',linewidth='5',label="Mean : 5.81")
plt.axvline(mode(Age,no_child),color='black',linestyle='dashed',linewidth='5',label='Mode : 5.00')
plt.axvline(median(Age,no_child),color='yellow',linestyle='dashed',linewidth='5',label='Median : 5.56')
plt.legend()
plt.ylim(0,20)
plt.show()
```

Q5) Consider the given two class linearly separable data (i.e., the two classes can be separated with the help of a line). Project each point of this data on the line x = 0 (i.e., on x-axis). After projection, plot the samples and comment on the separability of the two classes.

Ans 5) Sample plot for both classes-



From the plot we can observe that the Class 1 projections and Class 2 projections are not linearly separable since they overlap.

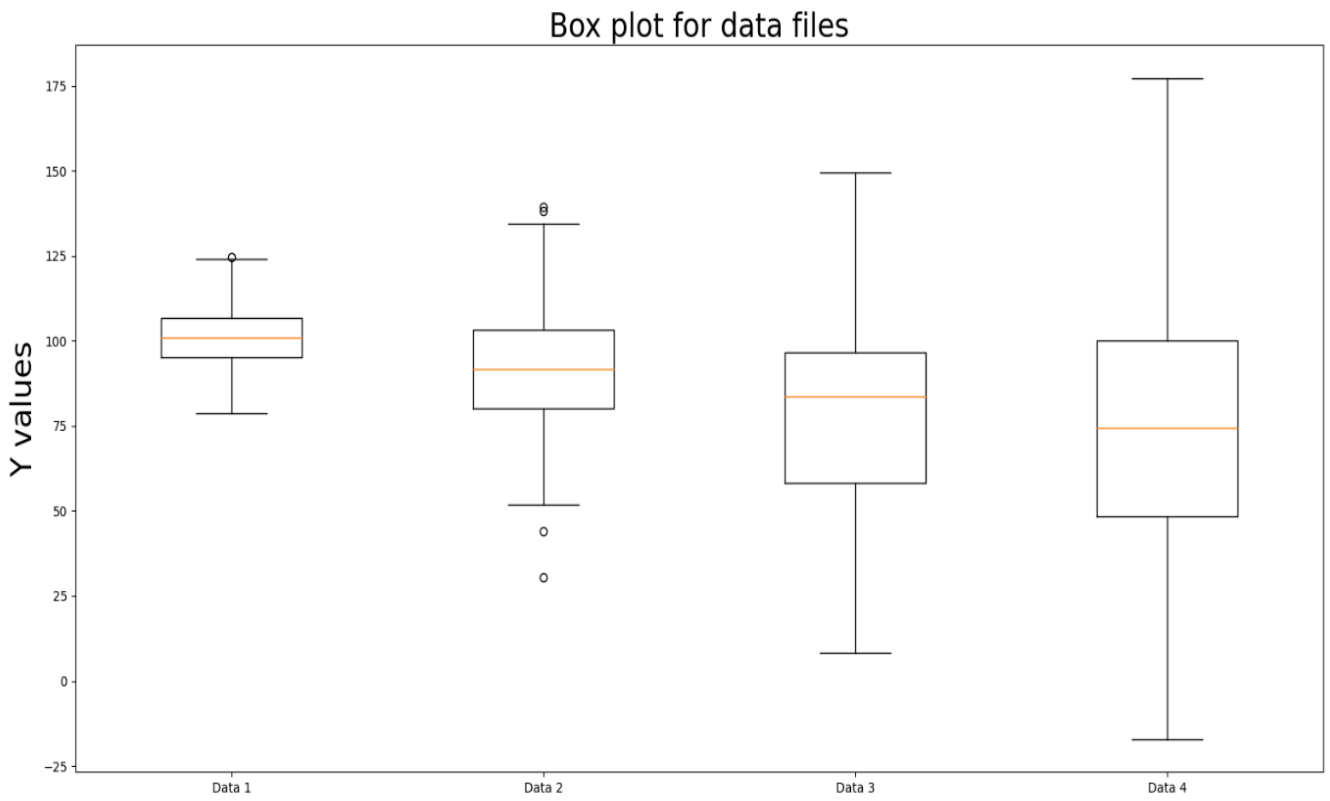
Code for plot-

```
@author: verma
"""

import matplotlib.pyplot as plt

# Q5
# Reading Data
class1=open("C:/Users/verma/OneDrive/Desktop/LAB 6/linearly_seperable_data/Class1.txt",'r+')
class2=open("C:/Users/verma/OneDrive/Desktop/LAB 6/linearly_seperable_data/Class2.txt",'r+')
x1=[]
y1=[]
for line in class1.readlines():
    l,k=line.split()
    x1.append(float(l))
    y1.append(float(k))

x2=[]
y2=[]
for line in class2.readlines():
    j,k=line.split()
    x2.append(float(j))
    y2.append(float(k))
```

```
for line in class2.readlines():
    j,k=line.split()
    x2.append(float(j))
    y2.append(float(k))

plt.scatter(x1,y1,color="orange",marker='.',label="Class 1",s=25)
plt.scatter(x2,y2,color='b',marker='.',label="Class 2",s=25)
plt.xlabel("X-Axis",fontsize=24)
plt.ylabel("Y-Axis",fontsize=24)
plt.title("Plot of Classes",fontsize=24)
plt.grid()
plt.show()

#Plotting projections

for i in range(0,len(x1)):
    y1[i]=0
    y2[i]=0
plt.scatter(x1,y1,color='red',label="Class 1 Projections")
plt.scatter(x2,y2,color='green',label="Class 2 Projections")
plt.legend()
plt.show()
```

Q6) Given data (data.csv) is a single column data. Draw the box plot of this single column data to find out the range in which maximum data is lying (Interquartile range).

Ans) Box Plot for the data is shown below-



The interquartile range for the boxplot-

```
In [37]: runfile('C:/Users/verma/OneDrive/Desktop/LAB 6/q6_boxplot/untitled8.py', wdir='C:/Users/
verma/OneDrive/Desktop/LAB 6/q6_boxplot')
Reloaded modules: jupyter_client.session, zmq.eventloop, zmq.eventloop.ioloop, tornado.platform,
tornado.platform.asyncio, tornado.gen, zmq.eventloop.zmqstream, jupyter_client.jsonutil,
jupyter_client.adapter, spyder, spyder.pil_patch, PIL, PIL._version, PIL.Image, PIL.ImageMode,
PIL.TiffTags, PIL._binary, PIL._util, PIL._imaging, cffi, cffi.api, cffi.lock, cffi.error,
cffi.model, IPython.lib.guisupport, IPython.external, IPython.external.qt_for_kernel,
IPython.utils.version, IPython.external.qt_loaders
Interquartile range for data is  31.19

In [38]:
```

Code for Boxplot-

```
1   # -*- coding: utf-8 -*-
2   """
3   Created on Sat Apr 10 22:37:08 2021
4
5   @author: verma
6   """
7
8   import pandas as pd
9   import matplotlib.pyplot as plt
10
11  # Q6
12  foo=pd.read_csv("d1.txt",sep=" ",header=None)
13  foo1=pd.read_csv("d2.txt",sep=" ",header=None)
14  foo2=pd.read_csv("d3.txt",sep=" ",header=None)
15  foo3=pd.read_csv("d4.txt",sep=" ",header=None)
16  foo.columns=["x"]
17  foo1.columns=["x"]
18  foo2.columns=["x"]
19  foo3.columns=["x"]
20
21  x1=list(foo.x)
22  x2=list(foo1.x)
23  x3=list(foo2.x)
24  x4=list(foo3.x)
25  data=[x1,x2,x3,x4]
26  plt.title("Box plot for data files",fontsize=24)
27  plt.ylabel("Values",fontsize=24)
28  plt.boxplot(data,labels=["Data 1","Data 2","Data 3","Data 4"])
29  plt.show()
```

Code for Interquartile range-

```
29  plt.boxplot(data,labels=["Data 1","Data 2","Data 3","Data 4"])
30  plt.show()
31  # for Inter quartile range
32  Q1 = np.percentile(data, 25, interpolation = 'midpoint')
33  Q2= np.percentile(data, 50, interpolation = 'midpoint')
34  Q3 = np.percentile(data, 75, interpolation = 'midpoint')
35  Interquartile= Q3-Q1
36  print("Interquartile range for data is ",round(Interquartile,2))
37
```