CS 584-04: Machine Learning

Fall 2019 Assignment 1

Question 1 (40 points)

Write a Python program to calculate the density estimator of a histogram. Use the field *x* in the NormalSample.csv file.

a) (5 points) According to Izenman (1991) method, what is the recommended bin-width for the histogram of x?

h = 2*(IQR)*N^(-1/3)

h = bin-width

IQR = interquartile range

N = total number of observations

b) (5 points) What are the minimum and the maximum values of the field x?

Minimum value of x is 26.300000 and maximum value of x is 35.400000.

c) (5 points) Let a be the largest integer less than the minimum value of the field x, and b be the smallest integer greater than the maximum value of the field x. What are the values of a and b?

a = 26

b = 36

d) (5 points) Use h = 0.1, minimum = a and maximum = b. List the coordinates of the density estimator. Paste the histogram drawn using Python or your favorite graphing tools.

Coordinates of the density estimator are as follow,

(26.05 , 0.0)

(26.150000000000002 , 0.0)

(26.250000000000004 , 0.00999000999000999)

(26.350000000000005 , 0.0)

(26.450000000000006 , 0.0)

(26.550000000000008 , 0.0)

(26.65000000000001 , 0.0)

(26.75000000000001 , 0.0)

(26.850000000000012 , 0.0)

(26.950000000000014 , 0.0)

(27.050000000000015 , 0.0)

(27.150000000000016 , 0.00999000999000999)

(27.250000000000018 , 0.0)

(27.35000000000002 , 0.0)

(27.45000000000002 , 0.0)

(27.550000000000022 , 0.0)

(27.650000000000023 , 0.01998001998001998)

(27.750000000000025 , 0.0)

(27.850000000000026 , 0.02997002997002997)

(27.950000000000028 , 0.00999000999000999)

(28.05000000000003 , 0.00999000999000999)

(28.15000000000003 , 0.049950049950049945)

(28.250000000000032 , 0.02997002997002997)

(28.350000000000033 , 0.01998001998001998)

(28.450000000000035 , 0.03996003996003996)

(28.550000000000036 , 0.03996003996003996)

(28.650000000000038 , 0.049950049950049945)

(28.75000000000004 , 0.07992007992007992)

(28.85000000000004 , 0.049950049950049945)

(28.950000000000042 , 0.049950049950049945)

(29.050000000000043 , 0.03996003996003996)

(29.150000000000045 , 0.10989010989010987)

(29.250000000000046 , 0.14985014985014983)

(29.350000000000048 , 0.07992007992007992)

(29.45000000000005 , 0.13986013986013984)

(29.55000000000005 , 0.1898101898101898)

(29.650000000000052 , 0.0899100899100899)

(29.750000000000053 , 0.09990009990009989)

(29.850000000000055 , 0.20979020979020976)

(29.950000000000056 , 0.15984015984015984)

(30.050000000000058 , 0.14985014985014983)

(30.15000000000006 , 0.21978021978021975)

(30.25000000000006 , 0.14985014985014983)

(30.350000000000062 , 0.2797202797202797)

(30.450000000000063 , 0.23976023976023975)

(30.550000000000065 , 0.1898101898101898)

(30.650000000000066 , 0.2697302697302697)

(30.750000000000068 , 0.19980019980019978)

(30.85000000000007 , 0.19980019980019978)

(30.95000000000007 , 0.16983016983016982)

(31.05000000000007 , 0.15984015984015984)

(31.150000000000073 , 0.2797202797202797)

(31.250000000000075 , 0.20979020979020976)

(31.350000000000076 , 0.2797202797202797)

(31.450000000000077 , 0.33966033966033965)

(31.55000000000008 , 0.2597402597402597)

(31.65000000000008 , 0.33966033966033965)

(31.75000000000008 , 0.2697302697302697)

(31.850000000000083 , 0.19980019980019978)

(31.950000000000085 , 0.33966033966033965)

(32.05000000000008 , 0.24975024975024973)

(32.150000000000084 , 0.3196803196803197)

(32.250000000000085 , 0.1898101898101898)

(32.35000000000009 , 0.23976023976023975)

(32.45000000000009 , 0.2797202797202797)

(32.55000000000009 , 0.22977022977022976)

(32.65000000000009 , 0.29970029970029965)

(32.75000000000009 , 0.21978021978021975)

(32.850000000000094 , 0.15984015984015984)

(32.950000000000095 , 0.1898101898101898)

(33.0500000000001 , 0.13986013986013984)

(33.1500000000001 , 0.12987012987012986)

(33.2500000000001 , 0.15984015984015984)

(33.3500000000001 , 0.05994005994005994)

(33.4500000000001 , 0.10989010989010987)

(33.550000000000104 , 0.05994005994005994)

(33.650000000000105 , 0.06993006993006992)

(33.75000000000011 , 0.05994005994005994)

(33.85000000000011 , 0.06993006993006992)

(33.95000000000011 , 0.02997002997002997)

(34.05000000000011 , 0.03996003996003996)

(34.15000000000011 , 0.049950049950049945)

(34.250000000000114 , 0.02997002997002997)

(34.350000000000115 , 0.00999000999000999)

(34.45000000000012 , 0.01998001998001998)

(34.55000000000012 , 0.01998001998001998)

(34.65000000000012 , 0.00999000999000999)

(34.75000000000012 , 0.00999000999000999)

(34.85000000000012 , 0.00999000999000999)

(34.950000000000124 , 0.0)

(35.050000000000125 , 0.0)

(35.15000000000013 , 0.0)

(35.25000000000013 , 0.00999000999000999)

(35.35000000000013 , 0.00999000999000999)

(35.45000000000013 , 0.0)

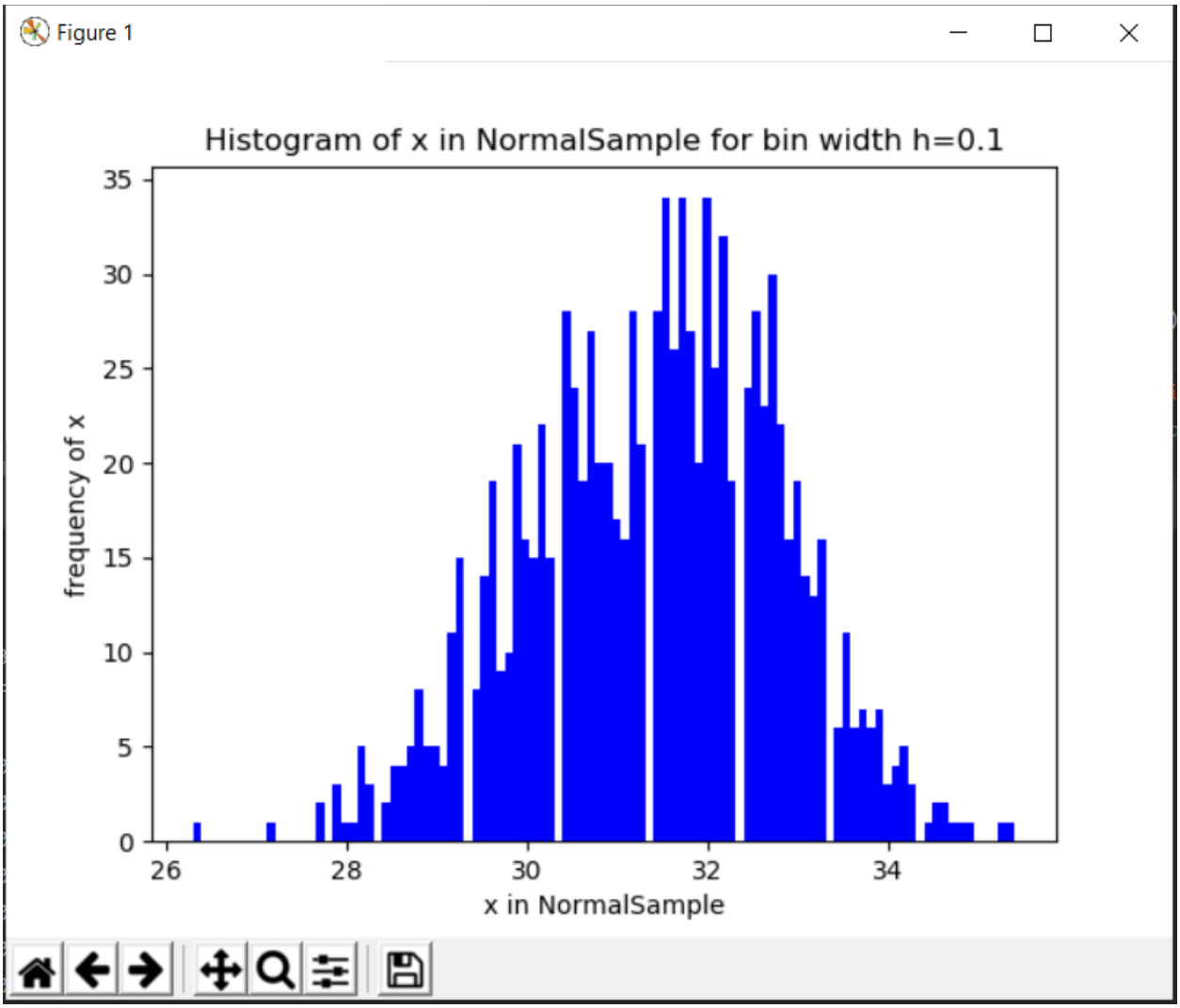
(35.55000000000013 , 0.0)

(35.650000000000134 , 0.0)

(35.750000000000135 , 0.0)

(35.850000000000136 , 0.0)

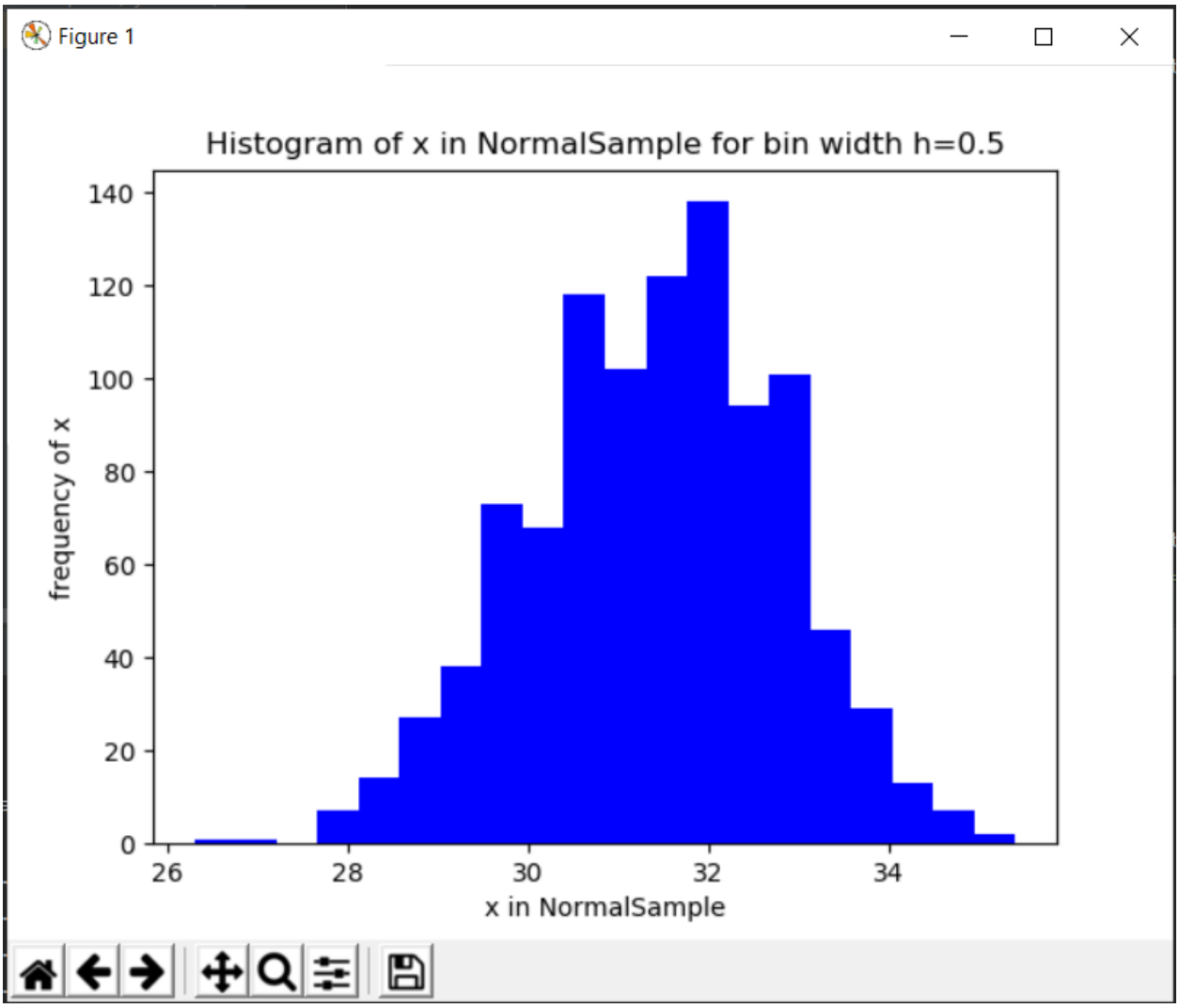
(35.95000000000014 , 0.0)



e) (5 points) Use h = 0.5, minimum = a and maximum = b. List the coordinates of the density estimator. Paste the histogram drawn using Python or your favorite graphing tools.

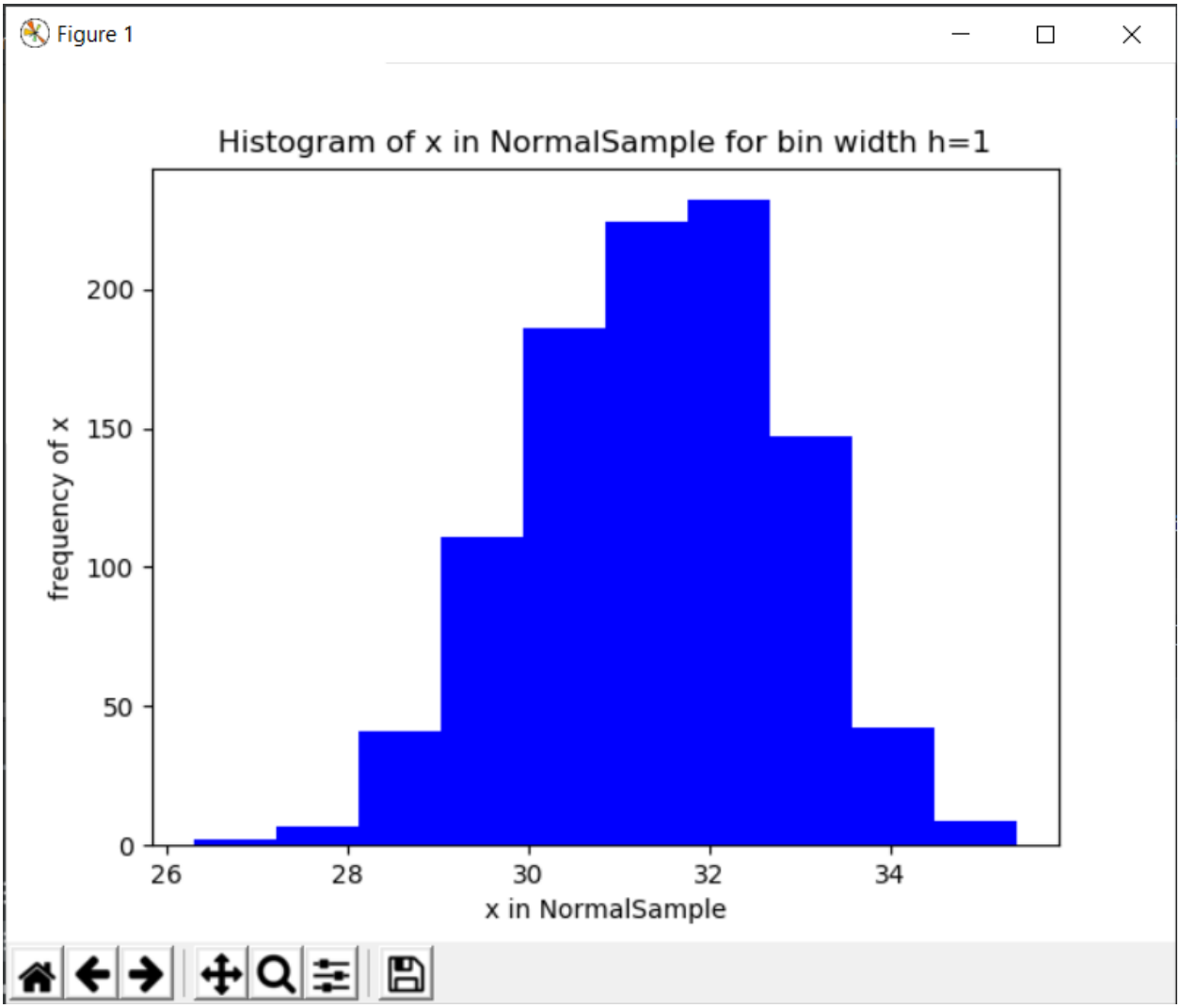
Coordinates of the density estimator are as follow,
(26.25 , 0.001998001998001998)
(26.75 , 0.0)

(27.25 , 0.001998001998001998)
(27.75 , 0.011988011988011988)
(28.25 , 0.029970029970029972)
(28.75 , 0.053946053946053944)
(29.25 , 0.1038961038961039)
(29.75 , 0.14985014985014986)
(30.25 , 0.2077922077922078)
(30.75 , 0.2057942057942058)
(31.25 , 0.25374625374625376)
(31.75 , 0.2817182817182817)
(32.25 , 0.25574425574425574)
(32.75 , 0.21978021978021978)
(33.25 , 0.11988011988011989)
(33.75 , 0.057942057942057944)
(34.25 , 0.029970029970029972)
(34.75 , 0.00999000999000999)
(35.25 , 0.003996003996003996)
(35.75 , 0.0)



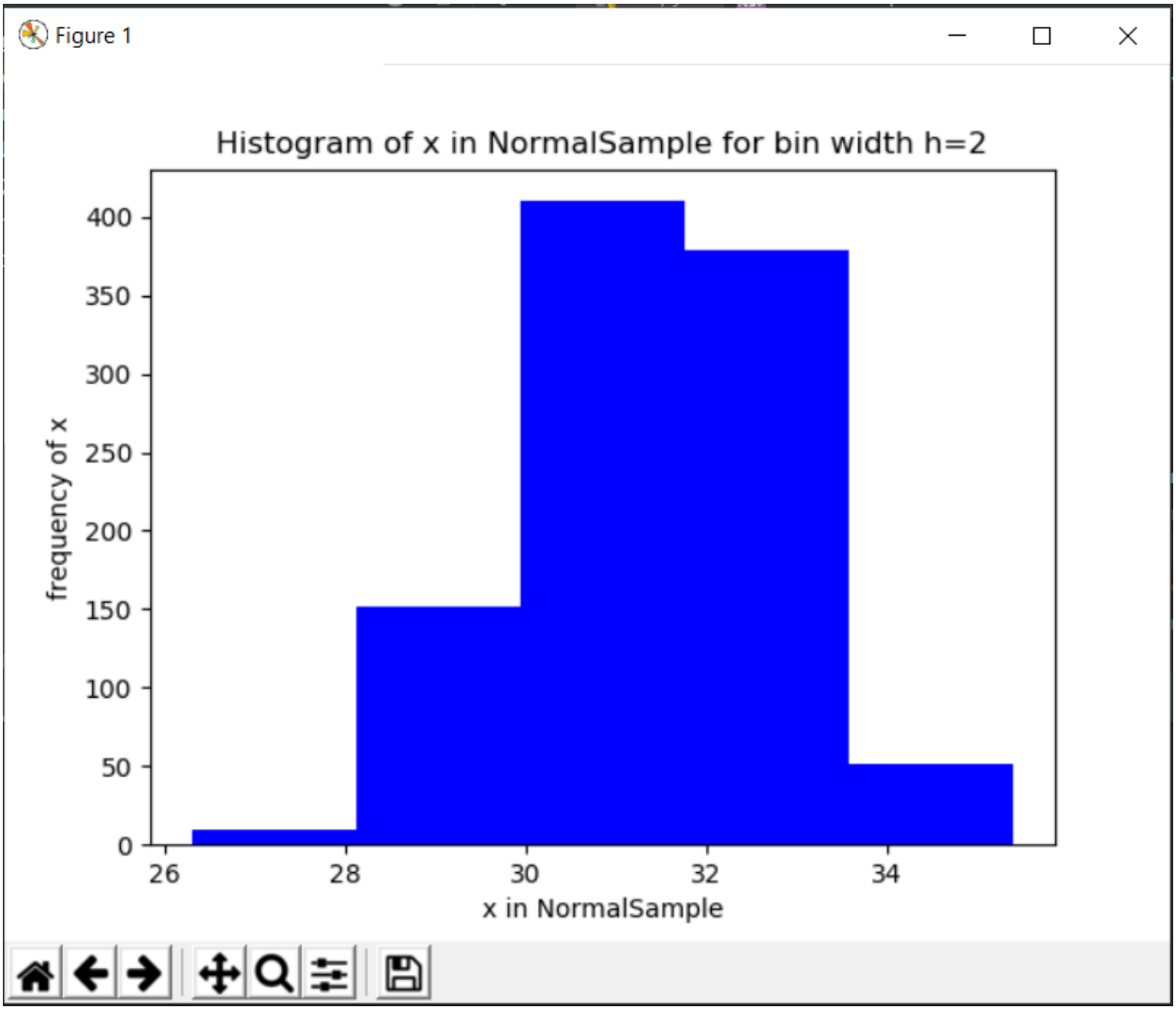
f) (5 points) Use h = 1, minimum = a and maximum = b. List the coordinates of the density estimator. Paste the histogram drawn using Python or your favorite graphing tools.

Coordinates of the density estimator are as follow,
(26.5 , 0.000999000999000999)
(27.5 , 0.006993006993006993)
(28.5 , 0.04195804195804196)
(29.5 , 0.12687312687312688)
(30.5 , 0.20679320679320679)
(31.5 , 0.2677322677322677)
(32.5 , 0.23776223776223776)
(33.5 , 0.08891108891108891)
(34.5 , 0.01998001998001998)
(35.5 , 0.001998001998001998)



g) (5 points) Use h = 2, minimum = a and maximum = b. List the coordinates of the density estimator. Paste the histogram drawn using Python or your favorite graphing tools.

Coordinates of the density estimator are as follow,
(27.0 , 0.003996003996003996)
(29.0 , 0.08441558441558442)
(31.0 , 0.23726273726273725)
(33.0 , 0.16333666333666333)
(35.0 , 0.01098901098901099)



h) (5 points) Among the four histograms, which one, in your honest opinions, can best provide your insights into the shape and the spread of the distribution of the field x? Please state your arguments.

According to izenman bin-width equation h=2*(IQR)*N^(-1/3)

Here N = 1001

IQR = Q3-Q1 = 32.4000-30.4000

Bin-width h = 0.3998 = 0.4

According to nice bin-width nice bin-width is 0.1.

According to shape and spread first histogram with bin-width h=0.1 provides better information. Some histograms have center shifted to right which shows that they don't show us a proper spread of data at both side of center like mirror image. But histogram with bin-width h=0.1 is showing some kind of mirror image(symmetric) at both end.

Question 2 (20 points)

Use in the NormalSample.csv to generate box-plots for answering the following questions.

a) **(5 points) What is the five-number summary of x? What are the values of the 1.5 IQR whiskers?**

Minimum: 26.300000

First quartile: 30.400000

Median: 31.500000

Third quartile: 32.400000

Maximum: 35.400000

Interquartile range IQR is (Q3-Q1) 2.000000

Lower whisker (max value of minimum and Q1-1.5IQR) is 27.400000

Upper whisker (min value of maximum and Q3+1.5IQR) is 35.400000

b) **(5 points) What is the five-number summary of x for each category of the group? What are the values of the 1.5 IQR whiskers for each category of the group?**

For category 0:

Minimum: 26.300000

First quartile: 29.400000

Median: 30.000000

Third quartile: 30.600000

Maximum: 32.200000

Interquartile range IQR is (Q3-Q1) 1.200000

Lower whiskers (max value of minimum and Q1-1.5IQR) is 27.600000

Upper whiskers (min value of maximum and Q3+1.5IQR) is 32.200000

For category 1:

Minimum: 29.100000

First quartile: 31.400000

Median: 32.100000

Third quartile: 32.700000

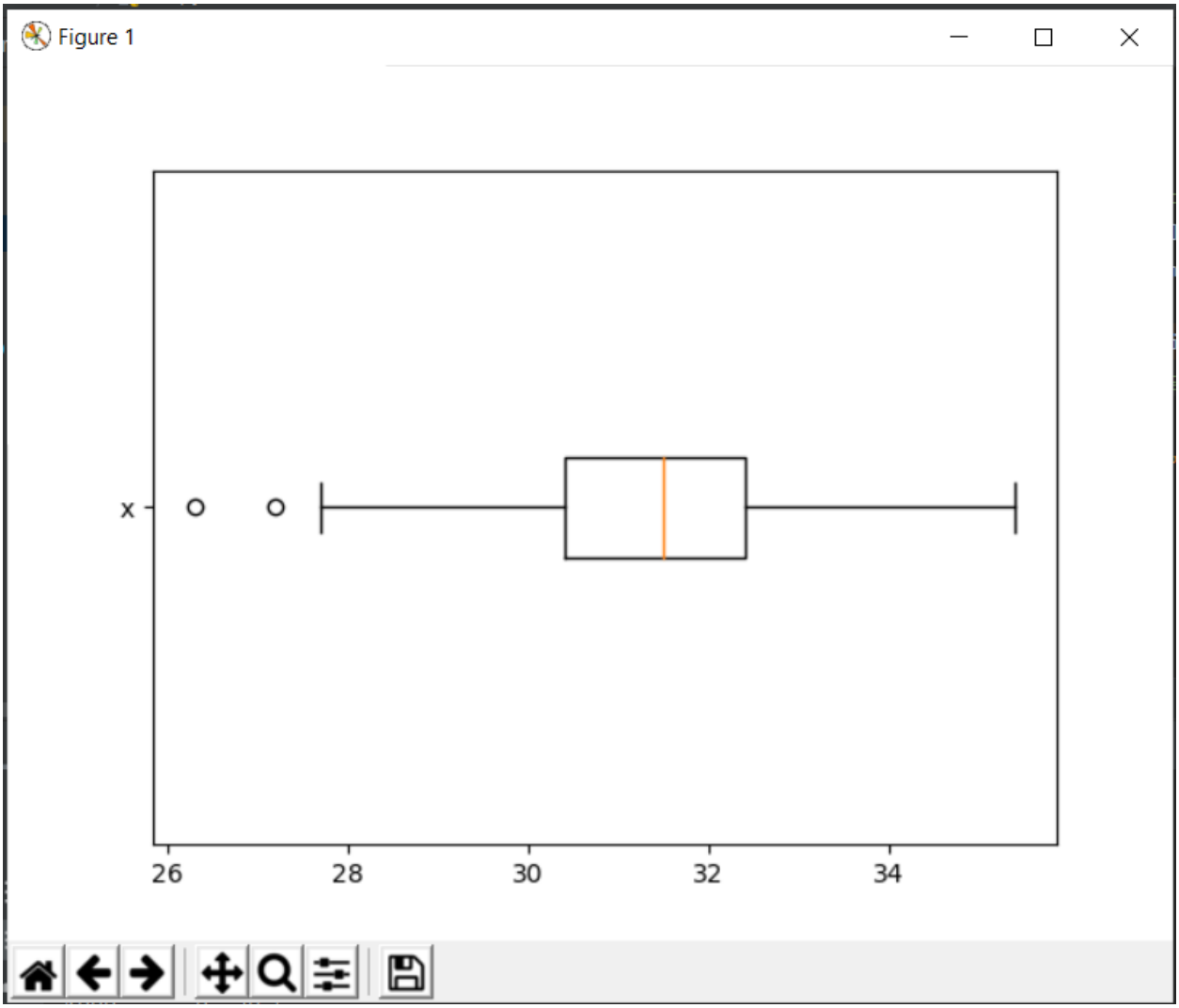
Maximum: 35.400000

Interquartile range IQR is (Q3-Q1) 1.300000

Lower whisker (max value of minimum and Q1-1.5IQR) is 29.450000

Upper whisker (min value of maximum and Q3+1.5IQR) is 34.650000

c) (5 points) Draw a boxplot of x (without the group) using the Python boxplot function. Can you tell if the Python's boxplot has displayed the 1.5 IQR whiskers correctly?



According to question2 section a,

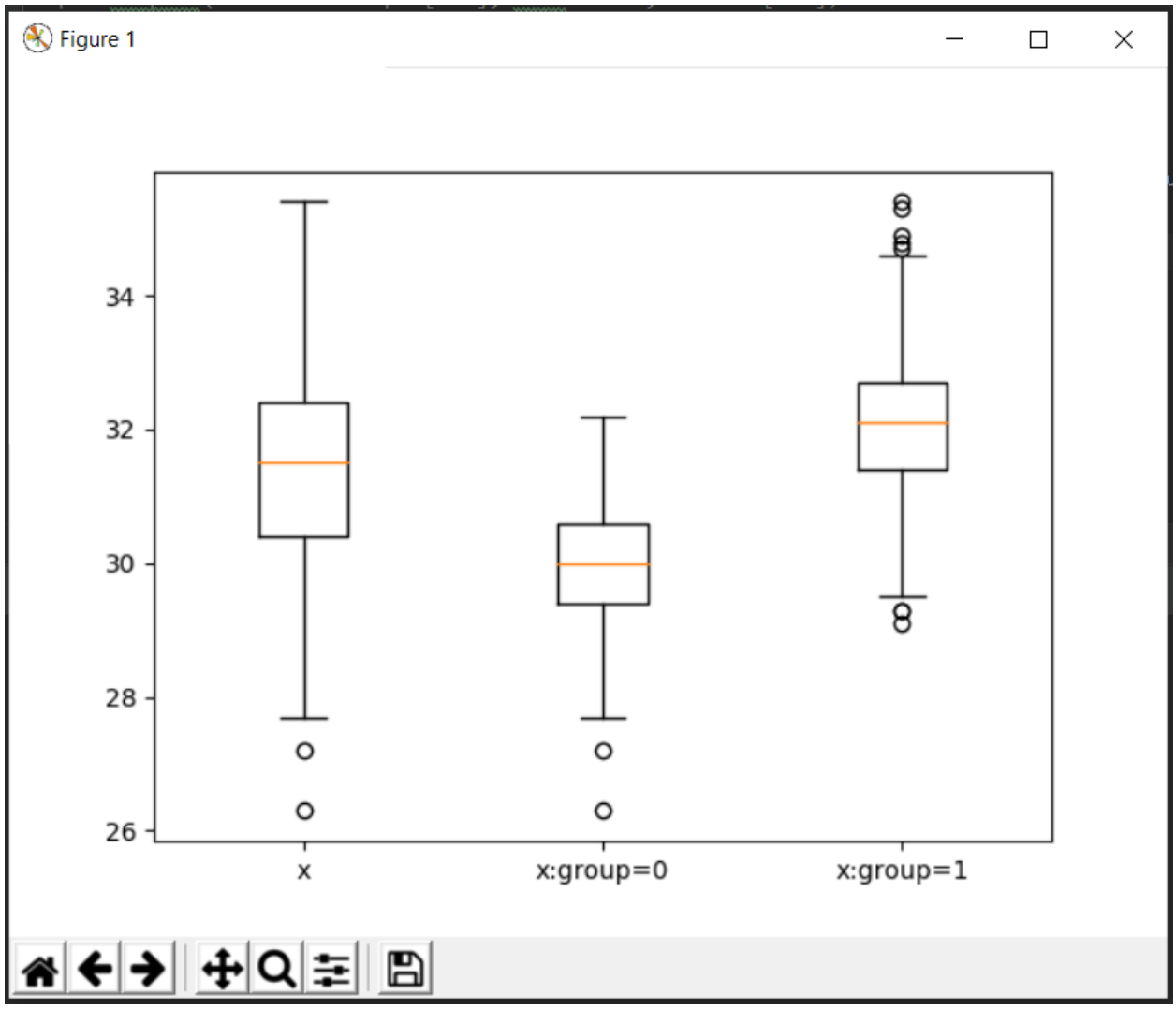
Lower whisker is 27.400000

Upper whisker is 35.400000

And in the graph, we can see that two dots are lower than lower whisker 27.400000.

So, according to me this python graph shows 1.5 IQR whiskers correctly.

d) (5 points) Draw a graph where it contains the boxplot of x, the boxplot of x for each category of Group (i.e., three boxplots within the same graph frame). Use the 1.5 IQR whiskers, identify the outliers of x, if any, for the entire data and for each category of the group.
***Hint: Consider using the CONCAT function in the PANDA module to append observations*.**



For the entire data there are outliers only below the lower whicker.
For category group=0 there are outliers only below the lower whicker.
For category group=1 there are outliers on both above the upper whisker and below the lower whicker.

Question 3 (40 points)

The data, FRAUD.csv, contains results of fraud investigations of 5,960 cases. The binary variable FRAUD indicates the result of a fraud investigation: 1 = Fraudulent, 0 = Otherwise. The other interval variables contain information about the cases.

1. **TOTAL_SPEND: Total amount of claims in dollars**
2. **DOCTOR_VISITS: Number of visits to a doctor**
3. **NUM_CLAIMS: Number of claims made recently**
4. **MEMBER_DURATION: Membership duration in number of months**
5. **OPTOM_PRESC: Number of optical examinations**
6. **NUM_MEMBERS: Number of members covered**

You are asked to use the Nearest Neighbors algorithm to predict the likelihood of fraud.

a) **(5 points) What percent of investigations are found to be fraudulent? Please give your answer up to 4 decimal places.**

Total Cases = 5960

Fraudulent cases count = 1189

Percent of investigations are found to be fraudulent = (1189*100)/5960 = 19.9497%

b) **(5 points) Use the BOXPLOT function to produce horizontal box-plots. For each interval variable, one box-plot for the fraudulent observations, and another box-plot for the non-fraudulent observations. These two box-plots must appear in the same graph for each interval variable.**

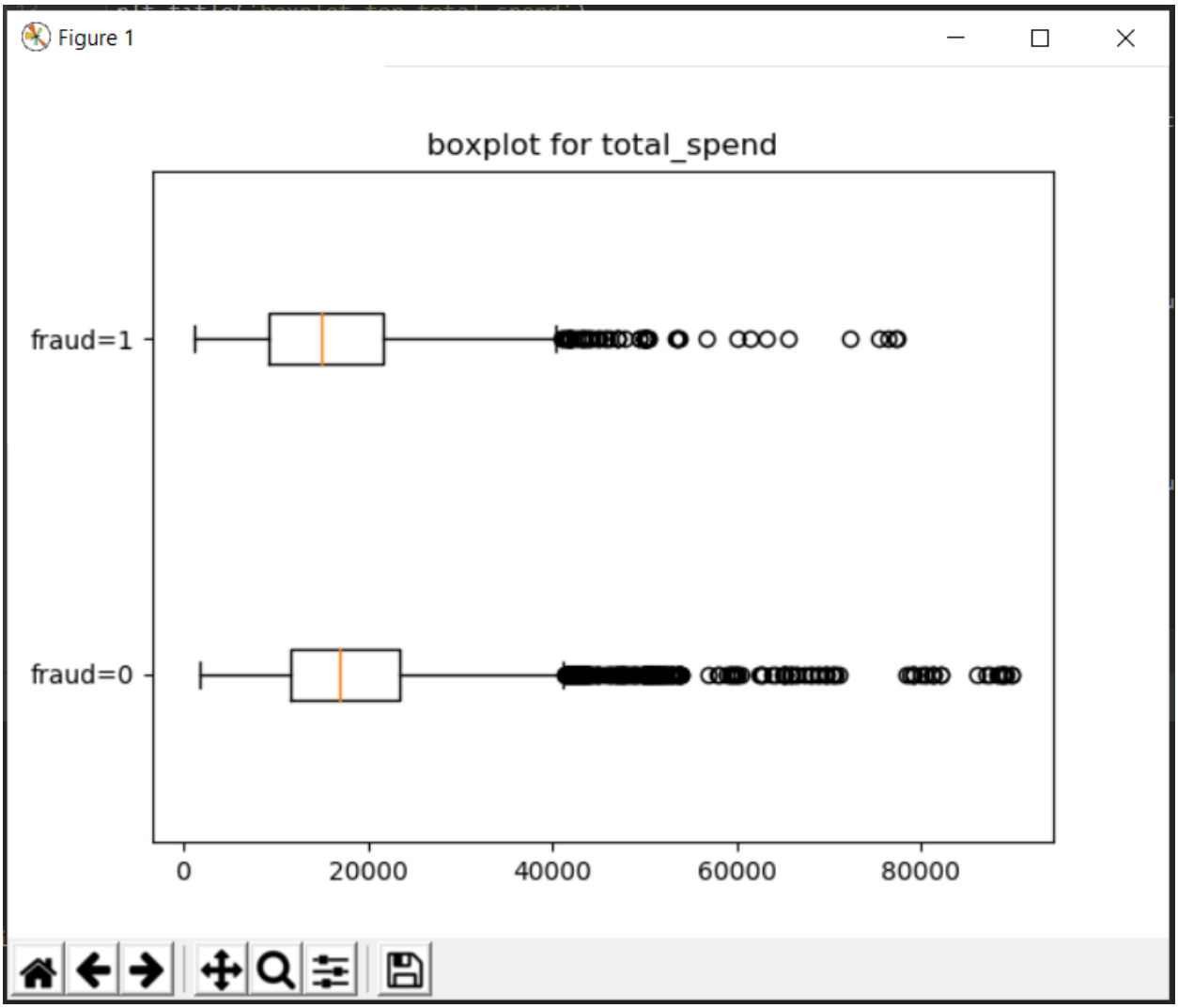


Figure 1
boxplot for total_spend
fraud=1
fraud=0
0
20000
40000
60000
80000

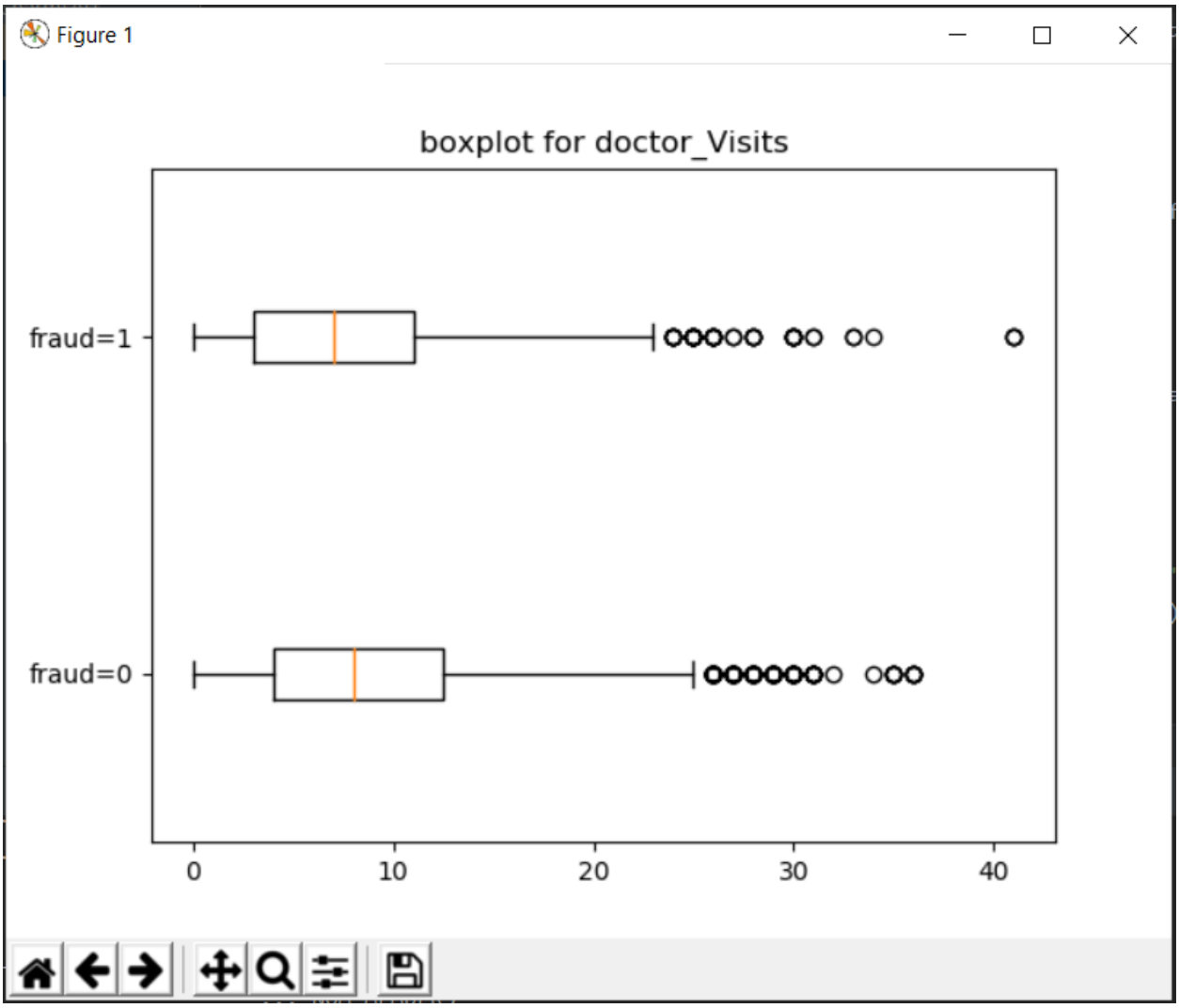


Figure 1
boxplot for doctor_Visits
fraud=1
fraud=0
0
10
20
30
40

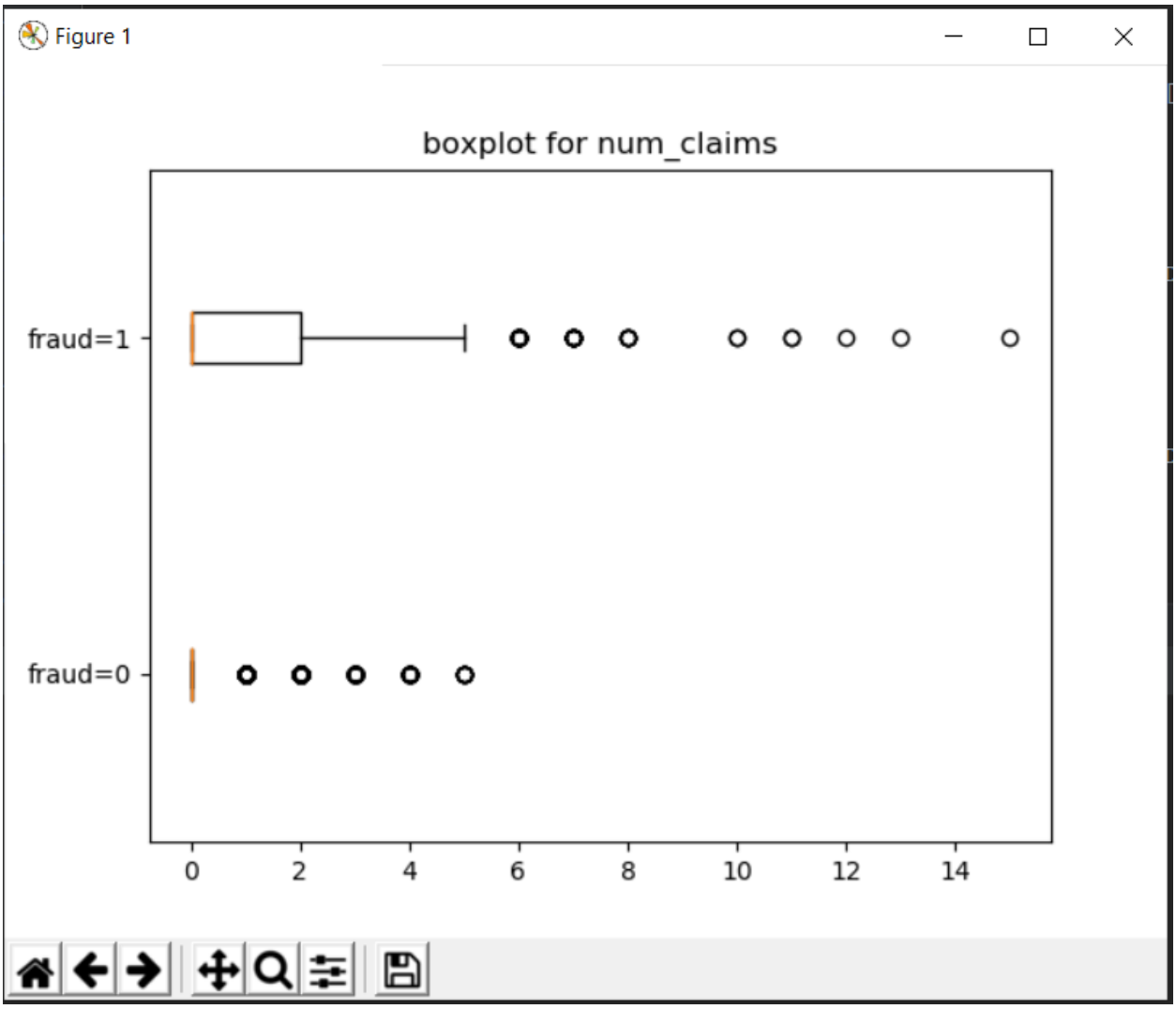


Figure 1
boxplot for num_claims
fraud=1
fraud=0
0
2
4
6
8
10
12
14

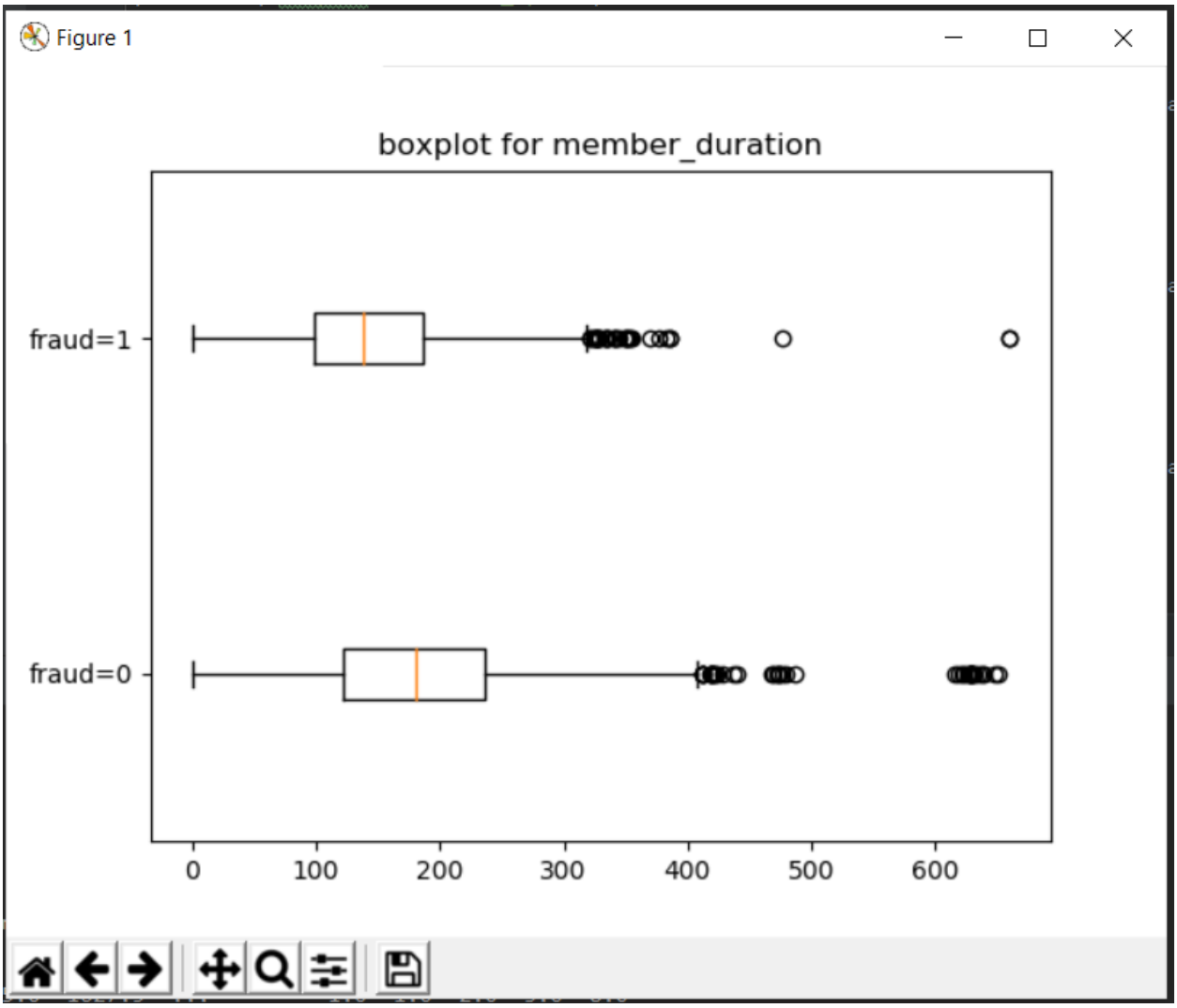


Figure 1
boxplot for member_duration
fraud=1
fraud=0
0
100
200
300
400
500
600

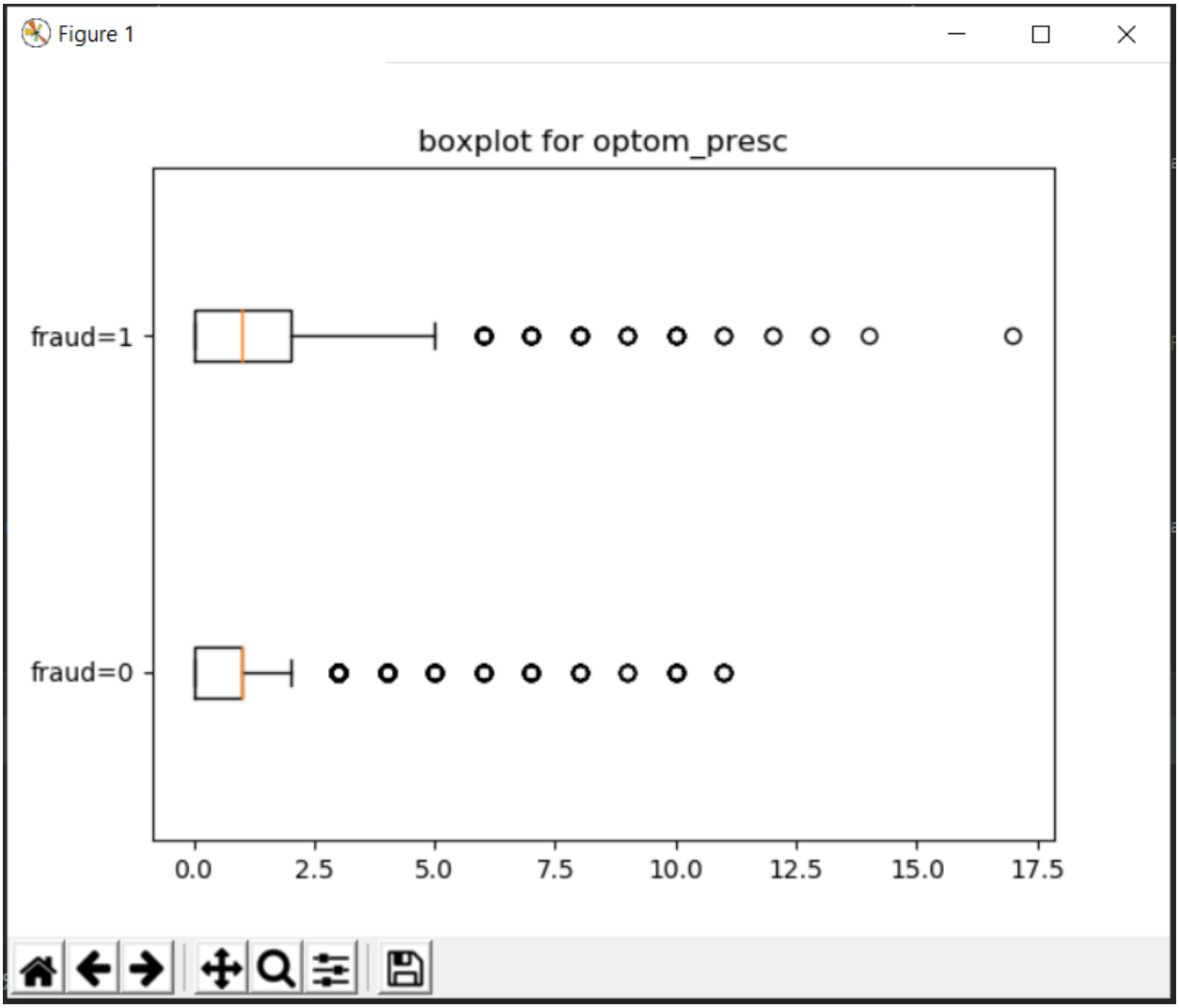
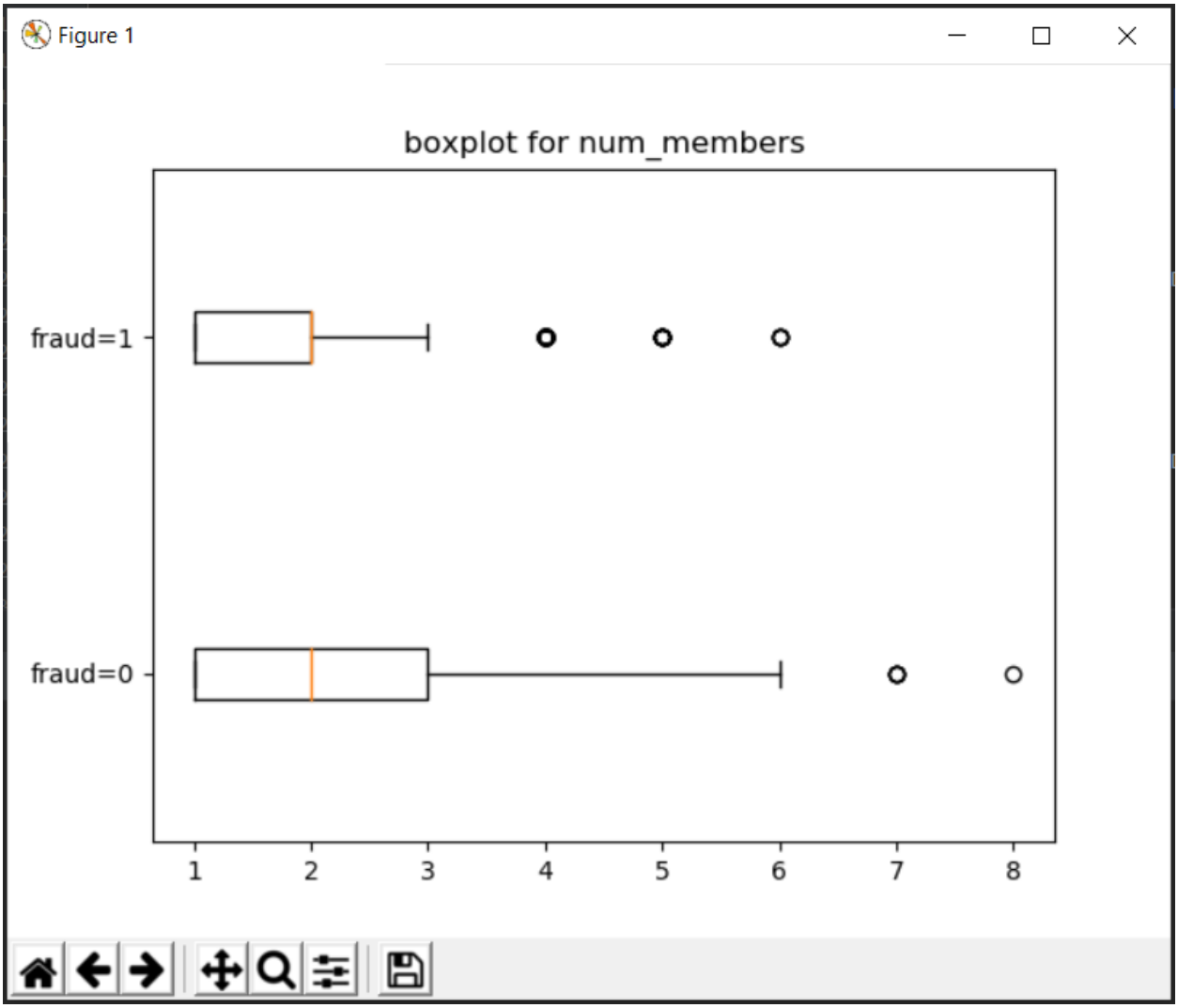


Figure 1
boxplot for optom_presc
fraud=1
fraud=0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5



c) (10 points) Orthonormalize interval variables and use the resulting variables for the nearest neighbor analysis. Use only the dimensions whose corresponding eigenvalues are greater than one.

i. (5 points) How many dimensions are used?

6 dimensions are used.

ii. (5 points) Please provide the transformation matrix? You must provide proof that the resulting variables are actually orthonormal.

Transformation matrix =

[[-6.49862374e-08 -2.41194689e-07 2.69941036e-07 -2.42525871e-07

-7.90492750e-07 5.96286732e-07]

[7.31656633e-05 -2.94741983e-04 9.48855536e-05 1.77761538e-03

3.51604254e-06 2.20559915e-10]

[-1.18697179e-02 1.70828329e-03 -7.68683456e-04 2.03673350e-05
1.76401304e-07 9.09938972e-12]
[1.92524315e-06 -5.37085514e-05 2.32038406e-05 -5.78327741e-05
1.08753133e-04 4.32672436e-09]
[8.34989734e-04 -2.29964514e-03 -7.25509934e-03 1.11508242e-05
2.39238772e-07 2.85768709e-11]
[2.10964750e-03 1.05319439e-02 -1.45669326e-03 4.85837631e-05
6.76601477e-07 4.66565230e-11]]

Here resultant matrix in identity. So, every vector in it has magnitude 1 and every pair of vectors are perpendicular to each other means their dot product will be 0. This proves that result is orthonormal.

d) (10 points) Use the NearestNeighbors module to execute the Nearest Neighbors algorithm using exactly <u>five</u> neighbors and the resulting variables you have chosen in c). The KNeighborsClassifier module has a score function.

i. (5 points) Run the score function, provide the function return value

Return value is 0.8778523489932886. that means model has 87.79% accuracy.

ii. (5 points) Explain the meaning of the score function return value.

Score function returns the accuracy of our trained model. It calculate accuracy on the bases of provided train data and targeted labels.

e) (5 points) For the observation which has these input variable values: TOTAL_SPEND = 7500, DOCTOR_VISITS = 15, NUM_CLAIMS = 3, MEMBER_DURATION = 127, OPTOM_PRESC = 2, and NUM_MEMBERS = 2, find its five neighbors. Please list their input variable values and the target values. *Reminder: transform the input observation using the results in c) before finding the neighbors*.

testData = [7500, 15, 3, 127, 2, 2]

targetData = [1]

transformed testdata = [[-0.02886529 0.00853837 -0.01333491 0.0176811 0.00793805 0.0044727]]

5 nearest neighbors = (array([[0. , 0.01045716, 0.01206995, 0.0121862 , 0.0140379]]), array([[588, 2897, 1199, 1246, 886]], dtype=int32))

f) (5 points) Follow-up with e), what is the predicted probability of fraudulent (i.e., FRAUD = 1)? If your predicted probability is greater than or equal to your answer in a), then the observation will be classified as fraudulent. Otherwise, non-fraudulent. Based on this criterion, will this observation be misclassified?

Predicted fraud probability = [[0. 1.]]

P(is fraud) = 1.0

P(is not fraud) = 0.0

This observation is fraudulent.

Here misclassification rate is 12.21%. based on this criterion this observation will not be misclassified.