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?

Answer: 0.3998667554864774

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

Answer: The minimum and maximum value of the field x are 26.3 and 35.4 respectively.

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?

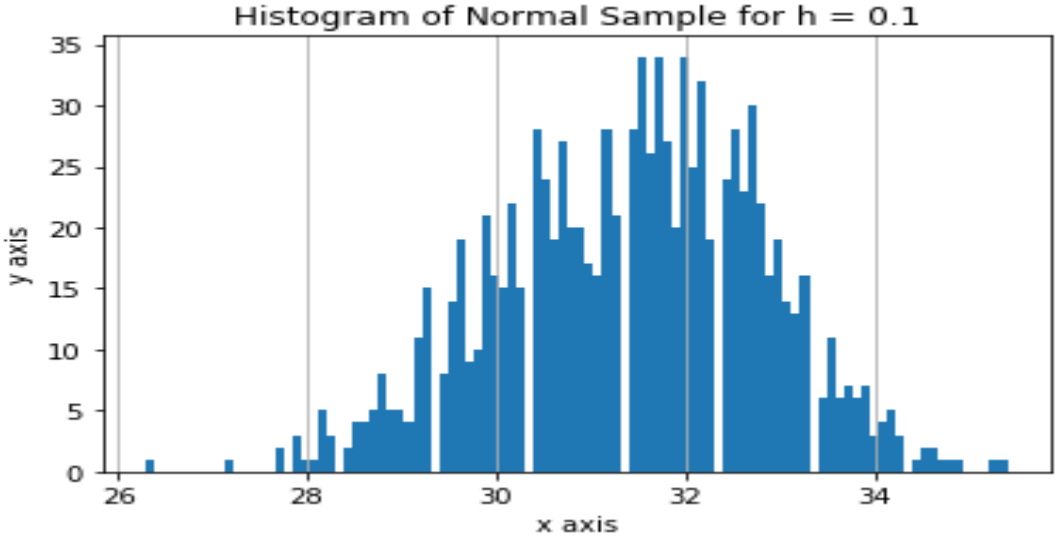
Answer: a = 26 and b = 36

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.

Answer: (mi, p(mi))

[(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),

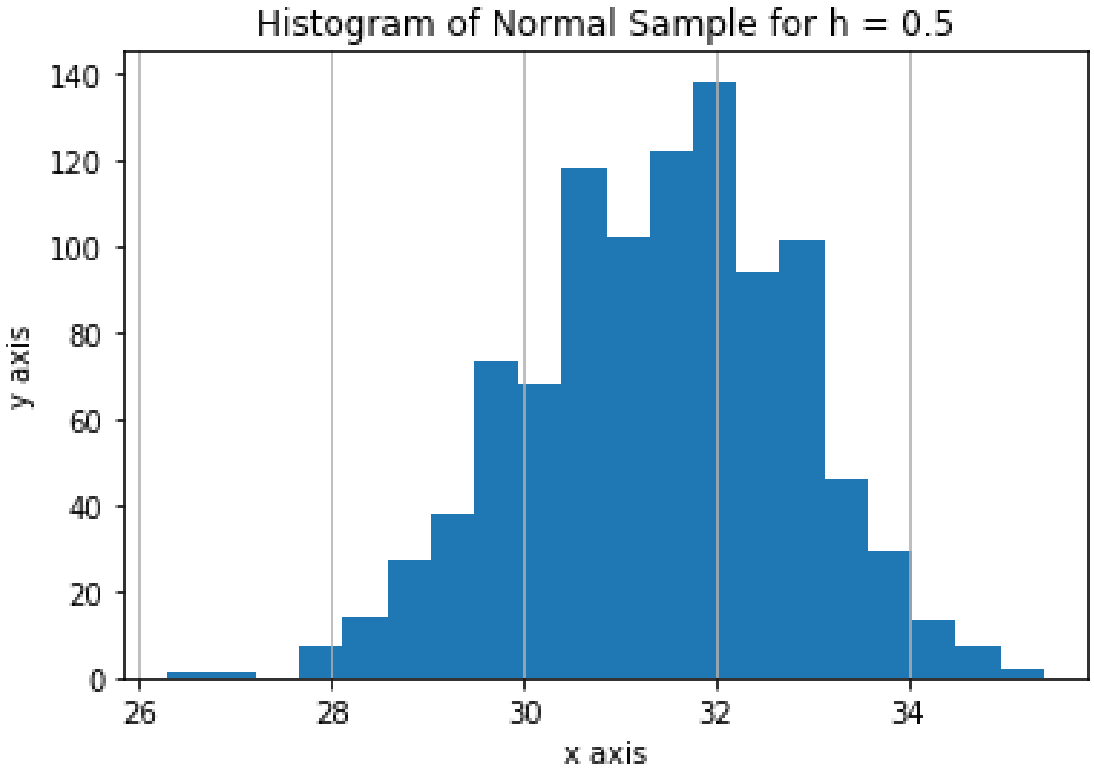
(30.25000000000006, 0.14985014985014983), (30.350000000000062, 0.2797202797202797), (30.450000000000063, 0.23976023976023975), (30.550000000000065, 0.1898101898101898), (30.650000000000066, 0.2697302697302697), (30.750000000000068, 0.19980019980019978), (30.85000000000007, 0.19980019980019978), (30.95000000000007, 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), (36.05000000000014, 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.

Answer : (mi, p(mi))

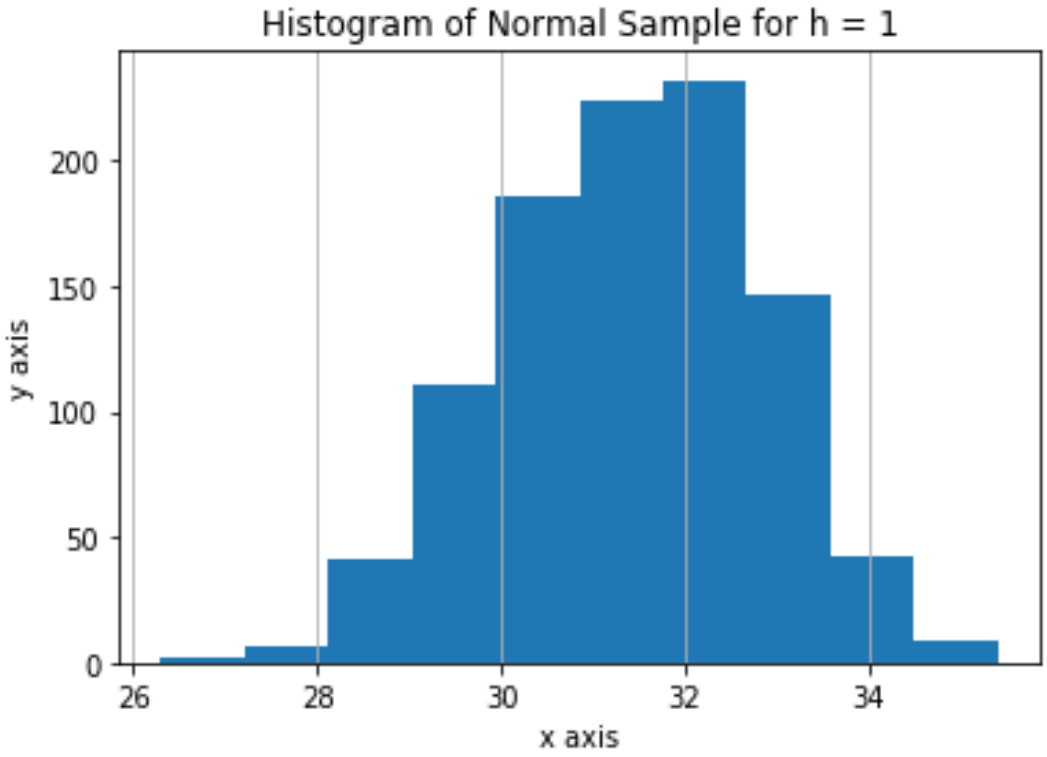
[(26.25, 0.001998001998001998), (26.75, 0.0), (27.25, 0.001998001998001998), (27.75, 0.0099900099900999), (28.25, 0.02197802197802198), (28.75, 0.04395604395604396), (29.25, 0.07592407592407592), (29.75, 0.11788211788211789), (30.25, 0.15984015984015984), (30.75, 0.17182817182817184), (31.25, 0.18581418581418582), (31.75, 0.2137862137862138), (32.25, 0.1998001998001998), (32.75, 0.18181818181818182), (33.25, 0.0979020979020979), (33.75, 0.05194805194805195), (34.25, 0.025974025974025976), (34.75, 0.00999000999000999), (35.25, 0.003996003996003996), (35.75, 0.0), (36.25, 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.

Answer : (mi, p(mi))

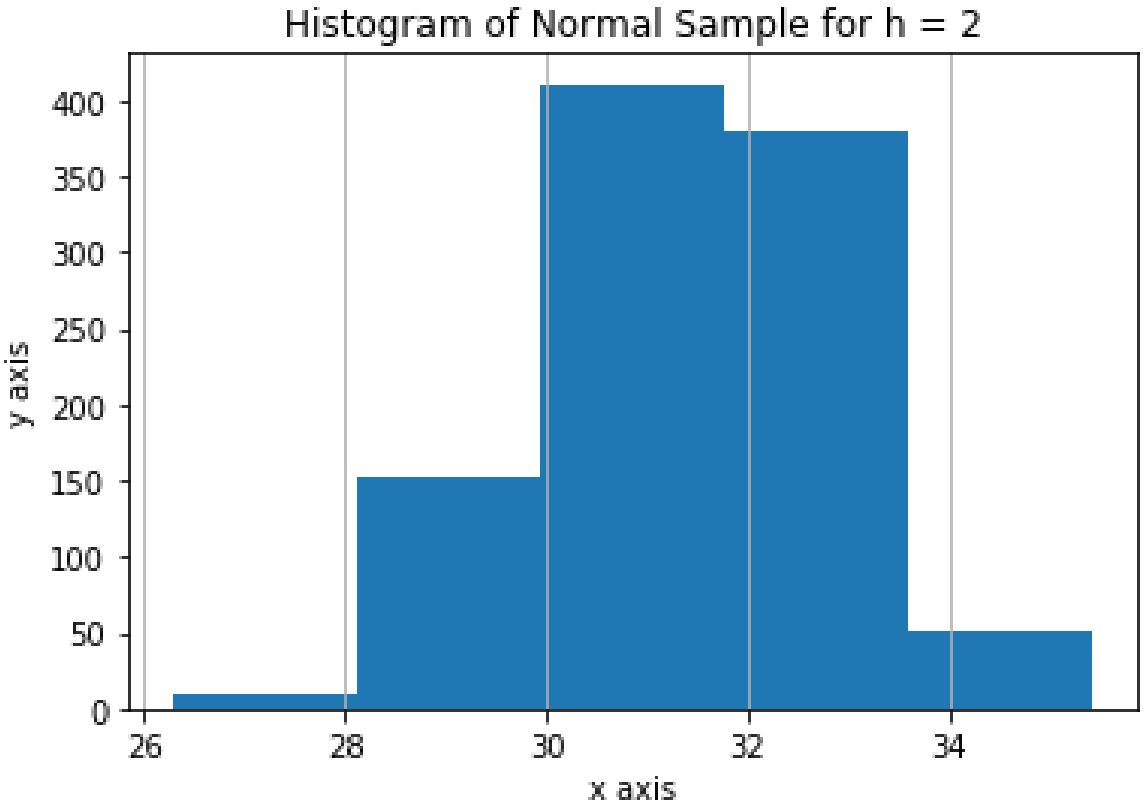
[(26.5, 0.000999000999000999), (27.5, 0.005994005994005994), (28.5, 0.03696303696303696), (29.5, 0.1108891108891109), (30.5, 0.18981018981018982), (31.5, 0.23376623376623376), (32.5, 0.21878121878121878), (33.5, 0.08591408591408592), (34.5, 0.01998001998001998), (35.5, 0.001998001998001998), (36.5, 0.0)]



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.

Answer: (mi, p(mi))

[(27.0, 0.0034965034965034965), (29.0, 0.07642357642357642), (31.0, 0.2202797202797203), (33.0, 0.16183816183816183), (35.0, 0.01098901098901099), (37.0, 0.0)]



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.

Answer: In my opinion, histogram with 0.5 value of h (the one in answer 1. e) provides us better insights than rest other histogram. Though it doesn't have such fine details like the histogram with value of h = 0.1 but it does give fine details better than rest other histograms. Also, it gives some insight about big picture not as good as histogram with value of h as 1 and 2 but better than histogram with 0.1 as value of h.

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?

Answer:

min 26.3
25% 30.4
50% 31.5
75% 32.4
max 35.4
Lower whisker = 27.4
Upper whisker = 35.4

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?

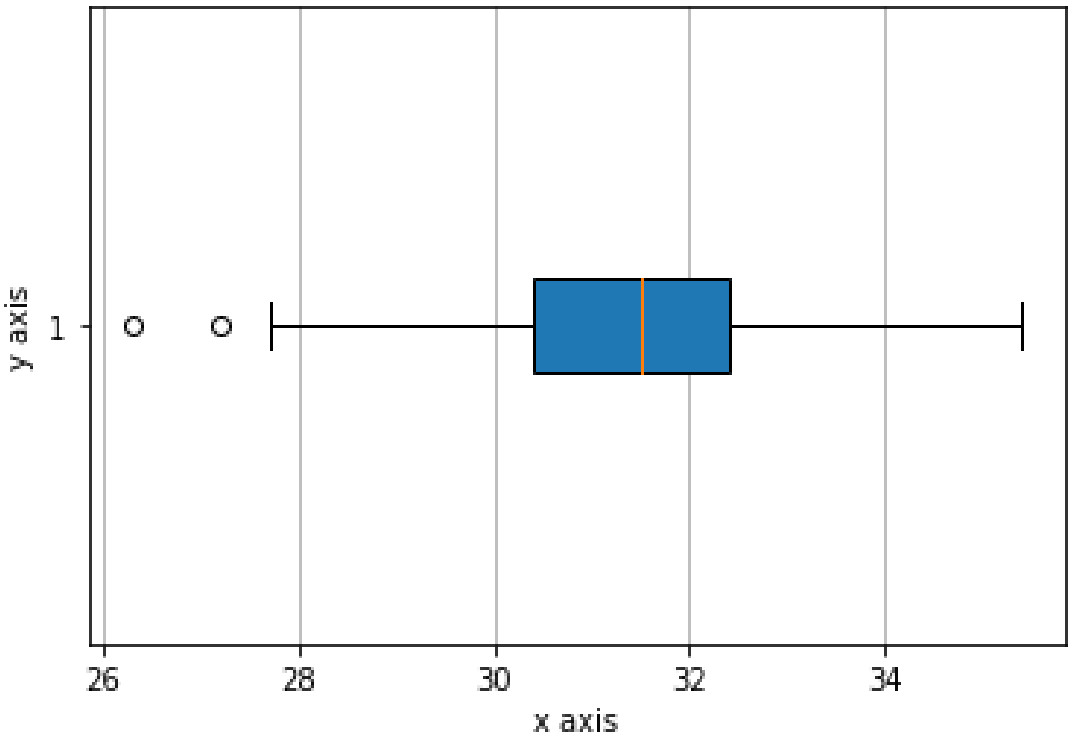
Answer:

Group 0

Min = 26.3
Q1 = 29.4
Q2 = 30.0
Q3 = 30.6
max = 32.2
Lower whisker = 27.6
Upper whisker = 32.4
Group 1
min = 29.1
Q1 = 31.4
Q2 = 32.1
Q3 = 32.7
max = 35.4
Lower whisker = 29.45
Upper whisker = 34.65

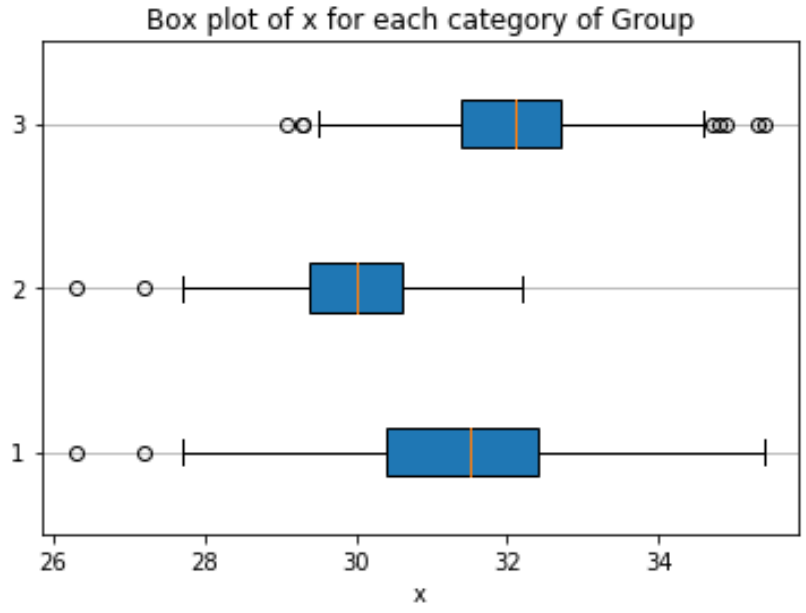
c) (5 points) Draw a boxplot of x (without the group) using the Python boxplot function. Can you tell if the Python's boxplot has displayed the 1.5 IQR whiskers correctly?

Answer: Yes



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.

Answer:



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

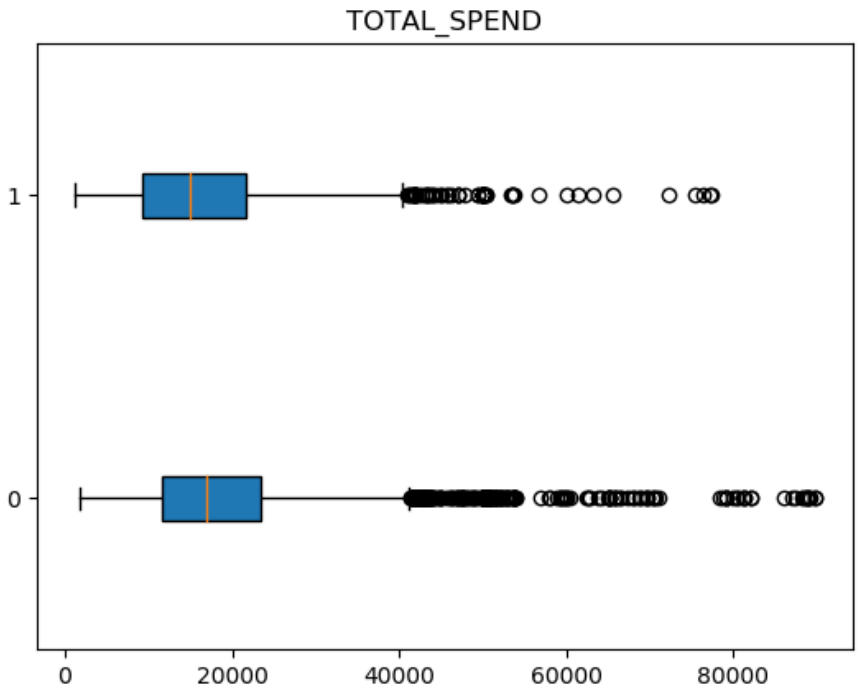
You are asked to use the Nearest Neighbors algorithm to predict the likelihood of fraud.

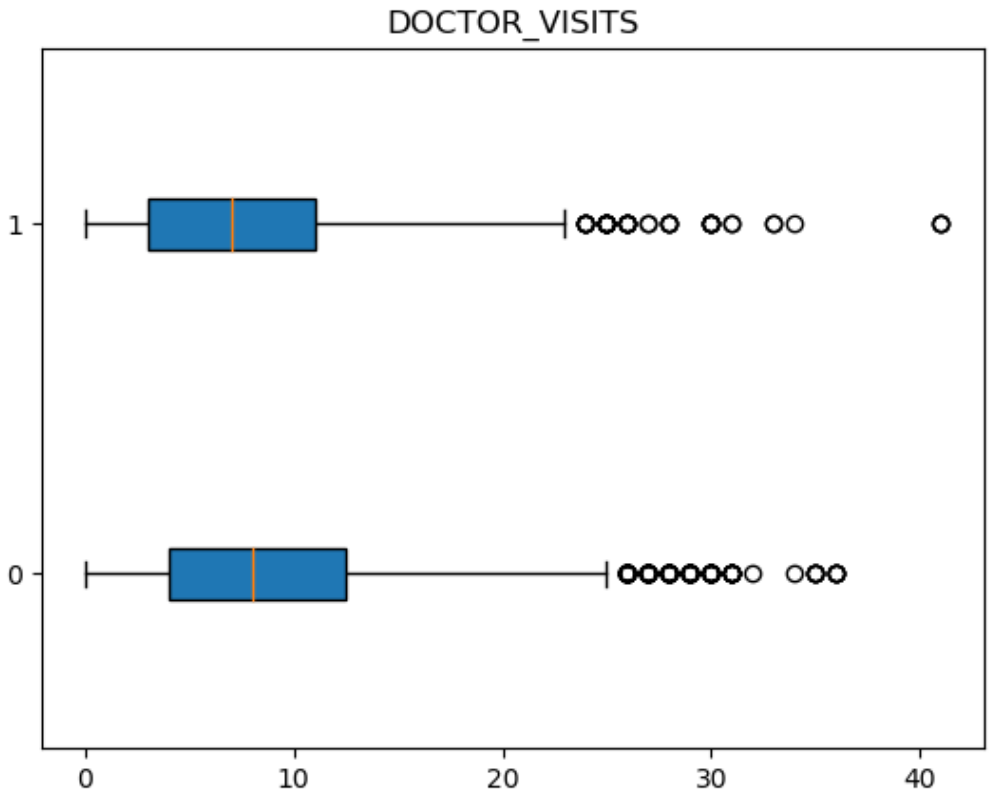
a) (5 points) What percent of investigations are found to be fraudulent? Please give your answer up to 4 decimal places.

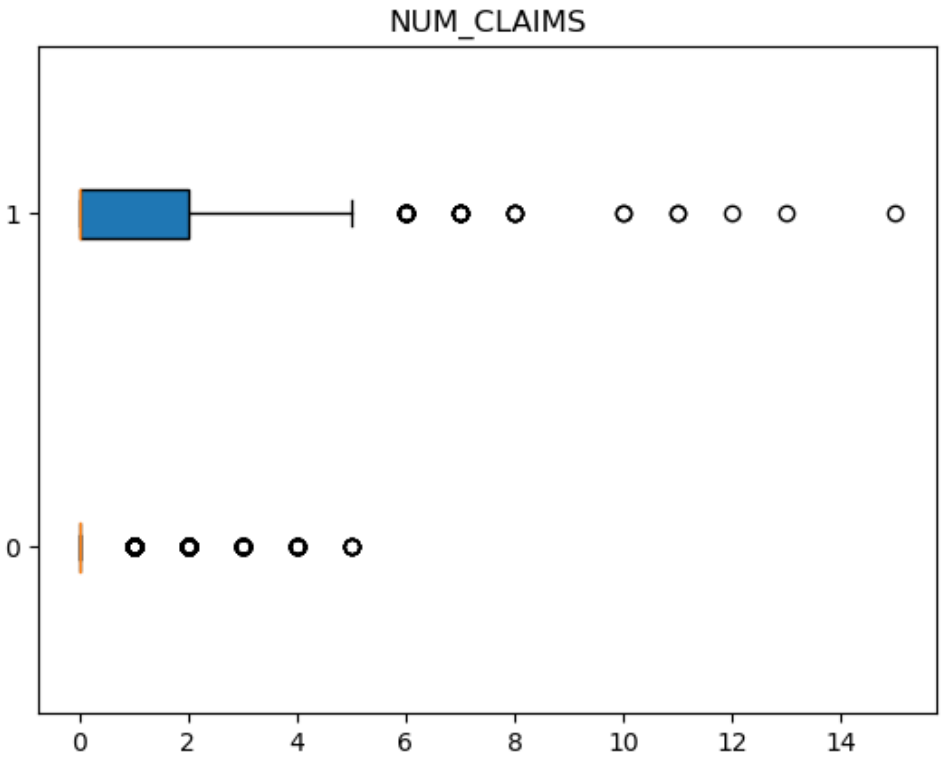
Answer: 19.9497

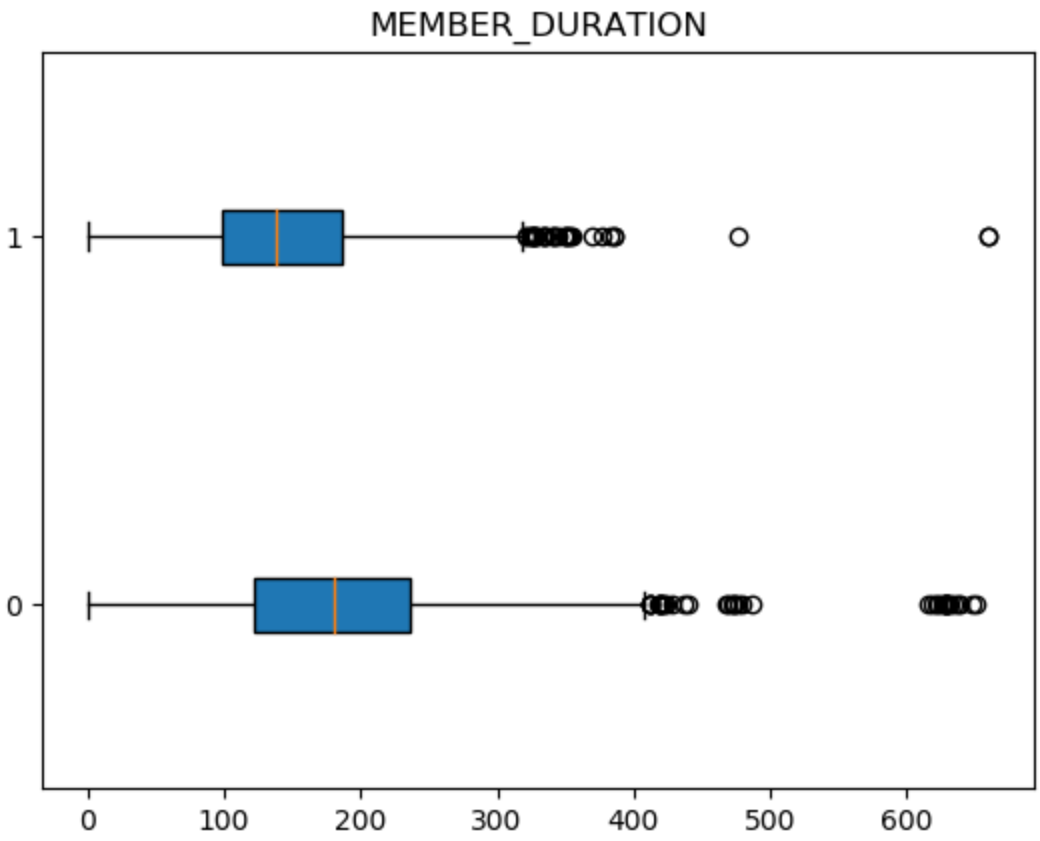
b) (5 points) Use the BOXPLOT function to produce horizontal box-plots. For each interval variable, one box-plot for the fraudulent observations, and another box-plot for the non-fraudulent observations. These two box-plots must appear in the same graph for each interval variable.

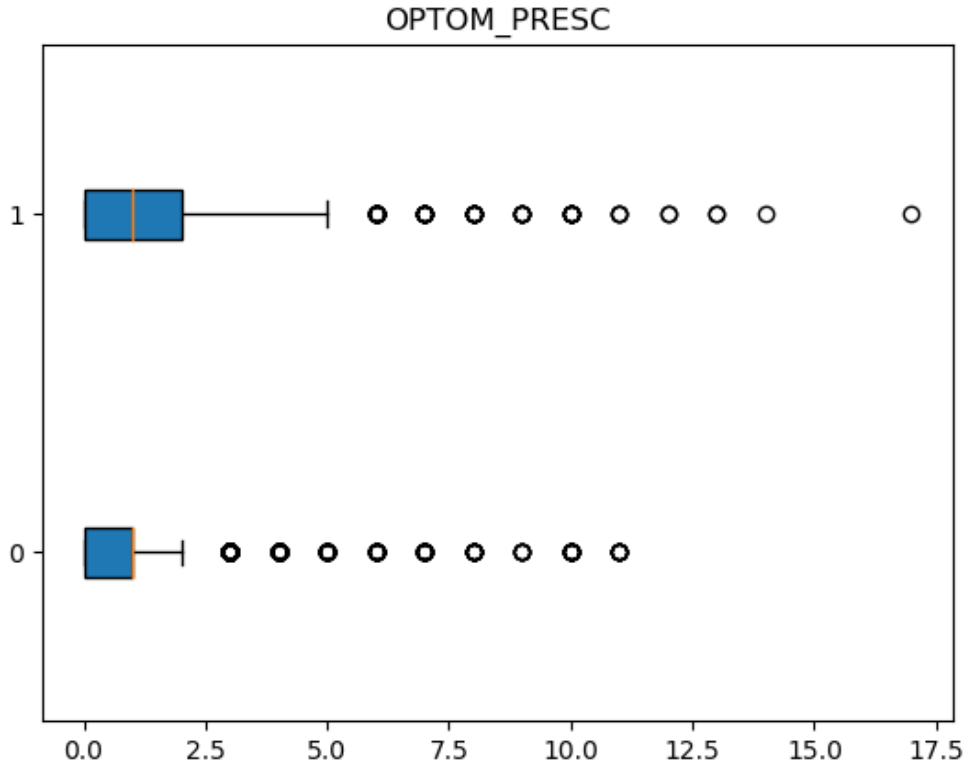
Answer:

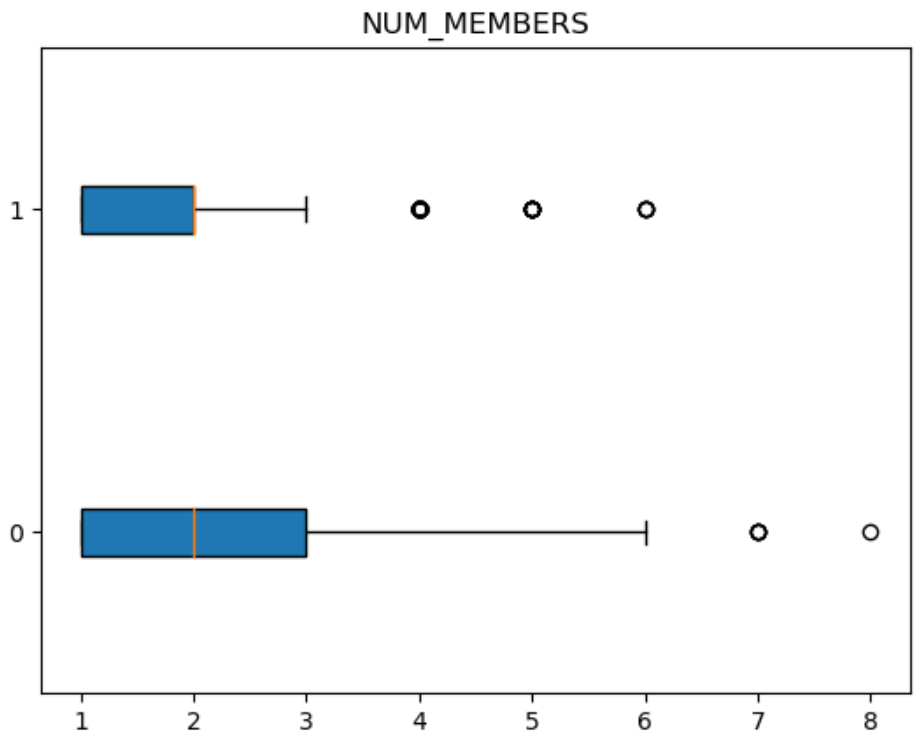



DOCTOR_VISITS
1
0
0
10
20
30
40


NUM_CLAIMS
1
0
0
2
4
6
8
10
12
14


MEMBER_DURATION
1
0
0
100
200
300
400
500
600


OPTOM_PRESC
1
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?

Answer: 6

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

Answer: 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]]
```

```
Transformed x =
 [[ 5.96859502e-03  1.02081629e-02 -6.64664861e-03  1.39590283e-02
   9.39352141e-03  6.56324665e-04]
 [-2.09672310e-02  5.01932025e-03  8.51930607e-04  5.16174400e-03
   1.22658834e-02  7.75702220e-04]
 [ 7.64597676e-03  1.97528525e-02 -7.38335310e-03 -1.71350853e-03
   1.50348109e-02  8.95075830e-04]
 ...
 [-7.18408819e-05 -1.62580211e-02  2.75078514e-02 -7.13245766e-03
  -4.74021952e-02  5.31896971e-02]
 [-1.80147801e-04 -1.62154130e-02  2.76213381e-02 -9.17125411e-03
  -4.76625006e-02  5.35474776e-02]
 [-2.21157680e-03 -2.73884697e-02  2.93391341e-02 -7.81347172e-03
  -4.70861917e-02  5.36071324e-02]]
Identity Matrix =
 [[ 1.00000000e+00 -3.00432422e-16 -4.61219604e-16  5.45323877e-15
   1.20996962e-15 -1.28911638e-16]
 [-3.00432422e-16  1.00000000e+00 -6.44449771e-16 -2.76820667e-14
  -1.23512311e-15  7.78890841e-16]
 [-4.61219604e-16 -6.44449771e-16  1.00000000e+00  3.50891191e-15
   1.00613962e-16 -2.25514052e-16]
 [ 5.45323877e-15 -2.76820667e-14  3.50891191e-15  1.00000000e+00
   1.14860378e-14 -3.47812057e-15]
 [ 1.20996962e-15 -1.23512311e-15  1.00613962e-16  1.14860378e-14
   1.00000000e+00 -6.31439345e-16]
 [-1.28911638e-16  7.78890841e-16 -2.25514052e-16 -3.47812057e-15
  -6.31439345e-16  1.00000000e+00]]
Orthonormalize x =
 [[-6.56324665e-04  9.39352141e-03  1.39590283e-02 -6.64664861e-03
   1.02081629e-02 -5.96859502e-03]
 [-7.75702220e-04  1.22658834e-02  5.16174400e-03  8.51930607e-04
   5.01932025e-03  2.09672310e-02]
 [-8.95075830e-04  1.50348109e-02 -1.71350853e-03 -7.38335310e-03
   1.97528525e-02 -7.64597676e-03]
 ...
 [-5.31896971e-02 -4.74021952e-02 -7.13245766e-03  2.75078514e-02
  -1.62580211e-02  7.18408819e-05]
 [-5.35474776e-02 -4.76625006e-02 -9.17125411e-03  2.76213381e-02
  -1.62154130e-02  1.80147801e-04]
 [-5.36071324e-02 -4.70861917e-02 -7.81347172e-03  2.93391341e-02
  -2.73884697e-02  2.21157680e-03]]
Identity Matrix =
 [[ 1.00000000e+00 -1.11022302e-16  9.67108338e-17 -7.63278329e-17
   1.99493200e-17 -7.91467586e-18]
```

```
[-1.11022302e-16  1.00000000e+00  1.83447008e-16  2.25514052e-17
 -1.38777878e-17 -3.03576608e-18]
[ 9.67108338e-17  1.83447008e-16  1.00000000e+00 -6.67868538e-17
 -7.91467586e-18  2.55465137e-17]
[-7.63278329e-17  2.25514052e-17 -6.67868538e-17  1.00000000e+00
 -9.10729825e-17  1.63660318e-16]
[ 1.99493200e-17 -1.38777878e-17 -7.91467586e-18 -9.10729825e-17
  1.00000000e+00  3.25748543e-16]
[-7.91467586e-18 -3.03576608e-18  2.55465137e-17  1.63660318e-16
  3.25748543e-16  1.00000000e+00]]
```

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

Answer: 0.8414429530201343

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

Answer: The score function return value indicates the accuracy of our model we developed. Also, it gives us an idea on data misclassified when we subtract it from 1.

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*.

Answer: My Neighbors = [[588 2897 1199 1246 886]]

Input variable value = [7500,15,3,127,2,2]

Target value = 1

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?

Answer: Predicted probability of fraudulent = 1

No, this observation will not be misclassified based on the mentioned criteria because predicted fraudulent probability is 1.