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?

```
Ans: Bin width for histogram of X = 0.3998667554864773
```

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

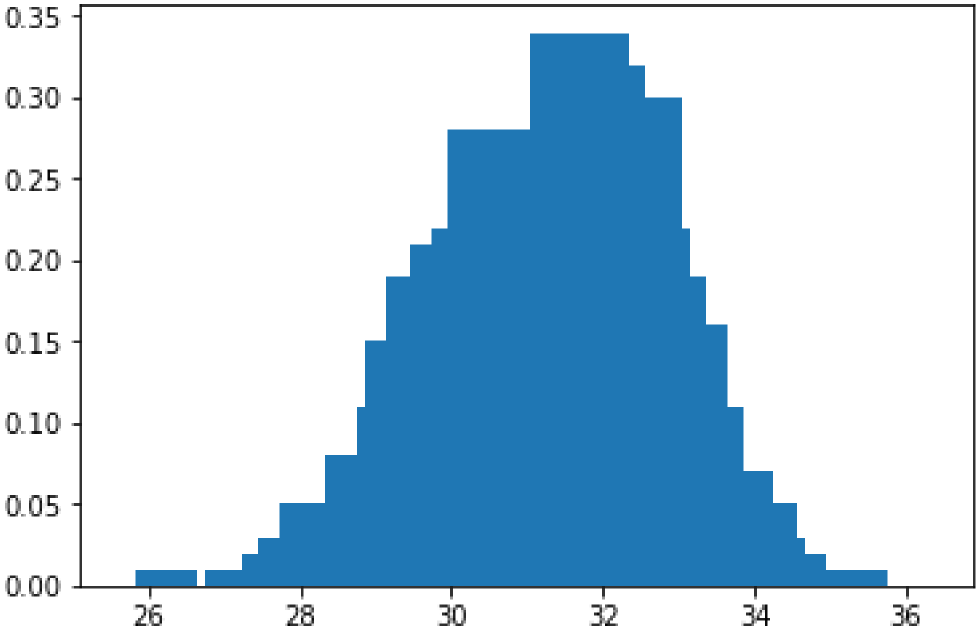
```
Ans:    Minimum = 26.3
         Maximum= 35.4
```

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?

Ans: a = 26 b =36

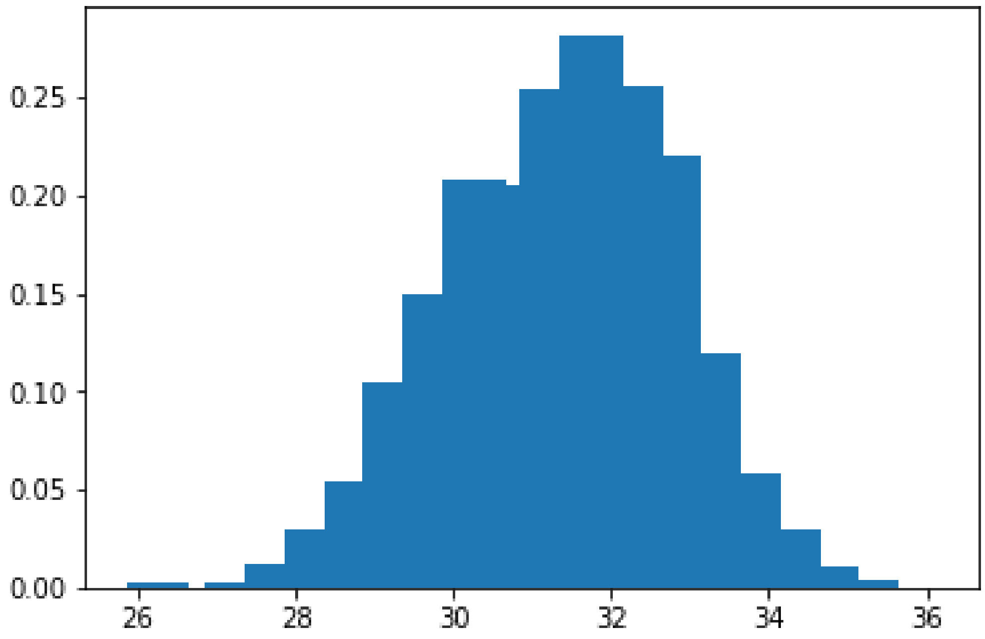
d) (5 points) Use h = 0.1, minimum = a and maximum = b. List the coordinates of the density estimator. Paste the histogram drawn using Python or your favorite graphing tools.

```
Coordinates by density estimation: [0.0, 0.0, 0.00999000999000999,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.00999000999000999, 0.0, 0.
0, 0.0, 0.0, 0.01998001998001998, 0.0, 0.02997002997002997, 0.009990
00999000999, 0.00999000999000999, 0.04995004995004995, 0.0299700299
7002997, 0.01998001998001998, 0.03996003996003996, 0.039960039960039
96, 0.04995004995004995, 0.07992007992007992, 0.04995004995004995,
 0.04995004995004995, 0.03996003996003996, 0.10989010989010987, 0.1
4985014985014983, 0.07992007992007992, 0.13986013986013984, 0.189810
1898101898, 0.0899100899100899, 0.09990009990009989, 0.2097902097902
0976, 0.15984015984015984, 0.14985014985014983, 0.21978021978021975,
 0.14985014985014983, 0.2797202797202797, 0.23976023976023975, 0.189
8101898101898, 0.2697302697302697, 0.19980019980019978, 0.1998001998
0019978, 0.16983016983016982, 0.15984015984015984, 0.279720279720279
7, 0.20979020979020976, 0.2797202797202797, 0.33966033966033965, 0.2
597402597402597, 0.33966033966033965, 0.2697302697302697, 0.19980019
980019978, 0.33966033966033965, 0.24975024975024973, 0.3196803196803
197, 0.1898101898101898, 0.23976023976023975, 0.2797202797202797, 0.
22977022977022976, 0.29970029970029965, 0.21978021978021975, 0.15984
015984015984, 0.1898101898101898, 0.13986013986013984, 0.12987012987
012986, 0.15984015984015984, 0.05994005994005994, 0.1098901098901098
7, 0.05994005994005994, 0.06993006993006992, 0.05994005994005994, 0.
06993006993006992, 0.02997002997002997, 0.03996003996003996, 0.04995
0049950049945, 0.02997002997002997, 0.00999000999000999, 0.019980019
98001998, 0.01998001998001998, 0.00999000999000999, 0.00999000999000
999, 0.00999000999000999, 0.0, 0.0, 0.0, 0.00999000999000999, 0.0099
9000999000999, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```



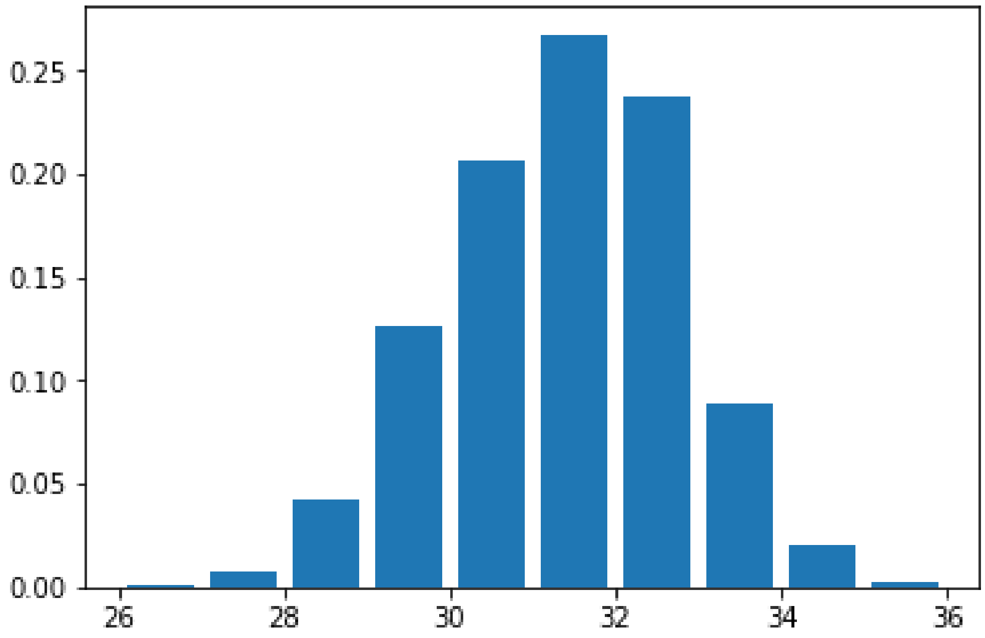
e) (5 points) Use h = 0.5, minimum = a and maximum = b. List the coordinates of the density estimator. Paste the histogram drawn using Python or your favorite graphing tools.

```
Coordinates by density estimation: [0.001998001998001998, 0.0, 0.001
998001998001998, 0.011988011988011988, 0.029970029970029972, 0.05394
6053946053944, 0.1038961038961039, 0.14985014985014986, 0.2077922077
922078, 0.2057942057942058, 0.25374625374625376, 0.2817182817182817,
 0.25574425574425574, 0.21978021978021978, 0.11988011988011989, 0.05
7942057942057944, 0.029970029970029972, 0.00999000999000999, 0.00399
6003996003996, 0.0]
```



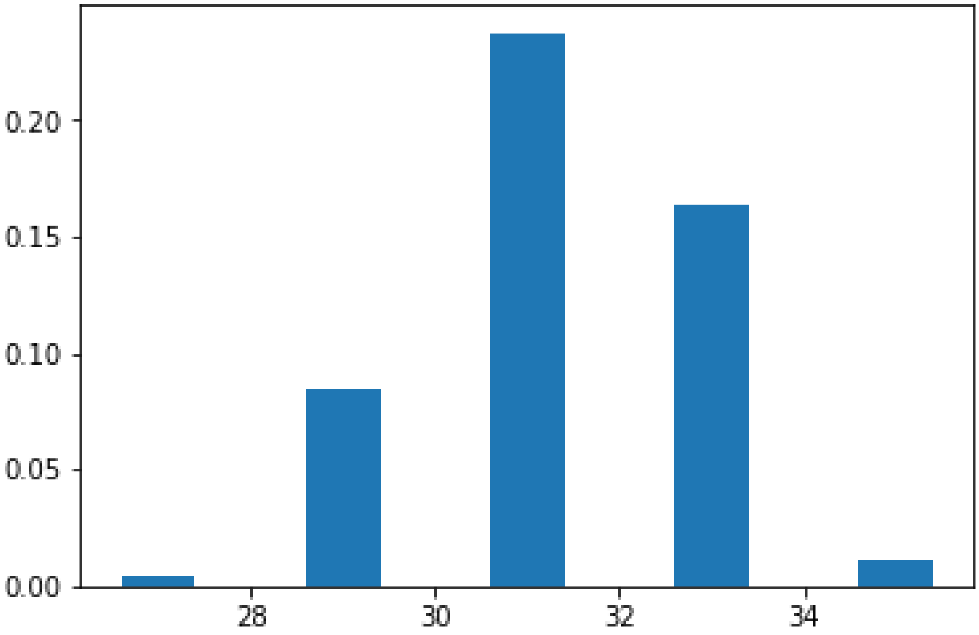
f) (5 points) Use h = 1, minimum = a and maximum = b. List the coordinates of the density estimator. Paste the histogram drawn using Python or your favorite graphing tools.

```
Coordinates by density estimation: [0.000999000999000999, 0.00699300
6993006993, 0.04195804195804196, 0.12687312687312688, 0.206793206793
20679, 0.2677322677322677, 0.23776223776223776, 0.0889110889110889l,
 0.01998001998001998, 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 by density estimation: [0.003996003996003996, 0.08441558
441558442, 0.23726273726273725, 0.16333666333666333, 0.010989010989
01099]
```



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.

Among the four histograms, h=0.1 provides too accurate details but it doesn't provide the insight for the shape. Whereas h=1 and h=2 though has the shape as clear as water it doesn't provide much details to notice. So I believe h=0.5 would be considered the best among the four.

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?

Min	Q-25	Q-50	Q-75	Max
26.3	30.4	31.5	32.4	35.4
1.5 IQR whiskers = 27.4 , 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?

Group – 0

```
Min  Q-25  Q-50 Q-75  Max
26.3 29.4  30.  30.6 32.2
```

1.5 IQR whiskers

```
27.599999999999994
32.400000000000006
```

Group-1

Min Q-25 Q-50 Q-75 Max

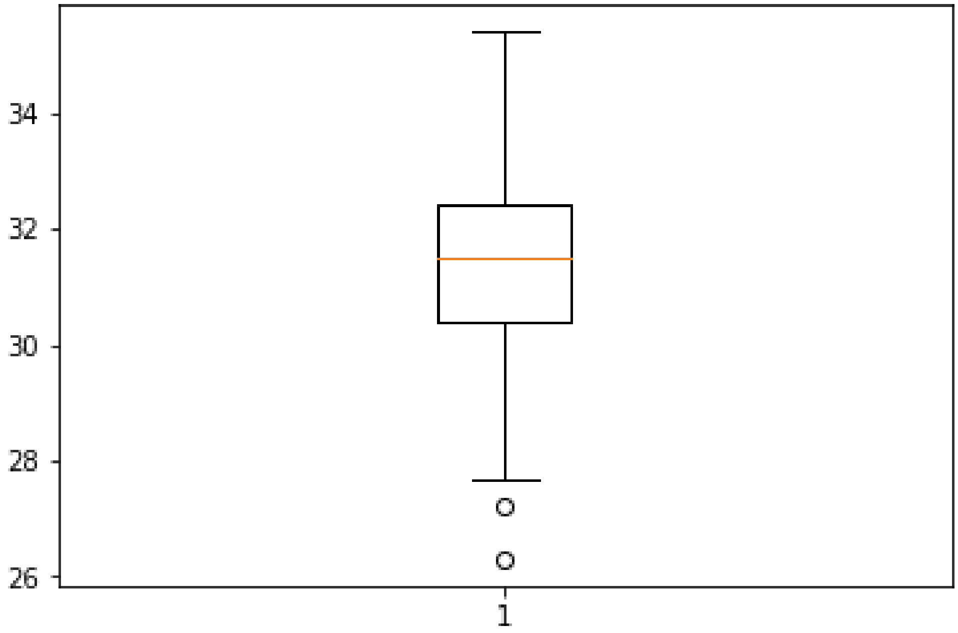
```
29.1 31.4 32.1 32.7 35.4
```

1.5 IQR whiskers

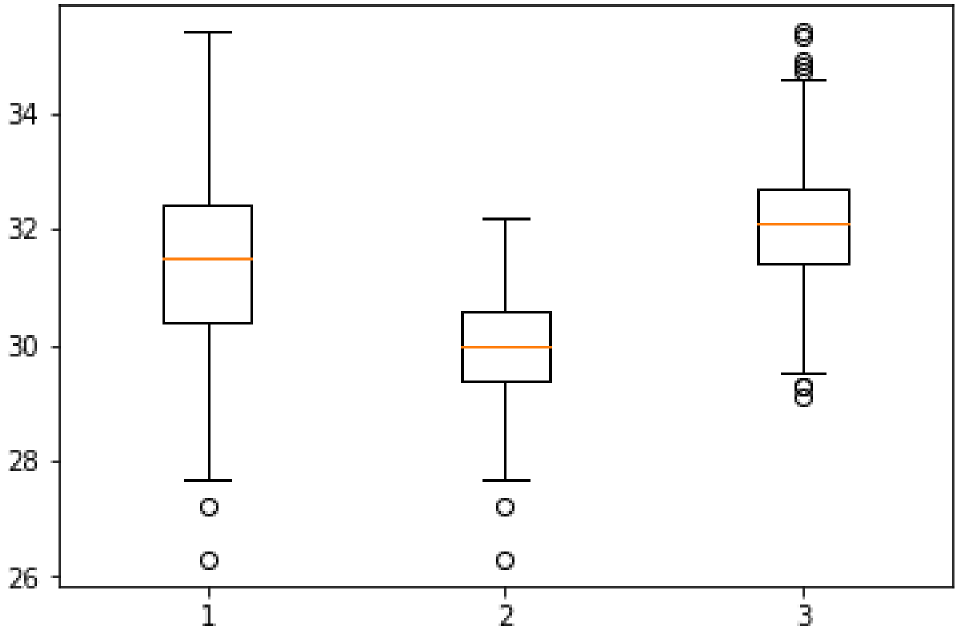
```
29.449999999999992
34.650000000000006
```

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?

The 1.5 IQR whisker values are min = 27.4 and max = 35.4 and if you observe from the group 0 and group 1 values of 1.5 IQR whiskers then it is min = 27.5 and max = 34.65 which is nearly equal and the difference is negligible. Hence i would say that it has plotted 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.



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.

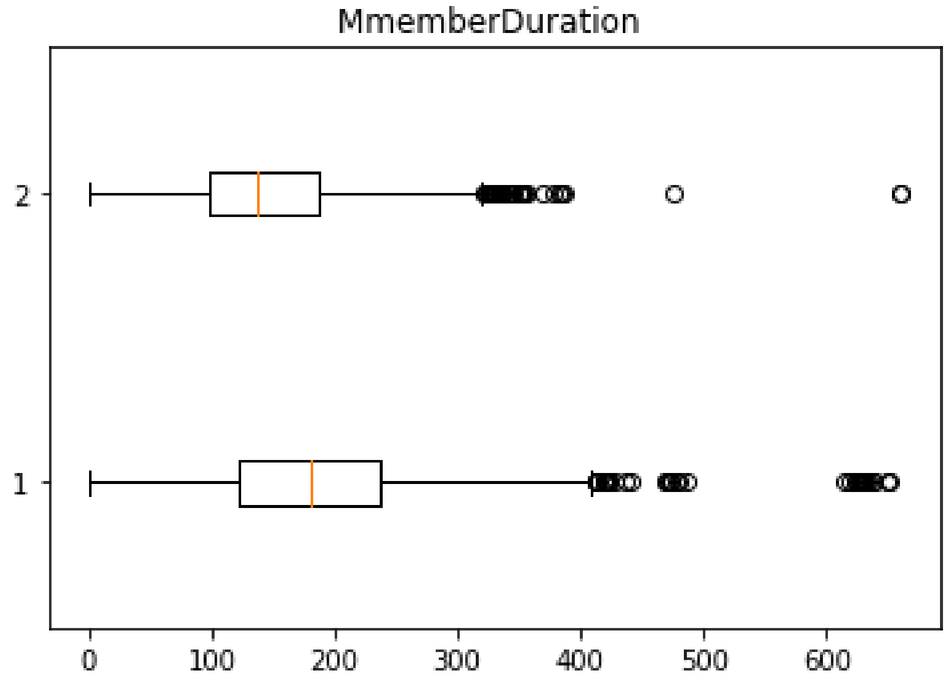
1. TOTAL_SPEND: Total amount of claims in dollars
2. DOCTOR_VISITS: Number of visits to a doctor
3. NUM_CLAIMS: Number of claims made recently
4. MEMBER_DURATION: Membership duration in number of months
5. OPTOM_PRESC: Number of optical examinations
6. NUM_MEMBERS: Number of members covered

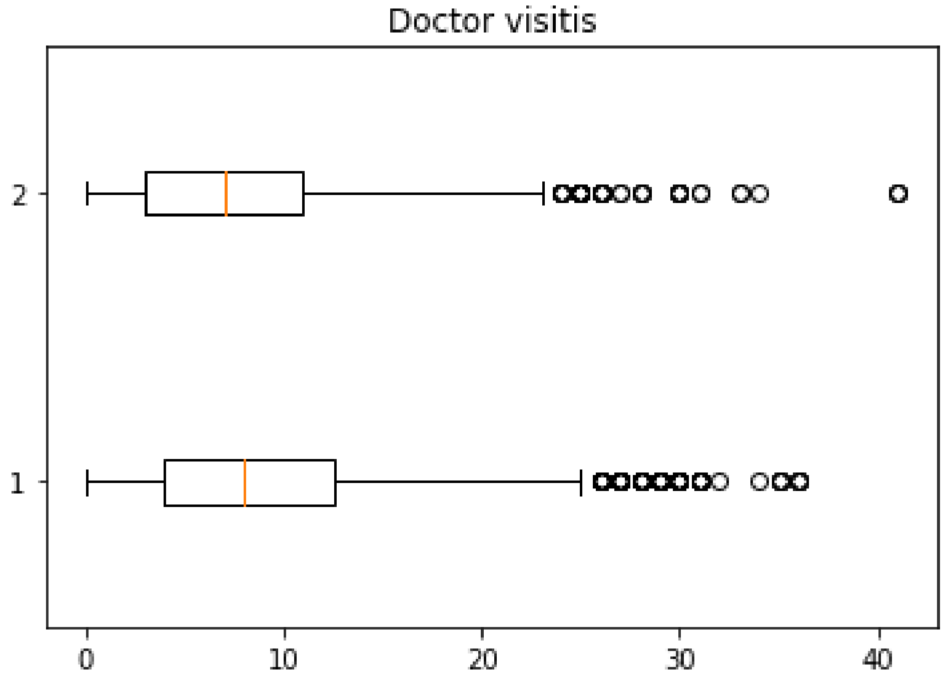
You are asked to use the Nearest Neighbors algorithm to predict the likelihood of fraud.

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

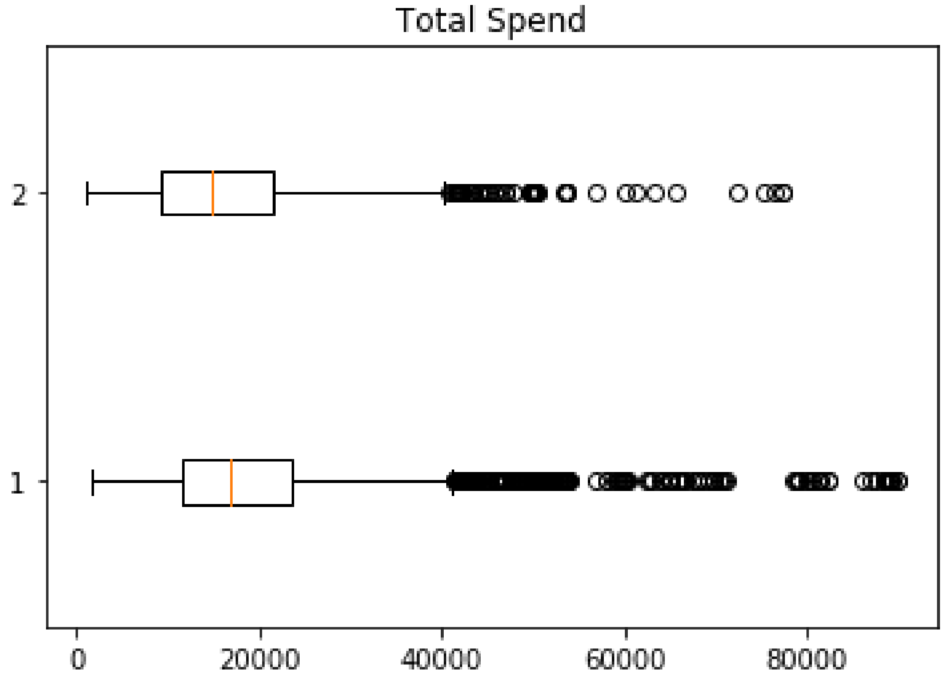
```
Fraud % =  19.9497
```

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.

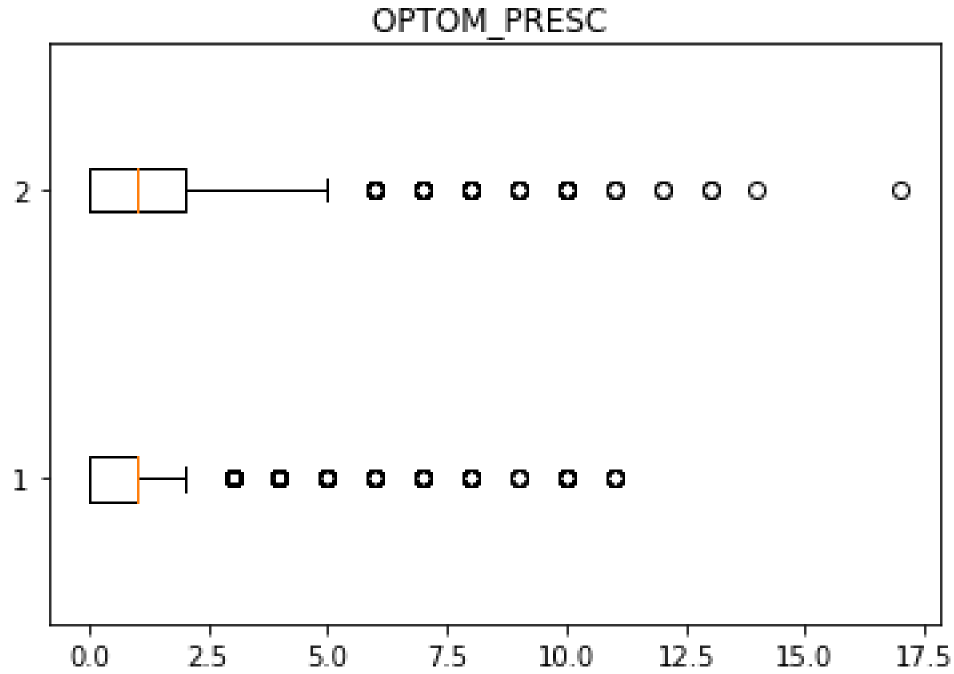


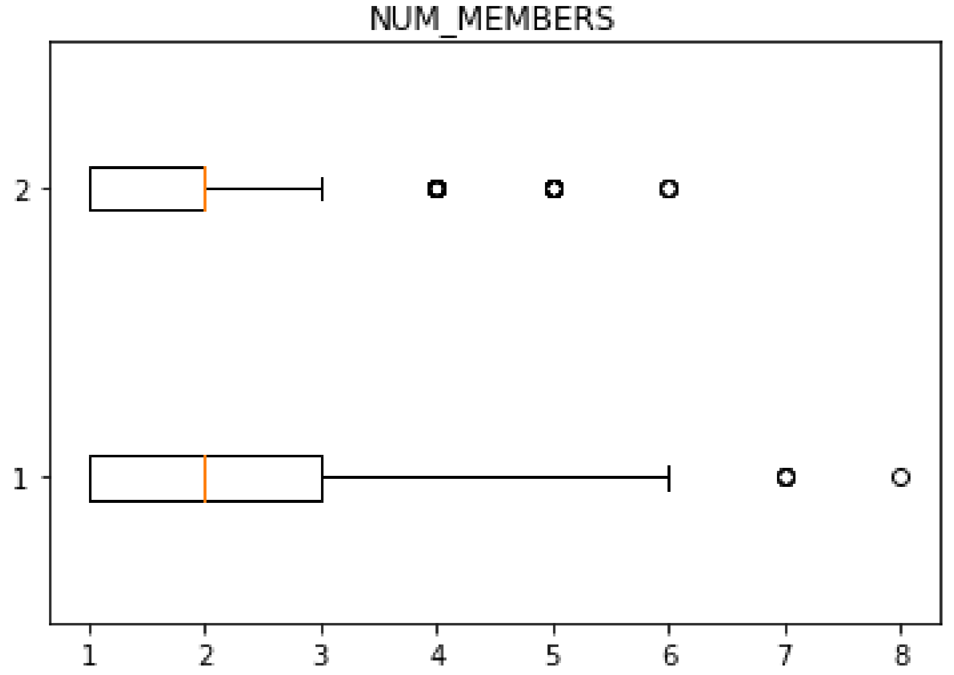


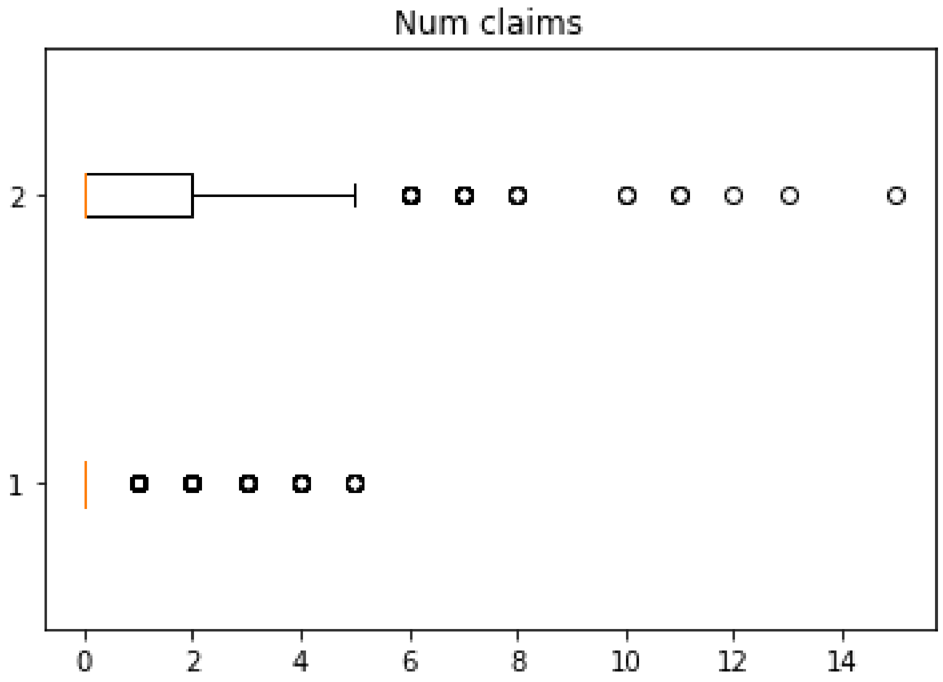
Doctor visitis



Total Spend


OPTOM_PRESC
2
1
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


NUM_MEMBERS
2
1
1
2
3
4
5
6
7
8



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?
Ans: 6

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]]
```

If you multiply the transformed X matrix with its transpose then it should give identity matrix like this.

transformedX = np.matmul(x,transformationMatrix)

xtx = np.matmul(transformedX.transpose(),transformedX);

```
[[ 1. -0.  0. -0.  0. -0.]
 [-0.  1.  0.  0. -0. -0.]
 [ 0.  0.  1. -0. -0.  0.]
 [-0.  0. -0.  1. -0.  0.]
 [ 0. -0. -0. -0.  1.  0.]
 [-0. -0.  0.  0.  0.  1.]]
```

d) (10 points) Use the NearestNeighbors module to execute the Nearest Neighbors algorithm using exactly five 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

```
Ans:   0.8778523489932886
```

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

It provides the accuracy of the values and the data given. This is based on the training and the testing data values provided to the scikit functions from the scikit libraries.

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*.

```
CASE_ID             589
FRAUD                 1
TOTAL_SPEND        7500
DOCTOR_VISITS        15
NUM_CLAIMS            3
MEMBER_DURATION     127
OPTOM_PRESC           2
NUM_MEMBERS           2
Name: 588, dtype: int64
-----------------------------------------------------------------
CASE_ID             2898
FRAUD                  1
TOTAL_SPEND        16000
DOCTOR_VISITS         18
NUM_CLAIMS             3
MEMBER_DURATION      146
OPTOM_PRESC            3
NUM_MEMBERS            2
Name: 2897, dtype: int64
-----------------------------------------------------------------
```

```
CASE_ID            1200
FRAUD                 1
TOTAL_SPEND       10000
DOCTOR_VISITS        16
NUM_CLAIMS            3
MEMBER_DURATION     124
OPTOM_PRESC           2
NUM_MEMBERS           1
Name: 1199, dtype: int64
----------------------------------------------------------------
CASE_ID            1247
FRAUD                 1
TOTAL_SPEND       10200
DOCTOR_VISITS        13
NUM_CLAIMS            3
MEMBER_DURATION     119
OPTOM_PRESC           2
NUM_MEMBERS           3
Name: 1246, dtype: int64
----------------------------------------------------------------
CASE_ID            887
FRAUD                1
TOTAL_SPEND       8900
DOCTOR_VISITS       22
NUM_CLAIMS           3
MEMBER_DURATION    166
OPTOM_PRESC          1
NUM_MEMBERS          2
Name: 886, dtype: int64
```

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?

Ans: The predicted probability for the above cases is 1 (FRAUD=1). Hence it is greater than the answer I obtained in (a). so based on that we can conclude that this won't be misclassified.