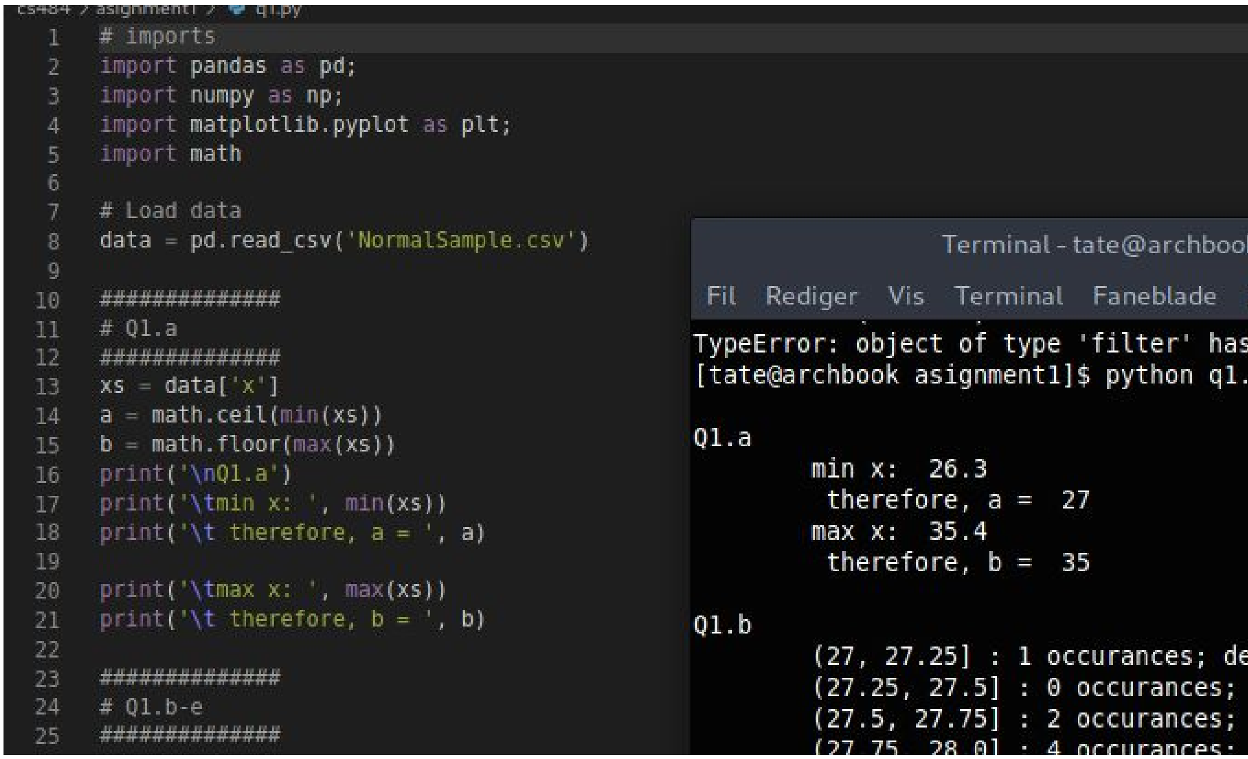
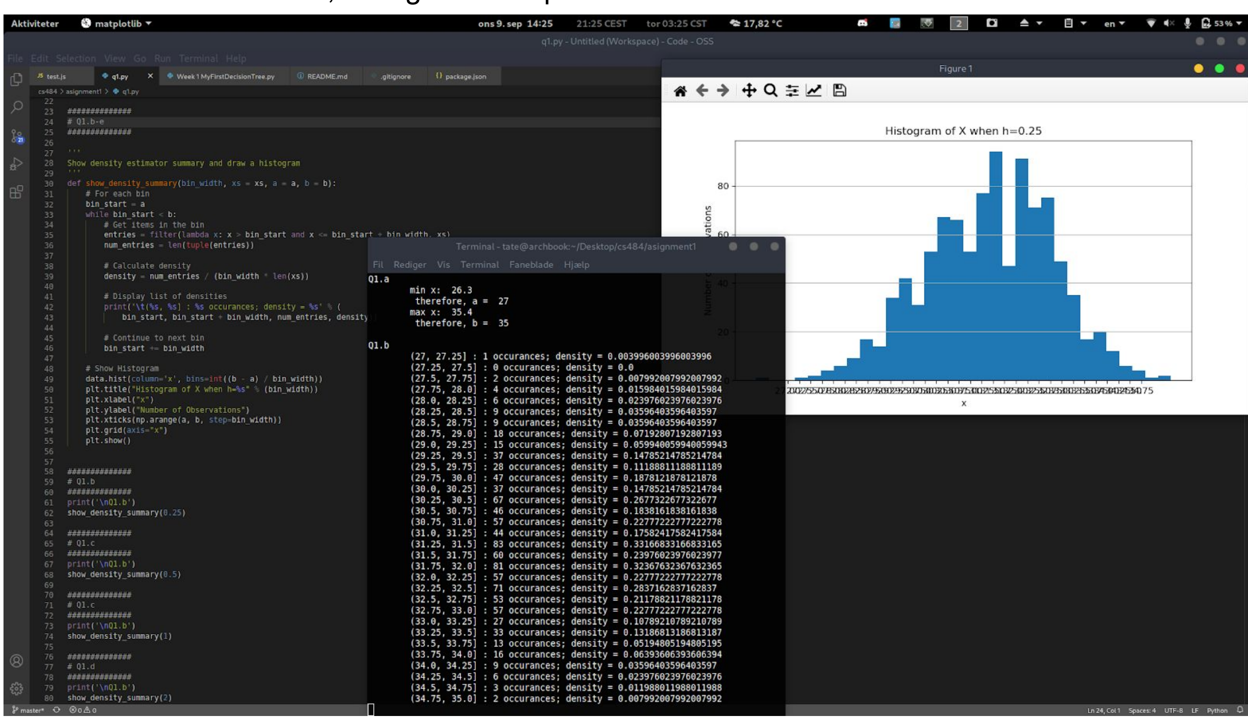
Dustin Van Tate Testa
CS484 Assignment 1

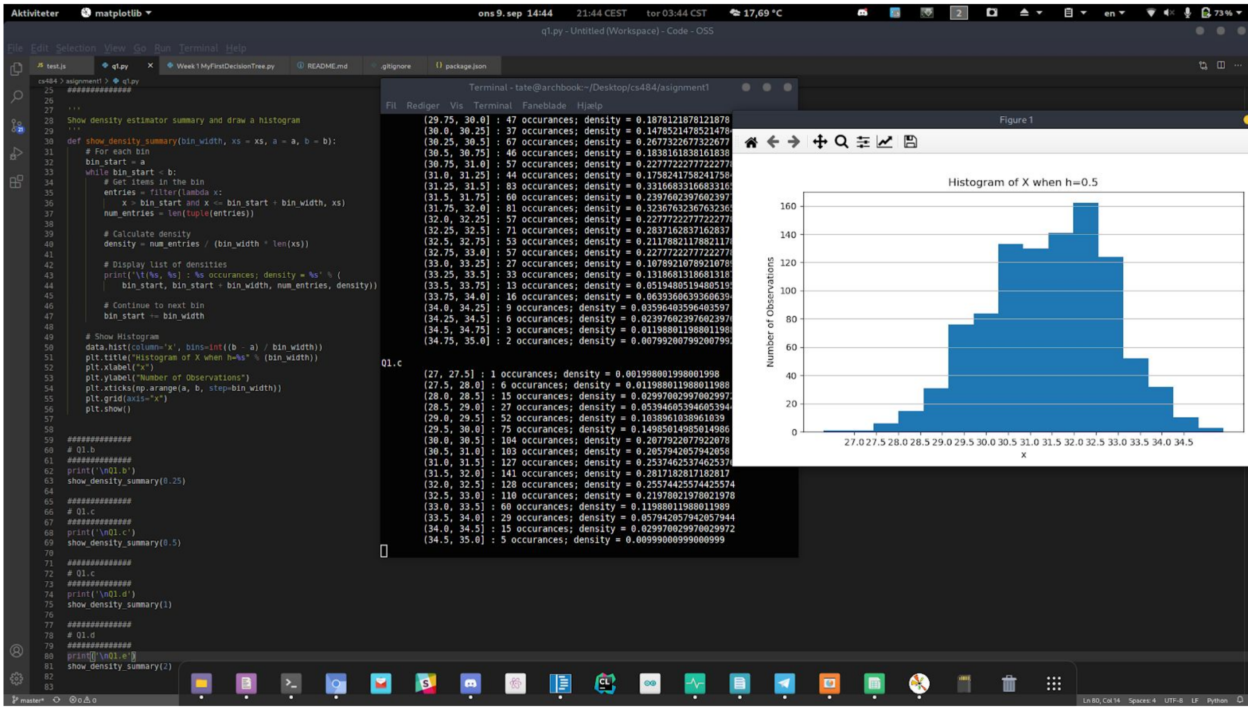
1. See attachment ‘q1.py’ for full python program
    a. A = 26 ; B = 36



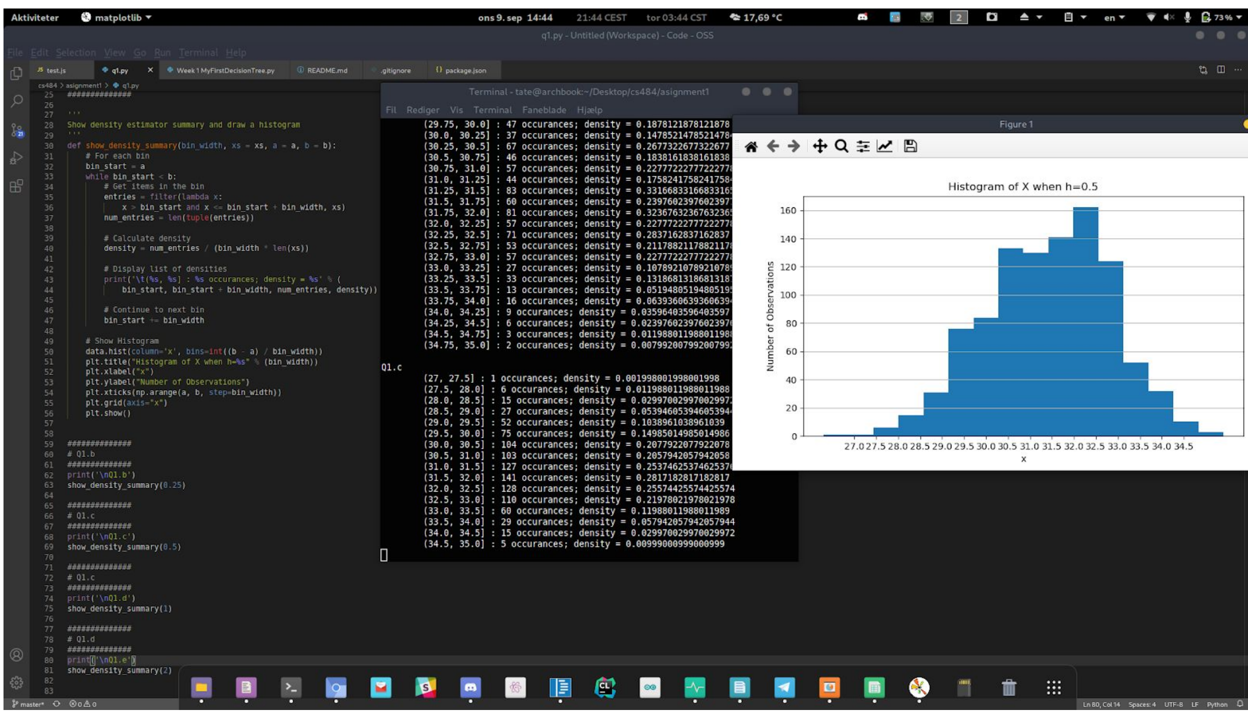
    b. List: see terminal window, histogram: see plot



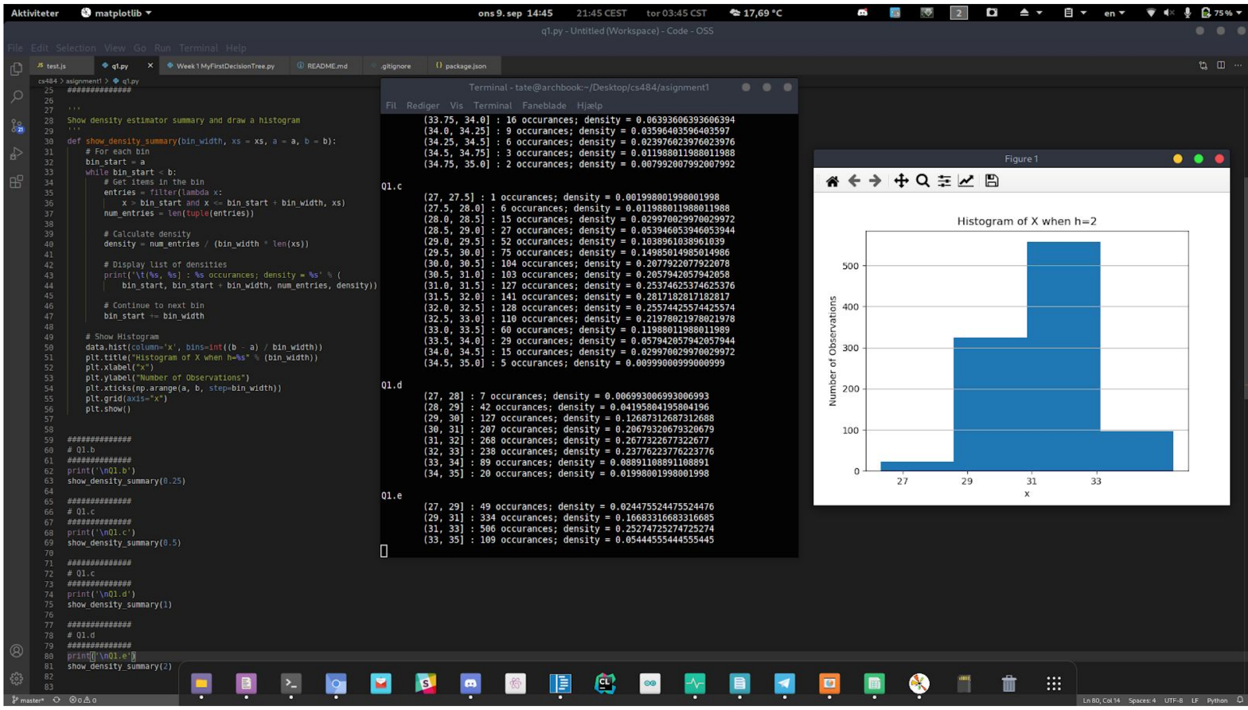
c. List: see terminal, histogram see plot



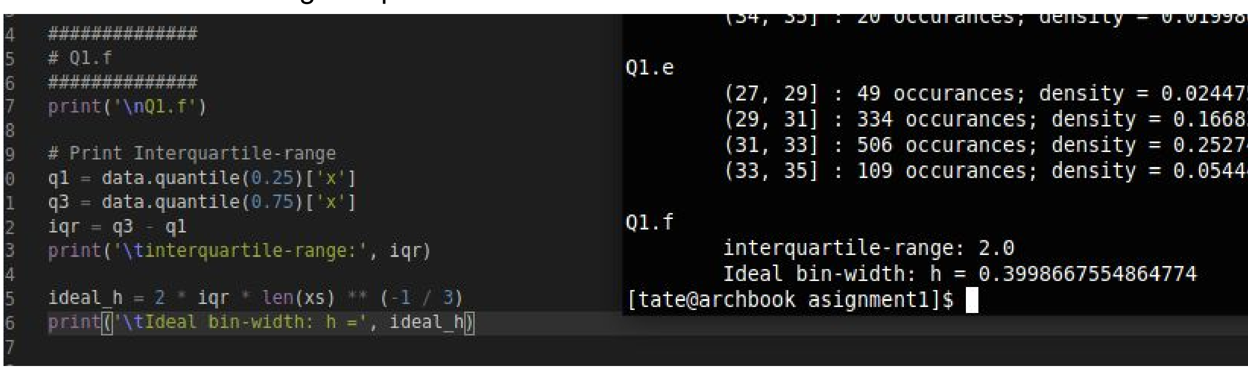
d. List: see terminal, histogram: see plot



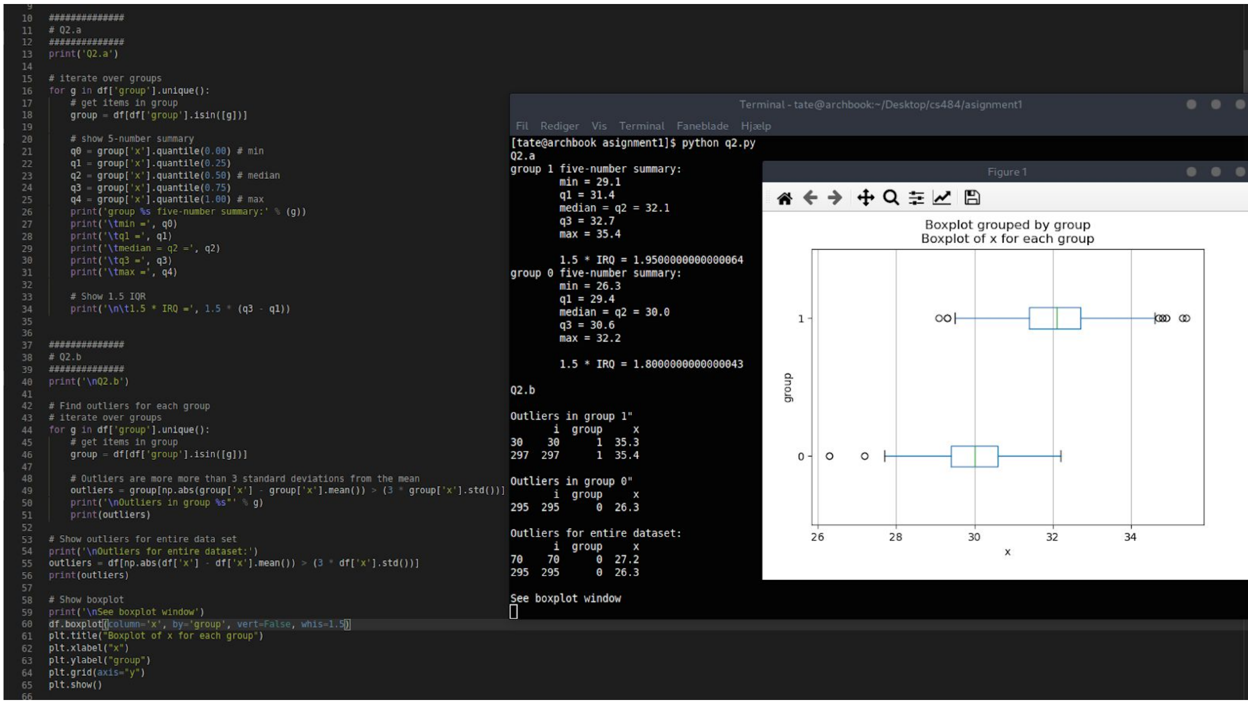
e. List: see terminal, plot see chart



f. **According to the strategy suggested by izenman, A. J. (1991) in the american statistical association journal, the ideal bin width (h) is ~0.4, among the 4 we tested here the 0.5 one is closest to this value.** Without context however, this value might not be the most appropriate. And there are some minor things you can see in the h=0.25 histogram that aren't present in the h=0.5 data set. For example, the x=(31.5, 31.75] interval has fewer members than its neighbors, which although unlikely, could potentially be statistically significant. For example, if these are ages of humans in years there may have been an approximately 3 month period with world events that caused fertility rates to decrease during that period.



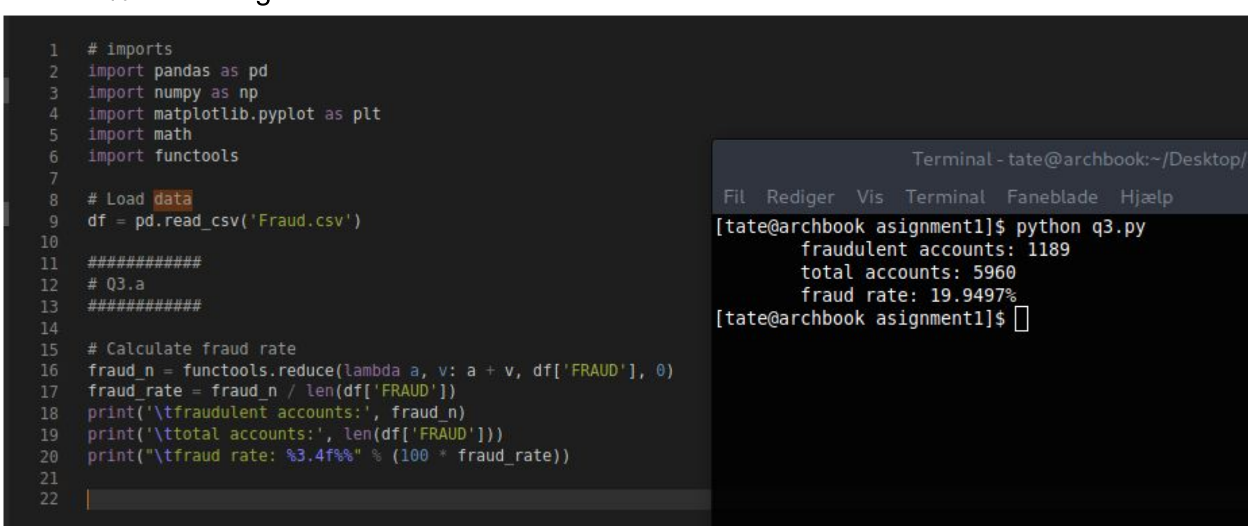
2. See attachment 'q2.py' for python program



a. The five number summary is as shown in the terminal above, the 1.5 iqr is also shown as 1.5 times the difference between q3 and q1 as shown in the terminal above

b. Note that there are only two unique groups for the dataset specified. See the terminal above for outliers, see the 'Figure 1' window above for the box-plot

3. See attached q3.py for full python program
   a. 19.9497% of investigations were found to be fraudulent



b. Code used to generate plots:

```python

############
# Q3.b
############
print('Q3.b')

# Box plot for each field
for field in df:
    # Skip case_id lol
    if field in ('CASE_ID', 'FRAUD'):
        continue

    # Draw boxplot
    df.boxplot(column=field, by='FRAUD', vert=False, whis=1.5, sym='')
    plt.title("Boxplot of %s in relation to FRAUD" % field)
    plt.xlabel(field)
    plt.ylabel('fraud')
    plt.grid(axis="y")
    plt.show()

```

```
^C
[tate@arc
Q3.a
      f
      t
      f
Q3.b
```

Plots:

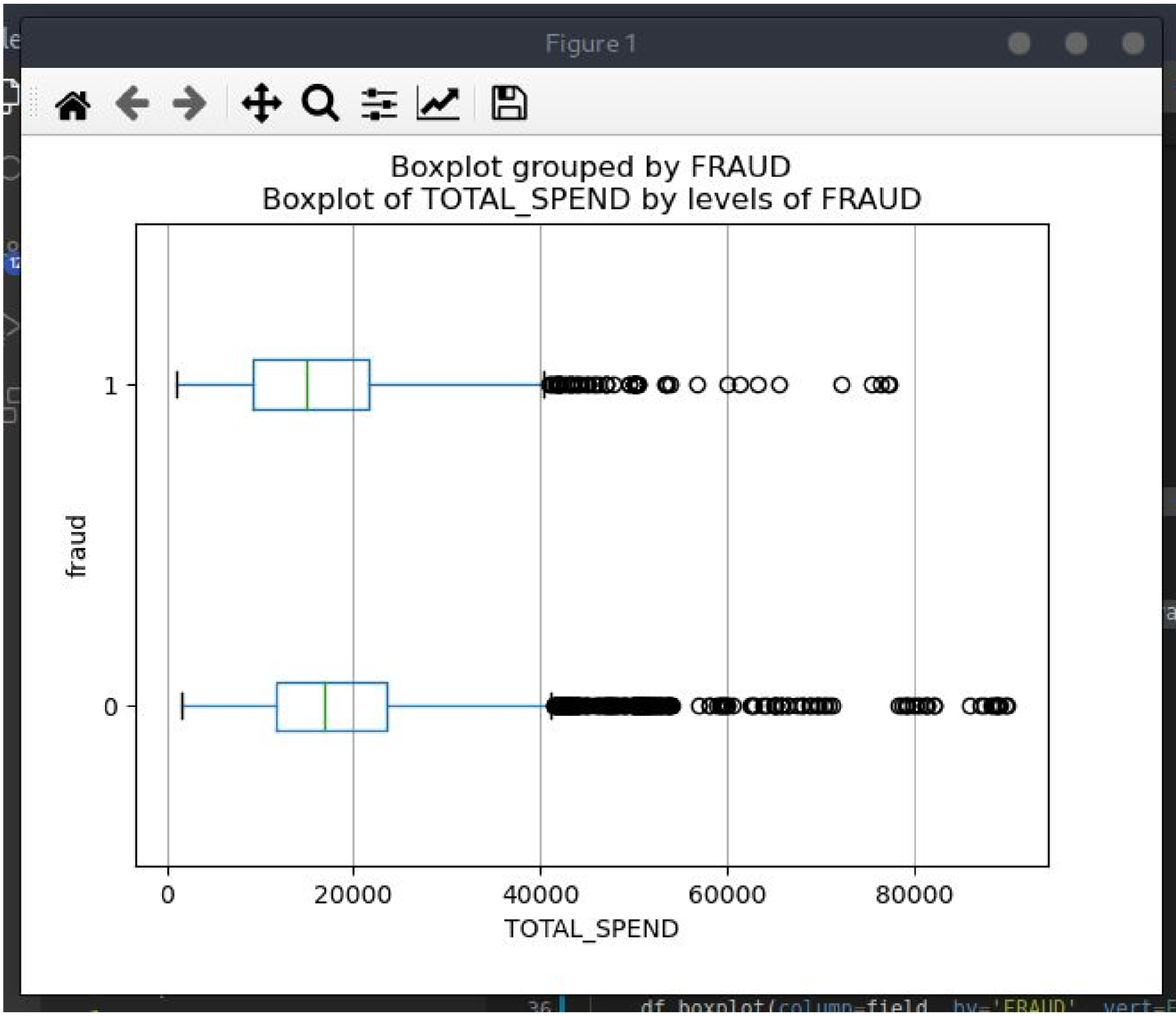


Figure 1
Boxplot grouped by FRAUD
Boxplot of TOTAL_SPEND by levels of FRAUD
fraud
1
0
0
20000
40000
60000
80000
TOTAL_SPEND

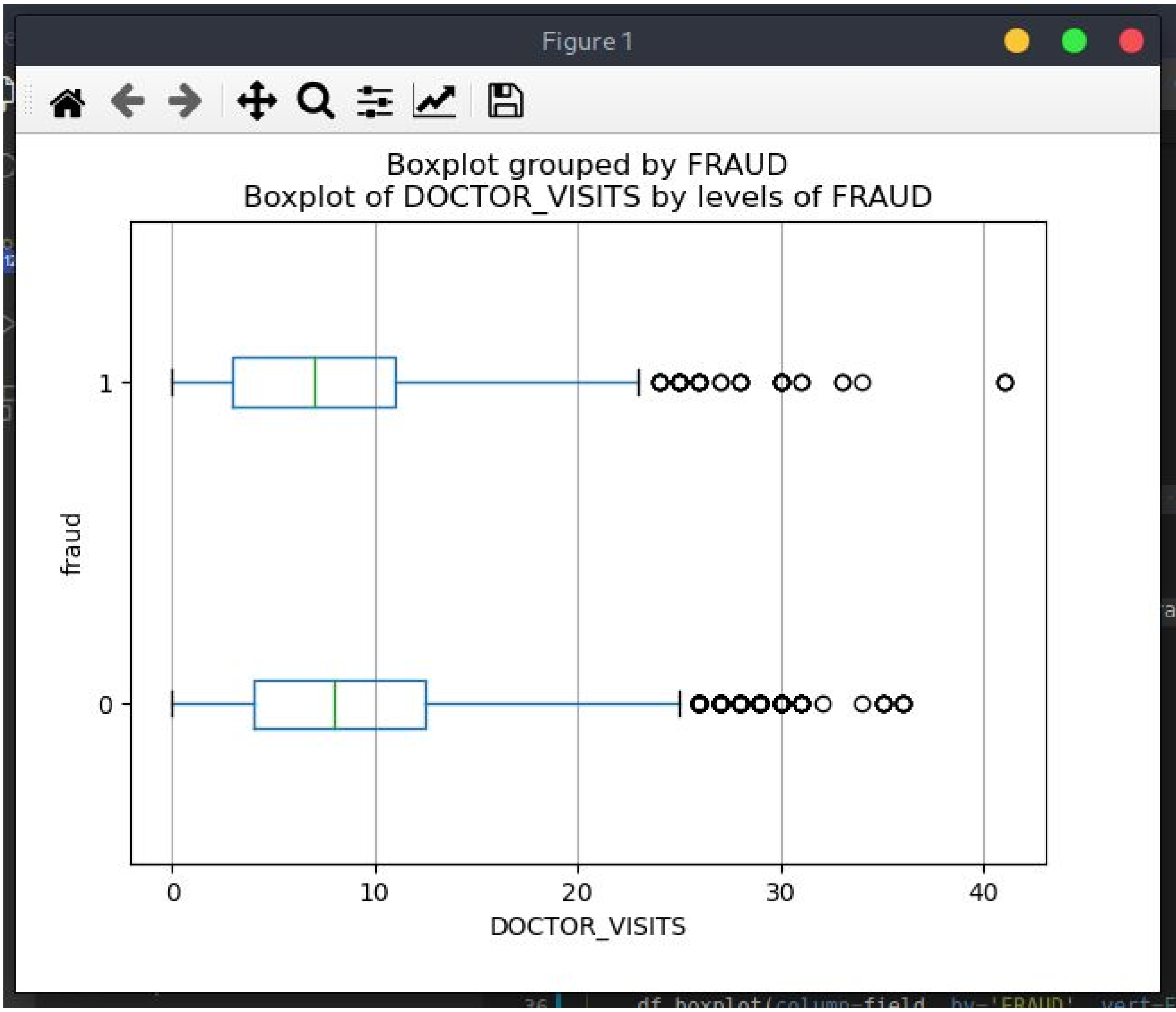


Figure 1
Boxplot grouped by FRAUD
Boxplot of DOCTOR_VISITS by levels of FRAUD
fraud
1
0
0
10
20
30
40
DOCTOR_VISITS

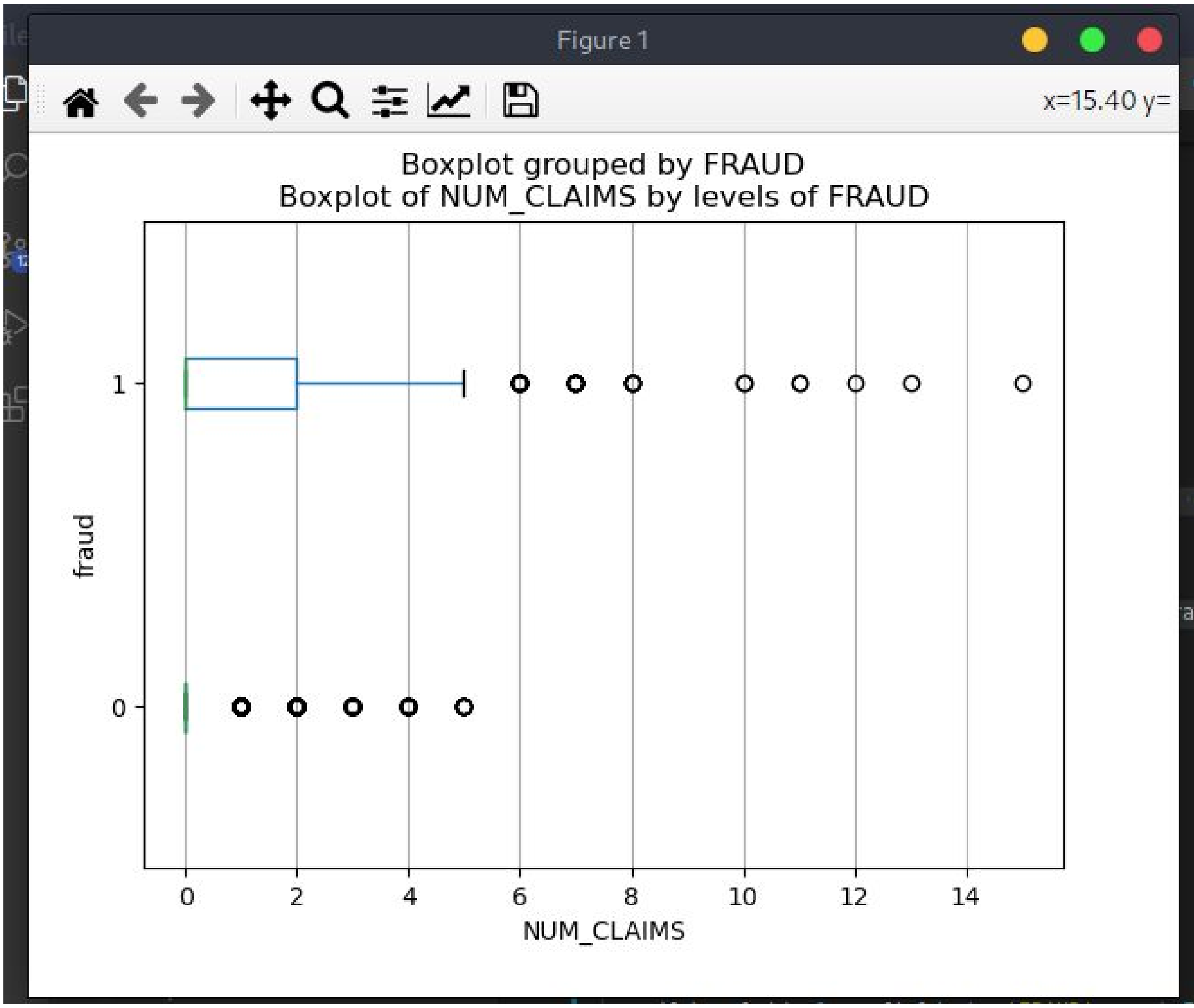


Figure 1
x=15.40 y=
Boxplot grouped by FRAUD
Boxplot of NUM_CLAIMS by levels of FRAUD
fraud
1
0
0
2
4
6
8
10
12
14
NUM_CLAIMS

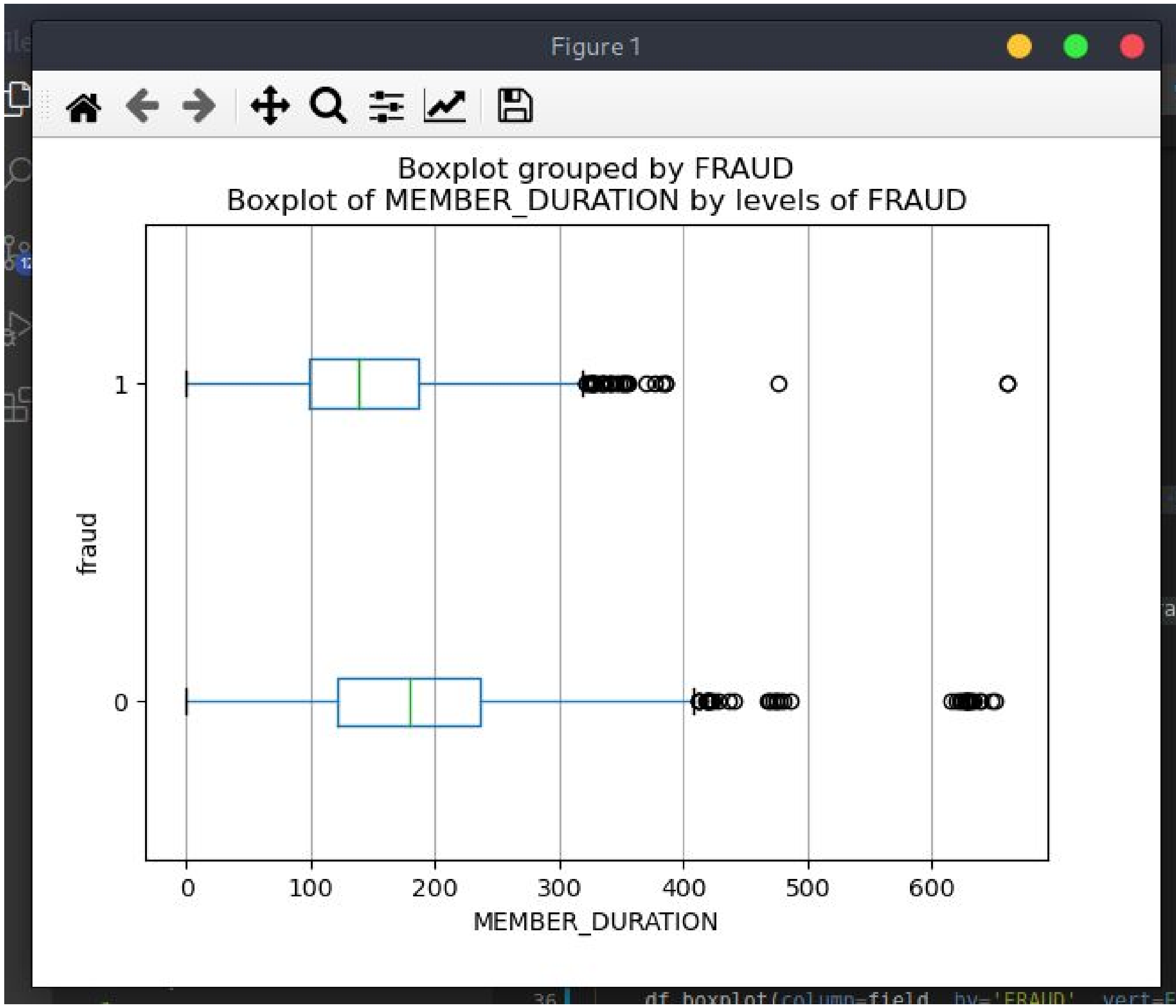


Figure 1
Boxplot grouped by FRAUD
Boxplot of MEMBER_DURATION by levels of FRAUD
fraud
1
0
0
100
200
300
400
500
600
MEMBER_DURATION

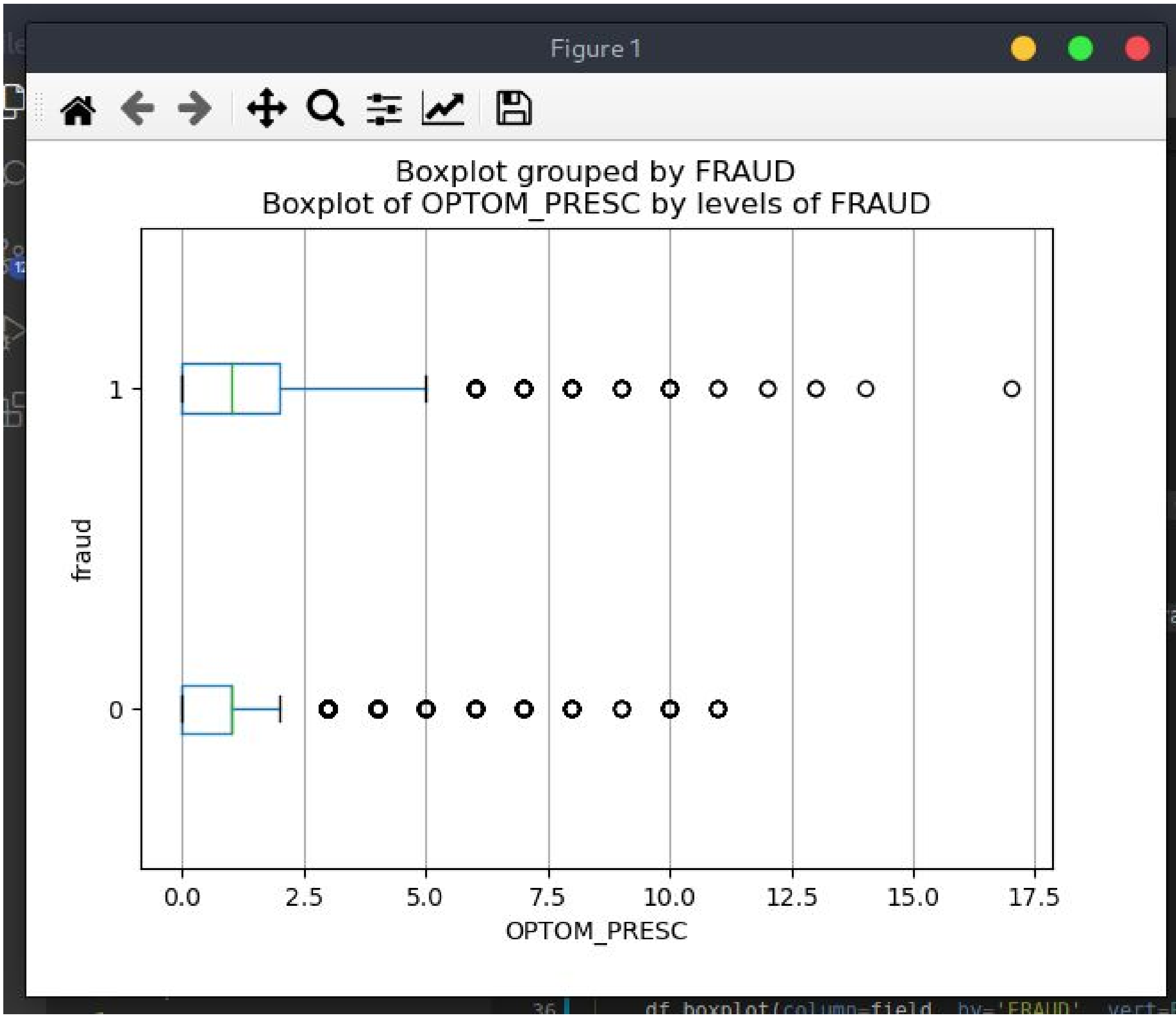
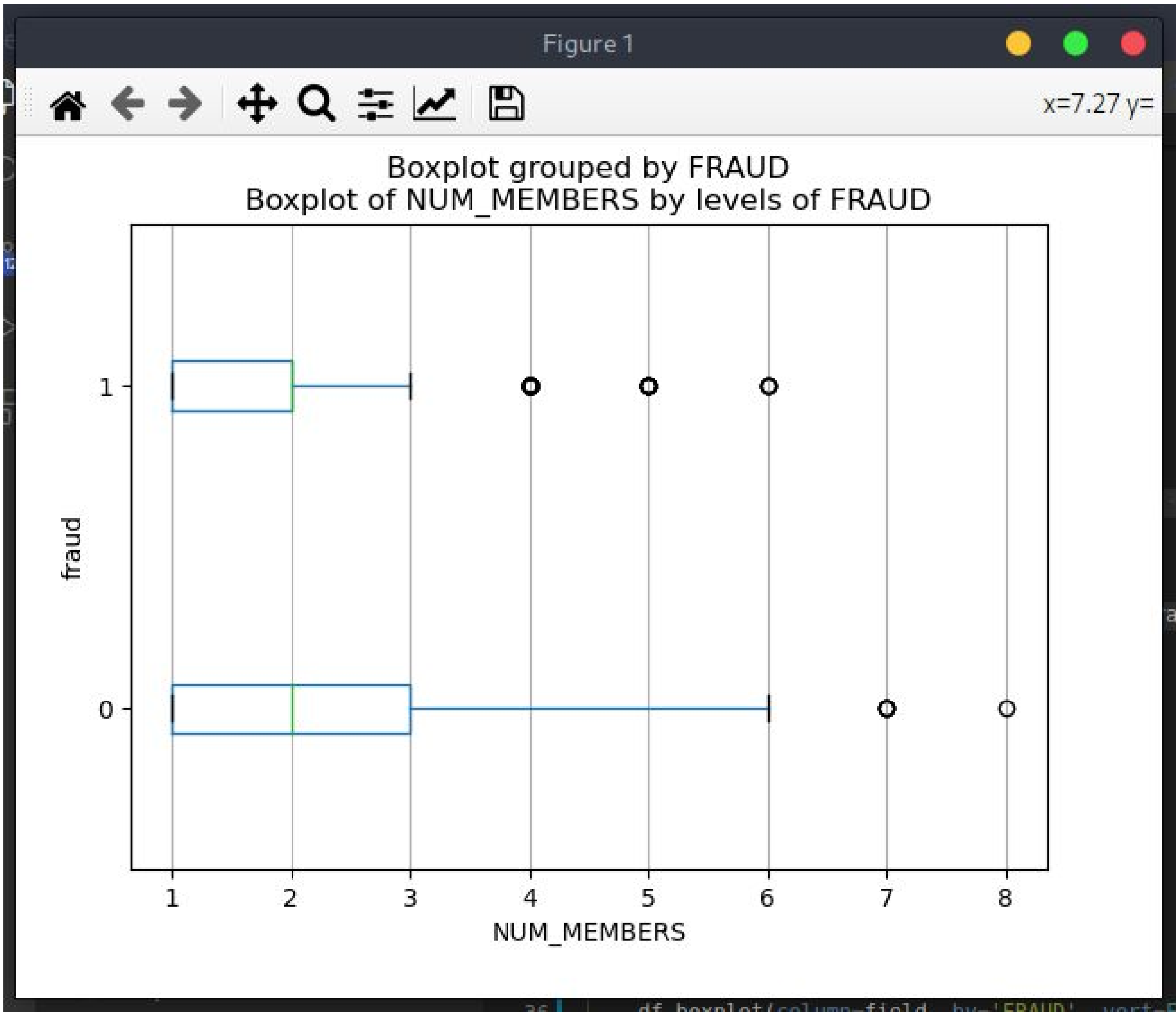
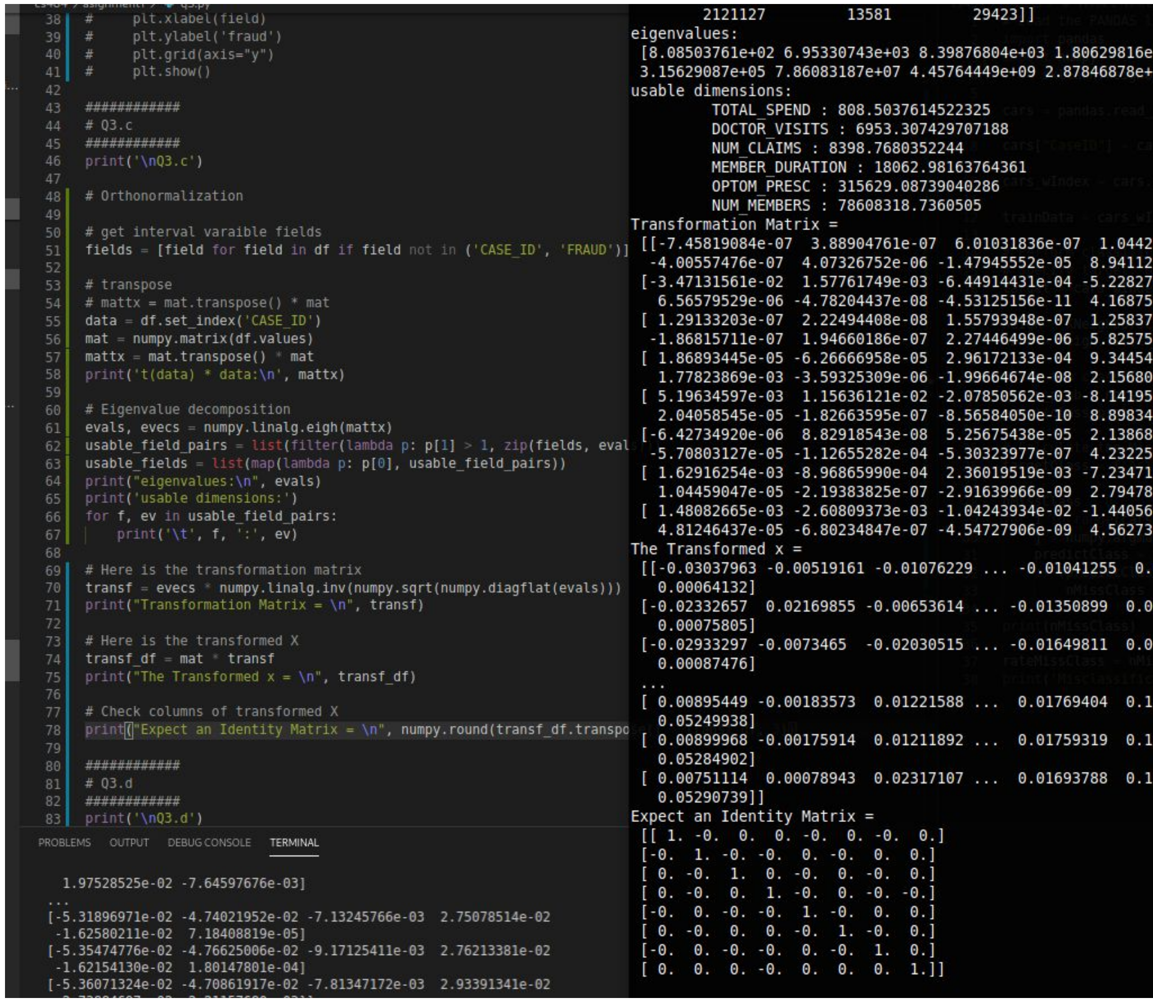


Figure 1
Boxplot grouped by FRAUD
Boxplot of OPTOM_PRESC by levels of FRAUD
fraud
1
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
OPTOM_PRESC



c.

i. There were some math concepts that I haven't yet covered in any of my classes (I'm just now taking linear algebra), however I did find the eigenvectors and eigenvalues using the same methods as used in the examples done in class. As you can see from the eigenvalues, all of them are above 1 so they should all be usable according to the criteria set forth in the assignment. (see 'usable dimensions' in screenshot)

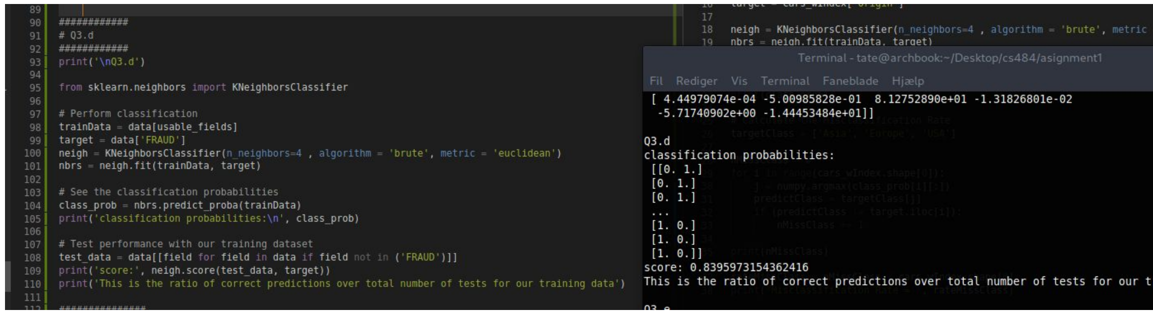


ii. By multiplying the orthonormalized matrix by the original transpose I got an identity matrix as expected. I'm not sure if this is the right test, I tried a few other methods for normalizeing the data but the checks for them were less clear (and probably wrong), so I decided to use the method from the book.

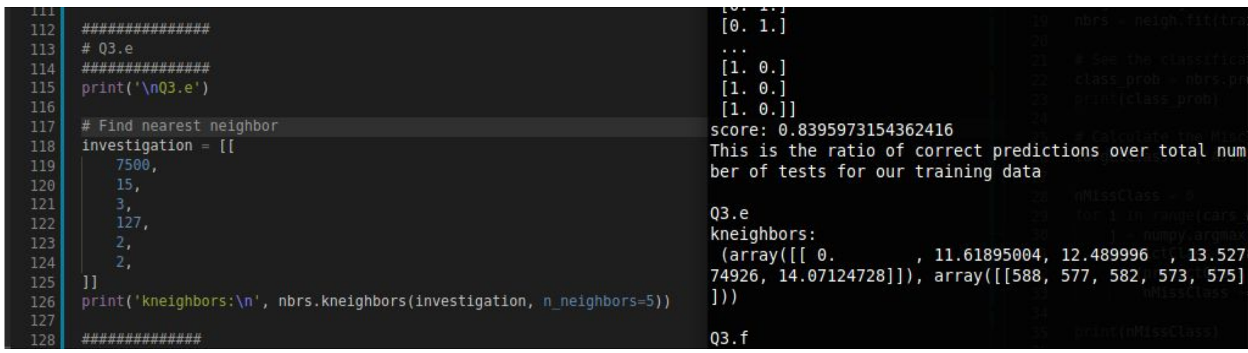
d.

i. As you can see in the image below the score function returned 0.8395973154362416

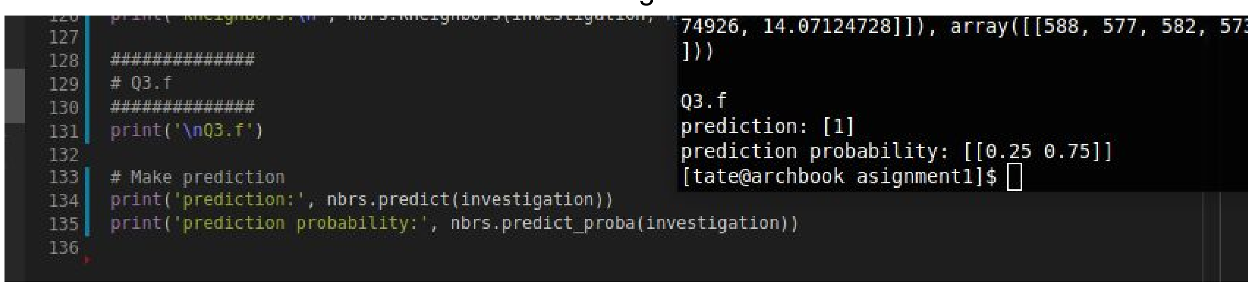
ii. This means that when I ran the classifier on our training data it was able to successfully correctly predict approximately 83.96% of them. However, because only approximately 19.9497% of the input data was fraudulent, our classifier is only about 3.91% better than always predicting fraud=0. However from the perspective of a company with millions of customers this could be worth it.



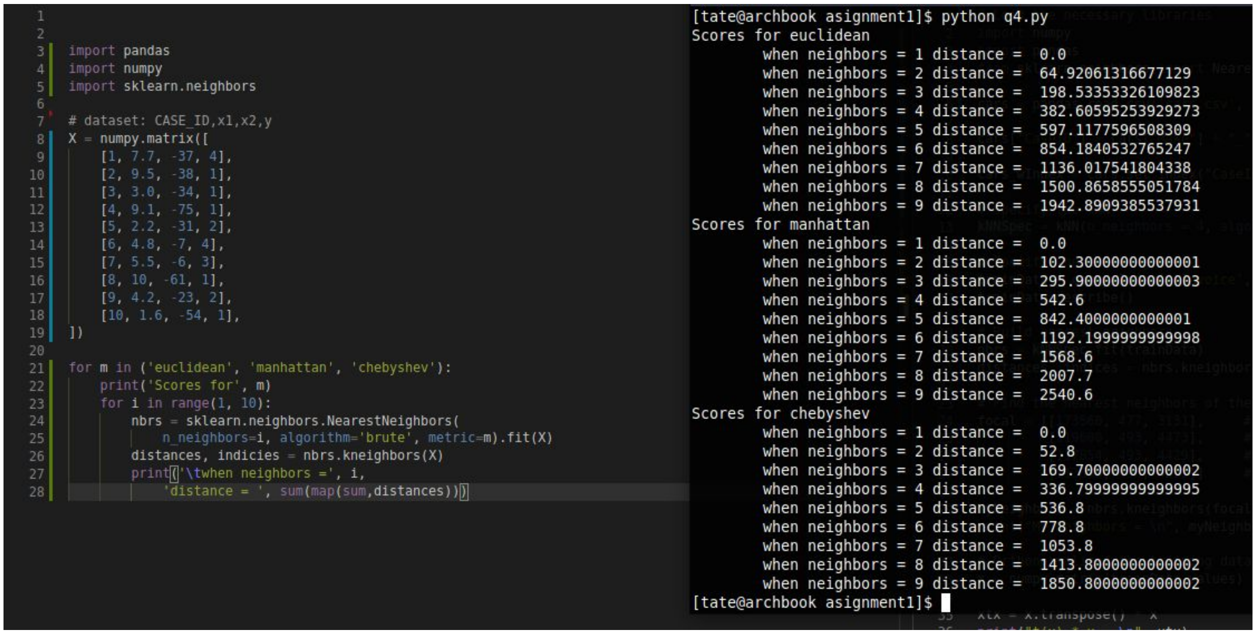
e. The output of kneighbors is as follows:
   - [0, 11.61895004, 12.489996 , 13.52774926, 14.07124728]
   - [588, 577, 582, 573, 575]



f. The knn classifier gave a 25% probability that the case is non-fraudulent and a 75% chance that the case is a fraud. This will result in the case being classified as a fraud as the probability is greater than the average of 20%. Because it was not given whether or not this case is actually fraudulent or not we cannot determine if it's been misclassified until an investigation has been done



4. I’m pretty confident I’m doing this wrong… Book doesn’t do a good job of clarifying how to find the error-rate...



```python


import pandas
import numpy
import sklearn.neighbors

# dataset: CASE_ID,x1,x2,y
X = numpy.matrix([
    [1, 7.7, -37, 4],
    [2, 9.5, -38, 1],
    [3, 3.0, -34, 1],
    [4, 9.1, -75, 1],
    [5, 2.2, -31, 2],
    [6, 4.8, -7, 4],
    [7, 5.5, -6, 3],
    [8, 10, -61, 1],
    [9, 4.2, -23, 2],
    [10, 1.6, -54, 1],
])

for m in ('euclidean', 'manhattan', 'chebyshev'):
    print('Scores for', m)
    for i in range(1, 10):
        nbrs = sklearn.neighbors.NearestNeighbors(
            n_neighbors=i, algorithm='brute', metric=m).fit(X)
        distances, indicies = nbrs.kneighbors(X)
        print('\twhen neighbors =', i,
            'distance = ', sum(map(sum,distances)))
```

```
[tate@archbook asignment1]$ python q4.py
Scores for euclidean
        when neighbors = 1 distance =  0.0
        when neighbors = 2 distance =  64.92061316677129
        when neighbors = 3 distance =  198.53353326109823
        when neighbors = 4 distance =  382.60595253929273
        when neighbors = 5 distance =  597.1177596508309
        when neighbors = 6 distance =  854.1840532765247
        when neighbors = 7 distance =  1136.017541804338
        when neighbors = 8 distance =  1500.8658555051784
        when neighbors = 9 distance =  1942.8909385537931
Scores for manhattan
        when neighbors = 1 distance =  0.0
        when neighbors = 2 distance =  102.30000000000001
        when neighbors = 3 distance =  295.90000000000003
        when neighbors = 4 distance =  542.6
        when neighbors = 5 distance =  842.4000000000001
        when neighbors = 6 distance =  1192.1999999999998
        when neighbors = 7 distance =  1568.6
        when neighbors = 8 distance =  2007.7
        when neighbors = 9 distance =  2540.6
Scores for chebyshev
        when neighbors = 1 distance =  0.0
        when neighbors = 2 distance =  52.8
        when neighbors = 3 distance =  169.70000000000002
        when neighbors = 4 distance =  336.79999999999995
        when neighbors = 5 distance =  536.8
        when neighbors = 6 distance =  778.8
        when neighbors = 7 distance =  1053.8
        when neighbors = 8 distance =  1413.8000000000002
        when neighbors = 9 distance =  1850.8000000000002
[tate@archbook asignment1]$
```

5. Yes