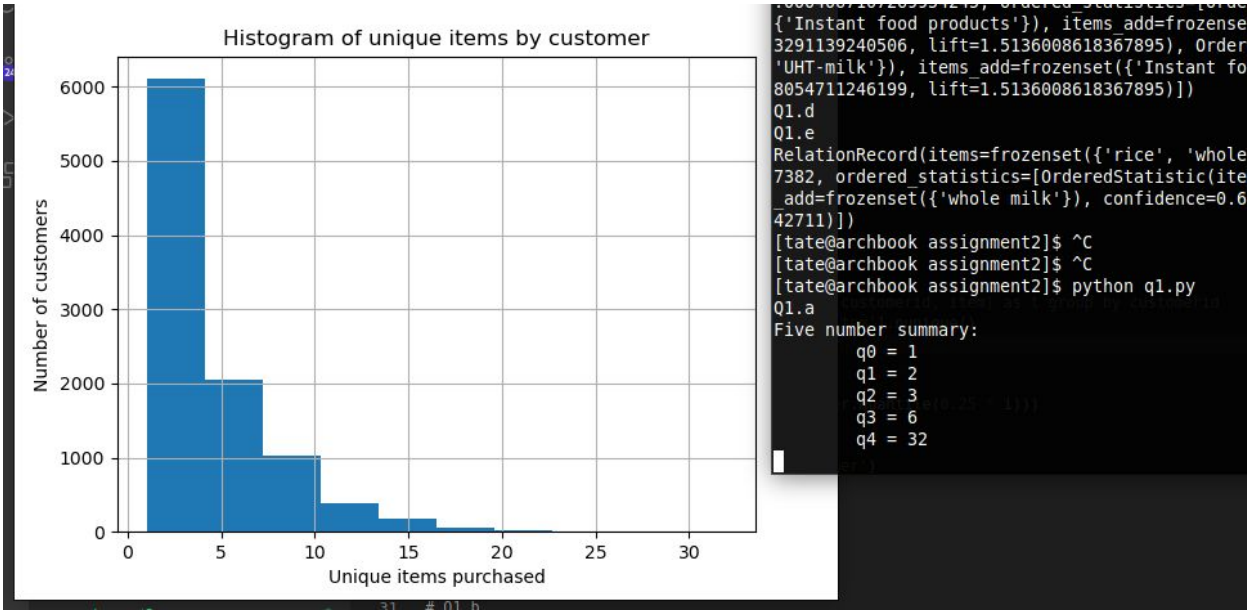
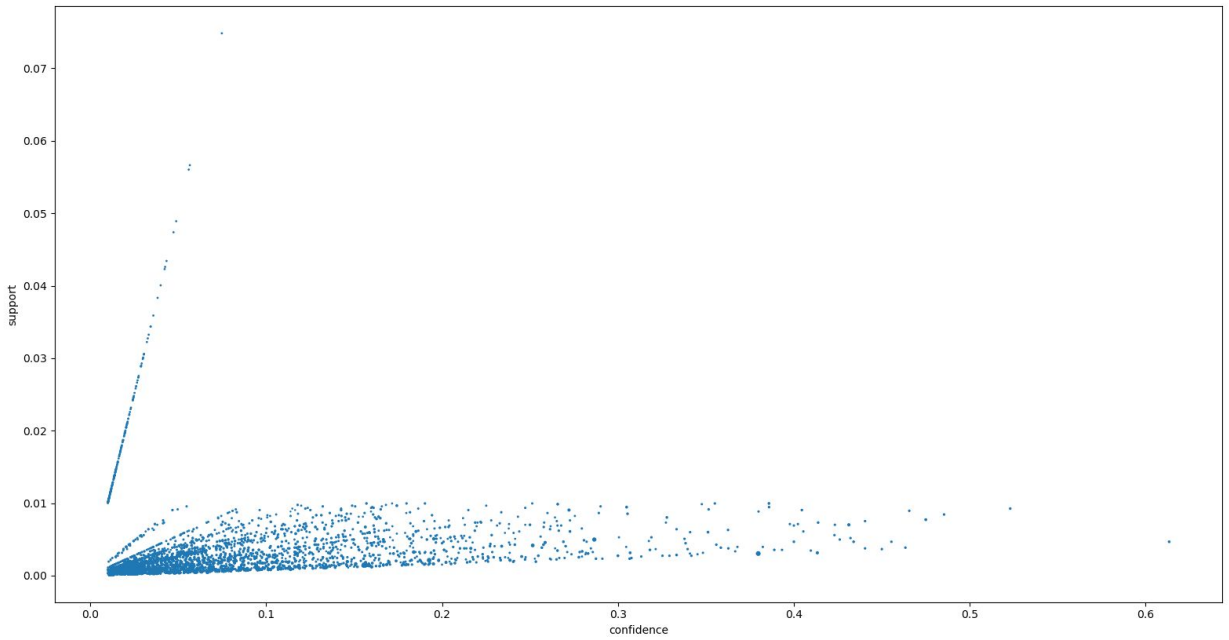
1. See the attached python file q1.py
    a. See histogram and 5 number summary (q0-4 are the 0-100 percentiles)



    b. There are 102 item-sets and the most popular item was purchased by 2513 unique customers
    c.
        i. Using a min support of 0.0001 I found 13177 association rules with a confidence above 1%.
        ii. When I limited the discovery to only look for two items (which makes most sense in this case) I found 4877 rules
        iii. I used (ii) for the rest of the tests as performance was significantly improved

d.



e. Rice -> whole milk

2. As you can see here the three centroids are closest to the tea/coffee, milk, and soda/water

```
import pandas as pd
import sklearn.cluster as cluster


df = pd.read_csv('cafe.csv')
X = pd.DataFrame({'x' : list(df['Frequency'])})
myCluster = cluster.KMeans(n_clusters = 3, random_state = 0).fit(X)

print('Cluster Assignment:', myCluster.labels_)
print('Cluster Centroid 0:', myCluster.cluster_centers_[0])
print('Cluster Centroid 1:', myCluster.cluster_centers_[1])
print('Cluster Centroid 1:', myCluster.cluster_centers_[2])
```

```
Terminal - tate@archbook:~/Des
Fil  Rediger  Vis  Terminal  Faneblade  Hjælp
[tate@archbook assignment2]$ python q2.py
Cluster Assignment: [1 2 2 2 0 0 1]
Cluster Centroid 0: [7.5]
Cluster Centroid 1: [80.]
Cluster Centroid 1: [28.33333333]
[tate@archbook assignment2]$
```

3. Struggling with the proof, I'm taking linear algebra in parallel with this course and haven't taken statistics yet, this problem is beyond my abilities
4. I was unable to understand the notation in the powerpoint, the book, or wikipedia. When I went to TA office hours the TA was unable to help and seemed just as lost as me. Regardless, I managed to find the Silhouette and Davies-Bouldin indecies for the

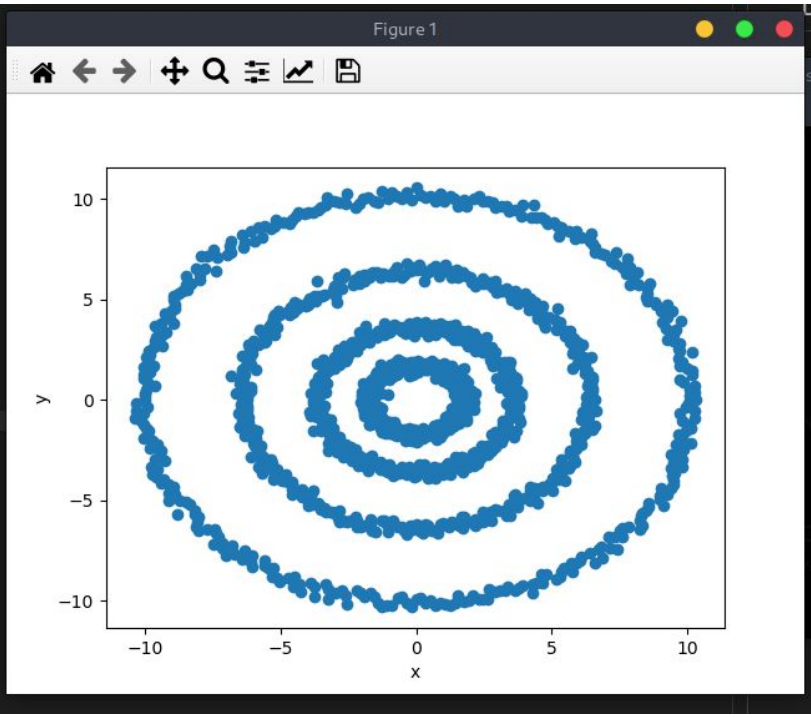
clusters.

```python

import pandas
import sklearn.cluster
import sklearn.metrics

print('Loading clusters...')
X = pandas.DataFrame({ 'x' : [-2, -1, 1, 2, 3, 4, 5, 7, 8] })
cs = sklearn.cluster.KMeans(n_clusters = 2, random_state = 0).fit(X)

# Verfiy cluster assignments are correct
print('\tCluster Assignment:', cs.labels_)

###
# Q4.a
###
print('Q4.a')
# Compute sillouette width
sil = sklearn.metrics.silhouette_score(X, cs.labels_)
print('\t', sil)


###
# Q4.c
###
print('Q4.c')

# Compute davies
dbs = sklearn.metrics.davies_bouldin_score(X, cs.labels_)
print('\t', dbs)
```

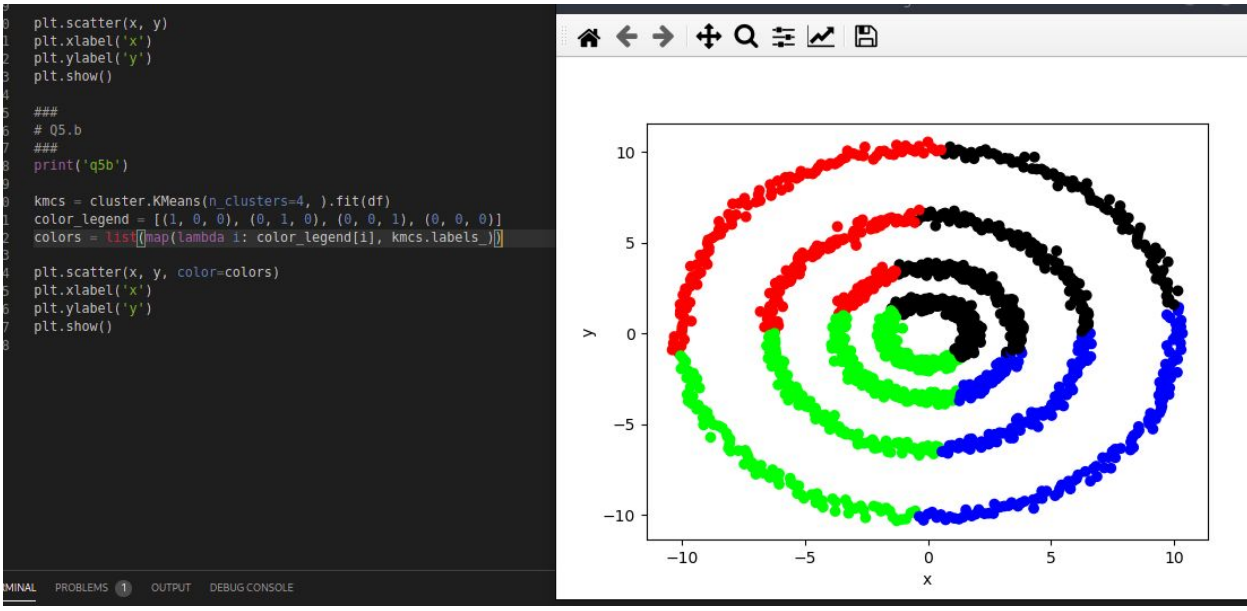
```
Terminal - tate@archbook:~/Desktop/cs
Fil  Rediger  Vis  Terminal  Faneblade  Hjælp
[tate@archbook assignment2]$ python q4.py
Loading clusters...
        Cluster Assignment: [0 0 0 0 0 1 1 1 1]
Q4.a
         0.4847097336067924
Q4.c
         0.5888888888888889
[tate@archbook assignment2]$
```

5. See program file q5.py
    a. It appears there are 4 custers in the dataset

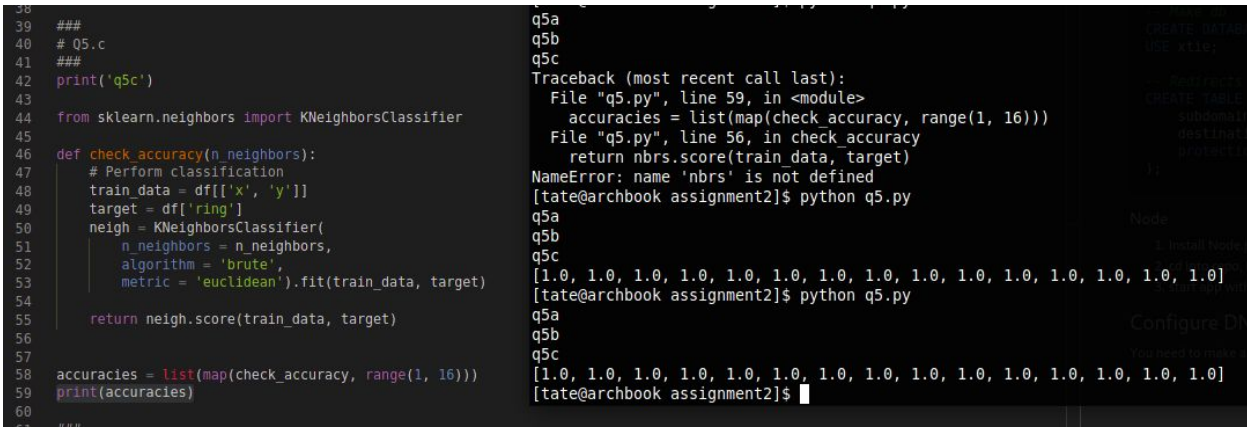
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
import sklearn.cluster as cluster



df = pd.read_csv('FourCircle.csv')

###
# Q5.a
###
print('q5a')

x = df['x']
y = df['y']

plt.scatter(x, y)
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



b. As you can see from the screenshot, the algorithm separates them by quadrant instead of by continuity



c. It seems that we only need one nearest neighbor to achieve needed accuracy. This makes sense in the case of the problem as any point will have a near neighbor on the relevant ring and it



d. And e. Because the math here goes well over what I've covered in coursework, I used the spectral clustering class that's built into sklearn to produce the correct

result

```
61
62  ###
63  # Q5.de
64  ###
65  print('q5e-d')
66
67  # Math go brrr
68  sccs = sklearn.cluster.SpectralClustering(
69      n_neighbors = 1,
70      n_clusters  = 4,
71      n_components= 3,
72  ).fit(df)
73  color_legend = [(1, 0, 0), (0, 1, 0), (0, 0, 1), (0, 0, 0)]
74  colors = list(map(lambda i: color_legend[i], sccs.labels_))
75
76  plt.scatter(x, y, color=colors)
77  plt.xlabel('x')
78  plt.ylabel('y')
79  plt.show()
```