customer_segments

March 1, 2016

1 Creating Customer Segments

In this project you, will analyze a dataset containing annual spending amounts for internal structure, to understand the variation in the different types of customers that a wholesale distributor interacts with.

Instructions:

- Run each code block below by pressing **Shift+Enter**, making sure to implement any steps marked with a TODO.
- Answer each question in the space provided by editing the blocks labeled "Answer:".
- When you are done, submit the completed notebook (.ipynb) with all code blocks executed, as well as a .pdf version (File > Download as).

```
In [70]: # Import libraries: NumPy, pandas, matplotlib
         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns

         # Tell iPython to include plots inline in the notebook
         %matplotlib inline

         # Read dataset
         data = pd.read_csv("wholesale-customers.csv")
         print "Dataset has {} rows, {} columns".format(*data.shape)
         print data.head()  # print the first 5 rows
```

```
Dataset has 440 rows, 6 columns
   Fresh  Milk  Grocery  Frozen  Detergents_Paper  Delicatessen
0  12669  9656     7561     214              2674          1338
1   7057  9810     9568    1762              3293          1776
2   6353  8808     7684    2405              3516          7844
3  13265  1196     4221    6404               507          1788
4  22615  5410     7198    3915              1777          5185
```

```
In [71]: %%javascript
         //Remove the scrolling windows within this notebook
         IPython.OutputArea.auto_scroll_threshold = 9999;
```

```
<IPython.core.display.Javascript object>
```

1.1 Feature Transformation

1) In this section you will be using PCA and ICA to start to understand the structure of the data. Before doing any computations, what do you think will show up in your computations? List one or two ideas for what might show up as the first PCA dimensions, or what type of vectors will show up as ICA dimensions.

Answer:

PCA - I think the first output dimension from PCA will highlight a large variantion in the size of spending and would likely distinguish small shops at one end from supermarkets. With this in mind I think the features of Milk and Grocery spending would inidcate this. - Another feature that might show up is the spending for restarants vs other clients which might be derived from spending on only Fresh and Delicatessen products.

ICA - I think the ICA will highlight strongly different types of clients such as restaurants who don't require frozen goods vs shops and supermarkets which will be based of Milk and Grocery

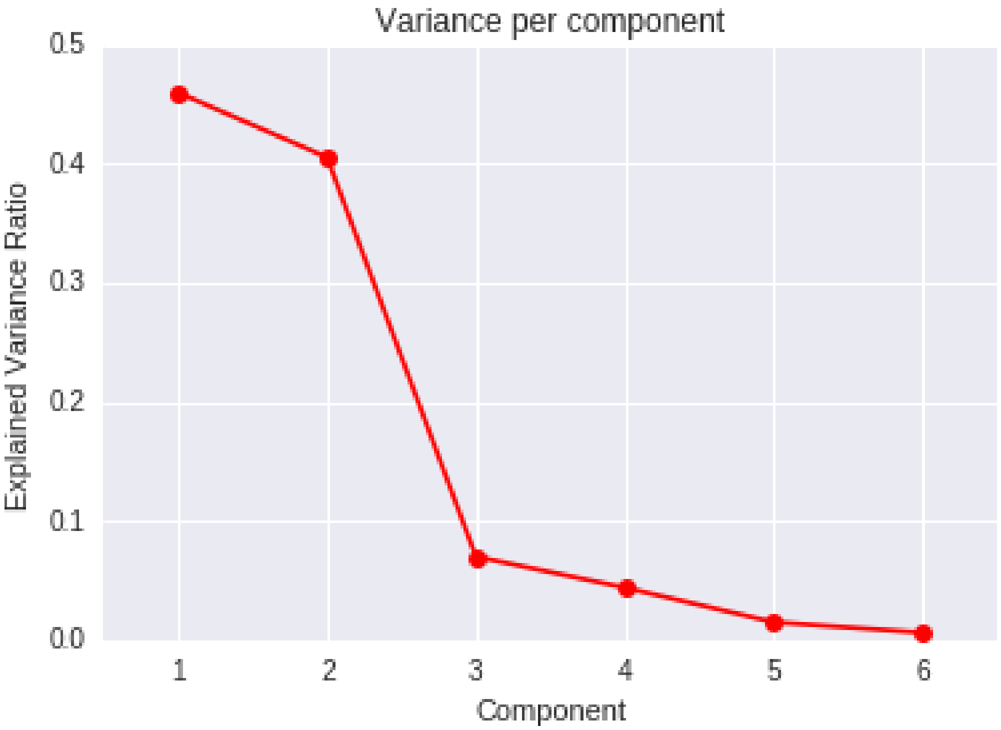
It should be noted that all features are in the same unit($), which gives us variation between clients that spend a lot of money and those that spend little, and due to this I will not scale the data

1.1.1 PCA

```
In [72]: # TODO: Apply PCA with the same number of dimensions as variables in the dataset
         print "\nData shape"
         print data.shape

         from sklearn.decomposition import PCA
         #No need to center the data as PCA will do this for us.
         pca = PCA(n_components=6)
         pca.fit(data)

         # Print the components and the amount of variance in the data contained in each dimension
         np.set_printoptions(precision=3)
         print "\nComponents"
         print data.columns.values
         print pca.components_
         print "\nExplained Variance Ratio"
         print pca.explained_variance_ratio_

         sns.set_style("darkgrid")
         plt.plot(range(1, len(pca.explained_variance_ratio_)+1), pca.explained_variance_ratio_, "-ro")
         plt.xlabel('Component')
         plt.ylabel('Explained Variance Ratio')
         plt.title('Variance per component')
         plt.xlim(xmin=0.5, xmax=6.5)
         plt.show()
```

```
Data shape
(440, 6)

Components
['Fresh' 'Milk' 'Grocery' 'Frozen' 'Detergents_Paper' 'Delicatessen']
[[-0.977 -0.121 -0.062 -0.152  0.007 -0.068]
 [-0.111  0.516  0.765 -0.019  0.365  0.057]
 [-0.179  0.51  -0.276  0.714 -0.204  0.283]
 [-0.042 -0.646  0.375  0.646  0.149 -0.02 ]
 [ 0.016  0.203 -0.16   0.22   0.208 -0.917]
 [-0.016  0.033  0.411 -0.013 -0.871 -0.265]]
```

```
Explained Variance Ratio
[ 0.46   0.405  0.07   0.044  0.015  0.006]
```



2) How quickly does the variance drop off by dimension? If you were to use PCA on this dataset, how many dimensions would you choose for your analysis? Why?

Answer:

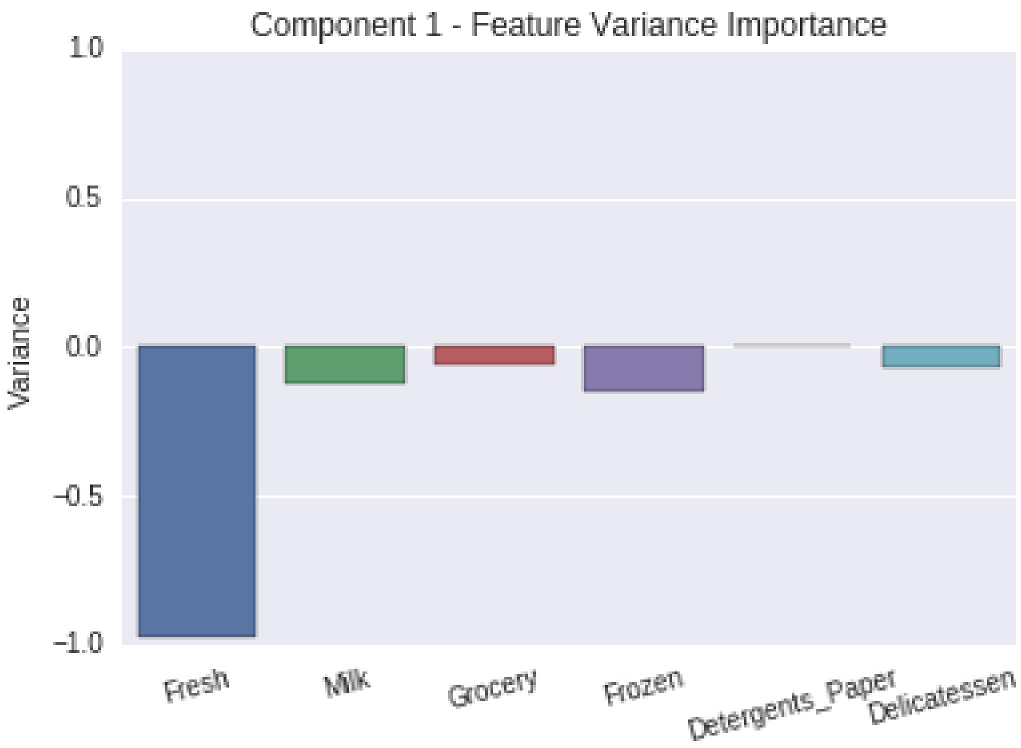
The variance is high for the first and second components at above .4, then drops quickly to the third component with it having a variance of < .1, as shown in the plot "Variance per component".

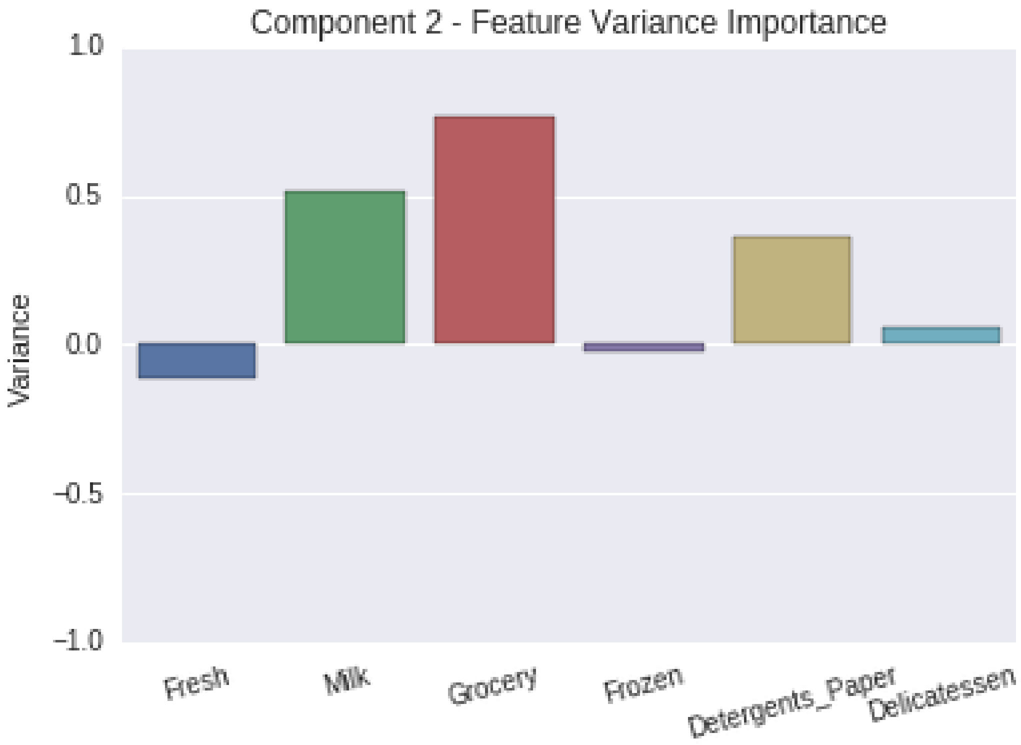
Due to this I would choose to use the first 2 components for analysis.

I would do this as the explained variation ratio is an indication of how much variation is captured in a component, and with components 3,4,5,6 having such a small value they would have little impact on the outcome and could be considered noise.

3) What do the dimensions seem to represent? How can you use this information?

```
In [73]: for i, component in enumerate(pca.components_[:2]):
             plt.ylim([-1.,1.])
             plt.ylabel('Variance')
             plt.title("Component %s - Feature Variance Importance" % (i + 1))
             plt.xticks(rotation=15)
             sns.barplot(data.columns.values, component)
             plt.show()
```


Component 1 - Feature Variance Importance
Variance
1.0
0.5
0.0
−0.5
−1.0
Fresh
Milk
Grocery
Frozen
Detergents_Paper
Delicatessen


Component 2 - Feature Variance Importance
Variance
1.0
0.5
0.0
−0.5
−1.0
Fresh
Milk
Grocery
Frozen
Detergents_Paper
Delicatessen

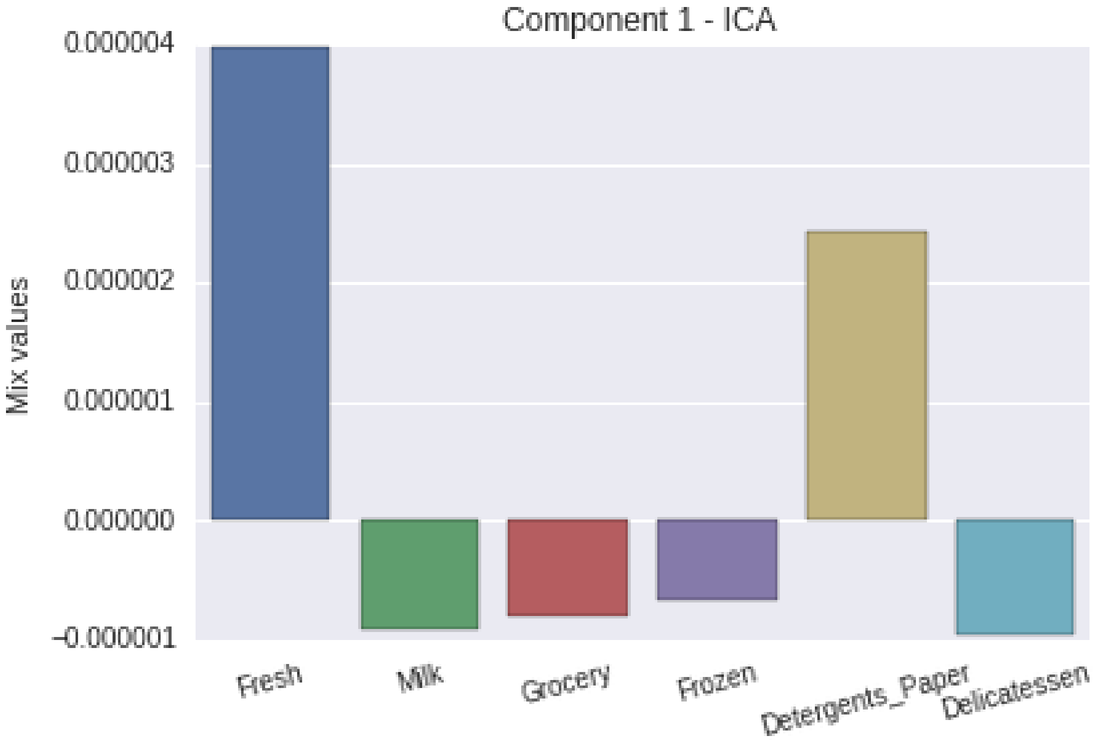
Answer: - The first component shows a large correlation with the Fresh feature. This would indicate that there is large variation of client spending on Fresh food vs spending on all features equally. - The second component shows a large variance between clients that spend more on Grocery, Milk and Detergents than on the features Fresh, Frozem and Delicatessen.

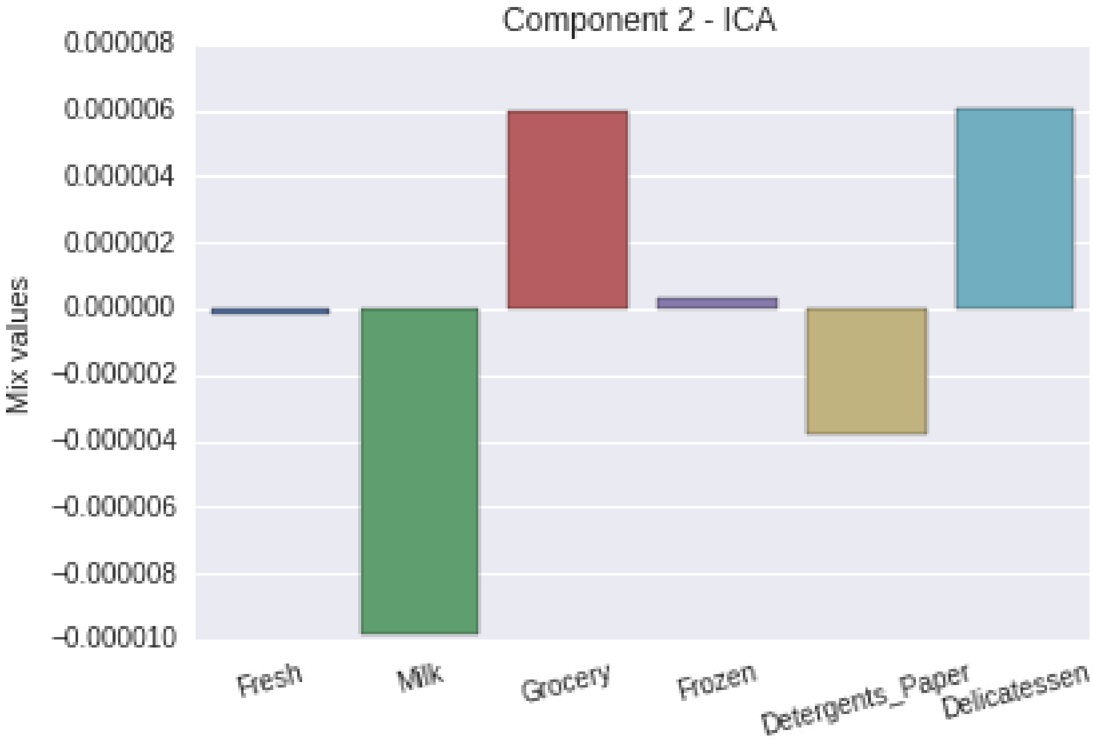
We can use this information to think about possible client types. For example, the large variation in Fresh spending could indicate some clients spend very little on fresh foods, while other clients spend the normal amount.

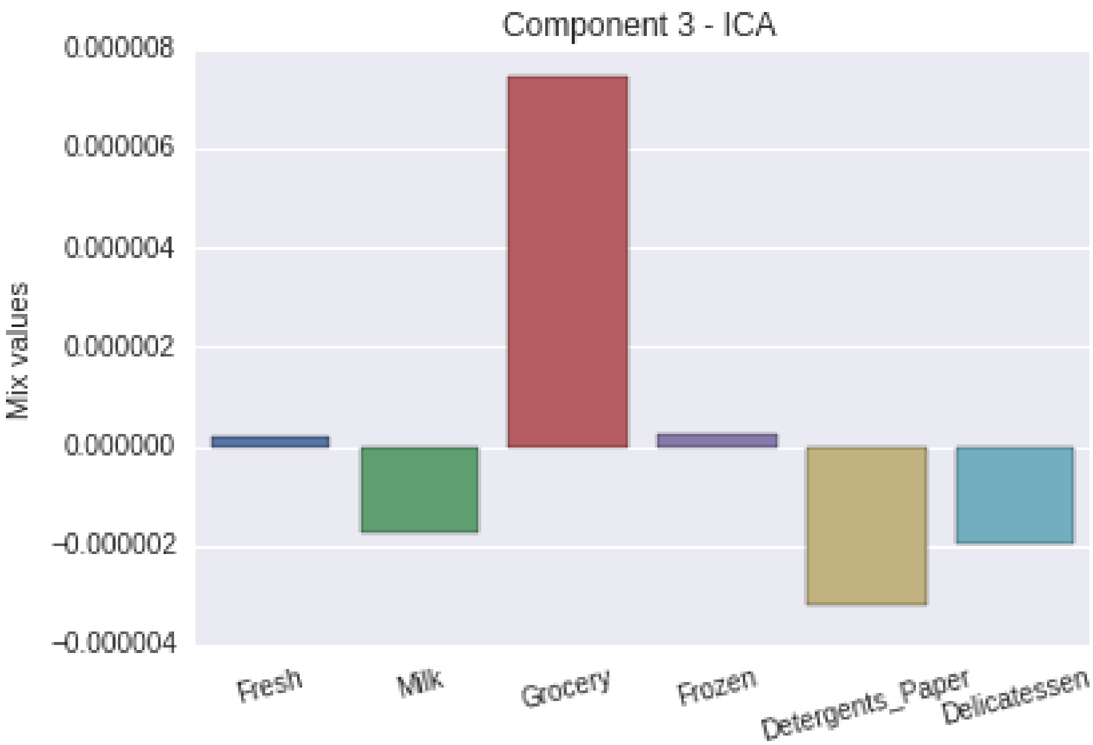
1.1.2 ICA

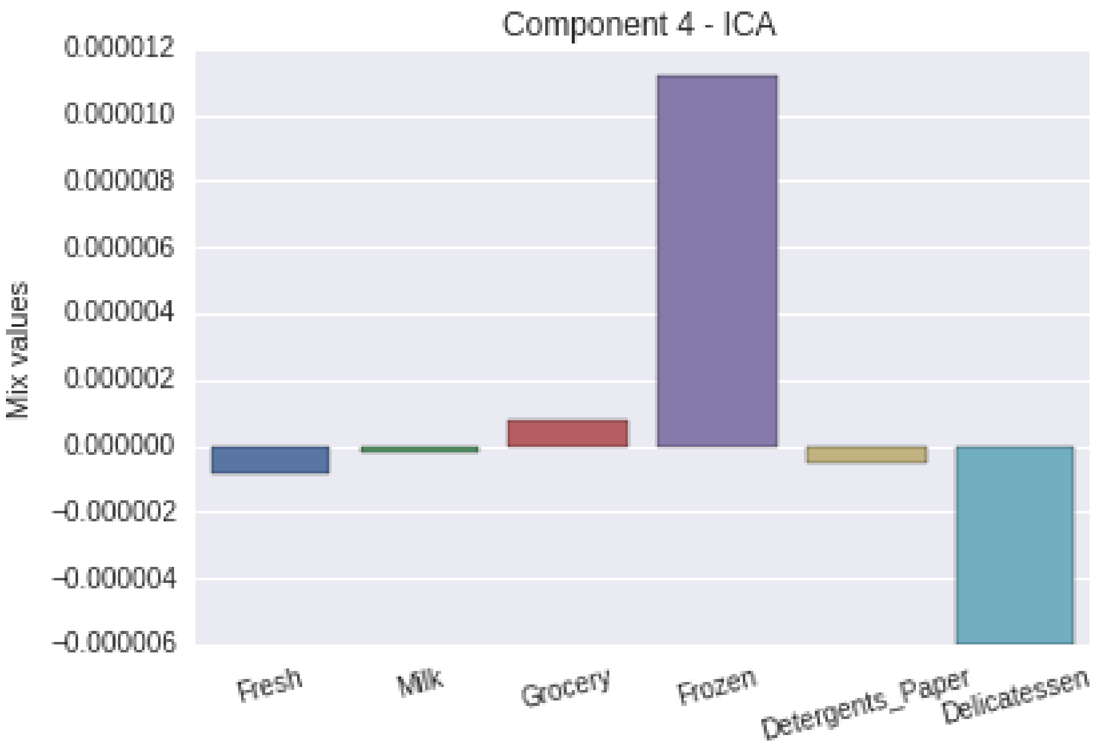
```
In [10]: # TODO: Fit an ICA model to the data
         # Note: Adjust the data to have center at the origin first!

         #Subtract the mean from the data
         #create new dataframe with data adjusted to have the mean centered around 0
         ica_data = pd.DataFrame()
         for i, column_name in enumerate(data.columns.values):
             mean = data[column_name].mean()
             ica_data[column_name] = data[column_name] - mean


         from sklearn.decomposition import FastICA
         ica = FastICA(n_components=6)
         ica.fit(ica_data)

         # Print the independent components
         print ica.components_
```

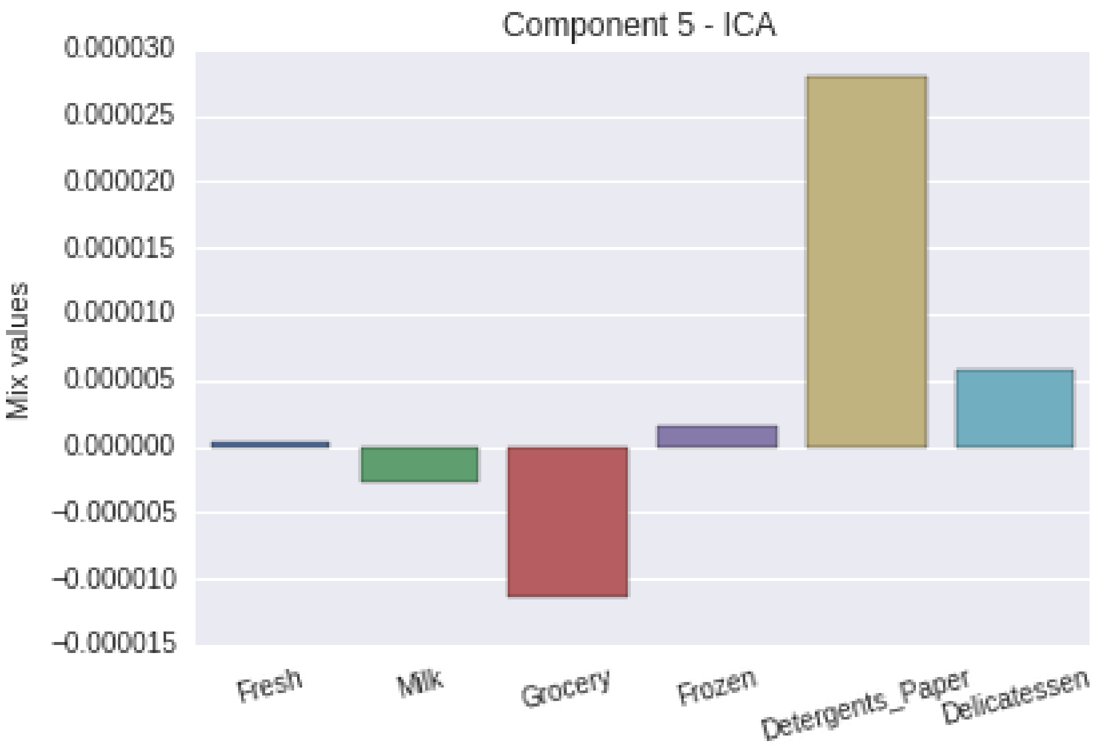
```
[[  3.979e-06  -8.981e-07  -7.856e-07  -6.688e-07   2.426e-06  -9.445e-07]
 [ -1.628e-07  -9.793e-06   5.967e-06   3.104e-07  -3.778e-06   6.021e-06]
 [  1.966e-07  -1.734e-06   7.445e-06   2.738e-07  -3.190e-06  -1.940e-06]
 [ -8.609e-07  -1.688e-07   7.857e-07   1.115e-05  -5.372e-07  -5.966e-06]
 [  2.638e-07  -2.614e-06  -1.137e-05   1.495e-06   2.795e-05   5.711e-06]
 [  3.891e-07   2.042e-07   5.935e-07   5.029e-07  -5.079e-07  -1.807e-05]]
```

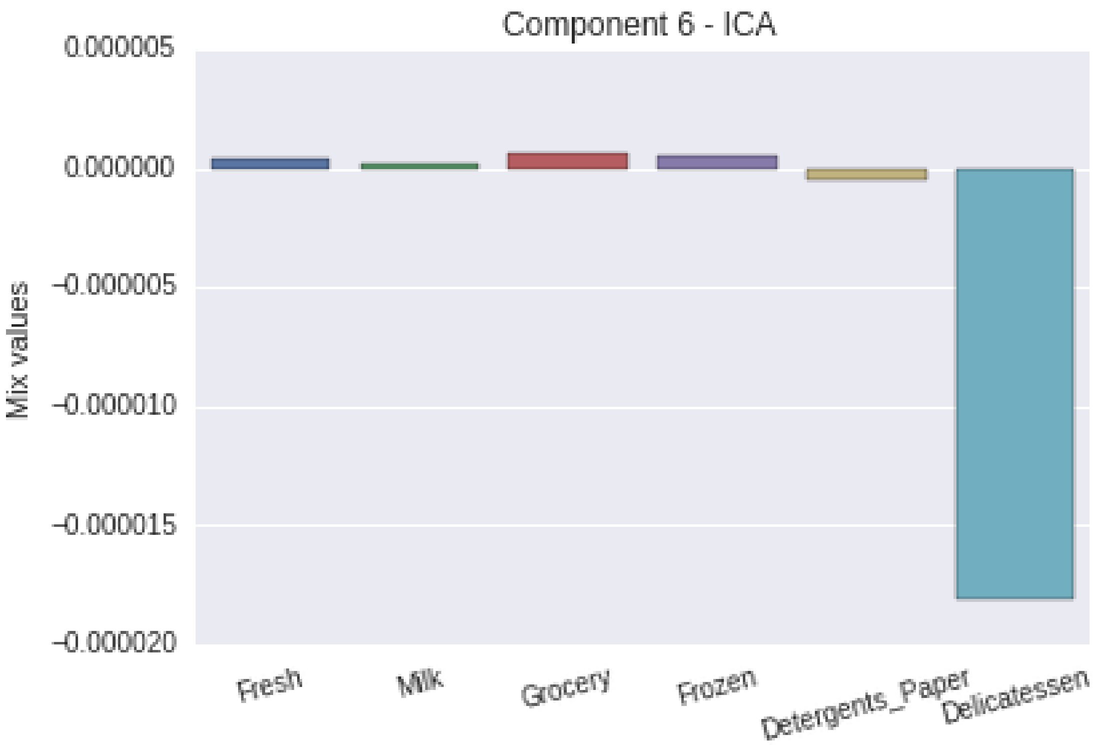

Component 1 - ICA
Mix values
0.000004
0.000003
0.000002
0.000001
0.000000
−0.000001
Fresh
Milk
Grocery
Frozen
Detergents_Paper
Delicatessen


Component 2 - ICA
Mix values
0.000008
0.000006
0.000004
0.000002
0.000000
−0.000002
−0.000004
−0.000006
−0.000008
−0.000010
Fresh
Milk
Grocery
Frozen
Detergents_Paper
Delicatessen


Component 3 - ICA
Mix values
0.000008
0.000006
0.000004
0.000002
0.000000
−0.000002
−0.000004
Fresh
Milk
Grocery
Frozen
Detergents_Paper
Delicatessen


Component 4 - ICA
Mix values
0.000012
0.000010
0.000008
0.000006
0.000004
0.000002
0.000000
−0.000002
−0.000004
−0.000006
Fresh
Milk
Grocery
Frozen
Detergents_Paper
Delicatessen


Component 5 - ICA
Mix values
0.000030
0.000025
0.000020
0.000015
0.000010
0.000005
0.000000
−0.000005
−0.000010
−0.000015
Fresh
Milk
Grocery
Frozen
Detergents_Paper
Delicatessen


Component 6 - ICA
Mix values
0.000005
0.000000
−0.000005
−0.000010
−0.000015
−0.000020
Fresh
Milk
Grocery
Frozen
Detergents_Paper
Delicatessen

4) For each vector in the ICA decomposition, write a sentence or two explaining what sort of object or property it corresponds to. What could these components be used for?

Answer: The components shown in the graphs above show the spending weights of the independent components.

- **ICA Component 1** Clients that spend on Fresh and Detergents and not on Milk, Grocery, Frozen and Delicatessen
- **ICA Component 2** Clients that spend on Grocery and Delicatessen and not on Milk and Detergents
- **ICA Component 3** Clients that spend largely on Grocery and not on Milk, Detergents and Delicatessen
- **ICA Component 4** Clients that spend largely on Frozen and not on Delicatessen
- **ICA Component 5** Clients that spend largely on Detergents, a little on Delicatessen and not on Grocery
- **ICA Component 6** Clients that do not spend on Delicatessen

These components can be used to identify different client types

1.2 Clustering

In this section you will choose either K Means clustering or Gaussian Mixed Models clustering, which implements expectation-maximization. Then you will sample elements from the clusters to understand their significance.

1.2.1 Choose a Cluster Type

5) What are the advantages of using K Means clustering or Gaussian Mixture Models?

Answer:

K Means

- Fast
- Simple to calculate
- At each iteration a point is classified with exactly one cluster
- Creates straight line boundaries
- A cluster cannot cross another cluster

GMM

- Fast
- Simple to calculate
- At each iteration a point has a value indicating how strongly it is within a cluster
- Creates elliptical boundaries
- A cluster can cross another cluster

I have chosen to use GMM as I believe we will get tighter clustering groups

6) Below is some starter code to help you visualize some cluster data. The visualization is based on this demo from the sklearn documentation.

```
In [97]: # Import clustering modules
         from sklearn.cluster import KMeans
         from sklearn.mixture import GMM
```

```
In [98]: # TODO: First we reduce the data to two dimensions using PCA to capture variation
         pca2 = PCA(n_components=2)
         reduced_data = pca2.fit_transform(data)
         print reduced_data[:10]  # print upto 10 elements
```

```
[[  -650.022   1585.519]
 [  4426.805   4042.452]
 [  4841.999   2578.762]
 [  -990.346  -6279.806]
 [-10657.999  -2159.726]
 [  2765.962   -959.871]
 [   715.551  -2013.002]
 [  4474.584   1429.497]
 [  6712.095  -2205.909]
 [  4823.634  13480.559]]
```

```
In [136]: # TODO: Implement your clustering algorithm here, and fit it to the reduced data for visualiz
          # The visualizer below assumes your clustering object is named 'clusters'
          #cl = GMM(n_components=4)
          cl = GMM(n_components=6)
          cl.fit(reduced_data)

          clusters = cl
          print clusters
```

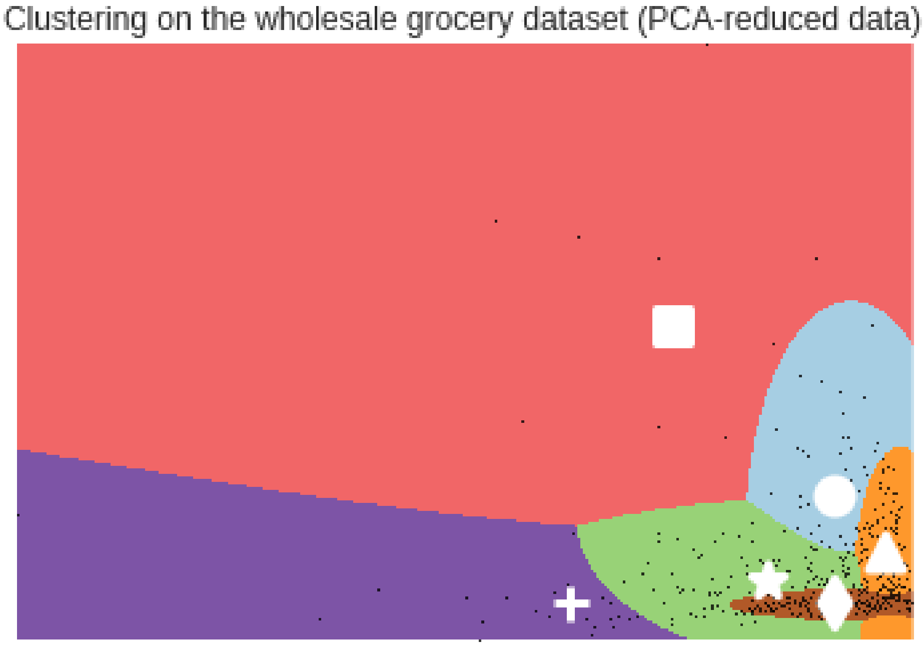
```
GMM(covariance_type='diag', init_params='wmc', min_covar=0.001,
  n_components=6, n_init=1, n_iter=100, params='wmc', random_state=None,
  thresh=None, tol=0.001, verbose=0)
```

```
In [129]: # Plot the decision boundary by building a mesh grid to populate a graph.
          x_min, x_max = reduced_data[:, 0].min() - 1, reduced_data[:, 0].max() + 1
          y_min, y_max = reduced_data[:, 1].min() - 1, reduced_data[:, 1].max() + 1
          hx = (x_max-x_min)/1000.
          hy = (y_max-y_min)/1000.
          xx, yy = np.meshgrid(np.arange(x_min, x_max, hx), np.arange(y_min, y_max, hy))

          # Obtain labels for each point in mesh. Use last trained model.
          Z = clusters.predict(np.c_[xx.ravel(), yy.ravel()])
```

```
In [130]: # TODO: Find the centroids for KMeans or the cluster means for GMM

          centroids = clusters.means_
          print centroids
```

```
[[  3174.314  13286.269]
 [ -5720.047  -2795.884]
 [-18420.863  45891.86 ]
 [  9534.819   2876.925]
 [-31564.062  -6796.88 ]
 [  2921.555  -7173.852]]
```

```
In [135]: # Put the result into a color plot
          Z = Z.reshape(xx.shape)
          plt.figure(1)
          plt.clf()
          plt.imshow(Z, interpolation='nearest',
                     extent=(xx.min(), xx.max(), yy.min(), yy.max()),
                     cmap=plt.cm.Paired,
                     aspect='auto', origin='lower')

          plt.plot(reduced_data[:, 0], reduced_data[:, 1], 'k.', markersize=2)
          for i, marker in enumerate("o*s^+d"[:len(centroids)]):
              dot = marker
              plt.scatter(centroids[:, 0][i], centroids[:, 1][i], marker=dot, s=169, linewidths=3, colo
          #plt.scatter(centroids[:, 0], centroids[:, 1], marker='$a$', s=169, linewidths=3, color='w',
          plt.title('Clustering on the wholesale grocery dataset (PCA-reduced data)')
          plt.xlim(x_min, x_max)
          plt.ylim(y_min, y_max)
          plt.xticks(())
          plt.yticks(())
          plt.show()
```



7) What are the central objects in each cluster? Describe them as customers.

Answer: From the PCA analysis work in the PCA section above above, we have descriptions for the x and y axis.

x-axis

The x axis shows the ratio of client spending on Fresh vs spending on the other features. Far left indicates less spending on Fresh than other features and the right indicates an equal spending on Fresh and other features.

y-axis

The y axis shows spending on Milk, Grocery and Detergents vs spending on Fresh, Frozen and Delicatessen. The top of the graph indicates increased spending on Milk, Grocery and Detergents, whereas the bottom indicates even spending.

Client Groups

- The square in the graph represents large spending on all items, I would classify these as large supermarkets

- The plus represents large spending on all items except fresh. These maybe large warehouse shops such as walmart
- The star represents medium spending on all items except Fresh. These could be petrol stations
- The circle represents medium spending on all products, these could be small supermarkets
- The triangle represents small and equal spending on all items. These could be corner stores.
- The diamond represents small spending on all items except fresh. These could be any number of small businesses not related to food selling.

1.2.2 Conclusions

** 8)** Which of these techniques did you feel gave you the most insight into the data?

Answer: ICA gave me a good understanding about the different vectors that underly the clients. For example, spending on Fresh and Detergents is an independent group (maybe restaurants?), where PCA seems to bundle all these independant components together.

9) How would you use that technique to help the company design new experiments?

Answer: I would use clustering to identify groups based on ICA instead of PCA, and then use ICA to further break down these groups into sub categories. With this information I could market products directly to these groups.

10) How would you use that data to help you predict future customer needs?

Answer: We could use this information to create a supervised classifier, which could show us how a client is changing over time, and what their new needs might be as they change groups.