Wholesale Customer Segmentation Analysis

Hierarchical Cluster Analysis

Sean O’Malley

Ingest, EDA and Data Manipulation

```
sales <- read.csv("/Users/SeanOMalley1/Desktop/MSDS\ 680\ ML/Wholesale\ customers\ data.csv")
```

Everything is numeric and of high data quality, so we can now move forward with the analysis without too much of a data manipulation headache.

I will perform the analysis with the z-score standardization of the data.

```
summary(sales)
```

```
##     Channel          Region          Fresh             Milk      
##  Min.   :1.000   Min.   :1.000   Min.   :     3   Min.   :   55  
##  1st Qu.:1.000   1st Qu.:2.000   1st Qu.:  3128   1st Qu.: 1533  
##  Median :1.000   Median :3.000   Median :  8504   Median : 3627  
##  Mean   :1.323   Mean   :2.543   Mean   : 12000   Mean   : 5796  
##  3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.: 16934   3rd Qu.: 7190  
##  Max.   :2.000   Max.   :3.000   Max.   :112151   Max.   :73498  
##     Grocery          Frozen        Detergents_Paper    Delicassen     
##  Min.   :    3   Min.   :   25.0   Min.   :    3.0   Min.   :    3.0  
##  1st Qu.: 2153   1st Qu.:  742.2   1st Qu.:  256.8   1st Qu.:  408.2  
##  Median : 4756   Median : 1526.0   Median :  816.5   Median :  965.5  
##  Mean   : 7951   Mean   : 3071.9   Mean   : 2881.5   Mean   : 1524.9  
##  3rd Qu.:10656   3rd Qu.: 3554.2   3rd Qu.: 3922.0   3rd Qu.: 1820.2  
##  Max.   :92780   Max.   :60869.0   Max.   :40827.0   Max.   :47943.0  
```

```
sales <- na.omit(sales)

z_sales <- as.data.frame(mapply(scale, sales))

summary(z_sales)
```

```
##     Channel           Region          Fresh            Milk        
##  Min.   :-0.6895   Min.   :-1.9931   Min.   :-0.9486   Min.   :-0.7779  
##  1st Qu.:-0.6895   1st Qu.:-0.7015   1st Qu.:-0.7015   1st Qu.:-0.5776  
##  Median :-0.6895   Median : 0.5900   Median :-0.2764   Median :-0.2939  
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000  
##  3rd Qu.: 1.4470   3rd Qu.: 0.5900   3rd Qu.: 0.3901   3rd Qu.: 0.1889  
##  Max.   : 1.4470   Max.   : 0.5900   Max.   : 7.9187   Max.   : 9.1732  
##     Grocery            Frozen         Detergents_Paper    Delicassen     
##  Min.   :-0.8364   Min.   :-0.62763   Min.   :-0.6037   Min.   :-0.5396  
##  1st Qu.:-0.6101   1st Qu.:-0.47988   1st Qu.:-0.5505   1st Qu.:-0.3960  
##  Median :-0.3363   Median :-0.31844   Median :-0.4331   Median :-0.1984  
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000  
##  3rd Qu.: 0.2846   3rd Qu.: 0.09935   3rd Qu.: 0.2182   3rd Qu.: 0.1047  
##  Max.   : 8.9264   Max.   :11.90545   Max.   : 7.9586   Max.   :16.4597  
```

```
z_sales2 <- z_sales[3:8]
```

Hierarchical Cluster Analysis

The overarching idea of a hierarchical clustering algorithm is to build a tree of data that successfully merges similar groups of points. Unlike k-means, hierarchical clustering only requires a measure of similarity between groups of data points.

Given a set of N items to be clustered, and a N*N distance, or similarity matrix, start by assigning each item to its own cluster. Thus, if you have N items, you can now have N clusters, each containing just one item. You then let the distances between the clusters equal the distances between the items they contain. Next, you find the closest pair of clusters, and merge them into a single cluster, that you now have one less cluster.

Then, compute the distances between the new cluster and each of the old clusters, repeating these steps until all items are clustered into a single cluster size of N. This looping process of sorts can be repeated via various methodologies, which I will explain further in the next question.

Additional HCA methodologies and distance measurements to consider

There are two approaches when considering hierarchical clusters:

- **Agglomerative Hierarchical Clustering** : This is a bottom up approach, where each observation starts in its own cluster. We can then compute the similarity between each cluster and then merge the two most similar ones at each iteration until there is only one cluster left.
- **Divisive Hierarchical Clustering** : This is a top down approach, where all observations start in one cluster, and then we split the cluster into the two least dissimilar clusters recursively until there is one for each observation.

Now in consideration of the measuring of the distance methodology between clusters, there are 4 common functions used for the measure of similarity:

- **Single Linkage** : Shortest distance between two points in each cluster.
- **Complete Linkage** : Longest distance between two points in each cluster.
- **Average Linkage** : Average distance between two points in each cluster.
- **Ward Method** : Sum of the squared distance from each point to the mean of the merged clusters.

Agglomerative Hierarchical Clustering

Wards minimum variance to perform agglomerative HCS using Euclidian distance

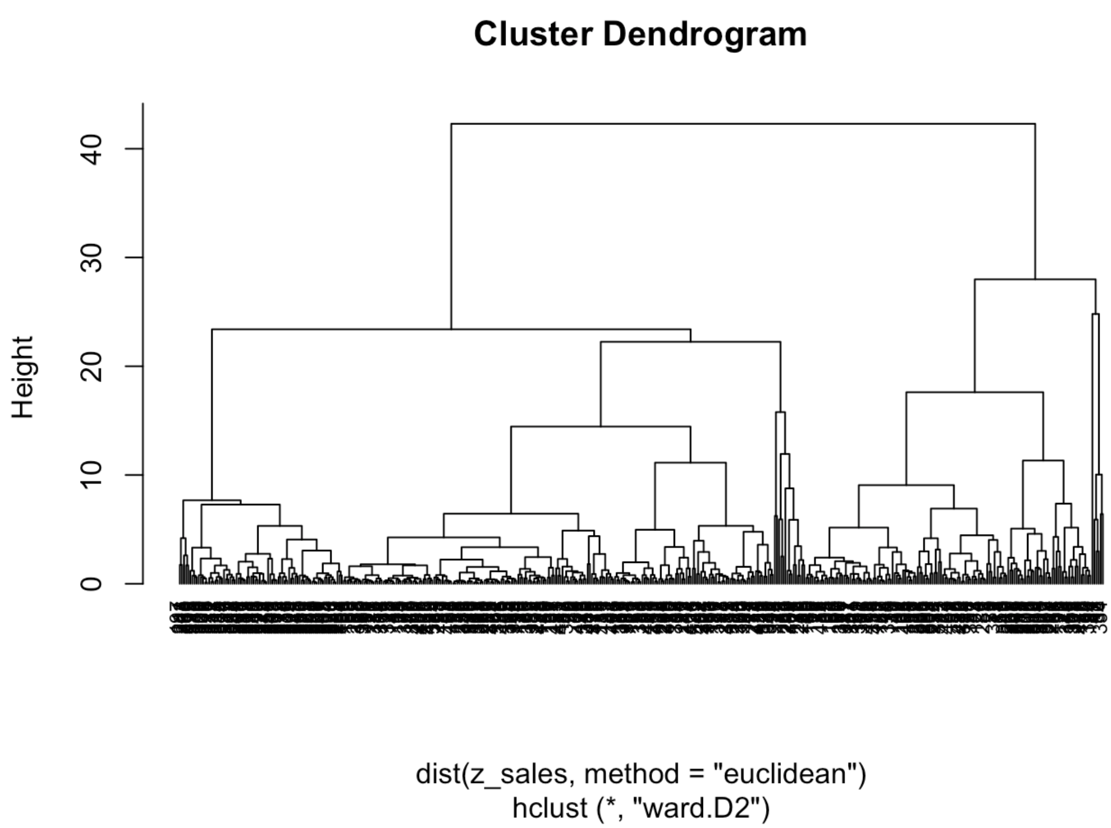
```
hclust1 <- hclust(dist(z_sales, method="euclidean"), method="ward.D2")

hclust1
```

```
##
## Call:
## hclust(d = dist(z_sales, method = "euclidean"), method = "ward.D2")
##
## Cluster method   : ward.D2
## Distance         : euclidean
## Number of objects: 440
```

Creating and runnning the below model, we see that we have 440 objects created, note, this is the total number of original observations.

```
plot(hclust1, hang = -0.01, cex = 0.7)
```



Now, lets visualize these objects in a dendrogram, and as we can see, our first go around at an HCA comes out a little messy, but you can begin to see clusters. Lets experiment with some of the other distance measures to see if we can gain some more context.

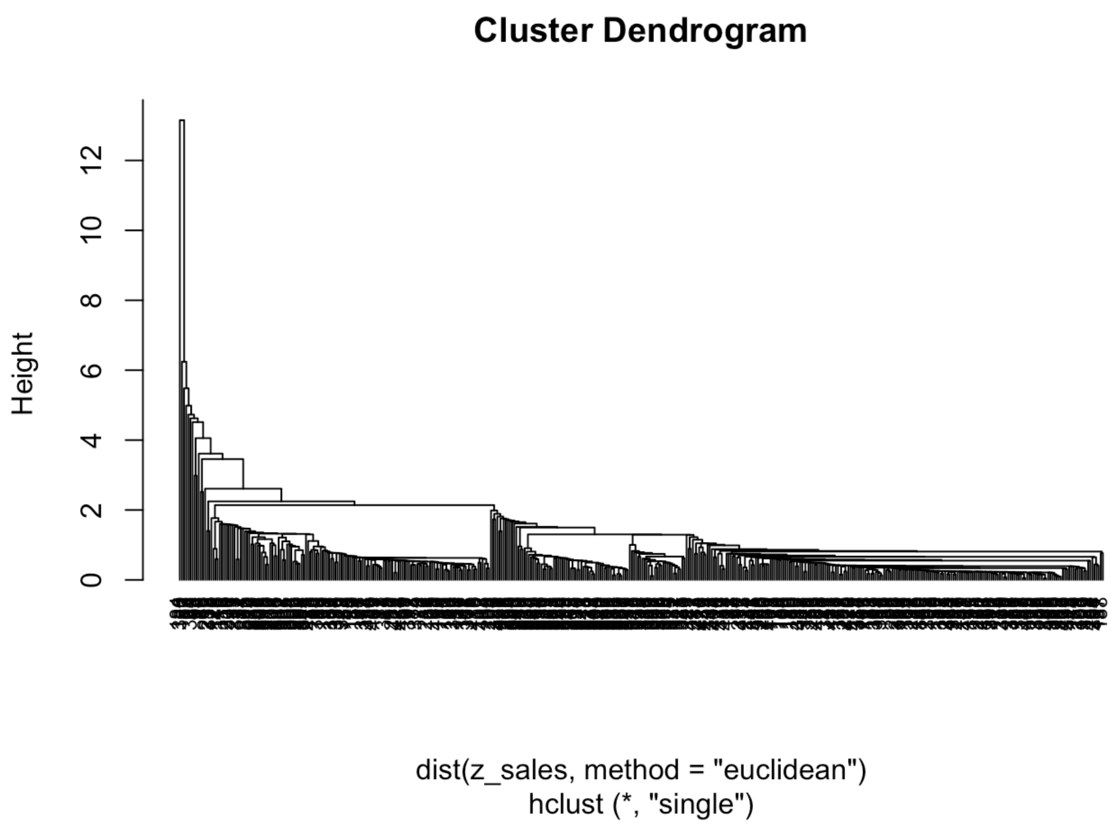
Single linkage measurement to perform agglomerative HCS using Euclidian distance

```
hclust2 <- hclust(dist(z_sales, method="euclidean"), method="single")

hclust2
```

```
##
## Call:
## hclust(d = dist(z_sales, method = "euclidean"), method = "single")
##
## Cluster method   : single
## Distance         : euclidean
## Number of objects: 440
```

```
plot(hclust2, hang = -0.01, cex = 0.7)
```



As anticipated due to the extreme simplicity of the single linkage method, the ward minimum variance method appeared to work much better in visually allowing us to see clusters of data.

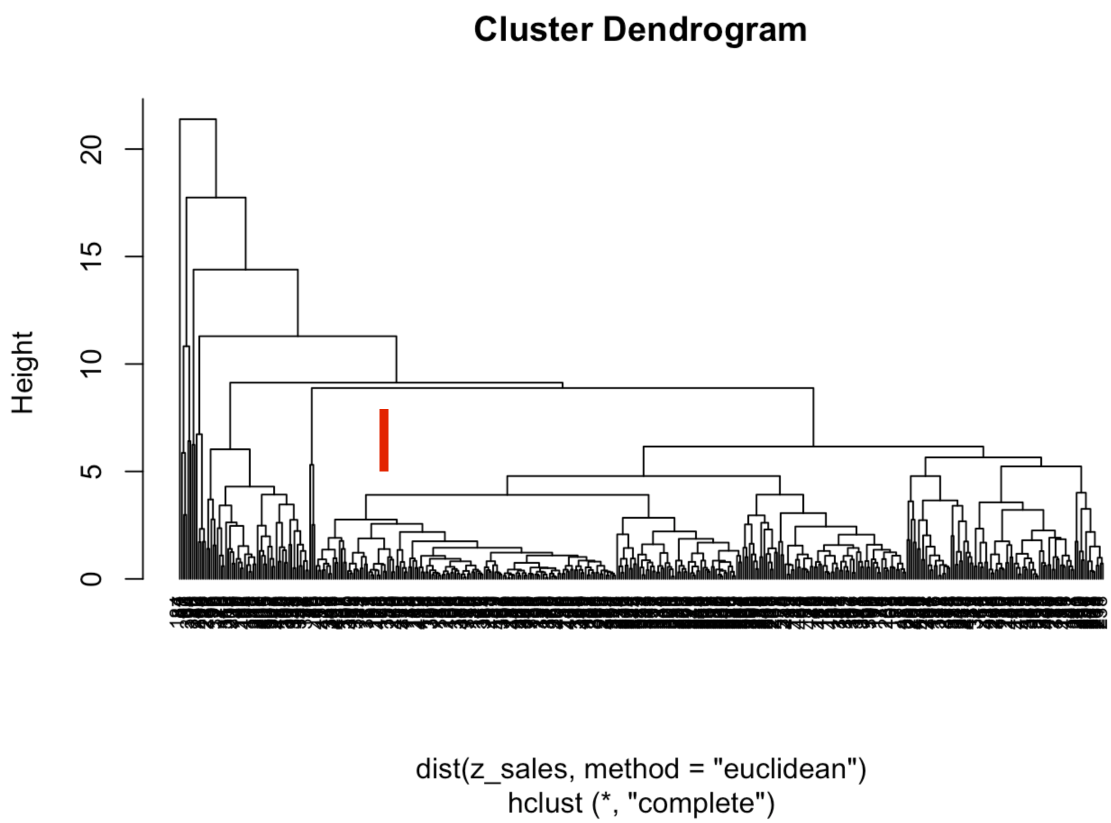
Complete linkage measurement to perform agglomerative HCS using Euclidian distance

```
hclust3 <- hclust(dist(z_sales, method="euclidean"), method="complete")

hclust3
```

```
##
## Call:
## hclust(d = dist(z_sales, method = "euclidean"), method = "complete")
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 440
```

```
plot(hclust3, hang = -0.01, cex = 0.7)
```



Pretty interesting, when looking at complete linkage we can see via the hierarchical structure of the dendrogram that some more visually obvious clustering is occurring, however it also appears to be skewed in an interesting left to right fashion.

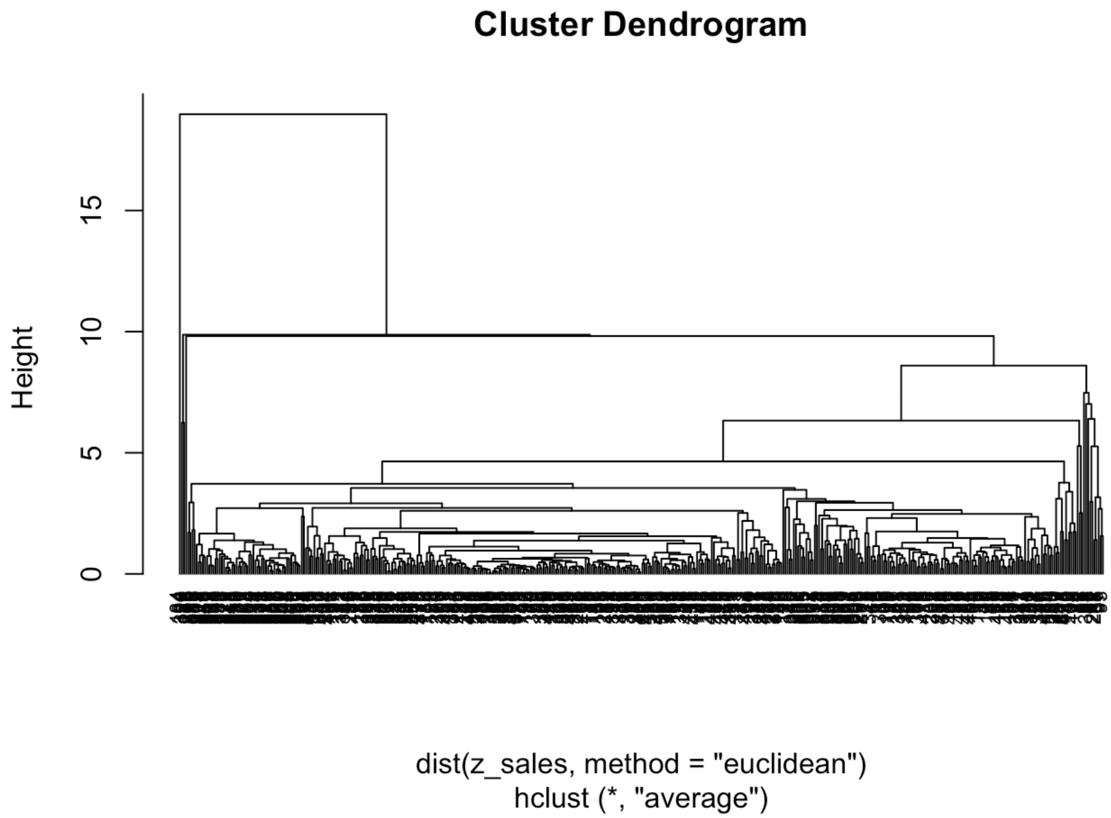
Average linkage measurement to perform agglomerative HCS using Euclidian distance

```
hclust4 <- hclust(dist(z_sales, method="euclidean"), method="average")

hclust4
```

```
##
## Call:
## hclust(d = dist(z_sales, method = "euclidean"), method = "average")
##
## Cluster method   : average
## Distance         : euclidean
## Number of objects: 440
```

```
plot(hclust4, hang = -0.01, cex = 0.7)
```



Using the average linkage method for distance measurement has removed much of the imbalance we have seen in the complete and single linkage distance methods, however does not portray the clusters visually as nicely as Ward's method.

A small conclusion about distance measurement

After further reading it appears that the imbalance occurring with the hierarchical structures of the complete and single linkage measurement methods are more or less highlighting the variance of size of clusters throughough a group, and this variance is more or less normalized, for lack of a better term, when using wards method in agglomerative HCA because of the sum of sqares vs. mean comparison.

Divisive Hierarchical Clustering

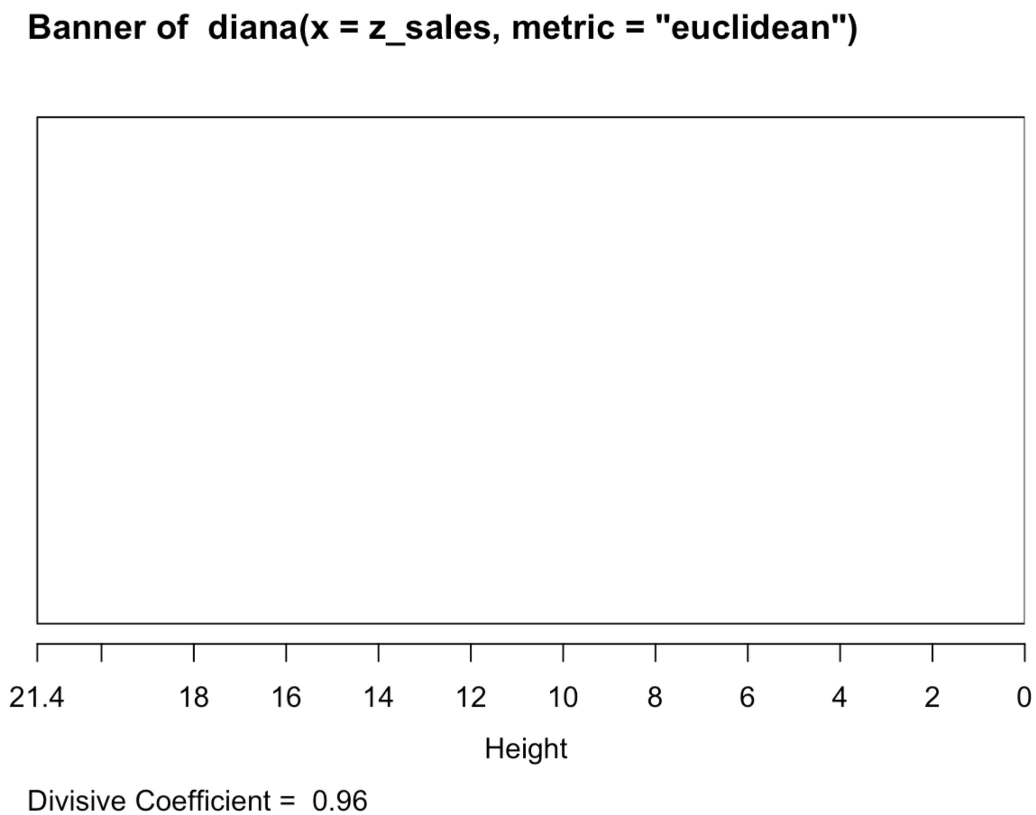
Now, for the top down approach.

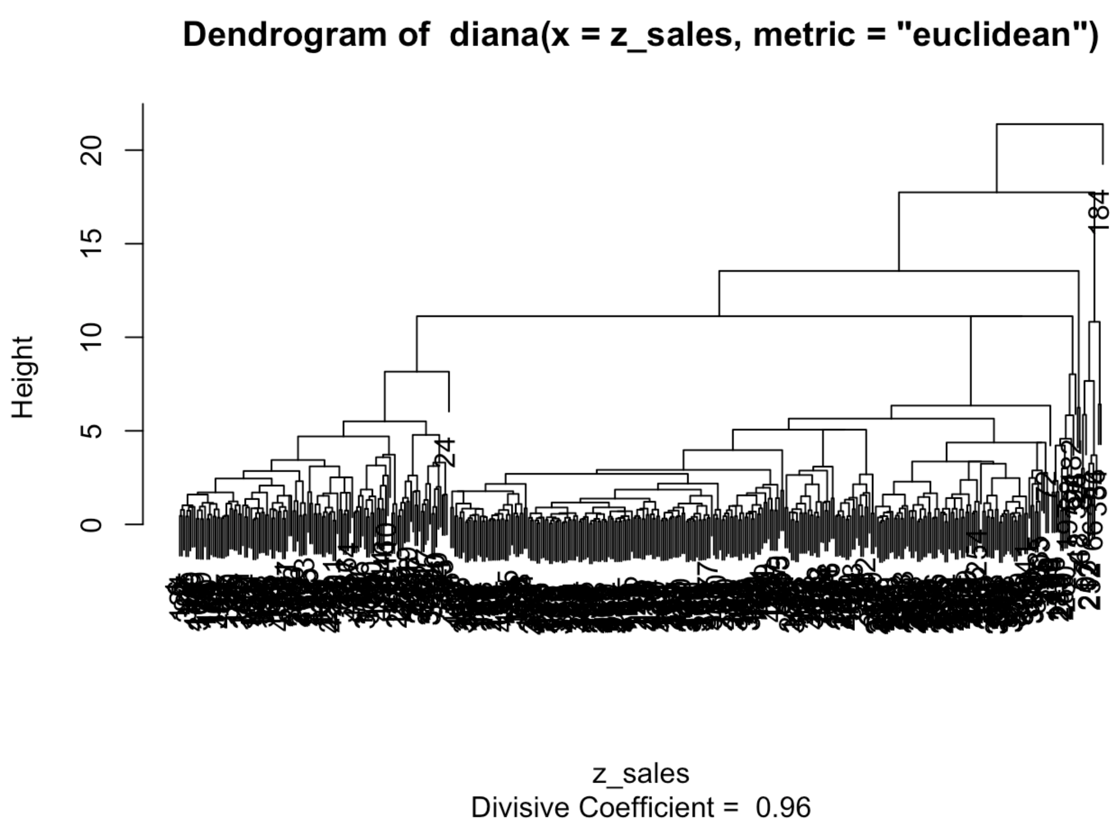
```
hclust5 <- diana(z_sales, metric = "euclidean")

hclust5$dc
```

```
## [1] 0.9610061
```

```
plot(hclust5)
```





Building this model, we see some similarities and differences with our output. First, we can see that the divisive coefficient is 0.96 on this model, this tells us the clusturing structure of the dataset in that how widely the clusters span to classify a dataset. We see a score of 0.96, which tells us that we have larger clusters in this output, this is consistent for divisive hierarchical clustering in that their strength is measuring large clusters, while agglomerative hierarchical clustering is more apt to measure small clusters.

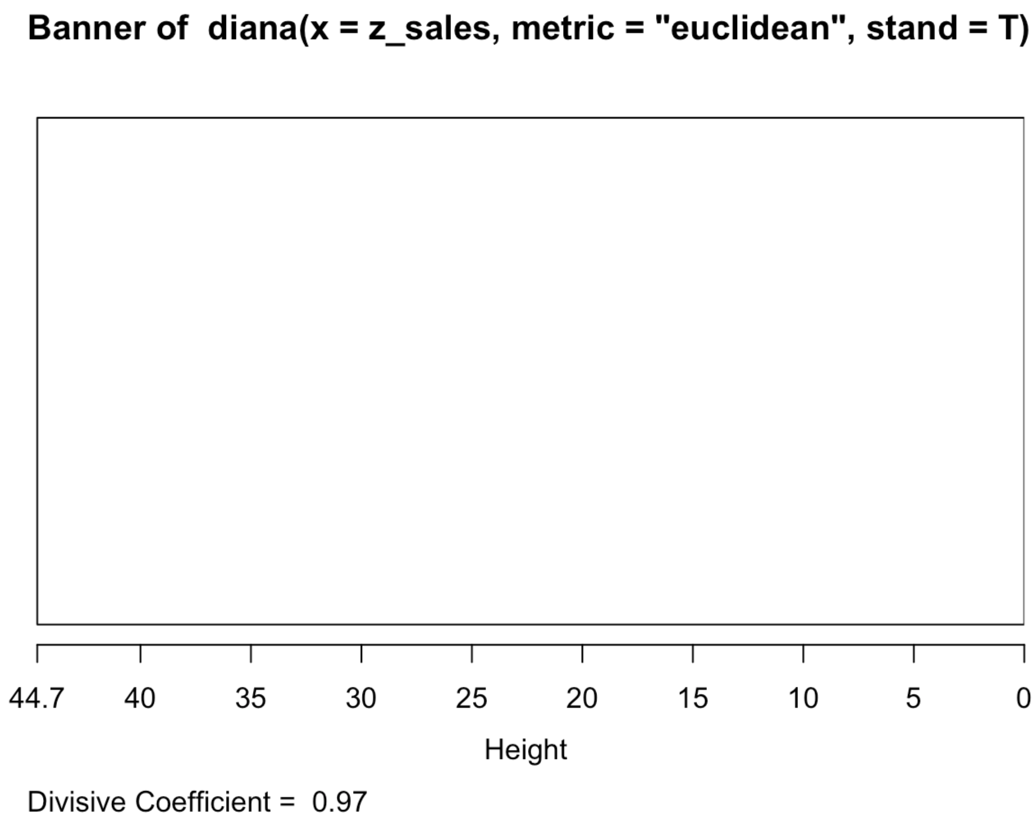
Lets have fun with this one and do some tuning of the model by setting the stand argument to true. When stand is set to true, it standardizes the dissimilarities between groups of data. Setting the agrument this way will more than likely raise our divisive coefficient.

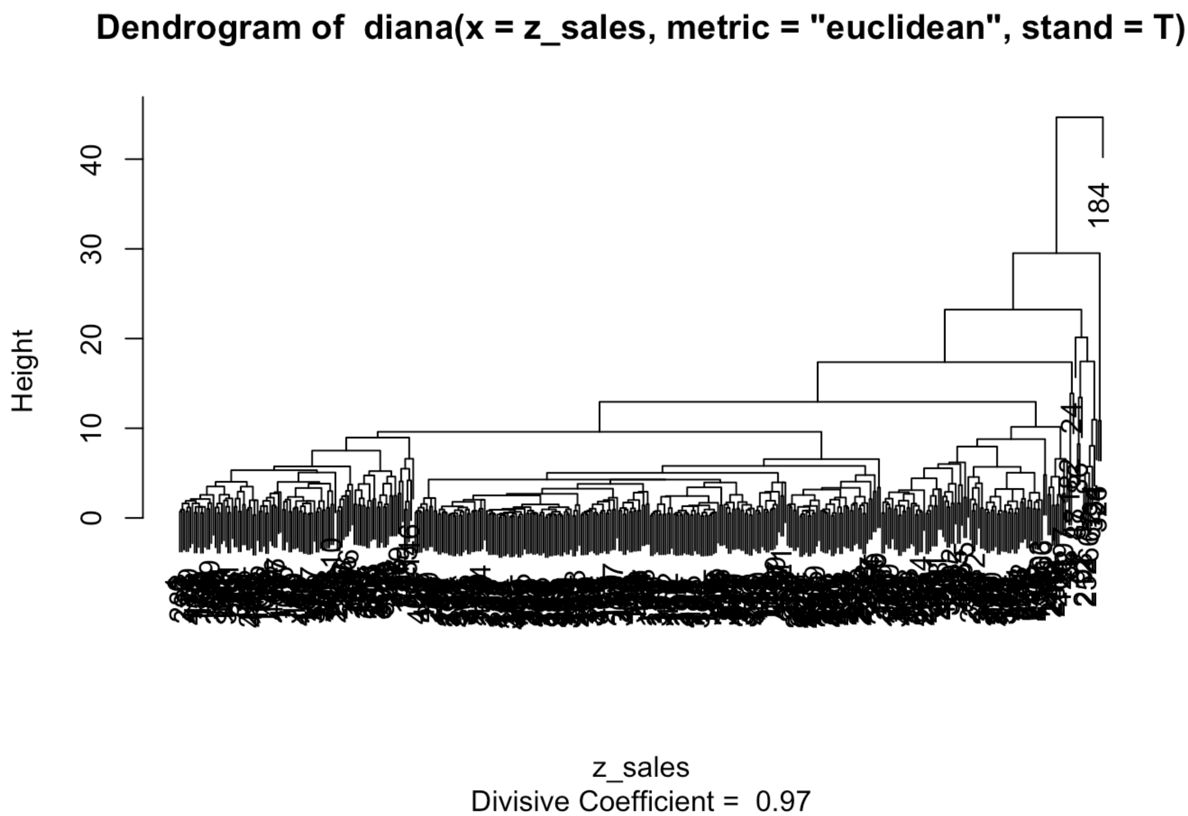
```
hclust6 <- diana(z_sales, metric = "euclidean", stand = T)

hclust6$dc
```

```
## [1] 0.9671841
```

```
plot(hclust6)
```





Looks like our dc increased as predicted, however looking at the dendrogram it doesn't entirely look like the model has improved in displaying more obvious clusters of data.

Cut top cluster into trees to definitively determine clusters of custers based upon sales

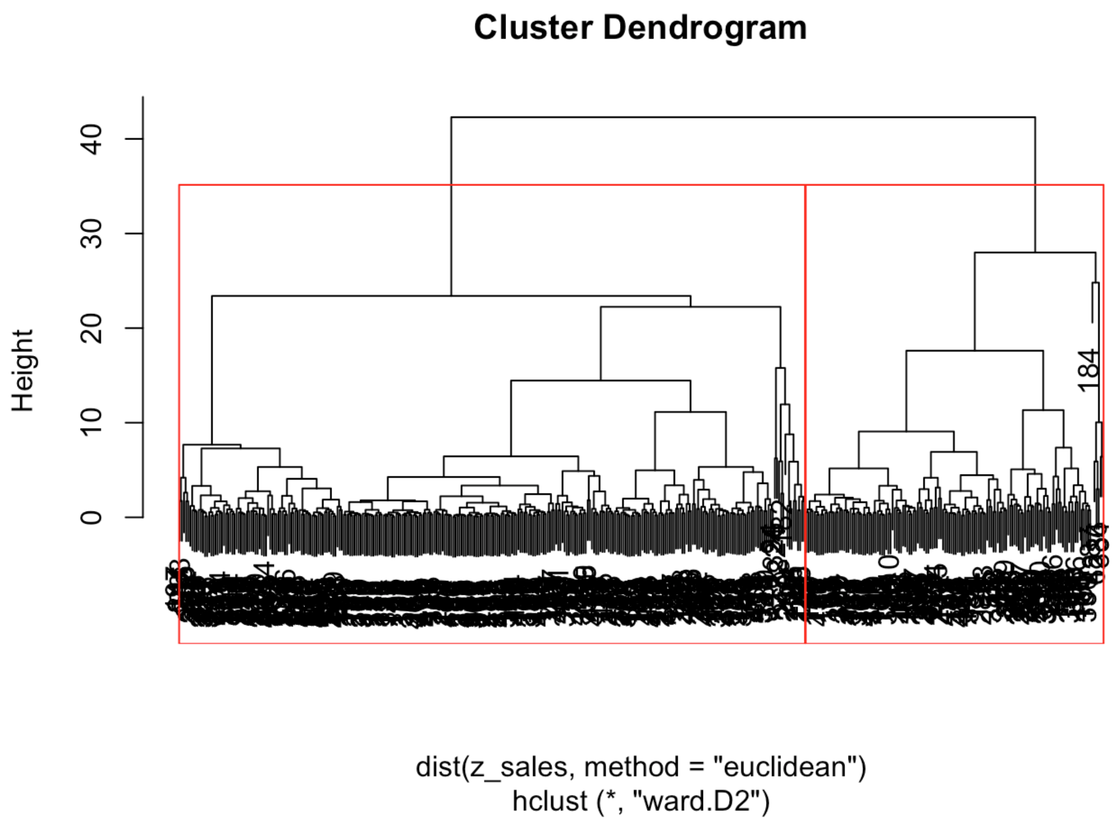
Wards minimum variance to perform agglomerative HCS using Euclidian distance

```
fit1 <- cutree(hclust1, k = 2)

table(fit1)
```

```
## fit1
##   1   2
## 142 298
```

```
plot(hclust1)
rect.hclust(hclust1, k = 2, border = "red")
```



Looking at this agglomerative HCS, we can visually see more clusters, but it runs into trouble when being classified. Because it works from the bottom up, we see really small groups of data falling into essentially clusters of their own as you move the k lower in value and it isn't until we get to the obvious split of two that we no longer have menially sized clusters fall into our analysis.

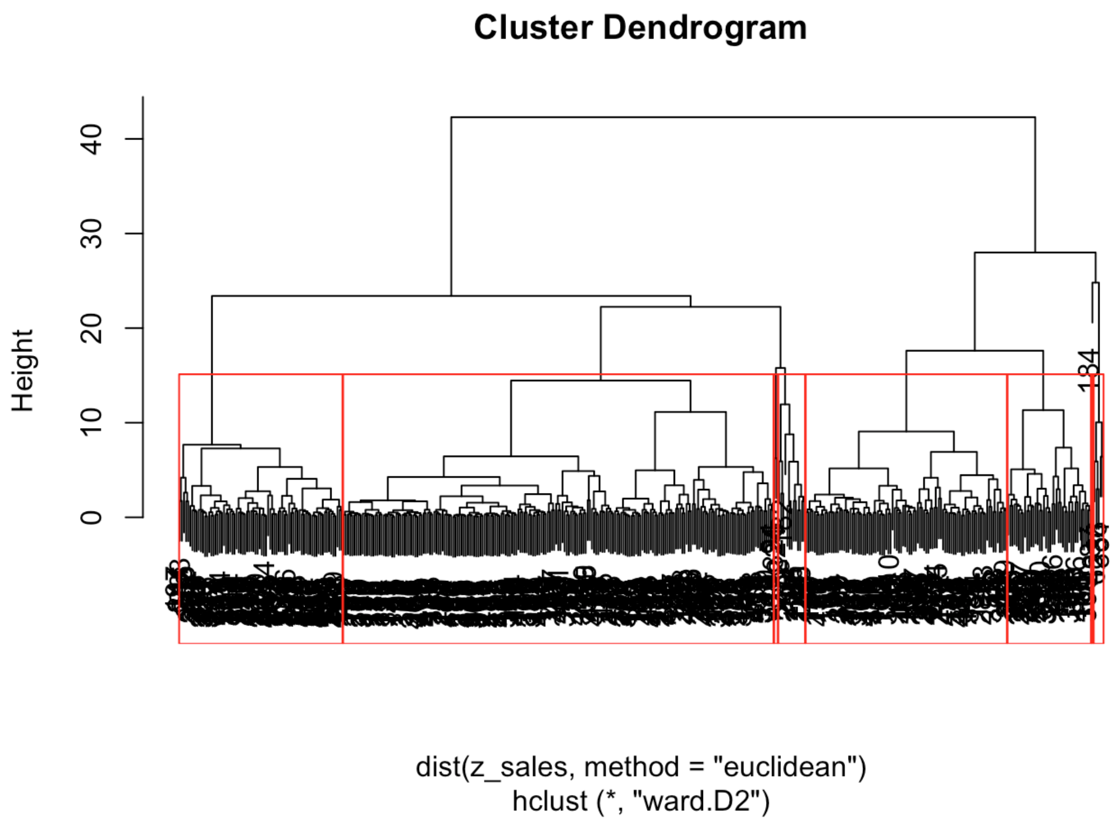
Knowing the strength of agglomerative HCS is towards many small clusters, I will repeat this using an much higher k to see if some understanding can be gained.

```
fit1.2 <- cutree(hclust1, k = 8)

table(fit1.2)
```

```
## fit1.2
##   1   2   3   4   5   6   7   8
##  96 205  13  40   5   2   1  78
```

```
plot(hclust1)
rect.hclust(hclust1, k = 8, border = "red")
```



Using a k of 8, this output is still messy, but makes much more sense compared to the previous output. There are still a few nonsensically small clusters being identified, all of which visually appear to be multiple levels of the tree down in the dendrogram, implying their overall distance from other clusters as an indication of a possible outlier. Lets summarize these clusters to see if we can learn anything about who these customers are, to visualize this I will .

```
fit1.2 <- as.data.frame(fit1.2)

output_table1 <- cbind(fit1.2, sales)

output_table1$fit1.2 <- as.factor(output_table1$fit1.2)

summary(output_table1)
```

```
##      fit1.2       Channel         Region          Fresh
##  2      :205   Min.   :1.000   Min.   :1.000   Min.   :     3
##  1      : 96   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:  3128
##  8      : 78   Median :1.000   Median :3.000   Median :  8504
##  4      : 40   Mean   :1.323   Mean   :2.543   Mean   : 12000
##  3      : 13   3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.: 16934
##  5      :  5   Max.   :2.000   Max.   :3.000   Max.   :112151
##  (Other):  3
##       Milk          Grocery          Frozen        Detergents_Paper
##  Min.   :   55   Min.   :    3   Min.   :   25.0   Min.   :    3.0
##  1st Qu.: 1533   1st Qu.: 2153   1st Qu.:  742.2   1st Qu.:  256.8
##  Median : 3627   Median : 4756   Median : 1526.0   Median :  816.5
##  Mean   : 5796   Mean   : 7951   Mean   : 3071.9   Mean   : 2881.5
##  3rd Qu.: 7190   3rd Qu.:10656   3rd Qu.: 3554.2   3rd Qu.: 3922.0
##  Max.   :73498   Max.   :92780   Max.   :60869.0   Max.   :40827.0
##
##    Delicassen
##  Min.   :    3.0
##  1st Qu.:  408.2
##  Median :  965.5
##  Mean   : 1524.9
##  3rd Qu.: 1820.2
##  Max.   :47943.0
##
```

```
describe.by(output_table1, output_table1$fit1.2)
```

```
## Warning: describe.by is deprecated. Please use the describeBy function
```

```
## $`1`
##                 vars  n     mean      sd  median  trimmed     mad  min
## fit1.2*            1 96     1.00    0.00     1.0     1.00    0.00    1
## Channel            2 96     2.00    0.00     2.0     2.00    0.00    2
## Region             3 96     2.85    0.43     3.0     2.97    0.00    1
## Fresh              4 96  9545.69 8735.79  7705.5  8414.96 7760.67   23
## Milk               5 96  7172.62 3238.28  6645.5  7037.46 3179.44  928
## Grocery            6 96 11342.32 4863.69 10694.5 10893.51 4375.15 2743
## Frozen             7 96  1597.96 1873.32  1012.0  1235.37 1018.55   33
## Detergents_Paper   8 96  4606.34 2329.53  4331.5  4563.77 2765.79  332
## Delicassen         9 96  1447.86 1287.09  1316.5  1279.60 1180.89    3
##                   max range  skew kurtosis     se
## fit1.2*             1     0   NaN      NaN   0.00
## Channel             2     0   NaN      NaN   0.00
## Region              3     2 -3.02     8.59   0.04
## Fresh           40721 40698  1.26     1.62 891.59
## Milk            16729 15801  0.46    -0.07 330.51
## Grocery         28986 26243  0.96     1.03 496.40
## Frozen          11559 11526  2.72     9.33 191.20
## Detergents_Paper 10069  9737  0.22    -0.70 237.76
## Delicassen       7844  7841  1.93     6.04 131.36
## 
## $`2`
##                 vars   n     mean       sd median  trimmed     mad min
## fit1.2*            1 205     2.00     0.00      2     2.00    0.00   2
## Channel            2 205     1.00     0.00      1     1.00    0.00   1
## Region             3 205     2.98     0.15      3     3.00    0.00   2
## Fresh              4 205 11659.55 10204.24   9061 10263.83 8972.70   3
## Milk               5 205  3104.87  3063.34   2102  2566.38 1755.40  55
## Grocery            6 205  3565.92  2966.27   2593  3078.21 1630.86   3
## Frozen             7 205  3213.70  3645.97   1752  2527.16 1951.10  25
## Detergents_Paper   8 205   735.24  1046.04    356   503.33  367.68   3
## Delicassen         9 205  1112.11  1088.15    776   924.72  701.27   3
##                   max range  skew kurtosis     se
## fit1.2*             2     0   NaN      NaN   0.00
## Channel             1     0   NaN      NaN   0.00
## Region              3     1 -6.12    35.65   0.01
## Fresh           43088 43085  1.13     0.69 712.70
## Milk            21858 21803  2.76    10.80 213.95
## Grocery         16483 16480  1.96     4.52 207.17
## Frozen          17866 17841  1.78     2.94 254.65
## Detergents_Paper  6907  6904  3.17    12.52  73.06
## Delicassen       5864  5861  1.87     3.91  76.00
## 
## $`3`
##                 vars  n     mean       sd median  trimmed      mad   min
## fit1.2*            1 13     3.00     0.00      3     3.00     0.00     3
## Channel            2 13     1.08     0.28      1     1.00     0.00     1
## Region             3 13     2.54     0.88      3     2.64     0.00     1
## Fresh              4 13 54537.92 23093.99  53205 52595.55 11215.87 18291
## Milk               5 13  8253.54 11202.37   4411  6392.55  1390.68   555
## Grocery            6 13  9451.69  6978.86   7336  9086.45  5273.61   902
## Frozen             7 13  8835.31  5022.38   6422  8529.00  1879.94  3012
## Detergents_Paper   8 13  1796.62  1587.10   1041  1654.18   650.86   212
## Delicassen         9 13  5435.38  5899.23   2498  4900.64  2342.51   230
##                    max range  skew kurtosis      se
## fit1.2*              3     0   NaN      NaN    0.00
## Channel              2     1  2.82     6.44    0.08
## Region               3     2 -1.13    -0.76    0.24
```

```
## Fresh            112151 93860  0.82     0.70 6405.12
## Milk              36423 35868  1.68     1.14 3106.98
## Grocery           22019 21117  0.64    -1.09 1935.59
## Frozen            18028 15016  0.81    -1.02 1392.96
## Detergents_Paper   4948  4736  1.00    -0.77  440.18
## Delicassen        16523 16293  0.87    -1.07 1636.15
##
## $`4`
##                  vars  n     mean      sd  median  trimmed     mad  min
## fit1.2*             1 40     4.00    0.00     4.0     4.00    0.00    4
## Channel             2 40     2.00    0.00     2.0     2.00    0.00    2
## Region              3 40     2.00    0.88     2.0     2.00    1.48    1
## Fresh               4 40  4841.00 4778.55  3531.5  4133.81 3518.21   18
## Milk                5 40 14486.12 6954.10 13089.5 13959.81 7221.74 3737
## Grocery             6 40 22490.05 8667.02 21876.0 22065.81 7489.35 6089
## Frozen              7 40  1573.33 1376.48  1196.0  1363.84  875.48   36
## Detergents_Paper    8 40 10896.33 4923.69 10768.0 10556.28 5101.63 3891
## Delicassen          9 40  1998.15 1802.19  1381.5  1765.53 1265.40   37
##                    max range skew kurtosis      se
## fit1.2*              4     0  NaN      NaN    0.00
## Channel              2     0  NaN      NaN    0.00
## Region               3     2 0.00    -1.73    0.14
## Fresh            22039 22021 1.52     2.33  755.56
## Milk             29892 26155 0.57    -0.72 1099.54
## Grocery          45828 39739 0.43    -0.07 1370.38
## Frozen            6746  6710 1.68     3.32  217.64
## Detergents_Paper 24231 20340 0.54    -0.35  778.50
## Delicassen        6372  6335 1.04    -0.14  284.95
##
## $`5`
##                  vars n    mean       sd median trimmed      mad   min
## fit1.2*             1 5     5.0     0.00      5     5.0     0.00     5
## Channel             2 5     2.0     0.00      2     2.0     0.00     2
## Region              3 5     2.8     0.45      3     2.8     0.00     2
## Fresh               4 5 25603.0 14578.73  22925 25603.0 19299.00  8565
## Milk                5 5 43460.6 25164.56  46197 43460.6 11952.72  4980
## Grocery             6 5 61472.2 21876.69  59598 61472.2 11416.02 32114
## Frozen              7 5  2636.0  3100.39   1026  2636.0  1326.93   131
## Detergents_Paper    8 5 29974.2  9032.28  26701 29974.2  9831.12 20070
## Delicassen          9 5  2708.8  2243.62   2017  2708.8  1374.37   903
##                    max range  skew kurtosis       se
## fit1.2*              5     0   NaN      NaN     0.00
## Channel              2     0   NaN      NaN     0.00
## Region               3     1 -1.07    -0.92     0.20
## Fresh            44466 35901  0.13    -1.98  6519.80
## Milk             73498 68518 -0.36    -1.49 11253.93
## Grocery          92780 60666  0.10    -1.51  9783.56
## Frozen            7782  7651  0.75    -1.37  1386.53
## Detergents_Paper 40827 20757  0.17    -2.13  4039.36
## Delicassen        6465  5562  0.77    -1.30  1003.38
##
## $`6`
##                  vars n    mean       sd  median trimmed      mad   min
## fit1.2*             1 2     6.0     0.00     6.0     6.0     0.00     6
## Channel             2 2     1.0     0.00     1.0     1.0     0.00     1
## Region              3 2     2.5     0.71     2.5     2.5     0.74     2
## Fresh               4 2 22015.5 15134.21 22015.5 22015.5 15866.04 11314
## Milk                5 2  9937.0  9683.12  9937.0  9937.0 10151.36  3090
## Grocery             6 2  7844.0  8176.98  7844.0  7844.0  8572.39  2062
## Frozen              7 2 47939.0 18285.78 47939.0 47939.0 19170.02 35009
```

```
## Detergents_Paper    8 2   671.5   849.24   671.5   671.5   890.30    71
## Delicassen          9 2  4153.5  2058.39  4153.5  4153.5  2157.92  2698
##                    max range skew kurtosis      se
## fit1.2*              6     0  NaN      NaN     0.0
## Channel              1     0  NaN      NaN     0.0
## Region               3     1    0    -2.75     0.5
## Fresh            32717 21403    0    -2.75 10701.5
## Milk             16784 13694    0    -2.75  6847.0
## Grocery          13626 11564    0    -2.75  5782.0
## Frozen           60869 25860    0    -2.75 12930.0
## Detergents_Paper  1272  1201    0    -2.75   600.5
## Delicassen        5609  2911    0    -2.75  1455.5
##
## $`7`
##                  vars n  mean sd median trimmed mad   min   max range skew
## fit1.2*             1 1     7 NA      7       7   0     7     7     0   NA
## Channel             2 1     1 NA      1       1   0     1     1     0   NA
## Region              3 1     3 NA      3       3   0     3     3     0   NA
## Fresh               4 1 36847 NA  36847   36847   0 36847 36847     0   NA
## Milk                5 1 43950 NA  43950   43950   0 43950 43950     0   NA
## Grocery             6 1 20170 NA  20170   20170   0 20170 20170     0   NA
## Frozen              7 1 36534 NA  36534   36534   0 36534 36534     0   NA
## Detergents_Paper    8 1   239 NA    239     239   0   239   239     0   NA
## Delicassen          9 1 47943 NA  47943   47943   0 47943 47943     0   NA
##                  kurtosis se
## fit1.2*                NA NA
## Channel                NA NA
## Region                 NA NA
## Fresh                  NA NA
## Milk                   NA NA
## Grocery                NA NA
## Frozen                 NA NA
## Detergents_Paper       NA NA
## Delicassen             NA NA
##
## $`8`
##                  vars  n     mean      sd median  trimmed     mad min
## fit1.2*             1 78     8.00    0.00    8.0     8.00    0.00   8
## Channel             2 78     1.00    0.00    1.0     1.00    0.00   1
## Region              3 78     1.28    0.45    1.0     1.23    0.00   1
## Fresh               4 78 11051.44 8351.20 9020.0 10276.64 8572.39 444
## Milk                5 78  3300.24 3861.62 1914.0  2555.23 1578.23 258
## Grocery             6 78  4012.73 3411.78 2833.0  3421.66 2066.74 489
## Frozen              7 78  2769.92 2935.43 1830.0  2258.23 1784.31  91
## Detergents_Paper    8 78   823.86 1174.72  379.0   548.47  383.25   5
## Delicassen          9 78  1071.60 1075.35  763.5   899.66  669.39   7
##                    max range skew kurtosis     se
## fit1.2*              8     0  NaN      NaN   0.00
## Channel              1     0  NaN      NaN   0.00
## Region               2     1 0.95    -1.11   0.05
## Fresh            31614 31170 0.74    -0.41 945.59
## Milk             23527 23269 2.86    10.13 437.24
## Grocery          16966 16477 1.78     3.19 386.31
## Frozen           18711 18620 2.69    10.08 332.37
## Detergents_Paper  5828  5823 2.41     5.40 133.01
## Delicassen        6854  6847 2.53     9.23 121.76
##
## attr(,"call")
## by.data.frame(data = x, INDICES = group, FUN = describe, type = type)
```

Conclusion of Clusters

We can see via the large output above that we have many similar groups to our previous kmeans analysis of this data, however the information is not nearly as easily discernable. When we look at the summary output by fit we see that we have cluster 7 as more of a large volume grocer without many detergents, which could imply that it could be somewhre like trader joes. Clusters 4, 5 and 1 are all large volume buyers with various proportions of the all items available. Thus they could be a WalMart, Costco or even a King Soopers. We see other clusters that have overall low volumes with a higher proportion of the basics, thus convenience grocers or even pharmacies.

Overall this has provided for some interesting insight for our wholesaler, whom may also have some additional context to this output given his/her experience in this vertical. Some actionable insight off of this data could be package deals, streamlining of logistics, or even direct shipment options based upon clusters to essentially understand what a retailer/grocer might order before they actually do.