Wholesale Customer Segmentation Analysis

K-Means and Hierarchical Cluster Analysis

Sean O'Malley

Objective : Using unsupervised learning, predict the customer segments.

Methodology : K-Means Cluster Analysis

Ingest Data

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

EDA and Data Manipulation

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 however perform a z-score standardization of the features with larger volumes, because the numeric standardization of high feature values in a multidimensional space such as k-means is wildly important for scale.

```
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  
```

```
glimpse(sales)
```

```
## Observations: 440
## Variables: 8
## $ Channel          <int> 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, ...
## $ Region           <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ...
## $ Fresh            <int> 12669, 7057, 6353, 13265, 22615, 9413, 12126,...
## $ Milk             <int> 9656, 9810, 8808, 1196, 5410, 8259, 3199, 495...
## $ Grocery          <int> 7561, 9568, 7684, 4221, 7198, 5126, 6975, 942...
## $ Frozen           <int> 214, 1762, 2405, 6404, 3915, 666, 480, 1669, ...
## $ Detergents_Paper <int> 2674, 3293, 3516, 507, 1777, 1795, 3140, 3321...
## $ Delicassen       <int> 1338, 1776, 7844, 1788, 5185, 1451, 545, 2566...
```

```
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
```

K-Means Cluster Analysis

To serve as a refresher, the k-means algorithm assigns each of the *n* examples to one of the *k* clusters, where *k* is a number that has been determined ahead of time. The goal is to minimize the differences within each cluster and maximize the differences between the clusters.

To outline the process for you, we will first use stats packages kmeans function to output a minimially viable product of sorts to get a feel for the process, we will then experiment to find the optimal k, visualizing the multidimensional space throughout. Finally, upon closing in on who our customer segments really are, we will assess and interpret the insight the algorithm has offered us.

k = 5

Create Cluster:

```
set.seed(777)

k_cluster5 <- kmeans(z_sales, 5)
```

View Cluster Output:

```
summary(k_cluster5$cluster)
```

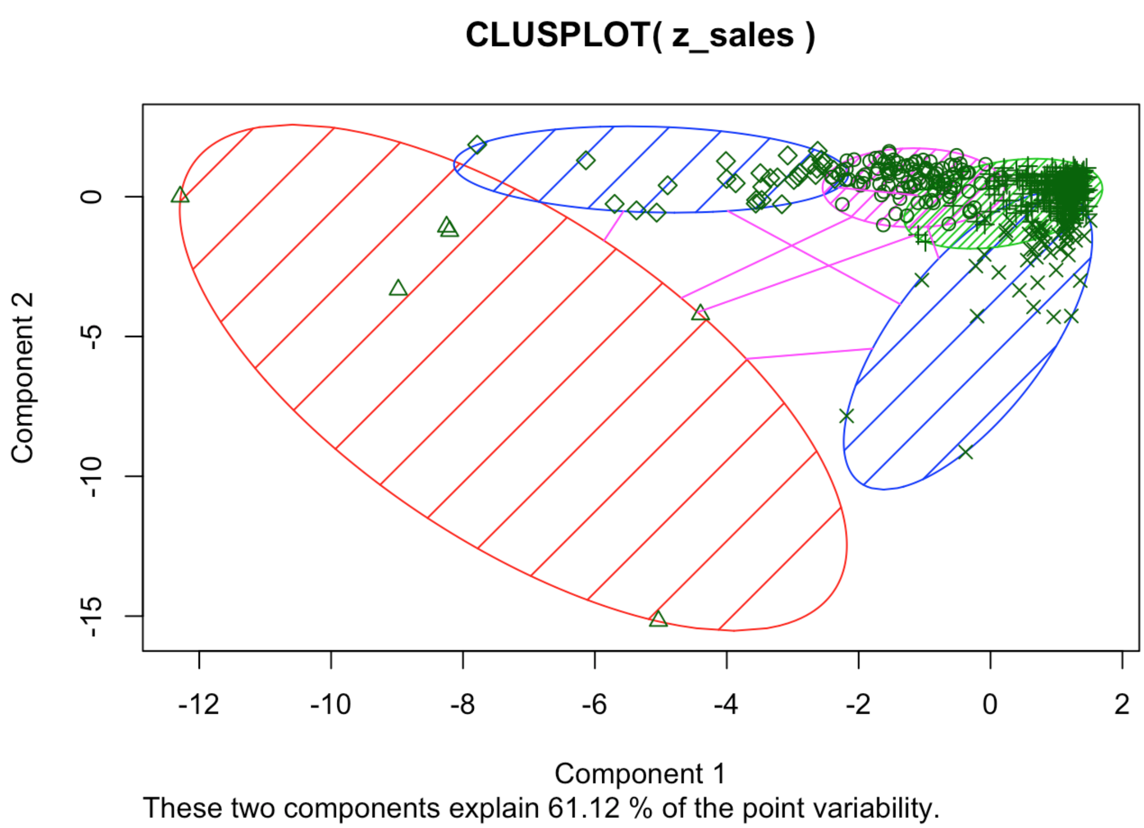
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   2.768   3.000   5.000
```

```
k_cluster5$centers
```

```
##      Channel      Region       Fresh        Milk     Grocery      Frozen
## 1  1.4470045  0.1840857 -0.2787551  0.2364237  0.3843672 -0.3318920
## 2  1.0909184  0.5899967  1.4583872  5.8244179  4.1134434  1.2464076
## 3 -0.6895122 -0.1111223 -0.2326434 -0.3835981 -0.4583531 -0.1396797
## 4 -0.6103820  0.2551543  1.7104343 -0.1252780 -0.2767402  1.3007500
## 5  1.4470045 -0.3140778 -0.4948948  1.3658517  2.0733753 -0.2882923
##   Detergents_Paper  Delicassen
## 1        0.3923482 -0.01885237
## 2        3.4626351  3.99782720
## 3       -0.4377247 -0.19351026
## 4       -0.4429612  0.37756656
## 5        2.3063358  0.16713186
```

Visualize Cluster:

```
clusplot(z_sales, k_cluster5$cluster, color = T, shade = T)
```



After just throwing code out to see how things worked we see that there are huge improvements that can be made upon this first model, and that our guess of 5 clusters was a little off. We will now perform analysis on how to find the best fit of k using the elbow method.

Improve Model Using Elbow Method:

The elbow method looks at the percentage of variance explained as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data. More precisely, if one plots the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the “elbow criterion”. This “elbow” cannot always be unambiguously identified.

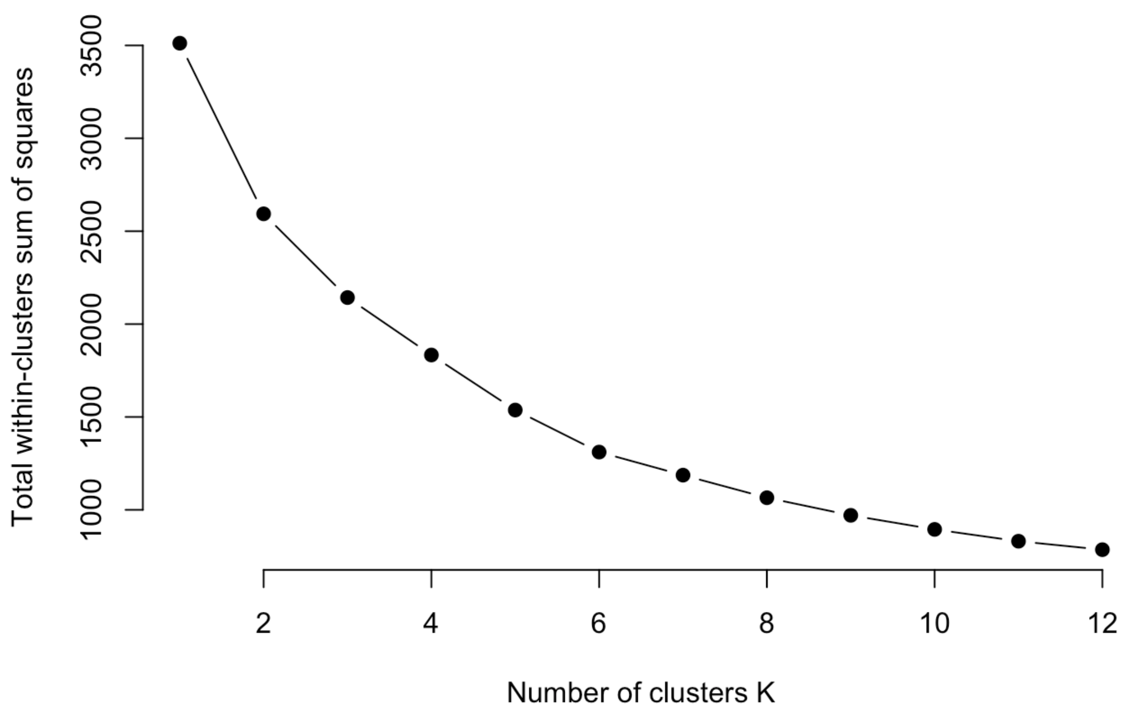
Source: link (https://www.r-bloggers.com/finding-optimal-number-of-clusters/)

I have created an object that sapplys a user defined function to an array with the goal of determining the optimal variance explained as more clusters are added within the range of 2 to 12 clusters

```
k_max <- 12

wss <- sapply(1:k_max, function(k){kmeans(z_sales, k, nstart=50, iter.max = 12 )$tot.withinss})

plot(1:k_max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```



Its a pretty tough call looking at this elbow graph, but I'd have to say that 2 or 3 clusters looks like the best option for n of clusters.

k = 3

Create Cluster:

```
k_cluster3 <- kmeans(z_sales, 3)
```

View Cluster Output:

```
summary(k_cluster3$cluster)
```

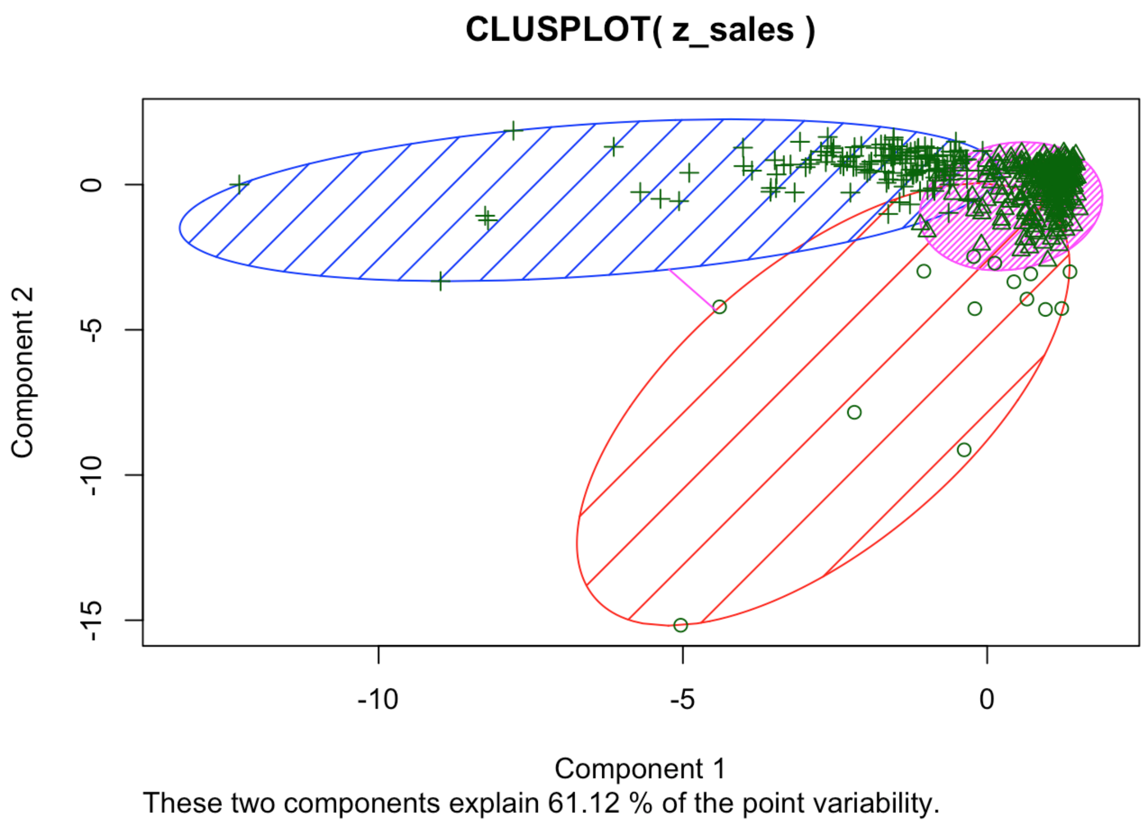
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   2.277   3.000   3.000
```

```
k_cluster3$centers
```

```
##      Channel      Region         Fresh       Milk    Grocery        Frozen
## 1 -0.5369039  0.31323918  2.607675597  0.9959545  0.3738463  3.283683566
## 2 -0.6526757 -0.06913154  0.004304482 -0.3669671 -0.4503362 -0.004713004
## 3  1.4470045  0.11516763 -0.277615869  0.6799786  0.9217916 -0.327976462
##   Detergents_Paper  Delicassen
## 1       -0.2863957  2.62854395
## 2       -0.4439111 -0.14568593
## 3        0.9760571  0.04006842
```

Visualize Cluster:

```
clusplot(z_sales, k_cluster3$cluster, color = T, shade = T)
```



Even looking at this output, a k of 3 looks pretty rough, looking at the points, there appears to be two defined groups more than anything. Lets test this to see if it works better.

k = 2

Create Cluster:

```
k_cluster2 <- kmeans(z_sales, 2)
```

View Cluster Output:

```
summary(k_cluster2$cluster)
```

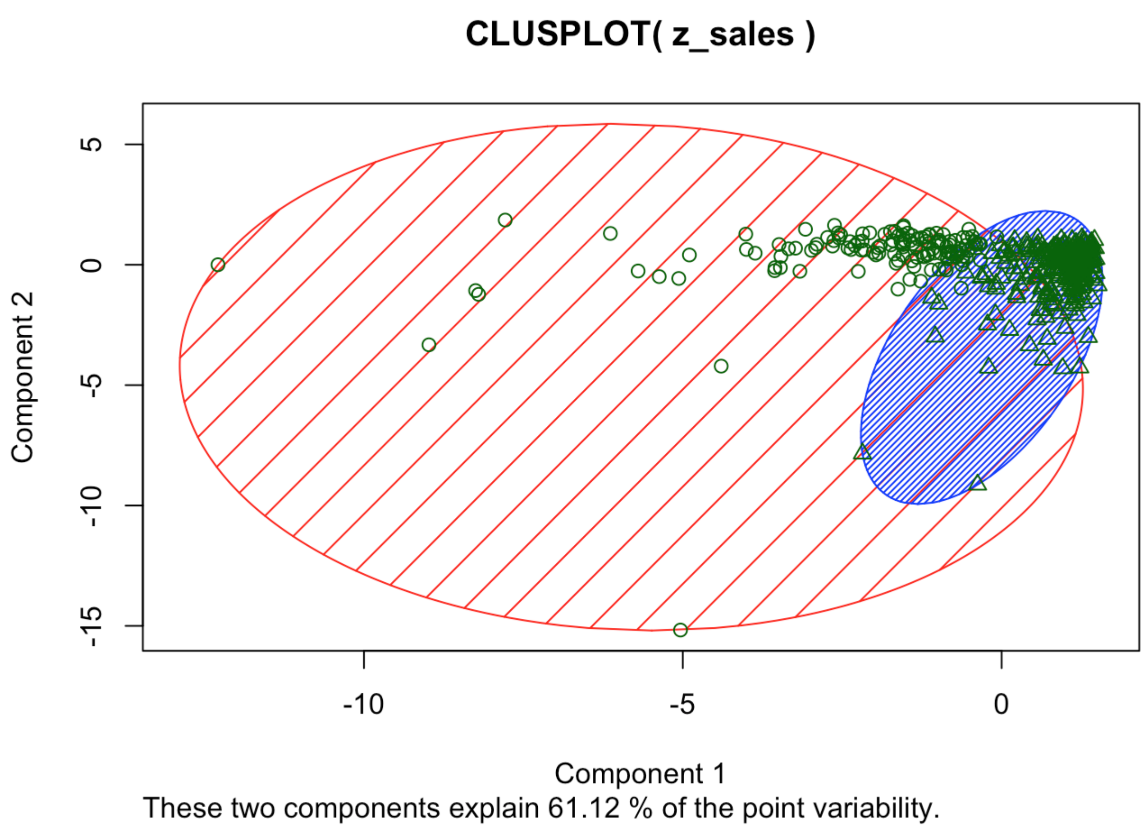
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   1.693   2.000   2.000
```

```
k_cluster2$centers
```

```
##      Channel      Region      Fresh       Milk    Grocery     Frozen
## 1  1.4311785  0.11165038 -0.2814087  0.7585190  0.9527008 -0.2769455
## 2 -0.6334724 -0.04941902  0.1245579 -0.3357379 -0.4216872  0.1225824
##   Detergents_Paper  Delicassen
## 1        0.9868659  0.20688608
## 2       -0.4368095 -0.09157253
```

Visualize Cluster:

```
clusplot(z_sales, k_cluster2$cluster, color = T, shade = T)
```



This model looks much better, it is however interesting looking at the clusters in that there does appear to be 2 very dense clusters, with another large scattering of datapoints that really, in my opinion, skew our ability to properly gain too much insight off of the data.

Looking at the two clusters we have, we see very opposite groups. One group is high on everything but frozen and fresh, while another is only high on frozen and fresh. Something I want to try and do next is another knn algorithm with elbow k process, but this time without region or channel in the analysis. To truely isolate the product types in order to find more unbiased / cleaner groups.

Data Subset Analysis

I am omitting channel and region in order to have the products speak for themselves in regard to who does and does not purchase product combineations together in whole sale markets.

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

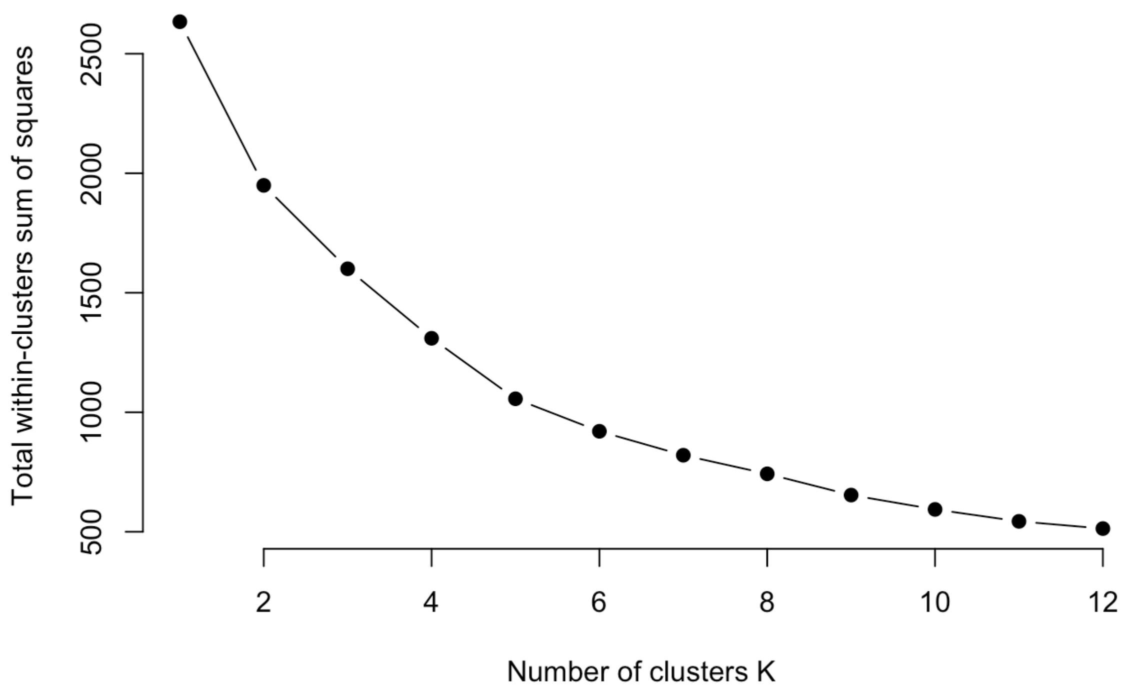
Elbow Test for K Selection:

Spot checking the output below, the elbow appears to be at k=5, lets explore this further, we can also see that by tracking the y axis, our within cluster sum of squares is much higher…indicative of our higher ability to explain the natural groups that exist within the dataset.

```
k_max <- 12

wss <- sapply(1:k_max, function(k){kmeans(z_sales2, k, nstart=50, iter.max = 12 )$tot.withinss})

plot(1:k_max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```



reduced, k = 5

Create Cluster:

```
k_cluster5r <- kmeans(z_sales2, 5)
```

View Cluster Output:

```
summary(k_cluster5r$cluster)
```

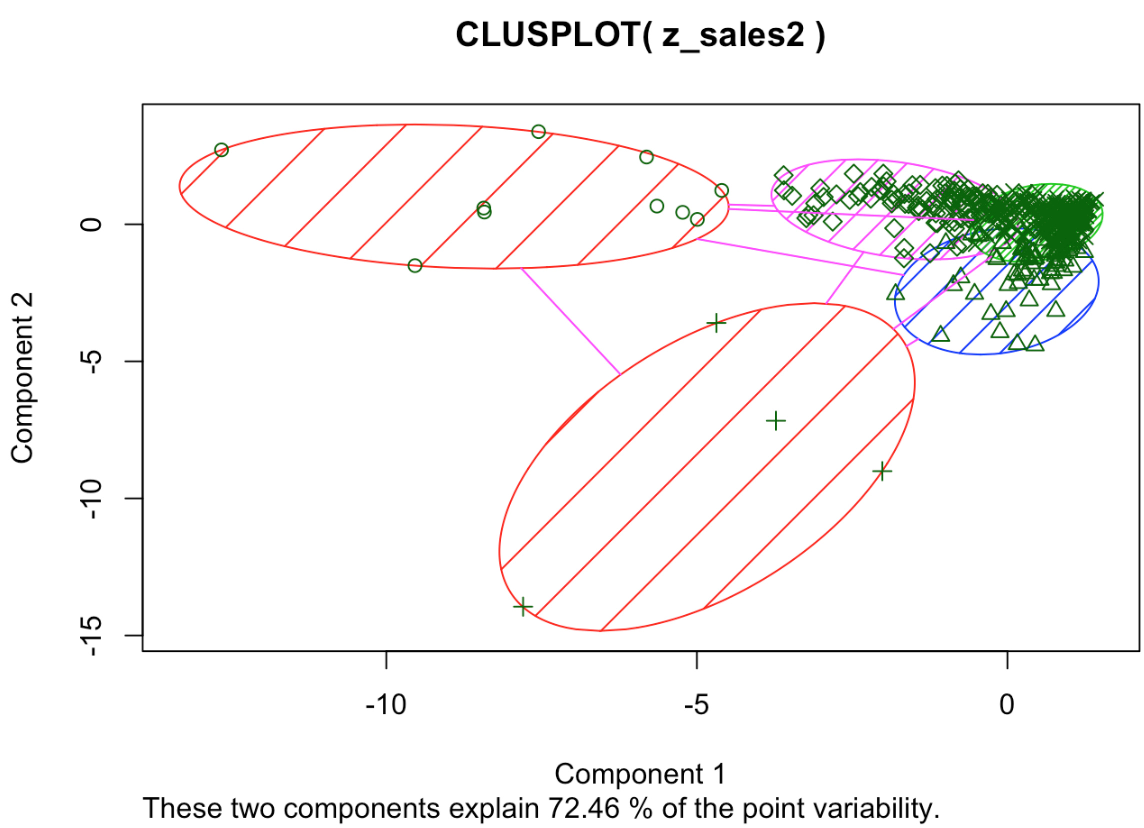
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   4.000   3.827   4.000   5.000
```

```
k_cluster5r$centers
```

```
##        Fresh       Milk    Grocery       Frozen Detergents_Paper
## 1  0.3134735  3.9174467  4.2707490 -0.003570131       4.61291490
## 2  1.4682925 -0.2275567 -0.2858837  0.819308264      -0.42997808
## 3  3.1644391  3.5092697  1.1090489  5.510889477      -0.03827575
## 4 -0.2599009 -0.3809098 -0.4350693 -0.175770281      -0.39558644
## 5 -0.5132347  0.6448681  0.8972434 -0.340250869       0.90563022
##    Delicassen
## 1  0.50279301
## 2  0.24464051
## 3  6.42932569
## 4 -0.19993976
## 5  0.04748801
```

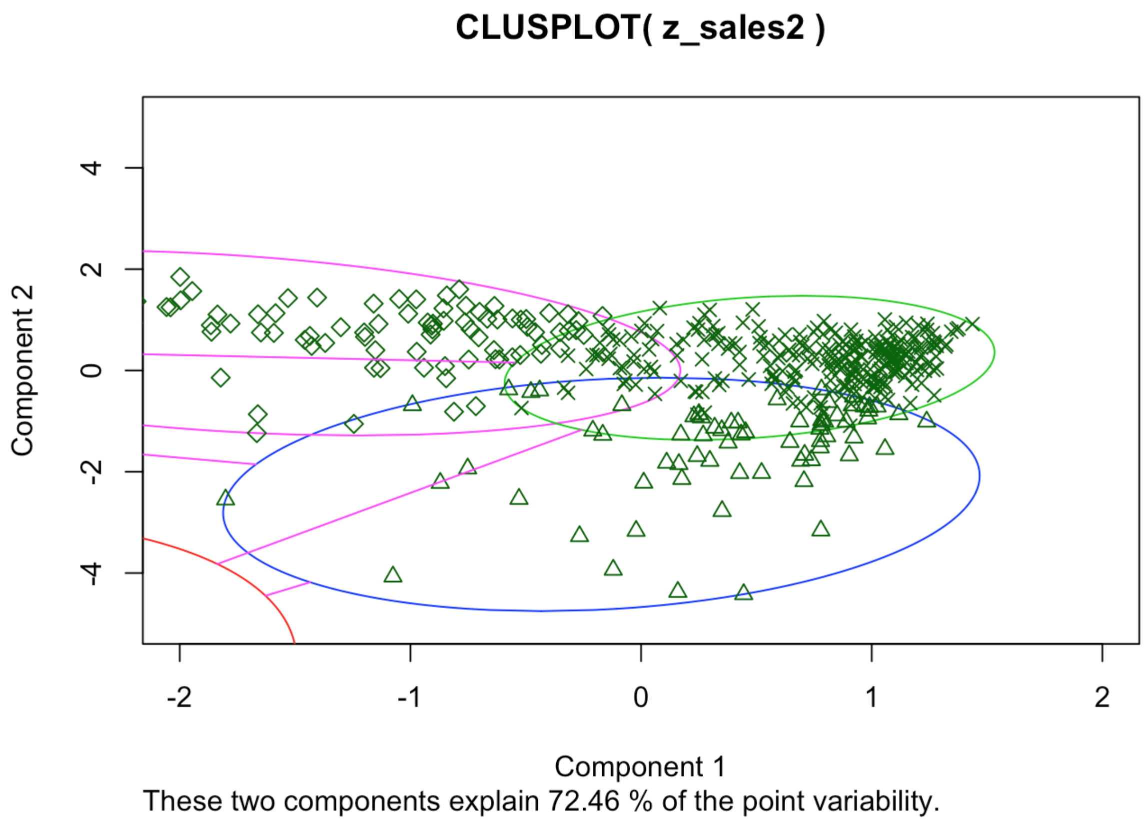
Visualize Cluster:

```
clusplot(z_sales2, k_cluster5r$cluster, color = T, shade = T)
```



First of all, we see a much larger abilty to explain the data at hand than any cluster combination from our previous z score dataset. We can still see the dense and sparse pockets of data within the multidimensional space. I am very happy with the improvement of the cluster analysis we were able to have, just for fun, lets see if we can zoom up on the incredibly dense clustered area to see more closely how well we did.

```
clusplot(z_sales2, k_cluster5r$cluster, color = TRUE,
         ylim = c(-5,5), xlim = c(-2,2))
```



As you zoom in, I can see that though there are defined groups here, that there is some overlap, lets try this thing one more time, but with a k of 4.

reduced, k = 4

Create Cluster:

```
k_cluster4r <- kmeans(z_sales2, 4)
```

View Cluster Output:

```
summary(k_cluster4r$cluster)
```

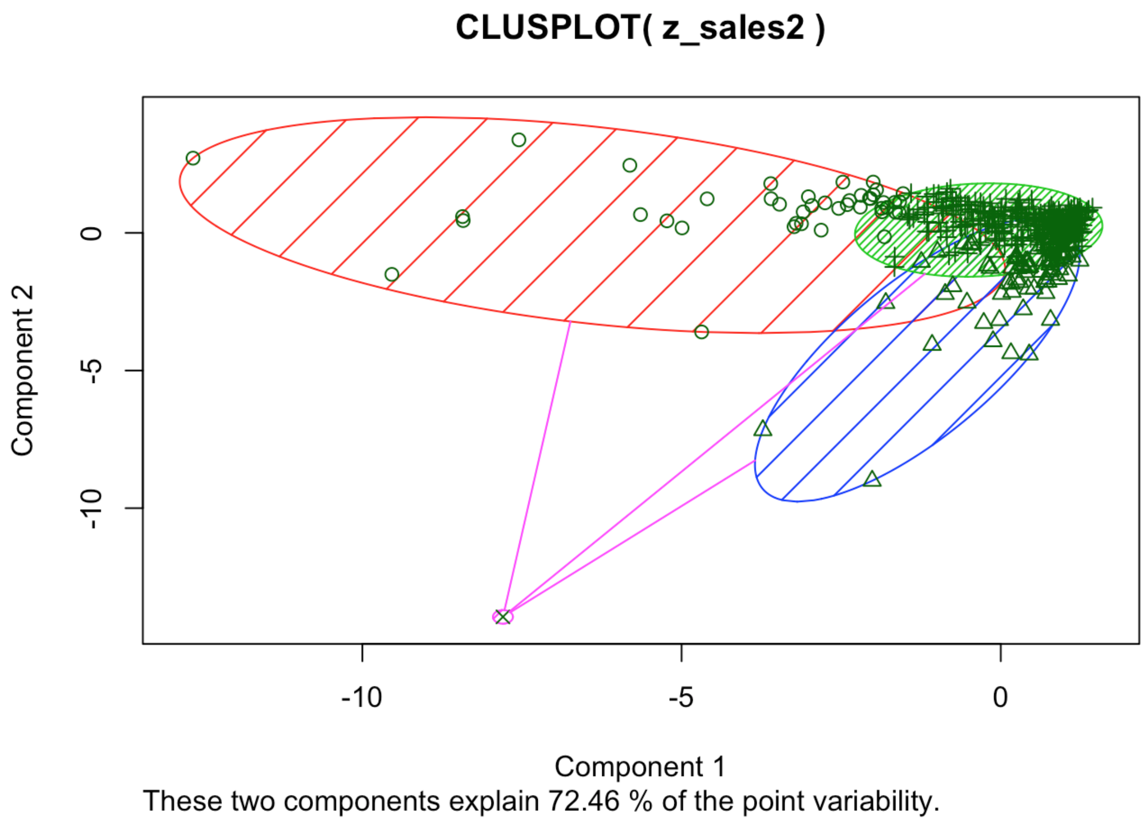
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.000   2.000   3.000   2.652   3.000   4.000
```

```
k_cluster4r$centers
```

```
##        Fresh       Milk    Grocery     Frozen Detergents_Paper Delicassen
## 1 -0.2913361  1.8504713  2.2226825 -0.2419393        2.2545210  0.2606580
## 2  1.5977702 -0.1216861 -0.2255888  1.0338481       -0.4010576  0.3381225
## 3 -0.2990412 -0.2331257 -0.2485398 -0.2036312       -0.2107264 -0.1544522
## 4  1.9645810  5.1696185  1.2857533  6.8927538       -0.5542311 16.4597113
```

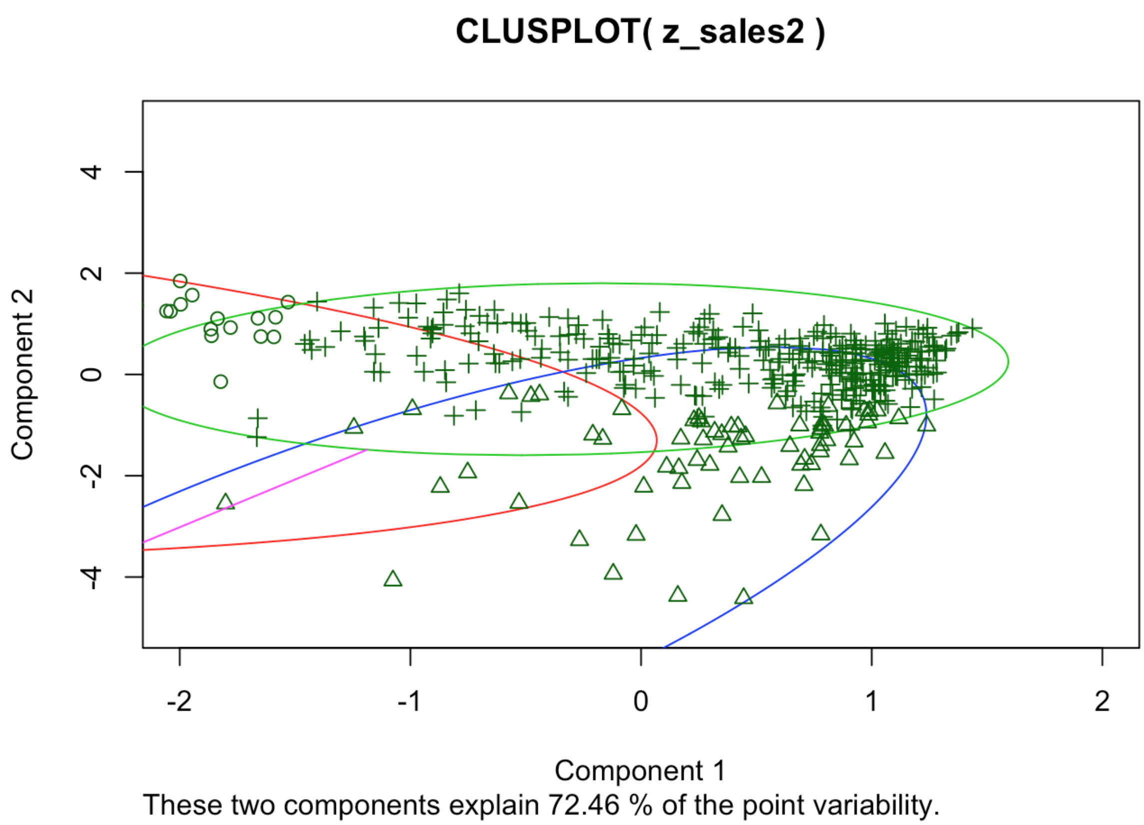
Visualize Cluster:

```
clusplot(z_sales2, k_cluster4r$cluster, color = T, shade = T)
```



Okay, this looks even better in my opinon, lets zoom in on the more dense area of the multidimensional space to determine how accurate this cluster analysis really is.

```
clusplot(z_sales2, k_cluster4r$cluster, color = TRUE,
         ylim = c(-5,5), xlim = c(-2,2))
```



We don't see quite a much overlap,when looking at things graphically, but to me the most exciting thing about this output is that you begin to see who your customers are more clearly. I have created pseudo-names for these segments to more properly understand who these wholesale buyers really are.

```
cluster_names <- c("One Stop Big Retailer",
                   "Small Stores",
                   "Boutique Shops",
                   "Pharmacy and Convenience")

output_table <- as.data.frame(cbind(cluster_names,k_cluster4r$centers))

output_table
```

```
##              cluster_names              Fresh               Milk
## 1    One Stop Big Retailer -0.291336087483456   1.85047127380293
## 2             Small Stores   1.59777024996465 -0.121686116747129
## 3           Boutique Shops  -0.29904117761657 -0.233125692974804
## 4 Pharmacy and Convenience   1.96458102242732   5.16961846101418
##              Grocery             Frozen   Detergents_Paper
## 1   2.22268246951446 -0.241939341512277     2.254520951901
## 2 -0.225588756403413   1.03384806938726 -0.401057620190001
## 3  -0.24853982935481  -0.20363119164084 -0.210726377030017
## 4     1.285753274688   6.89275382489014 -0.554231092977207
##           Delicassen
## 1  0.260657950499663
## 2  0.338122514308769
## 3 -0.154452238224764
## 4   16.4597112932408
```