Untitled

Yatharth Malik

March 22, 2017

Kmeans on “Wholesale customer data”

Loading required libraries

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

Exploring data

```
data = read.csv("Wholesale customers data.csv")
str(data)
```

```
## 'data.frame':    440 obs. of  8 variables:
##  $ Channel         : int  2 2 2 1 2 2 2 2 1 2 ...
##  $ Region          : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ Fresh           : int  12669 7057 6353 13265 22615 9413 12126 7579 5963 6006 ...
##  $ Milk            : int  9656 9810 8808 1196 5410 8259 3199 4956 3648 11093 ...
##  $ Grocery         : int  7561 9568 7684 4221 7198 5126 6975 9426 6192 18881 ...
##  $ Frozen          : int  214 1762 2405 6404 3915 666 480 1669 425 1159 ...
##  $ Detergents_Paper: int  2674 3293 3516 507 1777 1795 3140 3321 1716 7425 ...
##  $ Delicassen      : int  1338 1776 7844 1788 5185 1451 545 2566 750 2098 ...
```

```
summary(data)
```

```
##     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
```

Data preprocessing

Since the data contains outliers, we will remove top customer from each category

```
top.n.custs <- function (data,cols,n=5)
{
    idx.to.remove <-integer(0)
    for (c in cols)
    {
        col.order <-order(data[,c],decreasing=T) #
        idx <-head(col.order, n) #
        idx.to.remove <-union(idx.to.remove,idx)
    }
    return(idx.to.remove)
}
top.custs <-top.n.custs(data,cols=3:8,n=5)
length(top.custs)
```

```
## [1] 19
```

```
data[top.custs,]
```

```
##     Channel Region  Fresh  Milk Grocery Frozen Detergents_Paper Delicassen
## 182       1      3 112151 29627   18148  16745             4948       8550
## 126       1      3  76237  3473    7102  16538              778        918
## 285       1      3  68951  4411   12609   8692              751       2406
## 40        1      3  56159   555     902  10002              212       2916
## 259       1      1  56083  4563    2124   6422              730       3321
## 87        2      3  22925 73498   32114    987            20070        903
## 48        2      3  44466 54259   55571   7782            24171       6465
## 86        2      3  16117 46197   92780   1026            40827       2944
## 184       1      3  36847 43950   20170  36534              239      47943
## 62        2      3  35942 38369   59598   3254            26701       2017
## 334       2      2   8565  4980   67298    131            38102       1215
## 66        2      3     85 20959   45828     36            24231       1423
## 326       1      2  32717 16784   13626  60869             1272       5609
## 94        1      3  11314  3090    2062  35009               71       2698
## 197       1      1  30624  7209    4897  18711              763       2876
## 104       1      3  56082  3504    8906  18028             1480       2498
## 24        2      3  26373 36423   22019   5154             4337      16523
## 72        1      3  18291  1266   21042   5373             4173      14472
## 88        1      3  43265  5025    8117   6312             1579      14351
```

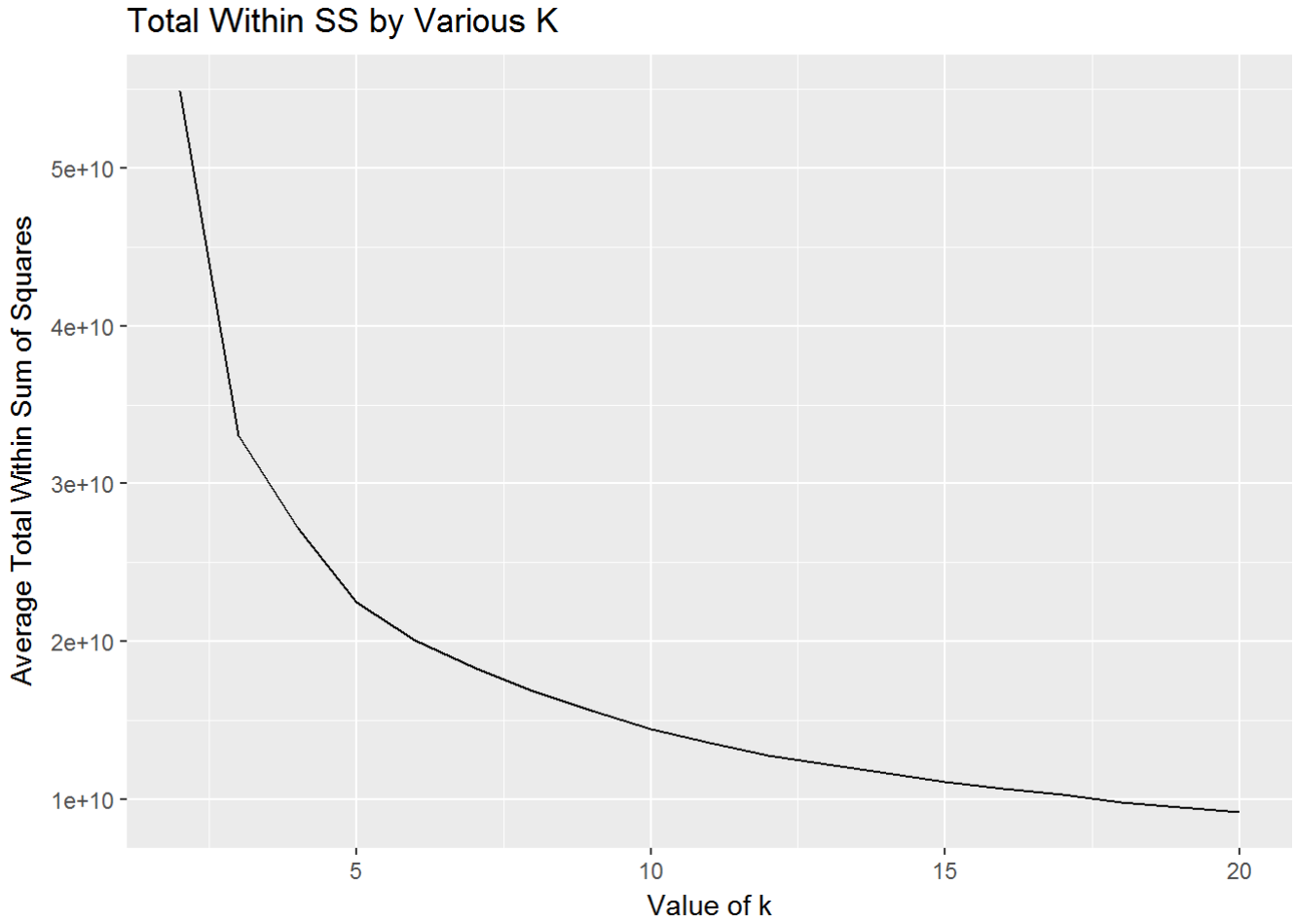
```
data.rm.top<-data[-c(top.custs),]
```

Finding appropriate value of k using “Elbow Method”

```
rng<-2:20
tries <-100
avg.totw.ss <-integer(length(rng))
for(v in rng)
{
    v.totw.ss <-integer(tries)
    for(i in 1:tries)
    {
        k.temp <-kmeans(data.rm.top,centers=v)
        v.totw.ss[i] <-k.temp$tot.withinss
    }
    avg.totw.ss[v-1] <-mean(v.totw.ss)
}
```

```
## Warning: did not converge in 10 iterations
```

```
df = data.frame(x = rng,y = avg.totw.ss)
ggplot(df,aes(x = x,y = y)) + geom_line() + xlab("Value of k") + ylab("Average Total Within Sum
 of Squares") + ggtitle("Total Within SS by Various K")
```



By observing the above plot, we choose optimal value of k = 5

Applying kmeans on dataset with k = 5

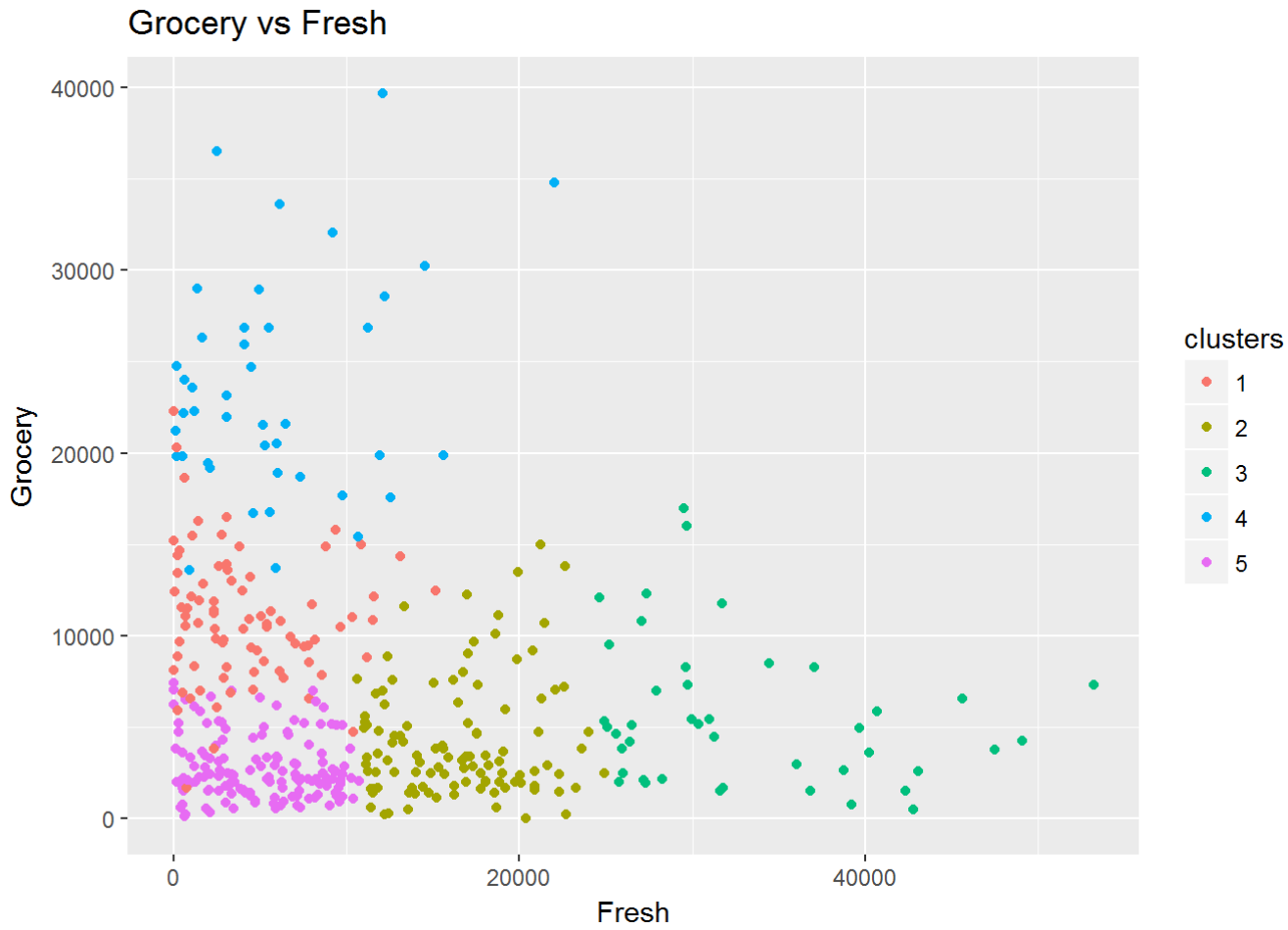
```
set.seed(76964057)
km = kmeans(data.rm.top[-c(1,2)],centers = 5)

km$centers
```

```
##       Fresh      Milk   Grocery   Frozen Detergents_Paper Delicassen
## 1  4189.747  7645.639 11015.277 1335.145        4750.4819  1387.1205
## 2 16470.870  3026.491  4264.741 3217.306         996.5556  1319.7593
## 3 33120.163  4896.977  5579.860 3823.372         945.4651  1620.1860
## 4  5830.214 15295.048 23449.167 1936.452       10361.6429  1912.7381
## 5  5043.434  2329.683  2786.138 2689.814         652.8276   849.8414
```

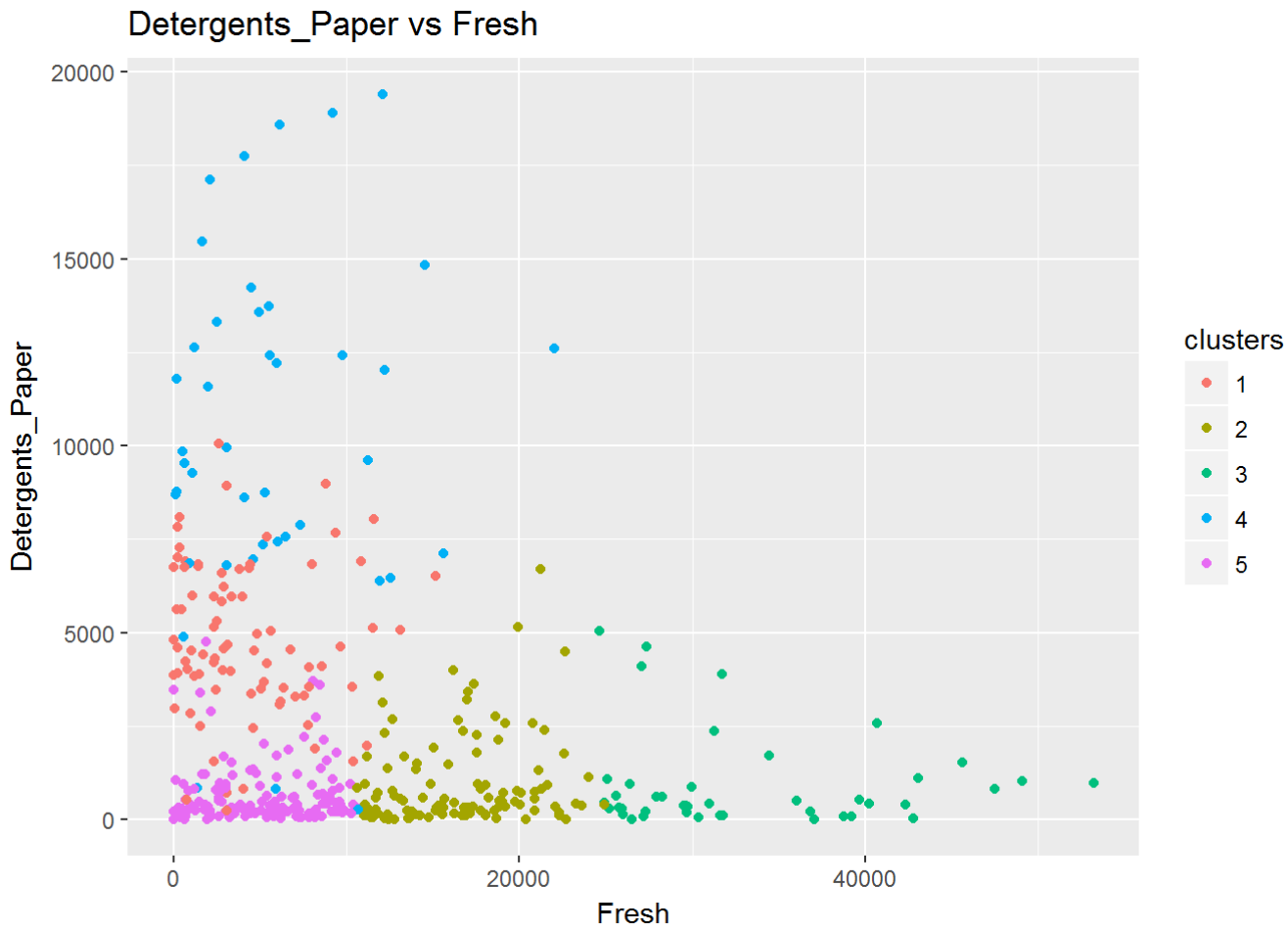
```
clusters = as.factor(km$cluster)
```

Visualizing the cluster

```
ggplot(data.rm.top,aes(x= Fresh,y = Grocery,color = clusters)) + geom_point() + ggtitle("Grocery
 vs Fresh")
```




```
ggplot(data.rm.top,aes(x= Fresh,y = Detergents_Paper,color = clusters)) + geom_point() +
ggtitle("Detergents_Paper vs Fresh")
```


Detergents_Paper vs Fresh
Detergents_Paper
20000
15000
10000
5000
0
0
20000
40000
Fresh
clusters
1
2
3
4
5