Assignment 5 - Fundamentals of Machine Learning

Julia Thacker

11/26/2021

```
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

library(ggplot2)
library(cluster)
library(fpc)
cereals.df<-read.csv("Cereals.csv",row.names=1)
cereals.df<-na.omit(cereals.df)
```

Read the file, labeled the rows by cereal names, and then removed the cereal name column. Also removed any cereals with missing values.

```
cereals.df$mfr=as.numeric(as.factor(as.character(cereals.df$mfr)))
cereals.df$type=as.numeric(as.factor(as.character(cereals.df$type)))
cereals.df$shelf=as.numeric(as.character(cereals.df$shelf))
```

Converted any categorical variables into numeric variables.

```
d<-dist(cereals.df,method = "euclidean")
summary(d)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.713  84.496 132.847 143.235 195.743 402.260
```

Computed Euclidean distance on the initial data.

```
cereals.df.norm<-sapply(cereals.df,scale)
row.names(cereals.df.norm)<-row.names(cereals.df)
head(cereals.df.norm)

##                                  mfr       type   calories    protein
## 100%_Bran                  0.2067288 -0.1162476 -1.8659155  1.3817478
## 100%_Natural_Bran          1.3834926 -0.1162476  0.6537514  0.4522084
## All-Bran                  -0.3816531 -0.1162476 -1.8659155  1.3817478
## All-Bran_with_Extra_Fiber -0.3816531 -0.1162476 -2.8737823  1.3817478
## Apple_Cinnamon_Cheerios   -0.9700351 -0.1162476  0.1498180 -0.4773310
```

```
## Apple_Jacks                 -0.3816531 -0.1162476  0.1498180 -0.4773310
##                                    fat     sodium       fiber      carbo
## 100%_Bran                    0.0000000 -0.3910227  3.22866747 -2.5001396
## 100%_Natural_Bran            3.9728810 -1.7804186 -0.07249167 -1.7292632
## All-Bran                     0.0000000  1.1795987  2.81602258 -1.9862220
## All-Bran_with_Extra_Fiber -0.9932203 -0.2702057  4.87924705 -1.7292632
## Apple_Cinnamon_Cheerios      0.9932203  0.2130625 -0.27881412 -1.0868662
## Apple_Jacks               -0.9932203 -0.4514312 -0.48513656 -0.9583868
##                                 sugars     potass    vitamins      shelf
## 100%_Bran                   -0.2542051  2.5605229 -0.1818422  0.9419715
## 100%_Natural_Bran            0.2046041  0.5147738 -1.3032024  0.9419715
## All-Bran                    -0.4836096  3.1248675 -0.1818422  0.9419715
## All-Bran_with_Extra_Fiber -1.6306324  3.2659536 -0.1818422  0.9419715
## Apple_Cinnamon_Cheerios      0.6634132 -0.4022862 -0.1818422 -1.4616799
## Apple_Jacks                  1.5810314 -0.9666308 -0.1818422 -0.2598542
##                                 weight       cups     rating
## 100%_Bran                   -0.2008324 -2.0856582  1.8549038
## 100%_Natural_Bran           -0.2008324  0.7567534 -0.5977113
## All-Bran                    -0.2008324 -2.0856582  1.2151965
## All-Bran_with_Extra_Fiber -0.2008324 -1.3644493  3.6578436
## Apple_Cinnamon_Cheerios     -0.2008324 -0.3038480 -0.9165248
## Apple_Jacks                 -0.2008324  0.7567534 -0.6553998
```

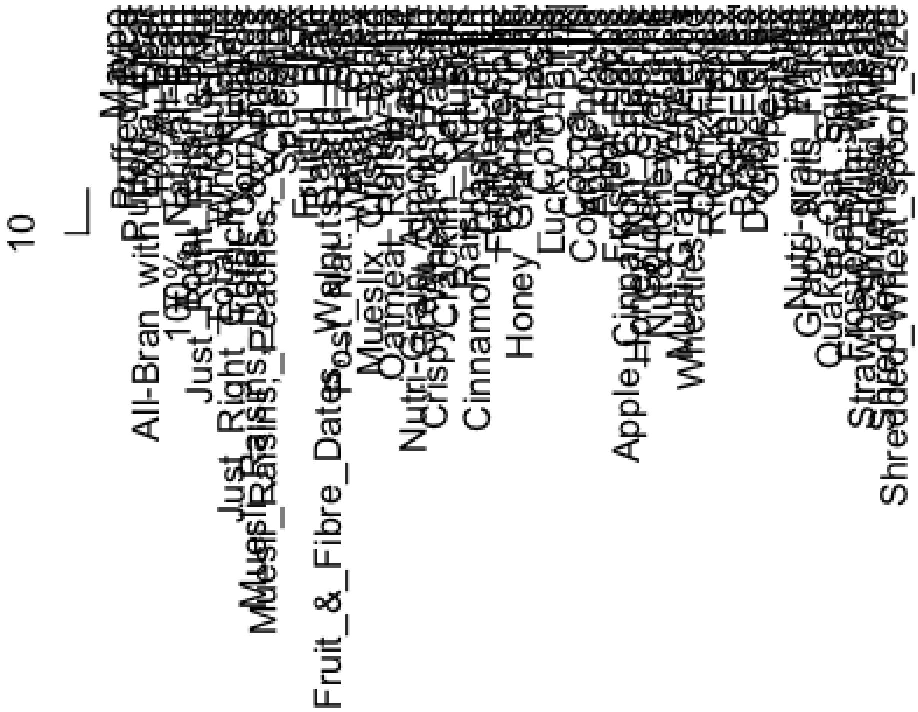
Normalized the data set.

```
d.norm<-dist(cereals.df.norm,method = "euclidean")
summary(d.norm)
```

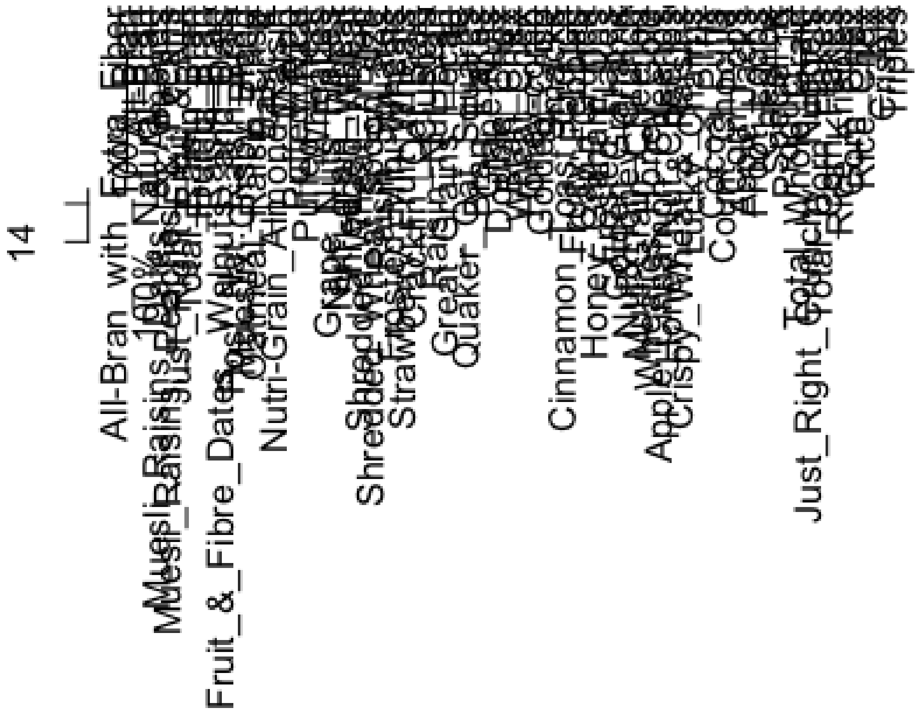
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.1431  4.0698  4.9765  5.1731  6.0529 12.1761
```

Computed Euclidean distance on the normalized data.

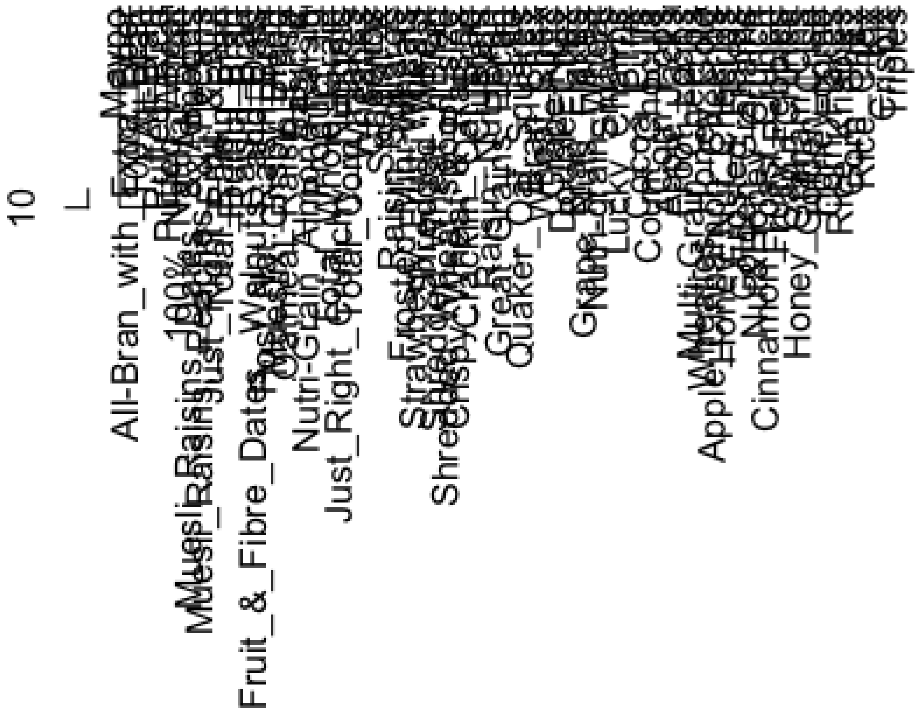
```
hc1<-hclust(d.norm,method = "single")
plot(hc1,hang=-1,ann=FALSE)
```



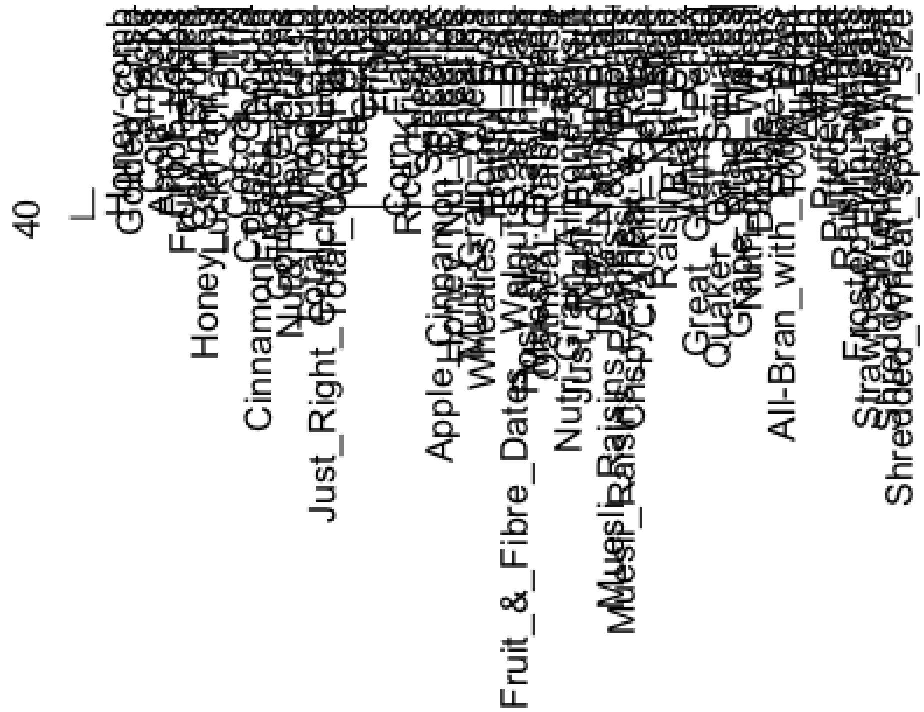

```r
hc2<-hclust(d.norm,method = "complete")
plot(hc2,hang=-1,ann=FALSE)
```



```r
hc3<-hclust(d.norm,method = "average")
plot(hc3,hang=-1,ann=FALSE)
```



```
hc4<-hclust(d.norm,method = "ward.D")
plot(hc4,hang=-1,ann=FALSE)
```



Computed and plotted the normalized data using single linkage, complete linkage, average linkage, and Ward.

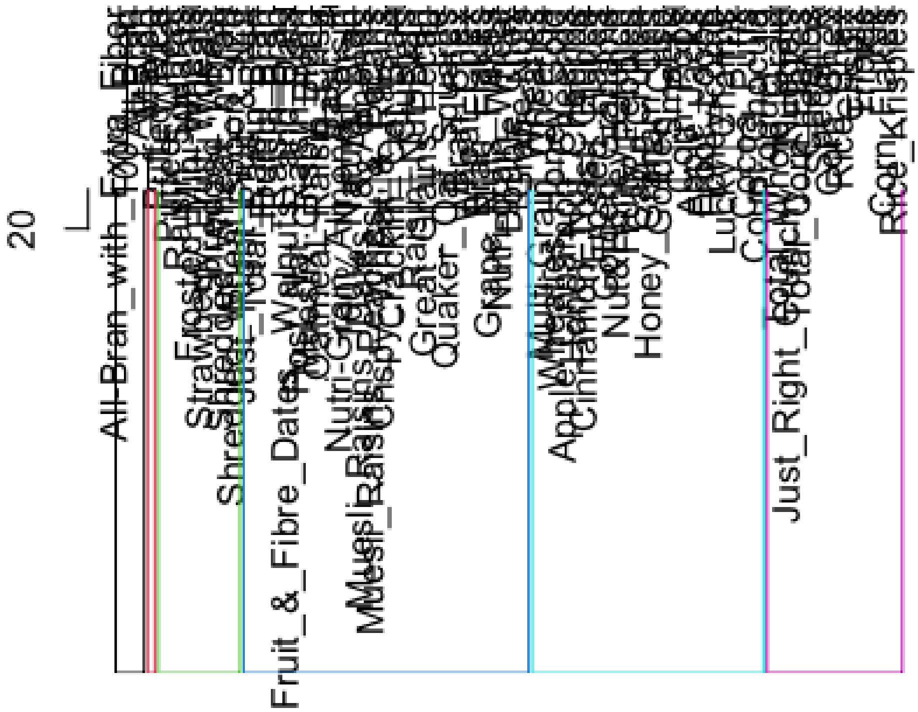
```
hc1b<-agnes(cereals.df.norm,method="single")
print(hc1b$ac)

## [1] 0.7994528

hc2b<-agnes(cereals.df.norm,method="complete")
print(hc2b$ac)

## [1] 0.8351367

hc3b<-agnes(cereals.df.norm,method="average")
print(hc3b$ac)

## [1] 0.8045719

hc4b<-agnes(cereals.df.norm,method="ward")
print(hc4b$ac)

## [1] 0.8891952
```

Used Agnes to compare single linkage, complete linkage, average linkage, and Ward. Ward has the highest Agglomerative coefficient, so this would be the best linkage method.

```
hc4<-hclust(d.norm,method = "ward.D2")
plot(hc4,hang=-1,ann=FALSE)
```

Plotted the data using the chosen method, Ward. Based on this plot, I would choose to separate the data into six clusters.

```
plot(hc4,hang=-1,ann=FALSE)
rect.hclust(hc4,k=6,border=1:6)
```



Plotted the data with visualization of the 6 clusters.

```
clusters1<-cutree(hc4,k=6)
clusters1

##                                100%_Bran
100%_Natural_Bran
##                                        1
2
##                                 All-Bran              All-
Bran_with_Extra_Fiber
##                                        1
1
##                  Apple_Cinnamon_Cheerios
Apple_Jacks
##                                        3
3
##                                  Basic_4
Bran_Chex
##                                        2
2
##                              Bran_Flakes
Cap'n'Crunch
##                                        2
```

```
3
##                               Cheerios
Cinnamon_Toast_Crunch
##                                      4
3
##                               Clusters
Cocoa_Puffs
##                                      2
3
##                              Corn_Chex
Corn_Flakes
##                                      4
4
##                              Corn_Pops
Count_Chocula
##                                      3
3
##                    Cracklin'_Oat_Bran
Crispix
##                                      2
4
##                Crispy_Wheat_&_Raisins
Double_Chex
##                                      2
2
##                            Froot_Loops
Frosted_Flakes
##                                      3
3
##                   Frosted_Mini-Wheats
Fruit_&_Fibre_Dates,_Walnuts,_and_Oats
##                                      5
2
##                          Fruitful_Bran
Fruity_Pebbles
##                                      2
3
##                           Golden_Crisp
Golden_Grahams
##                                      3
3
##                      Grape_Nuts_Flakes                              Grape-
Nuts
##                                      2
2
##                     Great_Grains_Pecan
Honey_Graham_Ohs
##                                      2
3
##                     Honey_Nut_Cheerios                              Honey-
```

```
comb
##                                       3
3
##             Just_Right_Crunchy__Nuggets
Just_Right_Fruit_&_Nut
##                                       4
2
##                                     Kix
Life
##                                       4
2
##                            Lucky_Charms
Maypo
##                                       3
6
##        Muesli_Raisins,_Dates,_&_Almonds
Muesli_Raisins,_Peaches,_&_Pecans
##                                       2
2
##                     Mueslix_Crispy_Blend                   Multi-
Grain_Cheerios
##                                       2
3
##                        Nut&Honey_Crunch              Nutri-Grain_Almond-
Raisin
##                                       3
2
##                         Nutri-grain_Wheat
Oatmeal_Raisin_Crisp
##                                       2
2
##                   Post_Nat._Raisin_Bran
Product_19
##                                       2
4
##                             Puffed_Rice
Puffed_Wheat
##                                       5
5
##                      Quaker_Oat_Squares
Raisin_Bran
##                                       2
2
##                          Raisin_Nut_Bran
Raisin_Squares
##                                       2
5
##                               Rice_Chex
Rice_Krispies
##                                       4
```

```
4
##                          Shredded_Wheat
Shredded_Wheat_'n'Bran
##                                       5
5
##                Shredded_Wheat_spoon_size
Smacks
##                                       5
3
##                               Special_K
Strawberry_Fruit_Wheats
##                                       4
5
##                       Total_Corn_Flakes
Total_Raisin_Bran
##                                       4
2
##                       Total_Whole_Grain
Triples
##                                       4
4
##                                    Trix
Wheat_Chex
##                                       3
2
##                                Wheaties
Wheaties_Honey_Gold
##                                       3
3
```

Each cereal was assigned a cluster number.

```
cerealshcclusters<-cbind(clusters1,cereals.df.norm)
```

Combined the cluster number with the original normalized data set.

```
set.seed(123)
trainindex<-createDataPartition(y=cereals.df.norm[,1],p=0.5)[[1]]
partitionA<-cereals.df.norm[trainindex,]
partitionB<-cereals.df.norm[-trainindex,]
```

Partitioned the data into set A and set B.

```
kA<-kmeans(partitionA,6)
kA

## K-means clustering with 6 clusters of sizes 6, 5, 4, 8, 2, 13
##
## Cluster means:
##            mfr       type     calories     protein        fat      sodium
fiber
```

```
## 1  0.3047924 -0.1162476 -0.01815976 -0.6322543 -0.8276836  1.0386455 -
0.6914590
## 2 -0.3816531 -0.1162476  1.35925815  0.4522084  0.3972881  0.2613893
0.7115336
## 3  0.7951107 -0.1162476 -1.99189884 -0.4773310 -0.9932203 -1.9616441 -
0.1756529
## 4  0.2067288 -0.1162476 -0.66907371  0.8007856 -0.1241525 -0.4740844
0.8043787
## 5 -0.9700351 -0.1162476 -0.10214866  1.8465175  0.4966101  0.9983732
0.1338308
## 6 -0.3363930 -0.1162476  0.14981803 -0.9778523 -0.1528031  0.1433603 -
0.6597171
##          carbo     sugars     potass   vitamins      shelf     weight
## 1  1.611201055 -0.9424187 -0.8020303 -0.1818422 -0.4601585 -0.2008324
## 2 -0.161814589  1.2598650  1.4600510  0.4909739  0.7016064  2.2891374
## 3  0.005208624 -1.6306324 -0.4022862 -1.3032024 -0.2598542 -2.1074193
## 4 -0.669308197 -0.3115562  0.7440388 -0.1818422  0.6415151 -0.2008324
## 5  0.454886505 -1.1718233  0.1267869  1.5001982 -0.2598542 -0.2008324
## 6 -0.513650469  0.9634038 -0.7767071 -0.1818422 -0.7220949 -0.2008324
##         cups      rating
## 1  1.0254391  0.05533471
## 2 -0.4396049 -0.64704425
## 3  0.4067550  1.69791638
## 4 -1.4280854  0.81182749
## 5  1.2870541  0.45177886
## 6  0.4173610 -0.93675271
## 
## Clustering vector:
##                 100%_Bran                  All-Bran
Apple_Jacks
##                         4                         4
6
##              Cap'n'Crunch                  Cheerios
Cinnamon_Toast_Crunch
##                         6                         5
6
##                  Clusters               Cocoa_Puffs
Corn_Flakes
##                         4                         6
1
##                 Corn_Pops                   Crispix
Double_Chex
##                         6                         1
1
##               Froot_Loops            Frosted_Flakes       Frosted_Mini-
Wheats
##                         6                         6
4
##              Golden_Crisp            Golden_Grahams                Grape-
Nuts
```

```
##                        6                          6
4
##        Honey_Nut_Cheerios                 Honey-comb
Kix
##                        6                          6
1
##                     Life      Mueslix_Crispy_Blend      Multi-
Grain_Cheerios
##                        4                          2
6
##      Oatmeal_Raisin_Crisp     Post_Nat._Raisin_Bran
Puffed_Rice
##                        2                          2
3
##              Puffed_Wheat        Quaker_Oat_Squares
Raisin_Bran
##                        3                          4
2
##            Raisin_Squares                 Rice_Chex
Rice_Krispies
##                        4                          1
1
##            Shredded_Wheat Shredded_Wheat_spoon_size
Total_Raisin_Bran
##                        3                          3
2
##         Total_Whole_Grain                      Trix
##                        5                          6
##
## Within cluster sum of squares by cluster:
## [1] 26.19969 28.41539 28.04241 68.78449 14.47801 51.71672
##  (between_SS / total_SS =  60.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
kA$centers
```

```
##          mfr       type    calories    protein        fat     sodium
fiber
## 1  0.3047924 -0.1162476 -0.01815976 -0.6322543 -0.8276836  1.0386455 -
0.6914590
## 2 -0.3816531 -0.1162476  1.35925815  0.4522084  0.3972881  0.2613893
0.7115336
## 3  0.7951107 -0.1162476 -1.99189884 -0.4773310 -0.9932203 -1.9616441 -
0.1756529
## 4  0.2067288 -0.1162476 -0.66907371  0.8007856 -0.1241525 -0.4740844
```

```
0.8043787
## 5 -0.9700351 -0.1162476 -0.10214866  1.8465175  0.4966101  0.9983732
0.1338308
## 6 -0.3363930 -0.1162476  0.14981803 -0.9778523 -0.1528031  0.1433603 -
0.6597171
##          carbo     sugars     potass   vitamins      shelf     weight
## 1  1.611201055 -0.9424187 -0.8020303 -0.1818422 -0.4601585 -0.2008324
## 2 -0.161814589  1.2598650  1.4600510  0.4909739  0.7016064  2.2891374
## 3  0.005208624 -1.6306324 -0.4022862 -1.3032024 -0.2598542 -2.1074193
## 4 -0.669308197 -0.3115562  0.7440388 -0.1818422  0.6415151 -0.2008324
## 5  0.454886505 -1.1718233  0.1267869  1.5001982 -0.2598542 -0.2008324
## 6 -0.513650469  0.9634038 -0.7767071 -0.1818422 -0.7220949 -0.2008324
##         cups      rating
## 1  1.0254391  0.05533471
## 2 -0.4396049 -0.64704425
## 3  0.4067550  1.69791638
## 4 -1.4280854  0.81182749
## 5  1.2870541  0.45177886
## 6  0.4173610 -0.93675271
```

```
partionaclusters<-cbind(partitionA,"Cluster_Number"=kA$cluster)
dist(kA$centers)
```

```
##          1        2        3        4        5
## 2 5.600918
## 3 4.968037 7.917139
## 4 4.827270 4.290047 4.585193
## 5 3.944669 5.071038 6.138305 4.396710
## 6 3.374490 4.518780 5.538560 4.351989 4.718626
```

Found the centers of data partition A.

```
kB<-kmeans(partitionB,kA$centers)
kB
```

```
## K-means clustering with 6 clusters of sizes 10, 10, 1, 1, 5, 9
##
## Cluster means:
##          mfr       type   calories    protein        fat      sodium
fiber
## 1  0.6185961 -0.1162476 -0.5052954  0.4522084 -0.5959322  0.06204118
0.2988887
## 2  0.3832434 -0.1162476  0.9561114  0.6381162  1.4898304 -0.21583799
0.4433144
## 3 -0.3816531 -0.1162476 -2.8737823  1.3817478 -0.9932203 -0.27020566
4.8792470
## 4 -1.5584170  8.4860776 -0.3541153  1.3817478  0.0000000 -1.96164410 -
0.8977815
## 5 -0.6170059 -0.1162476  0.3513914 -0.1055153 -0.1986441  0.72049400 -
0.5676655
## 6 -0.3816531 -0.1162476  0.2058106 -0.7871775  0.2207156  0.04526104 -
```

```
0.5997601
##        carbo     sugars      potass   vitamins      shelf     weight
## 1  0.5062783 -0.8277164  0.05624380 -0.2939782 -0.6204019 -0.2008324
## 2 -0.2260543  0.3422468  0.71934869 -0.2939782  0.9419715  0.6074300
## 3 -1.7292632 -1.6306324  3.26595362 -0.1818422  0.9419715 -0.2008324
## 4  0.3264071 -0.9424187 -0.04957081 -0.1818422 -0.2598542 -0.2008324
## 5  1.3028505 -0.5294905 -0.55748093  2.5094224  0.9419715  0.1902623
## 6 -0.6300506  0.9183071 -0.57472480 -0.1818422 -0.3933904 -0.2008324
##          cups     rating
## 1  0.14584702  0.8567145
## 2 -0.47778658 -0.2339385
## 3 -1.36444931  3.6578436
## 4  0.75675340  0.8892251
## 5  0.33251286 -0.2766806
## 6  0.01197556 -0.9444746
##
## Clustering vector:
##                      100%_Natural_Bran              All-
Bran_with_Extra_Fiber
##                                      2
3
##                  Apple_Cinnamon_Cheerios
Basic_4
##                                      6
2
##                              Bran_Chex
Bran_Flakes
##                                      1
1
##                              Corn_Chex
Count_Chocula
##                                      1
6
##                      Cracklin'_Oat_Bran
Crispy_Wheat_&_Raisins
##                                      2
6
## Fruit_&_Fibre_Dates,_Walnuts,_and_Oats
Fruitful_Bran
##                                      2
2
##                          Fruity_Pebbles
Grape_Nuts_Flakes
##                                      6
1
##                      Great_Grains_Pecan
Honey_Graham_Ohs
##                                      2
6
##              Just_Right_Crunchy__Nuggets
```

```
Just_Right_Fruit_&_Nut
##                                       5
5
##                            Lucky_Charms
Maypo
##                                       6
4
##         Muesli_Raisins,_Dates,_&_Almonds
Muesli_Raisins,_Peaches,_&_Pecans
##                                       2
2
##                        Nut&Honey_Crunch              Nutri-Grain_Almond-
Raisin
##                                       6
2
##                       Nutri-grain_Wheat
Product_19
##                                       1
5
##                         Raisin_Nut_Bran
Shredded_Wheat_'n'Bran
##                                       2
1
##                                  Smacks
Special_K
##                                       6
1
##                  Strawberry_Fruit_Wheats
Total_Corn_Flakes
##                                       1
5
##                                 Triples
Wheat_Chex
##                                       5
1
##                                Wheaties
Wheaties_Honey_Gold
##                                       1
6
##
## Within cluster sum of squares by cluster:
## [1] 68.82790 76.67667  0.00000  0.00000 23.62327 26.78758
##  (between_SS / total_SS =  63.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
kB$centers
```

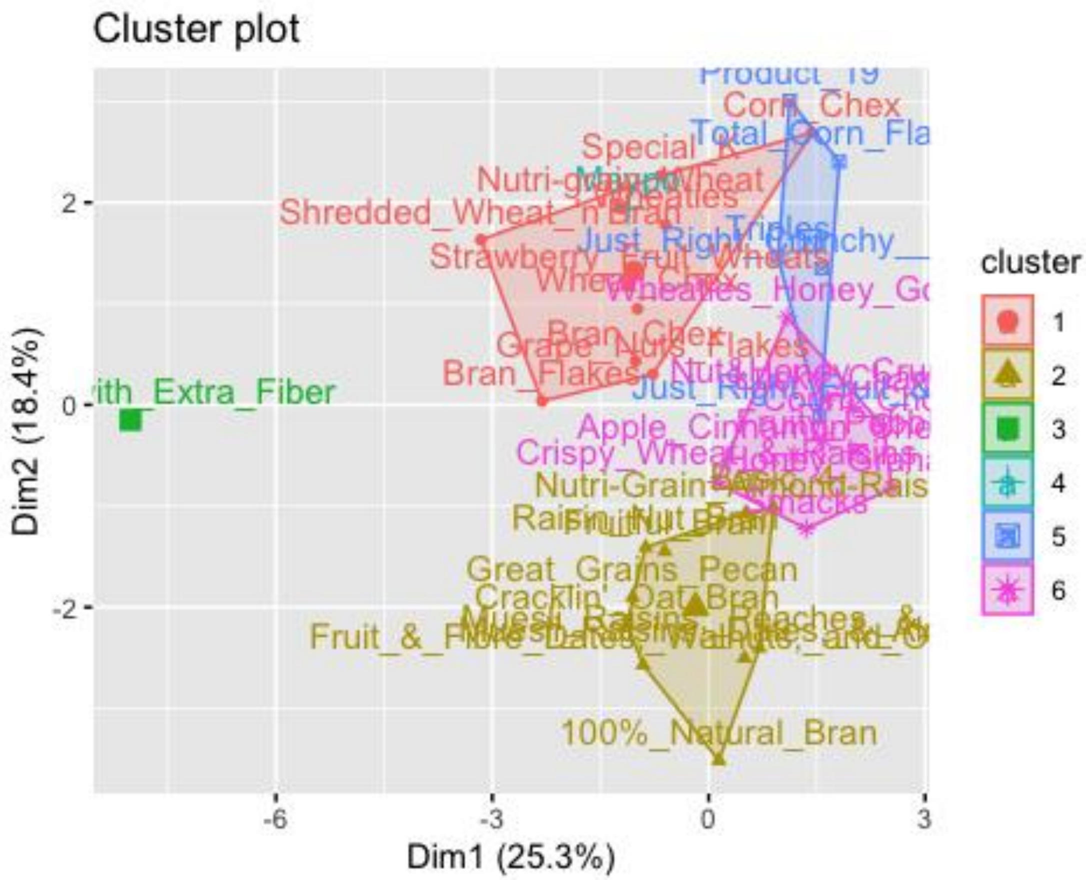
```
##          mfr       type   calories    protein        fat      sodium
fiber
## 1  0.6185961 -0.1162476 -0.5052954  0.4522084 -0.5959322  0.06204118
0.2988887
## 2  0.3832434 -0.1162476  0.9561114  0.6381162  1.4898304 -0.21583799
0.4433144
## 3 -0.3816531 -0.1162476 -2.8737823  1.3817478 -0.9932203 -0.27020566
4.8792470
## 4 -1.5584170  8.4860776 -0.3541153  1.3817478  0.0000000 -1.96164410 -
0.8977815
## 5 -0.6170059 -0.1162476  0.3513914 -0.1055153 -0.1986441  0.72049400 -
0.5676655
## 6 -0.3816531 -0.1162476  0.2058106 -0.7871775  0.2207156  0.04526104 -
0.5997601
##        carbo     sugars      potass   vitamins      shelf     weight
## 1  0.5062783 -0.8277164  0.05624380 -0.2939782 -0.6204019 -0.2008324
## 2 -0.2260543  0.3422468  0.71934869 -0.2939782  0.9419715  0.6074300
## 3 -1.7292632 -1.6306324  3.26595362 -0.1818422  0.9419715 -0.2008324
## 4  0.3264071 -0.9424187 -0.04957081 -0.1818422 -0.2598542 -0.2008324
## 5  1.3028505 -0.5294905 -0.55748093  2.5094224  0.9419715  0.1902623
## 6 -0.6300506  0.9183071 -0.57472480 -0.1818422 -0.3933904 -0.2008324
##          cups     rating
## 1  0.14584702  0.8567145
## 2 -0.47778658 -0.2339385
## 3 -1.36444931  3.6578436
## 4  0.75675340  0.8892251
## 5  0.33251286 -0.2766806
## 6  0.01197556 -0.9444746
```

```
partionbclusters<-cbind(partitionB,"Cluster_Number"=kB$cluster)
dist(kB$centers)
```

```
##           1         2         3         4         5
## 2  3.700232
## 3  7.566217  8.419455
## 4  9.278054  9.723597 12.176053
## 5  4.097141  4.481984  9.733020  9.833602
## 6  3.547011  3.363001  9.648629  9.651715  4.090764
```

Assigned the records in partition B a cluster based on the centroids from partition A.

```
fviz_cluster(kB,data=partitionB)
```



Maypo was in its own cluster in both the partitioned data and the original data set with all cereals. All Bran with extra fiber was in its own cluster in the partioned data set, but was in a cluster of 3 initially. The clusters overall in partition A had fairly even distribution amongst all clusters, except one cluster of two. Partition B had two clusters of only one cereal, and the original distribution resulted in one cluster of one and one cluster of 3. Many of the cereals did stay grouped together similar to how they were grouped in the first 6 clusters with all the data, but there was also a fair amount of separation and shift between cereals once separated into the partioned data sets.

```
hclust_stability=clusterboot(cereals.df.norm,clustermethod =
hclustCBI,method="ward.D2",k=6,count=FALSE)
hclust_stability

## * Cluster stability assessment *
## Cluster method:  hclust/cutree
## Full clustering results are given as parameter result
## of the clusterboot object, which also provides further statistics
## of the resampling results.
## Number of resampling runs:  100
##
## Number of clusters found in data:  6
##
##  Clusterwise Jaccard bootstrap (omitting multiple points) mean:
```

```
## [1] 0.8065814 0.6118701 0.8975055 0.7219622 0.5629313 0.6453571
## dissolved:
## [1] 21 38  0 22 58 36
## recovered:
## [1] 79 17 88 48 28 64
```

Summarized the stability of the clusters

```
clusters2=hclust_stability$result$partition
hclust_stability$bootmean

## [1] 0.8065814 0.6118701 0.8975055 0.7219622 0.5629313 0.6453571
```

For a goal of finding a cluster of “healthy cereals”, the data should still be normalized. This data set contains a variety of variables that are measured in different units on different scales. Normalizing the data allows each of these variables to be compared on the same scale. After normalizing the data, variables such as Calories, Protein, and Sugars, that contribute to how healthy or unhealthy the cereal is, could still be interpreted together and clustered appropriately. This would also allow all the data to be considered when determining whether the cereal is healthy or not.