# Homework 5 solution

```r
# hw5sol.r
#
# Question 1
#
library(cluster)
d1 = read.csv("universities.csv")
str(d1)
```

```
## 'data.frame':    25 obs. of  7 variables:
##  $ University: Factor w/ 25 levels "Brown","CalTech",..: 9 15 25 17 11 7 2 6 1 10 ...
##  $ SAT       : num  14 13.8 13.8 13.6 13.8 ...
##  $ Top10     : int  91 91 95 90 94 90 100 89 89 75 ...
##  $ Accept    : int  14 14 19 20 30 30 25 23 22 44 ...
##  $ SFRatio   : int  11 8 11 12 10 12 6 10 13 7 ...
##  $ Expenses  : num  39.5 30.2 43.5 36.5 34.9 ...
##  $ Grad      : int  97 95 96 93 91 95 81 95 94 87 ...
```

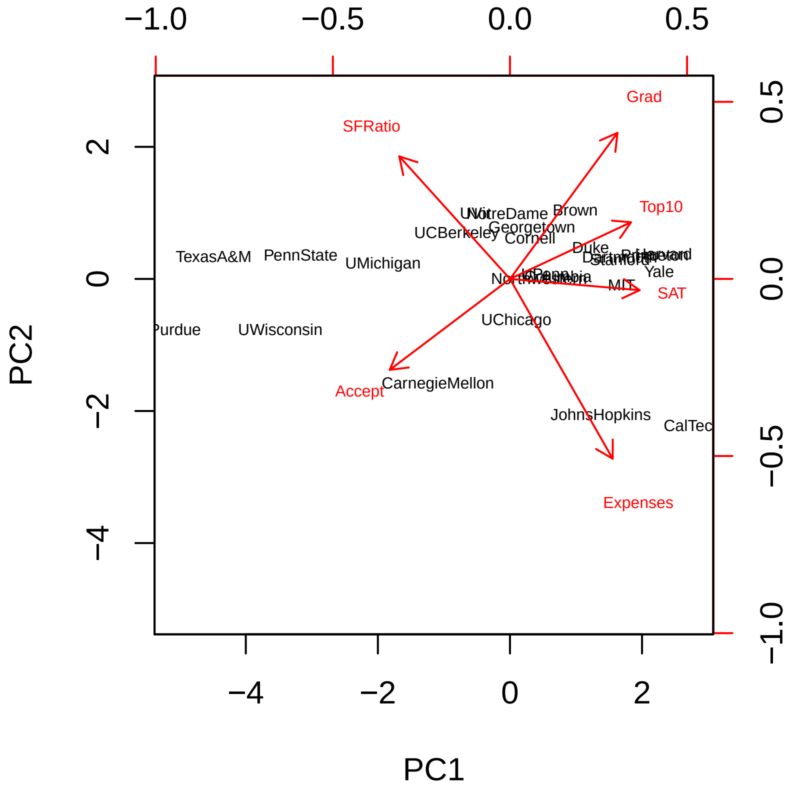
```r
head(d1)
```

```
##   University   SAT Top10 Accept SFRatio Expenses Grad
## 1    Harvard 14.00    91     14      11   39.525   97
## 2  Princeton 13.75    91     14       8   30.220   95
## 3       Yale 13.75    95     19      11   43.514   96
## 4   Stanford 13.60    90     20      12   36.450   93
## 5        MIT 13.80    94     30      10   34.870   91
## 6       Duke 13.15    90     30      12   31.585   95
```

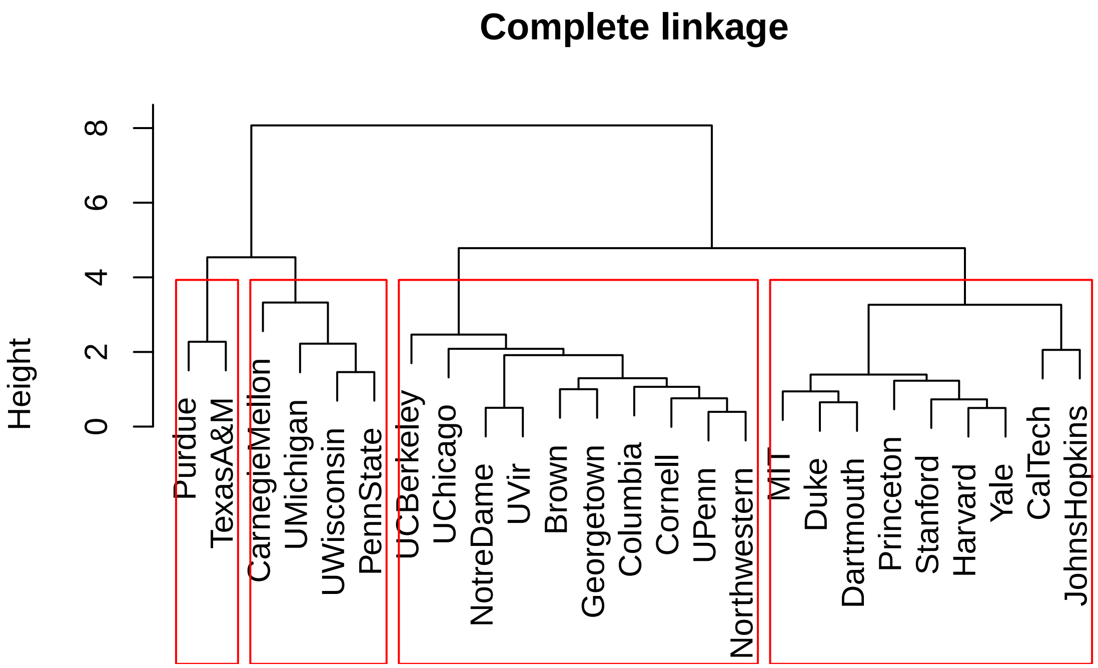
```r
#
# move the university name to the rownames
#
rownames(d1) = d1[,1]
d1$University = NULL
head(d1)
```

```
##             SAT Top10 Accept SFRatio Expenses Grad
## Harvard   14.00    91     14      11   39.525   97
## Princeton 13.75    91     14       8   30.220   95
## Yale      13.75    95     19      11   43.514   96
## Stanford  13.60    90     20      12   36.450   93
## MIT       13.80    94     30      10   34.870   91
## Duke      13.15    90     30      12   31.585   95
```

```r
#
# scaling the data
#
m1=prcomp(d1, scale=T)
#
# mirror image
#
m1$rotation = -m1$rotation
m1$x = -m1$x
biplot(m1,scale=0,cex=0.5)
```



```
#
# Clusters
# Expensive schools:  John Hopkins, CalTech
# Private schools are in the Top10 and High SAT range:
#                Princeton, Harvard, Stanford, Yale, Duke, Darmouth
# Schools with low  PC1: Texas AM, PennState, UM, UW, Purdue
# Schools with high PC2: Brown, UV, Georgetown, NotreDame, Cornell
# Average school (center of biplot): Northwestern, UPenn
#
# b) HClustering - complete
d2 = scale(d1)
distances = dist(d2)
hc1 = hclust(distances,method='complete')
plot(hc1,sub='',xlab='',main = '')
title('Complete linkage')
rect.hclust(hc1,k=4)
```



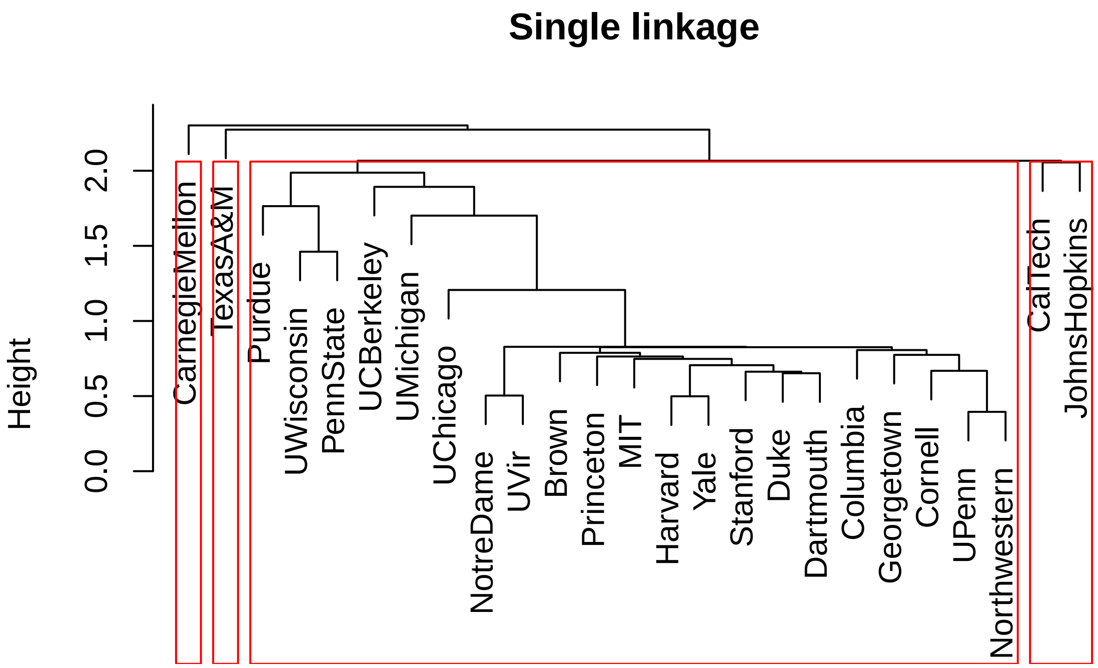
```
clusters1 = cutree(hc1,k=4)
clusters1
```

```
##       Harvard     Princeton          Yale      Stanford           MIT
##             1             1             1             1             1
##          Duke       CalTech     Dartmouth         Brown  JohnsHopkins
##             1             1             1             2             1
##      UChicago         UPenn       Cornell   Northwestern      Columbia
##             2             2             2             2             2
##     NotreDame          UVir     Georgetown CarnegieMellon     UMichigan
##             2             2             2             3             3
##    UCBerkeley    UWisconsin     PennState        Purdue      TexasA&M
##             2             3             3             4             4
```

```
# clusters are ordered as in d2 or d1
#
# dataframe with cluster assignments
#
d22 = d1
d22$Complete = clusters1
head(d22)
```

```
##             SAT Top10 Accept SFRatio Expenses Grad Complete
## Harvard   14.00    91     14      11   39.525   97        1
## Princeton 13.75    91     14       8   30.220   95        1
## Yale      13.75    95     19      11   43.514   96        1
## Stanford  13.60    90     20      12   36.450   93        1
## MIT       13.80    94     30      10   34.870   91        1
## Duke      13.15    90     30      12   31.585   95        1
```

```
#
# HClustering - single
#
hc2 = hclust(distances,method='single')
plot(hc2,sub='',xlab='',main = '')
title('Single linkage')
rect.hclust(hc2,k=4)
```


Single linkage

```
clusters2 = cutree(hc2,k=4)
#
# dataframe with cluster assignments
#
d23 = d1
d23$Single = clusters2
#
# compare clusters
#
d3 = merge(d22,d23)
d3
```

```
##       SAT Top10 Accept SFRatio Expenses Grad Complete Single
## 1   10.05    28     90      19    9.066   69        4      1
## 2   10.75    49     67      25    8.704   67        4      4
## 3   10.81    38     54      18   10.185   80        3      1
## 4   10.85    40     69      15   11.857   71        3      1
## 5   11.80    65     68      16   15.470   85        3      1
## 6   12.25    77     44      14   13.349   92        2      1
## 7   12.40    95     40      17   15.140   78        2      1
## 8   12.55    74     24      12   20.126   92        2      1
## 9   12.55    81     42      13   15.122   94        2      1
## 10 12.60    62     59       9   25.026   72        3      3
## 11 12.60    85     39      11   28.052   89        2      1
## 12 12.80    83     33      13   21.864   90        2      1
## 13 12.85    80     36      11   27.553   90        2      1
## 14 12.90    75     50      13   38.380   87        2      1
## 15 13.05    75     44       7   58.691   87        1      2
## 16 13.10    76     24      12   31.510   88        2      1
## 17 13.10    89     22      13   22.704   94        2      1
## 18 13.15    90     30      12   31.585   95        1      1
## 19 13.40    89     23      10   32.162   95        1      1
## 20 13.60    90     20      12   36.450   93        1      1
## 21 13.75    91     14       8   30.220   95        1      1
## 22 13.75    95     19      11   43.514   96        1      1
## 23 13.80    94     30      10   34.870   91        1      1
## 24 14.00    91      14       11    39.525    97         1       1
## 25 14.15   100      25        6    63.575    81         1       2
```

```
#
# c) K-means
#
set.seed(2)
k=4
kmeans = kmeans(d2,centers=k,nstart = 20)
assignments = kmeans$cluster
clusplot(d2,assignments,lines=0,color=T,shade=T,labels=k,cex=0.6,main='K-means Universities')
#
# K-means clusters separate Universities well
#
# find out which universities assigned to each cluster
#
d4 = d1
d4$kmeans = assignments
d4[order(d4$kmeans),]
```

```
##                   SAT Top10 Accept SFRatio Expenses Grad kmeans
## UChicago        12.90    75     50      13   38.380   87      1
## UPenn           12.85    80     36      11   27.553   90      1
## Cornell         12.80    83     33      13   21.864   90      1
## Northwestern    12.60    85     39      11   28.052   89      1
## NotreDame       12.55    81     42      13   15.122   94      1
## UVir            12.25    77     44      14   13.349   92      1
## Georgetown      12.55    74     24      12   20.126   92      1
## CarnegieMellon  12.60    62     59       9   25.026   72      1
## UMichigan       11.80    65     68      16   15.470   85      1
## UCBerkeley      12.40    95     40      17   15.140   78      1
## CalTech         14.15   100     25       6   63.575   81      2
## JohnsHopkins    13.05    75     44       7   58.691   87      2
## Harvard         14.00    91     14      11   39.525   97      3
## Princeton       13.75    91     14       8   30.220   95      3
## Yale            13.75    95     19      11   43.514   96      3
## Stanford        13.60    90     20      12   36.450   93      3
## MIT             13.80    94     30      10   34.870   91      3
## Duke            13.15    90     30      12   31.585   95      3
## Dartmouth       13.40    89     23      10   32.162   95      3
## Brown           13.10    89     22      13   22.704   94      3
## Columbia        13.10    76     24      12   31.510   88      3
## UWisconsin      10.85    40     69      15   11.857   71      4
## PennState       10.81    38     54      18   10.185   80      4
## Purdue          10.05    28     90      19    9.066   69      4
## TexasA&M        10.75    49     67      25    8.704   67      4
```

```
#
# Question 2
#
library(cluster)
d1 = read.csv("brands.csv")
str(d1)
```

```
## 'data.frame':    43 obs. of  10 variables:
##  $ Brand    : Factor w/ 43 levels "ACCheerios","AllBran",..: 1 6 7 10 17 19 21 23 25 29 ...
##  $ Manuf    : Factor w/ 3 levels "G","K","Q": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Calories : int  110 110 110 110 110 110 110 110 100 130 ...
##  $ Protein  : int  2 6 1 1 1 3 2 2 2 3 ...
##  $ Fat      : int  2 2 1 1 1 1 1 1 1 2 ...
##  $ Sodium   : int  180 290 180 180 280 250 260 180 220 170 ...
##  $ Fiber    : num  1.5 2 0 0 0 1.5 0 0 2 1.5 ...
##  $ Chydrates: num  10.5 17 12 12 15 11.5 21 12 15 13.5 ...
##  $ Sugar    : int  10 1 13 13 9 10 3 12 6 10 ...
##  $ Potassium: int  70 105 55 65 45 90 40 55 90 120 ...
```

```
rownames(d1) = d1[,1]
d1$Brand = NULL
head(d1)
```

```
##                  Manuf Calories Protein Fat Sodium Fiber Chydrates Sugar
## ACCheerios           G      110       2   2    180   1.5      10.5    10
## Cheerios             G      110       6   2    290   2.0      17.0     1
## CocoaPuffs           G      110       1   1    180   0.0      12.0    13
## CountChocula         G      110       1   1    180   0.0      12.0    13
## GoldenGrahams        G      110       1   1    280   0.0      15.0     9
## HoneyNutCheerios     G      110       3   1    250   1.5      11.5    10
##                  Potassium
## ACCheerios              70
## Cheerios               105
## CocoaPuffs              55
## CountChocula            65
## GoldenGrahams           45
## HoneyNutCheerios        90
```

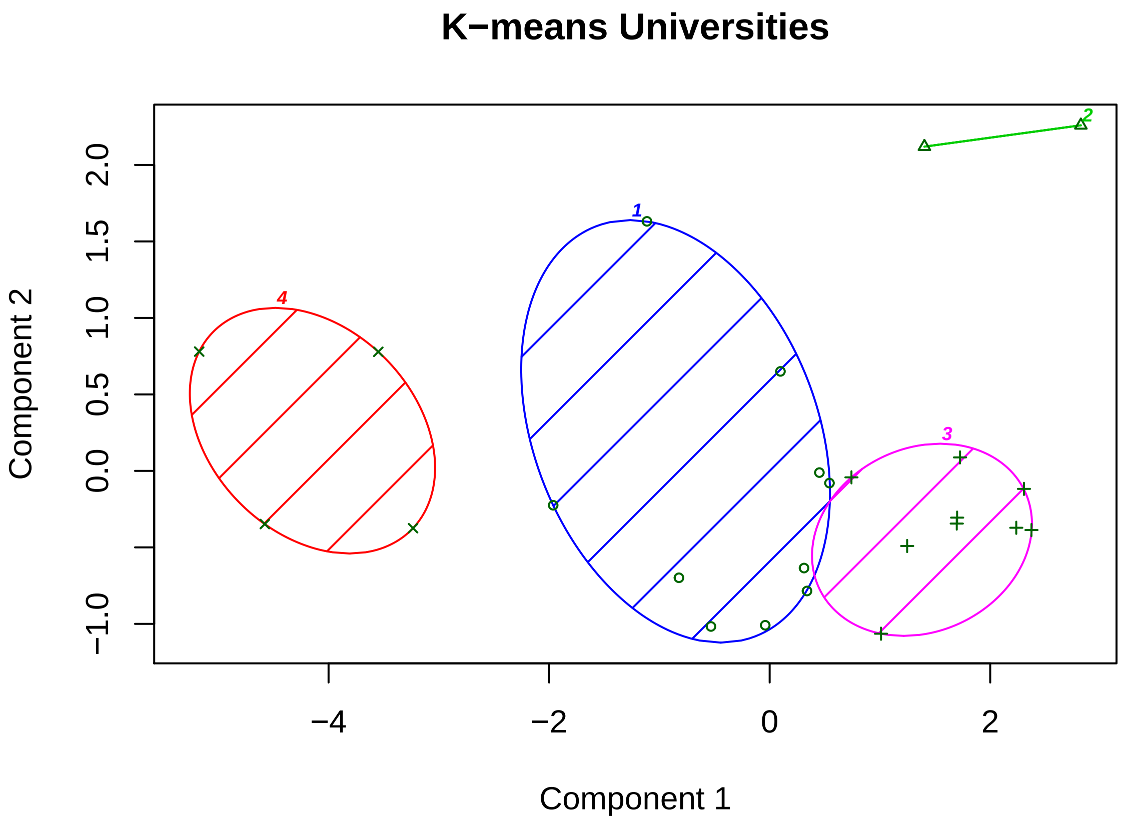
```
#
# dataframe with numeric columns only
#
d2 = d1
manuf = d2[,1]
table(manuf)
```

```
## manuf
##  G  K  Q
## 17 20  6
```

```
# d2$Brand = NULL
d2$Manuf = NULL
head(d2)
```

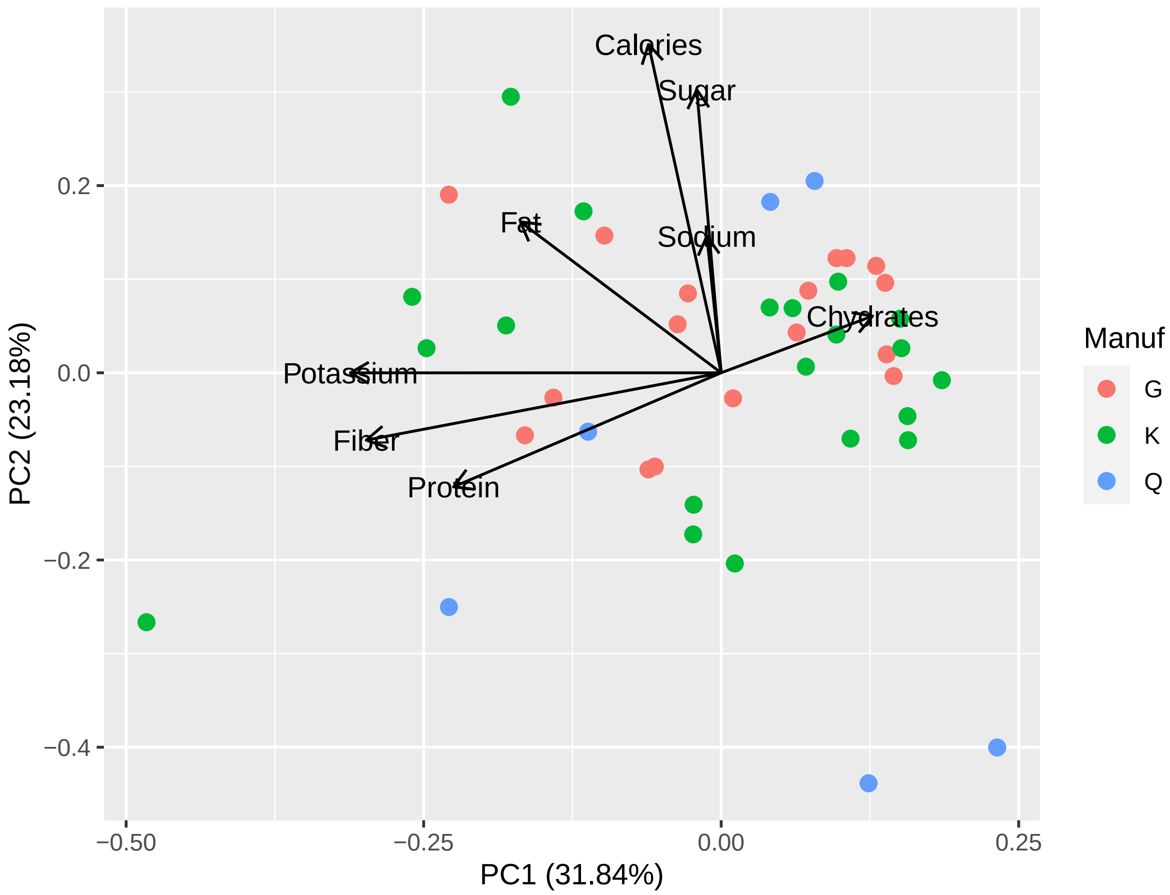
```
##                  Calories Protein Fat Sodium Fiber Chydrates Sugar Potassium
## ACCheerios            110       2   2    180   1.5      10.5    10        70
## Cheerios              110       6   2    290   2.0      17.0     1       105
## CocoaPuffs            110       1   1    180   0.0      12.0    13        55
## CountChocula          110       1   1    180   0.0      12.0    13        65
## GoldenGrahams         110       1   1    280   0.0      15.0     9        45
## HoneyNutCheerios      110       3   1    250   1.5      11.5    10        90
```

```
#
# a) principal components
#
m2 = prcomp(d2, scale=T)
#
# plot
library(ggfortify)   # autoplot()
```

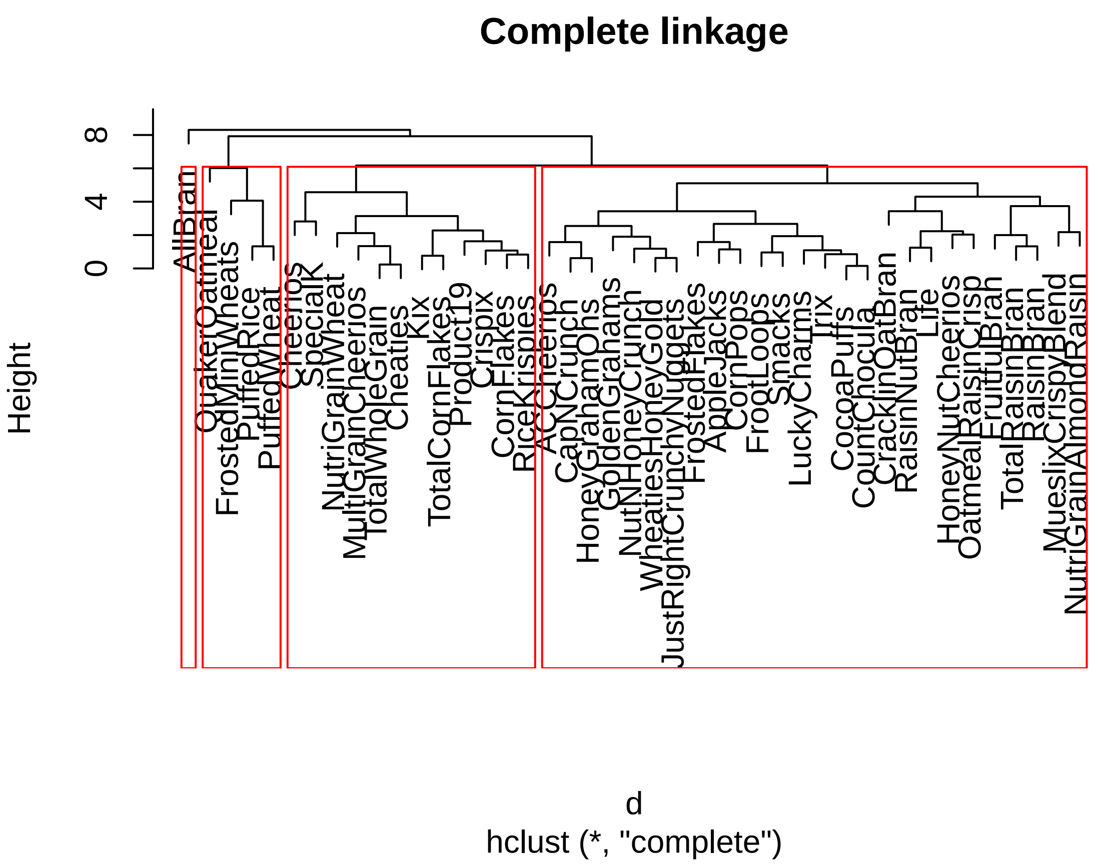


These two components explain 89.98 % of the point variability.

```
#
autoplot(m2,data=d1,colour = 'Manuf',loadings = TRUE,loadings.label = TRUE,size=2.5,
         loadings.colour = 'black',loadings.label.colour='black')
```



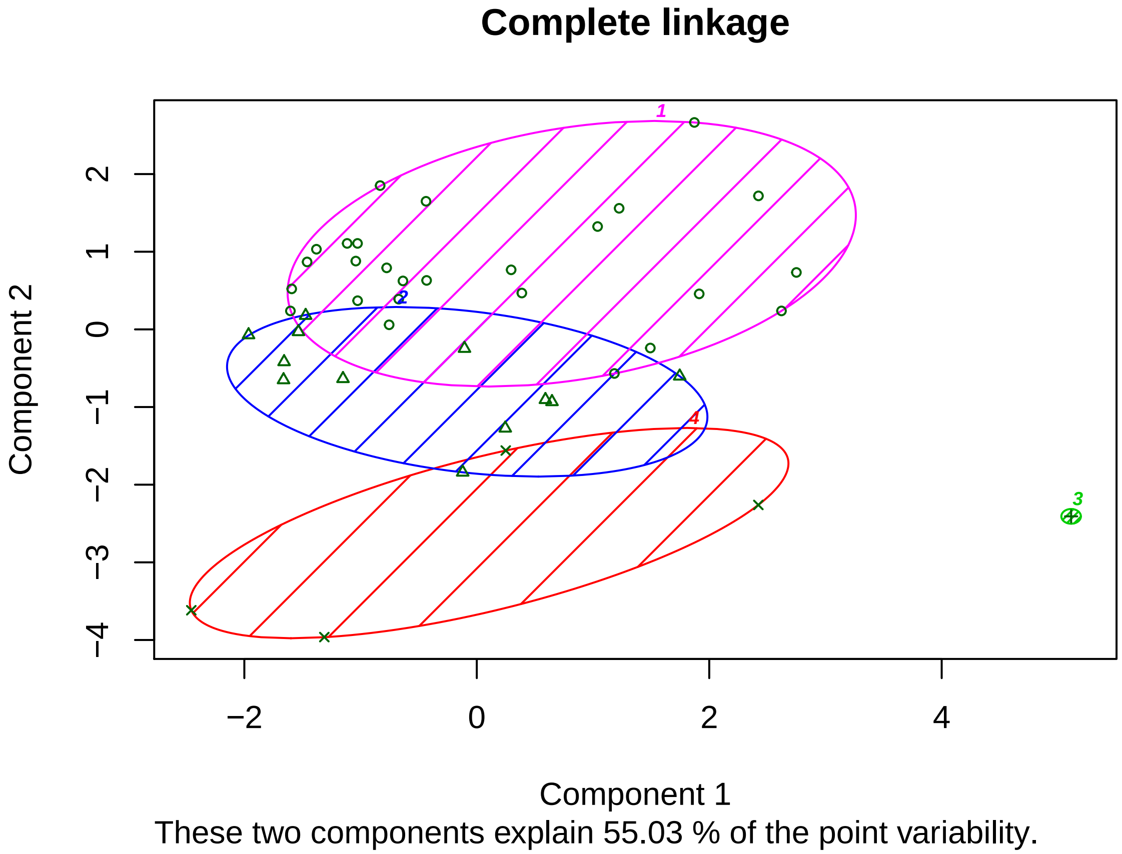
```
#
# Most G and K brands have lots of Carbohydrates.
# it is not clear that some brands are associated with more nutritional elements
#
# b) Complete and single linkage
#
# distances - scaled data
#
d3 = scale(d2)
d = dist(d3)
#
# complete linkage
#
hcomplete = hclust(d,method="complete")
plot(hcomplete,main='Complete linkage')
rect.hclust(hcomplete,k=4)
```



```
#
# add cluster assignment column
#
assignment = cutree(hcomplete,4)
dcomplete = data.frame(d1,assignment)
dcomplete = dcomplete[order(-assignment),]
dcomplete[,c(1,2,10)]
```

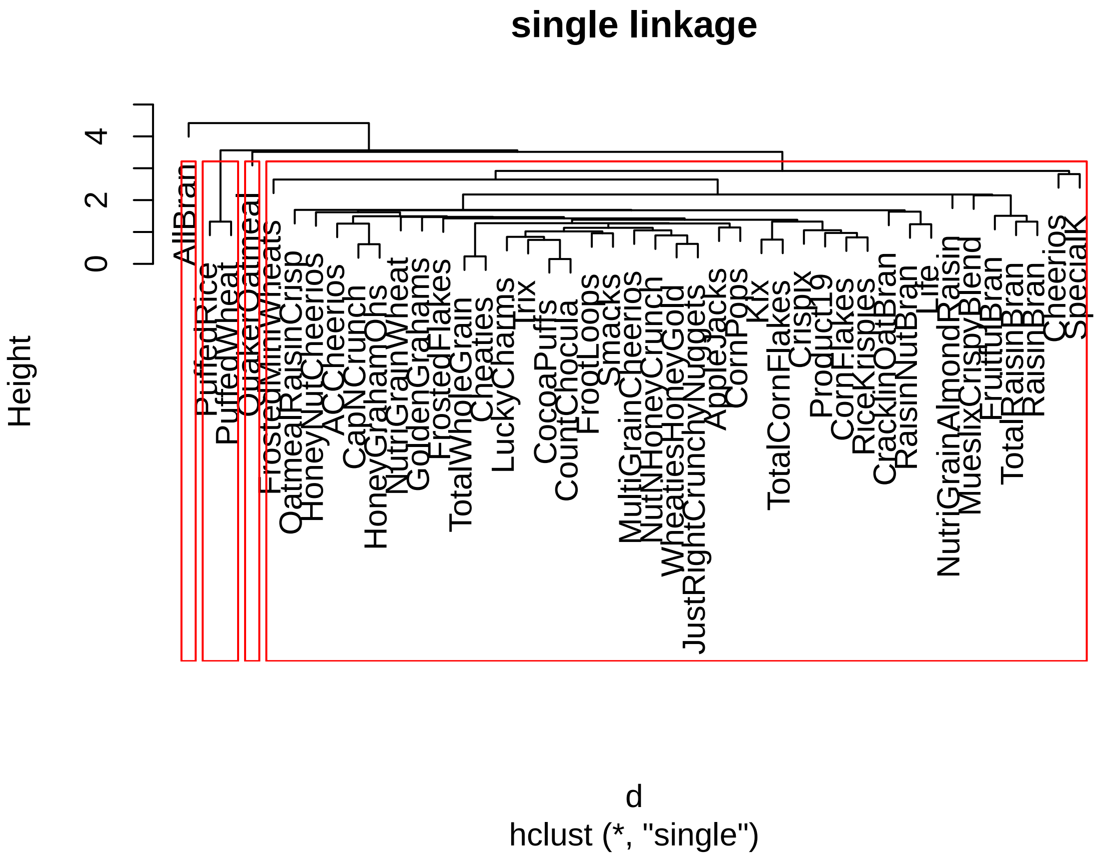
```
##                           Manuf Calories assignment
## FrostedMiniWheats             K      100          4
## PuffedRice                    Q       50          4
## PuffedWheat                   Q       50          4
## QuakerOatmeal                 Q      100          4
## AllBran                       K       70          3
## Cheerios                      G      110          2
## Kix                           G      110          2
## MultiGrainCheerios            G      100          2
## TotalCornFlakes               G      110          2
## TotalWholeGrain               G      100          2
## Cheaties                      G      100          2
## CornFlakes                    K      100          2
## Crispix                       K      110          2
## NutriGrainWheat               K       90          2
## Product19                     K      100          2
## RiceKrispies                  K      110          2
## SpecialK                      K      110          2
## ACCheerios                    G      110          1
## CocoaPuffs                    G      110          1
## CountChocula                  G      110          1
## GoldenGrahams               G      110          1
## HoneyNutCheerios            G      110          1
## LuckyCharms                 G      110          1
## OatmealRaisinCrisp          G      130          1
## RaisinNutBran               G      100          1
## TotalRaisinBran             G      140          1
## Trix                        G      110          1
## WheatiesHoneyGold           G      110          1
## AppleJacks                  K      110          1
## CornPops                    K      110          1
## CracklinOatBran             K      110          1
## FrootLoops                  K      110          1
## FrostedFlakes               K      110          1
## FruitfulBran                K      120          1
## JustRightCrunchyNuggets     K      110          1
## MueslixCrispyBlend          K      160          1
## NutNHoneyCrunch             K      120          1
## NutriGrainAlmondRaisin      K      140          1
## RaisinBran                  K      120          1
## Smacks                      K      110          1
## CapNCrunch                  Q      120          1
## HoneyGrahamOhs              Q      120          1
## Life                        Q      100          1
```

```
#
clusplot(d2,assignment,lines=0,color=T,shade=T,labels=4,cex=0.6,main='Complete linkage')
```



```
#
# single linkage
#
```

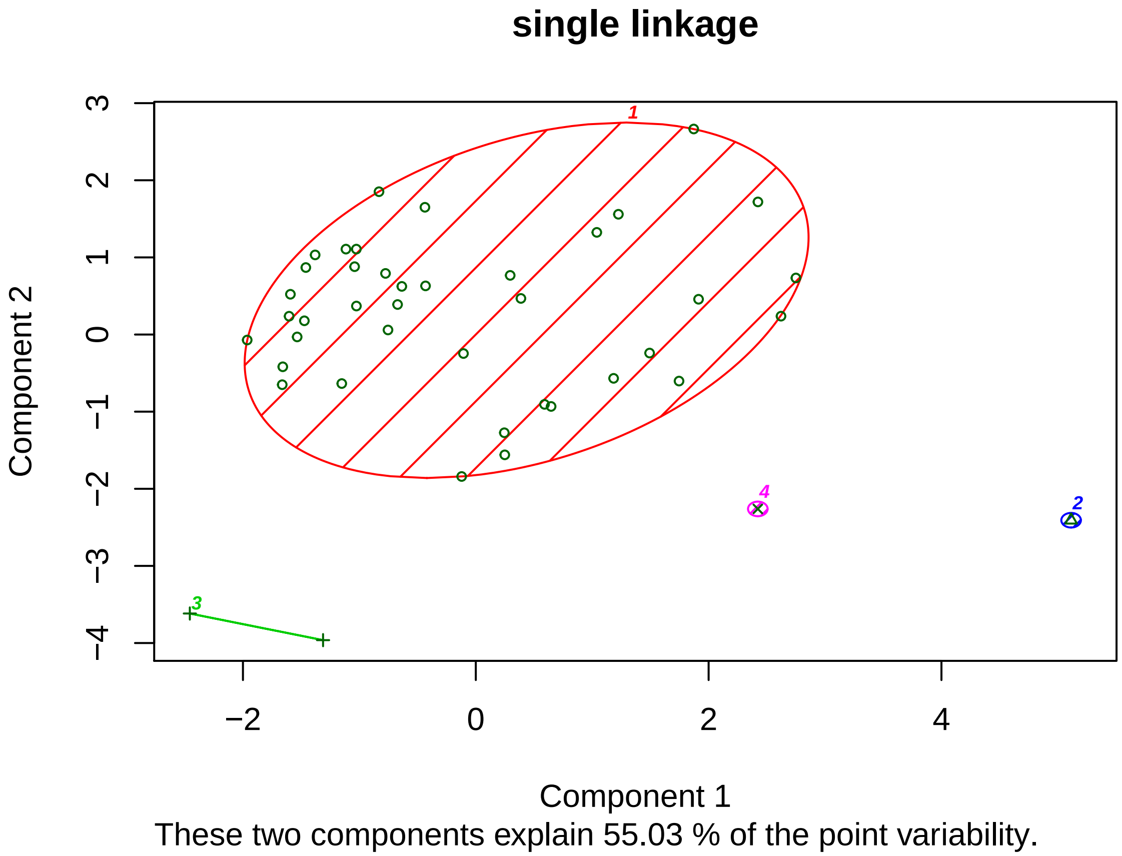
```
hsingle = hclust(d,method="single")
plot(hsingle,main='single linkage')
rect.hclust(hsingle,k=4)
```



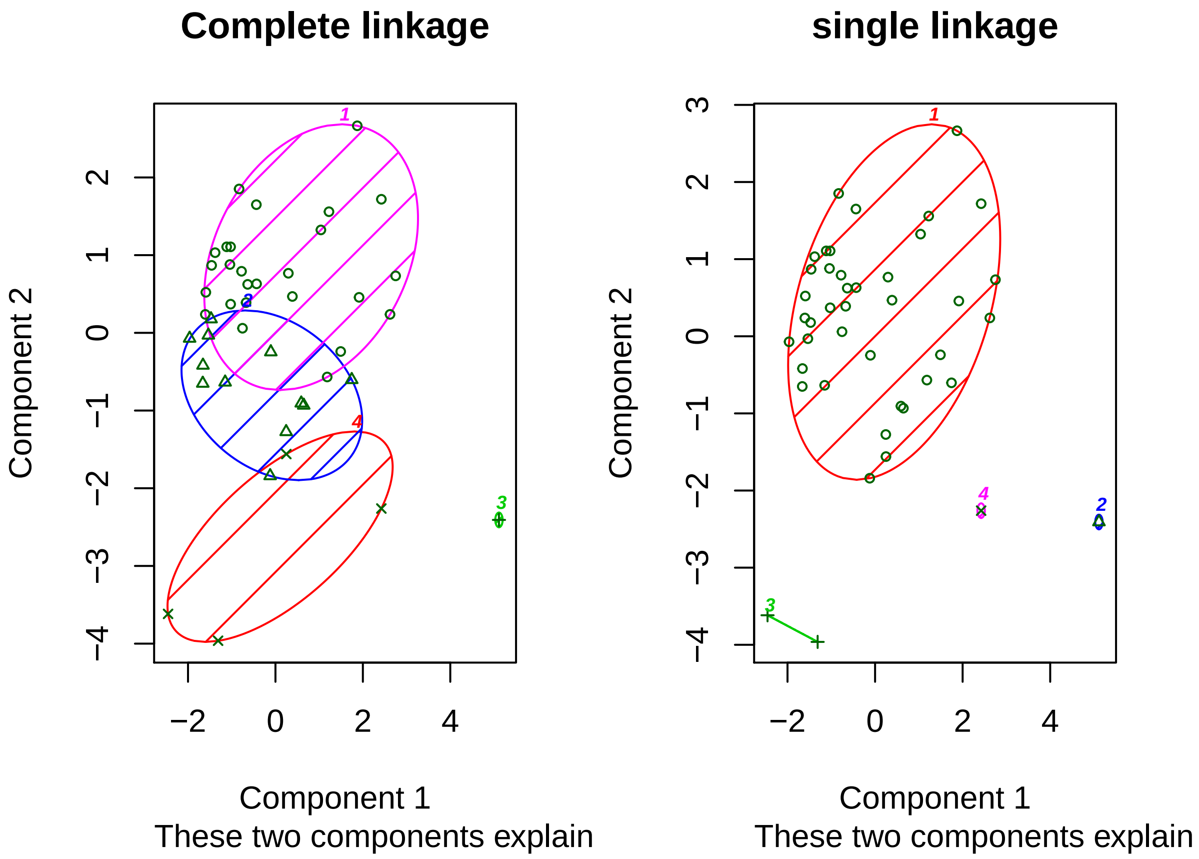
```
#
# add cluster assignment column
#
assignment2 = cutree(hsingle,4)
dsingle = data.frame(d1,assignment2)
dsingle = dsingle[order(-assignment2),]
dsingle[,c(1,2,10)]
```

```
##                           Manuf Calories assignment2
## QuakerOatmeal                 Q      100           4
## PuffedRice                    Q       50           3
## PuffedWheat                   Q       50           3
## AllBran                       K       70           2
## ACCheerios                    G      110           1
## Cheerios                      G      110           1
## CocoaPuffs                    G      110           1
## CountChocula                  G      110           1
## GoldenGrahams                 G      110           1
## HoneyNutCheerios              G      110           1
## Kix                           G      110           1
## LuckyCharms                   G      110           1
## MultiGrainCheerios            G      100           1
## OatmealRaisinCrisp            G      130           1
## RaisinNutBran                 G      100           1
## TotalCornFlakes               G      110           1
## TotalRaisinBran             G      140           1
## TotalWholeGrain             G      100           1
## Trix                        G      110           1
## Cheaties                    G      100           1
## WheatiesHoneyGold           G      110           1
## AppleJacks                  K      110           1
## CornFlakes                  K      100           1
## CornPops                    K      110           1
## CracklinOatBran             K      110           1
## Crispix                     K      110           1
## FrootLoops                  K      110           1
## FrostedFlakes               K      110           1
## FrostedMiniWheats           K      100           1
## FruitfulBran                K      120           1
## JustRightCrunchyNuggets     K      110           1
## MueslixCrispyBlend          K      160           1
## NutNHoneyCrunch             K      120           1
## NutriGrainAlmondRaisin      K      140           1
## NutriGrainWheat             K       90           1
## Product19                   K      100           1
## RaisinBran                  K      120           1
## RiceKrispies                K      110           1
## Smacks                      K      110           1
## SpecialK                    K      110           1
## CapNCrunch                  Q      120           1
## HoneyGrahamOhs              Q      120           1
## Life                        Q      100           1
```

```
#
clusplot(d2,assignment2,lines=0,color=T,shade=T,labels=4,cex=0.6,main='single linkage')
```



```
#
# compare plots
#
par(mfrow=c(1,2))
clusplot(d2,assignment,lines=0,color=T,shade=T,labels=4,cex=0.6,main='Complete linkage')
clusplot(d2,assignment2,lines=0,color=T,shade=T,labels=4,cex=0.6,main='single linkage')
```



```
par(mfrow=c(1,1))
#
# two largest clusters in complete linkage
# appear as a single big cluster in single linkage
#
# while brands 41-43 and 26 appear in a cluster using complete linkage
# only 41-42 appear in a cluster using single linkage, leaving 26 in the largest cluster
#
# cluster with only one brand contains
d1[c(18),]
```

```
##          Manuf Calories Protein Fat Sodium Fiber Chydrates Sugar Potassium
## AllBran     K       70       4   1    260     9         7     5       320
```

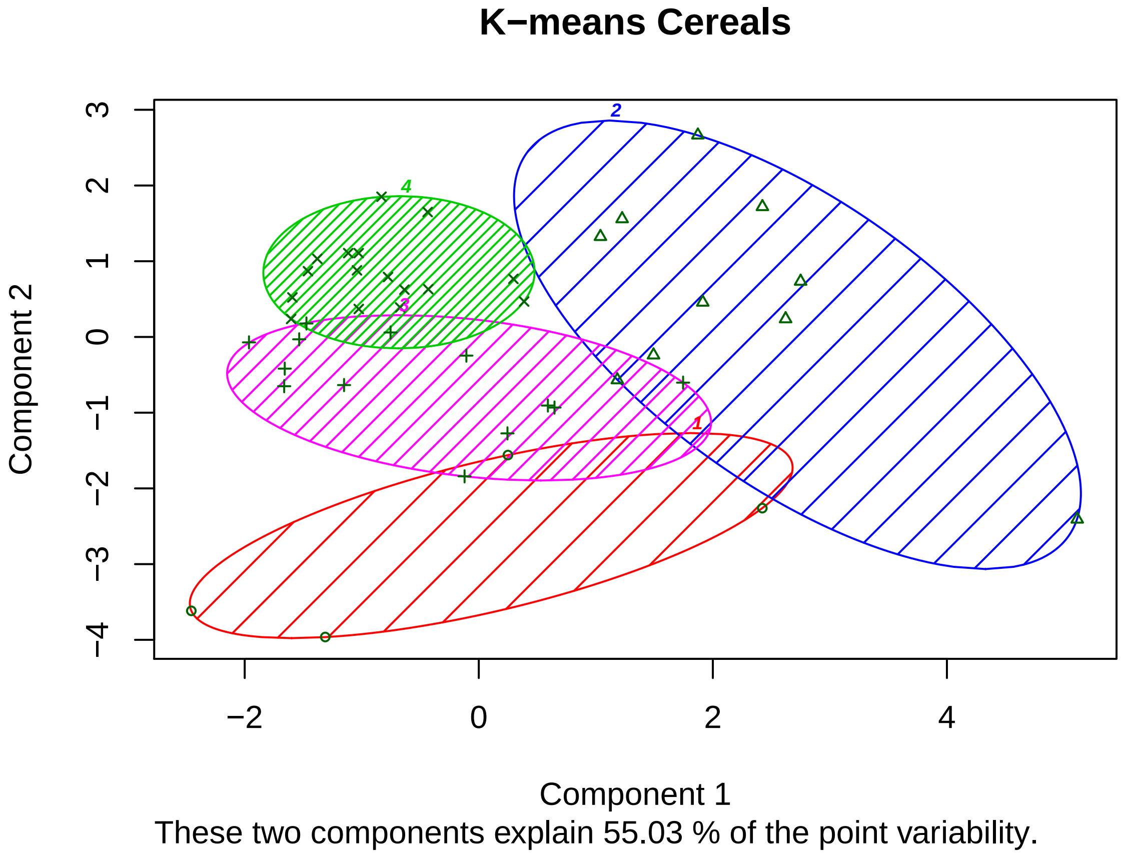
```
#
# cluster with four brands (single) that are split into two clusters (complete)
d1[c(26,41:43),]
```

```
##                    Manuf Calories Protein Fat Sodium Fiber Chydrates Sugar
## FrostedMiniWheats     K      100       3   0      0   3.0        14     7
## PuffedRice            Q       50       1   0      0   0.0        13     0
## PuffedWheat           Q       50       2   0      0   1.0        10     0
## QuakerOatmeal         Q      100       5   2      0   2.7         1     1
##                    Potassium
## FrostedMiniWheats       100
## PuffedRice               15
## PuffedWheat              50
## QuakerOatmeal           110
```

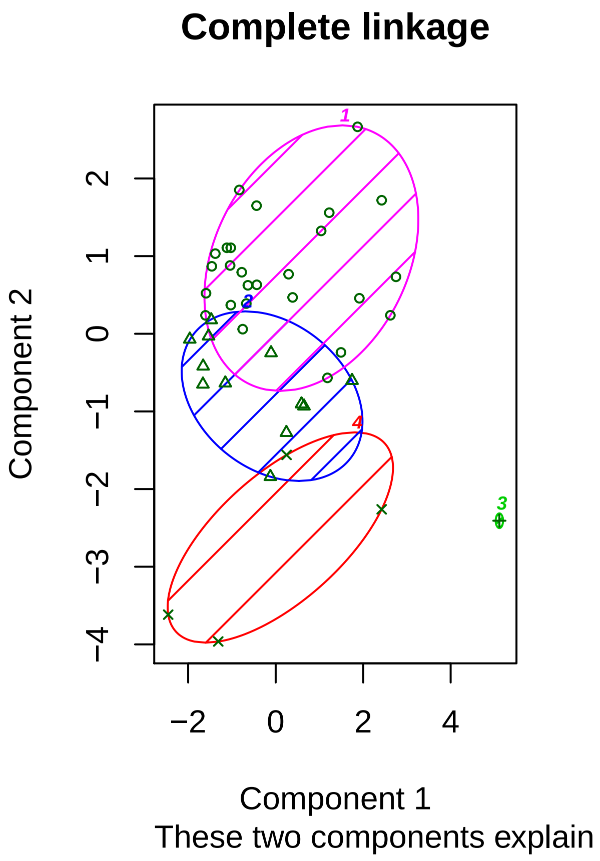
```
#
# c) kmeans
#
kmodel = kmeans(d3,4,nstart=20)
assignment3 = kmodel$cluster
dkmeans = data.frame(d1,assignment3)
dkmeans = dkmeans[order(-assignment3),]
dkmeans[,c(1,2,10)]
```

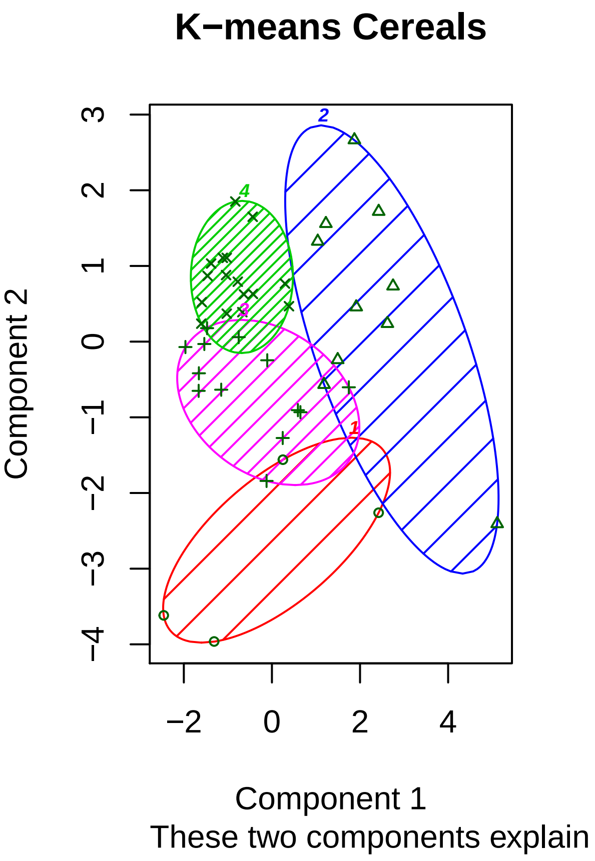
```
##                         Manuf Calories assignment3
## ACCheerios                  G      110           4
## CocoaPuffs                  G      110           4
## CountChocula                G      110           4
## GoldenGrahams               G      110           4
## HoneyNutCheerios            G      110           4
## LuckyCharms                 G      110           4
## Trix                        G      110           4
## WheatiesHoneyGold           G      110           4
## AppleJacks                  K      110           4
## CornPops                    K      110           4
## FrootLoops                  K      110           4
## FrostedFlakes               K      110           4
## NutNHoneyCrunch             K      120           4
## Smacks                      K      110           4
## CapNCrunch                  Q      120           4
## HoneyGrahamOhs              Q      120           4
## Cheerios                    G      110           3
## Kix                         G      110           3
## MultiGrainCheerios          G      100           3
## TotalCornFlakes             G      110           3
## TotalWholeGrain             G      100           3
## Cheaties                    G      100           3
## CornFlakes                  K      100           3
## Crispix                     K      110           3
## JustRightCrunchyNuggets     K      110           3
## NutriGrainWheat             K       90           3
## Product19                   K      100           3
## RiceKrispies                K      110           3
## SpecialK                    K      110           3
## OatmealRaisinCrisp          G      130           2
## RaisinNutBran               G      100           2
## TotalRaisinBran             G      140           2
## AllBran                     K       70           2
## CracklinOatBran             K      110           2
## FruitfulBran                K      120           2
## MueslixCrispyBlend          K      160           2
## NutriGrainAlmondRaisin      K      140           2
## RaisinBran                  K      120           2
## Life                        Q      100           2
## FrostedMiniWheats           K      100           1
## PuffedRice                  Q       50           1
## PuffedWheat                   Q       50            1
## QuakerOatmeal                 Q      100            1
```

```
# dkmeans
clusplot(d2,assignment3,lines=0,color=T,shade=T,labels=4,cex=0.6,main='K-means Cereals')
```



```
#
# compare k-means with Complete linkage
#
par(mfrow=c(1,2))
clusplot(d2,assignment,lines=0,color=T,shade=T,labels=4,cex=0.6,main='Complete linkage')
clusplot(d2,assignment3,lines=0,color=T,shade=T,labels=4,cex=0.6,main='K-means Cereals')
```





```
par(mfrow=c(1,1))
#
# Two clusters seem to be the same in both Complete linkage and k-means
#
# k-means seems to perform better since overall the clusters have less overlap
# than those from Complete linkage
#
```