MVA Project

Team Avengers

25 April 2018

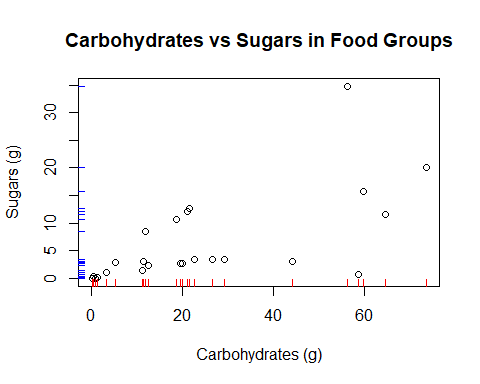
DATA PROCESSING

food <- read.csv("project.csv")  
food.2 <- food[,-c(1,3,4,5,6,7)]  
food.3 <- food[,-c(1,2,3,4,5,6,7)]  
food.group <- aggregate(.~FoodGroup, data=food.2, mean)  
labels <- food.group$FoodGroup  
food.group <- food.group[,-1]  
rownames(food.group) <- labels  
head(food.group)

## Energy\_kcal Protein\_g Fat\_g  
## American Indian/Alaska Native Foods 209.1152 15.996061 11.1097576  
## Baby Foods 151.9917 4.114669 5.5804420  
## Baked Products 391.8871 6.654831 14.6679046  
## Beef Products 212.7495 24.369736 12.3613531  
## Beverages 111.3079 2.570159 0.9848889  
## Breakfast Cereals 342.4959 8.298375 4.0041873  
## Carb\_g Sugar\_g Fiber\_g  
## American Indian/Alaska Native Foods 11.2539394 1.358424 2.148484848  
## Baby Foods 21.5708840 12.675525 0.866022099  
## Baked Products 59.7672020 15.663112 3.044918444  
## Beef Products 0.1084567 0.000000 0.002854123  
## Beverages 18.6854286 10.709016 0.718412698  
## Breakfast Cereals 73.5282645 20.133085 7.738016529  
## VitA\_mcg VitB6\_mg VitB12\_mcg  
## American Indian/Alaska Native Foods 64.04242 0.1449152 2.3158788  
## Baby Foods 109.74033 0.1376740 0.3563536  
## Baked Products 52.48557 0.1173174 0.1468632  
## Beef Products 78.88161 0.4347526 3.0737315  
## Beverages 81.02857 0.1946952 0.3040952  
## Breakfast Cereals 237.01653 1.3100441 3.0193939  
## VitC\_mg VitE\_mg Folate\_mcg  
## American Indian/Alaska Native Foods 7.4345455 0.7743030 7.460606  
## Baby Foods 18.9759669 1.3768232 32.030387  
## Baked Products 0.2568381 0.3973651 54.465496  
## Beef Products 0.3809725 0.2144186 7.073996  
## Beverages 28.4482540 0.7427302 17.784127  
## Breakfast Cereals 16.4134986 2.5443251 487.082645  
## Niacin\_mg Riboflavin\_mg Thiamin\_mg  
## American Indian/Alaska Native Foods 3.021352 0.2284848 0.09590909  
## Baby Foods 2.617983 0.2844420 0.22689503  
## Baked Products 3.087526 0.2575433 0.34269762  
## Beef Products 4.885542 0.2326131 0.07475370  
## Beverages 2.283838 0.1792444 0.07133651  
## Breakfast Cereals 12.297590 1.0017686 0.99712121  
## Calcium\_mg Copper\_mcg Iron\_mg  
## American Indian/Alaska Native Foods 61.31515 0.12346667 4.5130303  
## Baby Foods 136.77348 0.12300829 3.8978177  
## Baked Products 91.03513 0.08498996 2.9253325  
## Beef Products 12.87949 0.15434884 2.5219979  
## Beverages 73.34286 0.09829206 0.8360317  
## Breakfast Cereals 157.46556 0.16379614 15.9623967  
## Magnesium\_mg Manganese\_mg  
## American Indian/Alaska Native Foods 27.23030 0.2301212  
## Baby Foods 18.97790 0.5795028  
## Baked Products 24.50690 0.3131920  
## Beef Products 20.21353 0.6927167  
## Beverages 28.90159 2.6952286  
## Breakfast Cereals 69.79339 0.8867824  
## Phosphorus\_mg Selenium\_mcg Zinc\_mg  
## American Indian/Alaska Native Foods 181.98788 19.430303 1.2758182  
## Baby Foods 105.58287 4.911050 1.1687845  
## Baked Products 162.43664 8.120954 0.6284316  
## Beef Products 198.37949 24.482981 5.4205814  
## Beverages 70.12063 2.312063 0.6350159  
## Breakfast Cereals 220.45730 10.633333 5.8288430  
## VitA\_USRDA VitB6\_USRDA VitB12\_USRDA  
## American Indian/Alaska Native Foods 0.07115825 0.08524421 0.96494949  
## Baby Foods 0.12193370 0.08098473 0.14848066  
## Baked Products 0.05831730 0.06901026 0.06119302  
## Beef Products 0.08764623 0.25573685 1.28072146  
## Beverages 0.09003175 0.11452661 0.12670635  
## Breakfast Cereals 0.26335170 0.77061416 1.25808081  
## VitC\_USRDA VitE\_USRDA Folate\_USRDA  
## American Indian/Alaska Native Foods 0.082606061 0.05162020 0.01865152  
## Baby Foods 0.210844076 0.09178821 0.08007597  
## Baked Products 0.002853757 0.02649101 0.13616374  
## Beef Products 0.004233028 0.01429457 0.01768499  
## Beverages 0.316091711 0.04951534 0.04446032  
## Breakfast Cereals 0.182372207 0.16962167 1.21770661  
## Niacin\_USRDA Riboflavin\_USRDA  
## American Indian/Alaska Native Foods 0.1888345 0.1757576  
## Baby Foods 0.1636240 0.2188015  
## Baked Products 0.1929704 0.1981102  
## Beef Products 0.3053464 0.1789332  
## Beverages 0.1427399 0.1378803  
## Breakfast Cereals 0.7685993 0.7705912  
## Thiamin\_USRDA Calcium\_USRDA  
## American Indian/Alaska Native Foods 0.07992424 0.05109596  
## Baby Foods 0.18907919 0.11397790  
## Baked Products 0.28558135 0.07586261  
## Beef Products 0.06229475 0.01073291  
## Beverages 0.05944709 0.06111905  
## Breakfast Cereals 0.83093434 0.13122130  
## Copper\_USRDA Magnesium\_USRDA  
## American Indian/Alaska Native Foods 1.371852e-04 0.06483405  
## Baby Foods 1.366759e-04 0.04518548  
## Baked Products 9.443329e-05 0.05834976  
## Beef Products 1.714987e-04 0.04812745  
## Beverages 1.092134e-04 0.06881330  
## Breakfast Cereals 1.819957e-04 0.16617473  
## Phosphorus\_USRDA Selenium\_USRDA  
## American Indian/Alaska Native Foods 0.2599827 0.35327824  
## Baby Foods 0.1508327 0.08929181  
## Baked Products 0.2320523 0.14765370  
## Beef Products 0.2833993 0.44514511  
## Beverages 0.1001723 0.04203752  
## Breakfast Cereals 0.3149390 0.19333333  
## Zinc\_USRDA  
## American Indian/Alaska Native Foods 0.11598347  
## Baby Foods 0.10625314  
## Baked Products 0.05713015  
## Beef Products 0.49278013  
## Beverages 0.05772872  
## Breakfast Cereals 0.52989482

VISUALIZATION

#scatter plot, Carb\_g v Sugar\_g  
#plot(nutrition$Carb\_g, nutrition$Sugar\_g, xlab = "Carbohydrates (g)",   
# ylab = "Sugars (g)", main = "Carbohydrates vs Sugars in Foods" )  
#x <- jitter(nutrition$Carb\_g)  
#y <- jitter(nutrition$Sugar\_g)  
#plot(x, y, xlab = "Carbohydrates (g)",   
# ylab = "Sugars (g)", main = "Carbohydrates vs Sugars in Foods")  
  
#add rug  
#rug(x, side = 1, col = "red")  
#rug(y, side = 2, col = "blue")  
#as we can see, we need to aggregate the groups for proper analysis  
  
#need to aggregate food groups as we cannot tell anything from the data by   
#plotting every individual item  
  
#remove unnecssary categorical data columns for analysis  
food.2 <- food[,-c(1,3,4,5,6,7)]  
food.group <- aggregate(.~FoodGroup, data=food.2, mean)  
labels <- food.group$FoodGroup  
food.group <- food.group[,-1]  
rownames(food.group) <- labels  
  
#Food group carb\_g v Sugar\_g scatterplot  
plot(food.group$Carb\_g, food.group$Sugar\_g, xlab = "Carbohydrates (g)",   
 ylab = "Sugars (g)", main = "Carbohydrates vs Sugars in Food Groups" )  
#add rug  
rug(food.group$Carb\_g, side = 1, col = "red")  
rug(food.group$Sugar\_g, side = 2, col = "blue")



#Select Carbs and Sugars columns  
mydata <- food.group[, c("Carb\_g", "Sugar\_g")]  
  
#Sugars v Carbs bivariate-boxplot  
library(MVA)

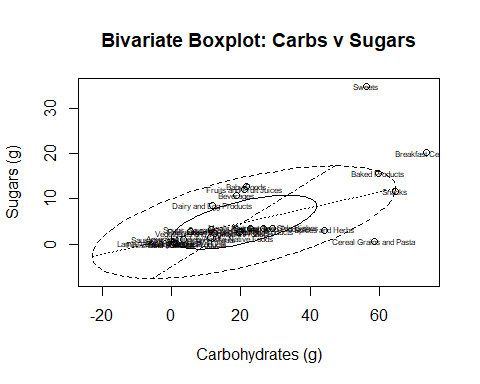
## Warning: package 'MVA' was built under R version 3.3.3

## Loading required package: HSAUR2

## Warning: package 'HSAUR2' was built under R version 3.3.3

## Loading required package: tools

bvbox(mydata, xlab = "Carbohydrates (g)", ylab = "Sugars (g)",  
 main = "Bivariate Boxplot: Carbs v Sugars")  
text(mydata, labels = rownames(mydata), cex = .5)



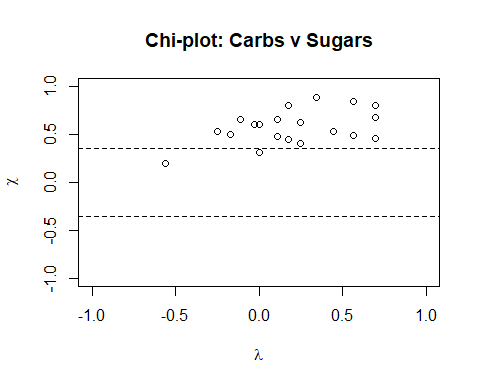
#Here, Sweets, breakfast cereals, and cereal, grains and Pasta are outliers  
  
#original correlation  
cor(food.group$Carb\_g, food.group$Sugar\_g)

## [1] 0.6498989

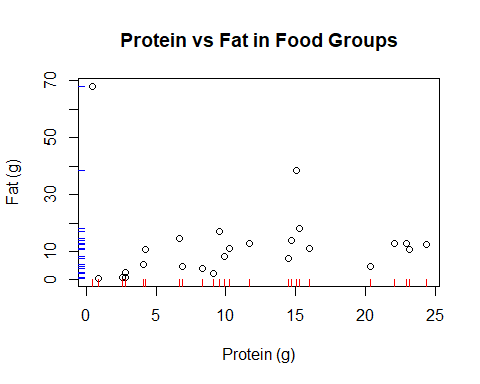
#remove outliers  
outlier <- match(c("Sweets", "Breakfast Cereals",  
 "Cereal Grains and Pasta"),  
 rownames(food.group))  
#correlation with outliers removed  
cor(food.group$Carb\_g[-outlier], food.group$Sugar\_g[-outlier])

## [1] 0.6815425

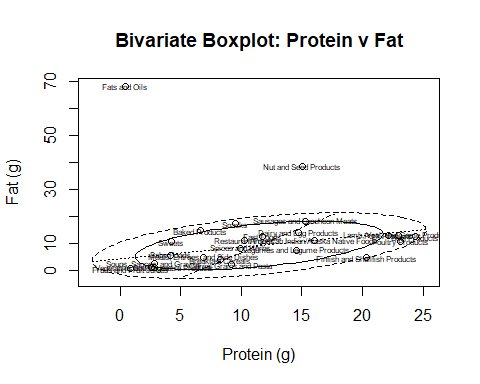
#chi-plot: sugars v carbs  
library(MVA)  
chiplot(food.group$Carb\_g, food.group$Sugar\_g, main = "Chi-plot: Carbs v Sugars")



#Protein v Fats Scatterplot  
plot(food.group$Protein\_g, food.group$Fat\_g, xlab = "Protein (g)",   
 ylab = "Fat (g)", main = "Protein vs Fat in Food Groups" )  
#add rug  
rug(food.group$Protein\_g, side = 1, col = "red")  
rug(food.group$Fat\_g, side = 2, col = "blue")



#Select Protein and Fats columns  
mydata <- food.group[, c("Protein\_g", "Fat\_g")]  
  
#Protein v Fat bivariate-boxplot  
bvbox(mydata, xlab = "Protein (g)", ylab = "Fat (g)",  
 main = "Bivariate Boxplot: Protein v Fat")  
text(mydata, labels = rownames(mydata), cex = .5)



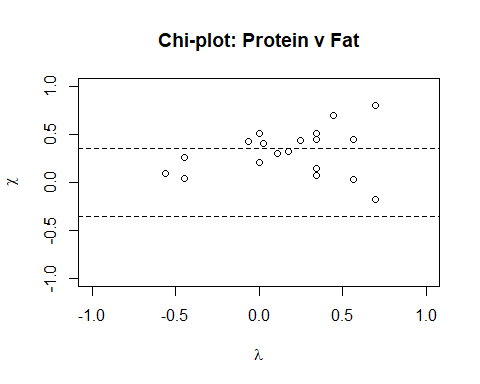
#Here, Fats and Oils and Nut and Seed products are outliers  
  
#original correlation  
cor(food.group$Protein\_g, food.group$Fat\_g)

## [1] -0.01815489

#remove outliers  
outlier <- match(c("Fats and Oils", "Nut and Seed Products"), rownames(food.group))  
#correlation with outliers removed  
cor(food.group$Protein\_g[-outlier], food.group$Fat\_g[-outlier])

## [1] 0.543399

#chi-plot: Protein v Fats  
chiplot(food.group$Protein\_g, food.group$Fat\_g, main = "Chi-plot: Protein v Fat")



PCA

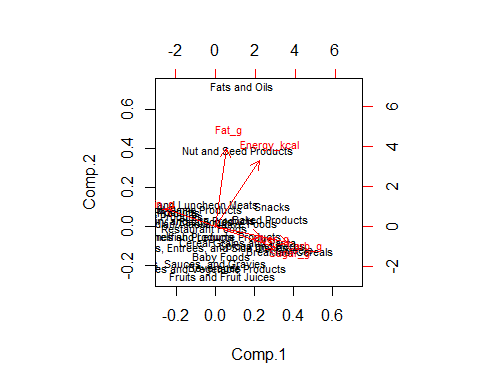
food\_data<- read.csv("project.csv")  
food\_data\_new <- food\_data[,c(2,8,9,10,11,12,13)]  
food\_group <- aggregate(.~FoodGroup, data=food\_data\_new, mean)  
labels <- food\_group$FoodGroup  
food\_group <- food\_group[,-1]  
rownames(food\_group) <- labels  
head(food\_group)

## Energy\_kcal Protein\_g Fat\_g  
## American Indian/Alaska Native Foods 209.1152 15.996061 11.1097576  
## Baby Foods 151.9917 4.114669 5.5804420  
## Baked Products 391.8871 6.654831 14.6679046  
## Beef Products 212.7495 24.369736 12.3613531  
## Beverages 111.3079 2.570159 0.9848889  
## Breakfast Cereals 342.4959 8.298375 4.0041873  
## Carb\_g Sugar\_g Fiber\_g  
## American Indian/Alaska Native Foods 11.2539394 1.358424 2.148484848  
## Baby Foods 21.5708840 12.675525 0.866022099  
## Baked Products 59.7672020 15.663112 3.044918444  
## Beef Products 0.1084567 0.000000 0.002854123  
## Beverages 18.6854286 10.709016 0.718412698  
## Breakfast Cereals 73.5282645 20.133085 7.738016529

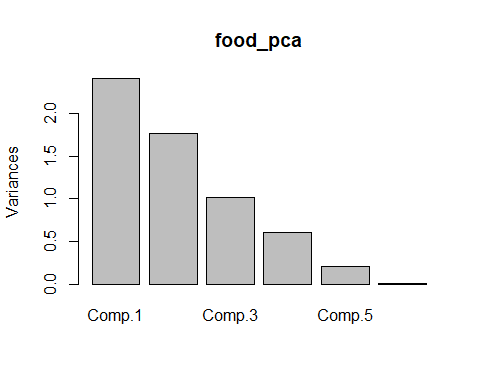
#cor\_food <- cor(food\_group)  
food\_pca <- princomp(food\_group,cor=T)  
summary(food\_pca, loadings = T)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  
## Standard deviation 1.5515373 1.3293332 1.0049777 0.7791812 0.45611868  
## Proportion of Variance 0.4012113 0.2945211 0.1683300 0.1011872 0.03467404  
## Cumulative Proportion 0.4012113 0.6957325 0.8640625 0.9652497 0.99992376  
## Comp.6  
## Standard deviation 2.138843e-02  
## Proportion of Variance 7.624413e-05  
## Cumulative Proportion 1.000000e+00  
##   
## Loadings:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6  
## Energy\_kcal 0.355 0.613 -0.183 0.201 -0.649  
## Protein\_g -0.404 0.163 0.466 -0.745 -0.130 0.144  
## Fat\_g 0.725 -0.128 0.186 -0.240 0.597  
## Carb\_g 0.583 -0.150 0.183 -0.250 0.586 0.446  
## Sugar\_g 0.481 -0.199 -0.412 -0.462 -0.588   
## Fiber\_g 0.362 0.748 0.317 -0.442

biplot(food\_pca,col=c("black", "red"), cex = 0.7, expand = 1.2)



screeplot(food\_pca)



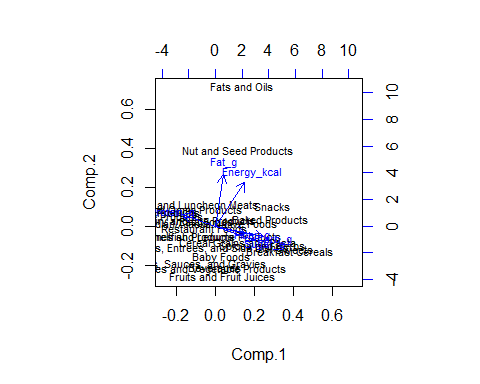
food\_data<- read.csv("project.csv")  
food\_data\_new2 <- food\_data[,c(2,23,24,25,26,27,28,29,30)]  
food\_group2 <- aggregate(.~FoodGroup, data=food\_data\_new, mean)  
labels <- food\_group2$FoodGroup  
food\_group2 <- food\_group2[,-1]  
rownames(food\_group2) <- labels  
head(food\_group2)

## Energy\_kcal Protein\_g Fat\_g  
## American Indian/Alaska Native Foods 209.1152 15.996061 11.1097576  
## Baby Foods 151.9917 4.114669 5.5804420  
## Baked Products 391.8871 6.654831 14.6679046  
## Beef Products 212.7495 24.369736 12.3613531  
## Beverages 111.3079 2.570159 0.9848889  
## Breakfast Cereals 342.4959 8.298375 4.0041873  
## Carb\_g Sugar\_g Fiber\_g  
## American Indian/Alaska Native Foods 11.2539394 1.358424 2.148484848  
## Baby Foods 21.5708840 12.675525 0.866022099  
## Baked Products 59.7672020 15.663112 3.044918444  
## Beef Products 0.1084567 0.000000 0.002854123  
## Beverages 18.6854286 10.709016 0.718412698  
## Breakfast Cereals 73.5282645 20.133085 7.738016529

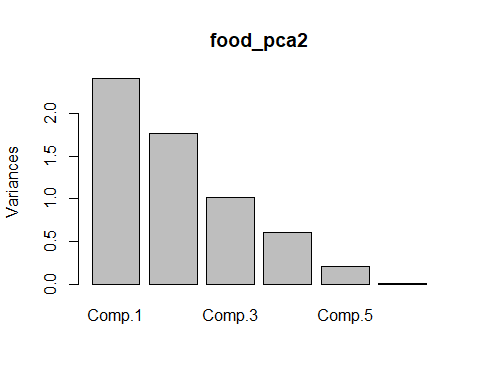
#cor\_food <- cor(food\_group)  
food\_group2\_scale<- scale(food\_group2)  
food\_pca2 <- princomp(food\_group2\_scale,cor=T)  
  
summary(food\_pca2, loadings = T)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  
## Standard deviation 1.5515373 1.3293332 1.0049777 0.7791812 0.45611868  
## Proportion of Variance 0.4012113 0.2945211 0.1683300 0.1011872 0.03467404  
## Cumulative Proportion 0.4012113 0.6957325 0.8640625 0.9652497 0.99992376  
## Comp.6  
## Standard deviation 2.138843e-02  
## Proportion of Variance 7.624413e-05  
## Cumulative Proportion 1.000000e+00  
##   
## Loadings:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6  
## Energy\_kcal 0.355 0.613 -0.183 0.201 -0.649  
## Protein\_g -0.404 0.163 0.466 -0.745 -0.130 0.144  
## Fat\_g 0.725 -0.128 0.186 -0.240 0.597  
## Carb\_g 0.583 -0.150 0.183 -0.250 0.586 0.446  
## Sugar\_g 0.481 -0.199 -0.412 -0.462 -0.588   
## Fiber\_g 0.362 0.748 0.317 -0.442

biplot(food\_pca2,col=c("black", "blue"), cex = 0.7, expand = 0.8)



screeplot(food\_pca2)



MULTIDIMENSIONAL SCALING

dist <- dist(scale(food.group))  
# apply MDS  
library(MVA)  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.3.3

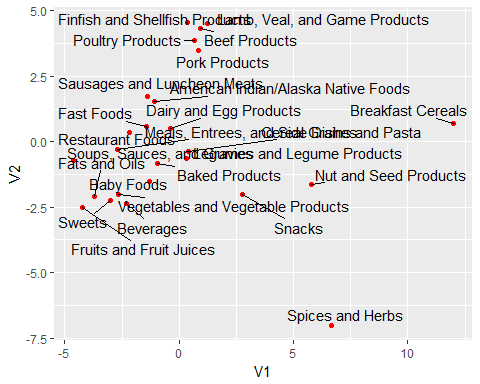
library(ggrepel)

## Warning: package 'ggrepel' was built under R version 3.3.3

food.mds = cmdscale(dist, eig = T)  
eignv <- abs(food.mds$eig)  
round(head(cumsum(eignv^2)/sum(eignv^2)), 2)

## [1] 0.63 0.82 0.91 0.96 0.99 0.99

scrs <- as.data.frame(food.mds$points)  
ggplot(scrs) + geom\_point(aes(x = V1, y = V2), color="red") + geom\_text\_repel(aes(V1, V2, label=rownames(scrs)), size = 4)

 Some food are near each other which suggests that they share some similarities across the nutritional ingredients. E.g., Lamb Products, Beef Products, Poultry Products, and Pork Products are close to each other and all have high calorie, protein, fat, folate, niacin and selenium.

Some group nature are indicated in the plot. E.g., Lamb Products , Beef Products, Poultry Products, and Pork Products are all types of meat products. Sweets, Beverages, and Fruits and Fruit Juices are all types of sweet food. Cereal Grains and Legume Products are all types of cereal food.

FACTOR ANALYSIS

Each factor captures a certain amount of the overall variance in the observed variables, and the factors are always listed in order of how much variation they explain.

library(psych)

## Warning: package 'psych' was built under R version 3.3.3

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library(GPArotation)  
  
  
#Finding the number of factors  
#This can be evaluated via methods such as `Parallel Analysis` and `eigenvalue`.  
  
parallel <- fa.parallel(food.group, fm = 'minres', fa = 'fa',quant = 0.95)

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.

## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

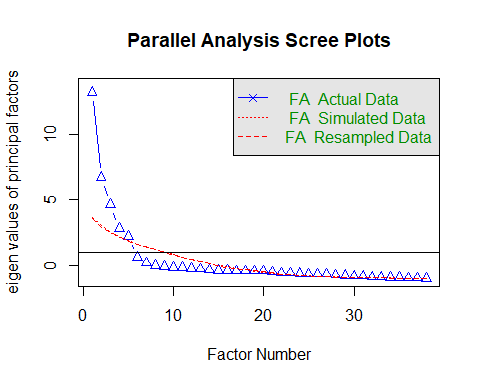
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done



## Parallel analysis suggests that the number of factors = 5 and the number of components = NA

#Blue line represents the eigen value  
  
factor\_5 <- fa(food.group,nfactors = 5,rotate = "oblimin",fm="minres", scores = "regression")

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

summary(factor\_5)

##   
## Factor analysis with Call: fa(r = food.group, nfactors = 5, rotate = "oblimin", scores = "regression",   
## fm = "minres")  
##   
## Test of the hypothesis that 5 factors are sufficient.  
## The degrees of freedom for the model is 523 and the objective function was 333.87   
## The number of observations was 25 with Chi Square = 2392.76 with prob < 3.3e-236   
##   
## The root mean square of the residuals (RMSA) is 0.04   
## The df corrected root mean square of the residuals is 0.05   
##   
## Tucker Lewis Index of factoring reliability = -0.169  
## RMSEA index = 0.772 and the 10 % confidence intervals are 0.37 NA  
## BIC = 709.29  
## With factor correlations of   
## MR1 MR3 MR2 MR4 MR5  
## MR1 1.00 0.29 0.20 0.06 0.15  
## MR3 0.29 1.00 0.01 0.19 -0.09  
## MR2 0.20 0.01 1.00 0.04 0.05  
## MR4 0.06 0.19 0.04 1.00 -0.07  
## MR5 0.15 -0.09 0.05 -0.07 1.00

print(factor\_5$loadings, cutoff = 0.5)

##   
## Loadings:  
## MR1 MR3 MR2 MR4 MR5   
## Energy\_kcal 0.895   
## Protein\_g 0.934   
## Fat\_g 0.994   
## Carb\_g 0.667   
## Sugar\_g -0.510   
## Fiber\_g 0.843   
## VitA\_mcg 0.881  
## VitB6\_mg 0.796   
## VitB12\_mcg 0.583   
## VitC\_mg 0.640   
## VitE\_mg 0.894   
## Folate\_mcg 0.987   
## Niacin\_mg 0.822   
## Riboflavin\_mg 0.861   
## Thiamin\_mg 0.931   
## Calcium\_mg 0.861   
## Copper\_mcg 0.644   
## Iron\_mg 0.700   
## Magnesium\_mg 0.819   
## Manganese\_mg 0.948   
## Phosphorus\_mg 0.769   
## Selenium\_mcg 0.928   
## Zinc\_mg 0.586   
## VitA\_USRDA 0.881  
## VitB6\_USRDA 0.796   
## VitB12\_USRDA 0.583   
## VitC\_USRDA 0.640   
## VitE\_USRDA 0.894   
## Folate\_USRDA 0.987   
## Niacin\_USRDA 0.822   
## Riboflavin\_USRDA 0.861   
## Thiamin\_USRDA 0.931   
## Calcium\_USRDA 0.861   
## Copper\_USRDA 0.644   
## Magnesium\_USRDA 0.819   
## Phosphorus\_USRDA 0.769   
## Selenium\_USRDA 0.928   
## Zinc\_USRDA 0.586   
##   
## MR1 MR3 MR2 MR4 MR5  
## SS loadings 9.379 7.170 7.076 4.100 3.441  
## Proportion Var 0.247 0.189 0.186 0.108 0.091  
## Cumulative Var 0.247 0.435 0.622 0.730 0.820

print(factor\_5$scores)

## MR1 MR3 MR2  
## American Indian/Alaska Native Foods -1.40882941 -2.46736908 0.6656731  
## Baby Foods -0.33143840 0.08747609 -0.6944528  
## Baked Products -12.05942670 -33.21043013 1.8643366  
## Beef Products -4.97018847 -14.17953928 2.3263362  
## Beverages 0.03474566 -0.51636022 -1.9683346  
## Breakfast Cereals 5.28406595 2.82305161 -0.4579788  
## Cereal Grains and Pasta 1.83361432 11.71681661 2.1366973  
## Dairy and Egg Products -2.79546844 -3.62183578 1.6659919  
## Fast Foods 2.35395446 10.95734128 1.4147522  
## Fats and Oils -1.57298642 -0.84564873 -0.6795244  
## Finfish and Shellfish Products -1.70843196 -4.49258150 1.9468109  
## Fruits and Fruit Juices -4.02277741 -4.01431796 0.8342632  
## Lamb, Veal, and Game Products 4.21235828 10.68817609 0.3306545  
## Legumes and Legume Products -2.40134759 -6.44317819 0.7747344  
## Meals, Entrees, and Side Dishes 8.40032423 3.35680802 -8.4989825  
## Nut and Seed Products -0.61061114 -2.16806595 1.7184432  
## Pork Products 1.98469073 6.58013298 1.6986265  
## Poultry Products -5.82453277 -13.60711538 3.1791181  
## Restaurant Foods 10.89260199 18.17752381 -6.4282058  
## Sausages and Luncheon Meats 6.69255910 15.43015334 -1.9979618  
## Snacks 0.87695052 1.40978792 -0.0044949  
## Soups, Sauces, and Gravies -10.27484516 -13.08109815 4.7798672  
## Spices and Herbs 0.79370313 5.32131253 -0.2832388  
## Sweets 4.90378312 13.96178905 -2.1711847  
## Vegetables and Vegetable Products -0.28246761 -1.86282898 -2.1519461  
## MR4 MR5  
## American Indian/Alaska Native Foods 2.2726350 3.4502032  
## Baby Foods -0.4560599 0.3001065  
## Baked Products 36.6844456 39.0940256  
## Beef Products 13.9614352 17.4258168  
## Beverages -2.0683436 0.4798808  
## Breakfast Cereals -2.8021184 -1.4861587  
## Cereal Grains and Pasta -5.4688671 -12.6768746  
## Dairy and Egg Products 6.7778358 6.0126258  
## Fast Foods -6.9103254 -13.1867172  
## Fats and Oils 5.0169270 1.9624021  
## Finfish and Shellfish Products 3.5399650 4.9242112  
## Fruits and Fruit Juices 8.5243337 6.1633152  
## Lamb, Veal, and Game Products -12.8141207 -11.8352013  
## Legumes and Legume Products 7.4523890 7.0452100  
## Meals, Entrees, and Side Dishes -25.0990616 -13.2648083  
## Nut and Seed Products 5.5450569 2.2634899  
## Pork Products -5.6149207 -8.1158389  
## Poultry Products 16.8060008 18.5318366  
## Restaurant Foods -33.4894105 -27.8231645  
## Sausages and Luncheon Meats -20.4771515 -20.6692094  
## Snacks 0.7485185 -1.2639689  
## Soups, Sauces, and Gravies 26.5104878 20.2179188  
## Spices and Herbs -0.9718248 -0.8870136  
## Sweets -16.3535356 -18.3629946  
## Vegetables and Vegetable Products -1.3142906 1.7009077

f5 <- fa(food.group,nfactors = 5)

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done  
  
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was  
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## The estimated weights for the factor scores are probably incorrect. Try a different factor extraction method.  
## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

f5

## Factor Analysis using method = minres  
## Call: fa(r = food.group, nfactors = 5)  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## MR1 MR3 MR2 MR4 MR5 h2 u2 com  
## Energy\_kcal 0.25 -0.12 0.01 0.90 -0.09 0.86 0.1410 1.2  
## Protein\_g -0.05 -0.06 0.93 -0.15 0.17 0.92 0.0759 1.1  
## Fat\_g -0.19 -0.17 0.09 0.99 0.16 1.00 0.0031 1.2  
## Carb\_g 0.67 0.17 -0.42 0.07 -0.41 0.74 0.2613 2.6  
## Sugar\_g 0.40 -0.04 -0.51 0.00 -0.19 0.35 0.6474 2.2  
## Fiber\_g 0.21 0.84 -0.13 0.02 -0.06 0.87 0.1279 1.2  
## VitA\_mcg 0.10 0.03 -0.04 0.31 0.88 0.87 0.1252 1.3  
## VitB6\_mg 0.80 0.23 0.12 0.00 0.22 0.94 0.0630 1.4  
## VitB12\_mcg 0.17 -0.22 0.58 -0.18 0.49 0.79 0.2108 2.7  
## VitC\_mg 0.04 0.64 -0.50 -0.26 0.18 0.69 0.3146 2.5  
## VitE\_mg 0.02 0.18 -0.16 0.89 0.20 0.92 0.0835 1.2  
## Folate\_mcg 0.99 0.01 -0.21 0.04 -0.01 0.95 0.0542 1.1  
## Niacin\_mg 0.82 -0.14 0.32 -0.10 0.19 0.93 0.0740 1.5  
## Riboflavin\_mg 0.86 0.08 0.06 -0.07 0.22 0.91 0.0887 1.2  
## Thiamin\_mg 0.93 -0.03 0.04 0.11 -0.30 0.90 0.1018 1.2  
## Calcium\_mg -0.05 0.86 -0.05 -0.04 0.13 0.71 0.2910 1.1  
## Copper\_mcg -0.05 0.64 0.47 0.23 -0.25 0.83 0.1749 2.5  
## Iron\_mg 0.41 0.70 -0.06 -0.05 0.25 0.87 0.1344 1.9  
## Magnesium\_mg 0.09 0.82 0.26 0.20 -0.23 0.99 0.0082 1.5  
## Manganese\_mg -0.09 0.95 -0.04 -0.04 0.07 0.84 0.1595 1.0  
## Phosphorus\_mg 0.18 0.28 0.77 0.12 -0.23 0.86 0.1359 1.6  
## Selenium\_mcg -0.02 -0.03 0.93 -0.04 -0.04 0.85 0.1464 1.0  
## Zinc\_mg 0.50 0.14 0.59 -0.02 0.23 0.86 0.1366 2.4  
## VitA\_USRDA 0.10 0.03 -0.04 0.31 0.88 0.87 0.1252 1.3  
## VitB6\_USRDA 0.80 0.23 0.12 0.00 0.22 0.94 0.0630 1.4  
## VitB12\_USRDA 0.17 -0.22 0.58 -0.18 0.49 0.79 0.2108 2.7  
## VitC\_USRDA 0.04 0.64 -0.50 -0.26 0.18 0.69 0.3146 2.5  
## VitE\_USRDA 0.02 0.18 -0.16 0.89 0.20 0.92 0.0835 1.2  
## Folate\_USRDA 0.99 0.01 -0.21 0.04 -0.01 0.95 0.0542 1.1  
## Niacin\_USRDA 0.82 -0.14 0.32 -0.10 0.19 0.93 0.0740 1.5  
## Riboflavin\_USRDA 0.86 0.08 0.06 -0.07 0.22 0.91 0.0887 1.2  
## Thiamin\_USRDA 0.93 -0.03 0.04 0.11 -0.30 0.90 0.1018 1.2  
## Calcium\_USRDA -0.05 0.86 -0.05 -0.04 0.13 0.71 0.2910 1.1  
## Copper\_USRDA -0.05 0.64 0.47 0.23 -0.25 0.83 0.1749 2.5  
## Magnesium\_USRDA 0.09 0.82 0.26 0.20 -0.23 0.99 0.0082 1.5  
## Phosphorus\_USRDA 0.18 0.28 0.77 0.12 -0.23 0.86 0.1359 1.6  
## Selenium\_USRDA -0.02 -0.03 0.93 -0.04 -0.04 0.85 0.1464 1.0  
## Zinc\_USRDA 0.50 0.14 0.59 -0.02 0.23 0.86 0.1366 2.4  
##   
## MR1 MR3 MR2 MR4 MR5  
## SS loadings 9.92 7.54 7.26 4.17 3.55  
## Proportion Var 0.26 0.20 0.19 0.11 0.09  
## Cumulative Var 0.26 0.46 0.65 0.76 0.85  
## Proportion Explained 0.31 0.23 0.22 0.13 0.11  
## Cumulative Proportion 0.31 0.54 0.76 0.89 1.00  
##   
## With factor correlations of   
## MR1 MR3 MR2 MR4 MR5  
## MR1 1.00 0.29 0.20 0.06 0.15  
## MR3 0.29 1.00 0.01 0.19 -0.09  
## MR2 0.20 0.01 1.00 0.04 0.05  
## MR4 0.06 0.19 0.04 1.00 -0.07  
## MR5 0.15 -0.09 0.05 -0.07 1.00  
##   
## Mean item complexity = 1.6  
## Test of the hypothesis that 5 factors are sufficient.  
##   
## The degrees of freedom for the null model are 703 and the objective function was 398.06 with Chi Square of 4179.67  
## The degrees of freedom for the model are 523 and the objective function was 333.87   
##   
## The root mean square of the residuals (RMSR) is 0.04   
## The df corrected root mean square of the residuals is 0.05   
##   
## The harmonic number of observations is 25 with the empirical chi square 63.68 with prob < 1   
## The total number of observations was 25 with Likelihood Chi Square = 2392.76 with prob < 3.3e-236   
##   
## Tucker Lewis Index of factoring reliability = -0.169  
## RMSEA index = 0.772 and the 90 % confidence intervals are 0.37 NA  
## BIC = 709.29  
## Fit based upon off diagonal values = 0.99

#Uniqueness for variable 1  
1 - sum(f5$loadings[1,]^2)

## [1] 0.1117016

From the Parallel analysis scree plot we can see that 5 factors will be sufficient to explain the all the variables.

Accuracy check: # There in no overlapping of variables in the factor. # RMSE = 0.04 # Goods of fit is 99%

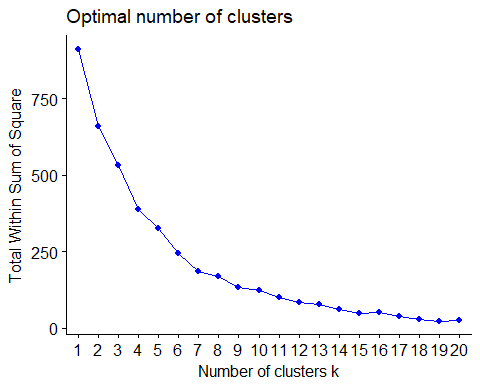
CLUSTER ANALSYSIS

#finding the number of clusters:  
scale\_food <- scale(food.group)  
  
# Elbow method  
library(factoextra)

## Warning: package 'factoextra' was built under R version 3.3.3

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

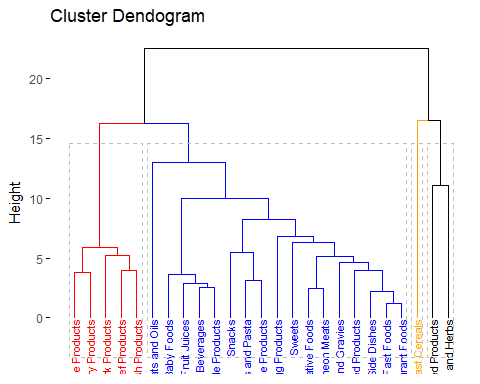
fviz\_nbclust(scale\_food, kmeans, method = "wss",verbose = interactive(),  
 linecolor = "blue",k.max = 20)



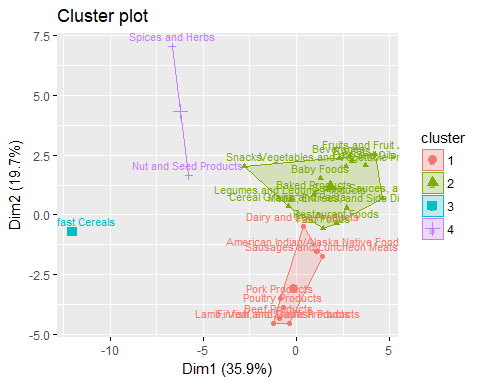
# optimum number of cluster is 4.  
  
  
# Dendogram:  
library(magrittr)

## Warning: package 'magrittr' was built under R version 3.3.3

res.hc <- food.group %>%  
 scale() %>% # Scale the data  
 dist(method = "euclidean") %>% # Compute dissimilarity matrix  
 hclust(method = "ward.D2")   
  
fviz\_dend(res.hc, k = 4, # Cut in four groups  
 cex = 0.6, # label size  
 k\_colors = c("red", "blue", "orange", "black"),  
 color\_labels\_by\_k = TRUE, # color labels by groups  
 rect = TRUE, # Add rectangle around groups  
 main = "Cluster Dendogram")



# K-means cluster:  
library("cluster")  
library("factoextra")  
k1 <- kmeans(scale\_food,4, nstart = 20)  
fviz\_cluster(k1, data = scale\_food,labelsize = 9,ggtheme = theme\_replace(),  
 ggtitle = "K-Means clustering")



From the elbow method we can see that the optimum number of clusters is 4. One cluster contains all the meat products group together. Cluster 2 contains all the vegetables, fruits products. Cluster 3 contain the nuts and seed products. Cluster 4 contains breakfast cereals.