2. Factor Analysis

Xin Tan, 583833

2019/3/11

# 1. NA imputation ----  
dat <- read\_excel("Data\_Chocolate\_allinterviews.xlsx", sheet = "AttributeRatingsStacked")  
dat<-as.data.frame(dat)  
str(dat)

## 'data.frame': 500 obs. of 15 variables:  
## $ Person : num 1 2 3 4 5 6 7 8 9 10 ...  
## $ Product : chr "Snickers" "Snickers" "Snickers" "Snickers" ...  
## $ crunchy : num 3 3 NA 4 3 2 4 5 5 3 ...  
## $ creamy : num 3 4 NA 4 3 3 4 5 1 4 ...  
## $ sweet : num 4 5 NA 5 4 4 4 5 4 5 ...  
## $ chocolaty : num 3 3 NA 5 3 3 4 5 5 5 ...  
## $ healthful : num 1 1 NA 1 1 1 2 1 1 1 ...  
## $ calorie : num 4 5 NA 4 4 5 4 5 5 5 ...  
## $ rich : num 4 5 NA 4 3 4 4 2 NA 5 ...  
## $ addiction : num 2 1 NA 3 2 3 3 1 NA 4 ...  
## $ accessible: num 4 5 NA 4 4 4 4 5 5 4 ...  
## $ handy : num 4 5 NA 4 4 5 NA 3 5 3 ...  
## $ wrapping : num 4 4 NA 4 2 3 3 5 5 3 ...  
## $ image : num 4 5 NA 4 4 5 4 5 5 5 ...  
## $ commercial: num 5 5 NA 5 4 5 4 NA NA 5 ...

bars<-dat  
summary(dat)

## Person Product crunchy creamy   
## Min. : 1.0 Length:500 Min. :1.000 Min. :1.000   
## 1st Qu.:13.0 Class :character 1st Qu.:2.000 1st Qu.:2.000   
## Median :25.5 Mode :character Median :4.000 Median :3.000   
## Mean :25.5 Mean :3.397 Mean :3.296   
## 3rd Qu.:38.0 3rd Qu.:5.000 3rd Qu.:4.000   
## Max. :50.0 Max. :5.000 Max. :5.000   
## NA's :21 NA's :20   
## sweet chocolaty healthful calorie   
## Min. :1.00 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:4.00 1st Qu.:3.000 1st Qu.:1.000 1st Qu.:4.000   
## Median :5.00 Median :4.000 Median :1.000 Median :5.000   
## Mean :4.49 Mean :3.969 Mean :1.624 Mean :4.378   
## 3rd Qu.:5.00 3rd Qu.:5.000 3rd Qu.:2.000 3rd Qu.:5.000   
## Max. :5.00 Max. :5.000 Max. :5.000 Max. :5.000   
## NA's :16 NA's :18 NA's :40 NA's :18   
## rich addiction accessible handy   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:3.000 1st Qu.:2.000 1st Qu.:4.000 1st Qu.:4.000   
## Median :4.000 Median :3.000 Median :4.000 Median :4.000   
## Mean :3.488 Mean :3.246 Mean :4.075 Mean :4.049   
## 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:5.000 3rd Qu.:5.000   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
## NA's :70 NA's :65 NA's :46 NA's :26   
## wrapping image commercial   
## Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:3.000 1st Qu.:3.000 1st Qu.:3.000   
## Median :4.000 Median :4.000 Median :4.000   
## Mean :3.607 Mean :3.868 Mean :3.767   
## 3rd Qu.:4.000 3rd Qu.:5.000 3rd Qu.:5.000   
## Max. :5.000 Max. :5.000 Max. :5.000   
## NA's :44 NA's :21 NA's :49

aggregate(dat[,-c(1,2)], by=list(bars$Product),mean, na.rm=TRUE)

## Group.1 crunchy creamy sweet chocolaty healthful calorie  
## 1 Balisto 4.465116 1.906977 4.045455 3.863636 2.279070 4.093023  
## 2 Bounty 1.978723 3.212766 4.437500 3.437500 1.659574 4.229167  
## 3 Duplo 4.020000 2.520000 4.480000 4.280000 1.632653 4.280000  
## 4 KinderBueno 3.770833 4.448980 4.571429 3.795918 1.488889 4.306122  
## 5 KinderRiegel 1.700000 4.060000 4.700000 4.520000 1.687500 4.300000  
## 6 KitKat 4.540000 2.160000 4.300000 4.140000 1.565217 4.280000  
## 7 Lion 4.186047 3.372093 4.568182 4.046512 1.568182 4.636364  
## 8 Mars 1.780000 4.320000 4.760000 3.860000 1.456522 4.560000  
## 9 Snickers 3.583333 3.437500 4.448980 3.958333 1.422222 4.604167  
## 10 Twix 4.140000 3.360000 4.540000 3.760000 1.510638 4.480000  
## rich addiction accessible handy wrapping image commercial  
## 1 3.421053 2.921053 3.487179 4.000000 3.414634 3.162791 3.179487  
## 2 3.534884 3.048780 4.022727 3.957447 3.911111 3.446809 2.904762  
## 3 3.093023 3.266667 3.978723 4.061224 3.553191 4.060000 4.367347  
## 4 3.476190 3.622222 3.833333 3.795918 3.729167 3.440000 3.000000  
## 5 3.444444 4.022222 4.468085 4.408163 4.000000 4.480000 4.446809  
## 6 3.266667 3.155556 4.145833 3.816327 3.531915 3.960000 3.895833  
## 7 3.820513 3.250000 3.800000 4.000000 3.550000 3.439024 3.205128  
## 8 3.666667 2.891304 4.250000 4.142857 3.489362 4.163265 3.978261  
## 9 3.600000 2.931818 4.347826 4.250000 3.276596 4.387755 4.347826  
## 10 3.577778 3.282609 4.276596 4.040816 3.595745 3.960000 4.063830

library(data.table)  
# use data.table  
bars.dt = as.data.table(bars)  
# melt data.table to "long" format  
bars.dt.long = melt(bars.dt, id.vars = c("Person", "Product"),   
 variable.name = "Attribute", value.name = "Value")   
  
str(bars.dt.long)

## Classes 'data.table' and 'data.frame': 6500 obs. of 4 variables:  
## $ Person : num 1 2 3 4 5 6 7 8 9 10 ...  
## $ Product : chr "Snickers" "Snickers" "Snickers" "Snickers" ...  
## $ Attribute: Factor w/ 13 levels "crunchy","creamy",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ Value : num 3 3 NA 4 3 2 4 5 5 3 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

head(bars.dt.long)

## Person Product Attribute Value  
## 1: 1 Snickers crunchy 3  
## 2: 2 Snickers crunchy 3  
## 3: 3 Snickers crunchy NA  
## 4: 4 Snickers crunchy 4  
## 5: 5 Snickers crunchy 3  
## 6: 6 Snickers crunchy 2

# mean values for each attribute (by each product and attribute)----  
# 加了一列：每个product对应attribute的平均分.为之后一步做准备  
bars.dt.long[, Mean.by.prod.att := mean(Value, na.rm = TRUE), by = .(Product, Attribute)]  
head(bars.dt.long)

## Person Product Attribute Value Mean.by.prod.att  
## 1: 1 Snickers crunchy 3 3.583333  
## 2: 2 Snickers crunchy 3 3.583333  
## 3: 3 Snickers crunchy NA 3.583333  
## 4: 4 Snickers crunchy 4 3.583333  
## 5: 5 Snickers crunchy 3 3.583333  
## 6: 6 Snickers crunchy 2 3.583333

# test  
# 每个product对应attribute的平均分[130\*13]  
unique(bars.dt.long[, .(Product, Attribute, Mean.by.prod.att)])

## Product Attribute Mean.by.prod.att  
## 1: Snickers crunchy 3.583333  
## 2: KinderBueno crunchy 3.770833  
## 3: Twix crunchy 4.140000  
## 4: Mars crunchy 1.780000  
## 5: KitKat crunchy 4.540000  
## ---   
## 126: Bounty commercial 2.904762  
## 127: KinderRiegel commercial 4.446809  
## 128: Balisto commercial 3.179487  
## 129: Lion commercial 3.205128  
## 130: Duplo commercial 4.367347

# impute NA----  
bars.dt.long[is.na(Value), Value := Mean.by.prod.att]  
bars.dt.2 = dcast(bars.dt.long, Person + Product ~ Attribute, value.var = "Value")  
  
head(bars.dt.2)

## Person Product crunchy creamy sweet chocolaty healthful calorie  
## 1: 1 Balisto 5 2 3 3 3 4  
## 2: 1 Bounty 2 4 5 3 2 4  
## 3: 1 Duplo 4 3 4 4 2 4  
## 4: 1 KinderBueno 4 4 5 4 1 5  
## 5: 1 KinderRiegel 3 3 4 5 3 4  
## 6: 1 KitKat 4 2 3 4 2 4  
## rich addiction accessible handy wrapping image commercial  
## 1: 4 3 3 3 2 3 3  
## 2: 3 3 3 4 4 3 3  
## 3: 4 4 4 4 4 3 3  
## 4: 4 3 3 2 3 4 1  
## 5: 4 5 4 4 4 4 3  
## 6: 3 3 4 4 4 4 4

summary(bars.dt.2)

## Person Product crunchy creamy   
## Min. : 1.0 Length:500 Min. :1.000 Min. :1.00   
## 1st Qu.:13.0 Class :character 1st Qu.:2.000 1st Qu.:2.00   
## Median :25.5 Mode :character Median :4.000 Median :3.00   
## Mean :25.5 Mean :3.416 Mean :3.28   
## 3rd Qu.:38.0 3rd Qu.:5.000 3rd Qu.:4.00   
## Max. :50.0 Max. :5.000 Max. :5.00   
## sweet chocolaty healthful calorie   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:4.000 1st Qu.:3.328 1st Qu.:1.000 1st Qu.:4.000   
## Median :5.000 Median :4.000 Median :1.000 Median :4.636   
## Mean :4.485 Mean :3.966 Mean :1.627 Mean :4.377   
## 3rd Qu.:5.000 3rd Qu.:5.000 3rd Qu.:2.000 3rd Qu.:5.000   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
## rich addiction accessible handy   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:3.000 1st Qu.:2.921 1st Qu.:3.979 1st Qu.:4.000   
## Median :3.476 Median :3.000 Median :4.000 Median :4.000   
## Mean :3.490 Mean :3.239 Mean :4.061 Mean :4.047   
## 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:5.000 3rd Qu.:5.000   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
## wrapping image commercial   
## Min. :1.000 Min. :1.00 Min. :1.000   
## 1st Qu.:3.000 1st Qu.:3.00 1st Qu.:3.000   
## Median :4.000 Median :4.00 Median :4.000   
## Mean :3.605 Mean :3.85 Mean :3.739   
## 3rd Qu.:4.000 3rd Qu.:5.00 3rd Qu.:5.000   
## Max. :5.000 Max. :5.00 Max. :5.000

attribute <- bars.dt.2  
## attribute is the dataset to be used to do 'Reducing Data Complexity'. Cleaned!  
  
#### End of the NA imputation  
  
  
  
# bars.gender  
socio<-read\_excel("Data\_Chocolate\_allinterviews.xlsx", sheet = "Social Demographic Questions")  
socio<-as.data.frame(socio)  
  
str(socio)

## 'data.frame': 50 obs. of 9 variables:  
## $ Person : num 1 2 3 4 5 6 7 8 9 10 ...  
## $ 28. What is your gender? : chr "male" "female" "female" "female" ...  
## $ 29. How old are you? : num 27 25 25 24 21 18 29 20 29 27 ...  
## $ 30. What is your main occupation? : chr "I am an employee." "I am an employee." "I am a student." "I am a student." ...  
## $ 31. What is your marital status? : chr "single" "single" "es ist kompliziert" "single" ...  
## $ 32. Do you have children? If yes, how many? : chr "0" "0" "not report" "0" ...  
## $ 33. Where do you live? : chr "City" "City" "City" "Village" ...  
## $ 34. In which state of Germany do you live? : chr "Sachsen-Anhalt" "Berlin" "Sachsen-Anhalt" "Sachsen-Anhalt" ...  
## $ 35. How often do you practice any kind of sport?: chr "1-3 times a week" "1-3 times a week" "4-7 times a week" "4-7 times a week" ...

library(plyr)  
socio<-rename(socio,c("28. What is your gender?"="Gender"))  
  
  
bars<-merge(bars,socio,by="Person")  
str(bars)

## 'data.frame': 500 obs. of 23 variables:  
## $ Person : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ Product : chr "Snickers" "KinderRiegel" "Balisto" "Mars" ...  
## $ crunchy : num 3 3 5 2 2 4 4 4 4 3 ...  
## $ creamy : num 3 3 2 4 4 2 4 3 4 4 ...  
## $ sweet : num 4 4 3 4 5 3 5 4 4 4 ...  
## $ chocolaty : num 3 5 3 4 3 4 4 4 4 4 ...  
## $ healthful : num 1 3 3 2 2 2 1 2 1 2 ...  
## $ calorie : num 4 4 4 4 4 4 5 4 4 4 ...  
## $ rich : num 4 4 4 4 3 3 4 4 4 4 ...  
## $ addiction : num 2 5 3 4 3 3 3 4 4 4 ...  
## $ accessible : num 4 4 3 4 3 4 3 4 4 4 ...  
## $ handy : num 4 4 3 4 4 4 2 4 4 4 ...  
## $ wrapping : num 4 4 2 4 4 4 3 4 4 4 ...  
## $ image : num 4 4 3 3 3 4 4 3 4 4 ...  
## $ commercial : num 5 3 3 3 3 4 1 3 4 3 ...  
## $ Gender : chr "male" "male" "male" "male" ...  
## $ 29. How old are you? : num 27 27 27 27 27 27 27 27 27 27 ...  
## $ 30. What is your main occupation? : chr "I am an employee." "I am an employee." "I am an employee." "I am an employee." ...  
## $ 31. What is your marital status? : chr "single" "single" "single" "single" ...  
## $ 32. Do you have children? If yes, how many? : chr "0" "0" "0" "0" ...  
## $ 33. Where do you live? : chr "City" "City" "City" "City" ...  
## $ 34. In which state of Germany do you live? : chr "Sachsen-Anhalt" "Sachsen-Anhalt" "Sachsen-Anhalt" "Sachsen-Anhalt" ...  
## $ 35. How often do you practice any kind of sport?: chr "1-3 times a week" "1-3 times a week" "1-3 times a week" "1-3 times a week" ...

bars.gender<-aggregate(bars[,-c(1,2,16:23)], by=list(bars$Product,bars$Gender),mean, na.rm=TRUE)  
str(bars.gender)

## 'data.frame': 20 obs. of 15 variables:  
## $ Group.1 : chr "Balisto" "Bounty" "Duplo" "KinderBueno" ...  
## $ Group.2 : chr "female" "female" "female" "female" ...  
## $ crunchy : num 4.43 1.79 4.14 3.93 1.69 ...  
## $ creamy : num 1.86 3.39 2.31 4.45 3.97 ...  
## $ sweet : num 4.11 4.59 4.55 4.66 4.66 ...  
## $ chocolaty : num 3.86 3.31 4.31 3.62 4.52 ...  
## $ healthful : num 2.37 1.64 1.64 1.6 1.52 ...  
## $ calorie : num 4 4.34 4.38 4.28 4.34 ...  
## $ rich : num 3.36 3.46 2.92 3.35 3.42 ...  
## $ addiction : num 2.79 2.96 3.19 3.78 4.08 ...  
## $ accessible: num 3.61 4.24 4.15 4 4.62 ...  
## $ handy : num 4.07 3.97 4.17 4.03 4.55 ...  
## $ wrapping : num 3.75 4.12 3.65 3.93 4.27 ...  
## $ image : num 3.19 3.43 4.07 3.28 4.59 ...  
## $ commercial: num 3.36 2.84 4.48 2.83 4.61 ...

need to run part 1 before run the following code

## 1. calculate euclidian distances

No need to rescaling the Data, otherwise problems might arise as calculating the Euclidean distance

setwd("~/Desktop/DA2/Final\_DA2")  
products.mean <- aggregate(.~Product, data = attribute, mean)  
  
rownames(products.mean ) <- products.mean [, 1] # use brand for the row names  
products.mean <- products.mean [, -c(1,2)] # remove brand name column  
products.mean

## crunchy creamy sweet chocolaty healthful calorie  
## Balisto 4.465116 1.906977 4.045455 3.863636 2.279070 4.093023  
## Bounty 1.978723 3.212766 4.437500 3.437500 1.659574 4.229167  
## Duplo 4.020000 2.520000 4.480000 4.280000 1.632653 4.280000  
## KinderBueno 3.770833 4.448980 4.571429 3.795918 1.488889 4.306122  
## KinderRiegel 1.700000 4.060000 4.700000 4.520000 1.687500 4.300000  
## KitKat 4.540000 2.160000 4.300000 4.140000 1.565217 4.280000  
## Lion 4.186047 3.372093 4.568182 4.046512 1.568182 4.636364  
## Mars 1.780000 4.320000 4.760000 3.860000 1.456522 4.560000  
## Snickers 3.583333 3.437500 4.448980 3.958333 1.422222 4.604167  
## Twix 4.140000 3.360000 4.540000 3.760000 1.510638 4.480000  
## rich addiction accessible handy wrapping image  
## Balisto 3.421053 2.921053 3.487179 4.000000 3.414634 3.162791  
## Bounty 3.534884 3.048780 4.022727 3.957447 3.911111 3.446809  
## Duplo 3.093023 3.266667 3.978723 4.061224 3.553191 4.060000  
## KinderBueno 3.476190 3.622222 3.833333 3.795918 3.729167 3.440000  
## KinderRiegel 3.444444 4.022222 4.468085 4.408163 4.000000 4.480000  
## KitKat 3.266667 3.155556 4.145833 3.816327 3.531915 3.960000  
## Lion 3.820513 3.250000 3.800000 4.000000 3.550000 3.439024  
## Mars 3.666667 2.891304 4.250000 4.142857 3.489362 4.163265  
## Snickers 3.600000 2.931818 4.347826 4.250000 3.276596 4.387755  
## Twix 3.577778 3.282609 4.276596 4.040816 3.595745 3.960000  
## commercial  
## Balisto 3.179487  
## Bounty 2.904762  
## Duplo 4.367347  
## KinderBueno 3.000000  
## KinderRiegel 4.446809  
## KitKat 3.895833  
## Lion 3.205128  
## Mars 3.978261  
## Snickers 4.347826  
## Twix 4.063830

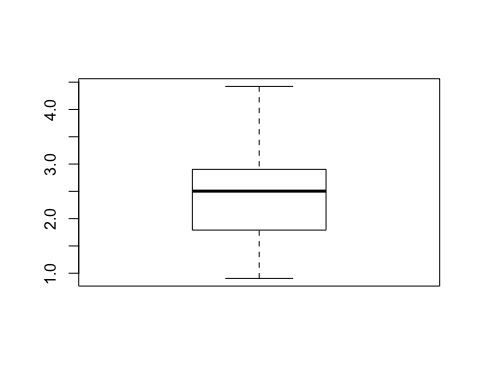
# 表similarity matrix/ calculate euclidian distances----  
  
products.dist <- dist(products.mean) # 算得是均值的Euclidean distance  
products.dist

## Balisto Bounty Duplo KinderBueno KinderRiegel KitKat  
## Bounty 3.057018   
## Duplo 2.023991 2.874081   
## KinderBueno 2.959558 2.314692 2.587829   
## KinderRiegel 4.422568 2.617738 3.049584 3.034293   
## KitKat 1.569771 3.114505 0.905386 2.738831 3.745032   
## Lion 1.947468 2.419498 1.802404 1.372694 3.346816 1.737796  
## Mars 4.092170 1.915837 3.055841 2.547024 1.631278 3.613951  
## Snickers 2.829710 2.583265 1.418313 2.252880 2.504016 1.889563  
## Twix 2.333038 2.593565 1.215404 1.776849 2.902997 1.434470  
## Lion Mars Snickers  
## Bounty   
## Duplo   
## KinderBueno   
## KinderRiegel   
## KitKat   
## Lion   
## Mars 2.877263   
## Snickers 1.790236 2.097552   
## Twix 1.188520 2.605959 0.961563

sm<-as.matrix(products.dist)  
write.csv(sm, file = "DistanceMatrix.csv")  
  
summary(products.dist)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.9054 1.7902 2.5040 2.3945 2.9030 4.4226

boxplot(products.dist)  
  
library("psych")



describe(products.dist)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 45 2.39 0.81 2.5 2.37 0.82 0.91 4.42 3.52 0.24 -0.38  
## se  
## X1 0.12

## 2.Suitability of factor analysis

Assumption: the model assumes that some factors linearly influence the observed model.Check beforehand:

* inverse correlation matrix
* Anti-image matrix
* Kaiser-Meyer-Olkin-Criteria
* Bartlett’s test of sphericity

Check afterwards

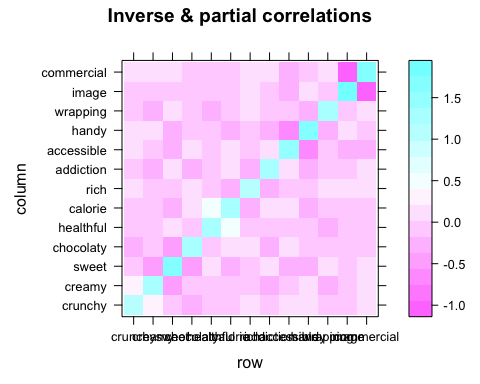
* communalities h2
* reproduced correlation matrix R\_hat

Factor analysis is based on a covariance matrix between variables. In other words, the candidate variables must have a certain correlation. If there is no correlation between the variables, or the correlation is small, the factor analysis will not be a suitable analysis method. The Kaiser-Meyer-Olkin measure of sampling adequacy allows us to know, whether this dataset is suitable for the degree of factor analysis?

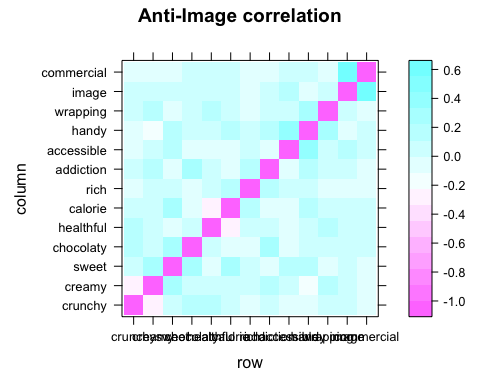
# Check beforehand  
  
library("psych")  
library("lattice")  
  
  
names(attribute)

## [1] "Person" "Product" "crunchy" "creamy" "sweet"   
## [6] "chocolaty" "healthful" "calorie" "rich" "addiction"   
## [11] "accessible" "handy" "wrapping" "image" "commercial"

# inverse and partial correlations  
  
p <- solve(cor(attribute[,3:15], use="complete.obs"))  
levelplot(p, main="Inverse & partial correlations")



# anti-image  
pr <- -p/sqrt(outer(diag(p), diag(p)))  
levelplot(pr, main="Anti-Image correlation")



# Kaiser-Meyer-Olkin & MSA  
KMO(attribute[,3:15])

## Kaiser-Meyer-Olkin factor adequacy  
## Call: KMO(r = attribute[, 3:15])  
## Overall MSA = 0.7  
## MSA for each item =   
## crunchy creamy sweet chocolaty healthful calorie   
## 0.45 0.65 0.75 0.71 0.55 0.65   
## rich addiction accessible handy wrapping image   
## 0.74 0.76 0.79 0.74 0.77 0.69   
## commercial   
## 0.66

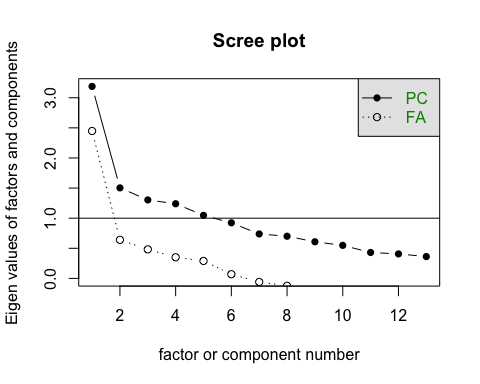
##KMO检验的数值变化从0——1，一般来说，KMO大于0.9适合作因子分析，若国小，表明变量偶对之间的相关不能被其他变量解释，进行因子分析不合适。KMO的值为0.7时为“还好’,0.6时为”中等”，0，5时就为“糟糕”了。本例中为近似0.7，表示还可以做因子分析。  
  
# Bartlett test of sphericity  
cortest.bartlett(attribute[,3:15])

## R was not square, finding R from data

## $chisq  
## [1] 1235.126  
##   
## $p.value  
## [1] 1.414912e-207  
##   
## $df  
## [1] 78

## wiki: Bartlett's test (see Snedecor and Cochran, 1989) is used to test if k samples are from populations with equal variances. Equal variances across populations is called homoscedasticity or homogeneity of variances. Some statistical tests, for example the analysis of variance, assume that variances are equal across groups or samples. The Bartlett test can be used to verify that assumption.  
  
  
## Bartlett检验的目的是确定所要求的数据是否曲子多元正态分布的总体，若差异检验的F值显著，表示索取数据来自正态分布，可以做进一步的分析。你给的结果中，sig显著，表示数据取自正态分布，很适合做因子分析。

# 几种不同的因子分析解释度。我应该会用ml  
  
library("psych")  
scree(attribute[,3:15])



# principal component extraction  
principal(attribute[,3:15], nfactors=4, rotate="none")

## Principal Components Analysis  
## Call: principal(r = attribute[, 3:15], nfactors = 4, rotate = "none")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## PC1 PC2 PC3 PC4 h2 u2 com  
## crunchy -0.06 0.37 0.22 0.74 0.73 0.27 1.7  
## creamy 0.46 -0.26 0.12 -0.49 0.53 0.47 2.7  
## sweet 0.64 -0.36 0.15 0.16 0.59 0.41 1.8  
## chocolaty 0.47 0.09 0.31 0.41 0.49 0.51 2.8  
## healthful 0.17 0.71 0.28 -0.23 0.67 0.33 1.7  
## calorie 0.40 -0.60 -0.13 0.32 0.64 0.36 2.5  
## rich 0.37 -0.23 0.37 -0.05 0.32 0.68 2.7  
## addiction 0.52 0.12 0.43 0.01 0.47 0.53 2.1  
## accessible 0.66 0.02 -0.16 -0.11 0.47 0.53 1.2  
## handy 0.67 0.02 0.08 0.00 0.46 0.54 1.0  
## wrapping 0.47 0.29 0.12 -0.28 0.40 0.60 2.6  
## image 0.60 0.28 -0.53 0.06 0.72 0.28 2.4  
## commercial 0.53 0.26 -0.61 0.14 0.74 0.26 2.5  
##   
## PC1 PC2 PC3 PC4  
## SS loadings 3.19 1.50 1.30 1.24  
## Proportion Var 0.25 0.12 0.10 0.10  
## Cumulative Var 0.25 0.36 0.46 0.56  
## Proportion Explained 0.44 0.21 0.18 0.17  
## Cumulative Proportion 0.44 0.65 0.83 1.00  
##   
## Mean item complexity = 2.1  
## Test of the hypothesis that 4 components are sufficient.  
##   
## The root mean square of the residuals (RMSR) is 0.09   
## with the empirical chi square 681.06 with prob < 9.9e-123   
##   
## Fit based upon off diagonal values = 0.8

# principal axis extraction  
fa(attribute[,3:15], nfactors=4, rotate="none", fm="pa")

## maximum iteration exceeded

## Factor Analysis using method = pa  
## Call: fa(r = attribute[, 3:15], nfactors = 4, rotate = "none", fm = "pa")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## PA1 PA2 PA3 PA4 h2 u2 com  
## crunchy -0.06 0.49 0.54 0.39 0.69 0.31 2.9  
## creamy 0.39 -0.24 -0.06 -0.22 0.26 0.74 2.4  
## sweet 0.59 -0.30 0.22 0.09 0.49 0.51 1.8  
## chocolaty 0.39 0.06 0.25 0.05 0.22 0.78 1.8  
## healthful 0.15 0.47 0.10 -0.41 0.42 0.58 2.3  
## calorie 0.37 -0.38 0.08 0.37 0.43 0.57 3.1  
## rich 0.29 -0.14 0.16 -0.08 0.14 0.86 2.3  
## addiction 0.44 0.04 0.21 -0.17 0.27 0.73 1.8  
## accessible 0.58 -0.01 -0.03 -0.05 0.34 0.66 1.0  
## handy 0.60 -0.03 0.14 -0.09 0.39 0.61 1.2  
## wrapping 0.39 0.11 0.05 -0.24 0.22 0.78 1.9  
## image 0.60 0.30 -0.35 0.17 0.60 0.40 2.4  
## commercial 0.55 0.31 -0.41 0.27 0.64 0.36 3.1  
##   
## PA1 PA2 PA3 PA4  
## SS loadings 2.59 0.98 0.81 0.73  
## Proportion Var 0.20 0.08 0.06 0.06  
## Cumulative Var 0.20 0.27 0.34 0.39  
## Proportion Explained 0.51 0.19 0.16 0.14  
## Cumulative Proportion 0.51 0.70 0.86 1.00  
##   
## Mean item complexity = 2.1  
## Test of the hypothesis that 4 factors are sufficient.  
##   
## The degrees of freedom for the null model are 78 and the objective function was 2.5 with Chi Square of 1235.13  
## The degrees of freedom for the model are 32 and the objective function was 0.34   
##   
## The root mean square of the residuals (RMSR) is 0.04   
## The df corrected root mean square of the residuals is 0.07   
##   
## The harmonic number of observations is 500 with the empirical chi square 152.71 with prob < 1.1e-17   
## The total number of observations was 500 with Likelihood Chi Square = 168.64 with prob < 1.7e-20   
##   
## Tucker Lewis Index of factoring reliability = 0.71  
## RMSEA index = 0.093 and the 90 % confidence intervals are 0.079 0.107  
## BIC = -30.22  
## Fit based upon off diagonal values = 0.96  
## Measures of factor score adequacy   
## PA1 PA2 PA3 PA4  
## Correlation of (regression) scores with factors 0.91 0.82 0.81 0.77  
## Multiple R square of scores with factors 0.82 0.68 0.66 0.59  
## Minimum correlation of possible factor scores 0.64 0.36 0.33 0.19

# maximum likelihood extraction  
fa(attribute[,3:15], nfactors=4, rotate="none", fm="ml")

## Factor Analysis using method = ml  
## Call: fa(r = attribute[, 3:15], nfactors = 4, rotate = "none", fm = "ml")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## ML1 ML2 ML3 ML4 h2 u2 com  
## crunchy -0.03 -0.04 0.11 0.18 0.05 0.95 1.9  
## creamy 0.32 0.19 -0.03 0.12 0.15 0.85 1.9  
## sweet 0.52 0.44 -0.17 0.18 0.52 0.48 2.5  
## chocolaty 0.37 0.20 0.08 0.46 0.39 0.61 2.4  
## healthful 0.15 -0.15 0.59 0.10 0.40 0.60 1.3  
## calorie 0.32 0.30 -0.45 -0.03 0.40 0.60 2.7  
## rich 0.23 0.26 0.04 0.00 0.12 0.88 2.0  
## addiction 0.38 0.18 0.22 0.23 0.28 0.72 2.9  
## accessible 0.60 0.17 0.10 -0.32 0.50 0.50 1.8  
## handy 0.57 0.34 0.20 -0.24 0.55 0.45 2.3  
## wrapping 0.37 0.08 0.29 -0.08 0.23 0.77 2.1  
## image 0.73 -0.44 -0.09 0.03 0.73 0.27 1.7  
## commercial 0.61 -0.36 -0.11 -0.01 0.52 0.48 1.7  
##   
## ML1 ML2 ML3 ML4  
## SS loadings 2.56 0.97 0.80 0.52  
## Proportion Var 0.20 0.07 0.06 0.04  
## Cumulative Var 0.20 0.27 0.33 0.37  
## Proportion Explained 0.53 0.20 0.16 0.11  
## Cumulative Proportion 0.53 0.73 0.89 1.00  
##   
## Mean item complexity = 2.1  
## Test of the hypothesis that 4 factors are sufficient.  
##   
## The degrees of freedom for the null model are 78 and the objective function was 2.5 with Chi Square of 1235.13  
## The degrees of freedom for the model are 32 and the objective function was 0.32   
##   
## The root mean square of the residuals (RMSR) is 0.05   
## The df corrected root mean square of the residuals is 0.07   
##   
## The harmonic number of observations is 500 with the empirical chi square 175.67 with prob < 9.4e-22   
## The total number of observations was 500 with Likelihood Chi Square = 156.62 with prob < 2.4e-18   
##   
## Tucker Lewis Index of factoring reliability = 0.736  
## RMSEA index = 0.089 and the 90 % confidence intervals are 0.075 0.102  
## BIC = -42.25  
## Fit based upon off diagonal values = 0.95  
## Measures of factor score adequacy   
## ML1 ML2 ML3 ML4  
## Correlation of (regression) scores with factors 0.92 0.83 0.76 0.69  
## Multiple R square of scores with factors 0.85 0.68 0.57 0.47  
## Minimum correlation of possible factor scores 0.70 0.36 0.14 -0.05

# unweighted least squares extraction  
fa(attribute[,3:15], nfactors=4, rotate="none")

## Factor Analysis using method = minres  
## Call: fa(r = attribute[, 3:15], nfactors = 4, rotate = "none")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## MR1 MR2 MR3 MR4 h2 u2 com  
## crunchy -0.07 0.85 0.39 -0.15 0.91 0.09 1.5  
## creamy 0.38 -0.26 0.02 0.16 0.24 0.76 2.1  
## sweet 0.59 -0.15 0.34 -0.03 0.49 0.51 1.8  
## chocolaty 0.39 0.15 0.18 0.06 0.21 0.79 1.8  
## healthful 0.15 0.34 -0.26 0.46 0.42 0.58 2.8  
## calorie 0.37 -0.22 0.35 -0.35 0.43 0.57 3.6  
## rich 0.29 -0.08 0.18 0.12 0.14 0.86 2.3  
## addiction 0.44 0.08 0.10 0.24 0.27 0.73 1.8  
## accessible 0.58 -0.03 -0.03 0.04 0.34 0.66 1.0  
## handy 0.60 0.01 0.11 0.14 0.39 0.61 1.2  
## wrapping 0.39 0.07 -0.07 0.25 0.22 0.78 1.9  
## image 0.60 0.16 -0.38 -0.26 0.60 0.40 2.3  
## commercial 0.55 0.17 -0.42 -0.38 0.64 0.36 2.9  
##   
## MR1 MR2 MR3 MR4  
## SS loadings 2.59 1.08 0.88 0.75  
## Proportion Var 0.20 0.08 0.07 0.06  
## Cumulative Var 0.20 0.28 0.35 0.41  
## Proportion Explained 0.49 0.20 0.17 0.14  
## Cumulative Proportion 0.49 0.69 0.86 1.00  
##   
## Mean item complexity = 2.1  
## Test of the hypothesis that 4 factors are sufficient.  
##   
## The degrees of freedom for the null model are 78 and the objective function was 2.5 with Chi Square of 1235.13  
## The degrees of freedom for the model are 32 and the objective function was 0.34   
##   
## The root mean square of the residuals (RMSR) is 0.04   
## The df corrected root mean square of the residuals is 0.07   
##   
## The harmonic number of observations is 500 with the empirical chi square 152.47 with prob < 1.3e-17   
## The total number of observations was 500 with Likelihood Chi Square = 168.07 with prob < 2.2e-20   
##   
## Tucker Lewis Index of factoring reliability = 0.712  
## RMSEA index = 0.093 and the 90 % confidence intervals are 0.079 0.106  
## BIC = -30.79  
## Fit based upon off diagonal values = 0.96  
## Measures of factor score adequacy   
## MR1 MR2 MR3 MR4  
## Correlation of (regression) scores with factors 0.91 0.93 0.83 0.77  
## Multiple R square of scores with factors 0.82 0.86 0.69 0.59  
## Minimum correlation of possible factor scores 0.64 0.72 0.38 0.18

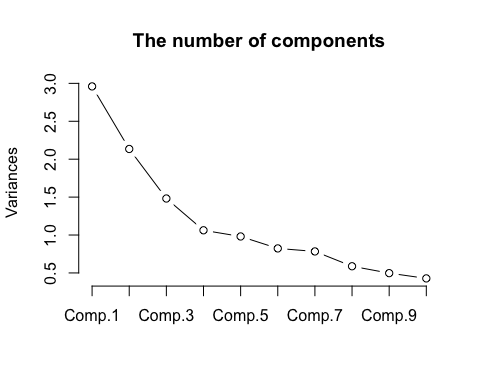
# ML with Kaiser normalization  
factanal(attribute[,3:15], factors=3)

##   
## Call:  
## factanal(x = attribute[, 3:15], factors = 3)  
##   
## Uniquenesses:  
## crunchy creamy sweet chocolaty healthful calorie   
## 0.985 0.841 0.531 0.834 0.570 0.590   
## rich addiction accessible handy wrapping image   
## 0.865 0.761 0.646 0.587 0.772 0.415   
## commercial   
## 0.334   
##   
## Loadings:  
## Factor1 Factor2 Factor3  
## crunchy -0.121   
## creamy 0.376 0.111   
## sweet 0.582 0.347   
## chocolaty 0.385 0.129   
## healthful 0.232 -0.605   
## calorie 0.268 0.114 0.571   
## rich 0.355   
## addiction 0.472   
## accessible 0.511 0.299   
## handy 0.617 0.172   
## wrapping 0.420 0.132 -0.185   
## image 0.232 0.729   
## commercial 0.123 0.806   
##   
## Factor1 Factor2 Factor3  
## SS loadings 1.991 1.381 0.899  
## Proportion Var 0.153 0.106 0.069  
## Cumulative Var 0.153 0.259 0.329  
##   
## Test of the hypothesis that 3 factors are sufficient.  
## The chi square statistic is 235.17 on 42 degrees of freedom.  
## The p-value is 1.08e-28

#eigen(cor(attribute[,3:15]))  
n <- princomp(attribute[,3:15])  
summary(n)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  
## Standard deviation 1.7204345 1.4610705 1.2172092 1.03073944 0.99043449  
## Proportion of Variance 0.2338307 0.1686427 0.1170458 0.08393114 0.07749556  
## Cumulative Proportion 0.2338307 0.4024735 0.5195193 0.60345040 0.68094596  
## Comp.6 Comp.7 Comp.8 Comp.9  
## Standard deviation 0.90739733 0.88517380 0.76696071 0.70527528  
## Proportion of Variance 0.06504596 0.06189883 0.04646988 0.03929548  
## Cumulative Proportion 0.74599192 0.80789075 0.85436063 0.89365611  
## Comp.10 Comp.11 Comp.12 Comp.13  
## Standard deviation 0.65418838 0.61495843 0.54529455 0.49259338  
## Proportion of Variance 0.03380889 0.02987561 0.02349025 0.01916913  
## Cumulative Proportion 0.92746500 0.95734062 0.98083087 1.00000000

plot(n,type = "l", main = "The number of components")

 The proportion of variance decreases as the number of the components increase, so here I would choose 4 components

## 3. How many factors to choose? large increase of explained variance in at least two or three items

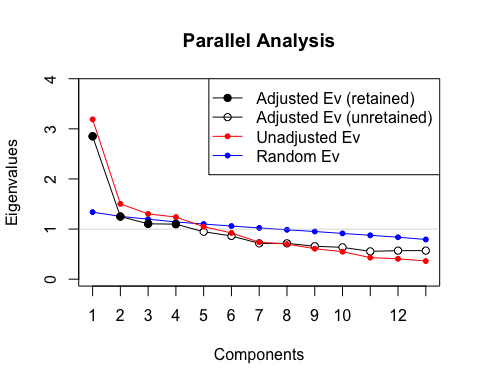
DA2 Parallel analysis of Horn 根据Horn, 选择4个component比较合适，不过3个也可以，因为多一个的提升也不大 Adjusted eigenvalues > 1 indicate dimensions to retain. (4 components retained)

## 可用/ Test/ DA2 Parallel analysis of Horn ----  
library("foreign")  
library("paran")

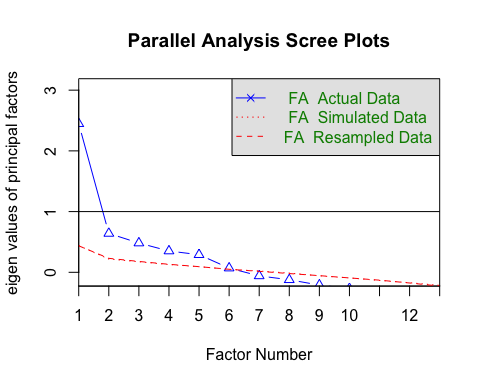
## Loading required package: MASS

x <-attribute[,-c(1,2)]  
paran(x, centile=95, all=T, graph=T)

##   
## Using eigendecomposition of correlation matrix.  
## Computing: 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%  
##   
##   
## Results of Horn's Parallel Analysis for component retention  
## 390 iterations, using the 95 centile estimate  
##   
## --------------------------------------------------   
## Component Adjusted Unadjusted Estimated   
## Eigenvalue Eigenvalue Bias   
## --------------------------------------------------   
## 1 2.850680 3.188346 0.337666  
## 2 1.250217 1.503445 0.253227  
## 3 1.105232 1.302805 0.197573  
## 4 1.097812 1.239937 0.142125  
## 5 0.946767 1.046915 0.100147  
## 6 0.863010 0.922971 0.059960  
## 7 0.714898 0.738174 0.023275  
## 8 0.714980 0.700422 -0.01455  
## 9 0.657092 0.608501 -0.04859  
## 10 0.635348 0.548641 -0.08670  
## 11 0.555037 0.430693 -0.12434  
## 12 0.569449 0.406696 -0.16275  
## 13 0.569997 0.362448 -0.20754  
## --------------------------------------------------   
##   
## Adjusted eigenvalues > 1 indicate dimensions to retain.  
## (4 components retained)



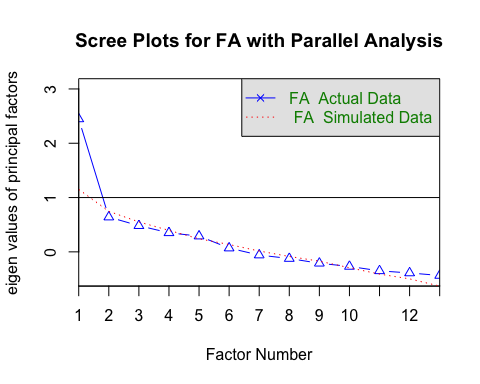
library("psych")  
fa.parallel(x, fa="fa")



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

## 根据Horn, 选择4个component比较合适，不过课堂上老师认为3个比较好  
## Adjusted eigenvalues > 1 indicate dimensions to retain. (4 components retained)  
  
# Scree Plots For FA with Parallel Ananlysis  
tmp <- attribute[,3:15]  
fa.parallel(tmp, n.obs=50, fa="fa",   
 main="Scree Plots for FA with Parallel Analysis")

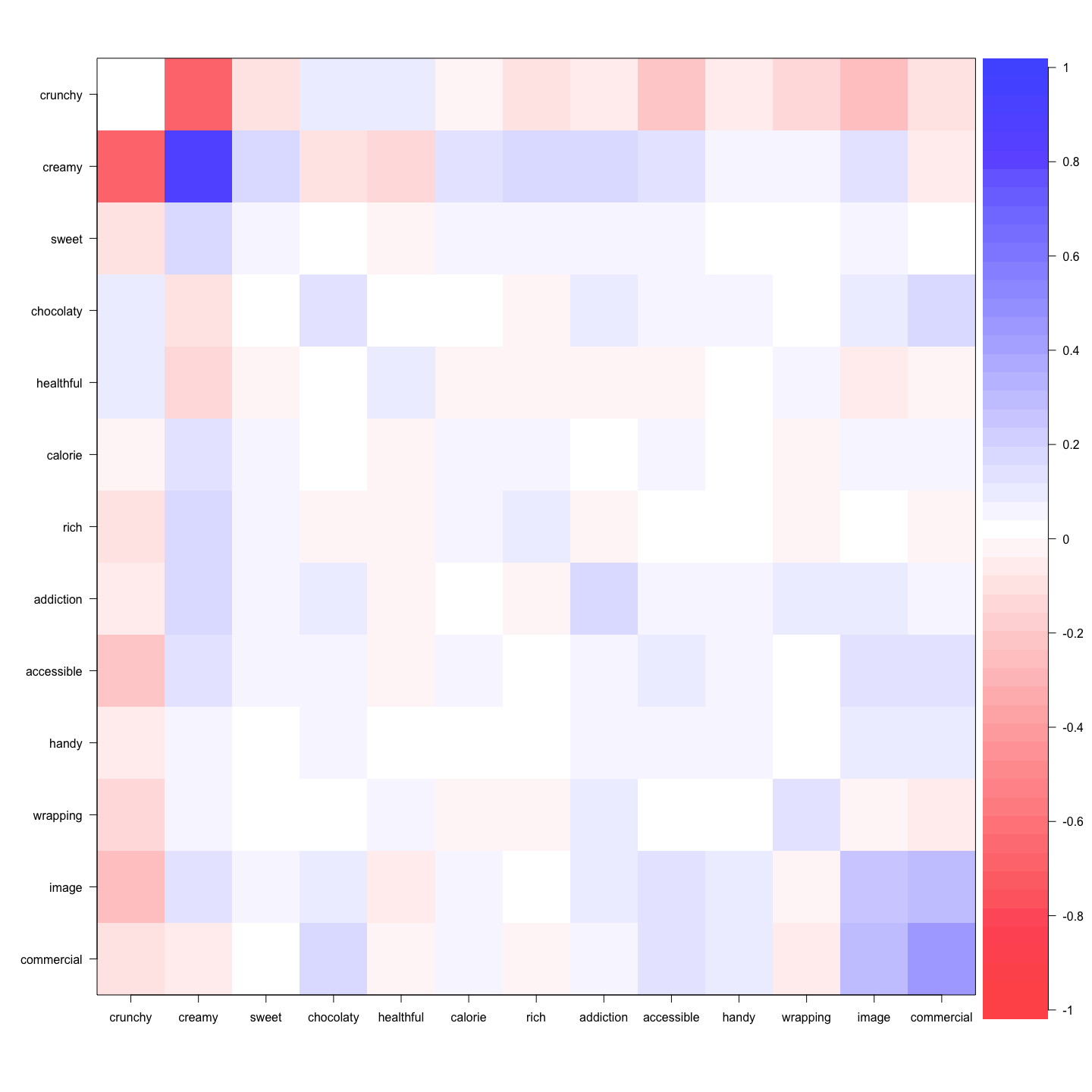
## Warning in fa.parallel(tmp, n.obs = 50, fa = "fa", main = "Scree Plots for  
## FA with Parallel Analysis"): You specified the number of subjects, implying  
## a correlation matrix, but do not have a correlation matrix, correlations  
## found



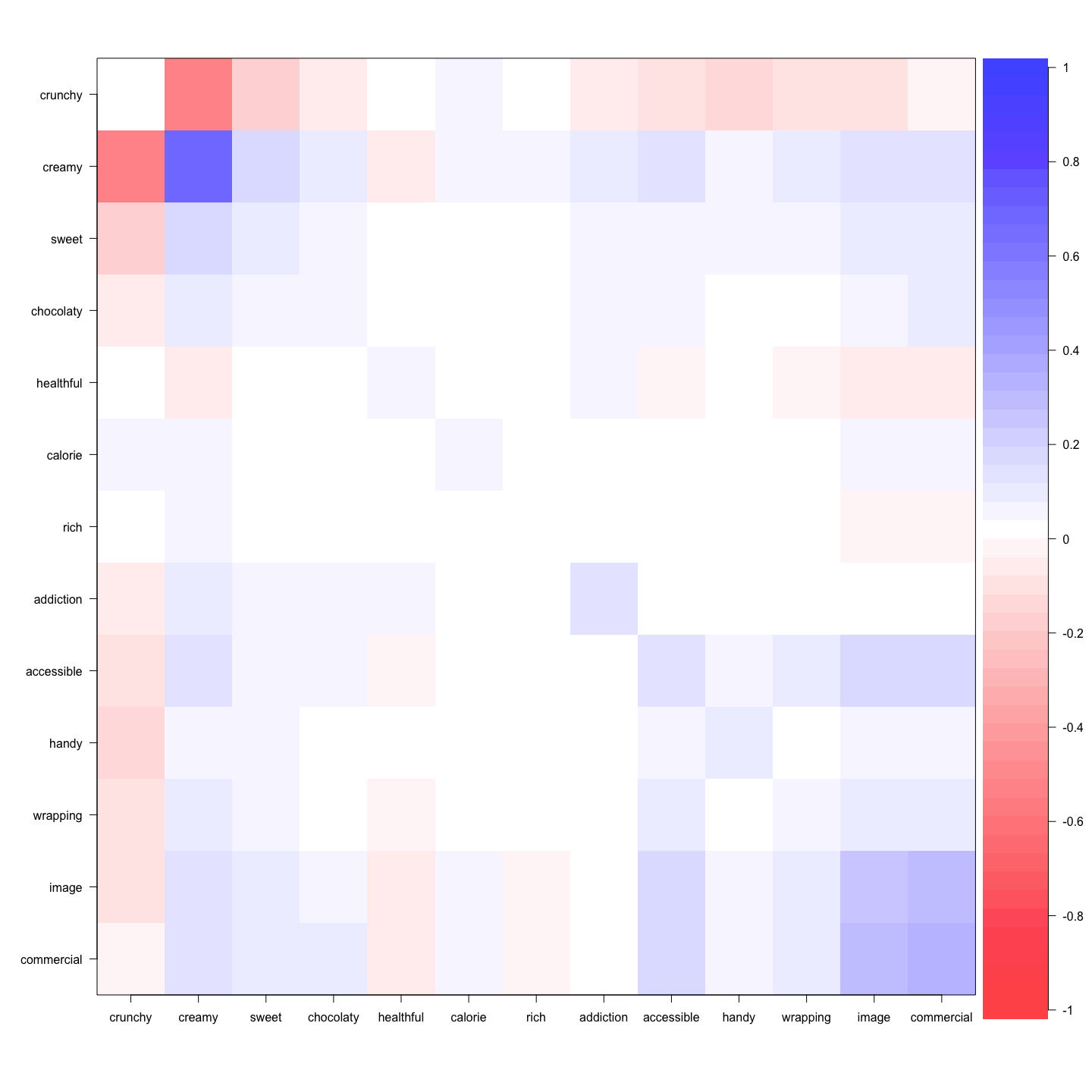
## Parallel analysis suggests that the number of factors = 1 and the number of components = NA

### End of the Test DA2 Parallel analysis of Horn

# correlation matrix by gender  
library(psych)  
tmp <- bars.gender[,3:15]  
  
par(mfrow = c(2,1))  
options(digits=2)  
female <- cov(bars.gender[1:10,3:15])  
cor.plot(female)



male <- cov(bars.gender[11:20,3:15])  
cor.plot(male)



## 4. EFA Rotation

library("psych")  
  
# ML with Kaiser normalization  
factanal(attribute[,3:15], factors=4, rotation = "varimax")

##   
## Call:  
## factanal(x = attribute[, 3:15], factors = 4, rotation = "varimax")  
##   
## Uniquenesses:  
## crunchy creamy sweet chocolaty healthful calorie   
## 0.95 0.85 0.47 0.61 0.60 0.60   
## rich addiction accessible handy wrapping image   
## 0.88 0.72 0.50 0.46 0.77 0.27   
## commercial   
## 0.48   
##   
## Loadings:  
## Factor1 Factor2 Factor3 Factor4  
## crunchy -0.100 0.182   
## creamy 0.334 0.102 0.161   
## sweet 0.620 0.270 -0.245   
## chocolaty 0.595 0.114 0.150   
## healthful 0.207 0.594   
## calorie 0.325 0.112 0.123 -0.512   
## rich 0.239 0.245   
## addiction 0.426 0.226 0.202   
## accessible 0.144 0.268 0.627 -0.113   
## handy 0.269 0.115 0.676   
## wrapping 0.142 0.135 0.402 0.182   
## image 0.150 0.825 0.161   
## commercial 0.107 0.701 0.143   
##   
## Factor1 Factor2 Factor3 Factor4  
## SS loadings 1.3 1.34 1.34 0.828  
## Proportion Var 0.1 0.10 0.10 0.064  
## Cumulative Var 0.1 0.21 0.31 0.373  
##   
## Test of the hypothesis that 4 factors are sufficient.  
## The chi square statistic is 157 on 32 degrees of freedom.  
## The p-value is 2.3e-18

# oblimin rotation without Kaiser normalization  
fa1 <- fa(attribute[,3:15], nfactors=3)

## Loading required namespace: GPArotation

fa1

## Factor Analysis using method = minres  
## Call: fa(r = attribute[, 3:15], nfactors = 3)  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## MR1 MR3 MR2 h2 u2 com  
## crunchy -0.07 0.00 0.13 0.021 0.98 1.5  
## creamy 0.41 -0.01 0.00 0.168 0.83 1.0  
## sweet 0.68 -0.02 -0.16 0.486 0.51 1.1  
## chocolaty 0.38 0.04 0.11 0.166 0.83 1.2  
## healthful 0.05 0.03 0.67 0.465 0.54 1.0  
## calorie 0.40 0.07 -0.44 0.380 0.62 2.0  
## rich 0.40 -0.11 0.03 0.139 0.86 1.2  
## addiction 0.47 -0.03 0.24 0.265 0.73 1.5  
## accessible 0.44 0.23 0.06 0.339 0.66 1.6  
## handy 0.58 0.08 0.13 0.395 0.60 1.1  
## wrapping 0.33 0.07 0.28 0.213 0.79 2.1  
## image 0.06 0.74 0.04 0.600 0.40 1.0  
## commercial -0.05 0.82 -0.03 0.637 0.36 1.0  
##   
## MR1 MR3 MR2  
## SS loadings 2.03 1.38 0.87  
## Proportion Var 0.16 0.11 0.07  
## Cumulative Var 0.16 0.26 0.33  
## Proportion Explained 0.48 0.32 0.20  
## Cumulative Proportion 0.48 0.80 1.00  
##   
## With factor correlations of   
## MR1 MR3 MR2  
## MR1 1.00 0.40 -0.01  
## MR3 0.40 1.00 0.11  
## MR2 -0.01 0.11 1.00  
##   
## Mean item complexity = 1.3  
## Test of the hypothesis that 3 factors are sufficient.  
##   
## The degrees of freedom for the null model are 78 and the objective function was 2.5 with Chi Square of 1235  
## The degrees of freedom for the model are 42 and the objective function was 0.48   
##   
## The root mean square of the residuals (RMSR) is 0.06   
## The df corrected root mean square of the residuals is 0.08   
##   
## The harmonic number of observations is 500 with the empirical chi square 261 with prob < 1.9e-33   
## The total number of observations was 500 with Likelihood Chi Square = 237 with prob < 4.9e-29   
##   
## Tucker Lewis Index of factoring reliability = 0.69  
## RMSEA index = 0.097 and the 90 % confidence intervals are 0.085 0.11  
## BIC = -24  
## Fit based upon off diagonal values = 0.92  
## Measures of factor score adequacy   
## MR1 MR3 MR2  
## Correlation of (regression) scores with factors 0.87 0.88 0.77  
## Multiple R square of scores with factors 0.76 0.78 0.60  
## Minimum correlation of possible factor scores 0.53 0.56 0.19

# apply Kaiser normalization   
fa2 <- fa(attribute[,3:15], nfactors=3, rotate="none")  
fa2 <- kaiser(fa2)  
fa2

##   
## Call: NULL  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## MR1 MR2 MR3 h2 u2  
## crunchy -0.03 0.14 0.00 0.021 0.98  
## creamy 0.39 -0.09 -0.01 0.168 0.83  
## sweet 0.60 -0.30 -0.01 0.486 0.51  
## chocolaty 0.39 0.02 -0.05 0.166 0.83  
## healthful 0.25 0.65 -0.04 0.465 0.54  
## calorie 0.24 -0.52 -0.09 0.380 0.62  
## rich 0.39 -0.06 0.09 0.139 0.86  
## addiction 0.51 0.14 0.01 0.265 0.73  
## accessible 0.43 -0.04 -0.25 0.339 0.66  
## handy 0.58 0.00 -0.11 0.395 0.60  
## wrapping 0.39 0.20 -0.09 0.213 0.79  
## image 0.05 0.01 -0.75 0.600 0.40  
## commercial -0.07 -0.05 -0.82 0.637 0.36  
##   
## MR1 MR2 MR3  
## SS loadings 1.95 0.89 1.43  
## Proportion Var 0.15 0.07 0.11  
## Cumulative Var 0.15 0.22 0.33  
## Proportion Explained 0.46 0.21 0.33  
## Cumulative Proportion 0.46 0.67 1.00  
## MR1 MR2 MR3  
## MR1 1.00 -0.10 -0.39  
## MR2 -0.10 1.00 -0.02  
## MR3 -0.39 -0.02 1.00

# compare loading sets (vector cosines)  
factor.congruence(fa1, fa2)

## MR1 MR2 MR3  
## MR1 0.98 -0.25 -0.14  
## MR3 0.08 -0.03 -1.00  
## MR2 0.28 0.95 -0.04

## 5. EFA Scores

library("psych")  
  
# ML with Kaiser normalization  
fa1 <-factanal(attribute[,3:15], factors=3, scores="regression")  
head(fa1$scores)

## Factor1 Factor2 Factor3  
## [1,] -1.342 -0.57 -1.15  
## [2,] -0.017 -0.78 -0.19  
## [3,] 0.134 -0.70 -0.64  
## [4,] -0.417 -1.42 0.85  
## [5,] 0.584 -0.36 -1.22  
## [6,] -0.733 0.35 -0.94

# oblimin rotation without Kaiser normalization  
fa2 <- fa(attribute[,3:15], nfactors=3)  
head(fa2$scores)

## MR1 MR3 MR2  
## [1,] -1.70 -0.93 0.81  
## [2,] -0.15 -0.74 0.09  
## [3,] -0.20 -0.68 0.60  
## [4,] -0.33 -1.42 -0.98  
## [5,] 0.14 -0.27 1.38  
## [6,] -0.97 0.16 0.74

# compare scores  
cor(fa1$scores, fa2$scores)

## MR1 MR3 MR2  
## Factor1 0.95 0.346 0.25  
## Factor2 0.30 0.973 0.12  
## Factor3 0.32 0.034 -0.94

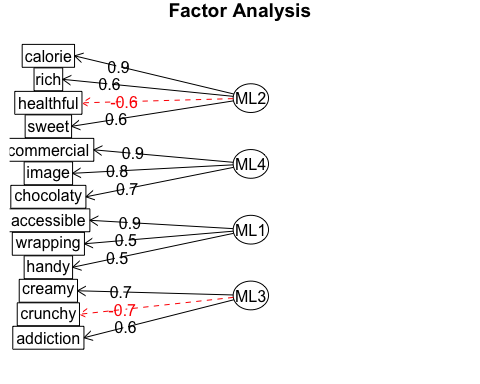
# caci 的  
a.fa.gender<-fa(bars.gender[,3:15],fm="ml", max.iter=1000,SMC=TRUE,scores='Anderson',nfactors=4, rotate ="varimax")  
a.fa.gender

## Factor Analysis using method = ml  
## Call: fa(r = bars.gender[, 3:15], nfactors = 4, rotate = "varimax",   
## scores = "Anderson", SMC = TRUE, max.iter = 1000, fm = "ml")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## ML2 ML4 ML1 ML3 h2 u2 com  
## crunchy -0.18 -0.01 -0.29 -0.66 0.55 0.449 1.5  
## creamy 0.67 0.01 0.05 0.73 1.00 0.005 2.0  
## sweet 0.59 0.03 0.40 0.40 0.67 0.333 2.6  
## chocolaty -0.08 0.69 -0.07 0.02 0.49 0.510 1.1  
## healthful -0.64 -0.19 -0.30 -0.12 0.55 0.448 1.7  
## calorie 0.92 0.11 0.23 -0.29 1.00 0.005 1.4  
## rich 0.65 -0.26 -0.23 0.08 0.55 0.453 1.6  
## addiction -0.07 0.16 0.01 0.58 0.36 0.636 1.2  
## accessible 0.28 0.36 0.86 0.21 1.00 0.005 1.7  
## handy 0.15 0.29 0.48 0.07 0.34 0.659 1.9  
## wrapping -0.18 -0.17 0.50 0.47 0.53 0.466 2.5  
## image 0.16 0.80 0.42 0.20 0.89 0.110 1.8  
## commercial 0.03 0.93 0.37 0.00 1.00 0.005 1.3  
##   
## ML2 ML4 ML1 ML3  
## SS loadings 2.68 2.37 1.99 1.88  
## Proportion Var 0.21 0.18 0.15 0.14  
## Cumulative Var 0.21 0.39 0.54 0.69  
## Proportion Explained 0.30 0.27 0.22 0.21  
## Cumulative Proportion 0.30 0.57 0.79 1.00  
##   
## Mean item complexity = 1.7  
## Test of the hypothesis that 4 factors are sufficient.  
##   
## The degrees of freedom for the null model are 78 and the objective function was 12 with Chi Square of 165  
## The degrees of freedom for the model are 32 and the objective function was 2.9   
##   
## The root mean square of the residuals (RMSR) is 0.08   
## The df corrected root mean square of the residuals is 0.12   
##   
## The harmonic number of observations is 20 with the empirical chi square 19 with prob < 0.97   
## The total number of observations was 20 with Likelihood Chi Square = 32 with prob < 0.44   
##   
## Tucker Lewis Index of factoring reliability = 0.98  
## RMSEA index = 0.2 and the 90 % confidence intervals are 0 NA  
## BIC = -63  
## Fit based upon off diagonal values = 0.96  
## Measures of factor score adequacy   
## ML2 ML4 ML1 ML3  
## Correlation of (regression) scores with factors 1.00 1.00 0.99 1.00  
## Multiple R square of scores with factors 1.00 0.99 0.99 0.99  
## Minimum correlation of possible factor scores 0.99 0.98 0.98 0.98

EFA <-print(fa(bars.gender[,3:15], fm="ml", nfactors =4, scores='Anderson',rotate ="varimax")$ loadings ,cut =0.3)

##   
## Loadings:  
## ML2 ML4 ML1 ML3   
## crunchy -0.659  
## creamy 0.673 0.735  
## sweet 0.594 0.395 0.396  
## chocolaty 0.691   
## healthful -0.639 -0.305   
## calorie 0.919   
## rich 0.647   
## addiction 0.578  
## accessible 0.360 0.862   
## handy 0.479   
## wrapping 0.500 0.469  
## image 0.803 0.425   
## commercial 0.928 0.365   
##   
## ML2 ML4 ML1 ML3  
## SS loadings 2.68 2.37 1.99 1.88  
## Proportion Var 0.21 0.18 0.15 0.14  
## Cumulative Var 0.21 0.39 0.54 0.69

fa.diagram(EFA, simple = TRUE)



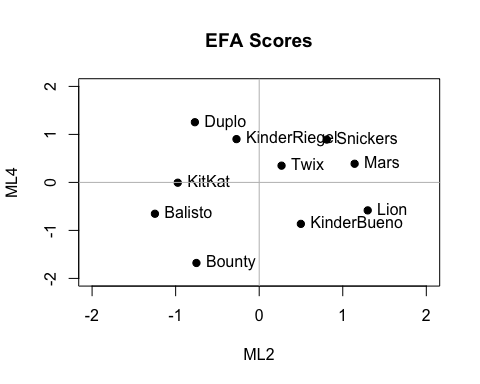
# • 不要性别  
EFAscore<- aggregate(a.fa.gender$scores[,1:2], by=list(bars.gender$Group.1) ,mean)  
colnames(EFAscore)[1] <- c("Product")  
EFAscore

## Product ML2 ML4  
## 1 Balisto -1.25 -0.6532  
## 2 Bounty -0.75 -1.6787  
## 3 Duplo -0.77 1.2540  
## 4 KinderBueno 0.50 -0.8642  
## 5 KinderRiegel -0.27 0.9020  
## 6 KitKat -0.97 -0.0064  
## 7 Lion 1.30 -0.5829  
## 8 Mars 1.14 0.3883  
## 9 Snickers 0.81 0.8924  
## 10 Twix 0.27 0.3487

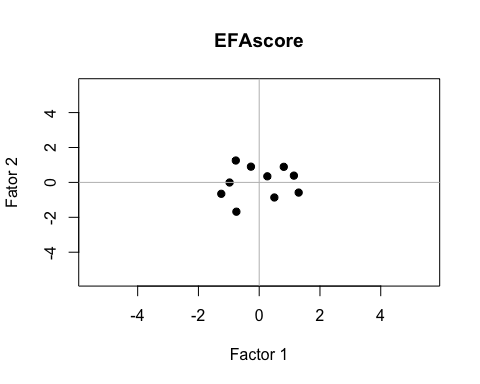
# ••••最后efa图/ test----  
EFAscore<- aggregate(a.fa.gender$scores[,1:2], by=list(bars.gender$Group.1) ,mean);EFAscore

## Group.1 ML2 ML4  
## 1 Balisto -1.25 -0.6532  
## 2 Bounty -0.75 -1.6787  
## 3 Duplo -0.77 1.2540  
## 4 KinderBueno 0.50 -0.8642  
## 5 KinderRiegel -0.27 0.9020  
## 6 KitKat -0.97 -0.0064  
## 7 Lion 1.30 -0.5829  
## 8 Mars 1.14 0.3883  
## 9 Snickers 0.81 0.8924  
## 10 Twix 0.27 0.3487

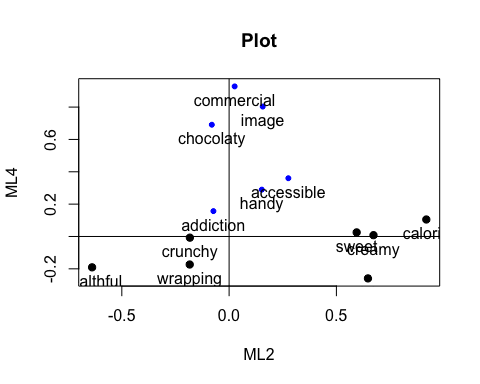
plot(EFAscore$ML2,EFAscore$ML4, ylim = c(-2, 2), xlim = c(-2, 2),  
 xlab = "ML2", ylab = "ML4",pch = 19,  
 main = "EFA Scores")  
text(EFAscore$ML2,EFAscore$ML4, labels = EFAscore$Group.1, cex = 1, pos = 4) # label the value  
## type="n") tells R not to plot symbols.   
## Instead, we add the brand labels to the plot with text(x, labels).  
abline(h = 0, v = 0, col = "grey")



# Test 完  
  
#• 前两个factor  
plot(EFAscore$ML2,EFAscore$ML4, xlab = "Factor 1", ylab = "Fator 2", main = "EFAscore",  
 pch = 19, ylim = c(-5.5, 5.5), xlim = c(-5.5, 5.5))  
text(EFAscore$ML2,EFAscore$ML4, labels = EFAscore$Product, cex = 1, pos = 4)  
abline(h = 0, v = 0, col = "grey")



# • attribute 前两个的plot  
factor.plot(a.fa.gender$loadings[,c("ML2", "ML4")], labels=rownames(a.fa.gender$loadings))



write.csv(EFAscore, file = "EFAscore")  
## Test图/ EFA\_ ml orthogonal rotation----