

# Fokkomponenselemzés és exploratoros faktorelemzés

Zoltan Kekecs

April 25, 2021

## Contents

<b>1</b>	<b>Absztrakt</b>	<b>2</b>
<b>2</b>	<b>Adat kezelés és leíró statisztikák</b>	<b>2</b>
2.1	Csomagok betöltése . . . . .	2
2.2	Saját funkciók betöltése . . . . .	2
2.3	A Humor Styles Questionnaire betöltése . . . . .	4
2.4	Az adatsor megtekintése . . . . .	5
2.5	A dimenzionalitás atka . . . . .	7
2.6	A korrelációs struktúra vizualizációja . . . . .	7
<b>3</b>	<b>Fokkomponenselemzés</b>	<b>11</b>
3.1	A fokkomponenselemzés modell megépítése . . . . .	11
3.2	Hogyan működik a PCA? . . . . .	11
3.3	PCA használata R-ben . . . . .	12
3.4	Hány fokkomponenst nyerünk ki? . . . . .	18
3.5	Fokkomponenselemzés eredményeinek értelmezése . . . . .	24
<b>4</b>	<b>Bevezetés a feltárási faktorelemzésbe (Exploratory Factor Analysis - EFA)</b>	<b>40</b>
4.1	Adatok faktorálhatósága . . . . .	41
4.2	Faktorextrakció . . . . .	42
4.3	Ideális faktorszám kiválasztása . . . . .	44
4.4	Faktorforgatás . . . . .	48
4.5	Faktorok interpretációja . . . . .	48

# 1 Absztrakt

Ebben a gyakorlatban a “dimenzionalitás átkaval” birkozunk meg. Ezt a változók számanak csökkentésével oldjuk meg a főkomponenselemzés (PCA) és az exploratoros faktorelemzés (EFA) segítségével.

## 2 Adat kezeles es leiro statisztikak

### 2.1 Csomagok betoltese

Ennek a gyakorlatnak a során az alábbi csomagokat fogjuk használni:

```
library(GGally) # for ggcorr
library(corr) # network_plot
library(ggcorrplot) # for ggcorrplot
library(FactoMineR) # multiple PCA functions
library(factoextra) # visualisation functions for PCA (e.g. fviz_pca_var)
library(paran) # for paran

library(psych) # for the mixedCor, cortest.bartlett, KMO, fa functions
library(car) # for vif
library(GPArotation) # for the psych fa function to have the required rotation functionalities
library(MVN) # for mvn function
library(ICS) # for multivariate skew and kurtosis test
library(tidyverse) # for tidy code
```

### 2.2 Saját funkciók betoltese

Az alábbi fviz\_loadnings\_with\_cor() a PCA és a faktorelemzés eredményeinke vizualizálására szolgál.

```
fviz_loadnings_with_cor <- function(mod, axes = 1, loadings_above = 0.4){
  require(factoextra)
  require(dplyr)
  require(ggplot2)

  if(!is.na(as.character(mod$call$call)[1])){
    if(as.character(mod$call$call)[1] == "PCA"){
      contrib_and_cov = as.data.frame(rbind(mod[["var"]][["contrib"]], mod[["var"]][["cor"]]))

      vars = rownames(mod[["var"]][["contrib"]])
      attribute_type = rep(c("contribution", "correlation"), each = length(vars))
      contrib_and_cov = cbind(contrib_and_cov, attribute_type)
      contrib_and_cov

      plot_data = cbind(as.data.frame(cbind(contrib_and_cov[contrib_and_cov[, "attribute_type"] == "contribution", ],
      names(plot_data) = c("contribution", "correlation", "vars")

      plot_data = plot_data %>%
        mutate(correlation = round(correlation, 2))

      plot = plot_data %>%
        ggplot() +
        aes(x = reorder(vars, contribution), y = contribution, gradient = correlation, label = correlation) +
        geom_col(aes(fill = correlation)) +
```

```

geom_hline(yintercept = mean(plot_data$contribution), col = "red", lty = "dashed") + scale_fill_gradient2()
xlab("variable") +
coord_flip() +
geom_label(color = "black", fontface = "bold", position = position_dodge(0.5))

}
} else if(!is.na(as.character(mod$Call)[1])){

  if(as.character(mod$Call)[1] == "fa"){
    loadings_table = mod$loadings %>%
      matrix(ncol = ncol(mod$loadings)) %>%
      as_tibble() %>%
      mutate(variable = mod$loadings %>% rownames()) %>%
      gather(factor, loading, -variable) %>%
      mutate(sign = if_else(loading >= 0, "positive", "negative"))

    if(!is.null(loadings_above)){
      loadings_table[abs(loadings_table[, "loading"]) < loadings_above, "loading"] = NA
      loadings_table = loadings_table[!is.na(loadings_table[, "loading"]), ]
    }

    if(!is.null(axes)){
      loadings_table = loadings_table %>%
        filter(factor == paste0("V", axes))
    }

    plot = loadings_table %>%
      ggplot() +
      aes(y = loading %>% abs(), x = reorder(variable, abs(loading)), fill = loading, label = round(loading, 1)) +
      geom_col(position = "dodge") +
      scale_fill_gradient2() +
      coord_flip() +
      geom_label(color = "black", fill = "white", fontface = "bold", position = position_dodge(0.5)) +
      facet_wrap(~factor) +
      labs(y = "Loading strength", x = "Variable")
  }
}

return(plot)
}

```

## 2.3 A Humor Styles Questionnaire betöltése

Alább betöltjük a “Humor Styles Questionnaire” adatbázist, ami a Martin et. al. (2003). kutatásából származik, akik a HSQ kerdoivet vettek fel 1071 személlyel.

Az adatbázis elso 32 oszlopa Q1-Q32 a kerdoiv egyes teteleire adott valaszokat tartalmazza minden személytol. A valaszadoknak mind a 32 allitasrol ertekelnie kellett, hogy mennyire igaz az ra nezve. A valaszok ordinalis skalan mozognak, 1-tol 5-ig: 1=“soha vagy nagyon ritkan igaz”“, 5=“nagyon gyakran vagy soha nem igaz“. Ilyen allitasok szerepelnek a kerdoivben mint:”Q1: Altalaban nem nevetek vagy viccelodok masokkal.” (Q1: “I usually don’t laugh or joke around much with other people.”)

A kerdoiv itemei a kovetkezoek:

- Q1. I usually don’t laugh or joke around much with other people.
- Q2. If I am feeling depressed, I can usually cheer myself up with humor.
- Q3. If someone makes a mistake, I will often tease them about it.
- Q4. I let people laugh at me or make fun at my expense more than I should.
- Q5. I don’t have to work very hard at making other people laugh—I seem to be a naturally humorous person.
- Q6. Even when I’m by myself, I’m often amused by the absurdities of life.
- Q7. People are never offended or hurt by my sense of humor.
- Q8. I will often get carried away in putting myself down if it makes my family or friends laugh.
- Q9. I rarely make other people laugh by telling funny stories about myself.
- Q10. If I am feeling upset or unhappy I usually try to think of something funny about the situation to make myself feel better.
- Q11. When telling jokes or saying funny things, I am usually not very concerned about how other people are taking it.
- Q12. I often try to make people like or accept me more by saying something funny about my own weaknesses, blunders, or faults.
- Q13. I laugh and joke a lot with my closest friends.
- Q14. My humorous outlook on life keeps me from getting overly upset or depressed about things.
- Q15. I do not like it when people use humor as a way of criticizing or putting someone down.
- Q16. I don’t often say funny things to put myself down.
- Q17. I usually don’t like to tell jokes or amuse people.
- Q18. If I’m by myself and I’m feeling unhappy, I make an effort to think of something funny to cheer myself up.
- Q19. Sometimes I think of something that is so funny that I can’t stop myself from saying it, even if it is not appropriate for the situation.
- Q20. I often go overboard in putting myself down when I am making jokes or trying to be funny.
- Q21. I enjoy making people laugh.
- Q22. If I am feeling sad or upset, I usually lose my sense of humor.
- Q23. I never participate in laughing at others even if all my friends are doing it.
- Q24. When I am with friends or family, I often seem to be the one that other people make fun of or joke about.
- Q25. I don’t often joke around with my friends.
- Q26. It is my experience that thinking about some amusing aspect of a situation is often a very effective way of coping with problems.
- Q27. If I don’t like someone, I often use humor or teasing to put them down.
- Q28. If I am having problems or feeling unhappy, I often cover it up by joking around, so that even my closest friends don’t know how I really feel.
- Q29. I usually can’t think of witty things to say when I’m with other people.
- Q30. I don’t need to be with other people to feel amused – I can usually find things to laugh about even when I’m by myself.
- Q31. Even if something is really funny to me, I will not laugh or joke about it if someone will be offended.
- Q32. Letting others laugh at me is my way of keeping my friends and family in good spirits.

A HSQ-n kívül az adatbázis tartalmaz más változókat is:

- gender - Faktor változó: 1=male, 2=female, 3=other)
- accuracy - Mennyire érezték pontosnak a válaszaikat a HSQ keredeskre. 0-100-as skálán értékelve. (A vizsgálati személyeknek azt mondták hogy az írjon 0-t aki nem szeretne hogy a kísérletben felhasználják az adatát).
- life\_stress - Ez egy új változó ami em volt benne az eredeti kutatásban (szimulált adat). Arra utal, hogy általánosságban mennyire stresszesnek érzi az életet a válaszadó 0-9-es skálán, ahol a 0 azt jelenti hogy semennyire, 9 azt jelenti, hogy extrém stresszes.

Az adatbázis tartalmaz hiányzó adatokat. Az egyszerűség kedvéért most zárjuk ki azokat a válaszadókat akiknek akár egy adatpont is hiányzik a sorából.

```
hsq <- read_csv("https://raw.githubusercontent.com/kekecsz/PSZB17-210-Data-analysis-seminar/master/seminar_data.csv")

##
## -- Column specification -----
## cols(
##   .default = col_double()
## )
## i Use `spec()` for the full column specifications.

hsq <- hsq %>%
  drop_na()

hsq %>%
  describe()
```

## 2.4 Az adatsor megtekintése

Vizsgáljuk meg az adatok strukturáját és az alapvető leíró statisztikákat.

```
str(hsq)

summary(hsq)
```

Mondjuk hogy szeretnénk meghatározni melyek az emberek humor stílusának legfőbb jellegzetességei, melyek meghatározzák a személy stresszel való viszonyát. Másnéven hogy a humor stílus mely aspektusaival segíthetnek bejósolni a személy általános stressz-szintjét. Egy módja hogy ezt meghatározzuk, hogy építünk egy regressziós modellt, amiben a life\_stress változót jósoljuk be a Humor Style Questionnaire itemeivel (Q1-Q32).

Amikor lefuttatjuk ezt a modellt, a regressziós együtthatók szignifikanciája alapján úgy tűnik hogy a modell egészen szignifikánsan jobb mint a null modell, és több olyan item is van aminek van szignifikáns hozzáadott értéke a modellhez, vagyis érdemes a humor stílust figyelembe venni a stressz meghatározásánál.

De ha arra vagyunk kíváncsiak hogy a humor stílus mely aspektusai fontosak, ezt nehéz ebből a regressziós elemzésből megállapítani. Egyrészt nem bizhatunk meg teljesen a p-értékekben, mert 32 statisztikai tesztet hajtottunk végre, a 32 item hozzáadott megmagyarázó értékenek tesztelekor, ami nagyban növeli az elsőfajú hiba (fals pozitív) valószínűségét. Vagyis nagy a valószínűsége hogy a szignifikánsként megjelölt prediktorok közül néhány nincs valódi összefüggésben a stresszel a populációban.

Másrészt figyelembe kell venni hogy a prediktorok korrelálnak egymással, vagyis bizonyos prediktorok a stressz varianciájának hasonló részét magyarázzák, és ebben a modellben lehet hogy az egyes prediktorok más korreláló prediktorok hatását elmaszkolják.

```
mod_allitems = lm(life_stress ~ Q1 + Q2 + Q3 + Q4 + Q5 + Q6 + Q7 + Q8 + Q9 + Q10 +
  Q11 + Q12 + Q13 + Q14 + Q15 + Q16 + Q17 + Q18 + Q19 + Q20 +
```

```

      Q21 + Q22 + Q23 + Q24 + Q25 + Q26 + Q27 + Q28 + Q29 + Q30 +
      Q31 + Q32,
      data = hsq)

```

```
summary(mod_allitems)
```

```

##
## Call:
## lm(formula = life_stress ~ Q1 + Q2 + Q3 + Q4 + Q5 + Q6 + Q7 +
##      Q8 + Q9 + Q10 + Q11 + Q12 + Q13 + Q14 + Q15 + Q16 + Q17 +
##      Q18 + Q19 + Q20 + Q21 + Q22 + Q23 + Q24 + Q25 + Q26 + Q27 +
##      Q28 + Q29 + Q30 + Q31 + Q32, data = hsq)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.3528 -0.8639  0.0307   0.8807  3.8981
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.084097   0.623579   0.135 0.892750
## Q1           -0.097889   0.050921  -1.922 0.054856 .
## Q2            0.007704   0.047341   0.163 0.870764
## Q3            0.030141   0.043687   0.690 0.490402
## Q4            0.170860   0.045619   3.745 0.000191 ***
## Q5           -0.066691   0.053180  -1.254 0.210121
## Q6            0.026375   0.055614   0.474 0.635433
## Q7           -0.023993   0.046146  -0.520 0.603235
## Q8            0.105826   0.050916   2.078 0.037935 *
## Q9           -0.052866   0.039836  -1.327 0.184797
## Q10           0.009067   0.050208   0.181 0.856732
## Q11          -0.015873   0.039779  -0.399 0.689967
## Q12           0.202653   0.044006   4.605 4.68e-06 ***
## Q13           0.026218   0.067533   0.388 0.697937
## Q14           0.052088   0.045885   1.135 0.256577
## Q15           0.015961   0.040418   0.395 0.693010
## Q16           0.163045   0.043615   3.738 0.000196 ***
## Q17           0.027425   0.054744   0.501 0.616506
## Q18           0.010161   0.050137   0.203 0.839442
## Q19           0.034051   0.041819   0.814 0.415709
## Q20           0.208081   0.055749   3.732 0.000201 ***
## Q21          -0.103708   0.066913  -1.550 0.121496
## Q22           0.062369   0.039423   1.582 0.113967
## Q23           0.044103   0.041631   1.059 0.289695
## Q24           0.222733   0.043525   5.117 3.74e-07 ***
## Q25           0.031822   0.070080   0.454 0.649871
## Q26          -0.072567   0.049911  -1.454 0.146287
## Q27          -0.020883   0.038454  -0.543 0.587211
## Q28           0.201723   0.036348   5.550 3.70e-08 ***
## Q29          -0.031136   0.044079  -0.706 0.480130
## Q30           0.006414   0.046809   0.137 0.891043
## Q31          -0.011109   0.042425  -0.262 0.793497
## Q32           0.148716   0.045016   3.304 0.000990 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
##
## Residual standard error: 1.302 on 960 degrees of freedom
## Multiple R-squared:  0.3438, Adjusted R-squared:  0.3219
## F-statistic: 15.72 on 32 and 960 DF,  p-value: < 2.2e-16
```

A prediktorok közötti korrelációról megbizonyosodhatunk ha lekerdezzük a prediktorok korrelációs matrixát a `cor()` függvénnyel. A `vif()` függvénnyel megnevezhetjük hogy ez az interkorreláció problémás multikollinearitáshoz vezet-e a modellünkben (ebben az esetben a `vif` 3 alatt marad minden esetben, vagyis nincs számottevő multikollinearitás, de más hasonló esetben amikor kerdoivek minden itemet a modellbe építjük ez is könnyen előfordulhat.)

```
hsq_items_only = hsq %>%
  select(Q1:Q32)

cor = hsq_items_only %>%
  cor()

cor

vif(mod_allitems)
```

## 2.5 A dimenzionalitás atka

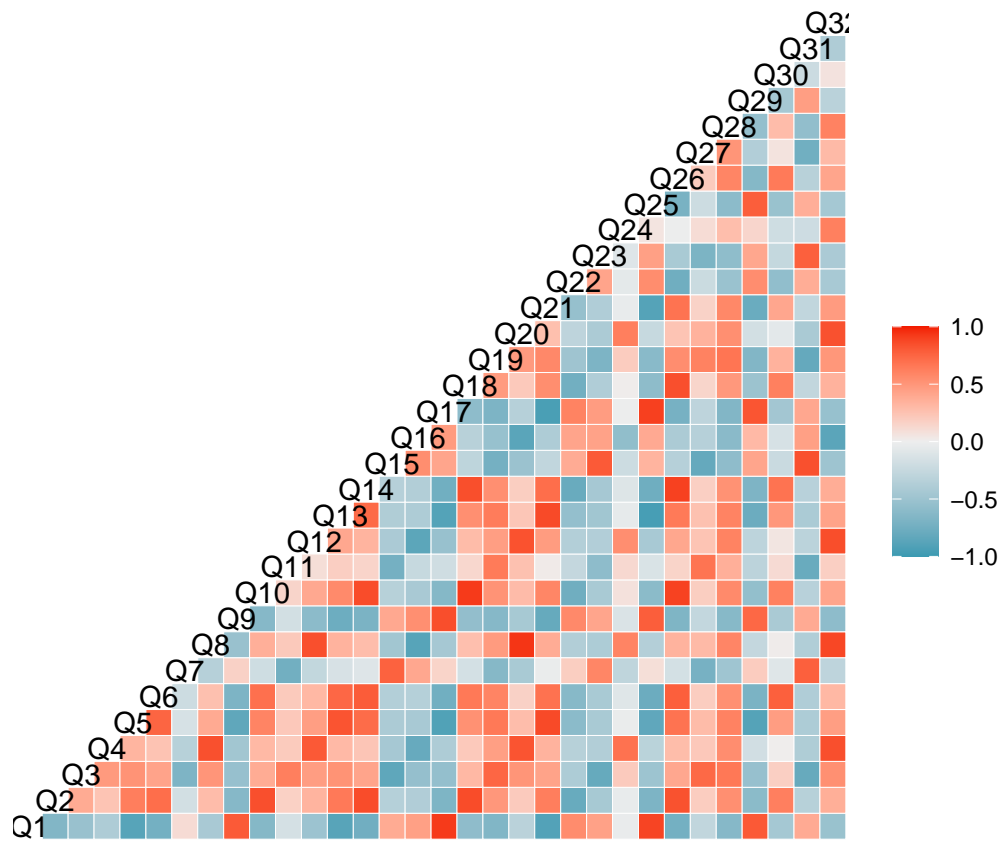
Vagyis 32 egymással korreláló prediktor bevonása a modellbe nem ideális annak a megértésére, hogy a humor stílus mely aspektusai fontosak a stressz meghatározásában. Sőt, ez kifejezetten problémás minél kisebb a minta-elemszám a prediktorok számához képest. Ezt a statisztikusok gyakran a **“dimenzionalitás atka”**-ként emlegetik. A “dimenzionalitás” arra utal, hogy minél több változó van a modellünkben, az adatokat annál több-dimenziós térben lehet modellezni. Például az egyszerű regresszióal amikor csak egy kimeneti változónk és egy prediktorunk van, az adatok egy kétdimenziós térben ábrázolhatók: a kimeneti változó az  $y$  tengelyen (dimenzión), a prediktor pedig az  $x$  tengelyen (dimenzión) ábrázolva. A regressziós egyenes ebben az esetben valóban egy egyenes. Amikor már két prediktorunk van, az adatok egy három dimenziós térben ábrázolhatók, és a regressziós modell egy regressziós sík, nem csak egy egyenes. Amikor 32 prediktorunk van a fenti példában, a modellünk egy 33 dimenziós felület. Minél több dimenzióban mozoghat a modell, annál flexibilisebb, vagyis annál nagyobb a túlillesztés valószínűsége. Ezért törekednünk kell a dimenziók (prediktorok) számának minimalizálására hogy elkerülhessük a túlillesztést.

A túlillesztésnek annál nagyobb a veszélye minél inkább közelít a prediktorok száma a megfigyelések számához. Amikor egy prediktorunk van, vagyis az adat kétdimenziós térben írható le. Ha csak két megfigyelésünk lenne, akkor a regressziós egyenes tökéletes illeszkedést érhetne el, mert mindig van egy olyan egyenes ami összeköt két pontot, és ezt a regresszió megtalálja. Belátható hogy ez a regressziós egyenes amit csak két megfigyelés alapján illesztettünk nagyon sérülékeny lesz az adatokban lévő hibára, és nem fogja megbízhatóan meggranagni a populációban található valószínű összefüggést a prediktor és a kimeneti változó között. Szóval ahogy a túlillesztésről szólt orán is láthattuk, a tökéletes illeszkedés a mintához valóban nem jó, mert a populáció helyett csak a mintában lévő látszólagos mintázatokra illeszkedik a modell amit nagyon befolyásolhat a mintavételi hiba. Ugyan ez a helyzet amikor egy 2 prediktoros modellt illesztünk 3 megfigyelésre. Mivel a regressziós modell itt egy regressziós sík, egy síkot mindig lehet úgy forgatni hogy pontosan összekosson 3 pontot, így az illeszkedés tökéletes lesz, ami erős túlillesztéshez vezet. Ebből extrapolálva könnyen látható hogy ahogy a prediktorok száma közelít a megfigyelések számához, a modellünk egyre kevesbe lesz megbízható a populáció megismeréséhez, és egyre sérülékenyebb lesz a túlillesztésre.

## 2.6 A korrelációs struktúra vizualizációja

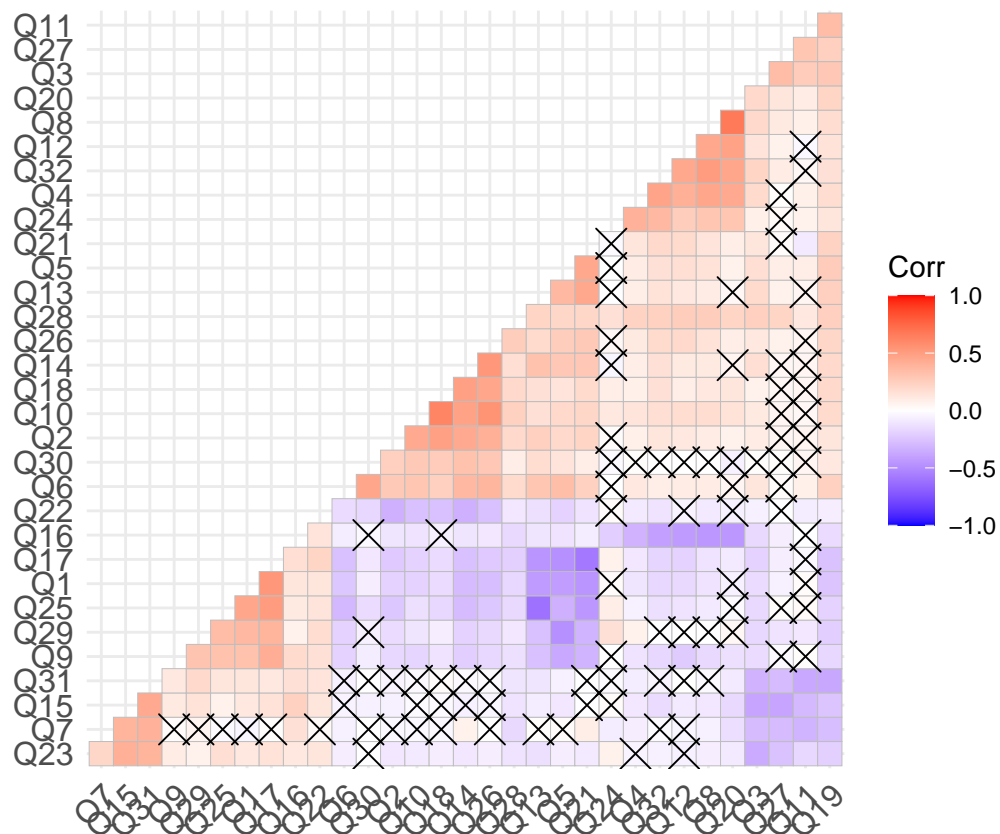
Néhány változó összefüggését könnyen átláthatóvá tehetjük vizualizáción keresztül. Azonban ha **sok változóval van dolgunk**, a vizualizáció és egyéb korábban tanult feltárási technikák kudarcot vallhatnak egyszerűen azért mert túl sok az információ amit nehéz átlátni és vizualizálni. Ez jól látszik az alábbi ábrán.

```
ggcorr(cor)
```

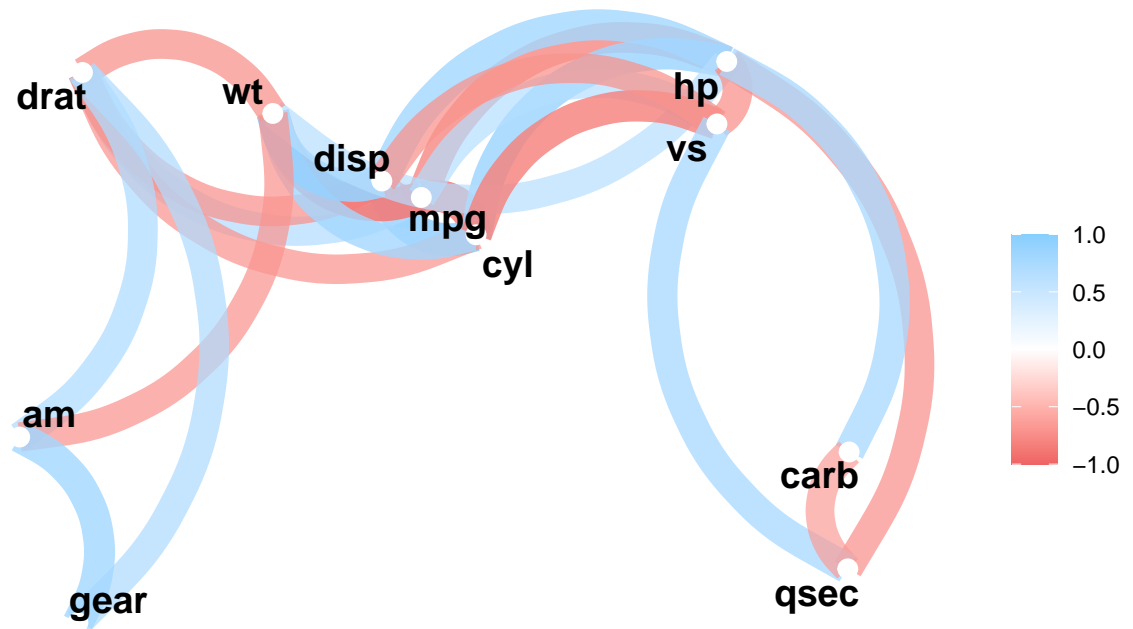


```
ggcorrplot(cor(hsq_items_only), p.mat = cor_pmat(hsq_items_only), hc.order=TRUE, type='lower')
```

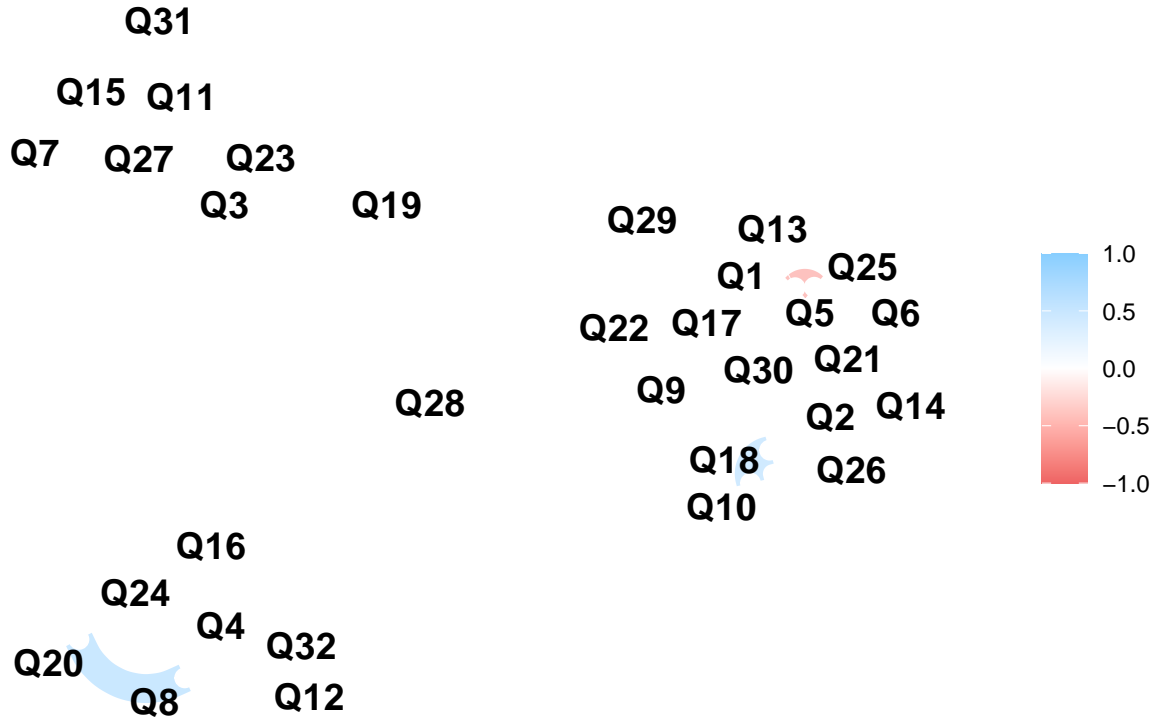




*# from a different dataset, mtcars, because there are too many variables in hsq to be displayed here*  
`cor(mtcars) %>% network_plot(min_cor=0.6)`



```
cor(hsq_items_only) %>% network_plot(min_cor=0.6)
```



### 3 Fokomponenselemzés

#### 3.1 A fokomponenselemzés modell megepítése

Amikor a dimenzionalitás atkával szembesülünk, az egyik megoldási lehetőség a változók (dimenziók) számanak csökkentése. Ha szimplán kizárnak néhány változót, mondjuk a változók modellhez hozzáadott bejoslo értéke alapján, az szinten hozzájárulna a túlilleszteshez (ezt korábbi orakon lathattuk).

Ehelyett egy másik lehetőség, hogy összevonjuk a modellben levo változókat valamilyen szempontrendszer szerint. Ha ranezunk a korrelációs matrixra és a korrelációs abrakra, lathatjuk hogy vannak klaszterek a változók között, és a klasztereken belül a változók jobban korrelálnak, mint klaszterek között. Ideális lenne, ha azokat a változókat vonnánk össze amelyek ugyanazon klaszteren belül vannak. A **fokomponenselemzés** egy matematikai megoldást jelent arra, hogy ezt hogyan tehetjük meg.

A **fokomponenselemzést** (principal component analysis, roviden PCA), használhatjuk arra, hogy lecsökkentsük a változók számát amivel dolgoznunk kell, úgy, hogy közben **a lehető legtöbb információt tartunk meg az adatok variabilitásáról**.

#### 3.2 Hogyan mukodik a PCA?

A fokomponenselemzés (PCA) során az a célunk általában hogy csökkentjük a dimenziók számát, amivel az adataink leírhatók. Ezt úgy erjük el, hogy eloszor új dimenziókat keresünk, amelyek minel nagyobb reszet magyarázzák az adatok változékonyosságának, majd eldobjuk azokat a dimenziókat, amik a variancia megertesének viszonylag kis reszert felnek. A PCA során eloszor azonosítunk egy elsodleges dimenziót, ami menten az adatok a legnagyobb varianciát mutatják. Ez után azonosítunk egy erre meroleges új dimenziót, ami a **fennmarado** variancia legnagyobb reszet képes magyarázni, és így tovább, addig amíg el nem erjük az eredetileg a fokomponenselemzésbe rakott változók számát. (Mivel a dimenziók **merolegesek egymásra**,

ezért két dimenzió a variancia mindig valamilyen új reszt írja le, **nincs redundancia** a dimenziók által megmagyarázott varianciában, vagyis a fokonponensek egymással nem korrelálnak.) Így a fokonponenselemzés végére egy új koordináta-rendszert kapunk, amiben az adatokat ábrázolhatjuk.

Fontos, hogy az új koordináta-rendszer dimenziói nagyban eltérnek abban, hogy az adatok mennyire variábilisek az adott dimenzióban. Az első néhány “fokonponens” (dimenzió) amit kiválasztottunk a variancia nagyon nagy reszt lefedi, és a fennmaradó dimenziókban az adatok alig mutatnak variabilitást. Egy adott dimenzióban a variancia mértékét **eigenvalue**-nak nevezzük. Ahogy az elsőtől az utolsóig azonosított dimenzió felé haladunk, az eigenvalue egyre csökken (vagyis a dimenzióban megfigyelhető variancia). Ez engedi meg, hogy csökkentsük a dimenziók számát, mert a PCA végén azonosított dimenziókon elhanyagolható lesz az adatok különbozása egymástól, így ezeket a dimenziókat elvethetjük, és csak a **leghasznosabb dimenziókat tartjuk meg**.

Kepzeljük el például, hogy egy kutatásban arra vagyunk kíváncsiak, hogy mennyire érettek az iskolakezdő első osztályosok. A kutatásban mérjük a gyerekek verbális készségeit, szociabilitását, és életkorát éveiben. Kiderül, hogy a vizsgált mintában szinte minden első osztályos 6 éves, vagyis szinte semmi variabilitás nincs az életkorban (ha csak éveiben mérjük). Ettől a változótól akár el is tekinthetünk a kutatásunkban, hiszen annyira nem különböznek benne a vizsgált személyeink. Ezzel szemben a szociabilitásban és a verbális készségekben nagyobb személyek közötti különbséget mérünk, ezért ezek fontos indikátorok a kutatásunkban. Abból a szempontból, hogy a gyerekeket megkülönböztessük érettségük szintjében. A fokonponenselemzés során módszeresen generalunk új változókat úgy, hogy azok direkt nagyon nagy vagy nagyon kevés varianciát magyarázzanak, hogy a kevés varianciát magyarázó változókat elvethessük úgy, hogy közben minél több információt tartsunk meg az adatok különbozásáról.

### 3.3 PCA használata R-ben

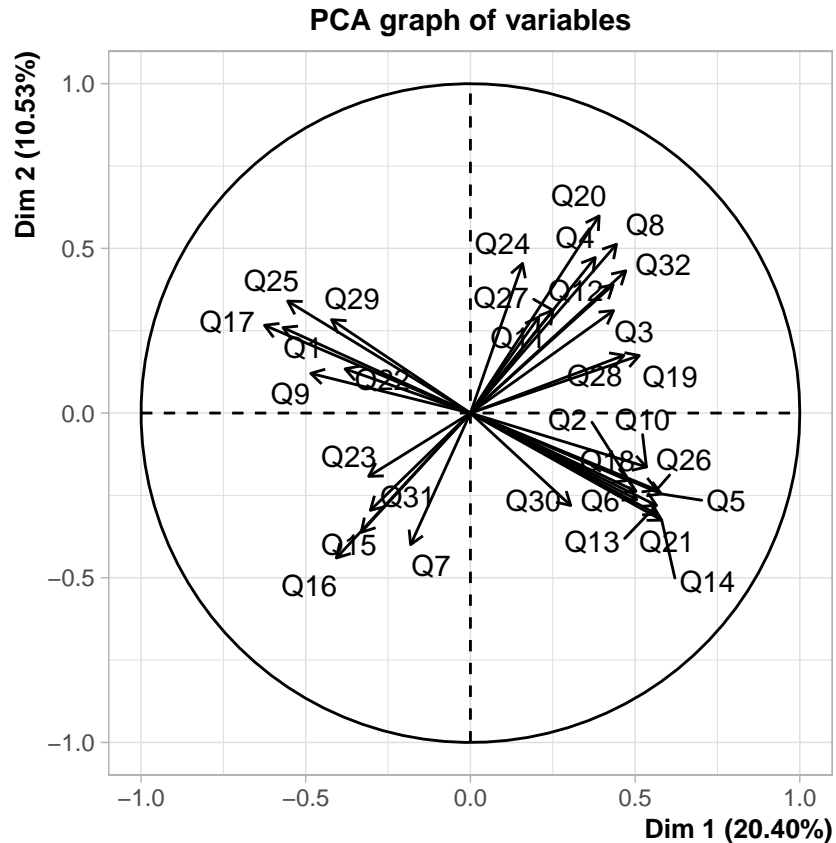
A fokonponenselemzést a **PCA()** (**principal component analysis**) funkcióval tudjuk elvégezni a **FactoMineR** package-ból. Az elemzés eredményét egy `pca_mod` modell objektumba mentettem el. Ami egyből kiad két **abrat a két legfontosabb fokonponensről (dimenzióról)** amik a leghatékonyabban írják le az adatokat.

Az egyik ábrán az látszik, hogy **az egyes megfigyelések** (ebben az esetben az egyes auto-modellek) **hol helyezkednek el a két dimenzió mentén**. A második ábra pedig arról szól, hogy a **dimenziók milyen korrelációt mutatnak az eredeti változokkal**. A szaggatott vonalak mutatják a fokonponenseket. A nyílak minél közelebb fekszenek a szaggatott vonalhoz, a változó annál inkább együttjár az adott dimenzióval a másik dimenzióval szemben.

Például a Q9 és a Q10 itemeket sokkal jobban leírja a Dim1 mint a Dim2. (a nyíl iránya alapján megállapítható, hogy a Q9 negatívan, a Q10 pozitívan korrelál a Dim1-el.) Ezzel szemben a Q8 változó nyíla a két dimenzió között helyezkedik el, ami azt jelenti, hogy midkettővel nagyjából azonos mértékben korrelál (ez lehet nagyon kicsi, vagy akár nagyon nagy korreláció is).

```
pca_mod <- PCA(hsq_items_only)
```





Arra oda kell figyelnünk hogy kategorikus változók ne kerüljenek a fokomponenselemzés változói közé.

A `PCA()` funkcióban lehetőségek van arra hogy meghatározzunk olyan változókat az adatbázisban, amiket nem szeretnénk beépíteni a PCA modellbe.

Azokat a folytonos változókat, amiket nem szeretnénk figyelembevenni a PCA során, a `quanti.sup` parameterben kell megadnunk, azokat pedig amik kategorikusak a `quali.sup` parameterben. Itt az adott változó oszlopszámát kell megadnunk, nem pedig a nevet, így ezt először ki kell keresnünk. Ezt megtehetjük a `which(names(hsq)) == "változó neve"` funkcióval.

```
which(names(hsq) == "gender")
```

```
## [1] 38
```

```
which(names(hsq) == "affiliative")
```

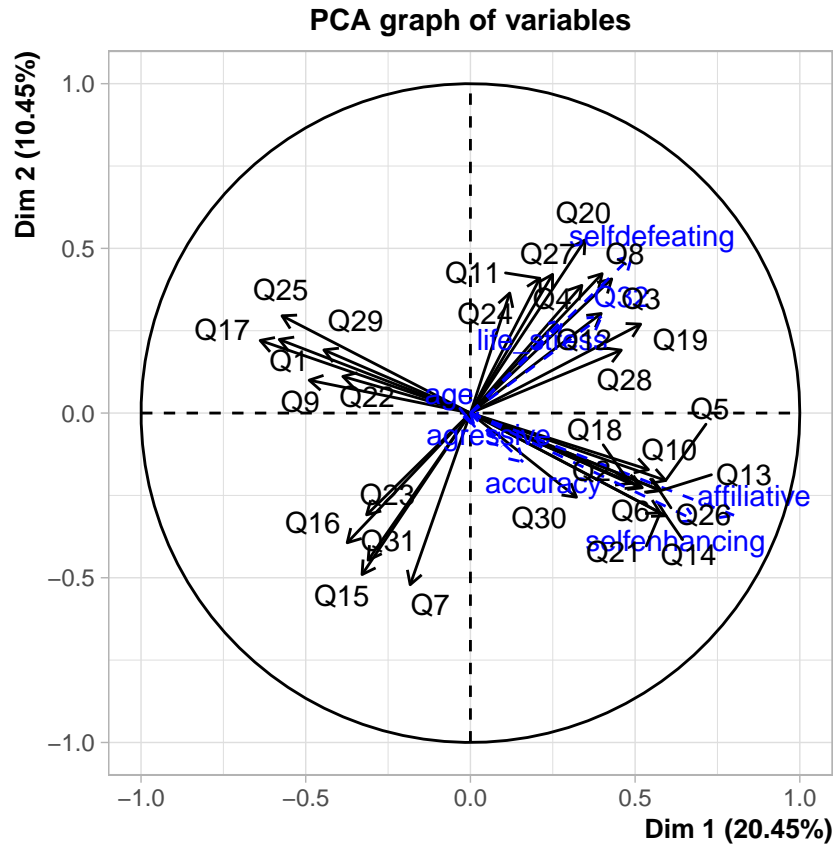
```
## [1] 33
```

```
which(names(hsq) == "life_stress")
```

```
## [1] 40
```

```
pca_mod2 <- PCA(hsq, quanti.sup = c(32, 33, 34, 35, 36, 37, 39, 40), quali.sup = 38)
```





```
summary(pca_mod2)
```

```
##
## Call:
## PCA(X = hsq, quanti.sup = c(32, 33, 34, 35, 36, 37, 39, 40),
##     quali.sup = 38)
##
##
## Eigenvalues
##
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5	Dim.6	Dim.7
## Variance	6.338	3.238	2.656	2.236	1.198	1.060	1.036
## % of var.	20.446	10.447	8.568	7.214	3.863	3.421	3.342
## Cumulative % of var.	20.446	30.892	39.461	46.675	50.538	53.959	57.301
##	Dim.8	Dim.9	Dim.10	Dim.11	Dim.12	Dim.13	Dim.14
## Variance	0.926	0.808	0.789	0.770	0.702	0.690	0.662
## % of var.	2.986	2.607	2.545	2.484	2.265	2.227	2.135
## Cumulative % of var.	60.286	62.893	65.438	67.921	70.187	72.413	74.549
##	Dim.15	Dim.16	Dim.17	Dim.18	Dim.19	Dim.20	Dim.21
## Variance	0.623	0.611	0.576	0.555	0.542	0.530	0.499
## % of var.	2.010	1.970	1.859	1.792	1.748	1.711	1.610
## Cumulative % of var.	76.559	78.529	80.388	82.180	83.928	85.639	87.248
##	Dim.22	Dim.23	Dim.24	Dim.25	Dim.26	Dim.27	Dim.28
## Variance	0.487	0.473	0.444	0.424	0.415	0.399	0.364
## % of var.	1.572	1.527	1.432	1.369	1.340	1.287	1.174
## Cumulative % of var.	88.821	90.348	91.780	93.149	94.488	95.775	96.949
##	Dim.29	Dim.30	Dim.31				

```
##
```



```

## Variance          0.349  0.309  0.288
## % of var.        1.127  0.996  0.929
## Cumulative % of var. 98.075 99.071 100.000
##
## Individuals (the 10 first)
##          Dist   Dim.1   ctr   cos2   Dim.2   ctr   cos2   Dim.3
## 1      | 3.825 | -0.679 0.007 0.032 | -1.301 0.053 0.116 | -1.436
## 2      | 4.483 | -1.969 0.062 0.193 | -0.748 0.017 0.028 | 1.133
## 3      | 3.708 | -0.137 0.000 0.001 | -1.154 0.041 0.097 | 0.223
## 4      | 4.336 | 0.015 0.000 0.000 | -3.455 0.371 0.635 | -0.377
## 5      | 3.696 | -1.985 0.063 0.289 | 0.294 0.003 0.006 | -0.176
## 6      | 7.524 | -5.089 0.412 0.458 | 2.670 0.222 0.126 | 0.235
## 7      | 4.351 | 2.136 0.073 0.241 | -0.737 0.017 0.029 | -0.686
## 8      | 4.578 | -1.164 0.022 0.065 | -1.101 0.038 0.058 | -2.182
## 9      | 9.066 | -3.920 0.244 0.187 | 1.809 0.102 0.040 | -1.511
## 10     | 5.932 | 2.837 0.128 0.229 | -2.112 0.139 0.127 | 1.446
##          ctr   cos2
## 1      | 0.078 0.141 |
## 2      | 0.049 0.064 |
## 3      | 0.002 0.004 |
## 4      | 0.005 0.008 |
## 5      | 0.001 0.002 |
## 6      | 0.002 0.001 |
## 7      | 0.018 0.025 |
## 8      | 0.181 0.227 |
## 9      | 0.087 0.028 |
## 10     | 0.079 0.059 |
##
## Variables (the 10 first)
##          Dim.1   ctr   cos2   Dim.2   ctr   cos2   Dim.3   ctr
## Q1      | -0.580 5.313 0.337 | 0.223 1.541 0.050 | 0.139 0.725
## Q2      | 0.514 4.162 0.264 | -0.224 1.547 0.050 | 0.131 0.650
## Q3      | 0.429 2.904 0.184 | 0.407 5.118 0.166 | -0.266 2.657
## Q4      | 0.338 1.806 0.114 | 0.388 4.639 0.150 | 0.461 7.985
## Q5      | 0.593 5.542 0.351 | -0.205 1.292 0.042 | -0.118 0.525
## Q6      | 0.522 4.295 0.272 | -0.227 1.587 0.051 | 0.014 0.007
## Q7      | -0.183 0.527 0.033 | -0.521 8.386 0.272 | 0.309 3.599
## Q8      | 0.400 2.524 0.160 | 0.423 5.530 0.179 | 0.503 9.515
## Q9      | -0.489 3.779 0.240 | 0.100 0.308 0.010 | 0.042 0.067
## Q10     | 0.541 4.610 0.292 | -0.172 0.910 0.029 | 0.285 3.048
##          cos2
## Q1      | 0.019 |
## Q2      | 0.017 |
## Q3      | 0.071 |
## Q4      | 0.212 |
## Q5      | 0.014 |
## Q6      | 0.000 |
## Q7      | 0.096 |
## Q8      | 0.253 |
## Q9      | 0.002 |
## Q10     | 0.081 |
##
## Supplementary continuous variables
##          Dim.1   cos2   Dim.2   cos2   Dim.3   cos2

```

```
## Q32          | 0.398 0.158 | 0.294 0.086 | 0.377 0.142 |
## affiliative  | 0.800 0.641 | -0.310 0.096 | -0.221 0.049 |
## selfenhancing | 0.683 0.467 | -0.326 0.106 | 0.225 0.051 |
## aggressive   | 0.095 0.009 | -0.101 0.010 | 0.180 0.032 |
## selfdefeating | 0.490 0.240 | 0.477 0.228 | 0.579 0.335 |
## age          | 0.012 0.000 | -0.040 0.002 | 0.024 0.001 |
## accuracy     | 0.160 0.026 | -0.149 0.022 | -0.084 0.007 |
## life_stress  | 0.280 0.079 | 0.283 0.080 | 0.329 0.108 |
##
## Supplementary categories
##          Dist   Dim.1   cos2 v.test   Dim.2   cos2 v.test   Dim.3
## 0          | 3.604 | 2.879 0.638 2.562 | -1.222 0.115 -1.521 | -0.595
## 1          | 0.428 | 0.237 0.307 3.220 | 0.284 0.439 5.385 | -0.123
## 2          | 0.506 | -0.296 0.341 -3.320 | -0.326 0.416 -5.125 | 0.147
## 3          | 2.811 | -1.347 0.230 -1.519 | -0.202 0.005 -0.318 | 0.507
##          cos2 v.test
## 0          0.027 -0.818 |
## 1          0.083 -2.583 |
## 2          0.084 2.547 |
## 3          0.032 0.882 |
```

### 3.4 Hány fokemponenst nyerjunk ki?

A fokomponenselemzés egy dimenzioredukciós technika, vagyis **celünk hogy kevesebb dimenzionk legyen** az elemzés végére, mint ahány változóval kezdtük az elemzést. Viszont hogyha ránézünk a model summary-ra, láthatjuk hogy a PCA funkció alapértelmezett módon **pontosan annyi dimenziót generált mint amennyi változónk volt**.

**Meg kell adnunk a PCA funkcionak, hogy dimenziót akarunk** kinyerni. De hogyan tudjuk eldönteni, mennyi az ideális számú dimenzió?

Erre számos módszer létezik.

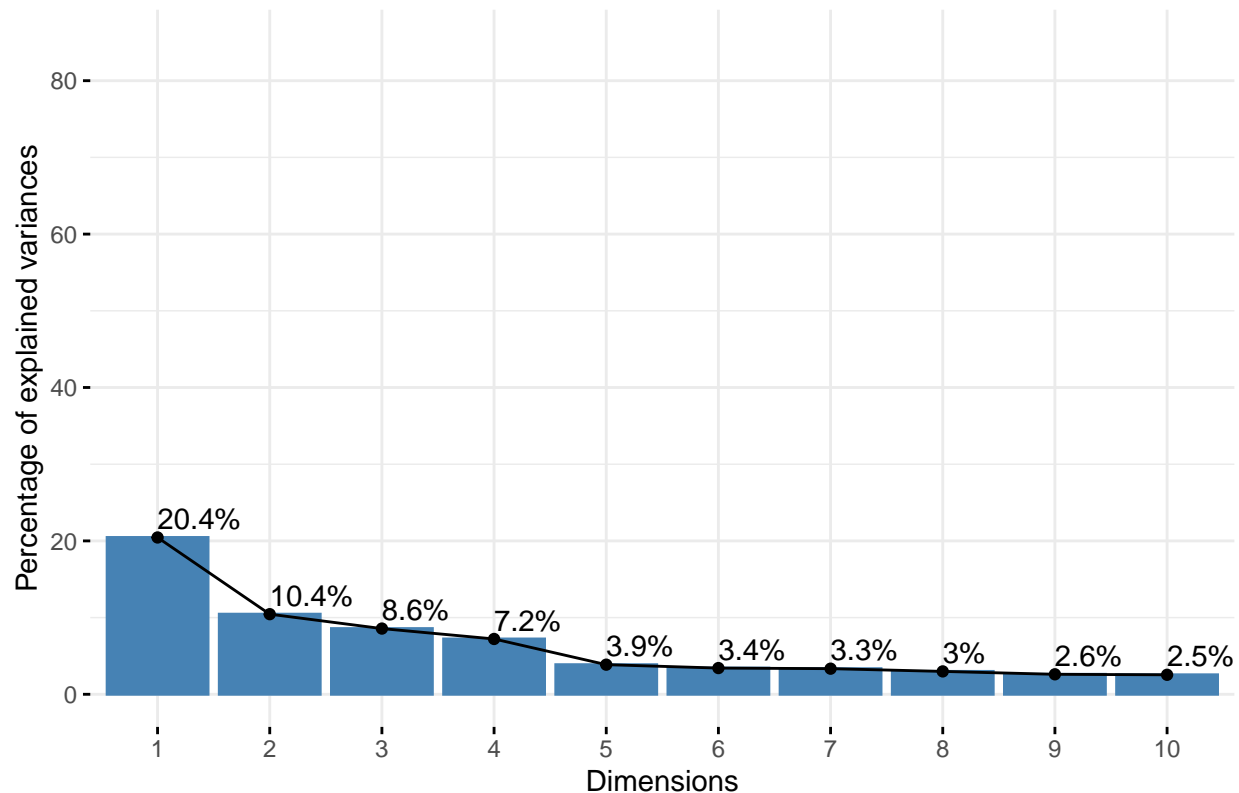
#### 1. Scree test

A legismertebb talán a scree-test, ami a megmagyarázott varianciaarányt ábrázoló ábra alapján vegezhető el. Ehhez először a `fviz_screplot()` funkcióval ábrázolnunk kell az egyes fokomponensek által megmagyarázott variancia mértéket, majd az ábra alapján meg kell állapítanunk, hol van a **“tores” a scree-plotban**, vagyis hol található az a pont, ami után már ellaposodik a megmagyarázott varianciaarányt ábrázoló görbe. **A torespont előtti dimenzional** kell hogy megálljon a dimenzió-extrakció, vagyis annál a dimenzional, ami meg szignifikánsan több varianciát képes megmagyarázni, mint a később kinyerend dimenziók. Ezt a megállási szabályt úgy is nevezzük hogy a **“konyok kritérium”**, mivel a scree plot egy konyokra emlékeztet, és mi a konyokpontot keressük a görbeben.

Ezen az ábrán úgy tűnik, hogy a **negyedik dimenzió** után már nem érdemes továbbmennünk, hiszen az ötödik dimenzional megtör a görbe és onnantól már nagyon alacsony a megmagyarázott varianciaarány. Vagyis az ideális dimenziószám ezen az adaton 4.

```
fviz_screplot(pca_mod2, addlabels = TRUE, ylim = c(0, 85))
```

## Scree plot



### The Kaiser-Guttman szabaly

Egy másik jól ismert kritérium, hogy azokat a dimenziókat kell megtartanunk, amelyeknek az **eigenvalue** értéke **1-nél magasabb**. Ez azért van, mert az **1-nél alacsonyabb** eigenvalue azt jelenti, hogy **a dimenzió kevesebb varianciát magyaráz meg mint az eredeti változók átlagosan**. A főkomponens elemzés lényege hogy **hasznos összefoglaló változókat** generáljunk amik **több változó információját tartalmazzak összevonva**. Azt pedig nem szeretnénk hogy meg az eredeti változókval is haszontalanabb változókat generáljunk, ezért az átlagos változókval kisebb varianciát megmagyarázó dimenziókat elutasítjuk.

A példánkban ez az elemzés is azt sugallja, hogy het dimenziót tartsunk meg, hiszen a nyolcadik dimenzióhoz tartozó eigenvalue már 1-nél kisebb.

```
get_eigenvalue(pca_mod2)
```

```
##      eigenvalue variance.percent cumulative.variance.percent
## Dim.1    6.3381690      20.4457065      20.44571
## Dim.2    3.2384908      10.4467446      30.89245
## Dim.3    2.6561704       8.5682915      39.46074
## Dim.4    2.2364863       7.2144721      46.67521
## Dim.5    1.1975779       3.8631545      50.53837
## Dim.6    1.0604390       3.4207709      53.95914
## Dim.7    1.0359658       3.3418250      57.30097
## Dim.8    0.9255154       2.9855336      60.28650
## Dim.9    0.8080793       2.6067073      62.89321
## Dim.10   0.7888049       2.5445319      65.43774
## Dim.11   0.7699272       2.4836360      67.92137
## Dim.12   0.7022608       2.2653575      70.18673
## Dim.13   0.6902394       2.2265788      72.41331
```

## Dim.14	0.6619274	2.1352498	74.54856
## Dim.15	0.6231722	2.0102328	76.55879
## Dim.16	0.6107458	1.9701478	78.52894
## Dim.17	0.5763014	1.8590369	80.38798
## Dim.18	0.5554613	1.7918106	82.17979
## Dim.19	0.5418445	1.7478855	83.92767
## Dim.20	0.5304575	1.7111531	85.63883
## Dim.21	0.4989770	1.6096031	87.24843
## Dim.22	0.4874399	1.5723868	88.82082
## Dim.23	0.4733018	1.5267800	90.34760
## Dim.24	0.4440585	1.4324469	91.78004
## Dim.25	0.4242522	1.3685556	93.14860
## Dim.26	0.4153240	1.3397550	94.48835
## Dim.27	0.3988724	1.2866853	95.77504
## Dim.28	0.3638795	1.1738049	96.94884
## Dim.29	0.3492450	1.1265969	98.07544
## Dim.30	0.3086315	0.9955854	99.07103
## Dim.31	0.2879818	0.9289734	100.00000

### Parallel elemzes

A harmadik (es egyben jelenleg a legelfogadottabb) technika a **parallel elemzes** technika. Ennek a lenyege hogy az eredeti adattablankhoz hasonló karakterisztikakkal rendelkező **adatokat generalunk veletlenszerűen**, de úgy, hogy abban a **változók ne korreláljanak egymással**. Ezt nagyon sokszor megismételjük, és ez alapján a nagy mennyiségű random minta alapján kiszámoljuk, **mi a veletlenszerűen várható eigenvalue mintázat**. Ez egyfajta **“null modelként”** funkcionál, amihez hasonlíthatjuk a saját adatainkon kapott eigenvalue-kat. Azokat a dimenziókat tartjuk meg, amiknek az **eigenvalue-ja magasabb mint a random mintákban az adott dimenzióhoz tartozó null-eigenvalue**.

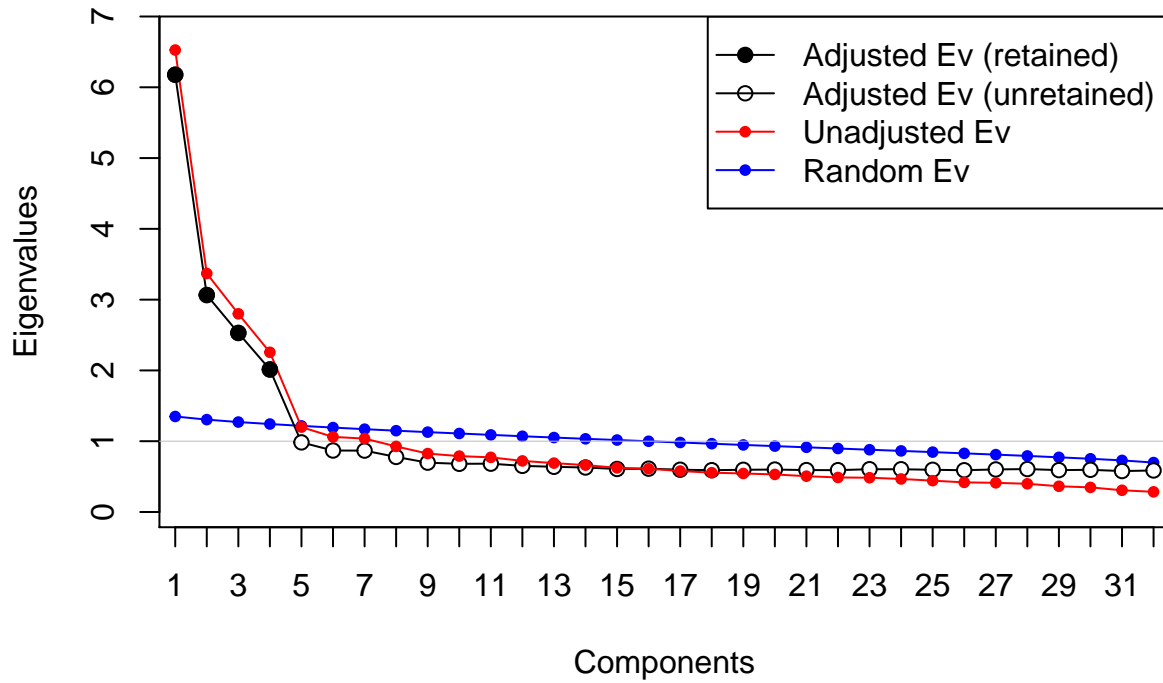
Ezt a parallel elemzést végezhetjük el a **paran()** funkcióval a paran package-ból. Ez a funkció a null eigenvalue görbe vizualizálására is képes a `graph = TRUE` paraméter beállításával, melyet összehasonlíthatunk az adatainkban kapott eigenvalue-val. Az output objektum `$Retained` komponense megmutatja, az elemzés hany dimenzió megtartását javasolja.

```
pca_ret = paran(hsq_items_only,
                 graph = TRUE)

##
## 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
## 960 iterations, using the mean estimate
##
## -----
## Component    Adjusted    Unadjusted    Estimated
##              Eigenvalue Eigenvalue    Bias
## -----
## 1             6.177089    6.526619    0.349529
## 2             3.065054    3.369944    0.304889
## 3             2.529440    2.800201    0.270760
## 4             2.013397    2.256052    0.242654
## -----
##
## Adjusted eigenvalues > 1 indicate dimensions to retain.
```

```
## (4 components retained)
```

## Parallel Analysis



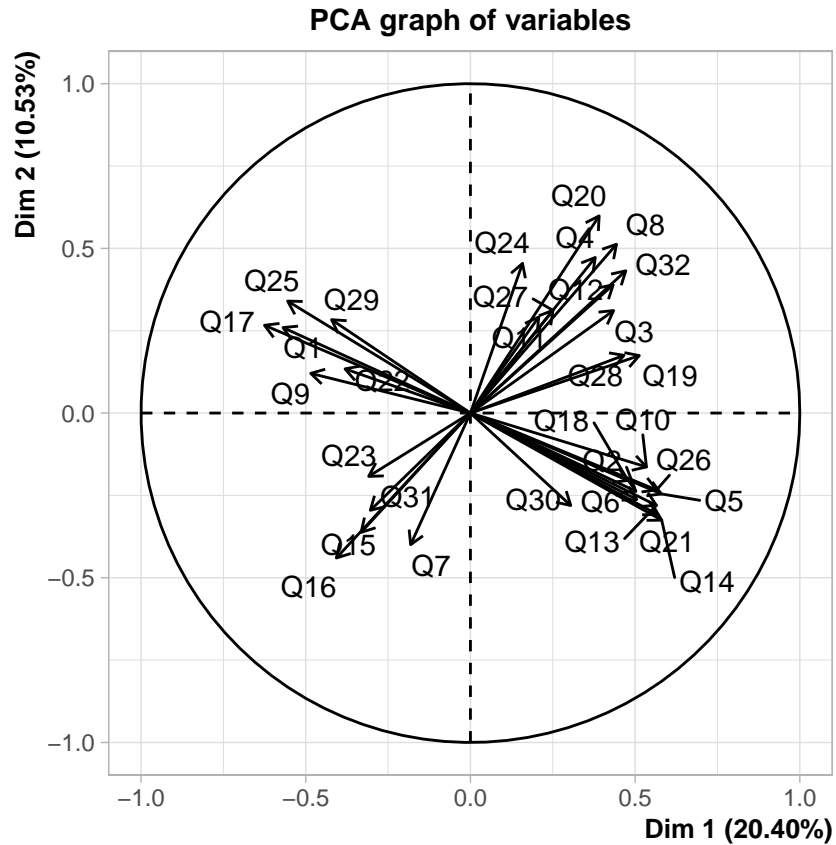
```
pca_ret$Retained
```

```
## [1] 4
```

Amint meghatároztuk az ideális dimenziók számát, újra lefuttathatjuk az elemzésünket, de ezúttal már specifikálva, mennyi dimenziót szeretnénk, a `npc` parameter beállításával.

```
pca_mod3 <- PCA(hsq_items_only, npc = 4)
```





```
summary(pca_mod3)
```

```
##
## Call:
## PCA(X = hsq_items_only, ncp = 4)
##
##
## Eigenvalues
##          Dim.1  Dim.2  Dim.3  Dim.4  Dim.5  Dim.6  Dim.7
## Variance      6.527   3.370   2.800   2.256   1.200   1.061   1.037
## % of var.     20.396  10.531   8.751   7.050   3.749   3.315   3.241
## Cumulative % of var. 20.396  30.927  39.677  46.728  50.476  53.791  57.032
##          Dim.8  Dim.9  Dim.10  Dim.11  Dim.12  Dim.13  Dim.14
## Variance      0.926   0.825   0.789   0.772   0.722   0.690   0.662
## % of var.      2.895   2.579   2.467   2.412   2.255   2.157   2.070
## Cumulative % of var. 59.927  62.506  64.973  67.385  69.640  71.797  73.867
##          Dim.15  Dim.16  Dim.17  Dim.18  Dim.19  Dim.20  Dim.21
## Variance      0.624   0.611   0.579   0.555   0.544   0.530   0.506
## % of var.      1.951   1.911   1.808   1.736   1.699   1.658   1.581
## Cumulative % of var. 75.818  77.728  79.537  81.273  82.972  84.630  86.211
##          Dim.22  Dim.23  Dim.24  Dim.25  Dim.26  Dim.27  Dim.28
## Variance      0.488   0.484   0.467   0.442   0.419   0.412   0.398
## % of var.      1.524   1.512   1.459   1.383   1.308   1.287   1.245
## Cumulative % of var. 87.735  89.247  90.706  92.089  93.397  94.684  95.929
##          Dim.29  Dim.30  Dim.31  Dim.32
## Variance      0.363   0.348   0.307   0.285
```

```
## % of var.          1.135   1.086   0.959   0.890
## Cumulative % of var. 97.064 98.151 99.110 100.000
##
## Individuals (the 10 first)
##      Dist      Dim.1      ctr      cos2      Dim.2      ctr      cos2      Dim.3      ctr
## 1 | 3.886 | -0.878 0.012 0.051 | -1.645 0.081 0.179 | -1.133 0.046
## 2 | 4.727 | -2.213 0.076 0.219 | -0.739 0.016 0.024 | 0.932 0.031
## 3 | 3.771 | -0.306 0.001 0.007 | -1.287 0.049 0.116 | 0.197 0.001
## 4 | 4.390 | -0.236 0.001 0.003 | -3.560 0.379 0.658 | 0.309 0.003
## 5 | 3.698 | -1.915 0.057 0.268 | 0.464 0.006 0.016 | -0.026 0.000
## 6 | 7.583 | -4.753 0.349 0.393 | 3.112 0.289 0.168 | -0.131 0.001
## 7 | 4.404 | 1.913 0.056 0.189 | -1.233 0.045 0.078 | -0.842 0.026
## 8 | 4.817 | -1.521 0.036 0.100 | -1.757 0.092 0.133 | -2.019 0.147
## 9 | 9.092 | -3.998 0.247 0.193 | 1.392 0.058 0.023 | -2.109 0.160
## 10 | 5.934 | 2.772 0.119 0.218 | -1.989 0.118 0.112 | 1.588 0.091
##      cos2
## 1 0.085 |
## 2 0.039 |
## 3 0.003 |
## 4 0.005 |
## 5 0.000 |
## 6 0.000 |
## 7 0.037 |
## 8 0.176 |
## 9 0.054 |
## 10 0.072 |
##
## Variables (the 10 first)
##      Dim.1      ctr      cos2      Dim.2      ctr      cos2      Dim.3      ctr      cos2
## Q1 | -0.569 4.959 0.324 | 0.259 1.997 0.067 | 0.067 0.158 0.004 |
## Q2 | 0.502 3.863 0.252 | -0.237 1.671 0.056 | 0.123 0.539 0.015 |
## Q3 | 0.435 2.893 0.189 | 0.311 2.870 0.097 | -0.370 4.876 0.137 |
## Q4 | 0.379 2.202 0.144 | 0.472 6.620 0.223 | 0.352 4.418 0.124 |
## Q5 | 0.580 5.151 0.336 | -0.246 1.789 0.060 | -0.064 0.144 0.004 |
## Q6 | 0.505 3.908 0.255 | -0.263 2.045 0.069 | 0.022 0.018 0.000 |
## Q7 | -0.181 0.505 0.033 | -0.399 4.735 0.160 | 0.464 7.677 0.215 |
## Q8 | 0.443 3.012 0.197 | 0.512 7.791 0.263 | 0.382 5.201 0.146 |
## Q9 | -0.485 3.601 0.235 | 0.120 0.431 0.015 | 0.010 0.004 0.000 |
## Q10 | 0.534 4.376 0.286 | -0.163 0.793 0.027 | 0.242 2.084 0.058 |
```

### 3.5 Fokomponenselemzés eredményeinek értelmezése

#### 3.5.1 a PCA modell objektum részei

A modell összefoglalóbol (**model summary**) további hasznos információk olvashatók ki.

Az **Eigenvalues** részben megtudhatjuk hogy az egyes dimenziók az **adatok teljes varianciajának** **hany százalékát magyarázzak meg (% of var)**, és hogy a legfontosabbtól a legalacsonyabbig egysévesel oszszvonva mekkora a több dimenzio által megmagyarázott **összesített varianciaarány (Cumulative % of var)**. Vagyis a Dim.3-hoz tartozó % of variance érték (8.75) azt mutatja, hogy a harmadikként kinyert dimenzio az adatok varianciajának 875%-át tudja megmagyarázni onmagában. Az Dim.3-hoz tartozó dim Cumulative % of var érték (39.68 pedig azt mutatja, hogy a Dimenzio 3 a Dim.1 és Dim.2-vel együtt közösen az adatok varianciajának (3968%-at képesek megmagyarázni. Ha csak az eigenvalue-t és a megmagyarázott varianciaarányokat tartalmazó táblázat érdekel minket, ezt kinyerhetjük úgy hogy csak a `pca_mod3$eig` komponenst listázzuk ki.



A model summary arrol is tartalmaz információt a Variables részében, hogy az egyes **valtozok hogyan korrelálnak az egyes új dimenziókkal** (a Dim.1, Dim.2, Dim.3 ... oszlopokban), és hogy mekkora a hozzájárulásuk az adott változó által megmagyarázott varianciahoz (a ctr oszlopban). Ez egy nagyon fontos táblázat, mert innen tudjuk leolvasni (az abrak mellett) hogy az egyes változókat mely dimenziók (faktorok) irják le leginkább. Bovebb információt találunk ha kilistazzuk a `pca_mod3$var` komponenst.

```
# Get the summary the outputs.
```

```
summary(pca_mod3)
```

```
##
## Call:
## PCA(X = hsq_items_only, ncp = 4)
##
##
## Eigenvalues
##
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5	Dim.6	Dim.7
## Variance	6.527	3.370	2.800	2.256	1.200	1.061	1.037
## % of var.	20.396	10.531	8.751	7.050	3.749	3.315	3.241
## Cumulative % of var.	20.396	30.927	39.677	46.728	50.476	53.791	57.032

```
##
```

	Dim.8	Dim.9	Dim.10	Dim.11	Dim.12	Dim.13	Dim.14
## Variance	0.926	0.825	0.789	0.772	0.722	0.690	0.662
## % of var.	2.895	2.579	2.467	2.412	2.255	2.157	2.070
## Cumulative % of var.	59.927	62.506	64.973	67.385	69.640	71.797	73.867

```
##
```

	Dim.15	Dim.16	Dim.17	Dim.18	Dim.19	Dim.20	Dim.21
## Variance	0.624	0.611	0.579	0.555	0.544	0.530	0.506
## % of var.	1.951	1.911	1.808	1.736	1.699	1.658	1.581
## Cumulative % of var.	75.818	77.728	79.537	81.273	82.972	84.630	86.211

```
##
```

	Dim.22	Dim.23	Dim.24	Dim.25	Dim.26	Dim.27	Dim.28
## Variance	0.488	0.484	0.467	0.442	0.419	0.412	0.398
## % of var.	1.524	1.512	1.459	1.383	1.308	1.287	1.245
## Cumulative % of var.	87.735	89.247	90.706	92.089	93.397	94.684	95.929

```
##
```

	Dim.29	Dim.30	Dim.31	Dim.32
## Variance	0.363	0.348	0.307	0.285
## % of var.	1.135	1.086	0.959	0.890
## Cumulative % of var.	97.064	98.151	99.110	100.000

```
##
## Individuals (the 10 first)
##
```

	Dist	Dim.1	ctr	cos2	Dim.2	ctr	cos2	Dim.3	ctr
## 1	3.886	-0.878	0.012	0.051	-1.645	0.081	0.179	-1.133	0.046
## 2	4.727	-2.213	0.076	0.219	-0.739	0.016	0.024	0.932	0.031
## 3	3.771	-0.306	0.001	0.007	-1.287	0.049	0.116	0.197	0.001
## 4	4.390	-0.236	0.001	0.003	-3.560	0.379	0.658	0.309	0.003
## 5	3.698	-1.915	0.057	0.268	0.464	0.006	0.016	-0.026	0.000
## 6	7.583	-4.753	0.349	0.393	3.112	0.289	0.168	-0.131	0.001
## 7	4.404	1.913	0.056	0.189	-1.233	0.045	0.078	-0.842	0.026
## 8	4.817	-1.521	0.036	0.100	-1.757	0.092	0.133	-2.019	0.147
## 9	9.092	-3.998	0.247	0.193	1.392	0.058	0.023	-2.109	0.160
## 10	5.934	2.772	0.119	0.218	-1.989	0.118	0.112	1.588	0.091

```
##
## cos2
## 1 0.085 |
## 2 0.039 |
## 3 0.003 |
## 4 0.005 |
## 5 0.000 |
## 6 0.000 |
```

```

## 7    0.037 |
## 8    0.176 |
## 9    0.054 |
## 10   0.072 |
##
## Variables (the 10 first)
##      Dim.1    ctr    cos2    Dim.2    ctr    cos2    Dim.3    ctr    cos2
## Q1 | -0.569  4.959  0.324 |  0.259  1.997  0.067 |  0.067  0.158  0.004 |
## Q2 |  0.502  3.863  0.252 | -0.237  1.671  0.056 |  0.123  0.539  0.015 |
## Q3 |  0.435  2.893  0.189 |  0.311  2.870  0.097 | -0.370  4.876  0.137 |
## Q4 |  0.379  2.202  0.144 |  0.472  6.620  0.223 |  0.352  4.418  0.124 |
## Q5 |  0.580  5.151  0.336 | -0.246  1.789  0.060 | -0.064  0.144  0.004 |
## Q6 |  0.505  3.908  0.255 | -0.263  2.045  0.069 |  0.022  0.018  0.000 |
## Q7 | -0.181  0.505  0.033 | -0.399  4.735  0.160 |  0.464  7.677  0.215 |
## Q8 |  0.443  3.012  0.197 |  0.512  7.791  0.263 |  0.382  5.201  0.146 |
## Q9 | -0.485  3.601  0.235 |  0.120  0.431  0.015 |  0.010  0.004  0.000 |
## Q10 | 0.534  4.376  0.286 | -0.163  0.793  0.027 |  0.242  2.084  0.058 |

```

```
pca_mod3$eig
```

```

##      eigenvalue percentage of variance cumulative percentage of variance
## comp 1    6.5266194                20.3956857                20.39569
## comp 2    3.3699443                10.5310758                30.92676
## comp 3    2.8002011                 8.7506285                39.67739
## comp 4    2.2560527                 7.0501646                46.72755
## comp 5    1.1995880                 3.7487124                50.47627
## comp 6    1.0608149                 3.3150464                53.79131
## comp 7    1.0370806                 3.2408769                57.03219
## comp 8    0.9263495                 2.8948422                59.92703
## comp 9    0.8252450                 2.5788906                62.50592
## comp 10   0.7894123                 2.4669134                64.97284
## comp 11   0.7718522                 2.4120381                67.38487
## comp 12   0.7216551                 2.2551722                69.64005
## comp 13   0.6903207                 2.1572520                71.79730
## comp 14   0.6622550                 2.0695470                73.86685
## comp 15   0.6242740                 1.9508561                75.81770
## comp 16   0.6114501                 1.9107815                77.72848
## comp 17   0.5786557                 1.8082991                79.53678
## comp 18   0.5554699                 1.7358435                81.27263
## comp 19   0.5437363                 1.6991759                82.97180
## comp 20   0.5304791                 1.6577472                84.62955
## comp 21   0.5059257                 1.5810178                86.21057
## comp 22   0.4877984                 1.5243699                87.73494
## comp 23   0.4839852                 1.5124538                89.24739
## comp 24   0.4668567                 1.4589273                90.70632
## comp 25   0.4424163                 1.3825510                92.08887
## comp 26   0.4185422                 1.3079445                93.39681
## comp 27   0.4118812                 1.2871287                94.68394
## comp 28   0.3984002                 1.2450006                95.92894
## comp 29   0.3633398                 1.1354370                97.06438
## comp 30   0.3476741                 1.0864816                98.15086
## comp 31   0.3069190                 0.9591218                99.10998
## comp 32   0.2848053                 0.8900167                100.00000

```

```
pca_mod3$var
```

```
## $coord
##          Dim.1      Dim.2      Dim.3      Dim.4
## Q1 -0.5688915  0.2594476  0.06654987  0.34960522
## Q2  0.5021178 -0.2373143  0.12284157  0.38417814
## Q3  0.4345612  0.3110200 -0.36950854  0.02163271
## Q4  0.3790587  0.4723304  0.35171209 -0.08316700
## Q5  0.5798081 -0.2455561 -0.06358979 -0.27455182
## Q6  0.5050051 -0.2625465  0.02225282  0.18236620
## Q7 -0.1814692 -0.3994541  0.46364367 -0.20124909
## Q8  0.4433948  0.5123844  0.38163200 -0.10676883
## Q9 -0.4847889  0.1204558  0.01028084  0.29329403
## Q10 0.5344008 -0.1634825  0.24158276  0.49608475
## Q11 0.2073377  0.2914197 -0.45068030  0.25594805
## Q12 0.4334654  0.3894721  0.38434674 -0.17975417
## Q13 0.5646683 -0.2802503 -0.09285473 -0.27570231
## Q14 0.5806946 -0.3250714  0.12846727  0.33887960
## Q15 -0.3316783 -0.3620736  0.52834991 -0.08033977
## Q16 -0.4069220 -0.4398788 -0.20043300  0.13094951
## Q17 -0.6251763  0.2665194  0.09778869  0.39147217
## Q18 0.4781092 -0.2045136  0.22455628  0.53160685
## Q19 0.5127333  0.1744786 -0.29166246  0.06097239
## Q20 0.3907534  0.5985140  0.31822294 -0.07475584
## Q21 0.5672222 -0.3077202  0.06856781 -0.36662308
## Q22 -0.3804674  0.1341635 -0.01415713 -0.25322693
## Q23 -0.3088811 -0.1922108  0.44485076 -0.07375981
## Q24 0.1581873  0.4544262  0.34650080  0.05169798
## Q25 -0.5549809  0.3399577  0.09318884  0.33462739
## Q26 0.5650082 -0.2333713  0.19231956  0.38389427
## Q27 0.2506611  0.3119157 -0.44034085  0.13298736
## Q28 0.4660091  0.1752175  0.03172224  0.06895569
## Q29 -0.4215349  0.2835606  0.30889584  0.26818215
## Q30 0.3045698 -0.2806700  0.04587713  0.38421190
## Q31 -0.3038096 -0.2951999  0.60015283 -0.07033962
## Q32 0.4717226  0.4318453  0.36617067 -0.12580811
##
## $cor
##          Dim.1      Dim.2      Dim.3      Dim.4
## Q1 -0.5688915  0.2594476  0.06654987  0.34960522
## Q2  0.5021178 -0.2373143  0.12284157  0.38417814
## Q3  0.4345612  0.3110200 -0.36950854  0.02163271
## Q4  0.3790587  0.4723304  0.35171209 -0.08316700
## Q5  0.5798081 -0.2455561 -0.06358979 -0.27455182
## Q6  0.5050051 -0.2625465  0.02225282  0.18236620
## Q7 -0.1814692 -0.3994541  0.46364367 -0.20124909
## Q8  0.4433948  0.5123844  0.38163200 -0.10676883
## Q9 -0.4847889  0.1204558  0.01028084  0.29329403
## Q10 0.5344008 -0.1634825  0.24158276  0.49608475
## Q11 0.2073377  0.2914197 -0.45068030  0.25594805
## Q12 0.4334654  0.3894721  0.38434674 -0.17975417
## Q13 0.5646683 -0.2802503 -0.09285473 -0.27570231
## Q14 0.5806946 -0.3250714  0.12846727  0.33887960
## Q15 -0.3316783 -0.3620736  0.52834991 -0.08033977
```

```

## Q16 -0.4069220 -0.4398788 -0.20043300 0.13094951
## Q17 -0.6251763 0.2665194 0.09778869 0.39147217
## Q18 0.4781092 -0.2045136 0.22455628 0.53160685
## Q19 0.5127333 0.1744786 -0.29166246 0.06097239
## Q20 0.3907534 0.5985140 0.31822294 -0.07475584
## Q21 0.5672222 -0.3077202 0.06856781 -0.36662308
## Q22 -0.3804674 0.1341635 -0.01415713 -0.25322693
## Q23 -0.3088811 -0.1922108 0.44485076 -0.07375981
## Q24 0.1581873 0.4544262 0.34650080 0.05169798
## Q25 -0.5549809 0.3399577 0.09318884 0.33462739
## Q26 0.5650082 -0.2333713 0.19231956 0.38389427
## Q27 0.2506611 0.3119157 -0.44034085 0.13298736
## Q28 0.4660091 0.1752175 0.03172224 0.06895569
## Q29 -0.4215349 0.2835606 0.30889584 0.26818215
## Q30 0.3045698 -0.2806700 0.04587713 0.38421190
## Q31 -0.3038096 -0.2951999 0.60015283 -0.07033962
## Q32 0.4717226 0.4318453 0.36617067 -0.12580811
##
## $cos2
##          Dim.1          Dim.2          Dim.3          Dim.4
## Q1 0.32363753 0.06731304 0.0044288853 0.1222238112
## Q2 0.25212231 0.05631808 0.0150900513 0.1475928452
## Q3 0.18884342 0.09673342 0.1365365582 0.0004679743
## Q4 0.14368548 0.22309599 0.1237013940 0.0069167499
## Q5 0.33617747 0.06029780 0.0040436615 0.0753787019
## Q6 0.25503012 0.06893065 0.0004951881 0.0332574304
## Q7 0.03293105 0.15956359 0.2149654501 0.0405011948
## Q8 0.19659894 0.26253776 0.1456429798 0.0113995829
## Q9 0.23502023 0.01450960 0.0001056956 0.0860213884
## Q10 0.28558417 0.02672654 0.0583622301 0.2461000811
## Q11 0.04298891 0.08492543 0.2031127326 0.0655094020
## Q12 0.18789226 0.15168850 0.1477224158 0.0323115623
## Q13 0.31885026 0.07854023 0.0086220015 0.0760117628
## Q14 0.33720617 0.10567144 0.0165038392 0.1148393858
## Q15 0.11001049 0.13109731 0.2791536322 0.0064544782
## Q16 0.16558554 0.19349334 0.0401733860 0.0171477733
## Q17 0.39084539 0.07103259 0.0095626284 0.1532504575
## Q18 0.22858840 0.04182580 0.0504255208 0.2826058383
## Q19 0.26289547 0.03044277 0.0850669913 0.0037176325
## Q20 0.15268822 0.35821901 0.1012658418 0.0055884355
## Q21 0.32174097 0.09469174 0.0047015447 0.1344124851
## Q22 0.14475541 0.01799985 0.0002004244 0.0641238788
## Q23 0.09540756 0.03694500 0.1978921964 0.0054405091
## Q24 0.02502321 0.20650317 0.1200628061 0.0026726810
## Q25 0.30800382 0.11557126 0.0086841601 0.1119754922
## Q26 0.31923424 0.05446216 0.0369868149 0.1473748141
## Q27 0.06283097 0.09729141 0.1939000666 0.0176856369
## Q28 0.21716450 0.03070119 0.0010063002 0.0047548877
## Q29 0.17769165 0.08040662 0.0954166400 0.0719216657
## Q30 0.09276277 0.07877568 0.0021047113 0.1476187847
## Q31 0.09230027 0.08714298 0.3601834227 0.0049476626
## Q32 0.22252219 0.18649034 0.1340809568 0.0158276803
##
## $contrib

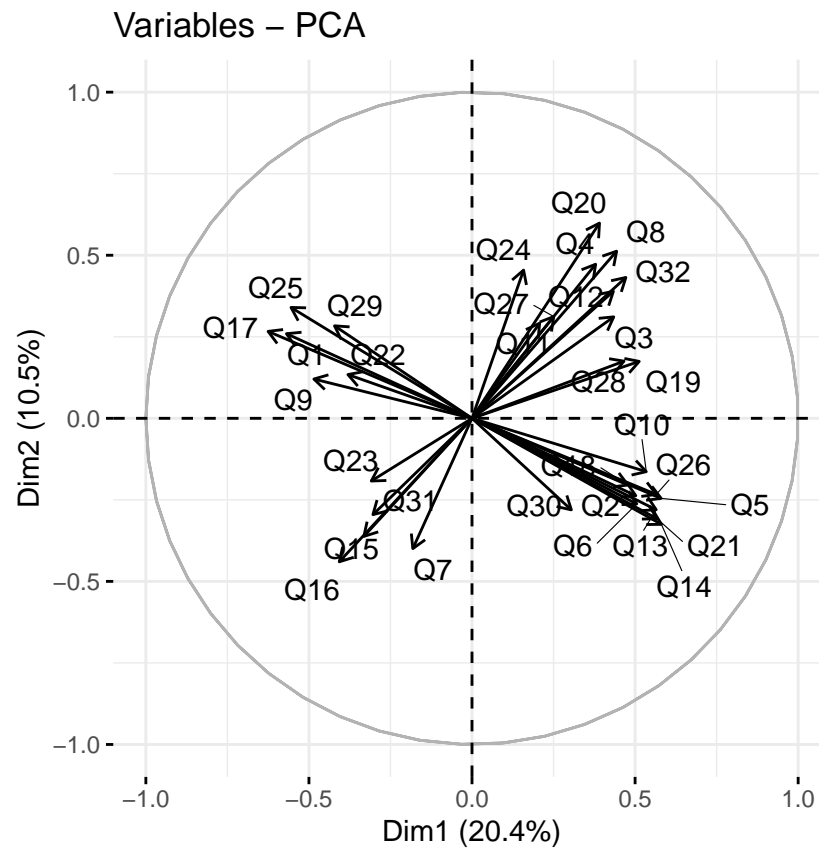
```

##	Dim.1	Dim.2	Dim.3	Dim.4
## Q1	4.9587314	1.9974527	0.158163113	5.41759565
## Q2	3.8629848	1.6711872	0.538891696	6.54208332
## Q3	2.8934339	2.8704752	4.875955405	0.02074306
## Q4	2.2015299	6.6201684	4.417589606	0.30658637
## Q5	5.1508667	1.7892817	0.144406109	3.34117652
## Q6	3.9075378	2.0454536	0.017684020	1.47414246
## Q7	0.5045652	4.7349029	7.676786071	1.79522382
## Q8	3.0122629	7.7905668	5.201161386	0.50528887
## Q9	3.6009489	0.4305590	0.003774572	3.81291579
## Q10	4.3756829	0.7930854	2.084215649	10.90843688
## Q11	0.6586704	2.5200841	7.253505134	2.90371776
## Q12	2.8788603	4.5012169	5.275421624	1.43221667
## Q13	4.8853816	2.3306092	0.307906509	3.36923707
## Q14	5.1666284	3.1357028	0.589380493	5.09027947
## Q15	1.6855662	3.8901922	9.969056487	0.28609608
## Q16	2.5370798	5.7417372	1.434660733	0.76007859
## Q17	5.9884814	2.1078270	0.341497912	6.79285815
## Q18	3.5024013	1.2411422	1.800782103	12.52656210
## Q19	4.0280496	0.9033613	3.037888616	0.16478483
## Q20	2.3394687	10.6298201	3.616377438	0.24770856
## Q21	4.9296726	2.8098905	0.167900250	5.95786114
## Q22	2.2179233	0.5341289	0.007157501	2.84230416
## Q23	1.4618221	1.0963089	7.067070807	0.24115169
## Q24	0.3834023	6.1277920	4.287649373	0.11846714
## Q25	4.7191938	3.4294709	0.310126298	4.96333680
## Q26	4.8912649	1.6161145	1.320862795	6.53241905
## Q27	0.9626878	2.8870333	6.924504981	0.78391951
## Q28	3.3273657	0.9110296	0.035936712	0.21076138
## Q29	2.7225680	2.3859925	3.407492377	3.18794267
## Q30	1.4212989	2.3375958	0.075162861	6.54323309
## Q31	1.4142126	2.5858878	12.862769718	0.21930617
## Q32	3.4094556	5.5339294	4.788261651	0.70156519

### 3.5.2 Az eredmények vizualizálása

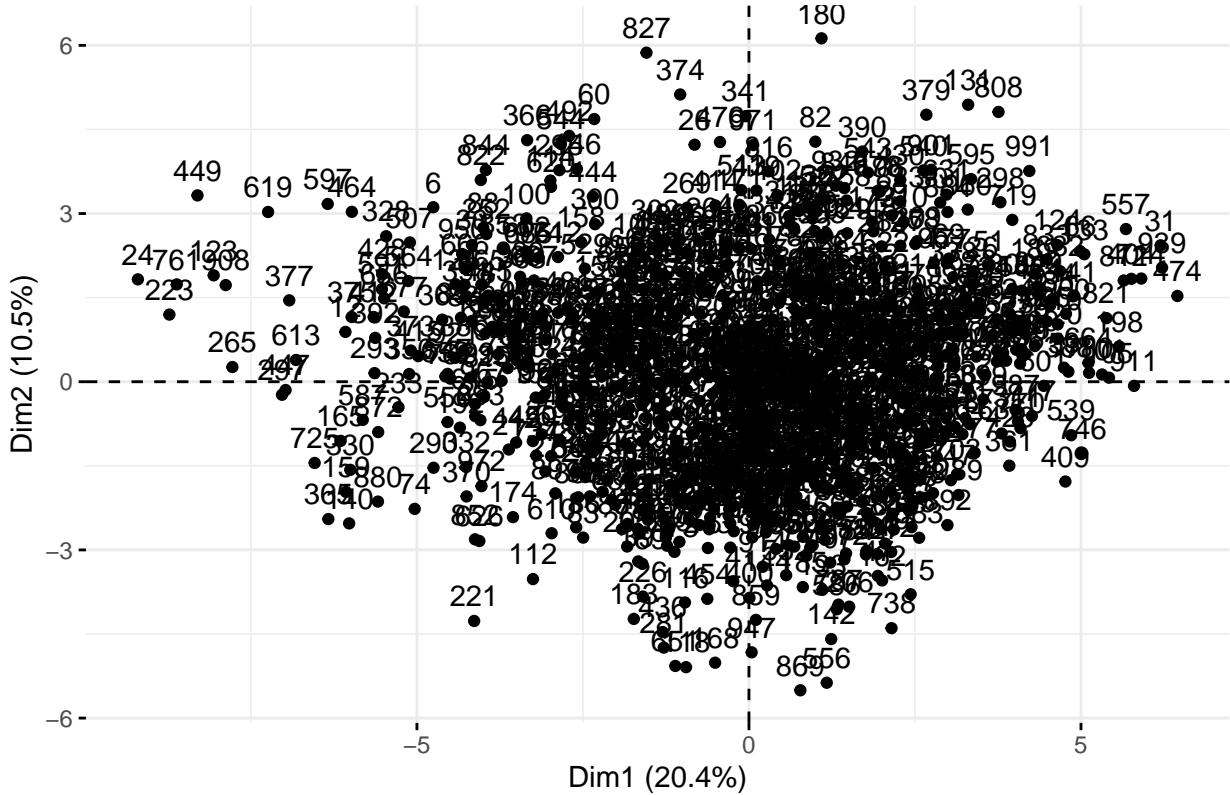
Az eredmények vizualizálása segíthet a komponensek értelmezésében. A `fviz_pca_var()` és a `fviz_pca_ind()` segítségével reprodukálhatjuk a PCA funció által eredetileg general abrakat. Sőt, a kettőt össze is vonhatjuk a `fviz_pca_biplot()` funkcióval. Így egyszerre láthatjuk hogy a két legfontosabb dimenzio mentén hol helyezkednek el az egyes megfigyelések (az autók), és hogy a dimenziók főleg mely változókat reprezentálják. (A `repel = T` parameterbeállítás arra jó hogy a feliratok ne fedjek egymást hanem elcsúsztatva szerepeljenek az ábrán ha túl közel lennének egymáshoz)

```
fviz_pca_var(pca_mod3, repel = T)
```

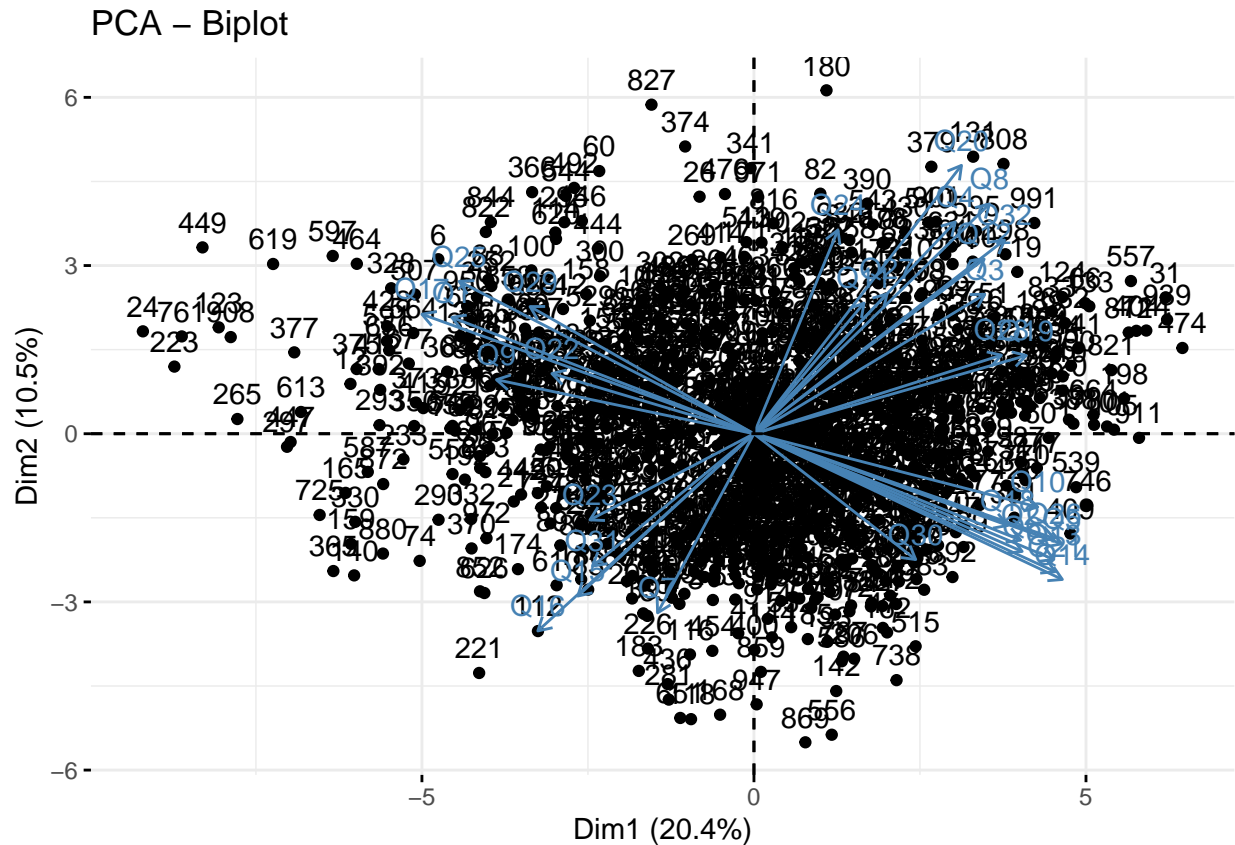


```
fviz_pca_ind(pca_mod3)
```

## Individuals – PCA



```
fviz_pca_biplot(pca_mod3)
```

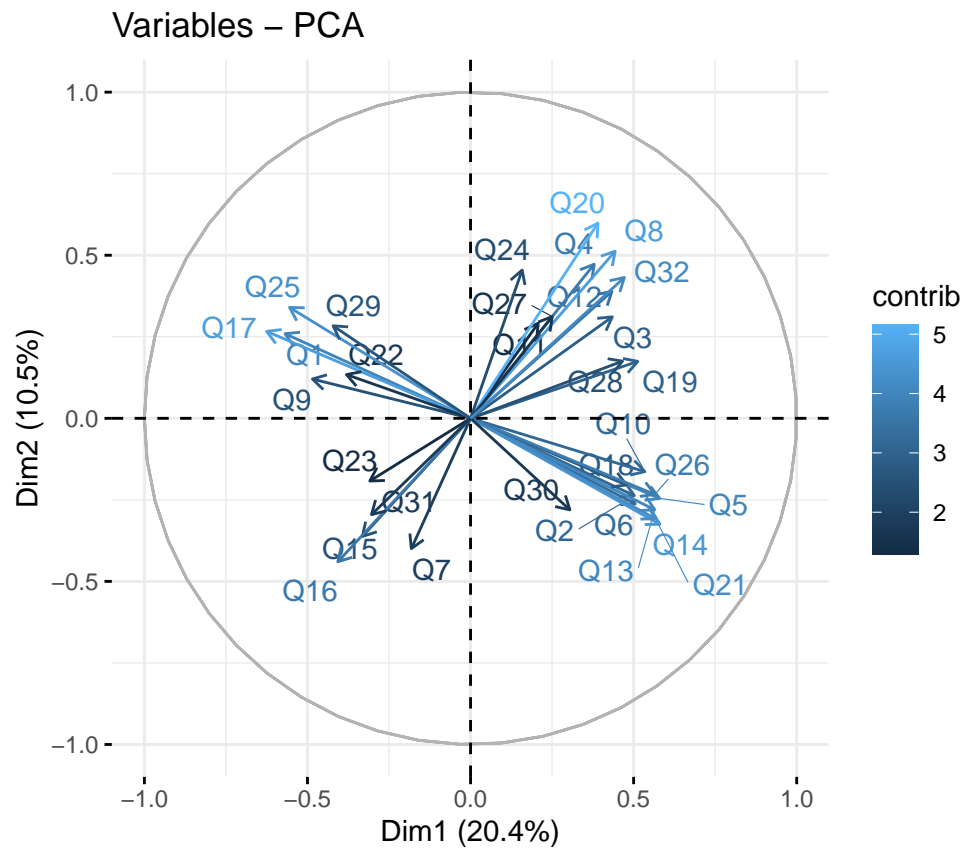


Az abrakat tovább tuningolhatjuk azzal, hogy abrazoljuk rajtuk az egyes változók vagy megfigyelesek hozzájárulását (contribution) az abrazolt dimenzióhoz a , col.var = "contrib" és , col.ind = "contrib" paramétereken keresztül.

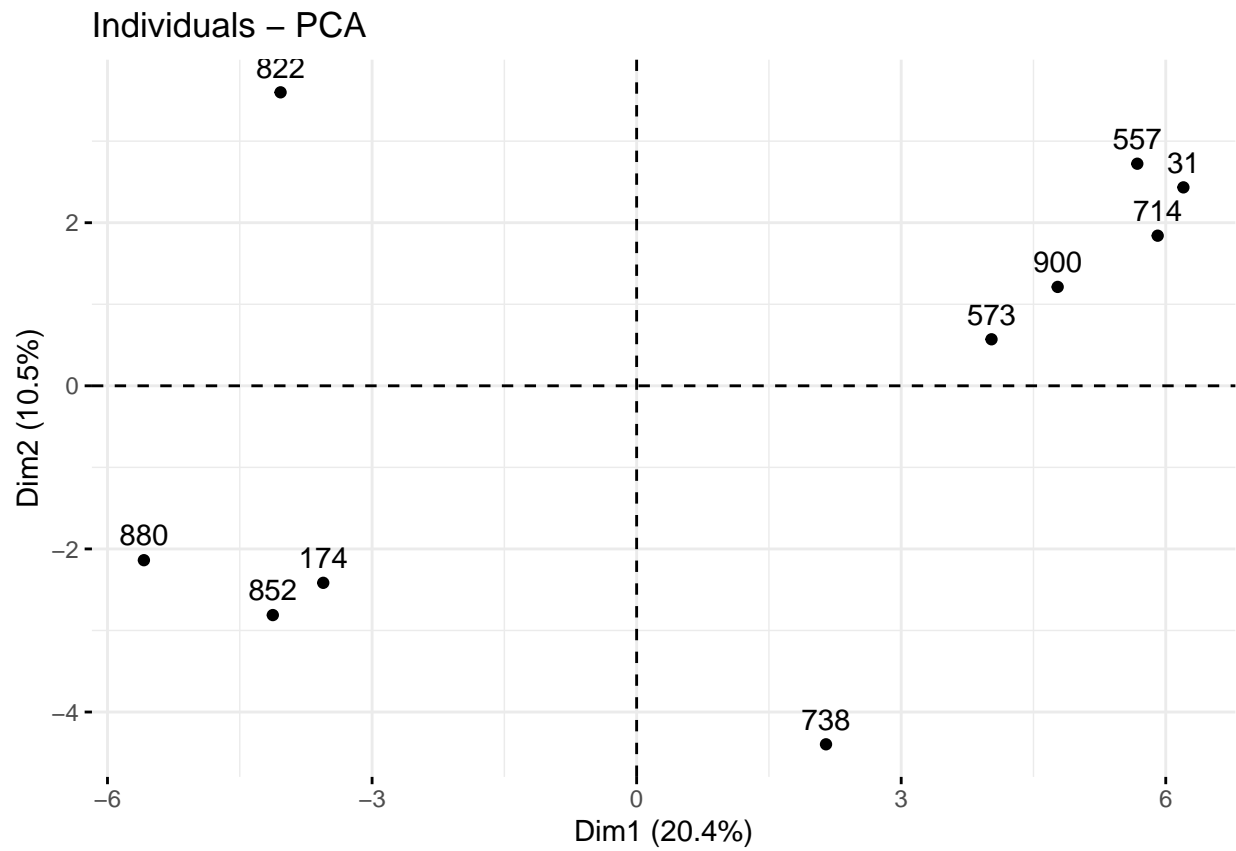
Azt is megtehetjük, hogy a select.ind = parameteren keresztül hogy csak bizonyos megfigyelesek tesztünk az abrara.Pl.  $\cos^2$  értékek azt mutatja, hogy az adott megfigyeles vagy változó mennyire jól reprezentált az adott dimenzió által. A select.ind = list(cos2 = 10) parameter beállításával meghatározhatjuk, hogy csak az a 10 megfigyeles szerepeljen az abrán, akiknek a két dimenzióra vonatkozó  $\cos^2$  összege a legmagasabb. Vagyis a két dimenzió által leírt dimenziótér 10 legreprezentatívabb megfigyelese.

```
fviz_pca_var(pca_mod3, repel = T, col.var = "contrib")
```

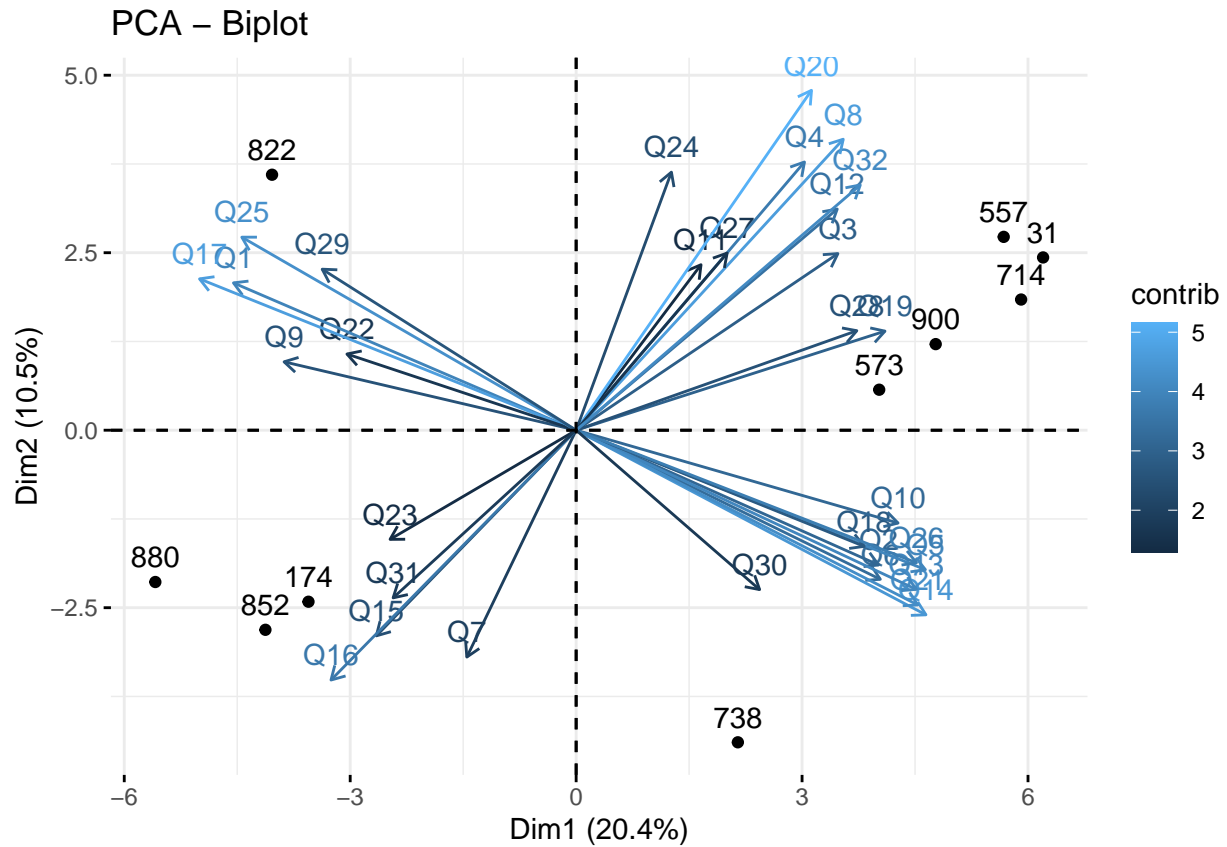




```
fviz_pca_ind(pca_mod3, select.ind = list(cos2 = 10))
```

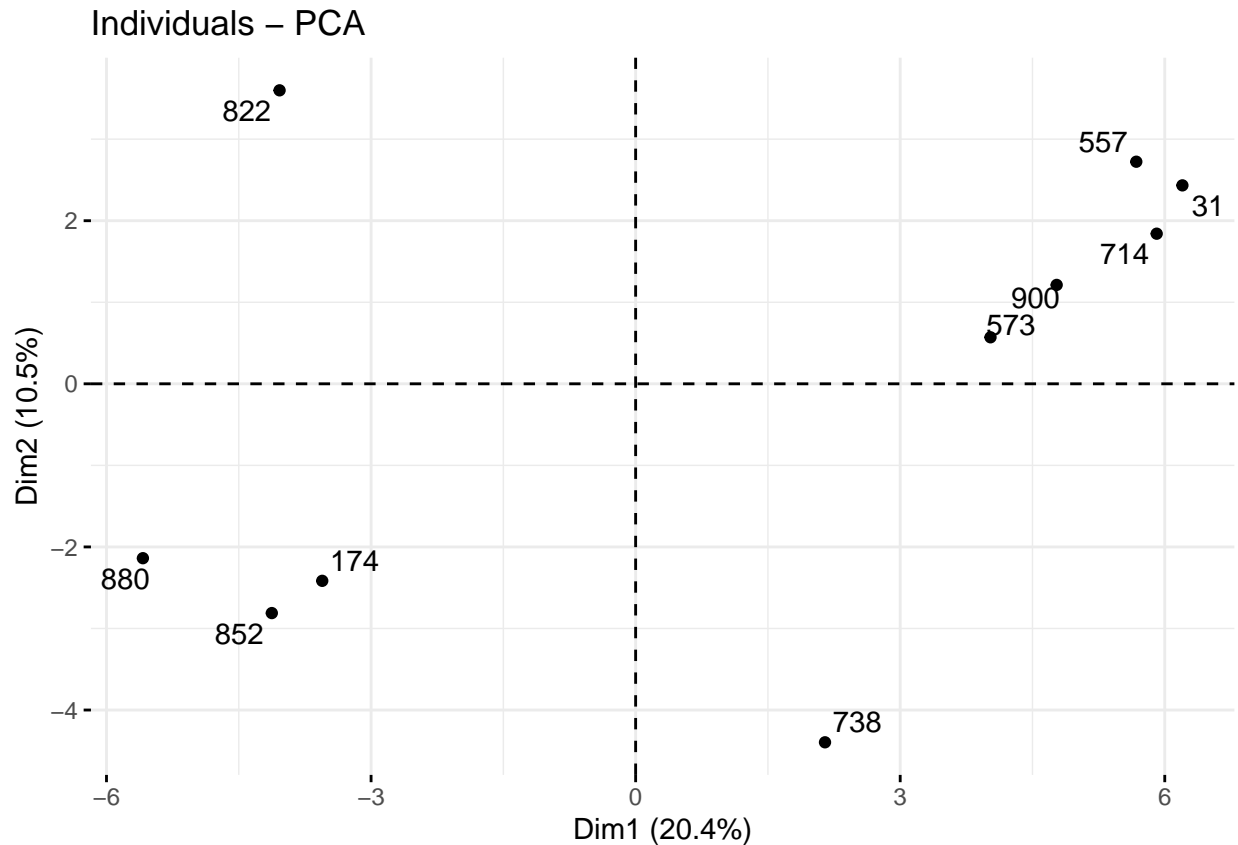


```
fviz_pca_biplot(pca_mod3, select.ind = list(cos2 = 10), col.var = "contrib")
```



Ez az ábra azt mutatja hogy a magas Dim.1, közepes Dim.2 legtipikusabb tagjai pl. az 573-as és a 900-as megfigyelesek, az alacsony Dim.2. közepes Dim.1 legtipikusabb tagja talán a 738-as személy, és az alacsony Dim.1. és magas Dim.2. legtipikusabb tagja a 822-es megfigyeles. Ez fontos lehet a dimenziók értelmezésében.

```
fviz_pca_ind(pca_mod3, select.ind = list(cos2 = 10), repel = T)
```



Egy másik fontos ábratípus az egyes dimenziók értelmezésének elősegítéséhez a `fviz_contrib()` által generált **barchart**, ami az egyes változók egyes dimenziókhoz való hozzájárulását mutatja meg.

Az `axes =` **parameterrel** állíthatjuk be, **melyik dimenzióra** vagyunk kíváncsiak. A **piros szaggatott vonal** azt mutatja, hogy mi lenne az **elvárt hozzájárulás százaléka** abban az esetben ha minden változó azonos mértékben járulna hozzá a dimenzio megmagyarázásához.

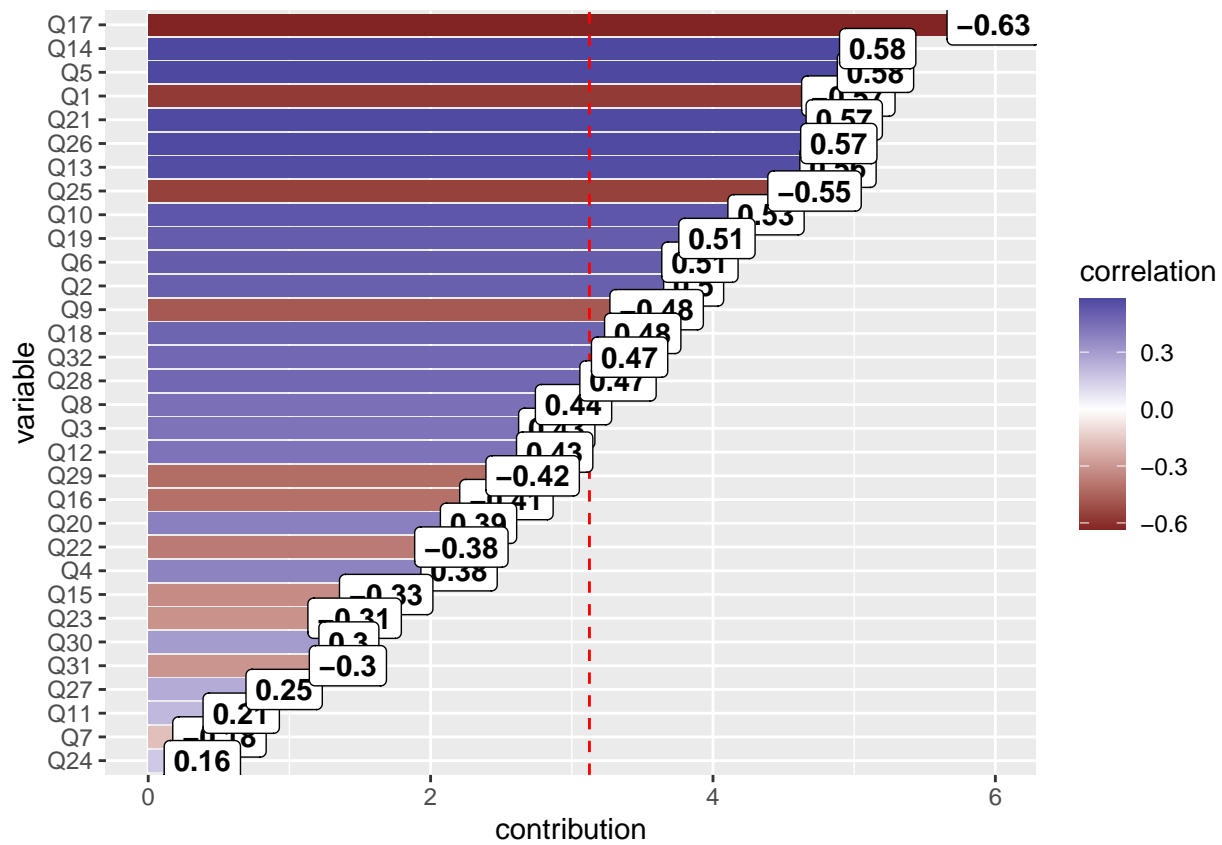
Ez az ábra akkor lenne igazán informatív ha a korreláció mértéke és iránya is egyértelmű lenne róla. Ezt onmagában nem tartalmazza a `fviz_contrib()` funkció, ezért a `fviz_loadings_with_cor()` saját funkció használatával helyettesítjük, melyen az oszlopok a korreláció szerint vannak színezve és a korreláció felíratként is szerepel az ábrán.

Ezek az ábra azt mutatják, hogy a Dim.1-hez elsősorban Q14, Q17, Q5, Q1 változók járulnak hozzá, míg a Dim.2-hoz elsősorban a Q20, Q8, Q4 és Q24 változók járulnak hozzá, míg a Dim.3-hoz a Q31, Q15, Q7.

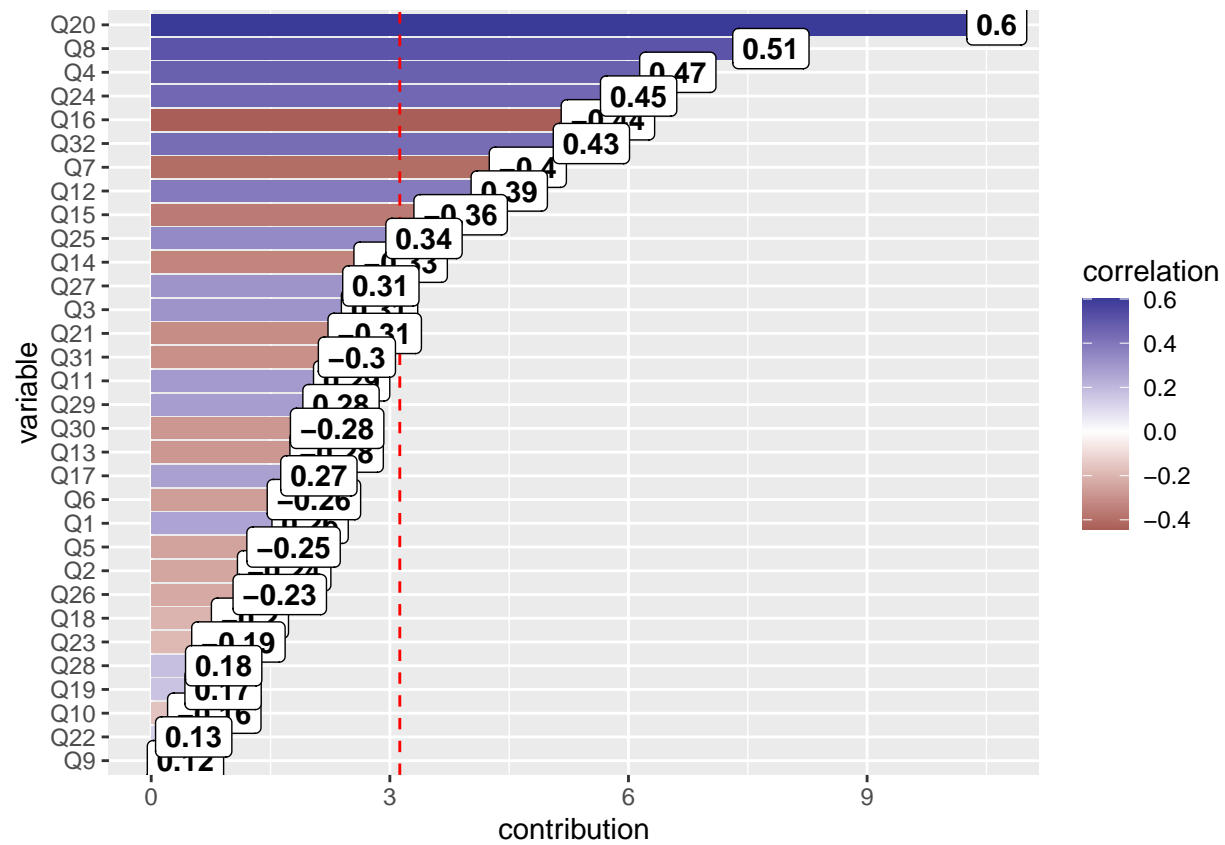
Ezek alapján az ábrák alapján, és a reprezentatív esetek ábrája alapján mit gondolsz, hogyan nevezhetnénk el az egyes dimenziókat?

```
# original functions in factoextra
# fviz_contrib(pca_mod3, choice = "var", axes = 1)
# fviz_contrib(pca_mod3, choice = "var", axes = 2)
# fviz_contrib(pca_mod3, choice = "var", axes = 3)

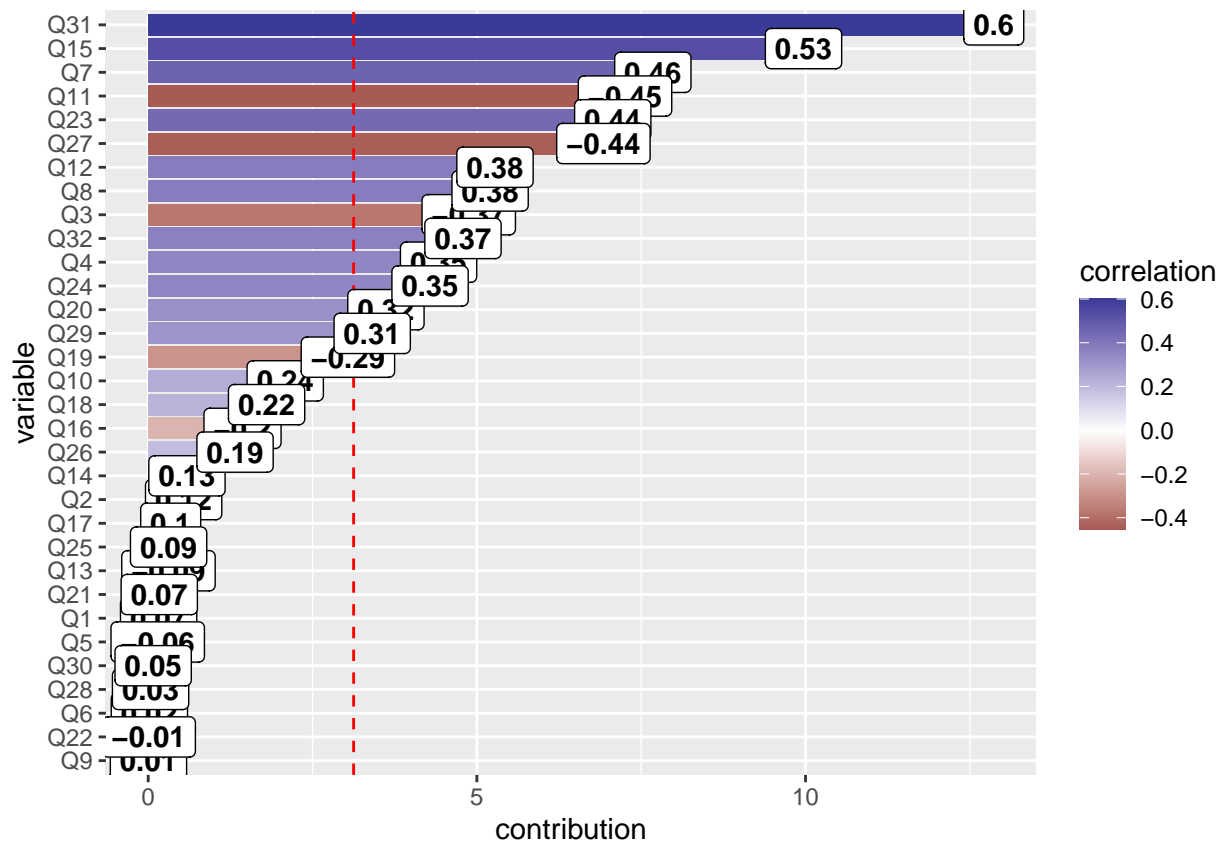
# using custom function for correlation color gradient
fviz_loadings_with_cor(mod = pca_mod3, axes = 1)
```



```
fviz_loadnings_with_cor(mod = pca_mod3, axes = 2)
```



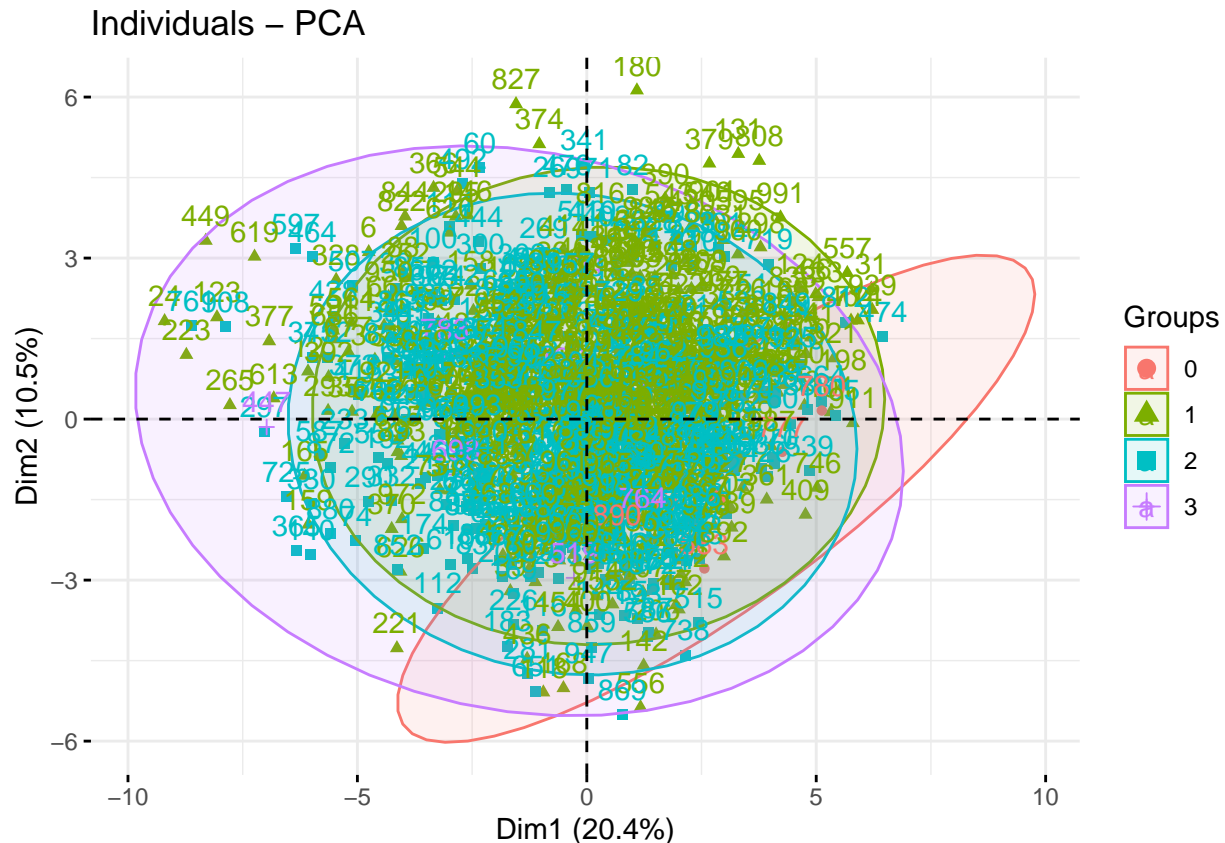
```
fviz_loadnings_with_cor(mod = pca_mod3, axes = 3)
```



A vizualizációt arra is használhatjuk, hogy csoportosítsuk a megfigyelesek a dimenziokon mutatott értékek alapján. Ezt az `addEllipses = T` parameterrel adhatjuk meg.

Hogyan jellemeznéd az egyes elipszisekben található embereket az alapján, hogy az 1. és 2. dimenzion milyen értékektől vesznek fel?

```
fviz_pca_ind(pca_mod3,
             label = "ind",
             habillage=factor(hsq$gender),
             addEllipses = T)
```



## 4 Bevezetés a feltárolt faktorelemzésbe (Exploratory Factor Analysis - EFA)

A faktorelemzés egy másik dimenzioredukciós technika, ami hasonlít a főkomponenselemzéshez. A kettő között fontos különbség, hogy a faktorelemzést akkor használjuk, ha feltételezzük, hogy a sok változónk háttérben közös okok, úgynevezett latens faktorok állnak, és ez okozza, hogy a megfigyelt változók korrelálnak egymással.

Amikor egy feltárolt faktorelemzési (EFA) modellt építünk, nem próbáljuk megmagyarázni a teljes varianciát az adatokban, mert megengedjük, hogy a latens faktorok csak részben magyarázzák a megfigyelt változók varianciáját. **A fennmaradó varianciát vagy merési hibát, vagy olyan faktorok befolyásolják, amelyek egyediak a megfigyelt változóra.** Ezért az EFA-ban minden egyes megfigyelt változóra tartozik egy "kommunalitás" (communality) érték. Ez az érték azt mutatja meg, hogy **az adott változóban megfigyelhető variancia mekkora hányadát magyarázzák a latens faktorok.** A fennmaradó varianciát a változóra egyedi faktor vagy merési hiba magyarázza (ezt egyediségnek, vagy uniqueness-nek is nevezzük).

A faktorelemzés legfontosabb lépései:

- Faktoralhatóság ellenőrzése
- Faktorkinyerés
- Ideális faktorszám kiválasztása
- Faktorforgatás
- Faktorok értelmezése



## 4.1 Adatok faktorálhatósága

Az adatfaktorálhatóság tesztelesekor azt a kérdést válaszoljuk meg, hogy van-e elegendő együttjárás (korreláció) a megfigyelhető változók között, ami lehetővé teszi az EFA elvégzését. Ennek tesztelésére két módszert is alkalmazunk: a Bartlett sphericity tesztet és a Kaiser-Meyer-Olkin tesztet.

Mindenek előtt azonban **az adatok korrelációs matrixára van szükségünk**, amin ezeket a teszteket lefuttathatjuk. Ezt megkaphatnánk a `cor()` funkcióval ha folytonos változokkal dolgoznánk, de ebben az adatbázisban **ordinalis adatokkal van dolgunk**, így egy másik funkciót használunk aminek a neve **`mixedCor()` a `psych` package-ból**. Ez a funkció képes az ordinalis adatok esetén használatos **“Polychoric Correlation”** meghatározására. A `mixedCor()` funkcióban meghatározzuk hogy melyek a folytonos változók, és melyek az ordinalis változók. A Q1-Q32 mind ordinalis, ezért csak a  $p = 1:32$ -t határozzuk meg, a  $c=t$  pedig NULL-ra állítjuk, mert nincs folytonos skalan mozgó (continuous) változó.

Fontos, hogy a korrelációs matrixot a `mixedCor()` **a `$rho` komponenseben tárolja**, ezért ezt kell elmentenünk egy új adatobjektumba. Mentsük el ezt a `hsq_correl` nevű objektumba.

```
hsq_mixedCor <- mixedCor(hsq, c=NULL, p=1:32)
hsq_correl = hsq_mixedCor$rho
```

---

### Gyakorlás

---

A fentebb tanultak alapján vizualizáld a változók közötti korrelációt. Használj több módszert is, pl. `ggcorr()`, `ggcorrplot()` `hc.order=TRUE`-val kombinálva, vagy `network_plot()`.

---

#### Bartlett sphericity teszt

A Bartlett teszt lenyelve hogy a **valós korrelációs matrixot** összehasonlítjuk egy **hipotetikus null-korrelációs matrix-al**, amiben minden korreláció 0 értéket vesz fel (identity matrix). A null hipotézis amit itt tesztelünk az, hogy a két korrelációs matrix nem különbözik egymástól. Ha a teszt szignifikans, az azt jelenti hogy **az adattábla változói korrelálnak egymással**.

Azonban fontos megjegyezni, hogy a Bartlett tesztnek van egy hatulutoje, megpedig hogy **nagy elemszámoknál szinte biztosan szignifikans** eredményt ad. Csak olyankor érdemes erre a mutatóra hagyatkozni a faktorálhatóság megállapításakor amikor a megfigyelek számát és a megfigyelt változók számának **aránya kisebb mint 5**. A mi esetünkben ez az arány  $993/32 = 31.03$ , vagyis a Bartlett teszt eredménye nem megbízható.

```
bfi_factorability <- cortest.bartlett(hsq_correl)
```

```
## Warning in cortest.bartlett(hsq_correl): n not specified, 100 used
```

```
bfi_factorability
```

```
## $chisq
## [1] 1280.585
##
## $p.value
## [1] 9.770759e-71
##
## $df
## [1] 496
```

#### Kaiser-Meyer-Olkin (KMO) teszt

A KMO teszt a **parciális korrelációs matrixot** hasonlítja össze a szokásos korrelációs matrixal. A parciális korreláció során meghatározzuk hogy **mekkora két változó közötti korreláció, ha kivonjuk a többi változó hatását a korrelációból**. A KMO érték azt mutatja, hogy mekkora a különbség a parciális korrelációk és a szokásos korrelációk között. A KMO egy különbség érték, a parciális korrelációk és a szokásos

korrelációk közötti különbséget jelzi. Azokban az esetekben ahol a változók sok közös varianciát hordoznak (vagyis valószínű hogy mögöttük egy közös latens faktor áll), a parciális korrelációk alacsonyak, vagyis a KMO index magas. A KMO tesztben az 1-hez közeli értékek jók jó faktorizálhatóságot mutatnak. A KMO index-nek legalább 0.6-nak kell lennie hogy úgy ítéljük hogy a változók faktorizálhatóak.

A mi példánkban a **KMO minden változó esetén magasabb 0.6-nal, és az összesített KMO is magasabb 0.6-nal**, így faktorizálhatónak tekinthetők a változók.

```
KMO(hsq_correl)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = hsq_correl)
## Overall MSA = 0.88
## MSA for each item =
##   Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  Q11  Q12  Q13  Q14  Q15  Q16
## 0.94 0.93 0.91 0.90 0.91 0.86 0.83 0.85 0.95 0.86 0.79 0.90 0.87 0.92 0.83 0.88
##   Q17  Q18  Q19  Q20  Q21  Q22  Q23  Q24  Q25  Q26  Q27  Q28  Q29  Q30  Q31  Q32
## 0.90 0.82 0.90 0.83 0.89 0.87 0.83 0.83 0.86 0.90 0.84 0.94 0.88 0.79 0.82 0.92
```

## 4.2 Faktorextrakció

A faktorokat az `fa()` funkcióval fogjuk kinyerni. Ez a funkció több faktorextrakciós módszert is kínál. A leggyakrabban használt módszer a **Maximum Likelihood Estimation** (mle) akkor ha a megfigyelt változók megfelelnek a többváltozós normalitás feltételének, míg a **Principal Axis Factoring** (paf) a preferált módszer akkor, ha a változók nem mutatnak többváltozós normalis eloszlást.

Az `mvn()` funkció az MVN package-ból és a `mvnrm.kur.test()` és a `mvnrm.skew.test()` funkciók az ICS package-ból segíthet eldönteni, hogy többváltozós normalis eloszlást mutatnak-e az adatok. Ha ezeknek a teszteknek a **p-értéke alacsonyabb 0.05-nel, akkor az a többváltozós normalitás sérülése utal.**

```
result <- mvn(hsq[,1:32], mvnTest = "hz")
result$multivariateNormality
```

```
##           Test           HZ p value MVN
## 1 Henze-Zirkler 1.001426         0 NO
```

```
mvnrm.kur.test(na.omit(hsq[,1:32]))
```

```
## Warning in pchisqsum(n * W.stat, df = dfs, a = chi.fac, method = "integration"):
## Package 'CompQuadForm' not found, using saddlepoint approximation
```

```
##
## Multivariate Normality Test Based on Kurtosis
##
## data: na.omit(hsq[, 1:32])
## W = 1707.3, w1 = 0.12457, df1 = 527.00000, w2 = 0.23529, df2 = 1.00000,
## p-value < 2.2e-16
```

```
mvnrm.skew.test(na.omit(hsq[,1:32]))
```

```
##
## Multivariate Normality Test Based on Skewness
##
## data: na.omit(hsq[, 1:32])
## U = 441.69, df = 32, p-value < 2.2e-16
```

Fent látható hogy mind a Henze-Zirkler teszt mind a többváltozós ferdeség és csúcsosság tesztek a normalitás feltételének sérülése utal. Így a **paf extrakciós módszert** használjuk majd.

A faktorextrakcióra a psych package `fa()` funkcióját használjuk. Ezen belül megadhatjuk a faktorextrakciós módszert az `fm` = parameteren belül. Itt `fm` = "pa"-t határozzuk meg, mert a paf módszert szeretnénk használni, de ha a többváltozós normalitás nem sérült volna, akkor ehelyett "mls"-t használtunk volna. Az alábbi példában meg nem akartam faktorforgatást alkalmazni, hogy lépésről lépésre tudjam bemutatni a faktorelemzés módszerét, így a `rotate` = értéket "none"-ra állítottam, de általában a faktorokat egyből el is forgatjuk valamelyik módszerrel (lásd alább). Az `nfactors` = parameterrel adhatjuk meg, hány faktort szeretnénk kinyerni. Egyelőre állítsuk ezt 5-re, lentebb tárgyaljuk majd, hogyan választjuk ki az ideális faktormennyiséget.

A modell objektum `$communality` komponenseben találjuk a változokhoz tartozó kommunalitás értékeket. Ezt legmagasabbtól legalacsonyabbig sorbarendezzük és kilistázzuk. Ahogy fentebb említettük a kommunalitás azt jelzi, hogy az egyes megfigyelt változokban tapasztalható variancia mekkora hanyadát magyarázzak a kinyert faktorok. Az output azt mutatja, hogy a Q17 "Általában nem szeretek viccelődni, vagy másokat szorakoztatni" ("I usually don't like to tell jokes or amuse people.") a legjobban reprezentált item az 5 faktoros struktúrában, aminek 68%-át képesek megmagyarázni az új faktorok. Ezzel szemben a Q22 "Amikor szomorú vagy ideges vagyok általában elvesztem a humorérzetemet" ("If I am feeling sad or upset, I usually lose my sense of humor.") a legkevesbe reprezentált item, varianciájának csak 25%-át magyarázza a jelenlegi faktorstruktúra.

Neha ahhoz hogy a faktorstruktúra jól mukodjon, érdemes a rosszul reprezentált itemeket kizárni. Ez főleg akkor fontos, ha kicsi a mintaelemszám. **Ha a megfigyelesek száma 250 alatti, akkor MacCallum et al. szerint elvárható hogy az itemek átlagos kommunalitása legalább 0.6 legyen.** A mi esetünkben ennél megengedőbbek is lehetünk, mert az elemszámunk nagyobb, de egy mélyebb faktorelemzés esetén így is érdemes lehet elgondolkodni a rosszul reprezentált itemek kizárában.

MacCallum, R. C., Widaman, K. F., Zhang, S., & Hong, S. (1999). Sample size in factor analysis. *Psychological methods*, 4(1), 84.

```
EFA_mod1 <- fa(hsq_correl, nfactors = 5, fm="pa")
```

```
# Sorted communality
```

```
EFA_mod1_common <- as.data.frame(sort(EFA_mod1$communality, decreasing = TRUE))
```

```
EFA_mod1_common
```

```
##      sort(EFA_mod1$communality, decreasing = TRUE)
## Q17                                0.6754689
## Q20                                0.6701555
## Q25                                0.6528782
## Q8                                  0.6265007
## Q21                                0.6240345
## Q10                                0.6140024
## Q15                                0.6047847
## Q18                                0.5909926
## Q14                                0.5796418
## Q13                                0.5557827
## Q32                                0.5458051
## Q26                                0.5424384
## Q1                                  0.5311824
## Q31                                0.5176279
## Q12                                0.4858949
## Q19                                0.4806134
## Q5                                  0.4741704
## Q4                                  0.4663425
## Q2                                  0.4442162
## Q16                                0.4360995
## Q6                                  0.4257846
```

```
## Q29 0.4025312
## Q7 0.3978300
## Q11 0.3930955
## Q3 0.3908073
## Q24 0.3558342
## Q23 0.3419339
## Q27 0.3300415
## Q9 0.3241293
## Q30 0.2867962
## Q28 0.2663399
## Q22 0.2543910
```

```
mean(EFA_mod1$communality)
```

```
## [1] 0.4777546
```

### 4.3 Ideális faktorszám kiválasztása

A fokkomponenselemzéshez hasonlóan meg kell határoznunk, hány faktort szeretnénk kinyerni az adatokból. Ahogy azt a fokkomponenselemzésnél is láttuk, ennek az eldöntésére használhatjuk a scree-tesztet, a Kaiser-Guttman kritériumot, vagy a parallel tesztet. Ezen felül a psych pacakge két újabb módszert is felkínál a döntéshozás elősegítésére: a very simple structure (VSS) kritériumot, és a Wayne Velicer's Minimum Average Partial (MAP) kritériumot. (A vss() funkció a psych package-ben)

Az alábbi példában a psych package fa.parallel függőjét és az nfactors függőjét használjuk arra, hogy a különböző kritériumok szerint eldönthessük, hány faktor megtartása lenne ideális.

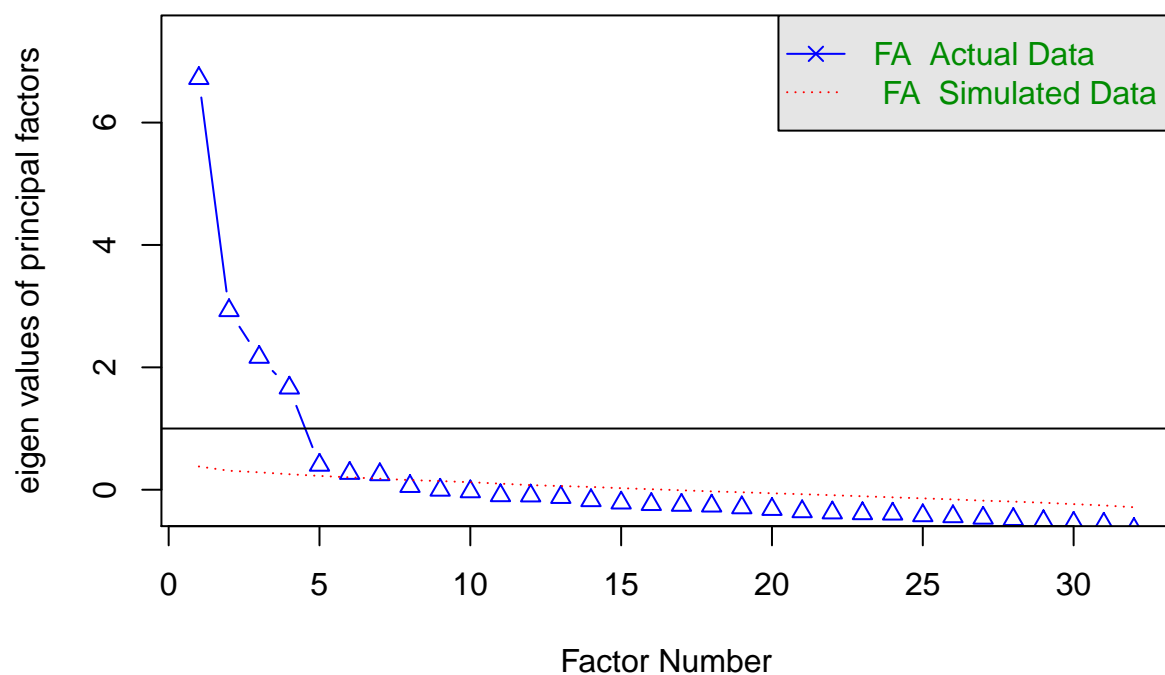
A különböző technikák által javasolt ideális faktorszámok a következők:

- scree-tesztet: 4
- Kaiser-Guttman kritérium: 4
- Parallel tesztet: 7
- VSS: 3-4
- MAP: 4

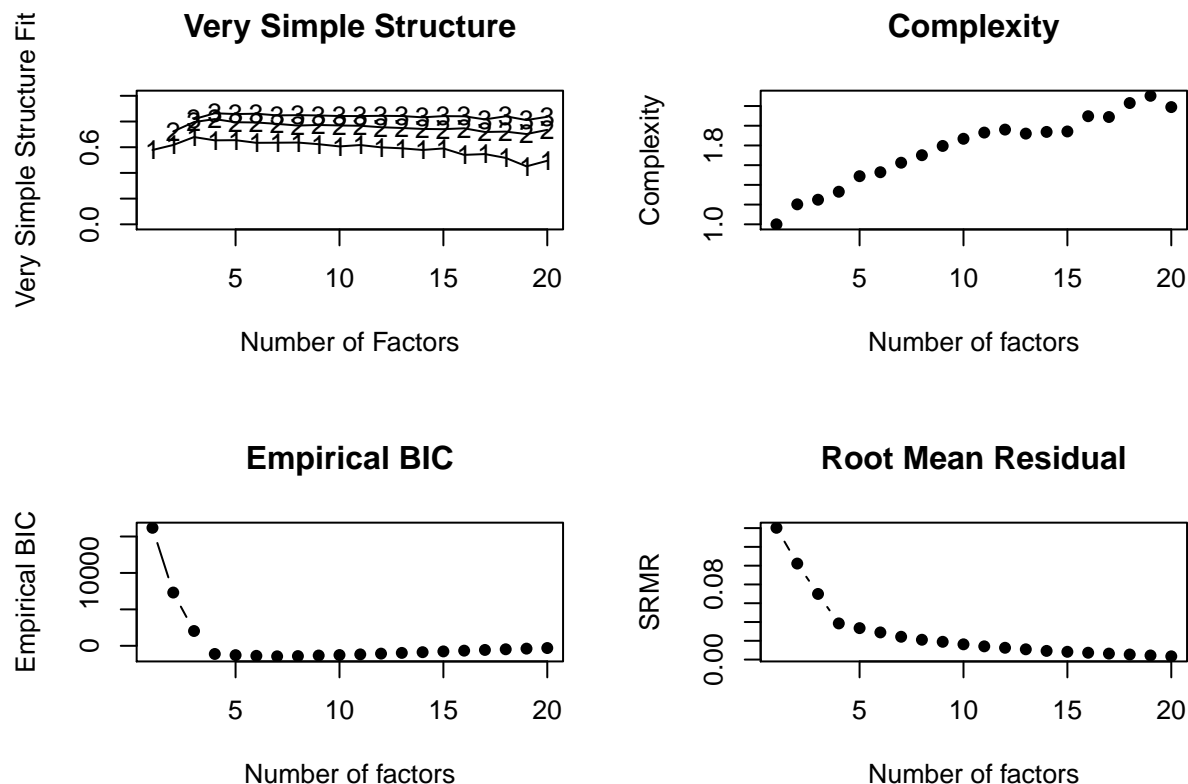
Ezek alapján úgy tűnik, a legtöbb technika szerint 4 latens faktor írja le az adatok variabilitását a legjobban. Alább még is építjük ezt a 4-faktoros modellt, és megvizsgáljuk a kommunalitás-táblázatot. A faktorelemzés során nagyon gyakori, hogy a folyamatot újra és újra megismételjük különböző bemeneti változokkal és különböző faktorszámokkal és rotációs módszerekkel, amíg elerjük a véglegesnek tekinthető faktorstruktúrát. A végleges faktorstruktúra ideális esetben jól értelmezhető a faktorok és a hozzájuk tartozó változó-töltések alapján.

```
fa.parallel(hsq_correl, n.obs = nrow(hsq),
            fa = "fa", fm = "pa")
```

## Parallel Analysis Scree Plots



```
## Parallel analysis suggests that the number of factors = 7 and the number of components = NA
n factors(hsq_correl, n.obs = nrow(hsq))
```



```
##
## Number of factors
## Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
##       n.obs = n.obs, plot = FALSE, title = title, use = use, cor = cor)
## VSS complexity 1 achieves a maximum of 0.68 with 3 factors
## VSS complexity 2 achieves a maximum of 0.82 with 4 factors
## The Velicer MAP achieves a minimum of 0.01 with 4 factors
## Empirical BIC achieves a minimum of -1440.89 with 7 factors
## Sample Size adjusted BIC achieves a minimum of -204.78 with 14 factors
##
## Statistics by number of factors
```

	vss1	vss2	map	dof	chisq	prob	sqresid	fit	RMSEA	BIC	SABIC	complex
## 1	0.58	0.00	0.034	464	8804	0.0e+00	40.0	0.58	0.1345	5602	7075	1.0
## 2	0.62	0.72	0.027	433	6069	0.0e+00	26.3	0.72	0.1145	3081	4457	1.2
## 3	0.68	0.80	0.019	403	4167	0.0e+00	17.2	0.82	0.0970	1386	2666	1.2
## 4	0.65	0.82	0.011	374	2283	1.5e-270	11.5	0.88	0.0717	-298	890	1.3
## 5	0.65	0.79	0.012	346	1980	9.3e-227	10.3	0.89	0.0690	-407	692	1.5
## 6	0.63	0.79	0.013	319	1632	8.3e-175	9.3	0.90	0.0644	-570	443	1.5
## 7	0.63	0.78	0.014	293	1297	3.6e-126	8.4	0.91	0.0587	-725	206	1.6
## 8	0.64	0.77	0.016	268	1099	4.6e-101	7.8	0.92	0.0559	-750	101	1.7
## 9	0.62	0.77	0.017	244	946	3.2e-83	7.3	0.92	0.0538	-738	37	1.8
## 10	0.61	0.77	0.020	221	708	2.5e-52	6.9	0.93	0.0471	-817	-115	1.9
## 11	0.62	0.77	0.022	199	584	1.5e-39	6.4	0.93	0.0441	-789	-157	1.9
## 12	0.60	0.76	0.025	178	500	2.7e-32	6.1	0.94	0.0426	-729	-163	2.0
## 13	0.59	0.75	0.028	158	399	7.8e-23	5.8	0.94	0.0392	-691	-189	1.9
## 14	0.58	0.74	0.032	139	313	2.0e-15	5.5	0.94	0.0355	-646	-205	1.9

```
## 15 0.59 0.74 0.036 121 269 3.4e-13 5.2 0.94 0.0350 -566 -182 1.9
## 16 0.54 0.75 0.040 104 223 1.3e-10 4.9 0.95 0.0339 -495 -165 2.1
## 17 0.55 0.72 0.046 88 170 4.1e-07 4.8 0.95 0.0305 -438 -158 2.1
## 18 0.52 0.72 0.052 73 129 6.4e-05 4.5 0.95 0.0277 -375 -143 2.2
## 19 0.45 0.70 0.058 59 73 1.0e-01 4.4 0.95 0.0155 -334 -147 2.3
## 20 0.49 0.74 0.066 46 49 3.6e-01 4.2 0.96 0.0076 -269 -123 2.2
## eChisq SRMR eCRMS eBIC
## 1 19394 0.1403 0.145 16192
## 2 10295 0.1022 0.109 7307
## 3 4812 0.0699 0.078 2031
## 4 1460 0.0385 0.044 -1120
## 5 1105 0.0335 0.040 -1282
## 6 824 0.0289 0.036 -1377
## 7 581 0.0243 0.032 -1441
## 8 439 0.0211 0.029 -1410
## 9 353 0.0189 0.027 -1331
## 10 260 0.0162 0.024 -1265
## 11 196 0.0141 0.022 -1177
## 12 156 0.0126 0.021 -1073
## 13 118 0.0109 0.019 -972
## 14 84 0.0092 0.017 -875
## 15 68 0.0083 0.017 -767
## 16 52 0.0073 0.016 -665
## 17 40 0.0064 0.015 -567
## 18 29 0.0054 0.014 -475
## 19 19 0.0044 0.013 -388
## 20 12 0.0035 0.011 -305
```

```
EFA_mod2 <- fa(hsq_correl, nfactors = 4, fm="pa")
```

```
EFA_mod2_common <- as.data.frame(sort(EFA_mod2$communality, decreasing = TRUE))
EFA_mod2_common
```

```
## sort(EFA_mod2$communality, decreasing = TRUE)
## Q20 0.6700371
## Q17 0.6697004
## Q25 0.6530751
## Q8 0.6301917
## Q21 0.6195670
## Q10 0.6145714
## Q18 0.5907715
## Q14 0.5678837
## Q32 0.5457790
## Q26 0.5449964
## Q13 0.5427269
## Q1 0.5316347
## Q31 0.5224554
## Q15 0.5181019
## Q12 0.4849208
## Q5 0.4602084
## Q4 0.4523891
## Q2 0.4241464
## Q29 0.4033447
## Q7 0.3973521
## Q3 0.3950130
```

```
## Q16 0.3775542
## Q19 0.3656149
## Q6 0.3630614
## Q11 0.3313680
## Q27 0.3311449
## Q9 0.3143381
## Q24 0.2969414
## Q23 0.2792367
## Q30 0.2700595
## Q28 0.2369152
## Q22 0.1856682
```

```
mean(EFA_mod2$communality)
```

```
## [1] 0.4559615
```

## 4.4 Faktorforgatas

A faktorforgatas celja hogy megkonnyitse a faktorok értelmezését. Így elkerülhető hogy az egész faktorstruktúra 1 vagy két nagyon domináns faktorból álljon, amire angyon erősek a töltések, míg a többi faktor értelmezése kódos. A faktorforgatas során az eredeti változók ugyan ott maradnak a “faktorterben”, viszont a faktorok dimenzio tengelyeit elforgatjuk, hogy jobban railleszkedjenek egyes változócsoportokra.

A faktorforgatasnak számos módszere ismert, de ezek két fő csoportba sorolhatók: ortogonális és oblique módszerek közé. Az ortogonális módszerek (mint pl. Quartimax, Equimax, vagy a pszichológiában leggyakrabban használt **Varimax** módszer) során a faktor dimenziók egymásra merőlegesek maradnak (ez azt jelenti hogy egymással nem korrelálnak majd a végso faktorok). Az oblique módszerek (mint pl. **Direct Oblimin** vagy a Promax) esetén viszont megengedett hogy a végso faktorok valamelyest korreláljanak egymással. Az exploratoros faktorelemzés során több módszert is kipróbálhatunk, de itt fontos az elméleti megalapozottság is. Elképzeltető hogy a faktorok korreláljanak egymással? Ha igen, akkor az oblique módszerekre érdemes hagyatkozni. (Általában a korrelálatlan faktorokat könnyebb értelmezni).

Az alapértelmezett faktorforgatási módszer a Direct Oblimin (“oblimin”). Próbáljuk ki a Promax (“promax”) és a Varimax (“varimax”) módszereket is.

```
EFA_mod2$rotation
```

```
## [1] "oblimin"
```

```
EFA_mod_promax <- fa(hsq_correl, nfactors = 4, fm="pa", rotate = "promax")
```

```
EFA_mod_varimax <- fa(hsq_correl, nfactors = 4, fm="pa", rotate = "varimax")
```

## 4.5 Faktorok interpretacioja

A faktorok értelmezése nem könnyű feladat. Sok területspecifikus tudásra van szükség a helyes faktórsztruktúra kiválasztásához és a helyes faktorértelmezéshez. Itt ezért csak a különböző vizualizációs módszereket mutatjuk be amik segíthetnek a faktorok értelmezésében.

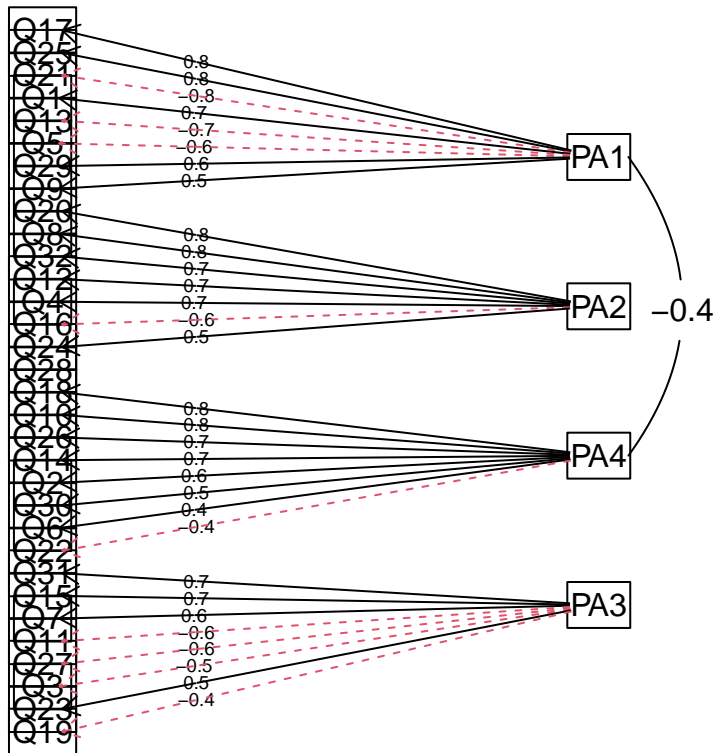
Az `fa.diagram()` funkció kirajzolja a model objektum alapján a faktorstruktúrát, és azt, hogy melyik változó melyik faktorra mutatja a legnagyobb faktortöltést (melyik faktossal a legnagyobb korrelációja). Az ábrán láthatóak az egyes korrelációs együtthatók is. A fekete nyilak pozitív, míg a piros nyilak negatív korrelációkat jeleznek.

További segítséget nyújthat a saját funkció amit a főkomponens-elemzésnél is használtunk: `fviz_loadings_with_cor()`. Itt a `fa()` modellek esetén megadhatjuk a `loading_above =` paramétert is, ahol specifikálhatjuk, hogy csak a bizonyos abszolút faktortöltés (korreláció) feletti megfigyelt változókat ábrázoljuk. Ez megkönnyíti az ábra átlathatóságát.



```
fa.diagram(EFA_mod2)
```

## Factor Analysis

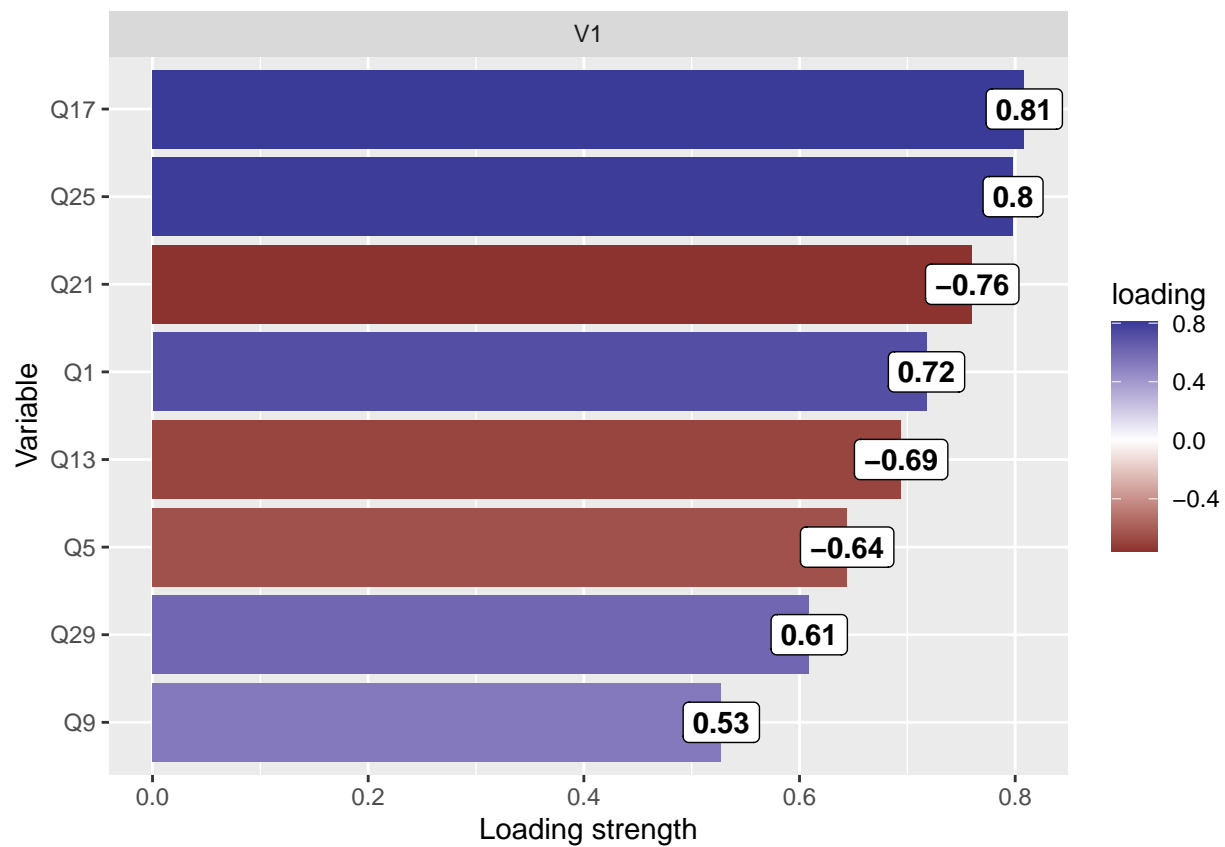


```
fviz_loadings_with_cor(EFA_mod2, axes = 1, loadings_above = 0.4)
```

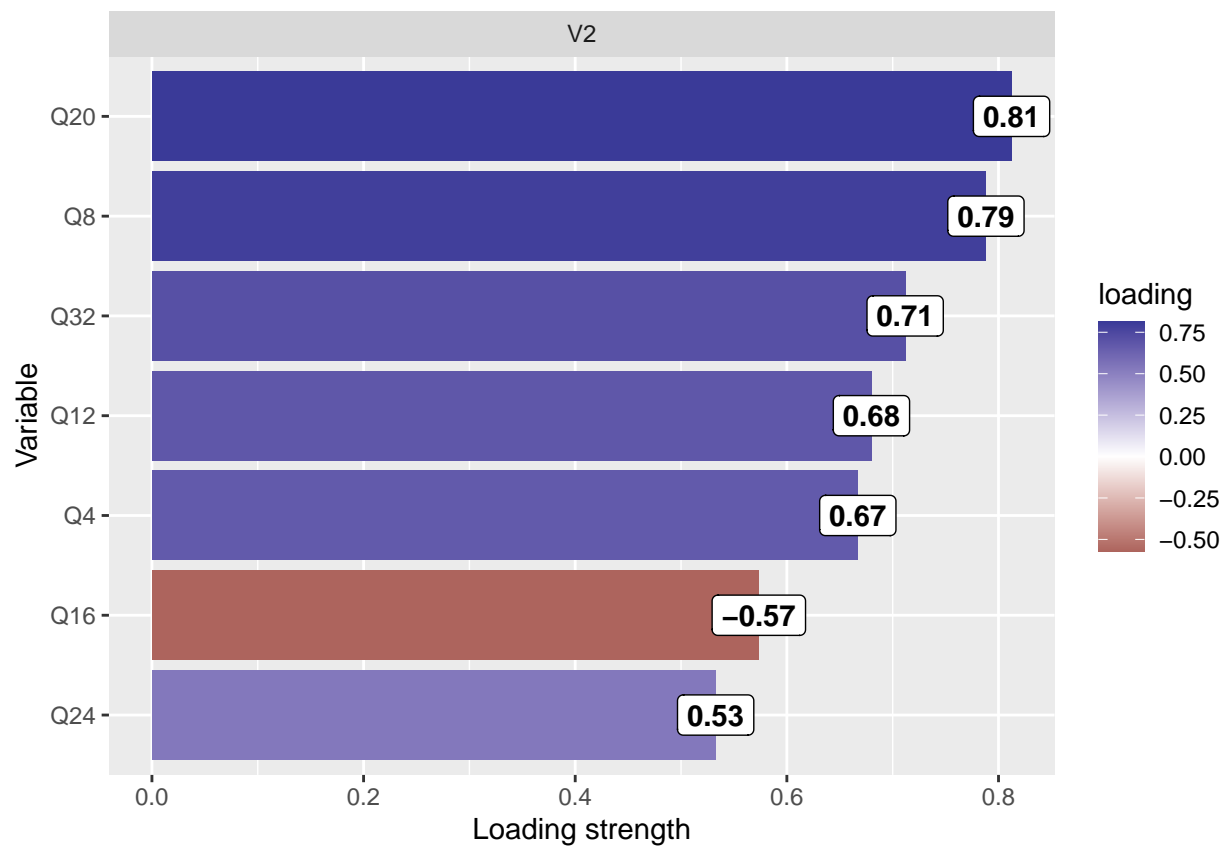
```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if `.name_repair` is
## Using compatibility `.name_repair`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.

## Warning: The `i` argument of `[<-`()' can't be a matrix as of tibble 3.0.0.
## Convert to a vector.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.

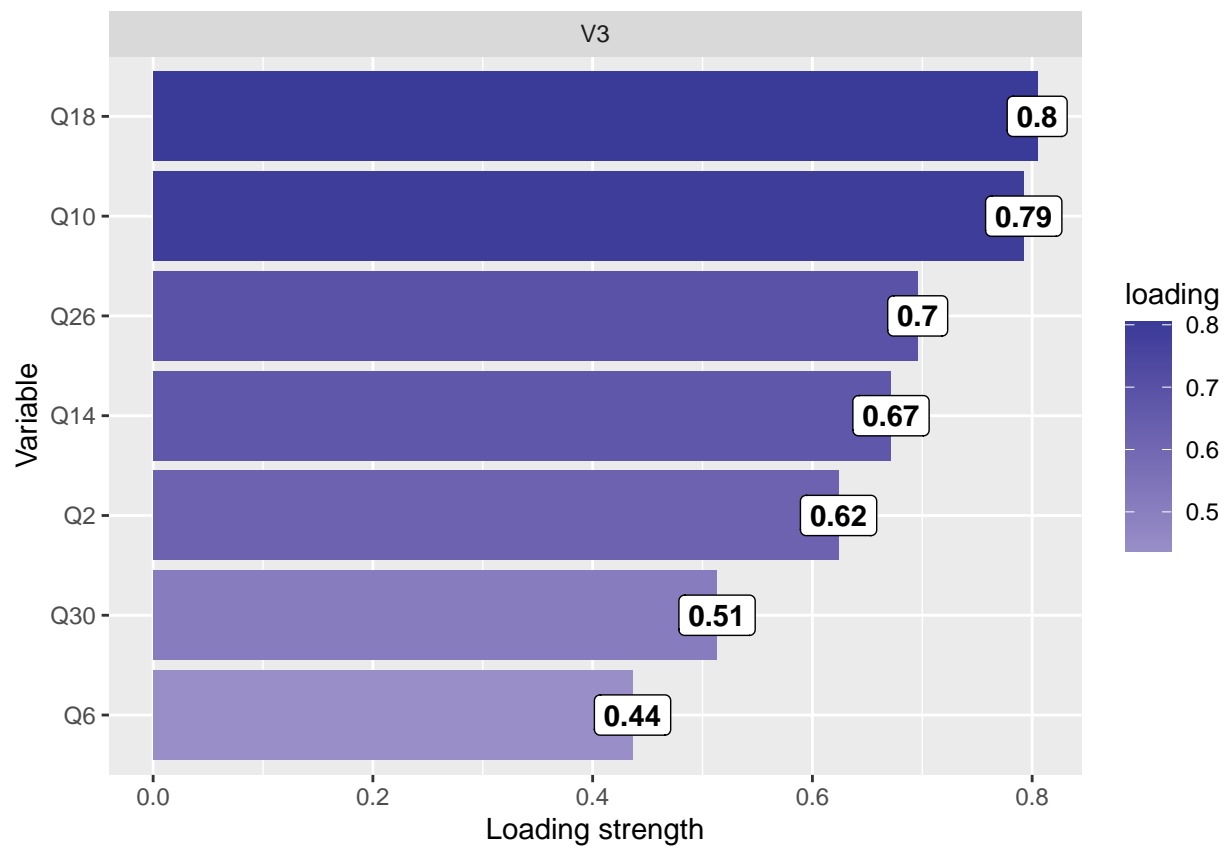
## Warning: The `i` argument of `[`()' can't be a matrix as of tibble 3.0.0.
## Convert to a vector.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```



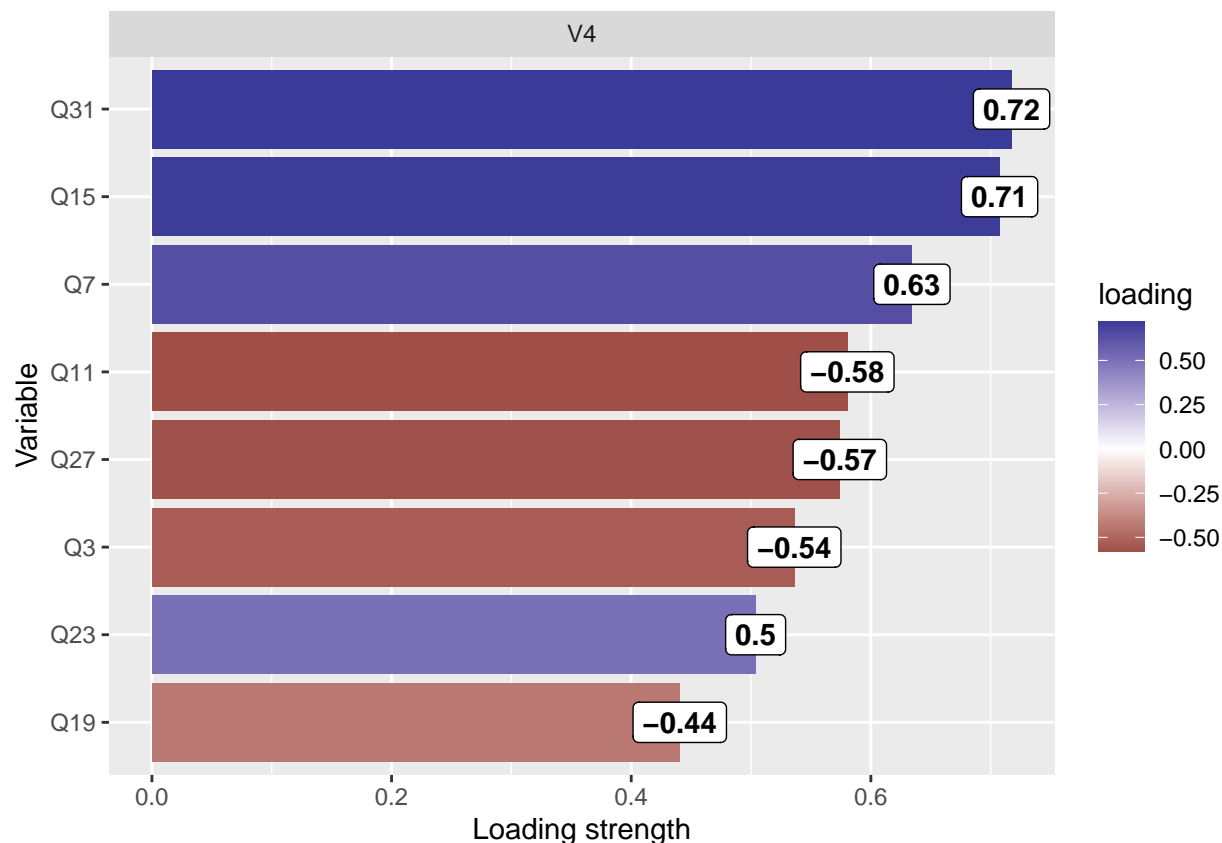
```
fviz_loadnings_with_cor(EFA_mod2, axes = 2, loadings_above = 0.4)
```



```
fviz_loadnings_with_cor(EFA_mod2, axes = 3, loadings_above = 0.4)
```



```
fviz_loadnings_with_cor(EFA_mod2, axes = 4, loadings_above = 0.4)
```



### Gyakorlas

A fent tanult technikákat a Big Five Inventory (bfi) adatbazison gyakorolhatod. Ez a psych package-be beépített adatbázis, ami 2800 személy válaszait tartalmazza a Big Five személyiségkerdoiv kérdéseire. Az első 25 oszlop a kerdoiv kérdéseire adott válaszokat tartalmazza, az utolsó három oszlop (gender, education, és age) pedig demográfiai kérdéseket tartalmaz. A részleteket az egyes itemekhez tartozó kérdésekről és a válaszok kódolásáról elolvashatod ha lefuttatod a `?bfi` parancsot.

Az adatbázist betöltheted a következő parancsokkal.

```
?bfi
```

```
data(bfi)
my_data_bfi = bfi[,1:25]
```

Ebben a feladatban csak az első 25 oszlopot használd, az eredeti kerdoiv kérdéseit. Végezz el feltárási faktorelemzést, és ez alapján határozd meg, hány faktor megtartása az ideális, mely faktorokra mely itemek töltenek leginkább, és ez alapján hogyan nevezned el a faktorokat. Melyek a faktorstruktúra által leginkább és a legkevesbe reprezentált itemek?