Fokomponenselemzes es exploratoros faktorelemzes

Zoltan Kekecs

29 April 2020

Contents

1 Absztrakt

Ebben a gyakorlatban a “dimenzionalitas atkaval” birkozunk meg. Ezt a valtozok szamanak csokkentesevel oldjuk meg a fokomponenselemzes es az exploratoros faktorelemzes segitsegevel.

Ez a gyakorlat nagy mertekben a DataCamp Dimensionality Reduction in R kurzusa alapjan lett osszeallitva. https://www.datacamp.com/courses/dimensionality-reduction-in-r

2 Adat kezeles es leiro statisztikak

2.1 Csomagok betoltese

Ennek a gyakorlatnak a soran az alabbi csomagokat fogjuk hasznalni:

```
library(tidyverse)  # for tidy code
library(GGally)  # for ggcorr
library(corrr)  # 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(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
```

2.2 Sajat funkciok betoltese

```
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", axes], contrib_and_cov[contrib_and_cov[,
                "attribute_type"] == "correlation", axes])),
                vars)
            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, 2)) + 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 cars beepitett adatbazis betoltese

A cars adatbazist fogjuk hasznalni, ami egy beepitett adatbazis az R-ben. Az adatbazis 32 kulonbozo automodellrol tartalmaz informaciokat az Motor Trend magazine 1974-es szamabol. Minden autonak 11 karakkterisztikajat tartalmazza az adattabla. Ezek a kovetkezok:

- mpg: uzemanyaghatekonysag: hany merfoldet tudunk megtenni az autoval 1 gallon benzinnel
- cyl: hengerek szama
- disp: hengerterfogat
- hp: loero
- drat: Rear axle ratio, ennek az uzemanyaghatekonysaghoz van koze, minel magasabb ez a szam annal rosszabb az uzemanyaghatekonysag
- wt: suly
- qsec: hany masodperc alatt tesz meg 1/4 merfoldet az auto. a gyorsulassal es sebesseggel fugg ossze
- vs: kategorikus valtozo: a motorblokk alakjarol szol: egyenes alaku (0), V alaku (1)
- am: kategorikus valtozo: automata valtos auto (0) kezi valtos auto (1).
- gear: sebessegek szama
- carb: karburatorok szama

```
data("mtcars")
```

2.4 Az adatsor megtekintese

Vizsgaljuk meg az adatok strukturajat es az alapveto leiro statisztikakat

```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
summary(mtcars)
```

```
##       mpg             cyl             disp             hp
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
##       drat             wt             qsec             vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
```

```
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

Tegyuk fel hogy autot szeretnenk vasarolni es szeretnenk a leheto legjobb dontest hozni azzal kapcsolatban hogy melyik autot valasszuk, de nem ertunk kulonosebben az autokhoz. Amikor megnezzuk a Motor Trend magazint, latjuk ezt a 11 karakterisztikat, de nincs idonk minden autot osszehasonlitani mind a 11 karakterisztika alapjan. Szeretnenk valahogy egyszerusiteni a helyzetet, es megtudni, mik a legfontosabb alapveto tulajdonsagain egy autonak ami alapjan a dontesunket meghozhatjuk.

Nezzuk meg az adatok korrelacios matrixat, hogy jobb kepet kapjunk arrol, mely valtozok fuggenek ossze egymassal es milyen szorosan.

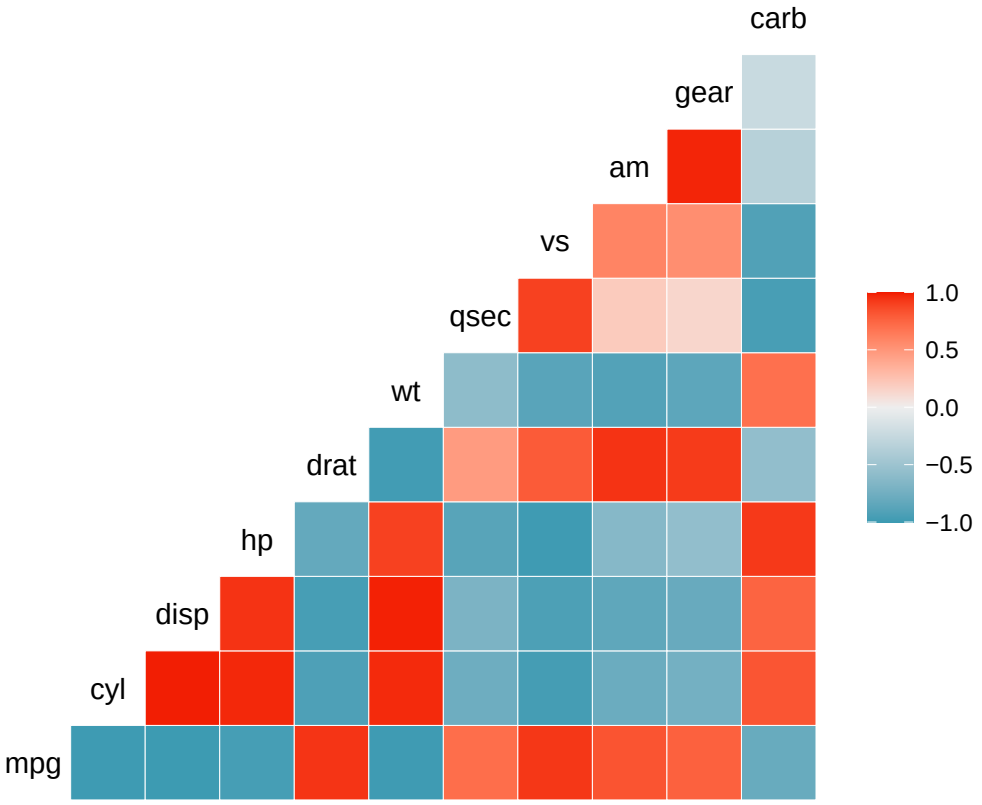
A korrelacios matrixot tobbfajta modon is vizualizalhatjuk. Az eredmeny alapjan lathato hogy vannak korrelacios "klaszterek", vagyis olyan valtozok amik csoportosan osszefuggenek egymassal, de a kep egyelore tul bonyolult, mert tul sok a valtozo.

Az alabbi peldaban hasznalt network_plot() funcio neha nem ranzolja ki a vonalakat. Sajnos ennek nem jottem ra a megoldasara. Ezt masok is tapasztaltak, itt lehet kovetni ezt a hibajelentest: https://github.com/tidymodels/corrr/issues/63

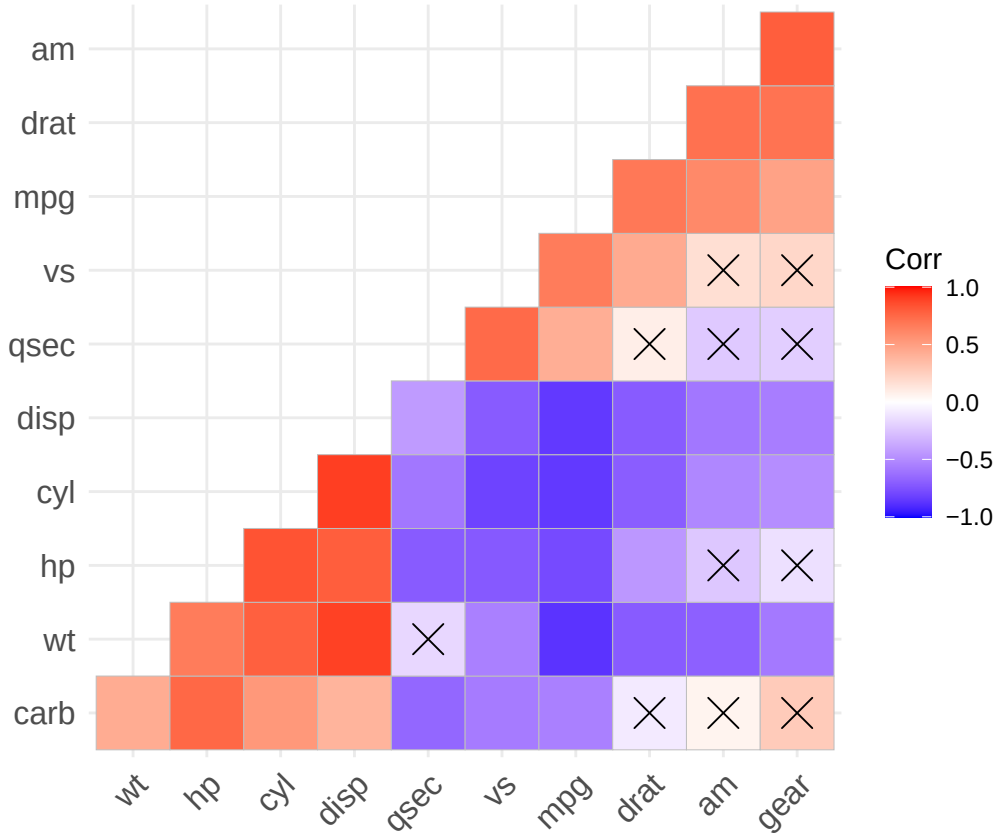
```
cor = mtcars %>% cor()
cor
```

```
##             mpg        cyl       disp         hp        drat         wt
## mpg   1.0000000 -0.8521620 -0.8475514 -0.7761684  0.68117191 -0.8676594
## cyl  -0.8521620  1.0000000  0.9020329  0.8324475 -0.69993811  0.7824958
## disp -0.8475514  0.9020329  1.0000000  0.7909486 -0.71021393  0.8879799
## hp   -0.7761684  0.8324475  0.7909486  1.0000000 -0.44875912  0.6587479
## drat  0.6811719 -0.6999381 -0.7102139 -0.4487591  1.00000000 -0.7124406
## wt   -0.8676594  0.7824958  0.8879799  0.6587479 -0.71244065  1.0000000
## qsec  0.4186840 -0.5912421 -0.4336979 -0.7082234  0.09120476 -0.1747159
## vs    0.6640389 -0.8108118 -0.7104159 -0.7230967  0.44027846 -0.5549157
## am    0.5998324 -0.5226070 -0.5912270 -0.2432043  0.71271113 -0.6924953
## gear  0.4802848 -0.4926866 -0.5555692 -0.1257043  0.69961013 -0.5832870
## carb -0.5509251  0.5269883  0.3949769  0.7498125 -0.09078980  0.4276059
##             qsec         vs          am       gear        carb
## mpg   0.41868403  0.6640389  0.59983243  0.4802848 -0.55092507
## cyl  -0.59124207 -0.8108118 -0.52260705 -0.4926866  0.52698829
## disp -0.43369788 -0.7104159 -0.59122704 -0.5555692  0.39497686
## hp   -0.70822339 -0.7230967 -0.24320426 -0.1257043  0.74981247
## drat  0.09120476  0.4402785  0.71271113  0.6996101 -0.09078980
## wt   -0.17471588 -0.5549157 -0.69249526 -0.5832870  0.42760594
## qsec  1.00000000  0.7445354 -0.22986086 -0.2126822 -0.65624923
## vs    0.74453544  1.0000000  0.16834512  0.2060233 -0.56960714
## am   -0.22986086  0.1683451  1.00000000  0.7940588  0.05753435
## gear -0.21268223  0.2060233  0.79405876  1.0000000  0.27407284
## carb -0.65624923 -0.5696071  0.05753435  0.2740728  1.00000000
```

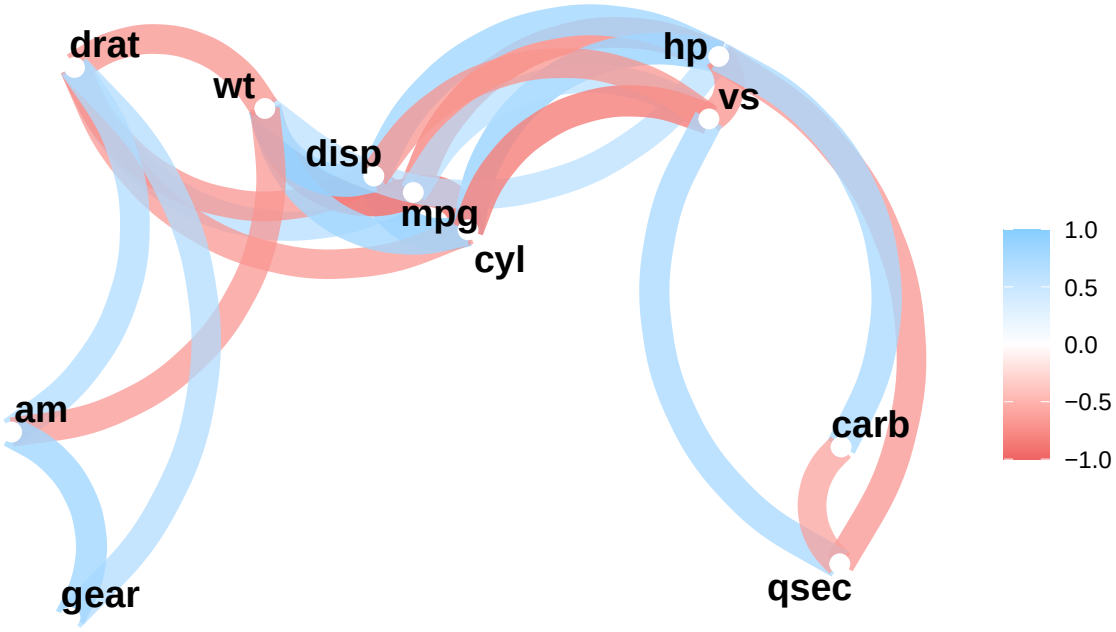
```
ggcorr(cor)
```




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




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



3 Fokomponenselemzes

3.1 A fokomponenselemzes modell megepitese

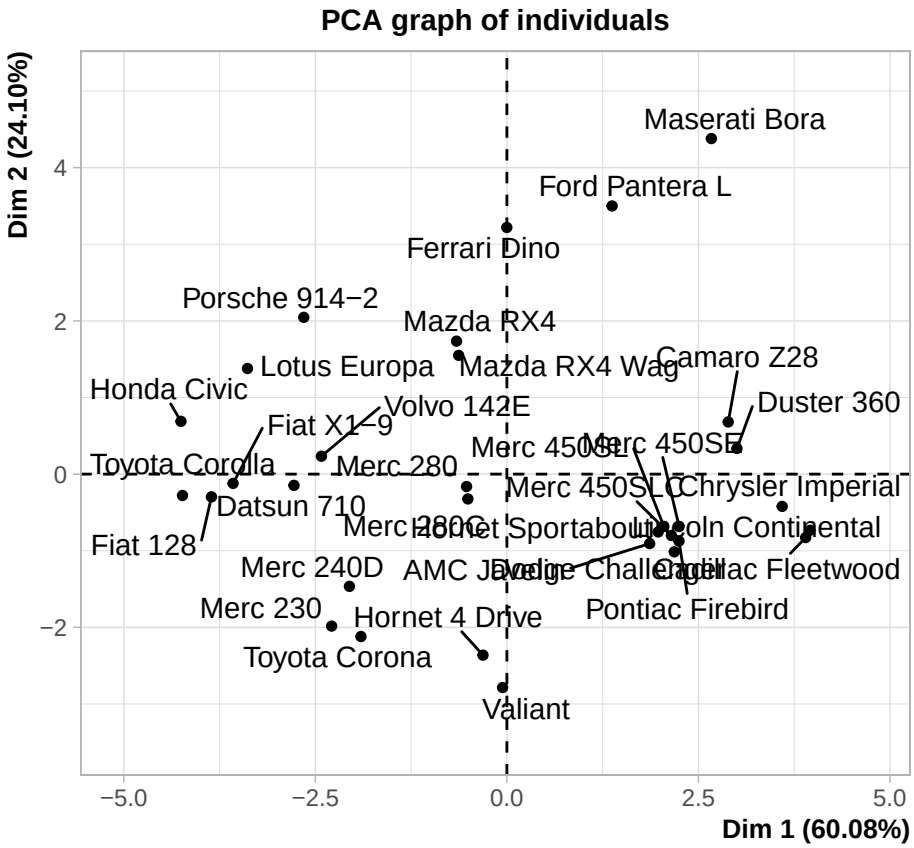
Nehany valtozo osszefuggeset konnyen atlathatova tehetjuk vizualizacion keresztul. Azonban ha sok valtozoval van dolgunk, a vizualizacio es egyeb korabban tanult feltaro adatelemzesi technikak kudarcot vallhatnak egyszeruen azert mert tul sok az informacio amit nehez atlatni es vizualizalni. Ebben a helyzetben segithet a fokomponenselemzes, amit arra hasznalunk hogy lecsokkentsuk a valtozok szamat amivel dolgoznunk kell, ugy, hogy kozben a leheto legtobb informaciot tartunk meg az adatok variabilitasarol.

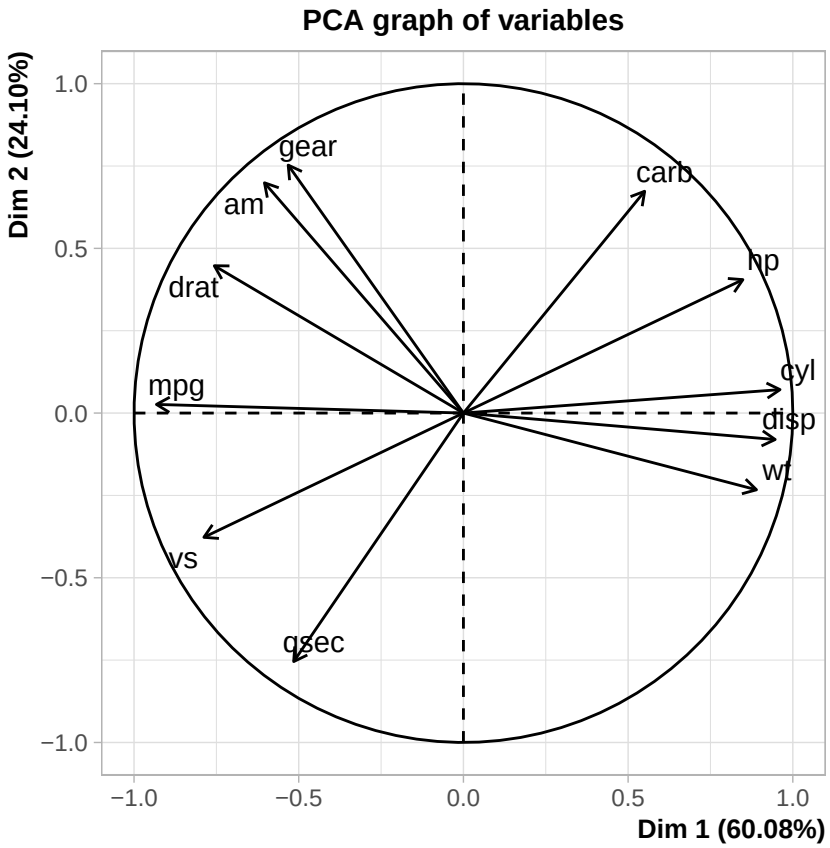
A fokomponenselemzest a PCA() (principal component analysis) funkcioval tudjuk elvegezni a FactoMineR package-bol. Az elemzes eredmenyet egy pca_mod modell objektumba mentettem el. A egybol kiad ket abrat a ket legfontosabb fokomponensrol (dimenziorol) amik a leghatekoonyabban irjak le az adatokat.

Az egyik abran az latszik, hogy az egyes megfigyelesek (ebben az esetben az egyes auto-modellek) hol helyezkednek el a ket dimenzio menten. A masodik abra pedig arrol szol, hogy a dimenziok milyen korrelaciot mutatnak az eredeti valtozokkal. A szaggatott vonalak mutatjak a fokomponenseket. A nyilak minel kozelebb fekszenek a szaggatott vonalhoz, a valtozo annal inkabb egyuttjar az adott dimenzioval a masik dimenzioval szemben.

Peladaul a cyl es az mpg valtozokat sokkal jobban leirja a Dim1 mint a Dim2. (a nyil iranya alapjan megallapithato hogy az mpg negativan, a cyl pozitivan korrelal a Dim1-el.) Ezzel szemben az ?carb valtozok nyila a ket dimenzio kozott helyezkedik el, ami azt jelenti hogy vagy midkettovel nagyjabol azonos mertekben korrelalnak (ez lehet nagyon kicsi, vagy akar nagyon nagy korrelacio is).

```
pca_mod <- PCA(mtcars)
```


PCA graph of individuals
Dim 2 (24.10%)
Dim 1 (60.08%)
Maserati Bora
Ford Pantera L
Ferrari Dino
Porsche 914−2
Mazda RX4
Mazda RX4 Wag
Lotus Europa
Camaro Z28
Honda Civic
Volvo 142E
Duster 360
Fiat X1−9
Merc 450SL
Merc 450SE
Toyota Corolla
Merc 280
Merc 450SLC
Chrysler Imperial
Datsun 710
Merc 280C
Hornet Sportabout
Lincoln Continental
Fiat 128
Merc 240D
AMC Javelin
Dodge Challenger
Cadillac Fleetwood
Merc 230
Hornet 4 Drive
Pontiac Firebird
Toyota Corona
Valiant
4
2
0
−2
−5.0
−2.5
0.0
2.5
5.0



Arra oda kell figyelnunk hogy kategorikus valtozok ne keruljenek a fokomponenselemzes valtozoi koze. a vs es am valtozok kategorikus valtozok (csak 0 es 1 erteket vehetnek fel), es a cyl valtozo is tekintheto kategorikusnak, hiszen csak harom erteket vesz fel az adattablankban: 4, 6, 8.

A PCA() funkcioban lehetosegunk van arra hogy meghatarozzunk olyan valtozokat az adatbazisban, amiket nem szeretnenk beepiteni a PCA modellbe. Azokat a folytonos valtozokat, amiket nem szeretnenk figyelembevenni a PCA soran, a quanti.sup parameterben kell megadnunk, azokat pedig amik kategorikusak a quali.sup parameterben. Itt az addott valtozo oszlopszamat kell megadnunk, nem pedig a nevet, igy ezt eloszor ki kell keresnunk. Ezt megtehetjuk a which(names(mtcars) == "valtozo neve") funkcioval.

Az alabbi peldaban a drat folytonos, es a cyl, vs, es az am kategorikus valtozokat kiemeljuk a modellbol, igy azok nincsenek figyelembe veve a pca_mod3 fokomponenseinek meghatarozasa soran, de az abrakon ettol meg szerepelnek. Ebben az esetben termeszetesen a modell ujra illesztesre kerul, es a szamszeru ertekek megvaltoznak a korabbi futtatashoz kepest amikor ezek a valtozok meg szerepeltek a modellben.

```
which(names(mtcars) == "drat")
```

```
## [1] 5
```

```
which(names(mtcars) == "cyl")
```

```
## [1] 2
```

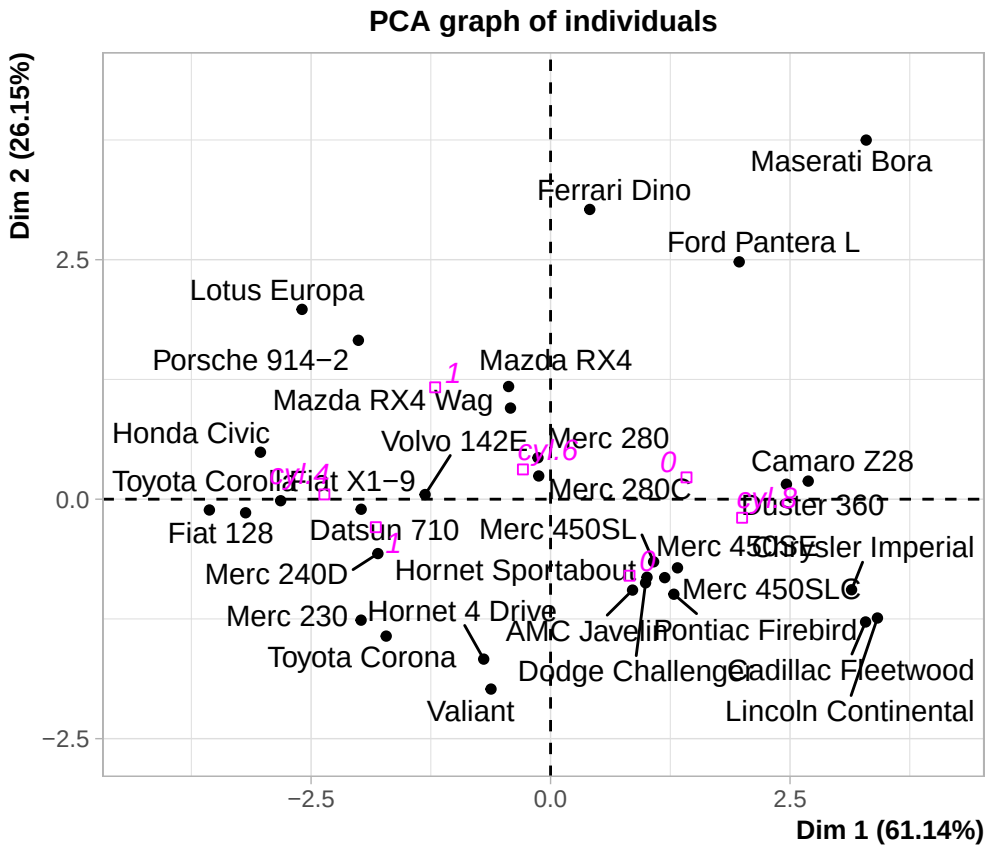
```
which(names(mtcars) == "vs")
```

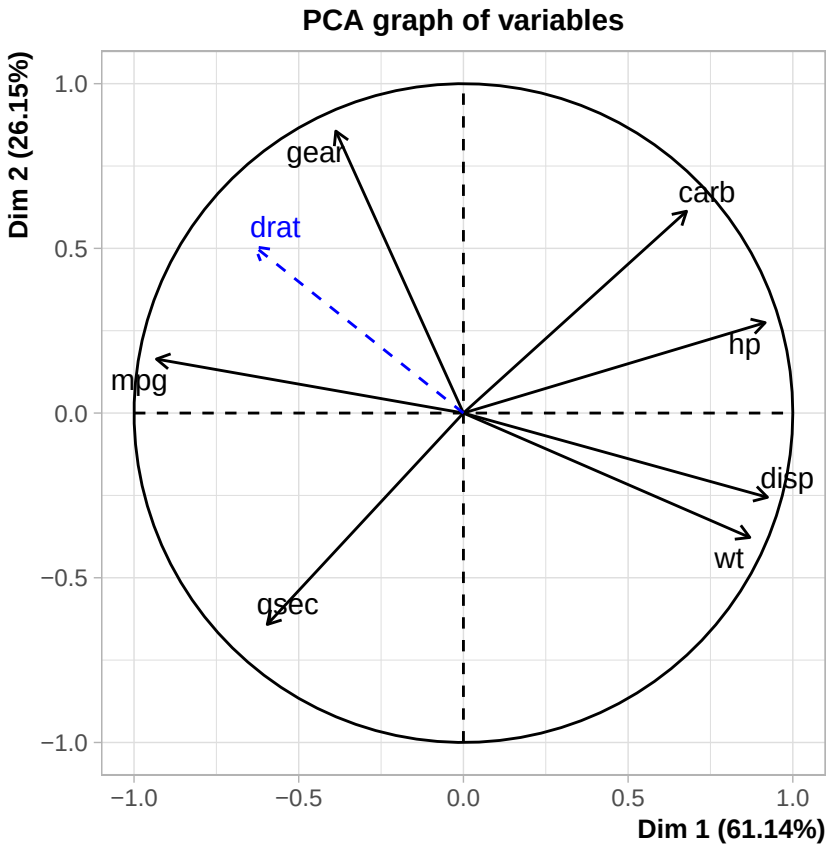
```
## [1] 8
```

```
which(names(mtcars) == "am")
```

```
## [1] 9
```

```
pca_mod2 <- PCA(mtcars, quanti.sup = 5, quali.sup = c(2, 8, 9))
```






```
summary(pca_mod2)
```

```
##
## Call:
## PCA(X = mtcars, quanti.sup = 5, quali.sup = c(2, 8, 9))
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
## Variance               4.280   1.831   0.458   0.184   0.120   0.104   0.024
## % of var.             61.138  26.151   6.550   2.624   1.713   1.488   0.337
## Cumulative % of var.  61.138  87.289  93.839  96.462  98.175  99.663 100.000
##
## Individuals (the 10 first)
##                       Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2
## Mazda RX4          |  1.551 | -0.438  0.140  0.080 |  1.176  2.361  0.575 |
## Mazda RX4 Wag      |  1.322 | -0.418  0.128  0.100 |  0.951  1.544  0.518 |
## Datsun 710         |  2.096 | -1.977  2.854  0.890 | -0.105  0.019  0.003 |
## Hornet 4 Drive     |  1.846 | -0.698  0.356  0.143 | -1.670  4.759  0.818 |
## Hornet Sportabout  |  1.669 |  1.007  0.741  0.364 | -0.815  1.134  0.238 |
## Valiant            |  2.139 | -0.622  0.282  0.084 | -1.982  6.709  0.859 |
## Duster 360         |  2.663 |  2.462  4.428  0.855 |  0.156  0.041  0.003 |
## Merc 240D          |  2.126 | -1.802  2.372  0.719 | -0.568  0.551  0.071 |
## Merc 230           |  3.168 | -1.978  2.857  0.390 | -1.263  2.723  0.159 |
## Merc 280           |  1.130 | -0.131  0.013  0.014 |  0.433  0.320  0.147 |
##                     Dim.3    ctr   cos2
```

```
## Mazda RX4         -0.116  0.092  0.006 |
## Mazda RX4 Wag      0.196  0.262  0.022 |
## Datsun 710        -0.276  0.520  0.017 |
## Hornet 4 Drive    -0.276  0.519  0.022 |
## Hornet Sportabout -0.995  6.746  0.355 |
## Valiant            0.256  0.447  0.014 |
## Duster 360        -0.759  3.930  0.081 |
## Merc 240D          0.839  4.802  0.156 |
## Merc 230           1.944 25.755  0.376 |
## Merc 280           0.917  5.729  0.659 |
##
## Variables
##                      Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr
## mpg               | -0.932 20.294  0.869 |  0.164  1.469  0.027 | -0.095  1.971
## disp              |  0.923 19.913  0.852 | -0.256  3.572  0.065 | -0.136  4.047
## hp                |  0.916 19.612  0.839 |  0.275  4.119  0.075 | -0.058  0.740
## wt                |  0.869 17.664  0.756 | -0.378  7.792  0.143 |  0.249 13.472
## qsec              | -0.596  8.289  0.355 | -0.642 22.505  0.412 |  0.458 45.663
## gear              | -0.387  3.506  0.150 |  0.856 40.036  0.733 |  0.208  9.465
## carb              |  0.677 10.721  0.459 |  0.613 20.508  0.375 |  0.336 24.641
##                     cos2
## mpg                0.009 |
## disp               0.019 |
## hp                 0.003 |
## wt                 0.062 |
## qsec               0.209 |
## gear               0.043 |
## carb               0.113 |
##
## Supplementary continuous variable
##                      Dim.1   cos2    Dim.2   cos2    Dim.3   cos2
## drat              | -0.633  0.400 |  0.505  0.255 |  0.083  0.007 |
##
## Supplementary categories
##                       Dist    Dim.1   cos2 v.test    Dim.2   cos2 v.test
## cyl.4              |  2.367 | -2.361  0.995 -4.600 |  0.049  0.000  0.147 |
## cyl.6              |  0.711 | -0.289  0.165 -0.411 |  0.311  0.191  0.676 |
## cyl.8              |  2.025 |  2.000  0.975  4.746 | -0.194  0.009 -0.704 |
## 0                  |  1.449 |  1.419  0.960  4.331 |  0.227  0.025  1.061 |
## 1                  |  1.862 | -1.824  0.960 -4.331 | -0.292  0.025 -1.061 |
## 0                  |  1.151 |  0.824  0.513  2.682 | -0.799  0.482 -3.975 |
## 1                  |  1.682 | -1.205  0.513 -2.682 |  1.168  0.482  3.975 |
##                     Dim.3   cos2 v.test
## cyl.4               0.049  0.000  0.289 |
## cyl.6               0.393  0.305  1.710 |
## cyl.8              -0.235  0.013 -1.701 |
## 0                  -0.169  0.014 -1.579 |
## 1                   0.218  0.014  1.579 |
## 0                   0.067  0.003  0.666 |
## 1                  -0.098  0.003 -0.666 |
```

3.2 Hany uj dimenziot generaljunk?

A fokomponenselemzes egy dmienzioredukcios technika, vagyis celunk hogy kevesebb dimenzionk legyen az elemzes vegere, mint ahany valtozoval kezdtuk az elemzest. Viszont hogyha ranezunk a model summary-ra, lathatjuk hogy a PCA funkcio alapertelmezett modon pontosan annyi dimenziot generalt mint amennyi valtozonk volt. Meg kell mondanunk a PCA funkcionak, hany dimenaziot akarunk kinyerni. De hogyan tudjuk eldonteni, mennyi az idealis szamu dimenzio?

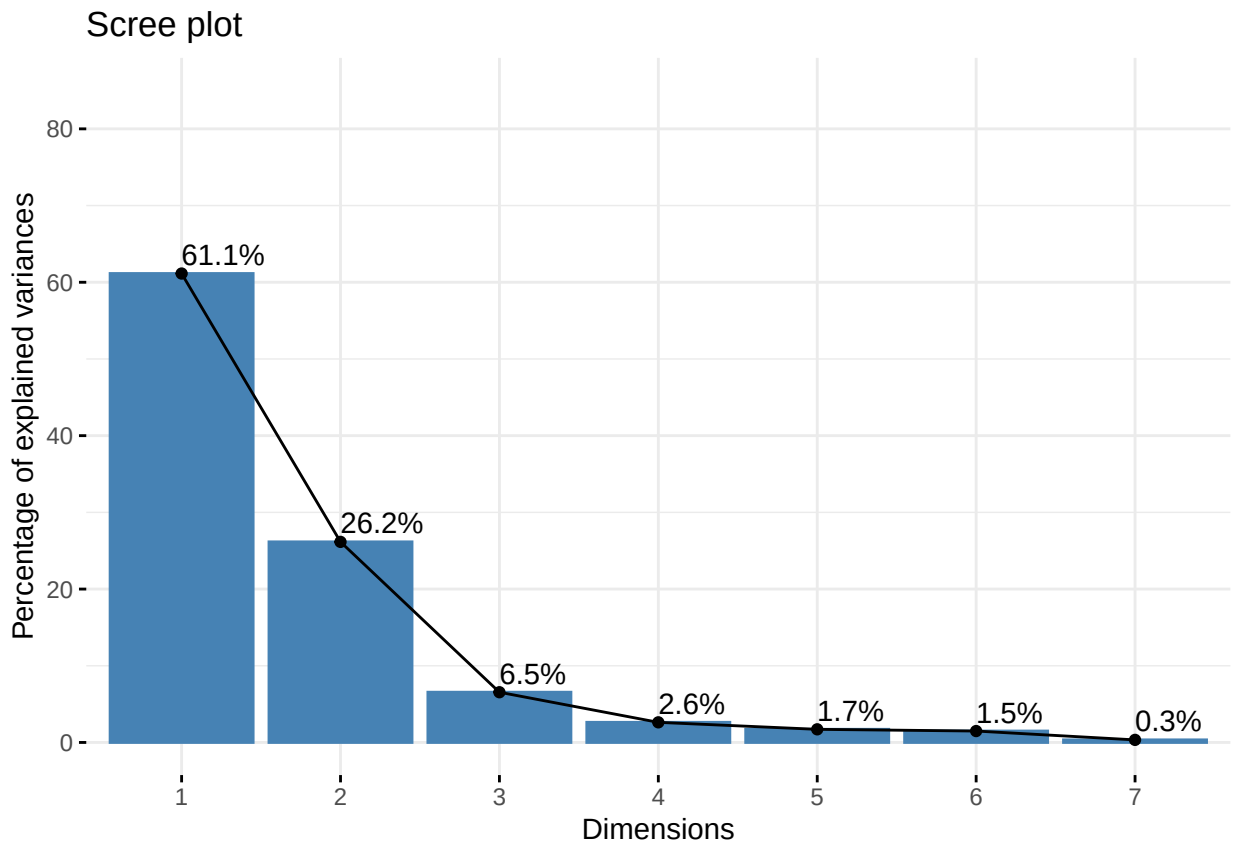
Erre szamos modszer letezik.

1. Scree test

A legismertebb talan a scree-test, ami a megmagyarazott varianciaaranyt abrazolo abra alapjan vegezheto el. Ehhez eloszor a fviz_screeplot() funkcioval abrazolnunk kell az egyes fokomponensek altal megmagyarazott variancia merteket, majd az abra alapjan meg kell allapitanunk, hol van a "tores" a scree-plotban, vagyis hol talalhato az a pont, ami utan mar ellaposodik a megmagyarazott varianciaaranyt abrazolo gorbe. A torespont elotti dimenzional kell hogy megalljon a dimenzio-extrakcio, vagyis annal a dimenzional, ami meg szignifikansan tobb varianciat kepes megmagyarazni, mint a kesobb kinyerd dimenziok. Ezt a megallasi szabalyt ugy is nevezik hogy a "konyok kriterium", mivel a scree plot egy konyokre emlekeztet, es mi a konyokpontot keressuk a gorbeben.

Ezen az abran ugy tunik, hogy a masodik dimenzio utan mar nem erdemes tovabbmennunk, hiszen a harmadik dimenzional megtorik a gorbe es onnantol mar nagyon alacsony a megmagyarazott varianciaarany. Vagyis az idealis dimenzioszam ezen az adaton 2.

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



The Kaiser-Guttman szabaly

Egy masik jol ismert kriterium, hogy azokat a dimenziokat kell megtartanunk, amelyeknek az eigenvalue

erteke 1-nel magasabb. Ez azert van, mert az 1-nel alacsonyabb eigenvalue azt jelenti, hogy a dimenzio kevesebb varianciat magyaraz meg mint az eredeti valtozok atlagosan. A fokomponens elemzes lenyege hogy hasznos osszefoglalo valtozokat generaljunk amik tobb valtozo informaciojat tartalmazzak osszevonva. Azt pedig nem szeretnenk hogy meg az eredeti valtozoinknal is haszontalanabb valtozokat generaljunk, ezert az atlagos valtozoinknal kisebb varianciat megmagyarazo dimenziokat elutasitjuk.

A peldankban ez az elemzes is azt sugallja, hogy ket dimenziot tartsunk meg, hiszen a harmadik dimenziohoz tartozo eigenvalue mar 1-nel kisebb.

```
get_eigenvalue(pca_mod2)
```

```
##       eigenvalue variance.percent cumulative.variance.percent
## Dim.1 4.27967149       61.1381642                    61.13816
## Dim.2 1.83055831       26.1508331                    87.28900
## Dim.3 0.45848582        6.5497975                    93.83879
## Dim.4 0.18365828        2.6236897                    96.46248
## Dim.5 0.11988871        1.7126958                    98.17518
## Dim.6 0.10416965        1.4881379                    99.66332
## Dim.7 0.02356773        0.3366819                   100.00000
```

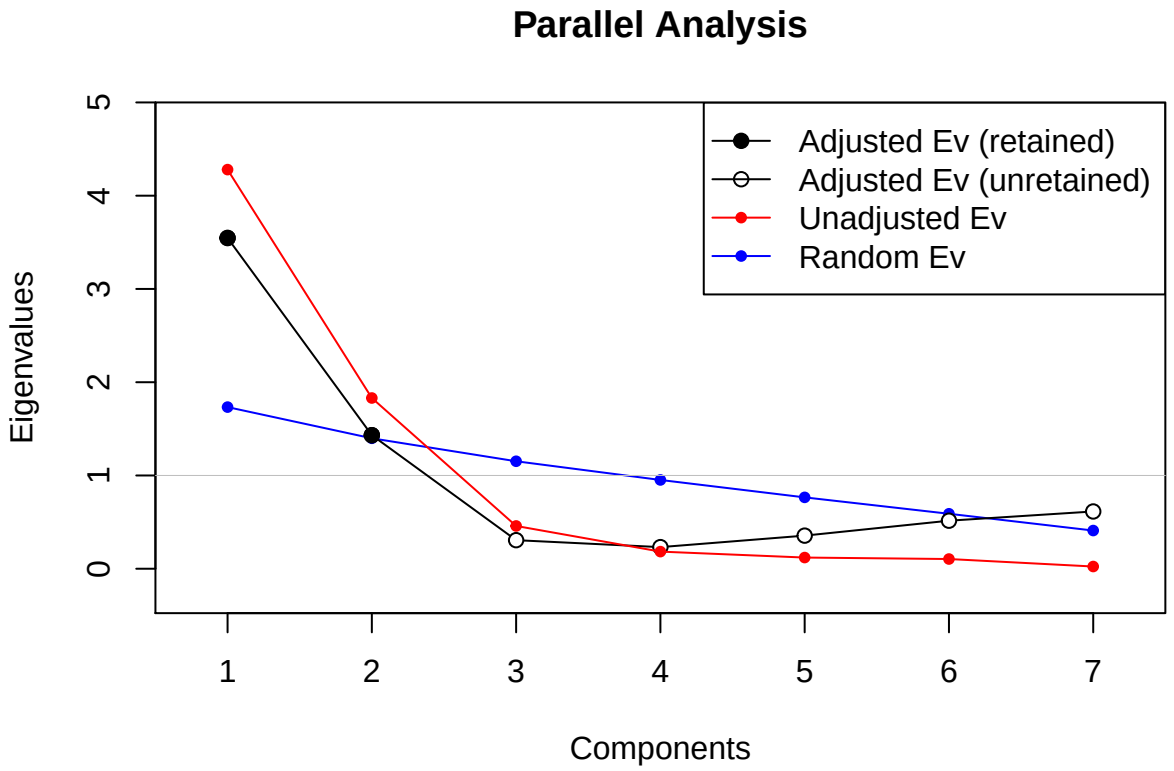
Parallel elemzes

A harmadik (es egyben jelenleg a legelfogadottabb) technika a parallel elemzes technika. Ennek a lenyege hogy az eredeti adattablankhoz hasonlo karakterisztikakkal rendelkezo adatokat generalunk veletlenszeruen, de ugy, hogy abban a valtozok ne korrelaljanak egymassal. Ezt nagyon sokszor megismeteljuk, es ez alapjan a nagy mennyisegu random minta alapjan kiszamoljuk, mi a veletlenszeruen varhato eigenvalue mintazat. Ez egyfajta "null modelkent" funkcional, amihez hasonlithatjuk a sajat adatainkon kapott eignevalue-kat. Azokat a dimenziokat tartjuk meg, amiknek az eigenvalue-ja magasabb mint a random mintakban az adott dimenziohoz tartozo null-eigenvalue.

Ezt a parallel elemzest begezhetjuk el a paran() funkcioval a paran package-bol. Ez a funkcio a null eigenvalue gorbe vizualizalasara is kepes a graph = TRUE prameter beallitasaval, melyet osszehasonlithatunk az adatainkban kapott eigenvalue-val. Az output objektum $Retained komponense megmutatja, az elemzes hany dimenzio megtartasat javasolja.

```
mtcars_pca_ret = paran(mtcars[, -c(2, 5, 8, 9)], 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
## 210 iterations, using the mean estimate
##
## ----------------------------------------------------
## Component   Adjusted    Unadjusted    Estimated
##             Eigenvalue  Eigenvalue    Bias
## ----------------------------------------------------
## 1           3.546540    4.279671      0.733131
## 2           1.431950    1.830558      0.398608
## ----------------------------------------------------
##
## Adjusted eigenvalues > 1 indicate dimensions to retain.
## (2 components retained)
```

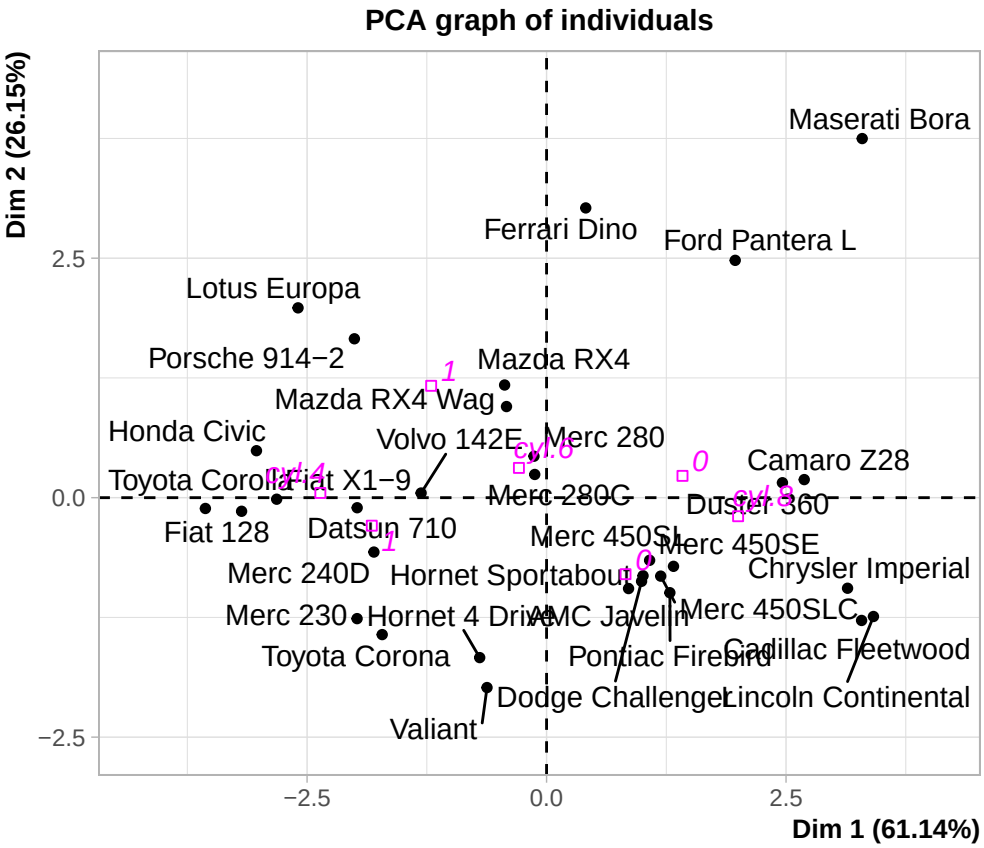


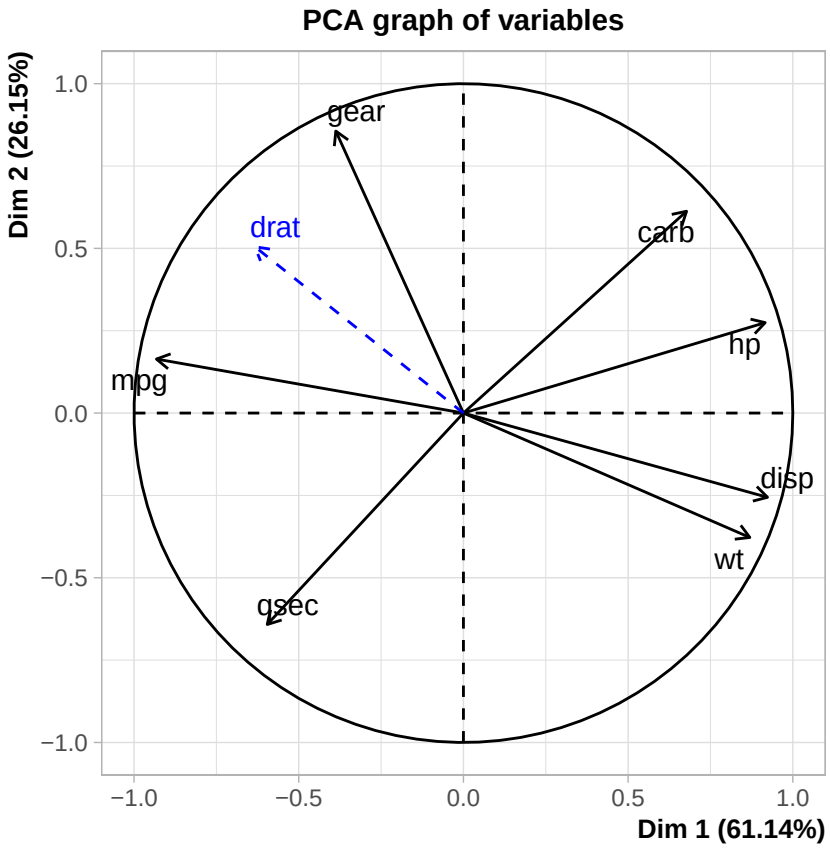

```
mtcars_pca_ret$Retained
```

```
## [1] 2
```

Amint meghataroztuk az idealis dimenziok szamat, ujra lefuttathatjuk az elemzesunket, de ezuttal mar specifikalmva, mennyi dimenziot szeretnenk, a npc parameter beallitasaval.

```
pca_mod3 <- PCA(mtcars, ncp = 2, quanti.sup = 5, quali.sup = c(2,
    8, 9))
```


PCA graph of individuals
Dim 2 (26.15%)
Dim 1 (61.14%)
2.5
0.0
−2.5
−2.5
0.0
2.5
Maserati Bora
Ferrari Dino
Ford Pantera L
Lotus Europa
Porsche 914−2
Mazda RX4
Mazda RX4 Wag
Honda Civic
Volvo 142E
Merc 280
Camaro Z28
Toyota Corolla
Fiat X1−9
Merc 280C
Duster 360
Fiat 128
Datsun 710
Merc 450SL
Merc 450SE
Merc 240D
Hornet Sportabout
Chrysler Imperial
Merc 230
Hornet 4 Drive
AMC Javelin
Merc 450SLC
Toyota Corona
Pontiac Firebird
Cadillac Fleetwood
Dodge Challenger
Lincoln Continental
Valiant
cyl:4
cyl:6
cyl:8
0
1
0
1
0




```
summary(pca_mod3)
```

```
##
## Call:
## PCA(X = mtcars, ncp = 2, quanti.sup = 5, quali.sup = c(2, 8,
##      9))
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
## Variance               4.280   1.831   0.458   0.184   0.120   0.104   0.024
## % of var.             61.138  26.151   6.550   2.624   1.713   1.488   0.337
## Cumulative % of var.  61.138  87.289  93.839  96.462  98.175  99.663 100.000
##
## Individuals (the 10 first)
##                       Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2
## Mazda RX4          |  1.551 | -0.438  0.140  0.080 |  1.176  2.361  0.575 |
## Mazda RX4 Wag      |  1.322 | -0.418  0.128  0.100 |  0.951  1.544  0.518 |
## Datsun 710         |  2.096 | -1.977  2.854  0.890 | -0.105  0.019  0.003 |
## Hornet 4 Drive     |  1.846 | -0.698  0.356  0.143 | -1.670  4.759  0.818 |
## Hornet Sportabout  |  1.669 |  1.007  0.741  0.364 | -0.815  1.134  0.238 |
## Valiant            |  2.139 | -0.622  0.282  0.084 | -1.982  6.709  0.859 |
## Duster 360         |  2.663 |  2.462  4.428  0.855 |  0.156  0.041  0.003 |
## Merc 240D          |  2.126 | -1.802  2.372  0.719 | -0.568  0.551  0.071 |
## Merc 230           |  3.168 | -1.978  2.857  0.390 | -1.263  2.723  0.159 |
## Merc 280           |  1.130 | -0.131  0.013  0.014 |  0.433  0.320  0.147 |
```

```
##
## Variables
##                       Dim.1    ctr   cos2    Dim.2    ctr   cos2
## mpg                | -0.932 20.294  0.869 |  0.164  1.469  0.027 |
## disp               |  0.923 19.913  0.852 | -0.256  3.572  0.065 |
## hp                 |  0.916 19.612  0.839 |  0.275  4.119  0.075 |
## wt                 |  0.869 17.664  0.756 | -0.378  7.792  0.143 |
## qsec               | -0.596  8.289  0.355 | -0.642 22.505  0.412 |
## gear               | -0.387  3.506  0.150 |  0.856 40.036  0.733 |
## carb               |  0.677 10.721  0.459 |  0.613 20.508  0.375 |
##
## Supplementary continuous variable
##                       Dim.1   cos2    Dim.2   cos2
## drat               | -0.633  0.400 |  0.505  0.255 |
##
## Supplementary categories
##                        Dist    Dim.1   cos2 v.test    Dim.2   cos2 v.test
## cyl.4              |  2.367 | -2.361  0.995 -4.600 |  0.049  0.000  0.147 |
## cyl.6              |  0.711 | -0.289  0.165 -0.411 |  0.311  0.191  0.676 |
## cyl.8              |  2.025 |  2.000  0.975  4.746 | -0.194  0.009 -0.704 |
## 0                  |  1.449 |  1.419  0.960  4.331 |  0.227  0.025  1.061 |
## 1                  |  1.862 | -1.824  0.960 -4.331 | -0.292  0.025 -1.061 |
## 0                  |  1.151 |  0.824  0.513  2.682 | -0.799  0.482 -3.975 |
## 1                  |  1.682 | -1.205  0.513 -2.682 |  1.168  0.482  3.975 |
```

3.3 Fokomponenselemzes eredmenyeinek ertelmezese

3.3.1 a PCA modell opjektum reszei

A modell osszefoglalobol (model summary) tovabbi hasznos informaciok olvashatok ki. Az Eigenvalues reszben megtudhatjuk hogy az egyes dimenziok az adatok teljes varianciajanak hany szazalekat magyarazzak meg (% of var), es hogy a legfontosabbtol a legalacsonyabbig egyesevel osszevonva mekkora a tobb dimenzio altal megmagyarazott osszesitett varianciaarany (Cumulative % of var). Vagyis a Dim.3-hoz tartozo % of variance ertek (6.55) azt mutatja, hogy a harmadikkent kinyert dimenzio az adatok varianciajanak 655%-at tudja megmagyarazni onmagaban. Az Dim.3-hoz tartozo dim Cumulative % of var ertek (93.84 pedig azt mutatja, hogy a Dimenzio 3 a Dim.1 es Dim.2-vel egyutt kozosen az adatok varianciajanak (9384%-at kepesek megmagyarazni. Ha csak az eigenvalue-t es a megmagyarazaott varianaciaaranyokat tartalmazo tablazat erdekel minket, ezt kinyerhetjuk ugy hogy csak a pca_mod3$eig komponenst listazzuk ki.

A model summary arrol is tartalmaz informaciot a Variables reszeben, hogy az egyes valtozok hogyan korrelalnak az egyes uj dimenziokkal (a Dim.1, Dim.2, Dim.2 ... oszlopokaban), es hogy mekkora a hozzajarulasuk az adott valtozo altal megmagyarazott varianciahoz (a ctr oszlopban). Ez egy nagyon fontos tablazat, mert innen tudjuk leolvasni (az abrak mellett) hogy az egyes valtozokat mely dimenziok (faktorok) irjak le leginkabb. Bovebb informaciot talalunk ha kilistazzuk a pca_mod3$var komponenst.

```
# Get the summary the outputs.
summary(pca_mod3)
```

```
##
## Call:
## PCA(X = mtcars, ncp = 2, quanti.sup = 5, quali.sup = c(2, 8,
##     9))
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
```

```
## Variance               4.280   1.831   0.458   0.184   0.120   0.104   0.024
## % of var.             61.138  26.151   6.550   2.624   1.713   1.488   0.337
## Cumulative % of var.  61.138  87.289  93.839  96.462  98.175  99.663 100.000
##
## Individuals (the 10 first)
##                       Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2
## Mazda RX4          |  1.551 | -0.438  0.140  0.080 |  1.176  2.361  0.575 |
## Mazda RX4 Wag      |  1.322 | -0.418  0.128  0.100 |  0.951  1.544  0.518 |
## Datsun 710         |  2.096 | -1.977  2.854  0.890 | -0.105  0.019  0.003 |
## Hornet 4 Drive     |  1.846 | -0.698  0.356  0.143 | -1.670  4.759  0.818 |
## Hornet Sportabout  |  1.669 |  1.007  0.741  0.364 | -0.815  1.134  0.238 |
## Valiant            |  2.139 | -0.622  0.282  0.084 | -1.982  6.709  0.859 |
## Duster 360         |  2.663 |  2.462  4.428  0.855 |  0.156  0.041  0.003 |
## Merc 240D          |  2.126 | -1.802  2.372  0.719 | -0.568  0.551  0.071 |
## Merc 230           |  3.168 | -1.978  2.857  0.390 | -1.263  2.723  0.159 |
## Merc 280           |  1.130 | -0.131  0.013  0.014 |  0.433  0.320  0.147 |
##
## Variables
##                      Dim.1    ctr   cos2    Dim.2    ctr   cos2
## mpg                | -0.932 20.294  0.869 |  0.164  1.469  0.027 |
## disp               |  0.923 19.913  0.852 | -0.256  3.572  0.065 |
## hp                 |  0.916 19.612  0.839 |  0.275  4.119  0.075 |
## wt                 |  0.869 17.664  0.756 | -0.378  7.792  0.143 |
## qsec               | -0.596  8.289  0.355 | -0.642 22.505  0.412 |
## gear               | -0.387  3.506  0.150 |  0.856 40.036  0.733 |
## carb               |  0.677 10.721  0.459 |  0.613 20.508  0.375 |
##
## Supplementary continuous variable
##                      Dim.1   cos2    Dim.2   cos2
## drat               | -0.633  0.400 |  0.505  0.255 |
##
## Supplementary categories
##                       Dist    Dim.1   cos2 v.test    Dim.2   cos2 v.test
## cyl.4              |  2.367 | -2.361  0.995 -4.600 |  0.049  0.000  0.147 |
## cyl.6              |  0.711 | -0.289  0.165 -0.411 |  0.311  0.191  0.676 |
## cyl.8              |  2.025 |  2.000  0.975  4.746 | -0.194  0.009 -0.704 |
## 0                  |  1.449 |  1.419  0.960  4.331 |  0.227  0.025  1.061 |
## 1                  |  1.862 | -1.824  0.960 -4.331 | -0.292  0.025 -1.061 |
## 0                  |  1.151 |  0.824  0.513  2.682 | -0.799  0.482 -3.975 |
## 1                  |  1.682 | -1.205  0.513 -2.682 |  1.168  0.482  3.975 |
```

```
pca_mod3$eig
```

```
##        eigenvalue percentage of variance cumulative percentage of variance
## comp 1 4.27967149             61.1381642                           61.13816
## comp 2 1.83055831             26.1508331                           87.28900
## comp 3 0.45848582              6.5497975                           93.83879
## comp 4 0.18365828              2.6236897                           96.46248
## comp 5 0.11988871              1.7126958                           98.17518
## comp 6 0.10416965              1.4881379                           99.66332
## comp 7 0.02356773              0.3366819                          100.00000
```

```
pca_mod3$var
```

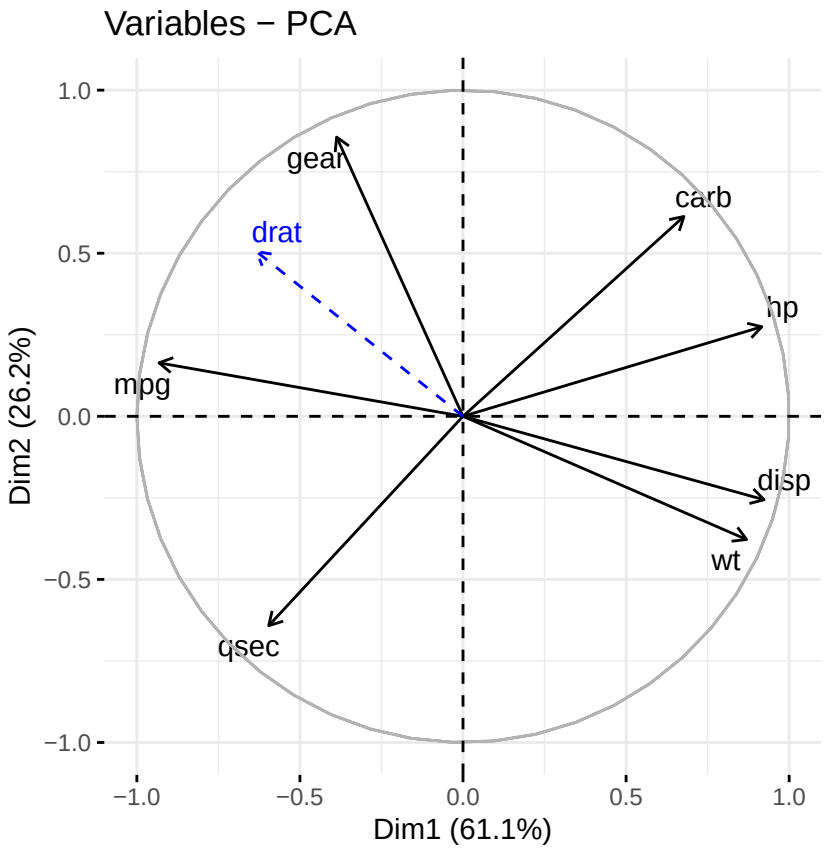
```
## $coord
```

```
##           Dim.1      Dim.2
## mpg  -0.9319530  0.1639662
## disp  0.9231535 -0.2557244
## hp    0.9161546  0.2745792
## wt    0.8694559 -0.3776643
## qsec -0.5956200 -0.6418408
## gear -0.3873752  0.8560856
## carb  0.6773530  0.6127089
##
## $cor
##           Dim.1      Dim.2
## mpg  -0.9319530  0.1639662
## disp  0.9231535 -0.2557244
## hp    0.9161546  0.2745792
## wt    0.8694559 -0.3776643
## qsec -0.5956200 -0.6418408
## gear -0.3873752  0.8560856
## carb  0.6773530  0.6127089
##
## $cos2
##          Dim.1      Dim.2
## mpg  0.8685364 0.02688491
## disp 0.8522124 0.06539496
## hp   0.8393393 0.07539375
## wt   0.7559536 0.14263036
## qsec 0.3547632 0.41195966
## gear 0.1500596 0.73288249
## carb 0.4588071 0.37541218
##
## $contrib
##          Dim.1     Dim.2
## mpg  20.294463  1.468673
## disp 19.913033  3.572405
## hp   19.612235  4.118621
## wt   17.663823  7.791632
## qsec  8.289496 22.504591
## gear  3.506334 40.036009
## carb 10.720615 20.508070
```

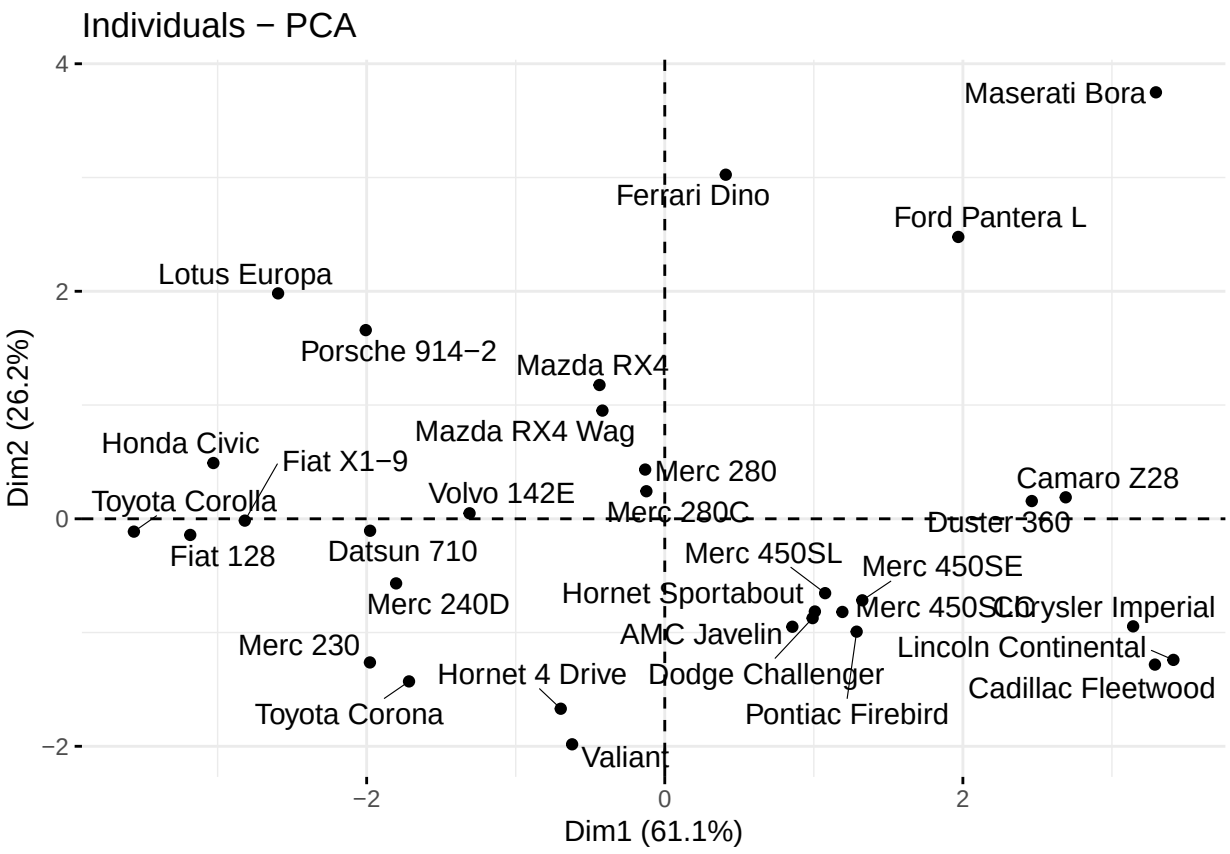
3.3.2 Az eredmenyek vizualizalasa

Az eredmenyek vizualizalasa segithet a komponensek ertelmezeseben. A fviz_pca_var() es a fviz_pca_ind() segitsegevel reprodukalhatjuk a PCA funcio altal eredetileg general abrakat. Sot, a kettot ossze is vonhatjuk a fviz_pca_biplot() funkcioval. Igy egyszerre lathatjuk hogy a ket legfontosabb dimenzio menten hol helyezkednek el az egyes megfigyelesek (az autok), es hogy a dimenziok foleg mely valtozokat reprezentaljak. (A repel = T parameterbeallitas arra jo hogy a feliratok ne fedjek egymast hanem elcsusztatva szerepeljenek az abran ha tul kozel lennenek egymashoz)

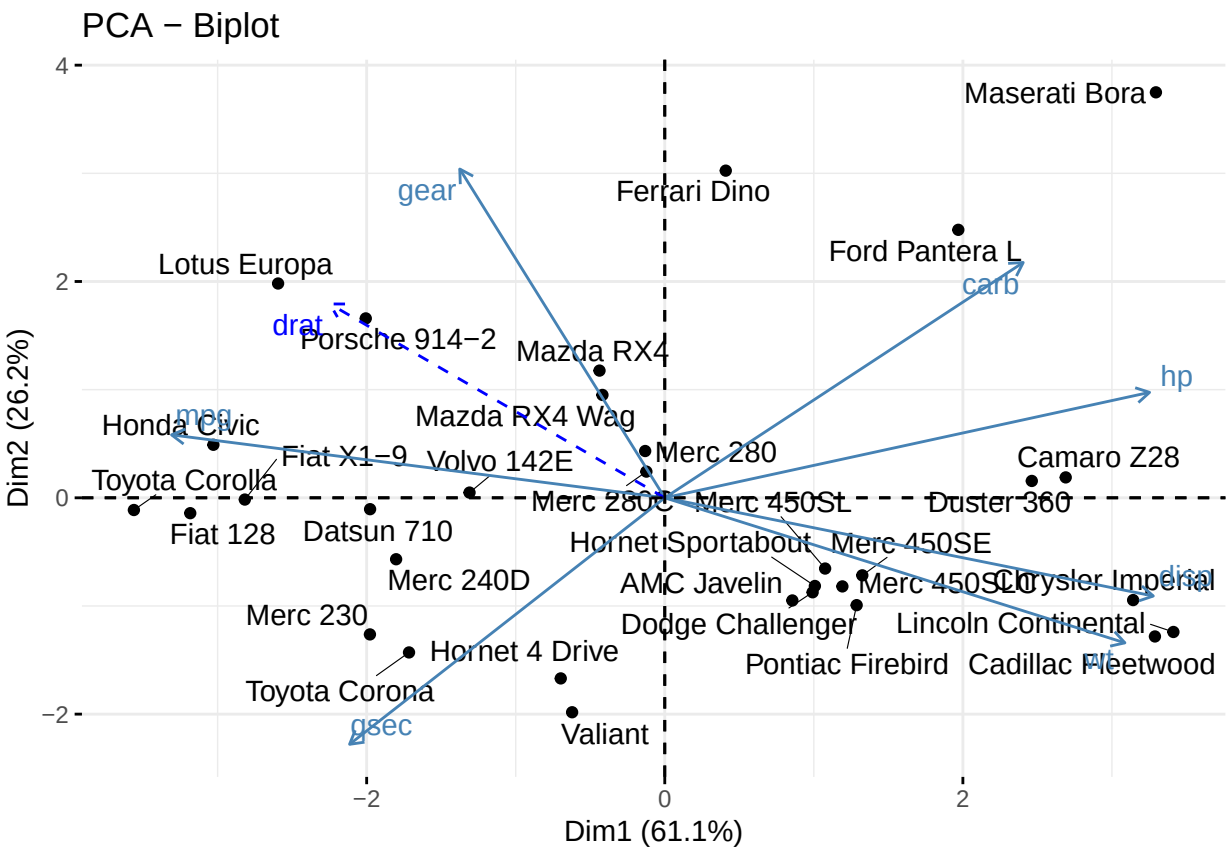
```
fviz_pca_var(pca_mod3, repel = T)
```




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




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

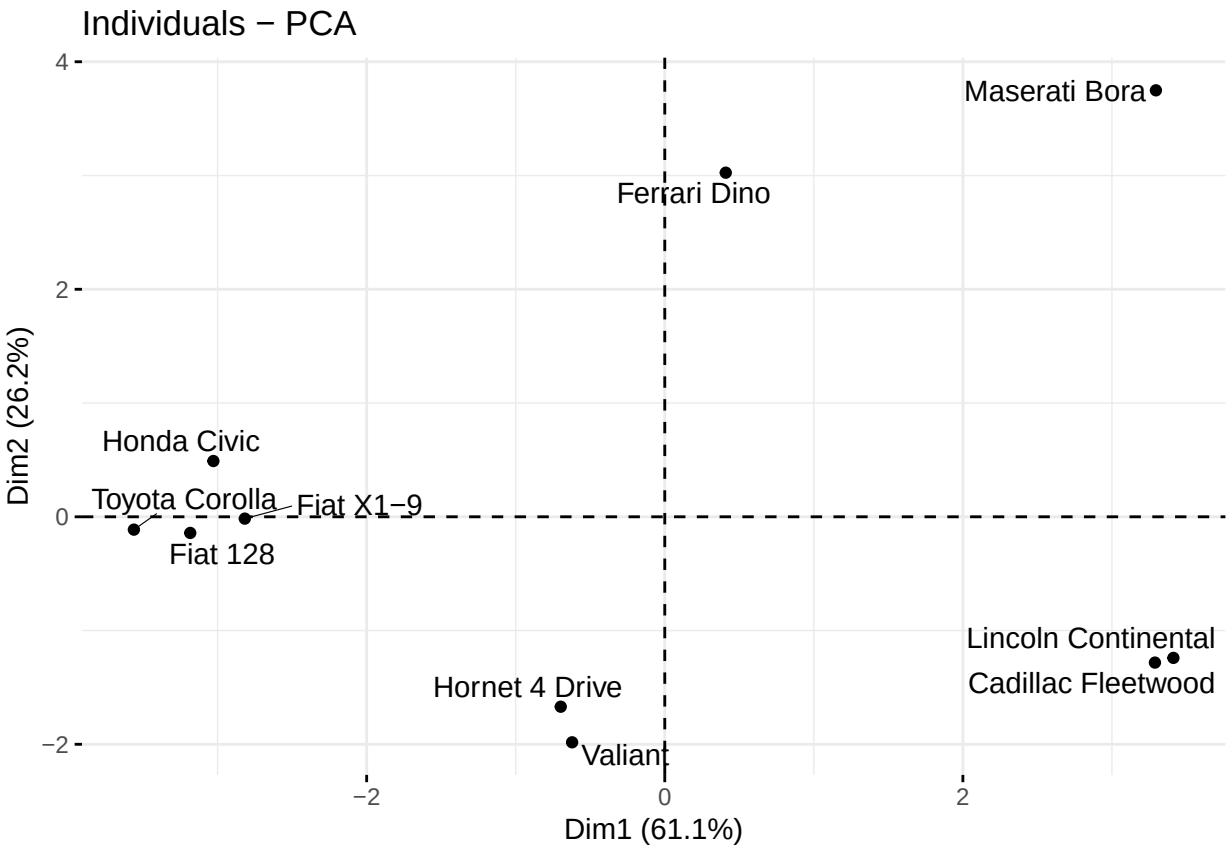


Az abrakat tovabb tunningolhatjuk azzal, hogy abrazoljuk rajtuk az egyes valtozok vagy megfigyelesek hozzajarulasat (contribution) az abrazolt dimenziohoz a , col.var = “contrib” es , col.ind = “contrib” parametereken keresztul.

Azt is megtehetjuk, hogy a select.ind = parameteren keresztul hogy csak bizonyos megfigyeleseket teszunk az abrara.Pl. cosˆ2 ertek azt mutatja, hogy az adott megfigyeles vagy valtozo menyire jol reprezentalt az adott dimenzio altal. A select.ind = list(cos2 = 10) parameter beallitasaval meghatarozhatjuk, hogy csak az a 10 megfigyeles szerepeljen az abran, akiknek a ket diemnziora vonatkozo cosˆ2 osszege a legmagasabb. Vagyis a ket dimenzio altal leirt dimenzioter 10 legreprezentativabb megfigyelese.

Ez az abra azt mutatja hogy az alacsony Dim.1, kozepes Dim.2 legtipikusabb tagjai pl. a Honda Civic, a Toyota Corolla, a magas Dim.2. kozepes Dim.1 legtipikusabb tagja talan a Ferrari Dino, mig a magas Dim1. es magas Dim.2. legtipikusabb tagja a Maserati Bora. Ez fontos lehet a dimenziok ertemezeseben.

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



Egy masik fontos abratipus az egyes dimenziok ertelmezesenek elosegitesehez a fviz_contrib() altal generalt barchart, ami az egyes valtozok egyes dimenziokhoz valo hozzajarulasat mutatja meg. Az axes = parameterrel allithatjuk be, melyik dimenziora vagyunk kivancsiak. A piros szaggatott vonal azt mutatja, hogy mi lenne az elvart hozzajarulas szazaleka abban az esetben ha minden valtozo azonos mertekben jarulna hozza a dimenzio megmagyarazasahoz.

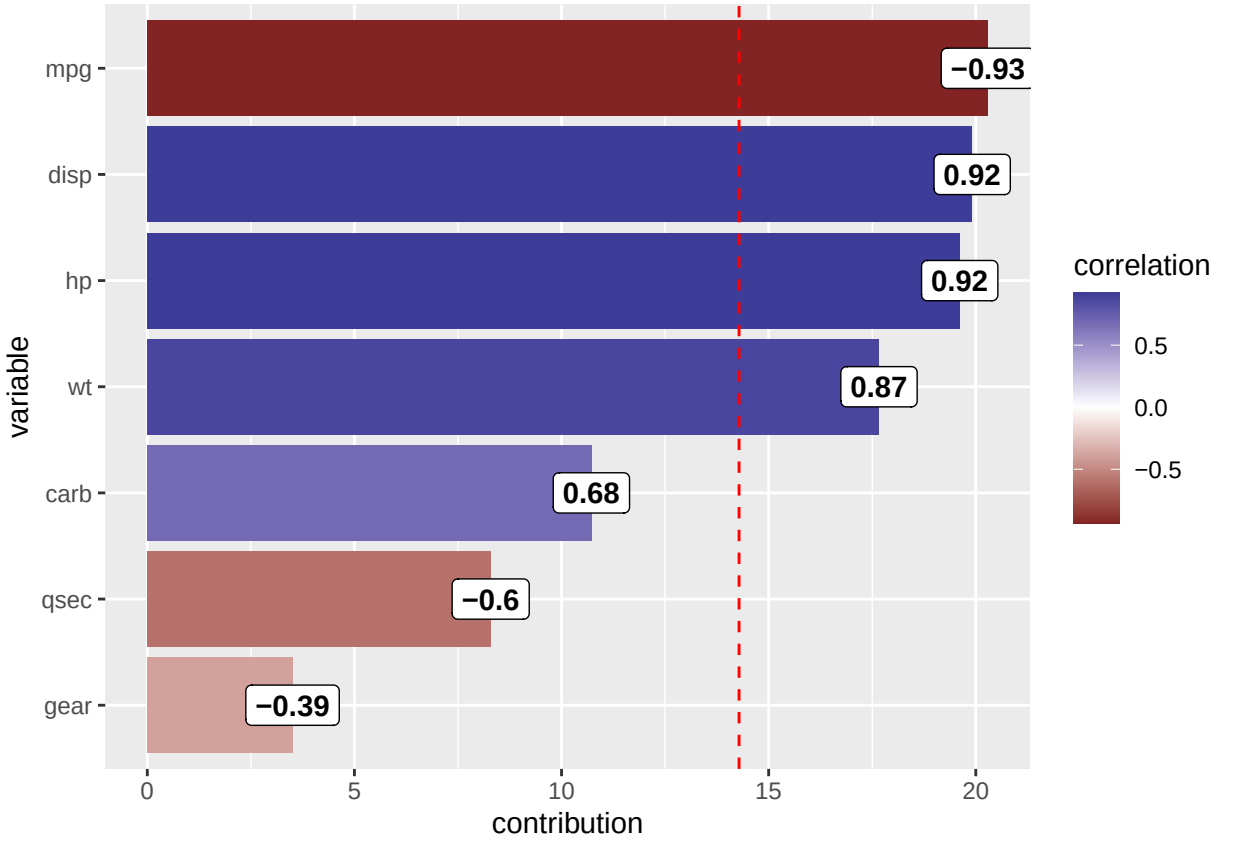
Ez az abra akkor lenne igazab informativ ha a korrelacio merteke es iranya is egyertelmu lenne rola. Ezt onmagaban nem tartalmazza a fviz_contrib() funkcio, ezert a fviz_loadnings_with_cor() sajat funkcio hasznalataval helyettesitjuk, melyen az oszlopok a korrelacio szerint vannak szinezve es a korrelacio feliratkent is szerepel az abran.

Ezek az abra azt mutatjak, hogy a Dim.1-hez elsosorban az mpg, dist, hp, es wt valtozok jarulnak hozza, mig a Dim.2-hoz elsosorban a gear qsec es a carb valtozok jarulnak hozza.

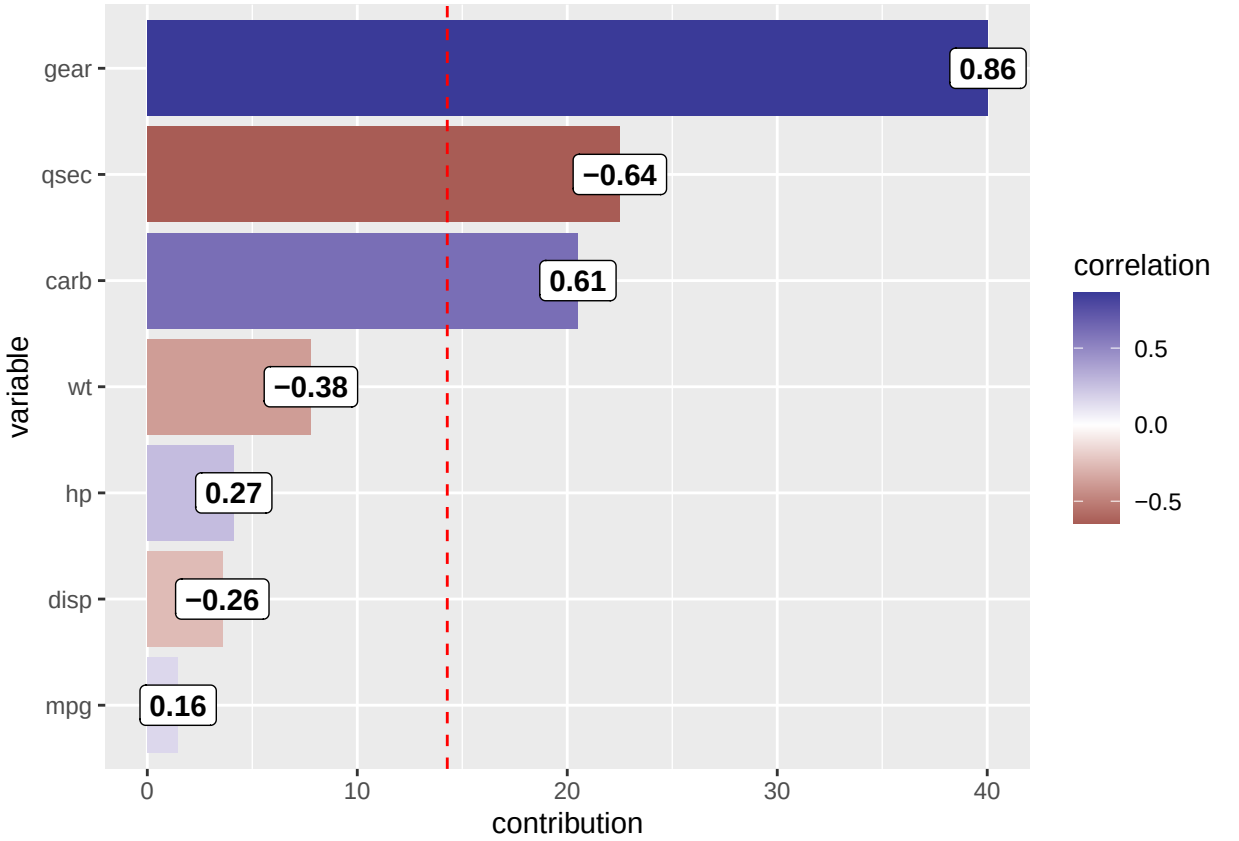
Ezek alapjan az abrak alapjan, es a reprezentativ esetek abraja alapjan mit gondolsz, hogyan nevezhetnek el az egyes es a kettes dimenziot?

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

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



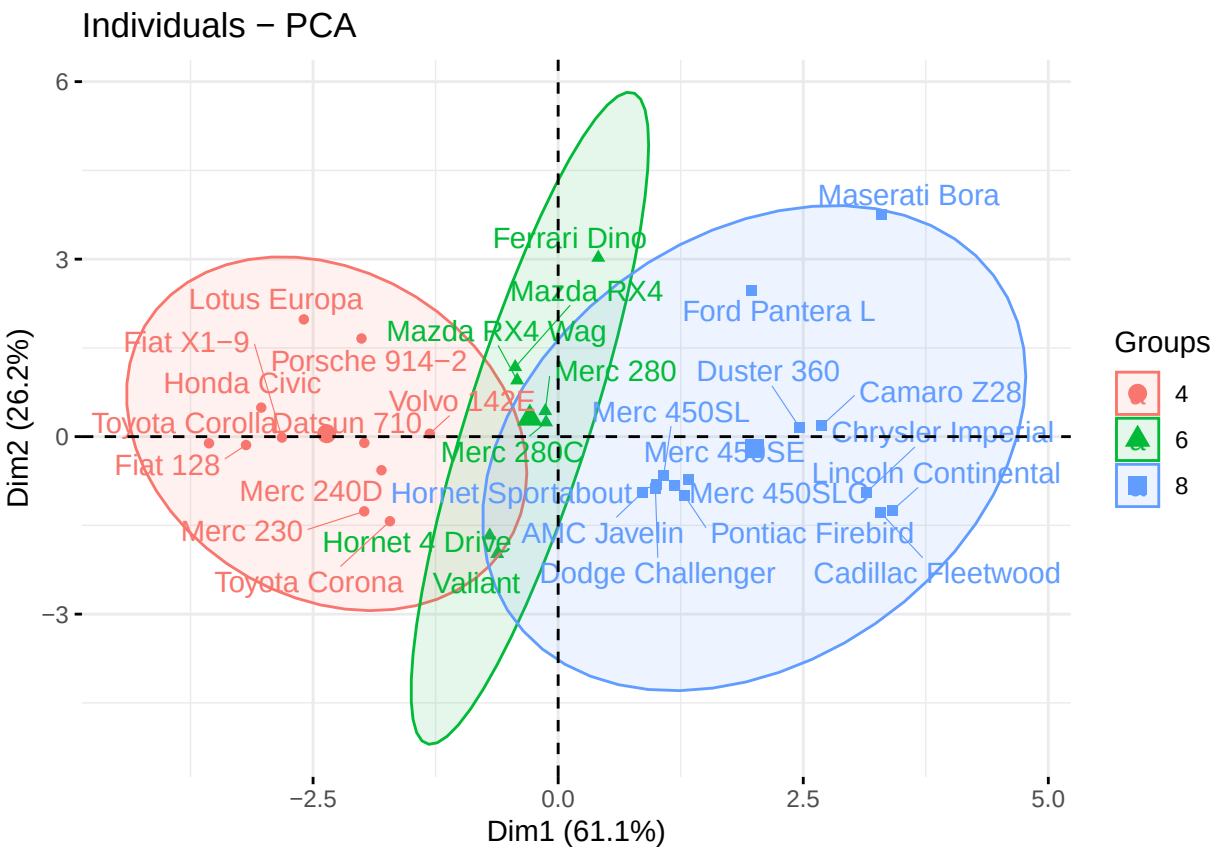

```
fviz_loadnings_with_cor(mod = pca_mod3, axes = 2)
```



A vizualizaciot arra is hasznalhatjuk, hogy csoportositsuk a megfigyeleseket a dimenziokon mutatott ertekuk alapjan. Ezt az addEllipses = T parameterrel adhatjuk meg.

Hogyan jellemezned az egyes elipszisekben talalhato autokat az alapjan, hogy az 1. es 2. dimenzion milyen ertekekt vesznek fel?

```
fviz_pca_ind(pca_mod3, label = "ind", repel = T, habillage = factor(mtcars$cyl),
    addEllipses = T)
```



4 Bevezetes a feltaro faktorelemzesbe (Exploratory Factor Analysis - EFA)

A faktorelemzes egy masik dimenzioredukcios technika ami hasonlit a fokomponenselemzeshez. A ketto kozott fontos kulonbseg hogy a faktorelemzest akkor hasznaljuk, ha feltetelezzuk hogy a sok valtozonk hattereben kozos okok, ugynevezett latens faktorok allnak, es ez okozza, hogy a megfigyelt valtozoink korrelalnak egymassal.

Amikor egy feltaro faktorelemzesi (EFA) modellt epitunk, nem probaljuk megmagyarazni a teljes varianciat az adatokban, mert megengedjuk hogy a latens faktorok csak reszben magyarazzak a megfigyelt valtozok varianciajat. A fennmarado varianciat vagy meresi hiba, vagy olyan faktorok befolyasoljak, amik egyediak a megfigyelt valtozora. Ezert az EFA-ban minden egyes megfigyelt valtozohoz tartozik egy “kommunalitas” (communality) ertek. Ez az ertek azt mutatja meg, hogy az adott valtozoban megfigyelheto variancia mekkora hanyadat magyarazzak a latens faktorok. A fennmarado varianciat a valtozora egyedi faktor vagy meresi hiba magyarazza (ezt egyedisegnek, vagy uniqueness-nek is nevezzuk).

A faktorelemzes legfontsabb lepesei:

- Faktoralhatosag ellenorzese
- Faktorkinyeres
- Idealis faktorszam kivalasztasa
- Faktorforgatas
- Faktorok ertelmezese

4.1 Uj adatok

Alabb betoltjuk a "Human Styles Questionnaire" adatbazist, ami a Martin et. al. (2003). kutatasabol szarmazik, akik a HSQ kerdoivet vettek fel 1071 szemellyel.

Az adatbazis elso 32 oszlopa Q1-Q32 a kerdoiv egyes teteleire adott valaszokat tartalmazza minden szemelytol. 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 faktorelemzessel az a celunk, hogy azonositsuk a hatterben megbuvo pszichologiai vonasokat, amik meghatarozzak az egyes embereke hogyan valaszolnak ezekre a tetelekre.

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

```
## Parsed with column specification:
## cols(
##   .default = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

```
hsq %>% describe()
```

4.2 Adatok faktoralhatosaga

Az adatfaktoralhatosag tesztelesekor azt a kerdest valaszoljuk meg, hogy van-e elegendo egyuttjaras (korrelacio) a megfigyelheto valtozok kozott, ami lehetove teszi az EFA elvegzeset. Ennek tesztelesere ket modszert is alkalmazunk: a Bartlett sphericity tesztet es a Kaiser-Meyer-Olkin tesztet.

Mindenek elott azonban az adatok korrelacios matrixara van szuksegunk, amin ezeket a teszteket lefuttathatjuk. Ezt megkaphatnank a cor() funkcioval ha folytonos valtozokkal dolgoznank, de ebben az adatbazisban ordinalis adatokkal van dolgunk, igy egy masik funkciot hasznalunk aminek a neve mixedCor() a psych package-bol. Ez a funkcio kepes az ordinalis adatok eseten hasznalatos "Polychoric Correlation" meghatarozasara. A mixedCor() funkcioban meghatarozzuk hogy melyek a folytonos valtozok, es melyek az ordinalis valtozok. A Q1-Q32 mind ordinalis, ezert csak a p = 1:32-t hatarozzuk meg, a c=-t pedig NULL-ra allitjuk, mert nincs folytonos skalan mozgo (continuous) valtozo.

Fontos, hogy a korrelacios matrixot a mixedCor() a $rho komponenseben tarolja, ezert ezt kell elmentenunk egy uj adatobjektumba. Mentsuk el ezt a hsq_correl nevu objektumba.

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

```
## Warning in matpLower(x, nvar, gminx, gmaxx, gminy, gmaxy): 70 cells were
## adjusted for 0 values using the correction for continuity. Examine your data
## carefully.
```

```
hsq_correl = hsq_mixedCor$rho
```

_____Gyakorlas_____

A fentebb tanultak alapjan vizualizald a valtozok kozotti korrelaciot. Hasznalj tobb modszert is, pl. ggcorr(), ggcorrplot() hc.order=TRUE-val kombinalva, vagy network_plot().

Bartlett sphericity teszt

A bartlett teszt lenyege hogy a valos korrelacios matrixot osszehasonlitjuk egy hipotetikus null-korrelacios matrix-al, amiben minden korrelacio 0 erteket vesze fel (identity matrix). A null hipotezis amit itt tesztelunk

az, hogy a ket korrelacios matrix nem kulonbozik egymastol. Ha a teszt szignifikans, az azt jelenti hogy az adattabla valtozoi korrelalnak egymassal.

Azonban fontos megjegyezni, hogy a Bartlett tesztnek van egy hatulutoje, megpedig hogy nagy elemszamoknal szinte biztosan szignifikans eredmenyt ad. Csak olyankor erdemes erre a mutatora hagyatkozni a faktoralhatosag megallapitasakor amikor amikor a megfigyelesek szama es a megfigyelt valtozok szamanak aranya kisebb mint 5. A mi esetunkben ez az arany $1071/32 = 33.5$, vagyis a Bartlett teszt eredmenye nem megbizhato.

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

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

```
bfi_factorability
```

```
## $chisq
## [1] 1270.981
##
## $p.value
## [1] 1.861713e-69
##
## $df
## [1] 496
```

Kaiser-Meyer-Olkin (KMO) teszt

A KMO teszt a parcialis korrelacios matrixot hasonlitja ossze a szokasos korrelacios matrixal. A parcialis korrelacio soran meghatarozzuk hogy mekkora ket valtozo kozotti korrelacio, ha kivonjuk a tobbi valtozo hatasat a korrelaciobol. A KMO ertek azt mutatja, hogy mekkora a kulonbseg a parcialis korrelaciok es a szokasos korrelaciok kozott. A KMO egy kulonbseg ertek, a parcialis korrelaciok es a szokasos korrelaciok kozotti kulonbseget jelzi. Azokban az esetekben ahol a valtozok sok kozos varianciat hordoznak (vagyis valoszinu hogy mogottuk egy kozos latens faktor all), a parcialis korrelaciok alacsonyak, vagyis a KMO index magas. A KMO tesztben az 1-hez kozeli ertekek jok jo faktoralhatosagot mutatnak. A KMO index-nek legalabb 0.6-nak kell lennie hogy ugy iteljuk hogy a valtozok faktoralhatoak.

A mi peldankban a KMO minden valtozo eseten magasabb 0.6-nal, es az osszesitett KMO is magasabb 0.6-nal, igy faktoralhatonak tekinthetok a valtozok.

```
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 0.94 0.91 0.90 0.91 0.88 0.83 0.85 0.95 0.86 0.79 0.90 0.87 0.93 0.83 0.88
##  Q17  Q18  Q19  Q20  Q21  Q22  Q23  Q24  Q25  Q26  Q27  Q28  Q29  Q30  Q31  Q32
## 0.90 0.83 0.90 0.82 0.90 0.88 0.83 0.83 0.85 0.90 0.84 0.93 0.88 0.80 0.82 0.91
```

4.3 Faktorextrakcio

A faktorokat az fa() funkcioval fogjuk kinyerni. Ez a funkcio tobb faktorextrakcios modszert is kinal. A leggyakrabban hasznalt modszer a **Maximum Likelihood Estimation** (mle) akkor ha a megfigyelt valtozok megfelelnek a tobbvaltozos normalitas feltetelenek, mig a **Principal Axis Factoring** (paf) a preferalt modszer akkor, ha a valtozok nem mutatnak tobbvaltozos normalis eloszlast.

Az mvn() funkcio az MVN package-bol es a mvnorm.kur.test() es a mvnorm.skew.test() funkciok az ICS package-bol segithet eldonteni, hogy tobbvaltozos normalis eloszlast mutatnak-e az adatok. Ha ezeknek a teszteknek a p-erteke alacsonyabb 0.05-nel, akkor az a tobbvaltozos normalitas serulesere utal.

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

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

```
mvnorm.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
```

```
mvnorm.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 lathato hogy mind a Henze-Zirkler teszt mind a tobbvaltozos ferdeseg es csucsossag tesztek a normalitas feltetelenek serulesere utal. Igy a paf extrakcios modszert hasznaljuk majd.

A faktorextrakciora a psych package fa() funkciojat hasznaljuk. Ezen belul megadhatjuk a faktoreztrakcios modszert az fm = parameteren belul. Itt fm = “pa”-t hatarozunk meg, mert a paf modszert szeretnenk hasznalni, de ha a tobbvaltozos normalitas nem serult volna, akkor ehelzett “mle”-t hasznaltunk volna. Az alabbi peldaban meg nem akartam faktorforgatast alkalmazni, hogy lepesrol lepesre tudjam bemutatni a faktorelemzes modszeret, igy a rotate = erteket “none”-ra allittam, de altalaban a faktorokat egybol el is forgatjuk valamelyik modszerrel (lasd alabb). Az nfactors = parameterrel adhatjuk meg, hany faktort szeretnenk kinyerni. Egyelore allitsuk ezt 5-re, lentebb tartgyaljuk majd, hogyan valasztjuk ki az idealis faktormennyiseget.

A modell objektum $communality komponenseben talaljuk a valtozokhoz tartozo kommunalitas ertekekt. Ezt legmagasabbtol legalacsonyabbig sorbarendezzuk es kilistazzuk. Ahogy fentebb emlitettuk a kommunalitas azt jelzi, hogy az egyes megifigyelt valtozokban tapasztalhato variancia mekkora hanyadat magyarazzak a kinyert faktorok. Az output azt mutatja, hogy a Q17 “Altalaban nem szeretek viccelodni, vagy masokat szorakoztatni” (“I usually don’t like to tell jokes or amuse people.”) a legjobban reprezentalt item az 5 faktoros strukturaban, aminek 68%-at kepesek megmagyarazni az uj faktorok. Ezzel szemben a Q22 “Amikor szomoru vagy ideges vagyok altalaban elvesztem a humorerzekemet” (“If I am feeling sad or upset, I usually lose my sense of humor.”) a legkevesbe reprezentalt item, varianciajanak csak 25%-at magyarazza a jelenlegi faktorstruktura.

Neha ahhoz hogy a faktorstruktura jol mukodjon, erdemes a rosszul reprezentalt itemeket kizarni. Ez foleg akkor fontos, ha kicsi a mintaelemszam. Ha a megfigyelesek szama 250 alatti, akkor MacCallum et al. szerint elvarhato hogy az itemek atlagos kommunalitasa legalabb 0.6 legyen. A mi esetunkben ennel megengedobbek is lehetunk, mert az elemszamunk nagyobb, de egy melyebb faktorelemzes eseten igy is erdemes lehet elgondolkodni a rosszul reprezentalt itemek kizarasan.

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

```r
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                                     0.6831095
## Q25                                     0.6735019
## Q20                                     0.6686710
## Q21                                     0.6295625
## Q8                                      0.6199212
## Q10                                     0.6152281
## Q18                                     0.5979830
## Q15                                     0.5872814
## Q13                                     0.5586916
## Q14                                     0.5562796
## Q1                                      0.5460518
## Q26                                     0.5392991
## Q32                                     0.5340582
## Q31                                     0.5077178
## Q12                                     0.4800099
## Q19                                     0.4783614
## Q5                                      0.4781500
## Q2                                      0.4558917
## Q16                                     0.4389957
## Q4                                      0.4369528
## Q11                                     0.4056135
## Q6                                      0.4047964
## Q29                                     0.3986026
## Q3                                      0.3968426
## Q7                                      0.3833806
## Q23                                     0.3600013
## Q27                                     0.3476014
## Q24                                     0.3311687
## Q9                                      0.3129809
## Q30                                     0.2661797
## Q28                                     0.2616243
## Q22                                     0.2548062
```

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

```
## [1] 0.4752911
```

4.4 Idealis faktorszam kivalasztasa

A fokomponenselemzeshez hasonloan meg kell hataroznunk, hany faktort szeretnenk kinyerni az adatokbol. Ahogy azt a fokomponenselemzesnel is lattuk, ennek az eldontesere hasznalhatjuk a scree-tesztet, a Kaiser-Guttman kriteriumot, vagy a parallel tesztet. Ezen felul a psych pacakge ket ujabb modszert is felkinal a donteshozas elosegitesere: a very simple structure (VSS) kriteriumot, es a Wayne Velicer's Minimum Average Partial (MAP) kriteriumot. (A vss() funkcio a psych package-ben)

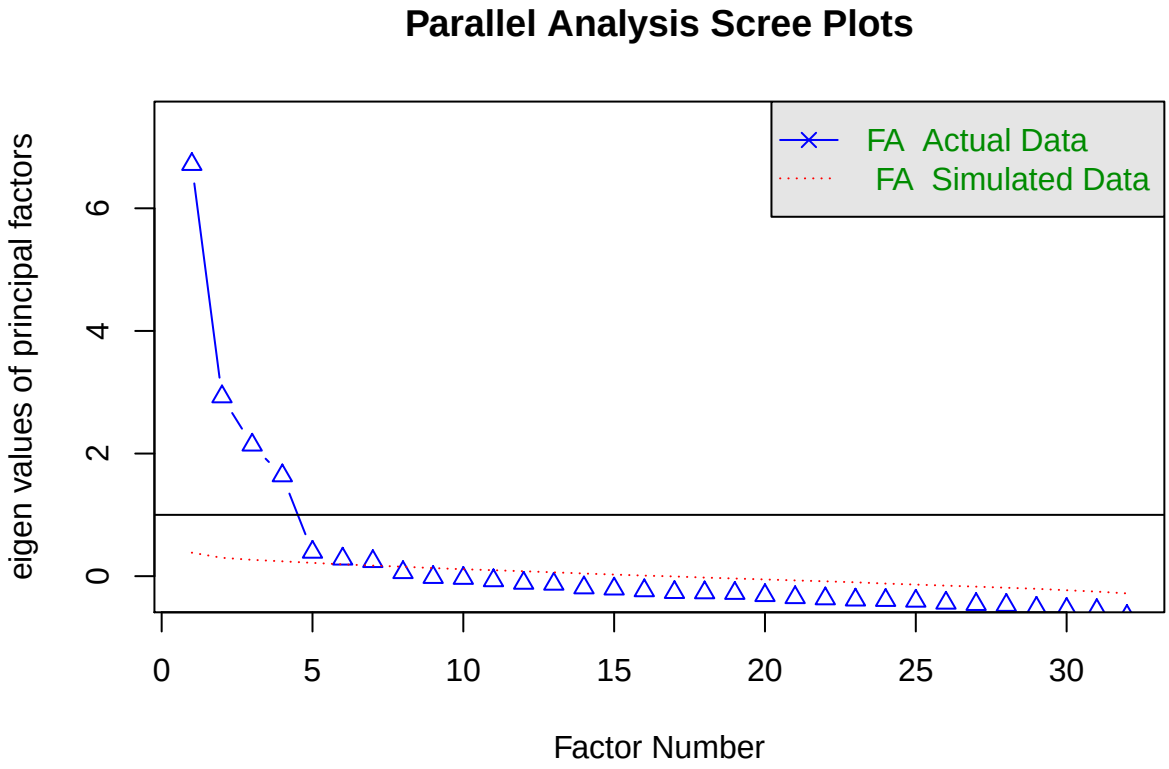
Az alabbi peldaban a psych package fa.parallel funciojat es az nfactors funkciot hasznaljuk arra, hogy a kulonbozo kriteriumok szerint eldonthessuk, hany faktor megtartasa lenne idealis.

A kulonbozo technikak altal javasolt idealis faktorszamok a kovetkezok:

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

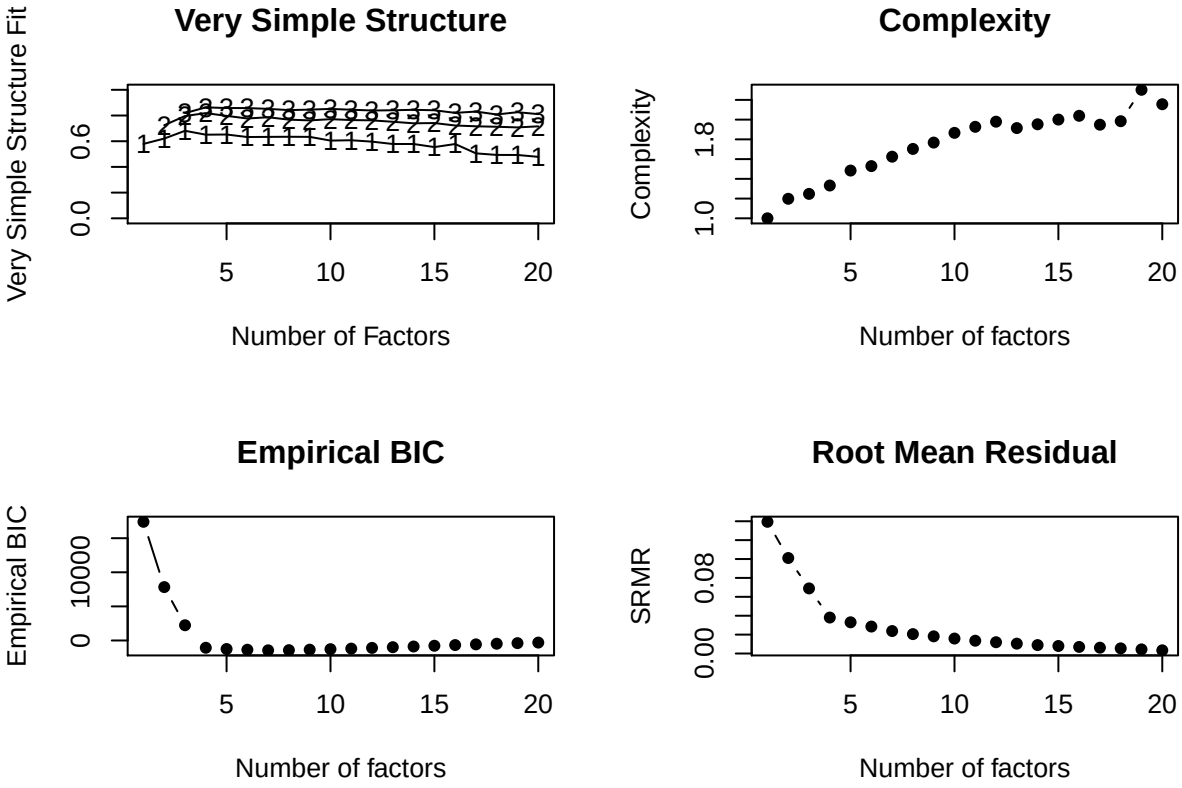
Ezek alapjan ugy tunik, a legtobb technika szerint 4 latens faktor irja le az adatok variabilitasat a legjobban. Alabb meg is epitjuk ezt a 4-faktoros modellt, es megvizsgaljuk a kommunalitas-tablazatot. A faktorelemzes soran nagyon gyakori, hogy a folyamatot ujra es ujra megismeteljuk kulonbozo bemeneti valtozokkal es kulonozo faktorszamokkal es rotacios modszerekkel, amig elerjuk a veglegesnek tekintheto faktorstrukturat. A vegleges faktorstruktura idealis esetben jol ertelmezheto a faktorok es a hozzajuk tartozo valtozo-toltesek alapjan.

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




```
## Parallel analysis suggests that the number of factors =  7  and the number of components =  NA
```

```
nfactors(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 maximimum of 0.68  with  3  factors
## VSS complexity 2 achieves a maximimum of 0.82  with  4  factors
## The Velicer MAP achieves a minimum of 0.01  with  4  factors
## Empirical BIC achieves a minimum of  -1443.27  with  7  factors
## Sample Size adjusted BIC achieves a minimum of  -211.42  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  9361  0.0e+00    39.8 0.58 0.134 6124  7598     1.0
## 2  0.62 0.72 0.026 433  6451  0.0e+00    26.0 0.72 0.114 3430  4806     1.2
## 3  0.68 0.80 0.019 403  4403  0.0e+00    17.1 0.82 0.096 1592  2872     1.2
## 4  0.65 0.82 0.011 374  2396 2.6e-291    11.6 0.88 0.071 -213   975     1.3
## 5  0.65 0.80 0.012 346  2068 1.7e-242    10.4 0.89 0.068 -346   753     1.5
## 6  0.63 0.78 0.013 319  1712 6.2e-189     9.4 0.90 0.064 -514   500     1.5
## 7  0.63 0.79 0.014 293  1337 8.2e-133     8.5 0.91 0.058 -707   223     1.6
## 8  0.63 0.77 0.016 268  1126 2.1e-105     7.9 0.92 0.055 -744   107     1.7
## 9  0.63 0.76 0.017 244   902  3.1e-76     7.4 0.92 0.050 -800   -25     1.8
## 10 0.60 0.77 0.020 221   734  2.7e-56     6.9 0.93 0.047 -808  -106     1.9
## 11 0.61 0.76 0.022 199   595  4.6e-41     6.5 0.93 0.043 -793  -161     1.9
## 12 0.60 0.76 0.025 178   470  3.3e-28     6.3 0.93 0.039 -772  -206     2.0
## 13 0.58 0.75 0.028 158   397  1.3e-22     5.9 0.94 0.038 -705  -203     1.9
## 14 0.58 0.74 0.032 139   317  6.6e-16     5.6 0.94 0.035 -653  -211     2.0
```

```
## 15 0.55 0.74 0.036 121   281  1.0e-14     5.3 0.94 0.035 -563  -179     2.0
## 16 0.58 0.72 0.040 104   213  1.8e-09     5.2 0.95 0.031 -513  -183     2.0
## 17 0.51 0.72 0.046  88   171  2.8e-07     4.9 0.95 0.030 -443  -163     1.9
## 18 0.49 0.71 0.052  73   133  2.5e-05     4.8 0.95 0.028 -377  -145     2.0
## 19 0.49 0.71 0.059  59    78  5.1e-02     4.4 0.95 0.017 -334  -146     2.3
## 20 0.48 0.72 0.068  46    51  2.7e-01     4.1 0.96 0.010 -270  -123     2.2
##    eChisq   SRMR eCRMS  eBIC
## 1   20643 0.1394 0.144 17406
## 2   10851 0.1011 0.108  7830
## 3    5044 0.0689 0.076  2233
## 4    1539 0.0381 0.044 -1070
## 5    1159 0.0330 0.040 -1254
## 6     866 0.0285 0.036 -1360
## 7     601 0.0238 0.031 -1443
## 8     449 0.0205 0.028 -1421
## 9     351 0.0182 0.026 -1351
## 10    266 0.0158 0.024 -1276
## 11    195 0.0135 0.021 -1193
## 12    152 0.0120 0.020 -1090
## 13    116 0.0104 0.019  -986
## 14     85 0.0089 0.017  -885
## 15     67 0.0080 0.016  -777
## 16     52 0.0070 0.015  -674
## 17     41 0.0062 0.015  -573
## 18     31 0.0054 0.014  -478
## 19     20 0.0044 0.013  -391
## 20     12 0.0034 0.011  -309
```

```r
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)
## Q17                                     0.6849707
## Q20                                     0.6693961
## Q25                                     0.6673788
## Q21                                     0.6258319
## Q8                                      0.6221044
## Q10                                     0.6182339
## Q18                                     0.5993342
## Q14                                     0.5485758
## Q1                                      0.5479302
## Q13                                     0.5457372
## Q26                                     0.5417358
## Q32                                     0.5372812
## Q15                                     0.5189649
## Q31                                     0.5091859
## Q12                                     0.4810001
## Q5                                      0.4530531
## Q2                                      0.4364573
## Q4                                      0.4315326
## Q3                                      0.4003995
## Q29                                     0.3980778
## Q7                                      0.3828435
```

```
## Q16                                    0.3756515
## Q19                                    0.3686066
## Q6                                     0.3584248
## Q27                                    0.3444689
## Q11                                    0.3293748
## Q9                                     0.3062011
## Q23                                    0.2801839
## Q24                                    0.2796452
## Q30                                    0.2484149
## Q28                                    0.2312722
## Q22                                    0.1850645
```

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

```
## [1] 0.4539792
```

4.5 Faktorforgatas

A faktorforgatas celja hogy megkonnyitse a faktorok ertelmezeset. Igy elkerulheto hogy az egesz faktorstruktura 1 vagy ket nagyon dominans faktorbol alljon, amire angyon erosek a toltesek, mig a tobbi faktor ertelmezese kodos. A faktorforgatas soran az eredeti valtozok ugyan ott maradnak a "faktorterben", viszont a faktorok dimenzio tengelyeit elforgatjuk, hogy jobban railleszkedjenek egyes valtozocsoportokra.

A faktorforgatasnak szamos modszere ismert, de ezek ket fo csoportba sorolhatok: ortogonalis es oblique modszerek koze. Az ortogonalis modszerek (mint pl. Quartimax, Equimax, vagy a pszichologiaban leggyakrabban hasznalt **Varimax** modszer) soran a faktor dimenziok egymasra merolegesek maradnak (ez azt jelenti hogy egymassal nem korrelalnak majd a vegso faktorok). Az oblique modszerek (mint pl. **Direct Oblimin** vagy a Promax) eseten viszont megengedett hogy a vegso faktorok valamellyest korrelaljanak egymassal. Az exploratoros faktorelemzes soran tobb modszert is kiprobalhatunk, de itt fontos az elmeleti megalapozottsag is. Elkepzelheto hogy a faktorok korrelaljanak egymassal? Ha igen, akkor az oblique modszerekre erdemes hagyatkozni. (Altalaban a korrelalatlan faktorokat konnyebb ertelmezni).

Az alapertelmezett faktorforgatasi modszer a Direct Oblimin ("oblimin"). Probaljuk ki a Promax ("promax) es a Varimax ("varimax") modszereket 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")
```

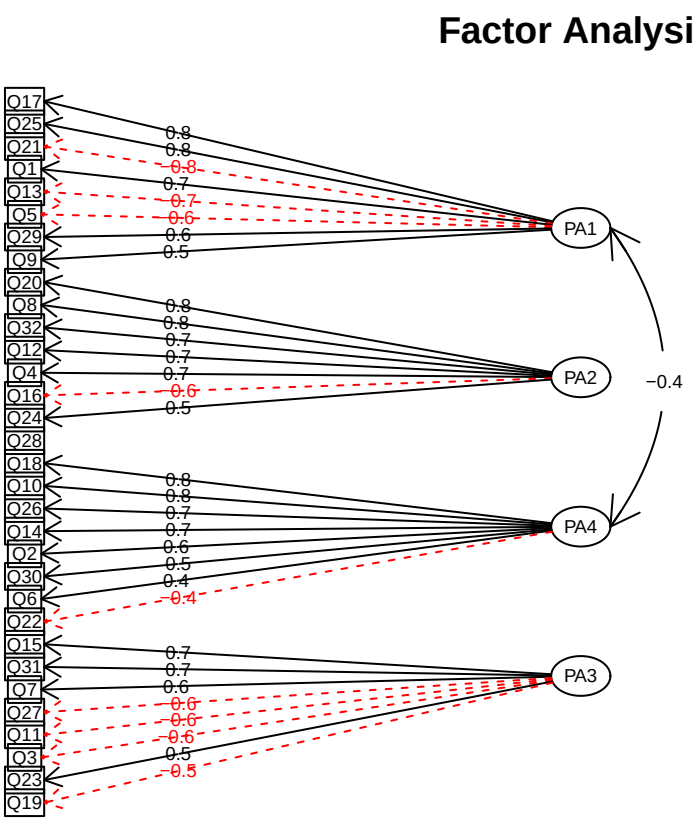
4.6 Faktorok interpretacioja

A faktorok ertelmezese nem konnyu feladat. Sok teruletspecifikus tudasra van szuseg a helyes faktrostruktura kivalasztasahoz es a helyes faktorertelmezeshez. Itt ezert csak a kulonbozo vizualizacios modszereket mutatjuk be amik segithetnek a faktorok ertelmezeseben.

Az fa.diagram() funkcio kirajzolja a model objektum alapjan a faktorstrukturat, es azt, hogy melyik valtozo melyik faktorra mutatja a legnagyobb faktortoltest (melyik faktorral a legnagyobb a korrelacioja). Az abran lathatoak az egyes korrelacios egyutthatok is. A fekete nyilak pozitiv, mig a piros nyilak negativ korrelaciokat jeleznek.

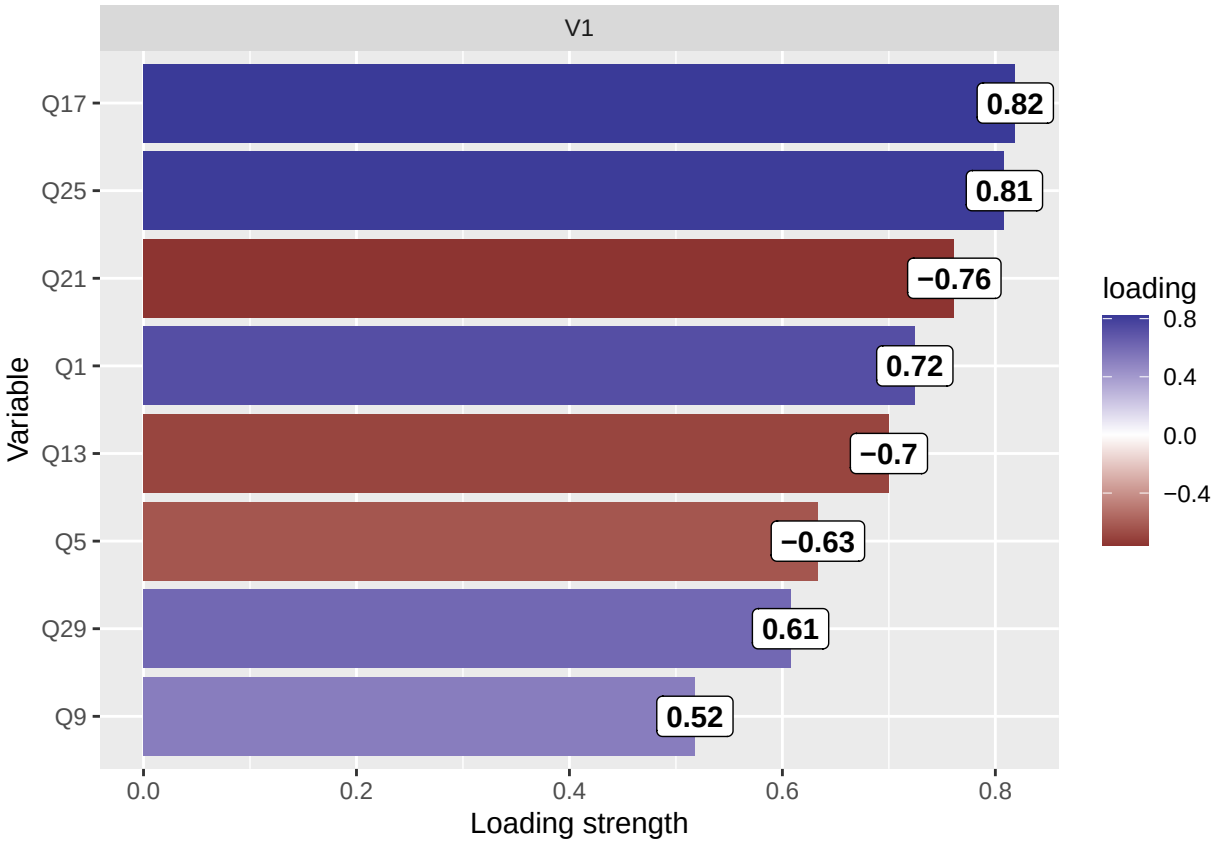
Tovabbi segitseget nyujthat a sajat funkcio amit a fokomponsneselemzesnel is hasznaltunk: fviz_loadnings_with_cor(). Itt a fa() modelek eseten megadhatjuk a loading_above = parameterst is, ahol specifikalhatjuk, hogy csak a bizonyos abszolut faktortoltes (korrelacio) feletti megfigyelt valtozokat abrazoljuk. Ez megkonnyitheti az abra atlathatosagat.

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

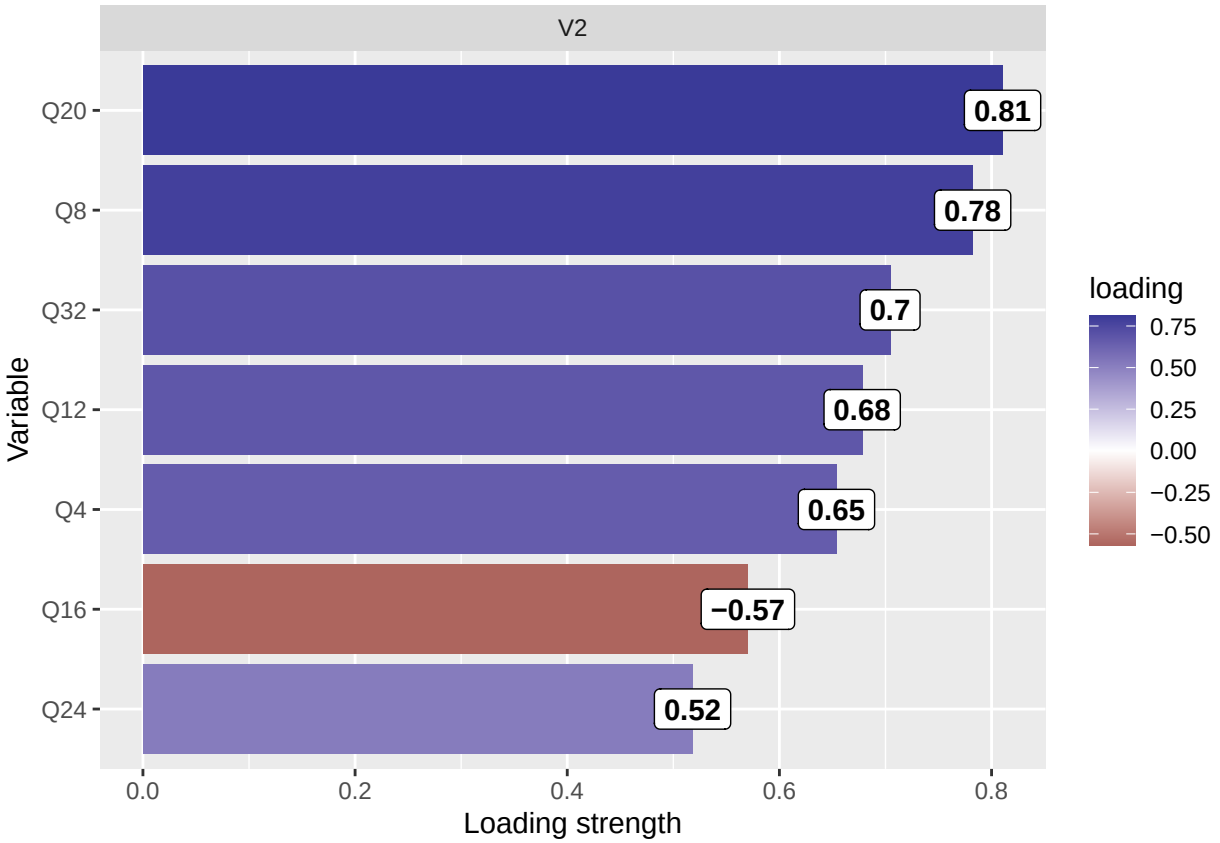



```
fviz_loadnings_with_cor(EFA_mod2, axes = 1, loadings_above = 0.4)
```

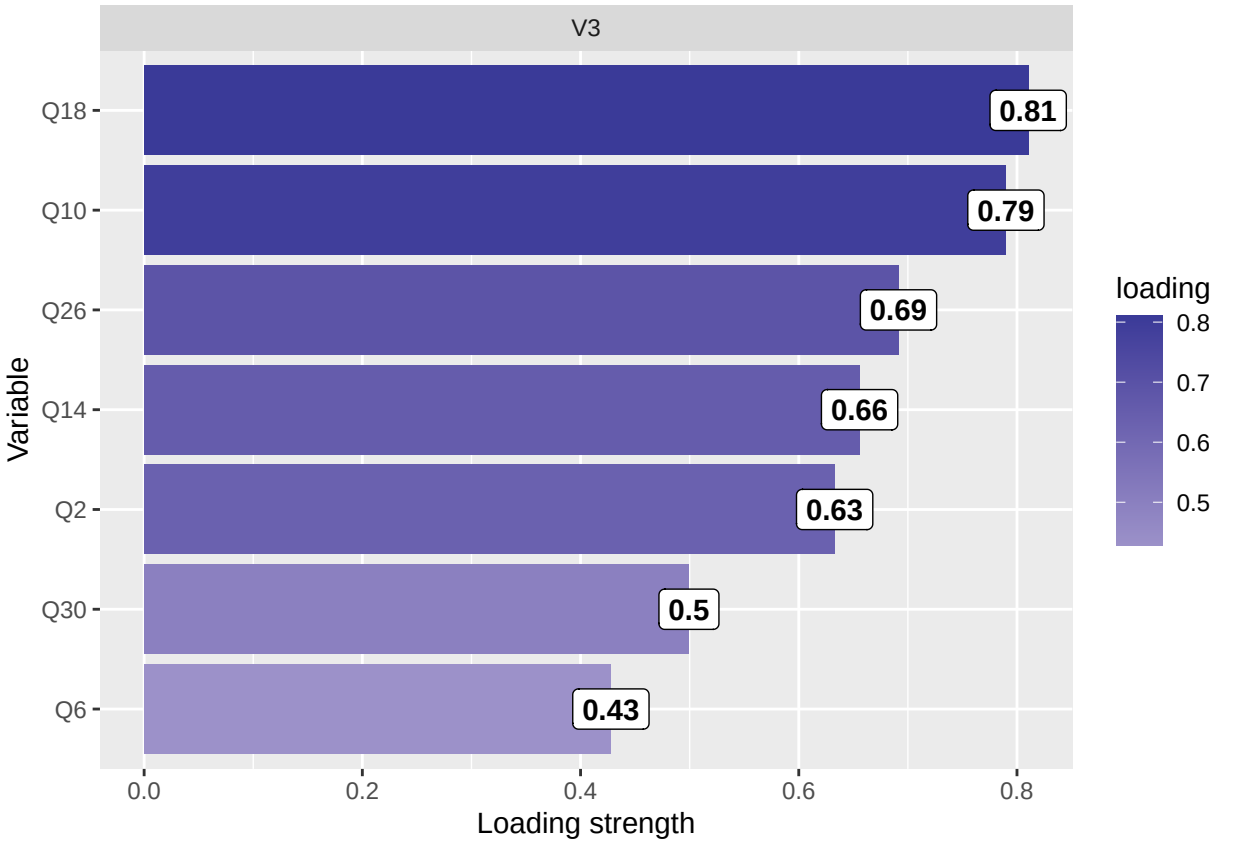
```
## Warning: `as_tibble.matrix()` requires a matrix with column names or a `.name_repair` argument. Using
## This warning is displayed once per session.
```



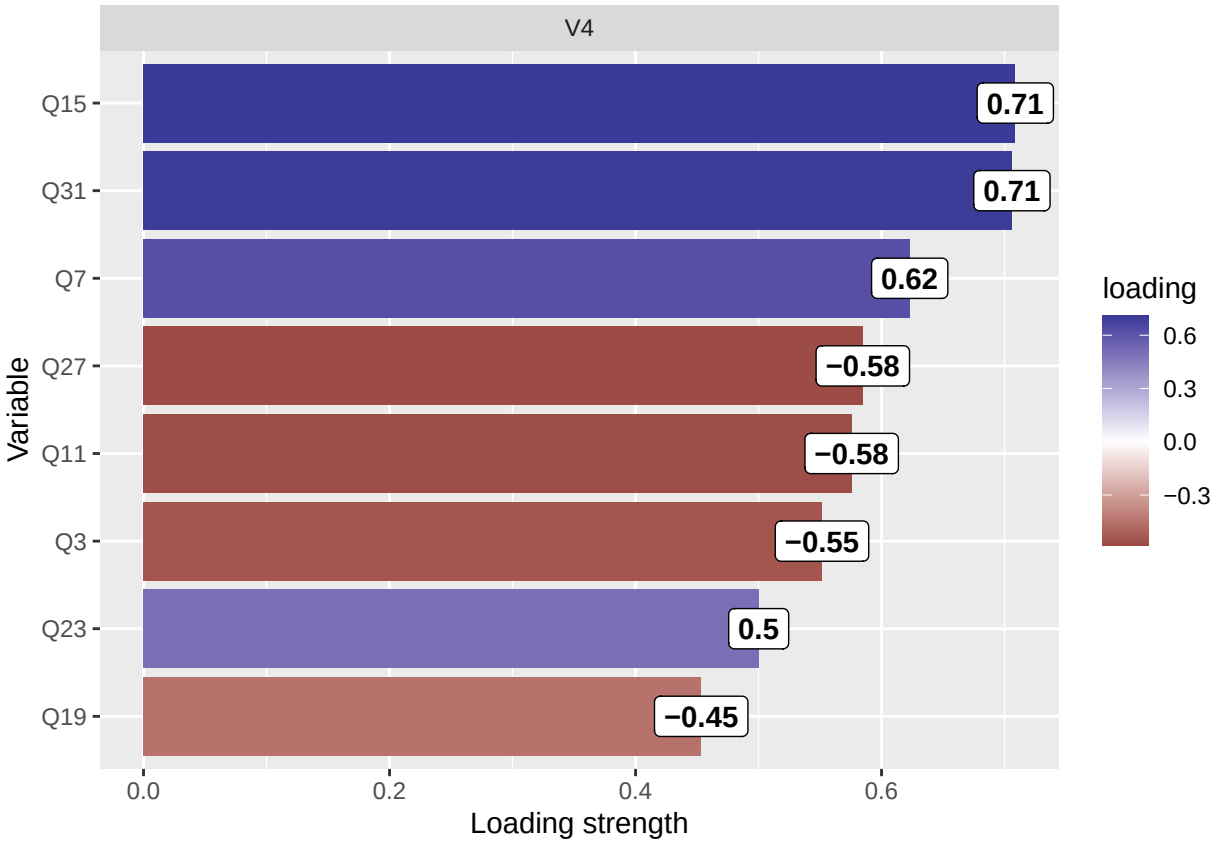

```
fviz_loadnings_with_cor(EFA_mod2, axes = 2, loadings_above = 0.4)
```




```
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 technikakat a Big Five Inventory (bfi) adatbazison gyakorolhatod. Ez a psych package-be beepitett adatbazis, ami 2800 szemely valaszait tartalmazza a Big Five szemelyisegkerdoiv kerdeseira. Az elso 25 oszlop a kerdoiv kerdeseire adott valaszokat tartalmazza, az utolso harom oszlop (gender, education, es age) pedig demografiai kerdeseket tartalmaz. A reszleteket az egyes itemekhez tartozo kerdesekrol es a valaszok kodolasarol elolvashatod ha lefuttatod a ?bfi parancsot.

Az adatbazist betoltheted a kovetkezo parancsokkal.

```
`?`(bfi)

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

Ebben a feladatban csak az elso 25 oszlopot hasznald, az eredeti kerdoiv kerdeseit. Vegezz el feltaro faktorelemzest, es ez alapjan hatarozd meg, hany faktor megtartasa az idealis, mely faktorokra mely itemek toltenek leginkabb, es ez alapjan hogyan nevezned el a faktorokat. Melyek a faktorstruktura altal leginkabb es a legkevesbe reprezentalt itemek?