PSZB17-210 - Seminar_4

Zoltan Kekecs

September 28, 2019

4. Ora - Adatexploracio

Az ora celja az adatexploracios modszerek elsajatitasa.

Package-ek betoltese

A kovetkezp package-ekre lesz szuksegunk

```
if (!require("gridExtra")) install.packages("gridExtra")
library(gridExtra) # for grid.arrange
if (!require("tidyverse")) install.packages("tidyverse")
library(tidyverse) # for dplyr and ggplot2
if (!require("psych")) install.packages("psych")
library(psych) # for describe
```

Adatok betoltese

Beolvassuk az orai kerdoiv adatait a read_csv funkcioval, es elmentjuk egy orai_adat nevu objektumba. A read_csv() funkcio a tidyverse resze, es egybol tibble formatumban menti el az adatainkat

```
orai_adat <- read_csv("https://tinyurl.com/statgyak2019-seminar-4-data")
```

Adatok attekintese

Mindig erdemes azzal kezdeni, hogy **megismerkedunk az adat** szerkezetevel es tartalmaval.

A **tibble objektum** meghivasaval kapthatunk nemi informaciot az adattabla szerkezeterol. Lathatjuk hany sor es hany oszlop van az adattablaban, es lathatjuk milyen class-ba tartoznak (chr, dbl ...)

```
orai_adat
```

```
## # A tibble: 25 x 31
##    kitoltesiido ID    nem   eletkor alvas_szukseges nemalvas_oka memoria
##    <chr>        <chr> <chr> <chr>             <dbl> <chr>          <dbl>
##  1 9/11/2019 1~ Süni  No    0-22                  8 Korán kelés~       7
##  2 9/11/2019 1~ :)    No    0-22                 10 stressz            8
##  3 9/11/2019 1~ must~ No    0-22                  8 nem tudok e~       7
##  4 9/11/2019 1~ alma  No    0-22                  9 Álmatlanság        7
##  5 9/11/2019 1~ pupu  No    23+                   9 szociális m~       7
##  6 9/11/2019 1~ alma  No    0-22                  8 tanulás, st~       8
##  7 9/11/2019 1~ Nind~ No    0-22                  7 Chatelés           8
##  8 9/11/2019 1~ brk   No    0-22                  8 túl sok pro~       9
##  9 9/11/2019 1~ ___   No    0-22                  8 inszomnia, ~      10
## 10 9/11/2019 1~ csoki No    0-22                  6 stressz            8
## # ... with 15 more rows, and 24 more variables: memoria_ront_A <chr>,
## #   memoria_ront_B <chr>, memoria_ront_C <chr>, memoria_ront_D <chr>,
## #   memoria_ront_E <chr>, memoria_ront_F <chr>, jegy_tipp <dbl>,
## #   valasztott_szam <dbl>, tippelt_gyakori_szam <dbl>, magassag <dbl>,
## #   cipomeret <dbl>, memoriateszt_hely <chr>, alvas_tegnap_1 <dbl>,
```

```
## #   alvas_tegnap_3 <dbl>, alvas_altalaban_1 <dbl>,
## #   alvas_altalaban_3 <dbl>, energiaszint_1 <dbl>, energiaszint_3 <dbl>,
## #   szocmedia_1 <dbl>, szocmedia_3 <chr>, hangulat_1 <dbl>,
## #   hangulat_3 <dbl>, stat_tudasszint_3 <dbl>, stat_gyakorlas_3 <dbl>
```

Ha az egyes valtozok **leirostatisztikaira** (descriptive statistics) vagyunk kivancsiak, kerhetjuk ezt a mar tanult modon.

Peldaul lekerhetjuk a valtozo alapveto legalacsonyabb es legmagasabb erteket, atlagat, medianjat, a kvartiliseket, es hogy hany hianyzo adat van (ha van) a **summary()** funkcioval (miutan a select funkcioval kivalasztottuk, melyik valtozora vagyunk kivancsiak)

```
orai_adat %>%
  select(jegy_tipp) %>%
  summary()
```

```
##    jegy_tipp
##  Min.   :3.00
##  1st Qu.:4.00
##  Median :4.00
##  Mean   :4.04
##  3rd Qu.:5.00
##  Max.   :5.00
```

Vagy megkapthatjuk ugyanezt az osszes valtozora, ha ugyanezt az egesz adattablara futtatjuk le. Persze a karakter osztalyba tartozo valtozoknal mindezeknek a leiro statisztikaknak nincs ertelme, ott csak a class informaciot kaptjuk az output-ban.

```
orai_adat %>%
  summary()
```

Gyakorlas

- Hany orat gyakorolt az az ember, aki a legtobb orat gyakorolta a masodik es a harmadik ora kozott (*stat_gyakorlas_3* valtozo)?
- Mekkora volt az atlagos energiaszint az elso es mekkora a haramdik oran (*energiaszint_1* es *energiaszint_3* valtozok)?

Megtobb leiro statisztika

A **Psych** package segitsegevel a **describe()** funkcio megtobb hasznos informaciot adhat

```
describe(orai_adat)
```

```
##                     vars  n   mean    sd median trimmed  mad   min  max
## kitoltesiido*          1 25    NaN    NA     NA     NaN   NA   Inf -Inf
## ID*                    2 25  32.00    NA     32   32.00 0.00 32.00   32
## nem*                   3 25    NaN    NA     NA     NaN   NA   Inf -Inf
## eletkor*               4 25    NaN    NA     NA     NaN   NA   Inf -Inf
## alvas_szukseges        5 25   8.32  0.90      8    8.33 1.48  6.00   10
## nemalvas_oka*          6 24    NaN    NA     NA     NaN   NA   Inf -Inf
## memoria                7 25   7.48  1.26      8    7.57 1.48  4.00   10
## memoria_ront_A*        8 23    NaN    NA     NA     NaN   NA   Inf -Inf
## memoria_ront_B*        9 19    NaN    NA     NA     NaN   NA   Inf -Inf
## memoria_ront_C*       10 19    NaN    NA     NA     NaN   NA   Inf -Inf
```

```
## memoria_ront_D*        11 14    NaN    NA     NA     NaN   NA   Inf -Inf
## memoria_ront_E*        12  7    NaN    NA     NA     NaN   NA   Inf -Inf
## memoria_ront_F*        13  4    NaN    NA     NA     NaN   NA   Inf -Inf
## jegy_tipp              14 25   4.04  0.73      4    4.05 1.48  3.00    5
## valasztott_szam        15 25   6.56  2.52      7    6.62 2.97  2.00   10
## tippelt_gyakori_szam   16 25   5.80  2.58      6    5.76 2.97  2.00   10
## magassag               17 25 161.07 34.32    168  166.62 8.90  1.82  190
## cipomeret              18 25  38.94  3.34     38   38.60 2.97 35.00   47
## memoriateszt_hely*     19 25    NaN    NA     NA     NaN   NA   Inf -Inf
## alvas_tegnap_1         20 25   7.92  1.26      8    7.90 1.48  6.00   10
## alvas_tegnap_3         21 23   7.61  1.03      8    7.68 1.48  5.00    9
## alvas_altalaban_1      22 25   7.04  1.02      7    7.10 1.48  5.00    9
## alvas_altalaban_3      23 23   7.13  0.63      7    7.16 0.00  6.00    8
## energiaszint_1         24 25   5.48  2.16      5    5.62 2.97  1.00    8
## energiaszint_3         25 23   6.17  1.92      7    6.26 1.48  3.00    9
## szocmedia_1            26 25   1.96  1.57      1    1.81 1.48  0.00    6
## szocmedia_3*           27 24   1.96  1.36      2    1.79 1.48  0.00    5
## hangulat_1             28 25   6.12  1.79      6    6.14 1.48  3.00    9
## hangulat_3             29 23   5.91  2.21      6    6.16 2.97  1.00    9
## stat_tudasszint_3      30 23   4.09  2.02      5    4.16 1.48  0.00    8
## stat_gyakorlas_3       31 23   4.59  3.54      3    4.08 2.22  0.00   14
##                       range  skew kurtosis   se
## kitoltesiido*          -Inf    NA       NA   NA
## ID*                    0.00    NA       NA   NA
## nem*                   -Inf    NA       NA   NA
## eletkor*               -Inf    NA       NA   NA
## alvas_szukseges        4.00 -0.30     0.22 0.18
## nemalvas_oka*          -Inf    NA       NA   NA
## memoria                6.00 -0.67     0.89 0.25
## memoria_ront_A*        -Inf    NA       NA   NA
## memoria_ront_B*        -Inf    NA       NA   NA
## memoria_ront_C*        -Inf    NA       NA   NA
## memoria_ront_D*        -Inf    NA       NA   NA
## memoria_ront_E*        -Inf    NA       NA   NA
## memoria_ront_F*        -Inf    NA       NA   NA
## jegy_tipp              2.00 -0.06    -1.22 0.15
## valasztott_szam        8.00 -0.38    -1.22 0.50
## tippelt_gyakori_szam   8.00  0.11    -1.16 0.52
## magassag             188.18 -3.95    15.59 6.86
## cipomeret             12.00  0.99    -0.05 0.67
## memoriateszt_hely*     -Inf    NA       NA   NA
## alvas_tegnap_1         4.00 -0.22    -1.12 0.25
## alvas_tegnap_3         4.00 -0.63    -0.11 0.22
## alvas_altalaban_1      4.00 -0.30    -0.71 0.20
## alvas_altalaban_3      2.00 -0.07    -0.63 0.13
## energiaszint_1         7.00 -0.27    -1.21 0.43
## energiaszint_3         6.00 -0.38    -1.35 0.40
## szocmedia_1            6.00  1.06     0.20 0.31
## szocmedia_3*           5.00  1.00    -0.11 0.28
## hangulat_1             6.00 -0.17    -0.98 0.36
## hangulat_3             8.00 -0.74    -0.30 0.46
## stat_tudasszint_3      8.00 -0.18    -0.99 0.42
## stat_gyakorlas_3      14.00  1.26     0.88 0.74
```

Gyakorlas

- Mi a statisztika gyakorlas ferdesegi mutatoja (skew/skewness) (*stat_gyakorlas_3* valtozo)?
- Hanyan valaszoltak az energia szint kerdesre az elso es a harmadik oran (*energiaszint_1* es *energiaszint_3* valtozok)?

Faktorok

Nehany karaktervaltozonak csak **korlatozott mennyisegu eleme** lehet, mint peldaul a nem (ferfi, no, egyeb), vagy az eletkor (0-22 vagy 23+) a mi kerdoivunkben. Ezeket megjelolhetjuk faktor (factor) osztalyu valtozokent, es akkor az R tobb informaciot fog adni rola.

A **levels()** funkcio megmutatja mik a faktorunk szintjei, de lathato ez akkor is ha csak meghivjuk a valtozot magat.

A **table()** funkcio pedig tablazatot keszit arrol, hogy az egyes csoportokban hany megfigyeles talalhato

```
orai_adat <- orai_adat %>%
              mutate(nem = factor(nem),
                     eletkor = factor(eletkor),
                     memoriateszt_hely = factor(memoriateszt_hely))

levels(orai_adat$nem)
```

```
## [1] "Férfi" "No"
```

```
orai_adat$nem
```

```
##  [1] No    No    No    No    No    No    No    No    No    No    No
## [12] No    No    No    No    No    No    No    No    No    Férfi No
## [23] No    Férfi Férfi
## Levels: Férfi No
```

```
table(orai_adat$eletkor)
```

```
##
## 0-22  23+
##   22    3
```

```
table(orai_adat$nem)
```

```
##
## Férfi    No
##     3    22
```

Igy mar a fenti **summary()** funkcio is kiadja az **egyes faktorszintekrol** (no, ferfi) hogy hanyan tartoznak oda.

```
orai_adat %>%
  select(nem) %>%
  summary()
```

```
##     nem
##  Férfi: 3
##  No   :22
```

Van, hogy szeretnenk **kizarni** bizonyos **faktorszinteket** az elemzesbol. Pl. ha valamelyik faktor szintbol nagyon keves megfigyeles van, mondjuk 23 even feluli eletkoruakbol, oket lehet hogy szeretnenk kizarni a kesobbi elemzesekbol hogy egyszerusitsuk az eredmenyeink ertelmezeset. Ezt a mar korabban tanult **filter()** funkcio segitsegevel konnyeden megtehetjuk, azonban arra figyelnunk kell, hogy az R megjegyzi a faktorszinteket, es azt azt kovetoen is a **valtozohoz rendelve tartja**, miutan mar az adott faktorszintbol nincs egy megifgyeles sem az adattablaban.

```
orai_adat %>%
  filter(eletkor != "23+") %>%
  select(eletkor) %>%
  summary()
```

```
##  eletkor
##  0-22:22
##  23+ : 0
```

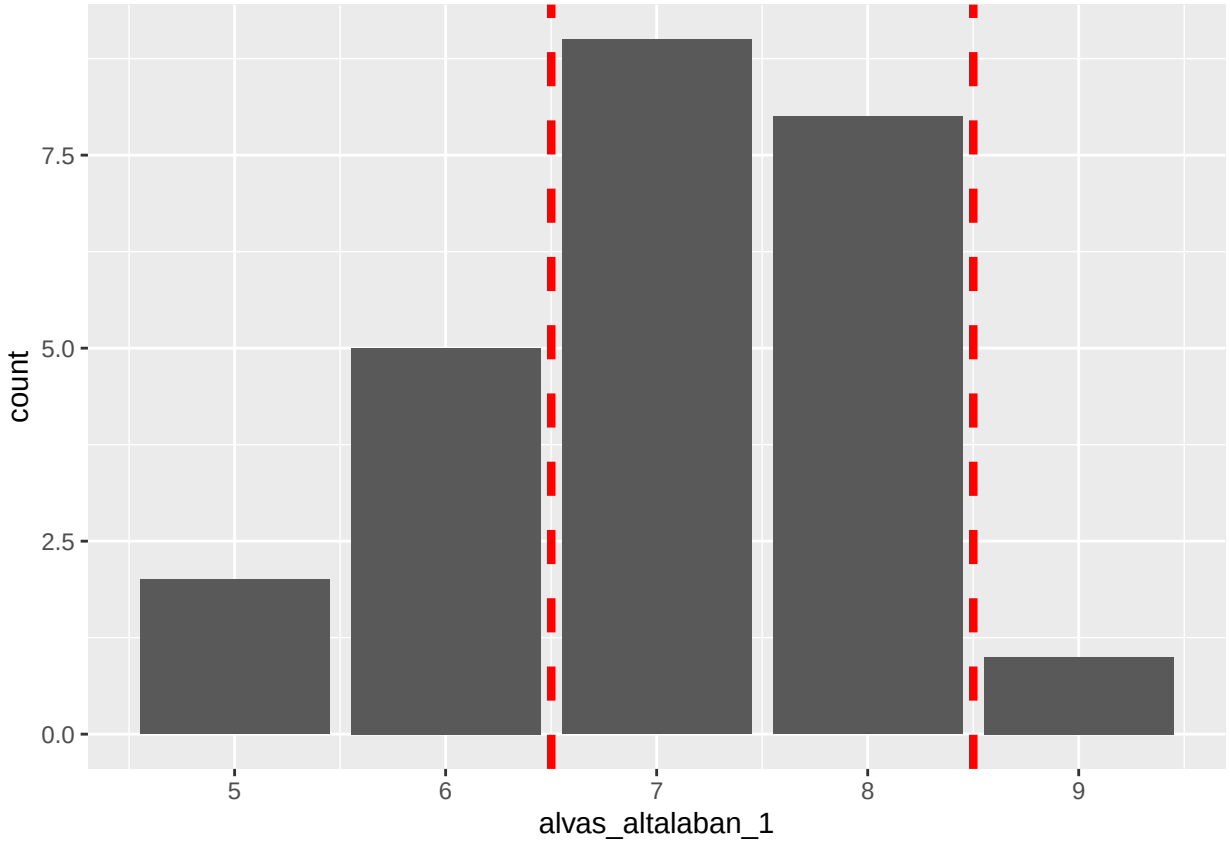
Igy ezeket a szinteket ejteni szoktuk a **droplevels()** funkcioval.

```
orai_adat %>%
  filter(eletkor != "23+") %>%
  select(eletkor) %>%
  droplevels() %>%
  summary()
```

```
##  eletkor
##  0-22:22
```

Elofordul, hogy egy **numerikus valtozot akarunk atalakitani faktorra**, pl. elkepzelheto hogy ossze akarjuk hasonlitani azokat akik 6 vagy kevesebb orat alszanak altalaban azokkal akik 7 vagy 8, vagy meg tobb orat alszanak.

```
orai_adat %>%
  ggplot() +
  aes(x = alvas_altalaban_1) +
  geom_bar() +
  geom_vline(xintercept = c(6.5, 8.5), linetype="dashed",
                color = "red", size=1.5)
```



Ilyenkor hasznalhatjuk a **mutate()** es **recode()** funkciok kombinaciojat hogy csinaljunk egy uj valtozot. Ebbe a kodba beleepitettem a **factor()** funkciot is, hogy azonnal meghatarozzuk, hogy ez az uj valtozo egy faktor, es nem egy egyszeru karaktervektor. A factor() funkcio nelkul is lefut a kod, de akkor meg kellene egy kulon sor ahol megadjuk hogy ez egy faktorvaltozo.

```
orai_adat = orai_adat %>%
  mutate(alvas_altalaban_kat_1 = factor(
                                        recode(
                                          alvas_altalaban_1,
                                          "5" = "keves",
                                          "6" = "keves",
                                          "7" = "eleg",
                                          "8" = "eleg",
                                          "9" = "sok"
                                          )
                                        ))


levels(orai_adat$alvas_altalaban_kat_1)
```

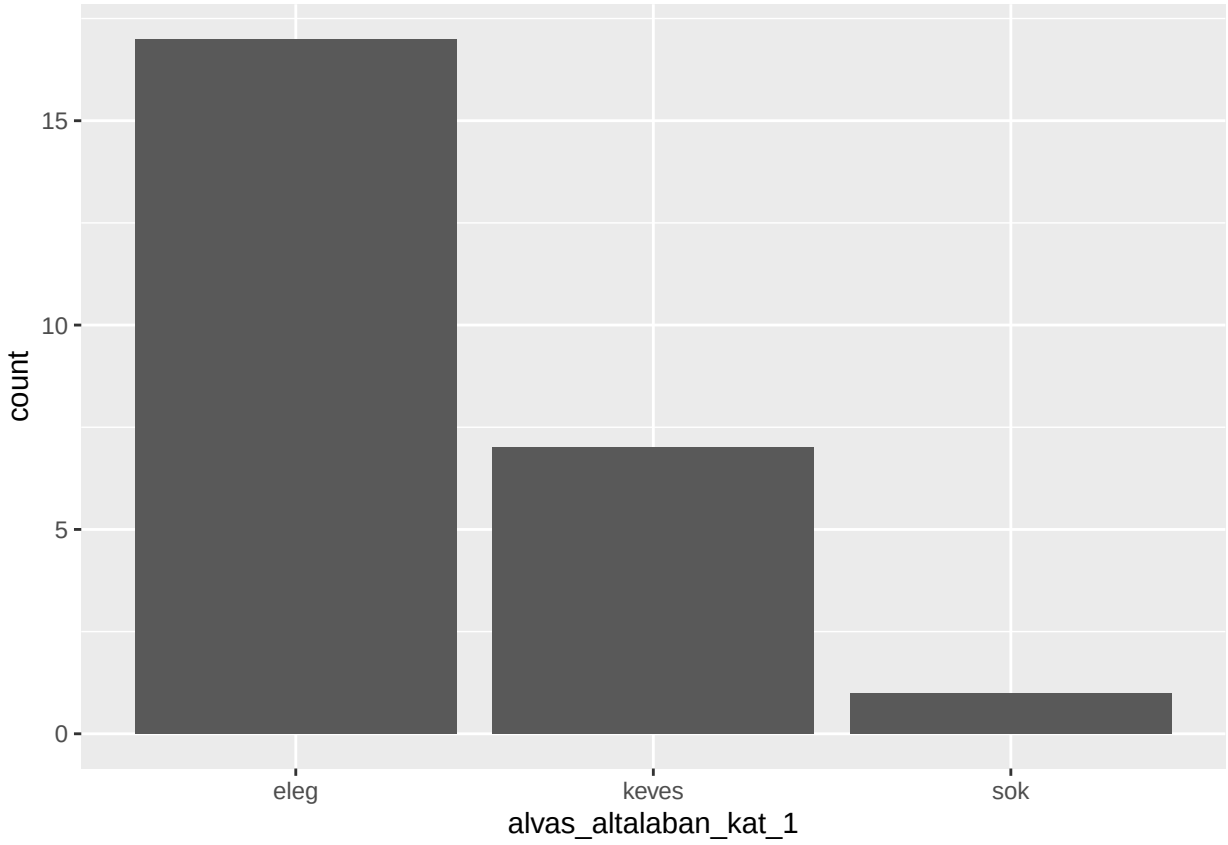
```
## [1] "eleg"  "keves" "sok"
```

Faktorszintek sorrendje, ordinalis valtozok

Amikor van ertelme a **sorrendisegnek** a faktorszintek kozott, **ordinalis valtozokrol** beszelunk (vagyis az egyik faktorszint alacsonyabb, vagy kisebb “erteku” mint a masik). Arra figyelnunk kell, hogy amikor faktorokat hozunk letre, az R automatikusan a faktorszintek neveinek **ABC sorrendje** alapjan rakja oket

sorba, es az abrakon is igy szemlelteti majd oket.

```
orai_adat %>%
  ggplot() +
  aes(x = alvas_altalaban_kat_1) +
  geom_bar()
```

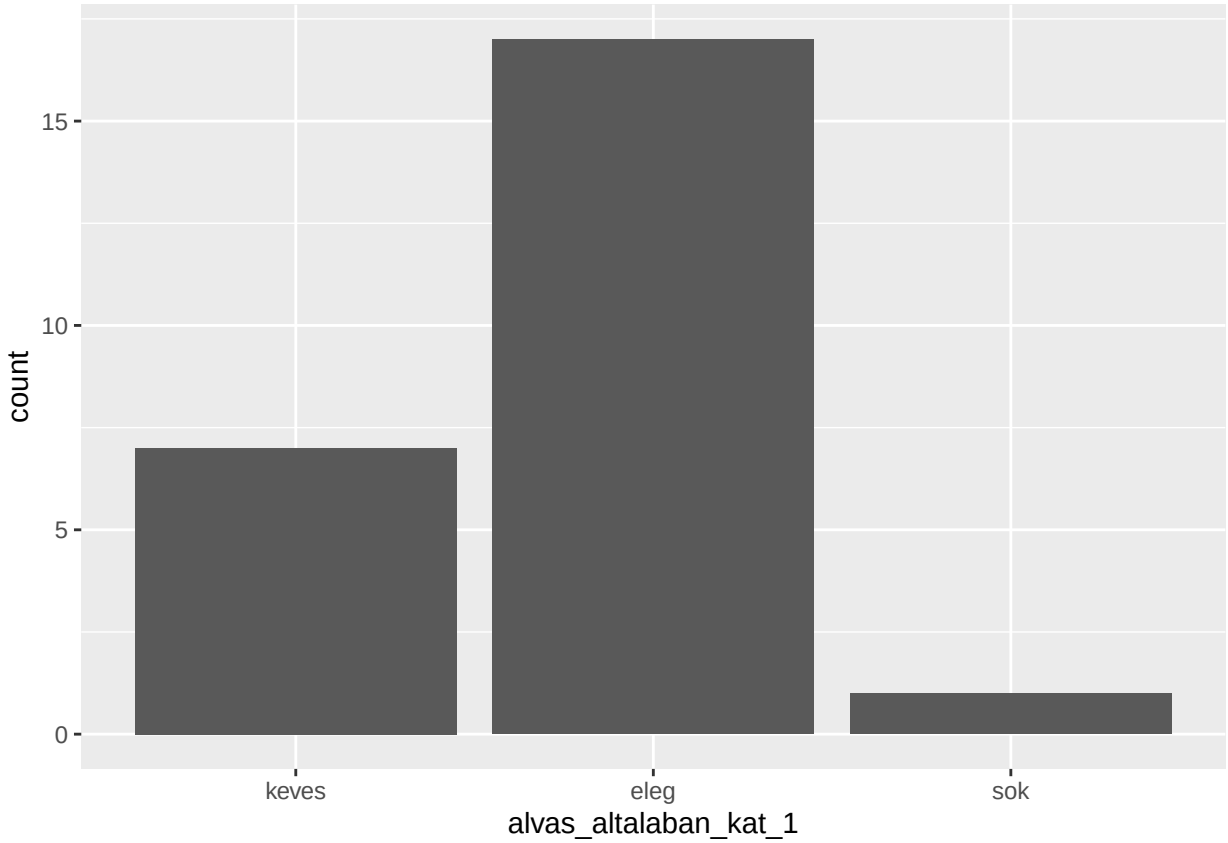


Ilyenkor erdemes meghatarozni a faktorszintek sorrendjet (**order**). Ezt legegyszerubben a factor() funkcion belul tehetjuk meg, az **ordered = T** beallitasaval, es a **levels =** resznel a szintek sorrendjenek meghatarozasaval.

```
orai_adat = orai_adat %>%
mutate(alvas_altalaban_kat_1 = factor(alvas_altalaban_kat_1, ordered = T, levels = c(
                                      "keves",
                                       "eleg",
                                       "sok")))
```

Igy mar az R minden funkcioja tudni fogja, hogy egy ordinalis valtozorol van szo, ahol fontos a sorrend, es tudni fogja a sorrendet is.

```
orai_adat %>%
  ggplot() +
  aes(x = alvas_altalaban_kat_1) +
  geom_bar()
```



Gyakorlas

- Csinalj egy kategorikus valtozot az elso orai hangulat alapjan (*hangulat_1* valotozo) ugy, hogy harom csoport alakuljon ki: 0-3 - rossz, 4-6 - kozepes, 7-10 - jo. (Emlekeztetoul: ezt a mutate() es a recode() funkciokkal tudod peldaul elerni.) Ezt az uj valtozot nevezd el *hangulat_kat_1* -nek, es az ezt az uj valtozot is tartalmazo adattablat mentsd el *orai_adat_harmashangulat_1* neven.
- Fontos, hogy a *hangulat_kat_1* valtozot faktor valotozokent jelold meg. (Ezt lehet az elozo lepesben a mutate() funkcion belul, vagy egy kulon lepesben, de mindenkeppen a factor() vagy az as.factor() funkciokat erdemes hozza hasznalni)
- Keszits egy tablazatot arrol, hogy hanyan esnek a *hangulat_kat_1* egyes kategoriaiba.
- Add meg a faktorszintek helyes sorrendjet: rossz, kozepes, jo (Ird felul a *hangulat_kat_1* korabbi valtozatat ezzel a valtozattal ahol a szintek mar helyes sorrendben vannak)
- Nezd meg a faktor szintjeit, hogy valoban helyes sorrendben szerepelnek-e

---

```
orai_adat %>%
  select(hangulat_1) %>%
  summary()
```

```
##    hangulat_1
##  Min.   :3.00
##  1st Qu.:5.00
##  Median :6.00
```

```
##  Mean   :6.12
##  3rd Qu.:8.00
##  Max.   :9.00
```

```
orai_adat_harmashangulat_1 <- orai_adat %>%
  mutate(hangulat_kat_1 = factor(recode(hangulat_1,
                                  "0" = "rossz",
                                  "1" = "rossz",
                                  "2" = "rossz",
                                  "3" = "rossz",
                                  "4" = "kozepes",
                                  "5" = "kozepes",
                                  "6" = "kozepes",
                                  "7" = "jo",
                                  "8" = "jo",
                                  "9" = "jo",
                                  "10" = "jo"
                                  ), ordered = T, levels = c("rossz", "kozepes", "jo")))

table(orai_adat_harmashangulat_1$hangulat_kat_1)
```

```
##
##   rossz kozepes      jo
##       3      12      10
```

```
orai_adat_harmashangulat_1 %>%
  select(hangulat_kat_1) %>%
  summary
```

```
##  hangulat_kat_1
##  rossz  : 3
##  kozepes:12
##  jo     :10
```

Exploracio vizualizacion keresztul

Egyes valtozok vizualizacioja

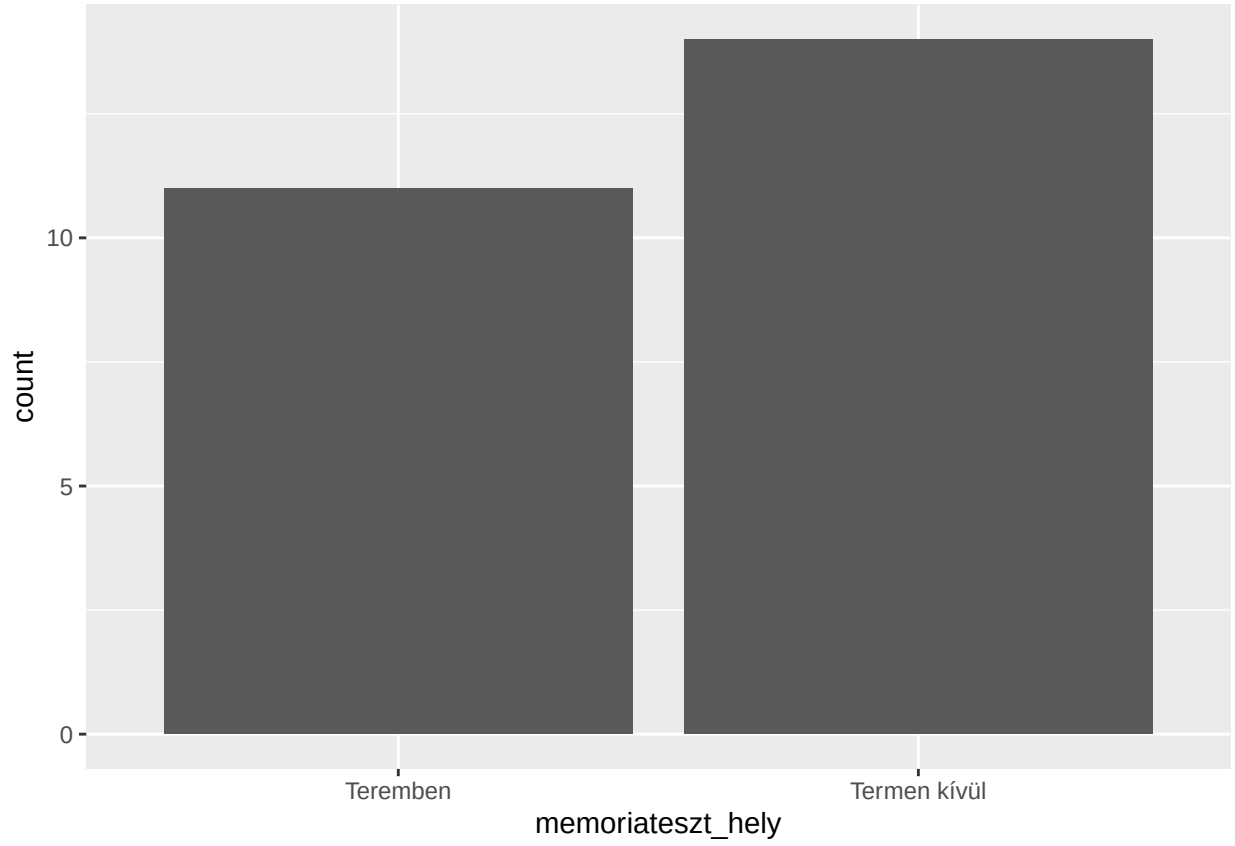
Az egyes valtozok **abrak** (plot) segitsegevel is megvizsgalhatok. A **kategorikus** valtozokat gyakran oszlopdiagrammal (**geom_bar**) abrazoljuk,

Mig a **numerikus** valtozokat inkabb **dotplot** , **histogram**, vagy **density plot** segitsegevel szoktuk abrazolni.

Az egyes valtozok vizualizacioja es a leiro statisztikak atvizsgalasa elengedhetetlen hogy azonositsuk az esetleges adatbeviteli **hibakat es egyeb nemvart furcsasagokat** az adataink kozott.

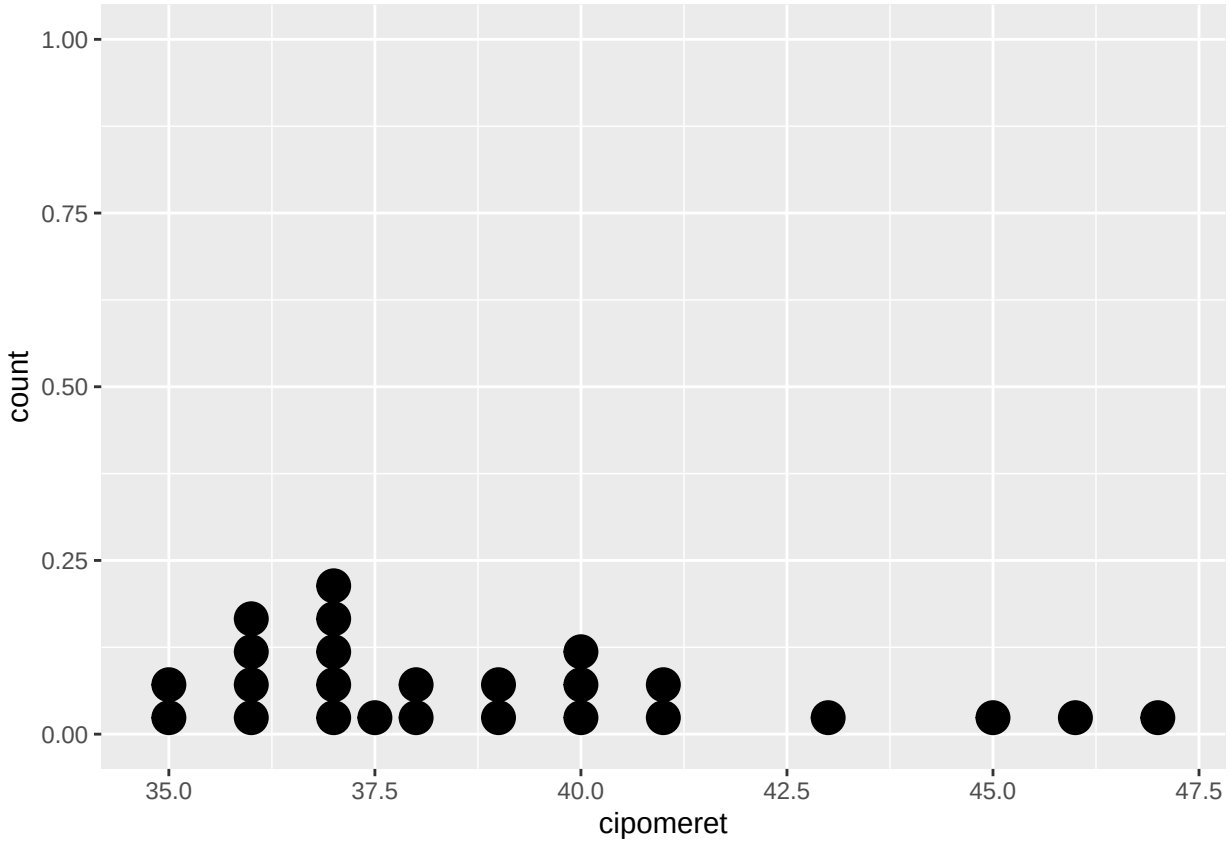
MINDING ellenorizd az adataidat ezekkel a modszerekkel mielott komolyabb adatelemzesbe kezdesz, hogy meggyozodj rola, hogy az adatok tisztak es megfelenek az elvarasaidnak.

```
orai_adat %>%
ggplot() +
  aes(x = memoriateszt_hely) +
  geom_bar()
```



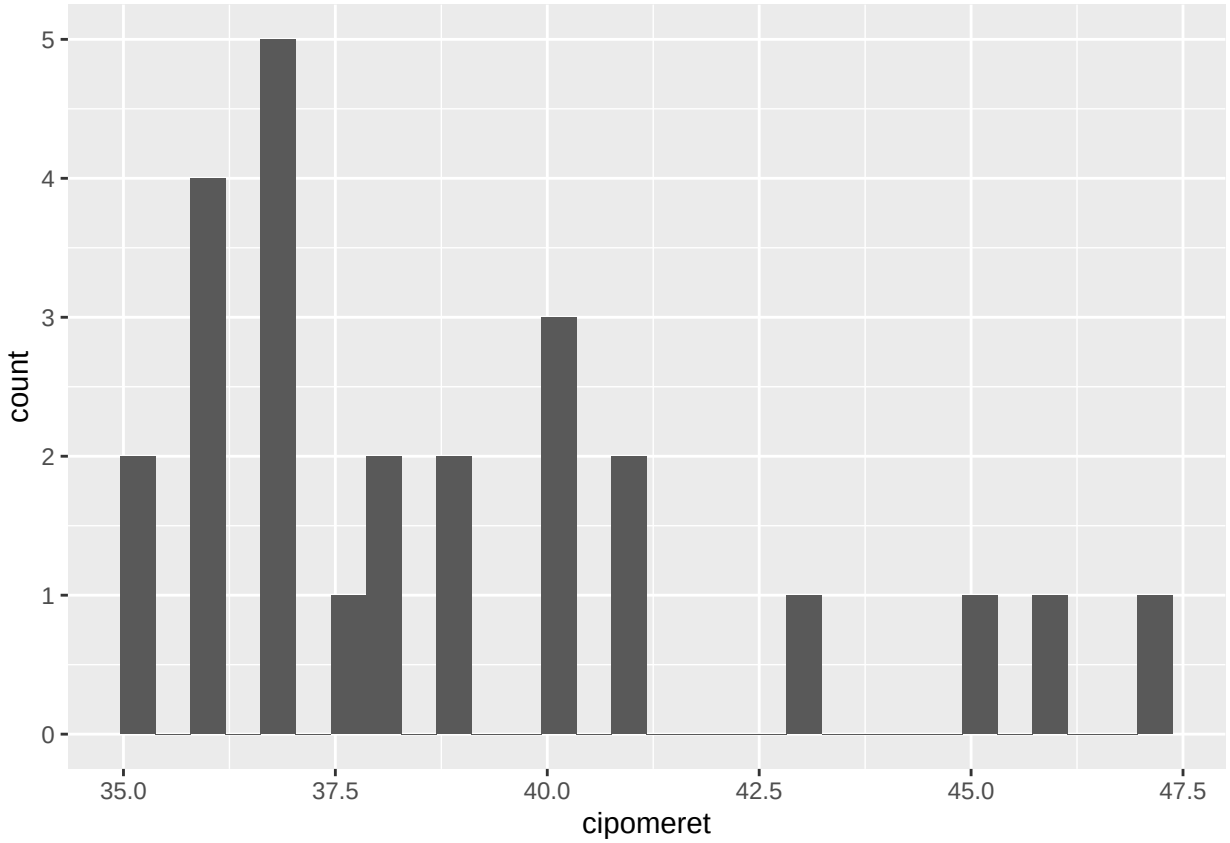

```
orai_adat %>%
ggplot() +
  aes(x = cipomeret) +
  geom_dotplot()
```

```
## `stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.
```

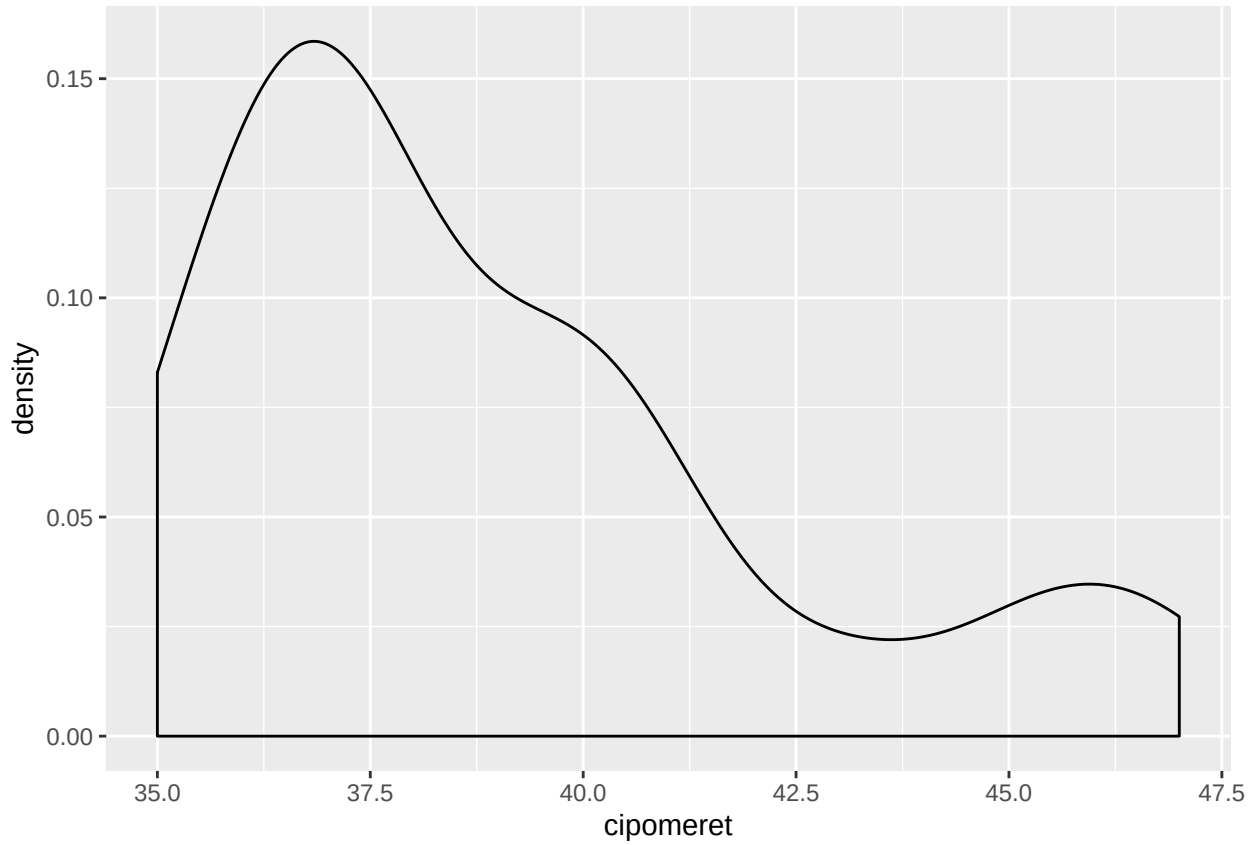



```
orai_adat %>%
ggplot() +
  aes(x = cipomeret) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
orai_adat %>%
ggplot() +
  aes(x = cipomeret) +
  geom_density()
```



Gyakorlas

Hasznald a fent tanult modszereket, hogy **azonositsd az orai_adat adattablaban levo hibakat** vagy nem vart furcsasagokat.

- A vizualizacion tul a View(), describe(), es summary() funciokat erdemes hasznalni az adatok elso attekintesere
- A numerikus (vagy eppen folytonos) valtozoknal vizsgald meg a minimum es maximum erteket es a hianyzo adatok mennyiseget, valamint az eloszlast.
- A kategorikus valtozoknal vizsgald meg az osszes faktorszintet es az egyes szintekhez tartozo megfigyelesek mennyiseget.

```
# View(orai_adat)

describe(orai_adat)
```

```
##                      vars  n   mean    sd median trimmed  mad   min  max
## kitoltesiido*           1 25    NaN    NA     NA     NaN   NA   Inf -Inf
## ID*                     2 25  32.00    NA     32   32.00 0.00 32.00   32
## nem*                    3 25   1.88  0.33      2    1.95 0.00  1.00    2
## eletkor*                4 25   1.12  0.33      1    1.05 0.00  1.00    2
## alvas_szukseges         5 25   8.32  0.90      8    8.33 1.48  6.00   10
## nemalvas_oka*           6 24    NaN    NA     NA     NaN   NA   Inf -Inf
## memoria                 7 25   7.48  1.26      8    7.57 1.48  4.00   10
```

```
## memoria_ront_A*          8 23    NaN    NA     NA     NaN   NA   Inf -Inf
## memoria_ront_B*          9 19    NaN    NA     NA     NaN   NA   Inf -Inf
## memoria_ront_C*         10 19    NaN    NA     NA     NaN   NA   Inf -Inf
## memoria_ront_D*         11 14    NaN    NA     NA     NaN   NA   Inf -Inf
## memoria_ront_E*         12  7    NaN    NA     NA     NaN   NA   Inf -Inf
## memoria_ront_F*         13  4    NaN    NA     NA     NaN   NA   Inf -Inf
## jegy_tipp               14 25   4.04  0.73      4    4.05 1.48  3.00    5
## valasztott_szam         15 25   6.56  2.52      7    6.62 2.97  2.00   10
## tippelt_gyakori_szam    16 25   5.80  2.58      6    5.76 2.97  2.00   10
## magassag                17 25 161.07 34.32    168  166.62 8.90  1.82  190
## cipomeret               18 25  38.94  3.34     38   38.60 2.97 35.00   47
## memoriateszt_hely*      19 25   1.56  0.51      2    1.57 0.00  1.00    2
## alvas_tegnap_1          20 25   7.92  1.26      8    7.90 1.48  6.00   10
## alvas_tegnap_3          21 23   7.61  1.03      8    7.68 1.48  5.00    9
## alvas_altalaban_1       22 25   7.04  1.02      7    7.10 1.48  5.00    9
## alvas_altalaban_3       23 23   7.13  0.63      7    7.16 0.00  6.00    8
## energiaszint_1          24 25   5.48  2.16      5    5.62 2.97  1.00    8
## energiaszint_3          25 23   6.17  1.92      7    6.26 1.48  3.00    9
## szocmedia_1             26 25   1.96  1.57      1    1.81 1.48  0.00    6
## szocmedia_3*            27 24   1.96  1.36      2    1.79 1.48  0.00    5
## hangulat_1              28 25   6.12  1.79      6    6.14 1.48  3.00    9
## hangulat_3              29 23   5.91  2.21      6    6.16 2.97  1.00    9
## stat_tudasszint_3       30 23   4.09  2.02      5    4.16 1.48  0.00    8
## stat_gyakorlas_3        31 23   4.59  3.54      3    4.08 2.22  0.00   14
## alvas_altalaban_kat_1*  32 25   1.76  0.52      2    1.76 0.00  1.00    3
##                        range  skew kurtosis   se
## kitoltesiido*           -Inf    NA       NA   NA
## ID*                     0.00    NA       NA   NA
## nem*                    1.00 -2.20     2.96 0.07
## eletkor*                1.00  2.20     2.96 0.07
## alvas_szukseges         4.00 -0.30     0.22 0.18
## nemalvas_oka*           -Inf    NA       NA   NA
## memoria                 6.00 -0.67     0.89 0.25
## memoria_ront_A*         -Inf    NA       NA   NA
## memoria_ront_B*         -Inf    NA       NA   NA
## memoria_ront_C*         -Inf    NA       NA   NA
## memoria_ront_D*         -Inf    NA       NA   NA
## memoria_ront_E*         -Inf    NA       NA   NA
## memoria_ront_F*         -Inf    NA       NA   NA
## jegy_tipp               2.00 -0.06    -1.22 0.15
## valasztott_szam         8.00 -0.38    -1.22 0.50
## tippelt_gyakori_szam    8.00  0.11    -1.16 0.52
## magassag              188.18 -3.95    15.59 6.86
## cipomeret              12.00  0.99    -0.05 0.67
## memoriateszt_hely*      1.00 -0.23    -2.02 0.10
## alvas_tegnap_1          4.00 -0.22    -1.12 0.25
## alvas_tegnap_3          4.00 -0.63    -0.11 0.22
## alvas_altalaban_1       4.00 -0.30    -0.71 0.20
## alvas_altalaban_3       2.00 -0.07    -0.63 0.13
## energiaszint_1          7.00 -0.27    -1.21 0.43
## energiaszint_3          6.00 -0.38    -1.35 0.40
## szocmedia_1             6.00  1.06     0.20 0.31
## szocmedia_3*            5.00  1.00    -0.11 0.28
## hangulat_1              6.00 -0.17    -0.98 0.36
```

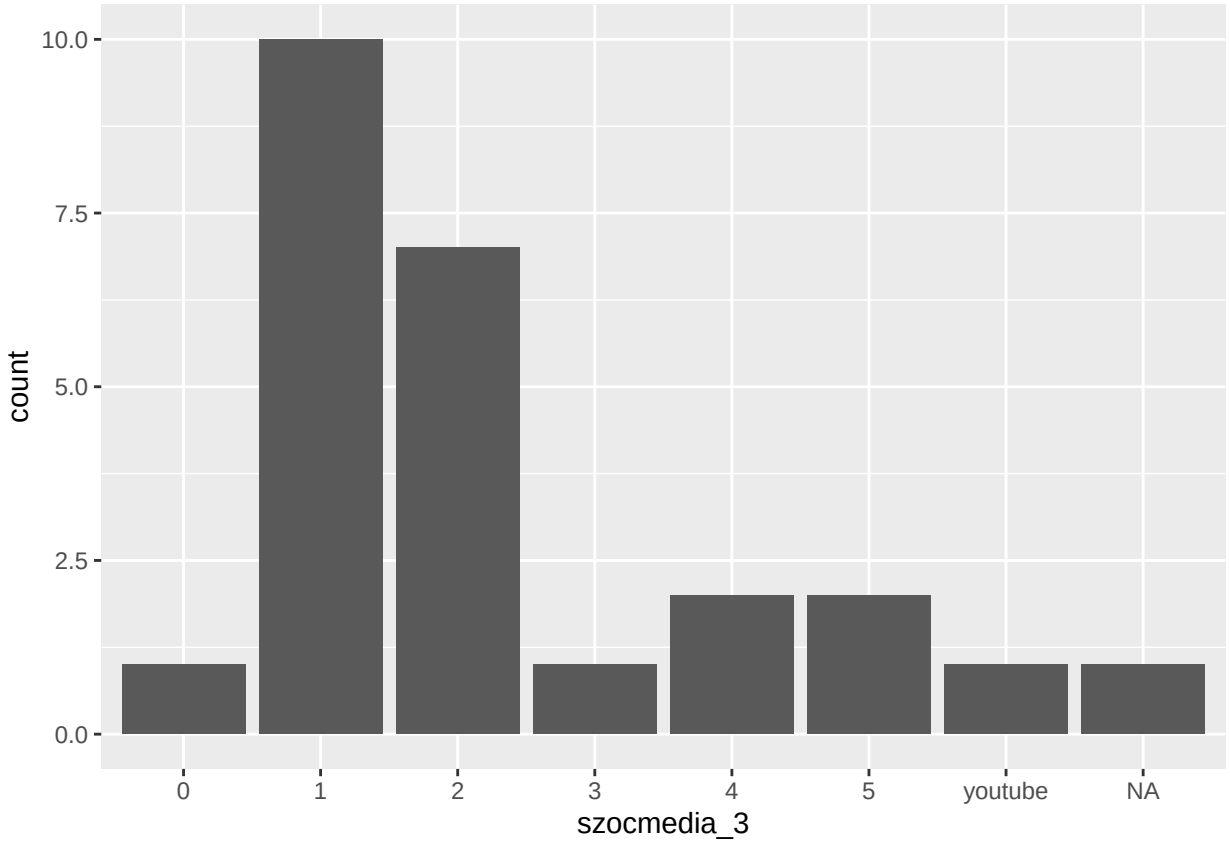
```
## hangulat_3               8.00 -0.74    -0.30 0.46
## stat_tudasszint_3        8.00 -0.18    -0.99 0.42
## stat_gyakorlas_3        14.00  1.26     0.88 0.74
## alvas_altalaban_kat_1*   2.00 -0.26    -0.45 0.10
```

```
orai_adat %>%
  summary()
```

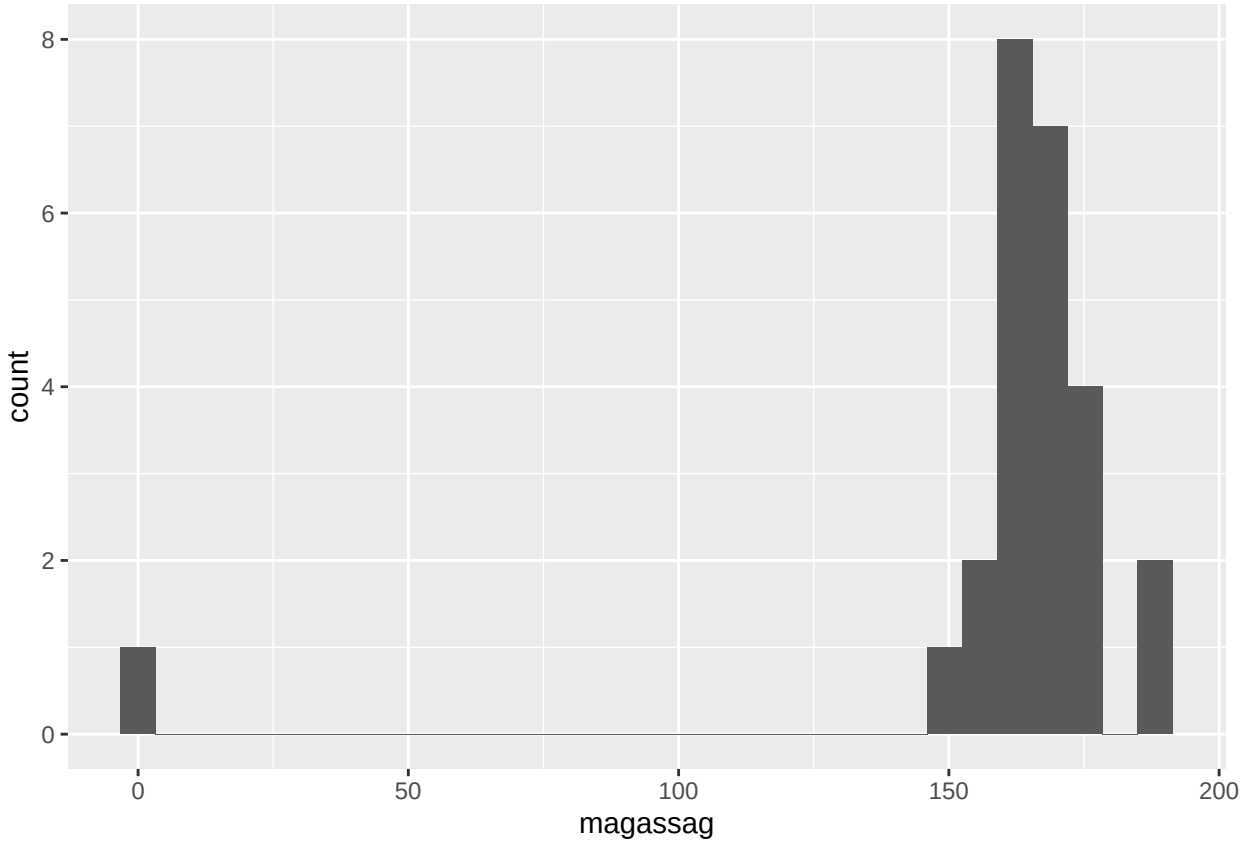
```
##  kitoltesiido            ID              nem     eletkor
##  Length:25          Length:25          Férfi: 3   0-22:22
##  Class :character   Class :character   No   :22   23+ : 3
##  Mode  :character   Mode  :character
##
##
##
##
##  alvas_szukseges nemalvas_oka          memoria      memoria_ront_A
##  Min.   : 6.00   Length:25          Min.   : 4.00   Length:25
##  1st Qu.: 8.00   Class :character   1st Qu.: 7.00   Class :character
##  Median : 8.00   Mode  :character   Median : 8.00   Mode  :character
##  Mean   : 8.32                      Mean   : 7.48
##  3rd Qu.: 9.00                      3rd Qu.: 8.00
##  Max.   :10.00                      Max.   :10.00
##
##  memoria_ront_B     memoria_ront_C     memoria_ront_D
##  Length:25          Length:25          Length:25
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  memoria_ront_E     memoria_ront_F       jegy_tipp    valasztott_szam
##  Length:25          Length:25          Min.   :3.00   Min.   : 2.00
##  Class :character   Class :character   1st Qu.:4.00   1st Qu.: 5.00
##  Mode  :character   Mode  :character   Median :4.00   Median : 7.00
##                                        Mean   :4.04   Mean   : 6.56
##                                        3rd Qu.:5.00   3rd Qu.: 8.00
##                                        Max.   :5.00   Max.   :10.00
##
##  tippelt_gyakori_szam    magassag        cipomeret
##  Min.   : 2.0         Min.   :  1.82   Min.   :35.00
##  1st Qu.: 3.0         1st Qu.:162.00   1st Qu.:37.00
##  Median : 6.0         Median :168.00   Median :38.00
##  Mean   : 5.8         Mean   :161.07   Mean   :38.94
##  3rd Qu.: 7.0         3rd Qu.:170.00   3rd Qu.:40.00
##  Max.   :10.0         Max.   :190.00   Max.   :47.00
##
##     memoriateszt_hely alvas_tegnap_1  alvas_tegnap_3  alvas_altalaban_1
##  Teremben    :11      Min.   : 6.00   Min.   :5.000   Min.   :5.00
##  Termen kívül:14      1st Qu.: 7.00   1st Qu.:7.000   1st Qu.:6.00
##                       Median : 8.00   Median :8.000   Median :7.00
##                       Mean   : 7.92   Mean   :7.609   Mean   :7.04
##                       3rd Qu.: 9.00   3rd Qu.:8.000   3rd Qu.:8.00
##                       Max.   :10.00   Max.   :9.000   Max.   :9.00
```

```
##                                      NA's   :2
##  alvas_altalaban_3 energiaszint_1 energiaszint_3   szocmedia_1
##  Min.   :6.00      Min.   :1.00   Min.   :3.000   Min.   :0.00
##  1st Qu.:7.00      1st Qu.:4.00   1st Qu.:4.500   1st Qu.:1.00
##  Median :7.00      Median :5.00   Median :7.000   Median :1.00
##  Mean   :7.13      Mean   :5.48   Mean   :6.174   Mean   :1.96
##  3rd Qu.:7.50      3rd Qu.:8.00   3rd Qu.:8.000   3rd Qu.:3.00
##  Max.   :8.00      Max.   :8.00   Max.   :9.000   Max.   :6.00
##  NA's   :2                        NA's   :2
##  szocmedia_3          hangulat_1     hangulat_3    stat_tudasszint_3
##  Length:25          Min.   :3.00   Min.   :1.000   Min.   :0.000
##  Class :character   1st Qu.:5.00   1st Qu.:5.000   1st Qu.:2.500
##  Mode  :character   Median :6.00   Median :6.000   Median :5.000
##                     Mean   :6.12   Mean   :5.913   Mean   :4.087
##                     3rd Qu.:8.00   3rd Qu.:8.000   3rd Qu.:6.000
##                     Max.   :9.00   Max.   :9.000   Max.   :8.000
##                                    NA's   :2       NA's   :2
##  stat_gyakorlas_3 alvas_altalaban_kat_1
##  Min.   : 0.000   keves: 7
##  1st Qu.: 2.500   eleg :17
##  Median : 3.000   sok  : 1
##  Mean   : 4.587
##  3rd Qu.: 6.000
##  Max.   :14.000
##  NA's   :2
```

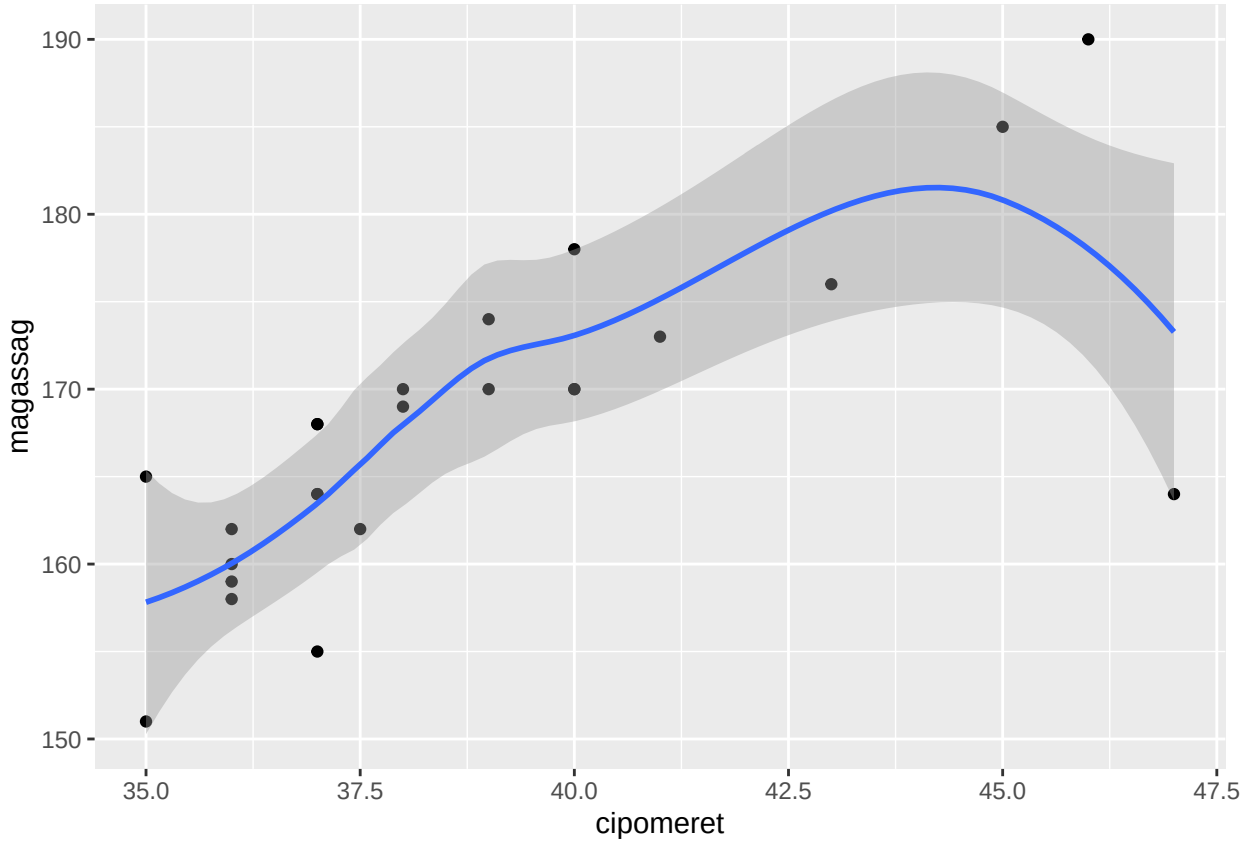
```
orai_adat %>%
  ggplot()+
    aes(x = szocmedia_3)+
    geom_bar()
```




```
orai_adat %>%
  ggplot()+
    aes(x = magassag)+
    geom_histogram()
```




```
orai_adat %>%
  filter(magassag>2) %>%
  ggplot()+
    aes(x = cipomeret, y = magassag)+
    geom_point() +
    geom_smooth()
```



A hibakat a kovetkezokeppen javithatjuk.

A **mutate()** es a **replace()** funkciok hasznalataval **cserelhetunk ki** ertekeket mas ertekekre. Azt, hogy ilyenkor hianyzo adatra (NA), vagy egy masik, valoszinu ertekre kell megvaltoztatni az erteket, a szituaciotol fogg. Altalaban a biztosabb megoldas ha hianyzo adatnak jeloljuk a kerdeses erteket (NA), de ez sok adatveszteshez vezethet. Ha eleg valszinu hogy mi a helyes valasz, beirhatjuk, DE **minden javitast fel kell tuntetni** a kutatasi jelentesben (es a ZH soran is), hogy az olvaso szamara tiszta legyen, hogy itt egy adathelyettesites vagy kizaras tortent!

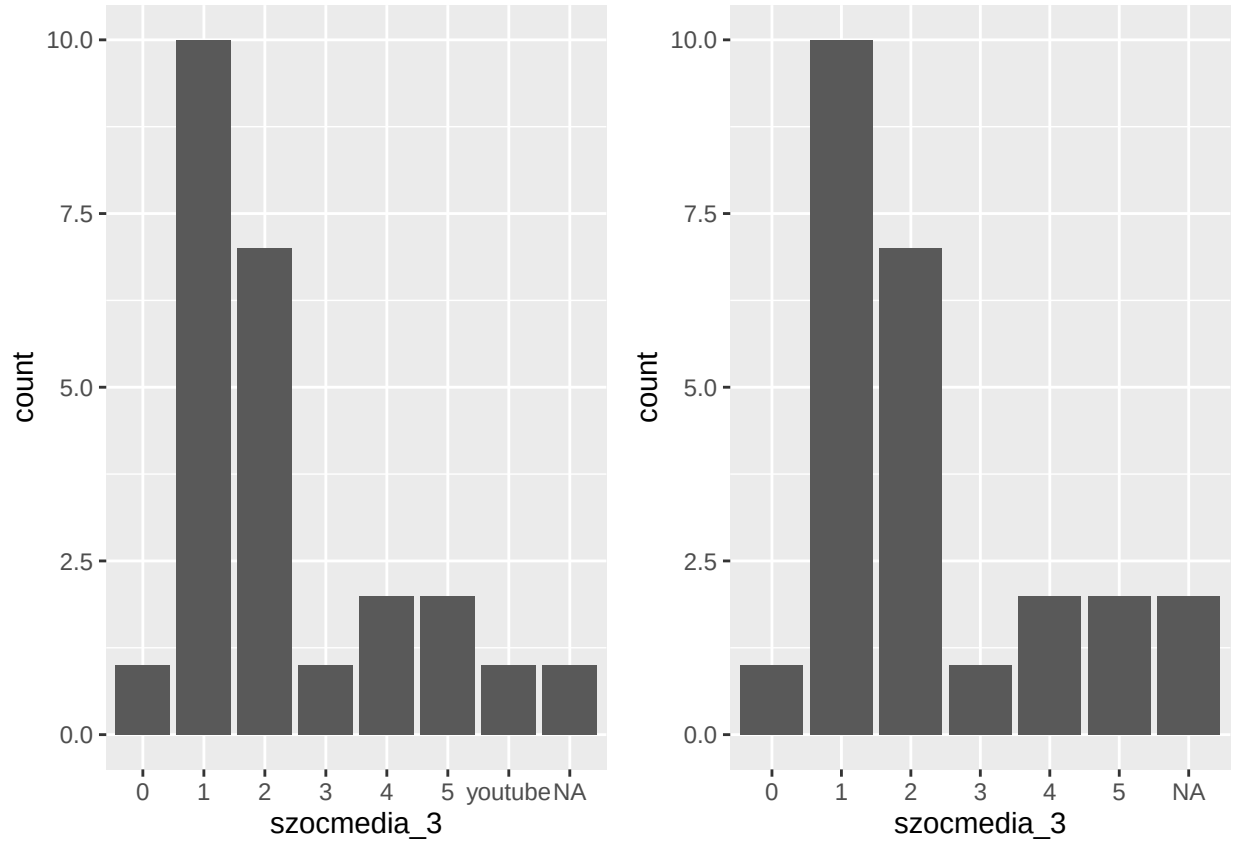
Mindig erdemes a javitott adatokat **uj adattablaba** elmenteni. A mi esetunkben az orai_adat_corrected nevet adtuk a javitott objektumnak. Igy a nyers adataink megmaradnak, ami hasznos lehet kesobbi muveleteknel.

```
orai_adat_corrected <- orai_adat %>%
  mutate(szocmedia_3 = replace(szocmedia_3,  szocmedia_3=="youtube", NA))

orai_adat_corrected = orai_adat_corrected %>%
  mutate(magassag = replace(magassag,  magassag == 1.82, 182))

orai_adat_corrected = orai_adat_corrected %>%
  mutate(cipomeret = replace(cipomeret, cipomeret == 47, 37))
```

Erdemes megbizonyosodni rola, hogy az adatcsere sikeres volt. Alabb az adatok vizualizaciojaval gyozodunk meg errol, de az adatok megjelenitesevel, vagy a leiro statisztikak lekerdezesevel is megteheto ez, ha az informativ.

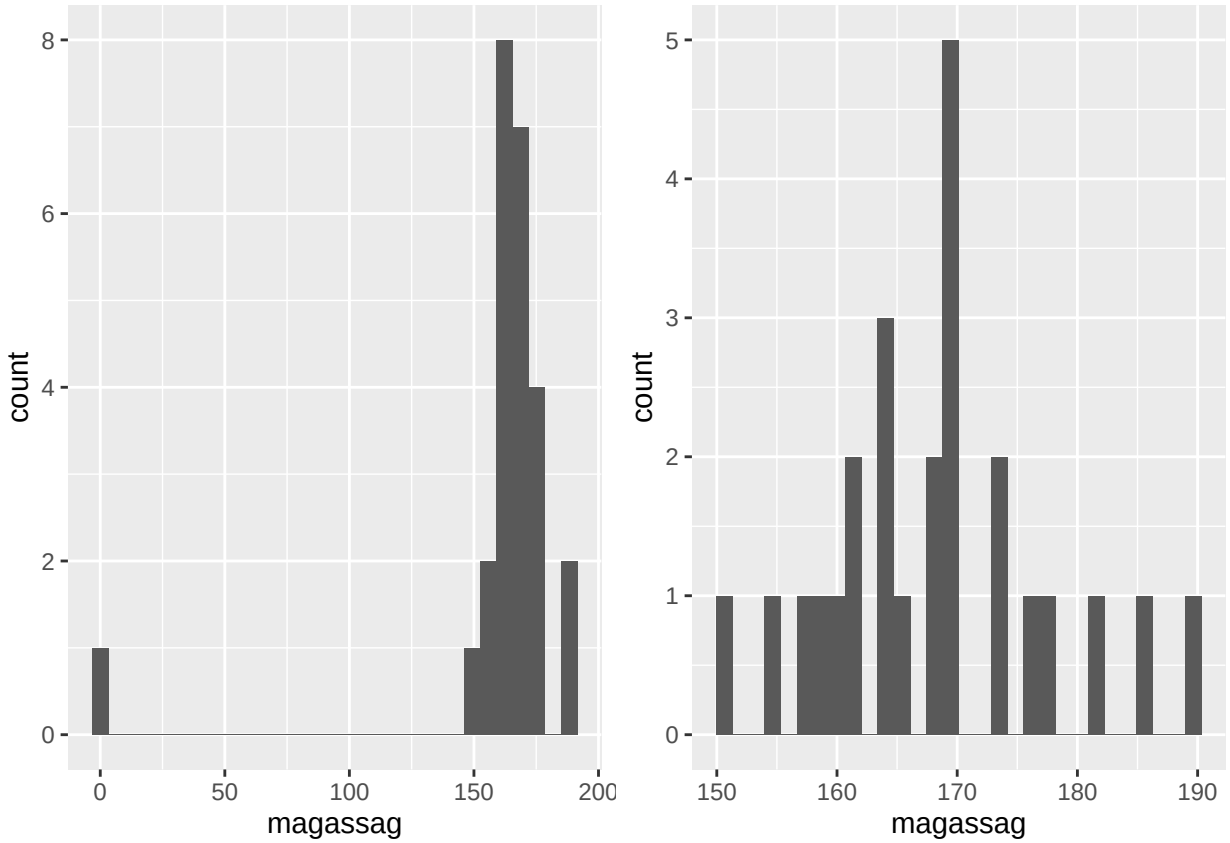
```
# hasznalhatnak meg az alabbiakat is arra,
# hogy megbizonyosodjunk abban, hogy sikeres volt a csere
```

```
# View(orai_adat_corrected)
# describe(orai_adat_corrected)
# summary(orai_adat_corrected$szocmedia_3)
# orai_adat_corrected$szocmedia_3

old_plot_szocmedia_3 <-
  orai_adat %>%
  ggplot()+
    aes(x = szocmedia_3)+
    geom_bar()

new_plot_szocmedia_3 <-
  orai_adat_corrected %>%
  ggplot()+
    aes(x = szocmedia_3)+
    geom_bar()

grid.arrange(old_plot_szocmedia_3, new_plot_szocmedia_3, ncol=2)
```

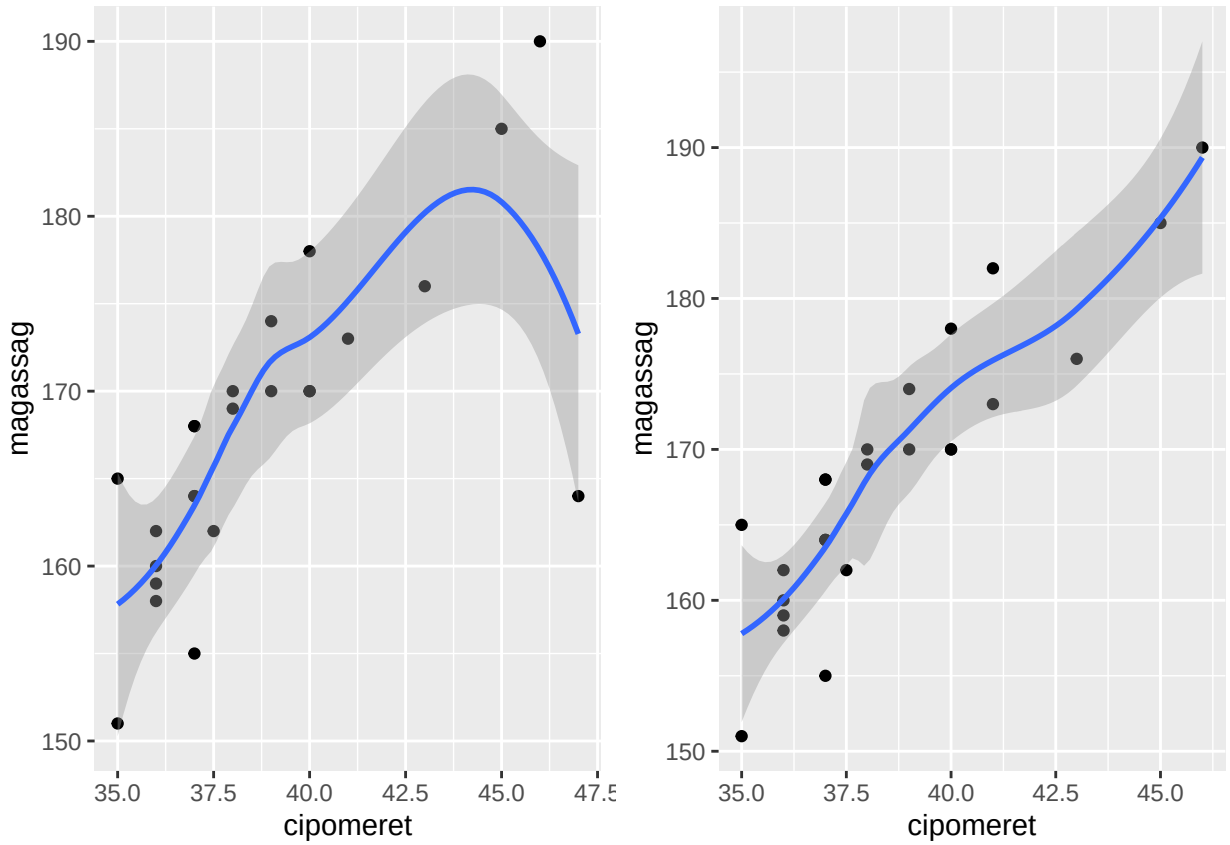



```
old_plot_magassag <-
  orai_adat %>%
  ggplot()+
    aes(x = magassag)+
    geom_histogram()

new_plot_magassag <-
```

```
  orai_adat_corrected %>%
  ggplot()+
    aes(x = magassag)+
    geom_histogram()


grid.arrange(old_plot_magassag, new_plot_magassag, ncol=2)
```




```
old_plot_cipomeret <-
  orai_adat %>%
  filter(magassag>2) %>%
  ggplot()+
    aes(x = cipomeret, y = magassag)+
    geom_point() +
    geom_smooth()

new_plot_cipomeret <-
  orai_adat_corrected %>%
  filter(magassag>2) %>%
  ggplot()+
    aes(x = cipomeret, y = magassag)+
    geom_point() +
    geom_smooth()

grid.arrange(old_plot_cipomeret, new_plot_cipomeret, ncol=2)
```



Tobb valtozo kapcsolatanak felterkepezese

Tobb valtozo kapcsolatat is felterkepezhetjuk tablazatok es abrak segitsegevel.

Ket kategorikus (csoportosito) valtozo kapcsolatanak felterkepezese

Generalunk kategorikus valtozokat

Az elso oran jelentett **hangulat_1** valtozobol csinalunk egy kategorikus valtozot, az faktorszintek a kovetkezok lesznek: 0-3 - "rossz", 4-6 - "kozepes", 7-10 - "jo". Ezt ugyan abba az adattablaba hangulat_kat_1 neven mentjuk el sorrendezett factorkent a fent tanultak alapjan. Mindehez a mutate(), recode(), es factor() funkciokat hasznaljuk.

```
orai_adat_corrected %>%
  select(hangulat_1) %>%
  summary()
```

```
##    hangulat_1
##  Min.   :3.00
##  1st Qu.:5.00
##  Median :6.00
##  Mean   :6.12
##  3rd Qu.:8.00
##  Max.   :9.00
```

```
orai_adat_corrected <- orai_adat_corrected %>%
  mutate(hangulat_kat_1 = factor(recode(hangulat_1,
                                        "0" = "rossz",
```

```
                                  "1" = "rossz",
                                  "2" = "rossz",
                                  "3" = "rossz",
                                  "4" = "kozepes",
                                  "5" = "kozepes",
                                  "6" = "kozepes",
                                  "7" = "jo",
                                  "8" = "jo",
                                  "9" = "jo",
                                  "10" = "jo"
                                  ), ordered = T, levels = c("rossz", "kozepes", "jo")))

orai_adat_corrected %>%
  select(hangulat_kat_1) %>%
  summary
```

```
##  hangulat_kat_1
##  rossz  : 3
##  kozepes:12
##  jo     :10
```

Hasonlokeppen kepzunk **szocmedia_1** valtozobol egy kategorikus valtozot, ahol azok akik 0 vagy 1 orat hasznaljak a szocialis mediat, a "nem rendszeres", azok akik 2 vagy tobb orat hasznaljak a szocialis mediat, a "nem rendszeres" kategoriaba (faktorszintre) esnek majd.

Itt nem szukseges az ordered = T es a levels = beallitasa, mert csak ket faktorszint van, es az ABC sorrend mar alapbol az intuitiv modon a "nem rendszeres" kategoriat teszi elsonek.

```
table(orai_adat_corrected$szocmedia_1)
```

```
##
##  0  1  2  3  5  6
##  2 12  3  5  2  1
```

```
orai_adat_corrected <- orai_adat_corrected %>%
  mutate(szocmedia_kat_1 = factor(recode(szocmedia_1,
                                  "0" = "nem rendszeres",
                                  "1" = "nem rendszeres",
                                  "2" = "rendszeres",
                                  "3" = "rendszeres",
                                  "4" = "rendszeres",
                                  "5" = "rendszeres",
                                  "6" = "rendszeres"
                                  )))

orai_adat_corrected %>%
  select(szocmedia_kat_1) %>%
  summary
```

```
##        szocmedia_kat_1
##  nem rendszeres:14
##  rendszeres    :11
```

Az elso oran kertunk egy tippet arrol, hogy az egyes emberek milyen jegyet fognak kapni a kurzus vegen az adatelemzes oran (**jegy_tipp** valtozo). Mivel itt szamban kertuk a valaszt, ez jelenleg egy numerikus valtozo, de csinaljunk most belole egy **kategorikus valtozot**, mert osszesen harom fajta tipp erkezett: 3 - "kozepes", 4 - "jo", 5 - "kivalo"

```r
table(orai_adat_corrected$jegy_tipp)
```

```
##
##  3  4  5
##  6 12  7
```

```r
orai_adat_corrected <- orai_adat_corrected %>%
  mutate(jegy_tipp_kat = factor(recode(jegy_tipp,
                                "3" = "kozepes (3)",
                                "4" = "jo (4)",
                                "5" = "kivalo (5)"
                                ), ordered = T, levels = c("kozepes (3)", "jo (4)", "kivalo (5)")))

orai_adat_corrected %>%
  select(jegy_tipp_kat) %>%
  summary()
```

```
##      jegy_tipp_kat
##  kozepes (3): 6
##  jo (4)     :12
##  kivalo (5) : 7
```

Feltaro elemzes

Most vizsgaljuk meg azt, hogy az, hogy az emberek mennyit alszanak altalaban (**alvas_altalaban_kat_1**) hogyan fugg ossze azzal, hogy mennyit hasznaljak a szocialis mediat (**szocmedia_kat_1**).

A legegyszerubb modja ket csoportosito valtozo kapcsolatanak megvizsgalasara a ket valtozo **keresztablazatanak (crosstab)** elkezsitese a **table()** funkcioval.

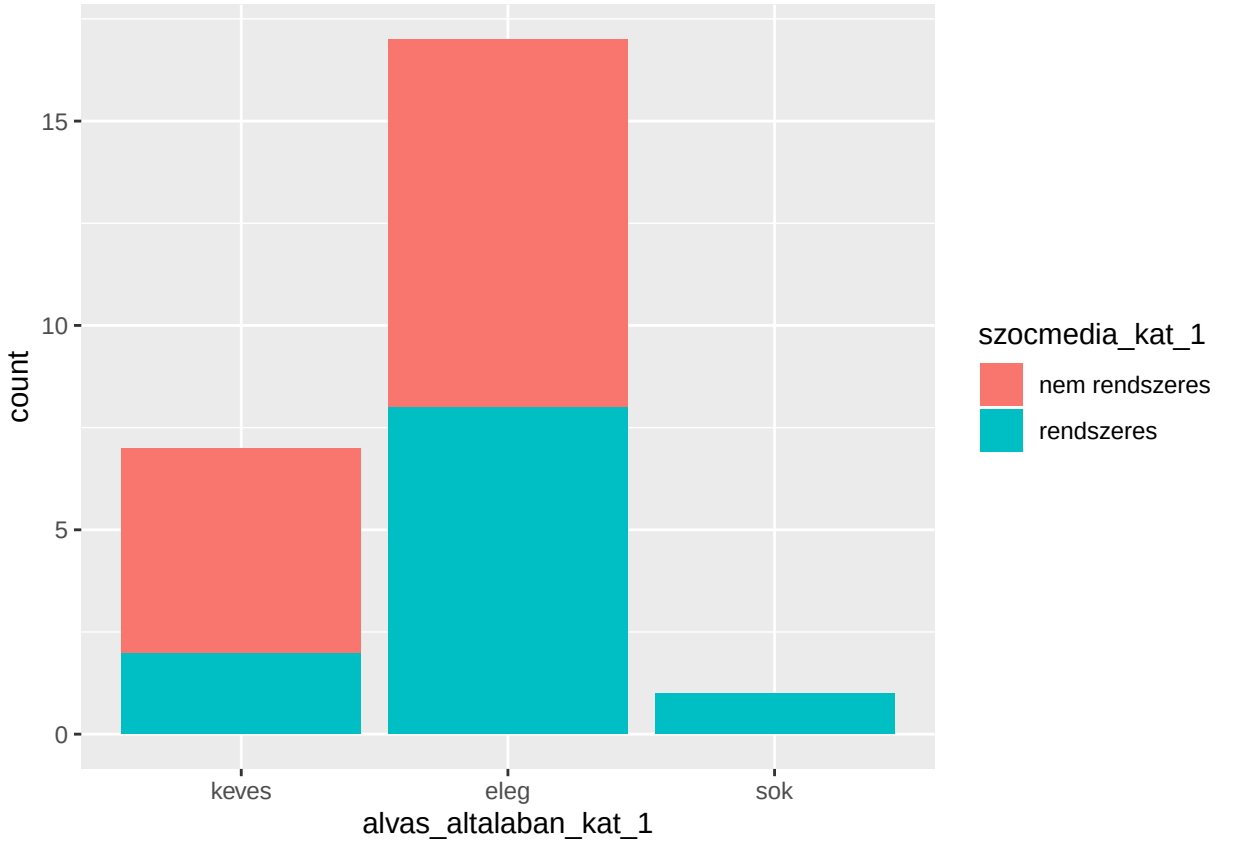
```r
table(orai_adat_corrected$alvas_altalaban_kat_1, orai_adat_corrected$szocmedia_kat_1)
```

```
##
##         nem rendszeres rendszeres
##   keves              5          2
##   eleg               9          8
##   sok                0          1
```

Sokszor ennel sokkal **szemleletesebb az abrak** (plot) hasznalata.

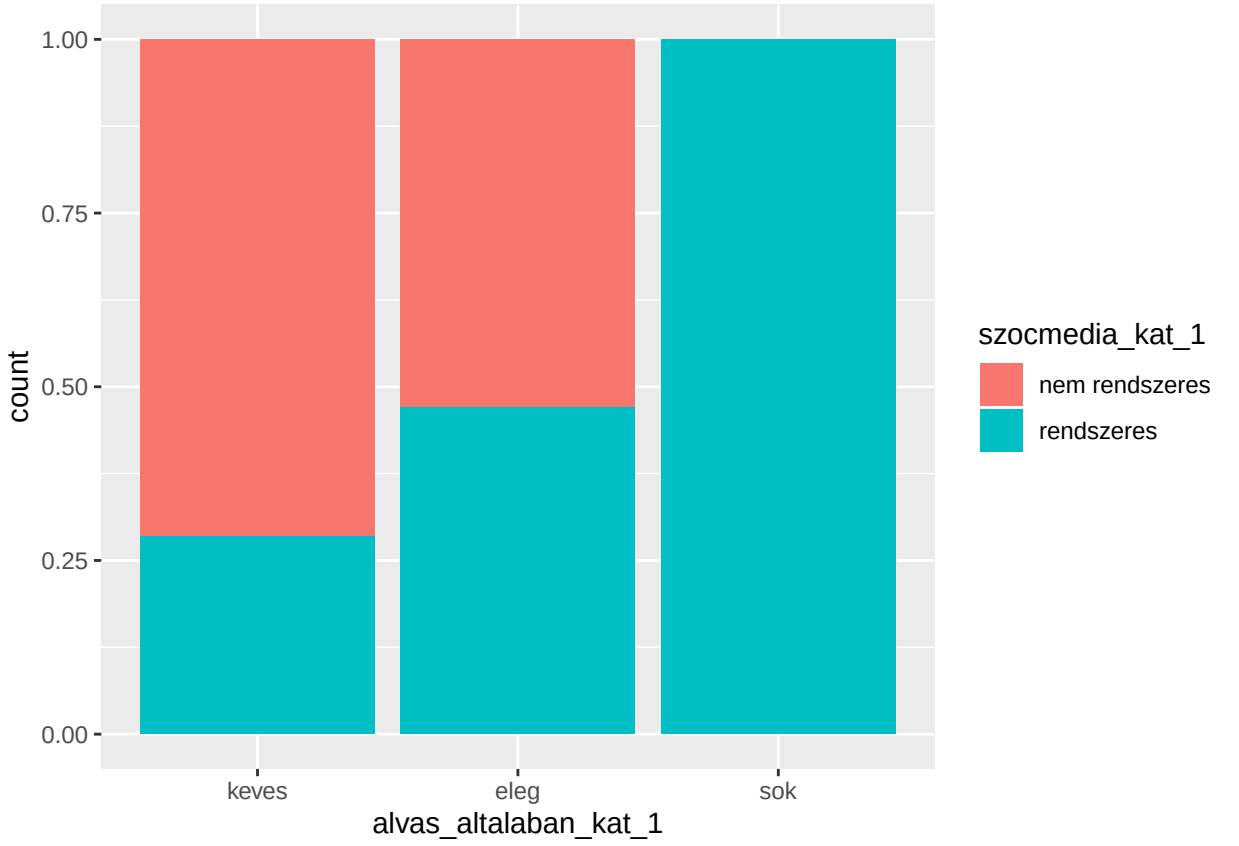
Erre az egyik lehetoseg a **stacked bar chart** (egymasra tornyozott oszlopdiagram, a **geom_bar()** geomot hasznaljuk) hasznalata. Itt az egyik valtozo kategoriai adjak meg hany oszlop lesz (ez a valtozo lesz az x tengelyen reprezentalva, igy ezt az "x =" reszen adhatjuk meg), a masik valtozo az oszlopokat szinekkel szegmentalja, ezt pedig a "**fill =**" reszen adhatjuk meg.

```r
orai_adat_corrected %>%
ggplot() +
  aes(x = alvas_altalaban_kat_1, fill = szocmedia_kat_1) +
  geom_bar()
```



Ha az egyes faktorszinteken nagyon **kulonbozo mennyisegu megfigyeles** van, ez a megjelenites neha felrevezeto kovetkeztetesekhez vezethet, igy neha hasznosabb ha az oszlopok nem szamossagot (count), hanem **reszaranyt (proportion)** jelolnek. Ha ezt szeretnenk, ahelyett hogy uresen hagynank a geom_bar() funkciot, a kovetkezot adjuk meg: **geom_bar(position = "fill")**.

```
orai_adat_corrected %>%
ggplot() +
  aes(x = alvas_altalaban_kat_1, fill = szocmedia_kat_1) +
  geom_bar(position = "fill")
```

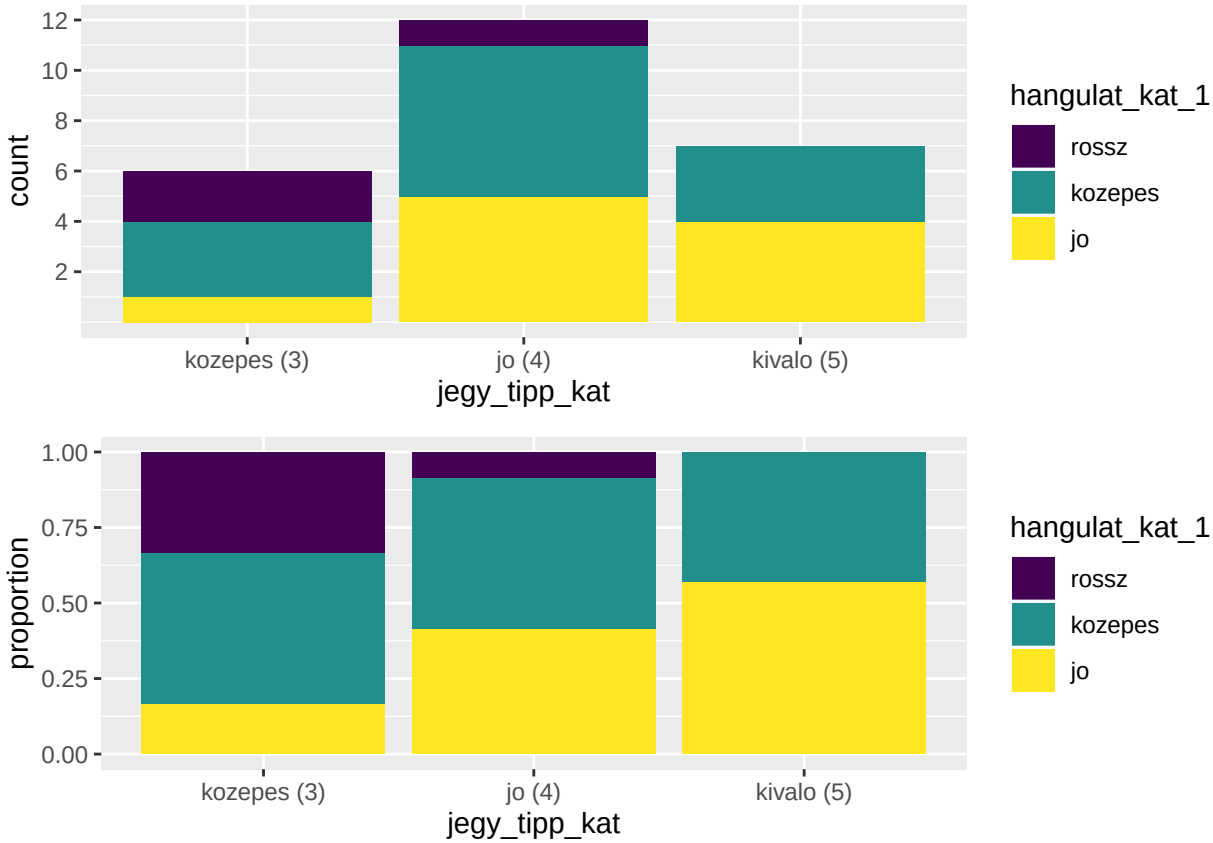


Gyakorlas

Hasznald a fent tanult modszereket, hogy megvizsgald a **jegy_tipp_kat** es a **hangulat_kat_1** valtozok kozotti osszefuggest. - hasznalj **geom_bar()** geomot a megjeleniteshez - probald meg mind a **szamossagot**, mind a **reszaranyt** kifejezo abrat megvizsgalni geom_bar(position = "fill") - milyen **kovetkeztetest** tudsz levonni az abrakrol?

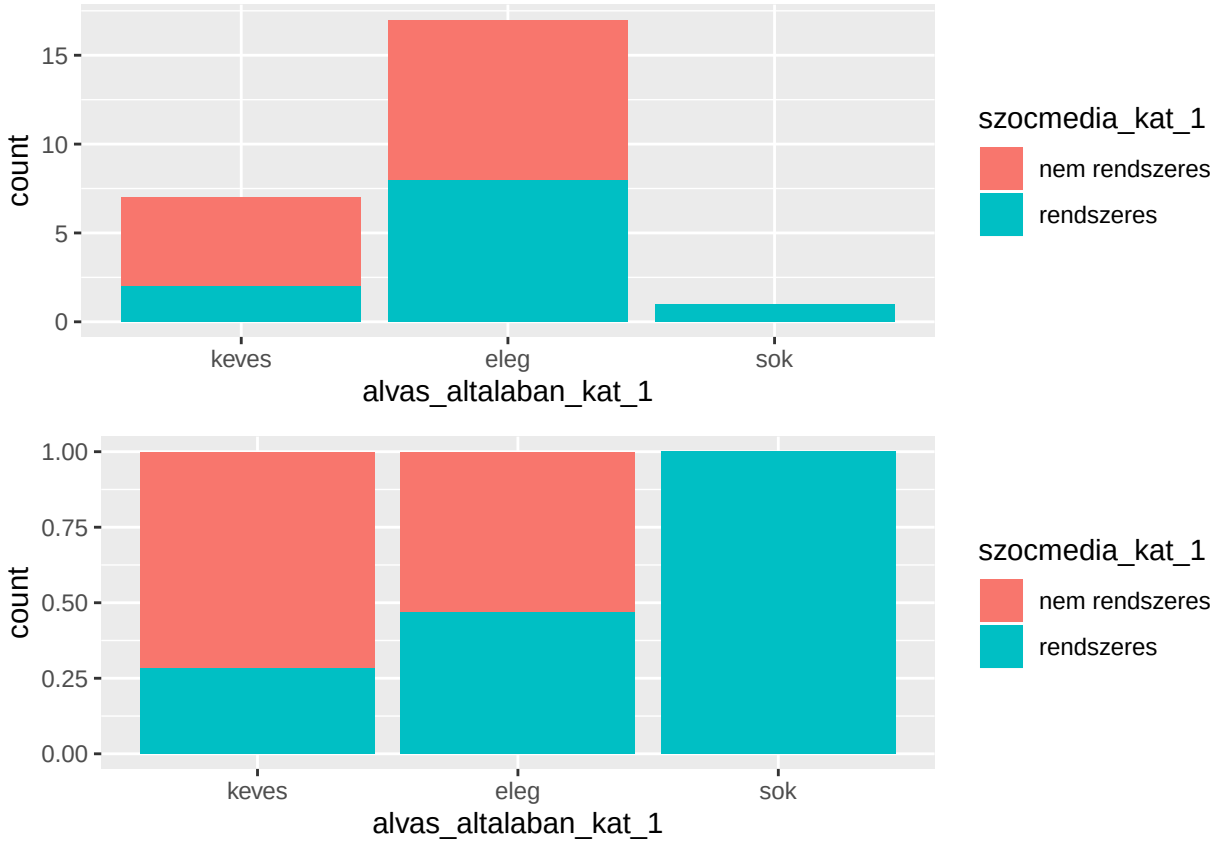
```
# az y tengelyen megvaltoztathatjuk a beosztast
# a scale_y_continuous(breaks = c(...))) funkcio hozzaadasaval
# a masodik abran az y tengelynek megvaltoztatjuk a feliratat

jegy_tipp_kat_hangulat_kat_plot_1 <-
orai_adat_corrected %>%
ggplot() +
  aes(x = jegy_tipp_kat, fill = hangulat_kat_1) +
  geom_bar() +
  scale_y_continuous(breaks = c(2, 4, 6, 8, 10, 12))

jegy_tipp_kat_hangulat_kat_plot_2 <-
orai_adat_corrected %>%
ggplot() +
  aes(x = jegy_tipp_kat, fill = hangulat_kat_1) +
  geom_bar(position = "fill") +
```

```
  ylab("proportion")

grid.arrange(jegy_tipp_kat_hangulat_kat_plot_1, jegy_tipp_kat_hangulat_kat_plot_2, nrow = 2)
```



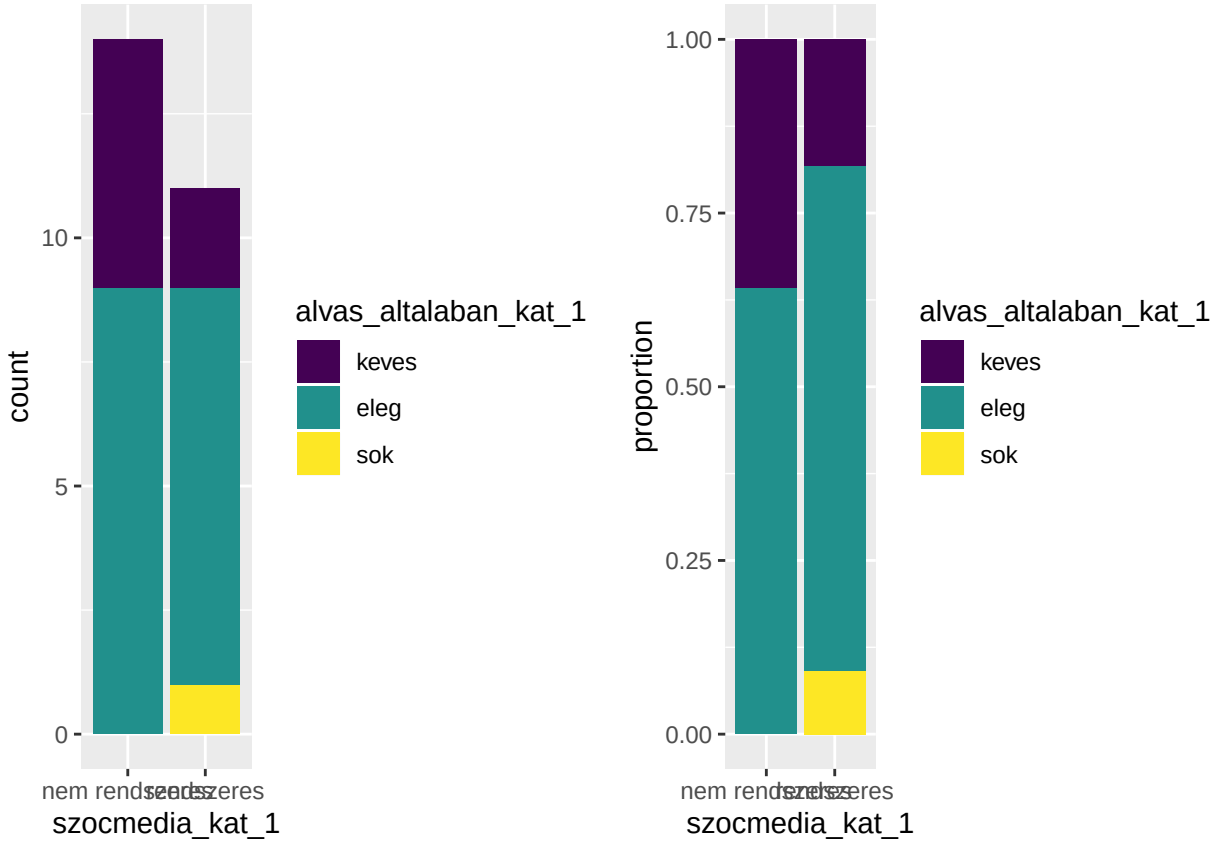
Ennel a megjelenitesnel fontos hogy ha az egyes megfigyelesek **keves megfigyelesbol allnak**, az abra megteveszto lehet, mert az abra nem jelzi a megfigyelesek szamat es igy azt, hogy milyen biztosak lehetunk az eredmenyben. Ilyen esetekben az egyik kategoriat ki lehet venni az abrarol, vagy a **szamossagot es a reszaranyt abrazolo abrakat egymas mellet** lehet bemutatni, hogy igy kiegeszitsek egymast. Ehhez hasznalhatjuk a **grid.arrange()** funkciot.

```
alvas_altalaban_szocmedia_plot_1 <-
orai_adat_corrected %>%
ggplot() +
  aes(x = alvas_altalaban_kat_1, fill = szocmedia_kat_1) +
  geom_bar()

alvas_altalaban_szocmedia_plot_2 <-
orai_adat_corrected %>%
ggplot() +
  aes(x = alvas_altalaban_kat_1, fill = szocmedia_kat_1) +
  geom_bar(position = "fill")

grid.arrange(alvas_altalaban_szocmedia_plot_1, alvas_altalaban_szocmedia_plot_2, nrow=2)
```

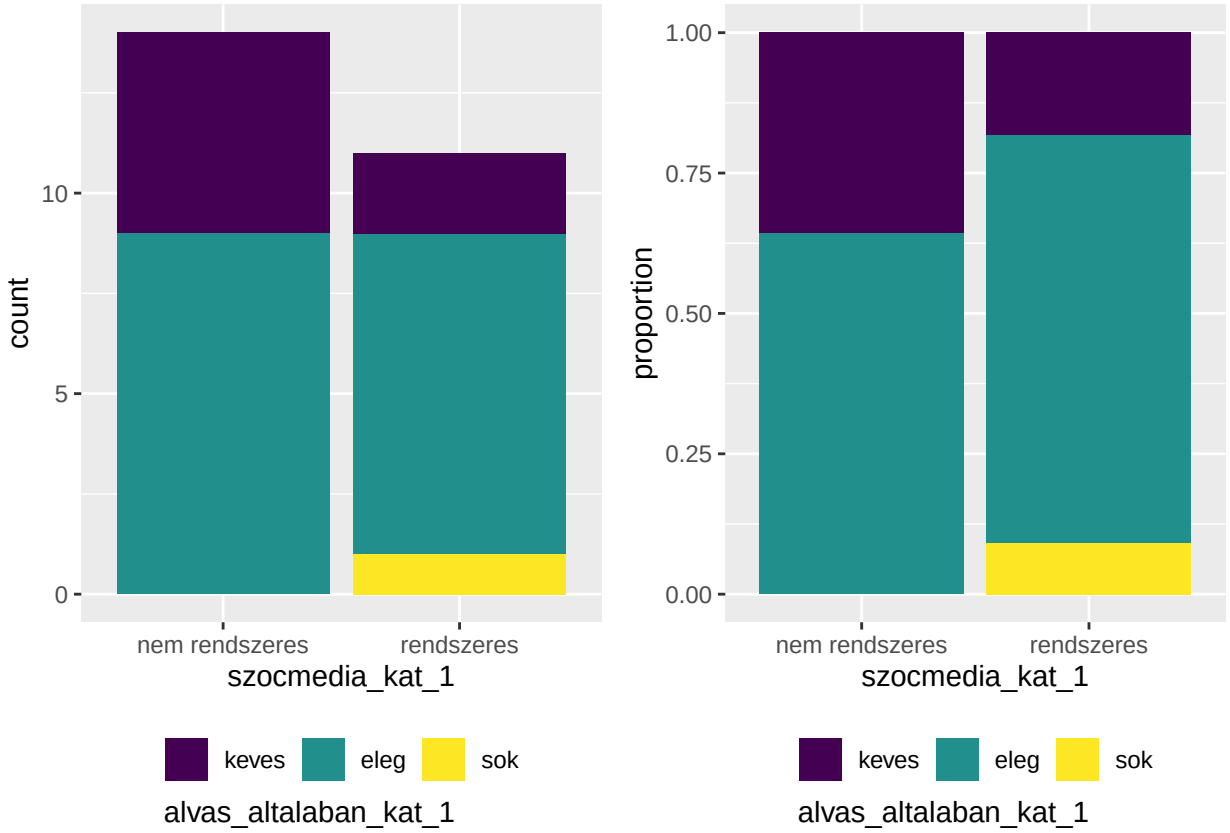


Az abra **interpretalhatosaga** attol fuggoen is **valtozhat**, hogy melyik valtozot tesszuk az x-tengelyre es melyiket szinkent abrazolva.

```
alvas_altalaban_szocmedia_plot_3 <-
orai_adat_corrected %>%
ggplot() +
  aes(x = szocmedia_kat_1, fill = alvas_altalaban_kat_1) +
  geom_bar()


alvas_altalaban_szocmedia_plot_4 <-
orai_adat_corrected %>%
ggplot() +
  aes(x = szocmedia_kat_1, fill = alvas_altalaban_kat_1) +
  geom_bar(position = "fill") +
  ylab("proportion")

grid.arrange(alvas_altalaban_szocmedia_plot_3, alvas_altalaban_szocmedia_plot_4, ncol=2)
```

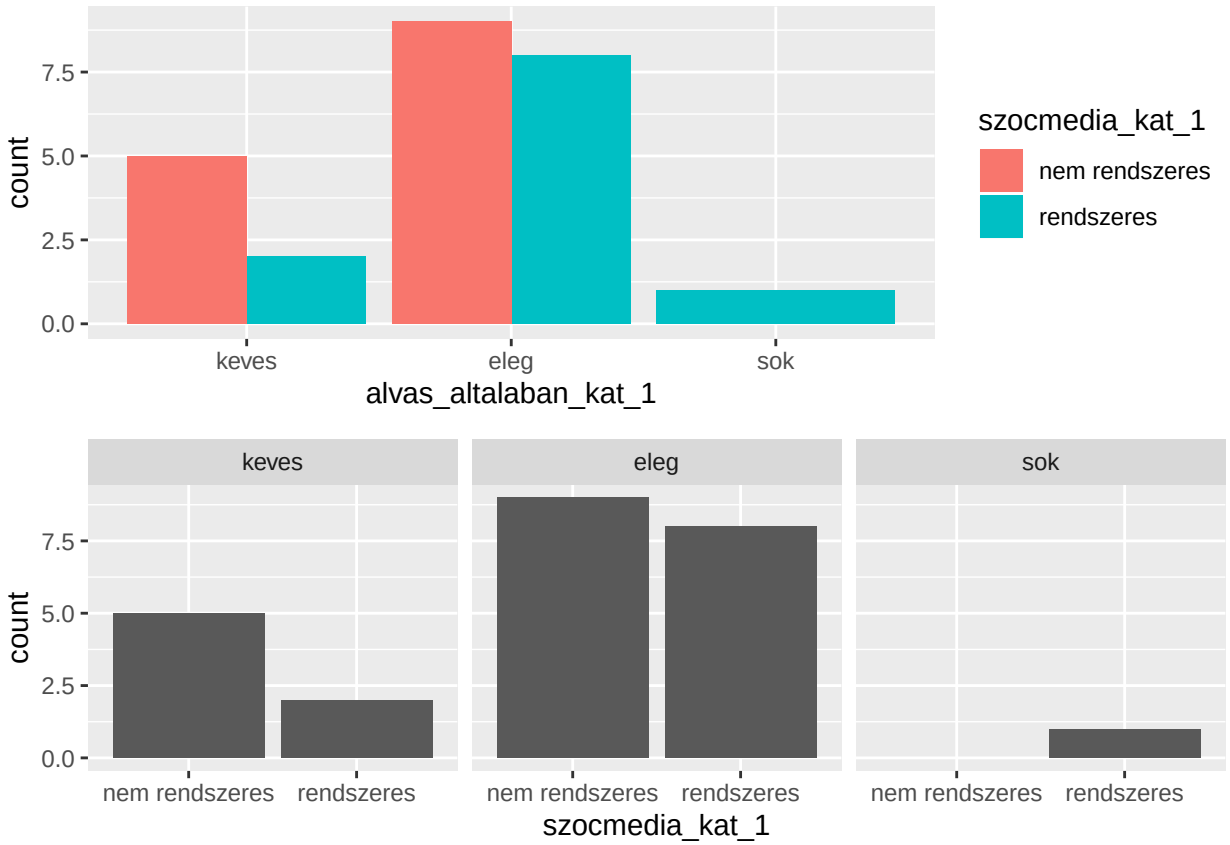



```
# a theme(legend.position) es a guides() funciok
# hasznalataval kontrollalhatjuk hogy hol es hogyan
# jelenjen meg a jelmagyarazat az abran

alvas_altalaban_szocmedia_plot_3 <-
orai_adat_corrected %>%
  ggplot() +
    aes(x = szocmedia_kat_1, fill = alvas_altalaban_kat_1) +
    geom_bar() +
    theme(legend.position="bottom") +
    guides(fill = guide_legend(title.position = "bottom"))


alvas_altalaban_szocmedia_plot_4 <-
orai_adat_corrected %>%
  ggplot() +
  aes(x = szocmedia_kat_1, fill = alvas_altalaban_kat_1) +
  geom_bar(position = "fill") +
  theme(legend.position="bottom") +
  guides(fill = guide_legend(title.position = "bottom")) +
  ylab("proportion")

grid.arrange(alvas_altalaban_szocmedia_plot_3, alvas_altalaban_szocmedia_plot_4, ncol=2)
```



Ujabb modja a barchart segitsegevel valo megjelenitesnek ha az oszlopok nem egymasra tornyozva, hanem **egymas mellett** jelennek meg, vagy ha a masodik valtozo szerint **kulon paneleken (facet)** jelennek meg.

```
alvas_altalaban_szocmedia_plot_1 <-
orai_adat_corrected %>%
ggplot() +
  aes(x = alvas_altalaban_kat_1, fill = szocmedia_kat_1) +
  geom_bar(position = "dodge")

alvas_altalaban_szocmedia_plot_2 <-
orai_adat_corrected %>%
ggplot() +
  aes(x = szocmedia_kat_1) +
  geom_bar() +
  facet_wrap(~ alvas_altalaban_kat_1)

grid.arrange(alvas_altalaban_szocmedia_plot_1, alvas_altalaban_szocmedia_plot_2, nrow=2)
```



Egy kategorikus es egy numerikus valtozo kapcsolata

Vizsgaljuk meg a **magassag** osszefuggeset azzal, hogy ez emberek mit tippeltek, milyen jegyet fognak kapni az elso oran (**jegy_tipp**).

Ezt megtehetjuk az atlagok csoportonkenti attekintesevel a **group_by()** es a **summarize(mean())** segitsegevel.

```
orai_adat_corrected %>%
  group_by(jegy_tipp_kat) %>%
    summarize(mean = mean(magassag))
```

```
## # A tibble: 3 x 2
##   jegy_tipp_kat  mean
##   <ord>         <dbl>
## 1 kozepes (3)    168
## 2 jo (4)         166.
## 3 kivalo (5)     172.
```

A ket valtozo kapcsolatat megvizsgalhatjuk **abrakkal** is. Pl. hasznalhatjuk a

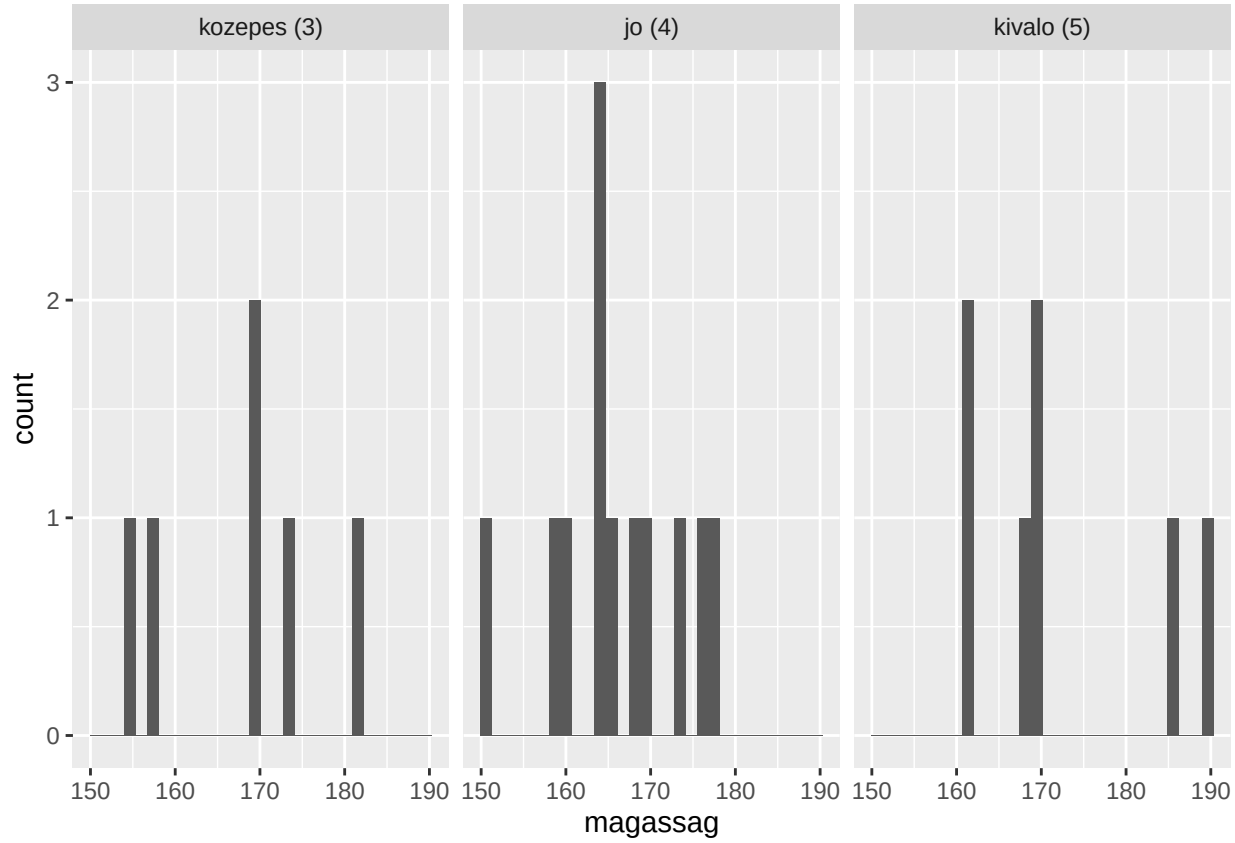
- **facet_wrap()** fuggvenyt a egy **geom_histogram()** vagy **geom_dotplot()** -al kobinalva
- a **geom_boxplot()** -ot
- esetleg hasznalhatunk egy egymasra illesztett **geom_density()** plot-ot.
- talan ebben az esetben a legtisztabb kepet a **geom_violin()** mutatja, ami a geom_boxplot() es a geom_density() keverekenek tekintheto. Ezt kiegeszithetunk egy **geom_point()** -al, hogy pontosan latsszon, hany megfigyelesen alapulnak az abra adatai.

Mindig erdemes **tobb megkozelitest** is hasznalni az adat-exploracio kozben, hogy minel reszletesebb kepet

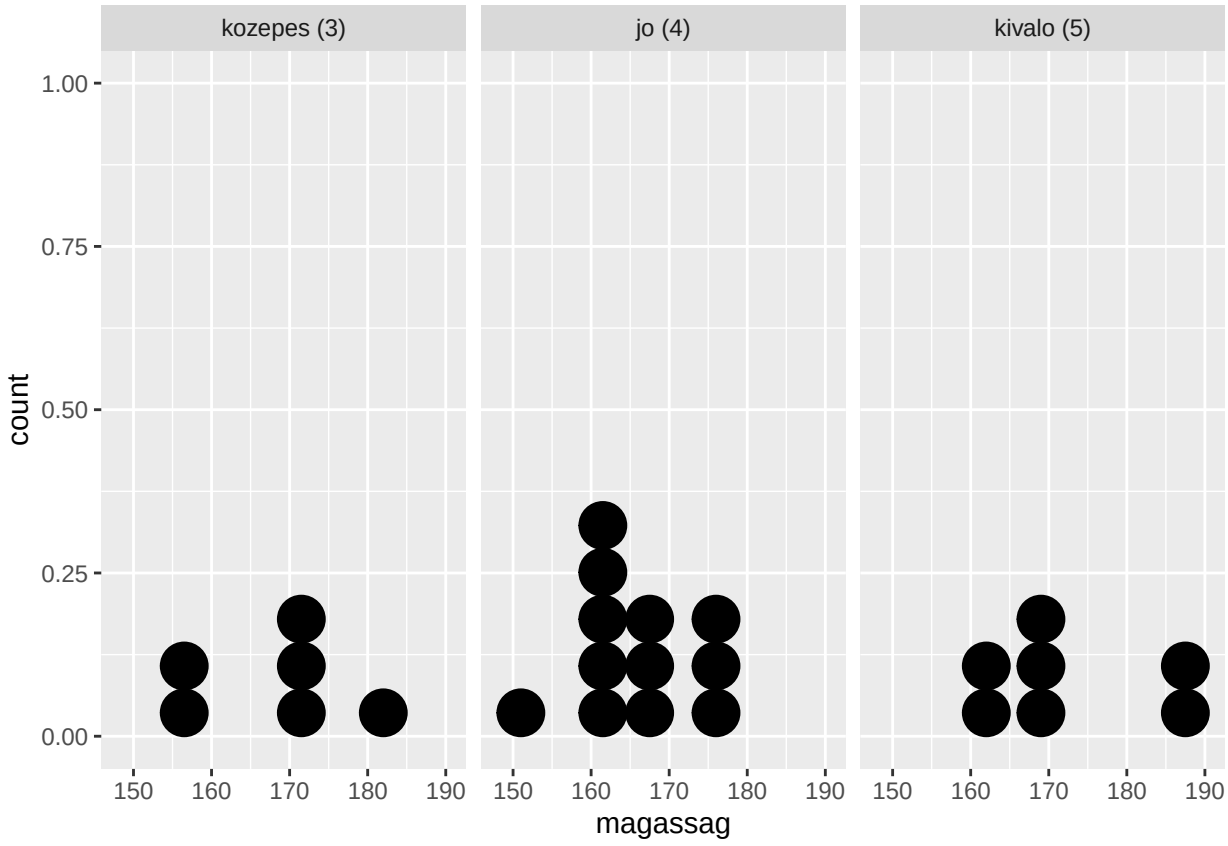
kaphassunk, es csokkentsuk a valoszinuseget hogy egyik vagy masik megkozelites hianyossagai felrevezetnek minket.

```
orai_adat_corrected %>%
  ggplot() +
    aes(x = magassag) +
    geom_histogram() +
    facet_wrap(~ jegy_tipp_kat)
```

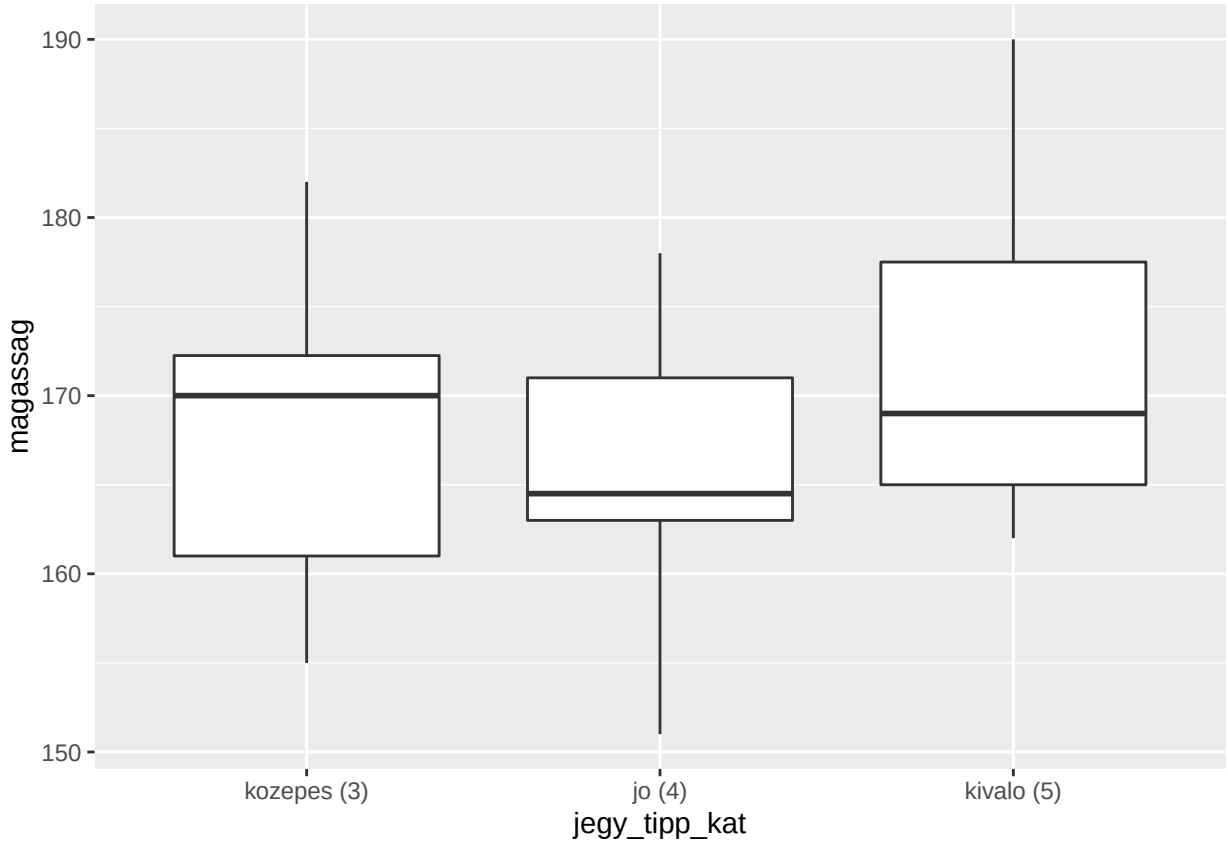
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



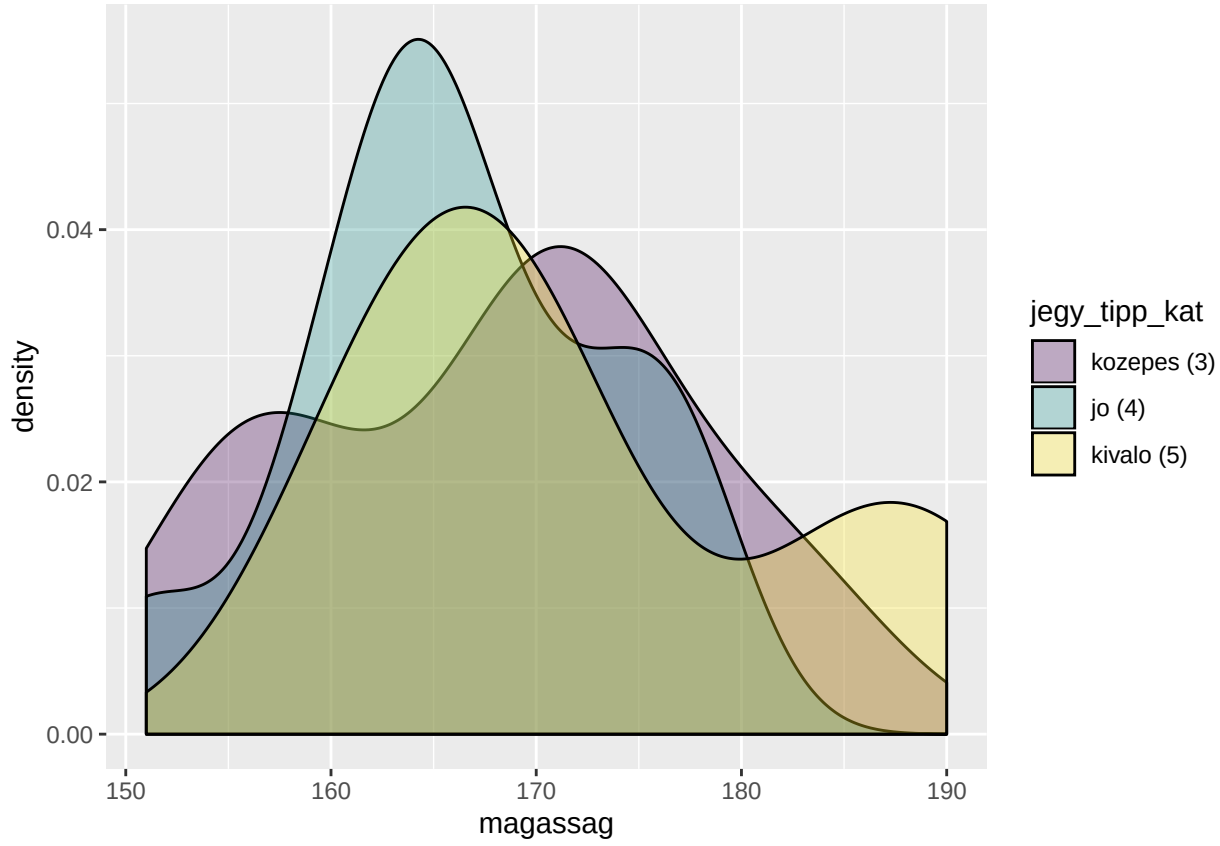

```
orai_adat_corrected %>%
  ggplot() +
    aes(x = magassag) +
    geom_dotplot(binwidth = 6) +
    facet_wrap(~ jegy_tipp_kat)
```



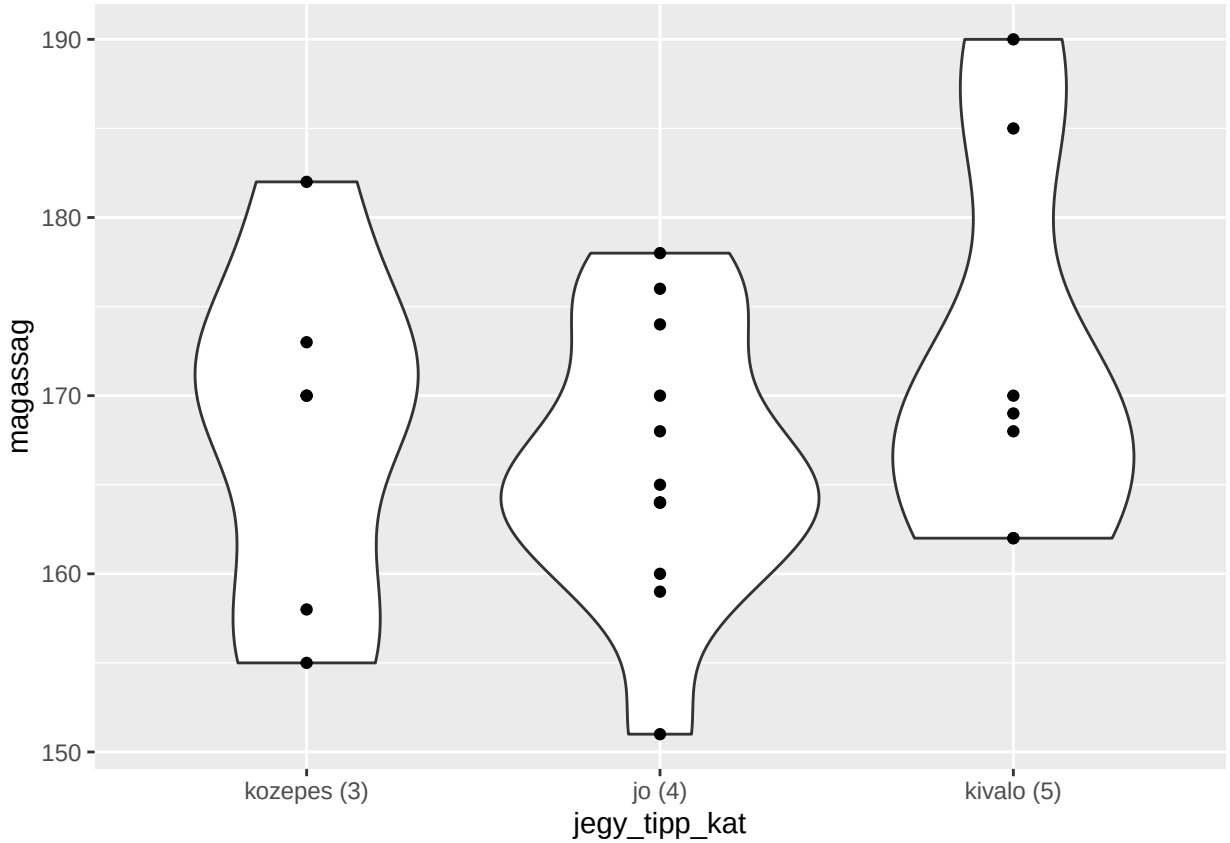

```
orai_adat_corrected %>%
  ggplot() +
    aes(x = jegy_tipp_kat, y = magassag) +
    geom_boxplot()
```




```
orai_adat_corrected %>%
  ggplot() +
    aes(x = magassag, fill = jegy_tipp_kat) +
    geom_density(alpha = 0.3)
```




```
orai_adat_corrected %>%
  ggplot() +
    aes(x = jegy_tipp_kat, y = magassag) +
    geom_violin() +
    geom_point()
```



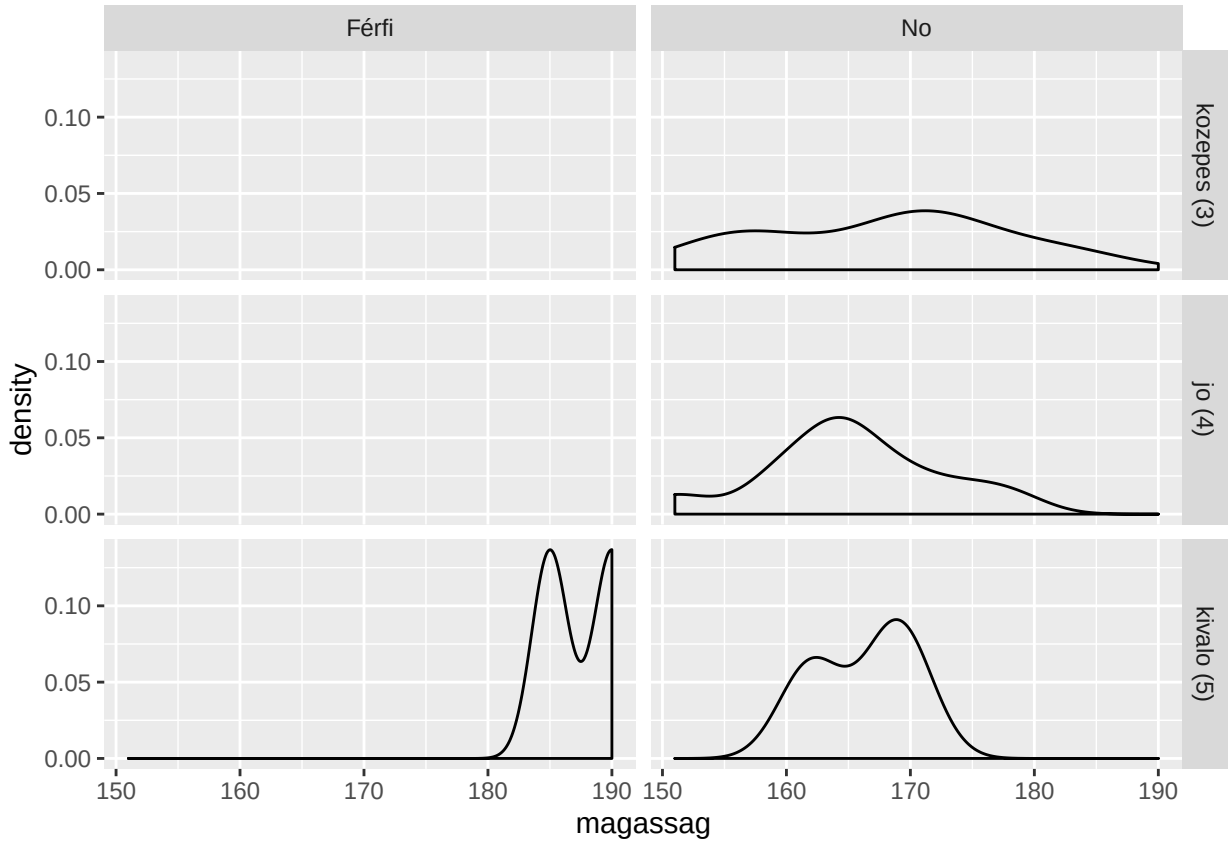
A fenti abran latszik, hogy bar az atlagok a fenti tablazatban kulonboztek, a maguknak kivalo pontot josolok tobbsege 170 cm koruli, de van ket **kiurgo magassagu szemely**, aki az atlagot felhuzza ebben a csoportban.

Az is elkepzelheto hogy egy **nemi hatast** latunk az eredmenyekben, hiszen a ferfiak magasabbak, es elkepzelheto, hogy nagyobb az adatelemzessel kapcsolatos onbizalmuk is. Probaljuk meg a **ferfiakra es a nokre kulon elkesziteni az abrat**.

Itt mar **harom valtozo** kapcsolatat abrazoljuk, amihez a facet_grid() funkciot lehet hasznalni, vagy kulonbozo esztetikai elemeket (aes()) lehet a kulonbozo valtozokhoz rendelni.

```
orai_adat_corrected %>%
  ggplot() +
    aes(x = magassag) +
    geom_density() +
    facet_grid(jegy_tipp_kat ~ nem)
```

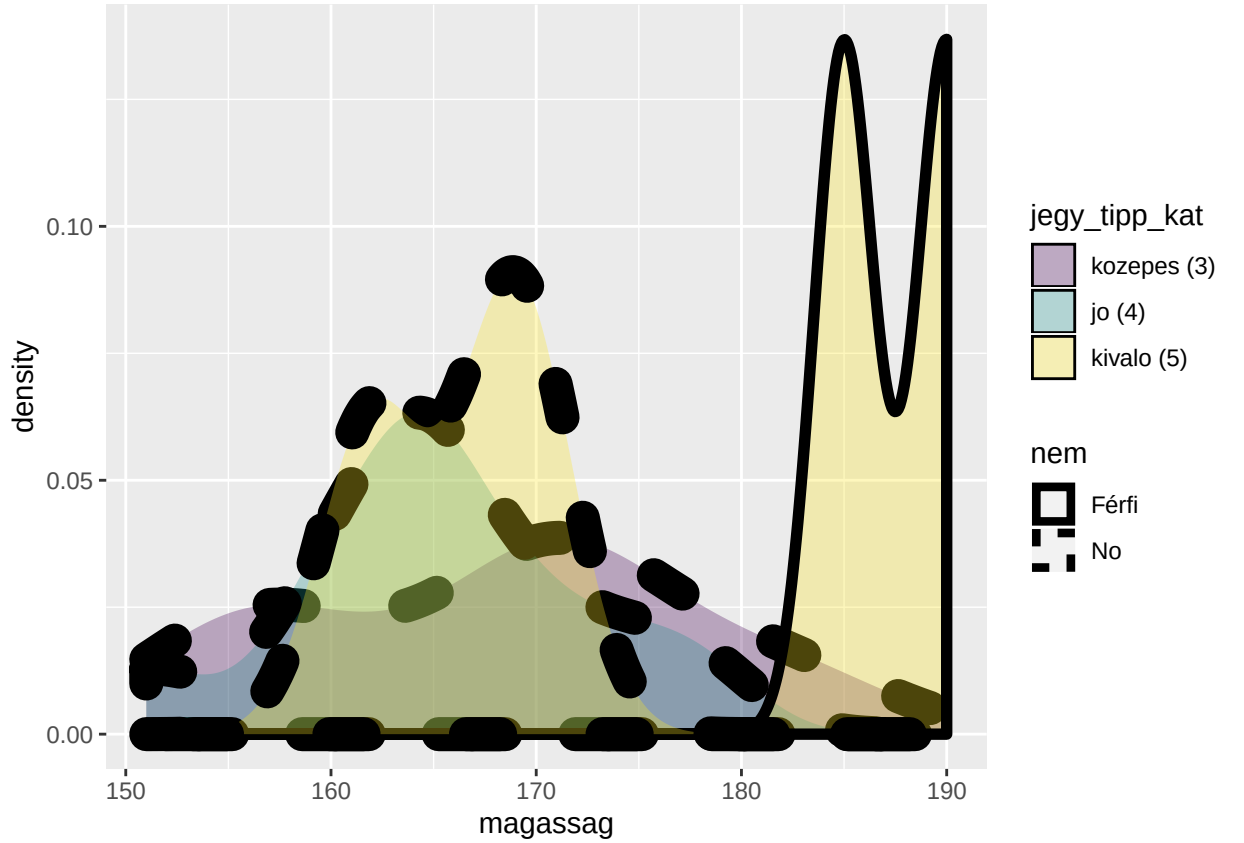
```
## Warning: Groups with fewer than two data points have been dropped.
```




```
orai_adat_corrected %>%
  ggplot() +
    aes(x = magassag, fill = jegy_tipp_kat, size = nem, lty = nem) +
    geom_density(alpha = 0.3)
```

```
## Warning: Using size for a discrete variable is not advised.
```

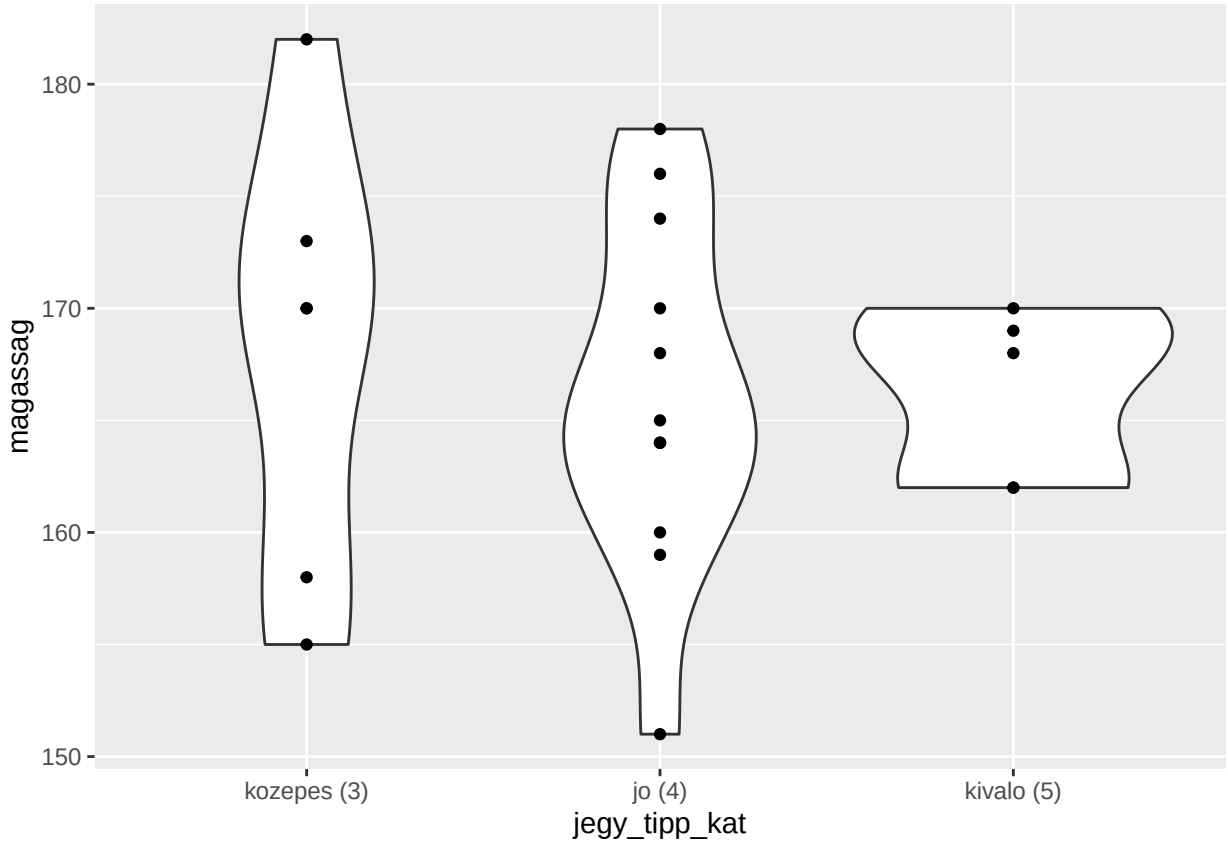
```
## Warning: Groups with fewer than two data points have been dropped.
```



Ha szeretnenk **kizarni az elemzesunkbol** ezeket az extrem ertekekt, a **filter()** funkcio beekelesevel a pipe-ba megepithetjuk a fenti abrankat es tablazatokat ugy, hogy csak a 185 cm-nel alacsonyabb emberekre fogkuszalunk.

Igy mar eltunik a korabbi atlagok kozotti kulonbseg.

```
orai_adat_corrected %>%
  filter(magassag < 185) %>%
    ggplot() +
      aes(x = jegy_tipp_kat, y = magassag) +
      geom_violin() +
      geom_point()
```




```
orai_adat_corrected %>%
  filter(magassag < 185) %>%
    group_by(jegy_tipp_kat) %>%
      summarize(mean = mean(magassag))
```

```
## # A tibble: 3 x 2
##   jegy_tipp_kat  mean
##   <ord>         <dbl>
## 1 kozepes (3)    168
## 2 jo (4)         166.
## 3 kivalo (5)     166.
```

Gyakorlas

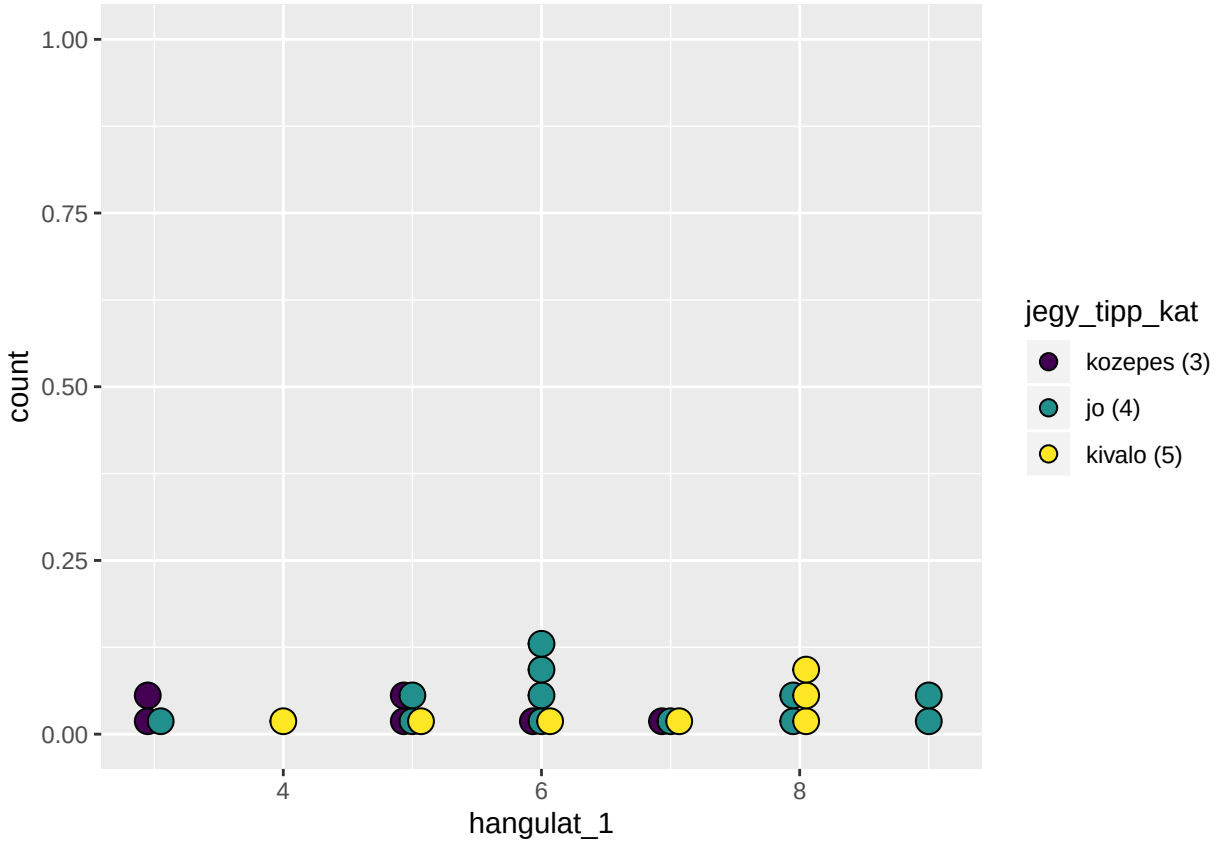
Hasznald a fent tanult modszereket, hogy megvizsgald a **jegy_tipp_kat** es a **hangulat_1** valtozok kozotti osszefuggest.

- hasznald a fenti geomokat, es keszits legalabb ket kulonbozo abrat mas-mas geomokkal

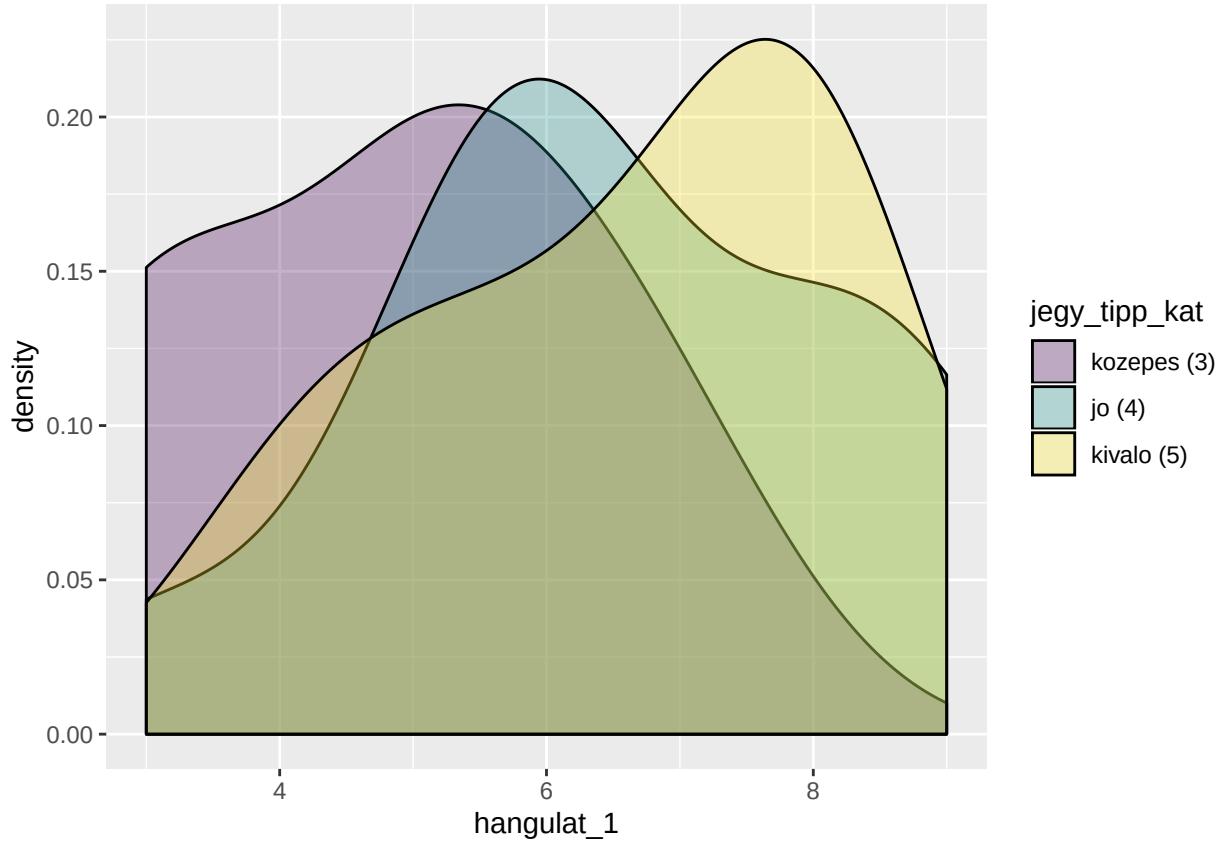
---

```
orai_adat_corrected %>%
  ggplot() +
    aes(x = hangulat_1) +
    geom_dotplot(aes(fill = jegy_tipp_kat), position = "dodge")
```

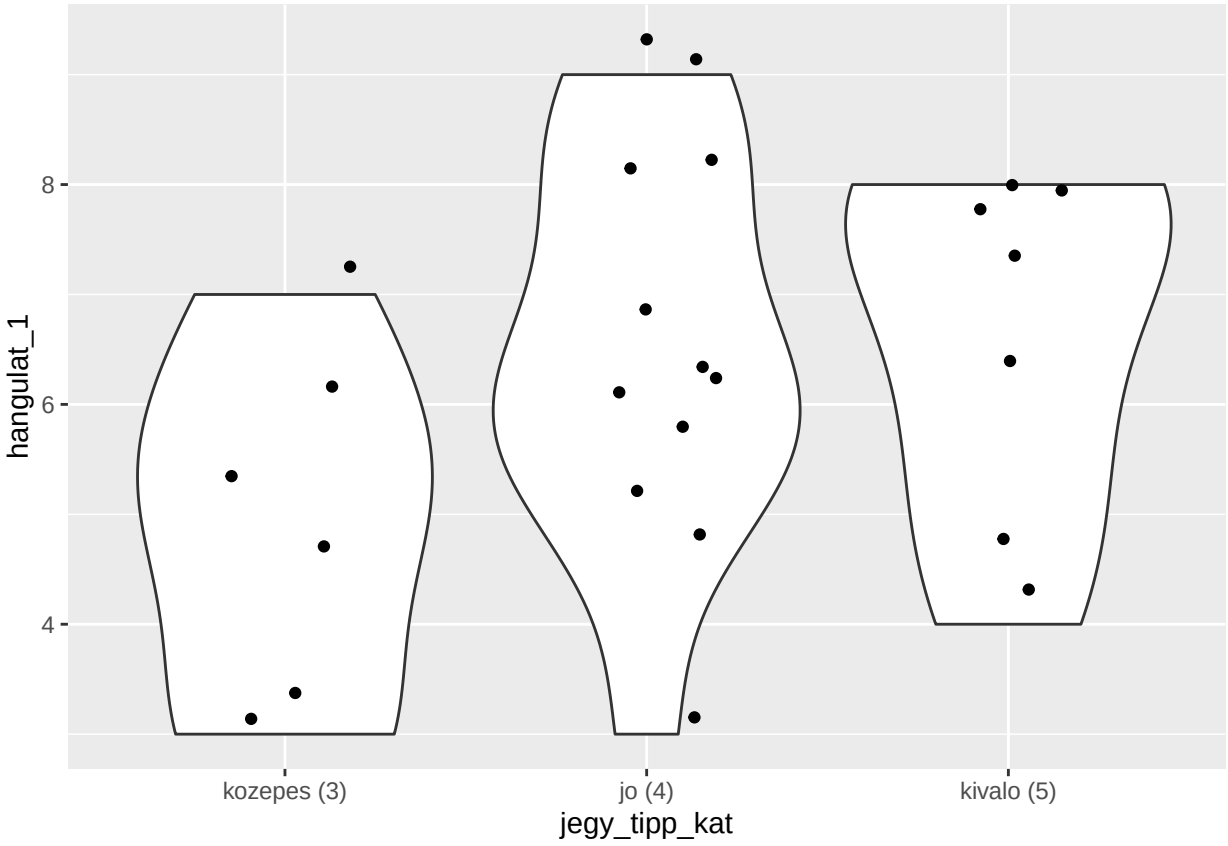
```
## `stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.
```




```
orai_adat_corrected %>%
  ggplot() +
    aes(x = hangulat_1, fill = jegy_tipp_kat) +
    geom_density(alpha = 0.3)
```




```
orai_adat_corrected %>%
  ggplot() +
    aes(x = jegy_tipp_kat, y = hangulat_1) +
    geom_violin() +
    geom_jitter(width = 0.2)
```



Ket numerikus valtozo kapcsolata

Ket numerikus valtozo kozotti kapcsolat jellemzesere altalaban a korrelacios egyutthatot szoktuk hasznalni (cor()). A **cor()** funkciot akar tobb mint ket valtozo paronkenti korrelaciojanak meghatarozasara is lehet hasznalni.

A **drop_na()** funkcioval kiejthetjuk azokat a megfigyeleseket, ahol a valtozok barmelyikeben hianyzo adat (NA) van. Ha ezt nem tesszuk meg, a cor() fuggveny NA eredmenyt adhna ha valamelyik valtozoban NA-val talalkozik.

```
orai_adat_corrected %>%
  select(c(alvas_tegnap_1, alvas_tegnap_3)) %>%
    drop_na() %>%
      cor()
```

```
##                alvas_tegnap_1 alvas_tegnap_3
## alvas_tegnap_1      1.0000000      0.5750546
## alvas_tegnap_3      0.5750546      1.0000000
```

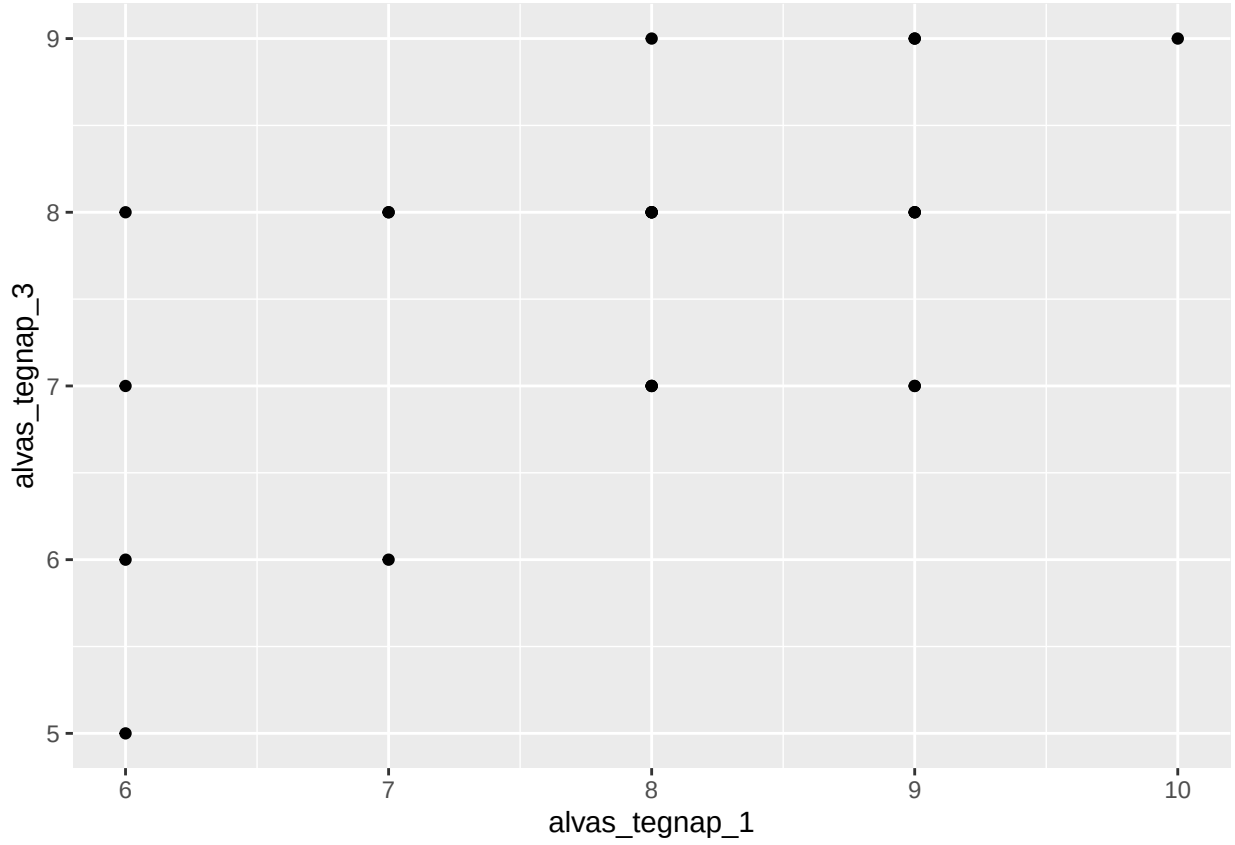
```
orai_adat_corrected %>%
  select(c(alvas_altalaban_1, alvas_tegnap_1, alvas_tegnap_3)) %>%
    drop_na() %>% #
    cor()
```

```
##                   alvas_altalaban_1 alvas_tegnap_1 alvas_tegnap_3
## alvas_altalaban_1         1.0000000      0.7101496      0.6740334
## alvas_tegnap_1            0.7101496      1.0000000      0.5750546
## alvas_tegnap_3            0.6740334      0.5750546      1.0000000
```

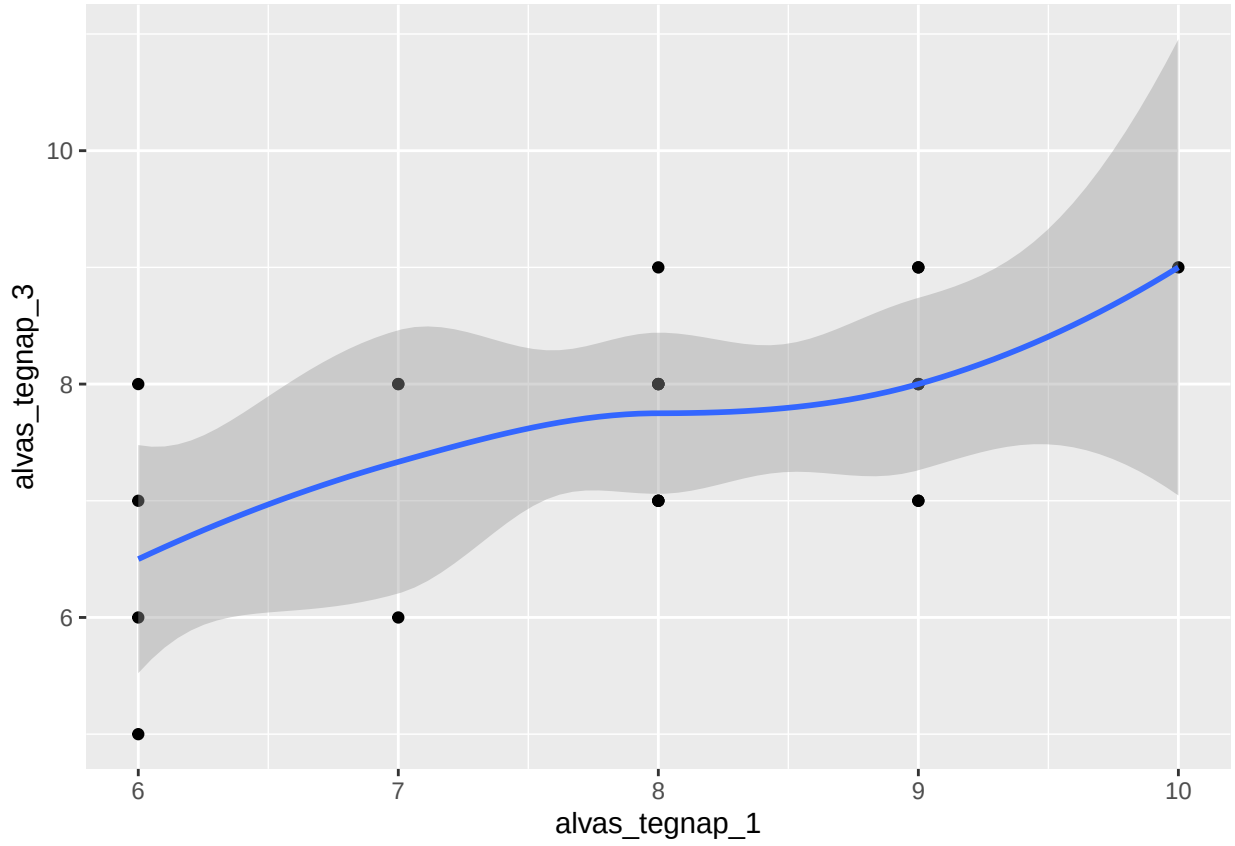
A numerikus valtozok kozotti kapcsolatot altalaban pont diagrammal szoktuk abrazolni (**geom_point()**)

A **geom_smooth()** layer hozzaadasaval kaphatunk a pontok kozott meghuzodo trendrol egy kepet. A kek vonal az ugyevezett trandvonal, a szurke sav a konfidencia intervallum. Ezekrol kesobb meg reszletesebben beszelunk majd

```
orai_adat_corrected %>%
  ggplot() +
    aes(x = alvas_tegnap_1, y = alvas_tegnap_3) +
    geom_point()
```




```
orai_adat_corrected %>%
  ggplot() +
    aes(x = alvas_tegnap_1, y = alvas_tegnap_3) +
    geom_point() +
    geom_smooth()
```

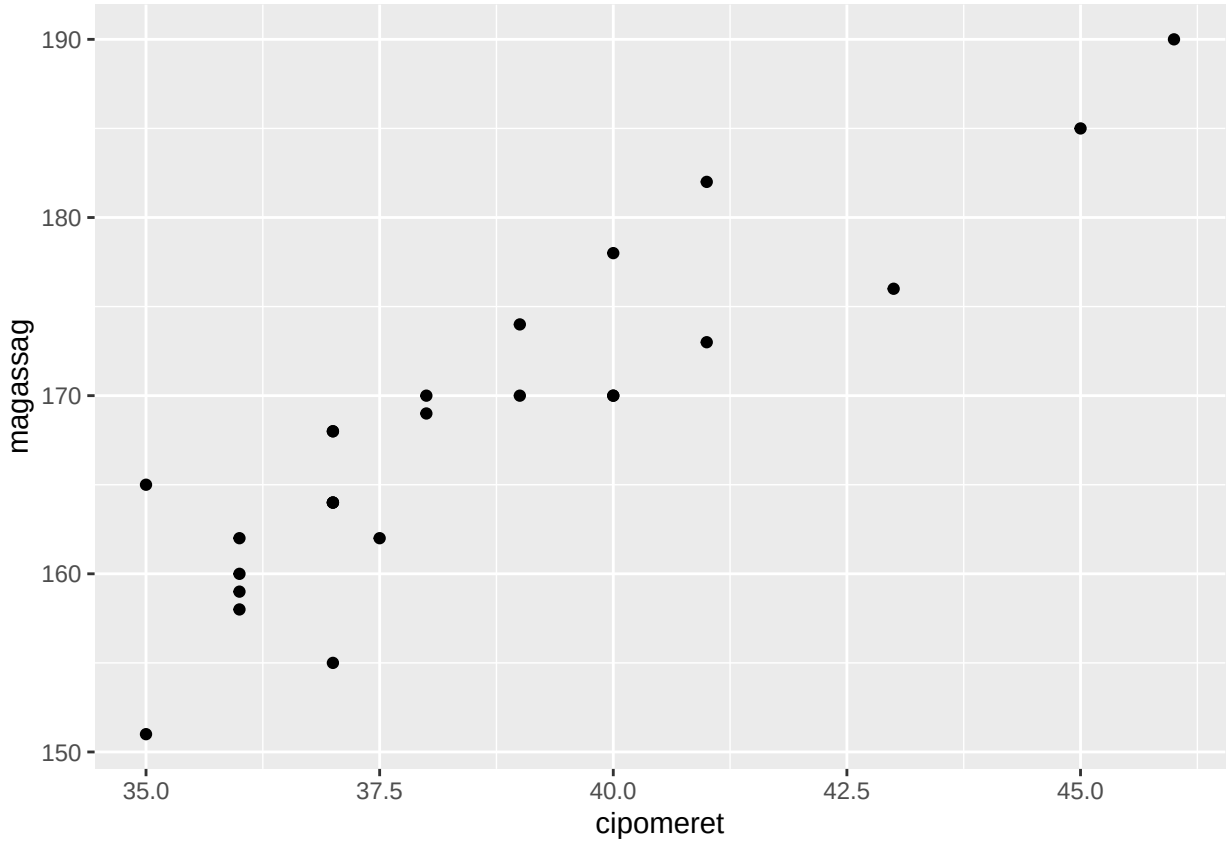


Gyakorlas

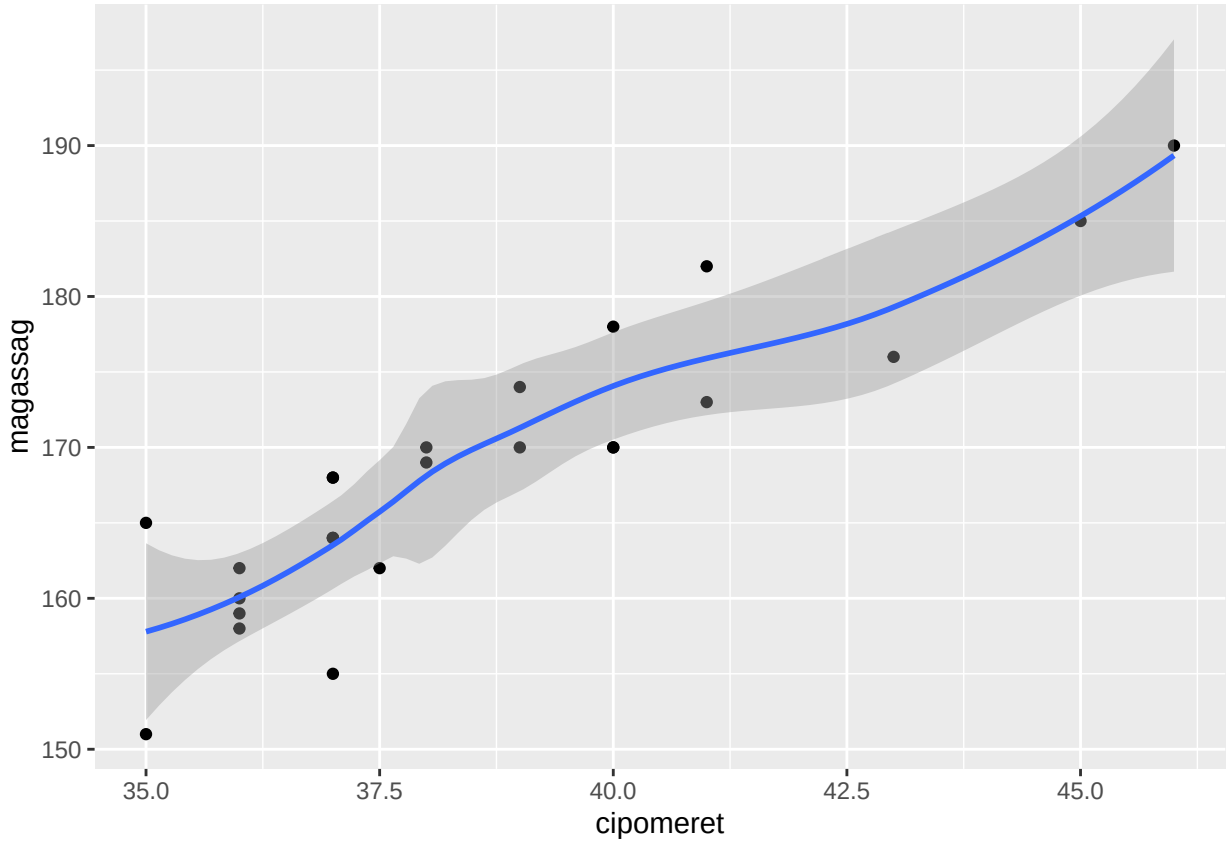
Milyen eros a kapcsolat a magassag es a cipomeret kozott?

- hatarozd meg a korrelacios egyutthatot a *magassag* es *cipomeret* valtozok kozott
- abrazold a *magassag* es *cipomeret* valtozok kapcsolatat

```
orai_adat_corrected %>%
  ggplot() +
    aes(x = cipomeret, y = magassag) +
    geom_point()
```

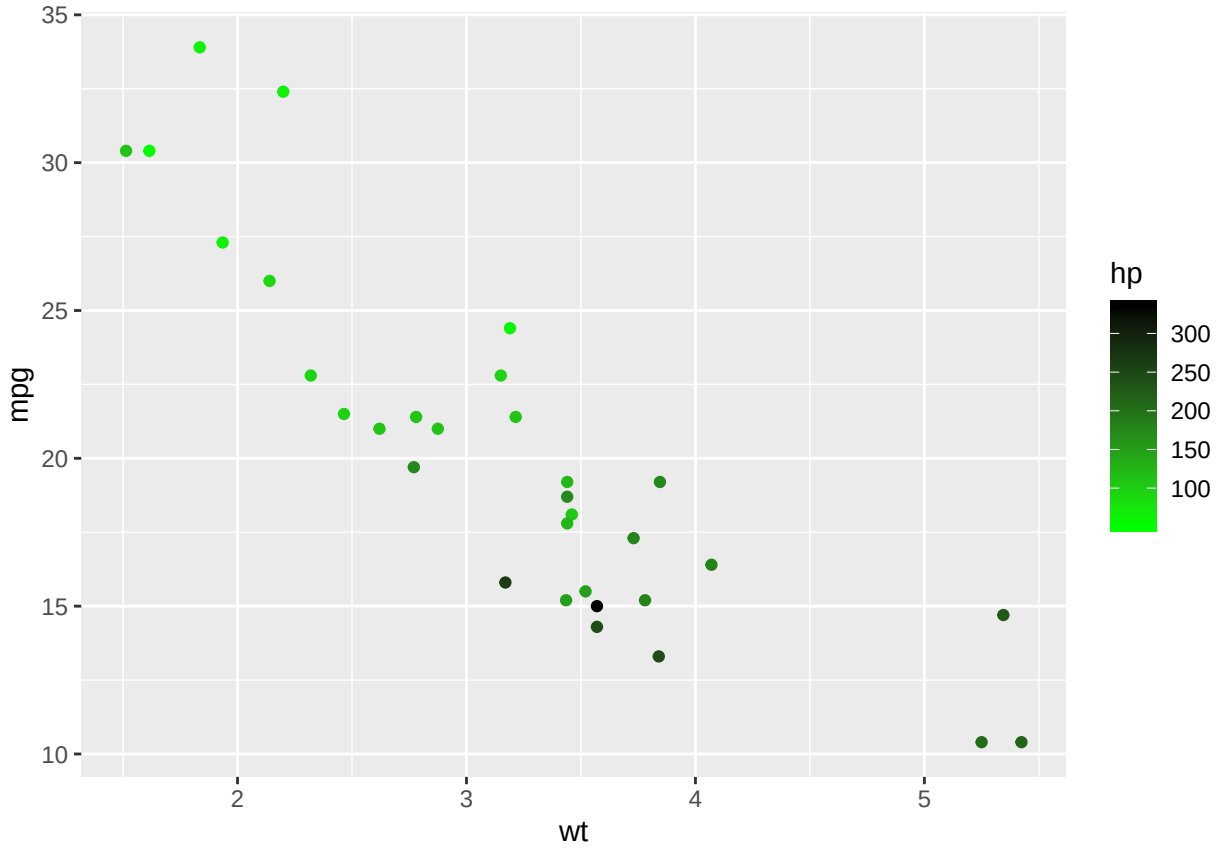



```
orai_adat_corrected %>%
  ggplot() +
    aes(x = cipomeret, y = magassag) +
    geom_point() +
    geom_smooth()
```



Tobb folytonos valtozo kapcsolata megjelenitheto peldaul ugy, hogy az egyik valtozot egy szinskalahoz rendeljuk az alabbi modon.

```
plot_szines <- ggplot(data = mtcars, aes(x = wt, y = mpg, col = hp)) +
  geom_point()
plot_szines +
  scale_colour_gradientn(colours=c("green","black"))
```


35
30
25
20
15
10
mpg
2
3
4
5
wt
hp
300
250
200
150
100