

PSZB17-210 - Seminar_4

Zoltan Kekecs

Marcus 4, 2020

4. Ora - Adatexploracio

Az ora celja az adatexploracios modszerek elsajaitasa.

Package-ek betoltese

A kovetkező package-ekre lesz szuksegunk

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

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** szerkezetével es tartalmával.

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

```
orai_adat
```

```
## # A tibble: 25 x 31
##   kitoltesiido ID      nem   életkor 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 változók **leirostatisztikaira** (descriptive statistics) vagyunk kíváncsiak, kerhetjük ezt a már tanult módon.

Peldaul lekerhetjük a változó alapvető legalacsonyabb és legmagasabb értéket, átlagát, medianját, a kvartiliseket, és hogy hány hiányzó adat van (ha van) a **summary()** funkcióval (miután a select funkcióval kiválasztottuk, melyik változóra vagyunk kíváncsiak)

```
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 megkaphatjuk ugyanezt az összes változóra, ha ugyanezt az egész adattablára futtatjuk le. Persze a karakter osztályba tartozó változók mindezeknek a leíró statisztikáknak nincs értelme, ott csak a class információt kaptjuk az output-ban.

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

Gyakorlás

- Hány órát gyakorolt az az ember, aki a legtöbb órát gyakorolta a második és a harmadik óra között (*stat_gyakorlas_3* változó)?
 - Mekkora volt az átlagos energiaszint az első és mekkora a harmadik órán (*energiaszint_1* és *energiaszint_3* változók)?
-

Meg több leíró statisztika

A **Psych** package segítségével a **describe()** funkció meg több hasznos információt adhat

```
describe(orai_adat)
```

```
##              vars  n   mean    sd median trimmed  mad   min  max  range
## kitoltesiido*      1 25    NaN    NA     NA      NaN   NA   Inf -Inf  -Inf
## ID*                2 25 32.00    NA    32   32.00 0.00 32.00  32   0.00
## nem*               3 25    NaN    NA     NA      NaN   NA   Inf -Inf  -Inf
## életkor*           4 25    NaN    NA     NA      NaN   NA   Inf -Inf  -Inf
## alvas_szukseges     5 25   8.32  0.90     8    8.33 1.48   6.00  10   4.00
```

## nemalvas_oka*	6	24	NaN	NA	NA	NaN	NA	Inf	-Inf	-Inf
## memoria	7	25	7.48	1.26	8	7.57	1.48	4.00	10	6.00
## memoria_ront_A*	8	23	NaN	NA	NA	NaN	NA	Inf	-Inf	-Inf
## memoria_ront_B*	9	19	NaN	NA	NA	NaN	NA	Inf	-Inf	-Inf
## memoria_ront_C*	10	19	NaN	NA	NA	NaN	NA	Inf	-Inf	-Inf
## memoria_ront_D*	11	14	NaN	NA	NA	NaN	NA	Inf	-Inf	-Inf
## memoria_ront_E*	12	7	NaN	NA	NA	NaN	NA	Inf	-Inf	-Inf
## memoria_ront_F*	13	4	NaN	NA	NA	NaN	NA	Inf	-Inf	-Inf
## jegy_tipp	14	25	4.04	0.73	4	4.05	1.48	3.00	5	2.00
## valasztott_szam	15	25	6.56	2.52	7	6.62	2.97	2.00	10	8.00
## tippelt_gyakori_szam	16	25	5.80	2.58	6	5.76	2.97	2.00	10	8.00
## magassag	17	25	161.07	34.32	168	166.62	8.90	1.82	190	188.18
## cipomeret	18	25	38.94	3.34	38	38.60	2.97	35.00	47	12.00
## memoriateszt_hely*	19	25	NaN	NA	NA	NaN	NA	Inf	-Inf	-Inf
## alvas_tegnap_1	20	25	7.92	1.26	8	7.90	1.48	6.00	10	4.00
## alvas_tegnap_3	21	23	7.61	1.03	8	7.68	1.48	5.00	9	4.00
## alvas_altalaban_1	22	25	7.04	1.02	7	7.10	1.48	5.00	9	4.00
## alvas_altalaban_3	23	23	7.13	0.63	7	7.16	0.00	6.00	8	2.00
## energiaszint_1	24	25	5.48	2.16	5	5.62	2.97	1.00	8	7.00
## energiaszint_3	25	23	6.17	1.92	7	6.26	1.48	3.00	9	6.00
## szocmedia_1	26	25	1.96	1.57	1	1.81	1.48	0.00	6	6.00
## szocmedia_3*	27	24	1.96	1.36	2	1.79	1.48	0.00	5	5.00
## hangulat_1	28	25	6.12	1.79	6	6.14	1.48	3.00	9	6.00
## hangulat_3	29	23	5.91	2.21	6	6.16	2.97	1.00	9	8.00
## stat_tudasszint_3	30	23	4.09	2.02	5	4.16	1.48	0.00	8	8.00
## stat_gyakorlas_3	31	23	4.59	3.54	3	4.08	2.22	0.00	14	14.00
##			skew	kurtosis	se					
## kitoltesiido*			NA	NA	NA					
## ID*			NA	NA	NA					
## nem*			NA	NA	NA					
## életkor*			NA	NA	NA					
## alvas_szukseges	-0.30		0.22	0.18						
## nemalvas_oka*			NA	NA	NA					
## memoria	-0.67		0.89	0.25						
## memoria_ront_A*			NA	NA	NA					
## memoria_ront_B*			NA	NA	NA					
## memoria_ront_C*			NA	NA	NA					
## memoria_ront_D*			NA	NA	NA					
## memoria_ront_E*			NA	NA	NA					
## memoria_ront_F*			NA	NA	NA					
## jegy_tipp	-0.06		-1.22	0.15						
## valasztott_szam	-0.38		-1.22	0.50						
## tippelt_gyakori_szam	0.11		-1.16	0.52						
## magassag	-3.95		15.59	6.86						
## cipomeret	0.99		-0.05	0.67						
## memoriateszt_hely*			NA	NA	NA					
## alvas_tegnap_1	-0.22		-1.12	0.25						
## alvas_tegnap_3	-0.63		-0.11	0.22						
## alvas_altalaban_1	-0.30		-0.71	0.20						
## alvas_altalaban_3	-0.07		-0.63	0.13						
## energiaszint_1	-0.27		-1.21	0.43						
## energiaszint_3	-0.38		-1.35	0.40						
## szocmedia_1	1.06		0.20	0.31						
## szocmedia_3*	1.00		-0.11	0.28						

```
## hangulat_1      -0.17    -0.98 0.36
## hangulat_3      -0.74    -0.30 0.46
## stat_tudasszint_3 -0.18    -0.99 0.42
## stat_gyakorlas_3  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    No
## [13] No    No    No    No    No    No    No    No    Férfi No    No    Férfi
## [25] 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 szeretnénk **kizarni** bizonyos **faktorszinteket** az elemzésből. Pl. ha valamelyik faktor szintből nagyon keves megfigyeles van, mondjuk 23 even felüli életkoruakból, oket lehet hogy szeretnénk kizarni a későbbi elemzésekben hogy egyszerűsítsuk az eredményeink értelmezését. Ezt a már korábban tanult **filter()** funkció segítségével könnyedén megtehetjük, azonban arra figyelni kell, hogy az R megjegyzi a faktorszinteket, és azt azt követően is a **változohoz rendelve tartja**, miután már az adott faktorszintből nincs egy megfigyeles sem az adattáblában.

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

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

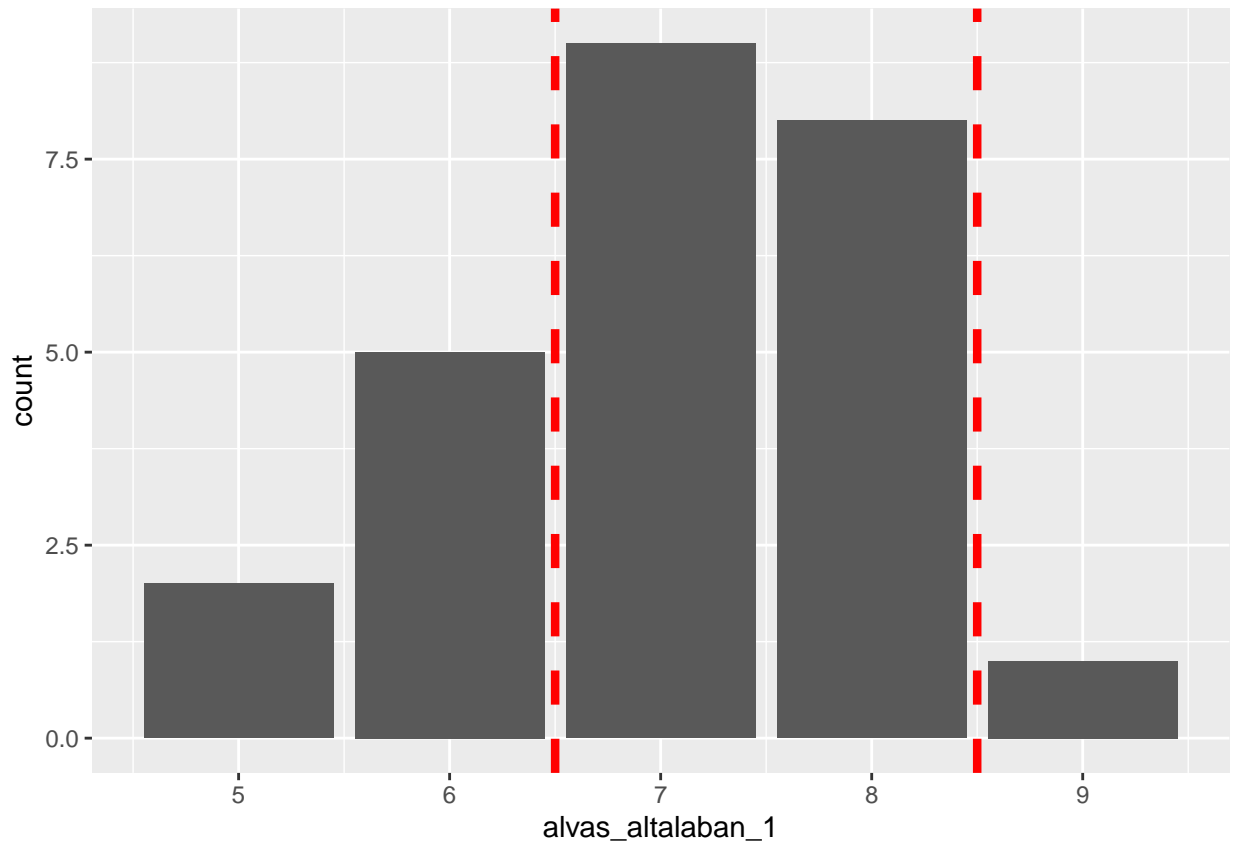
Igy ezeket a szinteket ejteni szoktuk a **droplevels()** funkcióval.

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

```
##  életkor
##  0-22:22
```

Elofordul, hogy egy **numerikus változót akarunk átalakítani faktorra**, pl. elképzelhető hogy össze akarjuk hasonlítani azokat akik 6 vagy kevesebb órát alszanak általában azokkal akik 7 vagy 8, vagy még több órát 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 használhatjuk a **mutate()** és **recode()** funkciók kombinációját hogy csináljunk egy új változót. Ebbe a kódba beleépítettem a **factor()** funkciót is, hogy azonnal meghatározzuk, hogy ez az új változó egy faktor, és nem egy egyszerű karaktervektor. A **factor()** funkció nélkül is lefut a kód, de akkor még kellene egy külön sor ahol megadjuk hogy ez egy faktorváltozó.

```
orai_adat = orai_adat %>%
  mutate(alvas_atalaban_kat_1 = factor(
    recode(
      alvas_atalaban_1,
      "5" = "keves",
      "6" = "keves",
      "7" = "eleg",
      "8" = "eleg",
      "9" = "sok"
    )
  ))

levels(orai_adat$alvas_atalaban_kat_1)
```

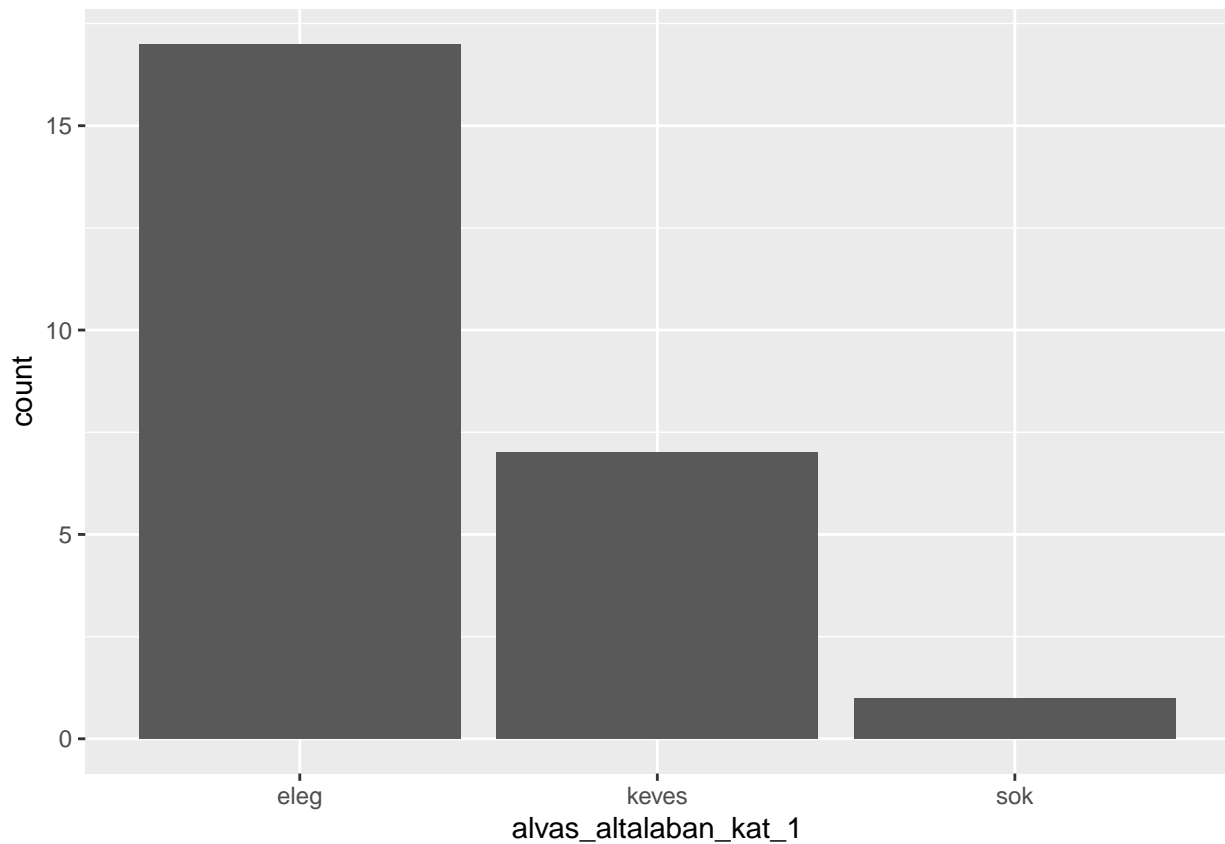
```
## [1] "eleg" "keves" "sok"
```

Faktorszintek sorrendje, ordinalis változók

Amikor van értelme a **sorrendiségnek** a faktorszintek között, **ordinalis változók**ról beszélünk (vagyis az egyik faktorszint alacsonyabb, vagy kisebb “értéku” mint a másik). Arra figyelni kell, hogy amikor faktorokat hozunk létre, az R automatikusan a faktorszintek neveinek **ABC sorrendje** alapján rakja őket

sorba, es az abrakon is így szemlelteti majd oket.

```
orai_adat %>%  
  ggplot() +  
  aes(x = alvas_atalaban_kat_1) +  
  geom_bar()
```

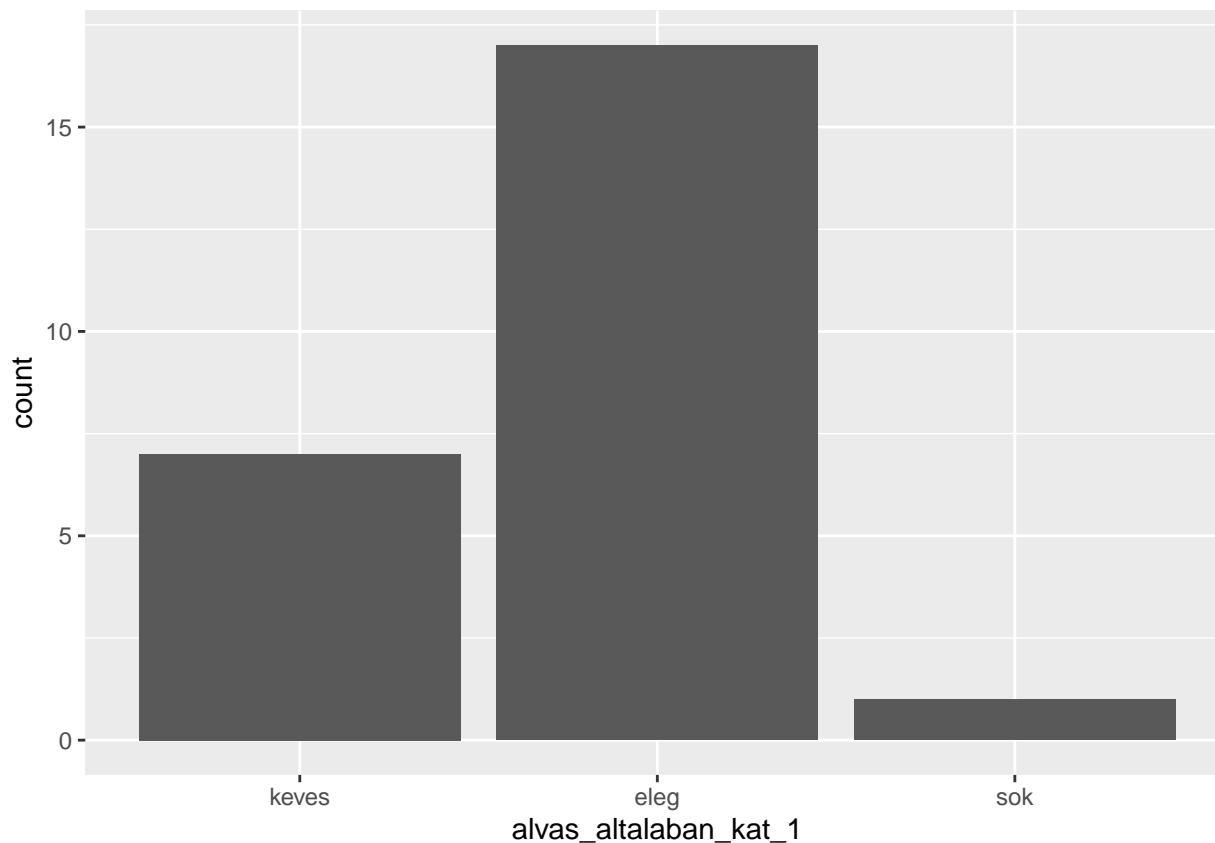


Ilyenkor erdemes meghatározni a faktorszintek sorrendjét (**order**). Ezt legegyszerűbben a `factor()` funkcion belül tehetjük meg, az **ordered = T** beállításával, es a **levels** = reszel a szintek sorrendjének meghatározásával.

```
orai_adat = orai_adat %>%  
  mutate(alvas_atalaban_kat_1 = factor(alvas_atalaban_kat_1, ordered = T, levels = c(  
    "keves",  
    "eleg",  
    "sok"))))
```

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

```
orai_adat %>%  
  ggplot() +  
  aes(x = alvas_atalaban_kat_1) +  
  geom_bar()
```



Gyakorlas

- Csinalj egy kategorikus változót az első órás hangulat alapján (*hangulat_1* változó) úgy, hogy három csoport alakuljon ki: 0-3 - rossz, 4-6 - közepes, 7-10 - jó. (Emlékeztetül: ezt a `mutate()` és a `recode()` funkciókkal tudod például elérni.) Ezt az új változót nevezd el *hangulat_kat_1*-nek, és az ezt az új változót is tartalmazó adattáblát mentsd el *orai_adat_harmashangulat_1* néven.
- Fontos, hogy a *hangulat_kat_1* változót faktor változóként jelöld meg. (Ezt lehet az előző lépésben a `mutate()` funkcion belül, vagy egy külön lépésben, de mindenképpen a `factor()` vagy az `as.factor()` funkciókat érdemes hozzá használni)
- Készíts egy táblázatot arról, hogy hányan esnek a *hangulat_kat_1* egyes kategóriaiba.
- Add meg a faktorszintek helyes sorrendjét: rossz, közepes, jó (Írd felül a *hangulat_kat_1* korábbi változatát ezzel a változattal ahol a szintek már helyes sorrendben vannak)
- Nézd meg a faktor szintjeit, hogy valóban 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 vizualizacian keresztul

Egyes valtozok vizualizacioja

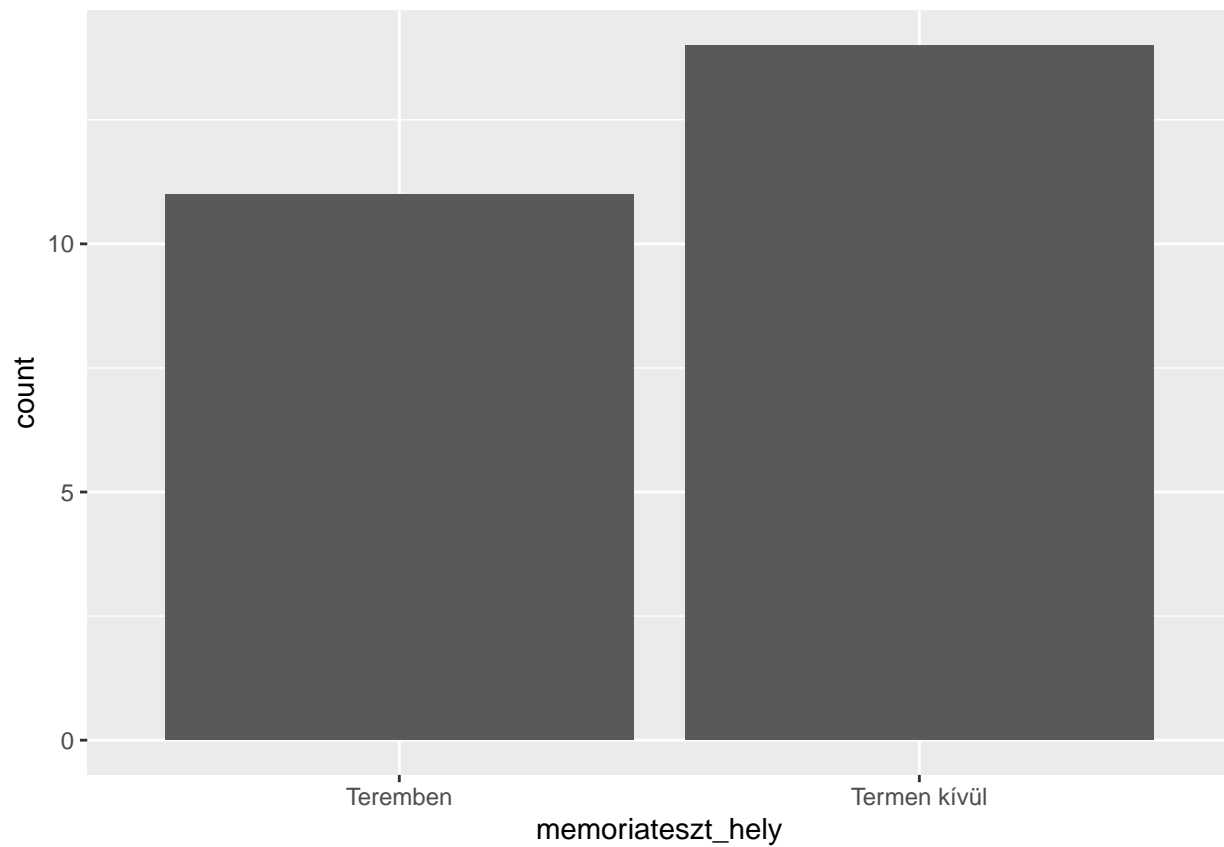
Az egyes valtozok **abrak** (plot) segitsegevel is megvizsgalhatok. A **kategorikus** valtozokat gyakran oszlopdiagrammal (**geom_bar**) abrazoljuk,

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

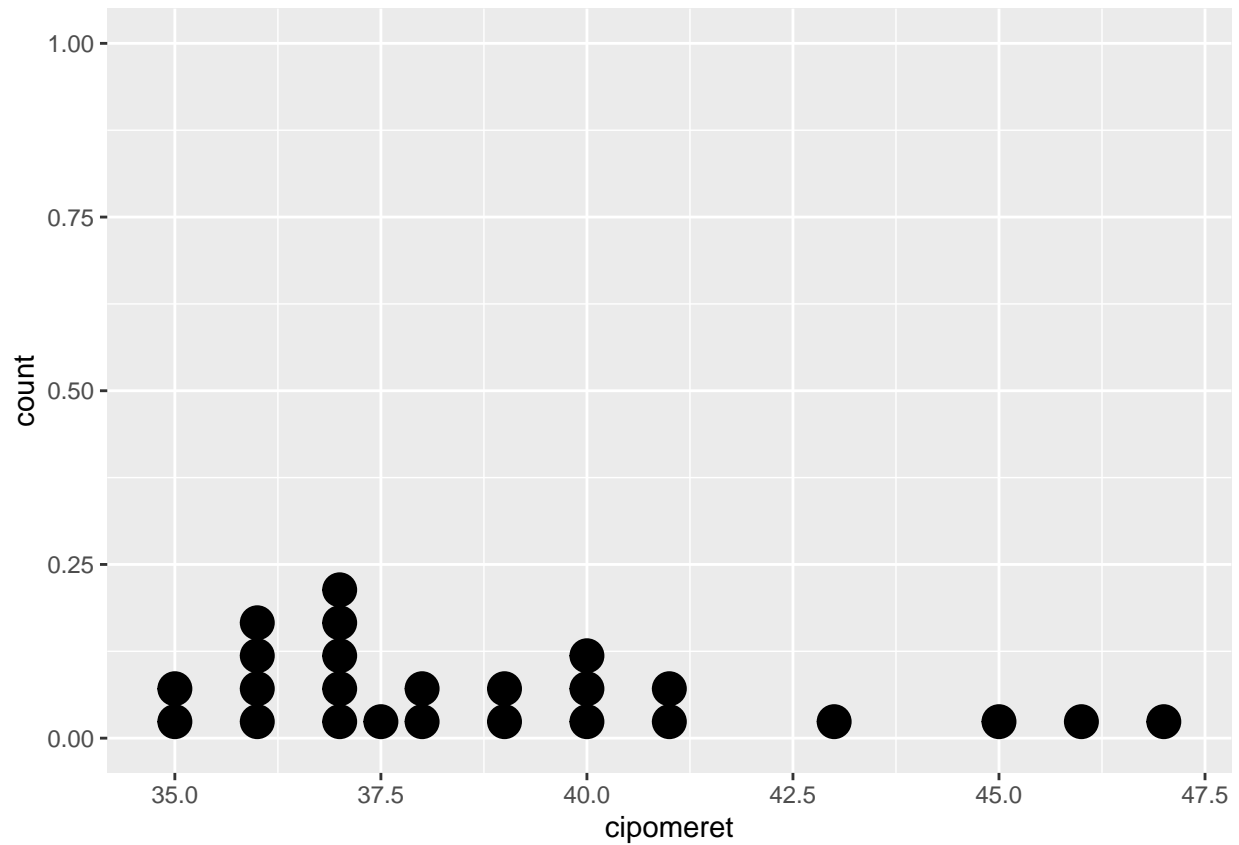
Az egyes valtozok vizualizacioja es a leiro statisztikak atvizsgalasa elengedhetetlen hogy azonositsuk az esetleges adatbeviteli **hibakat es egyeb nemvart furcsasagokat** az adataink kozott.

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

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

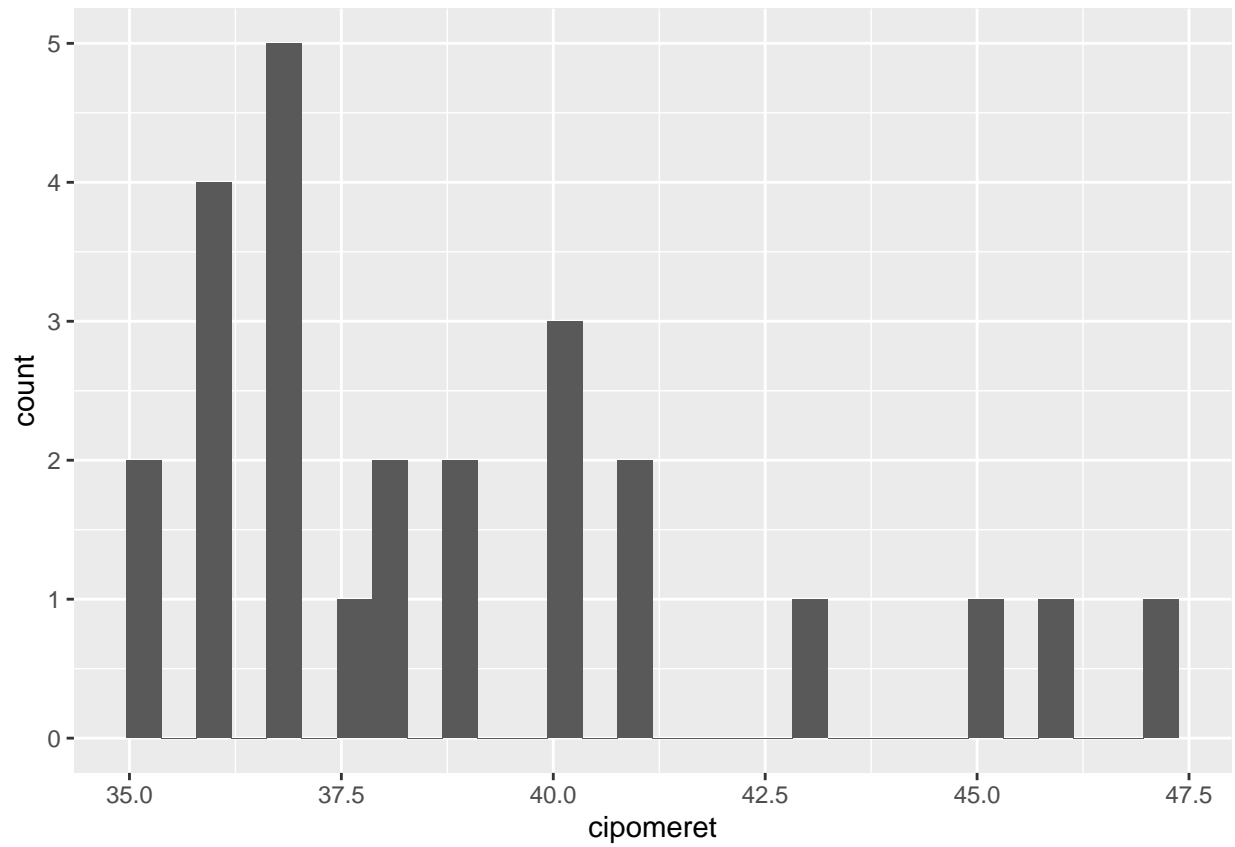


```
orai_adat %>%  
ggplot() +  
  aes(x = cipomeret) +  
  geom_dotplot()  
  
## `stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.
```

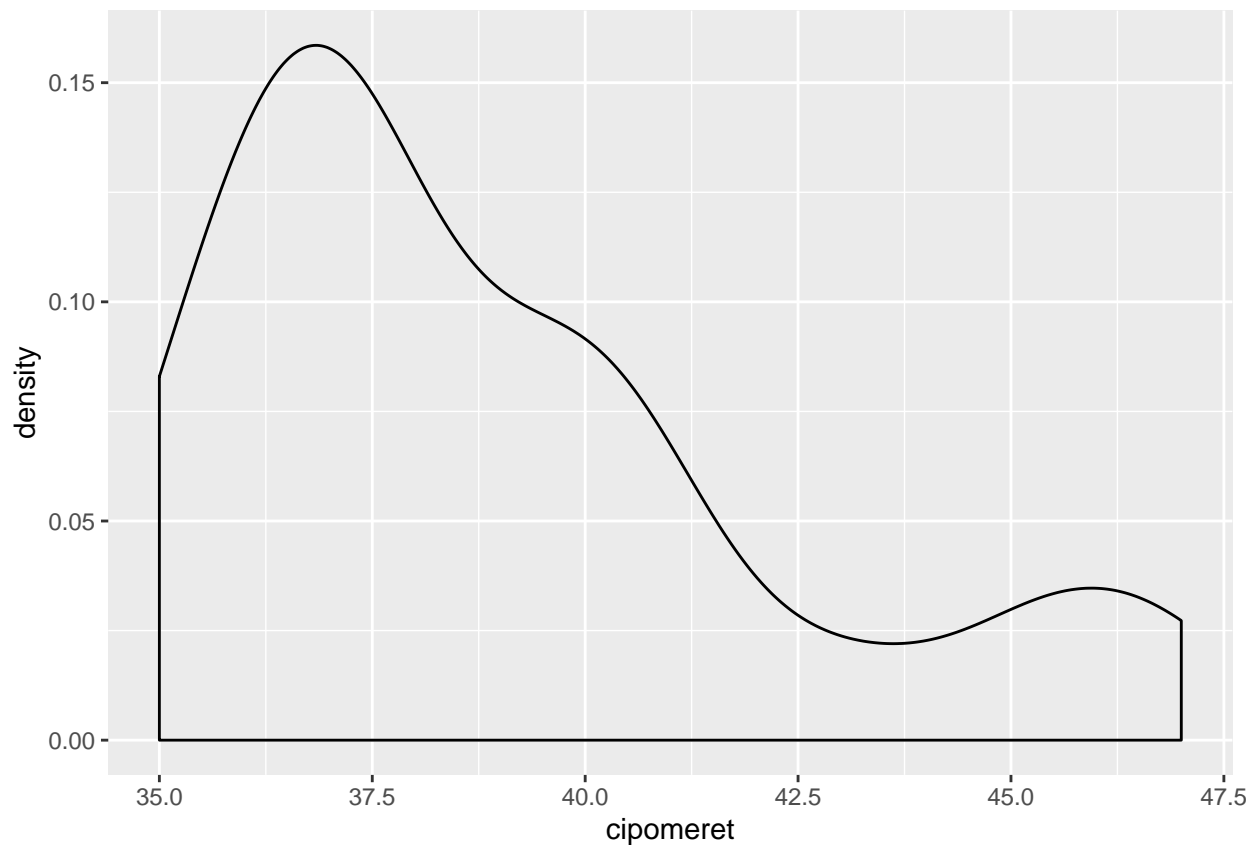


```
orai_adat %>%  
ggplot() +  
  aes(x = cipomeret) +  
  geom_histogram()
```

```
## `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 **azonosítsd az orai_adat adattáblában levo hibakat** vagy nem vart furcsasagokat.

- A vizualizacian tul a View(), describe(), es summary() funkciokat 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			
## életkor*			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			

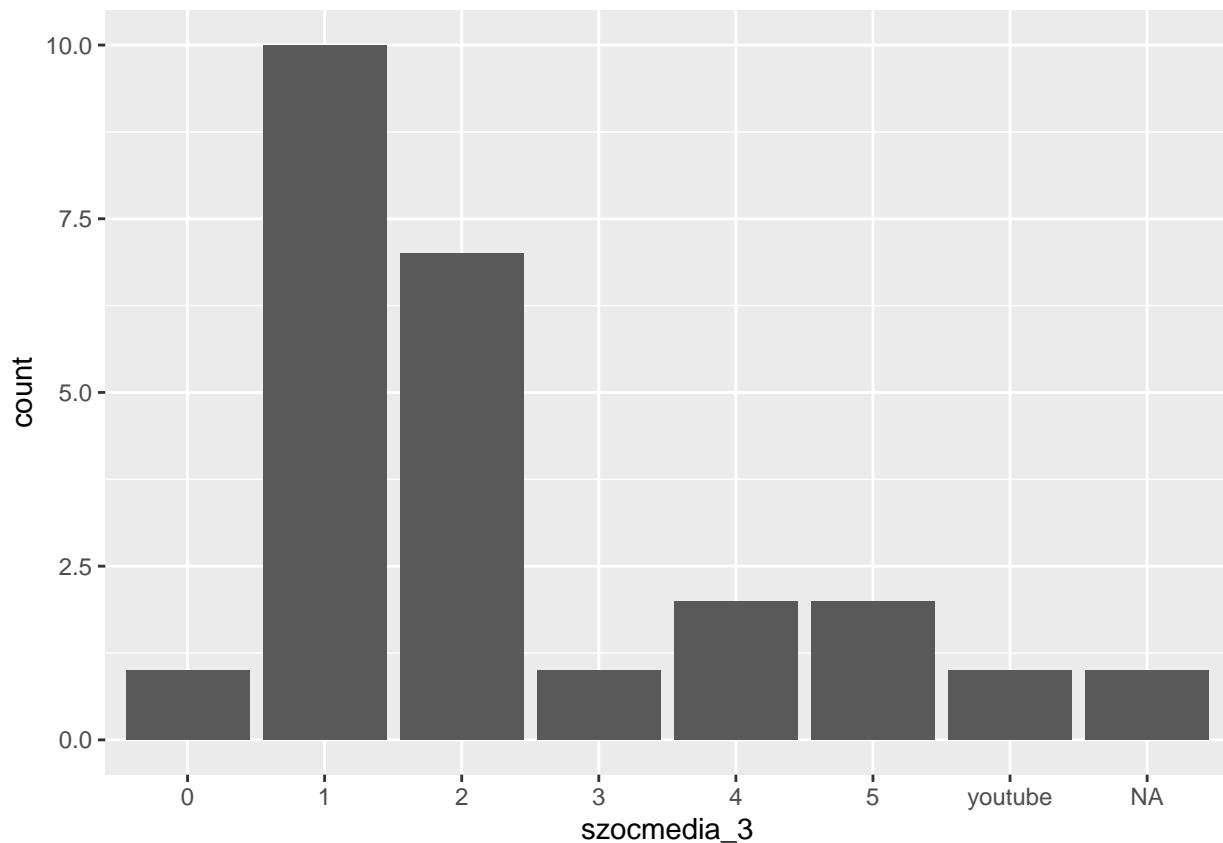
```
## hangulat_3          8.00 -0.74    -0.30 0.46
## stat_tudasszint_3   8.00 -0.18    -0.99 0.42
## stat_gyakorlas_3    14.00 1.26     0.88 0.74
## alvas_altalaban_kat_1* 2.00 -0.26    -0.45 0.10
```

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

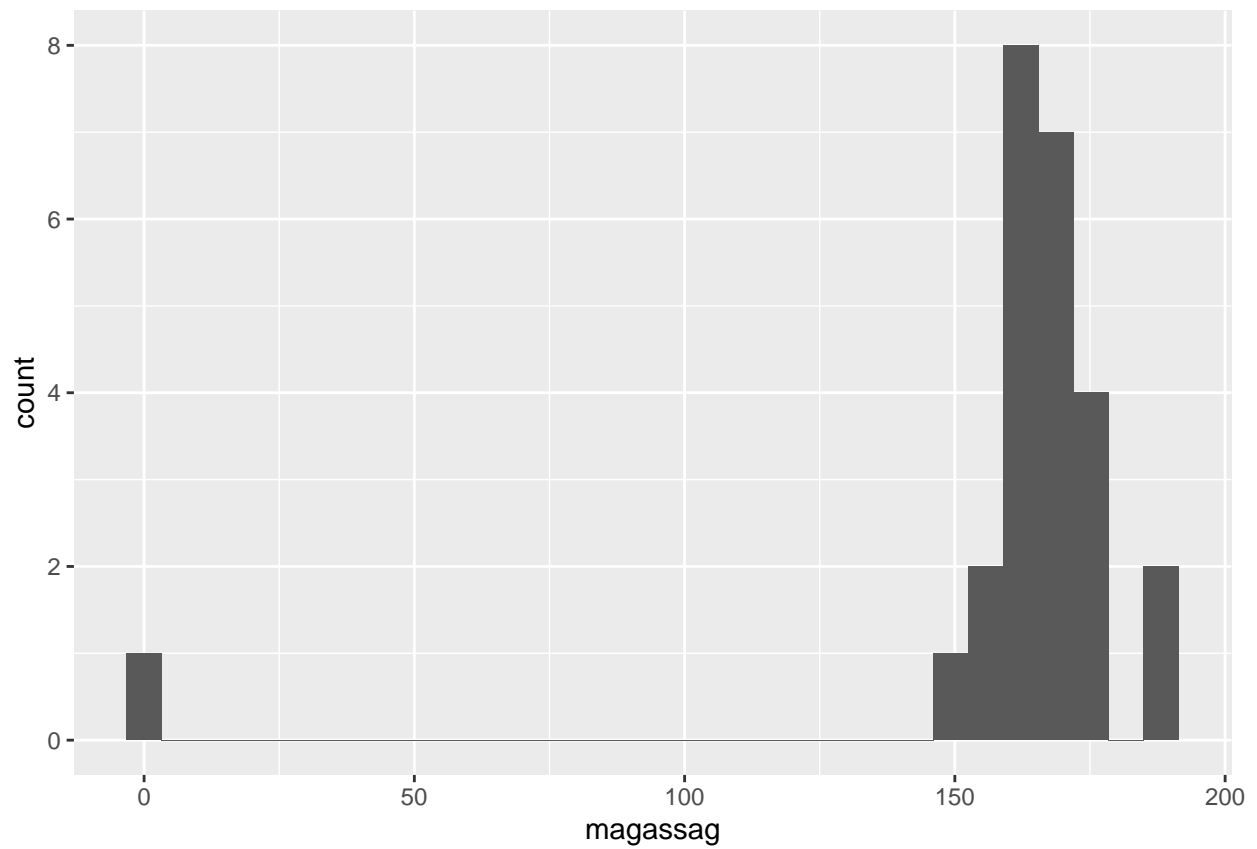
```
## kitoltesiido      ID          nem    etekkor  alvas_szukseges
## Length:25         Length:25      Férfi: 3   0-22:22   Min.   : 6.00
## Class :character   Class :character   No   :22   23+  : 3   1st Qu.: 8.00
## Mode  :character   Mode  :character           Median : 8.00
##                                     Mean   : 8.32
##                                     3rd Qu.: 9.00
##                                     Max.   :10.00
##
## nemalvas_oka      memoria      memoria_ront_A    memoria_ront_B
## Length:25         Min.   : 4.00    Length:25         Length:25
## Class :character   1st Qu.: 7.00    Class :character   Class :character
## Mode  :character   Median : 8.00    Mode  :character   Mode  :character
##                                     Mean   : 7.48
##                                     3rd Qu.: 8.00
##                                     Max.   :10.00
##
## memoria_ront_C     memoria_ront_D     memoria_ront_E     memoria_ront_F
## Length:25         Length:25         Length:25         Length:25
## Class :character   Class :character   Class :character   Class :character
## Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
## jegy_tipp         valasztott_szam tippelt_gyakori_szam    magassag
## Min.   :3.00      Min.   : 2.00    Min.   : 2.0      Min.   : 1.82
## 1st Qu.:4.00      1st Qu.: 5.00    1st Qu.: 3.0      1st Qu.:162.00
## Median :4.00      Median : 7.00    Median : 6.0      Median :168.00
## Mean   :4.04      Mean   : 6.56    Mean   : 5.8      Mean   :161.07
## 3rd Qu.:5.00      3rd Qu.: 8.00    3rd Qu.: 7.0      3rd Qu.:170.00
## Max.   :5.00      Max.   :10.00    Max.   :10.0      Max.   :190.00
##
## cipomeret         memoriateszt_hely alvas_tegnap_1    alvas_tegnap_3
## Min.   :35.00      Teremben :11      Min.   : 6.00      Min.   :5.000
## 1st Qu.:37.00      Termen kívül:14    1st Qu.: 7.00      1st Qu.:7.000
## Median :38.00                        Median : 8.00      Median :8.000
## Mean   :38.94                        Mean   : 7.92      Mean   :7.609
## 3rd Qu.:40.00                        3rd Qu.: 9.00      3rd Qu.:8.000
## Max.   :47.00                        Max.   :10.00      Max.   :9.000
##                                     NA's   :2
## alvas_altalaban_1 alvas_altalaban_3 energiaszint_1 energiaszint_3
## Min.   :5.00      Min.   :6.00      Min.   :1.00      Min.   :3.000
## 1st Qu.:6.00      1st Qu.:7.00      1st Qu.:4.00      1st Qu.:4.500
## Median :7.00      Median :7.00      Median :5.00      Median :7.000
## Mean   :7.04      Mean   :7.13      Mean   :5.48      Mean   :6.174
## 3rd Qu.:8.00      3rd Qu.:7.50      3rd Qu.:8.00      3rd Qu.:8.000
## Max.   :9.00      Max.   :8.00      Max.   :8.00      Max.   :9.000
```

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

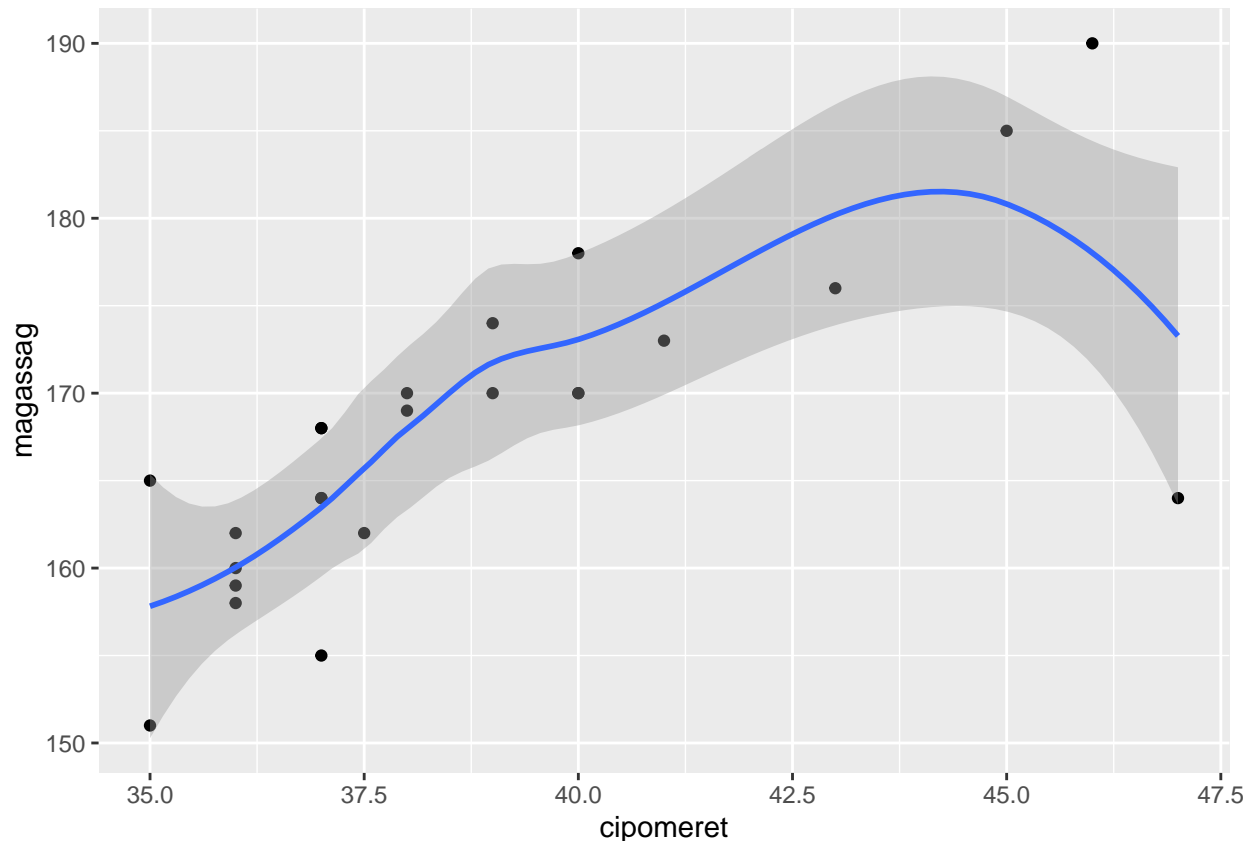
```
orai_adat %>%
  ggplot()+
  aes(x = szocmedia_3)+
  geom_bar()
```



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

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



A hibákat a következőképpen javíthatjuk.

A `mutate()` és a `replace()` funkciók használatával **cserélhetünk ki** értékeket más értékekre. Azt, hogy ilyenkor hiányzó adatra (NA), vagy egy másik, valószínű értékre kell megváltoztatni az értéket, a szituációtól függ. Általában a biztosabb megoldás ha hiányzó adatnak jelöljük a kérdéses értéket (NA), de ez sok adatvesztéshez vezethet. Ha elég valószínű hogy mi a helyes válasz, beírhatjuk, DE **minden javítást fel kell tüntetni** a kutatási jelentésben (és a ZH során is), hogy az olvasó számára tiszta legyen, hogy itt egy adathelyettesítés vagy kizárás történt!

Mindig érdemes a javított adatokat **új adattablába** elmenteni. A mi esetünkben az `orai_adat_corrected` nevet adtuk a javított objektumnak. Így a nyers adataink megmaradnak, ami hasznos lehet későbbi műveleteknel.

```
orai_adat_corrected <- orai_adat %>%
  mutate(szoimedia_3 = replace(szoimedia_3, szoimedia_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. Alább az adatok vizualizációjával győződünk meg erről, de az adatok megjelenítésével, vagy a leíró statisztikák lekérdezésével is megtehető ez, ha az informatív.

```
# használhatnak meg az alábbiakat is arra,
# hogy megbizonyosodjunk abban, hogy sikeres volt a csere
```

```

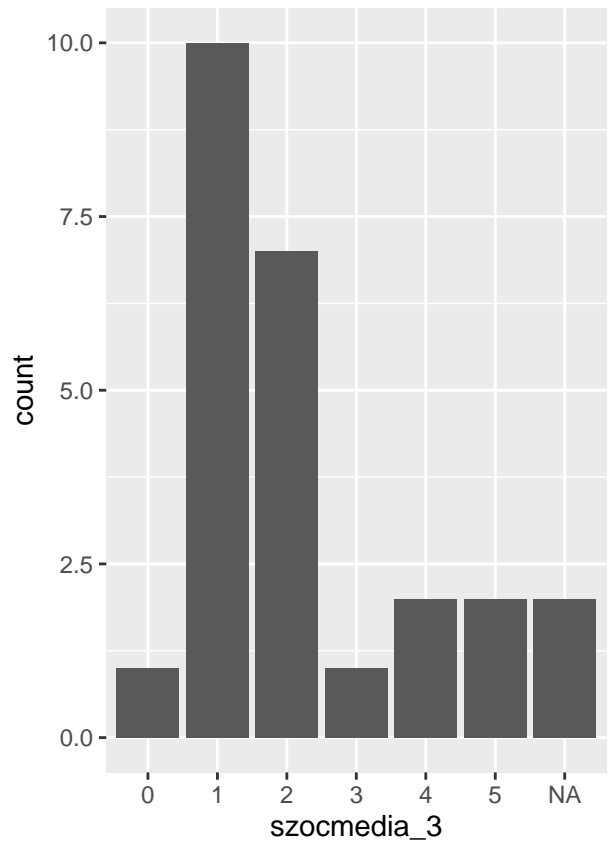
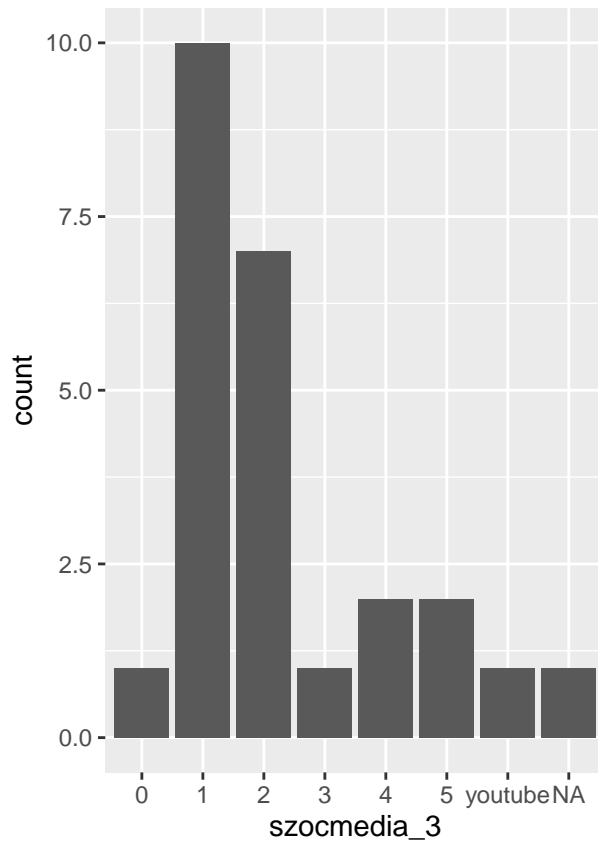
# View(orai_adat_corrected)
# describe(orai_adat_corrected)
# summary(orai_adat_corrected$szocmedia_3)
# orai_adat_corrected$szocmedia_3

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 <-

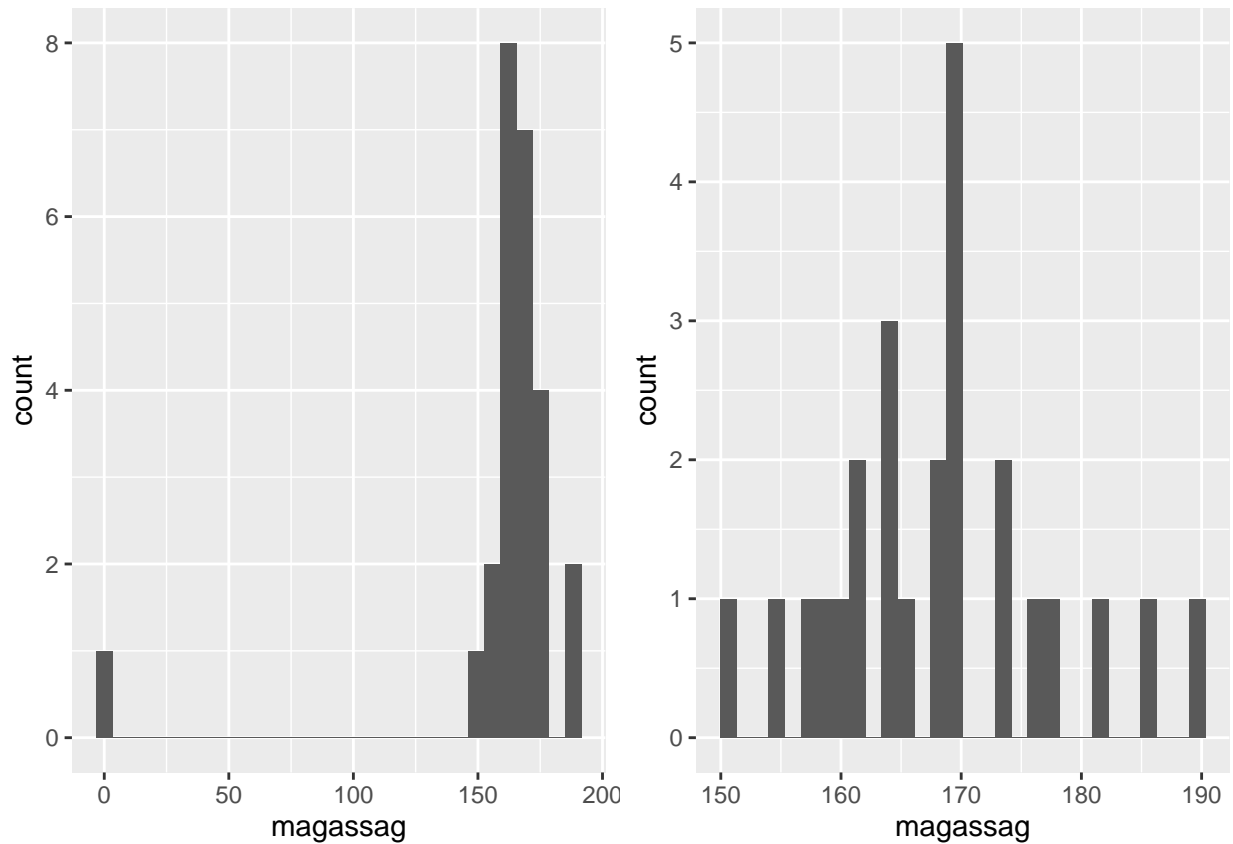
```

```

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

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

```



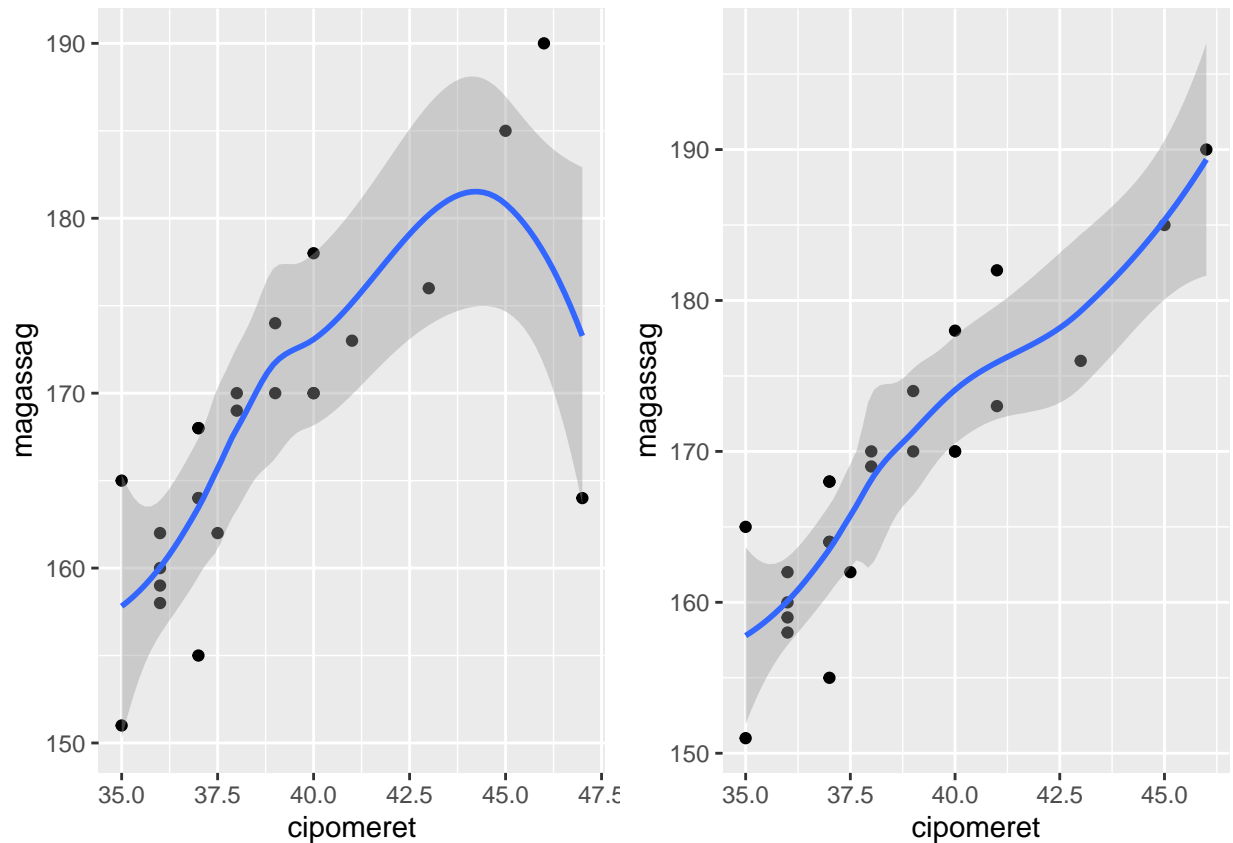
```

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 változó kapcsolatának felterkepezese

Több változó kapcsolatát is felterkepezhetjük táblázatok és ábrák segítségével.

Két kategorikus (csoportosított) változó kapcsolatának felterkepezese

Generaljunk kategorikus változókat

Az első órán jelentett **hangulat_1** változóból csinálunk egy kategorikus változót, az faktorszintek a következők lesznek: 0-3 - “rossz”, 4-6 - “közepes”, 7-10 - “jó”. Ezt ugyan abba az adattáblába **hangulat_kat_1** néven mentjük el sorrendezett faktorként a fent tanultak alapján. Mindehhez a `mutate()`, `recode()`, és `factor()` funkciókat használjuk.

```
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

```

Hasonlóképpen képzünk **szocmedia__1** változóból egy kategorikus változót, ahol azok akik 0 vagy 1 órát használnak a szociális médiát, a “nem rendszeres”, azok akik 2 vagy több órát használnak a szociális médiát, a “nem rendszeres” kategóriába (faktorszintre) esnek majd.

Itt nem szükséges az **ordered = T** és a **levels =** beállítása, mert csak két faktorszint van, és az ABC sorrend már alapból az intuitív módon a “nem rendszeres” kategóriát teszi elsőnek.

```
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 első órán kaptunk egy tippet arról, hogy az egyes emberek milyen jegyet fognak kapni a kurzus végén az adatelemzés órán (**jegy_tipp** változó). Mivel itt számban kaptuk a választ, ez jelenleg egy numerikus változó, de csináljunk most belőle egy **kategorikus változót**, mert összesen három fajta tipp érkezett: 3 - “közepes”, 4 - “jó”, 5 - “kiváló”

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

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

```
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 össze azzal, hogy mennyit hasznaljak a szocialis mediát (**szocmedia_kat_1**).

A legegyszerubb modja ket csoportosito valtozo kapcsolatának megvizsgalasara a ket valtozo **kereszt-tablazatanak (crosstab)** elkezsitese a **table()** funkcioval.

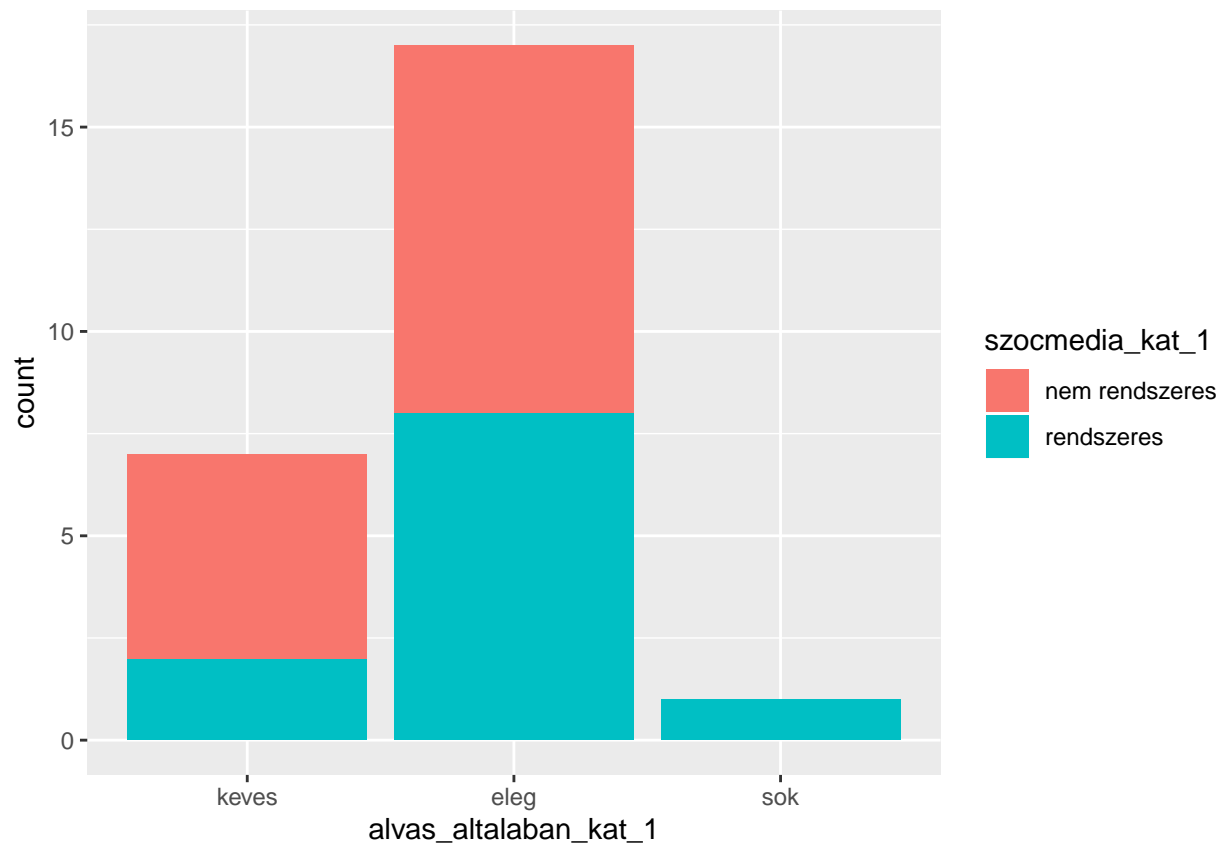
```
table(orai_adat_corrected$alvas_altalaban_kat_1, orai_adat_corrected$szocmedia_kat_1)
```

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

Sokszor ennél sokkal **szemleletesebb az abrak** (plot) használata.

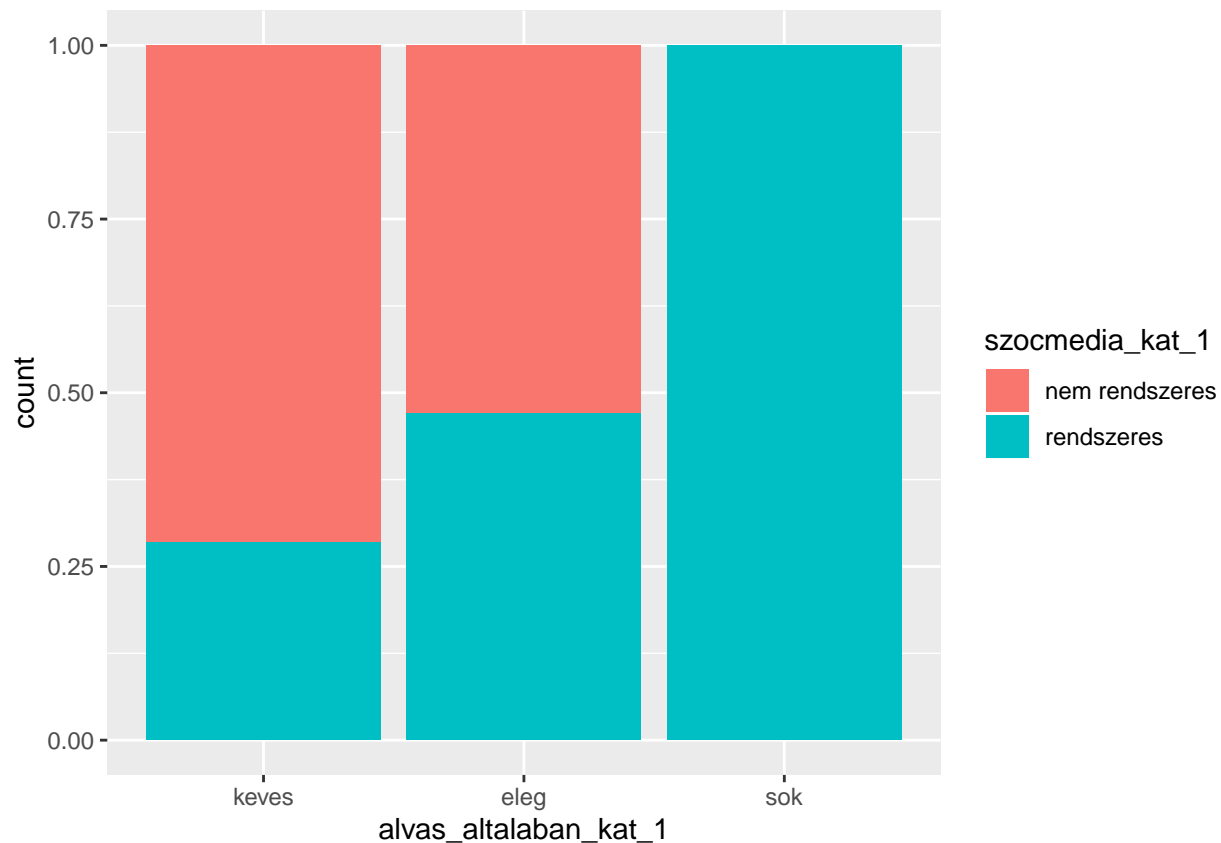
Erre az egyik lehetoseg a **stacked bar chart** (egymásra tornyozott oszlopdiagram, a **geom_bar()** geomot használjuk) használata. Itt az egyik valtozo kategoriai adjak meg hany oszlop lesz (ez a valtozo lesz az x tengelyen reprezentalva, így ezt az “x =” reszen adhatjuk meg), a másik valtozo az oszlopokat szinekké szegmentalja, ezt pedig a “fill =” reszen adhatjuk meg.

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



Ha az egyes faktorszinteken nagyon **különbozo mennyisegu megfigyeles** van, ez a megjelenites neha felrevezeto kovetkeztetesekehez vezethet, így neha hasznosabb ha az oszlopok nem számosságot (count), hanem **reszaranyt (proportion)** jelolnek. Ha ezt szeretnenk, ahelyett hogy uresen hagynank a `geom_bar()` funkciot, a kovetkezozt adjuk meg: `geom_bar(position = "fill")`.

```
orai_adat_corrected %>%
  ggplot() +
    aes(x = alvas_atalaban_kat_1, fill = szocmedia_kat_1) +
    geom_bar(position = "fill")
```

Gyakorlas

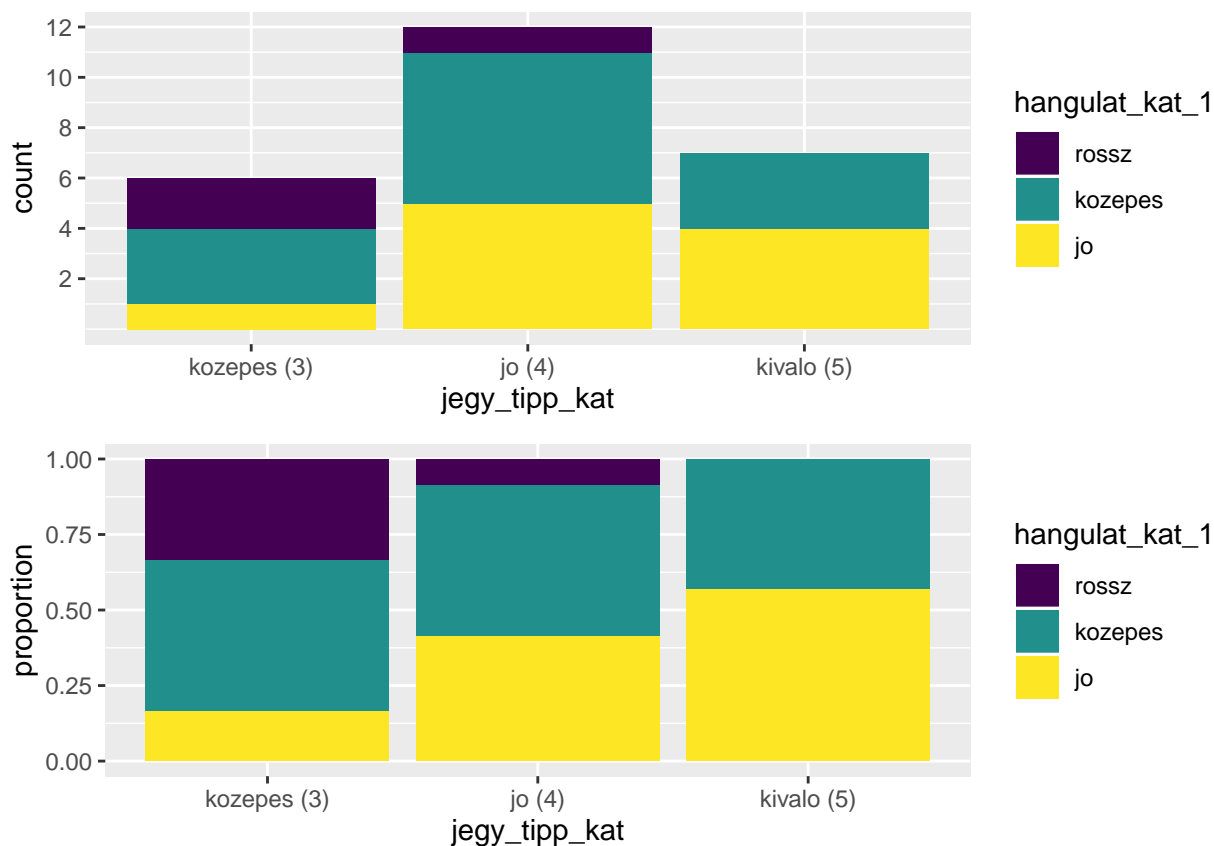
Hasznald a fent tanult modszereket, hogy megvizsgald a **jegy_tipp_kat** es a **hangulat_kat_1** változók közötti összefüggést. - hasznalj **geom_bar()** geomot a megjelenítéshez - próbald meg mind a **szamossagot**, mind a **reszaranyt** kifejező ábrát megvizsgálni **geom_bar(position = "fill")** - milyen **kovetkeztetést** tudsz levonni az ábrákról?

```
# 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)
```

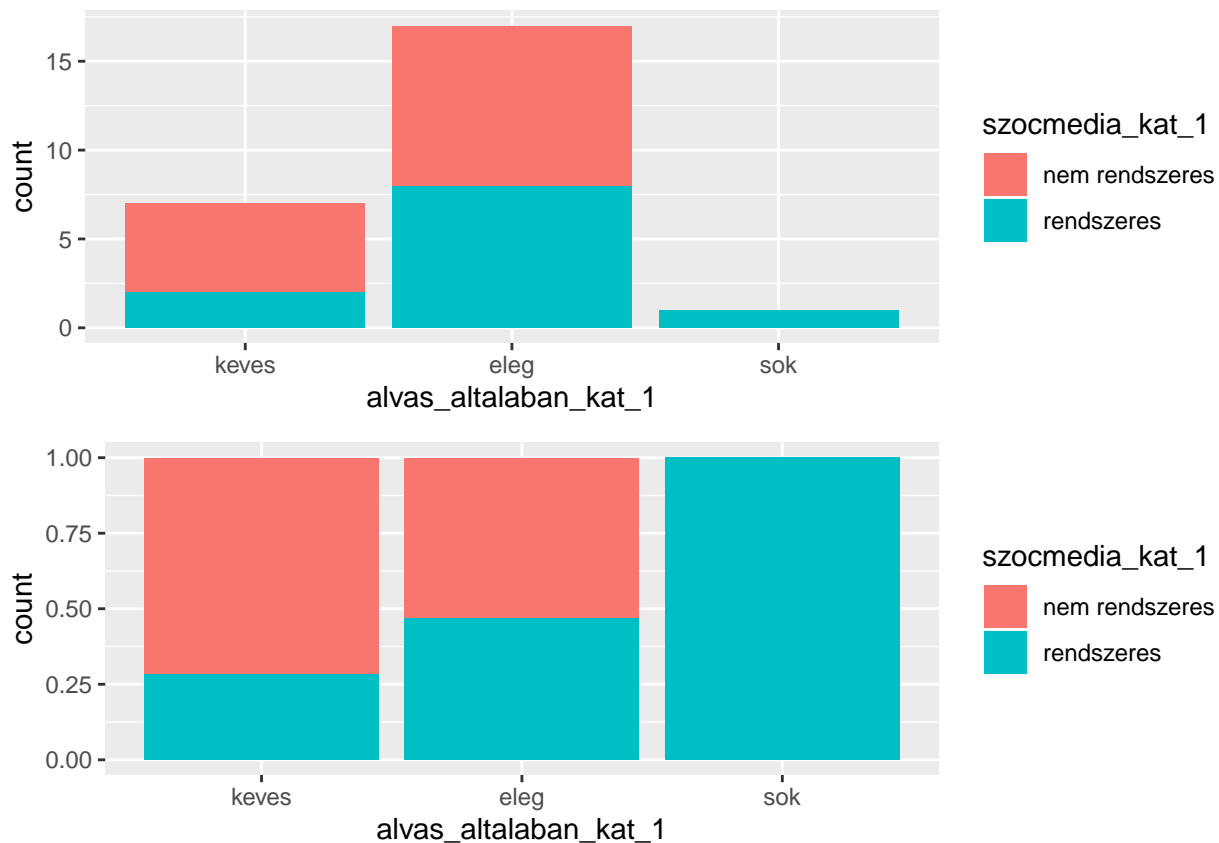


Ennel a megjelenítésnél fontos hogy ha az egyes megfigyelesek **keves megfigyelesbol allnak**, az abra megteveszto lehet, mert az abra nem jelzi a megfigyelesek szamat es így azt, hogy milyen biztosak lehetunk az eredményben. Ilyen esetekben az egyik kategoriat ki lehet venni az abrarol, vagy a **szamossagot es a reszaranyt abrazolo abrakat egymás mellet** lehet bemutatni, hogy így 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)
```

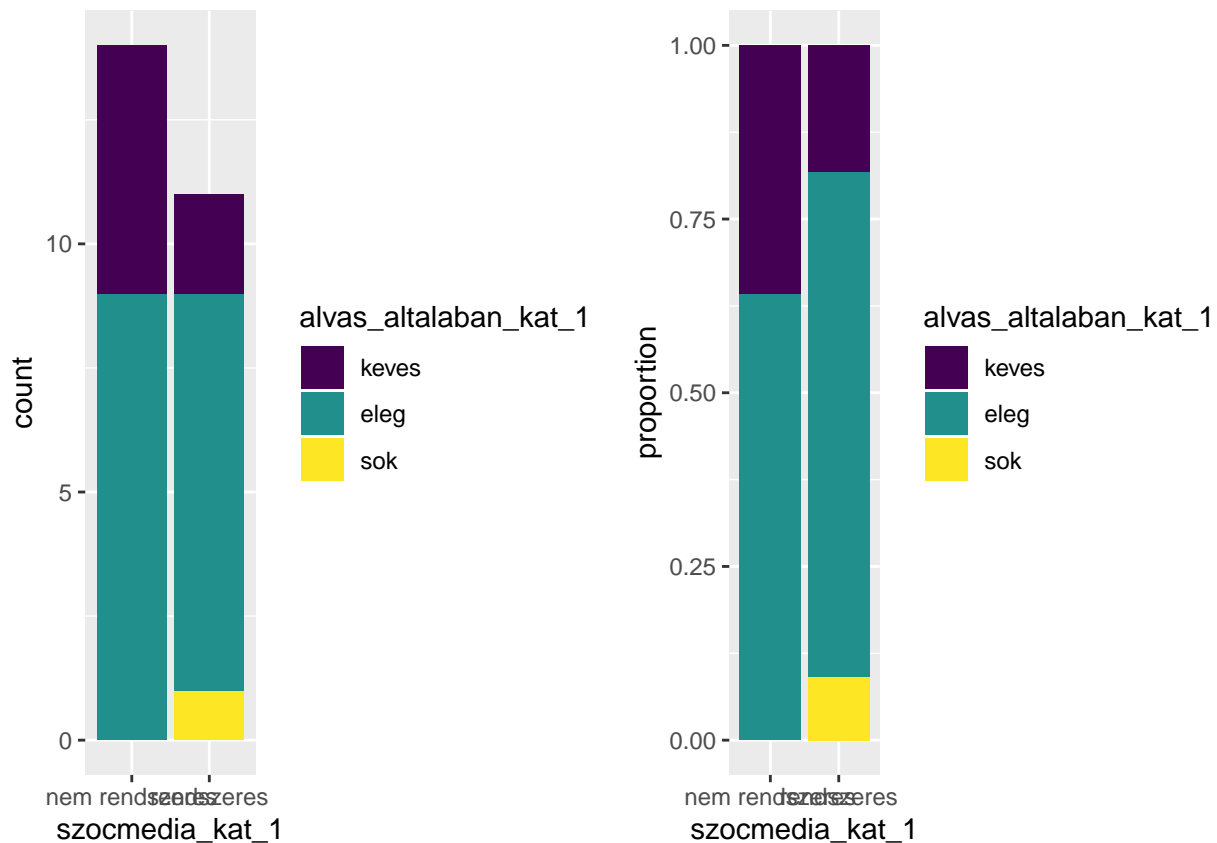


Az ábra **interpretálhatósága** attól függően is **változhat**, hogy melyik változót tesszük az x-tengelyre és melyiket szintként ábrázolva.

```
alvas_atalaban_szocmedia_plot_3 <-
orai_adat_corrected %>%
ggplot() +
  aes(x = szocmedia_kat_1, fill = alvas_atalaban_kat_1) +
  geom_bar()

alvas_atalaban_szocmedia_plot_4 <-
orai_adat_corrected %>%
ggplot() +
  aes(x = szocmedia_kat_1, fill = alvas_atalaban_kat_1) +
  geom_bar(position = "fill") +
  ylab("proportion")

grid.arrange(alvas_atalaban_szocmedia_plot_3, alvas_atalaban_szocmedia_plot_4, ncol=2)
```

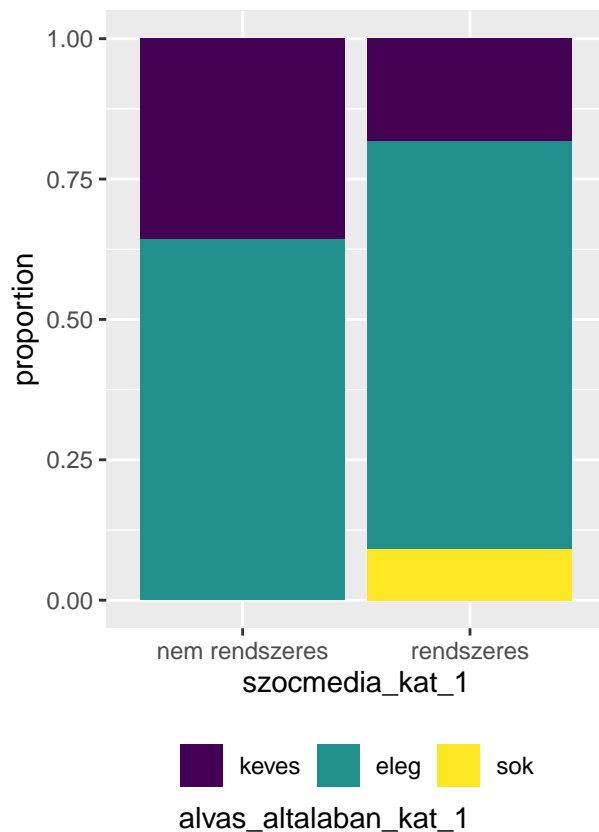
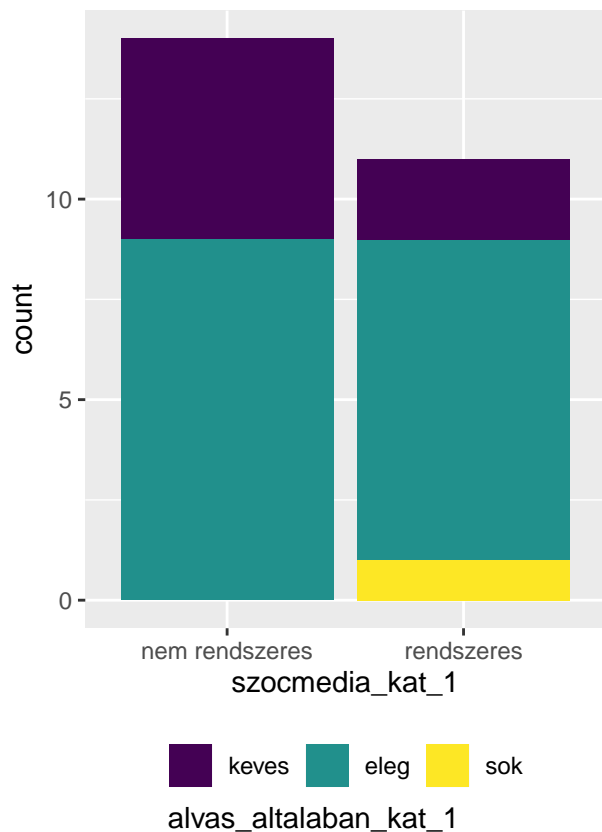


```
# a theme(legend.position) es a guides() funckiok
# hasznalataval kontrollalhatjuk hogy hol es hogyan
# jelenjen meg a jelmagyarazat az abran

alvas_atalaban_szocmedia_plot_3 <-
orai_adat_corrected %>%
  ggplot() +
  aes(x = szocmedia_kat_1, fill = alvas_atalaban_kat_1) +
  geom_bar() +
  theme(legend.position="bottom") +
  guides(fill = guide_legend(title.position = "bottom"))

alvas_atalaban_szocmedia_plot_4 <-
orai_adat_corrected %>%
  ggplot() +
  aes(x = szocmedia_kat_1, fill = alvas_atalaban_kat_1) +
  geom_bar(position = "fill") +
  theme(legend.position="bottom") +
  guides(fill = guide_legend(title.position = "bottom")) +
  ylab("proportion")

grid.arrange(alvas_atalaban_szocmedia_plot_3, alvas_atalaban_szocmedia_plot_4, ncol=2)
```

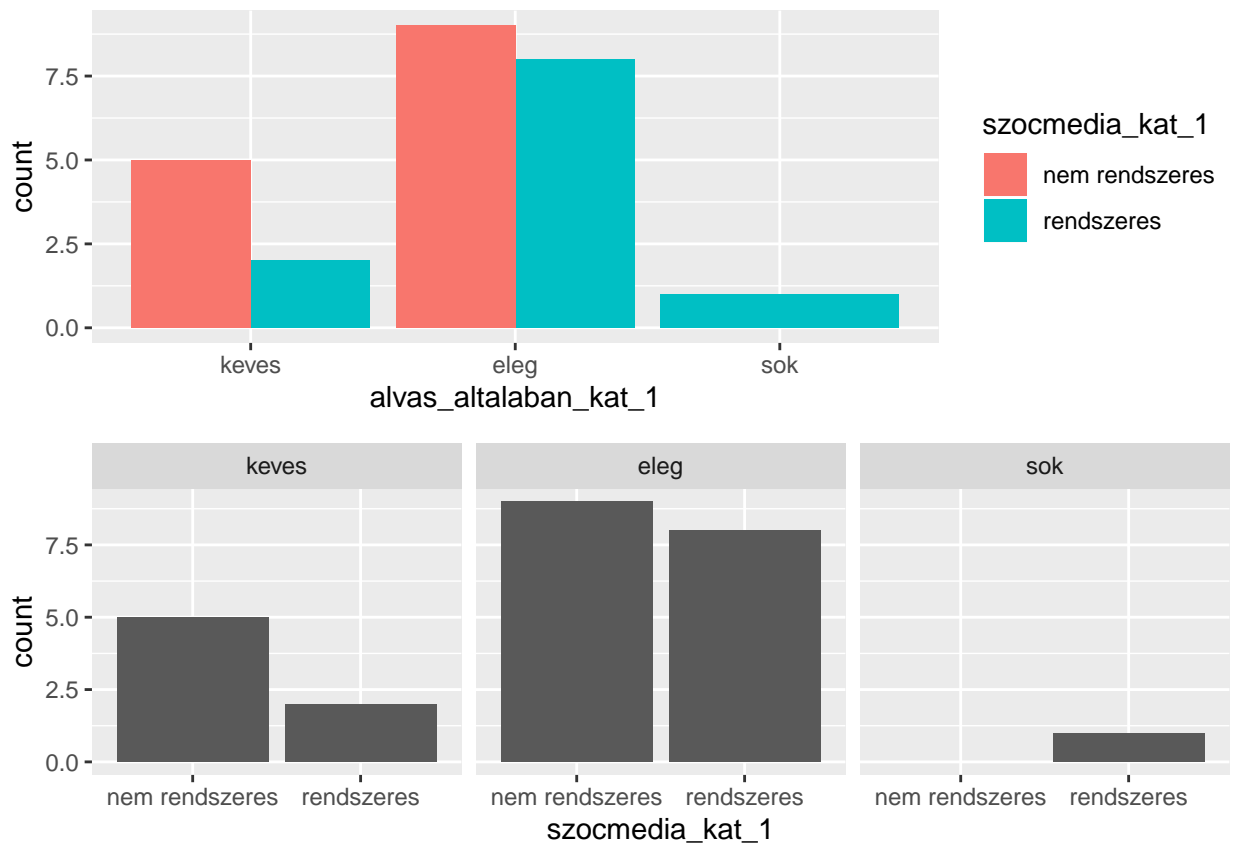


Ujabb modja a barchart segitsegevel valo megjelenitesnek ha az oszlopok nem egymásra tornyozva, hanem **egymas mellett** jelennek meg, vagy ha a masodik valtozo szerint **kulon paneleken (facet)** jelennek meg.

```
alvas_atalaban_szocmedia_plot_1 <-
orai_adat_corrected %>%
ggplot() +
  aes(x = alvas_atalaban_kat_1, fill = szocmedia_kat_1) +
  geom_bar(position = "dodge")

alvas_atalaban_szocmedia_plot_2 <-
orai_adat_corrected %>%
ggplot() +
  aes(x = szocmedia_kat_1) +
  geom_bar() +
  facet_wrap(~ alvas_atalaban_kat_1)

grid.arrange(alvas_atalaban_szocmedia_plot_1, alvas_atalaban_szocmedia_plot_2, nrow=2)
```



Egy kategorikus es egy numerikus valtozo kapcsolata

Vizsgáljuk meg a **magassag** összefüggését azzal, hogy ez emberek mit tippeltek, milyen jegyet fognak kapni az első órán (**jegy_tipp**).

Ezt megtehetjük az átlagok csoportonkénti áttekintésével a **group_by()** és a **summarize(mean())** segítségével.

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

```
## # A tibble: 3 x 2
##   jegy_tipp_kat mean
##   <ord>         <dbl>
## 1 közepes (3)   168
## 2 jó (4)       166.
## 3 kiváló (5)   172.
```

A két változó kapcsolata megvizsgálható **abrákkal** is. Pl. használhatjuk a

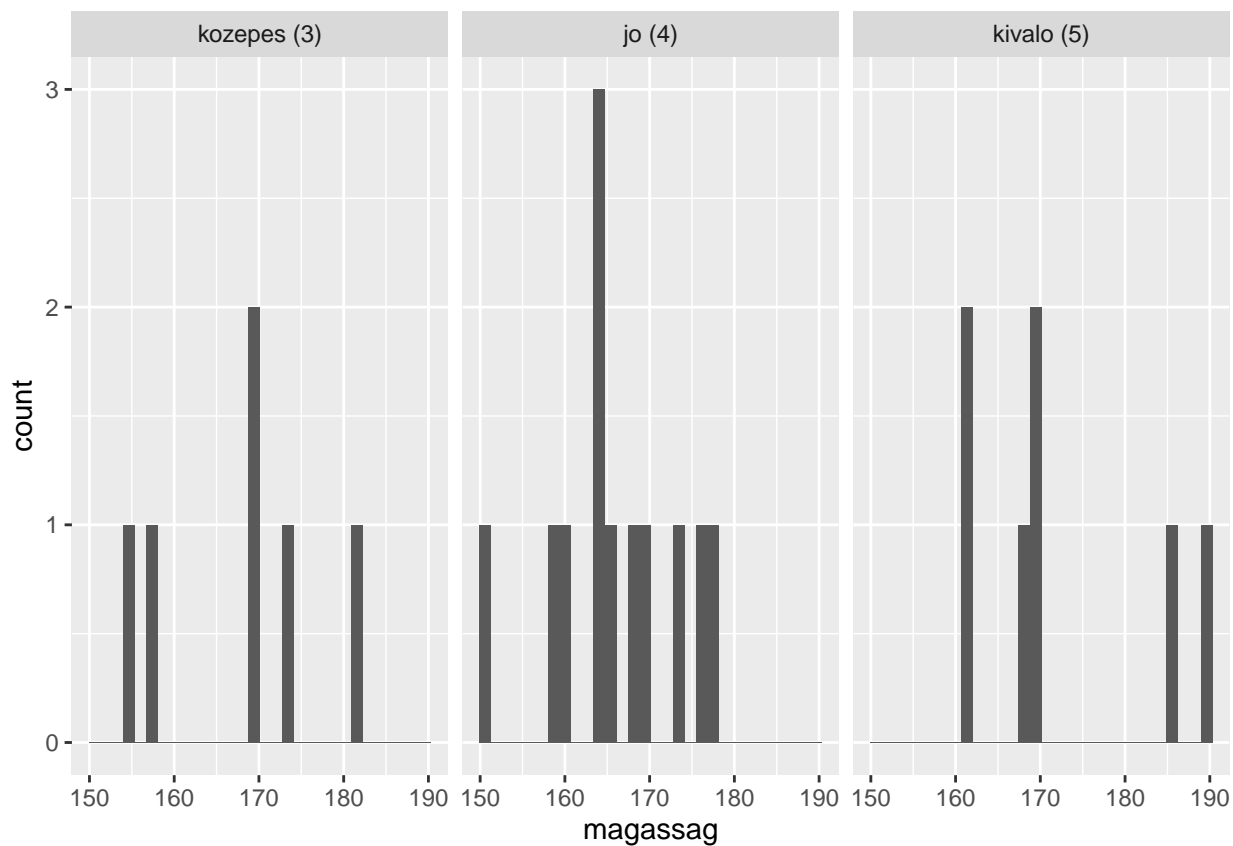
- **facet_wrap()** függvényt a egy **geom_histogram()** vagy **geom_dotplot()** -al kombinálva
- a **geom_boxplot()** -ot
- esetleg használhatunk egy egymásra illesztett **geom_density()** plot-ot.
- talán ebben az esetben a leglátványosabb képet a **geom_violin()** mutatja, ami a **geom_boxplot()** és a **geom_density()** keverékének tekinthető. Ezt kiegészíthetünk egy **geom_point()** -al, hogy pontosan látszon, hany megfigyelesen alapulnak az ábra adatai.

Mindig érdemes **több megközelítést** is használni az adat-exploráció közben, hogy minél részletesebb képet

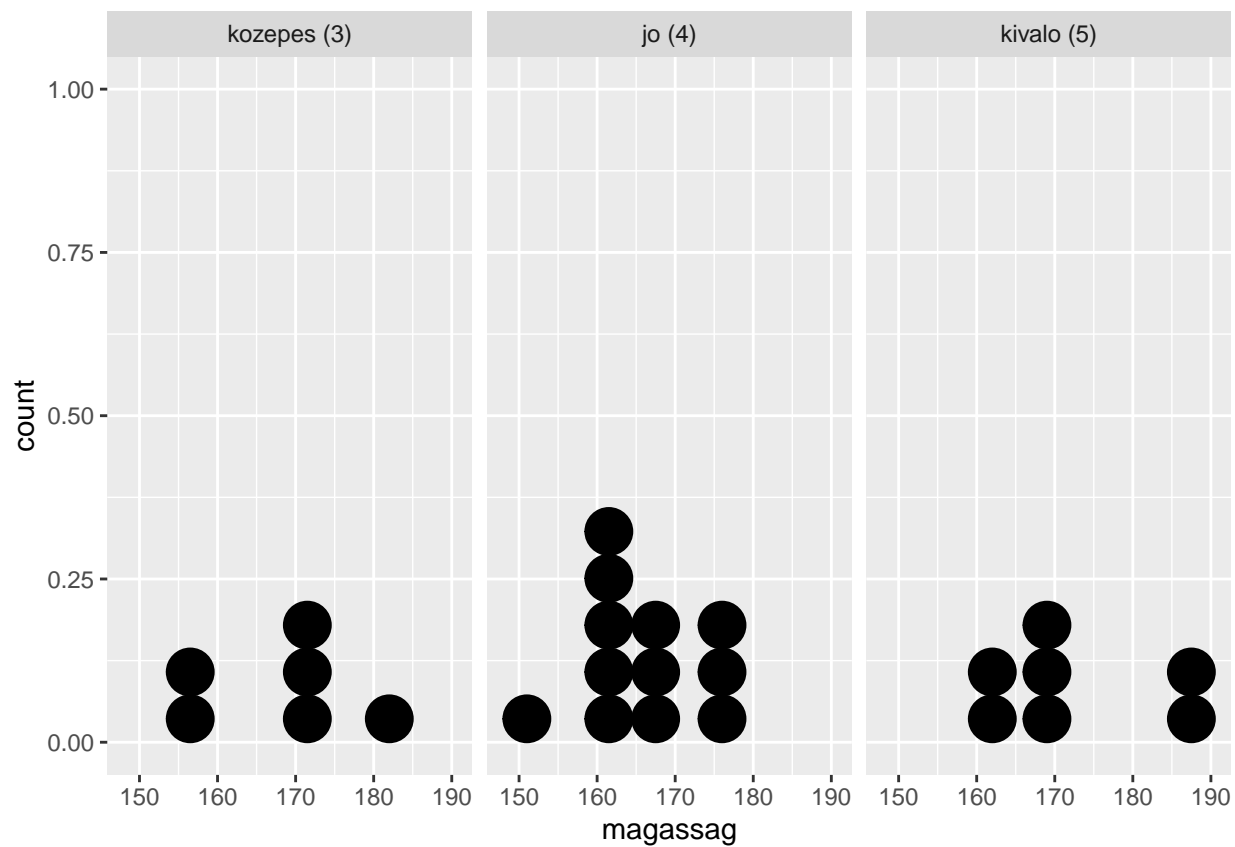
kaphassunk, es csokkentsuk a valoszinuseget hogy egyik vagy masik megkozelites hanyossagai felrevezetnek minket.

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

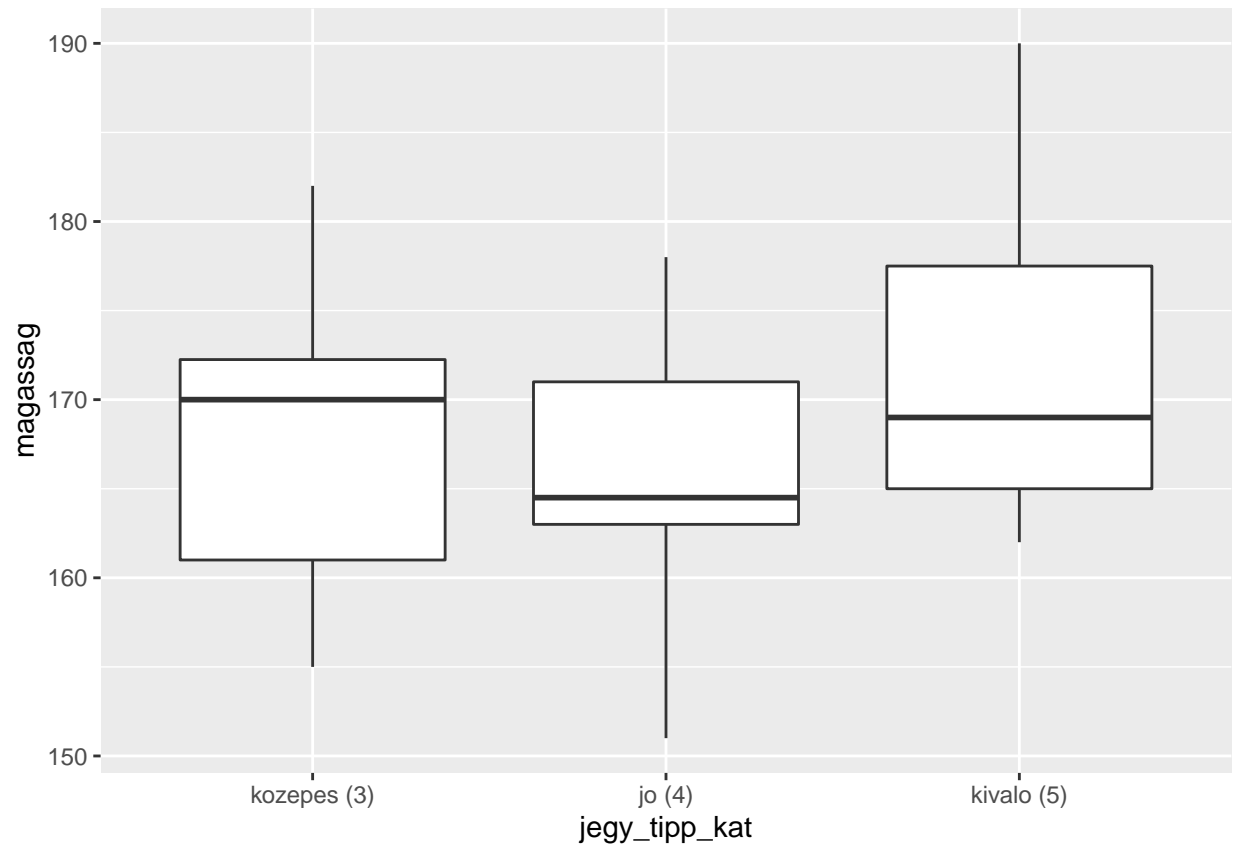
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



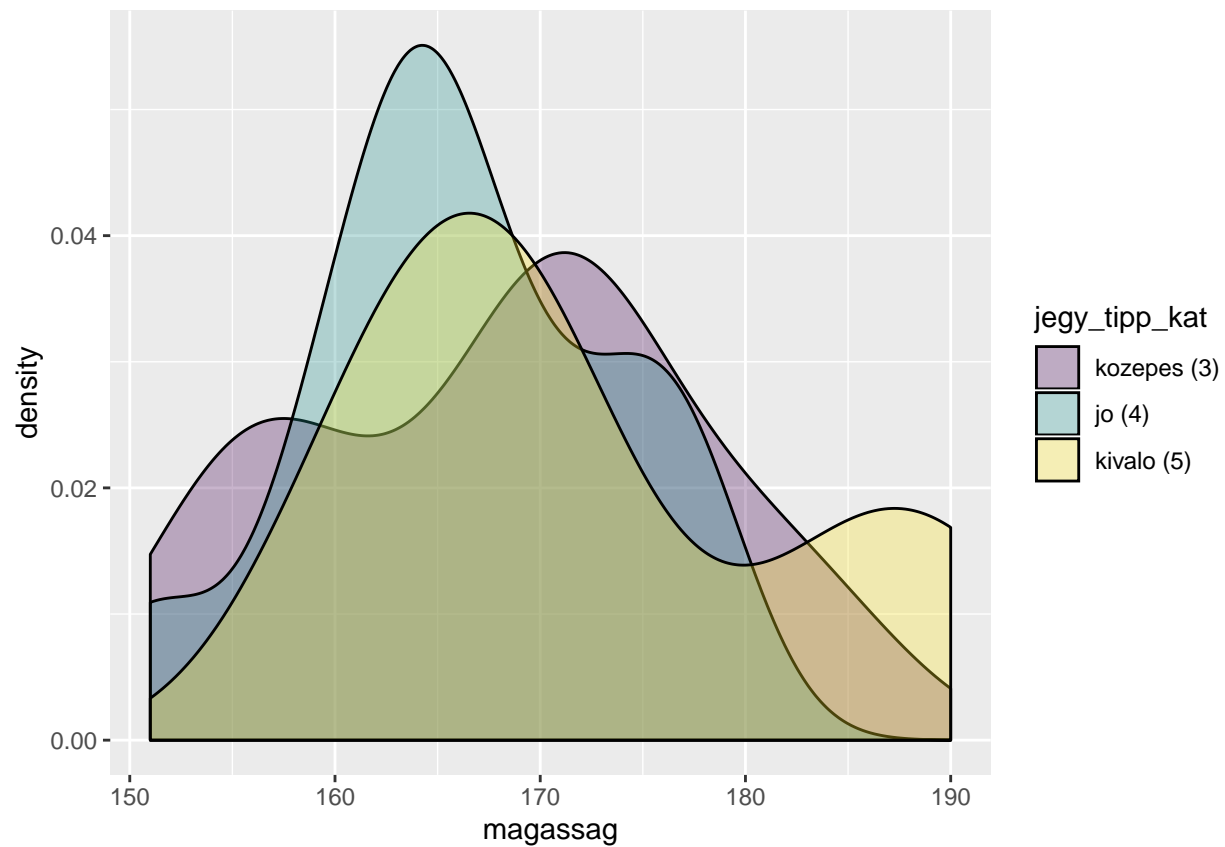
```
orai_adat_corrected %>%  
  ggplot() +  
    aes(x = magassag) +  
    geom_dotplot(binwidth = 6) +  
    facet_wrap(~ jegy_tipp_kat)
```



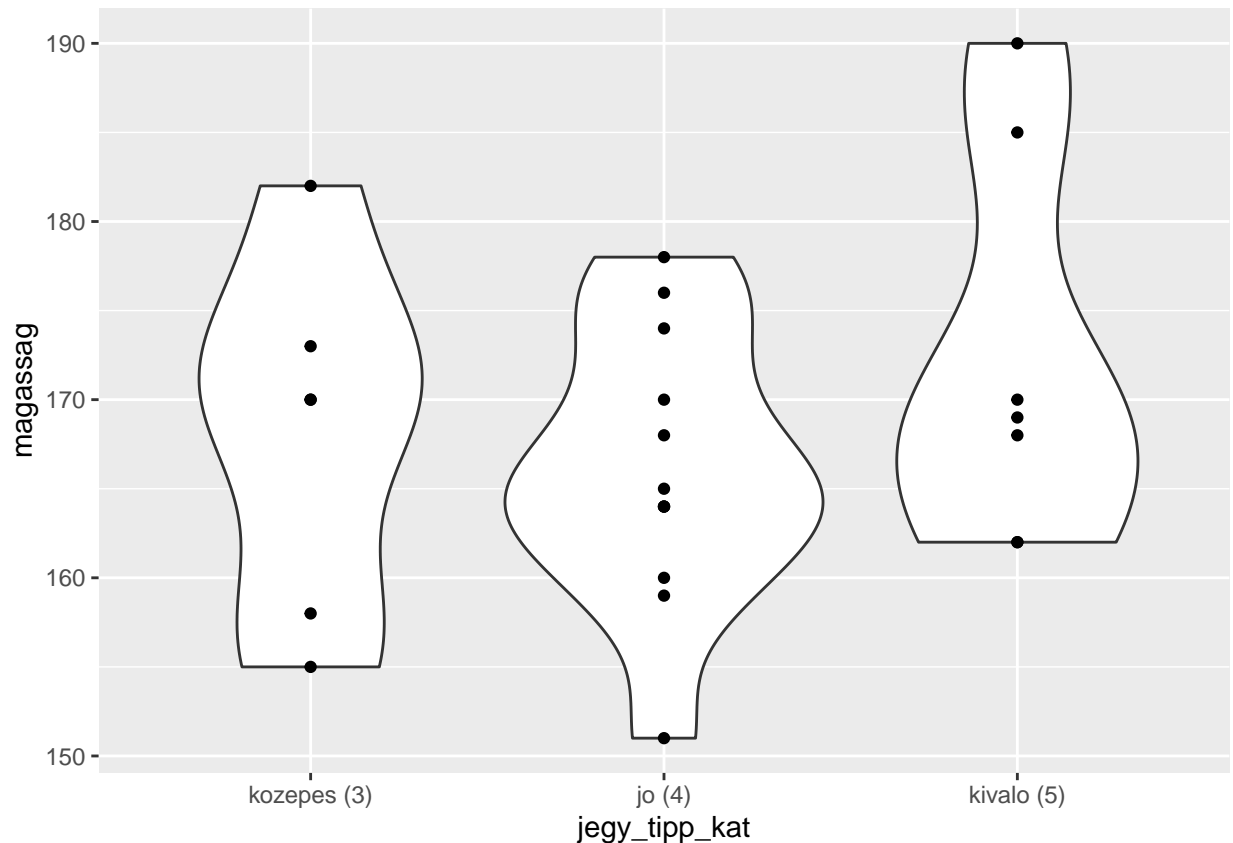
```
orai_adat_corrected %>%
  ggplot() +
    aes(x = jegy_tipp_kat, y = magassag) +
    geom_boxplot()
```

```
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()
```



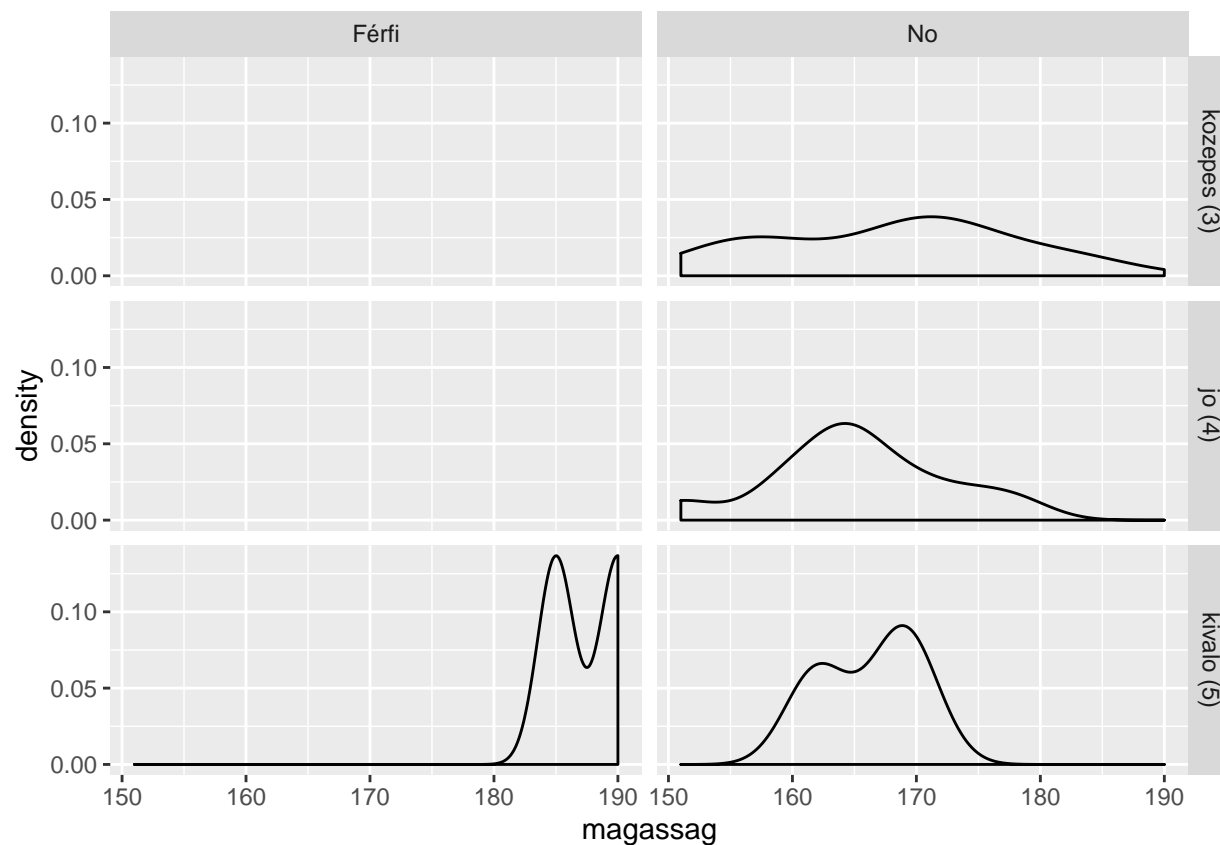
A fenti ábrán látszik, hogy bár az átlagok a fenti táblázatban különböztek, a maguknak kiváló pontot jósolók többsége 170 cm körüli, de van két **kiurgo magassagu személy**, aki az átlagot felhúzza ebben a csoportban.

Az is elképzelhető, hogy egy **nemi hatast** látunk az eredményekben, hiszen a férfiak magasabbak, és elképzelhető, hogy nagyobb az adatelemzéssel kapcsolatos önbizalmuk is. Próbáljuk meg a **ferfiakra és a nőkre külön elkészíteni az ábrát**.

Itt már **három változó** kapcsolatot ábrázoljuk, amihez a `facet_grid()` funkciót lehet használni, vagy különböző esztétikai elemeket (`aes()`) lehet a különböző változókhoz rendelni.

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

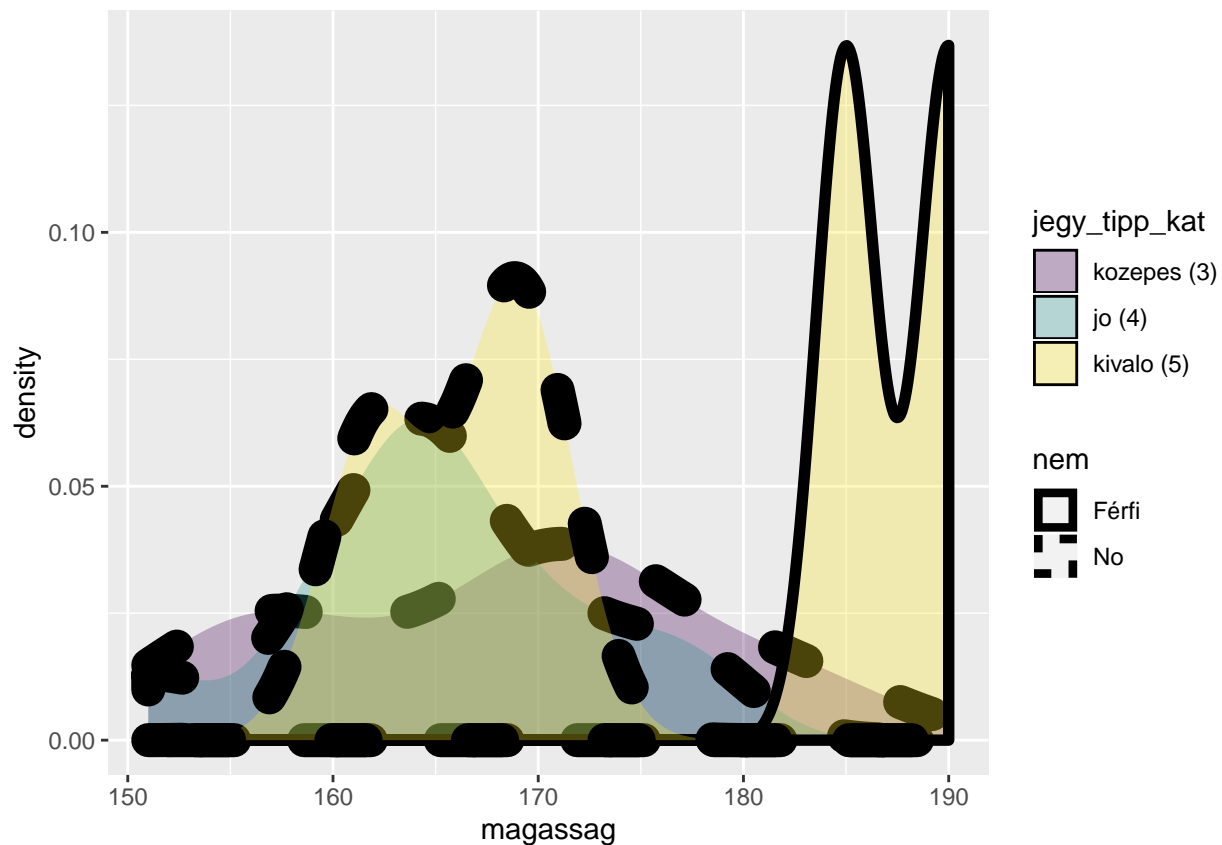
```
## Warning: Groups with fewer than two data points have been dropped.
```



```
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.
```

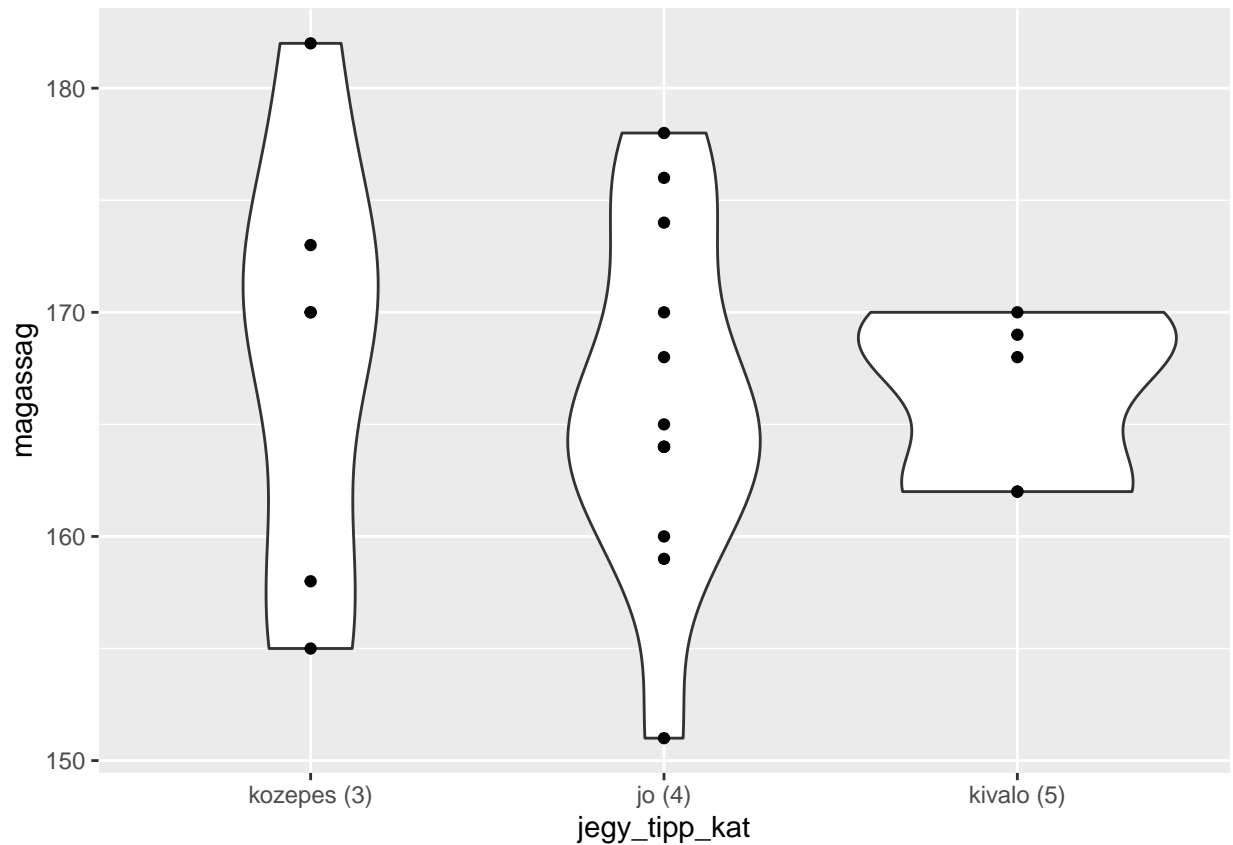
```
## Warning: Groups with fewer than two data points have been dropped.
```



Ha szeretnénk **kizarni az elemzesunkbol** ezeket az extrem ertekekt, a **filter()** funkcio beekesevel a pipe-ba megepithetjuk a fenti abrakat 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.
```

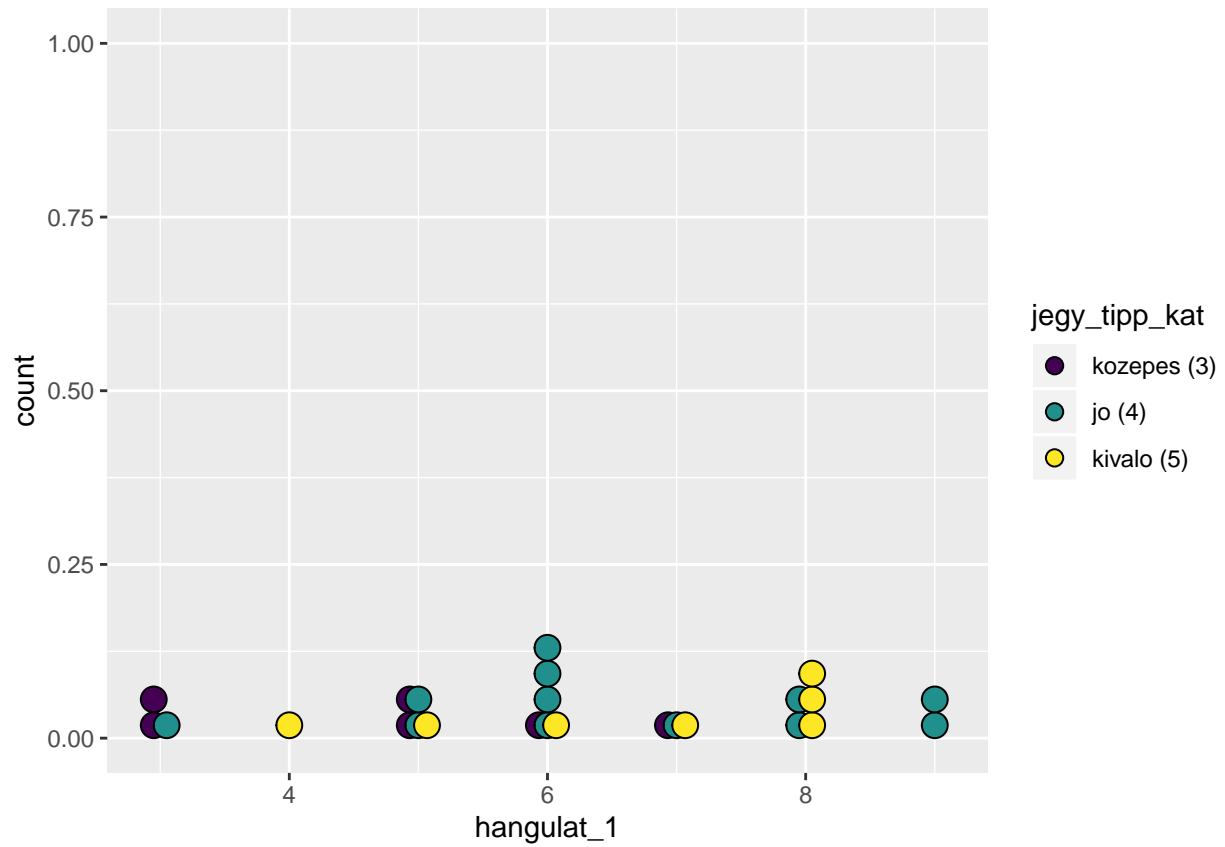
Gyakorlas

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

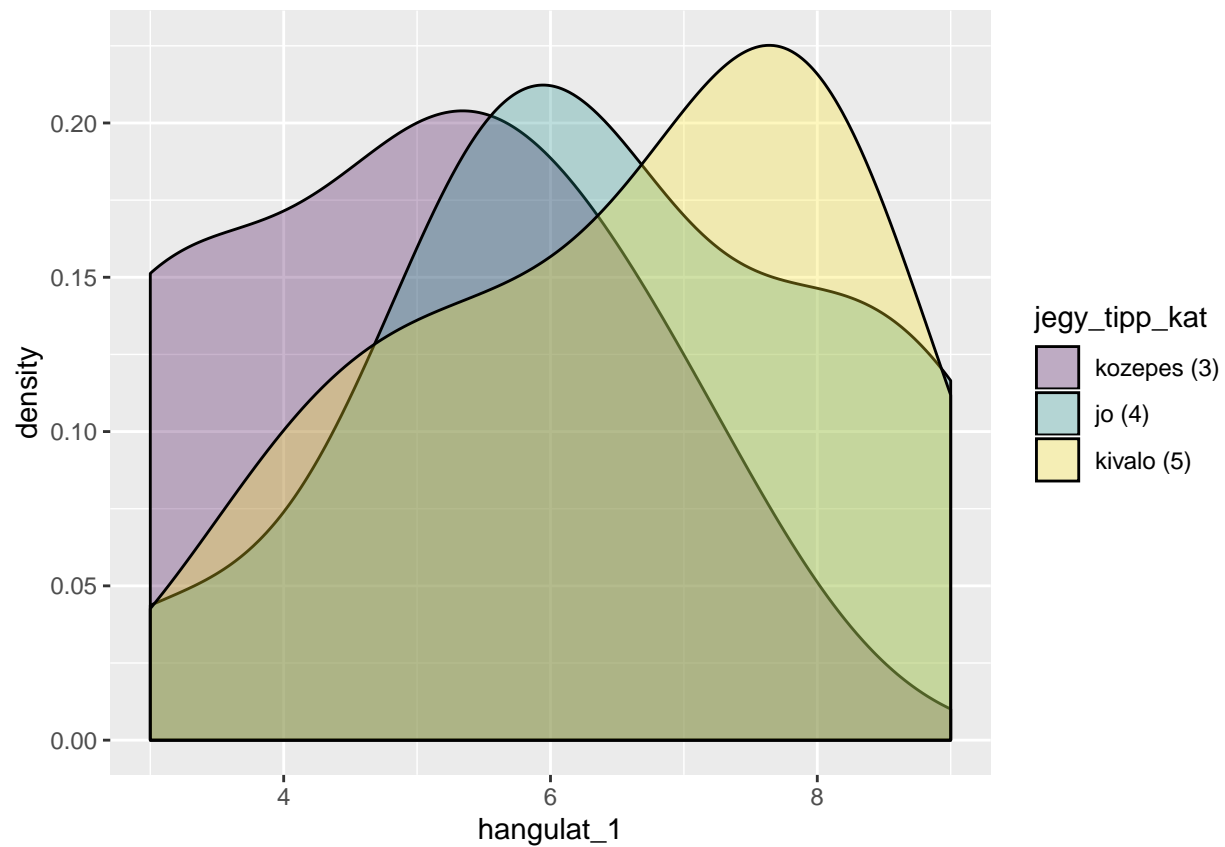
- hasznald a fenti geomokat, es keszits legalabb ket kulonbozo abrat mas-mas geomokkal

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

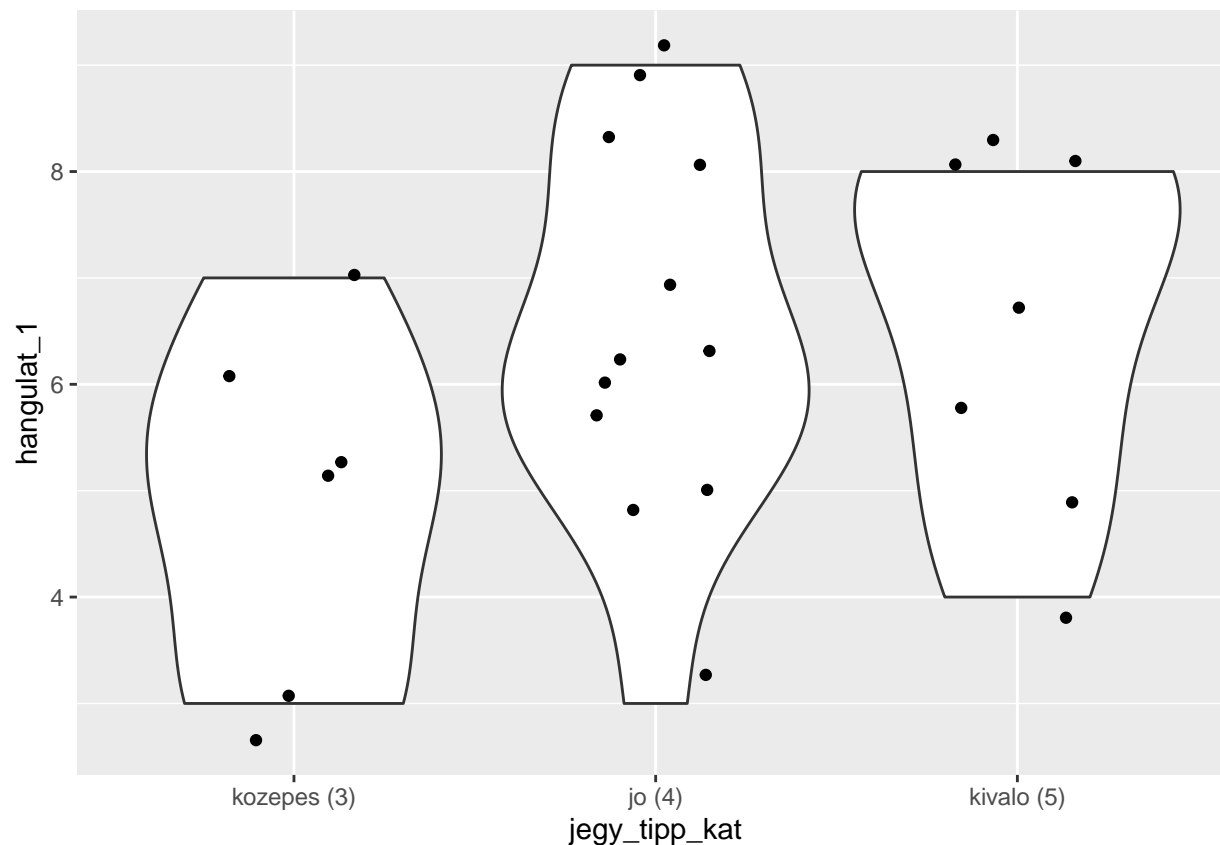
```
## `stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.
```



```
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)
```

Két numerikus változó kapcsolata

Két numerikus változó közötti kapcsolat jellemzésére általában a korrelációs együtthatót szoktuk használni (`cor()`). A `cor()` funkciót akár több mint két változó páronkénti korrelációjának meghatározására is lehet használni.

A `drop_na()` funkcióval kiejthetjük azokat a megfigyeléseket, ahol a változók bármelyikében hiányzó adat (NA) van. Ha ezt nem tesszük meg, a `cor()` függvény NA eredményt adhatna ha valamelyik változóban NA-val találkozunk.

```
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
```

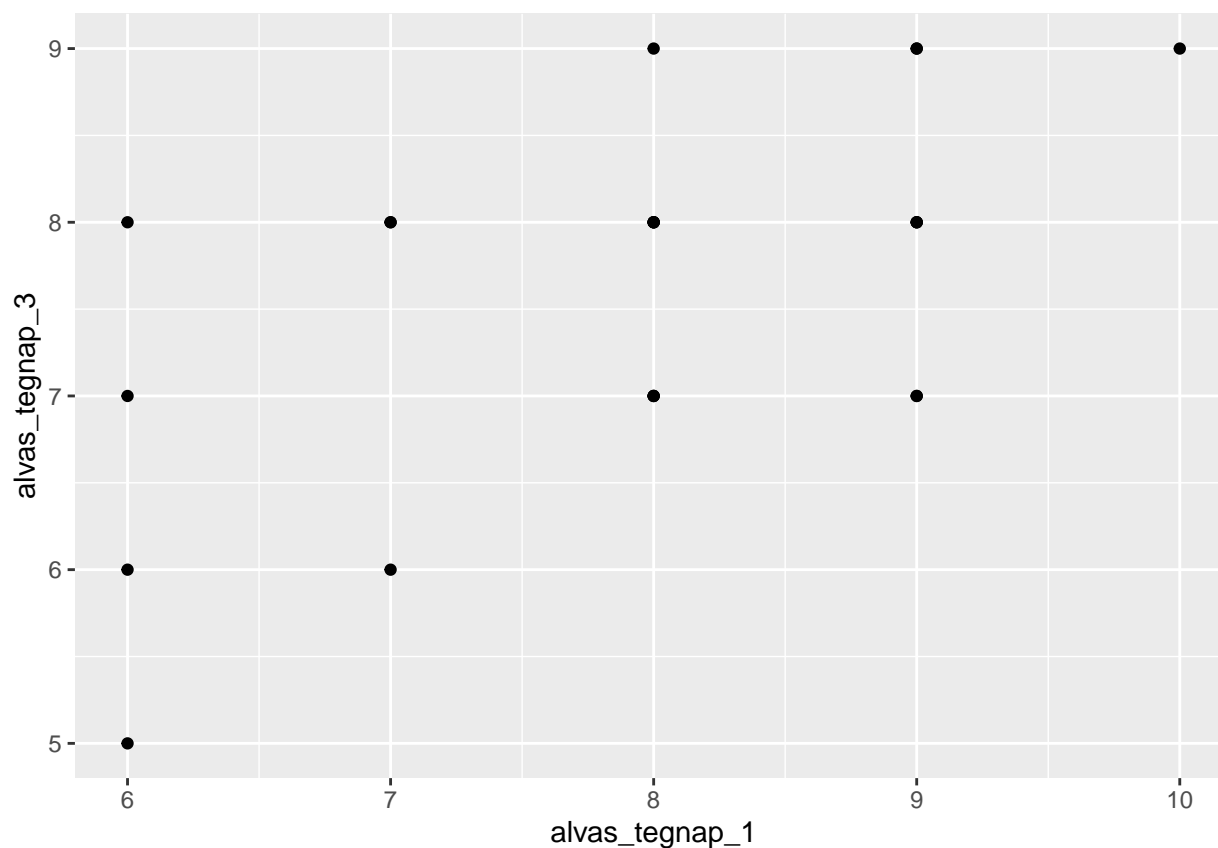
```
orai_adat_corrected %>%
  select(c(alvas_atalaban_1, alvas_tegnap_1, alvas_tegnap_3)) %>%
  drop_na() %>% #
  cor()
```

```
##           alvas_atalaban_1 alvas_tegnap_1 alvas_tegnap_3
## alvas_atalaban_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
```

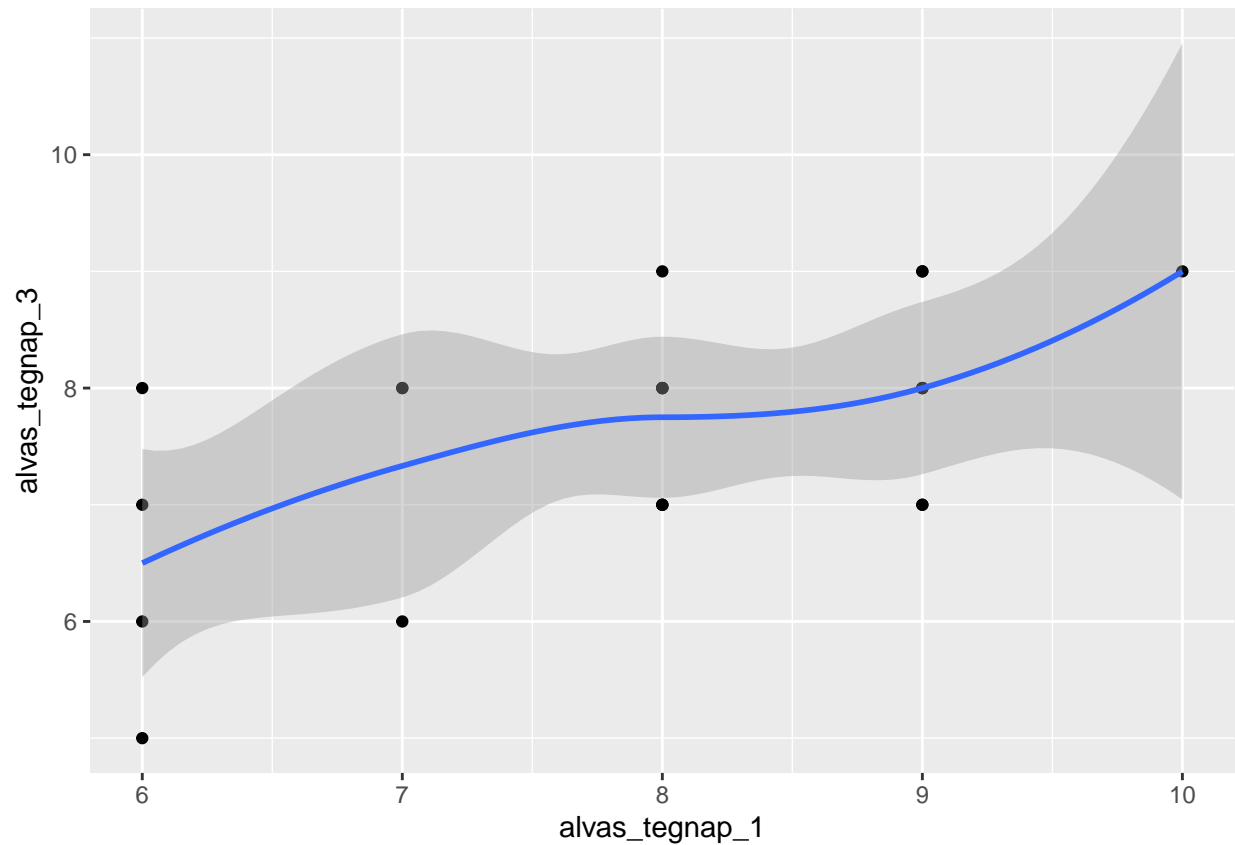
A numerikus változók közötti kapcsolatot általában pont diagrammal szoktuk ábrázolni (`geom_point()`)

A `geom_smooth()` layer hozzáadásával kaphatunk a pontok között meghúzott trendről egy képet. A fekete vonal az úgynevezett trendvonal, a szürke sáv a konfidencia intervallum. Ezekről később még részletesebben beszélünk 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()
```

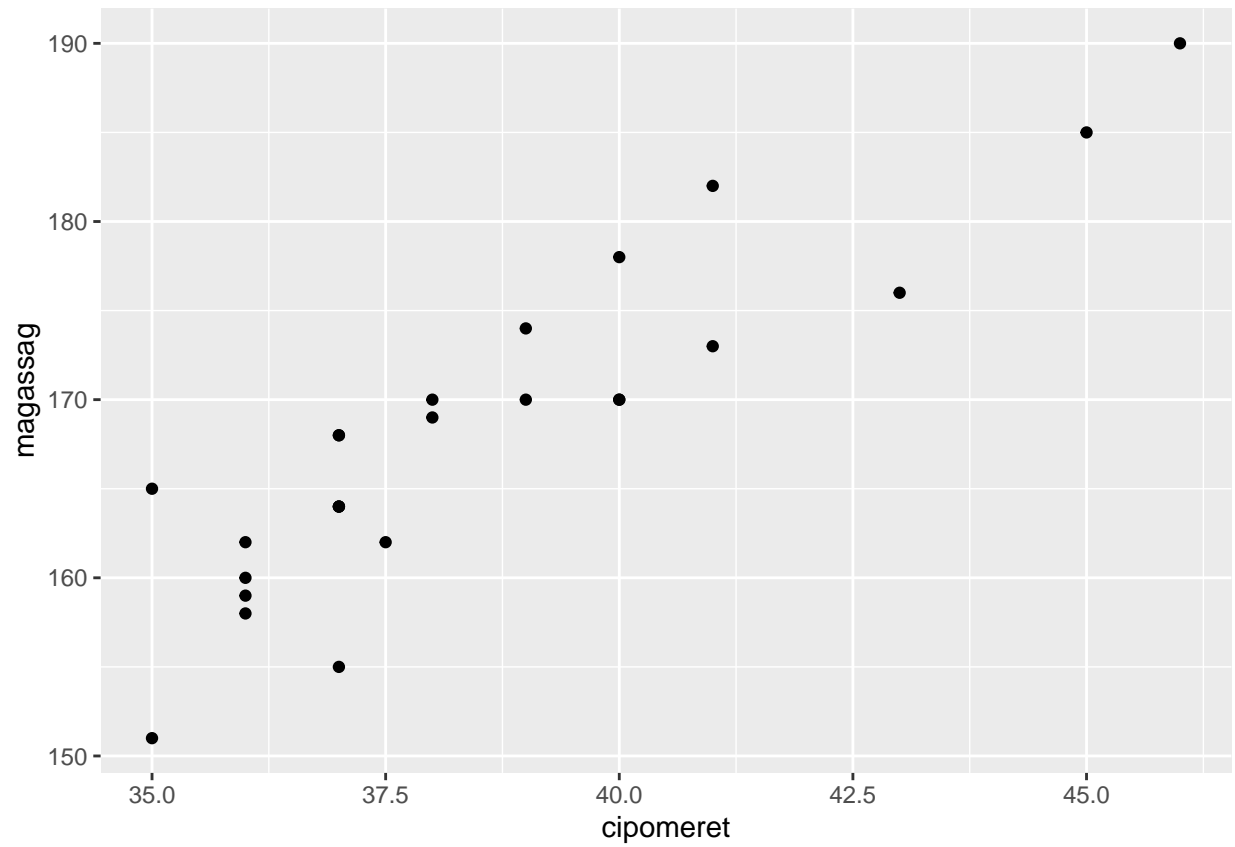


Gyakorlas

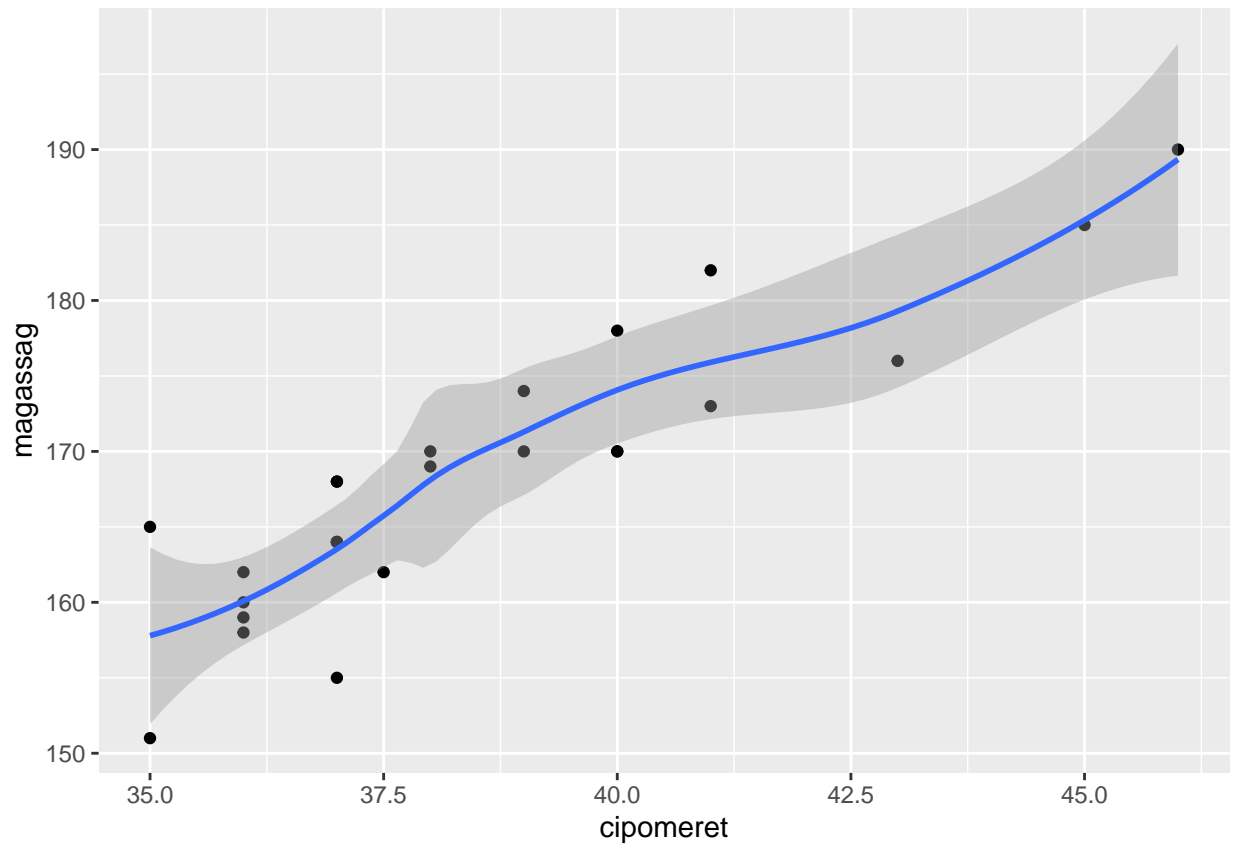
Milyen eros a kapcsolat a magassag es a cipomeret kozott?

- határozd meg a korrelacios egyutthatot a *magassag* es *cipomeret* valtozok kozott
- abrazold a *magassag* es *cipomeret* valtozok kapcsolatát

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



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



Több folytonos változó kapcsolata megjeleníthető például úgy, hogy az egyik változót egy színskálához rendeljük az alábbi módon.

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

