

# Data visualization in R

Zoltan Kekecs

Szeptember 21, 2021

## Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Absztrakt</b>                                       | <b>2</b>  |
| <b>2</b> | <b>Adatábrázolás alapjai a ggplot2-vel</b>             | <b>2</b>  |
| 2.1      | Adatok előkészítése ábra készítésre . . . . .          | 8         |
| <b>3</b> | <b>Geomok</b>  | <b>10</b> |
| 3.1      | Geomok eloszlás vizsgálatára . . . . .                 | 10        |
| 3.2      | Geomok két változó kapcsolatának vizsgálatára. . . . . | 13        |
| <b>4</b> | <b>Az ábrák testre szabása</b>                         | <b>16</b> |
| 4.1      | Szöveg ábrára rakása . . . . .                         | 16        |
| 4.2      | Pozíció (Position) . . . . .                           | 18        |
| 4.3      | Koordináta rendszerek . . . . .                        | 21        |
| 4.4      | Ábra panelekre osztása (faceting) . . . . .            | 22        |

# 1 Absztrakt

Ezen a gyakorlaton megtanuljuk hogyan készíthetünk szemléletes ábrákat az adatainkban található összefüggések megjelenítésére. A gyakorlat bemutatja az eloszlásfüggvények, hisztogram, pont-, oszlop-, vonal- és dobozdiagramok elkészítésének módját a ggplot2 package segítségével.

## 2 Adatábrázolás alapjai a ggplot2-vel

A gyakorlat során egy randi applikációból származó adattáblával fogunk dolgozni. Az adatok a Lovoo nevű appból származnak. Az adatokat Jeffrey Mvutu Mabilama gyűjtötte egy saját programmal 2015-ben. (Az adatokat a saját Lovoo profilján keresztül gyűjtötte, és csak azoknak az adatát érte el, akit neki az App ajánlott, ezért csak nők szerepelnek az adatbázisban. Emiatt vélhetően az adatbázis más szempontból is torzított.)

Az adatok a Kaggle-ről elérhetőek, ami egy adatelemzéssel és machine learninggel foglalkozó oldal.

<https://www.kaggle.com/jmmvutu/dating-app-lovoo-user-profiles>

Ezt az adatbázist betölthetjük az alábbi kóddal. A kód lefuttatása után a környezetben (environment) megjelenik a lovoo\_data adattábla.

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

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

lovoo_data <- lovoo_data %>%
  mutate(isOnline = factor(isOnline),
         verified = factor(verified),
         isNew = factor(isNew))
```

Nézzük meg az adattábla alapvető tulajdonságait a megszokott módon.

```
lovoo_data
View(lovoo_data)
str(lovoo_data)
```

Az ábrázoláshoz a **ggplot2** nevű csomagot használjuk majd.

Töltsd be ezt a csomagot! A **tidyverse** csomag tartalmazza a ggplot2-t, így az alábbi kódban ezen keresztül töltöm be a ggplot2-t. Így a %>% (pipe) operátort is használni tudjuk és egyéb dplyer funkciókat.

```
library(tidyverse)
```

A ggplot2 csomag rengeteg funkciót tartalmaz. Ezek átlátásához segítséget nyújthat a ggplot cheatsheet. (Több csomaghoz is van ilyen, érdemes rájuk keresni!) <https://www.maths.usyd.edu.au/u/UG/SM/STAT3022/r/current/Misc/data-visualization-2.1.pdf>

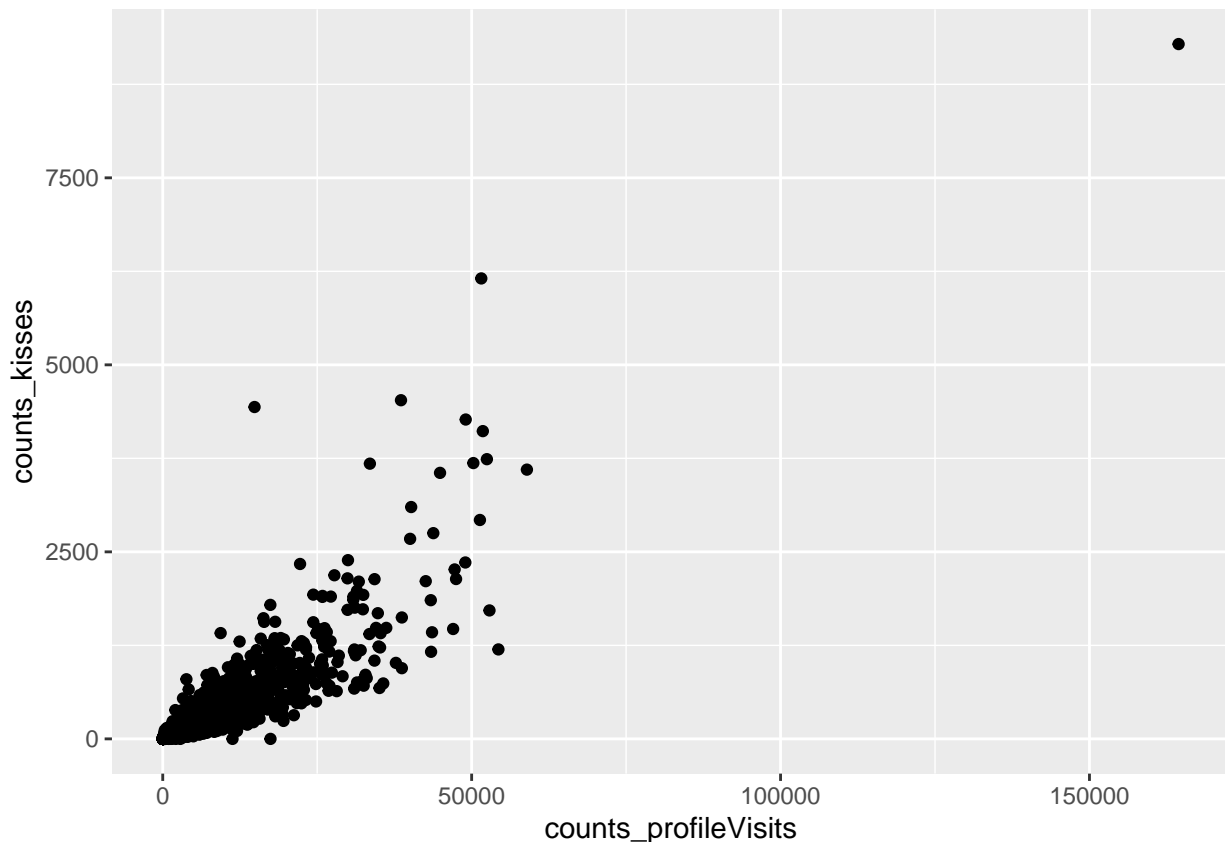
Először is ábrázoljuk, hogyan függ a profil megtekintések száma a like-ok (a lovoo-ban kissek) számával!

Nézd hogyan lehet a **pipe** `%>%` operátort használni ahhoz, hogy a ggplot funkciót a `lovoo_data` adatbázisra alkalmazzuk.

A ggplotba a sorok végén “+” jelet használunk ahhoz, hogy **új elemet** adjunk hozzá az ábránkhoz. Az aesthetics `aes()` funkcióval határozzuk meg, hogy az adattáblából **melyik változókat** akarjuk ábrázolni és melyik tengelyeken, vagy egyéb vizualizációs elemekben. A `geom_...*` funkciókkal határozzuk meg, **milyen típusú ábrát** szeretnénk rajzolni.

Az hogy **mennyire jár együtt a megtekintések és a like-ok** jól látszik egy **pontdiagramon**, ezért most a `geom_point()` geomet használjuk. Ez minden egyes megfigyelést egy pontként ábrázol egy kétdimenziós koordináta-rendszerben.

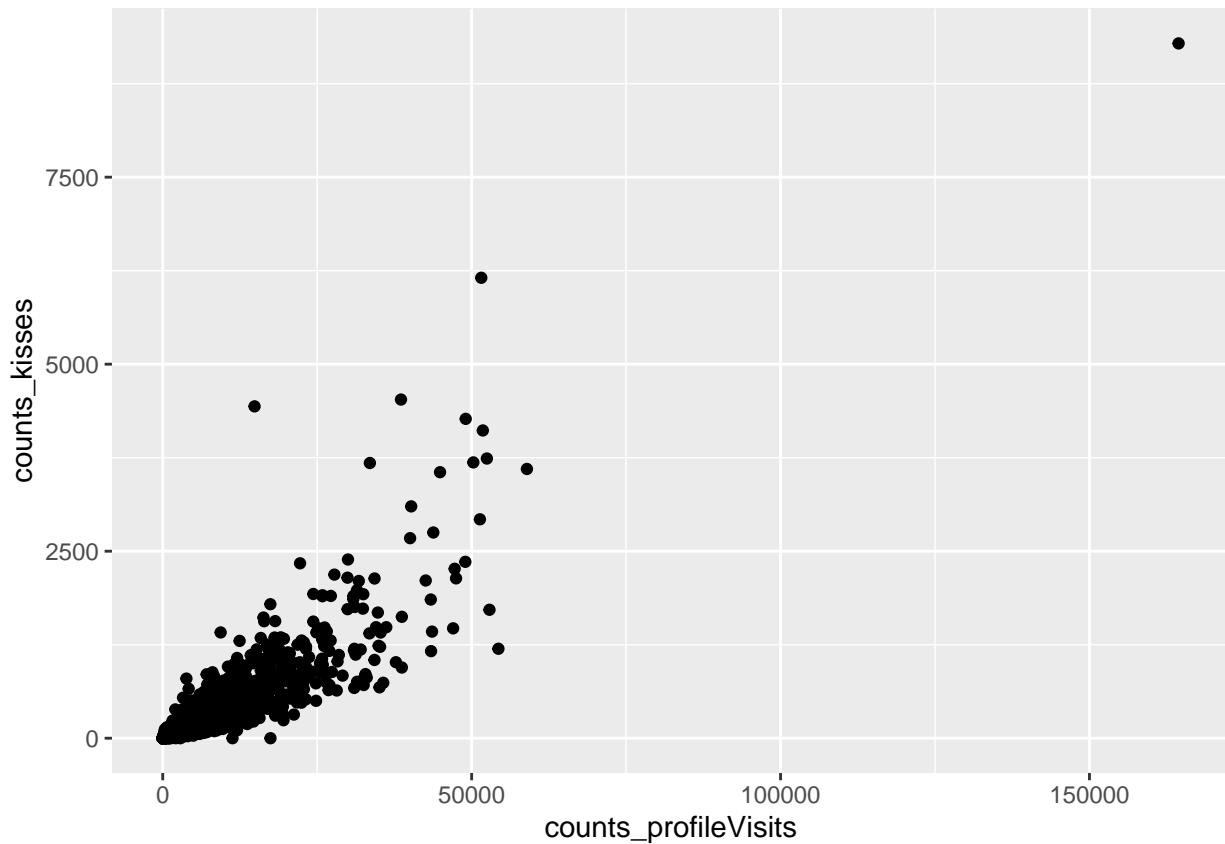
```
lovoo_data %>%  
  ggplot() +  
    aes(x = counts_profileVisits,  
        y = counts_kisses) +  
    geom_point()
```



Az ábránkat is, mint minden más az R-ben, elmenthetjük egy **objektumba**, és amikor kilistázzuk ezt az objektumot (vagyis a konzolban lefuttatjuk az objektum nevét), akkor az ábra újra megjelenik.

```
plot1 <- lovoo_data %>%
  ggplot() +
    aes(x = counts_profileVisits,
        y = counts_kisses) +
    geom_point()
```

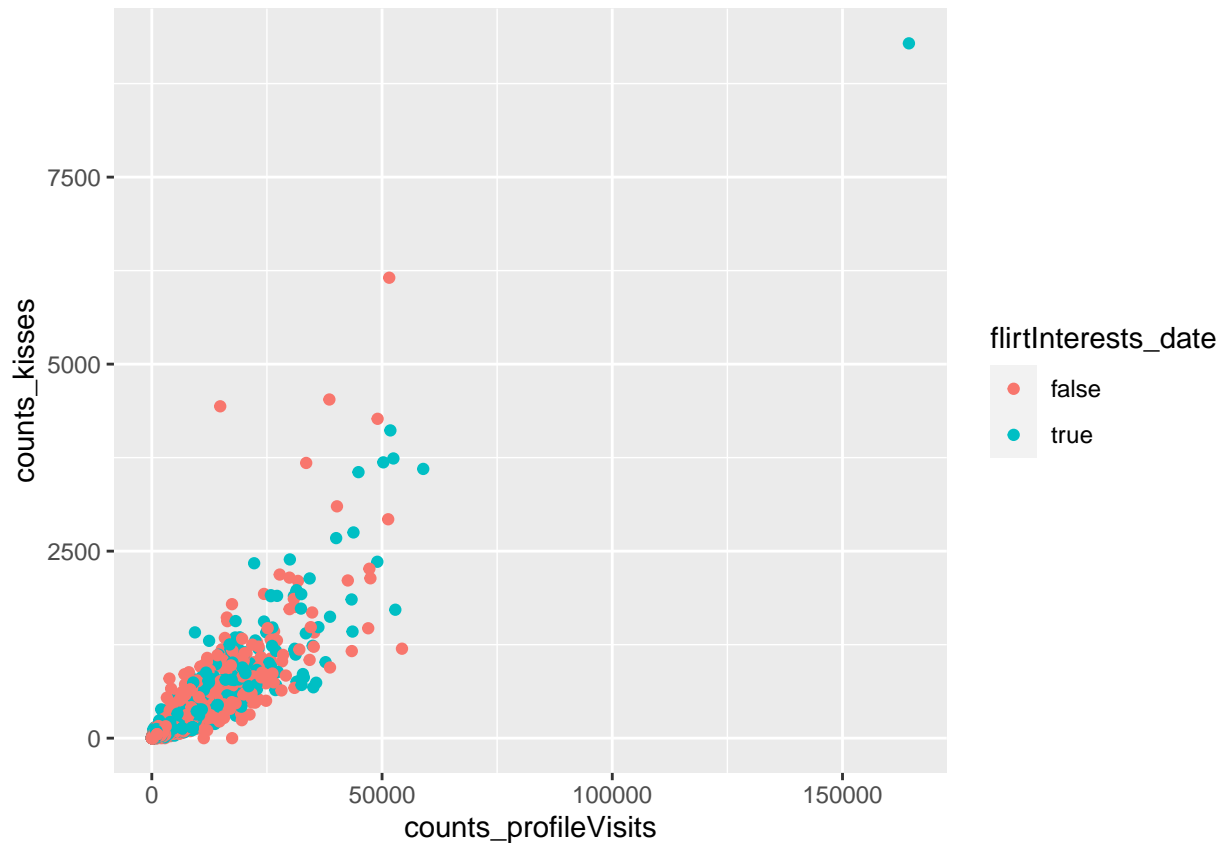
plot1



Az alábbi példán láthatjuk, hogy hogyan tudunk egy **új változót bevonni** a megjelenítésben. Ebben az esetben azt, hogy az illető érdeklődik-e randizás iránt (**flirtInterests\_date**) fogjuk színekkel ábrázolni. Mivel a **geom\_point**-ot használjuk, ez a pontok színét fogja befolyásolni, de ha más geomot használnánk, azokban is hatna ez a színezésre, hiszen az **aes()** általános aesthetics részben specifikáltuk.

```
plot2 <- lovoo_data %>%
  ggplot() +
    aes(x = counts_profileVisits,
        y = counts_kisses,
        color = flirtInterests_date) +
    geom_point()
```

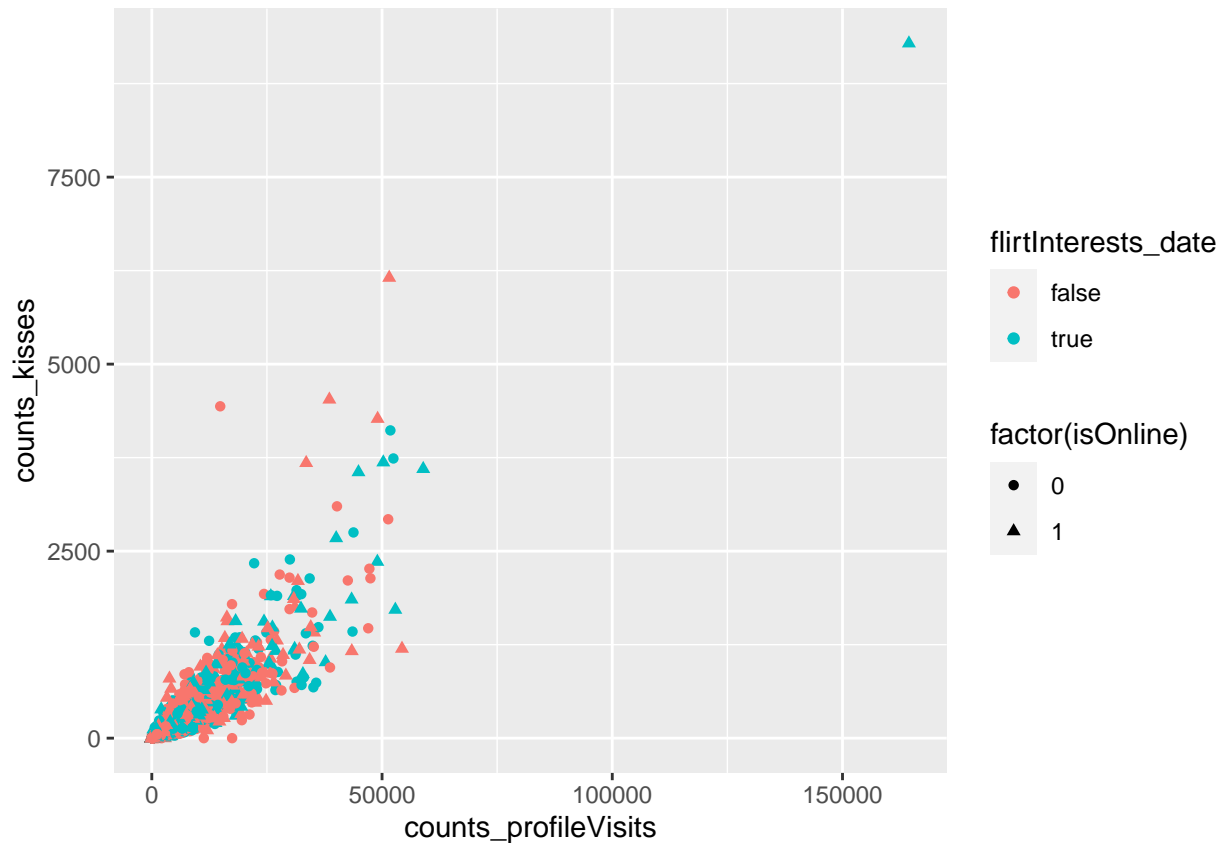
plot2



Nem kell mindig kiírunk a teljes ábra kódot amikor valami új elemet szeretnénk hozzáadni az ábrázoláshoz. Ha az ábrát korábban elmentettük objektumként, akkor az **objektumhoz + jellel hozzáadhatjuk az új elemeket**.

Erről az ábráról például látszik hogy a legtöbb like-ot és kiss-t kapó user-ek az adatletöltés pillanatában online voltak (ami akár utalhat az általános jelenlétükre az appban).

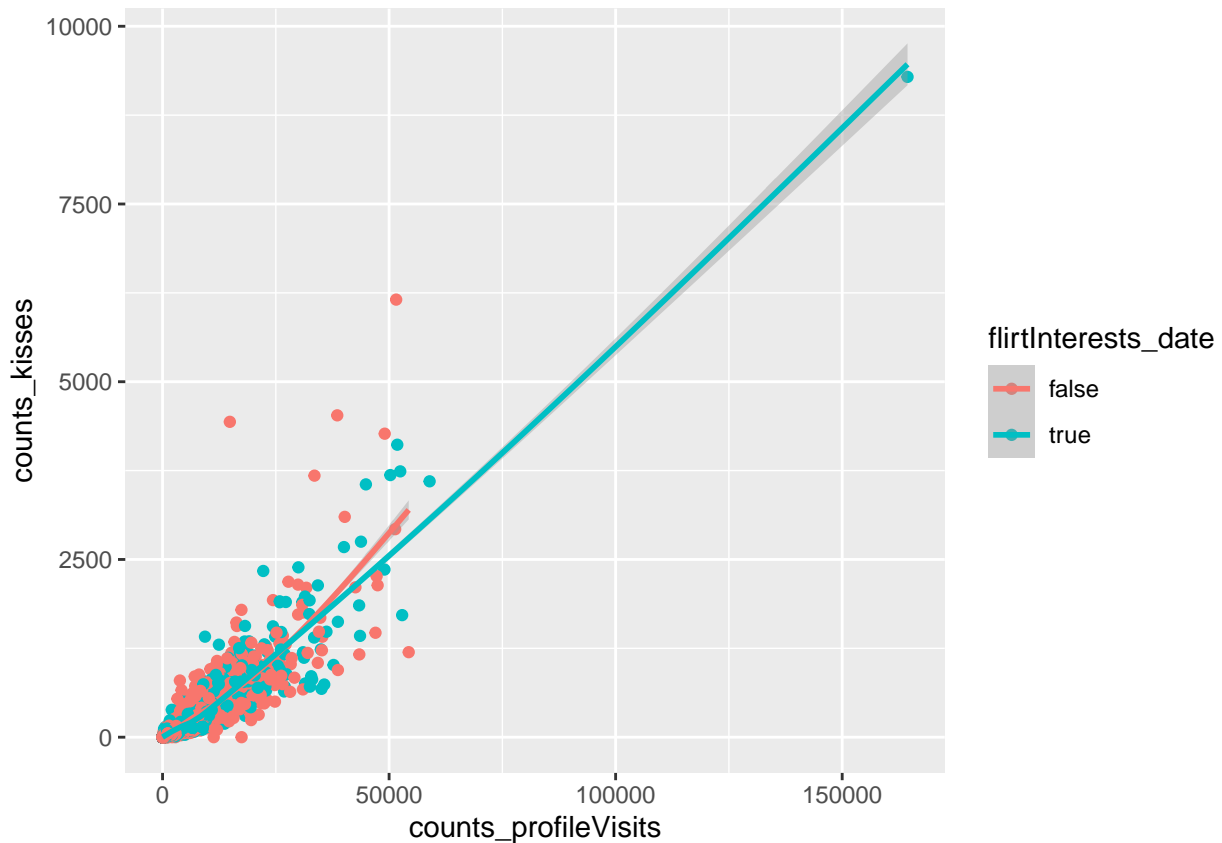
```
plot2 + aes(shape = factor(isOnline))
```



Hozzáadhatunk új geomokat is az ábrához hasonló módon. Itt például a **geom\_smooth** geomot adtuk hozzá a korábbi ábrához, ami egy vonalat illeszt az adatpontokra, és ezzel igyekszik vizualizálni az adatokban lévő trendeket.

```
lovoo_data %>%
  ggplot() +
    aes(x = counts_profileVisits,
        y = counts_kisses,
        color = flirtInterests_date) +
    geom_point() +
    geom_smooth()

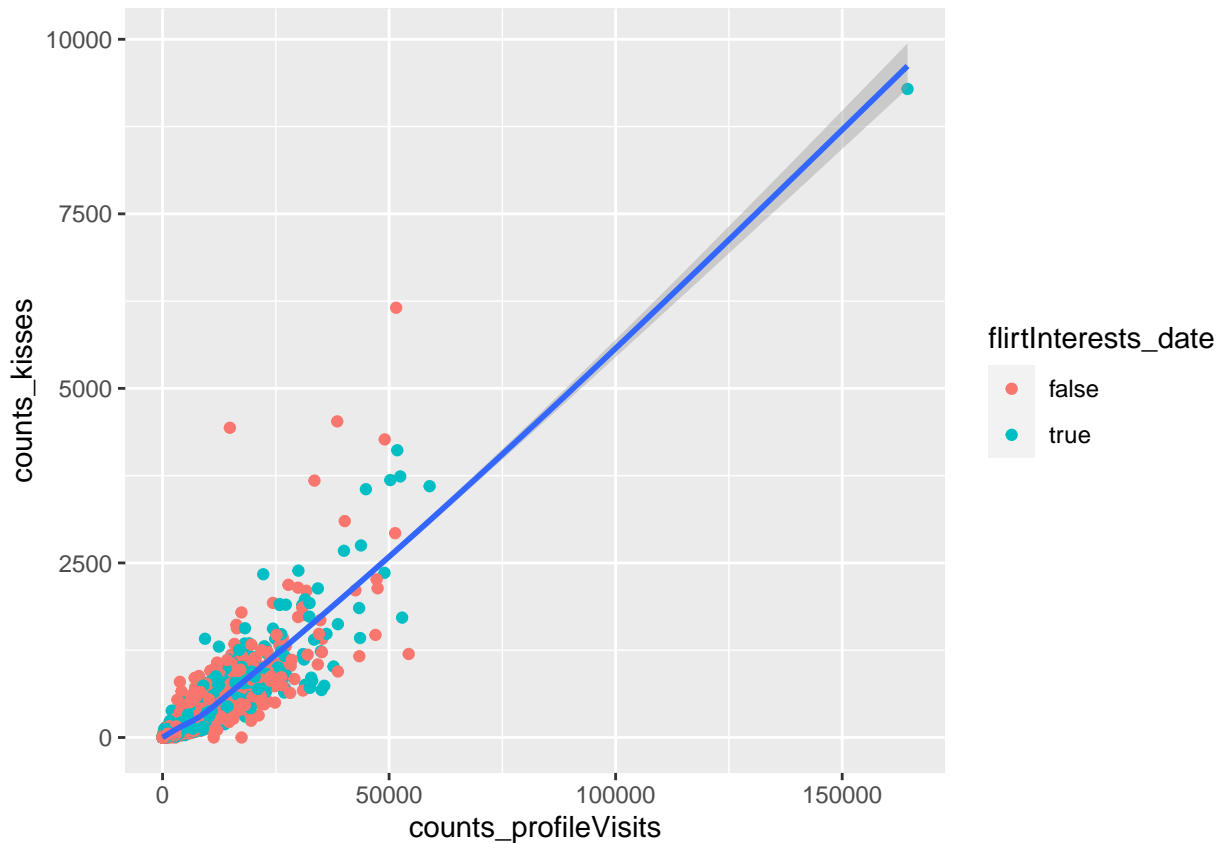
## 'geom_smooth()' using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Mivel az egész ggplot-ra vonatkozó `aes()` funkció tartalmazza a `color = flirtInterests_date` részt, ezért ez **minden geomra hat**, így látható hogy a `geom_smooth` vonalai is a `flirtInterests_date` csoportonként lettek kirajzolva, mindegyik a megfelelő színnel. Azonban megtehetjük, hogy az egyes változókat csak bizonyos geomokon jelenítjük meg. Ezt úgy tudjuk elérni ha a **geom funkcióján belül** specifikálunk egy `aes()` függvényt. Az alábbi kódban a szín szerinti csoportosítás csak a pontokban jelenik meg, a simított vonalban nem

```
lovoo_data %>%
  ggplot() +
    aes(x = counts_profileVisits,
        y = counts_kisses) +
    geom_point(aes(color = flirtInterests_date)) +
    geom_smooth()

## 'geom_smooth()' using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Ha szeretnéd egyes **geomok tulajdonságait konstans értékre állítani** ahelyett hogy az adatok alapján változnának (pl. szeretnéd egy geom színét átszínezni anélkül hogy ez megfigyelésenként vagy adatcsoportonként változna), az adott paramétert a **geom függvényén belül** kell megadni. Ha használasz `aes()` függvényt is, akkor fontos hogy ez a paraméter az `aes()` függvényen kívül legyen specifikálva.

Alább a pontok formáját és kitöltési színét, valamint a simított vonal színét állítjuk be konstans értékekre.

A pontok formájának (`shape`) meghatározásához számokat szoktunk használni. Az hogy melyik szám mit jelent itt találod: <http://www.sthda.com/english/wiki/ggplot2-point-shapes>

A színeket be lehet írni angolul. Egy részletesebb útmutató erről: <http://sape.inf.usi.ch/quick-reference/ggplot2/colour>

```
lovoo_data %>%
  ggplot() +
    aes(x = counts_profileVisits,
        y = counts_kisses) +
    geom_point(aes(color = flirtInterests_date), shape = 21, fill = "white") +
    geom_smooth()
```

## 2.1 Adatok előkészítése ábra készítésre

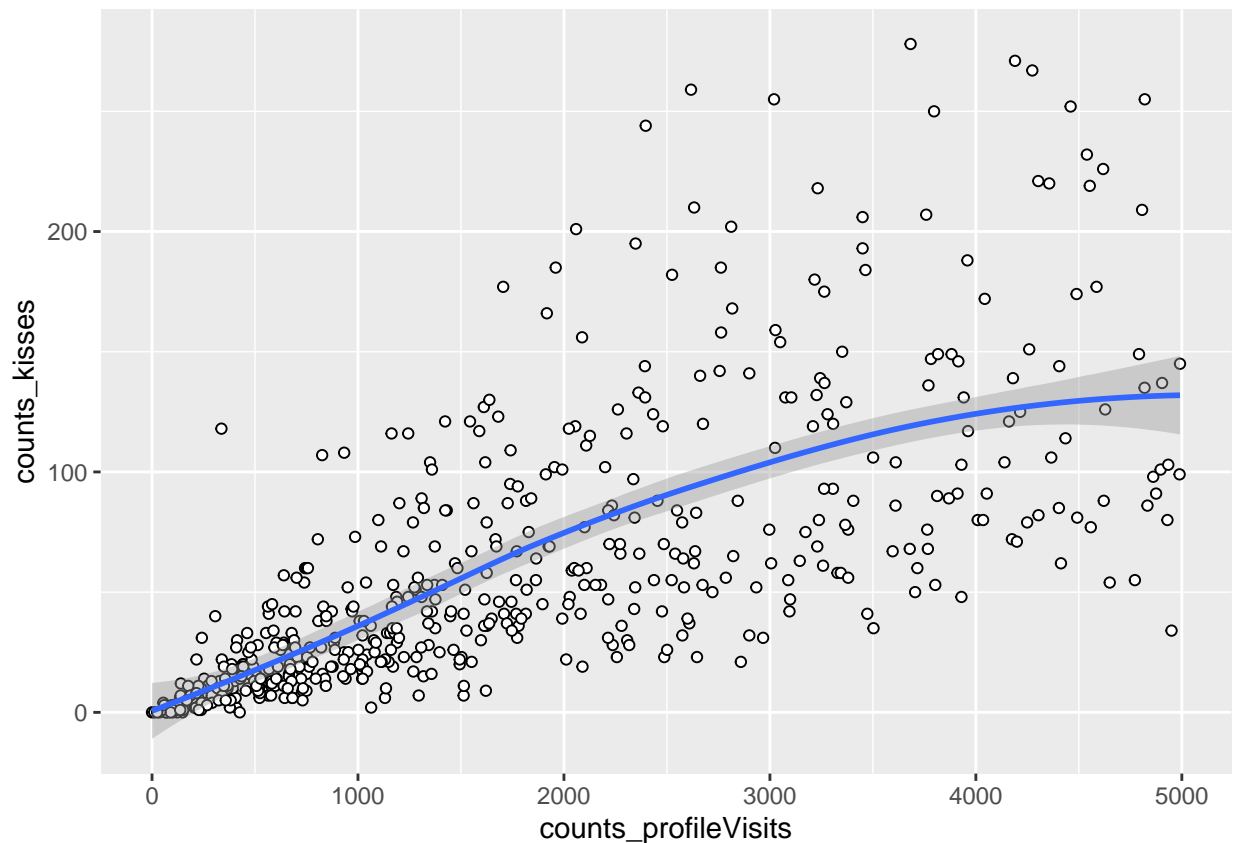
Ahogy azt korábban is láttuk, a ggplot egy pipe végére is berakható, így a korábban tanult adatkezelő funkciókkal előkészítheted az adatokat, amit ábrázolni akarsz.

Például az alábbi kóddal elérjük, hogy csak az 5000 profil megtekintés alatti és a 300 kiss alatti felhasználók adatait ábrázoljuk. Azt is meghatározzuk, hogy csak a külső forrásból megerősített user-ek adatai jelenjenek meg az ábrán.



```
lovoo_data %>%
  filter(counts_kisses < 300, counts_profileVisits < 5000, verified == 1) %>%
  ggplot() +
    aes(x = counts_profileVisits,
         y = counts_kisses) +
    geom_point(shape = 21, fill = "white") +
    geom_smooth()
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```




---

### Gyakorlás

Továbbra is használjuk a lovoo\_data adatbázist.

- Szűrd az adatokat úgy, hogy csak azoknak a felhasználóknak az adata látsszon, akik nyitottak a randizásra (flirtInterests\_date == "true")
  - Ábrázold az összefüggést az életkor (age) és a profilmegettekintések száma (counts\_profileVisits) között egy pontdiagrammon.
  - Az ábrán az is szerepeljen, hogy a profil külső forrásból megerősített-e (verified).
  - Illessz egy trendvonalat is az ábrára (geom\_smooth), amin belül a method = "lm"-paramétert használd.
-

## 3 Geomok

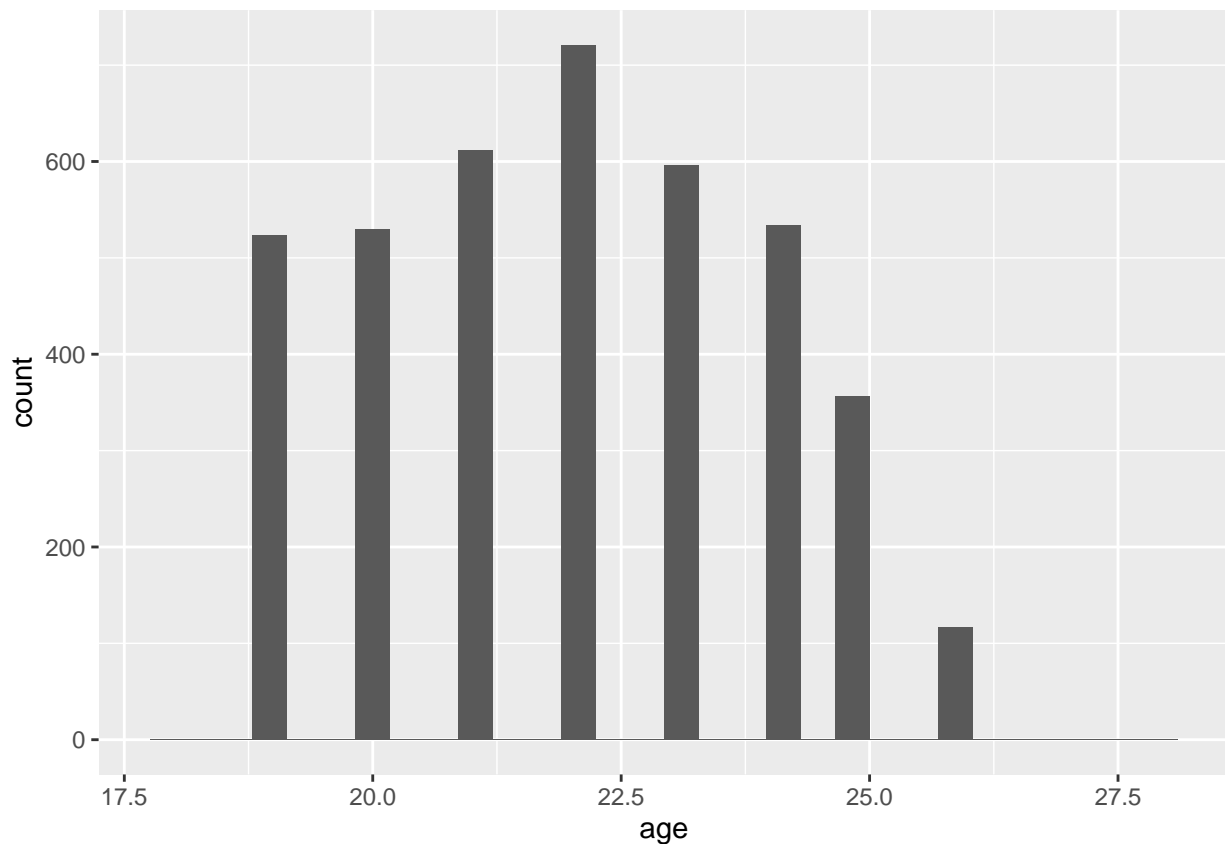
### 3.1 Geomok eloszlás vizsgálatára

Számos fajta geom van. Alább látható néhány gyakran használt geom amit az adatok eloszlásának vizualizációjára szoktunk használni.

#### 3.1.1 Hisztogramm

```
lovoo_data %>%  
  ggplot() +  
    aes(x = age) +  
    geom_histogram()
```

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

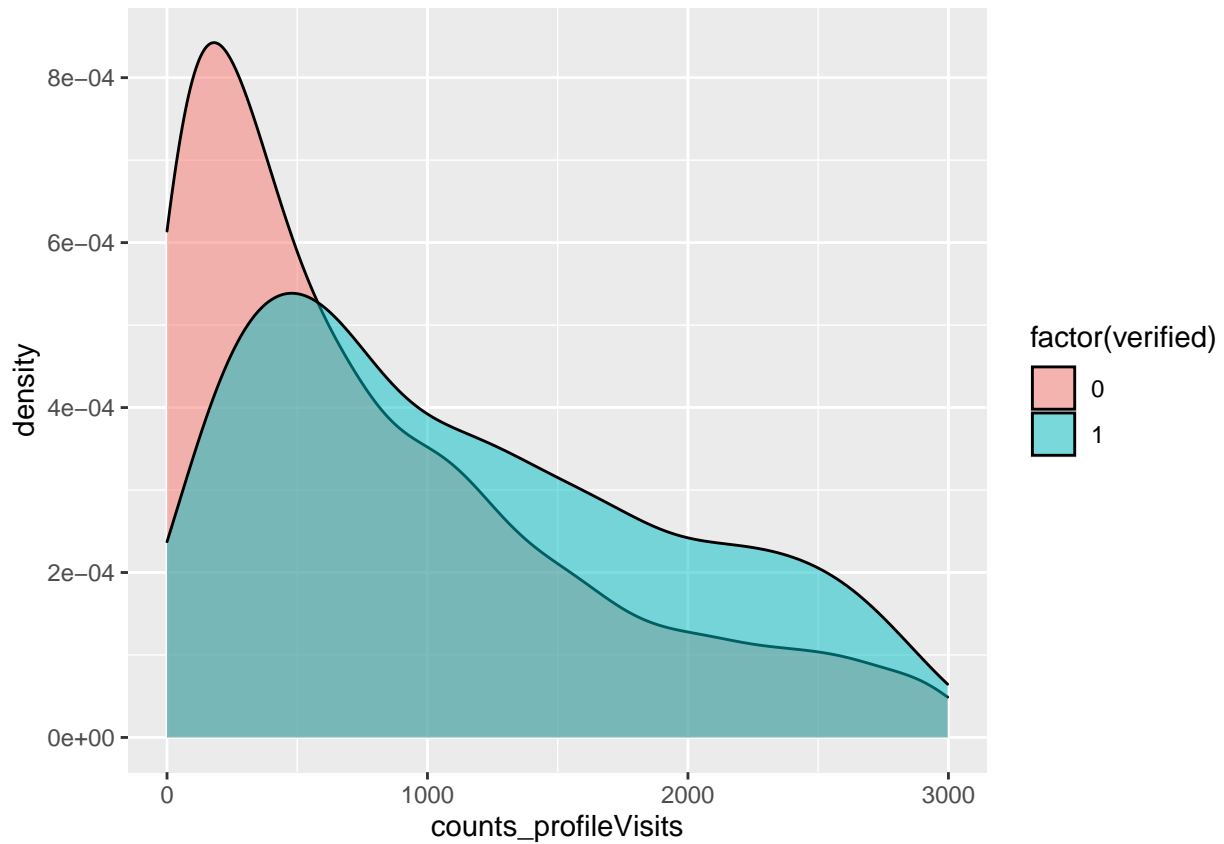


#### 3.1.2 Sűrűségi ábra (density plot)

A sűrűségfüggvény az egyes értékek előfordulásának arányát szemlélteti.

Az alpha-val azt adjuk meg, hogy mennyire átlátszó az ábra, 0-1 közötti értéket vehet fel.

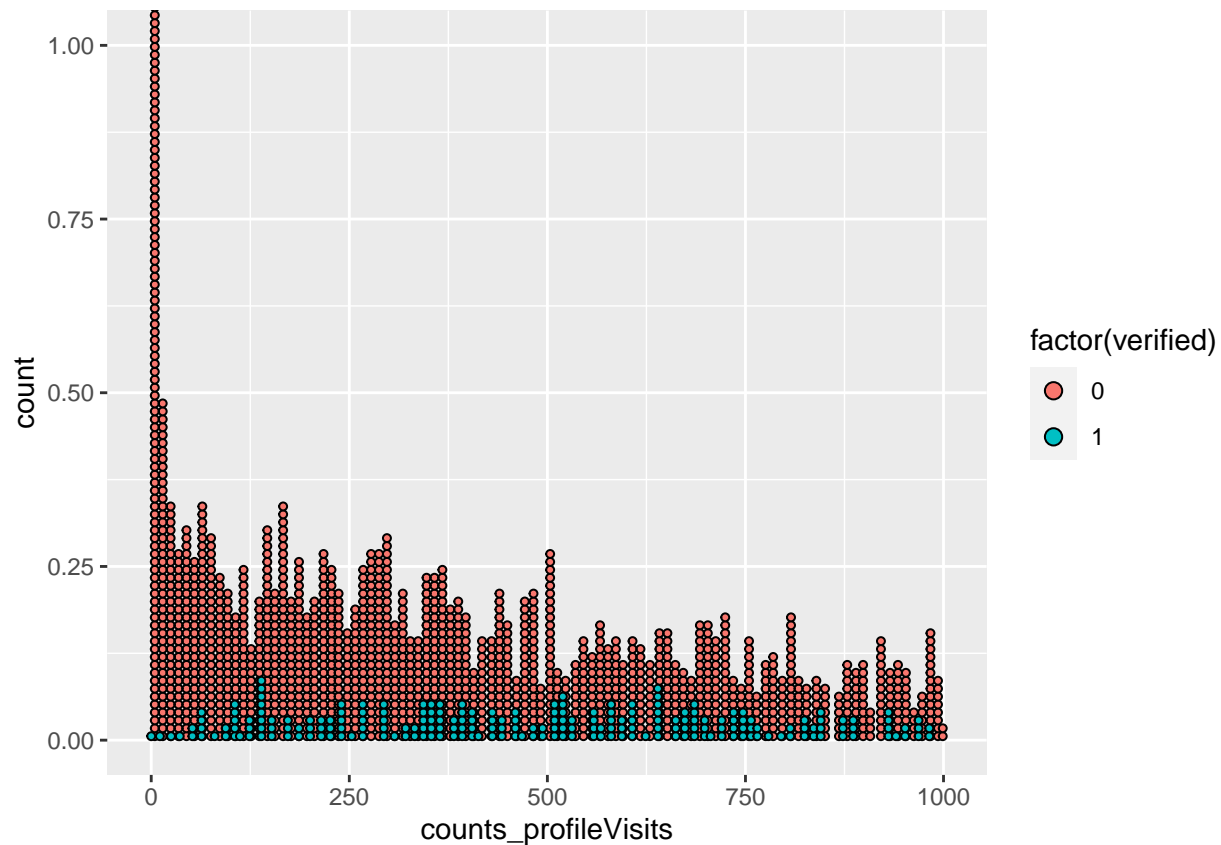
```
lovoo_data %>%
  filter(counts_profileVisits < 3000) %>%
  ggplot() +
    aes(x = counts_profileVisits, fill = factor(verified)) +
    geom_density(alpha = .5)
```



### 3.1.3 Pöttydiagramm (dotplot)

A pöttydiagramban jól látszik a megfigyelések száma is is, ami a sűrűségfüggvényről hiányzik. A `binwidth` paraméterrel beállíthatjuk, hogy az egyes körök hány megfigyelést ábrázoljanak.

```
lovoo_data %>%
  filter(counts_profileVisits < 1000) %>%
  ggplot() +
    aes(x = counts_profileVisits, fill = factor(verified)) +
    geom_dotplot(binwidth = 10)
```



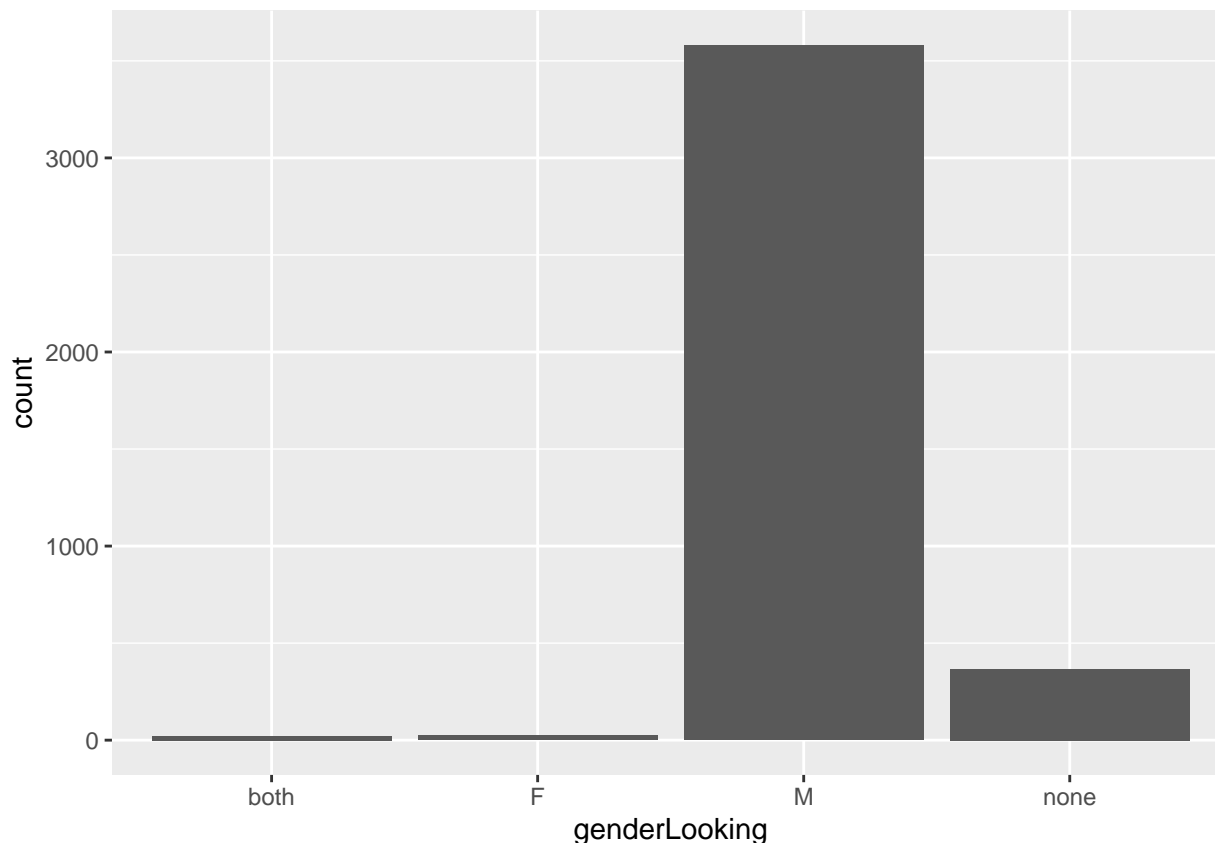
### 3.1.4 Oszlopdiaagram (geom\_bar)

Kategorikus változók eloszlását vizsgálhatjuk az oszlopdiaagrammal.

Itt például azt ábrázoljuk, hogy a felhasználók milyen nemű párt keresnek a lovoo-n.

- F - female (nő)
- M - male (férfi)
- both - both male and female (férfi és nő)
- none - no preference (nincs megjelölve preferencia)

```
lovoo_data %>%
  ggplot() +
    aes(x = genderLooking) +
    geom_bar()
```



### Gyakorlás

Használjuk a lovoo\_data adatbázist a következő gyakorlófeladatokhoz.

- Szűrj az adatokat, hogy csak a külső forrásból megerősített felhasználók adatai szerepeljenek benne (verified == 1)
- Hozz létre egy ábrát, melyet egy “my\_first\_plot” nevű objektumhoz rendelj hozzá. Ezen az ábrán vizsgáld meg a feltöltött képek számának (counts\_pictures) eloszlását. Tetszőleges geomot használhatsz. A ggplot2 cheatsheet segíthet kitalálni, melyik a legjobb geom erre a célra.

<https://www.maths.usyd.edu.au/u/UG/SM/STAT3022/r/current/Misc/data-visualization-2.1.pdf>

Tipp: a counts\_pictures egy folytonos (continuous) változó. Mivel egy változó eloszlását vizualizáljuk, ezért érdemes a cheatsheet “One Variable” dobozából választani geomot.

- Most módosítsd az ábrát úgy, hogy legyen látható, hogy az eloszlás milyen az új felhasználóknál és a régiéknél (isNew) Ehhez használj tetszőleges aes()-t, pl.: color, fill, linetype, size. A ggplot2 cheatsheet segít hogy az általad választott geomnál melyik a releváns aes()

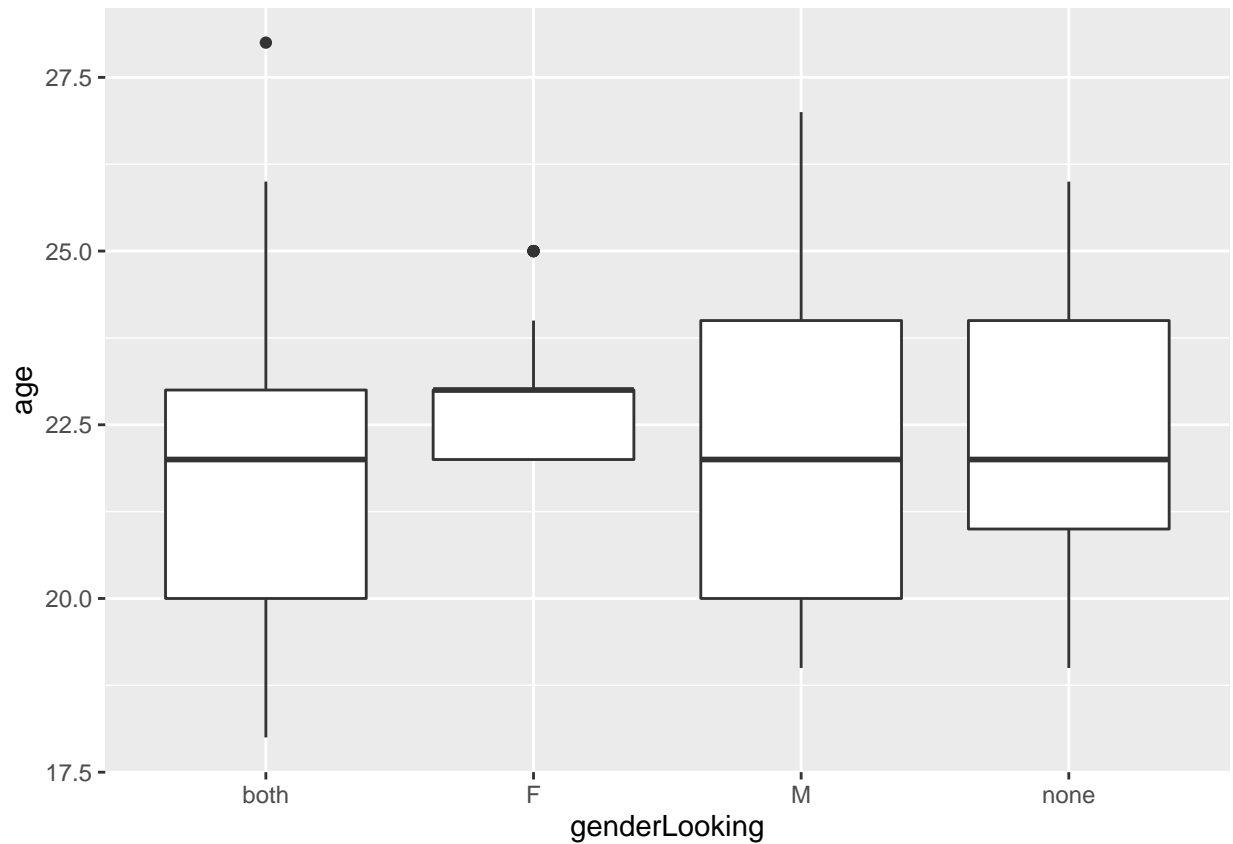
## 3.2 Geomok két változó kapcsolatának vizsgálatára.

Fentebb láthattuk, hogy a geom\_point segítségével két folytonos változó kapcsolatát ábrázolhatjuk. Alább megismerünk újabb geomokat, amik folytonos és egy kategorikus változók kapcsolatának ábrázolására is alkalmasak.

### 3.2.1 Doboz ábra (boxplot)

Az alábbi doboz ábra (boxplot) az életkort mutatja annak metszetében hogy milyen nemű partnert keres az illető. Ez az ábra típus a mediánt mutatja középen, és az adatok szóródását körülötte, a kvartilisek szerint felosztva.

```
lovoo_data %>%  
  ggplot() +  
    aes(y = age, x = genderLooking) +  
    geom_boxplot()
```



### 3.2.2 Hegedű ábra (violin plot)

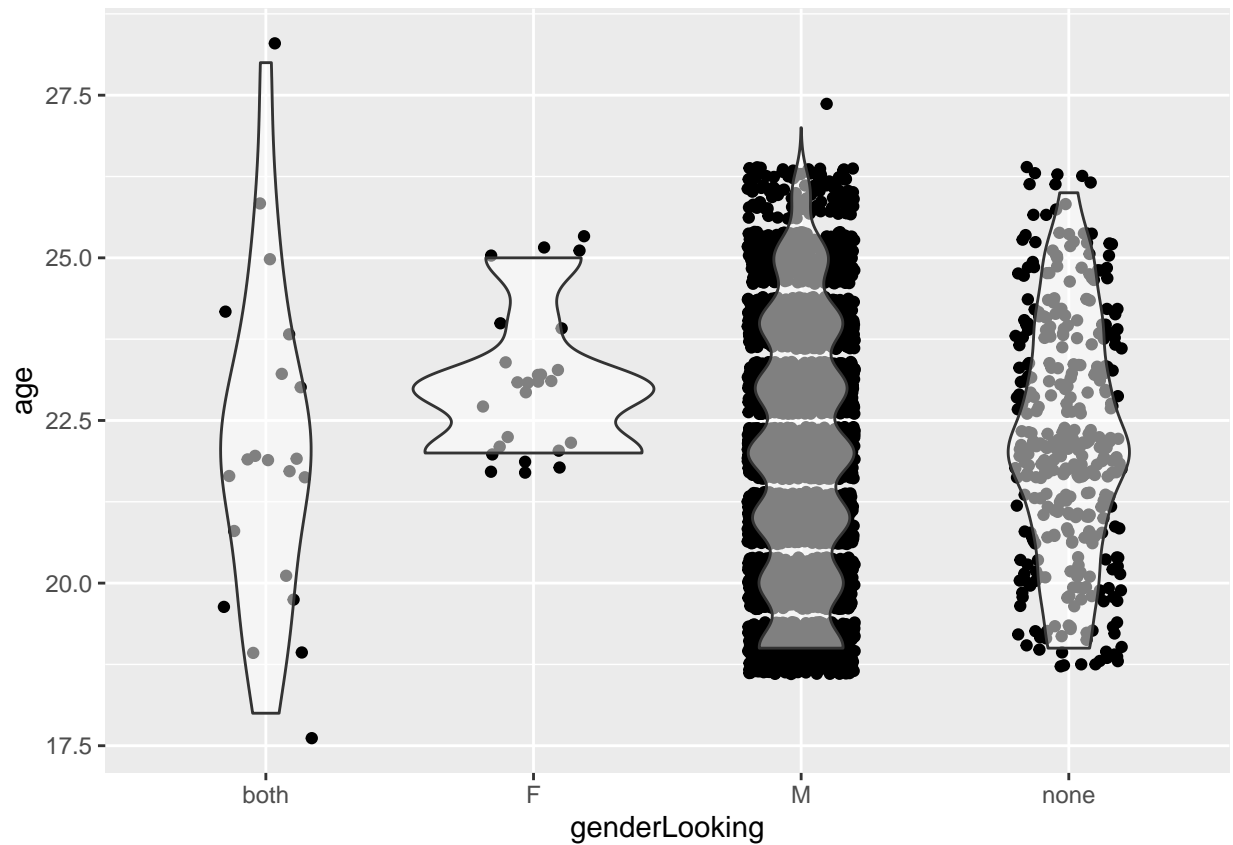
A célja ugyanaz, mint a doboz ábrának, de jobban szemlélteti az adatok eloszlását. Gyakorlatilag a doboz ábra és a density plot keveréke.

```
lovoo_data %>%  
  ggplot() +  
    aes(y = age, x = genderLooking) +  
    geom_violin()
```



A szétszórás “jitter” pozíció segítségével random zajt adhatunk az adatokhoz, így az átfedéseket megszüntetve jobban látjuk az adatpontokat. Erre van egy külön geom is, a `geom_jitter`. Ez ugyan azt az eredményt adja, mintha a `geom_point`-ban a `position`-t “jitter”-ként specifikáltuk volna.

```
lovoo_data %>%  
  ggplot() +  
    aes(y = age, x = genderLooking) +  
    geom_jitter(width = 0.2) +  
    geom_violin(alpha = 0.5)
```



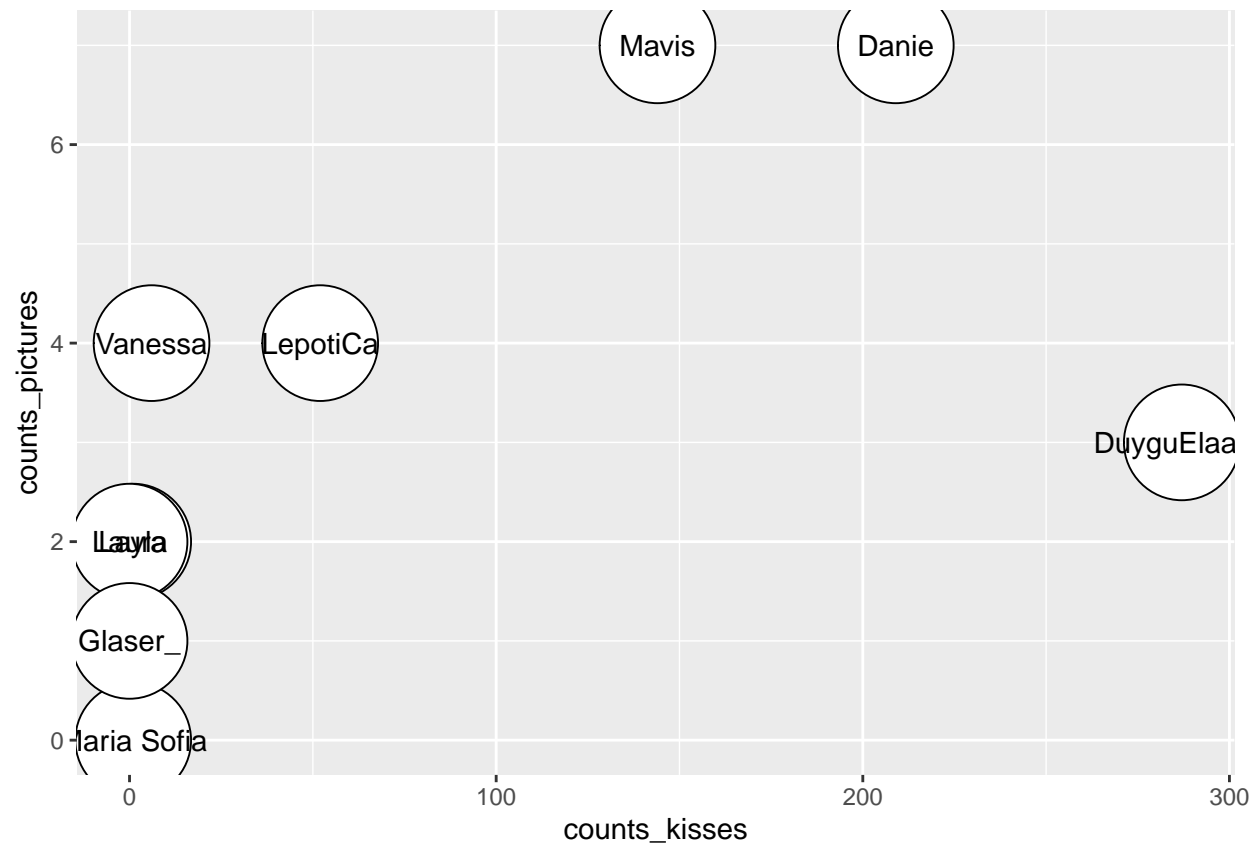
## 4 Az ábrák testre szabása

### 4.1 Szöveg ábrára rakása

használhatjuk a `geom_text`-et

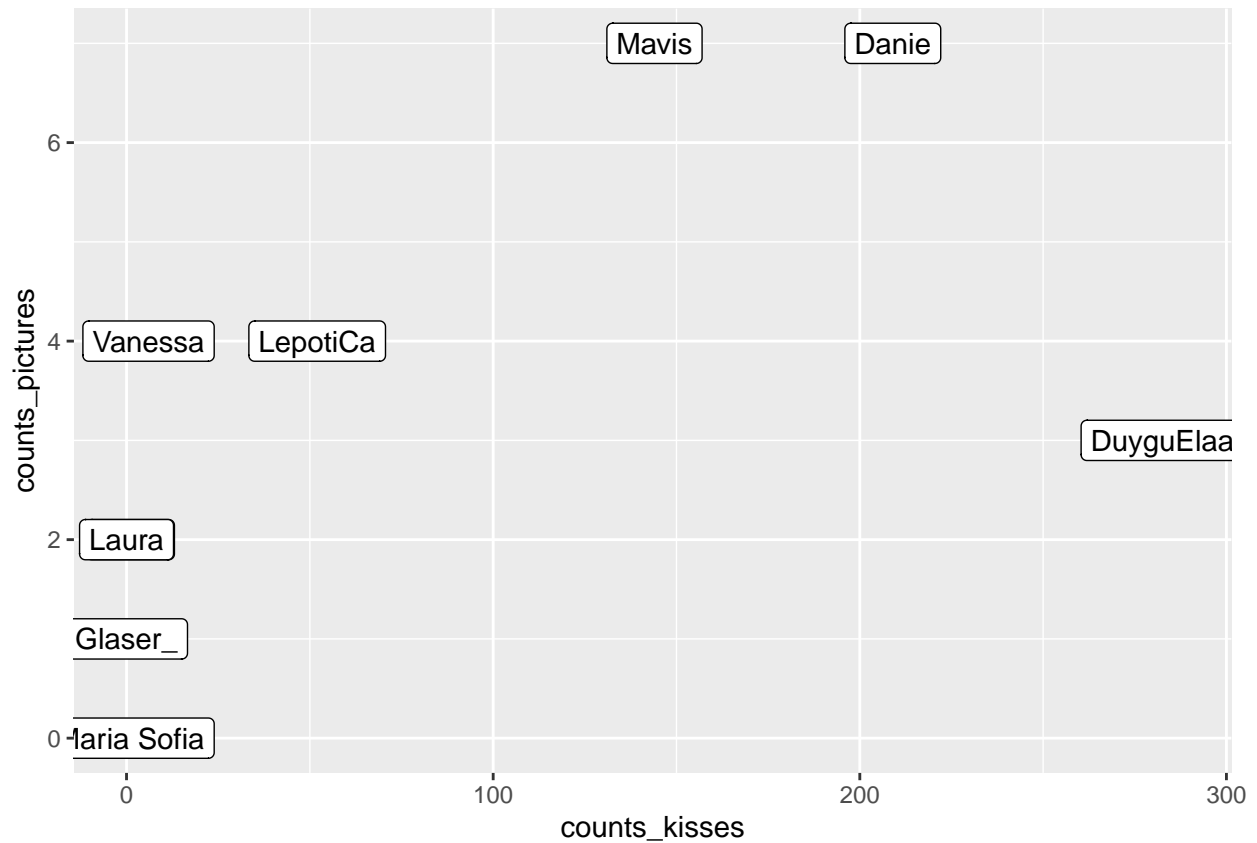
```
lovoo_data %>%
  filter(genderLooking == "F", age < 23) %>%
  ggplot() +
  aes(x = counts_kisses,
       y = counts_pictures,
       label = name) +
  geom_point(shape=21, fill = "white", size = 20) +
  geom_text()
```





vagy a `geom_label`-t.

```
lovoo_data %>%
  filter(genderLooking == "F", age < 23) %>%
  ggplot() +
  aes(x = counts_kisses,
       y = counts_pictures,
       label = name) +
  geom_point(shape=21, fill = "white", size = 5) +
  geom_label()
```

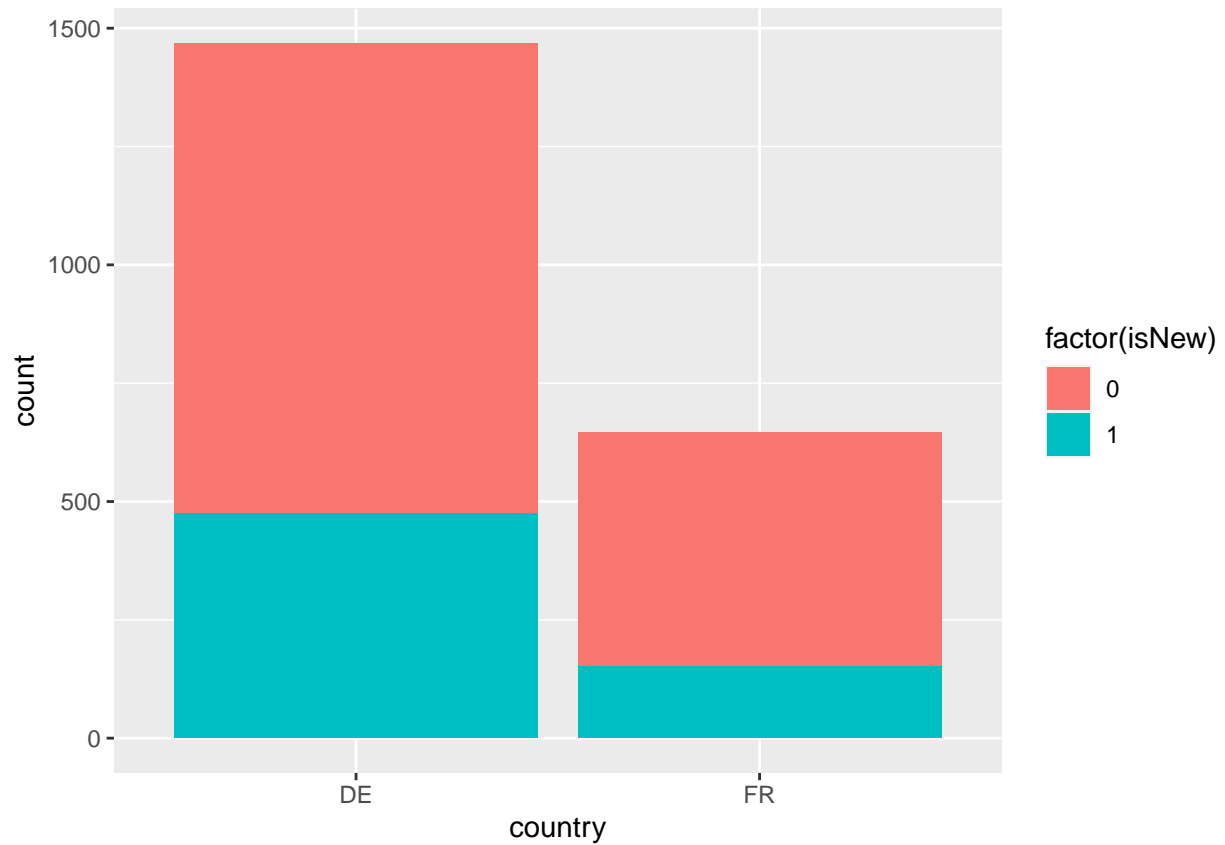


## 4.2 Pozíció (Position)

### 4.2.1 oszlopdiagram (geom\_bar)

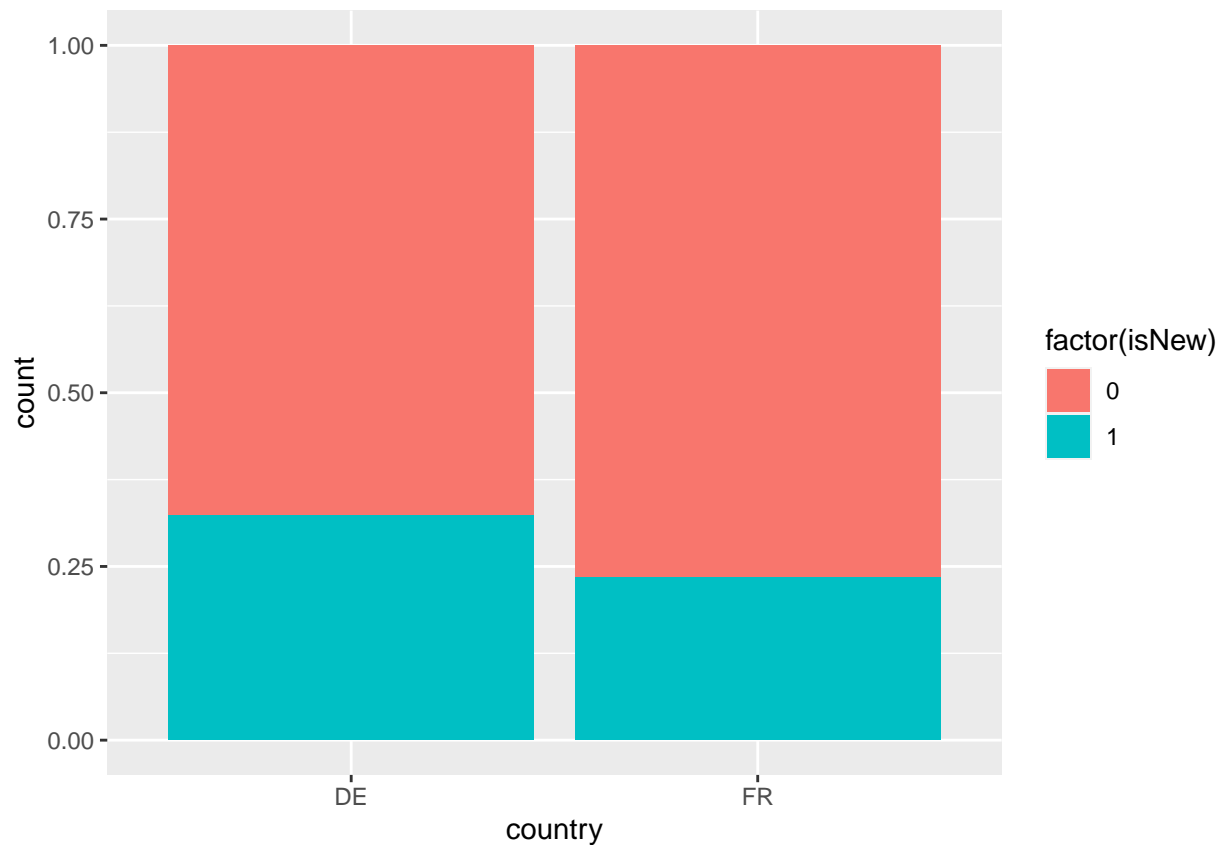
Hozzunk létre egy halmozott barplotot (stacked bar), ami a mennyiséget mutatja

```
lovoo_data %>%
  filter(country == "DE" | country == "FR") %>%
  ggplot() +
    aes(x = country, group = factor(isNew), fill = factor(isNew)) +
    geom_bar(position = "stack")
```



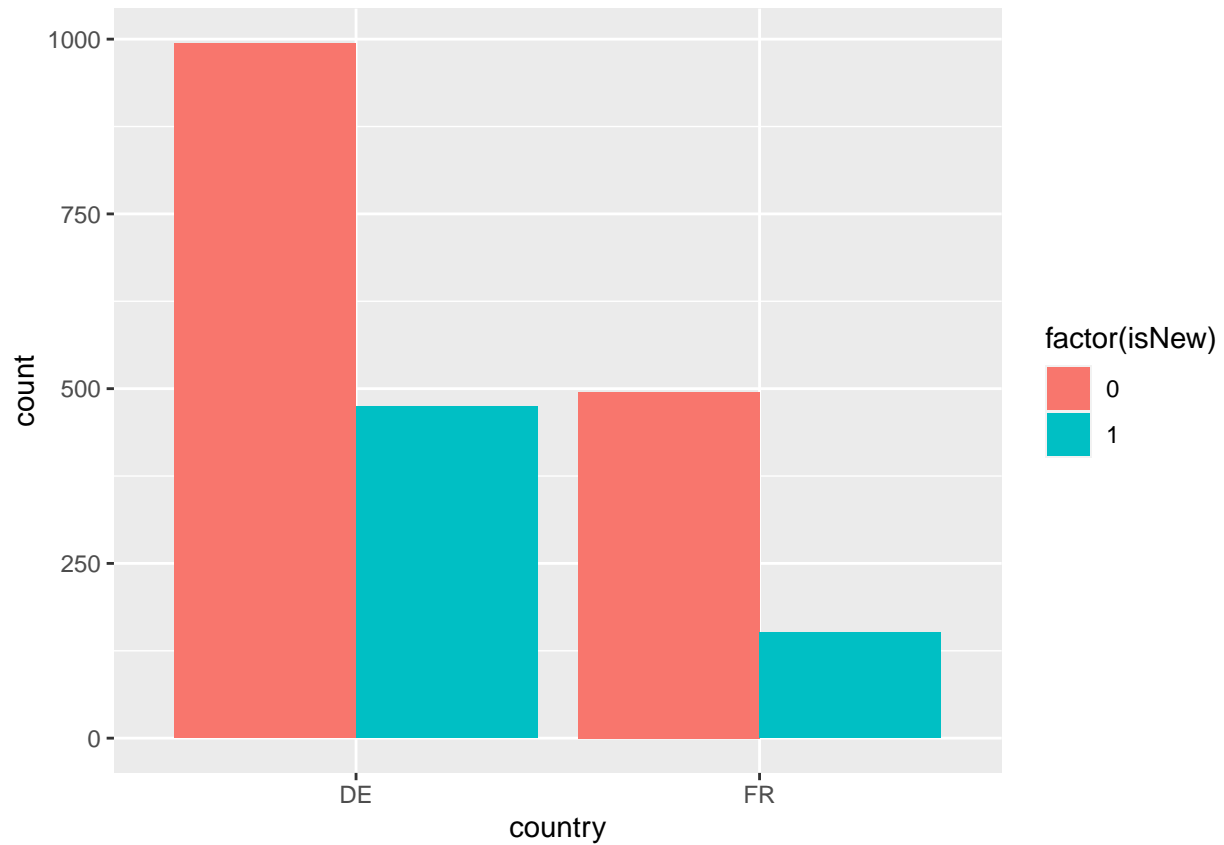
Arányként (proportion) is megmutathatjuk a csoportok mennyisége közti összefüggést, ha a “stack” helyett a “fill” position-t adjuk meg.

```
lovoo_data %>%  
  filter(country == "DE" | country == "FR") %>%  
  ggplot() +  
    aes(x = country, group = factor(isNew), fill = factor(isNew)) +  
    geom_bar(position = "fill")
```



Vagy ábrázoljuk egymás mellett a mennyiséget, hogy könnyebben összehasonlíthatók legyenek a csoportok. Ezt a “dodge” beállítással érhetjük el a position paraméterben.

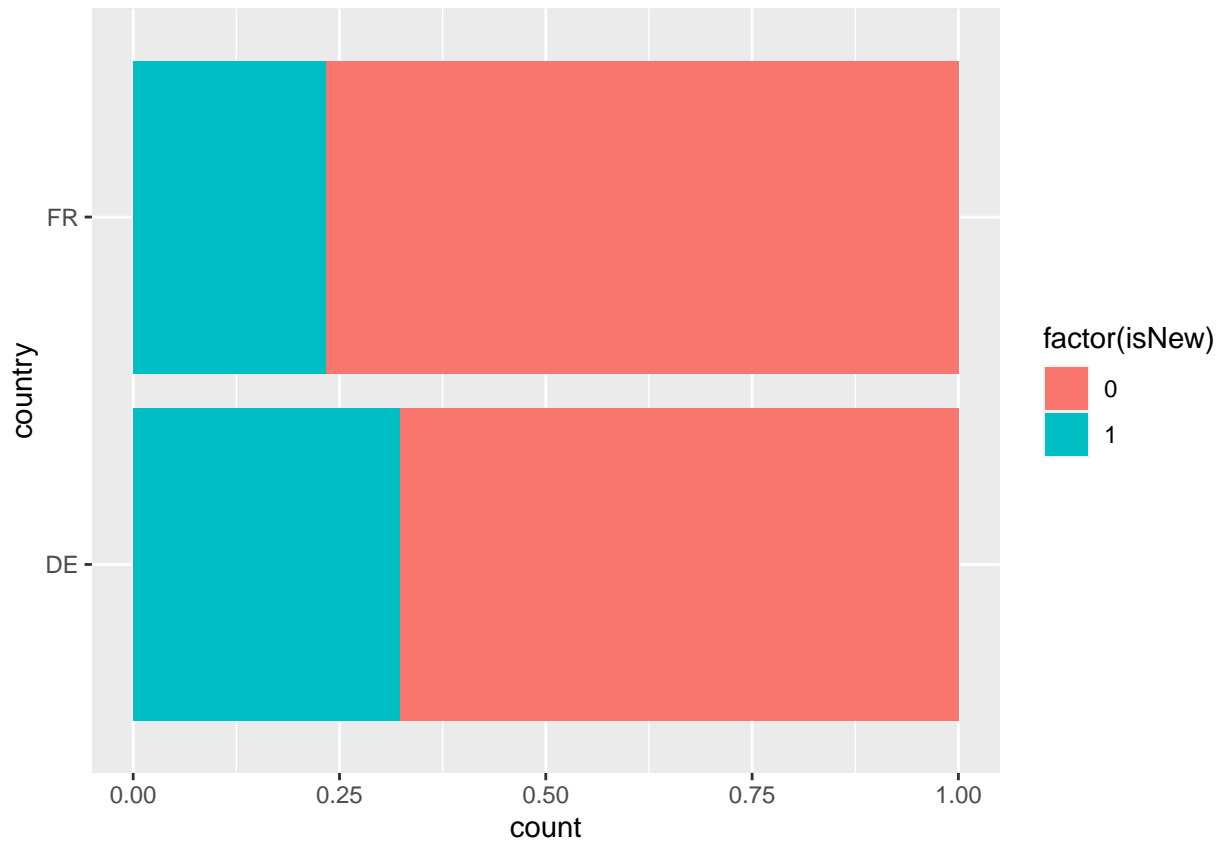
```
lovoo_data %>%  
  filter(country == "DE" | country == "FR") %>%  
ggplot() +  
  aes(x = country, group = factor(isNew), fill = factor(isNew)) +  
  geom_bar(position = "dodge")
```



### 4.3 Koordináta rendszerek

Cseréljük meg az x és y koordinátákat

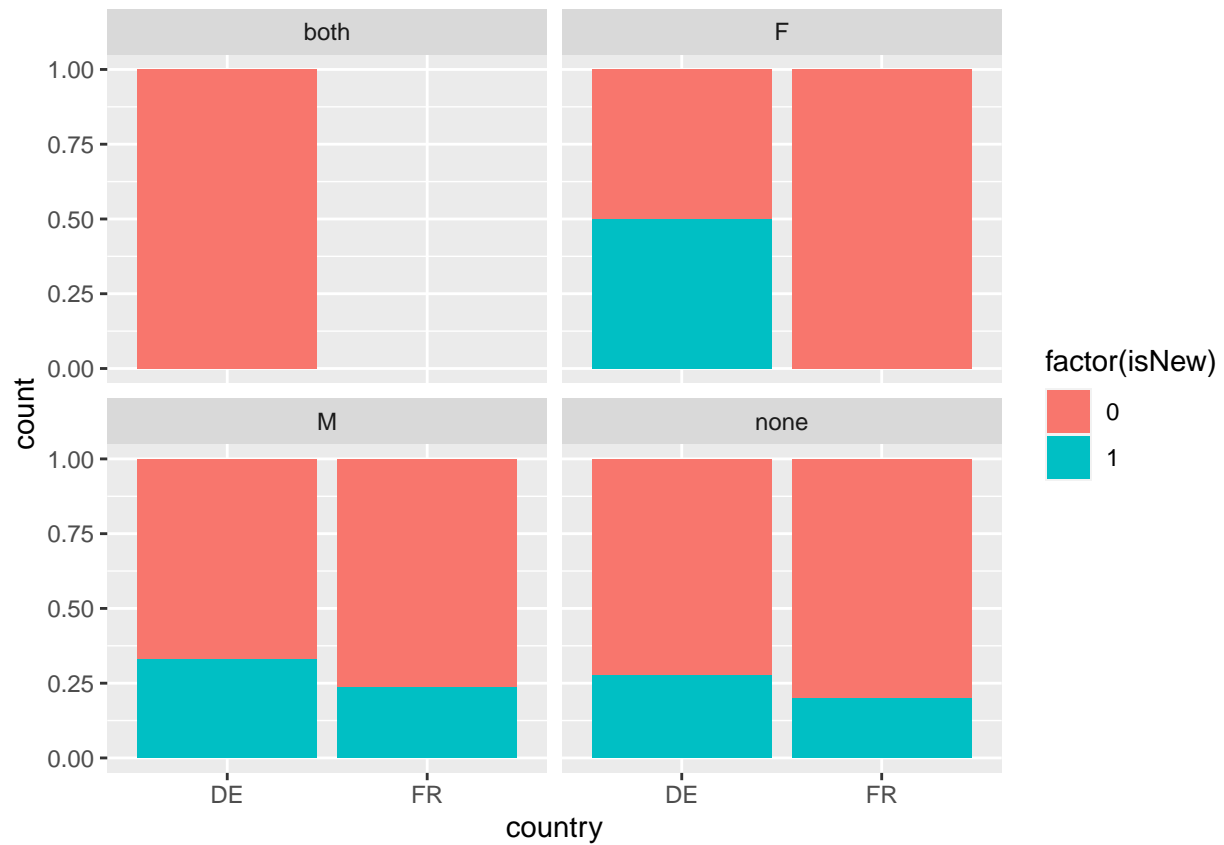
```
lovoo_data %>%  
  filter(country == "DE" | country == "FR") %>%  
  ggplot() +  
    aes(x = country, group = factor(isNew), fill = factor(isNew)) +  
    geom_bar(position = "fill") +  
    coord_flip()
```



#### 4.4 Ábra panelekre osztása (faceting)

A facetelésnél valamilyen adatokban lévő szempont alapján több ábrát vizsgálunk meg egyszerre. Figyelj rá, hogy a faceteléshez felhasznált változó elé “~” jelet kell tenni!

```
lovoo_data %>%
  filter(country == "DE" | country == "FR") %>%
  ggplot() +
    aes(x = country, group = factor(isNew), fill = factor(isNew)) +
    geom_bar(position = "fill") +
    facet_wrap(~genderLooking)
```



Két változót is használhatunk a faceteléshez, az első a sor, a második az oszlop

```
lovoo_data %>%
  filter(country == "DE" | country == "FR") %>%
  ggplot() +
    aes(x = age) +
    geom_density() +
    facet_grid(factor(isNew)~genderLooking)
```

