Data visualization in R

Zoltan Kekecs, Marton Kovacs

February 23, 2020

Contents

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 Ismétlés

Az alábbi gyakorláshoz használd a Tidyverse pakcage-et és az alapvető funkciókat amit a dplyr-ből tanultunk!

Gyakorlás

- Töltsd be a tidyverse csomagot!
- Telepítsd és töltsd be a "gapminder" csomagot!
- Töltsd be a gapminder adattáblát!
- Szűrd meg az adatokat úgy hogy csak a 2007-ből (year) származó adatokkal dolgozzunk, és számold ki az átalagos várható életkort (lifeExp) kontinensenként (continent) ezeken a 2007-es adatokon.
- Nézd meg, hogy hány mérés tartozik az egyes országokhoz (country). (Segítség: Minden mérés egy sor.)
- Hogy könnyebben átlássuk a populációt, hozz létre egy "pop_thousand" nevű változót, amiben a meglévő populáció (pop) értékek el vannak osztva ezerrel. Az adattáblát amiben már ez az új változó is benne van mentsd el egy új objektumba amit "gapminder_with_pop_thousand"-nak nevezz el.

---

3 Adatábrázolás alapjai a ggplot2-vel

A gyakorlat során egy movies nevű adatbázissal fogunk dolgozni ami filmekről szóló adatokat tartalmaz. Az adatok az IMDB és a Rottent Tomato film-review oldalakról származnak. Ezt az adatbázist betölthetjük az alábbi kóddal. A kód lefuttatása után a környezetben (environment) megjelenik a movies adattábla.

```
load(url("https://stat.duke.edu/~mc301/data/movies.Rdata"))
```

Nézzük meg az adattábla alapvető tulajdonságait a megszokott módon.

```
movies

View(movies)

str(movies)
```

Ennek az adatbázisnak az adatait fogjuk ábrázolni. 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) operatárot is használni tudjuk és egyéb dplyer funkciókat.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

A ggplot2 csomag rengeteg funciót tartalmaz. Ezek átlátásához segítséget nyújthat a ggplot cheatsheet. (Több csomaghoz is van ilyen, érdemes rájuk keresni!) https://rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf

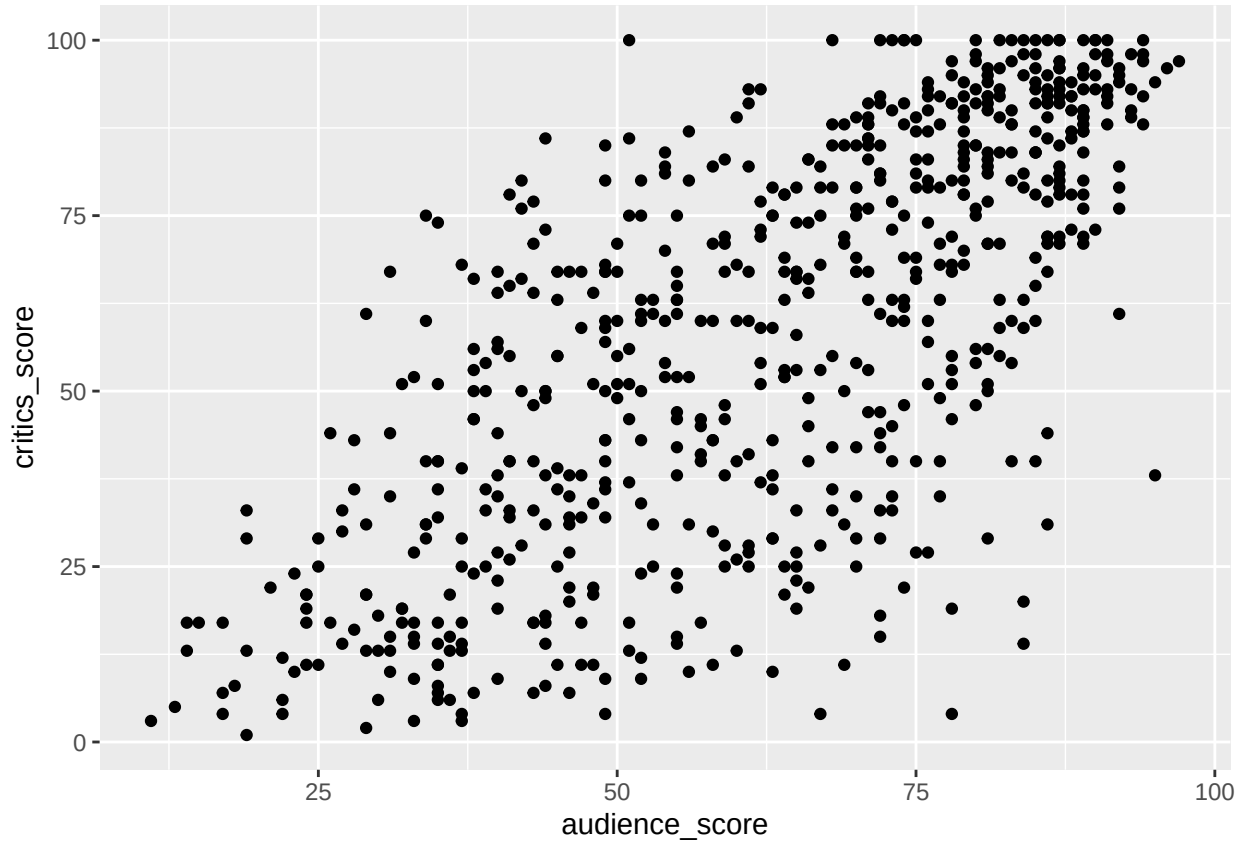
Először is vizualizáljuk mennyire értenek egyet a nézők a kritikusokkal!

Nézd hogyan lehet a **pipe** %>% operátort használni ahhoz, hogy a ggplot funkciót a movies 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 elemben. A **geom_...*** funkciókkal határozzuk meg, **milyen vizualizációs elemek** szerepeljenek az ábrán.

Az hogy **mennyire jár együtt a nézők és a kritikusok véleménye** jól látszik egy **pontdiagramon**, ezért most a **geom_point()** geomot használjuk. ez minden egyes megfigyelést egy pontként ábrázol egy kétdimenziós koordinátarendszerben.

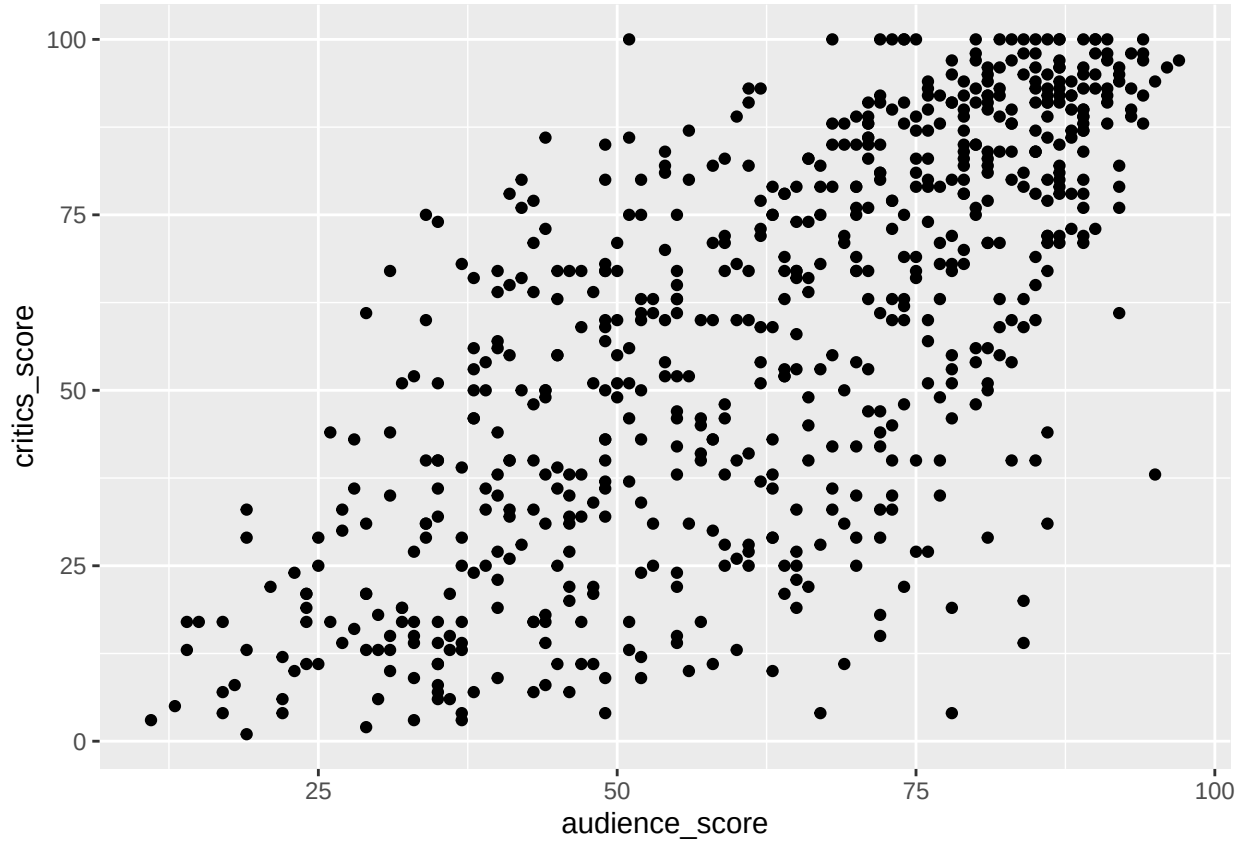
```
movies %>%
  ggplot() +
    aes(x = audience_score,
      y = critics_score) +
    geom_point()
```



Az ábránkat is, mint minden mást R-ben elmenthetünk egy **objektumba**, és amikor újra lefuttatjuk ezt az objektumot, akkor az ábra újra megjelenik.

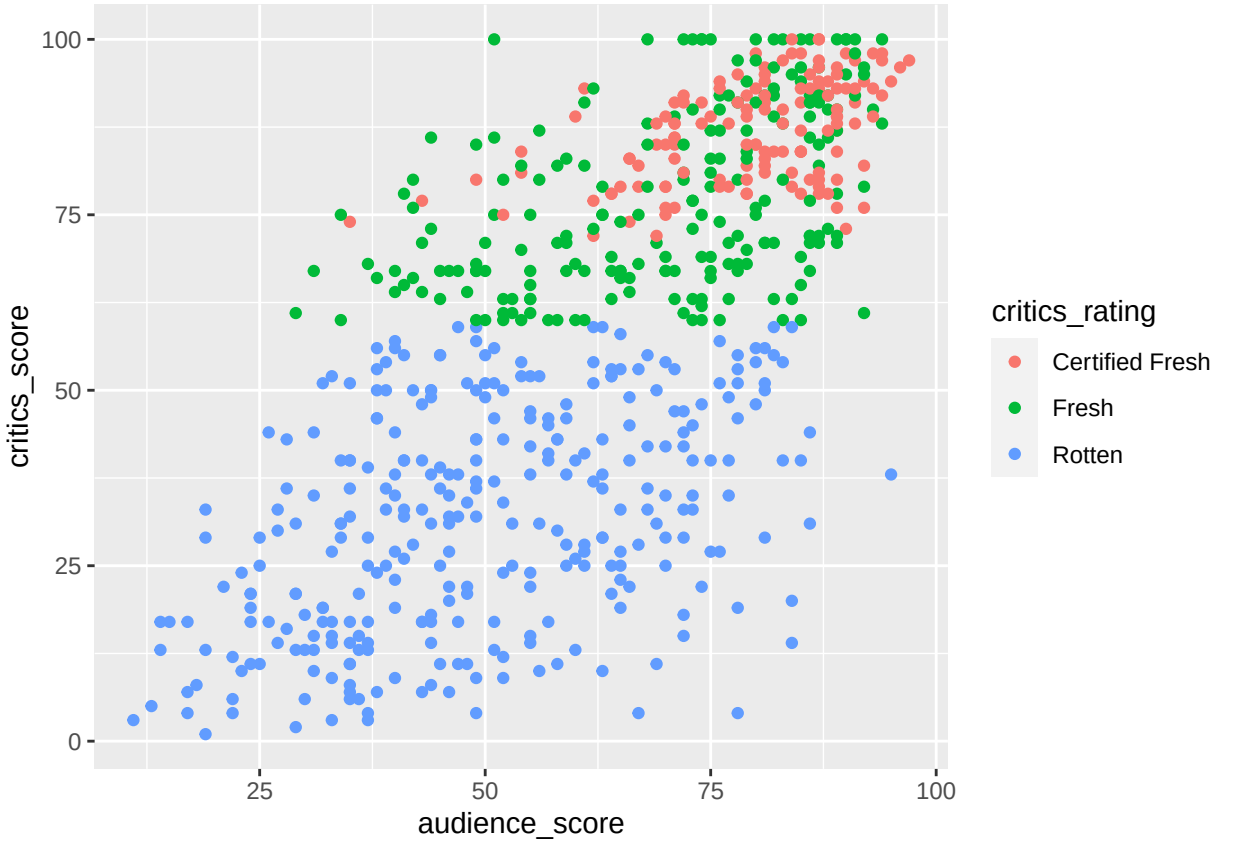
```
plot1 <- movies %>%
          ggplot() +
```

```
            aes(x = audience_score,
                y = critics_score) +
            geom_point()

plot1
```



Az alábbi példán láthatjuk, hogy hogyan tudunk egy **új változót bevonni** a megjelenítésben. Ebben az esetben egy másik **kritikus értékelést** jelenítünk meg színekkel. Mivel a geom_point-ot használjuk, ez a pontok színét fogja befolyásolni, de ha más geomot használnák, azokban is hatna ez a színezésre, hiszen az aes() általános aesthetics részben specifikáltuk.

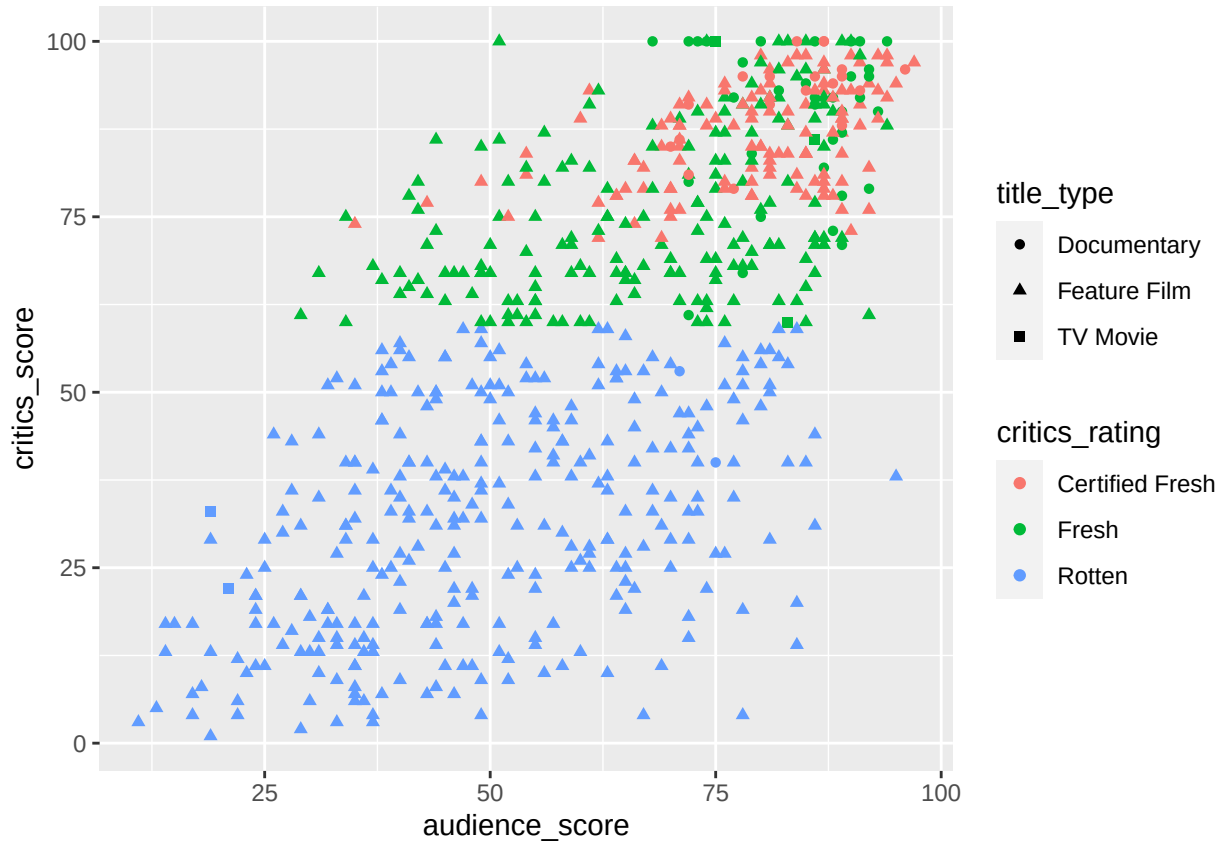
```
plot2 <- movies %>%
          ggplot() +
            aes(x = audience_score,
                y = critics_score,
                color = critics_rating) +
            geom_point()

plot2
```



Nem kell mindig kiírnunk 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 legjobb (100 pont) kritikus értékelást kapott filmek közül számos film dokumentumfilm.

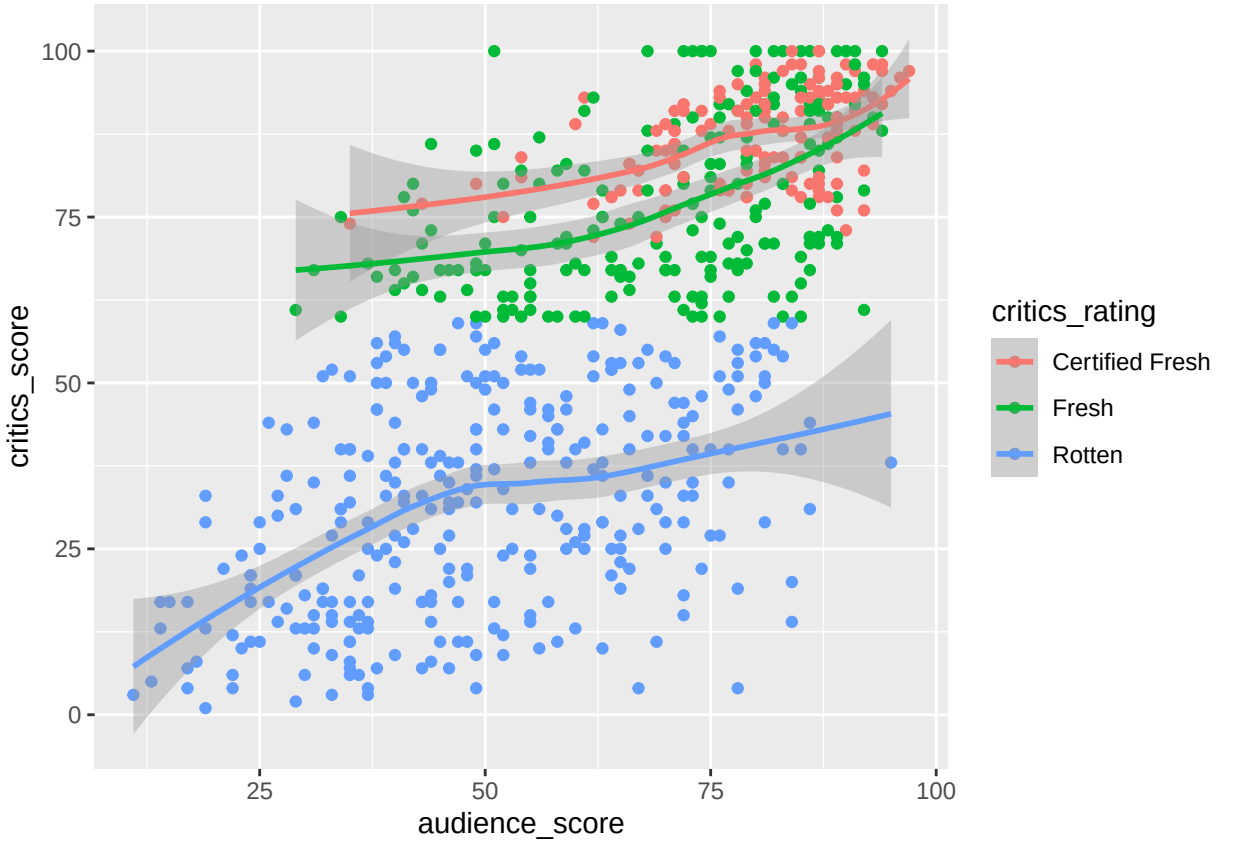
```
plot2 + aes(shape = title_type)
```



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.

```
movies %>%
  ggplot() +
    aes(x = audience_score,
        y = critics_score,
            color = critics_rating) +
        geom_point() +
        geom_smooth()
```

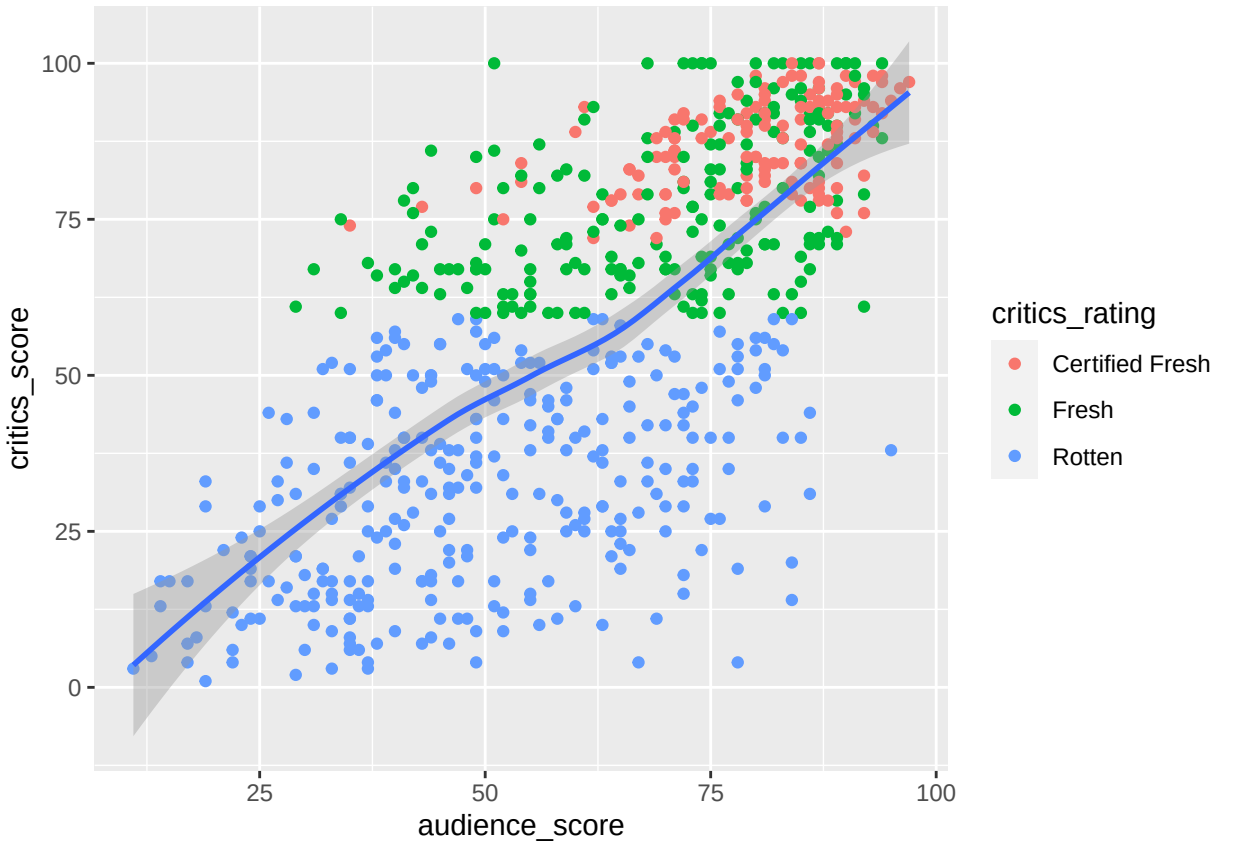
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Mivel az egész ggplot-ra vonatkozó aes() funkció tartalmazza a color = critics_rating részt, ezért ez **minden geomra hat**, így látható hogy a geom_smooth vonalai is a critics_rating csoportinként 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

```
movies %>%
  ggplot() +
    aes(x = audience_score,
        y = critics_score) +
    geom_point(aes(color = critics_rating)) +
    geom_smooth()
```

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



Ha szeretnéd egyes **geomok tulajdonságait konstans értékre állítani** ahelyett hogy az adatok alapján változnánka (pl. szeretnéd egy geom színét átszínezni anélkül hogy ez megfigyelésenként vagy adatcsoportonként változna), az adott paramétert a **geom függvényén belül** kell megadni. Ha használsz 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

```
movies %>%
  ggplot() +
    aes(x = audience_score,
        y = critics_score) +
    geom_point(aes(color = critics_rating), shape = 21, fill = "white") +
    geom_smooth(color = "tomato2")
```

Gyakorlás

Továbbra is használjuk a movies adatbázist.

- Ábrázold az összefüggést az IMDB értékelések (imdb_rating) és a között hogy egy adott filmre hány értékelés jött (imdb_num_votes).
- Alakítsd úgy a fenti ábrát hogy a műfaj (genre) hatása is szerepeljen rajta.

4 Geomok

4.1 Geomok eloszlás vizsgálatára

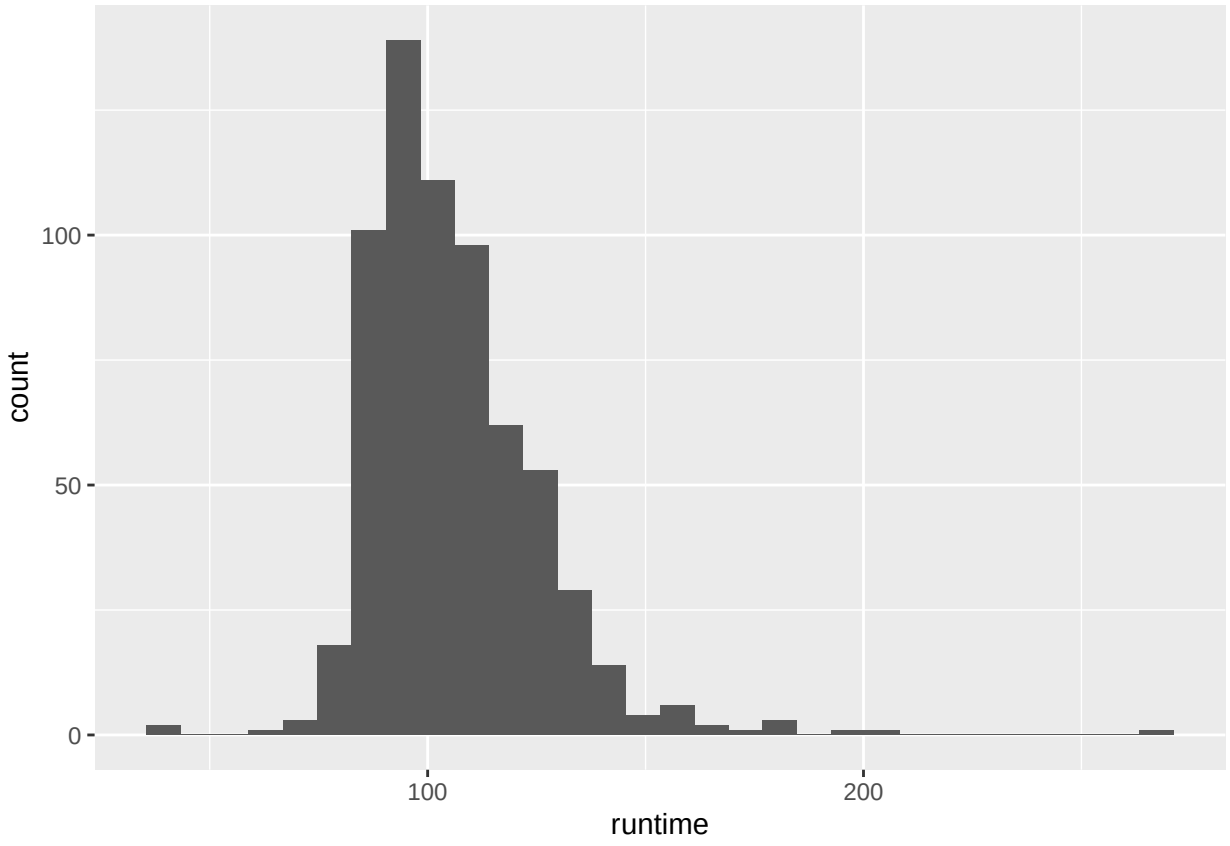
Számos fajta geom van. Alább látható néhány gyakran használt geom amit az adatok eloszlásának vizualizációjára szoktunk használni.

4.1.1 Hisztogramm

```
movies %>%
  ggplot() +
    aes(x = runtime) +
    geom_histogram()
```

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

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```

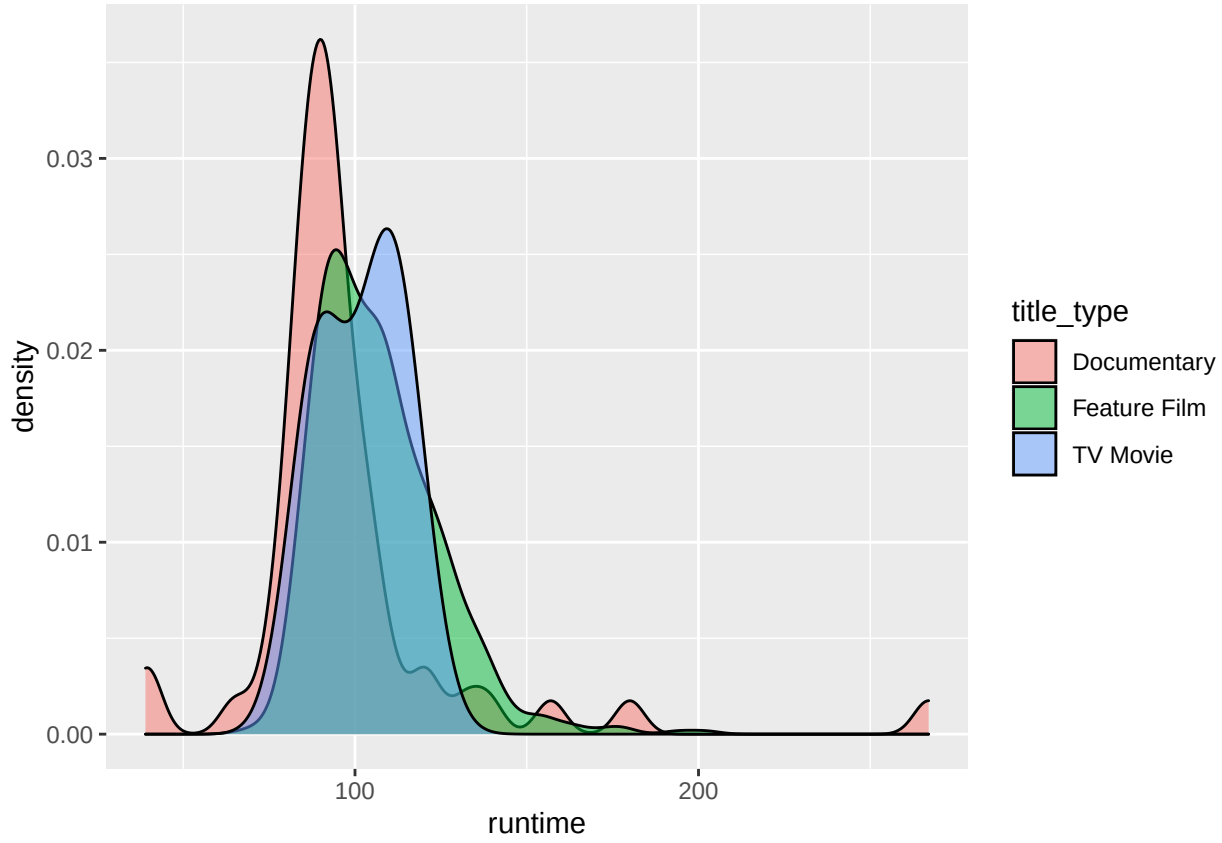


4.1.2 Sűrűségi ábra (density plot)

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

```
movies %>%
  ggplot() +
    aes(x = runtime, fill = title_type) +
    geom_density(alpha = .5)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```

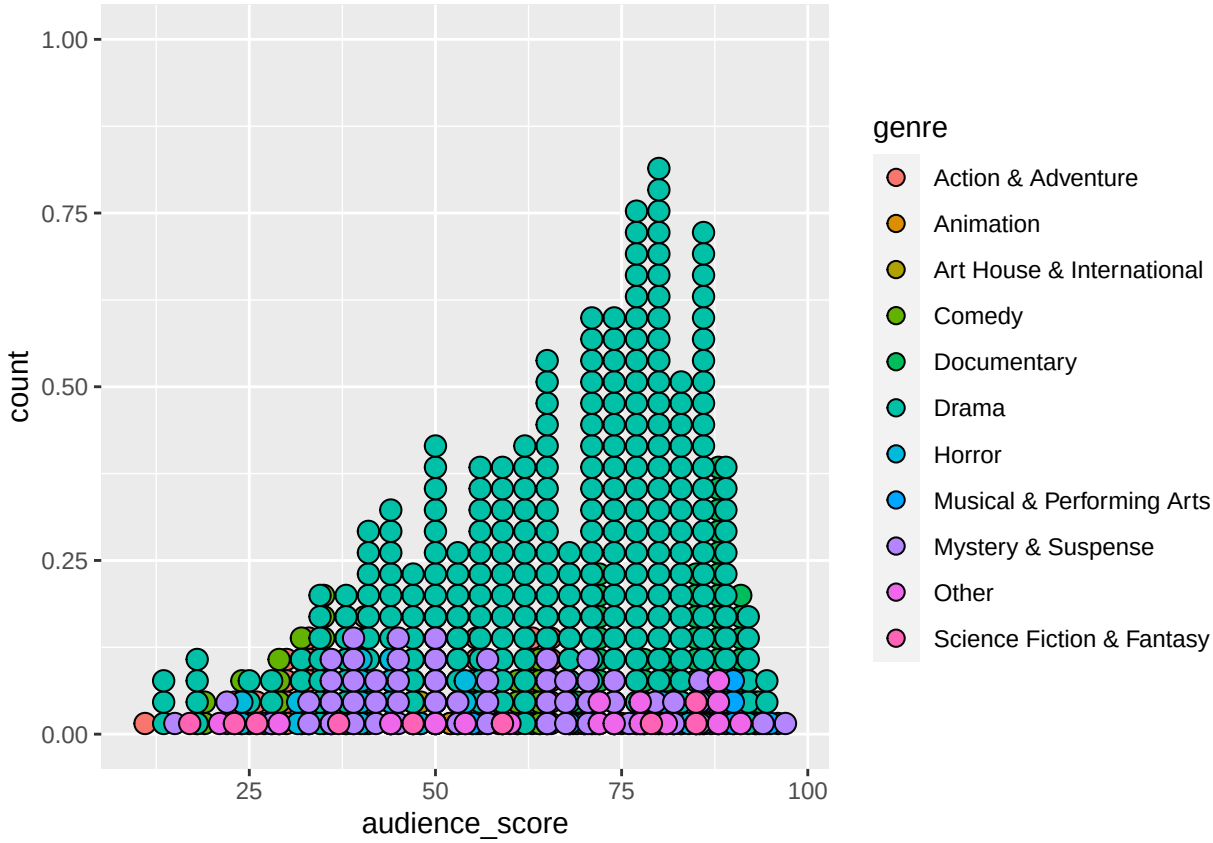


4.1.3 Pöttydiagramm (dotplot)

A sűrűségfüggvény és a hisztogramm egy változata ahol jól látszik a megfigyelések száma is.

```
movies %>%
  ggplot() +
    aes(x = audience_score, fill = genre) +
    geom_dotplot()
```

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



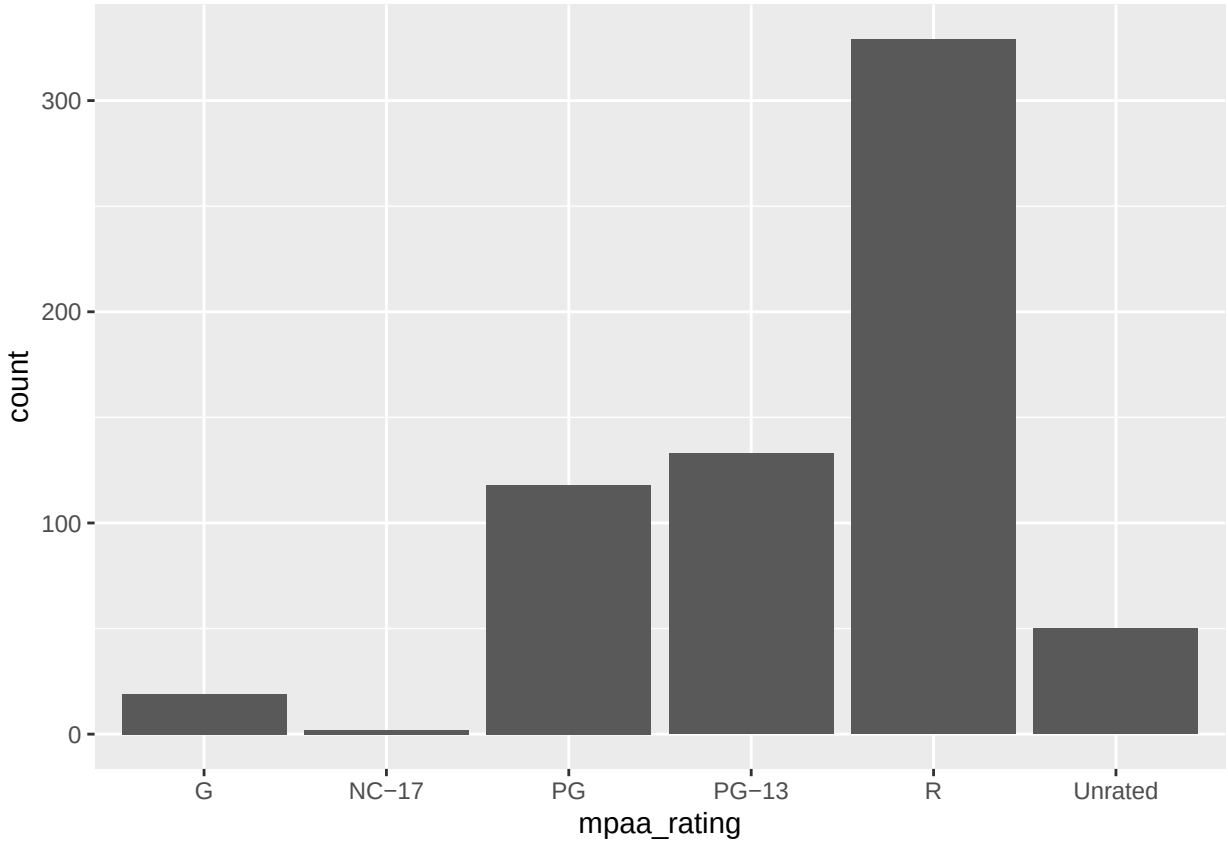
4.1.4 Oszlopdiagram (geom_bar)

Kategorikus változók eloszlását vizsgálhatjuk az oszlopdiagrammal.

Az alábbi ábra megmutatja hogy hogyan oszlanak meg a filmek az adatbázisban a Motion Picture Association of America (MPAA) film rating system szerint.

- G – General Audiences
- PG – Parental Guidance Suggested
- PG-13 – Parents Strongly Cautioned
- R – Restricted (Under 17 requires accompanying parent or adult guardian)
- NC-17 – Adults Only

```
movies %>%
  ggplot() +
    aes(x = mpaa_rating) +
    geom_bar()
```



Gyakorlás

Használjuk a movies adatbázist a következő gyakorlófeladatokhoz.

load(url("https://stat.duke.edu/~mc301/data/movies.Rdata"))

- Hozz létre egy ábrát, melyet egy "my_first_plot" nevű objektumhoz rendelj hozzá. Ezen az ábrán vizsgáld meg kritikusok által adott értékelás (critics_score) eloszlását. Tetszőleges geomot használhatsz. A ggplot2 cheatsheet segíthet kitalálni, melyik a legjobb geom erre a célra.

https://rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf

Tipp: a critics_score 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 hogyan különbözik azoknál a filmeknél amiket jelöltek a legjobb film oszkárdíjra (best_pic_nom) azokhoz képest amelyeket nem. Ehhez használj tetszőleges aes()-t, pl.: color, fill, linetype, size. A ggplot2 cheatsheet segít hogy az általad választott geomnál melyik a releváns aes()

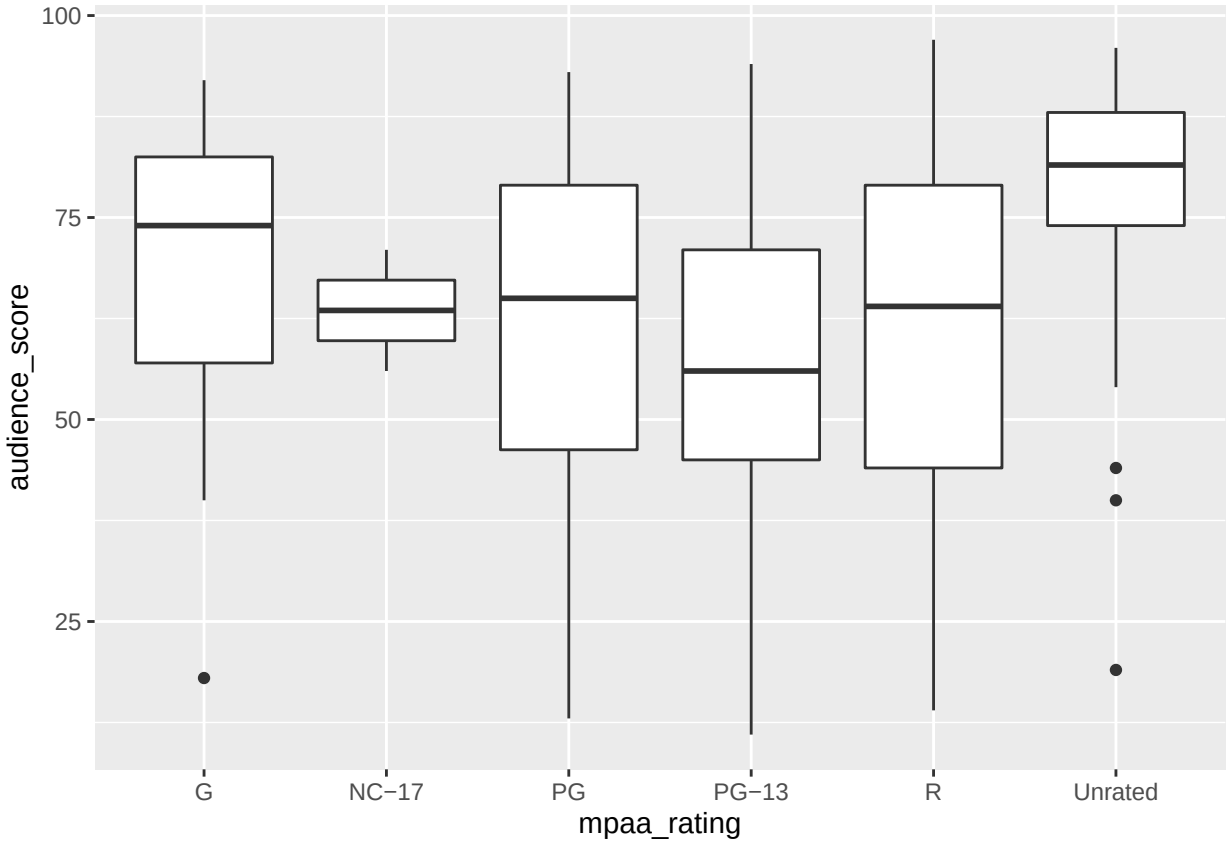
4.2 Geomok két változó kapcsolatának vizsgálatára.

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.

4.2.1 Doboz ábra (boxplot)

Az alábbi doboz ábra (boxplot) a nézői értékelést mutatja a Motion Picture Association of America (MPAA) film rating system kategóriái szerint. Ez az ábra típus a mediánt mutatja középen, és az adatok szóródását körülötte, a kvartilisek szerint felosztva.

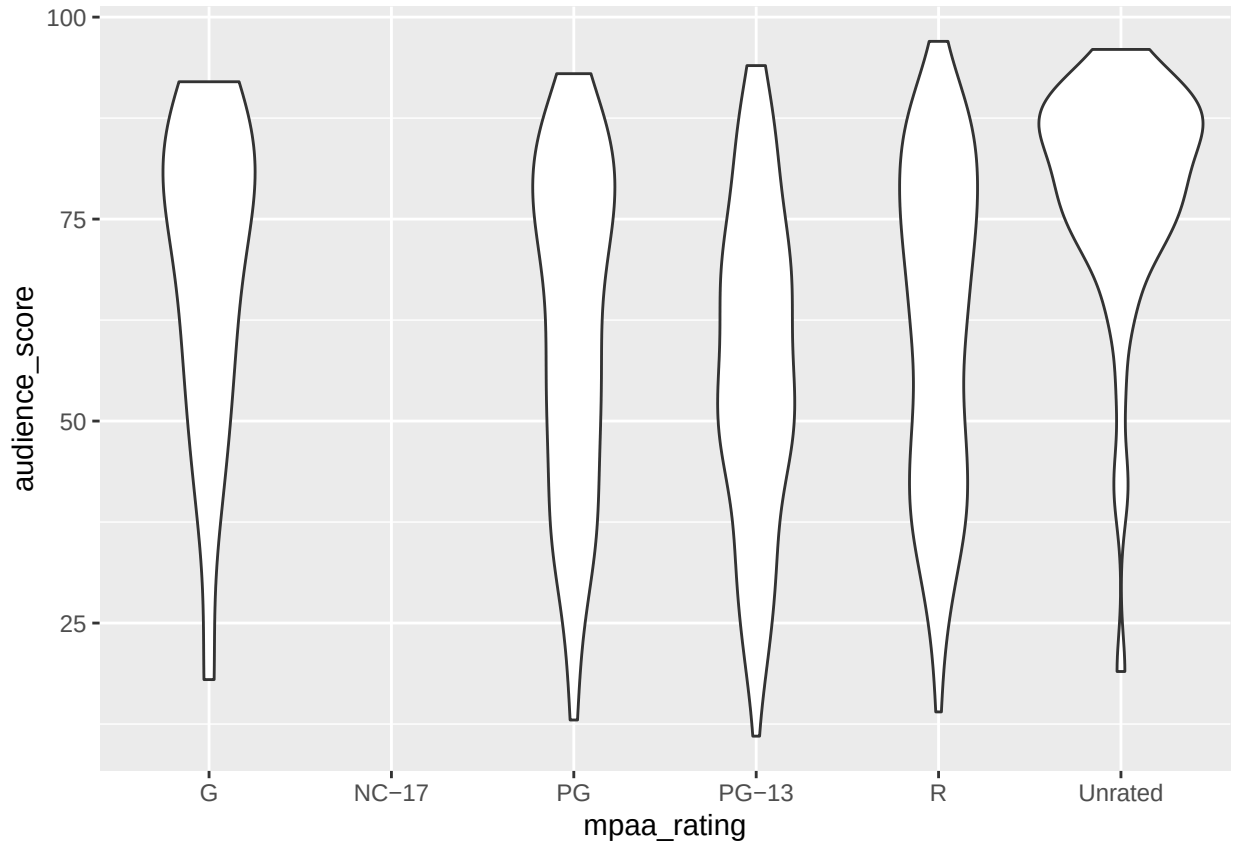
```
movies %>%
  ggplot() +
    aes(y = audience_score, x = mpaa_rating) +
    geom_boxplot()
```



4.2.2 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.

```
movies %>%
  ggplot() +
  aes(y = audience_score, x = mpaa_rating) +
  geom_violin()
```




```
movies %>%
  ggplot() +
  aes(y = audience_score, x = critics_rating) +
  geom_violin()
```

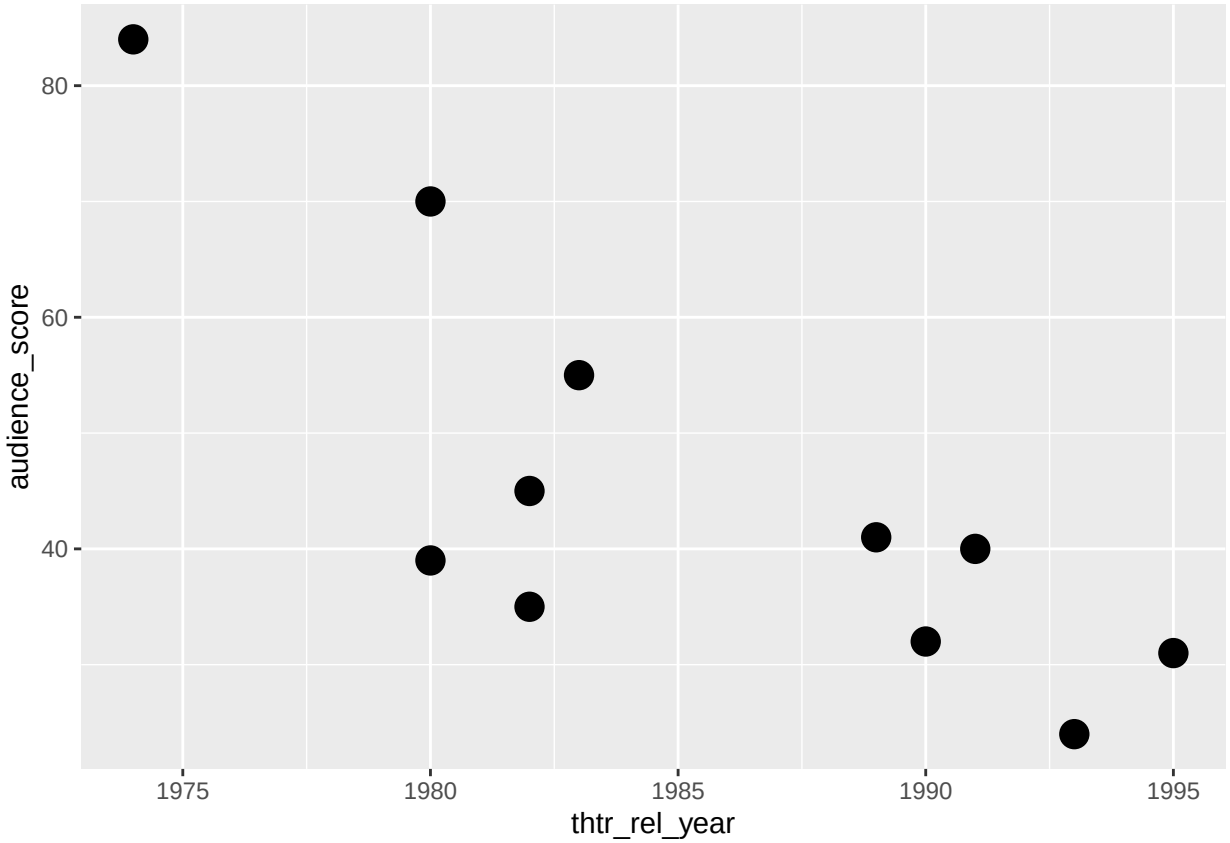
5 Az ábrák testre szabása

5.1 Adatok előkészítése ábra készítésre

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

Például az alábbi kóddal kiválasztjuk azokat a horrorfilmeket, amelyek 1972 és 2002 között jelentek meg, és csak azokat az adatokat ábrázoljuk.

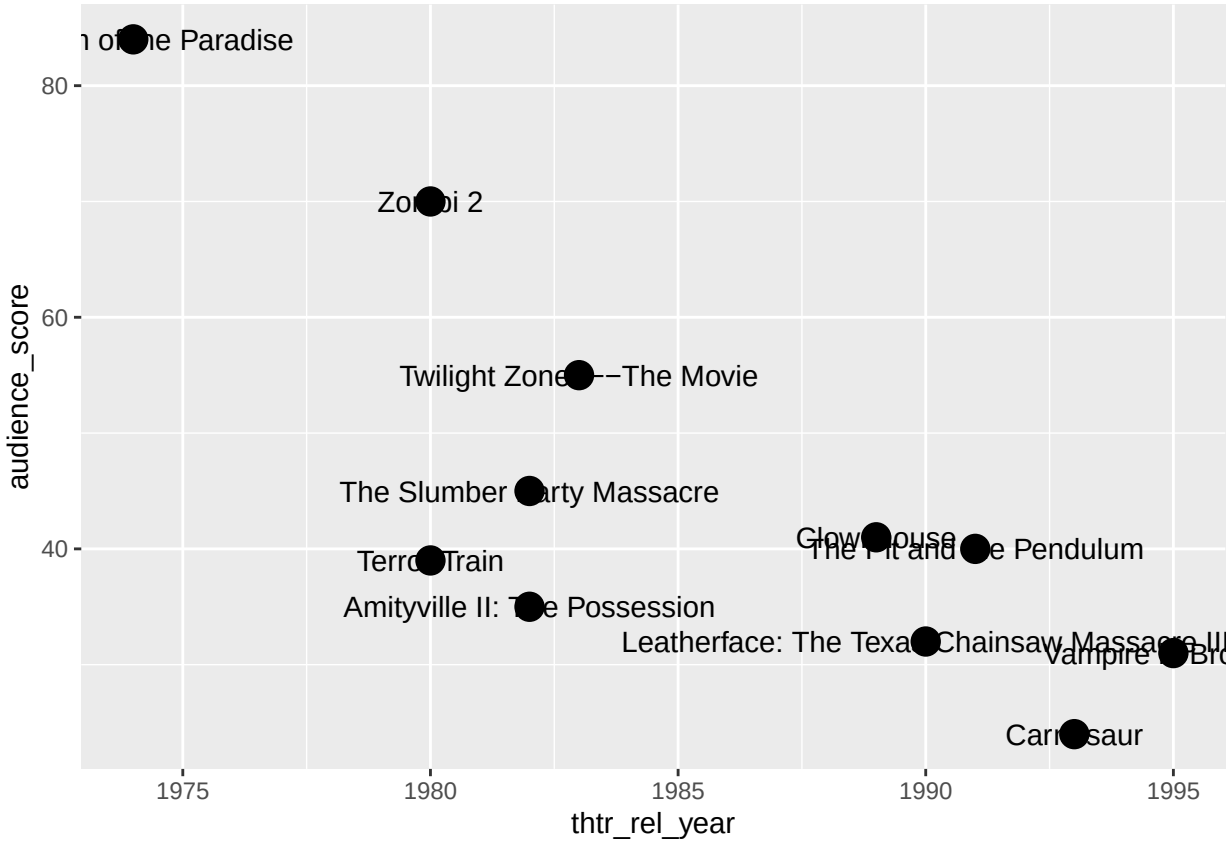
```
movies %>%
  filter(genre == "Horror", thtr_rel_year > 1972, thtr_rel_year < 2002) %>%
  ggplot() +
  aes(x = thtr_rel_year,
      y = audience_score) +
  geom_point(shape=16, fill = "white", size = 5)
```



5.2 Szöveg ábrára rakása

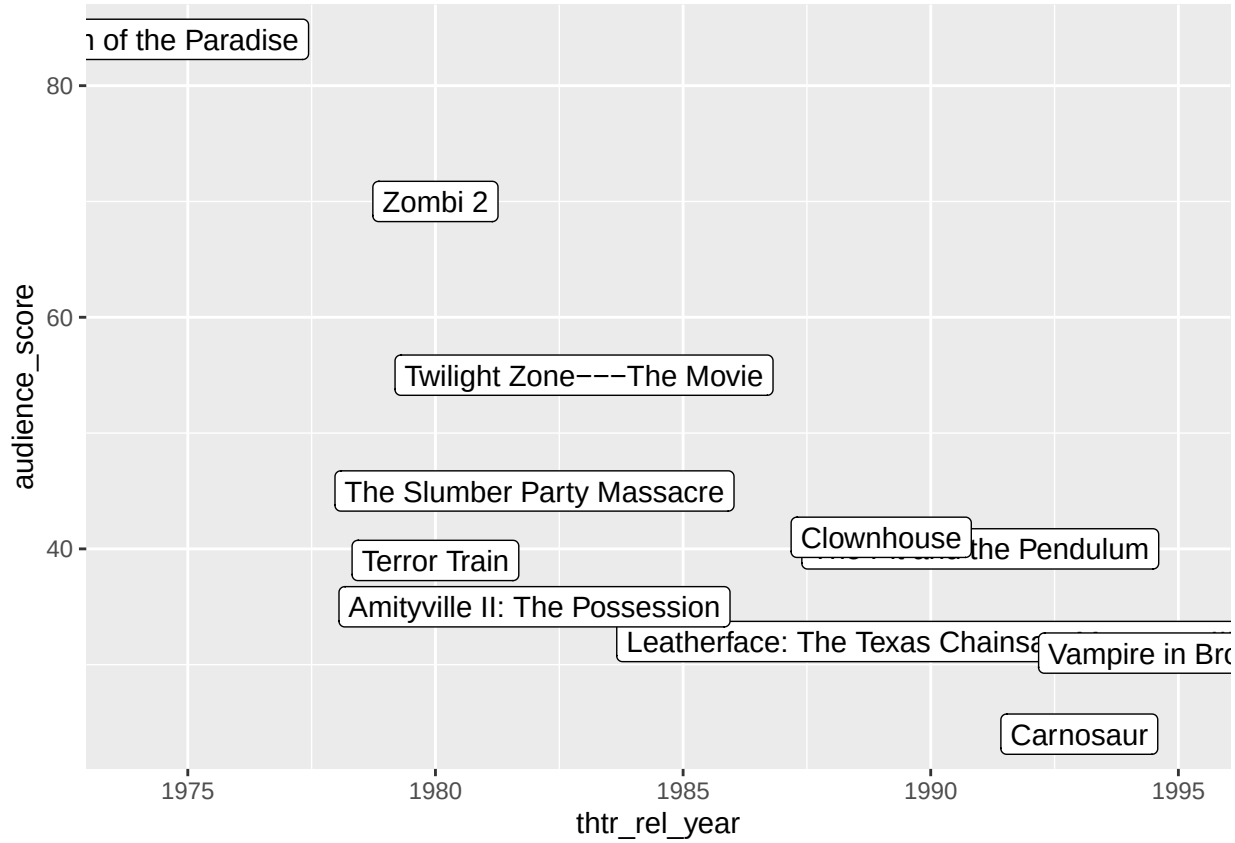
használhatjuk a geom_text-et

```
movies %>%
  filter(genre == "Horror", thtr_rel_year > 1972, thtr_rel_year < 2002) %>%
  ggplot() +
  aes(x = thtr_rel_year,
      y = audience_score,
      label = title) +
  geom_point(shape=16, fill = "white", size = 5) +
  geom_text()
```



vagy a geom_label-t.

```
movies %>%
  filter(genre == "Horror", thtr_rel_year > 1972, thtr_rel_year < 2002) %>%
  ggplot() +
  aes(x = thtr_rel_year,
      y = audience_score,
      label = title) +
  geom_point(shape=16, fill = "white", size = 5) +
  geom_label()
```



Gyakorlás

Továbbra is használjuk a movies adatbázist.

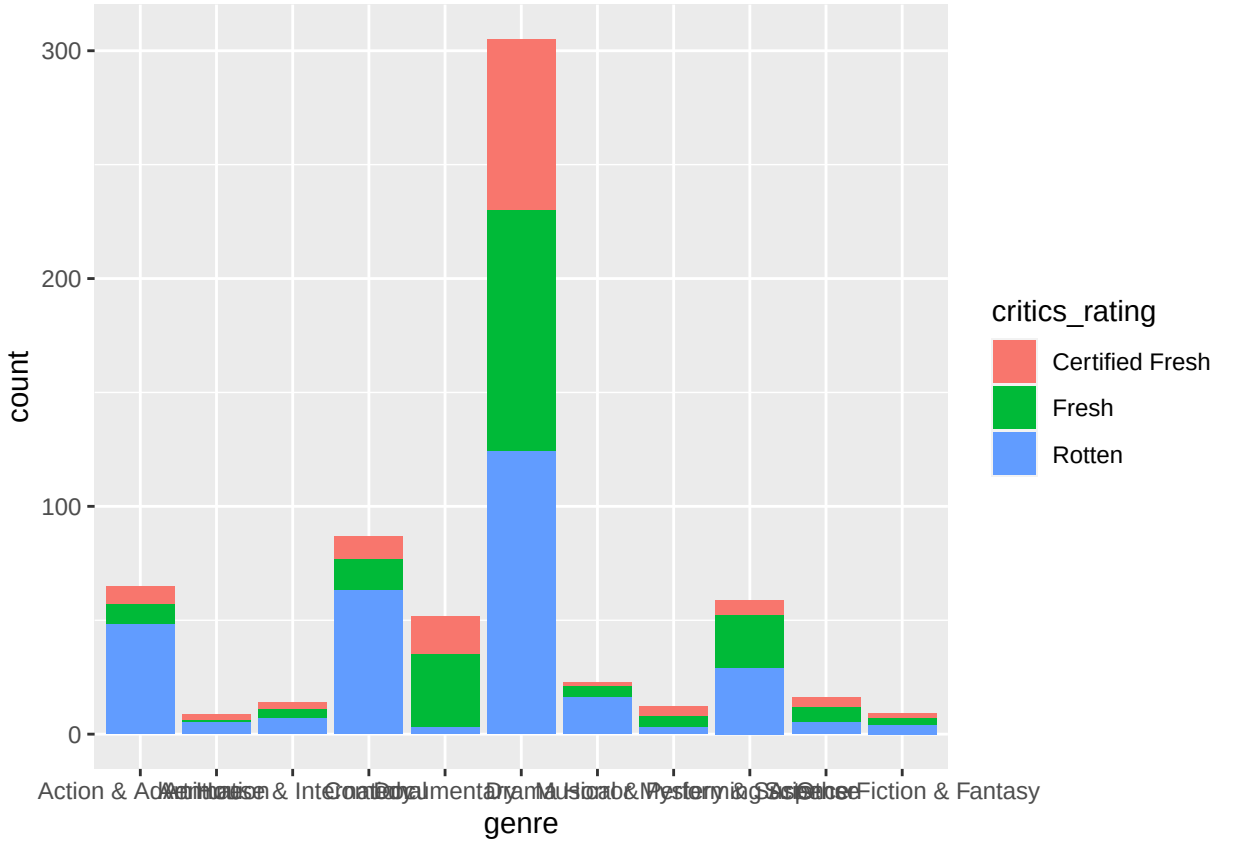
- Ábrázold a nézői értékelés (audience_score) és a kritikusi értékelés (critics_score) kapcsolatát egy pontdiagrammal úgy, hogy csak az 1995-ben megjelent filmek szerepeljenek az ábrán.
- Tedd rá a filmek címét az ábrára feliratként, hogy minden ponton szerepeljen a film címe.

5.3 Pozíció (Position)

5.3.1 oszlopdiagram (geom_bar)

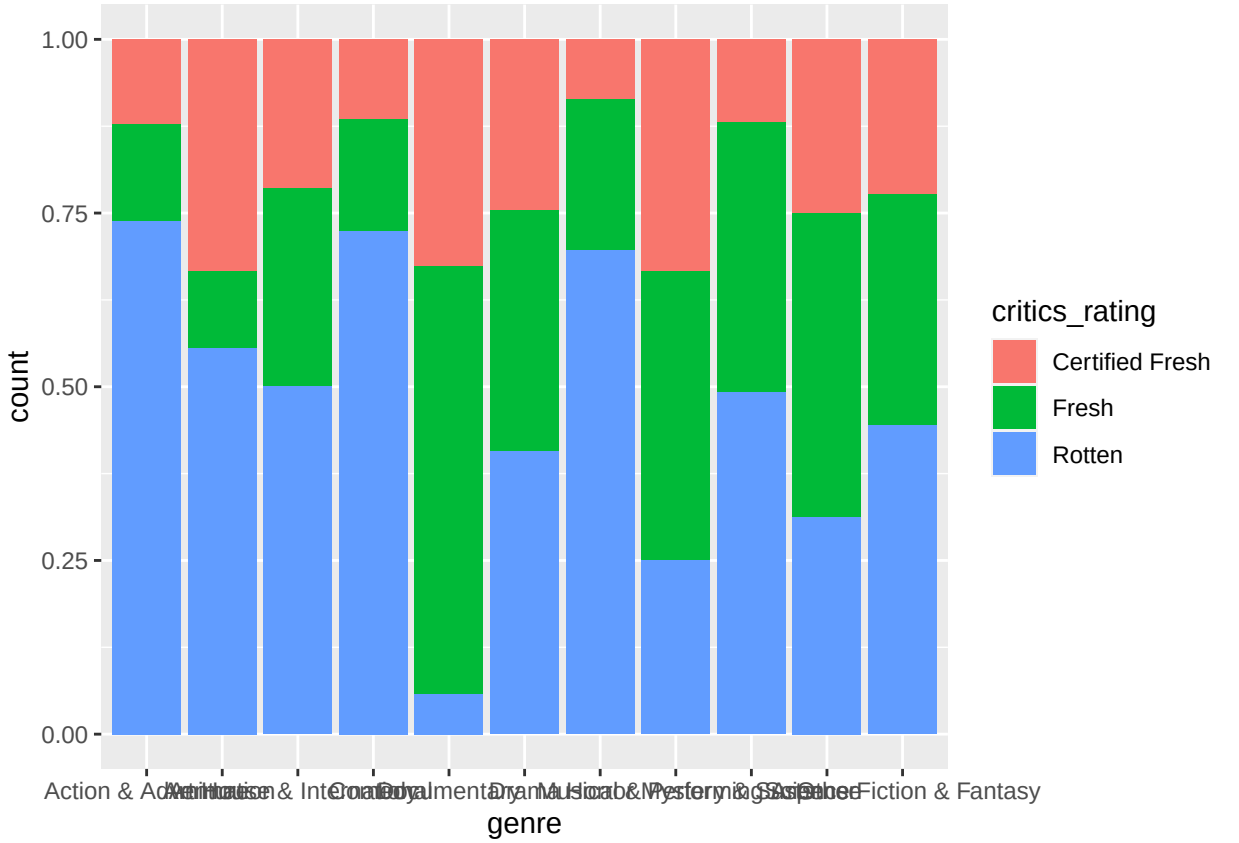
Hozzunk létre egy halmozott barplotot (stacked bar), ami a mennyiséget mutatja

```
ggplot(data = movies) +
  aes(x = genre, group = critics_rating, fill = critics_rating) +
  geom_bar(position = "stack")
```



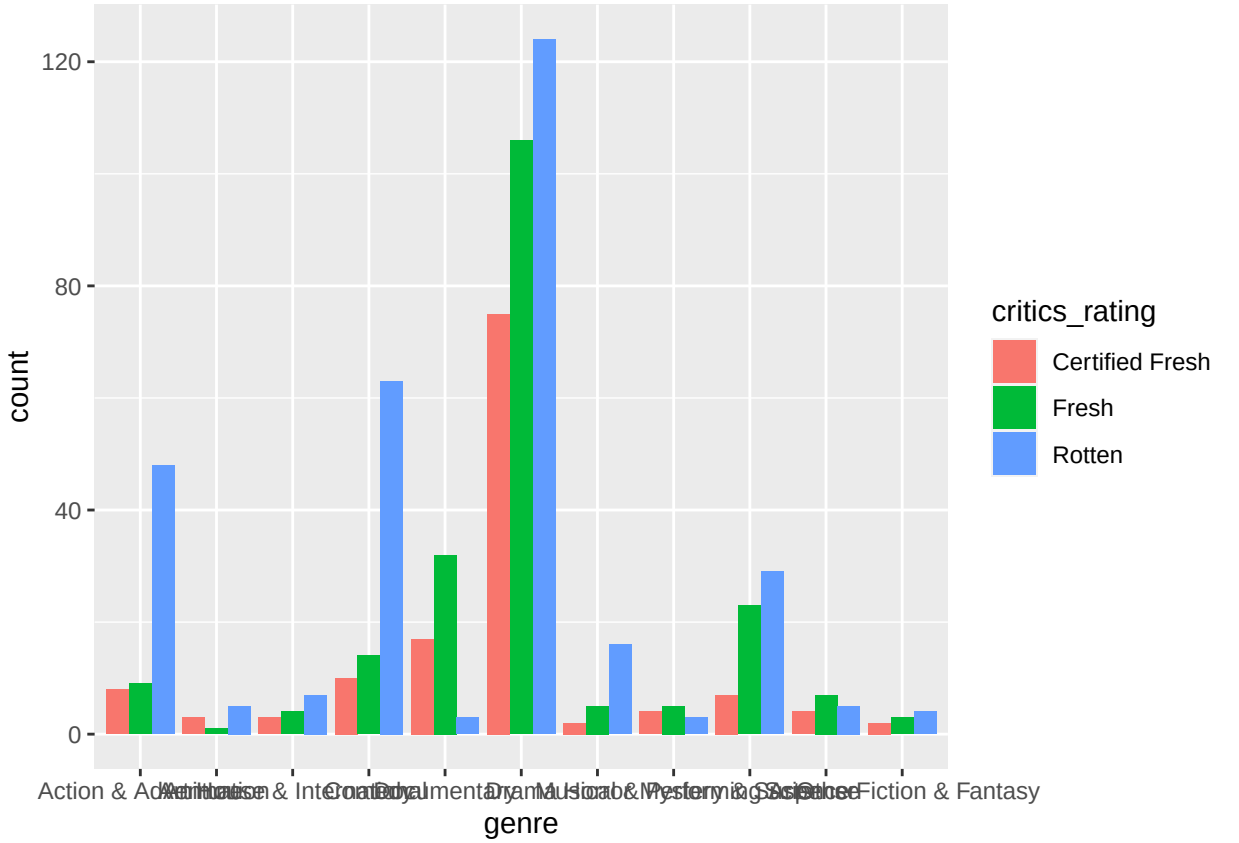
Arányként (proportion) is megmutathatjuk a csoportok mennyisége közti összefüggést, ha aa “stack” helyett a “fill” position-t adjuk meg.

```
ggplot(data = movies) +
  aes(x = genre, group = critics_rating, fill = critics_rating) +
  geom_bar(position = "fill")
```



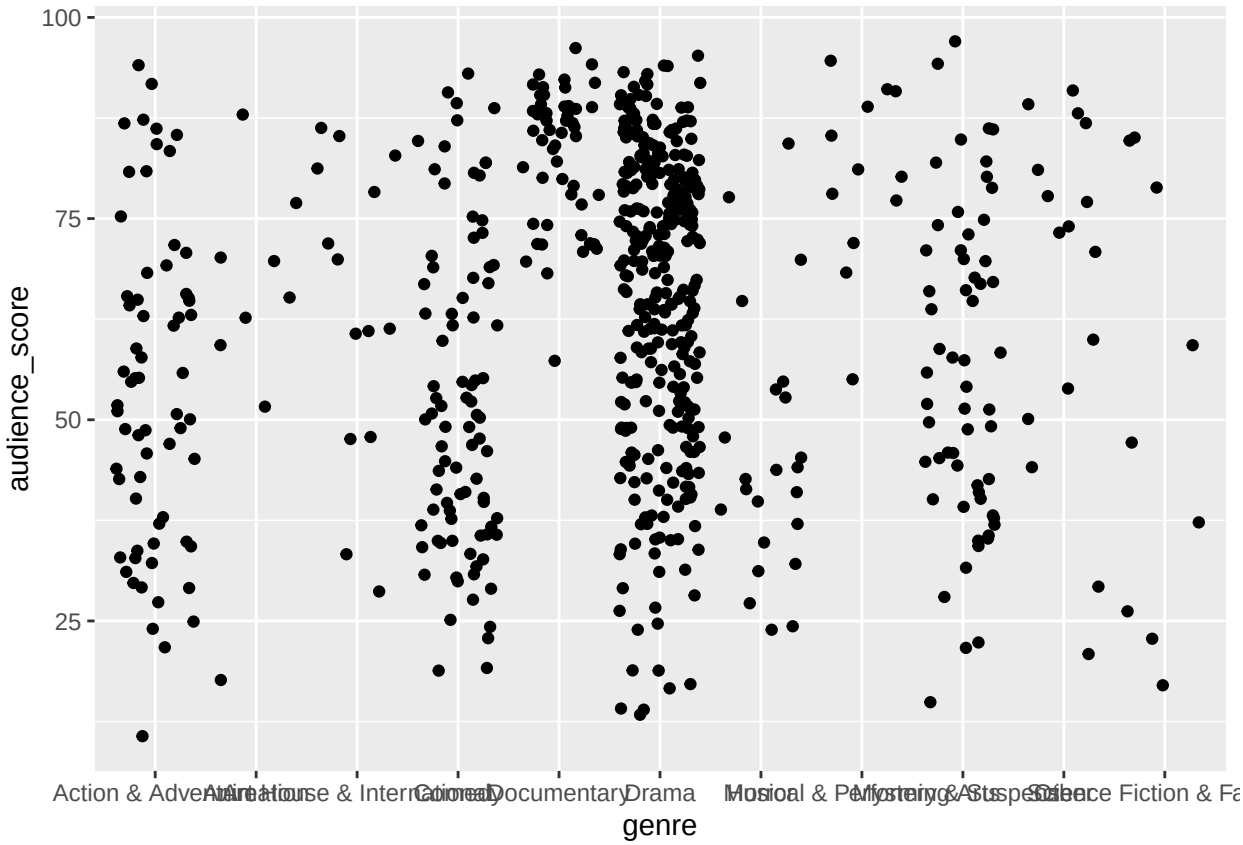
Vagy ábrázoljuk egymás mellett a mennyiséget, hogy könnyebben összehasonlíthatók legyenek a csoportok. Ezt a "dodge" beállítással érhetjük el a position paraméterben.

```
ggplot(data = movies) +
  aes(x = genre, group = critics_rating, fill = critics_rating) +
  geom_bar(position = "dodge")
```



A szétszórás "jitter" pozíció segítségével random zajt adhatunk az adatokhoz, így az átfedéseket megszűntetve 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.

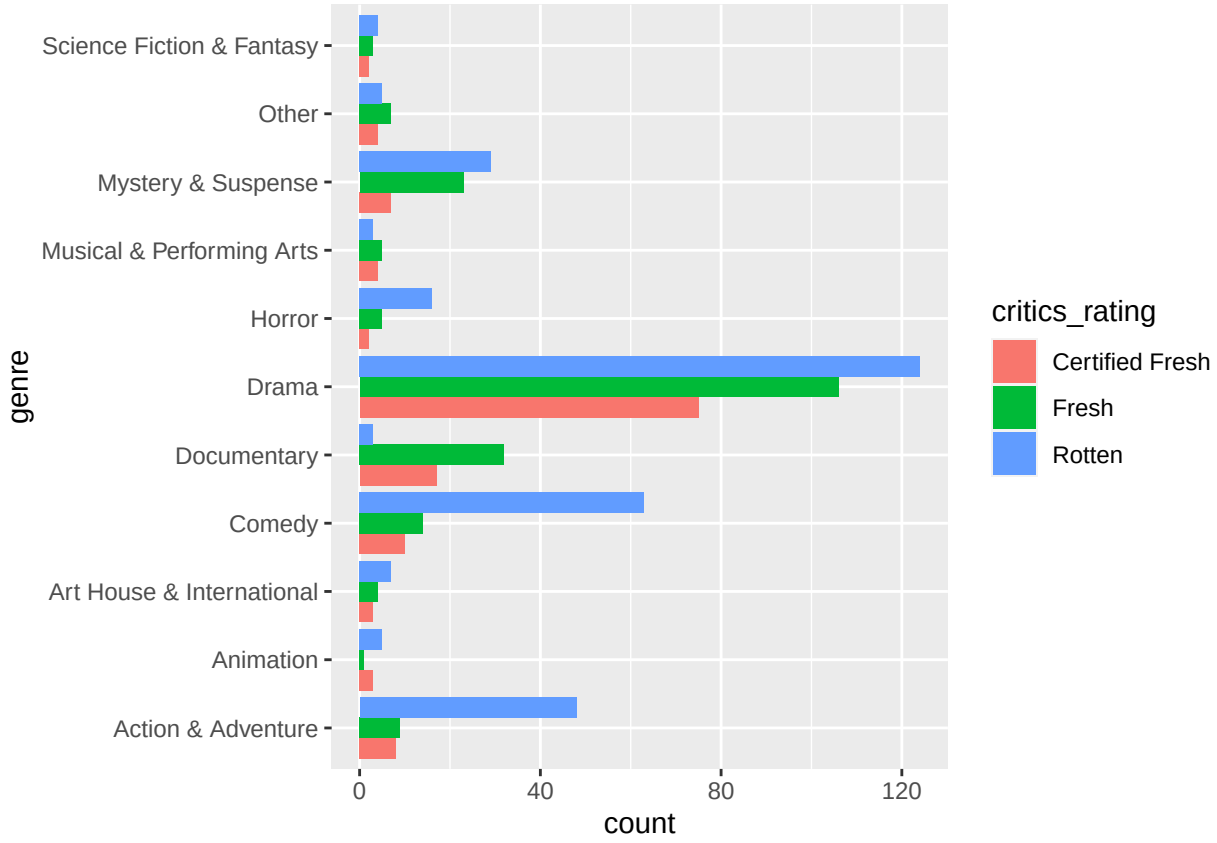
```
ggplot(data = movies) +
  aes(x = genre, y = audience_score) +
  geom_jitter()
```



5.4 Koordináta rendszerek

Cseréljük meg az x és y koordinátákat

```
ggplot(data = movies) +
  aes(x = genre, group = critics_rating, fill = critics_rating) +
  geom_bar(position = "dodge") +
  coord_flip()
```



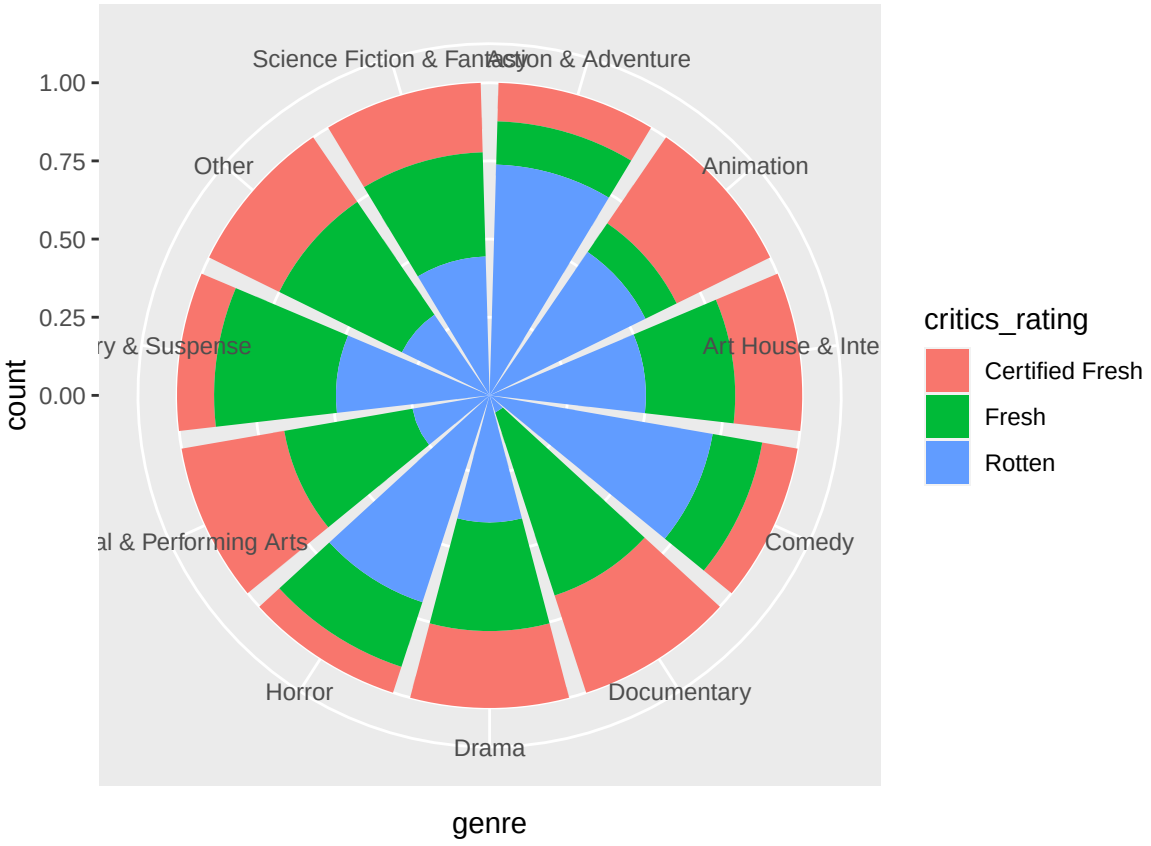
Gyakorlás

- Most ábrázoljuk csak a legnagyobb bevételt behozó filmeket (top200_box == "yes").
- Nézzük meg, hogy melyik film milyen imdb pontot kapott (imdb_rating). A filmek címe (title) szerepeljen az egyik tengelyen és legyen olvasható.

5.5 Polar ábra

A polar ábra a pie chart és a proportion oszlopdiagram kombinációjaként fogható fel.

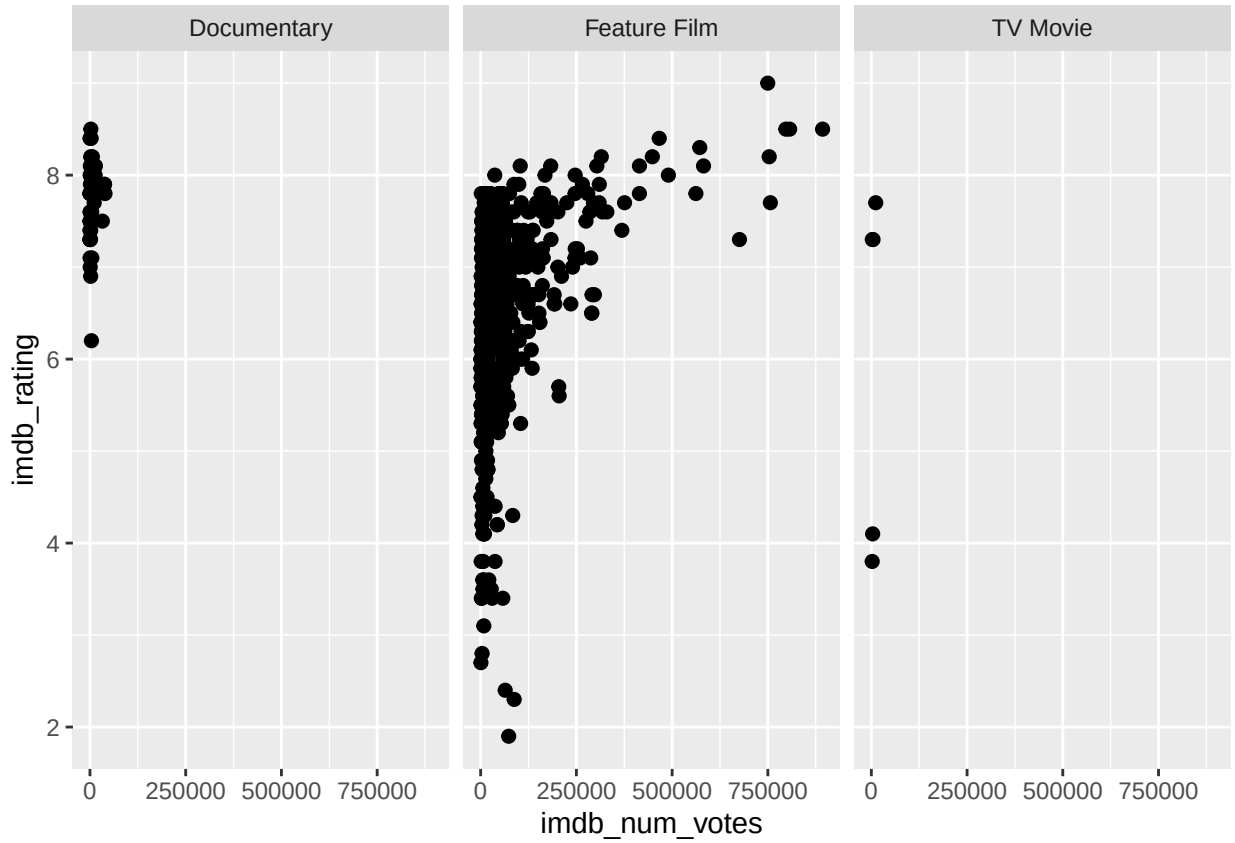
```
ggplot(data = movies) +
  aes(x = genre, group = critics_rating, fill = critics_rating) +
  geom_bar(position = "fill") +
  coord_polar()
```



5.6 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!

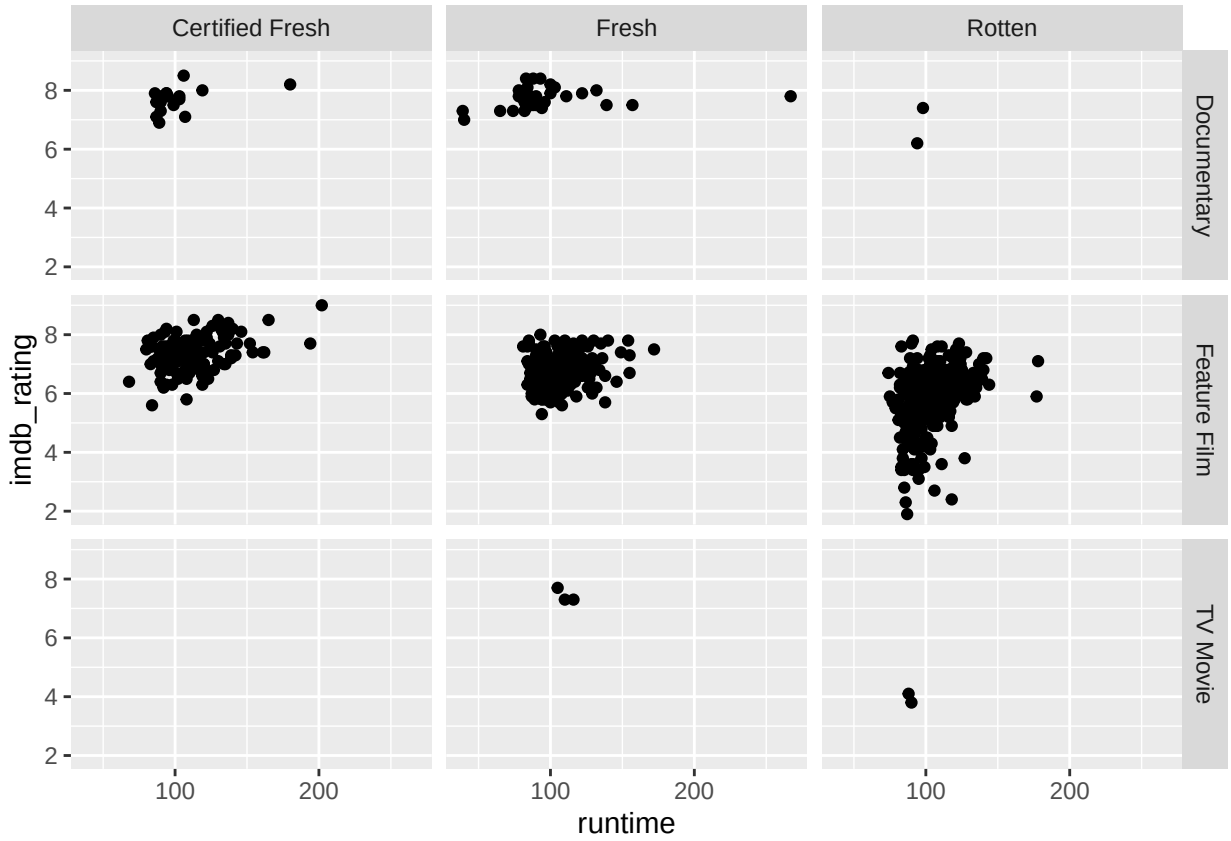
```
ggplot(data = movies) +
  aes(y = imdb_rating, x = imdb_num_votes) +
  geom_point(size = 2) +
  facet_wrap(~title_type)
```



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

```
ggplot(data = movies) +
  aes(y = imdb_rating, x = runtime) +
  geom_point() +
  facet_grid(title_type ~ critics_rating)
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```


Certified Fresh
Fresh
Rotten
Documentary
Feature Film
TV Movie
imdb_rating
runtime
2
4
6
8
100
200