PSZB17-210 - Seminar_4

Zoltan Kekecs

Sept 28, 2021

4. Ora - Adatexploracio

Az ora celja az adatexploracios modszerek elsajatitasa.

Package-ek betoltese

A kovetkezo 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 a WHO altal legutobb feltoltott COVID-19 adatokat a read_csv() funkcioval, es elmentjuk egy COVID_data nevu objektumba. A **read_csv()** funkcio a tidyverse resze, es egybol tibble formatumban menti el az adatainkat.

```
COVID_data_raw <- read_csv("https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owi
```

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 ...)

```
COVID_data_raw
```

```
## # A tibble: 344,861 x 67
##    iso_code continent location    date       total_cases new_cases
##    <chr>    <chr>     <chr>       <date>           <dbl>     <dbl>
##  1 AFG      Asia      Afghanistan 2020-01-03          NA         0
##  2 AFG      Asia      Afghanistan 2020-01-04          NA         0
##  3 AFG      Asia      Afghanistan 2020-01-05          NA         0
```

```
##  4 AFG      Asia      Afghanistan 2020-01-06          NA         0
##  5 AFG      Asia      Afghanistan 2020-01-07          NA         0
##  6 AFG      Asia      Afghanistan 2020-01-08          NA         0
##  7 AFG      Asia      Afghanistan 2020-01-09          NA         0
##  8 AFG      Asia      Afghanistan 2020-01-10          NA         0
##  9 AFG      Asia      Afghanistan 2020-01-11          NA         0
## 10 AFG      Asia      Afghanistan 2020-01-12          NA         0
## # i 344,851 more rows
## # i 61 more variables: new_cases_smoothed <dbl>, total_deaths <dbl>,
## #   new_deaths <dbl>, new_deaths_smoothed <dbl>, total_cases_per_million <dbl>,
## #   new_cases_per_million <dbl>, new_cases_smoothed_per_million <dbl>,
## #   total_deaths_per_million <dbl>, new_deaths_per_million <dbl>,
## #   new_deaths_smoothed_per_million <dbl>, reproduction_rate <dbl>,
## #   icu_patients <dbl>, icu_patients_per_million <dbl>, ...
```

Leiro statisztikak

Ha az egyes valtozok **leiro statisztikaira** (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)

```
COVID_data_raw %>%
  select(total_cases) %>%
  summary()
```

```
##   total_cases
##  Min.   :        1
##  1st Qu.:     7910
##  Median :    68890
##  Mean   :  6571274
##  3rd Qu.:   726902
##  Max.   :770874669
##  NA's   :37897
```

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.

```
COVID_data_raw %>%
  summary()
```

Az exploracio megmutatta hogy van nehany irrealisztikus adat. Ennek az az oka hogy kontinensekre es regiokra lebontott osszefoglalo adatokat is tartalmaz a tablazat. Ezeket ugy tudjuk legkonnyebben kivenni hogy kivesszuk azokat a sorokat, ahol a continent valtozo NA erteket vesz fel. (Vedd eszre hogy ezt "!" es az is.na() funkciok kombinaciojaval oldjuk meg. A ! jelentese "NOT".)

```
COVID_data <- COVID_data_raw %>%
  filter(!is.na(continent))
```

```
COVID_data %>%
  select(total_cases) %>%
  summary()
```

```
##   total_cases
##  Min.   :        1
##  1st Qu.:     7267
##  Median :    57601
##  Mean   :  1636940
##  3rd Qu.:   532912
##  Max.   :103436829
##  NA's   :37639
```

```
COVID_data_raw %>%
  select(total_cases) %>%
  summary()
```

```
##   total_cases
##  Min.   :        1
##  1st Qu.:     7910
##  Median :    68890
##  Mean   :  6571274
##  3rd Qu.:   726902
##  Max.   :770874669
##  NA's   :37897
```

Gyakorlas

- Hany regisztralt eset volt osszesen Magyarorszagon a tegnapi napig (*total_cases*)?
- Mi volt a legmagasabb uj eset-szam Magyarorszagon (*new_cases*)?

Megtobb leiro statisztika

A **Psych** package segitsegevel a **describe()** funkcio megtobb hasznos informaciot adhat. Ez a funkcio elsosorban szam-valtozok leirasara szolgal, es karakter tipusu kategorikus valtozok eseten sok warning message-et ad, ezert erdemes a funciot csak a szam-valtozokra lefuttatni (ezt alabb a select() funkcioval erem el.)

```
COVID_data %>%
  select(-date, -iso_code, -continent, -location, -contains("tests"), -positive_rate) %>%
  describe()
```

```
##                                 vars      n        mean           sd
## total_cases                        1 290814  1636939.74   7066721.87
## new_cases                          2 318970     2417.02     39470.48
## new_cases_smoothed                 3 317771     2426.01     35519.03
```

```
## total_deaths                                    4 269503    21653.50     82400.36
## new_deaths                                      5 319010       21.84       149.79
## new_deaths_smoothed                             6 317840       21.91       127.84
## total_cases_per_million                         7 290814   101234.13    151673.33
## new_cases_per_million                           8 318970      148.29      1196.88
## new_cases_smoothed_per_million                  9 317771      148.85       616.64
## total_deaths_per_million                       10 269503      862.90      1101.30
## new_deaths_per_million                         11 319010        0.91         5.25
## new_deaths_smoothed_per_million                12 317840        0.92         2.84
## reproduction_rate                              13 183741        0.91         0.40
## icu_patients                                   14  37459      678.59      2179.69
## icu_patients_per_million                       15  37459       16.31        23.09
## hosp_patients                                  16  38664     3950.36     10016.41
## hosp_patients_per_million                      17  38664      130.59       153.43
## weekly_icu_admissions                          18  10136      341.58       528.66
## weekly_icu_admissions_per_million              19  10136       10.35        13.88
## weekly_hosp_admissions                         20  23093     4337.76     11101.43
## weekly_hosp_admissions_per_million             21  23093       86.16        89.50
## total_vaccinations                             22  66878 85868152.08 348008970.47
## people_vaccinated                              23  63504 30523003.22 114847697.84
## people_fully_vaccinated                        24  60334 27433712.27 104700247.50
## total_boosters                                 25  36317 15388292.74  39410647.50
## new_vaccinations                               26  53011   204694.34   1007757.07
## new_vaccinations_smoothed                      27 167203    81721.80    563089.93
## total_vaccinations_per_hundred                 28  66878      118.30        86.10
## people_vaccinated_per_hundred                  29  63504       52.06        29.98
## people_fully_vaccinated_per_hundred            30  60334       47.23        29.73
## total_boosters_per_hundred                     31  36317       37.14        31.59
## new_vaccinations_smoothed_per_million          32 167203     2014.78      3277.73
## new_people_vaccinated_smoothed                 33 166854    30240.84    209424.76
## new_people_vaccinated_smoothed_per_hundred     34 166854        0.08         0.19
## stringency_index                               35 197651       42.71        24.91
## population_density                             36 291332      406.23      1847.27
## median_age                                     37 270857       30.51         9.11
## aged_65_older                                  38 261324        8.70         6.11
## aged_70_older                                  39 268129        5.50         4.15
## gdp_per_capita                                 40 265416    19001.92     19991.25
## extreme_poverty                                41 170549       13.88        20.17
## cardiovasc_death_rate                          42 266024      264.43       121.21
## diabetes_prevalence                            43 279625        8.56         4.95
## female_smokers                                 44 199204       10.82        10.81
## male_smokers                                   45 196472       32.90        13.62
## handwashing_facilities                         46 129591       50.69        32.11
## hospital_beds_per_thousand                     47 234668        3.10         2.56
## life_expectancy                                48 315869       73.72         7.41
## human_development_index                        49 257801        0.72         0.15
## population                                     50 328453 33410028.36 134724317.05
## excess_mortality_cumulative_absolute           51  11944    51133.97    144322.76
## excess_mortality_cumulative                    52  11944        9.74        12.38
## excess_mortality                               53  11944       11.46        25.36
## excess_mortality_cumulative_per_million        54  11944     1644.92      1927.69
##                                                    min          max        range
## total_cases                                       1.00 1.034368e+08 1.034368e+08
## new_cases                                         0.00 6.966046e+06 6.966046e+06
```

```
## new_cases_smoothed                                0.00 5.882129e+06 5.882129e+06
## total_deaths                                      1.00 1.127152e+06 1.127151e+06
## new_deaths                                        0.00 1.144700e+04 1.144700e+04
## new_deaths_smoothed                               0.00 4.190000e+03 4.190000e+03
## total_cases_per_million                           0.00 7.375545e+05 7.375545e+05
## new_cases_per_million                             0.00 2.288720e+05 2.288720e+05
## new_cases_smoothed_per_million                    0.00 3.724178e+04 3.724178e+04
## total_deaths_per_million                          0.00 6.504190e+03 6.504190e+03
## new_deaths_per_million                            0.00 6.036600e+02 6.036600e+02
## new_deaths_smoothed_per_million                   0.00 1.486400e+02 1.486400e+02
## reproduction_rate                                -0.07 5.870000e+00 5.940000e+00
## icu_patients                                      0.00 2.889100e+04 2.889100e+04
## icu_patients_per_million                          0.00 1.806800e+02 1.806800e+02
## hosp_patients                                     0.00 1.544970e+05 1.544970e+05
## hosp_patients_per_million                         0.00 1.526850e+03 1.526850e+03
## weekly_icu_admissions                             0.00 4.838000e+03 4.838000e+03
## weekly_icu_admissions_per_million                 0.00 2.249800e+02 2.249800e+02
## weekly_hosp_admissions                            0.00 1.539770e+05 1.539770e+05
## weekly_hosp_admissions_per_million                0.00 7.092600e+02 7.092600e+02
## total_vaccinations                                0.00 3.491077e+09 3.491077e+09
## people_vaccinated                                 0.00 1.310292e+09 1.310292e+09
## people_fully_vaccinated                           1.00 1.276760e+09 1.276760e+09
## total_boosters                                    1.00 8.269130e+08 8.269130e+08
## new_vaccinations                                  0.00 2.474100e+07 2.474100e+07
## new_vaccinations_smoothed                         0.00 2.242429e+07 2.242429e+07
## total_vaccinations_per_hundred                    0.00 4.069000e+02 4.069000e+02
## people_vaccinated_per_hundred                     0.00 1.290700e+02 1.290700e+02
## people_fully_vaccinated_per_hundred               0.00 1.268900e+02 1.268900e+02
## total_boosters_per_hundred                        0.00 1.504700e+02 1.504700e+02
## new_vaccinations_smoothed_per_million             0.00 1.171130e+05 1.171130e+05
## new_people_vaccinated_smoothed                    0.00 6.785334e+06 6.785334e+06
## new_people_vaccinated_smoothed_per_hundred        0.00 1.171000e+01 1.171000e+01
## stringency_index                                  0.00 1.000000e+02 1.000000e+02
## population_density                                0.14 2.054677e+04 2.054663e+04
## median_age                                       15.10 4.820000e+01 3.310000e+01
## aged_65_older                                     1.14 2.705000e+01 2.591000e+01
## aged_70_older                                     0.53 1.849000e+01 1.797000e+01
## gdp_per_capita                                  661.24 1.169356e+05 1.162744e+05
## extreme_poverty                                   0.10 7.760000e+01 7.750000e+01
## cardiovasc_death_rate                            79.37 7.244200e+02 6.450500e+02
## diabetes_prevalence                               0.99 3.053000e+01 2.954000e+01
## female_smokers                                    0.10 4.400000e+01 4.390000e+01
## male_smokers                                      7.70 7.810000e+01 7.040000e+01
## handwashing_facilities                            1.19 1.000000e+02 9.881000e+01
## hospital_beds_per_thousand                        0.10 1.380000e+01 1.370000e+01
## life_expectancy                                  53.28 8.675000e+01 3.347000e+01
## human_development_index                           0.39 9.600000e-01 5.600000e-01
## population                                       47.00 1.425887e+09 1.425887e+09
## excess_mortality_cumulative_absolute         -37726.10 1.289777e+06 1.327503e+06
## excess_mortality_cumulative                     -44.23 7.655000e+01 1.207800e+02
## excess_mortality                                -95.92 3.776300e+02 4.735500e+02
## excess_mortality_cumulative_per_million       -2752.92 1.029292e+04 1.304584e+04
##                                                     se
## total_cases                                   13104.19
```

```
## new_cases                                      69.89
## new_cases_smoothed                             63.01
## total_deaths                                  158.73
## new_deaths                                      0.27
## new_deaths_smoothed                             0.23
## total_cases_per_million                       281.26
## new_cases_per_million                           2.12
## new_cases_smoothed_per_million                  1.09
## total_deaths_per_million                        2.12
## new_deaths_per_million                          0.01
## new_deaths_smoothed_per_million                 0.01
## reproduction_rate                               0.00
## icu_patients                                   11.26
## icu_patients_per_million                        0.12
## hosp_patients                                  50.94
## hosp_patients_per_million                       0.78
## weekly_icu_admissions                           5.25
## weekly_icu_admissions_per_million               0.14
## weekly_hosp_admissions                         73.05
## weekly_hosp_admissions_per_million              0.59
## total_vaccinations                        1345701.70
## people_vaccinated                          455744.83
## people_fully_vaccinated                    426252.22
## total_boosters                             206803.83
## new_vaccinations                             4376.96
## new_vaccinations_smoothed                    1377.07
## total_vaccinations_per_hundred                  0.33
## people_vaccinated_per_hundred                   0.12
## people_fully_vaccinated_per_hundred             0.12
## total_boosters_per_hundred                      0.17
## new_vaccinations_smoothed_per_million           8.02
## new_people_vaccinated_smoothed                512.70
## new_people_vaccinated_smoothed_per_hundred      0.00
## stringency_index                                0.06
## population_density                              3.42
## median_age                                      0.02
## aged_65_older                                   0.01
## aged_70_older                                   0.01
## gdp_per_capita                                 38.80
## extreme_poverty                                 0.05
## cardiovasc_death_rate                           0.24
## diabetes_prevalence                             0.01
## female_smokers                                  0.02
## male_smokers                                    0.03
## handwashing_facilities                          0.09
## hospital_beds_per_thousand                      0.01
## life_expectancy                                 0.01
## human_development_index                         0.00
## population                                 235076.59
## excess_mortality_cumulative_absolute         1320.57
## excess_mortality_cumulative                     0.11
## excess_mortality                                0.23
## excess_mortality_cumulative_per_million        17.64
```

Gyakorlas

- Mi az egy millio fore eso uj esetek (*new_cases_per_million*) atlaga (mean)?
- Hany valid (nem NA) adat szerepel az adatbazisban az egy fore eso gdp-rol (*gdp_per_capita*)?

Faktorok

Nehany karaktervaltozonak csak **korlatozott mennyisegu eleme** lehet, mint peldaul a continent (North America, Asia, Africa, Europe, South America, Oceania). Ezeket megjelolhetjuk faktor (factor) osztalyu valtozokent, es akkor az R tobb informaciot fog adni rola.

```
COVID_data <- COVID_data %>%
              mutate(continent = factor(continent),
                     location = factor(location))

levels(COVID_data$continent)
```

```
## [1] "Africa"        "Asia"          "Europe"        "North America"
## [5] "Oceania"       "South America"
```

```
table(COVID_data$continent)
```

```
##
##        Africa          Asia        Europe North America       Oceania
##         77749         68263         74667         55926         32737
## South America
##         19111
```

```
COVID_data <- COVID_data %>%
              mutate(continent = factor(continent))
```

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

Amikor kilistazzuk a faktor valtozot, akkor is kiirja az R a lista aljara, hogy milyen faktorszintek vannak.

```
levels(COVID_data$continent)

table(COVID_data$continent)

COVID_data$continent
```

Alabb csinalunk egy COVID_data_latest valtozot, amivel csak 2023-09-01-én beekrezett adatok szerepelnek, hogy kisebb legyen az adattabla amivel dolgozunk.

```
COVID_data_latest = COVID_data %>%
  filter(date == "2023-09-01")
```

Miutan egy valtozot faktorkent azonositottunk, bizonyos funkciok kepesek felhasznalni ezt az informaciot. Peldaul igy mar a fenti **summary()** funkcio is kiadja az **egyes faktorszintekrol** hogy hany megfigyeles tartozik az egyes kategoriakba (faktorszintekbe).

```
COVID_data %>%
 mutate(continent = as.character(continent)) %>%
  select(continent) %>%
  summary()
```

```
##   continent
##  Length:328453
##  Class :character
##  Mode  :character
```

```
# continent is already recognized as a factor variable
COVID_data_latest %>%
  select(continent) %>%
  summary()
```

```
##          continent
##  Africa       :57
##  Asia         :48
##  Europe       :54
##  North America:41
##  Oceania      :24
##  South America:14
```

Van, hogy szeretnenk **kizarni** bizonyos **faktorszinteket** az elemzesbol. Pl. ha valamelyik faktor szintbol nagyon keves megfigyeles van, mondjuk Oceaniat, mondjuk mert ugy gondoljuk hogy az tulsagosan "elszigetelt" a vilag tobbi reszetol, 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**. A **faktorszintek meg akkor is megmaradnak ha nem marad egy megfigyeles sem** az adott faktorszinten az adattablaban.

```
COVID_data_latest %>%
  filter(continent != "Oceania") %>%
  select(total_cases, continent) %>%
  summary()
```

```
##   total_cases                continent
##  Min.   :       26   Africa       :57
##  1st Qu.:    37338   Asia         :48
##  Median :   272922   Europe       :54
##  Mean   :  3634884   North America:41
##  3rd Qu.:  1379702   Oceania      : 0
##  Max.   :103436829   South America:14
##  NA's   :6
```

Igy ezeket a szinteket ejthetjuk a **droplevels()** funkcioval.

```
COVID_data_latest_noOceania = COVID_data_latest %>%
  filter(continent != "Oceania") %>%
  mutate(continent = droplevels(continent))



COVID_data_latest_noOceania %>%
  select(continent) %>%
  summary()
```

```
##          continent
##  Africa       :57
##  Asia         :48
##  Europe       :54
##  North America:41
##  South America:14
```

Faktorszintek egymashoz viszonyitott erteke

Legtobbszor a faktorszintek kozott nincs "ertekbeli" kulonbseg, egyszeruen csoportnevekrol van szo, de neha egy meghatarozott relacio van kozottuk, pl. a legmagasabb iskolai vegzettsge lehet vegzettsge nelkuli < altalanos iskolai < kozepiskolai < felsofoku ... Ittfaktorszinteknek van egy meghatarozott hierarchiaja, vagy sorrendje. Ilyen valtozo nincs ebben az adatbazisban, de konnyeden csinalhatunk ilyen faktor valtozot.

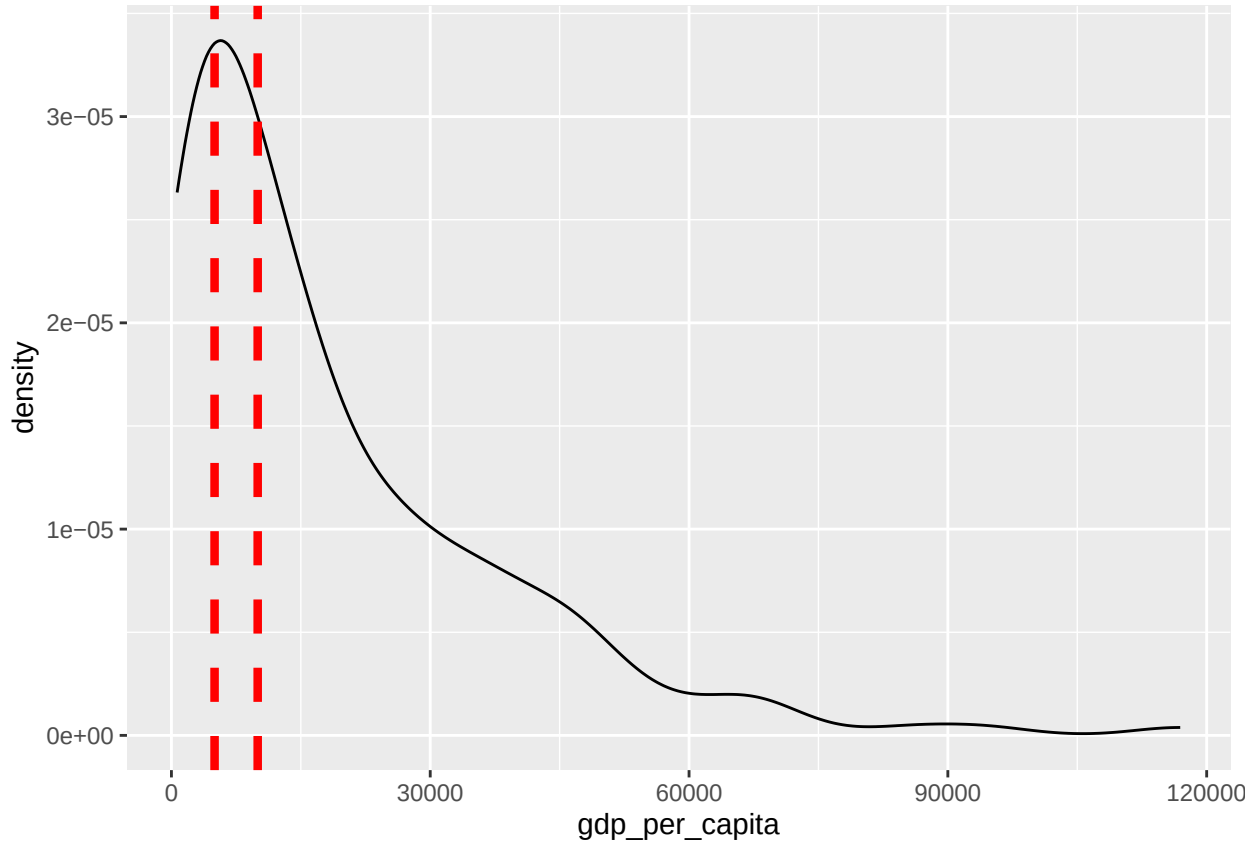
Ehhez arra van szuksegunk, hogy egy **numerikus valtozot alakitsunk faktorra**, pl. elkepzelheto hogy ossze akarjuk hasonlitani azokat az orszagokat ahol 5000 alatti a gdp_per_capita azokkal akinel e feletti, hogy hogyan kulonboznek a COVID adatok.

```
COVID_data_latest %>%
  select(gdp_per_capita, continent) %>%
  drop_na() %>%
  group_by(continent) %>%
  summarize(mean_gdp = mean(gdp_per_capita))
```

```
## # A tibble: 6 x 2
##   continent     mean_gdp
##   <fct>            <dbl>
## 1 Africa           5444.
## 2 Asia            22057.
## 3 Europe          33361.
## 4 North America   21655.
## 5 Oceania         10618.
## 6 South America   13841.
```

```
COVID_data_latest %>%
  select(gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
  aes(x = gdp_per_capita) +
  geom_density() +
  geom_vline(xintercept = c(5000, 10000), linetype="dashed",
                color = "red", size=1.5)
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Ilyenkor hasznalhatjuk a **mutate()** es **case_when()** 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.

```
COVID_data = COVID_data %>%
  mutate(gdp_per_capita_kat = factor(
                                     case_when(gdp_per_capita < 5000 ~ "small",
                                               gdp_per_capita >= 5000 & gdp_per_capita < 10000 ~ "medi
                                               gdp_per_capita > 10000 ~ "large")))
levels(COVID_data$gdp_per_capita_kat)
```

```
## [1] "large"  "medium" "small"
```

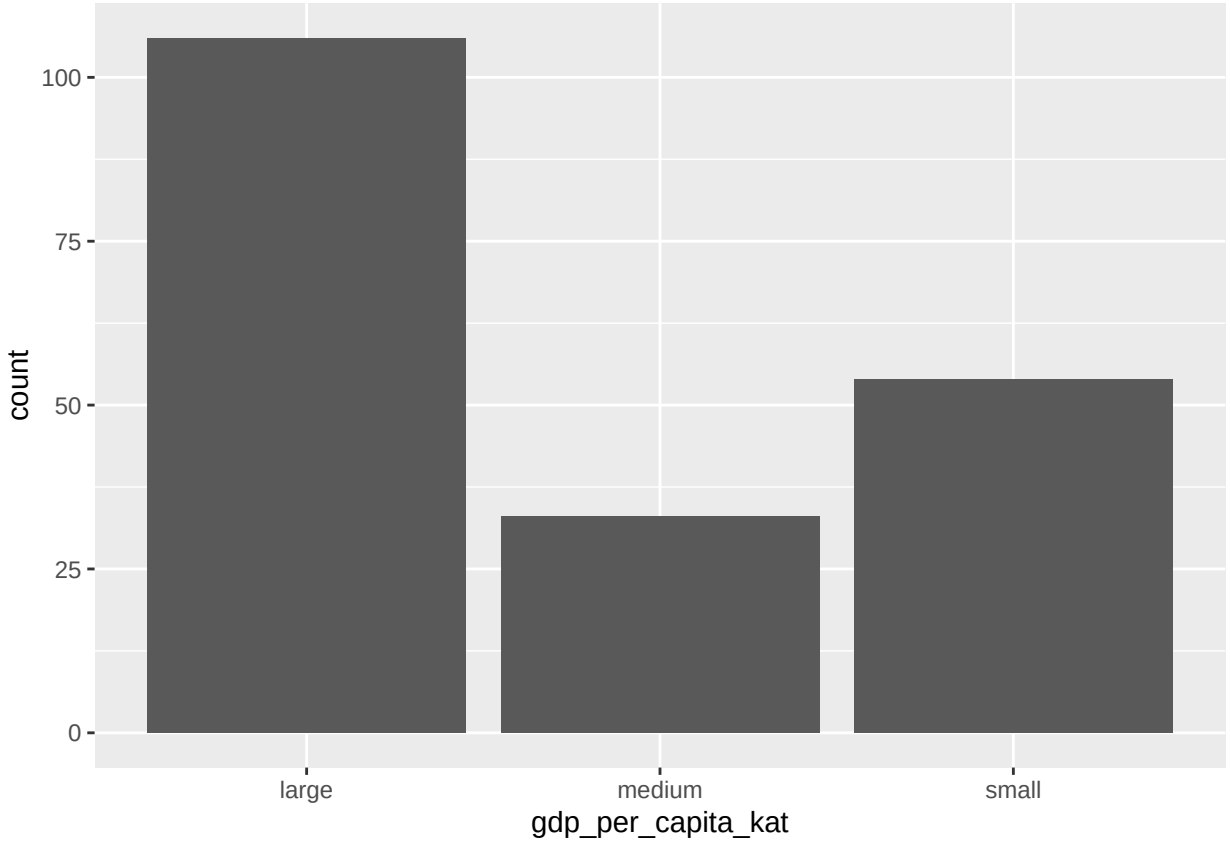
```
# ugyanez a COVID_data_latest -al

COVID_data_latest = COVID_data_latest %>%
  mutate(gdp_per_capita_kat = factor(
                                     case_when(gdp_per_capita < 5000 ~ "small",
```

```
                                                gdp_per_capita >= 5000 & gdp_per_capita < 10000 ~ "medi
                                                gdp_per_capita > 10000 ~ "large")))
```

Amikor abrat rajzolunk erreol a valtozorol, lathatjuk hogy a faktorszintek sorrendje “large”, “medium”, es “small” az abran.

```
COVID_data_latest %>%
  select(gdp_per_capita_kat) %>%
  drop_na() %>%
  ggplot() +
  aes(x = gdp_per_capita_kat) +
  geom_bar()
```

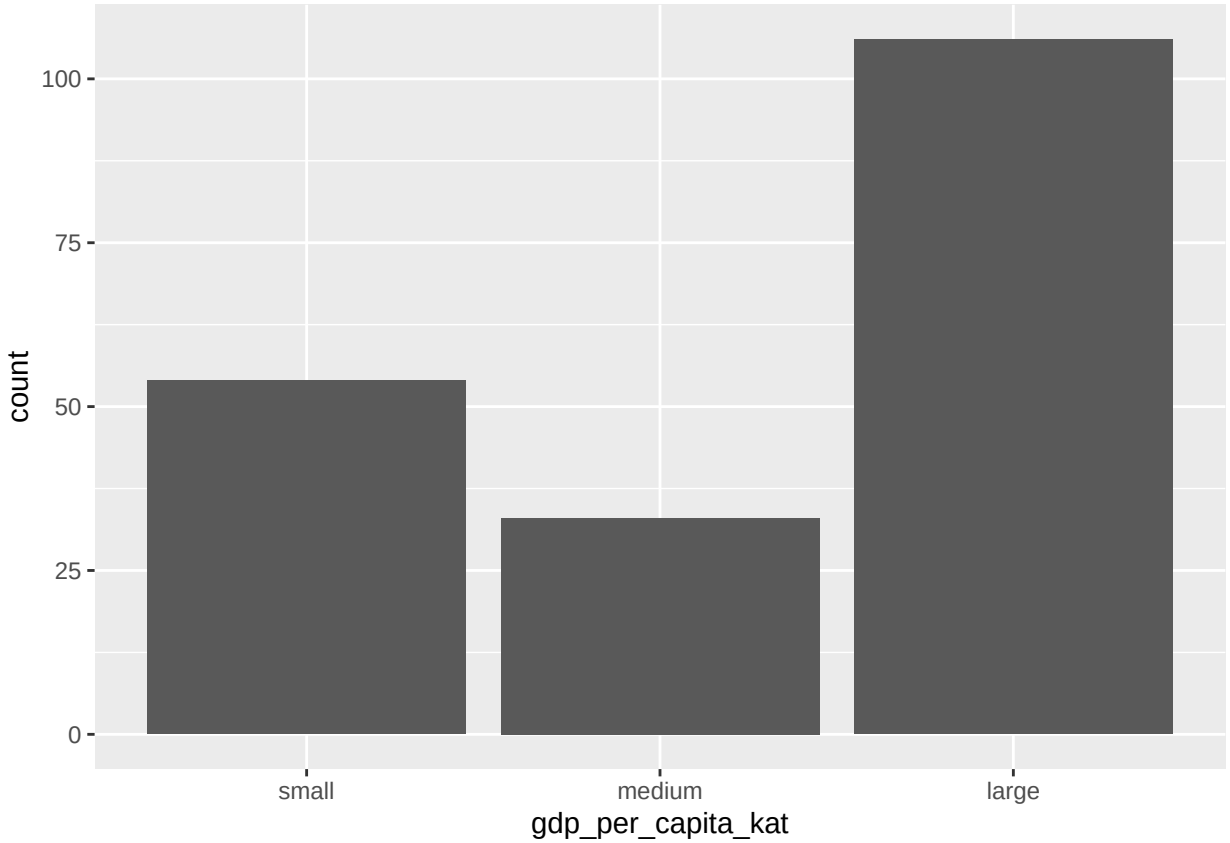


Ez nem feltetlenul intuitiv abrazolas, hiszen altalaban a kisebbtol a nagyobbig szoktunk haladni balrol jobbra. De az R nem tudja mit jelentenek a faktorszintek nevei. A faktorszintek sorrendjenek meghatarozasanal ezert alapertelmezett modon **abc sorrendet hasznal**.

Specifikalhatjuk maskepp is a faktroszintek sorrendjet a factor funkcioban a **levels = c()** parameteren keresztul egy vektorban megadva.

```
COVID_data_latest = COVID_data_latest %>%
mutate(gdp_per_capita_kat = factor(gdp_per_capita_kat, levels = c(
                                                "small",
                                                 "medium",
                                                 "large")))
```

```
COVID_data_latest %>%
  select(gdp_per_capita_kat) %>%
  drop_na() %>%
  ggplot() +
  aes(x = gdp_per_capita_kat) +
  geom_bar()
```



Attol meg hogy megadjuk a levels-el a faktorszintek listazasi sorrendjet, az R meg mindig egyenrangukent kezeli a faktorszinteket. Ha azt szeretnenk ha az R ugy ertekelne hogy a faktorszintek valamilyen hierarchikus sorrendben van, vagyis **ordinalis valtozokent**, akkor ezt a factor() funkcion belul az **ordered = T** parameter beallitasaval tehetjuk meg.

Ha ezt teszuk, a faktor valtozo kilistazasakor relacio-jelek kerulnek a faktorszintek koze, es mas funkciok is fel tudjak majd hasznalni ezt az informaciot.

```
COVID_data_latest = COVID_data_latest %>%
mutate(gdp_per_capita_kat = factor(gdp_per_capita_kat, ordered = T, levels = c(
                                        "small",
                                         "medium",
                                         "large")))
COVID_data_latest$gdp_per_capita_kat
```

```
##   [1] small  large  large  <NA>   <NA>   medium <NA>   large  large  medium
##  [11] large  large  large  large  large  large  small  large  large  large
##  [21] medium small  large  medium medium <NA>   large  large  large  <NA>
##  [31] large  large  small  small  small  small  large  medium large  small
```

```
##  [41] small  large  large  large  small  small  <NA>   large  small  large
##  [51] <NA>   <NA>   large  large  small  large  small  medium large  large
##  [61] large  medium <NA>   large  small  large  medium small  <NA>   <NA>
##  [71] medium large  large  <NA>   <NA>   large  small  medium large  small
##  [81] <NA>   large  <NA>   large  <NA>   <NA>   medium <NA>   small  small
##  [91] medium small  small  large  large  medium large  large  large  large
## [101] <NA>   large  large  medium large  <NA>   medium large  small  small
## [111] medium large  small  medium large  large  small  small  large  <NA>
## [121] large  large  small  small  large  large  small  large  small  <NA>
## [131] small  large  <NA>   large  small  medium <NA>   large  large  <NA>
## [141] medium small  medium medium large  small  large  <NA>   large  medium
## [151] small  medium <NA>   <NA>   large  <NA>   <NA>   large  large  medium
## [161] large  small  large  small  medium large  medium <NA>   large  large
## [171] large  large  <NA>   large  large  small  <NA>   <NA>   large  large
## [181] <NA>   <NA>   large  medium large  small  large  <NA>   small  large
## [191] large  small  large  large  large  large  small  <NA>   large  large
## [201] small  large  large  small  large  large  large  <NA>   <NA>   small
## [211] small  large  medium small  <NA>   medium large  large  large  large
## [221] <NA>   small  small  medium large  large  large  <NA>   large  medium
## [231] small  <NA>   large  medium <NA>   small  small  small
## Levels: small < medium < large
```

Kategorikus valtozo ujrakodolasa

Egy masik funkcio amivel manipulalhatjuk a faktorszinteket, a **recode()**. Ha kategorikus valtozokat szeretnenk atkodolni, mondjuk ha szeretnenk a deli felteket az eszaki feltekevel osszehasonlitani, ezt a kovetkezokeppen tehetjuk:

```
COVID_data = COVID_data %>%
  mutate(continent_south_north = factor(recode(continent,
                                               "Oceania" = "South",
                                               "South America" = "South",
                                               "Africa" = "South",
                                               "Asia" = "North",
                                               "Europe" = "North",
                                               "North America" = "North")))

levels(COVID_data$continent_south_north)
```

```
## [1] "South" "North"
```

```
COVID_data_latest = COVID_data_latest %>%
  mutate(continent_south_north = factor(recode(continent,
                                               "Oceania" = "South",
                                               "South America" = "South",
                                               "Africa" = "South",
                                               "Asia" = "North",
                                               "Europe" = "North",
                                               "North America" = "North")))
```

Gyakorlas

- szurd az adatokat ugy hogy csak a tegnapi adatokkal dolgozzunk.
- csinalj egy uj kategorikus valtozot (nevezzuk ezt *new_cases_per_million_kat*-nak) a mutate() funkcio hasznalataval amiben azok az orszagok ahol a *new_cases_per_million* valtozo 20 alatt van "small", ahol 20 vagy a felett van "large" kategoriaba keruljenek.
- figyelj oda hogy faktorkent jelold meg ezt az uj valtozot (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)
- mentsd el ezt a valtozot az eredeti adatobjektumban ugy hogy kesobb is lehessen vele dolgozni
- keszits egy tablazatot arrol, hogy hanyan esnek a *new_cases_per_million_kat* egyes kategoriaiba.
- Add meg a faktorszintek helyes sorrendjet: small, large (Ird felul a *new_cases_per_million_kat* korabbi valtozatat ezzel a valtozattal ahol a szintek mar helyes sorrendben vannak, vagy ezt a sorrendezest is bele vonhatod az eredeti funkcioba, amivel a valtozot generaltad)
- Ellenorizd, hogy valoban helyes sorrendben szerepelnek-e a faktor szintjei.

Exploracio vizualizacion keresztul

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

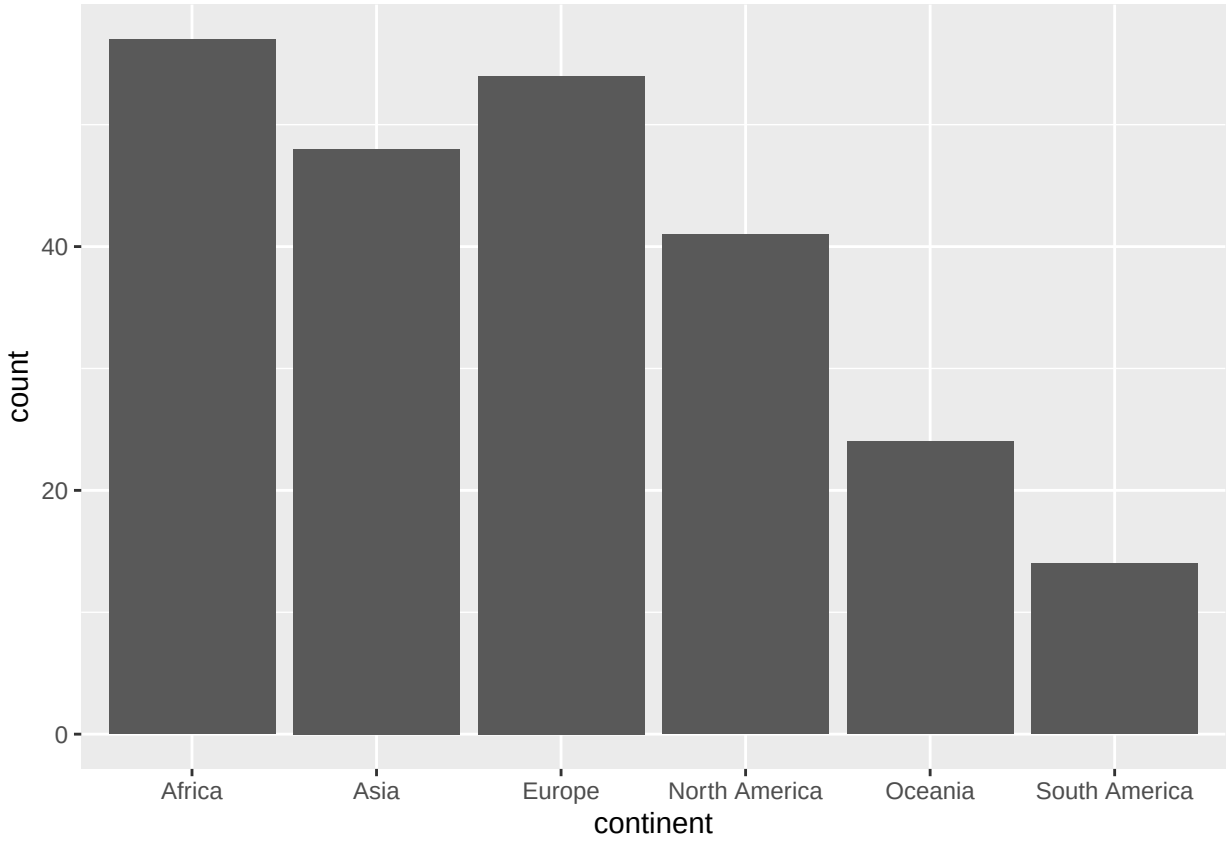
MINDING ellenorizd az adataidat ezekkel a modszerekkel mielott komolyabb adatelemzesbe kezdesz, hogy meggyozodj rola, hogy az adatok tisztak es megfelenek az elvarasaidnak.

Egyes valtozok vizualizacioja

Az egyes valtozok peldaul **abrak** (plot) segitsegevel megvizsgalhatok.

A **kategorikus** valtozokat gyakran oszlopdiagrammal (**geom_bar**) abrazoljuk,

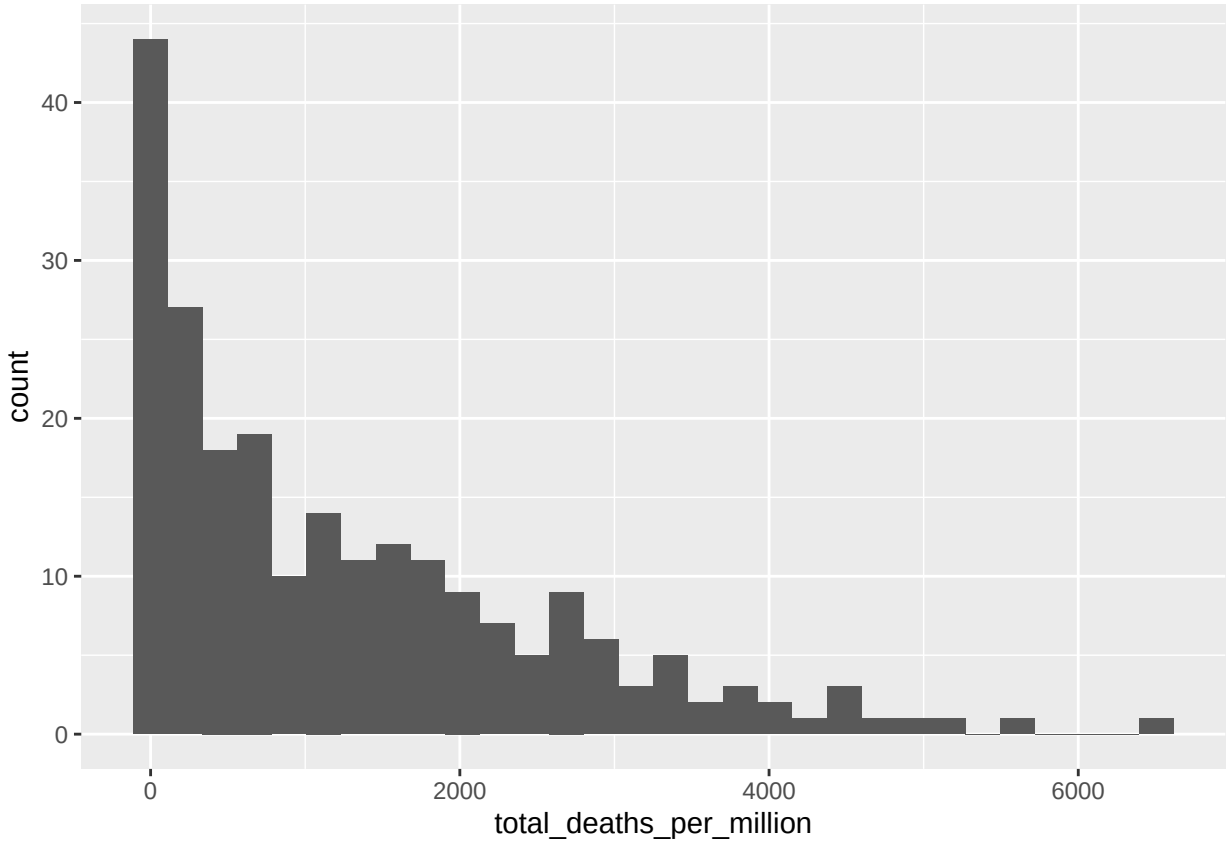
```
COVID_data_latest %>%
ggplot() +
  aes(x = continent) +
  geom_bar()
```




```
COVID_data_latest %>%
ggplot() +
  aes(x = total_deaths_per_million) +
  geom_histogram()
```

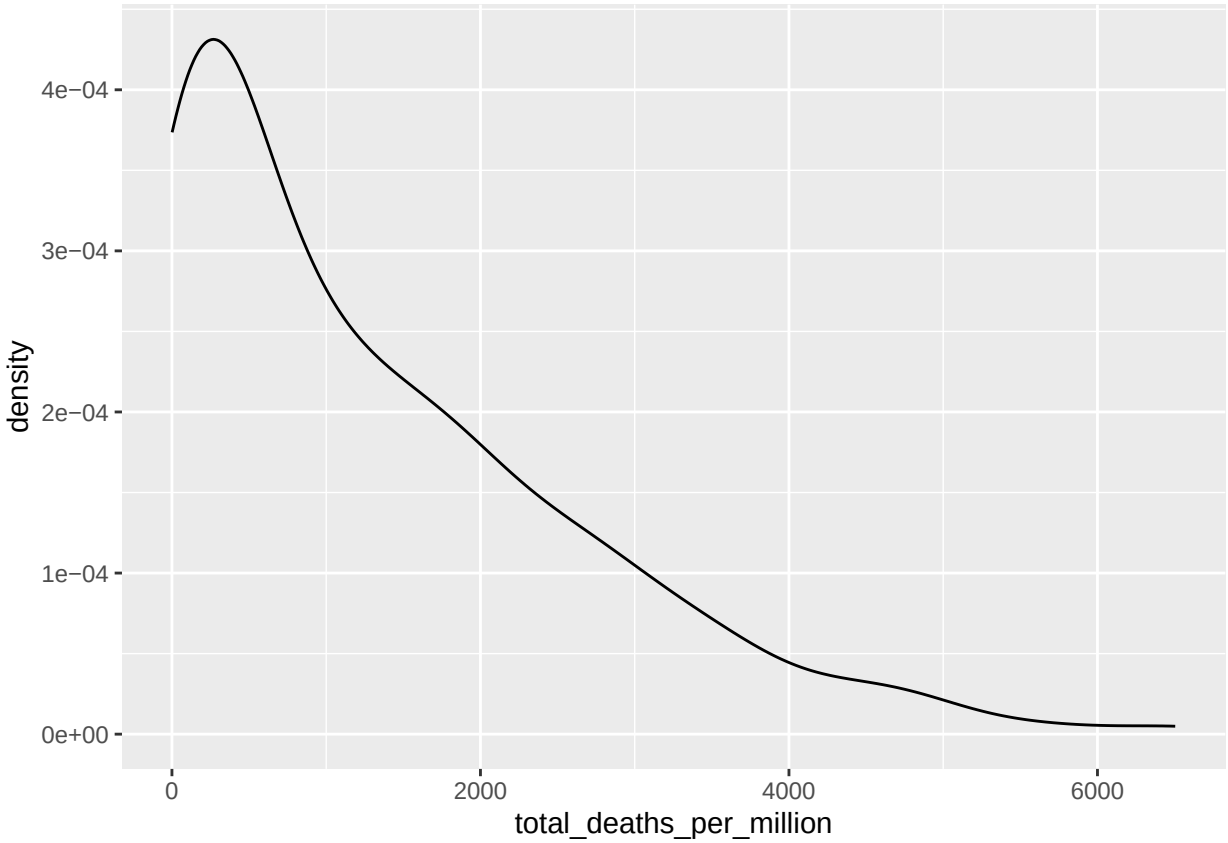
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 12 rows containing non-finite values (`stat_bin()`).
```




```
COVID_data_latest %>%
ggplot() +
  aes(x = total_deaths_per_million) +
  geom_density()
```

```
## Warning: Removed 12 rows containing non-finite values (‘stat_density()‘).
```



Gyakorlas

Szurd az adatokat ugy hogy csak a 2020-09-07-en jeletett adatokkal dolgozzunk, és úgy, hogy csak a total_cases, new_cases, people_vaccinated, location, continent változók legyenek a vizsgált adatbázisban.

Hasznald a fent tanult modszereket, hogy **azonositsd az COVID_data 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.

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 valoszinu hogy mi a helyes valasz, beirhatjuk, DE **minden javitast fel**

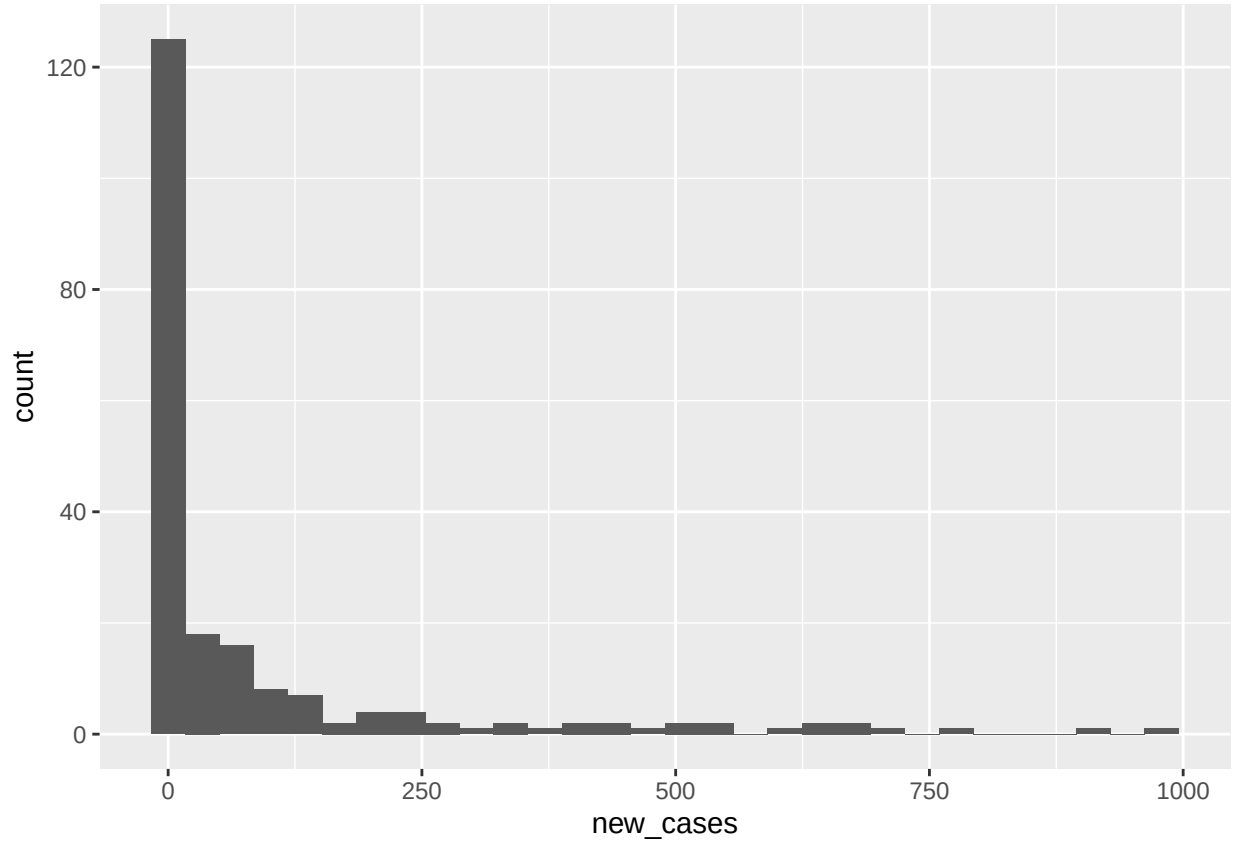
kell tuntetni a kutatasi jelentesben (es a ZH soran is), hogy az olvaso szamara tiszta legyen, hogy itt egy adathelyettesites vagy kizaras tortent!

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

```
COVID_data %>%
  filter(date == "2020-09-07") %>%
  select(new_cases) %>%
  summary()
```

```
##    new_cases
##  Min.   :    0
##  1st Qu.:    0
##  Median :    8
##  Mean   : 1098
##  3rd Qu.:  161
##  Max.   :90802
##  NA's   :7
```

```
COVID_data %>%
  filter(date == "2020-09-07", new_cases < 1000) %>%
  ggplot()+
    aes(x = new_cases)+
    geom_histogram()
```



```r
COVID_data_corrected <- COVID_data %>%
  mutate(new_cases = replace(new_cases,  new_cases=="-7953", NA))
```

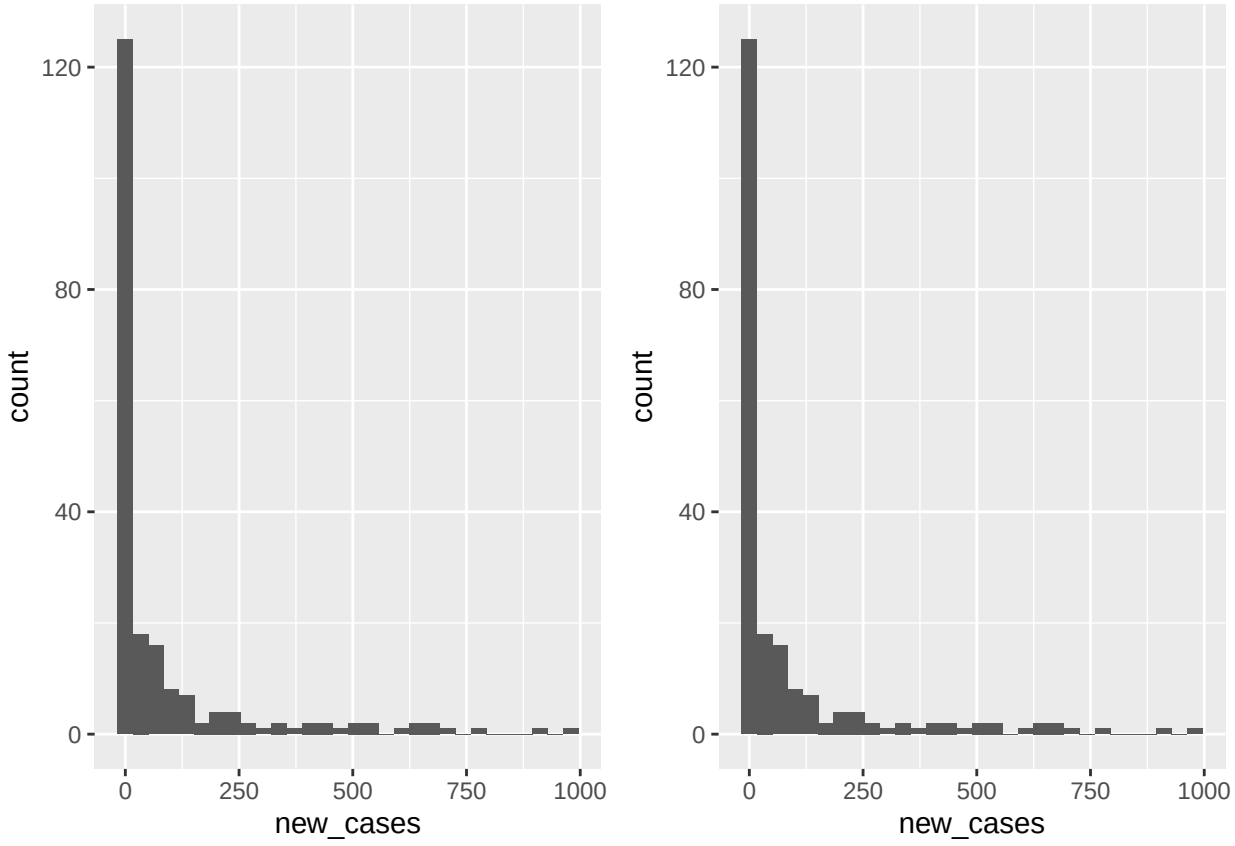
Erdemes **megbizonyosodni rola**, hogy az adatcsere sikeres volt. Alabb az adatok vizualizaciojaval gyozodunk meg errol, de az adatok megjelenitesevel, vagy a leiro statisztikak lekerdezesevel is megteheto ez, ha az informativ.

```r
# hasznalhatnak meg az alabbiakat is arra,
# hogy megbizonyosodjunk abban, hogy sikeres volt a csere
# View(COVID_data_corrected)
# describe(COVID_data_corrected)
# summary(COVID_data_corrected$szocmedia_3)
# COVID_data_corrected$szocmedia_3

old_plot <-
  COVID_data %>%
  filter(date == "2020-09-07", new_cases < 1000) %>%
  ggplot()+
    aes(x = new_cases)+
    geom_histogram()

new_plot <-
  COVID_data_corrected %>%
  filter(date == "2020-09-07", new_cases < 1000) %>%
  ggplot()+
    aes(x = new_cases)+
    geom_histogram()


grid.arrange(old_plot, new_plot, ncol=2)
```



Tobb valtozo kapcsolatanak felterkepezese

Tobb valtozo kapcsolatat is felterkepezhetjuk tablazatok es abrak segitsegevel.

Ket kategorikus (csoportosito) valtozo kapcsolatanak felterkepezese

Feltaro elemzes

Most vizsgaljuk meg azt, hogy 2020-09-28-an mi az osszefuggese a gdp kategorianak (*gdp_per_capita_kat*) a kontinenssel (*continent*) ahol az orszag elhelyezkedik.

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

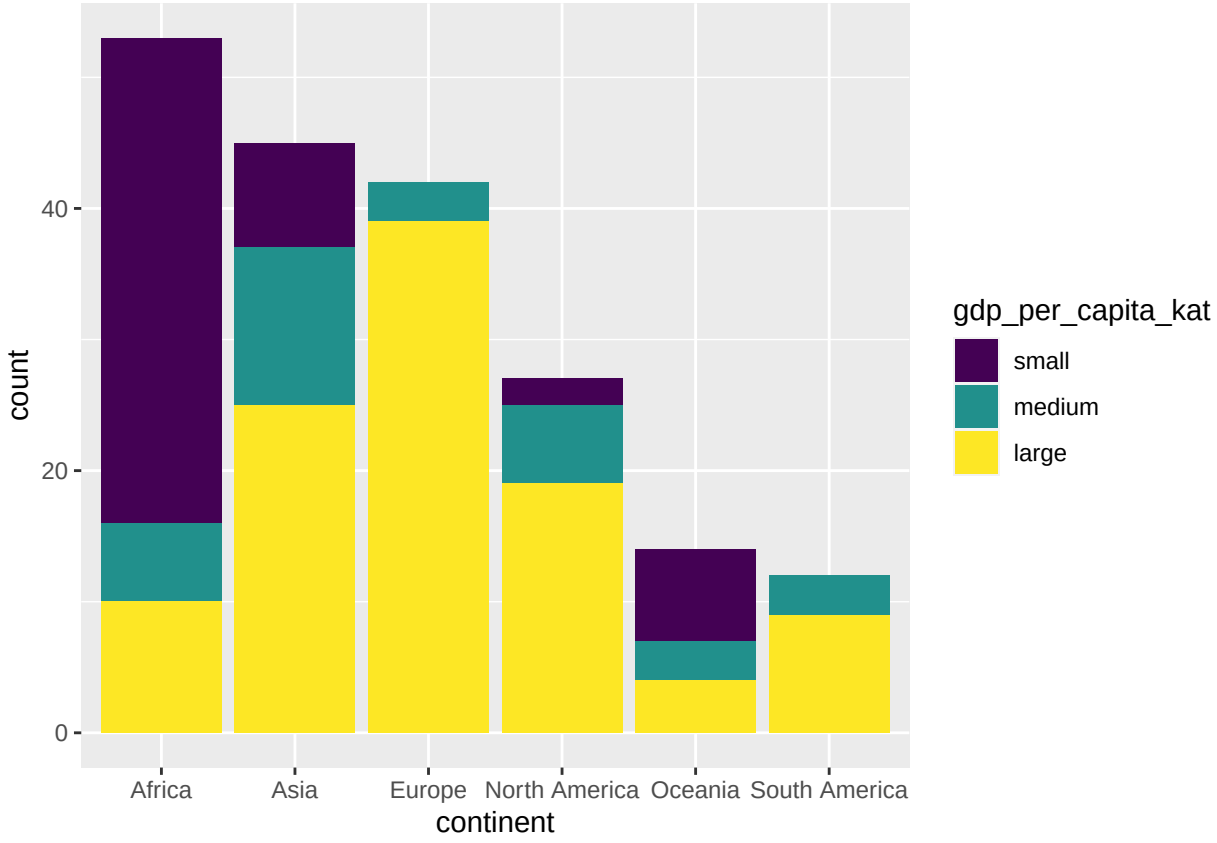
```
table(COVID_data_latest$gdp_per_capita_kat, COVID_data_latest$continent)
```

```
##
##          Africa Asia Europe North America Oceania South America
##   small      37    8      0             2       7             0
##   medium      6   12      3             6       3             3
##   large      10   25     39            19       4             9
```

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

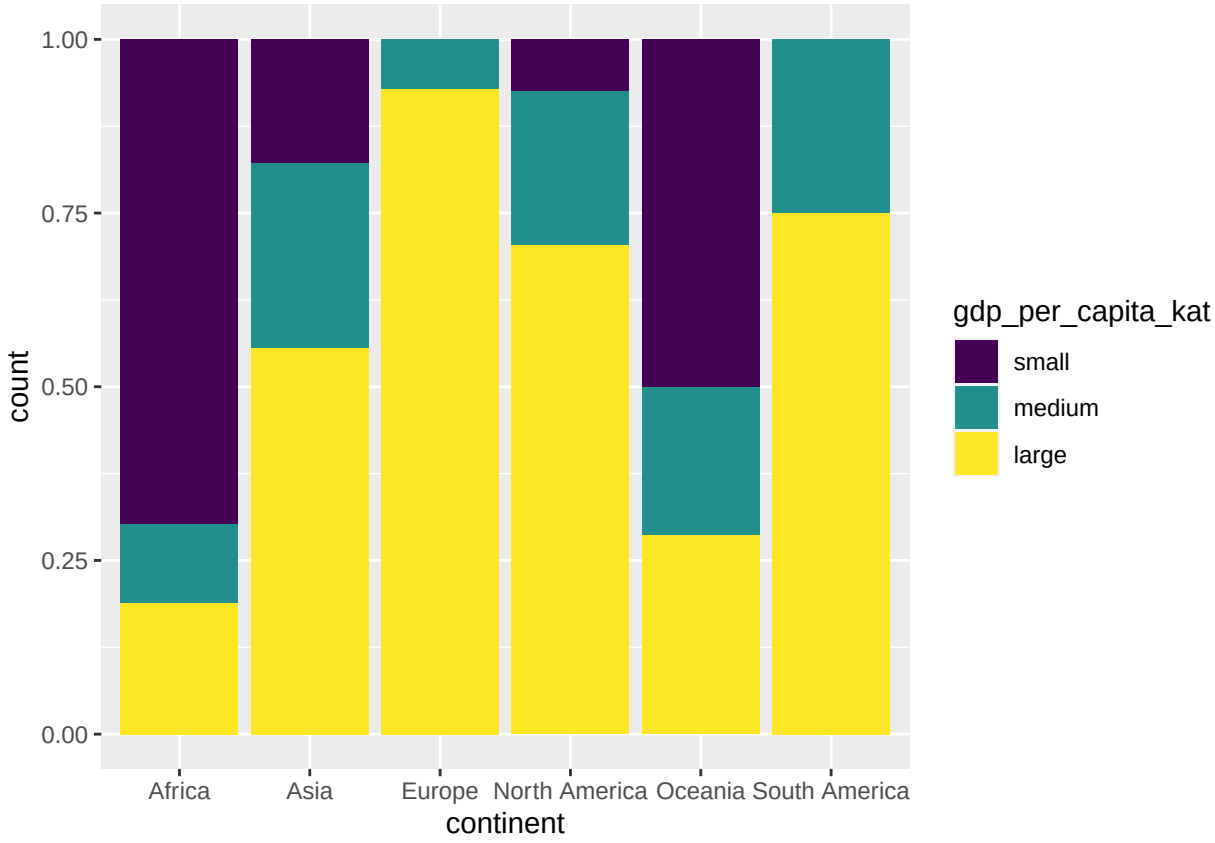
Erre az egyik lehetoseg a **stacked bar chart** (egymasra tornyozott oszlopdiagram, a **geom_bar()** geomot hasznaljuk) hasznalata. Itt az egyik valtozo kategoriai adjak meg hany oszlop lesz (ez a valtozo lesz az x tengelyen reprezentalva, igy ezt az "x =" reszen adhatjuk meg), a masik valtozo az oszlopokat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

```
COVID_data_latest %>%
  drop_na(gdp_per_capita_kat) %>%
ggplot() +
  aes(x = continent, fill = gdp_per_capita_kat) +
  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")**. Vagy hasznalhatjuk az eltolt oszlopdiagramot (dodged barchart) (a **position = "dodge"** parameter megadasaval a geom_bar() -on belul)

```
COVID_data_latest %>%
  drop_na(gdp_per_capita_kat) %>%
ggplot() +
  aes(x = continent, fill = gdp_per_capita_kat) +
  geom_bar(position = "fill")
```



Gyakorlas

Hasznald a fent tanult modszereket, hogy megvizsgald a COVID_data_latest adatbazisban a **new_cases_per_million_ka** es a **continent** 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?

```
# a fenti gyakorlashoz a new_cases_per_million_kat valtozot igy lehet legeneralni:

COVID_data = COVID_data %>%
  mutate(new_cases_per_million_kat = factor(
                                    case_when(new_cases_per_million < 20 ~ "small",
                                              new_cases_per_million >= 20 ~ "large"), ordered = T, lev

levels(COVID_data$new_cases_per_million_kat)
```

```
## [1] "small" "large"
```

```
# ugyanez a COVID_data_latest -al

COVID_data_latest = COVID_data_latest %>%
```

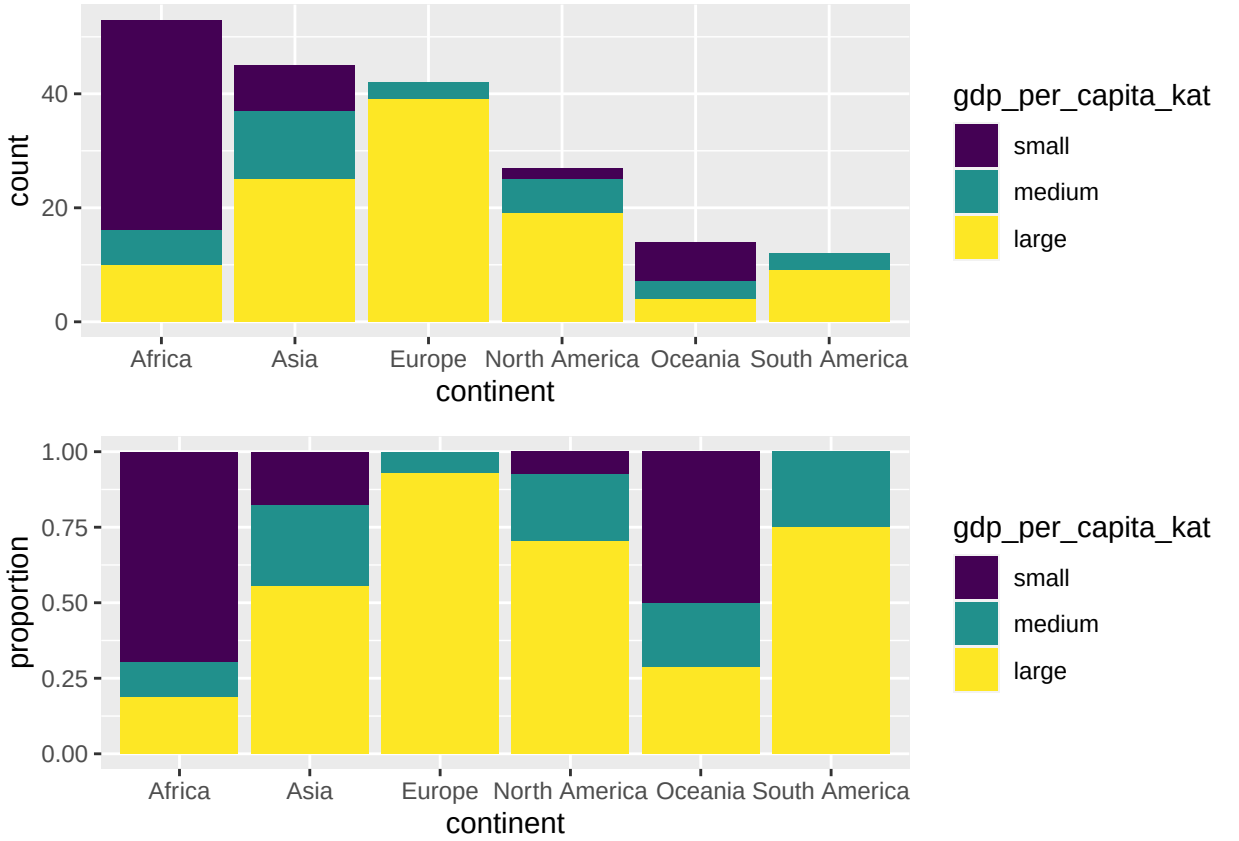
```
  mutate(new_cases_per_million_kat = factor(
                                      case_when(new_cases_per_million < 20 ~ "small",
                                                new_cases_per_million >= 20 ~ "large"), ordered = T, lev
```

geom_bar() 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.

```
szamossag_plot <-
COVID_data_latest %>%
  drop_na(gdp_per_capita_kat) %>%
ggplot() +
  aes(x = continent, fill = gdp_per_capita_kat) +
  geom_bar()

reszarany_plot <-
COVID_data_latest %>%
  drop_na(gdp_per_capita_kat) %>%
ggplot() +
  aes(x = continent, fill = gdp_per_capita_kat) +
  geom_bar(position = "fill") +
  ylab("proportion")

grid.arrange(szamossag_plot, reszarany_plot, nrow=2)
```



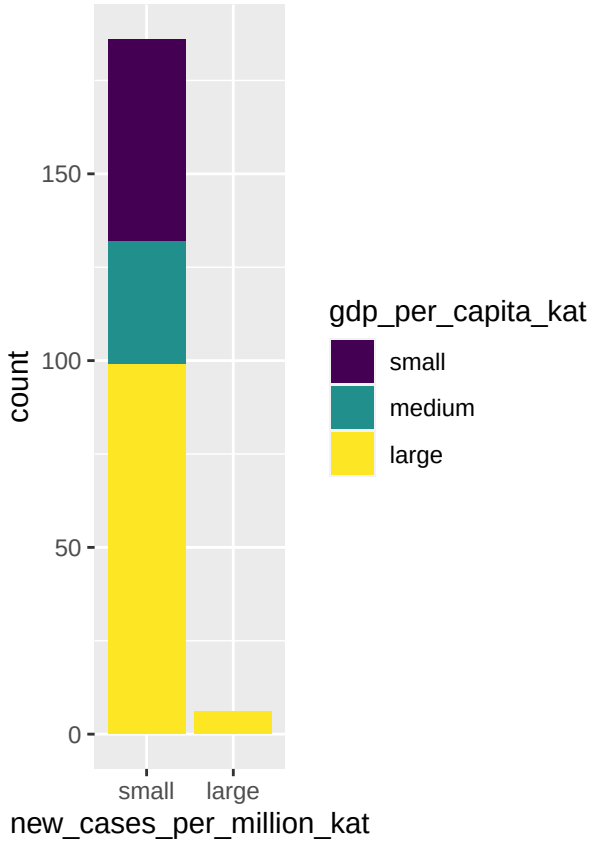
A theme(legend.position) es a guides() funciok hasznalataval kontrollalhatjuk hogy hol es hogyan jelenjen meg a **jelmagyarazat** az abran. Az abra **interpretalhatosaga** attol fuggoen is **valtozhat**, hogy melyik valtozot tesszuk az x-tengelyre es melyiket szinkent abrazolva.

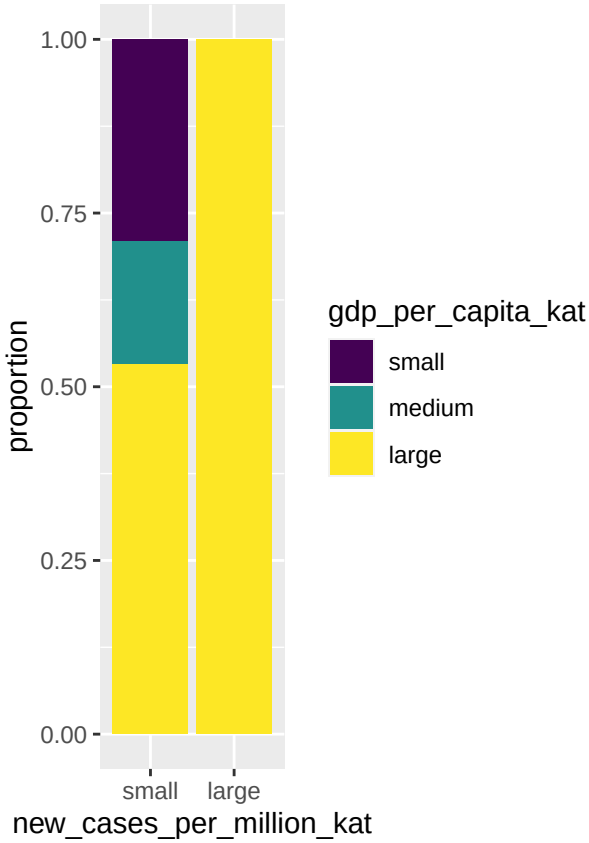
Az alabbi abrakon az egymillio fore vetitett uj esetek szamanak kapcsolatat nezzuk meg a gdp-vel. Mindket valtozo eseten a csoportositott valtozot (_kat) hasznaljuk.

```
barchart_plot_3 <-
COVID_data_latest %>%
  select(new_cases_per_million_kat, gdp_per_capita_kat) %>%
  drop_na() %>%
ggplot() +
  aes(x = new_cases_per_million_kat, fill = gdp_per_capita_kat) +
  geom_bar()


barchart_plot_4 <-
COVID_data_latest %>%
  select(new_cases_per_million_kat, gdp_per_capita_kat) %>%
  drop_na() %>%
ggplot() +
  aes(x = new_cases_per_million_kat, fill = gdp_per_capita_kat) +
  geom_bar(position = "fill") +
  ylab("proportion")

grid.arrange(barchart_plot_3, barchart_plot_4, ncol=2)
```



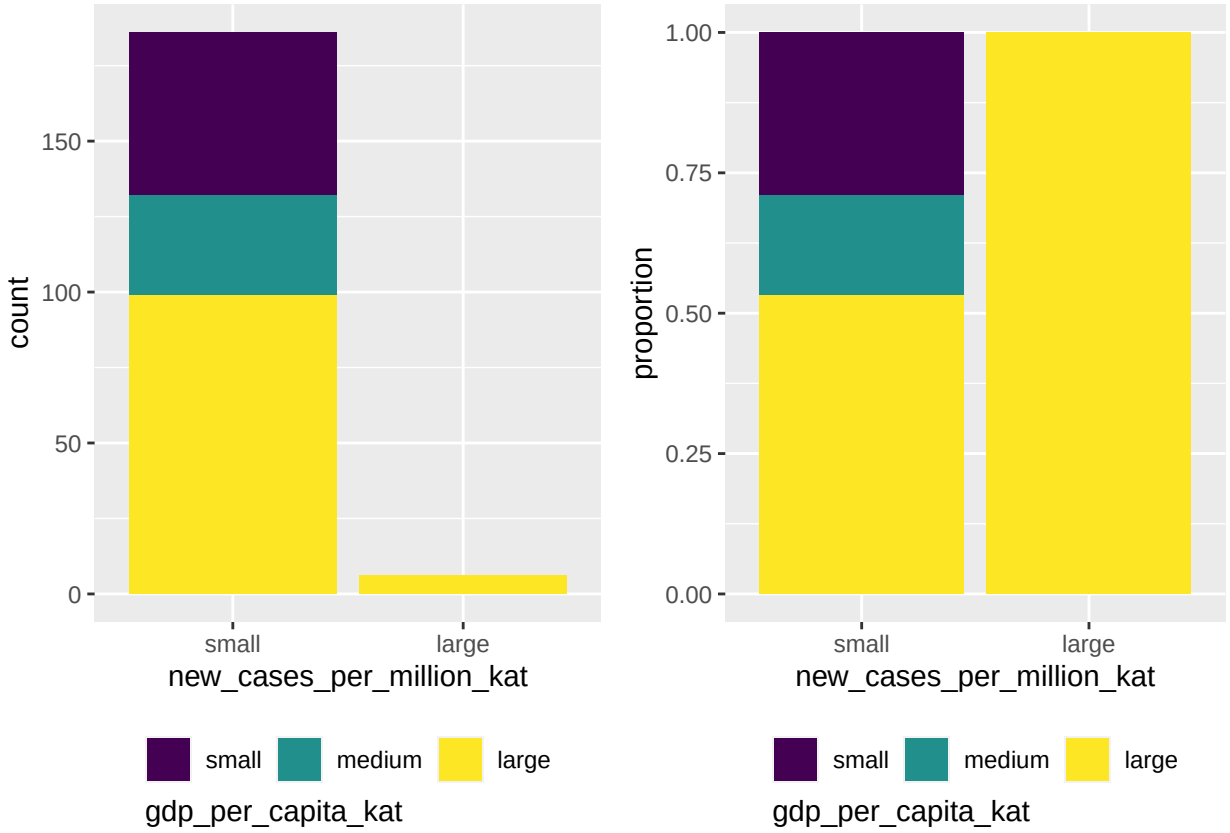



```
# a theme(legend.position) es a guides() funciok
# hasznalataval kontrollalhatjuk hogy hol es hogyan
# jelenjen meg a jelmagyarazat az abran

barchart_plot_3 <-
COVID_data_latest %>%
  select(new_cases_per_million_kat, gdp_per_capita_kat) %>%
  drop_na() %>%
ggplot() +
    aes(x = new_cases_per_million_kat, fill = gdp_per_capita_kat) +
    geom_bar() +
    theme(legend.position="bottom") +
    guides(fill = guide_legend(title.position = "bottom"))


barchart_plot_4 <-
COVID_data_latest %>%
  select(new_cases_per_million_kat, gdp_per_capita_kat) %>%
  drop_na() %>%
ggplot() +
  aes(x = new_cases_per_million_kat, fill = gdp_per_capita_kat) +
  geom_bar(position = "fill") +
  theme(legend.position="bottom") +
  guides(fill = guide_legend(title.position = "bottom")) +
  ylab("proportion")
```

```
grid.arrange(barchart_plot_3, barchart_plot_4, ncol=2)
```

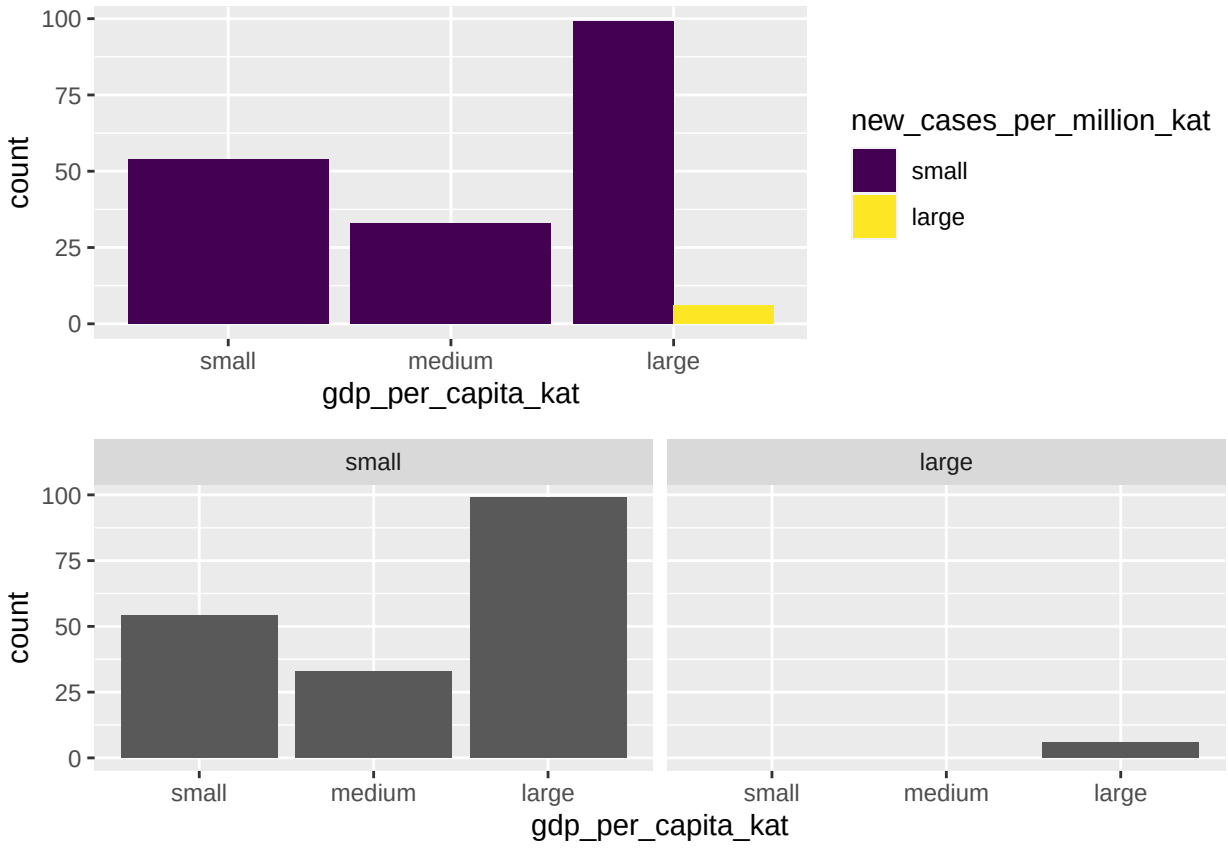


Ujabb modja a barchart segitsegevel valo megjelenitesnek ha az oszlopok nem egymasra tornyozva, hanem **egymas mellett** jelennek meg, vagy ha a masodik valtozo szerint **kulon paneleken (facet)** jelennek meg.

```
barchart_plot_5 <-
COVID_data_latest %>%
  select(new_cases_per_million_kat, gdp_per_capita_kat) %>%
  drop_na() %>%
ggplot() +
  aes(x = gdp_per_capita_kat, fill = new_cases_per_million_kat) +
  geom_bar(position = "dodge")

barchart_plot_6 <-
COVID_data_latest %>%
  select(new_cases_per_million_kat, gdp_per_capita_kat) %>%
  drop_na() %>%
ggplot() +
  aes(x = gdp_per_capita_kat) +
  geom_bar() +
  facet_wrap(~ new_cases_per_million_kat)

grid.arrange(barchart_plot_5, barchart_plot_6, nrow=2)
```



Egy kategorikus es egy numerikus valtozo kapcsolata

Vizsgaljuk meg hogy hogyan alakul az egy fore juto GDP kontinensenkent. A GDP ebben az esetben egy folytonos valtozó (gdp_per_capita), es ennek az osszefuggeset szeretnenk megvizsgalni egy kategorikus valtozoval (continent).

Az exploraciot kezdhetjuk leiro statisztikak lekerdezesevel csoportonkent. Peldaul ha arra vagyunk kivancsiak, milyen a GDP atlaga es szorasa kontinensenkent, ezt megvizsgalhatjuk a **group_by()** es a **summarize()** segitsegevel.

```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  group_by(continent) %>%
    summarize(mean = mean(gdp_per_capita),
              sd = sd(gdp_per_capita))
```

```
## # A tibble: 6 x 3
##   continent       mean     sd
##   <fct>          <dbl>  <dbl>
## 1 Africa         5444.  6183.
## 2 Asia          22057. 25131.
## 3 Europe        33361. 18030.
## 4 North America 21655. 15404.
## 5 Oceania       10618. 13216.
## 6 South America 13841.  5110.
```

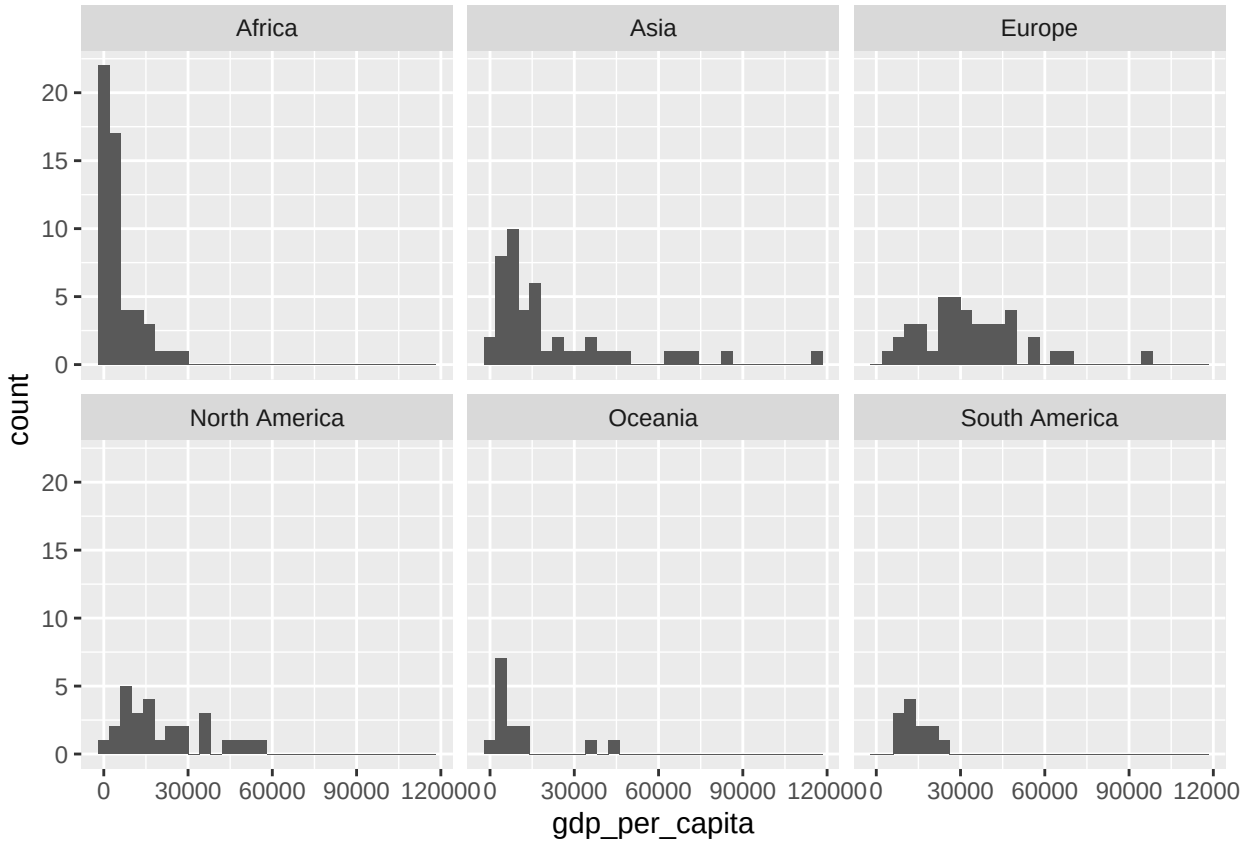
A ket valtozo kapcsolatat megvizsgalhatjuk **abrakkal** is. Pl. hasznalhatjuk a

- **facet_wrap()** fuggvenyt egy **geom_histogram()**-al kobinalva
- a **geom_boxplot()** -ot
- esetleg hasznalhatunk egy egymasra illesztett **geom_density()** plot-ot ahol a kategoriak mas mas szinnel vannak jelolve.
- 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.
- az egyik kedvencem a **geom_violin()** a **geom_jitter()**-el valo kombinacioban

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.

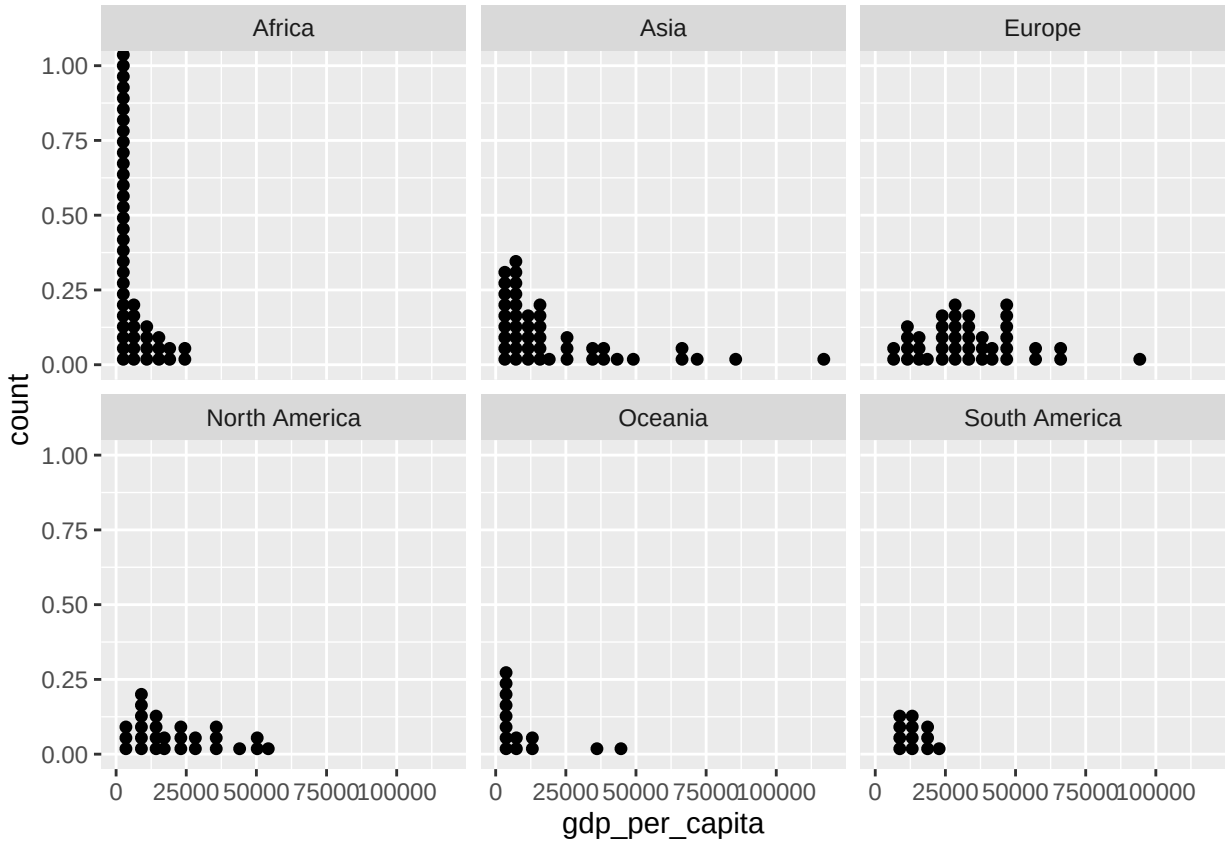
```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = gdp_per_capita) +
    geom_histogram() +
    facet_wrap(~ continent)
```

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

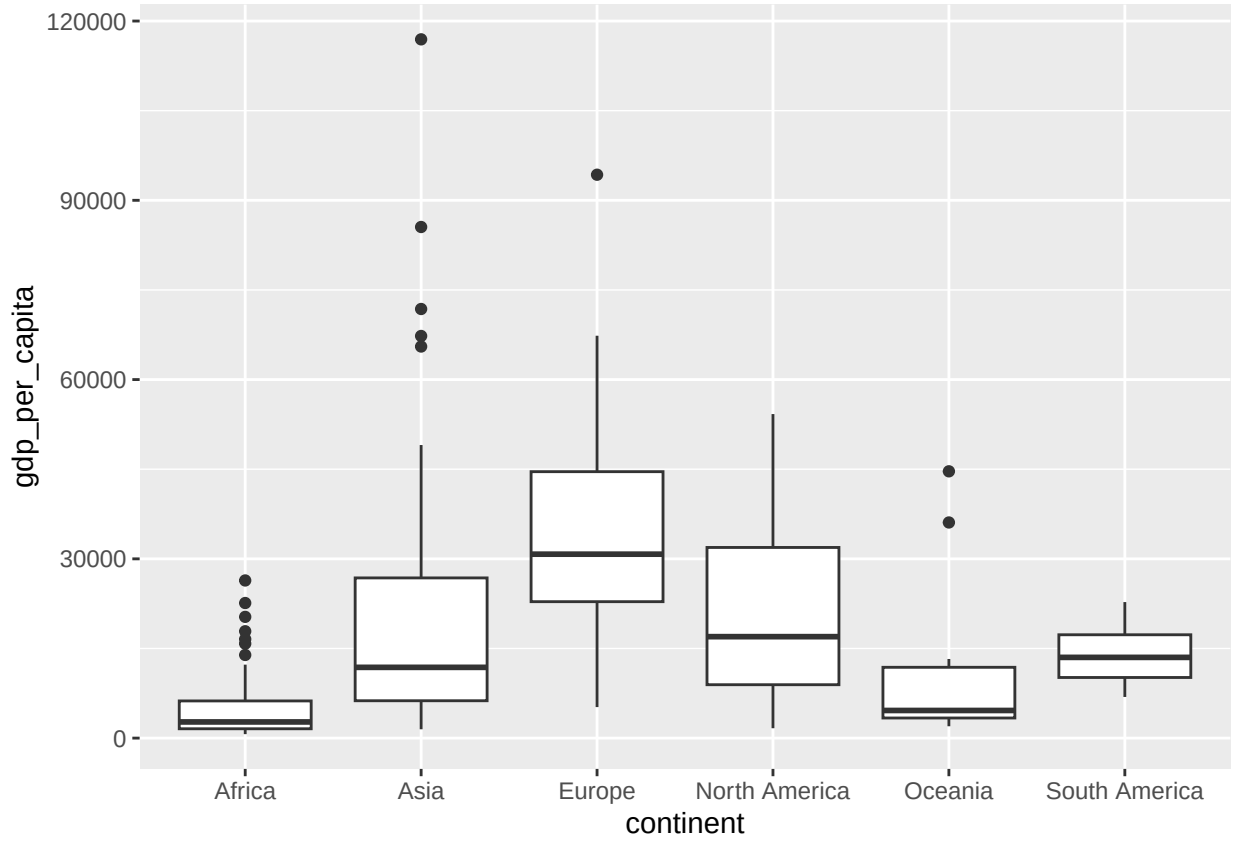


```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = gdp_per_capita) +
    geom_dotplot() +
    facet_wrap(~ continent)
```

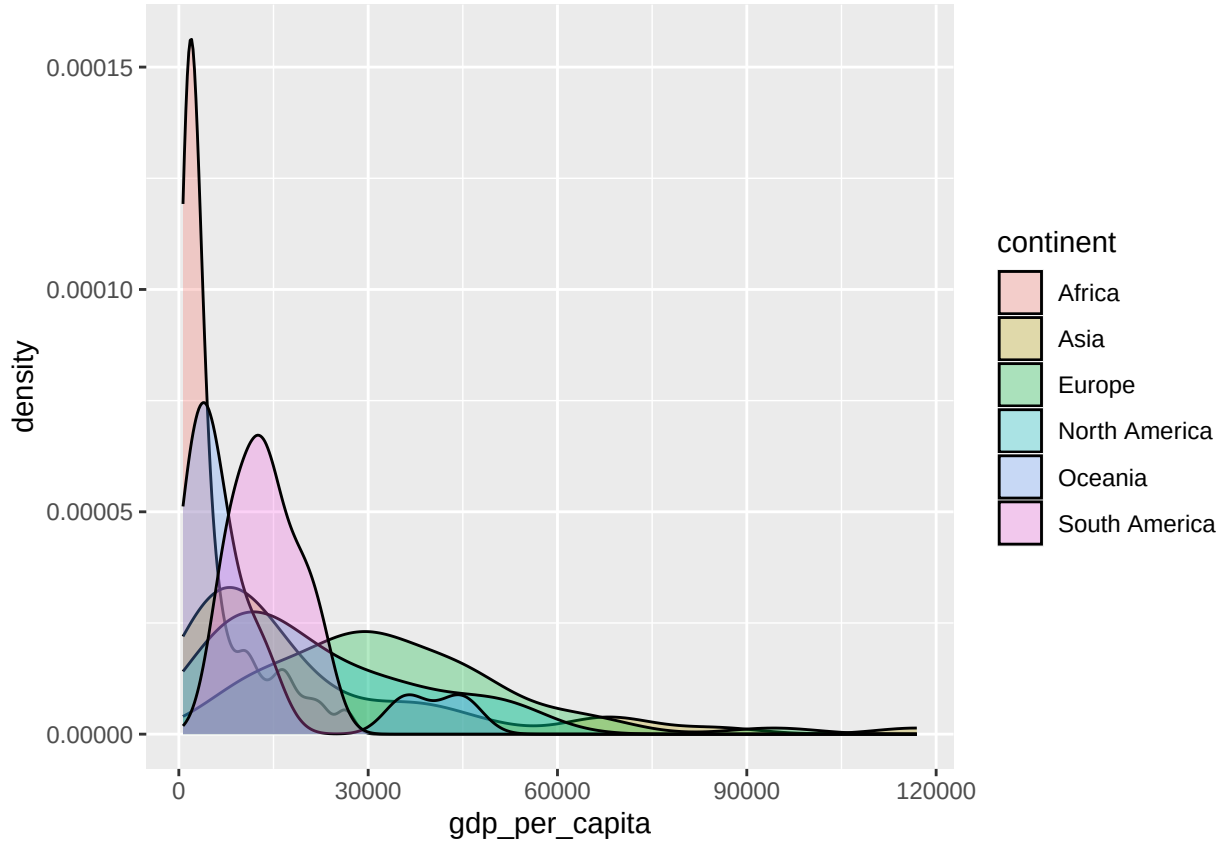
```
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```



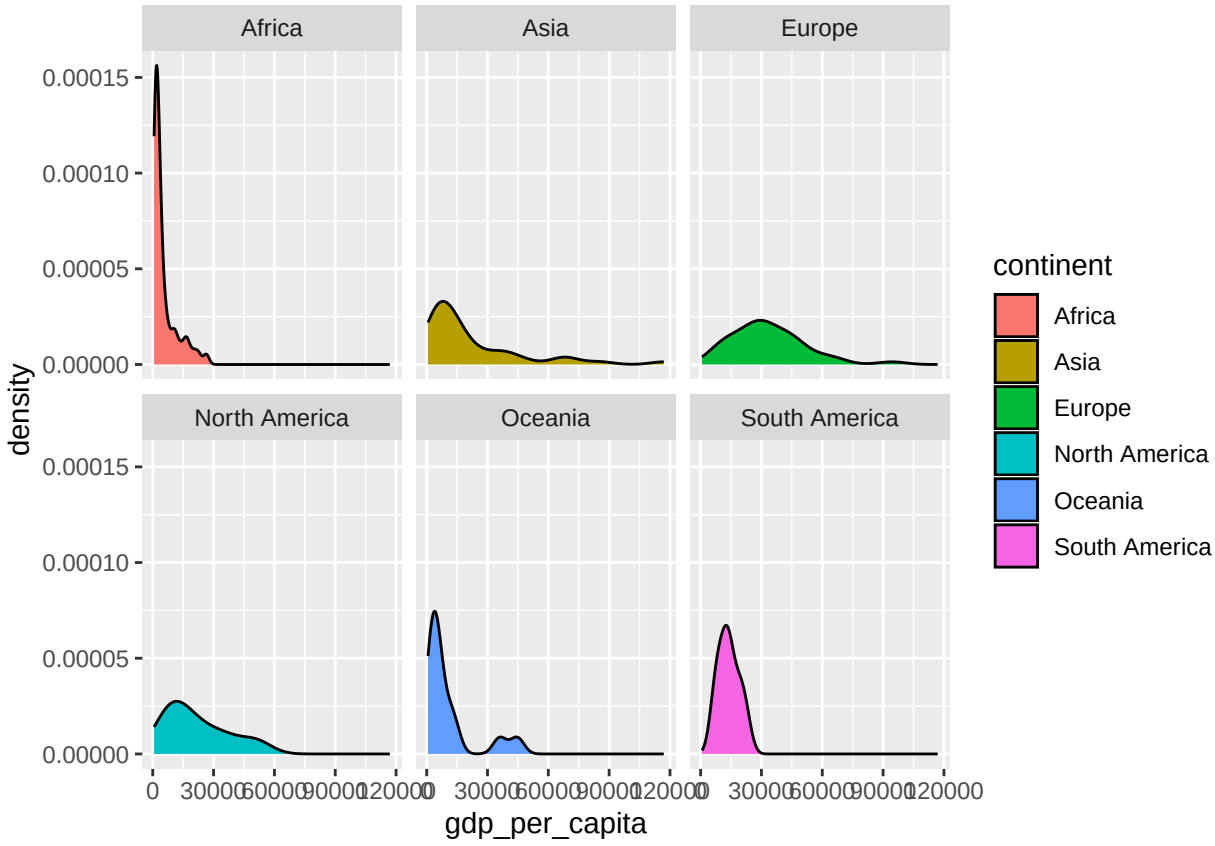

```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = continent, y = gdp_per_capita) +
    geom_boxplot()
```



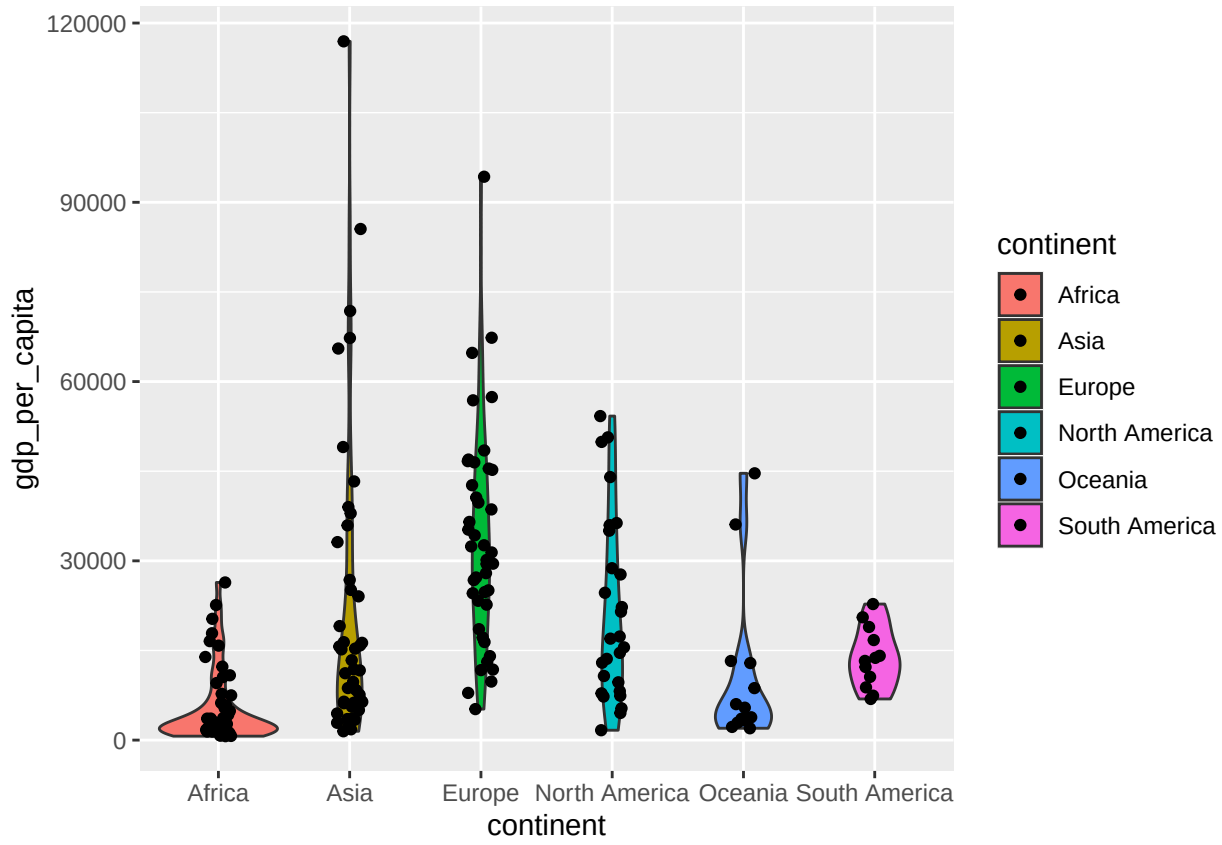

```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = gdp_per_capita, fill = continent) +
    geom_density(alpha = 0.3)
```




```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = gdp_per_capita, fill = continent) +
    geom_density()+
  facet_wrap(~continent)
```



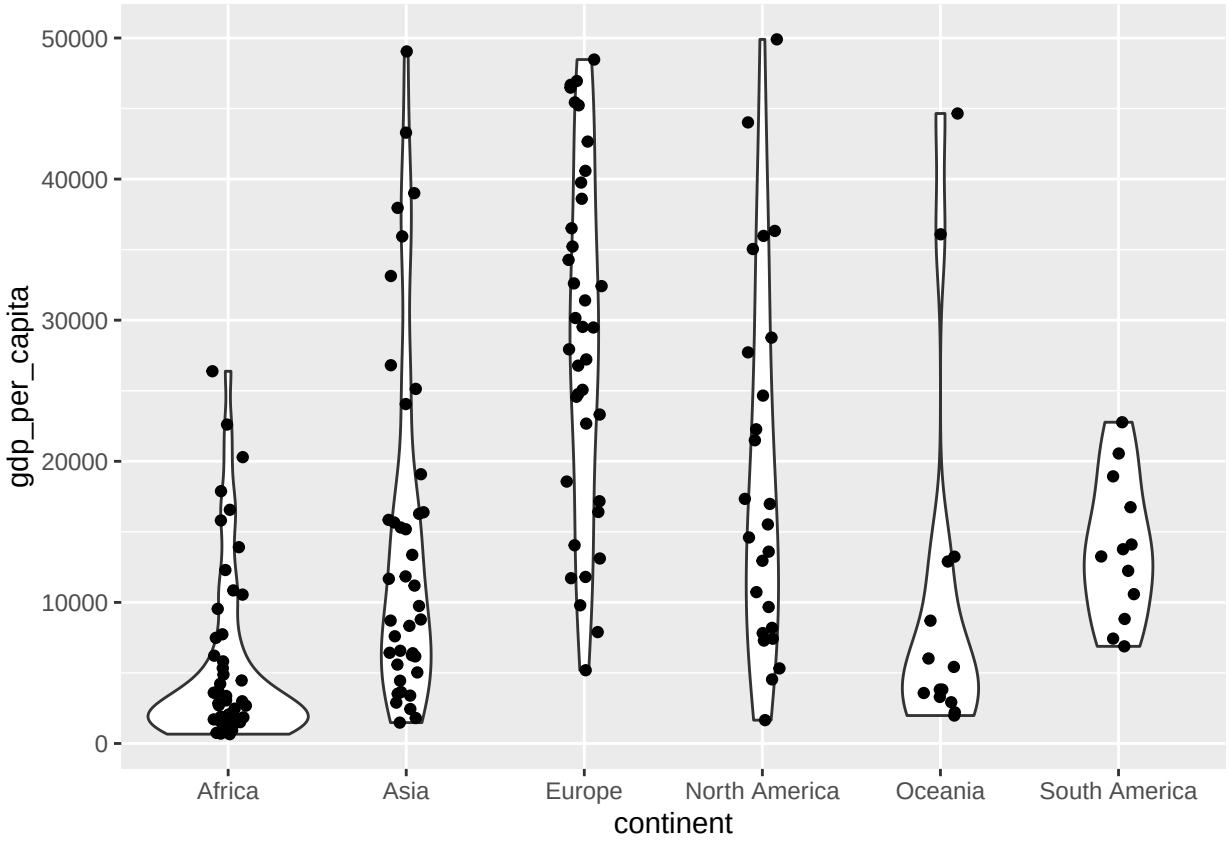

```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = continent, y = gdp_per_capita, fill = continent) +
    geom_violin() +
    geom_jitter(width = 0.1)
```



A fenti abran latszik, hogy Azsiaban a legtobb orszagban viszonylag alacsony a gdp, viszont van nehany **kiurgo ertek**, az atlagot felhuzza ebben a csoportban.

Ha szeretnenk **kizarni az elemzesunkbol** az extrem ertekekt, a **filter()** funkcio beekelesevel a pipe-ba megepithetjuk a fenti abrankat es tablazatokat ugy, hogy csak a 50000-nel alancsonyabb GDP-ju orszagok keruljenek az abrara.

```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  filter(gdp_per_capita < 50000) %>%
    ggplot() +
      aes(x = continent, y = gdp_per_capita) +
      geom_violin() +
      geom_jitter(width = 0.1)
```




```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  filter(gdp_per_capita < 50000) %>%
    group_by(continent) %>%
      summarize(mean = mean(gdp_per_capita),
                sd = sd(gdp_per_capita))
```

```
## # A tibble: 6 x 3
##   continent       mean     sd
##   <fct>          <dbl>  <dbl>
## 1 Africa         5444.  6183.
## 2 Asia          14636. 12549.
## 3 Europe        28661. 12390.
## 4 North America 19192. 13095.
## 5 Oceania       10618. 13216.
## 6 South America 13841.  5110.
```

Ha szeretnenk latni hogy a kisebb vagy nagyobb uj esetszammal jellemezheto orszagok (new_cases_per_million_kat) hogyan kulonboznek a GDP tekinteteben kontinensenkent akkor mar **harom valtozo** kapcsolatat kell abrazolnunk. Ehhez a facet_grid() funkciot lehet hasznalni, vagy kulonbozo esztetikai elemeket (aes()) lehet a kulonbozo valtozokhoz rendelni.

Gyakorlas

Hasznald a fent tanult modszereket, hogy megvizsgald a **total_cases_per_million** es a **gdp_per_capita_kat** valtozok kozotti osszefuggest.

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

---

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.

```
COVID_data_latest %>%
  select(new_cases_per_million, gdp_per_capita) %>%
  drop_na() %>%
      cor()
```

```
##                       new_cases_per_million gdp_per_capita
## new_cases_per_million              1.000000       0.216336
## gdp_per_capita                     0.216336       1.000000
```

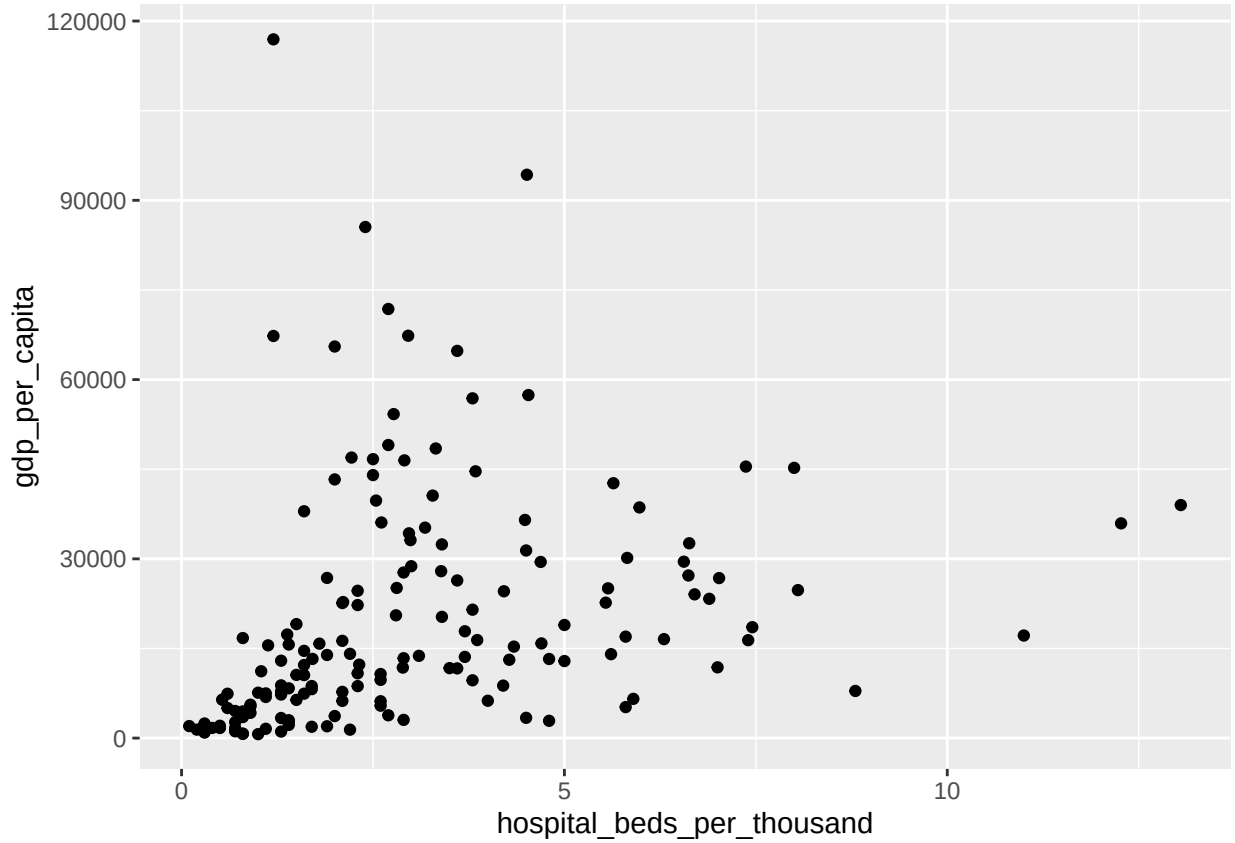
```
COVID_data_latest %>%
  select(new_cases_per_million, gdp_per_capita, hospital_beds_per_thousand) %>%
  drop_na() %>%
      cor()
```

```
##                            new_cases_per_million gdp_per_capita
## new_cases_per_million                 1.00000000      0.2175096
## gdp_per_capita                        0.21750963      1.0000000
## hospital_beds_per_thousand            0.08871957      0.2946892
##                            hospital_beds_per_thousand
## new_cases_per_million                      0.08871957
## gdp_per_capita                             0.29468918
## hospital_beds_per_thousand                 1.00000000
```

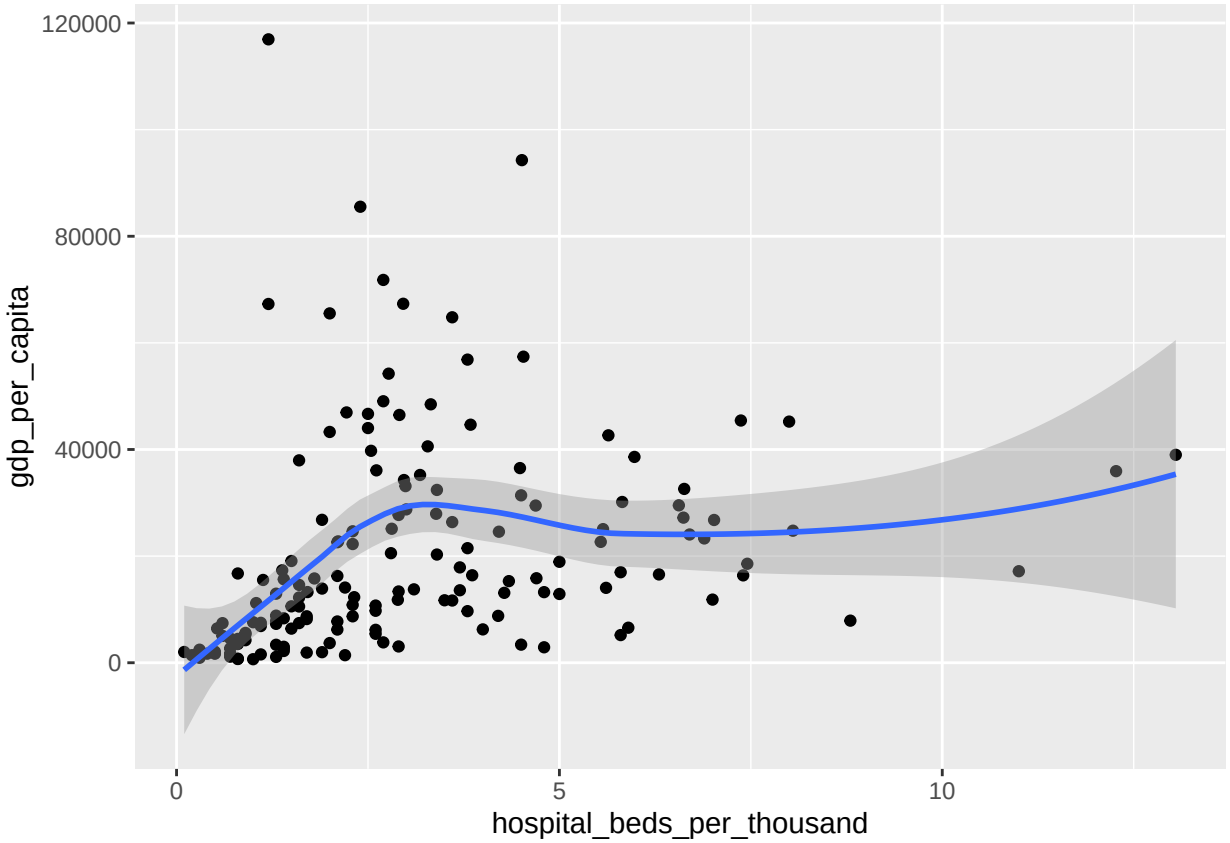
A numerikus valtozok kozotti kapcsolatot altalaban pont diagrammal szoktuk abrazolni (**geom_point()**)

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

```
COVID_data_latest %>%
  select(hospital_beds_per_thousand, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = hospital_beds_per_thousand, y = gdp_per_capita) +
    geom_point()
```




```
COVID_data_latest %>%
  select(hospital_beds_per_thousand, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = hospital_beds_per_thousand, y = gdp_per_capita) +
    geom_point() +
    geom_smooth()
```



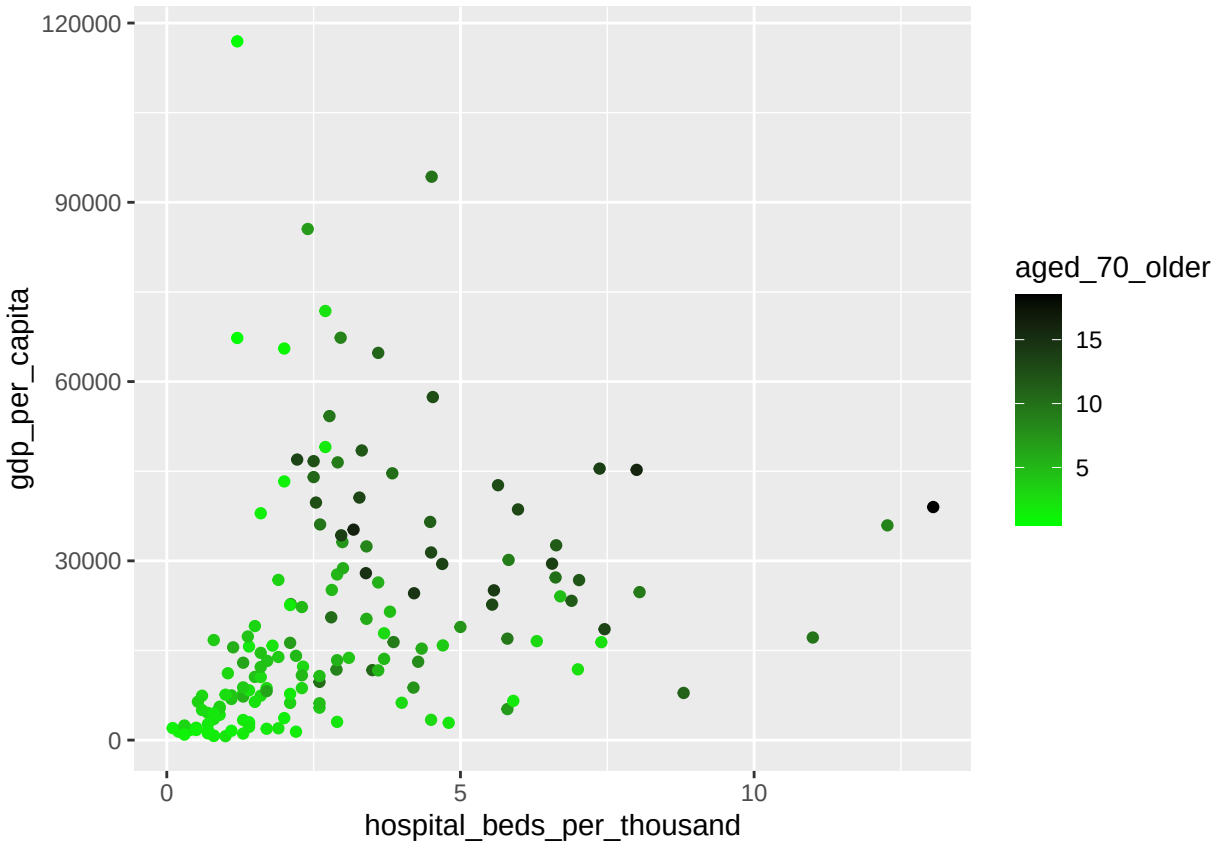
Gyakorlas

Milyen eros a kapcsolat a aged_70_older es a gdp_per_capita kozott?

- hatarozd meg a korrelacios egyutthatot a valtozok kozott
- abrazold a valtozok kapcsolatat

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

```
COVID_data_latest %>%
  select(hospital_beds_per_thousand, gdp_per_capita, aged_70_older) %>%
  drop_na() %>%
  ggplot() +
    aes(x = hospital_beds_per_thousand, y = gdp_per_capita, col = aged_70_older) +
    geom_point()+
  scale_colour_gradientn(colours=c("green","black"))
```


120000
90000
60000
30000
0
gdp_per_capita
0
5
10
hospital_beds_per_thousand
aged_70_older
15
10
5