PSZB17-210 - Seminar_4

Zoltan Kekecs

Oct 02, 2024

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 my_data nevu objektumba. A **read_csv()** funkcio a tidyverse resze, es egybol tibble formatumban menti el az adatainkat.

```
my_data <- read_csv("https://raw.githubusercontent.com/kekecsz/PSZB17-210-Data-analysis-seminar/refs/heads/master/seminar_04/StudentPerformanceFactors.csv")
```

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 …)

```
my_data
```

```
## # A tibble: 6,607 × 20
##    Hours_Studied Attendance Parental_Involvement Access_to_Resources
##            <dbl>      <dbl> <chr>                <chr>
##  1            23         84 Low                  High
##  2            19         64 Low                  Medium
##  3            24         98 Medium               Medium
##  4            29         89 Low                  Medium
##  5            19         92 Medium               Medium
##  6            19         88 Medium               Medium
##  7            29         84 Medium               Low
##  8            25         78 Low                  High
##  9            17         94 Medium               High
## 10            23         98 Medium               Medium
## # i 6,597 more rows
## # i 16 more variables: Extracurricular_Activities <chr>, Sleep_Hours <dbl>,
## #   Previous_Scores <dbl>, Motivation_Level <chr>, Internet_Access <chr>,
## #   Tutoring_Sessions <dbl>, Family_Income <chr>, Teacher_Quality <chr>,
## #   School_Type <chr>, Peer_Influence <chr>, Physical_Activity <dbl>,
## #   Learning_Disabilities <chr>, Parental_Education_Level <chr>,
## #   Distance_from_Home <chr>, Gender <chr>, Exam_Score <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)

```
my_data %>%
  select(Attendance) %>%
  summary()
```

```
##    Attendance
##  Min.   : 60.00
##  1st Qu.: 70.00
##  Median : 80.00
##  Mean   : 79.98
##  3rd Qu.: 90.00
##  Max.   :100.00
```

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

```
my_data %>%
  summary()
```

Gyakorlas

- Mi volt a legalacsonyabb részvételi arány (*Attendance*)?
- Mi az átlagos alvásmennyiség (*Sleep_Hours*) azok között a hallgatók között, akiknek alacsony a hozzáférése az erőforrásokhoz (Access_to_Resources == “Low”)?

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. Itt harom ilyen valtozot valsztok ki a select() funkcioval.

```
my_data %>%
  select(Hours_Studied, Attendance, Exam_Score) %>%
  describe()
```

```
##               vars    n  mean    sd median trimmed   mad min max range skew
## Hours_Studied    1 6607 19.98  5.99     20   19.97  5.93   1  44    43 0.01
## Attendance       2 6607 79.98 11.55     80   79.97 14.83  60 100    40 0.01
## Exam_Score       3 6607 67.24  3.89     67   67.11  2.97  55 101    46 1.64
##               kurtosis   se
## Hours_Studied     0.02 0.07
## Attendance       -1.19 0.14
## Exam_Score       10.56 0.05
```

Faktorok

Nehany karaktervaltozonak csak **korlatozott mennyisegu eleme** lehet, mint peldaul a Parental_Education_Level (ebben az adatbazisban csak College, High School, Postgraduate szinteket vesz fel). Ezeket megjelolhetjuk faktor (factor) osztalyu valtozokent, es akkor az R tobb informaciot fog adni rola.

```
table(my_data$Parental_Education_Level)
```

```
##
##      College  High School Postgraduate
##         1989         3223         1305
```

```
names(my_data)
```

```
##  [1] "Hours_Studied"              "Attendance"
##  [3] "Parental_Involvement"       "Access_to_Resources"
##  [5] "Extracurricular_Activities" "Sleep_Hours"
##  [7] "Previous_Scores"            "Motivation_Level"
##  [9] "Internet_Access"            "Tutoring_Sessions"
## [11] "Family_Income"              "Teacher_Quality"
## [13] "School_Type"                "Peer_Influence"
## [15] "Physical_Activity"          "Learning_Disabilities"
## [17] "Parental_Education_Level"   "Distance_from_Home"
## [19] "Gender"                     "Exam_Score"
```

```
my_data <- my_data %>%
             mutate(Parental_Education_Level = factor(Parental_Education_Level),
                    Parental_Involvement = factor(Parental_Involvement),
                    Access_to_Resources = factor(Access_to_Resources),
                    Extracurricular_Activities = factor(Extracurricular_Activities),
                    Motivation_Level = factor(Motivation_Level),
                    Internet_Access = factor(Internet_Access),
                    Family_Income = factor(Family_Income),
                    Teacher_Quality = factor(Teacher_Quality),
                    School_Type = factor(School_Type),
                    Peer_Influence = factor(Peer_Influence),
                    Learning_Disabilities = factor(Learning_Disabilities),
                    Distance_from_Home = factor(Distance_from_Home),
                    Gender = factor(Gender))
```

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).

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. (A table() sima karakter valtozokkal is mukodik, nem csak faktorokkal)

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

```
levels(my_data$Parental_Education_Level)
```

```
## [1] "College"      "High School"  "Postgraduate"
```

```
table(my_data$School_Type)
```

```
##
##    Home Private  Public
##       6    2006    4595
```

```
my_data %>%
  select(Gender) %>%
  summary()
```

```
##     Gender
##  Female:2793
##  Male  :3814
```

```
my_data$School_Type
```

Van, hogy szeretnenk **kizarni** bizonyos **faktorszinteket** az elemzesbol. Pl. ha valamelyik faktor szintbol nagyon keves megfigyeles van, vagy csak a kutatasi kerdesunk nem vonatkozik az adott reszere a populacionak.

Az alabbi peldaban a School_Type valtozobol kizarjuk a “Home” szintet, vagyis azokat a valaszadokat akik otthoni iskolaba jarnak.

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.

```
levels(my_data$School_Type)
```

```
## [1] "Home"    "Private" "Public"
```

```
my_data %>%
  filter(School_Type != "Home") %>%
  select(Exam_Score, School_Type) %>%
  summary()
```

```
##    Exam_Score      School_Type
##  Min.   : 55.00   Home   :   0
##  1st Qu.: 65.00   Private:2006
##  Median : 67.00   Public :4595
##  Mean   : 67.23
##  3rd Qu.: 69.00
##  Max.   :101.00
```

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

```
my_data_noHomeSchooled = my_data %>%
  filter(School_Type != "Home") %>%
  mutate(School_Type = droplevels(School_Type))

my_data_noHomeSchooled %>%
  select(Exam_Score, School_Type) %>%
  summary()
```

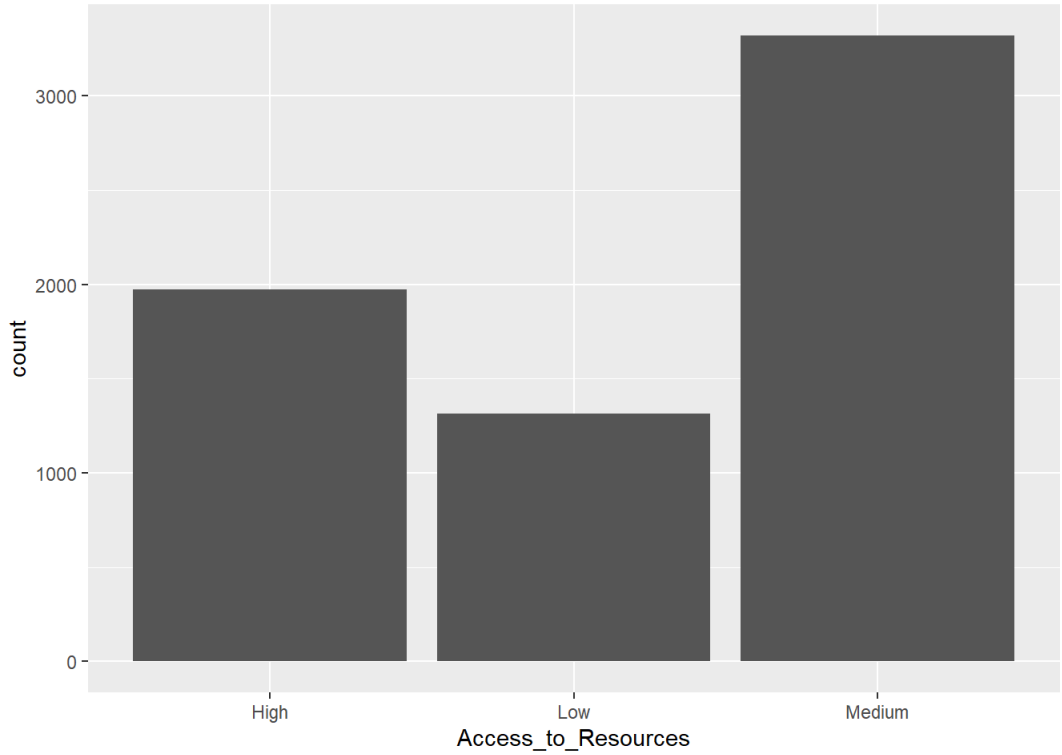
```
##    Exam_Score      School_Type
##  Min.   : 55.00   Private:2006
##  1st Qu.: 65.00   Public :4595
##  Median : 67.00
##  Mean   : 67.23
##  3rd Qu.: 69.00
##  Max.   :101.00
```

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 vegzettsege nelkuli < altalanos iskolai < kozepiskolai < felsofoku … Itt faktorszinteknek van egy meghatarozott hierarchiaja, vagy sorrendje. Ebben az adatbazisban sok ilyen valtozo van, pl. Access_to_Resources.

Amikor abrat rajzolunk errol a valtozorol, lathatjuk hogy a faktorszintek sorrendje “High”, “Low”, “Medium”.

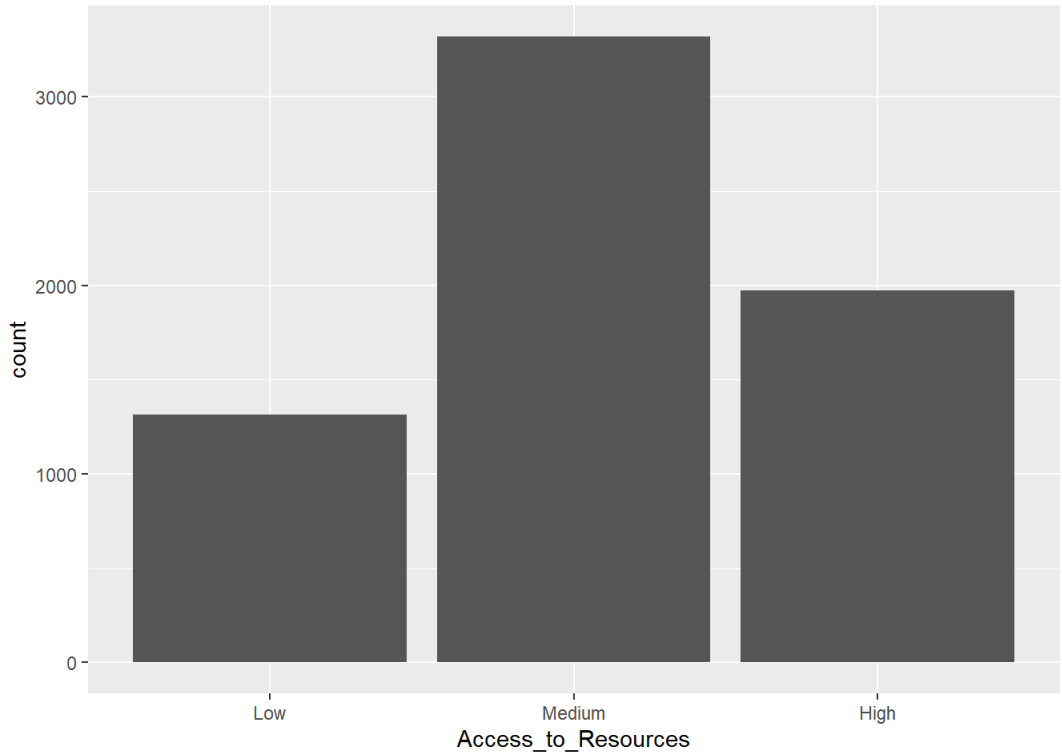
```
my_data %>%
  ggplot() +
  aes(x = Access_to_Resources) +
  geom_bar()
```



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.

```
my_data = my_data %>%
mutate(Access_to_Resources = factor(Access_to_Resources, levels = c(
                                          "Low",
                                           "Medium",
                                           "High")))
my_data %>%
  ggplot() +
  aes(x = Access_to_Resources) +
  geom_bar()
```



Attol meg hogy megadjuk a levels-el a faktorszintek listazasi sorrendjet, az R meg mindig egyenrangukent kezeli a faktorszinteket, csak most mar jo sorrendben irja ki oket.

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.

```
my_data = my_data %>%
mutate(Access_to_Resources = factor(Access_to_Resources, ordered = T, levels = c(
                                          "Low",
                                           "Medium",
                                           "High")))

head(my_data$Access_to_Resources)
```

```
## [1] High   Medium Medium Medium Medium Medium
## Levels: Low < Medium < High
```

Kategorikus valtozo letrehozasa es ujrakodolasa

Ha egy folytonos valtozo alapjan szeretnenk egy kategorikus valtozot letrehozni, hasznalhatjuk a **mutate()** es **case_when()** funkciok kombinaciojat hogy csinaljunk egy uj valtozot.

Mondjuk a vizsgan elert szazalek alapjan hozzunk letre ertekelesi csoportkat.

Ebbe a kodba beleepitettem a **factor()** funkciot is, hogy azonnal meghatarozzuk, hogy ez az uj valtozo egy faktor, es hogy ordinalis valtozo, hiszen a kulonbozo szinteknek van ertek-relacioja.

```
my_data = my_data %>%
  mutate(Grade = factor(
    case_when(Exam_Score < 60 ~ "Poor",
              Exam_Score >= 60 & Exam_Score < 80 ~ "Good",
              Exam_Score > 80 ~ "Excellent"), levels = c("Poor", "Good", "Excellent"), ordered = T))

levels(my_data$Grade)
```

```
## [1] "Poor"      "Good"      "Excellent"
```

Egy masik funkcio amivel manipulalhatjuk a faktorszinteket, a **recode()**. Ha kategorikus valtozokat szeretnenk atkodolni, mondjuk ha szeretnenk az iment letrehozott Grade valtozo alapjan egy ujrakodolt uj valozot letrehozni, azt a kovetkezo keppen thehetjuk:

```
my_data = my_data %>%
  mutate(Grade_passfail = factor(recode(Grade,
                                        "Poor" = "Failed",
                                        "Good" = "Passed",
                                        "Excellent" = "Passed")))

levels(my_data$Grade_passfail)
```

```
## [1] "Failed" "Passed"
```

Gyakorlas

- szurd az adatokat ugy hogy ne legyenek benne az otthon tanulo (School_Type == “Home”) hallgatok.
- csinalj egy uj kategorikus valtozot (nevezzuk ezt *Sleep_Categorical*-nak) a mutate() funkcio hasznalataval amiben azok az orszagok ahol a *Sleep_Hours* valtozo 6 alatt van “inadequate”, ahol 6 vagy a felett van “adequate” 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 *Sleep_Categorical* egyes kategoriaiba.
- Add meg a faktorszintek helyes sorrendjet: az “inadequate” szint legyen elorebb sorolva, mint az “adequate” szint (Ird felul a *Sleep_Categorical* 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

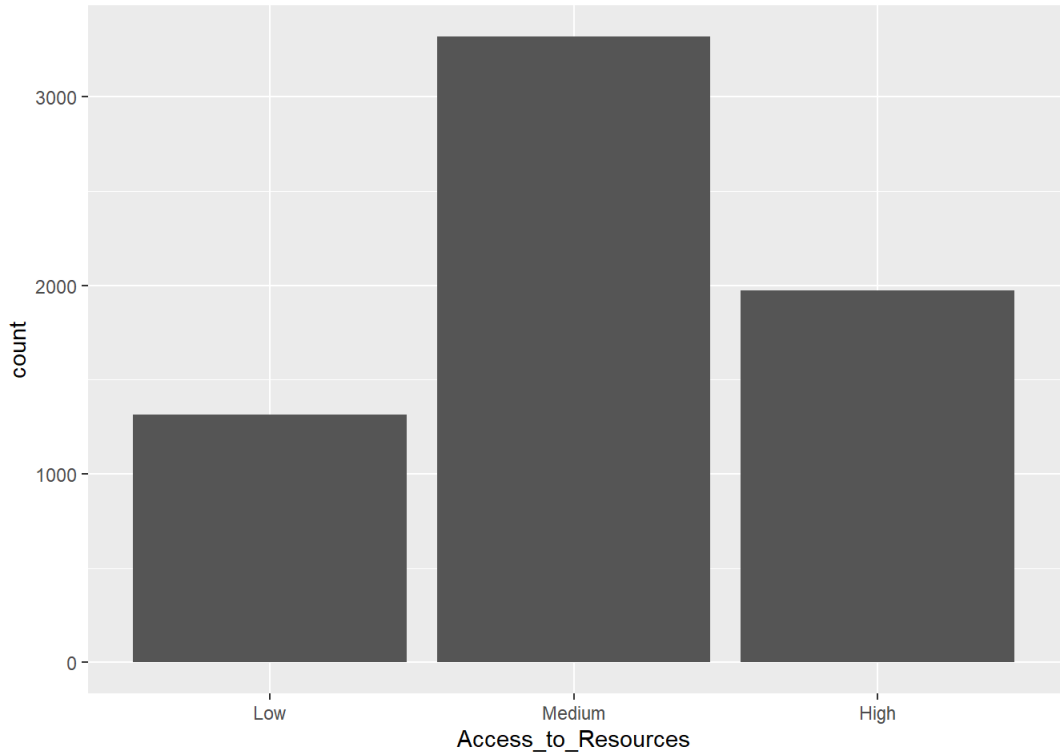
Az egyes valtozok vizualizacioja es a leiro statisztikak atvizsgalasa elengedhetetlen hogy azonositsuk az esetleges adatbeviteli **hibakat es egyeb nemvart furcsasagokat** az adataink kozott.

Egyes valtozok vizualizacioja

Az egyes valtozok peldaul **abrak** segitsegevel megvizsgalhatok.

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

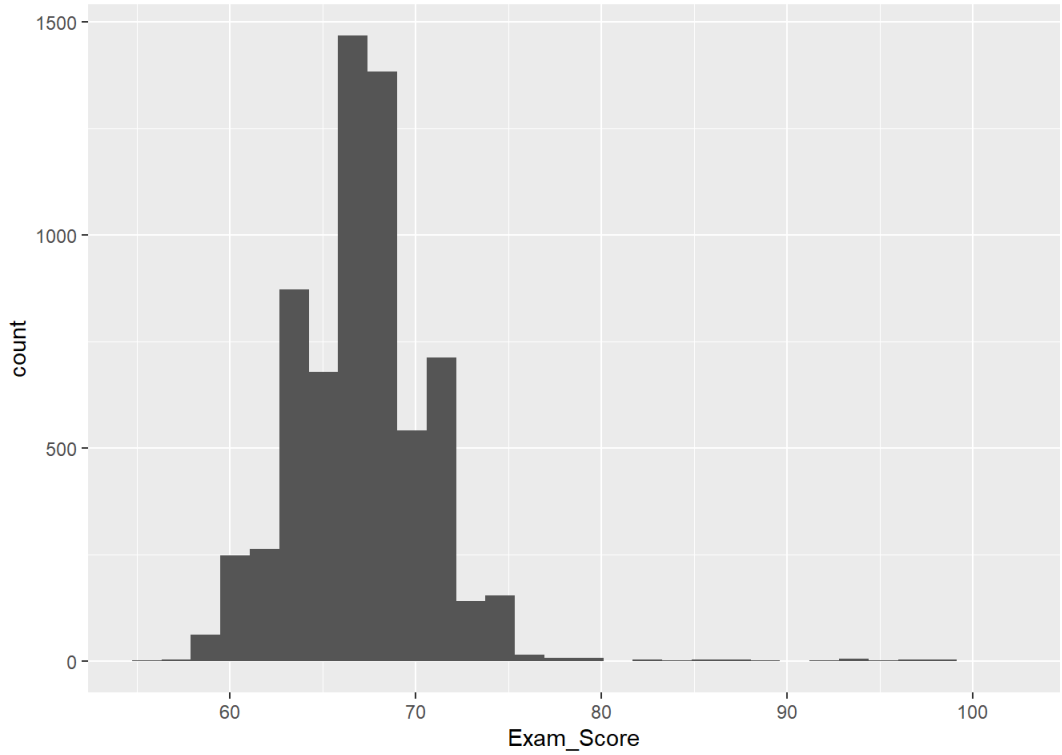
```
my_data %>%
ggplot() +
  aes(x = Access_to_Resources) +
  geom_bar()
```



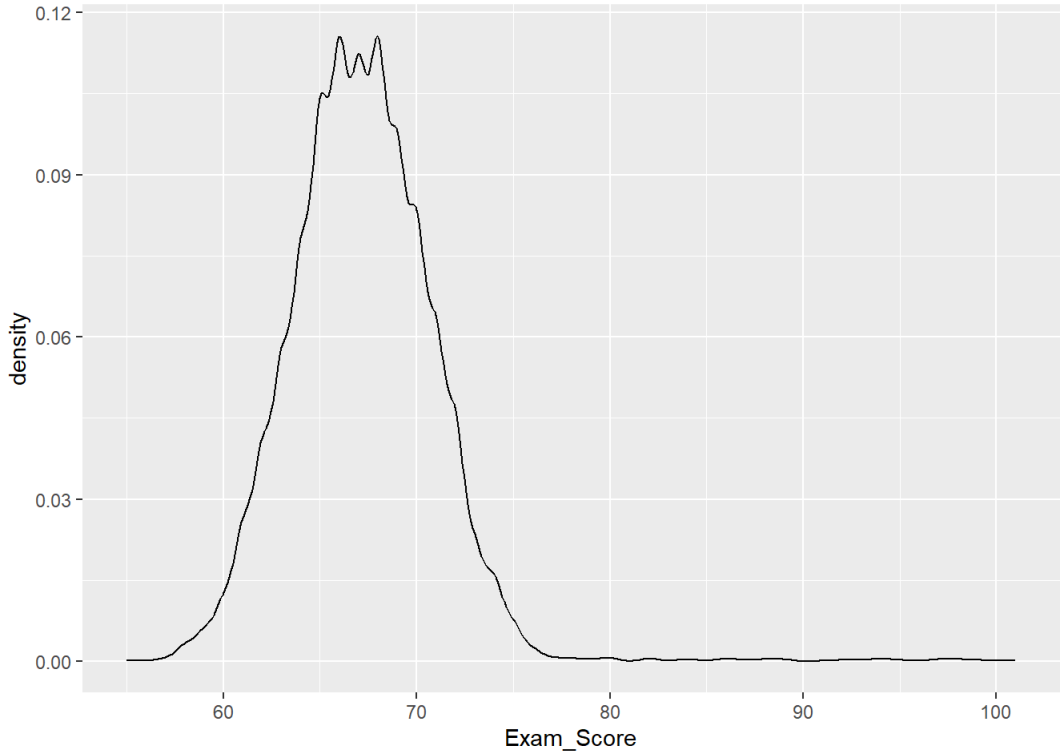
A **folytonos** valtozokat gyakran histogrammal (**geom_histogram**) vagy surusegabraval (**geom_density**) abrazoljuk,

```
my_data %>%
ggplot() +
  aes(x = Exam_Score) +
  geom_histogram()
```

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




```
my_data %>%
ggplot() +
  aes(x = Exam_Score) +
  geom_density()
```



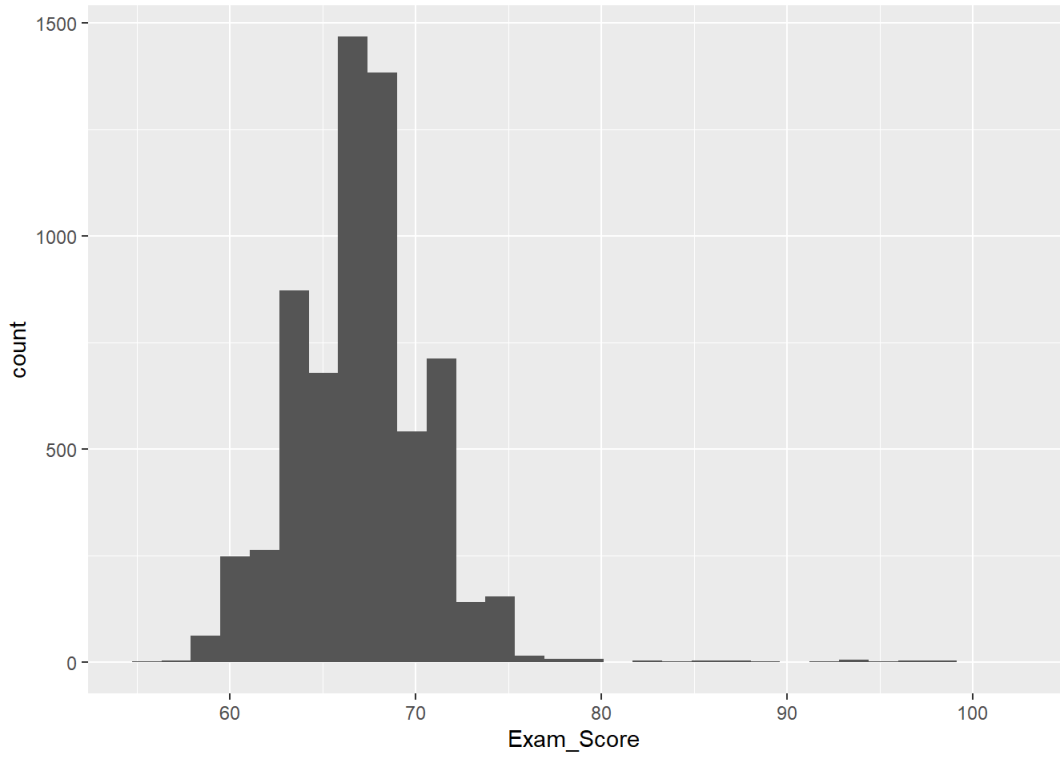
Hibaellenorzes

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

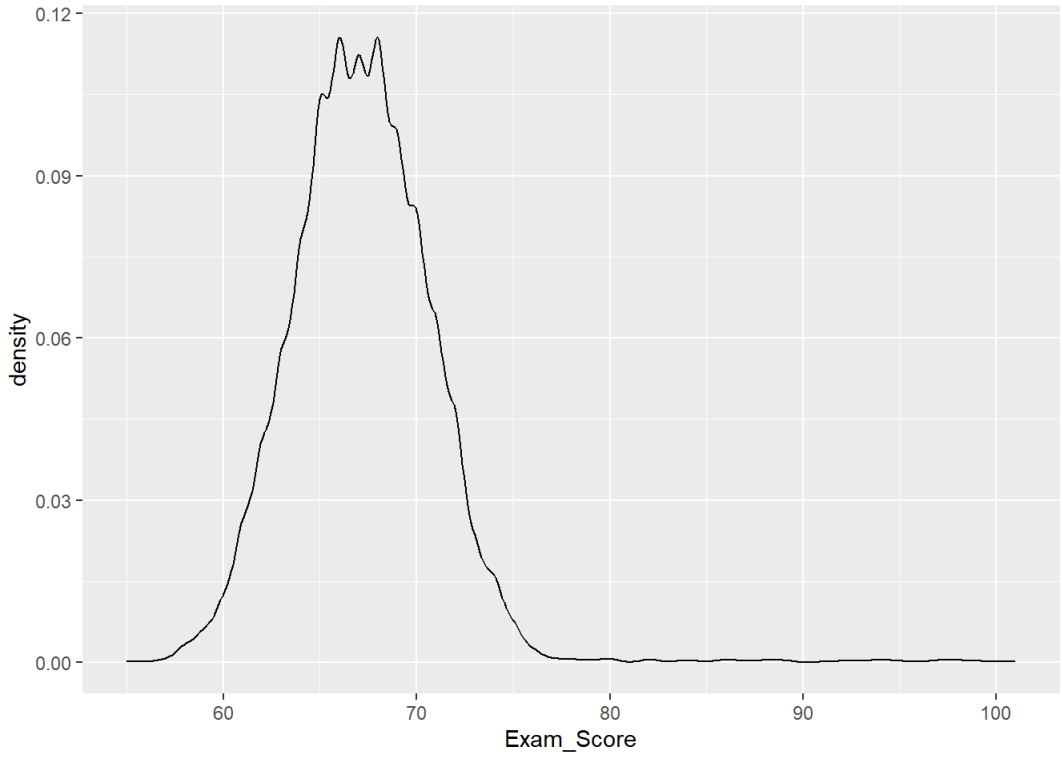
Ehhez hasznalhatsz mind adatvizualizaciot, mind a fentebb tanult leiro statisztikat (summarize(), summary(), describe() funkciokkal).

```
my_data %>%
ggplot() +
  aes(x = Exam_Score) +
  geom_histogram()
```

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




```
my_data %>%
ggplot() +
  aes(x = Exam_Score) +
  geom_density()
```



```
my_data %>%
  select(Exam_Score) %>%
  summary()
```

```
##    Exam_Score
##  Min.   : 55.00
##  1st Qu.: 65.00
##  Median : 67.00
##  Mean   : 67.24
##  3rd Qu.: 69.00
##  Max.   :101.00
```

Gyakorlas

Szurd az adatokat ugy hogy csak a Previous_Scores, Peer_Influence, Parental_Involvement, es a Sleep_Hours valtozokkal dolgozz.

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

- A vizualizacion tul a View(), describe(), table(), 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, esetleg a felvett ertekek mennyiseget, ha nincs tul sok felveheto ertek.
- 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 ertek, 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 a my_data_corrected nevet adtuk a javitott objektumnak. Igy a nyers adataink megmaradnak, ami hasznos lehet kesobbi muveleteknel.

```
my_data %>%
  select(Exam_Score) %>%
  summary()
```

```
##    Exam_Score
##  Min.   : 55.00
##  1st Qu.: 65.00
##  Median : 67.00
##  Mean   : 67.24
##  3rd Qu.: 69.00
##  Max.   :101.00
```

```
my_data_corrected <- my_data %>%
  mutate(Exam_Score = replace(Exam_Score,  Exam_Score=="101", NA))

my_data_corrected %>%
  select(Exam_Score) %>%
  summary()
```

```
##    Exam_Score
##  Min.   : 55.00
##  1st Qu.: 65.00
##  Median : 67.00
##  Mean   : 67.23
##  3rd Qu.: 69.00
##  Max.   :100.00
##  NA's   :1
```

Erdemes **megbizonyosodni rola**, hogy az adatcsere sikeres volt, az uj javitott adat vizualizaciojaval, vagy a leiro statisztikak lekerdezesevel.

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 a csalad anyagi helyzete (*Family_Income*) milyen osszefuggest mutat az interneteleressel. (*Internet_Access*).

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

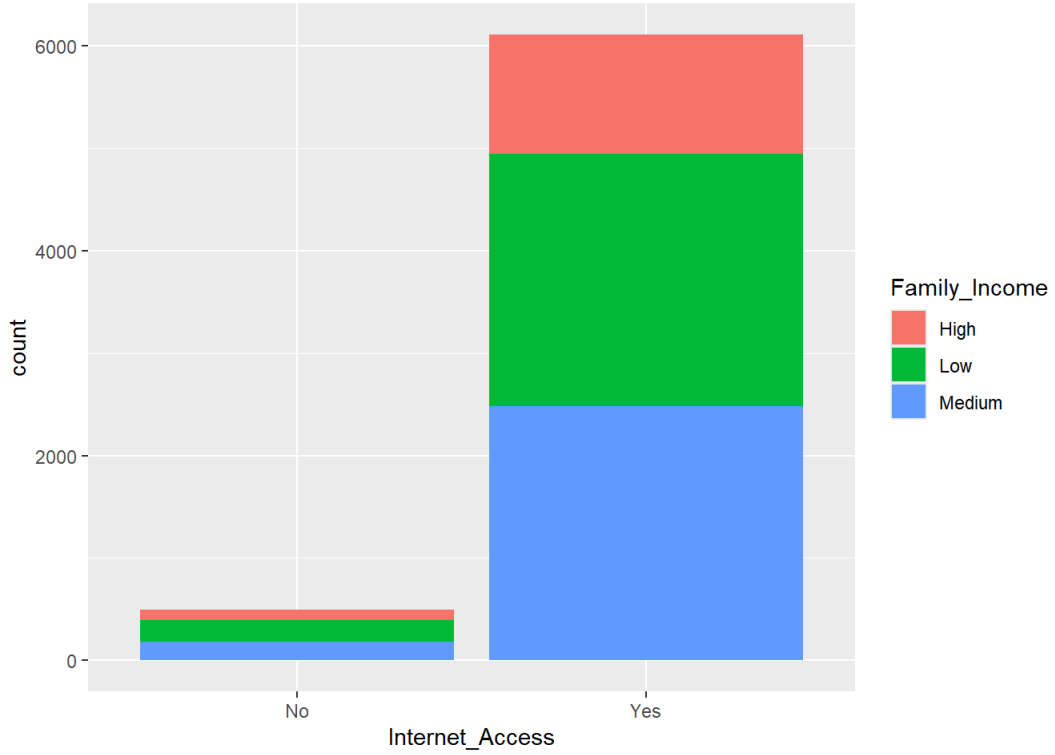
```
table(my_data_corrected$Family_Income, my_data_corrected$Internet_Access)
```

```
##
##             No  Yes
##   High     102 1167
##   Low      211 2461
##   Medium   186 2480
```

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

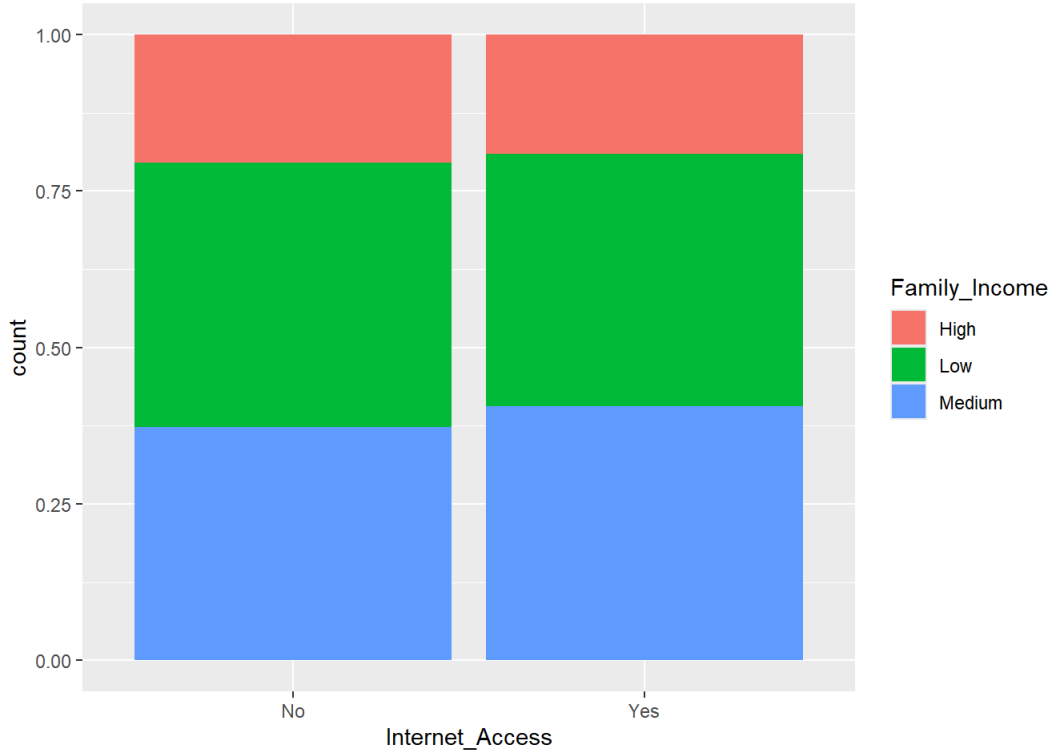
Erre az egyik lehetoseg a **stacked bar chart** (egymasra tornyozott oszlopdiagram, a **geom_bar()** geomot hasznaljuk) hasznalata. Itt az egyik valtozo kategoriai adjak meg hany oszlop lesz (ez a valtozo lesz az x tengelyen reprezentalva, igy ezt az “x =” reszen adhatjuk meg), a masik valtozo az oszlopokat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

```
my_data_corrected %>%
  drop_na(Family_Income) %>%
ggplot() +
  aes(x = Internet_Access, fill = Family_Income) +
  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)

```
my_data_corrected %>%
  drop_na(Family_Income) %>%
ggplot() +
  aes(x = Internet_Access, fill = Family_Income) +
  geom_bar(position = "fill")
```



Gyakorlas

Hasznald a fent tanult modszereket, hogy megvizsgald a my_data_corrected adatbazisban a **Sleep_Categorical** es a **Distance_from_Home** 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 Sleep_Categorical valtozot igy lehet legeneralni:

my_data_corrected = my_data_corrected %>%
  mutate(Sleep_Categorical = factor(
                                    case_when(Sleep_Hours < 6 ~ "inadequate",
                                              Sleep_Hours >= 6 ~ "adequate"), ordered = T, levels = c("inadequate",
"adequate")))

levels(my_data_corrected$Sleep_Categorical)
```

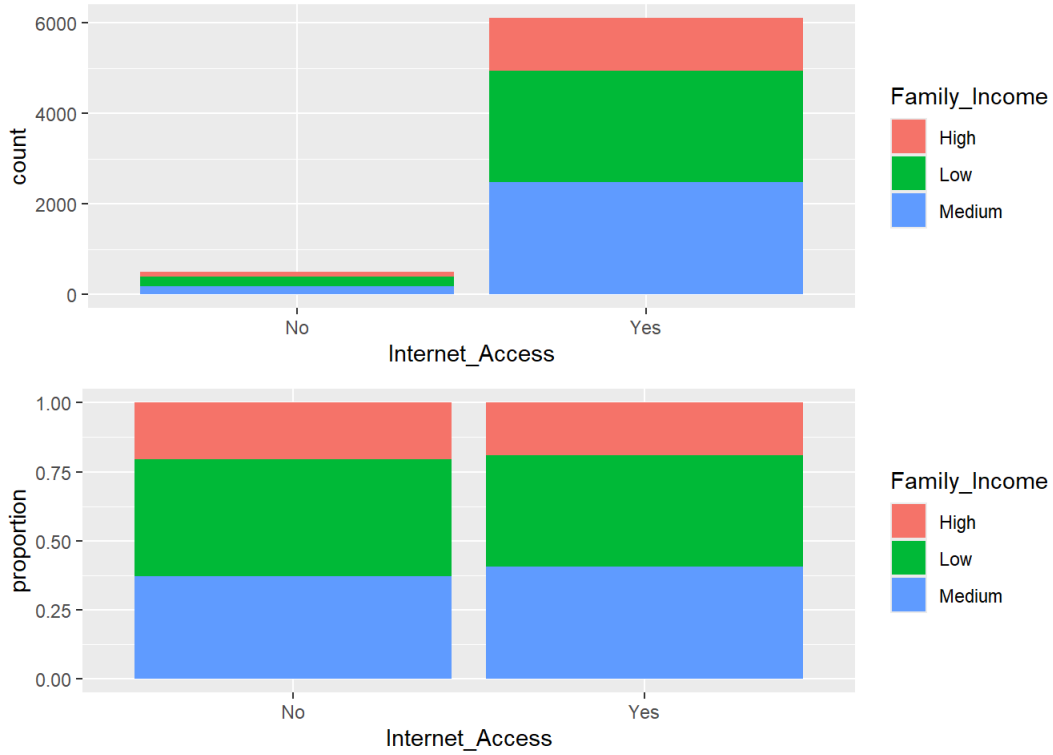
```
## [1] "inadequate" "adequate"
```

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 <-
my_data_corrected %>%
  drop_na(Family_Income) %>%
ggplot() +
  aes(x = Internet_Access, fill = Family_Income) +
  geom_bar()

reszarany_plot <-
my_data_corrected %>%
  drop_na(Family_Income) %>%
ggplot() +
  aes(x = Internet_Access, fill = Family_Income) +
  geom_bar(position = "fill") +
  ylab("proportion")

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



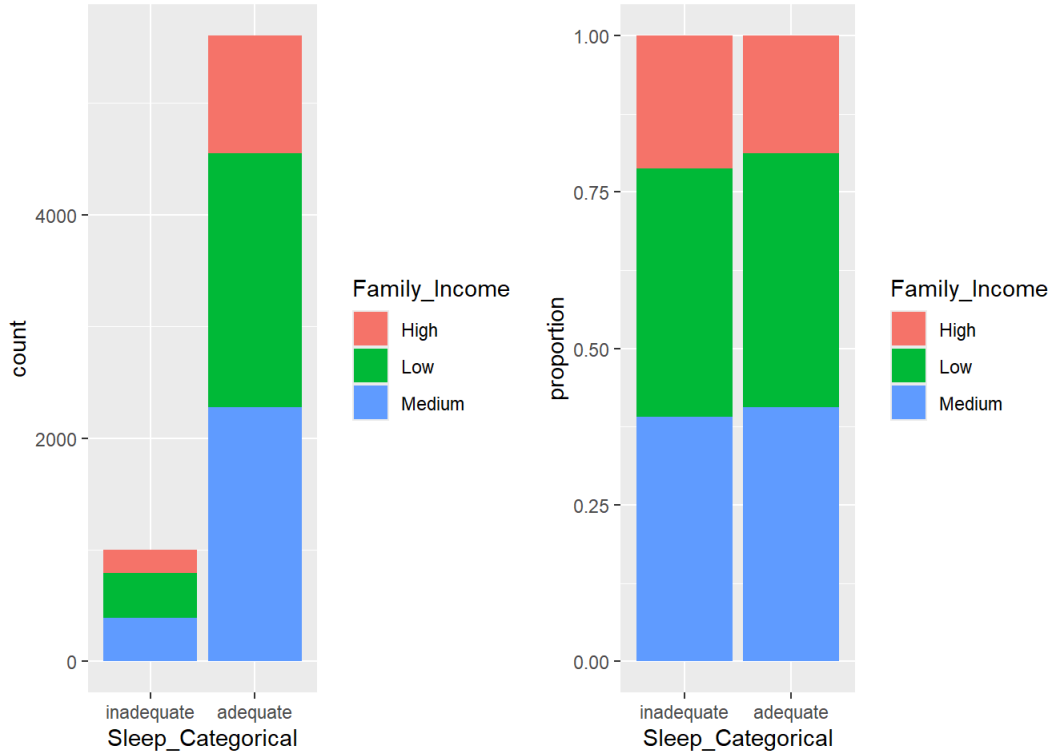
A theme(legend.position) es a guides() funciok hasznalataval kontrollalhatjuk hogy hol es hogyan jelenjen meg a **jelmagyarazat** az abran. Az abra **interpretalhatosaga** attol fuggoen is **valtozhat**, hogy melyik valtozot tesszuk az x-tengelyre es melyiket szinkent abrazolva.

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 <-
my_data_corrected %>%
  select(Sleep_Categorical, Family_Income) %>%
  drop_na() %>%
ggplot() +
  aes(x = Sleep_Categorical, fill = Family_Income) +
  geom_bar()


barchart_plot_4 <-
my_data_corrected %>%
  select(Sleep_Categorical, Family_Income) %>%
  drop_na() %>%
ggplot() +
  aes(x = Sleep_Categorical, fill = Family_Income) +
  geom_bar(position = "fill") +
  ylab("proportion")

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

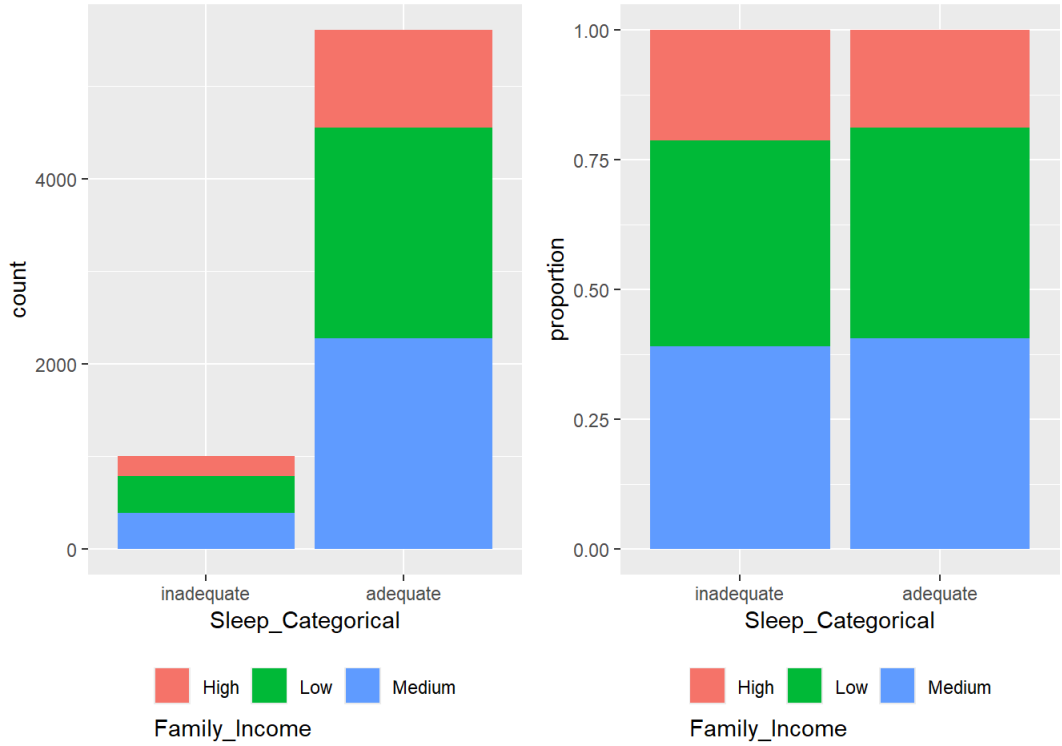



```
# a theme(legend.position) es a guides() funciok
# hasznalataval kontrollalhatjuk hogy hol es hogyan
# jelenjen meg a jelmagyarazat az abran

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


barchart_plot_4 <-
my_data_corrected %>%
  select(Sleep_Categorical, Family_Income) %>%
  drop_na() %>%
ggplot() +
  aes(x = Sleep_Categorical, fill = Family_Income) +
  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)
```

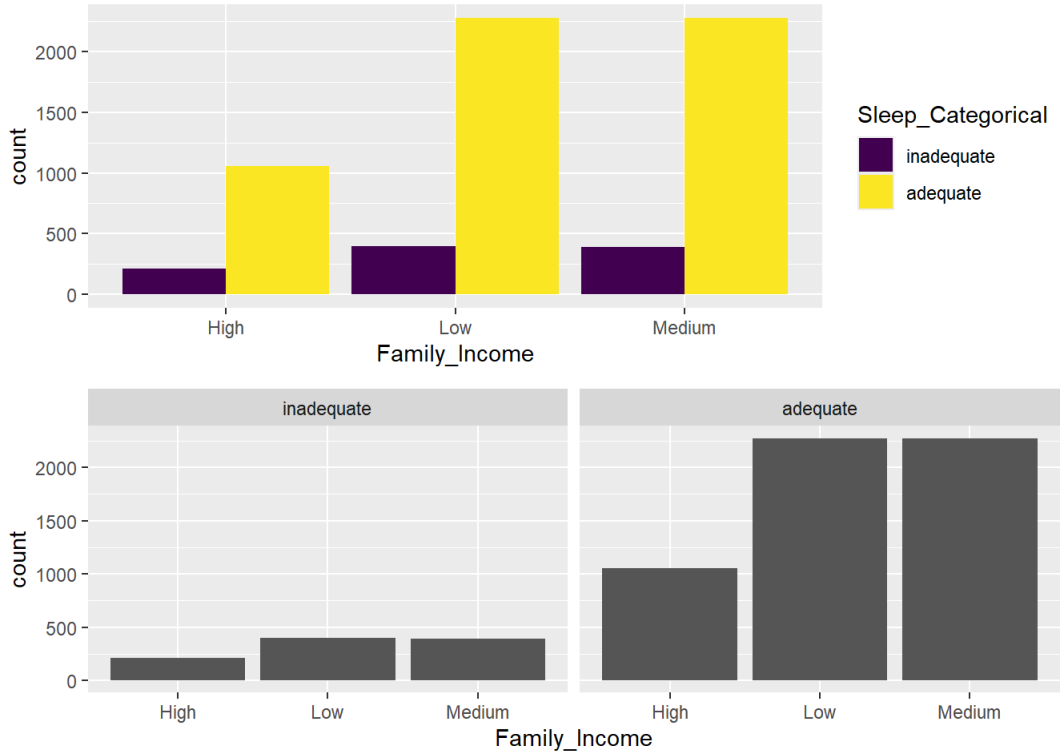


Ujabb modja a barchart segitsegevel valo megjelenitesnek ha az oszlopok nem egymasra tornyozva, hanem **egymas mellett** jelennek meg, vagy ha a masodik valtozo szerint **kulon paneleken (facet)** jelennek meg.

```
barchart_plot_5 <-
my_data_corrected %>%
  select(Sleep_Categorical, Family_Income) %>%
  drop_na() %>%
ggplot() +
  aes(x = Family_Income, fill = Sleep_Categorical) +
  geom_bar(position = "dodge")

barchart_plot_6 <-
my_data_corrected %>%
  select(Sleep_Categorical, Family_Income) %>%
  drop_na() %>%
ggplot() +
  aes(x = Family_Income) +
  geom_bar() +
  facet_wrap(~ Sleep_Categorical)

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



Egy kategorikus es egy numerikus valtozo kapcsolata

Vizsgaljuk meg hogy hogyan alakul a vizsgateljesitmeny (Exam_Score) attol fuggoen hogy milyen a szuloi bevonodas (Parental_Involvement). Az Exam_Score egy folytonos numerikus valtozo, mig a Parental_Involvement kategorikus valtozo.

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.

```
my_data_corrected = my_data_corrected %>%
  mutate(Parental_Involvement = factor(Parental_Involvement, ordered = T, levels = c("Low", "Medium", "High")))

my_data_corrected %>%
  select(Parental_Involvement, Exam_Score) %>%
  drop_na() %>%
  group_by(Parental_Involvement) %>%
    summarize(mean = mean(Exam_Score),
              sd = sd(Exam_Score))
```

```
## # A tibble: 3 × 3
##   Parental_Involvement  mean    sd
##   <ord>                <dbl> <dbl>
## 1 Low                   66.3  3.86
## 2 Medium                67.1  3.73
## 3 High                  68.1  3.95
```

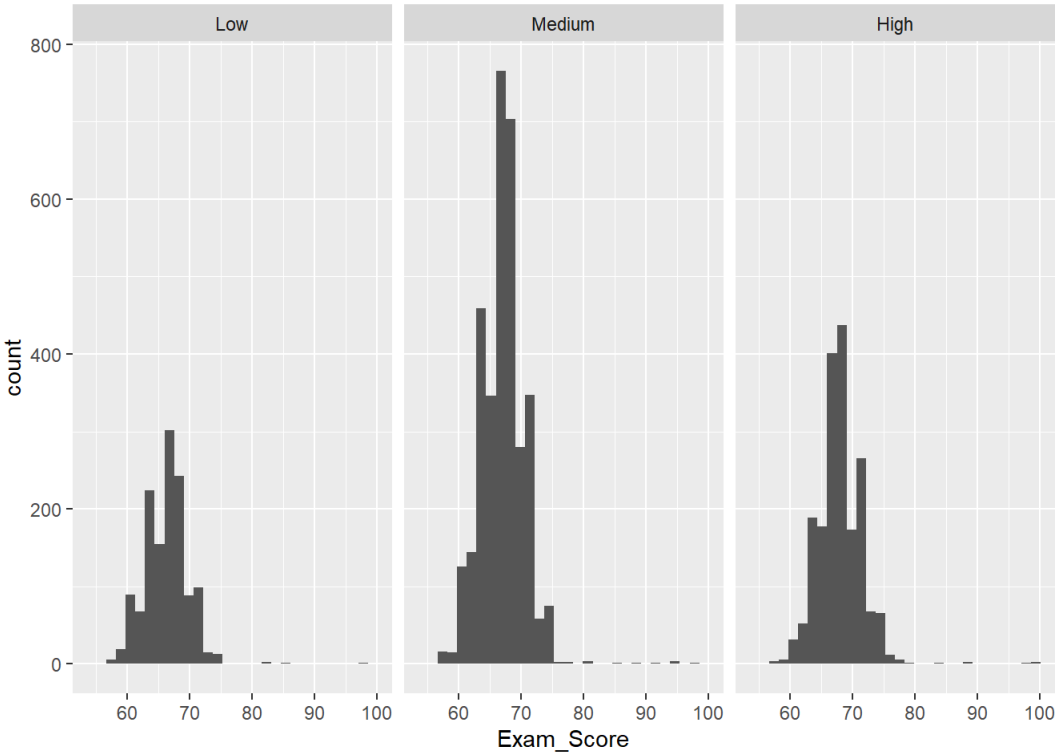
A ket valtozo kapcsolatat megvizsgalhatjuk **abrakkal** is. Pl. hasznalhatjuk a

- **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

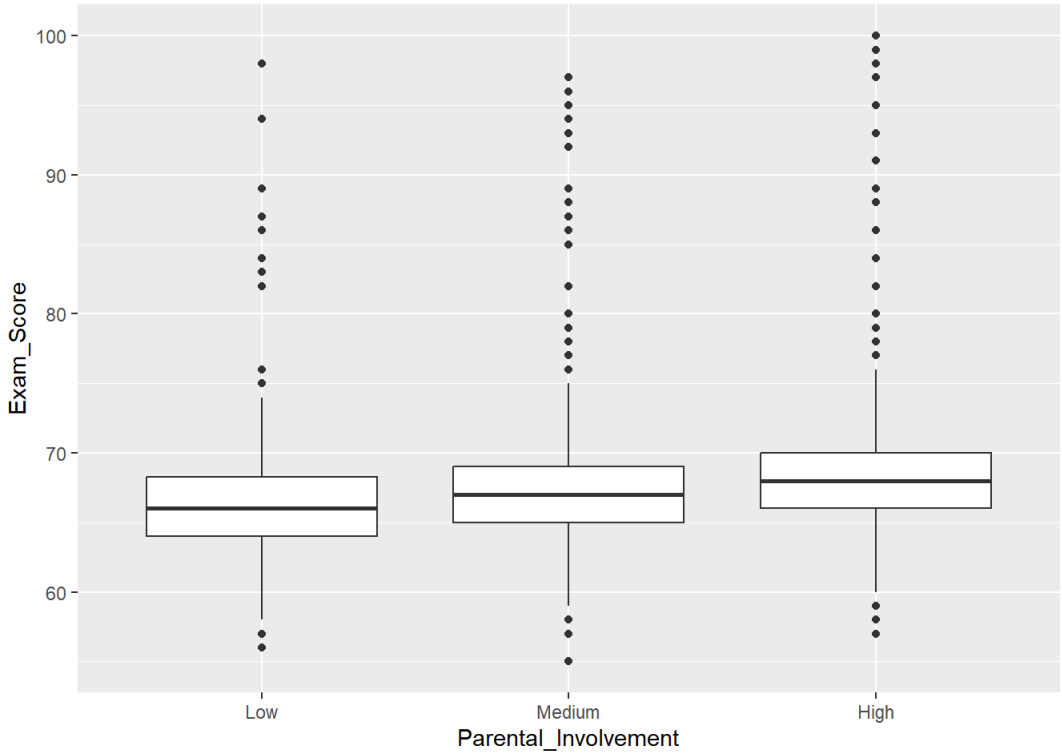
Mindig erdemes **tobb megkozelitest** is hasznalni az adat-exploracio kozben, hogy minel reszletesebb kepet kaphassunk, es csokkentsuk a valoszinuseget hogy egyik vagy masik megkozelites hianyossagai felrevezetnek minket.

```
my_data_corrected %>%
  select(Parental_Involvement, Exam_Score) %>%
  drop_na() %>%
  ggplot() +
    aes(x = Exam_Score) +
    geom_histogram() +
    facet_wrap(~ Parental_Involvement)
```

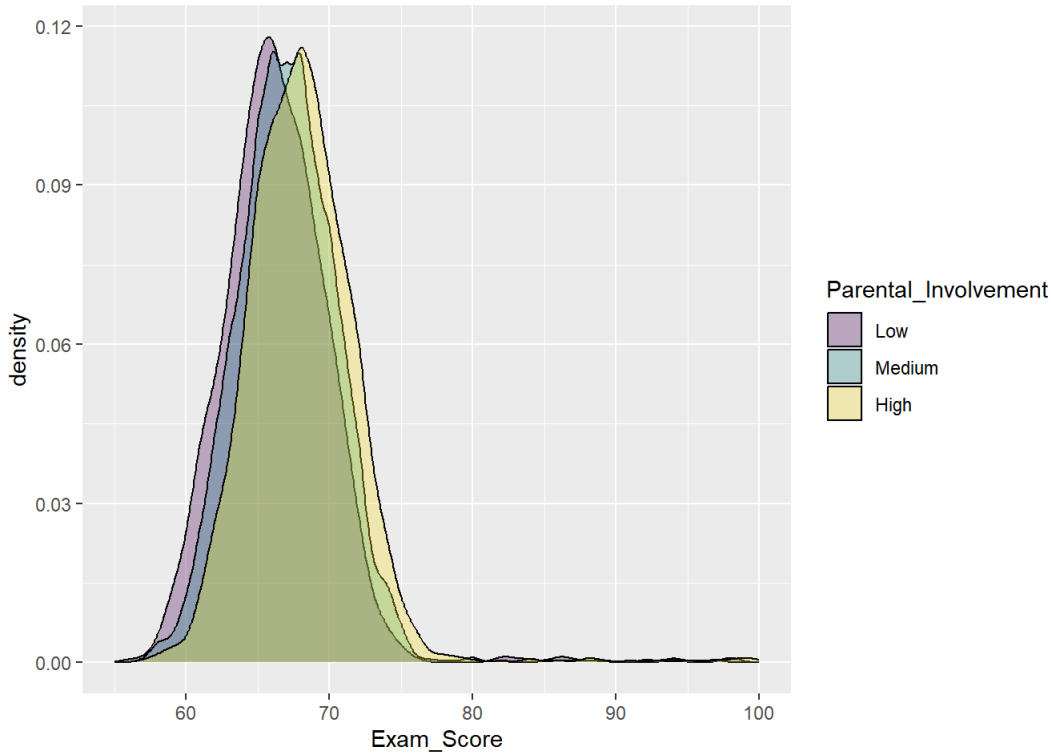
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



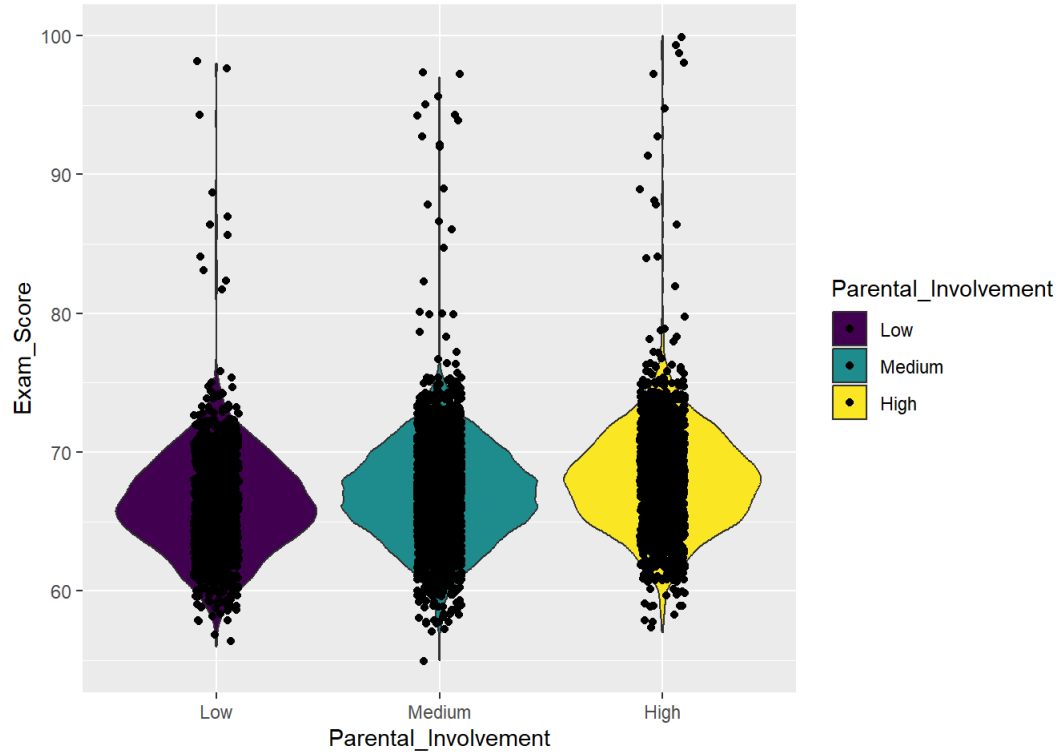

```
my_data_corrected %>%
  select(Parental_Involvement, Exam_Score) %>%
  drop_na() %>%
  ggplot() +
    aes(x = Parental_Involvement, y = Exam_Score) +
    geom_boxplot()
```




```
my_data_corrected %>%
  select(Parental_Involvement, Exam_Score) %>%
  drop_na() %>%
  ggplot() +
    aes(x = Exam_Score, fill = Parental_Involvement) +
    geom_density(alpha = 0.3)
```




```
my_data_corrected %>%
  select(Parental_Involvement, Exam_Score) %>%
  drop_na() %>%
  ggplot() +
    aes(x = Parental_Involvement, y = Exam_Score, fill = Parental_Involvement) +
    geom_violin() +
    geom_jitter(width = 0.1)
```



Gyakorlas

Hasznald a fent tanult modszereket, hogy megvizsgald az **Exam_Score** es a **Learning_Disabilities** 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.

```
my_data_corrected %>%
  select(Exam_Score, Hours_Studied) %>%
  drop_na() %>%
      cor()
```

```
##               Exam_Score Hours_Studied
## Exam_Score     1.0000000     0.4465136
## Hours_Studied  0.4465136     1.0000000
```

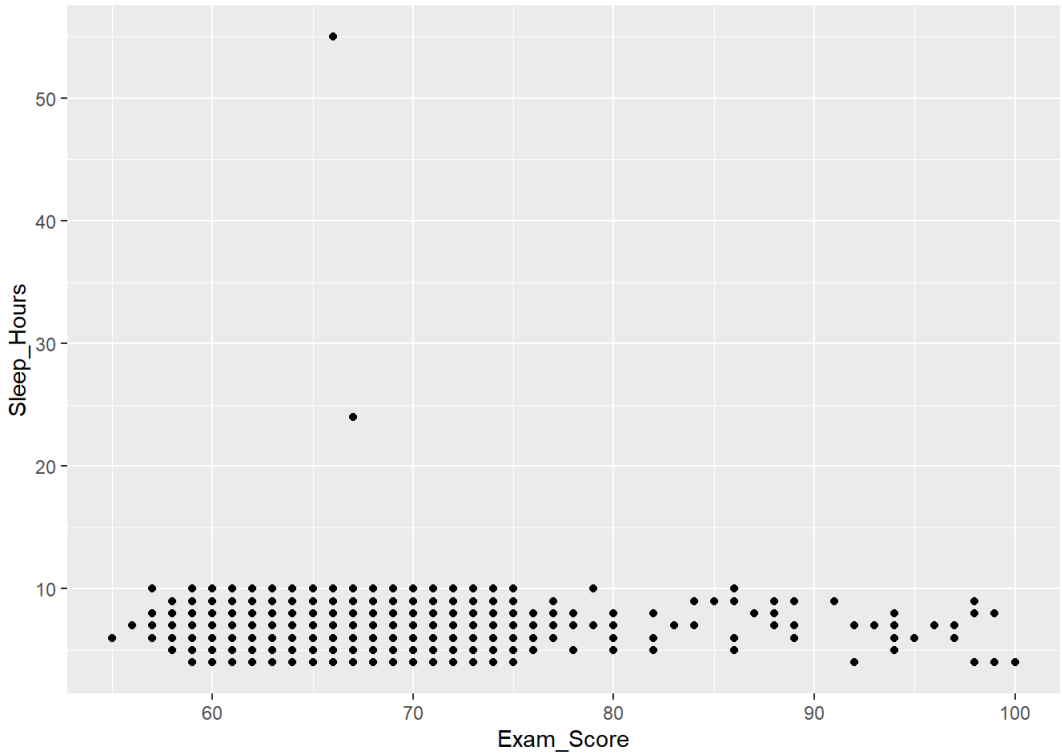
```
my_data_corrected %>%
  select(Exam_Score, Hours_Studied, Previous_Scores) %>%
  drop_na() %>%
      cor()
```

```
##                Exam_Score Hours_Studied Previous_Scores
## Exam_Score       1.0000000    0.44651362      0.17446099
## Hours_Studied    0.4465136    1.00000000      0.02463018
## Previous_Scores  0.1744610    0.02463018      1.00000000
```

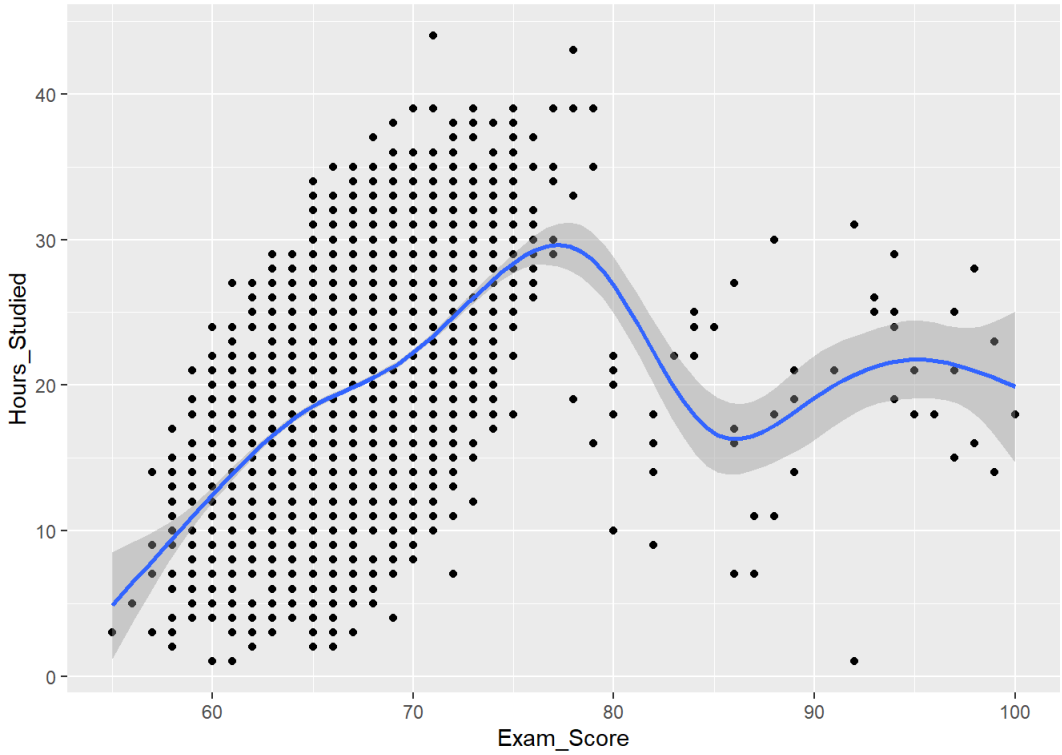
A numerikus valtozok kozotti kapcsolatot altalaban pont diagrammal szoktuk abrazolni (**geom_point()**)

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

```
my_data_corrected %>%
  select(Exam_Score, Sleep_Hours) %>%
  drop_na() %>%
  ggplot() +
    aes(x = Exam_Score, y = Sleep_Hours) +
    geom_point()
```




```
my_data_corrected %>%
  select(Exam_Score, Hours_Studied) %>%
  drop_na() %>%
  ggplot() +
    aes(x = Exam_Score, y = Hours_Studied) +
    geom_point() +
    geom_smooth()
```



Gyakorlas

Milyen eros a kapcsolat az Exam_Score es a Sleep_Hours 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.

```
my_data_corrected %>%
  select(Exam_Score, Hours_Studied, Tutoring_Sessions) %>%
  drop_na() %>%
  ggplot() +
    aes(x = Exam_Score, y = Hours_Studied, col = Tutoring_Sessions) +
    geom_point()+
  scale_colour_gradientn(colours=c("green","black"))
```