Gagnavísindi - Kennsluefni

Viðar Ingason

3/8/2020

Inngangstexti

1 Efni námskeiðsins

Efni námskeiðins skiptist í fjóra flokka eftir dögum.

1. Inngangur að R og RMarkdown
2. Gagnavinnsla með dplyr pakkanum og myndræn framsetning með ggplot2 pakkanum
3. Lestur gagna og notkun tidyr pakkans til að koma gögnum á tidy-format
4. Notkun stýrisetningar (if-else), for loop, tímasetja script og, ef tími gefst, helstu föll úr purrr pakkanum

2 Tími 1

Lærum að vinna með útreikninga í R og fáum góða tilfinningu fyrir R. Búum til breytur og lærum að vinna með mismunandi tegundir breyta. Áttum okkur á *working directory*, *relative path* og *absolute path*. Lærum að lesa inn gögn úr m.a. úr CSV og Excel.

2.1 RStudio og R

R er 25 ára gamalt forritunarmál hannað fyrir tölfræðivinnslu. Í dag er R eitt mest notaðasta forritið þegar kemur að gagnavísindum (e. data science). Þó R sé eitt vinsælasta forritið opna mjög fáir R með beinum hætti. Langflestir styðjast við RStudio sem keyrir R í bakgrunn. Þannig er ekki hægt að nota RStudio nema vera með R uppsett á tölvunni. Þó næstum allir opni ávalt RStudio er samt talað um að vera R forritari eða kunna á R. Það er aldrei talað um að forrita í RStudio.

2.1.1 RStudio stillingar

1. Sýna hvernig eigi að búa til project

- Existing directory eða new

2. working directory
3. Opna Global options

- Haka af 'Restore .Rdata.....
- Save workplace - Never

- Tab width: 4

Þegar þið opnið RStudio þá eru þið staðsett á tilteknum stað í tölvunni ykkar. Til að vita hvar þið eruð stödd notið þið `getwd()` fallið.

Staðsetningar eru mikilvægar og með því að búa alltaf til *Project* leysum við stóran hluta þess vanda að eiga við staðsetningar.
- Getum notað *relative path* í stað *absolute path*

Ef verkefnið okkar er staðsett í **C:/Users/vidar/Rwd/verkefni1** og inní Verkefni1 möppunni eru gögn sem ég vil nota þá get ég lesið þau inn með read_csv("gogn.csv") í stað þess að þurfa að slá inn alla slóðina, þ.e. **read_csv("C:/Users/vidar/Rwd/verkefni1/gogn.csv")**

2.1.2 Inngangur að R

R er hannað með það í huga að framkvæma útreikninga og tölfræðivinnslu. Hefðbundnir útreikningar svo sem samlagning eða tölfræðisamantekt (e. summary statistic) er innbyggt í R (með tölfræðisamantekt er átt við útreikninga á lægsta gildi, hæsta gildi, meðaltali, miðgildi og efri og neðri fjórðungsmörkum). Mat einfaldra tölfræðilíkana og myndræn framsetning er einnig innbyggð. Með þessu er átt við að ekki þarf að hlaða inn neinum viðaukum, svokölluðum pökkum, í R til að framkvæma allt ofangreint.

Styrkleikar R byggja þó á því sem kallast pakkar (e. packages). Í dag eru til um 15 þúsund pakkar í R. Pakki er í stuttu máli safn af kóða sem búið er að skrifa til að framkvæma tiltekin verkefni. Margir pakkar eru í raun ekki að opna nýjar dyr heldur eingöngu að einfalda okkur lífið til muna.

Í námskeiðslýsingunni eru meðal annars nefndir pakkarnir dplyr, tidyr og ggplot2. Þessum pökkum, ásamt nokkrum í viðbót frá sama teymi forritara, hefur verið vafið saman í safn af R pökkum sem kallast tidyverse. Komið verður betur inná það seinna í námskeiðinu.

Til að setja upp pakka þarf aðeins að skrifa í Console-inn `install.packages("nafn pakkans")`. Þannig er hægt að setja dplyr pakkan upp með `install.packages("dplyr")`. Ekki er þörf á að setja upp dplyr, tidyr og ggplot2 sérstaklega heldur nægir að setja upp safnið tidyverse með `install.packages("tidyveres")`.

2.2 Útreikningar með R

Inn í RStudio notum við það sem kallast *Console* til að keyra R kóðan okkar. Prófum að leggja saman tvær tölur. Þið sjáið að fyrir framan svarið stendur [1]. Þetta er númer raðar (e. rownumber). Kem inn á það á eftir.

```
1 + 1
```

```
## [1] 2
```

Athugið að ég set bil á milli 1 og +. Þetta er ekki nauðsynlegt en hjálpar til við að gera lestur kóðan skýrari.

2.3 Breytur

Breytur (e. variables) eru notaðar til að geyma safn upplýsinga sem hægt er að nota hvenær og hvar sem er seinna í kóðanum. Til að búa til breytu notum við *arrow operator*: `<-`
Það má hugsa þetta þannig að við erum að taka það sem er hægra megin við `<-` og setja það inn í breytuna sem er vinstra megin.

```r
x <- 10
x
```

```
## [1] 10
```

```r
y <- 1:10
y
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

Nöfn á breytum geta inniahaldið alla enska stafi, punkt (.) og undirstrik (_). Ég sjálfur nota alltaf undirstrik.

```r
x_1 <- 1:10
x_2 <- 11:20

x_1
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```r
x_2
```

```
##  [1] 11 12 13 14 15 16 17 18 19 20
```

Ef við búum til breytu, t.d. `x <- 1` en ég vil breyta henni þá get ég skrifað yfir hana t.d. `x <- 2`. Nú er breytan `x` = 2 en ekki lengur 1. En við hins vegar viljum henda breytunni þá er það gert með **remove** eða shortcut-inu **rm**

```r
rm(x_2)
x_2
```

Error: object 'x_2' not found.

2.4 Gagnategund (e. data types)

Án þess að fara of djúpt ofan í gagnategundir þá tel ég mikilvægt að koma inn á það. Það eru fjórar megintegundir gagna í R.

- numeric
- character
- Date
- logical (TRUE/FALSE)

2.4.1 Töluleg gögn (e. numeric data)

Þetta samanstendur af tölum í hvaða formi sem við þekkjum þær. Sértilfelli er tegundin *integer* sem eru allar heilar tölur.

```r
x <- 4
is.numeric(x)
```

```
## [1] TRUE
```

```r
class(x)
```

```
## [1] "numeric"
```

2.4.2 Character gögn

Character breytur eru mjög mikilvægar í allri gagnavinnslu. R hefur tvær leiðir til að meðhöndla character breytur. Character og factor.

```r
x <- "forritun"
x
```

```
## [1] "forritun"
```

```r
y <- factor("forritun")
y
```

```
## [1] forritun
## Levels: forritun
```

Takið eftir að `x` er með orðið "forritun" með gæsalöppum á meðan y sýnir orðið án gæsalappa auk þess að birta okkur upplýsingar um levels (stig) breytunnar. Komum betur inná factor í hlutanum um vectora. Athugið að characters eru case sensitive. Það þýðir að FORRITUN, Forritun og forritun eru ekki það sama.

Til að kanna hvort þvær breytur séu eins getum við notað fallið `identical`.

```r
x1 <- "Forritun"
x2 <- "FORRITUN"

identical(x1, x2)
```

```
## [1] FALSE
```

`identical` er sérstaklega mikilvægt þegar unnið er með `character` gögn. Stundum leynist bil fyrir framan eða aftan orð sem getur orsakað leiðindi.

```r
identical("mamma", "mamma ")
```

```
## [1] FALSE
```

```r
library(tidyverse)
x <- "mamma "
x
```

```
## [1] "mamma "
```

```r
str_trim(x)
```

```
## [1] "mamma"
```

Til að finna út lengd á character, þ.e. fjölda stafa og tákna, notum við fallið `nchar`.

```r
nchar(x)
```

```
## [1] 6
```

```r
nchar("Námskeið")
```

```
## [1] 8
```

```r
nchar(12345)
```

```
## [1] 5
```

```r
nchar(y)
```

```
## Error in nchar(y): 'nchar()' requires a character vector
```

nchar vikar ekki á `factor` gögn.

2.4.3 Dagsetningar (e. dates)

Dagsetningar geta verið erfiðar að vinna með (það á við um öll forritunarmál). Aðal leiðin til að vinna með dagsetningar í R er `Date`. Date geymir aðeins upplýsingar um dagsetninguna. Það eru til aðrar leiðir sem geyma upplýsingar um dagsetninguna og tíma, POSIXct. Báðar eru táknaðar sem fjöldi daga (Date) er sekúntna (POSIXct) frá 1. janúar 1970.

```r
date1 <- as.Date("2020-01-01")
date1
```

```
## [1] "2020-01-01"
```

```r
class(date1)
```

```
## [1] "Date"
```

```r
as.numeric(date1)
```

```
## [1] 18262
```

`as.numeric()` breytir, ef það getur, breytu í numeric.

```r
as.numeric("a")
```

```
## [1] NA
```

Mikilvægir pakkar til að vinna með dagsetningar
lubridate, **zoo** og **anytime** eru hjálplegir pakkar þegar unnið er með dagsetningar. Alla jafna þarf maður aðeins á **lubridate** og **zoo** að halda. Það er ekki nema að dagsetningin sé á leiðinlegu formati sem **anytime** kemur til bjargar.

```r
library(anytime)
date_2 <- "200902"
anydate(date_2)
```

```
## [1] "2009-02-01"
```

```r
library(zoo)
date_q <- "2019Q2"
date_quarter <- as.yearqtr(date_q)
date_quarter
```

```
## [1] "2019 Q2"
```

```r
class(date_quarter)
```

```
## [1] "yearqtr"
```

Mikilvæg ef unnið er með tímaraðir að nota `as.yearqtr`. Það hjálpar t.d. við að raða gögnunum rétt frá elstu til nýjustu gagnapunktanna.

2.4.4 Logical

Logical er leið til að meðhöndla gögn sem eru annað hvort TRUE eða FALSE. Tölulega séð er TRUE 1 og FALSE 0.

```r
TRUE * 10
```

```
## [1] 10
```

```r
FALSE * 10
```

```
## [1] 0
```

```r
x <- TRUE
class(TRUE)
```

```
## [1] "logical"
```

T og F eru shortcut fyrir TRUE og FALSE. Athugið að hér verðu að skrifa TRUE með stórum stöfum. true virkar ekki. TRUE og FALSE eru mikilvæg. Þau geta verið útkoman úr samanburði sem við gerum.

```r
2 == 3
```

```
## [1] FALSE
```

```r
2 < 3
```

```
## [1] TRUE
```

```r
2 > 3
```

```
## [1] FALSE
```

```r
4 >= 3
```

```
## [1] TRUE
```

```r
3 <= 4
```

```
## [1] TRUE
```

Mikilvægi logicals koma í ljós hér að neðan þegar við fjöllum um vectora.

2.5 Vectorar

Breyta þarf ekki að vera ein tala. Vektor er safn gilda sem öll eru af sömu tegund (t.d. tölur). Fyrir ykkur sem eru enn fersk í stærðfræðinni þá er þetta ekki það sama og vector í stærðfræði, þ.e. það er ekki til row eða column vector í R. Einfaldasta leiðin til að búa til vector er með **c** sem stendur fyrir concatenate.

```r
x <- c(1, 3, 2, 4)
x
```

```
## [1] 1 3 2 4
```

```r
y <- c("karl", "kona", "strákur", "stelpa")
y
```

```
## [1] "karl"    "kona"    "strákur" "stelpa"
```

```r
z <- c(2, 1, "karl", "kona")
z
```

```
## [1] "2"    "1"    "karl" "kona"
```

2.5.1 Unnið með vectora

Ef við erum með vecot af tölum og viljum margfalda þær allar með sömu tölunni er það mjög auðvelt í R.

```r
x <- 1:10
x * 3
```

```
##  [1]  3  6  9 12 15 18 21 24 27 30
```

```r
x + 2
```

```
##  [1]  3  4  5  6  7  8  9 10 11 12
```

```r
x / 4
```

```
##  [1] 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00 2.25 2.50
```

```r
x^2
```

```
##  [1]   1   4   9  16  25  36  49  64  81 100
```

```r
sqrt(x)
```

```
##  [1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751 2.828427
##  [9] 3.000000 3.162278
```

Ef við viljum búa til röð frá 0 upp í 20 sem hækkar alltaf um 2 þá notum við `seq()` fallið.

```r
y <- seq(0, 20, 2)
```

Ef við gleymum hvað kemur fyrst í einhverju falli eins og `seq` er hægt að fletta upp fallinu með `?seq()`. Hægt er að margfalda saman tvo vectora. Jafnvel af mismunandi lengd. Þá ser styttri vector-inn recycle-aður. Fáum warning ef lengri vector-inn er ekki margfeldi (slétt tala) af þeim styttri.

```r
x <- 1:5
y <- 6:10
x*y
```

```
## [1]  6 14 24 36 50
```

```r
x <- 1:5
y <- 1:9
x*y
```

```
## [1]  1  4  9 16 25  6 14 24 36
```

Hér kemur nytsemi logicals í ljós. R hefur mörg innbyggð gagnasett. Eitt þeirra er árlegar mælingar á rennsli árinnar Níl. Hér sáum við það sem ég nefndi að ofan með stafina fyrir framan það sem prentast út. Seinni línan byrjar á 16 sem er þá sextánda stakið í vector-num.

```
Nile
```

```
## Time Series:
## Start = 1871
## End = 1970
## Frequency = 1
##   [1] 1120 1160  963 1210 1160 1160  813 1230 1370 1140  995  935 1110  994 1020
##  [16]  960 1180  799  958 1140 1100 1210 1150 1250 1260 1220 1030 1100  774  840
##  [31]  874  694  940  833  701  916  692 1020 1050  969  831  726  456  824  702
##  [46] 1120 1100  832  764  821  768  845  864  862  698  845  744  796 1040  759
##  [61]  781  865  845  944  984  897  822 1010  771  676  649  846  812  742  801
##  [76] 1040  860  874  848  890  744  749  838 1050  918  986  797  923  975  815
##  [91] 1020  906  901 1170  912  746  919  718  714  740
```

```
length(Nile)
```

```
## [1] 100
```

Við getum notað innbyggð föll í R til að fá fá meiri upplýsingar um Nile gagnasettið.

```
mean(Nile)
```

```
## [1] 919.35
```

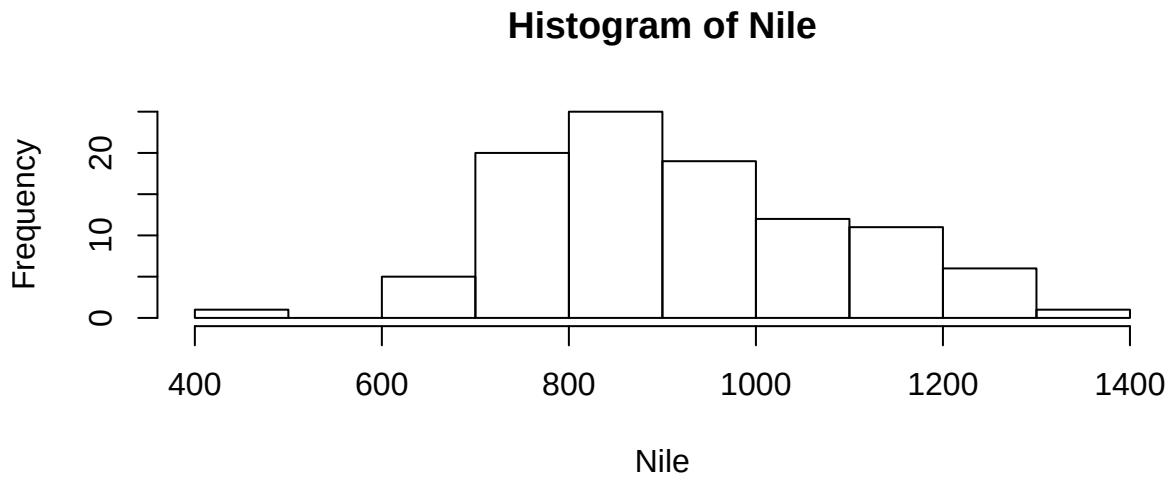
```
sd(Nile)
```

```
## [1] 169.2275
```

```
summary(Nile)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   456.0   798.5   893.5   919.4  1032.5  1370.0
```

```
hist(Nile)
```



Hér sjáum við notkun á Base plot í R. Kynnum ggplot2 pakkanum síðar. Grafið sýnir að einhver gildi eru yfir 1.000. Til að komast að því hversu mörg ár flæðið var yfir 1.000

```
Nile > 1000
```

```
## Time Series:
## Start = 1871
## End = 1970
## Frequency = 1
##   [1]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE
##  [13]  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [25]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
##  [97] FALSE FALSE FALSE FALSE
```

Líkt og nefnt var að ofan er TRUE = 1 og FALSE = 0 í R. Notum þær upplýsignar.

```
sum(Nile > 1000)
```

```
## [1] 30
```

```
mean(Nile > 1000)
```

```
## [1] 0.3
```

Hér er R í raun að *endurnota* (e. recycle) töluna 1.000 þ.e. í bakgrunni býr R til vectorinn `c(1000, 1000,....,1000)`.

Ef við viljum aðeins ákveðin gildi úr vector getum við sótt þau.

```
x <- 10:1
x
```

```
##  [1] 10  9  8  7  6  5  4  3  2  1
```

```
x[1]
```

```
## [1] 10
```

```
x[3]
```

```
## [1] 8
```

```
x[c(1, 4)]
```

```
## [1] 10  7
```

```r
x[x > 5]
```

```
## [1] 10  9  8  7  6
```

Ef við viljum vinna sérstaklega með þau gildi þar sem flæðið er yfir t.d. 1.000 þá getum við tekið þau út og búið til nýjan vector, aðeins með þeim gildum

```r
Nile_1000 <- Nile[Nile > 1000]
Nile_1000
```

```
##  [1] 1120 1160 1210 1160 1160 1230 1370 1140 1110 1020 1180 1140 1100 1210 1150
## [16] 1250 1260 1220 1030 1100 1020 1050 1120 1100 1040 1010 1040 1050 1020 1170
```

2.5.1.1 Factor vectorar

Factor breytur skipta einna mest máli þegar kemur að líkanagerð

```r
q <- c("Grunnskóli", "Framhaldsskóli", "Háskóli", "Háskóli", "Framhaldsskóli", "Grunnskóli", "Framhalds
q_factor <- as.factor(q)
q_factor
```

```
## [1] Grunnskóli     Framhaldsskóli Háskóli        Háskóli        Framhaldsskóli
## [6] Grunnskóli     Framhaldsskóli Háskóli
## Levels: Framhaldsskóli Grunnskóli Háskóli
```

Stundum hefur factor breyta augljósa röðun, líkt og að ofan.

```r
m <- factor(q,
            levels = c("Grunnskóli", "Framhaldsskóli", "Háskóli"),
            ordered = TRUE)
m
```

```
## [1] Grunnskóli     Framhaldsskóli Háskóli        Háskóli        Framhaldsskóli
## [6] Grunnskóli     Framhaldsskóli Háskóli
## Levels: Grunnskóli < Framhaldsskóli < Háskóli
```

2.6 Missing data

```r
x <- c(1, 2, NA, 4, NA)
is.na(x)
```

```
## [1] FALSE FALSE  TRUE FALSE  TRUE
```

Fyllum inn í NA með núlli

```r
x[is.na(x)] <- 0
x
```

```
## [1] 1 2 0 4 0
```

2.7 Workign directories og path

R vísar alltaf í einhverja tiltekna möppu á tölvunni. Getum fundið út hvar við erum "staðsett".

```
getwd()
```

```
## [1] "C:/Users/vidar/Documents/Rwd/teaching"
```

Hægt er að breyta um möppu sem R vísar í með `setwd()`. Þetta er algjörlega óþarfi ef við notum Project sem ég sýni á eftir. Einu skipting sem þetta er í raun nauðsynlegt er þegar við tímasetjum script.

2.8 Data frame

Data frame eru mikilvægasta gagna geymsla (e. data structure) sem er í boði í R þegar kemur að gagnavísindum. Hver og einn column í data frame er vector og verður að vera af sömu tegund. Allir dálkar verða að vera jafn langir.

Þeir sem þekkja Excel munu sjá mikil líkindi þarna á milli.

Mikilvæg föll eru `dim`, `nrow` og `ncol`.

2.9 Lestur gagna

Það er sjaldnast þannig að við erum að vinna með gögn sem við sláum sjálf inn. R er með innbyggð föll fyrir ansi margt. Oft kemur þó að því að við viljum skrifa okkar eigin föll.

Fall er eitthvað sem tekur inn einhverjar upplýsingar, framkvæmir útreikningar með þær upplýsingar og skilar okkur svo niðurstöðunum.

Það er góður tími að búa til föll ef við erum að keyra sama kóðan aftur og aftur.

```
margf_2 <- function(x, y) {
  m <- x * y
  paste("Útkoman er", m)
}
```

Getum einnig sett inn default argument

```
margf_3 <- function(x, y, z = 100) {
  m <- x * y / z
  paste("Útkoman er", m)
}
```

3 R Markdown

- YAML
- Inline Code
- Code Chunks

Búa til markdown skjal. Sýna Documents og presentations. Fara yfir YAML header-inn. Fara yfir knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE)

3.1 Texti - Options

Valmöguleiki	Sjálfvaldið	Áhrif
echo	TRUE	Viltu sýna kóðan eða ekki
warning	TRUE	Á að sýna *warnings*
message	TRUE	Á að sýna *message*
cache	FALSE	Á að geyma reiknifrekar útreikninga
fig.width	7	Plot breidd í tommum
fig.height	7	Plot hæði í tommum

3.2

4 Tími 2 - Gagnagreining með dplyr og tidyr

```
library(tidyverse)

mbl <- read_csv("./slides/lecture_2/mbl_hreint.csv")
str(mbl)
```

```
## tibble [4,664 x 16] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Gata         : chr [1:4664] "Akurhvarf 3" "Garðhús 53" "Svöluhöfði 7" "Rjúpnasalir 10" ...
##  $ Stadur       : chr [1:4664] "Kópavogi" "Reykjavík" "Mosfellsbæ" "Kópavogi" ...
##  $ Verd         : num [1:4664] 4.99e+07 5.35e+07 1.08e+08 4.89e+07 6.29e+07 ...
##  $ Fasteignamat : num [1:4664] 45450000 45550000 89300000 43750000 60400000 ...
##  $ Brunabotamat : num [1:4664] 40240000 37000000 66340000 36390000 34510000 ...
##  $ Ahvilandi    : num [1:4664] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Tegund       : chr [1:4664] "Fjölbýli" "Fjölbýli" "Einbýli" "Fjölbýli" ...
##  $ Byggingarar  : num [1:4664] 2005 1991 2001 2003 1984 ...
##  $ Size         : num [1:4664] 105 114 218 94 188 ...
##  $ Herbergi     : num [1:4664] 3 4 6 3 4 3 3 8 4 4 ...
##  $ Svefnherbergi: num [1:4664] 2 2 4 2 2 2 2 6 3 3 ...
##  $ Stofur       : num [1:4664] 1 2 2 1 2 1 1 2 1 1 ...
##  $ Badherbergi  : num [1:4664] 1 1 1 1 2 1 1 3 2 1 ...
##  $ Inngangur    : chr [1:4664] "Sameiginlegur" "Sérinngangur" "Sérinngangur" "Sameiginlegur" ...
##  $ Bilskur      : chr [1:4664] "Já" "Já" "Já" "Nei / Ekki vitað" ...
##  $ Nybygging    : chr [1:4664] "Nei / Ekki vitað" "Nei / Ekki vitað" "Nei / Ekki vitað" "Nei / Ekki v
##  - attr(*, "spec")=
##   .. cols(
##   ..   Gata = col_character(),
##   ..   Stadur = col_character(),
##   ..   Verd = col_double(),
##   ..   Fasteignamat = col_double(),
##   ..   Brunabotamat = col_double(),
##   ..   Ahvilandi = col_double(),
##   ..   Tegund = col_character(),
##   ..   Byggingarar = col_double(),
```

```
##   ..   Size = col_double(),
##   ..   Herbergi = col_double(),
##   ..   Svefnherbergi = col_double(),
##   ..   Stofur = col_double(),
##   ..   Badherbergi = col_double(),
##   ..   Inngangur = col_character(),
##   ..   Bilskur = col_character(),
##   ..   Nybygging = col_character()
##   .. )
```

4.1 Læra inn á gögnin

Áður en greining hefst er gott að fá tilfinningu fyrir gögnunum

Ein góð leið er að nota gogn %>% count(breyta) ef um categorical breytu er að ræða

Fyrir númeríska breytur virkar `summary()` fallið vel.

4.2 Filter

Þegar við skrifum langar pípur er gott að koma `filter()` eins framarlega og hægt er. Með því er gagnasettið orðið minna og auðveldara að vinna með.

```
filter(mbl, Verd > 100000000, Tegund == "Fjölbýli")
```

```
## # A tibble: 44 x 16
##    Gata  Stadur   Verd Fasteignamat Brunabotamat Ahvilandi Tegund Byggingarar
##    <chr> <chr>   <dbl>        <dbl>         <dbl>     <dbl> <chr>         <dbl>
##  1 Bæja~ Kópav~ 1.40e8     86100000     114090000         0 Fjölb~         2019
##  2 Tryg~ Reykj~ 1.24e8     79450000      73050000         0 Fjölb~         2017
##  3 Smyr~ Reykj~ 1.08e8            0             0         0 Fjölb~         2019
##  4 Smyr~ Reykj~ 1.09e8            0             0         0 Fjölb~         2019
##  5 Tryg~ Reykj~ 1.24e8     85600000      72880000         0 Fjölb~         2017
##  6 Garð~ Reykj~ 2.79e8    205350000     142580000         0 Fjölb~         1968
##  7 Skól~ Seltj~ 1.70e8    139000000     107800000         0 Fjölb~         1950
##  8 Urri~ Garða~ 1.35e8      8150000             0         0 Fjölb~            0
##  9 Bæja~ Kópav~ 1.19e8     72500000     100840000         0 Fjölb~         2019
## 10 Garð~ Garða~ 1.17e8     73950000      56460000         0 Fjölb~         2016
## # ... with 34 more rows, and 8 more variables: Size <dbl>, Herbergi <dbl>,
## #   Svefnherbergi <dbl>, Stofur <dbl>, Badherbergi <dbl>, Inngangur <chr>,
## #   Bilskur <chr>, Nybygging <chr>
```

Hér setjum við gagnasettið sem fyrsta input í `filter()` fallið. Næstu tvö input eru síðan þær breytur sem við erum að filter-a.

Í seinasta tíma töluðum við stuttlega um %>%. Í raun er heppilegra að nota seinni leiðina því seinna í dag munum við búa til langar pípur.

```
mbl %>%
  filter(Verd > 100000000,
         Tegund == "Fjölbýli")
```

```
## # A tibble: 44 x 16
##    Gata  Stadur   Verd Fasteignamat Brunabotamat Ahvilandi Tegund Byggingarar
##    <chr> <chr>   <dbl>        <dbl>        <dbl>     <dbl> <chr>        <dbl>
##  1 Bæja~ Kópav~ 1.40e8     86100000    114090000         0 Fjölb~        2019
##  2 Tryg~ Reykj~ 1.24e8     79450000     73050000         0 Fjölb~        2017
##  3 Smyr~ Reykj~ 1.08e8            0            0         0 Fjölb~        2019
##  4 Smyr~ Reykj~ 1.09e8            0            0         0 Fjölb~        2019
##  5 Tryg~ Reykj~ 1.24e8     85600000     72880000         0 Fjölb~        2017
##  6 Garð~ Reykj~ 2.79e8    205350000    142580000         0 Fjölb~        1968
##  7 Skól~ Seltj~ 1.70e8    139000000    107800000         0 Fjölb~        1950
##  8 Urri~ Garða~ 1.35e8      8150000            0         0 Fjölb~           0
##  9 Bæja~ Kópav~ 1.19e8     72500000    100840000         0 Fjölb~        2019
## 10 Garð~ Garða~ 1.17e8     73950000     56460000         0 Fjölb~        2016
## # ... with 34 more rows, and 8 more variables: Size <dbl>, Herbergi <dbl>,
## #   Svefnherbergi <dbl>, Stofur <dbl>, Badherbergi <dbl>, Inngangur <chr>,
## #   Bilskur <chr>, Nybygging <chr>
```

Ef við viljum filtera eftir mörgum viðmiðum notum við **%in%** og boolean operation Hérna veljum við Verð á bilinu 50 - 100 milljónir. Þá veljum við einnig tegund sem er Einbýli eða Raðhús. Niðurstaðan er gagnasett með fasteignum sem uppfylla bæði skilyrðin

```
mbl %>%
  filter(Verd >= 50000000 & Verd <= 100000000,
         Tegund %in% c("Einbýli", "Fjölbýli"))
```

```
## # A tibble: 1,869 x 16
##    Gata  Stadur   Verd Fasteignamat Brunabotamat Ahvilandi Tegund Byggingarar
##    <chr> <chr>   <dbl>        <dbl>        <dbl>     <dbl> <chr>        <dbl>
##  1 Garð~ Reykj~ 5.35e7     45550000     37000000         0 Fjölb~        1991
##  2 Dalb~ Kópav~ 6.29e7     60400000     34510000         0 Fjölb~        1984
##  3 Hrau~ Reykj~ 5.59e7     42550000     34950000         0 Fjölb~        1990
##  4 Ögur~ Kópav~ 7.29e7     26350000            0         0 Fjölb~        2019
##  5 Tjar~ Reykj~ 5.97e7     51750000     31600000         0 Fjölb~        1945
##  6 Ögur~ Kópav~ 7.29e7     27700000            0         0 Fjölb~        2019
##  7 Ögur~ Kópav~ 8.49e7     26650000            0         0 Einbý~        2019
##  8 Ögur~ Kópav~ 7.29e7     29500000            0         0 Fjölb~        2019
##  9 Bjar~ Akran~ 6.49e7     57450000     57590000         0 Einbý~        1974
## 10 Hafn~ Kópav~ 7.49e7     30550000            0         0 Fjölb~        2019
## # ... with 1,859 more rows, and 8 more variables: Size <dbl>, Herbergi <dbl>,
## #   Svefnherbergi <dbl>, Stofur <dbl>, Badherbergi <dbl>, Inngangur <chr>,
## #   Bilskur <chr>, Nybygging <chr>
```

Stundum viljum við sækja gögn sem uppfylla eitt skilyðir **eða** annað en ekki endilega bæði samtímis.

```
mbl %>%
  filter(Tegund == "Einbýli", Verd > 150000000 | Size > 350)
```

```
## # A tibble: 34 x 16
##    Gata  Stadur   Verd Fasteignamat Brunabotamat Ahvilandi Tegund Byggingarar
##    <chr> <chr>   <dbl>        <dbl>        <dbl>     <dbl> <chr>        <dbl>
##  1 Enni~ Kópav~ 1.60e8    146050000    165850000         0 Einbý~        2006
##  2 Ástu~ Mosfe~ 1.33e8    111900000    141100000         0 Einbý~        2017
```

```
##  3 Smár~ Reykj~ 1.98e8    162525000    101160000         0 Einbý~        1939
##  4 Voga~ Reykj~ 1.42e8    129850000    141450000         0 Einbý~        1978
##  5 Klei~ Kópav~ 1.59e8    118800000    116750000         0 Einbý~        2006
##  6 Skel~ Reykj~ 1.54e8    154450000     81600000         0 Einbý~        1968
##  7 Másh~ Reykj~ 9.90e7     94750000    108850000         0 Einbý~        1979
##  8 Efst~ Mosfe~ 1.24e8     17100000            0         0 Einbý~        2019
##  9 Kvis~ Selfo~ 5.49e7            0            0         0 Einbý~        1983
## 10 Bjar~ Akure~ 7.69e7     87400000    118550000         0 Einbý~        1956
## # ... with 24 more rows, and 8 more variables: Size <dbl>, Herbergi <dbl>,
## #   Svefnherbergi <dbl>, Stofur <dbl>, Badherbergi <dbl>, Inngangur <chr>,
## #   Bilskur <chr>, Nybygging <chr>
```

Oft erum við einnig með gögn þar sem auðveldara er að skilgreina hvað við viljum ekki frekar en að skilgreina hvað við viljum

```
mbl %>%
  filter(Tegund != "Fjölbýli")
```

```
## # A tibble: 1,089 x 16
##    Gata  Stadur   Verd Fasteignamat Brunabotamat Ahvilandi Tegund Byggingarar
##    <chr> <chr>   <dbl>        <dbl>        <dbl>     <dbl> <chr>         <dbl>
##  1 Svöl~ Mosfe~ 1.08e8     89300000     66340000         0 Einbý~         2001
##  2 Enni~ Kópav~ 1.60e8    146050000    165850000         0 Einbý~         2006
##  3 Ögur~ Kópav~ 8.49e7     26650000            0         0 Einbý~         2019
##  4 Bjar~ Akran~ 6.49e7     57450000     57590000         0 Einbý~         1974
##  5 MÁNA~ Hvera~ 4.85e7     29800000     53800000         0 Atvin~         2008
##  6 Gagn~ Selfo~ 3.46e7      2440000            0         0 Atvin~         1978
##  7 Gagn~ Selfo~ 3.25e7      2410000            0         0 Atvin~         1978
##  8 Gagn~ Selfo~ 2.00e8      2440000            0         0 Atvin~         1978
##  9 Illu~ Vestm~ 3.39e7     27600000     37650000         0 Einbý~         1932
## 10 Rauð~ Hafna~ 3.95e7     25400000     36950000         0 Atvin~         2006
## # ... with 1,079 more rows, and 8 more variables: Size <dbl>, Herbergi <dbl>,
## #   Svefnherbergi <dbl>, Stofur <dbl>, Badherbergi <dbl>, Inngangur <chr>,
## #   Bilskur <chr>, Nybygging <chr>
```

```
mbl %>%
  filter(!Tegund %in% c("Fjölbýli", "Einbýli"))
```

```
## # A tibble: 448 x 16
##    Gata  Stadur   Verd Fasteignamat Brunabotamat Ahvilandi Tegund Byggingarar
##    <chr> <chr>   <dbl>        <dbl>        <dbl>     <dbl> <chr>         <dbl>
##  1 MÁNA~ Hvera~ 4.85e7     29800000     53800000         0 Atvin~         2008
##  2 Gagn~ Selfo~ 3.46e7      2440000            0         0 Atvin~         1978
##  3 Gagn~ Selfo~ 3.25e7      2410000            0         0 Atvin~         1978
##  4 Gagn~ Selfo~ 2.00e8      2440000            0         0 Atvin~         1978
##  5 Rauð~ Hafna~ 3.95e7     25400000     36950000         0 Atvin~         2006
##  6 Hval~ Reykj~ 5.00e6      3660000            0         0 Atvin~         2009
##  7 Stra~ Hafna~ 7.49e7     44500000     57400000         0 Atvin~         1905
##  8 Álfh~ Hafna~ 4.15e7     18300000            0         0 Atvin~         2019
##  9 Goða~ Akure~ 2.36e7     18450000     25150000         0 Atvin~         2012
## 10 Stra~ Sandg~ 1.99e7      2570000      6680000         0 Atvin~         1965
## # ... with 438 more rows, and 8 more variables: Size <dbl>, Herbergi <dbl>,
## #   Svefnherbergi <dbl>, Stofur <dbl>, Badherbergi <dbl>, Inngangur <chr>,
## #   Bilskur <chr>, Nybygging <chr>
```

Stundum erum við með `character` breytu og viljum filter-a breytur sem innihalda eitthvað sameiginlegt

```
mbl %>%
  filter(str_detect(Stadur, "bæ")) %>%
  count(Stadur)
```

```
## # A tibble: 4 x 2
##   Stadur          n
##   <chr>       <int>
## 1 Garðabæ       247
## 2 Mosfellsbæ    311
## 3 Reykjanesbæ   237
## 4 Snæfellsbæ      1
```

Mörg gagnasett hafa tvær raðir sem eru eins (duplicated) en við viljum aðeins aðra þeirra

```
df1 <- tibble(a = c(1, 2, 1, 4),
              b = c(1, 2, 1, 2),
              c = c("a", "b", "a", "b"))

distinct(df1)
```

```
## # A tibble: 3 x 3
##       a     b c
##   <dbl> <dbl> <chr>
## 1     1     1 a
## 2     2     2 b
## 3     4     2 b
```

4.3 mutate

Breytum ásettu verði í milljónir króna í stað krónur ásamt því að reikna út ásett verð umfram fasteignamat og setjum það í milljónir króna

```
mbl %>%
  mutate(verd_milljon = Verd / 1000000,
         verd_umfram_mat = (Verd - Fasteignamat) / 1000000) %>%
  select(verd_milljon, verd_umfram_mat)
```

```
## # A tibble: 4,664 x 2
##    verd_milljon verd_umfram_mat
##           <dbl>           <dbl>
##  1         49.9            4.45
##  2         53.5            7.95
##  3        108.            19.2
##  4         48.9            5.15
##  5         62.9            2.5
##  6         38.9           -2.5
##  7         55.9           13.4
##  8        160.            13.8
##  9         72.9           46.6
## 10         59.7            7.95
## # ... with 4,654 more rows
```

4.4 Select

Í gagnavinnslu er mjög algengt að vilja aðeins vinna með hluta gagnanna. T.d. bara 5 breytur af 20.

```
mbl %>%
  select(Gata, Verd, Tegund)
```

```
## # A tibble: 4,664 x 3
##    Gata                 Verd Tegund
##    <chr>               <dbl> <chr>
##  1 Akurhvarf 3      49900000 Fjölbýli
##  2 Garðhús 53       53500000 Fjölbýli
##  3 Svöluhöfði 7    108500000 Einbýli
##  4 Rjúpnasalir 10   48900000 Fjölbýli
##  5 Dalbrekka 23     62900000 Fjölbýli
##  6 Sólvallagata 33  38900000 Fjölbýli
##  7 Hraunbær 103     55900000 Fjölbýli
##  8 Ennishvarf 8    159900000 Einbýli
##  9 Ögurhvarf 4D     72900000 Fjölbýli
## 10 Tjarnargata 10   59700000 Fjölbýli
## # ... with 4,654 more rows
```

```
mbl %>%
  select(Gata:Tegund)
```

```
## # A tibble: 4,664 x 7
##    Gata          Stadur         Verd Fasteignamat Brunabotamat Ahvilandi Tegund
##    <chr>         <chr>         <dbl>        <dbl>        <dbl>     <dbl> <chr>
##  1 Akurhvarf 3   Kópavogi     4.99e7     45450000     40240000         0 Fjölbý~
##  2 Garðhús 53    Reykjavík    5.35e7     45550000     37000000         0 Fjölbý~
##  3 Svöluhöfði 7  Mosfellsbæ   1.08e8     89300000     66340000         0 Einbýli
##  4 Rjúpnasalir ~ Kópavogi     4.89e7     43750000     36390000         0 Fjölbý~
##  5 Dalbrekka 23  Kópavogi     6.29e7     60400000     34510000         0 Fjölbý~
##  6 Sólvallagata~ Reykjavík    3.89e7     41400000     20050000         0 Fjölbý~
##  7 Hraunbær 103  Reykjavík    5.59e7     42550000     34950000         0 Fjölbý~
##  8 Ennishvarf 8  Kópavogi     1.60e8    146050000    165850000         0 Einbýli
##  9 Ögurhvarf 4D  Kópavogi     7.29e7     26350000            0         0 Fjölbý~
## 10 Tjarnargata ~ Reykjavík    5.97e7     51750000     31600000         0 Fjölbý~
## # ... with 4,654 more rows
```

```
mbl %>%
  select(-Gata)
```

```
## # A tibble: 4,664 x 15
##    Stadur   Verd Fasteignamat Brunabotamat Ahvilandi Tegund Byggingarar  Size
##    <chr>   <dbl>        <dbl>        <dbl>     <dbl> <chr>         <dbl> <dbl>
##  1 Kópav~ 4.99e7     45450000     40240000         0 Fjölb~         2005 105.
##  2 Reykj~ 5.35e7     45550000     37000000         0 Fjölb~         1991 114.
##  3 Mosfe~ 1.08e8     89300000     66340000         0 Einbý~         2001 218.
##  4 Kópav~ 4.89e7     43750000     36390000         0 Fjölb~         2003  94
##  5 Kópav~ 6.29e7     60400000     34510000         0 Fjölb~         1984 188.
##  6 Reykj~ 3.89e7     41400000     20050000         0 Fjölb~         1928  67.5
```

```
##  7 Reykj~ 5.59e7     42550000     34950000         0 Fjölb~        1990 112.
##  8 Kópav~ 1.60e8    146050000    165850000         0 Einbý~        2006 541.
##  9 Kópav~ 7.29e7     26350000            0         0 Fjölb~        2019 144.
## 10 Reykj~ 5.97e7     51750000     31600000         0 Fjölb~        1945 107.
## # ... with 4,654 more rows, and 7 more variables: Herbergi <dbl>,
## #   Svefnherbergi <dbl>, Stofur <dbl>, Badherbergi <dbl>, Inngangur <chr>,
## #   Bilskur <chr>, Nybygging <chr>
```

```
mbl %>%
  select(starts_with("B"))
```

```
## # A tibble: 4,664 x 4
##    Brunabotamat Byggingarar Badherbergi Bilskur
##           <dbl>       <dbl>       <dbl> <chr>
##  1     40240000        2005           1 Já
##  2     37000000        1991           1 Já
##  3     66340000        2001           1 Já
##  4     36390000        2003           1 Nei / Ekki vitað
##  5     34510000        1984           2 Já
##  6     20050000        1928           1 Nei / Ekki vitað
##  7     34950000        1990           1 Já
##  8    165850000        2006           3 Nei / Ekki vitað
##  9            0        2019           2 Nei / Ekki vitað
## 10     31600000        1945           1 Nei / Ekki vitað
## # ... with 4,654 more rows
```

```
mbl %>%
  select(ends_with("mat"))
```

```
## # A tibble: 4,664 x 2
##    Fasteignamat Brunabotamat
##           <dbl>        <dbl>
##  1     45450000     40240000
##  2     45550000     37000000
##  3     89300000     66340000
##  4     43750000     36390000
##  5     60400000     34510000
##  6     41400000     20050000
##  7     42550000     34950000
##  8    146050000    165850000
##  9     26350000            0
## 10     51750000     31600000
## # ... with 4,654 more rows
```

```
mbl %>%
  select(contains("herbergi"))
```

```
## # A tibble: 4,664 x 3
##    Herbergi Svefnherbergi Badherbergi
##       <dbl>         <dbl>       <dbl>
##  1        3             2           1
##  2        4             2           1
```

```
##  3        6             4           1
##  4        3             2           1
##  5        4             2           2
##  6        3             2           1
##  7        3             2           1
##  8        8             6           3
##  9        4             3           2
## 10        4             3           1
## # ... with 4,654 more rows
```

```
mbl %>%
  select(1, 7)
```

```
## # A tibble: 4,664 x 2
##    Gata            Tegund
##    <chr>           <chr>
##  1 Akurhvarf 3     Fjölbýli
##  2 Garðhús 53      Fjölbýli
##  3 Svöluhöfði 7    Einbýli
##  4 Rjúpnasalir 10  Fjölbýli
##  5 Dalbrekka 23    Fjölbýli
##  6 Sólvallagata 33 Fjölbýli
##  7 Hraunbær 103    Fjölbýli
##  8 Ennishvarf 8    Einbýli
##  9 Ögurhvarf 4D    Fjölbýli
## 10 Tjarnargata 10  Fjölbýli
## # ... with 4,654 more rows
```

Getum einnit notað `select` fallið til að endurraða gögnunum

```
mbl %>%
  select(Svefnherbergi, Badherbergi, everything())
```

```
## # A tibble: 4,664 x 16
##    Svefnherbergi Badherbergi Gata  Stadur   Verd Fasteignamat Brunabotamat
##            <dbl>       <dbl> <chr> <chr>   <dbl>        <dbl>        <dbl>
##  1             2           1 Akur~ Kópav~ 4.99e7     45450000     40240000
##  2             2           1 Garð~ Reykj~ 5.35e7     45550000     37000000
##  3             4           1 Svöl~ Mosfe~ 1.08e8     89300000     66340000
##  4             2           1 Rjúp~ Kópav~ 4.89e7     43750000     36390000
##  5             2           2 Dalb~ Kópav~ 6.29e7     60400000     34510000
##  6             2           1 Sólv~ Reykj~ 3.89e7     41400000     20050000
##  7             2           1 Hrau~ Reykj~ 5.59e7     42550000     34950000
##  8             6           3 Enni~ Kópav~ 1.60e8    146050000    165850000
##  9             3           2 Ögur~ Kópav~ 7.29e7     26350000            0
## 10             3           1 Tjar~ Reykj~ 5.97e7     51750000     31600000
## # ... with 4,654 more rows, and 9 more variables: Ahvilandi <dbl>,
## #   Tegund <chr>, Byggingarar <dbl>, Size <dbl>, Herbergi <dbl>, Stofur <dbl>,
## #   Inngangur <chr>, Bilskur <chr>, Nybygging <chr>
```

4.5 Summarise

Summarise er notað til að draga saman upplýsingar í eina tölu. T.d. meðalfjölda fermetra eftir tegund húsnæðis í Reykjavík. Summarise er mjög oft notað með `group_by`.

```
mbl %>%
  filter(Stadur == "Reykjavík") %>%
  group_by(Tegund) %>%
  summarise(Medal_fermetrar = mean(Size),
            Medal_verd = mean(Verd)) %>%
  mutate(Medal_fermetraverd = Medal_verd/Medal_fermetrar)
```

```
## # A tibble: 3 x 4
##   Tegund         Medal_fermetrar Medal_verd Medal_fermetraverd
##   <chr>                    <dbl>      <dbl>              <dbl>
## 1 Atvinnuhúsnæði           348.   91101840.            261640.
## 2 Einbýli                  242.  110338750             456485.
## 3 Fjölbýli                  95.4  55875106.            585601.
```

4.6 Arrange

Með `arrange` fallinu má raða breytum upp á nýtt. Þetta getur verið gríðarlega mikilvægt í útreikningum.

Skiptum nú um gagnasett. Til að nota næsta gagnasett setjum við inn pakkann **gapminder** og notum gagnasett sem einnig heitir gapminder.

```
library(gapminder)
gapminder
```

```
## # A tibble: 1,704 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.
##  2 Afghanistan Asia       1957    30.3  9240934      821.
##  3 Afghanistan Asia       1962    32.0 10267083      853.
##  4 Afghanistan Asia       1967    34.0 11537966      836.
##  5 Afghanistan Asia       1972    36.1 13079460      740.
##  6 Afghanistan Asia       1977    38.4 14880372      786.
##  7 Afghanistan Asia       1982    39.9 12881816      978.
##  8 Afghanistan Asia       1987    40.8 13867957      852.
##  9 Afghanistan Asia       1992    41.7 16317921      649.
## 10 Afghanistan Asia       1997    41.8 22227415      635.
## # ... with 1,694 more rows
```

Rugla röðuninni á upplaflega gapminder gagnasettinu

```
ny_rodun <- sample(nrow(gapminder), nrow(gapminder))
gap_rugl <- gapminder[ny_rodun,]
gap_rugl
```

```
## # A tibble: 1,704 x 6
##    country         continent  year lifeExp      pop gdpPercap
##    <fct>           <fct>     <int>   <dbl>    <int>     <dbl>
##  1 France          Europe     2007    80.7 61083916    30470.
##  2 Swaziland       Africa     1972    49.6   480105     3365.
##  3 Guatemala       Americas   1962    47.0  4208858     2750.
```

```
##  4 Italy           Europe     1957    67.8 49182000     6249.
##  5 Italy           Europe     1992    77.4 56840847    22014.
##  6 Slovak Republic Europe     2007    74.7  5447502    18678.
##  7 Senegal         Africa     1972    45.8  4588696     1598.
##  8 Togo            Africa     2002    57.6  4977378      886.
##  9 Portugal        Europe     2007    78.1 10642836    20510.
## 10 Puerto Rico     Americas   1962    69.6  2448046     5108.
## # ... with 1,694 more rows
```

Raða gögnunum upp á nýtt og reikna út hagvöxt, þ.e. breytingu á gdpPercap. Byrja á því að gera þetta með röngum hætti, þ.e. gleyma að raða eftir ári líka.

4.7 Kynni fyrst lead og lag.

Ef við viljum nota tafið gildi (e. lag) af einhverri breytu getum við ntoað `lag` fallið í **dplyr** pakkanum. Ef við viljum nota næsta gildi á eftir getum við notað `lead` fallið í sama pakka. `lag` og `lead` eru svokölluð *window function*. Window function er ólíkt t.d. `sum` og `mean` þar sem útkoman er ein tala. Í window function er útkoman n tölur.

```
df <- tibble(x = 1:10)
df %>%
  mutate(x_lag= lag(x),
         x_lead = lead(x))
```

```
## # A tibble: 10 x 3
##        x x_lag x_lead
##    <int> <int>  <int>
##  1     1    NA      2
##  2     2     1      3
##  3     3     2      4
##  4     4     3      5
##  5     5     4      6
##  6     6     5      7
##  7     7     6      8
##  8     8     7      9
##  9     9     8     10
## 10    10     9     NA
```

```
gap_rugl %>%
  arrange(country) %>%
  mutate(gdp_growth = gdpPercap/lag(gdpPercap, 1) - 1)
```

```
## # A tibble: 1,704 x 7
##    country     continent  year lifeExp      pop gdpPercap gdp_growth
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>      <dbl>
##  1 Afghanistan Asia       1972    36.1 13079460      740.   NA
##  2 Afghanistan Asia       1957    30.3  9240934      821.    0.109
##  3 Afghanistan Asia       2002    42.1 25268405      727.   -0.115
##  4 Afghanistan Asia       1962    32.0 10267083      853.    0.174
##  5 Afghanistan Asia       1992    41.7 16317921      649.   -0.239
##  6 Afghanistan Asia       1967    34.0 11537966      836.    0.288
##  7 Afghanistan Asia       1982    39.9 12881816      978.    0.170
```

```
##  8 Afghanistan Asia       1997    41.8 22227415      635.   -0.350
##  9 Afghanistan Asia       1952    28.8  8425333      779.    0.227
## 10 Afghanistan Asia       1977    38.4 14880372      786.    0.00855
## # ... with 1,694 more rows
```

Laga villinu hér

```
gap_rugl %>%
  arrange(country, year) %>%
  mutate(gdp_growth = gdpPercap/lag(gdpPercap, 1) - 1)
```

```
## # A tibble: 1,704 x 7
##    country     continent  year lifeExp      pop gdpPercap gdp_growth
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>      <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.    NA
##  2 Afghanistan Asia       1957    30.3  9240934      821.     0.0531
##  3 Afghanistan Asia       1962    32.0 10267083      853.     0.0393
##  4 Afghanistan Asia       1967    34.0 11537966      836.    -0.0198
##  5 Afghanistan Asia       1972    36.1 13079460      740.    -0.115
##  6 Afghanistan Asia       1977    38.4 14880372      786.     0.0623
##  7 Afghanistan Asia       1982    39.9 12881816      978.     0.244
##  8 Afghanistan Asia       1987    40.8 13867957      852.    -0.128
##  9 Afghanistan Asia       1992    41.7 16317921      649.    -0.238
## 10 Afghanistan Asia       1997    41.8 22227415      635.    -0.0216
## # ... with 1,694 more rows
```

Finnum þau lönd og það ár þar sem væntur lífaldur lækkaði milli tveggja samliggjandi ára

```
gapminder %>%
  group_by(country) %>%
  filter(lifeExp < lag(lifeExp))
```

```
## # A tibble: 102 x 6
## # Groups:   country [52]
##    country  continent  year lifeExp     pop gdpPercap
##    <fct>    <fct>     <int>   <dbl>   <int>     <dbl>
##  1 Albania  Europe     1992    71.6 3326498     2497.
##  2 Angola   Africa     1987    39.9 7874230     2430.
##  3 Benin    Africa     2002    54.4 7026113     1373.
##  4 Botswana Africa     1992    62.7 1342614     7954.
##  5 Botswana Africa     1997    52.6 1536536     8647.
##  6 Botswana Africa     2002    46.6 1630347    11004.
##  7 Bulgaria Europe     1977    70.8 8797022     7612.
##  8 Bulgaria Europe     1992    71.2 8658506     6303.
##  9 Bulgaria Europe     1997    70.3 8066057     5970.
## 10 Burundi  Africa     1992    44.7 5809236      632.
## # ... with 92 more rows
```

Finnum rank á væntum lífaldri eftir nýjast árinu (athugið mínusinn á undan lifeExp til að fá rétta röð á rank)

```r
gapminder %>%
  filter(year == max(year)) %>%
  mutate(rank = min_rank(-lifeExp)) %>%
  arrange(rank)
```

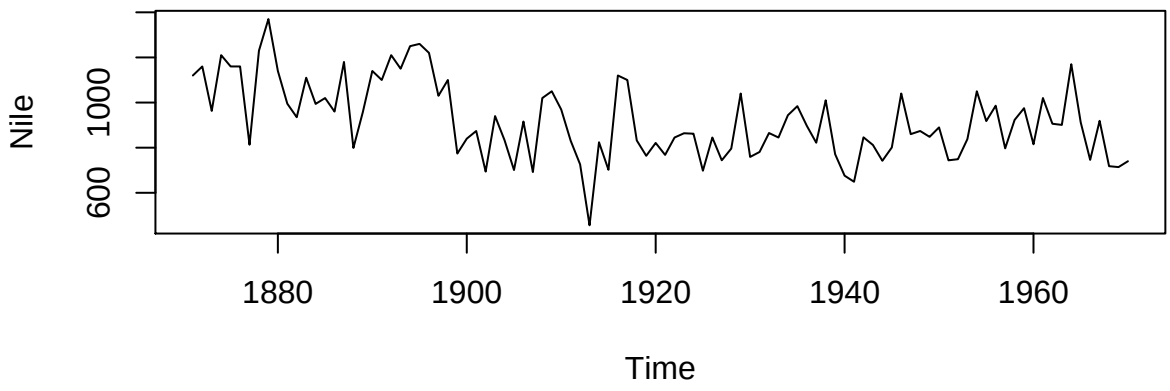
```
## # A tibble: 142 x 7
##    country          continent  year lifeExp       pop gdpPercap  rank
##    <fct>            <fct>     <int>   <dbl>     <int>     <dbl> <int>
##  1 Japan            Asia       2007    82.6 127467972    31656.     1
##  2 Hong Kong, China Asia       2007    82.2   6980412    39725.     2
##  3 Iceland          Europe     2007    81.8    301931    36181.     3
##  4 Switzerland      Europe     2007    81.7   7554661    37506.     4
##  5 Australia        Oceania    2007    81.2  20434176    34435.     5
##  6 Spain            Europe     2007    80.9  40448191    28821.     6
##  7 Sweden           Europe     2007    80.9   9031088    33860.     7
##  8 Israel           Asia       2007    80.7   6426679    25523.     8
##  9 France           Europe     2007    80.7  61083916    30470.     9
## 10 Canada           Americas   2007    80.7  33390141    36319.    10
## # ... with 132 more rows
```

5 Tími 2 - Myndræn framsetning með ggplot2

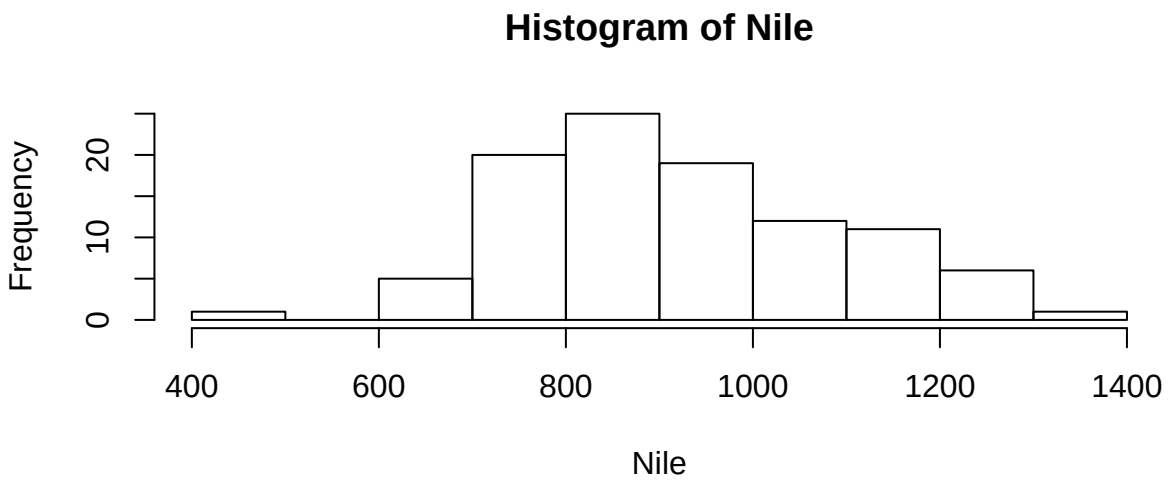
5.1 Base R

Með `plot()`, `hist()` og `boxplot()` má búa til einföld gröf á mjög einfaldan og snöggan máta

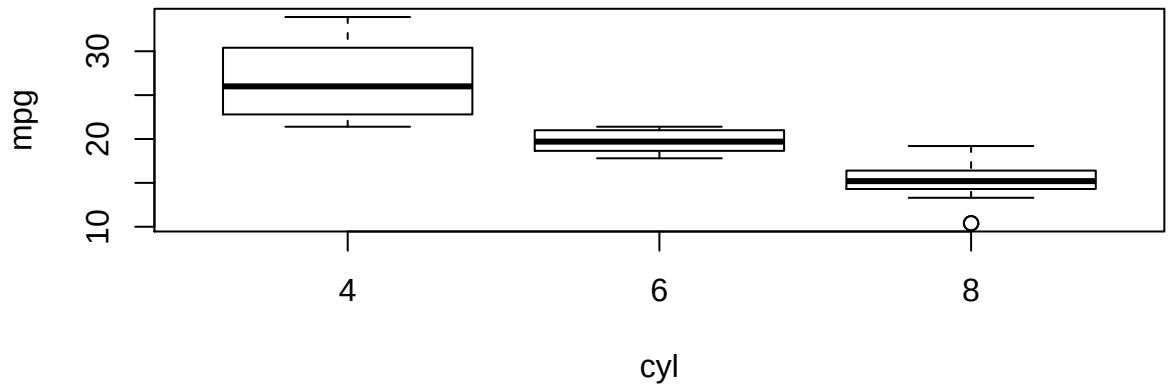
```r
plot(Nile)
```




```r
hist(Nile)
```




```
boxplot(mpg ~ cyl, data = mtcars)
```

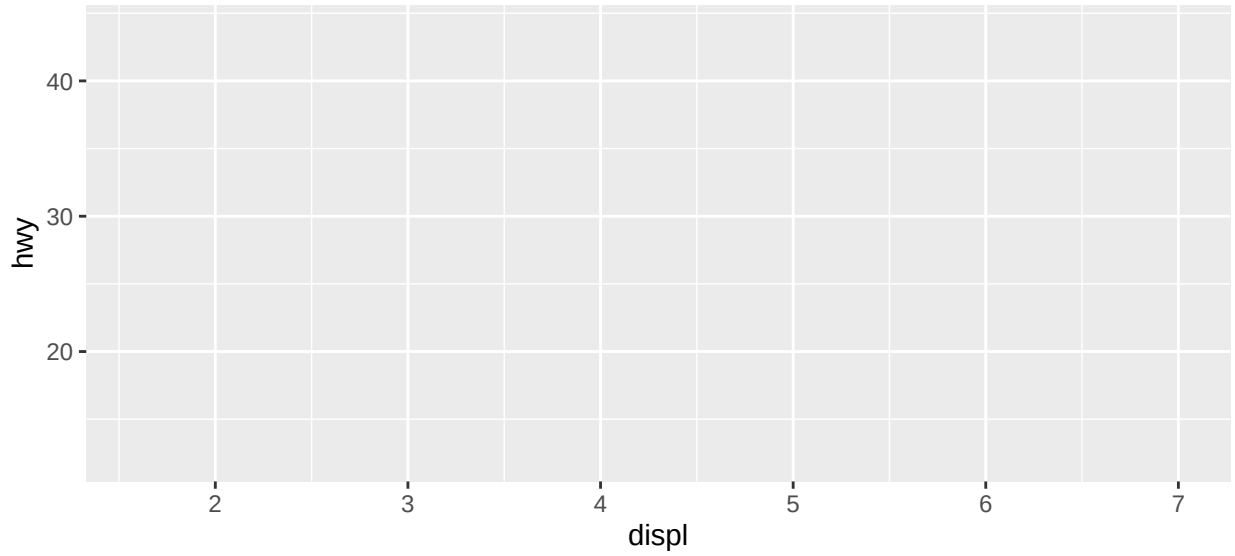


Án layers birtist ekkert

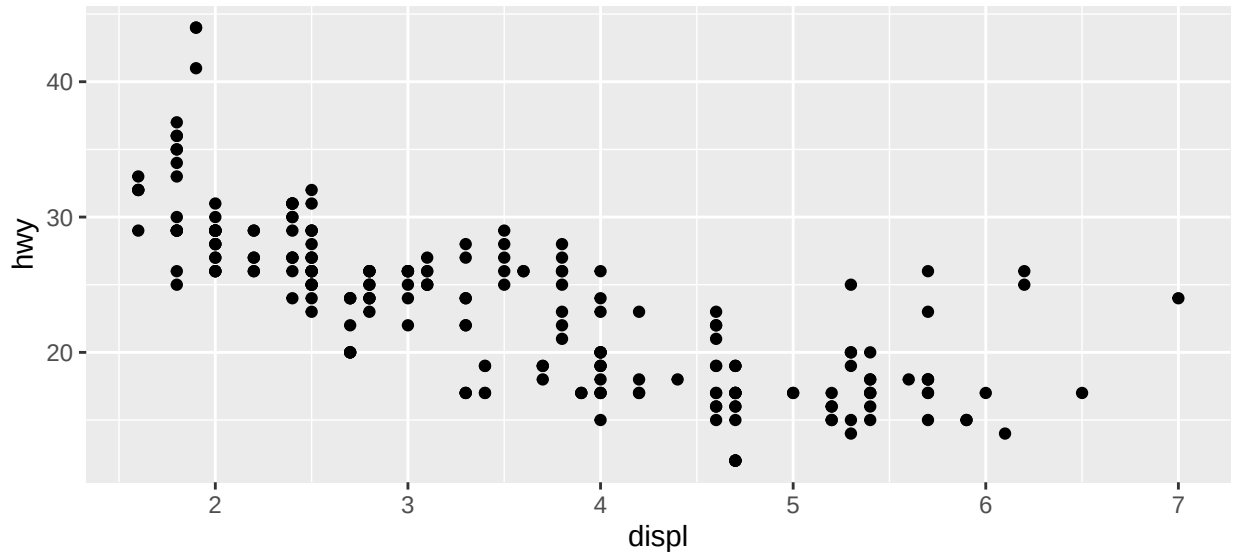
```
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
```

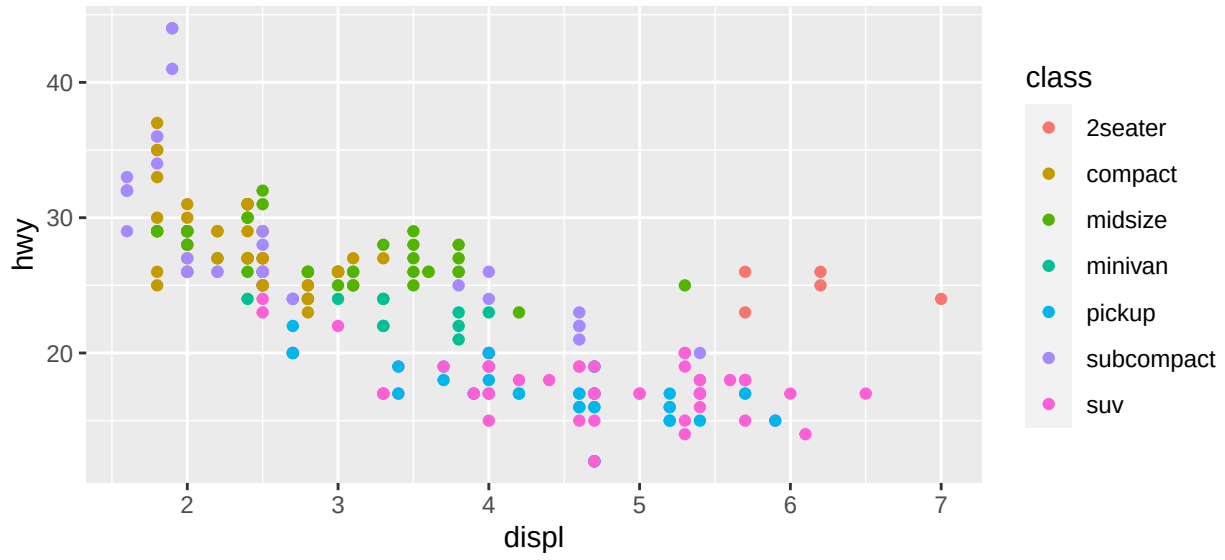
```
ggplot(mpg,
       aes(x = displ,
           y = hwy))
```




```
ggplot(mpg,
       aes(x = displ,
           y = hwy)) +
  geom_point()
```

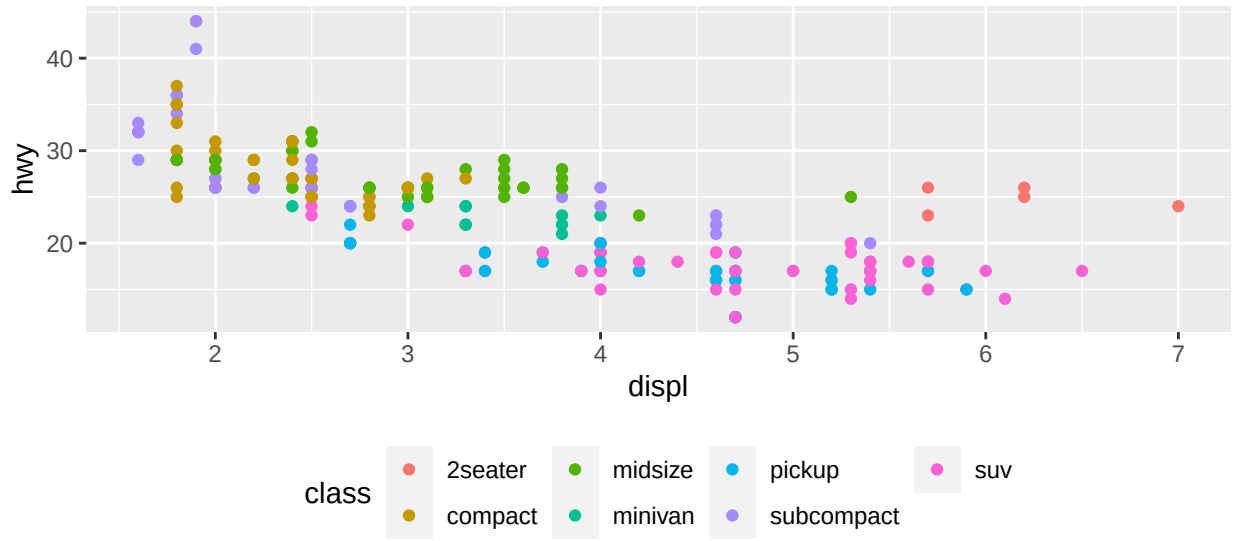



```
ggplot(mpg,
       aes(x = displ,
           y = hwy,
           col = class)) +
  geom_point()
```



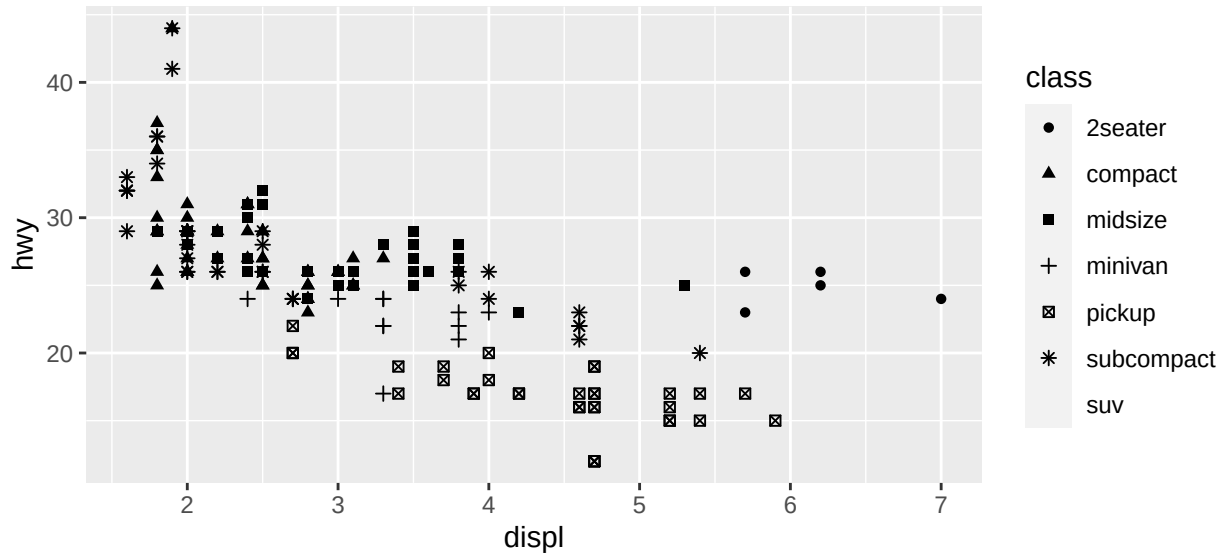
Ef við viljum breyta staðsetningu á legend notum við `theme()`.

```
ggplot(mpg,
       aes(x = displ,
           y = hwy,
           col = class)) +
  geom_point() +
  theme(legend.position = "bottom")
```



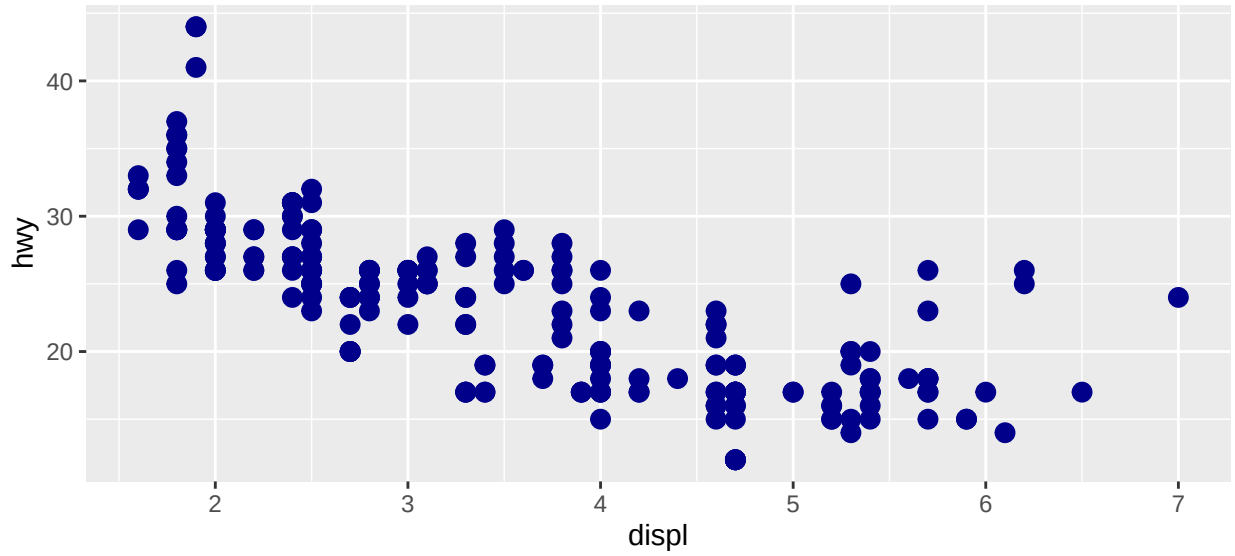
Athugið að ggplot notar aðeins sex mismunandi shape í einu, SUV dettur hér út

```
ggplot(mpg,
       aes(x = displ,
           y = hwy,
           shape = class)) +
  geom_point()
```



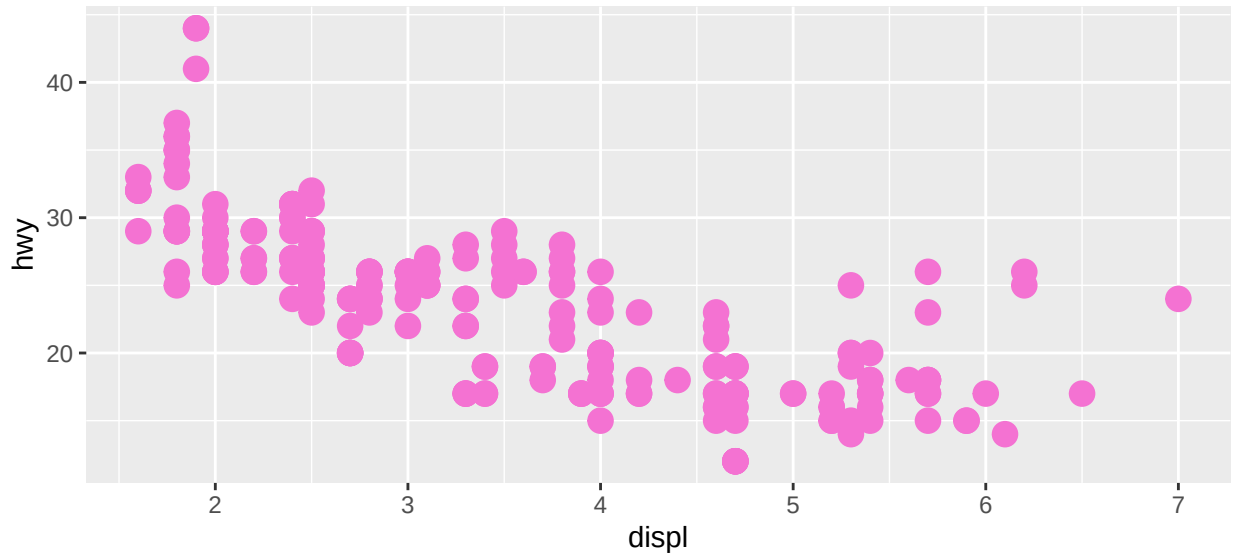
Getum einnig haldið okkur við einn lit en breytt honum. Hér hefur liturinn engar upplýsingar um neina breytu líkt og á undan. Því setjum við ekki litinn inn í `aes()` fallið heldur í `geom_point`.

```
ggplot(mpg,
       aes(x = displ,
           y = hwy)) +
  geom_point(col = "darkblue", size = 3)
```

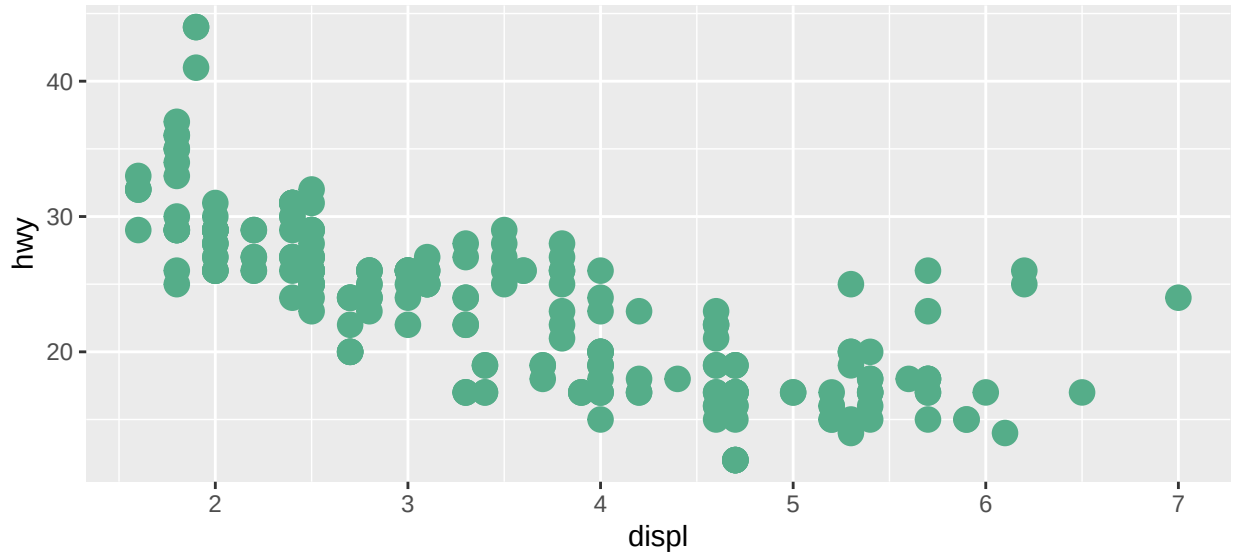


Getum líka notað RGB liti eða hex. Til að nota RGB liti notum við `rgb()` fallið.

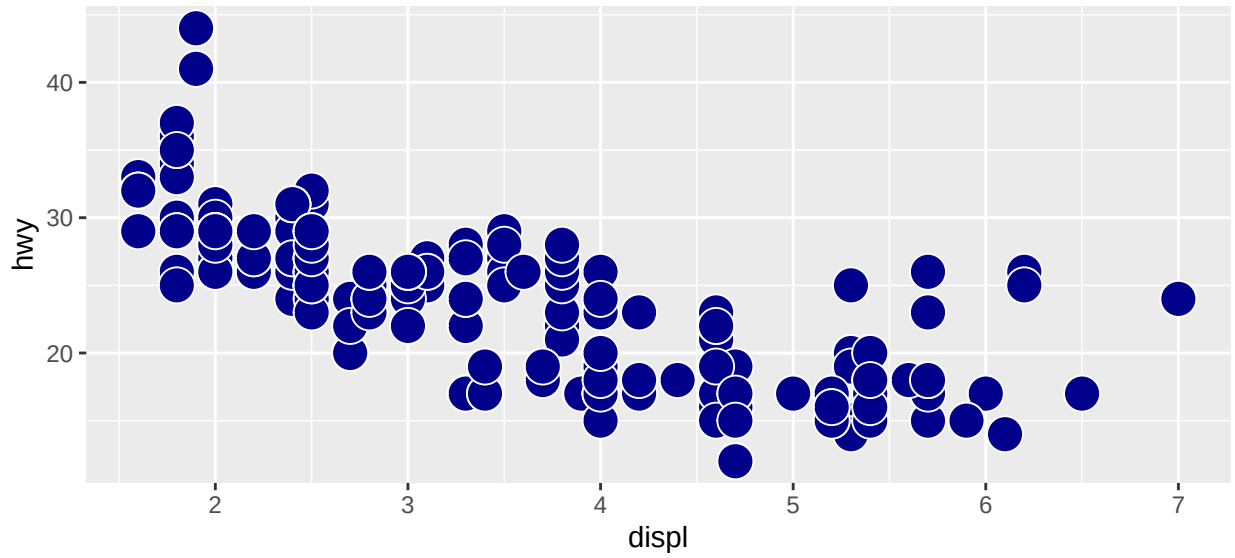
```
ggplot(mpg,
       aes(x = displ,
           y = hwy)) +
  geom_point(col = rgb(244, 114, 210, maxColorValue = 255), size = 4)
```




```
ggplot(mpg,
       aes(x = displ,
           y = hwy)) +
  geom_point(col = "#55ad89", size = 4)
```

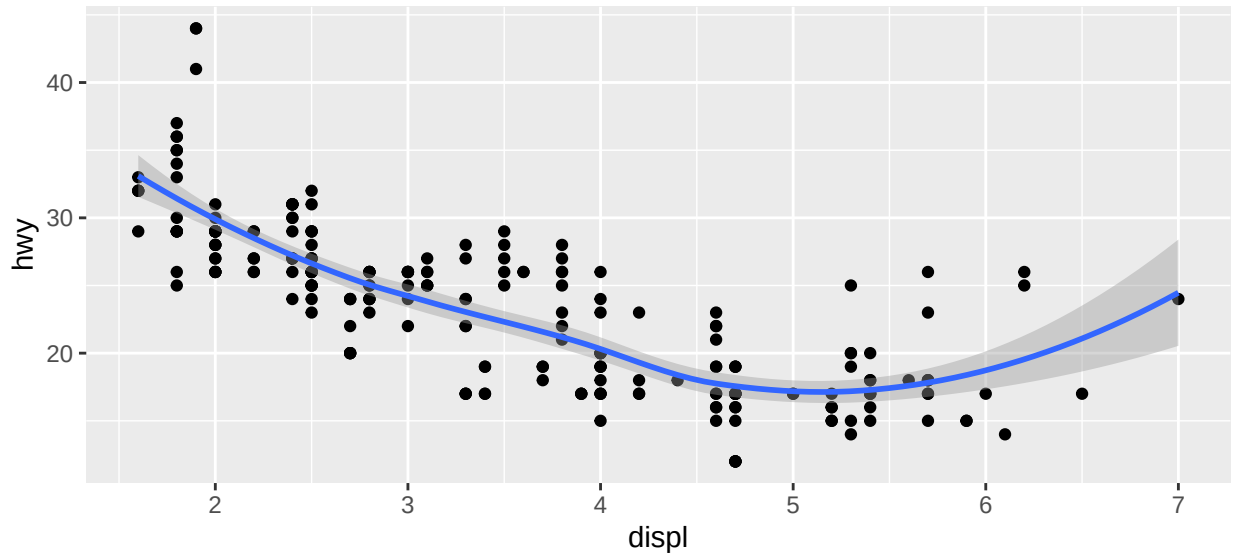



```
ggplot(mpg,
       aes(x = displ, y = hwy)) +
  geom_point(pch = 21,
             fill = "darkblue",
             col = "white",
             size = 6)
```

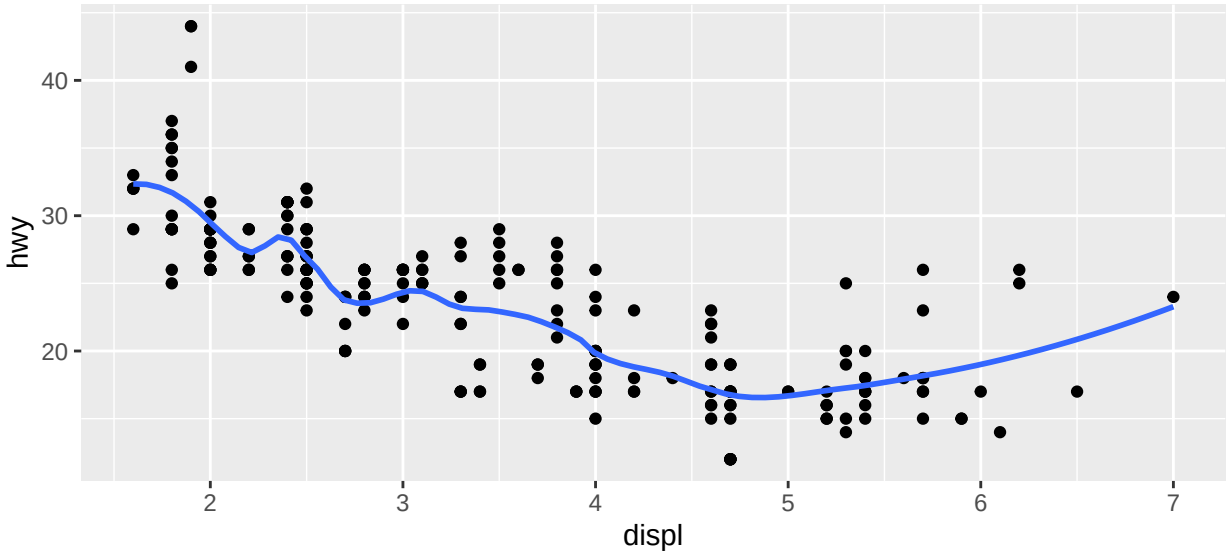


Inná scatter plot getum við bætt jöfnu bestu línu eða öðru til að lýsa leitni. Hversu mikið ferillinn sveigist er stjórnað með *span* argumentinu og hvort við viljum öryggisbil eða ekki með *se*

```
mpg %>%
  ggplot(aes(x = displ,
             y = hwy)) +
  geom_point() +
  geom_smooth()
```

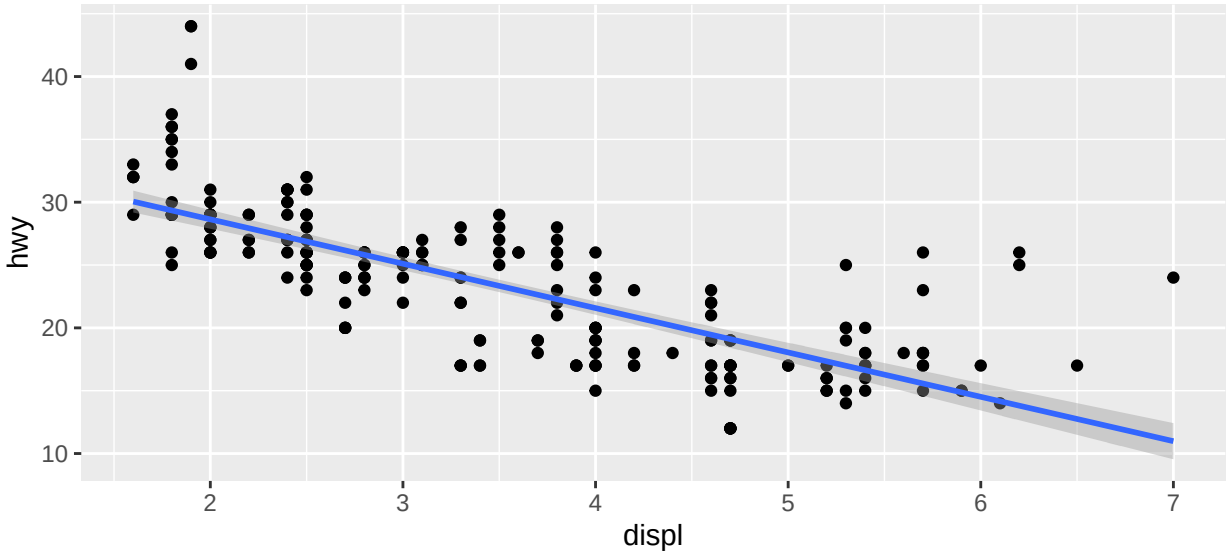



```
mpg %>%
  ggplot(aes(x = displ,
             y = hwy)) +
  geom_point() +
  geom_smooth(span = 0.3,
              se = FALSE)
```



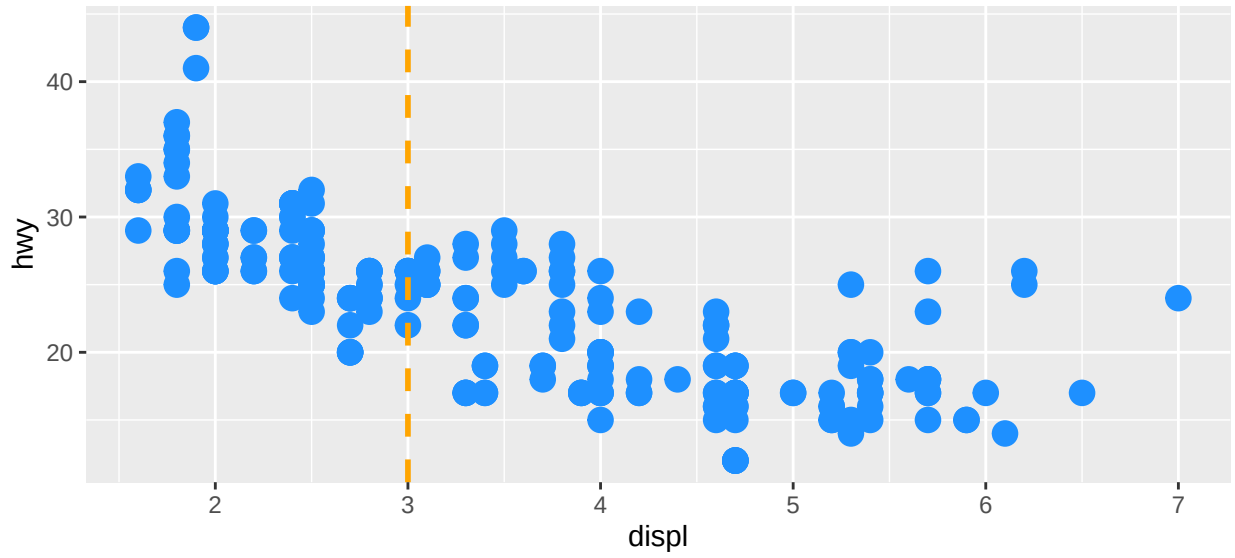
Ef við viljum halda okkur við beina línu getum við gert það með *method* argumentinu

```
mpg %>%
  ggplot(aes(x = displ,
             y = hwy)) +
  geom_point() +
  geom_smooth(method = "lm")
```

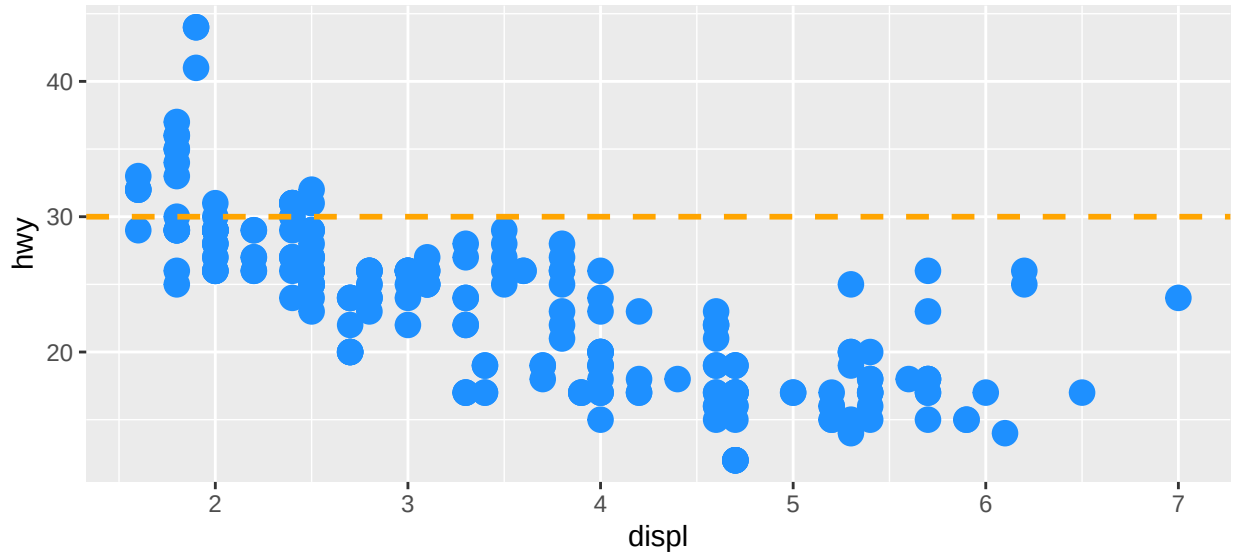


Ef við viljum bæta inn beinni línu, lóðréttri, láréttri eða með halla notum við `geom_hline()`, `geom_vline()` eða `geom_abline()`.

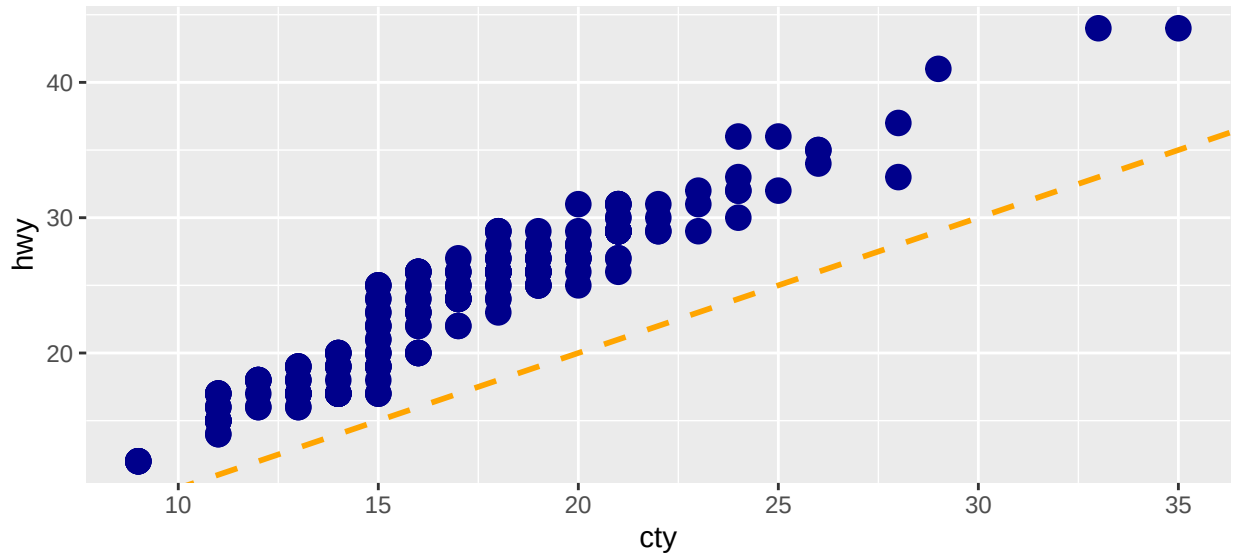
```
# geom_vline()
mpg %>%
  ggplot(aes(x= displ,
             y = hwy)) +
  geom_point(col = "dodgerblue", size = 4) +
  geom_vline(xintercept = 3, lwd = 1, linetype = "dashed", col = "orange")
```




```
# geom_hline()
mpg %>%
  ggplot(aes(x= displ,
             y = hwy)) +
  geom_point(col = "dodgerblue", size = 4) +
  geom_hline(yintercept = 30, lwd = 1, linetype = "dashed", col = "orange")
```

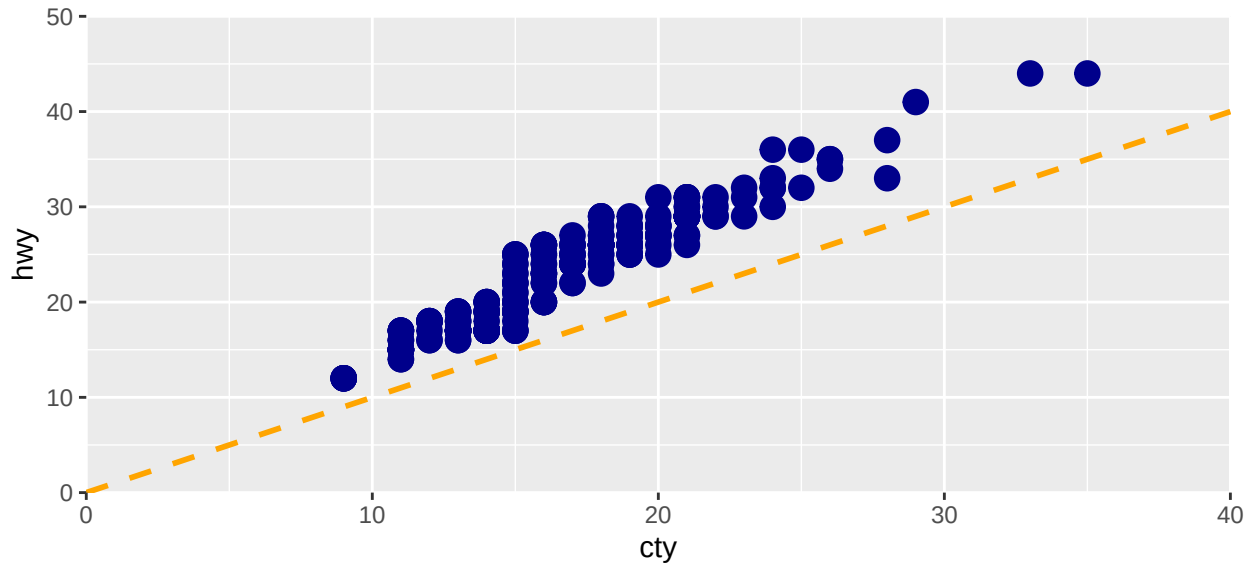



```
# geom_abline()
mpg %>%
  ggplot(aes(x = cty,
             y = hwy)) +
  geom_point(col = "darkblue", size = 4) +
  geom_abline(intercept = 0, slope =  1, lwd = 1, linetype = "dashed", col = "orange")
```



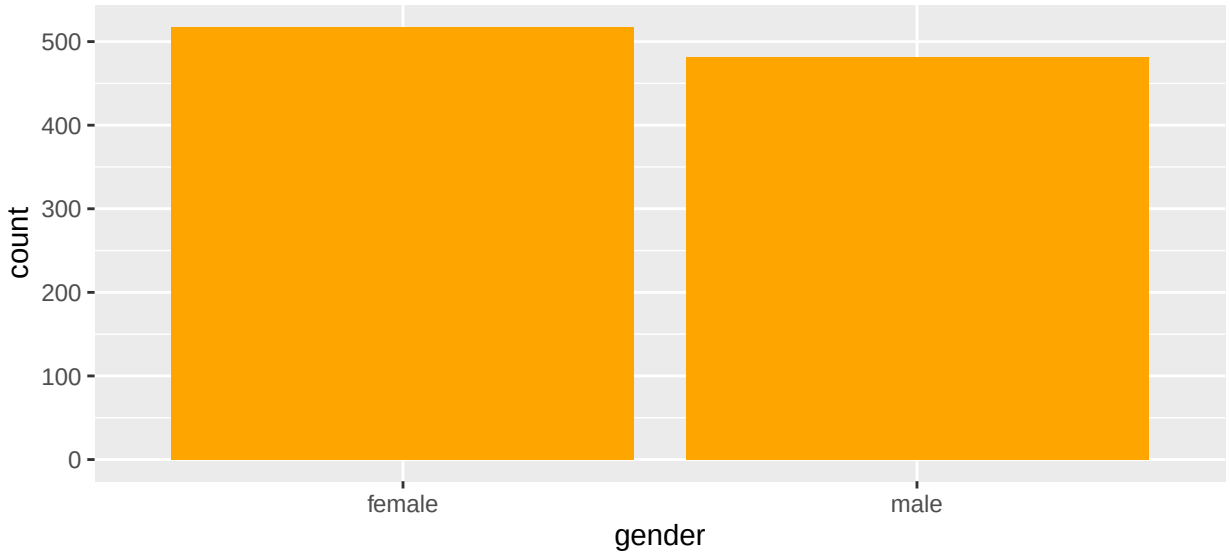
Til að gröf verði automatískt nice að horfa á þá sker ggplot í burtu autt pláss. Sjáum að x-ásinn byrjar í ~7.5 og y-ásinn í 10. Ef við viljum að ásarnir byrji í núlli er auðveldast að gera það með eftirfarandi hætti. Takið eftir að þrátt fyrir að hafa sagt grafinu að byrja í 0 á báðum ásum þá sjáum við aðeins fyrir neðan núll.

```
mpg %>%
  ggplot(aes(x = cty,
             y = hwy)) +
  geom_point(col = "darkblue", size = 4) +
  geom_abline(intercept = 0, slope =  1, lwd = 1, linetype = "dashed", col = "orange") +
  coord_cartesian(ylim = c(0, 50), xlim = c(0, 40)) +
  scale_y_continuous(expand = c(0, 0)) +
  scale_x_continuous(expand = c(0, 0))
```



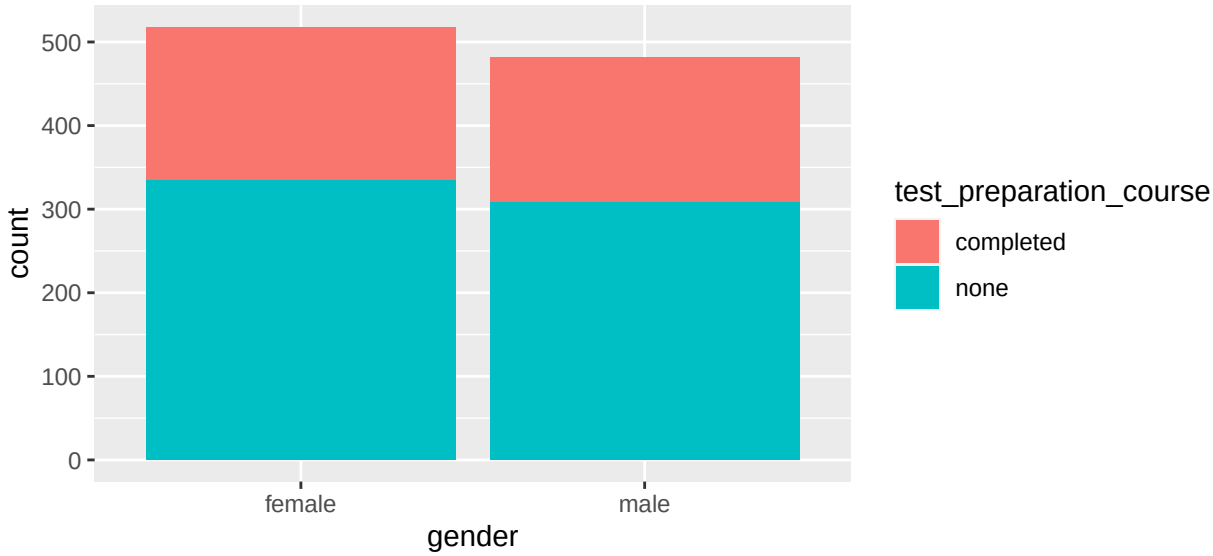
Þegar við notum bar chart eða histogram þá notum við **fill** argumentið til að stjórna litnum. Color argumentið stjórnar línunni utan um stöplana.

```r
exam <- read_csv("StudentsPerformance.csv") %>% janitor::clean_names()

ggplot(exam,
       aes(x = gender)) +
  geom_bar(fill = "orange")
```



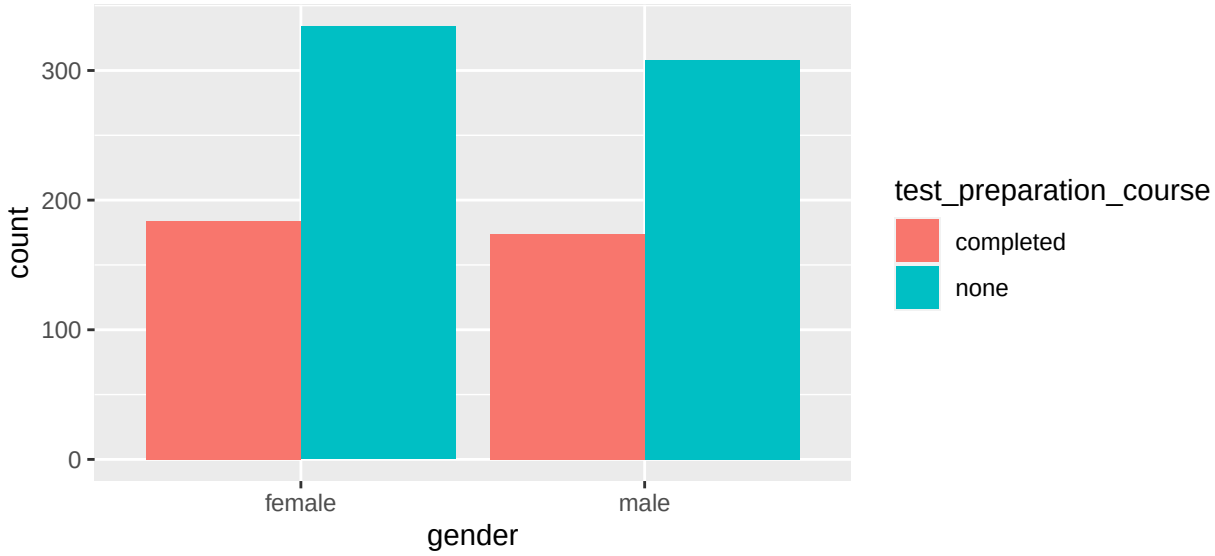
Getum skipt körlum og konum enn meira niður, t.d. eftir því hvort þau tóku undirbúningskúrs eða ekki.

```r
exam %>%
  ggplot(aes(x = gender,
             fill = test_preparation_course)) +
  geom_bar()
```



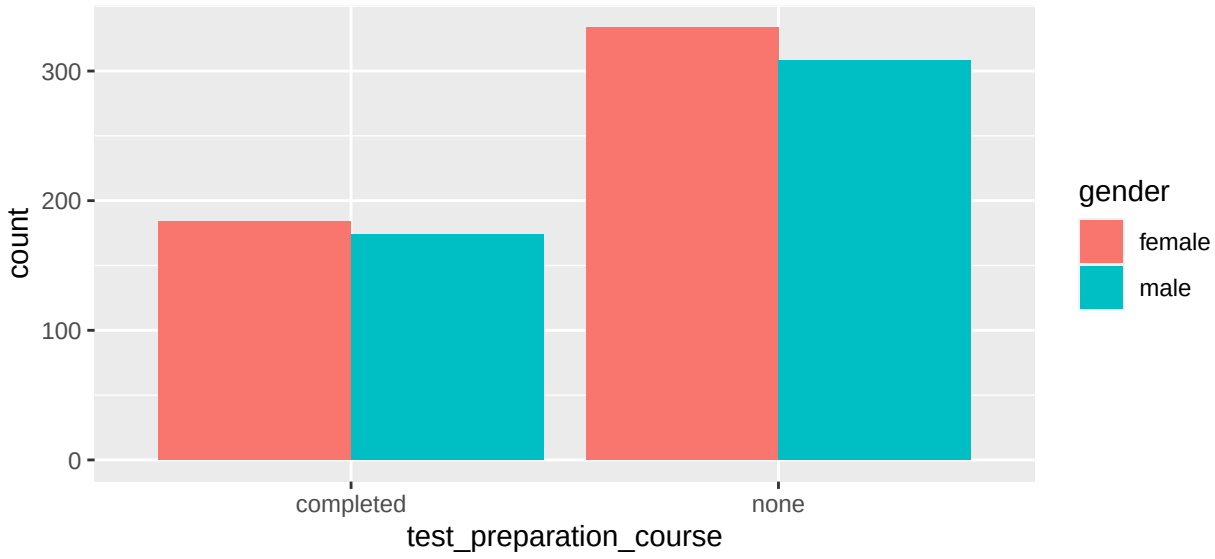
Eflaust meira lýsandi að stilla þeim upp hlið við hlið

```
exam %>%
  ggplot(aes(x = gender,
             fill = test_preparation_course)) +
  geom_bar(position = "dodge")
```



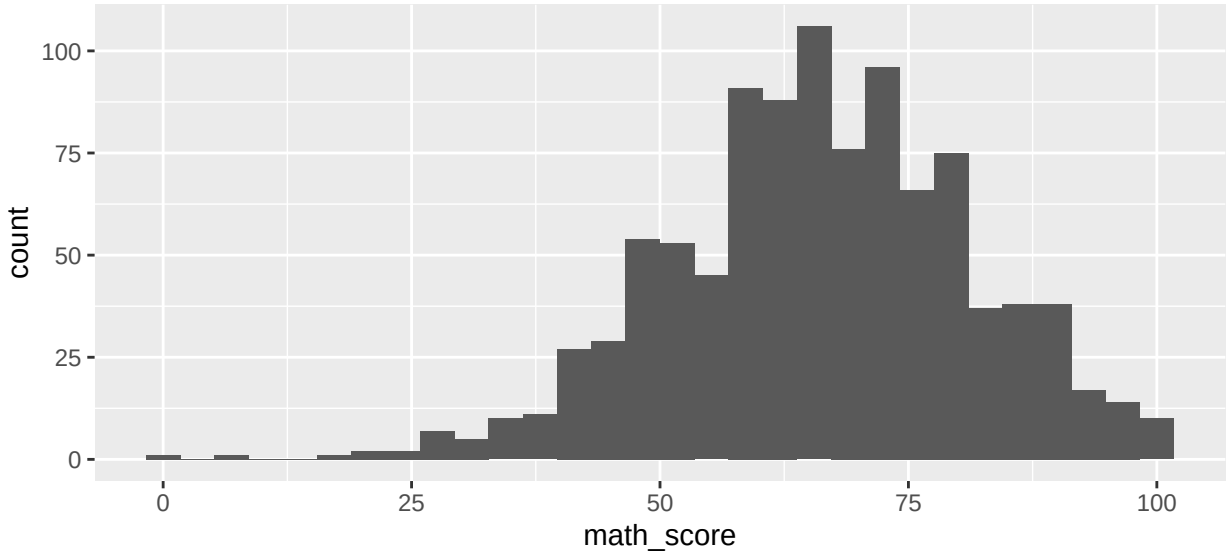
Getum skipt á gender og test_preparation_course

```
exam %>%
  ggplot(aes(x = test_preparation_course,
             fill = gender)) +
  geom_bar(position = "dodge")
```



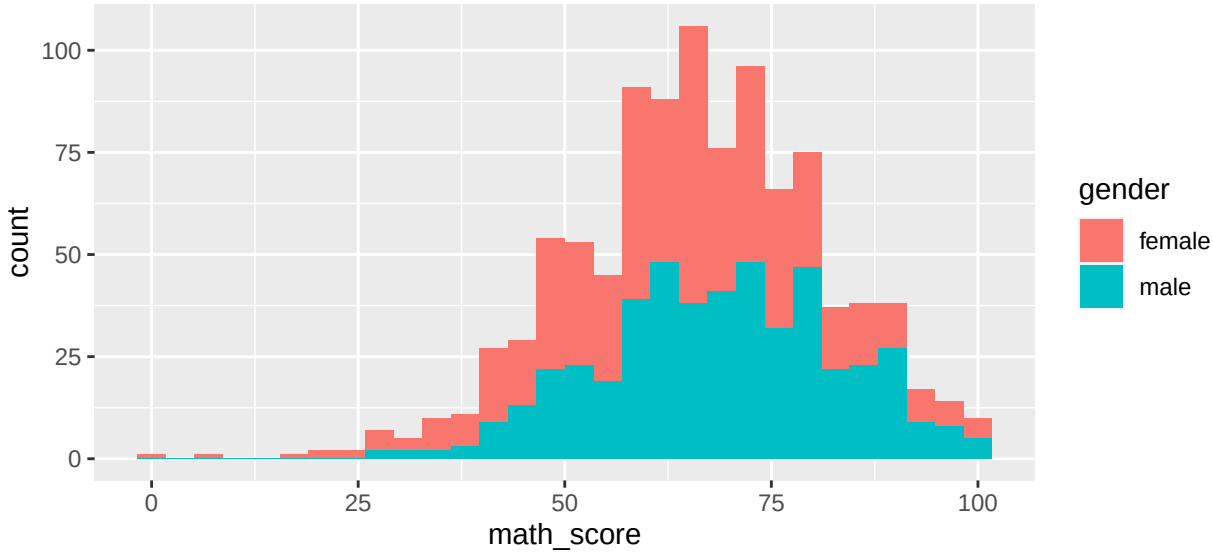
Skoðum dreifingu á niðurstöðum úr stærðfræðinni

```
exam %>%
  ggplot(aes(x = math_score)) +
  geom_histogram()
```



Skoðum eftir kyni. Ekki mjög gagnlegt að gera þetta svona

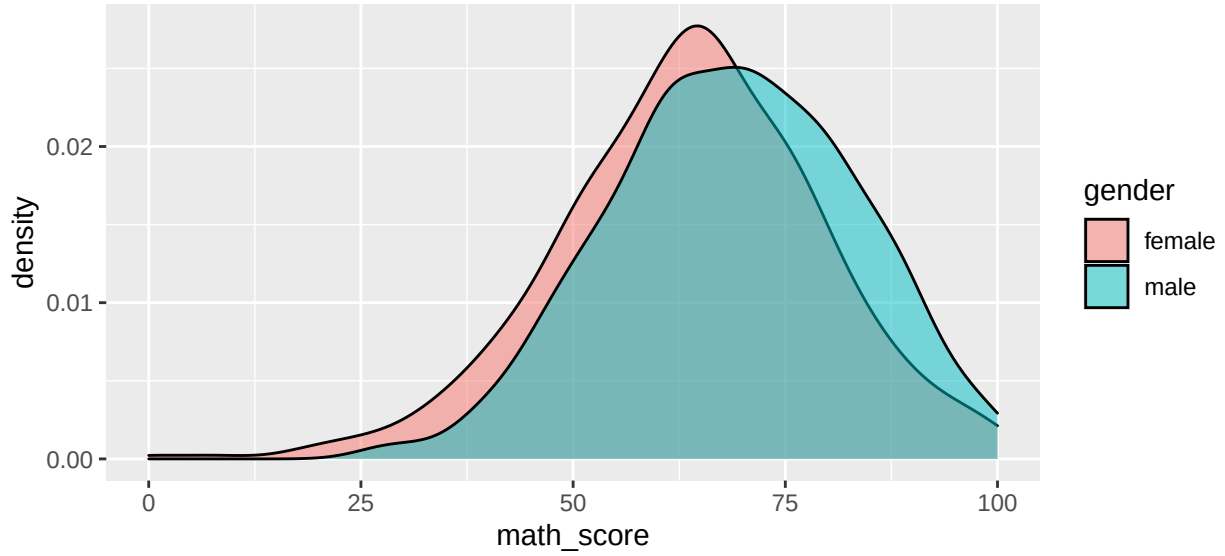
```
exam %>%
  ggplot(aes(x = math_score,
             fill = gender)) +
  geom_histogram()
```



Skára að nota density

```
exam %>%
  ggplot(aes(x = math_score,
```

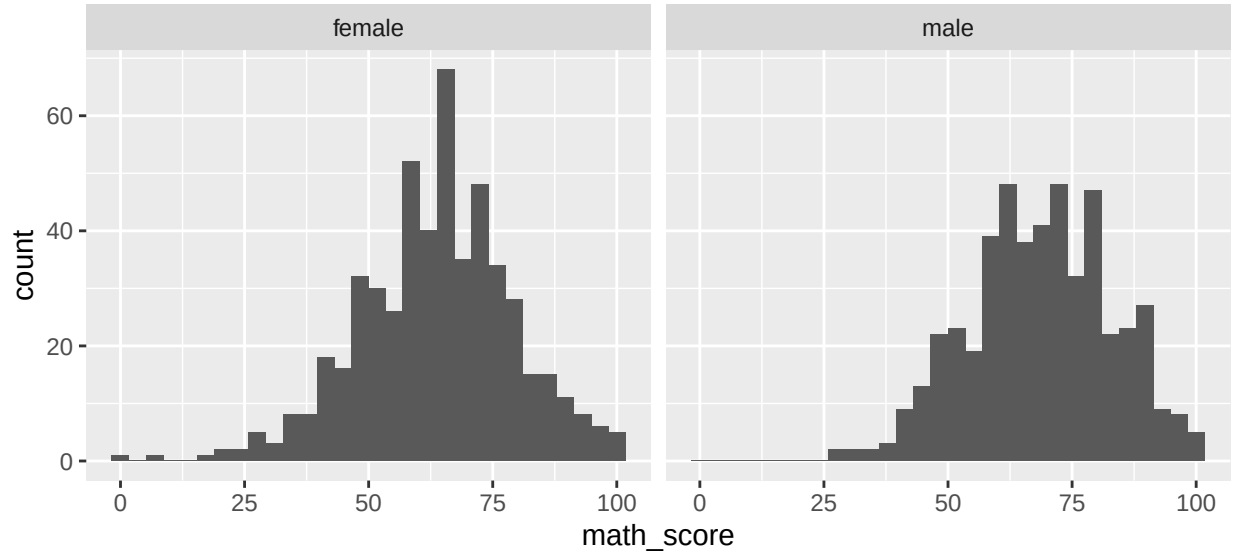
```
            fill = gender)) +
  geom_density(alpha = 0.5)
```



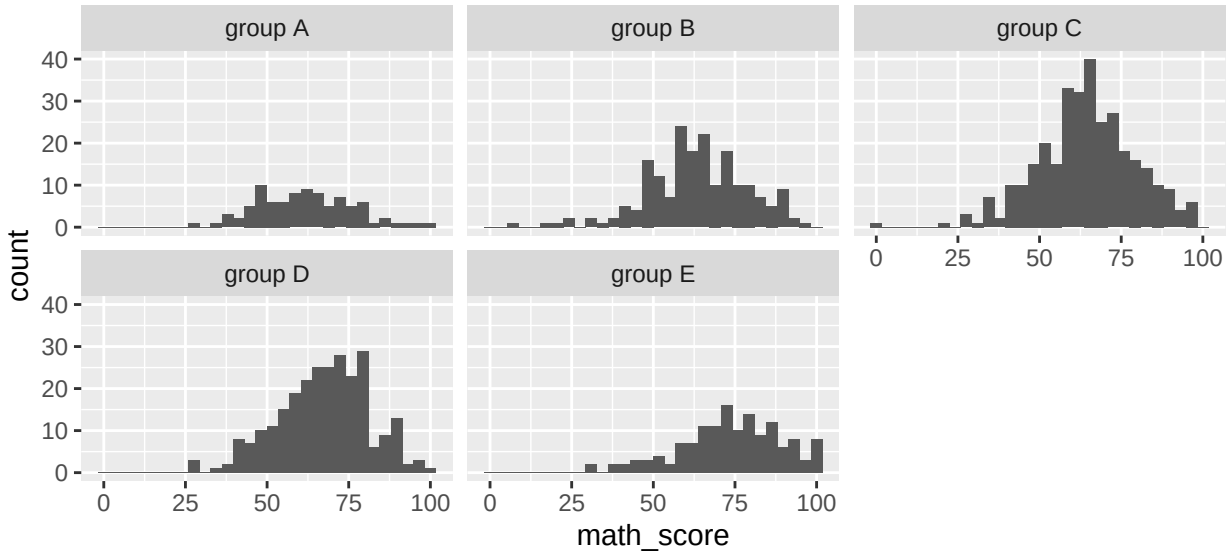
5.2 Skoðum faceting (opna glærur fyrst)

`facet_wrap()` er notað til að skipta gögnunum upp í litlar myndir út frá einni categorical breytu

```
exam %>%
  ggplot(aes(x = math_score)) +
  geom_histogram() +
  facet_wrap(~ gender) # sýna líka ncol = 1
```

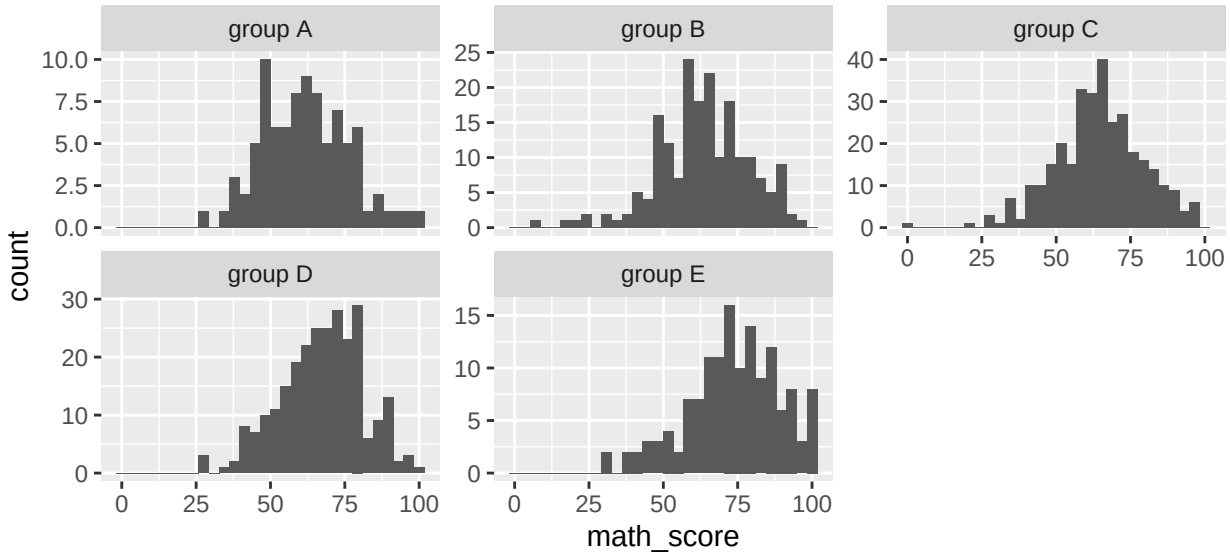


```
exam %>%
  ggplot(aes(x = math_score)) +
  geom_histogram() +
  facet_wrap(~ race_ethnicity)
```



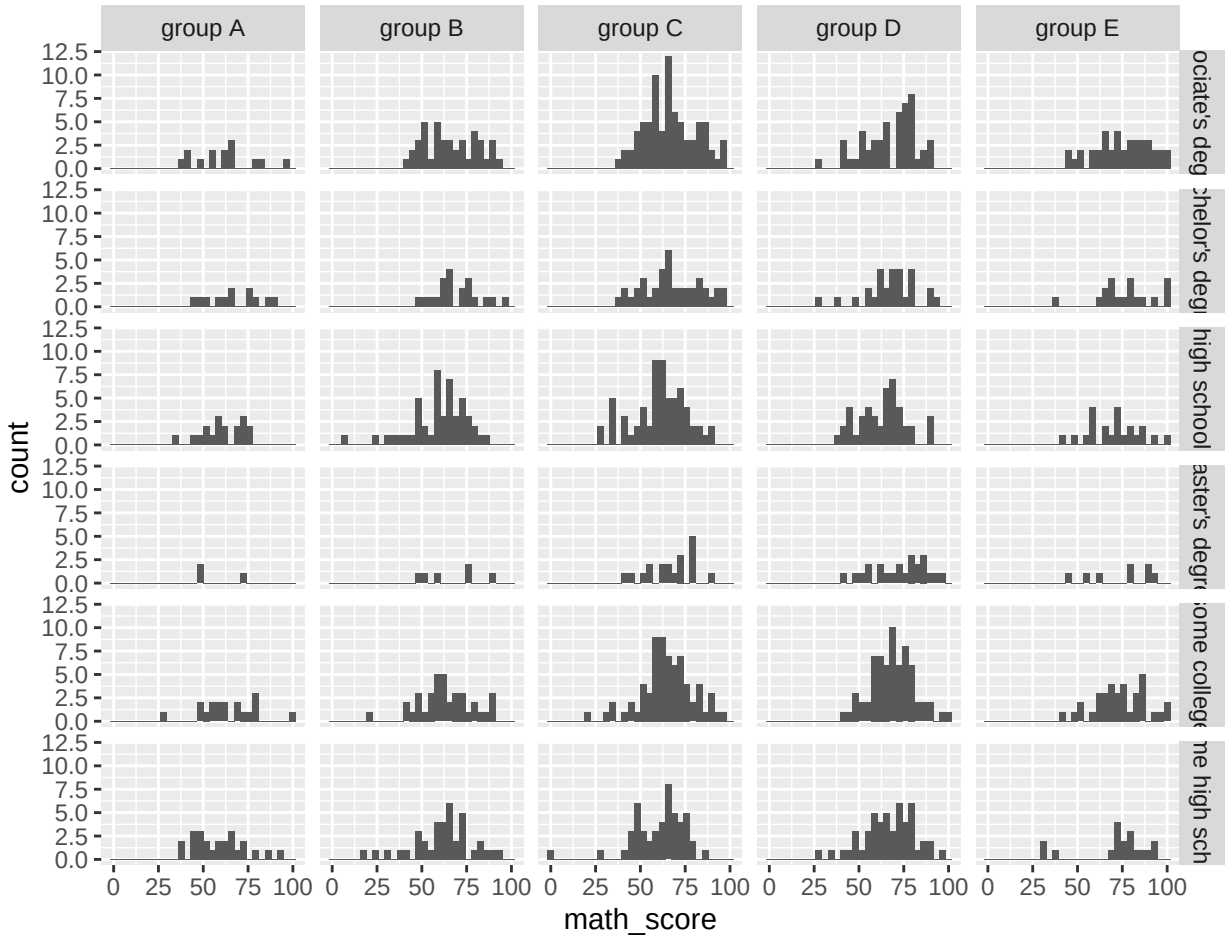
Það eru mjög fái í group A, sjáum það á hæð stöplana. Þetta er mjög algengt þegar facet er notað. Getum lagað þetta með **scales** skipuninni inní `facet_wrap()`.

```
exam %>%
  ggplot(aes(x = math_score)) +
  geom_histogram() +
  facet_wrap(~ race_ethnicity, scales = "free_y")
```



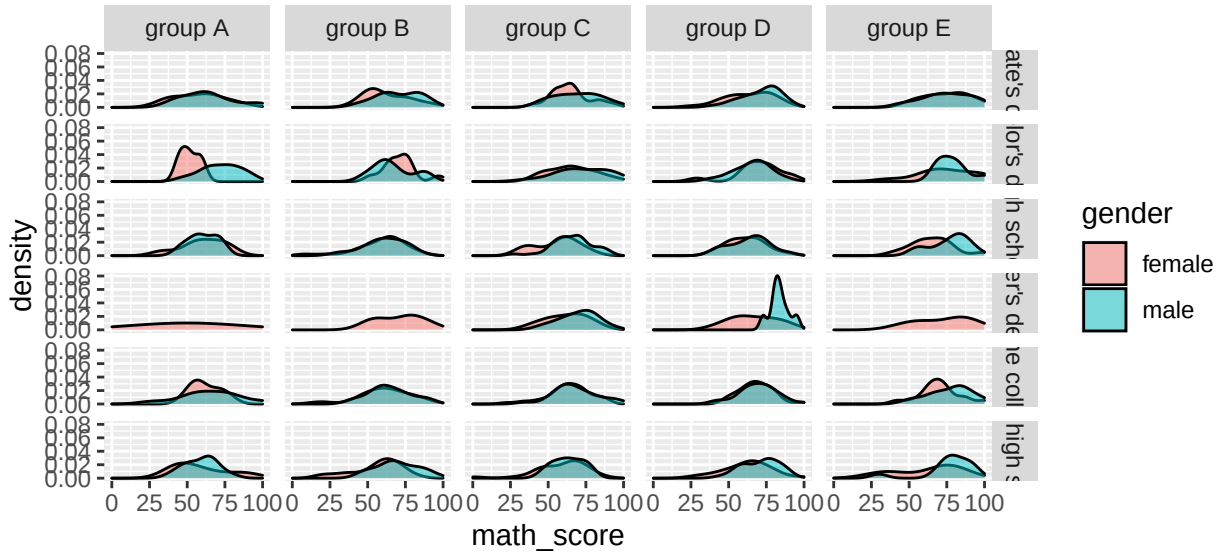
`facet_grid()` er mjög líkt `facet_wrap()` nema býður upp á það að nota tvær categorical breytur

```
exam %>%
  ggplot(aes(x = math_score)) +
  geom_histogram() +
  facet_grid(parental_level_of_education ~ race_ethnicity)
```



Göngum enn lengar og bætum kyni við

```
exam %>%
  ggplot(aes(x = math_score, fill = gender)) +
  geom_density(alpha = 0.5) +
  facet_grid(parental_level_of_education ~ race_ethnicity)
```



Stundum gengur illa að skipta gögnunum upp í grid og skoða t.d. dreifingu. Þetta á við þegar einn hópurinn er mjög fjölmennur. Þá er hægt að nota *scales* argumentið í `facet_wrap()` og `facet_grid()`.

5.3 Column chart

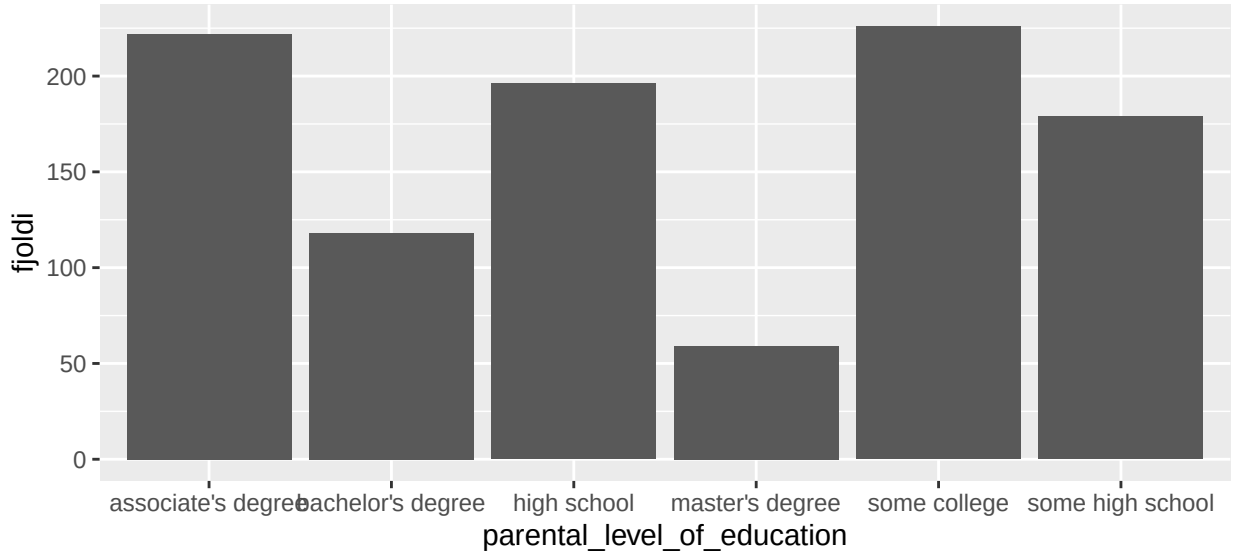
Að ofan höfum við notað `geom_histogram()` og `geom_bar()`. Bæði föllin telja fyrir okkur og birta niðurstöðurnar.

Oft er það þó þannig að við viljum ekki láta R telja fyrir okkur heldur erum við með upplýsingarnar tilbúnar hjá okkur.

```
exam_calc <- exam %>%
  group_by(parental_level_of_education) %>%
  summarise(fjoldi = n())

exam_calc
```

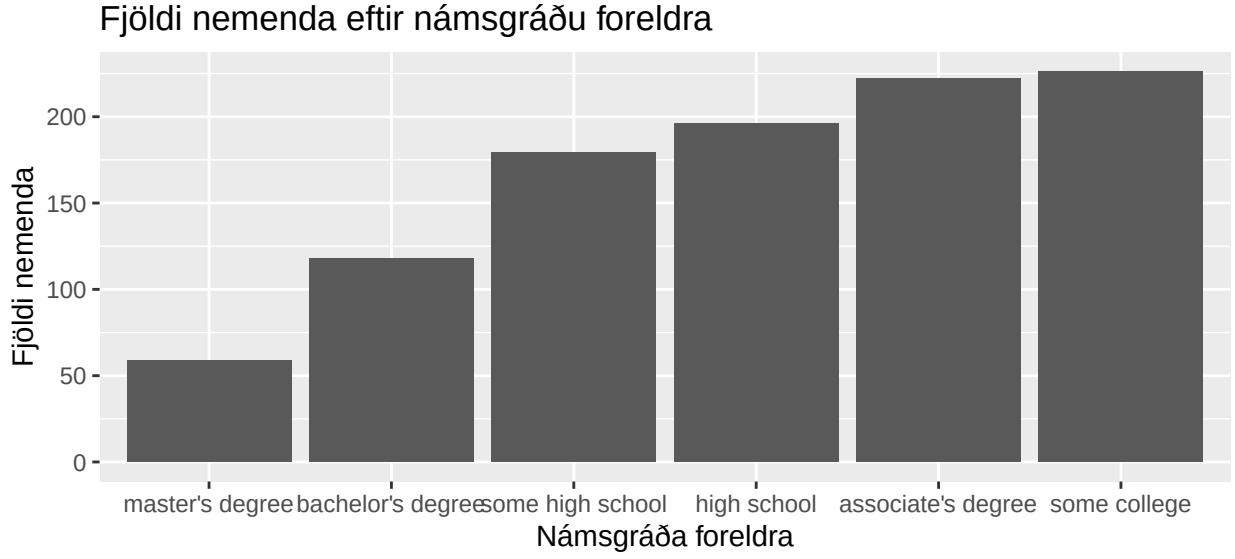
```
## # A tibble: 6 x 2
##   parental_level_of_education fjoldi
##   <chr>                         <int>
## 1 associate's degree              222
## 2 bachelor's degree               118
## 3 high school                     196
## 4 master's degree                  59
## 5 some college                    226
## 6 some high school                179
```

```
exam_calc %>%
  ggplot(aes(x = parental_level_of_education,
             y = fjoldi)) +
  geom_col()
```



Hér er stöplunum raðað eftir stafrófsröð. Gætum vilja raða eftir fjölda. Getum breytt röðuninni með því að setja mínus (-) fyrir framan fjolda

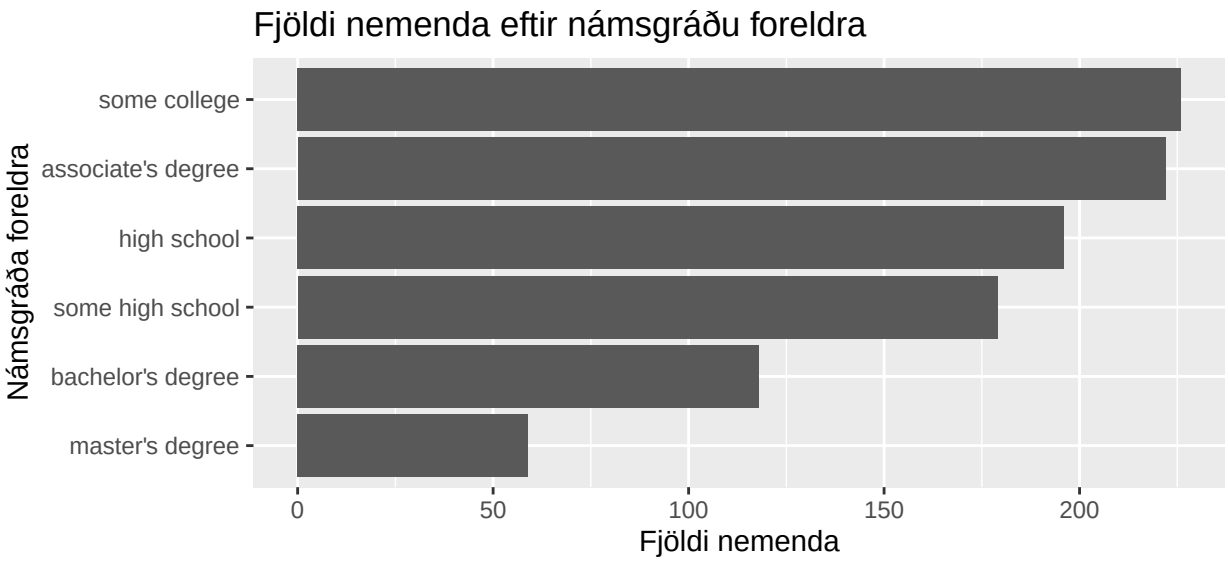
```
exam_calc %>%
  ggplot(aes(x = fct_reorder(parental_level_of_education, fjoldi),
             y = fjoldi)) +
  geom_col() +
  labs(x = "Námsgráða foreldra", y = "Fjöldi nemenda",
       title = "Fjöldi nemenda eftir námsgráðu foreldra")
```



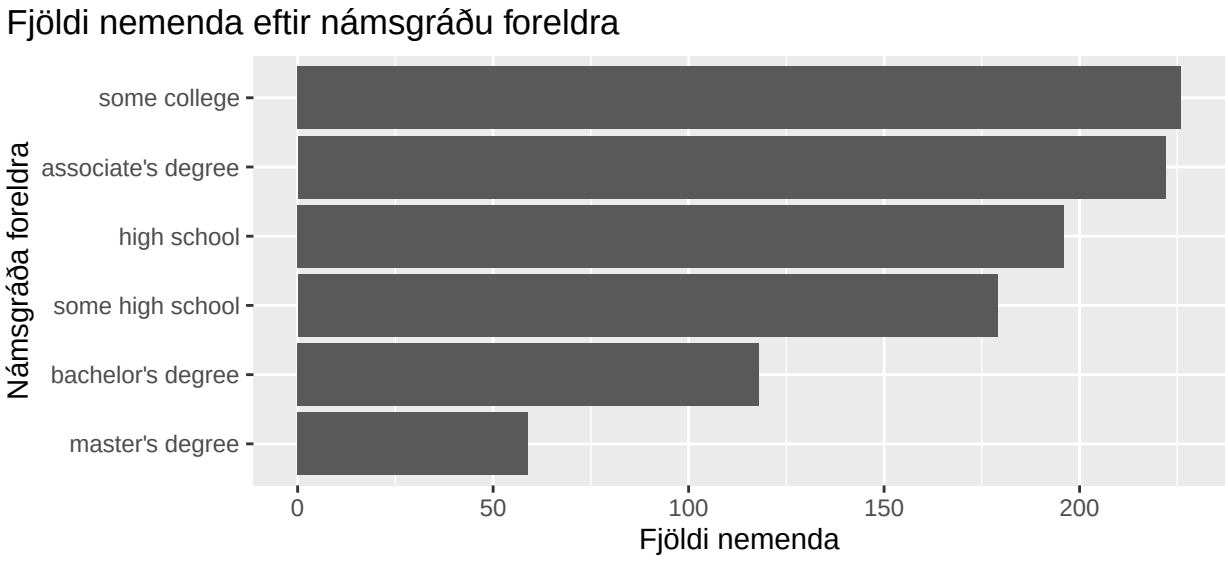
Það sem einnig er mjög algengt er að nöfnin fyrir x breytuna eru mjög löng. Athugið að hér er titillinn á grafinu yfir plot-area en byrjar ekki alveg lengst til vinstri. Getum lagað það.

```
exam_calc %>%
  ggplot(aes(x = fct_reorder(parental_level_of_education, fjoldi),
             y = fjoldi)) +
```

```
  geom_col() +
  labs(x = "Námsgráða foreldra", y = "Fjöldi nemenda",
       title = "Fjöldi nemenda eftir námsgráðu foreldra") +
  coord_flip()
```



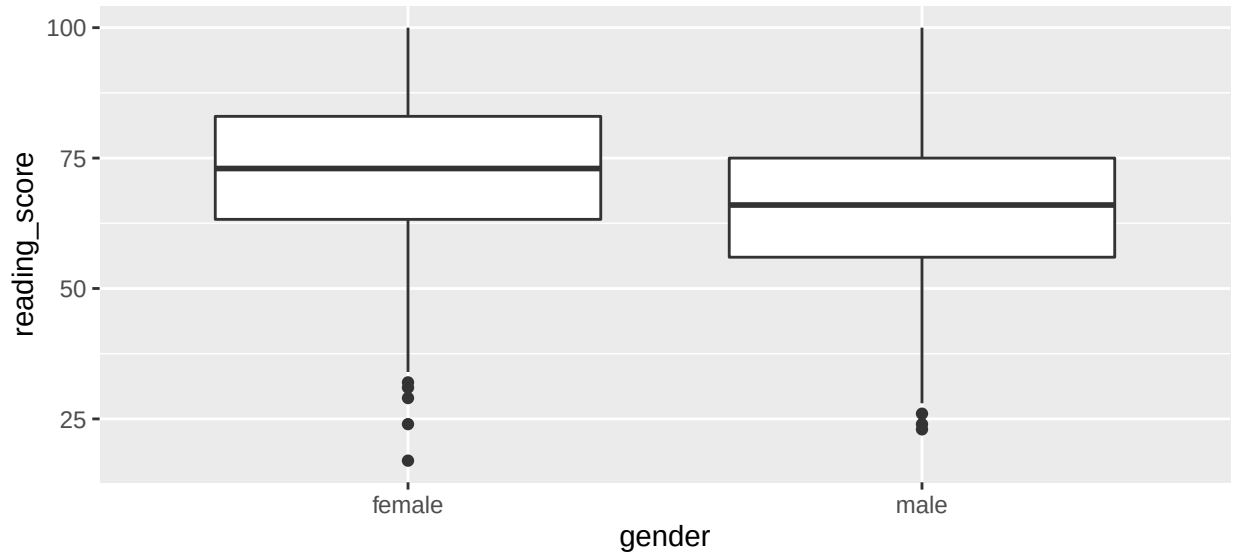

```
exam_calc %>%
  ggplot(aes(x = fct_reorder(parental_level_of_education, fjoldi),
             y = fjoldi)) +
  geom_col() +
  labs(x = "Námsgráða foreldra", y = "Fjöldi nemenda",
       title = "Fjöldi nemenda eftir námsgráðu foreldra") +
  coord_flip() +
  theme(plot.title.position = "plot")
```



5.4 Boxplot

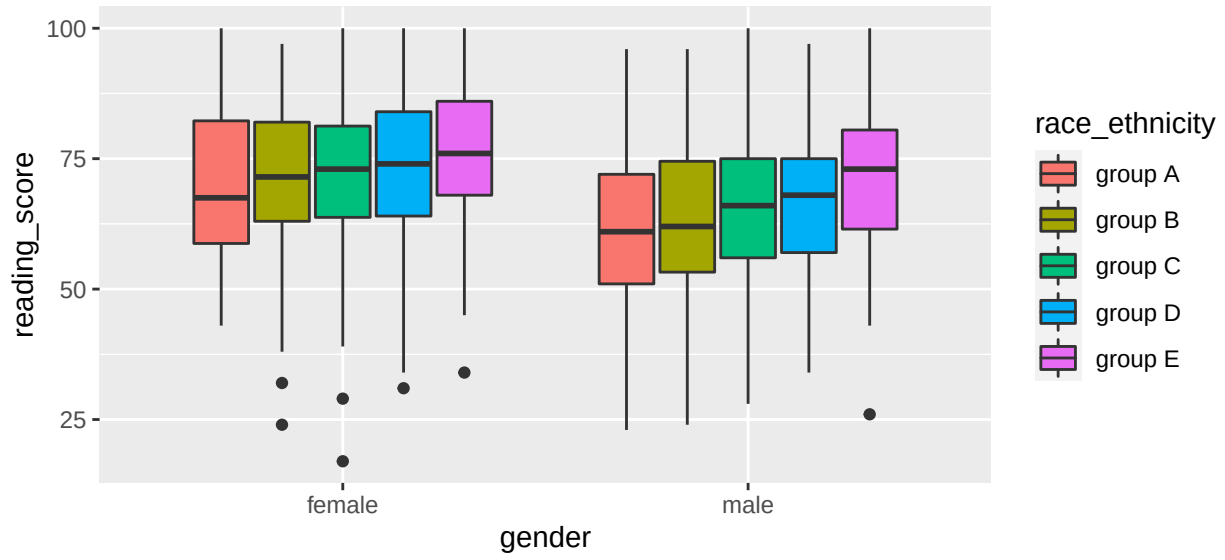
Neðangreint graf kallast box-plot. Box plot samanstendur af *hinges* og *whiskers*. Hinges eru efri og neðri mörk kassans. Efri mörkin eru efri fjórðungsmörk og neðri mörkin eru neðri fjórðungsmörk dreifingarinnar. Neðri fjórðungsmörk eru skilgreind á þann hátt að 25% gildanna eru undir neðri fjórðungsmörkum. Á sama hátt eru 25% gildanna yfir efri fjórðungsmörkunum. Whiskers eru lóðréttu strikin sem koma upp úr og niður úr boxinu. Efra strikið nær frá efri mörkum boxins upp í hæsta gildið þó aldrei hærra en 1.5 x IQR (inter quartile range eða bil milli efri og neðri marka boxins), sama á við um neðra strikið. Gildi sem fara út fyrir whiskers kallast útlagar. Svarta línan fyrir miðju kassans er miðgildi dreifingarinnar. Miðgildi er skilgrient þannig að 50% af gildunum eru fyrir ofan miðgildið og 50% eru fyrir niðan.

```
exam %>%
  ggplot(aes(x = gender, y = reading_score)) +
  geom_boxplot()
```



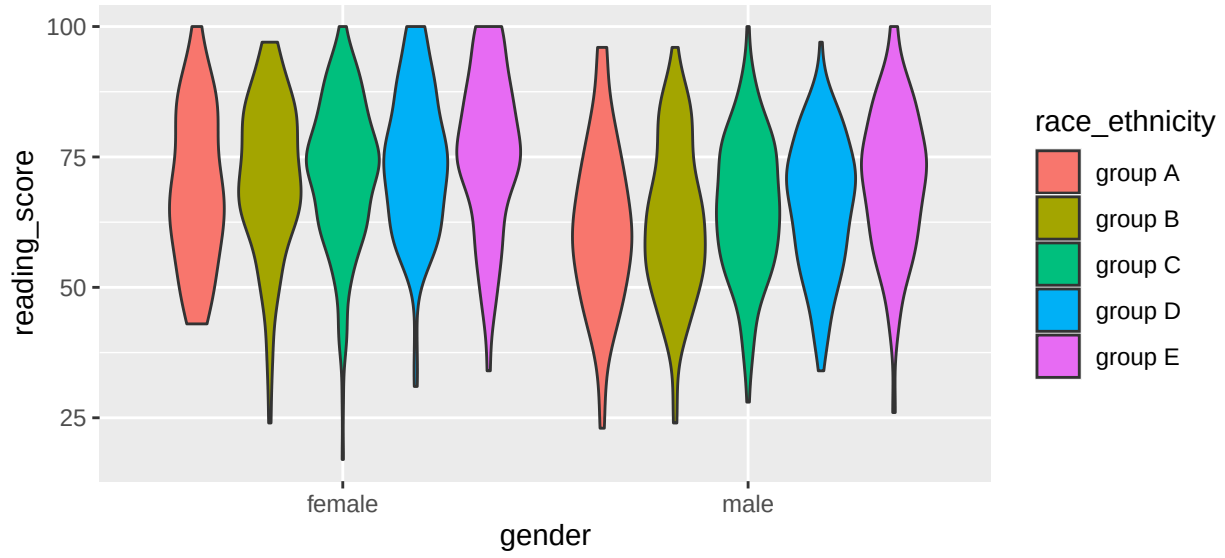
Með því að bæta við **fill** fyrir aðra breytu getum við borið saman dreifingar innan kynja

```
exam %>%
  ggplot(aes(x = gender, y = reading_score, fill = race_ethnicity)) +
  geom_boxplot()
```



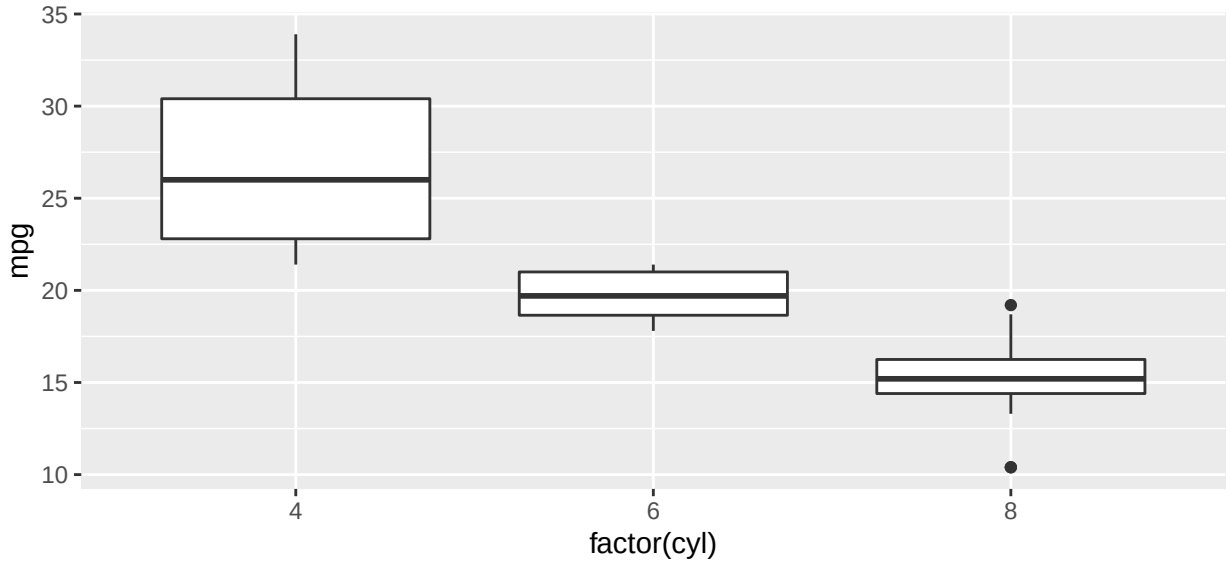
Tölfræðingar eru ekki sammála um gæði boxplot-a. Ókostir boxplot er að við sjáum ekki almennilega dreifinguna. Hún gæti verið tvítoppa. Þá segir boxplot-ið lítið. Önnur leið til að skoða dreifingar er að nota `geom_violin`

```
exam %>%
  ggplot(aes(x = gender, y = reading_score, fill = race_ethnicity)) +
  geom_violin()
```

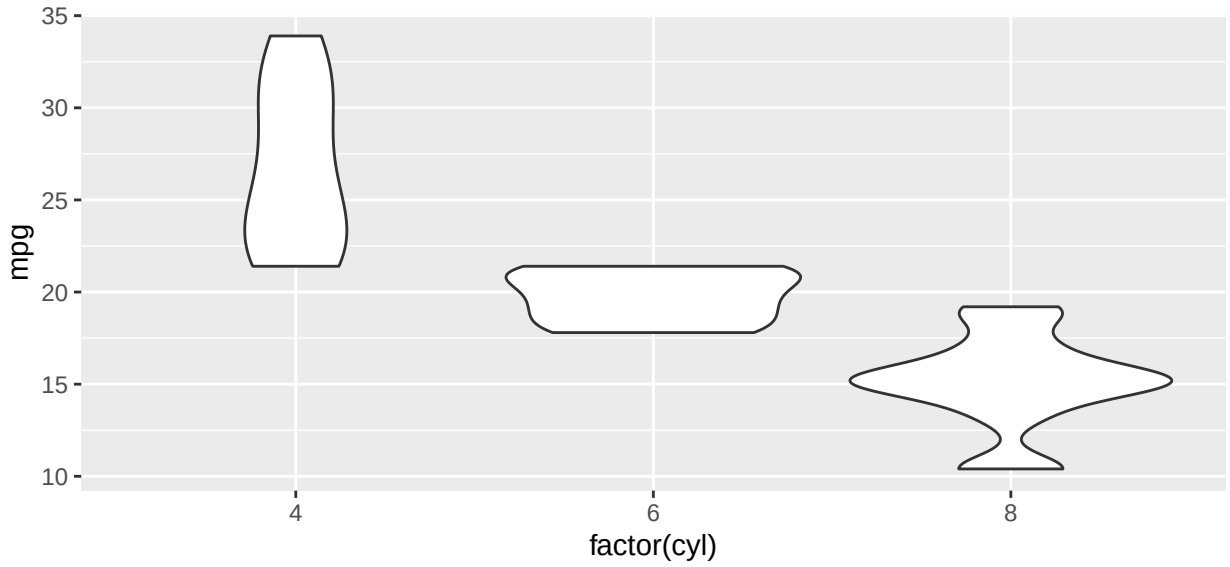


Gott dæmi hvenær `geom_violin` geymir meiri upplýsingar en `boxplot`

```
p <- ggplot(mtcars, aes(factor(cyl), mpg))
p + geom_boxplot()
```




```
p + geom_violin()
```



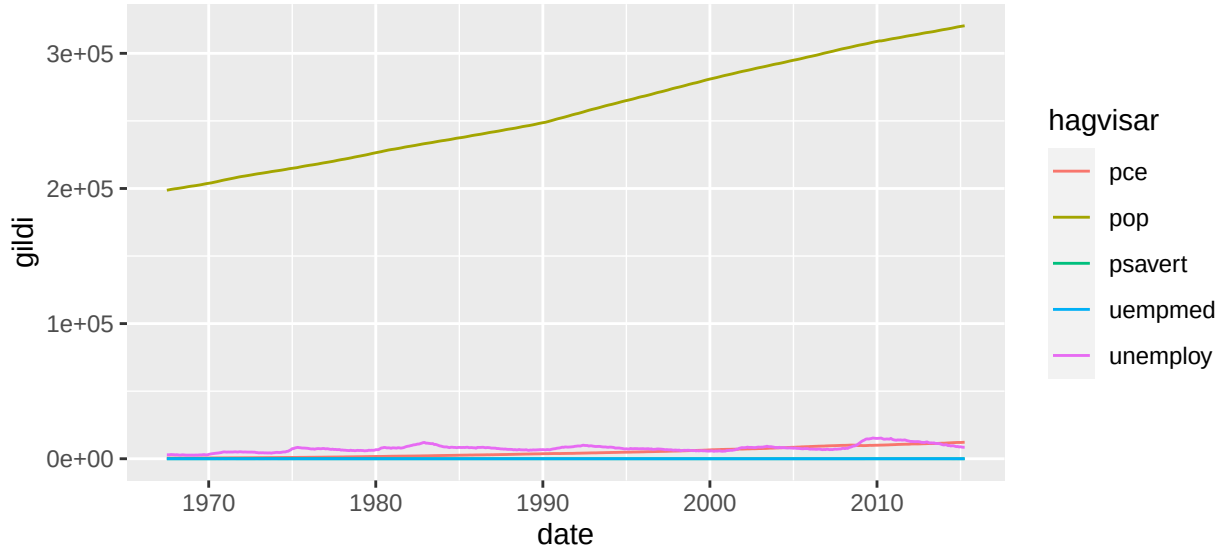
Aðeins um line graph. Viljum skoða nokkrar tímaraðir á sama tíma. Þær eru þó allar á mismunandi skala. Þurfum að nota **dplyr** og **ggplot**

```
econ <- economics %>%
  pivot_longer(cols = pce:unemploy, # eða 2:6
               names_to = "hagvisar",
               values_to = "gildi")
```

Virkar ekki að setja þetta allt á eitt graf út af mismunandi gildum

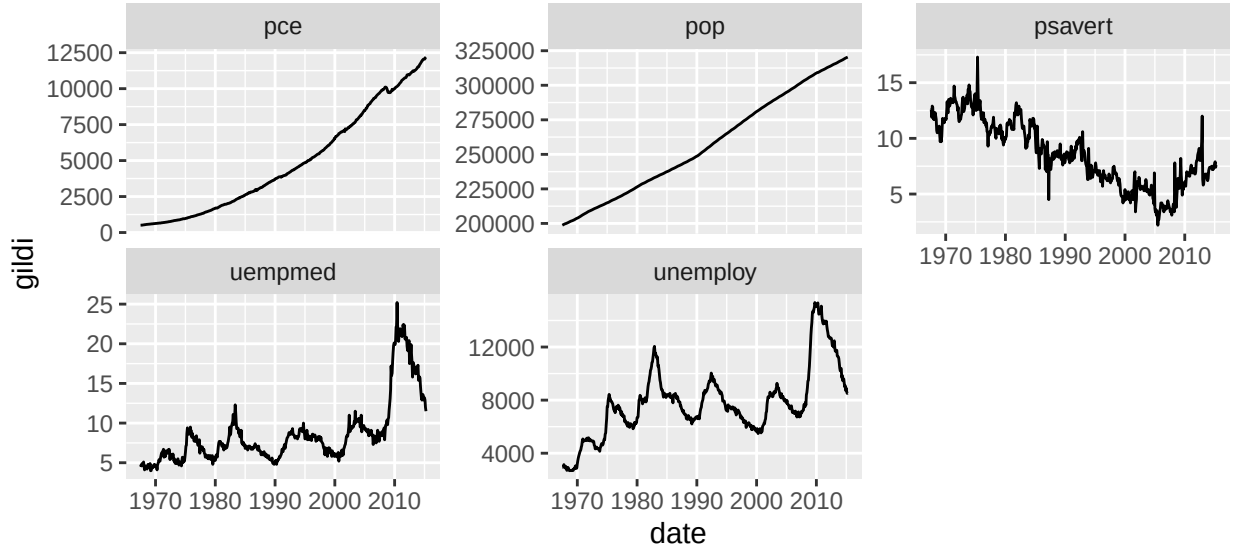
```
ggplot(econ,
       aes(x = date,
           y = gildi,
```

```
          col = hagvisar)) +
  geom_line()
```



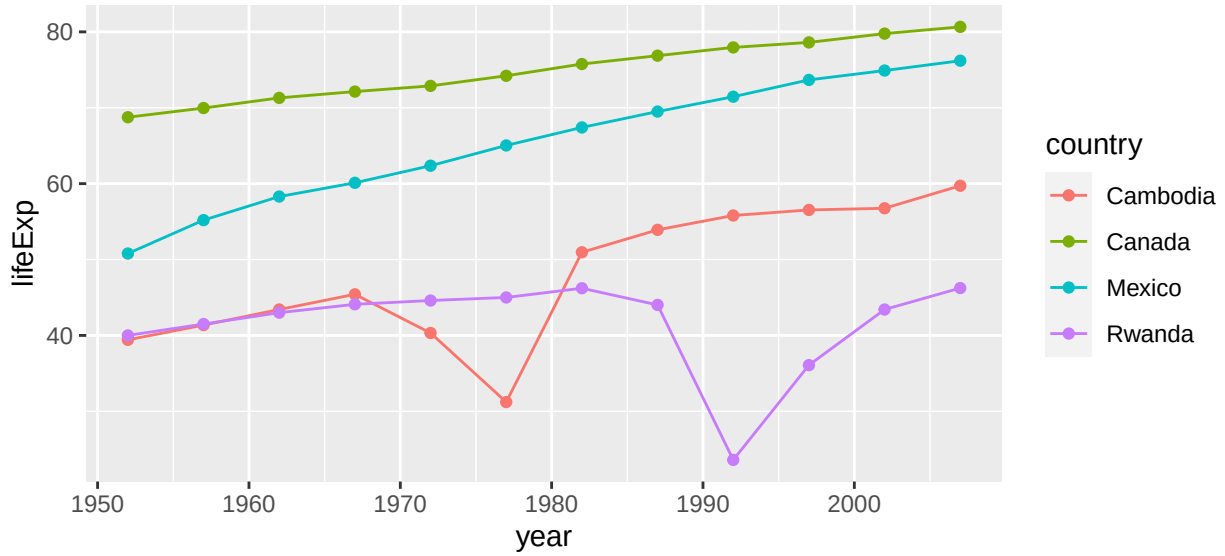
Notum facet-ing

```
ggplot(econ,
       aes(x = date,
           y = gildi)) +
  geom_line() +
  facet_wrap(~ hagvisar, scales = "free_y")
```



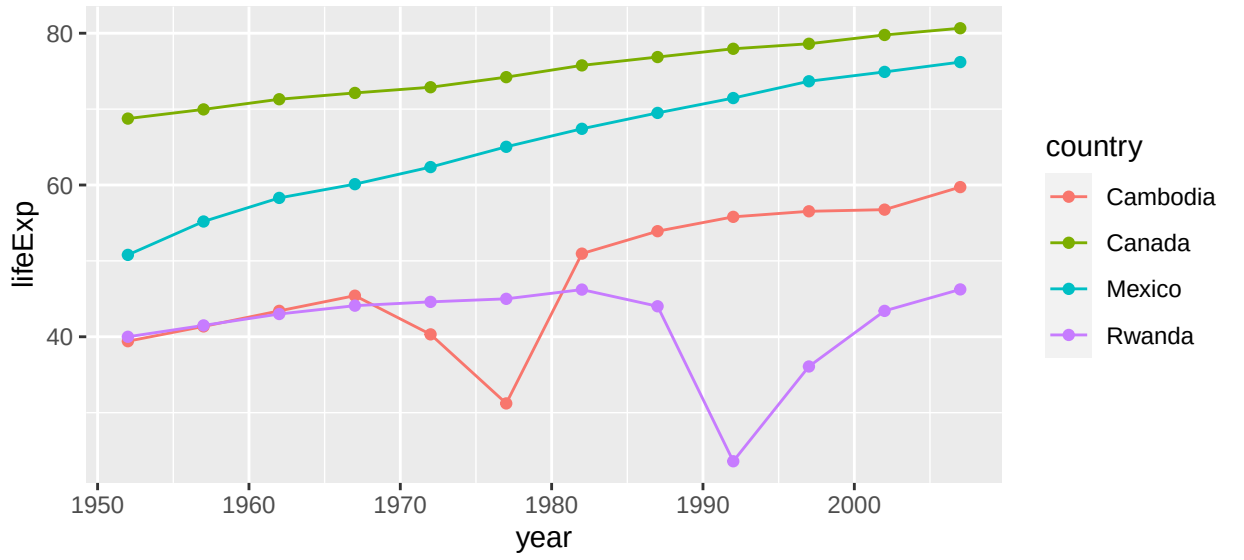
Stundum viljum við hafa gögnin á einu grafi þ.e. ekki nota facet. Að því gefnu að þetta sé í hægt. Til að fínisera myndina er eitt sem alla jafna mörgum finnst mikilvægt en það er að raða legend-inu eftir röðun gagnanna m.v. nýjasta gagnapunkinn.

```
gapminder %>%
  filter(country %in% c("Canada", "Rwanda", "Cambodia", "Mexico")) %>%
  ggplot(aes(x = year,
             y = lifeExp,
             col = country)) +
  geom_line() +
  geom_point()
```



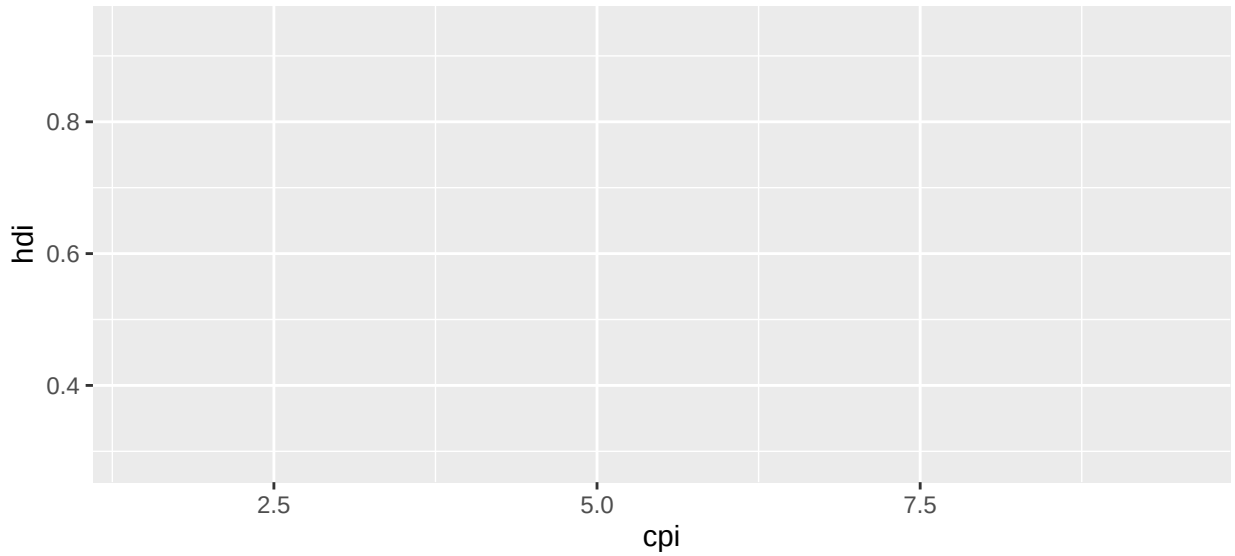
Röðunin er eftir stafrófsröð (by default). Viljum Canada -> Mexico -> Cambodia -> Rwanda

```
gapminder %>%
  filter(country %in% c("Canada", "Rwanda", "Cambodia", "Mexico")) %>%
  ggplot(aes(x = year,
             y = lifeExp,
             col = country)) +
  geom_line() +
  geom_point()
```

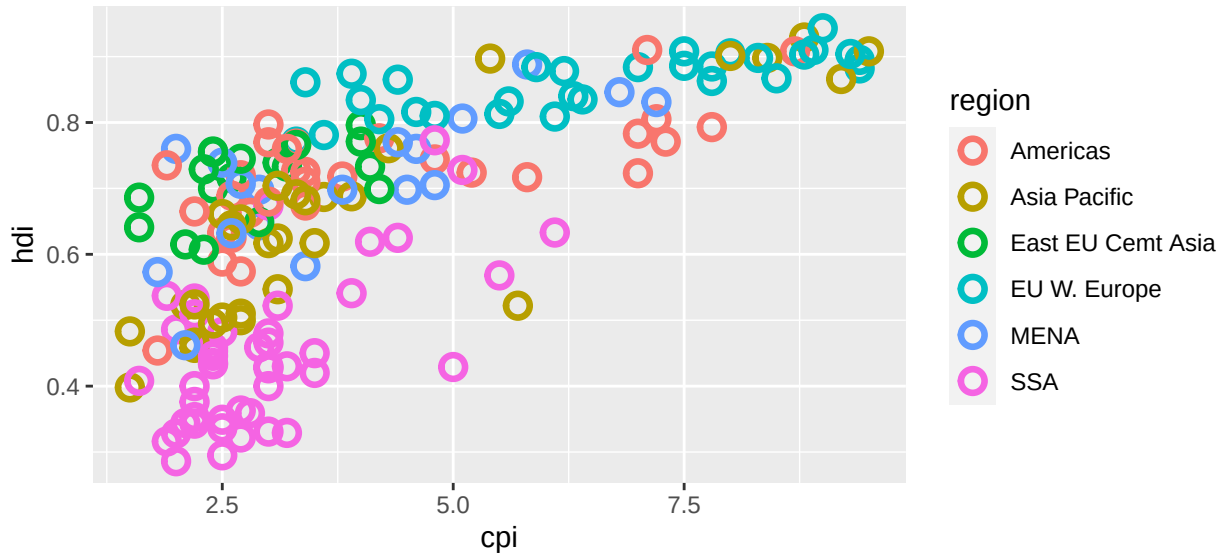


5.4.1 Verkefni - HDI CPI

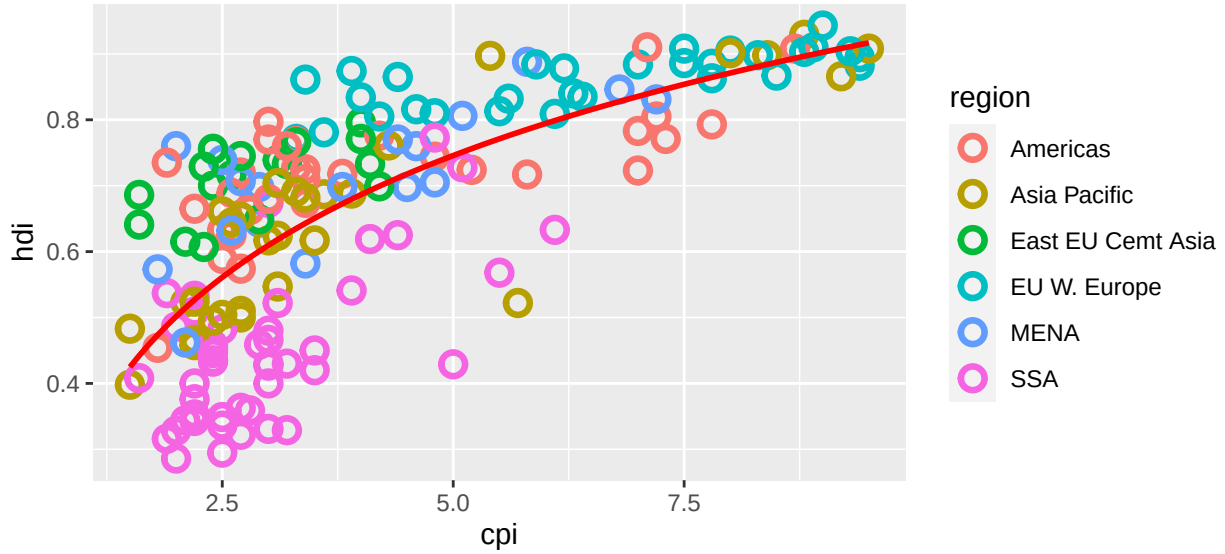
```
hdi <- readxl::read_excel("./data/cpi_hdi_final.xlsx") %>% janitor::clean_names()
hdi$hdi <- as.numeric(hdi$hdi)
hdi$cpi <- as.numeric(hdi$cpi)


(p1 <- ggplot(hdi,
       aes(x = cpi, y = hdi)))
```




```
(p2 <- p1 + geom_point(aes(color = region),
                       size = 3,
                       stroke = 1.75,
                       shape = 1))
```

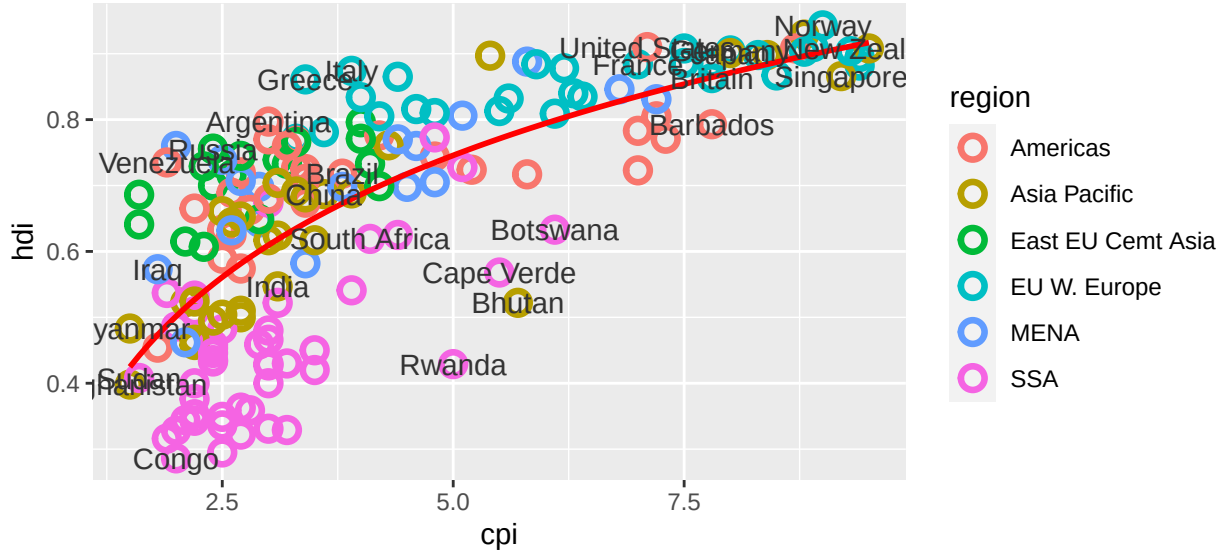



```
(p3 <- p2 +
    stat_smooth(method = "lm",
                formula = y ~ log(x), se = FALSE,
                col = "red"))
```

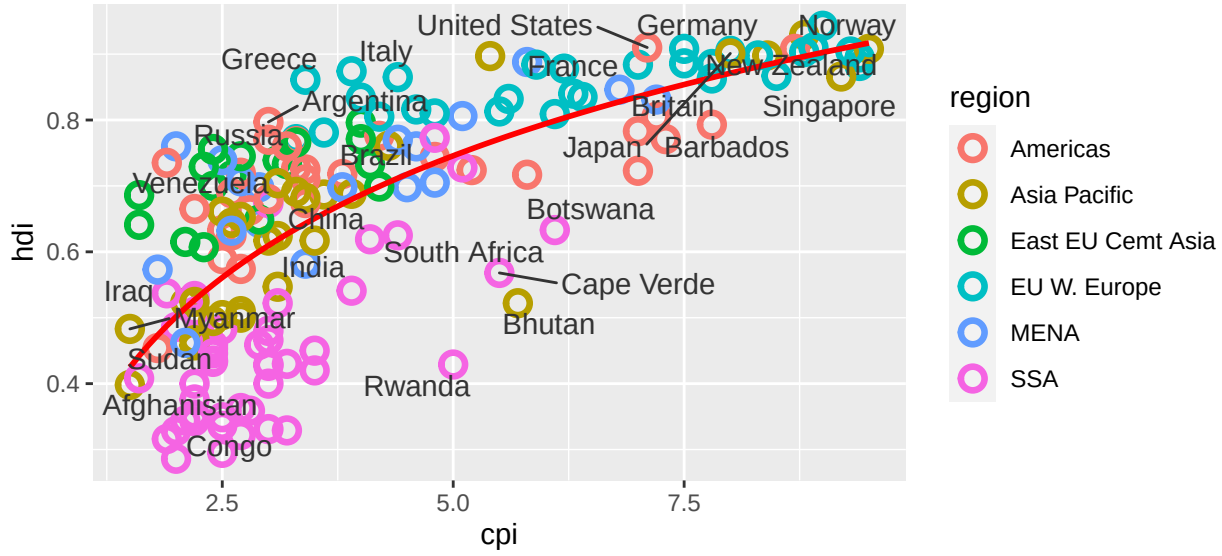



```
pointsToLabel <- c("Russia", "Venezuela", "Iraq", "Myanmar", "Sudan",
                   "Afghanistan", "Congo", "Greece", "Argentina", "Brazil",
                   "India", "Italy", "China", "South Africa", "Spane",
                   "Botswana", "Cape Verde", "Bhutan", "Rwanda", "France",
                   "United States", "Germany", "Britain", "Barbados", "Norway", "Japan",
                   "New Zealand", "Singapore")

 # texti
(p4 <- p3 +
    geom_text(aes(label = country),
```

```
            color = "gray20",
            data = filter(hdi, country %in% pointsToLabel)))
```

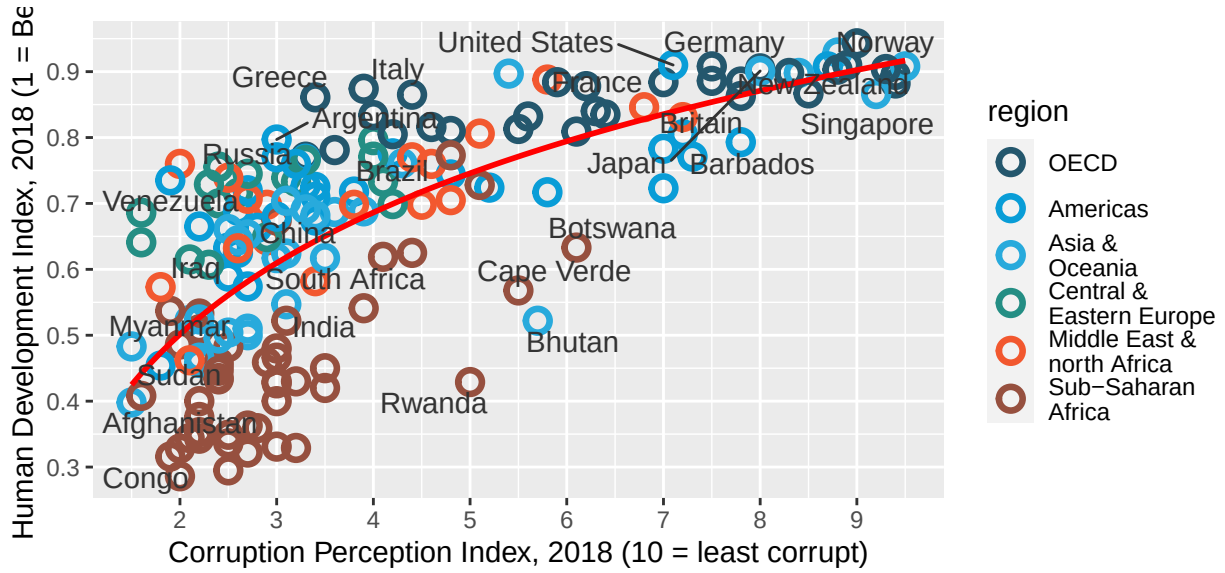



```
# ggrepl
library(ggrepel)
(p4 <- p3 +
    geom_text_repel(aes(label = country),
                    color = "gray20",
                    data = filter(hdi, country %in% pointsToLabel),
                    force = 10))
```

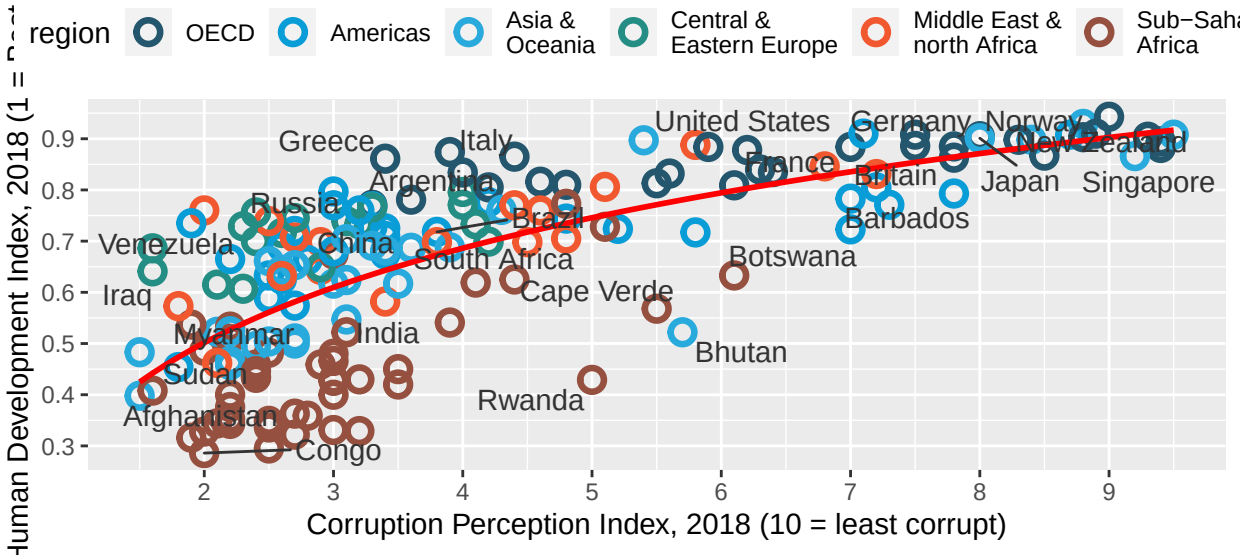


5.4.2 Breyti labels og order

```
hdi$region <- factor(hdi$region,
                     levels = c("EU W. Europe",
                                "Americas",
                                "Asia Pacific",
                                "East EU Cemt Asia",
                                "MENA",
                                "SSA"),
                     labels = c("OECD",
                                "Americas",
                                "Asia &\nOceania",
                                "Central &\nEastern Europe",
                                "Middle East &\nnorth Africa",
                                "Sub-Saharan\nAfrica"))

p4$data <- hdi


(p5 <- p4 +
    scale_x_continuous(name = "Corruption Perception Index, 2018 (10 = least corrupt)",
                       breaks = 1:10) +
    scale_y_continuous(name = "Human Development Index, 2018 (1 = Best",
                       breaks = seq(0.2, 1, by = 0.1)) +
    scale_color_manual(values = c("#24576D",
                                  "#099DD7",
                                  "#28AADC",
                                  "#248E84",
                                  "#F2582F",
                                  "#96503F")))
```




```
(p6 <- p5 +
    theme(legend.position = "top") +
    guides(color = guide_legend(nrow = 1)))
```



6 Helstu villur

gleyma c fyrir framan svigan þegar þú býrð til vector gleyma að loka pdf skjali áður en þú knit-ar nýtt gleyma að ungroup()-a

7 Hvar á að leita að hjálp

stackoverflow community.rstudio.com stats.stackexchange fyrir tölfræðistpurningar

8 Advanced og auka ef tími

Gögn af netinu Gögn úr gagnagrunni Gögn úr PDF OECD og Eurostat

Flexdashboard Inngangur að Shiny?