Introduksjon til R og RStudio

ISF 12.10.2023

Lise Rødland

Institutt for statsvitenskap, UiO

Dagens gjennomgang

- Kort intro til R og RStudio
    - Prosjekt/mappesystem
    - Hvordan kjøre kode
    - Pakker
- Funksjoner
    - Laste inn data
    - Bli kjent med data
    - Bearbeide data
    - Visualisering
    - Undersøke sammenhenger

Hvordan organisere arbeidet?

- Vi jobber i RStudio
- Fortell R hvilken mappe du vil jobbe i
    - `setwd("filbane")`
    - opprett et prosjekt
- Samme problem kan løses på mange måter, jeg bruker tidyverse:
    - R for Data Science: r4ds.hadley.nz
    - tidyverse.org

Litt kode helt i starten

- Lagre objekt: <- (og =)
- Legg til kommentar: #
- Vi kjører kode med:
  - Windows og Linux: `ctrl` + `enter`
  - Mac: `cmd` + `enter`
- R er sensitivt for store/små bokstaver

```
# Dette er en kommentar:
nyttobjekt <- 2  # Dette er også en kommentar
```

Installere og laste inn pakker

```r
# Installere pakken (første gang du bruker den):
install.packages("pakkenavn")

# Laste inn pakken (hver gang du starter RStudio/R):
library(pakkenavn)

# Bruke en funksjon i en pakke uten å laste inn hele pakken:
pakkenavn::funksjon()
```

Laste inn og lagre data

- Funksjon avhenger av format på data
- Noen dataformat krever ekstra pakker: `haven`, `readr`, `readxl`
- Noen datasett har egne pakker som lar deg laste inn data direkte: `essurvey`, `eurostat`, `manifestoR`, `stortingscrape`

```
# For mange filtyper ser det slik ut:
objekt <- read_filtype("filnavn.csv")
```

Laste inn data

```r
# Rdata:
load("filnavn.Rdata")

# csv:
library(readr)
objekt <- read_csv("filnavn.csv")  # Separert med ,
objekt <- read_csv2("filnavn.csv") # Separert med ;

# Excel:
library(readxl)
objekt <- read_excel("filnavn.xlsx")

# SPSS:
library(haven)
objekt <- read_sav("filnavn.sav")

# Stata:
library(haven)
objekt <- read_dta("filnavn.dta")

# SAS:
library(haven)
objekt <- read_sas("filnavn.sas7bdat")
```

shorturl.at/DLOQY

Lagre data

```
# Rdata:
write(objekt, file = "filnavn.Rdata")

# csv:
library(readr)
write_csv(objekt, file = "filnavn.csv")  # Separert med ,
write_csv2(objekt, file = "filnavn.csv") # Separert med ;

# Excel:
library(xlsx)
write_xlsx(objekt, file = "filnavn.xlsx")

# SPSS:
library(haven)
write_sav(objekt, path = "filnavn.sav")

# Stata:
library(haven)
write_dta(objekt, path = "filnavn.dta")

# SAS:
library(haven)
write_sas(objekt, path = "filnavn.sas7bdat")
```

Bli kjent med data

```
library(haven)
valg <- read_dta("valgdata.dta")
summary(valg)
```

```
##     kjonn                utd             stemte          alder
##  Length:1000        Min.   :0.000   Min.   :0.0000   Min.   :18.00
##  Class :character   1st Qu.:1.000   1st Qu.:1.0000   1st Qu.:44.00
##  Mode  :character   Median :2.000   Median :1.0000   Median :52.00
##                     Mean   :1.337   Mean   :0.8793   Mean   :52.02
##                     3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:60.00
##                     Max.   :2.000   Max.   :1.0000   Max.   :89.00
##                     NA's   :196     NA's   :22
##     demo_mis
##  Min.   :0.000
##  1st Qu.:1.000
##  Median :2.000
##  Mean   :1.745
##  3rd Qu.:2.000
##  Max.   :3.000
##  NA's   :30
```

Variabeloversikt

Valgdata er fiktive data med følgende variabler:

- `kjonn`: Mann, Kvinne
- `utd`: 0 = Grunnskole, 1 = Videregående, 2 = Universitet/høyskole
- `stemte ved valget`: 0 = Nei, 1 = Ja
- `alder`: antall år
- `demo_mis`: Hvor fornøyd med måten demokratiet virker på Norge? 0 = meget fornøyd og 3 = ikke fornøyd i det hele tatt.

Frekvenstabeller

```
table(valg$utd)  # $ henter ut variabelen utd i datasettet valg
```

```
##
##   0   1   2
## 136 261 407
```

```
table(valg$utd, useNA = "always") # Ber R rapportere missingverdier
```

```
##
##    0    1    2 <NA>
##  136  261  407  196
```

Frekvenstabeller (relativ)

```
prop.table(table(valg$utd))
```

```
##
##         0         1         2
## 0.1691542 0.3246269 0.5062189
```

```
prop.table(table(valg$utd, useNA = "always"))
```

```
##
##     0     1     2  <NA>
## 0.136 0.261 0.407 0.196
```

Krysstabeller

```
table(valg$stemte, valg$utd)
```

```
##
##       0   1   2
##   0  21  33  46
##   1 115 217 356
```

```
prop.table(table(valg$stemte, valg$utd), 1) # 1 angir % av rad
```

```
##
##              0         1         2
##   0 0.2100000 0.3300000 0.4600000
##   1 0.1671512 0.3154070 0.5174419
```

```
prop.table(table(valg$utd, valg$stemte), 2) # 2 angir % av kolonne
```

```
##
##              0         1
##   0 0.2100000 0.1671512
##   1 0.3300000 0.3154070
##   2 0.4600000 0.5174419
```

Deskriptiv statistikk

```
summary(valg$demo_mis)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   1.000   2.000   1.745   2.000   3.000      30
```

```
# Standardavvik
sd(valg$demo_mis)
```

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

```
# Husk at du må fortelle hvordan R skal håndtere missingverdier
sd(valg$demo_mis, na.rm = TRUE)
```

```
## [1] 0.702424
```

Andre nyttige funksjoner med liknende oppbygging som `sd()`: `mean()`, `median()`, `var()`, `quantile()`, `min()`, `max()`

Deskriptiv statitstikk for grupperte data

```
library(tidyverse)

valg %>%
  group_by(kjonn) %>%  # Grupperingsvariabel
  summarise(# nyttnavn = funksjon(variabel),
            snitt = mean(demo_mis, na.rm = TRUE),
            sd = sd(demo_mis, na.rm = TRUE))
```

```
## # A tibble: 2 x 3
##   kjonn  snitt    sd
##   <chr>  <dbl> <dbl>
## 1 Kvinne  1.69 0.712
## 2 Mann    1.79 0.691
```

Bearbeide data: Logiske operatorer

Logiske operatoren er nyttige både når man skal omkode variabler og når man skal hente ut observasjoner eller variabler. Disse kombineres ofte med `filter()` og `case_when`.

Operator	Betydning
==	er lik
<	mindre enn
>	st<U+00F8>rre enn
<=	mindre eller lik
>=	st<U+00F8>rre eller lik
!=	ulik
!x	ikke x
&	og
|	eller

Velge observasjoner og variabler

```
library(tidyverse)

# Bruker <- til å lage et nytt objekt som heter valg_menn
valg_menn <- valg %>%
  # filter() er for rader/observasjoner
  filter(kjonn == "Mann") %>%
  # select() er for kolonner/variabler
  select(demo_mis, stemte)

summary(valg_menn)
```

```
##     demo_mis         stemte
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.:1.0000
##  Median :2.000   Median :1.0000
##  Mean   :1.794   Mean   :0.8718
##  3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :3.000   Max.   :1.0000
##  NA's   :15      NA's   :13
```

Omkoding av variabler

```
# Skriver over opprinnelig objekt med valg <- valg
valg <- valg %>%
  mutate(# varnavn = funksjon() eller formel,
         utd_chr = case_when(utd == 0 ~ "Grunnskole",
                             utd == 1 ~ "VGS",
                             utd == 2 ~ "Universitet/hoyskole"),
         alder_sentr = alder - mean(alder, na.rm = TRUE),
         alder_chr = as.character(alder))
```

Objektklasser

Funksjoner som `mean()` og `sd()` krever at variabelen har klassen `numeric` eller `integer` (også angitt som `dbl`). Det som ser ut som tall når du kikker på datasettet kan likevel være lagret som `character` eller `factor`. Du kan sjekke ved å bruke funksjonen `class()`.

```
class(valg$kjonn)
```

```
## [1] "character"
```

```
class(valg$alder)
```

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

Objektklasser forts.

```
valg %>%
  select(alder, alder_chr) %>%  # Velger ut variablene alder og alder
  head(., 3)                    # Printer de første tre observasjoner
```

```
## # A tibble: 3 x 2
##   alder alder_chr
##   <dbl> <chr>
## 1    53 53
## 2    72 72
## 3    24 24
```

```
# Variabler kan "se" ut som tall, men ha klassen "character":
class(valg$alder_chr)
```

```
## [1] "character"
```

```
class(valg$alder)
```

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

Visualisering: `ggplot2`

1. Fortell ggplot() hvor du vil hente data fra.
2. Fortell ggplot() hvilken sammenheng du vil plotte.
3. Fortell ggplot() hvordan du vil fremstille sammenhengen.
4. Legg til geoms_ etter behov en etter en.
5. Bruk funksjoner til å justere skala, etiketter, tittel o.l..

Legg til nye *lag* ved hjelp av +.

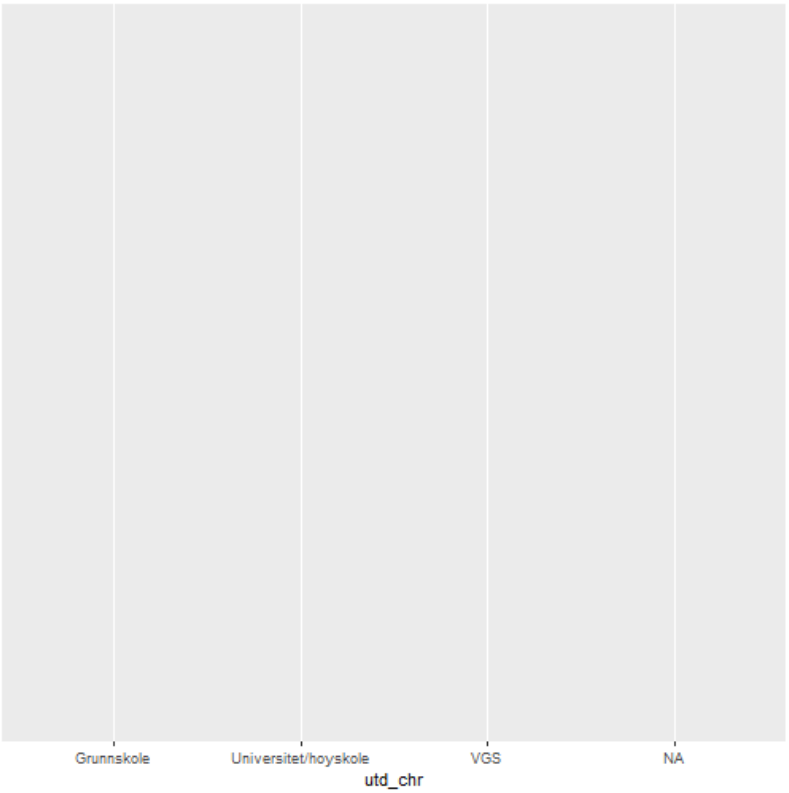
Se Healy, K. J. 2019. *Data Visualization: A Practical Introduction*. Princeton University Press. Fritt tilgjengelig på socviz.co.

Steg 1: Angi datakilde

```
library(tidyverse)
ggplot(data = valg)                          # Angir datakilde
```

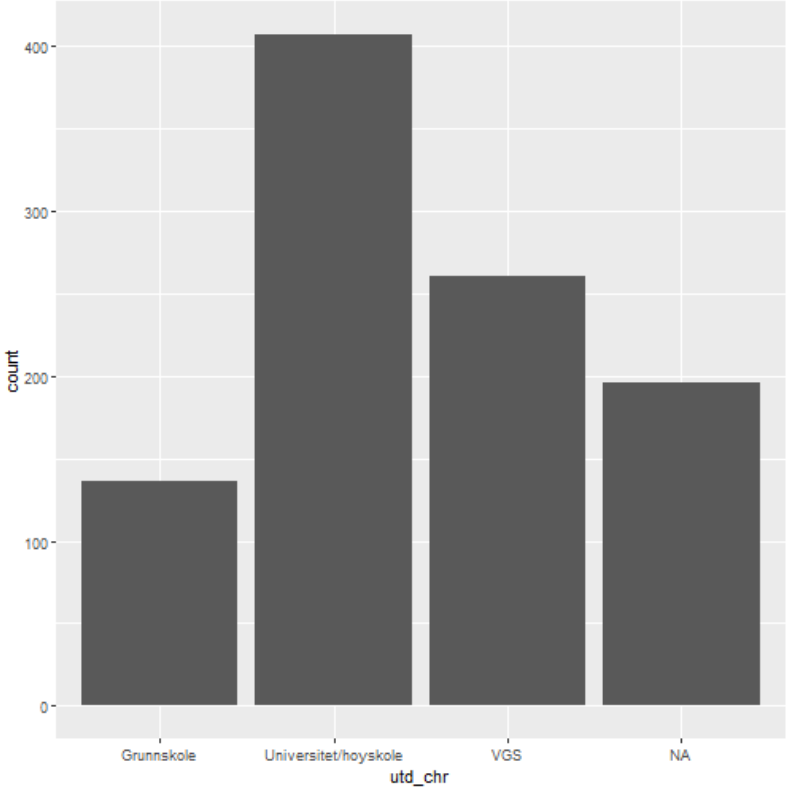
Steg 2: Angi sammenheng

```
library(tidyverse)
ggplot(data = valg,                          # Angir datakilde
       mapping = aes(x = utd_chr))           # Angir variabel
```



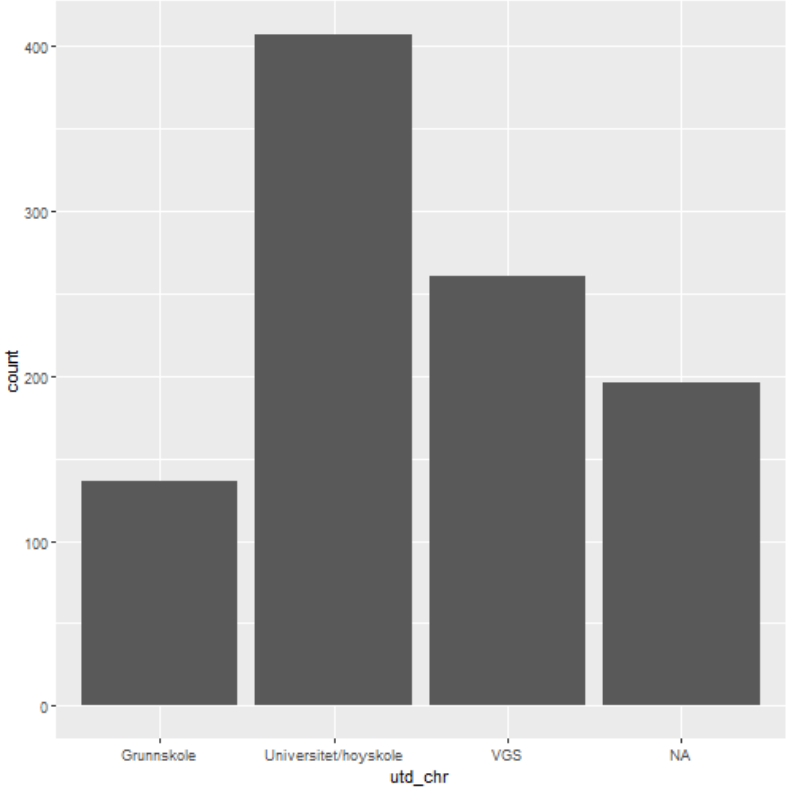
Steg 3: Angi type visualisering

```
library(tidyverse)
ggplot(data = valg,                          # Angir datakilde
       mapping = aes(x = utd_chr)) +           # Angir variabel
  geom_bar()                                 # Angir type visualisering
```



Steg 4: Legg til flere geoms

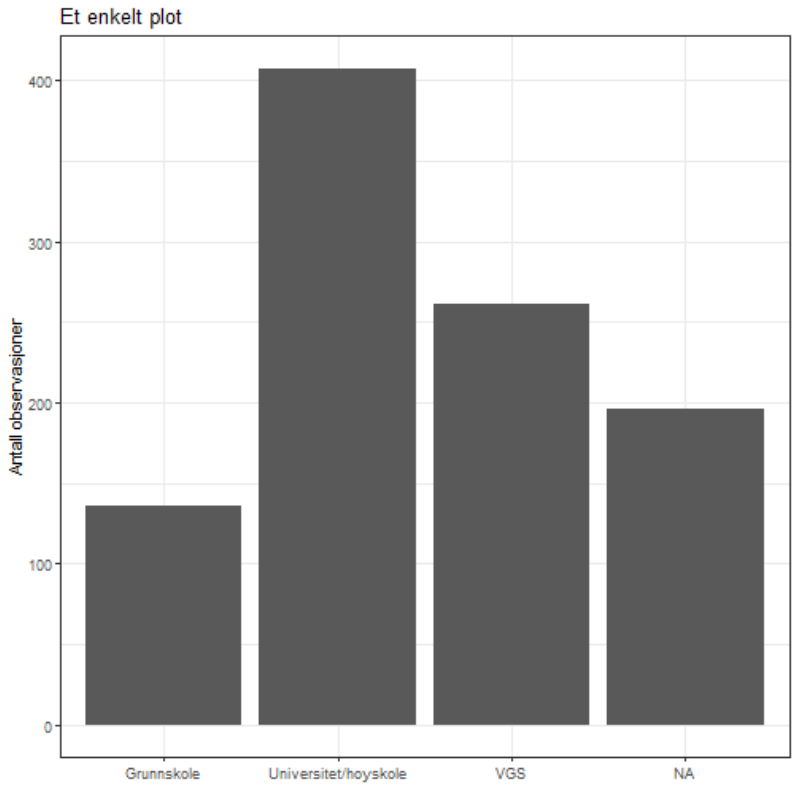
```
library(tidyverse)
ggplot(data = valg,                          # Angir datakilde
       mapping = aes(x = utd_chr)) +           # Angir variabel
  geom_bar()                                 # Angir type visualisering
```



Steg 5: Juster skala, etiketter etc.

```
library(tidyverse)
ggplot(data = valg,                          # Angir datakilde
       mapping = aes(x = utd_chr)) +           # Angir variabel
  geom_bar() +                               # Angir type visualisering
  labs(x = element_blank(),                  # Fjerner tittel på x-aksen
       y = "Antall observasjoner",           # Legger til tittel på y-akse
       title = "Et enkelt plot") +           # Legger til tittel
  theme_bw()                                 # Endrer design
```

Steg 5: Juster skala, etiketter etc.



Korrelasjonstabell

```
library(corrr)
correlate(valg)
```

```
## Non-numeric variables removed from input: kjonn, utd_chr, and alder_chr
## Correlation computed with
## * Method: 'pearson'
## * Missing treated using: 'pairwise.complete.obs'

## # A tibble: 5 x 6
##   term            utd  stemte   alder demo_mis alder_sentr
##   <chr>         <dbl>   <dbl>   <dbl>    <dbl>       <dbl>
## 1 utd          NA       0.0443  0.0465  -0.198       0.0465
## 2 stemte        0.0443 NA       0.156   -0.120       0.156
## 3 alder         0.0465  0.156  NA       -0.0764      1
## 4 demo_mis     -0.198  -0.120  -0.0764  NA          -0.0764
## 5 alder_sentr   0.0465  0.156   1       -0.0764     NA
```

Er korrelasjonen statistisk signifikant?

```
cor.test(valg$demo_mis, valg$stemte, use = "pairwise")
```

```
##
##     Pearson's product-moment correlation
##
## data:  valg$demo_mis and valg$stemte
## t = -3.7091, df = 946, p-value = 0.0002201
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.18200984 -0.05648515
## sample estimates:
##        cor
## -0.1197259
```

Regresjonsanalyse

```
# Syntaks OLS
lm(avhengig_variabel ~ uavhengig_variabel_1 + uavhengig_variabel_2,
   data = mitt_datasett)

# Syntaks logistisk regresjon
glm(avhengig_variabel ~ uavhengig_variabel_1 + uavhengig_variabel_2,
   data = mitt_datasett,
   family = "binomial")
```

Noen typer regresjon krever egne pakker som f.eks. flernivå (`lme4`), multinomisk (`nnet`). Pakken `tidymodels` tilbyr flere typer modeller, men har en litt annen oppbygging.

Eksempel på regresjonsanalyse (OLS)

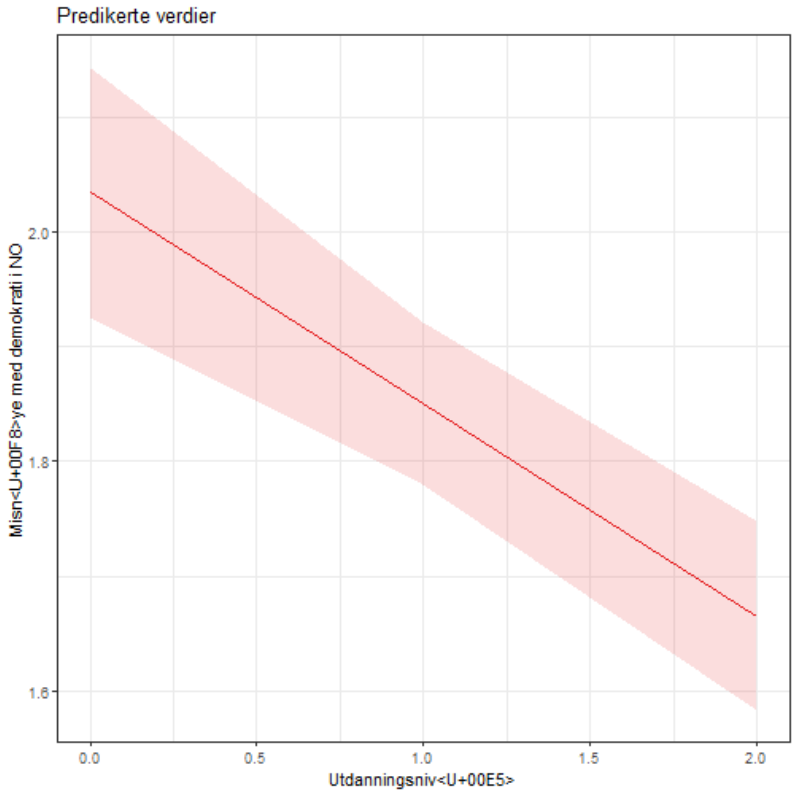
```
reg1 <- lm(demo_mis ~ utd,
           data = valg)
reg2 <- lm(demo_mis ~ utd + kjonn,
           data = valg)

summary(reg2)
```

```
##
## Call:
## lm(formula = demo_mis ~ utd + kjonn, data = valg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9425 -0.6652  0.1504  0.4264  1.4264
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.94248    0.05884  33.011  < 2e-16 ***
## utd         -0.18445    0.03345  -5.514 4.79e-08 ***
## kjonnMann    0.09160    0.05013   1.827    0.068 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
stargazer::stargazer(reg1, reg2, type = "text")
```

```
##
## ===============================================================
##                                 Dependent variable:
##                     ------------------------------------------
##                                       demo_mis
##                            (1)                      (2)
## ---------------------------------------------------------------
## utd                      -0.188***                -0.184***
##                           (0.033)                  (0.033)
##
## kjonnMann                                          0.092*
##                                                    (0.050)
##
## Constant                 1.995***                 1.942***
##                           (0.051)                  (0.059)
##
## ---------------------------------------------------------------
## Observations                779                      779
## R2                         0.039                    0.043
## Adjusted R2                0.038                    0.041
## Residual Std. Error   0.699 (df = 777)         0.698 (df = 776)
## F Statistic         31.735*** (df = 1; 777) 17.585*** (df = 2; 776)
## ===============================================================
## Note:                                *p<0.1; **p<0.05; ***p<0.01
```

```r
sjPlot::plot_model(reg2,                          # modellobjekt
                   terms = "utd",                 # variabel du vil visualis
                   type = "pred") +               # type plot
  labs(title = "Predikerte verdier",              # tittel på visualisering
       x = "Utdanningsnivå",                      # tittel på x-aksen
       y = "Misnøye med demokrati i NO") +        # tittel på y-akse
  theme_bw()                                      # endrer design
```



Tips og triks på veien:

- Så lenge du lagrer endringer i nye objekter så kan du ikke gjøre noe feil.
- Forsøk deg frem med endringer og se hva som skjer.
- Les hjelpefiler.
- I tilfelle trøbbel:
  - Har du lastet inn de nødvendige pakkene?
  - Har variabelen "riktig" klasse?
  - Er alle parentesene lukket?
  - Er det noen store bokstaver som skulle vært små eller andre skrivfeil?
  - Les feilmeldingen i Console. Søk på nett eller spør ChatGPT.

Ressurser

- Læringsmateriell STV1020: https://github.com/martigso/STV1020/tree/main
- Øvingsoppgaver: https://shinyibv02.uio.no/connect/?fbclid=IwAR2tF5yHFLF1ymRYFaUuHUc6uxV1k6F24bZe_Cdki54bGfKXndHmgt0ne_
- R for Data Science: r4ds.hadley.nz
- ggplot2: https://ggplot2.tidyverse.org/
- Tidyverse stilguide: https://style.tidyverse.org/
- tidymodels: https://www.tidymodels.org/

Takk for meg!

Slides created via the R package **xaringan**.