Seminar 6: Multippel regresjon

Lise Rødland

April 12, 2021

I dag skal vi se på fem ting:

1. Laste inn data (repetisjon)
2. Omkoding av variabler (repetisjon)
3. Plotting (repetisjon)
4. Kjøre en regresjonsmodell med en uavhengig variabel (nytt)
5. Tolkning og fremstilling av regresjonsresultater (nytt)

Laste inn pakker

Det aller første vi gjør er å laste inn pakkene vi skal bruke i dag ved hjelp av `library(pakkenavn)`:

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------ tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts --------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(stargazer)
```

```
##
## Please cite as:

##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

Dersom du ikke har brukt disse pakkene før må du huske å kjøre `install.packages("pakkenavn")` først. Dersom du får en feilmelding av typen "Error in library(pakkenavn) : there is no package called 'pakkenavn'" så kan det indikere at pakken ikke er installert. Prøv å kjøre `install.packages("pakkenavn")` og `library(pakkenavn)` igjen.

Laste inn data

Det neste vi skal gjøre er å laste inn datasettet vi skal jobbe med i dag. Vi skal jobbe videre med datasettet fra Kellstedt og Whitten som vi så på forrige gang. Disse dataene er i `Rdata`-format og vi bruker derfor funksjonen `load()`. Husk at hvilken funksjon du bruker for å laste inn data avhenger av hvilket format dataene har. Dersom du er usikker på hvilken funksjon du skal bruke så sjekk dokumentet jeg har lastet opp i Canvas. Vi laster inn data:

```
# Bytt ut det som står i "" med din egen filbane:
load("../../data/ANES1996small.RData")
```

Bli kjent med data

Det er alltid lurt å bli litt kjent med datasettet før en begynner å kjøre analyser. Vi har sett på flere koder for dette, blant annet i seminar fem, men vi gjentar noen av dem her.

Vi finner navnene på variablene:

```
names(ANES1996small)
```

```
##  [1] "v960066"  "v960067"  "v960070"  "v960071"  "v960073"  "v960115"
##  [7] "v960119"  "v960272"  "v960281"  "v960284"  "v960292"  "v960293"
## [13] "v960365"  "v960385"  "v960420"  "v960568"  "v960571"  "v960605"
## [19] "v960606"  "v960610"  "v960701"  "v961039"  "v961300"  "religion"
```

Dette er ikke veldig intuitive variabelnavn så senere skal vi endre navn på de vi skal bruke i analysen vår.

Vi kan bruke `View()` for å undersøke datasettet nærmere:

```
View(ANES1996small)
```

Et alternativ til view for å bare se noen observasjoner er `head()`:

```
head(ANES1996small)
```

```
## # A tibble: 6 x 24
##   v960066 v960067 v960070 v960071 v960073 v960115 v960119 v960272 v960281
##     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1       1       1       3       3       2       1     574       0       0
## 2       1       1       2       2       1       1       0      60      30
## 3       1       1       2       2       2       1       0      70      85
## 4       1       1       2       2       3       1       0      30      40
## 5       2       1       4       2       3       1       0      40      60
## 6       1       1       3       3       4       1    7323      70      15
## # ... with 15 more variables: v960284 <dbl>, v960292 <dbl>, v960293 <dbl>,
## #   v960365 <dbl>, v960385 <dbl>, v960420 <dbl>, v960568 <dbl>, v960571 <dbl>,
## #   v960605 <dbl>, v960606 <dbl>, v960610 <dbl>, v960701 <dbl>, v961039 <dbl>,
## #   v961300 <dbl>, religion <dbl>
```

Ved å bruke for eksempel `View()` får vi mer informasjon om hva slags struktur datasettet vårt har. Dersom vi jobber i store datasett er det lurt å bruke funksjoner som `head()` og `tail()` isteden for å bruker `View()` eller «trykke» på datasettet i environment. `View` krever mye fra pc'en.

For å titte på enkelt variabler bruker vi syntaksen `datasett$variabel`. Det kan også være praktisk å se denne informasjonen i en tabell. Da kan vi bruke table(`datasett$variabel`). For eksempel så vet jeg at variabelen `v960066` er det man kaller en dikotom variabel for kjønn. At en variabel er dikotom betyr bare at den har to verdier. Denne variabelen tar verdien 1 dersom respondenten er en mann og 2 dersom respondenten er en kvinne. Vi kan undersøke den i en tabell:

```
table(ANES1996small$v960066, useNA = "always")
```

```
##
##    1    2 <NA>
##  768  946    0
```

Forberede data for analyse

Før vi begynner på regresjonsanalysen så skal vi endre navn på noen av variablene våre så de blir mer intuitive, lage et subset med de variablene vi vil bruke og omkode kjønnsvariabelen til det man kaller en dummyvariabel. En dummyvariabel tar verdiene 0 og 1.

Først endrer vi navnene på de variabelene vi vil bruke i datasettet vårt ved hjelp av `rename()`. `rename()` har syntaksen `rename(nyttnavn = gammeltnavn)`. I praksis blir det:

```
# renaming existing variables, creating a dummy variable for female, and the interactive term
ANES1996small2 = ANES1996small %>%
  rename(hillary_thermo = v960281,
         income = v960701,
         womenmvmt_thermo = v961039) %>%
  mutate(female = ifelse(v960066==1, 0, 1))
```

Vi lager et subset av det opprinnelige ESS datasettet, da tar vi kunmed de variablene vi vil ha til resten av analysen. Dette er ikke nødvendig i strengforstand, men vi gjør det for å gjøre datasettet litt mer oversiktlig. Vi bruker select() fra dplyr for å velge ut variablene vi vil ha med i datasettet.Husk også at det er lurt å lagredet nye datasettet i et nytt objekt, slik at vi ikke overskriver det originale datasettet.ess <-ESS8SE %>% select(gndr, agea, eduyrs, nwspol, stfdem, vote)Det er alltid lurt å sjekke hvor mye missing vi har i data. Da bruker vi table() og complete.cases(),FALSE viser til hvilke enheter som har NA på minst en variabel, TRUE viser til enhetene som har verdier på alle variablertable(complete.cases(ess))Vi bruker funksjonen na.omit() til å kaste ut enheter med missing verdier. Å kaste ut missing er en mulig strategi, men dette bør vi alltid tenke igjennom. Dette gjelder spesielt om dataharmye missing verdier.ess <-na.omit(ess) Vi tar en titt på avhengig variabel til regresjonsanalysen. Denne variabelen måler hvor fornøyd respondenten er med demokratiet målt på en skala fra 0-10.str(ess$stfdem$)summary(ess$stfdem)Videre tar vi en titt på noen av de uavhengige variablene.Variabelen gndr er kodet 1 = mann og 2 = kvinne.table(ess$gndr$)Vi vil kode om denne slik at mann = 0 og kvinne = 1.Dette er en noe mer intuitiv koding av denne variabelen.Det kan også $v_relurt å tenke over hvilke forventninger vi har til variabel
$-ess$vote)Vi vil ha en dikotom variabel som fanger hvorvidt personen stemte ved siste valg eller ikke. Vi vil har de som ikke var stemmeberettiget sammen med de som ikke stemte i forrige valg. Dvs, kodet som nei.ess <-ess %>%mutate(vote2 = if_else(vote >= 2, 0, 1))Sjekk ?if_else() for hjelpefilen til denne funksjonen Koden sier at respondentene som har verdien 2 eller høyere på vote skal ha verdien 0 i den nye variabelen. De som har verdier under 2 får verdien 1 på den nye variabelen.Sjekk omkodingen med table(), vi kan også sammenligne den omkodede variablene med den originalevariabelenved å bruke table(ess$vote$).På denne måten kan vi se at omkodingen er riktig.table(ess$vote2)Vi endrer navnene på variablene, for å gjøre det hele mer oversiktlig. Dette er helt valgfritt, men dere vil ofte oppleve at variablene i diverse datasett har navn som er lite intuitive.Vi bruker renamefunksjonensom er en enkelt kode for å endre navn. Sjekk ?rename for mer informasjon.ess <-ess %>%rename(alder = agea, utdanning = eduyrs,

nyheter = nwspol, demokrati = stfdem)Vi kan sjekke at omkodningen og navnebyttene er korrekte ved å bruke headfunksjonenhead(ess)Kjøre OLS regresjonen, vi brukerfunksjonenlmfor å kjøre en OLS i RFor å legge til uavhengige variabler bruker vi +, avhengig variabler kommer først og spesifiseres med~Vi forteller R hvilket datasett vi vil bruke med data =Vi lagrer også modellen i et objekt, slik at vi kan bruke verdiene seneremod1 <-lm(demokrati ~ vote2 + nyheter + utdanning + alder + kjønn, data = ess)Vi bruker summary for å se resultatene av modellen vår.summary(mod1) Her har jeg lagt til ett samspillsledd mellom stemmegivning og konsum av politiske nyheter. En måte å legge til samspill på er ved å bruke * mellom samspillsvariablene. Det er også mulig å opprette en ny variabel med mutate hvor man spesifiserer samspillsleddet.mod2 <-lm(demokrati ~ vote2 * nyheter + utdanning +alder + kjønn, data = ess)Legg merke til at vi for koeffisienter for både samspillsleddet og koeffisienter for variablene i samspillet individuelt. Dette er viktiginformasjon når vi skal tolke resultatene fra en modell med samspill. Her plotter vi regresjonslinjen for utdannings variabelen vår.Det er viktig å velge plot etter hvilke variabler vi er interesserte i å visualisere. Her har vi to kontinuerlige variablersom er rimelig enkelt å plotte. Dersom vi vil plott dummyvariabler eller kategoriske variabler må vi finne andre plots. En jukselapp til ggplot finnes her: https://rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdfggplot(ess, aes(x = utdanning, y = demokrati)) + stat_smooth(method = "lm", col = "red")Videre skal vi se på noen grafiske verktøy for å vurdere om enkelteforutsetninger for OLS er oppfylt Da må vi først lagre restleddene og verdiene fra modellen vår i datasettetDette kan også gjøres med mutate, men her velger vi å bruke en enkelt kode fra base.Vi bruker residfunksjonen for å trekke ut restleddene fra modell 1. fitted.values ligger allerede innbakt i modellobjektet, derfor kan vi enkelt trekke disse ut med $ og lagre disse i datasettet vårt. ess$mod1Resid < -resid(mod1)ess$mod1Fitted <-mod1$fitted.valuesVi vil så vurdere restleddenes fordeling med et histogram, er restleddene våre normalfordelte?Her bruker vi restleddenefra mod 1 som vi la inn i datasettet vårt med koden ovenfor.ggplot(ess,aes(x = mod1Resid)) + geom_histogram()Nå skal vi forsøke å lage en figur som plotter restleddene mot modellens verdierDette gjør vi for å vurdereeventuellhetroskedastisitetVi brukerverdiene som i lagret i datasettet på x aksen og modellens restledd på y aksen. Vi legger til de ulike enhetene med geom_point, så trekker vi en linje gjennompunktene med geom_smooth.ggplot(ess,aes(x = mod1Fitted, y = mod1Resid)) +geom_point() + geom_smooth() Slike plott kanvære noe vanskelig å tolke,hvor enkeleller vanskelig tolkningen blir avhenger ofte av hvordan variablene våre er kodet. Vi kan også bruke plot() for å få ulike figurer for diagnosistikk, vi vil få 4 ulike plot med denne funksjonen.Disse plottene er ikke like fine som de vi lager i ggplot, men de kan hjelpe oss med å få en rask oversikt over ulike diagnosistikk plot(mod1) Vi lager en fin regresjonstabell med begge modellene våre side ved side.Stargazer pakken kan bruken til svært mye i R. Husk å sjekke hjelepfilen ?stargazer. Vi bruker type = til å spesifisere hvilket format vi vil ha tabellen i. covariate.labels bruker til å legge til nye navn til de uavhengige variablenei modellen. Det er viktig å legge inn navnene i samme rekkefølge som i regresjonsmodellen, vist ikke risikerer vi å gi feil navn til variablene. Dep.var.labels bruker til å gi navn til avhengig variabel. stargazer(mod1, mod2, type = "text", title = c("Modeller"), covariate.labels = c("Evne til politisk deltakelse", "Politiske nyheter","Utdanning","Alder","Kjønn","Samspill"),dep.var.labels = c("Tilfredshet med demokratiet"))Vi ønsker ofte å bruke tabellene våre i eksempelviswordVi kan lagre tabellen som en html-fil og deretter bruke den i eksempelvis wordDette gjør vi ved å sette type = "html",Deretter legger vi til argumentet out = "navnpåtabell.htm"stargazer(mod1, mod2, type = "html", title = c("Modeller"), covariate.labels = c("Evne til politisk deltakelse", "Politiske nyheter","Utdanning","Alder", "Kjønn","Samspill"),dep.var.labels = c("Tilfredshet med demokratiet"),out = "regtabell.htm")Tabellen vi da lagers i working directory. Vi kan da bruke «åpne i»i mappen hvor filen ligger –og velge åpne i word.