Seminar 5

Louisa Boulaziz

April 7, 2021

I dag skal vi jobbe med det samme datasettet som sist gang ettersom vi kjenner variablene. Det kan være lurt når vi begynner med regresjonsanalyse. Vi begynner med å laste inn pakker og data. Vi skal bruke en ny pakke i dag. Denne må dere ta install.packages("modlr") først, også library som under.

```
# Pakker
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.0 --
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.6     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

library(modelr)

# Laser inn data
data <- read.csv("https://raw.githubusercontent.com/louisabo/STV4020A/master/SEMINAR3/interr
```

Jeg bruker read.csv fordi dataene er csv-fil og de er lastet ned direkte fra internett. Når jeg gjør dette trenger jeg i prinsippet ikke å sette et "working

directory."
Vi skal jobbe videre med NA. Se på noen andre koder vi kan bruke for å fiske ut NA. NA som dere husker er missing-verdier og disse må vi identifisere. Koden complete.cases viser hvor mange fullstendige eller ufullstendige rader det er i datasettet. Det vil si at denne koden identifiserer alle rader som har NA på en eller flere variable. Du kan bruke table for å printe dem

```
table(complete.cases(data)) # Viser ufullstendige rader

##
## FALSE  TRUE
##   183  2562

table(is.na(data$internettbruk)) # Viser antall NA på variabel

##
## FALSE  TRUE
##  2740     5

# Vi kan legge NA i et eget datasett:

missing_2 <- data %>%
  filter_all(any_vars(is.na(.)))
```

Vi kan også lage en ny variabel i datasettet som flagger NA-rader – altså rader eller observasjoner som har NA på en eller flere variable. Dette kan vi gjøre på alle variable, eventuelt bare flagge rader som har NA på en spesifikk variabel. Vi bruker mutate fra tidyverse-pakken til å opprette to nye variabler.

```
data <- data %>%
  mutate(complete = complete.cases(.),   # ny var 1
         internett_na = is.na(internettbruk)) # ny var 2


table(data$internett_na)

##
## FALSE  TRUE
##  2740     5

table(data$complete)

##
## FALSE  TRUE
##   183  2562
```

Her fjerner alle NA også jobber vi videre. Merk at jeg også fjerner de nye variablene jeg nettopp opprettet fordi ønsker jeg ikke å ha med meg videre. Du kan bruke select funksjonen etter drop$_n a()ogderetterbaretallforåsitilRhvilkevariableduvelge.Jegvilvelgedeførste$ 5.

```
data <- data %>%
  drop_na() %>%
  select(1:5)
```

Før vi går videre med regresjonsanalyse må vi repetere litt oppretting av nye variable. Dette er veldig viktig å kunne og dere må øve på dette før prøven. Som alltid i R finnes det mye kode som gjør det samme. Jeg skal vise litt forskjellige omkodinger. Først omkoder vi variabelen kjønn. Nå viser den 1 for menn og 2 for kvinner. Jeg vil at den skal vise 0 for menn og 1 for kvinner.

```
data$kjonn_1 <- ifelse(data$kjonn == 1, 0, 1)

# Sjekker at det blir riktig
table(data$kjonn_1)

##
##    0    1
## 1211 1351
```

Vi skal foreta en litt vanskeligere omkoding hvor vi koder om utdanning til en variabel som måler utdanning i kategori istedet for kontiuerlig. Vi skal opprette fire kategorier:

- Alle som har antall år med utdanning 0-10 = lav
- Alle som har antall år med utdanning 11-20 = mellom
- Alle som har antall år med utdanning 21-30 = hoy
- Alle som har antall år med utdanning 31-37 = veldig hoy

Mutate er bra for store og flere omkodinger samtidig. Jeg vil bare legge til at denne koden er ikke noe jeg forventer dere må huske, men det er viktig at dere prøver å se om dere forstår logikken og syntaksen. Det er ekstremt viktig å prøve å forstå hva koden gjør, mer enn å huske den utenatt.

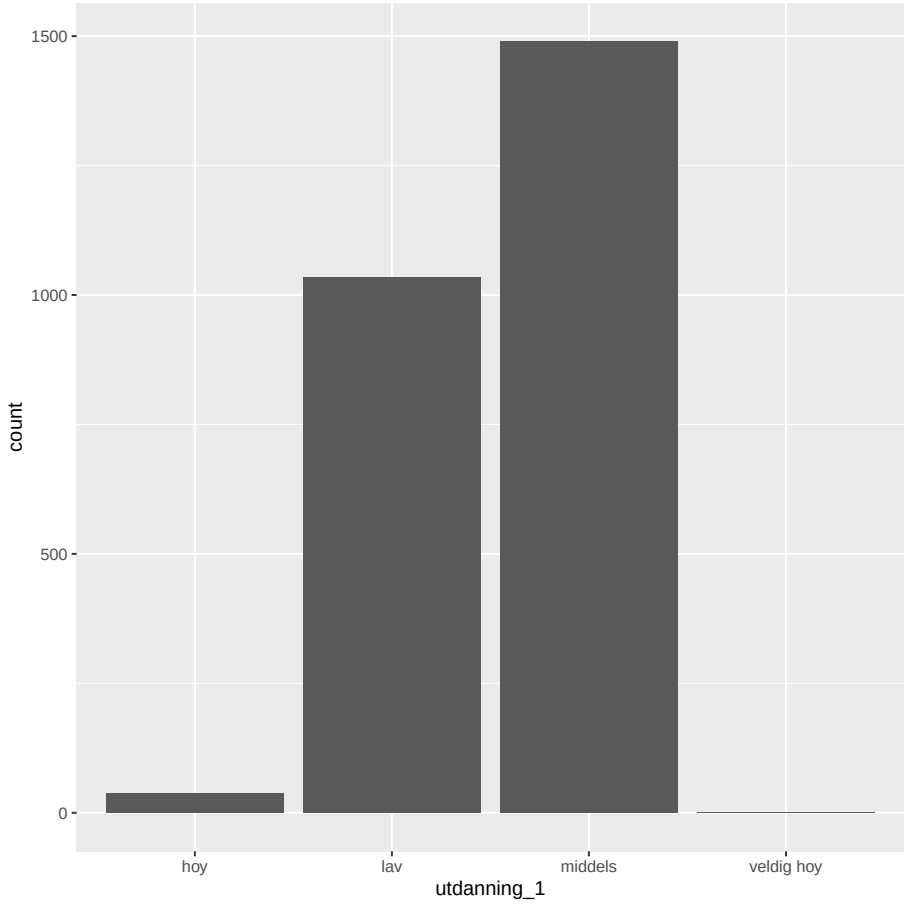
```
data <- data %>%
  mutate(utdanning_1 = case_when(utdanning >=0 &
                                   utdanning <= 10 ~ "lav",
```

```
                                  utdanning >=11 &
                                    utdanning <= 20
                                  ~"middels"
                                    , utdanning >= 21 &
                                    utdanning <= 30 ~ "hoy",
                                  utdanning >=30 & utdanning
                                  <= 37 ~ "veldig hoy"))

# Ser at det blir riktig:
table(data$utdanning_1)

##
##        hoy        lav    middels veldig hoy
##         37       1034       1490          1

table(data$utdanning_1, data$utdanning)

##
##                0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
##   hoy          0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   lav          2   3  11  33  12 227  31  25 490  67 133   0   0   0   0   0
##   middels      0   0   0   0   0   0   0   0   0   0   0 131 136 595 126  93
##   veldig hoy   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##
##               16  17  18  19  20  21  22  23  24  25  26  27  30  37
##   hoy          0   0   0   0   0  10   8  10   2   2   1   1   3   0
##   lav          0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   middels    110  57 135  55  52   0   0   0   0   0   0   0   0   0
##   veldig hoy   0   0   0   0   0   0   0   0   0   0   0   0   0   1
```

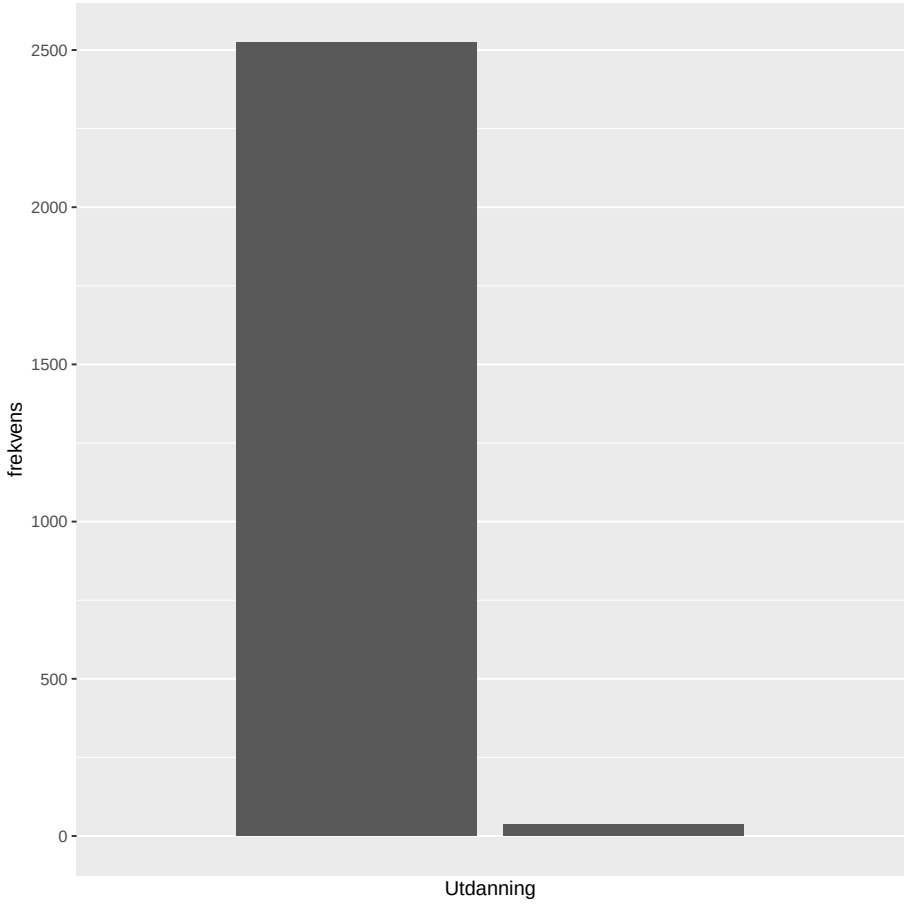
Som om denne koden ikke var nok, så skal vi omkode denne variabelen til å lage en dikotom utdanningsvariabel av den vi har laget med fire nivåer. Nå bruker jeg bare ifelse og ikke mutate for å vise. Disse kodene blir litt lange så det gjelder å holde tunga rett i munnen. Her koder jeg om alle som har lav og middels utdanning til 0 og alle som har høy og veldig høy utdanning til 1.

```
data$utdanning_2 <- ifelse(data$utdanning_1 == "lav", 0,
                           ifelse(data$utdanning_1 == "middels", 0,
                                  ifelse(data$utdanning_1 == "hoy", 1,
                                         ifelse(data$utdanning_1 == "veldig hoy", 1,
                                                ifelse(data$utdanning_1)))))

# Ser at det blir riktig

table(data$utdanning_2)
```

```
##
##    0    1
## 2524   38

table(data$utdanning_1, data$utdanning_2)

##
##                  0    1
##   hoy            0   37
##   lav         1034    0
##   middels     1490    0
##   veldig hoy     0    1

# Vi kan plotte variablene dataene våre:

ggplot(data, aes(utdanning_1)) +
  geom_bar()
```




```
ggplot(data, aes(utdanning_2)) +
  geom_bar() +
  scale_x_discrete() +
  xlab("Utdanning") +
  ylab("frekvens")
```



Før vi går videre til regresjon så sjekker vi klasse og fjerner de variablene vi ikke vil ha i datasettet. Jeg bruker igjen select og subsetter. Deretter koder jeg om den nye kjonnvariabelen til å bare hete kjonn.

```
str(data)
```

```
## 'data.frame': 2562 obs. of  8 variables:
##  $ internettbruk: int  5 5 1 1 5 1 5 1 4 1 ...
##  $ kjonn        : int  2 1 2 2 2 1 2 2 1 2 ...
##  $ alder        : int  67 45 73 86 53 77 35 66 52 86 ...
##  $ utdanning    : int  18 11 8 3 17 18 18 16 10 3 ...
##  $ tillit       : int  8 6 0 6 6 0 3 6 6 7 ...
##  $ kjonn_1      : num  1 0 1 1 1 0 1 1 0 1 ...
##  $ utdanning_1  : chr  "middels" "middels" "lav" "lav" ...
##  $ utdanning_2  : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
# Vi subsetter og tar med de variablene vi er interessert i
```

```r
df <- data %>%
  select(internettbruk, alder, utdanning, tillit, kjonn_1)

df <- df %>%
  rename(kjonn = kjonn_1)
```

REGRESJON MED NUMERISK UAVHENGIG VARIABEL For å kjøre en lineær regresjon i R så bruker vi funksjonen lm(). lm() har følgende syntaks:

```r
lm(avhengig_variabel ~ uavhengig_variabel, data = mitt_datasett)
```

Dersom datasettet ditt har manglende informasjon (missing/NA) så må du legge til et element som sier hvor- dan regresjonen skal forholde seg til dette. Ved å legge til na.action = "na.exclude" i lm()så beholder R informasjon om hvilke observasjoner som mangler data.

Vi skal nå kjøre en regresjon med internettbruk og alder. Vi har fjernet NA så vi trenger ikke na.action-argumentet. Resulatene fra regresjonen vi får ved hjelp av summary() og stargazer() er veldig nyttig. Vi får vite koeffisientene, standardfeilene og informasjon vi kan bruke til å evaluere modellen vår. I seminar skal vi bruke en del tid på å tolke disse tabellene.

```r
mod <- lm(internettbruk ~ alder, data = df)
summary(mod)

##
## Call:
## lm(formula = internettbruk ~ alder, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4400 -0.9129 -0.0071  0.9378  3.1973
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.487113   0.070921   91.47   <2e-16 ***
## alder       -0.055111   0.001292  -42.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.255 on 2560 degrees of freedom
## Multiple R-squared:  0.4153,Adjusted R-squared:  0.4151
## F-statistic:  1818 on 1 and 2560 DF,  p-value: < 2.2e-16
```

```r
# For mer oversikt bruk stargazer

stargazer(mod, type = "text")
```

```
##
## ================================================
##                          Dependent variable:
##                      ---------------------------
##                             internettbruk
## ------------------------------------------------
## alder                         -0.055***
##                                (0.001)
##
## Constant                      6.487***
##                                (0.071)
##
## ------------------------------------------------
## Observations                    2,562
## R2                              0.415
## Adjusted R2                     0.415
## Residual Std. Error       1.255 (df = 2560)
## F Statistic          1,818.322*** (df = 1; 2560)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

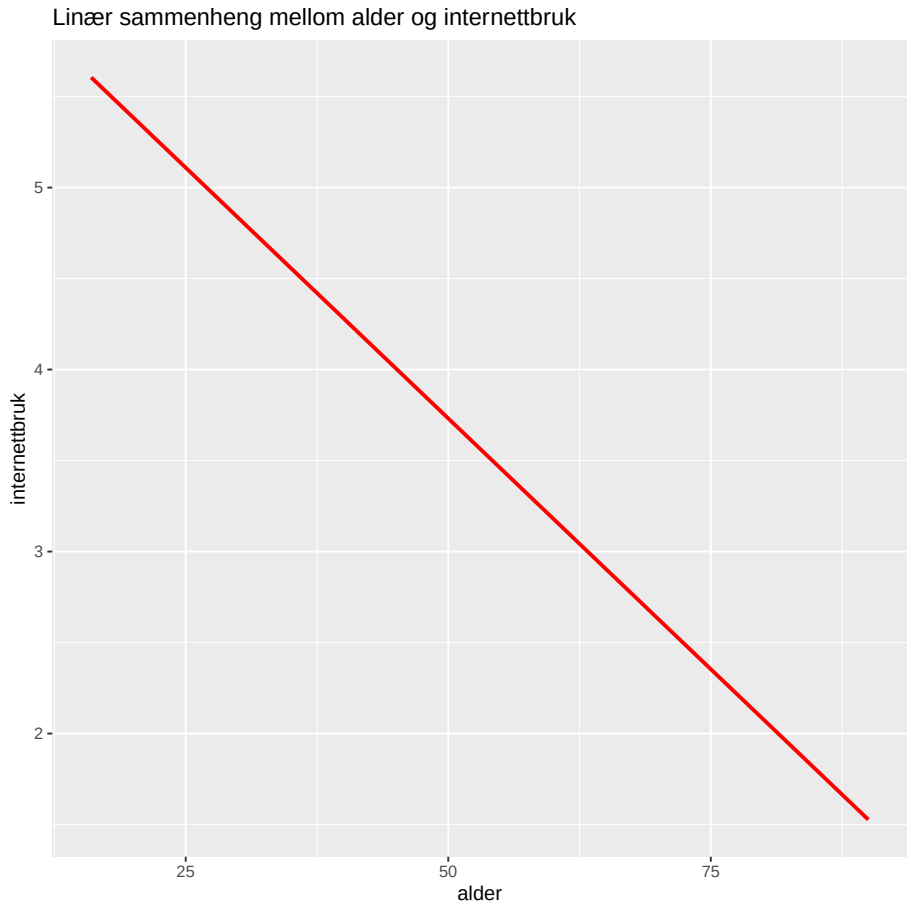
Vi må tolke resultatene vi får ut. Da kan det være viktig å først bare tenke på retning og signfikans. Deretter bør vi gå tilbake å tenke på hva er analyseenheten/observasjonsenheten og hvilket målenivå er variablene jeg analyserer? Vi må tolke både konstantledd og estimatet.

Ved hjelp av stargazer så kan vi også lagre tabeller lokalt på PC-en som vi kan bruke i word-dokumenter og liknende. Da endrer vi på type argumentet og legger til et out argument. out argumentet forteller i hvilken mappe du vil lagre filen samt hva filen skal hete. Da får du en .htm-fil som ser omtrent ut som i eksempelet under. Den kan du høyreklikke på og velge åpne i word dersom du skal ha tabellen inn i en oppgave eller liknende. Dette viste jeg sist gang.

Jeg synes ofte det vanskelig å tolke tallene, derfor er det så viktig å plotte tallene man får ut. Akkurat slik vi gjore sist gang. Det som er fint nå er at med modellen vi har laget har vi en OLS linje som vi kan plotte. Det er nå vi skal bruke modelr-pakken. Denne står det mye mer om i kapittel 23 i R for data Science som er lenket til på Canvas. I plottet under skal vi plotte prediksjonene våre – altså de predikerte verdiene som model vår estimerer på bakgrunn av observasjonene i dataene. Vi lagrer prediksjonene i et eget objekt som jeg kaller prediksjoner.

```r
prediksjoner <- df %>%
  add_predictions(mod, "internettbruk")

# add_pred tar to argumenter: model og avhengig var

# Ggplot
ggplot(df, aes(alder, internettbruk)) +
  geom_smooth(data = prediksjoner, colour = "red",
              size = 1) +
  ggtitle("Linær sammenheng mellom alder og internettbruk")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs =
"cs")'
```



```
# Legg merge til at linjen kommer fra prediksjons-dataene vi nettopp lagde.
```

Vi kan se konfidensintervallet til estimatet og konstantleddet med følgende kode.

```
confint(mod)
```

```
##                   2.5 %      97.5 %
## (Intercept)  6.3480442  6.62618271
## alder       -0.0576456 -0.05257699
```

Regresjon med dikotom uavhengig variabel

Vi skal nå kjøre regresjon med tillit og kjønn. Dette gjør vi mest for å øve oss på å tolke resultater. Tenk på målenivå de to ulike variablene.

```
mod1 <- lm(tillit ~ kjonn, data = df)
summary(mod1)
```

```
##
## Call:
## lm(formula = tillit ~ kjonn, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3666 -2.1710  0.6334  1.8290  5.8290
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.36664    0.07215  60.521   <2e-16 ***
## kjonn       -0.19565    0.09936  -1.969    0.049 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.511 on 2560 degrees of freedom
## Multiple R-squared:  0.001512,Adjusted R-squared:  0.001122
## F-statistic: 3.878 on 1 and 2560 DF,  p-value: 0.04904
```

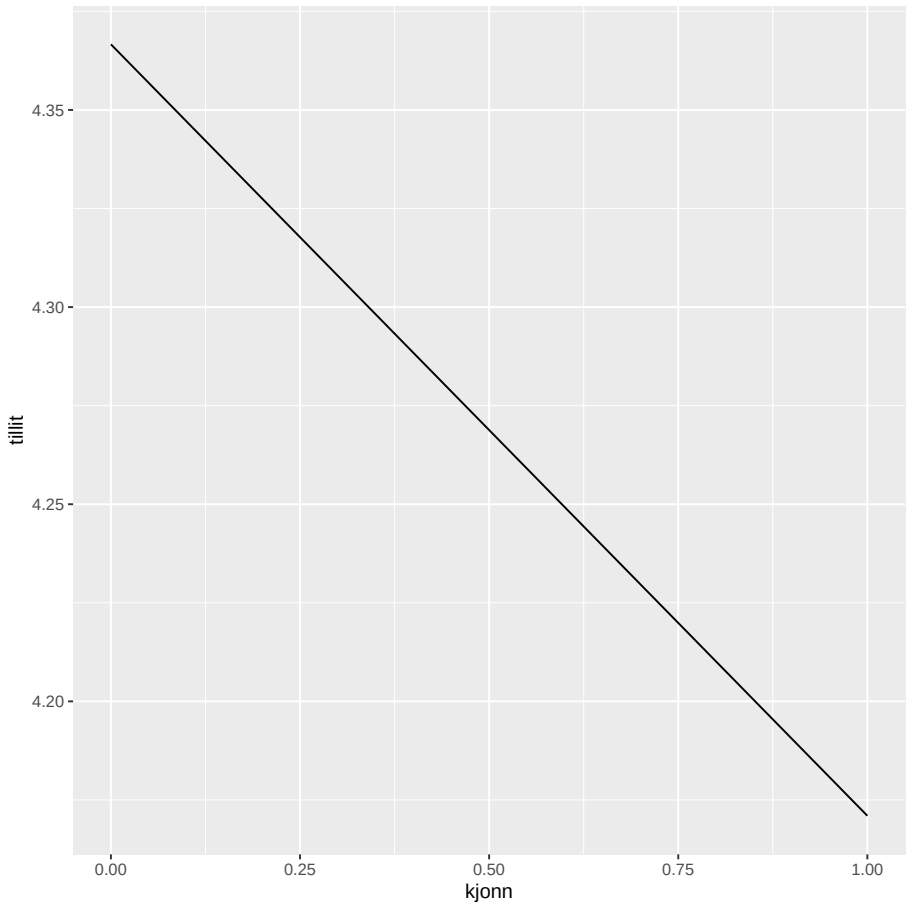
```
# Vi kan også printe disse resultatene med
# stargazer

stargazer(mod1, type = "text")
```

```
##
## ===============================================
```

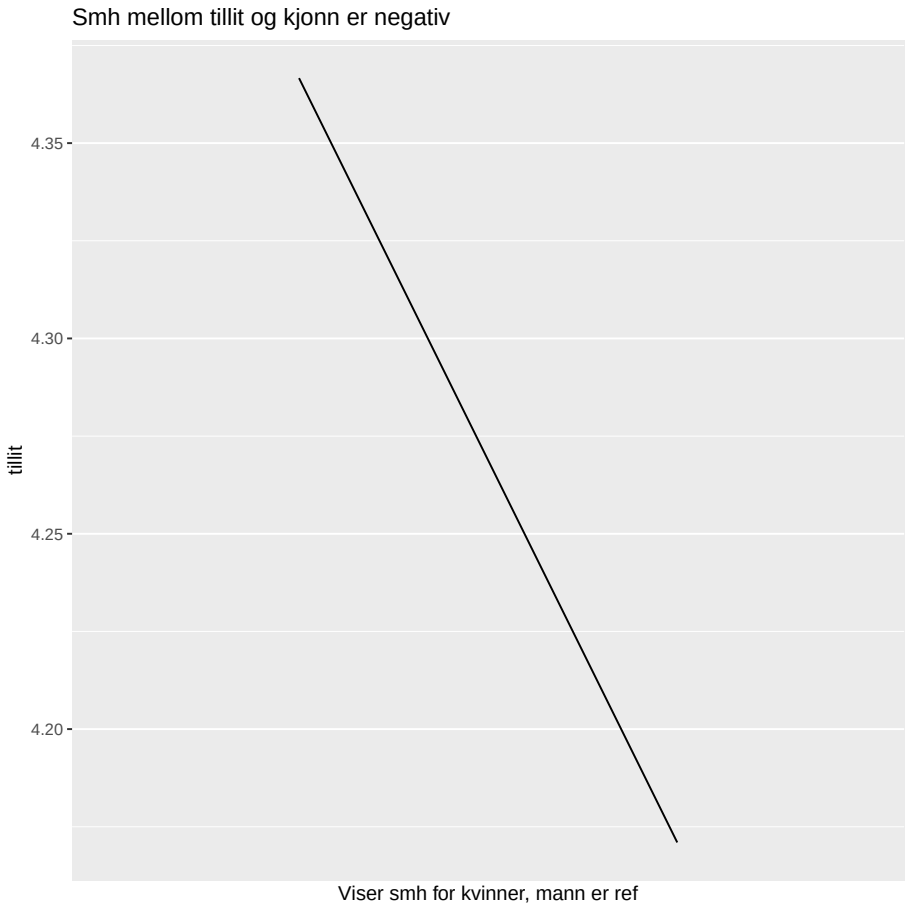
```
##                         Dependent variable:
##                     ---------------------------
##                                 tillit
## ------------------------------------------------
## kjonn                          -0.196**
##                                 (0.099)
##
## Constant                       4.367***
##                                 (0.072)
##
## ------------------------------------------------
## Observations                     2,562
## R2                               0.002
## Adjusted R2                      0.001
## Residual Std. Error       2.511 (df = 2560)
## F Statistic           3.878** (df = 1; 2560)
## ================================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

Vi plotter også resultatene fra denne regresjonen.

```
prediksjoner1 <- df %>%
  add_predictions(mod1, "tillit")


# Det ser litt rart ut
ggplot(df, aes(kjonn, tillit)) +
  geom_line(data = prediksjoner1)
```




```
# Vi kan legge på argumenter:
ggplot(df, aes(kjonn, tillit)) +
  geom_line(data = prediksjoner1) +
  scale_x_discrete() +
  ggtitle("Smh mellom tillit og kjonn er negativ") +
  xlab("Viser smh for kvinner, mann er ref")
```


Smh mellom tillit og kjonn er negativ
4.35
4.30
4.25
4.20
tillit
Viser smh for kvinner, mann er ref