R-seminar 4: Bivariat analyse

STV1020 Vår 2021

Uke 14

Dette skal vi gjennomgå i seminaret

1. Laste inn data
2. Missing, NA, Not Available
3. Statistiske mål
4. Univariat analyse
5. Bivariat analyse

Opplegg

1. **Laste inn data**

   Det finnes mange typer av data. Datasettet vi skal bruke i dette seminaret er en csv-fil. Da må vi bruke koden read.csv

```
## Setter working directory
setwd()


## Error in setwd():  argument "dir" is missing, with no default


## Laster inn datasett fra github
data <- read.csv("https://raw.githubusercontent.com/louisabo/STV4020A/master/SEMINAR3/internet

## Lagrer datasett til R-type i working directory
save(data, file = "internettbruk.Rdata")

rm(data)

load("internettbruk.Rdata")

## Henter opp relevante pakker fra biblioketet
library(tidyverse)
```

   Bare som en illustrasjon, kan vi bruke koden "save" for å lagre datasettet (slik som i seminar 3). Spesifiserer vi .Rdata lagrer vi det som en R-fil. Ser dere

.rda noe sted, er det bare en eldre måte enn Rdata å spesifisere R-filer på. Vi velger datasettet i global environment (det heter data) også velger vi navnet vi vil lagre filen som, her internettbruk. Legg merke til at vi skriver ".Rdata" – som indikerer filformatet. Du kan feks også laste inn data fra excel eller lagre det som excel, men da må du laste ned en pakke som gjør det.

Datasettet heter internettbruk og omhandler internettbruken til italienere. Det betstår av et utvalg variabler hentet fra European Social Survey (ESS) runde 9 (2018). Enhetene er italienske statsborgere og samlet innholder datasettet 2745 observasjoner og 5 variabler:

(a) Kjonn – Mann = 1, Kvinne = 2

(b) Alder – Alder til respondenten

(c) Utdanning – Antall år med fullført utdanning

(d) Tillit – Tillit til det italienske parlament (0-10), 0 = ingen tillit, 10 = fullstendig tillit

(e) Internettbruk – Hvor ofte bruker respondenten internett? (1-5), 1 = aldri, 5 = hver dag.

Før vi går videre vil vi se på dataene våre. Disse kodene her har dere sikkert sett før:

```
## Inspiserer datasettet
View(data)
head(data)
names(data)
summary(data) # Denne viser alt, målenivå, NAs, gjennomsnitt osv
```

2. **Missing - NA - NOT AVAILBLE**

Det finnes mange grunner til at det er tomme celler/manglende verdier eller svar i dataene. Vi skal vise hvordan vi kan finne missing verdier og hva man kan gjøre med de. Det er viktig å teoretisk begrunne hvordan man håndterer NA-verdier på bakgrunn av utvalget av populasjonen. Er missing-verdier systematiske eller er de tilfeldige? Når vi skal finne missing er det mest vanlig er å bruke følgende kode:

```
sum(is.na(data))

## [1] 200

# Denne teller totalt antall missing i data. Kan være flere missing på en rad.

sum(is.na(data$internettbruk))

## [1] 5
```

```r
# Viser hvor mange missing det er på en variabel

# Sjekker complete cases: dvs hvor mange observasjoner som ikke har missing på én eller flere
sum(complete.cases(data))


## [1] 2562
```

Complete cases viser hvor mange observasjoner som er fullstendige, altså ikke har missing verdier.

Du kan også bruke summary(), som gir oss masse informasjon om hver enkelt variabel. Legg merke til at NAs er på slutten.

```r
summary(data)


##  internettbruk       kjonn           alder         utdanning
##  Min.   :1.000   Min.   :1.000   Min.   :16.00   Min.   : 0.0
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.:36.00   1st Qu.: 8.0
##  Median :5.000   Median :2.000   Median :52.00   Median :12.0
##  Mean   :3.629   Mean   :1.527   Mean   :51.28   Mean   :11.5
##  3rd Qu.:5.000   3rd Qu.:2.000   3rd Qu.:67.00   3rd Qu.:14.0
##  Max.   :5.000   Max.   :2.000   Max.   :90.00   Max.   :37.0
##  NA's   :5                       NA's   :21      NA's   :85
##      tillit
##  Min.   : 0.000
##  1st Qu.: 2.000
##  Median : 5.000
##  Mean   : 4.251
##  3rd Qu.: 6.000
##  Max.   :10.000
##  NA's   :89
```

Prøv å se om du forstår hva som står på hjelpefilen for NA. Vanligvis må vi beskrive hvordan NA er. Vi må også velge hva vi skal gjøre med dem. Veldig vanlig er å fjerne NA hvis de er 'missing at random' eller missing completely at random.' Du kan velge å fjerne alle missing verdier eller bare missing verdier på spesifikke variable. Når vi begynner med analyser så vil R ta høyde for de tomme cellene, R fjerner dem automatisk. Det er som når vi bruker gjennomsnittet – man kan ikke regne gjennomsnittet av missing, derfor må vi si til R hvordan R skal håndtere missing.

```r
## Fjerner alle missing -- dvs alle observasjoner som har missing
no_na_data <- data %>%
  drop_na()
# Vi får like mange observasjoner som det er complete cases

## Fjerne missing på en variabel (eller flere)
no_na_data1 <- data %>%
  drop_na(internettbruk) # Du kan legge til flere variable med komma

sum(is.na(no_na_data1$internettbruk))
```

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

3. **Statistiske mål**

   Statistiske mål forteller oss noe om fordelingen til ulike variabler, som for eksempel gjennomsnitt, median og standardavvik, men også minimum- og maksimumverdier. Statistiske mål på sentraltendens er gjennomsnitt, median og modus. Statistiske mål på spredning i dataene er standardavviket og varians. Det er lurt å se på de statistiske målene, og plotte variablene også.

   For å finne statistiske mål raskt, er summary()-funksjonen fin.

```
summary(data)
```

```
##  internettbruk       kjonn           alder         utdanning
##  Min.   :1.000   Min.   :1.000   Min.   :16.00   Min.   : 0.0
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.:36.00   1st Qu.: 8.0
##  Median :5.000   Median :2.000   Median :52.00   Median :12.0
##  Mean   :3.629   Mean   :1.527   Mean   :51.28   Mean   :11.5
##  3rd Qu.:5.000   3rd Qu.:2.000   3rd Qu.:67.00   3rd Qu.:14.0
##  Max.   :5.000   Max.   :2.000   Max.   :90.00   Max.   :37.0
##  NA's   :5                       NA's   :21      NA's   :85
##      tillit
##  Min.   : 0.000
##  1st Qu.: 2.000
##  Median : 5.000
##  Mean   : 4.251
##  3rd Qu.: 6.000
##  Max.   :10.000
##  NA's   :89
```

   Hva forteller dette oss, for hver enkelt variabel?

   For å kun finne gjennomsnittet til en variabel i datasettet kan vi bruke funksjonen mean(). Det samme gjelder for de andre statistiske målene.

```
mean(data$internettbruk, na.rm = TRUE) # Må fjerne missingverdier
```

```
## [1] 3.628832
```

```
# Hva blir gjennomsnittlig internettbruk blant respondentene?
mean(data$kjonn, na.rm = TRUE) # Gir det mening å ta gjennomsnitt til kjønn?
```

```
## [1] 1.52714
```

```
median(data$internettbruk, na.rm = TRUE)
```

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

```
max(data$internettbruk, na.rm = TRUE)
```

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

```
min(data$internettbruk, na.rm = TRUE)
```

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

Det er viktig å vite variabelenes målenivå. Hvilke statistiske mål som er relevante, avhenger av variablenes målenivå. Å ta gjennomsnittet til kjønn gir ikke mening, fordi det er en kategorisk variabel.

Standardavvik er et statistisk mål, og det viser respondentenes gjennomsnittlige avstand fra gjennomsnittet. Vi kan bruke funksjonen sd().

```
sd(data$internettbruk, na.rm = TRUE)
```

```
## [1] 1.645191
```

```
# Hva forteller dette standardavviket oss?
```

Variansen er standardavviket opphøyd i annen. Dermed er standardavviket kvadratroten av variansen. Det er enklere å tolke standardavvik enn varians. Vi ser likevel på hvordan man finner variansen.

```
# Lagrer variansen i et eget objekt
varians <- var(data$internettbruk, na.rm = TRUE)
sqrt(varians) # bruker funksjonen sqrt() for å finne kvadratroten
```

```
## [1] 1.645191
```

4. Univariat analyse: Deskriptiv statistikk med én variabel

Når vi kun har én variabel vi vil beskrive, har vi å gjøre med univariate fordelinger. Da blir vi kjent med variablene hver for seg. En univariat fordeling gir oss informasjon om hvordan observasjonene fordeler seg på en variabels ulike verdier. Igjen gir summary()-funksjonen en rask oversikt over statistiske mål og deskriptiv statistikk. Det er her nyttig å gjøre seg godt kjent med de ulike statistiske målene. Men den univariate analysen kan ta ting et skritt videre, med for eksempel tabeller og histogrammer.

```
summary(data)
```

```
##  internettbruk       kjonn           alder         utdanning
##  Min.   :1.000   Min.   :1.000   Min.   :16.00   Min.   : 0.0
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.:36.00   1st Qu.: 8.0
##  Median :5.000   Median :2.000   Median :52.00   Median :12.0
```

```
##  Mean   :3.629   Mean   :1.527   Mean   :51.28   Mean   :11.5
##  3rd Qu.:5.000   3rd Qu.:2.000   3rd Qu.:67.00   3rd Qu.:14.0
##  Max.   :5.000   Max.   :2.000   Max.   :90.00   Max.   :37.0
##  NA's   :5                       NA's   :21      NA's   :85
##      tillit
##  Min.   : 0.000
##  1st Qu.: 2.000
##  Median : 5.000
##  Mean   : 4.251
##  3rd Qu.: 6.000
##  Max.   :10.000
##  NA's   :89
```

```
# Det er også lurt å gjøre seg kjent med målenivået til variablene.
# str() finner ut det.
str(data)
```

```
## 'data.frame': 2745 obs. of  5 variables:
##  $ internettbruk: int  5 5 1 5 1 5 1 5 1 4 ...
##  $ kjonn        : int  2 1 2 1 2 2 1 2 2 1 ...
##  $ alder        : int  67 45 73 21 86 53 77 35 66 52 ...
##  $ utdanning    : int  18 11 8 8 3 17 18 18 16 10 ...
##  $ tillit       : int  8 6 0 NA 6 6 0 3 6 6 ...
```

For kategoriske variabler, på nominalnivå eller ordinalnivå, kan vi bruke frekvenstabeller for å beskrive dataene med tall, og søylediagram for å beskrive dataene grafisk.

Variabelen for kjønn er kategorisk og på nominalnivå. En frekvenstabell forteller oss hvor mange respondenter som er menn og hvor mange som er kvinner. Vi kan bruke funksjonen table(). Disse viser den absolutte fordelingen, altså totalt antall observasjoner for hver verdi. Vi kan også få den relative fordelingen mellom kategoriene, som viser prosentvis fordeling. Vi bruker prop.table()-funksjonen

```
## Frekvenstabell for kjønn
table(data$kjonn)
```

```
##
##    1    2
## 1298 1447
```

```
# Lagrer table i et objekt
tabell <- table(data$kjonn)

## Relativ fordeling i prosent
tabell_pct <- prop.table(table(data$kjonn))*100
tabell_pct
```

```
##
##        1        2
## 47.28597 52.71403
```

Vi kan gjøre det samme for internettbruk, som er på ordinalnivå.

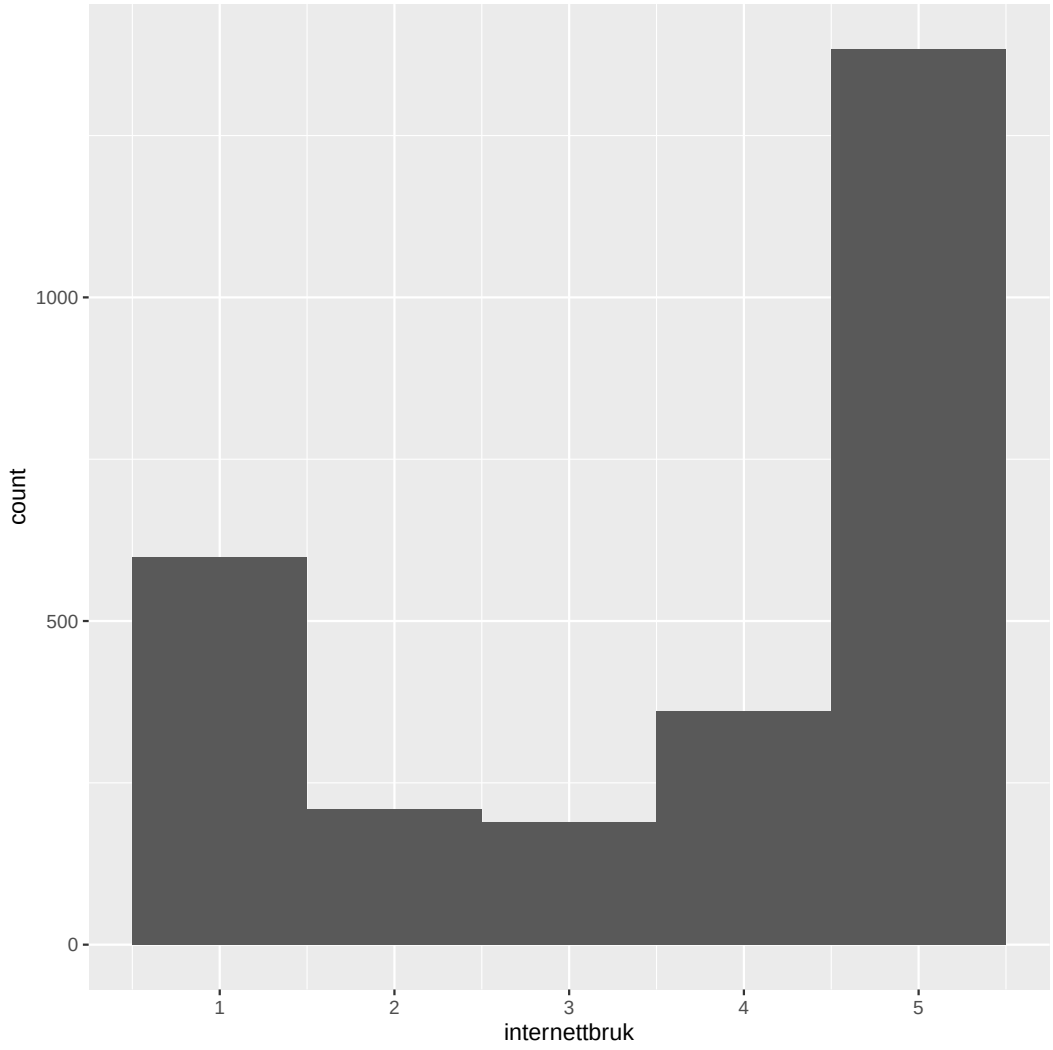
```
tabell2 <- table(data$internettbruk)
tabell2


##
##    1    2    3    4    5
##  598  209  189  360 1384
```

Det er alltid et poeng å lage grafer og figurer for å beskrive dataene. Det gir et godt visuelt og mer intutitvt inntrykk av dataene. For kategoriske variabler kan vi lage søylediagram for å beskrive frekvensfordelingene til variablene.

For å få søylediagram bruker vi funksjonen ggplot-funksjonen som er i pakken tidyverse.

```
# Søylediagram for internettbruk
ggplot(data, aes(internettbruk)) +
  geom_bar(width = 1)
```

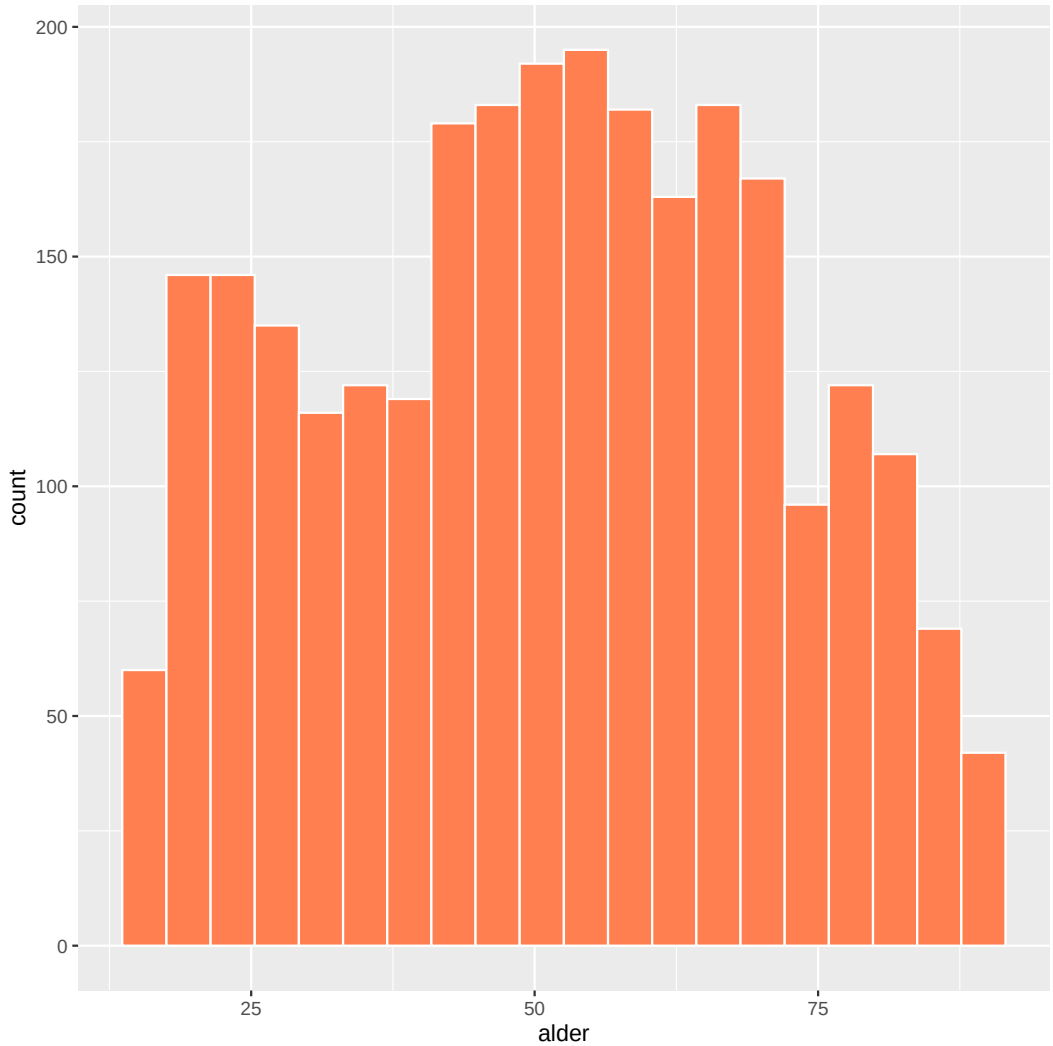


```
# Her ser vi tydelig at det er flest som oppgir 5 som alternativ

# Prøv å legg på titler osv...
```

Grafiske fremstillinger er også nyttig med kontinuerlige variabler. Da kan vi blant annet bruke histogrammer. Den deler opp i "kategorier". Vi bruker ggplot, men endrer geom-argumentet. Legg merke til argumentet bins – dette bestemmer hvor mange søyler vi ønsker. Prøv å endre argumentet å se hva som skjer. Vi kan som kjent også legge til flere argumenter. Bruk hjelpefilen til ggplot, eller søk rundt på nettet.

```
ggplot(data, aes(alder)) +
        geom_histogram(bins = 20,
                       fill = "coral",
                       color = "white")
```



I en større oppgave ønsker man ofte å presentere alle variablenes deskriptive statistikk i en felles tabell. Funksjonen stargazer() er fin til å gjøre dette. Først

må vi innstallere pakken (hvis det ikke er gjort fra før), og hente den opp fra biblioteket.

```
# Installerer stargazer-pakken og henter den fra biblioteket
install.packages("stagazer")


## Error in contrib.url(repos, "source"):  trying to use CRAN without setting
a mirror


library(stargazer)

stargazer(data,
          type = "text")


##
## ===============================================================
## Statistic       N    Mean  St. Dev.  Min   Pctl(25) Pctl(75)  Max
## ---------------------------------------------------------------
## internettbruk 2,740 3.629   1.645   1.000   2.000    5.000   5.000
## kjonn         2,745 1.527   0.499     1       1        2       2
## alder         2,724 51.277  19.429  16.000  36.000   67.000  90.000
## utdanning     2,660 11.504  4.331   0.000   8.000    14.000  37.000
## tillit        2,656 4.251   2.525   0.000   2.000    6.000   10.000
## ---------------------------------------------------------------
```

Tabellen kan også gjøres om til html-format, som vi kan åpne i nettleseren, kopiere og lime inn i et word-dokument.

```
stargazer(data,
          type = "html",
          out = "deskriptiv.html")
```

5. **Bivariat analyse: Deskriptiv statistikk med to variabler**

   Bivariat analyse brukes når man analyserer to variabler, og er nyttig for å få oversikt over sammenhengen mellom to variabler. I tillegg forteller det oss noe om hvor mye to variabler korrelrerer, altså hvor mye de henger sammen. Bivariat statistikk er også nyttig for å teste korrelasjonens statistiske signifikans.

   Dersom vi har to kategoriske variabler vi ønsker å sammenlikne, kan vi presentere dem i en krysstabell. Ta bruker vi funksjonen table(). Vi kan opprette en krysstabell mellom internettbruk og kjønn i et nytt objekt kalt krysstabell.

```
krysstabell <- table(data$internettbruk, data$kjonn)
krysstabell


##
##       1   2
##   1 227 371
##   2  90 119
```

```
##   3  92  97
##   4 184 176
##   5 702 682


# Tolk tabellen. Er det noen forskjell på hvor ofte menn og kvinner bruker
# internett?
```

Denne tabellen oppgir frekvensfordelingen i absolutte tall. Vi kan også finne relative tall, altså andeler.

```
prop.table(krysstabell, margin = 1)*100


##
##            1        2
##   1 37.95987 62.04013
##   2 43.06220 56.93780
##   3 48.67725 51.32275
##   4 51.11111 48.88889
##   5 50.72254 49.27746


# margin = 1 brukes for å regne ut fordelingen per linje, feks hvor mange
# menn relativt til kvinner som oppgir 1 på skalaen for internettbruk
```

Kjikvadrattesten tester sammenhengen mellom to kategoriske variabler. Den sammenlikner krysstabellen vi har, men en hypotetisk tabell fra et annet utvalg der det ikke er noen sammeheng mellom variablene. Så tester den sannsynligheten for at tabellen vår er generert ved en tilfeldighet. Vi bruker funksjonen chisq.test()

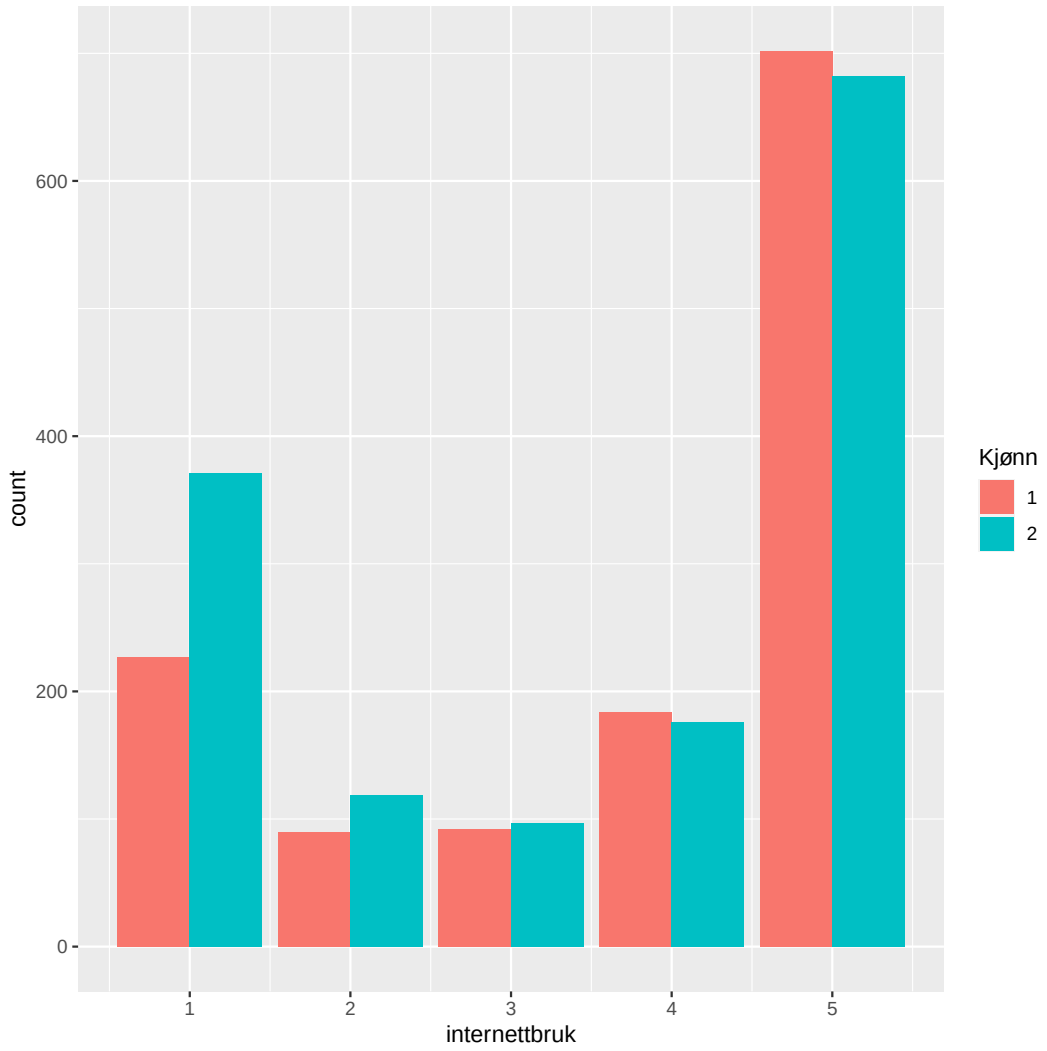
```
chisq.test(krysstabell)


##
##  Pearson's Chi-squared test
##
## data:  krysstabell
## X-squared = 31.18, df = 4, p-value = 2.813e-06


# X-squared, altså kjikvadratet til de to variablene er på 31.18
```

Vi kan lage søylediagrammer for å presentere sammenhengen grafisk. Igjen, det er alltid lurt, blant annet fordi det er lettere å se sammenhenger raskt.

```
ggplot(data, aes(x = internettbruk, fill = as.factor(kjonn))) +
  geom_bar(position = "dodge") + # "dodge", "fill" (relative tall), "stack"
  labs(fill = "Kjønn")
```



Vi avslutter med bivariat analyse med to kontinuerlige variabler. (Dette er en forsmak på bivariat regresjonsanalyse.) Hensikten med dette er å beskrive korrelasjonen mellom variablene. Vi kan beskrive denne sammenhengen med Pearsons r eller teste om korrelasjonen er statistisk signifikant.

Pearsons r beskriver styrken og retningen til korrelasjonen mellom to variabler. Den varierer fra -1 (negativ sammenheng) til 1 (positiv sammenheng). 0 indikerer ingen sammenheng. La oss teste med alder og utdanning. Vi bruker cor() funksjonen. Vi bruker pairwise.complete.obs, som betyr at vi beholder alle observasjoner med observasjoner på begge variablene.

```
# str(data)
R <- cor(x = data$alder,
         y = data$utdanning,
         use = "pairwise.complete.obs",
         method = "pearson") # pearson er default
R


## [1] -0.4090912
```

```
# Hva forteller dette oss?
```

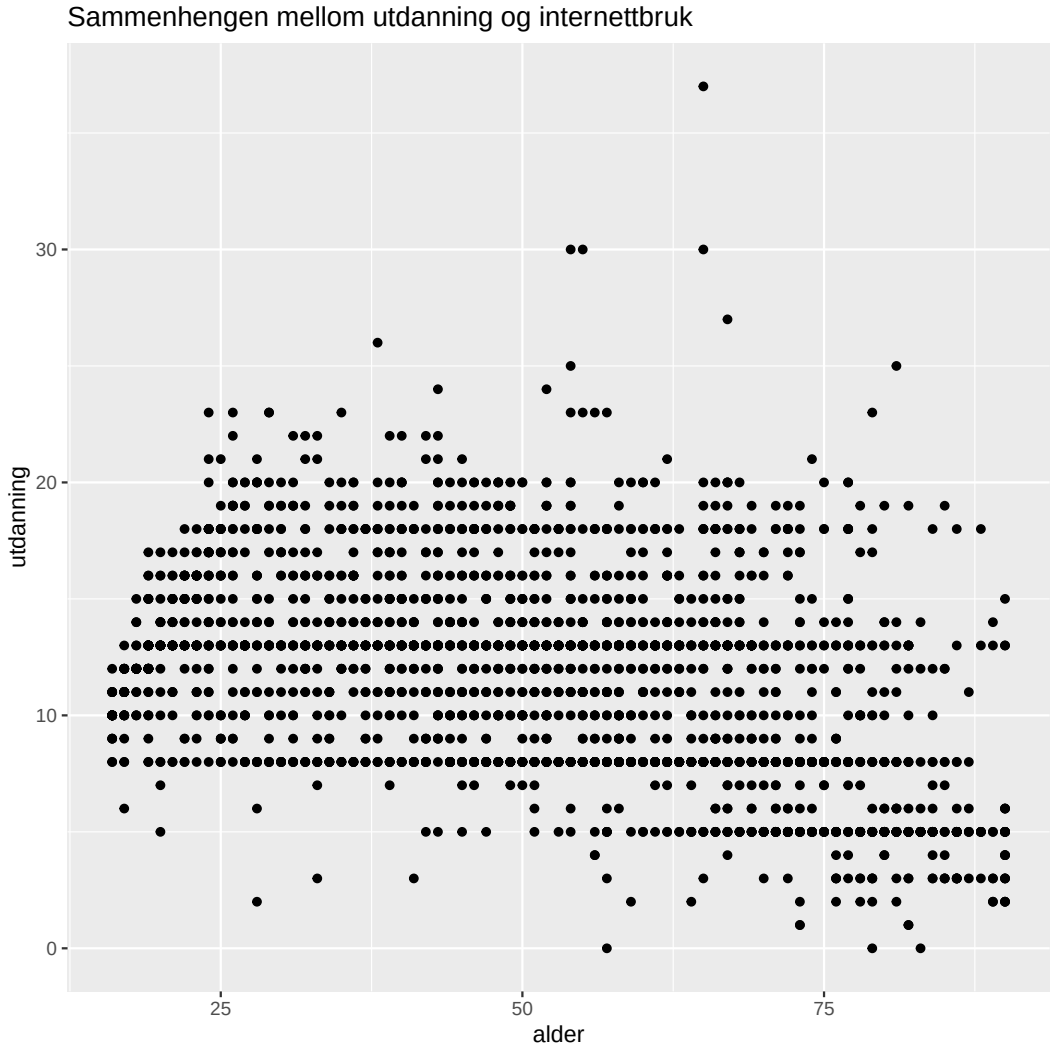
Vi kan også sette opp en korrelasjonsmatrise for å utforske alle de bivariate korrelasjonene i datasettet mellom de akutelle variablene.

```
cor(data,
    use = "pairwise.complete.obs",
    method = "pearson")
```

```
##               internettbruk       kjonn       alder  utdanning      tillit
## internettbruk     1.0000000 -0.10206670 -0.64360948  0.5583489  0.15587219
## kjonn            -0.1020667  1.00000000  0.06688781 -0.0528283 -0.04115814
## alder            -0.6436095  0.06688781  1.00000000 -0.4090912 -0.09849861
## utdanning         0.5583489 -0.05282830 -0.40909116  1.0000000  0.13911901
## tillit            0.1558722 -0.04115814 -0.09849861  0.1391190  1.00000000
```

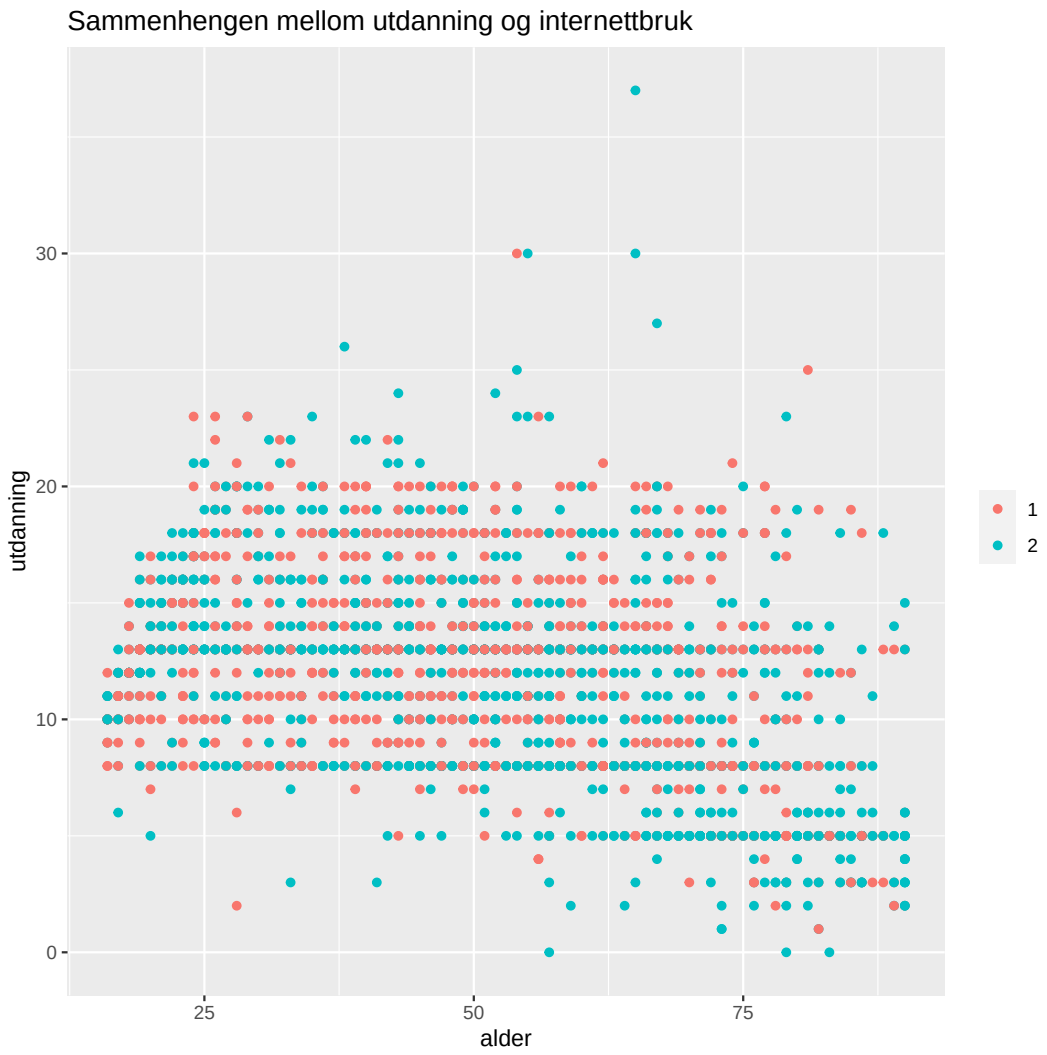
Spredningsdiagrammer egner seg godt for å grafisk fremstille sammenhengen mellom to kontinuerlige variabler. Den viser hvor hver respondent (observasjonsenhet) plasserer seg på x-aksen og y-aksen. Vi bruker ggplot med et annet geom-argument, nemlig point

```
## Spredningsdiagram alder og utdanning
ggplot(data, aes(alder, utdanning)) +
  geom_point() +
  labs(title = "Sammenhengen mellom utdanning og internettbruk")
```



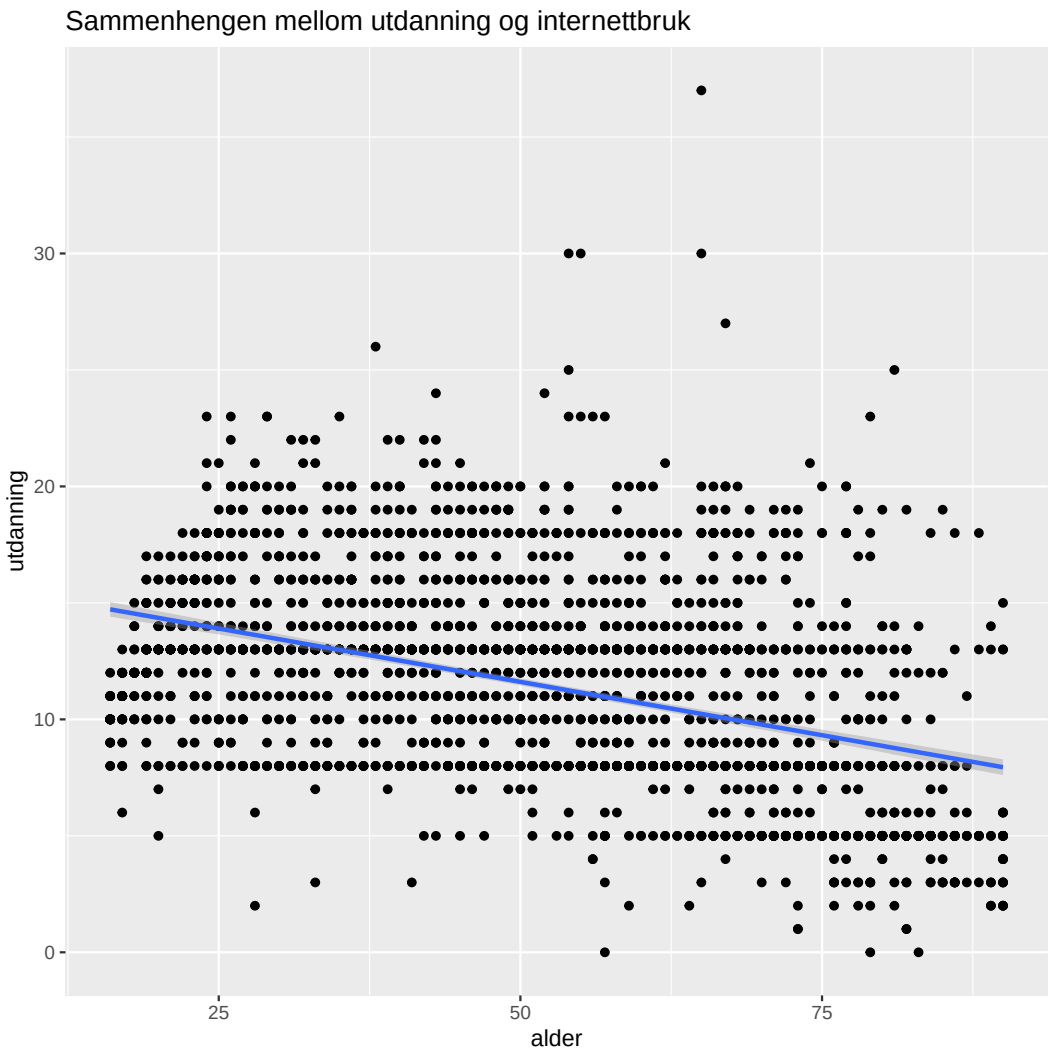

```
# Hva viser spredningsdiagrammet oss?

## Spredningsdiagram, med farge på kjønn
ggplot(data, aes(alder, utdanning, color = as.factor(kjonn))) +
  geom_point() +
  labs(title = "Sammenhengen mellom utdanning og internettbruk") +
  theme(legend.title = element_blank()) # Ingen tittel på legend
```




```
## Spredningsdiagram, med lineær linje (regresjonslinje) og punktestimater
ggplot(data, aes(alder, utdanning)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "Sammenhengen mellom utdanning og internettbruk")


## `geom_smooth()` using formula 'y ~ x'
```



Dette er begynnelsen på en regresjonsanalyse, som er tema for R-seminar 5.