Seminar 4: BIVARIAT ANALYSE

STV1020 Vår 2021

Dette skal vi gjennomgå i dette seminaret

1. Laste inn data og ulike typer av data.

   Det finnes mange typer av data.

```
  data <- read.csv("https://raw.githubusercontent.com/louisabo/STV4020A/master

save(data, file = "internettbruk.rda")

load("internettbruk.rda") #forutsetter setwd/prosjekt
```

   Bare som en illustrasjon så kan dere bruke koden "save" for å lagredata. Jeg velger datasettet i global environment (det heter data) også velger jeg et navn. Legg merke til at jeg skriver ".rda" – som indikerer filformatet. Du kan også laste inn f.eks data fra excel eller lagre det som excel, men da trenger du å laste ned en pakke som gjør det.

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

   (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 har dere sikkert sett før:

```r
# View()

# head()
# tail()

# summary() #denne viser alt -- målenivå, NAs, gjennomsnitt osv

# range() #spennet
# (data) #fordeling

# Vi kan også se på fordelinger vha av plotting.
```

2. **Missing - NA - NOT AVAILBLE** Det finnes mange grunner til at det er tomme celler/manglende verdier/svar i dataene. Vi skal vise hvordan vi kan finne missing verdier og hva man kan gjøre med de. Men 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

```r
sum(is.na(data)) # Teller total missing i data. Kan være flere missing på en rad.
```

```
## [1] 200
```

```r
sum(is.na(data$internettbruk)) # Viser hvor mange missing det er på en variabel
```

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

Du kan også bruke denne koden så får du opp 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.

```
# Bruk pakken tidyverse
# Subset
```

3. **Statistiske mål** (hva er på pensum?)

   Statistiske mål forteller oss noe om fordelingen til ulike variabler, som for eksempel gjennomsnitt, median og standardavvik, men også minimum- og maksimumverdier, typetall og frekvens. Statistiske mål på sentraltendens er gjennomsnitt, median og modus. Statistiske mål på spredning i dataene er standardavviket og varians.

   For å finne enkelte 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.

   Modus/typetall har ingen egen funksjon i R. Da må vi i så fall lage funksjonen selv.

   For å kun finne gjennomsnittet til en variabel i datasettet kan vi bruke funksjonen mean().

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

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

```
# Hva blir gjennomsnittlig internettbruk blant respondentene?
```

Her forsøker jeg å kun finne gjennomsnittet til kjønnsvariabelen.

```
  mean(data$kjonn,
       na.rm = TRUE)


## [1] 1.52714
```

Gir det mening å se på gjennomsnitt for kjønn? Det er viktig å vite variabelenes målenivå. Hvilke statistiske mål som er relevante, avhenger av variablenes målenivå.

Standardavvik er også 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. Jeg viser likevel 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
# tibble() og str() finner ut det. Vi må installere pakkene og hente
# de i biblioteket
install.packages("tibble")
```

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

```
library(tibble)
```

```
tibble(data)
```

```
## # A tibble: 2,745 x 5
##    internettbruk kjonn alder utdanning tillit
##            <int> <int> <int>     <int>  <int>
##  1             5     2    67        18      8
##  2             5     1    45        11      6
##  3             1     2    73         8      0
##  4             5     1    21         8     NA
##  5             1     2    86         3      6
##  6             5     2    53        17      6
##  7             1     1    77        18      0
##  8             5     2    35        18      3
##  9             1     2    66        16      6
## 10             4     1    52        10      6
## # ... with 2,735 more rows
```

```
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 kake- 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().

```
  table(data$kjonn)


##
##    1    2
## 1298 1447


# Lagrer table i et objekt, som vi kan eksportere til Word

tabell <- table(data$kjonn)
tabell1 <- data.frame(tabell)

# Bruker stargazer-pakken til å eksportere tabellen
#library(stargazer)
#library(texreg)
# hvordan eksportere tabellen
```

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
```

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

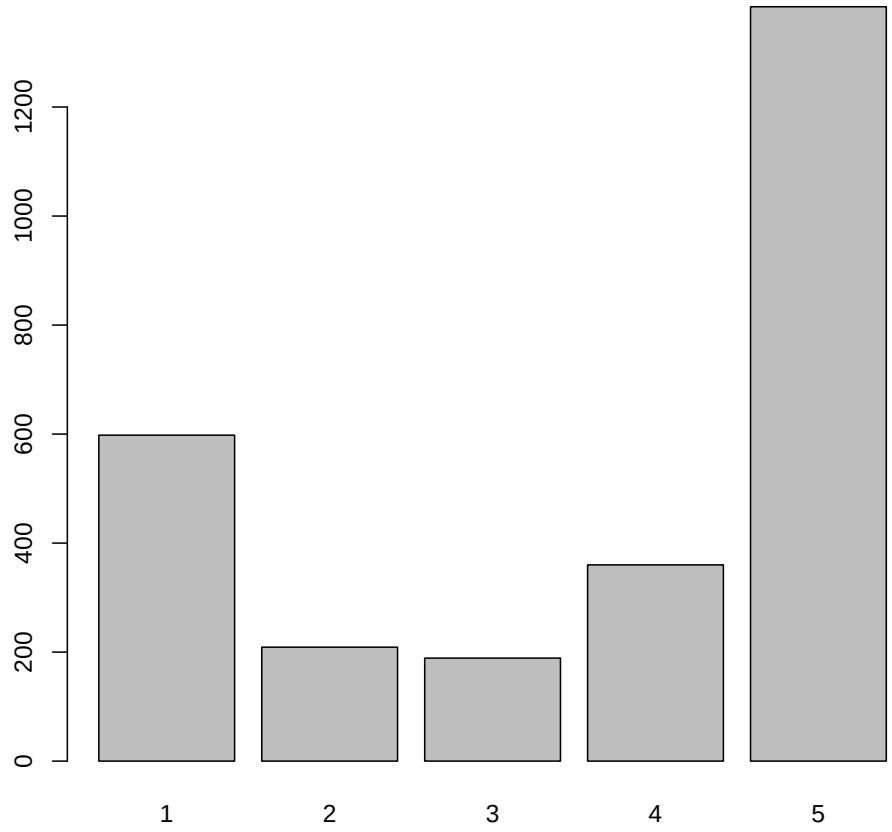
```
  tabell3 <- prop.table(table(data$internettbruk))
tabell3


##
##          1          2          3          4          5
## 0.21824818 0.07627737 0.06897810 0.13138686 0.50510949
```

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

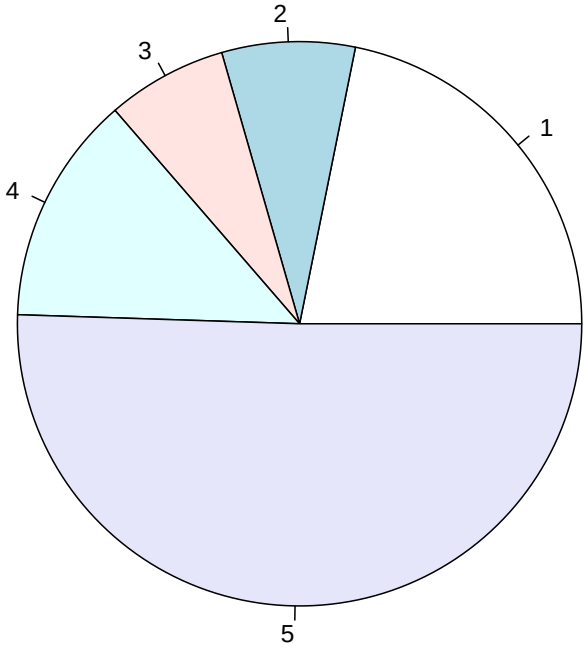
For å få søylediagram bruker vi funksjonen barplot(). Den illustrerer absolutte og relative fordelinger like godt.

For å få et kakediagram bruker vi funksjonen pie().

```
# Søylediagram for internettbruk
barplot(tabell2)
```




```
# Kakediagram for internettbruk
pie(tabell2)
```




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

For kontinuerlige variabler, på intervall- og forholdstallsnivå, kan vi også bruke frekvenstabeller. Dersom vi lager en frekvenstabell for alder, må vi omkode den til kategorier bed hjelp av cut()-funksjonen. I argumentet breaks = forteller jeg R hvor mange kategorier det skal være, og hvor bruddet skal være. Først undersøker jeg variabelen.

```
  summary(data$alder) # Min er 16 år, og maks er 90 år. Lager så kategorier
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   16.00   36.00   52.00   51.28   67.00   90.00      21
```

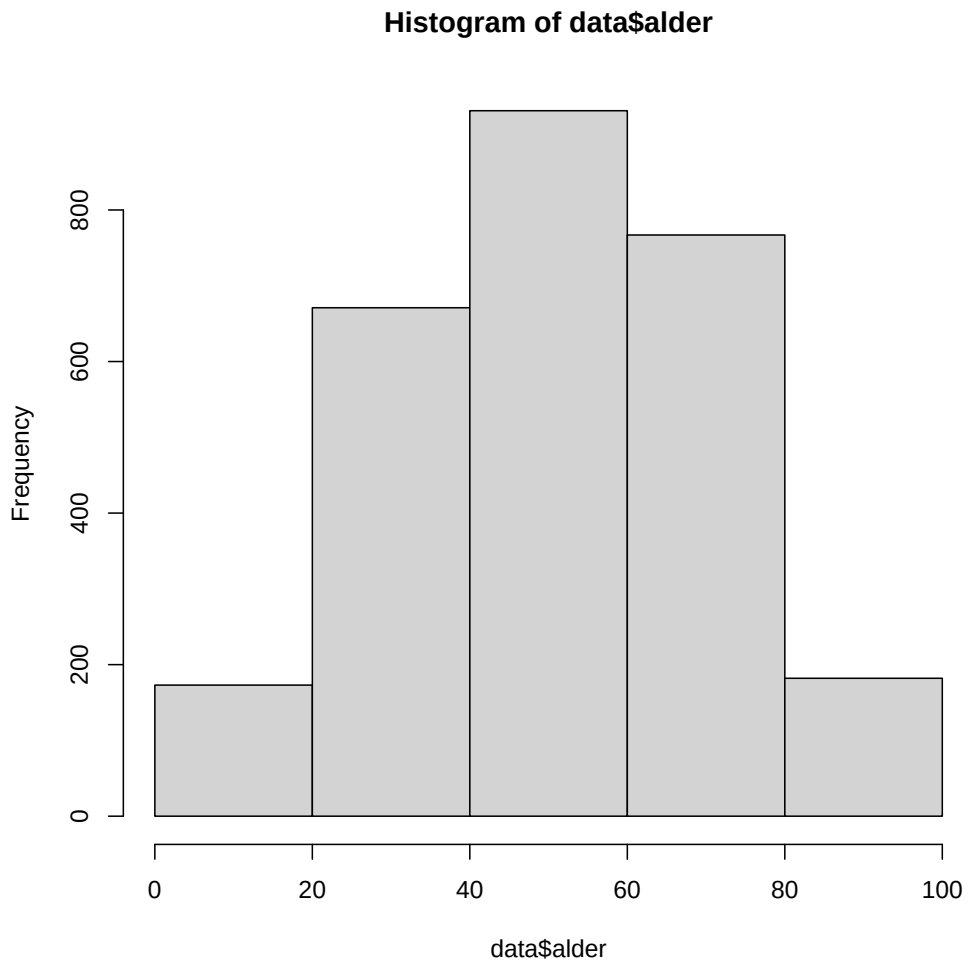
```
omkodet_alder <- cut(data$alder,
                     breaks = c(16, 30, 45, 60, 75, 90)) # 16-30
# 31-45
# 46-60
# 61-75
# 76-90
table(omkodet_alder)
```

```
## omkodet_alder
## (16,30] (30,45] (45,60] (60,75] (75,90]
##     487     548     712     609     340
```

Hva finner vi?

Grafiske fremstillinger er også nyttig med kontinuerlige variabler. Da kan vi blant annet bruke histogrammer. Også her må vi dele opp i kategorier. hist()-funksjonen gjør dette for oss, og vi trenger ikke gjøre det "for hånd", slik som ovenfor.

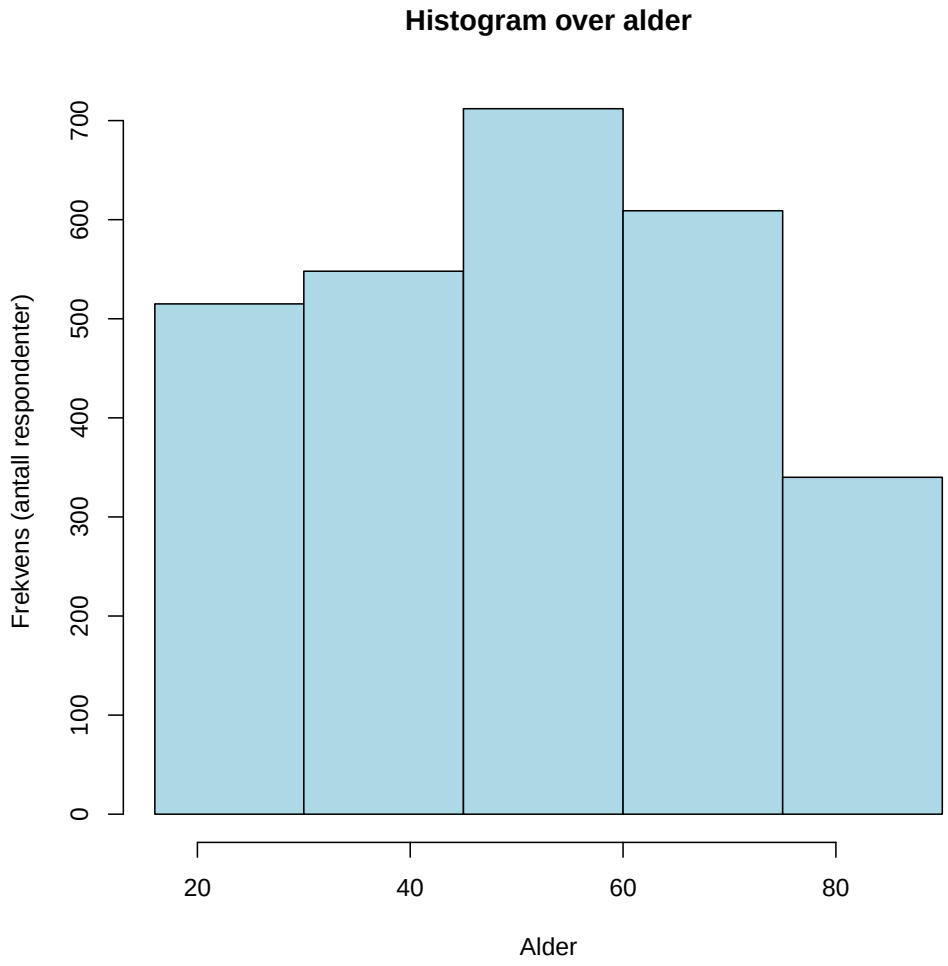
```
  hist(data$alder,
       breaks = 5) # Vi definerer fem kategorier, som brytes der R ønsker
```




```
# Virker bruddpunktet godt tilpasset dataen her?

# Vi kan også definere våre egne bruddpunkter for kategoriene
hist(data$alder,
     breaks = c(16, 30, 45, 60, 75, 90),
```

```
     probability = FALSE, # FALSE for å få frekvens
     # Vi kan også legge til flere argumenter. Bruk hjelpefilen ?hist
     # for å finne ut hva de betyr:
     main = "Histogram over alder",
     xlab = "Alder",
     ylab = "Frekvens (antall respondenter)",
     col = "lightblue")
```

```
## Warning in plot.histogram(r, freq = freq1, col = col, border = border, angle
= angle, :  the AREAS in the plot are wrong -- rather use 'freq = FALSE'
```



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.

```
  install.packages("stagazer")
```

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

```r
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
## ---------------------------------------------------------------


# Vis hvordan du gjøre dette om til html.
```

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

   Bivariat analyse brukes når man analyserer to variabler. Bivariat analyse er nyttig for å få oversikt over sammenhengen mellom to variabler, i tillegg til at det forteller 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. (En oppgave kan være å forklare en annen/eller seg selv hva statistisk signifkans er).

   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.

```r
  krysstabell <- table(data[, c("internettbruk", "kjonn")])
krysstabell


##              kjonn
## internettbruk   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.

```r
  prop.table(krysstabell, margin = 1)
```

```
##              kjonn
## internettbruk         1         2
##             1 0.3795987 0.6204013
##             2 0.4306220 0.5693780
##             3 0.4867725 0.5132275
##             4 0.5111111 0.4888889
##             5 0.5072254 0.4927746
```

```r
# 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()

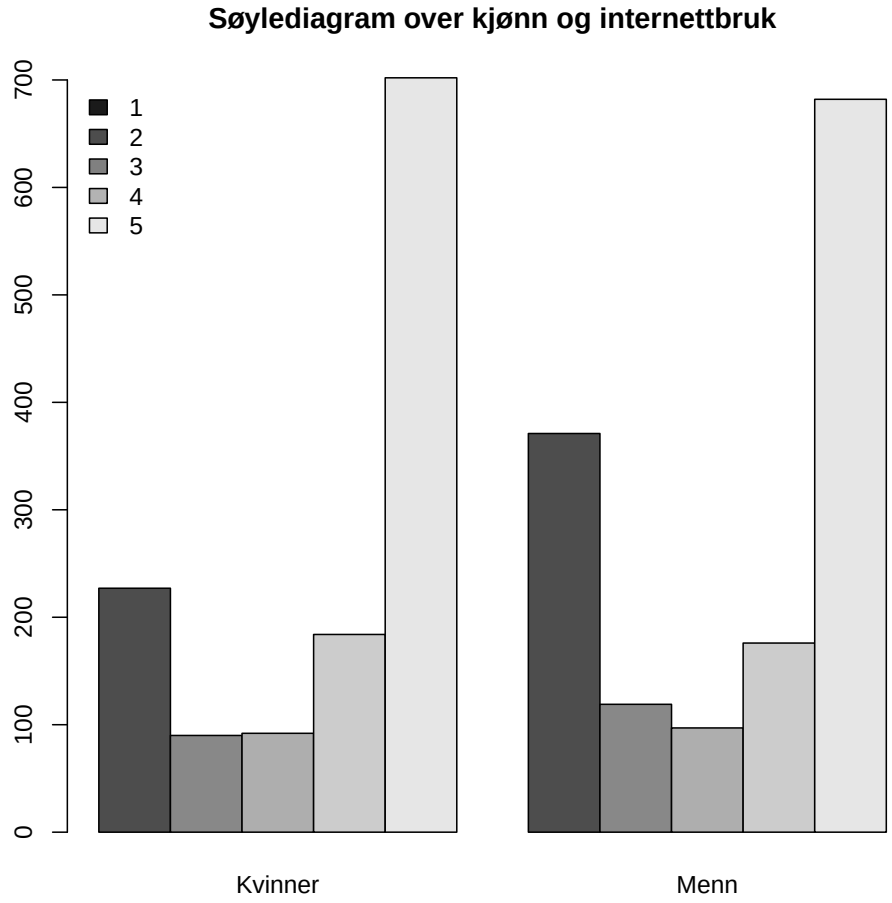
```r
  chisq.test(krysstabell)
```

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

```r
# X-squared, altså kjikvadratet er på 31.18, og
```

Vi kan lage søylediagrammer for å presentere sammenhengen grafisk. Igjen, det er alltid lurt, også for deg selv. Det er mer intuitivt å tolke, og lettere å se sammenhenger raskt. I søylediagrammet legges begge variablene inn, nemlig kjønn og internettbruk, ved bruk av barplot()-funksjonen. Finn ut hva legend() gjør.

```r
  barplot(krysstabell,
          beside = TRUE,
          main = "Søylediagram over kjønn og internettbruk",
          names.arg = c("Kvinner", "Menn"))
legend("topleft",
       bty = "n",
       legend = c("1", "2", "3", "4", "5"),
       fill = c(gray(0.1), gray(0.3), gray(0.5), gray(0.7), gray(0.9)))
```



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.

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

## [1] -0.4090912

# Hva forteller dette oss?
```

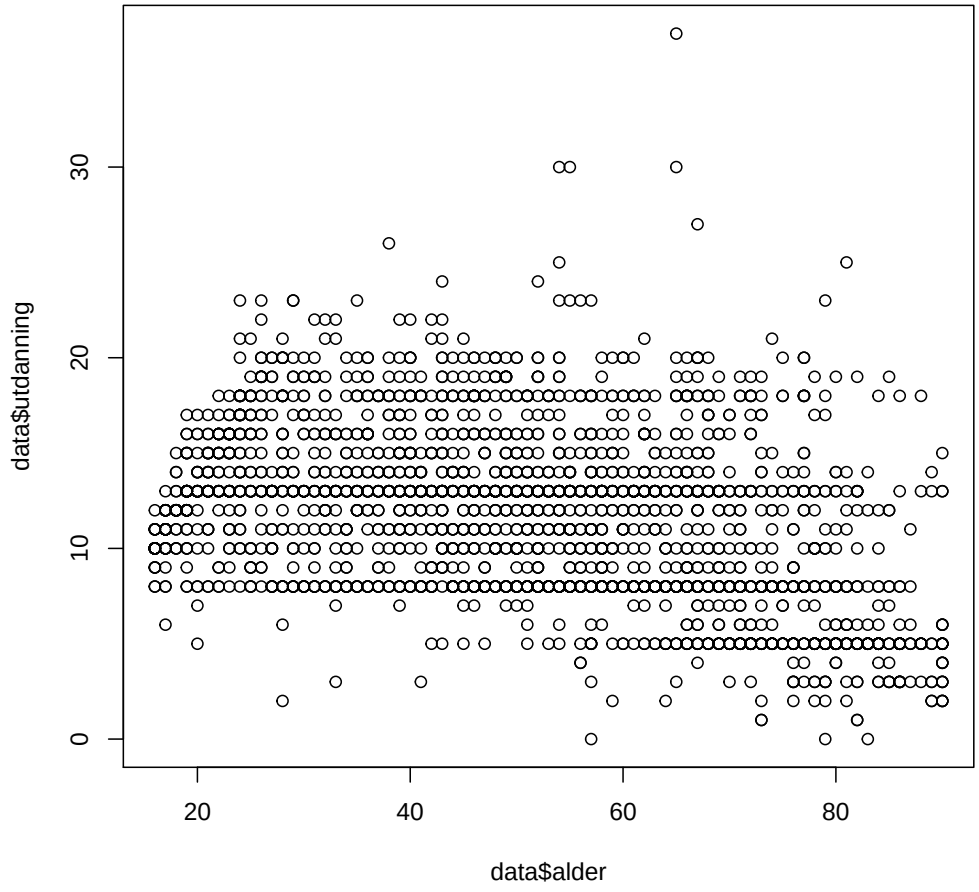
Vi kan også sette opp en korrelasjonsmatrise for å utforske alle de bivariate korrelasjonene i datasettet mellom de akutelle variablene.

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

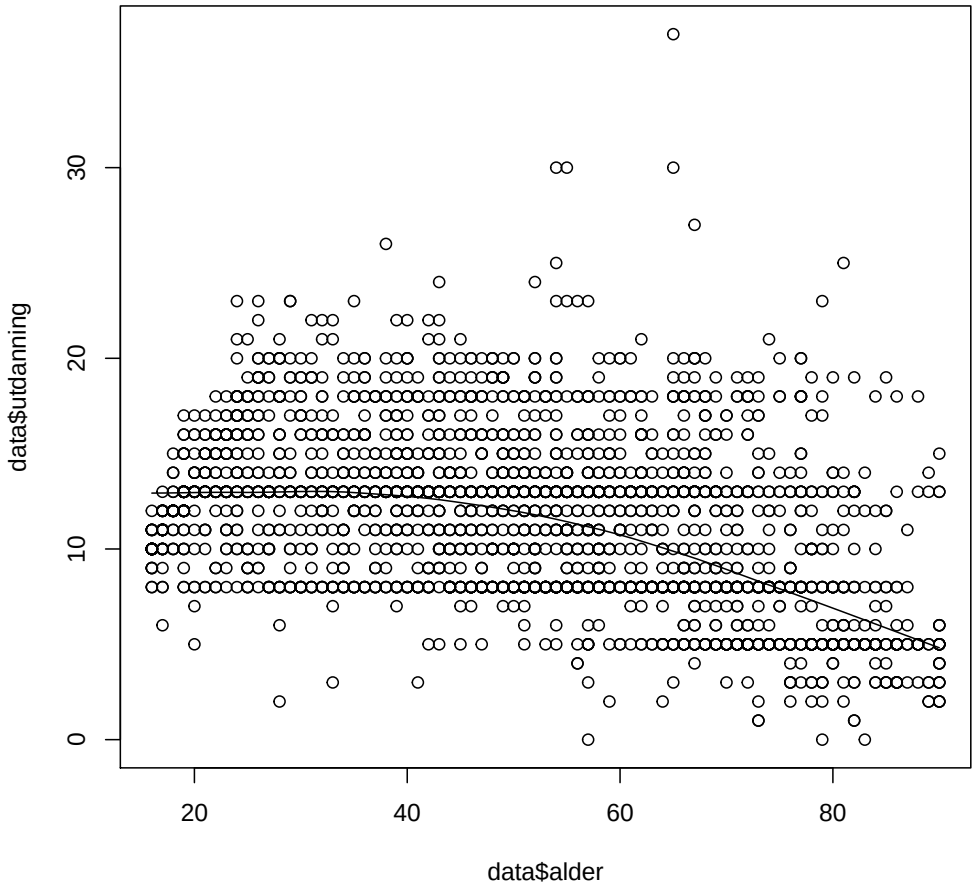
```
##               internettbruk       kjonn       alder  utdanning      tillit
## internettbruk     1.0000000 -0.10206670 -0.64360948  0.5583489  0.15587219
## kjonn            -0.1020667  1.00000000  0.06688781 -0.0528283 -0.04115814
## alder            -0.6436095  0.06688781  1.00000000 -0.4090912 -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 plot()-funksjonen.

```
  plot(x = data$alder, # Uavhengig variabel
       y = data$utdanning) # Avhengig variabel
```



```r
# Hva viser spredningsdiagrammet oss?

# Her legger vi til en støttelinje som viser den gjennomsnittlige
# sammenhengen mellom to variabler:
scatter.smooth(data$utdanning ~ data$alder)
```




```r
# Hva viser dette oss?
```

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