Seminar 2

STV1020 Vår 2021

Plan for seminaret:

1. **Pakker**

   R-pakker er utvidelser til programmeringsspråket R. De inneholder kode, data, og dokumentasjon som gir oss tilgang til funksjoner som løser ulike problemer og gjør koding enklere. Første gang man skal bruke en pakke må man installere den.

   Så må vi "hente den fra biblioteket" for å fortelle R at vi ønsker å bruke pakken, dette må vi gjøre hver gang vi åpner R på nytt og ønsker å bruke pakken.

```
install.packages("tidyverse") # husk hermetegn!

library(tidyverse)
```

2. **Laste inn data**

   Det finnes ulike typer datasett som man kan laste inn i R og man bruker ulike funksjoner for å laste disse inn i R. Noen av funskjonene krever at vi først har installert en pakke.

   For eksempel så er datasettet vi skal bruke i dag en STATA-fil, den kan lastes inn på følgende måte:

```
install.packages("foreign") # husk hermetegn!
library(foreign)

df <- read.dta("ESS9NO.dta")
```

3. **Organisering av arbeidet**

   Overskrifter og tekst

   Hvordan man organiserer et R-script kommer an på hva man selv synes er mest oversiktlig, men det er viktig at man klarer å holde oversikt over hva man har kodet og forstår hva man har gjort når man kommer tilbake til et script.

   F.eks. kan det være lurt å lage overskrifter og underoverskrifter til oppgaver eller elementer av arbeidet, som tydeliggjør hvorfor du har inkludert akkurat denne koden og hva du tenker at den skal gjøre.

   Det kan også være lurt å inkludere #tekst som forklarer hva du gjør og hvorfor.

   Organisering av data

   Når man bruker større datasett som ESS, så inneholder datasettet ofte mange flere variabler enn de vi ønsker å bruke i våre analyser og variablene har navn som kan være vanskelig å huske, f.eks. nwspol.

   Derfor kan det være lurt å fjerne de variablene vi ikke skal bruke og gi variablene navn som er lette for oss å forstå og huske.

   Her endrer vi navnet til variablene til noe mer intuitivt. Vi bruker en pipe (%¿%), som tar outputen til et utsagn og gjør det til inputen til det neste utsagnet. Pipen kan sees på som ordet "så". Rename-funksjonen lar oss forandre navnet til variabler og bruker syntaksen nytt_navn = gammelt_navn.

```
df <- df %>% rename(
    news = nwspol,
    interest = polintr,
    age = yrbrn)
```

   Her velger vi hvilke variabler vi vil ha med oss videre i datasettet. Da det blir mer oversiktlig,ettersom dette er et stort datasett med veldig mange variabler. Select() er en funksjon som lar oss velge variabler i en datasett. I dette tilfellet velger vi variabler ved å refere til navnene deres.

```
df <- df %>% select(news, interest, vote, age)
```

4. **Målenivå**

(a) **Nominalnivå**

Når variabler er på nominalnivå kan egenskapen deles i to eller flere gjensidig utelukkende kategorier.

F.eks. variabelen "vote", man har enten stemt, ikke stemt, eller så er man ikke berettiget til å stemme.

Hvis man har to gjensidig utelukkende kategorier så kan man bruke disse i binomisk regresjonanalyse, så i dette tilfellet kunne man vurdert å fjerne kategorien ikke berettiget til å stemme.

(b) **Ordinalnivå**

Når variabler er på ordinalnivå kan de deles i to eller flere kategorier som kan rangordnes. Så dette at verdiene kan rangordnes er det som skiller ordinalnivå fra nominalnivå.

F.eks. variabelen "interest", man kan være ikke interessert, lite interessert, ganske interessert, eller veldig interessert.

(c) **Intervallnivå**

Når variabler er intervallnivå kan verdiene graderes på en skala med et tilfeldig nullpunkt, der en skalaenhet utgjør like mye av den underliggende egenskapen over hele skalaen.

F.eks. temperatur eller år.

(d) **Forholdstallsnivå**

Når variabler er på forholdstallsnivå kan egenskapen graderes på en skala med et absolutt nullpunkt, der en skalaenhet utgjør like mye av den underliggende egenskapen over hele skalaen. Så forskjellen mellom intervallnivå og forholdstallnivå er at variabler på forholdstall har et definert nullpunkt.

F.eks. variabelen "news", som viser hvor mye tid en respondent bruker på nyheter hver dag, og "age", som viser alderen til respondentene.

5. **Klasser og målenivå**

Variabler på nominalnivå og ordinalnivå vil være av klassen "factor".

```
class(df$interest)
```

Variabler på intervallnivå vil være av klassen "numeric" eller "integer", det samme gjelder forholdstallnivå.

6. **Utforske data**

   Det er mange ulike måter å utforske data på. Vi skal se på funksjonene: summary, str, levels, head, og tail.

   For å få et deskriptivt sammendrag av et objekt kan vi bruke summary-funksjonen.

```
summary(df$vote)
```

   Et alternativ til summmary-funksjonen er str-funksjonen, som viser det interne strukturen til et R-objekt.

```
str(df)
```

   For å undersøke nivåene til en variable kan man bruke levels-funksjonen, outputen man får er nivåene til variablen og verdiene deres.

```
levels(df$interest)
```

   Hvis man vil se de første eller siste radene i et datasett, kan man bruke henholdvis head- og tail-funksjonene. Man kan også velge for eksempel å bare se på de første eller siste verdiene til en bestemt variabel.

```
head(df$interest)
tail(df$interest)
```

7. **Plotting**

   Det er gøy å kunne visualisere dataene våre, både for vår egen del, men også for de som skal lese oppgavene våre. For å få fine grafer kan man bruke pakken ggplot.

   Hvordan kan vi visualisere hvordan fordelingen av politisk interesse er? Her kan vi bruke geom_bar for å lage et histogram.

   Hvor mange innenfor hvert nivå av politisk interesse stemte?

   Hvordan fordeler tiden man bruker på nyheter på alder? Her kan vi bruke geom_point for å lage et spredningsplott.

   Hvordan fordeler alder seg på interesse? Vi kan lage et boksplott med geom_boxplot.

   Hvis dere vil utforske hvordan man kan tilpasse de ulike diagrammene vi har sett på og mange andre, kan denne siden være nyttig: https://www.r-graph-gallery.com/index.html