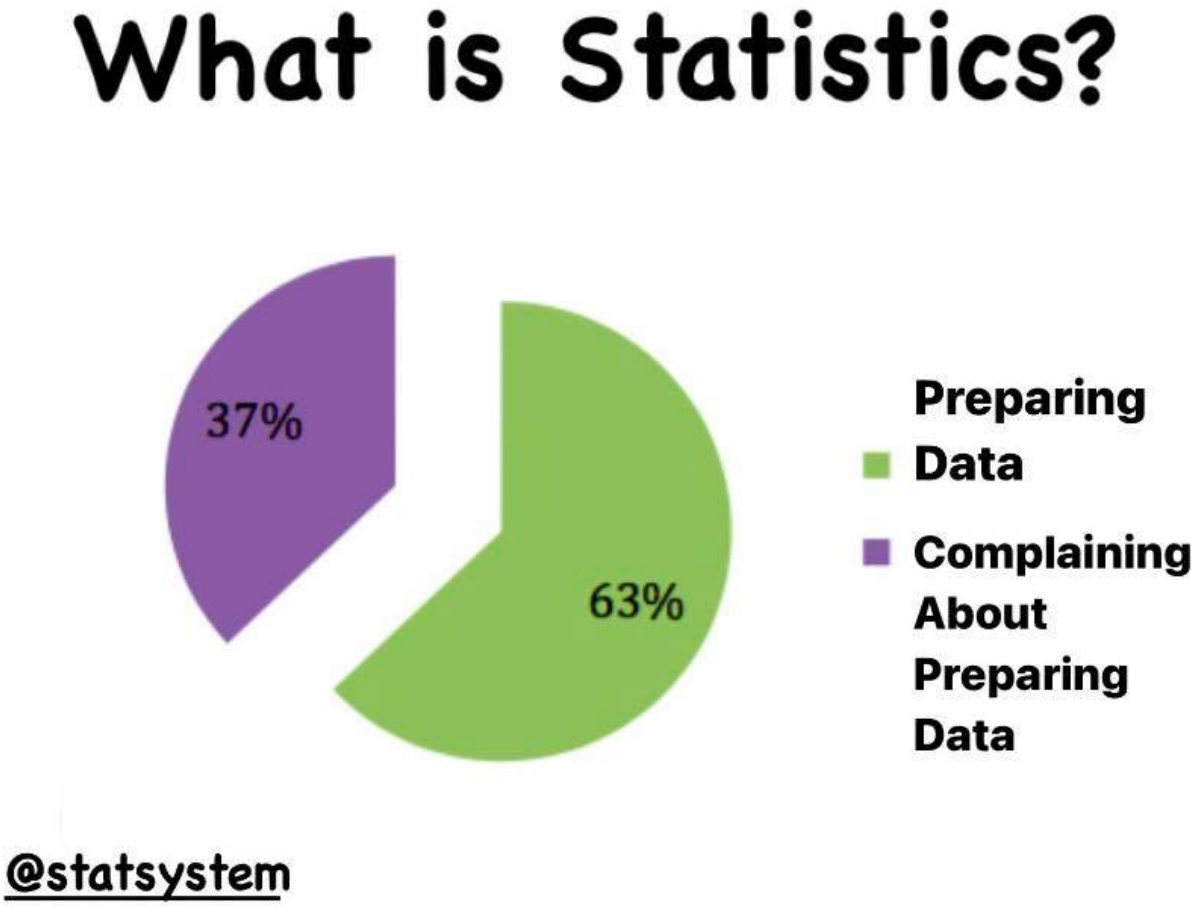
Seminar 2

Eric

31 1 2021

Andre seminar, andre ting å lære!

I dag skal vi lære hvordan å bruke logiske tester, og å få data inn i det formatet vi ønsker. Fra forrige gang kan dere huske at vi lagde våre egne datasett. Som oftest er det jo ikke sånn at vi samler inn data selv, men vi får det fra noen andre. I min master f.eks. bruker jeg data fra World Value Survey. Siden de ikke blir laget på den måten jeg ønsker å bruke dem må jeg bruke en del på forandre formen på dataene. Når vi jobber med statistikk er det som oftest denne delen som faktisk tar tid. Å drive med analyse, kjøre regresjonsmodeller, sjekke modellene våre osv. går ganske fort når dataene er riktige.



Det finnes ganske mange måter å gjøre "data-wrangling" (det fine ordet på å forandre data, google-tip) på. Et stort skille går på om du gjøre det i base-R eller med en pakke. Det finnes ganske mange pakker der ute som eksisterer for å gjøre dette enklere, men i dette seminaret vil vi fokusere på det som kalles tydiverse. Skulle du være interesert i å fortsette med R kan det være fint å se på alternativene, men vil står for at tidyverse er det beste.

Tidyverse er teknisk sett ett sett med pakker laget for å fungere godt sammen. Nøyaktig hva som kommer fra de enkeltstående pakkene er som oftest ikke så veldig interesant, men skulle du trenge det står det alltid øverst i hjelpefilen.

```r
#Før vi skal bruke en pakke må vi innstalere den, med install.packages() funksjonen
install.packages("tidyverse")
```

```r
#Hver gang vi skal bruke den må vi også kjøre library()
library(tidyverse)
```

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

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     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()
```

Nå som vi har pakkene inne, kan vi først se litt på hvordan å laste inn data. Data kommer i ganske mange filformater, og hvordan vi laster dem inn vil være avhengig av typen fil. Noen ganske vanlige, som f.eks. .csv, har R innebygget støtte for. Det finnes også datatyper som .rds som er laga nettopp for R. Filformater laga for andre progammer som f.eks. excel og xlxs må en ha egne pakker for. Når du derfor skal laste inn data må du først se på typen. Er du usikker på hvordan er det som oftest bare å google seg frem til det, det finnes garantert en løsning der ute!

Datatypen vi skal bruke i dette seminaret er .csv, eller "comma seperated values." Dette er rett og slett en tekstfil som inneholder alle dataene våre. For å laste inn dette bruker vi funksjonen read.csv()

```r
# read.csv() funksjonen fungerer sånn at du bruker <- for å lage et objekt, og så i parantesen skriver
#inn linken til filen. Dette kan enten være en fil på din data, f.eks.
#read.csv("dokumenter/r-filer/data.csv")

#Eller som vi gjør her en link til internett.
ESS <- read.csv("https://raw.githubusercontent.com/egen97/4020A_RSeminar/master/ESS_Selected.csv")
```

Om dere ser i enviornment nå vil dere se at vi har fått en data.frame som heter "ESS", og har 434065 observasjoner av 24 variabler. Dette er data fra the European Social Survey, en spørreundersøkelse som går i flere Europeiske land og stiller spørsmål relevant for samfunsvitenskapene. Noe av det første vi bør gjøre er å få en oversikt over hva dataene inneholder. En lett måte å gjøre det på er gjennom str() funksjonen.

```r
str(ESS)
```

```
## 'data.frame':    434065 obs. of  24 variables:
##  $ Time_News                  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Trust_People               : int  7 6 0 8 8 0 5 6 7 4 ...
##  $ People_Fair                : int  7 3 3 5 8 5 9 7 8 3 ...
##  $ Pol_Interest               : int  3 1 2 2 3 1 3 2 3 3 ...
##  $ Trust_Police               : int  10 5 8 9 4 6 6 7 8 5 ...
##  $ Trust_Politicians          : int  0 0 2 4 4 0 5 4 3 5 ...
##  $ vote                       : int  2 1 1 1 1 2 1 1 1 1 ...
##  $ Party_Voted_NO             : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Left_Right                 : int  6 6 5 5 5 NA NA 6 5 5 ...
##  $ Satisfied_Gov              : int  7 0 7 3 5 0 5 5 3 5 ...
##  $ Gov_Reduce_IncomDif        : int  2 1 2 4 4 1 NA 4 4 1 ...
##  $ LGBT_Free                  : int  1 1 1 3 2 1 NA 2 1 1 ...
##  $ Religous                   : int  8 5 7 7 10 3 8 1 6 5 ...
##  $ Climate_Human              : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Responsibility_Climate     : int  NA NA NA NA NA NA NA NA NA NA ...
```

```
##  $ Goverment_Climate            : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Basic_Income                 : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Important_Rules              : int  1 6 2 3 6 5 4 5 2 1 ...
##  $ Important_Equal_Oppurtunities: int  1 1 2 2 1 1 2 1 2 1 ...
##  $ Income                       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Gender                       : int  1 1 2 1 2 2 2 2 1 2 ...
##  $ Age                          : int  54 50 63 44 41 63 75 41 47 52 ...
##  $ Country                      : chr  "AT" "AT" "AT" "AT" ...
##  $ essround                     : int  1 1 1 1 1 1 1 1 1 1 ...
```