Bygge statistiske modeller med kategoriske prediktorvariabler (t-tester, ANOVA)

Christian Magelssen

2021-03-29

Contents

Chapter 1

Introduksjon

I dette kapittelet skal vi lære å bygge statistiske modeller for å teste om **to eller flere grupper er forskjellige på en avhengig variabel som er kontinuerlig**.

Husk tilbake til forelesninger nr. 2 der vi sa at en variabel kan sies å være **kontinuerlig** når vi kan bestemme hvor presist vi ønsker å måle den. For eksempel regnes tid som en kontuerlig variabel fordi det (i prinsippet) ikke finnes noen grenser hvor presist vi kan måle det; vi kan måle det i år, måneder, uker, dager, timer, minutter, sekunder, tideler, hundredeler eller tusendeler.

Grupper defineres i psykologifaget som en samling mennesker som deler bestemte karakterstikker. Det kan være spillere på et fotballag, individer på et treningssenter, eller menn og kvinner. Dette er også eksempler på naturlig inndelte grupper i samfunnet. Noen ganger kan det være interessant å se om disse gruppene er forskjellige. For eksempel kan det være interessant å se om individer som trener på treningssenter er sterkere enn de som ikke trener på treningssenter. Men i eksperimenter er vi som oftest interessert i om to grupper, som var like før behandling, har blitt forskjellige etter behandling. Vi randomiserer individer i to ulike grupper, slik at vi sikrer at vi blander disse individene godt (f.eks kjønn, motivasjon, interesser). Hvis eksperimentet har blitt gjennomført godt at det ikke er noen andre forklaringer på at disse to gruppene har blitt forskjellige etter intervensjonsperioden, så kan vi trekke en slutning om disse to gruppene trolig ikke kommer fra samme populasjon lenger; eksperimentet har gjort at disse to gruppene trolig kommer fra to forskjellige populasjoner.

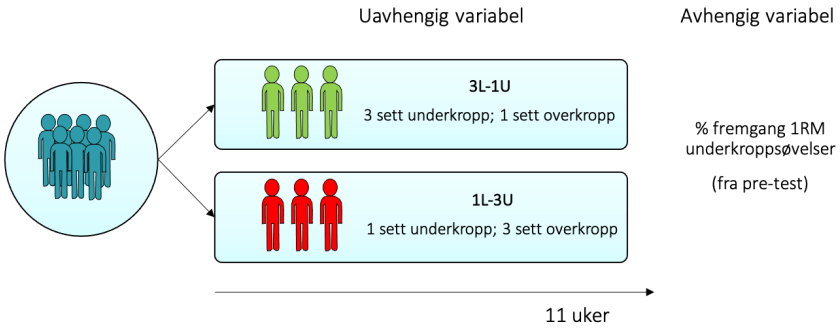
Chapter 2

Datasett

2.0.1 Bør man trene med ett eller flere sett i styrketrening?

Et spørsmål mange treningsentusiaster lurer på er hvor mange serier som er best å gjennnomføre for å få maksimal treningseffekt i styrketrening. Noen mener at ett sett er tilstrekkelig, mens andre mener at et hardere treningstimuli er nødvendig og at to eller flere sett derfor er bedre. En forsker som var tidlig ute med å undersøke dette er deres egen Bent Rønnestad.

Eksperimentet ble gjennomført som et **between-subject design** med to grupper: en gruppe trente 1 sett på underkroppen og 3 sett på overkroppen; En annen gruppe trente 3 sett på underkroppen og 1 sett på overkroppen. Disse gruppene kalte han henholdsvis **1L-3U** og **3L-1U** (L=lower; U=Upper). De to gruppene trente 3 ganger i uken i totalt 11 uker. Forskerne ville så se hva som ga best fremgang på 1RM. Den avhengige variabelen ble derfor %-fremgang på 1RM på underkroppsøvelser. De fant at 3L-1U hadde større fremgang enn 1L-3U fra pre til post (41 vs 21 % endring). Denne forskjellen var signifikant ved en uavhengig t-test. Med andre ord kan det se ut til at det kan lønne seg å trene flere sett per styrketreningsøkt.



Vi har ikke tilgang til dette datasettet, men jeg har simulert dette datasettet i R basert på verdiene jeg fant i artikkelen. Datasettet blir tilnærmet likt, men siden det er en simulering blir det aldri helt identisk. Datasettet ser du i tabellen under.

```
knitr::kable(
  dat, booktabs = TRUE,
  caption = 'Simulert datasett'
)
```

Du kan få nøyaktig samme datsett ved å klippe ut og lime inn følgende kode i en skript-fil i R (husk å laste inn tidyverse-pakken, library(tidyverse)). Du kan også laste ned datasettet som en .csv fil fra canvas.

```
set.seed(2002) #viktig å ha med denne for å få nøyaktig samme datasett
tre.sett <- rnorm(n = 12, mean = 41, sd = 5) #12 individer
ett.sett <-rnorm(n = 12, mean = 21, sd = 5) #12 individer

#lager en tibble fra tidyverse-pakken. Må ha lastet inn tidyverse library(tidyverse) i
dat <- tibble(individ = seq(1:24),
              gruppe = rep(c("tre.sett ", "ett.sett"), c(length(tre.sett), length(ett.s
              rm = c(tre.sett , ett.sett))
```

Før du går videre er det greit at du gjør deg kjent med datasettet som vi har generert. Studer datasettet og svar på følgende spørsmål:

1. Hvor mange kolonner er det i tabellen over?
2. Hvor mange deltakere var med i studien?
3. Hvilke to verdier kan variabelen gruppe? og

2.0.2 Regne gjennomsnitt for de to gruppene

Bra! Det er alltid viktig å bli kjent med sitt eget datasett, men nå som du har det kan vi gå videre. Vi er interessert i om det er forskjeller mellom de

Table 2.1: Simulert datasett

individ	gruppe	rm
1	tre.sett	40.46704
2	tre.sett	49.07223
3	tre.sett	47.94131
4	tre.sett	44.51389
5	tre.sett	52.28750
6	tre.sett	40.01750
7	tre.sett	49.48425
8	tre.sett	29.21048
9	tre.sett	40.59293
10	tre.sett	37.58676
11	tre.sett	35.42651
12	tre.sett	42.49354
13	ett.sett	17.70576
14	ett.sett	17.07181
15	ett.sett	18.26811
16	ett.sett	25.42594
17	ett.sett	32.70313
18	ett.sett	19.10226
19	ett.sett	22.23827
20	ett.sett	22.27148
21	ett.sett	26.17889
22	ett.sett	20.34857
23	ett.sett	23.52773
24	ett.sett	17.95966

to gruppene ("tre.sett" vs. ett.sett) på % fremgang fra pre- til post-test. Så kanskje vi kan starte med å se om det er forskjeller i gjennomsnitt mellom to gruppene? Dette kan enkelt gjøres i R, Jamovi eller excel. Her er en kode for å gjøre dette i R:

```
#jeg lager et oobjekt som heter mean_rm
mean_rm <- dat %>%
  #Jeg grupperer etter gruppe, slik at jeg får et mean for hver gruppe istf. for å få
  #group_by er en funksjon for dette
  group_by(gruppe) %>%
  #deretter bruker jeg summarise funksjonen for å regne gjennomsnitt
  summarise(mean.fremgang.1RM = mean(rm))
```

Koden gir oss følgende tabell: \begin{table}

\caption{Gjennomsnittlige %-vis fremgang for de to gruppene}

gruppe	mean.fremgang.1RM
ett.sett	21.90013
tre.sett	42.42450

\end{table}

Hvilken gruppe hadde mest fremgang? ett.sett tre.sett‘

2.0.3 Figur av datasettet

Vi kan også presentere dataen i en figur. For denne typen data er det veldig vanlig å bruke et **stolpediagram**:

Et stolpediagram er pent å se på, men er egentlig designet for å kategoriske data. For eksempel er det fint å bruke dette når vi skal presentere frekvensen antall som har kjørt bil til skolen og antall personer som har gått. Les Beyond Bar and Line Graphs: Time for a New Data Presentation Paradigm. Deretter svar på følgende spørsmål for å se om du har forstått problemene ved å bruke stolpediagram på kontinuerlig data.

"Stolpediagram er designet for kontinuerlig kategorisk data. Høyden på stolpen representerer (bruk det norske begrepet!), hvilket vil si at det også må ligge noen observasjoner over og under stolpen. Man blir dermed ikke lurt lurt ved å bruke et stolpediagram". Et stolpediagram viser heller ikke standard error standardavvik CI fordelingen av observasjonene, og dette spesielt være problematisk ved store små. Derfor anbefaler forfatterne i artikkelen at man viser dataen mer ved å for eksempel bruke et bar graph scatterplot. Hvis man likevel ønsker å bruke et stolpediagram for å presentere dataen er det viktig at man forteller om man har brukt SE, SD eller CI. Stanard error for

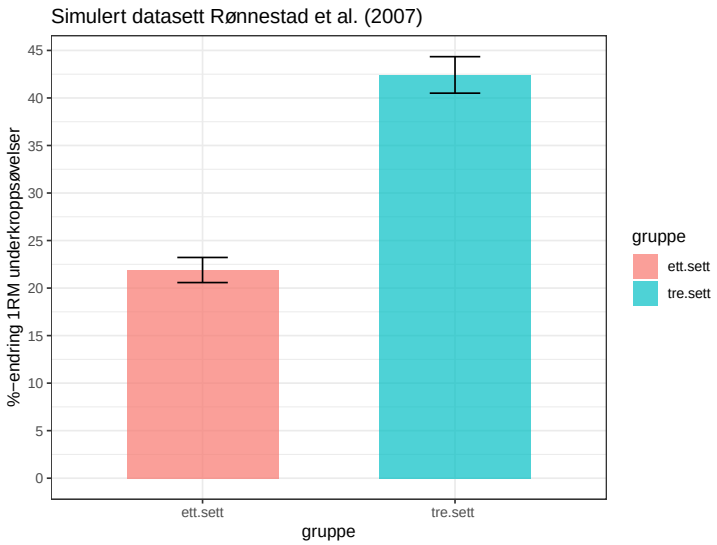


Figure 2.1: Here is a nice figure!

gjennomsnittet regnes ved å ta $SD/sqrt(N)$, så ved store utvalg vil standard error være høyt lite. Standardavviket er kun $sqrt(varians/n-1)$, så denne vil i større mindre grad være påvirket av utvalgsstørrelsen".

Chapter 3

Koding av kategoriske variabler

I tabellen på s. kan du se at vi har en tabell med tre kolonner: en kolonne for hver variabel vi har i vårt datasett. Variabelen **gruppe** er en kategorisk vaiabel som har to ulike verdier: "ett.sett" og "tre.sett". Dette er de to gruppene som vi skal teste om er forskjellige. I programmeringsverdenen kalles disse denne typen data for et tekstobjekt, "strings" (python/javascript) eller "characters" (R). På norsk kalles disse verdiene for ord. Uansett navn er problemet at vi ikke kan putte ord inn i en statistisk modell; vi er nødt til å representere denne kategoriske vaiabelen med tallverdier. Det er flere måter å gjøre dette på, men de forskjellige måtene gir ulik resultat. Derfor må vi vie en god del tid på dette. Vi går gjennom to måter å gjøre dette på.

3.1 Dummykoding

En vanlig metode kalles **dummykoding** eller **treatment-koding**. Den går ut på å lage en eller flere variabler med 0 og 1 som de to mulige verdiene. Antall variabler vi trenger avhenger av antall grupper vi vil sammenligne. Siden vårt datasett kun inneholder to grupper, så trenger vi kun en variabel. Vi kan den ene gruppen og den andre 1. Hovedregelen er at vi gir 0 til baselinegruppe og 1 til den eksperimentelle gruppen. Vi gir derfor 0 til 1.sett-gruppen og 1 til 3.sett-gruppen. Gjør dette før du går videre.

I R og Jamovi kan du gjøre det med følgende if/else statement. I R kan du bruke følgende kode:

```
#lager et nytt objekt som heter dummykodet.dat
dummykodet.dat <- dat %>%
```

Table 3.1: Dummy koding

individ	gruppe	rm	dummykodet
1	tre.sett	40.467	1
2	tre.sett	49.072	1
3	tre.sett	47.941	1
4	tre.sett	44.514	1
5	tre.sett	52.288	1
6	tre.sett	40.018	1
7	tre.sett	49.484	1
8	tre.sett	29.210	1
9	tre.sett	40.593	1
10	tre.sett	37.587	1
11	tre.sett	35.427	1
12	tre.sett	42.494	1
13	ett.sett	17.706	0
14	ett.sett	17.072	0
15	ett.sett	18.268	0
16	ett.sett	25.426	0
17	ett.sett	32.703	0
18	ett.sett	19.102	0
19	ett.sett	22.238	0
20	ett.sett	22.271	0
21	ett.sett	26.179	0
22	ett.sett	20.349	0
23	ett.sett	23.528	0
24	ett.sett	17.960	0

```
# her lager jeg en ny kolonne som heter dummykoder. If gruppe == 'ett.sett', gi verd
mutate(dummykodet = if_else(gruppe == "ett.sett", 0, 1))
```

I jamovi ville jeg sett følgende video:
https://www.youtube.com/watch?v=iITxK27LfZk

3.2 Kontrastkoding

Kontrastkoding er et alternativ til dummykoding. Det er en regel som er viktig å følge for å ha en gyldig kontrastkode, og det er at summen av kontrastkodene blir 0. For eksempel er -0.5 og 0.5 gyldige kontrastkoder fordi summen av disse blir 0. Det samme er -10 og +10. 0 og 1 derimot, slik vi har med en dummykodet variabel, er ikke er en gyldig kontrastkode fordi summen av disse blir 1. **Hvilke verdier vi velger å bruke på vår kontrastkodede**

Table 3.2: Kontrastkoding

individ	gruppe	rm	dummykodet	kontrastkodet
1	tre.sett	40.467	1	0.5
2	tre.sett	49.072	1	0.5
3	tre.sett	47.941	1	0.5
4	tre.sett	44.514	1	0.5
5	tre.sett	52.288	1	0.5
6	tre.sett	40.018	1	0.5
7	tre.sett	49.484	1	0.5
8	tre.sett	29.210	1	0.5
9	tre.sett	40.593	1	0.5
10	tre.sett	37.587	1	0.5
11	tre.sett	35.427	1	0.5
12	tre.sett	42.494	1	0.5
13	ett.sett	17.706	0	-0.5
14	ett.sett	17.072	0	-0.5
15	ett.sett	18.268	0	-0.5
16	ett.sett	25.426	0	-0.5
17	ett.sett	32.703	0	-0.5
18	ett.sett	19.102	0	-0.5
19	ett.sett	22.238	0	-0.5
20	ett.sett	22.271	0	-0.5
21	ett.sett	26.179	0	-0.5
22	ett.sett	20.349	0	-0.5
23	ett.sett	23.528	0	-0.5
24	ett.sett	17.960	0	-0.5

variabel betyr ingenting for den statistiske test vi gjennomfører, men gjør at vi må fortolke resultatene litt forskjellig. Med en kontrastkode på +10 og -10 er det en 20 enhets forskjell, mens det ved +0.5 og -0.5 kun er enhet forskjell.

```
#lager et nytt objekt som heter dummykodet.dat
kontrastkodet.dat <- dummykodet.dat %>%
  # her lager jeg en ny kolonne som heter kontrastkodet. If gruppe == 'ett.sett', gi verdien -0.5
  mutate(kontrastkodet = if_else(gruppe == "ett.sett", -0.5, +0.5)
         )
```

Spørsmålet dere sikkert lurer på er hvorfor vi dummykoder og kontrastkoder gruppe-variabelen vår. Det korte svaret er at vo gjør det fordi vi skal se at disse to måtene å kode på produserer forskjellige svar.

Chapter 4

Hvilken modell skal vi velge?

Her tenker jeg å si litt om at vi har ulike muligheter.

4.1 Null-Hypothesis Significance Testing (NHST)

I forelesningene har vi snakket en god del om å teste hypoteser. Paradigmet vi bruker kalles **Null-Hypothesis Significance Testing (NHST)**, og går ut på at forskeren fremstiller to hypoteser:

1. **H0**: En null-hypotese som sier at det ikke er noen effekt (f.eks. ingen forskjeller mellom grupper, ingen sammenheng mellom variablene)
2. **H1**: En alternativ/eksperimentell hypotese som sier at det er en effekt (f.eks. det er en forskjell mellom gruppene)

Forskeren gjennomfører deretter en **statistisk test** som representerer den alternative hypotesen. Utfallet av testen er en **verdi**, for eksempel en *z-verdi*, *t-verdi* eller *f-verdi*, som vi kan bruke til å regne ut sannsynligheten for, gitt at null-hypotesen er sann. Forskjellige tester opererer med forskjellige navn på verdiene sine (sorry, men det er bare slik det er).

4.2 H0: Null-hypotese

Som dere allerede vet dreier vitenskapen seg om å bygge statistiske modeller. Vi har lært at vi kan en bruke en lineær ligning til alle våre statistiske

oppgaver:

$$data = (modell) + error$$

Data er den avhengige variabelen som vi har målt for alle deltakerne og som vi kan bruke en modell til å predikere. En **modell** er egentlig bare en representasjon av denne dataen, og tre enkle modeller som dere har lært er: **mean**, **median** og **mode**. Et fellestrekk for disse modellene er at de sier noe om den typiske skåren i datasettet. Mode sier hvilken skår som hadde flest observasjoner, medianen sier noe om den midterste verdien i datasettet når observasjonene er rangert fra høy til lav, mens gjennomsnittet…ja, den sier hva som var gjennomsnittet. I de aller fleste tilfeller er **mean** den modellen vi ønsker å benytte (en lang redegjørelse hvorfor kan jeg ikke gi her, men se tidligere forelesninger).

I vår studie ønsker vi å teste om det er forskjeller mellom de to gruppene som har blitt disponert for ulikt treningsopplegg (3 versus 1 sett). Det første vi må spørre oss er om vi virkelig trenger å vite noe om hvordan disse deltakerne har trent eller om det er nok å bare bruke **mean** som modell.

$$fremgang.1RM = (mean) + error$$

Det blir enklere å se hva vi mener med **mean som modell** når vi setter dette opp i en tabell, slik om i tabellen under. På denne måten ser du at modellen sier at alle deltakerne hadde samme fremgang i 1RM (som var gjennomsnittet).

Mean som modell

individ

gruppe

rm

modell.mean

error

1

tre.sett

40.467

32.162

8.305

2

tre.sett

49.072

32.162

16.910

3

tre.sett

47.941

32.162

15.779

4

tre.sett

44.514

32.162

12.352

5

tre.sett

52.288

32.162

20.125

6

tre.sett

40.018

32.162

7.855

7

tre.sett

49.484

32.162

17.322

8

tre.sett

29.210

32.162

-2.952

9

tre.sett

40.593

32.162

8.431

10

tre.sett

37.587

32.162

5.424

11

tre.sett

35.427

32.162

3.264

12

tre.sett

42.494

32.162

10.331

13

ett.sett

17.706

32.162

-14.457

14

ett.sett

17.072

32.162

-15.091

15

ett.sett

18.268

32.162

-13.894

16

ett.sett

25.426

32.162

-6.736

17

ett.sett

32.703

32.162

0.541

18

ett.sett

19.102

32.162

-13.060

19

ett.sett

22.238

32.162

-9.924

20

ett.sett

22.271

32.162

-9.891

21

ett.sett

26.179

32.162

-5.983

22

ett.sett

20.349

32.162

-11.814

23

ett.sett

23.528

32.162

-8.635

24

ett.sett

17.960

32.162

-14.203

La oss prøve et eksempel for å se hvordan modellen virker. For individ 1 målte vi en fremgang i 1RM underkropp på **40.467**, men modellen vår sa **32.162**. Så modellen hadde en error på **8.305**.

$$fremgang.1RM = (mean) + error$$

$$40.467 = 32.162 + 8.305$$

Prøv modellen du også: For individ nr. 8, sier modellen at individet hadde en skår på , men denne personen hadde faktisk en skår på . Modellen bommet derfor med .

Vi kan fortsette slik for alle deltakerne vi har hatt med i studien, men husk fra tidligere forelesninger at vi egentlig ikke er interessert i hvor mye modellen bommer for hvert enkelt individ totalt, men totalt for hvert enkelt individ. Vi kan summere all erroren, men gjør vi dette får vi null 0 3 -3. Dette er fordi

modellen både underestimerer og overestimerer tar feil, så vi ender opp med både negative og positive tall. Summerer vi disse får vi 0.Vi løser dette problemet effektivt ved å regne **Squared error (error^2)**. Hvis vi summerer error^2 får vi **Sum of Squared Error**.

Vi kan regne ut hvor mye **Sum of Squared Error** vi får hvis vi bruker mean som modell. Da får vi . Dette tallet er viktig! Det representerer hvor mye error det i en modell der vi bruker mean som modell. Dette er null-hypotesen vår. Hvis det ikke er noen forskjell mellom gruppene er det like gret å bare å bare bruke mean som modell. Men hvis vi finner ut at modellen vår blir bedre (dvs. reduserer Sum of Squared Error) ved å legge til en prediktorvariabel som består er av gruppevariabelen vår, da bør vi gjøre dette. Hvordan vi gjør dette blir temaet i neste avsnitt.

Før du går videre er det greit å visualisere hvordan null-hypotesen ser ut rent visuelt. Den rød streken i figuren under representerer modellen vår som er mean. Som du ser, så gjør den ingen justeringer for de ulike individene.

```
ggplot(dat, aes(individ, rm)) +
  geom_point(size=2.2, color="#00BFC4") +
  geom_hline(yintercept = 32, color="#f8766D", size=1.2) +
  scale_x_continuous(breaks = seq(1, 25)) +
  scale_y_continuous(breaks = seq(0, 60, 5)) +
  labs(x="deltaker", y="% fremgang 1RM") +
  theme_bw()
```

4.3 H1: Alternativ hypotese

I forrige avsnitt definerte vi **null-hypotesen (H0)** som en modell som gir samme prediksjon for hver enkelt deltaker uavhengig av hvilken treningsgruppe de tilhører. Og vi regnet ut hvor mye Sum of Squared Error sam var i denne modellen (som var 3243.784). Det vi skal spørre om nå er om det et poeng på å bruke en mer kompleks modell der vi legger til en prediktorvariabel.

$$data = (modell) + error$$

Modell 1

$$Y_i = b0 + error_i$$

Modell h1

$$Y_i = b0 + b1 + error_i$$

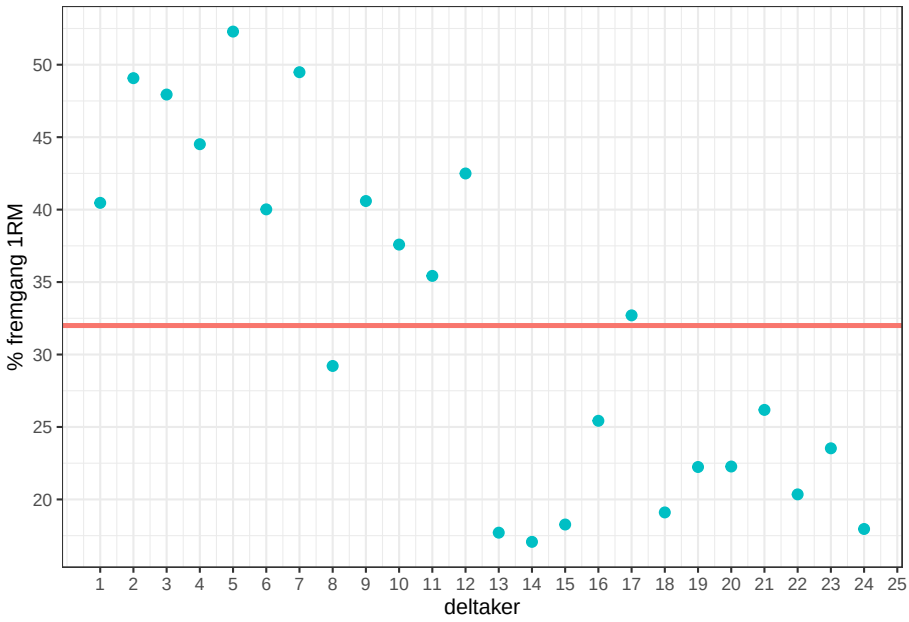


Figure 4.1: **CAPTION THIS FIGURE!!**

Så det vi skal finne ut av er om mod

$$Yi = b0 + error$$

Det vi skal gjøre nå er å spørre oss nå er om det er behov for å legge til en ekstra prediktorvariabel i modellen eller om det er nok å beholde den ene prediktorvariabelen? Så det vi skal spørre oss om nå er

$$= mean + error$$

En annen måte å si dette på er om det er forskjeller mellom de to gruppene eller om det ikke er det. Hvis det ikke er noen forskjeller mellom gruppene, så vil b1 være liten, og vil egentlig bare ende opp med samme prediksjon uansett hvilken gruppe de har tilhørt. Jeg har nå regnet ut hva verdiene i modellen er, så vi få et inntrykk av hvordan modellen virker før vi finner ut hvordan vi regner disse verdiene. Når jeg har brukt dummykoding har jeg fått følgende verdier i min modell:

Husk at gruppen vår er dummykodet som 0 og 1. Så hvis jeg har å gjøre med et individ som tilhører gruppe 0, så blir resultatet følgende

20.52*0 = 0, så vår prediksjon av et individ som tihører gruppe 0 blir da kun 21.90. Hvis vi har en person tilhørte gruppe 1, så blir vår prediksjon

Som blir 42.42. Hva er spesielt med disse verdiene? Gå tilbake til.

```
library(tidyverse)
```

```
set.seed(2002) #viktig å ha med denne for å få nøyaktig samme datasett
tre.sett <- rnorm(n = 12, mean = 41, sd = 5) #12 individer
ett.sett <-rnorm(n = 12, mean = 21, sd = 5) #12 individer

#lager en tibble fra tidyverse-pakken. Må ha lastet inn tidyverse library(tidyverse) i scriptfile
dat <- tibble(individ = seq(1:24),
              gruppe = rep(c("tre.sett ", "ett.sett"), c(length(tre.sett), length(ett.sett))),
              rm = c(tre.sett , ett.sett))


#lager et nytt objekt som heter dummykodet.dat
dat <- dat %>%
  # her lager jeg en ny kolonne som heter dummykoder. If gruppe == 'ett.sett', gi verdien 0, else
  mutate(dummykodet = if_else(gruppe == "ett.sett", 0, 1)
  )


dat <- dat %>%
  # her lager jeg en ny kolonne som heter dummykoder. If gruppe == 'ett.sett', gi verdien 0, else
  mutate(kontrast = if_else(gruppe == "ett.sett", -0.5, 0.5)
  )
```

```
dat <- dat %>%
  mutate(dummykodet = as.factor(dummykodet),
         kontrast = as.factor(kontrast))
```

```
lm(rm ~ dummykodet, dat)
```

```
##
## Call:
## lm(formula = rm ~ dummykodet, data = dat)
##
## Coefficients:
## (Intercept)  dummykodet1
##       21.90        20.52
```

```
library(tidyverse)



ggplot(dat, aes(y=rm, x=individ)) +
  geom_point(aes(colour=gruppe)) +
  geom_smooth(method="lm", formula=y~x) +

  scale_x_discrete()
```

Her fikk jeg en intercept på 21.90 og en slope på 20.52. La oss prøve å få disse to til.

$$Y = b0 + b1 + error$$

$$Y = b0 + (b1 * 0) + error$$

$$Y = b0 + (b1*1) + error \ \$\$$$

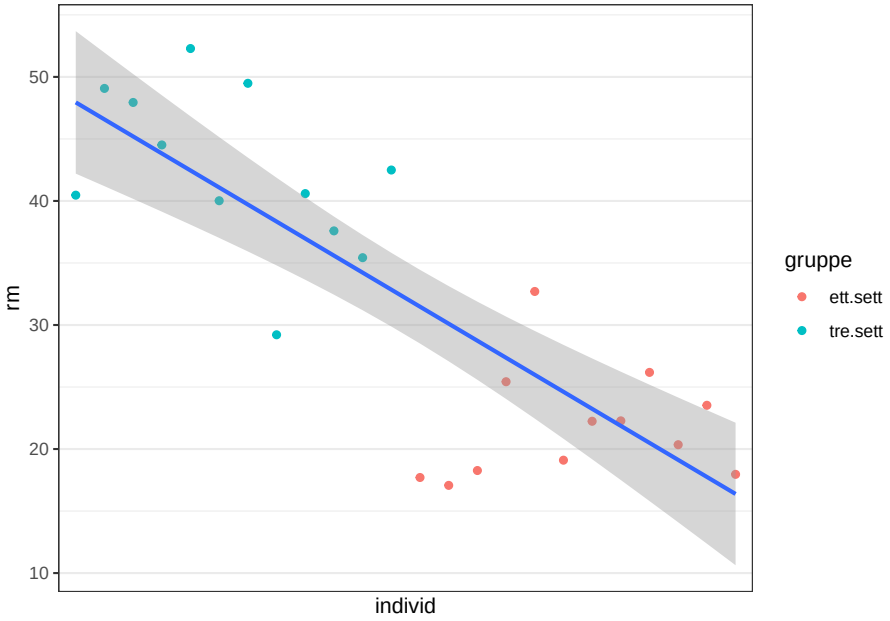


Figure 4.2: **CAPTION THIS FIGURE!!**

Chapter 5

H1: Alternativ hypotese

I forrige avsnitt redegjorde vi for at **null-hypotesen (H0)** er en ligning som består av kun en, som var gjennomsnittet for alle deltakerne i studien (uavhengig gruppe).

$$data = (modell) + error$$

$$Yi = b0 + error$$

$$= mean + error$$

$$1RM = mean + error$$

Det vi skal gjøre nå er å spørre oss nå er om det er behov for å legge til en ekstra prediktorvariabel i modellen eller om det er nok å beholde den ene prediktorvariabelen? Så det vi skal spørre oss om nå er

$$= mean + error$$

En annen måte å si dette på er om det er forskjeller mellom de to gruppene eller om det ikke er det. Hvis det ikke er noen forskjeller mellom gruppene, så vil b1 være liten, og vil egentlig bare ende opp med samme prediksjon uansett hvilken gruppe de har tilhørt. Jeg har nå regnet ut hva verdiene i modellen er, så vi få et inntrykk av hvordan modellen virker før vi finner ut hvordan vi regner disse verdiene. Når jeg har brukt dummykoding har jeg fått følgende verdier i min modell:

Husk at gruppen vår er dummykodet som 0 og 1. Så hvis jeg har å gjøre med et individ som tilhører gruppe 0, så blir resultatet følgende

20.52*0 = 0, så vår prediksjon av et individ som tihører gruppe 0 blir da kun 21.90. Hvis vi har en person tilhørte gruppe 1, så blir vår prediksjon

Som blir 42.42. Hva er spesielt med disse verdiene? Gå tilbake til.

```
library(tidyverse)
```

```
set.seed(2002) #viktig å ha med denne for å få nøyaktig samme datasett
tre.sett <- rnorm(n = 12, mean = 41, sd = 5) #12 individer
ett.sett <-rnorm(n = 12, mean = 21, sd = 5) #12 individer

#lager en tibble fra tidyverse-pakken. Må ha lastet inn tidyverse library(tidyverse) i
dat <- tibble(individ = seq(1:24),
              gruppe = rep(c("tre.sett ", "ett.sett"), c(length(tre.sett), length(ett.s
              rm = c(tre.sett , ett.sett))


#lager et nytt objekt som heter dummykodet.dat
dat <- dat %>%
  # her lager jeg en ny kolonne som heter dummykoder. If gruppe == 'ett.sett', gi verd
  mutate(dummykodet = if_else(gruppe == "ett.sett", 0, 1)
  )


dat <- dat %>%
  # her lager jeg en ny kolonne som heter dummykoder. If gruppe == 'ett.sett', gi verd
  mutate(kontrast = if_else(gruppe == "ett.sett", -0.5, 0.5)
  )
```

```
dat <- dat %>%
  mutate(dummykodet = as.factor(dummykodet),
         kontrast = as.factor(kontrast))
```

```
lm(rm ~ dummykodet, dat)
```

```
##
## Call:
## lm(formula = rm ~ dummykodet, data = dat)
##
## Coefficients:
## (Intercept)  dummykodet1
##       21.90        20.52
```

```
library(tidyverse)



ggplot(dat, aes(y=rm, x=individ)) +
  geom_point(aes(colour=gruppe)) +
  geom_smooth(method="lm", formula=y~x) +

  scale_x_discrete()
```

Her fikk jeg en intercept på 21.90 og en slope på 20.52. La oss prøve å få disse to til.

$$Y = b0 + b1 + error$$

$$Y = b0 + (b1 * 0) + error$$

$$Y = b0 + (b1*1) + error \ \$\$$$

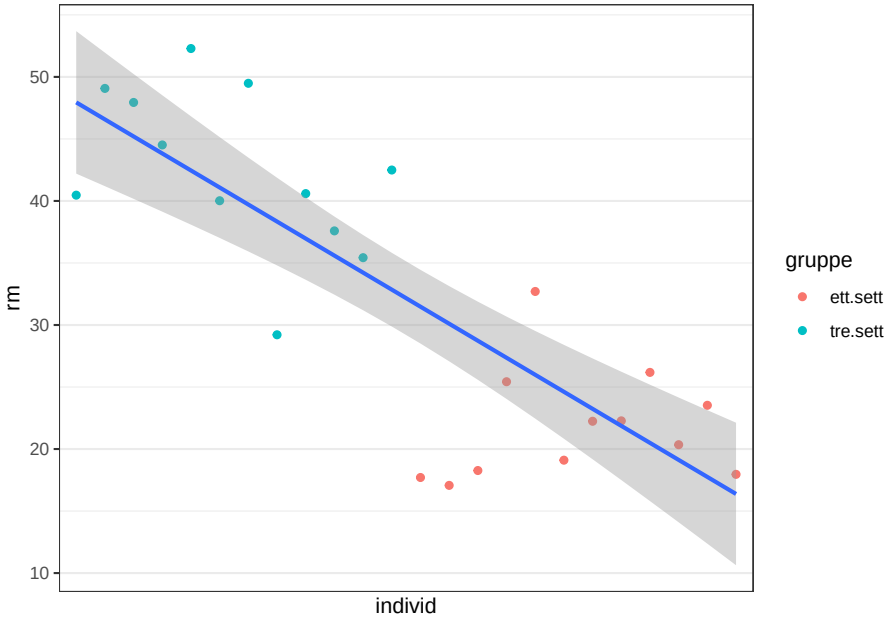


Figure 5.1: **CAPTION THIS FIGURE!!**

Chapter 6

f-ratio og signifikanstesting

We have finished a nice book.

Chapter 7

F-tester i Jamovi

We have finished a nice book.

Chapter 8

Koding av kategoriske variabler

I tabellen på s. kan du se at vi har en tabell med tre kolonner: en kolonne for hver variabel vi har i vårt datasett. Variabelen **gruppe** er en kategorisk vaiabel som har to ulike verdier: "ett.sett" og "tre.sett". Dette er de to gruppene som vi skal teste om er forskjellige. I programmeringsverdenen kalles disse denne typen data for et tekstobjekt, "strings" (python/javascript) eller "characters" (R). På norsk kalles disse verdiene for ord. Uansett navn er problemet at vi ikke kan putte ord inn i en statistisk modell; vi er nødt til å representere denne kategoriske vaiabelen med tallverdier. Det er flere måter å gjøre dette på, men de forskjellige måtene gir ulik resultat. Derfor må vi vie en god del tid på dette. Vi går gjennom to måter å gjøre dette på.

8.1 Dummykoding

En vanlig metode kalles **dummykoding** eller **treatment-koding**. Den går ut på å lage en eller flere variabler med 0 og 1 som de to mulige verdiene. Antall variabler vi trenger avhenger av antall grupper vi vil sammenligne. Siden vårt datasett kun inneholder to grupper, så trenger vi kun en variabel. Vi kan den ene gruppen og den andre 1. Hovedregelen er at vi gir 0 til baselinegruppe og 1 til den eksperimentelle gruppen. Vi gir derfor 0 til 1.sett-gruppen og 1 til 3.sett-gruppen. Gjør dette før du går videre.

I R og Jamovi kan du gjøre det med følgende if/else statement. I R kan du bruke følgende kode:

```
#lager et nytt objekt som heter dummykodet.dat
dummykodet.dat <- dat %>%
```

Table 8.1: Dummy koding

individ	gruppe	rm	dummykodet
1	tre.sett	40.467	1
2	tre.sett	49.072	1
3	tre.sett	47.941	1
4	tre.sett	44.514	1
5	tre.sett	52.288	1
6	tre.sett	40.018	1
7	tre.sett	49.484	1
8	tre.sett	29.210	1
9	tre.sett	40.593	1
10	tre.sett	37.587	1
11	tre.sett	35.427	1
12	tre.sett	42.494	1
13	ett.sett	17.706	0
14	ett.sett	17.072	0
15	ett.sett	18.268	0
16	ett.sett	25.426	0
17	ett.sett	32.703	0
18	ett.sett	19.102	0
19	ett.sett	22.238	0
20	ett.sett	22.271	0
21	ett.sett	26.179	0
22	ett.sett	20.349	0
23	ett.sett	23.528	0
24	ett.sett	17.960	0

```
# her lager jeg en ny kolonne som heter dummykoder. If gruppe == 'ett.sett', gi verd
mutate(dummykodet = if_else(gruppe == "ett.sett", 0, 1))
```

I jamovi ville jeg sett følgende video:
https://www.youtube.com/watch?v=iITxK27LfZk

8.2 Kontrastkoding

Kontrastkoding er et alternativ til dummykoding. Det er en regel som er viktig å følge for å ha en gyldig kontrastkode, og det er at summen av kontrastkodene blir 0. For eksempel er -0.5 og 0.5 gyldige kontrastkoder fordi summen av disse blir 0. Det samme er -10 og +10. 0 og 1 derimot, slik vi har med en dummykodet variabel, er ikke er en gyldig kontrastkode fordi summen av disse blir 1. **Hvilke verdier vi velger å bruke på vår kontrastkodede**

Table 8.2: Kontrastkoding

individ	gruppe	rm	dummykodet	kontrastkodet
1	tre.sett	40.467	1	0.5
2	tre.sett	49.072	1	0.5
3	tre.sett	47.941	1	0.5
4	tre.sett	44.514	1	0.5
5	tre.sett	52.288	1	0.5
6	tre.sett	40.018	1	0.5
7	tre.sett	49.484	1	0.5
8	tre.sett	29.210	1	0.5
9	tre.sett	40.593	1	0.5
10	tre.sett	37.587	1	0.5
11	tre.sett	35.427	1	0.5
12	tre.sett	42.494	1	0.5
13	ett.sett	17.706	0	-0.5
14	ett.sett	17.072	0	-0.5
15	ett.sett	18.268	0	-0.5
16	ett.sett	25.426	0	-0.5
17	ett.sett	32.703	0	-0.5
18	ett.sett	19.102	0	-0.5
19	ett.sett	22.238	0	-0.5
20	ett.sett	22.271	0	-0.5
21	ett.sett	26.179	0	-0.5
22	ett.sett	20.349	0	-0.5
23	ett.sett	23.528	0	-0.5
24	ett.sett	17.960	0	-0.5

variabel betyr ingenting for den statistiske test vi gjennomfører, men gjør at vi må fortolke resultatene litt forskjellig. Med en kontrastkode på +10 og -10 er det en 20 enhets forskjell, mens det ved +0.5 og -0.5 kun er enhet forskjell.

```
#lager et nytt objekt som heter dummykodet.dat
kontrastkodet.dat <- dummykodet.dat %>%
  # her lager jeg en ny kolonne som heter kontrastkodet. If gruppe == 'ett.sett', gi verdien -0.5
  mutate(kontrastkodet = if_else(gruppe == "ett.sett", -0.5, +0.5)
         )
```

Spørsmålet dere sikkert lurer på er hvorfor vi dummykoder og kontrastkoder gruppe-variabelen vår. Det korte svaret er at vo gjør det fordi vi skal se at disse to måtene å kode på produserer forskjellige svar.

Chapter 9

Hvilken modell skal vi velge?

Her tenker jeg å si litt om at vi har ulike muligheter.