Bygge statistisk modeller med kategoriske variabler (ANOVA, t-test)

Christian Magelssen

2021-04-01

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**.

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.

Andre ganger kan det være interessant å teste om to grupper, som var like før et eksperiment, har blitt forskjellige fordi vi har behandlet dem ulikt. 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

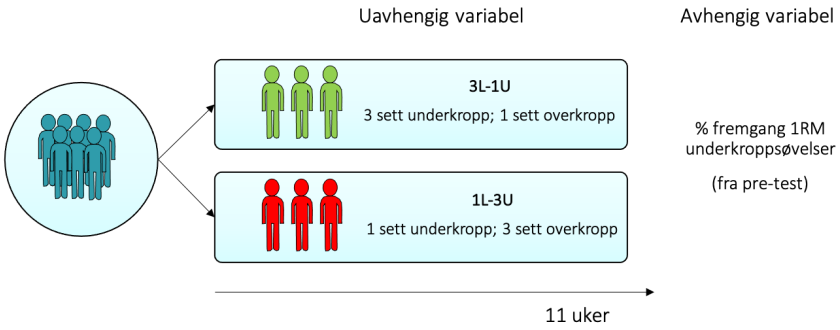
Datasett

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

Mange utrente lurer på hvor mange serier de bør gjennnomføre for å oppnå maksimal treningseffekt i styrketrening. Noen føler at de blir slitne etter ett sett og at dette derfor er tilstrekkelig. Andre mener at et hardere treningstimuli er nødvendig, selv om man er utrent, og at to eller flere sett derfor er bedre. En forsker som var tidlig ute med å undersøke var Bent Rønnestad (Rønnestad et al., 2007)

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. Forskergruppen ville så se hva som ga mest fremgang i 1RM på underkroppsøvelser. Den avhengige variabelen ble derfor %-fremgang på 1RM på underkroppsøvelser.



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

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))
```

Oppgave 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:

a) Hvor mange kolonner er det i tabellen over?
b) Hvor mange deltakere var med i studien?
c) Hvilke to verdier kan variabelen gruppe? og

2.2 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 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:

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

```
#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} **Oppgave** a) Hvilken gruppe hadde mest fremgang? ett.sett tre.sett‘

2.3 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 (Weissgerber et al., 2015)(https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.1002128). Deretter svar på følgende spørsmål for å se om du har forstått problemene ved å bruke stolpediagram på kontinuerlig data.

Oppgave a) ”Stolpediagram er designet for kontinuerlig kategorisk data. b) Høyden på stolpen representerer (bruk det norske begrepet!), hvilket vil si at det også må ligge noen observasjoner over og under stolpen. c) Et stolpediagram viser ikke standard error standardavvik CI fordelingen av observasjonene, og dette spesielt være problematisk ved store små. d) Forfatterne av artikkelen anbefaler mer bruk av bar graph scatterplot for kontinuerlige variabler. e) Ofte er det brukt error sammen med stolpediagram.

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 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”.

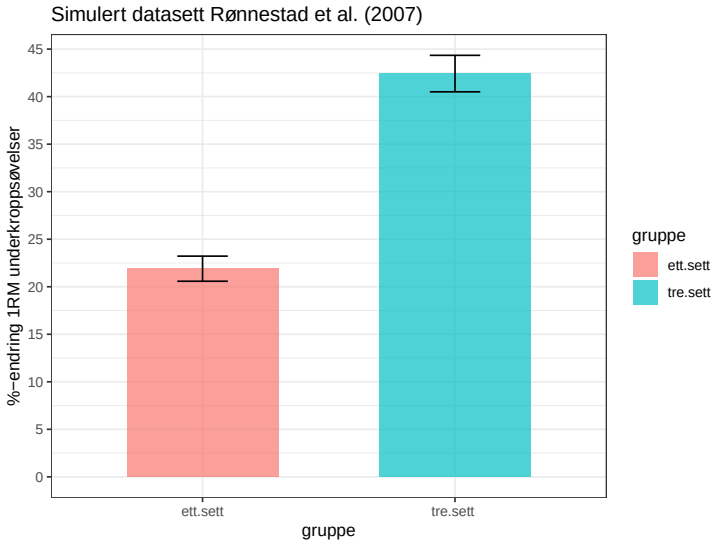


Figure 2.1: Here is a nice figure!

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

Bygge statistisk modeller

4.1 Hensikten med modellbygging

Modellbygging er en av forskernes viktigste oppgaver. Vi bygger modeller for å predikere hva en person faktisk har skåret på den avhengige variabelen. 'Ok', sier du, ', men hvorfor skal vi bygge modeller når vi faktisk har målt personen sin prestasjon på den avhengige variabelen?'. Svaret på dette spørsmålet er at vi bygger modell fordi vi ønsker å forstå relasjonene mellom variablene vi har målt. Hvis vi klarer å bygge en god modell med variablene vi ha målt, så har vi en god forståelse av hvordan variablene henger sammen. Da vil modellen vår

Hovedideen med slik modellbygging er at vi ønsker å bygge en statistisk modell til å predikere hva en person faktisk har hatt som skår på den avhengige variabelen. Til dette kan vi bruke en lineær modell som er en variant av følgende ligning:

$$data_i = (modell) + error_i$$

Data er den avhengige variabelen som vi har målt hos alle deltakerne og som vi kan bruke en modell til å predikere. Å predikere er et verb som benyttes mye i statistikken, og er synonymt med å forutsi. Min måte å tenke på det er at vi ønsker å si hva en person hadde som faktisk observasjon på den målte avhengig variabelen. Legg også merke til den lille i-en som står bak data og error i ligningen. Denne betyr individ, og sier at vi kan predikere et individ sin skår på den avhengige variabelen med modellen som vi har bygget. **Modell** er egentlig bare en representasjon av denne dataen, mens **error** er hvor mye modellen bommer fra den faktisk observasjonen (dvs. data). Dette blir kanskje mer konkret om vi bruker et eksempel:

La oss si at du er lege og at du får inn en pasient som sier hun har feber. Du vet at den normale kroppstemperaturen er 37 så dette blir modellen din.

$$data = 37 + error$$

Det neste du gjør er å ta en febermåling av pasienten, og du måler kroppstemperaturen hennes til å være 40.

$$40 = 37 + error$$

Modellen din bommer derfor mer 3, fordi 40-37 = 3.

$$40 = 37 + 3$$

Formelt sett regner vi error for en hvilken som helst modell ved å få error alene i ligningen.

$$data = modell + error$$

$$error = data - modell$$

$$3 = 40 - 37$$

Dette var en superenkel modell, men viser hvordan vi kan bruke slike modeller. Ofte bygger vi ikke modeller for ett individ, men flere. Tenk bare hvor mange deltakere vi har med i en studie. Modellen vi bygger bør være en god representasjon av alle disse individene. Med andre ord bør erroren være så liten som mulig. **Dette er essensielt!** Vi ønsker å bygge statistiske modeller som er gode, og vi ønsker å sammenligne ulike modeller for å se hvilke av disse modellene som reduserer erroren mest mulig.

Det er en mer presis og korrekt måte å skrive ligningen over på, og som du ofte ser i artikler og statistikkbøker:

$$data = (modell) + error$$

$$data = (b_0) + error$$

$$Y_i = (b_0 + b_1 X_i) + error$$

Her er $ **Y__i** $ den avhengige variabelen som vi faktisk har målt for et individ, *i*. **X__i** er dette individets faktiske måling på variabel X, som vi ofte kaller for prediktorvariabel. Som det fremgår av den siste ligningen har også $ b1 $ hektet på seg. Denne forteller oss forholdet mellom prediktorvariabelen (Xi) og den avhengig variabelen (). Vi skriver den liten liten *b* fordi dette er noe vi estimerer fra et utvalg. **b0** er vår prediksjon når Xi er **null** og **0**.

I figurene under ser tre ulike modelle med uinteressante X og Y variabler. Alle har samme b0, mens de har forskjellig b1. Husk at b1 forteller om forholdet mellom X og Y. I modell A ser du at når X øker så øker Y med 0.4 for hver enhets økning i X. I modell B er det ingen relasjon mellom X og Y, så for en enhets økning X, vil Y være den samme. I modell C er det en negativ relasjon mellom X og Y. Denne modellen sier at for en enhets økning i X, vil vi forvente Y går ned med 0.4 (siden den er negativ).

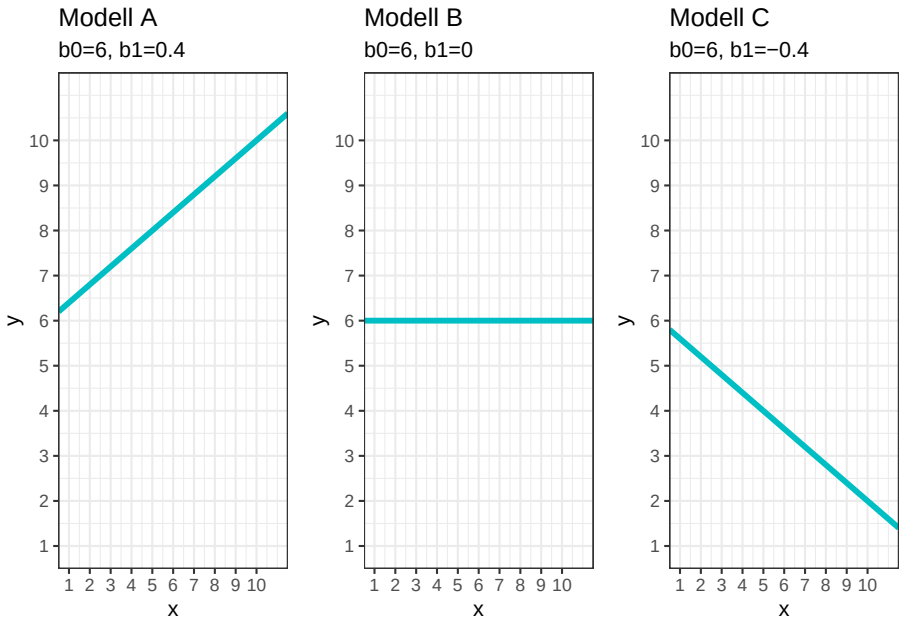


Figure 4.1: **CAPTION THIS FIGURE!!**

4.1.1 Test kunnskapen din

La oss si at vi hatt med et målt et individ sin **X** og **Y** (du kan bytte ut X og Y med hvilken som helst variabel (f.eks. høyde, vekt), hvis du vil). Individet sitt mål på X er 8. Hvis du bruker modell A, hva vil du forvente at denne personen har på *Y*?

.

I figuren ser du tre modeller som har ulike b1, men samme b0. b0 er verdien på Y når X = null/0.

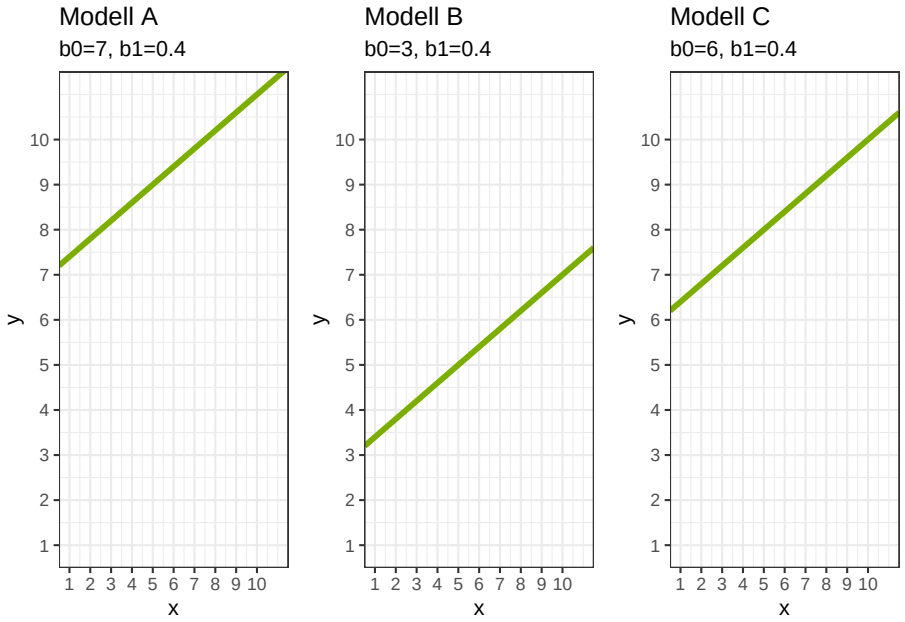


Figure 4.2: **CAPTION THIS FIGURE!!**

4.1.2 Test kunnskapen din

La oss si at vi hatt med et målt et individ sin **X** og **Y** (du kan bytte ut X og Y med hvilken som helst variabel (f.eks. høyde, vekt), hvis du vil). Individet sitt mål på X er 3. Hvis du bruker modell B, hva vil du forvente at denne personen har på Y?

.

4.2 Modellbygging med 'Null-Hypothesis Significance Testing (NHST)'

Nå som du har en fått en innføring i hvordan du kan bygge modeller er det på tide at vi begynner å spesifisere hvilke modeller vi skal bygge. Som du sikkert er kjent med jobber forskere innenfor et paradigme som kalles for **Null-Hypothesis Significance Testing (NHST)**. Dette 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)

For å teste disse hypotesene må forskeren bygge to modeller: en modell for null-hypotesen (vi kaller denne for **null-modellen**) og en **alternativ-modell**. Vi regner ut hvor mye error det er i hver av disse modellene for å se hvilke av disse modellene det er klokt å benytte. Husk at målet er å benytte modeller som er gode og som har lite error. Hvis null-modellen er god nok, så er det ikke noe poeng å bruke den alternative modellen. Men hvis den alternative modellen er mye bedre enn null-modellen, da bør benytte denne. 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).

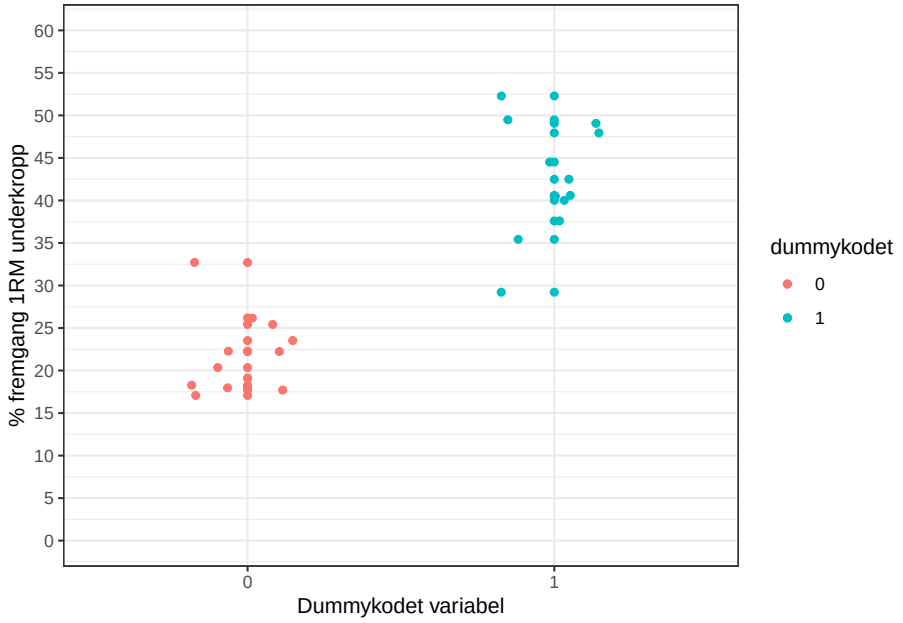


Figure 4.3: **CAPTION THIS FIGURE!!**

4.3 Null-modellen (null-hypotesen)

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). Husk at vi har laget en variabel hvor vi har kodet disse som 0 og 1. Null-hypotesen er at det ikke er noen forskjeller mellom gruppene. En annen måte å formulere dette på er om vi blir bedre til å predikere et individs skår hvis vi vet hvilken gruppe de tilhører eller om vi kun trenger en enkel modell. Den enkleste modellen vi kan benytte er gjennomsnittet i % fremgang 1RM for alle deltakerne. Dette er modellen som representerer null-hypotesen. Med andre ord vår null-modell

$$Y_i = (b_0) + error$$

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

Det er ofte enklere å se denne modellen i tabellform, slik som dere ser under.

Null-modellen (mean)

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 hvordan denne modellen virker. For individ 1 målte vi en fremgang i 1RM underkropp på **40.467**, men modellen vår sa **32.162**. Så modellen bommet med 8.305, dvs. 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. Husk at vi ikke er interessert i hvir mye bommer for hvert enkelt individ, men for alle indivene. Summer derfor all erroren for alle indidene, hvilket tall får du da? null 0 3 -3. (tenk over hvorfor du får dette svaret før du leser videre)

Som du så i forrige oppgave blir det feil å summere alle erroren, men ved å regne **Sum of Squared Error** løser vi dette problemet effektivt. Det vi gjør er å gange error med seg selv (error^2) før vi summerer alt dette sammen.

Hvis vi regner ut **Sum of Squared Error** for null-modellen fpr vi: . Dette tallet er viktig! Dette er null-hypotesen vår. Hvis det ikke er noen forskjell mellom de to treningsgruppene våre er det like greit å bruke denne null-modellen. 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.

Før du går videre er det greit å visualisere hvordan null-hypotesen ser ut rent visuelt. Den prikkete streken i figuren under representerer modellen vår som er mean. Som du ser, så gjør den ingen justeringer for de ulike individene. Erroren er avstanden fra den linjen og opp til hvert datapunkt.

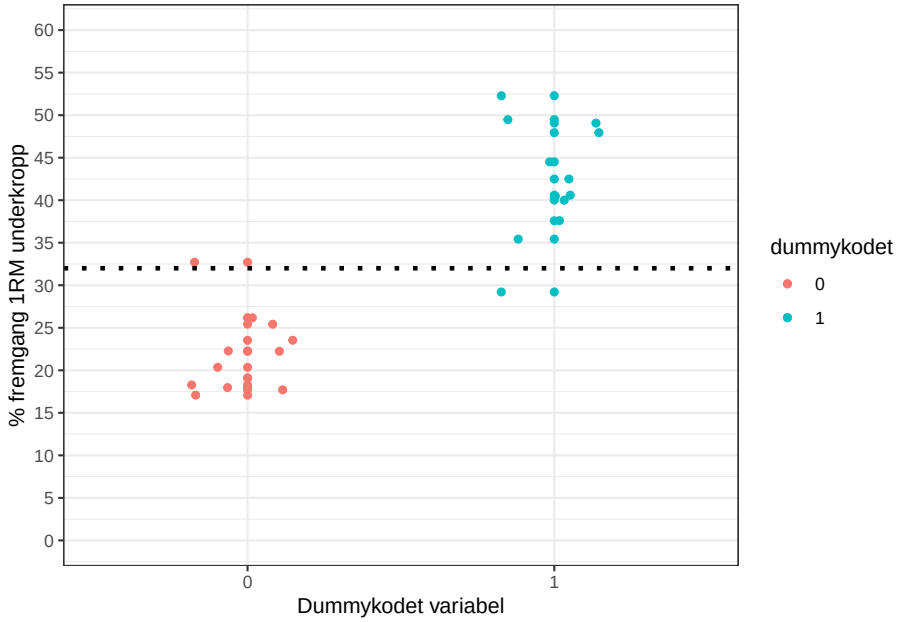


Figure 4.4: **CAPTION THIS FIGURE!!**

4.4 H1: Alternativ modell (alternativ hypotese)

I forrige avsnitt sa vi at **null-hypotesen (H0)** reresenterer en en modell som gir samme prediksjon for alle deltakerne som var med i studien uavhengig av hvilken treningsgruppe de tilhører. Vi kalte denne for null-modellen. Vi regnet oss også frem til at denne modellen ga oss en Som of Squared error på 3243.784.

$$Y_i = (b_0) + error_i$$

$$Fremgang.1RM_i = (mean) + error_i$$

Spørsmålet vi skal stille i dette avsnittet er om vi kan redusere error fra denne ved å benytte en mer kompleks modell som benytter (vår dummykodede

kategoriske variabel) som prediktorvariabel:

$$Y_i = (b_0 + b_1 X_i) + error$$

Prediktorvariabelen b1 er en gruppevariabelen vår som vi dummykodet med tallene 0 og 1.

$$Fremgang.1RM_i = b_0 + b_1(Gruppe) + error_i$$

For å fokusere på holde dette på et overordnet nivå, så vil jeg gi dere de estimerte verdiene for b0 og b1. Målet er å vise dere hvordan denne modellen fungerer. Senere skal gå gjennom hvordan vi regner ut disse verdiene.

$$Fremgang.1RM_i = b_0(21.90) + b_1(20.52 * Gruppe) + error_i$$

Modellen sier at vi har en intercept på 21.90. Dette er forventede verdien på Y (Fremgang.1RM) når prediktorvariabelen er 0. Modellen sier også at b1 er 20.52. Med andre ord den forventede økning i Y for en enhets økning i X. Husk at vi lagde en gruppe-variabel der vi kodet de to gruppene våre med 0 og 1. Så hvis et individ tilhørte gruppe 0, blir vår prediksjon:

$$Fremgang.1RM_i = 21.90 + b_1(20.52 * 0) + error_i$$

0*20.52 = 0, så blir stående igjen med b0, vår prediksjon av Y når er nulll

$$Fremgang.1RM_i = 21.90 + 0 + error_i$$

$$Fremgang.1RM_i = 21.90 + error_i$$

Hvis individet derimot tilhører 1 predikerer modellen at individet sin skår blir 42.48.

$$Fremgang.1RM_i = 21.90 + b1(20.52 * 1) + error_i$$

$$Fremgang.1RM_i = 42.48 + error_i$$

Visualisert fremstilt blir modellen vår seendes slik ut:

```
ggplot(dat, aes(dummykodet, rm, color=dummykodet)) +
  geom_point()+
  scale_y_continuous(breaks = seq(0, 60, 5)) +
  coord_cartesian(ylim = c(0, 60)) +
  geom_jitter(width = 0.2) +
  stat_summary(geom = "line", fun = mean, group = 1, color="black", linetype="dotted",
  labs(y="% fremgang 1RM underkropp", x="Dummykodet variabel") +
  theme_bw()
```

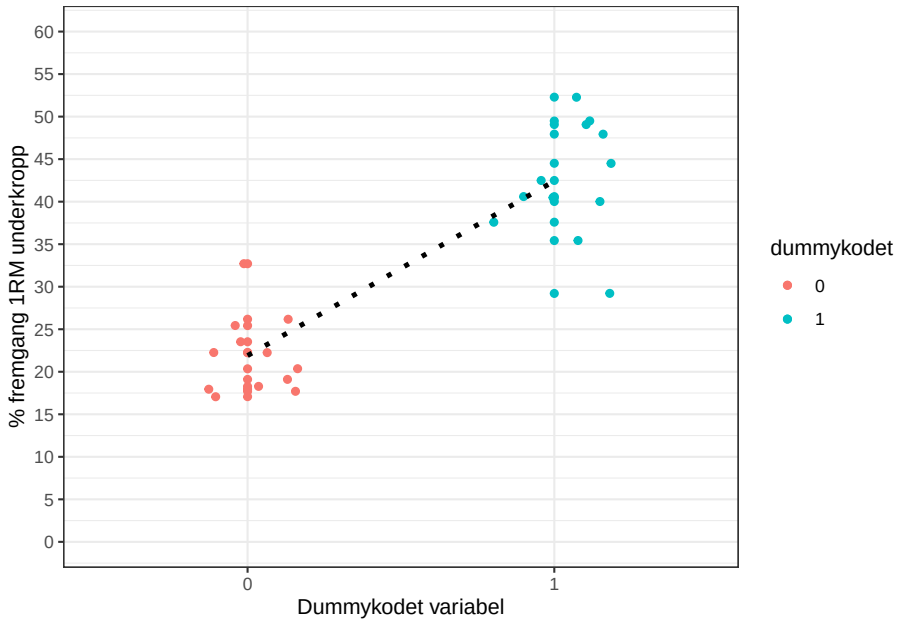


Figure 4.5: **CAPTION THIS FIGURE!!**

Oppgaver Tabellen under viser 6 individer som tilhørte treningsgruppe. Du ser deres faktiske fremgang i 1RM kolonnen. La oss bruke det vi har lært til å predikere disse personene sin fremgang. Vi bruker samme modell som over

$$Fremgang.1RM_i = b_0(21.90) + b_1(20.52 * Gruppe) + error_i$$

a) Hva predikerer modellen at individ nummer 3 hadde i skår? (to desimaler)

b) Hva hadde individ nr i skår?

c) hvor mye error blir det?

d) i Squared Error blir denne erroren?

e) nå som du har jobbet med denne modellen, så lurer jeg på om det er noe kjent med disse verdiene i modellen. Gå tilbake til [link] hvis du trenger et hint.

bo er (norskt ord) for gruppen som er kodet med 0. b1 er (norsk ord) mellom gruppen som er kodet med 0 og gruppen som er kodet med 1. b0 + b1 (norsk ord) for gruppen som er kodet med 1.

Table 4.1: 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

I forrige oppgave regnet du ut error for ett enkelt individ. Men vi er interessert i den totale erroren for modellen. Formelen for denne er:

total error in den alternative modellen:

$$\text{SS\_R} = \sum_{n=1}^{N}(observert_i - modell_i)^2$$

Med andre ord er det kvadraten av den faktiske observasjonen - hva modellen sa. Bruk formelen til å regne ut dette. (to desimaler

4.5 Variansanalyse (ANOVA-tabell)

Nå som vi har regnet ut hvor mye error det er i hver av disse modellene - null-modellen og den alternative modellen - er det klart for å gjøre en sammenligning av disse modellene. Modellene vi sammenligner er om alternative-modellen reduserer error mer enn null-modellen. Fra figurene under kan det slik ut; linjen til høyre ser ut til å ligge nærmere observasjonene enn linjen i figuren til venstre.

The total sum of squares (SST) = 3243.784 (SST) er hvor mye error vi får ved å bruke null-modellen.

The residual sum of squares (SSR) = 716.3 (dette er hvor mye error som er igjen etter at vi brukte modellen vår)

En naturlig ting å gjøre er å regne hvor mye error den alternative modellen vår har redusert error med. Man kaller denne for **The model sum of squares (SSM)**:

The model sum of squares (SSM) = (SST) - (SSR) =

Hvis vi regner hvor mye error modellen vår har redusert error med, så kan vi se at modellen vår har redusert error med 78 %. Dette er ekstremt mye, og det er sjeldent man finner en så høy prosent. Denne verdien har mange forskjellige navn i statistikken, *proportional reduction in error (PRE)“, R2 og n2”. Og den er viktig. Jeg velger å bruke R2.

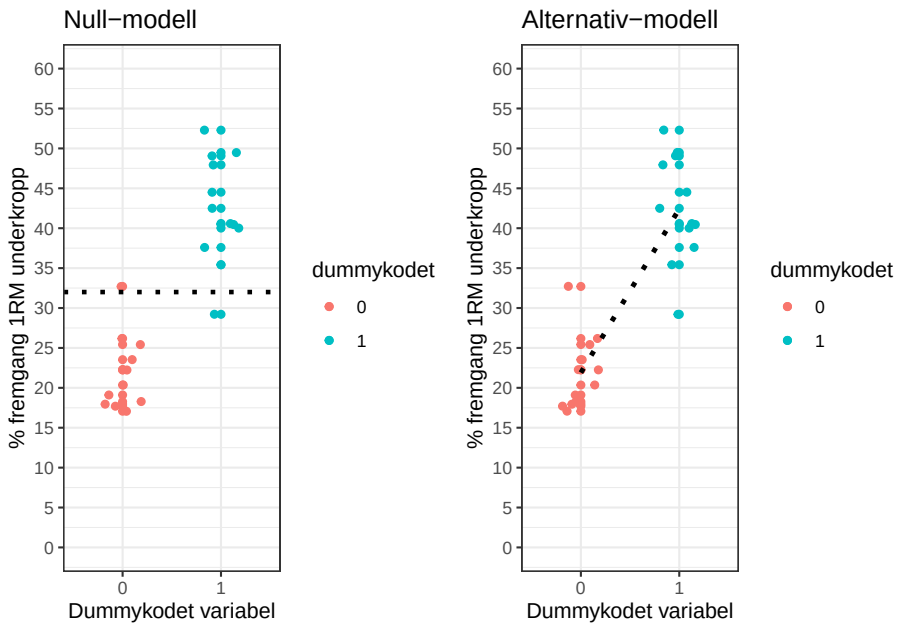


Figure 4.6: **CAPTION THIS FIGURE!!**

$$R2 = (SS_T - SS_R)/SS_T$$

$$R2 = (3243.784 - 716.3)/3243.784$$

$$R2 = 0.7791826$$

Når dere bygger statistiske modeller i R,Jamovi eller SPSS vil dere få en ANOVA-tabell de. Her ser du de samme verdiene som vi har regnet manuelt.

```
#aov er en forkortolse for analysis of variance (ANOVA)
#dette er funksjon som kommer mer R.
aov(rm ~ dummykodet, dat)
```

```
## Call:
##    aov(formula = rm ~ dummykodet, data = dat)
##
## Terms:
##                 dummykodet Residuals
```

```
## Sum of Squares   2527.4962  716.2875
## Deg. of Freedom          1        22
##
## Residual standard error: 5.706008
## Estimated effects may be unbalanced
```

4.6 F-test

```
## # A tibble: 24 x 6
##       ss sum.ss mod.m  error sum.error   pre
##    <dbl>  <dbl> <dbl>  <dbl>     <dbl> <dbl>
##  1  8.30  3244.  42.4   3.81      716. 0.779
##  2 16.9   3244.  42.4  44.3       716. 0.779
##  3 15.8   3244.  42.4  30.5       716. 0.779
##  4 12.4   3244.  42.4   4.38      716. 0.779
##  5 20.1   3244.  42.4  97.4       716. 0.779
##  6  7.86  3244.  42.4   5.77      716. 0.779
##  7 17.3   3244.  42.4  49.9       716. 0.779
##  8 -2.95  3244.  42.4 174.        716. 0.779
##  9  8.43  3244.  42.4   3.34      716. 0.779
## 10  5.42  3244.  42.4  23.4       716. 0.779
## # ... with 14 more rows
```

4.7 T-test

ANOVA. En annen måte vi kan teste om det er forskjeller mellom to grupper er å teste om det er forskjeller er å kjøre en uavhengig t-test. Ofte sier man at det - Og når ikke. Men har istf. valgt å bygge opp den statistiske kunnskapen vår ved å kjøre en ligning. Så når blir ikke forskjellen så stor.

Chapter 5

Hvordan finne linjen i modellen?

Nå som vi er kjent med hvordan vi kan bygge og teste statistiske modeller, er det på tide å vise hvordan vi finner regresjonslinjen som vi skal bruke. Mer presist, hvilke verdier skal vi ha for $b0$ og $b1$ som beskriver denne linjen? Hittil har dere fått disse verdiene av meg, men det vi skal lære nå er hvordan vi kan regne ut disse verdiene for hånd. En viktig sannhet om denne linjen er at regresjonslinjen (les modellen) er plassert slik at den reduserer Sum of Squared Error mest mulig. Med andre ord, verdiene på $b0$ og $b1$ (som beskriver denne linjen) er slik at det er umulig å redusere error mer. Spørsmålet er hvordan vi finner verdiene på $b0$ og $b1$ som beskriver denne linjen. En tilnærming kan være å gjette seg fram til hva b_0 og b_1 skal være. Vi kan teste ut ulike verdier for b0 og b1, og evaluere hvor mye sum of Squared Error disse gir. I figuren under har jeg prøvd tre ulike modeller, og regner ut hvor mye sum of squared error disse gir.

Oppgave a) Hvilken av modellene over gir mest sum of squared error? (SSModel)

b0=30 b1=7 b0=25 b1=28 b0=10 b1=30

Vi kan holde på slik med slik prøving-og-feiling til vi faktisk finner linjen som reduserer error mest. Det er bare å teste nok verdier. **Minste kvadraters metode** garanterer oss å alltid gi oss er svar. Jeg har laget en video til dere som viser vi kan prøve-og-feile til vi kommer frem til en løsning (se denne):

Det er en mer effektiv måte å løse dette problemet på. For å finne **b1** kan vi bruke følgende formel:

$$b_1 = \frac{SCP}{SS_x}$$

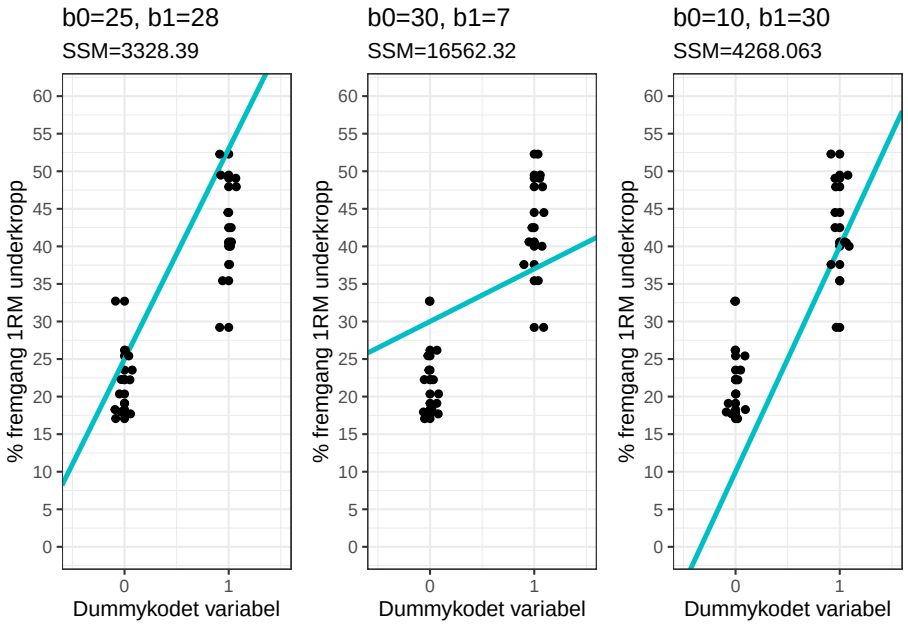


Figure 5.1: **CAPTION THIS FIGURE!!**

Her var det et nytt begrep, **SCP**. SCP står for sum of cross-product deviations. Det brukes til å finne relasjonen mellom to variabler, og er grunnlaget for en rekke utregninger i statistikken, så det kan være lurt å lære seg. SCP finner ut av om en person som er over eller under gjennomsnittet på en variabel, også er over eller under gjennomsnittet på den andre variabelen.

$$SCP = \frac{\sum_{n=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{SS_x}$$

$\bar{x}$ er gjennomsnittet på x-variabelen (gruppe), mens $\bar{y}$ er gjennomsnittet for y-variabelen (1RM).

Vi kan også en generell formell for å løse dette problemet på. Det første vi må gjøre er å regne noe som heter **sum of cross-product deviations (SCP)**. Formelen for dette er others are negative, so they'll cancel out. Instead we square the deviances before adding them up. We want to do something similar here, but at the same time gain some insight into whether the deviations for one variable are matched by similar deviations in the other. The answer is to multiply the deviation for one variable by the corresponding deviation for the other. If both deviations are positive or negative then this will give us a positive value (indicative of the deviations being in the same direction), but if one deviation is positive and the other negative then the resulting product will

be negative (indicative of the deviations being opposite in direction). The deviations of one variable multiplied by the corresponding deviations of a second variable are known as the cross-product deviations. If we want the total of these cross-product deviations we can add them up, which gives us the sum of crossproduct $\sum_{n=1}^{N}(x-\bar{x})(y-\bar{y})$ Hva denne gjør er

Det neste er å faktor inn hvor mye

of deviation from the mean, if it varies a lot, we would expect the outcome to show a lot of deviation from its mean too. Conversely, if the predictor deviates only a little from its mean (it has little variance) then the outcome should likewise show only small deviations from its mean. Therefore, what we expect to happen with the outcome depends on how much the predictor deviates from its mean. If the predictor deviates a little from the mean, then the SCP should be smaller than if the predictor deviates a lot from the mean. Therefore, we need to factor in how much the predictor deviates from its mean: we want the regression coefficient to reflect the total relationship between the predictor and outcome relative to how much the predictor deviates from its mean. For a single variable, how do we quantify the total degree to which it deviates from its mean?'

Bibliography

Rønnestad, B., Egeland, W., Kvamme, N., Refsnes, P., Kadi, F., and Raastad, T. (2007). Dissimilar effects of one- and three-set strength training on strength and muscle mass gains in upper and lower body in untrained subjects. *The Journal of Strength and Conditioning Research*, 21:157–63.

Weissgerber, T. L., Milic, N. M., Winham, S. J., and Garovic, V. D. (2015). Beyond Bar and Line Graphs: Time for a New Data Presentation Paradigm. *PLOS Biology*, 13(4):1–10. Publisher: Public Library of Science.