

Seminar 5

Martin Søyland

Laste inn data

Vi fortsetter der vi slapp sist! Altså med European Social Survey data (Norge subsett).

```
load("./data/ess_norge.rda")
```

Variabler

Tabellen under viser en kort variabelbeskrivelse. Jeg har lagt inn labels i selve datasettet og det skal vi jobbe med i første del denne gangen.

Variabel	Målenivå	Beskrivelse
idno	Forholdstall	ID indikator for hver enhet
party_vote	Nominal	Parti stemt på forrige Stortingsvalg
party_vote_short	Nominal	Forkortet navn på parti stemt på forrige Stortingsvalg
gender	Dikotom	Kjønn
year_born	Forholdstall	År født
income_feel	Ordinal	Hvordan føler du din økonomiske situasjon er for tiden?
income_decile	Ordinal	Husholningens inntekt i desiler
trust_parl	Ordinal	Hvor mye stoler du på det nasjonale parlamentet?
trust_legalsys	Ordinal	Hvor mye stoler du på rettsvesenet?
trust_police	Ordinal	Hvor mye stoler du på politiet?
trust_politicians	Ordinal	Hvor mye stoler du på politikere?
trust_polparties	Ordinal	Hvor mye stoler du på politiske partier?
trust_eurparl	Ordinal	Hvor mye stoler du på det europeiske parlamentet?
trust_unitednations	Ordinal	Hvor mye stoler du på FN?

Fikse problemer vi fant sist

Vi kjørte en multinomisk logistisk modell sist, som viste seg å ha flere mangler: 1. Små partier er problematiske (liten N) 2. Vi mangler noen kontrollvariabler (kanskje man systematisk stemmer med venstre som ung f.eks, samtidig som alder påvirker hvor mye man stoler på politikere). 3. Kanskje AV er ordinal?

Eliminere små partier

Vi fjerner først de minste partiene. Noen ganger kan det være lurt å heller slå sammen kategoriene (ref forelesningen til Solveig).

```
larger_parties <- ess_nor[which(ess_nor$party_vote_short != "RØDT" &
                              ess_nor$party_vote_short != "KYST" &
                              ess_nor$party_vote_short != "MDG" &
                              is.na(ess_nor$party_vote_short) == FALSE), ]
table(larger_parties$party_vote_short, useNA = "always")
```

```
##
##      A  FRP      H  KRF      SP  SV      V <NA>
##  339  136  328   52   59   49   65    0
```

Fikse kontrollvariabler

Videre har vi noen kontrollvariabler vi vil inkludere i regresjonen. Men disse må vi “pynte” litt på først.

Vi skal kontrollere for fire ting: inntekt, hvor fornøyd respondenten er med økonomien sin, kjønn og alder. Tanken med alle er den samme: de er bakenforliggende variabler (for tillit til politikere), og de kan tenkes å påvirke både vår avhenige og uavhengige variabel (*backdoor path*).

To inntektsvariabler

Når vi subsetter blir labels på variablene fjernet av en eller annen grunn... så vi må kopiere dem over til det nye datasettet med pakken **labelled** og funksjonen **copy_labels** først. Deretter kan vi sjekke hvilke verdier vi ikke vil ha med videre fra de forskjellige variablene.

```
library(labelled)

larger_parties$income_feel <- copy_labels(ess_nor$income_feel,
                                         larger_parties$income_feel)
attr(larger_parties$income_feel, "labels")
```

```
## Living comfortably on present income          Coping on present income
##                                     1                                     2
##      Difficult on present income      Very difficult on present income
##                                     3                                     4
##                                     Refusal                             Don't know
##                                     7                                     8
##                                     No answer
##                                     9
```

```
larger_parties$income_feel2 <- ifelse(larger_parties$income_feel > 4, NA,
                                     larger_parties$income_feel)
```

```
table(larger_parties$income_feel2, larger_parties$income_feel, useNA = "always")
```

```
##
##      1  2  3  4  8 <NA>
## 1   676  0  0  0  0  0
## 2    0 290  0  0  0  0
## 3    0  0 51  0  0  0
## 4    0  0  0 10  0  0
## <NA>  0  0  0  0  1  0
```

```
larger_parties$income_decile <- copy_labels(ess_nor$income_decile,
                                             larger_parties$income_decile)
attr(larger_parties$income_decile, "labels")
```

```
## J - 1st decile R - 2nd decile C - 3rd decile M - 4th decile
##      1      2      3      4
## F - 5th decile S - 6th decile K - 7th decile P - 8th decile
##      5      6      7      8
## D - 9th decile H - 10th decile Refusal Don't know
##      9     10     77     88
##      No answer
##      99
```

```
larger_parties$income_decile2 <- ifelse(larger_parties$income_decile > 10, NA,
                                         larger_parties$income_decile)
```

```
table(larger_parties$income_decile2, larger_parties$income_decile, useNA = "always")
```

```
##
##      1  2  3  4  5  6  7  8  9 10 77 88 <NA>
## 1   67  0  0  0  0  0  0  0  0  0  0  0  0
## 2    0 104  0  0  0  0  0  0  0  0  0  0  0
## 3    0  0 111  0  0  0  0  0  0  0  0  0  0
## 4    0  0  0 129  0  0  0  0  0  0  0  0  0
## 5    0  0  0  0 100  0  0  0  0  0  0  0  0
## 6    0  0  0  0  0 97  0  0  0  0  0  0  0
## 7    0  0  0  0  0  0 96  0  0  0  0  0  0
## 8    0  0  0  0  0  0  0 113  0  0  0  0  0
## 9    0  0  0  0  0  0  0  0 84  0  0  0  0
## 10   0  0  0  0  0  0  0  0  0 100  0  0  0
## <NA>  0  0  0  0  0  0  0  0  0  0  4 23  0
```

Kjønn og alder

Kjønn ser ut til å være kodet på en fornuftig måte, så her trenger vi ikke gjøre noe. Alder kan vi regne ut med å trekke fødselsår fra året surveyen ble utført (2014). Så sentrerer vi variabelen til median.

```
table(larger_parties$gender) # Ca like mange, så referansekategori er ikke viktig

##
## female    male
##      449     498

larger_parties$age <- 2014 - larger_parties$year_born
larger_parties$age_sen <- larger_parties$age - median(larger_parties$age) # ingen har NA
summary(larger_parties$age_sen)

##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## -31.0000 -12.0000   0.0000   0.1284  13.0000  45.0000
```

Multinomisk med kontroller

Da er det bare å plugge inn variablene i en regresjon. Vi fortsetter med pakken **nnet** og funksjonen `multinom()`.

```
library(nnet)
party_reg2 <- multinom(party_vote_short ~ trust_politicians + income_decile2 +
                      income_feel2 + age_sen + gender,
                      data = larger_parties, Hess = TRUE, na.action = "na.exclude")

## # weights:  49 (36 variable)
## initial  value 1799.966888
## iter   10 value 1531.259350
## iter   20 value 1494.441701
## iter   30 value 1423.359640
## iter   40 value 1418.969480
## final   value 1418.968988
## converged

summary(party_reg2)

## Call:
## multinom(formula = party_vote_short ~ trust_politicians + income_decile2 +
##      income_feel2 + age_sen + gender, data = larger_parties, na.action = "na.exclude",
##      Hess = TRUE)
##
## Coefficients:
##      (Intercept) trust_politicians income_decile2 income_feel2      age_sen
## FRP  0.17841649      -0.32569309      -0.10229573   0.31972571   0.003462876
```

```
## H    -0.09318171      -0.08562603      0.09729868  -0.10389600 -0.012897728
## KRF -2.53923080        0.01949806     -0.04717871   0.41788976  0.019003260
## SP  -1.97443673      -0.07586865      0.03839968   0.15689899  0.002527822
## SV  -2.42250768        0.05181546      0.08443923  -0.05090562 -0.024670163
## V   -1.82544458        0.04574622      0.03755968  -0.10574486 -0.019699933
##      gendermale
## FRP  0.8855034
## H    0.1965203
## KRF -0.5717698
## SP   0.4189833
## SV  -0.6784468
## V   -0.2667930
##
## Std. Errors:
##      (Intercept) trust_politicians income_decile2 income_feel2      age_sen
## FRP  0.5565825      0.06061594      0.04704955      0.1884082 0.006931332
## H    0.4413251      0.04743898      0.03288934      0.1599230 0.005299506
## KRF  0.9859463      0.10757993      0.07798980      0.3155987 0.011936977
## SP   0.7808610      0.08299798      0.05912498      0.2711081 0.009467118
## SV   0.8988108      0.09870182      0.06465865      0.3092188 0.010655519
## V    0.7627909      0.08401835      0.05555361      0.2675204 0.008960022
##      gendermale
## FRP  0.2375854
## H    0.1661980
## KRF  0.3933107
## SP   0.3015149
## SV   0.3424900
## V    0.2843868
##
## Residual Deviance: 2837.938
## AIC: 2909.938
```

```
confint(party_reg2)
```

```
## , , FRP
##
##              2.5 %      97.5 %
## (Intercept)  -0.91246523  1.26929821
## trust_politicians -0.44449816 -0.20688803
## income_decile2  -0.19451115 -0.01008032
## income_feel2    -0.04954766  0.68899907
## age_sen         -0.01012228  0.01704804
## gendermale      0.41984458  1.35116227
##
```

```

## , , H
##
##                2.5 %      97.5 %
## (Intercept)    -0.95816294  0.771799519
## trust_politicians -0.17860472  0.007352672
## income_decile2    0.03283676  0.161760603
## income_feel2     -0.41733938  0.209547387
## age_sen         -0.02328457 -0.002510886
## gendermale      -0.12922177  0.522262419
##
## , , KRF
##
##                2.5 %      97.5 %
## (Intercept)    -4.471650113 -0.60681148
## trust_politicians -0.191354740  0.23035085
## income_decile2   -0.200035920  0.10567849
## income_feel2     -0.200672248  1.03645176
## age_sen         -0.004392785  0.04239931
## gendermale      -1.342644552  0.19910503
##
## , , SP
##
##                2.5 %      97.5 %
## (Intercept)    -3.50489624 -0.44397723
## trust_politicians -0.23854171  0.08680441
## income_decile2   -0.07748315  0.15428252
## income_feel2     -0.37446312  0.68826110
## age_sen         -0.01602739  0.02108303
## gendermale      -0.17197512  1.00994168
##
## , , SV
##
##                2.5 %      97.5 %
## (Intercept)    -4.1841445 -0.660870824
## trust_politicians -0.1416366  0.245267468
## income_decile2   -0.0422894  0.211167866
## income_feel2     -0.6569634  0.555152142
## age_sen         -0.0455546 -0.003785729
## gendermale      -1.3497149 -0.007178752
##
## , , V
##
##                2.5 %      97.5 %

```

```
## (Intercept)      -3.32048736 -0.330401795
## trust_politicians -0.11892672  0.210419161
## income_decile2    -0.07132338  0.146442753
## income_feel2      -0.63007518  0.418585468
## age_sen           -0.03726125 -0.002138614
## gendermale        -0.82418092  0.290594866
```

Plotte effekter

Ofte foretrekker jeg å supplere en regresjonstabell/koeffisientplot med å vise effekten av X en man fokuserer på i oppgaven/paperet mer substansielt. En måte å gjøre dette på er, som vi var litt innom i bivariat regresjon sist, å lage et *test set* man regner ut regresjonsligningen på. Funksjonen `predict()` er ekstremt kraftig her; den fungerer på så og si alle regresjonsfunksjoner!

Det første vi gjør er å konstruere et hypotetisk datasett (test set), der vi lar vår X *in focus* variere fra minimum til maksimumsverdi. Alle de andre variablene vi har som X er i regresjonen setter vi til snitt/median for kontinuerlige variabler og til en troverdig kategori på de kategoriske variablene:

Når vi holder på med multinomisk regresjon

```
test_set2 <- data.frame(trust_politicians = 0:10,
                        income_decile2 = median(larger_parties$income_decile2, na.rm = TRUE),
                        income_feel2 = median(larger_parties$income_feel2, na.rm = TRUE),
                        age_sen = 0,
                        gender = "female")
```

Når vi har opprettet dette objektet kan vi binde sammen kolonnene av test settet vår og predikerte verdier på test settet. Mer spesifikt bruker vi regresjonsobjektet vårt (alle beta og X er) og regner ut ligningen gitt at data ser ut som test settet vårt (*newdata*-argumentet sier altså at vi skal løse ligningen for vårt hypotetiske datasett). Legg også merke til argumentet *type*, som med multinomisk regresjon enten kan gi partiet med høyest sannsynlighet for hver rad (dette er *default*), eller sannsynlighet for hver rad å havne i hvert parti.

```
test_set2 <- cbind(test_set2, predict(party_reg2, newdata = test_set2, type = "probs"))
head(test_set2)
```

```
##   trust_politicians income_decile2 income_feel2 age_sen gender      A
## 1                0                5            1      0 female 0.2552720
## 2                1                5            1      0 female 0.2828051
## 3                2                5            1      0 female 0.3084808
## 4                3                5            1      0 female 0.3319519
## 5                4                5            1      0 female 0.3530615
## 6                5                5            1      0 female 0.3717960
##          FRP          H          KRF          SP          SV          V
## 1 0.25189014 0.3409532 0.02416958 0.05023893 0.03282244 0.04465376
## 2 0.20148769 0.3467304 0.02730367 0.05159113 0.03829641 0.05178565
## 3 0.15868751 0.3471730 0.03036896 0.05216350 0.04399489 0.05913134
```

```
## 4 0.12329433 0.3429305 0.03332306 0.05203127 0.04986002 0.06660888
## 5 0.09468291 0.3348069 0.03613999 0.05129681 0.05585098 0.07416084
## 6 0.07199117 0.3236397 0.03880702 0.05007204 0.06194244 0.08175161
```

Vi gjør også et siste lille triks før vi plotter ut effekten av *trust_politicians*. `ggplot()` (og veldig mange andre funksjoner i R) liker nemlig best data i **long** heller enn **wide format**. Pakken **reshape2** kan hjelpe oss med denne konverteringen, og er generelt en pakke som brukes veldig mye.

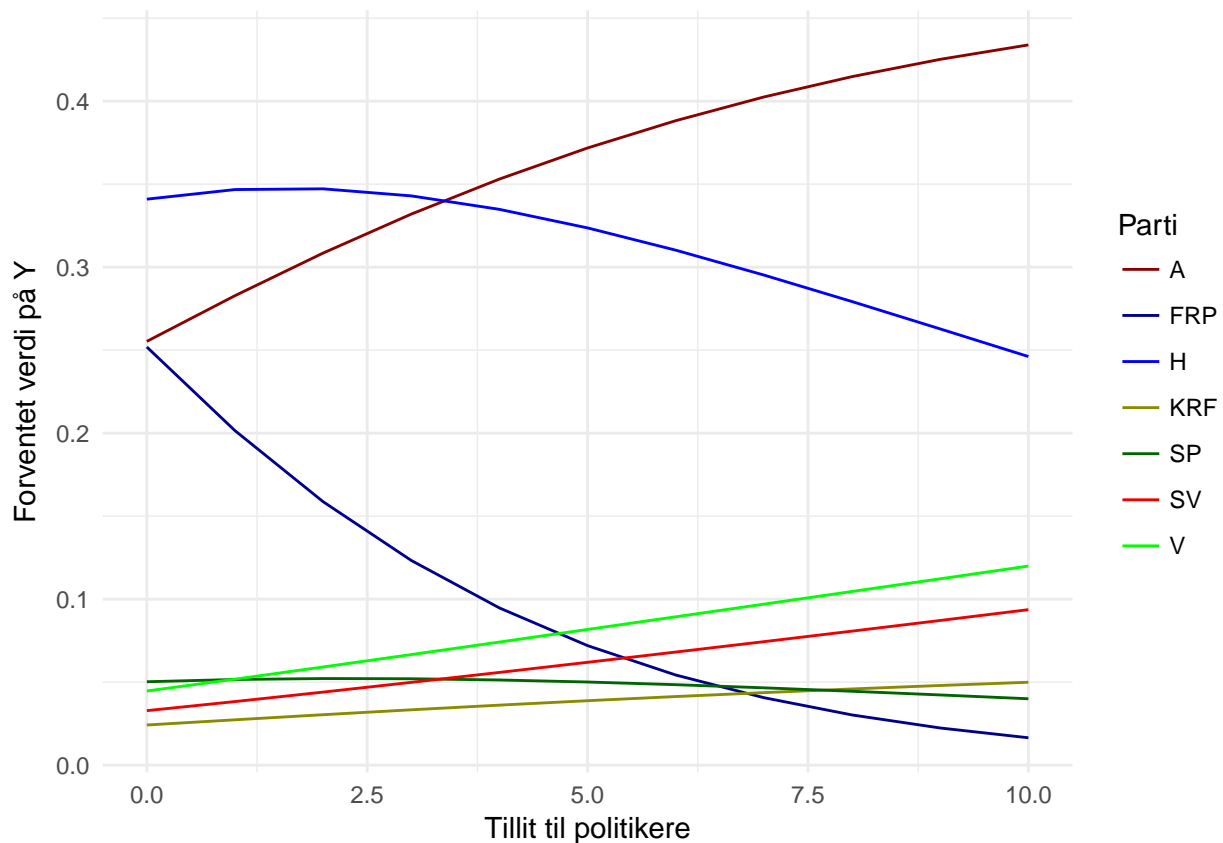
Her “smelter” vi ned data slik at vi får en egen kolonne for parti (heller enn en kolonne per parti).

```
library(reshape2)
test_set2 <- melt(test_set2, measure.vars = c("A", "FRP", "H", "KRF", "SP", "SV", "V"))
```

Til slutt kan vi bruke `ggplot()` som vi ellers ville gjort. Da får vi et veldig godt bilde av hvordan variabelen vår oppfører seg i regresjonen. Vi ser også at Solveig var smartere enn meg når hun slo sammen til tre kategorier...

```
library(ggplot2)

ggplot(test_set2, aes(x = trust_politicians, y = value, color = variable)) +
  geom_line() +
  theme_minimal() +
  scale_color_manual(values = c("darkred", "darkblue", "blue", "yellow4",
                                "darkgreen", "red2", "green")) +
  labs(x = "Tillit til politikere", y = "Forventet verdi på Y", color = "Parti")
```

Husk at alt bortsett fra *trust_politicians* er konstant her; *all else equal*!

Men hva med flernivå!?

La oss kikke på vår *trust_politicians* som avhengig variabel i en komparativ sammenheng. Jeg har juksa litt, og fiksa alle variablene på forhånd. Helt identisk med det vi har gjort over. Her vil vi kjapt på forskjellene mellom det som kalles *fixed effects* og flernivåanalyse.

Laste inn ESS med alle land

Vi starter med å laste inn datasettet ESS, nå med alle land som er inkludert (15 stk).

```
rm(list = ls())
load("./data/ess.rda")

table(ess$country)
```

```
##
##   Austria   Belgium   Czechia   Denmark   Estonia   Finland
##    1795      1769      2148      1502      2051      2087
##   France    Germany   Ireland Netherlands   Norway    Poland
```

```
##          1917          3045          2390          1919          1436          1615
##   Slovenia      Sweden Switzerland
##          1224          1791          1532
```

Flernivå vs. fixed effects

Fixed effects er en samlebetegnelse for å kontrollere bort varians som kan åpne *backdoor paths* i regresjonen vår på en enkel måte: vi bare plugges inn dummyset for, i akkurat denne sammenhengen, land. Har man paneldata vil du også lage fixed effects for år (eller periode etc). Legg merke til at R squared ofte blir høy når vi gjør dette.

```
fe <- lm(trust_politicians ~ gender + age + income_decile + factor(income_feel) + factor(country),
        data = ess)
summary(fe)
```

```
##
## Call:
## lm(formula = trust_politicians ~ gender + age + income_decile +
##      factor(income_feel) + factor(country), data = ess)
##
## Residuals:
## <Labelled double>
##      Min       1Q   Median       3Q      Max
## -5.6770 -1.4818  0.0422  1.4955  9.3317
##
## Labels:
##  value      label
##    0 No trust at all
##    1              1
##    2              2
##    3              3
##    4              4
##    5              5
##    6              6
##    7              7
##    8              8
##    9              9
##   10 Complete trust
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.7398851   0.0946548   39.511 < 2e-16 ***
## gendermale    -0.0239495   0.0304023   -0.788  0.4309
## age          -0.0021299   0.0008782   -2.425  0.0153 *
```

```
## income_decile          0.0411008  0.0065285   6.296 3.13e-10 ***
## factor(income_feel)2   -0.4126915  0.0369196 -11.178 < 2e-16 ***
## factor(income_feel)3   -0.7837628  0.0572979 -13.679 < 2e-16 ***
## factor(income_feel)4   -1.4178480  0.0988744 -14.340 < 2e-16 ***
## factor(country)Belgium  0.6188830  0.0857380   7.218 5.47e-13 ***
## factor(country)Czechia -0.0275912  0.0882235  -0.313  0.7545
## factor(country)Denmark  1.2636160  0.0876844  14.411 < 2e-16 ***
## factor(country)Finland  1.1220931  0.0816979  13.735 < 2e-16 ***
## factor(country)France   -0.7205740  0.0835083  -8.629 < 2e-16 ***
## factor(country)Germany  0.3207948  0.0768653   4.173 3.01e-05 ***
## factor(country)Ireland  -0.0323463  0.0848401  -0.381  0.7030
## factor(country)Netherlands 1.3022529  0.0848780  15.343 < 2e-16 ***
## factor(country)Norway   1.5883521  0.0877161  18.108 < 2e-16 ***
## factor(country)Poland   -1.3450449  0.0942341 -14.273 < 2e-16 ***
## factor(country)Slovenia -1.5644311  0.0975438 -16.038 < 2e-16 ***
## factor(country)Sweden   1.2824515  0.0846265  15.154 < 2e-16 ***
## factor(country)Switzerland 1.6069388  0.0896376  17.927 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.072 on 18688 degrees of freedom
## (9513 observations deleted due to missingness)
## Multiple R-squared:  0.2103, Adjusted R-squared:  0.2095
## F-statistic: 261.9 on 19 and 18688 DF, p-value: < 2.2e-16
```

Husk også:

<i>income_feel</i> beskrivelse	Verdi
Living comfortably on present income	1
Coping on present income	2
Difficult on present income	3
Very difficult on present income	4

Ganske tydelig her at det er forskjeller mellom de forskjellige landene. La oss prøve oss på en veldig enkel flernivå her. Vi skal komme tilbake til det neste gang også.

De fleste behov innenfor flernivåanalyse kan tilfredsstilles med pakken **lme4**. OLS flernivå kan kjøres med funksjonen `lmer()`. Syntaksen for denne funksjonen er også veldig lik alle andre regresjonsanalyser, med ett viktig unntak. Legg merke til leddet `(1|country)` under: her sier vi at vi har hierarkisk struktur i data, der *country* er nivå 2 enheter. Dette er altså bare en *random intercept* modell.

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'lme4'

## The following object is masked from 'package:stats':
##
##      sigma

trust_polit <- lmer(trust_politicians ~ age + gender + income_feel + income_decile + (1|country),
                    data = ess)
summary(trust_polit)

## Linear mixed model fit by REML ['lmerMod']
## Formula: trust_politicians ~ age + gender + income_feel + income_decile +
##      (1 | country)
##      Data: ess
##
## REML criterion at convergence: 80456.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.7395 -0.7118  0.0212  0.7239  4.4259
##
## Random effects:
##      Groups   Name      Variance Std.Dev.
##   country (Intercept) 1.103    1.050
##   Residual          4.293    2.072
## Number of obs: 18708, groups:  country, 14
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   4.5653970  0.2941614  15.520
## age           -0.0021387  0.0008782  -2.435
## gendermale    -0.0239065  0.0304024  -0.786
## income_feel   -0.4250526  0.0239053 -17.781
## income_decile  0.0404821  0.0065170   6.212
##
## Correlation of Fixed Effects:
##              (Intr) age    gndrml incm_f
## age           -0.180
## gendermale    -0.044 -0.023
## income_feel   -0.218  0.096  0.008
## income_decl   -0.215  0.173 -0.053  0.484
```

Andre spesifikasjoner:

```

# lmer(AV ~ 1 + (1 | Gruppe_nivå2), data = data)
# = random intercept only

# lmer(AV ~ UV_nivå1 + (1 | Gruppe_nivå2), data = data)
# = random intercept plus fixed effect

# lmer(AV ~ UV_nivå1 + (UV_nivå1 | Gruppe_nivå2), data = data)
# = random intercept, random slope

# lmer(AV ~ UV_nivå1 + UV_nivå2 + (1 + UV_nivå1 | Gruppe_nivå2), data = data)
# = random intercept, individual and group predictor

# lmer(AV ~ UV_nivå1 * UV_nivå2 + (1 + UV_nivå1 | Gruppe_nivå2), data = data)
# = random intercept, cross-level interaction

library(lme4)

trust_polit2 <- lmer(trust_politicians ~ age + gender + income_feel + income_decile +
  (income_feel|country),
  data = ess) # random intercept, random slope
summary(trust_polit2)

## Linear mixed model fit by REML ['lmerMod']
## Formula: trust_politicians ~ age + gender + income_feel + income_decile +
## (income_feel | country)
## Data: ess
##
## REML criterion at convergence: 80441.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.7389 -0.7163  0.0157  0.7309  4.2182
##
## Random effects:
##   Groups   Name      Variance Std.Dev. Corr
##   country (Intercept) 1.31295  1.1458
##           income_feel 0.01532  0.1238  -0.48
## Residual                4.28594  2.0703
## Number of obs: 18708, groups:  country, 14
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   4.543332   0.318757  14.253

```

```
## age          -0.002288  0.000880  -2.600
## gendermale   -0.022150  0.030396  -0.729
## income_feel  -0.409760  0.041221  -9.940
## income_decile 0.040375  0.006527   6.186
##
## Correlation of Fixed Effects:
##          (Intr) age    gndrml incm_f
## age          -0.165
## gendermale   -0.041 -0.023
## income_feel  -0.490  0.053  0.004
## income_decl -0.199  0.174 -0.053  0.283
```

Vi kan så hente ut random effekter og konstantledd for alle land med funksjonen `coef()/ranef()`, plote konstantleddene med `dotplot()` fra pakken `lattice`, etc.

```
coef(trust_polit2)
```

```
## $country
##          (Intercept)          age  gendermale income_feel income_decile
## Austria          3.986733 -0.002288301 -0.02215007  -0.3175013   0.04037456
## Belgium          5.101886 -0.002288301 -0.02215007  -0.5828189   0.04037456
## Czechia          3.944041 -0.002288301 -0.02215007  -0.3284894   0.04037456
## Denmark          5.564742 -0.002288301 -0.02215007  -0.5133032   0.04037456
## Finland          5.447022 -0.002288301 -0.02215007  -0.4977879   0.04037456
## France           3.506165 -0.002288301 -0.02215007  -0.4433574   0.04037456
## Germany          4.519658 -0.002288301 -0.02215007  -0.4333807   0.04037456
## Ireland          4.218391 -0.002288301 -0.02215007  -0.4572423   0.04037456
## Netherlands      5.669750 -0.002288301 -0.02215007  -0.5390614   0.04037456
## Norway           5.731659 -0.002288301 -0.02215007  -0.4018351   0.04037456
## Poland           2.734074 -0.002288301 -0.02215007  -0.3672752   0.04037456
## Slovenia         2.269234 -0.002288301 -0.02215007  -0.2264686   0.04037456
## Sweden           5.359384 -0.002288301 -0.02215007  -0.3544172   0.04037456
## Switzerland     5.553910 -0.002288301 -0.02215007  -0.2736997   0.04037456
##
## attr(,"class")
## [1] "coef.mer"
```

```
ranef(trust_polit2)
```

```
## $country
##          (Intercept)  income_feel
## Austria      -0.55659917  0.092258575
## Belgium       0.55855443 -0.173059021
## Czechia      -0.59929125  0.081270462
## Denmark       1.02140970 -0.103543326
## Finland       0.90368965 -0.088028046
```

```
## France      -1.03716702 -0.033597486
## Germany     -0.02367388 -0.023620857
## Ireland     -0.32494121 -0.047482368
## Netherlands  1.12641837 -0.129301471
## Norway      1.18832667  0.007924751
## Poland      -1.80925822  0.042484660
## Slovenia    -2.27409833  0.183291247
## Sweden       0.81605192  0.055342733
## Switzerland 1.01057833  0.136060147
```

```
lattice::dotplot(ranef(trust_polit2, condVar = TRUE))
```

```
## $country
```

