

Seminar 5

Martin Søyland

Laste inn data

Vi fortsetter der vi slapp sist! Altså med European Social Survey data.

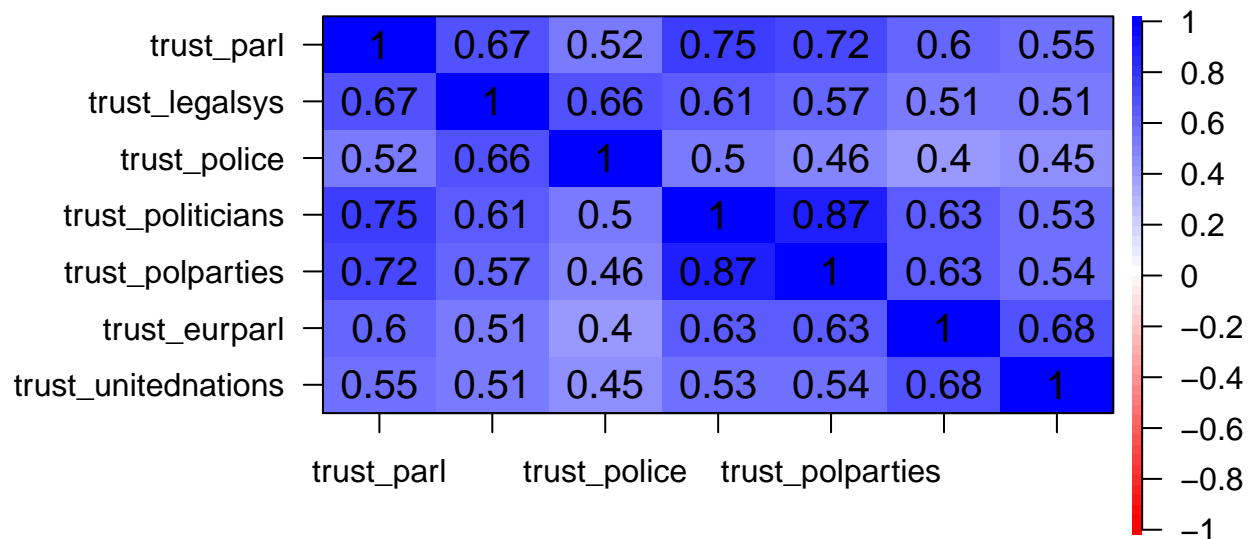
```
rm(list = ls())  
load("./data/ess.rda")
```

Faktoranalyse

Skal ikke skrive så mye om den substansielle tolkningen av faktoranalyse, Christophersen er mye ryddigere på dette. Så les pensum!

La oss tenke oss at våre tillitsvariabler **sammen** måler et (eller flere) underliggende konsepter – for eksempel politisk tillit. Det første vi vil gjøre er å bare sjekke korrelasjonen mellom disse variablene. Jeg bruker `cor.plot()` for å visualisere kjapt hvordan korrelasjonsmatrisen ser ut. Vi kunne brukt `ggplot()` også her, men det tar litt lenger tid (når dere skriver oppgaver: pass på å gjøre figurer/tabeller finere enn dette).

```
korrel <- cor(ess[, c("trust_parl", "trust_legalsys", "trust_police",  
                     "trust_politicians", "trust_polparties", "trust_eurparl",  
                     "trust_unitednations")], use = "complete.obs")  
  
library(psych)  
  
cor.plot(korrel, numbers = TRUE)
```



```
# Et triks når det er et mønster i variablene vi skal ha:
# cor(ess[, c(names(ess)[which(grepl("trust_", names(ess)))])], use = "complete.obs")
```

Korrelasjonene er altså ganske høye mellom alle variablene – de laveste er rundt 0.40. La oss kjøre på med faktoranalysene. Vi gjør først en prinsipal komponentanalyse. Denne er *eksplorerende*, i at vi ikke bestemmer på forhånd hvor mange faktorer variablene deler seg inn i. Funksjonen vi bruker er `princomp()`, som kan spesifiseres på flere måter. Jeg har lagt opp en formel som sier at vi skal bruke alle variablene i `data`-argumentet (`ess`) – her trekker jeg (som i korrelasjonene over) ut bare *trust*-variablene. Jeg tar også vare på scores.

Ladningene vist under, viser ikke veldig sterke mønster (for meg hvertfall). Legg merke til at denne type faktoranalyse går fra 1 faktor til antall variabler du putter inn (her 7).

Ser vi på screeplotet derimot, er det tydelig at én faktor forklarer mye. Likevel kan vi ikke ignorere faktor 2 og 3 heller. Hvordan kan det ha seg?

```
trust_prin <- princomp(~., ess[, c("trust_parl", "trust_legalsys", "trust_police",
                                   "trust_politicians", "trust_polparties",
                                   "trust_eurparl", "trust_unitednations")],
                      scores = TRUE)
```

```
loadings(trust_prin)
```

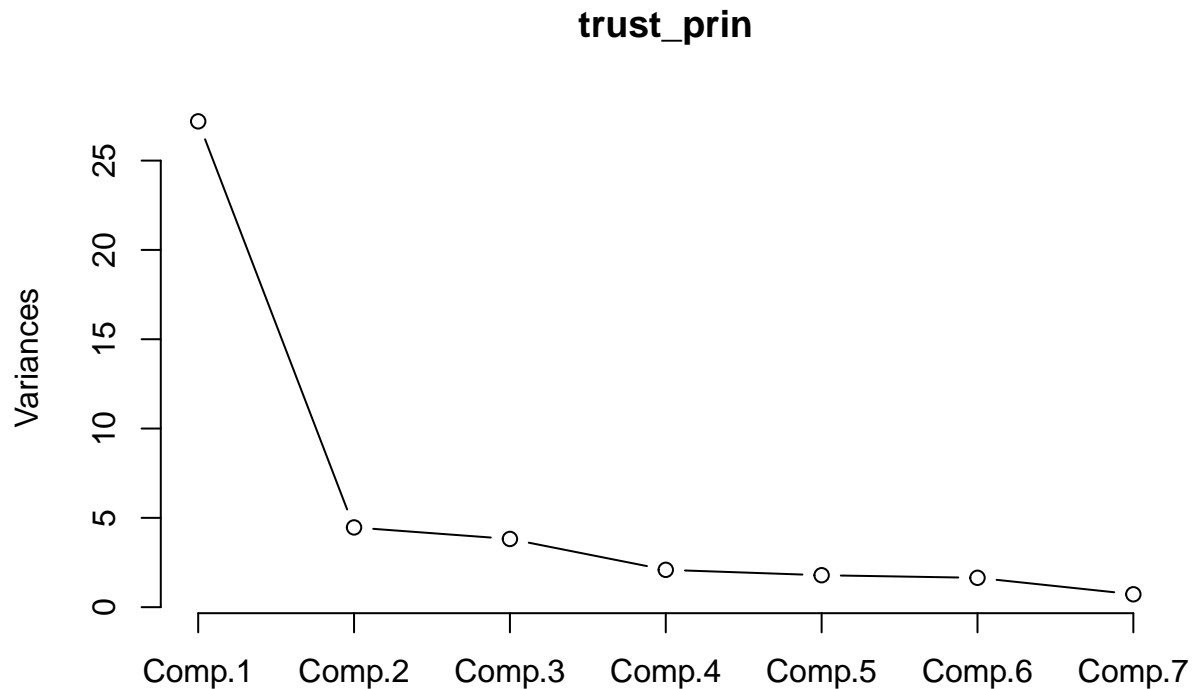
```
##
## Loadings:
##
```

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7
## trust_parl	-0.424		0.296	0.502	0.387	0.571	
## trust_legalsys	-0.400	0.520	-0.114	0.477	-0.366	-0.442	
## trust_police	-0.306	0.601	-0.265	-0.597		0.331	
## trust_politicians	-0.397	-0.124	0.424	-0.257		-0.211	0.732
## trust_polparties	-0.384	-0.185	0.417	-0.309		-0.301	-0.677
## trust_eurparl	-0.369	-0.454	-0.282		-0.659	0.379	
## trust_unitednations	-0.355	-0.335	-0.629		0.523	-0.302	

```
##
##
```

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7
## SS loadings	1.000	1.000	1.000	1.000	1.000	1.000	1.000
## Proportion Var	0.143	0.143	0.143	0.143	0.143	0.143	0.143
## Cumulative Var	0.143	0.286	0.429	0.571	0.714	0.857	1.000

```
screeplot(trust_prin, type = "lines")
```



La oss kjøre en faktoranalyse der vi setter antall faktorer til 3. Vi kan ha en ganske tydelig, men uhøytidelig, teoretisk antagelse om hvorfor våre variabler deler seg i tre faktorer: en faktor handler om nasjonal politikk (parlament, politiske partier, politikere), en handler om rettsvesen (rettssystem, politi) og en handler om internasjonale institusjoner (EU, FN). Her bruker vi funksjonen `factanal()` som er satt opp ganske likt som `princomp()`. Forskjellen er at vi setter akkurat hvor mange faktorer vi skal ha i analysen (3).

```
trust_factor <- factanal(~., 3, ess[, c("trust_parl", "trust_legalsys", "trust_police",
                                       "trust_politicians", "trust_polparties",
                                       "trust_eurparl", "trust_unitednations")])

print(loadings(trust_factor), cutoff = .5)
```

```
##
## Loadings:
##           Factor1 Factor2 Factor3
## trust_parl      0.593
## trust_legalsys           0.861
## trust_police           0.604
## trust_politicians  0.845
## trust_polparties  0.813
## trust_eurparl                0.602
## trust_unitednations           0.828
##
##           Factor1 Factor2 Factor3
## SS loadings    2.154   1.671   1.442
```

```
## Proportion Var    0.308    0.239    0.206
## Cumulative Var    0.308    0.546    0.752
```

Her er alle ladninger lavere enn 0.5 sjult. Vi ser at den teoretiske antagelsen vår ser ganske rimelig ut. Magi!

Nedefor er to eksempler på rotasjon. Dette kan dere bruke om dere skulle trenge det til oppgaven

```
varimax(loadings(trust_factor), normalize = TRUE)
```

```
## $loadings
##
## Loadings:
##               Factor1 Factor2 Factor3
## trust_parl      0.593   0.471   0.325
## trust_legalsys   0.306   0.861   0.239
## trust_police     0.264   0.604   0.250
## trust_politicians 0.845   0.330   0.277
## trust_polparties  0.813   0.292   0.305
## trust_eurparl    0.450   0.262   0.603
## trust_unitednations 0.248   0.282   0.828
##
##               Factor1 Factor2 Factor3
## SS loadings      2.154   1.670   1.443
## Proportion Var    0.308   0.239   0.206
## Cumulative Var    0.308   0.546   0.752
##
## $rotmat
##               [,1]      [,2]      [,3]
## [1,]  1.000000e+00 -0.0001547644 4.355716e-05
## [2,]  1.547545e-04  0.9999999622 2.271306e-04
## [3,] -4.359231e-05 -0.0002271238 1.000000e+00
```

```
promax(loadings(trust_factor))
```

```
## $loadings
##
## Loadings:
##               Factor1 Factor2 Factor3
## trust_parl      0.534   0.291
## trust_legalsys           0.999
## trust_police           0.650
## trust_politicians  1.005
## trust_polparties   0.960
## trust_eurparl      0.285           0.594
## trust_unitednations -0.136           1.006
##
```

```
##               Factor1 Factor2 Factor3
## SS loadings    2.318    1.510    1.381
## Proportion Var  0.331    0.216    0.197
## Cumulative Var  0.331    0.547    0.744
##
## $rotmat
##           [,1]      [,2]      [,3]
## [1,]  1.5218255 -0.4399056 -0.4543902
## [2,] -0.4647856  1.4155118 -0.2809153
## [3,] -0.4614796 -0.3549989  1.4473241
```

Det siste som da gjenstår, er å lage våre tre faktorer om til en additiv indeks – det er også mulig å vekte faktorene forskjellig. Her summerer vi rett fram veridene for variablene på hver faktor og deler på antall variabler.

```
ess$political_trust <- (ess$trust_parl + ess$trust_politicians + ess$trust_polparties) / 3
ess$legal_trust <- (ess$trust_legalsys + ess$trust_police) / 2
ess$international_trust <- (ess$trust_unitednations + ess$trust_eurparl) / 2
```

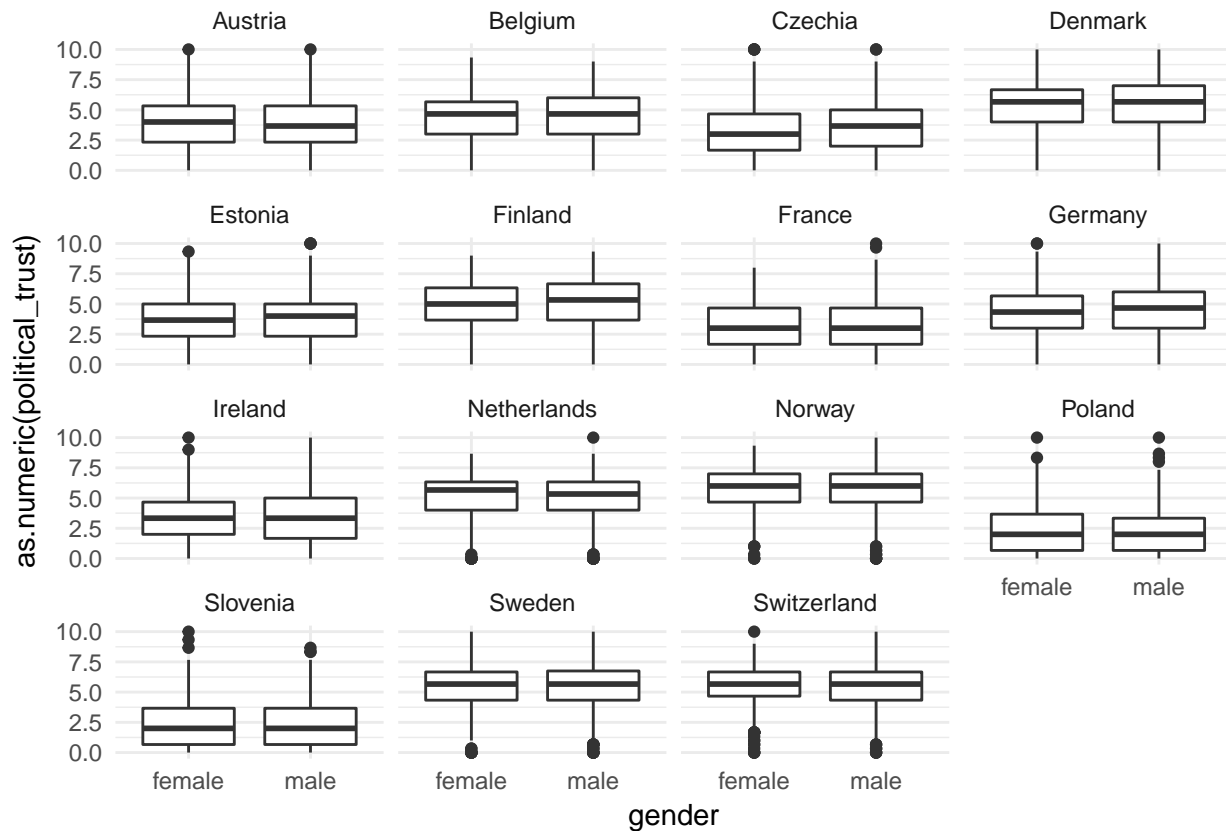
```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
## The following objects are masked from 'package:psych':
##
##      %+%, alpha
```

```
plot_data <- ess[which(is.na(ess$gender) == FALSE), ]

ggplot(plot_data, aes(x = gender, y = as.numeric(political_trust))) +
  geom_boxplot() +
  facet_wrap(~country) +
  theme_minimal()
```

```
## Warning: Removed 520 rows containing non-finite values (stat_boxplot).
```



Litt mer flernivåanalyse

I denne seksjonen går vi tilbake til litt flernivåanalyse. Her bruker vi en av våre konstruerte indekser: *political_trust*. Jeg lager et eget datasett der jeg fjerner enhetene som har NA på en eller flere av variablene vi skal bruke, og drar ut bare variablene vi skal bruke. Dette er hovedsakelig fordi det er en enkel måte å teste model fit på med sekvensiell analyse.

Først estimerer jeg en null-modell, med bare random intercept mellom land. Deretter legger jeg inn variablene sekvensielt.

```
library(lme4)

## Loading required package: Matrix
##
## Attaching package: 'lme4'
## The following object is masked from 'package:stats':
##
##      sigma

reg_data <- na.omit(ess[, c("political_trust", "income_feel", "income_decile",
                           "gender", "age", "country")])
```

```
trust_polit0 <- lmer(political_trust ~ (1|country),
                    data = reg_data)
summary(trust_polit0)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: political_trust ~ (1 | country)
## Data: reg_data
##
## REML criterion at convergence: 76915.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.9772 -0.6766  0.0757  0.7147  3.9957
##
## Random effects:
## Groups Name Variance Std.Dev.
## country (Intercept) 1.377 1.174
## Residual 3.743 1.935
## Number of obs: 18479, groups: country, 14
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 4.232 0.314 13.48
```

ICC

```
1.377 / (1.377 + 3.743)
```

```
## [1] 0.2689453
```

```
trust_polit1 <- lmer(political_trust ~ income_feel + (1|country),
                    data = reg_data)
```

```
trust_polit2 <- lmer(political_trust ~ income_feel + income_decile + (1|country),
                    data = reg_data)
```

```
trust_polit3 <- lmer(political_trust ~ income_feel + income_decile + age + gender +
                    (1|country),
                    data = reg_data)
```

```
trust_polit4 <- lmer(political_trust ~ income_feel + income_decile + age + gender +
                    (gender|country),
                    data = reg_data)
```

To mål på model-fit, som straffer at vi legger inn flere variabler

```
# AIC(trust_polit0, trust_polit1, trust_polit2, trust_polit3, trust_polit4)
```

```
# BIC(trust_polit0, trust_polit1, trust_polit2, trust_polit3, trust_polit4)
```

```
##
```

```
## Please cite as:
```

```
## Hlavac, Marek (2015). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
## R package version 5.2. http://CRAN.R-project.org/package=stargazer
```

Table 1:

	<i>Dependent variable:</i>				
	political_trust				
	(1)	(2)	(3)	(4)	(5)
income_feel		−0.508*** (0.019)	−0.409*** (0.022)	−0.418*** (0.022)	−0.418*** (0.022)
income_decile			0.056*** (0.006)	0.051*** (0.006)	0.051*** (0.006)
age				−0.004*** (0.001)	−0.004*** (0.001)
gendermale				0.048* (0.028)	0.038 (0.044)
Constant	4.233*** (0.314)	5.133*** (0.289)	4.647*** (0.294)	4.847*** (0.299)	4.851*** (0.301)
Observations	18,479	18,479	18,479	18,479	18,479
Log Likelihood	−38,457.530	−38,121.930	−38,081.600	−38,078.850	−38,075.310
Akaike Inf. Crit.	76,921.070	76,251.860	76,173.190	76,171.710	76,168.610
Bayesian Inf. Crit.	76,944.540	76,283.160	76,212.310	76,226.480	76,239.030

Note:

*p<0.1; **p<0.05; ***p<0.01

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

```
lattice::dotplot(ranef(trust_polit4, condVar = TRUE))
```

```
## $country
```