

Seminar 4

Martin Søyland

Disposisjon

1. Laste inn data (.rda / .RData)
2. Omkoding #advanced
3. Multinomisk logistisk regresjon
4. Rangert logistisk regresjon

Laste inn data

Denne gangen skal vi laste inn et datasett lagret i en R-fil. Dette er litt annerledes enn funksjonene vi har gjort tidligere (altså read.*()-familien). Det finnes to typer filer som R-data kan lagres i: *.rda* og *.RData*. Disse er, såvidt jeg vet, helt identiske i format; de har bare forskjellig filtypenavn. Vi skal bruke European Social Survey, men bare med enhetene fra Norge (kommer tilbake med flere land når vi skal ta flernivåanalyse).

```
load("./data/ess_norge.rda")
head(ess_nor, 3)
```

```
##   idno                party_vote party_vote_short gender year_born
## 1 6306          Progress Party (FRP)          FRP   male    1971
## 2 6312             Labour Party (A)              A  female   1968
## 3 6318 Christian Democratic Party (KRF)          KRF  female   1977
##   income_feel income_decile trust_parl trust_legalsys trust_police
## 1           1             6           8             6           9
## 2           3             2           6             9          10
## 3           1             2           5            10           9
##   trust_politicians trust_polparties trust_eurparl trust_unitednations
## 1                 6                 6                 7                 9
## 2                 6                 5                 5                 8
## 3                 5                 5                 7                 9
```

Her oppretter R objektet for oss (se “ess_nor” i environment). Analysen vi skal gjøre er å se om tillit til politikere påvirker hvilket parti man stemmer på. Avhengig variabel blir derfor variabelen **party_vote_short**. Kan dere gjette på hvilken type regresjon vi skal gjøre?

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?

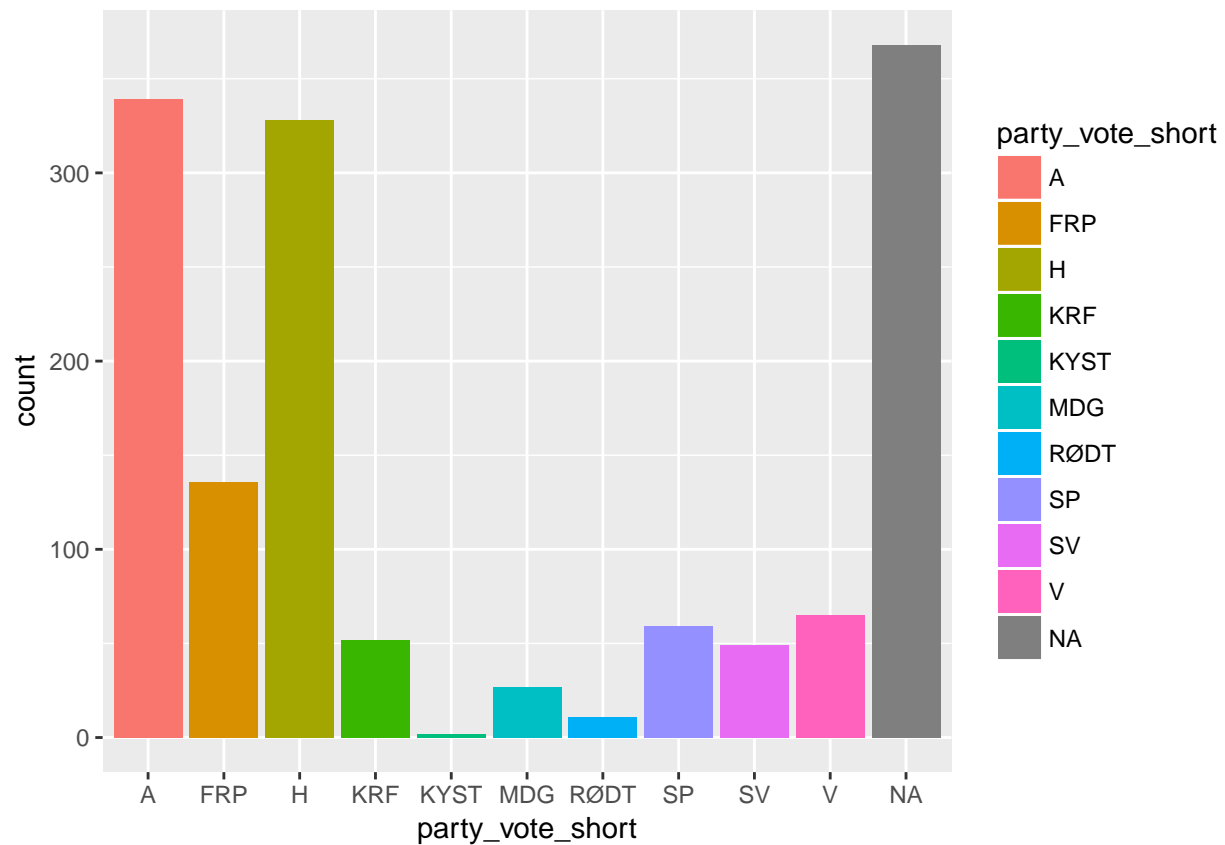
Avhengig variabel

La oss først kikke litt på vår avhengige variabel. Dette kan man gjøre med både tabeller og figurer, men jeg liker best figurer.

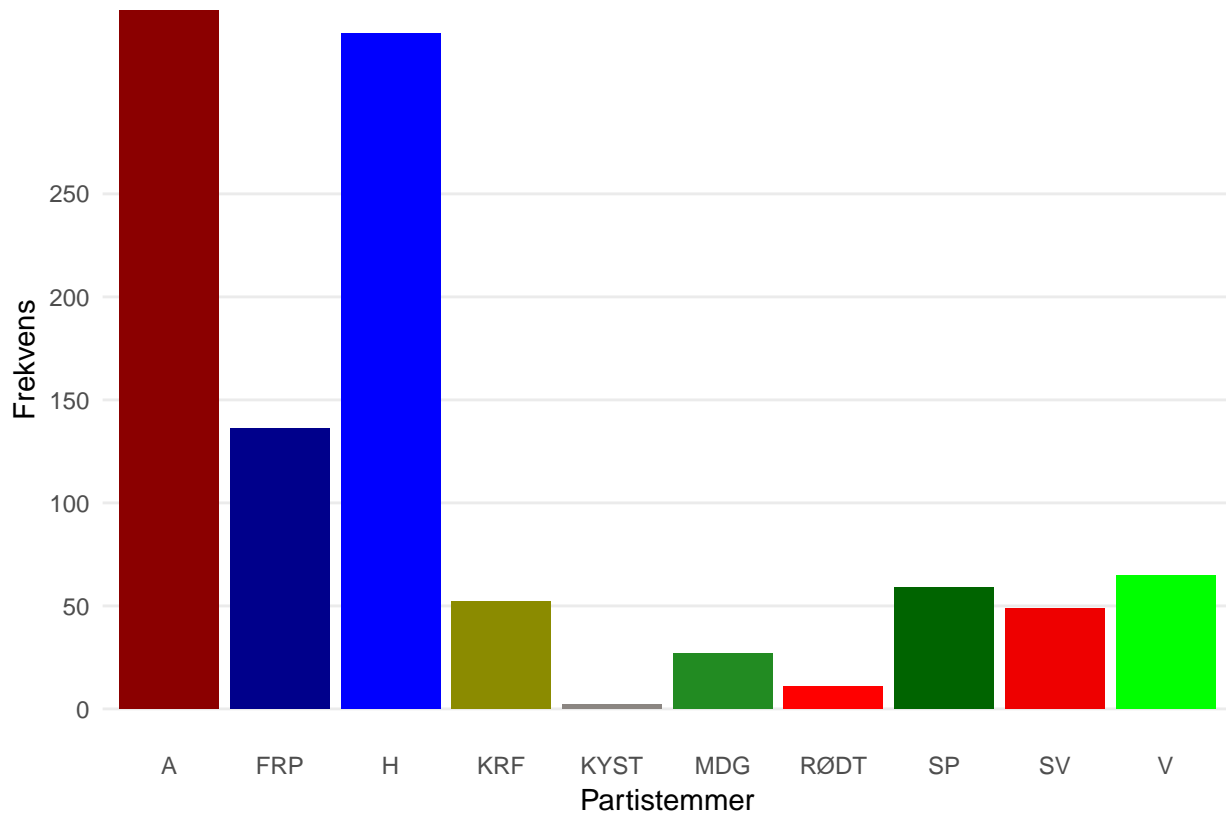
```
table(ess_nor$party_vote_short, useNA = "always")
```

```
##
##      A  FRP      H  KRF KYST  MDG RØDT   SP   SV    V <NA>
##  339  136  328   52    2   27   11   59   49   65  368
```

```
library(ggplot2)
ggplot(ess_nor, aes(x = party_vote_short, fill = party_vote_short)) +
  geom_bar()
```



```
ggplot(ess_nor[which(is.na(ess_nor$party_vote_short) == FALSE), ],
  aes(x = party_vote_short, fill = party_vote_short)) +
  geom_bar() +
  scale_fill_manual(values = c("darkred", "darkblue", "blue", "yellow4", "seashell4",
    "forestgreen", "red1", "darkgreen", "red2", "green")) +
  labs(x = "Partistemma", y = "Frekvens", fill = "Parti") +
  scale_y_continuous(breaks = seq(0, 260, 50)) +
  theme_minimal() +
  theme(panel.grid.minor = element_blank(),
    panel.grid.major.x = element_blank(),
    legend.position = "none")
```



Her har jeg lagt inn et plot som viser antall enheter i data som stemte på de forskjellige partiene og et plot der jeg fikser litt på det estetiske. **ggplot** har en milliard måter å endre på plots på, så her er det bare å leke seg til perfektion.

For selve innholdet i plottet kan vi merke oss at vi har noen partier som er veldig små. Dette gjelder spesielt partiene *Rødt*, *MDG*, og *Kystpartiet*.

Uavhengig variabel

Variabelen vi skal fokusere på er **trust_politicians**. Derfor er det lurt å også kikke litt på denne. Fra tabellen ser vi at den går fra 0 til 10, men hva betyr egentlig det?

```
table(ess_nor$trust_politicians, useNA = "always")
```

```
##
##    0    1    2    3    4    5    6    7    8    9   10 <NA>
##   31   29   79  114  175  310  289  256  115   20   11    7
```

```
attributes(ess_nor$trust_politicians)
```

```
## $labels
## No trust at all          1          2          3
##              0          1          2          3
##              4          5          6          7
##              4          5          6          7
```

```
##           8           9 Complete trust
##           8           9           10
##
## $class
## [1] "labelled"
##
## $label
## [1] "Trust in politicians"
```

```
attr(ess_nor$trust_politicians, "labels")
```

```
## No trust at all           1           2           3
##           0           1           2           3
##           4           5           6           7
##           4           5           6           7
##           8           9 Complete trust
##           8           9           10
```

Da vet vi at 0 er ingen tillit og 10 er fullstendig tillit til politikere på denne variabelen. Kan den brukes som en forholdstallsvariabel? Eller burde vi holde oss til ordinalnivå?

Bivariat multinomisk logistisk regresjon

La oss kjøre en bivariat regresjon, med partistemme som avhengig og tillit til politikere som uavhengig, for å se om det er noen problemer med data. Hvis vi ikke gjør noe med vår avhengig variabel, hvilket parti vil ende opp som referansekategori og er dette en rimelig referansekategori?

Multinomisk logistisk regresjon kan kjøres med noen forskjellige pakker i R (det ligger faktisk ikke inne i base-pakken). Vi kjøre med pakken **nnet** her, og funksjonen `multinom()`.

```
# install.packages("nnet")
library(nnet)

party_reg <- multinom(party_vote_short ~ trust_politicians, data = ess_nor,
                      na.action = "na.exclude", Hess = TRUE)
```

```
## # weights: 30 (18 variable)
## initial value 2456.858294
## iter 10 value 2040.810326
## iter 20 value 1854.994218
## final value 1850.496475
## converged
```

```
summary(party_reg)
```

```
## Call:
## multinom(formula = party_vote_short ~ trust_politicians, data = ess_nor,
```

```
##      na.action = "na.exclude", Hess = TRUE)
##
## Coefficients:
##      (Intercept) trust_politicians
## FRP      0.7870249      -0.34755333
## H      0.2910194      -0.06012313
## KRF     -2.0076255      0.02406672
## KYST    -6.1537874      0.17776420
## MDG     -1.9223488      -0.11467751
## RØDT    -3.2210641      -0.03816324
## SP      -1.3087091      -0.08214659
## SV      -2.4546861      0.08897213
## V       -1.9493480      0.05349526
##
## Std. Errors:
##      (Intercept) trust_politicians
## FRP      0.2788468      0.05505258
## H      0.2470196      0.04351956
## KRF      0.4961181      0.08529714
## KYST     2.6838789      0.43178272
## MDG      0.5947308      0.10892014
## RØDT     0.9695161      0.17137182
## SP       0.4338177      0.07795266
## SV       0.5399175      0.09046729
## V       0.4607200      0.07834122
##
## Residual Deviance: 3700.993
## AIC: 3736.993

# Regner litt på Arbeiderpartiet vs. Fremskrittspartiet
exp(0.7870249) / (1 + exp(0.7870249))

## [1] 0.6871922

exp(0.7870249 + (-0.34755333 * 10)) / (1 + exp(0.7870249 + (-0.34755333 * 10)))

## [1] 0.06365486

# En enkel måte å sjekke om effekten er signifikant på 5% nivå
confint(party_reg)

## , , FRP
##
##      2.5 %      97.5 %
## (Intercept)      0.2404953  1.3335545
## trust_politicians -0.4554544 -0.2396523
```

```

##
## , , H
##
##          2.5 %      97.5 %
## (Intercept)      -0.1931301  0.77516895
## trust_politicians -0.1454199  0.02517363
##
## , , KRF
##
##          2.5 %      97.5 %
## (Intercept)      -2.9799991 -1.035252
## trust_politicians -0.1431126  0.191246
##
## , , KYST
##
##          2.5 %      97.5 %
## (Intercept)     -11.4140933 -0.8934815
## trust_politicians -0.6685144  1.0240428
##
## , , MDG
##
##          2.5 %      97.5 %
## (Intercept)     -3.0879998 -0.75669779
## trust_politicians -0.3281571  0.09880205
##
## , , RØDT
##
##          2.5 %      97.5 %
## (Intercept)     -5.1212808 -1.3208475
## trust_politicians -0.3740458  0.2977193
##
## , , SP
##
##          2.5 %      97.5 %
## (Intercept)     -2.158976 -0.45844201
## trust_politicians -0.234931  0.07063782
##
## , , SV
##
##          2.5 %      97.5 %
## (Intercept)     -3.51290495 -1.3964673
## trust_politicians -0.08834051  0.2662848
##

```

```
## , , V
##
##                2.5 %    97.5 %
## (Intercept)      -2.8523426 -1.0463534
## trust_politicians -0.1000507  0.2070412

# Vi kan også se på hvordan modellen tenker med tanke på sannsynligheter
test_set <- data.frame(trust_politicians = 0:10)

predict(party_reg, newdata = test_set)
```

```
## [1] FRP FRP H   H   H   A   A   A   A   A   A
## Levels: A FRP H KRF KYST MDG RØDT SP SV V
```

```
predict(party_reg, newdata = test_set, type = "probs")
```

```
##           A           FRP           H           KRF           KYST           MDG
## 1  0.1867178 0.41019121 0.2497893 0.02507755 0.0003968532 0.02730991
## 2  0.2167551 0.33637939 0.2730526 0.02982089 0.0005503201 0.02826832
## 3  0.2456989 0.26935359 0.2914533 0.03462634 0.0007451630 0.02857130
## 4  0.2725412 0.21106256 0.3044295 0.03934482 0.0009873752 0.02825890
## 5  0.2966253 0.16227323 0.3119980 0.04386476 0.0012836907 0.02742380
## 6  0.3176445 0.12275508 0.3146109 0.04811726 0.0016420844 0.02618526
## 7  0.3355689 0.09160926 0.3129702 0.05207069 0.0020722296 0.02466570
## 8  0.3505486 0.06760290 0.3078636 0.05572010 0.0025858681 0.02297503
## 9  0.3628242 0.04942798 0.3000511 0.05907611 0.0031971011 0.02120314
## 10 0.3726622 0.03586340 0.2902038 0.06215600 0.0039226314 0.01941850
## 11 0.3803155 0.02585471 0.2788822 0.06497762 0.0047819850 0.01767016
##           RØDT           SP           SV           V
## 1  0.007452387 0.05044529 0.01603722 0.02658243
## 2  0.008327312 0.05394216 0.02034945 0.03255449
## 3  0.009085830 0.05632308 0.02521313 0.03892938
## 4  0.009701066 0.05754923 0.03057002 0.04555532
## 5  0.010162991 0.05769523 0.03636738 0.05230557
## 6  0.010475639 0.05691113 0.04256819 0.05908997
## 7  0.010652385 0.05538112 0.04915478 0.06585471
## 8  0.010711227 0.05329082 0.05612706 0.07257482
## 9  0.010671197 0.05080712 0.06349805 0.07924405
## 10 0.010550139 0.04806931 0.07128852 0.08586546
## 11 0.010363651 0.04518776 0.07952221 0.09244425
```

Det er ganske åpenbart at modellen vår ikke er veldig god. Noen problemer: 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

Først oppretter vi et nytt datasett der vi fjerner enheter fra de små partiene. Legg merke til at SV, KRF, SP, og V også er veldig nær smertegrensen her; vi kan få problemer med disse også.

```
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

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$age <- larger_parties$age - median(larger_parties$age) # ingen har NA
summary(larger_parties$age)
```

```
##      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.

```
party_reg2 <- multinom(party_vote_short ~ trust_politicians + income_decile2 +
                      income_feel2 + age + 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 + gender, data = larger_parties, na.action = "na.exclude",
##      Hess = TRUE)
##
## Coefficients:
##      (Intercept) trust_politicians income_decile2 income_feel2      age
## 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
## 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 -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             -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             -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             -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             -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               -0.03726125 -0.002138614
## gendermale        -0.82418092  0.290594866
```

Dette er ekstremt god trening for å tolke på egenhånd!

```
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 = 0,
                        gender = "female")
```

```
predict(party_reg2, newdata = test_set2)
```

```
## [1] H H H H A A A A A A
## Levels: A FRP H KRF SP SV V
```

```
predict(party_reg2, newdata = test_set2, type = "probs")
```

```
##           A           FRP           H           KRF           SP           SV
## 1  0.2552720 0.25189014 0.3409532 0.02416958 0.05023893 0.03282244
## 2  0.2828051 0.20148769 0.3467304 0.02730367 0.05159113 0.03829641
## 3  0.3084808 0.15868751 0.3471730 0.03036896 0.05216350 0.04399489
## 4  0.3319519 0.12329433 0.3429305 0.03332306 0.05203127 0.04986002
## 5  0.3530615 0.09468291 0.3348069 0.03613999 0.05129681 0.05585098
## 6  0.3717960 0.07199117 0.3236397 0.03880702 0.05007204 0.06194244
## 7  0.3882353 0.05427788 0.3102168 0.04132078 0.04846590 0.06812112
## 8  0.4025111 0.04063113 0.2952304 0.04368367 0.04657681 0.07438198
## 9  0.4147757 0.03023066 0.2792606 0.04590103 0.04448932 0.08072468
## 10 0.4251828 0.02237501 0.2627756 0.04797917 0.04227356 0.08715090
## 11 0.4338767 0.01648570 0.2461438 0.04992423 0.03998621 0.09366250
##           V
## 1  0.04465376
## 2  0.05178565
## 3  0.05913134
## 4  0.06660888
## 5  0.07416084
## 6  0.08175161
## 7  0.08936222
## 8  0.09698489
## 9  0.10461809
## 10 0.11226297
## 11 0.11992082
```

Rangert logistisk (for spesielt interesserte)

Under viser jeg kode for rangert logistisk regresjon. Som Solveig nevnte i forelesning, er dette noe som kan dukke opp som en mulig modell i en evt semester- eller masteroppgave.

```
larger_parties$party_vote_short <- factor(larger_parties$party_vote_short,
                                          levels = c("SV", "A", "SP", "KRF", "V", "H", "FRP"))

library(MASS)
rang_party <- polr(party_vote_short ~ trust_politicians + income_decile2 + income_feel2 + age + gender,
                  data = larger_parties, Hess = TRUE, na.action = "na.exclude")
summary(rang_party)

## Call:
## polr(formula = party_vote_short ~ trust_politicians + income_decile2 +
##       income_feel2 + age + gender, data = larger_parties, na.action = "na.exclude",
##       Hess = TRUE)
##
## Coefficients:
##               Value Std. Error  t value
## trust_politicians -0.1756222   0.034223 -5.13176
## income_decile2     0.0010090   0.023891  0.04223
## income_feel2       0.1206543   0.112104  1.07627
## age                -0.0007166   0.003791 -0.18903
## gendermale         0.4731314   0.121327  3.89963
##
## Intercepts:
##      Value   Std. Error t value
## SV|A    -3.5717    0.3522  -10.1421
## A|SP     -1.0291    0.3221   -3.1949
## SP|KRF   -0.7787    0.3216   -2.4214
## KRF|V    -0.6325    0.3214   -1.9681
## V|H      -0.3570    0.3211   -1.1117
## H|FRP     1.4273    0.3256    4.3833
##
## Residual Deviance: 2934.301
## AIC: 2956.301
## (103 observations deleted due to missingness)
```

Stargazertabell

```
library(stargazer)
```

```
##
```

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>

```
stargazer(party_reg, font.size = "footnotesize",
  star.cutoffs = c(.05, .01, .001),
  column.sep.width = ".01cm",
  no.space = FALSE,
  covariate.labels = c("Politikertillit (0-10)"))
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu

% Date and time: lø., sep. 16, 2017 - kl. 16.27 +0200

Table 2:

	<i>Dependent variable:</i>								
	FRP	H	KRF	KYST	MDG	RØDT	SP	SV	V
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Politikertillit (0-10)	-0.348*** (0.055)	-0.060 (0.044)	0.024 (0.085)	0.178 (0.432)	-0.115 (0.109)	-0.038 (0.171)	-0.082 (0.078)	0.089 (0.090)	0.053 (0.078)
Constant	0.787** (0.279)	0.291 (0.247)	-2.008*** (0.496)	-6.154* (2.684)	-1.922** (0.595)	-3.221*** (0.970)	-1.309** (0.434)	-2.455*** (0.540)	-1.949*** (0.461)
Akaike Inf. Crit.	3,736.993	3,736.993	3,736.993	3,736.993	3,736.993	3,736.993	3,736.993	3,736.993	3,736.993

Note:

*p<0.05; **p<0.01; ***p<0.001

```
stargazer(party_reg2, font.size = "footnotesize",
  star.cutoffs = c(.05, .01, .001),
  column.sep.width = ".01cm",
  no.space = FALSE,
  covariate.labels = c("Politikertillit (0-10)", "Inntekt (desil)",
    "Øk. tilfredshet", "Alder", "Kjønn (mann)"))
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu

% Date and time: lø., sep. 16, 2017 - kl. 16.27 +0200

```
stargazer(rang_party, font.size = "footnotesize",
  star.cutoffs = c(.05, .01, .001),
  column.sep.width = ".01cm",
  no.space = FALSE,
  covariate.labels = c("Politikertillit (0-10)", "Inntekt (desil)",
    "Øk. tilfredshet", "Alder", "Kjønn (mann)"))
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu

% Date and time: lø., sep. 16, 2017 - kl. 16.27 +0200

Table 3:

	<i>Dependent variable:</i>					
	FRP	H	KRF	SP	SV	V
	(1)	(2)	(3)	(4)	(5)	(6)
Politikertillit (0-10)	−0.326*** (0.061)	−0.086 (0.047)	0.019 (0.108)	−0.076 (0.083)	0.052 (0.099)	0.046 (0.084)
Inntekt (desil)	−0.102* (0.047)	0.097** (0.033)	−0.047 (0.078)	0.038 (0.059)	0.084 (0.065)	0.038 (0.056)
Øk. tilfredshet	0.320 (0.188)	−0.104 (0.160)	0.418 (0.316)	0.157 (0.271)	−0.051 (0.309)	−0.106 (0.268)
Alder	0.003 (0.007)	−0.013* (0.005)	0.019 (0.012)	0.003 (0.009)	−0.025* (0.011)	−0.020* (0.009)
Kjønn (mann)	0.886*** (0.238)	0.197 (0.166)	−0.572 (0.393)	0.419 (0.302)	−0.678* (0.342)	−0.267 (0.284)
Constant	0.178 (0.557)	−0.093 (0.441)	−2.539* (0.986)	−1.974* (0.781)	−2.423** (0.899)	−1.825* (0.763)
Akaike Inf. Crit.	2,909.938	2,909.938	2,909.938	2,909.938	2,909.938	2,909.938
<i>Note:</i>				*p<0.05; **p<0.01; ***p<0.001		

Table 4:

	<i>Dependent variable:</i>	
	<i>party_vote_short</i>	
Politikertillit (0-10)	−0.176*** (0.034)	
Inntekt (desil)	0.001 (0.024)	
Øk. tilfredshet	0.121 (0.112)	
Alder	−0.001 (0.004)	
Kjønn (mann)	0.473*** (0.121)	
Observations	925	
<i>Note:</i>		
*p<0.05; **p<0.01; ***p<0.001		