# Om oppgavene 4C1 og 4C3 fra Wooldridge

```
suppressPackageStartupMessages(library(wooldridge))
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(car))
suppressPackageStartupMessages(library(multcomp))
suppressPackageStartupMessages(library(olsrr))
suppressPackageStartupMessages(library(mctest))
suppressPackageStartupMessages(library(lmtest))
```

## Aller først; standard lineær regresjon i R

R bruker en såkalt `formula` for å spesifisere modellen. Dette innebærer at operasjoner som "+", "-", ":", "*", "^" og tom. "%in" får helt ny betyding (se ?formula for detaljer). Ønsker vi at de skal ha sin vanlige betydning må vi sette dem inn i en `I()` funksjon (fra help: *Change the class of an object to indicate that it should be treated 'as is'.)*. Funksjoner kan vi fritt bruke inne i en formula så vi kan gjerne ha `log(price)` som en variabel. Det er altså ingen grunn til først å lage en ny variabel `lprice = log(price)`.

For å vise hvordan en enkel multippel regresjon kan formuleres og tolkes i R kan vi ta utgangspunkt i eksempel 3.5 Wooldridge. Pakken `wooldridge` er lastet så vi trenger bare

```
data(crime1)
```

for å få tilgang til datasettet `crime1`. Sjekker vi klassen til crime1 vha. '`class(crime1)`` får vi til svar: data.frame. Dataene er altså klar til bruk.

```
# spesifiserer modellen, _mr for multippel regresjon
mod_mr = "narr86 ~ pcnv + ptime86 + qemp86"
lm_mr <- lm(mod_mr, data=crime1)
```

For å se rapporten fra regresjonen kan vi skrive:

```
summary(lm_mr)
```

```
##
## Call:
## lm(formula = mod_mr, data = crime1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7118 -0.4031 -0.2953  0.3452 11.4358
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.711772   0.033007  21.565  < 2e-16 ***
```

```
## pcnv        -0.149927   0.040865  -3.669 0.000248 ***
## ptime86     -0.034420   0.008591  -4.007 6.33e-05 ***
## qemp86      -0.104113   0.010388 -10.023  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8416 on 2721 degrees of freedom
## Multiple R-squared:  0.04132,    Adjusted R-squared:  0.04027
## F-statistic:  39.1 on 3 and 2721 DF,  p-value: < 2.2e-16
```

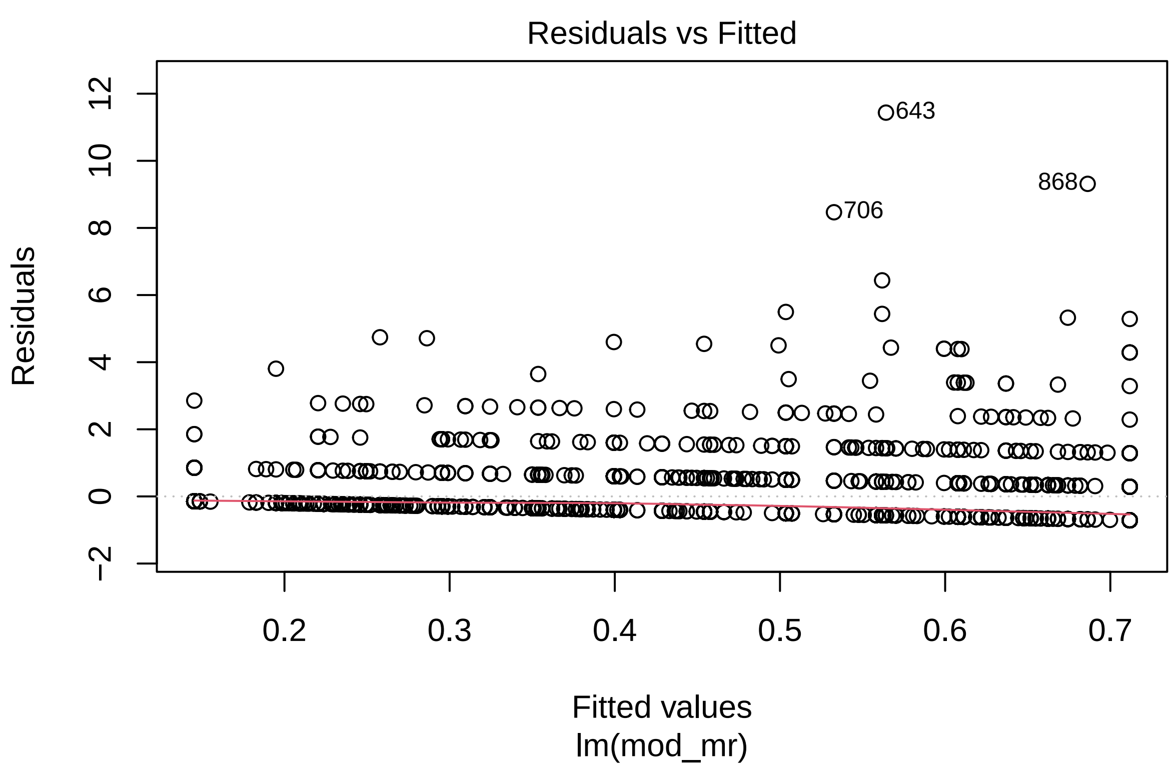
Her er da en standard rapport fra en regresjon, med estimerte koeffisienter, standard-feil og t-verdier for å teste om koeffisientene er forskjellig fra 0.

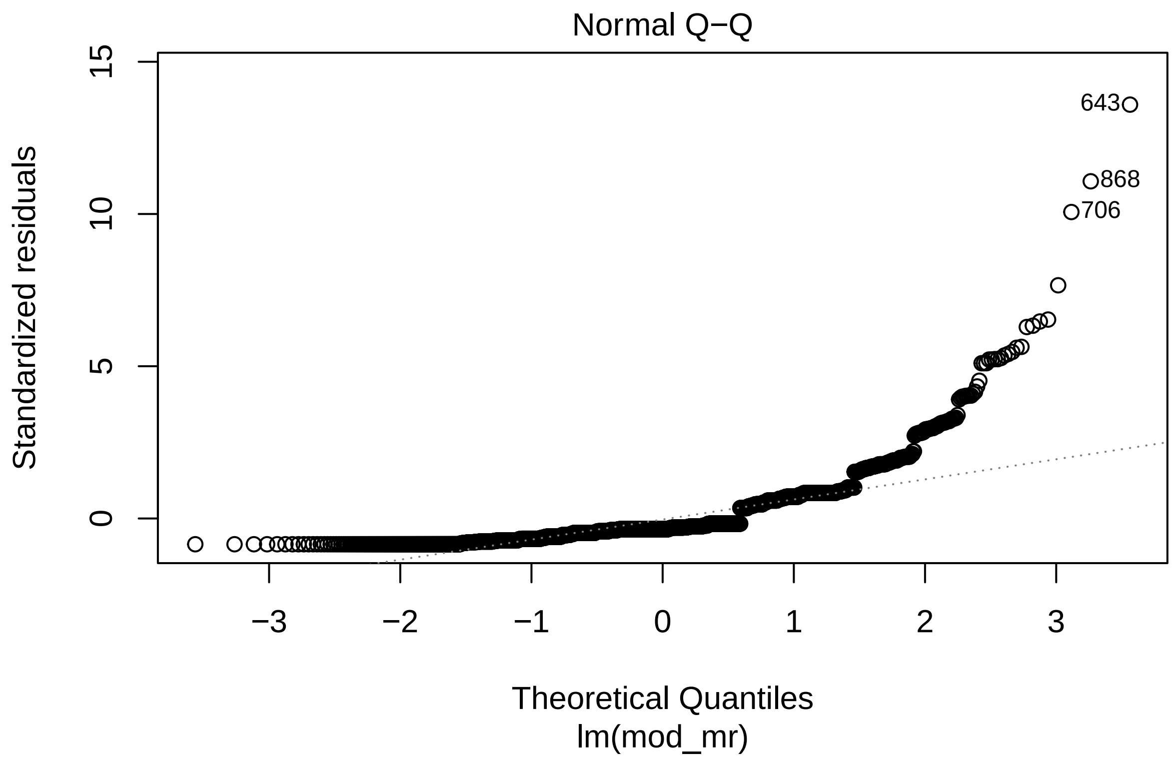
Vi ser at alle koeffisientene er klart signifikant forskjellige fra null. Fortegnene er også som forventet

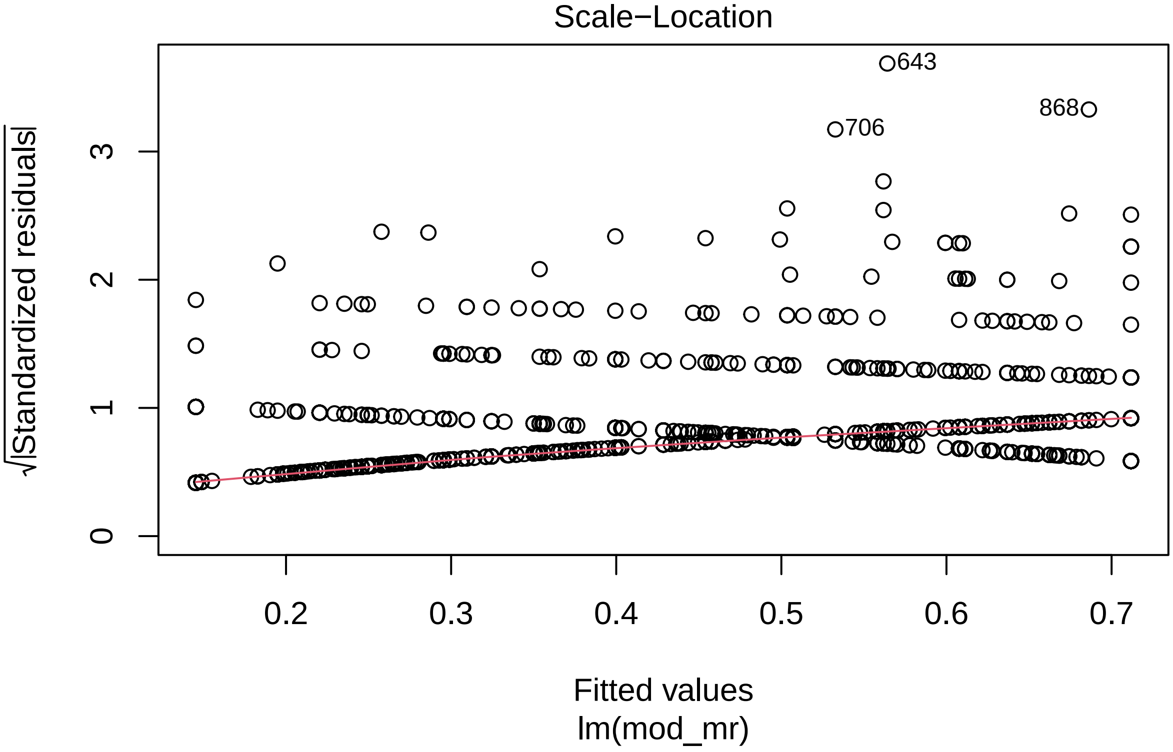
- Datasettet inneholder informasjon om menn fra California født 1960 eller 1961 som har vært arrestert *minst* en gang i årene før 1986.
- narr86: antall ganger en person ble arrestert i 1986
- pcnv: andel av arresterte før 1986 som ble dømt. Koeffisienten ha verdien -0,1499, dvs alt annet like vil høyere andel dømt føre til færre arrestasjoner 1986.
- avgsen: gjennomsnittlig straff sonet for tidligere forhold
- ptime86: antall måneder sonet i fengsel i 1986. Koeffisienten har verdien -0,0344, dvs. dess mer tid i fengsel 1986 dess færre arrestasjoner (vanskelig å bli arrestert hvis du alt soner)
- qemp86: muligheter for å få jobb 1986. Koeffisienten er -0,1041, dvs. dess bedre jobbmarkedet er dess færre arrestasjoner.

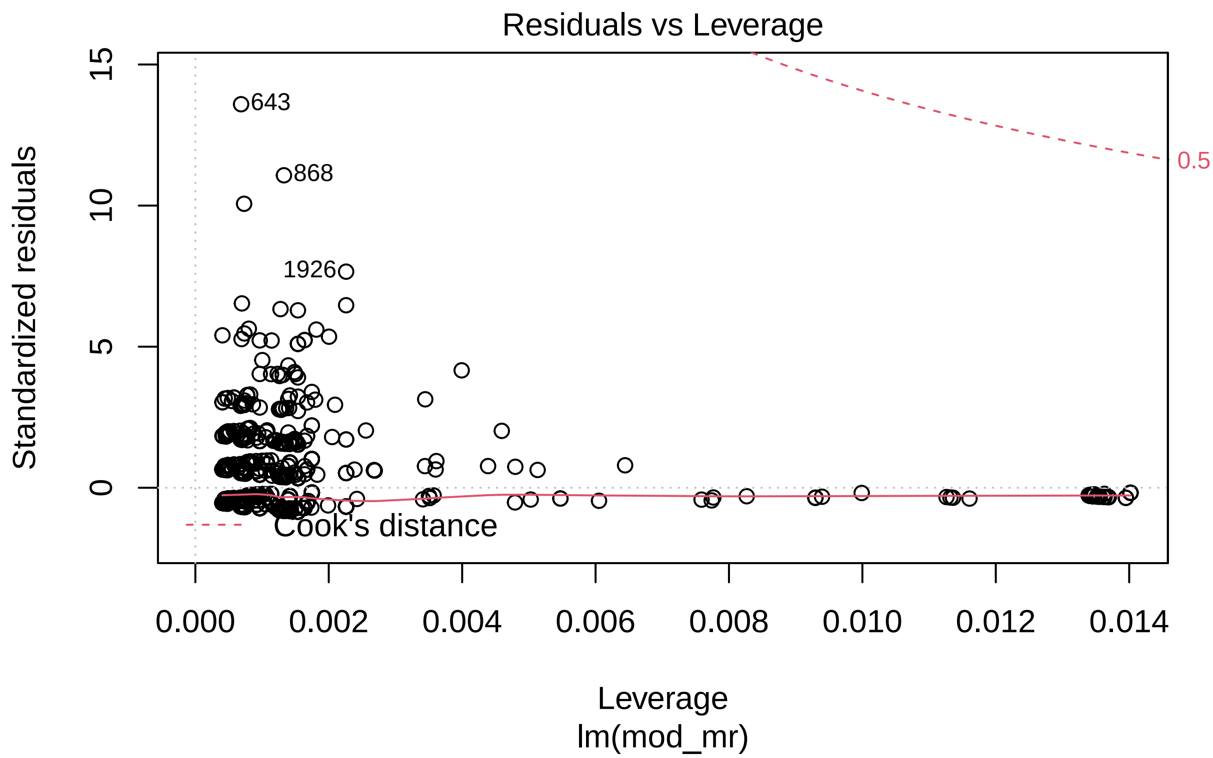
Ønsker vi mer diagnostikk for modellen kan vi

```
plot(lm_mr)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(mod_mr)
643
868
706


Scale−Location
√|Standardized residuals|
Fitted values
lm(mod_mr)
643
868
706




Så utvider vi modellen med å ta med variabelen **avgsen**. Dette gir oss

```
mod_mr_2 = "narr86 ~ pcnv + avgsen + ptime86 + qemp86"
lm_mr_2 <- lm(mod_mr_2, data=crime1)
```

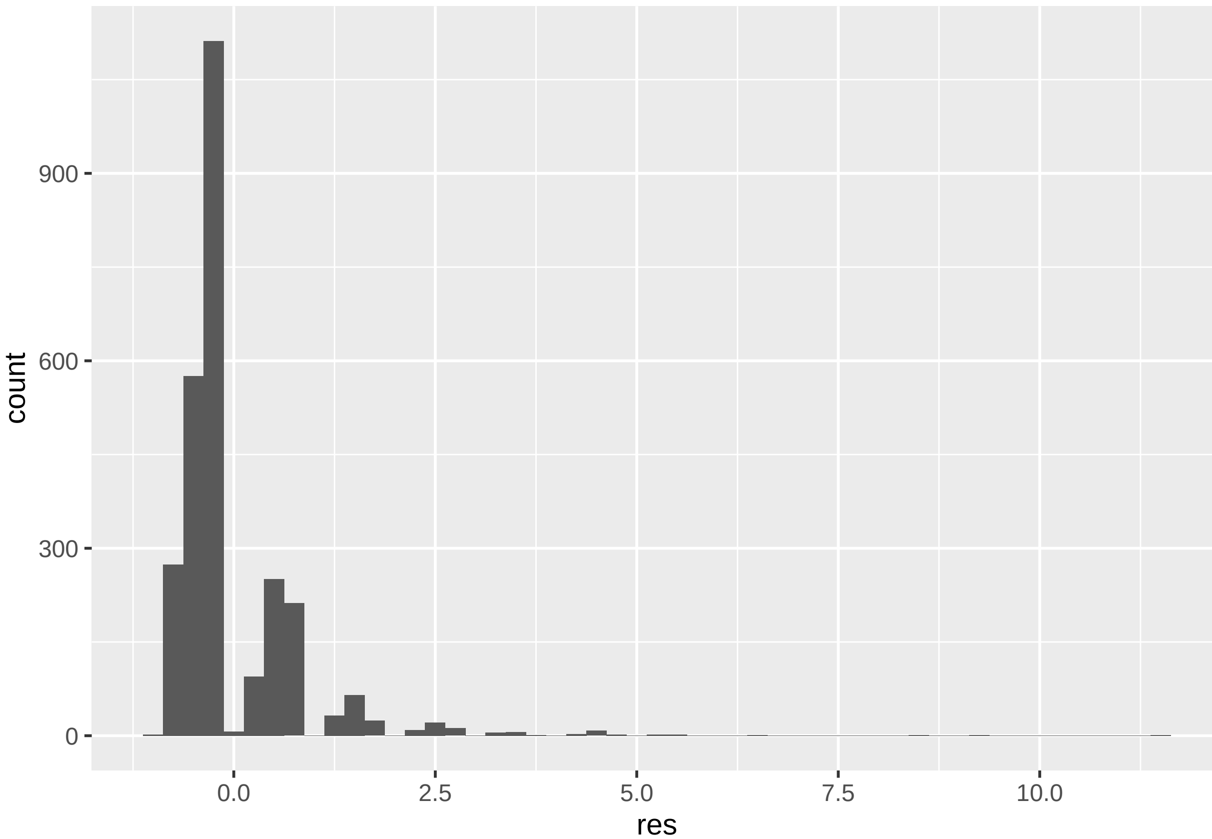
```
summary(lm_mr_2)
```

```
##
## Call:
## lm(formula = mod_mr_2, data = crime1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9330 -0.4247 -0.2934  0.3506 11.4403
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.706756   0.033151  21.319  < 2e-16 ***
## pcnv        -0.150832   0.040858  -3.692 0.000227 ***
## avgsen       0.007443   0.004734   1.572 0.115993
## ptime86     -0.037391   0.008794  -4.252 2.19e-05 ***
## qemp86      -0.103341   0.010396  -9.940  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8414 on 2720 degrees of freedom
## Multiple R-squared:  0.04219,    Adjusted R-squared:  0.04079
## F-statistic: 29.96 on 4 and 2720 DF,  p-value: < 2.2e-16
```

Ser at variabelen **avgsen** bare gir en liten økning i R-squared, koeffisienten er ikke signifikant og fortegnet er motsatt av forventet. Positivt fortegn, dvs at lengre straffer skulle medføre flere arrestasjoner. Taler for at `avgsen` kanskje ikke er en god forklaringsvariabel.

Ønsker vi å sjekke om residualene er normalfordelt kan vi grafisk utforske dette vha.

```
data.frame(res=residuals(lm_mr_2)) %>%
  ggplot(mapping=aes(x=res)) +
  geom_histogram(binwidth = .25)
```




Ser ikke så bra ut. Kanskje vi skal prøve med å log-transformere avhengig variabel. Sjekker først variabelen vha. summary

```
summary(crime1$narr86)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.4044  1.0000 12.0000
```

Ser at vi har mange verdier lik 0, går ikke bra med ln. Et «triks» som ofte blir brukt er å legge til en liten positiv verdi, f.eks 0.01. Vi må huske at bruker vi «+» så må denne beskyttes av I() i en formula.

```
mod_mr_2_log = "log(I(narr86 + 0.01)) ~ pcnv + avgsen + ptime86 + qemp86"
lm_mr_2_log <- lm(mod_mr_2_log, data=crime1)
```

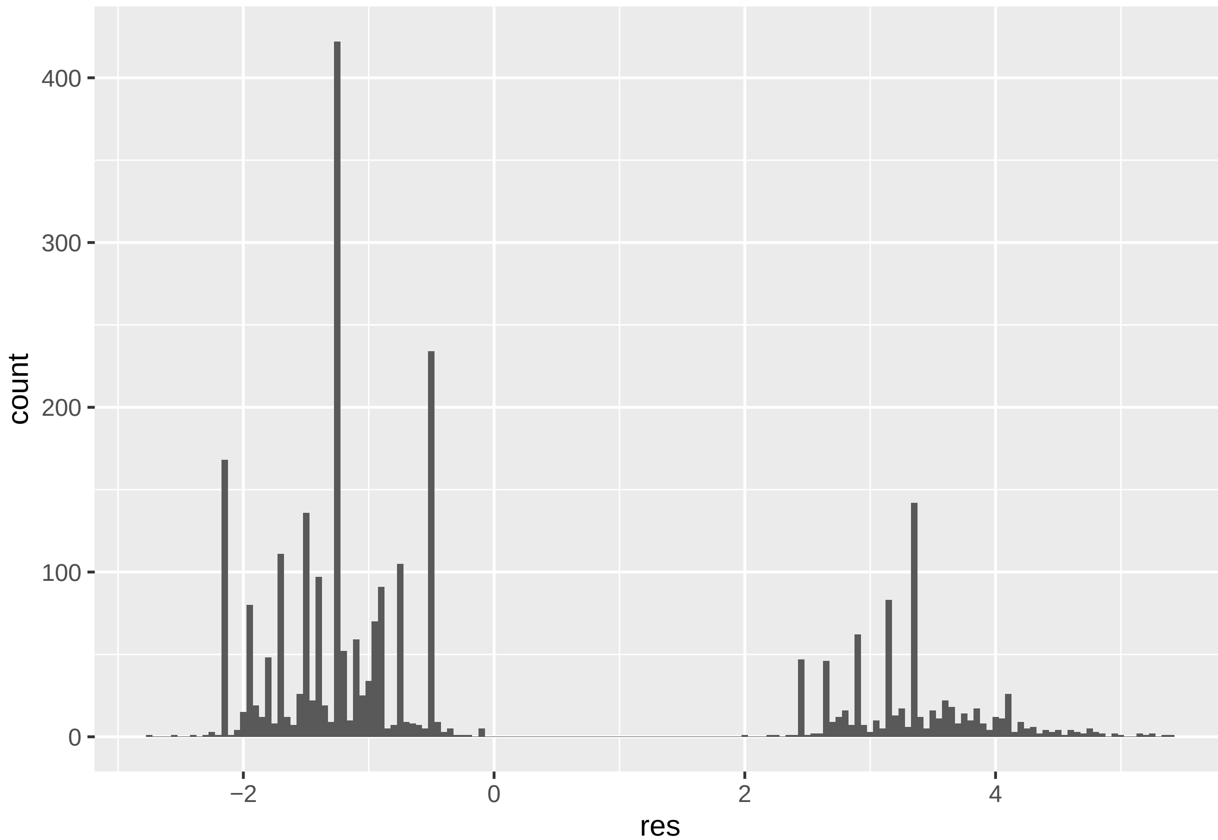
```
summary(lm_mr_2_log)
```

```
##
## Call:
## lm(formula = mod_mr_2_log, data = crime1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.730 -1.421 -1.046  2.674  5.400
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.43764    0.08411 -28.983  < 2e-16 ***
## pcnv        -0.74599    0.10366  -7.197 7.94e-13 ***
## avgsen       0.01850    0.01201   1.540    0.124
## ptime86     -0.11073    0.02231  -4.963 7.36e-07 ***
## qemp86      -0.22684    0.02638  -8.600  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.135 on 2720 degrees of freedom
## Multiple R-squared:  0.04868,    Adjusted R-squared:  0.04728
## F-statistic: 34.79 on 4 and 2720 DF,  p-value: < 2.2e-16
```

Vi ser at log-transformasjonen av den avhengige variabelen øker $R^2$ . Variabelen `avgsen` er fremdeles ikke signifikant og har motsatt fortegn fra forventet.

Sjekker residualene på ny

```
data.frame(res = as.numeric(residuals(lm_mr_2_log))) %>%
  ggplot(mapping=aes(x = res)) +
  geom_histogram(binwidth = .05)
```

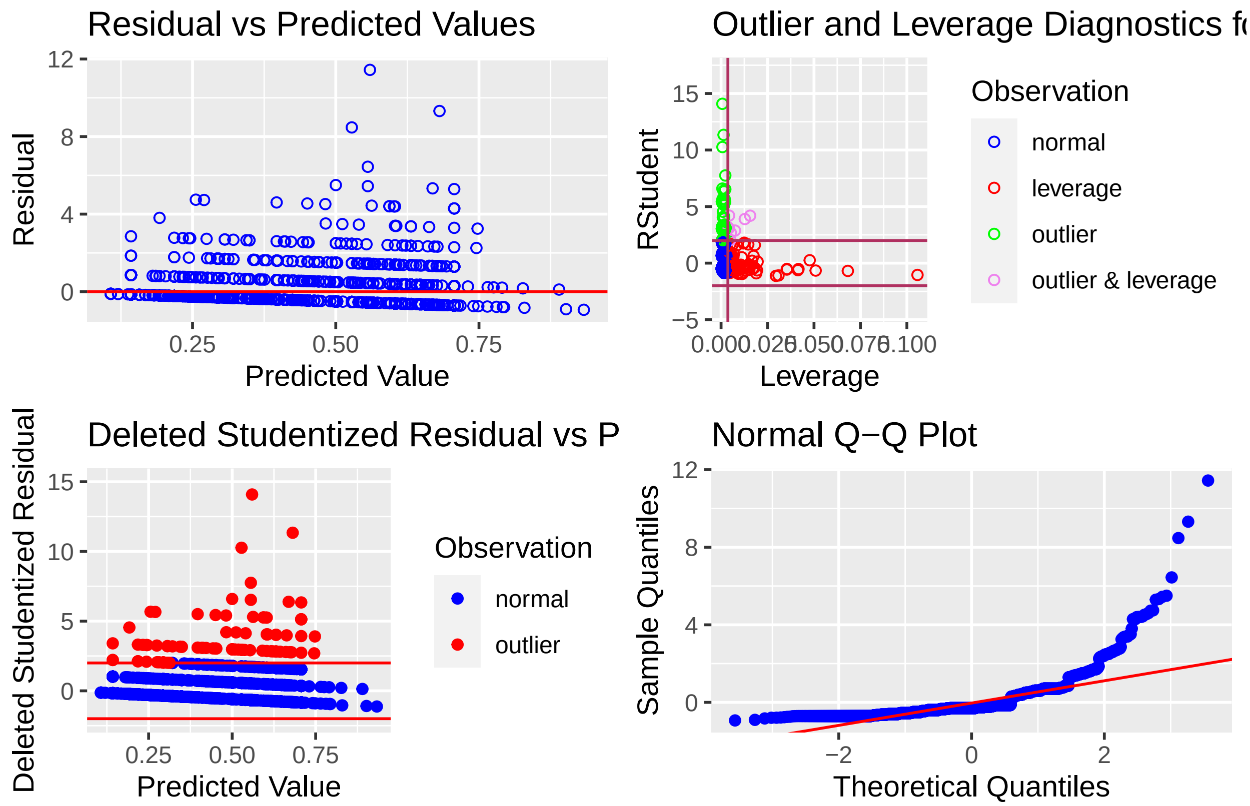



Kanskje noe bedre, men residualene ser ut til å være langt fra normalfordelt.

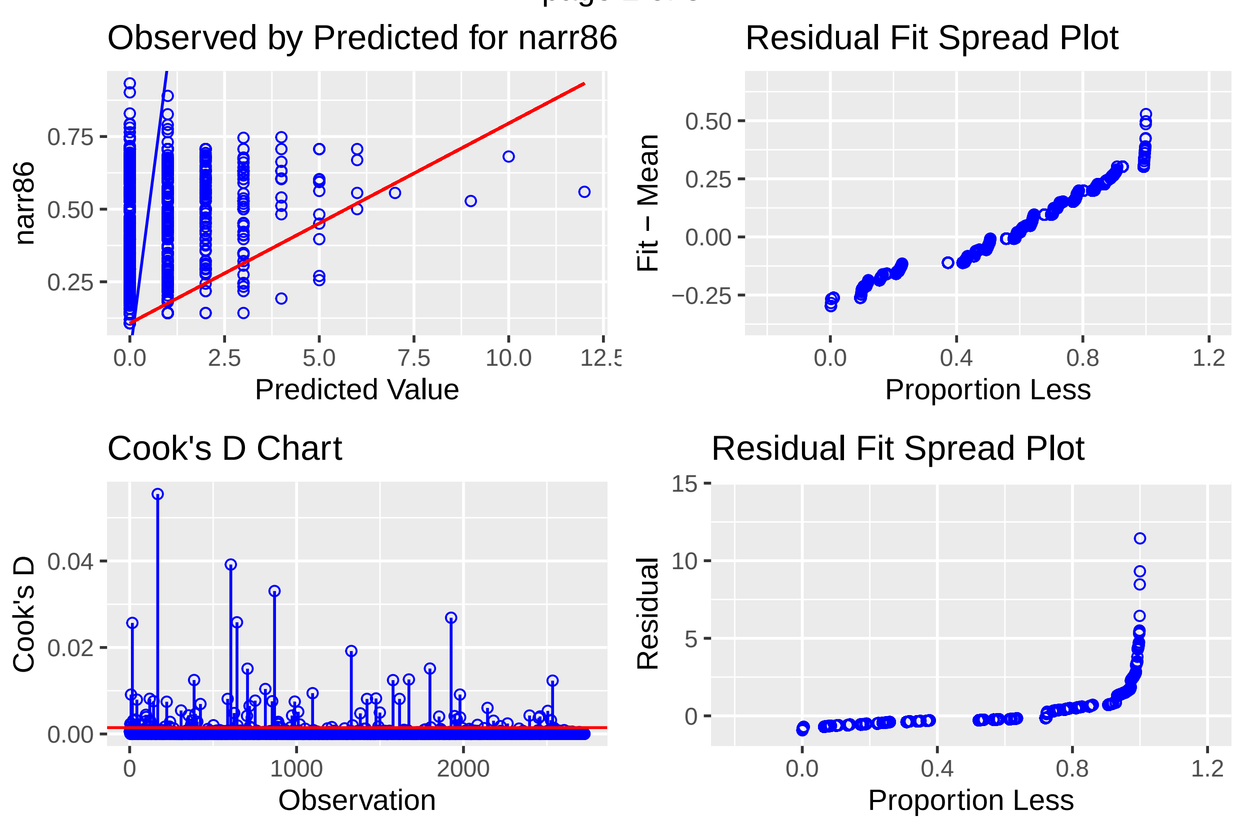
Ønsker en å teste nærmere en regresjonsmodell er pakken `olsrr` et flott verktøy. Pakken har utmerket dokumentasjon (se intro olsrr ) og er laget spesielt mht. å teste om forutsetningene for en lineær regresjon er brutt. Vi bruker modellen `lm_mr_2` siden `olsrr` ikke ser ut til å like modeller som inneholder bruk av `I()` funksjonen, dvs. skulle vi brukt modellen `lm_mr_2_log` måtte vi først laget oss en ny variabel `lnarr86 = log(narr86 + 0.01)`.

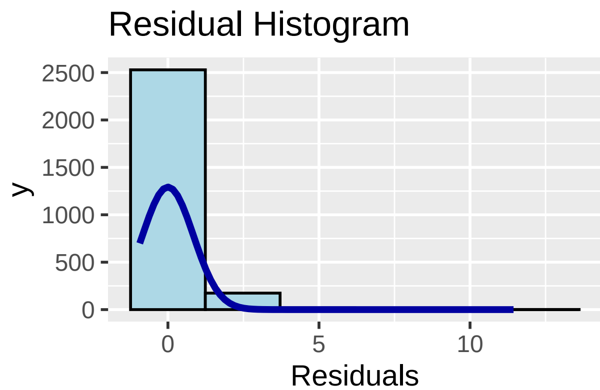
```
ols_plot_diagnostics(lm_mr_2)
```

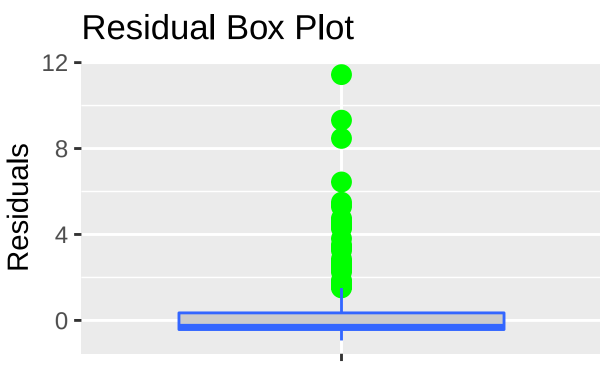
page 1 of 3




page 2 of 3






Vi ser klart at det er problemer forbundet med modellen vår.

```
ols_test_normality(lm_mr_2)
```

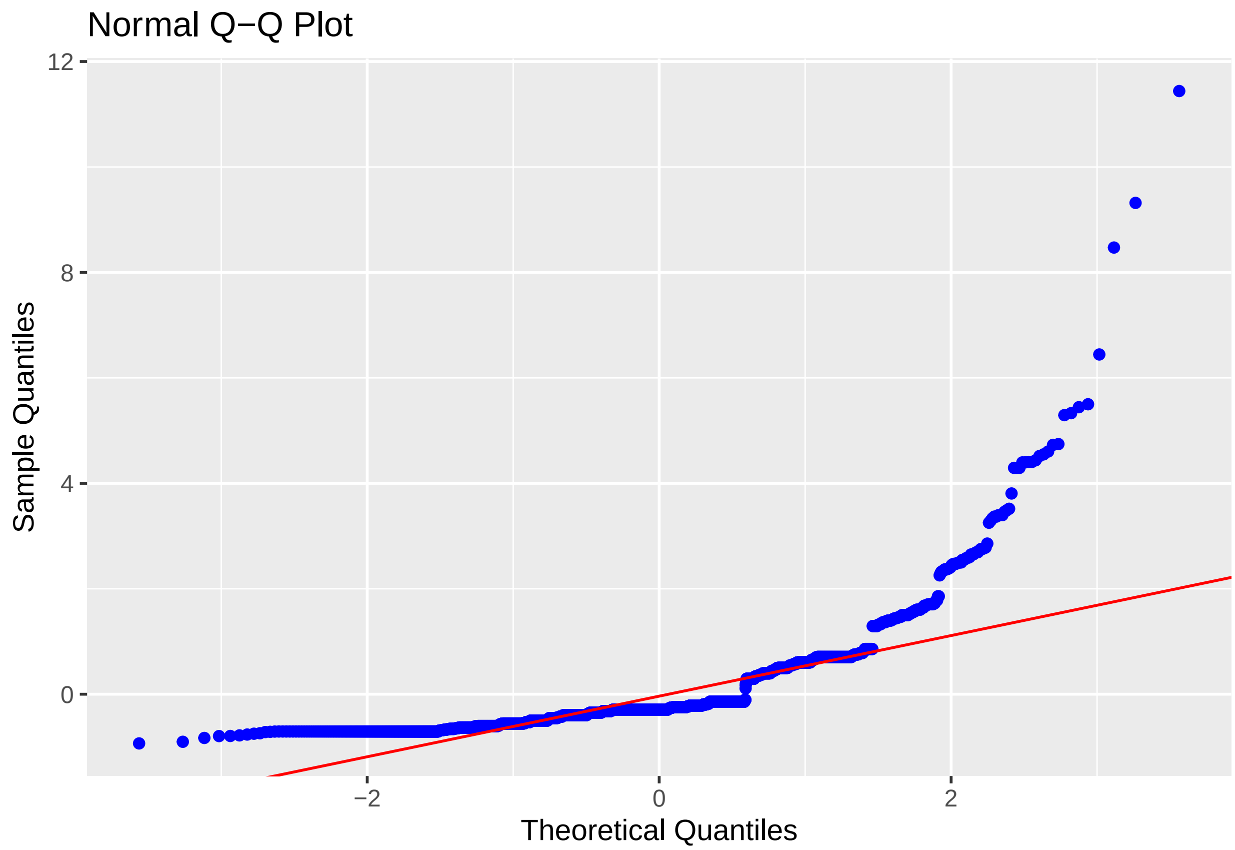
**Test for normalitet residualer**

```
## Warning in ks.test(y, "pnorm", mean(y), sd(y)): ties should not be present for
## the Kolmogorov-Smirnov test

## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.6601         0.0000
## Kolmogorov-Smirnov        0.2877         0.0000
## Cramer-von Mises        389.9559         0.0000
## Anderson-Darling        228.9276         0.0000
## -----------------------------------------------
```

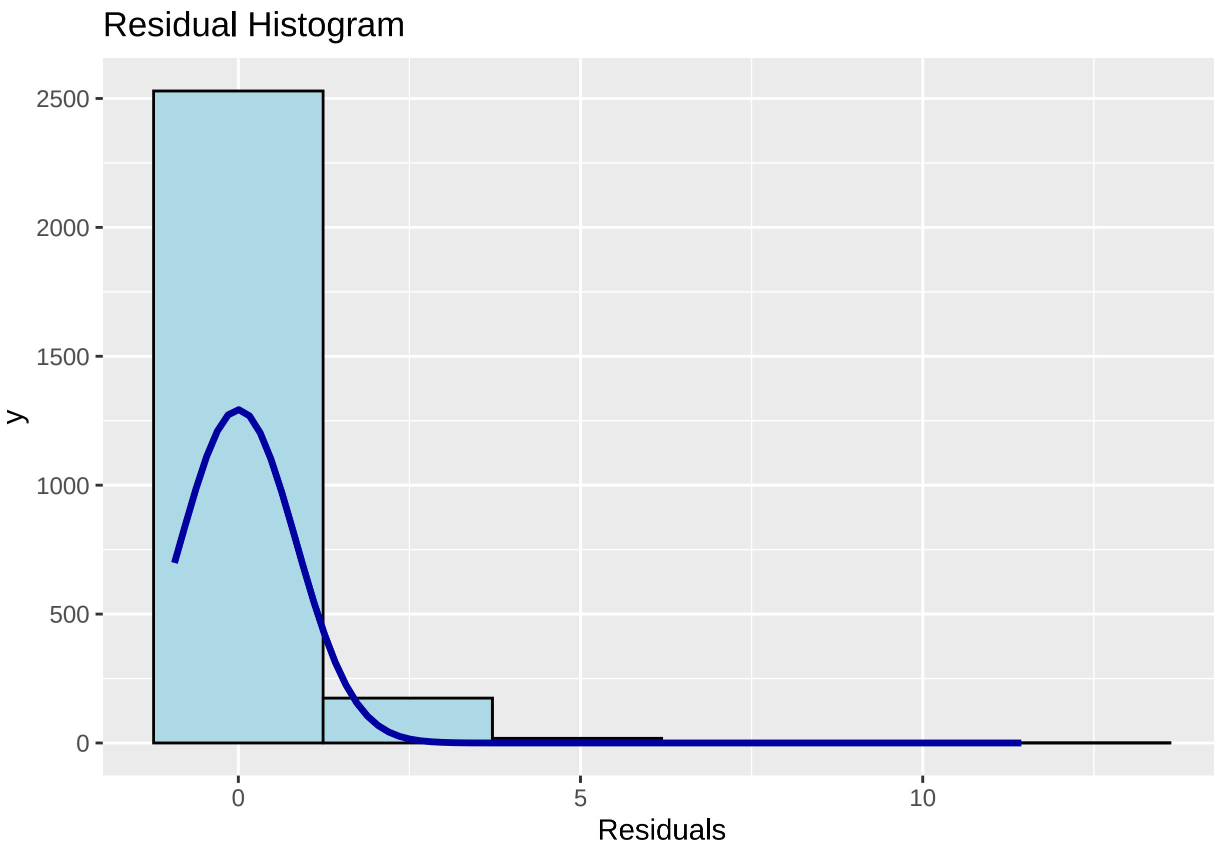
Samtlige tester gir at vi kan forkaste null-hypotesen om normalfordelte residualer.

```
ols_plot_resid_qq(lm_mr_2)
```




Hvis residualene skulle være normalfordelt skulle de blå prikkene ligge langs en tilnærmet rett linje. Vi ser igjen at residualene i regresjonen ikke er normalfordelte.

```
ols_plot_resid_hist(lm_mr_2)
```




Igjen liten støtte for antakelsen om normalfordelte residualer.

**Test for heteroskedastisitet** En poplær test for heteroskedastisitet er Breusch-Pagan.

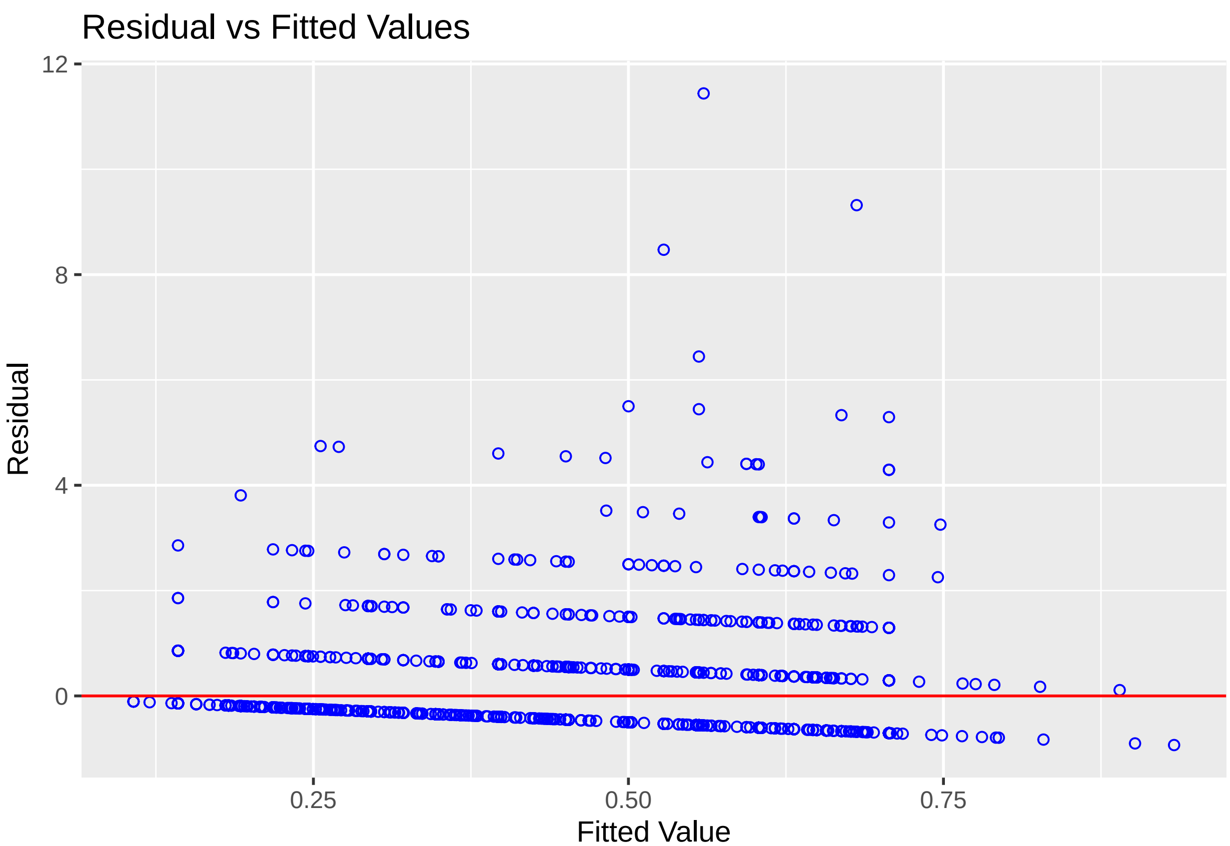
```
ols_test_breusch_pagan(lm_mr_2)
```

```
##
##  Breusch Pagan Test for Heteroskedasticity
##  -----------------------------------------
##  Ho: the variance is constant
##  Ha: the variance is not constant
##
##               Data
##  ----------------------------------
##  Response : narr86
##  Variables: fitted values of narr86
##
##          Test Summary
##  --------------------------------
##  DF            =    1
##  Chi2          =    512.6237
##  Prob > Chi2   =    1.703632e-113
```

Vi ser at vi kan forkaste null-hypotesen om konstant varians.

**Plot av residualer mot modellverdier (fitted values)** Et annet populært diagnose-plot er «Residuals vs Fitted Values».

```
ols_plot_resid_fit(lm_mr_2)
```




Ser ikke bra ut. Burde ligge som et tilfeldig jevnt bånd langs den horisontale aksen uten noe klarer mønster.

Til sist er det verdt å nevne at `olsrr` også har en informativ regresjon-rapport.

```
ols_regress(lm_mr_2)
```

```
##                            Model Summary
## ---------------------------------------------------------------
## R                        0.205       RMSE                0.841
## R-Squared                0.042       Coef. Var         208.053
## Adj. R-Squared           0.041       MSE                 0.708
## Pred R-Squared           0.039       MAE                 0.553
## ---------------------------------------------------------------
##  RMSE: Root Mean Square Error
##  MSE: Mean Square Error
##  MAE: Mean Absolute Error
##
##                                   ANOVA
## ----------------------------------------------------------------------
##                 Sum of
##                Squares          DF     Mean Square      F          Sig.
## ----------------------------------------------------------------------
```

```
## Regression      84.824           4         21.206    29.956    0.0000
## Residual      1925.523        2720          0.708
## Total         2010.347        2724
## ----------------------------------------------------------------------
##
##                                  Parameter Estimates
## ----------------------------------------------------------------------------------------
##       model      Beta    Std. Error    Std. Beta      t        Sig      lower     upper
## ----------------------------------------------------------------------------------------
## (Intercept)     0.707         0.033                 21.319    0.000     0.642     0.772
##        pcnv    -0.151         0.041       -0.069    -3.692    0.000    -0.231    -0.071
##      avgsen     0.007         0.005        0.030     1.572    0.116    -0.002     0.017
##     ptime86    -0.037         0.009       -0.085    -4.252    0.000    -0.055    -0.020
##      qemp86    -0.103         0.010       -0.194    -9.940    0.000    -0.124    -0.083
## ----------------------------------------------------------------------------------------
```

Vi ser at rapporten bl.a inneholder konfidensintervall for koeffisientene (lower upper).

## Restricted models

Dette er viktig stoff og er dekket i avsnittene 4-2c, 4-4 og 4-5. Viser først eksemplene fra disse avsnittene løst vha. R.

## Eksempler løst i R

### Fra avsnitt 4-2c

#### Ex. 4.4

Modellen

$$\mathsf{log}(\mathsf{crime}) = \beta_0 + \beta_1 \mathsf{log}(\mathsf{enroll}) + u$$

skriver vi i R som "log(crime) ~ log(enroll)" konstantleddet kommer automatisk. Skulle vi ønske *uten* konstantledd (generelt frarådet) skriver vi "log(crime) ~ log(enroll) - 1"

```r
data(campus)
# cc crime campus, finnes også variablene lcrime og lenroll i datasettet der
# ln av variablene alt er tatt. I R er det like lett å bruke funksjonen selv
mod_cc <- "log(crime) ~ log(enroll)"
lm_cc <- lm(mod_cc, data=campus)
# uten konstantledd
# mod_cc <- "log(crime) ~ log(enroll) - 1"
```

```r
summary(lm_cc)
```

```
##
## Call:
## lm(formula = mod_cc, data = campus)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5136 -0.3858  0.1174  0.4363  2.5782
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -6.6314     1.0335  -6.416 5.44e-09 ***
## log(enroll)   1.2698     0.1098  11.567  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8946 on 95 degrees of freedom
## Multiple R-squared:  0.5848, Adjusted R-squared:  0.5804
## F-statistic: 133.8 on 1 and 95 DF,  p-value: < 2.2e-16
```

Elastisiteten for `campus crime` mht. `enroll` (universitetsstørrelse) er altså estimert til $1,2698 \approx 1,27$, dvs. 1% økning i antall studenter gir 1,27% økning i kriminalitet. Et sentralt punkt her er om denne elastisiteten er signifikant større enn 1. Er den det vil vi ha en *relativ* økning i kriminalitet når størrelsen øker. Altså at et dobbelt så stort universitet vil ha *mer enn* dobbelt så høy kriminalitet.

Ønsker å test H0: $\beta_1 = 1$ mot H1: $\beta_1 > 1$. I summary ovenfor er det hypotesene H0: $\beta_1 = 0$ mot H1: $\beta_1 \neq 1$ som blir testet. Hva gjør vi? Vi regner ut ny t-verdi vha. $\frac{\text{estimat - verdi i hypotese}}{\text{standard error}}$, dvs $\frac{1,2698-1}{0,1098} = 2,457$.

Vi må så finne kritisk verdi eller p-verdi (husk ensidig H1 her)

```r
# obs. in campus
dim(campus)
```

```
## [1] 97  7
```

```r
num_obs <- dim(campus)[1]
num_obs
```

```
## [1] 97
```

Antall frihetsgrader (95 DF) kan vi også lese direkte ut fra siste linje i summary ovenfor.

p-verdi (For å se fordelinger kjent av base R kjør `?distributions` i Console)

```r
# p-verdi ensidig H1
pt(2.457, df=num_obs-2, lower.tail = FALSE)
```

```
## [1] 0.007912419
```

kritisk verdi, ensidig H1 ulike $\alpha$

```r
# alpha lik 0,05 ensidig
qt(0.05, df = num_obs-2, lower.tail = FALSE)
```

```
## [1] 1.661052
```

```r
# alpha lik 0,05 ensidig
qt(0.01, df = num_obs-2, lower.tail = FALSE)
```

```
## [1] 2.366243
```

Vi ser at $\beta_1$ er signifikant forskjellig fra 1 på nivå $\alpha = 1\%$, altså har vi en overproporsjonal økning i kriminalitet når universitetsstørrelsen øker.

For ordens skyld: Wooldrige skriver en del om å lage konfidensintervall. For å finne konfidensintervall for modellen ovenfor gjør vi følgende

```r
# default 5%
confint(lm_cc)
```

```
##                 2.5 %    97.5 %
## (Intercept) -8.683207 -4.579534
## log(enroll)  1.051827  1.487693
```

```r
# 1%
confint(lm_cc, level=0.99)
```

```
##                  0.5 %    99.5 %
## (Intercept) -9.3481083 -3.914632
## log(enroll)  0.9812058  1.558315
```

**Ex. 4.5**

```r
# Boston housing data; hprice2. See Wooldridge or give command ?hprice2 in console
# for description of the variables
data(hprice2)
mod_hp <- "log(price) ~ log(nox) + log(dist) + rooms + stratio"
lm_hp <- lm(mod_hp, data=hprice2)
```

```r
summary(lm_hp)
```

```
##
## Call:
## lm(formula = mod_hp, data = hprice2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.05890 -0.12427  0.02128  0.12882  1.32531
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.083861   0.318111  34.843  < 2e-16 ***
## log(nox)    -0.953539   0.116742  -8.168 2.57e-15 ***
## log(dist)   -0.134339   0.043103  -3.117  0.00193 **
## rooms        0.254527   0.018530  13.736  < 2e-16 ***
```

```
## stratio     -0.052451   0.005897  -8.894  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.265 on 501 degrees of freedom
## Multiple R-squared:  0.584,  Adjusted R-squared:  0.5807
## F-statistic: 175.9 on 4 and 501 DF,  p-value: < 2.2e-16
```

Now $\beta_1$ (coefficient estimate for log(nox) lik -0,9535) er priselastisiteten for boliger mht. nox utslipp. Vi ønsker å teste H0: $\beta_1 = -1$ mot H1: $\beta_1 \neq -1$. Vi benytter samme teknikk som ovenfor, men husker at nå er alternativ hypotese tosidig. Vi regner altså ut ny t-verdi vha. $\frac{\text{estimat - verdi i hypotese}}{\text{standard error}}$, dvs $\frac{-0{,}9535-(-1)}{0{,}1167} = \frac{0{,}0465}{0{,}1167} = 0,3985$. Finner p-verdi og kritiske t-verdier. Antall frihetsgrader er 501.

```r
# p-verdi tosidig H1
2*pt(0.3985, df=501, lower.tail=FALSE)
```

```
## [1] 0.6904314
```

Kritisk t-verdi 5% nivå

```r
# alpha lik 0,05 tosidig
qt(0.05/2, df = 501, lower.tail = FALSE)
```

```
## [1] 1.96471
```

For ordens skyld også

```r
# alpha lik 0,05 tosidig
qt(0.05/2, df = 501, lower.tail = TRUE)
```

```
## [1] -1.96471
```

Vi kan altså ikke forkaste H0, dvs. det er lite bevis for at $\beta_1$ er forskjellig fra -1.

## Fra avsnitt 4-4

Testing av hypoteser med én lineær kombinasjon av parametre.

Modellen

$$\mathsf{log(wage)} = \beta_0 + \beta_1 \mathsf{jc} + \beta_2 \mathsf{univ} + \beta_3 \mathsf{exper} + u$$

```r
data(twoyear)
#variable lwage in dataset is log(wage)
mod_2y <- "lwage ~ jc + univ +exper"
lm_2y <- lm(mod_2y, data = twoyear)
```

```r
summary(lm_2y)
```

```
##
## Call:
## lm(formula = mod_2y, data = twoyear)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.10362 -0.28132  0.00551  0.28518  1.78167
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.4723256  0.0210602  69.910   <2e-16 ***
## jc          0.0666967  0.0068288   9.767   <2e-16 ***
## univ        0.0768762  0.0023087  33.298   <2e-16 ***
## exper       0.0049442  0.0001575  31.397   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4301 on 6759 degrees of freedom
## Multiple R-squared:  0.2224, Adjusted R-squared:  0.2221
## F-statistic: 644.5 on 3 and 6759 DF,  p-value: < 2.2e-16
```

Vi ser at det lønner seg både med junior college (jc) og college (univ), begge koeffisientene er signifikant forskjellig fra 0. Det vi er mest interessert i er om $\beta_{\mathsf{univ}}$ er *signifikant* større enn $\beta_{\mathsf{jc}}$. Altså om det *lønner seg* å velge universitet fremfor junior college.

For å test trenger vi å beregne t-verdien $t = \frac{\hat{\beta}_1 - \hat{\beta}_2}{\mathsf{se}(\hat{\beta}_1 - \hat{\beta}_2)}$. Problemet er at vi *ikke* finner $\mathsf{se}(\hat{\beta}_1 - \hat{\beta}_2)$ i standard rapporten for regresjon.

**NB!** $\mathsf{se}(\hat{\beta}_1 - \hat{\beta}_2) \neq \mathsf{se}(\hat{\beta}_1) - \mathsf{se}(\hat{\beta}_2)$

Vi bruker derfor et «triks» der vi skriver om modellen slik at $\mathsf{se}(\hat{\beta}_1 - \hat{\beta}_2)$ vil bli rapportert i standard `summary` fra modellen.

Definerer en ny parameter $\theta_1 = \beta_1 - \beta_2$ som gir at $\beta_1 = \theta_1 + \beta_2$. Det vi ønsker å teste er H0: $\theta = 0$ mot H1: $\theta < 0$. Vi er nå på jakt etter $\mathsf{se}(\theta_1)$ som vil være lik $\mathsf{se}(\hat{\beta}_1 - \hat{\beta}_2)$. Vi kan da skrive om modellen som

$$\mathsf{log}(\mathsf{wage}) = \beta_0 + \beta_1\mathsf{jc} + \beta_2\mathsf{univ} + \beta_3\mathsf{exper} + u = \beta_0 + (\theta_1 + \beta_2)\mathsf{jc} + \beta_2\mathsf{univ} + \beta_3\mathsf{exper} + u$$

som gir

$$\mathsf{log}(\mathsf{wage}) = \beta_0 + \theta_1\mathsf{jc} + \beta_2(\mathsf{univ} + \mathsf{jc}) + \beta_3\mathsf{exper} + u$$

Vi kan altså få tak i $\mathsf{se}(\hat{\beta}_1 - \hat{\beta}_2)$ ved å kjøre modellen

```r
# Legg merke til bruk av I() funksjonen. Denne trengs siden + har  en spesiell
# betydning i R sitt formula «språk». Inne i I() blir det summen av univ og jc for
# hver student
mod_2y_b <- "lwage ~ jc + I(univ + jc) + exper"
lm_2y_b <- lm(mod_2y_b, data = twoyear)
```

```r
summary(lm_2y_b)
```

```
##
## Call:
## lm(formula = mod_2y_b, data = twoyear)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.10362 -0.28132  0.00551  0.28518  1.78167
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.4723256  0.0210602  69.910   <2e-16 ***
## jc          -0.0101795  0.0069359  -1.468    0.142
## I(univ + jc)  0.0768762  0.0023087  33.298   <2e-16 ***
## exper        0.0049442  0.0001575  31.397   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4301 on 6759 degrees of freedom
## Multiple R-squared:  0.2224, Adjusted R-squared:  0.2221
## F-statistic: 644.5 on 3 and 6759 DF,  p-value: < 2.2e-16
```

Da kan vi lese ut standard error for $\theta_1$, som jo også er standard error for $(\hat{\beta}_1 - \hat{\beta}_2)$ som var det vi var på jakt etter. Da kan vi enkelt regne ut t-verdien

$$t = \frac{\hat{\beta}_1 - \hat{\beta}_2}{\mathsf{se}(\hat{\beta}_1 - \hat{\beta}_2)} = \frac{-0,01018}{0,00694} \approx -1,467$$

Finner p-verdi (ensidig)

```
# p-verdi ensidig H1
pt(-1.467, df=6759)
```

```
## [1] 0.07121129
```

Vi kan altså *ikke* på 5% nivå konkludere med at et år utdanning på college gir signifikant høyere lønn enn et år på junior college. På 10% nivå derimot kan vi konkludere med at ett år på college gir signifikant mer uttelling i lønn enn ett år på junior college.

Teknikken ovenfor kan vi *alltid* få til i et statistikkprogram. Finnes imidlertid pakker/rutiner som forsøker å forenkle dette. To slike, *car* og *multcomp* er vist nedenfor. Begge bruker F-test (istdenfor t-test) som samsvarer mer med avsnitt 4-5, men konklusjonene blir de samme.

**Med pakken `car`** Enklere måte (bruker F-test jmf. avsnitt 4-5)

```
# vi har lastet car så linearHypothesis er tilgjengelig
linearHypothesis(lm_2y, "jc - univ = 0")
```

```
## Linear hypothesis test
##
## Hypothesis:
## jc - univ = 0
##
## Model 1: restricted model
## Model 2: lwage ~ jc + univ + exper
##
```

```
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1   6760 1250.9
## 2   6759 1250.5  1   0.39853 2.154 0.1422
```

```r
# pf: p value F distribution
pf(2.154, 1, Inf, lower.tail = FALSE)
```

```
## [1] 0.142199
```

Som en ser er resultatet ovenfor tosidig H1, ønsker en ensidig H1: $\beta_1 < \beta_2$ blir $p = 0,142199/2 \approx 0,071$.

**Med pakken `multcomp`** Med pakken `multcomp` er det enkelt å formulere ensidige hypoteser også. Denne er kanskje den enkleste å bruke.

```r
library(multcomp)
# Specify the linear hypothesis
glht_mod <- glht(
  model = lm_2y,
  linfct = c("jc - univ <= 0")
)

# Inspect summary
summary(glht_mod)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = mod_2y, data = twoyear)
##
## Linear Hypotheses:
##                 Estimate Std. Error t value Pr(>t)
## jc - univ <= 0 -0.010180   0.006936  -1.468  0.929
## (Adjusted p values reported -- single-step method)
```

$\mathsf{Pr}(< t) = 1 - \mathsf{Pr}(> t) = 1 - 0,929 = 0,071$

```r
# Inspect confidence interval
confint(glht_mod)
```

```
##
##   Simultaneous Confidence Intervals
##
## Fit: lm(formula = mod_2y, data = twoyear)
##
## Quantile = -1.6451
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##                Estimate lwr      upr
## jc - univ <= 0 -0.01018 -0.02159      Inf
```

Vi ser at konfidensintervallet inneholder null så vi kan ikke forkaste at $\beta_1 = \beta_2$, dvs. vi kan ikke forkaste at jc gir samme uttelling som univ på 5% nivå.

## Gjennomgangseksemplet avsnitt 4-5

Baseball (mlb1). Vi trenger ikke forstå baseball for å kunne forstå eksemplet.

```
data(mlb1)
mod_bb <- "log(salary) ~ years + gamesyr + bavg + hrunsyr + rbisyr"
lm_bb <- lm(mod_bb, data = mlb1)
```

Historien er da at variablene `bavg, hrunsyr, rbisyr` angir spillernes individuelle ferdigheter. Spørsmålet er om dette betyr noe for lønn eller om det bare er hvor lenge en har spillet (years) og gjennomsnittlig antall kamper per år en har fått spille (gamesyr) som bestemmer lønnsnivået.

```
summary(lm_bb)
```

```
##
## Call:
## lm(formula = mod_bb, data = mlb1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.02508 -0.45034 -0.04013  0.47014  2.68924
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.119e+01  2.888e-01  38.752  < 2e-16 ***
## years       6.886e-02  1.211e-02   5.684 2.79e-08 ***
## gamesyr     1.255e-02  2.647e-03   4.742 3.09e-06 ***
## bavg        9.786e-04  1.104e-03   0.887    0.376
## hrunsyr     1.443e-02  1.606e-02   0.899    0.369
## rbisyr      1.077e-02  7.175e-03   1.500    0.134
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7266 on 347 degrees of freedom
## Multiple R-squared:  0.6278, Adjusted R-squared:  0.6224
## F-statistic: 117.1 on 5 and 347 DF,  p-value: < 2.2e-16
```

Det vi ønsker å teste er om individuelle ferdigheter er overflødig i modellen, dvs om $\beta_3 = 0$, $\beta_4 = 0$ og $\beta_5 = 0$.

Læreboken gjør dette «manuelt» vha. SSR fra restricted og unrestricted model.

```
mod_bb_r <- "log(salary) ~ years + gamesyr"
lm_bb_r <- lm(mod_bb_r, data = mlb1)
```

```
summary(lm_bb_r)
```

```
##
## Call:
## lm(formula = mod_bb_r, data = mlb1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.66858 -0.46412 -0.01177  0.49219  2.68829
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.223804   0.108312 103.625  < 2e-16 ***
## years        0.071318   0.012505   5.703  2.5e-08 ***
## gamesyr      0.020174   0.001343  15.023  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7527 on 350 degrees of freedom
## Multiple R-squared:  0.5971, Adjusted R-squared:  0.5948
## F-statistic: 259.3 on 2 and 350 DF,  p-value: < 2.2e-16
```

```r
(ssr_u <- sum(residuals(lm_bb)^2))
```

```
## [1] 183.1863
```

```r
(ssr_r <- sum(residuals(lm_bb_r)^2))
```

```
## [1] 198.3115
```

F verdien blir da (n=353 obs, k =5 og q = 3

```r
(F_bb <- (ssr_r -ssr_u)/ssr_u * ((353-5-1)/3))
```

```
## [1] 9.550254
```

F-verdien kan så sjekkes opp mot tabell eller

```r
pf(9.5503, 3, 347, lower.tail = FALSE)
```

```
## [1] 4.473429e-06
```

Kritisk verdi 1%

```r
qf(c(0.005, 0.995), 3, 347)
```

```
## [1] 0.02387536 4.35317230
```

Vi ser at vi kan forkaste hypotesen om at individuelle ferdigheter ikke har betydning for lønnen.

```r
linearHypothesis(lm_bb, c("bavg = 0", "hrunsyr = 0", "rbisyr = 0"))
```

**Samme med bruk av `linearHypothesis`**

```
## Linear hypothesis test
##
## Hypothesis:
## bavg = 0
## hrunsyr = 0
## rbisyr = 0
##
## Model 1: restricted model
## Model 2: log(salary) ~ years + gamesyr + bavg + hrunsyr + rbisyr
##
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    350 198.31
## 2    347 183.19  3    15.125 9.5503 4.474e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Specify the linear hypothesis
glht_mod <- glht(
  model = lm_bb,
  linfct = c("bavg + hrunsyr + rbisyr = 0")
)

# Inspect summary
summary(glht_mod)
```

**Samme med bruk av `multcomp`**

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = mod_bb, data = mlb1)
##
## Linear Hypotheses:
##                                Estimate Std. Error t value Pr(>|t|)
## bavg + hrunsyr + rbisyr == 0  0.02617    0.01045   2.506   0.0127 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

`multcomp` bruker her en mer avansert simultan test (som jeg ikke tror vi skal bekymre oss om å forstå nå), men konklusjonen blir den samme.

**Ex. 4.9**

```r
data(bwght)
summary(bwght[,4:7])
```

```
##      bwght          fatheduc        motheduc          parity
```

```
##  Min.   : 23.0   Min.   : 1.00   Min.   : 2.00   Min.   :1.000
##  1st Qu.:107.0   1st Qu.:12.00   1st Qu.:12.00   1st Qu.:1.000
##  Median :120.0   Median :12.00   Median :12.00   Median :1.000
##  Mean   :118.7   Mean   :13.19   Mean   :12.94   Mean   :1.633
##  3rd Qu.:132.0   3rd Qu.:16.00   3rd Qu.:14.00   3rd Qu.:2.000
##  Max.   :271.0   Max.   :18.00   Max.   :18.00   Max.   :6.000
##                  NA's   :196     NA's   :1
```

Vi har 196 NA i `fatheduc` og 1 i `motheduc`. Vi velger å jobbe med komplette observasjoner.

```r
mod_bw <- "bwght ~ cigs + parity + faminc + motheduc +fatheduc"
lm_bw <- lm(mod_bw, data = bwght[complete.cases(bwght),])
```

```r
summary(lm_bw)
```

```
##
## Call:
## lm(formula = mod_bw, data = bwght[complete.cases(bwght), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -95.796 -11.960   0.643  12.679 150.879
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 114.52433    3.72845  30.716  < 2e-16 ***
## cigs         -0.59594    0.11035  -5.401 8.02e-08 ***
## parity        1.78760    0.65941   2.711  0.00681 **
## faminc        0.05604    0.03656   1.533  0.12559
## motheduc     -0.37045    0.31986  -1.158  0.24702
## fatheduc      0.47239    0.28264   1.671  0.09492 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.79 on 1185 degrees of freedom
## Multiple R-squared:  0.03875,    Adjusted R-squared:  0.03469
## F-statistic: 9.553 on 5 and 1185 DF,  p-value: 5.986e-09
```

```r
linearHypothesis(lm_bw, "motheduc +  fatheduc = 0")
```

```
## Linear hypothesis test
##
## Hypothesis:
## motheduc  + fatheduc = 0
##
## Model 1: restricted model
## Model 2: bwght ~ cigs + parity + faminc + motheduc + fatheduc
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   1186 464090
## 2   1185 464041  1    49.257 0.1258 0.7229
```

Dropper å kjøre den simultane testen fra `multcomp`.

```
# Specify the linear hypothesis
glht_mod_bw <- glht(
  model = lm_bw,
  linfct = c("motheduc +  fatheduc = 0")
)

# Inspect summary
summary(glht_mod_bw)
```

Mor og fars utdannelse blir ikke-signifikant når variablene `cigs`, `parity` og `faminc` blir inkludert.

# Oppgavene

## Oppgave 4C1

i. Tolkning $\beta_1$?

   Holder alle andre variabler enn $\mathsf{expendA}$ fast. Gir oss

$$\Delta\mathsf{VoteA} = \beta_1 \mathsf{log}(\mathsf{expendA}) = \frac{\beta_1}{100}(100\Delta\mathsf{log}(\mathsf{expendA})) \approx \frac{\beta_1}{100}\%\Delta\mathsf{expendA}$$

   Altså gir $\beta_1$ oss tilnærmet antall prosentpoeng økning i $\mathsf{voteA}$ når $\mathsf{expendA}$ øker med 1%. Altså antall prosentpoeng økning (f.eks fra 12,1% til 12,7%, dvs. 0,6 prosentpoeng økning) når vi øker $\mathsf{expendA}$ med 1% (f.eks fra 20 millioner til $20 \cdot 1,01 = 20,2$ millioner).

   Eksempeltallene er selvsagt tatt rett ut av løse luften som en illustrasjon. Et viktig poeng er at den første størrelsen er *prosentpoeng* mens den andre er en relativ størrelse (*prosentvis endring*). Disse to begrepene blandes ofte.

ii. H0: $\beta_1 = -\beta_2$ eller H0: $\beta_1 + \beta_2 = 0$.

iii. Hvis expendA og expendB økes med samme prosentvise størrelse (f.eks fra 10 mill. til 10,5 mill. for A og fra 30 mill. til 31,5 mill. for B) vil As andel av stemmene være uendret?

```
# load dataset vote1 from wooldridge package
data(vote1)
mod1 <- "voteA ~ log(expendA) + log(expendB) + prtystrA"
lm1 <- lm(mod1, data=vote1)
```

```
summary(lm1)
```

```
##
## Call:
## lm(formula = mod1, data = vote1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.3968  -5.4174  -0.8679   4.9551  26.0660
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  45.07893    3.92631   11.48   <2e-16 ***
## log(expendA)  6.08332    0.38215   15.92   <2e-16 ***
## log(expendB) -6.61542    0.37882  -17.46   <2e-16 ***
## prtystrA      0.15196    0.06202    2.45   0.0153 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.712 on 169 degrees of freedom
## Multiple R-squared:  0.7926, Adjusted R-squared:  0.7889
## F-statistic: 215.2 on 3 and 169 DF,  p-value: < 2.2e-16
```

Vi ser fra summary at de estimerte koeffisientene for `log(expendA)` er 6.08332 og -6.61542 for `log(expendB)`. En økning på 1% i expendA vil altså gi $6{,}1/100 = 0{,}0608$ prosentpoeng økning i andelen stemmer for kandidat A. Likeledes vil 1% økning i expendB, alle andre variabler holdt fast, gi en reduksjon på $6{,}62/100 = 0{,}0662$ prosentpoeng i kandidat As andel av stemmene. Vi kan ikke teste hypotesen fra ii) utfra resultatene ovenfor siden vi ikke kjenner $\mathsf{se}(\beta_1 - \beta_2)$.

iv. For å teste hypotesen fra ii) må vi skrive om modellen med $\theta_1 = \beta_1 + \beta_2$ som gir $\beta_1 = \theta_1 - \beta_2$. Vi setter inn i modellen og får

$$\mathsf{voteA} = \beta_0 + (\theta_1 - \beta_2)\mathsf{log(expendA)} + \beta_2\mathsf{log(expendB)} + \beta_3\mathsf{prtystrA}$$

som gir oss

$$\mathsf{voteA} = \beta_0 + \theta_1\mathsf{log(expendA)} + \beta_2(\mathsf{log(expendB)\ \text{-}\ log(expendA)}) + \beta_3\mathsf{prtystrA}$$

Vi kan nå kjøre en standard regresjon på denne modellen og standard error for koeffisienten til `log(expendA)` vil være $\mathsf{se}(\beta_1 - \beta_2)$ som vi manglet.

```r
# Merk bruk av I() funksjonen. Inne i denne virker +, * etc som
# vanlig og ikke som operasjoner i Rs formula språk
mod2_r <- "voteA ~ log(expendA) + I(log(expendA) - log(expendB)) + prtystrA"
lm2_r <- lm(mod2_r, data=vote1)
```

```r
summary(lm2_r)
```

```
##
## Call:
## lm(formula = mod2_r, data = vote1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.3968  -5.4174  -0.8679   4.9551  26.0660
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     45.07893    3.92631  11.481   <2e-16 ***
## log(expendA)                    -0.53210    0.53309  -0.998   0.3196
## I(log(expendA) - log(expendB))  6.61542    0.37882  17.463   <2e-16 ***
## prtystrA                         0.15196    0.06202   2.450   0.0153 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.712 on 169 degrees of freedom
## Multiple R-squared:  0.7926, Adjusted R-squared:  0.7889
## F-statistic: 215.2 on 3 and 169 DF,  p-value: < 2.2e-16
```

Da får vi at $t = \frac{-0.53210}{0.53309} = -0.9981429$. Vi kan altså ikke forkaste H0.

Gjøre det samme «automagisk» i R

```
linearHypothesis(lm1, "log(expendA) + log(expendB) = 0")
```

```
## Linear hypothesis test
##
## Hypothesis:
## log(expendA)  + log(expendB) = 0
##
## Model 1: restricted model
## Model 2: voteA ~ log(expendA) + log(expendB) + prtystrA
##
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1    170 10111
## 2    169 10052  1    59.261 0.9963 0.3196
```

Her altså F-test, mens t-ovenfor. Merk at $t^2 = F$. Så $-0.9981429^2 = 0,99629$.

## Oppgave 4C3

i)

```
mod_hp_1 <- "log(price) ~ sqrft + bdrms"
lm_hp_1 <- lm(mod_hp_1, data = hprice1)
```

```
summary(lm_hp_1)
```

```
##
## Call:
## lm(formula = mod_hp_1, data = hprice1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.75448 -0.12322 -0.01993  0.11938  0.62948
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.766e+00  9.704e-02  49.112  < 2e-16 ***
## sqrft       3.794e-04  4.321e-05   8.781  1.5e-13 ***
## bdrms       2.888e-02  2.964e-02   0.974    0.333
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.1971 on 85 degrees of freedom
## Multiple R-squared:  0.5883, Adjusted R-squared:  0.5786
## F-statistic: 60.73 on 2 and 85 DF,  p-value: < 2.2e-16
```

Dette gir oss at $\theta_1 = 150\beta_1 + \beta_2 = 150 \cdot 0.0003794 + 0.02888 = 0,0858$. Dvs. prisen øker med 8,58%.

ii) Vi har $\theta_1 = 150\beta_1 + \beta_2$ som gir $\beta_2 = \theta_1 - 150\beta_1$. Setter inn for $\beta_2$ og får

$$\log(price) = \beta_0 + \beta_1 \mathsf{sqrft} + (\theta_1 - 150\beta_1)\mathsf{bdrms} + u = \beta_0 + \theta_1 \mathsf{bdrms} + \beta_1(\mathsf{sqrft} - 150\mathsf{bdrms}) + u$$

```
# hprice1
mod_hp <- "log(price) ~ bdrms + I(sqrft - 150 * bdrms)"
lm_hp <- lm(mod_hp, data = hprice1)
```

```
summary(lm_hp)
```

```
##
## Call:
## lm(formula = mod_hp, data = hprice1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.75448 -0.12322 -0.01993  0.11938  0.62948
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            4.766e+00  9.704e-02  49.112  < 2e-16 ***
## bdrms                  8.580e-02  2.677e-02   3.205   0.0019 **
## I(sqrft - 150 * bdrms) 3.794e-04  4.321e-05   8.781  1.5e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1971 on 85 degrees of freedom
## Multiple R-squared:  0.5883, Adjusted R-squared:  0.5786
## F-statistic: 60.73 on 2 and 85 DF,  p-value: < 2.2e-16
```

Ser at $\theta_1 = 8.580e - 02 = 8.580 \cdot 10^{-2} = 0,0858$.

iii) Er nok $\theta_1$ og *ikke* $\theta_2$ som menes. Ser at standard error er $4.321 \cdot 10^{-05}$.

Vi gjør det enkelt og finner konfidensintervall vha. `confint`

```
confint(lm_hp)
```

```
##                               2.5 %       97.5 %
## (Intercept)            4.5730767914 4.9589776356
## bdrms                  0.0325803713 0.1390223615
## I(sqrft - 150 * bdrms) 0.0002935289 0.0004653631
```

altså fra 3,258% til 13,902%.