ST1201 Statistiske metoder, høsten 2018

Bruk av R

Mette Langaas, Institutt for matematiske fag

November 2018

Contents

(Siste endringer: 07.11.2018)

R og RStudio

- R laster du ned fra CRAN og velger Windows, Linux eller Mac. https://cran.r-project.org/
- I tillegg laster du ned og installerer Rstudio - som er den IDEen vi nå bruker mest https://www.rstudio.com/products/rstudio/download/ du velger den som heter FREE helt til venstre.
- Hvis du trenger hjelp med dette må du ta kontakt med orakel@ntnu.no

RStudio - hvordan bruke?

Når du starter RStudio har du trolig disse fire vinduene:

- **Source** (aka script window) - upper left window: where you write your code and keep track of your work. Hvis du ikke ser denne så trykk på bildet av et rektangel (ved siden av det kortere rektangel) eller velg File-New File for å lage et script.
- **Console** - lower left window: where the R commands are executed (so here is where you R installation lives). Sometimes also referred to as command window.
- **Environment/History/Connections/Presentation/Git** - upper right window: the objects that you have in your workspace, and the commands you have executed, and more.
- **Files/Plots/Packages/Help/Viewer**- lower right: overview of your files, the plots you produce, the packages you have installed and loaded, and more.

Source window: (Make the source window active.) To start writing a script press File- New File- R Script. To open an existing file, press File- Open File- and select the file you want to open. The file will open in the source window. To save this file, press File- Save as- and go to the working directory there you want to put your R files and save the file as "name".R (example: `myRintro.R`). Files with R code usually have extension `.R`.

Alternativt kunne du ha laget et R prosjekt - det er kanskje enklere og lurere! Vi gjør det sammen!

Console window: (Make the console window active.) To see your working directory, you can write `getwd()`, and you will get your location as output. You can also set your working directory to a certain folder of choise by writing `setwd("location")` (Example: `setwd("M:/Documents/ST1201/")`). Now you are certain that all your files will be put in this folder.

Hvis du allerede har laget et prosjekt så husk å legge filer i prosjektet (vi gjør det sammen).

Quitting: It is always important to be able to quit a program: when you are finished you may choose RStudio-Quit Rstudio (top menu outside of the windows). Alternatively, you may write `q()`in the console window to quit R (the parenthesis is because q is a function that can have arguments to be given within the parentheses and you call the function without any arguments). You will be asked if you want to save your script and workspave. If you want to reuse your script later (and of cause you want to do that - we aim at reproduciable research), you should save it! If you answer yes to "Save workspace image" all the objects you have created are found in a `.RData` file (more about objects soon). This could be useful if you don't want to run all the commands in the script again, because if you start R in the same working directory all the objects you have created will be automatically availble to you. More on objects next.

Jukseark: **RStudio IDE cheat sheet**: https://github.com/rstudio/cheatsheets/raw/master/rstudio-ide.pdf

Grunnleggende

Objekter

En kalkulator? En tabell?

```
2 + 2
rnorm(100)
pnorm(0, 0, 1)
qnorm(0.3)
qnorm(0.5)
```

Tilordning

= og <- er det samme. Historisk grunn til at mange bruker den siste (da den første ikke var lov i "gamle dager").

Vektorisert evaluering.

```
x = 5
vekt = c(40, 50, 70, 80, 90)
hoyde = c(1.3, 1.7, 1.9, 1.8)
bmi = vekt/hoyde^2
```

Funksjoner

Vi har allerede sett `rnorm`, `pnorm` og `qnorm`- de er alle funksjoner - som tar argumenter.

Hjelp

Hvis du vet hva en funksjon heter - kanskje `t.test`men lurer på hva den gjør, så kan du bare skrive

```
`?`(t.test)
help(t.test)
```

da vil en hjelpeside dukke opp i nedre høyre vindu hvis du trykker på fanen som heter `Help`.

Navngiving

- CamelCase: writing compound words such that each word in the middle of the phrase begins with a capital letter, with no intervening spaces or punctuation. "iPhone", "eBay", "FedEx", "DreamWorks", and "HarperCollins".
- snake case: writing compound words where the elements are separated with one underscore character (_) and no spaces, with each element's initial letter usually lowercased within the compound and the first letter either upper- or lowercase—as in "foo_bar" and "Hello_world".

Pakker

Det er ikke alle datasett og funksjoner som man har bruk for som allerede finnes inne i R-basispakken. Det betyr at det er noen såkalte `pakker` man må installere. Da kan du skrive i Console-vinduet

```
install.packages("gamlss.data")
```

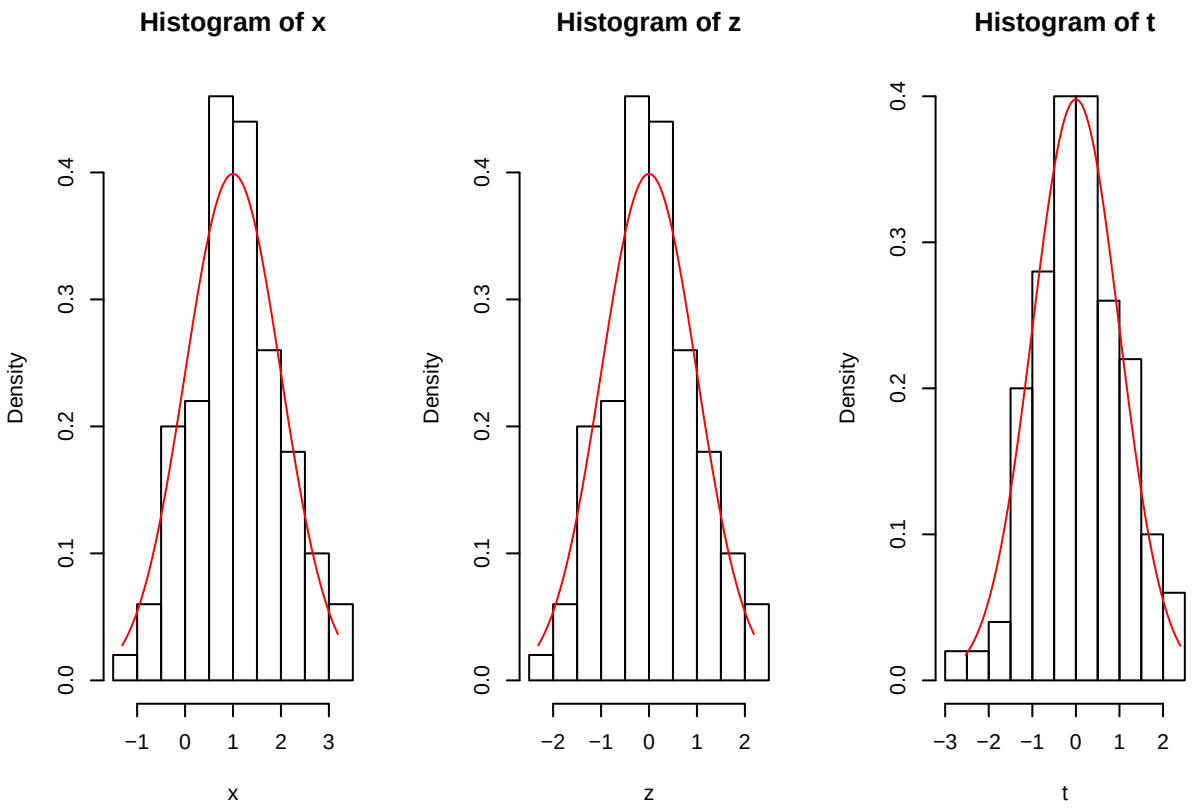
for en pakke som inneholder data vi skal bruke under. For å gjøre pakken tilgjengelig må vi gjøre den aktiv ved å skrive `library("gamlss.data")` eller gå til Packages fanen nede til høyre, velge pakken ved å krysse av. Da ser du i Console-vinduet at pakken er blitt lagt til.

Kapittel 7: Inferens i normalfordelingen

7.2-7.3 Trekke fra normalfordelingen

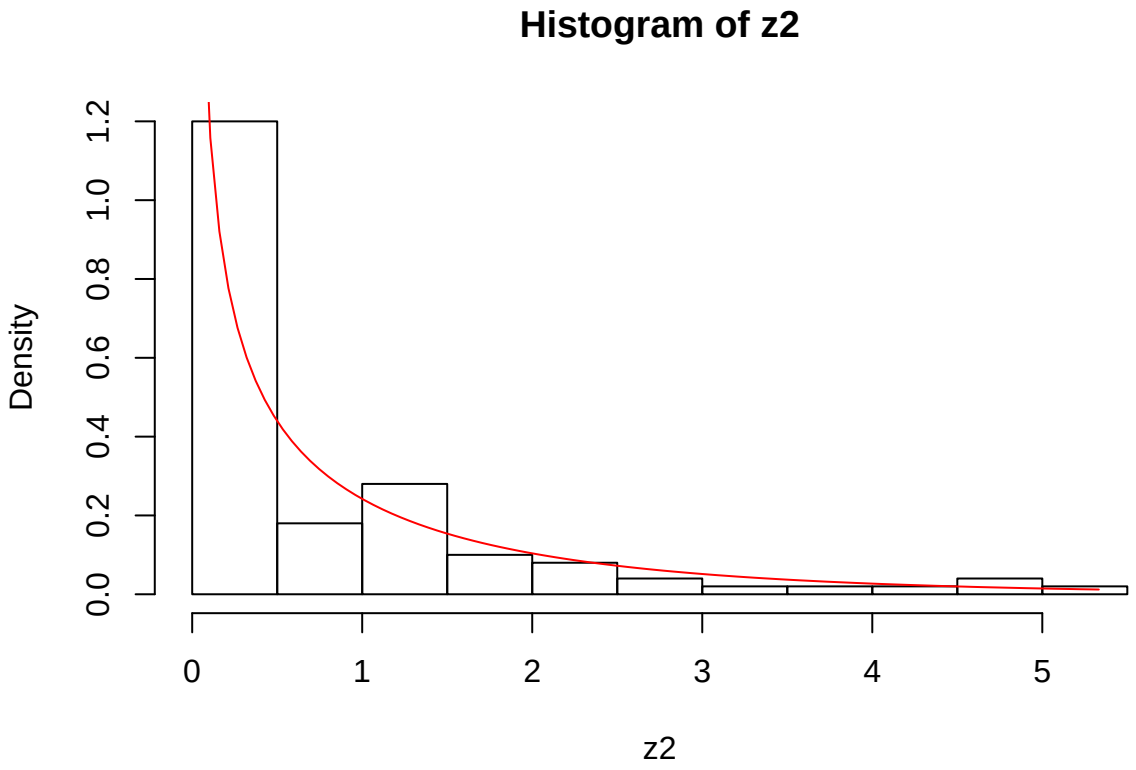
Normal og t-fordeling.

```
set.seed(123)
n = 100
mu = 1
sigma = 1
x = rnorm(n, mean = mu, sd = sigma)
z = (x - mu)/sigma
t = (x - mu)/sd(x)
par(mfrow = c(1, 3))
hist(x, freq = FALSE)
mydnorm = function(x) return(dnorm(x, mean = mu, sd = sigma))
curve(mydnorm, min(x), max(x), col = 2, add = TRUE)
hist(z, freq = FALSE)
curve(dnorm, min(z), max(z), col = 2, add = TRUE)
hist(t, freq = FALSE)
mydt = function(x) return(dt(x, df = n - 1))
curve(mydt, min(t), max(t), col = 2, add = TRUE)
```

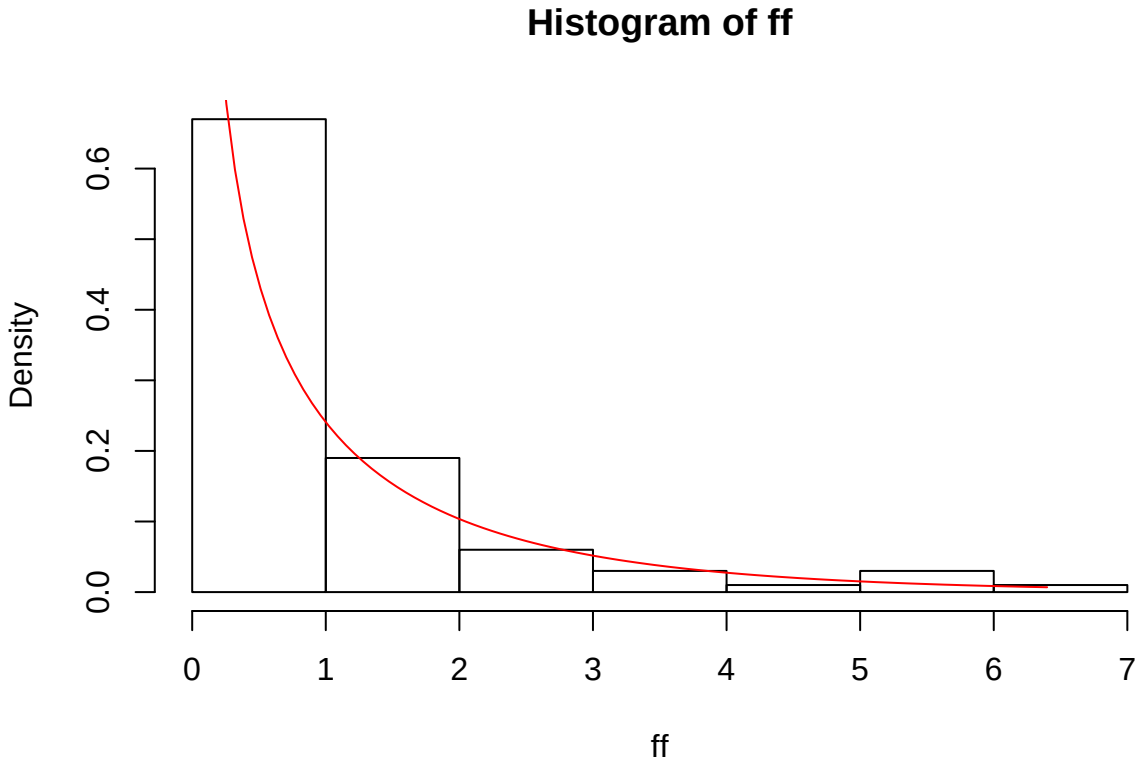


Så over til kjikvadrat og F.

```
z2 = z^2
hist(z2, freq = FALSE)
mychi2 = function(x) return(dchisq(x, df = 1))
curve(mychi2, min(z2), max(z2), col = 2, add = TRUE)
```




```
ff = t^2
hist(ff, freq = FALSE)
myf = function(x) return(df(x, 1, n - 1))
curve(myf, min(ff), max(ff), col = 2, add = TRUE)
```



Bruk av `ggplot` kan du lese om i https://www.math.ntnu.no/emner/TMA4268/2018v/1Intro/Rintermediate.

html.

7.4 Inferens om μ

t-test i ett utvalg - både test og konfidensintervall

```
set.seed(123)
x1 = rnorm(100, 10, 1)
t.test(x1, mu = 10)
t.test(x1, alternative = "greater", mu = 10)
```

```
##
##  One Sample t-test
##
## data:  x1
## t = 0.99041, df = 99, p-value = 0.3244
## alternative hypothesis: true mean is not equal to 10
## 95 percent confidence interval:
##   9.909283 10.271528
## sample estimates:
## mean of x
##  10.09041
##
##
##  One Sample t-test
##
## data:  x1
## t = 0.99041, df = 99, p-value = 0.1622
## alternative hypothesis: true mean is greater than 10
## 95 percent confidence interval:
##  9.938843      Inf
## sample estimates:
## mean of x
##  10.09041
```

Areal i haler av `t`-fordeling med `pt`.

7.5 Inferens om σ^2

Vi har ikke noe egen funksjon for intervall og hypotesetest i base R, men det skal være en funksjon `varTest` i pakken `EnvStat` (men den har jeg ikke brukt.) Test den ut selv!

Vi kan god bare programmere det selv.

```
set.seed(1201)
n = 100
mu = 0
sigma0 = 1
y = rnorm(n, mean = mu, sd = sigma)
testobs = (var(y) * (length(y) - 1))/sigma0
pchisq(testobs, length(y) - 1, lower.tail = var(y) < sigma0)  #ensidig test
```

```
## [1] 0.186913
```

Areal i haler av kjikvadratfordelingen med `pchisq`.

Kapittel 9: Inferens med to utvalg

9.1 Test av to forveningsverdier: t-test - med like eller ulike varianser

```
# Copper wires: two sample t-test with equal variances
dsA <- c(5179, 5203, 5207, 5195, 5207, 5202, 5203, 5208, 5216, 5193)
dsB <- c(5190, 5159, 5153, 5206, 5168, 5186, 5194, 5200)
t.test(dsA, dsB, var.equal = TRUE)
t.test(dsA, dsB)  # Welch dfs
```

```
##
##  Two Sample t-test
##
## data:  dsA and dsB
## t = 2.702, df = 16, p-value = 0.01571
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   4.157857 34.442143
## sample estimates:
## mean of x mean of y
##    5201.3    5182.0
##
##
##  Welch Two Sample t-test
##
## data:  dsA and dsB
## t = 2.5242, df = 10.001, p-value = 0.03016
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   2.264261 36.335739
## sample estimates:
## mean of x mean of y
##    5201.3    5182.0
```

9.3 Test av to varianser: F-test

```
# equal variances, with two variances
var.test(dsA, dsB)
```

```
##
##  F test to compare two variances
##
## data:  dsA and dsB
## F = 0.27124, num df = 9, denom df = 7, p-value = 0.07304
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.05623615 1.13840339
## sample estimates:
## ratio of variances
##          0.2712392
```

9.4 Test av to binomiske sannsynligheter

Først en binomisk sannsynlighet (var i 6.3).

```
x = rbinom(1, size = 100, p = 0.2)
# sann er 0.2, men tester lik 0.3
binom.test(x, 100, 0.3)
```

```
##
##  Exact binomial test
##
## data:  x and 100
## number of successes = 18, number of trials = 100, p-value =
## 0.00849
## alternative hypothesis: true probability of success is not equal to 0.3
## 95 percent confidence interval:
##  0.1103112 0.2694771
## sample estimates:
## probability of success
##                   0.18
```

To andeler

```
# Larsen og Max side 478
x = c(34, 19)
n = c(40, 35)
prop.test(x, n)
prop.test(x, n, correct = FALSE)  # same as bottom of page 478
# alternatives:
mat = matrix(c(x, n - x), ncol = 2)
fisher.test(mat)  #eksakt via hypergeo, men ikke i pensum?
chisq.test(mat)
```

```
##
##  2-sample test for equality of proportions with continuity
##  correction
##
## data:  x out of n
## X-squared = 7.0781, df = 1, p-value = 0.007803
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.08165624 0.53262947
## sample estimates:
##    prop 1    prop 2
## 0.8500000 0.5428571
##
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  x out of n
## X-squared = 8.4952, df = 1, p-value = 0.003561
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.1084420 0.5058438
## sample estimates:
```

```
##    prop 1    prop 2
## 0.8500000 0.5428571
##
##
##  Fisher's Exact Test for Count Data
##
## data:  mat
## p-value = 0.005018
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##   1.437621 17.166416
## sample estimates:
## odds ratio
##   4.666849
##
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  mat
## X-squared = 7.0781, df = 1, p-value = 0.007803
```

9.5 er CI som inngår under hver test over.

Kapittel 10: Goodness of fit og kontingenstabeller

10.2 Multinomisk fordeling

`rmultinom(n,size,prob)` og `dmultinom`

Ikke noen spesielle funksjoner (så vidt jeg vet for GOF med kjente og ukjente parametere (10.2 og 10.3), men man kan lett programmere selv, eller sette inn forventingsverdier.

```r
observed = c(770, 230)  # observed frequencies
expected = c(0.75, 0.25)  # expected proportions
chisq.test(x = observed, p = expected)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  observed
## X-squared = 2.1333, df = 1, p-value = 0.1441
```

10.5 Kontingenstabeller

(bare teste uavhengighet ikke homogeneity)

Hårfarge og øyefarge-data: er de to uavhengige størrelser?

```r
library(kableExtra)
eyehair <- cbind(c(1, 1, 1, 2, 2, 2), c(1, 2, 3, 1, 2, 3), c(29, 12, 7, 10,
    12, 10))
colnames(eyehair) <- c("eye", "hair", "count")
eyehairtable <- rbind(eyehair[1:3, 3], eyehair[4:6, 3])
print(eyehairtable)
```

```
res = chisq.test(eyehairtable)
res$observed
res$expected
res
# ikke bruke chisq.test hvis sm<U+00E5> celletall, da b<U+00F8>r eksakt test
# heller brukes (ikke pensum)
fisher.test(eyehairtable)
```

```
##      [,1] [,2] [,3]
## [1,]   29   12    7
## [2,]   10   12   10
##      [,1] [,2] [,3]
## [1,]   29   12    7
## [2,]   10   12   10
##      [,1] [,2] [,3]
## [1,] 23.4 14.4 10.2
## [2,] 15.6  9.6  6.8
##
##  Pearson's Chi-squared test
##
## data:  eyehairtable
## X-squared = 6.8602, df = 2, p-value = 0.03238
##
##
##  Fisher's Exact Test for Count Data
##
## data:  eyehairtable
## p-value = 0.0335
## alternative hypothesis: two.sided
```

Blodtyper:

```
bloodtypetable <- rbind(c(176, 41, 19, 164), c(112, 16, 6, 66), c(48, 8, 4,
    40))
print(bloodtypetable)
chisq.test(bloodtypetable)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  176   41   19  164
## [2,]  112   16    6   66
## [3,]   48    8    4   40
##
##  Pearson's Chi-squared test
##
## data:  bloodtypetable
## X-squared = 8.2001, df = 6, p-value = 0.2238
```

Kapittel 11: Enkel lineær regresjon

11.2-11.3 Minste kvadratsums metode og lineær modell

(for 11.2 antar ikke normalfordelte støyledd)

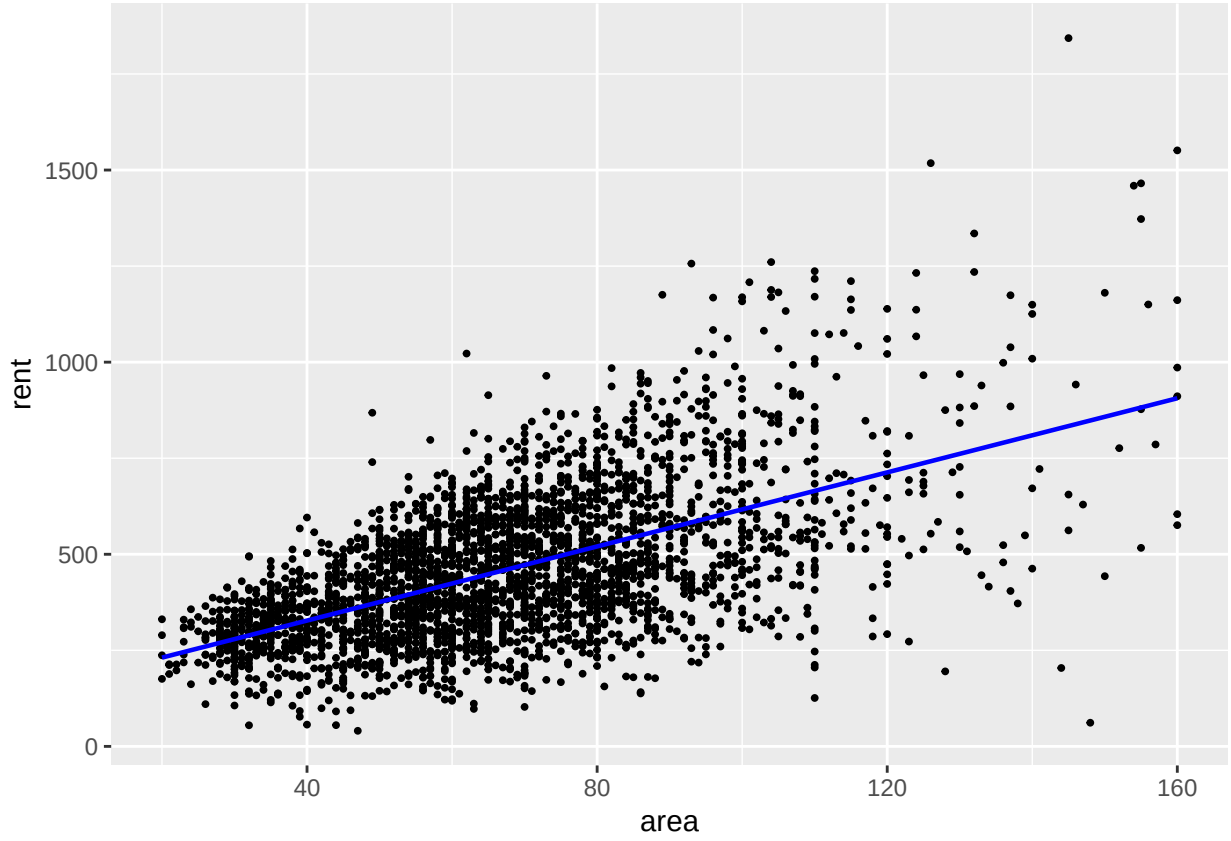
Bruk litt tid på summary-utskriften!

```
library("gamlss.data")  #datasettet med boligpriser i munchen

munich1.lm = lm(rent ~ area, data = rent99)  # tilpasser enkel line<U+00E6>r regresjon - modelformel
summary(munich1.lm)
munich1.beta0 = coef(munich1.lm)[1]
munich1.beta1 = coef(munich1.lm)[2]

library(ggplot2)  #for vakre plott, dere m<U+00E5> jo se noen av disse

ggplot(rent99, aes(x = area, y = rent)) + geom_point(size = 0.7) + geom_line(aes(x = area,
    y = munich1.beta0 + area * munich1.beta1), col = "blue", size = 0.8)
```

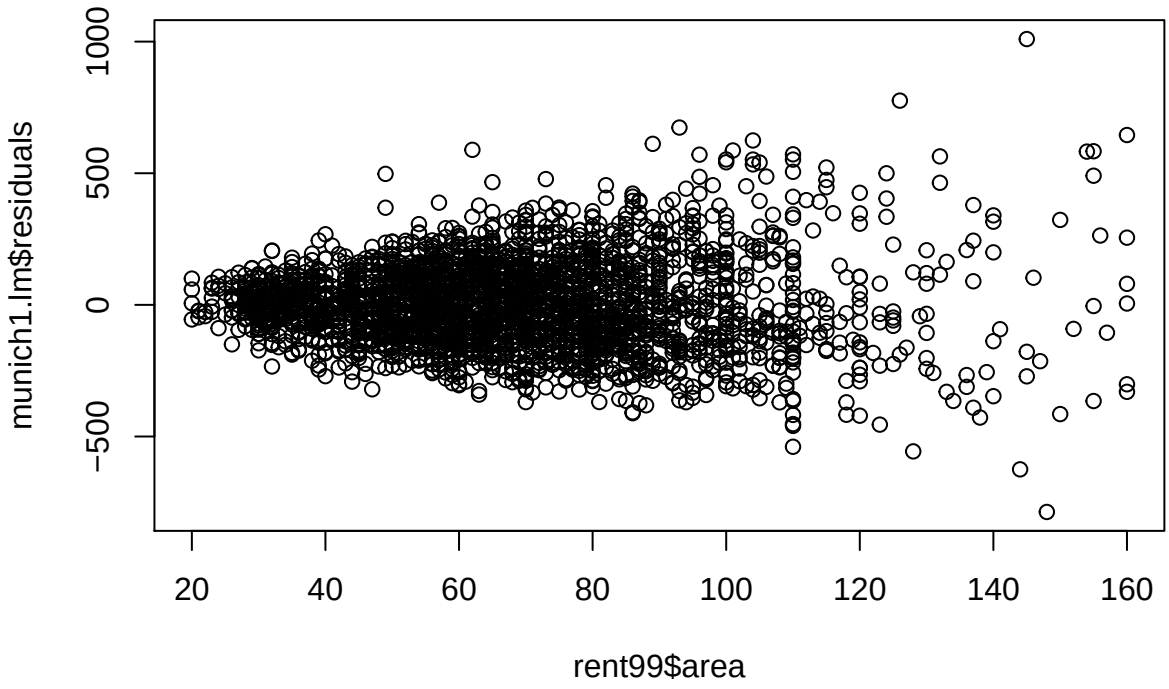



```
##
## Call:
## lm(formula = rent ~ area, data = rent99)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -786.63 -104.88   -5.69   95.93 1009.68
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 134.5922     8.6135   15.63   <2e-16 ***
## area          4.8215     0.1206   39.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

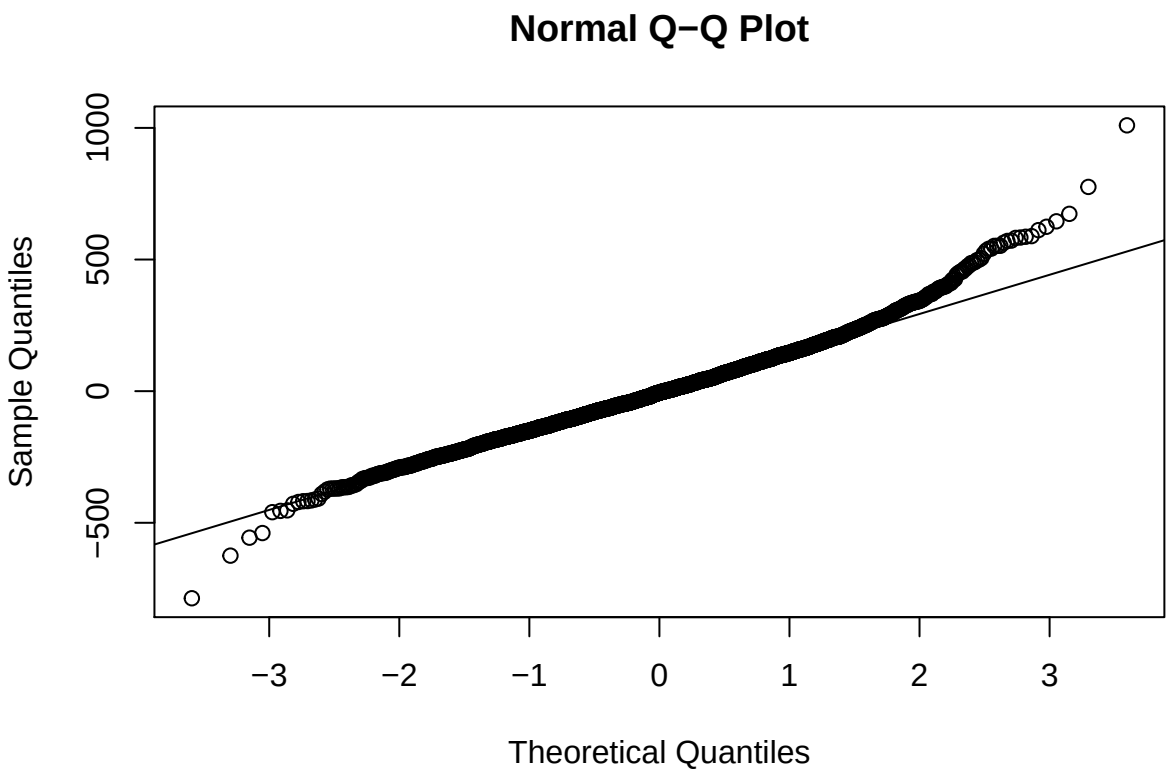
```
##
## Residual standard error: 158.8 on 3080 degrees of freedom
## Multiple R-squared:  0.3417, Adjusted R-squared:  0.3415
## F-statistic:  1599 on 1 and 3080 DF,  p-value: < 2.2e-16
```

Residualplott

```
plot(rent99$area, munich1.lm$residuals)
```




```
qqnorm(munich1.lm$residuals)
qqline(munich1.lm$residuals)
```




```
anova(munich1.lm)
confint(munich1.lm)  # konfidensintervall for beta0 og beta1
predict(munich1.lm, interval = "confidence")[1:10, ]  # 10 f<U+00F8>rste punkt for konfidensintervall f
predict(munich1.lm, interval = "prediction")[1:10, ]  # 10 f<U+00F8>rste boliger med predikerte verdier
```

```
## Analysis of Variance Table
##
## Response: rent
##             Df   Sum Sq  Mean Sq F value    Pr(>F)
## area         1 40299098 40299098  1598.5 < 2.2e-16 ***
## Residuals 3080 77646265    25210
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##                  2.5 %     97.5 %
## (Intercept) 117.703417 151.480972
## area          4.585017   5.057912
##         fit      lwr      upr
## 1  259.9503 248.6740 271.2265
## 2  269.5932 258.7247 280.4617
## 3  279.2361 268.7699 289.7024
## 4  279.2361 268.7699 289.7024
## 5  279.2361 268.7699 289.7024
## 6  279.2361 268.7699 289.7024
## 7  284.0576 273.7902 294.3250
## 8  284.0576 273.7902 294.3250
## 9  288.8791 278.8089 298.9492
## 10 293.7005 283.8259 303.5751
##         fit       lwr      upr
## 1  259.9503 -51.57151 571.4720
## 2  269.5932 -41.91409 581.1005
```

```
## 3  279.2361 -32.25739 590.7296
## 4  279.2361 -32.25739 590.7296
## 5  279.2361 -32.25739 590.7296
## 6  279.2361 -32.25739 590.7296
## 7  284.0576 -27.42931 595.5445
## 8  284.0576 -27.42931 595.5445
## 9  288.8791 -22.60140 600.3595
## 10 293.7005 -17.77368 605.1747
```

Her er det litt mer styr å finne ut hva hva alt gjør, f.eks. for å finne mer `predict` må du vite at du kan bruke `predict.lm` fordi vi jobber med objekter av klasse `lm`.

```
`?`(predict.lm)
```

11.4 Korrelasjonskoeffisient og R^2

```
cor(rent99$area, rent99$rent)^2
summary(munich1.lm)$r.squared
```

```
## [1] 0.341676
## [1] 0.341676
```

11.5 Bivariat normal

Tja, kan trekke data med `mvrnorm` fra `MASS` pakken, og regne sannsynligheter fra `mvtnorm`-pakken med ulike algoritmer for integrasjon.

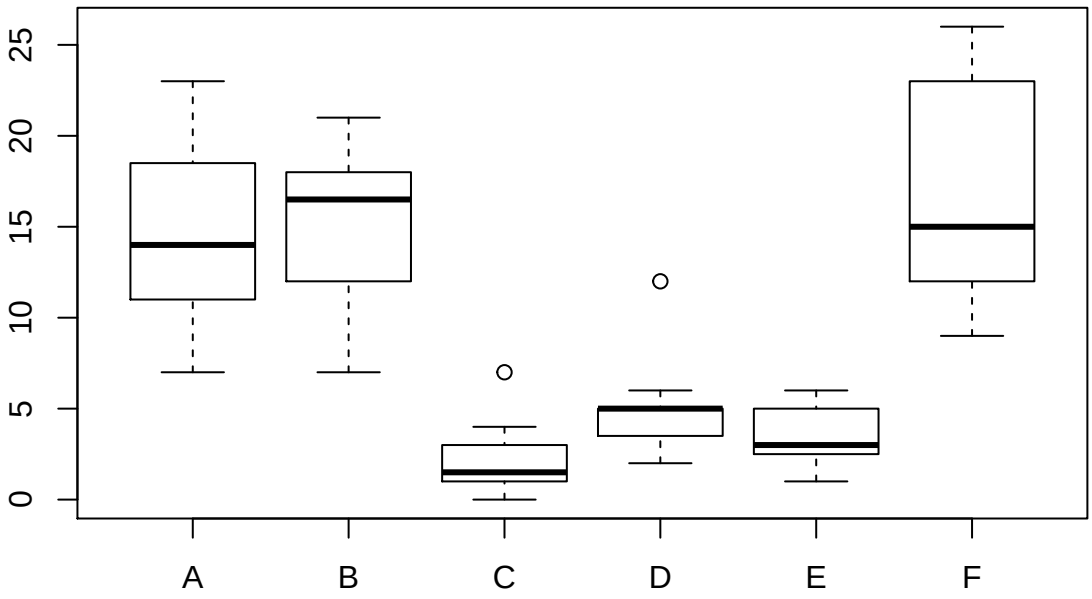
Test av korrelasjonen gjøres gjerne med kall til t-fordeling som på side 587, alternativt brukes Fishers tranform på side 588.

Kapittel 12: Enveis variansanalyse

12.2 F-test

Vi ser på et datasett med 6 ulike typer insektssprak og differanse mellom antall insekter før og etter at man har sprayet med sprayen. Er det forskjell på sprayene?

```
attach(InsectSprays)
boxplot(count ~ spray)
```




```
tapply(count, spray, mean)
tapply(count, spray, var)

oneway.test(count ~ spray, var.equal = TRUE)

res = aov(count ~ spray, data = InsectSprays)
summary(res)
```

```
##         A         B         C         D         E         F
## 14.500000 15.333333  2.083333  4.916667  3.500000 16.666667
##         A         B         C         D         E         F
## 22.272727 18.242424  3.901515  6.265152  3.000000 38.606061
##
##  One-way analysis of means
##
## data:  count and spray
## F = 34.702, num df = 5, denom df = 66, p-value < 2.2e-16
##
##             Df Sum Sq Mean Sq F value Pr(>F)
## spray        5   2669   533.8    34.7 <2e-16 ***
## Residuals   66   1015    15.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Multippel testing

```
# Tukey posthoc
TukeyHSD(res)

# tests for homogeneity of variances
bartlett.test(count ~ spray, data = InsectSprays)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
```

```
##
## Fit: aov(formula = count ~ spray, data = InsectSprays)
##
## $spray
##            diff        lwr       upr     p adj
## B-A   0.8333333  -3.866075  5.532742 0.9951810
## C-A -12.4166667 -17.116075 -7.717258 0.0000000
## D-A  -9.5833333 -14.282742 -4.883925 0.0000014
## E-A -11.0000000 -15.699409 -6.300591 0.0000000
## F-A   2.1666667  -2.532742  6.866075 0.7542147
## C-B -13.2500000 -17.949409 -8.550591 0.0000000
## D-B -10.4166667 -15.116075 -5.717258 0.0000002
## E-B -11.8333333 -16.532742 -7.133925 0.0000000
## F-B   1.3333333  -3.366075  6.032742 0.9603075
## D-C   2.8333333  -1.866075  7.532742 0.4920707
## E-C   1.4166667  -3.282742  6.116075 0.9488669
## F-C  14.5833333   9.883925 19.282742 0.0000000
## E-D  -1.4166667  -6.116075  3.282742 0.9488669
## F-D  11.7500000   7.050591 16.449409 0.0000000
## F-E  13.1666667   8.467258 17.866075 0.0000000
##
##
##  Bartlett test of homogeneity of variances
##
## data:  count by spray
## Bartlett's K-squared = 25.96, df = 5, p-value = 9.085e-05
```

ANOVA fra `lm`

Jeg tilpasser alltid en regresjon, men med en faktor som eneste kovariat og `contr.sum`, og deretter bruke funksjonen `anova` på `lm`-objektet. Det krever at data er på individelt nivå og ikke i en variansanalysetabell.

```
fit = lm(count ~ spray, contrasts = list(spray = "contr.sum"), data = InsectSprays)
summary(fit)
fit$contrasts
anova(fit)
```

```
##
## Call:
## lm(formula = count ~ spray, data = InsectSprays, contrasts = list(spray = "contr.sum"))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.333 -1.958 -0.500  1.667  9.333
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.5000     0.4622  20.554  < 2e-16 ***
## spray1        5.0000     1.0335   4.838 8.22e-06 ***
## spray2        5.8333     1.0335   5.644 3.78e-07 ***
## spray3       -7.4167     1.0335  -7.176 7.87e-10 ***
## spray4       -4.5833     1.0335  -4.435 3.57e-05 ***
## spray5       -6.0000     1.0335  -5.805 2.00e-07 ***
## ---
```

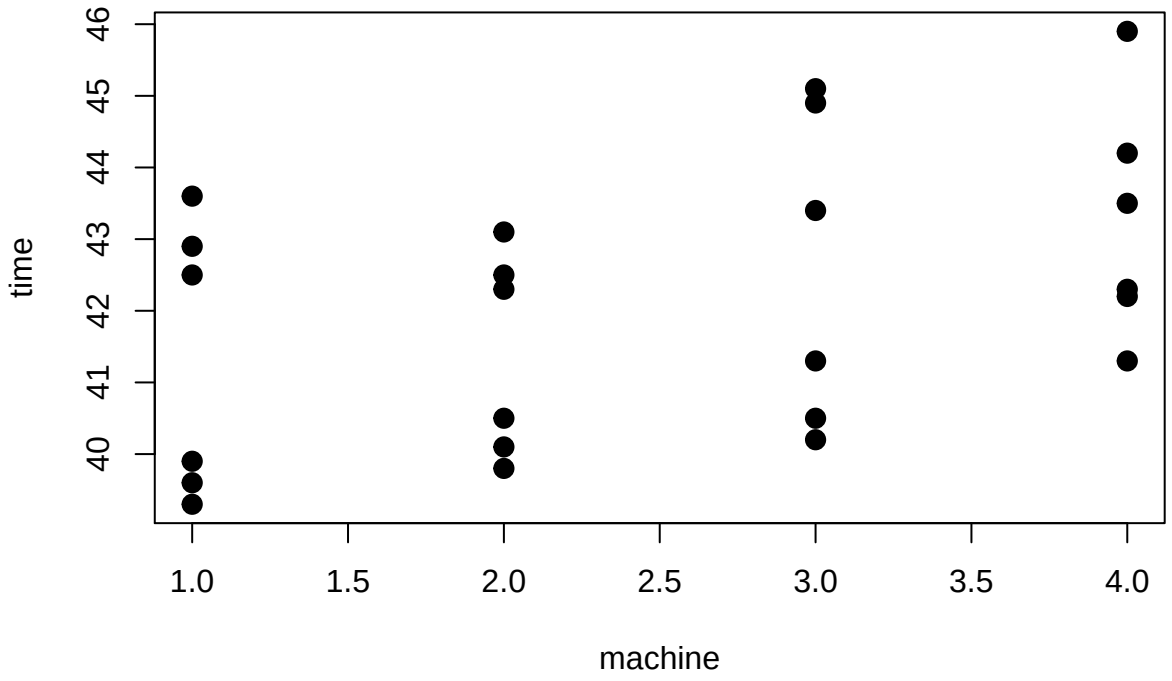
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.922 on 66 degrees of freedom
## Multiple R-squared:  0.7244, Adjusted R-squared:  0.7036
## F-statistic:  34.7 on 5 and 66 DF,  p-value: < 2.2e-16
##
## $spray
## [1] "contr.sum"
##
## Analysis of Variance Table
##
## Response: count
##           Df Sum Sq Mean Sq F value    Pr(>F)
## spray      5 2668.8  533.77  34.702 < 2.2e-16 ***
## Residuals 66 1015.2   15.38
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Observer at vi får de samme resultatene

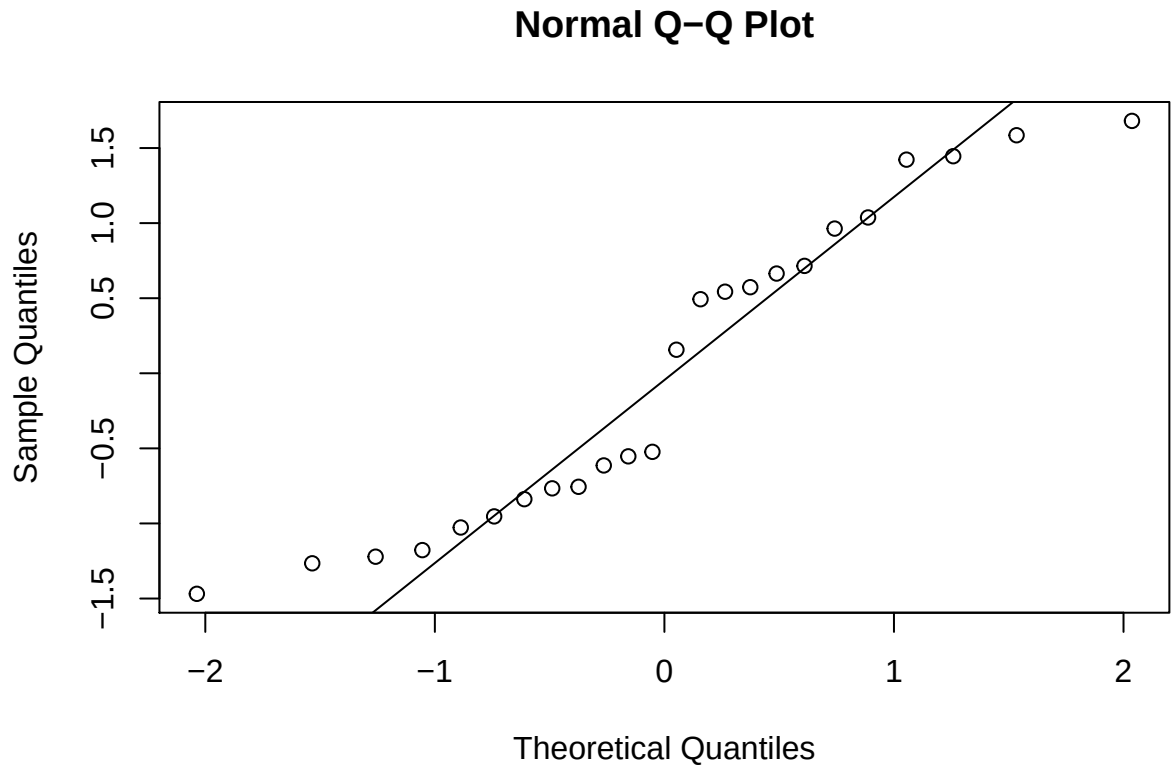
Kapittel 13: Randomiserte blokkdesign

Eksempel der vi har 6 ulike maskiner som vi lurer på om der ulike, men så tar vi ikke hensyn til at det er en blokk-effekt (operator av maskinen).

```r
ds <- matrix(c(42.5, 39.8, 40.2, 41.3, 39.3, 40.1, 40.5, 42.2, 39.6, 40.5, 41.3,
    43.5, 39.9, 42.3, 43.4, 44.2, 42.9, 42.5, 44.9, 45.9, 43.6, 43.1, 45.1,
    42.3), ncol = 6, nrow = 4)
dsmat <- data.frame(cbind(c(ds), rep(1:6, each = 4), rep(1:4, 6)))
colnames(dsmat) <- c("time", "operator", "machine")
fit <- lm(time ~ as.factor(machine), data = dsmat)
anova(fit)
plot(dsmat[, 1] ~ dsmat[, 3], pch = 20, ylab = "time", cex = 2, xlab = "machine")
```

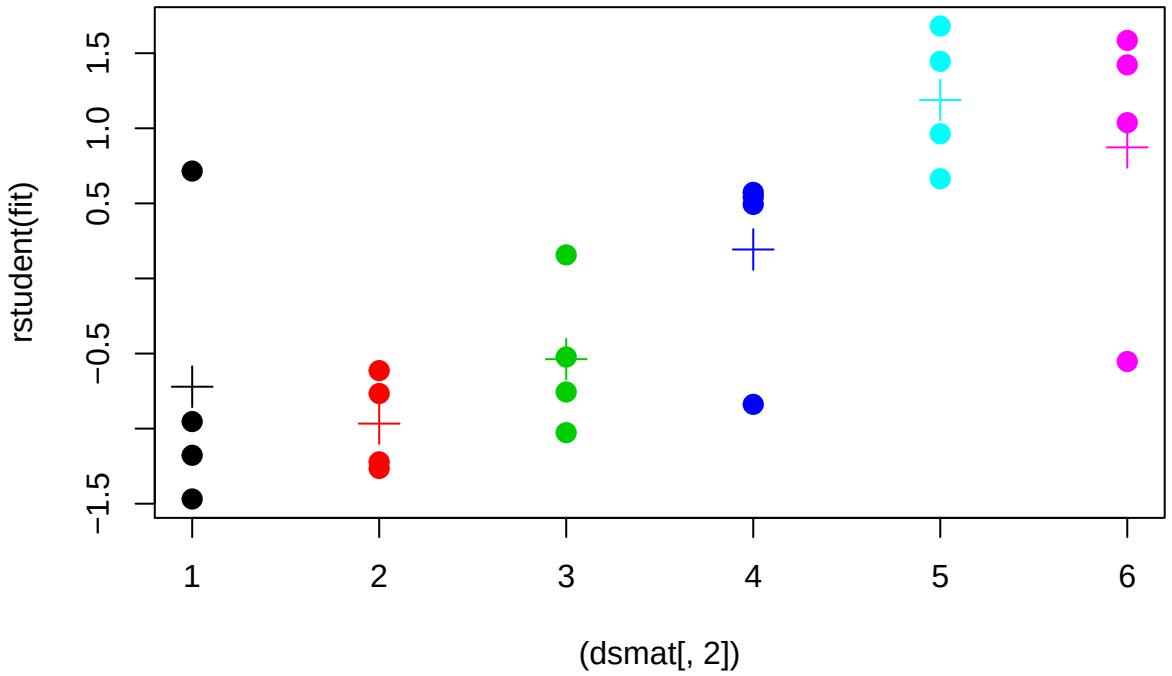



```
qqnorm(rstudent(fit))
qqline(rstudent(fit))
```

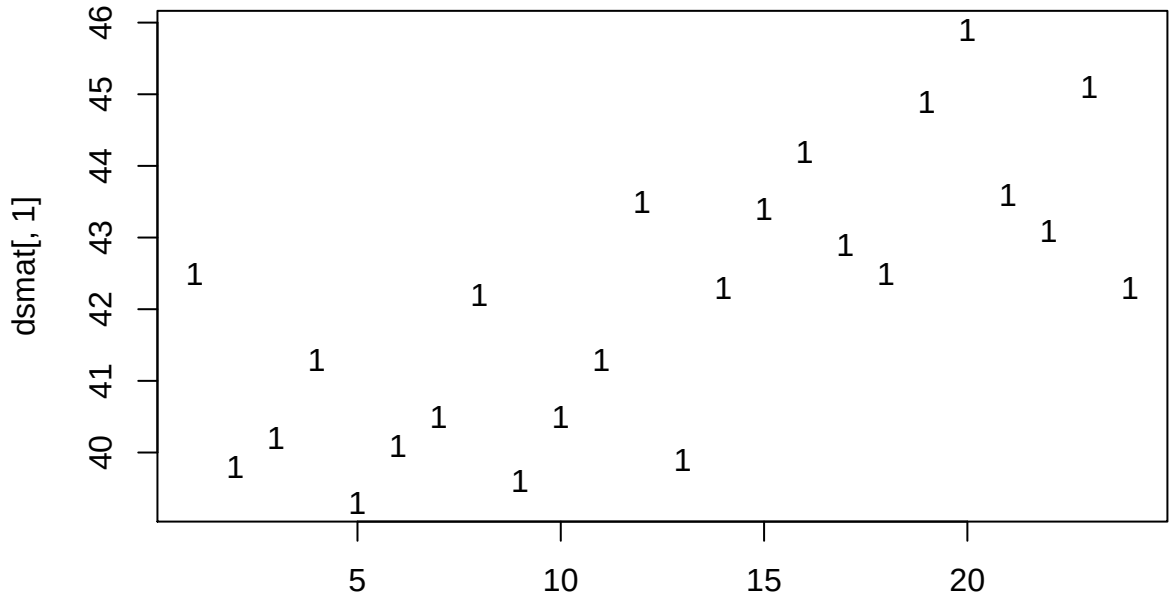



```
plot((dsmat[, 2]), rstudent(fit), pch = 20, cex = 2, col = rep(1:6, each = 4))
points(1, mean(rstudent(fit)[1:4]), col = 1, pch = 3, cex = 2)
points(2, mean(rstudent(fit)[5:8]), col = 2, pch = 3, cex = 2)
points(3, mean(rstudent(fit)[9:12]), col = 3, pch = 3, cex = 2)
points(4, mean(rstudent(fit)[13:16]), col = 4, pch = 3, cex = 2)
```

```
points(5, mean(rstudent(fit)[17:20]), col = 5, pch = 3, cex = 2)
points(6, mean(rstudent(fit)[21:24]), col = 6, pch = 3, cex = 2)
```

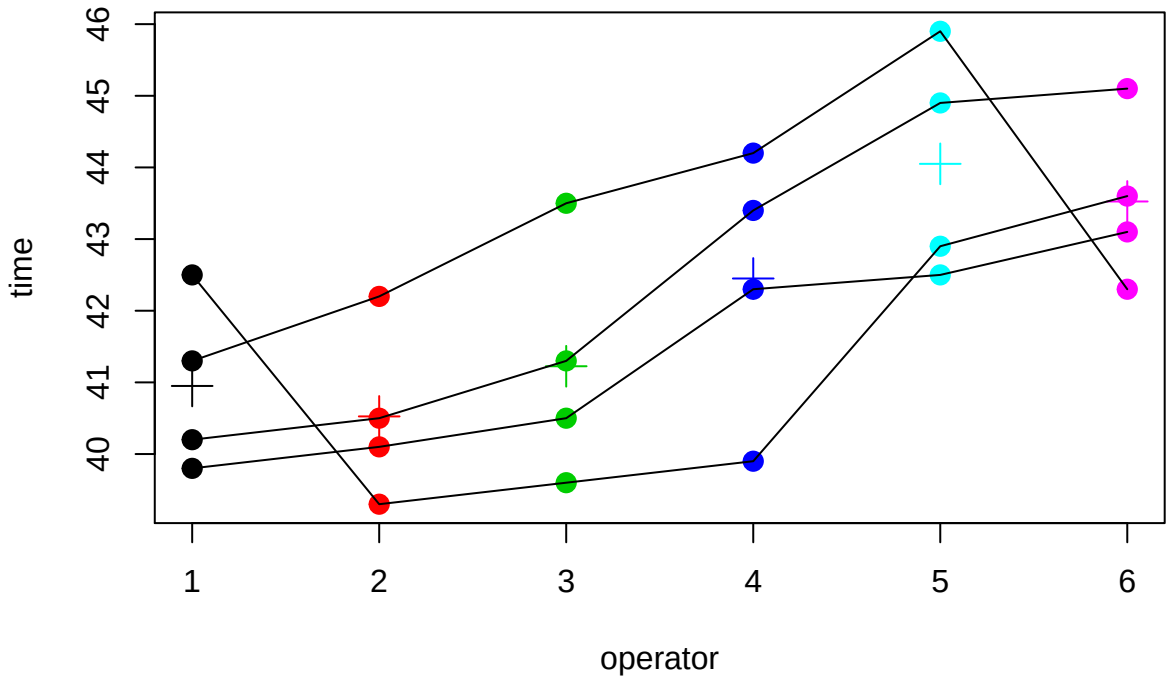



```
matplot(dsmat[, 1])
```




```
plot((dsmat[, 2]), dsmat[, 1], pch = 20, cex = 2, col = rep(1:6, each = 4),
    ylab = "time", xlab = "operator")
points(1, mean(dsmat[1:4, 1]), col = 1, pch = 3, cex = 2)
points(2, mean(dsmat[5:8, 1]), col = 2, pch = 3, cex = 2)
points(3, mean(dsmat[9:12, 1]), col = 3, pch = 3, cex = 2)
points(4, mean(dsmat[13:16, 1]), col = 4, pch = 3, cex = 2)
points(5, mean(dsmat[17:20, 1]), col = 5, pch = 3, cex = 2)
points(6, mean(dsmat[21:24, 1]), col = 6, pch = 3, cex = 2)
```

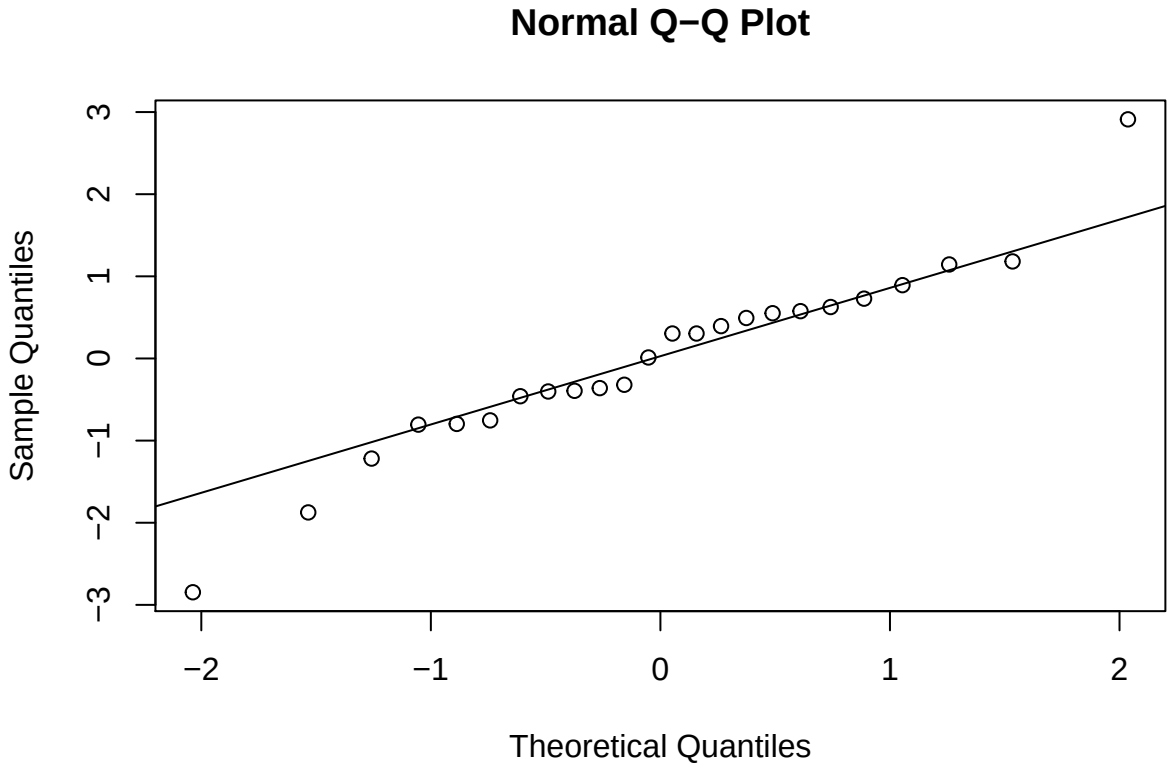
```
lines(1:6, dsmat[dsmat[, 3] == 1, 1], col = 1)
lines(1:6, dsmat[dsmat[, 3] == 2, 1], col = 1)
lines(1:6, dsmat[dsmat[, 3] == 3, 1], col = 1)
lines(1:6, dsmat[dsmat[, 3] == 4, 1], col = 1)
```




```
## Analysis of Variance Table
##
## Response: time
##                    Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(machine)  3 15.925  5.3082  1.6101 0.2186
## Residuals          20 65.935  3.2968
```

Alt går mye bedre når vi tar hensyn til det!

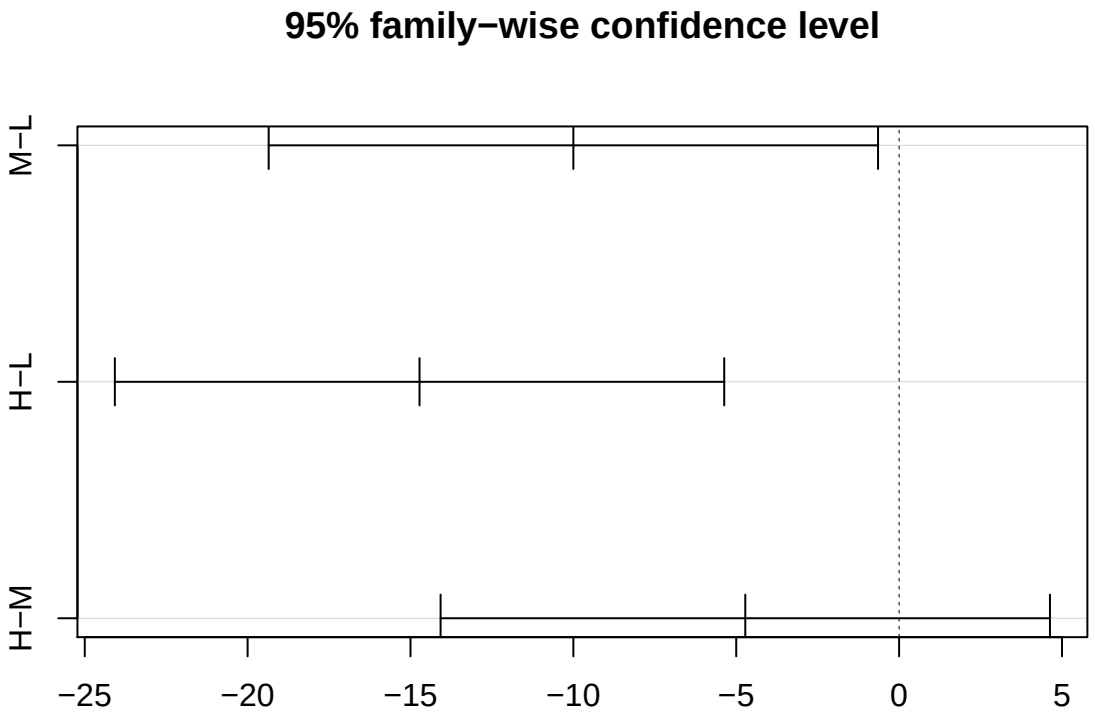
```
fit2 <- lm(time ~ as.factor(machine) + as.factor(operator), data = dsmat)
anova(fit2)
qqnorm(rstudent(fit2))
qqline(rstudent(fit2))
```




```
## Analysis of Variance Table
##
## Response: time
##                     Df Sum Sq Mean Sq F value   Pr(>F)
## as.factor(machine)   3 15.925  5.3082  3.3388 0.047904 *
## as.factor(operator)  5 42.087  8.4174  5.2944 0.005328 **
## Residuals           15 23.848  1.5899
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Tukey HSD krever at vi bruker `aov`-objekt. Eksemple fra `?TukeyHSD` med ull og spenning.

```
summary(fm1 <- aov(breaks ~ wool + tension, data = warpbreaks))
TukeyHSD(fm1, "tension", ordered = TRUE)
plot(TukeyHSD(fm1, "tension"))
```




```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## wool         1    451   450.7   3.339 0.07361 .
## tension      2   2034  1017.1   7.537 0.00138 **
## Residuals   50   6748   135.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##     factor levels have been ordered
##
## Fit: aov(formula = breaks ~ wool + tension, data = warpbreaks)
##
## $tension
##          diff        lwr      upr     p adj
## M-H  4.722222 -4.6311985 14.07564 0.4474210
## L-H 14.722222  5.3688015 24.07564 0.0011218
## L-M 10.000000  0.6465793 19.35342 0.0336262
```

13.3 Parret t-test

LDL-kolestrol målt før og etter en diett hos 13 kvinner, er det endret av dietten?

```
ldl = read.csv("http://www.math.ntnu.no/~mettela/TMA4255/Data/LDLbeforeafter.csv")
t.test(ldl$V1, ldl$V2, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  ldl$V1 and ldl$V2
```

```
## t = 1.557, df = 12, p-value = 0.1454
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.08687254  0.52191101
## sample estimates:
## mean of the differences
##               0.2175192
```

Kapittel 14: Ikke-parametriske metoder

14.2 Sign test

```
ds3 <- read.table("http://www.math.ntnu.no/~mettela/TMA4255/Data/data11_3.txt")
ds3
diff <- ds3[, 2] - ds3[, 1]
diff
# for the sign test differences of 0 is omitted
binom.test(sum(diff > 0), sum(diff != 0))
# this means binom.test(8,10) since 8 are positive differences and 10 are
# differences that are not 0.
```

```
##     V1  V2
## 1  6.4 6.6
## 2  5.8 5.8
## 3  7.4 7.8
## 4  5.5 5.7
## 5  6.3 6.0
## 6  7.8 8.4
## 7  8.6 8.8
## 8  8.2 8.4
## 9  7.0 7.3
## 10 4.9 5.8
## 11 5.9 5.8
## 12 6.5 6.5
##  [1]  0.2  0.0  0.4  0.2 -0.3  0.6  0.2  0.2  0.3  0.9 -0.1  0.0
##
##  Exact binomial test
##
## data:  sum(diff > 0) and sum(diff != 0)
## number of successes = 8, number of trials = 10, p-value = 0.1094
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.4439045 0.9747893
## sample estimates:
## probability of success
##                    0.8
```

14.3 Wilcoxon test

Ett utvalg

```r
# ett utvalg wilcoxon rank-sum test
wilcox.test(diff)
# or, equivalently
wilcox.test(ds3$V1, ds3$V2, paired = TRUE)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  diff
## V = 47.5, p-value = 0.04657
## alternative hypothesis: true location is not equal to 0
##
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  ds3$V1 and ds3$V2
## V = 7.5, p-value = 0.04657
## alternative hypothesis: true location shift is not equal to 0
```

To utvalg:

- Is it harder to maintain your balance while you are concentrating?
- Nine elderly and eight young people stood barefoot on a "force platform" and was asked to maintain a stable upright position and to react as quickly as possible to an unpredictable noise by pressing a hand held button.
- The noise came randomly and the subject concentrated on reacting as quickly as possible. The platform automatically measured how much each subject swayed in millimeters in both the forward

```r
# to utvalg
ds <- read.table("https://folk.ntnu.no/mettela/TMA4255/Data/balance.txt", header = TRUE)
ds
grp1 <- ds[(ds[, 4] == "elderly"), 3]
grp2 <- ds[(ds[, 4] == "young"), 3]
wilcox.test(grp1, grp2)
```

```
##    subject_No forward_backward side_side Age_Group
## 1           1               19        14   elderly
## 2           2               30        41   elderly
## 3           3               20        18   elderly
## 4           4               19        11   elderly
## 5           5               29        16   elderly
## 6           6               25        24   elderly
## 7           7               21        18   elderly
## 8           8               24        21   elderly
## 9           9               50        37   elderly
## 10          1               25        17     young
## 11          2               21        10     young
## 12          3               17        16     young
## 13          4               15        22     young
## 14          5               14        12     young
## 15          6               14        14     young
## 16          7               22        12     young
## 17          8               17        18     young
##
##  Wilcoxon rank sum test with continuity correction
##
```

```
## data:  grp1 and grp2
## W = 53, p-value = 0.1108
## alternative hypothesis: true location shift is not equal to 0
```

14.4 Kruskal-Wallis test

Ser tilbake på insektssprayeksemplet:

```
kruskal.test(count ~ spray, data = InsectSprays)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  count by spray
## Kruskal-Wallis chi-squared = 54.691, df = 5, p-value = 1.511e-10
```

Andre ressurser

- Intro til R: https://www.math.ntnu.no/emner/TMA4268/2018v/1Intro/Rbeginner.html
- Sannsynlighetsfordelinger (pdf,pmf og cdf): https://www.math.ntnu.no/emner/TMA4268/2018v/1Intro/Rintermediate.html
- Peter Dalgaard: An introduction to R.