BIO2150 Biostatistikk og studiedesign

Oppgaver i R

Det anbefales at studentene på egenhånd utfører oppgavene nedenfor for å få økt trening i å bruke R. Oppgavene er bakgrunnsmateriale for det som blir gjennomgått på noen av forelesningene til HAa, og danner basis for den kunnskapen som dere trenger for bl.a. å kunne skrive resultatene fra laboratorieøvelsene i Plantelab og Statpraksis.

Innholdsfortegnelse:

Oppgave 1 Variansanalyse og lineære modeller

Eksperiment:

Du arbeider i et firma som skal selge nye gjødselblandinger brukt i dyrking av korn. Det utføres et feltforsøk hvor effekten av tre gjødseltyper (FERTIL 1, 2 og 3) på avlingen (YIELD) undersøkes. Det er 10 forsøksfelt (eksperimentelle enheter eller replikater) for hver av henholdsvis gjødseltypene F1, F2 og F3. De 30 feltene høstes, og avlingen i tonn for hvert felt bestemmes, og gir vårt datasett eller datamateriale. Er det signifikante forskjeller mellom effekten av gjødseltypene FERTIL 1, 2 og 3 ? Hvilken gjødseltype vil du anbefale som gir størst avling ?
Vi skal undersøke i hvilken grad FERTIL kan forklare variasjonen i datasettet.
Datasettet er hentet fra: Grafen, A. & Hails, R.: *Modern statistics for the life sciences*. Oxford University Press 2003

Variansanalyse (ANOVA) og t-test

En **t-test** kan brukes hvis man skal sammenligne populasjonsgjennomsnittene for to grupper, men er lite egnet hvis man ønsker å sammenligne flere gjennomsnitt (kan gi **type II-feil**, et falskt negativt resultat hvor man beholder en gal nullhypotese). For normalfordelte uavhengige data med **homogen konstant varians** (**homoskedastisitet**) kan man bruke variansanalyse for å sammenligne tre eller flere grupper (behandlinger), hvor det alltid inngår en faktorvariabel. I **enveis ANOVA** er det bare en faktorvariabel (kategorisk variabel). Varians (*Var(X)*, s^2,) angir variasjonen i datasettet og inngår som en teoretisk parameter (σ^2)i sannsynlighetstetthetsfunksjonen for normalfordeling sammen med forventningen *E(X)*, (μ), det teoretiske gjennomsnittet for populasjonen (populasjonsgjennomsnittet). μ og σ^2 er parameterverdier vi ikke kan bestemme nøyaktig, men vi kommer med forslag til et estimat ved å bruke prøver som vi har tatt ut fra populasjonen. Det er derfor viktig at prøvene er representative, uavhengige og med mange nok **replikater**. Vi bruker greske bokstavene μ og σ^2 (sigma) når det er tale om den teoretiske populasjonen, og $\bar{x}$ og s^2 når det gjelder vårt datasett (prøven). Vårt datasett bruker vi til å lage **estimater** av de ukjente teoretiske

parameterverdiene for populasjonen. **Estimering** gir et forslag til parameterverdier ut fra datasettet vårt.
I ANOVA deles varians (variasjonen) i to deler: **Variasjon innen grupper** og **variasjon mellom grupper**.

ANOVA, modellseleksjon og ekstra kvadratsum

ANOVA blir også benyttet til **modellseleksjon, for å** sammenligne modeller og undersøke i hvilken grad en forklaringsvariabel skal være med i en modell eller ikke. Den ekstra kvadratsummen er kvadratsummen for forklart variasjon i den ene modellen minus dem tilsvarende kvadratsummen for den andre modellen. Til dette anvendes **AIC** (**Akaikes informasjonskriterium**) som gir en større og økt straff desto flere modellparametre man putter inn i modellen. Man velger modellen som har lavest AIC-verdi.

Storgjennomsnitt (stormiddeltall) og gruppegjenomsnitt (gruppemiddeltall)

Vi kan beregne et stort gjennomsnitt av alle dataene vi har, **storgjennomsnitt** (stormiddeltallet, ”grand mean”), og vi kan beregne gjennomsnitt av dataene for alle gruppene, **gruppegjennomsnitt (gruppemiddeltallet)**. Spørsmålet er hvor mye av variasjonen er tilfeldig, og hvor mye av variasjonen kan forklares av de forskjellige faktorene (behandlingene, gruppene, ”treatment”). En av variasjonene er den tilfeldige variasjonen innen faktornivået (”within”,”treatment”), som er variasjon av hver måling minus gruppegjennomsnittet.
Den andre variasjonen er mellom storgjennomsnittet og gruppegjennomsnittet (”between”,”random”). Summen av kvadrerte avvik er summen av kvadratet av avstanden mellom hvert datapunkt og gjennomsnittet for alle datapunktene.
ANOVA forteller ikke hvilke grupper som er forskjellig fra hvem.

En generell lineær modell for eksperimentet vårt er

```
YIELD ~ FERTIL + error
```

responsvariabel (y) ~ forklaringsvariabel(x) + error (feil)

~ betyr modellert av. Verdien for y blir forklart eller modellert av x, (y ~ x)
Modellen blir brukt til å **prediktere** (forutsi) verdien av responsvariabel (avhengig variabel) gitt verdien av forklaringsvariabel (uavhengig variabel, prediktor).
En **generell lineære modell** blir anvendt innen variansanalyse og lineær regresjon, men vi skal seinere se på **generaliserte lineære modeller** hvor vi kan ta hensyn til om responsvariabelen følger **normalfordeling**, **binomial fordeling** (myntkastfordeling) eller **poisson-fordeling** ved å anvende **link-funksjoner**.
Du kan ha datafilen i txt-format lagret i en mappe på din datamaskin som du definerer med File < Change dir… i R-vindu. Sjekk hvilke filer du har liggende ved **dir()** og les inn filen ved **read.table()**. Hvis tallene i txt-filen inneholder komma (,) må dette erstattes med punktum (.), og dette kan gjøres i R ved å angi ,…header = TRUE, dec = ","). Eller alternativt

```
#Last inn datafil oppg1.txt fra arbeidsdirektoriet:
oppg1 <- read.table("oppg1.txt", header = TRUE)
attach(oppg1) #husk detach(oppg1) etter bruk
```

```
names(oppg1) #angir navnene til et objekt
[1] "FERTIL" "YIELD"
```
Bruk av **attach ()** gjør at R finner fram til variabelnavn direkte uten å angi dem med dollartegn ($) for eksempel **oppg1$FERTIL**, eller at man alltid må angi for eksempel …,data = oppg1). Bruk av attach() har en ulempe hvis man opererer med flere datasett med samme variabelnavn. Uansett er det lurt å bruke **detach()** når man er ferdig med et datasett og har brukt attach().
Først må man sjekke datasettet. Oppsummering datasettet angir minimums- og maksimumsverdi, gjennomsnitt og median (midtverdien), 1. og 3. kvartil.
```
summary(oppg1)
 FERTIL      YIELD
 F1:10   Min.   :3.070
 F2:10   1st Qu.:3.828
 F3:10   Median :4.540
         Mean   :4.644
         3rd Qu.:5.048
         Max.   :7.140
```
Det er 10 **replikater** for hver behandling F1, F2 og F3
Man kan se på strukturen til datasettet med
str(oppg1) .
Vi kan se hvilke klasser et objekt har:
```
sapply(oppg1, class)
   FERTIL     YIELD
 "factor" "numeric"
```
FERTIL er en faktorvariabel eller kategorisk variabel (prediktor, forklaringsvariabel, uavhengig diskret variabel) med tre nivåer med i alt 30 eksperimentelle enheter (forsøksenheter). **YIELD** er en kontinuerlig responsvariabel (avhengig variabel).
```
dim(oppg1)
[1] 30  2
```
De 11 første dataene i datasettet, men generelt gir **head(oppg1)** de 6 første verdiene i objektet, og **tail(oppg1)** de 6 siste
```
oppg1[1:11,]
   FERTIL YIELD
1      F1  6.27
2      F1  5.36
3      F1  6.39
4      F1  4.85
5      F1  5.99
6      F1  7.14
7      F1  5.08
8      F1  4.07
9      F1  4.35
10     F1  4.95
11     F2  3.07
```
Generelt kan vi se hva som står i en kolonne ved **oppg1$YIELD**
oppg1[["YIELD"]] eller **oppg1[, "YIELD"]** som gir samme resultat. Dollartegn ($) brukes for å angi et element i et objekt, og klammeparentes [] brukes for å angi hvilke elementer man har i en vektor eller matrise.

Den sentral tendens

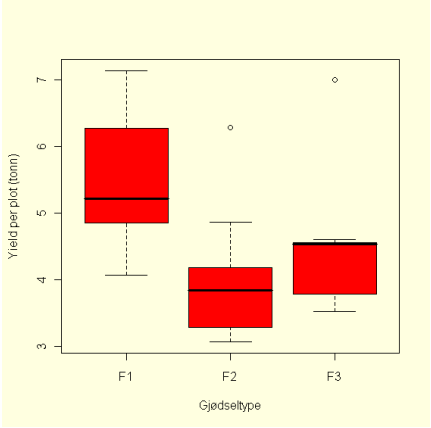
Den sentrale tendens kan angis som **gjennomsnitt** (middelverdi, ”mean”), **median** (midtverdi) eller **mode** (den det er flest av). For å kunne beregne median må data sorteres fra den minste til den største. Gjennomsnittet for vårt datasett angis som $\bar{x}$,

men gjennomsnittet for den teoretiske populasjonen angis med den greske bokstaven my (μ).
Spredningsmål er en numerisk oppsummering av populasjonen: varians, standardavvik, standardfeil, maksimum-minimum, og interkvartilområde. Varians (kvadratsum) har en prøvefordeling som følger **kjikvadratfordeling** (”chi-square distribution”)(χ^2).

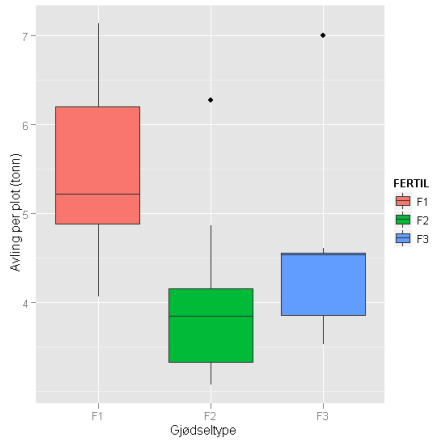
F-fordeling

Med fordeling mener vi en sannsynlighetstetthetsfordeling. Forholdet mellom to varianser følger **F-fordeling**, og benyttes i variansanalyse. F-fordelingen har to frihetsgrader (df, ”degrees of freedom”), en for hver av de to variansene, middelkvadratet (”Mean square”, MS_G) for gruppene dividert på middeltallet for error (MS_E), en for teller og en for nevner. F-fordelingen er høyreskjev, har en lang hale mot høyre, og skjevheten (”skew”) avhenger av de to verdiene for antall frihetsgrader (df). Vi sammenligner den teoretiske F-verdien (kritisk tabellverdi for F-ratio) med vår F-observator som vi har regnet ut fra datasettet, og undersøker hvor sannsynlig er det å få vår F-verdi sammenlignet med den teoretiske. Nullhypotesen (H_0)er at det ikke er noen forskjell mellom kritisk tabellverdi og vår F-observator regnet ut fra tallene i datasettet.

```
#Et boksplot av data
par(bg = "lightyellow")
plot(FERTIL, YIELD, col = 2, xlab = "Gjødseltype",
     ylab = "Yield per plot (tonn)", data = oppg1)
```




```
#alternative figurer via ggplot2
library(ggplot2) #må først lastes ned via CRAN
qplot(FERTIL,YIELD ,data=oppg1,geom=c("boxplot"),fill=FERTIL,
     xlab="Gjødseltype", ylab="Avling per plot (tonn)")
```

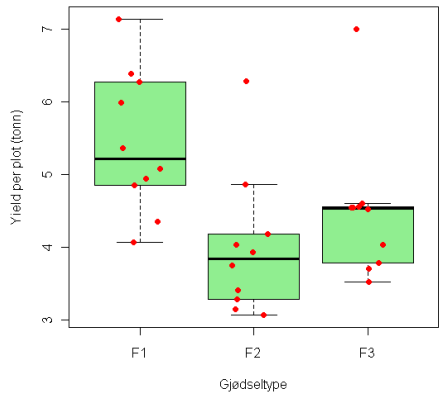


Boksplot

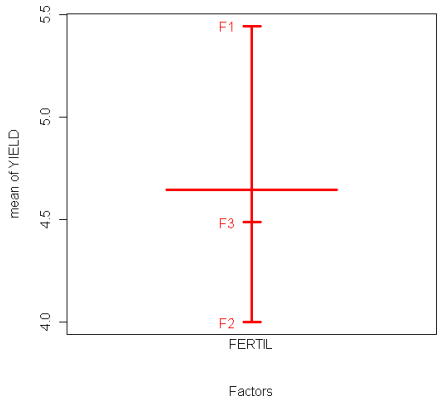
Boksplot viser **median** (svart strek midt i boksen = 50% persentil, midtverdien). Boksen omfatter 1. og 3. kvartil, **interkvartilområde** = 25% - 75% **persentil**. Værhår (vertikal prikket strek) omfatter måleverdier som ligger 1.5 ganger interkvartilavstanden. **Utliggere** er ekstreme observasjoner som ligger utenfor 1.5 ganger interkvartilavstanden. For en normalfordeling vil medianen (midtverdien) ligge omtrent midt i boksen, og værhårene går omtrent like langt ut på begge sider av boksen. Kommandoen **notch = TRUE** (hakk, innskjæring, Notchplot) viser et hakk eller innsnitt for 95% konfidensintervallet rundt median, og hvis de to innsnittene ikke overlapper så er medianverdiene signifikant forskjellig. Hvis innhakkene ikke overlapper hverandre er medianene signifikant forskjellig på 5% nivå.

Her er en annen måte hvor datapunktene vises samtidig:

```
boxplot(YIELD ~ FERTIL,col = "lightgreen",
     xlab =  "Gjødseltype",ylab = "Yield per plot (tonn)",
     outline =  FALSE, data = oppg1)
#bruker jitter for å spre punktene
points(jitter(as.numeric(oppg1$FERTIL)), oppg1$YIELD,
     col =  2, pch =  16)
```




```
#design av eksperimentet
plot.design (oppg1, col = 2, lwd = 3)
```

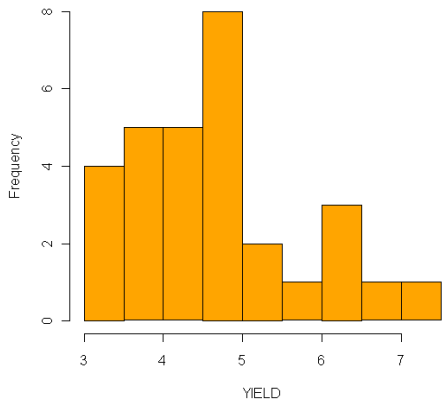


Figuren viser stormiddeltallet og middeltallene (gjennomsnittene) for de tre forskjellige behandlingene med FERTIL, dvs. F1, F2 og F3.

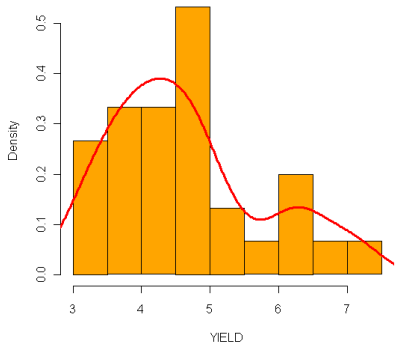
```
#Sjekking av data, finn antall måledata
n <- length(YIELD); n
 [1] 30
#Finn antall nivåer av faktoren FERTIL
levels(FERTIL)
[1] "F1" "F2" "F3"
```

I vårt tilfelle hvor faktorene er angitt med bokstavkombinasjoner er det greit, men i noen tilfeller er faktorene kodet som tall. Da må man gi beskjed til R om at variabelen er en faktor ved å angi variabelnavnet på nytt som en faktor for eksempel

```
FERTIL <- factor(FERTIL)
#Er FERTIL en faktor ?
is.factor(FERTIL)
[1] TRUE
#Finner forventet verdi (stormiddeltallet (M)) til YIELD
mean(YIELD)
[1] 4.643667
#Finn varians til YIELD
var(YIELD)
[1] 1.256721
#Gi et sammendrag av dataene, deskriptiv statistikk
summary(oppg1)
 FERTIL      YIELD
 F1:10   Min.   :3.070
 F2:10   1st Qu.:3.828
 F3:10   Median :4.540
         Mean   :4.644
         3rd Qu.:5.048
         Max.   :7.140
#Spredningen på dataene største og minste
range(YIELD)
[1] 3.07 7.14
#Lag et histogram av YIELD
hist(YIELD, col = "orange", main="")
```




```
#histogram med sannsynlighetstetthet
hist(YIELD, prob =  TRUE, col = "orange", main="")
lines(density(YIELD), col = 2, lwd = 3)
```

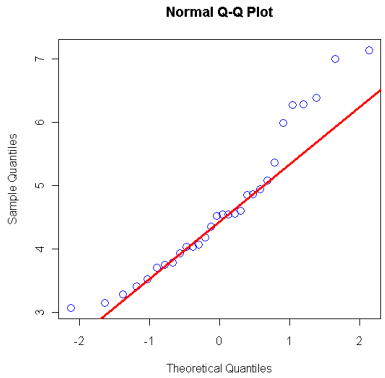


Allerede nå kan vi se avvik skjevfordelte data

Histogram og frekvensfordeling

Frekvensfordeling får man ved å telle og gruppere antall objekter/subjekter med en bestemt verdi eller **verdiintervall**. Høyden på kolonnene viser andel og antall av dataene som befinner seg innen hvert intervall. **Sannsynlighetsfordeling** er andelen er proporsjonen av antall objekter med en bestemt verdi, og kan uttrykkes i et **histogram**. En **symmetrisk fordeling** (eks. normalfordeling, t-fordeling) kan brettes omkring en midtlinje, og de to halvdelene er speilbilder av hverandre. **Mode** er den vanligste respons, den det er flest av, og en fordeling med bare en topp kalles **unimodal**. Har den to topper er den **bimodal**. En **venstreskjev fordeling** har lang hale til venstre. En **høyreskjev fordeling** har lang hale til høyre.

```
#Sjekk normalfordeling av YIELD med qqnorm() og qqline()
qqnorm(YIELD, col = 4, cex = 1.5)
qqline(YIELD, col =  2, lwd = 3)
```



Vi ser at datapunktene ligger på en bueform, og det gjenspeiles også i histogrammet som skjevfordeling (”skew”).

En-veis variansanalyse (ANOVA)

En-veis ANOVA har en faktorvariabel, to-veis ANOVA har to faktorvariable.

```
#Anovamodell variansanalyse
summary(aov(YIELD ~ FERTIL))
```

```
            Df  Sum Sq Mean Sq F value   Pr(>F)
FERTIL       2 10.8227  5.4114  5.7024 0.008594 **
Residuals   27 25.6221  0.9490
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

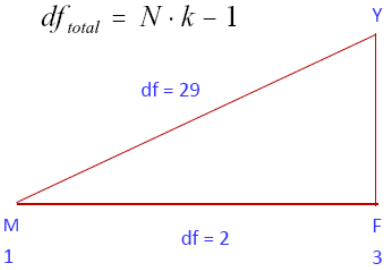
Vi har for kvadratsummene at **totalkvadratsummen** SS_T er lik **gruppekvadratsummen** (kvadratsummen for gruppene, SS_G, summen av kvadrerte avvik mellom stormiddeltallet og gruppemiddeltallene) pluss **feilkvadratsummen** (kvadratsummen for residualene, error eller feil, SS_E, summen av kvadrerte avvik mellom gruppemiddeltallene og alle måleverdiene i datasettet.
SST (SS_T)= SSF (SS_G) + SSE (SS_E)
36.4449 = 10.8227 + 25.6221
Vi kommer tilbake til dette litt lenger ned i teksten.

Middelkvadratsummen (”Mean Sq”) er kvadratsummen (SS, ”Sum Sq”) dividert på antall frihetsgrader (Df, df).

F-verdien er forholdet mellom de to middelkvadratene:
F = FMS/EMS = 5.4114/0.9490 = 5.7024
Testobservator blir regnet ut fra vårt datasett
Er denne F-verdien signifikant ? Vi sammenligner med den kritiske tabellverdien av **F** for p = 0.05 dvs. 0.95, og henholdsvis 2 og 27 frihetsgrader. Nullhypotesen (H_0) om at det ikke er noen forskjell forkastes, og den alternative hypotesen H_A beholdes.
Konklusjon: det er signifikante forskjeller i effekten av gjødseltype, gjødseltype forklarer variasjon i biomasse (YIELD). Figuren nedenfor viser hvorfor man får de forskjellige frihetsgradene (df), hvor M er stormiddeltallet.

N – antall måledata i hver gruppe (10)
k- antall grupper (3)

$$df_{total} = N \cdot k - 1$$



$$df_{feil} = N \cdot k - k$$

$$df_{grupper} = k - 1$$

Effektstørrelser og konfidensintervall

Mange er kritiske til bruk av nullhypoteser og bruk av p < 0.05-testing innen **statistisk inferens**. Bestemmelse av **konfidensintervall** og effektstørrelser er viktigere. Signifikansnivået, α = 0.05, (konfidens 0.95) som grenseverdi er subjekt valgt, fordi Fisher mente den var en passelig verdi. ANOVA og regresjon blir erstattet av **generaliserte lineære modeller** (GLM). Ved gjentatte sammenligninger må man endre verdien for p ved for eksempel Bonferroni-korreksjon med **p.adjust**.

```
anova(lm(YIELD ~ FERTIL))
#kritisk tabellverdi for F
qf(0.95, 2, 27)
[1] 3.354131
#Sannsynligheten for å få F-verdien vi har funnet
1-pf(5.7024, 2, 27)
[1] 0.00859422
```

Sannsynligheten for å få en slik verdi vi fikk hvis forventede verdier (middeltallene, gjennomsnittene) var like er ca. 0.8%, konklusjon det er signifikant forskjell mellom gjødseltypene.

Lineær regresjonsmodell

En lineær modell er av type:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

hvor β_0 er **intercept** (skjæringspunkt) og β_1 er stigningskoeffisient (stigningstallet) i en regresjonsbasert tilnærming. Vi ønsker å finne konfidensintervall for parameterverdiene β_0 og β_1. **Konfidensintervall** (95% konfidensintervall) til parameterestimatene er en god måte å bestemme størrelsen på en effekt. Konfidensintervall er nyttige selv om nullhypotesen ikke blir forkastet. Vi antar at $\varepsilon \sim N(0,\sigma^2)$ dvs. feilene (residualene) er uavhengige, har lik varians og er normalfordelte. Hvis derimot det skulle vise seg variansen ikke er konstant (variabel varians, **heteroskedasitisitet**) så må man enten **transformere** dataene eller bruke en ikke-parametrisk testmetode. Modellen er bare en approksimasjon (tilnærming) av underliggende realiteter. Vi bruker en-veis ANOVA, og fordeler totalvarians i respons på faktorvarians og error-varians.

Generelt betyr lm(YIELD ~ FERTIL), tilpass en lineær modell av YIELD som funksjon av FERTIL, hvor FERTIL de tre gjødselbehandlingene som forklarer mengde avling (YIELD).

Nullmodell

Den enkleste modellen er **nullmodellen**, y ~ 1, hvor det ikke er med noen forklaringsvariabel. Nullmodellen angir storgjennomsnittet for alle dataene, intercept i modellen Den sier noe om variasjonen i biomasse, men helt uavhengig av hvilken gjødseltype.

```
#Nullmodell
mod0 <- lm(YIELD ~ 1)
summary(mod0)
Call:
lm(formula = YIELD ~ 1)
Residuals:
    Min      1Q  Median      3Q     Max
-1.5737 -0.8162 -0.1037  0.4038  2.4963

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.6437     0.2047   22.69   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 1.121 on 29 degrees of freedom
```

Vi ser at nullmodellen gir intercept (β_0) som er lik forventet verdi for alle YIELD, altså **stormiddeltallet** (”grand mean”). t-verdien er lik estimatet dividert på standardfeilen. Antall **frihetsgrader** (df) er lik n – 1, hvor n = 30 er antall datapunkter i datasettet Vi trekker fra 1 fordi vi allerede har brukt datasettet vårt en gang tidligere da vi regnet ut storgjennomsnittet, og som vi benytter i utregning av kvadratsummen (SS).

En ANOVA av nullmodellen:

```
anova(mod0)
Analysis of Variance Table

Response: YIELD
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 29 36.445   1.257
```

Vi ser at vi får bare en kvadratsum, og det er totalkvadratsummen SS_T (også kalt SSY nedenfor), Nullhypotesen er (H_0): Gjødslingen ikke har noen effekt på avlingen. Hvis vi starter med nullmodellen og nå legger til forklaringsvariabelen vår, FERTIL har vi en **forlengsseleksjon** i modellseleksjonen. I vårt eksperiment har vi imidlertid bare en forklaringsvariabel, men i mer kompliserte eksperimenter har vi flere forklaringsvariable som etter hvert kan puttes inn i modellen og man kan studere hvilken effekt de har på estimatene av parameterverdier.

```
#Stor modell
mod1 <- lm(YIELD ~ FERTIL)
summary(mod1)
Call:
aov(formula = YIELD ~ FERTIL)
Residuals:
    Min      1Q  Median      3Q     Max
```

```
-1.3750 -0.6715 -0.0720  0.1740  2.5130
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.4450     0.3081  17.676 2.28e-16 ***
FERTILF2     -1.4460     0.4357  -3.319  0.00259 **
FERTILF3     -0.9580     0.4357  -2.199  0.03663 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9742 on 27 degrees of freedom
Multiple R-Squared: 0.297,      Adjusted R-squared: 0.2449
F-statistic: 5.702 on 2 and 27 DF,  p-value: 0.008594
```

```
coef(mod1) #koeffisientverdiene i modellen
(Intercept)    FERTILF2    FERTILF3
      5.445      -1.446      -0.958
```

Vi ser at intercept (β_0) er lik forventet verdi (middeltallet, gjennomsnitt) for F1 og at estimatene for F2 og F3 er lik avvikene fra intercept. FERTILF1 blir brukt som referanse og settes lik 0. Det faktornivået som kommer først i alfabetet blir satt lik intercept, altså her F1. Fra p-verdi og F-observator (”F-statistic”) konkluderer vi med at det er en forskjell mellom gruppene FERTIL 1,2 og 3.
Ved å dividere estimatet på standardfeilen får man **t-verdien**. Vi kan se bort fra fortegnet på t-verdien siden det er de absolutte forskjellene vi ser på.
t-observator (den kritiske verdien for t):

```
#kritisk verdi for t
qt(0.975, 27)
[1] 2.051831
```

Siden vår tabellverdi er større enn den kritiske verdien for t forkastes nullhypotesen om at det ikke er noen forskjeller mellom F1, F2 og F3. Det er imidlertid bedre å se på sannsynligheten, vi bruker kommandoen **qt** og ganger med 2 siden vi bruker en **to-halet test**, en **tosidig test** benytter begge halene på den statistiske fordelingen

```
#Sannsynligheten for å få t-verdier iflg. tabell
2*pt(-3.319, 27)
[1] 0.002592942
2*pt(-2.199, 27)
[1] 0.03662561
```

Dette er sannsynlighetene du finner i tabellen.

R^2

R^2 , andelen av total varianse i y som er forklart av x, viser hvor mye av variasjonen som forklares av FERTIL, dvs. SSF/SST, dvs. 29.7% av variasjonen

```
10.8227/36.445
[1] 0.2969598
```

R^2 kan ikke brukes til modellseleksjon fordi modeller med flere forklaringsvariable har alltid høyere R^2.

Lineær modell

FERTIL

$$YIELD \sim 5.445 + \begin{matrix} 0.0000 & 1 \\ -1.446 & 2 \\ -0.9580 & 3 \end{matrix} + error$$

Regner vi ut får vi:

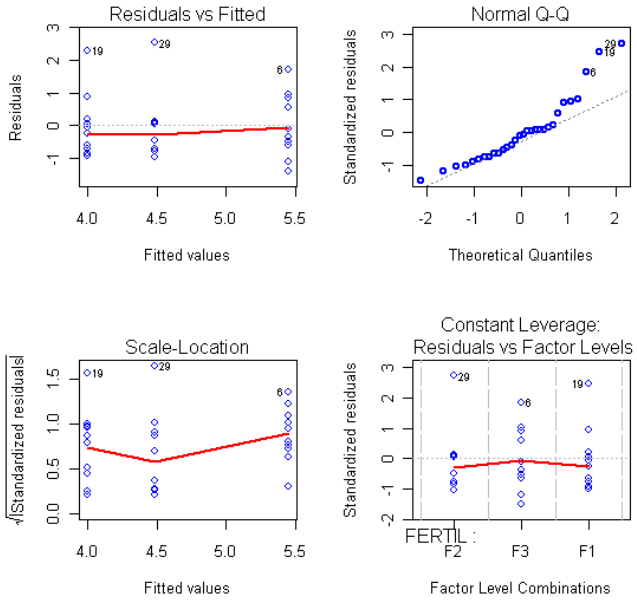
FERTIL

$$YIELD \sim \begin{matrix} 5.4450 \\ 3.9990 \\ 4.4870 \end{matrix} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} + error$$

Diagnostiske plot

Undersøker modellantakelsene

```
#Plotter modellen
par(mfrow = c(2,2))
plot(mod1, col = 4, lwd = 2)
```

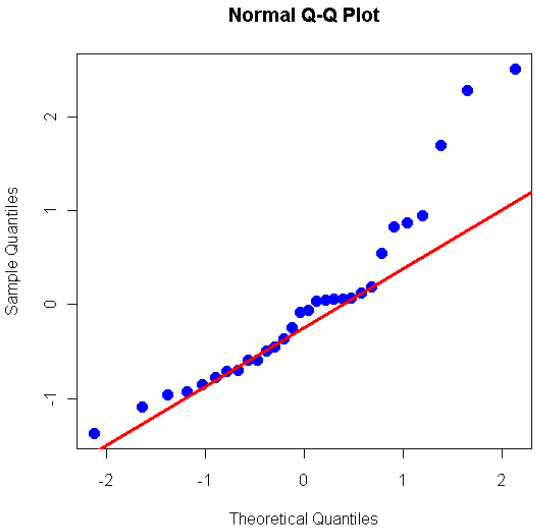


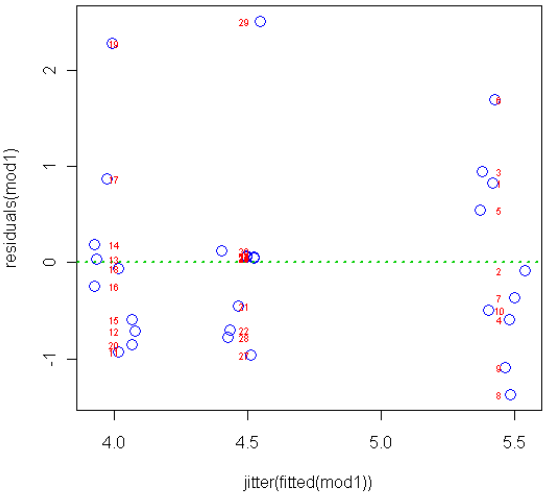
Diagnostikk av ANOVA modellen

Linære modeller forutsetter residualene (feilene) er normalfordelte, uavhengige og har samme (konstant) varians. Vi må sjekke for ikke-konstant varians (**heteroskedastitet**).

Må plotte residualer og tilpassete verdier og lage et kvantil-kvantil-plot (QQ-plot) av residualene, og oppfylles forutsetningene vil punktene bli liggende på omtrent en rett linje. Når data prediktert av modellen plottes mot residualene skal dette gi punkter som ligger tilfeldig uten noe mønster på hver side av den horisontale nullinjen.

```
qqnorm(residuals(mod1), col=4, cex=1.5, pch=16)
qqline(residuals(mod1), col=2, lwd=3)
plot(jitter(fitted(mod1)), residuals(mod1), col=4,cex = 1.5)
abline(h = 0, lty = 3, col = 3, lwd = 2)
text(fitted(mod1),residuals(mod1), col = 2,
     labels = rownames(oppg1),cex = 0.6)  #nummer på punktene
```





Vi ser at residualene ikke er helt normalfordelte,det blir en ”bananform” noe som gjenspeiles i den tidligere kvantil-kvantil-testen.

Designmatrisen viser kodingen:

```
model.matrix(mod1)
```

```
   (Intercept) FERTILF2 FERTILF3
1            1        0        0
2            1        0        0
3            1        0        0
4            1        0        0
5            1        0        0
6            1        0        0
7            1        0        0
8            1        0        0
9            1        0        0
10           1        0        0
11           1        1        0
12           1        1        0
13           1        1        0
14           1        1        0
15           1        1        0
16           1        1        0
17           1        1        0
18           1        1        0
19           1        1        0
20           1        1        0
21           1        0        1
22           1        0        1
23           1        0        1
24           1        0        1
25           1        0        1
26           1        0        1
27           1        0        1
28           1        0        1
29           1        0        1
30           1        0        1
attr(,"assign")
[1] 0 1 1
attr(,"contrasts")
attr(,"contrasts")$FERTIL
[1] "contr.treatment"
```

Vi kan lage en modell uten intercept og ser da at vi får forventede verdier direkte (gjennomsnittsverdier, middelverdier). R^2 blir ikke riktig siden vi har utelatt intercept. F-test tilsvarer nullhypotesen om at forventet gjennomsnittsrespons er lik 0. Dette er uinteressant.

```
mod2 <- lm(YIELD ~ FERTIL-1)
```

```
summary(mod2)
Call:
lm(formula = YIELD ~ FERTIL - 1)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3750 -0.6715 -0.0720  0.1740  2.5130

Coefficients:
         Estimate Std. Error t value Pr(>|t|)
FERTILF1   5.4450     0.3081   17.68 2.28e-16 ***
FERTILF2   3.9990     0.3081   12.98 4.02e-13 ***
FERTILF3   4.4870     0.3081   14.57 2.62e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9742 on 27 degrees of freedom
Multiple R-squared: 0.9625,     Adjusted R-squared: 0.9583
F-statistic:   231 on 3 and 27 DF,  p-value: < 2.2e-16
```

Vi kan se hvilke navn på objekter vi får i modellen:

```
names(mod1)
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "contrasts"     "xlevels"       "call"          "terms"
[13] "model"
```

Test for normalitet: Shapiro-Wilk test

En annen mulighet til å teste for normalitet er **Shapiro-Wilk test()**. Nullhypotesen er at datene følger normalfordeling, og hvis p <0.05 forkastes nullhypotesen

```
shapiro.test(YIELD)
        Shapiro-Wilk normality test
data:  YIELD
W  =   0.9272, p-value  =   0.04148
```

Dette gjenspeiler det vi finner i QQ-plottet, om at avvikene fra normalfordelingen er i grenseland for det vi kan tillate. Hvis p <0.05 er ikke dataene normalfordelte, og p må være > 0.05 for at vi skal konkludere med normalfordeling.

Konstant varians og Bartlett-test

Konstant (homogen) varians er en forutsetning for anova og regresjon og dette kan testes i en **bartlett.test().** Nullhypotesen er data med konstant varians, og er p <0.05 indikerer dette at dataene ikke har konstant varians.
Alternativ er **fligner.test()**. Hvis vi bare har to prøver hvor varians skal sammenlignes bruker vi en Fishers F-test (**var.test()**).

```
bartlett.test(YIELD ~ FERTIL)
        Bartlett test of homogeneity of variances
data:  YIELD by FERTIL
Bartlett's K-squared  = 2e-04, df  = 2, p-value = 1
```

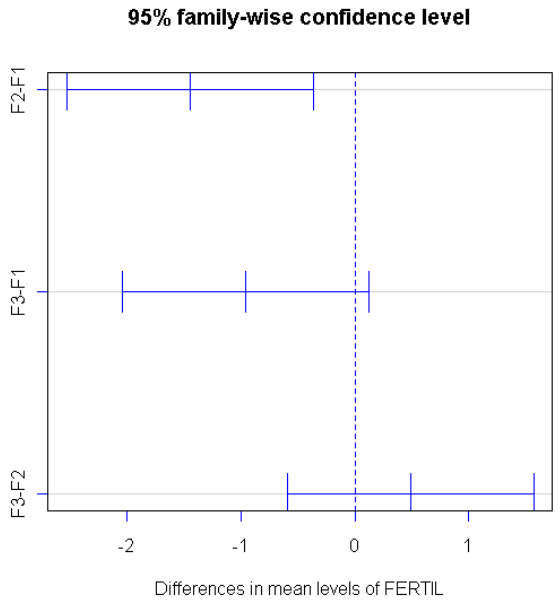
Vi kan konkludere med at varians er homogen, hvis p < 0.05 hadde vi ikke hatt homogen og konstant varians. Vi får høy p-verdi, og konkluderer at det er ingen evidens for ikke-konstant varianse

Post-hoc tester – Tukey-Kramer test

Variansanalyse tester om det er signifikante forskjeller mellom prøver og gruppemiddeltall, men angir ikke hvem som er forskjellig fra hvem. Det finnes imidlertid en rekke post-hoc tester som gjør dette bl.a. Tukey's Honestly Significant Difference (**TukeyHSD()**), også kalt **Tukey-Kramer test**.

```
TukeyHSD(aov(YIELD ~ FERTIL))
  Tukey multiple comparisons of means
    95% family-wise confidence level
Fit: aov(formula = YIELD ~ FERTIL)
$FERTIL
        diff        lwr        upr     p adj
F2-F1 -1.446 -2.5261662 -0.3658338 0.0070788
F3-F1 -0.958 -2.0381662  0.1221662 0.0894812
F3-F2  0.488 -0.5921662  1.5681662 0.5102335
plot(TukeyHSD(aov(YIELD ~ FERTIL)))
```

Hvis horisontale streker krysser striplet linje er det ingen signifikant forskjell. Det er signifikant forskjell mellom FERTIL 1 og 2.



function() og tapply()

Det går også an å lage seg en funksjon av x funksjon(x) som vi kaller varians og deretter setter vi inn YIELD i stedet for x i funksjonen og vi ser at vi får samme verdi for varians som ovenfor

```
varians <- function(x) {sum((x-mean(x))^2/(length (x)-1))}
varians(YIELD)
[1] 1.256721
```

Kommandoen **tapply()** er nyttig for å finne forventet verdi (middeltall, gjennomsnitt) eller varianse for datasettet:

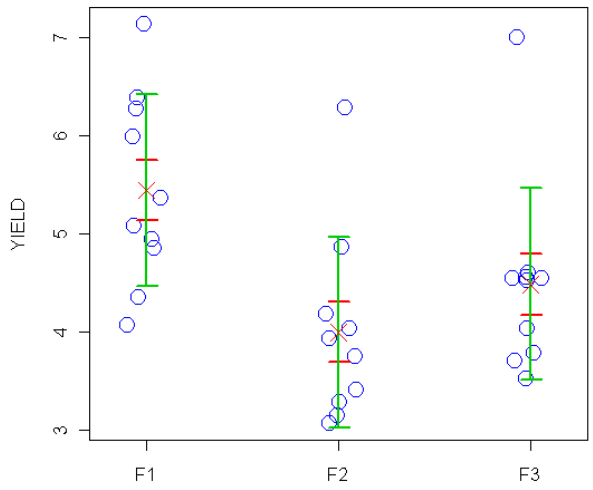
```
tapply(YIELD,FERTIL, mean)
   F1    F2    F3
5.445 3.999 4.487
tapply(YIELD, FERTIL, var)
       F1        F2        F3
0.9525389 0.9442989 0.9500678
```

Vi kan sjekke at vi har like mange tall for hver FERTIL:

```
tapply(YIELD, FERTIL, length)
F1 F2 F3
10 10 10
```

Vi kan plotte både standardavvik (”standard deviation”, grønn), kvadratroten av varians, og standardfeil (”standard error”, rød).
Hentet fra Dahlgaard, P.: *Introductory statistics with R*. 2e. Springer 2008 s. 135

```
stripchart(YIELD ~ FERTIL,method = "jitter",vert = TRUE,
     pch = 1, cex = 2, col = 4)
m <- tapply(YIELD, FERTIL, mean) #gjennomsnittsverdi
s <- tapply(YIELD, FERTIL, sd) #standardavvik
n <- tapply(YIELD, FERTIL, length)
se <- s/sqrt(n) #standardfeil
arrows(1:3, m + se, 1:3, m - se, angle = 90, code = 3,
     length = 0.1, lwd = 2, col = 2)
arrows(1:3, m+s, 1:3, m-s, angle =  90, code =  3,
     length = 0.1, lwd = 2,col = 3)
points(m, pch = 4, cex = 2, col = 2)
```



Her vises en annen måte å finne middeltallene for F1, F2 og F3. To likhetstegn == etter hverandre tilsvarer logisk er lik

```
mean(YIELD[FERTIL == "F1"])
[1] 5.445
mean(YIELD[FERTIL == "F2"])
[1] 3.999
mean(YIELD[FERTIL == "F3"])
[1] 4.487
```

Vi kan for eksempel se på estimatene til de to første tallene i koeffisientmodellen:

```
mean(YIELD[FERTIL == "F2"])-mean(YIELD[FERTIL == "F1"])
[1] -1.446
```

Tilsvarende for det siste estimatet:

```
mean(YIELD[FERTIL == "F3"])-mean(YIELD[FERTIL == "F1"])
```

```
[1] -0.958
```

Tilsvarende kan man beregne varians:

```
varF1 <- var(YIELD[FERTIL == "F1"])
varF1
[1] 0.9525389
varF2 <- var(YIELD[FERTIL == "F2"])
varF2
[1] 0.9442989
varF3 <- var(YIELD[FERTIL == "F3"])
varF3
[1] 0.9500678
```

Hvordan finner man tallene i ANOVA-tabellen ?
Finn avvik fra stormiddeltallet (MY):

```
YIELD - mean(YIELD)
 [1]  1.62633333  0.71633333  1.74633333  0.20633333  1.34633333  2.49633333
 [7]  0.43633333 -0.57366667 -0.29366667  0.30633333 -1.57366667 -1.35366667
[13] -0.60366667 -0.45366667 -1.23366667 -0.89366667  0.22633333 -0.70366667
[19]  1.63633333 -1.49366667 -0.60366667 -0.85366667 -0.08366667 -0.09366667
[25] -0.09366667 -0.11366667 -1.11366667 -0.93366667  2.35633333 -0.03366667
```

Kvadrer avvikene fra stormiddeltallet:

```
(YIELD - mean(YIELD))^2
 [1] 2.644960111 0.513133444 3.049680111 0.042573444 1.812613444 6.231680111
 [7] 0.190386778 0.329093444 0.086240111 0.093840111 2.476426778 1.832413444
[13] 0.364413444 0.205813444 1.521933444 0.798640111 0.051226778 0.495146778
[19] 2.677586778 2.231040111 0.364413444 0.728746778 0.007000111 0.008773444
[25] 0.008773444 0.012920111 1.240253444 0.871733444 5.552306778 0.001133444
```

Summer de kvadrerte avvikene fra stormiddeltallet og finn SSY:

```
SSY <- sum((YIELD - mean(YIELD))^2)
SSY
[1] 36.4449
```

Finn avvikene (MF) fra stormiddeltallet til de enkelte middeltallene for de 3 gjødseltypene:

```
MF <- tapply(YIELD, FERTIL,mean) - mean(YIELD)
MF
        F1         F2         F3
 0.8013333 -0.6446667 -0.1566667
```

Vi deler opp YIELD fordelt på de 3 gjødseltypene:

```
F1 <- YIELD[1:10]; F1
[1] 6.27 5.36 6.39 4.85 5.99 7.14 5.08 4.07 4.35 4.95
F2 <- YIELD[11:20]; F2
[1] 3.07 3.29 4.04 4.19 3.41 3.75 4.87 3.94 6.28 3.15
F3 <- YIELD[21:30]; F3
[1] 4.04 3.79 4.56 4.55 4.55 4.53 3.53 3.71 7.00 4.61
```

subset()

En annen måte å plukke ut en del av et datasett er å bruke kommandoen **subset**, og logisk er lik ==

```
F1 <- subset(YIELD,FERTIL == "F1") #plukker ut F1
F2 <- subset(YIELD,FERTIL == "F2") #plukker ut F2
F3 <- subset(YIELD,FERTIL == "F3") #plukker ut F3
```

Beregn avvik av YIELD fra gjennomsnittsverdiene for de 3 gjødseltypene: FY, her fordelt på F1, F2 og F3

```
FY1 <- F1 - mean(F1); FY1
 [1]  0.825 -0.085  0.945 -0.595  0.545  1.695 -0.365 -1.375 -1.095 -0.495
FY2 <- F2 - mean(F2); FY2
 [1] -0.929 -0.709  0.041  0.191 -0.589 -0.249  0.871 -0.059  2.281 -0.849
FY3 <- F3 - mean(F3); FY3
 [1] -0.447 -0.697  0.073  0.063  0.063  0.043 -0.957 -0.777  2.513  0.123
```

Beregn feilkvadratsummen SSE (kobler sammen vektorene FY1,FY2,FY3):

```
FY <- c(FY1, FY2, FY3); FY
 [1]  0.825 -0.085  0.945 -0.595  0.545  1.695 -0.365 -1.375 -1.095 -0.495
[11] -0.929 -0.709  0.041  0.191 -0.589 -0.249  0.871 -0.059  2.281 -0.849
[21] -0.447 -0.697  0.073  0.063  0.063  0.043 -0.957 -0.777  2.513  0.123
SSE <- sum(FY^2); SSE
 [1] 25.62215
```

Hvis vi nå samler alle dataene vi har fått i en tabell:

Plot	FERTIL	M	F	Y	MY	MF	FY
1	F1	4.6436	5.445	6.27	1.63	0.80	0.82
2	F1	4.6436	5.445	5.36	0.72	0.80	-0.09
3	F1	4.6436	5.445	6.39	1.75	0.80	0.94
4	F1	4.6436	5.445	4.85	0.21	0.80	-0.60
5	F1	4.6436	5.445	5.99	1.35	0.80	0.55
6	F1	4.6436	5.445	7.14	2.50	0.80	1.70
7	F1	4.6436	5.445	5.08	0.44	0.80	-0.37
8	F1	4.6436	5.445	4.07	-0.57	0.80	-1.38
9	F1	4.6436	5.445	4.35	-0.29	0.80	-1.10
10	F1	4.6436	5.445	4.95	0.31	0.80	-0.50
11	F2	4.6436	3.999	3.07	-1.57	-0.64	-0.93
12	F2	4.6436	3.999	3.29	-1.35	-0.64	-0.71
13	F2	4.6436	3.999	4.04	-0.60	-0.64	0.04
14	F2	4.6436	3.999	4.19	-0.45	-0.64	0.19
15	F2	4.6436	3.999	3.41	-1.23	-0.64	-0.59
16	F2	4.6436	3.999	3.75	-0.89	-0.64	-0.25
17	F2	4.6436	3.999	4.87	0.23	-0.64	0.87
18	F2	4.6436	3.999	3.94	-0.70	-0.64	-0.06
19	F2	4.6436	3.999	6.28	1.64	-0.64	2.28
20	F2	4.6436	3.999	3.15	-1.49	-0.64	-0.85
21	F3	4.6436	4.487	4.04	-0.60	-0.16	-0.45
22	F3	4.6436	4.487	3.79	-0.85	-0.16	-0.70
23	F3	4.6436	4.487	4.56	-0.08	-0.16	0.07
24	F3	4.6436	4.487	4.55	-0.09	-0.16	0.06
25	F3	4.6436	4.487	4.55	-0.09	-0.16	0.06
26	F3	4.6436	4.487	4.53	-0.11	-0.16	0.04
27	F3	4.6436	4.487	3.53	-1.11	-0.16	-0.96
28	F3	4.6436	4.487	3.71	-0.93	-0.16	-0.78
29	F3	4.6436	4.487	7.00	2.36	-0.16	2.51
30	F3	4.6436	4.487	4.61	-0.03	-0.16	0.12
d.f.		1	3	30	29	2	27
SS					36.44	10.82	25.62

SSY (SS_T,36.44) = SSF (SS_G,10.82) + SSE (SS_E,25.62)

Her er et eksempel på hvordan datasettet kan splittes på de enkelte faktorene. Den lager en liste med vektorer med basis til nivåene til en faktor:

```
sy <- split(YIELD, FERTIL); sy
```

```
$F1
 [1] 6.27 5.36 6.39 4.85 5.99 7.14 5.08 4.07 4.35 4.95

$F2
 [1] 3.07 3.29 4.04 4.19 3.41 3.75 4.87 3.94 6.28 3.15

$F3
 [1] 4.04 3.79 4.56 4.55 4.55 4.53 3.53 3.71 7.00 4.61
```

En annen måte å plukke ut en del av et datasett er å bruke kommandoen **subset**. Eller bruk følgende hvor man lager objektene f1, f2, f3:

```
f1 <- YIELD[FERTIL == "F1"]
f2 <- YIELD[FERTIL == "F2"]
f3 <- YIELD[FERTIL == "F3"]
```

Kommandoen **aggregate()** sammen med **list** kan brukes til å samle data i grupper for eksempel beregne gjennomsnitt av YIELD.

```
fertilfaktor <- list(FERTIL = FERTIL)
YIELDmean <- aggregate(YIELD,by = fertilfaktor, mean)
YIELDmean
```

```
  FERTIL     x
1     F1 5.445
2     F2 3.999
3     F3 4.487
```

Kontrastene som er brukt i sammenligningene. Vi ser at F1 er brukt som referanse:

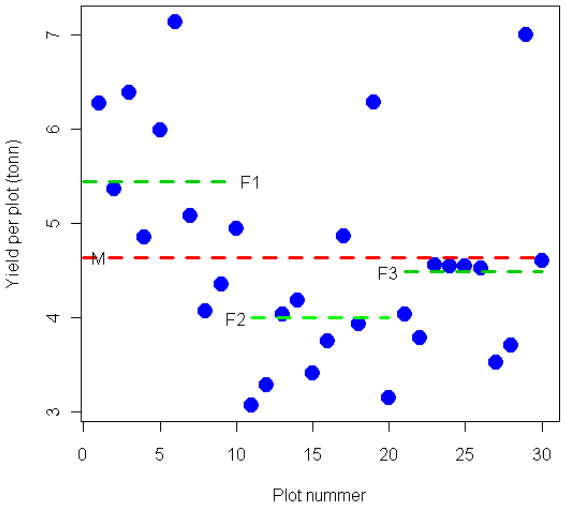
```
contrasts(FERTIL)
```

```
   F2 F3
F1  0  0
F2  1  0
F3  0  1
```

Detaljer om avvik fra gjennomsnittene

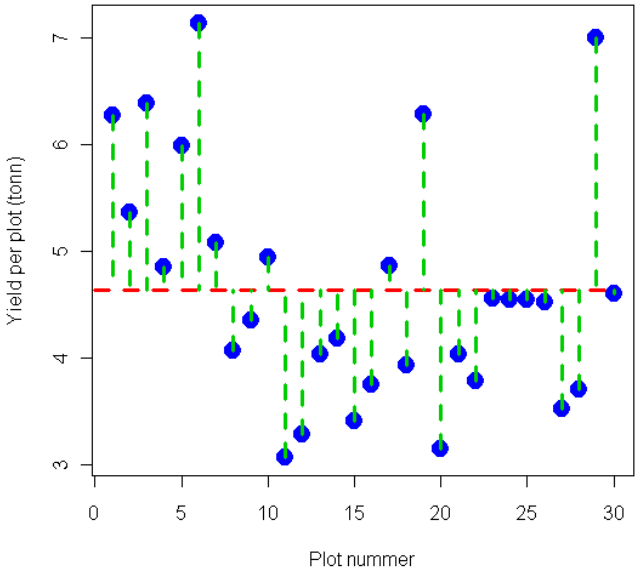
Nedenfor er det vist på figurer hva som skjer når vi utfører ANOVA på datasettet. Finner stormiddeltallet M og middeltallene for de 3 gjødseltypene F1, F2, F3

```
plot(YIELD,xlab =  "Plot nummer",
     ylab = "Yield per plot (tonn)", col = 4, pch = 19,
     cex =  2)
lines(c(0,30),c(mean(YIELD), mean(YIELD)), lty = 2,
     col = "red", lwd = 3)
lines(c(0,10),c(mean(F1),mean(F1)),lty = 2,
     col = 3, lwd = 3)
lines(c(11,20),c(mean(F2), mean(F2)),lty =  2,
     col =  "green", lwd =  3)
lines(c(21,30),c(mean(F3),mean(F3)), lty =  2,
     col = 3, lwd =  3)
text(11, 5.45, "F1")
text(10, 4, "F2")
text(20, 4.49, "F3")
text(1, 4.64, "M")
```



Bestemmer avvik fra stormiddeltallet:

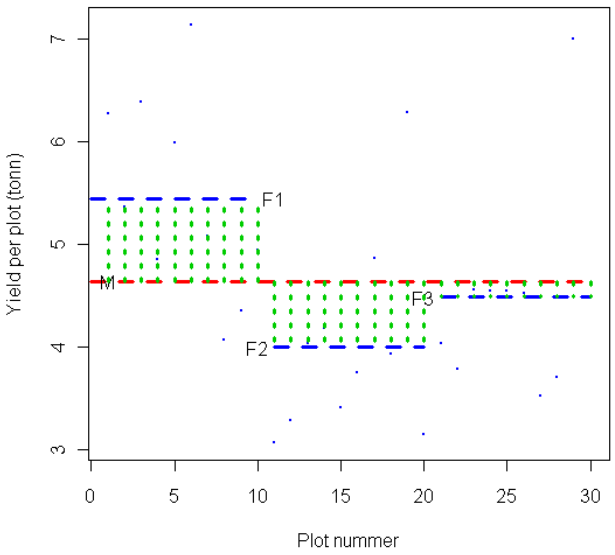
```
plot(YIELD, col = 4, pch = 19,cex = 2, xlab = "Plot nummer",
     ylab =  "Yield per plot (tonn)")
lines(c(0,30),c(mean(YIELD),mean(YIELD)),lty =  2,lwd =  3,
     col = 2)
for(i in 1:30)lines(c(i,i), c(YIELD[i], 4.64), lty =  2,
     lwd = 3, col = 3)
```



Bestemmer avvik mellom stormiddeltallet og gjennomsnittene av F1, F2 og F3.

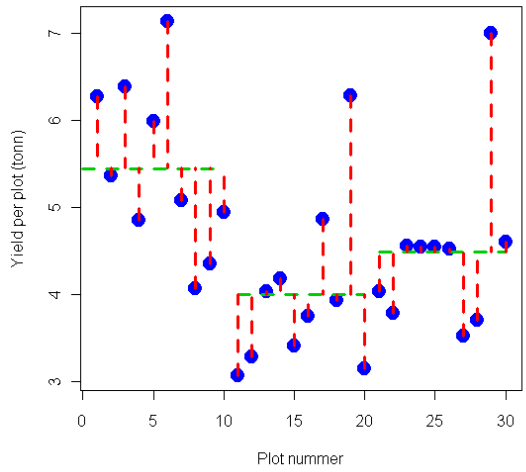
```
plot(YIELD,xlab =  "Plot nummer",
     ylab = "Yield per plot (tonn)", col = 4, pch = 19,
     cex =  0.2)
lines(c(0,30),c(mean(YIELD), mean(YIELD)), lty = 2,
     col = "red", lwd = 3)
```

```
lines(c(0,10),c(mean(F1),mean(F1)),lty = 2,
     col = 4, lwd = 3)
lines(c(11,20),c(mean(F2), mean(F2)),lty =  2,
     col = 4, lwd =  3)
lines(c(21,30),c(mean(F3),mean(F3)), lty =  2,
     col = 4, lwd =  3)
text(11, 5.45, "F1")
text(10, 4, "F2")
text(20, 4.49, "F3")
text(1, 4.64, "M")

for(i in 1:10)lines(c(i,i), c(4.64, mean(F1)), lty =  3,
     lwd = 3, col = 3)
for(i in 11:20)lines(c(i,i),c(4.64, mean(F2)), lty =  3,
     lwd =  3, col = 3)
for(i in 21:30)lines(c(i,i),c(4.64, mean(F3)), lty =  3,
     lwd =  3, col = 3)
```



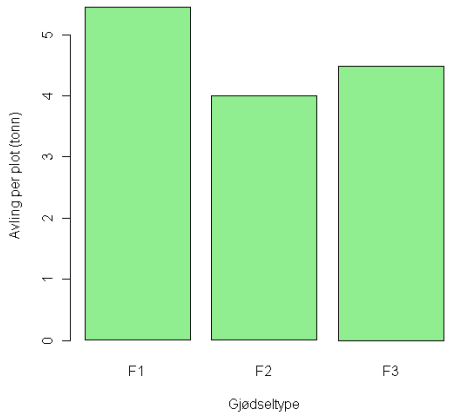
Til slutt bestemmes avvik fra hver av gjennomsnittene for F1, F2 og F3:

```
plot(YIELD,xlab =  "Plot nummer",ylab =  "Yield per plot (tonn)",col =
"blue",pch =  19,cex =  2)
lines(c(0,10),c(mean(F1),mean(F1)),lty =  2,lwd = 3,col = 3)
lines(c(11,20),c(mean(F2),mean(F2)),lty =  2,lwd =  3,col = 3)
lines(c(21,30),c(mean(F3),mean(F3)),lty =  2,lwd =  3,col = 3)
for(i in 1:10)lines(c(i,i),c(YIELD[i],5.45),lty =  2,lwd =  3,col = 2)
for(i in 11:20)lines(c(i,i),c(YIELD[i],4.00), lty =  2, lwd =  3,
     col = 2)
for(i in 21:30)lines(c(i,i),c(YIELD[i],4.49), lty =  2, lwd =  3,
     col = 2)
```



Hvis vi skal få R til å lage stolpediagram med error-bar må vi lage et program som gjør dette. Kommandoen barplot behøver en y-akse i form av en vektor, i vårt tilfelle bruker vi gjennomsnittsverdiene:

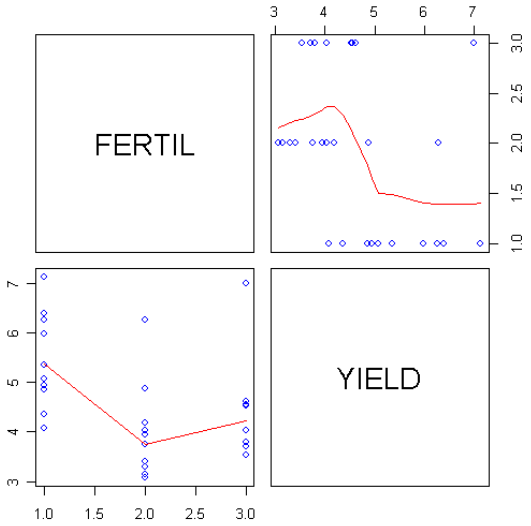
```
labels <- as.character(levels(FERTIL))
yakse <- as.vector(tapply(YIELD,FERTIL, mean))
barplot(yakse,names =  labels,xlab =  "Gjødseltype",
      ylab =  "Avling per plot (tonn)",col = "lightgreen")
```



Man kan også bruke kommandoen nedenfor for å lage et barplot (stolpediagram).

```
b barplot(tapply(YIELD,list(FERTIL),mean),xlab = "Gjødseltype",
      ylab = "Yield per plot (tonn)",col = "lightgreen")
```

```
pairs(oppg1, panel = panel.smooth,col = 4)
```



Vi bruker ANOVA når vi skal sammenligne flere prøver.
Skal vi sammenligne bare to prøver er det forskjellige kommandoer som kan brukes: Students t-test(**t.test()**)som forutsetter normalfordeling eller Wilcoxon-rangeringssumtest (**wilcox.test()**) som ikke forutsetter normalfordeling (ikke-parametrisk test). Rangering (rang) er en ordnet tallrekke etter størrelse.
Andre tester er **korrelasjonstest** (**cor.test()**), kjikvadrattest på tellinger og kontingenstabeller (**chisq.test()**, Fishers test på telledata (**fisher.test()**) avhengig av prøvetype, forutsetninger og formål.

Vi kan gjøre en parvis sammenligning mellom F1 og F2.
Det er en innebygget F-test (**var.test()**) som sammenligner varians. Er varians forskjellig kan vi ikke sammenligne middeltallene med en t-test. Vi gjør først en F-test på F1 og F2:

```
F1 <- YIELD[1:10]; F1
F2 <- YIELD[11:20]; F2
var.test(F1, F2)
```

```
        F test to compare two variances

data:  F1 and F2
F =   1.0087, num df  =   9, denom df  =   9, p-value  =   0.9899
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.2505533 4.0611252
sample estimates:
ratio of variances
          1.008726
```

Vi beholder hypotesen, varians er ikke signifikant forskjellig for F1 og F2.

Bruker Students t-test for to uavhengige prøver med konstant varians og normalfordelte errors. H_0 er at det ikke er noen forskjell mellom F1 og F2, med frihetsgrader n_1-n_2-2 = 18 df. Er det samme prøven som sammenlignes før og etter

en behandling brukes **parvis t-test** og man legger inn kommandoen **paired = TRUE**.

```
t.test(F1, F2, var.equal = TRUE)
```

```
        Welch Two Sample t-test

data:  F1 and F2
t  = 3.3201, df = 18, p-value  = 0.003808
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.5309905 2.3610095
sample estimates:
mean of x mean of y
    5.445     3.999
```

Konklusjon: H_0 forkastes, F1 og F2 er signifikant forskjellig.
Vi kan se at vi får den samme p-verdi (halesannsynlighet) ved å lage en lineær modell med de to prøvene F1 og F2 som responsvariabel.

```
f1f2 <- c(F1,F2); f1f2 #
u <- factor(c(rep(1,10), rep(2,10))); u
summary(lm(f1f2 ~ u)) #u er prediktor
```

```
Call:
lm(formula  =  f1f2  ~  u)
Residuals:
    Min      1Q  Median      3Q     Max
-1.3750 -0.6235 -0.1670  0.6150  2.2810
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.4450     0.3080   17.68    8e-13 ***
u2           -1.4460     0.4355   -3.32  0.00381 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9739 on 18 degrees of freedom
Multiple R-squared: 0.3798,     Adjusted R-squared: 0.3454
F-statistic: 11.02 on 1 and 18 DF,  p-value: 0.003808
```

Ekvivalentene til kommandoene **pnorm()** og **qnorm()** for normalfordeling er **pt()** og **qt()** for t-fordelingen.
kritisk tabellverdi for t med df = 18:

```
qt(0.975, 18)
```

```
[1] 2.100922
```

Beregnet t-verdi, testobservator, er større enn kritisk t-verdi og nullhypotesen om at F1 og F2 er like forkastes.

Konfidensintervall = ± t-verdi · standardfeil

Wilcoxon er en ikke-parametrisk test som ikke forutsetter normalfordeling og er basert på rangeringer:

```
wilcox.test(F1, F2)
```

```
        Wilcoxon rank sum test

data:  F1 and F2
W  =  88, p-value  =  0.002879
alternative hypothesis: true mu is not equal to 0
```

Skew og kurtosis

Hentet fra: Crawly, M.J.C.:*The R Book*. John Wiley & Sons Ltd.2007 s. 285-289

Skew (skjevfordeling), positiv skew til venstre og negativ skew til høyre:

```
skew <- function(x){
     m3 <- sum((x-mean(x))^3)/length(x)
     s3 <- sqrt(var(x))^3
     m3/s3}
skew(YIELD)
[1] 0.6949096
```

Hvor sannsynlig er det å få en t-verdi 0.6949096 når det ikke er noen skew ?

```
1 - pt(1.553865, 29)
[1] 0.06553163
```

Vi ser at p-verdien er >0.05 som indikerer at det er sannsynlig at skewverdien ikke er forskjellig fra 0.

Kurtosis (spisshet):

```
kurtosis <- function(x){
     m4 <- sum((x-mean(x))^4)/length(x)
     s4 <- var(x)^2
     m4/s4-3}
kurtosis(YIELD)
[1] -0.5169758
kurtosis(YIELD)/sqrt(24/length(YIELD))
[1] -0.5779965
1-pt(-0.5779965,29)
[1] 0.716136
```

Kruskal-Wallis test

Kruskal-Wallis rangerings sumtest **kruskal.test()** er et ikke-parametrisk alternativ til enveis anova. Legg merke til at her påtreffes kjikvadratfordelingen.

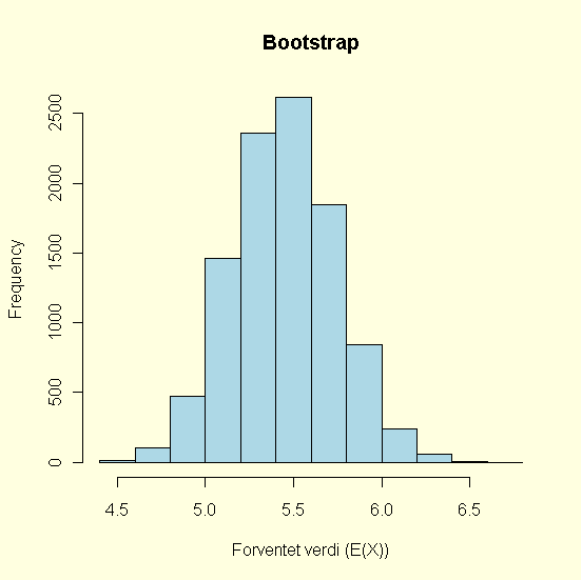
```
kruskal.test(YIELD ~ FERTIL)
        Kruskal-Wallis rank sum test
data:  YIELD by FERTIL
Kruskal-Wallis chi-squared  =   10.5847, df  =   2, p-value  =   0.00503
```

Bootstrap konfidensintervall

Ved bootstrap og **resampling** er det mulig å estimere gjennomsnittsverdi og konfidensintervall. Vi bruker kommandoen **sample(, replace = TRUE)** for å lage et tilfeldig uttrekk fra prøven med tilbakelegging.

```
F1 <- YIELD[1:10]; F1
[1] 6.27 5.36 6.39 4.85 5.99 7.14 5.08 4.07 4.35 4.95
F2 <- YIELD[11:20]; F2
F3 <- YIELD[21:30]; F3
#resampler først for F1
Y <- sample(F1, replace = TRUE); Y
[1] 4.95 4.85 5.08 4.85 6.27 4.95 5.36 5.36 4.95 4.07
#antall simuleringer
n <- 10000
#Lager matriser for å lagre forventede verdier fra hver
#simulering
xmu <- matrix(NA,n,1) #matrise for gjennomsnitt
sdx <- matrix(NA, n, 1)  #matrise for standardavvik
for(i in 1:n)
    {
```

```
    xmu[i] <- mean(sample(F1, replace = T))
    }
#Lager et histogram over forventede verdier (gjennomsnitt)
hist(xmu, col = "lightblue",xlab = "Estimat av forventet verdi
(E(X))",  main =  "Bootstrap")
#Bestemmer gjennomsnitt, standardfeil og kritisk t-verdi 9 df
ex <- mean(xmu); ex
[1] 5.447315
sta <- sd(xmu); sta
[1] 0.2917465
tobs <- qt(0.975, 9);tobs #kritisk verdi for t df = 9
[1] 2.262157
ex+tobs*sta
[1] 6.107291
ex-tobs*sta
[1] 4.787338
quantile(xmu,c(0.25, .975))
  25% 97.5%
5.247 6.029
```



Bootstrap og resampling for F1. Gjennomsnittsverdi ved resampling estimert til 5.447, 95% konfidensintervall CI: 4.787-6.107
Kvantinintervallet for gjennomsnittet: 5.247-6.029.
Gjør tilsvarende for F2 og F3.

Vi kan se hvilke objekter vi har laget:

```
objects()
```

Noen ganger kan det være lurt å renske opp i objektene du har samlet i minnet ved å fjerne dem:

```
rm(list =  ls(all = TRUE))
```

Datasett i R

R inneholder flere datasett bl.a. i pakken datasets.

```
library(datasets)
data()
```

I pakken MASS (Venables & Ripley) er det et datasett fra et klassisk faktorielt eksperiment med undersøkelse av effekten av nitrogen (N), fosfat (P) og kalium (K) på vekst (yield) av erter, fordelt på 6 blokker, og som ligner på vårt labeksperiment.

```
library(MASS)
data(npk)
attach(npk)
names(npk)
npk
?npk
par(mar =  c(12,4,4,2))
par(las =  3) #vertikal aksetekst
boxplot(yield ~  N * P * K, xlab = "NPK",ylab = "yield")
summary(aov(yield ~ block + N*P*K)) #anova-tabell
detach(oppg1)
```

Oppgave 2 Lineær regresjon

Datasettet til oppgave 2 er hentet fra R, og viser sammenhengen mellom volum (cubic ft), diameter (girth, inch) og høyde (height, ft)for 31 felte kirsebærtrær. Vi må regne om fra tommer (1 inch = 2.54 cm) og fot (1 foot = 30.38 cm)til centimeter. Transformasjon vil si å endre måleskalaen.

```
library(datasets)
data(trees)
trees2 <- transform(trees, DIAM = Girth * 2.54/100,
     HEIGHT =  Height * 30.48/100,
     VOLUME =  Volume * (30.48/100)^3)
trees2
oppg2 <- trees2[,c("DIAM","HEIGHT","VOLUME")]
```

Eksperiment:

En skogeier ønsker å finne sammenhengen mellom volumet (VOLUME) av hoggbart skogvirke ut fra måling av høyden (HEIGHT) av trærne og diameter (DIAMET) målt 1.5 meter over bakken. Som statistisk utvalg hogges 31 trær, og fra disse måles volum, høyde og diameter 1.5 meter over bakken. Lag forslag til en modell som skogeieren kan benytte for å beregne hogstvolum ut fra diameter og høyde av trærne. Vær oppmerksom på **korrelasjon** mellom høyde(HEIGHT) og diameter(DIAMET). Korrelerte prediktorer og kan ikke begge inngå i modellen samtidig !

Modell

VOLUME ~ HEIGHT + DIAMET + error

som må erstattes med en av de to følgende:

VOLUME ~ HEIGHT + error

VOLUME ~ DIAMET + error

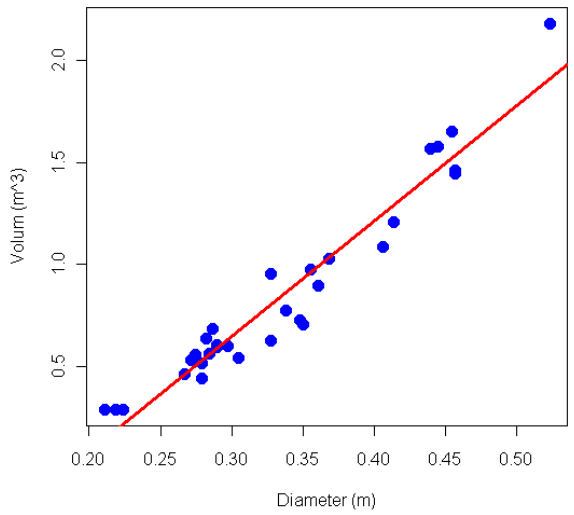
Laster inn datasettet:

```
oppg2 <- read.table("oppg2.txt",header =  TRUE)
attach(oppg2) #husk detach() når du er ferdig med oppgaven
names(oppg2)
```

```
[1] "DIAMET" "HEIGHT" "VOLUME"
summary(oppg2)
     DIAMET           HEIGHT           VOLUME       
 Min.   :0.2108   Min.   :19.20   Min.   :0.2888  
 1st Qu.:0.2807   1st Qu.:21.95   1st Qu.:0.5493  
 Median :0.3277   Median :23.16   Median :0.6853  
 Mean   :0.3365   Mean   :23.16   Mean   :0.8543  
 3rd Qu.:0.3874   3rd Qu.:24.38   3rd Qu.:1.0562  
 Max.   :0.5232   Max.   :26.52   Max.   :2.1804  
dim(oppg2)
[1] 31  3

oppg2[1:5,]
   DIAMET  HEIGHT    VOLUME
1 0.21082 21.3360 0.2916635
2 0.21844 19.8120 0.2916635
3 0.22352 19.2024 0.2888318
4 0.26670 21.9456 0.4643963
5 0.27178 24.6888 0.5323567
```

Første trinn i regresjonsanalyse er å lage et punktskyplot.
Vi plotter datasettet og trekker en linje ut fra en lineær modell:

```
plot(VOLUME ~ DIAMET, col = 4, pch =  16,cex = 1.5,
     xlab = "Diameter (m)",ylab = "Volum (m^3)")
abline(lm(VOLUME ~ DIAMET),col = 2, lwd = 3)
```

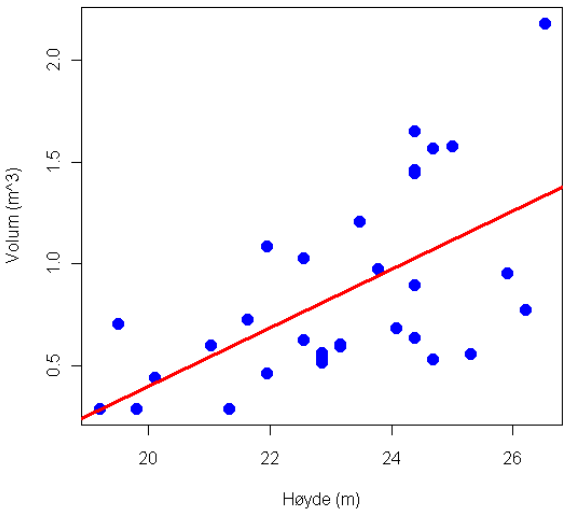


Lineær regresjon angir en relasjon mellom avhengig variabiel y og forklaringsvariabel x(prediktorvariabel, uavhengig variabel, regressor, kovariat). Med **minste kvadraters metode** bestemmes skjæringspunkt (”intercept”) β_0 (også kalt α eller a) og stigningstall (”slope”) β_1 (β eller b) ved å **minimalisere residualkvadratsummen** SS_R.

$$y = \beta_0 + \beta_1 x + \varepsilon$$

Forventet forandring i y ved en økning på 1 i x. Intercept er y-verdien når x=0. Man må skille regresjon fra korrelasjon[-1,1](Pearson, Spearman rang koeffisient, Kandalls tau), hvor sistnevnte er en assosiasjon mellom to tilfeldige varisble
De blå punktene er de observerte verdiene, den rød linjen er de tilpassete verdiene (”fitted”), og residualene er den loddrette avstanden fra tilpasset linje til de enkelte måleverdiene, dvs. forskjell mellom målt verdi og verdi prediktert av modellen. En tilpasset linje går gjennom $(\bar{x}, \bar{y})$, gjennomsnittet for x og y.

```
plot(VOLUME ~ HEIGHT, col = 4, pch = 16, cex = 1.5,
     xlab = "Høyde (m)",ylab = "Volum (m^3)")
abline(lm(VOLUME ~ HEIGHT), col = 2, lwd = 3)

library(ggplot2) #alternativt
ggplot(x, aes(x=DIAMET, y=VOLUME, color = Art)) +
    geom_point(size=2, shape=16) +    #
    geom_smooth(method=lm)   # regresjonslinje
```



En bivariat lineær regresjonsmodell gir en rett linje med formen:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

hvor ε_i er residualer som ikke er forklart av modellen, og følger normalfordeling $\varepsilon_i \sim N(0,\sigma_i^2)$. Forutsetter

- **normalfordeling**, hvis man gjentar forsøket flere ganger så vil verdiene for x følge normalfordeling.
- **homogen varians**, varians er lik for alle x, spredningen for alle verdiene er like. Populasjonsvarians er like, dvs. $\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \ldots = \sigma_n^2$
- **uavhengighe**t, y-verdien for en verdi av x skal ikke påvirke verdien av en annen y-verdi.

Den totale variasjonen i y ,

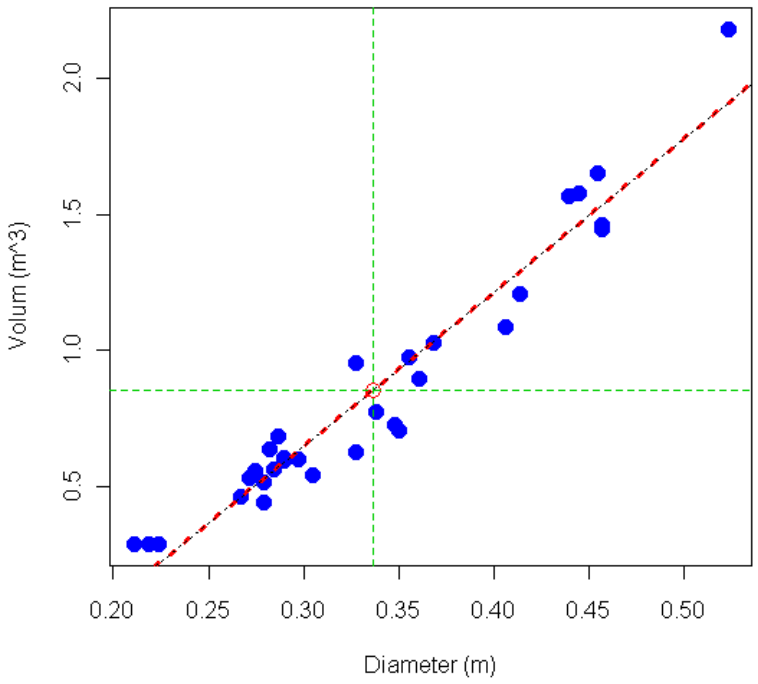
$$SS_{total} = SS_{regresjon} + SS_{residual}$$

hvor $SS_{regresjon}$ er variasjon i y som er forklart av x, og $SS_{residual}$ er uforklart variasjon. Antall frihetsgrader (df) for regresjonskvadratsummen er antall regresjonsparametre minus 1,(df = 2-1 = 1), for residualene er frihetsgradene df = n-2, n er antall par (x,y) og totalantall frihetsgrader er df = n-1

Vi kan finne forventet verdi av volum og høyde og lage et nytt aksesystem med origo i dette gjennomsnittet.
Vi har da fjernet skjæringspunktet med y-aksen β_0 og vi har en rett linje som går gjennom origo med stigningstall β_1.

Vi svinger linjen rundt origo til de kvadrerte avvikene blir minst mulig:

```
plot(VOLUME ~ DIAMET,col = 4, pch = 16, cex = 1.5,
     xlab = "Diameter (m)",ylab = "Volum (m^3)")
mx <- mean(DIAMET)
my <- mean(VOLUME)
points(mx,my,cex = 1.5,col = 2)
abline(v = mx, h = my, lty = 2, col = 3)
#Nye x og y med origo i storgjennomsnitt
xny <- DIAMET-mx
yny <- VOLUME-mx
XY <- sum(xny*yny) #produktsummen
XX <- sum(xny^2)  #kvadratsummen
b1 <- XY/XX;b1  #finner stigningstallet
b0 <- my - b1 * mx; b0  #finner skjæringspunkt (intercept)
#trekker regresjonslinje
abline(b0, b1, lty = 3, col = 2, lwd = 3)
#Ser at dette stemmer med en lineær modell
mod <- lm(VOLUME ~ DIAMET)
abline(mod, lty = 3)
```



$$y = -1.04612 + 5.64760x + \varepsilon$$

Deretter må man med ANOVA undersøke om stigningstallet β_1 er signifikant forskjellig fra null ($H_0 = \beta_1 = 0$ med $F = MS_{regresjon}/MS_{residual}$), og hvor god regresjonslinjen er til å prediktere.

Vi kan også finne stigningstallet via følgende, hvor s_y og s_x er standardavviket for henholdsvis *y* og *x* og *r* er korrelasjonskoeffisienten

$$\beta_1 = r \cdot \frac{s_y}{s_x}$$

Korrelasjonskoeffisienten r er:

$$r = \frac{1}{n-1} \cdot \sum_{i=1}^{n} \frac{(x_i - \bar{x})}{s_x} \cdot \frac{(y_i - \bar{y})}{s_y}$$

```
#Alternativ regnemåte for stigningstall
sdx <- sd(DIAMET) #standardavvik
sdy <- sd(VOLUME)
r <- cor(DIAMET, VOLUME) #korrelasjonskoeff
b12 <- (sdy/sdx)*r; b12
5.647602
b02 <- mean(VOLUME) - b12 * mean(DIAMET); b02
[1] -1.046122
```

Konfidensintervallet for intercept β_0, SE er standardfeilen og *t* er kritisk verdi for t ved n-2 frihetsgrader:

$$\beta_0 \pm t \cdot SE_{\beta o}$$

Konfidensintervallet for stigningstallet β_1 blir:

$$\beta_1 \pm t \cdot SE_{\beta 1}$$

For å teste nullhypotesen H0:$\beta_1 = 0$ så beregner vi *t*, finner hvor sannsynlig det er å få en slik t-verdi sammenlignet med dem kritiske t-verdi med n-2 frihetsgrader.

$$t = \frac{\beta_1}{SE_{\beta 1}}$$

Vi skal nå se på nærmere detaljer og starter først med en lineær regresjon og ser på sammenhengen mellom diameter og volum.

Modell: VOLUME ~ DIAMET

ANOVA-tabellen:

```
model1 <- lm(VOLUME ~ DIAMET)
summary.aov(model1)
            Df Sum Sq Mean Sq F value    Pr(>F)
DIAMET       1 6.0794  6.0794  419.36 < 2.2e-16 ***
Residuals   29 0.4204  0.0145
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model1 <- lm(VOLUME ~ DIAMET)
summary(model1)

Call:
lm(formula = VOLUME ~ DIAMET)

Residuals:
      Min        1Q    Median        3Q       Max
-0.228386 -0.087972  0.004303  0.098961  0.271468

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.04612    0.09529  -10.98 7.62e-12 ***
DIAMET       5.64760    0.27578   20.48  < 2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1204 on 29 degrees of freedom
Multiple R-squared: 0.9353,     Adjusted R-squared: 0.9331
F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16
```

Vi ser at vi får de samme koeffisientene som tidligere:
En lineær regresjonslinje er av typen

$$\mu_y = \beta_0 + \beta_1 x + \varepsilon$$

hvor gjennomsnittsverdien μ endrer seg når x endres.

$$y = -1.04612 + 5.64760x + \varepsilon$$

Koeffisientene gir estimater av intercept β_0 og og stigningskoeffisient β_1, inkludert standardfeil, t-verdi og p-verdi og en eller flere * angir signifikansnivå.
Residualene sier litt om residualene og fordeling av disse, middelverdiene av residualene skal være lik 0, og medianverdien bør ligge i nærheten av dette, og man kan se på kvartilene om dataene ser balansert ut. Verdien for skjæringspunkt med y-aksen, α, er interessant bare hvis man skal avgjøre om regresjonslinjen går gjennom origo eller ikke, men mest interesse knytter det seg til stigningskoeffisienten β. R^2 er kvadratet av Pearsons korrelasjonskoeffisient. F-test for hypotesen om at regresjonskoeffisienten er lik 0.

Formulert som en lineær modell:

$$VOLUME \sim -1.04612 + 5.64760 \cdot DIAMET + error$$

Vi kan gjøre det tilsvarende for sammenhengen mellom volum og høyde:

```
model2 <- lm(VOLUME ~ HEIGHT)
summary.aov(model2)
            Df Sum Sq Mean Sq F value    Pr(>F)
HEIGHT       1 2.3263  2.3263  16.165 0.0003784 ***
Residuals   29 4.1735  0.1439
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

model2 <- lm(VOLUME ~ HEIGHT)
summary(model2)

> Call:
lm(formula = VOLUME ~ HEIGHT)

Residuals:
     Min       1Q   Median       3Q      Max
-0.60242 -0.28018 -0.08195  0.34171  0.84532

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.46707    0.82892  -2.976 0.005835 **
HEIGHT       0.14338    0.03566   4.021 0.000378 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3794 on 29 degrees of freedom
Multiple R-squared: 0.3579,     Adjusted R-squared: 0.3358
F-statistic: 16.16 on 1 and 29 DF,  p-value: 0.0003784
```

Som gir regresjonsmodellen:

$$VOLUME \sim -2.46707 + 0.14338 \cdot HEIGHT + error$$

Source	df	SS	MS	F	F tabell (5%)
Regression	1	SSR	SSR	F = SSR/s^2	qf(0.95,1,n-2)
Error	n-2	SSE	s^2 = SSE/(n-2)		
Total	n-1	SST			

ANOVA-tabell for regresjon df – antall frihetsgrader, SS – kvadratsum, MS-middelkvadratsum, SST (SS_T) – totalkvadratsum, SSE (SS_E)– feilkvadratsum, SSR (SS_R) - regresjonskvadratsum

Varians = SS/df
SST = SSR + SSE
$y = \beta_0 + \beta_1 x$
R^2 = SSR/SST = 2.32631/6.499815 = 0.357904
Adjusted R^2 = 1- [(SSE/(n-2))/(SST/(n-1))] = 1- [(4.173505/29))/(6.499815/30)] = 0.3357628

Fra R^2 = 0.357904 ser vi at 35.8% av variasjonen kan forklares med regresjonslinjen.
R^2 øker med hvor mye variasjon som blir forklart.
Jusert R^2 (Adjusted R^2)er lik:

$$R^2_{adj} = 1 - \left(\frac{\frac{SSE}{n-2}}{\frac{SST}{n-1}}\right) = 1 - \left(\frac{n-1}{n-p}\right) \cdot (1 - R^2)$$

SSE er feilkvadratsum, SST er totalkvadratsum, *n* er prøvestørrelsen, *p* er antall parametere i modellen og p = 2 for vanlig lineære regresjon. Adjusted R^2 minsker med antall parametere som tillegges modellen.

t-verdien får vi ved å dividere estimatene på standardfeilen. F-verdi (F-ratio, F-observator) er:
F = SSR/s^2 = 2.32631/0.1439140 = 16.16458

```
#tabellverdi for kritisk verdi av F
qf(0.95, 1, 29)
[1] 4.182964
```

Sannsynligheten for å få en F-verdi vi har fått:

```
1-pf(16.16458, 1, 29)
[1] 0.0003783709
```

Det vil si vi forkaster nullhypotesen om et det ikke er noen lineær sammenheng mellom VOLUME og HEIGHT.
Responsvariabel VOLUME (y) og forklaringsvariabel HEIGHT (x)

Trenger følgende tall:

$\sum x$ = 718.1088
$\sum x^2$ = 16748.00
$\sum y$ = 26.48475

$\sum y^2 = 29.12697$
$\sum xy = 629.7384$
$(\sum y)^2 = 701.4418$
$(\sum x)^2 = 515680.2$

```
sum(HEIGHT)
[1] 718.1088
sum(VOLUME)
[1] 26.48475
sum(HEIGHT^2)
[1] 16748.00
sum(VOLUME^2)
[1] 29.12697
sum(HEIGHT*VOLUME)
[1] 629.7384
(sum(HEIGHT)^2)
[1] 515680.2
(sum(VOLUME)^2)
[1] 701.4418
```

Kovarianse mellom VOLUME og HEIGHT:

```
VOLUME * HEIGHT
  [1]  6.222933  5.778438  5.546264 10.191455 13.143248 14.112507  8.886452
 [8] 11.781281 15.604803 12.881730 16.500698 13.775036 14.037417 12.684944
[15] 12.363871 14.178965 24.796791 20.338029 15.748940 13.754322 23.225953
[22] 21.888152 23.184525 23.800776 28.311324 38.730637 39.421114 40.254867
[29] 35.559616 35.214377 57.818900
```

Noen av kvadratsummene vi trenger for å regne ut og noen forenklete formler for lommekalkulator:

$$SSY = \sum_{i=1}^{n} (y_i - \overline{y})^2$$

$$SSX = \sum_{i=1}^{n} (x_i - \overline{x})^2$$

$$SSXY = \sum_{i=1}^{n} (x_i - \overline{x})^2 (y_i - \overline{y})^2$$

$$\beta_1 = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\beta_1 = \frac{SSXY}{SSX}$$

$$\beta_0 = \overline{y} - \beta_1 \overline{x}$$

Kovariansen mellom x og y:

$$cov(x, y) = \rho s_x s_y$$

hvor r eller rho(ρ) er korrelasjonskoeffisienten

SST = $\sum y^2 - 1/n(\sum y)^2$

SSX = $\sum x^2 - 1/n\ (\sum x)^2$

SSXY = ∑ xy – 1/n(∑ x ∑ y)

SST = 29.12697 – 1/31*701.4418 = 6.499815
SST = SSR + SSE = 6.0794 + 0.4202 = 6.4998

SSX = 16748.00 – 1/31*515680.2 = 113.1548

SSXY = 629.7384 – 1/31* (718.1088 *26.48475) = 16.22446

Stigningskoeffisienten β_1:
β_1 = SSXY/SSX = 16.22446/113.1548 = 0.1433829

Skjæring med y-aksen, intercept (linje gjennom stormiddeltallet)
β_0 = y_m – bx_m = ∑y/31 – b*∑x/31 = 26.48475/31 – 0.1433829 * 718.1088 /31 = -2.467089

VOLUME = -2.467089 + 0.1433829 * HEIGHT

Regresjonskvadratsummen SSR:

SSR = b*SSXY = 0.1433829 * 16.22446 = 2.32631

Feilkvadratsummen SSE:

SSE = SST – SSR = 6.499815 – 2.32631 = 4.173505

Standardfeilen til regresjonsstigningskoeffisienten b:

s^2 = SSE/(n-2) = 4.173505/29 = 0.1439140
SE_b = $\sqrt{s^2}$/SSX = √0.1439140/113.1548 = 0.03566277

Standardfeilen til skjæring med y-aksen (β_0): (x_m-gjennomsnitt av x, √ = sqrt)

$SE_{\beta 0}$ = $\sqrt{s^2}$[1/n+ ∑ x^2/SSX] = √ ((0.1439140*16748.00
)/(31 * 113.1548)) = 0.8289258

t-verdier:

$t_{\beta 0}$ = a/SE = -2.467089/0.8289258 = -2.976248
$t_{\beta 1}$ = b/SE = 0.1433829 /0.03566277 = 4.020521

Sannsynligheten for å få en slik t-verdi:

```
2*pt(-2.976248, 29)
[1] 0.005834442
```

95% konfidensintervall (CI_β) for β_1:

$CI_{\beta 1}$ = t (α = 0.025, df = (n-2))·$SE_{\beta 1}$

```
#kritisk tabellverdi for t
```

```
qt(0.975, 29)
[1] 2.045230
```

Konfidensintervall for *β0* og *β1*:

$CI_{\beta 1}$ = 0.1433829 ± 2.045230*0.03566277 = 0.1433829 ± 0.07293857

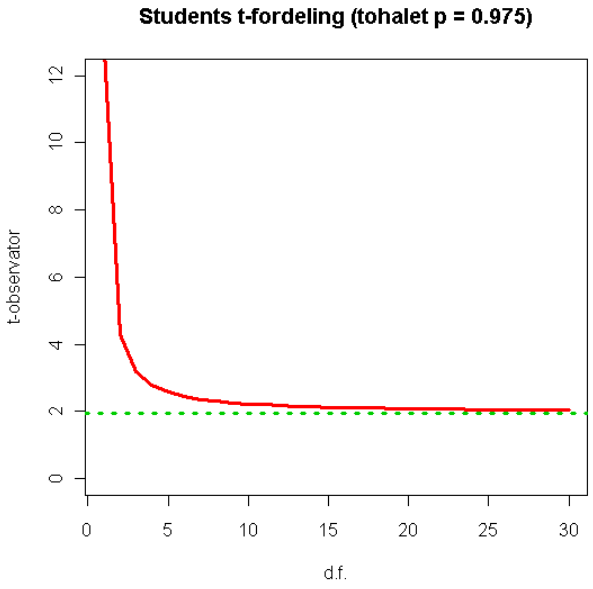
$CI_{\beta 0}$ = -2.467089 ± 2.045230*0.8289258 = -2.467089 ± 1.695344

```
#95% konfidensintervall for parametere
confint(model2)
                 2.5 %     97.5 %
(Intercept) -4.16240288 -0.7717291
HEIGHT       0.07044363  0.2163201
```

Som er det samme som vi har funnet over

Students t-fordeling:
Når *n* øker, dvs. antall frihetsgrader df øker så nærmer t-verdien seg 1.96 fra normalfordelingen:

```
plot(1:30,qt(0.975,1:30),type =  "l",col = 2, lwd =  3,
     xlab  = "d.f.", ylab =  "t-observator",
     ylim =  c(0,12), main =  "Students t-fordeling (tohalet p = 0.975)")
abline(h = 1.96, lty = 3, col = 3, lwd = 3)
```



Vi kan bruke modellen til å prediktere verdier. Hvor stort er volumet ved høyde 23 meter ?

```
model2 <- lm(VOLUME ~ HEIGHT)
predict(model2, list(HEIGHT = 23))
        1
0.8307173
```

Svar: 0.83 m^3.

```
#med konfidensintervall
predict(model2,list(HEIGHT = 23),interval = "conf")
        fit       lwr       upr
1 0.8307173 0.6908479 0.9705868
```

Vi kan finne den totale varians SST også på en annen måte via nullmodellen lm(VOLUME ~ 1):

```
deviance(lm(VOLUME ~ 1))
[1] 6.499813
```

Devians angir dataavvik fra modellen og er lik -2ln(likelihood). Tilsvarer feilkvadratstummen SS_E.

Feilkvadratsummen (SSE) finner vi ved:

```
deviance(lm(VOLUME ~ HEIGHT))
[1] 4.173513
```

SST = SSR+SSE, dvs. SSR = 6.499813-4.173513 = 2.3263

Som vi også finner igjen i ANOVA-tabellen:

```
summary(aov(lm(VOLUME ~ HEIGHT)))
            Df Sum Sq Mean Sq F value    Pr(>F)
HEIGHT       1 2.3263  2.3263  16.165 0.0003784 ***
Residuals   29 4.1735  0.1439
```

R^2 som angir hvor stor prosent var variasjonen som blir forklart av regresjonsmodellen:

$$R^2 = \frac{SST - SSE}{SST} = \frac{6.499813 - 4.173513}{6.499813} = 0.3579026$$

Som er det samme som:

```
summary(lm(VOLUME ~ HEIGHT))[[8]]
[1] 0.3579026
```

Korrelasjonskoeffisent

Korrelasjonskoeffisienten *r for prøven,* rho (ρ)for populasjonen :

$$r = \frac{SSXY}{\sqrt{SSX \cdot SST}} = \frac{16.22446}{\sqrt{113.1548 \cdot 6.499813}} = 0.5982509$$

Som er det samme som korrelasjonen:

```
cor(VOLUME, HEIGHT)
[1] 0.5982497
```

Pearsons korrelasjonstest brukes som default, andre korrelasjonstester mellom parvise variable finner du ved `?cor.test`

```
cor.test(VOLUME, DIAMET)
        Pearson's product-moment correlation
data:  VOLUME and DIAMET
t =   20.4783, df =   29, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9322519 0.9841887
sample estimates:
      cor
0.9671194
```

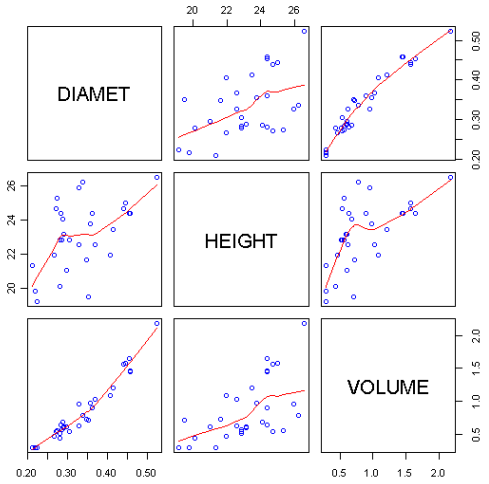
Korrelasjonsmatrisen

```
cor(oppg2) #korrelasjonsmatrise
          DIAMET    HEIGHT    VOLUME
```

```
DIAMET 1.0000000 0.5192801 0.9671194
HEIGHT 0.5192801 1.0000000 0.5982497
VOLUME 0.9671194 0.5982497 1.0000000
```

Vi kan plotte i multipanel med kommandoen **pairs()**:

```
pairs(oppg2, panel = panel.smooth,col = 4)
```



Matriser i statistiske modeller

Skrevet som matrise er en lineær regresjonsmodell lik

$$Y = X \cdot b + e$$

$$y_i = \beta_0 + \beta_1 \cdot x_i + e_i \quad for\ i = 1, 2, 3, \ldots n$$

hvor *e* er normalfordelte feil-ledd (error) og vi ønsker å bestemme *b*.

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \cdot \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_n \end{pmatrix} = \begin{pmatrix} \beta_0 + \beta_1 \cdot x_1 + e_1 \\ \beta_0 + \beta_1 \cdot x_2 + e_2 \\ \beta_0 + \beta_1 \cdot x_3 + e_3 \\ \vdots \\ \beta_0 + \beta_1 \cdot x_n + e_n \end{pmatrix}$$

Nå skal vi gjøre det samme regnestykket med datasett trees3, men nå i form av matriser:

```
VOLUME %*% HEIGHT
          [,1]
[1,] 629.7384
VOLUME %*% VOLUME
          [,1]
[1,] 29.12697
HEIGHT %*% HEIGHT
          [,1]
[1,] 16748.00
```

Vi lager matrisen Y med VOLUME.

```
Y <- VOLUME
```

I matrisen X må vi lage en kolonne med 1-tall og binde disse sammen med HEIGHT.

```
X <- cbind(1, HEIGHT)
```

$\sum y^2 =$ 29.12697

```
t(Y) %*% Y
          [,1]
[1,] 29.12697
```

Vi finner en matrise som inneholder n, √x og √x^2 ved å ta den transponerte matrisen ^tX og gange den med X:

```
tXX <- t(X) %*% X; tXX
                HEIGHT
       31.0000   718.1088
HEIGHT 718.1088 16748.0026
```

Vi finner en matrise som inneholder ∑y og ∑xy:

```
tXY <- t(X) %*% Y; tXY
           [,1]
       26.48475
HEIGHT 629.73836
```

Vi løser matriseligningene og finner β_1

```
b <- solve(tXX,tXY); b
             [,1]
      -2.4670660
HEIGHT  0.1433819
```

Som vi ser er lik koeffisientene i modellen:

VOLUME ~ -2.46707 + 0.14338·HEIGHT + error

$$\begin{pmatrix} 31 & 718.1088 \\ 718.1088 & 16748.0026 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{\beta_0} \\ \boldsymbol{\beta_1} \end{pmatrix} = \begin{pmatrix} 26.48475 \\ 629.73836 \end{pmatrix} \rightarrow \begin{pmatrix} \boldsymbol{\beta_0} \\ \boldsymbol{\beta_1} \end{pmatrix} = \begin{pmatrix} -2.4670660 \\ 0.1433819 \end{pmatrix}$$

Som er det samme som de to ligningene:

$$\beta_0 \cdot n + \beta_1 \cdot \sum x = \sum y$$
$$\beta_0 \cdot \sum x + \beta_1 \cdot \sum x^2 = \sum xy$$

De samme regneoperasjonene kan nå gjøres i matriseform hvis vi har m>1 x-variable, noe som viser den store fleksibiliteten ved å regne med matriser.

Stigningskoeffisienten (slope) β_1

$$\beta_1 = \frac{\sum(x-\bar{x})\cdot(y-\bar{y})}{\sum(x-\bar{x})^2}$$

Skjæringspunktet (intercept) :

$$\beta_0 = \bar{y} - \beta_1 \cdot \bar{x}$$

Appendiks oppgave 2

Nedenfor vises det hvordan man ved regresjonen først finner stormiddeltallet og dreier linjen med senter i stormiddeltallet slik at de kvadrerte avvikene blir minst mulig. Vi har laget et nytt x-,y-koordinatsystem med origo i stormiddeltallet, og har fått en linje av typen y = bx

```
mean(VOLUME)
[1] 0.8543467
```

Antall målepunkter:

```
length(VOLUME)
```

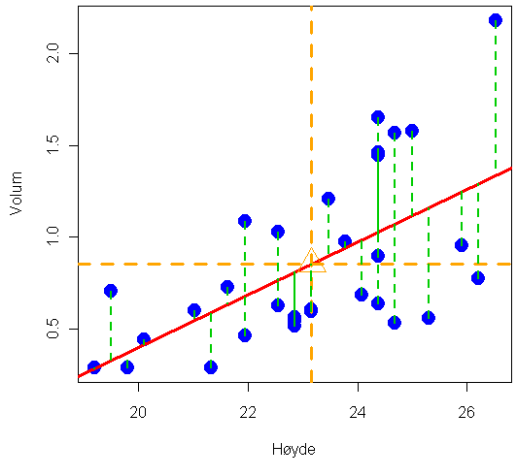
```
[1] 31
```

Tilpasset prediktert linje:

```
fitted <- predict(lm(VOLUME ~ HEIGHT))
```

Trekker prikkete linjer fra hvert punkt ned til regresjonslinjen. Vi lager et nytt koordinatsystem med origo i stormiddeltallet. Regresjonslinjen går da gjennom origo og blir av typen y = bx. Regresjonslinjen vippes rundt det nye origo inntil de kvadrerte avvik blir så liten som mulig. De loddrette linjene fra hvert målepunkt ned/opp til regresjonslinjen blir mål på residualene.

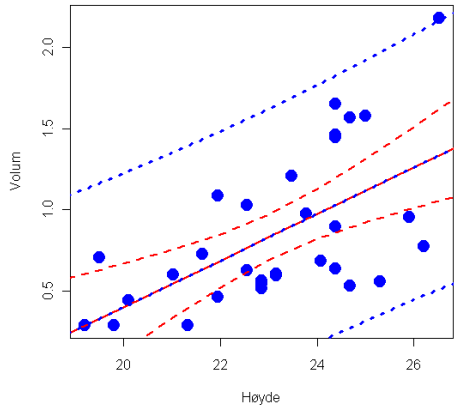
```
plot(HEIGHT,VOLUME,xlab =  "Høyde",
     ylab =  "Volum",pch = 19,cex = 2,col = 4)
abline(lm(VOLUME ~ HEIGHT),col =  2, lwd =  3)
abline(h = mean(VOLUME), lty = 2, lwd = 3, col = "orange")
abline(v = mean(HEIGHT), lty = 2, lwd = 3, col = "orange")
points(mean(HEIGHT), mean(VOLUME),
     pch = 24, col = "orange", cex = 3)
fitted <- predict(lm(VOLUME ~ HEIGHT))
for(i in 1:31)lines(c(HEIGHT[i],
     HEIGHT[i]),c(VOLUME[i],fitted[i]),lty = 2,
     lwd = 2,col = 3)
```



Konfidensintervall og prediksjonsintervall

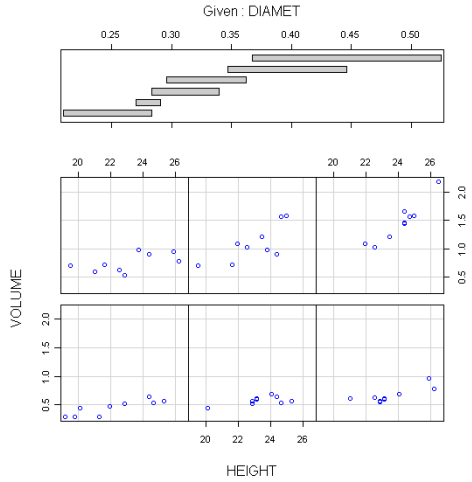
Konfidensintervall og prediksjonsintervall ved å lage en ny dataramme, bruke tilleggskommandoen **interval = "conf"** og plotte med **matlines**.

```
plot(HEIGHT, VOLUME, xlab = "Høyde",
     ylab = "Volum", pch = 19,cex = 2, col = 4)
hoyde <- data.frame(HEIGHT = 18:27)
conf <- predict(lm(VOLUME ~ HEIGHT),
     int =  "confidence", newdata = hoyde)
predheight <- hoyde$HEIGHT
matlines(predheight,conf,lty =  c(1,2,2),
     col = 2, lwd =  2)
conf2 <- predict(lm(VOLUME ~ HEIGHT),
     int = "prediction", newdata = hoyde)
matlines(predheight, conf2, lty = 3, col = 4, lwd = 3)
```



Vi kan også plotte med kommandoen **coplot**:

```
coplot(VOLUME ~ HEIGHT|DIAMET,col = 4)
```



Undersøker homogenitet ved å plotte residualene versus x, og ser som det er en økning eller minskning i residualene langs x-aksen, eller som her plotte residualene versus tilpassete (”fitted”) verdier. Hvis det er økt spredning for større tilpassete verdier så indikerer dette heterogenitet. Cook´s avstand ser på endring i regresjonsparametre ved å utelate innflytelsesrike observasjoner, med test observator D.

Vi tester om model1 og model2 er signifikant forskjellige ved å bruke ANOVA:

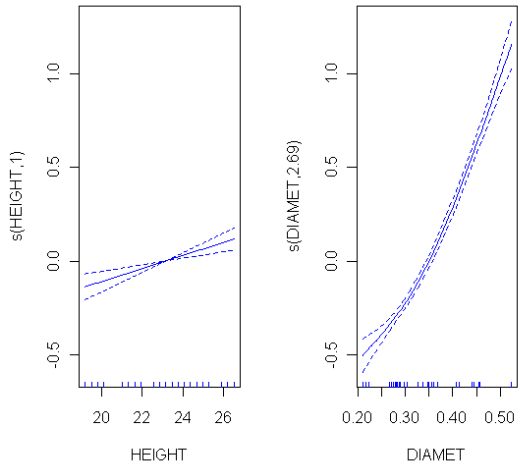
```
cor(HEIGHT, DIAMET)
[1] 0.5192801
cor(VOLUME, DIAMET)
[1] 0.9671194
cor(VOLUME, HEIGHT)
[1] 0.5982497
```

I tilfeller hvor det ikke er noen lineære interaksjon mellom avhengig og uavhengig variable kan man bruke **Generaliserte additive modeller** (**GAM**) med glattingsfunksjon **s**. Dette er ikke påkrevet i vårt tilfelle, men her er vist hvordan dette eventuelt kan se ut.

```
library(mgcv)
This is mgcv 1.4-1
model5 <- gam(VOLUME ~ s(HEIGHT) + s(DIAMET))
```

```
summary(model5)

Family: gaussian
Link function: identity

Formula:
VOLUME  ~  s(HEIGHT) + s(DIAMET)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.85435    0.01364   62.65   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
            edf Ref.df      F p-value
s(HEIGHT) 1.000  1.500  10.67 0.00102 **
s(DIAMET) 2.693  3.193 216.45 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj)  =    0.973   Deviance explained  =   97.7%
GCV score   =  0.0067944   Scale est.  =   0.0057657  n  =   31
```

```
par(mfrow =  c(1,2))
plot(model5, col = 4)
```



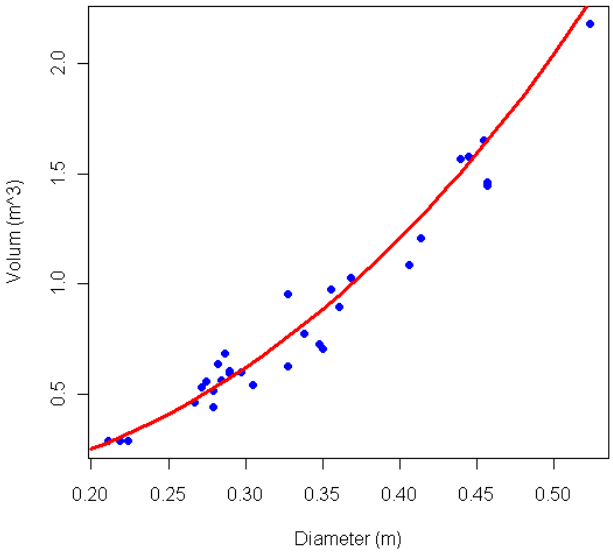
Vi kan i stedet lage en teoretisk betraktning med treet betraktet som en konus hvor volumet blir:
VOLUME = $1/3 \cdot \pi \cdot \text{radius}^2 \cdot \text{høyde}$
Radius = 1/2 diameter
VOLUME = $(\pi \cdot \text{diameter}^2 \cdot \text{høyde})/12$

Imidlertid foretar vi en empirisk justering:
VOLUME = $(\pi \cdot \text{diameter}^2 \cdot \text{høyde})/10$

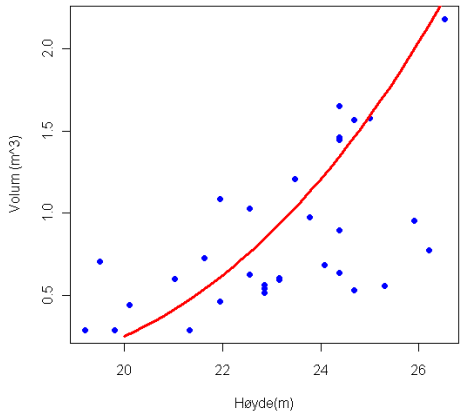
La D og H være henholdsvis diameter og høyde:

```
D <- seq(0.2, 0.55, 0.01)
H <- seq(20, 27, 0.2)
```

```
length(D)
[1] 36
length(H)
[1] 36
>
VOL <- (pi * D^2 * H)/10
plot(VOLUME ~ DIAMET, col = 4,pch = 16, xlab = "Diameter (m)",
     ylab = "Volum (m^3)")
lines(VOL ~ D,col = 2, lwd = 3)
```

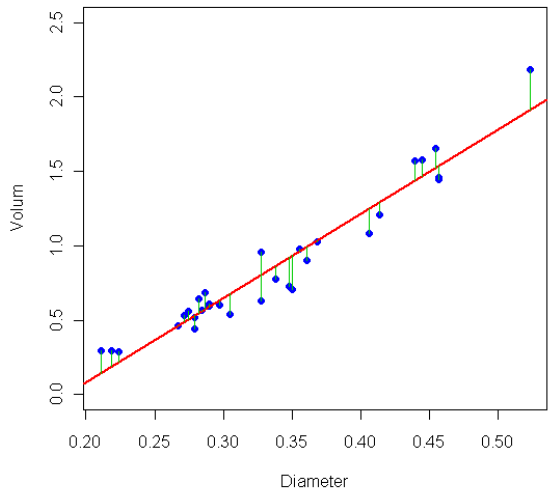



```
D <- seq(0.2, 0.55, 0.01)
H <- seq(20, 27, 0.2)
VOL <- (pi * D^2 * H)/10
plot(VOLUME ~ HEIGHT, col = 4,xlab = "Høyde(m)",
     ylab = "Volum (m^3)", pch = 16)
lines(VOL ~ H, col = 2, lwd = 3)
```



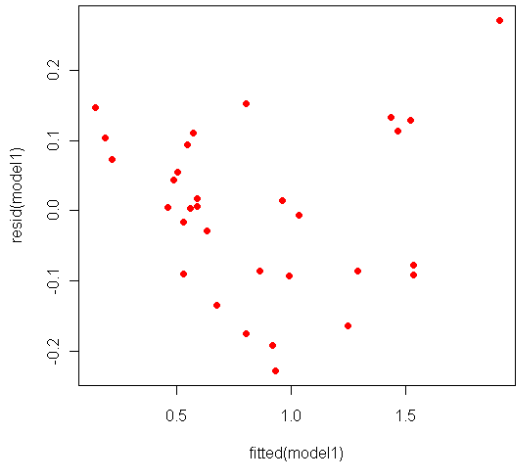

```
model1 <- lm(VOLUME ~ DIAMET)
fitted(model1)
```

```
resid(model1)
plot(DIAMET, VOLUME, ylim =  c(0, 2.5), xlab =  "Diameter",
     ylab =  "Volum", col = 4, pch=16)
abline(lm(VOLUME ~ DIAMET), col = 2, lwd=2)
segments(DIAMET, fitted(model1), DIAMET, VOLUME, col = 3)
```

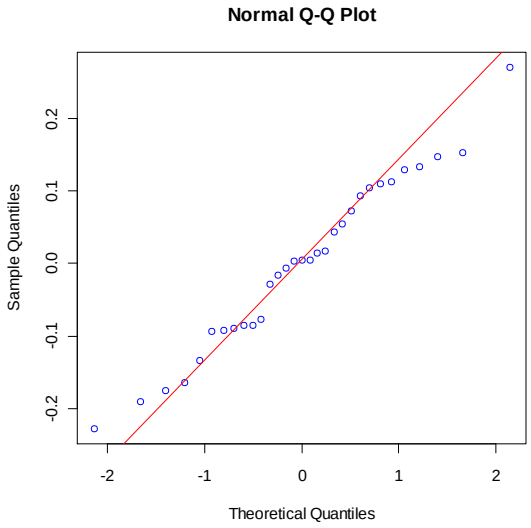


Se på tilpassete verdier og residualer:

```
plot(fitted(model1), resid(model1), col = 2)
```




```
qqnorm(resid(model1), col = 4)
qqline(resid(model1), col = 2)
```

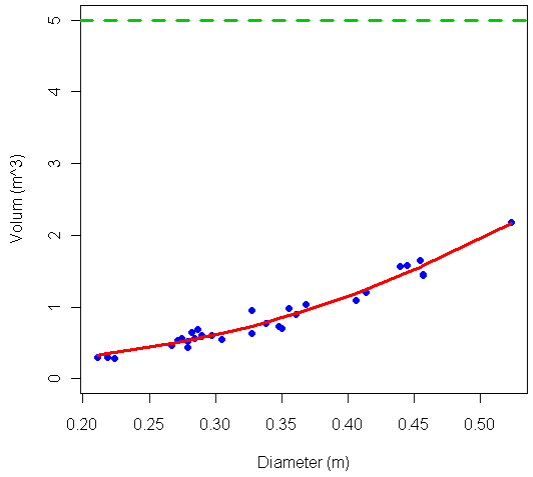



```
shapiro.test(resid(model1))
    Shapiro-Wilk normality test

data:  resid(model1)
W  =  0.9789, p-value  =  0.7811
```

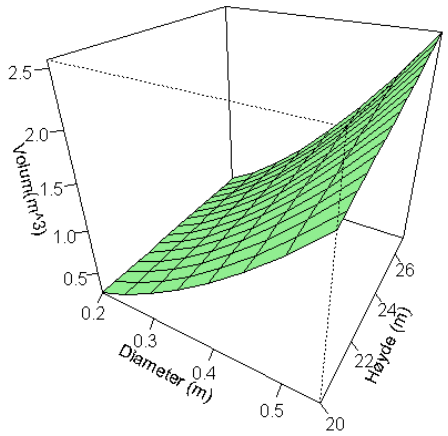
Som viser normalfordeling av residualene (p>0.05)

Vi kan også forsøke en selvstartende modell **SSlogis**:

```
logis <- nls(VOLUME ~ SSlogis(DIAMET, Asym, xmid, scal))
summary(logis)
predikt <- predict(logis)
plot(VOLUME ~ DIAMET, col = 4,xlab =  "Diameter (m)",
     ylim = c(0,5), ylab = "Volum (m^3)")
lines(DIAMET, predikt, col = 2, lwd = 3)
abline(h = coef(logis)["Asym"], col = 3, lty = 2, lwd = 3)
Formula: VOLUME  ~ SSlogis(DIAMET, Asym, xmid, scal)

Parameters:
     Estimate Std. Error t value Pr(>|t|)
Asym  4.99617    2.12330   2.353   0.0259 *
xmid  0.55847    0.09580   5.830 2.90e-06 ***
scal  0.13106    0.01759   7.452 4.08e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0944 on 28 degrees of freedom

Number of iterations to convergence: 3
Achieved convergence tolerance: 5.625e-07
```



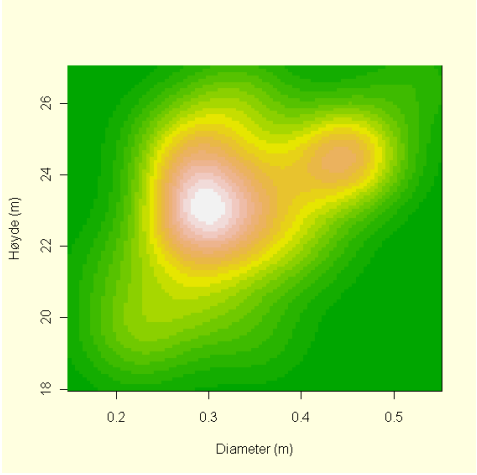

```
#3D av modellen
D <- seq(0.2, 0.55, 0.05) #diameter
H <- seq(20, 27, 0.5) #høyde
f <-  function (D,H)(pi * D^2 * H)/10 #volum
V  <-  outer(D, H, f)
V[is.na(V)] <-  1
op  <-  par(bg =   "lightyellow")
persp(D, H, V,theta =  30,phi = 30,
     col = "lightgreen", ticktype = "detailed",
     xlab = "Diameter (m)", ylab = "Høyde (m)",
     zlab = "Volum(m^3)")
```



Med kommandoen kde2d() i pakken MASS kan man lage et todimensjonalt kjernetetthetsestimat lagt på et rutenett:

```
library(MASS)
?kde2d
tetthet <- kde2d(DIAMET, HEIGHT, n =  100,
```

```
     lims =  c(0.15, 0.55, 18, 27))
image(tetthet, col  = terrain.colors(20),
     xlab = "Diameter (m)", ylab = "Høyde (m)")
detach(oppg2)
```



Oppgave 3 Blokkdesign to-veis ANOVA

To faktorvariable

Eksperiment:

En bonde ønsker å finne ut hvilken sort av bønner (BEAN 1-6) som gir størst avling (YIELD) på jorda han eier. For å unngå effekten av gradienter på forsøksfeltet deles forsøket inn i fire blokker (BLOCK 1-4). Er det noen signifikant forskjell mellom bønnesortene ? Hvilken bønnesort vil du anbefale at bonden benytter ?

Datasett fra: Grafen, A. & Hails, R.: *Modern statistics for the life sciences*. Oxford University Press 2003

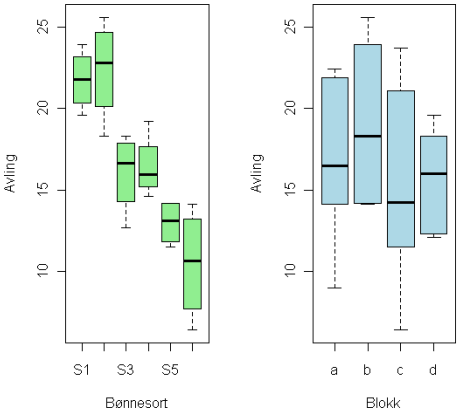
Modell: YIELD ~ BEAN + BLOCK + error

```
oppg3 <- read.table("oppg3.txt",header =  T)
attach(oppg3)#husk detach() ved avslutning
names(oppg3)
[1] "YIELD" "BLOCK" "BEAN"
summary(oppg3)
     YIELD       BLOCK BEAN
 Min.   : 6.40   a:6   S1:4
 1st Qu.:13.75   b:6   S2:4
 Median :16.00   c:6   S3:4
 Mean   :16.68   d:6   S4:4
 3rd Qu.:19.98         S5:4
 Max.   :25.60         S6:4
dim(oppg3)
[1] 24  3
oppg3[1:10,]
#Hele
oppg3
   YIELD BLOCK BEAN
1    9.0     a   S6
2   14.6     a   S4
3   18.3     a   S3
4   14.1     a   S5
5   21.9     a   S2
```

```
6    22.4      a    S1
7    14.2      b    S5
8    14.1      b    S6
9    17.4      b    S3
10   25.6      b    S2
```

Plotting av datasettet:

```
par(mfrow = c(1,2))
plot(BEAN, YIELD, col = "lightgreen", xlab =  "Bønnesort",
     ylab = "Avling")
plot(BLOCK, YIELD,col = "lightblue" , xlab =  "Blokk",
     ylab = "Avling")
```




```
#ANOVA-tabell
summary(aov(YIELD ~ BEAN+BLOCK))
summary(aov(YIELD ~ BEAN+BLOCK))
            Df Sum Sq Mean Sq F value    Pr(>F)
BEAN         5 444.43   88.89 23.4757 1.341e-06 ***
BLOCK        3  52.89   17.63  4.6567   0.01713 *
Residuals   15  56.79    3.
#Lineær modell
model <- lm(YIELD ~ BEAN+BLOCK)
summary(model)
Call:
lm(formula = YIELD ~ BEAN + BLOCK)

Residuals:
    Min      1Q  Median      3Q     Max
-3.1167 -1.1729  0.2333  0.8375  2.8083

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   21.792      1.192  18.288 1.15e-11 ***
BEANS2         0.625      1.376   0.454  0.65616
BEANS3        -5.675      1.376  -4.125  0.00090 ***
BEANS4        -5.325      1.376  -3.870  0.00151 **
BEANS5        -8.775      1.376  -6.378 1.24e-05 ***
BEANS6       -11.300      1.376  -8.213 6.22e-07 ***
BLOCKb         2.350      1.123   2.092  0.05388 .
BLOCKc        -1.517      1.123  -1.350  0.19703
BLOCKd        -1.000      1.123  -0.890  0.38745
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.946 on 15 degrees of freedom
Multiple R-Squared: 0.8975,     Adjusted R-squared: 0.8428
F-statistic: 16.42 on 8 and 15 DF,  p-value: 4.047e-06
```

```
model2 <- lm(YIELD ~ BEAN)
summary(model2)

Call:
lm(formula = YIELD ~ BEAN)

Residuals:
   Min     1Q Median     3Q    Max
-4.075 -1.456 -0.250  1.456  3.650

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   21.750      1.234  17.621 8.47e-13 ***
BEANS2         0.625      1.746   0.358  0.72447
BEANS3        -5.675      1.746  -3.251  0.00443 **
BEANS4        -5.325      1.746  -3.051  0.00688 **
BEANS5        -8.775      1.746  -5.027 8.76e-05 ***
BEANS6       -11.300      1.746  -6.474 4.34e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.469 on 18 degrees of freedom
Multiple R-Squared: 0.802,      Adjusted R-squared: 0.7471
F-statistic: 14.59 on 5 and 18 DF,  p-value: 8.579e-06
```

Hvis man sammenligner dette med middeltallene ser man hvordan estimatene i modellen kommer fram:

```
tapply(YIELD, BEAN, mean)
    S1     S2     S3     S4     S5     S6
21.750 22.375 16.075 16.425 12.975 10.450
```

```
anova(model, model2, test = "F")
Analysis of Variance Table

Model 1: YIELD ~ BEAN + BLOCK
Model 2: YIELD ~ BEAN
  Res.Df     RSS Df Sum of Sq      F  Pr(>F)
1     15  56.795
2     18 109.690 -3   -52.895 4.6567 0.01713 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Modell:

$$YIELD \sim 21.792 + \underbrace{\begin{matrix}0.000 \\ 0.625 \\ -5.675 \\ -5.325 \\ -8.775 \\ -11.300\end{matrix}}_{BEANS} \begin{matrix}S1 \\ S2 \\ S3 \\ S4 \\ S5 \\ S6\end{matrix} + \underbrace{\begin{matrix}0.000 \\ 2.350 \\ -1.517 \\ -1.000\end{matrix}}_{BLOCK} \begin{matrix}a \\ b \\ c \\ d\end{matrix} + error$$

BLOCKa og BEAN S1 er brukt som referanse (settes alfabetisk) og settes lik 0.

```
#Kontraster
contrasts(BEAN)
    S2 S3 S4 S5 S6
S1   0  0  0  0  0
S2   1  0  0  0  0
S3   0  1  0  0  0
S4   0  0  1  0  0
S5   0  0  0  1  0
S6   0  0  0  0  1
contrasts(BLOCK)
  b c d
a 0 0 0
b 1 0 0
c 0 1 0
d 0 0 1
detach(oppg3)
```

En lineær modell betyr ikke nødvendigvis en rettlinjet sammenheng mellom responsvariabel og avhengig variabel. Noen modeller er ikke-linære og kan bli linearisert ved transformering. Fikserte faktorer (”Fixed factors”) er vanligvis krysset, dvs. alle kombinasjoner er representert, mens randome faktorer (”random factors”) er ofte nestet.
Se: Splitplotdesign og ANCOVA.

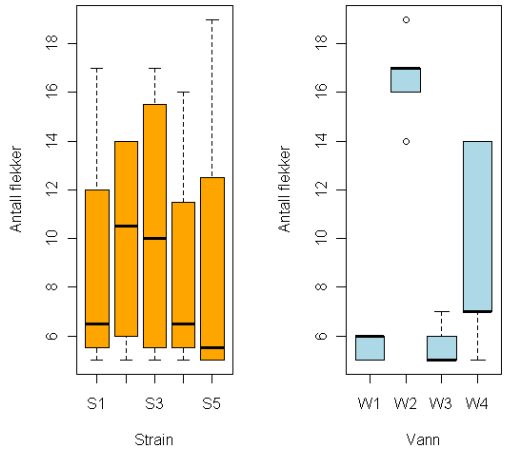
Oppgave 4 Tellinger og Poisson-fordeling

Eksperiment: Du skal utføre et feltforsøk for å teste hvordan forskjellige stammer av bygg (*Hordeum vulgare*) er resistent mot meldugg (*Erysiphe graminis*) under forskjellige vannforhold. Det er fem stammer (STRAIN) bygg som sammenlignes under fire forskjellige vanningsregimer (WATER). For hver behandling plukkes det tilfeldig ut 6 blader hvor antall flekker med soppinfeksjon (SPOTS) telles. Er det signifikant forskjell mellom stammene når det gjelder resistens (toleranse) for soppsykdommen meldugg ? WATER og STRAIN er kategoriske variable. Når det gjelder telling av hendelser eller objekter som skjer tilfeldig i tid eller rom bør man straks tenke på Poisson-fordeling.
Datasett fra: Grafen, A. & Hails, R.: *Modern statistics for the life sciences*. Oxford University Press 2003

Modell: SPOTS ~ WATER + STRAIN

```
oppg4 <- read.table("oppg4.txt",header = T)
attach(oppg4)#husk detach()
names(oppg4)
[1] "WATER"  "STRAIN" "SPOTS"
summary(oppg4)
WATER  STRAIN     SPOTS
 W1:5   S1:4   Min.   : 5.0
 W2:5   S2:4   1st Qu.: 5.0
 W3:5   S3:4   Median : 6.5
 W4:5   S4:4   Mean   : 9.3
        S5:4   3rd Qu.:14.0
               Max.   :19.0
oppg4[1:6,]
  WATER STRAIN SPOTS
1    W1     S1     6
```

```
2     W1      S2      5
3     W1      S3      5
4     W1      S4      6
5     W1      S5      6
6     W2      S1     17
par(mfrow = c(1,2))
plot(STRAIN, SPOTS, col = "orange",xlab =  "Strain",
     ylab = "Antall flekker")
plot(WATER,SPOTS,col = "lightblue", xlab = "Vann",
     ylab =  "Antall flekker")
```




```
#ANOVA-tabell
summary(aov(SPOTS ~ WATER+STRAIN))
            Df Sum Sq Mean Sq F value    Pr(>F)
WATER        3 403.40  134.47 20.6607 4.955e-05 ***
STRAIN       4  12.70    3.18  0.4878    0.7448
Residuals   12  78.10    6.51
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Vi lager en matrise med navn flekker og setter inn verdiene for spots:

```
flekker <- matrix(c(6,17,5,7,5,14,7,14,5,17,6,14,6,16,5,7,6,19,5,5),
     nrow =  4)
flekker
     [,1] [,2] [,3] [,4] [,5]
[1,]    6    5    5    6    6
[2,]   17   14   17   16   19
[3,]    5    7    6    5    5
[4,]    7   14   14    7    5

#Kjikvadrattest
chisq.test(flekker)

        Pearson's Chi-squared test

data:  flekker
X-squared  =   7.8323, df  =   12, p-value  =   0.7981

#kritiske verdi
```

```
qchisq(0.95, 1)
[1] 3.841459
```

Appendiks oppgave 4

F.eks er en 2x2 kontingenstabell er av følgende type:

	Kolonne 1	Kolonne 2	Rad total
Rad 1	a	b	a+b
Rad 2	c	d	c+d
Kolonne total	a+c	b+d	n

Generelt for kontingenstabeller med r antall rader og k antall kolonner har antall frihetsgrader:

$$df = (r-1)\cdot(k-1)$$

For en 2x2 kontingenstabell blir df = 1.

Sannsynligheten for et gitt utkomme er:

$$p = \frac{(a+b)!\,(c+d)!\,(a+c)!\,(b+d)!}{a!\,b!\,c!\,d!\,n!}$$

Vi kan vurdere signifikansen mellom observert frekvens (O) fra forventet frekvens (E) med tre typer tester: Pearsons kjikvadrat (χ^2), G-test eller Fisher´s eksakttest.

Pearsons χ^2
Testobservator χ^2 er gitt ved:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

Vi ser på den tilsvarende kontingenstabellen fra denne oppgaven:

	S1	S2	S3	S4	S5	Total
W1	6	5	5	6	6	28
W2	17	14	17	16	19	83
W3	5	7	6	5	5	28
W4	7	14	14	7	5	47
	35	40	42	34	35	186

Kontingenstabellen for melduggresistens i bygg

S1	S2	S3	S4	S5	Tot
28/186*35/186	28/186*40/186	28/186*42/186	28/186*34/186	28/186*35/186	28
83/186*35/186	83/186*40/186	83/186*42/186	83/186*4/186	83/186*35/186	83
28/186*35/186	28/186*40/186	28/186*42/186	28/186*4/186	28/186*35/186	28
47/186*35/186	47/186*40/186	47/186*42/186	47/186*4/186	47/186*35/186	47
35	40	42	34	35	186

Sannsynligheten for kombinasjonene

	S1	S2	S3	S4	S5	Total
W1	5.2688	6.0215	6.3226	5.1182	5.2688	28
W2	15.6182	17.8494	18.7419	15.1720	15.6182	83
W3	5.2688	6.0215	6.3226	5.1183	5.2688	28
W4	8.8441	10.1075	10.6129	8.5914	8.8441	47
	35	40	42	34	35	186

Forventet frekvens. Tallene i tabellen over *186

Generaliserte modeller (GLM)

I GLM kan man spesifisere forskjellige error-strukturer for eksempel **family = poisson** eller family = binomial. Linkfunksjone relaterer middelverdien til responsvariabel til dens linære prediktor. Hvis man spesifiserer family settes følgende linkfunksjoner automatisk:

Error	Linkfunksjon
Normalfordelt	identitet
Poissonfordelt	log
Binomialfordelt	logit
Gammafordelt	resiprok

I **generaliserte additive modeller** (gam) bruker man glattingsfunksjon s(), som egentlig er ”splines”.

Modell med interaksjon:

```
model <- glm(SPOTS ~ WATER * STRAIN, poisson)
summary(model)
```

Modell uten interaksjon:

```
model2 <- glm(SPOTS ~ WATER + STRAIN, poisson)
summary(model2)
model3 <- glm(SPOTS ~ WATER, poisson)
summary(model3)
```

```
Call:
glm(formula = SPOTS ~ WATER, family = poisson)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.57711  -0.25829  -0.02519   0.16708   1.39776

Coefficients:
             Estimate Std. Error   z value Pr(>|z|)
(Intercept)  1.723e+00  1.890e-01     9.116  < 2e-16 ***
WATERW2      1.087e+00  2.185e-01     4.972 6.62e-07 ***
WATERW3     -1.233e-16  2.673e-01 -4.62e-16   1.0000
WATERW4      5.179e-01  2.387e-01     2.170   0.0300 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 49.6844  on 19  degrees of freedom
Residual deviance:  9.3124  on 16  degrees of freedom
```

```
AIC: 96.489

Number of Fisher Scoring iterations: 4
```

Test om model2 er signifikant forskjellig fra model3:

```
anova(model2, model3, test = "Chi")
Analysis of Deviance Table

Model 1: SPOTS  ~  WATER + WATER:STRAIN
Model 2: SPOTS  ~  WATER
  Resid. Df Resid. Dev  Df Deviance P(>|Chi|)
1         0 -5.329e-15
2        16     9.3124 -16  -9.3124    0.9000
```

```
par(mfrow = c(2, 2))
plot(model3)
```

```
#ANOVA-tabell
model <- aov(SPOTS ~ WATER + STRAIN)
summary(model)
            Df Sum Sq Mean Sq F value    Pr(>F)
WATER        3 403.40  134.47 20.6607 4.955e-05 ***
STRAIN       4  12.70    3.18  0.4878    0.7448
Residuals   12  78.10    6.51
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Tar kvadratroten av SPOTS:

```
SQRTSP <- c(sqrt(SPOTS))
model4 <- glm(SQRTSP ~ WATER + STRAIN)
summary(model4)
par(mfrow = c(2,2))
plot(model4)
```

Vi kan prediktere forventede verdier. Funksjonen type = ”response” gjør at man ikke behøver å tilbaketransformere.

```
yv <- predict(model2, type = "response")
yv; detach(oppg4)
       1        2        3        4        5        6        7        8
 5.268817  6.021505  6.322581  5.118280  5.268817 15.618280 17.849462 18.741935
       9       10       11       12       13       14       15       16
15.172043 15.618280  5.268817  6.021505  6.322581  5.118280  5.268817  8.844086
      17       18       19       20
10.107527 10.612903  8.591398  8.844086
```

Dette ser man er omtrent tallene i tabellen vist foran.

Oppgave 5: Ikke-lineær regresjon

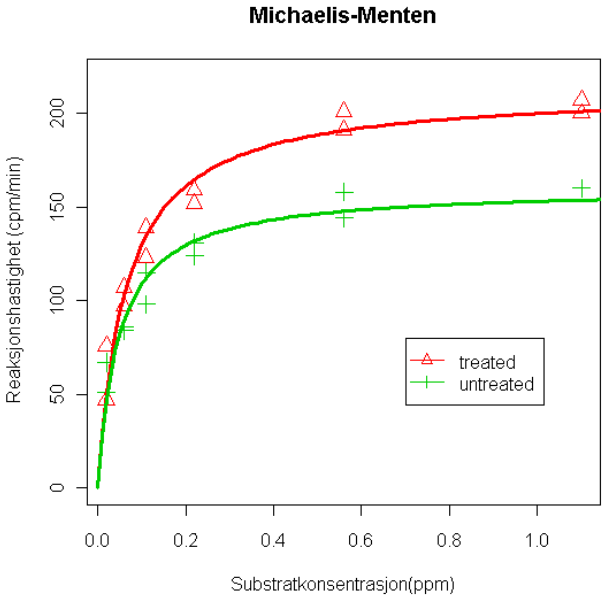
Datasettet Puromycin fra R inneholder reaksjonshastighet (rate, cpm/min) for en enzymreaksjon versus substratkonsentrasjon (conc, ppm) i celler med eller uten behandling med puromycin (state, treated/untreated). Vm er maksimal reaksjonshastighet, Km er substratkonsentrasjonen ved halvparten av maksimal reaksjonshastighet.

Tilpasser en Michaelis-Menten modell med parameterverdier K_m (reaksjonsraten ved $V_m/2$)og V_m (maksimal reaksjonshastighet)

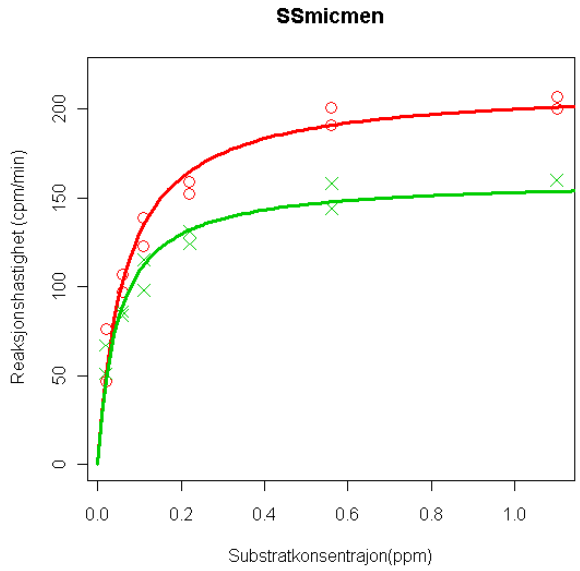
$$f(x) = \frac{V_m x}{K_m + x}$$

hvor y = f(x) er reaksjonsraten. Startverdiene finner man best ved øyemål ved å se på en grafisk fremstilling av resultatet.

```
library(datasets)
data(Puromycin)
attach(Puromycin)
names(Puromycin)
?Puromycin
#Michaelis-Menten modell
#ikke-lineær modell mm1 for "treated"
mm1 <- nls(rate ~ Vm * conc/(Km + conc),
     data =  Puromycin,
subset = state ==  "treated", start = c(Vm = 200,Km = 0.05))
#ikke-lineær modell mm2 for "untreated"
mm2 <- nls(rate ~ Vm * conc/(Km + conc),
data = Puromycin, subset = state == "untreated",
     start = c(Vm = 160, Km = 0.05))
plot(conc[state == "treated"], rate[state == "treated"],
     ylim =  c(0,220), pch = 2, col = 2, cex = 1.5,
     xlab = "Substratkonsentrasjon(ppm)",
     ylab = " Reaksjonshastighet (cpm/min)",
     main =  "Michaelis-Menten")
points(conc[state == "untreated"],
     rate[state == "untreated"],col = 3,
     pch = 3,cex = 1.5)
#plotter predikterte linjer ifølge modellene
x2 <- seq(0, 1.2, 0.01)
ym <- predict(mm1, list(conc = x2))
     lines(x2, ym, col = 2, lwd = 3)
ym2 <- predict(mm2, list(conc = x2))
lines(x2, ym2, col = 3, lwd = 3)
legend(0.7, 80, levels(Puromycin$state),
     col = 2:3,lty = 1, pch = 2:3)
```




```
#eller via den selvstyrende funksjonen SSmicmen
#ikke-lineær regresjon selvstartende funksjon
plot(conc[state == "treated"],
     rate[state == "treated"], ylim = c(0, 220),
     col = 2, cex = 1.5,
     xlab = "Substratkonsentrajon(ppm)",
     ylab = " Reaksjonshastighet (cpm/min)",
     main = "SSmicmen")
#modell1
mod1 <- nls(rate ~  SSmicmen(conc,Vm,K),
     data =  Puromycin,
     subset =  state == "treated")
summary(mod1)
x2 <- seq(0, 1.2, 0.01)
y3 <- predict(mod1, list(conc = x2))
lines(x2, y3, col = 2,lwd = 3)
points(conc[state == "untreated"],
     rate[state == "untreated"],
     col = 3, pch = 4, cex = 1.5)
#modell2
mod2 <- nls(rate ~ SSmicmen(conc,Vm,K),data = Puromycin,
     subset = state == "untreated")
summary(mod2)
y4 <- predict(mod2,list(conc = x2))
lines(x2, y4, col = 3, lwd = 3)
```

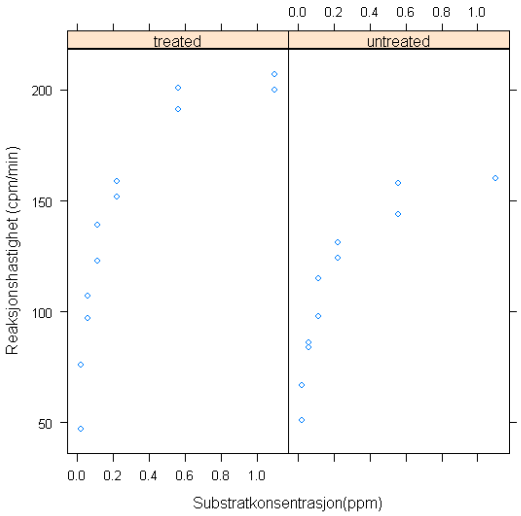



```
Formula: rate  ~  SSmicmen(conc, Vm, K)

Parameters:
    Estimate Std. Error t value Pr(>|t|)
Vm 1.603e+02  6.480e+00  24.734 1.38e-09 ***
K  4.771e-02  7.782e-03   6.131 0.000173 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.773 on 9 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 3.942e-06
```

Eksempler på andre selvstartende funksjoner:
SSasymp: asymptotisk regresjonsmodell
SSbiexp: bieksponensiell modell
SSfpl: fire parameters logistisk modell
SSfol: første ordens kompartmentmodell
SSgompertz: Gompertz vekstmodell
SSlogis: logistisk regresjon
SSweibull: Weibull vekstmodell
Alternativt xy-plot fra pakken lattice():

```
library(lattice)
xyplot(rate ~ conc|state,
     xlab = "Substratkonsentrasjon(ppm)",
     ylab = " Reaksjonshastighet (cpm/min)",
     data = Puromycin)
library(lattice)
xyplot(rate ~ conc|state,
     xlab = "Substratkonsentrasjon(ppm)",
     ylab = " Reaksjonshastighet (cpm/min)",
     data = Puromycin)
mm3 <- nls(rate ~ Vm[state] * conc/(Km[state] + conc),
     data =  uromycin,start = list(Vm =  c(160,160),
```

```
     Km = c(0.05,0.05)))
summary(mm3)
Formula: rate  ~  Vm[state] * conc/(Km[state] + conc)
Parameters:
     Estimate Std. Error t value Pr(>|t|)
Vm1 2.127e+02  6.608e+00  32.185  < 2e-16 ***
Vm2 1.603e+02  6.896e+00  23.242 2.04e-15 ***
Km1 6.412e-02  7.877e-03   8.141 1.29e-07 ***
Km2 4.771e-02  8.281e-03   5.761 1.50e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.4 on 19 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 3.015e-06
```



Ved oppsummering av modellen finner vi modellformel og estimerte parameterverdier for K_m og V_m med tilhørende standardfeil. t-verdien angir resultatet fra nullhypotesen om at parameterverdiene er lik 0. Den estimerte residualstandardfeilen med her 19 frihetsgrader, samt antall iterasjoner før konvergering er angitt.

Ved lineær regresjon finner man den best tilpassete linjen ved minste kvadraters metode, det vil si at man minimaliserer de kvadrerte avvik fra den estimerte linjen. Hvis vi har n prediktorverdier X, responsverdier Y, og en funksjon f som avhenger av parameterverdier β, så er residualkvadratsummen RSS lik:

$$RSS(\beta) = \sum_{i=1}^{n} \left(y_i - f(x_i, \beta)\right)^2$$

Ved ikke-lineær regresjon så vil f være ikke-lineær og vi må bruke numeriske optimaliseringsmetoder i stedet.Gauss-Newton algoritmen for ikke-lineær regresjon brukes trinnvis i en iterasjonsprosess ut fra startparameterverdier. Når man har estimert verdien(e) for β, beta-hatt, så vil varians σ^2 med df frihetsgrader bli:

$$\sigma^2 = \frac{RSS(\hat{\beta})}{df}$$

Vi finner minimumsverdien for RSS via kommandoen deviance(). Man kan også finne logeritmen til likelihood-funksjonen. Likelihoodfunksjonen L er lik:

$$L(\beta, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} e^{\left(-\frac{RSS(\beta)}{2\sigma^2}\right)}$$

Og maksimumverdien av likelihoodfunksjonen er lik:

$$L(\hat{\beta}, \hat{\sigma}^2) = \frac{1}{\left(2\pi \frac{RSS(\hat{\beta})}{n}\right)^{\frac{n}{2}}} e^{\left(-\frac{n}{2}\right)}$$

De estimerte standardfeilene til de estimerte parameterverdiene K_m og V_m er lik kvadratroten til diagonalen i den estimerte varianse-kovarianse-matrisen.

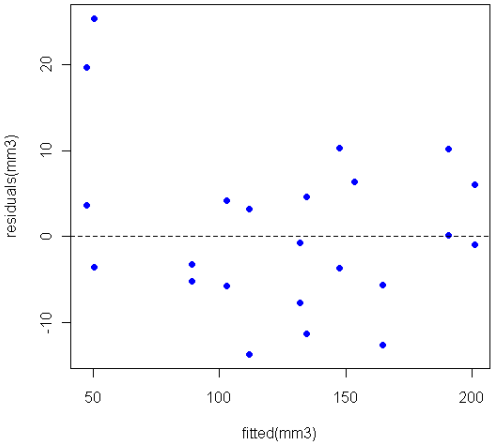
$$\widehat{Var}(\hat{\beta}) = s^2 M$$

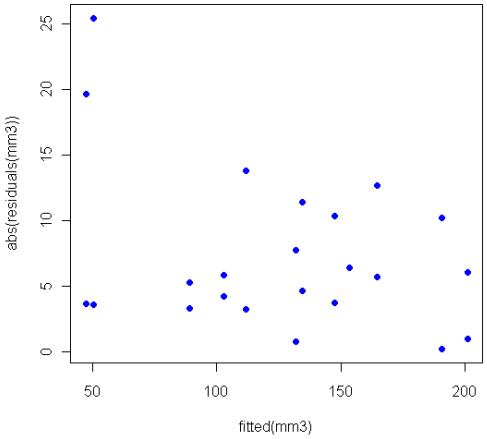
hvor M er den inverste matrisen til de andrederiverte av loglikelihoodfunksjonen (Hesse-matrisen).
I dette eksemplet blir de funksjonen RSS lik:

$$RSS(K_m, V_m) = \sum_{i=1}^{13} \left(TOTAL_i - \frac{V_m \cdot conc}{K_m + conc}\right)^2$$

Som vanlig må man sjekke forutsetningene homogen varianse (homoskedastisitet) , normalfordelte residualer, uavhengige målinger, og bruk av riktig funksjon f.
Sjekking av homogen varianse med Levenes test eller Bartletts. Man kan også se på et plot av tilpassete verdier vs. absoluttverdien av residualene.

```
confint(mm3) #95% konfidensintervall modell
Waiting for profiling to be done...
            2.5%        97.5%
Vm1 198.88882755 227.45563194
Vm2 145.85284482 176.28020956
Km1   0.04856093   0.08354202
Km2   0.03158687   0.06965928
coef(mm3) #parameterverdier modell
         Vm1          Vm2          Km1          Km2
212.68376387 160.28012513   0.06412131   0.04770831
deviance(mm3)#minimum RSS(estimerte parametre)
[1] 2055.053
logLik(mm3) #loglikelihood
'log Lik.' -84.30006 (df =  5)
#residualplot
plot(fitted(mm3), residuals(mm3), col = 4, pch = 16)
abline(a = 0, b = 0, lty = 2)
#tilpasset verdier vs. residualer
plot(fitted(mm3), abs(residuals(mm3)), col = 4,pch = 16)
```



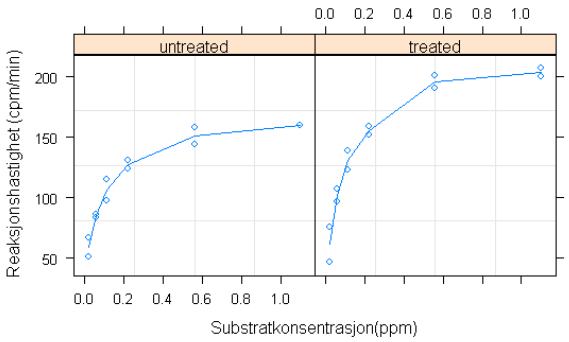



```
#Levenes test homogen varianse
#levene.test eller LeveneTest
library(car)
with(Puromycin,leveneTest(rate, as.factor(conc)))
Levene's Test for Homogeneity of Variance (center  =   median)
      Df F value Pr(>F)
group  5  1.0381 0.4272
      17
```

Vi forkaster ikke hypotesen om at varians er homogen.

En annen mulighet er å lage et groupedData objekt, fra pakken nlme. Man kan derved lage et plot med en enkel kommando.

```
library(nlme)
g <- groupedData(rate ~ conc|state,data = Puromycin)
plot(g, xlab = "Substratkonsentrasjon(ppm)",
     ylab = " Reaksjonshastighet (cpm/min)")
```




```
mm4 <- nlsList(rate ~ Vm * conc/(Km + conc),
     data =  g, start = c(Vm = 200,Km = 0.05))
summary(mm4)
Call:
  Model: rate  ~  Vm * conc/(Km + conc) | state
```

```
   Data: g

Coefficients:
   Vm
          Estimate Std. Error  t value     Pr(>|t|)
untreated 160.2801   6.896016 23.24242 1.384615e-09
treated   212.6836   6.608086 32.18535 3.241147e-11
   Km
            Estimate  Std. Error  t value     Pr(>|t|)
untreated 0.04770823 0.008281162 5.761055 1.727050e-04
treated   0.06412103 0.007876766 8.140527 1.565143e-05

Residual standard error: 10.40003 on 19 degrees of freedom
```

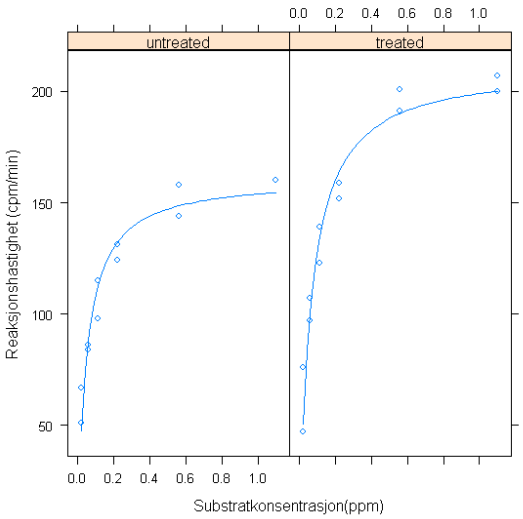
Veien er nå kort over til **mikseteffektmodeller** (blandete modeller) med randome og fikserte faktorer. Mikseteffektmodeller kan håndtere romlige og tidsserie pseudoreplikerte data. Datasettet Puromycin er ikke egnet, vi er ikke ute etter et felles estimat for Km og Vm. Vi antar nå at Km og Vm ikke er uavhengige av hverandre i de to behandlingene, og viser prinsippet for hvordan man går fram. Vi definerer modellen som en mikseteffektmodell:

```
mm5 <- nlme(mm4) #mikset effektmodell
summary(mm5)
```

```
Nonlinear mixed-effects model fit by maximum likelihood
  Model: rate  ~  Vm * conc/(Km + conc)
 Data: g
       AIC      BIC    logLik
  189.3117 196.1247 -88.65586
Random effects:
 Formula: list(Vm  ~  1, Km  ~ 1)
 Level: state
 Structure: General positive-definite, Log-Cholesky parametrization
         StdDev       Corr
Vm       25.687895610 Vm
Km        0.008019748 0.999
Residual  9.892516050

Fixed effects: list(Vm  ~  1, Km  ~ 1)
       Value Std.Error DF  t-value p-value
Vm 186.52527 19.594606 20 9.519215       0
Km   0.05595  0.008218 20 6.808481       0
 Correlation:
   Vm
Km 0.829

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-1.43945721 -0.53927629 -0.01285436  0.52787402  2.57513001

Number of Observations: 23
Number of Groups: 2
```

Med kommandoen augPred kan man nå laget et plot med tilpassete kurver:

```
plot(augPred(mm5),xlab = "Substratkonsentrasjon(ppm)",
     ylab = "Reaksjonshastighet (cpm/min)")
```



Oppgave 6: Tidsserieanalyse av klimadata fra Blindern

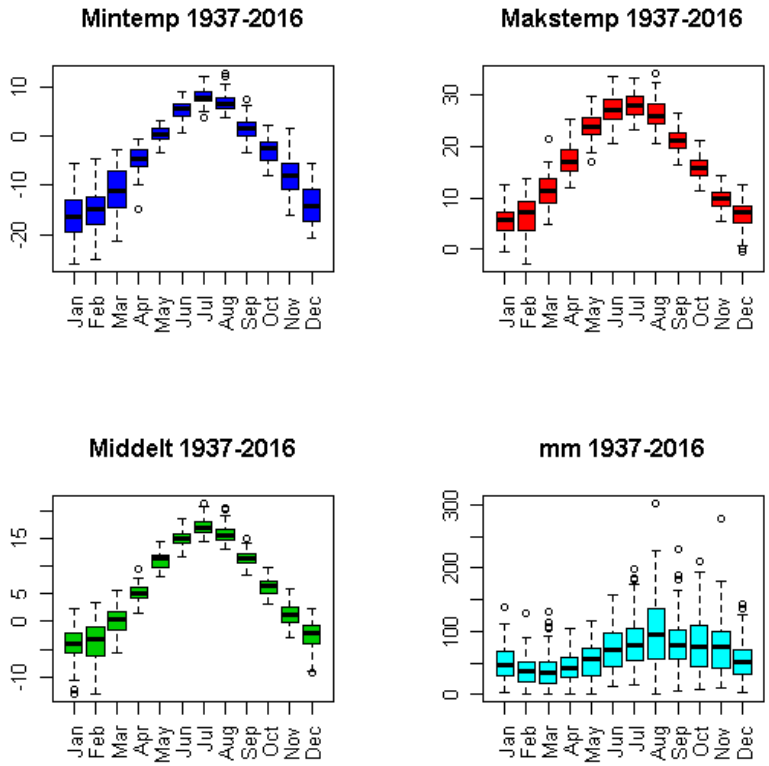
Tidsseriedata er korrelerte og ikke uavhengig av hverandre, de er **autokorrelerte** og er et eksempel på **pseudoreplikasjon**. **Mikset effektmodeller** kan gi korreksjon av pseudoreplikerte data (tid og romlig.

Observasjonsstudium:
I dag mener mange at observerte klimaendringer skyldes bl.a. antropogene utslipp av karbondioksid (CO_2) fra forbrenning av fossilt brensel, metan (CH_4) fra landbruk og petroleumsindustri, nedhogging av CO_2-assimilerende skog, lystgass (N_2O) fra mineralgjødsel, og andre utslipp av drivhusgasser, samt endring i hydrologisk syklus på Jorden. Du har fått i oppgave å gå igjennom klimadata fra Blindern fra 1937-2016 for å undersøke om man kan observere en trend i klima. Datasettet inneholder månedlig minimums-, maksimums- og middeltemperatur samt millimeter nedbør, hentet fra Eklima, Met.inst.

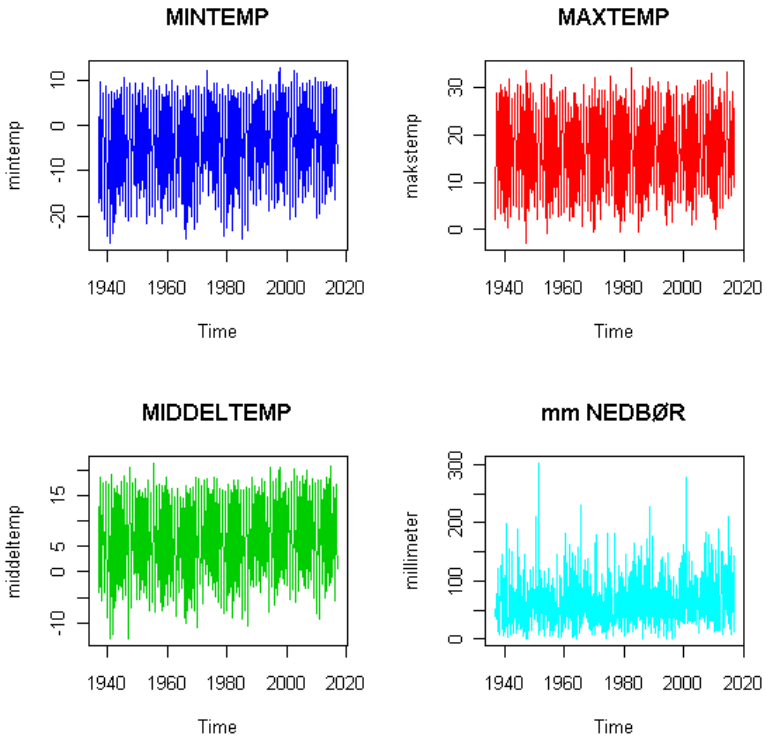
Laster inn datasettet

```
oppg6 <- read.table("oppg6.txt", header = T)
attach(oppg6)
names(oppg6)
 [1] "AAR"  "MINT" "MDLT" "MAXT" "MM"
head(oppg6)
#Boksplot for temperatur og nedbør
mintemp <- ts(MINT,start = c(1937, 1), frequency = 12)
makstemp <- ts(MAXT,start = c(1937, 1), frequency = 12)
middeltemp <- ts(MDLT,start = c(1937, 1), frequency = 12)
millimeter <- ts(MM,start = c(1937, 1), frequency = 12)
par(mfrow= c(2, 2))
boxplot(split(mintemp,cycle(mintemp)), las=3,
     col = 4, names = month.abb,main = "Mintemp 1937-2016")
boxplot(split(makstemp, cycle(makstemp)), col = 2, las=3,
```

```
     names =  month.abb, main =  "Makstemp 1937-2016")
boxplot(split(middeltemp, cycle(middeltemp)), las=3,
     col = 3, names =  month.abb, main =  "Middelt 1937-2016")
boxplot(split(millimeter, cycle(millimeter)), las=3,
     col =  "cyan", names =  month.abb, main =  "mm 1937-2016")
```



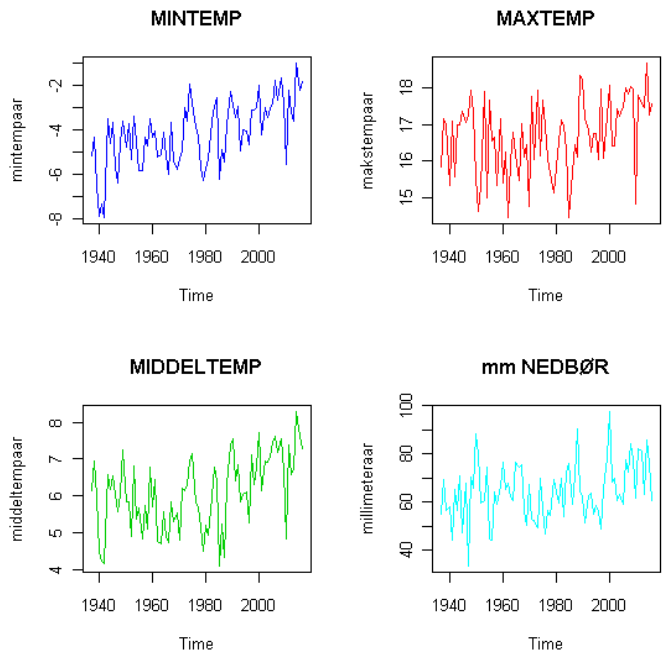

```
par(mfrow = c(2, 2)) #ser uoversiktigelig ut
ts.plot(mintemp, col = 4, main = "MINTEMP")
ts.plot(makstemp, col = 2, main = "MAXTEMP")
ts.plot(middeltemp, col =  3, main = "MIDDELTEMP")
ts.plot(millimeter, col =  5, main = "mm NEDBØR")
```



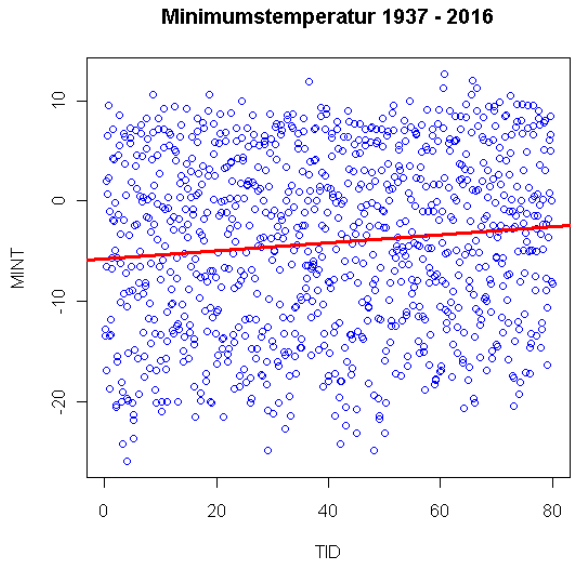

```
summary(oppg6)
```

```
      AAR             MINT             MDLT             MAXT       
 Min.   : 1.194   Min.   :-26.000   Min.   :-13.100   Min.   :-2.90  
 1st Qu.: 3.946   1st Qu.:-11.600   1st Qu.: -0.600   1st Qu.: 9.10  
 Median : 6.698   Median : -3.300   Median :  5.700   Median :16.30  
 Mean   : 6.698   Mean   : -4.174   Mean   :  6.076   Mean   :16.67  
 3rd Qu.: 9.450   3rd Qu.:  3.400   3rd Qu.: 13.100   3rd Qu.:24.40  
 Max.   :12.202   Max.   : 12.700   Max.   : 21.300   Max.   :34.20  
       MM        
 Min.   :  0.00  
 1st Qu.: 34.00  
 Median : 58.00  
 Mean   : 64.56  
 3rd Qu.: 86.00  
 Max.   :303.00  
```

```
#regner årsgjennomsnitt med aggregate
mintempaar <- aggregate(mintemp, FUN = mean)
makstempaar <- aggregate(makstemp, FUN = mean)
middeltempaar <- aggregate(middeltemp, FUN = mean)
millimeteraar <- aggregate(millimeter, FUN = mean)
par(mfrow = c(2, 2))
plot(mintempaar, col =  4, main = "MINTEMP")
plot(makstempaar,col =  2, main = "MAXTEMP")
plot(middeltempaar,col = 3, main =  "MIDDELTEMP")
plot(millimeteraar,col = 5, main =  "mm NEDBØR")
```




```
#Minimumstemperatur
index <- 1:960
TID <- index/12
plot(TID, MINT, col = 4,
     main = "Minimumstemperatur 1937 - 2016")
model <- lm(MINT ~ TID)
abline(model,col = 2,lwd = 3)
```



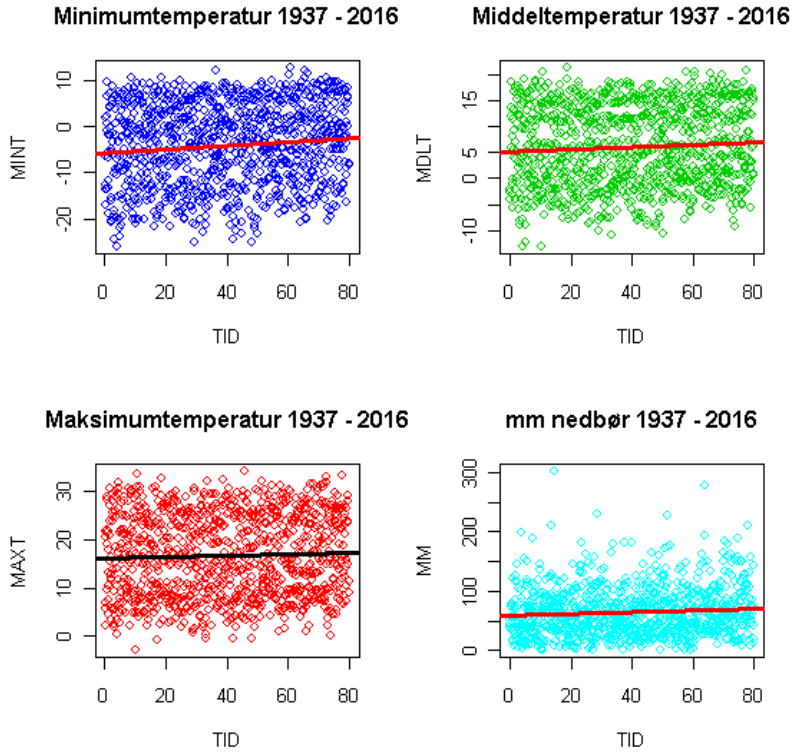

```
summary(model)

Call:
lm(formula = MINT ~ TID)

Residuals:
     Min       1Q   Median       3Q      Max
```

```
-21.0455  -7.4440   0.7777   7.6332  16.2096

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.74704    0.57083 -10.068  < 2e-16 ***
TID          0.03929    0.01235   3.182  0.00151 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.836 on 958 degrees of freedom
Multiple R-squared: 0.01046,    Adjusted R-squared: 0.009423
F-statistic: 10.12 on 1 and 958 DF,  p-value: 0.001511
```

Nullhypotesen forkastes, p=0.0015, det har skjedd en økning i minimumstemperaturen på Blindern i perioden 1937-2016.

Tilsvarende for maksimumstemperatur, minimumstemperatur, og millimeter nedbør:

```
par(mfrow=c(2,2))
plot(TID, MINT, col = 4,
     main = "Minimumtemperatur 1937 - 2016")
model <- lm(MINT ~ TID)
abline(model,col = 2, lwd = 3)

plot(TID, MDLT, col = 3,
     main = "Middeltemperatur 1937 - 2016")
model2 <- lm(MDLT ~ TID)
abline(model2, col = 2, lwd = 3)
summary(model2)
plot(TID, MAXT, col = 2,
     main = "Maksimumtemperatur 1937 - 2016")
model3 <- lm(MAXT ~ TID)
abline(model3,col = 1, lwd = 3)
summary(model3)
plot(TID, MM, col = 5,
     main = "mm nedbør 1937 - 2016")
model4 <- lm(MM ~ TID)
abline(model4,col = 2, lwd = 3)
```



Det har ikke skjedd noen signifikant endring i middeltemperatur (p = 0.056) og maksimumstempertur (p = 0.223) målestasjon Blindern stnr. 18700 i perioden 1937-2015. Nullhypotesen forkastes imidlertid for minimumstemperatur og mm nedbør, det har skjedd en økning i minimumstemperaturen på Blindern i perioden 1937-2016 (p = 0.0023), for nedbør en økning (p = 0.00512). Vi kan støtte hypotesen om høyere minimumstemperatur og mer nedbør, i hvert fall på Blindern.

Autokorrelasjon

Tidsrekkedata er autokorrelerte og gir pseudoreplikasjon. I vanlig regresjon forutsetter man at restleddet (feilleddet)ε_t er normalfordelt og gjensidig uavhengig, hvit støy med lite informasjonsinnhold:

$$Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t$$

I en tidsserie (tidsrekke) er ikke verdiene uavhengig av hverandre, de er **seriekorrelerte** eller **autokorrelerte**, og man kan ikke bruke vanlig regresjonsmodell. Vi kan beskrive korrelasjonen mellom den opprinnelige tidsrekken og en tidsforskjøvet rekke.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
	x_1	x_2	x_3	x_4	x_5	X_6	x_7	x_8

Den øverste rekken fra x_1 til x_t er den opprinnelig, og den under er forskjøvet med en tidsenhet, og man ser på korrelasjonskoeffisientene mellom disse to rekkene, kalt **autokorrelasjon lag 1**. Vi kan bruke et ledd i rekken x_t til å prediktere (forutsi) hva det neste leddet i rekken x_{t+1} vil bli. Hvis vi lager en tidsforskyvning på to ledd, **autokorrelasjon lag 2**, så sier dette noe om hvor god prediksjonen to tidsledd fram. Øverst en opprinnelige observasjonsrekken, nederst er rekken forflyttet to tidstrinn

frem, og på nytt ser vi på korrelasjonen mellom de leddene som står overfor hverandre

x_1	x_2	x_3	x_4	x_5	x_6	x_7		
		x_1	x_2	x_3	x_4	x_5	x_6	x_7

Slik kan man fortsette å forskyve og se på korrelasjonen mellom opprinnelig tidsrekke og forskjøvet tidsrekke, lag 3, lag4, …, hvor måleenheten for lag nummer er måleenheten i tidsrekken.

Hvis korrelasjonskoeffisienten mellom x_t og x_{t+1} er lik r og man antar at korrelasjonskoeffisienten mellom x_{t+2} og x_{t+2} også er lik r så vil korrelasjonen to trinn fram være r^2. Imidlertid, hvis det er avvik mellom de to r i de to tidstrinnene så blir dette beskrevet av **partiell autkorrelasjon**, som angir avviket fra r^2, og gir utfyllende informasjon om strukturen i rekken.

I en rekke med hvit støy kan man ikke fra et ledd i rekken prediktere hva neste ledd vil bli, autokorrelasjonen blir lik 0 for alle lag nummer.

Auroregressiv AR-modell (kursorisk)

Vi kan se på minimumstemperaturen som en autoregressiv AR-modell, AR(8):

```
mintemp <- ts(MINT,start =  c(1937,1),frequency = 12)
#tilpasser autoregressiv modell
mintemp.ar <- ar(aggregate(mintemp,FUN = mean),
     method =  "mle")
mean(aggregate(mintemp, FUN =  mean)) #intercept
 -4.17375
mintemp.ar$order #p-te orden AR(8)
[1] 8
mintemp.ar$ar #parameterverdier alfa
[1]  0.42446483  0.20908044 -0.09325543 -0.24218712  0.23850714  0.19903231
[7] -0.13538633  0.30063601
```

I en autogregresiv modell eller prosess så blir hvert ledd i tidsrekken påvirket av foregående ledd i kjeden og hvit støy (w_t). I en bevegelig gjennomsnitt modell eller prosess (MA) så blir et ledd i kjeden bare påvirket av hvit støy, nåværende og den forrige verdi

Første ordens bevegelig gjennomsnitt MA(1) er gitt ved:

$$x_t = w_t - \beta_1 w_{t-1}$$

hvor β_1 kan være positiv eller negativ.

Hvis vi ser på neste ledd i rekken blir dette:

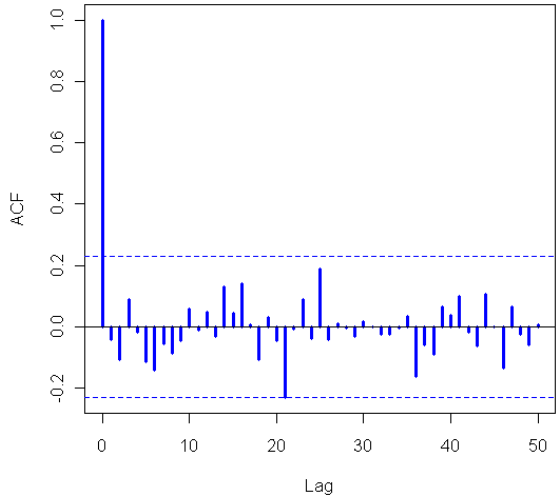
$$x_{t+1} = w_{t+1} - \beta_1 w_t$$

og det er et felles hvit støy wt i de to trinnene som kan endre fortegn avhengig av fortegnet på β_1, dvs. det er autokorrelasjon lag 1 og retningen blir avhengig av

fortegnet på β_1. Kommer man til tredje ledd i rekken x_{t+2} så har dette ingen felles ledd med startverdien, altså autokorrelasjon lag 2 blir lik 0.

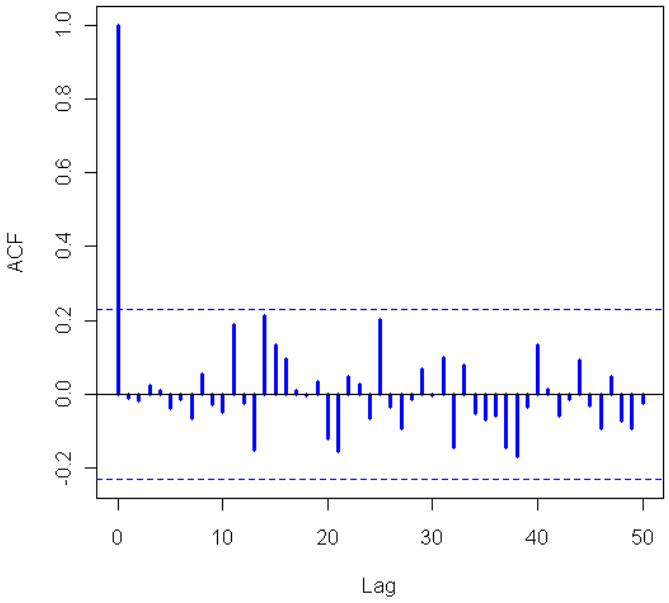
$$x_{t+2} = w_{t+2} - \beta_1 w_{t+1}$$

```
acf(mintemp.ar$res[-(1:mintemp.ar$order)],
     lag = 50, col =  4, lwd=3,  main =  "")
```



Et korrelogram som viser residualene for en AR(8) modell tilpasset minimumstemperaturen på Blindern. Korrelogrammet indikerer hvit støy, det vil si det er en stokastisk modell som forklarer minimumstemperaturen.
Tilsvarende for middeltemperaturen AR(6) hvor det er indikasjoner på at det ikke er en ren stokastisk hvit støy

```
middeltemp <- ts(MDLT,start =  c(1937,1),frequency =  12)
middeltemp.ar <- ar(aggregate(middeltemp,FUN =  mean),method =
"mle")
mean(aggregate(middeltemp,FUN =  mean)) #intercept
6.075938
middeltemp.ar$order #p-te orden
[1] 6
middeltemp.ar$ar #parameterverdier alfa
1]  0.37152374  0.25809385 -0.09654315 -0.16066068  0.05329475  0.30801947
acf(middeltemp.ar$res[-(1:mintemp.ar$order)],lag =  50,
     col =  4, lwd=3, main =  "")
```



Det vil si at prediktert gjennomsnittstemperatur blir

$$\hat{x}_t = 6 + 0.6076(x_{t-1} - 6) + \varepsilon_t$$

hvor ε er et error-ledd. Vi ser at storgjennomsnittet for middeltemperaturen er 6°C.

```
mean(MDLT)
```
```
6.075938
```

Mange tidsserier har en **trend** (m_t), en **sesongeffekt** (s_t) og et resterende feilledd ε_t. Det vil si for hver observasjon i tidsserien x_t kan vi ha en additiv modell

$$x_t = m_t + s_t + \varepsilon_t$$

Det er også andre mulige modeller som:

$$\log x_t = m_t + s_t + \varepsilon_t$$

eller en multiplikativ modell:

$$x_t = m_t \cdot s_t + \varepsilon_t$$

Feileddet er korrelert og bør være normalfordelt med gjennomsnittsverdi ca. 0.

Trenden kan beregnes via glattingsfunksjon som et **bevegelig gjennomsnitt** omkring et angitt antall tidsserieverdier, unntatt de første og siste.

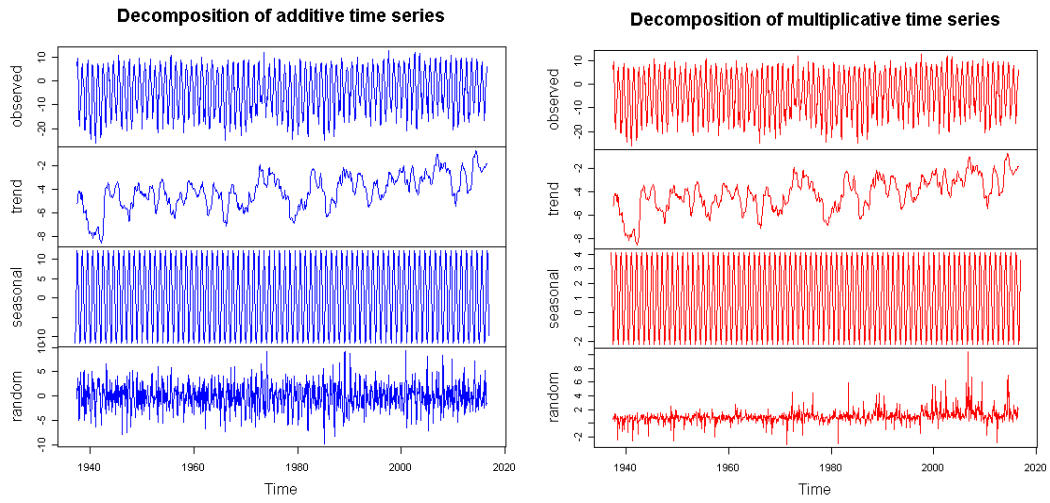
Noen ganger er det behov for å sesongjustere tidsserien.

Sinus og cosinus kan brukes til å lage en glattingsfunksjon for modeller som inneholder en sesongeffekt eller årstidsvariasjon:

$$y = a + b\sin(2\pi t) + c\cos(2\pi t) + \varepsilon$$

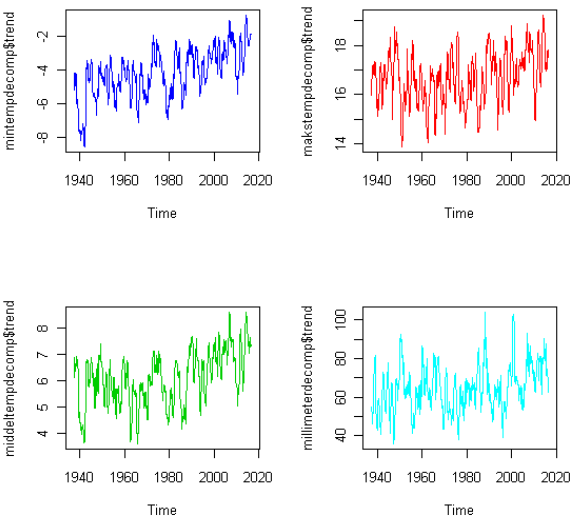
Vi kan dele opp tidsserien i trend, sesongvariasjon og feil:

```
#minimumstemperatur additiv tidsserie
plot(decompose(mintemp), col =  4)
#minimumstemperatur mulitplikativ tidsserie
plot(decompose(mintemp, type =  "mult"), col =  2)
```

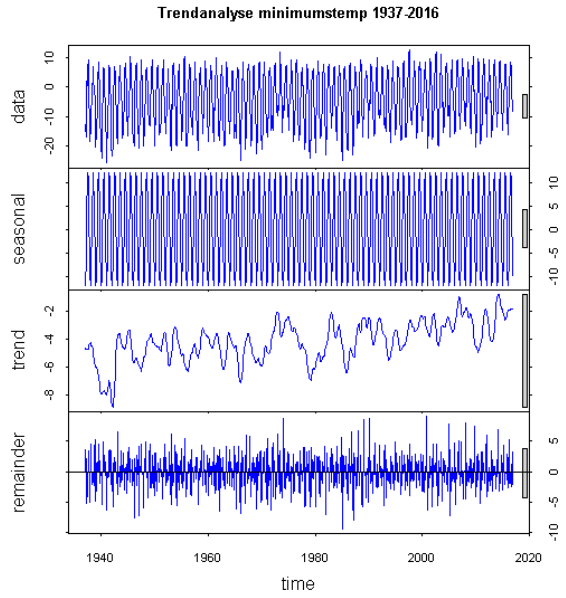



```
#alternativt
#minimumstemperatur fordelt på trend og sesong
mintempdecomp <- decompose(mintemp)
plot(mintempdecomp,col = 4)
makstempdecomp <- decompose(makstemp)
middeltempdecomp <- decompose(middeltemp)
millimeterdecomp <- decompose(millimeter)
par(mfrow = c(2,2))
plot(mintempdecomp$trend,col = 4)
plot(makstempdecomp$trend,col = 2)
plot(middeltempdecomp$trend,col = 3)
plot(millimeterdecomp$trend,col = 5)
```

Trendene for minimums-, maksimums- og middeltemperatur samt nedbør i tidsperioden:



```
#Trendanalyse
minimum <- stl(mintemp,"period")
plot(minimum,col = 4,
     main = "Trendanalyse minimumstemp 1937-2015")
```



Forventningsverdien (forventning) E(X) er lik gjennomsnitt for hele populasjonen (μ), dvs. det teoretiske gjennomsnittet for alle objektene i populasjonen.
Vi kan fra vår prøve lage et estimat av av populasjonsgjennomsnittet:

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

Vi bruker greske bokstaver for å angi populasjonsverdiene, de vi aldri finner, men som vi estimerer ut fra prøven vi har.
Gjennomsnittet av de kvadrerte avvik blir:

$$Var(X) = E(x-\mu)^2$$

som er lik varians for populasjonen, Var(X), angitt som σ^2 for populasjonen og s^2 for prøven.
Varians for prøven:

$$Var(X) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}$$

Standardavviket (sd) er lik kvadratroten av varians, lik σ for populasjonen og s for prøven.
Hvis vi har variable i par (x,y) så er kovarians γ(x,y)for populasjoen lik:

$$\gamma(x,y) = E\left[(x-\mu_x)(y-\mu_y)\right]$$

Kovarians er en lineær assosiasjon mellom variablene x og y.
Ut fra vår prøve får vi et estimat av kovarians:

$$Cov(x,y) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{n-1}$$

hvor $(\bar{x},\bar{y})$ er gjennomsnittsverdiene for prøven.

Korrelasjonen for populasjonen, ρ (rho), og for et variabelpar ρ(x,y) blir:

$$\rho(x,y) = \frac{E\left[(x-\mu_x)(y-\mu_y)\right]}{\sigma_x\sigma_y} = \frac{\gamma(x,y)}{\sigma_x\sigma_y}$$

For prøven blir korrelasjonen Cor(x,y):

$$Cor(x,y) = \frac{Cov(x,y)}{sd(x)sd(y)}$$

Autokorrelasjon (seriell korrelasjon) er korrelasjon for en variabel med seg selv ved forskjellig tid.
For en andreordens stasjonær tidsserie har vi en **autokovariansefunksjon** (acvf), som funksjon av lag k, hvor vi for to variable (x,y) setter x = x_t og y = x_{t+k}, hvor μ er lik gjennomsnittet for både x og y:

$$\gamma_k = E[(x_t - \mu)(x_{t+k} - \mu)]$$

lag k autokorrelasjonsfunksjonen (acf), ρ_k, blir lik:

$$\rho_k = \frac{\gamma_k}{\sigma^2}$$

hvor $\rho_0 = 1$
Fra vår prøve kan vi lage de tilsvarende estimater for populasjonen:
Autokovariansefunksjonen for prøven, ck, blir:

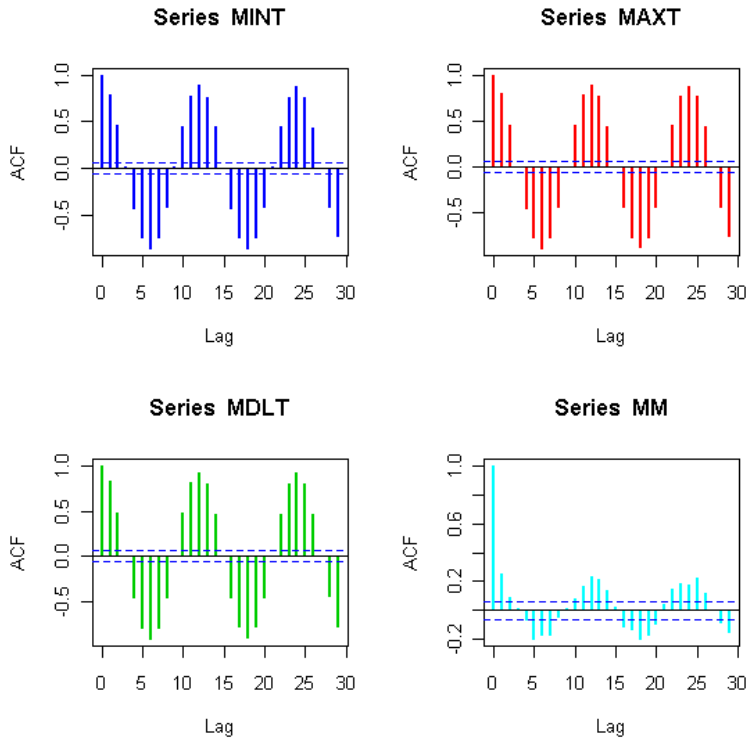
$$c_k = \frac{1}{n}\sum_{t=1}^{n-k}(x_t - \bar{x})(x_{t+k} - \bar{x})$$

Autokorrelasjonen med intervall [-1,1] blir:

$$r_k = \frac{c_k}{c_0}$$

Det betyr at autokorrelasjon ved tid 0 blir lik 1.
lag k får samme måleenhet som tidsintervallene i tidsserien.
Vi får følgende **korrelogram**:

```
#autokorrelasjon
par(mfrow = c(2,2))
acf(MINT, col = 4, lwd = 2)
acf(MAXT, col = 2, lwd = 2)
acf(MDLT, col = 3, lwd = 2)
acf(MM, col = 5, lwd = 2)
```



Autokorrelasjon r_k er på y-aksen og lag k på x-aksen, med enhet som prøveintervallet, her måned i dette eksemplet. Figuren viser årstidsvariasjonen. Ved lag0 er autokorrelasjonen lik 1. Fordelingen er autokorrelasjonen rk er omtrent normalfordelt (når $\rho_k = 0$) med gjennomsnitt -1/n og varianse 1/n. Den stiplete linjen angir:

$$-\frac{1}{n} \pm \frac{2}{\sqrt{n}}$$

Hvis r_k faller utenfor denne stiplete linjen forkastes nullhypotesen om at $\rho_k = 0$ ($\alpha = 0.05$).

Vi kan for eksempel finne autokorrealsjonen for lag1.

```
#lag1 for minimumstempertur
acf(MINT)$acf[2]
[1] 0.7952796
detach(oppg6)
```

Været denne måneden er mer korrelert med forrige måneds vær enn med måneden før der igjen.

Det er tre hovedtyper tidsseriemodeller:
Bevegelig gjennomsnitt modeller(MA, ”mowing average”) som angir gjennomsnitt av et spesifisert antall tidsserieverdier rundt en bestemt tidsserieverdi.
Autoregressive modeller (AR)
Autoregressive bevegelig gjennomsnitt (ARMA).

Gjennomsnittet beregnes ut fra heltall og et estimat for et sentrert bevegelig gjennomsnitt blir:

$$\hat{m}_t = \frac{\frac{1}{2}x_{t-6} + x_{t-5} + x_{t-4} + x_{t-3} + x_{t-2} + x_{t-1} + x_t + x_{t+1} + x_{t+2} + x_{t+3} + x_{t+4} + x_{t+5} + \frac{1}{2}x_{t+6}}{12}$$

En annen type glattingsfunksjon er stl() basert på lokalt vektet regresjonsteknikk (loess).

En tidsserie x_t kan uttrykkes som en **autoregressiv modell** av p-te orden (AR(p)):

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \cdots + \alpha_p x_{t-p} + w_t$$

hvor w_t er hvit støy og α er parameterverdier.
Uttrykt som et p-te grads polynom og tilbakeskiftoperator B i polynomligningen:

$$\left(1 - \alpha_1 B - \alpha_2 B^2 - \cdots - \alpha_p B^p\right) x_t = w_t$$

En første ordens autoregressiv prosess, AR(1) blir

$$x_t = \alpha x_{t-1} + w_t$$

hvor w_t er hvit støy med gjennomsnitt μ = 0 og varianse σ^2.
Det kan vises at autokorrelasjonsfunksjonen γ_k blir:

$$\gamma_k = \frac{\alpha^k \sigma^2}{(1 - \alpha^2)}$$

AR-prosesser kan være stasjonære eller ikke-stasjonære, bestemt ut fra røttene til polynomligningen. Hvis absoluttverdiene til alle røttene i polynomligningen over er større enn 1 så sies prosessen å være stasjonær.

Partiell autokorrelasjon ved lag k er den korrelasjonen vi sitter igjen med etter å ha fjernet korrelasjon i leddene for kortere lag.

Tidsserier kan ha autokorrelerte error og en regresjonstilpasning kan gjøres med GLS (generalisert minste kvadrat) som finnes i pakken nlme.

Litteratur:

Crawley, M.J.: *The R book*, John Wiley & Sons, Ltd. 2007.

Dahlgaard, P.: *Introductory statistics with R*. 2e. Springer 2008.

Grafen, A. & Hails, R.: *Modern statistics for the life sciences*. Oxford University Press 2003.

Whitlock, M.C. & Schluter, D.: *The analysis of biological data*. Roberts and Company publ. 2009.

R Development Core Team (2011). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Oversettelse:

Biased – forventningsskjev

Confounding - konfundering
Density – tetthet
Power - teststyrke
Sample – utvalg
Skewness – skjevhet
Statistic – Observator
Test statistics – testobservator
Unbiased - Forventningsrett

HAa/2015