# SD2 - uge 2, torsdag

*Anne Petersen*

## Opgave 2.A (5.1 i lærebog)

Vi starter med at loade data og kigge lidt på det. Vi bruger desuden `attach()` for lettere adgang til variablene:

```
setwd("C:/Users/zms499/Dropbox/Arbejde/STATforLIFE2/uge2")
org_data <- read.table("organic.txt", header=T)
head(org_data)
```

```
##        TREAT TIME organic
## 1 Ivermectin    8  3028.7
## 2 Ivermectin    8  2805.7
## 3 Ivermectin    8  3061.3
## 4 Ivermectin    8  3113.4
## 5 Ivermectin    8  2938.1
## 6 Ivermectin    8  3063.4
```

```
attach(org_data)
```

Bemærk at datasættet består af tre variable, `organic` (respons/y-variabel/afhængig variabel), `TIME` (forklarende variabel/x-variabel/uafhængig variabel) og `TREAT` (forklarende variabel/x-variabel/uafhængig variabel). Begge forklarende variable er kategoriske og altså skal vi sørge for at R ved, at den skal betragte dem som faktorer. Da `TREAT` er kodet med bogstaver, vil R som udgangspunkt tolke den som en faktor. Men vi må selv gemme `TIME` som en faktor:

```
TIME <- factor(TIME)
```

I Example 3.2 er der foretaget en analyse af data, hvor slutmodellen er en additiv model. Vi fitter altså denne model:

```
model <- lm(organic ~ TREAT + TIME)
```

For god ordens skyld, tester vi også, om der er nogen effekter, der kan fjernes fra denne model:

```
drop1(model, test="F")
```

```
## Single term deletions
##
## Model:
## organic ~ TREAT + TIME
##        Df Sum of Sq     RSS    AIC F value    Pr(>F)
## <none>               664410 361.63
## TREAT   1   1731549 2395959 405.81  83.397 1.991e-10 ***
## TIME    2    768092 1432502 385.29  18.497 4.586e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

og det er der ikke (husk at `drop1()` giver resultaterne for tests hvor vi prøver at fjerne hver af de mulige effekter i modellen, en ad gangen). Vi kan også betragte parameterestimaterne for at få en fornemmelse af hvilken model det er, vi arbejder med:

```
summary(model)
```

```
##
## Call:
## lm(formula = organic ~ TREAT + TIME)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -285.244  -84.107    9.294   98.035  267.656
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2583.29      48.03  53.784  < 2e-16 ***
## TREATIvermectin   438.63      48.03   9.132 1.99e-10 ***
## TIME12           -197.13      58.83  -3.351  0.00208 **
## TIME16           -357.15      58.83  -6.071 8.83e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 144.1 on 32 degrees of freedom
## Multiple R-squared:   0.79,  Adjusted R-squared:  0.7703
## F-statistic: 40.13 on 3 and 32 DF,  p-value: 5.912e-11
```

Nu er vi klar til at starte på at udføre modelkontrol. Bemærk at modellen kan skrives

$$Y_i = \alpha_{\text{TREAT}(i)} + \beta_{\text{TIME}(i)} + e_i$$

hvor vi antager at $e_i$'erne er uafhænhige og identisk fordelte (iid) med $e_1 \sim N(0, \sigma^2)$. Hvilke antagelser er der samlet set i sådan en model, som vi bør overveje validiteten af?

1. Uafhængighed mellem observationerne
2. Normalfordeling af fejlene/residualerne/støjleddene
3. Varianshomogenitet, dvs. alle residualerne ($e_i$'erne) kan beskrives ved den samme varians, $\sigma^2$

Den første antagelse kan vi ikke undersøge validiteten af empirisk. Der må fageksperter (jer!) overveje, om det i det konkrete eksperiment er rimeligt at antage uafhængighed mellem observationerne. Den anden antagelse kan vi undersøge med et såkaldt QQ-plot, hvor vi sammenligner de observerede residualers fordeling med en normalfordeling. Den tredje antagelse kan vi undersøge vha. et residualplot, hvor vi plotter de observerede residualer mod modellens fittede værdier, dvs. modellens prædiktioner for hver enkel observation i datasættet.

Vi starter med at bestemme residualerne og gemme dem som en ny variabel. Bemærk at vi finder de standardiserede residualer, dvs. residualer hvor residualmiddelværdien er trukket fra og hvor der derefter er delt med spredningen af residualerne.

```
res <- rstandard(model)
```

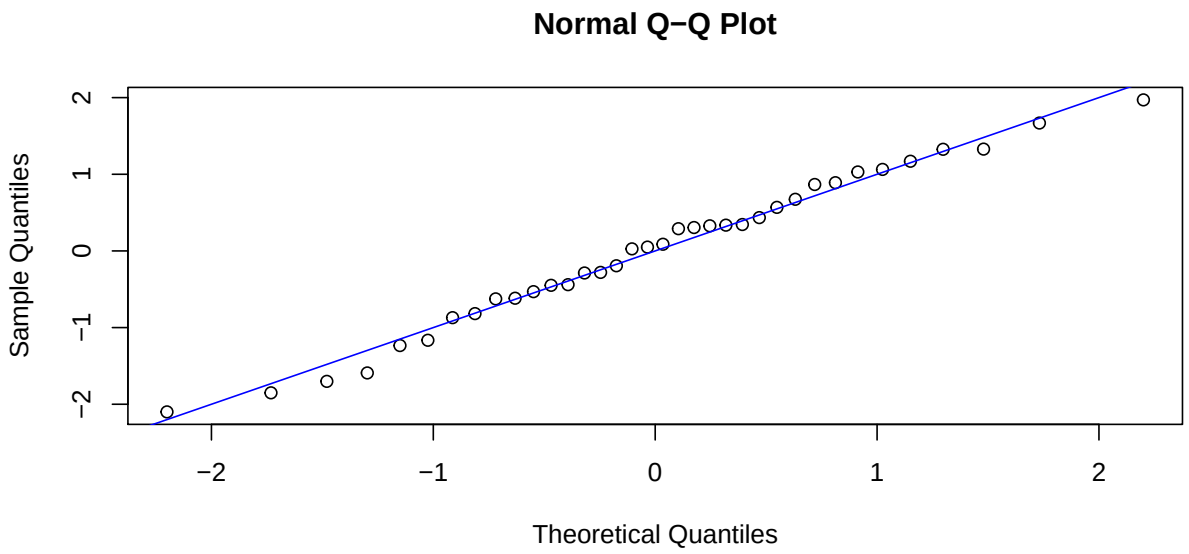
Vi bestemmer nu de fittede værdier:

```r
fit <- fitted(model)

#alternativ metode:
fit <- predict(model)
```

Og vi laver et QQ-plot:

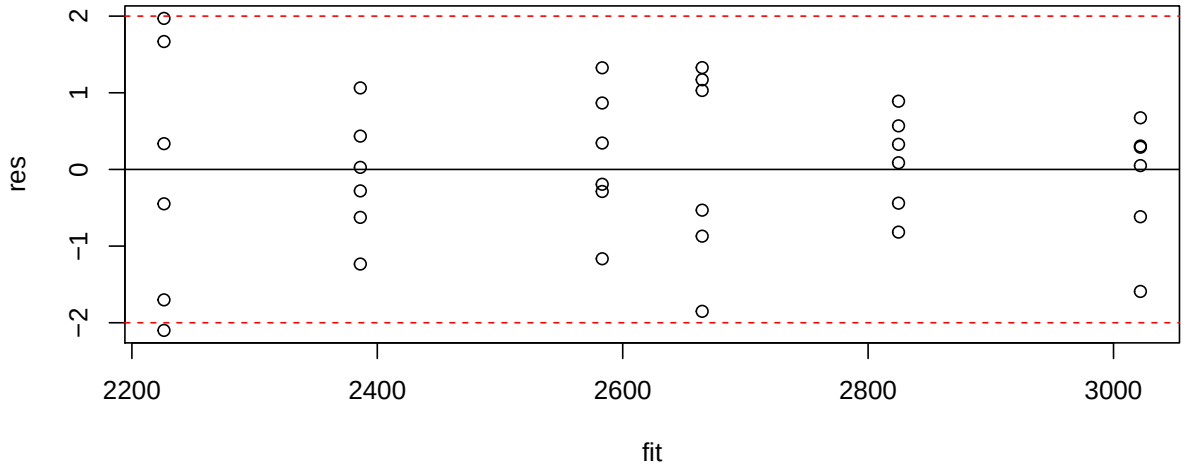
```r
qqnorm(res)
abline(0,1, col="blue")
```




Vi tilføjer desuden en ret linje med intercept 0 og hældning 1. Hvis antagelsen om normalfordeling (antagelse 2) er opfyldt, bør punkterne ligge tæt omkring denne linje. Vi ser på plottet, at det i høj grad er tilfældet. Der er lidt afvigelser, særligt i halerne af fordelingen (dvs. små og store fittede værdier), men det ser ikke alarmerende ud. Vi konkluderer altså at antagelsen om normalfordeling af residualerne er rimelig.

Vi laver nu et residualplot for at undersøge om antagelse 3 også er rimelig:

```r
plot(fit,res)
abline(0,0)
abline(-2,0, col="red", lty="dashed")
abline(2,0, col="red", lty="dashed")
```



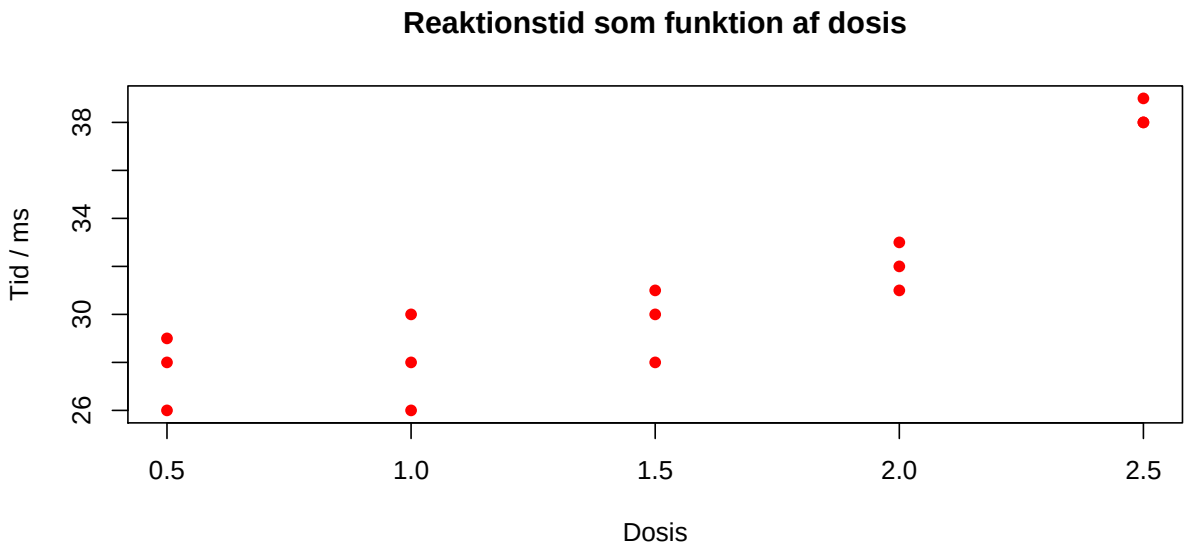

Hvis antagelsen om varianshomogenitet er opfyldt, bør punkterne ligge tilfældigt omkring 0 (den sorte linje) over hele x-aksen. Dvs. at der ikke må være nogle systematiske tendenser, hvor fx. små fittede værdier giver små residualer (tæt på 0) og store fittede værdier giver store residualer (langt fra 0). På dette plot ser det umiddelbart fint ud - der er ikke umiddelbart nogen systematiske tendenser. Vi konkluderer altså, at antagelsen om varianshomogenitet ser ud til at være fornuftig.

Vi har tilføjet to røde, stiplede linjer ved hhv. y=-2 og y=2. Disse svarer til 0.025 og 0.975-fraktilerne for en standard normalfordeling. Altså kan vi også supplere vores undersøgelse af normalfordelingsantagelsen fra ovenfor ved at afgøre, om mere end 5% af observationerne ligger over/under disse linjer. Det ser ikke ud til at være tilfældet, og altså bekræfter dette plot også, at normalfordelingsantagelsen er rimelig.

## Opgave 2.B

Vi starter med at loade data, kigge på det og attache:

```
reak_data <- read.table("opg2B.txt", header=T)
attach(reak_data)
plot(dosis, tid, xlab = "Dosis", ylab = "Tid / ms",
     main = "Reaktionstid som funktion af dosis",
     pch = 16, col = "red")
```




Vi fitter nu to modeller:

- Model A: Effekt af dosis og dosis$^2$
- Model B: Effekt af dosis

Model A kan skrives formelt som

$$Y_i = \alpha + \beta \cdot x_i + \gamma \cdot x_i^2 + e_i$$

for $i = 1...15$, hvor vi antager at $e_i$'erne er iid med $e_1 \sim N(0, \sigma^2)$. $Y_i$ repræsenterer den $i$'te observations reaktionstid, mens $x_i$ repræsenterer denne observations dosis. Tilsvarende (og med de samme antagelser) kan vi skrive model B som

$$Y_i = \alpha + \beta \cdot x_i + e_i$$

Spørgsmålet, som kan belyses ved at sammenligne de to modeller er altså hvorvidt sammenhængen mellem dosis og reaktionstid bedst kan beskrives som kvadratisk (model A) eller blot lineær (model B). Vi fitter de to modeller i R:

```
dosis2<-dosis*dosis
modelA<-lm(tid~dosis+dosis2)
modelB<-lm(tid~dosis)
```

Bemærk, at de to modeller er nestede (dvs. indeholdt i hinanden) og altså kan vi bruge en F-test til at afgøre, om vi kan fjerne det kvadratiske led fra model A og dermed reducere til model B:

```
anova(modelB, modelA)
```

```
## Analysis of Variance Table
##
## Model 1: tid ~ dosis
## Model 2: tid ~ dosis + dosis2
##   Res.Df    RSS Df Sum of Sq     F   Pr(>F)
## 1     13 57.200
```

```
## 2     12 22.819  1    34.381 18.08 0.001123 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Vi finder en meget lav $p$-værdi ($p = 0.001$) og konkluderer altså, at der er signifikant effekt af det kvadratiske led. Vi kan nu gå videre med model A og undersøge om det lineære led og interceptet er signifikant forskellige fra 0:

```
summary(modelA)
```

```
##
## Call:
## lm(formula = tid ~ dosis + dosis2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8762 -1.0429  0.1238  0.8333  2.3048
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  29.8667     1.7076  17.491 6.63e-10 ***
## dosis        -5.7905     2.6026  -2.225  0.04603 *
## dosis2        3.6190     0.8511   4.252  0.00112 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.379 on 12 degrees of freedom
## Multiple R-squared:  0.9086, Adjusted R-squared:  0.8934
## F-statistic: 59.66 on 2 and 12 DF,  p-value: 5.82e-07
```

og vi ser (under `Pr(>|t|)`) at alle effekter er signifikant forskellige fra 0. Dermed er slutmodellen model A. Vi kan aflæse modellens paramterestimater fra `summary()`-outputtet og vi finder at

$$\hat{\alpha} = 29.8667,\ \hat{\beta} = -5.7905,\ \hat{\gamma} = 3.6190$$

og

$$s = 1.379$$

Vi finder desuden 95% konfidensintervaller for hver af middelværdiparametrene (dvs. $\alpha, \beta$ og $\gamma$):

```
confint(modelA)
```

```
##                  2.5 %     97.5 %
## (Intercept)  26.146209 33.5871242
## dosis       -11.460948 -0.1200042
## dosis2        1.764605  5.4734903
```

Vi prædikterer nu reaktionstiden for en person med dosis 1.8 og en person med dosis 4.0 vha. `predict()`:

```
predData <- data.frame(dosis=c(1.8, 4),
                       dosis2=c(1.8, 4)^2)
predict(modelA, new=predData)
```

```
##        1        2
## 31.16952 64.60952
```

Bemærk, at vi ikke ud fra data kan sige noget om hvad der vil ske for en dosis=4, da jo de betragtede doser ligger i intervallet [0.5,2.5]:

```
range(dosis)
```

```
## [1] 0.5 2.5
```

Der er ingen grund til at antage at reaktionstiden påvirkes på samme måde for doser uden for dette interval, som modellen forudsiger at den vil gøre for doser i intervallet! Altså er det ikke nødvendigvis meningsfuldt at bruge modellen til dette formål.

Vi laver nu 95%-konfidensintervaller for værdierne fra ovenfor vha. `estimable()`:

```
library(gmodels)
est18<-c(1,1.8,1.8^2) #1 for intercept, 1.8 for dosis,
                      #1.8^2 for dosis2
est4<-c(1,4,4^2)
est=rbind(est18,est4)
estimable(modelA,est,conf.int=0.95)
```

```
##       Estimate Std. Error  t value DF     Pr(>|t|) Lower.CI Upper.CI
## est18 31.16952  0.5209276 59.83466 12 4.440892e-16 30.03452 32.30453
## est4  64.60952  5.0658043 12.75405 12 2.445371e-08 53.57208 75.64696
```

Bemærk at konfidensintervallet for dosis=4 er langt bredere end det for dosis=1.8.

Alternativt kan vi også finde konfidensintervallerne vha. `predict()`-funktionen:

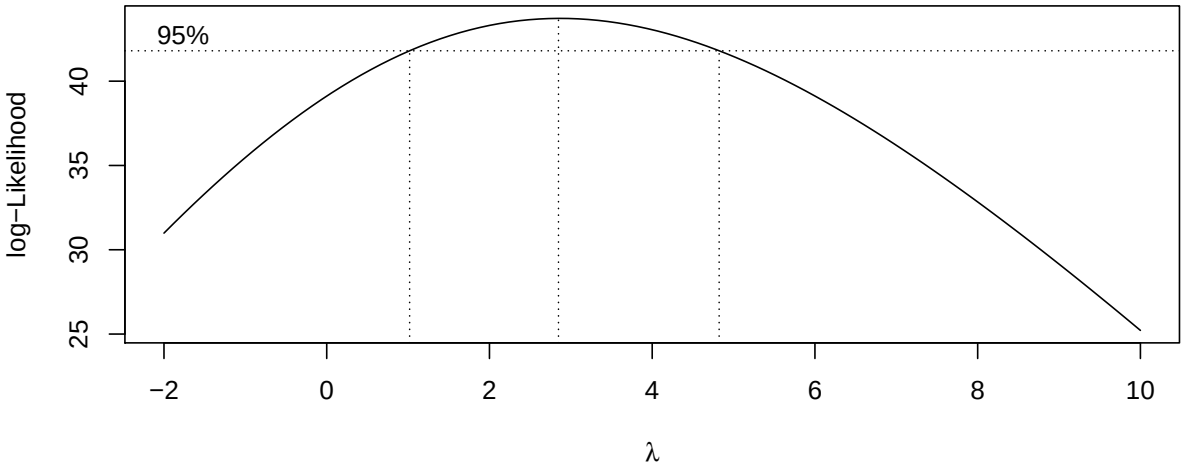
```
predict(modelA, predData, interval="confidence")
```

```
##        fit      lwr      upr
## 1 31.16952 30.03452 32.30453
## 2 64.60952 53.57208 75.64696
```

# Lidt om boxcox

Vi tegner boxcox-plot for modellen fra opg. 2.A:

```
library(MASS)
boxcox(model, lambda=-2:10)
```




og vi gemmer lambda-værdierne og de dertilhørende log-likelihood-værdier:

```
bc <- boxcox(model,lambda=-2:10, plotit=F)
#plotit=F fordi vi ikke vil se endnu et plot
```

Husk at være sikker på at max ligger i lambda-intervallet - ellers nytter det ikke noget!

Vi finder ud af hvilken værdi af lamdba der maksimerer log-likelihood-funktionen:

```
bc$x[which.max(bc$y)]
```

```
## [1] 3
```

og altså vil vi forvente, at vi får det bedste modelfit (mht. normalfordelingsantagelsen), hvis vi bruger $Y_i^3$ som responsvariabel.