SD2 - uge 5, tirsdag

Anne Petersen

Vi starter med at sætte working directory:

```
setwd("C:/Users/Anne/Dropbox/Arbejde/STATforLIFE2/uge5")
```

Opgave 6.1 fra dokument på Absalon

Vi indlæser data og definerer et nyt datasæt som delmængden af det fulde datasæt, hvor `day` er 91:

```
goatdata <- read.table("goats1.txt", header=T)
goat91 <- subset(goatdata, day==91) #vælger kun data fra dag 91
head(goat91) #kigger på data
```

```
##    goat feed   w0 day weight
## 5     1    1 20.4  91   22.3
## 10    2    1 10.3  91   12.5
## 15    3    1 12.5  91   15.4
## 20    4    1 10.8  91   12.5
## 25    5    1 13.6  91   15.8
## 30    6    1 19.0  91   19.6
```

```
attach(goat91)
goat <- factor(goat) #laver "goat" til faktor
feed <- factor(feed) #laver "feed" til faktor
```

Vi laver nu en ny variabel, `y`, som er forholdet mellem vægten dag 91 og startvægten (`w0`):

```
y <- weight/w0
```

Vi vil nu gerne analysere data. Vores mål er at afgøre, om vægtforøgningen over de 91 dage afhænger af fodertypen. Vi kan bruge `y` fra ovenfor som respons, da den er et mål for vægtøgningen. Ved at inddrage `feed` som forklarende variabel, gør vi desuden muligt at sammenligne vægtforøgelsen for forskellige fodertyper. Sidst, men ikke mindst, bør vi også inddrage startvægten, `w0`, for at tage højde for at den relative vægtøgning (`y`) måske kan afhænge af gedens startvægt. Vi opstiller altså følgende model:

$$Y_i = \alpha(\text{feed}_i) + \beta \cdot \text{w}_0 + e_i$$

for $i = 1, ..., 28$, hvor det antages at e_i'erne er iid med $e_1 \sim N(0, \sigma^2)$. Vi fitter den beskrevne model i R:

```
model <- lm(y ~ w0+feed-1)
```

og vi ser, om denne model kan reduceres, dvs. om effekterne af hhv. `feed` eller `w0` er insignifikante:

```
model1 <- lm(y ~ w0)
model2 <- lm(y ~ feed-1)
anova(model1, model)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ w0
## Model 2: y ~ w0 + feed - 1
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     26 0.25985
## 2     23 0.05897  3   0.20088 26.116 1.376e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

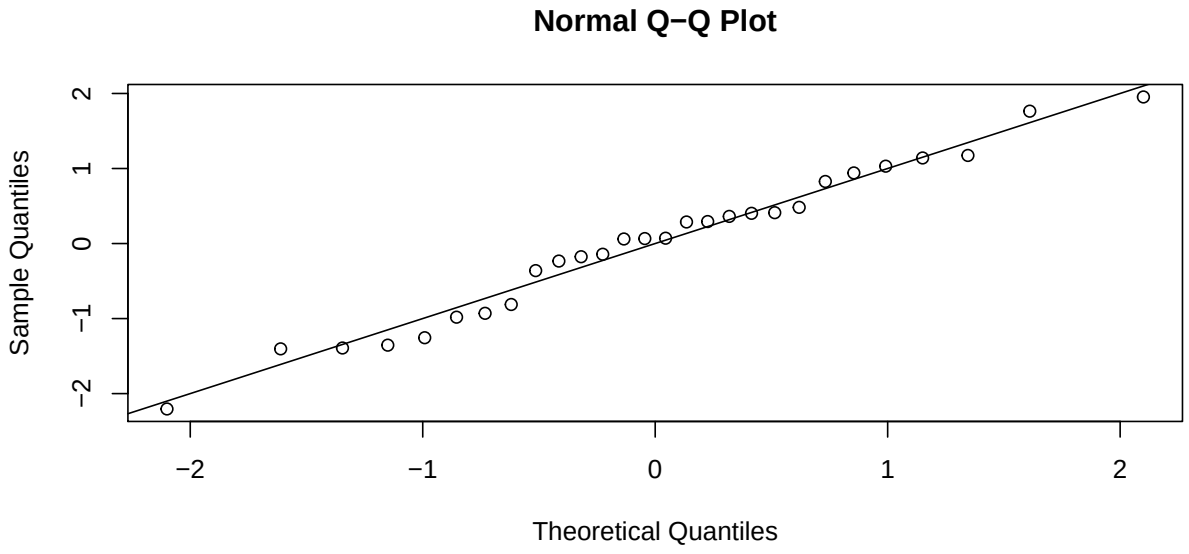
```
anova(model2, model)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ feed - 1
## Model 2: y ~ w0 + feed - 1
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1     24 0.096282
## 2     23 0.058970  1  0.037311 14.552 0.0008901 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Vi finder signifikante effekter af både `feed` ($p = 1.4 \cdot 10^{-7}$) og `w0` ($p = 0.00089$) og altså er vores slutmodel `model`.

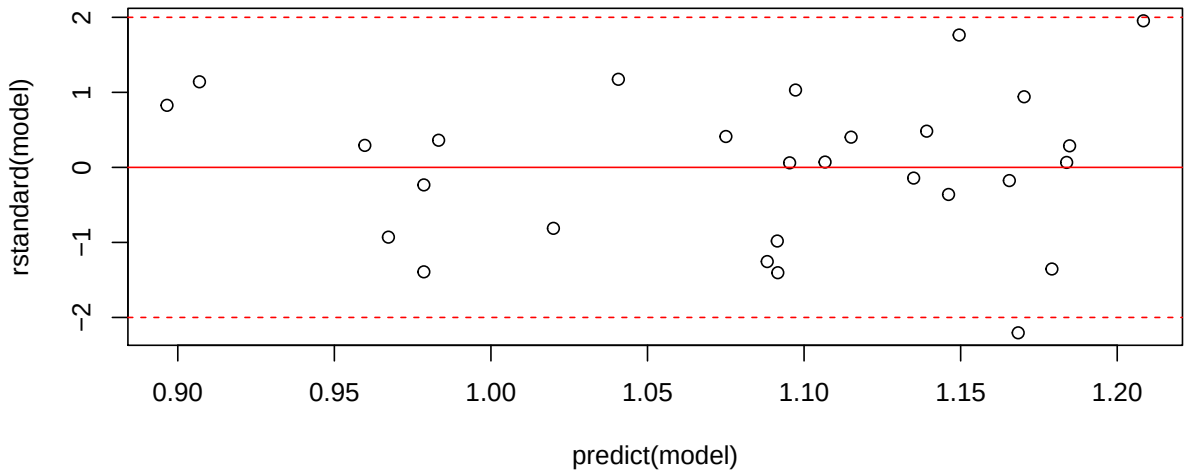
Vi kontrollerer nu modellens antagelser. Vi starter med at undersøge rimeligheden af normalfordelingsantagelsen vha. et QQ-plot:

```
qqnorm(rstandard(model))
abline(0,1)
```



Vi ser, at punkterne ligger pænt omkring den rette linje, hvilket indikerer, at normalfordelingsantagelsen er rimelig. Vi undersøger nu hvordan det forholder sig med varianshomogenitetsantagelsen (dvs. antagelsen om at vi kan bruge samme σ til at beskrive fordelingen af alle e_i'erne) vha. et residualplot:

```r
plot(predict(model), rstandard(model))
abline(0,0, col="red")
abline(-2, 0, col="red", lty=2)
abline(2, 0, col="red", lty=2)
```



Vi ser ikke umiddelbart en systematisk placering af punkterne, hvilket tyder på at varianshomogenitetsantagelsen er fornuftig. Vi finder lidt flere numerisk store residualværdier for store fittede værdier, men samtidig også flere små residualværdier her, og da det er der, det meste af data er koncentreret, tyder det ikke på problemer med antagelsen. Bemærk, at vi også har indtegnet stiplede linjer ved hhv. -2 og 2, og at der maksimalt må være 95% af observationerne, som falder uden for disse linjer i følge normalfordelingsantagelsen. Vi ser, at residualplottet derved bekræfter konklusionen fra QQ-plottet ovenfor; normalfordelingsantagelsen fremstår rimelig.

Det ser altså ud til, at modellen i rimelig grad opfylder sine antagelser. Vi betragter derfor nu parameterestimaterne for at konkludere hvad den siger om sammenhængen mellem vægtøgning og fodertype:

```r
library(pander)
coefs <- summary(model)$coefficients[,c(1,4)]
cis <- confint(model)
names <- c("beta", "alpha(feed1)", "alpha(feed2)",
           "alpha(feed3)", "alpha(feed4)")
rownames(coefs) <- names
pander(cbind(coefs, cis))
```

	Estimate	Pr(>|t|)	2.5 %	97.5 %
beta	-0.009429	0.0008901	-0.01454	-0.004316
alpha(feed1)	1.267	1.167e-20	1.186	1.349
alpha(feed2)	1.201	6.255e-21	1.126	1.276
alpha(feed3)	1.285	4.763e-21	1.206	1.364
alpha(feed4)	1.066	5.28e-20	0.9932	1.139

(Kommentarer til koden: Det meste der sker her, handler om at vise resultaterne pænt. Funktionen `pander` i pakken `pander` laver pæne tabeller og vi gemmer den information, vi er interesserede i fra `summary(model)` under `coefs` for at kunne lave en overskuelig tabel uden for meget information. Bemærk at to separate kald til hhv. `summary(model)` og `confint(model)` fint kunne bruges til at besvare spørgsmålet.)

Bemærk først, at en stor `y`-værdi svarer til en stor relativ øgning i vægt over perioden. Det negative parameterestimat svarende til `w0` indikerer altså, at vi forventer mindre relative vægtforøgelser for store end for små geder. Vi bemærker desuden, at `feed4` har et meget lavere parameterestimat end de øvrige fodertyper, dvs. at det ser ud til, at `feed4` giver en markant mindre vægtforøgelse end de øvrige fodertyper. Bemærk, at forsøget er balanceret med 7 observationer i hver fodertype. Vi kan derfor konkludere, at eftersom $\hat{\alpha}_{\text{feed4}} = 1.066$ ikke er indeholdt i nogen af de andre $\hat{\alpha}'ers$ konfidensintervaller, er der signfikant forskel på vægtforøgelsen for `feed4` og de øvrige fodertyper. Tilsvarende ser vi, at der ikke er signifkant forskel mellem `feed1` og `feed2` eller `feed3`, men at der er en (lille) signifikant forskel på `feed2` og `feed3`.

Vi vil nu prædiktere vægten efter 91 dage for en, hvis svartvægt var 12 kg. Vi bestemmer prædiktioner for alle de 4 fodertyper. For god orden skyld, starter vi lige med at undersøge, om det overhovedet er rimeligt at sige noget om en ged med en startvægt på 12 kg ud fra vores datasæt:

```
summary(w0)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.00   10.45   11.20   12.93   16.92   20.50
```

Det ser fornuftigt ud - 12 ligger omkring midten af fordelingen af `w0` og vores datasæt er således meget velegnet til at svare på spørgsmålet.

Vi vil nu prædiktere den forventede vægt efter 91 dage. Bemærk, at vores model prædikterer den forventede vægtøgning og altså skal vi gange de umiddelbare resultater med startvægten, for at opnå det rigtige tal:

```
new_pred <- data.frame(w0=12, feed=c("1", "2", "3", "4"))
12*predict(model, new_pred, interval="confidence")
```

```
##        fit      lwr      upr
## 1 13.85086 13.36209 14.33962
## 2 13.05408 12.57827 13.52990
## 3 14.05914 13.57635 14.54193
## 4 11.43744 10.96235 11.91252
```

```
#Alternativ metode (estimable):
library(gmodels)
est1 <- c(12,1,0,0,0)
est2 <- c(12,0,1,0,0)
est3 <- c(12,0,0,1,0)
est4 <- c(12,0,0,0,1)
ests <- rbind(est1,est2,est3,est4)
12*estimable(model, ests, conf.int=0.95)
```

```
##      Estimate Std. Error  t value  DF Pr(>|t|) Lower.CI Upper.CI
## est1 13.85086  0.2362728 703.4676 276        0 13.36209 14.33962
## est2 13.05408  0.2300117 681.0481 276        0 12.57827 13.52990
## est3 14.05914  0.2333840 722.8847 276        0 13.57635 14.54193
## est4 11.43744  0.2296605 597.6178 276        0 10.96235 11.91252
```

Opgave 6.3

Vi indlæser data, betagter det og gemmer passende variable som faktorer:

```
data <- read.table("maj04_2.txt", header=T)
head(data)
```

```
##   rotte behandling uge fedt glucose
## 1     4  kastreret   2 1206     6.7
## 2     4  kastreret  10 1677     5.9
## 3     6  kastreret   2 1222     6.0
## 4     6  kastreret  10 1813     5.4
## 5     8  kastreret   2 1749     6.0
## 6     8  kastreret  10 2441     5.8
```

```
data$rotte <- factor(data$rotte)
data$uge <- factor(data$uge)
```

(Tegn selv et faktordiagram)

Vi vil nu opstille en statistisk model, så vi kan undersøge, om fedtfordelingen for rotter (`fedt`) afhænger af, om de er blevet kastrerede eller ej (`behandling`). Vi vil desuden tage højde for, at effekten kan afhænge af hvor lang tid efter det eventuelle indgreb, vi måler, og inddrager derfor `uge` som forklarende variabel i vekselvirkning med `behandling`. Vi er ikke specielt interesserede i netop de rotter, som indgår i forsøget, og vi har to målinger pr. rotte. Vi inddrager derfor `rotte` som tilfældig effekt. Vi opstiller altså følgende model:

$$Y_i = \alpha(\text{uge}_i \times \text{behandling}_i) + b(\text{rotte}_i) + e_i$$

for $i = 1...44$ og hvor vi antager at e_i'erne er iid med $e_1 \sim N(0, \sigma^2)$ og at $b(1), ..., b(22)$ er iid med $b(j) \sim N(0, \sigma_b^2)$ for alle $j = 1, ..., 22$. Vi fitter denne model i R:

```
library(nlme)
```

```
## Warning: package 'nlme' was built under R version 3.1.3
```

```
modelA <- lme(fedt ~ uge*behandling, random=~1|rotte, method="ML",
              data)
```

Vi undersøger, om vi kan reducere til en additiv model, dvs. vi ser om der er signifikant effekt af vekselvirkningen mellem `uge` og 'behandling´:

```
modelB <- lme(fedt ~ uge+behandling, random=~1|rotte, method="ML",
              data)
anova(modelA,modelB)
```

```
##        Model df      AIC      BIC    logLik   Test   L.Ratio p-value
## modelA     1  6 618.0334 628.7386 -303.0167
## modelB     2  5 616.5358 625.4568 -303.2679 1 vs 2 0.5023943  0.4784
```

Vi finder $p = 0.4784$ og konkluderer dermed, at der ikke er signifikant effekt af vekselvirkningen. Vi reducerer derfor til `modelB` og undersøger, om der er nogen af hovedeffekterne her, som heller ikke er signifikante:

```
modelC <- lme(fedt ~ behandling, random=~1|rotte, method="ML", data)
modelD <- lme(fedt ~ uge, random=~1|rotte, method="ML", data)
anova(modelB, modelC)
```

```
##        Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## modelB     1  5 616.5358 625.4568 -303.2679
## modelC     2  4 675.2241 682.3609 -333.6121 1 vs 2 60.68833  <.0001
```

```
anova(modelB, modelD)
```

```
##        Model df      AIC      BIC    logLik   Test    L.Ratio p-value
## modelB     1  5 616.5358 625.4568 -303.2679
## modelD     2  4 614.6090 621.7458 -303.3045 1 vs 2 0.07321221  0.7867
```

Vi ser, at der er signifikant effekt af `uge` ($p < 0.0001$), men ikke af `behandling´ ($p=0.7867$). Vi reducerer derfor til`modelD‘ og undersøger, om vi kan reducere denne model til den tomme model:

```
modelE <- lme(fedt ~ 1, random=~1|rotte, method="ML", data)
anova(modelD, modelE)
```

```
##        Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## modelD     1  4 614.6090 621.7458 -303.3045
## modelE     2  3 673.2688 678.6214 -333.6344 1 vs 2 60.65979  <.0001
```

og vi finder $p < 0.0001$, hvilket betyder, at der er en signifikant effekt af `uge`. Vores slutmodel er dermed `modelD` som, vi opskriver formelt:

$$Y_i = \gamma(\text{uge}_i) + b(\text{rotte}_i) + e_i$$

for $i = 1...44$ med antagelser og notation som ovenfor. Vi finder modellens parameterestimater (og husker at genfitte modellen med REML-estimation for at opnå pålidelige estimater):

```
modelD_REML <- lme(fedt ~ uge, random=~1|rotte, method="REML", data)
summary(modelD_REML)
```

```
## Linear mixed-effects model fit by REML
##  Data: data
##        AIC      BIC    logLik
##   595.1718 602.1225 -293.5859
##
## Random effects:
##  Formula: ~1 | rotte
##         (Intercept) Residual
## StdDev:     244.864 156.7615
##
## Fixed effects: fedt ~ uge
##                 Value Std.Error DF  t-value p-value
## (Intercept) 1461.5909  61.98701 21 23.57899       0
## uge10        761.8182  47.26537 21 16.11789       0
##  Correlation:
##       (Intr)
## uge10 -0.381
```

```
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -1.81710650 -0.47495527  0.02547204  0.45108803  1.80542211
##
## Number of Observations: 44
## Number of Groups: 22
```

```
#Alternativ metode:
  #Betragter først systematiske effekter
    fixef(modelD_REML)
```

```
## (Intercept)       uge10
##   1461.5909    761.8182
```

```
  #Betrager dernæst tilfældige effekter
    VarCorr(modelD_REML)
```

```
## rotte = pdLogChol(1)
##             Variance StdDev
## (Intercept) 59958.39 244.8640
## Residual    24574.17 156.7615
```

```
#Konfidensintervaller:
intervals(modelD_REML)
```

```
## Approximate 95% confidence intervals
##
##  Fixed effects:
##                 lower      est.     upper
## (Intercept) 1332.6819 1461.5909 1590.4999
## uge10        663.5245  761.8182  860.1119
## attr(,"label")
## [1] "Fixed effects:"
##
##  Random Effects:
##   Level: rotte
##                     lower    est.    upper
## sd((Intercept)) 169.1973 244.864 354.3696
##
##  Within-group standard error:
##    lower     est.    upper
## 115.8502 156.7615 212.1202
```

Uanset metoden, ser vi at $\hat{\alpha}_{uge2} = 1461.6$ (konfidensinterval: $[1332.7, 1590.5]$), mens $\hat{\alpha}_{uge10} = 1461.6 + 761.8 = 2223.4$ (konfidensinterval: $[2094.5, 2352.3]$). Der sker altså generelt en forøgelse i responsvariablen, fedtfordeling, over tid. Vi så desuden, at denne forøgelse ikke påvirkes af hvorvidt rotterne kastreres eller ej. Vi finder desuden at $\hat{\sigma}_b = 244.8640$ og at $s = 156.7615$. Dermed forklarer den tilfældige effekt af `rotte` størstedelen af den tilfældige variation i data, helt præcist forklarer den

$$\frac{\hat{\sigma}_b^2}{s^2 + \hat{\sigma}_b^2} = \frac{59958.39}{24574.17 + 59958.39} = 0.71 = 71\%$$

af variationen.