SD2 - uge 2, tirsdag

Anne Petersen

Opgave 2.2 fra dokument på Absalon

Vi indlæser data og attacher det for lettere adgang til variablene:

```
setwd("C:/Users/Anne/Dropbox/Arbejde/STATforLIFE2/uge2")
terbuthyl <- read.table("ex34.txt", header=T)
attach(terbuthyl)
head(terbuthyl, 10) #de første 10 observationer
```

```
##    TEMP LUC ADP mineral
## 1    10   1   1    5.30
## 2    10   1   0    2.30
## 3    20   1   1    5.20
## 4    20   1   0    3.59
## 5    10   1   1    4.84
## 6    10   1   0    2.26
## 7    20   1   1    5.60
## 8    20   1   0    3.48
## 9    10   0   1    5.21
## 10   10   0   0    2.27
```

Vi ser hvor mange observationer der er i hver kombination af `TEMP` og `LUC` i datasættet:

```
table(TEMP,LUC)
```

```
##     LUC
## TEMP 0 1
##   10 4 4
##   20 4 4
```

Vi ser, at der er lige mange (4) observationer i hver kombination. Hvis vi betragter eksperimentet som et tofaktorforsøg med faktorerne `TEMP` og `LUC`, er det altså balanceret. Lad os se, om det også er balanceret, hvis vi inkluderer den tredje faktor, `ADP`:

```
table(TEMP, LUC, ADP)
```

```
## , , ADP = 0
##
##     LUC
## TEMP 0 1
##   10 2 2
##   20 2 2
##
## , , ADP = 1
##
##     LUC
## TEMP 0 1
##   10 2 2
##   20 2 2
```

Det er det - der er nemlig netop 2 observationer i hver unik kombination af de tre faktorer. Dermed er forsøget balanceret.

Vi omdanner nu alle faktorvariablene til faktorer i R:

```
TEMP <- factor(TEMP)
LUC <- factor(LUC)
ADP <- factor(ADP)
```

Og vi fitter model A fra opgavebeskrivelsen, dvs. en trefaktormodel med andenordens vekselvirkninger (`TEMP:LUC:ADP`), førsteordens vekselvirkninger (`TEMP:LUC`, `TEMP:ADP`, `ADP:LUC`) og hovedeffekter/marginale effekter (`TEMP`, `LUC`, `ADP`):

```
modelA <- lm(mineral ~ TEMP*LUC*ADP)
```

Vi fitter desuden endnu en model, som kun inkluderer andenordens vekselvirkningen fra ovenfor:

```
modelB <- lm(mineral ~ TEMP:LUC:ADP)
```

Vi sammenligner nu de to modellers `summary()`-output:

```
summary(modelA)
```

```
##
## Call:
## lm(formula = mineral ~ TEMP * LUC * ADP)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2500 -0.0925  0.0000  0.0925  0.2500
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.5200     0.1560  16.159 2.16e-07 ***
## TEMP20             0.9800     0.2205   4.443  0.00216 **
## LUC1              -0.2400     0.2205  -1.088  0.30821
## ADP1               2.8650     0.2205  12.990 1.17e-06 ***
## TEMP20:LUC1        0.2750     0.3119   0.882  0.40367
## TEMP20:ADP1       -0.6600     0.3119  -2.116  0.06724 .
## LUC1:ADP1         -0.0750     0.3119  -0.240  0.81603
## TEMP20:LUC1:ADP1  -0.2650     0.4411  -0.601  0.56463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2206 on 8 degrees of freedom
## Multiple R-squared:  0.9856, Adjusted R-squared:  0.973
## F-statistic: 78.13 on 7 and 8 DF,  p-value: 9.845e-07
```

```
summary(modelB)
```

```
##
## Call:
```

```
## lm(formula = mineral ~ TEMP:LUC:ADP)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2500 -0.0925  0.0000  0.0925  0.2500
##
## Coefficients: (1 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         5.4000     0.1560  34.625 5.29e-10 ***
## TEMP10:LUC0:ADP0   -2.8800     0.2205 -13.058 1.12e-06 ***
## TEMP20:LUC0:ADP0   -1.9000     0.2205  -8.615 2.55e-05 ***
## TEMP10:LUC1:ADP0   -3.1200     0.2205 -14.146 6.06e-07 ***
## TEMP20:LUC1:ADP0   -1.8650     0.2205  -8.456 2.92e-05 ***
## TEMP10:LUC0:ADP1   -0.0150     0.2205  -0.068    0.947
## TEMP20:LUC0:ADP1    0.3050     0.2205   1.383    0.204
## TEMP10:LUC1:ADP1   -0.3300     0.2205  -1.496    0.173
## TEMP20:LUC1:ADP1       NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2206 on 8 degrees of freedom
## Multiple R-squared:  0.9856, Adjusted R-squared:  0.973
## F-statistic: 78.13 on 7 and 8 DF,  p-value: 9.845e-07
```

For `modelA` ser vi, at estimaterne viser dels effekten af `TEMP`, `LUC` og `ADP` særskilt, dels effekten af kombinationerne af disse faktorer. Bemærk at referencegruppen er `TEMP=10`, `LUC=0`, `ADP=0`. For `modelB` ser vi, at estimaterne viser effekten af hver kombination af `TEMP`, `LUC` og `ADP`. Bemærk at referencegruppen nu er `TEMP=20`, `LUC=1`, `ADP=1` (derfor er der bare angivet `NA`'er for denne kombination). Bemærk at de to modeller indeholder den samme information, bare skrevet forskelligt. De har fx. samme estimat for residualvariansen (σ), nemlig 0.2206, og de dikterer de samme estimater for hver af de $2 \cdot 2 \cdot 2 = 8$ forskellige kombinationer af de tre faktorer (her vist ved at de prædikterer den samme værdi for hver af de 16 observationer i datasættet):

```
round(predict(modelA), 10) == round(predict(modelB), 10)
```

```
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##   16
## TRUE
```

Bemærk at vi benytter afrundingskommandoen `round()` til sammenligningen, fordi R ellers vil konkludere at visse prædiktioner er forskellige pga. afrundingsfejl. Ved at skrive `round(predict(modelA), 10)` ser vi på prædiktionerne fra `modelA` med 10 decimaler - og det at der står `TRUE` ovenfor betyder, at de to modeller prædikterer ens op til (mindst) de første 10 decimaler.

Vi bliver nu bedt om at gennemføre modelreduktion. Jf. det hierarkiske princip, vil vi først teste, om vi kan fjerne vekselvirkningseffekten som har den største orden, dvs. effekten `TEMP:LUC:ADP`. For at gøre modelreduktionen fuldstændig transparant, genfitter vi modellerne fra ovenfor, så man eksplicit kan læse klart alle effekter, som indgår. Bemærk at `modelA1` nedenfor og `modelA` er fuldstændigt identiske.

```
modelA1 <- lm(mineral ~ TEMP + LUC + ADP + TEMP:LUC + TEMP:ADP +
                LUC:ADP + TEMP:LUC:ADP)
modelC <- lm(mineral ~ TEMP + LUC + ADP + TEMP:LUC + TEMP:ADP +
                LUC:ADP)
anova(modelC, modelA1)
```

```
## Analysis of Variance Table
##
## Model 1: mineral ~ TEMP + LUC + ADP + TEMP:LUC + TEMP:ADP + LUC:ADP
## Model 2: mineral ~ TEMP + LUC + ADP + TEMP:LUC + TEMP:ADP + LUC:ADP +
##     TEMP:LUC:ADP
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1      9 0.40671
## 2      8 0.38915  1  0.017556 0.3609 0.5646
```

Vi finder en teststørrelse på $F = 0.36$ ved $(1, 8)$ frihedsgrader, svarende til $p = 0.5646$. Vi konkluderer altså, at der ikke er signifikant effekt af andenordensvekselvirkningen `TEMP:LUC:ADP` og går videre med den reducerede `modelC`. I denne model kan vi forsøge at fjerne de tre førsteordens vekselvirkningsled, altså `TEMP:LUC`, `TEMP:ADP` og `TEMP:ADP`. Men hvilken rækkefølge skal vi forsøge at fjerne dem i? Én strategi er at prøve at se hvad der sker, hvis vi fjerner én af dem fra modellen. Hvis nogen af disse reduktioner så fører til $p > 0.05$, kan vi vælge at gå videre med den model, som giver den største p-værdi. Dette kan gøres smart vha. funktionen `drop1()`:

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

```
## Single term deletions
##
## Model:
## mineral ~ TEMP + LUC + ADP + TEMP:LUC + TEMP:ADP + LUC:ADP
##          Df Sum of Sq     RSS     AIC F value   Pr(>F)
## <none>                0.40671 -44.756
## TEMP:LUC  1   0.02031 0.42701 -45.976  0.4494 0.519469
## TEMP:ADP  1   0.62806 1.03476 -31.815 13.8983 0.004712 **
## LUC:ADP   1   0.04306 0.44976 -45.146  0.9528 0.354515
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Vi ser, at den største p-værdi opnås ved at fjerne `TEMP:LUC`-leddet, så det gør vi. Her fås $F = 0.4494$ og $p = 0.52$. Vi ser derefter, om modellen nu kan reduceres yderligere:

```
modelD <- lm(mineral ~ TEMP + LUC + ADP + TEMP:ADP + LUC:ADP)
drop1(modelD, test="F")
```

```
## Single term deletions
##
## Model:
## mineral ~ TEMP + LUC + ADP + TEMP:ADP + LUC:ADP
##          Df Sum of Sq     RSS     AIC F value   Pr(>F)
## <none>                0.42701 -45.976
## TEMP:ADP  1   0.62806 1.05507 -33.504 14.7081 0.003291 **
## LUC:ADP   1   0.04306 0.47007 -46.439  1.0083 0.338986
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

og vi ser, at vi bør fjerne `LUC:ADP`-leddet ($F = 1.0083$, $p = 0.34$). Vi fitter denne nye model som `modelE` og ser, om vi også kan fjerne det sidste vekselvirkningsled:

```
modelE <- lm(mineral ~ TEMP + LUC + ADP + TEMP:ADP)
modelF <- lm(mineral ~ TEMP + LUC + ADP)
anova(modelF, modelE)
```

```
## Analysis of Variance Table
##
## Model 1: mineral ~ TEMP + LUC + ADP
## Model 2: mineral ~ TEMP + LUC + ADP + TEMP:ADP
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1     12 1.09812
## 2     11 0.47007  1   0.62806 14.697 0.002777 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

og vi finder $F = 14.697$ og $p = 0.003$ for testen af ingen effekt af `TEMP:ADP`. Altså er der signifikant effekt af dette vekselvirkningsled. Bemærk dog, at `LUC` ikke længere indgår i et vekselvirkningsled. Altså kan vi godt teste om denne effekt kan fjernes:

```
modelG <- lm(mineral ~ TEMP + ADP + TEMP:ADP)
anova(modelG, modelE)
```

```
## Analysis of Variance Table
##
## Model 1: mineral ~ TEMP + ADP + TEMP:ADP
## Model 2: mineral ~ TEMP + LUC + ADP + TEMP:ADP
##   Res.Df     RSS Df Sum of Sq      F  Pr(>F)
## 1     12 0.64022
## 2     11 0.47007  1   0.17016 3.9818 0.07136 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

og det kan den - vi finder nemlig $F = 3.9818$ svarende til $p = 0.07$. Altså reducrer vi til `modelG`. Vi tjekker om vekselvirkningsleddet `TEMP:ADP` nu er blevet insignifikant:

```
modelH <- lm(mineral ~ TEMP + ADP)
anova(modelH, modelG)
```

```
## Analysis of Variance Table
##
## Model 1: mineral ~ TEMP + ADP
## Model 2: mineral ~ TEMP + ADP + TEMP:ADP
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1     13 1.26828
## 2     12 0.64022  1   0.62806 11.772 0.004976 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

og det er det ikke. Det betyder, at `modelG` ikke kan reduceres yderligere og det er dermed vores slutmodel.

Vi finder nu parameterestimaterne for hver af de fire grupper, som bestemt af vores slutmodel, `modelG`:

```
summary(modelG)
```

```
##
## Call:
## lm(formula = mineral ~ TEMP + ADP + TEMP:ADP)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38750 -0.10750 -0.00250  0.07625  0.37000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.4000     0.1155  20.781 8.91e-11 ***
## TEMP20        1.1175     0.1633   6.842 1.79e-05 ***
## ADP1          2.8275     0.1633  17.312 7.47e-10 ***
## TEMP20:ADP1  -0.7925     0.2310  -3.431  0.00498 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.231 on 12 degrees of freedom
## Multiple R-squared:  0.9763, Adjusted R-squared:  0.9704
## F-statistic: 164.6 on 3 and 12 DF,  p-value: 5.168e-10
```

Bemærk at referencegruppen er `TEMP=10`, `ADP=0`. Altså får vi følgende parameterestimater:

$$\text{TEMP=10, ADP=0:} 2.4000$$

$$\text{TEMP=10, ADP=1:} 2.4000 + 2.8275 = 5.2275$$

$$\text{TEMP=20, ADP=0:} 2.4000 + 1.1175 = 3.5175$$

$$\text{TEMP=20, ADP=1:} 2.4000 + 1.1175 + 2.8275 - 0.7925 = 5.5525$$

Estimatet for spredningen i slutmodellen ses at være

$$s = 0.231$$

Taller er aflæst som "residual standard error" i `summary()`-outputtet. Vi gemmer det også lige under navnet `sigma`:

```
sigma <- summary(modelG)$sigma
```

Vi finder nu LSD-værdien svarende for `TEMP*ADP`, dvs. den mindste forskel der skal være mellem to forskellige grupper af `TEMP` og `ADP`-kombinationer, før den er signifikant.

```
(LSD_TEMP.ADP <- qt(0.975,12)*sigma*sqrt(2/4))
```

```
## [1] 0.3558612
```

Bemærk, at vi bruger 12 frihedsgrader fordi der er 16 observationer, fratrukket $2 \cdot 2$ kategorier. Vi ganger med $\sqrt{2/4}$ fordi vi sammenligner to tal som stammer fra en faktor (`TEMP:ADP`), som er balanceret med 4 observationer i hver kategori. Bemærk desuden, at $s \cdot \sqrt{1/2}$ er det samme som

$$\text{SD(et gruppegennemsnit - et andet gruppegennemsnit)}$$

og da forsøget er balanceret, er denne størrelse det samme, uanset hvilke to grupper, der sammenlignes. Specielt kan vi finde tallet ved at kigge på i `summary()`-outputtets standard error for fx. `TEMP20`, som vi gemmer:

```
(SD_diff <- summary(modelG)$coefficients[2,2])
```

```
## [1] 0.163328
```

```
sigma*sqrt(2/4)
```

```
## [1] 0.163328
```

og altså kan LSD-størrelsen her også udregnes ved

```
SD_diff*qt(0.975,12)
```

```
## [1] 0.3558612
```