Gennemgang uge 3 MatStat

Cecilie Olesen Recke

26/2/2021

EH 10.5

X_1, ..., X_5 er uafhængige reelle normalfordelte variable med samme varians σ^2 og $EX_1 = \alpha_1$, $EX_2 = EX_3 = \alpha_1 - \alpha_2$, $EX_4 = EX_5 = \alpha_1 + \alpha_2$, hvor α_1, α_2 er ukendte parametre.

Vi har altså givet en lineær normal model på $\mathbb{R}^5$ med den sædvanelige præcision, hvor middelværdiunderrummet har form

$$L = \{A\alpha \mid \alpha \in \mathbb{R}^2\}$$

Hvor $\alpha = (\alpha_1, \alpha_2)$ og designmatricen A er givet ved

```
A<- matrix(c(1,1,1,1,1,0,-1,-1,1,1),nrow=5)
A
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    1   -1
## [3,]    1   -1
## [4,]    1    1
## [5,]    1    1
```

a) Givet data x (nedenfor) bestemt maksimaliseringsestimatoren $\hat{\alpha}$.

Pr korollar 10.21 har vi $\hat{\alpha} = (A^T A)^{-1} A^T x$, det regnes i R

```
x <- matrix(c(-0.187,-1.731,-0.184,2.252,1.775),nrow=5) #data
alphahat <- solve(t(A)%*%A,t(A)%*%x) #Find MLE
```

Vi finder altså $\hat{\alpha} =$ (0.385, 1.4855).

b) Find en central estimator $\tilde{\sigma^2}$ og udregn for data

Ved korollar 10.21 og diskussionen efterfølgende fås at $\tilde{\sigma^2} = \frac{||x - A\hat{\alpha}||^2}{5-2}$, det regnes i R

```
sigmatildesq <- sum((x-A%*%alphahat)^2)/(5-2)
```

Vi finder altså $\tilde{\sigma^2}$ til 0.5731163.

c) Angiv den simultane fordeling af $\hat{\alpha}$ og $\tilde{\sigma^2}$

Pr korollar 10.21 fås at $\hat{\alpha} \sim \mathcal{N}(\alpha, \sigma^2 (A^T A)^{-1})$ og $\tilde{\sigma^2}$ er χ^2 fordelt med $5-2$ frihedsgrader og skalaparameter $\sigma^2/(5-2)$ (idet $\tilde{\sigma^2} = 5/(5-2)\hat{\sigma^2}$ og korollar 10.21 giver at $\hat{\sigma^2}$ er χ^2-fordelt med $5-2$ frihedsgrader og skalaparameter $\sigma^2/5$.

Derudover angiver korollar 10.21 også, at de er uafhængige. Så for at få den simultane fordeling kan man tage produktet af deres marginale fordelinger. Det er ikke en almindeligt navngivet fordeling.

d) Angiv et 95 % konfidensområde for α_1 og α_2.

Til det bruges eksempel 10.30 (specifikt ligning 10.41). Vi ser nemlig at, for $\psi_1 = (1,0)^T$ fås at $\psi_1^T \alpha = \alpha_1$ og $\psi_2 = (1,0)^T$ fås at $\psi_2^T \alpha = \alpha_2$.

```
psi1 <- c(1,0)
psi2 <- c(0,1)
se1 <- sqrt(t(psi1)%*%solve(t(A)%*%A)%*%psi1 * sigmatildesq) #Standardafvigelse for alpha1
se2 <- sqrt(t(psi2)%*%solve(t(A)%*%A)%*%psi2 * sigmatildesq) #Standardafvigelsen for alpha2
fval <- qf(0.95,1,3) #f-fraktilen
CI1 <- alphahat[1] + c(-1,1)*se1*sqrt(fval)
```

```
## Warning in c(-1, 1) * se1: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
CI2 <- alphahat[2] + c(-1,1)*se2*sqrt(fval)
```

```
## Warning in c(-1, 1) * se2: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
CI1
```

```
## [1] -0.6924509  1.4624509
```

```
CI2
```

```
## [1] 0.2808733 2.6901267
```

Vi bemærker at standardafvigelserne i se1 og se2 bare er lig $\tilde{\sigma^2}$ gange henholdsvis plads (1,1) og (2,2) i $(A^T A)^{-1}$, hvilket netop er approksimationen af variansen for henholdsvis $\hat{\alpha_1}$ og $\hat{\alpha_2}$, ved at bruge fordelingen fundet i c), nemlig $\hat{\alpha} \sim \mathcal{N}(\alpha, \sigma^2 (A^T A)^{-1})$. Konfidensintervallet man kommer frem til ved at bruge eksempel 10.30 ligner altså det man kender nemlig approksimation af standardafvigelsen gange kvadratroden af den passende F-fraktil (som er lig 97.5% fraktilen af t-fordelingen med 3 frihedsgrader, hvilket et måden i nok har set konfidensintervaller for en parameter på i SS og også den formel i var givet i en opgave i sidste uge).

HS11

Delspørgsmål 1) er lavet som en del af EH10.5

2) Prøv kommandoer og genfind estimater

```
#Kommandoer fra opgave
model <- lm(x ~ A-1)
model.matrix(model)
```

```
##   A1 A2
## 1  1  0
## 2  1 -1
## 3  1 -1
## 4  1  1
## 5  1  1
## attr(,"assign")
## [1] 1 1
```

```
summary(model)
```

```
##
## Call:
## lm(formula = x ~ A - 1)
##
## Residuals:
```

```
##       1       2       3       4       5
## -0.5720 -0.6305  0.9165  0.3815 -0.0955
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## A1   0.3850     0.3386   1.137   0.3381
## A2   1.4855     0.3785   3.924   0.0294 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.757 on 3 degrees of freedom
## Multiple R-squared:  0.8477, Adjusted R-squared:  0.7461
## F-statistic: 8.347 on 2 and 3 DF,  p-value: 0.05945
```

```
#Find estimater i modellen
coef(model)
```

```
##     A1     A2
## 0.3850 1.4855
```

```
summary(model)$sigma^2
```

```
## [1] 0.5731163
```

Vi genfinder vores designmatrix og estimater for parametre i modellen.

Bemærk at vi skriver $A-1$, når vi bruger lm kommandoen og ikke blot A. Det er fordi R automatisk tilføjer en intercept parameter (altså hvad der svarer til at tilføje en 1-søjle til designmatricen), hvilket vi ikke vil have. Man kan sige at vi jo allerede har det idet α_1 på den måde svarer til en intercept parameter, så modellen vil blive overparametriseret idet søjlerne da ikke er lineært uafhængige. Det kan R dog indse, så den vil fjerne en af de to når den skal lave udregningen og derved alligvel finde samme estimater (prøv evt. selv at køre koden og find designmatrix og estimater for at tjekke det), men der vil optræde en NA række i listen over koefficienter.

3) Sammenhæng mellem vcov(model) og $(A^TA)^{-1}$? Vi finder at vcov netop er lig $(A^TA)^{-1}\tilde{\sigma^2}$.

```
vcov(model)
```

```
##           A1        A2
## A1 0.1146233 0.0000000
## A2 0.0000000 0.1432791
```

```
solve(t(A)%*%A)*sigmatildesq
```

```
##           [,1]      [,2]
## [1,] 0.1146233 0.0000000
## [2,] 0.0000000 0.1432791
```

4) Sammenhæng mellem diagonalerne i vcov(model) og standard errors i summary(model)

Standard errors er lig kvadratroden af diagonalelementerne i vcov(model):

```
sqrt(vcov(model))
```

```
##           A1        A2
## A1 0.3385606 0.0000000
## A2 0.0000000 0.3785222
```

Hvis man sammenligner diagonalelementerne ovenfor med standard erros i summary i 2) ser man de er ens.

HS13

1) Indlæs data og fit kvadratisk regressionsmodel

```
#Indlæs data (med tidyverse)
library(readr)
paddy <- read_table2("paddy.txt")
```

```
## Parsed with column specification:
## cols(
##   days = col_double(),
##   yield = col_double()
## )
```

```
## Warning: 1 parsing failure.
## row col  expected    actual        file
##  12  -- 2 columns 3 columns 'paddy.txt'
```

```
#Fit kvadratisk model
daysSqr <- (paddy$days)^2
kvadReg <- lm(yield ~ days + daysSqr,data=paddy)
coef(kvadReg)
```

```
##  (Intercept)         days      daysSqr
## -1070.397689   293.482948    -4.535802
```

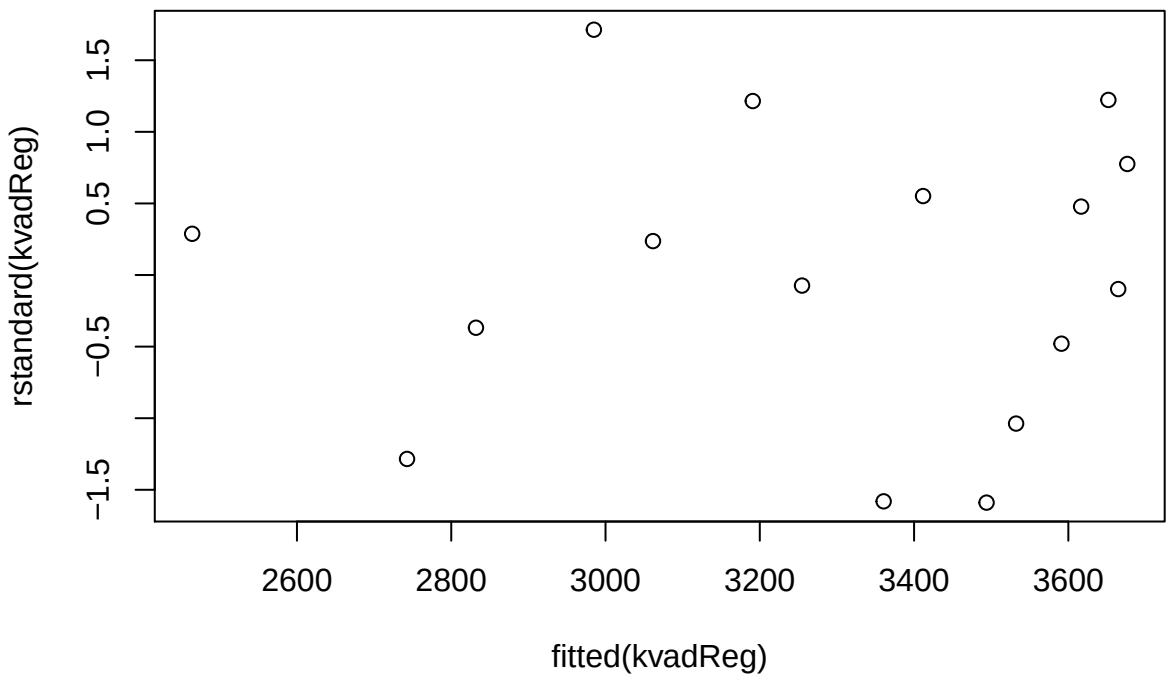
```
summary(kvadReg)$sigma^2 #Sigmatilde i anden
```

```
## [1] 41568.34
```

Vi genfinder estimaterne fra eksempel 11.10 i EH.

2) Lav tilhørende residualplot

```
plot(fitted(kvadReg),rstandard(kvadReg))
```



Der er ikke umiddelbart nogen sammenhæng mellem hvor stor fejlen er og hvor stor den fittede værdi er.

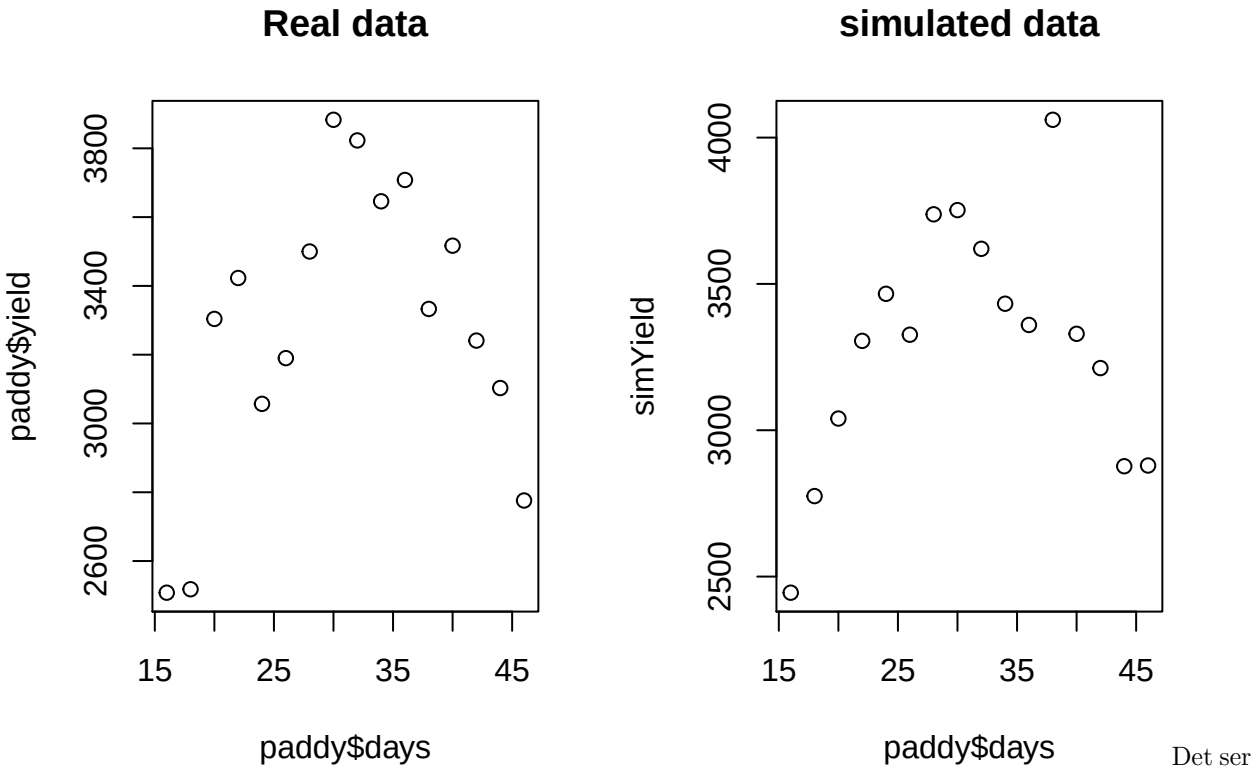
3) Antag at estimaterene fra 1) er de sande og simuler derfra, lav derefter (days,simYield) plot, se om det ser fornuftigt ud

```
xi0 <- fitted(kvadReg)  #Middelværdivektor
sigma0 <- summary(kvadReg)$sigma #Kvadratrod er sigmatilde^2, dvs standardafvigelse
simYield <- rnorm(16, xi0, sigma0) #Simulation fra modellen
```

Modellen er netop at X følger en normalfordeling med xi0 som middelværdi og sigma0 som standardafvigelse.

Vi Laver et (days,simYield) (altså med simuleret data) og sammenligner med det rigtige (days,simYield) plot for data.

```
par(mfrow=c(1,2))

plot(paddy$days,paddy$yield,main="Real data")
plot(paddy$days,simYield,main="simulated data")
```



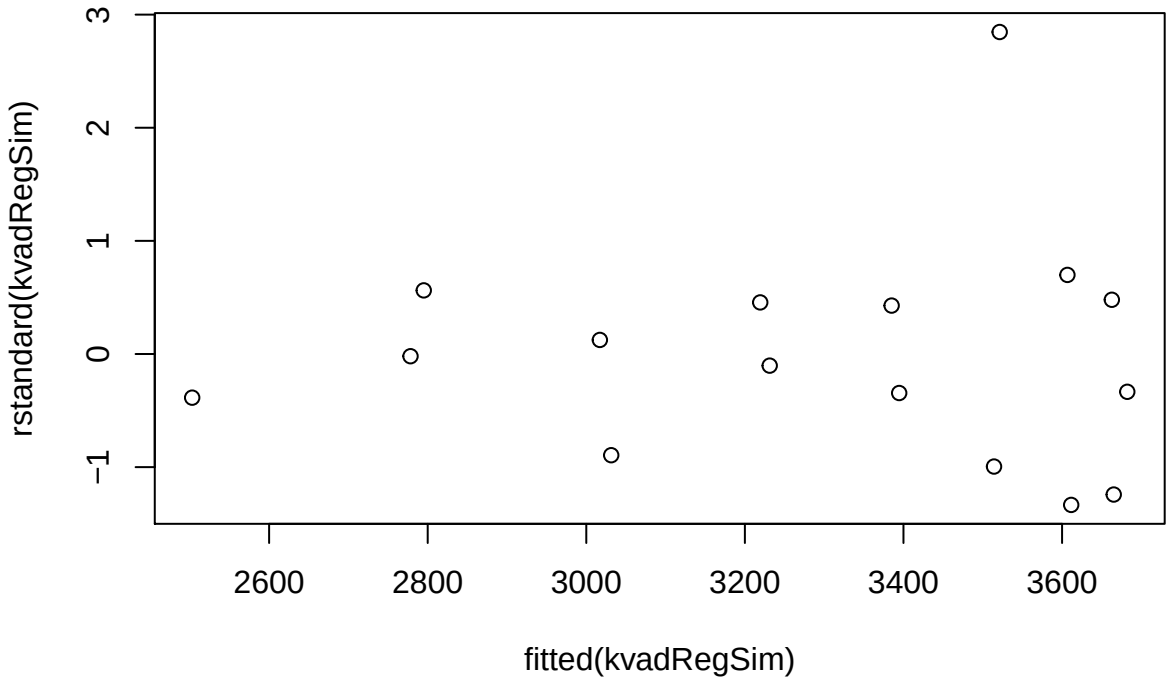
Det ser umiddelbart fornuftigt ud, man kan evt gøre det et par gange.

4) Fit regressionsmodel til simuleret data og lav residualplot

```
kvadRegSim <- lm(simYield ~ paddy$days + daysSqr)
coef(kvadRegSim)
```

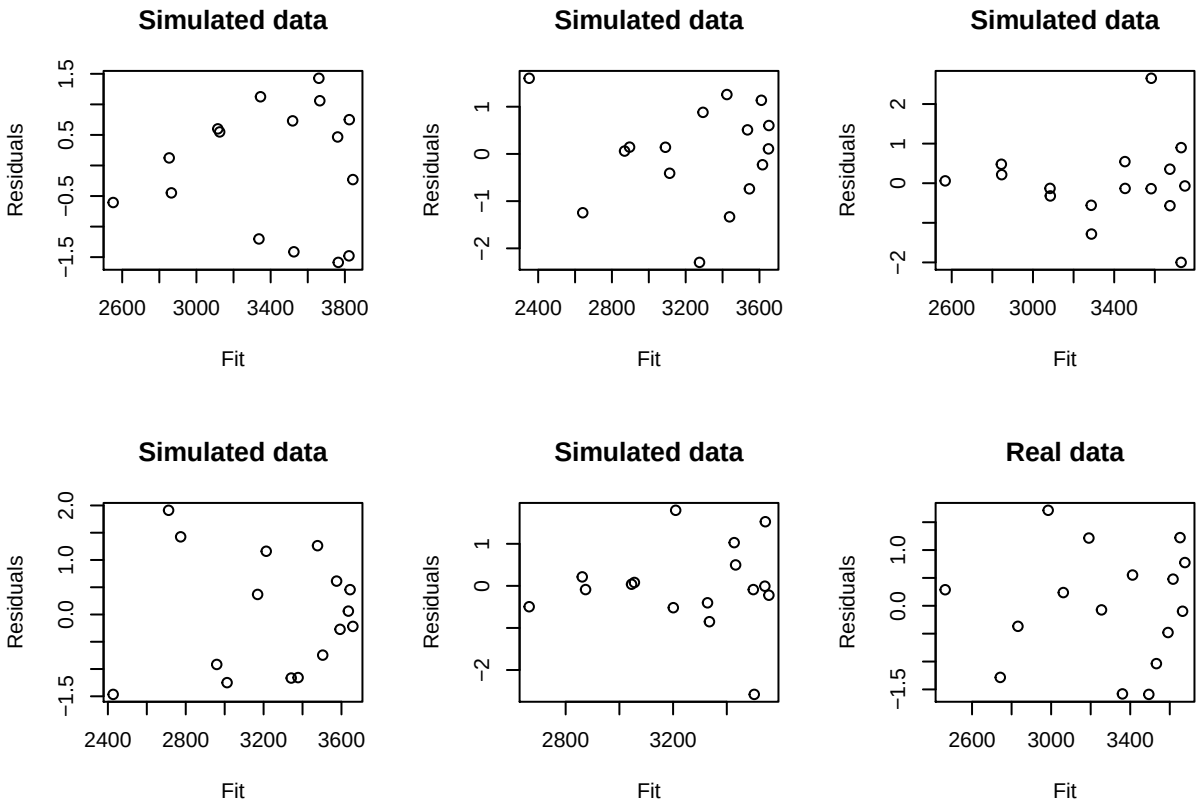
```
##  (Intercept)   paddy$days      daysSqr
## -1015.810748   293.032227    -4.569306
```

```
#Tilhørende residualplot
plot(fitted(kvadRegSim),rstandard(kvadRegSim))
```



5) Gentag 4) nogle gange og sammenlign med det rigtige residualplot

```
par(mfrow=c(2,3)) #Lav 5 simulerede residual plots
for (i in 1:5) {
  simYield <- rnorm(16, xi0, sigma0) #Simulation fra modellen
  kvadRegSim <- lm(simYield ~ paddy$days + daysSqr)
  plot(fitted(kvadRegSim),rstandard(kvadRegSim),main="Simulated data",xlab="Fit",ylab="Residuals")
}

plot(fitted(kvadReg),rstandard(kvadReg),main="Real data",xlab="Fit",ylab="Residuals") #real data
```



Umiddelbart ser residualplottet for ægte data ikke værre eller bedre ud end for simuleret data, så det virker til modellen er brugbar til at beskrive data.

6) Estimater for optimale høsttidspunkt og maksimalt udbytte

Vi bruger koden angivet i opgaven. Formlerne er præcis formlerne for toppunkt af en parabel, hvilket netop er den model vi har fittet data til, så hvis vi (som vi argumenterede for ovenfor) tror det er en rimelig model, vil toppunktsformlerne netop give fornuftige estimater for optimalt høsttidspunkt og maksimalt udbytte

```
est <- coef(kvadReg)
optDay <- -est[2]/2/est[3]
optYield <- - (est[2]^2 - 4*est[1]*est[3]) / 4 / est[3]

optDay
```

```
##     days
## 32.35183
```

```
optYield
```

```
##     days
## 3676.957
```

7) Hvorfor teorien i EH kap 9 og 10 ikke umiddelbart giver os fordelingen af optimalt høsttidspunkt

Høsttidspunktet er ikke en affin transformation af vores parameterestimater, hvilket er hvad teorien i kap 9 og 10 kan håndtere. Specifikt er de centrale værktøjer korollar 10.21, der giver hvordan MLE er normalfordelt og korollar 9.46 der viser hvordan en affin tranformation af en normalfordelt variabel igen er normalfordelt. Men det kan altså ikke bruges her.

8) + 9) Lav 1000 simulationer for optimalt høsttidspunkt

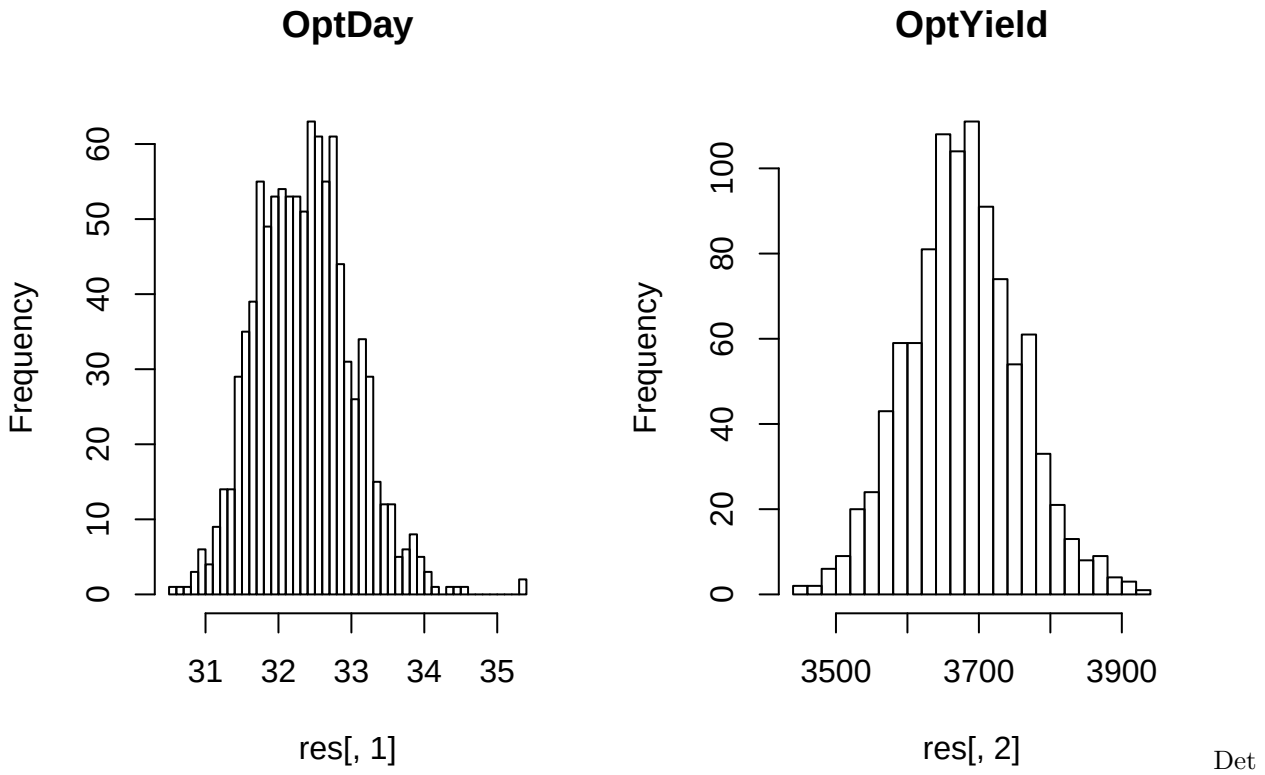
Vi tager udgangspunkt i koden angivet i opgaven

```
res <- matrix(NA, 1000, 2) # Initialiering af matrix for (i in 1:1000)
for (i in 1:1000) # Løkke
{
  simYield <- rnorm(16, xi0, sigma0) #Simulation
  simModel <- lm(simYield ~ days + daysSqr,data=paddy) #regression
  simEst <- coef(simModel) #Træk estimater ud
  simOptDay <- -simEst[2]/2/simEst[3] # Optimal dag
  simOptYield <- - (simEst[2]^2 - 4*simEst[1]*simEst[3]) / 4 / simEst[3] #optimal høst
  res[i,1] <- simOptDay  # Læg værdien paa plads (i,1)
  res[i,2] <- simOptYield
}
```

10) Histogrammer for de to estimater, hvilke type fordeling har de?

Vi prøver først bare at tegne histogrammerne

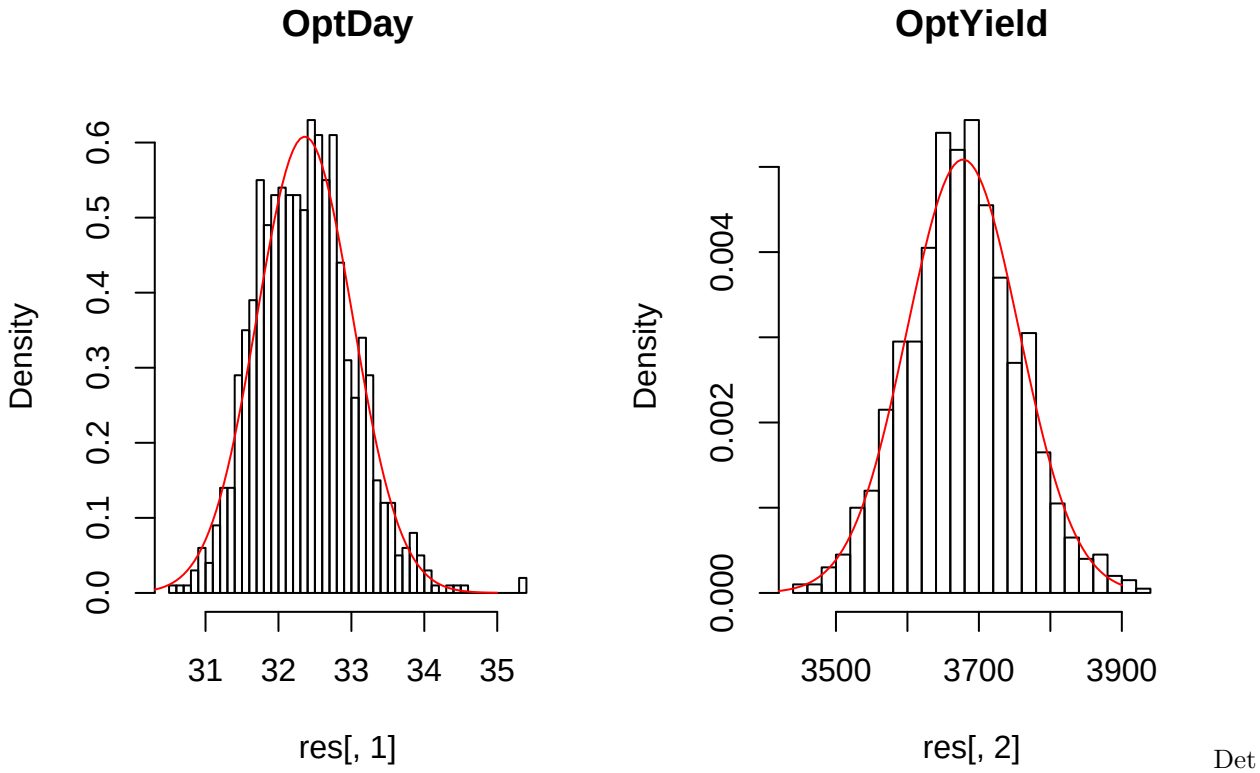
```
par(mfrow=c(1,2))
hist(res[,1],main="OptDay",breaks=40)
hist(res[,2],main="OptYield",breaks=30)
```



Det kunne godt ligne de er normalfordelte, vi prøver derfor at tegne kurven for den normalfordeling med samme middelværdi og varians som data for at se hvordan det passer

```
#Funktioner for hvordan normalfordelingerne skulle være:
fDay <- function(x) dnorm(x, mean = mean(res[,1]) , sd = sd(res[,1]))
fYield <- function(x) dnorm(x,mean=mean(res[,2]), sd = sd(res[,2]))

par(mfrow=c(1,2))
#Plot normalfordelingerne ovenpå
hist(res[,1],main="OptDay",breaks=40,prob=TRUE)
plot(fDay,30,35,add=TRUE, col="red")
```

```
hist(res[,2],main="OptYield",breaks=30,prob=TRUE)
plot(fYield,3400,3900,add=TRUE, col="red")
```



Det virker stadig som et okay bud, at de kunne være approksimativt normalfordelt.

11) Et approksimativt konfidensinterval for det optimale høsttidspunkt

Vi gennemgår to metoder. I og med vi fandt i 10) at de godt begge kunne ligne at være normalfordelt kan man altså regne et approksimativt 95 % konfidensinterval ved brug af den antagelse, $est \pm 1.96 \cdot sd$, hvor est angiver hhv estimatet for optimal dag og udbytte fundet i 6) og sd er standardafvigelsen, hvor vi altså ville bruge den fundet fra det simulerede data i 9), da hvis data rent faktisk var normalfordelt ville dette være et eksakt 95 % konfidensinterval.

```
#KI for optimal dag
KIday1 <- c(optDay - 1.96*sqrt(var(res[,1])),optDay + 1.96*sqrt(var(res[,1])))

#KI for yield
KIYield1 <- c(optYield - 1.96*sqrt(var(res[,2])),optYield + 1.96*sqrt(var(res[,2])))
```

En anden metode er direkte at bruge vores simulerede data og så tage 2.5% og 97.5% fraktilen derfra, så antager man ikke noget om fordelingen. Usikkerheden ligger nu i,at vi bruger vores simulerede data.

```
KIday2 <- quantile(res[,1],probs=c(0.025,0.975))
KIYield2 <- quantile(res[,2],probs=c(0.025,0.975))
```

Vi kan nu sammenligne intervallerne

```
KIday1
```

```
##     days     days
## 31.06558 33.63807
```

```
KIday2
```

```
##     2.5%    97.5%
## 31.20943 33.71625
```

```
KIYield1
```

```
##     days     days
## 3523.238 3830.676
```

```
KIYield2
```

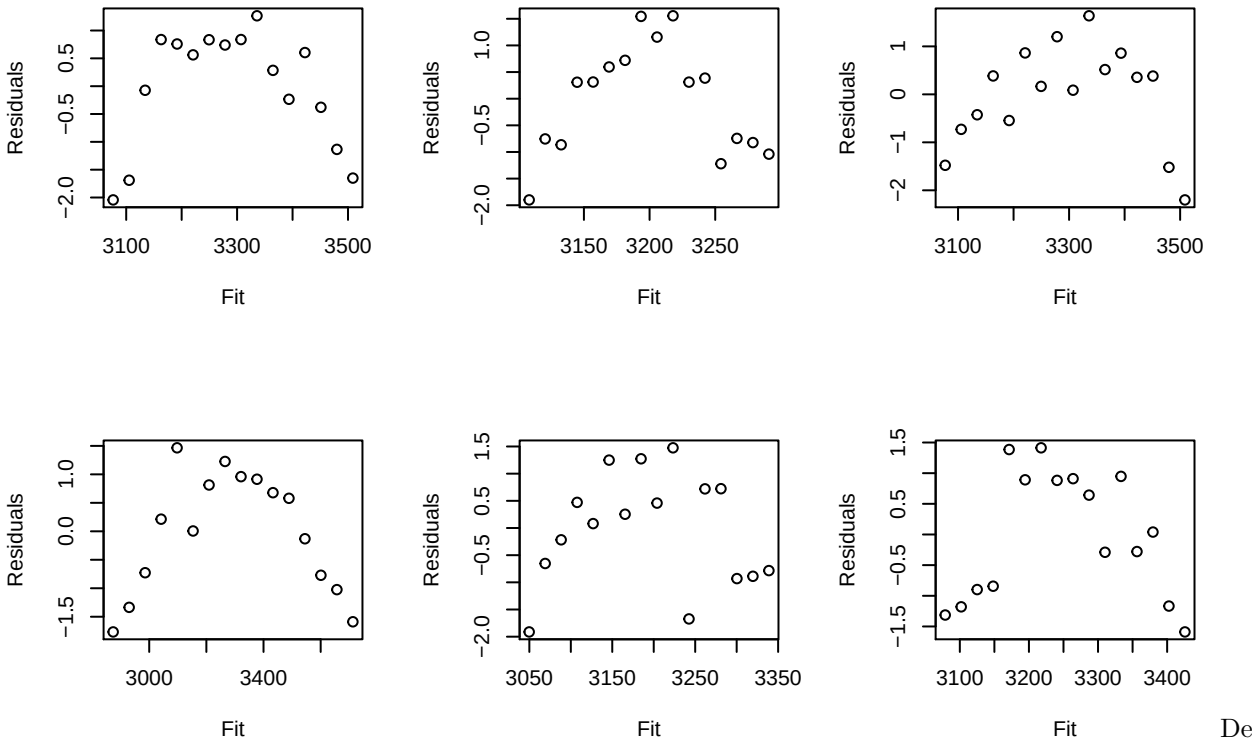
```
##     2.5%    97.5%
## 3528.826 3838.037
```

Intervallerne er ikke langt fra hinanden.

12) Hvis der er problemer med middelværdien. Simuler data fra kvadratisk model, fit til lineær og forklar residualplot

Vi gør det et par gange

```
par(mfrow=c(2,3))
for (i in 1:6) {
  simYield <- rnorm(16, xi0, sigma0)
  LinRegSim <- lm(simYield ~ days,data=paddy)

plot(fitted(LinRegSim),rstandard(LinRegSim) ,xlab="Fit",ylab="Residuals")
}
```



De ligner parabler. Der er altså en tydelig sammenhæng mellem størrelsen af den fittede værdi og hvor stor fejlen er.

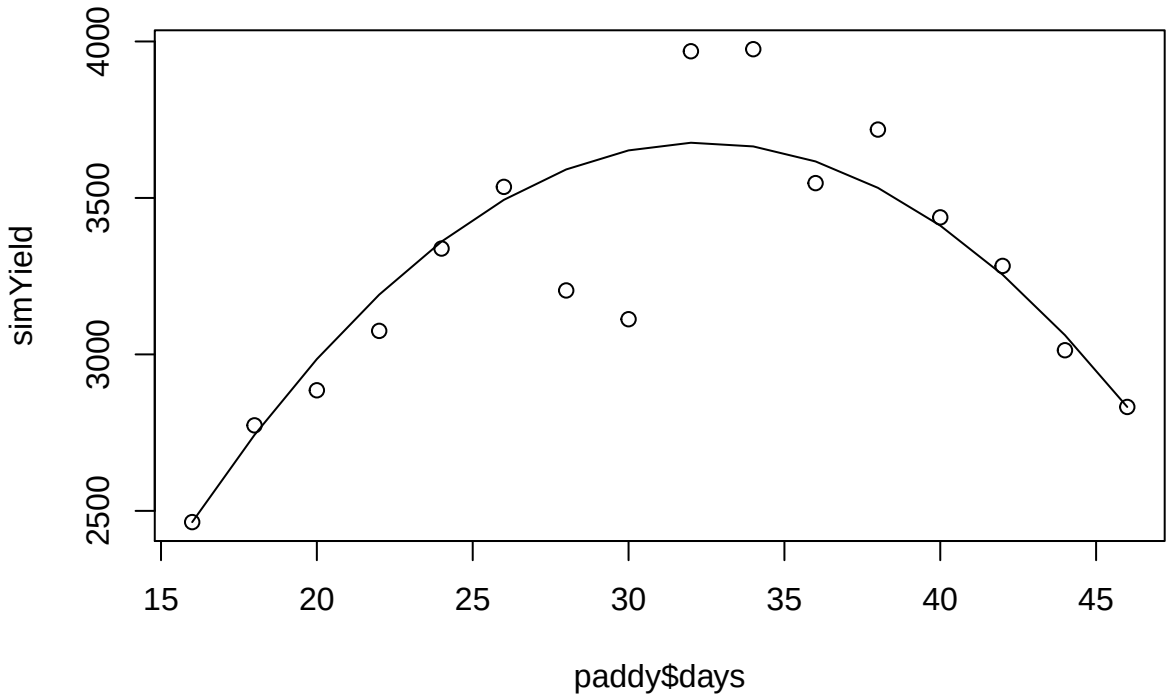
13) Når der er problemer med variansen. Kør givne kode, fit til kvadratisk model og tegn residual plot

Vi kører den givne kode

```
newSD <- 25*(15-abs(paddy$days-31))
newSD
```

```
##  [1]   0  50 100 150 200 250 300 350 350 300 250 200 150 100  50   0
```

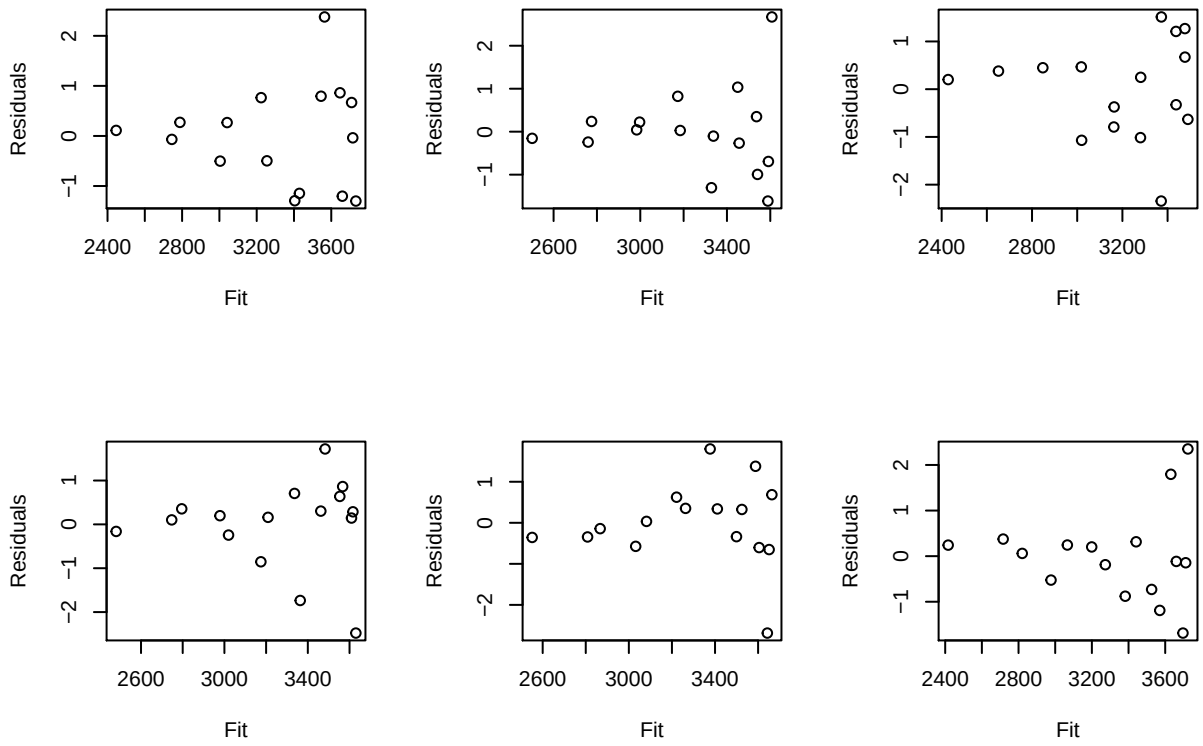
```
simYield <- rnorm(16, xi0, newSD)
plot(paddy$days, simYield)
points(paddy$days, xi0, type="l")
```



Vi simulerer altså fra en model, hvor vi har ladet variansen variere (der er stor varians for dagene i midten og lille for de første og sidste dage), men samme middelværdi som tidligere. Vi ser på plottet ovenfor at den model vi har fittet ligner at passe godt på data (modsat den lineære model vi prøvede i spørgsmålet før). Der er dog problemer med residualplottet igen, fordi vi med vilje har sørget for at vores modelantagelse om konstant varians ikke er opfyldt (se nedenfor).

Vi fitter nu til kvadratisk regressionsmodel og tegner residualplot

```
par(mfrow=c(2,3))
for (i in 1:6) {
  simYield <- rnorm(16, xi0, newSD)
  kvadRegSim <- lm(simYield ~ days + daysSqr,data=paddy)
  plot(fitted(kvadRegSim),rstandard(kvadRegSim) ,xlab="Fit",ylab="Residuals")
}
```



Vores residualplots ligner nu trumpeter. Der er altså større fejl jo større værdien af de fittede værdier er. Det giver god mening med den variansstruktur vi valgte. Vi lod der nemlig være stor varians ved de midterste dage, hvilket jo netop også er der høstudbyttet er størst. Men som nævnt viser det altså at vores modelantagelse om konstant varians ikke holder i dette tilfælde.

Ekstraopgave om regression og konfidensintervaller

Vi betragter modellen inspireret af EH eksempel 11.13

$$R_i = \alpha + \beta M_i + \gamma r_i + \varepsilon_i$$

Hvor $\varepsilon_1, ..., \varepsilon_N$ alle er uafhængige $\mathcal{N}(0, \sigma^2)$ fordelte. Lad $\hat{\theta} = (\hat{\alpha}, \hat{\beta}, \hat{\gamma})^T$.

Vi betragter altså en lineær normal model

Spørgsmål 1: Vi skal finde variansmatricen for $\hat{\theta}$

Pr korollar 10.21 har vi $\hat{\theta} \sim \mathcal{N}(\theta, (A^T A)^{-1}\sigma^2)$, hvor A er designmatricen, dvs

```
CAPM <- read_csv("CAPM.csv") #Indlæs data
```

```
## Parsed with column specification:
## cols(
##   Month = col_character(),
##   r = col_double(),
##   M = col_double(),
##   R = col_double()
## )
```

```
A <- matrix(c(rep(1,20),CAPM$M,CAPM$r),nrow=20)
```

Vi kan altså udregne variansmatricen direkte fra designmatricen A. Dog er σ^2 en ukendt parameter, så variansmatricen er givet som følgende gange σ^2:

```r
solve(t(A)%*%A)
```

```
##              [,1]         [,2]         [,3]
## [1,]  0.929462053 -0.005258217 -2.091840812
## [2,] -0.005258217  0.002778923 -0.004321725
## [3,] -2.091840812 -0.004321725  5.078616443
```

Hvis vi vil udregne et approksimation for variansmatricen kan vi regne $\tilde{\sigma^2}$:

```r
N <- 20
k <- 3
X <- CAPM$R
thetahat <- solve(t(A)%*%A,t(A)%*%X)
sigmatildesq <- sum((X-A%*%thetahat)^2)/(N-k)

#Approksimativ variansmatrix
Appvarians <- solve(t(A)%*%A)*sigmatildesq
Appvarians
```

```
##             [,1]       [,2]        [,3]
## [1,]  38.0005763 -0.2149795 -85.5238319
## [2,]  -0.2149795  0.1136148  -0.1766915
## [3,] -85.5238319 -0.1766915 207.6366120
```

Spørgsmål 2: Fordeling af $\hat{\beta}+\hat{\gamma}$

Opgaven efterspørger eksplicit at man viser $\hat{\beta}+\hat{\gamma} \sim \mathcal{N}(\beta+\gamma, se^2)$ med $se^2 = a\sigma^2$.

Idet

$$\hat{\beta}+\hat{\gamma} = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \\ \hat{\gamma} \end{pmatrix}$$

Følger det af korollar 9.46 (og fordelingen af $\hat{\theta}$ fundet i a)) at

$$\hat{\beta}+\hat{\gamma} \sim \mathcal{N}(\beta+\gamma, \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} (A^TA)^{-1} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \sigma^2)$$

Altså er $\begin{pmatrix} 0 & 1 & 1 \end{pmatrix} (A^TA)^{-1} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$ det a som efterspørges i opgaveformuleringen. Vi kan nu regne a og en approksimation af $se^2 = a\sigma^2$ ved at erstatte σ^2 med det centrale variansestimat ligesom i a)

```r
psi <- matrix(c(0,1,1),nrow=1)
a <- psi %*% solve(t(A)%*%A) %*% t(psi)
setildesq <- a*sigmatildesq
setilde <- sqrt(setildesq)
setilde
```

```
##          [,1]
## [1,] 14.40128
```

Vi har derved fundet det efterspurgte $\tilde{se}$.

Spørgsmål 3: De forskellige konfidensintervaller og ideen i dem

Man kan lave approksimativt konfidensinterval for $\beta+\gamma$ ved

$$\hat{\beta}+\hat{\gamma} \pm 1.96 \cdot \tilde{se}$$

Det giver mening, da vi i spørgsmålet ovenfor fandt at $\hat{\beta}+\hat{\gamma} \sim \mathcal{N}(\beta+\gamma, se^2)$. Så hvis vi havde den sande værdi af se så ville et eksakt konfidensinterval være $\hat{\beta}+\hat{\gamma} \pm 1.96 \cdot \tilde{se}$, idet 1.96 er 97.5 % fraktilen for standardnormalfordelingen. Men vi kender ikke den sande værdi se, så ved at indsætte en approksimation for den får vi derved også at approksimativt konfidensinterval.

Intervallet givet i eksempel 10.30 (ligning 10.41) er er derimod eksakt og ved at sætte $\psi = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$ fås.

$$\hat{\beta}+\hat{\gamma} \pm \sqrt{z_5} \sqrt{\begin{pmatrix} 0 & 1 & 1 \end{pmatrix} (A^T A)^{-1} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \tilde{\sigma^2}}$$

Hvor z_5 er 95% fraktilen i F-fordelingen med $(1, N-k)$ frihedsgrader (hvor i dette tilfælde er $N=20$ og $k=3$). Vi bemærker dog, at det der står under den store kvadratrod netop er lig $\tilde{se}$ fundet i opgave 2), så den eneste forskel mellem de to intervaller er altså hvorvidt fraktilen tages fra en standardnormalfordelingen eller kvadratroden af en F-fordeling med passende frihedsgrader (kvadratroden af F-fordelingen er faktisk præcis 97.5% fraktilen for t-fordelingen med $N-k$ frihedsgrader, som måske er den måde i oftere har set dette eksakte interval angivet på).

For store N vil disse fraktiler dog stortset være ens. I vores tilfælde hvor vi altså har $N-k=17$ er der stadig lidt forskel

```
sqrt(qf(0.95,1,N-k))
```

```
## [1] 2.109816
```

Spørgsmål 4: Udregning af konfidensintervallerne og hvad de siger om hypotesen $\beta+\gamma=1$

```
KI1 <- psi%*%thetahat + c(-1,1)*1.96*sqrt(setildesq)
```

```
## Warning in c(-1, 1) * 1.96 * sqrt(setildesq): Recycling array of length 1 in vector-array arithmetic
##   Use c() or as.vector() instead.
```

```
## Warning in psi %*% thetahat + c(-1, 1) * 1.96 * sqrt(setildesq): Recycling array of length 1 in arra
##   Use c() or as.vector() instead.
```

```
KI2 <- psi%*%thetahat + c(-1,1)*sqrt(setildesq)*sqrt(qf(0.95,1,N-k))
```

```
## Warning in c(-1, 1) * sqrt(setildesq): Recycling array of length 1 in vector-array arithmetic is dep
##   Use c() or as.vector() instead.
```

```
## Warning in psi %*% thetahat + c(-1, 1) * sqrt(setildesq) * sqrt(qf(0.95, : Recycling array of length
##   Use c() or as.vector() instead.
```

```
KI1
```

```
## [1]  0.949856 57.402871
```

```
KI2
```

```
## [1] -1.20768 59.56041
```

Begge konfidensintervaller indeholder 1, så vi kan på signifikansniveau 95% ikke afvise hypotesen $\beta+\gamma=1$.

EH 11.3

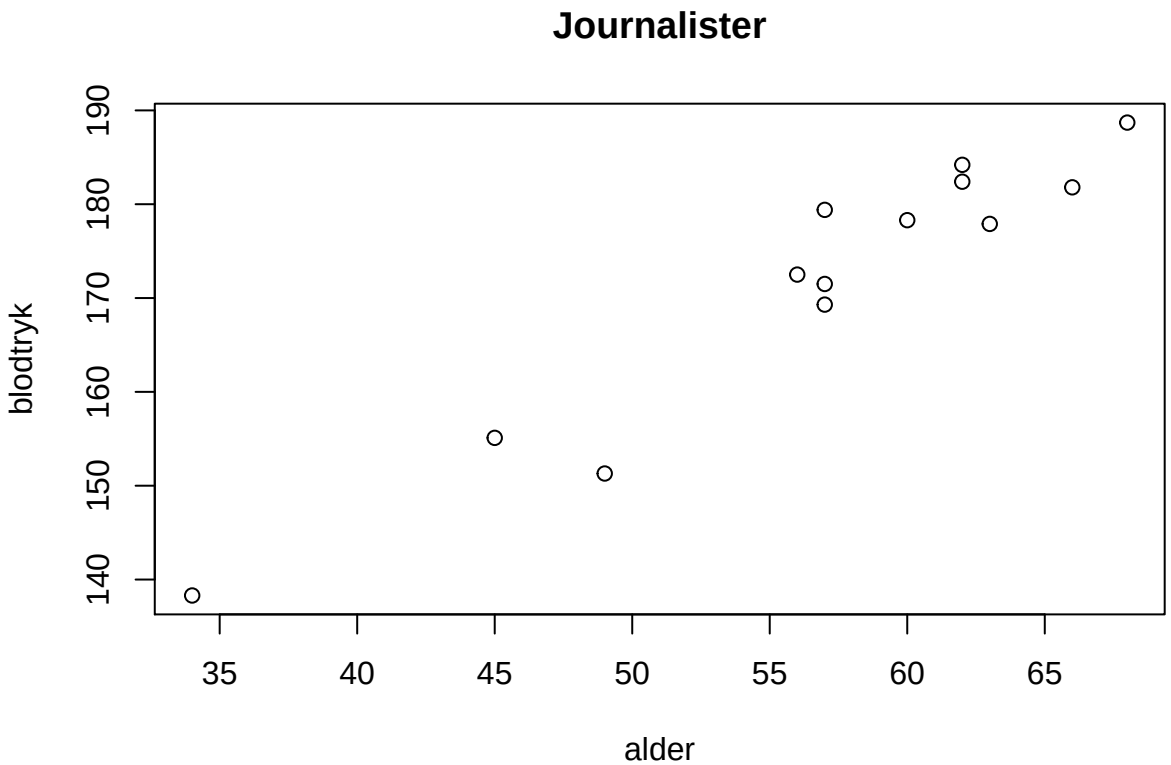
a) Plot blodtryk mod alder

```
blodtryk <- read_table2("EHopg11dot3.txt") #Indlæs data
```

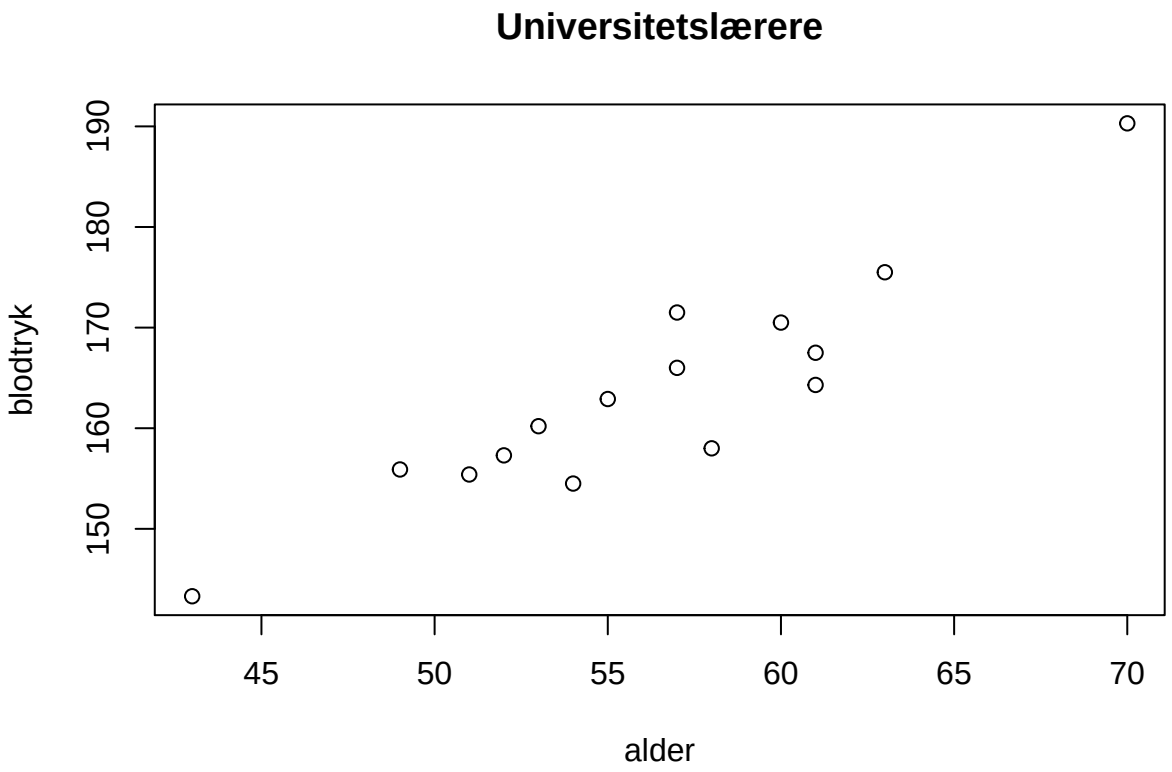
```
## Parsed with column specification:
## cols(
##   Alder = col_double(),
##   Blodtryk = col_double(),
##   Fag = col_character()
## )
```

```
## Warning: 19 parsing failures.
## row col  expected    actual              file
##   2  -- 3 columns 4 columns 'EHopg11dot3.txt'
##   3  -- 3 columns 4 columns 'EHopg11dot3.txt'
##   4  -- 3 columns 4 columns 'EHopg11dot3.txt'
##   5  -- 3 columns 4 columns 'EHopg11dot3.txt'
##   6  -- 3 columns 4 columns 'EHopg11dot3.txt'
## ... ... ......... ......... ..................
## See problems(...) for more details.
```

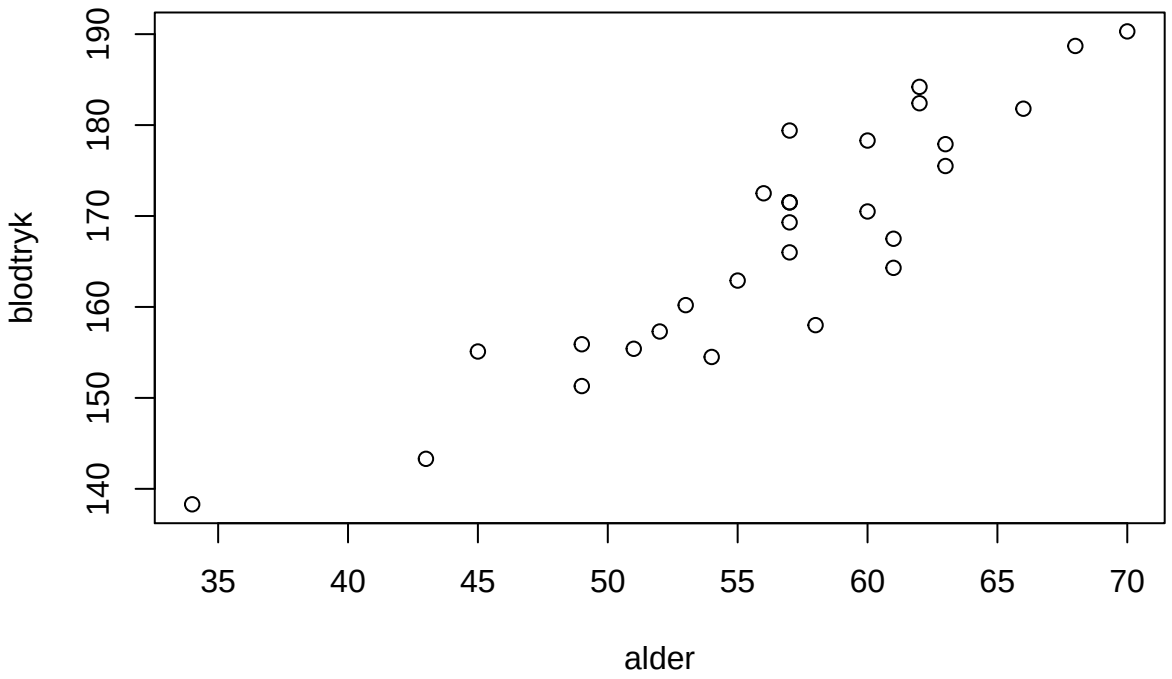
```
blodtrykJ <- subset(blodtryk,Fag=="J") #Subset af data for Journalister
blodtrykU <- subset(blodtryk,Fag=="U") #Subset af data for universitetslærere

#Plot hver for sig
plot(blodtrykJ$Alder,blodtrykJ$Blodtryk,main="Journalister",xlab="alder",
     ylab="blodtryk")
```




```
plot(blodtrykU$Alder,blodtrykU$Blodtryk,main="Universitetslærere",xlab="alder",
     ylab="blodtryk")
```




```
#Plot samlet
plot(blodtryk$Alder,blodtryk$Blodtryk,main="Samlet",xlab="alder",
     ylab="blodtryk")
```



Det virker rimeligt både for det samlede data og for grupperne hver for sig at opstille en lineær regressionsmodel.

b) Lineær regressionsmodel kun for gruppen med journalister

Vores model er altså den linære normale model hvor B_i (blodtryk for journalisterne) er uafhængige, normalfordelte med samme varians σ^2 og at $EB_i = \alpha + \beta a_i$, hvor a_i angiver alderen. (se afsnit 11.1)

Det kan regnes i R ved følgende, hvor vi kan aflæse estimaterne:

```
LinregJ <- lm(Blodtryk ~ Alder,data=blodtrykJ)
coefficients(LinregJ)
```

```
## (Intercept)       Alder
##   84.995220    1.529568
```

```
summary(LinregJ)$sigma^2
```

```
## [1] 18.1495
```

c) Lineær regressionsmodel kun for gruppen med universitetslærere Vores model er altså den linære normale model hvor B_i (blodtryk for universitetslærene) er uafhængige, normalfordelte med samme varians σ^2 og at $EB_i = \alpha + \beta a_i$, hvor a_i angiver alderen. (se afsnit 11.1)

Det kan regnes i R ved følgende, hvor vi kan aflæse estimaterne:

```
LinregU <- lm(Blodtryk ~ Alder,data=blodtrykU)
coefficients(LinregU)
```

```
## (Intercept)       Alder
##   75.629862    1.562384
```

```
summary(LinregU)$sigma^2
```

```
## [1] 19.44954
```

d) Lineær regressionsmodel for de to grupper hvor de estimeres på en gang med hver deres rette linje Det svarer altså til den linære normale model hvor B_i (blodtryk) er uafhængige, normalfordelte med samme varians σ^2 og

$$EB_i = \alpha_J 1_J(i) + \beta_J 1_J(i) a_i + \alpha_U 1_U(i) + \beta_U 1_U(i) a_i$$

Hvor a_i angiver alderen og $1_J(i)$ er 1 hvis person i er journalist og 0 ellers og tilsvarende for $1_U(i)$.

Hvilket svarer til følgende designmatrix (idet vi er heldige og data er givet sådan at de første 13 observationer er journalister og de sidste 15 er universitetslærere)

```
A <- matrix(c(rep(1,13),rep(0,15),blodtrykJ$Alder,rep(0,15),rep(0,13),rep(1,15),
              rep(0,13),blodtrykU$Alder),nrow=28)
A
```

```
##       [,1] [,2] [,3] [,4]
##  [1,]    1   68    0    0
##  [2,]    1   62    0    0
##  [3,]    1   49    0    0
##  [4,]    1   62    0    0
##  [5,]    1   34    0    0
##  [6,]    1   66    0    0
##  [7,]    1   45    0    0
##  [8,]    1   60    0    0
##  [9,]    1   57    0    0
## [10,]    1   57    0    0
## [11,]    1   63    0    0
## [12,]    1   56    0    0
```

```
## [13,]    1   57    0    0
## [14,]    0    0    1   55
## [15,]    0    0    1   49
## [16,]    0    0    1   60
## [17,]    0    0    1   54
## [18,]    0    0    1   58
## [19,]    0    0    1   51
## [20,]    0    0    1   43
## [21,]    0    0    1   52
## [22,]    0    0    1   53
## [23,]    0    0    1   61
## [24,]    0    0    1   61
## [25,]    0    0    1   57
## [26,]    0    0    1   57
## [27,]    0    0    1   70
## [28,]    0    0    1   63
```

Hvorfor vi derfor igen bare kan bruge lm til at udføre lineær regression

```
LinregSamlet <- lm(blodtryk$Blodtryk ~ A-1)
coefficients(LinregSamlet)
```

```
##        A1        A2        A3        A4
## 84.995220  1.529568 75.629862  1.562384
```

```
summary(LinregSamlet)$sigma^2
```

```
## [1] 18.85369
```

Vi genfinder estimaterne for hældning og intercept for de to grupper hver for sig. Forskellen på b)+c) og d) er altså hvorvidt man antager de har en fælles ukendt varians eller tillader de to grupper hver deres varians.

En anden metode i R er, at bruge $*$ notationen for interaktion, som indgår senere i kurset.

```
linregSamletv2 <-  lm(Blodtryk ~ Fag*Alder - 1, blodtryk)
model.matrix(linregSamletv2)
```

```
##    FagJ FagU Alder FagU:Alder
## 1     1    0    68          0
## 2     1    0    62          0
## 3     1    0    49          0
## 4     1    0    62          0
## 5     1    0    34          0
## 6     1    0    66          0
## 7     1    0    45          0
## 8     1    0    60          0
## 9     1    0    57          0
## 10    1    0    57          0
## 11    1    0    63          0
## 12    1    0    56          0
## 13    1    0    57          0
## 14    0    1    55         55
## 15    0    1    49         49
## 16    0    1    60         60
## 17    0    1    54         54
## 18    0    1    58         58
## 19    0    1    51         51
## 20    0    1    43         43
```

```
## 21    0    1    52         52
## 22    0    1    53         53
## 23    0    1    61         61
## 24    0    1    61         61
## 25    0    1    57         57
## 26    0    1    57         57
## 27    0    1    70         70
## 28    0    1    63         63
## attr(,"assign")
## [1] 1 1 2 3
## attr(,"contrasts")
## attr(,"contrasts")$Fag
## [1] "contr.treatment"
```

```
coefficients(linregSamletv2)
```

```
##        FagJ        FagU       Alder  FagU:Alder
## 84.99521966 75.62986190  1.52956813  0.03281584
```

```
summary(linregSamletv2)$sigma^2
```

```
## [1] 18.85369
```

Bemærk at R bruger en anden parametrisering af middelværdiunderrummet end vi gjorde da vi opskrev modellen og lavede vores tilhørende designmatrix. R parametriserer med kontraster, så for at genfinde estimatet for hældningen for universitetslærene skal de sidste to parametre i oversigten lægges sammen.

e) Fælles lineær regressionsmodel, hvor begge gruppe estimeres med samme rette linje Vores model er altså den linære normale model hvor B_i (blodtryk) er uafhængige, normalfordelte med samme varians σ^2 og at $EB_i = \alpha + \beta a_i$, hvor a_i angiver alderen. (se afsnit 11.1)

Det kan regnes i R ved følgende, hvor vi kan aflæse estimaterne:

```
Linreg <- lm(Blodtryk ~ Alder,data=blodtryk)
coefficients(Linreg)
```

```
## (Intercept)       Alder
##   79.660299    1.552729
```

```
summary(Linreg)$sigma^2
```

```
## [1] 32.53739
```

Jeg bemærkede at det måske var underligt at variansen nu er højere, men det giver jo god mening idet vi har mindre fleksibilitet i vores model idet vi antager at journalister og universitetslærene kan modelleres med samme rette linje.