Matematisk Statistik: Modelbaseret Inferens

Cross validation

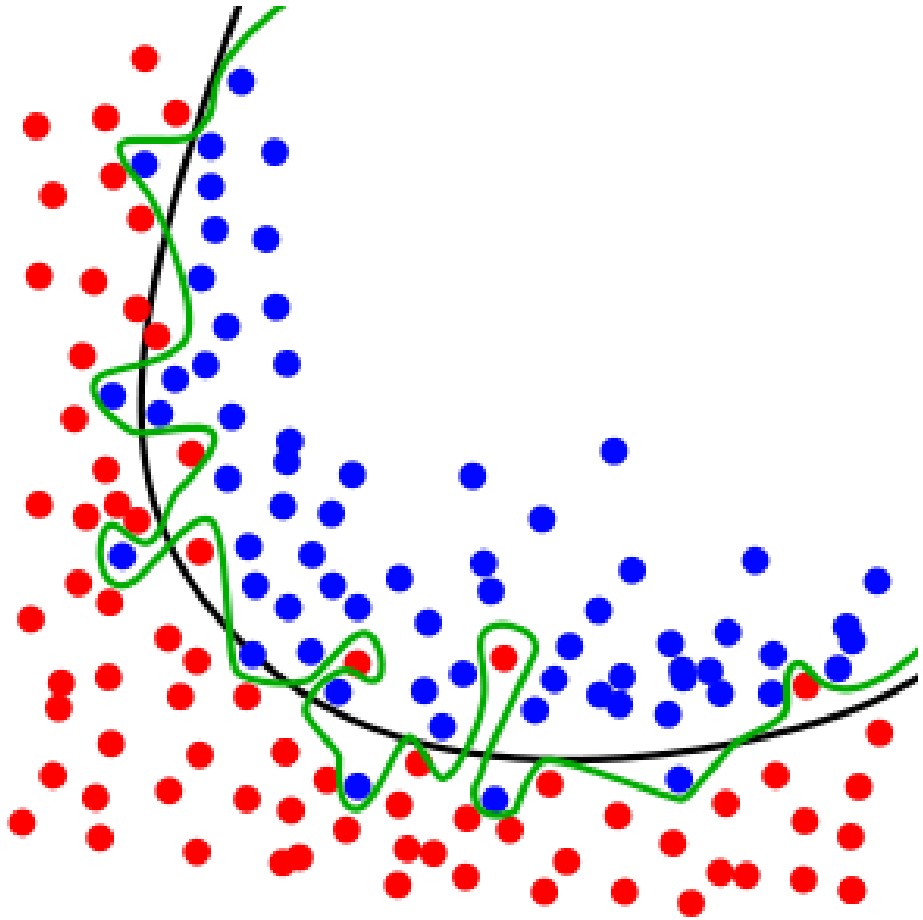
Jens Ledet Jensen

I dag

Multipel regression med mange forkarende variable

- Hvordan undgår vi overfitting ?
- Hvordan vælger vi den bedste multiple regressionsmodel ?

Illustration af overfitting



Grønne kurve modellerer de tilfældige udsving i data
ikke god til efterfølgende prædiktion

Model: $X_i \sim N(\beta_1 t_{i1} + \cdots + \beta_k t_{ik}, \sigma^2)$

$$\hat{\boldsymbol{\beta}} = (\mathbf{H}^\top \mathbf{H})^{-1} \mathbf{H}^\top \mathbf{X} \sim N(\boldsymbol{\beta}, \sigma^2 (\mathbf{H}^\top \mathbf{H})^{-1})$$

Backward selektion: starte med alle k led, teste led væk successivt

Forward selektion: starte uden led og addere led successivt

Problem: kan ikke stole på $s^2(M)$ til at udvælge model

Bruge prædiktionsvarians som "målestok"

$\hat{\beta}$ estimeres ud fra data (træningsdata)

Hvor gode er vi til at forudsige respons $\tilde{X}$ hørende til nye værdier $\tilde{t}_1, \ldots \tilde{t}_k$

Prædiktionsvarians: $E\{(\tilde{X} - (\hat{\beta}_1\tilde{t}_1 + \cdots + \hat{\beta}_k\tilde{t}_k))^2|\text{data}\}$

$= E\{(\tilde{X} - (\beta_1\tilde{t}_1 + \cdots + \beta_k\tilde{t}_k) - ((\hat{\beta}_1 - \beta_1)\tilde{t}_1 + \cdots + (\hat{\beta}_k - \beta_k)\tilde{t}_k))^2|\text{data}\}$

$= \sigma^2 + \{(\hat{\beta}_1 - \beta_1)\tilde{t}_1 + \cdots + (\hat{\beta}_k - \beta_k)\tilde{t}_k\}^2$

Ikke kun interesseret i én $\mathbf{t}$-værdi: tage "middelværdi" over $\mathbf{t}$

Problem: kender ikke β, så hvad gør vi?

Overfitting: eksempel

Sande model: $E(X_i) = \alpha + \beta_1 t_i$

Fitter model: $E(X_i) = \alpha + \beta_1 t_i + \beta_2 t_i^2 + \cdots + \beta_k t_i^k$

Typisk: jo større k er jo bedre et fit får vi: $s(M_k)$ er lille

hvis $k = n$ er $s(M_k) = 0$!

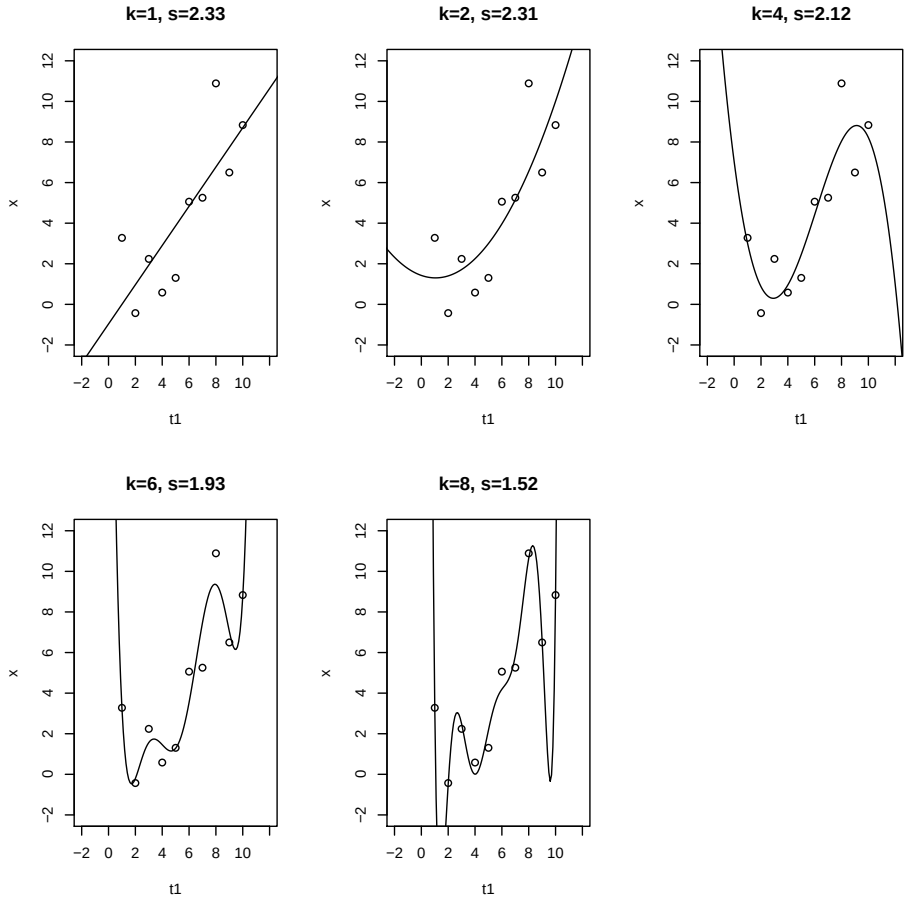
Hvis vi overfitter giver dette typisk en dårligere prediktor:

$E\{(X_{\text{ny}} - \text{Prediktor}(t_{\text{ny}}))^2 | \text{data}\}$ bliver større

k=1	k=2	k=4	k=6	k=8
2.3	3.2	4.2	5.9	14.4

For viste data, t_{ny} uniform
Kender sande β

Overfitting: eksempel



Prøv selv i R

Kør koden nogel gange og se at det sidste spredningsskøn typisk er mindre end det første

```
t=c(1:10)

x=rnorm(10)


c(summary(lm(x~poly(t,2)))$sigma,
summary(lm(x~poly(t,6)))$sigma,
summary(lm(x~poly(t,8)))$sigma)
```

Tage middelværdi af prædiktionsvarians mht $\hat{\beta}_1, \ldots, \hat{\beta}_k$ (det er denne der bruges, når vi laver prædiktionsintervaller)

$$E((\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\mathsf{T}\tilde{\mathbf{t}})^2 = \sigma^2\tilde{\mathbf{t}}^\mathsf{T}(\mathbf{H}^\mathsf{T}\mathbf{H})^{-1}\tilde{\mathbf{t}}$$

Eksempel: $\mathbf{H} = \begin{pmatrix} 1 & t_1 \\ \vdots & \\ 1 & t_n \end{pmatrix}$: $\quad \sigma^2(\frac{1}{n} + \frac{(\bar{t}-\tilde{t})^2}{\mathrm{SSD}_t})$

Eksempel: $\mathbf{H}^\mathsf{T}\mathbf{H} = \mathrm{diag}(w_1, \ldots, w_k)$:

$\sigma^2 \sum_{j=1}^{k} \frac{\tilde{t}_j^2}{w_j}$, stiger med k uanset om $\beta_k = 0$

overfitting!

Hvis $\beta_{k_0+1} = \cdots = \beta_k = 0$ og vi kun bruger $t_1, \ldots, t_{k_0}$:

$$\sigma^2 \sum_{j=1}^{k_0} \frac{\tilde{t}_j^2}{w_j}$$

Hvis vi kun bruger $t_1, \ldots, t_{k_0}$ men $\beta_{k_0+1}, \ldots, \beta_k$ er ikke nul:

$$\sigma^2 \sum_{j=1}^{k_0} \frac{\tilde{t}_j^2}{w_j} + (\beta_{k_0+1}\tilde{t}_{K_0+1} + \cdots + \beta_k \tilde{t}_k)^2$$

Prædiktionsvariansen vil falde sålænge vi inkluderer relevante led, men stige når vi inkuderer irrelevante led

Beregne prædiktionsvarians ud fra et testsæt $(\tilde{t}_{i1}, \ldots, \tilde{t}_{ik}, \tilde{x}_i)$ $i = 1, \ldots, m$

Skøn over prædiktionsvariansen: $\frac{1}{m} \sum_{i=1}^{m} (\tilde{x}_i - (\hat{\beta}_1 \tilde{t}_1 + \cdots + \hat{\beta}_k \tilde{t}_k))^2$

MEN: typisk har vi ikke et testsæt til rådighed

Alternativ: Dele oprindelige datasæt op i et "træningssæt" og et "testsæt"

crossvalidation

Træningssæt:

Data $x_1, \ldots, x_n$ bruges til at estimere model

$$x_i \sim\sim N(\alpha + \beta_1 t_{1i} + \cdots + \beta_k t_{ki}, \sigma^2)$$

$t_1, \ldots, t_k$: forklarende variable (regressionsvariable)

Testsæt: $\tilde{x}_i, \tilde{t}_{1i}, \ldots, \tilde{t}_{ki}$, $i = 1, \ldots, m$

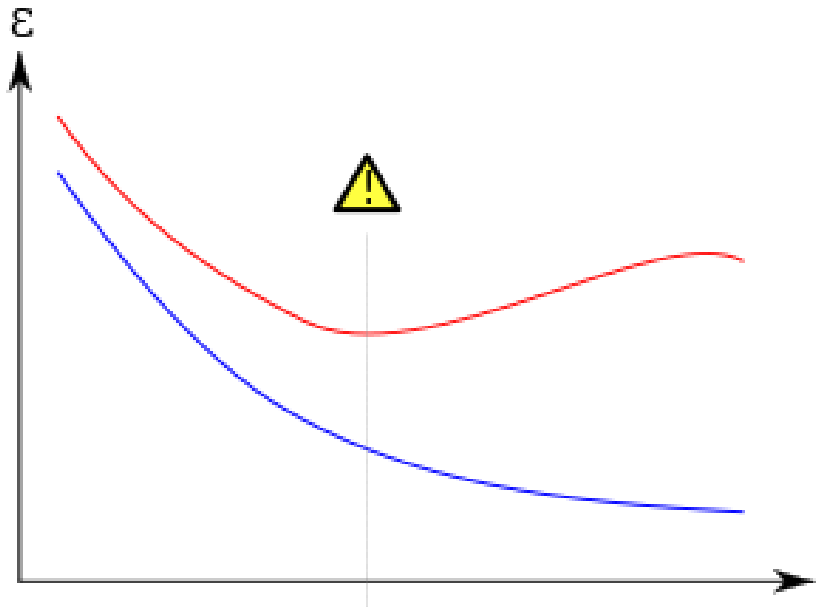
data der IKKE blev brugt til at finde skøn $\hat{\alpha}$, $\hat{\beta}$

Beregn prædikterede værdier $\hat{\xi}_i^P = \hat{\alpha} + \hat{\beta}_1 \tilde{t}_{1i} + \cdots + \hat{\beta}_k \tilde{t}_{ki}$

Skøn over prædiktionsspredning: $s_P = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (\tilde{x}_i - \hat{\xi}_i^P)^2}$

Illustration af overfitting

1.-akse: antal variable i model; 2.-akse: hvor godt passer model til data



Blå kurve viser bedre og bedre tilpasning til trænings-data ved inkludering af flere variable

Rød kurve viser prædiktionsspredning på test-data

Testdata - crossvalidation

Hvordan vurderer vi om vi overfitter, hvis vi ikke har et test-datasæt ?

Crossvalidation: Data deles op i et træningssæt og et testsæt

Træningssættet bruges til at estimere parametre i model M

Testsættet bruges til at beregne prædikterede værdier $\tilde{\xi}_i^P(M)$ baseret på estimerede parametre fra træningssættet og de forklarende variable i testsættet, index i løber over testsæt

giver bidrag til prædiktionsvariansen på formen $(x_i - \hat{\xi}_i^P)^2$

Prædiktionsspredning ved crossvalidation: $s_{\text{cv}} = \sqrt{\frac{1}{m}\sum(x_i - \hat{\xi}_i^P)^2}$

m er antal led i summen

Procedure kan gentages med forskellige valg af træningssæt og testsæt

5-fold crossvalidation:

Betragt tilfælde med 50 observationer

Vi deler dem op i 5 blokke: 1-10, 11-20, 21-30, 31-40 og 41-50

1) estimerer model baseret på 1-40, tester på 41-50

2) estimerer model baseret på 1-30 + 41-50 og tester på 31-40

⋮

5) estimerer model baseret på 11-50 og tester på 1-10

I beregning af prædiktionsspredningen får vi nu 50 led

Eventuelt gentage flere gange hvor vi tilfældigt deler op i 5 blokke

Illustration af 5-fold crossvalidation

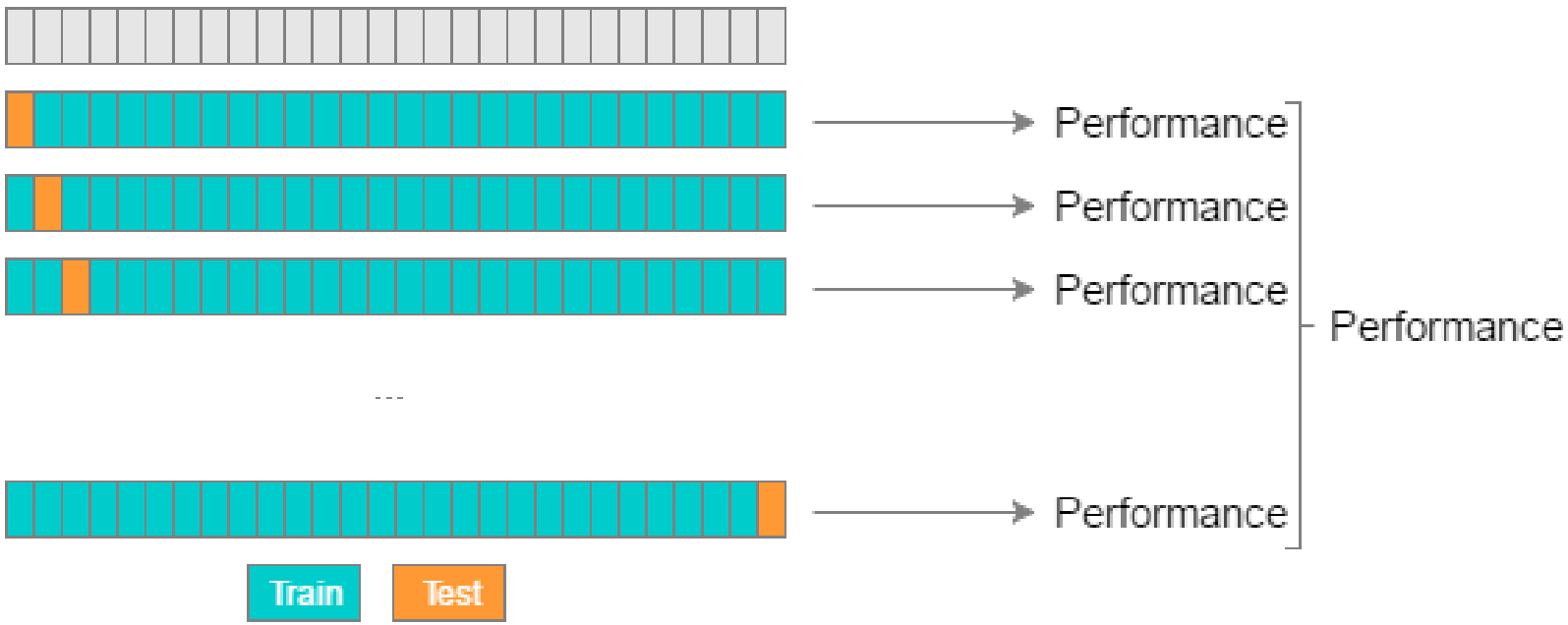
Leave one out crossvalidation (LOOCV):

Specialtilfælde af crossvalidation, hvor testsættet kun består af 1 observation:

- observation i tages ud
- model estimeres fra de resterende $n-1$ observationer
- prædikteret værdi $\hat{\xi}_i^{-i}(M)$ beregnes for den udeladte observation

Prædiktionsspredning ved crossvalidation $s_{\text{cv}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \hat{\xi}_i^{-i}(M)\right)^2}$

Illustration af LOOCV



Overfitting: eksempel

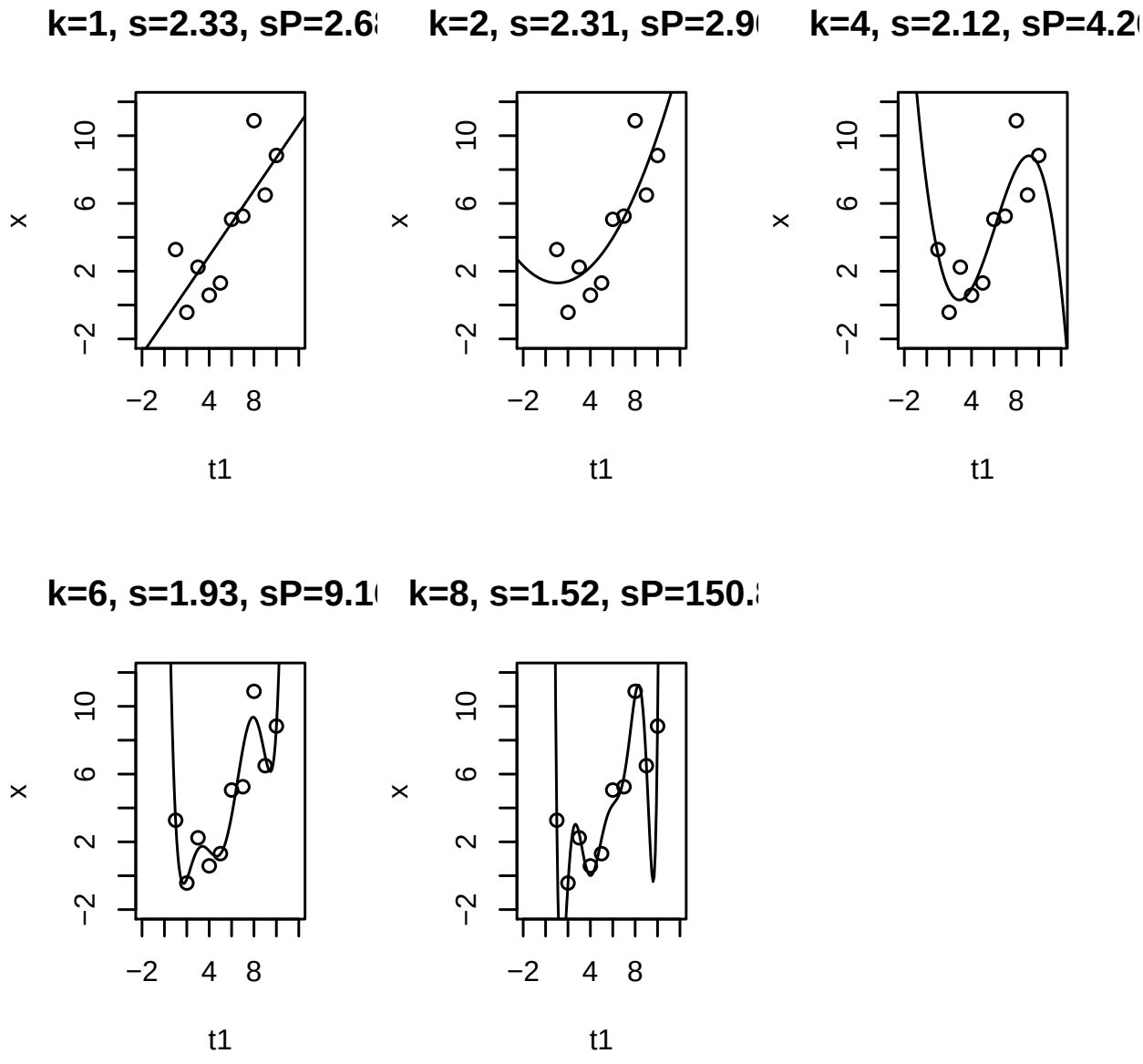
Sande model: $E(X_i) = \alpha + \beta_1 t_i$

Fitter model: $E(X_i) = \alpha + \beta_1 t_i + \beta_2 t_i^2 + \cdots + \beta_k t_i^k$

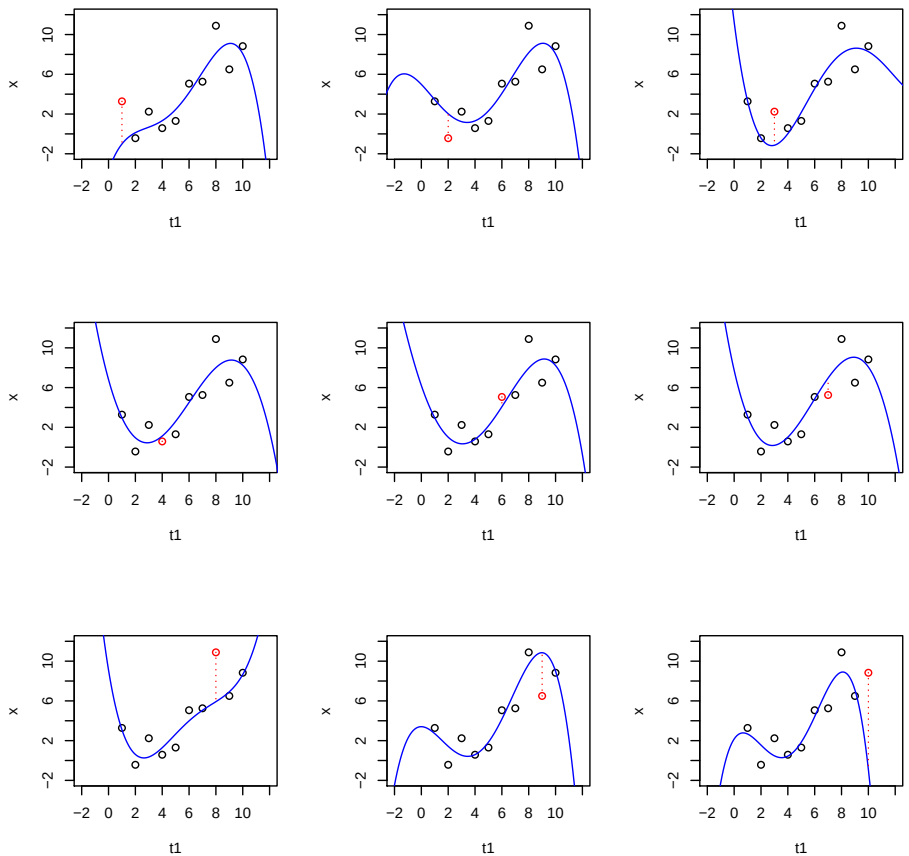
Typisk: jo større k er jo bedre et fit får vi: $s(M_k)$ er lille

hvis $k = n$ er $s(M_k) = 0$!

Fitte polynomium: LOOCV



Fitter 4.gradspolynomium med et punkt udeladt



$$s^2_{\text{cv}} = \frac{1}{10}(4.3^2 + (-2.5)^2 + 3.4^2 + \cdots + (-4.4)^2 + 9.7^2) = 4.26^2$$

Prøv selv i R

```
t=c(1:10)

x=t+rnorm(10)

lmUD=lm(x~poly(t,6))

summary(lmUD)$sigma

sqrt(mean( ( lmUD$residuals/(1-lm.influence(lmUD)$h) )^2 ) )
```

Leave one out crossvalidation er indført

Næste: bruge dette i multipel regression

LOOCV for multipel regressionsmodel

Model: $X_i \sim N(\alpha + \beta_1 t_{i1} + \cdots + \beta_k t_{ik}, \sigma^2)$

Observation i udelades, model estimeres:

estimater: $\hat{\alpha}^{-i}, \hat{\beta}_1^{-i}, \ldots, \hat{\beta}_k^{-i}$

Prædikteret værdi beregnes:

$$\hat{\xi}_i^{-i} = \hat{\alpha}^{-i} + \hat{\beta}_1^{-i} t_{1i} + \cdots + \hat{\beta}_k^{-i} t_{ki}$$

prædiktionsfejl: $x_i - \hat{\xi}_i^{-i}$

Prædiktionsspredning:

$$s_{\text{cv}} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\xi}_i^{-i})^2}$$

Hjemmelavet funktion til LOOCV uden variabelselektion:

```
loocv=function(fit){
  h=lm.influence(fit)$h
  return(sqrt(mean((residuals(fit)/(1-h))^2)))
}

lmUD=lm(IR~Sa+BD+MC+OC)

loocv(lmUD)
```

Vis kørsel i R

LOOCV for nedsivnings-data

Respons: Infiltration Rate (IR)
Regressionsvariable: Sand, Silt, BD (bulk density), PD (particle density), MC (moisture content), OC (organic carbon)

Sa, Si, BD, PD, MC, OC

Backward selection: Sa+BD+MC+OC

$s(M_B) = 2.42, \quad s_{cv}(M_B) = 3.02$

Forward selection: Sa+BD

$s(M_F) = 2.80, \quad s_{cv}(M_F) = 3.12$

s_{cv} afspejler både usikkerhed i skøn $\hat{\xi}_i^{-i}$ og spredning σ i model

Begge modeller er lige gode, ingen tegn på overfitting

LOOCV i en multipel regressionsmodel er vist

I multiple regressionsmodeller med få variable bibringer crossvalidation ofte ikke noget nyt

Næste: Multipel regression med mange variable

Data

Data er analyseret i artiklen "Application of near-infrared reflectance spectroscopy to compositional analysis of biscuits and biscuit dough", Journal of the Science of Food and Agriculture, 1984

Ønsker at kunne prædiktere mængden af vand i dej ud fra gennemlysning med lys

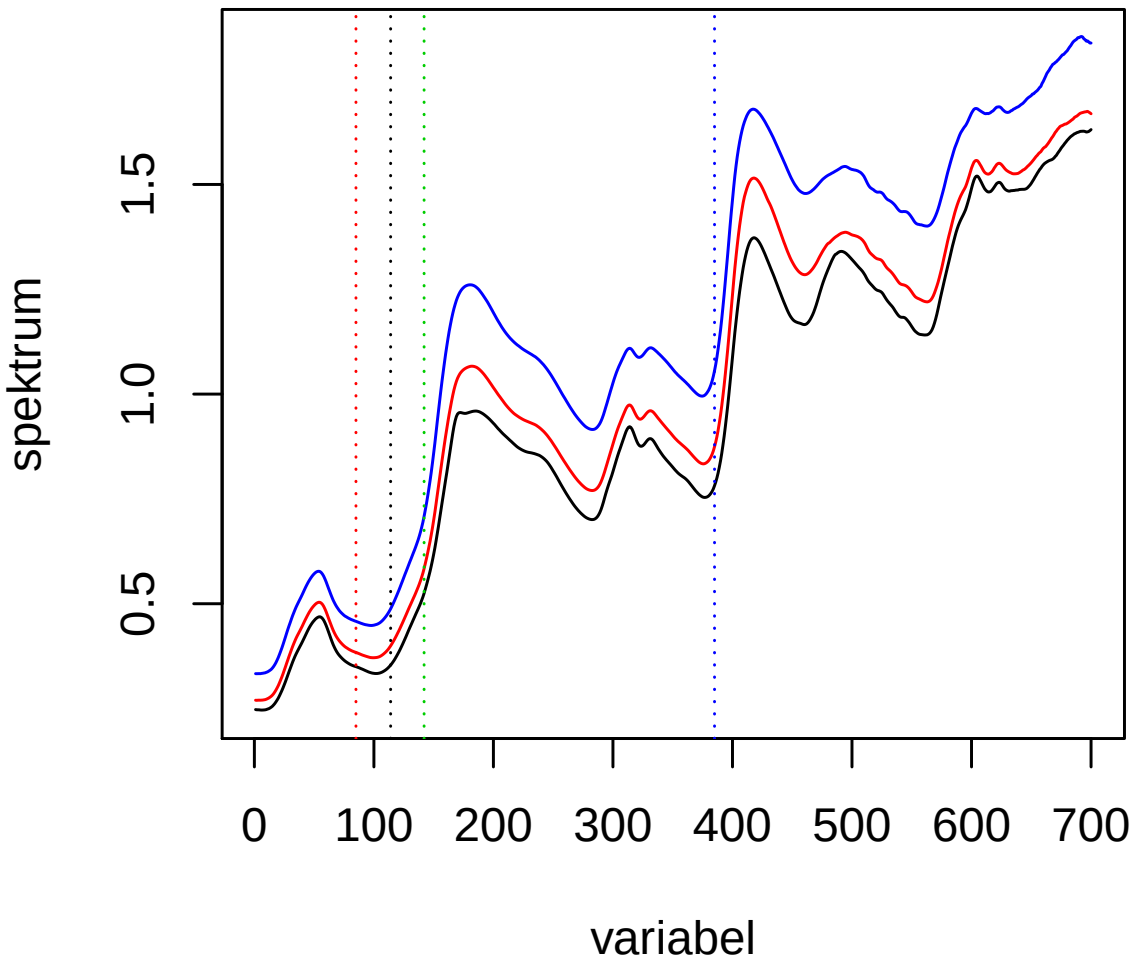
NIR: near infrared reflectance spectroscopy
måler reflektion ved 700 bølgelænger i intervallet 1100-2500 nm

Data: vektor x med vandindhold i $n = 39$ prøver

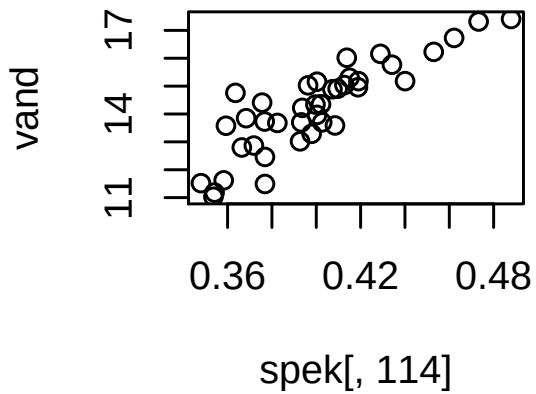
39×700 matrix med reflektioner
hver række giver spektrum for prøve, hver søjle er en forklarende variabel

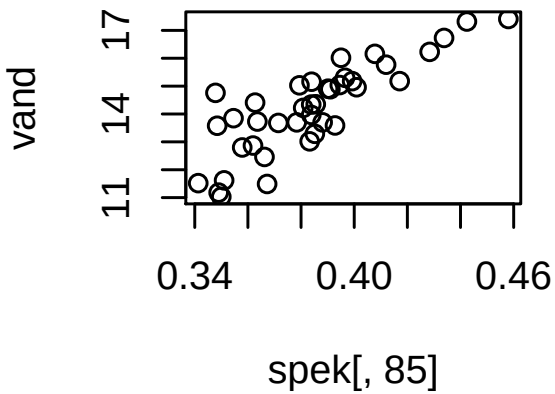
multipel regression med 700 forklarende variable!

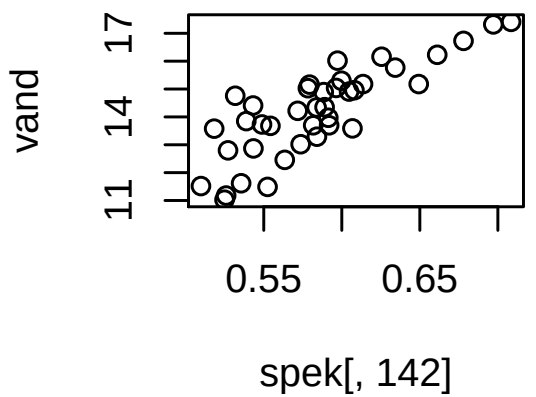
Tre spektre

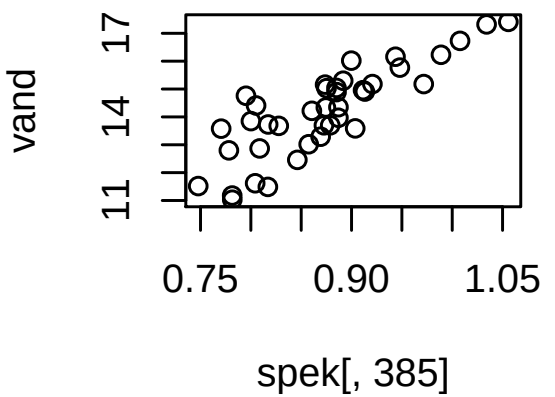


Regression på individuelle spektre









Når antal forklarende variable er større end antal prøver:

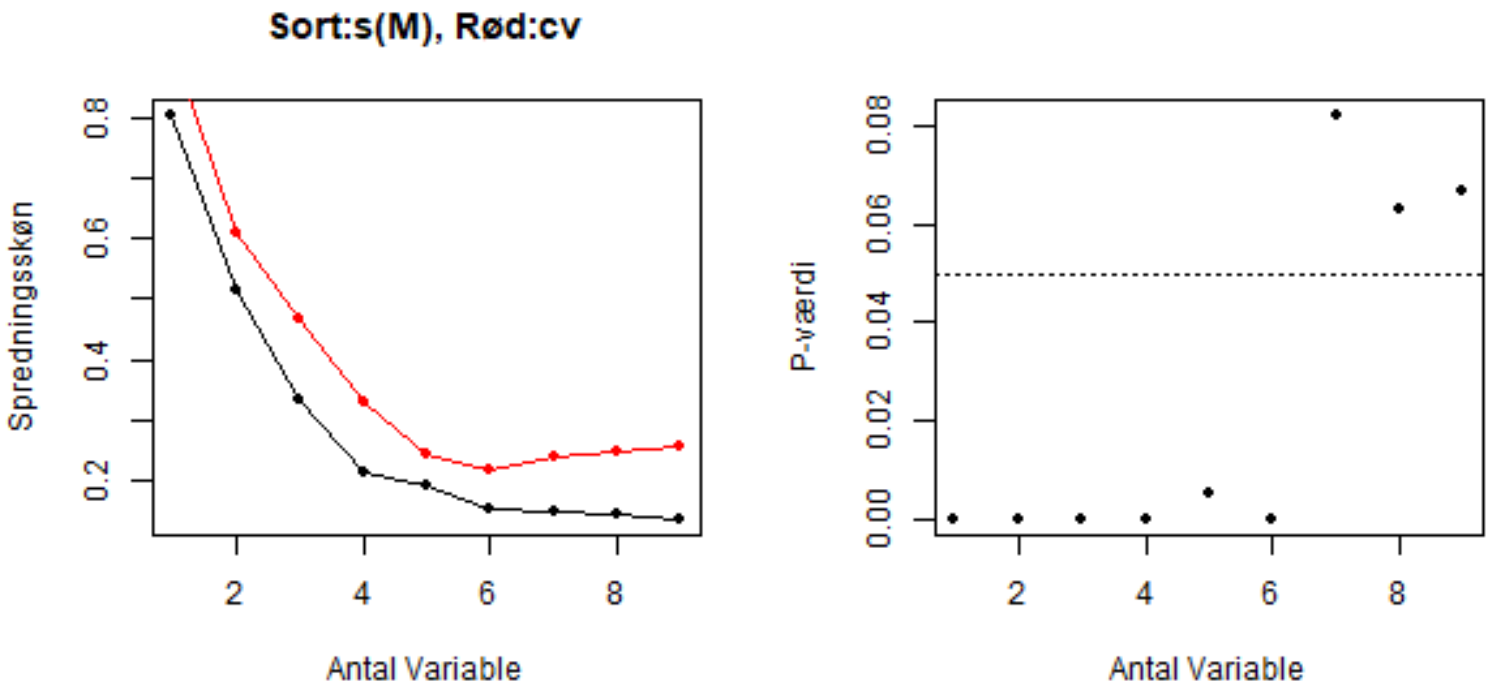
de forventede værdier i multipel regresion = responsværdier

$s^2(M) = 0$

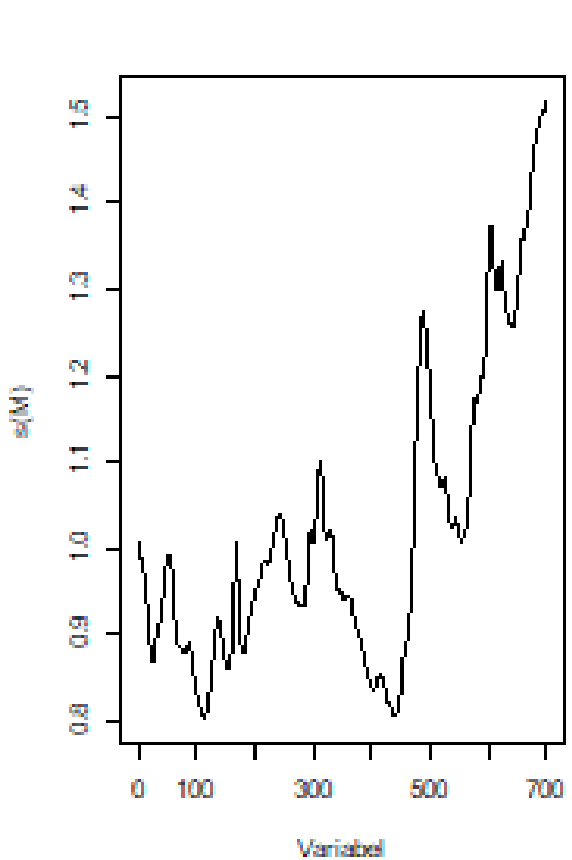
Enten lægge restriktioner på $\hat{\beta}$ (ridge-regression, LASSO-regression)

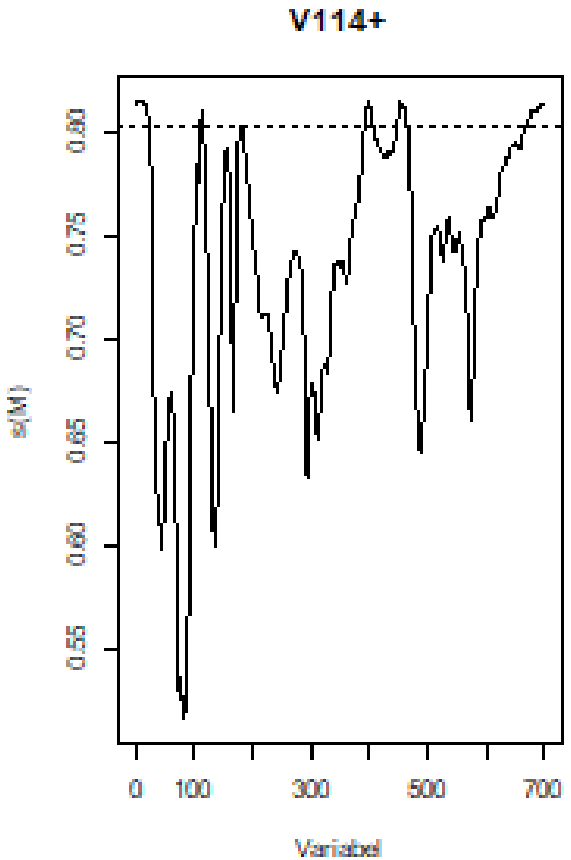
eller: forward selektion

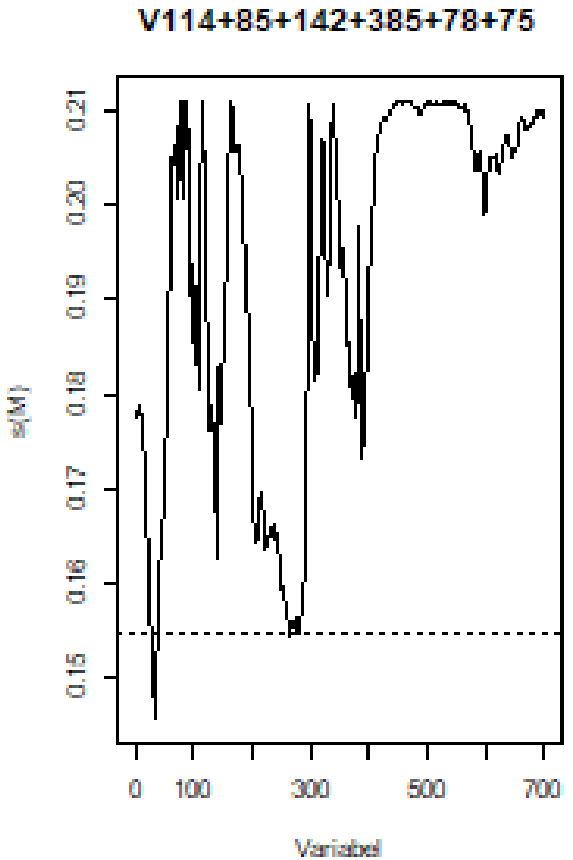
nødvendigt med crossvalidation: figur i webbog afsnit 5.7



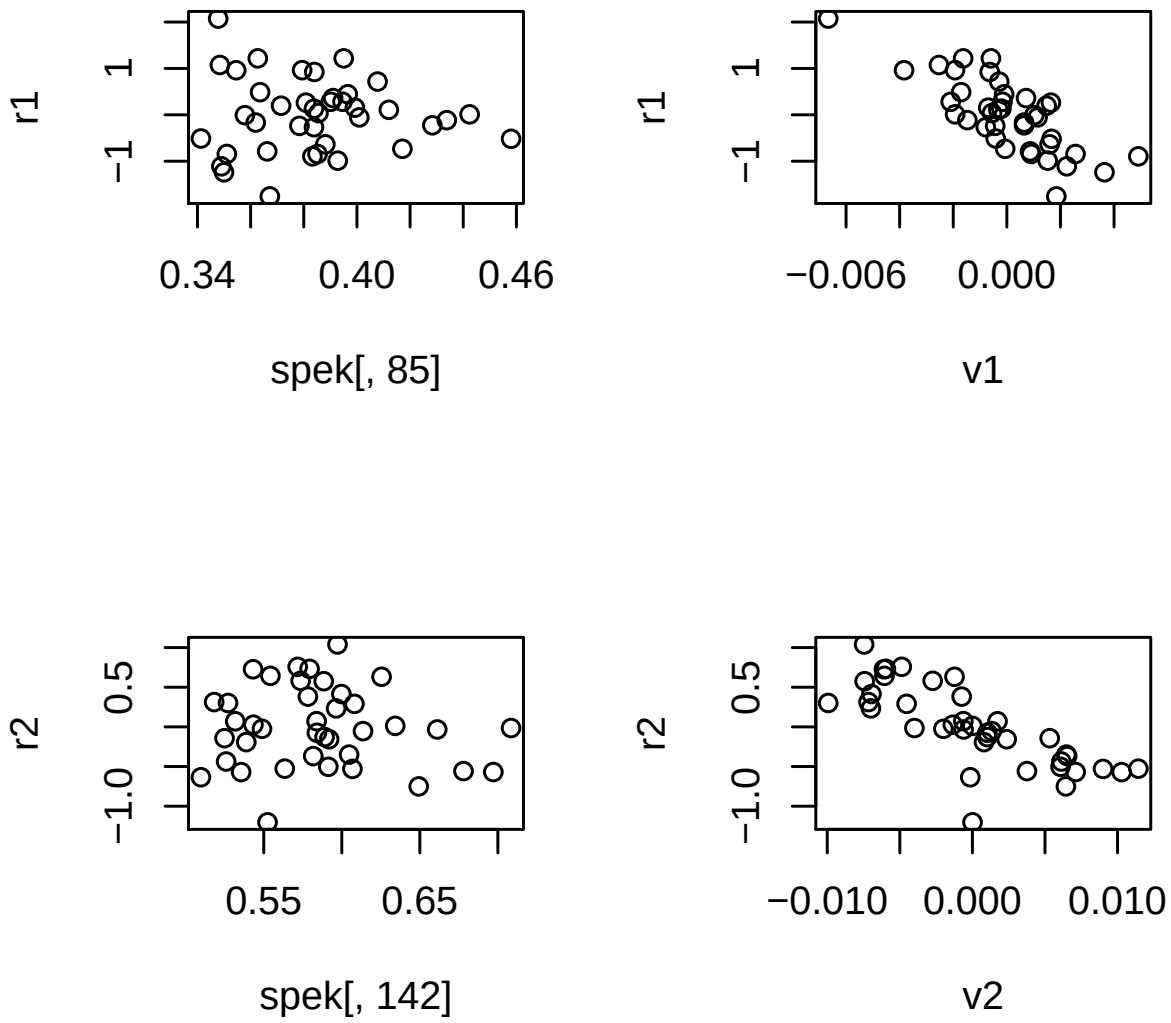
Figur med summary(lm(x∼tj))$sigma, $j = 1, \ldots, 700$
Figur med summary(lm(x∼t114+tj))$sigma, $j = 1, \ldots, 700$







Fra 1 til 2 variable



Finde næste led i forward proceduren:

FWstep(T,x,variable)

T: $n \times k$ matrix med forklarende variable

x: vektor af længde n med responsværdier

variable: vektor med numre på de variable der allerede er inkluderet

Tjek it out: webbog afsnit 5.7: se næste slide

(først vise dej-kørsel i mit eget R, forklar program)

Gå til webbog afsnit 5.7 første kodevindue
Fjern alt fra og med linje 21 til og med linje 29. Indsæt i stedet

```
dat=matrix(nottem,20,12)

spek=dat[,-6]

sukker=dat[,6]
```

Lav forward selektion ved at køre koden flere gange

R-beregning: crossvalidation

OBS: ikke crossvalidation som for en given multipel regressionsmodel

LOOCV med j led i forward algoritme: Fjerner observation i fra datasæt, kører forward selektion på reducerede datasæt indtil j variable er inkluderet

Finde cross validation prædiktionsspredning:

FWcrossval(T,x,m)

m: det maksimale antal led der ønskes undersøgt

Tjek it out: webbog afsnit 5.7 (langsom!)

se figur på tidligere slide, webbog afsnit 5.5 og 5.7

Crossvalidation prædiktionsspredninger fra FWcrossval:

1	2	3	4	5	6	7	8	9	10
0.92	0.61	0.47	0.33	0.24	0.22	0.24	0.25	0.26	0.28

Tyder på at 6 variable er et godt valg

Gå til webbog afsnit 5.7 andet kodevindue

Fjern alt fra og med linje 30 til og med linje 38. Indsæt i stedet

```
dat=matrix(nottem,20,12)

spek=dat[,-6]

sukker=dat[,6]
```

Kør FWcrossval

Prædiktionsspredning fra ægte testsæt

Der er indsamlet 31 nye prøver

Beregner prædiktionsspredning på disse baseret på forward selektion på oprindelige data (træningssæt) med 2 til 10 variable (kode afsnit 5.8)

Antal	2	3	4	5	6	7	8	9	10
Variabelnummer	114+85	+142	+285	+78	+75	+81	+206	+166	+205
Prædiktionsspredning	0.38	0.28	0.31	0.26	0.25	0.26	0.25	0.21	0.21

Tyder på at vi godt kan tage lidt flere end 6 variable

Sceneskift

Slut med nyt materiale i dag

Næste: Regne eksamensopgave