

STK4900/9900 - Lecture 5

Program

1. Checking model assumptions
 - Linearity
 - Equal variances
 - Normality
 - Influential observations
 - Importance of model assumptions
2. Selection of predictors
 - Forward and backward selection
 - Criteria for selecting predictors
3. High dimensional regression

Section 4.7

Chapter 5: only some main points

Assumptions for linear regression

Model: $y_i = \eta_i + \varepsilon_i$



The diagram shows the equation $y_i = \eta_i + \varepsilon_i$ with two green curly braces under the terms η_i and ε_i . The brace under η_i is labeled "systematic part" in green, and the brace under ε_i is labeled "random part (error)" in red.

(1) Linearity: $\eta_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_p x_{pi}$

(2) Constant variance (homoscedasticity): $\text{Var}(\varepsilon_i) = \sigma_\varepsilon^2$ for all i

(3) Normally distributed errors: $\varepsilon_i \sim N(0, \sigma_\varepsilon^2)$

(4) Uncorrelated errors: $\text{Cov}(\varepsilon_i, \varepsilon_j) = 0$ for all $i \neq j$

We will here focus on the three first assumptions and return to the 4th in the second part of the course

Fitted values and residuals

Fitted values: $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i} + \dots + \hat{\beta}_p x_{pi}$

Residuals: $r_i = y_i - \hat{y}_i$

Diagnostic – plots of the residuals

Plots of the residuals may be used to check:

- (1) Linearity
- (2) Constant variance
- (3) Normal errors (including outliers)

(1) Check of linearity

Assume that the correct form of the systematic part of the model is

$$\eta_i = \beta_0 + \beta_1 x_{1i} + \dots + \beta_{j-1} x_{j-1,i} + f_j(x_{ji}) + \beta_{j+1} x_{j+1,i} + \dots + \beta_p x_{pi}$$

i.e. the model is linear in all predictors, except possibly for the j -th

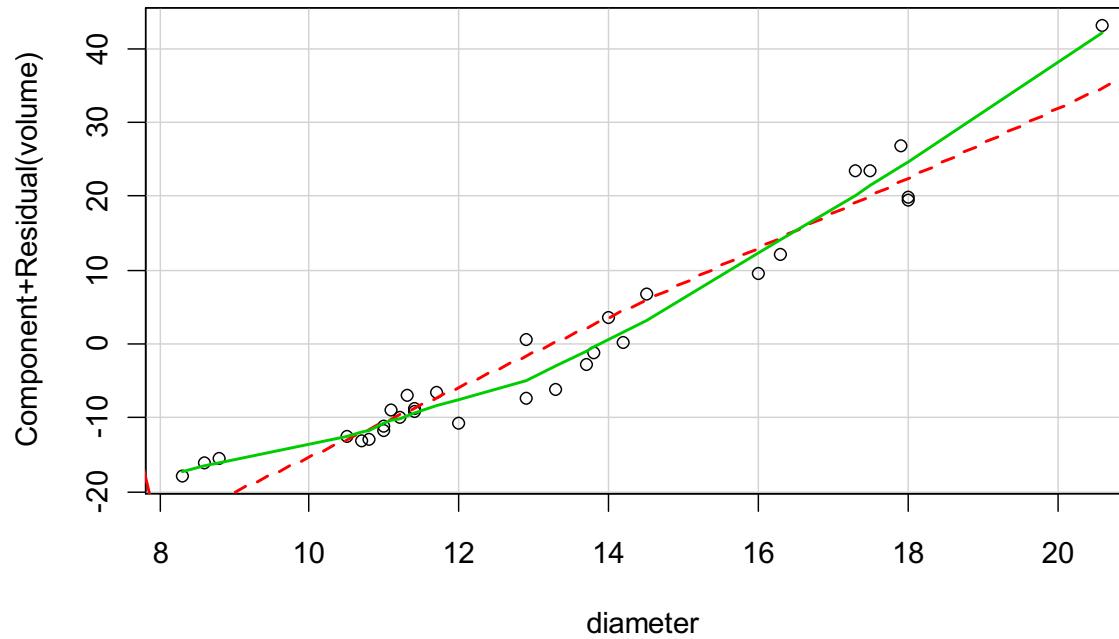
We may estimate the function $f_j(x)$ based on a plot of the *partial residuals* $\hat{\beta}_j x_{ji} + r_i$ versus the values of the predictor (x_{ji})

In the text book the plot is denoted a component-plus-residual plot (CPR plot)

To obtain a CPR plot in R, we have to use the "car" library

Example: tree data

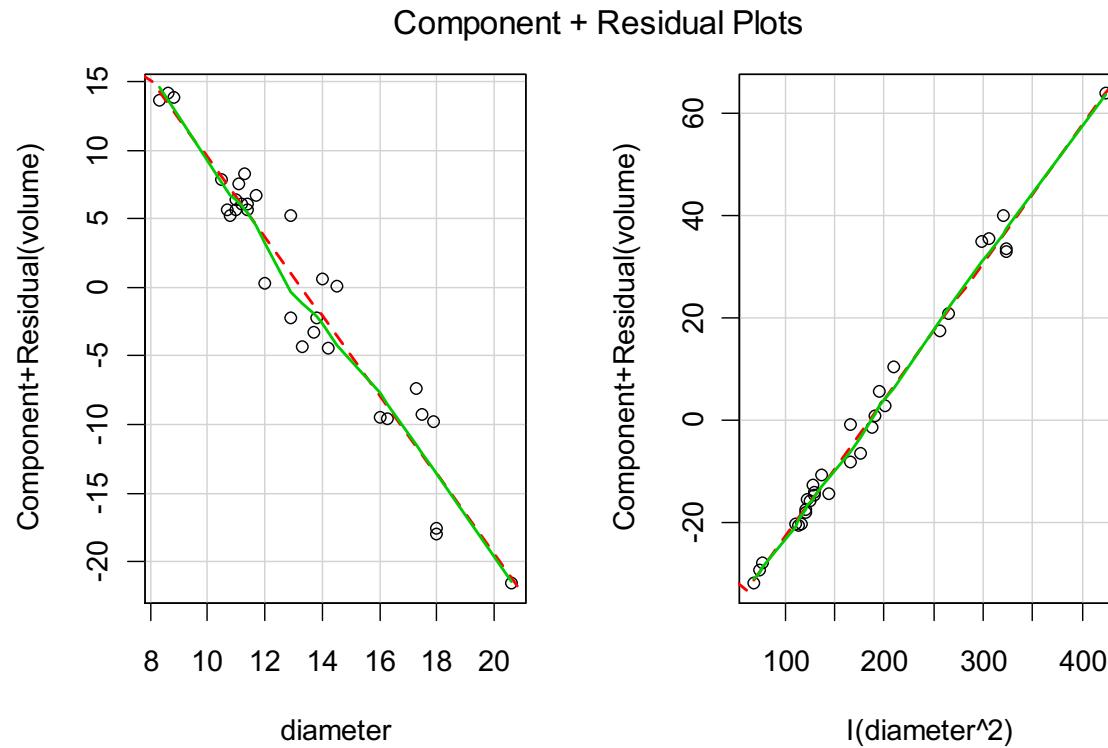
We fit a model with volume as outcome and diameter and height as predictors, and make a CPR plot for diameter:



The plot indicates that a second degree polynomial may be more appropriate

```
trees=read.table("http://www.uio.no/studier/emner/matnat/math/STK4900/data/trees.txt",header=T)
fit.both=lm(volume~diameter+height, data=trees)
library(car)
crPlots(fit.both, terms=~diameter)
```

We fit a model that also has a second degree term for diameter, and make a CPR plots for diameter and diameter²



The plots indicate that the linearity assumption is reasonable both for diameter and diameter² (i.e. linearity in the parameters)

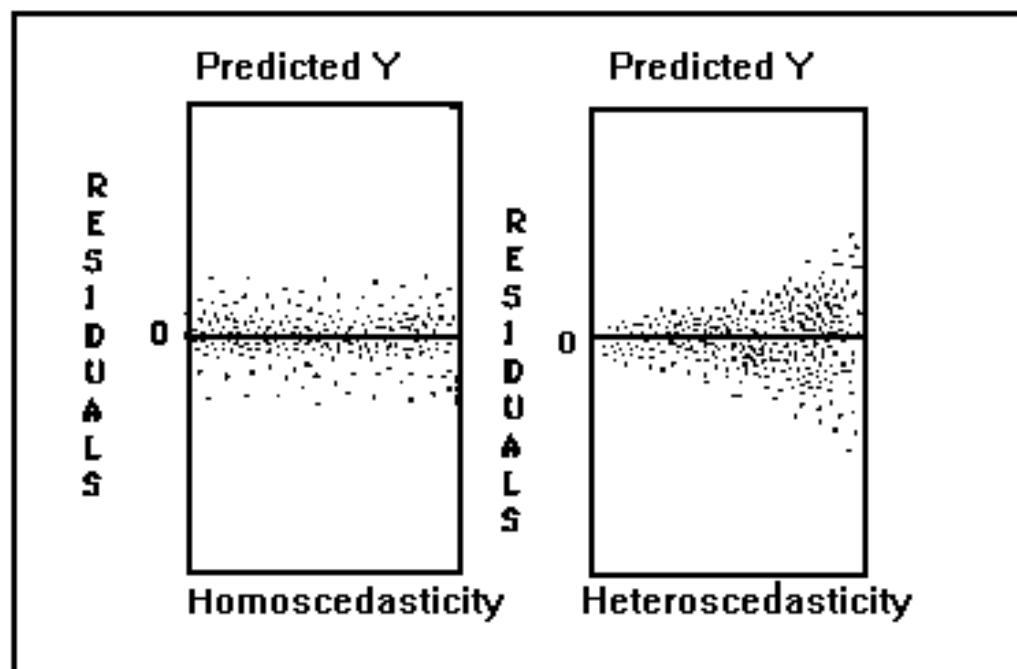
```
fit.sq=lm(volume~diameter+I(diameter^2)+height, data=trees)  
crPlots(fit.sq, terms=~diameter+I(diameter^2))
```

(2) Check of constant variance (homoscedasticity)

If the model is correctly specified, there should be no systematic patterns in the residuals

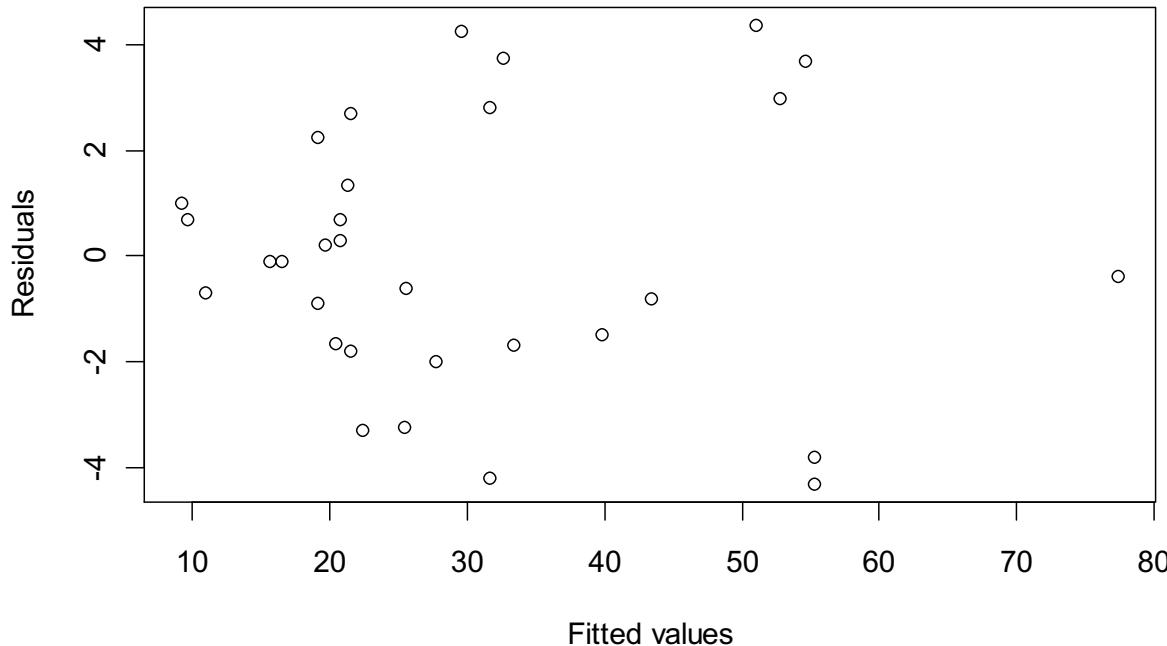
A plot of the residuals versus the fitted (or predicted) values may be used to check the assumption of equal variances

If the variances increase with the expected outcome, the plot will have a fan like shape (like the right hand plot below)



Example: tree data

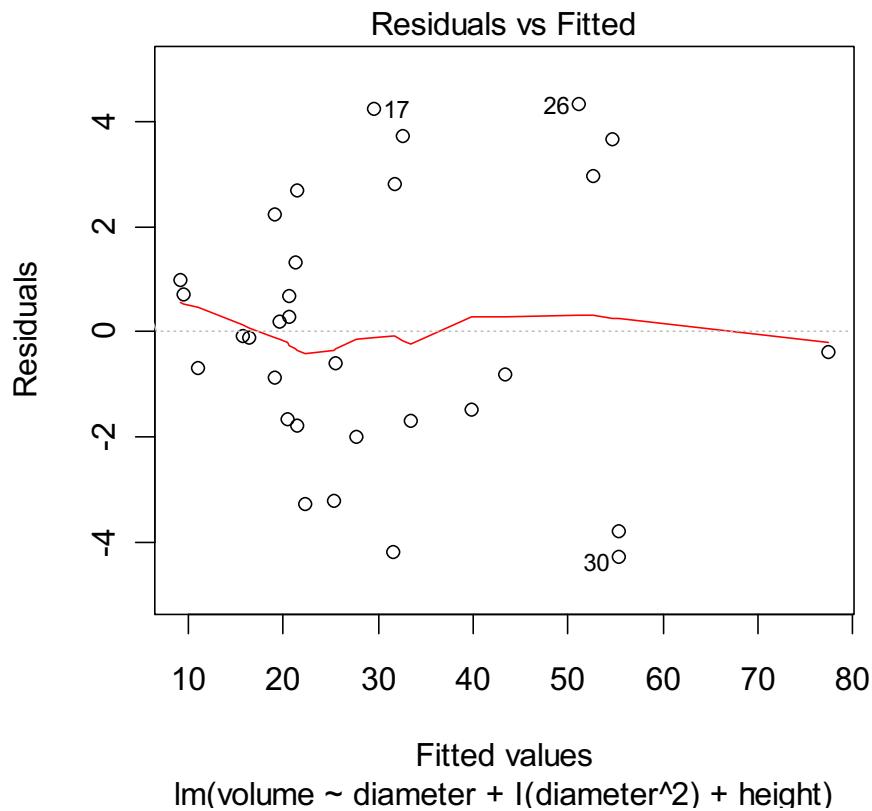
We fit a model with volume as outcome and diameter, diameter², and height as predictors, and plot the residuals versus the fitted values



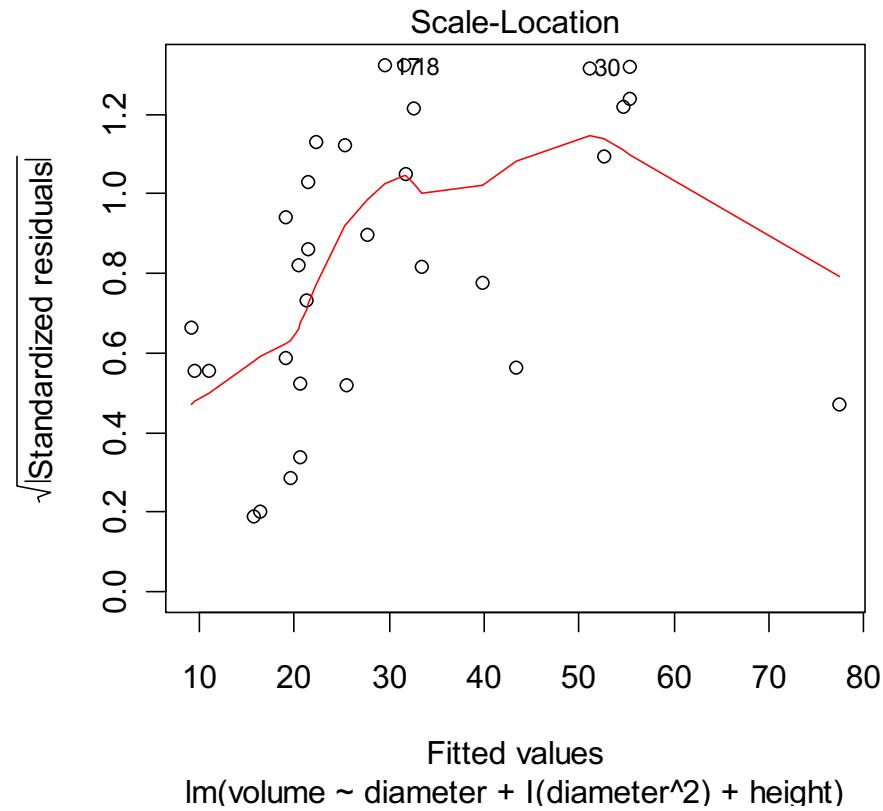
The plot is quite reasonable, but there may be some indication of increasing variances

```
fit.sq=lm(volume~diameter+I(diameter^2)+height, data=trees)
plot(fit.sq$fit, fit.sq$res, xlab="Fitted values", ylab="Residuals")
```

[plot\(fit.sq,1\)](#)



[plot\(fit.sq,3\)](#)



The added line helps to see if there is a pattern in the residuals (which may be due to non-linearities)

The fitted lines may not be trusted where there is little data (i.e. in the right-hand part of the plots above)

The added line helps to see if the variance (or standard deviation) is increasing (heteroscedasticity)

(3) Check of normality

If the model is correctly specified, the residuals should behave as a sample from a normal distribution with mean zero

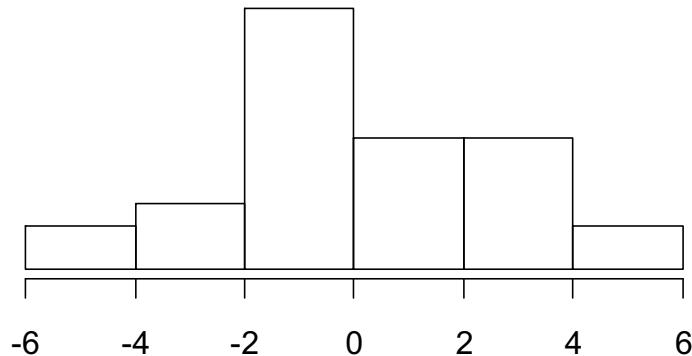
Various plots may be made to check this:

- Histogram of residuals
- Boxplot of residuals
- Normal Q-Q plot of residuals

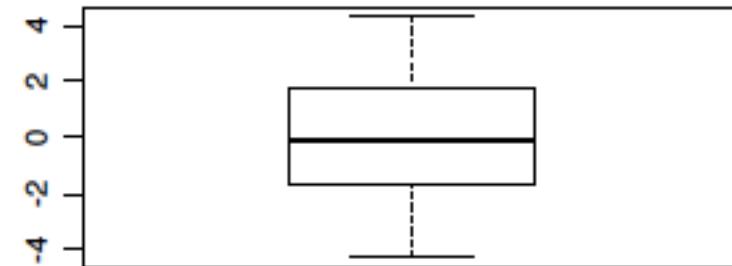
Example: tree data

We fit a model with volume as outcome and diameter, diameter², and height as predictors, and make different plots of the residuals

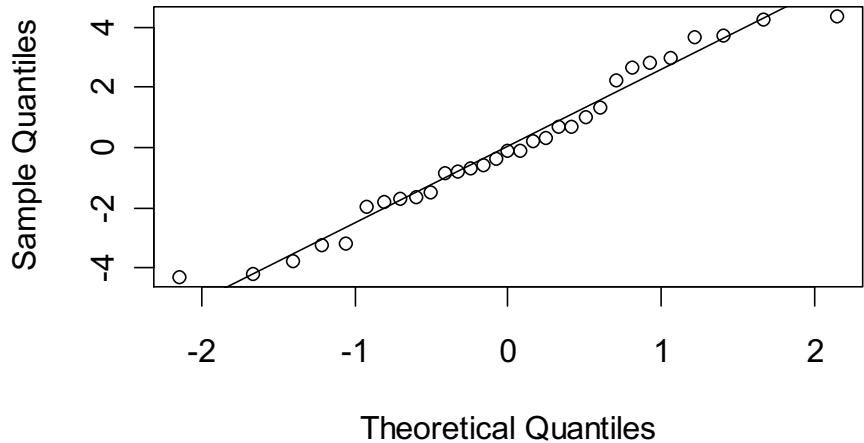
Histogram



Boxplot



Q-Q plot



The Q-Q plot should be close to a straight line if the residuals are normally distributed

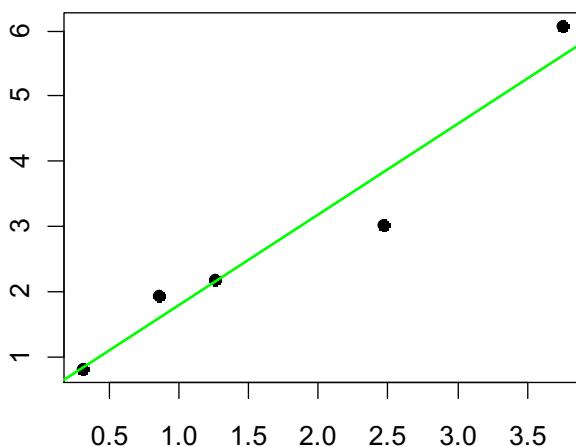
```
hist(fit.sq$res)
boxplot(fit.sq$res)
qqnorm(fit.sq$res); qqline(fit.sq$res)
Alternative: plot(fit.sq,2)
```

Influential observations

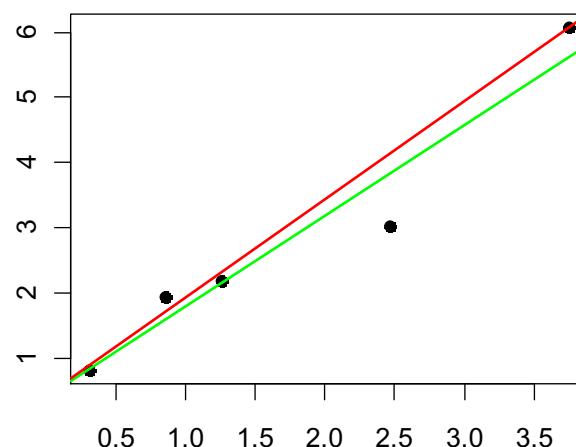
Consider the erosion example:

Amount of water (l/s)	0.31	0.85	1.26	2.47	3.75
Erosion (kg)	0.82	1.95	2.18	3.02	6.07

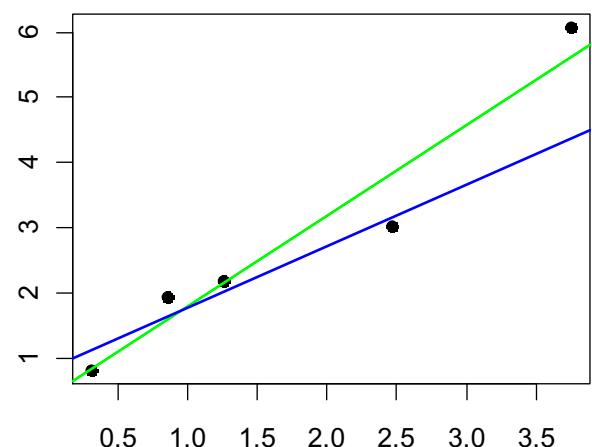
Least squares fit:



Least squares fit without
the 4th observation:



Least squares fit without
the last observation:



The last observation has a larger influence on the slope estimate than the 4th observation

A measure for the influence of an observation is the change in the estimate(s) when the model is fitted leaving out the observation

These "dfbetas" (delete-and-fit-betas) are easily computed in R:

R-commands:

```
fit=lm(erosion~water)  
summary(fit)  
dfbeta(fit) # dfbetas
```

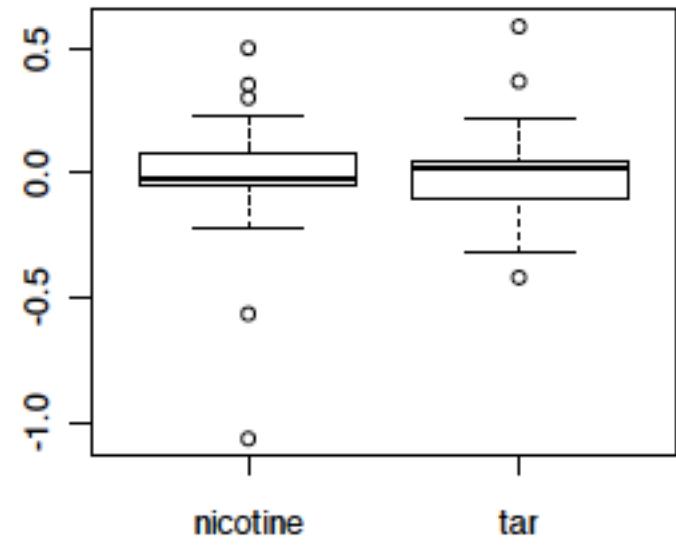
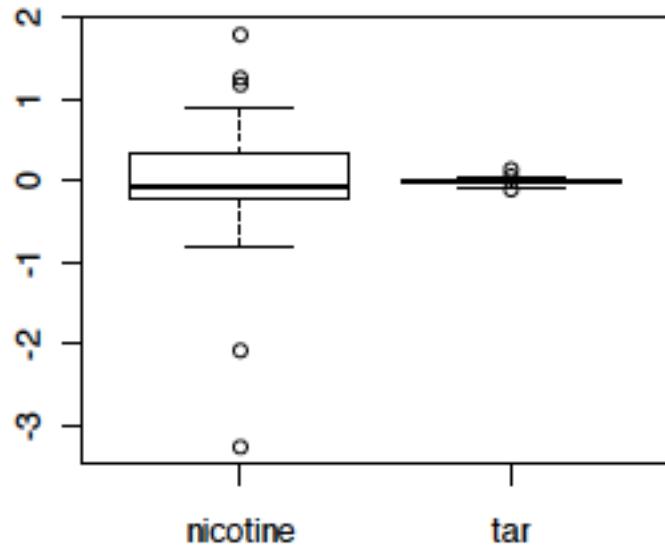
R-output (edited):

	Estimate	Std. Error
(Intercept)	0.4061	0.4454
water	1.3900	0.2096

The command
"dfbetas(fit)"
gives standardized
dfbetas that may be
more appropriate for
multiple linear
regression when the
predictors are on
different scales

(Intercept)	water
1 -0.0164	0.0059
2 0.2066	-0.0596
3 0.0089	-0.0018
4 -0.0362	-0.1093
5 -0.4386	0.4511

Boxplots of dfbetas (left) and standardized dfbetas (right) for the cigarette data (omitting the intercept, which usually is of less interest)



It may be useful to inspect observations that have a large influence on the estimates

```
cigarettes=read.table("http://www.uio.no/studier/emner/matnat/math/STK4900/data/cigarettes.txt",
                      header=T)
fit.cig=lm(CO~nicotine+tar,data=cigarettes)
boxplot(dfbeta(fit.cig))
boxplot(dfbetas(fit.cig))
```

The importance of model assumptions

Without linearity of the predictors we have a wrong specification of the systematic part of the model:

- The effect of a predictor may be wrongly estimated
- A predictor may be important, but we do not know
- Serious nonlinearity jeopardizes the analysis

If the variances are not equal (and/or the errors are correlated):

- The estimates of the β_j 's will be unbiased
- The standard errors can be wrongly estimated
- Confidence intervals and P-values can be flawed
- Prediction intervals are flawed

If the errors are not normal – but the other model assumptions are true:

- Estimates of standard errors are valid
- Test statistics are not exactly t- and F-distributed, but for large n they are approximately so
- The distributional assumptions are not critical

A few influential observations may, however, have large effects on the estimates. How these are treated may be critical for the conclusions on the relations between covariates and response

Model misfit and possible improvements

Non-linearity:

- Transform x_{ji} , e.g. $\log(x_{ji})$
- Transform y_i , e.g. $\log(y_i)$
- Include second order term(s) and/or interaction(s)
- GAM (Generalized additive models, 4.10.1, more on slide 20-23)

Heteroscedasticity:

- Transform y_i , typically log-transform or root-transform
- (More advanced: use weighted least-squares or a generalized linear model)

Non-normality:

- Transform y_i , e.g. $\log(y_i)$
- Bootstrap
- For large n the problem can be ignored

Influential observations:

- Check the coding of the observations
- Run the regression without the influential observations
How different are the estimates?

Generalized additive models (GAM)

Similarly to CPR-plots we can extend the linear model

$$\eta_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_p x_{pi}$$

by assuming a general functional dependency on the covariates

$$\eta_i = \beta_0 + f_1(x_{1i}) + f_2(x_{2i}) + \dots + f_p(x_{pi})$$

Thus the terms $\beta_j x_{ji}$ are replaced by functions $f_j(x_{ji})$.

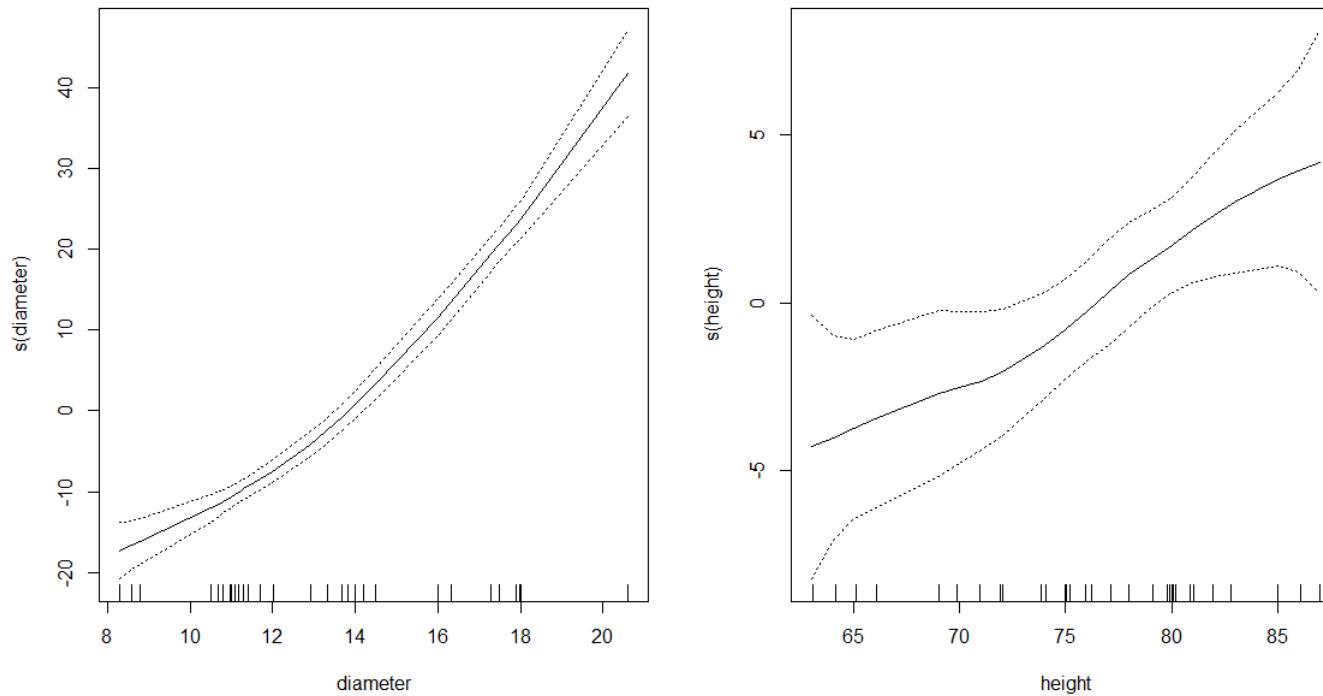
These functions are assumed to be smooth (continuous and having derivatives)

Loading the library gam in R allows for actually estimating and plotting these functions (based on regression splines)

It is also possible to estimate confidence intervals for the estimated curves and testing whether there is a significant non-linearity in the models.

Example GAM: tree data

We fit a model with volume as outcome depending on smooth functions of diameter and height:



```
library(gam)
fit.gam.both=gam(volume~s(diameter)+s(height), data=trees)
par(mfrow=c(1,2))
plot(fit.gam.both,se=T)
```

Example GAM: tree data (contd)

The functional dependency is specified by writing `s(diameter)` and `s(height)` in the model fitting statement. We could (for instance) force the dependency on height to be linear by instead writing

```
fit.gam.dia=gam(volume~s(diameter)+height, data=trees)
```

We obtain confidence interval by specifying `se=T` in the plot command

Here we are not able to force a straight line within the confidence limits for the diameter-function. This indicates that there is a significant non-linearity for this variable.

It is, however, possible to let a straight line go through the intervals for height. This indicates that there is no important non-linearity for this variable.

We can test this more carefully, next slide.

Example GAM: tree data (contd)

The functional dependency is specified by writing `s(diameter)` and `s(height)` in the model fitting statement. We could (for instance) force the dependency on height to be linear by writing `s(diameter)+height`

Then non-linearities can be tested with standard F-tests:

```
fit.gam.dia=gam(volume~s(diameter)+height, data=trees)
anova(fit.both,fit.gam.dia,fit.gam.both)
Analysis of Variance Table
```

Model 1: $\text{volume} \sim \text{diameter} + \text{height}$

Model 2: $\text{volume} \sim s(\text{diameter}) + \text{height}$

Model 3: $\text{volume} \sim s(\text{diameter}) + s(\text{height})$

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	28	421.92			
2	25	180.56	3.0000	241.36	10.13
3	22	174.73	3.0002	5.84	0.24

The non-linearity for diameter is clearly significant,
there is no reason to include a non-linear term for height.

Selection of predictors

When there are a number of predictors, a choice has to be made on which ones to include in a regression model

In general we would like to have

- a simple model
- with good empirical fit

These two aims may be conflicting and the trade-off between them may depend on the objectives of the study

Possible objectives:

- Study the effect of one predictor while adjusting for the effects of the other predictors (the predictor of main interest should always be included in the model)
- Identify important predictors for an outcome
- Predict the outcome for a new unit where only the values of the predictors are available

Sub-models

Consider a model with p possible predictors:

$$E(y_i | \mathbf{x}_i) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_p x_{pi}$$

There are 2^p possible ways to make a sub-model
(i.e. a model with some of the predictors)

- For $p = 10$ there are $2^p = 1024$ different sub-models
- For $p = 20$ there are $2^p \approx 10^6$ different sub-models

For each numeric covariate one may also include e.g. a quadratic term

Further one may take interactions into account

Except for small values of p it is not feasible to investigate all possible sub-models

We need strategies for deciding which sub-models to consider

Forward selection:

1. Fit all p models with only one predictor
2. Choose the predictor that "contributes most"
3. Run $p - 1$ regressions with this predictor and another one
4. Choose the model that "fits" best
5. Continue adding predictors until "no improvement"

Since predictors that have been included on an earlier stage need not continue to be important later on, step 4 can be supplemented with deletion of predictors that no longer contribute (stepwise regression)

Backward selection:

1. Fit the model with all p predictors
2. Compare the model with all predictors with the p different models where one predictor has been left out
3. Leave out the "least important" predictor
4. Compare the model now obtained with the $p - 1$ different models where one more predictor has been left out
5. Leave out the "least important" predictor
6. Continue in this way until a model is obtained that only contains "important" predictors

Criteria for inclusion / exclusion

When using forward or backward selection, one needs a criterion for when to include/exclude a predictor

Different criteria may be used, and the choice between them may depend on the objectives of the study

Some possibilities:

- P-values
- adjusted R^2
- cross-validated R^2

P-values

Forward selection:

- include at each step the most significant predictor (lowest P-value)

Backward selection:

- exclude at each step the least significant predictor (largest P-value)

P-values are mainly used when the objective is either

- to study the effect of one predictor while adjusting for the effects of the other predictors
- to identify important predictors for an outcome

Need to decide a cut-off for when to include/exclude a predictor

Often 5% is used, but the text book recommends a more liberal cut-off (combined with backward selection) when the aim is to correct for possible confounders

Ordinary R²

The coefficient of determination

$$R^2 = \frac{MSS}{TSS} = 1 - \frac{RSS}{TSS}$$

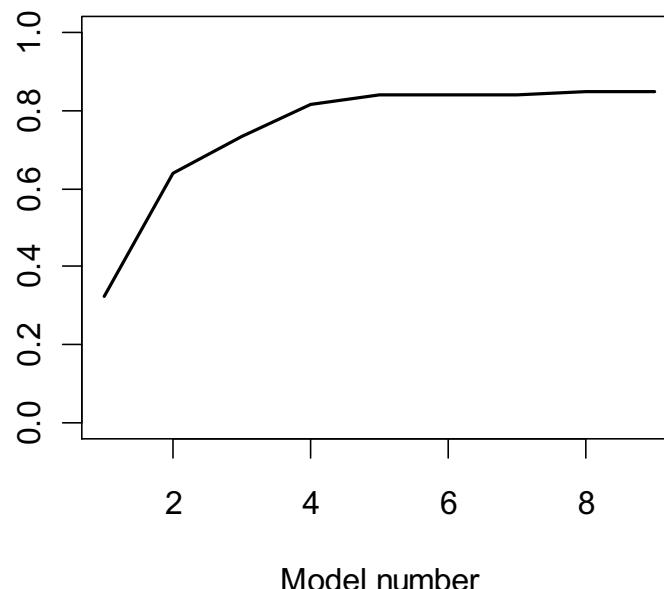
measures the proportion of the total variability in the outcomes that is accounted for by the predictors in the model

It could be tempting to choose the model with the largest R²

But then we would end up with a model including all predictors

Example of R² from practical exercise 14.e

Maximum for the largest model



The adjusted R²

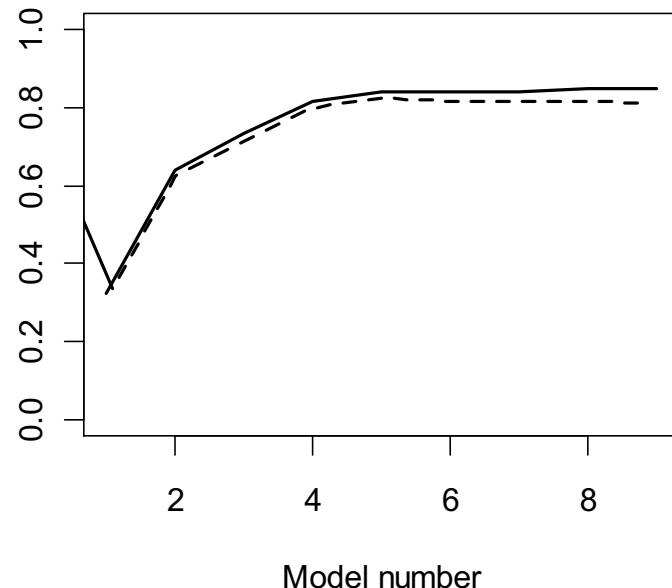
$$R_{\text{adj}}^2 = 1 - \frac{RSS/(n-p-1)}{TSS/(n-1)}$$

penalizes including more predictors

The adjusted R² will have a maximum over the different models considered, and it may therefore be used to select predictors

Example of adjusted R²
from practical exercise 14.e
(dashed line)

Maximum for model 5



Cross validation

A drawback with R^2 and adjusted R^2 is that the observations are used twice:

- estimate the β_j 's
- evaluate the predictions of the y_i 's :

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i} + \dots + \hat{\beta}_p x_{pi}$$

Idea:

- Estimate the regression model without using the observation y_i
- Predict y_i using the obtained estimates

Denote this prediction $\hat{y}_i^{(-i)}$

Cross validated R²

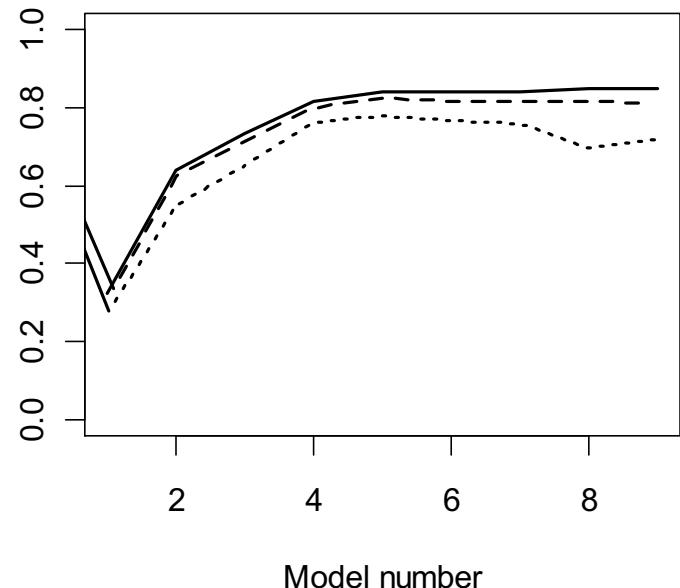
$$R_{\text{cv}}^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i^{(-i)})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

The cross-validated R² will have a maximum over the different models considered, and it may therefore be used to select predictors

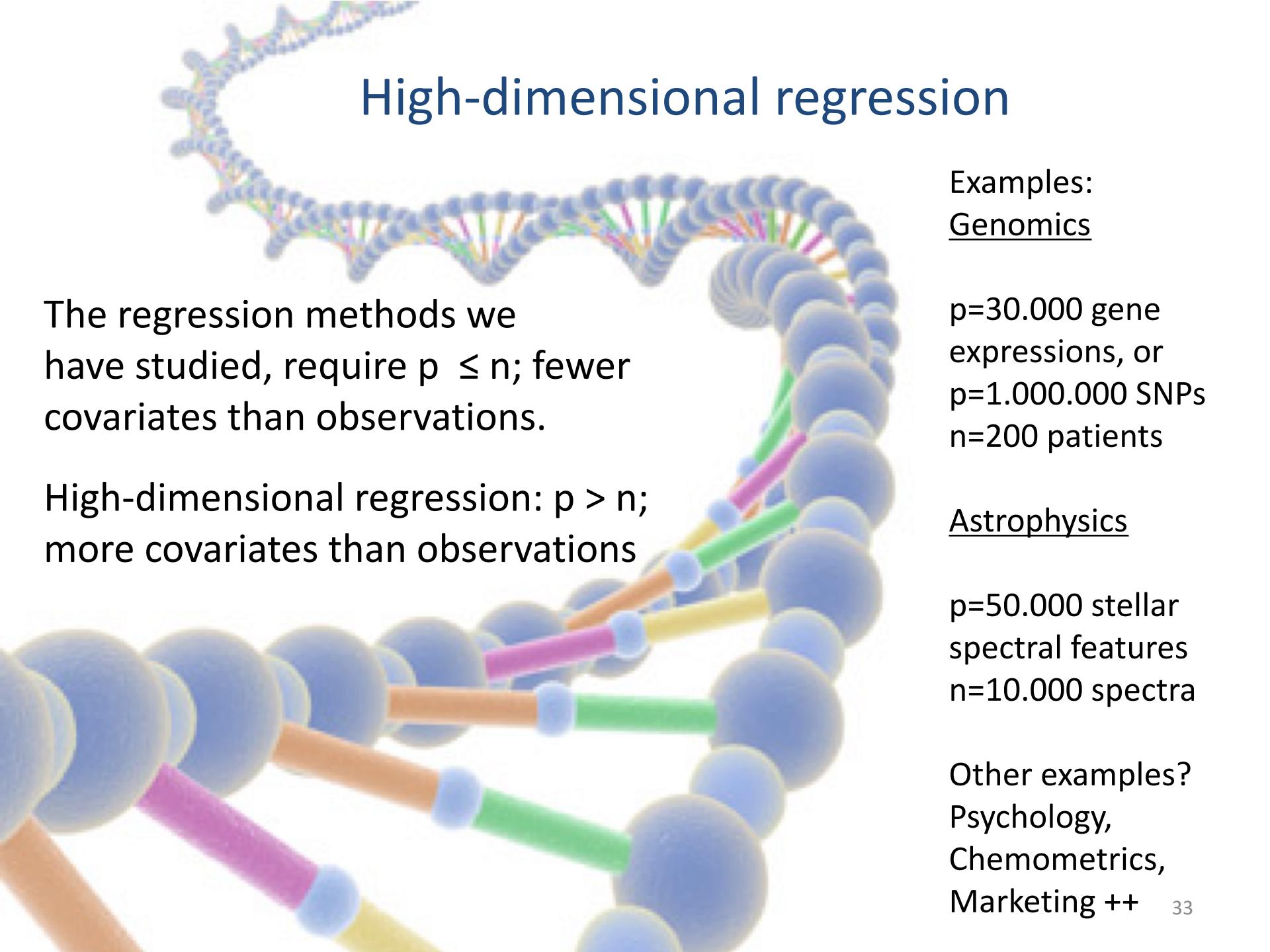
Example of crossvalidated R²
from practical exercise 14.e
(dotted line)

Maximum for model 5, which is the same as for the adjusted R²

But often the cross-validated R² will give smaller models than the adjusted R²



We have described "leave-one-out" cross validation.
Alternative versions of cross-validation exist, e.g. 10-fold cross validation



High-dimensional regression

The regression methods we have studied, require $p \leq n$; fewer covariates than observations.

High-dimensional regression: $p > n$; more covariates than observations

Examples:
Genomics

$p=30.000$ gene expressions, or
 $p=1.000.000$ SNPs
 $n=200$ patients

Astrophysics

$p=50.000$ stellar spectral features
 $n=10.000$ spectra

Other examples?
Psychology,
Chemometrics,
Marketing ++

Remember the multiple linear regression model:

Data: $(y_i, x_{1i}, x_{2i}, \dots, x_{pi}) \quad i = 1, \dots, n$

y_i = outcome for unit no. i

x_{ji} = predictor (covariate) no. j for unit no. i

Model:

$$\begin{aligned} y_i &= E(y_i | \mathbf{x}_i) + \varepsilon_i \\ &= \underbrace{\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_p x_{pi}}_{\text{systematic part (linear predictor)}} + \underbrace{\varepsilon_i}_{\text{random part (noise) } N(0, \sigma_\varepsilon^2)\text{-distributed}} \end{aligned}$$

The least squares estimators from Lecture 3 are not unique when $p > n$. We need some regularization to find estimates for the $p+1$ coefficients!

For $p < n$, in Ordinary Least Squares (OLS) regression, we are minimizing the MSE

$$\text{MSE} = (y - X_0\beta_0 - X_1\beta_1 - \dots - X_p\beta_p)^2 = (y - X\beta)^2$$

resulting in

$$\hat{\beta} = \operatorname{argmin}_{\beta} (y - X\beta)^2$$

We can constrain the regression coefficients in order to stabilize estimation and shrink or even eliminate the coefficients of unimportant predictors. Works also when $p > n$!

L₂ penalty $\hat{\beta} = \operatorname{argmin}_{\beta} (y - X\beta)^2 + \lambda \|\beta\|_2^2$ Ridge

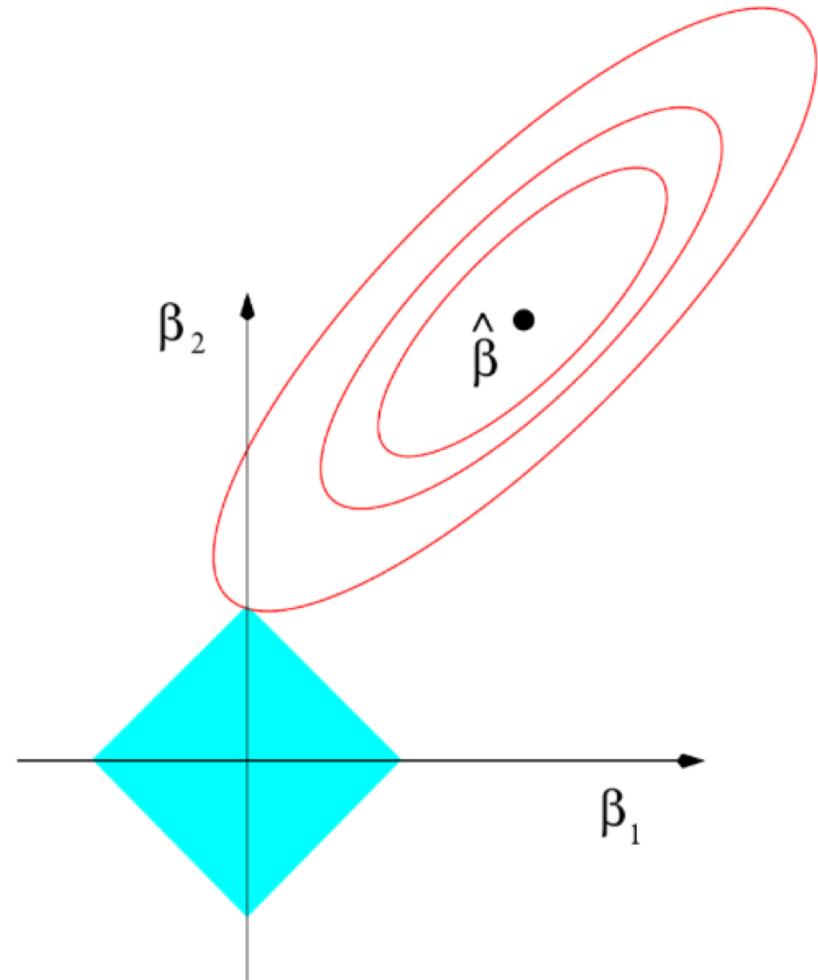
Penalty parameter

L₁ penalty $\hat{\beta} = \operatorname{argmin}_{\beta} (y - X\beta)^2 + \lambda \|\beta\|_1$ Lasso

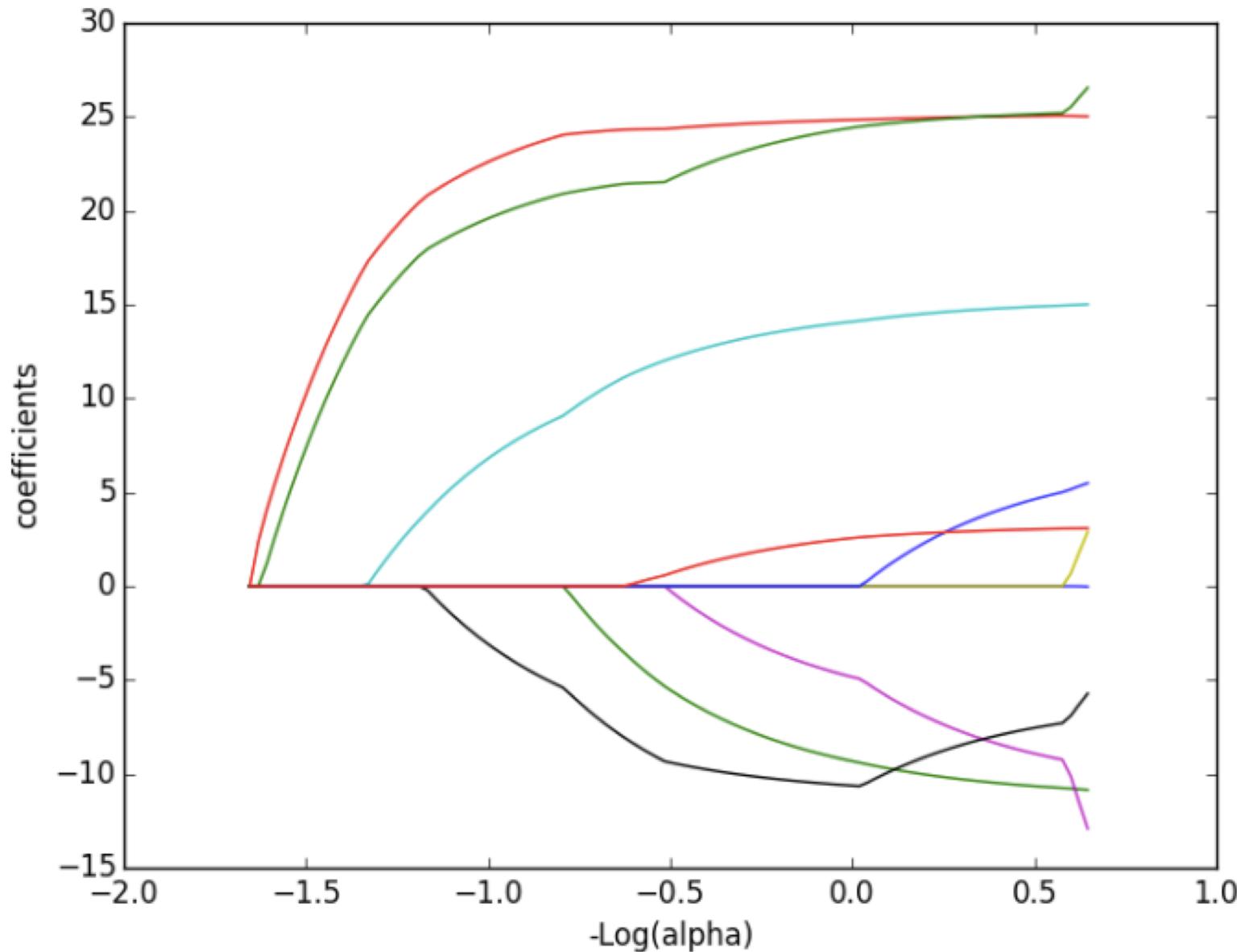
Lasso regression

$$PRSS_{\lambda}^{lasso}(\beta) = \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

- Lasso leads to shrinkage of the coefficients, but even more important, it zeros out the coefficients of the unimportant variables – variable selection!
- This is due to the shape of the L1 penalty
- Challenging optimization problem, several fast algorithms



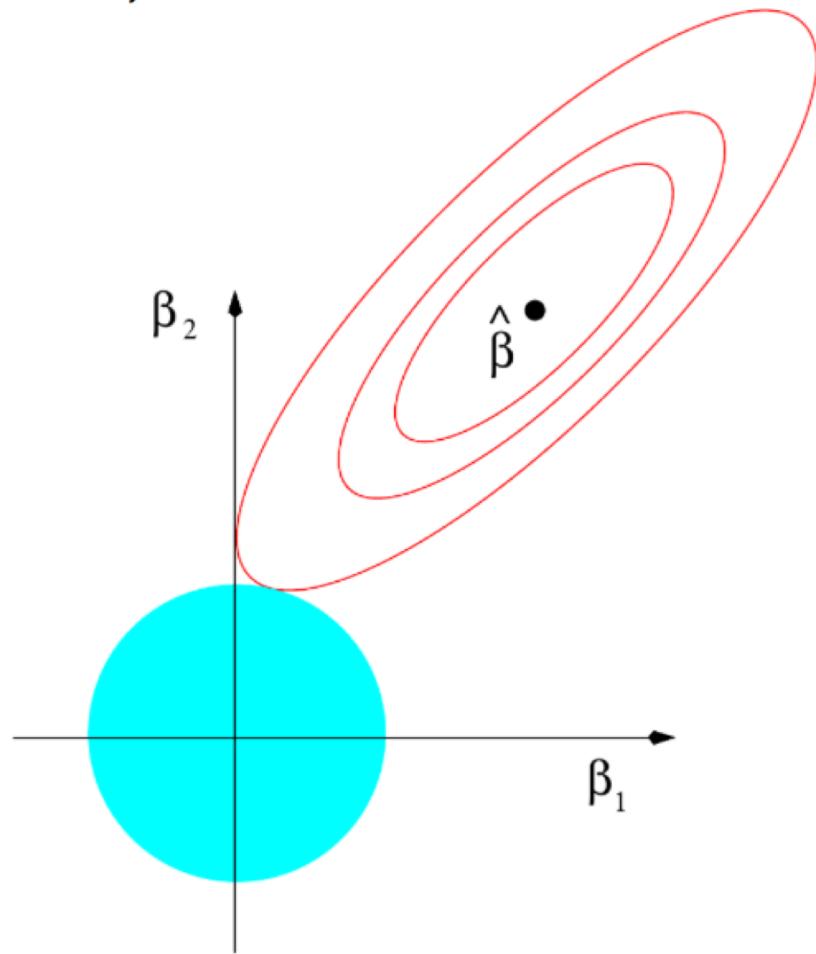
A Lasso plot – the size of λ determines the amount of zero coefficients



Ridge regression

$$PRSS_{\lambda}^{ridge}(\beta) = \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

- Ridge regression only shrinks the coefficients, introducing bias, but reducing variance of the estimators.
- No variable selection
- Explicit solution to the optimization problem



Lasso cont.

Lasso assumes what we call *sparsity*: That only a few of the p covariates matter. If sparsity is true, Lasso will recover it (theorems etc.).

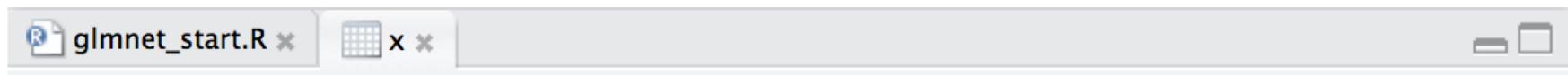
λ might be chosen by the data via *(K-fold) Cross Validation*.

Lasso and/or Ridge in R: Recommend the package `glmnet` (incl. linear, logistic, Poisson, Cox regression)

Install the R-package glmnet:

```
>install.packages("glmnet", repos = "http://cran.us.r-project.org")
>library(glmnet)
```

Data should be organized in a vector **y** ($n \times 1$) (the response variable) and a matrix **x** ($n \times p$) (the p predictor variables)



The screenshot shows the RStudio interface with a data frame titled "glmnet_start.R". The title bar also shows "100 observations of 20 variables". The data frame has 100 rows and 20 columns, labeled V1 through V20. The first few rows of data are visible.

	V1	V2	V3	V4	V5	V6	V7
1	0.27385621	-0.036672202	0.85472694	0.967524215	1.41548975	0.52340587	0.56268818
2	2.24481689	-0.546030016	0.23406507	-1.335030427	1.31307582	0.52127458	-0.6100346
3	-0.12542303	-0.606878202	-0.85392169	-0.148777203	-0.66468279	0.60661641	0.16172065
4	-0.54357344	1.108358273	-0.10424799	1.016526228	0.69990418	1.65501642	0.48996346
5	-1.45939839	-0.274494523	0.11190596	-0.851787700	0.31528387	1.05074928	1.38635753
6	1.06320807	-0.753523175	-1.38255341	1.076226985	0.37003310	1.49872124	-0.3604525
7	0.11584499	-0.966302384	0.27417918	0.018531025	-0.21038723	0.54408945	-2.5054893
8	0.38970673	0.399794284	-0.46666982	0.477628120	0.84829129	0.25743529	-0.0994948

This is a piece of an example of a data matrix **x**, with 20 predictors measured for 100 subjects (so here $p < n$, for comparison with OLS)

R glmnet_start.R x y

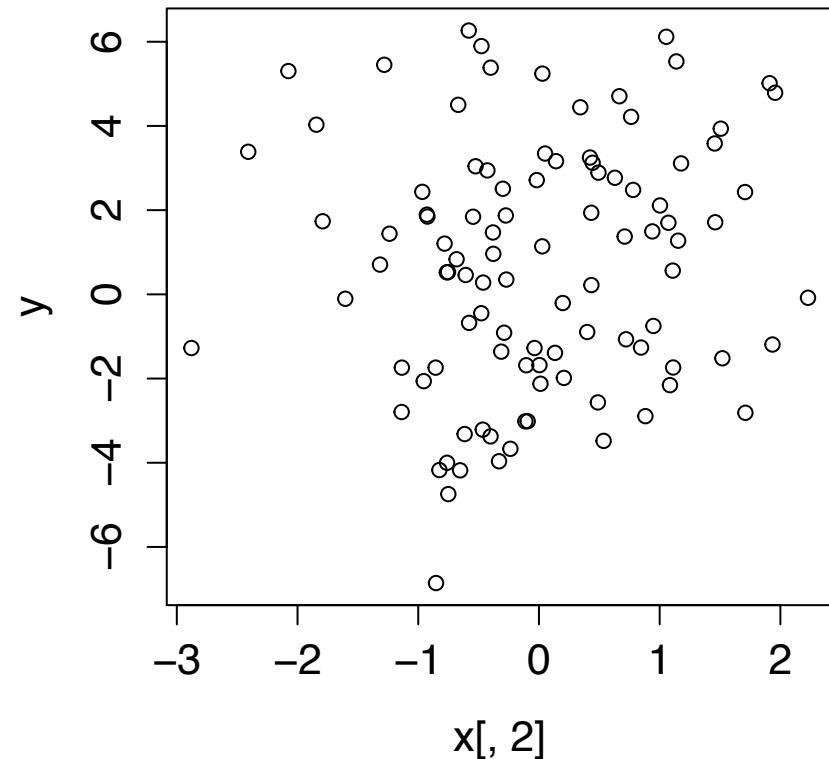
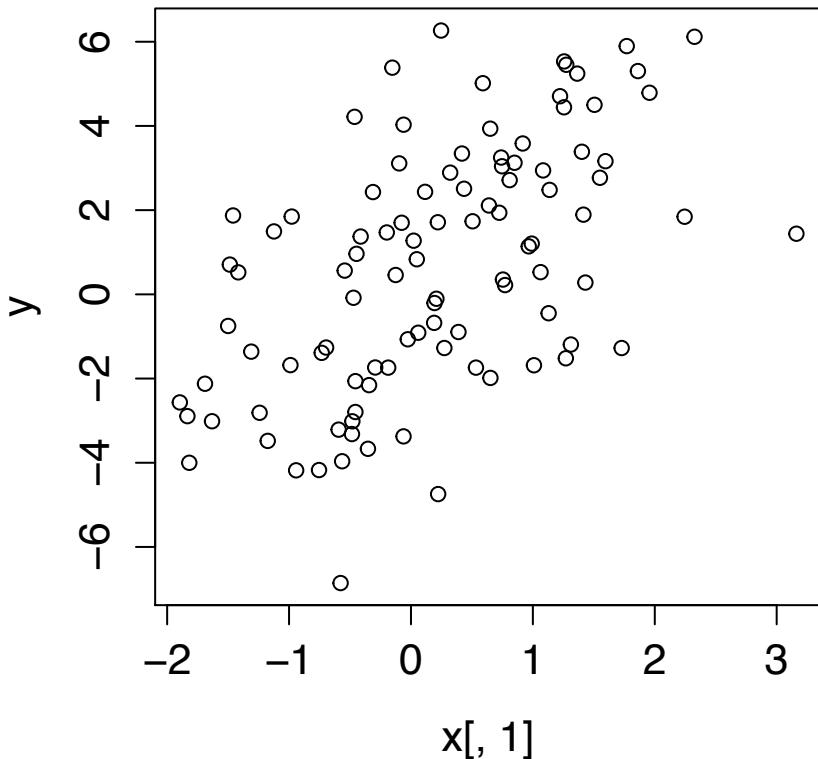
~/glmnet_start.R

100 observations of 1 variables

	V1
1	-1.27488603
2	1.84342510
3	0.45923632
4	0.56404074
5	1.87296326
6	0.52753173
7	2.43465887
8	-0.89459612

And this is the corresponding y ,
with 100 responses

Two first predictors:



Running glmnet with cross validation for the penalty parameter:

Does cross-validation for lambda

Continuous response, normal noise

alpha=1:Lasso
alpha=0:Ridge
0<alpha<1: net

```
>cv.fit.lasso=cv.glmnet(x,y,family="gaussian",alpha=1)
```

```
>lambda.min=cv.fit.lasso$lambda.min
```

```
>beta.lasso=coef(cv.fit.lasso, s=lambda.min)
```

The lambda value that minimizes the CV-curve

The estimated coefficients for lambda.min

```
> beta.lasso
```

21 x 1 sparse Matrix of class "dgCMatrix"

	1
(Intercept)	0.14867414
V1	1.33377821
V2	.
V3	0.69787701
V4	.
V5	-0.83726751
V6	0.54334327
V7	0.02668633
V8	0.33741131
V9	.
V10	.
V11	0.17105029
V12	.
V13	.
V14	-1.07552680
V15	.
V16	.
V17	.
V18	.
V19	.
V20	-1.05278699

LASSO!

OLS

	Estimate	Pr(> t)
(Intercept)	0.109068	0.347598
x1	1.381072	< 2e-16 ***
x2	0.025016	0.811399
x3	0.767490	9.68e-10 ***
x4	0.066767	0.537749
x5	-0.905978	7.07e-12 ***
x6	0.618388	2.28e-08 ***
x7	0.124492	0.248793
x8	0.401052	0.000138 ***
x9	-0.036556	0.732835
x10	0.136530	0.212670
x11	0.251597	0.026115 *
x12	-0.069913	0.532250
x13	-0.049396	0.660097
x14	-1.164018	< 2e-16 ***
x15	-0.147334	0.254664
x16	-0.051572	0.644480
x17	-0.055904	0.597418
x18	0.057081	0.591626
x19	-0.006423	0.944577
x20	-1.148534	2.08e-14 ***

NB! For $p > n$, the OLS cannot be found!!

1. Good luck with the assignment!
2. See you in March!