R Summer School @ AUEB
Linear and Generalized Linear Models

Ioannis Kosmidis
Department of Statistical Science, University College London

i.kosmidis@ucl.ac.uk

Athens University of Economics and Business
24th, 25th June 2014

This is version 0.95 of the slides and is subject to minor changes. The version that will be presented will be made available electronically.

Outline

Aim

- To provide an introduction to regression modelling via linear and generalized linear models using **R**.

Learning outcomes

On successful completion of this module, you will:

- be able to identify settings where regression analysis is useful
- have developed an understanding of the basic methodology underlying regression analyses
- be able to use **R** for carrying out regression analyses and interpret their results
- be able to apply modern techniques and tools associated to regression models

Emphasis will be placed on **ideas, methods, and the associated computational tools rather than on the mathematical details of the topic**.

Reference material

- This set of notes has been influenced largely by the following textbooks:

Fox, J. (2011). *An R Companion to Applied Regression* (2nd ed.). SAGE Publications.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.

Maindonald, J. and W. J. Braun (2010). *Data Analysis and Graphics Using R: An Example-Based Approach* (3rd ed.). Cambridge University Press.

McCullagh, P. and J. Nelder (1989). *Generalized Linear Models* (2nd ed.). Chapman and Hall.

- Furthermore, data sets and materials are used from various **R** packages (see next slide). The efforts of the authors of those packages need to be acknowledged.

Resources

- Delivered resources include:
  - this set of slides
  - an **R** script with code chunks to reproduce the analyses herein
- You may install the **R** packages that are used or mentioned on these slides by typing:

```
# Update the installed packages first
update.packages(ask = FALSE, repos = "http://cran.rstudio.com/")
# Install the packages for this tutorial
RatAUEBpackages <- c("alr3", "brglm", "car", "caret", "elasticnet",
    "fortunes", "ggplot2", "gridExtra", "GGally", "leaps", "lmtest",
    "mfp", "relimp")
install.packages(RatAUEBpackages, repos = "http://cran.rstudio.com/")
```

- The slides contain some exercises. Below is the first:

Exercise

Install the **R** packages that are used in this tutorial.

Loading the required packages

- Before running through the code on these slides we need to load some packages in **R**.

```
library(MASS)
library(alr3)
library(car)
library(relimp)
library(fortunes)
library(mfp)
library(ggplot2)
library(GGally)
library(gridExtra)
library(leaps)
library(elasticnet)
library(caret)
library(brglm)
```

Structure

- Linear regression models
- Generalized linear models
- Model selection

Outline

Simple linear regression: Setting

- **Data**: $(y_1, x_1), \ldots, (y_n, x_n)$.
- **Response**: $y_1, \ldots, y_n$ are observations on the attribute that we wish to "explain".
- **Covariate** (a.k.a. explanatory variable or input): $x_1, \ldots, x_n$ are scalar observations of the covariate which can be used to "explain" the response.
- **Model**:

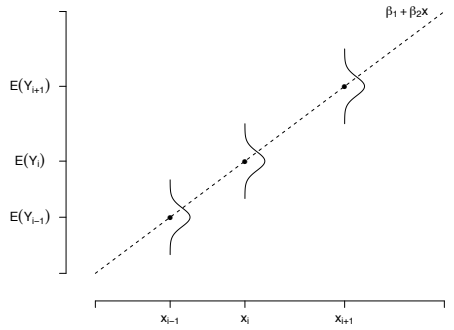
$$Y_i = \beta_1 + \beta_2 x_i + \epsilon_i \quad (i = 1, \ldots, n),$$

where:

- $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. with $\epsilon_i \sim N(0, \sigma^2)$
- β_1, β_2 are unknown regression parameters to be estimated from the data.

Simple linear regression: Expectation

$$E(Y|x) = \beta_1 + \beta_2 x$$



Simple linear regression: Usefulness

With a linear model we can:

- Describe linear relationships and, importantly, nonlinear relationships after transformation.
- Assess the significance of the covariate in explaining the variability in the response.
- Predict new values of the response at given values of the covariate.

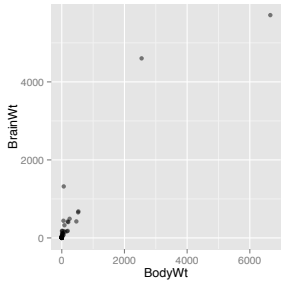
The brain weight data

- **Data**: the average body weight in kilograms and the average brain weight in grams for 62 species of mammals.

```
## Get the brain weight data out of the alr3 package
Brains <- brains
head(Brains)

##            BrainWt  BodyWt
## Arctic_fox    44.5   3.385
## Owl_monkey    15.5   0.480
## Beaver         8.1   1.350
## Cow          423.0 464.983
## Gray_wolf    119.5  36.328
## Goat         115.0  27.660
```

- **Task**: Describe the average brain weight as a function of the average body weight.

The brain weight data

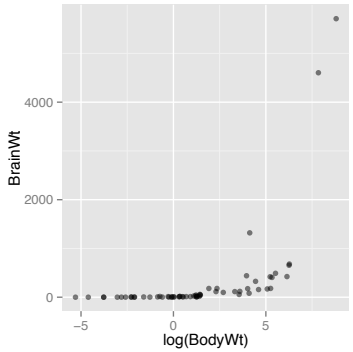


- Let Z_i be the average brain weight and g_i the average body weight for the ith mammal.
- The model

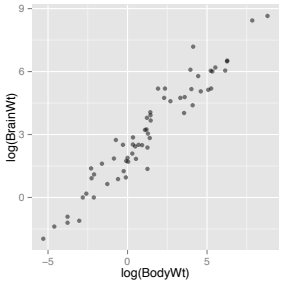
$$Z_i = \gamma_0 + \gamma_1 g_i + \epsilon_i \,,$$

will most probably not fit the data well...

The brain weight data



The brain weight data



- Let Y_i be the **log** brain weight and x_i the **log** average body weight for for the ith mammal.
- The model

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i \,,$$

will most probably fit the transformed data very well!

- The **implied model** is

$$Z_i = \exp(\beta_1) g_i^{\beta_2} \delta_i \,,$$

where $\delta_1, \ldots, \delta_n$ are i.i.d multiplicative errors.

Fitting the simple linear model

- **Least squares estimates**: Minimise the sum of squared errors:

$$(\hat{\beta}_1, \hat{\beta}_2)^T = \arg \min_{(\beta_1,\beta_2)\in\Re^2} \sum_{i=1}^{n} (y_i - \beta_1 - \beta_2 x_i)^2$$

$\rightarrow$

$$\hat{\beta}_2 = C_{xy}/C_{xx} \quad \text{and} \quad \hat{\beta}_1 = \bar{y} - \hat{\beta}_2 \bar{x} \,,$$

where $C_{xy} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$ and $C_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2$.

- The least squares estimates do not depend on the distributional assumption on $\epsilon_1, \ldots, \epsilon_n$.

The `lm` function for fitting linear models

The `lm` function is the interface of **R** for fitting linear models. It has the following general form:

```
lm(formula, data, subset, weights, ...)
```

- `formula` is a model formula specifying the regression equation.
- `data` is an optional data frame that contains the data required by `formula`. If not specified then `lm` searches for those in the search path.
- `subset` is an optional argument (logical or a vector of subscripts) that specifies which observations should or should not be used when fitting the model.
- `weights` is an optional argument that specifies weights for weighted least squares estimation.
- `...` are other arguments that can be passed to `lm` (**type `?lm` for details**)

Model formulas in **R**

A model formula for the `lm` function has the general form

```
y ~ model
```

The response `y` (e.g. `BrainWt`) is modelled by a series of "terms" specified symbolically in `model`. That series of "terms" is separated by special operators (e.g. `+` and `-`). The main operations in `model` are

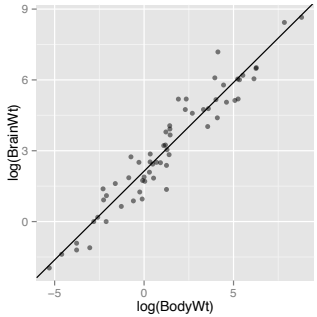
Model	Interpretation
`X + Z`	Include both `A` and `B`
`X - Z`	Exclude `Z` from `X`
`X:Z`	Include all "interactions" of `X` and `Z`
`X*Z`	`X + Z + X:Z`

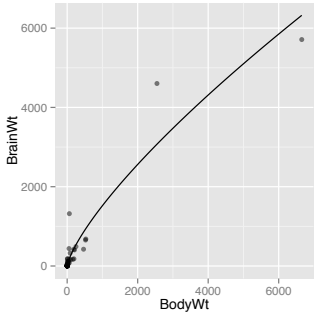
- The number `1` is reserved for the intercept, which will be included unless explicitly excluded (e.g. `BodyWt - 1`).
- To do arithmetic in the `model` it is necessary to "protect" the operation within a function call (e.g. `log(BrainWt)` or `I(BrainWt^2)` where `I` is the identity function).

More details and other special operations are given in `?formula`.

The brain weight data

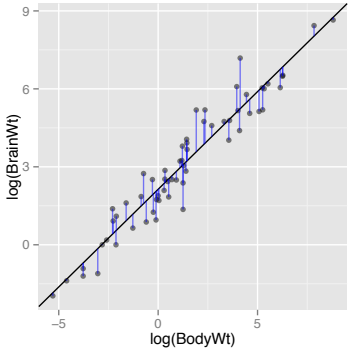
```
# Fit the simple linear model to the brain weight data
LogWeightsLM <- lm(log(BrainWt) ~ log(BodyWt), data = Brains)
# Extract the least squares estimates for beta1 and beta2
(LogWeightsCoefs <- coef(LogWeightsLM))

## (Intercept) log(BodyWt)
##      2.1348      0.7517
```





Fitted line

→ Least squares brings the model close to the data: **the sum of squared distances of the data-points from the fitted line is the smallest amongst all possible lines.**



- Fitted value:

$$\hat{Y}_i = \hat{\beta}_1 + \hat{\beta}_2 x_i$$

- Residual:

$$E_i = Y_i - \hat{Y}_i$$

The residuals are the lengths of the **blue** segments.

Inference

Under the assumption that $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. with $\epsilon_1 \sim N(0, \sigma^2)$, it can be shown that

- $\hat{\beta}_2 \sim N(\beta_2, \sigma^2 / C_{xx})$
- The **residual sample variance**

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^n E_i^2}{n-2}$$

is a consistent and unbiased estimator for σ^2.

- The t statistic

$$T = \frac{\hat{\beta}_2 - \beta_2}{\hat{\sigma}/\sqrt{C_{xx}}}$$

has a t_{n-2} distribution.

Hypothesis testing

- **Is the brain weight significant in explaining body weight?**
- Mathematical hypothesis: $H_0 : \beta_2 = 0$ versus $H_1 : \beta_2 \neq 0$
- Same as **comparing** the model

$$Y_i = \beta_1 + \beta_2 x_i + \epsilon_i \quad (\text{or } Z_i = \exp(\beta_1) g_i^{\beta_2} \delta_i)$$

with the **intercept-only model**

$$Y_i = \beta_1 + \epsilon_i \quad (\text{or } Z_i = \exp(\beta_1)\delta_i).$$

- We can test H_0 using the T statistic.

```
coef(summary(LogWeightsLM))

##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   2.1348    0.09604   22.23 1.183e-30
## log(BodyWt)   0.7517    0.02846   26.41 9.834e-35
```

- Body weight is very significant in explaining brain weight (the t-value for β_2 is 26.41 on 60 df giving a tiny p-value).

Confidence intervals

- Confidence interval for β_2 at level $100(1-\alpha)\%$:

$$\left(\hat{\beta}_2 - t_{n-2;1-\alpha/2}\frac{\hat{\sigma}}{\sqrt{C_{xx}}}, \hat{\beta}_2 + t_{n-2;1-\alpha/2}\frac{\hat{\sigma}}{\sqrt{C_{xx}}}\right).$$

```
confint(LogWeightsLM, level = 0.95)

##                2.5 % 97.5 %
## (Intercept) 1.9427 2.3269
## log(BodyWt) 0.6948 0.8086
```

- The 95% confidence interval for β_2 does not include zero. Hence, at $\alpha = 0.05$, we reject the hypothesis $H_0 : \beta_2 = 0$.

Exercise

Repeat the analysis of the brain weight data using square roots instead of logarithms.

- What is the implied model for the average weight?
- Fit the linear model on the square roots of the average weights, report the coefficients, and plot the fitted line.
- Assuming that the model assumptions are adequate, test the hypothesis that the slope parameter is zero and construct a 99% confidence interval for it. What do you conclude?
- Do you think that the model fitted on the square roots is as good as the model fitted on logarithms is?

Outline

Normal linear model: Setting

- **Data**: $(y_1, \mathbf{x}_1^T), \ldots, (y_n, \mathbf{x}_n^T)$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^T$.
- **Response**: $y_1, \ldots, y_n$
- **Covariates** (a.k.a. explanatory variables or inputs): $(x_{11}, \ldots, x_{1p})', \ldots, (x_{n1}, \ldots, x_{np})^n$.
- **Model**:

$$Y_i = \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \epsilon_i \quad (i = 1, \ldots, n),$$

where:

- $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. with $\epsilon_i \sim N(0, \sigma^2)$
- $\beta_1, \ldots, \beta_p$ are unknown regression parameters to be estimated from the data.

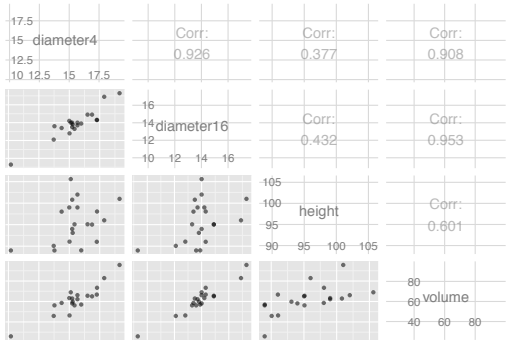
The Trees data

- **Data**: 4 variables measured on each of a sample of 20 trees. The variables are:
  - **diameter4:** the diameter of the trunk (in inches) at four feet from the ground (an easy place to measure!),
  - **diameter16:** the diameter of the trunk (in inches) at sixteen feet from the ground,
  - **height:** the height (in feet) of the tree, and
  - **volume:** the volume of timber (in cubic feet) obtained from the tree.

```
load("RatAUEB_LM_GLM_data.RData")
str(Trees)

## 'data.frame': 20 obs. of  4 variables:
##  $ diameter4 : num  10.2 13.7 15.4 14.4 15 ...
##  $ diameter16: num  9.3 12.1 13.3 13.4 14.2 12.8 14 13.5 14 13.8 ...
##  $ height    : num  89 90.1 95.1 98 99 ...
##  $ volume    : num  25.9 45.9 56.2 58.6 63.4 ...
```


The Trees data

```
(TreesPairs <- ggpairs(Trees, alpha = I(0.5)))
```



- **Task:** Find a predictive model for the volume of timber from all or some of the three other variables.

The Trees data

Candidate models:

$$\text{“volume”} = \beta_1 + \beta_2 \text{“diameter4”} + \beta_3 \text{“diameter16”} + \beta_4 \text{“height”} + \epsilon$$

$$\text{“volume”} = \beta_1 + \beta_2 \text{“diameter4”} + \beta_3 \text{“diameter16”} + \beta_4 log(\text{“height”}) + \epsilon$$

Normal linear model: Matrix form

Key ingredients:

- Model matrix $\mathbf{X}$ ($\dim(\mathbf{X}) = n \times p$):

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}$$

```
FormulaTrees <- volume ~ diameter4 + diameter16 + height
XTrees <- model.matrix(FormulaTrees, data = Trees)
head(XTrees)

##   (Intercept) diameter4 diameter16 height
## 1           1     10.20        9.3  89.00
## 2           1     13.72       12.1  90.07
## 3           1     15.43       13.3  95.08
## 4           1     14.37       13.4  98.03
## 5           1     15.00       14.2  99.00
## 6           1     15.02       12.8  91.05
```

Normal linear model: Matrix form

Key ingredients:

- Response vector $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ $(\dim(\mathbf{Y}) = n \times 1)$

```
(YTrees <- Trees$volume)

##  [1] 25.93 45.87 56.20 58.60 63.36 46.35 68.99 62.91 58.13
## [10] 59.79 56.20 66.16 62.18 57.01 65.62 65.03 66.74 73.38
## [19] 82.87 95.71
```

- Parameters vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ $(\dim(\boldsymbol{\beta}) = p \times 1)$
- Vector of errors $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^T$ $(\dim(\boldsymbol{\epsilon}) = n \times 1)$

The model

$$Y_i = \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \epsilon_i \quad (i = 1, \ldots, n)$$

can be written as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

Exercise

- What is the model matrix and the response vector for the simple linear regression model $Y_i = \beta_1 + \beta_2 x_i + \epsilon_i$?.
- Write down the equations implied by the Normal linear model for the first and fourth tree.

Least squares estimation

- Minimise the sum of squared errors

$$\sum_{i=1}^{n} (Y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$

with respect to $\boldsymbol{\beta} \in \Re^p$.

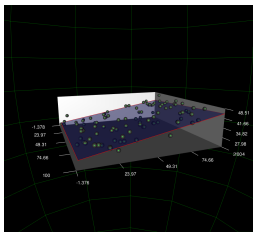
$\rightarrow$ $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$

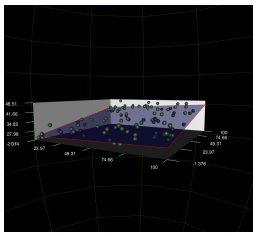
(note that this does not work for $p > n$)

```
drop(solve(t(XTrees) %*% XTrees) %*% (t(XTrees) %*% YTrees))

## (Intercept)   diameter4  diameter16      height
##   -108.5758      1.6258      5.6714      0.6938

TreesLM1 <- lm(volume ~ diameter4 + diameter16 + height, data = Trees)
coef(TreesLM1)

## (Intercept)   diameter4  diameter16      height
##   -108.5758      1.6258      5.6714      0.6938
```


Fitted hyperplane

- The least squares estimates do not depend on the distributional assumption on $\boldsymbol{\epsilon}$.
- Least squares: **the sum of squared distances of the data-points from the fitted hyperplane is the smallest amongst all possible hyperplanes**.
- Fitted values: $\hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{H}\mathbf{Y}$, where $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is the **"hat" matrix**.
- Residuals: $\mathbf{E} = \mathbf{Y} - \hat{\mathbf{Y}} = (\mathbf{I} - \mathbf{H})\mathbf{Y}$





Inference

Under the assumption that $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. with $\epsilon_1 \sim N(0, \sigma^2)$, it can be shown that

- $\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2 \mathbf{V})$ with $\mathbf{V} = (\mathbf{X}^T\mathbf{X})^{-1}$
- $\hat{\boldsymbol{\beta}}$ is the minimum variance unbiased estimator.
- The **residual sample variance**

$$\hat{\sigma}^2 = \frac{\mathbf{E}^T\mathbf{E}}{n-p}$$

is a consistent and unbiased estimator for σ^2

- $\mathbf{E}^T\mathbf{E} = \sum_{i=1}^n E_i^2$ is called **residual sum of squares** and has $\mathrm{df} = n - p$ degrees of freedom.
- The t statistic

$$T_j = \frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}\sqrt{v_{jj}}} \quad (j = 1, \ldots, p),$$

has a t_{n-p} distribution, where v_{jj} is the $(j,j)th$ element of $\mathbf{V}$.

Individual hypotheses tests

- Interest is on hypotheses of the form

$H_0 : \beta_j = 0$ given that $\beta_1, \ldots, \beta_{j-1}, \beta_{j+1}, \ldots, \beta_p$ are in the model.

- We can use the statistic T_j.

E.g. `Trees` data: **Is height significant for explaining volume when the diameters are in the model?**

- For this we need to test $H_0 : \beta_4 = 0$ versus $H_1 : \beta_4 \neq 0$ given that $\beta_1, \beta_2, \beta_3$ are in the model.

Same as comparing the model

$$\text{“volume”} = \beta_1 + \beta_2 \,\text{“diameter4”} + \beta_3 \,\text{“diameter16”} + \beta_4 \,\text{“height”} + \epsilon$$

to

$$\text{“volume”} = \beta_1 + \beta_2 \,\text{“diameter4”} + \beta_3 \,\text{“diameter16”} + \epsilon$$

`summary` method for `lm` objects

```
(TreesLM1Summary <- summary(TreesLM1))

##
## Call:
## lm(formula = volume ~ diameter4 + diameter16 + height, data = Trees)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.255 -1.677 -0.128  1.523  4.999
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -108.576     14.142   -7.68  9.4e-07 ***
## diameter4      1.626      1.026    1.58  0.13261
## diameter16     5.671      1.202    4.72  0.00023 ***
## height         0.694      0.163    4.25  0.00061 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.1 on 16 degrees of freedom
## Multiple R-squared:  0.959, Adjusted R-squared:  0.951
## F-statistic:  125 on 3 and 16 DF,  p-value: 2.59e-11
```


Tree data

Quick conclusions:

- The height of the tree appears to be significant in explaining volume even if the diameters are in the model (t-value of 4.254 on 16 df giving a p-value < 0.001).
- The diameter at 4 feet from the ground seems to be insignificant in explaining volume when diameter at 16 feet from the ground and the height of the tree are included in the model (t-value of 1.58 on 16 df giving a p-value of 0.132)

`confint` method for `lm` objects

Confidence interval for β_j at level $100(1-\alpha)\%$:

$$\left(\hat{\beta}_j - t_{n-p;1-\alpha/2}\hat{\sigma}\sqrt{v_{jj}}, \hat{\beta}_j + t_{n-p;1-\alpha/2}\hat{\sigma}\sqrt{v_{jj}}\right)$$

```
confint(TreesLM1, level = 0.95)

##                 2.5 %  97.5 %
## (Intercept) -138.5559 -78.596
## diameter4     -0.5492   3.801
## diameter16     3.1227   8.220
## height         0.3481   1.039
```


Expected values

Interest is on inference for the expected value $\mu_{\mathbf{z}}$ of the response at a setting $\mathbf{z} = (z_1, \ldots, z_p)^T$.

- A consistent and unbiased estimator of the expected value is

$$\hat{\mu}_{\mathbf{z}} = \hat{\beta}_1 z_1 + \ldots + \hat{\beta}_p z_p \,.$$

- A $100(1-\alpha)\%$ confidence interval for $\mu_{\mathbf{z}}$ is

$$\left(\hat{\mu}_{\mathbf{z}} - t_{n-p;1-\alpha/2}\hat{\sigma}\sqrt{\mathbf{z}^T\mathbf{V}\mathbf{z}}, \hat{\mu}_{\mathbf{z}} + t_{n-p;1-\alpha/2}\hat{\sigma}\sqrt{\mathbf{z}^T\mathbf{V}\mathbf{z}}\right) ,$$

E.g. Provide a confidence interval for the expected volume of timber from trees that have diameters 14 and 10 inches at 4 and 16 feet from the ground, respectively and height 90 feet.

$\rightarrow$ $\mathbf{z} = (1, 14, 10, 90)^T$.

`predict` method for `lm` objects: Confidence interval

```
newdata <- data.frame(diameter4 = 14, diameter16 = 10, height = 90)
predict(TreesLM1, newdata = newdata, interval = "confidence")

##     fit   lwr   upr
## 1 33.34 26.45 40.22
```

We can also use `predict` for multiple observations

```
(newdata1 <- rbind(newdata, c(13.5, 8.2, 82)))

##   diameter4 diameter16 height
## 1      14.0       10.0     90
## 2      13.5        8.2     82

predict(TreesLM1, newdata = newdata1, interval = "confidence")

##     fit    lwr   upr
## 1 33.34 26.453 40.22
## 2 16.77  6.351 27.18
```


Predicted values

Interest is on inference on the predicted value of the response at a setting $\mathbf{z} = (z_1, \ldots, z_p)^T$. The predicted value at $\mathbf{z}$ is

$$\hat{Y}_{\mathbf{z}} = \hat{\mu}_{\mathbf{z}} + \epsilon \,,$$

- A $100(1-\alpha)\%$ prediction interval at $\mathbf{z}$ is

$$\left(\hat{\mu}_{\mathbf{z}} - t_{n-p;1-\alpha/2}\hat{\sigma}\sqrt{1+\mathbf{z}^T\mathbf{V}\mathbf{z}}, \hat{\mu}_{\mathbf{z}} + t_{n-p;1-\alpha/2}\hat{\sigma}\sqrt{1+\mathbf{z}^T\mathbf{V}\mathbf{z}}\right) .$$

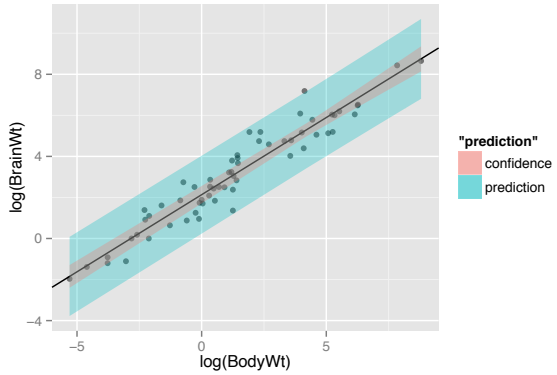
E.g. Provide a prediction interval for a newly discovered tree that has diameters 12.35 and 11.77 inches at 4 and 16 feet from the ground, respectively and height 89.2 feet.

$\rightarrow$ $\mathbf{z} = (1, 12.35, 11.77, 89.2)^T$.

`predict` method for `lm` objects: Prediction interval

```
newdata <- data.frame(diameter4 = 12.35, diameter16 = 11.77,
    height = 89.2)
predict(TreesLM1, newdata = newdata, interval = "prediction")

##     fit   lwr   upr
## 1 40.14 32.72 47.56
```


Brain weight data: 99% prediction & confidence intervals



Model comparisons: Rationale

- Interest is on hypotheses of the form

$$H_0 : \beta_s = \beta_t = \ldots = \beta_u = 0$$

for some distinct $s, t, \ldots, u \in \{1, \ldots, p\}$ (can a subset of the parameters be omitted if the other are in the model?).

E.g.: Does diameter affect volume of timber if height is taken into account?

- Can we omit `diameter4` and `diameter16` from `TreesLM1`?
- Compare the models:

Full: "volume" $= \beta_1 + \beta_2$ "diameter4" $+ \beta_3$ "diameter16" $+ \beta_4$ "height" $+ \epsilon$

Nested: "volume" $= \beta_1 + \beta_4$ "height" $+ \epsilon$

If no evidence against $H_0 : \beta_2 = \beta_3 = 0$ then, abiding to the **parsimony principle**, we can live with the smaller, nested model.

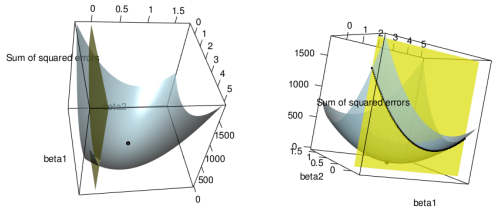
Model comparisons: How?

- The **residual sum of squares** of the full model (RSS_{full}) is always smaller or equal to that of the nested model (RSS_{nest}).

$$\mathrm{RSS}_{full} \geq \mathrm{RSS}_{nest} .$$

(convexity of the residual sum of squares + column space of X_{nest} is $\subseteq$ of column space of X_{full}).

$\rightarrow$ Check if $\mathrm{RSS}_{full} - \mathrm{RSS}_{nest}$ is **"significantly" large**.



Model comparisons: The F-test

Hypothesis:

- Hypothesis: $H_0 : \beta_s = \beta_t = \ldots = \beta_u = 0$,
- **Test statistic:**

$$F = \frac{\frac{\mathrm{RSS}_{nest} - \mathrm{RSS}_{full}}{\mathrm{df}_{nest} - \mathrm{df}_{full}}}{\frac{\mathrm{RSS}_{full}}{\mathrm{df}_{full}}}$$

- The F statistic is the **relative increase in residual sum of squares when moving from the full to the nested model** scaled by $\mathrm{df}_{full}/(\mathrm{df}_{nest} - \mathrm{df}_{full})$.
- Under H_0,

$$F \sim F_{\mathrm{df}_{nest} - \mathrm{df}_{full}, \mathrm{df}_{full}}$$

Model comparisons: Does diameter affect volume?

- Compare the models:

Full: "volume" $= \beta_1 + \beta_2$ "diameter4" $+ \beta_3$ "diameter16" $+ \beta_4$ "height" $+ \epsilon$

Nested: "volume" $= \beta_1 + \beta_4$ "height" $+ \epsilon$

```
# Remove the diameters from the full model and refit.
TreesLM2 <- update(TreesLM1, ~. - diameter4 - diameter16)
# Same as TreesLM2 <- lm(volume ~ height, data = Trees). Now,
# use the anova function to compare full vs nested
anova(TreesLM2, TreesLM1)

## Analysis of Variance Table
##
## Model 1: volume ~ height
## Model 2: volume ~ diameter4 + diameter16 + height
##   Res.Df  RSS Df Sum of Sq   F  Pr(>F)
## 1     18 2392
## 2     16  153  2      2238 117 2.9e-10 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


F-test for one parameter and t-test

The F-test is equivalent to the t-test when testing for a **single** parameter ($F = T^2$ for the statistics!).

```
# E.g. Test whether the slope is zero in LogWeightsLM
coef(summary(LogWeightsLM))["log(BodyWt)", ]

##   Estimate Std. Error    t value   Pr(>|t|)
##  7.517e-01  2.846e-02  2.641e+01  9.834e-35

anova(update(LogWeightsLM, ~. - log(BodyWt)), LogWeightsLM)

## Analysis of Variance Table
##
## Model 1: log(BrainWt) ~ 1
## Model 2: log(BrainWt) ~ log(BodyWt)
##   Res.Df RSS Df Sum of Sq   F Pr(>F)
## 1     61 365
## 2     60  29  1       336 697 <2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Model comparisons: Testing for all terms

- A useful test is testing whether all terms can be omitted from the model and be left with an intercept-only model. That is

$$H_0 : \beta_2 = \ldots = \beta_p = 0\,.$$

In other words: Is the regression worthwhile?

Full: $Y_i = \beta_1 + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon_i$

Nested: $Y_i = \beta_1 + \epsilon_i$

- If we do not have evidence against H_0 then we really need to rethink our covariates.

Model comparisons: Testing for all terms

```
TreesLM1Summary

##
## Call:
## lm(formula = volume ~ diameter4 + diameter16 + height, data = Trees)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.255 -1.677 -0.128  1.523  4.999
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -108.576     14.142   -7.68  9.4e-07 ***
## diameter4      1.626      1.026    1.58  0.13261
## diameter16     5.671      1.202    4.72  0.00023 ***
## height         0.694      0.163    4.25  0.00061 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.1 on 16 degrees of freedom
## Multiple R-squared:  0.959, Adjusted R-squared:  0.951
## F-statistic:  125 on 3 and 16 DF,  p-value: 2.59e-11

# Also try anova(TreesLM1, update(TreesLM1, . ~ 1))
```


Coefficient of determination

Write $\mathrm{RSS}_{nest} = \mathrm{RSS}_{full} + \mathrm{SS}_H$. When testing for all terms:

- RSS_{nest} is the residual sum of squares for the intercept-only model and is equal to

$$\mathrm{SS}_{total} = \sum_{i=1}^{n} (Y_i - \bar{Y})^2 .$$

- SS_H is the "sum of squares due to regression" (called SS_{regr}).

$$\mathrm{SS}_{total} = \mathrm{SS}_{resid} + \mathrm{SS}_{regr}$$

where SS_{resid} is RSS_{full}.

- The total variation of the response observations (about their mean) is decomposed into variation explained by the regression model SS_{regr} plus the unexplained variation SS_{resid}.

Coefficient of determination

- Built a measure that captures **the percentage of variation in the response observations (about their mean) that can be explained by the covariates in the model.**
-

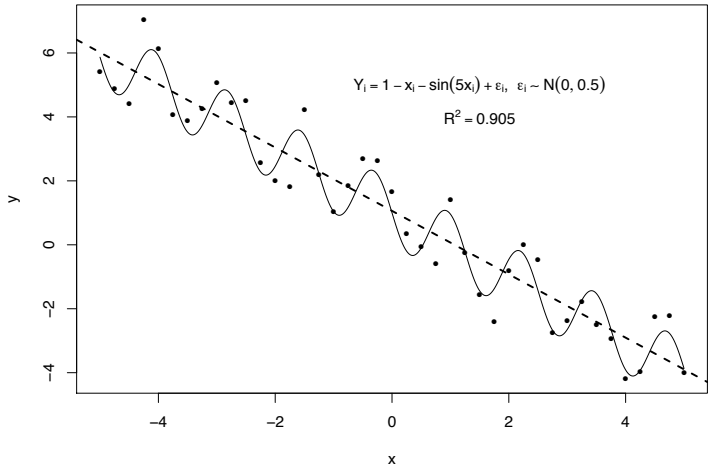
$$R^2 = \frac{\mathrm{SS}_{regr}}{\mathrm{SS}_{total}} = 1 - \frac{\mathrm{SS}_{resid}}{\mathrm{SS}_{total}}$$

- The fact $0 \leq \mathrm{SS}_{resid} \leq \mathrm{SS}_{total}$ implies that $0 \leq R^2 \leq 1$ (can be made exactly 1 if $p = n$).
- It can be shown than $R^2 = \hat{\rho}^2_{\mathbf{Y},\hat{\mathbf{Y}}}$ (square of the sample correlation of the responses and the fitted values).
- The fact that the residual sum of squares decreases when new covariates are added implies that R^2 **always increases by the addition of new covariates**.

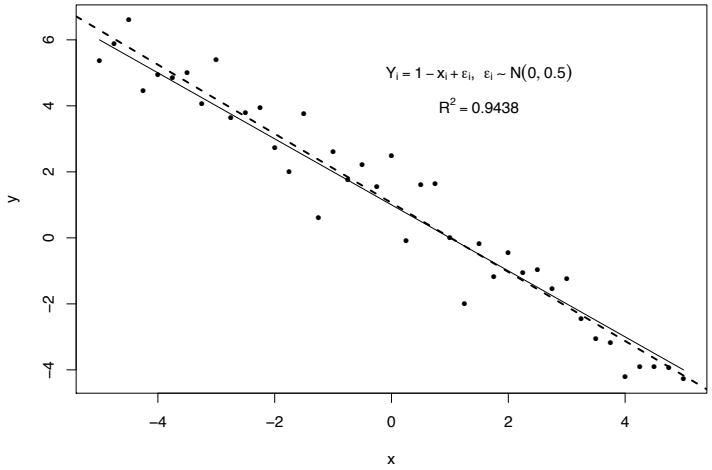
Coefficient of determination

- Interpretation:
  - R^2 close to 0: the covariates provide little information on explaining the variability in the response observations.
  - R^2 close to 1: the covariates explain most of the variability in the responses.
- Misinterpretation:
  - When R^2 is used as a measure of "goodness-of-fit" or "predictive quality".

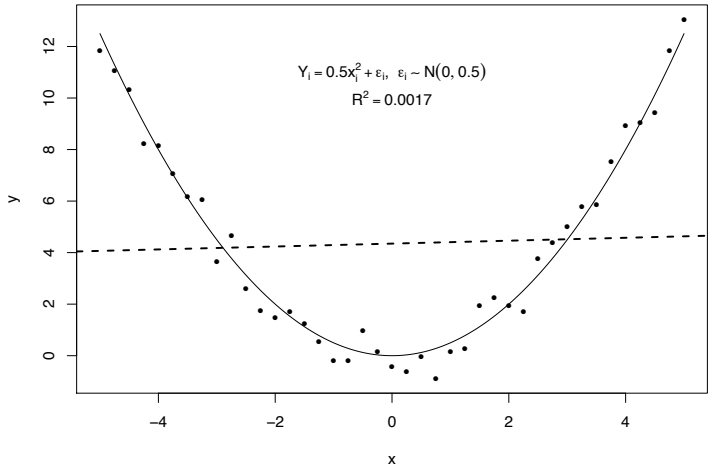
Wrong model but high R^2



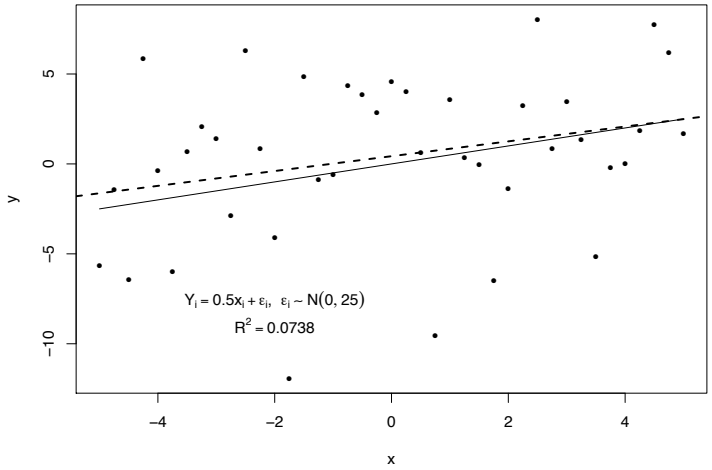
Correct model and high R^2



Wrong model and low R^2



Correct model and low R^2



Exercise

Use the `Trees` data to find a nested model that is an as good description of volume as the model with `diameter4`, `diameter16` and `height` is. Try to ensure that the new model does not sacrifice much in terms of how much of the variation in volume it explains.

Categorical variables as explanatory

```
str(Duncan)

## 'data.frame': 45 obs. of  4 variables:
##  $ type     : Factor w/ 3 levels "bc","prof","wc": 2 2 2 2 2 2 2 2 3 2 ...
##  $ income   : int  62 72 75 55 64 21 64 80 67 72 ...
##  $ education: int  86 76 92 90 86 84 93 100 87 86 ...
##  $ prestige : int  82 83 90 76 90 87 93 90 52 88 ...

head(Duncan)

##            type income education prestige
## accountant prof     62        86       82
## pilot      prof     72        76       83
## architect  prof     75        92       90
## author     prof     55        90       76
## chemist    prof     64        86       90
## minister   prof     21        84       87
```


Categorical variables as explanatory

```
DuncanPairs <- ggpairs(Duncan, color = "type", alpha = I(0.5))
```

- Explain `prestige` in terms of `type`, `income`, `education`

Categorical variables as explanatory: Dummy variables

- `type` cannot be included directly as a covariate in a regression model because it is a **factor** and its value is one of the labels (**levels**) "blue-collar", "professional", "white-collar" ("bc", "prof", "wc" in short).
- One way to do so is using **dummy variables**.
- **Treatment contrasts:** To include a factor with q levels as covariate information in **a model with intercept**, use $q - 1$ dummy variables. The omitted level is called a **baseline level**.

E.g. Dummy variables for `type`:

$$T_{i1} = \begin{cases} 1, & \text{if } \texttt{type}_i = \text{"prof"} \\ 0, & \text{else} \end{cases}, \quad T_{i2} = \begin{cases} 1, & \text{if } \texttt{type}_i = \text{"bc"} \\ 0, & \text{else} \end{cases}$$

- There are several types of **contrasts** that can be used to code categorical variables (type `?contrasts` for information).

Categorical variables as explanatory: Dummy variables

- **R** can set the model matrix for us (by default **R** uses the first level as baseline).

```
## The model matrix is
head(model.matrix(prestige ~ income + education + type, data = Duncan),
    n = 12)

##            (Intercept) income education typeprof typewc
## accountant           1     62        86        1      0
## pilot                1     72        76        1      0
## architect            1     75        92        1      0
## author               1     55        90        1      0
## chemist              1     64        86        1      0
## minister             1     21        84        1      0
## professor            1     64        93        1      0
## dentist              1     80       100        1      0
## reporter             1     67        87        0      1
## engineer             1     72        86        1      0
## undertaker           1     42        74        1      0
## lawyer               1     76        98        1      0
```


Categorical variables as explanatory: Implied equations

```
DuncanLM <- lm(prestige ~ income + education + type, data = Duncan)
summary(DuncanLM)

##
## Call:
## lm(formula = prestige ~ income + education + type, data = Duncan)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14.89  -5.74  -1.75   5.44  28.97
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1850     3.7138   -0.05   0.9605
## income        0.5975     0.0894    6.69  5.1e-08 ***
## education     0.3453     0.1136    3.04   0.0042 **
## typeprof     16.6575     6.9930    2.38   0.0221 *
## typewc      -14.6611     6.1088   -2.40   0.0211 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.74 on 40 degrees of freedom
## Multiple R-squared:  0.913, Adjusted R-squared:  0.904
## F-statistic:  105 on 4 and 40 DF,  p-value: <2e-16
```


Categorical variables as explanatory: Implied equations

- We can now fit the model

$$P_i = \beta_1 + \beta_2 I_i + \beta_3 E_i + \beta_4 T_{i1} + \beta_5 T_{i2} + \epsilon_i \quad (i = 1, \ldots, 45),$$

where P_i, I_i, and E_i are the prestige, the income and the education for the ith profession, respectively.

type	Implied equation
"bc"	$P_i = \beta_1 + \beta_2 I_i + \beta_3 E_i + \epsilon_i$
"prof":	$P_i = (\beta_1 + \beta_4) + \beta_2 I_i + \beta_3 E_i + \epsilon_i$
"wc':	$P_i = (\beta_1 + \beta_5) + \beta_2 I_i + \beta_3 E_i + \epsilon_i$

- Hence, the model implies three **parallel** planes in the 3-dimensional space spanned by `prestige`, `income` and `education`.

Exercise

Use **R** to fit the model

$$P_i = \beta_1 + \beta_3 E_i + \beta_4 T_{i1} + \beta_5 T_{i2} + \epsilon_i \,.$$

How many lines does this model imply on the scatterplot of `prestige` versus `education`? Extract and plot the fitted lines.

Categorical variables as explanatory: Significance

- The significance of a factor needs to be assessed by comparing the model with all dummy variables versus the model without the dummy variables (e.g. using an F-test).
- The `drop1` method does this for the terms in `formula`, dropping one term each time ("Type II" sums of squares).

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

## Single term deletions
##
## Model:
## prestige ~ income + education + type
##           Df Sum of Sq  RSS AIC F value  Pr(>F)
## <none>                 3798 210
## income     1      4246 8044 241   44.72 5.1e-08 ***
## education  1       877 4675 217    9.24  0.0042 **
## type       2      3709 7507 236   19.53 1.2e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Exercise

Compare the output of

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

to the outputs of

```
summary(DuncanLM)
```

and

```
anova(update(DuncanLM, ~. - type), DuncanLM)
```

Detect and explain the correspondences.

Interaction of a categorical and a continuous variable

- We may want to allow **the effect of `education` and `income` on `prestige` to be different for each level of `type`**. In other words we want to have hyperplanes that are not parallel.
- This can be done using **interactions**.
- Including the interactions of `type` with the other variables:

$$P_i = \beta_1 + \beta_2 I_i + \beta_3 E_i + \beta_4 T_{i1} + \beta_5 T_{i2} + \beta_6 (I_i T_{i1}) + \beta_7 (I_i T_{i2}) + \beta_8 (E_i T_{i1}) + \beta_9 (E_i T_{i2}) + \epsilon_i$$

- $(I_i T_{i1})$, $(I_i T_{i2})$ and $(E_i T_{i1})$, $(E_i T_{i2})$ describe the interaction of `type` with `income` and `education`, respectively.

type	Implied equation
"bc"	$P_i = \beta_1 + \beta_2 I_i + \beta_3 E_i + \epsilon_i$
"prof":	$P_i = (\beta_1 + \beta_4) + (\beta_2 + \beta_6) I_i + (\beta_3 + \beta_8) E_i + \epsilon_i$
"wc':	$P_i = (\beta_1 + \beta_5) + (\beta_2 + \beta_7) I_i + (\beta_3 + \beta_9) E_i + \epsilon_i$

Interaction of a categorical and a continuous variable

```
head(model.matrix(prestige ~ (income + education) * type, data = Duncan), n = 12)

##             (Intercept) income education typeprof typewc income:typeprof
## accountant            1     62        86        1      0              62
## pilot                 1     72        76        1      0              72
## architect             1     75        92        1      0              75
## author                1     55        90        1      0              55
## chemist               1     64        86        1      0              64
## minister              1     21        84        1      0              21
## professor             1     64        93        1      0              64
## dentist               1     80       100        1      0              80
## reporter              1     67        87        0      1               0
## engineer              1     72        86        1      0              72
## undertaker            1     42        74        1      0              42
## lawyer                1     76        98        1      0              76
##             income:typewc education:typeprof education:typewc
## accountant              0                86                0
## pilot                   0                76                0
## architect               0                92                0
## author                  0                90                0
## chemist                 0                86                0
## minister                0                84                0
## professor               0                93                0
## dentist                 0               100                0
## reporter               67                 0               87
## engineer                0                86                0
## undertaker              0                74                0
## lawyer                  0                98                0
```

Interaction of a categorical and a continuous variable

```
DuncanLMinteraction <- lm(prestige ~ (income + education) * type, data = Duncan)
summary(DuncanLMinteraction)


##
## Call:
## lm(formula = prestige ~ (income + education) * type, data = Duncan)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.263  -5.534  -0.243   5.106  22.520
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -3.9505     6.7940   -0.58    0.565
## income               0.7834     0.1307    5.99  7.1e-07 ***
## education            0.3196     0.2798    1.14    0.261
## typeprof            32.0078    14.1092    2.27    0.029 *
## typewc              -7.0432    20.6383   -0.34    0.735
## income:typeprof     -0.3691     0.2039   -1.81    0.079 .
## income:typewc       -0.3603     0.2596   -1.39    0.174
## education:typeprof   0.0186     0.3184    0.06    0.954
## education:typewc     0.1068     0.3622    0.29    0.770
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.65 on 36 degrees of freedom
## Multiple R-squared:  0.923, Adjusted R-squared:  0.906
## F-statistic: 54.2 on 8 and 36 DF,  p-value: <2e-16
```


Interaction of a categorical and a continuous variable

- Again `drop1` can be used to test for the significance of the **interaction terms**.
- Caution: Removing one of the "main effects" from the model (i.e. `education`, `income`, `type`) while having an interaction term with it hardly ever makes sense.
- **R** knows this:

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

## Single term deletions
##
## Model:
## prestige ~ (income + education) * type
##                 Df Sum of Sq  RSS AIC F value Pr(>F)
## <none>                       3351 212
## income:type      2       372 3723 213    2.00   0.15
## education:type   2        12 3362 208    0.06   0.94
```


Exercise

Use **R** to fit the model that has response `prestige` and covariates `education`, `type` and their interaction. How many lines does this model imply on the scatterplot of `prestige` versus `education`? Extract and plot the fitted lines.

Testing for relative importance

- Distinct groups of covariates:
  - E.g. covariates on hospital characteristics and covariates on patient characteristics in predicting some health index
  - E.g. a group of dummy variables and the remaining covariates as in for the `Duncan` data.
- In such cases we may be interested in testing the contribution of a group A of covariates relative to that of a group B.

E.g.: Are the diameters more important than height in explaining volume of timber?

Testing for relative importance

- Write

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon = \mathbf{X}_A\boldsymbol{\beta}_A + \mathbf{X}_B\boldsymbol{\beta}_B + \mathbf{X}_C\boldsymbol{\beta}_C + \epsilon$$

where

$\mathbf{X}_A$ $[n\times p_A]$, X_B $[n\times p_B]$ and $\mathbf{X}_C$ $[n\times p_C]$ are matrices such that

$$\mathbf{X} = \left[\ \mathbf{1} \,\middle|\, \mathbf{X}_A \,\middle|\, \mathbf{X}_B \,\middle|\, \mathbf{X}_C\ \right]$$

(i.e. $p = 1 + p_A + p_B + p_C$),
$\beta = (\beta_1, \boldsymbol{\beta}_A^T, \boldsymbol{\beta}_B^T, \boldsymbol{\beta}_C^T)^T$.

→ We want to test whether $\mathbf{X}_A\boldsymbol{\beta}_A$ contributes more than $\mathbf{X}_B\boldsymbol{\beta}_B$ in explaining $\mathbf{Y}$.

Testing for relative importance

- Suppose that each of the covariates has mean zero across the observations. The quantity

$$\omega_{AB} = \frac{\boldsymbol{\beta}_A^T\mathbf{X}_A^T\mathbf{X}_A\boldsymbol{\beta}_A}{\boldsymbol{\beta}_B^T\mathbf{X}_B^T\mathbf{X}_B\boldsymbol{\beta}_B}$$

is the ratio of the variances of the contributions of the covariates in group A and the covariates in group B.
- We wish to test $H_0 : \omega_{AB} = 1$.
- Under H_0 and using the Normality of $\hat{\beta}$, it can be shown that

$$\frac{\log\hat{\omega}_{AB}}{\sqrt{\hat{v}_{AB}}} \overset{\text{appr}}{\sim} N(0,1)\,,$$

where $\overset{\text{appr}}{\sim}$ means approximately in a large-sample sense, and $\hat{v}_{AB}$ is the estimated variance of $\hat{\omega}_{AB}$ (with a closed-form expression coming from the **delta method**).

Testing for relative importance

Are the diameters more important than height in explaining volume?

```
# Perhaps also check the reference in ?relimp
relimp(TreesLM1, set1 = c(2, 3), set2 = c(4))  # 1 is the Intercept!!

##
## Relative importance summary for model
##     lm(formula = volume ~ diameter4 + diameter16 + height, data = Trees)
##
##        Numerator effects ("set1")
## 1                      diameter4
## 2                     diameter16
##        Denominator effects ("set2")
## 1                          height
## 2
##
## Ratio of effect standard deviations: 3.57
## Log(sd ratio):                 1.273   (se 0.27)
##
## Approximate 95% confidence interval for log(sd ratio): (0.744,1.801)
## Approximate 95% confidence interval for sd ratio:      (2.105,6.056)
```


Sequential tests for regression coefficients

- Key question: **What is the contribution of each of the covariates in the reduction of SS_{total}?**

Model	Residual sum of squares	df
$Y_i = \beta_1 + \epsilon_i$	RSS_1	$n-1$
$Y_i = \beta_1 + \beta_2 x_{i2} + \epsilon_i$	RSS_2	$n-2$
$\vdots$	$\vdots$	
$Y_i = \beta_1 + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon_i$	RSS_p	$n-p$

- **Hypothesis:** $H_0 : \beta_j = 0$ given than $\beta_1, \ldots, \beta_{j-1}$ are in the model.
- Hence, if the reduction in SS_{total} by the jth covariate is significantly large we have evidence against H_0.

Sequential tests for regression coefficients

- Expand $\mathrm{SS}_{regr} = \mathrm{SS}_{total} - \mathrm{SS}_{resid}$ as

$$\mathrm{SS}_{regr} = (\mathrm{RSS}_1 - \mathrm{RSS}_2) + (\mathrm{RSS}_2 - \mathrm{RSS}_3) + \ldots + (\mathrm{RSS}_{p-1} - \mathrm{RSS}_p)$$

- $\mathrm{RSS}_1 - \mathrm{RSS}_2, \ldots, \mathrm{RSS}_{p-1} - \mathrm{RSS}_p$ are called **sequential sums of squares** (a.k.a. "Type I" sums of squares).
- Under the model assumptions, the sequential sums of squares are mutually independent and independent from RSS_p.
- Under H_0

$$F_j = \frac{(\mathrm{RSS}_{j-1} - \mathrm{RSS}_j)/1}{\mathrm{RSS}_p/(n-p)} \sim F_{1,n-p}\,,$$

- Corresponding results apply when considering groups of covariates (e.g. a set of dummy variables) with appropriate changes to the dfs in the F statistic and its distribution.

Sequential tests for regression coefficients

```
anova(TreesLM1)

## Analysis of Variance Table
##
## Response: volume
##            Df Sum Sq Mean Sq F value  Pr(>F)
## diameter4   1   3086    3086   322.1 5.1e-12 ***
## diameter16  1    332     332    34.6 2.3e-05 ***
## height      1    173     173    18.1 0.00061 ***
## Residuals  16    153      10
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

→ All reductions in sum of squares appear to be significant with the largest reduction coming from `diameter4`.

Exercise

Check what `anova(DuncanLM)` returns. The row `type` in the output has two degrees of freedom? Explain why.

Sequential tests for regression coefficients: Order matters

- The order that the covariates enter the model generally affects the size of the sequential sums of squares.

```
anova(lm(volume ~ diameter16 + diameter4 + height, data = Trees))

## Analysis of Variance Table
##
## Response: volume
##            Df Sum Sq Mean Sq F value  Pr(>F)
## diameter16  1   3401    3401   355.0 2.4e-12 ***
## diameter4   1     16      16     1.7 0.21068
## height      1    173     173    18.1 0.00061 ***
## Residuals  16    153      10
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Order does not matter only when $\mathbf{X}$ has mutually orthogonal columns (that is $\mathbf{G}_j^T\mathbf{G}_k = 0$ for every $j \neq k$ $(j, k = 1, \ldots, p)$ where $\mathbf{G}_j$ is the jth column of $\mathbf{X}$).

Sequential tests for regression coefficients: Order matters

This definitely looks like an inconvenient property of sequential sums of squares but it leaves some room for interpretation:

- If a term enters early and significantly reduces SS_{total} this does not really tells us much. One should try and re-order the terms to check whether a significant reduction persists when the same term enters late in the model.
- If a term is entered late in the model and causes a significant reduction in SS_{total} then this is a valuable term in explaining the response.

Exercise

Try several different orderings of the covariates for the `Trees` data. Which variables seem to be important? Why do you think that `diameter4` is rendered unimportant whenever it enters after `diameter16`?

In case you have seen "Type III" sums of squares...

```
fortune("have been fed this nonsense")
```

```
##
## I'm really curious to know why the "two types" of sum of
## squares are called "Type I" and "Type III"! This is a very
## common misconception, particularly among SAS users who have
## been fed this nonsense quite often for all their
## professional lives. Fortunately the reality is much
## simpler. There is, by any sensible reckoning, only ONE type
## of sum of squares, and it always represents an improvement
## sum of squares of the outer (or alternative) model over the
## inner (or null hypothesis) model. What the SAS highly
## dubious classification of sums of squares does is to
## encourage users to concentrate on the null hypothesis model
## and to forget about the alternative. This is always a very
## bad idea and not surprisingly it can lead to nonsensical
## tests, as in the test it provides for main effects "even in
## the presence of interactions", something which beggars
## definition, let alone belief.
##    -- Bill Venables
##       R-help (November 2000)
```


Assumptions of the Normal linear model

The assumptions of the Normal linear model involve:

1. **linearity:** the expected value of the response is a linear combination of the parameters $\beta_1, \ldots, \beta_p$ and p covariates i.e. $E(Y_i) = \beta_1 x_{i1} + \ldots + \beta_p x_{ip} \quad (i = 1, \ldots, n)$
2. **homoscedasticity of error components:** the variance of ϵ_i is σ^2 for all $i \in \{1, \ldots, n\}$,
3. **Normality of error components:** the error components $\epsilon_1, \ldots, \epsilon_n$ are Normally distributed,
4. **independence of error components:** the error components are mutually independent (ϵ_i is independent of ϵ_j for $i \neq j$).

Checking the assumptions

- If any of those assumptions fails then the Normal linear model may not be adequate for the data at hand.

→ It is essential to be able to **diagnose** any problems arising from the invalidity of those assumptions.

- Prove the validity of the above assumptions from data is not possible...
- But we can develop graphical and other methods that **provide indications against those assumptions**.
- The residual $E_i = Y_i - \hat{Y}_i$ is an estimate of the error component and hence it is a key quantity in diagnosing departures from the model assumptions.

The residuals under the model assumptions

- Under the model assumptions the residuals $E_1, \ldots, E_n$ have a multivariate Normal distribution with

$$\begin{aligned} \mathrm{E}(E_i) &= 0 \\ \mathrm{Var}(E_i) &= \sigma^2(1 - h_{ii}) \\ \mathrm{Cov}(E_i, E_j) &= -\sigma^2 h_{ij} \quad (i \neq j) \end{aligned}$$

where h_{ij} is the (i, j)th component of the hat matrix. (recall that $\mathbf{E} = (\mathbf{I} - \mathbf{H})\mathbf{Y}$ with $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$.)

- Hence, under the model assumptions the residuals
  - **have expectation zero**
  - **have unequal variances** (unless $h_{11} = \ldots = h_{nn}$)
  - **are correlated** (unless $\mathbf{H}$ is diagonal).

The residuals under the model assumptions

- While many graphical methods are based on the residuals, their unequal variances creates difficulties can make direct comparison and interpretation difficult.
- **Standardised residuals:**

$$\tilde{E}_i = \frac{E_i}{\hat{\sigma}\sqrt{1 - h_{ii}}}$$

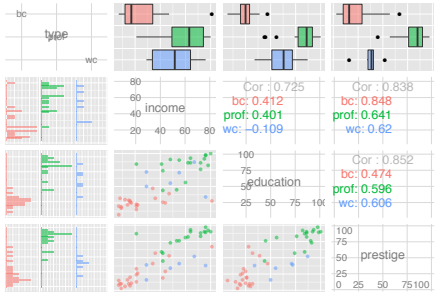
Under the model assumptions, the standardised residual has mean zero and its variance is approximately one, and hence its distribution is comparable with that of standard Normal variables.

(Note: there is another type of residuals that are called "studentized" and have the added benefit of having an exact t distributions.)

Response versus covariates

→ *Scatterplots of response against each covariate*. Include any important grouping factors in the data and check for nonlinear response-covariate relationships.

```
DuncanPairs
```



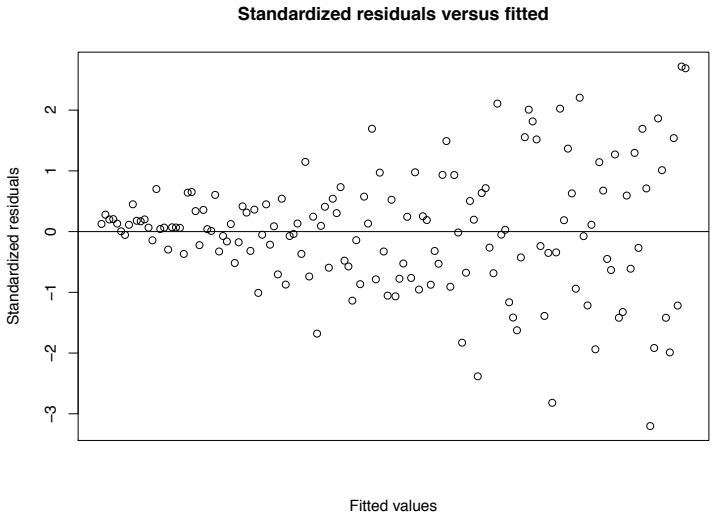
Residuals versus covariates and fitted values and Q-Q plots

- → *Plot (standardised) residuals against each covariate.*
- → *Plot (standardised) residuals against fitted values.*
  (Note: it does not make sense to plot residuals against the response because $\mathrm{Cov}(Y_i, E_i) = \mathrm{Var}(E_i)$ while $\mathrm{Cov}(\hat{Y}_i, E_i) = 0$)
- → *Normal Q-Q plot of the standardized residuals.*

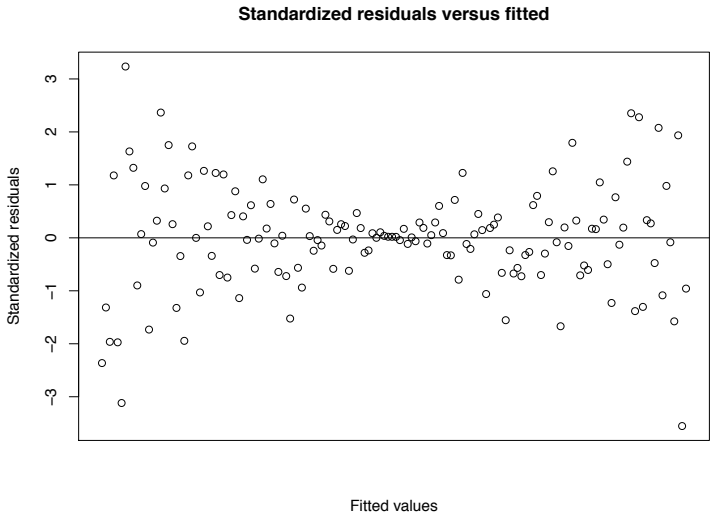
Under the model assumptions:

- **Linearity:** We expect to see the points to be randomly scattered about zero. Any systematic patterns provide evidence against linearity.
- **Homoscedasticity:** We expect to see the points to form a roughly horizontal band. Departures from this provide evidence against homoscedasticity.
- **Normalitty:** We expect to see the points on the Q-Q plot to roughly lie on a 45^o line.

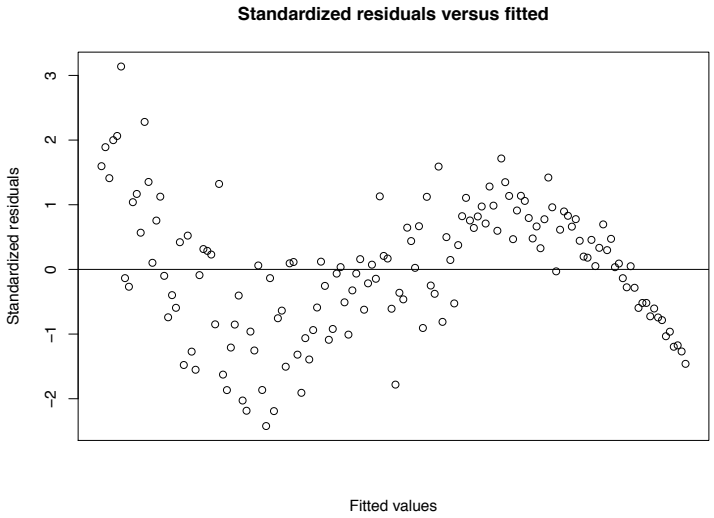
Residuals vs fitted values - Homoscedasticity not OK



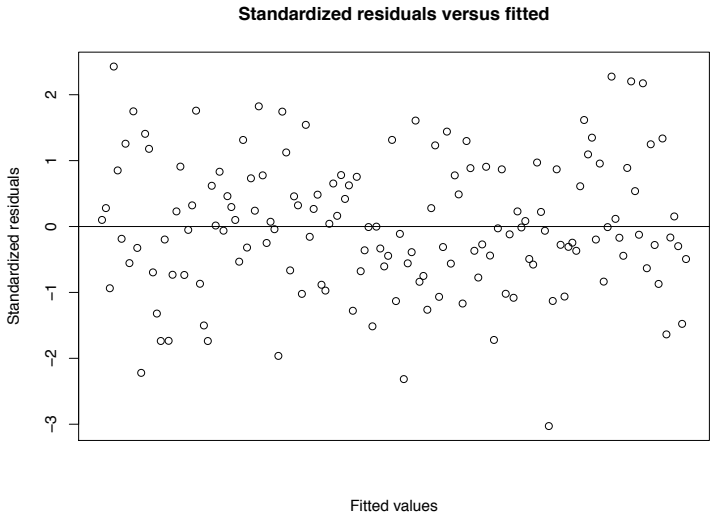
Residuals vs fitted values - Homoscedasticity not OK



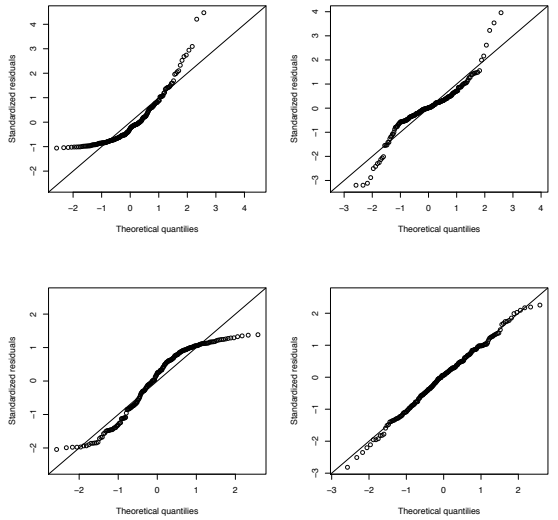
Residuals vs fitted values - Linearity not OK, Homoscedasticity not OK



Residuals vs fitted values - Linearity and homoscedasticity seem OK



Q-Q plots of residuals

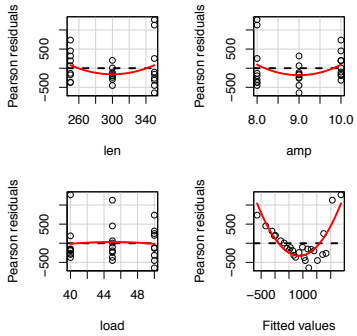


Residuals versus covariates and fitted values

```
str(Wool)

## 'data.frame': 27 obs. of  4 variables:
##  $ len   : int  250 250 250 250 250 250 250 250 250 300 ...
##  $ amp   : int  8 8 8 9 9 9 10 10 10 8 ...
##  $ load  : int  40 45 50 40 45 50 40 45 50 40 ...
##  $ cycles: int  674 370 292 338 266 210 170 118 90 1414 ...
```

Residuals versus covariates and fitted values

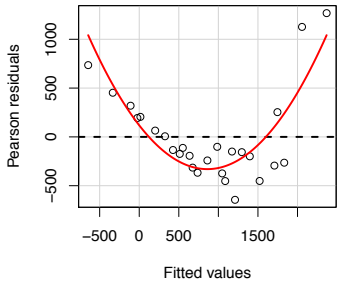
```
WoolLM <- lm(cycles ~ ., data = Wool)
residualPlots(WoolLM)
```




```
##           Test stat Pr(>|t|)
## len           1.209    0.239
## amp           1.413    0.172
## load         -0.237    0.815
## Tukey test    7.841    0.000
```

Residuals versus covariates and fitted values

```
residualPlots(WoolLM, ~1)
```

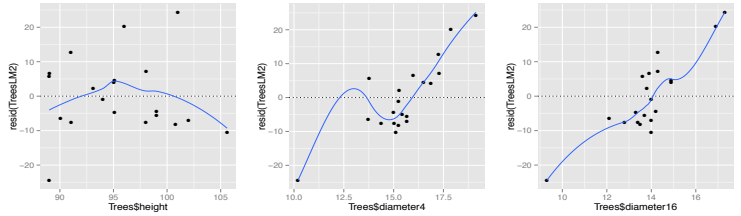



```
##            Test stat Pr(>|t|)
## Tukey test     7.841        0
```

Residual versus omitted covariate: A quick significance test

→ *Plot the residuals from a model that excludes a covariate vs that covariate*. If that covariate should not have been in the model then expect to see no pattern. Any systematic pattern hints towards including that covariate.

```
# Recall that TreesLM2 has no diameter covariates
zeroLine <- geom_abline(intercept = 0, slope = 0, lty = 3)
smoothLoess <- stat_smooth(method = "loess", se = FALSE)
q1 <- qplot(Trees$height, resid(TreesLM2)) + zeroLine + smoothLoess
q2 <- qplot(Trees$diameter4, resid(TreesLM2)) + zeroLine + smoothLoess
q3 <- qplot(Trees$diameter16, resid(TreesLM2)) + zeroLine + smoothLoess
grid.arrange(q1, q2, q3, ncol = 3)
```



Departures from linearity and homoscedasticity

- The reason of an apparent failure of the homoscedasticity assumption can be the failure of the linearity assumption.
- **Consequences of departures from linearity**: if linearity fails the model is inadequate, especially for prediction.

→ Possible remedies include **transformations of the response or covariates** and use of **nonlinear regression models**.

- **Consequences of departures from homoscedasticity**: least squares estimators are still unbiased, nevertheless the estimator $\hat{\sigma}^2$ can be severely biased and this affects the performance of standard hypothesis tests and confidence intervals.

→ Possible remedies include **transformations of the response or covariates**, use of **weighted least squares**, and use of **generalized linear models** that allow fitting non-Normal distributions with variance that depends on the mean.

Departures from Normality

- **Consequences of departures from Normality**: if the Normality assumption is severely violated then the performance of hypothesis tests and confidence intervals can be compromised. Though, these procedures are generally robust to small departures from Normality.

→ A possible remedy is to fit the model on a **transformed** version of the response which may have a distribution closer to Normal. Transformations like inverse, log and square root are amongst the popular ones.

The power family of transformations

- Assume that all $y_1, \ldots, y_n$ are positive (if not then just add a large number to all of them). We can consider that the linear model that applies on

Power family of transformation (a.k.a. Box-Cox transformation)

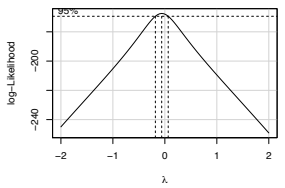
$$y_i^{(\lambda)} = \begin{cases} \frac{y_i^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \log y_i, & \lambda = 0 \end{cases}$$

- Different values of λ give rise to a wealth of transformations for the response: e.g. the **inverse** ($\lambda = -1$), the **log** ($\lambda = 0$), **square** ($\lambda = 2$), **square and cube roots** ($\lambda = 1/2$ and $\lambda = 1/3$), and the **original scale** ($\lambda = 1$).

→ *Profile likelihood/Wald tests and confidence intervals for λ.*

The power family of transformations

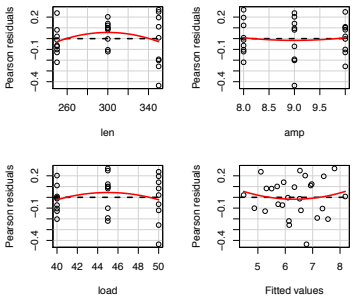
```
boxCox(WoolLM)
```




```
summary(powerTransform(WoolLM))

## bcPower Transformation to Normality
##
##    Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
## Y1   -0.0592   0.0611          -0.1789           0.0606
##
## Likelihood ratio tests about transformation parameters
##                             LRT df   pval
## LR test, lambda = (0)  0.9213  1 0.3371
## LR test, lambda = (1) 84.0757  1 0.0000
```

The power family of transformations

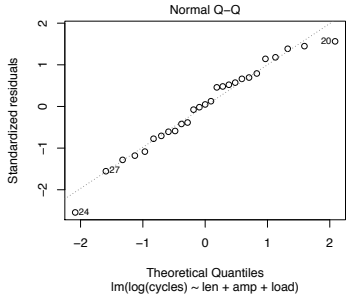
```
LogWoolLM <- update(WoolLM, log(cycles) ~ .)
residualPlots(LogWoolLM)
```




```
##            Test stat Pr(>|t|)
## len           -1.142    0.266
## amp            0.315    0.756
## load          -0.884    0.386
## Tukey test     0.562    0.574
```

The power family of transformations

```
# Using the plot.lm method of R to construct a Q-Q plot of
# statndardized residuals
plot(LogWoolLM, which = 2)
```




```
# This is the same as
qqnorm(rstandard(LogWoolLM), ylim = c(-2.5, 2))
qqline(rstandard(LogWoolLM), col = "red")
# We can also use qqPlot method of the car package
qqPlot(LogWoolLM)
# qqPLot compares studentized residuals against a t
```

Exercise

Use the `Ornstein` data and fit the linear model

```
OrnsteinLM <- lm(interlocks + 1 ~ assets + sector + nation, data = Ornstein)
```

- Examine the model assumptions.
- Use plots to find out transformations of one or more covariates that can potentially improve the model.
- Examine the model assumptions for the model with the transformed covariate(s) and diagnose any problems.
- Find a good λ for a power transformation of the response. Fit a new model to the transformed response and show any improvements (Hint: you can use the function `bcPower` that will transform the response at the best λ).

Independence

- Consequences of departures from independence: Departures from independence can have a serious effect on the performance of standard hypothesis tests and confidence intervals.
- Hard to check for all possible departures from independence. An arsenal of tests (with relative weaknesses and strengths) is provided in the `lmtest` **R** package.
- When there is a notion of "order" in the data, a common departure from independence is serial correlation of the errors, and this may show up in the residuals.

Serially correlated errors

If there are any time or other ordered variables in the data:

→ *Plot the standardized residuals against each of the ordered variables*. Expect to see no obvious patterns (clusters of points with monotone relationships, periodicites, etc.).

→ *Plot the ith vs the $(i-h)$th ordered residuals for lags $h = 1, 2, \ldots$.* Expect to see no trends (increasing or decreasing).

→ *Durbin-Watson test for serial correlation*:

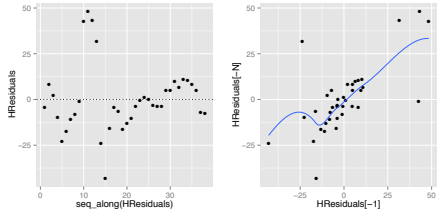
- Null hypothesis: H_0 : no serial correlation
- Possible alternatives: H_1 : serial correlation, H_1 : negative serial correlation or H_1 : positive serial correlation
- Test statistic: $D = \dfrac{\sum_{i=2}^{n}(E_i - E_{i-1})^2}{\sum_{i=1}^{n} E_i^2}$.
- p-values can be obtained using bootstrap. (Note: the calculation of the exact distribution of D under H_0 is possible but only practical for small n because it depends on $\mathbf{X}$).

Serially correlated errors

```
N <- nrow(Hartnagel)
# Fit the linear model in ?durbinWatsonTest
HartnagelLM <- lm(fconvict ~ tfr + partic + degrees + mconvict,
    data = Hartnagel)
# Extract the residuals
HResiduals <- residuals(HartnagelLM)
# Observations are ordered here by Hartnage£year so plot the
# residuals versus order
p1 <- qplot(y = HResiduals) + zeroLine
# Plot the ith versus the (i-1)th residual
p2 <- qplot(HResiduals[-1], HResiduals[-N]) + smoothLoess
```

Serially correlated errors

```
grid.arrange(p1, p2, ncol = 2)
```




```
durbinWatsonTest(HartnagelLM)

##  lag Autocorrelation D-W Statistic p-value
##    1          0.6883        0.6169       0
##  Alternative hypothesis: rho != 0
```


Influential observations

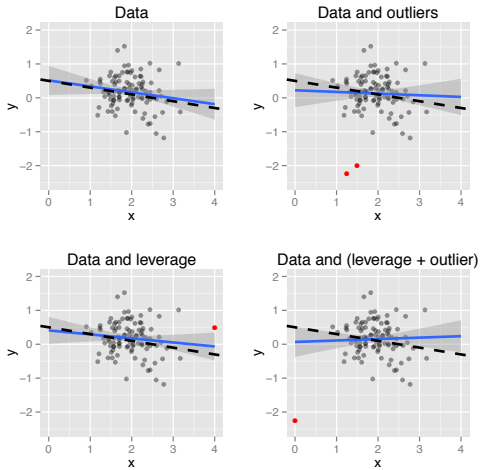
Influential observations are points that "stand out" from the main bulk of data and fall in either or both of the following categories:

- **Outliers**: observations that are located "far" in the y-direction from the cloud of points $\{(y_i, x_{i1}, \ldots, x_{ip})\}$.
- **Leverage points**: observations that have some extreme or unusual combination of covariate values (located "far" in the $\mathbf{x}$ "direction").

Usually outliers and/or leverage points arise because of:

- Isolated extreme cases in the system under study.
- Problems in data recording.
- Inadequacy of the model.

Influential observations



- Both outliers and leverage points can have a serious effect on the model fit.
- They tend to "pull" the regression hyperplane towards them.
- They deserve closer inspection. A possible strategy is to **detect them** and repeat the analysis without them, checking how much they affect the output.

Influential observations

- **Detecting outliers**: Outliers can be detected by visual inspection of residual plots. If the model assumptions are adequate we expect each standardized residual to lie roughly between -2 and 2 with high probability (if $Z \sim N(0,1)$, the $\mathrm{P}(|Z| > 2) \simeq 0.95$).

→ **Rule of thumb:** Examine observations with $|\bar{E}_i| > 2$.

- **Detecting leverage points:**. Note that

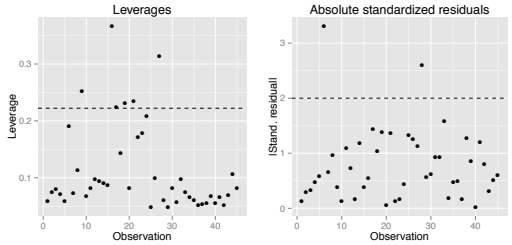
$$\mathrm{Var}(E_i) = \mathrm{Var}(Y_i - \hat{Y}_i) = \sigma^2(1 - h_{ii}) .$$

Now, h_{ii} depends only on $\mathbf{X}$ and $0 \leq h_{ii} \leq 1$. This means that if $h_{ii} \simeq 1$ then 1 df is used to perfectly fit a single observation! For this reason h_{ii} is called **leverage** (a.k.a. "hat value")
A good model matrix has $h_{ii} = p/n$ (because $\sum_{i=1}^{n} h_{ii} = p$).
(Also note that: $h_{ii} = \mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i$).

→ **Rule of thumb:** Examine observations with $h_{ii} > 2p/n$.

Influential observations

```
## Let's check influence manually here
hthres <- 2 * length(coef(DuncanLM))/nobs(DuncanLM)
r1 <- qplot(y = hatvalues(DuncanLM)) + geom_abline(intercept = hthres,
    slope = 0, lty = 2) + labs(x = "Observation", y = "Leverage",
    title = "Leverages")
rthres <- 2
r2 <- qplot(y = abs(rstandard(DuncanLM))) + geom_abline(intercept = rthres,
    slope = 0, lty = 2) + labs(x = "Observation", y = "|Stand. residual|",
    title = "Absolute standardized residuals")
grid.arrange(r1, r2, ncol = 2)
```



Influential observations: Cook's distance

- A popular measure of influence in regression is **Cook's distance**:

$$C_i = \frac{\bar{E}_i^2 h_{ii}}{p(1 - h_{ii})},$$

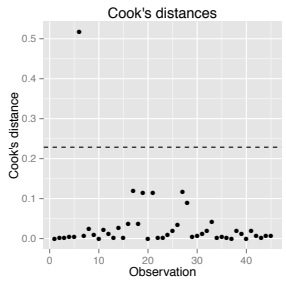
→ It results by measuring the distance of the fitted values from the fitted values calculated after removing the ith observation.

→ A large value of C_i results if an observation has a large leverage or a large standardised residual.

→ **Rule of thumb**: Examine observations with $C_i > 8/(n - 2p)$ (a combination of the two previous rules of thumb).

Influential observations

```
# Cook's distances for DuncanLM
cthres <- 8/(nobs(DuncanLM) - 2 * length(coef(DuncanLM)))
qplot(y = cooks.distance(DuncanLM)) + geom_abline(intercept = cthres,
    slope = 0, lty = 2) + labs(x = "Observation", y = "Cook's distance",
    title = "Cook's distances")
```

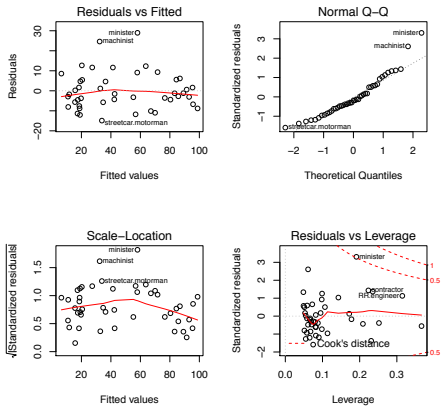


Model checking

- Scatterplots of the variables in the data: `pairs` (out of the box), `ggpairs` (`GGally` package), `scatterplotMatrix` (`car` package).
- **Linearity/Homoscedasticity:** Plot (standardized) residuals against each covariate and the fitted values: `plot.lm` (out of the box), `residualPlots` (`car` package).
- **Normality:** Normal Q-Q plot of the standardised residuals: `plot.lm` (out of the box), `qqPlot` (`car` package, studentized residuals), `qqnorm(rstandard(...))` (out of the box).
- If there are any problems with Linearity/Homoscedasticity/Normality attempt response/covariate transformations: for response transformations `powerTransform` (`car` package).
- **Independence:** If order information is available, plot the residuals versus order, plot ith vs $(i - h)$th ordered residual for some lag h, perform a Durbin-Watson test: `durbinWatson` (`car` package).
- **Influential observations:** `plot.lm` (out of the box), `influencePlot` (`car` package). For manual checks you can extract leverages, residuals and Cook's distances using generic methods like `hatvalues`, `residuals`, `rstandard`, `rstudent`, `cooks.distance`.

Model diagnostics

```
# plot.lm produces the following 4 plots for model checking
# by default
par(mfrow = c(2, 2))
plot(DuncanLM)
```



Collinearity

- **Perfect collinearity**: One or more covariates are linear combinations of the others.
  (Note: $\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y$ requires the inversion of $\mathbf{X}^T\mathbf{X}$. This inverse is not unique if there is a nonzero vector $\mathbf{c}$ such that $\mathbf{X}\mathbf{c} = (0, \ldots, 0)^T$)

→ Estimation is not possible (estimates are not unique).

- **Near collinearity**: One or more covariates are *almost* linear combinations of the others.

→ Estimation is possible and standard errors can be calculated but they may be misleading:

  - Estimates with signs that do not make sense.
  - Insignificant regression parameters regardless of the strength in the individual response-covariate associations ($T_j = \hat{\beta}_j / (\hat{\sigma}\sqrt{v_{jj}})$ where v_{jj} is the jth component of $\mathbf{V} = (\mathbf{X}^T\mathbf{X})^{-1}$).

Perfect collinearity

```
# Add an extra covariate in DuncanLM that is the linear
# combination of income and education. The addition of that
# covariate causes perfect collinearity. lm detects this,
# drops the covariate and proceeds with the rest.
update(DuncanLM, . ~ . + I(2 * income + 3 * education))

##
## Call:
## lm(formula = prestige ~ income + education + type + I(2 * income +
##     3 * education), data = Duncan)
##
## Coefficients:
##                   (Intercept)
##                        -0.185
##                        income
##                         0.598
##                     education
##                         0.345
##                     typeprof
##                        16.658
##                     typewc
##                       -14.661
## I(2 * income + 3 * education)
##                            NA
```

Diagnosing collinearity

- **Scatterplots of pairs of covariates**

→ Any rough linear relationships may lead to near collinearity (cannot reveal complex dependencies...).

- **Variance inflation factors**:

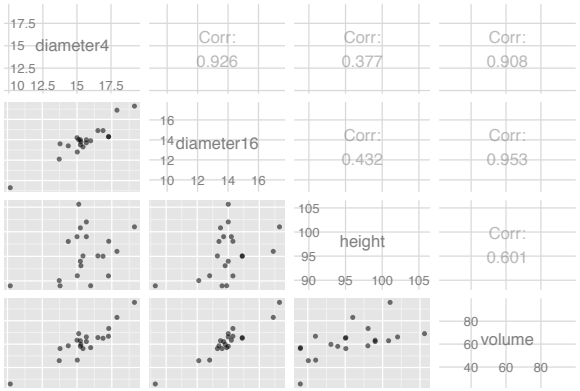
$$\mathrm{VIF}_j = \frac{1}{1 - R_j^2} \quad (j = 1, \ldots, p)\,,$$

where R_j^2 is the coefficient of determination for the regression of the jth covariate as a response and all the other covariates as explanatory variables. VIF_j have value 1 if all covariates are linearly independent and large values indicate collinearity.

→ **Rule of thumb:** Values greater than 5 are considered large.

Diagnosing collinearity

```
TreesPairs
```



Diagnosing collinearity

```
# The variance inflation factors for diameter4 and diameter6
# are larger than 7 which means that the estimated standard
# errors for diameter4 and diameter15 are inflated by more
# than sqrt(7) compared to the case of linear independence
vif(TreesLM1)

##  diameter4 diameter16     height
##      7.087      7.470      1.234
```

Diagnosing collinearity

```
# Let's remove diameter4 and recalculate VIF's
TreesLM3 <- lm(volume ~ diameter16 + height, data = Trees)
# Everything looks reasonable in terms of VIF's
vif(TreesLM3)

## diameter16     height
##      1.229      1.229
```

Diagnosing collinearity

```
summary(TreesLM3)

##
## Call:
## lm(formula = volume ~ diameter16 + height, data = Trees)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.231 -1.839 -0.401  1.092  6.937
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -105.903     14.652   -7.23  1.4e-06 ***
## diameter16     7.413      0.509   14.57  4.9e-11 ***
## height         0.677      0.170    3.98  0.00096 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.23 on 17 degrees of freedom
## Multiple R-squared:  0.953, Adjusted R-squared:  0.947
## F-statistic:  171 on 2 and 17 DF,  p-value: 5.52e-12
```

Diagnosing collinearity

```
TreesLM1Summary

##
## Call:
## lm(formula = volume ~ diameter4 + diameter16 + height, data = Trees)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.255 -1.677 -0.128  1.523  4.999
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -108.576     14.142   -7.68  9.4e-07 ***
## diameter4      1.626      1.026    1.58  0.13261
## diameter16     5.671      1.202    4.72  0.00023 ***
## height         0.694      0.163    4.25  0.00061 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.1 on 16 degrees of freedom
## Multiple R-squared:  0.959, Adjusted R-squared:  0.951
## F-statistic:  125 on 3 and 16 DF,  p-value: 2.59e-11
```


Remedies for collinearity

- Remove covariates (possibly in a "systematic way") and check for significant changes in the estimates or the significance of the other covariates.
- Apply ridge regression which proceeds by minimising

$$\sum_{i=1}^{n}(Y_i - \beta_1 - \sum_{j=2}^{p}\beta_p x_{ip}) + \lambda \sum_{j=2}^{p}\beta_j^2 \,,$$

with respect to $\boldsymbol{\beta} \in \Re^p$ for a fixed $\lambda > 0$ and where all covariates are centered to have zero mean. The value of λ is usually chosen to minimise some approximation of the predictive error. As λ grows, $\beta_2, \ldots \beta_p$ are shrunk towards zero.

Outline

Binary responses: Normal linear model

- Consider a binary response Y_i that takes values 0 and 1 and a covariate vector x_i. The Normal linear model for this data is

$$Y_i = \boldsymbol{\beta}^T \mathbf{x}_i + \epsilon_i \quad (i = 1, \ldots, n) \,,$$

with $\epsilon_1, \ldots, \epsilon_n$ being i.i.d with $\epsilon_1 \sim N(0, \sigma^2)$.
- Assuming that Y_i is 1 with probability π_i and 0 with probability $1 - \pi_i$, the model implies that

$$\pi_i = \boldsymbol{\beta}^T \mathbf{x}_i \quad (i = 1, \ldots, n) \,,$$

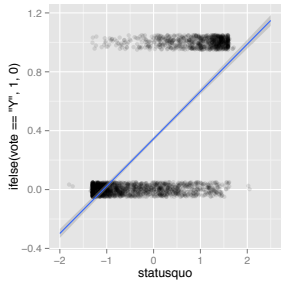
and that the error around π_i is normally distributed. This is called the **linear probability model**.

Binary responses: Normal linear model

- → **Modelling probabilities**: π_i can only take values between 0 and 1 while $\boldsymbol{\beta}\mathbf{x}_i$ can take any values over the real line. Hence, $\boldsymbol{\beta}$ ought to be restricted during fitting so that all fitted probabilities are between 0 and 1.
- → **Normality for a binary response**: The Normality assumption is inadmissible. Y_i can only take two values (0 and 1), while $\pi_i + \epsilon_i$ can take any value on the real line.
- → **Simple interpretation**: The probability changes by β_j for an 1-unit increase of the j-th covariate given that everything else is fixed.

Binary responses: Normal linear model

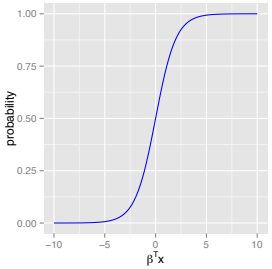
```
# A national survey conducted in April/May 1988 by
# FLASCO/Chile on the intention to vote for Pinochet. Status
# quo is a scale of support for the status quo.
ChilePlot <- ggplot(data = Chile, aes(x = statusquo, y = ifelse(vote ==
    "Y", 1, 0)))
AddJitter <- geom_jitter(position = position_jitter(height = 0.05),
    alpha = I(0.1))  # we add jitter to avoid overplotting
(ChilePlotLM <- ChilePlot + AddJitter + stat_smooth(method = "lm",
    fullrange = TRUE) + xlim(-2, 2.5))
```



Binary responses: beyond the Normal linear model

- Suppose that $Y_1, \ldots, Y_n$ are independent Bernoulli random variables with $\mathrm{E}(Y_i) = \mathrm{P}(Y_i = 1) = \pi_i$.
- Link π_i to the $\boldsymbol{\beta}^T\mathbf{x}_i$ using some monotone function that maps $(0, 1)$ on the real line.

$$\log \frac{\pi_i}{1 - \pi_i} = \boldsymbol{\beta}^T\mathbf{x}_i \quad (i = 1, \ldots, n)$$

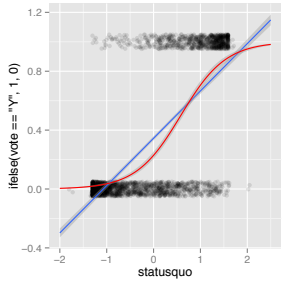


E.g.

- Estimate $\boldsymbol{\beta}$ using **maximum likelihood**.

Binary responses: beyond the Normal linear model

```
# Add the fitted logistic curve to ChilePlot
ChilePlotLM + stat_smooth(method = "glm", family = "binomial",
    col = "red", fullrange = TRUE)
```



Generalized linear model: Setting

Component	Normal linear model	Generalized linear model (GLM)
Random	$Y_1, \ldots, Y_n$ are independent and Y_i has a Normal distribution with mean μ_i and variance σ^2	$Y_1, \ldots, Y_n$ are independent and Y_i has a distribution from the exponential family with mean μ_i and dispersion ϕ
Structural	The linear predictor is $\eta_i = \boldsymbol{\beta}^T \mathbf{x}_i$	The linear predictor is $\eta_i = \boldsymbol{\beta}^T \mathbf{x}_i$
Link function	μ_i is linked to η_i as $\mu_i = \eta_i$	μ_i is linked to η_i via an invertible link function g as $g(\mu_i) = \eta_i$

E.g. Logistic regression: $Y_1, \ldots, Y_n$ are independent with $Y_i \sim \text{Binomial}(m_i, \pi(\mathbf{x}_i))$, where

$$\log \frac{\pi(\mathbf{x}_i)}{1 - \pi(\mathbf{x}_i)} = \boldsymbol{\beta}^T \mathbf{x}_i \quad (i = 1, \ldots, n)$$

Exponential family of distributions

- A random variable Y from the exponential family of distributions has density or mass function of the form

$$f(y\,;\theta) = \exp\left\{\frac{y\theta - b(\theta)}{\phi} + c(y;\phi)\right\}.$$

- It can be shown that

$$\mathrm{E}(Y) = \mu = b'(\theta)\,,$$
$$\mathrm{Var}(Y) = \phi b''(\theta) = b''(b'^{-1}(\mu)) = \phi u(\mu)\,.$$

- $u(\mu)$ is the variance function and ϕ is called the dispersion parameter and is often known.
- If ϕ is known then θ is called the natural parameter.

Exponential family of distributions

- Distributions like the Normal, binomial, Poisson and gamma are all special cases of the exponential family.
- The Normal linear model is just a special GLM ($g(\mu) = \eta$ and Normality for the random component).

→ GLMs extend the linear regression model and can handle responses that are counts (Poisson), proportions (binomial) and strictly positive (gamma).

Exponential family of distributions

	Support	θ	$b(\theta)$	μ	ϕ	$u(\mu)$
Normal(μ, σ^2)	$(-\infty, \infty)$	μ	$\frac{\theta^2}{2}$	θ	σ^2	1
Poisson(μ)	$\{0, 1, \ldots\}$	$\log \mu$	e^θ	e^θ	1	μ
Binomial$(m, \pi)/m$	$\frac{\{0, 1, \ldots, m\}}{m}$	$\log \frac{\pi}{1-\pi}$	$\log(1 + e^\theta)$	$\frac{e^\theta}{1+e^\theta}$	$\frac{1}{m}$	$\mu(1-\mu)$
Gamma(μ, ν)	$(0, \infty)$	$-\frac{1}{\mu}$	$-\log(-\theta)$	$-\frac{1}{\theta}$	$\frac{1}{\nu}$	μ^2

Some other well-known distributions that are of an exponential family form are the inverse Gaussian and the negative binomial

Link functions

- Standard link functions include

Link	$g(\mu)$	**R** knows it as
identity	μ	`identity`
log	$\log \mu$	`log`
logit	$\log \frac{\mu}{1-\mu}$	`logit`
probit	$\Phi^{-1}(\mu)$	`probit`
complementary log-log	$\log(-\log(1-\mu))$	`cloglog`
inverse	$1/\mu$	`inverse`
inverse-squared	$1/\mu^2$	`1/mu^2`
square root	$\sqrt{\mu}$	`sqrt`

where Φ^{-1} is the inverse of the cumulative distribution function of a $N(0,1)$ random variable.

`family` objects in **R**

- Link functions which make the natural parameter equal to the linear predictor ($\theta = \eta$) are called **canonical link functions**.
- Canonical link functions have nice theoretical properties (closed-form sufficient statistics) but have no particular advantage over other links in terms of model fit. The selection of link functions needs to be based on quality of fit considerations.
- The user can specify the **random component and link function combination** of a GLM via a `family` object. The canonical link is chosen by default.
- `family` objects contain the necessary distribution- and link-specific objects (like functions for the inverse link, derivatives of the link, and so on) that **R** uses for fitting GLMs.

`family` objects in **R**: Examples

```
# Binomial with canonical link (logit)
binomial()

##
## Family: binomial
## Link function: logit

# Now Binomial with cloglog link
binomial("cloglog")

##
## Family: binomial
## Link function: cloglog

# Normal with canonical link (identity)
gaussian()

##
## Family: gaussian
## Link function: identity
```

```
# Poisson with canonical link (log)
poisson()

##
## Family: poisson
## Link function: log

# Gamma with canonical link (inverse)
Gamma()

##
## Family: Gamma
## Link function: inverse

# The components of a family object
names(poisson("1/mu^2"))

##  [1] "family"     "link"       "linkfun"
##  [4] "linkinv"    "variance"   "dev.resids"
##  [7] "aic"        "mu.eta"     "initialize"
## [10] "validmu"    "valideta"   "simulate"
```

- Type `?family` for more details.

Maximum likelihood estimation

- The log-likelihood function is

$$l(\boldsymbol{\beta}, \phi; \mathbf{Y}) = \frac{1}{\phi} \sum_{i=1}^{n} \left[Y_i h(\boldsymbol{\beta}\mathbf{x}_i) - b(h(\boldsymbol{\beta}\mathbf{x}_i)) + \log c(Y_i; \phi) \right] ,$$

  where $h(\eta) = b'^{-1}(g^{-1}(\eta))$ (for canonical links $h(\eta) = \eta$).
- For general GLMs the maximization of $l(\boldsymbol{\beta}, \phi; \mathbf{Y})$ is done numerically and it can be shown that $\hat{\boldsymbol{\beta}}$ can be obtained independently of $\hat{\phi}$ (e.g. via Iteratively Weighted Least Squares).
- For GLMs with an unknown dispersion parameter (e.g. for Gamma responses), $\hat{\phi}$ can be obtained either by maximum likelihood or by using **deviance residuals** (**R** uses the latter).
- For Normal linear models the least squares estimator of $\boldsymbol{\beta}$ is also the maximum likelihood estimator.

The glm function for fitting GLMs

- The glm function is the interface of **R** for fitting GLMs. It has the following general form:

  `glm(formula, data, subset, weights, family, ...)`

  - `formula`, `data`, `subset` and `weights` are as for `lm`.
  - `family` is a valid family object (default is Normal with identity link).
  - `...` are other arguments that can be passed to `glm` (**type `?glm` for details**).

Maximum likelihood estimation

```
Chile1 <- na.omit(Chile)
ChileGLM <- glm(ifelse(vote == "Y", 1, 0) ~ statusquo + age +
    sex + education + income, data = Chile1, family = binomial("logit"))
coef(ChileGLM)

## (Intercept)   statusquo         age        sexM educationPS
##  -1.193e+00   2.088e+00   8.610e-04  -6.063e-02  -1.213e-01
##  educationS      income
##  -2.235e-01   3.067e-06
```


Exercise

Use `glm` to fit the Normal linear model with `prestige` as response and `income`, `education` and `type` as covariates (the `Duncan` data). Compare the results with those from `lm`.

Inference

Under the model assumptions, it can be shown that

- $\hat{\boldsymbol{\beta}} \overset{\text{appr}}{\sim} N(\boldsymbol{\beta}, \phi \mathbf{V}(\boldsymbol{\beta}))$, with $\mathbf{V}(\beta) = (\mathbf{X}^T \mathbf{W}(\boldsymbol{\beta}) \mathbf{X})^{-1}$ where $\mathbf{W}(\boldsymbol{\beta})$ is a diagonal matrix with ith diagonal entry the "quadratic weight" $w_i(\boldsymbol{\beta}) = g'(\mu_i)^{-2}/V(\mu_i)$.
- The z-statistic

$$Z_j = \frac{\hat{\beta}_j - \beta_j}{\sqrt{\phi v_{jj}(\hat{\beta})}},$$

  has approximately a $N(0, 1)$ distribution.
- If ϕ is estimated then **R** uses a t_{n-p} distribution to approximate the distribution of Z_j.

Individual hypotheses tests

```
# Individual hypotheses tests using the above large sample
# results. Interpretation is the same as for linear models.
summary(ChileGLM)


##
## Call:
## glm(formula = ifelse(vote == "Y", 1, 0) ~ statusquo + age + sex +
##     education + income, family = binomial("logit"), data = Chile1)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.285  -0.456  -0.227   0.562   2.895
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.19e+00   2.18e-01   -5.47  4.5e-08 ***
## statusquo    2.09e+00   8.03e-02   26.00  < 2e-16 ***
## age          8.61e-04   4.36e-03    0.20    0.843
## sexM        -6.06e-02   1.20e-01   -0.50    0.615
## educationPS -1.21e-01   2.15e-01   -0.56    0.572
## educationS  -2.23e-01   1.41e-01   -1.58    0.114
## income       3.07e-06   1.74e-06    1.76    0.079 .
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3129.1  on 2430  degrees of freedom
## Residual deviance: 1789.8  on 2424  degrees of freedom
```


Confidence intervals

```
# Profile likelihood confidence intervals are used by default
# for GLM objects
confint(ChileGLM)


##                  2.5 %     97.5 %
## (Intercept) -1.622e+00 -0.7674295
## statusquo    1.934e+00  2.2491887
## age         -7.688e-03  0.0094065
## sexM        -2.968e-01  0.1756838
## educationPS -5.434e-01  0.2990561
## educationS  -5.006e-01  0.0534844
## income      -3.393e-07  0.0000065


# For the familiar 'estimate +- quantile * se' confidence
# intervals use
confint.default(ChileGLM)


##                  2.5 %     97.5 %
## (Intercept) -1.620e+00 -7.653e-01
## statusquo    1.931e+00  2.245e+00
## age         -7.681e-03  9.403e-03
## sexM        -2.968e-01  1.755e-01
## educationPS -5.422e-01  2.997e-01
## educationS  -5.004e-01  5.338e-02
## income      -3.521e-07  6.485e-06
```


Model comparisons

- **Scaled deviance:** Let l_A be the log-likelihood for a model A with p parameters maximized over the regression coefficients and l_0 be the log-likelihood for the "saturated model" where μ_i are estimated as y_i (n parameters). If ϕ is not known then is should be replaced by an estimate of it. The scaled deviance is defined as

$$D_A = 2(l_A - l_0)\,,$$

- D_A is a generalization of the residual sum of squares and large values suggest poor fit.
- If D_B is the scaled deviance for a model B with q parameters, $p > q$, then

$$D_A - D_B \overset{\text{appr}}{\sim} \chi^2_{p-q}\,.$$

→ Can be used to produce Analysis of Deviance (ANODE) tables (same discussion to Sequential sums of squares).

Model comparisons

```
# R provides a wide range of tests
anova(ChileGLM, test = "LRT")


## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: ifelse(vote == "Y", 1, 0)
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                       2430       3129
## statusquo  1     1333      2429       1796   <2e-16 ***
## age        1        0      2428       1795    0.518
## sex        1        0      2427       1795    0.665
## education  2        2      2425       1793    0.354
## income     1        3      2424       1790    0.078 .
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Testing for relative importance

- `relimp` applies directly on `glm` objects for testing the contribution of a set of covariates relative to another.

Exercise

Use `ChileGLM` to test for the contribution of `statusquo` in explaining the voting intention relative to that of all the other covariates.

Residuals for GLMs

Residual	Formula	Notes
Response	$Y_i - \hat{\mu}_i$	
Pearson	$\frac{Y_i - \hat{\mu}_i}{\sqrt{u(\hat{\mu})}}$	
Standardized Pearson	$\frac{Y_i - \hat{\mu}_i}{\sqrt{u(\hat{\mu})(1 - h_{ii})}}$	
Deviance	$sign(Y_i - \hat{\mu}_i)\sqrt{d_i}$	$D_A = \sum_{i=1}^{n} d_i$
Standardized deviance	$\frac{sign(Y_i - \hat{\mu}_i)\sqrt{D_i}}{\sqrt{1 - h_{ii}}}$	
$\vdots$		

Residuals for GLMs

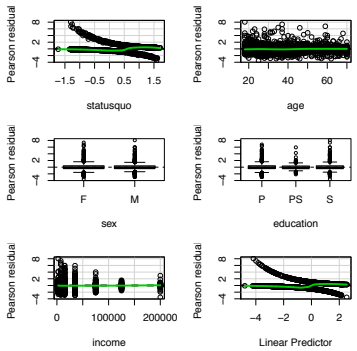
```
head(residuals(ChileGLM, type = "response"))

##        1        2        3        4        5        6
##  0.26598 -0.01742  0.18911 -0.03928 -0.02670 -0.03452

head(residuals(ChileGLM, type = "pearson"))

##       1       2       3       4       5       6
##  0.6020 -0.1331  0.4829 -0.2022 -0.1656 -0.1891

head(residuals(ChileGLM, type = "deviance"))

##       1       2       3       4       5       6
##  0.7864 -0.1875  0.6475 -0.2831 -0.2327 -0.2650
```

- Under the model assumptions, the standardized deviance residuals have distributions close to $N(0, 1)$.
- Note that for discrete data, the residuals are discrete.

Residuals for GLMs

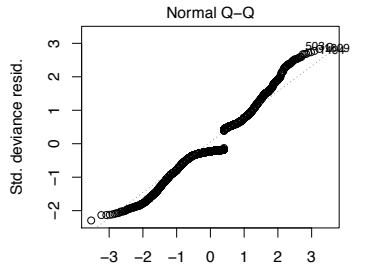
- Essentially all the **R** procedures we saw for linear models apply to the examination of residuals for GLMs.

```
# Residual plots for a GLM
residualPlots(ChileGLM)
```



Residuals for GLMs

```
# A Normal Q-Q plot of the standardized deviance residuals
plot(ChileGLM, which = 2)
```



Influence measures for GLMs

- The **leverage** for GLMs is the ith diagonal element of $H = \mathbf{X}(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}\mathbf{W}$. We estimate h_{ii} at $\hat{\boldsymbol{\beta}}$.

→ In the case of GLMs the leverage depends on $\mathbf{y}$.

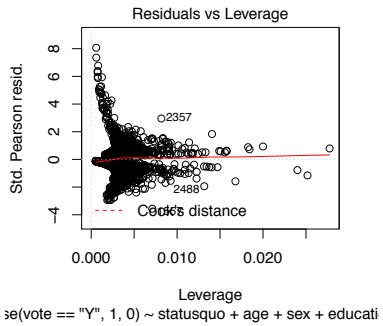
- **Cook's distance** can also approximated for GLMs

$$C_i = \frac{R_{P,i}\, h_{ii}}{p\hat{\phi}(1 - h_{ii})},$$

where $R_{P,i}$ is the ith standardized Pearson residual.

Influence measures for GLMs

```
plot(ChileGLM, which = 5)
```



Logistic regression

- **Responses**: $y_1, \ldots, y_n$ ("successes")
- **Totals**: $m_1, \ldots, m_n$ ("totals")
- **Covariates**: $(x_{11}, \ldots, x_{1p})^T, \ldots, (x_{n1}, \ldots, x_{np})^T$.
- **Model**: $Y_1, \ldots, Y_n$ are independent with

$$Y_i \sim \text{Binomial}(m_i, \pi_i),$$

and

$$\log \frac{\pi_i}{1 - \pi_i} = \sum_{j=1}^{p} \beta_j x_{ij}$$

Logistic regression: interpretation of parameters

- Ignoring the i subscript, suppose there are two covariate settings $\mathbf{x}$ and $\mathbf{z}$ where $\mathbf{z}$ is exactly equal to $\mathbf{x}$ apart from the jth covariate which is $z_j = x_j + 1$. Then,

$$\frac{\pi(\mathbf{z})}{1-\pi(\mathbf{z})} = e^{\beta_j}\frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}$$

→ Given that the other covariates remain fixed, β_j is the log of the **odds ratio** for the occurrence of the event at $x_j + 1$ to the occurrence of the event at x_j

Logistic regression in **R**

```
data(lizards)
(LizardsGLM <- glm(cbind(grahami, opalinus) ~ height + diameter +
    light + time, data = lizards, family = binomial("logit")))

##
## Call:  glm(formula = cbind(grahami, opalinus) ~ height + diameter +
##     light + time, family = binomial("logit"), data = lizards)
##
## Coefficients:
##  (Intercept)   height>=5ft  diameter>2in    lightshady
##        1.945         1.130        -0.763        -0.847
##   timemidday      timelate
##        0.227        -0.737
##
## Degrees of Freedom: 22 Total (i.e. Null);  17 Residual
## Null Deviance:     70.1
## Residual Deviance: 14.2  AIC: 83
```


Logistic regression in **R**

```
summary(LizardsGLM)

##
## Call:
## glm(formula = cbind(grahami, opalinus) ~ height + diameter +
##     light + time, family = binomial("logit"), data = lizards)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6601  -0.4195   0.0898   0.6712   1.4872
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.945      0.341    5.69  1.2e-08 ***
## height>=5ft     1.130      0.257    4.40  1.1e-05 ***
## diameter>2in   -0.763      0.211   -3.61  0.00031 ***
## lightshady     -0.847      0.322   -2.63  0.00858 **
## timemidday      0.227      0.250    0.91  0.36398
## timelate       -0.737      0.299   -2.46  0.01373 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 70.102  on 22  degrees of freedom
## Residual deviance: 14.205  on 17  degrees of freedom
## AIC: 83.03
##
## Number of Fisher Scoring iterations: 4
```


Logistic regression in **R**

```
lizardsT <- within(lizards, total <- grahami + opalinus)
(LizardsGLMTotals <- glm(grahami/total ~ height + diameter +
    light + time, weights = total, data = lizardsT, family = binomial("logit")))

##
## Call:  glm(formula = grahami/total ~ height + diameter + light + time,
##     family = binomial("logit"), data = lizardsT, weights = total)
##
## Coefficients:
##  (Intercept)   height>=5ft  diameter>2in    lightshady
##        1.945         1.130        -0.763        -0.847
##   timemidday      timelate
##        0.227        -0.737
##
## Degrees of Freedom: 22 Total (i.e. Null);  17 Residual
## Null Deviance:     70.1
## Residual Deviance: 14.2  AIC: 83
```

Exercise

- Interpret the coefficient for `lightshady` and `diameter>2inch` for the `LizardsGLM` fit.
- Check the model assumptions for the `LizardsGLM` fit.
- Can you fit an intercept only model to the grahami counts without using the `glm` function?
- Test the hypothesis that `LizardsGLM` is an as good description of grahami counts as the model with only an intercept is.
- Test the hypothesis that `time` is significant in explaining grahami counts.

Outline

Model selection

- With P covariates available we need to consider 2^P models (with linear predictors).

E.g. For $P = 10$ we need to consider 1024 models!

- We need:
  - A systematic way to search the "model space".
  - A criterion to compare models.

Information criteria

- **Information criteria** like Akaike's information criterion (AIC) and the Bayes' information criterion (BIC) attempt to approximate how far is a model from a "true model".

$$\text{AIC} = -2l + 2p$$
$$\text{BIC} = -2l + p \log n$$

where l is the maximized log-likelihood for a given model with p parameters.

- Best candidate model is take to be the one that minimizes AIC or BIC.

Mallows C_p

- The Mallows C_p is a model selection criterion for linear regression and has the form

$$C_p = \frac{\mathrm{RSS}_p}{s^2} + 2p - n\,.$$

where s^2 is usually taken to be $\hat{\sigma}^2$ from the model with all available covariates (say P), and RSS_p is the residual sum of squares from the model with $p < P$ covariates.

- Best candidate model is the one that has C_p closer to p.

Prediction error

- **Prediction error** is usually estimated by splitting the data into two sets:
  - Training set: $\mathbf{X}^{(t)}$ and $\mathbf{y}^{(t)}$ which is used to estimate the model.
  - Validation set: $\mathbf{X}^{(v)}$ and $\mathbf{y}^{(v)}$ which is used to estimate the prediction error.

  The prediction error can then be estimated as

$$\widehat{PE} = \sum_{i=1}^{n^{(v)}} \left[y_i^{(v)} - \mathbf{x}_i^{(v)} \hat{\boldsymbol{\beta}}^{(t)} \right]^2$$

- For small n, **cross-validation** can be used where the data is split into K parts ("folds"), and all but the kth part are used to estimate the model. Then, the kth part is used to calculate the prediction error, and the process is iterated over $k = 1, \ldots, K$. Then the cross-validated prediction error is the prediction error averaged over all folds.
  (for $k = n$ we get the well-known leave-one-out cross-validation)

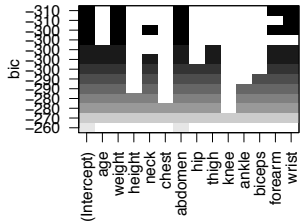
Best subsets regression

- A greedy approach for model selection is to fit all possible models with p variables and select the best one according to some model selection criterion.
- For linear models selection can be based on Mallows C_p (can be extended to other criteria) or R^2.
- The `regsubsets` function of the `leaps` package does this for a range of p values.

→ Best subsets can be prohibitively expensive for large p and for models other than linear.

Best subsets

```
data(bodyfat)
Bodyfat <- bodyfat[-match(c("case", "brozek", "density"), names(bodyfat))]
yBodyfat <- c(Bodyfat[, 1])
xBodyfat <- as.matrix(Bodyfat[, -1])
# nvmax is the maximum number of covariates to be considered
BodyfatBS <- regsubsets(x = xBodyfat, y = yBodyfat, nvmax = 13)
plot(BodyfatBS)
```



Stepwise methods

- The `step` function can perform **stepwise regression**, **forward selection**, and **backwards elimination** based on AIC (see also the `stepAIC` function in the `MASS` package).

→ There is much debate on the use of stepwise method because they lack theoretical justification and have been found to fit very complex models to completely random data.

```
BodyfatLM <- lm(siri ~ ., data = Bodyfat)
step(BodyfatLM)
```


Shrinkage penalties

- Idea: "relax" best subsets to get a "continuous" selection process by maximising a penalised version of the log-likelihood:

$$l(\beta, \phi; \mathbf{Y}) + \boldsymbol{\lambda}^T J(\boldsymbol{\beta}),$$

where $\boldsymbol{\lambda}$ is a vector of "tuning parameters" and $J(\beta)$ is a **shrinkage penalty**.
- There is a wide range of shrinkage penalties depending on the modelling strategies. Popular examples include the **LASSO** and the **elastic net**. If β_1 is the intercept
  - LASSO:

$$\boldsymbol{\lambda}^T J(\boldsymbol{\beta}) = \lambda_1 \sum_{j=2}^{p} |\beta_j|$$

  - Elastic net:

$$\boldsymbol{\lambda}^T J(\boldsymbol{\beta}) = \lambda_1 \sum_{j=2}^{p} |\beta_j| + \lambda_2 \sum_{j=2}^{p} \beta_j^2$$

Shrinkage penalties

- Selection of $\boldsymbol{\lambda}$ can be done by minimising a model selection criterion.
- The values of the regression parameters versus λ reveal the **solution path**.
- Extremely efficient procedures have been developed for the task (e.g. the entire LASSO solution path can be computed at the same computational cost as least squares!).

→ Both LASSO and elastic net can select a model in cases where $p > n$.

→ The LASSO has the ability to shrink the regression parameters to exactly zero.

→ The elastic net can do what the LASSO does but is more robust to collinearity and can reveal "grouping effects".

Shrinkage penalties

- The `caret` package is a "meta-package" that provides a unified interface to a wide variety of training methods.
- LASSO and elastic net are included in the supported methods.
- The main function of `caret` is `train`. Check `?train` or `http://caret.r-forge.r-project.org` for details.

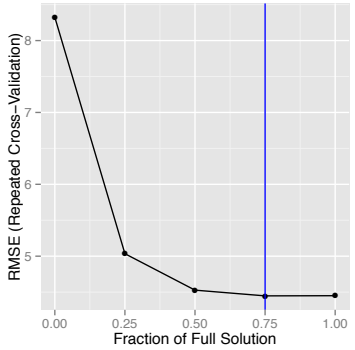
```
# Use the caret package that can do both LASSO and elastic
# net.  Chose 10-fold cross-validation with 10 repetitions
fitControl <- trainControl(method = "repeatedcv", repeats = 10,
    number = 10, verboseIter = FALSE)
# We let the fraction (J(beta)/J(hat{beta})) to vary from
# almost 0 to 1. This is equivalent to setting a grid for
# lambda1
tuneParsLASSO <- expand.grid(fraction = seq(1e-04, 1, length = 5))
# For elastic net we set lambda (this is lambda2 in previous
# slides) to vary from 0 to 10
tuneParsEnet <- expand.grid(fraction = seq(1e-04, 1, length = 5),
    lambda = seq(0, 10, length = 5))
```

Shrinkage penalties: LASSO example

```
# Apply LASSO to the bodyfat setting
BodyfatLASSO <- train(x = xBodyfat, y = yBodyfat, method = "lasso",
    trControl = fitControl, tuneGrid = tuneParsLASSO)
BodyfatLASSO


## The lasso
##
## 252 samples
##  13 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
##
## Summary of sample sizes: 227, 228, 227, 227, 228, 227, ...
##
## Resampling results across tuning parameters:
##
##   fraction  RMSE  Rsquared  RMSE SD  Rsquared SD
##   1e-04     8     0.7       0.8      0.09
##   0.3       5     0.7       0.5      0.09
##   0.5       5     0.7       0.6      0.07
##   0.8       4     0.7       0.5      0.07
##   1         4     0.7       0.5      0.07
##
## RMSE was used to select the optimal model using
##  the smallest value.
## The final value used for the model was fraction = 0.8.
```

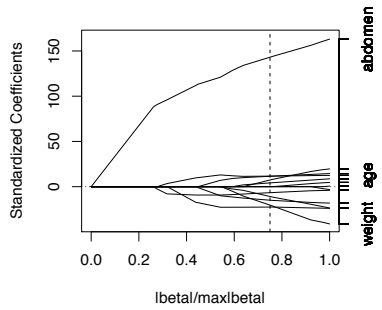

Shrinkage penalties: LASSO example

```
# Plot the root mean squared error as a function of the
# fraction
BestTuning <- BodyfatLASSO$bestTune$fraction
ggplot(BodyfatLASSO) + geom_vline(xintercept = BestTuning, col = "blue")
```



Shrinkage penalties: LASSO example

```
# Solution path
plot(BodyfatLASSO$finalModel)
abline(v = BestTuning, lty = 2)
```



Shrinkage penalties: LASSO example

```
# enet output (from elasticnet package)
BodyfatLASSO$finalModel


##
## Call:
## enet(x = as.matrix(x), y = y, lambda = 0)
## Cp statistics of the Lasso fit
## Cp: 698.40  93.62  85.47  65.41  30.12  30.51  19.39  20.91  18.68  17.41  1
## DF:  1  2  3  4  5  6  7  8  9 10 11 12 13 14
## Sequence of  moves:
##      abdomen height age wrist neck forearm hip weight
## Var        6      3   1    13    4      12   7      2
## Step       1      2   3     4    5       6   7      8
##      biceps thigh ankle chest knee
## Var      11     8    10     5    9 14
## Step      9    10    11    12   13 14
```

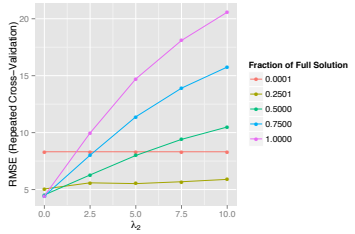
Shrinkage penalties: LASSO example

```
# coefficients at optimal fraction (in case you want to see
# those)
predict(BodyfatLASSO$finalModel, mode = "fraction", s = BestTuning,
    type = "coefficient")$coef
```

```
##      age   weight   height     neck    chest  abdomen
##  0.05842 -0.04384 -0.11436 -0.38836  0.00000  0.83779
##      hip    thigh     knee    ankle   biceps  forearm
## -0.09566  0.08865  0.00000  0.00000  0.08691  0.35930
##    wrist
## -1.52336
```


Shrinkage penalties: Elastic net example

```
# Apply Elastic net to the bodyfat setting: this may take
# some time
BodyfatEnet <- train(x = xBodyfat, y = yBodyfat, method = "enet",
    trControl = fitControl, tuneGrid = tuneParsEnet)
# Plot the root mean squared error as a function of the
# tuning parameters. Here we replace the awkward label for
# lambda2 with lambda2
ggplot(BodyfatEnet) + labs(x = expression(lambda[2]))
```




```
# Again the lasso fit is selected (lambda2 = zero)
```