Applied Statistical Regression
AS 2015 – Multiple Regression

Marcel Dettling

Institute for Data Analysis and Process Design

Zurich University of Applied Sciences

marcel.dettling@zhaw.ch

http://stat.ethz.ch/~dettling

ETH Zürich, October 12, 2015

What is Regression?

The answer to an everyday question:

How does a target variable of special interest depend on several other (explanatory) factors or causes.

Examples:

- growth of plants, depends on fertilizer, soil quality, …
- apartment rents, depends on size, location, furnishment, …
- car insurance premium, depends on age, sex, nationality, …

Regression:

- quantitatively describes relation between predictors and target
- high importance, most widely used statistical methodology

Applied Statistical Regression
AS 2015 – Multiple Regression

Example: Mortality Due to Air Pollution

Researchers at General Motors collected data on 60 US Standard Metropolitan Statistical Areas (SMSAs) in a study of whether air pollution contributes to mortality.

City	Mortality	JanTemp	JulyTemp	RelHum	Rain	Educ	Dens	NonWhite	WhiteCllr	Pop	House	Income	HC	NOx	SO2
Akron, OH	921.87	27	71	59	36	11.4	3243	8.8	42.6	660328	3.34	29560	21	15	59
Albany, NY	997.87	23	72	57	35	11.0	4281	3.5	50.7	835880	3.14	31458	8	10	39
Allentown, PA	962.35	29	74	54	44	9.8	4260	0.8	39.4	635481	3.21	31856	6	6	33
Atlanta, GA	982.29	45	79	56	47	11.1	3125	27.1	50.2	2138231	3.41	32452	18	8	24
Baltimore, MD	1071.29	35	77	55	43	9.6	6441	24.4	43.7	2199531	3.44	32368	43	38	206
Birmingham, AL	1030.38	45	80	54	53	10.2	3325	38.5	43.1	883946	3.45	27835	30	32	72
Boston, MA	934.70	30	74	56	43	12.1	4679	3.5	49.2	2805911	3.23	36644	21	32	62
Bridgeport, CT	899.53	30	73	56	45	10.6	2140	5.3	40.4	438557	3.29	47258	6	4	4
Buffalo, NY	1001.90	24	70	61	36	10.5	6582	8.1	42.5	1015472	3.31	31248	18	12	37
Canton, OH	912.35	27	72	59	36	10.7	4213	6.7	41.0	404421	3.36	29089	12	7	20
Chattanooga, TN	1017.61	42	79	56	52	9.6	2302	22.2	41.3	426540	3.39	25782	18	8	27
Chicago, IL	1024.89	26	76	58	33	10.9	6122	16.3	44.9	606387	3.20	36593	88	63	278
Cincinnati, OH	970.47	34	77	57	40	10.2	4101	13.0	45.7	1401491	3.21	31427	26	26	146
Cleveland, OH	985.95	28	71	60	35	11.1	3042	14.7	44.6	1898825	3.29	35720	31	21	64
Columbus, OH	958.84	31	75	58	37	11.9	4259	13.1	49.6	124833	3.26	29761	23	9	15
Dallas, TX	860.10	46	85	54	35	11.8	1441	14.8	51.2	1957378	3.22	38769	1	1	1

→ see http://lib.stat.cmu.edu/DASL/Stories/AirPollutionandMortality.html

Multiple Linear Regression

We use linear modeling for a multiple predictor regression:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + E_i$$

- there are now p predictors
- the problem cannot be visualized in a scatterplot
- there will be n observations of response and predictors
- goal: estimating the coefficients $\beta_0, \beta_1, ..., \beta_p$ from the data

IMPORTANT: simple linear regression of the response on each of the predictors does not equal multiple regression, where *all predictors are used simultanously*.

Data Preparation: Visualization

Because we cannot inspect the data in a xy-scatterplot, data visualization and data preparation becomes an important task. We need to identify the necessary variable transformations, mitigate the effect of outliers, …

Step 1: Plotting the marginal distribution (i.e. histograms)

```
> par(mfrow=c(4,4))
> for (i in 1:15) hist(apm[,i], main= "...")
```

Step 2: Identify erroneous and missing values

```
> any(is.na(apm))
[1] FALSE
```

Applied Statistical Regression
AS 2015 – Multiple Regression

Data Preparation: Transformations

Linear regression and its output are easier to comprehend if one is using an intuitive scale for the variables. Please note that linear transformations do not change the results. However, any non-linear transformation will do so.

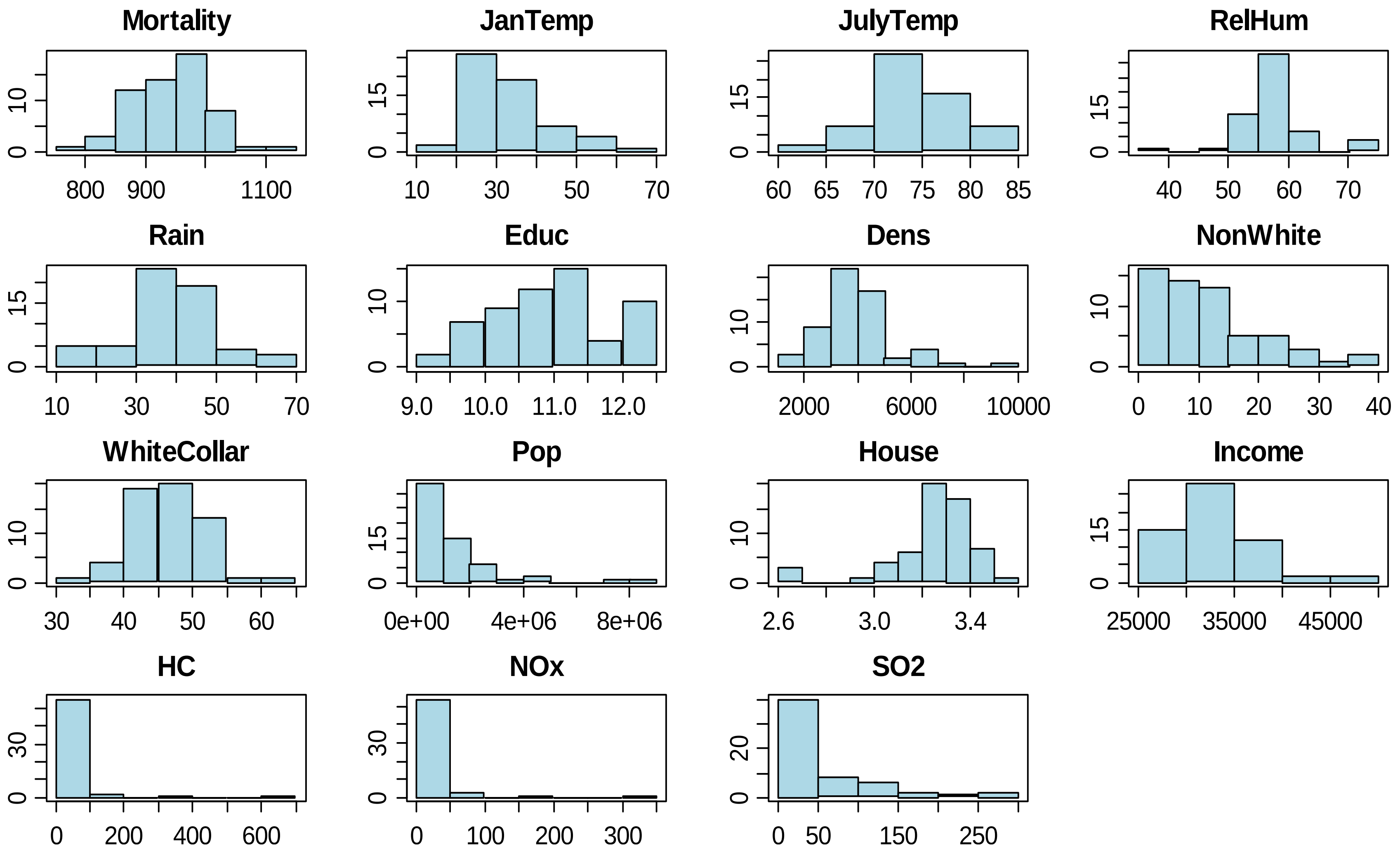
Step 3: linear transformations $x' = ax + b$

```
> apm$JanTemp  <- (5/9)*(apm$JanTemp-32)
> apm$JulyTemp <- (5/9)*(apm$JulyTemp-32)
> apm$Rain     <- (2.54)*apm$Rain
```

Step 4: log-transformation $x' = \log(x)$

For all variables where it is necessary/beneficial...

Data Preparation: Transformations



Why Simple Regression Is Not Enough

Performing many simple lineare regressions of the response on any of the predictors is not the same as multiple regression!

Observation	x1	x2	yy
1	0	-1	1
2	1	0	2
3	2	1	3
4	3	2	4
5	0	1	-1
6	1	2	0
7	2	3	1
8	3	4	2

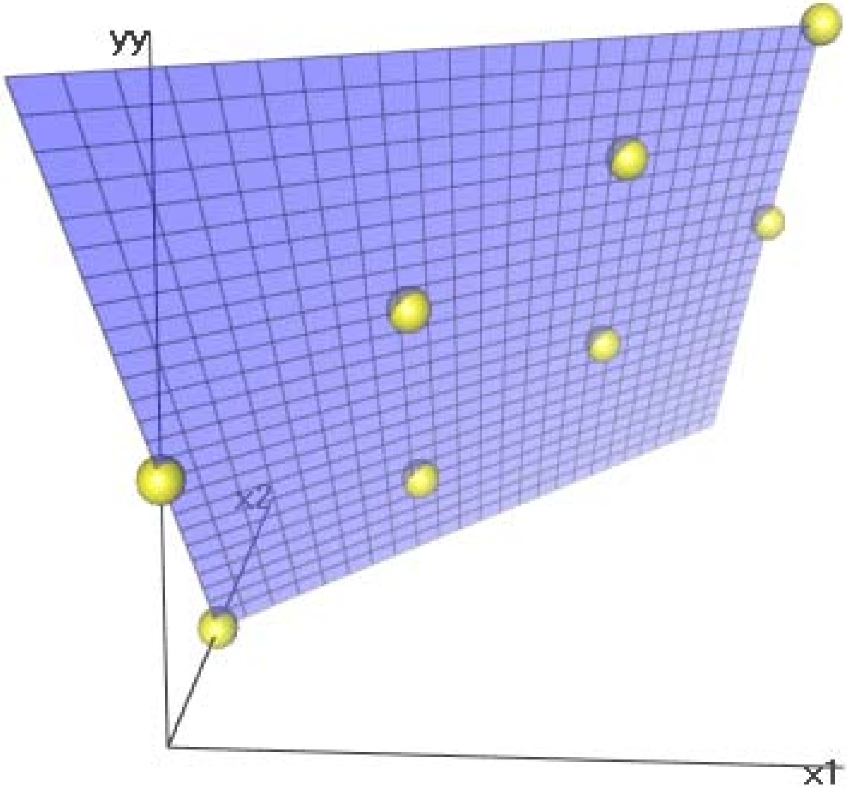
We have $y_i = \hat{y}_i = 2x_{i1} - x_{i2}$, i.e. a perfect fit.
Hence, all residuals are zero and we estimate $\hat{\sigma}_E^2 = 0$.

→ *The result can be visualized with a 3d-plot!*

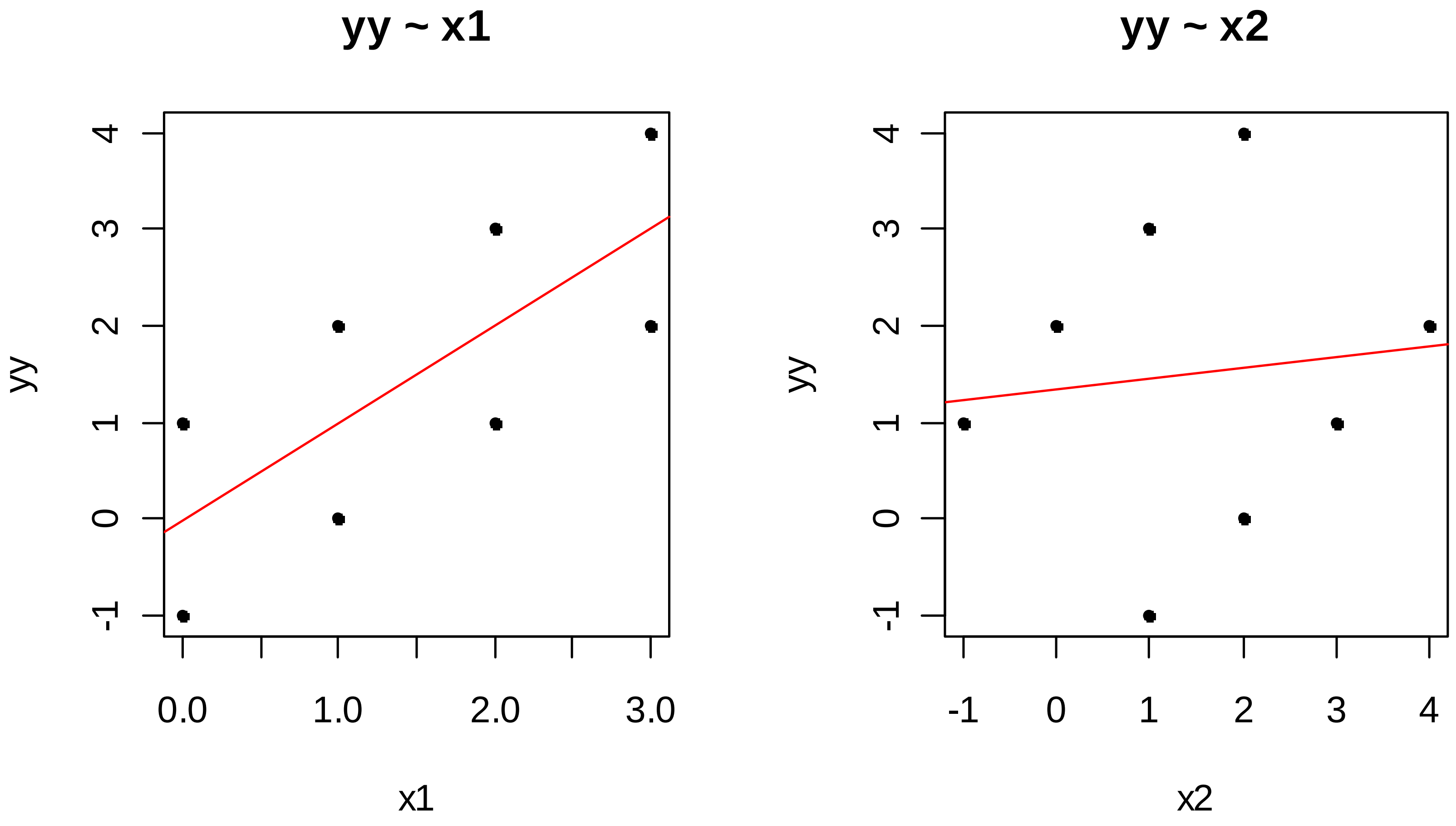
Applied Statistical Regression
AS 2015 – Multiple Regression

Why Simple Regression Is Not Enough

```
> library(Rcmdr)
> scatter3d(yy ~ x1 + x2, axis.scales=FALSE)
```



Why Simple Regression Is Not Enough



The Multiple Linear Regression Model

In colloquial notation, the model is:

$$Mortality_i = \beta_0 + \beta_1 \cdot JanTemp_i + \beta_2 \cdot JulyTemp_i + ... + \beta_{14} \cdot \log(SO_2)_i + E_i$$

More generally, the multiple linear regression model specifies the relation between response y and predictors $x_1,...,x_p$. There are observations $i = 1,...,n$. We use the double index notation:

$$y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} + E_i \quad \text{for} \quad i = 1,...,n$$

Here, β_0 is the intercept and $\beta_1,...,\beta_p$ are regression coefficients.

The regression coefficient β_j is the increase in the response, if the predictor x_j increases by 1 unit, but all other predictors remain unchanged.

Matrix Notation

In matrix notation, the multiple linear regression model can be written as:

$$y = X\beta + E$$

The elements in this equation are as follows:

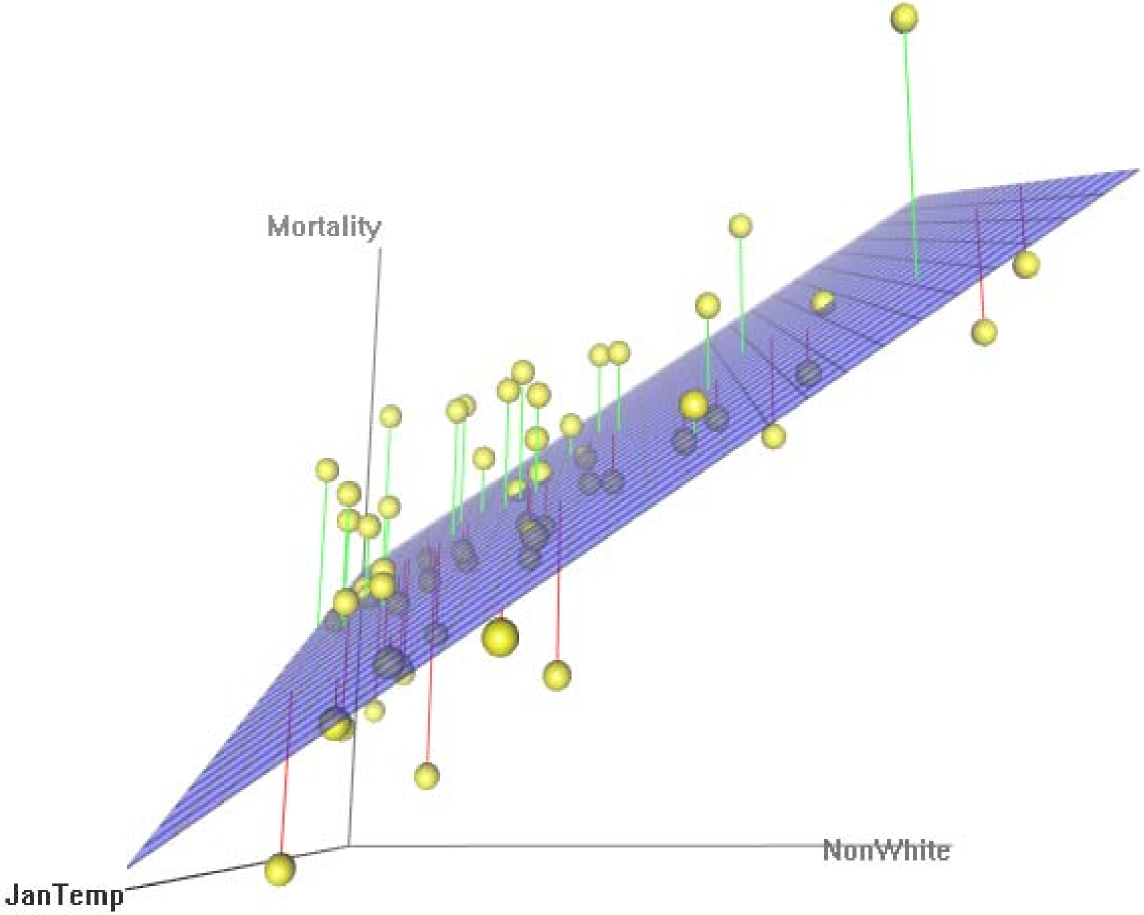
→ **see blackboard…**

Applied Statistical Regression

AS 2015 – Multiple Regression

Fitting Multiple Regression Models

Toy example: $Mortality_i = \beta_0 + \beta_1 \cdot JanTemp_i + \beta_2 \cdot NonWhite_i + E_i$



Least Squares Algorithm

The *paradigm* is to determine the regression coefficients such that the *sum of squared residuals is minimal*. This amounts to minimizing the quality function:

$$Q(\beta_0, \beta_1, ..., \beta_p) = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}))^2$$

We can take partial derivatives with respect to $\beta_0, \beta_1, ..., \beta_p$ and so obtain a linear equation system with $(p+1)$ unknowns and the same number of equations.

→ **Mostly (but not always...), there is a unique solution.**

Normal Equations and Their Solutions

The least squares approach leads to the normal equations, which are of the following form:

$(X^T X)\beta = X^T y$ resp. $\hat{\beta} = (X^T X)^{-1} X^T y$

- Unique solution if and only if X has full rank
- Predictor variables need to be linearly independent

- If X has not full rank, the model is "badly formulated"
- Design improvement mandatory!!!

- Necessary (not sufficient) condition: $p < n$
- Do not over-parametrize your regression!

Multiple Regression in R

In R, multiple linear least squares regression is carried out with command `lm()`. The syntax is as follows:

```
fit <- lm(Mortality ~ JanTemp + JulyTemp + RelHum +
                      Rain + Educ + Dens + NonWhite +
                      WhiteCollar + log(Pop) + House +
                      Income + log(HC) + log(NOx) +
                      log(SO2), data=apm)
```

An often useful short notation is:

```
fit <- lm(Mortality ~ ., data=apm)
```

Except for the response, all variables in `apm` are predictors.

Estimating the Error Variance

For producing confidence intervals for the coefficients, testing the regression coefficients and producing a prediction interval for future observation, having an estimate of the error variance is indispensable.

$$\hat{\sigma}_E^2 = \frac{1}{n-(p+1)} \sum_{i=1}^{n} r_i^2$$

The estimate is given by the "average residual". The division by $n-(p+1)$ is for obtaining an unbiased estimator. Here, p is the number of predictors, and $(p+1)$ is the number of coefficients which are estimated!!!

Assumptions on the Error Term

The assumptions are identical to simple linear regression.

- $E[E_i] = 0$, i.e. the hyper plane is the correct fit
- $Var(E_i) = \sigma_E^2$, constant scatter for the error term
- $Cov(E_i, E_j) = 0$, uncorrelated errors
- $E_i \sim N(0, \sigma_E^2)$, the errors are normally distributed

Note: As in simple linear regression, we do not require Gaussian distribution for OLS estimation and certain optimality results, i.e. the Gauss-Markov theorem.

But: All tests and confidence intervals rely on the Gaussian, and there are better estimates for non-normal data

Properties of the Estimates

Gauss-Markov Theorem:

The OLS regression coefficients are unbiased, and they have minimal variance among all estimators that are linear and unbiased (=*BLUE*, *Best Linear Unbiased Estimates*).

Distribution of the Estimates:

If additionally, the errors are iid and follow a normal distribution, the estimated regression coefficients and the fitted values will also be normally distributed. In this case, the covariance matrix is explicitly known, which allows for construction of tests and confidence intervals.

Hat Matrix

The matrix notation and some mathematics allow for deeper insight into the function of the OLS estimator. We study:

$\hat{y} = X\hat{\beta}$ which are the fitted values in matrix notiation

Now we replace the coefficient vector with the OLS solution:

$$\hat{y} = X\hat{\beta} = X(X^TX)^{-1}X^Ty = Hy$$

Hence for obtaining the fitted values, we apply the hat matrix $H = X(X^TX)^{-1}X^T$ to the response vector y. This clarifies that the OLS estimator is linear. Moreover, H is the orthogonal projection of y on the space spanned by the columns of X.

Benefits of Linear Regression

- **Inference on the relation between y and $x_1,\ldots,x_p$**

The goal is to understand if and how strongly the response variable depends on the predictor. There are performance indicators as well as statistical tests adressing the issue.

- **Prediction of (future) observations**

The regression equation can be employed to predict
the response value for any given predictor configuration.

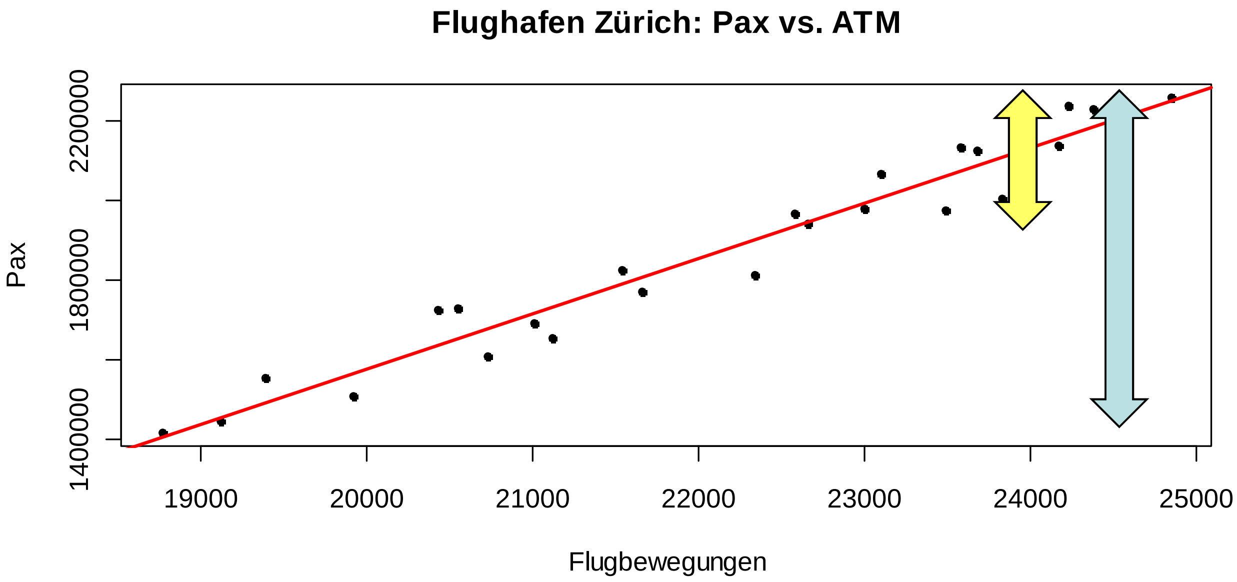
$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_p x_p$$

However, this mostly will not work well for extrapolation!

R^2: *The Coefficient of Determination*

The coefficient of determination R^2 tells which portion of the total variation is accounted for by the regression hyperplane.

→ For multiple linear regression, visualization is impossible!
→ The number of predictor used should be taken into account.



Coefficient of Determination

The coefficient of determination, also called *multiple R-squared*, is aimed at describing the goodness-of-fit of the multiple linear regression model:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2} \in [0,1]$$

It shows the proportion of the total variance which has been explained by the predictors. The extreme cases 0 and 1 mean:…

Adjusted Coefficient of Determination

If we add more and more predictor variables to the model, R-squared will always increase, and never decreases

Is that a realistic goodness-of-fit measure?

→ **NO, we better adjust for the number of predictors!**

The adjusted coefficient of determination is defined as:

$$adjR^2 = 1 - \frac{n-1}{n-(p+1)} \cdot (1-R^2) \in [0,1]$$

Hence, the adjusted R-squared is always (but in many cases irrelevantly) smaller than the plain R-squared. The biggest discrepancy is with small n, large p and small R^2.

Confidence Interval for Coefficient β_j

We can give a 95%-CI for the regression coefficient β_j . It tells which values, besides the point estimate $\hat{\beta}_j$, are plausible too.

Note: This uncertainty comes from sampling effects

95%-VI for $\beta_j : \hat{\beta}_j \pm qt_{0.975;n-(p+1)} \cdot \hat{\sigma}_{\hat{\beta}_j}$

In R:

```
> fit <- lm(Mortality ~ ..., data=apm)

> confint(fit, "Educ") ## or confint(fit)
    2.5 %   97.5 %
Educ -31.03177 4.261925
```

Testing the Coefficient β_j

There is a statistical hypothesis test which can be used to check whether $\hat{\beta}_j$ is significantly different from zero, or different from any other arbitrary value b. The null hypothesis is:

$$H_0 : \beta_j = 0, \text{ resp. } H_0 : \beta_j = b$$

One usually tests two-sided on the 95%-level. The alternative is:

$$H_A : \beta_j \neq 0, \text{ resp. } H_A : \beta_j \neq b$$

As a test statistic, we use:

$$T = \frac{\hat{\beta}_j}{\hat{\sigma}_{\hat{\beta}_j}}, \text{ resp. } T = \frac{\hat{\beta}_j - b}{\hat{\sigma}_{\hat{\beta}_j}}, \text{ both follow a } t_{n-(p+1)} \text{ distribution.}$$

Reading R-Output

```
> summary(fit)

Coefficients:   Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.297e+03  2.934e+02   4.422 6.32e-05 ***
JanTemp       -2.368e+00  8.851e-01  -2.676   0.0104 *
JulyTemp      -1.752e+00  2.031e+00  -0.863   0.3931
RelHum         3.852e-01  1.071e+00   0.360   0.7208
[output partially ommitted...]
log(NOx)       3.610e+01  1.415e+01   2.551   0.0142 *
log(SO2)      -3.687e+00  7.359e+00  -0.501   0.6189
---
Residual standard error: 34.48 on 44 degrees of freedom
Multiple R-squared: 0.7685, Adjusted R-squared: 0.6949
F-statistic: 10.43 on 14 and 44 DF, p-value: 8.793e-10
```

Note: due to space constraints, this is only a part of the output!

Individual Parameter Tests

These tests quantify the effect of the predictor x_j on the response y after having subtracted the linear effect of all other predictor variables on y.

Be careful, because of:

a) The *multiple testing problem*: when doing many tests, the total type I error increases. By how much?
→ **See blackboard...**

b) It can happen that all individual tests do not reject the null hypothesis, although some predictors have a significant effect on the response. **Reason**: *correlated predictors!*

Individual Parameter Tests

These tests quantify the effect of the predictor x_j on the response y after having subtracted the linear effect of all other predictor variables on y.

Be careful, because of:

c) The p-values of the individual hypothesis tests are based on the assumption that the other predictors remain in the model and do not change. Therefore, you must not drop more than one single non-significant predictor at a time!

Solution: *drop one*, *re-evaluate the model*, *drop one*, ...

Simple Variable Selection

Goal: Dropping all predictors from the regression model which are not necessary, i.e. do not show a significant impact on the response.

How: In a step-by-step manner, the least significant predictor is dropped from the model, as long as its p-value still exceeds the value of 0.05.

In R:

```
> fit <- update(fit, . ~ . - RelHum)
> summary(fit)
```

→ **Exercise: try do to this for the Mortality Data**

Comparing Hierachical Models

Idea: Correctly comparing two multiple linear regression models when the smaller has >1 predictor less than the bigger.

Where and why do we need this?

- for the 3 pollution variables in the mortality data.
- soon also for the so-called factor/dummy variables.

Idea: We compare the residual sum of squares (RSS):

Big model: $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_q x_q + \beta_{q+1} x_{q+1} + \ldots + \beta_p x_p$

Small model: $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_q x_q$

The big model must contain all the predictors from the small model, else they are not hierarchical and the test does not apply.

Comparing Hierarchical Models

Null hypothesis:

$H_0 : \beta_{q+1} = \beta_{q+2} = ... = \beta_p = 0$, versus the alternative hypothesis that at least one $\beta_j \neq 0,\ j = q+1,...p$

The test compares the RSS of the big and the small model:

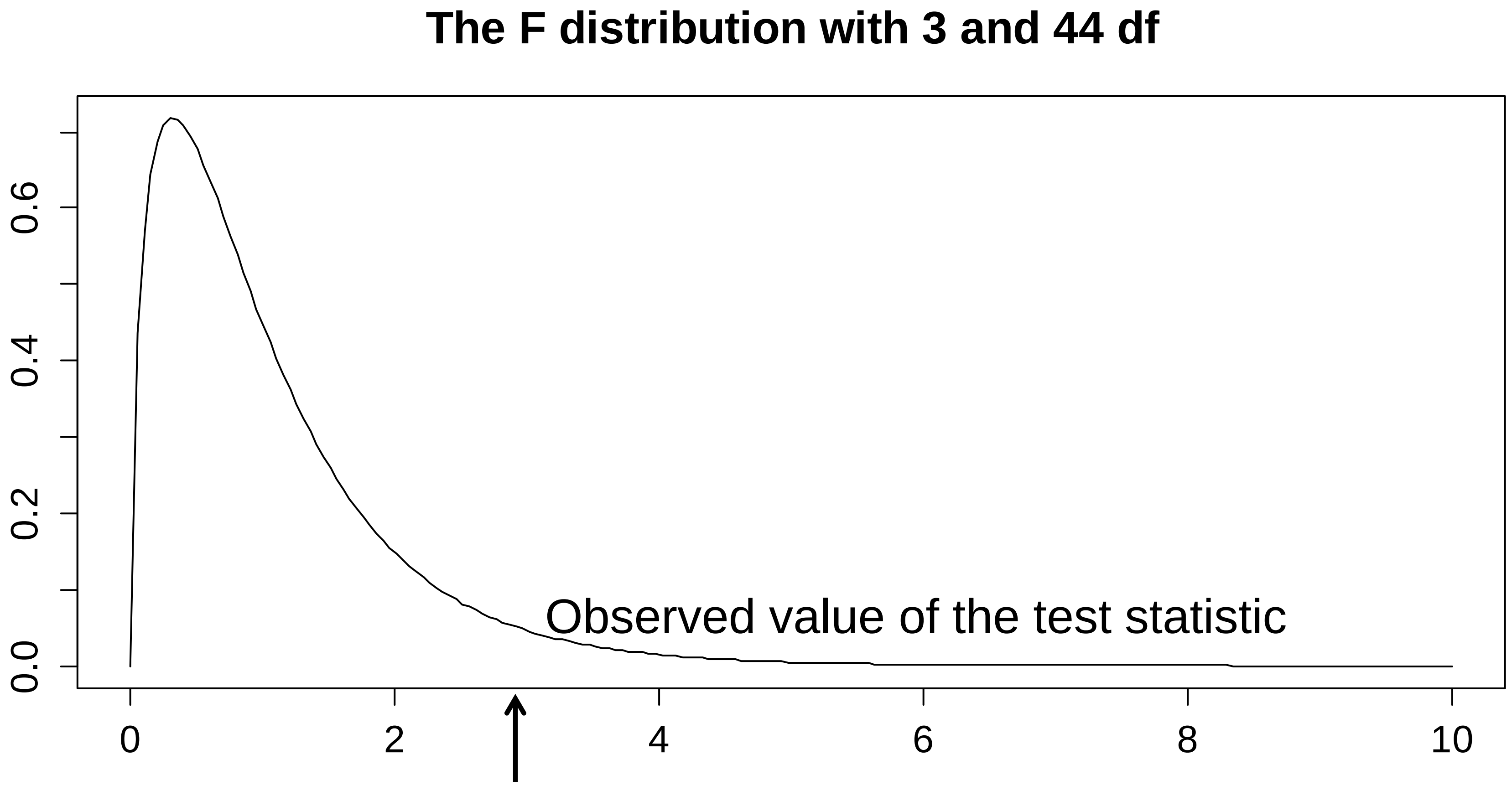
$$F = \frac{n-(p+1)}{p-q} \cdot \frac{RSS_{Small} - RSS_{Big}}{RSS_{Big}} \sim F_{p-q,n-(p+1)}$$

→ If the F-value is small ($p \geq 0.05$), the two models perform equally. There is no evidence against the null and we can continue working with the small model.

Comparing Hierachical Models in R

```
> f.big   <- lm(Mortality ~ ..., data=apm)
> f.small <- update(f.big,.~.-log(HC)-log(NOx)-log(SO2))

> anova(f.small, f.big)

Analysis of Variance Table

Model 1: Mortality ~ JanTemp + JulyTemp + RelHum + Rain +
         Educ + Dens + NonWhite + WhiteCollar + log(Pop) +
         House + Income

Model 2: Mortality ~ JanTemp + JulyTemp + RelHum + Rain +
         Educ + Dens + NonWhite + WhiteCollar + log(Pop) +
         House + Income + log(HC) + log(NOx) + log(SO2)

  Res.Df   RSS Df Sum of Sq      F  Pr(>F)
1     47 61142
2     44 52312  3    8829.3 2.4755 0.07388 .
```

Density Function of the F-distribution



The Global F-Test

***Idea**: is there any relation between response and predictors?*

This is another hierachical model comparison. The full model is tested against a small model with only the intercept, but without any predictors.

We are testing the null $H_0 : \beta_1 = \beta_2 = ... = \beta_p = 0$ against the alternative $H_A : \beta_j \neq 0$ for at least one predictor x_j. This test is again based on comparing the RSS:

$$F = \frac{n-(p+1)}{p} \cdot \frac{RSS_{Small} - RSS_{Big}}{RSS_{Big}} \sim F_{p,n-(p+1)}$$

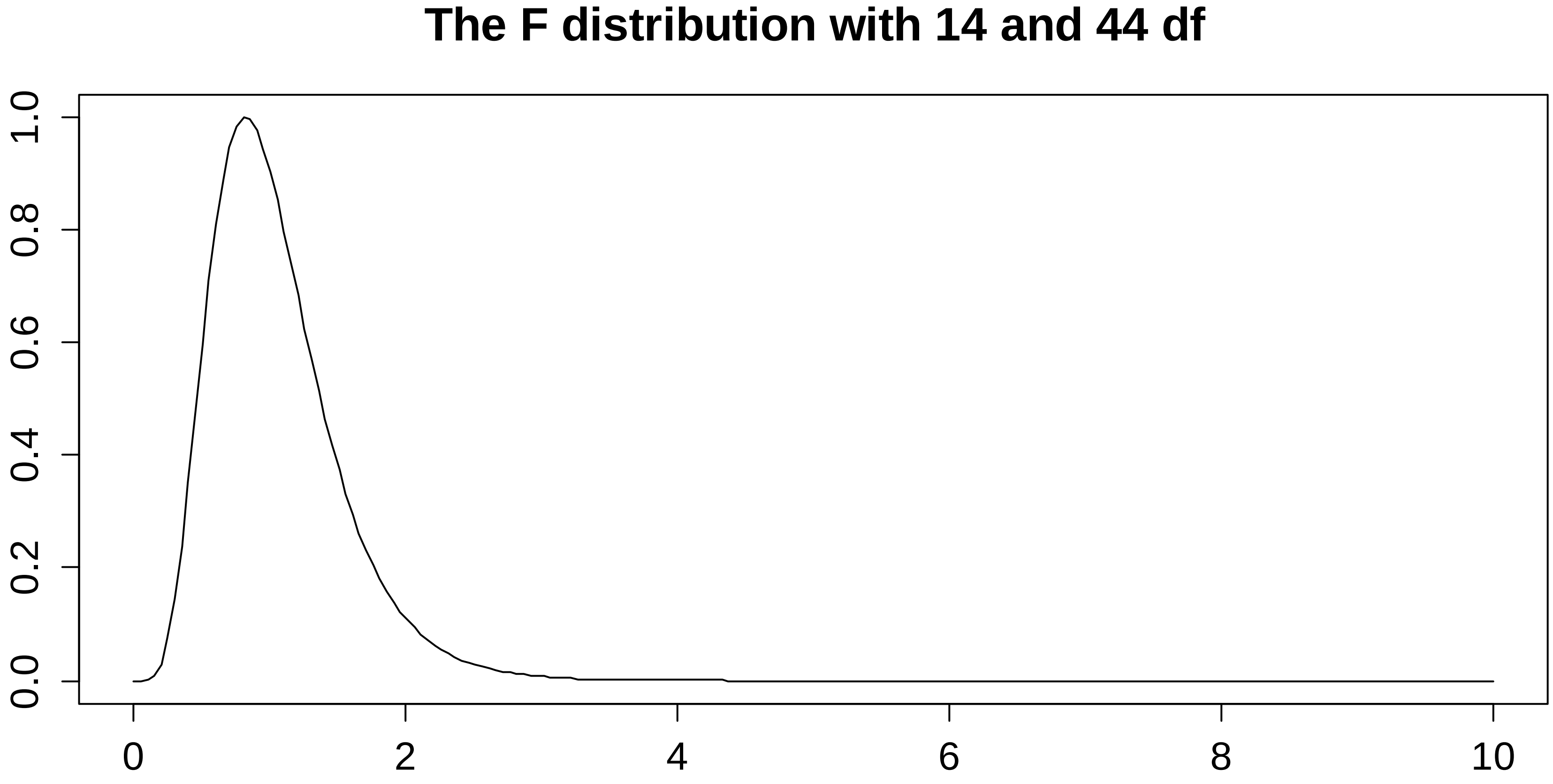
→ **Test statistic and p-value are shown in the R summary!**

Reading R-Output

```
> summary(fit)

Coefficients:   Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.297e+03  2.934e+02   4.422 6.32e-05 ***
JanTemp       -2.368e+00  8.851e-01  -2.676   0.0104 *
JulyTemp      -1.752e+00  2.031e+00  -0.863   0.3931
RelHum         3.852e-01  1.071e+00   0.360   0.7208
[output partially ommitted...]
log(NOx)       3.610e+01  1.415e+01   2.551   0.0142 *
log(SO2)      -3.687e+00  7.359e+00  -0.501   0.6189
---
Residual standard error: 34.48 on 44 degrees of freedom
Multiple R-squared: 0.7685, Adjusted R-squared: 0.6949
F-statistic: 10.43 on 14 and 44 DF, p-value: 8.793e-10
```

Note: due to space constraints, this is only a part of the output!

Density Function of the F-distribution



Applied Statistical Regression

AS 2015 – Multiple Regression

Prediction

The regression equation can be employed for predicting the response value in any given predictor configuration.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_{.1} + \hat{\beta}_2 x_{.2} + \ldots + \hat{\beta}_p x_{.p}$$

Note:

This can be a predictor configuration that was not part of the original data. For example a (new) city, for which only the predictors are known, but the mortality is not.

Be careful:

Only interpolation, i.e. prediction within the range of observed y-values works well, extrapolation yields non-reliable results.

Prediction in R

We can use the regression fit for predicting new observations. The syntax is as follows

```
> fit.big <- lm(Mortality ~ ., data=mt)
> dat       <- data.frame(JanTemp=..., ...)
> predict(fit.big, newdata=dat)
1 932.488
```

The x-values need to be provided in a data frame. The variable (column) names need to be identical to the predictor names. Of course, all predictors need to be present.

Then, it is simply applying the `predict()`-procedure.

Confidence- and Prediction Interval

The confidence interval for the fitted value and the prediction interval for future observation also exist in multiple regression.

a) 95%-CI for the fitted value $E[y \mid x]$

```
> predict(fit, newdata=dat, interval="conf")
```

b) 95%-PI for a future observation $\hat{y}$:

```
> predict(fit, newdata=dat, interval="pred")
```

- The visualization of these intervals is no longer possible in the case of multiple regression
- It is possible to write explicit formulae for the intervals using the matrix notation. We omit them here.

Versatility of Multiple Linear Regression

Despite that we are using linear models only, we have a versatile and powerful tool. While the response is always a continuous variable, different predictor types are allowed:

- **Continuous Predictors**
  Default case, e.g. *temperature*, *distance*, *pH-value*, …
- **Transformed Predictors**
  For example: $log(x),\ sqrt(x),\ arcsin(\sqrt{x}),...$
- **Powers**
  We can also use: $x^{-1},\ x^{2},\ x^{3},\ ...$
- **Categorical Predictors**
  Often used: *sex*, *day of week*, *political party*, …

Categorical Predictors

The canonical case in linear regression are *continuous predictor variables* such as for example:

→ *temperature, distance, pressure, velocity, ...*

While in linear regression, we *cannot have categorical response*, it is perfectly valid to have *categorical predictors*:

→ *yes/no, sex (m/f), type (a/b/c), shift (day/evening/night), ...*

Such categorical predictors are often also called **factor variables**. In a linear regression, each level of such a variable is encoded by a dummy variable, so that $(\ell - 1)$ degrees of freedom are spent.

Regression with a Factor Variable

The lathe (*in German: Drehbank*) **dataset:**

- y lifetime of a cutting tool in a turning machine
- $\tilde{x}$ tool type, A or B

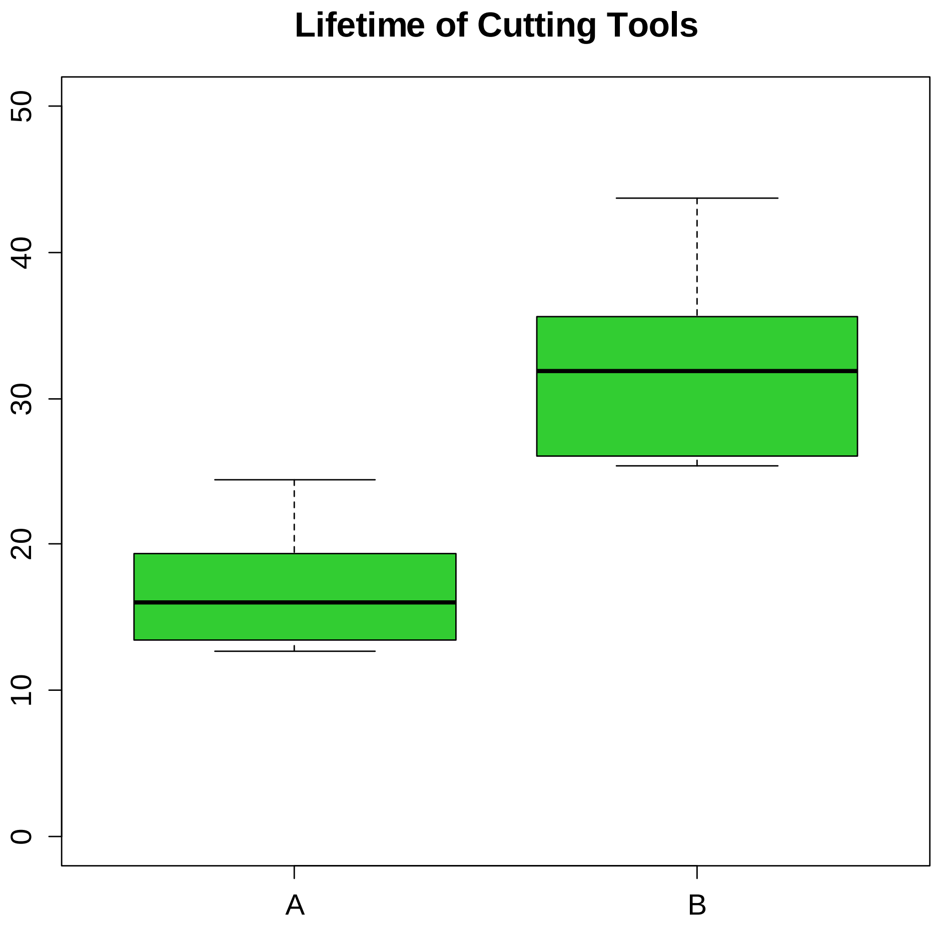
Dummy variable encoding:

$$x = \begin{cases} 0 & tool \;\; type \;\; A \\ 1 & tool \;\; type \;\; B \end{cases}$$

We set up a simple linear regression model:

$$y = \beta_0 + \beta_1 x + E$$

Typical Visualization & Question



Is the difference in lifetime between tools A and B just random, or is it significant?

Answer/Method:

→ 2-Sample-t-Test

→ Linear Regression

Interpretation of the Factor Model

→ **See blackboard…**

```
> summary(fit)

Call: lm(formula = hours ~ tool, data = lathe)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   17.110      1.628  10.508 4.14e-09 ***
toolB         14.818      2.303   6.435 4.68e-06 ***
---
Residual standard error: 5.149 on 18 degrees of freedom
Multiple R-squared: 0.697,  Adjusted R-squared: 0.6802
F-statistic: 41.41 on 1 and 18 DF,  p-value: 4.681e-06
```

Another View: t-Test

→ **The 1-factor-model is a t-test for non-paired data!**

```
> t.test(hours ~ tool, data=lathe, var.equal=TRUE)

Two Sample t-test

data:  hours by tool
t = -6.435, df = 18, p-value = 4.681e-06
alternative hypothesis: true diff in means is not 0
95 percent confidence interval:
 -19.655814  -9.980186
sample estimates:
mean in group A mean in group B
         17.110          31.928
```

Now: Continuous & Categorical Predictor

The lathe (*in German: Drehbank*) **dataset:**

- y lifetime of a cutting tool in a turning machine
- x_1 speed of the machine in rpm
- $\tilde{x}_2$ tool type A or B

Dummy variable encoding:

$$x_2 = \begin{cases} 0 & tool \;\; type \;\; A \\ 1 & tool \;\; type \;\; B \end{cases}$$

Multiple linear regression model:

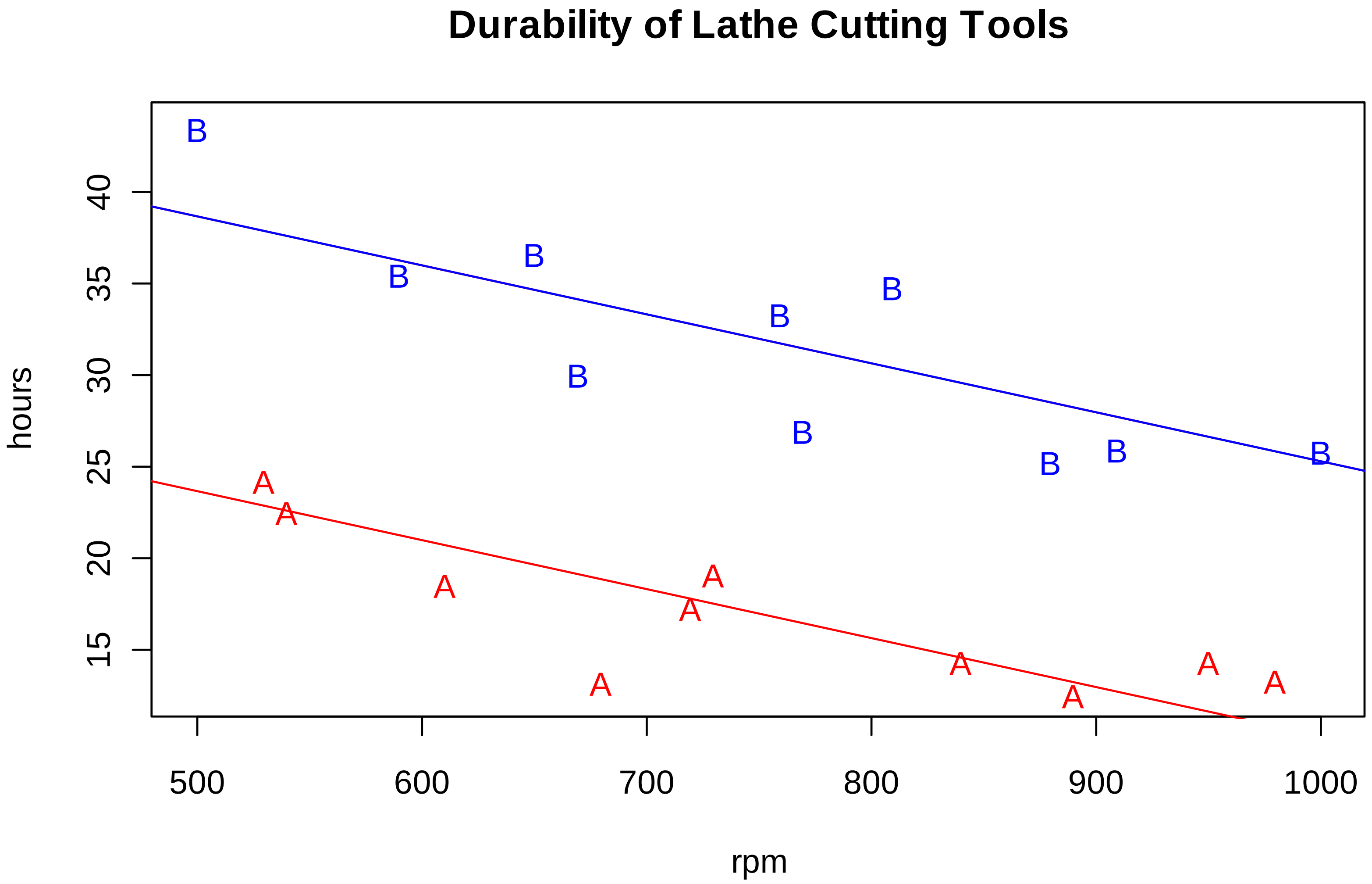
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + E$$

Interpretation of the Model

→ **see blackboard…**

```
> summary(lm(hours ~ rpm + tool, data = lathe))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 36.98560    3.51038  10.536 7.16e-09 ***
rpm         -0.02661    0.00452  -5.887 1.79e-05 ***
toolB       15.00425    1.35967  11.035 3.59e-09 ***
---
Residual standard error: 3.039 on 17 degrees of freedom
Multiple R-squared: 0.9003,  Adjusted R-squared: 0.8886
F-statistic: 76.75 on 2 and 17 DF,   p-value: 3.086e-09
```

The Dummy Variable Fit



A Model with Interactions

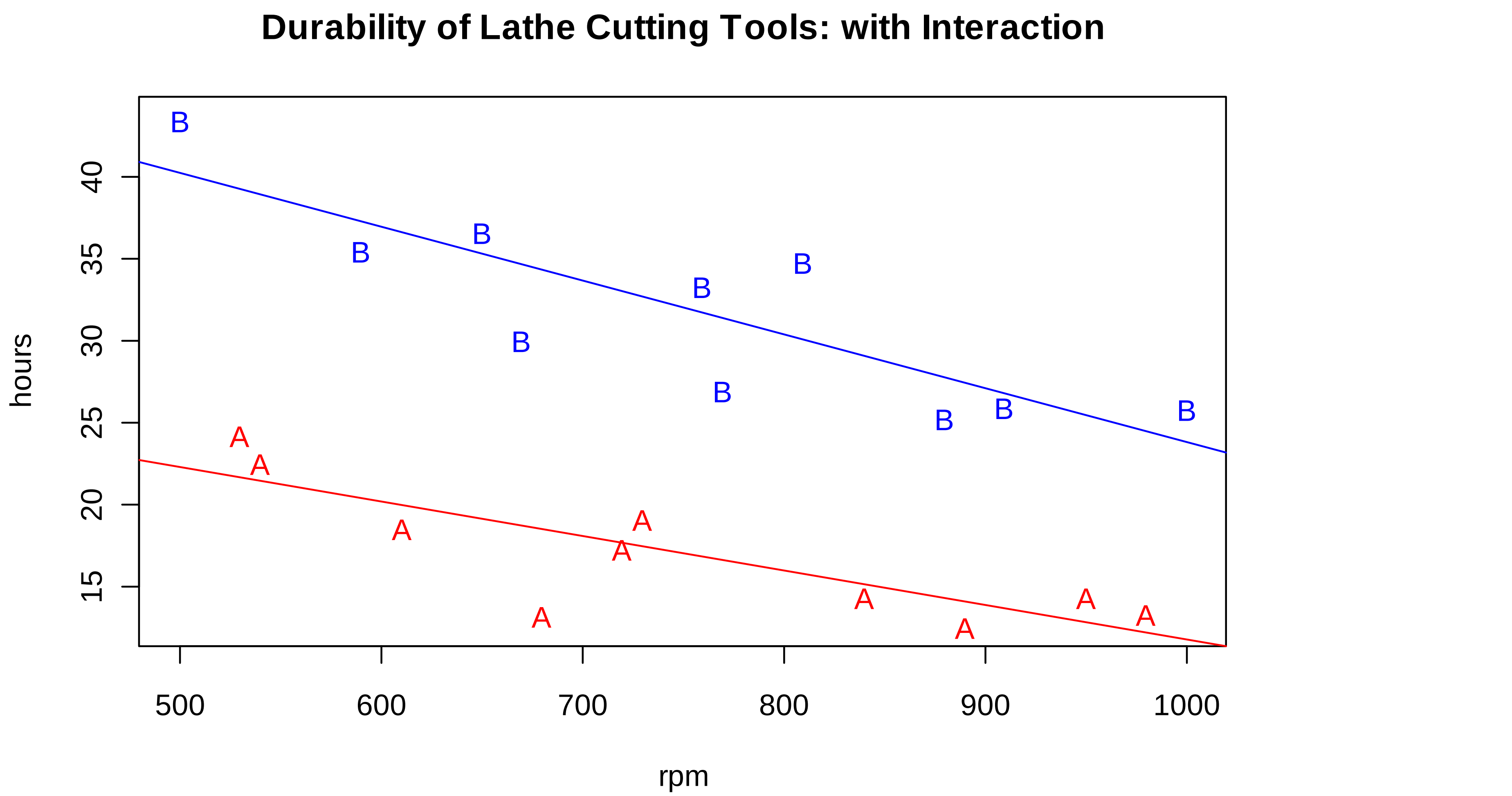
Question: do the slopes need to be identical?

→ with the appropriate model, the answer is no!

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + E$$

→ **see blackboard for model interpretation…**

Different Slopes for the Regression Lines



Summary Output

Result for the model with interaction term:

```
> summary(lm(hours ~ rpm + tool + rpm:tool, data=lathe))

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 32.774760   4.633472   7.073 2.63e-06 ***
rpm         -0.020970   0.006074  -3.452  0.00328 **
toolB       23.970593   6.768973   3.541  0.00272 **
rpm:toolB   -0.011944   0.008842  -1.351  0.19553
---
Residual standard error: 2.968 on 16 degrees of freedom
Multiple R-squared: 0.9105,  Adjusted R-squared: 0.8937
F-statistic: 54.25 on 3 and 16 DF,  p-value: 1.319e-08
```

→ The interaction term is not significant!

How Complex the Model Needs to Be?

Question 1: do we need different slopes for the two lines?

$$H_0 : \beta_3 = 0 \text{ against } H_A : \beta_3 \neq 0$$

→ no, see individual test for the interaction term on previous slide!

Question 2: is there any difference altogether?

$$H_0 : \beta_2 = \beta_3 = 0 \text{ against } H_A : \beta_2 \neq 0 \; and/or \; \beta_3 \neq 0$$

→ this is a hierarchical model comparison

→ we try to exclude interaction and dummy variable together

R offers convenient functionality for this test, see next slide!

Testing the Tool Type Variable

Hierarchical model comparison with anova():

```
> fit1 <- lm(hours ~ rpm, data=lathe)
> fit2 <- lm(hours ~ rpm + tool + rpm:tool, data=lathe)
> anova(fit1, fit2)
Model 1: hours ~ rpm
Model 2: hours ~ rpm + tool + rpm:tool
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1      18 1282.08
2      16  140.98  2    1141.1 64.755 2.137e-08 ***
```

→ Our model `fit2`, where the tool type has an interaction with `rpm`, performs significantly better than the simpler model where only `rpm` is present. The best model is in between.

Categorical Input with More Than 2 Levels

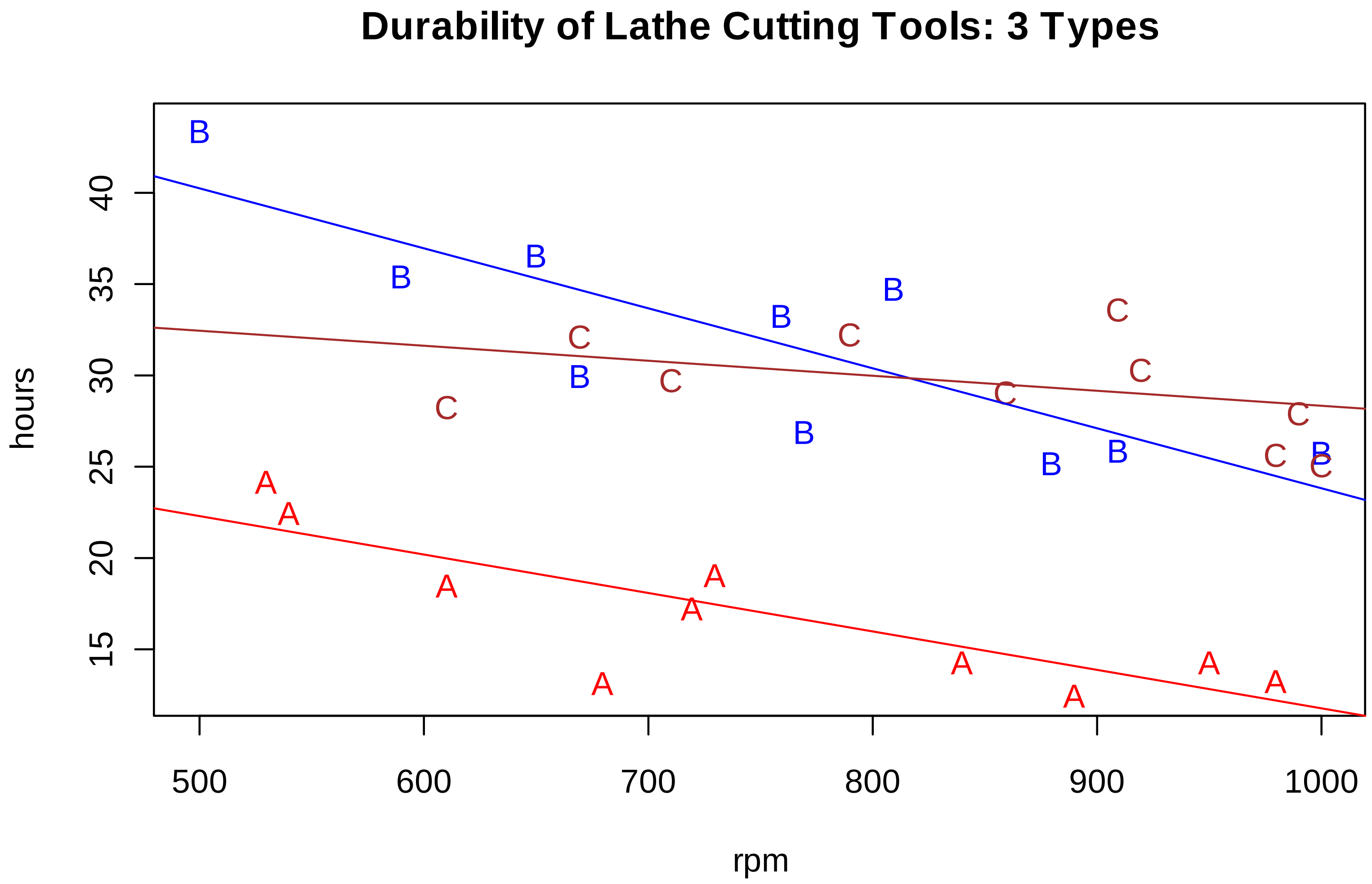
Variable $\tilde{x}_2$ is categorical, there are now 3 levels A, B, C.
We encode this information by two dummy variables x_2 and x_3:

x_2	x_3	
0	0	*for observations of type A*
1	0	*for observations of type B*
0	1	*for observations of type C*

Main effect model: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + E$

With interactions: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1 x_2 + \beta_5 x_1 x_3 + E$

Three Types of Cutting Tools



Summary Output

```
> summary(lm(hours ~ rpm + tool + rpm:tool, data = abc.lathe)

Coefficients:Estimate Std. Error t value Pr(>|t|)
(Intercept) 32.774760   4.496024   7.290 1.57e-07 ***
rpm         -0.020970   0.005894  -3.558  0.00160 **
toolB       23.970593   6.568177   3.650  0.00127 **
toolC        3.803941   7.334477   0.519  0.60876
rpm:toolB   -0.011944   0.008579  -1.392  0.17664
rpm:toolC    0.012751   0.008984   1.419  0.16869
---
Residual standard error: 2.88 on 24 degrees of freedom
Multiple R-squared: 0.8906,    Adjusted R-squared: 0.8678
F-statistic: 39.08 on 5 and 24 DF,  p-value: 9.064e-11
```

This summary is of limited use for deciding about model complexity. We require hierarchical model comparisons!

Inference with Factor Variables

In a regression model where factor variables that have >2 levels and/or interaction terms are present, the `summary()` function does not provide useful information for variable selection. We have to work with `drop1()` instead!

```
> drop1(fit.abc, test="F")

Single term deletions

Model: hours ~ rpm + tool + rpm:tool
         Df Sum of Sq    RSS    AIC F value  Pr(>F)
<none>                199.10 68.779
rpm:tool  2    59.688 258.79 72.645  3.5974 0.04301 *
```

`drop1()` performs correct model comparisons and respects the model hierarchy. In our particular example, the interaction term is significant and should stay in the model!

Inference with Categorical Predictors

Do not perform individual hypothesis tests on factors that have more than 2 levels, they are meaningless! Hierarchical model comparisons are the alternative.

Question 1: do we have different slopes?

$H_0: \beta_4 = 0 \;\; and \;\; \beta_5 = 0$ against $H_A: \beta_4 \neq 0 \;\; and/or \;\; \beta_5 \neq 0$

Question 2: is there any difference altogether?

$H_0: \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$ against $H_A: any \; of \;\; \beta_2, \beta_3, \beta_4, \beta_5 \neq 0$

→ Again, R provides convenient functionality: `anova()`

Are the Tools Different?

```
> f.sma <- lm(hours ~ rpm, data=abc.lathe)
> f.big <- lm(hours ~ rpm + tool + rpm:tool, data=abc.lathe)
> anova(f.sma, f.big)
Analysis of Variance Table

Model 1: hours ~ rpm
Model 2: hours ~ rpm + tool + rpm:tool
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     28 1681.3
2  24  199.1  4    1482.2 44.665 8.811e-11 ***
```

→ There is a highly significant difference among the tool types if variable `rpm` is taken into account.

→ The decrease in lifetime with increasing rpm is not the same for the 3 tool types. However, with $p = 0.043$ only weakly significant.

Residual Analysis – Model Diagnostics

Why do it? And what is it good for?

a) To make sure that estimates and inference are valid

- $E[E_i] = 0$
- $Var(E_i) = \sigma_E^2$
- $Cov(E_i, E_j) = 0$
- $E_i \sim N(0, \sigma_E^2 I),\ i.i.d$

b) Identifying unusual observations

Often, there are just a few observations which "are not in accordance" with a model. However, these few can have strong impact on model choice, estimates and fit.

Residual Analysis – Model Diagnostics

Why do it? And what is it good for?

c) Improving the model

- Transformations of predictors and response
- Identifying further predictors or interaction terms
- Applying more general regression models

- There are both model diagnostic graphics, as well as numerical summaries. The latter require little intuition and can be easier to interpret.

- However, the graphical methods are far more powerful and flexible, and are thus to be preferred!

Applied Statistical Regression
AS 2015 – Multiple Regression

Residuals vs. Errors

All requirements that we made were for the errors E_i. However, they cannot be observed in practice. All that we are left with are the residuals r_i , which are only estimates of the errors.

But:

- The residuals R_i do share some properties of the errors E_i , but not all – there are some important differences.
- In particular, even in cases where the R_i are uncorrelated and have constant variance, the residuals R_i feature some estimation-related correlation and non-constant variance.

→ *Does residual analysis make sense?*

Standardized/Studentized Residuals

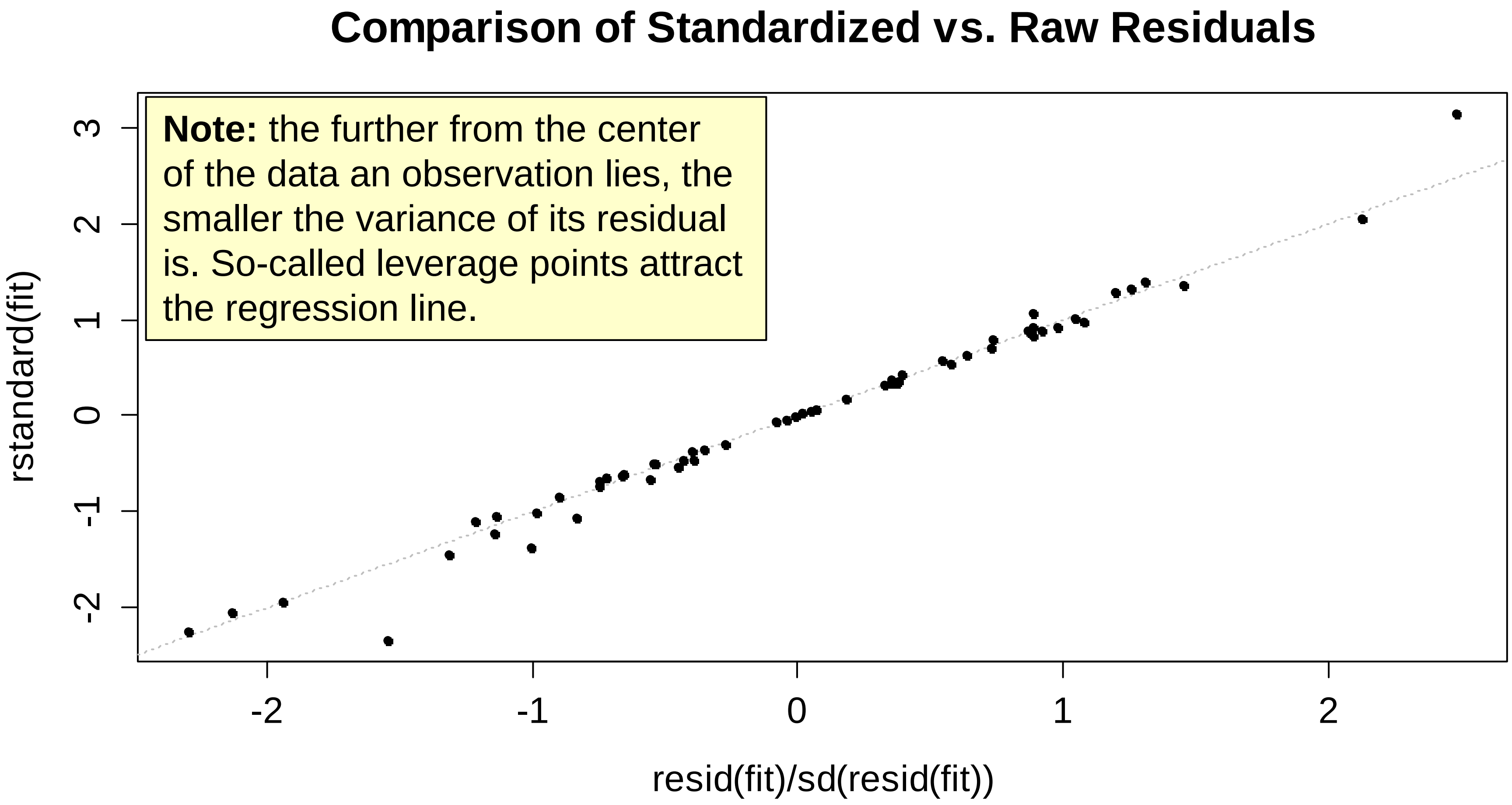
- The *estimation-induced* correlation and heteroskedasticity in the residuals R_i is usually very small. Thus, residual analysis using the raw residuals r_i is both useful and sensible.

- One can try to improve the raw residual r_i with dividing it by an estimate of its standard deviation.

  $\tilde{r}_i = \dfrac{r_i}{\hat{\sigma}_E \cdot \sqrt{1-H_{ii}}}$, H_{ii} is the diagonal element of hat matrix

  If $\hat{\sigma}_E$ is the residual standard error, we speak of *Standardized Residuals*. Sometimes, one also uses a different estimate $\hat{\sigma}_E$ that was obtained by ignoring the i^{th} data point. One then speaks of *Studentized Residuals*.

Standardized vs. Raw Residuals



Note: the further from the center of the data an observation lies, the smaller the variance of its residual is. So-called leverage points attract the regression line.

Toolbox for Model Diagnostics

There are 4 "standard plots" in R:

- Residuals vs. Fitted, *aka* Tukey-Anscombe-Plot
- Normal Plot (*uses standardized residuals*)
- Scale-Location-Plot (*uses standardized residuals*)
- Leverage-Plot (*uses standardized residuals*)

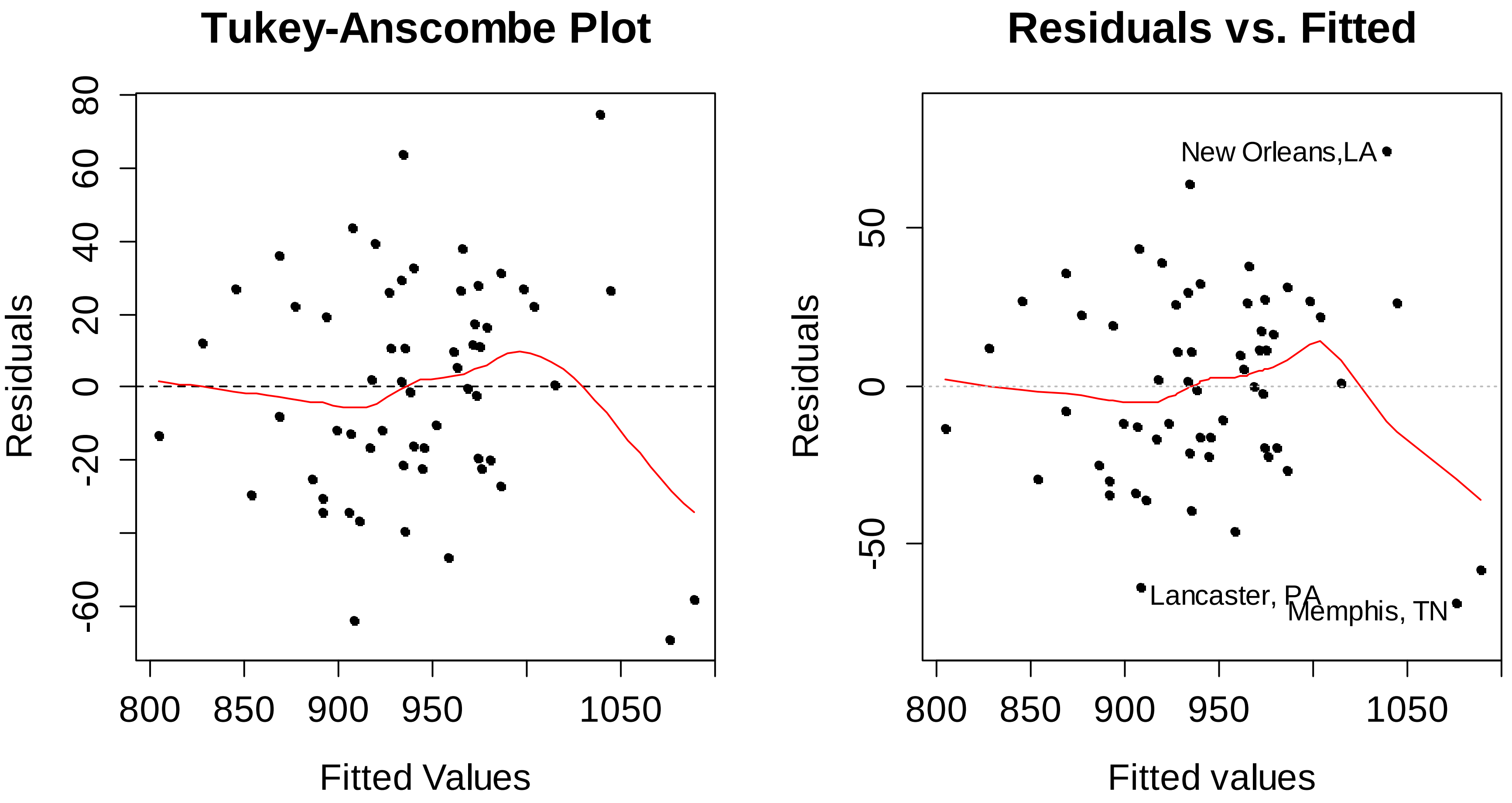
In R: > plot(fit)

Some further tricks and ideas:

- Residuals vs. predictors
- Partial residual plots
- Residuals vs. other, arbitrary variables
- Important: Residuals vs. time/sequence

Tukey-Anscombe-Plot: Residuals vs. Fitted

Plot r_i vs. $\hat{y}_i$: `> plot(fit, which=1, pch=20)`



Tukey-Anscombe-Plot: Residuals vs. Fitted

Some statements:

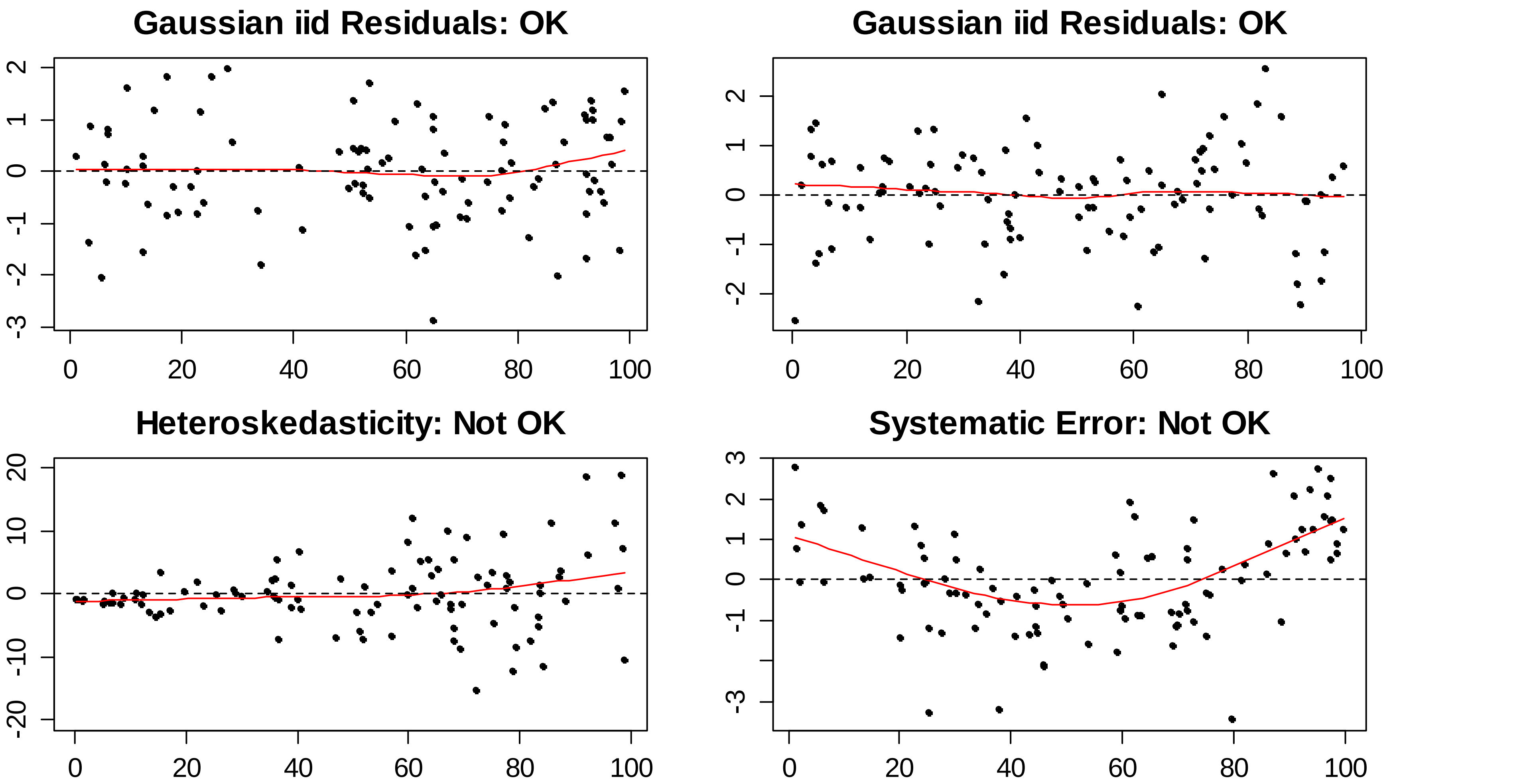
- is the most important residuals plot!
- is useful for finding structural model deficiencies $E[E_i] \neq 0$
- if $E[E_i] \neq 0$, the response/predictor relation might be nonlinear, or some important predictors/interactions may be missing.
- it is also possible to detect non-constant variance (→ then, the smoother does not deviate from 0)

When is the plot OK?

- the residuals scatter around the x-axis without any structure
- the smoother line is horizontal, with no systematic deviation
- there are no outliers

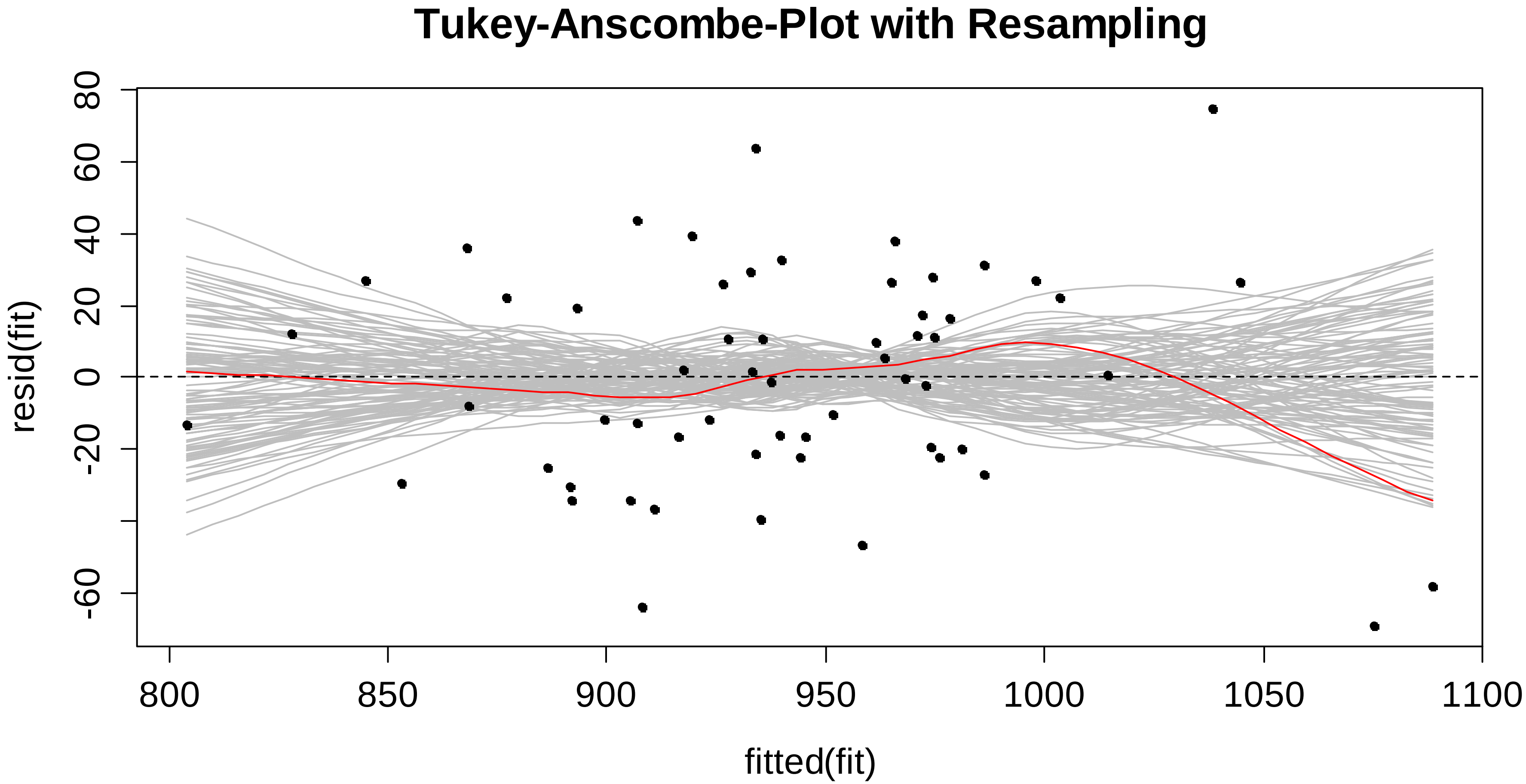
Applied Statistical Regression
AS 2015 – Multiple Regression

Tukey-Anscombe-Plot: Residuals vs. Fitted



Tukey-Anscombe-Plot: Residuals vs. Fitted

```
> resplot(fit, plots=1)
```



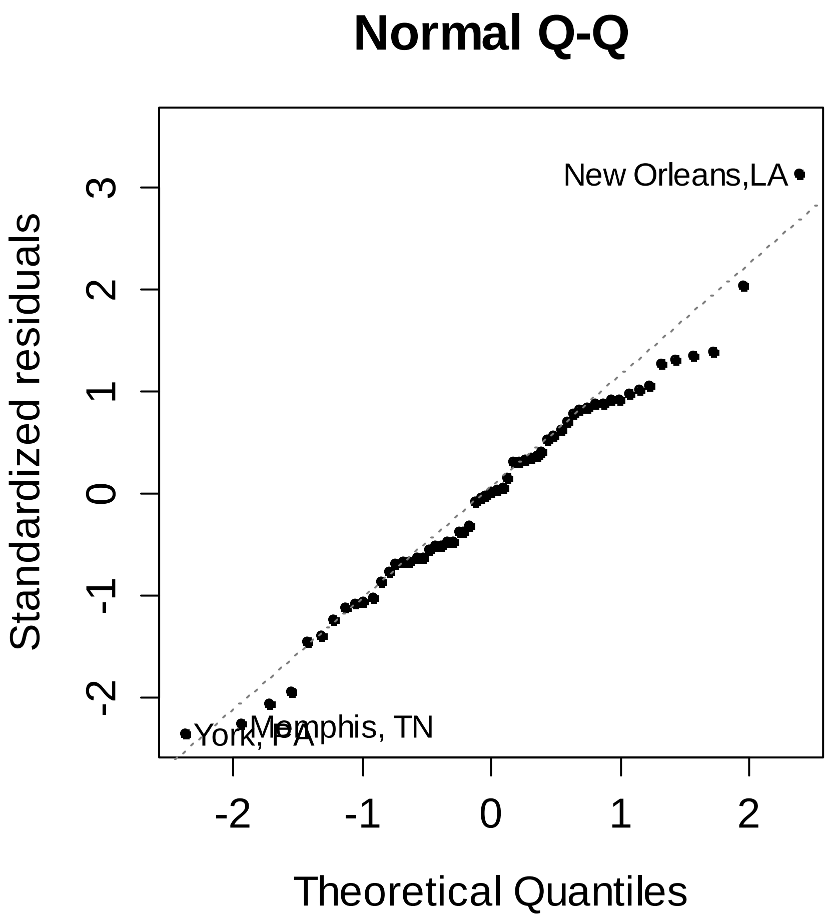
Tukey-Anscombe-Plot

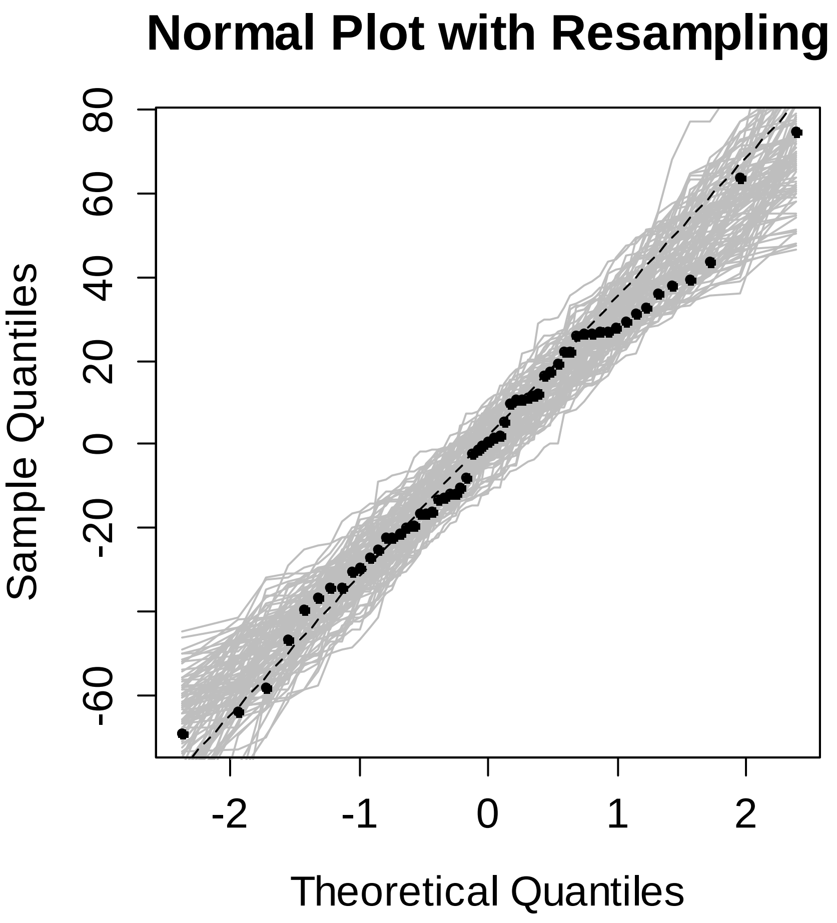
If the Tukey-Anscombe-Plot is not OK:

- If a systematic error is present, i.e. if the smoother deviates from the x-axis and hence $E[E_i] \neq 0$, it is mandatory to take some action. We recommend:
  - "*fit a better model*". In many cases, performing some log-transformations on the response and/or predictor(s) helps.
  - sometimes it also means that important predictors are missing. These can be completely novel variables, terms of higher order or interaction term.
- Non-constant variance: transformations usually help!

Normal Plot

Plot $\tilde{r}_i$ vs. `qnorm(i/(n+1),0,1)`: `> plot(fit, which=2)`





Applied Statistical Regression

AS 2015 – Multiple Regression

Normal Plot

Is useful for:

- for identifying non-iid or non-Gaussian errors: $E_i \overset{!}{\sim} N(0, \sigma_E^2 I)$

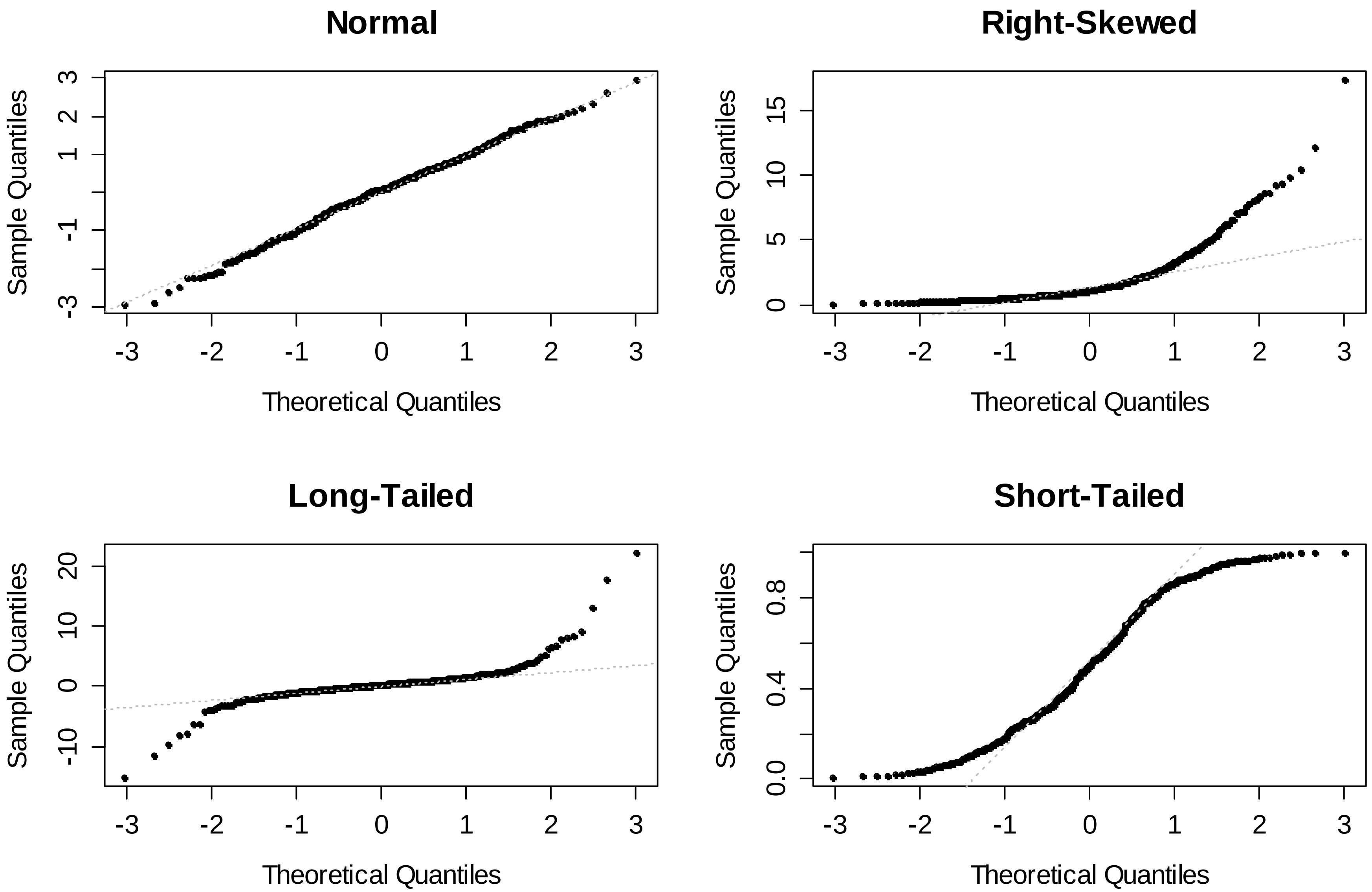
When is the plot OK?

- the residuals $\tilde{r}_i$ must not show any systematic deviation from line which leads to the 1st and 3rd quartile.
- a few data points that are slightly "off the line" near the ends are always encountered and usually tolerable
- skewed residuals need correction: they usually tell that the model structure is not correct. Transformations may help.
- long-tailed, but symmetrical residuals are not optimal either, but often tolerable. Alternative: *robust regression*!

Applied Statistical Regression

AS 2015 – Multiple Regression

Normal Plot

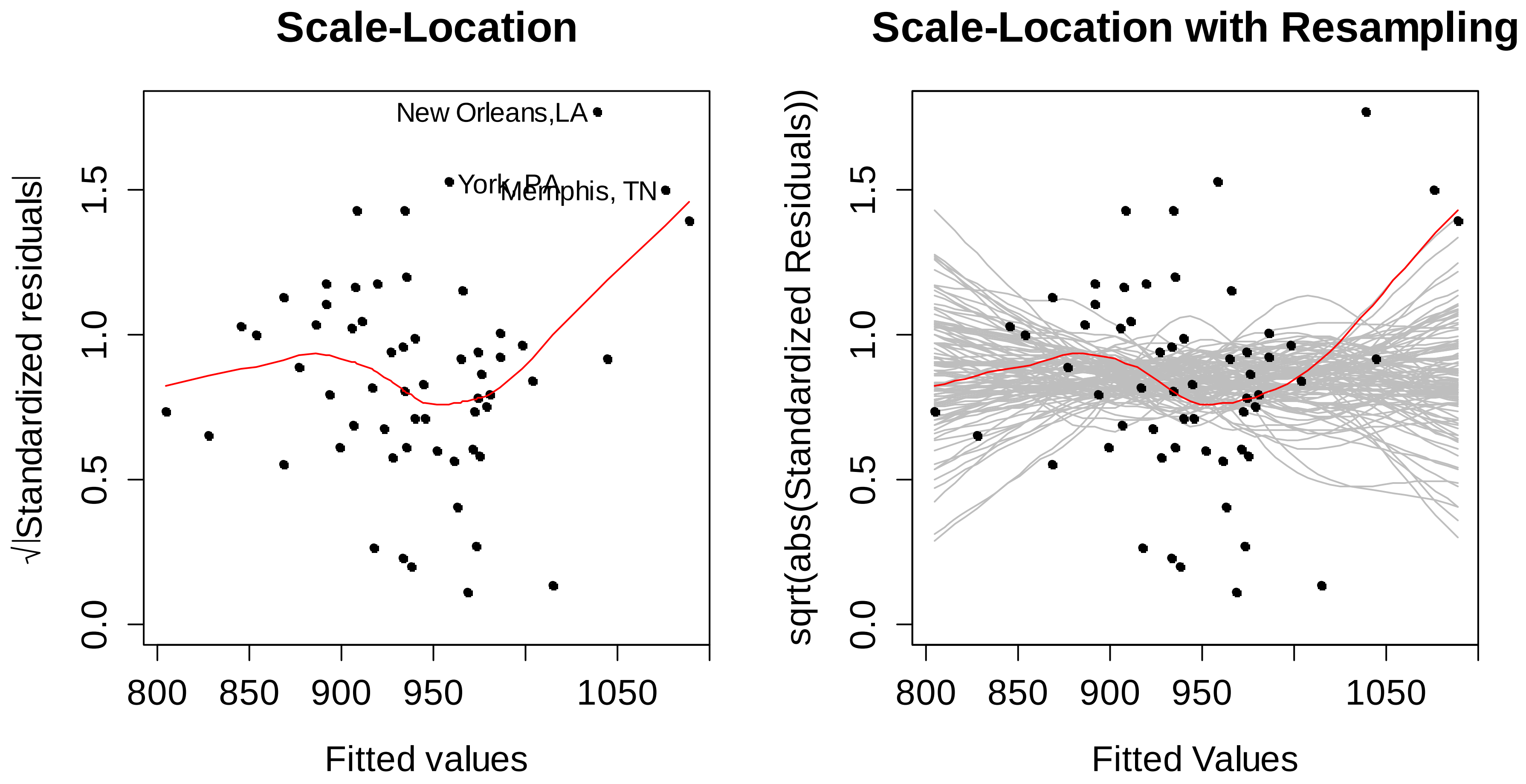


Applied Statistical Regression

AS 2015 – Multiple Regression

Scale-Location-Plot: $\sqrt{|\tilde{r}_i|}$ ***vs.*** $\hat{y}_i$

```
> plot(fit, which=3) oder resplot(fit, plots=3)
```



Scale-Location-Plot

Is useful for:

- identifying heteroskedasticity: $Var(E_i) \neq \sigma_E^2$
- if that is the case, the model often has structural deficencies, i.e. the fitted relation is not correct. Use a transformation!
- there are cases where we expect non-constant variance and do not want to use a transformation. This can the be tackled by applying *weighted regression*.

When is the plot OK?

- the smoother line runs horizontally along the x-axis, without any systematic deviations.

Unusual Observations

- There can be observations which do not fit well with a particular model. These are called ***outliers***. The property of being an outlier strongly depends on the model used.

- There can be data points which have strong impact on the fitting of the model. These are called ***influential observations***.

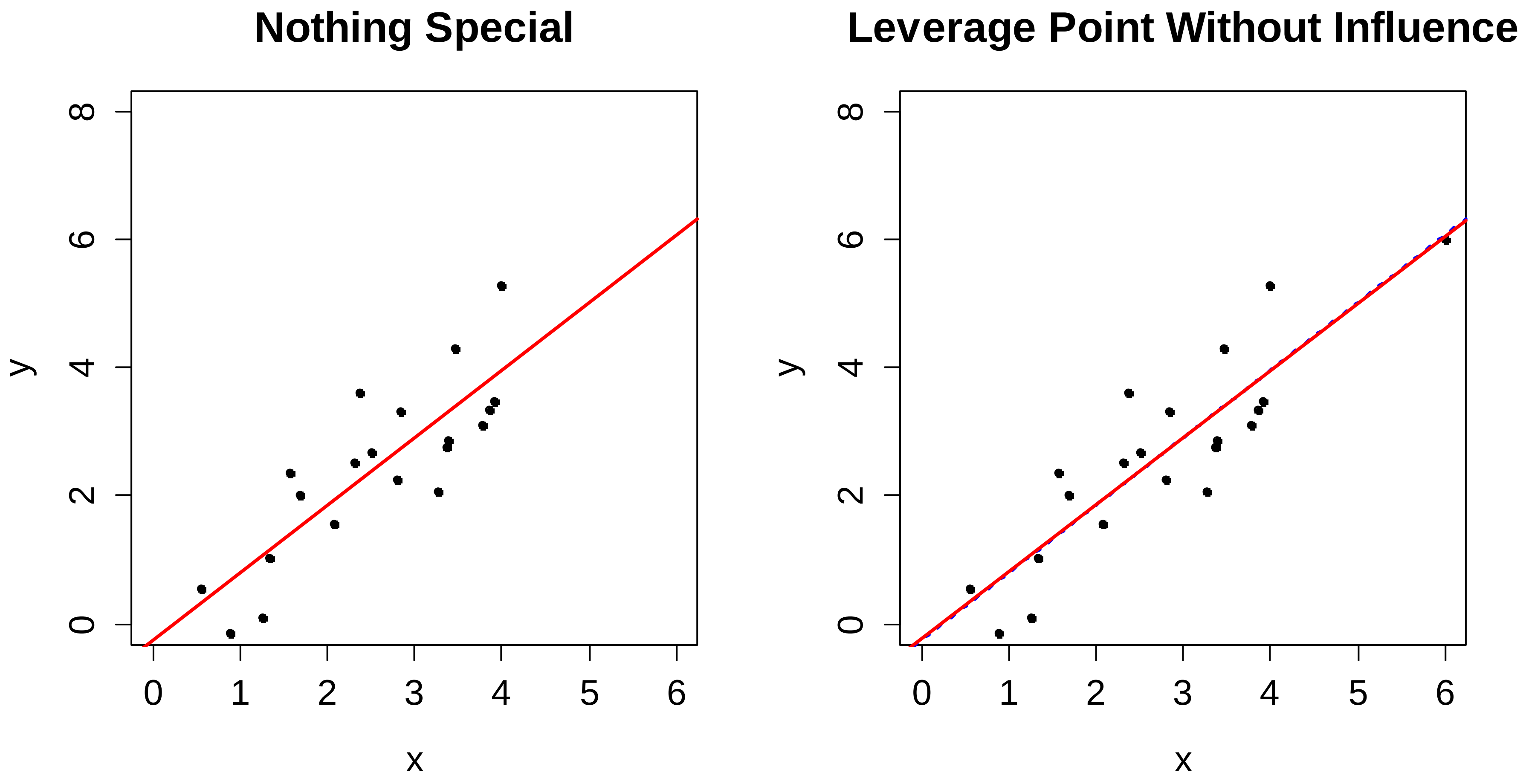
- A data point must fall under **none, one or both** the above definitions – there is no other option.

- A ***leverage point*** is an observation that lies at a "different spot" in predictor space. This is potentially dangerous, because it can have strong influence on the fit.

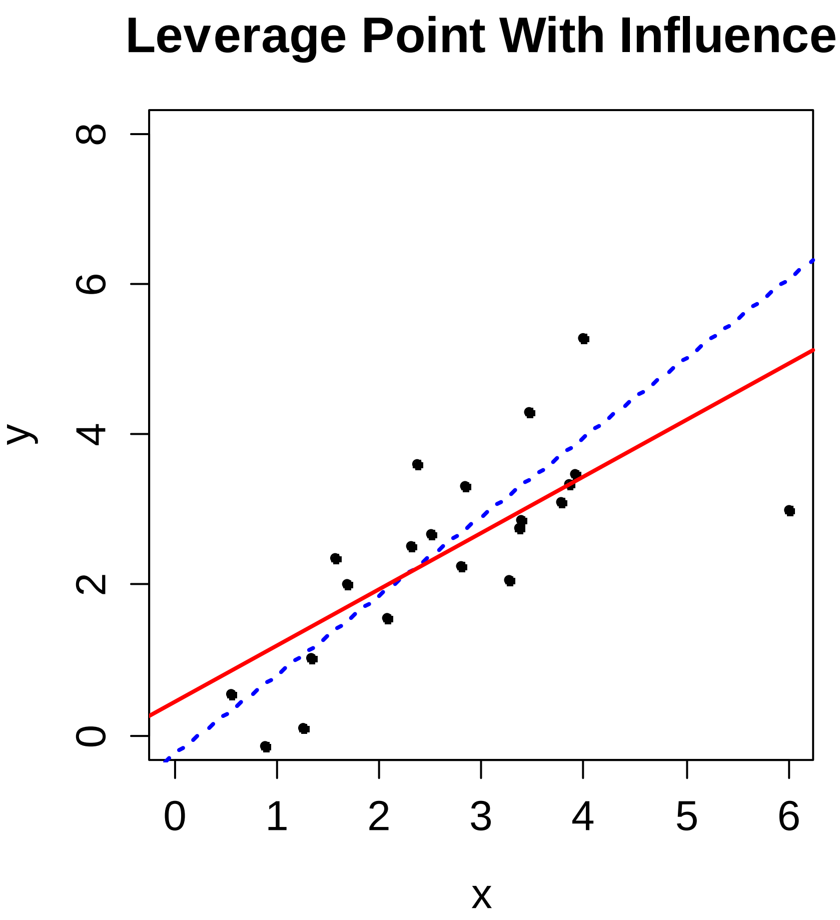
Applied Statistical Regression

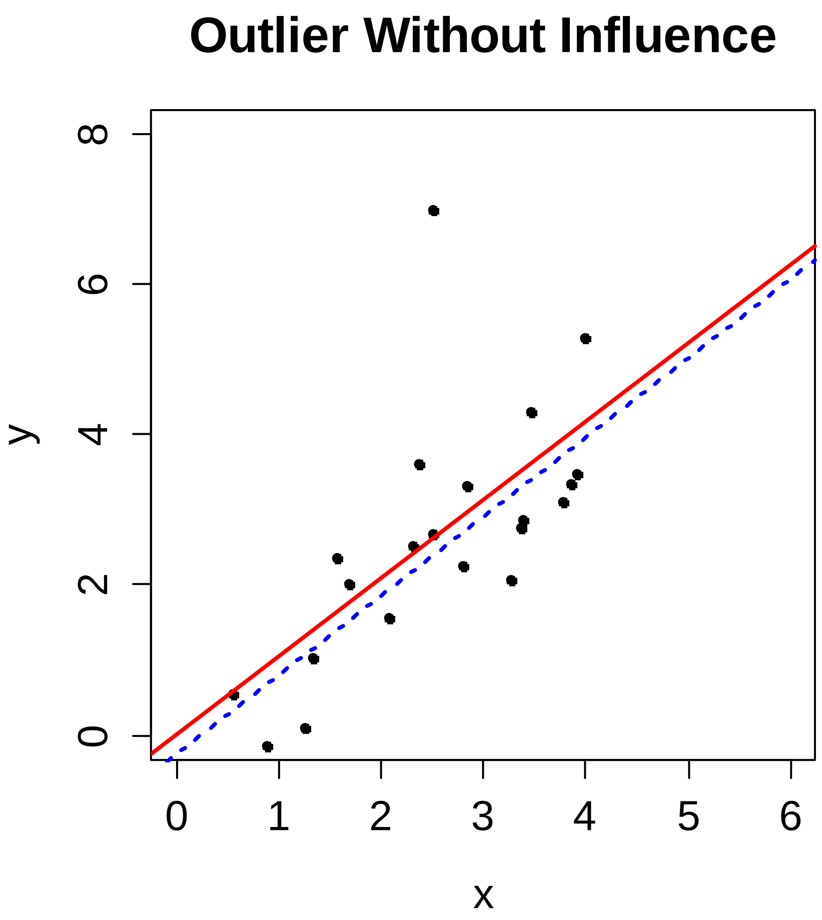
AS 2015 – Multiple Regression

Unusual Observations



Unusual Observations


Leverage Point With Influence


Outlier Without Influence

Influence Diagnostics

The effect of a single data point on the regression results can be inferred, if the analysis is repeated without that particular data point.

→ *Repeating this for all data points requires computing and evaluating n regressions. This is pretty laborious!*

→ *Moreover, a quantitative criterion is required, with which the change in the results over all data points is captured and measured.*

The concepts of **Leverage** and **Cook's Distance** allow for pinning down the change in the results when data points are omitted, and this even without recomputing the regression.

Leverage

The leverage of data point i quantifies, how atypical it is positioned in predictor space. The further from the bulk a data point lies, the more it can attract the regression line.

If y_i changes by Δy_i, then $H_{ii}\Delta y_i$ is the change in $\hat{y}_i$. High leverage means that the i^{th} data point may force the regression relation to strongly adapt to the data point.

Remarks:

- Leverage points are different from the bulk of data
- The average value for leverage is given by $(p+1)/n$
- We say a data point has high leverage if $H_{ii} > 2(p+1)/n$

Cook's Distance

Cook's Distance is a computational concept, with which the potential change in all the fitted values can be measured if data point i is omitted from the analysis. We have:

$$D_i = \frac{\sum (\hat{y}_k^{[-i]} - \hat{y}_k)^2}{(p+1)\sigma_E^2} = \frac{H_{ii}}{1-H_{ii}} \cdot \frac{\tilde{r}_i^2}{(p+1)}$$

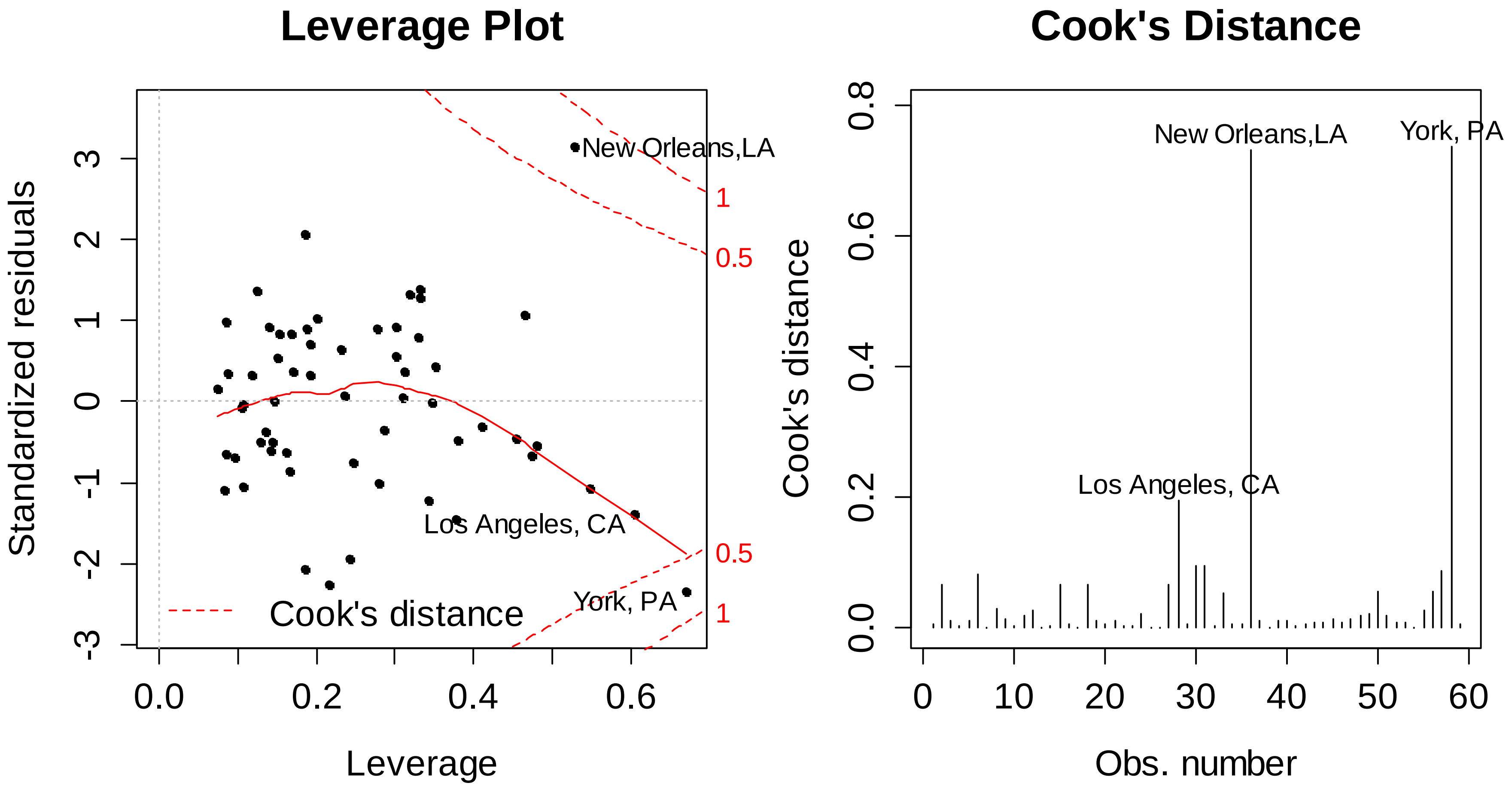
Cook's Distance can be computed directly, i.e. without fitting the regression n times. It measures the influence of a data point and depends on leverage and standardized residual.

Hint:

→ Data points with $D_i > 0.5$ are called influential. If $D_i > 1$ the data point is potentially damaging to the regression problem.

Leverage-Plot: $\tilde{r}_i$ vs. H_{ii} & Cook’s Distance

```
> plot(fit, which=5) bzw. plot(fit, which=4)
```



Leverage-Plot & Cook's Distance

Is useful for:

- identifying outliers, leverage points, influential observations and uncritical data points at one and the same time.

When is the plot OK?

- no extreme outliers in y-direction, no matter where
- high leverage, here $h_{ii} > 2(p+1)/n = 2(4+1)/50 = 0.2$ is always potentially dangerous, especially if it is in conjunction with large residuals!
- This is visualized by the Cook's Distance lines in the plot: >0.5 requires attention, >1 potentially damaging!

How to Deal with Influential Observations?

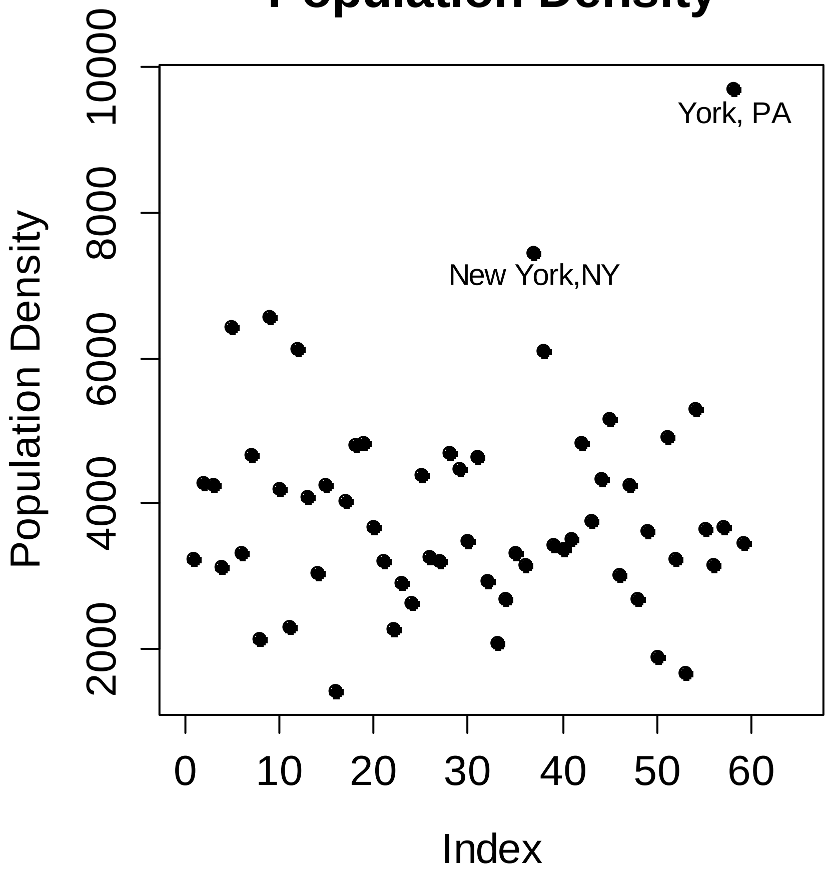
What can be done with data points that have $D_i > 0.5$:

- First check the data for gross errors, misprints, typos, ...
- Influential observations often appear if the input is not suitable, i.e. if predictors are extremely skewed, because log-transformations were forgotten.
- Simply omitting these data points is always a delicate matter and should at least be reported openly. Often, influential data points tell much about the benefits and limits of a model and create opportunities and ideas as how to improve a model.

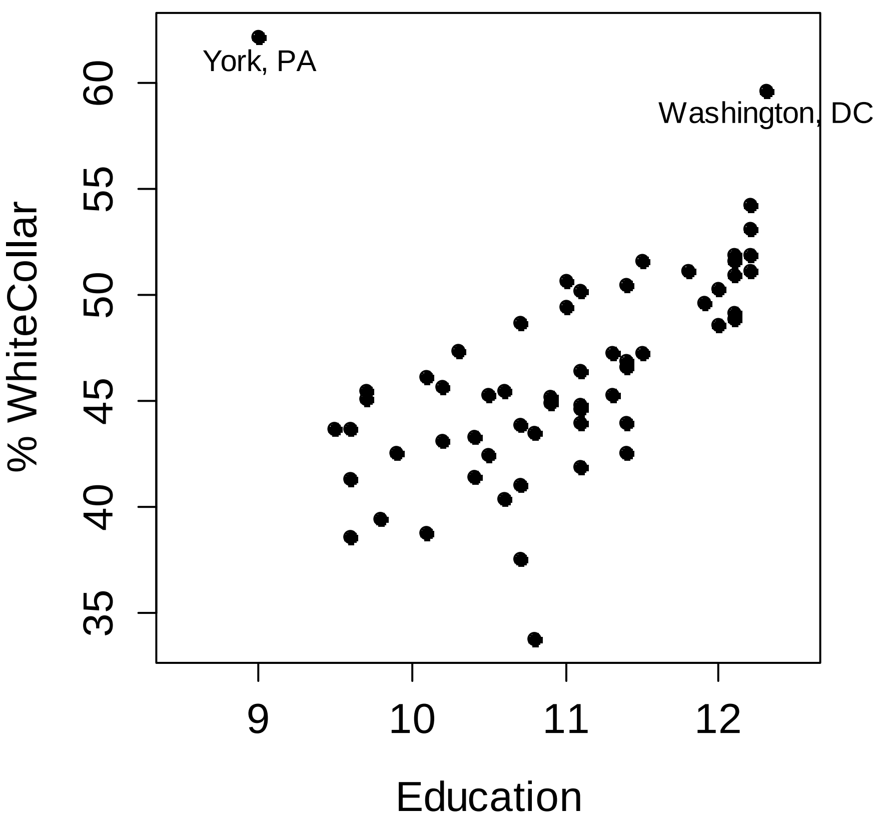
Applied Statistical Regression

AS 2015 – Multiple Regression

Beispiel: Mortality Dataset



Population Density



% White Collar vs. Education

More Residual Plots

General Remark:

We are allowed to plot the residuals versus any arbitrary variable we wish. This includes:

- predictors that were used
- potential predictors which were not (yet) used
- other variables, e.g. time/sequence of the observations

The rule is:

No matter what the residuals are plotted against, there must not be any non-random structure. Else, the model has some deficiencies, and needs improvement!

Example

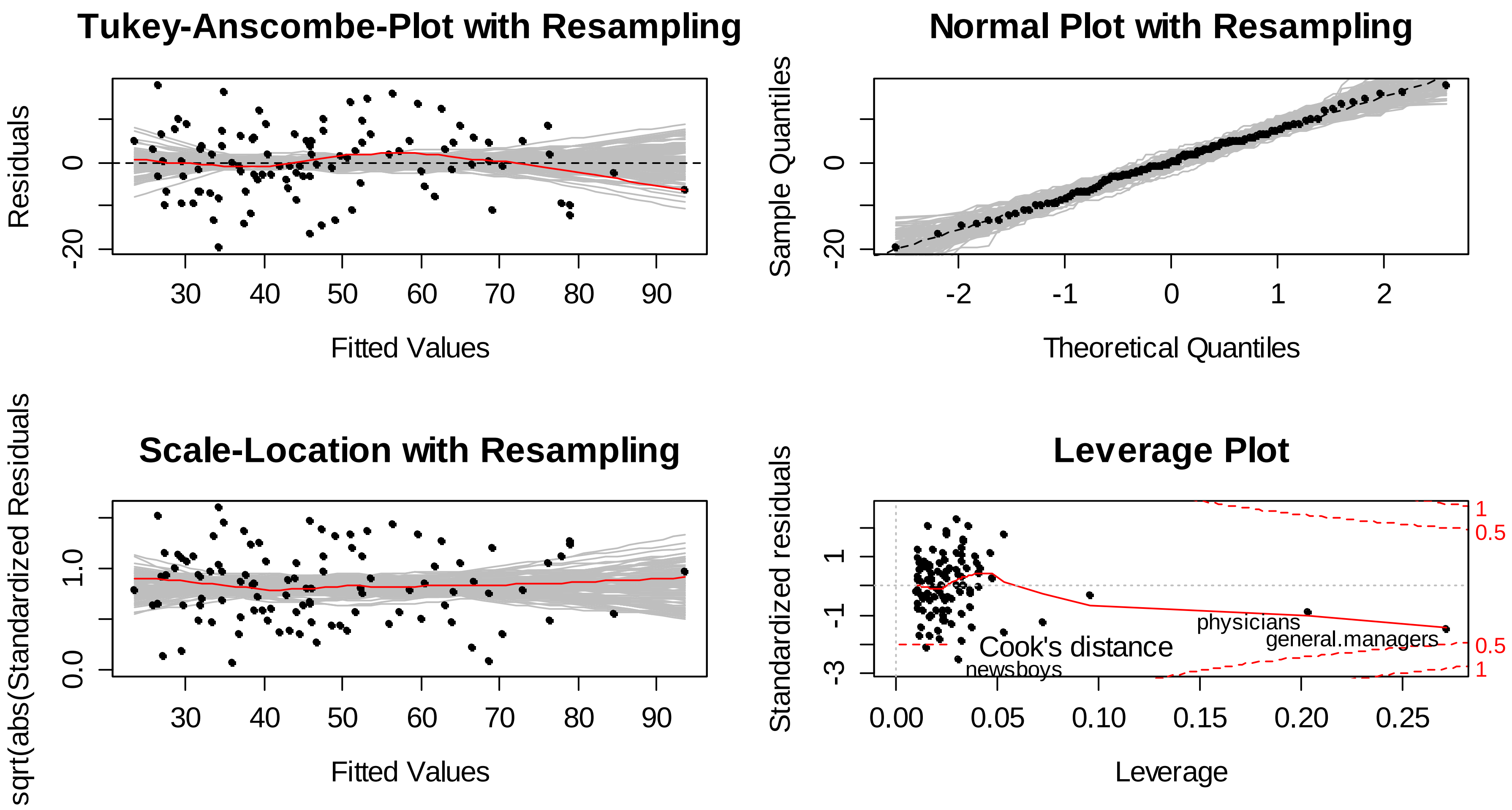
Description of the Dataset:

We are given a measurement of the prestige of 102 different profession. Moreover, we have 5 different variables that could be used as predictors. The data origin from Canada.

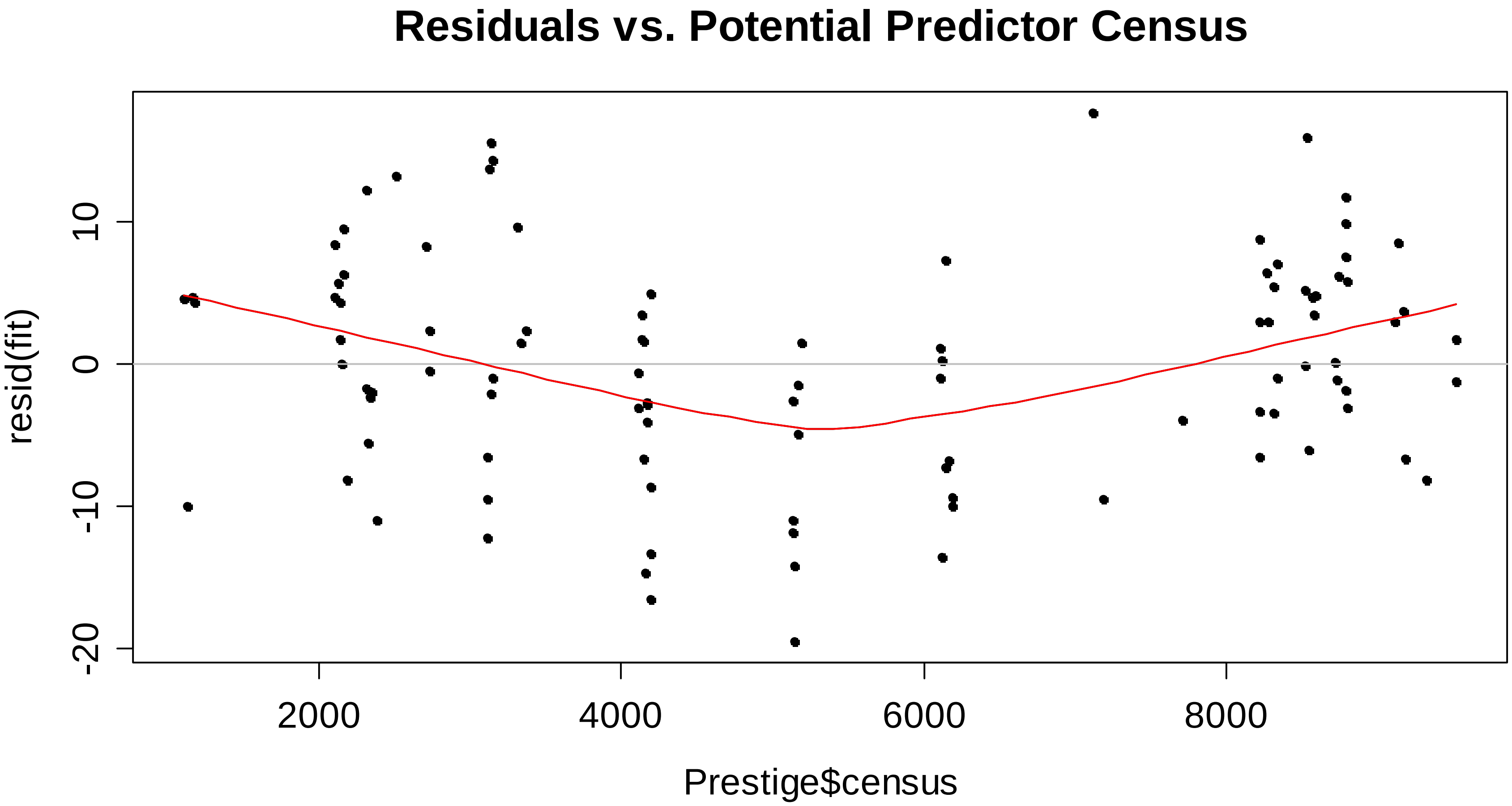
```
                      educ income  women prest  cens  type
gov.administrator    13.11  12351  11.16  68.8  1113  prof
general.managers     12.26  25879   4.02  69.1  1130  prof
accountants          12.77   9271  15.70  63.4  1171  prof
```

We start with fitting the model: `prestige ~ income + education`, the other three remaining (potential) predictors variables are first omitted in order the study the deficiencies in the model.

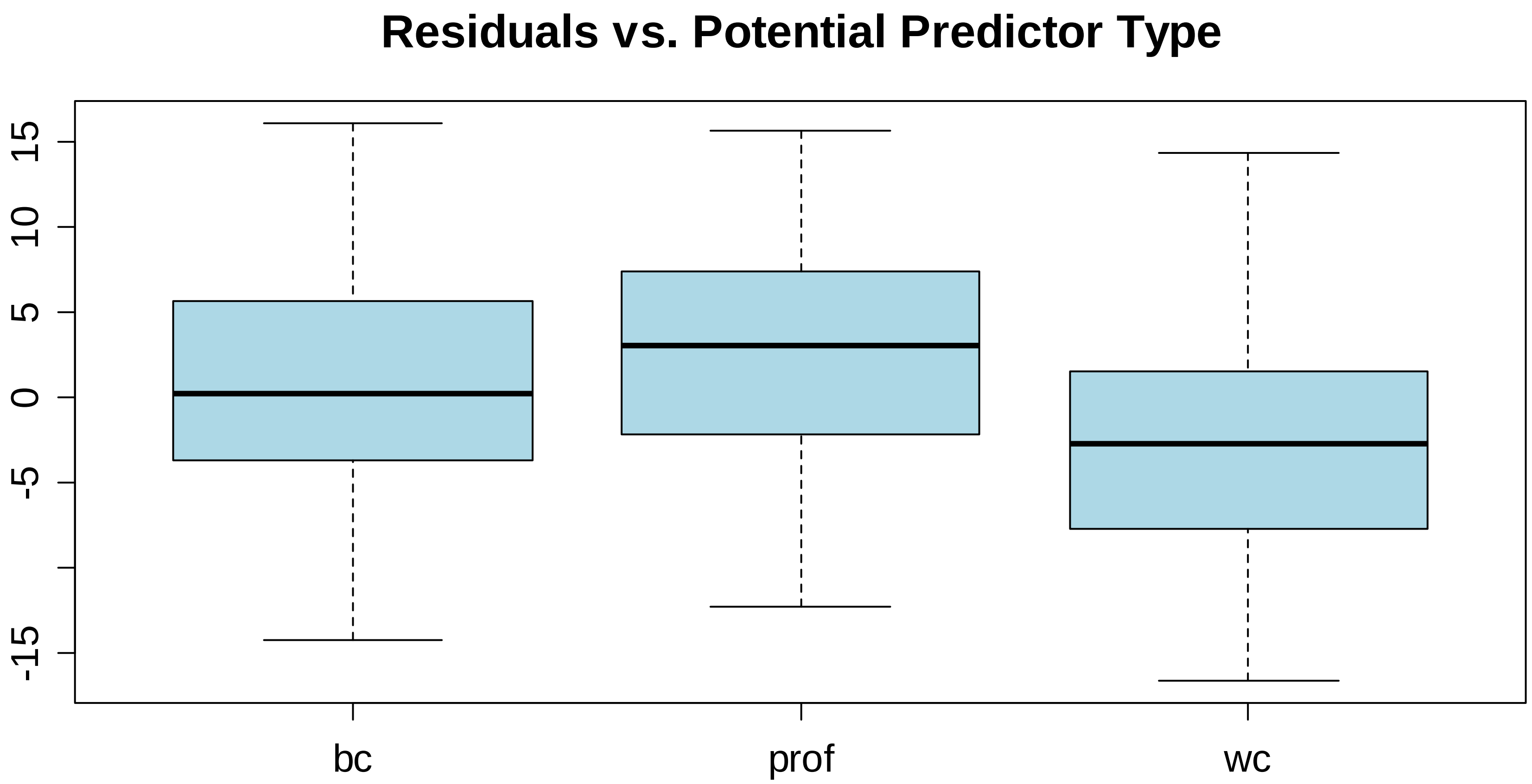
Standard Residual Plots with 2 Predictors



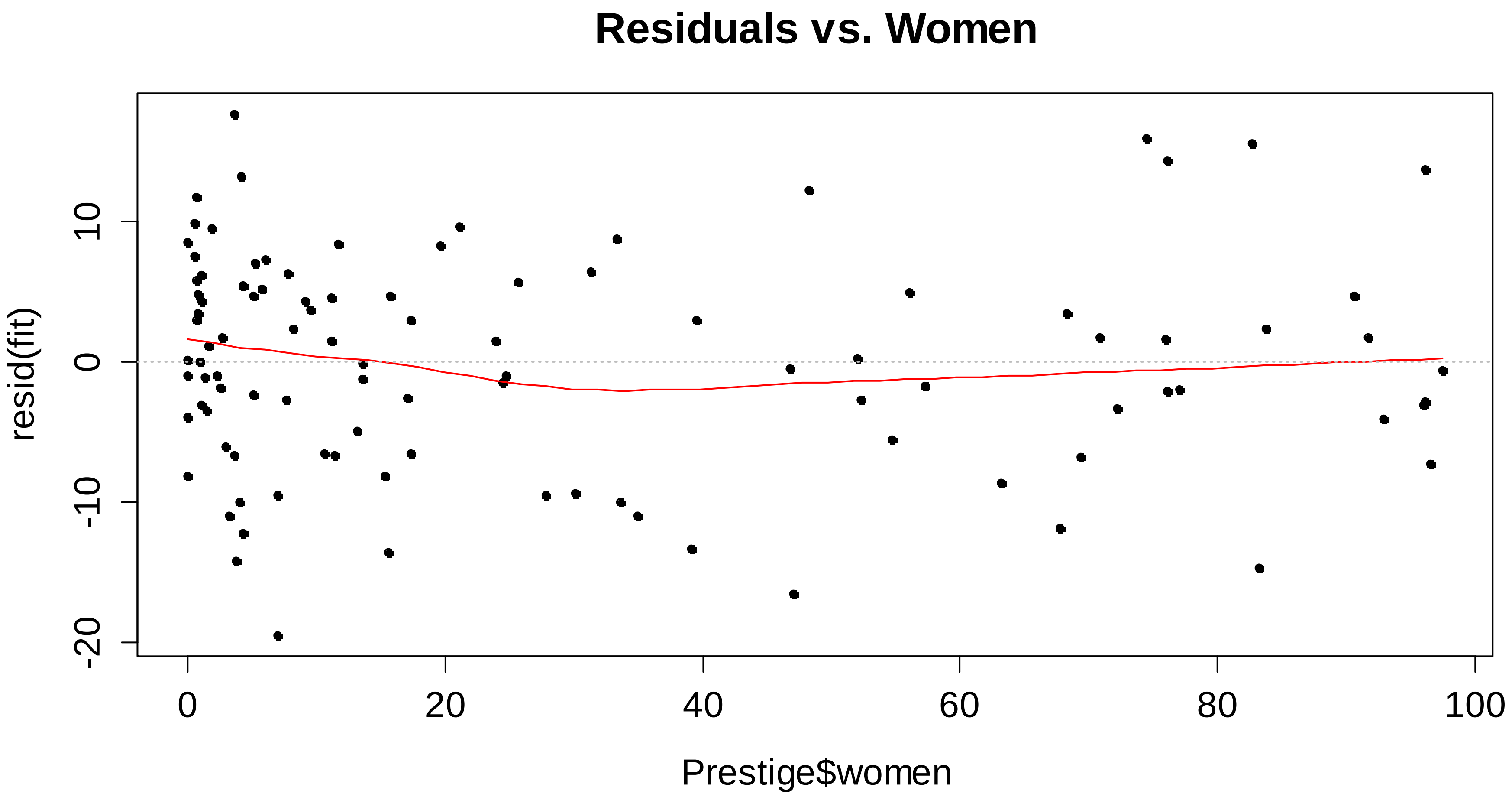
Residuals vs. Census



Residuals vs. Type



Residuals vs. Variable Women



Motivation for Partial Residual Plots

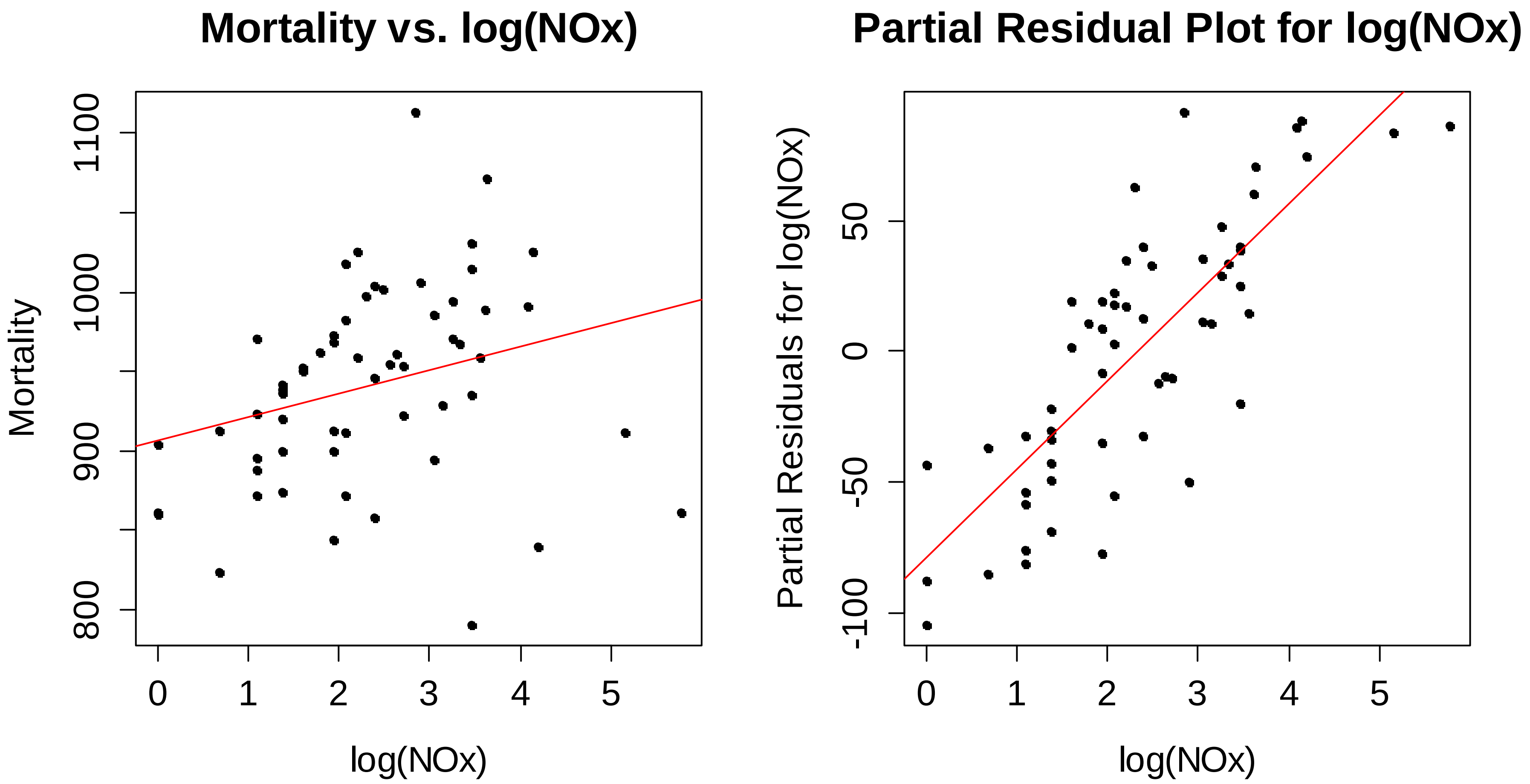
Problem:

We sometimes want to learn about the relation between one predictor and the response, and also visualize it. Is it also of importance whether that relation is linear or not.

How can we infer this?

- the plot of response y vs. predictor x_k can be deceiving!
- the reason is that the other predictors $x_1, ..., x_{k-1}, x_{k+1}, ..., x_p$ also influence the response and thus blur our impression
- thus, we require a plot which only shows the "isolated“ influence of predictor x_k on the response y.

Partial Residual Plots: First Example



Partial Residual Plots

Idea:

We remove the estimated effect of all the other predictors from the response and plot this versus the predictor x_k.

$$y-\sum_{k\neq j} x_j\hat{\beta}_j = \hat{y} + r - \sum_{k\neq j} x_j\hat{\beta}_j = x_k\hat{\beta}_k + r$$

We then plot these so-called partial residuals versus the predictor x_k. We require the relation to be linear!

Partial residual plots in R:

```
- library(car); crPlots(...)
- library(faraway); prplot(...)
- residuals(fit, type="partial")
```

Partial Residual Plots: Example

We try to predict the prestige of a number of 102 different profession with a set of 2 predictors:

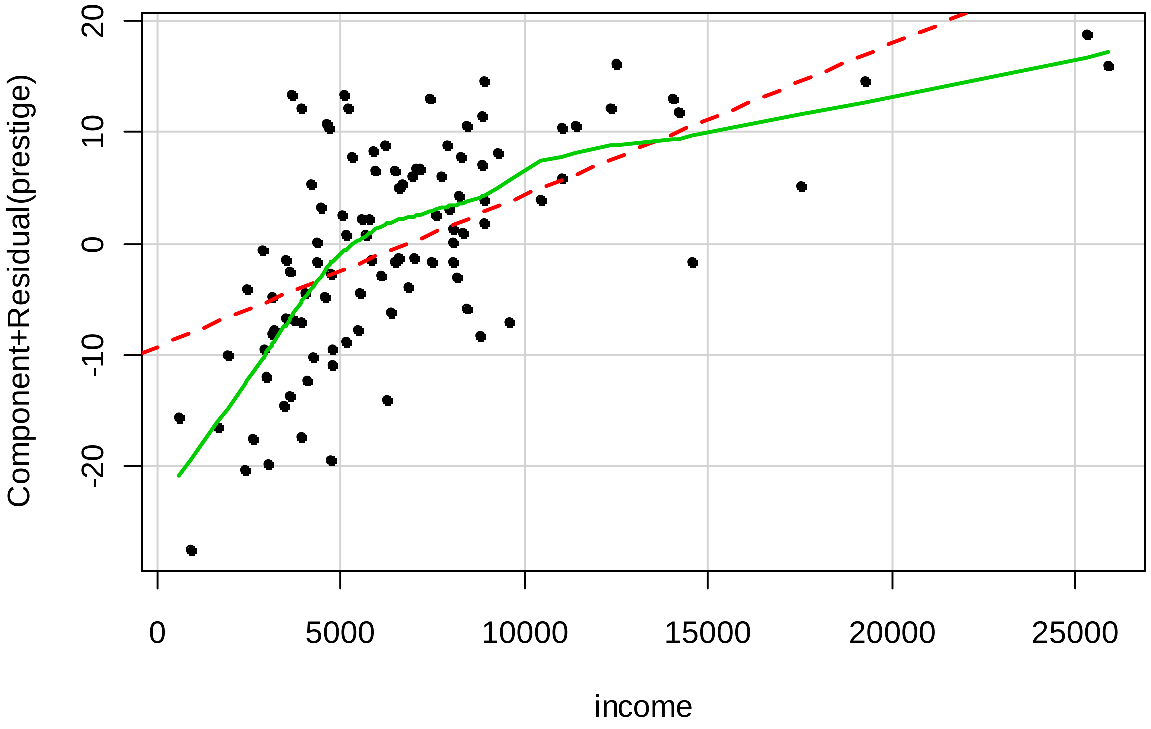
```
prestige ~ education + income
```

```
> data(Prestige)
> head(Prestige)
                    education income women prestige census type
gov.administrators      13.11  12351 11.16     68.8   1113 prof
general.managers        12.26  25879  4.02     69.1   1130 prof
accountants             12.77   9271 15.70     63.4   1171 prof
purchasing.officers     11.42   8865  9.11     56.8   1175 prof
chemists                14.62   8403 11.68     73.5   2111 prof
...
```

Applied Statistical Regression
AS 2015 – Multiple Regression

Partial Residual Plots: Example

```
> library(car); data(Prestige)
> fit <- lm(prestige ~ education + income, data=Prestige)
> crPlots(fit, layout=c(1,1))
```



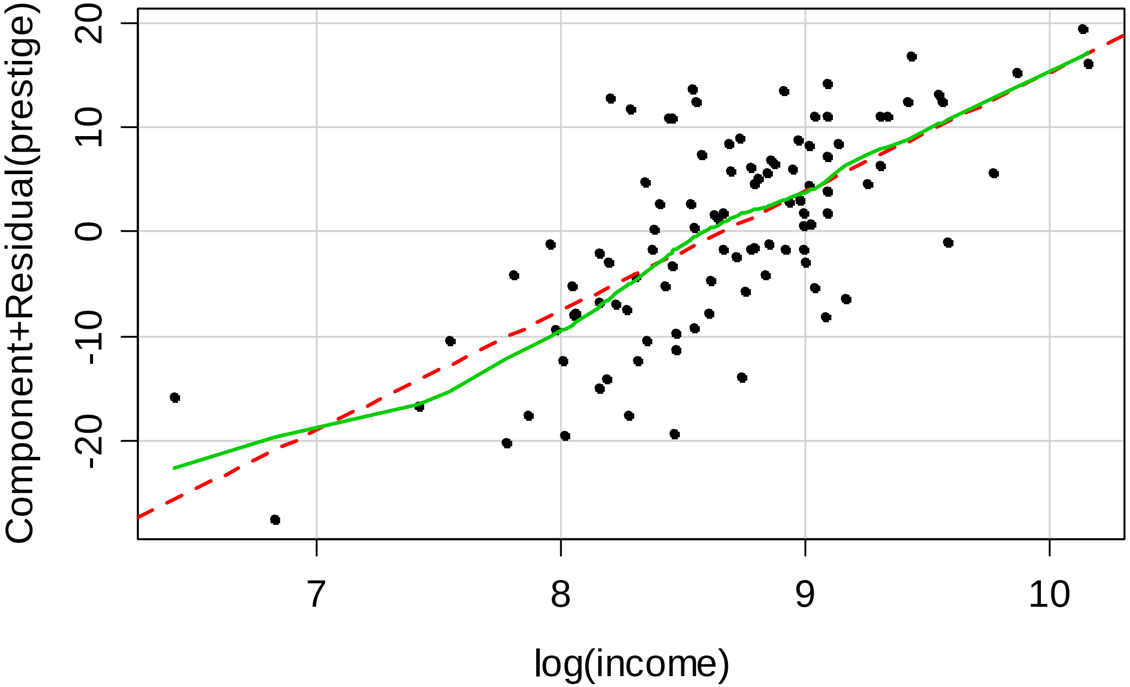
Evident non-linear influence of income on prestige.

→ not a good fit!

→ correction needed

Partial Residual Plots: Example

```
> library(car); data(Prestige)
> fit <- lm(prestige ~ education + log(income), Prestige)
> crPlots(fit, layout=c(1,1))
```



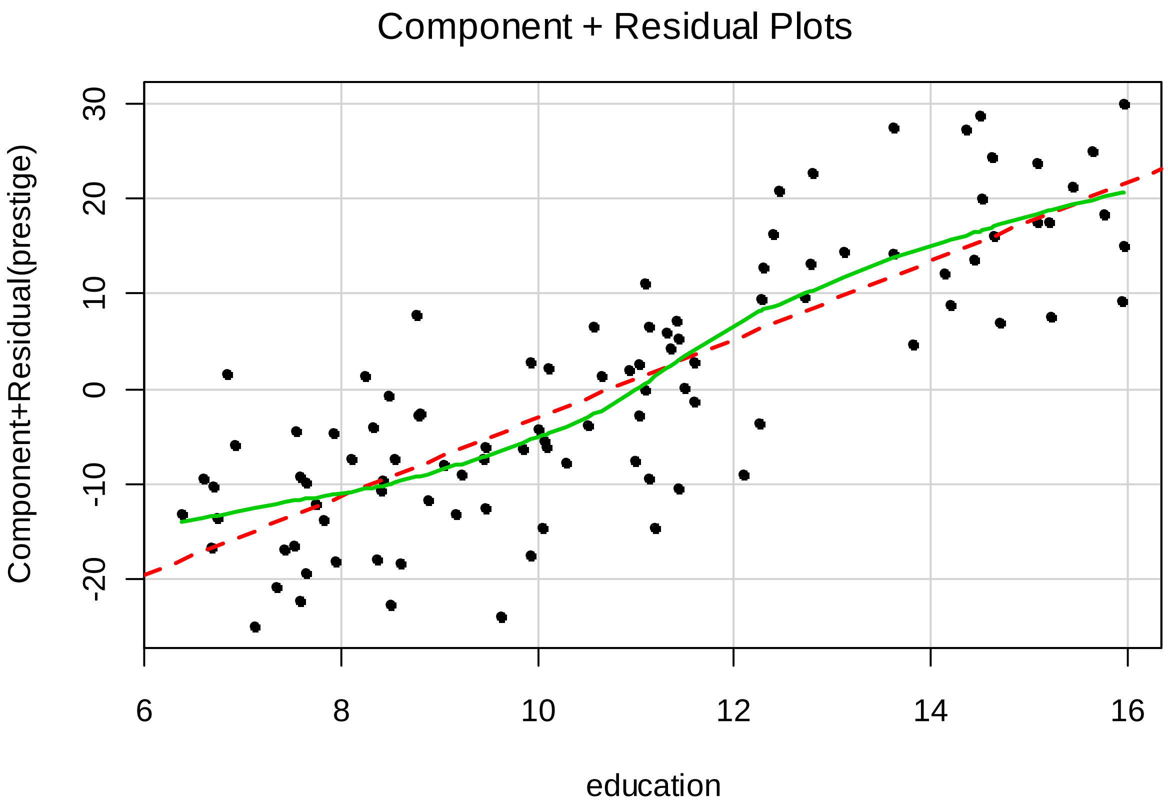
After a log-trsf of predictor 'income', things are fine

Applied Statistical Regression

AS 2015 – Multiple Regression

Partial Residual Plots: Education

```
> library(car); data(Prestige)
> fit <- lm(prestige ~ education + income, data=Prestige)
> crPlots(fit, layout=c(1,1))
```

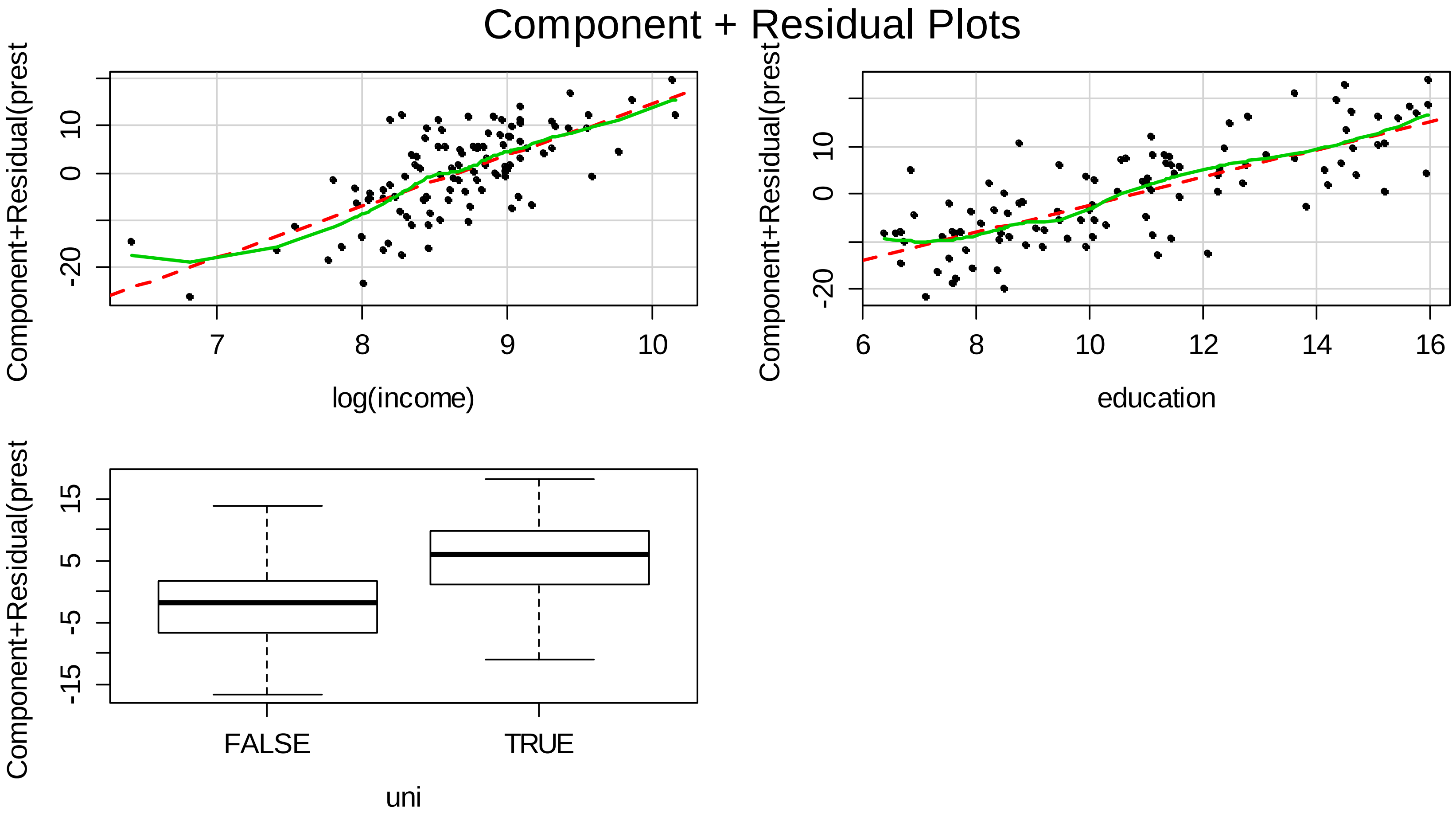


For variable education, we seem to have made a reasonable choice:

→ +/- linear relation
→ <12y vs. >12y ???

After Adding a Factor Variable…

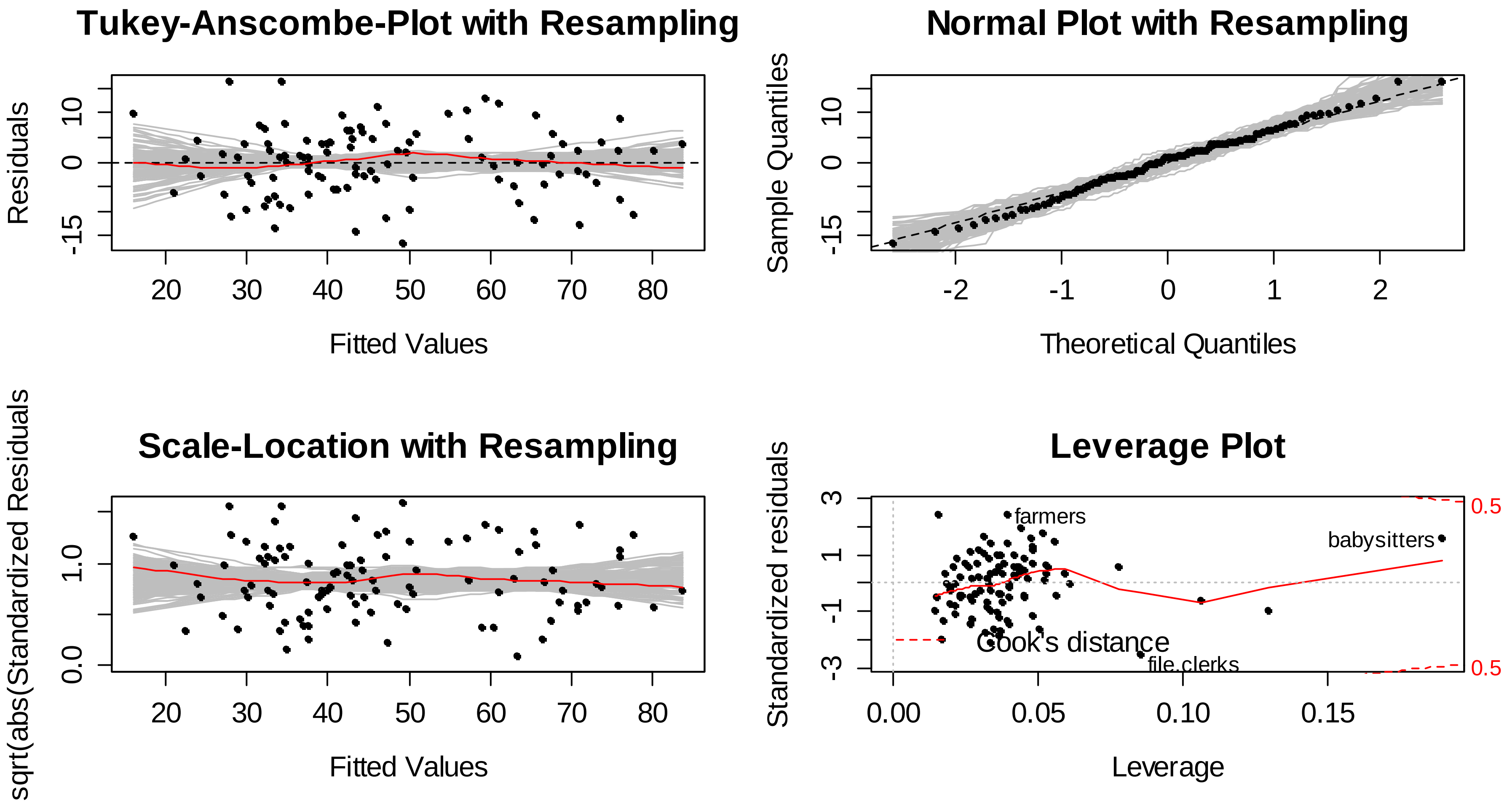
```
> fit <- lm(prestige ~ education + log(income) + uni, …)
> crPlots(fit)
```



Applied Statistical Regression

AS 2015 – Multiple Regression

After Adding a Factor Variable…



Partial Residual Plots

Summary:

Partial residual plots show the marginal relation between a predictor x_k and the response y, after/when the effect of the other variables is accounted for.

When is the plot OK?

If the red line with the actual fit from the linear model, and the green line of the smoother do not show systematic differences.

What to do if the plot is not OK?

- improve model using trsf./additional predictors/interactions.
- use Generalized Additive Models (GAM), tbd later.

Checking for Correlated Errors

Background:

For LS-fitting we require uncorrelated errors. For data which have timely or spatial structure, this condition happens to be violated quite often…

Example:

- `library(faraway); data(airquality)`
- `Ozone ~ Solar.R + Wind`
- measurements from 153 consecutive days in New York
- there is a total of 111 complete observations
- data have a timely sequence

What is the Problem?

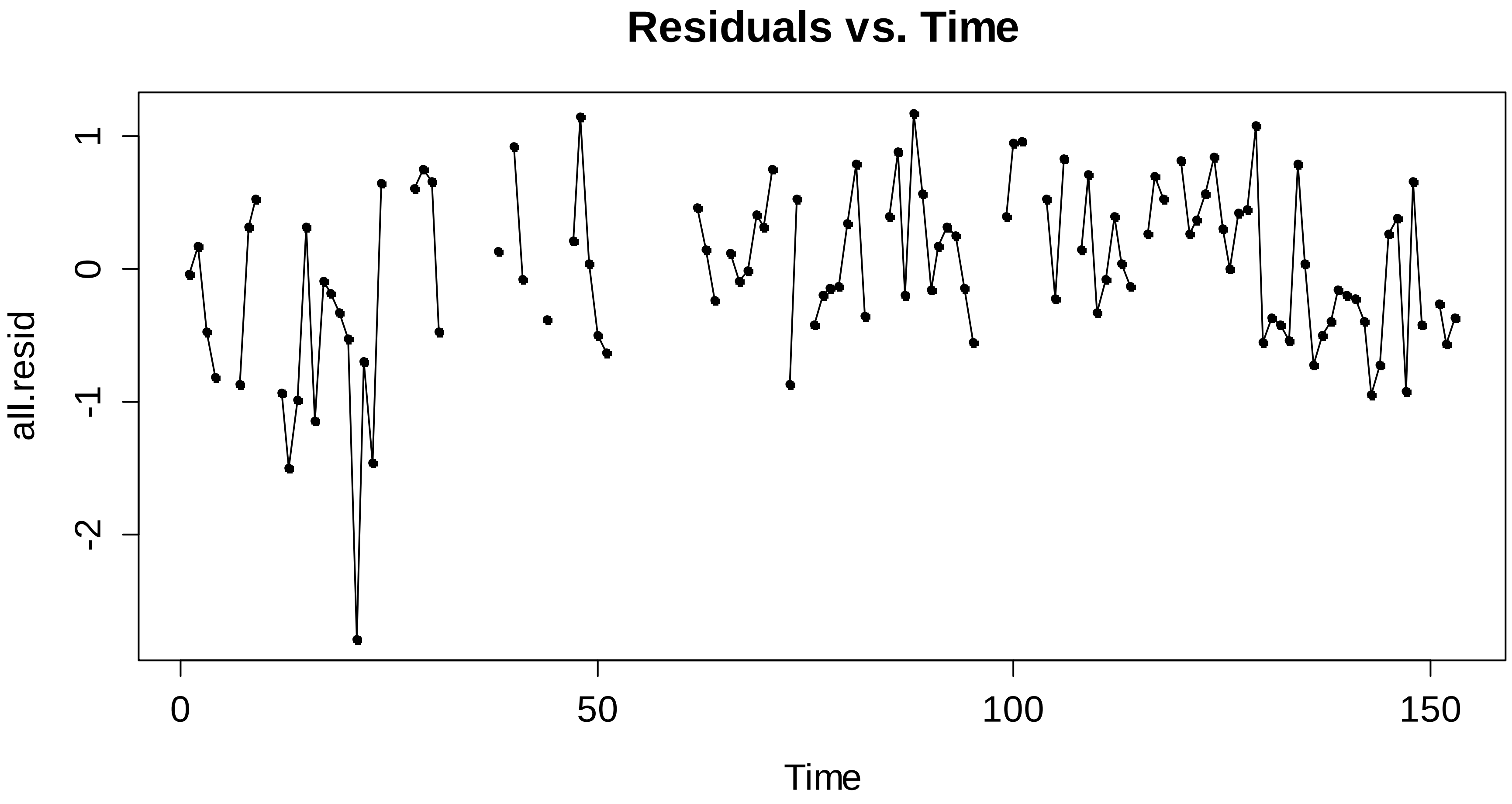
Theory: *If the errors/residuals are correlated, …*

- …the OLS procedure still results in unbiased estimates of both the regression coefficients and fitted values.

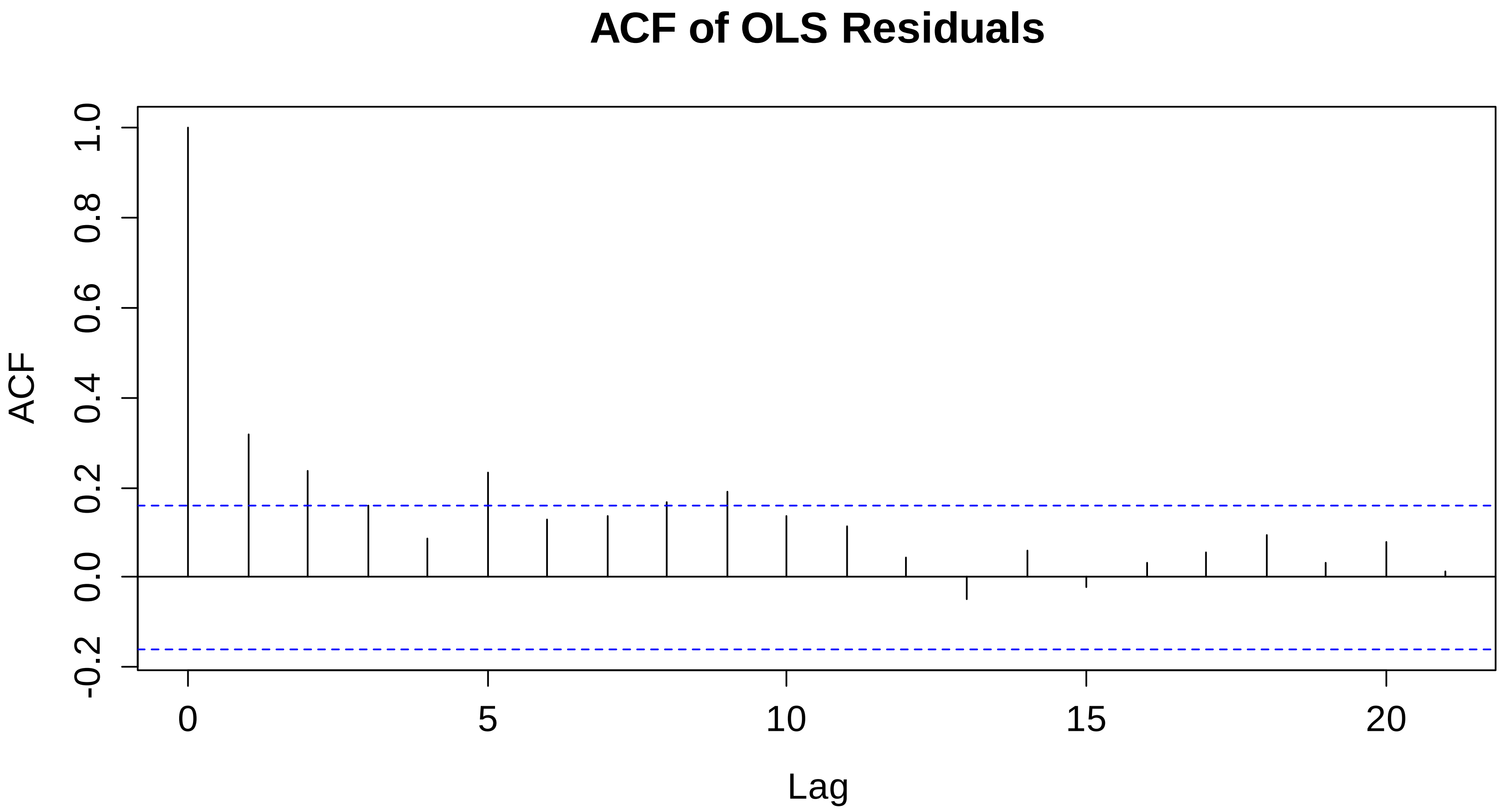
- …the OLS estimator is no longer efficient, i.e. there are alternative regression estimators that yield more precise results. These should be used!

- …the standard errors for the coefficients are biased and will inevitably lead to flawed inference results (i.e. tests and confidence intervals). The standard errors can be either too small (majority of cases), or too large.

Residuals vs. Time/Index



Autocorrelation of Residuals



Durbin-Watson-Test

The Durbin-Watson-Test checks if consecutive residuals feature any sequential correlation of simple form:

Test statistic: $DW = \dfrac{\sum_{i=2}^{n} (r_i - r_{i-1})^2}{\sum_{i=1}^{n} r_i^2}$

- under the null hypothesis "no correlation", the test statistic has a χ^2- distribution. The p-value can be computed.

- the DW-test is somewhat problematic, because it will only detect simple correlation structure. When more complex dependency exists, it has very low power.

Durbin-Watson-Test

R-Hints:

```
> library(lmtest)
> dwtest(Ozone ~ Solar.R + Wind, data=airquality)
        Durbin-Watson test
data:  Ozone ~ Solar.R + Wind
DW = 1.6127, p-value = 0.01851
alternative hypothesis: true autocorrelation is greater than 0
```

The null hypothesis is rejected for the alternative that the true autocorrelation exceeds zero. From this we conclude that the residuals are not uncorrelated here.

While the estimated coefficients and fitted values are still valid, the inference results (i.e. *p-values in the summary*) are not!!!

Residuals vs. Time/Index

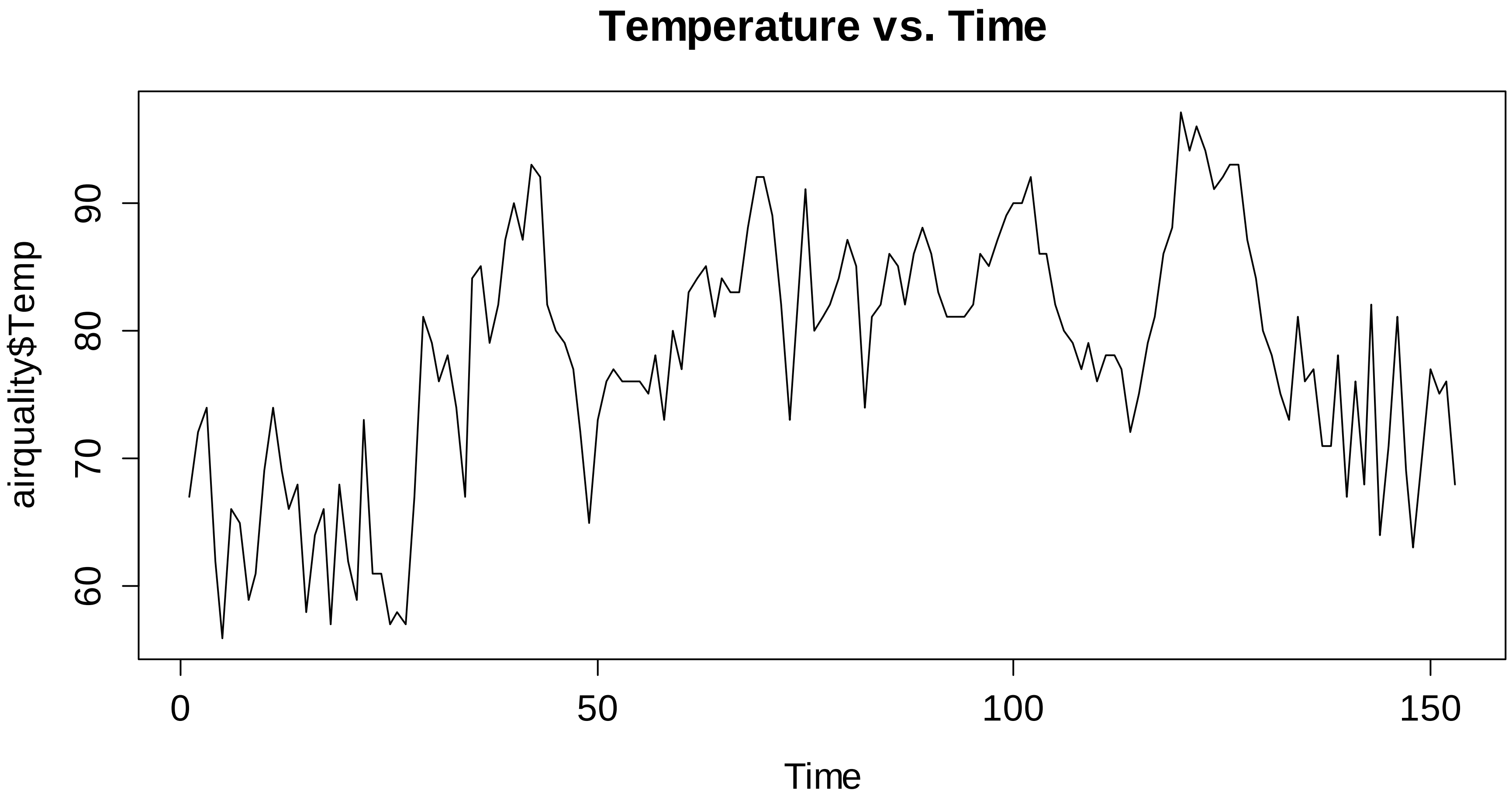
When is the plot OK?

- There is no systematic structure present
- There are no long sequences of pos./neg. residuals
- There is no back-and-forth between pos./neg. residuals
- The p-value in the Durbin-Watson test is >0.05

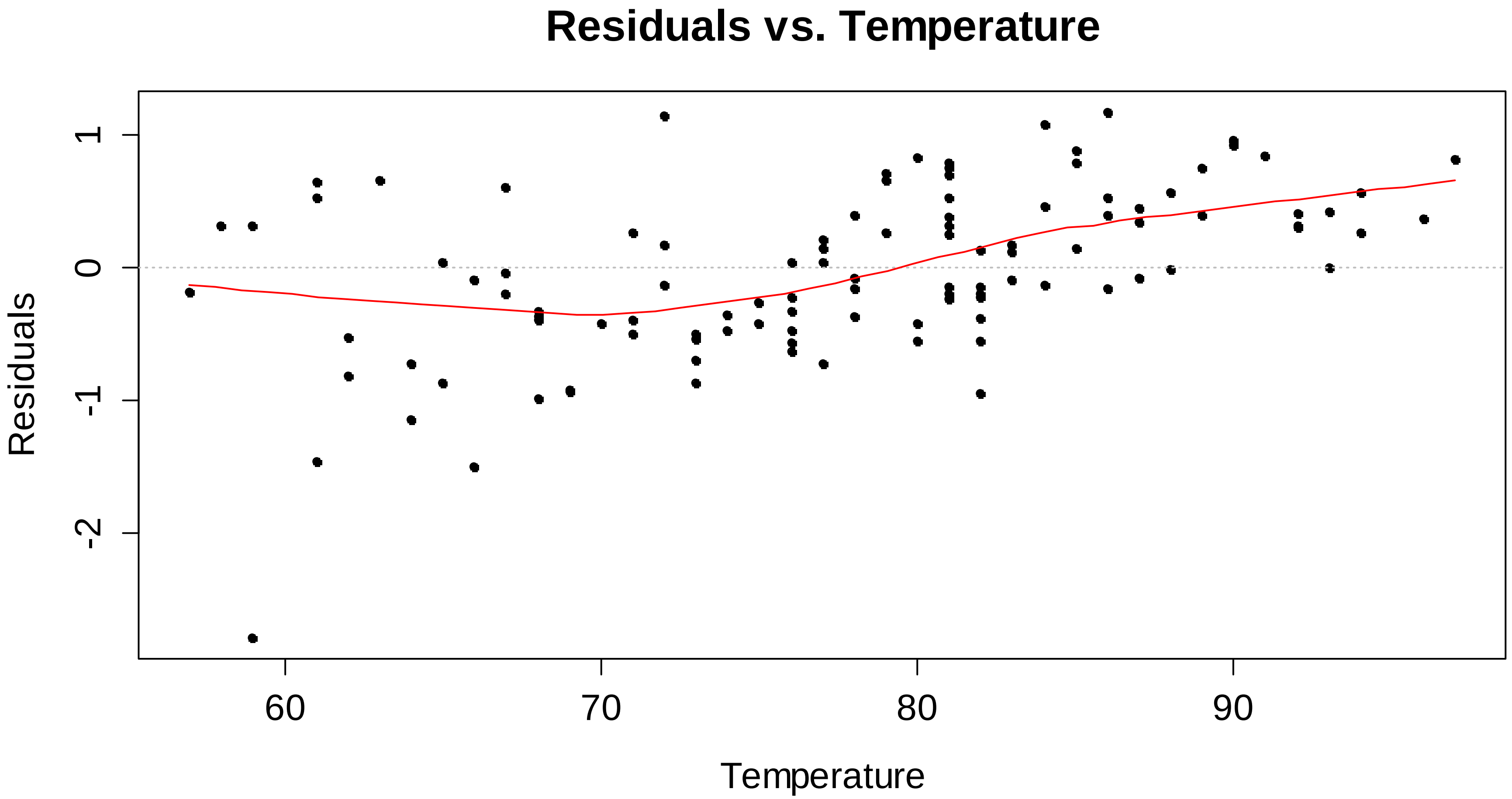
What to do if the plot is not OK?

1) Search for and add the "forgotten" predictors
2) Using the generalized least squares method (GLS)
   → to be discussed in *Applied Time Series Analysis*
3) Estimated coefficients and fitted values are not biased, but confidence intervals and tests are: be careful!

Airquality: How to Solve the Issue?



Residuals vs. Temperature



Results with the Improved Model

```
> summary(lm(log(Ozone) ~ Solar.R+Wind+Temp, data=…))
Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.2621323  0.5535669  -0.474 0.636798
Solar.R      0.0025152  0.0005567   4.518 1.62e-05 ***
Wind        -0.0615625  0.0157130  -3.918 0.000158 ***
Temp         0.0491711  0.0060875   8.077 1.07e-12 ***
---
Residual standard error: 0.5086 on 107 degrees of fr
Multiple R-squared: 0.6644, Adjusted R-squared: 0.655
F-statistic: 70.62 on 3 and 107 DF, p-value: < 2.2e-16

> dwtest(log(Ozone) ~ Solar.R + Wind + Temp, data=…)
Durbin-Watson test: log(Ozone) ~ Solar.R + Wind + Temp
DW = 1.8068, p-value = 0.1334
```

→ The residuals are no longer correlated!

Multicollinearity

We already know that a multiple linear OLS regression does not have a unique solution if its design is singular, i.e. if some of the predictors are exactly linearly dependent.

- If the rows of X are linearly dependent, then $X^T X$ does not have full rank and its inverse $(X^T X)^{-1}$ does not exist.

Multicollinearity means that there is not perfect dependence among the rows of X, but still the rows show strong correlation, *aka* collinearity.

- In these cases, there is a (technically) unique solution, but it is often highly variable and poorly suited for practice.

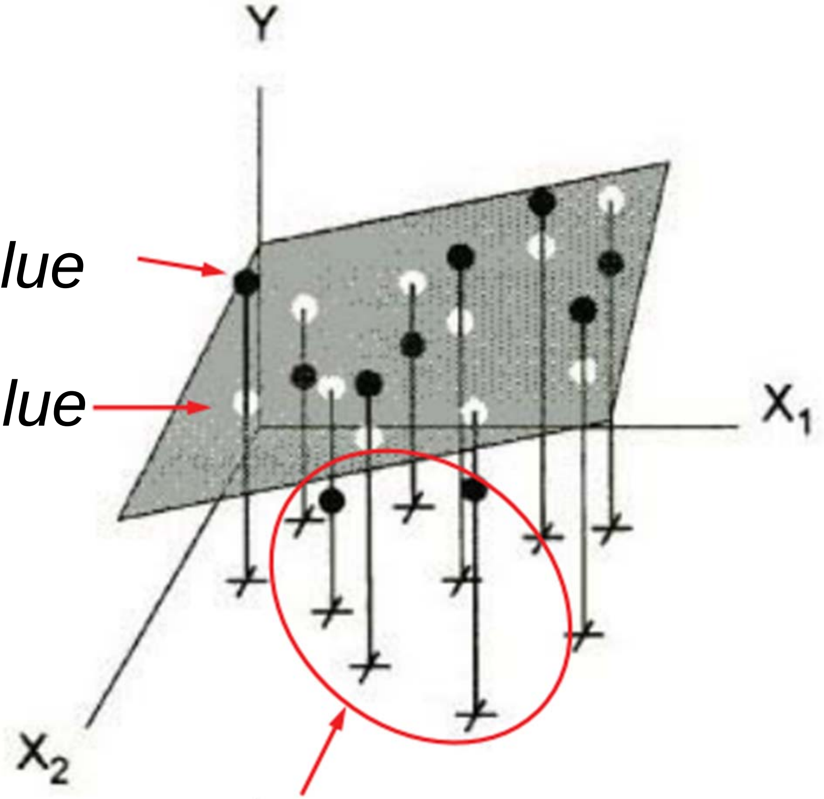
Multicollinearity – Consequences

The result of a multiple linear OLS regression with multicollinear predictors is often poos for practical use. In particular:

- The estimated coefficients feature large or even very large standard errors. Hence, they are imprecisely estimated with huge confidence intervals.
- Typical case: the global F-Test turns out to be significant, but none of the individual predictors is significant.
- The computation of the estimated coefficients is numerically problematic, if the condition number of $X^T X$ is poor.
- Extrapolation may yield extremely poor results!

Visualization of Multicollinearity

In this example, the predictors (x_1, x_2) have minimal correlation.



Observed y-value

Fitted Value

The regression plane and the coefficients are precisely determined.

Projection of data points on the (x_1, x_2)-plane.

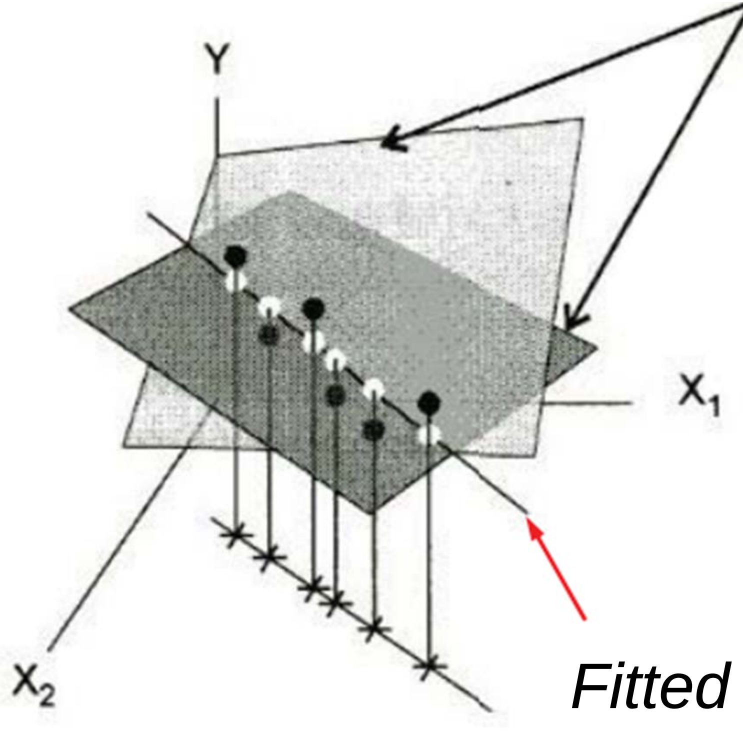
Visualization of Multicollinearity

In this example, the predictors (x_1, x_2) are linearly dependent.

The regression plane and the coefficients are not uniquely determined.

One of the predictors can be removed from the model.

Different planes



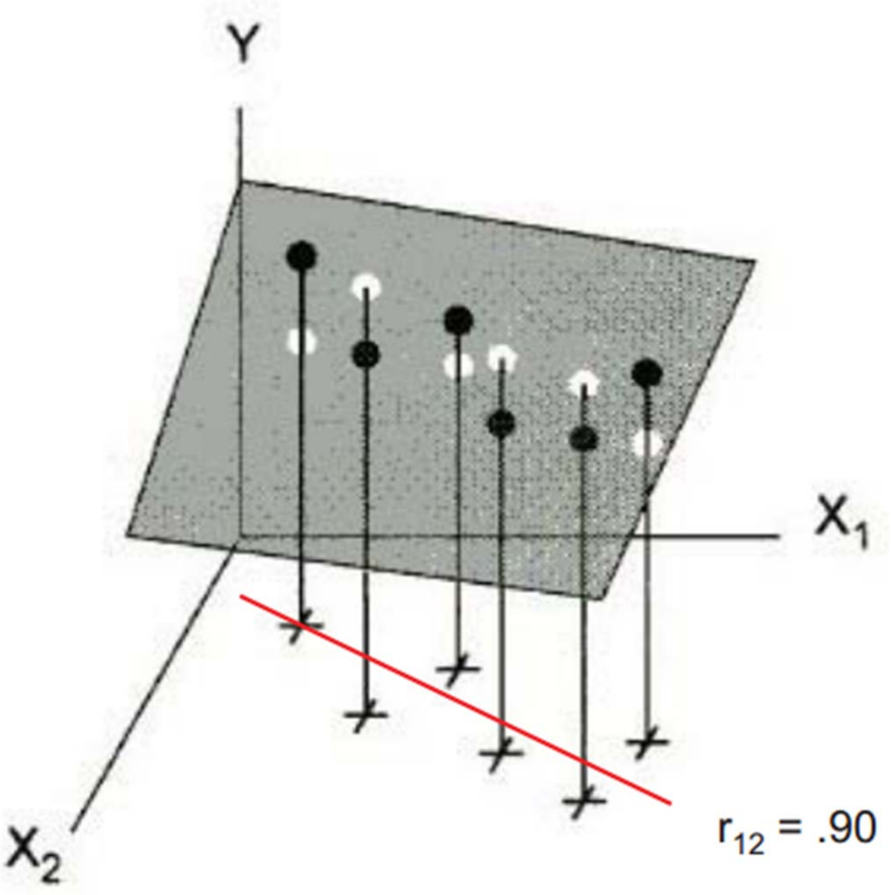
Fitted Values are identical!

Visualization of Multicollinearity

In this example, the predictors (x_1, x_2) are strongly correlated.

Regression plane and coefficients are unique but imprecise.

A small change in the data will often create a big change in the regression result.



Example

Understanding how car drivers adjust their seat would greatly help engineers to design better cars. Thus, the measured

hipcenter = horizontal distance of hips to steering wheel

and tried to explain it with several predictors, namely:

Age	age in years
Weight	weight in pounds
HtShoes, Ht, Seated	height w/o, w/ shoes, seated height
Arm, Thigh, Leg	arm, thigh and leg length

We first fit a model with all these (correlated!) predictors

Example: Fit with All Predictors

```
> library(faraway); data(seatpos)
> summary(lm(hipcenter~., data=seatpos))
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 436.43213  166.57162   2.620   0.0138 *
Age           0.77572    0.57033   1.360   0.1843
Weight        0.02631    0.33097   0.080   0.9372
HtShoes      -2.69241    9.75304  -0.276   0.7845
Ht            0.60134   10.12987   0.059   0.9531
Seated        0.53375    3.76189   0.142   0.8882
Arm          -1.32807    3.90020  -0.341   0.7359
Thigh        -1.14312    2.66002  -0.430   0.6706
Leg          -6.43905    4.71386  -1.366   0.1824

Residual standard error: 37.72 on 29 degrees of freedom
Multiple R-squared: 0.6866, Adjusted R-squared: 0.6001
F-statistic:  7.94 on 8 and 29 DF,  p-value: 1.306e-05
```

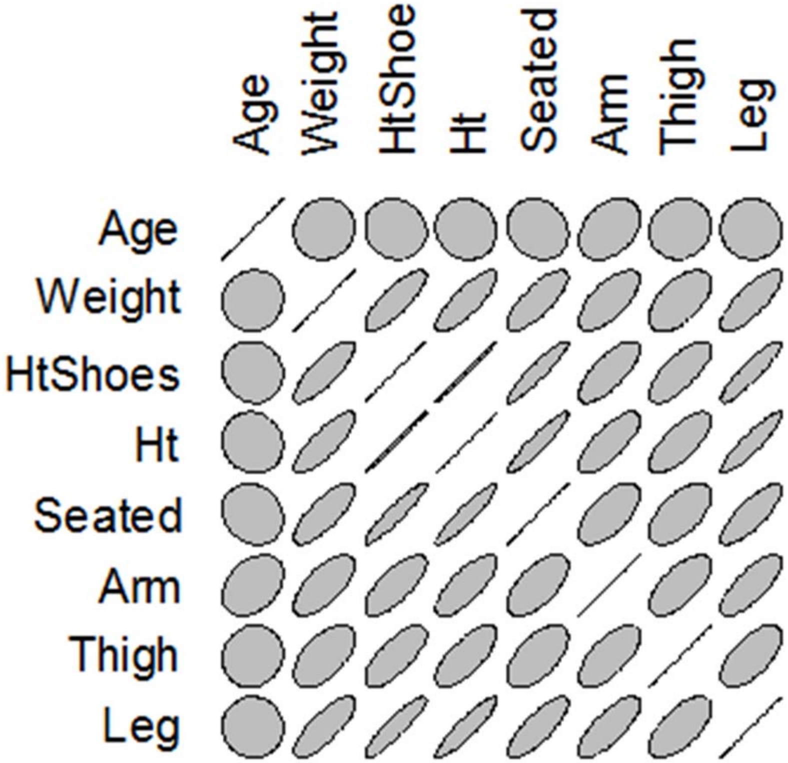
How to Identify Multicollinearity?

A simple option consists of analyzing all pairwise correlation coefficients among the predictors variables:

```
> library(ellipse)
> plotcorr(cor(dat))
```

Note: this will not identify all situations where there is multicollinearity!!!

Reason: multicollinearity is not always a pairwise phenomenon!!!



Variance Inflation Factor

The variance of a regression coefficient can be rewritten as:

$$Var(\hat{\beta}_k) = \sigma_E^2 \cdot \frac{1}{1-R_k^2} \cdot \frac{1}{\sum_{i=1}^{n}(x_{ik}-\overline{x}_k)^2}$$

Error Variance *VIF* *Design-Component*

$VIF = 1/(1-R_k^2)$ is obtained by determining R-Squared in a regression, where x_k is the response variable and all other predictors maintain their role.

The higher the collinearity between x_k and other predictors, the higher are R_k^2 and hence also the $VIF = 1/(1-R_k^2)$.

Determination of VIFs

```
> library(faraway)
> vif(fit)
       Age     Weight    HtShoes         Ht
  1.997931   3.647030 307.429378 333.137832
    Seated        Arm      Thigh        Leg
  8.951054   4.496368   2.762886   6.694291
```

A $VIF \geq 5$ corresponds to a $R_k^2 \geq 0.8$ and has to be seen as a critical multicollinearity. $VIF \geq 10$ means that $R_k^2 \geq 0.9$ and hence that dangerous multicollinearity is present.

In our example, some particular VIFs are even higher. The standard error of `Ht` is inflated by about factor 18.

Dealing with Multicollinearity

- In many cases, multicollinearity among the predictors is not so simple to cure and well-thought-out action is needed.

- The simplest option is the so-called *amputation*. It means that among all collinear predictors, all except one will be discarded.

- In our example, amputation would reduce to the predictors `Age`, `Weight`, `Ht`. However, in this example this is not really satisfying:

  - we are discarding valuable information
  - leg and arm length with respect to height are important!

- Rather than to ampute, we can create new variables!

Amputation

```
> fit <- lm(hipcenter ~ Age + Weight + Ht, data=seatpos)
> summary(fit)
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 528.297729 135.312947   3.904 0.000426 ***
Age           0.519504   0.408039   1.273 0.211593
Weight        0.004271   0.311720   0.014 0.989149
Ht           -4.211905   0.999056  -4.216 0.000174 ***
---
Multiple R-Squared: 0.6562 Adjusted R-Squared: 0.6258
```

R-Squared values were 0.6866 and 0.6001 before amputation.

```
> vif(fit)
     Age   Weight       Ht
1.093018 3.457681 3.463303
```

Example: Generating New Variables

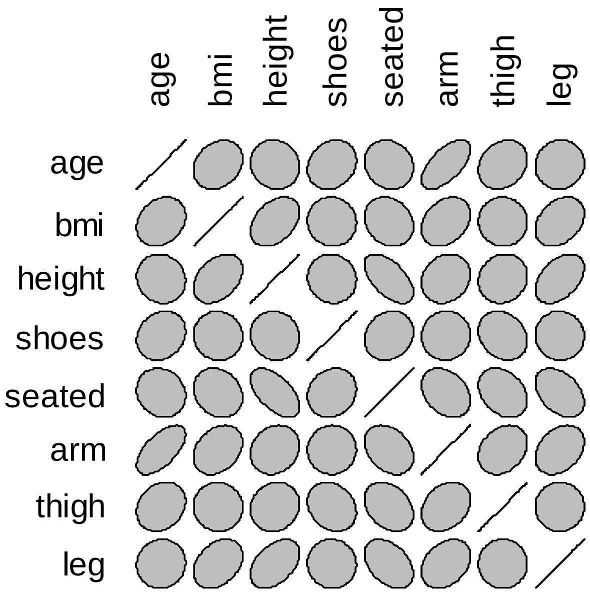
The body height is certainly a key predictors when it comes to the position of the driver seat. We leave this as it was, and change several of the other predictors:

```
age    <- Age
bmi    <- (Weight*0.454)/(Ht/100)^2
shoes  <- HtShoes-Ht
seated <- Seated/Ht
arm    <- Arm/Ht
thigh  <- Thigh/Ht
leg    <- Leg/Ht
```

Does this solve the correlation problem...?

Example: New Correlation Matrix

```
> vif(fit00)
     age      bmi   height    shoes   seated
1.994473 1.408055 1.968447 1.155285 1.851884
     arm    thigh      leg
2.044727 1.284893 1.480397
```



Multicollinearity does not seem to be an issue anymore. But we managend to keep all variables.

Example: Summary with New Predictors

```
> summary(lm(hipc~., data=new.seatpos))

             Estimate Std. Error t value Pr(>|t|)
(Intercept) -632.0063   490.0451  -1.290    0.207
age           -0.7402     0.5697  -1.299    0.204
bmi           -0.4234     2.2622  -0.187    0.853
height         3.6521     0.7785   4.691 5.98e-05 ***
shoes          2.6964     9.8030   0.275    0.785
seated      -171.9495   631.3719  -0.272    0.787
arm          180.7123   655.9536   0.275    0.785
thigh        141.2007   443.8337   0.318    0.753
leg         1090.0111   806.1577   1.352    0.187

Residual standard error: 37.71 on 29 degrees of freedom
Multiple R-squared: 0.6867, Adjusted R-squared: 0.6002
F-statistic: 7.944 on 8 and 29 DF,  p-value: 1.3e-05
```

Ridge Regression

A computational procedure that can deal with collinearity.

Using a penalization approach, shrinkage is applied to the coefficients. They will be biased, but less variable:

Ridge estimator: $\hat{\beta} = (X^T X + \lambda I)^{-1} X^T y$

Alternative view: $\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} r_i^2 + \lambda \sum_{j=1}^{p} \beta_j^2$

λ is the penalty parameter. It controls the amount of shrinkage. The bigger it is, the smaller the the coefficients become, and the better the conditions of matrix $(X^T X + \lambda I)$ is. This facilitates its inversion, i.e. multicollinearity is mitigated.

Ridge Regression: Fitting

```
> library(MASS)
> my.sp$hipcenter <- scale(my.sp$hipcenter, scale=FALSE)
> ll     <- seq(0,100,by=0.1)
> fit.rr <- lm.ridge(hipcenter ~ ., lambda=ll, data=sp)
```

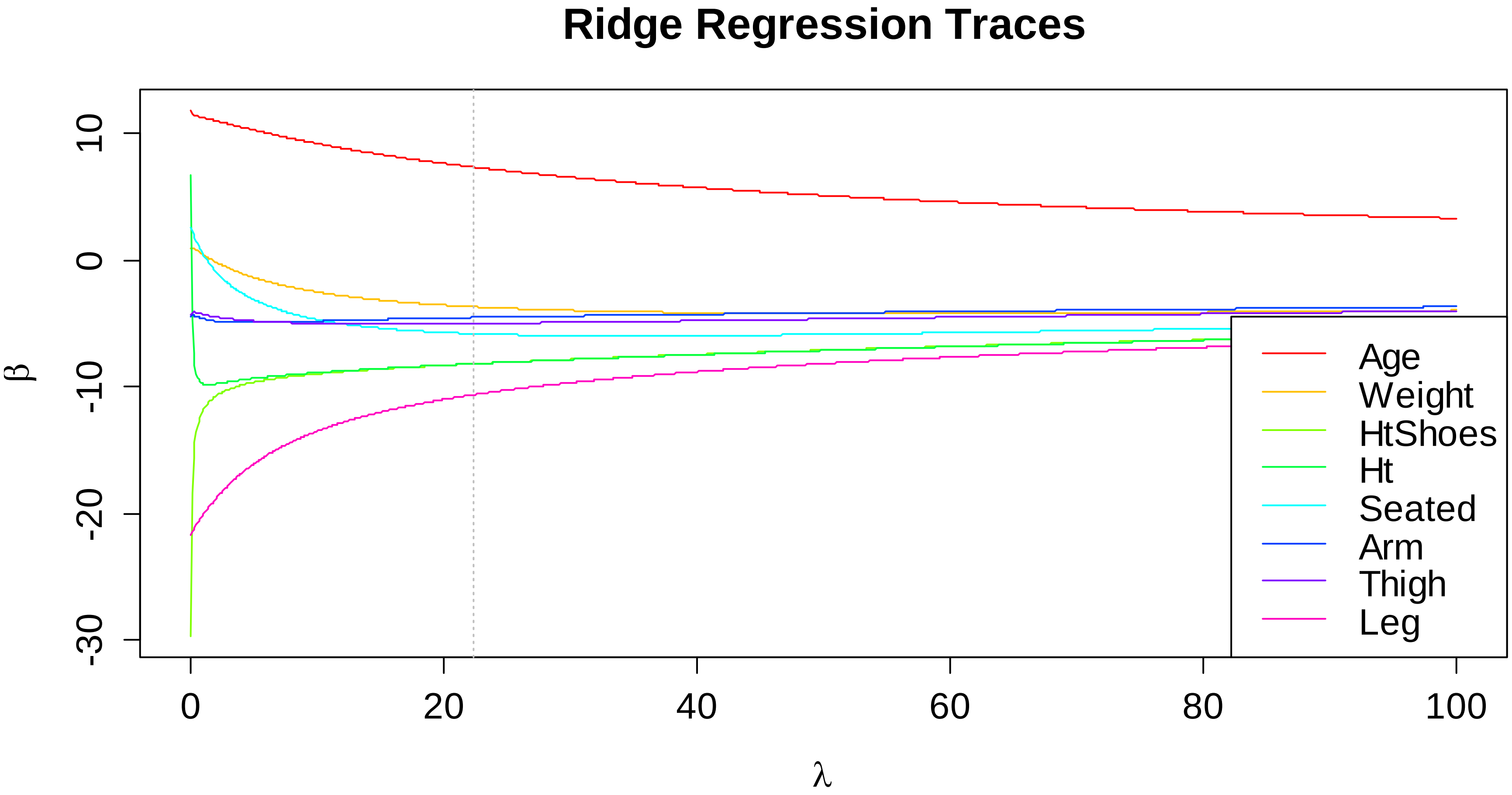
- The response variable must be centered

- The choice of the penalty parameter λ is ambiguous. The R function allows for a simultaneous fit with several λ, so that multiple solutions can be compared.

- R function `select(fit.rr)` presents several approaches that allow for determining the optimal penalty parameter. Even a visualization is possible.

Ridge Regression: Choice of λ

```
> matplot(fit.rr$lambda, t(fit.rr$coef), col=…)
```



Variable Selection: Why?

We want to keep a model small, because of

1) **Simplicity**
   → *among several explanations, the simplest is the best*

2) **Noise Reduction**
   → *unnecessary predictors leads to less accuracy*

3) **Collinearity**
   → *removing excess predictors facilitates interpretation*

4) **Prediction**
   → *less variables, less effort for data collection*

Method or Process?

- **Variable selection is not a method!** The search for the best predictor set is an iterative process. It also involves *estimation*, *inference* and *model diagnostics*.
- For example, outliers and influential data points will not only change a particular model – they can even have an impact on the model we select. Also variable transformations will have an impact on the model that is selected.
- Some iteration and experimentation is often necessary for variable selection. *The ultimate aim is finding a model that is smaller, but as good or even better than the original one.*

Example: Mortality Data

```
> summary(fit.orig)
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 1496.4915   572.7205   2.613  0.01224 *
JanTemp       -2.4479     0.8808  -2.779  0.00798 **
...
Dens          11.9490    16.1836   0.738  0.46423
NonWhite     326.6757    62.9092   5.193 5.09e-06 ***
WhiteCollar -146.3477   112.5510  -1.300  0.20028
...
---
Residual standard error: 34.23 on 44 degrees of freedom
Multiple R-squared: 0.7719, Adjusted R-squared: 0.6994
F-statistic: 10.64 on 14 and 44 DF,  p-value: 6.508e-10
```

Note: due to space constraints, this is only part of the output.

Backward Elimination with p-Values

Aim: Reducing the regression model such that the remaining predictors show a significant relation to the response.

How: We start with the full model and then exclude the least significant predictor in a step-by-step manner, as long as its p-value is greater than $\alpha_{crit} = 0.05$.

In R:

```
> fit <- update(fit, . ~ . - RelHum)
> summary(fit)
```

→ *Re-fit the model after each exclusion!*

→ *Wording:* **Backward Elimination with p-Values**

→ For prediction, one also uses $\alpha_{crit} = 0.10 / 0.15 / 0.20$

Example: Final Result

```
> ft09 <- update(ft08, .~.-WhiteCollar); summary(ft09)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 992.2069    79.6994  12.449  < 2e-16 ***
JanTemp      -2.1304     0.5017  -4.246 8.80e-05 ***
Rain          1.8122     0.5066   3.577 0.000752 ***
Educ        -16.4207     6.1202  -2.683 0.009710 **
NonWhite    268.2564    38.8832   6.899 6.56e-09 ***
NOx          18.3230     4.3960   4.168 0.000114 ***
---
Residual standard error: 33.47 on 53 degrees of freedom
Multiple R-squared: 0.7373, Adjusted R-squared: 0.7125
F-statistic: 29.75 on 5 and 53 DF,  p-value: 2.931e-14
```

→ 9 predictors are eliminated, 5 remain in the final model.

Interpretation of the Result

- The remaining predictors are now “more significant” than before. This is almost always the case. Do not overestimate the importance of these predictors!

- Collinearity among the predictors is usually at the root of this observation. The predictive power is first spread out among several predictors, then it becomes concentrated.

- **Important**: the removed variables can still be related to the response. If we run a simple linear regression, they can even be significant. In the multiple linear model however, there are other, better, more informative predictors.

Alternatives for Variable Selection

Backward elimination that is based on p-values requires laborious handwork (*in R*) and has a few disadvantages...

- When the principal goal is prediction, then the resulting models are often too small, i.e. there are bigger models which yield a more accurate prognosis.
- From a (theoretical) mathematical viewpoint variable selection via the AIC/BIC criterions is more suitable.
- In a step-by-step backward elimination, the best model is often missed. Evaluating more models can be very beneficial for finding *the best one*...

The AIC/BIC Criteria

→ *Gauging Goodness-of-Fit vs. The Number of Predictors*

AIC-Criterion:

$$\begin{aligned} AIC &= -2\max(\log likelihood) + 2p \\ &= const + n\log(RSS/n) + 2p \end{aligned}$$

BIC-Criterion:

$$\begin{aligned} BIC &= -2\max(\log likelihood) + p\log n \\ &= const + n\log(RSS/n) + p\log n \end{aligned}$$

AIC/BIC allow for comparison of models that are not hierarchical. But they need to be fitted on exactly the same data points.

Backward Elimination with AIC/BIC

Aim: Reducing the regression model such that the remaining predictors are *necessary* for describing the response.

How: We start with the full model and then in a step-by-step manner exclude the predictor that leads to the biggest improvement in AIC/BIC.

In R:

```
> fit <- lm(Mortality ~ JanTemp + …, data=apm)
> fit.aic <- step(fit, dir="backward", k=2)
> fit.bic <- step(fit, dir="backward", k=log(59))
```

→ *The selection stops when AIC/BIC cannot be improved anymore. Predictors do not need to be significant.*

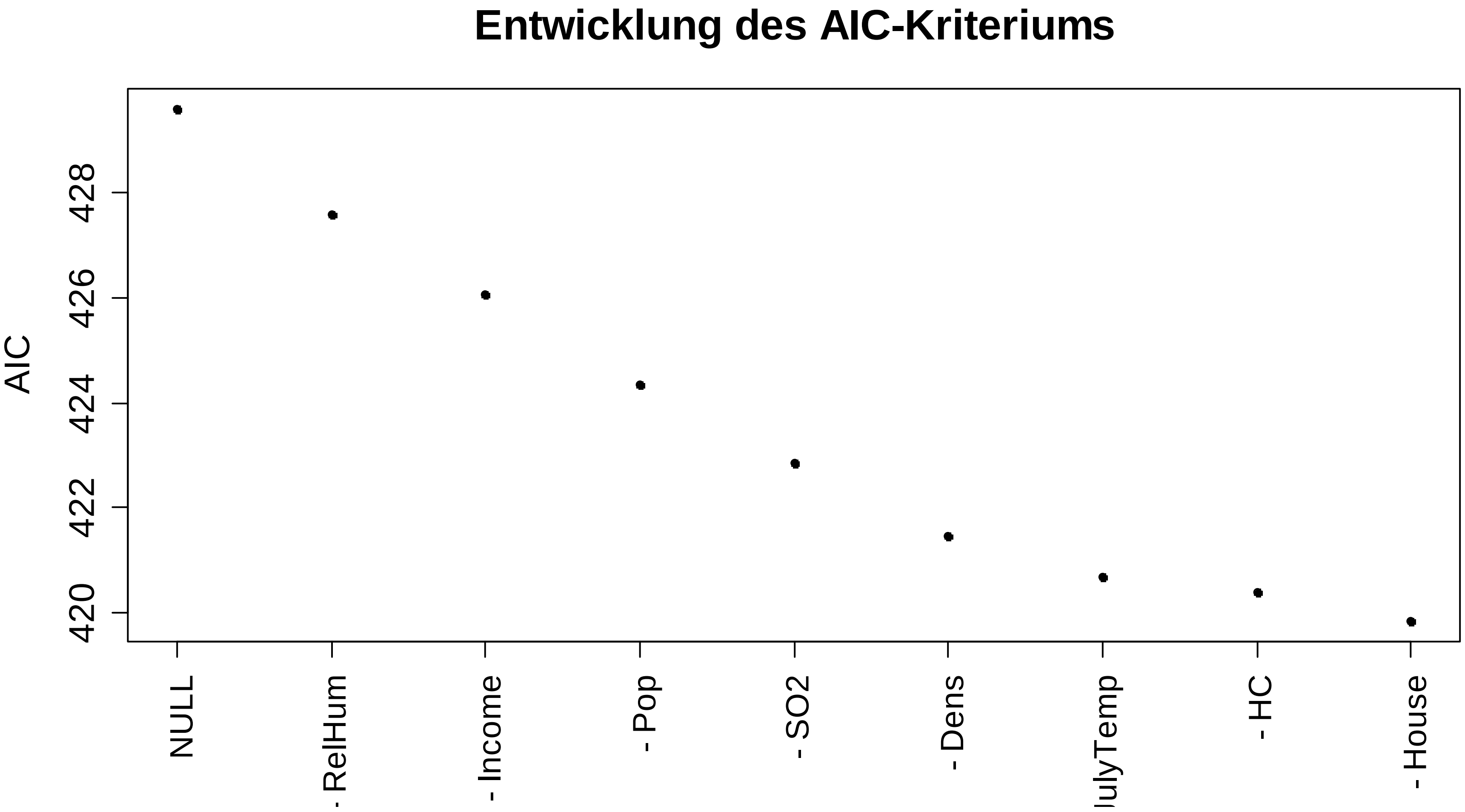
Example: Models with AIC/BIC

```
AIC:         Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1035.5384    85.1924  12.155  < 2e-16 ***
JanTemp       -2.0188     0.5043  -4.003 0.000200 ***
Rain           1.9637     0.5146   3.816 0.000363 ***
Educ         -11.7708     6.9613  -1.691 0.096842 .  
NonWhite     261.5379    38.8830   6.726 1.35e-08 ***
WhiteCollar -139.2913   102.0379  -1.365 0.178101    
NOx           19.4440     4.4372   4.382 5.73e-05 ***
```

```
BIC:        Estimate Std. Error t value Pr(>|t|)    
(Intercept) 992.2069    79.6994  12.449  < 2e-16 ***
JanTemp      -2.1304     0.5017  -4.246 8.80e-05 ***
Rain          1.8122     0.5066   3.577 0.000752 ***
Educ        -16.4207     6.1202  -2.683 0.009710 ** 
NonWhite    268.2564    38.8832   6.899 6.56e-09 ***
NOx          18.3230     4.3960   4.168 0.000114 ***
```

Visualization of Variable Selection

```
> plot(fit.aic$anova$AIC, ...)
```



AIC or BIC?

Usually, both criteria lead to similar models. BIC penalizes bigger models harder, with factor $\log n$ instead of factor 2.

→ *"BIC models" tend to be smaller than "AIC models"!*

Rule of the thumb for criterion choice:

- **BIC** is used when we are after a small model that is easy to interpret, i.e. in cases where understanding the predictor-response relation is the primary goal.
- **AIC** is used when the principal aim is the prediction of future observations. In these cases, small out-of-sample error is key, but neither the number or meaning of the predictors.

Forward Selection

1) Start with an empty model (*intercept only, no predictors*)
2) In a step-by-step manner, the best suited predictor is added, i.e. the one which leads to the lowest AIC/BIC value.
3) Continue adding predictors to the model until the AIC/BIC value cannot be improved anymore.

```
> f.null <- lm(Mortality ~ 1, data=apm)
> myscop <- list(upper=formula(Mortality ~ JanTemp+…))
> f.forw <- step(f.null, scope=myscop, dir="forward")
```

→ Forward Selection is used with big datasets, where there are too many predictors for the number of observations.

Stepwise Model Search

- This is an alternation of forward and backward steps. We can either start from the full model (1. step backwards) or from the empty model (1. step forward).

- In each forward step, all predictors can be added, also these that were excluded before. In each backward step, any of the predictors can be kicked out of the model (again).

→ Similar to Backward Elimination resp. Forward Search

→ Not much more time consuming, but more exhaustive

→ Default method in R function `step()`

→ Recommended!

Stepwise Model Search in R

Starting from the empty model:

```
> f.null <- lm(Mortality ~ 1, data=apm)
> f.full <- lm(Mortality ~ JanTemp + …, data=apm)
> fit    <- step(f.null, scope=list(upper=f.full))
```

Starting from the full model:

```
> f.full <- lm(Mortality ~ JanTemp + …, data=apm)
> fit  <- step(f.full, scope=list(lower=f.null))
```

Note:

Argument `scope=...` allows specifying arbitrary minimal and maximal models for both cases. Then some predictors can be added or be removed from the model.

Alternative Search Heuristics

All Subsets Regression

- When m predictors are present, there are in fact 2^m different models that could be tried for finding the best one.

- In cases where m is small (i.e. $m \approx 10-20$) all submodels (up to a certain size) can be tried and evaluated by computing the AIC/BIC criterion.

→ Complete search, but enormous computing time needed

→ Yields a good solution, but not the causal model either

→ Recommended for small dataset where it is feasible

→ R implementation with function `leaps()`

All Subsets Regression in R

R commands:

```
> library(leaps)
> out <- regsubsets(Mortality~., nbest=1,
                    data=mortality, nvmax=14)
> summary(out)
> plot(out)
```

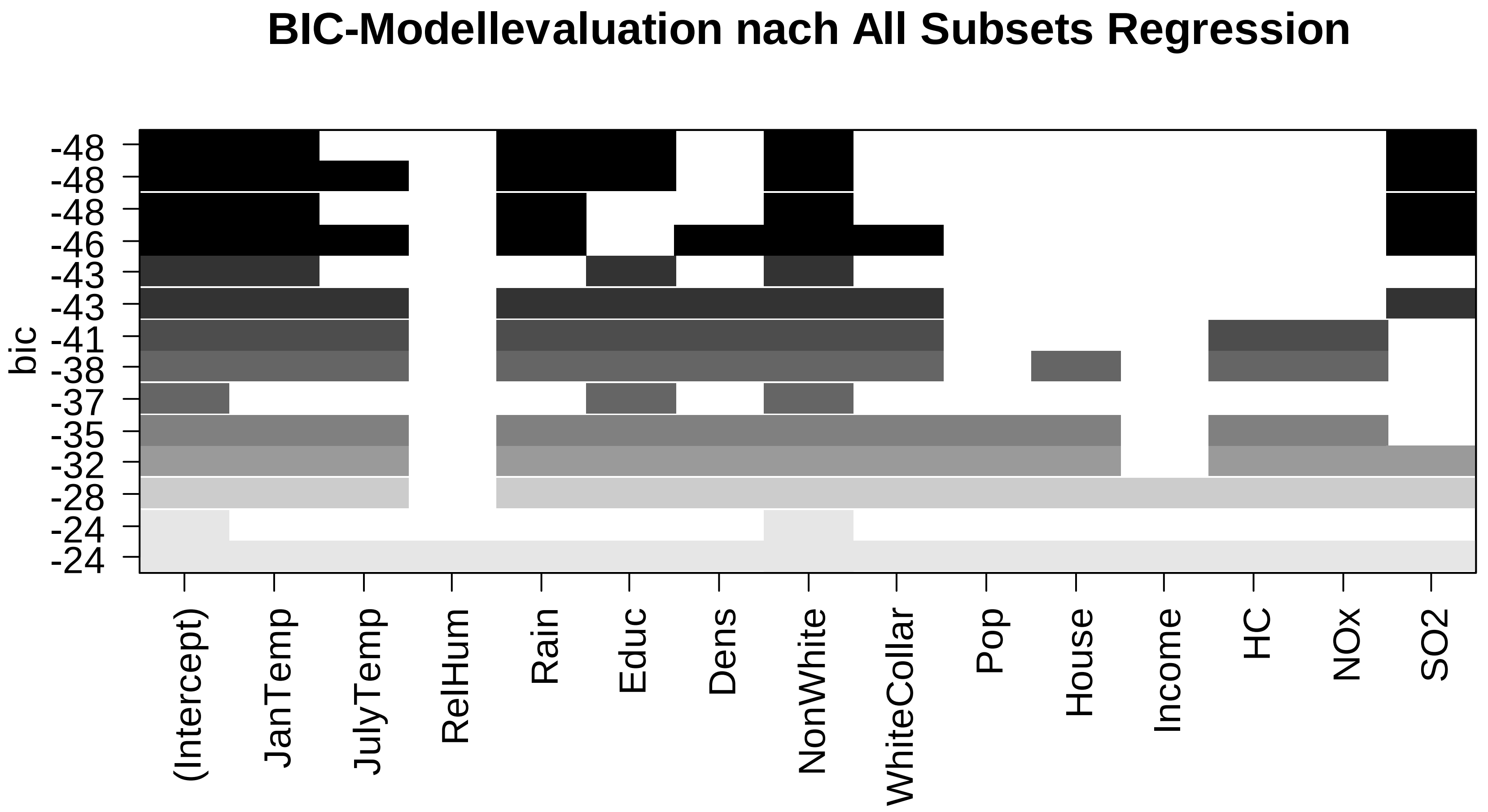
Note:

The procedure starts with the empty model and for each number of predictors, identifies the `nbest=1` models. By typing `~.` in the formula, all predictors are allowed. The maximum model size that is search can be determined with `nvmax=14`.

Applied Statistical Regression

AS 2015 – Multiple Regression

Visualization of All Subsets Selection



Hierarchy in Model Selection

Models with Polynomial Terms and/or Interactions

Main effect and lower order terms must not be removed from the model if higher order terms and/or interactions remain.

Factor Variables

- If the coefficient of a dummy variable is non-significant, the dummy cannot be removed from the model! We either keep the entire factor variable, or remove it fully.
- If variable selection is done manually, then hierachical model comparisons need to be done. R function `step()` does this correctly, but `regsubsets()` unfortunately not.

Lasso

Least Absolute Shrinkage and Selection Operator, i.e. an alternative regression technique that can deal with collinearity, and performs shrinkage / variable selection at the same time.

Idea: $\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} r_i^2 + \lambda \sum_{j=1}^{p} \left|\beta_j\right|$

In contrast to Ridge Regression, the coefficients will not only be shrunken by the L_1 penalization, but depending on λ, some will be zero and hence the respective predictors eliminated.

Also here, coefficients are artificially shrunken and hence biased. The benefit is that they are less variable.

Facts about Lasso

- In contrast to the OLS and Ridge estimators, there is no explicit solution to Lasso. This means that the solution has to be found by numerical optimization.

- Using the Coordinate Descent procedure in R allows for finding the solution up to problem size around $np \approx 10^6$.

- In contrast to the OLS and Ridge estimators, Lasso is not a linear estimator. There is no hat matrix H, s.t. $\hat{y} = Hy$.

- As a consequence, the concept of degrees of freedom does not exist for Lasso and there is no trivial procedure for choosing the optimal penalty parameter λ.

R-Code for Lasso

Without standardizing, coefficients on the original scale:

```
> library(glmnet)
> xx  <- model.matrix(hipcenter~0+., my.sp)
> yy  <- my.sp$hipcenter
> fit   <- glmnet(xx,yy)
```

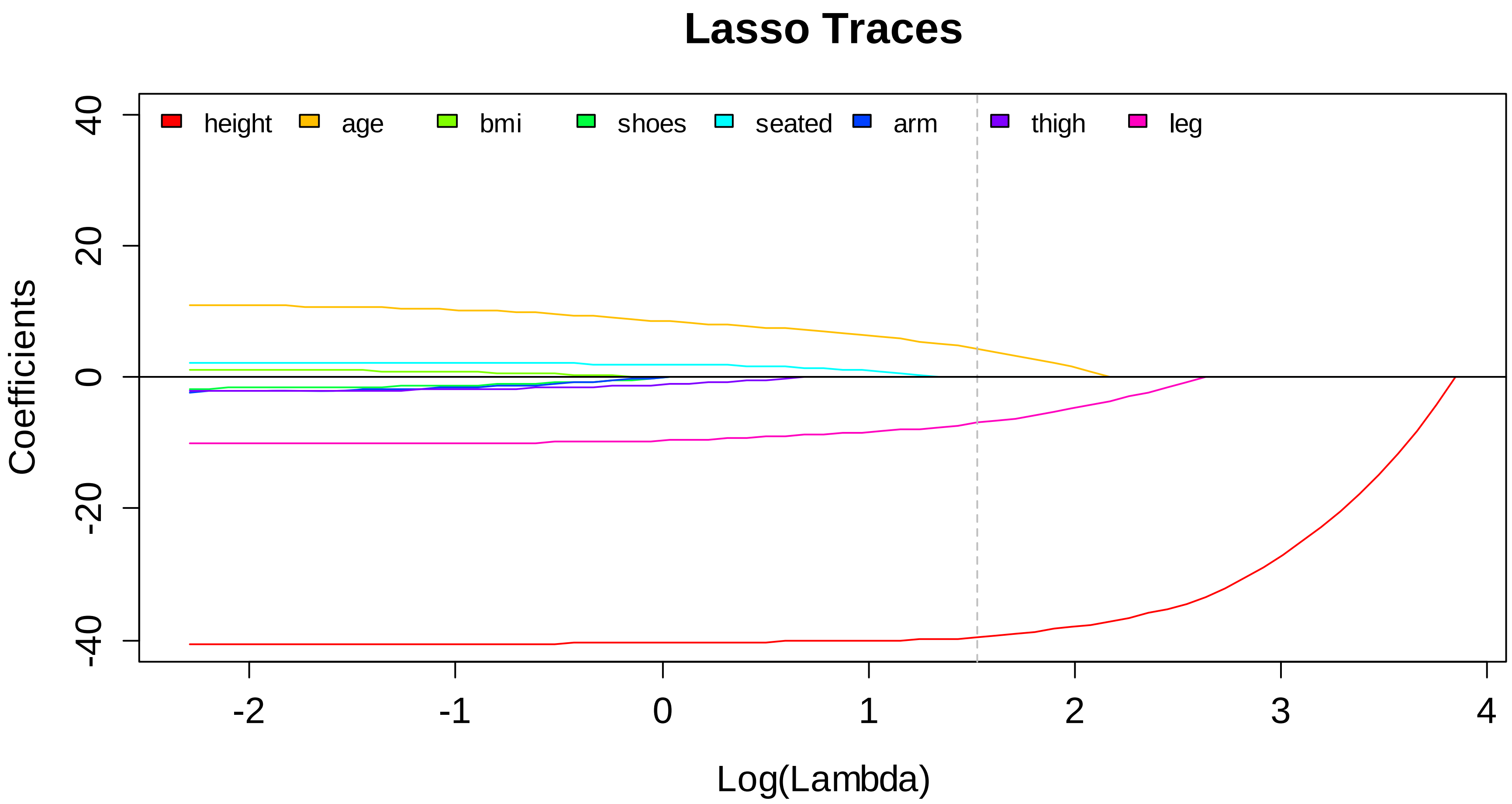
With standardization, which makes coefficients comparable:

```
> zz  <- xx
> for (i in 1:ncol(zz)) zz[,i] <- scale(zz[,i])
> fit.z <- glmnet(zz, yy, standardize=FALSE)
```

Choose your approach depending on the problem/question!

Applied Statistical Regression
AS 2015 – Multiple Regression

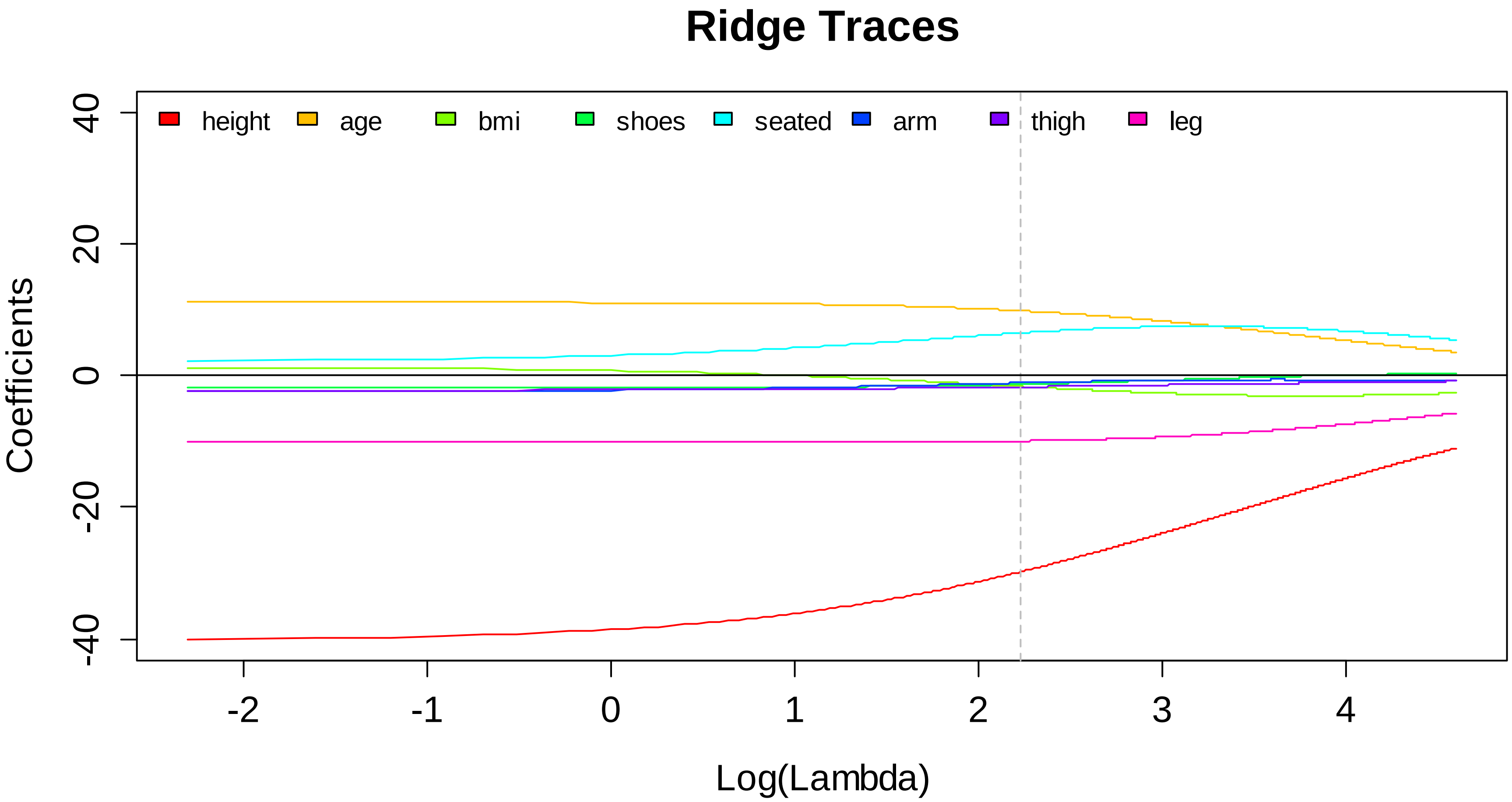
Output: Lasso Traces



Applied Statistical Regression

AS 2015 – Multiple Regression

Comparison: Ridge Traces



Choice of Lambda

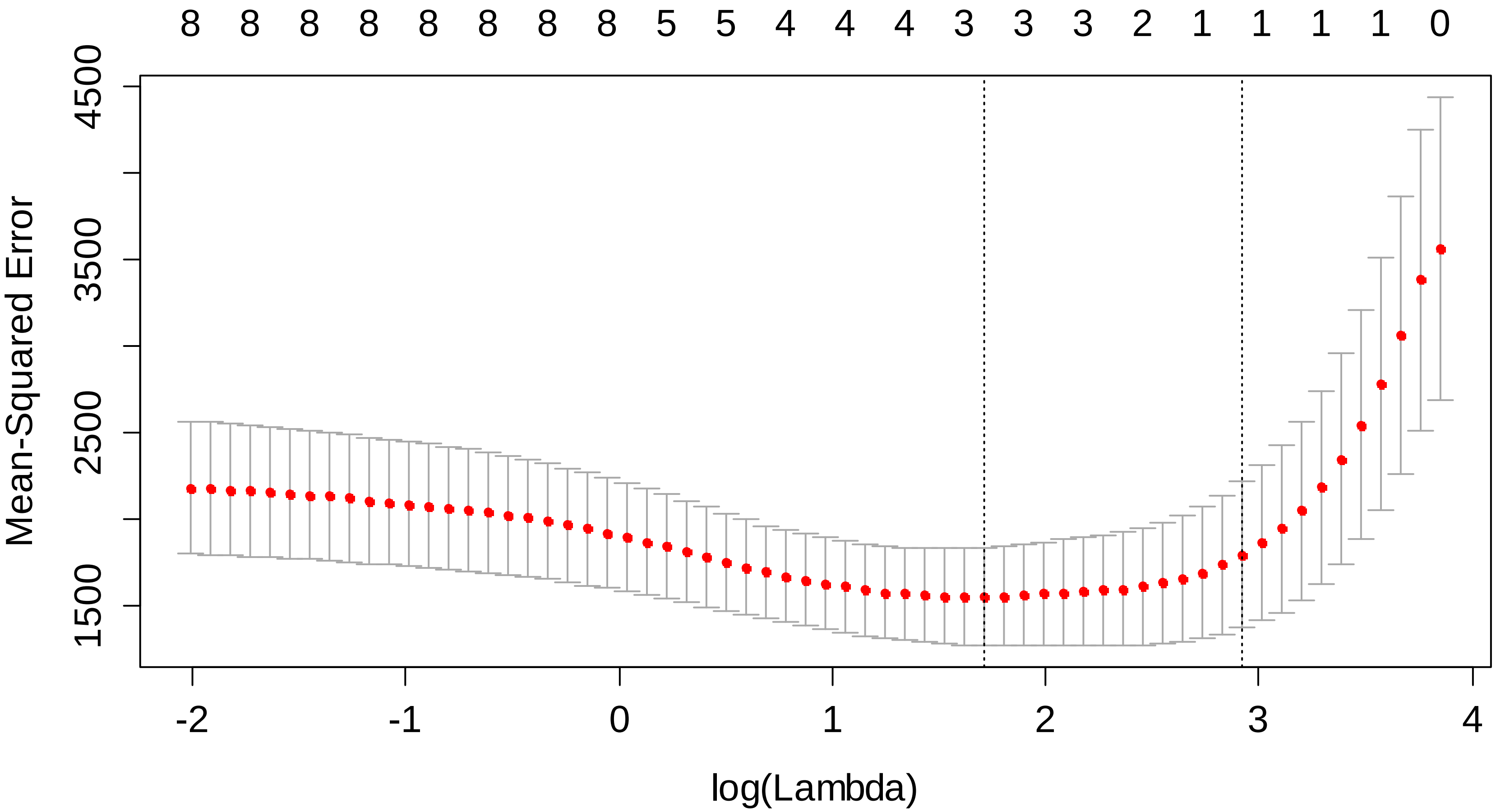
The R function yields values of R-Squared, λ and the number of predictors used in a particular model. These can be plotted for choosing the penalty parameter:

→ λ is determined such that a small modell with only few predictors but still reasonably good R-Squared results.

It is better and more professional to use a cross validation approach, where the predictive performance is determined with various penalty parameters λ.

→
```
> cvfit <- cv.glmnet(zz,yy)
```

Choice of λ with Cross Validation



Lasso: Summary

- The lecturers view is that Lasso predominantly is a method for variable selection. There is a convenient interface for determining the correct penalty with cross validation.
- Due to the built-in shrinkage property, Lasso is much less susceptible to multicollinearity. However, too many collinear predictors can still hamper model interpretation in practice.
- Inference on the fitted model is at best difficult, or even close to impossible. One can compare standardized coefficients.
- The standard Lasso only works with numerical predictors. Extension to factor variables exist, see → *Group Lasso*.

Variable Selection: Round Up

- The standard procedure is to use R function `step()` with default settings, starting from the full model.

- Alternatives exist with other search heuristics, modified criteria and modern methods such as the Lasso.

- Usually, each procedure yields a different "best model". These are often quite instable, i.e. small changes in the data can produce markedly different results.

→ We thus recommend, to not only consider the "best model" according to a particular procedure, but to also take some (similarly good) competitive models (if they exist).

Cross Validation

Definition:

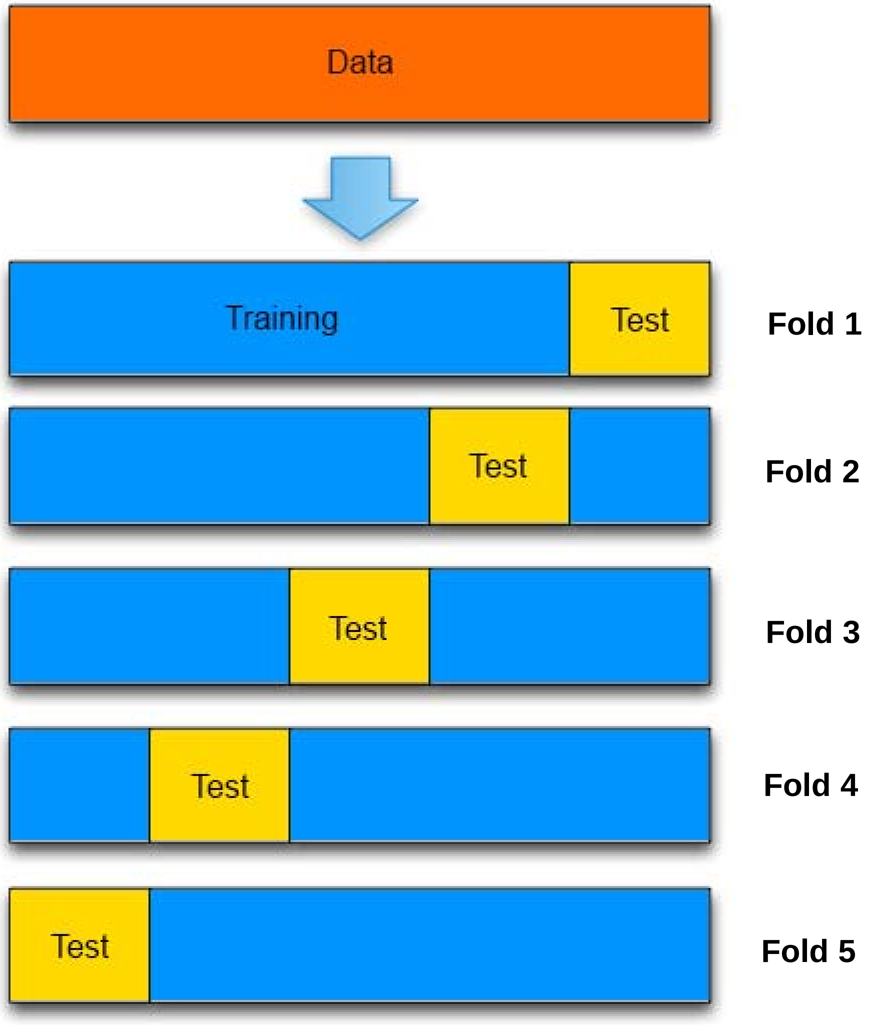
Cross Validation is a *technique for estimating the performance of a predictive model*, i.e. assessing how the prediction error will generalize to a new, independent dataset.

Rationale:

Cross Validation serves to prevent *overfitting*. On a given data set, a *bigger model* always yields a *better fit*, i.e. *smaller RSS*, *higher R-squared*, *less error variance*, et cetera.

While *AIC/BIC* and the *adjusted R-squared* try to overcome this problem by penalizing for model size, its use is limited in reality.

Cross Validation: How It Works…



Data

Training Test Fold 1

Test Fold 2

Test Fold 3

Test Fold 4

Test Fold 5

- In this schematic example, 5-fold CV is illustrated.
- Each observations belongs to exactly 1 test set. The test sets are of roughly equal size.
- Also, each data point belongs to exactly 4 training sets.
- In each fold, the test RSS is recorded. The CV-RSS is the summed result over all folds.

Cross Validation: Evaluation

By splitting the data into k training and test sets we obtain a predicted value $\hat{y}_i^{Test}$ for each of the data points, from which we can derive the squared prediction error $(y_i - \hat{y}_i^{Test})^2$. We then determine the mean squared prediction error:

$$MSPE_{CV} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{Test})^2$$

This is used as a quality criterion for the model. There is no need for a penalty term for model size p as with AIC selection. Bigger models do not have an advantage here – why?

Cross Validation: Advantages

Cross validation is somewhat more laborious than AIC-based variable selection. In contrast, it is a very general procedure that underlies very few restrictions. The only key point is that the same response variable is predicted.

→ We can perform cross validation on datasets with different number of observations, or even on different datasets.

→ The models which are considered in a comparison need not to be hierarchical, and can be arbitrarly different.

→ It is possible to infer the effect of response variable transformations, Lasso, Ridge, robust procedures, …

Cross Validation: When to Use?

AIC/BIC and Adjusted R-squared do not work if:

- *The response variable is transformed:* for investigating whether we obtain better predictions from a model with transformed response or not, cross validation is a must.
- *The sample is not identical:* if we need to check whether excluding data points from the fit yields better predictions for the entire sample, we require cross validation.
- The *performance of alternative methods* such as Lasso, Ridge or Robust Regression shall be investigated. In this case, neither tests nor AIC-comparisons can serve.
- One predominantly aims for a *good prediction model*.

Cross Validation in Practice

As we have seen, CV is the most flexible, but also the most laborious quality criterion. In most cases, doing a systematic variable selection with CV is too time consuming.

Hence, one usually reduces to some promising models by applying other tools and then compares these against each other by CV:

→ There is some R functionality for CV:

```
> library(DAAG)
> CVlm(data, formula, fold.number, …)
```

→ This function is quite poor. In most cases, you will need to set up a CV loop by yourself, see next slide…

Cross Validation Loop

By using R function `for()`**:**

```
> rss   <- c()
> fo    <- 5
> sb    <- round(seq(0,nrow(dat),length=(fo+1)))
> for (i in 1:fo)
> {
>   test   <- (sb[((fo+1)-i)]+1):(sb[((fo+2)-i)])
>   train  <- (1:nrow(dat))[-test]
>   fit    <- lm(res ~ p1+..., data=dat[train,])
>   pred   <- predict(fit, newdata=dat[test,])
>   rss[i] <- sum((dat$response[test] - pred)^2)
> }
```

Cross Validation: Example

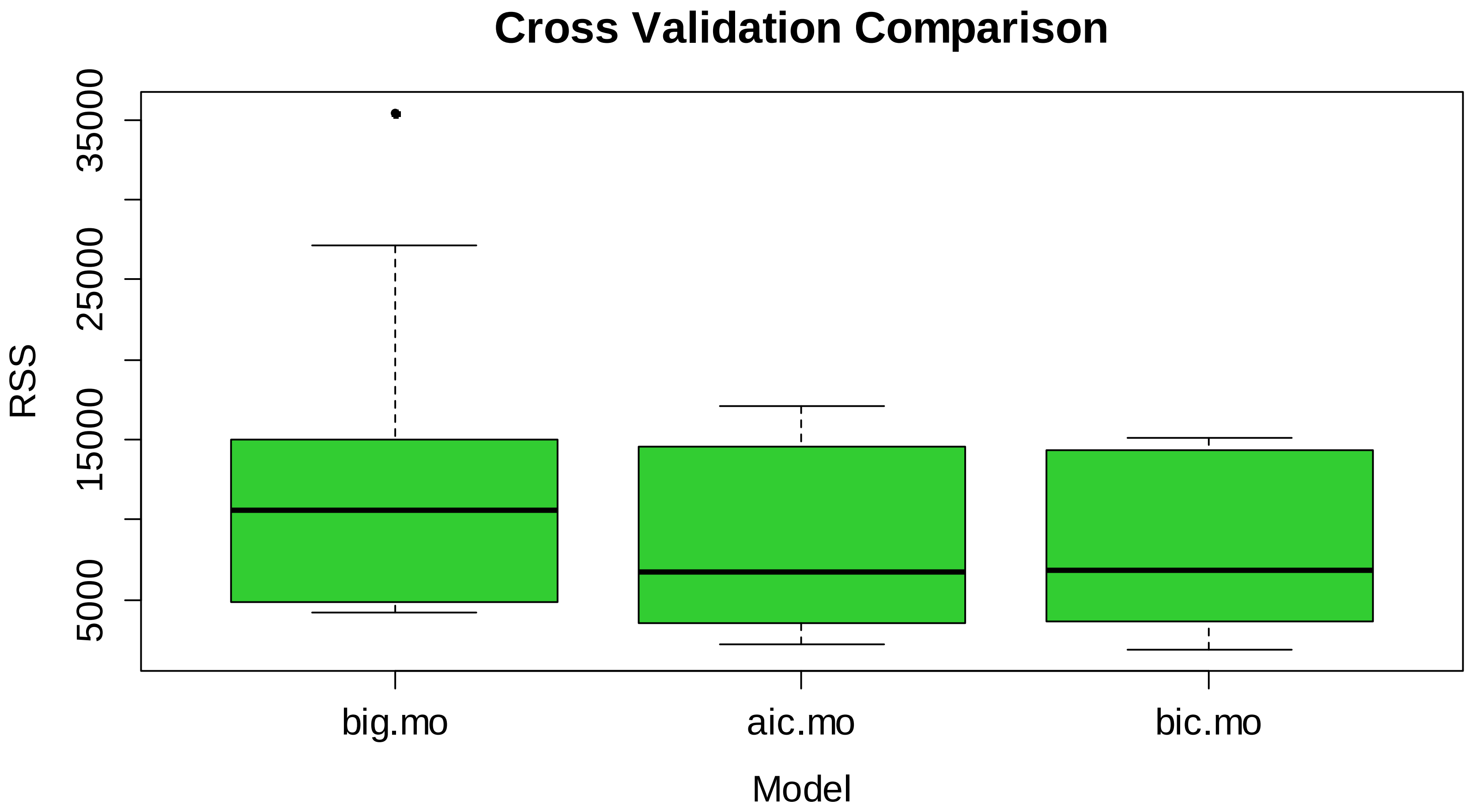
We compare the performance for mortality predictions using the full model with all predictors, as well as with the two smaller models that originate from AIC- and BIC-selection:

```
> rss <- data.frame(rss1, rss2, rss3)
> apply(rss,2,mean)
   big.mo    aic.mo    bic.mo
13213.182  8774.873  8329.638
```

As we can see, the BIC model yields the lowest MSPE. Hence, the smallest model with only 5 predictors is superior to the bigger models when it comes to out-of-sample performance.

→ Plotting the SPEs can yield further insight…

Displaying the Prediction Errors



Modelling Strategies

We have learnt a number of techniques for dealing with multiple linear regression problems. The often asked question is in which order the tools need to be applied:

Data Preparation → Transformation → Estimation → Model Diagnostics → Variable Refinement & Selection → Evaluation

This is a good generic solution, but not an always-optimal strategy.

Professional regression analysis is the search for structure in the data. It requires technical skill, flexibility and intuition. The analyst must be alert to the obvious as well as to the non-obvious, and needs the flair to find the unexpected.

Modelling Strategies

0) Data Screening & Processing

- learn the meaning of all variables
- give short and informative names
- check for impossible values, errors
- better to have missing than wrong data!!!
- are there systematic or random missings?

1) Variable Transformations

- bring all variables to a suitable scale
- use statistical and specific knowledge
- log-transform variables on a relative scale
- break obvious collinearities already at this point

Modelling Strategies

2) Fitting a First Model

Fit a big model with (potentially too) many predictors

- use all if $p < n/5$!!!

- *or* preselect manually according to previous knowledge

- *or* preselect with forward search and a p-value of 0.2

3) Model Diagnostics

- generate the 4 standard plots in R

- a systematic error is non-tolerable, improve the model!!!

- be aware to influential data points, try to understand them

- take care with non-constant variance & long-tailed errors

- think about potential correlation in the residuals

Modelling Strategies

4) Variable Selection

- try to reduce the model to what is utterly required
- run a stepwise search from the full model with AIC/BIC
- if feasible, an all-subset-search with AIC/BIC is even better
- the residual plots must not (substantially) degrade in quality!

5) Refining the Model

- use partial residual plots or plots against other variables
- think about potential non-linearities/factorization in predictors
- interaction terms can improve the fit drastically
- are there still any collinearities that disturb?
- may methods such as Lasso or Ridge help?

Modelling Strategies

6) Plausibility

- implausible predictors, wrong signs, results against theory?
- remove if (appropriate) and no drastic change in outcome

7) Evaluation

- cross validation for model comparison & performance
- derive test results, confidence and prediction intervals

8) Reporting

- be honest and openly report manipulations & decisions
- regression models are descriptive, but not causal!
- do not confuse significance and relevance!

Significance vs. Relevance

The larger a sample, the smaller the p-values for the very same predictor effect. Thus do not confuse small p-values with an important predictor effect!!!

With large datasets, we can have:

- statistically significant results which are practically useless
- e.g. high evidence that the response value is lowered by 0.1% which is often a practically totally meaningless result.

Bear in mind that generally:

- most predictors have influence, thus $\beta_j = 0$ hardly ever holds
- the point null hypothesis is thus usually wrong in practice
- we just need enough data so that we are able to reject it

Significance vs. Relevance

Absence of Evidence $\neq$ Evidence of Absence

- if one fails to reject a null hypothesis $\beta_j = 0$ we do not have a proof that the predictor does not influence the response.

- things may change if we have more data, or even if the data remain the same, but the set of predictors is altered.

Measuring the Relevance of Predictors:

- maximum effect of a predictor variable on the response: $\beta_j \cdot (\max_i x_{ij} - \min_i x_{ij})$
- this can be compared to the total span in the response, or it can be plotted vs. the (logarithmic) p-value.

Mortality: Which Predictors Are Relevant?

```
> summary(fit.step)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1031.9491    80.2930  12.852  < 2e-16 ***
JanTemp       -2.0235     0.5145  -3.933  0.00025 ***
Rain           1.8117     0.5305   3.415  0.00125 **
Educ         -10.7463     7.0797  -1.518  0.13510
NonWhite       4.0401     0.6216   6.500  3.1e-08 ***
WhiteCollar   -1.4514     1.0451  -1.389  0.17082
log(Nox)      19.2481     4.5220   4.257  8.7e-05 ***
---
Residual standard error: 33.72 on 52 degrees of freedom
Multiple R-squared: 0.7383, Adjusted R-squared: 0.7081
F-statistic: 24.45 on 6 and 52 DF,  p-value: 1.543e-13
```

Mortality: Which Predictors Are Relevant?

Implementing the idea of maximum predictor effect:

```
> mami   <- function(col) max(col)-min(col)
> ranges <- apply(mort,2,mami)[c(2,5,6,8,9,14)]
> ranges
JanTemp    Rain  Educ  NonWhite  WhiteCollar  log.NOx
  55.00   55.00  3.30     37.70        28.40     5.77
>
> rele   <- abs(ranges*coef(fit.step)[-1])
> rele
JanTemp    Rain  Educ  NonWhite  WhiteCollar  log.NOx
 111.29   99.64 35.46    152.31        41.22   110.97
```

Predictor contributions are quite evenly distributed here.
Maximum span in the response is **322.43**

Relevance: Standardized Coefficients

Another way of quantifying the impact of a particular predictor is by standardizing all predictors to mean zero and unit variance. This makes the coefficients β_j directly comparable.

```
> library(relaimpo)

> calc.relimp(fit.or, type="betasq", rela=TRUE)
Total response variance: 3896.423
Proportion of variance explained by model: 73.83%
Metrics are normalized to sum to 100% (rela=TRUE).
                betasq
JanTemp     0.14838879    NonWhite    0.46467477
Rain        0.15460185    WhiteCollar 0.01903859
Educ        0.02938728    logNOx      0.18390873
```

Relevanz: LMG Criterion

The relatively simple approaches from before can be shown as being theoretically unfounded. Better in this regard is the LMG criterion. It is relatively complicated, we do not give details:

```
> library(relaimpo)

> calc.relimp(fit.or, type="lmg", rela=TRUE)
Total response variance: 3896.423
Proportion of variance explained by model: 73.83%
Metrics are normalized to sum to 100% (rela=TRUE).
                betasq
JanTemp     0.06027426    NonWhite    0.42530033
Rain        0.16412106    WhiteCollar 0.05357159
Educ        0.15815923    logNOx      0.13857353
```

What is a Good Model?

- The *true model* is a concept that exists in theory & simulation, but whether it does in practice remains unclear. Anyway, it is not realistic to identify the true model in observational studies.

- A **good model** is *useful* for the task at hand, *correct*ly describes the data without any systematical errors, has good *predictive* power and is *practical*/applicable for future use.

- Regression models in observational studies are *always only descriptive, but never causal*. A good model yields an accurate idea which of the observed variables drives the variation in the response, but not necessarily reveals the true mechanisms.