Module 3: Linear Regression

TMA4268 Statistical Learning V2020

Stefanie Muff, Department of Mathematical Sciences, NTNU

January xx, 2020

Introduction

Learning material for this module

- James et al (2013): An Introduction to Statistical Learning. Chapter 3.

We need more statistical theory than is presented in the textbook, which you find in this module page.

Module overview

Todo

- Extensions and challenges (self study)
  - qualitative predictors: dummy coding (needed)
  - non-additivity: including interactions (useful)
- Important results in MLR
- Summing up with team Kahoot! (if time permits!)

Linear regression

- Very simple approach for *supervised learning*.
- Parametric.
- Quantitative response vs. one or several explanatory variables.
- Aims:
  - **Prediction** - “black box”
  - **Explanation** - understanding the relationship between *explanatory variables* and the response
- Is linear regression too simple? Maybe, but very useful. Important to *understand* because many learning methods can be seen as generalization of linear regression.

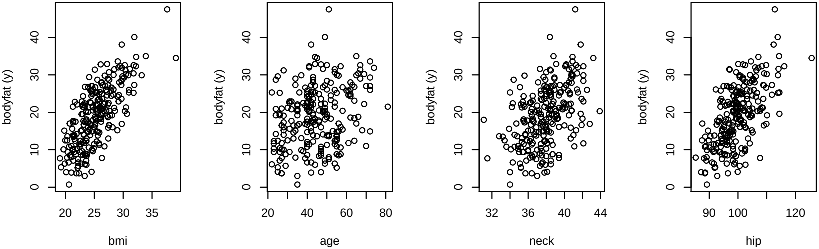
Motivating example: Prognostic factors for body fat

(From Theo Gasser & Burkhardt Seifert *Grundbegriffe der Biostatistik*)

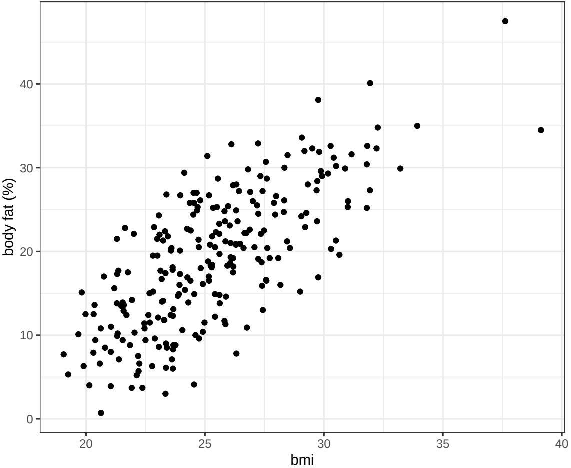
Body fat is an important indicator for overweight, but difficult to measure.

Question: Which factors allow for precise estimation (prediction) of body fat?

Study with 243 male participants, where body fat (%) and BMI and other predictors were measured. Some scatterplots:



For a good predictive model we need to dive into *multiple linear regression*. However, wer start with the simple case of *only one predictor variable*:



Interesting questions

1. How good is BMI as a predictor for body fat?
2. How strong is this relationship?
3. Is the relationship linear?
4. Are also other variables associated with `bodyfat`?
5. How well can we predict the bodyfat of a person?

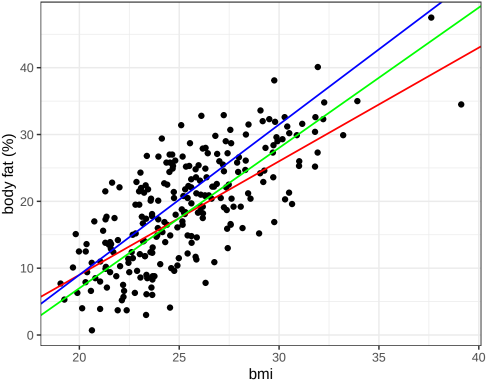
Simple Linear Regression

- One quantitative response Y is modelled
- from *one covariate* x (=simple),
- and the relationship between Y and x is assumed to be *linear*.

If the relation between Y and x is perfectly linear, all instances of (x, Y), given by (x_i, y_i), $i = 1, \ldots, n$, lie on a straight line and fulfill

$$y_i = \beta_0 + \beta_1 x_i \ .$$

But which is the "true" or "best" line, if the relationship is not exact?



Task: Estimate the intercept and slope parameters (by "eye") and write it down (we will look at your answers later).

It is obvious that

- the linear relationship does not describe the data perfectly.
- another realization of the data (other 243 males) would lead to a slightly different picture.

⇒ We need a **model** that describes the relationship between BMI and bodyfat.

The simple linear regression model

In the linear regression model the dependent variable Y is related to the independent variable x as

$$Y = \beta_0 + \beta_1 x + \varepsilon \ , \qquad \varepsilon \sim N(0, \sigma^2) \ .$$

In this formulation Y is a random variable $Y \sim N(\beta_0 + \beta_1 x, \sigma^2)$ where

$$Y \quad = \quad \underbrace{\text{expected value}}_{\mathrm{E}(Y)=\beta_0+\beta_1 x} \quad + \quad \underbrace{\text{error}}_{\varepsilon} \ .$$

Note:

- The model for Y given x has three parameters: β_0 (intercept), β_1 (slope coefficient) and σ^2 .
- x is the independent/ explanatory / regressor variable.
- Y is the dependent / outcome / response variable.

Modeling assumptions

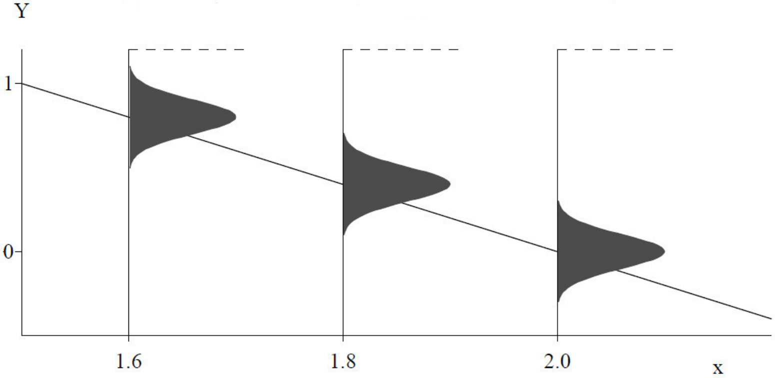
The central assumption in linear regression is that for any pairs (x_i, Y_i), the error $\varepsilon_i \sim N(0, \sigma^2)$. This implies

1. The expected value of ε_i is 0: $\mathrm{E}(\varepsilon_i) = 0$.
2. All ε_i have the same variance: $\mathrm{Var}(\varepsilon_i) = \sigma^2$.
3. All ε_i are normally distributed.
4. ε is independent of any variable, observation number etc.
5. $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ are independent of each other.

Visualization of the regression assumptions

The assumptions about the linear regression model lie in the error term

$$\varepsilon \sim N(0, \sigma^2) \ .$$



Note: The true regression line goes through $\mathrm{E}(Y)$.

Parameter estimation (“model fitting”)

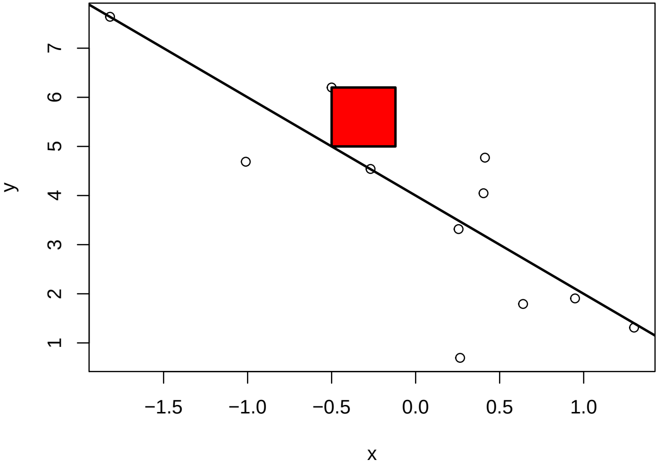
In a regression analysis, the task is to estimate the **regression coefficients** β_0, β_1 and the **residual variance** σ^2 for a given set of (x, y) data.

- **Problem:** For more than two points (x_i, y_i), $i = 1, \ldots, n$, there is generally no perfectly fitting line.
- **Aim**: We want to find the parameters (a, b) of the best fitting line $Y = a + bx$.
- **Idea:** Minimize the deviations between the data points (x_i, y_i) and the regression line.

But what are we actually going to minimize?

Least squares

Remember the **Least Squared Method**. Graphically, we are minimizing the sum of the squared distances over all points:



- Mathematically, β_0 and β_1 are estimated such that the sum of squared vertical distances (residual sum of squares)

$$RSS = \sum_{i=1}^{n} e_i^2 \ , \qquad \text{where} \quad e_i = y_i - (a + bx_i)$$

is being minimized.

- The respective "best" estimates are called $\hat{\beta}_0$ and $\hat{\beta}_1$.
- We can predict the value of the response for a (new) observation of the covariate at x.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x.$$

- The i-th *residual* of the model is the difference between the i-th *observed* response value and the i-th *predicted* value, and is written as:

$$e_i = Y_i - \hat{y}_i.$$

- We may regard the residuals as *predictions* (not estimates) of the error terms ε_i.

(The error terms are random variables and can not be estimated - they can be predicted. It is only for parameters that we speak about estimates.)

Least squares estimators:

Using n observed independent data points

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \ ,$$

the least squares estiamtes for simple linear regression are given as

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \tag{1}$$

and

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \ , \tag{2}$$

where $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$ and $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ are the sample means.

Do-it-yourself “by hand”

Go to the Shiny gallery and try to “estimate” the correct parameters.

You can do this here:

https://gallery.shinyapps.io/simple_regression/

Example continued: Body fat

Assume a linear relationship between the % bodyfat (`bodyfat`) and the BMI (`bmi`), we can get the LS estimates using R as follows:

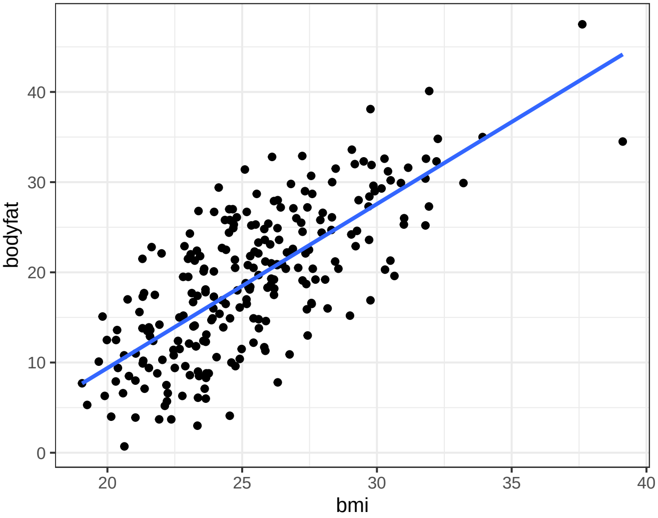
```
r.bodyfat = lm(bodyfat ~ bmi, data = d.bodyfat)
```

The estimates (and more information) can be obtained as follows:

```
summary(r.bodyfat)$coef
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -26.984368  2.7689004 -9.745518 3.921511e-19
## bmi           1.818778  0.1083411 16.787522 2.063854e-42
```

We see that the model fits the data quite well. It captures the essence. It looks that a linear relationship between `bodyfat` and `bmi` is a good approximation.



Questions:

- The blue line gives the estimated model. Explain what the line means in practice. Is this result plausible?
- Compare the estimates for β_0 and β_1 to the estimates you gave at the beginning - were you close?
- How does this relate to the *true* (population) model?
- By looking at the spread of the points around the line, can you detect any violations of the modelling assumptions?
- Finally: **What could the regression line look like if another set of 243 males were used for estimation?**

Uncertainty in the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$

Note: $\hat{\beta}_0$ and $\hat{\beta}_1$ are themselves random variables and as such contain uncertainty!

Let us look again at the regression output, this time only for the coefficients. The second column shows the standard error of the estimate:

```
summary(r.bodyfat)$coef
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -26.984368  2.7689004 -9.745518 3.921511e-19
## bmi           1.818778  0.1083411 16.787522 2.063854e-42
```

→ The logical next question is: what is the distribution of the estimates?

Distribution of the estimators for $\hat{\beta}_0$ and $\hat{\beta}_1$

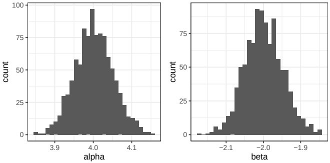
To obtain an intuition, we generate data points according to model

$$y_i = 4 - 2x_i + \varepsilon_i \ , \quad \varepsilon_i \sim N(0, 0.5^2).$$

In each round, we estimate the parameters and store them:

```r
set.seed(1)
niter <- 1000
pars <- matrix(NA, nrow = niter, ncol = 2)
for (ii in 1:niter) {
    x <- rnorm(100)
    y <- 4 - 2 * x + rnorm(100, 0, sd = 0.5)
    pars[ii, ] <- lm(y ~ x)$coef
}
```

Doing it 1000 times, we obtain the following distributions for $\hat{\beta}_0$ and $\hat{\beta}_1$:



Accuracy of the parameter estimates

- The standard errors of the estimates are given by the following formulas:

$$\mathrm{Var}(\hat{\beta}_0) = \mathrm{SE}(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \right]$$

  and

$$\mathrm{Var}(\hat{\beta}_1) = \mathrm{SE}(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

- $\mathrm{Cov}(\hat{\beta}_0, \hat{\beta}_1)$ is in general different from zero.

Note: We will *derive a general version* of these formulas for multiple linear regression, because without matrix notation this is very cumbersome.

Under the assumption that $\varepsilon \sim N(0, \sigma^2)$, we have in addition that

$$\hat{\alpha} \sim N(\alpha, \sigma^2_{\beta_0}) \quad \text{and} \quad \hat{\beta} \sim N(\beta, \sigma^2_{\beta_1}) \ .$$

This implies that that $\hat{\beta}_0$ and $\hat{\beta}_1$ as defined in formulas (1) and (2).

Design issue with data collection

Recall that

$$\mathrm{SE}(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \ ,$$

thus for a given σ^2, the standard error is only dependent on the *design* of the x_i's!

- Would we like the $\mathrm{SE}(\hat{\beta}_1)^2$ large or small? Why?
- If it is possible for us to choose the x_i's, which strategy should we use to choose them?
- Assume x can take values from 1 to 10 and we choose $n = 10$ values. Which is the best design?
    - evenly in a grid: $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$.
    - only lower and upper value: $[1, 1, 1, 1, 1, 10, 10, 10, 10, 10]$.
    - randomly drawn from a uniform distribution on $[1, 10]$.

```
x1 = seq(1:10)
x2 = c(rep(1, 5), rep(10, 5))
x3 = runif(10, 1, 10)

sd1 = sqrt(1/sum((x1 - mean(x1))^2))
sd2 = sqrt(1/sum((x2 - mean(x2))^2))
sd3 = sqrt(1/sum((x3 - mean(x3))^2))

print(c(sd1, sd2, sd3))
```

```
## [1] 0.11009638 0.07027284 0.11505715
```

→ The second design - all observations at extremes - is best!

Residual standard error (RSE)

- **Problem**: σ is usually no known, but needs to be estimated.
- Remember: The residual sum of squares is $\text{RSS} = \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$.
- An estimate of σ, the residual standard error, RSE, is given by

$$\hat{\sigma} = \text{RSE} = \sqrt{\frac{1}{n-2}\text{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \ .$$

- It is related to the amount the response variables deviate from the estimated regression line.
- So actually we have

$$\hat{\text{SE}}(\hat{\beta}_1)^2 = \frac{\hat{\sigma}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \ ,$$

but we usually just write $\text{SE}(\hat{\beta}_1)^2$ (without the extra hat).

If the simple linear regression assumptions are fulfilled, that is, $\varepsilon_i \sim N(0, \sigma^2)$ and all ε_i independent, then it can be shown that

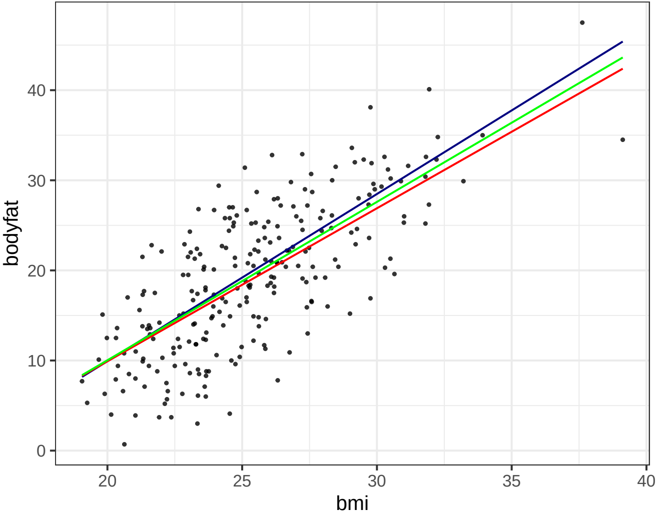
$$\frac{\mathrm{RSE}^2(n-2)}{\sigma^2} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sigma^2} \sim \chi^2_{n-2}$$

The estimated standard errors can be seen using the `summary()` function:

```
summary(r.bodyfat)$coef
```

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -26.984368  2.7689004 -9.745518 3.921511e-19
## bmi           1.818778  0.1083411 16.787522 2.063854e-42
```

To illustrate this point further, again fit the bodyfat example, but each time with only half of the data (randomly selected points each time). See how the model fit varies:



Testing and Confidence Intervals

After the regression parameters and their uncertainties have been estimated, there are typically two fundamental questions:

1. **“Are the parameters compatible with some specific value?”** Typically, the question is whether the slope β_1 might be 0 or not, that is: “Is x an informative predictor or not?”

$\Rightarrow$ This leads to a **statistical test**.

2. “Which values of the parameters are compatible with the data?”

$\Rightarrow$ This leads us to determine **confidence intervals**.

Let’s first go back to the output from the bodyfat example:

```r
summary(r.bodyfat)$coef
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -26.984368  2.7689004 -9.745518 3.921511e-19
## bmi           1.818778  0.1083411 16.787522 2.063854e-42
```

Besides the estimate and the standard error (which we discussed before), there is a `t value` and a probability `Pr(>|t|` that we need to understand.

How do these things help us to answer the two questions above?

Testing the effect of a covariate

Remember: in a statistical test you first need to specify the *null hypothesis*. Here, typically, the null hypothesis is

$$H_0 : \quad \beta_1 = 0 \ .$$

In words: H_0 = "There is no relationship between X and Y."

- Note 1: However, you might want to test against another null hypothesis, like $\beta_1 = c$.
- Note 2: Included in H_0 is the assumption that the data follow the simple linear regression model!

Here, the *alternative hypothesis* is given by

$$H_A : \quad \beta_1 \neq 0$$

Remember: To carry out a statistical test, we need a *test statistic*. This is some type of **summary statistic** that follows a known distribution under H_0. For our purpose, we use the so-called **T-statistic**

$$\boxed{T = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)} \ .}$$

Note: If you want to test against another value than $\beta_1 = 0$, the formula is

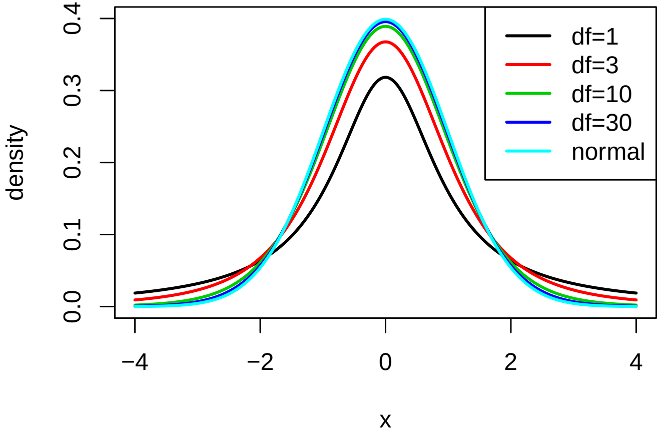
$$\boxed{T = \frac{\hat{\beta}_1 - c}{SE(\hat{\beta}_1)}}$$

Distribution of parameter estimators

We will *derive a general version* for multiple linear regression!
Brief recap:

- Under H_0, T has a t-distribution with $n - 2$ degrees of freedom ($n =$ number of data points; compare to Chapter 8.6 in Walepole et al).

Recap: The t-distribution



- The t-distribution has heavier tails than the normal distribution.
- For df ≥ 30 the t and Normal distribution are pretty similar.

Hypothesis tests for bodyfat example

So let's again go back to the bodyfat regression output:

```r
summary(r.bodyfat)$coef
```

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -26.984368  2.7689004 -9.745518 3.921511e-19
## bmi           1.818778  0.1083411 16.787522 2.063854e-42
```

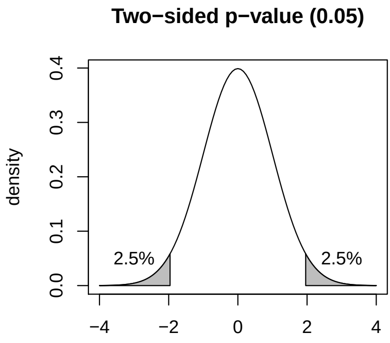
Task: Use the above formulas to derive the T-statistics.

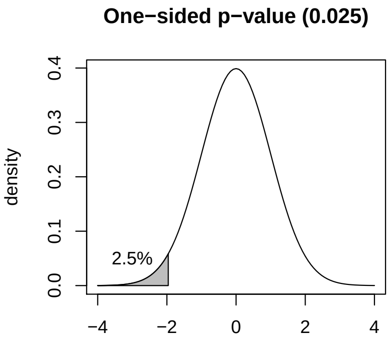
- The last column contains the *p-values* of the tests $\beta_0 = 0$ and $\beta_1 = 0$.
- The p-value for `bmi` is very small ($p < 0.0001$). **What does this mean?**

Recap: Formal definition of the p-value

Formal definition of p-value: the probability to observe a data summary (e.g., an average) that is at least as extreme as the one observed, given that the Null Hypothesis is correct.

Example (normal distribution): Assume the observed test-statistic leads to a z-value = -1.96 $\Rightarrow$ $\mathrm{P}(|z| \geq 1.96) = 0.05$ and $\mathrm{P}(z \leq -1.96) = 0.025$.


Two−sided p−value (0.05)


One−sided p−value (0.025)

Recap: Two types of errors

In the testing setup, we typically *reject the null hypothesis* if the p-value is small enough. Typical cutoffs for the *significance level* (α) are 5% or 1%.

However, this means we can make two types of errors:

- Type I error:
- Type II error:

Cautionary notes regarding p-values:

- The (mis)use of p-values is heavily under critique in the scientific world!!!
- Simple yes/no decisions do often stand on very wiggly scientific ground!!

(See reading tasks for this week.)

Confidence intervals

The t-distribution can be used to create confidence intervals for the regression parameters. The lower and upper limits of a 95% confidence interval for β_j are

$$\hat{\beta}_j \pm t_{1-\alpha/2,n-2} \cdot \mathrm{SE}(\hat{\beta}_j) \quad j = 0, 1.$$

Interpretation of this confidence interval:

- There is a 95% probability that the interval will contain the *true* value of β_j.
- **It is the range of parameter estimates that are *compatible with the data* .**

→ If n is large, the normal approximation to the t-distribution can be used (and is used in the textbook).

Doing this for the bodfat example "by hand" is not hard. We have $241(= 243 - 2)$ degrees of freedom:

```
coefs <- summary(r.bodyfat)$coef
beta <- coefs[2, 1]
sdbeta <- coefs[2, 2]
beta + c(-1, 1) * qt(0.975, 241) * sdbeta
```

```
## [1] 1.605362 2.032195
```

Even easier: directly ask R to give you the CIs.

```
confint(r.bodyfat, level = c(0.95))
```

```
##                  2.5 %     97.5 %
## (Intercept) -32.438703 -21.530032
## bmi           1.605362   2.032195
```

Interpretation: for an increase in the bmi by one index point, roughly 1.82 percentage points more bodyfat are expected, and all true values for β_1 between 1.61 and 2.03 are compatible with the observed data.

Confidence and prediction ranges

- Based on the joint distribution of the intercept and slope it is possible to find the distribution for the linear predictor $\hat{\beta}_0 + \hat{\beta}_1 x$, and then confidence intervals for $\beta_0 + \beta_1 x$.

→ **Confidence range**

- Accounting for the fact that we also have an error in the equation ε, we can also find the distribution of future observations.

→ **Prediction range**

Todo ev say more about confidence and prediction ranges, or put this into exercise class.

Model accuracy

Measured by

1. The **residual standard error (RSE)**, which provides an **absolute measure** of *lack of fit* (see above).

2. The **coefficient of determination** R^2, which measures the proportion of y's variance explained by the model (between 0 and 1), is a **relative measure** of *lack of fit*:

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2},$$

where

$$\text{TSS} = \sum_{i=1}^n (y_i - \bar{y})^2$$

is the *total sum of squares*, a measure for the toal variability in Y.

R^2 in the bodyfat example

```r
summary(r.bodyfat)$r.squared
```

```
## [1] 0.5390391
```

Compare this to the squared correlation between the two variables:

```r
cor(d.bodyfat$bodyfat, d.bodyfat$bmi)^2
```

```
## [1] 0.5390391
```

→ In simple linear regression, R^2 is the squared correlation between the independent and the dependent variable.

Multiple Linear Regression

Remember that the bodyfat dataset contained much more information than only bmi and bodyfat:

- `bodyfat`: % of body fat.
- `age`: age of the person.
- `weight`: body weighth.
- `height`: body height.
- `bmi`: bmi.
- `abdomen`: circumference of abdomen.
- `hip`: circumference of hip.

Model

We assume

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1 + ... + \beta_p X_p + \varepsilon \ , \qquad (3)$$

where X_j is the jth predictor and β_j the respective regression coeffficient.

Assume we have n sampling units $(x_{1i}, \ldots, x_{pi}, y_i)$, $1 \leq i \leq n$, such that each represent an instance of equation (3), we can use the data matrix

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & ... & x_{1p} \\ 1 & x_{21} & ... & x_{2p} \\ \vdots & ... & ... & \vdots \\ 1 & x_{n1} & ... & x_{np} \end{bmatrix}$$

to write the model in matrix form:

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

Notation

- $\mathbf{Y}$: $(n \times 1)$ vector of responses [e.g. one of the following: rent, weight of baby, ph of lake, volume of tree]
- $\mathbf{X}$: $(n \times (p+1))$ design matrix, and $\mathbf{x}_i^T$ is a $(p+1)$-dimensional row row vector for observation i.
- $\boldsymbol{\beta}$: $((p+1) \times 1)$ vector of regression parameters $(\beta_0, \beta_1, \dots, \beta_p)^\top$.
- $\boldsymbol{\varepsilon}$: $(n \times 1)$ vector of random errors.
- We assume that pairs $(\mathbf{x}_i^T, y_i)$ $(i = 1, ..., n)$ are measured from *independent* sampling units.

Remark: other books, including the book in TMA4267 and TMA4315 define p to include the intercept. This may lead to some confusion about p or $p+1$ in formulas...

Classical linear model

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

Assumptions:

1. $\mathrm{E}(\boldsymbol{\varepsilon}) = \mathbf{0}$.
2. $\mathrm{Cov}(\boldsymbol{\varepsilon}) = \mathrm{E}(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T) = \sigma^2\mathbf{I}$.
3. The design matrix has full rank, $\mathrm{rank}(\mathbf{X}) = p + 1$. (We assume $n >> (p + 1)$.)

The classical *normal* linear regression model is obtained if additionally

4. $\boldsymbol{\varepsilon} \sim N_n(\mathbf{0}, \sigma^2\mathbf{I})$ holds. Here N_n denotes the n-dimensional multivarate normal distribution.

Design matrix in R

```
r.bodyfat = lm(bodyfat ~ bmi + age, data = d.bodyfat)
head(model.matrix(r.bodyfat))
```

```
##   (Intercept)   bmi age
## 1           1 23.65  23
## 2           1 23.36  22
## 3           1 24.69  22
## 4           1 24.91  26
## 5           1 25.54  24
## 6           1 26.48  24
```

```
head(d.bodyfat$bmi)
```

```
## [1] 23.65 23.36 24.69 24.91 25.54 26.48
```

```
head(d.bodyfat$age)
```

```
## [1] 23 22 22 26 24 24
```

Distribution of the response vector

Assume that

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \ , \quad \boldsymbol{\varepsilon} \sim N_n(\mathbf{0}, \sigma^2\mathbf{I}) \ .$$

Q:

- What is the mean $\mathrm{E}(\mathbf{Y})$?
- The covariance matrix $\mathrm{Cov}(\mathbf{Y})$ given $\mathbf{X}$?
- What is then the distribution of $\mathbf{Y}$?

A:

Parameter estimation

In multiple linear regression parameters in β are estimated with maximum likelihood and least squares. These two methods give the same estimator when we assume the normal linear regression model.

Least squares and maximum likelihood estimator for $\boldsymbol{\beta}$:

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

The estimator is found by minimizing the RSS for a multiple linear regression model:

$$\begin{aligned}\text{RSS} &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - ... - \hat{\beta}_p x_{ip})^2 \\ &= \sum_{i=1}^{n}(y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 = (\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})\end{aligned}$$

The estimator is found by solving the system of $(p+1)$ equations

$$\frac{\partial \text{RSS}}{\partial \boldsymbol{\beta}} = \mathbf{0}\ .$$

→ Derivation on the board.

Example continued

```r
r.bodyfat3 <- lm(bodyfat ~ bmi + age + neck + hip + abdomen, data = d.bodyfat)
summary(r.bodyfat3)
```

```
##
## Call:
## lm(formula = bodyfat ~ bmi + age + neck + hip + abdomen, data = d.bodyfat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3727 -3.1884 -0.1559  3.1003 12.7613
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.74965    7.29830  -1.062  0.28939
## bmi          0.42647    0.23133   1.844  0.06649 .
## age          0.01457    0.02783   0.524  0.60100
## neck        -0.80206    0.19097  -4.200 3.78e-05 ***
## hip         -0.31764    0.10751  -2.954  0.00345 **
## abdomen      0.83909    0.08418   9.968  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.392 on 237 degrees of freedom
## Multiple R-squared:  0.7185, Adjusted R-squared:  0.7126
## F-statistic:   121 on 5 and 237 DF,  p-value: < 2.2e-16
```

Reproduce the values under `Estimate` by calculating without the use of `lm`.

```
X = model.matrix(r.bodyfat3)
Y = d.bodyfat$bodyfat
betahat = solve(t(X) %*% X) %*% t(X) %*% Y
print(betahat)
```

```
##                    [,1]
## (Intercept) -7.74964673
## bmi          0.42647368
## age          0.01457356
## neck        -0.80206081
## hip         -0.31764315
## abdomen      0.83909391
```

Distribution of the regression parameter estimator

1. We assumed alassical *normal* linear regression model with $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon} \sim N_n(\mathbf{0}, \sigma^2\mathbf{I})$, with full-rank matrix $\mathbf{X}$, leading to
$$\mathbf{Y} \sim N_n(\mathbf{X}\boldsymbol{\beta}, \sigma^{\mathbf{2}}\mathbf{I}) \ .$$
2. Then we "found" that an estimator for β is
$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}.$$

What are

- The mean $\mathrm{E}(\hat{\boldsymbol{\beta}})$?
- The covariance matrix $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$?
- The distribution of $\hat{\boldsymbol{\beta}}$?

Distribution of the regression parameter estimator (summary)

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

This can be written as $\hat{\boldsymbol{\beta}} = \mathbf{C}\mathbf{Y}$ where

- $\mathbf{C} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$
- $\mathbf{Y} \sim N_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$.

Therefore:

- $\mathrm{E}(\hat{\boldsymbol{\beta}}) = \mathbf{C}\mathrm{E}(\mathbf{Y}) = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \boldsymbol{\beta}$.
- $\mathrm{Cov}(\hat{\boldsymbol{\beta}}) = \mathbf{C}\mathrm{Cov}(\mathbf{Y})\mathbf{C}^T = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\sigma^2\mathbf{I}((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)^T = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2$.
- $\hat{\boldsymbol{\beta}}$ is multivariate normal $(p+1)$ dimensions.

So: $\hat{\beta} \sim N_{p+1}(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$.
(Todo: Use PropertiesBetahatMLR.pdf for a derivation)

How does this compare to simple linear regression? Not so easy to see a connection!

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x} \text{ and } \hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

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

Often we use centered data (and also scaled) to ease interpretation.

Another data set: Ozone

New York, 1973: 111 observations of

- `ozone` : ozone concentration (ppm); **response variable**
- `radiation` : solar radiation (langleys)
- `temperature` : daily maximum temperature (F)
- `wind` : wind speed (mph)

```
library(ElemStatLearn)
data(ozone)
head(ozone)
```

```
##   ozone radiation temperature wind
## 1    41       190          67  7.4
## 2    36       118          72  8.0
## 3    12       149          74 12.6
## 4    18       313          62 11.5
## 5    23       299          65  8.6
## 6    19        99          59 13.8
```

```r
ozone.lm = lm(ozone ~ temperature + wind + radiation, data = ozone)
```

```r
head(model.matrix(ozone.lm))
```

```
##   (Intercept) temperature wind radiation
## 1           1          67  7.4       190
## 2           1          72  8.0       118
## 3           1          74 12.6       149
## 4           1          62 11.5       313
## 5           1          65  8.6       299
## 6           1          59 13.8        99
```

```r
head(ozone$ozone)
```

```
## [1] 41 36 12 18 23 19
```

```r
summary(ozone.lm)
```

```
##
## Call:
## lm(formula = ozone ~ temperature + wind + radiation, data = ozone)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.485 -14.210  -3.556  10.124  95.600
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -64.23208   23.04204  -2.788  0.00628 **
## temperature   1.65121    0.25341   6.516 2.43e-09 ***
## wind         -3.33760    0.65384  -5.105 1.45e-06 ***
## radiation     0.05980    0.02318   2.580  0.01124 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.17 on 107 degrees of freedom
## Multiple R-squared:  0.6062, Adjusted R-squared:  0.5952
## F-statistic: 54.91 on 3 and 107 DF,  p-value: < 2.2e-16
```

Remember: $\hat{\beta} \sim N_{p+1}(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$. The covariance matrix can be obtained as follows:

```r
vcov(ozone.lm)
```

```
##               (Intercept)  temperature          wind     radiation
## (Intercept) 530.93558002 -5.503192281 -1.043562e+01  0.0266688733
## temperature  -5.50319228  0.064218138  8.034556e-02 -0.0015749279
## wind        -10.43562350  0.080345561  4.275126e-01 -0.0003442514
## radiation     0.02666887 -0.001574928 -3.442514e-04  0.0005371733
```

Four important questions

1. Is at least one of the predictors $X_1, \ldots, X_p$ useful in predicting the response?
2. Do all the predictors help to explain Y, or is only a subset of predictors useful?
3. How well does the model fit the data?
4. Given a set of predictor variables, what reaposne value should we predict, and how accurate is our prediction?

1. Relationship between predictors and response?

Question is whether we could as well omit all predictor variables at the same time, that is

$$H_0 : \beta_1 = \beta_2 = \ldots = \beta_p = 0$$

vs.

$$H_1 : \text{at least one } \beta_j \text{ is non-zero.}$$

To answer this, we need the F-statistic

$$F = \frac{(\mathrm{TSS} - \mathrm{RSS})/p}{\mathrm{RSS}/(n-p-1)} \sim F_{p,(n-p-1)} \ ,$$

where total sum of squares $\mathrm{TSS} = \sum_i (y_i - \bar{y})^2$, and residual sum of squares $\mathrm{RSS} = \sum_i (y_i - \hat{y}_i)^2$. Under the Normal regression assumptions, F follows an $F_{p,(n-p-1)}$ distribution (ref to Walepole book).

- If H_0 is true, F is expected to be 1.
- Otherwise, we expect that the numerator is larger than the denominator (because the regression then explains a lot of variation) and thus F is greater than 1. For an observed value f_0, the p-value is given as

$$p = P(F_{p,n-p-1} > f_0) \ .$$

Checking the F-value in the `R` output:

```r
summary(r.bodyfat)
```

```
##
## Call:
## lm(formula = bodyfat ~ bmi + age, data = d.bodyfat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.0415  -3.8725  -0.1237   3.9193  12.6599
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -31.25451    2.78973 -11.203  < 2e-16 ***
## bmi           1.75257    0.10449  16.773  < 2e-16 ***
## age           0.13268    0.02732   4.857 2.15e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.329 on 240 degrees of freedom
## Multiple R-squared:  0.5803, Adjusted R-squared:  0.5768
## F-statistic: 165.9 on 2 and 240 DF,  p-value: < 2.2e-16
```

Conclusion?

More complex hypotheses

Sometimes we don't want to test if all β's to zero at the same time, but only a subset $1, \ldots, q$:

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_q = 0 \text{ vs. } H_1 : \text{at least one different from zero.}$$

Again, the F-test can be used, but now F is calculated like

$$F = \frac{(\text{RSS}_0\text{-RSS})/(q)}{\text{RSS}/(n-p-1)} \sim F_{q,n-p-1} \ ,$$

where

- Large model: RSS with $p + 1$ regression parameters
- Small model: RSS_0 with $q + 1$ regression parameters

Example in R

- **Question:** Do `weight` and `height` explain something of `bodyfat`, on top of the variables `bmi` and `age`?
- Fit both models and use the `anova()` function to carry out the F-test:

```r
r.bodyfat.large = lm(bodyfat ~ bmi + age, data = d.bodyfat)
r.bodyfat.small = lm(bodyfat ~ bmi + age + weight + height, data = d.bodyfat)
anova(r.bodyfat.large, r.bodyfat.small)
```

```
## Analysis of Variance Table
##
## Model 1: bodyfat ~ bmi + age
## Model 2: bodyfat ~ bmi + age + weight + height
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    240 6816.2
## 2    238 6702.9  2    113.28 2.0112 0.1361
```

Inference about a single predictor β_j

A special case is

$$H_0 : \beta_j = 0 \text{ vs } H_1 : \beta_j \neq 0$$

- Nothing new: We did it for simple linear regression!
- However, now the F-statistic becomes

$$F = \frac{(\text{RSS}_0\text{-RSS})/(p-1)}{\text{RSS}/(n-p-1)} \sim F_{1,n-p-1} \ ,$$

and you know (see e.g. Walepole et al p. xy (todo)) that

$$F_{1,n-p-1} = t^2_{n-p-1} \ ,$$

thus we can use a T-statistics with $(n-p-1)$ degrees of freedom to get the p-value.

Going back again:

```
summary(r.bodyfat)$coef
```

```
##                Estimate Std. Error    t value     Pr(>|t|)
## (Intercept) -31.2545057 2.78973238 -11.203406 1.039096e-23
## bmi           1.7525705 0.10448723  16.773060 2.600646e-42
## age           0.1326767 0.02731582   4.857137 2.149482e-06
```

However:

- Only checking the individual p-values is dangerous. Why? → e.g., multiple testing problem.
- Not possible if $n > p$ → need other approaches (see e.g., module 6).

Inference about β_j: confidence interval

Using that

$$T_j = \frac{\hat{\beta}_j - \beta_j}{\mathrm{SE}(\hat{\beta})} \sim t_{n-p-1} \ ,$$

we can create confidence intervals for β_j in the same manner as we did for simple linear regression. For the typical confidence level $\alpha = 0.05$ we have

$$\hat{\beta}_j \pm t_{0.975,n-p-2} \cdot \mathrm{SE}(\hat{\beta}_j) \ .$$

```r
confint(r.bodyfat)
```

```
##                   2.5 %      97.5 %
## (Intercept) -36.7499929 -25.7590185
## bmi           1.5467413   1.9583996
## age           0.0788673   0.1864861
```

2. Deciding on important variables

3. Model Fit

4. Predictions

Estimator for σ^2

$$
\begin{aligned}
\text{RSS} &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - ... - \hat{\beta}_p x_{ip})^2 \\
&= \sum_{i=1}^{n}(y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}})^2 = (\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})
\end{aligned}
$$

Restricted maximum likelihood estimator for σ^2:

$$
\hat{\sigma}^2 = \frac{1}{n-p-1}(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \frac{\text{RSS}}{n-p-1}
$$

with $\frac{(n-p-1)\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-p-1}$.

Distribution of regression parameters (contd.)

$\hat{\beta} \sim N_{p+1}(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$.

- unbiased
- covariance matrix dependent on the design (and σ^2)

Multicollinearity: columns in design matrix (that is, the covariates) are correlated, which may lead to difficulty in "identifying" the effect of each covariate on the response, and thus large variances (and covariances) for the elements of $\hat{\boldsymbol{\beta}}$.

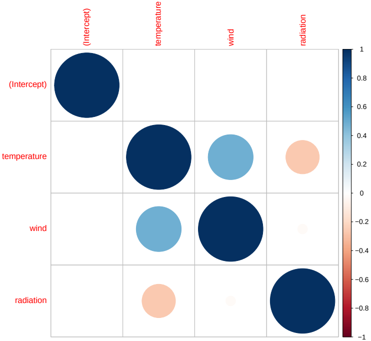
The *variance inflation factor (VIF)* is the ratio of the variance of $\hat{\beta}_j$ when fitting a model with the chosen covariates divided by the variance of $\hat{\beta}_j$ in a simple linear regression.

- VIF=1: absence of collinearity
- VIF exceeding 5 or 10 might be problematic.
- Solution: drop a covariate (that do not add much since it is correlated with other covariates).

```
oz1 = as.data.frame(apply(ozone, 2, scale, scale = FALSE))
fitoz = lm(ozone ~ temperature + wind + radiation, data = oz1)
vif(fitoz)
```

```
## temperature        wind   radiation
##    1.431201    1.328979    1.095241
```

```
corrplot(cov2cor(vcov(fitoz)), cex.lab = 0.7)
```



Prediction intervals

Once we have estimated the coeffients $\hat{\beta}_0$, $\hat{\beta}_1$,.., $\hat{\beta}_p$, we can make a prediction for a response value Y_0 for a new observation $\mathbf{x}_0 = (x_1, x_2, ..., x_p)$ as before:

$$\hat{Y}_0 = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + ... + \hat{\beta}_p x_p = \mathbf{x}_0^T \hat{\boldsymbol{\beta}}.$$

This is an intuitive point estimate.
Remember, one aim for regression was to "construct a model to predict the response from a set of (one or several) explanatory variables- more or less black box".
To assess the uncertainty in this prediction we present a prediction interval for the Y_0.

After some work (see “details on the derivation” below):

$P(\mathbf{x}_0^T\hat{\beta} - t_{\alpha/2,n-p-1}\hat{\sigma}\sqrt{1+\mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0} \leq Y_0 \leq$
$\mathbf{x}_0^T\hat{\beta} + t_{\alpha/2,n-p-1}\hat{\sigma}\sqrt{1+\mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0}) = 1-\alpha$

A $(1-\alpha)\%$ PI for Y_0 is when we insert numerical values for the upper and lower limits: $[\mathbf{x}_0^T\hat{\beta} - t_{\alpha/2,n-p-1}\hat{\sigma}\sqrt{1+\mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0}, \mathbf{x}_0^T\hat{\beta} + t_{\alpha/2,n-p-1}\hat{\sigma}\sqrt{1+\mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0}]$.

PIs can be found in R using `predict` on an `lm` object, but make sure that `newdata` is a `data.frame` with the same names as the original data.

Example: Using the Munich rent index data

We want to predict the rent - with PI - for an appartment with area 50, location 2 (“good”), nice bath and kitchen and with central heating.

```r
library(gamlss.data)
fit = lm(rent ~ area + location + bath + kitchen + cheating, data = gamlss.data
newobs = gamlss.data::rent99[1, ]
newobs[1, ] = c(NA, NA, 50, NA, 2, 1, 1, 1, NA)
predict(fit, newdata = newobs, interval = "prediction", type = "response")
```

```
##        fit      lwr      upr
## 1 602.1298 315.5353 888.7243
```

Q

1. When is a prediction interval of interest?

A: Always? Gives useful information in addition to a point prediction.

2. Explain the result from `predict` above.

A: Fit is point prediction, lwr is lower and upr is upper limit of the 95% PI (default with 95, and `level=0.99` gives 99).

Details in the derivation of the PI

We start to look at the difference between the unobserved response Y_0 (for a given covariate vector $\mathbf{x}_0$) and the point prediction $\hat{y}_0$, $Y_0 - \hat{y}_0$. First, we assume that the unobserved response at covariate $\mathbf{x}_0$ is independent of our previous observations and follows the same distibution, that is $Y_0 \sim N(\mathbf{x}_0^T\beta, \sigma^2)$. Further,

$$\hat{y}_0 = \mathbf{x}_0^T\hat{\beta} \sim N(\mathbf{x}_0^T\beta, \sigma^2\mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0).$$

Then, for $Y_0 - \mathbf{x}_0^T\hat{\beta}$ we have
$\mathrm{E}(Y_0 - \mathbf{x}_0^T\hat{\beta}) = 0$ and $\mathrm{Var}(Y_0 - \mathbf{x}_0^T\hat{\beta}) = \mathrm{Var}(Y_0) + \mathrm{Var}(\mathbf{x}_0^T\hat{\beta}) =$
$\sigma^2 + \sigma^2\mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0$
so that

$$Y_0 - \mathbf{x}_0^T\hat{\beta} \sim N(0, \sigma^2(1 + \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0))$$

Inserting our REML-estimate for σ^2 gives

$$T = \frac{Y_0 - \mathbf{x}_0^T \hat{\beta}}{\hat{\sigma}\sqrt{1 + \mathbf{x}_0^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}_0}} \sim t_{n-p-1}.$$

Then, we start with

$$P(-t_{\alpha/2,n-p-1} \leq \frac{Y_0 - \mathbf{x}_0^T \hat{\beta}}{\hat{\sigma}\sqrt{1 + \mathbf{x}_0^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}_0}} \leq t_{\alpha/2,n-p-1}) = 1 - \alpha$$

and solve so that Y_0 is in the middle, which gives

$P(\mathbf{x}_0^T \hat{\beta} - t_{\alpha/2,n-p-1} \hat{\sigma}\sqrt{1 + \mathbf{x}_0^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}_0} \leq Y_0 \leq$
$\mathbf{x}_0^T \hat{\beta} + t_{\alpha/2,n-p-1} \hat{\sigma}\sqrt{1 + \mathbf{x}_0^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}_0}) = 1 - \alpha$

Coefficient of determination, R^2

no change from simple linear regression.

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2}.$$

1. The interpretation of this coefficient is that the closer it is to 1 the better the fit to the data. If $R^2 = 1$ then all residuals are zero - that is, perfect fit to the data.
2. In a simple linear regression the R^2 equals the squared correlation coefficient between the response and the predictor. In multiple linear regression R^2 is the squared correlation coefficient between the observed and predicted response.

3. If we have two models M1 and M2, where model M2 is a submodel of model M1, then

   $$R^2_{M_1} \geq R^2_{M_2}.$$

   This can be explained from the fact that $\text{RSS}_{M_1} \leq \text{RSS}_{M_2}$. (More in the Recommended exercises.)

Model assessment and selection

Quality measures

To assess the quality of the regression we can report the R^2 coefficient of determination. However, since adding covariates to the linear regression can not make the RSS larger, this means that adding covariates can not make the R^2 smaller. This means that RSS and R^2 are only useful measures for comparing models with the same number of regression parameters estimated.

If we consider two models with the same model complexity then RSS can be used to choose between (or compare) these models.

But, if we want to compare models with different model complexity we need to look at other measures of quality for the regression.

→ **Model selection is a major point in Module 6.**

R^2 adjusted

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

Choose the model with the *largest* R^2_{adj}.

Challenges - for model fit

1. Non-linearity of data
2. Correlation of error terms
3. Non-constant variance of error terms
4. Normality of error terms
5. Outliers
6. High leverage points
7. Collinearity

Diagnostic plots

Plotting residuals - and what to do when assumptions are violated?

1. Plot the residuals against the predicted values, $\hat{y}_i$.

- Dependence of the residuals on the predicted value: wrong regression model?
- Nonconstant variance: transformation or weighted least squares is needed?

2. Plot the residuals, against predictor variable or functions of predictor variables. Trend suggest that transformation of the predictors or more terms are needed in the regression.

3. Assessing normality of errors: QQ-plots and histograms of residuals. As an additional aid a test for normality can be used, but must be interpreted with caution since for small sample sizes the test is not very powerful and for large sample sizes even very small deviances from normality will be labelled as significant.
4. Plot the residuals versus time or collection order (if possible). Look for dependence or autocorrelation.

Residuals can be used to check model assumptions, and also to *discover outliers*.

Different types of residuals

If can be shown that the vector of residuals, $\mathbf{e} = (e_1, e_2, \ldots, e_n)$ have a normal (singular) distribution with mean $\mathrm{E}(\mathbf{e}) = \mathbf{0}$ and covariance matrix $\mathrm{Cov}(\mathbf{e}) = \sigma^2(\mathbf{I} - \mathbf{H})$ where $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$.

This means that the residuals (possibly) have different variance, and may also be correlated.

Q: How can we say that the residuals can have different variance and may be correlated? Why is that a problem?

A:

We would like to check the model assumptions - we see that they are all connected to the error terms. But, but we have not observed the error terms ε so they can not be used for this. However, we have made "predictions" of the errors - our residuals. And, we want to use our residuals to check the model assumptions.

That is, we want to check that our errors are independent, homoscedastic (same variance for each observation), and not dependent on our covariates - and we want to use the residuals (observed) in place of the errors (unobserved). Then it would have been great if the residuals have these properties when the underlying errors have. To amend our problem we need to try to fix the residual so that they at least have equal variances. We do that by working with *standardized* or *studentized residuals*.

Standardized residuals:

$$r_i = \frac{e_i}{\hat{\sigma}\sqrt{1 - h_{ii}}}$$

where h_{ii} is the ith diagonal element of the hat matrix $\mathbf{H}$.

In R you can get the standardized residuals from an `lm`-object (named `fit`) by `rstandard(fit)`.

Studentized residuals:

$$r_i^* = \frac{e_i}{\hat{\sigma}_{(i)}\sqrt{1 - h_{ii}}}$$

where $\hat{\sigma}_{(i)}$ is the estimated error variance in a model with observation number i omitted. This seems like a lot of work, but it can be shown that it is possible to calculated the studentized residuals directly from the standardized residuals.

In R you can get the studentized residuals from an `lm`-object (named `fit`) by `rstudent(fit)`.

Diagnostic plots in R

More information on the plots here:
http://data.library.virginia.edu/diagnostic-plots/ and
http://ggplot2.tidyverse.org/reference/fortify.lm.html

You can use the function `fortify.lm` in `ggplot2` to create a dataframe from an `lm`-object, which `ggplot` uses automatically when given a `lm`-object. This can be used to plot diagnostic plots.

For simplicity we use the Munch rent index with `rent` as response and only `area` as the only covariate. (You may change the model to a more complex one, and rerun the code chunks.)

```
fit <- lm(rent ~ area, data = rent99)  # Run a regression analysis
format(head(fortify(fit)), digits = 4L)
```

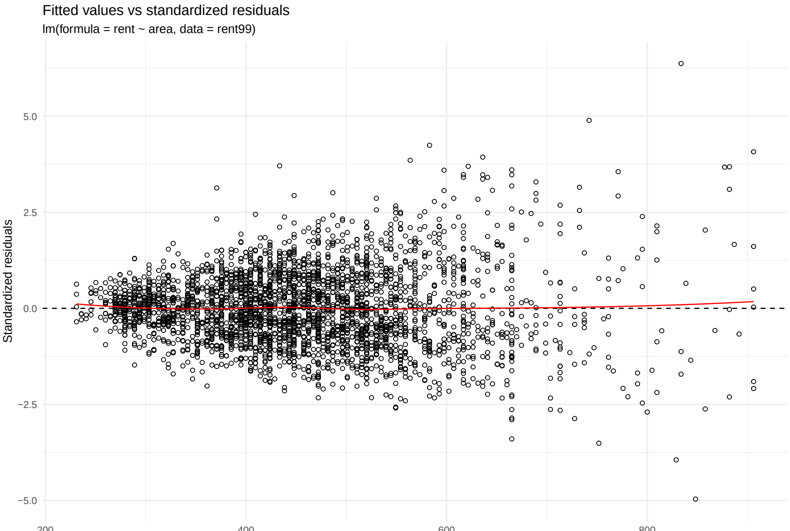
```
##    rent area     .hat .sigma   .cooksd .fitted  .resid .stdresid
## 1 109.9   26 0.001312  158.8 5.870e-04   260.0 -150.00   -0.9454
## 2 243.3   28 0.001219  158.8 1.678e-05   269.6  -26.31   -0.1658
## 3 261.6   30 0.001130  158.8 6.956e-06   279.2  -17.60   -0.1109
## 4 106.4   30 0.001130  158.8 6.711e-04   279.2 -172.83   -1.0891
## 5 133.4   30 0.001130  158.8 4.779e-04   279.2 -145.85   -0.9191
## 6 339.0   30 0.001130  158.8 8.032e-05   279.2   59.79    0.3768
```

```
# to show what fortify implicitly does in ggplot for more information
# ggplot2::fortify.lm
```

Residuals vs fitted values

A plot with the fitted values of the model on the x-axis and the residuals on the y-axis shows if the residuals have non-linear patterns. The plot can be used to test the assumption of a linear relationship between the response and the covariates. If the residuals are spread around a horizontal line with no distinct patterns, it is a good indication on no non-linear relationships, and a good model.
Does this look like a good plot for this data set?

```r
ggplot(fit, aes(.fitted, .stdresid)) + geom_point(pch = 21) + geom_hline(yinterc
    linetype = "dashed") + geom_smooth(se = FALSE, col = "red", size = 0.5,
    method = "loess") + labs(x = "Fitted values", y = "Standardized residuals",
    title = "Fitted values vs standardized residuals", subtitle = deparse(fit$c
    theme_minimal()
```

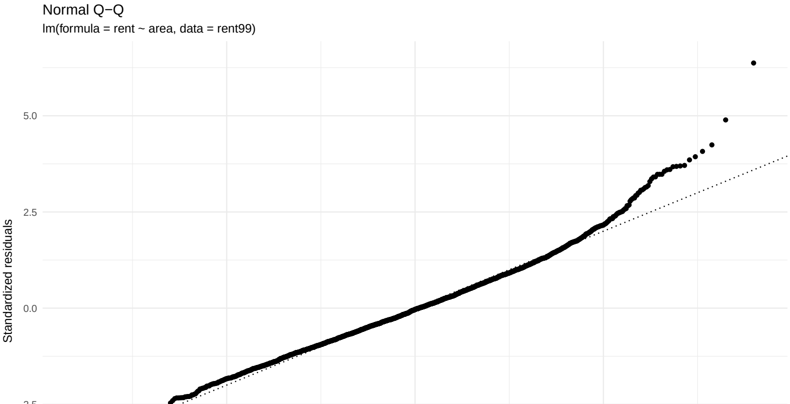


A: Ok linear assumption, but not constant spread.

Normal Q-Q

This plot shows if the residuals are Gaussian (normally) distributed. If they follow a straigt line it is an indication that they are, and else they are probably not.

```r
ggplot(fit, aes(sample = .stdresid)) + stat_qq(pch = 19) + geom_abline(intercept
    slope = 1, linetype = "dotted") + labs(x = "Theoretical quantiles",
    y = "Standardized residuals", title = "Normal Q-Q", subtitle = deparse(fit$
    theme_minimal()
```

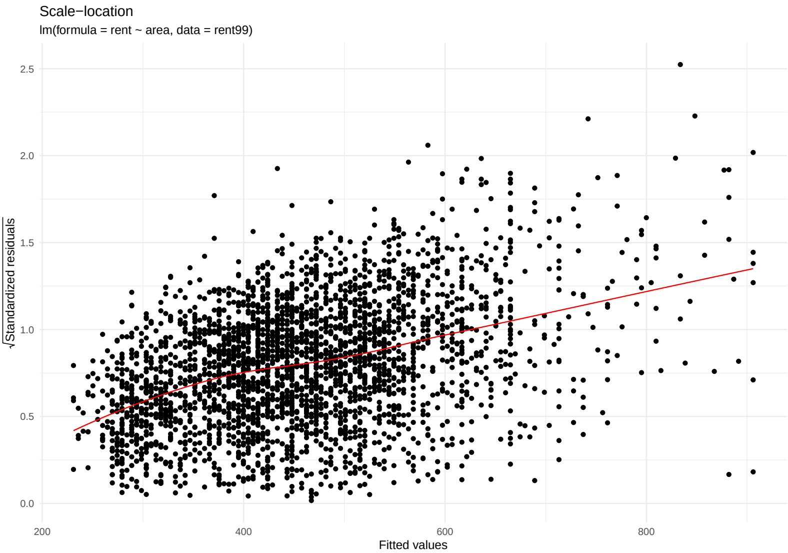


A: Not normal.

Scale-location

This is also called spread-location plot. It shows if the residuals are spread equally along the ranges of predictors. Can be used to check the assumption of equal variance (homoscedasticity). A good plot is one with a horizontal line with randomly spread points.
Is this plot good for your data?

```r
ggplot(fit, aes(.fitted, sqrt(abs(.stdresid)))) + geom_point() + geom_smooth(se
    col = "red", size = 0.5, method = "loess") + labs(x = "Fitted values",
    y = expression(sqrt("Standardized residuals")), title = "Scale-location",
    subtitle = deparse(fit$call)) + theme_minimal()
```



A: Confirms our observation of not constant variance.

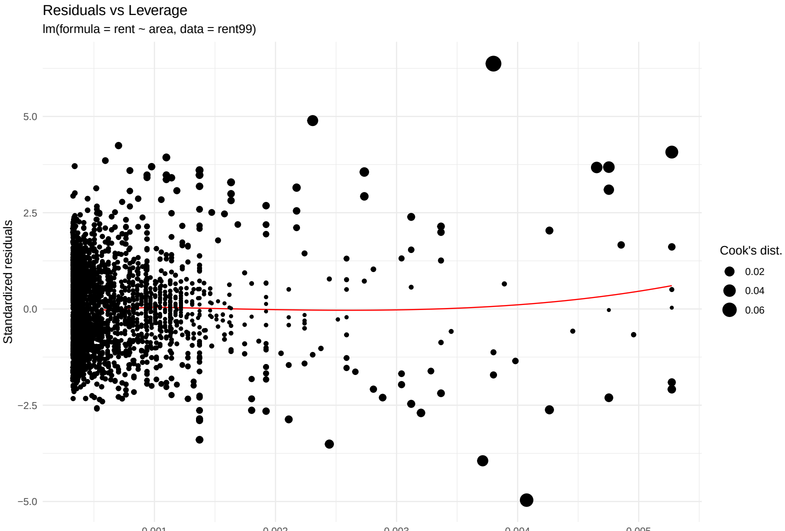
Residual vs Leverage

This plot can reveal influential outliers. Not all outliers are influential in linear regression; even though data have extreme values, they might not be influential to determine the regression line (the results don't differ much if they are removed from the data set). These influential outliers can be seen as observations that does not get along with the trend in the majority of the observations. In `plot.lm`, dashed lines are used to indicate the Cook's distance, instead of using the size of the dots as is done here.

Cook's distance is the Euclidean distance between the $\hat{\mathbf{y}}$ (the fitted values) and $\hat{\mathbf{y}}_{(i)}$ (the fitted values calculated when the i-th observation is omitted from the regression). This is then a measure on how much the model is influences by observation i. The distance is scaled, and a rule of thumb is to examine observations with Cook's distance larger than 1, and give some attention to those with Cook's distance above 0.5.

Leverage is defined as the diagonal elements of the hat matrix, i.e., the leverage of the i-th data point is h_{ii} on the diagonal of $\mathbf{H} = \mathbf{X}(\mathbf{X^T X})^{\mathbf{-1}}\mathbf{X^T}$. A large leverage indicated that the observation (i) has a large influence on the estimation results, and that the covariate values $(\mathbf{x}_i)$ are unusual.

```
ggplot(fit, aes(.hat, .stdresid)) + geom_smooth(se = FALSE, col = "red",
    size = 0.5, method = "loess") + geom_point(aes(size = .cooksd)) +
    scale_size_continuous("Cook's dist.") + labs(x = "Leverage", y = "Standardiz
    title = "Residuals vs Leverage", subtitle = deparse(fit$call)) +
    theme_minimal()
```



Residuals vs Leverage

lm(formula = rent ~ area, data = rent99)

A:Some observations does not fit our model, but if we fit a more complex model this may change.

Extensions and challenges in multiple regression

The section is a self study section, where the dummy variable part is the most important and will be used in this course.

Qualitative covariates

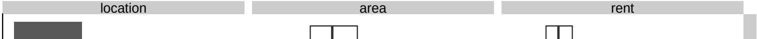
See Rob Tibshirani explains - from ca 9 minutes

Qualitative predictors can be included in a linear regression model by introducing dummy variables

Example: consider our `rent` dataset with `rent` as response, and continuous covariate `area` and categorical covariate `location`. Let the `location` be a factor with levels `average, good, top`.

```r
library(gamlss.data)
library(dplyr)
library(GGally)

ds = dplyr::select(rent99, location, area, rent)
levels(ds$location)
# change to meaningful names
levels(ds$location) = c("average", "good", "top")
ggpairs(ds)
```



Categorical covariates may either be ordered or unordered. We will only consider unordered categories here. In general, we could like to estimate regression coefficients for all levels for the categorical covariates. However, if we want to include an intercept in our model we can only include codings for one less variable than the number of levels we have - or else our design matrix will not have full rank.

Q: Assume you have a categorical variable with three levels. Check for yourself that making a design matrix with one intercept and three columns with dummy (0-1) variable coding will result in a matrix that is singular.

```r
# make 'wrong' dummy variable coding with 3 columns
n = length(ds$location)
X = cbind(rep(1, n), ds$area, rep(0, n), rep(0, n), rep(0, n))
X[ds$location == "average", 3] = 1
X[ds$location == "good", 4] = 1
X[ds$location == "top", 5] = 1
X[c(1, 3, 69), ]
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1   26    0    1    0
## [2,]    1   30    1    0    0
## [3,]    1   55    0    0    1
```

```r
require(Matrix)
dim(X)
```

```
## [1] 3082    5
```

```r
rankMatrix(X)
```

```
## [1] 4
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
```

This is why we need to instead work with different ways of coding categorical variables. One solution is to not include an intercept in the model, but that is often not what we want. We will look at two other solutions - one where we decide on a reference category (that we not include in the coding, and therefore is kind of included in the intercept - this is called “treatment coding”) and one where we require that the the sum of the coeffisients are zero (called “effect coding”). This mainly effects how we interpret parameter estimates and communicate our findings to the world. We will here restrict our discussion to “treatment coding”.

If we fit a regression model with `lm` to the data with `rent` as response and `area` and `location` as covariates, a model matrix is made `-` and how to handle the categorical variable is either specified the call to `lm` in `contrasts=list(location="contr.treatment")` (or to model.matrix) or globally for all categorical variables with `options(contrasts=c("contr.treatment","contr.poly"))-` where first element give choice for unordered factor (then treatment contrast is default) and second for ordered (and then this polynomial contrast is default). We will only work with unordered factors now.

Dummy variable coding

This is the default coding. The reference level is automatically chosen as the “lowest” level (sorted alphabetically). For our example this means that the reference category for location is “average”.

$$x_{i\text{locationgood}} = \begin{cases} 1 \text{ if } i \text{ -th location} = \text{"good"} \\ 0 \text{ if } i \text{ -th location } \neq \text{ "good"} \end{cases}$$

$$x_{i\text{locationtop}} = \begin{cases} 1 \text{ if } i \text{ -th location} = \text{"top"} \\ 0 \text{ if } i \text{ -th location } \neq \text{ "top"} \end{cases}$$

$$y_i = \beta_0 + \beta_1 x_{i\text{area}} + \beta_2 x_{i\text{locationgood}} + \beta_3 x_{i\text{locationtop}} + \varepsilon_i$$

$$= \begin{cases} \beta_0 + \beta_1 x_{i\text{area}} + \beta_2 + \varepsilon_i & \text{if } i \text{ -th location} = \text{"good"} \\ \beta_0 + \beta_1 x_{i\text{area}} + \beta_3 + \varepsilon_i & \text{if } i \text{ -th location} = \text{"top"} \\ \beta_0 + \varepsilon_i & \text{if } i \text{ -th location} = \text{"average"} \end{cases}$$

If we instead wanted “good” to be reference category we could relevel the factor.

```
X1 = model.matrix(~area + location, data = ds)
X1[c(1, 3, 69), ]
```

```
##    (Intercept) area locationgood locationtop
## 1            1   26            1           0
## 3            1   30            0           0
## 69           1   55            0           1
```

```
ds$locationRELEVEL = relevel(ds$location, ref = "good")
X2 = model.matrix(~area + locationRELEVEL, data = ds)
X2[c(1, 3, 69), ]
```

```
##    (Intercept) area locationRELEVELaverage locationRELEVELtop
## 1            1   26                      0                  0
## 3            1   30                      1                  0
## 69           1   55                      0                  1
```

So, what does this mean in practice? Model 1 has **average** as reference category and model 2 **good**.

```
fit1 = lm(rent ~ area + location, data = ds, contrasts = list(lo
summary(fit1)
```

```
##
## Call:
## lm(formula = rent ~ area + location, data = ds, contrasts = l
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -790.98 -100.89   -4.87   94.47 1004.98
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  128.0867     8.6947  14.732  < 2e-16 ***
## area           4.7056     0.1202  39.142  < 2e-16 ***
## locationgood  28.0040     5.8662   4.774 1.89e-06 ***
## locationtop  131.1075    18.2614   7.180 8.73e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
##
## Residual standard error: 157.1 on 3078 degrees of freedom
```

Interactions

See Trevor Hastie explains

To illustrate how interactions between covariates can be included we use the ozone data set from the ElemStatLearn library. This data set is measurements from 1973 in New York and contains 111 observations of the following variables:

- ozone : ozone concentration (ppm)
- radiation : solar radiation (langleys)
- temperature : daily maximum temperature (F)
- wind : wind speed (mph)

We start by fitting a multiple linear regression model to the data, with ozone as our response variable and temperature and wind as covariates.

```
##   ozone radiation temperature wind
## 1    41       190          67  7.4
## 2    36       118          72  8.0
## 3    12       149          74 12.6
## 4    18       313          62 11.5
## 5    23       299          65  8.6
## 6    19        99          59 13.8
```

```
##
## Call:
## lm(formula = ozone ~ temperature + wind, data = ozone)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.160 -13.209  -3.089  10.588  98.470
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -67.2008    23.6083  -2.846  0.00529 **
## temperature   1.8265     0.2504   7.293 5.32e-11 ***
## wind         -3.2993     0.6706  -4.920 3.12e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.72 on 108 degrees of freedom
## Multiple R-squared:  0.5817, Adjusted R-squared:  0.574
## F-statistic:  75.1 on 2 and 108 DF,  p-value: < 2.2e-16
```

The model can be written as:

$$Y = \beta_0 + \beta_1 x_t + \beta_2 x_w + \varepsilon$$

In this model we have assumed that increasing the value of one covariate is independent of the other covariates. For example: by increasing the `temperature` by one-unit always increases the response value by $\beta_2 \approx 1.651$, regardless of the value of `wind`.

However, one might think that the covariate `wind` (wind speed) might act differently upon `ozone` for different values of `temperature` and vice verse.

$$\begin{aligned} Y &= \beta_0 + \beta_1 x_t + \beta_2 x_w + \beta_3 \cdot (x_t \cdot x_w) + \varepsilon \\ &= \beta_0 + (\beta_1 + \beta_3 x_w) \cdot x_t + \beta_2 x_w + \varepsilon \qquad . \\ &= \beta_0 + \beta_1 x_t + (\beta_2 + \beta_3 x_t) \cdot x_w + \varepsilon \end{aligned}$$

We fit this model in `R`. An interaction term can be included in the model using the `*` symbol.

Q: Look at the `summary` below. Is this a better model than without the interaction term? It the term significant?

```r
ozone.int = lm(ozone ~ temperature + wind + temperature * wind, data = ozone)
summary(ozone.int)
```

```
##
## Call:
## lm(formula = ozone ~ temperature + wind + temperature * wind,
##     data = ozone)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.929 -11.190  -3.037   8.209  97.440
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -239.94146   48.59004  -4.938 2.92e-06 ***
## temperature         4.00151    0.59311   6.747 8.02e-10 ***
## wind               13.60882    4.28070   3.179  0.00193 **
## temperature:wind   -0.21747    0.05446  -3.993  0.00012 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.36 on 107 degrees of freedom
## Multiple R-squared:  0.636,  Adjusted R-squared:  0.6258
## F-statistic: 62.31 on 3 and 107 DF,  p-value: < 2.2e-16
```

Below we see that the interaction term is highly significant. The p-value is very small, so that there is strong evidence that $\beta_3 \neq 0$. Furthermore, R^2_{adj} has increased, indicating that more of the variability in the data has been explained by the model (than without the interaction).

Interpretation of the interaction term:

- If we now increase the `temperature` by 10° F, the increase in `wind` speed will be

$$(\hat{\beta}_1 + \hat{\beta}_3 \cdot x_w) \cdot 10 = (4.0 - 0.22 \cdot x_w) \cdot 10 = 40 - 2.2x_w \text{ units.}$$

- If we increase the `wind` speed by 10 mph, the increase in `temperature` will be

$$(\hat{\beta}_2 + \hat{\beta}_3 \cdot x_t) \cdot 10 = (14 - 0.22 \cdot x_t) \cdot 10 = 140 - 2.2x_t \text{ units.}$$

The hierarchical principle

It is possible that the interaction term is higly significant, but the main effects are not.

In our `ozone.int` model above: the main effects are `temperature` and `wind`. The hierarchical principle states that if we include an interaction term in our model, the main effects are also to be included, even if they are not significant. This means that if the coefficients $\hat{\beta}_1$ or $\hat{\beta}_2$ would be insignificant, while the coefficient $\hat{\beta}_3$ is significant, $\hat{\beta}_1$ and $\hat{\beta}_2$ should still be included in the model.

There reasons for this is that a model with interaction terms, but without the main effects is hard to interpret.

Interactions between qualitative (discrete) and quantitative (continuous) covariates

We create a new variable `temp.cat` which is a `temperature` as a qualitative covariate with two levels and fit the model:

$$y = \beta_0 + \beta_1 x_w + \begin{cases} \beta_2 + \beta_3 x_w & \text{if temperature=''low''} \\ 0 & \text{if temperature = ''high''} \end{cases}$$

$$= \begin{cases} (\beta_0 + \beta_2) + (\beta_1 + \beta_3) \cdot x_w & \text{if temperature=''low''} \\ \beta_0 + \beta_1 x_w & \text{if temperature=''high''''} \end{cases}$$

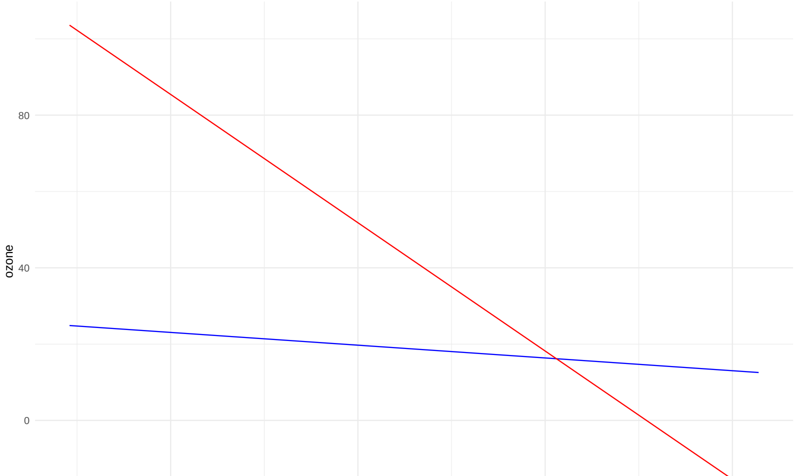
```
temp.cat = ifelse(ozone$temperature < mean(ozone$temperature), "low",
    "high")
ozone2 = cbind(ozone, temp.cat)
print(head(ozone2))
```

```
##   ozone radiation temperature wind temp.cat
## 1    41       190          67  7.4      low
## 2    36       118          72  8.0      low
## 3    12       149          74 12.6      low
## 4    18       313          62 11.5      low
## 5    23       299          65  8.6      low
## 6    19        99          59 13.8      low
```

```
ozone.int2 = lm(ozone ~ wind + temp.cat + temp.cat * wind, data = ozone2)
summary(ozone.int2)
```

```
##
## Call:
## lm(formula = ozone ~ wind + temp.cat + temp.cat * wind, data = ozone2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.291  -9.091  -1.307  11.227  71.815
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
```

```
interceptlow = coef(ozone.int2)[1] + coef(ozone.int2)[3]
slopelow = coef(ozone.int2)[2] + coef(ozone.int2)[4]
intercepthigh = coef(ozone.int2)[1]
slopehigh = coef(ozone.int2)[2]
ggplot(ozone) + geom_line(aes(y = interceptlow + slopelow * wind, x = wind),
    col = "blue") + geom_line(aes(y = intercepthigh + slopehigh * wind,
    x = wind), col = "red") + ylab("ozone") + theme_minimal()
```



Important results in multiple linear regression

- Linear regression assumes a linear relationship between the response variable and the covariates.
- Simple linear regression has only one covariate and has the form $Y = \beta_0 + \beta_1 X + \varepsilon$.
- Muliple linear regression has p covariates and has the form $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \varepsilon$.
- Quantitative (categorical) covariates can be included using dummy variables.
- Correlations among the covariates can mask each others effects in the linear model.
- Parameter estimates can be obtained by minimizing the least squares (RSS) or by maximum likelihood estimation.
- We can calculate the standard errors of the parameter estimates, and use this to obtain confidence intervals.
- We can test the hypothesis of $H_0 : \beta_j = 0$ against $H_1 : \beta_j \neq 0$ by a t-test.
- We use the F-statistic to test if at least one of the covariates are useful.

- Not only additive effects: Interactions between covariates can be included in the model (also between qualitative and quantitative covariates).
- Transformations of the response variable or of a covariate can be useful if the relationship is not linear. A linear model can then be fit to the transformed variables.
- The overall accuracy of the model can be evaluated by calculating the R^2 statistic, AIC score and by using diagnostic plots.
- Model selection can not be based on RSS or R^2.
- Multiple linear regression might require subset selection if the number of covariates is high.

Further reading

- Need details on the simple linear regression: From TMA4240/TMA4245 Statistics we have the thematic page for Simple linear regression (in Norwegian).
- Need more advanced thory: Theoretical version (no simple linear regression) from TMA4315 Generalized linear models H2018: TMA4315M2: Multiple linear regression
- Slightly different presentation (more focus on multivariate normal theory): Slides and written material from TMA4267
- And, same source, but now Slides and written material from TMA4267 Linear Statistical Models in 2017, Part 3: Hypothesis testing and ANOVA
- Videoes on YouTube by the authors of ISL, Chapter 2

R packages

If you want to look at the .Rmd file and `knit` it, you need to first install the following packages (only once).

```
# packages to install before knitting this R Markdown file to kn
# the Rmd
install.packages("knitr")
install.packages("rmarkdown")

# nice tables in Rmd
install.packages("kableExtra")

# cool layout for the Rmd
install.packages("prettydoc")  # alternative to github

# plotting
install.packages("ggplot2")  # cool plotting
install.packages("ggpubr")  # for many ggplots
install.packages("GGally")  # for ggpairs
# datasets
install.packages("ElemStatLearn")  # for ozone data set
```

Acknowledgements

Thanks to Julia Debik for contributing to this module page.