

Kurs Bio144:

Datenanalyse in der Biologie

Stefanie Muff & Owen L. Petchey

Lecture 7: ANCOVA, short introduction to Linear Algebra
15. April 2019

Overview

- ANCOVA
- ANCOVA as special cases of a linear model
- Introduction to linear Algebra

Note:

ANCOVA = ANalysi of COVAriance (Kovarianzanalyse)

Course material covered today

The lecture material of today is based on the following literature:

- “Getting Started with R” chapter 6.3
- “Lineare regression” chapters 3.A (p. 43-45) and 3.4, 3.5 (p. 39-42)

Recap of ANOVA

- ANOVA is a method to test if the means of **two or more groups are different**.
- Post-hoc tests and contrasts, including correction for p -values, to understand the differences between the groups.
- Two-way ANOVA for factorial designs, interactions.
- Interpretation of the results.
- Model checking.
- ANOVA is a special case of linear regression with categorical covariates.

Analysis of Covariance (ANCOVA)

An ANCOVA is an analysis of variance (ANOVA), including also at least one continuous covariate.

→ ANCOVA is a combination of ANOVA and regression, so in principle, there is nothing new to it...

Given a categorical covariate x_i and a continuous covariate z_i . Then the ANCOVA equation (without interactions) is given as

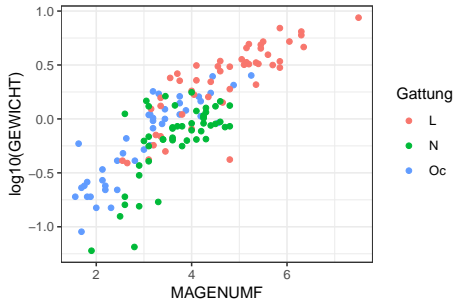
$$y_i = \beta_0 + \beta_1 x_i^{(1)} + \dots + \beta_k x_i^{(k)} + \beta_z z_i + e_i ,$$

where $x_i^{(k)}$ is the k th dummy variable ($x_i^{(k)}=1$ if i th observation belongs to category k , 0 otherwise).

Note 1: It is straightforward to add an interaction of x_i with z_i .

Note 2: Again, for identifiability reason, we typically set $\beta_1 = 0$.

Example: Remember the earthworm study from week 3:
“Magenumfang” was used to predict “Gewicht” of the worm, including as covariate also the worm species.



Categorical and **continuous** covariates were used to predict a continuous outcome (the weight of the worm).

```
> r.lm <- lm(log(GEWICHT) ~ MAGENUMF + Gattung,d.wurm)
> summary(r.lm)$coef
```

```

              Estimate Std. Error    t value    Pr(>|t|)
(Intercept) -2.5355459  0.22147279 -11.4485663 8.617670e-22
MAGENUMF      0.7118725  0.04528843  15.7186392 1.232126e-32
GattungN      -0.5151344  0.11009219  -4.6791186 6.760621e-06
GattungOc     -0.0907298  0.12791000  -0.7093254 4.793107e-01
```

Remember: What do the p -values ($p = 0.48$ and $p < 0.0001$) of the categorical covariate “Gattung” mean?

To understand if “Gattung” has an effect, **we need to carry out an F -test** (see slides 46/47, week 3). To this end, plot the ANOVA table:

```
> anova(r.lm)
```

Analysis of Variance Table

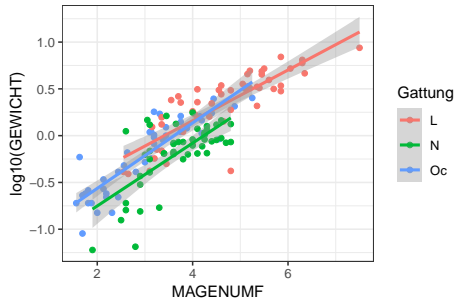
Response: log(GEWICHT)

```

      Df Sum Sq Mean Sq F value    Pr(>F)
MAGENUMF    1  104.866   104.866   409.69 < 2.2e-16 ***
Gattung      2    7.177    3.589    14.02 2.842e-06 ***
Residuals  139   35.579    0.256
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Remember that we can also include an **interaction** term between MAGENUMF and Gattung to allow for different slopes:



The **F-test** is also needed to check whether the respective interaction term is needed:

```
> r.lm2<- lm(log(GEWICHT) ~ MAGENUMF * Gattung,d.wurm)
> summary(r.lm2)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-2.1274390	0.30859958	-6.893849	1.816840e-10
MAGENUMF	0.6237919	0.06484623	9.619555	4.705940e-17
GattungN	-1.1510597	0.49399105	-2.330123	2.126052e-02
GattungOc	-0.7810625	0.39669094	-1.968945	5.097754e-02
MAGENUMF:GattungN	0.1500462	0.12178262	1.232082	2.200288e-01
MAGENUMF:GattungOc	0.1817710	0.10199782	1.782106	7.694701e-02

```
> anova(r.lm2)
```

Analysis of Variance Table

Response: log(GEWICHT)

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
MAGENUMF	1	104.866	104.866	414.4743	< 2.2e-16 ***
Gattung	2	7.177	3.589	14.1835	2.521e-06 ***
MAGENUMF:Gattung	2	0.917	0.458	1.8112	0.1673
Residuals	137	34.662	0.253		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

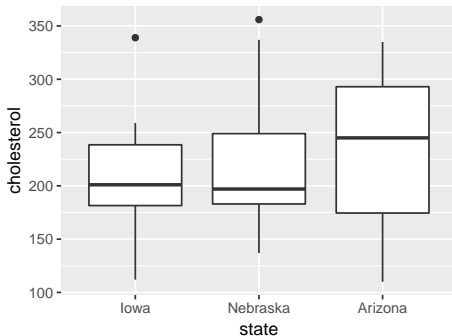
→ $p = 0.167$, thus interaction is probably not relevant.

A new example: cholesterol levels

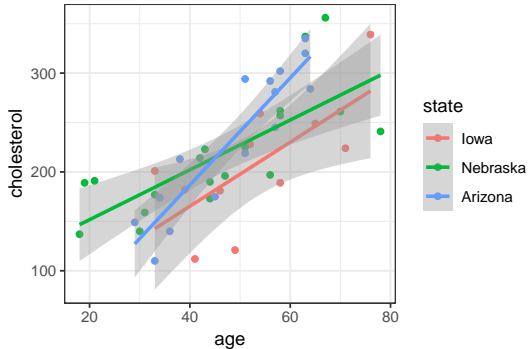
Example: Cholesterol levels [mg/ml] for 45 women from three US states (Iowa, Nebraska, Arizona), were measured.

Question: Do these levels differ between the states?

Age (years) may be a relevant covariable.



The scatter plot gives an idea about the model that might be useful here:



→ We include state, age and the interaction of the two.

Doing the analysis:

```
> r.lm <- lm(cholesterol ~ age*state,data=d.chol)
> anova(r.lm)
```

Analysis of Variance Table

Response: cholesterol

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
age	1	96524	96524	61.8961	1.424e-09 ***
state	2	11474	5737	3.6789	0.03438 *
age:state	2	12665	6332	4.0606	0.02501 *
Residuals	39	60819	1559		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Interpretation?

Compare the results from the previous slide to the estimated coefficients:

```
> r.lm <- lm(cholesterol ~ age*state,data=d.chol)
> summary(r.lm)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	35.8112138	50.4494150	0.7098440	0.482023781
age	3.2381449	0.9234015	3.5067573	0.001158119
stateNebraska	65.4865523	56.7347108	1.1542590	0.255418196
stateArizona	-65.7063829	66.7677031	-0.9841043	0.331130339
age:stateNebraska	-0.7177069	1.0643772	-0.6742975	0.504099737
age:stateArizona	2.1787633	1.2672928	1.7192264	0.093502039

Note: The p -values for the age coefficient is not the same as in the anova table...

Compare the results for “state” also to the model without interaction:

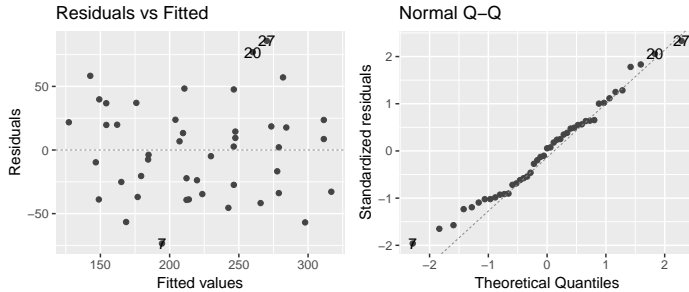
```
> r.lm2 <- lm(cholesterol ~ age ,data=d.chol)
> summary(r.lm2)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	68.247111	22.830703	2.989269	4.610215e-03
age	3.134982	0.448522	6.989584	1.332242e-08

Note: the p -value for ‘state’ is similar to the previous slide (anova table).

Reason: anova tests the models against one another in the **order** specified.

As always, some model checking is necessary:



→ This seems ok.

A final word on ANCOVA

ANCOVA unifies several concepts that we approached in this course so far:

- Linear regression
- Categorical covariates
- Interactions (of continuous and categorical covariates)
- Analysis of Variance (ANOVA)

As such, it is a **special case of the linear model**.

An introduction to linear Algebra

Who has some knowledge of linear Algebra?

Overview

- The basics about
 - vectors
 - matrices
 - matrix algebra
 - matrix multiplication
- Why is linear Algebra useful?
- What does it have to do with data analysis and statistics?
- Regression equations in matrix notation.

Motivation

Why are vectors, matrices and their algebraic rules useful?

Example 1: The observations for a covariate \mathbf{x} or the response \mathbf{y} for all individuals $1 \leq i \leq n$ can be stored in a vector (vectors and matrices are always given in **bold** letters):

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}.$$

Example 2: Covariance matrices for multiple variables. Say we have $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$. The **covariance matrix** is then given as

$$\begin{pmatrix} \text{Var}(\mathbf{x}^{(1)}) & \text{Cov}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \\ \text{Cov}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) & \text{Var}(\mathbf{x}^{(2)}) \end{pmatrix}.$$

Example 3: The **data** (e.g. of some regression model) can be stored in a **matrix**:

$$\tilde{\mathbf{X}} = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_2^{(1)} & x_2^{(2)} \\ \dots & \dots & \dots \\ 1 & x_n^{(1)} & x_n^{(2)} \end{pmatrix} .$$

This is the so-called **design matrix** with a vector of 1's in the first column.

Example 4: A linear regression model can be written compactly using **matrix multiplication**:

$$\mathbf{y} = \tilde{\mathbf{X}} \cdot \tilde{\boldsymbol{\beta}} + \mathbf{e} ,$$

with $\tilde{\boldsymbol{\beta}}$ the vector of regression coefficients and \mathbf{e} the vector of residuals.

Why do we discuss this topic in our course?

- Useful for **compact notation**.
- Enables you to **understand many statistical texts** (books, research articles) that remain inaccessible otherwise.
- Useful for **efficient coding**, e.g. in R, which helps to increase speed and to reduce error rates.
- More advanced concepts often rely on linear algebra, e.g. **principal component analysis** (PCA) or **random effects** models.
- Is part of a **general education** (Allgemeinbildung) ;-)

Matrices

An $n \times m$ **Matrix** is given as

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix},$$

with rows $i = 1, \dots, n$ and columns $j = 1, \dots, m$.

Quadratic matrix: $n = m$. Example:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 3 & 2 \\ 6 & 1 & 9 \end{pmatrix}$$

Symmetric matrix: $a_{ij} = a_{ji}$. Example:

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix} .$$

The diagonal of a quadratic matrix is given by $(a_{11}, a_{22}, \dots, a_{nn})$.

Example: the diagonal of the above matrix is given as

$$(a_{11}, a_{22}, a_{33}) = (1, 3, 5) .$$

Diagonal matrix: A matrix that has entries $\neq 0$ **only on the diagonal**.

Example:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{pmatrix} .$$

Transposing a matrix: Given a matrix **A**. Exchange the rows by the columns and vice versa. This leads to the **transposed matrix** \mathbf{A}^\top :

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix} \Rightarrow \mathbf{A}^\top = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \vdots & \vdots & & \vdots \\ a_{1m} & a_{2m} & \dots & a_{nm} \end{pmatrix}.$$

Examples (note also the change of dimensions):

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \Rightarrow \mathbf{A}^\top = \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix} \Rightarrow \mathbf{A}^\top = \begin{pmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{pmatrix}$$

- Transposing a matrix **twice** leads to the original matrix:

$$(\mathbf{A}^\top)^\top = \mathbf{A} .$$

- When a matrix is **symmetric**, then

$$\mathbf{A}^\top = \mathbf{A} .$$

This is true in particular for diagonal matrices.

Vectors

A vector is nothing else than n numbers written in a column:

$$\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Transposing a vector leads to a *row vector*:

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}^{\top} = \begin{pmatrix} b_1 & b_2 & \dots & b_n \end{pmatrix}$$

Note: By definition (by default), a vector is always a column vector.

Addition and subtraction

- Adding and subtracting matrices and vectors is only possible when the objects have the **same dimension**.
- Examples: Elementwise addition (or subtraction)

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} + \begin{pmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \end{pmatrix} = \begin{pmatrix} 4 & 4 & 4 \\ 10 & 10 & 10 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 9 \end{pmatrix} = \begin{pmatrix} -2 \\ -5 \end{pmatrix}$$

- But this addition is **not defined**:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} + \begin{pmatrix} 3 & 6 \\ 2 & 5 \\ 1 & 4 \end{pmatrix} =$$

Multiplication by a scalar

Multiplication with a “number” (scalar) is simple: Multiply each element in a vector or a matrix.

Examples:

$$3 \cdot \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} = \begin{pmatrix} 3 & 6 & 9 \\ 12 & 15 & 18 \end{pmatrix}$$

$$-2 \cdot \begin{pmatrix} 1 \\ 4 \\ -2 \end{pmatrix} = \begin{pmatrix} -2 \\ -8 \\ 4 \end{pmatrix}$$

Matrix multiplication

The multiplication of two matrices **A** and **B** is **defined if**
number of columns in **A** = number of rows in **B**.

It is easiest to explain matrix multiplication with an example:

$$\begin{pmatrix} 2 & 1 \\ -1 & 0 \\ 3 & 1 \end{pmatrix} \cdot \begin{pmatrix} 3 & 1 \\ 4 & -2 \end{pmatrix} = \begin{pmatrix} 2 \cdot 3 + 1 \cdot 4 & 2 \cdot 1 + 1 \cdot (-2) \\ (-1) \cdot 3 + 0 \cdot 4 & (-1) \cdot 1 + 0 \cdot (-2) \\ 3 \cdot 3 + 1 \cdot 4 & 3 \cdot 1 + 1 \cdot (-2) \end{pmatrix}$$
$$= \begin{pmatrix} 10 & 0 \\ -3 & -1 \\ 13 & 1 \end{pmatrix}$$

► Matrix multiplication app

Matrix multiplication rules I

Attention: Matrix multiplication does **not** follow the same rules as scalar multiplication!!!

- It can happen that $\mathbf{A} \cdot \mathbf{B}$ can be calculated, but $\mathbf{B} \cdot \mathbf{A}$ is not defined (see example on previous slide).
- In general: $\mathbf{A} \cdot \mathbf{B} \neq \mathbf{B} \cdot \mathbf{A}$, even if both are defined.
- It can happen that $\mathbf{A} \cdot \mathbf{B} = \mathbf{0}$ (0 matrix), although both $\mathbf{A} \neq \mathbf{0}$ and $\mathbf{B} \neq \mathbf{0}$.
- The **Assoziativgesetz** holds: $\mathbf{A} \cdot (\mathbf{B} \cdot \mathbf{C}) = (\mathbf{A} \cdot \mathbf{B}) \cdot \mathbf{C}$.
- The **Distributivgesetz** holds:

$$\mathbf{A} \cdot (\mathbf{B} + \mathbf{C}) = \mathbf{A} \cdot \mathbf{B} + \mathbf{A} \cdot \mathbf{C}$$

$$(\mathbf{A} + \mathbf{B}) \cdot \mathbf{C} = \mathbf{A} \cdot \mathbf{C} + \mathbf{B} \cdot \mathbf{C}$$

Matrix multiplication rules II

- Transposing inverts the order: $(\mathbf{A} \cdot \mathbf{B})^\top = \mathbf{B}^\top \cdot \mathbf{A}^\top$.
- The product $\mathbf{A} \cdot \mathbf{A}^\top$ is **always symmetric**.
- All these rules also hold for **vectors**, which can be interpreted as $n \times 1$ matrices:

$$\mathbf{a} \cdot \mathbf{b}^\top = \begin{pmatrix} a_1 b_1 & a_1 b_2 & \dots & a_1 b_m \\ a_2 b_1 & a_2 b_2 & \dots & a_2 b_m \\ \vdots & \vdots & & \vdots \\ a_n b_1 & a_n b_2 & \dots & a_n b_m \end{pmatrix}$$

If \mathbf{a} and \mathbf{b} have the **same length**:

$$\mathbf{a}^\top \cdot \mathbf{b} = \sum_i a_i b_i$$

Short exercises

Given vectors **a** and **b** and matrix **C**:

$$\mathbf{a} = \begin{pmatrix} 1 \\ -2 \\ 3 \\ 0 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} -2 \\ 4 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 1 & 0 \\ -1 & 2 \end{pmatrix}$$

Calculate, if defined

- $\mathbf{a}^T \cdot \mathbf{b}$
- $\mathbf{a} \cdot \mathbf{b}^T$
- $\mathbf{C} \cdot \mathbf{a}$
- $\mathbf{C} \cdot \mathbf{b}$

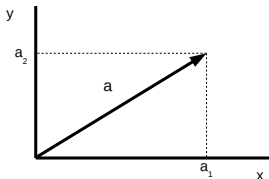
The length of a vector

The **length of a vector** $\mathbf{a}^T = (a_1, a_2, \dots, a_n)$ is defined as $\|\mathbf{a}\|$ with

$$\|\mathbf{a}\|^2 = \mathbf{a}^T \cdot \mathbf{a} = \sum_i a_i^2.$$

This is basically the **Pythagoras** idea in 2, 3, ... n dimensions.

In 2 dimensions: $\|\mathbf{a}\| = \sqrt{a_1^2 + a_2^2}$:



Identity matrix (Einheitsmatrix)

The identity matrix (of dimension m) is probably the simplest matrix that exists. It has 1's on the diagonal and 0's everywhere else:

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

Multiplication with the identity matrix leaves a $m \times n$ matrix \mathbf{A} unchanged:

$$\mathbf{A} \cdot \mathbf{I} = \mathbf{A} .$$

Inverse matrix

Given a quadratic matrix \mathbf{A} that fulfills

$$\mathbf{B} \cdot \mathbf{A} = \mathbf{I} ,$$

then \mathbf{B} is called the **inverse** of \mathbf{A} (and vice versa). One then writes

$$\mathbf{B} = \mathbf{A}^{-1} .$$

Note:

- In that case it also holds that $\mathbf{A} \cdot \mathbf{B} = \mathbf{I}$.
- Therefore: $\mathbf{A} = \mathbf{B}^{-1} \Leftrightarrow \mathbf{B} = \mathbf{A}^{-1}$

- The inverse of \mathbf{A} may **not exist**. If it exists, \mathbf{A} is **regular**, otherwise **singular**.

- $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$.

- The inverse of a matrix product is given as

$$(\mathbf{A} \cdot \mathbf{B})^{-1} = \mathbf{B}^{-1} \cdot \mathbf{A}^{-1} .$$

- It is

$$(\mathbf{A}^{\top})^{-1} = (\mathbf{A}^{-1})^{\top} .$$

Therefore one may also write $\mathbf{A}^{-\top}$.

Linear regression in matrix notation

Linear regression with n data points can be understood as an **equation system with n equations**.

Remember example 4 from slide ?? : We said that a linear regression model can be written compactly using **matrix multiplication**:

$$\mathbf{y} = \tilde{\mathbf{X}} \cdot \tilde{\boldsymbol{\beta}} + \mathbf{e} .$$

Task: Verify this now, using a model with two variables $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ and

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \tilde{\mathbf{X}} = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_2^{(1)} & x_2^{(2)} \\ \dots & \dots & \dots \\ 1 & x_n^{(1)} & x_n^{(2)} \end{pmatrix}, \quad \tilde{\boldsymbol{\beta}} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix}, \quad \mathbf{e} = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix} .$$

It can be shown (see Stahel 3.4f,g) that the least-squares estimates $\hat{\beta}$ can be calculated as

$$\hat{\beta} = (\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}})^{-1} \cdot \tilde{\mathbf{X}}^\top \cdot \mathbf{y}$$

Does this look complicated?

Let's test this in R

Doing linear algebra in R

Let us look at model $\mathbf{y} = \tilde{\mathbf{X}} \cdot \tilde{\boldsymbol{\beta}} + \mathbf{e}$ with coefficients $\beta_0 = 10, \beta_1 = 5, \beta_2 = -2$ and variables

i	$x_i^{(1)}$	$x_i^{(2)}$
1	0	4
2	1	1
3	2	0
4	3	1
5	4	4

Thus the model is given as

$$y_i = 10 + 5x_i^{(1)} - 2x_i^{(2)} + e_i, \text{ for } 1 \leq i \leq n.$$

Let us start by generating the “true” response, calculated as $\tilde{\mathbf{X}}\tilde{\boldsymbol{\beta}}$:

```
> x1 <- c(0,1,2,3,4)
> x2 <- c(4,1,0,1,4)
> Xtilde <- matrix(c(rep(1,5),x1,x2),ncol=3)
> Xtilde
```

```
      [,1] [,2] [,3]
[1,]    1    0    4
[2,]    1    1    1
[3,]    1    2    0
[4,]    1    3    1
[5,]    1    4    4
```

```
> t.beta <- c(10,5,-2)
> t.y <- Xtilde%*%t.beta
> t.y
```

```
      [,1]
[1,]     2
[2,]    13
[3,]    20
[4,]    23
[5,]    22
```

Matrix multiplication in R is done by the `%*%` symbol.

Next, we generate the vector containing the $e_i \sim N(0, \sigma_e^2)$ with $\sigma_e^2 = 1$:

```
> t.e <- rnorm(5,0,1)
> t.e
[1] 0.7606833 -0.3257157 0.6830309 0.9070262 0.9342162
```

which we add to the “true” $\mathbf{y} = \tilde{\mathbf{X}}\tilde{\boldsymbol{\beta}}$ values, to obtain the “observed” values:

```
> t.Y <- t.y + t.e
> t.Y
      [,1]
[1,] 2.760683
[2,] 12.674284
[3,] 20.683031
[4,] 23.907026
[5,] 22.934216
```

It is now possible to fit the model with `lm`:

```
> r.lm <- lm(t.Y ~ x1 + x2)
> summary(r.lm)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	10.069826	0.5556231	18.12348	0.003030672
x1	5.157981	0.1866953	27.62780	0.001307540
x2	-1.896970	0.1577864	-12.02239	0.006847617

Alternatively, we can use formula

$$\hat{\beta} = (\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^\top \mathbf{y}$$

to find the parameter estimates:

```
> solve(t(Xtilde) %*% Xtilde) %*% t(Xtilde) %*% t.Y  
  
[1,]  
[1,] 10.069826  
[2,] 5.157981  
[3,] -1.896970
```

- `solve()` calculates the **inverse** (here the inverse of $\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}$).
- `t()` gives the **transposed** (here of $\tilde{\mathbf{X}}^\top$).

Task: Do this calculation by yourself and verify for each step that the dimensions of the matrices and the vector are indeed fitting, so that this expression is defined.

Appendix

Some R commands for matrix algebra

Reading vectors and a matrices into R:

```
> a <- c(1,2,3)
```

```
> a
```

```
[1] 1 2 3
```

```
> A <- matrix(c(1,2,3,4,5,6),byrow=T,nrow=2)
```

```
> B <- matrix(c(6,5,4,3,2,1),byrow=T,nrow=2)
```

```
> A
```

```
      [,1] [,2] [,3]  
[1,]    1    2    3  
[2,]    4    5    6
```

```
> B
```

```
      [,1] [,2] [,3]  
[1,]    6    5    4  
[2,]    3    2    1
```

Adding and subtracting:

```
> A + B
```

```
      [,1] [,2] [,3]  
[1,]    7    7    7  
[2,]    7    7    7
```

```
> A - B
```

```
      [,1] [,2] [,3]  
[1,]   -5   -3   -1  
[2,]    1    3    5
```

However, be careful, R sometimes does unreasonable things:

```
> A + a
```

```
      [,1] [,2] [,3]  
[1,]     2     5     5  
[2,]     6     6     9
```

What happened here??

Matrix multiplication:

```
> C <- A %*% t(B)
> C
```

```
      [,1] [,2]
[1,]   28  10
[2,]   73  28
```

```
> A%*%a
```

```
      [,1]
[1,]   14
[2,]   32
```

Matrix inversion (possible for quadratic matrices only):

```
> solve(C)
```

```
      [,1]      [,2]
[1,]  0.5185185 -0.1851852
[2,] -1.3518519  0.5185185
```

```
> C %*% solve(C)
```

```
      [,1] [,2]
[1,]    1    0
[2,]    0    1
```

Why does `solve(A)` or `solve(B)` not work?