Statistical analysis of repeated measurements with linear mixed models and R-package LMMstar
#1
Introduction to linear mixed models and analysis of single group studies.

Julie Forman and Brice Ozenne
Section of Biostatistics, University of Copenhagen

November 18, 2022

Contents

Sections marked with $\star$ contains a high amount of mathematical/technical details.

1 Introduction

This tutorial is the first in a series developed for the Ph.D. course Statistical Analysis of Correlated and Repeated Measurements for Health Researchers taught every fall at the Faculty of Health Science of the University of Copenhagen.

In this, we offer a crash course on linear mixed models for anyone planning to analyze quantitative repeated measurements data.

For a gentle introduction we exemplify single group studies. Similar analyses can be applied to study designs where repeated measurements on the same subjects have been made over e.g.:

- Pre-specified, follow-up times as in the exemplified case study.
- Pre-specified treatments in a fixed sequence cross-over study.
- Pre-specified locations of the body in a study with physiological correlation.

The analyses in the tutorial were made with the R-package `LMMstar` which we have created to make linear mixed models more readily applicable for the R-users among our students and health science collaborators.

A similar tutorial is available for making analyses with SAS statistical software.

1.1 How to use this tutorial

The tutorial is intended to function as a self-learning text. You are encouraged to track the analyses of the case study data while reading about them. To run the analyses you will need:

1. The R-packages `LMMstar`, `lattice`, and `psych` available from CRAN
2. The program file `gastricbypass.R` available with the tutorial.

If you want to view the case study data outside `R` or use a different software for analysis, you can use the text file `gastricbypass.txt`.

1.2 How to report problems with R-package `LMMstar`

Unexpected behavior of the LMMstar package can be reported to:

https://github.com/bozenne/LMMstar/issues.

Please include a minimal reproducible example[1] in your report, otherwise it is likely that we will not be able to identify and solve the issue.

We do NOT provide support for other packages or R programming in general.

[1] https://stackoverflow.com/help/minimal-reproducible-example

1.3 Cite us!

We have invested considerable time and effort in developing the `LMMstar`-package and the tutorial. To support future updates, we kindly ask that you cite our work when you use it in your research.

```
To cite the 'LMMstar' package in publications use:

  Brice Ozenne and Julie Forman (2022). LMMstar: Repeated
  measurement models for discrete times. R package version
  0.7.6.
```

To cite this tutorial use:

"Julie Forman and Brice Ozenne (2022): *Introduction to linear mixed models for repeated measurements data: Analysis of single group studies with R-package LMMstar*. Lecture notes. Faculty of Health Science. University of Copenhagen. URL `https://absalon.ku.dk/courses/47665`".

Note that, as of this moment, the tutorial has not been formally peer-reviewed. We expect that it will be revised in the future, but the core content should remain more or less the same.

1.4 Where to learn more

Lecture notes, exercises, and other material can be found at our course webpage:

`https://absalon.ku.dk/courses/47665`

If you want to learn more about **longitudinal data analysis in general**, a thorough introduction is given in the book Applied Longitudinal Analysis by Fitzmaurice, Laird & Ware (2012). Students at the University of Copenhagen have free access to the e-version at the Royal Library. Please note that, even though the book describes SAS programing, supplementary R-programs can be found at the book webpage: `https://content.sph.harvard.edu/fitzmaur/ala2e/`

If you want to learn more about **various mixed model analyses in R**, a detailed account is given in the book by Pinheiro & Bates (2006), who developed the `nlme`-package, Pinheiro et al. (2019).

Before engaging in analysis of repeated measurements data, we strongly recommend that you acquire **basic R programming skills**. An e-learning course, developed by associate professor Susanne Rosthøj at the Section of Biostatistics, can be found at: `http://r.sund.ku.dk/`

2 Case: The gastric bypass study.

The gastric bypass study, Jorsal et al. (2020), was a small single group follow-up study in which $n = 20$ obese subjects were followed prior to and after gastric bypass surgery.

Follow-up times: 3 months before surgery (baseline), 1 week before surgery, 1 week after surgery, and 3 months after surgery (end of follow-up).

Outcomes: Gene-expressions from mucosal biopsies, gut hormones from blood samples, and clinical data.

The primary aim of the study was to describe changes in the gut profile in response to the treatment. For complete recording clinical outcomes were also analyzed. Please note that the subjects underwent a diet in preparation for the surgery. Hence, genuine physiological changes occurred already prior to surgery.

For simplicity, we will focus on two of the secondary outcomes from the study: bodyweight (kg) and serum glucagon (a gut hormone). The latter was assessed by a mixed meal tolerance test and summarized by total area under the curve (AUC, pmol/oz x minutes).

The motivation for picking these two particular outcomes is to highlight their statistical features and their implications for analysis. The original study contained many more outcomes, some of which were of greater scientific interest than the ones presented here.

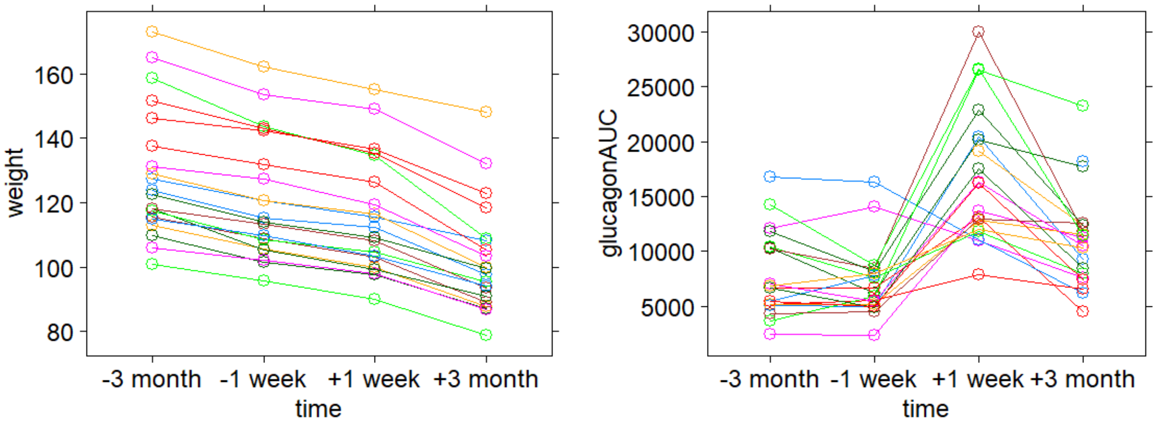


Figure 1: Bodyweight (kg) and serum glucagon (AUC) of $n = 20$ obese subjects, recorded -3 months before, -1 week before, +1 week after, and +3 months after gastric bypass surgery.

The case study data is shown in figure 1. We note a decreasing trend in bodyweight throughout the study, both in the individual subjects and in the population overall. Concerning glucagon, surgery tend to increase levels in the population, but the long term effect is less pronounced than the short term effect. On the individual level, the majority of the subjects seem to experience an increase following surgery, but there are also subjects who experience a drop. Overall individual glucagon levels are much less predictable than individual bodyweights. In statistical terms we would say that the serial correlation in bodyweight is much stronger than in glucagon. The tutorial contains a step-by-step analysis of the bodyweights. The corresponding analysis of glucagon data is left as an exercise (see the course webpage for questions and solutions).

3 Single group studies in general

In this section we discus the statistical properties of single group studies in general.

3.1 What is a balanced single group study?

From a repeated measurements point of view, a balanced single group study is when we have:

1. A single sample of subjects from a population of interest,
2. Outcomes planned to be collected repeatedly for a common replication factor, e.g.
    - Follow-up time in a follow-up study
    - Treatment in a fixed sequence cross-over study.
    - Location of the body in a study with physiological correlation)

Usually the aim of such a study is to evaluate the effect of the replication factor. E.g. when we analyze the gastric bypass study, we aim to compare the outcomes between the different follow-up times to see if systematic changes have occurred.

3.2 Note the difference: Balanced vs unbalanced design

A single group study is balanced when the replication factor is intentionally the same for all the subjects in the study. It is important to notice that this is a property of the study design, not of the data. When data is collected from a follow-up study, subjects are seldom assessed at the *exact* same times. However, as long as the deviations from schedule are small and random this has little consequence for statistics. Data may still be analyzed using the linear mixed model described in this tutorial.

In contrary, unbalanced study designs occur when follow-up times vary between the individual subjects, e.g. in retrospective registry studies where follow-up was not planned but took place according to the individual patients needs. Analyzing such data requires substantially different modeling which is not described in this tutorial.

3.3 Note the difference: Complete vs incomplete data

Data is incomplete when some of the planned measurements are missing.

From a technical point of view incomplete data do not prevent analysis. Linear mixed models may still be applied and are even optimal (in terms of statistical power) as long as the missing data are *missing at random*, that is missing for harmless reasons (see lecture on missing data).

However, statistical results may be biased if data is missing for specific reasons that are related to the outcome. E.g. if the most ill patients drop out of a follow-up study, those who remain no longer make up a representative sample from the original study population. Their outcomes will leave an overly optimistic impression of how the well the population is faring. Therefore potential causes for the missing data and their impact on the outcomes of the study should be considered prior to analysis. If you have data that are missing for all but harmless reasons, we recommend you consult with a statistician.

3.4 Choosing a statistical analysis: Linear mixed model or paired t-tests?

If your interest is to compare time points two by two in a single group follow-up study, then you could also make the comparisons with the plain old paired t-test. The linear mixed model and the paired t-tests will give you the same results if the data is complete and if the sample size (number of subjects) is reasonably large.

If you have missing outcomes, the linear mixed model is optimal under a missing at random assumption (see lecture on missing data), whereas the paired t-tests have less statistical power and may be biased.

If you have complete data and a small sample size, the paired t-tests should be preferred since this computes p-values and confidence intervals by exact formula while numerical approximation in the linear mixed model analysis may lead to less accurate results (see sections 6.9–6.10).

If you have both a small sample size and missing data, then consult with a statistician.

4 Preparing data for analysis

A trimmed version of the original records from the gastric bypass study are contained in the dataset `gastricbypassW` in the `LMMstar`-package. It contains the following variables:

Variable	Contents
`id`	Pseudo-ID (anonymized).
`weight1-4`	Bodyweight in kg at 1st, 2nd, 3rd, and 4th follow-up.
`glucagonAUC1-4`	Total AUC for glucagon at 1st, 2nd, 3rd, and 4th follow-up.

4.1 Load the case study data

To get a quick start on learning linear mixed model analyses, load the data with:

```
# Load the LMMstar-package:
library(LMMstar)

# Load the case study data (wide format)
data("gastricbypassW")
```

For smoother programming we make a copy of the data with the shorter name `wide`:

```
wide <- gastricbypassW
```

If you prefer to read data from the external file `gastricbypass.txt`, see appendix A.

4.2 Inspecting the data in R

Once you have managed to load the case study data in R, you should be able to locate the dataframe `wide` in the workspace. If you are running R Studio, look for the name `wide` in the **Environment** window.

To summarize the contents of the data, we use the command:

```
str(wide)
```

which results in:

```
'data.frame': 20 obs. of  9 variables:
 $ id          : Factor w/ 20 levels "1","2","3",..: 1 2 3 ...
 $ weight1      : num  127 165 110 146 113 ...
 $ weight2      : num  121 153 102 142 106 ...
 $ weight3      : num  115.5 149.2 97.7 136.7 99.9 ...
 $ weight4      : num  108.1 132 87.1 123 87.7 ...
 $ glucagonAUC1: num  5032 12142 10321 6693 7090 ...
 $ glucagonAUC2: num  4942 14084 6202 6632 NA ...
 $ glucagonAUC3: num  20421 10946 20121 13090 19155 ...
 $ glucagonAUC4: num  9249 7612 17705 4551 12345 ...
```

This tells us that the dataframe `wide` contains nine variables and 20 records, the names of the variables, their data types, and some of the recorded values.

The outcomes are described as `num`, meaning that R regards them as continuous, numerical variables, where as `id` is described as `Factor` which means that R regards it as a categorical variable.

4.3 Numerical and categorical variables in R

Please note that `id` is numbered 1, 2, 3,... merely for convenience. It is not so that subject number four is twice as much of a person as subject number two, and we won't get subject six by adding the two!

If you have chosen to load the data from `gastricbypass.txt` (appendix A), R will mistake `id` for a numerical variable unless you change its format to categorical with the `factor`-function:

```
wide$id <- factor(wide$id)
```

4.4 The long and the wide data format

In order to perform linear mixed model analyses it is important to distinguish two different formats for the same data.

1. **The wide format** with one record (dataline) per subject.
2. **The long format** with several records (datalines) per subject, one for each occasion.

The dataframe `wide` is in the wide format. This is the natural format when collecting data from a balanced study; It is compact and makes it easy to compare measurements from the same subject between occasions. E.g the records from id=2 and id=3 in the `wide` data reads:

id	weight1	weight2	weight3	weight4	glucagonAUC1	...	glucagonAUC4
2	165.2	153.4	149.2	132.0	12142.50	...	7612.50
3	109.7	101.6	97.7	87.1	10321.35	...	17704.50

From this we see that id=2 had an initial weight of 165.2 kg which had dropped to 132.0 kg at end of study. Moreover, glucagon AUC decreased from baseline to end of study in this subject. Similarly id=3 had an initial bodyweight of 109.7 kg which decreased to 87.1 kg, while at the same time glucagon AUC increased in this subject.

To perform a mixed model analysis data must be structured in the long format where each measurement appear in separate lines in the dataset and an additional variable identifies the occasion. E.g. in the gastric bypass study id=2 contributes four records:

id	visit	time	weight	glucagonAUC
2	1	-3 month	165.2	12142.5
2	2	-1 week	153.4	14083.5
2	3	+1 week	149.2	10945.5
2	4	+3 month	132.0	7612.5

Here the same id-number is repeated four-fold because it does not change between occasions. In contrast, the outcome variables `weight` and `glucagonAUC` vary between occasions. Moreover, note that only two variables are needed to contain the outcomes in the long data, while eight were needed in the wide data.

4.5 R-program: Transforming data from the wide to the long format

Fortunately we do not have to rearrange data manually to make a long version that can be used in the mixed model analyses. This R code transforms the `wide` data and stores it in a new dataframe called `long`:

```
# Reshape data from wide to long format

long <- reshape(wide,
     direction="long",
     idvar="id",
     varying=list(
      c("weight1","weight2","′weight3","weight4"),
      c("glucagonAUC1", "glucagonAUC2", "glucagonAUC3", "glucagonAUC4")
      ),
     v.names=c("weigh"′,"glucagonAUC"),
     timevar="visit")
```

Here the `idvar` argument indicates the variable identifying the subjects. In the long data, this will be used to link the multiple records from the same person. The `varying` argument should contain the names of the variables containing the repeated measurements in the wide data. Here we input a list because we have two different outcomes (weight and glucagon). Each element of the list names the repeated measurements for a particular outcome. The argument `v.names` is used to name the columns in the long data corresponding to the two outcomes. The argument `timevar` names the replication factor in the long data. It is important to notice that R automatically numbers the replicates as 1, 2, 3, so if you want different names or numbers you have to change them afterwards. Here we add the categorical variable `time` to the data:

```
# Add a categorical version of the follow-up time variable to the data:

time.names <- c("-3 month","-1 week","+1 week","+3 month")
long$time <- factor(long$visit, labels=time.names)
```

Please check the contents of the long dataset, e.g. by opening it from the **Environment** window in R Studio. Make sure to notice the differences compared to the wide data format.

If you have problems running the above code, you can make a shortcut by loading the long data directly from the `LMMstar`-package:

```
# Load the case study data in long format
data("gastricbypassL")
long <- gastricbypassL
```

4.6 Why have a categorical *and* a numerical version of the time variable?

Now our `long` data contains two different variables describing the same follow-up times:

- `visit` is a numerical variable which numbers follow-up times as `1,2,...`.
- `time` is a categorical variable which labels follow-up times as `"-3 month"`, `"-1 week"`, `"+1 week"`, and `"+3 month"`.

Keeping both versions is convenient as R often treats categorical and numerical variables differently. This goes for instance for graphical functions.

5 Descriptive statistics

Before we proceed to making statistical analyses, we will discus descriptive statistics suitable for single group studies. The graphical displays will help us not only to spot erroneous measurements that could otherwise bias the analyses, but also to make assessments of the trends in the data which will help us interpret the output from the linear mixed model analyses. Finally the descriptive statistics will help us judge whether data is approximately normally distributed or it should be transformed to better fulfill model assumptions.

Some of the descriptive statistics contained in this section are vital for the mixed model analyses. They will help you to correct errors in your R code as well as in your data. Others are mainly included for pedagogical reasons to help you become familiar with repeated measurements data and the multivariate normal distribution in general.

5.1 Spaghettiplots

We recommend to never analyze repeated measurements without making spaghettiplots. The spaghettiplot is the most informative descriptive figure for repeated measurements, since it shows the data in full detail. From this you can get an impression of both individual outcomes and the distribution across the sample. Once you get accustomed to the plot, you will be able to identify all of the most important features of your data:

- Trend.
- Variability.
- Skewness.
- Outliers.
- Serial correlation (aka tracking effect).

Spaghettiplots can be made for instance with the `xyplot`-function in the `lattice`-package.

```
library(lattice)
xyplot(weight~time, data=long, group=id, type="b")
```

Note that `xyplot` must be applied to data which is in the long format. The argument `group=id` connects the outcomes belonging to the same `id`. The argument `type="b"` implies that both lines and markers are drawn, while `type="l"` means lines only. Additional options can be supplied to alter the appearance of the plot, but usually the defaults are sufficient to get an impression of the data.

In figure 2 below we immediately notice a decreasing trend in bodyweight, both in the individuals and in the sample as a whole. Variability seems to be roughly the same at each follow-up. Moreover, we see that the distribution of the bodyweights is somewhat skewed with a few extremely heavy subjects, a majority of moderately heavy subjects, and no normal weight subjects.

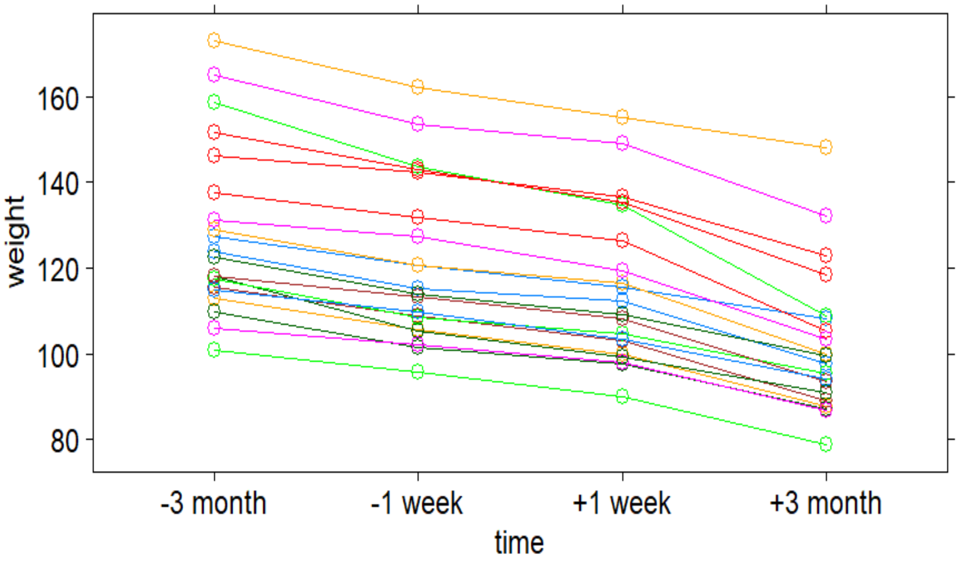


Figure 2: Spaghettiplot of bodyweights from the gastric bypass study made with `xyplot`.

Finally, we note a strong tracking effect, meaning that subjects rarely change positions. The heaviest subject at baseline is the heaviest throughout the study and similarly with the lightest subject. This tells us that that the serial correlation is strong, which also makes sense from a biological point of view. Bodyweights do not change dramatically from day to day, they are highly predictable over short time durations.

5.2 Scatterplot matrices

If you want to get a more familiar picture of how strong the correlation is between your repeated measurements and whether they are multivariate normal distributed, you can make scatterplot matrices with the `plot`-function. Please note that this applies to data in the wide format.

```
# Extract replicate outcomes to simplify code:
weight.replicates <- wide[,c('weight1','weight2','weight3','weight4')]

# Simple scatterplot matrices:
plot(weight.replicates)
```

When interpreting the scatterplots (figure 3 below), you should look for the elliptical shape that characterizes the multivariate normal distribution.

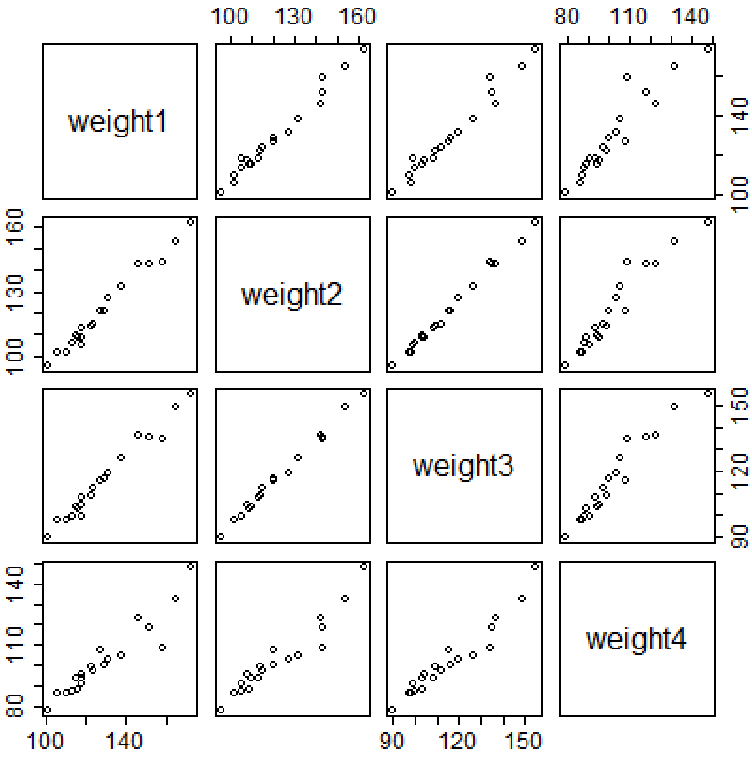


Figure 3: Scatterplot matrices of weights from the gastric bypass study made with `plot`

The scatterplot matrix confirms the strong correlation we already noted in the spaghettiplot. Also we confirm that the distribution of the weights at each separate follow-up time is skewed, albeit not dramatically so. The skew trend seems to be mainly driven by one or two outlying subjects. However, we know that bodyweight has a skew distribution in the general population, so this is likely a genuine finding.

A more informative scatterplot matrix with superimposed normal distributions and including the estimated correlations can be made with the `pairs.panels` function in the `psych`-package:

```
# Scatterplot matrices with effects (psych-package):
library(psych)
pairs.panels(weight.replicates, ellipses=TRUE, smooth=FALSE)
```

We use `ellipses=TRUE` to add correlation ellipses to the plot and `smooth=FALSE` to omit smoothed trends (like in a regression), which would otherwise disturb the picture.

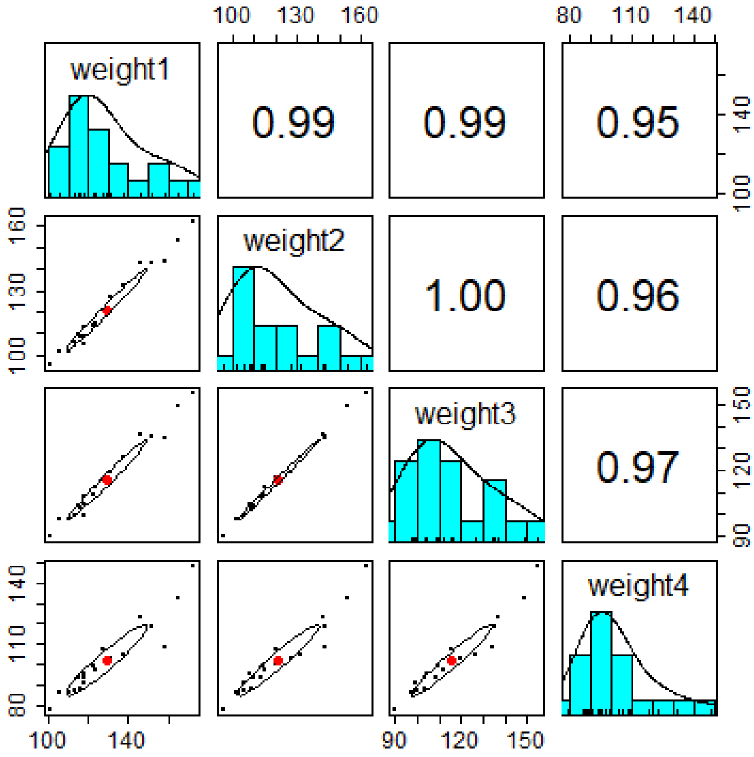


Figure 4: Scatterplot matrices of weights from the gastric bypass study made with `pairs.panels`

5.3 Is data normally distributed - and does it matter?

The problem with normality is not different between linear mixed models and ordinary linear models. If the sample size (number of subjects) is tiny, the assumption that data is normally distributed is crucial but impossible to verify. This is why small sample size is always a limitation.

If sample size is large, on the other hand, the statistical results are robust to all but substantial skewnes and extreme outliers. This means that estimates, confidence intervals, and p-values are statistically valid regardless of deviations from the normal distribution. However, one should be a little cautious when interpreting estimated mean differences as the mean doesn't necessarily describe the most typical outcome in the population.

If the data deviates substantially from the normal distribution, you can try to improve the fit by applying a transformation, e.g. a logarithm. Further considerations on transformation are given in appendix D. See in addition sections 6.9–6.10 for considerations on sample size.

5.4 Trends in summary statistics over time

Summary statistics for the various follow-up times can be computed with the summarize-function in LMMstar. Default output is number of non-missing and missing observations, mean, standard deviation, median, quartiles, and ranges (minimum and maximum):

```
# Summary statistics:
w.summaries <- summarize(weight~time, data=long, na.rm=TRUE)
print(w.summaries, digits=4)
```

```
  outcome      time observed missing  mean    sd   min median   max
1  weight -3 month       20       0 129.0 20.27 100.9  123.1 173.0
2  weight  -1 week       20       0 121.2 18.91  95.7  114.5 162.2
3  weight  +1 week       20       0 115.7 18.28  89.9  110.6 155.0
4  weight +3 month       20       0 102.4 17.05  78.8   98.5 148.0
```

The sample means confirm the decreasing trend we saw in the spaghettiplot. What was less obvious from the plot, is that the standard deviations also decrease over time. Usually decreasing variability with decreasing level suggest a logarithmic transformation, but note that the linear mixed model used in the analysis do **not** assume that the variance is homogeneous. Hence, variance heterogeneity is **not** a model deviation. The medians are, however, consistently smaller than the means, which confirms the skewness we saw in the plots, and this is a model violation, albeit a fairly harmles one.

We can use the xyplot-function to plot the sample means and standard deviations over time:

```
# Plot trends in summary statistics:
xyplot(mean~time, data=w.summaries, type="b")
xyplot(sd~time, data=w.summaries, type="b")
```

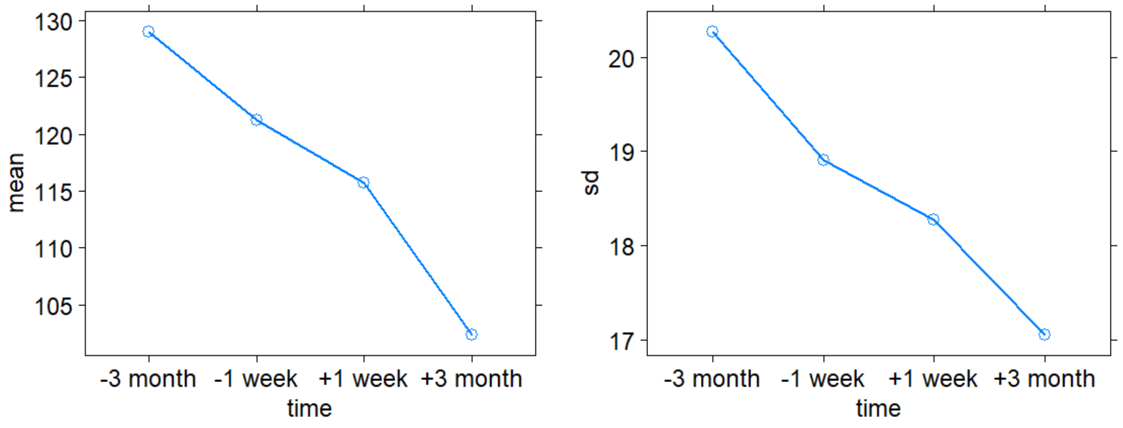


Figure 5: Trends in sample means and standard deviations from the gastric bypass study.

6 About linear mixed models

In this section we introduce linear mixed models and discus their statistical properties. We have attempted to make the descriptions as non-technical as possible, explaining the theory in words and pictures as well as in formulae. However, statistics rely on mathematical foundations, so formulae cannot be avoided completely. The most technical subsections are marked with $\star$ and can be skipped by non-mathematician readers. The philosophy of this tutorial is that full technical understanding of the linear mixed models is not needed to apply the models in practice. If you prefer a mathematically rigorous introduction we refer to Fitzmaurice, Laird & Ware (2012) and Pinheiro & Bates (2006)

6.1 What is a linear mixed model?

A linear mixed model is a multivariate normal model for repeated measurements including to distinct set of parameters.

1. Mean parameters describing trends in the population, usually expressed in terms of regression coefficients for covariates.

2. Covariance parameters, describing variance and correlation in the population.

The first part implies that interpreting results from linear mixed model analyses is easy if you are familiar with ordinary linear models for independent data.

The extra challenge that comes with a linear mixed model analysis is to choose a suitable model for the covariance. Overall we have three different ways of doing this:

- Direct specification of a covariance pattern (`lmm`- or `gls`-functions in R).
- Indirect specification via random effects (`lme`- or `lmer`-functions in R).
- A combination of the two (possible in SAS, but not in R).

We will introduce the different types of models on a case to case basis in our lectures and exercise classes, starting with the unstructured covariance pattern in section 6.5 below.

It should be noted that although some overlap exist between the three classes of models, different specifications of the covariance in general will give rise to substantially different statistical results.

Although the covariance parameters are rarely of an interest of their own, modeling the covariance correctly is important for making statistical conclusions. If the covariance is misspecified, the standard errors for the mean parameters may be biased, meaning that you get confidence intervals that are too narrow or too wide and p-values that are smaller or larger than they ought to be. In case there are missing data, misspecification of the covariance pattern may also inflict bias in the estimates for the mean parameters.

6.2 Describing linear mixed models in research papers

When describing a particular linear mixed model in the statistical methods section of a research paper it is important that you specify not only which covariates (fixed effects) entered the model, but also what model you chose for the covariance.

A description of the model for the case study would be something like this:

Case: *To analyze changes in bodyweight over time we applied a linear mixed model including follow-up time (categorical) as a fixed effect. To account for the correlation in the repeated measurements as well as possible variance heterogeneity over time, we assumed an unstructured covariance pattern.*

The model for the case study is described in further detail in sections 6.4 and 6.5 below.

6.3 The linear mixed model described in formulae*

To get a good understanding of linear mixed models, you need to know the basic formula defining the models and the terminology related to them. Phrased in mathematical terms, we describe repeated measurements on subjects $i = 1,\ldots,n$ over occasions $j = 1,\ldots,k$ as:

$$Y_{ij} = \beta_1 X_{ij1} + \ldots + \beta_{im} X_{ijm} + \varepsilon_{ij}$$

where $X_{ij1},\ldots,X_{ijm}$ are the covariates for subject i at occasion j, $\beta_1,\ldots,\beta_m$ are the regression parameters. The first covariate in the formula, X_{ij1}, will often be a constant equal to 1, so that the corresponding regression parameter β_1 is an intercept.

The covariates in a linear mixed model are referred to as *fixed effects*.

The error terms in the formula, $\varepsilon_{i1},\ldots,\varepsilon_{ik}$, are assumed to follow a multivariate normal distribution with zero mean. In statistical theory, the correlation between them is implicitly modeled in the $k \times k$ dimensional residual covariance matrix:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \ldots & \sigma_{1k} \\ \vdots & \ddots & \vdots \\ \sigma_{k1} & \ldots & \sigma_k^2 \end{pmatrix}$$

which is a mathematically convenient way of reporting variances and correlations in a compact manner. For interpretation we prefer looking at standard deviations and correlations. These can be recovered from the covariance matrix by use of the formulae:

$$\sigma_j = \sqrt{\sigma_j^2} \quad \text{and} \quad \rho_{jl} = \frac{\sigma_{jl}}{\sigma_j \sigma_l}$$

However, in practice you get the standard deviation and correlation as default output when analyzing linear mixed models with the `LMMstar`-package, so you need not worry all that much about covariances.

In the statistical literature, the covariance matrix is sometimes referred to as the R matrix. This should not be confused with the software.

6.4 Case: Oneway ANOVA-like linear mixed model

To model changes in mean bodyweight over time, we specify a linear mixed model with follow-up time as a categorical covariate (fixed effect). This is similar to a oneway ANOVA model for independent data where we have a separate mean parameter for each follow-up time.

However, when we analyse data in R, the default model specification (section 7.1) will give us estimates of the difference in means with respect to a reference time point (usually baseline). I.e. the default model parameters are the regression parameters $\beta_1, \beta_2, \beta_3, \beta_4$ that describes the mean bodyweight in the population over time as in the following table:

Time	Population mean	Estimated mean
-3 months	$\mu_1 = \beta_1$	128.97
-1 week	$\mu_2 = \beta_1 + \beta_2$	121.24 = 128.97 - 7.73
+1 week	$\mu_3 = \beta_1 + \beta_3$	115.70 = 128.97 - 13.27
+3 months	$\mu_4 = \beta_1 + \beta_4$	102.37 = 128.97 - 26.605

Table 1: Population means $(\mu_1, \mu_2, \mu_3, \mu_4)$ over time and corresponding estimates in the gastric bypass study. Changes in mean since baseline are described by the regression parameters β_2, β_3, and β_4. The intercept parameter β_1 is the mean at baseline. Estimates were obtained from the R-output in section 7.4. The regression parameters are further illustated in figure 6 below.

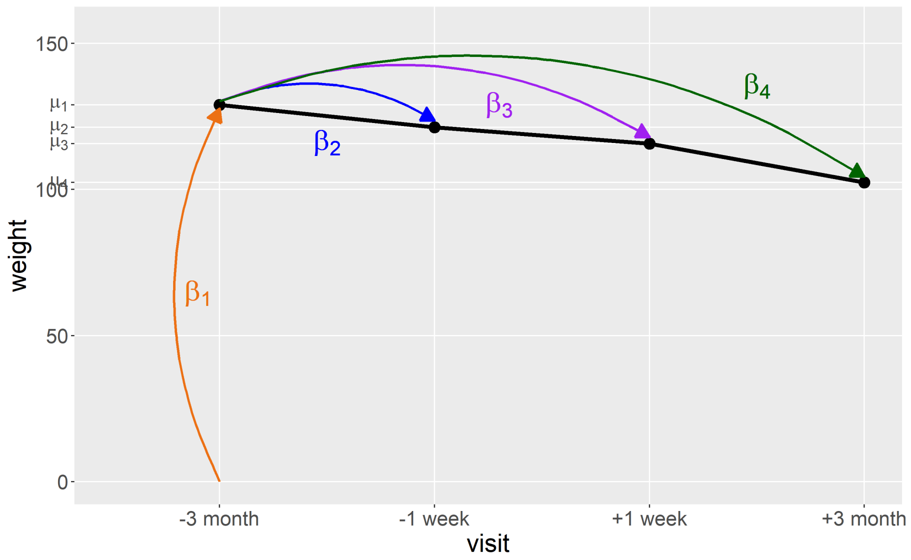


Figure 6: Graphical display of the estimated means over time (black curve) and regression coefficients (colored arrows) in the gastic bypass study.

Note that it is possible to choose a different reference point than baseline. This way we can estimate differences in means between any two follow-up times. Also it is possible to estimate the mean parameters μ_1, μ_2, μ_3 and μ_4 instead of the regression parameters β_1, β_2, β_3, and β_4. Check the R syntax in section 7.5 to learn how to do this.

6.5 The unstructured covariance pattern

Specifying a model for the covariance is easy when it comes to balanced single group studies. We can assume an unstructured covariance pattern, which models variances and correlations without any restrictions. This is a great advantage as the covariance can never become misspecified.

With repeated measurements over k occasions, the unstructured covariance pattern has:

- k variance parameters, denoted $\sigma_1^2, \ldots, \sigma_k^2$.
- $k(k-1)/2$ correlation parameters denoted $\rho_{12}, \ldots, \rho_{k-1,k}$.

In the case study with $k = 4$ follow-up times, we have 4 district variances and 6 distinct correlations, leading to the following R covariance matrix:

$$\Sigma_{UN} = \begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_{1,2} & \sigma_1\sigma_3\rho_{1,3} & \sigma_1\sigma_4\rho_{1,4} \\ \sigma_1\sigma_2\rho_{1,2} & \sigma_2^2 & \sigma_2\sigma_3\rho_{2,3} & \sigma_2\sigma_4\rho_{2,4} \\ \sigma_1\sigma_3\rho_{1,3} & \sigma_2\sigma_3\rho_{2,3} & \sigma_3^2 & \sigma_3\sigma_4\rho_{3,4} \\ \sigma_1\sigma_4\rho_{1,4} & \sigma_2\sigma_4\rho_{2,4} & \sigma_3\sigma_4\rho_{3,4} & \sigma_4^2 \end{pmatrix}$$

with estimated standard deviations $(\hat{\sigma}_1, \hat{\sigma}_2, \hat{\sigma}_3, \hat{\sigma}_4) = (20.27, 18.91, 18.28, 17.05)$ and estimated correlations $(\hat{\rho}_{1,2}, \hat{\rho}_{1,3}, \hat{\rho}_{1,4}, \hat{\rho}_{2,3}, \hat{\rho}_{2,4}, \hat{\rho}_{3,4}) = (0.990, 0.986, 0.946, 0.997, 0.959, 0.966)$.

We recommend that you specify your linear mixed model with an unstructured covariance pattern whenever possible, since it guarantees that you do not make any wrong model assumptions concerning the covariance. Unfortunately the unstructured covariance pattern cannot be applied when the number of follow-up times exceed the number of subjects in the sample. Nor can it be used and when the study design is unbalanced.

6.6 Random effect of subject vs unstructured covariance

Having an unstructured covariance pattern is preferable to having a more restrictive covariance pattern as we have argued in the previous section. Nevertheless, many health science researchers handle repeated measurements by using a different type of linear mixed model, namely one with a random effect of subject. This is unfortunate as the random effect gives rise to very restrictive assumptions about the variances and correlations.

The assumed R covariance matrix would be a so-called compound symmetry pattern:

$$\Sigma_{CS} = \begin{pmatrix} \sigma^2 & \sigma^2\rho & \sigma^2\rho & \sigma^2\rho \\ \sigma^2\rho & \sigma^2 & \sigma^2\rho & \sigma^2\rho \\ \sigma^2\rho & \sigma^2\rho^2 & \sigma^2 & \sigma^2\rho \\ \sigma^2\rho & \sigma^2\rho^2 & \sigma^2\rho & \sigma^2 \end{pmatrix}$$

In the case study with estimated standard deviation $\hat{\sigma} = 18.66$ **common to all occasions** and estimated correlation $\hat{\rho} = 0.970$ **between any two occasions** (see appendix B for R-code and output).

In other words, by modeling correlation by having a random effect of subject (compound symmetry pattern) in your linear mixed model, you assume that both variances and correlations are constant over time. This is most often **not** a realistic assumption.

Covariance patterns and random effects models are further discussed in lectures 2 and 3.

6.7 Maximum likelihood estimation*

This section contains a short and mostly intuitive description of what is actually a delicate matter for mathematical statisticians.

To assess what parameters fit the data well, we use the likelihood function. In a linear mixed model this is the multivariate normal probability density.

$$L(\mu,\Sigma)=\prod_{i=1}^{n}\left(\frac{1}{\sqrt{(2\pi)^k|\Sigma|}}e^{-\frac{1}{2}(y_i-\mu)^T\Sigma^{-1}(y_i-\mu)}\right) \tag{1}$$

The product is taken over subjects $i=1,\dots,n$ (assumed independent so that their probability densities can be multiplied). Here y_i is the k-dimensional vector observation for subject i, μ is the mean vector, Σ is the residual covariance matrix, Σ^{-1} its inverse, and $|\Sigma|$ its determinant. If you are not familiar with matrix computation, think of the univariate normal density instead.

The key idea in maximum likelihood estimation is as follows:

- We assume that the k-dimensional outcomes $y_1,\dots,y_n$ have been sampled from a k-dimensional multivariate normal distribution with population mean μ and population covariance matrix Σ.

- If we knew the true population parameters μ and Σ, then the likelihood function, i.e. the probability density (1), would tell us which outcomes were more or less likely. In a normal distribution the most likely outcomes are close to the population mean where the density attains its maximum value. With increasing distance from the mean, the likelihood values decrease and outcome values become correspondingly rare.

- In practice, we do not know what the true population parameters are, but we do know a bunch of likely outcomes, namely our data, $y_1,\dots,y_n$. Hence we use the likelihood function (1) to determine which parameters would make the observed outcomes most likely.

This is the likelihood principle which states that the parameters that fit the data the best are the ones which have the highest likelihood values i.e. the maximum likelihood estimates.

To make matters even more complicated, the parameters of linear mixed models are usually estimated by use of a variant of the likelihood function called the restricted likelihood function, see e.g. Fitzmaurice, Laird & Ware (2012)). This is because the restricted likelihood estimates of the covariance parameters are more accurate in small samples. Therefore restricted maximum likelihood (abbreviated REML) is the default estimation method in most software including the `LMMstar`-package.

Case: The restricted maximum likelihood estimates from a balanced single group study with complete data are the sample means, the sample SDs, and the sample correlations,

$$\hat{\mu}_1 = 129.0, \quad \hat{\mu}_2 = 121.2, \quad \hat{\mu}_3 = 115.7, \quad \hat{\mu}_4 = 102.4,$$
$$\hat{\sigma}_1 = 20.3, \quad \hat{\sigma}_2 = 18.9, \quad \hat{\sigma}_3 = 18.3, \quad \hat{\sigma}_4 = 17.1,$$
$$\hat{\rho}_{12} = 0.990, \quad \hat{\rho}_{13} = 0.986, \quad \hat{\rho}_{14} = 0.946, \quad \hat{\rho}_{23} = 0.997, \quad \hat{\rho}_{24} = 0.959, \quad \hat{\rho}_{34} = 0.966.$$

These are identical to the summary statistics from section 5.

6.8 Numerical optimisation and convergence*

An important difference between linear mixed models and ordinary linear models is that in linear mixed models the estimates and standard errors cannot in general be computed using textbook formula. Balanced single group studies with complete data are an exception to this rule, but software packages will ignore this fact when analysing a particular data set. Instead they use numerical optimisation to approximate the maximum likelihood estimates.

To give you a rough impression of how numerical optimisation works, we will briefly describe what `lmm` and other software for linear mixed models does. Again, this is a simplified description of a rather complex technical matter.

For ease of computation the numerical optimisation operates on the deviance function which is -2 times the natural logarithm of the likelihood function (1), i.e.

$$\text{deviance}(\mu, \Sigma) = k \cdot n \cdot \log(2\pi) + n \cdot \log|\Sigma| + \sum_{i=1}^{n} (y_i - \mu)^T \Sigma^{-1} (y_i - \mu) \tag{2}$$

Higher values of likelihood correspond to lower values of deviance. Hence the maximum of the likelihood function is the same as the minimum of the deviance function.

In practice, the deviance (2) is usually replaced by the deviance of the restricted likelihood function, see e.g. Fitzmaurice, Laird & Ware (2012) for details.

This is how the numerical optimisation works:

1. `lmm` initially makes a qualified guess of what the estimates could be.
   - Mean parameters are initially estimated as if data was independent (i.e. using model formulae for the ordinary linear model, like with the `lm`-function).
   - Covariance parameters are estimated using more complicated computations that are specific to the particular covariance pattern, see Pinheiro & Bates (2006) for details.
2. Goodness of fit between the candidate estimates and the data is evaluated by the deviance function. Lower values of deviance are better (they correspond to higher likelihood).
3. R computes the gradient (i.e. the multivariate first order derivative) and the Hessian (i.e. the multivariate second order derivative) of the deviance function to move the candidate estimates in a direction where the deviance will be smaller (and the likelihood higher).

4. Steps 2-3 are iterated until the candidate is within a certain tolerance of the maximum likelihood estimates (in the sense that the gradient is close to zero), or until the number of iterations reach an upper limit set to avoid that the program runs forever.

We say that the numerical optimisation has converged, if it stops when it has located the maximum likelihood estimates. Section 7.2 describes how to assess convergence in practice.

6.9 Inference: Confidence intervals and hypothesis testing*

Recall that 95% confidence intervals in linear models are computed from the formula:

$$\begin{array}{rcl} \hat{\beta} & \pm & t_{0.975}(\text{df}) \times \text{s.e.}(\hat{\beta}) \\ \text{estimate} & \pm & \text{roughly } 2 \times \text{standard error} \end{array}$$

where $t_{0.975}(\text{df})$ is formally the 97.5% quantile in a t-distribution with model specific denominator degrees of freedom (df). Likewise the hypothesis $H_0 : \beta = 0$ can be tested using a t-type test statistic:

$$t = \frac{\hat{\beta}}{\text{s.e.}(\hat{\beta})}$$

which is compared to the t-distribution with df degrees of freedom.

The trouble with linear mixed models is that there are no simple general formula to compute the degrees of freedom. As sample size becomes large $t_{0.975}(\text{ddf}) \approx 1.96$. Otherwise the residual degrees of freedom that are used for t-tests and corresponding confidence intervals can only be computed exactly for some very specific balanced study designs when the data is also complete. In any other instance, degrees of freedom must be approximated using more of less complex computations.

The most recent method for approximating denominator degrees of freedom in linear mixed models is due to Kenward & Roger (2009). This has been implemented in PROC MIXED in SAS and in the pbkrtest-package in R, whereas the LMMstar-package relies on the Satterthwaite approximation, see e.g. Hrong-Tai Fai & Cornelius (1996). It is important to notice that a poor approximation of the denominator degrees of freedom may lead to biased inference in small samples, namely confidence intervals that are too narrow and p-values that are too small.

Case: The table below contains 95% confidence intervals for the mean weight change from baseline (-3 months) to final follow-up (+3 months) using approximate degrees of freedom from the gls-function in the nlme-package in R, the lmm-function in the LMMstar-package in R, and PROC MIXED in SAS, respectively. As benchmark we have used the exact 95% confidence interval for the paired t-test.

Denominator degrees of freedom method	95% CI for the mean change	Width relative to reference
No correction (gls-function in R)	-26.6050 (-29.6892;-23.5207)	0.9511
Satterthwaite (lmm-function in R)	-26.6050 (-29.8483;-23.3617)	1.0001
Kenward & Roger (2009) (SAS)	-26.6050 (-29.8479;-23.3621)	1.0000
Exact 95% CI (paired t-test)	-26.6050 (-29.8479;-23.3621)	Reference

In this case the approximate 95% confidence interval we get from PROC MIXED in SAS is spot on and `lmm` is very close to it, while `gls` underestimates the width by 5%. Note that this is solely due to a poor approximation to the degrees of freedom. The estimate and the standard error for the mean change will always agree between the linear mixed model and the paired t-test when:

1. The study design is balanced
2. The data is complete
3. The covariance is modeled by an unstructured pattern.

Also under these conditions we get a theoretical expression of the worst case attenuation factor occuring when `gls` computes the degrees of freedom as if the repeated measurements were independent data:

$$\frac{\text{width of confidence interval from } \texttt{gls}}{\text{width of correct confidence interval}} = \frac{t_{0.975}((n-1)*k)}{t_{0.975}((n-1)}$$

where $t_{0.975}(\mathrm{df})$ the 97.5% quantile in a t-distribution with df degrees of freedom, n is the number of subjects and k is the number of replicates.

In the specific example with $n = 20$ and $k = 4$, we can verify the attenuation factor:

```
qt(0.975, df=(20-1)*4) / qt(0.975, df=20-1)
```

```
[1] 0.9515766
```

6.10 Important: Sample size matters!

Confidence intervals and p-values in linear mixed models are all based on statistical large sample theory. Hence, you need to consider sample size (number of subjects) to assess whether your statistical results are valid.

When is sample size large enough? As usual, there is no strict answer to this question. Indeed if deviations from the normal distribution are severe (e.g. if extreme outliers are present), if the number of replicates is large, or if correlations are extremely close to -1 or 1, then statistical results may be compromised even in larger samples. However, years of working with linear mixed models have left us with the following experience:

Sample size		Impact of normal distribution	Impact of degrees of freedom
Tiny	$n < 10$	Linear mixed model analysis disrecommended⋆⋆	
Borderline	$10 \leq n < 15$	Normal distribution matters	Large impact
Small	$15 \leq n < 30$	Moderate deviations tolerable	Moderate impact
Moderate	$30 \leq n < 60$	Larger deviations tolerable	Small impact
Large	$n \geq 60$	Most deviations tolerable	Hardly any impact.

⋆⋆ We suggest you use t-tests or non-parametric statistics instead.

Please note that these recommendations are for balanced designs with complete data. In case some of the outcomes are missing, the effective sample size will be smaller than the number of subjects in the sample. The number of complete cases (subjects with complete data) is a lower bound for the effective sample size.

Also note that when several groups are considered, n is the number of subjects **per group**.

6.11 Predicted values and residuals

Predicted values and residuals are useful for model validation. We can make residual plots and QQ-plots similar to those for ordinary linear models to evaluate a linear mixed model. However, linear mixed models are more complex and for this reason we use several different types of predicted values and residuals to check different aspects of the model specification.

Three different kinds of predicted values can be considered:

1. Ordinary predicted values (based on covariates).
2. Dynamical predictions (based on covariates and previous observations).
3. Subject specific predictions (for random effects models, see lecture 3).

The ordinary predicted values are what we get from the estimated linear model:

$$\text{Predicted value for subject } i \text{ at occasion } j\colon\; \hat{Y}_{ij} = \hat{\beta}_1 X_{ij1} + \ldots + \hat{\beta}_m X_{ijm}$$

These are similar to predicted values from an ordinary linear model. We recommend using the predicted values for an initial check of your model specification (see section 7.6). If your predicted values deviate substantially from the trends in the descriptive statistics, you have likely made an error in your R code or a poor choice of model.

The (ordinary) residuals are obtained by subtracting predicted values from observed:

$$\text{Residual for subject } i \text{ at occasion } j\colon\; e_{ij} = Y_{ij} - \hat{Y}_{ij}$$

These are similar to the residuals from an ordinary linear models. Only residuals belonging to the same subject will be correlated and they may have different standard deviations even if the model is correctly specified. E.g. the unstructured covariance pattern (section 6.5) assumes potentially different standard deviations for the different occasions.

The residuals allows you to make diagnostics plots that are better for checking normality than the simple scatterplots in section 5. However, since standard deviations from different occasions may differ in a mixed model, we cannot pool the ordinary residuals in a QQ-plot to check that they are normally distributed without first standardizing them. Similar to ordinary linear models, residuals may be standardized in two different ways:

- **Pearson residuals** have been standardized by dividing with the estimated standard deviations from each occasion in turn. The resulting residuals have similar standard deviations but are still correlated across occasions.

- **Studentized residuals** are leave-one-out versions of the Pearson residuals making it easier to detect influential outliers.

Dynamical predictions are what we get from predicting outcomes based on covariates and previous outcomes in conjecture. A mathematical property of the multivariate normal distribution is that it implies a linear model for either outcome with any other as predictor, i.e.

$$Y_{ij} = \beta_1 X_{ij1} + \ldots + \beta_m X_{ijm} + \gamma_1 Y_{i1} + \ldots + \gamma_{j-1} Y_{ij-1} + \text{error} \tag{3}$$

We will explain this in terms of two repeated measurements, Y_1 and Y_2, highlighting the connection between correlation and prediction. See figure 7 for illustration.

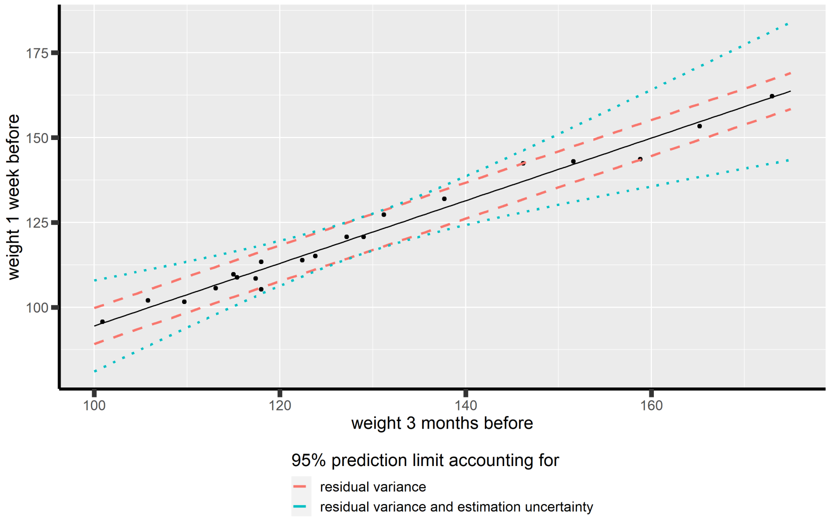


Figure 7: Dynamical prediction of the second bodyweight from the first bodyweight in the gastric bypass study. The strong correlation implies that the predictive accuracy is high.

Assuming that Y_1 and Y_2 follow a joint normal distribution with mean μ_1 and μ_2, standard deviations σ_1 and σ_2 and correlation ρ, then we have the following linear regression for predicting Y_2 based on Y_1 :

$$Y_{i2} = \gamma_0 + \gamma_1 \cdot Y_{i1} + \varepsilon_i^{Y_2|Y_1} \tag{4}$$

where the $\varepsilon^{Y_2|Y_1}$s are prediction errors following a normal distribution with zero mean and variance $\sigma^2_{2|1}$, called the prediction error variance.

The intercept, the slope, and the prediction error variance are determined by the parameters in the multivariate normal distribution:

$$\gamma_0 = \mu_2 - \rho \frac{\sigma_2}{\sigma_1} \mu_1, \quad \gamma_1 = \rho \frac{\sigma_2}{\sigma_1}, \quad \text{and} \quad \sigma^2_{2|1} = (1 - \rho^2)\sigma_2^2 \tag{5}$$

We note that the slope $\gamma_1 = \rho \frac{\sigma_2}{\sigma_1}$ has the interpretation that the predicted values of Y_2 increases with ρ standard deviations every time the predictor Y_1 increases with one standard deviation. Moreover, we see that the prediction error variance decreases as the correlation gets stronger. In other words, Y_2 is more accurately predicted from Y_1 if the correlation is strong. This is in good accordance with our understanding of the correlation.

From the prediction models we obtain residuals that are estimates of the dynamical prediction errors. These can be used for further validation of the linear mixed model. A noteworthy result from mathematical statistical theory is that the prediction errors are independent of each other. Hence, if the linear mixed model is correctly specified the standardized residuals from the dynamical prediction models should be approximately independent and normally distributed with zero mean and a standard deviation of one. The standardized dynamical residuals are called *scaled residuals* in the `LMMstar`-package to mimic PROC MIXED in SAS.

Case: Figure 8 illustrates the different residuals obtained from the gastric bypass study.

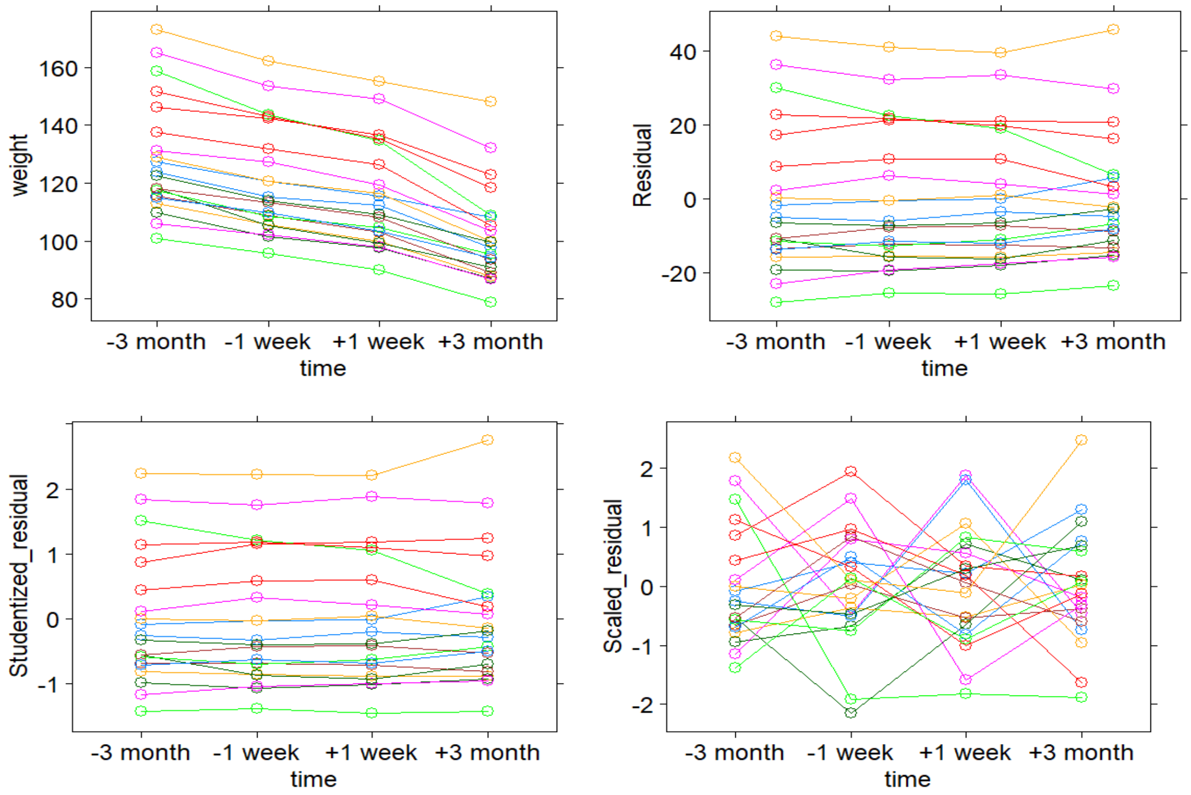


Figure 8: Residuals from fitting the linear mixed model to the bodyweights from the gastric bypass study. From upper left to lower right panel the plots are: Original data, ordinary residuals, studentized residuals, and scaled residuals. Pearson residuals have been omitted since they are practically indistinguishable from the studentized residuals.

Assuming that data is truly multivariate normal distributed all residuals should be symmetrically distributed around zero. Further the standardized residuals (Pearson, studentized and scaled) should have a homogeneous variance across occasions. The tracking effect from the original data persist in all but the scaled residuals, which ought to be uncorrelated if all the assumptions for the linear mixed model hold true.

For a worked example, we will take a closer look at the predicted values and residuals for `id=2`:

id	visit	weight	predicted	Residual	PearsonRes	StudentizedRes	ScaledRes
2	1	165.2	128.97	36.23	1.79	1.83	1.79
2	2	153.4	121.24	32.16	1.70	1.74	-0.47
2	3	149.2	115.70	33.50	1.83	1.88	1.88
2	4	132.0	102.36	29.64	1.74	1.78	-0.46

As for all subjects, the (ordinary) predicted values are equal to the estimated means from the separate occasions, which again are equal to the sample means because the data is complete. The (ordinary) residuals are the differences between the observed weights and the predicted values, e.g.

$$e_{i1} = Y_{i1} - \hat{\mu}_1 = 165.2 - 128.97 = 36.23.$$

This shows that `id=2` had an initial weight which is 36.2 kg above the estimated population mean. We get the Pearson residual by dividing with the estimated standard deviation, that is

$$\frac{e_{i1}}{\hat{\sigma}_1} = \frac{36.23}{20.27} = 1.79$$

showing that `id=2` had an initial weight which is 1.79 standard deviations above the estimated population mean. The subsequent Pearson residuals show that `id=2` remains about 1.7 to 1.8 standard deviations above the estimated population mean throughout the study, which makes good sense considering the strong serial correlation in the bodyweights.

The Studentized residuals are highly similar to the Pearson residuals, only slightly larger.

As to the scaled residuals, the first is identical to the Pearson residual (since we have no previous weights to predict from at baseline). At the second visit, we predict the second weight from the first using the linear regression model in figure 7 and we compute the prediction error:

$$\hat{Y}_{i2}^{|Y_1} = 154.7 \quad e_{i2}^{|Y_1} = Y_{i2} - \hat{Y}_{i2}^{|Y_1} \simeq 153.40 - 154.69 = -1.29$$

We next standardize this by dividing with the estimated standard deviation of the prediction error distribution, $\frac{-1.29}{2.78} = -0.47$. We note that the scaled residual is negative since the second observed weight of `id=2` is smaller than what the model predicts from his initial weight. However, the deviation from the prediction is not unusually large, it is only about one half standard deviation of the prediction error distribution. The third scaled residual is 1.88. Thus a visit 3, `id=2` had a higher weight than what the model predicts from his two previous weights. The deviation is fairly large but still within the normal range ± 2. At the final visit, `id=2` had a weight which is smaller than what the model predicts from the three previous weights, but well within the normal range of the estimated prediction error distribution.

The different kinds of residuals and their use for making model diagnostics are further described in chapter 10 of Fitzmaurice, Laird & Ware (2012). Predicted values and residuals from the case study are further investigated in section 7.6.

7 Analysis and interpretation of the linear mixed model

In the following section we will describe how to analyze the gastric bypass data using the lmm-function in the LMMstar-package in R. We will describe the the R-code and the interpretation of the output.

7.1 R program: Specifying the linear mixed model

To make a linear mixed model analysis of the gastric bypass data, we first set the reference point for the categorical covariate time. This will be the intercept in the linear mixed model.

```
# Set reference point (intercept) for time factor:
long$time <- relevel(long$time, ref="-3 month")
```

In this case we have chosen baseline as the reference point. Please note that you can switch to another follow-up time if you want to compare with that instead.

To do the linear mixed model analysis we need to use the lmm-function which is found in the LMMstar-package. Once we have loaded the package, we use lmm to specify a model object.

```
### Main analysis of changes in mean since baseline ####
fit.main <- lmm(weight~time,
               repetition=~visit|id,
               structure="UN",
               data=long)
```

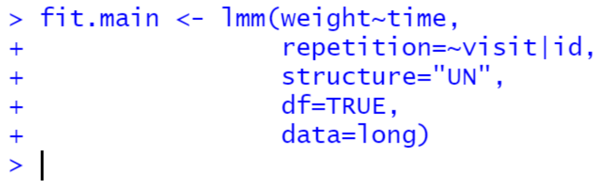
As to the essential parts of the code:

- The model formula, weight∼time, specifies which variable is the outcome (left of ∼) and which is the covariate (right of ∼). It is possible to have more than one covariate in a mixed model, but in this case we only have only one, namely time which is a categorical variable (a factor in R). Note that in case the time-variable was numerical, R would perform an analysis similar to linear regression only with correlated data.
- repetition=∼visit|id together with structure="UN" specifies an unstructured *correlation* matrix for the repeated measurements. We have to supply id and visit (or time) as shown in the above so that R knows which repeated measurements belong to the same subject and what their temporal ordering is.
- data specifies which dataframe the analysis is based on. This must be long format.
- If sample size is large, the optional argument df=FALSE can be used to speed up computation with little impact on the results. With small to moderate sample sizes, it is recommended to keep the default which is to compute the denominator degrees of freedom using the Satterthwaite approximation.

7.2 Checking convergence

You don't need to understand how the numerical optimisation within `lmm` works to analyze your data. However, you do need to know whether or not your attempt to fit the linear mixed model has succeeded. That is, whether or not the numerical optimisation has converged, see section 6.8.

Fortunately, checking convergence is easy. If your attempt to fit the linear mixed model is successful, then R produces no output in the console. It ought to look like this:



```
> fit.main <- lmm(weight~time,
+                 repetition=~visit|id,
+                 structure="UN",
+                 df=TRUE,
+                 data=long)
> |
```

At the same time, you can check whether the model object has been added to your workspace (in R Studio, look for `fit.main` in the list of objects in the Environment window). On the other hand, if the numerical optimisation fails to converge, R will return a more or less comprehensible error message.

So what if the model doesn't converge? Most often non-convergence is caused by simple errors like misspelling the name of a variable or applying `lmm` to the wrong dataframe. If you fix the error, the analysis will proceed smoothly. Changing the scale of the outcome, e.g. from body weight in g to bodyweight in kg, makes the optimisation procedure run more smoothly as does standardisation of numerical covariates (subtract the mean and divide by the standard deviation). Otherwise convergence issues often occur when trying to fit a too complex model to a too small dataset. Extreme outliers may also cause problems. If you cannot figure out why your mixed models doesn't converge, you should get help from a statistician.

Beware of false convergence: In rare cases it happens that R mistakes some implausible values for the maximum likelihood estimates or their standard error. This is called *false convergence*. Before you conclude on an analysis, you should therefore check that estimated covariance parameters and standard errors look reasonable and use fitted values and residuals for model validation as described in section 7.6 below. If you suspect that your attempt to fit the linear mixed model has resulted in false convergence, you should get help from a statistician.

7.3 R program: Extracting results from the model object

Assuming that you attempt to fit the linear mixed model was successful, R has assigned a model object to the name `fit.main`. From this you can extract different kinds of statistical results. Here we describe a number of useful *extractor functions*:

- `summary(fit.main)` returns estimates, approximate p-values[2], an overall model summary, and a load of other more or less useful information. A commented version of the full output can be found in appendix C.

[2] *Approximate* means that confidence intervals and p-values are valid for large samples, see sections 6.9–6.10.

- confint(fit.main) returns estimates and approximate 95% confidence intervals[2].
- plot(fit.main) display the fitted values over time.
- If you want to extract the estimated covariance matrix use sigma(fit.main)
- anova(fit.main) tests the global null hypothesis $H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$. This will make an approximate F-test[2] with respect to the degrees of freedom inherited from the model object.
- The predicted population means can be computed with the predict-function. In order to use it, we first need to construct a new data frame containing the values of the covariates we want to make predictions for. Here we extract the distinct values of time from the original data:

  ```
  # Make a dataset with covariate values for prediction:
  pred <- long[,c('time')]

  # Reduce to one of each value:
  pred <- unique(pred)

  # Add predicted means to the dataframe:
  pred <- cbind(pred, predict(fit.main, newdata=pred))
  ```

  Once the covariates and the predicted values have been stored in the same dataframe it is easy to plot the predictions with xyplot:

  ```
  xyplot(estimate~time, data=pred, type="b")
  ```

- residuals(fit.main, type="studentized") extracts the studentized residuals. By varying the type argument, you can also get the ordinary residuals, the pearson residuals, and the scaled residuals. To assess how well the linear mixed model fits the data, we recommend making the following diagnostic plots:

  ```
  par(mfrow=c(2,2))
  plot(fitted(fit.main), residuals(fit.main, type="studentized"))
  abline(h=0)
  qqnorm(residuals(fit.main, type="studentized"))
  abline(0,1)
  plot(fitted(fit.main), residuals(fit.main, type="scaled"))
  abline(h=0)
  qqnorm(residuals(fit.main, type="scaled"))
  abline(0,1)
  ```

7.4 Interpretation of estimates, confidence intervals, and p-values

Mean parameters: The most interesting part of the output is the estimated time effect, i.e. the estimated mean changes in bodyweight (kg) since baseline. These are reported for time = -1 week, time = +1 week, and time = +3 month follow-up. The intercept in this model corresponds to the estimated population mean at baseline (time = -3 month):

```
Fixed effects: weight ~ time

             estimate    se     df   lower   upper p.value    
(Intercept)    128.97 4.532 18.981 119.483 119.483  <0.001 ***
time-1 week     -7.73 0.697 18.974   -9.19   -9.19  <0.001 ***
time+1 week    -13.27 0.839 18.969 -15.027 -15.027  <0.001 ***
time+3 month  -26.605 1.549 18.964 -29.848 -29.848  <0.001 ***

The columns lower and upper correspond to the 95% confidence interval
Degrees of freedom were computed using a Satterthwaite approximation
```

The p-values for the time effects are all very small, so there is strong evidence that the mean bodyweight has decreased since baseline at all follow-up times. Estimated changes are -7.7 kg, -13.3 kg, and -26.6 kg at -1 week, +1 week, and -3 months follow-up, respectively. The confidence intervals tell us that the mean bodyweight is substantially lower at all follow-up times compared with baseline. This is no surprise, since both the pre-surgery diet and the gastric bypass has a well documented effect. Finally note that both the p-values and the confidence intervals are computed using a Satterthwaite approximation. This is important since the sample size is small (only $n = 20$ subjects).

Covariance parameters: The covariance parameters are usually of secondary interest to the mean parameters in linear mixed models analyses. However, you should still check on them, firstly since implausible values indicate errors in the data or the program, and secondly because they will help you get a better understanding of linear mixed models and the multivariate normal distributions. Finally, you might need the estimated covariance parameters to check the assumptions of a preceding power calculation or to make a new calculation for a future study.

For the gastric bypass study, the estimated correlations are:

```
  - correlation structure: ~visit | id 
          1     2     3     4
    1 1.000 0.990 0.986 0.946
    2 0.990 1.000 0.997 0.959
    3 0.986 0.997 1.000 0.966
    4 0.946 0.959 0.966 1.000
```

and the estimated standard deviations are:

```
  - variance structure: ~visit 
      standard.deviation     ratio
    1           20.26942 1.0000000
    2           18.91022 0.9329435
    3           18.27535 0.9016220
    4           17.05391 0.8413614
```

Please note that these estimates are identical to the summary statistics in section 5. This will always be the case if we analyse complete data from a balanced single group study with a model that is fully flexible (i.e. time as a categorical covariate and an unstructured covariance pattern).

Type 3 tests: If you would rather test the hypothesis $H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$, i.e. that no changes in means whatsoever occur over time. This is what you get from the F-test using the `anova`-function:

```
                       ** mean coefficients **
 - F-test
     statistic df.num df.denom      p.value
time  121.6594      3 18.97809 1.427969e-12
```

Note the usual limitation of the F-test: If the hypothesis is rejected, further comparisons are needed to investigate between which particular follow-up times significant differences occur. On the other hand, if the hypothesis is not rejected, you still want to estimate the differences between the follow-up times to assess whether this is likely due to lack of effect or lack of power. Hence, you might as well skip the F-test and do the pairwise comparisons to begin with (don't forget to make adjustment for multiple testing).

One further issue with the F-test, is that it is based on approximate degrees of freedom which are valid in large samples (see section 6.10). If the sample size is small as in the gastric bypass study, the p-value will be somewhat too small. However, when as in this case $P < 0.0001$. this has no impact on the strength of the evidence overall.

7.5 R Program: Alternative model specification (means over time)

Often researchers want to report estimated means for the various occasions along with the estimated changes in means which we obtained from the program in section 7.1.

We can make `lmm` estimate means over time rather than changes in means over time with a small change to the syntax. Adding `-1` to the model formula, makes R fit the model without an intercept, meaning that each follow-up time gets a separate mean parameter:

```
### Alternative syntax: means over time ####
fit.means <- lmm(weight~-1+time,
                 repetition=~visit|id,
                 structure="UN",
                 data=long)
```

Note that, besides the change to the model formula, the syntax is identical to that of section 7.1. From this we extract the estimated means with approximate 95% confidence intervals:

```
# Extract estimated means and confidence intervals:
confint(fit.means)
```

```
             estimate lower upper
time-3 month      129 119.5   138
time-1 week       121 112.4   130
time+1 week       116 107.1   124
time+3 month      102  94.4   110
```

When interpreting the confidence intervals it is important to keep the difference between independent and paired samples in mind. Estimated means from baseline and 1 week before surgery are contained in each others confidence intervals. Nevertheless, we concluded a significant difference in mean body weight with $P < 0.0001$ in section 7.4. This is a consequence of the strong correlation in the data: The paired t-test has much higher statistical power than the two-sample t-test. Recall that, we can only infer a significant difference from non-overlapping confidence intervals when samples are independent. This is a fact you might have to remind the reviewer of your paper otherwise you risk getting it rejected due to a false claim of statistical errors.

7.6 Model validation using predicted values and residuals

Predicted population means and standard deviations: As an initial model check we plot the predicted population means over time. These should be compared to the similar plots of the summary statistics (section 5.4). If the predictions deviate substantially from the summary statistics, this would indicate either an error in the code or a poor choice of model.

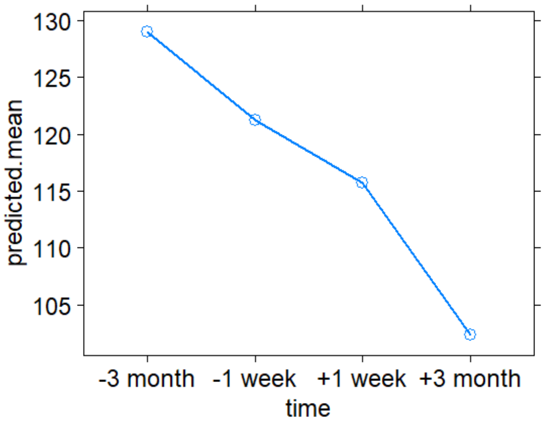


Figure 9: Predicted population means over time from the linear mixed model analysis of the gastric bypass study.

As we have already noted, estimated means and standard deviations for the bodyweights are identical to the summary statistics from section 5. This will always be the case when you have complete data from a balanced single group study and fit a fully flexible model for the mean and the covariance (i.e. time as a categorical covariate and an unstructured covariance pattern).

Residualplots are useful for checking the modeling assumption that data follows a multivariate normal distribution and for detecting outliers in the data. A large studentized residual (either positive or negative) indicates that the outcome is far from the expected mean outcome in the population measured in number of standard deviations. Similarly a large scaled residual indicates that the observation is far from the expected outcome predicted by the past outcomes.

Figure 10 displays the studentized and scaled residuals which are directly comparable to the standard normal distribution. If data is truly normal, the points ought to be on the diagonal lines in QQ-plots save from small random deviations. In this case the the QQ-plot of the studentized residuals seems to be smiling indicating that the distribution of the residuals is skew compared to the normal distribution.

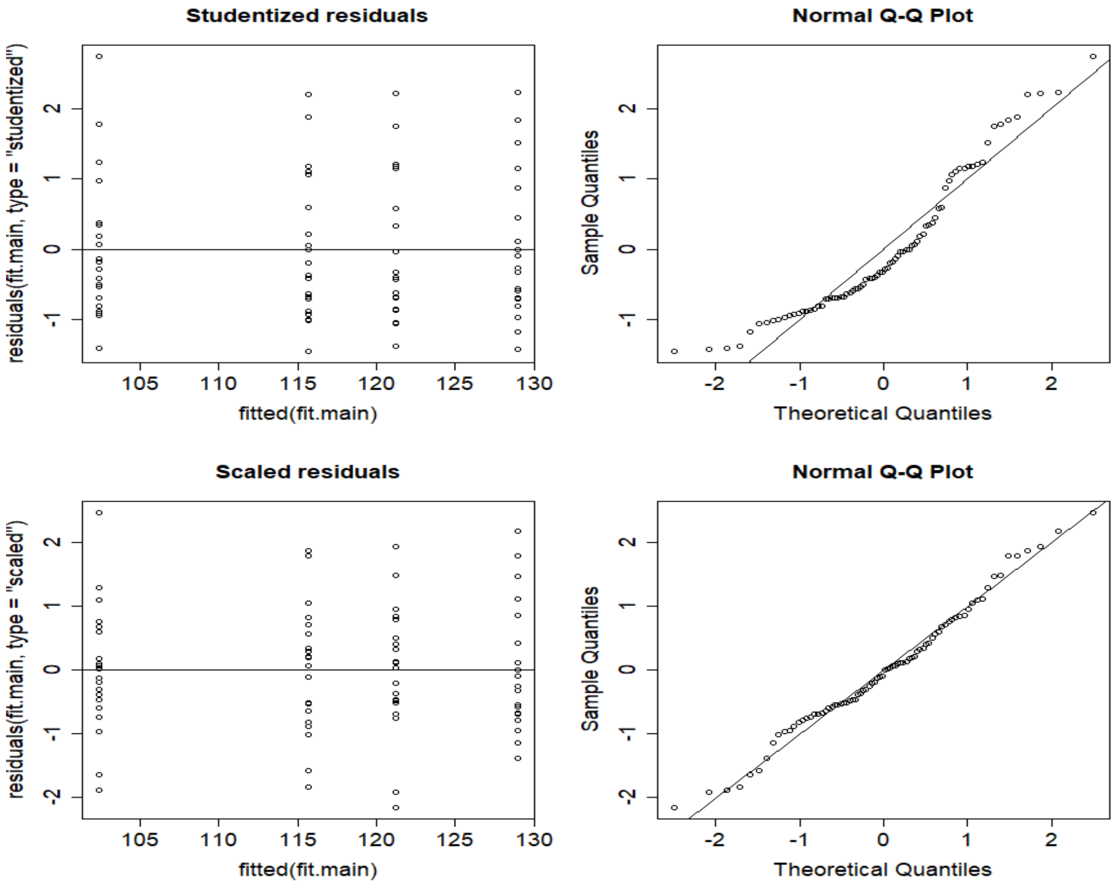


Figure 10: Studentized and scaled residuals from the linear mixed model analysis of the gastric bypass study.

To identify which subjects the outlying residuals belong to, we could match their indices with the rows in the long data. However, the most extreme outliers should be obvious already in the spaghettiplots which are easier to interpret since data is shown on its original scale. In this case the residual values > 2 all belong to the same person (the heaviest participant in the study).

References

Fitzmaurice, G. M.; Laird, N. M. & Ware, J. H. (2012). *Applied longitudinal analysis*, volume 998. John Wiley & Sons.

Hrong-Tai Fai, A. & Cornelius, P. L. (1996). "Approximate F-tests of multiple degree of freedom hypotheses in generalized least squares analyses of unbalanced split-plot experiments". *Journal of statistical computation and simulation*, 54(4):363–378.

Jorsal, T.; Christensen, M. M.; Mortensen, B.; Nygaard, E. B.; Zhang, C.; Rigbolt, K.; Wandall, E.; Langholz, E.; Friis, S.; Worm, D. et al. (2020). "Gut Mucosal Gene Expression and Metabolic Changes After Roux-en-Y Gastric Bypass Surgery". *Obesity*, 28(11):2163–2174.

Kenward, M. G. & Roger, J. H. (2009). "An improved approximation to the precision of fixed effects from restricted maximum likelihood". *Computational Statistics & Data Analysis*, 53(7):2583–2595.

Pinheiro, J. & Bates, D. (2006). *Mixed-effects models in S and S-PLUS*. Springer Science & Business Media.

Pinheiro, J.; Bates, D.; DebRoy, S.; Sarkar, D. & R Core Team (2019). `nlme`*: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-140.

A Appendix: Importing data from an external file.

Rather than creating data from within R, we usually import data from an external file. If you prefer this option, a text file version of the gastric bypass data can be found at the course webpage together with the program files.

To read the data from `gastricbypass.txt`, first download it to your computer. Next run the code below. Please note that you have to change the path in the program so that it matches the *working directory* where you have stored the file.

```
# Set working directory:
setwd("C:/Documents/teaching/repeated/casestudies/gastricbypass")

# Read data from gastricbypass.txt:
wide <- read.table("gastricbypass.txt", header=TRUE, na.strings=".")

# Change id to a categorical variable:
wide$id <- factor(wide$id)
```

Note that missing data appear as .'s in the datafile. The argument `na.strings="."` ensure that they will be represented appropriately with `NA`s in R. The argument `header=TRUE` tells R that the first line in the datafile contains the names of the variables.

B Appendix: Random effect and compound symmetry*.

In section 6.6 we considered the more restrictive compound symmetry pattern, which is equivalent to having a random effect of subject in a linear mixed model.

The direct specification would define the linear mixed model as:

$$Y_{ij} = \beta_1 X_{ij1} + \ldots + \beta_m X_{ijm} + \varepsilon_{ij}$$

where for each subject $i \in \{1, \ldots, n\}$, the residual vector $(\varepsilon_{i1}, \ldots, \varepsilon_{ik})$ follows a multivariate normal distribution with mean 0 and covariance:

$$\Sigma_{CS} = \begin{pmatrix} \sigma^2 & \sigma^2\rho & \ldots & \sigma^2\rho & \sigma^2\rho \\ \sigma^2\rho & \sigma^2 & \ldots & \sigma^2\rho & \sigma^2\rho \\ \vdots & & \ddots & & \vdots \\ \sigma^2\rho & \sigma^2\rho & \ldots & \sigma^2 & \sigma^2\rho \\ \sigma^2\rho & \sigma^2\rho & \ldots & \sigma^2\rho & \sigma^2 \end{pmatrix}$$

We can introduce $\tau = \sigma^2\rho$ and $\sigma_*^2 = \sigma^2 - \tau$. If $\rho > 0$, we have $\sigma_*^2 = \sigma^2(1-\rho) > 0$ so both τ and σ_*^2 are strictly positive. Hence, we can re-express the covariance matrix as:

$$\Sigma_{CS} = \begin{pmatrix} \tau + \sigma_*^2 & \tau & \ldots & \tau & \tau \\ \tau & \tau + \sigma_*^2 & \ldots & \tau & \tau \\ \vdots & & \ddots & & \vdots \\ \tau & \tau & \ldots & \tau + \sigma_*^2 & \tau \\ \tau & \tau & \ldots & \tau & \tau + \sigma_*^2 \end{pmatrix} = \tau J + \sigma_*^2 I$$

where I is the identity matrix and J is a matrix of 1s. This leads to the random intercept model

$$Y_{ij} = \beta_1 X_{ij1} + \ldots + \beta_m X_{ijm} + u_i + \xi_{ij}$$

where the random effect u_i is normally distributed with mean 0 and variance τ and independent of the residuals ξ_{ij}. These residuals are independent and normally distributed with variance σ_*^2.

We demonstrate the equivalence between the two models on our case study. We use the `lme` function from the nlme package to fit the random effects model:

```
fit.rintercept <- lme(weight~time,
                      random=~1|id,
                      data = long)
```

and get the following estimates for $\sqrt{\tau}$ and $\sqrt{\sigma_*^2}$:

```
 Random effects:
  Formula: ~1 | id
         (Intercept) Residual
 StdDev:    18.37874 3.246283
```

Next, we run `lmm` with`structure = "CS"` to fit the compound symmetry pattern:

```
fit.CS <- lmm(weight~time,
              repetition=~visit|id,
              structure="CS",
              data=long)
```

We obtain the following estimates for σ^2 and ρ:

```
> sigma2 <- coef(fit.CS, effect = "variance")^2
> rho <- coef(fit.CS, effect = "correlation")
> c(sigma2,rho)
      sigma          Rho
348.3162479   0.9697449
```

We verify that the log-Likelihood are the same for the two models:

```
>logLik(fit.rintercept)
’log Lik.’ -249.5052 (df=6)
> logLik(fit.CS)
[1] -249.5052
```

Using that $\sqrt{\tau} = \sqrt{\sigma^2\rho}$ and $\sqrt{\sigma_*^2} = \sqrt{\sigma^2(1-\rho)}$, we can match the covariance parameters of the random effects model:

```
> c(sqrt(sigma2*rho),sqrt(sigma2*(1-rho)))
    sigma     sigma
18.378735  3.246283
```

C Appendix: What is in the model summary from `lmm`?

The `summary`-function produces a lot of output. Here we will comment on it, bit by bit.

First the output tells us what dataframe the analysis has been applied to. R makes a summary of how many subjects there are in the data and how many observation there are per cluster and in total.

```
Dataset: long

  - 20 clusters
  - 80 observations
  - 4 observations per cluster
```

Next, we get summary statistics for the outcome and covariates in the model:

```
Summary of the outcome and covariates:

    $ weight: num  127 165 110 146 113 ...
    $ time  : Factor w/ 4 levels "-3 month","-1 week",..: 1 1 1 1 1 1 1 1 1 1 ..
    reference level: time=-3 month
```

If any of the covariates are categorical, we are reminded what the reference levels are.

The next part of the output tells us what numerical optimisation R has performed:

```
Estimation procedure

  - Restricted Maximum Likelihood (REML)
  - log-likelihood :-223.3793
  - parameters: mean = 4, variance = 4, correlation = 6
```

Restricted maximum likelihood (REML) is recommended for mixed models so this is all good. The deviance (`log-likelihood`) provides an overall goodness of fit measure for the model.

After this, we get estimates of the covariance parameters starting with the correlations:

```
Residual variance-covariance: unstructured

  - correlation structure: ~visit | id
          1     2     3     4
    1 1.000 0.990 0.986 0.946
    2 0.990 1.000 0.997 0.959
    3 0.986 0.997 1.000 0.966
    4 0.946 0.959 0.966 1.000
```

If you compare these to the correlation coefficients we computed as descriptive statistics, you will see that they are just the same. This will happen any time you have complete data from a balanced single group study. The same occurs for the estimated standard deviations (save from a tiny numerical error):

```
  - variance structure: ~visit
      standard.deviation     ratio
    1           20.26942 1.0000000
    2           18.91022 0.9329435
    3           18.27535 0.9016220
    4           17.05391 0.8413614
```

Note that the column `ratio` compares the standard deviations at follow-up to the standard deviation at -3 months (reference point). We see that the standard deviation at -1 week is a factor 0.93 smaller, and so forth.

The most interesting part of the summary is of course the estimates for the effect of the covariates; This is the part of the summary output which we commented on in section 7.4:

```
Fixed effects: weight ~ time

             estimate    se     df   lower   upper p.value    
(Intercept)    128.97 4.532 18.981 119.483 119.483  <0.001 ***
time-1 week     -7.73 0.697 18.974   -9.19   -9.19  <0.001 ***
time+1 week    -13.27 0.839 18.969 -15.027 -15.027  <0.001 ***
time+3 month  -26.605 1.549 18.964 -29.848 -29.848  <0.001 ***

The columns lower and upper correspond to the 95% confidence interval
Degrees of freedom were computed using a Satterthwaite approximation
```

Along with the estimates we get standard error, t-test statistics, and a p-values for testing the hypothesis that the parameter in consideration is zero. It is important to know that degrees of freedom used to evaluate the t-tests are computed according to the Satterthwaite approximation. Without this approximation, the confidence intervals would be too narrow and the p-values would be too small, see sections 6.9 – 6.10.

D Appendix: Analysis of log-transformed data.

We previously noted that the distribution of the bodyweights is skewed with few very heavy subjects, a majority of moderately heavy subjects, and no normal weight subjects. Also the weights of the most heavy subjects appeared more variable over time than those of the lightest subjects in the sample. Finally there was a trend that the standard deviations decreased along with the means. All of this speaks in favour of a logarithmic-transformation.

Overall recommendations on log-transformation is to use it when:

- It is desirable that changes over time or differences between treatments should be estimated in relative terms, i.e. as percentwise changes or differences.
- It is a substantial improvement on the original model fit.
- There is precedence to transform outcomes of this particular type in the literature.

If you are in doubt whether your data should be transformed it is usually better to stick to the original scale. Otherwise consult with a statistician about it.

Figure 11 below compares the spaghettiplot of log2-transformed data with that of the original data and figure 12 shows the scatterplot matrix for the log2-transformed data. The distribution of the log2-transformed data is more symmetric and therefore closer to a normal distribution than the original data.

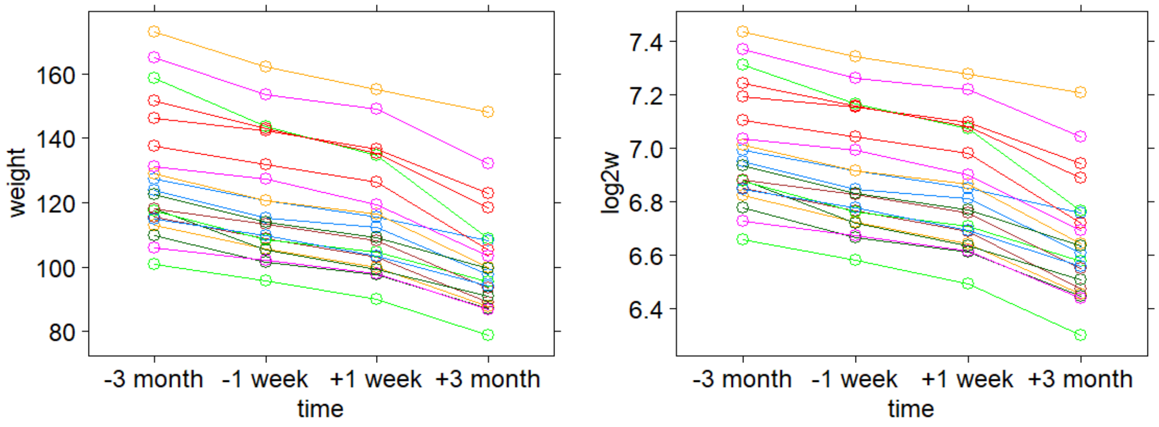


Figure 11: Bodyweights from the gastric bypass study on original scale (left) and after log2-transformation (right).

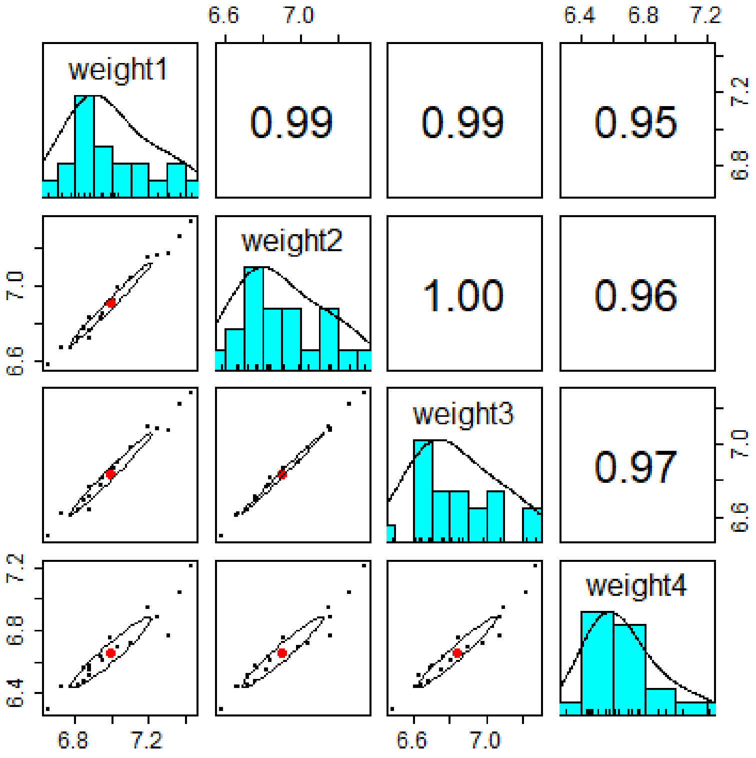


Figure 12: Scatterplot matrices of log2-transformed weights from the gastric bypass study. This should be compared with figure 4 showing the data on the original scale.

.

Making an analysis of log-transformed data is similar to making an analysis of untransformed data save from the first and the last step in the analysis. The first step would be to add the log-transformed outcome to the long data and review the spaghettiplot:

```
# Add log-transformed outcome to long data:
long$log2weight <- log2(long$weight)

# Spaghettiplot:
xyplot(log2w~time, data=long, group=id, type="b")
```

Then we redo the mixed model analysis. Please note that the R code is identical to that in section 7.1 save from the name of the model object and the outcome:

```
fit.log <- lmm(log2weight~time,
               repetition=~visit|id,
               structure="UN",
               df=TRUE,
               data=long)
```

From here on the analysis proceeds as in section 7. After we extract the estimates and confidence intervals for reporting, the final step is to back-transform for interpretation:

```
# Extract and back-transform estimates and CIs:
2^confint(fit.log)

                 estimate       lower       upper
(Intercept)   127.5466585 118.8507570 136.8788092
time-1 week     0.9401516   0.9304585   0.9499458
time+1 week     0.8969349   0.8863761   0.9076195
time+3 month    0.7929295   0.7753979   0.8108574
```

Hence, we estimate that median bodyweight has decreased by -6.0% (95% CI: -6.9% to -5.1%) at first follow-up, by -10.3% (95% CI: -11.3% to -9.3%) at second follow-up, and by -20.7% (95% CI: -22.4% to -19.0%) at end of study.

Similarly, we could obtain estimates of means over time on log2-scale and back-transform them into estimated medians on the original scale.[3] Note that the estimated median bodyweight at baseline is 127.5 kg (95% CI: 119.3 to 136.4 kg). This is smaller than the estimated mean of 129.0 kg we found when analyzing the data on the original scale, albeit not much smaller as the distribution of the bodyweights is only moderately skew.

To assess whether the linear mixed model fits the log2-transformed data well we inspect the residual plots (figure 13 on next page).

[3] Since transformation preserves quantiles (including the median) and since in a symmetric distribution (including the normal distribution) the mean is the same as the median. In case data on log-scale is only approximately normal (symmetric), back-transformed means will be *geometric means* on the original scale.

```
# Residualplots for model validation:
par(mfrow=c(2,2))
plot(fitted(fit.log), residuals(fit.log, type="studentized"))
abline(h=0)
qqnorm(residuals(fit.log, type="studentized"))
abline(0,1)
plot(fitted(fit.log), residuals(fit.log, type="scaled"))
abline(h=0)
qqnorm(residuals(fit.log, type="scaled"))
abline(0,1)
```

After the transformation, the residuals show a slightly better fit to the normal distribution.

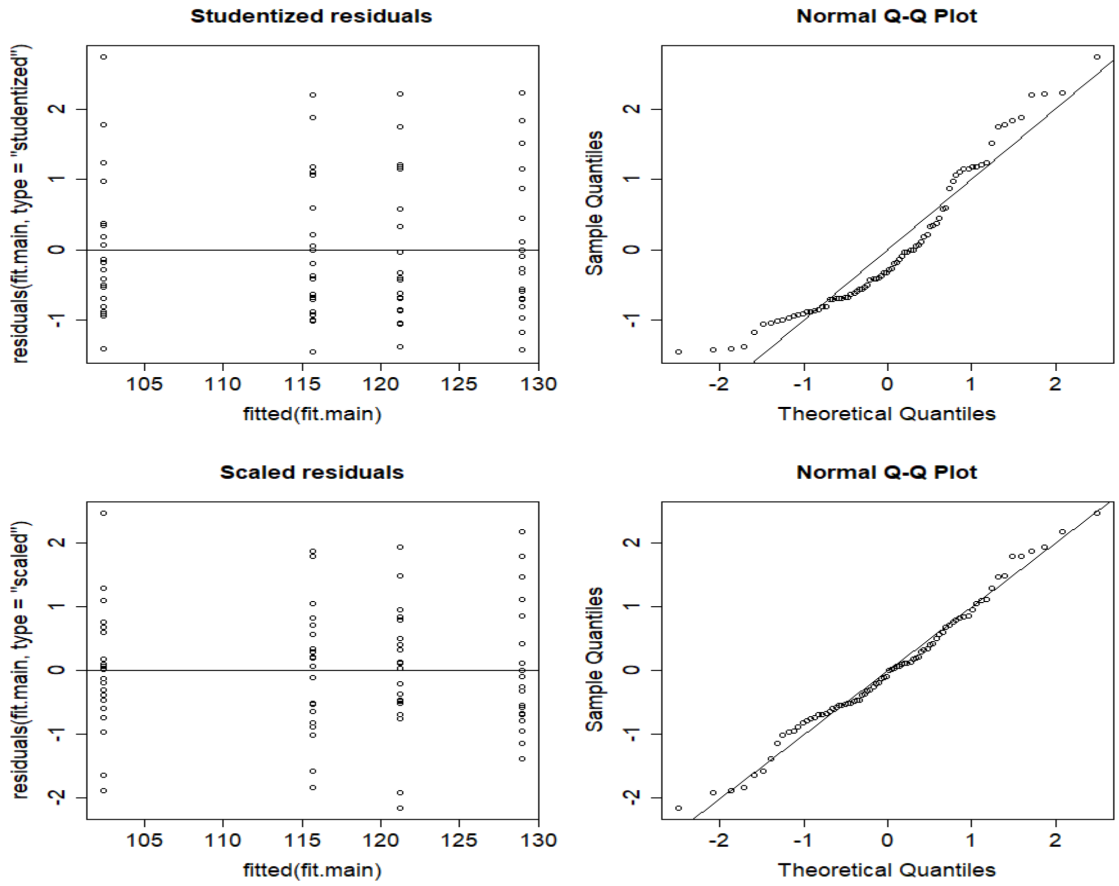


Figure 13: Residuals from the mixed model analysis of the log2-transformed data.