

# **Longitudinal Data Modeling**

Dr. Marietta Kirchner      Dr. Alexandra Lauer

September 24, 2024

# Table of contents

<b>Preface</b>	<b>4</b>
<b>1 Introduction</b>	<b>5</b>
1.1 Workshop Structure . . . . .	5
1.2 Longitudinal Data . . . . .	5
1.3 Basics about RStudio (pre-read) . . . . .	6
1.4 Example data . . . . .	6
<b>2 Longitudinal Data Exploration and Visualization</b>	<b>7</b>
2.1 Introduction . . . . .	7
2.2 Task 1 - Exploration of dataset all2 - 15 Minutes working time . . . . .	7
2.2.1 Task 1 Discussion, possible solution . . . . .	7
<b>3 Correlation structure, covariance matrices</b>	<b>11</b>
3.1 Overview - different covariance matrices . . . . .	11
3.1.1 Independence structure (VC) . . . . .	11
3.1.2 Compound symmetry (CS) . . . . .	11
3.1.3 Unstructured (UN) . . . . .	12
3.1.4 Toeplitz structure (TOEP) . . . . .	12
3.1.5 Autoregressive structure (AR(1)) . . . . .	12
3.1.6 Spatial (SP) . . . . .	12
3.2 Selecting the covariance structure . . . . .	13
3.3 Task - Exploration of correlation in the data . . . . .	13
<b>4 Inference from Longitudinal Data</b>	<b>14</b>
4.1 Categorical Time . . . . .	15
4.1.1 Unstructured (US) . . . . .	15
4.1.2 Compound Symmetry (CS) . . . . .	18
4.1.3 Toeplitz (TOEP) . . . . .	19
4.1.4 Spatial Power (SP_EXP) . . . . .	20
4.1.5 Conclusion . . . . .	22
4.2 Continuous Time . . . . .	22
4.3 Baseline as a Response (cLDA + LDA) . . . . .	23
4.4 (Adjusted) LS Means from MMRMs . . . . .	23
4.4.1 Observed vs. balanced margins . . . . .	25
4.4.2 Contrasts . . . . .	29

4.5	Fit diagnostics . . . . .	31
4.5.1	Addendum on RS&I Models . . . . .	36
<b>5</b>	<b>Missing Data</b>	<b>37</b>
5.1	Missing Data Mechanisms . . . . .	39
5.2	Missing data handling I (descriptive stats + visualisations) . . . . .	40
5.3	Missing data handling II (analytic approaches) . . . . .	40
<b>6</b>	<b>Sensitivity Analyses</b>	<b>41</b>
6.1	Simple approaches . . . . .	41
6.2	Missing covariates (baseline data) . . . . .	41
6.3	Baseline complete, missingness in post-baseline values . . . . .	42
6.4	Multiple imputation . . . . .	42
6.5	Handling nonignorable missingness (MNAR) . . . . .	42
<b>7</b>	<b>Inferences from binary longitudinal data</b>	<b>43</b>
	<b>References</b>	<b>44</b>

# Preface

# 1 Introduction

## 1.1 Workshop Structure

This class focuses on the longitudinal modeling of data from Patient Reported Outcomes (PROs). It is meant to be hands-on class with applications in R.

Content and structure follow the book by (Mallinckrodt 2016). We would like to extend our warmest gratitude towards Dr. Mallinckrodt for providing the example data for the workshop.

The following topics will be covered:

- Welcome and Introduction (WS session 1)
- Exploration and visualization of longitudinal data (WS session 1/2)
- Inferences from longitudinal data (WS session 3 + 4)
- Assessment of missingness patterns (WS session 5)
- Sensitivity analyses to assess the impact of missingness (WS session 6)
- Annex: Inferences from longitudinal binary data (WS session 7)

## 1.2 Longitudinal Data

This workshop focuses on the analysis of data observed in randomized clinical trials (RCTs). Here, patients have assessments taken at the start of their treatment and then subsequently throughout the course of the trial based on a pre-specified schedule of assessments. The measurement at the start of the treatment is usually referred to as the baseline.

Researchers can be interested in

1. the occurrence of a certain event during the course of the trial, e.g. death or a cardiac event, or the time to the occurrence of such an event, or
2. the longitudinal profile from multiple repeated measurements taken, with a focus on either estimates at a landmark visit or across several time points.

The outcomes under point 1. can be handled via a comparison of the percentages of patients with events between treatment arms, or a time-to-event analysis. Both are out of scope of this workshop.

## **1.3 Basics about RStudio (pre-read)**

Alex to add pre-read (YouTube + Cheat Sheets)

## **1.4 Example data**

## 2 Longitudinal Data Exploration and Visualization

### 2.1 Introduction

- Data on individuals followed over time with information collected at several time points.
- Clusters are the individuals who are followed over time.
- Repeated observations may or may not be taken at regular times (balanced, fixed occasions, do not differ between subjects).
- Our interest is in the change from baseline.

Datasets used in this course: Example data is taken from (Mallinckrodt 2016). Contain data on the HAMD17 (Hamilton 17-item rating scale for depression). Two treatment arms are included: drug vs. placebo. Assessments were taken at baseline and weeks 1, 2, 4, 6, and 8

There are 3 data sets created from the original data: - *all2* Subsample of the large dataset with n=50, visits: weeks 2, 4, 8 - *high2* Large dataset with n=100, high dropout = 70% (drug), 60% (placebo) - *low2* Large dataset with n=100, low dropout = 18%

### 2.2 Task 1 - Exploration of dataset all2 - 15 Minutes working time

- Are the data balanced and equally spaced?
- Number of observations by week?
- Summary statistics for HAMD17 (change from baseline) by week.
- Plot trajectories for each individual, different colors for each treatment group (or panels). Add mean to your plot. ODER Mean plot separat evtl. mit CI
- Generate and interpret the group-wise boxplots of the change from baseline.
- Plot trajectories separately by gender. Comment on the plots.

#### 2.2.1 Task 1 Discussion, possible solution

TO DO: Datensatz aufbereiten, Label, Time=Faktor (t) Alex: library gtsummary okay? Oder sollen wir es anders machen

Table: Summary statistics for HAMD17 by treatment and week in the all2 dataset

```
all2 %>%
  select(change, group, avisit) %>%
  tbl_strata(strata=group,
    ~.x %>%
      tbl_summary(by = avisit,
        statistic = list(
          all_continuous() ~ "{mean} ({sd})",
          digits = all_continuous() ~ 2 )
      )
  )
```

Table printed with `knitr::kable()`, not {gt}. Learn why at <https://www.danielsjoberg.com/gtsummary/articles/rmarkdown.html>  
 To suppress this message, include `message = FALSE` in code chunk header.

	Week 2, N = 25	Week 4, N = 25	Week 8, N = 25	Week 2, N = 25	Week 4, N = 25	Week 8, N = 25
change	-4.20 (3.66)	-6.80 (4.25)	-9.88 (4.85)	-5.24 (5.49)	-8.60 (5.39)	-13.24 (5.54)

Figure: trajectories

```
ggplot(data = all2, aes(x = avisit, y = change, group=subject)) +
  geom_point() + geom_line() + facet_grid(.~group)
```



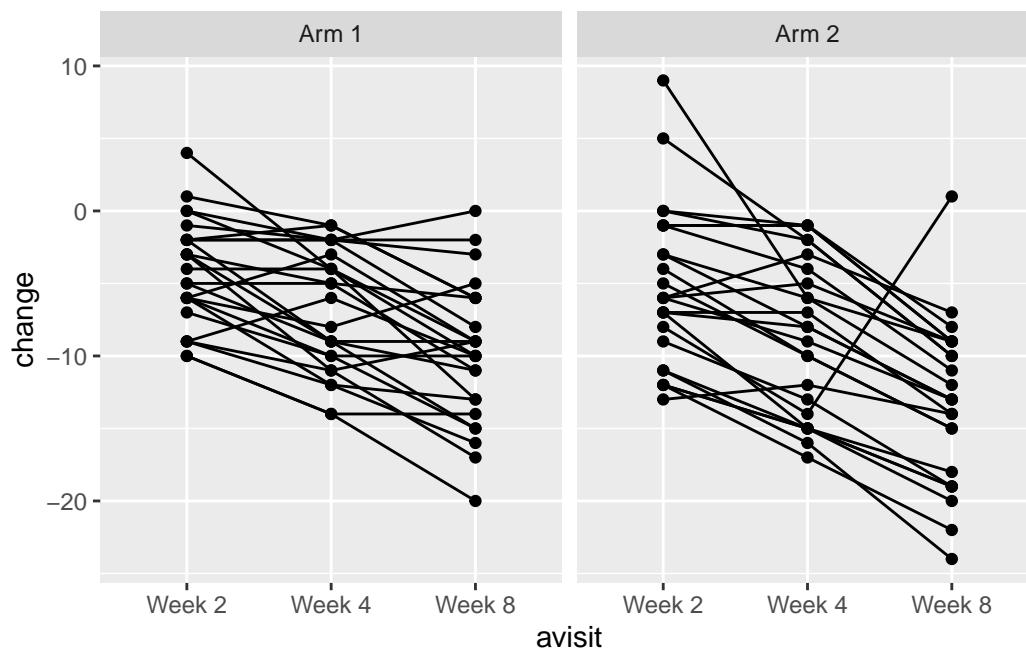


Figure 2.1: Individual trajectories of HAMD17 by treatment group

```
ggplot(data = all2, aes(x = avisit, y = change)) +
  geom_point() + facet_grid(.~group) +
  stat_summary(aes(group = 1), geom = "line", fun.y = mean,
    size = 1, col="blue") +
  stat_summary(aes(group = 1), geom = "point", fun.y = mean,
    shape=17,size = 3, col="blue")
```

Warning: The ``fun.y`` argument of ``stat_summary()`` is deprecated as of ggplot2 3.3.0.  
i Please use the ``fun`` argument instead.

Warning: Using ``size`` aesthetic for lines was deprecated in ggplot2 3.4.0.  
i Please use ``linewidth`` instead.

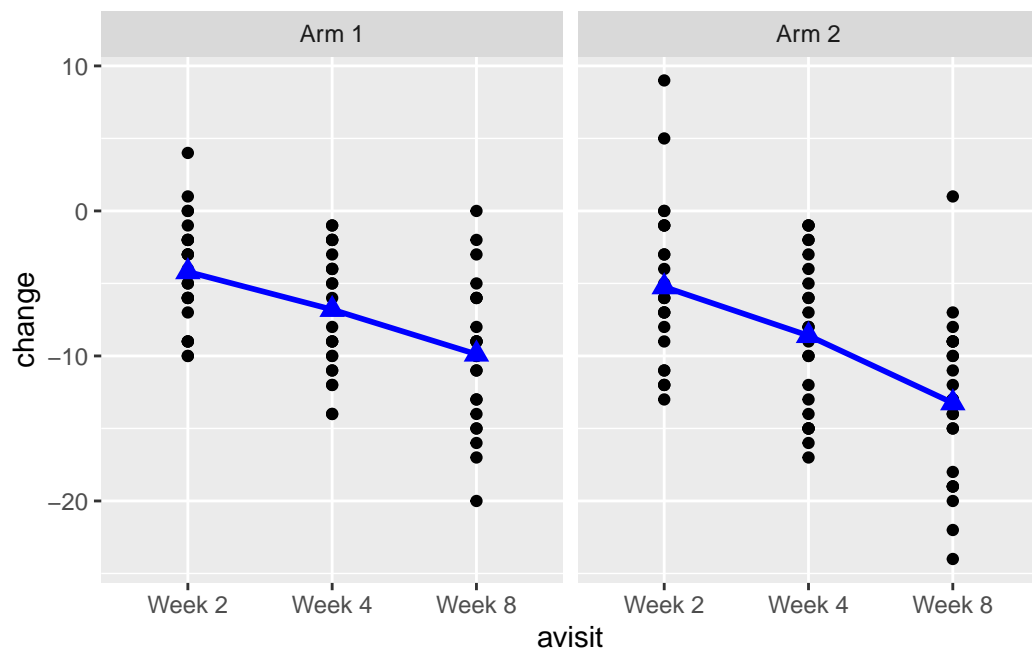


Figure 2.2: Mean change from baseline by treatment group

## 3 Correlation structure, covariance matrices

- model within-subject error correlation
- different residual covariance structures can be implemented

### 3.1 Overview - different covariance matrices

- Variance components (VC) independence structure
- Compound symmetry (CS) also known as exchangeable
- Toeplitz (TOEP)
- First order auto regressive (AR(1))
- Unstructured (UN)

Selected covariance structures for data with three assessment times are shown below. Note that with three assessment times, the number of parameters estimated for the various structures did not differ as much as would be the case with more assessment times. Thus, results from different covariance structures are more similar than would be the case with more assessment times.

#### 3.1.1 Independence structure (VC)

Constant variance. It is assumed to be no correlation between assessments.

$$\begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$$

#### 3.1.2 Compound symmetry (CS)

Constant variance and constant covariance across all assessments. Also known as exchangeable.

$$\begin{bmatrix} \sigma^2 + \sigma_1 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma^2 + \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma^2 + \sigma_1 \end{bmatrix}$$

### 3.1.3 Unstructured (UN)

This is the most general (saturated) model

$$\begin{bmatrix} \sigma_1^2 & \sigma_{21} & \sigma_{31} \\ \sigma_{21} & \sigma_2^2 & \sigma_{32} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{bmatrix}$$

### 3.1.4 Toeplitz structure (TOEP)

Homogenous variances and heterogenous correlations. Same correlation value is used whenever the degree of adjacency is the same e.g. correlation between times 1 and 2 = correlation between times 2 and 3.

$$\begin{bmatrix} \sigma^2 & \sigma^2\rho_1 & \sigma^2\rho_2 \\ \sigma^2\rho_1 & \sigma^2 & \sigma^2\rho_1 \\ \sigma^2\rho_2 & \sigma^2\rho_1 & \sigma^2 \end{bmatrix}$$

### 3.1.5 Autoregressive structure (AR(1))

Correlation decreases as time between observations increases.

$$\begin{bmatrix} \sigma^2 & \sigma^2\rho & \sigma^2\rho^2 \\ \sigma^2\rho & \sigma^2 & \sigma^2\rho \\ \sigma^2\rho^2 & \sigma^2\rho & \sigma^2 \end{bmatrix}$$

### 3.1.6 Spatial (SP)

Spatial covariance structures does not require that the timepoints are consistent between subjects. Instead, as long as the distance between visits can be quantified in terms of time and/or other coordinates, the spatial covariance structure can be applied.

For spatial exponential, the covariance structure is defined as follows:

$$\rho_{ij} = \rho^{d_{ij}}$$

where

$$d_{ij}$$

is the distance between time point i and time point j.

## 3.2 Selecting the covariance structure

There are a variety of considerations when selecting the covariance structure: - number of parameters - interpretation of the structure - model fit UN is the most flexible. Choose a reasonable covaraiance structure which is the best compromise between model fit and complexity. E.g. use AIC as it penalises more complex models.

## 3.3 Task - Exploration of correlation in the data

- Compute the empirical correlations between measurement timepoints (e.g. correlation between baseline and post-baseline changes).
- Looking at these correlations, comment on the suitability of the correlation structures VC, CS, UN, AR(1).

1 + 1

[1] 2

## 4 Inference from Longitudinal Data

This section will focus on the application of Mixed Model with Repeated Measures (MMRMs). Our main focus will be the modeling of the means of the data. MMRMs are generalizations of standard linear models in the way that data is allowed to be correlated between subsequent measurements from the same subject and exhibit non-constant variability.

The primary assumptions for MMRMs are:

- The data are normally distributed
- The means (expected values) of the data are linear in terms of a certain set of parameters.
- The variances and covariances of the data are in terms of a different set of parameters, and they exhibit a structure matching one of those outlined in the former chapter.

[Alex to add reference to PROC MIXED]

The mixed linear model can be described via the following formula

$$y_i = X_i\beta + Z_i\gamma_i + \varepsilon_i, i = 1, \dots, N$$

where  $y$  is the vector of responses (observed data, dependent variable),  $\beta$  is an unknown vector of fixed effects with known design matrix  $X$ ,  $\gamma$  is an unknown vector of random effects with known design matrix  $Z$ , and  $\varepsilon$  is an unknown random error vector. Furthermore  $N$  denotes the total number of subjects in our analysis. For the sake of readability, we will omit the subject index and simplify the above formula to

$$y = X\beta + Z\gamma + \varepsilon.$$

We will further assume that  $\gamma$  and  $\varepsilon$  are uncorrelated Gaussian random variables with expectation 0 and variances  $G$  and  $R$ , respectively. Then the variance-covariance matrix of  $y$  is given by

$$\text{Var}(y) := V = ZGZ' + R.$$

In this case  $ZGZ'$  comprises the random effects component, and  $R$  is the within-subject component.

In this workshop we will focus on the case where only the within-subject component is accounted for, via modeling of the  $R$  matrix. The random effects component  $Z\gamma$  will be omitted. In this case we will have  $\text{Var}(y) = V = R$ , resulting in a model given by

$$y = X\beta + \varepsilon.$$

## 4.1 Categorical Time

You can start and familiarise yourself with the main function `mmrm()` using the command

```
library(mmrm)
?mmrm
```

Two inputs are strictly required to get `mmrm()` to work:

- A model formula
- The dataset, containing the response, as well as all fixed effects and variables in the covariance matrix.

**Exercise:** Fit a model `fit_cat_time` using the dataset `all2`, with `change` as dependent variable, baseline value, visit, baseline by visit interaction and treatment by visit interaction as fixed effects and an unstructured covariance matrix for visits within each subject.

- How do different choices for covariance matrices change the results? What is the difference on the estimation procedure?
- You can obtain a summary of the fit results via `summary(fit_cat_time)`. How do you interpret the fit summary?
- Look at the structure of the fit summary and try to extract the estimate of the  $R$  matrix.
- How do other choices of variance-covariance structures influence the estimation?

### 4.1.1 Unstructured (US)

Unstructured corresponds to a saturated variance-covariance matrix and involves the estimation of  $m(m+1)/2$  variance components, where  $m$  is the number of follow-up visits. In our case, we can see that a total of 6 variance parameters were estimated.

```
fit_cat_time <- mmrm::mmrm(
  formula = change ~ basval*avisit + trt*avisit + us(avisit | subject),
  data = all2,
```

```

    control = mmrm_control(method = "Kenward-Roger")
  )

summary(fit_cat_time)

```

mmrm fit

```

Formula:      change ~ basval * avisit + trt * avisit + us(avisit | subject)
Data:         all2 (used 150 observations from 50 subjects with maximum 3
timepoints)
Covariance:   unstructured (6 variance parameters)
Method:       Kenward-Roger
Vcov Method:  Kenward-Roger
Inference:    REML

```

Model selection criteria:

AIC	BIC	logLik	deviance
822.4	833.9	-405.2	810.4

Coefficients:

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	1.98452	3.27479	47.00000	0.606	0.54743
basval	-0.31235	0.15905	47.00000	-1.964	0.05548 .
avisitWeek 4	-0.90862	2.39866	47.00000	-0.379	0.70654
avisitWeek 8	-10.58630	3.45922	47.00000	-3.060	0.00365 **
trt2	-1.18993	1.27265	47.00000	-0.935	0.35457
basval:avisitWeek 4	-0.08542	0.11650	47.00000	-0.733	0.46704
basval:avisitWeek 8	0.24779	0.16801	47.00000	1.475	0.14691
avisitWeek 4:trt2	-0.80100	0.93217	47.00000	-0.859	0.39454
avisitWeek 8:trt2	-2.20106	1.34432	47.00000	-1.637	0.10825

---

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

Covariance estimate:

	Week 2	Week 4	Week 8
Week 2	20.6112	15.3034	12.2766
Week 4	15.3034	21.3565	17.6648
Week 8	12.2766	17.6648	27.6127

We can assess the structure of the fit summary via



```
str(summary(fit_cat_time))
```

List of 15

```
$ cov_type      : chr "us"
$ reml          : logi TRUE
$ n_groups      : int 1
$ n_theta       : int 6
$ n_subjects    : int 50
$ n_timepoints  : int 3
$ n_obs         : int 150
$ beta_vcov     : num [1:9, 1:9] 10.724 -0.501 -2.675 -4.267 -1.047 ...
..- attr(*, "dimnames")=List of 2
.. ..$ : chr [1:9] "(Intercept)" "basval" "avisitWeek 4" "avisitWeek 8" ...
.. ..$ : chr [1:9] "(Intercept)" "basval" "avisitWeek 4" "avisitWeek 8" ...
$ varcor        : num [1:3, 1:3] 20.6 15.3 12.3 15.3 21.4 ...
..- attr(*, "dimnames")=List of 2
.. ..$ : chr [1:3] "Week 2" "Week 4" "Week 8"
.. ..$ : chr [1:3] "Week 2" "Week 4" "Week 8"
$ method        : chr "Kenward-Roger"
$ vcov          : chr "Kenward-Roger"
$ coefficients   : num [1:9, 1:5] 1.985 -0.312 -0.909 -10.586 -1.19 ...
..- attr(*, "dimnames")=List of 2
.. ..$ : chr [1:9] "(Intercept)" "basval" "avisitWeek 4" "avisitWeek 8" ...
.. ..$ : chr [1:5] "Estimate" "Std. Error" "df" "t value" ...
$ n_singular_coefs: int 0
$ aic_list       :List of 4
..$ AIC          : num 822
..$ BIC          : num 834
..$ logLik       : num -405
..$ deviance     : num 810
$ call           : language mrm::mrm(formula = change ~ basval * avisit + trt * avisit +
- attr(*, "class")= chr "summary.mrm"
```

and then extract the covariance matrix

```
summary(fit_cat_time)$varcor
```

```
      Week 2   Week 4   Week 8
Week 2 20.61117 15.30339 12.27661
Week 4 15.30339 21.35648 17.66478
Week 8 12.27661 17.66478 27.61271
```

### 4.1.2 Compound Symmetry (CS)

We can choose different types of covariance structures by modification of the model formula.

The compound symmetry structure assumes equal variances (diagonal elements are all equal) and equal covariances (off-diagonal elements are all equal). From the model summary we can see that two variance-covariance parameters are estimated.

This model is the most simple choice of repeated measures variance-covariance modeling. In most cases, it is overly simplistic, but can be a good fallback option in case of model non-convergence (especially when prespecification of analysis methods is required).

```
fit_cat_time_cs <- mmrm::mmrm(  
  formula = change ~ basval*avisit + trt*avisit + cs(avisit | subject),  
  data = all2,  
  control = mmrm_control(method = "Kenward-Roger")  
)  
  
summary(fit_cat_time_cs)
```

mmrm fit

Formula: change ~ basval \* avisit + trt \* avisit + cs(avisit | subject)  
Data: all2 (used 150 observations from 50 subjects with maximum 3 timepoints)  
Covariance: compound symmetry (2 variance parameters)  
Method: Kenward-Roger  
Vcov Method: Kenward-Roger  
Inference: REML

Model selection criteria:

AIC	BIC	logLik	deviance
827.2	831.0	-411.6	823.2

Coefficients:

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	1.98452	3.48848	76.39000	0.569	0.571107
basval	-0.31235	0.16943	76.39000	-1.844	0.069126 .
avisitWeek 4	-0.90862	2.91606	94.00000	-0.312	0.756042
avisitWeek 8	-10.58630	2.91606	94.00000	-3.630	0.000461 ***
trt2	-1.18993	1.35569	76.39000	-0.878	0.382845
basval:avisitWeek 4	-0.08542	0.14163	94.00000	-0.603	0.547856
basval:avisitWeek 8	0.24779	0.14163	94.00000	1.750	0.083448 .

```

avisitWeek 4:trt2      -0.80100      1.13324  94.00000  -0.707 0.481424
avisitWeek 8:trt2      -2.20106      1.13324  94.00000  -1.942 0.055098 .
---

```

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

Covariance estimate:

```

      Week 2  Week 4  Week 8
Week 2 23.1948 15.0832 15.0832
Week 4 15.0832 23.1948 15.0832
Week 8 15.0832 15.0832 23.1948

```

### 4.1.3 Toeplitz (TOEP)

Use of the Toeplitz structure is not a very sensible choice here, as visits are not equally spaced, i.e. the difference between baseline and time1, and time1 and time2 is 2 weeks, respectively, while the difference between time2 and time3 is 4 weeks. Toeplitz thus ignores the differences in time spacing.

We can see that the covariance estimates for responses at Week 2 (time1) and Week 4 (time2) are the same as the ones for responses at Week 4 (time2) and Week 8 (time3), although their time difference doubles.

The same line of reasoning for the lack of sensibility of the Toeplitz structure can be applied to the autoregressive structure (AR(1)). The example is not shown here.

```

fit_cat_time_toep <- mmrm::mmrm(
  formula = change ~ basval*avisit + trt*avisit + toep(avisit | subject),
  data = all2,
  control = mmrm_control(method = "Kenward-Roger")
)

summary(fit_cat_time_toep)

```

mmrm fit

```

Formula:      change ~ basval * avisit + trt * avisit + toep(avisit | subject)
Data:         all2 (used 150 observations from 50 subjects with maximum 3
timepoints)
Covariance:    Toeplitz (3 variance parameters)
Method:        Kenward-Roger
Vcov Method:   Kenward-Roger
Inference:     REML

```

Model selection criteria:

AIC	BIC	logLik	deviance
818.6	824.3	-406.3	812.6

Coefficients:

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	1.98452	3.52061	74.67000	0.564	0.57466
basval	-0.31235	0.17099	74.67000	-1.827	0.07174 .
avisitWeek 4	-0.90862	2.56351	92.39000	-0.354	0.72381
avisitWeek 8	-10.58630	3.43000	56.08000	-3.086	0.00315 **
trt2	-1.18993	1.36818	74.67000	-0.870	0.38724
basval:avisitWeek 4	-0.08542	0.12450	92.39000	-0.686	0.49436
basval:avisitWeek 8	0.24779	0.16659	56.08000	1.487	0.14249
avisitWeek 4:trt2	-0.80100	0.99623	92.39000	-0.804	0.42344
avisitWeek 8:trt2	-2.20106	1.33297	56.08000	-1.651	0.10428

---

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

Covariance estimate:

	Week 2	Week 4	Week 8
Week 2	23.6312	17.3491	12.3576
Week 4	17.3491	23.6312	17.3491
Week 8	12.3576	17.3491	23.6312

#### 4.1.4 Spatial Power (SP\_EXP)

The choice of the spatial power variance-covariance structure makes sense here, as the visits are not equally spaced. In this case, two parameters are estimated. The first parameter is the variance (diagonal elements) and second one is the time difference between subsequent visits.

Note that in this example, we need to use the numeric week variable, as spatial power requires the information about the distance of subsequent visits in the estimation of the variance-covariance matrix.

We can see from the fit summary, that the covariance displayed is a 2 \* 2 square matrix. As the distance will be used to derive the corresponding element in that matrix, unit distance is used here.

```
fit_cat_time_sp <- mmrm::mmrm(  
  formula = change ~ basval*avisit + trt*avisit + sp_exp(week | subject),
```

```

data = all2,
control = mmrm_control(method = "Kenward-Roger")
)

summary(fit_cat_time_sp)

```

mmrm fit

Formula: change ~ basval \* avisit + trt \* avisit + sp\_exp(week | subject)  
 Data: all2 (used 150 observations from 50 subjects with maximum 3 timepoints)  
 Covariance: spatial exponential (2 variance parameters)  
 Method: Kenward-Roger  
 Vcov Method: Kenward-Roger  
 Inference: REML

Model selection criteria:

AIC	BIC	logLik	deviance
818.5	822.4	-407.3	814.5

Coefficients:

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	1.98452	3.54179	76.71000	0.560	0.57690
basval	-0.31235	0.17202	76.71000	-1.816	0.07331 .
avisitWeek 4	-0.90862	2.24976	84.19000	-0.404	0.68733
avisitWeek 8	-10.58630	3.52591	117.26000	-3.002	0.00327 **
trt2	-1.18993	1.37641	76.71000	-0.865	0.39000
basval:avisitWeek 4	-0.08542	0.10927	84.19000	-0.782	0.43653
basval:avisitWeek 8	0.24779	0.17125	117.26000	1.447	0.15057
avisitWeek 4:trt2	-0.80100	0.87430	84.19000	-0.916	0.36220
avisitWeek 8:trt2	-2.20106	1.37024	117.26000	-1.606	0.11089

---

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

Covariance estimate:

	0	1
0	23.9079	21.3749
1	21.3749	23.9079

### 4.1.5 Conclusion

While the unstructured variance-covariance matrix provides the highest degree of flexibility, we can see from the AIC and BIC estimates that spatial power in our example provides and even better fit in comparison to the model complexity. Note that this is also true for the Toeplitz structure, but we rejected this approach as the unequal spacing of visits renders this approach nonsensible.

## 4.2 Continuous Time

Time as continuous effect -> single df for time and trt-by-time interaction

Modeling: - Need a visit for structure of covariance matrix - Implicit assumption is for the covariance between values for two timepoints to be equal, regardless of the specific timing

```
fit_cont_time <- mmrm::mmrm(  
  formula = change ~ basval*week + trt*week + us(avisit | subject),  
  weights = all2$week,  
  data = all2,  
  control = mmrm_control(method = "Kenward-Roger")  
)
```

Can also apply non-linear transformations of time variable, in case the anticipated effect is not necessarily linear in time:

```
all2$timesq <- all2$week^2  
  
fit_cont_timesq <- mmrm::mmrm(  
  formula = change ~ basval*timesq + trt*timesq + us(avisit | subject),  
  weights = all2$week,  
  data = all2,  
  control = mmrm_control(method = "Kenward-Roger")  
)
```

model checks - residuals per time point

## 4.3 Baseline as a Response (cLDA + LDA)

## 4.4 (Adjusted) LS Means from MMRMs

LS Means are means of the dependent variable adjusted for covariates in the statistical model. We can obtain LS Means estimates and contrasts allowing for a treatment comparison using the `emmeans` package.

**Example:** Calculate the observed (raw) means of changes along with number of patients by treatment group from the dataset `all2` overall and by visit. Then take the model `fit_cat_time` and derive the respective LS Means from the model. What do you observe?

```
# Raw means

all2 %>%
  dplyr::group_by(group) %>%
  dplyr::summarise(
    N = dplyr::n(),
    Mean = mean(change),
    .groups = "drop"
  )

# A tibble: 2 x 3
  group      N  Mean
<fct> <int> <dbl>
1 Arm 1     75 -6.96
2 Arm 2     75 -9.03

all2 %>%
  dplyr::group_by(group, avisit) %>%
  dplyr::summarise(
    N = dplyr::n(),
    Mean = mean(change),
    .groups = "drop"
  )

# A tibble: 6 x 4
  group avisit      N  Mean
<fct> <fct>  <int> <dbl>
1 Arm 1 Week 2    25  -4.2
```

2	Arm 1	Week 4	25	-6.8
3	Arm 1	Week 8	25	-9.88
4	Arm 2	Week 2	25	-5.24
5	Arm 2	Week 4	25	-8.6
6	Arm 2	Week 8	25	-13.2

The respective LS Means from the model with time as a fixed factor yields the following estimates:

```
library(emmeans)

emmeans::ref_grid(fit_cat_time)
```

```
'emmGrid' object with variables:
  basval = 19.56
  avisit = Week 2, Week 4, Week 8
  trt = 1, 2
```

```
emmeans(fit_cat_time, ~trt)
```

NOTE: Results may be misleading due to involvement in interactions

trt	emmean	SE	df	lower.CL	upper.CL
1	-6.90	0.836	47	-8.58	-5.22
2	-9.09	0.836	47	-10.77	-7.41

Results are averaged over the levels of: avisit  
Confidence level used: 0.95

```
emmeans(fit_cat_time, ~trt*avisit)
```

trt	avisit	emmean	SE	df	lower.CL	upper.CL
1	Week 2	-4.13	0.899	47	-5.93	-2.32
2	Week 2	-5.31	0.899	47	-7.12	-3.51
1	Week 4	-6.70	0.916	47	-8.55	-4.86
2	Week 4	-8.70	0.916	47	-10.54	-6.85
1	Week 8	-9.86	1.033	47	-11.94	-7.79
2	Week 8	-13.26	1.033	47	-15.33	-11.18

Confidence level used: 0.95



### 4.4.1 Observed vs. balanced margins

In the example above we have used the standard option for the weights in the calculation of LS Means. We will delve deeper into the following two options and will try to understand the difference:

- `weights = "equal"`: Each stratum induced by covariate levels is assigned the same weight in the calculation of the LS Means. This is the default option.
- `weights = "proportional"`: Each stratum induced by covariate levels is assigned a weight according to their observed proportion in the calculation of the LS Mean. This option gives each stratum a weight corresponding to its size. Estimates using this option are reflective of the balance of covariates in the data.

**Exercise:** Based on the `fit_cat_time` model, compare the LS Means for the change in the response variable by treatment overall and treatment by visit interaction using the different options for `weight`. Compare the results for the two LS Means options to the observed means and to one another.

Discuss the following points:

- Why is there no difference between LS Means estimates for the overall treatment effect and the treatment by visit interaction? (Hint: Create a frequency table)

Now update the `fit_cat_time` model to `fit_cat_time2`, and include the covariate `gender`. Estimate the same LS Means for the change in the response variable by treatment (overall) and treatment by visit interaction.

- Why is there a difference now between results from the different LS Means options? (Hint: another frequency table can help)
- What effect could missing data have on the estimation, even in the case of `fit_cat_time`? I.e. what would happen if this data was not complete but subject to missingness, with the degree of missing data increasing over time and being disproportionate between treatment arms?

#### Solution:

We first calculate the LS Means, using the different `weights` options and find they are indeed identical.

```
# These will yield the same results:  
emmeans(fit_cat_time, ~trt, weights = "equal")
```

NOTE: Results may be misleading due to involvement in interactions

trt	emmean	SE	df	lower.CL	upper.CL
1	-6.90	0.836	47	-8.58	-5.22
2	-9.09	0.836	47	-10.77	-7.41

Results are averaged over the levels of: avisit  
Confidence level used: 0.95

```
emmeans(fit_cat_time, ~trt, weights = "proportional")
```

NOTE: Results may be misleading due to involvement in interactions

trt	emmean	SE	df	lower.CL	upper.CL
1	-6.90	0.836	47	-8.58	-5.22
2	-9.09	0.836	47	-10.77	-7.41

Results are averaged over the levels of: avisit  
Confidence level used: 0.95

```
emmeans(fit_cat_time, ~trt*avisit, weights = "equal")
```

trt	avisit	emmean	SE	df	lower.CL	upper.CL
1	Week 2	-4.13	0.899	47	-5.93	-2.32
2	Week 2	-5.31	0.899	47	-7.12	-3.51
1	Week 4	-6.70	0.916	47	-8.55	-4.86
2	Week 4	-8.70	0.916	47	-10.54	-6.85
1	Week 8	-9.86	1.033	47	-11.94	-7.79
2	Week 8	-13.26	1.033	47	-15.33	-11.18

Confidence level used: 0.95

```
emmeans(fit_cat_time, ~trt*avisit, weights = "proportional")
```

trt	avisit	emmean	SE	df	lower.CL	upper.CL
1	Week 2	-4.13	0.899	47	-5.93	-2.32
2	Week 2	-5.31	0.899	47	-7.12	-3.51
1	Week 4	-6.70	0.916	47	-8.55	-4.86
2	Week 4	-8.70	0.916	47	-10.54	-6.85
1	Week 8	-9.86	1.033	47	-11.94	-7.79

```
2    Week 8 -13.26 1.033 47    -15.33    -11.18
```

Confidence level used: 0.95

Now we can update the model to include the covariate `gender`. We can specify this a new model using the `mmrm()` function again, or simply use `update()` to add the new covariate to the model. Either way is fine, and a look into the model formula from the fit summary shows the two approaches work interchangeably.

```
fit_cat_time2 <- update(fit_cat_time, . ~ . + gender)
summary(fit_cat_time2)
```

mmrm fit

Formula:

```
change ~ basval + avisit + trt + (us(avisit | subject)) + gender +
      basval:avisit + avisit:trt
```

Data: all12 (used 150 observations from 50 subjects with maximum 3 timepoints)

Covariance: unstructured (6 variance parameters)

Method: Kenward-Roger

Vcov Method: Kenward-Roger

Inference: REML

Model selection criteria:

AIC	BIC	logLik	deviance
817.0	828.5	-402.5	805.0

Coefficients:

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	0.47589	3.23944	46.14000	0.147	0.88385
basval	-0.30674	0.15200	45.44000	-2.018	0.04951 *
avisitWeek 4	-0.90862	2.39786	47.00000	-0.379	0.70645
avisitWeek 8	-10.58630	3.45626	47.00000	-3.063	0.00362 **
trt2	-0.34868	1.30287	46.74000	-0.268	0.79016
genderM	2.32931	1.29556	45.99000	1.798	0.07876 .
basval:avisitWeek 4	-0.08542	0.11646	47.00000	-0.734	0.46689
basval:avisitWeek 8	0.24779	0.16786	47.00000	1.476	0.14657
avisitWeek 4:trt2	-0.80100	0.93186	47.00000	-0.860	0.39439
avisitWeek 8:trt2	-2.20106	1.34318	47.00000	-1.639	0.10795

---

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

Covariance estimate:

```
      Week 2  Week 4  Week 8
Week 2 18.8295 14.3160 12.0002
Week 4 14.3160 21.1623 18.1813
Week 8 12.0002 18.1813 28.8384
```

A look into the reference grid shows us the new factor levels for **gender**. Note that **gender** itself will not be included in the `emmeans()` statement, but the output indicates the averaging over its levels (same for the levels of **avisit**)

```
# Reference grid shows us the new levels
emmeans::ref_grid(fit_cat_time2)
```

'emmGrid' object with variables:

```
basval = 19.56
avisit = Week 2, Week 4, Week 8
trt = 1, 2
gender = F, M
```

```
# These two won't yield the same results
emmeans(fit_cat_time2, ~trt*avisit, weights = "equal")
```

trt	avisit	emmean	SE	df	lower.CL	upper.CL
1	Week 2	-4.36	0.869	45.7	-6.11	-2.61
2	Week 2	-4.71	0.923	46.8	-6.57	-2.85
1	Week 4	-6.94	0.920	46.4	-8.79	-5.09
2	Week 4	-8.09	0.972	48.1	-10.04	-6.13
1	Week 8	-10.10	1.063	45.5	-12.24	-7.96
2	Week 8	-12.65	1.108	48.2	-14.88	-10.42

Results are averaged over the levels of: gender  
Confidence level used: 0.95

```
emmeans(fit_cat_time2, ~trt*avisit, weights = "proportional")
```

trt	avisit	emmean	SE	df	lower.CL	upper.CL
1	Week 2	-4.55	0.890	46.2	-6.34	-2.75
2	Week 2	-4.89	0.890	46.2	-6.69	-3.10

1	Week 4	-7.13	0.941	47.1	-9.02	-5.23
2	Week 4	-8.27	0.941	47.1	-10.17	-6.38
1	Week 8	-10.29	1.081	46.6	-12.46	-8.11
2	Week 8	-12.83	1.081	46.6	-15.01	-10.66

Results are averaged over the levels of: gender  
Confidence level used: 0.95

The following frequency table shows the imbalance in the distribution of the `gender` variable. We can see that Treatment 1 has more men than women, whereas Treatment 2 has more women than men.

```
table(all2$trt, all2$gender)
```

	F	M
1	30	45
2	57	18

The data is no longer balanced across the covariates in the model. The `weights = "equal"` option is agnostic to this imbalance and assigns all levels equal weights, whereas the `weights = "proportional"` assigns a weight reflecting the proportional size of the stratum over which the average is taken.

## 4.4.2 Contrasts

Most of the times, the quantity we are truly interested in when reading out a study, is the contrast between treatment arms. This contrast can be built either based on LS Means at some landmark time point, or as a longitudinal (linear) combination of LS Means from multiple time points.

We can use the `pairs()` or the `contrast()` functions, where the latter provides more flexibility for the calculation of linear combinations from multiple time points.

```
lsmns <- emmeans::emmeans(fit_cat_time, ~trt*avisit, weights = "proportional")
pairs(lsmns, reverse = TRUE, adjust = NULL)
```

contrast	estimate	SE	df	t.ratio	p.value
trt2 Week 2 - trt1 Week 2	-1.19	1.273	47.0	-0.935	0.3546
trt1 Week 4 - trt1 Week 2	-2.58	0.659	47.0	-3.917	0.0003

```

trt1 Week 4 - trt2 Week 2    -1.39 1.284 61.3  -1.082  0.2835
trt2 Week 4 - trt1 Week 2    -4.57 1.284 61.3  -3.559  0.0007
trt2 Week 4 - trt2 Week 2    -3.38 0.659 47.0  -5.133  <.0001
trt2 Week 4 - trt1 Week 4    -1.99 1.296 47.0  -1.536  0.1313
trt1 Week 8 - trt1 Week 2    -5.74 0.950 47.0  -6.043  <.0001
trt1 Week 8 - trt2 Week 2    -4.55 1.370 73.3  -3.321  0.0014
trt1 Week 8 - trt1 Week 4    -3.16 0.716 47.0  -4.416  0.0001
trt1 Week 8 - trt2 Week 4    -1.17 1.381 61.1  -0.846  0.4007
trt2 Week 8 - trt1 Week 2    -9.13 1.370 73.3  -6.664  <.0001
trt2 Week 8 - trt2 Week 2    -7.94 0.950 47.0  -8.361  <.0001
trt2 Week 8 - trt1 Week 4    -6.55 1.381 61.1  -4.742  <.0001
trt2 Week 8 - trt2 Week 4    -4.56 0.716 47.0  -6.373  <.0001
trt2 Week 8 - trt1 Week 8    -3.39 1.462 47.0  -2.319  0.0248

```

```

### This is the same as the following
prs <- contrast(lsmns, method = "revpairwise", adjust = NULL)

```

Note that both `pairs()` and `contrast()` provide multiple options for fine-tuning. We chose `adjust = NULL` in order to not perform any multiplicity adjustment (default method would have been the Tukey method). We also chose `reverse = TRUE` to reverse the order of comparisons performed by `pairs()`, as the default would have given us the contrast for Treatment 1 - Treatment 2. Consequently, we applied `method = "revpairwise"` in the `contrast()` function.

We can obtain the coefficients in the calculation of the contrasts via `coef()`:

```
coef(prs)
```

```

      trt avisit c.1 c.2 c.3 c.4 c.5 c.6 c.7 c.8 c.9 c.10 c.11 c.12 c.13
trt1 Week 2   1 Week 2  -1  -1   0  -1   0   0  -1   0   0   0   -1   0   0
trt2 Week 2   2 Week 2   1   0  -1   0  -1   0   0  -1   0   0   0  -1   0
trt1 Week 4   1 Week 4   0   1   1   0   0  -1   0   0  -1   0   0   0  -1
trt2 Week 4   2 Week 4   0   0   0   1   1   1   0   0   0  -1   0   0   0
trt1 Week 8   1 Week 8   0   0   0   0   0   0   1   1   1   1   0   0   0
trt2 Week 8   2 Week 8   0   0   0   0   0   0   0   0   0   0   1   1   1
      c.14 c.15
trt1 Week 2   0   0
trt2 Week 2   0   0
trt1 Week 4   0   0
trt2 Week 4  -1   0
trt1 Week 8   0  -1
trt2 Week 8   1   1

```

The output above is probably more than we wanted. We are only interested in contrasts between Treatments 1 and 2 at the same time points. Here `contrast()` provides more flexibility. Instead of parsing a string with the name of a method to the `method` argument, we provide a named list of coefficients. These coefficients are identical with the ones we can see in the coefficient matrix above. We can use it as a guide.

```
contrast(
  lsmns,
  method = list(
    "Difference Trt 2 - Trt 1 at Week 4" = c(0, 0, -1, 1, 0, 0),
    "Difference Trt 2 - Trt 1 at Week 8" = c(0, 0, 0, 0, -1, 1)
  ),
  adjust = NULL)
```

contrast	estimate	SE	df	t.ratio	p.value
Difference Trt 2 - Trt 1 at Week 4	-1.99	1.30	47	-1.536	0.1313
Difference Trt 2 - Trt 1 at Week 8	-3.39	1.46	47	-2.319	0.0248

This way of computing LS Means from our MMRM allows us to calculate all kinds of linear combinations of LS Means. Assume we were interested in the longitudinal mean of changes from baseline averaged over Weeks 2, 4 and 8. This would look like this:

```
contrast(
  lsmns,
  method = list(
    "Difference Trt 2 - Trt 1 Averaged over Weeks 2, 4 and 8" = c(-1, 1, -1, 1, -1, 1)/3
  ),
  adjust = NULL)
```

contrast	estimate	SE	df
Difference Trt 2 - Trt 1 Averaged over Weeks 2, 4 and 8	-2.19	1.18	47
t.ratio	-1.850		
p.value	0.0705		

## 4.5 Fit diagnostics

The following section closely follows the content in Chapter 10 in (Fitzmaurice 2011).

Our analysis should be concluded with a look into the fit diagnostics, more specifically, the residuals. Residuals are defined by the difference between the true responses and the fitted values from the model:

$$r := y - X\hat{\beta},$$

where  $\hat{\beta}$  are the estimated coefficients from our model. Residuals provide an estimate of the true vector of random errors

$$\varepsilon = y - X\beta.$$

As per our modeling assumptions,  $\varepsilon$  should follow a normal distribution with mean zero. The mean of the residuals is zero and therefore identical with the mean of the error term. For the covariance of the residuals however, the variance-covariance matrix of  $\varepsilon$  only serves us as an approximation (as suggested by (Fitzmaurice 2011) for all ‘practical applications’):

$$Cov(r) \approx Cov(\varepsilon) = R.$$

This assumption has several implications on the residual diagnostics:

- The variance is not necessarily constant. Plotting the fitted values versus the residuals might therefore lead to a non-constant range. An examination of the residual variance or autocorrelation among residuals is therefore not very meaningful.
- Residuals from analyses of longitudinal data can exhibit correlation with the covariates. Scatterplots of residuals versus selected covariates can therefore reveal systematic trends (which normally should not be the case).

A transformation of residuals to achieve constant variance and zero correlation is therefore often useful. This transformation uses the so-called *Cholesky decomposition* of the variance-covariance matrix  $R$ . Let  $L$  be a lower triangular matrix, such that

$$R = L L',$$

then the transformed residuals are given by

$$r^* = L^{-1}(y - X\beta).$$

In the `mmrm` package, transformed residuals can be derived using the `type = "normalized"` option.

**Exercise:** Which visualisations can you think of that make sense to assess the goodness of fit here? Create a new `tibble` (or `data.frame`) containing the variables of importance and try plotting them in a meaningful way. Discuss the results within your group.

**Solution:**

To avoid repetition, let us first save the important variables to perform fit diagnostics in a `tibble`.

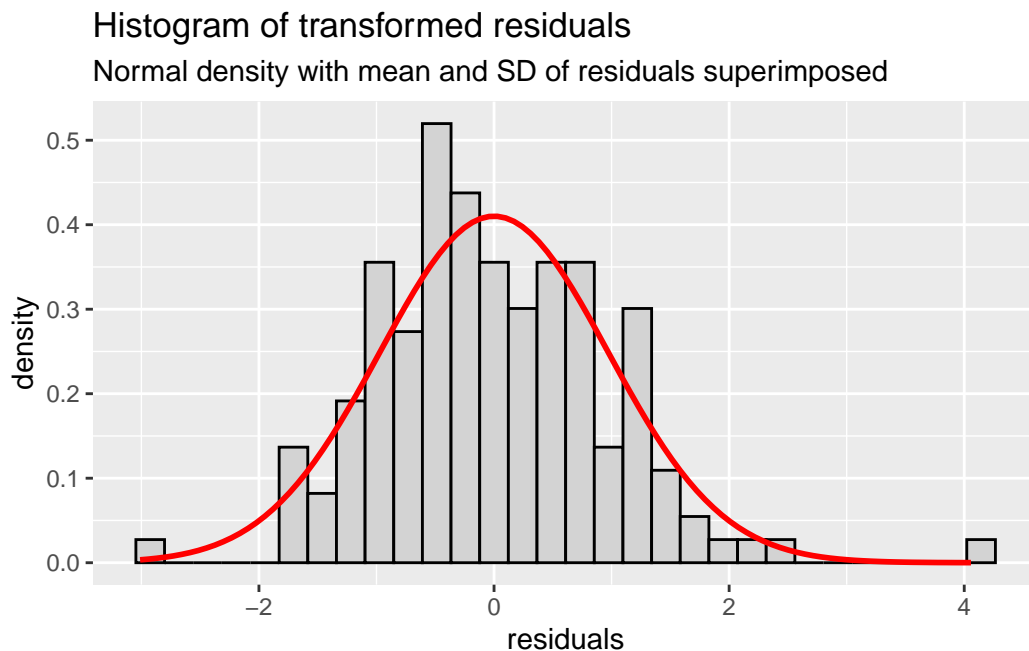


```
df_residuals <- dplyr::tibble(
  residuals = residuals(fit_cat_time, type = "normalized"),
  predictions = fitted(fit_cat_time),
  all2
)
```

We can firstly look into a histogram of transformed residuals. The shape should resemble the density function of normal distribution with mean zero and positive variance. Superimposing the density function with mean and SD derived from the model residuals, let's us see that this is indeed the case. We can also detect a slight skewness to the right.

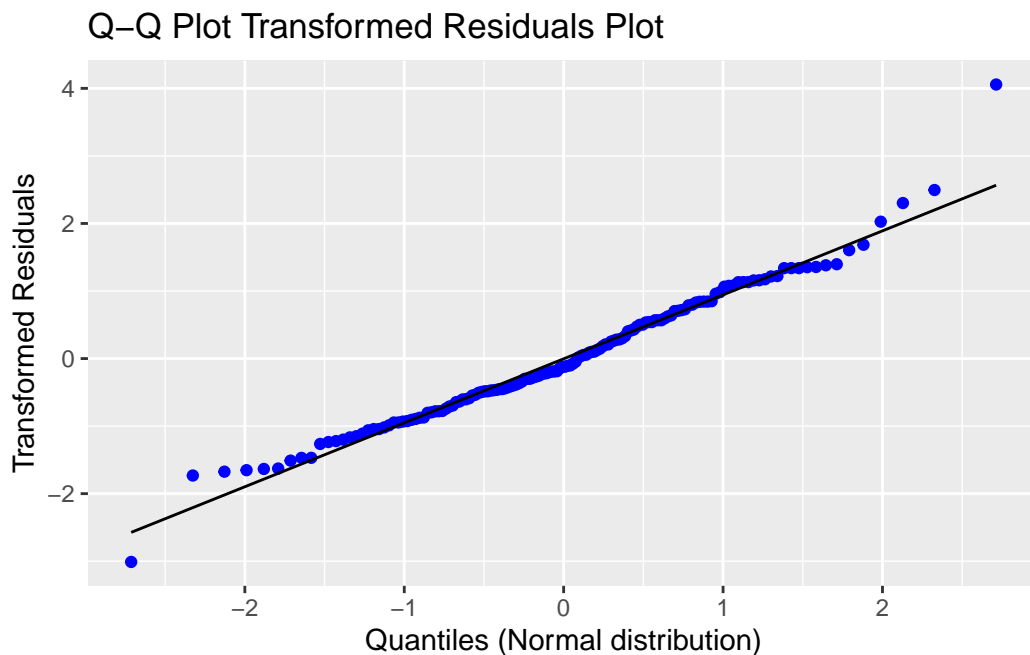
```
library(ggplot2)

df_residuals %>%
  ggplot(aes(x = residuals)) +
  geom_histogram(aes(y = after_stat(density)), fill='lightgray', col='black') +
  stat_function(fun = dnorm, args = list(mean=mean(df_residuals$residuals), sd=sd(df_residuals$residuals)),
  ggtitle(
    label = "Histogram of transformed residuals",
    subtitle = "Normal density with mean and SD of residuals superimposed"
  )
)
```



Alternatively, we can create a Q-Q-Plot

```
df_residuals %>%  
  ggplot(aes(sample = residuals)) +  
  stat_qq(color = "blue") +  
  stat_qq_line() +  
  labs(  
    x = "Quantiles (Normal distribution)",  
    y = "Transformed Residuals"  
  ) +  
  ggtitle(  
    label = "Q-Q Plot Transformed Residuals Plot"  
  )  
)
```



How to interpret the Q-Q plot:

We can use the following fourfold table to assess the shape characteristics derivable from this plot, depending on where the data on which end of the plot is bend compared to the linear trend line:

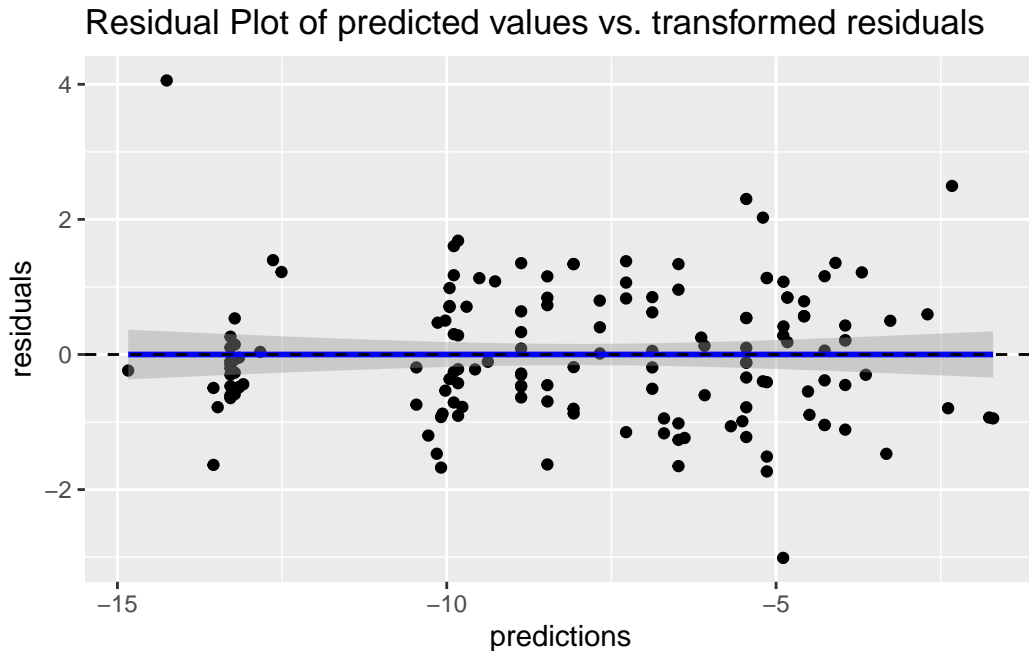
Upper right corner	
Above	Below

Lower left corner	Above	Skewed to the Right	Light-tailed
Lower left Corner	Below	Heavy-tailed	Skewed to the Left

---

We can see that our data is skewed to the right, as the data in the upper right corner and data in the lower left corner of the plot bend above the linear trend line. This is also a trend we can observe from the histogram.

```
df_residuals %>%
  ggplot(aes(x = predictions, y = residuals)) +
  geom_point() +
  geom_smooth(method = lm, color = "blue") +
  geom_hline(yintercept = 0, show.legend = FALSE, linetype = 2) +
  ggtitle(
    label = "Residual Plot of predicted values vs. transformed residuals"
  )
```

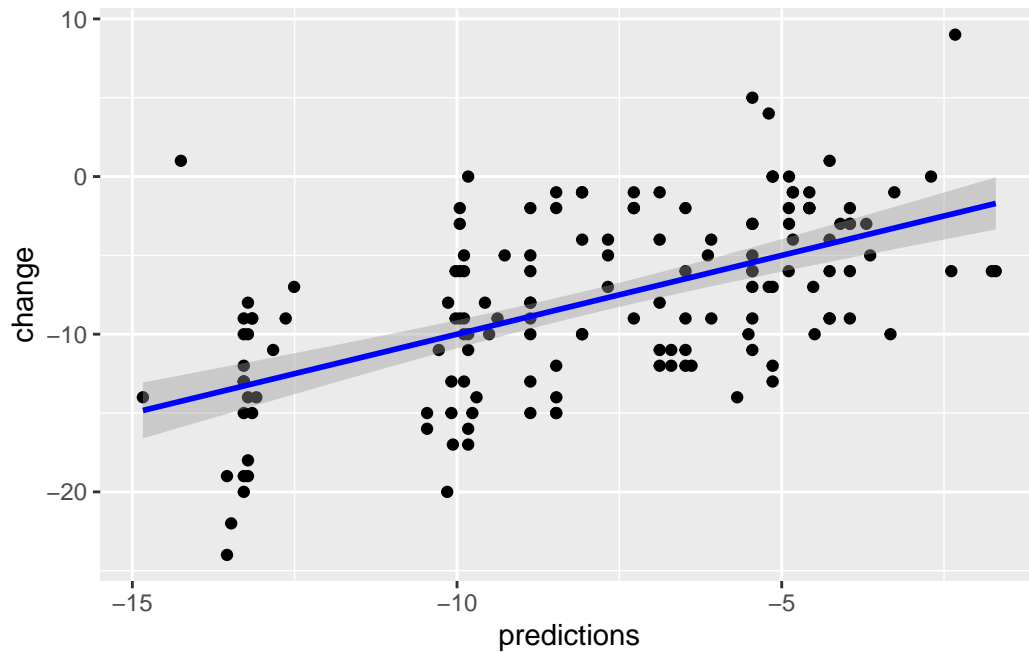


What do we see?

- The points in the plot look well dispersed and symmetric around zero. The fitted line shows no departure from zero.
- There is no systematic trend, but a rather random scatter.

- There are a couple of outliers.

```
df_residuals %>%  
  ggplot(aes(x = predictions, y = change)) +  
  geom_point() +  
  geom_smooth(method = lm, color = "blue")
```



#### 4.5.1 Addendum on RS&I Models

Different dosing/ assessment frequency between treatment arms in parallel design -> oncology (chemo with fixed cycles vs immune-therapy)

## 5 Missing Data

So far, we conducted all our analyses on the basis of complete data. This is a blissful, yet highly unusual setting.

Our dataset contains a second variable `chgdrops`, which is subject to missingness. Let's rerun our initial MMRM with `chgdrops` as dependent variable, baseline value, visit, baseline by visit interaction and treatment by visit interaction as fixed effects and an unstructured covariance matrix for visits within each subject.

This formulation is very similar to the one at the beginning of the former chapter. How do the results differ in terms of LS Means of change from baseline by treatment arm over time?

```
fit_cat_time <- mmrm::mmrm(  
  formula = chgdrops ~ basval*avisit + trt*avisit + us(avisit | subject),  
  data = all2,  
  control = mmrm_control(method = "Kenward-Roger")  
)  
  
summary(fit_cat_time)
```

mmrm fit

```
Formula:      chgdrops ~ basval * avisit + trt * avisit + us(avisit | subject)  
Data:        all2 (used 129 observations from 50 subjects with maximum 3  
timepoints)  
Covariance:  unstructured (6 variance parameters)  
Method:      Kenward-Roger  
Vcov Method: Kenward-Roger  
Inference:   REML
```

Model selection criteria:

AIC	BIC	logLik	deviance
709.2	720.7	-348.6	697.2

Coefficients:

Estimate	Std. Error	df	t value	Pr(> t )
----------	------------	----	---------	----------

(Intercept)	1.98452	3.27498	46.99000	0.606	0.54745
basval	-0.31235	0.15906	46.99000	-1.964	0.05549 .
avisitWeek 4	-0.90712	2.44043	39.90000	-0.372	0.71208
avisitWeek 8	-11.82291	3.34959	36.17000	-3.530	0.00115 **
trt2	-1.18993	1.27272	46.99000	-0.935	0.35460
basval:avisitWeek 4	-0.07256	0.11799	39.85000	-0.615	0.54206
basval:avisitWeek 8	0.31809	0.16104	35.86000	1.975	0.05598 .
avisitWeek 4:trt2	-0.90513	1.00404	40.53000	-0.901	0.37266
avisitWeek 8:trt2	-1.70761	1.42670	38.06000	-1.197	0.23875

---

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

Covariance estimate:

	Week 2	Week 4	Week 8
Week 2	20.6136	15.5273	13.4180
Week 4	15.5273	21.6600	17.7420
Week 8	13.4180	17.7420	27.3112

```
model_lsmeans <- emmeans::emmeans(fit_cat_time, ~trt*avisit, weights = "proportional")
model_lsmeans
```

trt	avisit	emmean	SE	df	lower.CL	upper.CL
1	Week 2	-4.10	0.900	47.0	-5.91	-2.29
2	Week 2	-5.29	0.899	47.0	-7.10	-3.48
1	Week 4	-6.42	0.974	46.5	-8.38	-4.46
2	Week 4	-8.52	0.951	44.8	-10.43	-6.60
1	Week 8	-9.73	1.142	40.4	-12.03	-7.42
2	Week 8	-12.62	1.114	40.1	-14.88	-10.37

Confidence level used: 0.95

```
emmeans::emmeans(fit_cat_time, ~trt*avisit, weights = "proportional") %>%
  contrast(
    list(
      "Difference in LS Means at Week 8" = c(0, 0, 0, 0, -1, 1),
      "Difference in longitudinal LS Means to Week 8" = c(-1, 1, -1, 1, -1, 1)/3
    )
  )
```

contrast	estimate	SE	df	t.ratio
----------	----------	----	----	---------

Difference in LS Means at Week 8	-2.90	1.60	40.3	-1.814
Difference in longitudinal LS Means to Week 8	-2.06	1.23	46.8	-1.671
p.value				
	0.0772			
	0.1014			

To understand the nature of the differences between the model using **change** as a response variable and the one with **chgdrops**, we need to look closer into the extent of missing data and understand its nature.

## 5.1 Missing Data Mechanisms

To understand the nature of missing data in our clinical trial, we consider the following taxonomy, introduced by (Roderick JA Little 2019). We differentiate between the following three types of missing data:

- **Missing Completely at Random (MCAR):** Conditional on all covariates in our analysis, the probability of missingness does not depend on either observed or unobserved values of the response variable.
- **Missing at Random (MAR):** Conditional on all covariates and observed response values in our analysis, the probability of missingness does not depend on the unobserved values of the response variable.
- **Missing not at Random (MNAR):** Conditional on all covariates and observed response values in our analysis, the probability of missingness does depend on the unobserved values of the response variable.

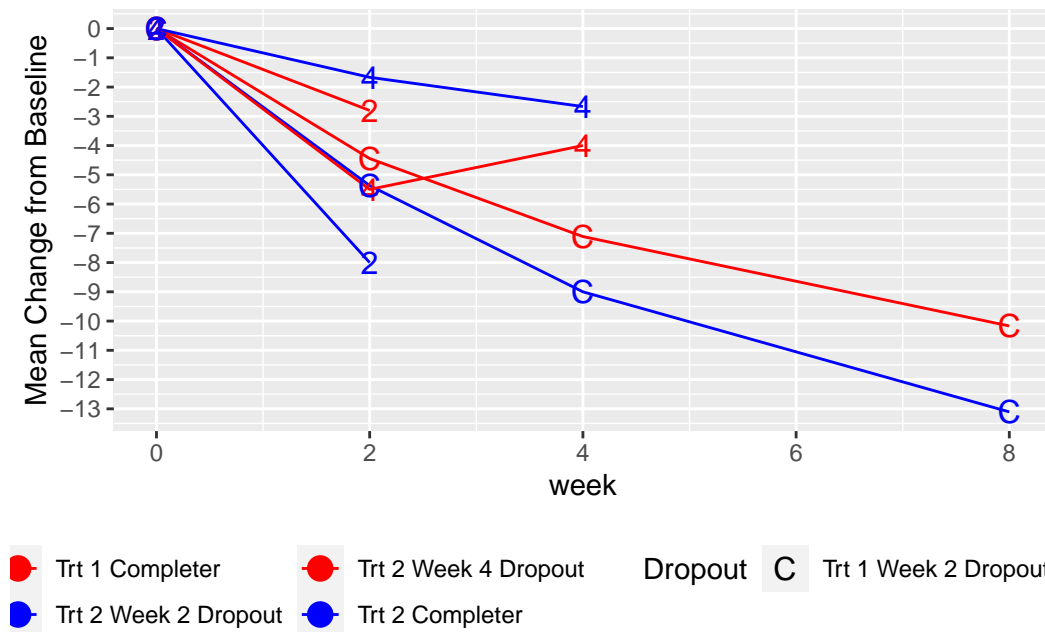
(Craig Mallinckrodt 2016) give the following interpretation around the three types of missingness:

*“With MCAR, the outcome variable is not related to the probability of dropout (after taking into account covariates). In MAR, the observed values of the outcome variable are related to the probability of dropout, but the unobserved outcomes are not (after taking into account covariates and observed outcomes). In MNAR the unobserved outcomes are related to the probability of dropout even after the observed outcomes and covariates have been taken into account.”*

The following two sections outline handling strategies for missing data. However, the best approach to handle missing data is to minimise its extent. While the occurrence of missing data can rarely be avoided at all (think about the collection of questionnaire data in oncology studies and the missing data after subjects die), it is important to pursue an “as complete as can be” data collection.

Baseline and screening data are of utmost importance in a pursuit of data completeness. If a screening value is missing, but was meant to be used as a covariate, this subjects' whole data will be dropped from the analysis even if all responses were observed. If the baseline response variable was missing we are unable to compute a change from baseline, which also leads to the loss of this subjects' data in the model (although LDA models are still able to provide an estimate) even if all post-baseline values were observed.

## 5.2 Missing data handling I (descriptive stats + visualisations)



## 5.3 Missing data handling II (analytic approaches)



## 6 Sensitivity Analyses

Sensitivity analyses with respect to missing data - only?

- MMRM is an appropriate choice for the primary analysis in many longitudinal clinical trials under the MAR assumption.
- MCAR and MAR missingness can be ignored in likelihood-based analyses. MAR: future outcomes for subjects who discontinued should be similar to the future outcomes of subjects who continued if they had the same values of past (observed) outcomes, covariates,...
- Flexibility in modeling treatment effects over time and the within-patient error correlation structure makes MMRM a widely useful analysis.
- Consider sensitivity analyses to check model assumptions e.g. MNAR methods
- Comparing results from sensitivity analyses: how much inference rely on the assumptions
- Here, inference with regard to the treatment effect. Thus, investigate how treatment effects vary depending on assumptions e.g. about missing data.
- Uncertainty from incompleteness cannot be objectively evaluated from observed data so there is a need for missing data sensitivity analyses.

Mit Alexandra abstimmen: was genau, machst du MI in deinem Teil, MMRM vs. MI+ANCOVA?, was ist eure primary analysis + typische sensitivity analyses MI: eingehen auf welche Variablen im Modell e.g. at least those from the primary model

### 6.1 Simple approaches

In general, not recommended for use. Methods are of historic interest and provide a useful starting point - LOCF: used in the past, justified as it was thought that it provides conservative estimates - complete case (observed case/completers analysis): creates selection bias, may cause overestimation of within group effects particularly at the last scheduled visit

### 6.2 Missing covariates (baseline data)

- missing baseline value of the outcome: MI or use of mean imputation (Paper: ),
- MMRM not efficient or potential biased estimates as subjects with missing covariates are excluded from the analysis

### 6.3 Baseline complete, missingness in post-baseline values

- for MMRM: at least one post-baseline value needed
- Alternative: LDA where baseline is part of the response vector
- when implemented in similar manners: MI and MMRM have similar assumptions and yield similar results. Thus, MI implemented similarly to MMRM is not a sensitivity analysis

### 6.4 Multiple imputation

- MI very useful for sensitivity analyses

### 6.5 Handling nonignorable missingness (MNAR)

- Assumption of MAR is often reasonable, but possibility of data missing not at random (MNAR) is difficult to rule out.
- Thus, analysis under MNAR needed
- Analysis under MNAR: these methods are heavily assumption driven and the assumptions are not testable as we do not have the missing data
- Consider a sensitivity analysis framework allowing assessment of robustness of results to the various assumptions
- MNAR methods: Pattern-mixture, delta-adjustment method (controlled imputation; another method is reference-based imputation)

## **7 Inferences from binary longitudinal data**

## References

- Craig Mallinckrodt, Ilya Lipkovich. 2016. *Analyzing Longitudinal Clinical Trial Data: A Practical Guide*. Vol. 1. USA: Chapman; Hall/CRC. <https://doi.org/10.1201/9781315186634>.
- Fitzmaurice, Laird, G. M. 2011. *Applied Longitudinal Analysis*. Vol. 2. USA: New York, Wiley. <https://doi.org/10.1002/9781119513469>.
- Mallinckrodt, Craig. 2016. *Analyzing Longitudinal Clinical Trial Data*. Chapman; Hall/CRC. <https://doi.org/10.1201/9781315186634>.
- Roderick JA Little, Donald B. Rubin. 2019. *Statistical Analysis with Missing Data*. Vol. 3. USA: New York, Wiley. <https://doi.org/10.1002/9781119482260>.