Lecture 8: repeated measurements

Lecture notes and R code

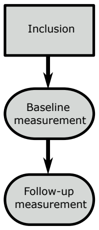
Brice Ozenne

1 Study design

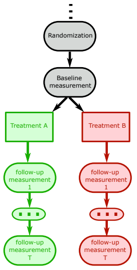
Repeated measurements arise when one (or several) variable(s) are measured at different occasions on the same experimental unit.
Typically a clinical score may be measured repeatedly over time on the **same** subject.

Classical designs involving repeated measurements:

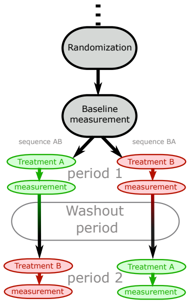
- **test re-test study**: one group of subjects undergo a baseline and a follow-up measurement.
  → expect the same value at baseline and follow-up
  → used to assess the stability of a measurement device.
  → each individual is its own control.



- **longitudinal study**: two groups of subjects undergo a baseline measurement, get treatment, and then undergo several follow-up measurements.
  → assess the effect of a drug/exposure over time.
  → another group as control.



- **cross-over study**: one group of subjects undergo several follow-up measurements, each under a different treatment condition.
  → period effect can be controlled for using multiple sequences of treatment.
  → assess the stability/reliability of an instrument.
  → each individual is its own control.

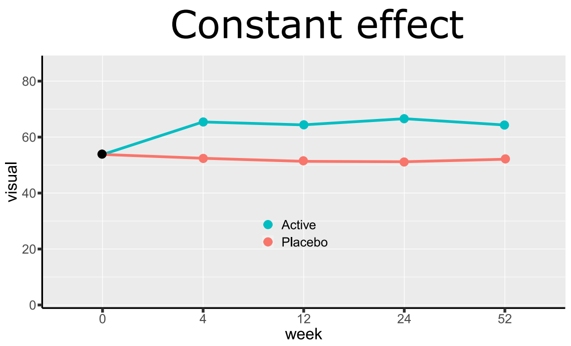


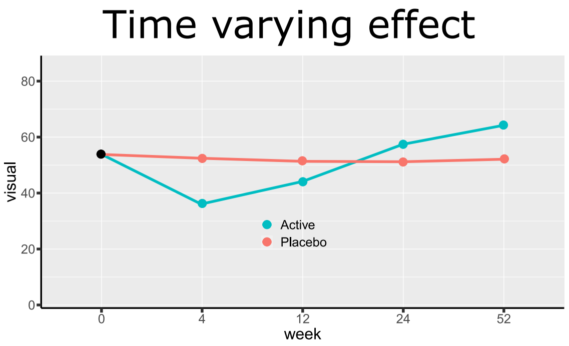
⚠ In those designs, follow-up times are decided by design and the same for all patients (independence between data collection and outcome value).

2 Benefit of repeated measurements

Getting more **inside**:

- e.g. understanding the time dynamic of the treatment effect.
  Can be constant, delayed, early detrimental and then beneficial, . . .


Constant effect


Time varying effect

Getting more **precise** estimates:

- remove biological variability by modeling within patient changes.

Example: consider an outcome $Y_i(t)$ (e.g. visual acuity) that is a function of time t and individual i:

$$Y_i(t) = \alpha(t) + \beta X_i + \gamma Z_i(t) + \varepsilon_i(t)$$

The outcome may depend:

- on time via $\alpha(t)$ (e.g. worse vision as a disease develops)
- on individual characteristics X_i (e.g. genes) that are assumed constant over time and have an effect β also constant over time.
- on instruments Z_i that may change over time (e.g. distance between eyes and eye chart) with an effect γ independent of time.
- other unknown reasons $\varepsilon_i(t)$

The change in outcome between baseline and time 4 is:

$$Y_i(4) - Y_i(0) = \alpha(4) - \alpha(0) + \gamma(Z_i(4) - Z_i(0)) + \varepsilon_i(4) - \varepsilon_i(0)$$

Working on the change:

- only removes covariate effects that are time independent
- when $\mathbb{V}ar\left[Y_i(4) - Y_i(0)\right] < \mathbb{V}ar\left[Y_i(4)\right]$, is a less variable outcome than working on the value at follow-up. Assuming constant variance, this means that[1]: $\mathbb{C}or(Y_i(4), Y_i(0)) > 0.5$.

[1] $\mathbb{V}ar\left[Y_i(4) - Y_i(0)\right] = \mathbb{V}ar\left[Y_i(4)\right] + \mathbb{V}ar\left[Y_i(0)\right] - 2 * \sqrt{\mathbb{V}ar\left[Y_i(4)\right]\mathbb{V}ar\left[Y_i(0)\right]}\mathbb{C}or(Y_i(4), Y_i(0))$

3 Challenges with repeated measurements

A key difference between repeated measurements data and "usual" data is that
(A) **observations are no more independent**:

- observations belonging to the same individual are usually (strongly) correlated.
- most of statistical results require independence.

Intuitively: ignoring the dependence would be like running a study with new patients at each timepoint instead of the same patients followed over time.

→ will lead to incorrect p-values as an incorrect sample size is used.

→ will lead to sub-optimal estimates, as information can generally be shared between timepoints which reduces uncertainty[2].

(B) **Missing data** is another difficulty that arises with this type of design. As the follow-up time increases, patient are more likely to quit or pause their participation in the study for reasons:

- unrelated to the outcome (**independent censoring**),
  e.g. fatigue/inconvenience, forgotten appointments, loss of motivation, ...
- related to the outcome (**informative censoring**),
  e.g. full recovery, worsening of the disease, treatment side effect, ...

Technical issues can also generate missing values for reasons :

- unrelated to the outcome (**independent censoring**),
  e.g. breakdown of a measurement device
- related to the outcome (**informative censoring**),
  e.g. detection limits of a device

Patient may also experience an event ending their participation (**competing events**):

- death, disease affecting the outcome (e.g. becoming blind in an incident)

Handling competing events and informative censoring is beyond the scope of this lecture, as they require specific methods:

⚠ complete case analysis will generally output incorrect results

(C) **Multiple testing** issues also frequently arise with repeated measurements:

→ multiple parameters describing the treatment effect
(e.g. at week 4, 12, ...)

→ multiple statistical approaches, depending on the type of model used
how it is specified
(e.g. multiple t-tests, mixed model with linear treatment effect, ...)

[2] e.g. when assuming a constant gender effect, observations from all timepoints are used to estimate this effect leading to a less variable estimate compared to using only information from a single timepoint.

4 Descriptive statistics with repeated measurements

Start by identifying a **level** at which we have **independent replicates**:

- e.g. observations are grouped within patients.
  i.e. independent replicates of the vector of outcomes $\mathbf{Y} = (Y(0), Y(4)), Y(12), Y(24), Y(52))$.

Recommended descriptive statistics include:

- overview of the data with a spaghetti plot
- summary statistics at each timepoint like median, quantiles at each time
- correlation across repetitions
- missing data patterns
  percentage of missing data (per group) across timepoints.

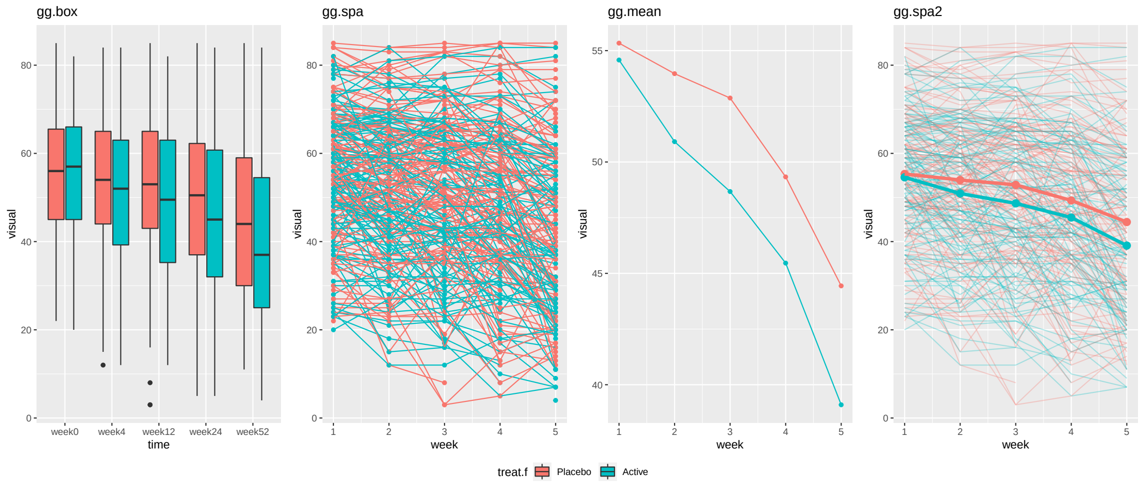
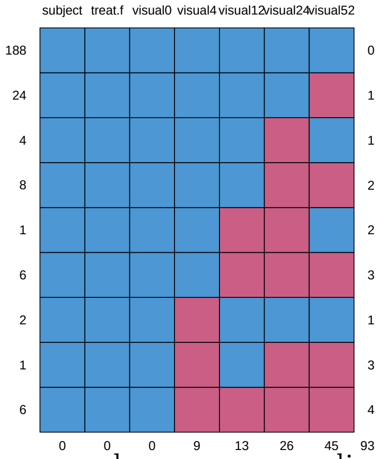


Figure 1: Boxplot (left panel), spaghetti plot (middle left panel), mean plot (middle right panel), and combination of the two (middle panel).



	subject	treat.f	visual0	visual4	visual12	visual24	visual52	
188	🟦	🟦	🟦	🟦	🟦	🟦	🟦	0
24	🟦	🟦	🟦	🟦	🟦	🟦	🟥	1
4	🟦	🟦	🟦	🟦	🟦	🟥	🟦	1
8	🟦	🟦	🟦	🟦	🟦	🟥	🟥	2
1	🟦	🟦	🟦	🟦	🟥	🟥	🟦	2
6	🟦	🟦	🟦	🟦	🟥	🟥	🟥	3
2	🟦	🟦	🟦	🟥	🟦	🟦	🟦	1
1	🟦	🟦	🟦	🟥	🟦	🟥	🟥	3
6	🟦	🟦	🟦	🟥	🟥	🟥	🟥	4
	0	0	0	9	13	26	45	93

Figure 2: Missing data patterns, each row corresponding to a pattern. Blue indicates observed data and red missing data. The number on the left (resp. right) is number of observations (resp. missing data) in a pattern.

5 Statistical analysis

There are several valid[3] ways to analyze repeated measurements. Two are presented:

- a **univariate approach**: simple to implement. Especially relevant when one is interested in a single timepoint and there is no missing data (or there is only independent censoring and the sample size is large).

- a **multivariate approach**: more complex to implement, interpret, and report but can better handle censoring or unbalanced cross-over designs. It can also use parsimonious models when analyzing multiple timepoints, e.g. to obtain a concise representation of the time dynamic of the treatment effect.

Example: to simplify, we will describe these methods when applied to a longitudinal study, with two-arms: active ($G = 1$) vs. placebo ($G = 0$), randomized, and no covariates. We will denote by:

- $\alpha_g(t)$ the expected outcome value at time t in the arm g.
- $\sigma_g^2(t)$ the variance of outcome at time t in the arm g.
- $\rho_g(t_1, t_2)$ the correlation between outcome values from the same subject at time t_1 and t_2 in the arm g.

5.1 Univariate approach

1. Before collecting the data, we know that the interest lies in the treatment effect at a specific timepoint:
   - e.g. timepoint ($t = 52$) to assess the long-term treatment effect

2. We will then typically be interested in compare the change in outcome from baseline between two treatment groups:
   - e.g. $\Delta Y(52) = Y(52) - Y(0)$ in the Active vs. Placebo group.
   ⚠ doing so will implicitely perform a complete case analysis, i.e. exclude individuals with a missing value at one of the two timepoints

3. We make a graphical display of the data

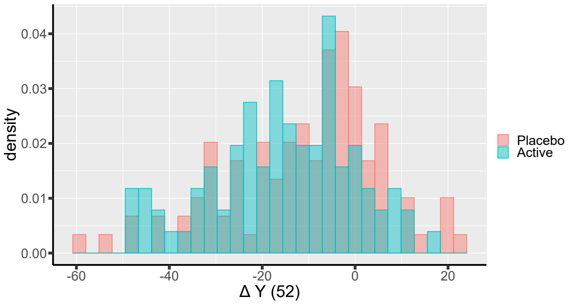


Figure 3

[3] in the sense unbiased under reasonable assumptions

4. If the mean provides a reasonable summary of the treatment effect, we can perform a two-sample t-test on the change in outcome.

- in our example the corresponding statistical model is:

$$\Delta Y(52)|G = 0 \sim \mathcal{N}(\mu_0, \tau_0)$$
$$\Delta Y(52)|G = 1 \sim \mathcal{N}(\mu_1, \tau_1)$$
$$\text{where for } g \in \{0,1\} \quad \mu_g = \alpha_g(52) - \alpha_g(0)$$
$$\text{and} \quad \tau_g = \sigma_g^2(52) + \sigma_g^2(0) - 2\sigma_g(52)\sigma_g(0)\rho_g(0,52)$$

(normality ease the description of the model but is not a requirement)
We are then testing the null hypothesis of no treatment effect on the mean:

$$\mu_1 - \mu_0 = 0$$

⚠ if this approach is performed at multiple timepoints to find the timepoint at which the treatment effect is the strongest, the procedure need to be adjusted for multiple comparisons.
Note that because of missing values, the patients kept in the complete case analysis may not be the same at all timepoints.
⚠ Using a standard linear model, e.g. `lm` in R , instead of a t-test leads to assume no treatment effect on the variance $\sigma_1^2 = \sigma_0^2$.

5.2 Multivariate approach

This approach exploits properties of the multivariate normal distribution.
1. Before collecting the data, we either know that the interest lies in the treatment effect at **a specific timepoint** or impose **a specific shape** to the treatment effect.

- e.g. timepoint $(t = 52)$
- e.g. the effect of the treatment is proportional to the number of weeks from baseline.

2. Drop-out prevents us to observe the outcome at follow-up times for all indiviudals. For individual i belonging to group g, we would like to **guess the missing follow-up values** (e.g. $Y_i(52)$) based on:

a) the observed values for this individual at other/earlier timepoints (e.g. $Y_i(0)$)

b) the average outcome in the population at the missing timepoints (e.g. $\alpha_g(52)$)

c) the correlation between the values at missing and observed timepoints (e.g. $\rho_g(0,52)$) in the population will determine how much weight is given to a).

Specific cases:

- No correlation: our best guess is b), i.e. a) is not used.
- Perfect correlation: our best guess is b) corrected by how much a) deviates from the population mean.

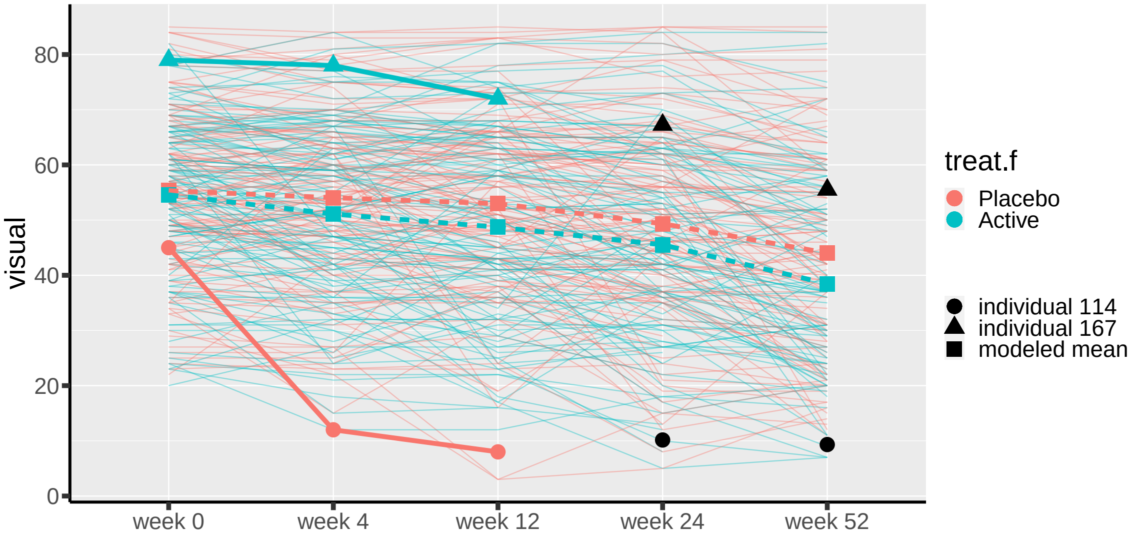


Figure 4: Illustration of what we would like to perform in step 2. The black dots and triangles are our "guesses". Compared to the other figures, 5 timepoints are used instead of 3 to better visualize the correlation over time.

- restricting to timepoints 0 and 52 weeks, we would guess for individual i in arm $g \in \{0,1\}$:

$$\hat{Y}_i(52) = \alpha_g(52) + \rho_g(0,52)\frac{\sigma_g(52)}{\sigma_g(0)}\left(Y_i(0) - \alpha_g(0)\right) \tag{1}$$

Looking more closely at Equation 1, one can see that it is equivalent to a linear regression with as:
outcome: $Y_i(52)$, i.e. timepoint(s) with missing values
covariate: $Y_i(0)$, i.e. timepoint(s) with observed values
intercept: $\alpha_g(52) - \rho_g(0,52)\frac{\sigma_g(52)}{\sigma_g(0)}\alpha_g(0)$
slope: $\rho_g(0,52)\frac{\sigma_g(52)}{\sigma_g(0)}$.
We can visualize the corresponding data with a simple scatterplot:

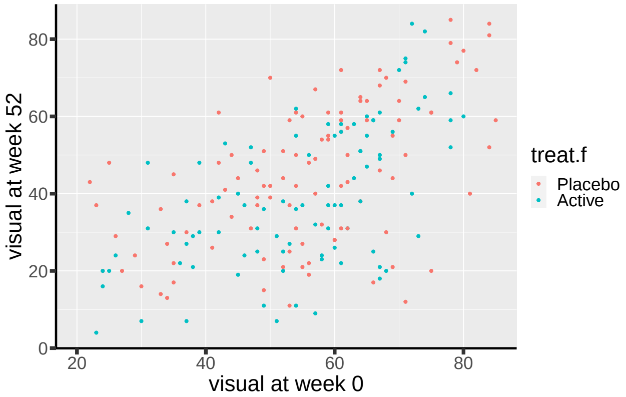


Figure 5

3. The **mixed model** can be viewed as a convenient way to guess the missing values (across several missing data patterns) and account for the uncertainty about the guesses when comparing the groups. The mixed model requires more than modeling than the t-test, as the mean, variance, and correlation over time are used to handle missing values.

- with only two timepoints, the statistical model for the control group is:

$$\begin{bmatrix} Y(0)|G=0 \\ Y(52)|G=0 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \alpha_0(0) \\ \alpha_0(52) \end{bmatrix}, \begin{bmatrix} \sigma_0^2(0) & \sigma_0(0)\sigma_0(52)\rho_0(0,52) \\ \sigma_0(0)\sigma_0(52)\rho_0(0,52) & \sigma_0^2(52) \end{bmatrix} \right)$$

while for the active group it is:

$$\begin{bmatrix} Y(0)|G=1 \\ Y(52)|G=1 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \alpha_1(0) \\ \alpha_1(52) \end{bmatrix}, \begin{bmatrix} \sigma_1^2(0) & \sigma_1(0)\sigma_1(52)\rho_1(0,52) \\ \sigma_1(0)\sigma_1(52)\rho_1(0,52) & \sigma_1^2(52) \end{bmatrix} \right)$$

⚠ In presence of informative censoring, it is critical that the mean, variance, and correlation structure over time is well-specify if we want to correctly "guess" and therefore avoid bias.

There are several equivalent ways to **parametrize** the model. An intuitive one is to model the mean at each timepoint in each group with one parameter (e.g. $\alpha_0(0)$, $\alpha_0(52)$, $\alpha_1(0)$, $\alpha_1(52)$). Most software use a different parametrisation where:

- the mean for reference group and timepoint is selected, e.g. α for the control group at time 0
- other means are modeled relative to this reference, e.g. the active group at time 0 has mean $\alpha + \beta$. This makes it easy to test effects like group effects ($\beta = 0$ instead of $\alpha_0(0) - \alpha_1(0) = 0$ in the other parametrisation).

- with only two timepoints, we will reparametrize the model as:

$$\begin{bmatrix} Y(0)|G=0 \\ Y(52)|G=0 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \alpha \\ \alpha + \beta_{time} \end{bmatrix}, \begin{bmatrix} \sigma_0^2(0) & \sigma_0(0)\sigma_0(52)\rho_0(0,52) \\ \sigma_0(0)\sigma_0(52)\rho_0(0,52) & \sigma_0^2(52) \end{bmatrix} \right)$$

while for the active group it is:

$$\begin{bmatrix} Y(0)|G=1 \\ Y(52)|G=1 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \alpha + \beta_{group} \\ \alpha + \beta_{time} + \beta_{group} + \beta_{interaction} \end{bmatrix}, \begin{bmatrix} \sigma_1^2(0) & \sigma_1(0)\sigma_1(52)\rho_1(0,52) \\ \sigma_1(0)\sigma_1(52)\rho_1(0,52) & \sigma_1^2(52) \end{bmatrix} \right)$$

where $\beta_{interaction}$ encodes the group difference at follow-up.

4. We will then **test null hypothesis** (same as in the univariate approach), i.e. no treatment effect on the mean change from baseline. This will involve a specific combination of parameters that depends on the parametrization of the model.

- Depending on the parametrisation we will either test $(\alpha_1(52) - \alpha_1(0)) - (\alpha_0(52) - \alpha_0(0)) = 0$ or $\beta_{interaction} = 0$

6 In R

6.1 Data management

For illustration, we will use only part of the dataset presented during the lecture:

```
## load dataset
data(armd.wide, package = "nlmeU")
## only keep specific columns
keep.col <- c("subject","treat.f",
       "visual0","visual4","visual12","visual24","visual52")
dfW <- armd.wide[,keep.col,drop=FALSE]
## display the first rows
head(dfW,5)
```

```
  subject treat.f visual0 visual4 visual12 visual24 visual52
1       1  Active      59      55       45       NA       NA
2       2  Active      65      70       65       65       55
3       3 Placebo      40      40       37       17       NA
4       4 Placebo      67      64       64       64       68
5       5  Active      70      NA       NA       NA       NA
```

Converting a dataset from wide to long format:

```
## convert to long format
dfL <- reshape(dfW, direction = "long", idvar = "subject",
        varying = paste0("visual",c(0,4,12,24,52)),
        timevar = "week", v.names = "visual")
## re-order dataset by subject
dfL <- dfL[order(dfL$subject),]
## remove row names
rownames(dfL) <- NULL
## character version of the time variable
dfL$time <- factor(dfL$week, levels = 1:5,
     labels = paste0("week",c(0,4,12,24,52)))
## display the first rows
head(dfL,6)
```

```
  subject treat.f week visual   time
1       1  Active    1     59  week0
2       1  Active    2     55  week4
3       1  Active    3     45 week12
4       1  Active    4     NA week24
5       1  Active    5     NA week52
6       2  Active    1     65  week0
```

Converting a dataset from long to wide format:

```
## easier to use when only a single time column
## this is why column week is removed
dfW.bis <- reshape(dfL[,c("subject","treat.f","visual","time")],
     direction = "wide",
     idvar = c("subject","treat.f"),
     timevar = "time",
     v.names = "visual"
     )
head(dfW.bis,5)
```

Complete case: restrict dataset to patients having full data over all timepoints:

```
dfW.CC <- dfW[rowSums(is.na(dfW))==0,]
dfL.CC <- dfL[dfL$subject %in% dfW.CC$subject,]
```

6.2 Descriptive statistics (numbers)

Computing summary statistics, in the long format, stratified on a variable (e.g. at each timepoint):

- using base R functions like `tapply`:

```
tapply(X = dfL$treat.f=="Active", ## "outcome" variable
       INDEX = dfL$time, ## "grouping" variable
       FUN = mean) ## function to apply
```

```
    week0     week4    week12    week24    week52 
0.5041667 0.5041667 0.5041667 0.5041667 0.5041667 
```

```
## using your own function by explaining what to do to a generic x
tapply(X = dfL$treat.f=="Active",
       INDEX = dfL$week,
       FUN = function(x){100*mean(x)})
```

```
       1        2        3        4        5 
50.41667 50.41667 50.41667 50.41667 50.41667 
```

- using pre-defined functions:

```
library(LMMstar)
dfS <- summarize(visual ~ week + treat.f, data = dfL, na.rm = TRUE)
dfS
```

```
   outcome week treat.f observed missing     mean       sd min median max
1   visual    1 Placebo      119       0 55.33613 15.00129  22   56.0  85
2   visual    2 Placebo      117       2 53.96581 15.90973  12   54.0  84
3   visual    3 Placebo      117       2 52.87179 17.20091   3   53.0  85
4   visual    4 Placebo      112       7 49.33036 18.51242   5   50.5  85
5   visual    5 Placebo      105      14 44.43810 18.53683  11   44.0  85
6   visual    1  Active      121       0 54.57851 14.82270  20   57.0  82
7   visual    2  Active      114       7 50.91228 15.81114  12   52.0  84
8   visual    3  Active      110      11 48.67273 17.47665  12   49.5  82
9   visual    4  Active      102      19 45.46078 18.08050   5   45.0  84
10  visual    5  Active       90      31 39.10000 18.40069   4   37.0  84
```

Correlation matrix:

```
cor(dfW[,paste0("visual",c(0,4,12,24,52))], use = "pairwise")
```

```
           visual0   visual4  visual12  visual24  visual52
visual0  1.0000000 0.8543813 0.7442610 0.6611932 0.5593174
visual4  0.8543813 1.0000000 0.8425869 0.7387614 0.6135206
visual12 0.7442610 0.8425869 1.0000000 0.8220768 0.7021200
visual24 0.6611932 0.7387614 0.8220768 1.0000000 0.8355586
visual52 0.5593174 0.6135206 0.7021200 0.8355586 1.0000000
```

Missing data pattern:

```
library(mice)
md.pattern(dfW, plot = FALSE)
```

```
    subject treat.f visual0 visual4 visual12 visual24 visual52
188       1       1       1       1        1        1        1  0
24        1       1       1       1        1        1        0  1
4         1       1       1       1        1        0        1  1
8         1       1       1       1        1        0        0  2
1         1       1       1       1        0        0        1  2
6         1       1       1       1        0        0        0  3
2         1       1       1       0        1        1        1  1
1         1       1       1       0        1        0        0  3
6         1       1       1       0        0        0        0  4
          0       0       0       9       13       26       45 93
```

6.3 Descriptive statistics (graph)

Boxplot plot (see left panel of Figure 1):

```
library(ggplot2)
gg.box <- ggplot(dfL, aes(x = time, y = visual, fill = treat.f))
gg.box <- gg.box + geom_boxplot()
gg.box
```

Spaghetti plot (see middle left panel of Figure 1):

```
gg.spa <- ggplot(dfL, aes(x = week, y = visual,
     group = subject, color = treat.f))
gg.spa <- gg.spa + geom_point() + geom_line()
gg.spa
```

Mean plot (see middle right panel of Figure 1):

```
gg.mean <- ggplot(dfS, aes(x = week, y = mean,
      group = treat.f, color = treat.f))
gg.mean <- gg.mean + geom_point() + geom_line() + ylab("visual")
gg.mean
```

Combine spaghetti and mean plot (see right panel of Figure 1):

```
gg.spa2 <- ggplot(mapping = aes(x = week, color = treat.f))
gg.spa2 <- gg.spa2 + geom_line(data = dfL, alpha = 0.3,
          aes(y = visual, group=subject))
gg.spa2 <- gg.spa2 + geom_point(data = dfS, aes(y = mean), size = 3)
gg.spa2 <- gg.spa2 + geom_line(data = dfS, aes(y = mean, group = treat.f),
    size = 1.5)
gg.spa2
```

Density plot (see Figure 3):

```
dfW$change52 <- dfW$visual52 - dfW$visual0
gg.dens <- ggplot(dfW, aes(change52, color = treat.f, fill = treat.f))
gg.dens <- gg.dens + geom_histogram(alpha = 0.45, aes(y=..density..),
   position = "identity")
gg.dens <- gg.dens + xlab("\u0394 Y (52)")
gg.dens
```

Scatterplot (see Figure 5):

```
gg.sca <- ggplot(dfW, aes(x=visual0, y=visual52,
     group = treat.f, color = treat.f))
gg.sca <- gg.sca + geom_point()
gg.sca <- gg.sca + xlab("visual at week 0") + ylab("visual at week 52")
gg.sca
```

6.4 Statistical analysis: univariate

Group comparisons using t-tests:

```
## compute change from baseline
dfW.CC$dvisual4 <- dfW.CC$visual4 - dfW.CC$visual0
dfW.CC$dvisual52 <- dfW.CC$visual52 - dfW.CC$visual0
## divide dataset according to treatment group
dfW.CCa <- dfW.CC[dfW.CC$treat.f == "Active",]
dfW.CCp <- dfW.CC[dfW.CC$treat.f == "Placebo",]
## difference in change at time 4 between groups
tt4 <- t.test(dfW.CCa$dvisual4,dfW.CCp$dvisual4)
## difference in change at time 12 between groups
tt52 <- t.test(dfW.CCa$dvisual52, dfW.CCp$dvisual52)
tt52
```

```

        Welch Two Sample t-test

data:  dfW.CCa$dvisual52 and dfW.CCp$dvisual52
t = -1.7781, df = 184.09, p-value = 0.07704
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -8.7909377  0.4566924
sample estimates:
mean of x mean of y
-15.12791 -10.96078
```

Adjustment for multiple comparisons:

```
p.adjust(c(tt4$p.value,tt52$p.value), method = "bonferroni")
```

```
[1] 0.06081505 0.15408670
```

T-test in a regression model:

```
e.tt <- lmm(dvisual52 ~ treat.f, repetition =~treat.f|subject,
     structure = "IND", data = dfW.CC)
model.tables(e.tt) ## short version of summary(e.tt)
```

```
              estimate   se  df  lower  upper  p.value
(Intercept)     -10.96 1.64 101 -14.22 -7.699 1.42e-09
treat.fActive    -4.17 2.34 184  -8.79  0.457 7.70e-02
```

6.5 Statistical analysis: multivariate

Fit a mixed model:

```
library(LMMstar)
e.lmm <- lmm(visual ~ time*treat.f, ## model for the mean
      repetition = ~time|subject, ## structure of the data
      structure = "UN", ## model for the variance
      data = dfL) ## data
logLik(e.lmm)
```

```
[1] -4151.224
```

Estimated model parameters for the mean structure:

```
model.tables(e.lmm)
```

```
                         estimate    se  df  lower   upper  p.value
(Intercept)                55.336 1.367 238  52.64 58.0289 0.00e+00
timeweek4                  -1.281 0.765 231  -2.79  0.2254 9.52e-02
timeweek12                 -2.352 1.091 220  -4.50 -0.2007 3.23e-02
timeweek24                 -6.020 1.318 212  -8.62 -3.4211 8.42e-06
timeweek52                -11.311 1.599 193 -14.46 -8.1576 2.70e-11
treat.fActive              -0.758 1.925 238  -4.55  3.0348 6.94e-01
timeweek4:treat.fActive    -2.204 1.087 232  -4.35 -0.0617 4.38e-02
timeweek12:treat.fActive   -3.508 1.560 222  -6.58 -0.4330 2.55e-02
timeweek24:treat.fActive   -3.070 1.895 216  -6.81  0.6661 1.07e-01
timeweek52:treat.fActive   -4.866 2.317 199  -9.44 -0.2963 3.70e-02
```

Extract reference level (i.e. what the `(Intercept)` corresponds to)

```
levels(e.lmm)$reference
```

```
   time   treat.f
"week0" "Placebo"
```

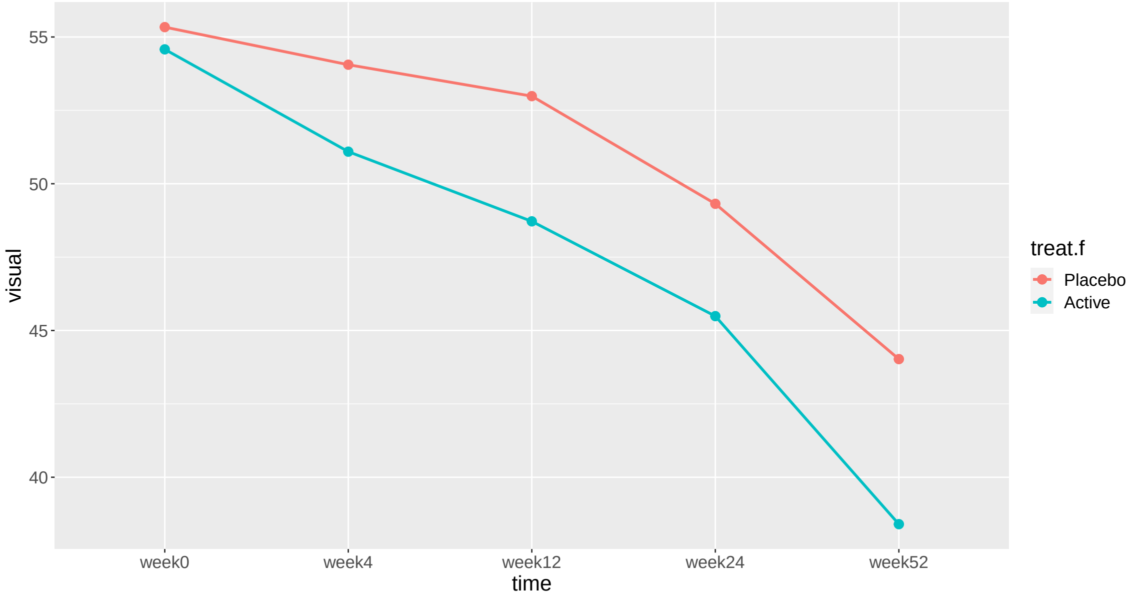
Testing multiple model parameters:

```
anova(e.lmm)
```

```
                     ** mean coefficients **
 - F-test
              statistic df.num df.denom      p.value
time         13.7048511      4 202.3355 6.600918e-10
treat.f       0.1548786      1 238.0257 6.942684e-01
time:treat.f  1.8397879      4 207.1469 1.224733e-01
```

Display of the model fit:

```
plot(e.lmm, ci = FALSE)
```



Estimated mean at each timepoint and in each group:

```
dummy.coef(e.lmm)
```

```
     time treat.f estimate       se       df    lower    upper
1   week0 Placebo 55.33613 1.366923 238.0249 52.64332 58.02895
2   week4 Placebo 54.05485 1.460500 234.7088 51.17749 56.93222
3  week12 Placebo 52.98448 1.588206 232.4446 49.85536 56.11359
4  week24 Placebo 49.31611 1.721041 223.2780 45.92455 52.70768
5  week52 Placebo 44.02519 1.767665 210.6591 40.54061 47.50977
6   week0  Active 54.57851 1.355579 238.0266 51.90805 57.24898
7   week4  Active 51.09301 1.456179 238.4434 48.22439 53.96163
8  week12  Active 48.71891 1.597738 240.5417 45.57157 51.86626
9  week24  Active 45.48891 1.748162 234.4195 42.04479 48.93302
10 week52  Active 38.40129 1.835338 224.4565 34.78459 42.01799
```

Variance and correlation estimates:

```
confint(e.lmm, effects = c("correlation","variance"))
```

```
                   estimate  lower  upper
sigma                14.911 13.623 16.321
k.week4               1.066  0.997  1.140
k.week12              1.158  1.061  1.264
k.week24              1.246  1.129  1.377
k.week52              1.261  1.123  1.416
rho(week0,week4)      0.857  0.819  0.888
rho(week0,week12)     0.739  0.674  0.793
rho(week0,week24)     0.664  0.583  0.732
rho(week0,week52)     0.517  0.409  0.611
rho(week4,week12)     0.840  0.797  0.874
rho(week4,week24)     0.749  0.684  0.802
rho(week4,week52)     0.591  0.494  0.675
rho(week12,week24)    0.825  0.777  0.863
rho(week12,week52)    0.698  0.620  0.763
rho(week24,week52)    0.840  0.793  0.877
Note: estimates and confidence intervals for sigma, k, rho have been back-transformed.
```