Data analysis based on SISSREM: Shiny Interactive, Supervised and Systematic report from REpeated Measures data.

Pablo Hernández Alonso
Máster de Bioinformática y Bioestadística
Área 2 – Subárea 2

Nombre Consultor/a: Núria Pérez Álvarez
Nombre Profesor/a responsable de la asignatura: Carles Ventura Royo

Fecha Entrega: 06/2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Data analysis based on SISSREM: Shiny Interactive, Supervised and Systematic report from REpeated Measures data*
Nombre del autor:	*Pablo Hernández Alonso*
Nombre del consultor/a:	*Núria Pérez Álvarez*
Nombre del PRA:	*Carles Ventura Royo*
Fecha de entrega (mm/aaaa):	06/2019
Titulación:	*Bioinformática y Bioestadística*
Área del Trabajo Final:	*Área 2 – Subárea 2*
Idioma del trabajo:	*Inglés*
Palabras clave	*Repeated measures, Linear-mixed model, Shiny app.*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

Los métodos longitudinales son los procedimientos de elección para los científicos que ven sus fenómenos de interés como dinámicos. Sin embargo, dada la dificultad del uso de modelos lineales mixtos (MLM), se emplean otras aproximaciones más sencillas, pero subóptimas y en ocasiones desaconsejadas por la estructura de los datos. El objetivo de este trabajo es desarrollar una metodología sistemática y supervisada para que investigadores biomédicos con nivel bajo-medio de estadística puedan realizar un análisis de medidas repetidas.

Mediante el uso del lenguaje de programación R, hemos desarrollado una aplicación en línea Shiny denominada SISSREM (*Shiny Interactive, Supervised and Systematic report from REpeated Measures data*). Ésta puede: i) instruir al usuario en la comprensión de un análisis MLM para medidas repetidas con una base de datos de ejemplo; ii) permitir al usuario analizar sus propios datos; y iii) permitir al usuario crear un informe interactivo, supervisado y sistemático para exportarlo desde la aplicación Shiny. El núcleo principal de la aplicación consiste en un recorrido guiado a través de un análisis predeterminado con una base de datos de ejemplo y las decisiones sistemáticas que deberían realizarse en un análisis de MLM. Por lo tanto, se ha estructurado en diferentes módulos que permiten explorar y tratar los datos, así como realizar el análisis de MLM, guardar datos y/o generar un informe en formato .PDF, .HTML o .DOCX.

SISSREM (*https://sissrem.shinyapps.io/SISSREM_v1/*) es una aplicación funcional cuyo objetivo es simplificar el uso y difundir la utilidad de los MLM en el área de investigación biomédica.

Abstract (in English, 250 words or less):

Longitudinal methods are the procedures of choice for scientists who see their phenomena of interest as dynamic. However, given the difficulty of using linear mixed models (LMM), other simpler approaches are used, but suboptimal and sometimes discouraged by the structure of the data. The objective of this work is to develop a systematic and supervised methodology so that biomedical researchers with low-average level of statistics can perform an analysis of repeated measures.

By using R programming language, we have developed a Shiny online application named SISSREM (Shiny Interactive, Supervised and Systematic report from REpeated Measures data). It can: i) instruct the user in the understanding of a LMM analysis for repeated measures with an example database; ii) allow the user to analyze their own data; and iii) allow the user to create an interactive, supervised and systematic report to be exported from the Shiny application. The main core of the application consists of a guided tour through a predetermined analysis with a sample database and the systematic decisions that should be made in an LMM analysis. Therefore, it has been structured in different modules that allow you to explore and process the data, as well as perform the LMM analysis, save data and/or generate a report in .PDF, .HTML or .DOCX format.

SISSREM (*https://sissrem.shinyapps.io/SISSREM_v1/*) is a functional application whose objective is to simplify the use and disseminate the usefulness of LMM in biomedical research.

Index

List of Figures

1. Introduction

1.1 Context and justification of the Project

Even though there are many software to perform linear-mixed models (LMMs) of repeated measures (RM) and/or longitudinal data, the procedure for performing them is difficult and frequently requires the supervision from a Statistician. Given this LMM complexity, the current solution for many researchers is to skip them and conduct alternative – with reduced accuracy and sometimes inappropriate - approaches such as RM-ANOVA. Considering the fact that many biomedical researchers employ data such as longitudinal or hierarchical, that should be analyzed using LMM, our Shiny app is intended to guide researchers with low-to-medium statistical knowledge to understand (via an example) and/or perform (via their own data) the results of a LMM problem. The R code of this Shiny app will be accessible to the overall community. This is of relevance considering that: i) Researchers will have the possibility to use the Shiny app; ii) Researchers and programmers will have the possibility of modifying the code and adapting it to their needs; iii) Researchers and programmers will have the possibility to use the core R code design to generate alternative analysis beyond LMMs.

1.2 Objectives of the Project

The main objective of this final master project (FMP) is: *to develop and publish a Shiny R code-based application to generate an **interactive**, **systematic** and **supervised** Rmarkdown exportable report (.PDF or .HTML) from user's data coming from studies with repeated measures; together with instruct on how to perform and interpret linear-mixed models with an example database.*

This is split into **five main sequential objectives**:

- **Objective 1. Systematic procedure for RM analysis:** to systematically define the procedure for conducting LMMs in case of RMs data.
- **Objective 2. R code for RM analysis*:** to write the R code for conducting the analysis (both pre-analysis and main analysis) and allow its execution with an example.
- **Objective 3. Shiny app development:** to generate the code of the Shiny app core and integrate the R code.
- **Objective 4. Rmarkdown report development:** to generate the Rmarkdown report.
- **Objective 5. Videotutorial:** to record a video informing on how to use our Shiny app and relevant FAQs (frequently asked questions).

** During the generation of the R code, it will be evaluated in a testing Shiny app to assure its further compatibility.*

1.2.1 Specific objectives

Objectives 1-4 are essential for the development of the Shiny app, whereas the importance of the Objective 5 will depend on the success of the previous objectives. Moreover, within each main objective (1-4) we will prioritize each sub-objective according to the general Calendar (Tasks and Milestones).

OBJECTIVE 1: Systematic procedure for RM analysis

1. To design the procedure to pre-analyze the data (Exploratory Data Analysis (EDA) using statistical tests and/or graphs.
2. To design the procedure for conducting LMM of RM data.

OBJECTIVE 2: R code for RM analysis
Write R code*:

1. To allow the user to follow a guided explanation on how to treat and interpret RM data analysis (using LMMs) with an example database.
2. To allow the user to upload a database to be analyzed.
3. To allow/inform the user about of their dataset (i.e., inform about missing values) based on data-preanalysis such as EDA and other tests and plots (i.e., correlation among variables, etc.).

OBJECTIVE 3: Shiny app development
Write Shiny R code:

1. To design the main core layout of the Shiny app (i.e., menu, panels).
2. To define the main layout modules to be included in the application.
3. To integrate the R code from **Objective 2** into the Shiny code*.

OBJECTIVE 4: Rmarkdown report development

1. To define the statistical tests and graphs to be included in the final systematic and supervised report (choosing .PDF/.HTML formats) that will be generated using a predefined otherwise modifiable Rmarkdown file (.Rmd).

OBJECTIVE 5: Videotutorial

1. To define a script of the information that should be included in the video
2. To record the video on how to use the Shiny app

** During the generation of the R code, it will be evaluated in a testing Shiny app to assure its further compatibility.*

1.3 Approach and Methods

The present FMP was initially defined to construct an R code able to generate a report analysis (i.e., output of results) of data coming from a RMs structure. Therefore, we could have decided to generate an Rmarkdown (.Rmd) file for that purpose. However, we finally decided to define this report as interactive, supervised and systematic. For that reason, we finally decided to take advantage of the powerful and free potential of the Shiny web applications [1] and elaborate our report in a Shiny app way. This is the best option considering our objectives of not only allowing the user to perform statistical approaches online, but also to learn how to do them in a supervised and systematic way. To make the overall procedure transparent to the supervisor, the app will be accessible and will include the date of last update.

1.4 Work Planning

Considering the objectives of this FMP, we will priorize the information to be included in the Shiny app. This means that we will focus on dealing with the minimum information to be included in the analysis (R code), in the application (Shiny app) and in the Rmarkdown report. As we cannot fully predict the time to be invest into the definition of each module of the Shiny app or Rmarkdown report, this priorization will be of great importance during the FMP execution. We will always make sure that the milestones are met in their right time in order not to harm the overall FMP.

The following **Gantt chart** shows the chronogram for the project.

Tasks and milestones are listed below:

OBJECTIVE 1: Systematic procedure for RM analysis

T1. Bibliographic documentation about the topic(s): RM, EDA, LMM, assumptions
T2. Setting the pre-analysis of the data (data and plots)
T3. Setting the analysis of LMM (data and plots)

M1. Systematic procedure on how to perform LMM [22/03/2019]

OBJECTIVE 2: R code for RM analysis

T4. Selection of a database to conduct the example
T5. Select the best libraries and functions to pre-analyze and conduct LMM
T6. Set R code for describing the database (edit data, inform about missing values, etc.)
T7. Write the supervised output comments to be included in the analysis (Shiny and Rmarkdown report)

M2. R code for pre-analyzing and analyzing LMM [14/04/2019]

OBJECTIVE 3: Shiny app development

T8. Generation of a Rstudio Cloud to host the Shiny app
T9. Definition of the main core layout of the Shiny app

T10. Integration of the generated R code into the Shiny app

T11. Definition of the main modules of the Shiny app

M3. Generation of a beta Shiny app version with the basic modules [31/04/2019]

OBJECTIVE 4: Rmarkdown report development

T12. Definition of the overall scheme of the Rmarkdown report

T13. Write of the Rmarkdown code with the systematic information to be included

T14. Integrate Rmarkdown into the Shiny app

M4. Generation of the Rmarkdown report [07/05/2019]

OBJECTIVE 5: Videotutorial

T15. Definition of a script on what to record about the Shiny app use

T16. Record the video of how to use the Shiny app

M5. Record of a how-to-use the Shiny app video [13/05/2019]

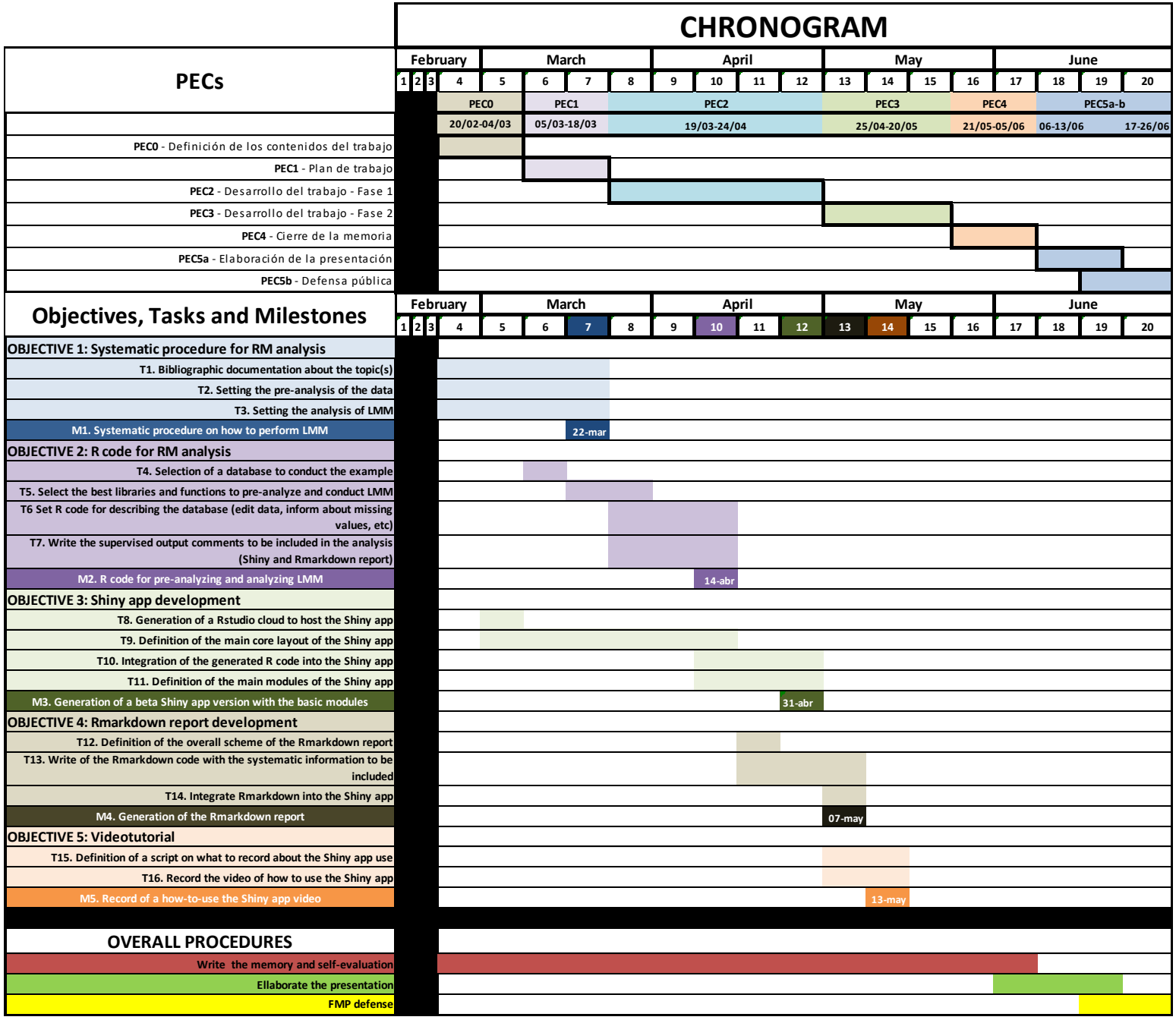


Figure 1. Gantt chart of the project

1.5 Brief summary of the products obtained

In this FMP we have created: i) a systematic procedure for performing LMMs with RMs data; ii) the R-Shiny code for creating a web-based program to execute the analysis (https://sissrem.shinyapps.io/SISSREM_v1/); iii) the R code for generating a Rmarkdown report for the LMM analysis in the Shiny application (SISSREM). This application works as a platform to analyze RM data in an interactive, systematic and supervised way.

1.6 Brief description of the other chapters of the memory

In the following chapters, we will explain the use of LMMs specifically applied to the analysis of repeated measures data. The advantages of using LMMs versus other statistical approaches will be also discussed and documented. Moreover, we will describe the programming language used for the implementation of the Shiny application. Finally, we will show the results (i.e., execution of the application) and discussion, conclusions, glossary, references and annex included.

As a general consideration, R packages will be written in **bold blue**, whereas functions within packages will be written in *italics blue*.

2. Overall statistical approach

2.1 Statistical models

A statistical model is a mathematical model that embodies a set of statistical assumptions concerning the generation of sample data (and similar data from a larger population). A statistical model represents, often in considerably idealized form, the data-generating process [2]. As described by Herman Adèr quoting Kenneth Bollen "*a statistical model is a formal representation of a theory*" [3].

Therefore, *what is the general problem that we aim to address by modeling?*
It is defined by:

- **Material:** a collection of variables, each variable being a vector of readings of a specific trait on the samples in an experiment.
- **Problem:** In what way does a variable Y depend on other variables ($X_1, \ldots, X_n$) in the study.

The response variable is the one whose content we are trying to model with other variables, called the explanatory variables. In any given model there is one **response variable** (Y) and there may be many **explanatory variables** (like $X_1, \ldots .X_n$). Therefore, a statistical model defines a mathematical relationship between the X_i's and Y. The model is a representation of the real Y that aims to replace it as far as possible. At least the model should capture the dependence of Y on the X_i's

Before starting with LMM, we will give a brief introduction of "normal linear model" and "repeated measures ANOVA", as they are referred and compared to the LMMs during this project.

2.2 Normal multiple linear regression model

The equation for the normal multiple linear regression model (**Equation 1**) is as follows:

$$y_i = \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi} + \varepsilon_i$$

Equation 1. Normal multiple linear regression

This model is form by a unique **random effect (RE)**, the so called error term ($\boldsymbol{\varepsilon_i}$, normally and independently distributed (NID) (0, σ^2)). The parameters of the model are the regression coefficients (β_1, β_2,…, β_p), and the error variance (σ^2). Usually, $x_{1i} = 1$, and so β_1 is a constant or intercept.
Or in a matrix form as shown in **Equation 2**:

$$y = X\beta + \varepsilon$$
$$\varepsilon \sim N_n(0, \sigma^2 I_n)$$

Equation 2. Matrix form of the normal multiple linear regression

Where:

$\mathbf{y} = (y_1, y_2, ..., y_n)'$ is the response vector,
$\mathbf{X}$ is the model matrix, with characteristic row $x'_i = (x_{1i}, x_{2i}, ..., x_{pi})$,
$\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_p)'$ is the vector of regression coefficients,
$\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, ..., \varepsilon_n)'$ is the vector of errors,
$\mathbf{N_n}$ represents the n-variable multivariate-normal distribution,
$\mathbf{0}$ is an $n \times 1$ vector of zeroes and
$\mathbf{I}_n$ is the order- n identity matrix.

2.3 Repeated measures ANOVA

RM-ANOVA is the equivalent of the one-way ANOVA, but for related, not independent groups, and it is the extension of the dependent *t*-test [4, 5]. A RM-ANOVA is also referred to as a within-subjects ANOVA or ANOVA for correlated samples. All these names imply the nature of the RM-ANOVA, that of a test to detect any overall differences between related means.

There are many complex designs that can require the use of RMs, but the most simple case is the one-way RM-ANOVA. This test requires one independent variable and one dependent variable. The dependent variable must be continuous (interval or ratio) and the independent variable categorical (either nominal or ordinal).

We can analyze data using a RM-ANOVA for two types of study design. Studies that investigate either:

- ✓ Changes in mean scores over three or more time points
- ✓ Differences in mean scores under three or more different conditions.

The main feature that defines the structure of the RM design is that the subjects are being measured more than once on the same dependent variable. While the RM-ANOVA is used when there is just one independent variable, if there are two independent variables (e.g., both time and condition are measured), we will need to use a two-way RM-ANOVA.

In the following subsection we will explore the LMMs based on the book by Brady T. West and collaborators [6]. After LMMs, we will go back to RM-ANOVA to justify its specific use concerning each different situations.

2.4 Linear-mixed models

2.4.1 An overview

A linear mixed model (LMM), also known as mixed-effect models, is a parametric linear model for clustered (e.g., students in classrooms), longitudinal, or RMs (i.e., subjects are measured repeatedly over time or under different conditions) – an therefore dependent – data that quantifies the relationships between a continuous dependent variable and various predictor variables while controlling for non-independence among data points [7]. A LMM may include both **fixed-effect** (FE) parameters associated with one or more continuous or categorical covariates and REs associated with one or more **random factors** (RF) [8]. The mix of fixed and random effects gives the LMM its name. Whereas FE parameters describe the relationships of the covariates to the dependent variable for an entire population, REs are specific to clusters or subjects within a population [9]. Consequently, REs are directly used in modeling the random variation in the dependent variable at different levels of the data. It should be noted that non-LMMs [10] and generalized LMMs [11, 12] (in which the dependent variable may be a binary, ordinal, or count variable) are not considered in this project.

The theoretical base of LMMs is well-established, and the methodology has applications in many areas not involving repeated measures (reviewed in [13]). McLean *et al.* (1991) provided a general introduction to LMMs [14], and Ware (1985) [15] gave an overview of their application to the analysis of RMs. Very similar LMMs are known under many different names: mixed (effect) models (in statistics specially biostatistics); Multilevel models (sociology); Hierarchical models (psychology); Random coefficient models (econometrics). LMMs provide researchers with powerful and flexible analytic tools for these types of data [6].

Because of their advantage in dealing with missing values, LMMs are often preferred over more traditional approaches such as RM-ANOVA. The use of LMMs methodology for the analysis of RMs is becoming increasingly common due to the development of widely available software. However, LMMs for RMs are still hard to conduct and interpret.

2.4.2 Models for longitudinal or repeated-measures data

Longitudinal data arise when multiple observations are made on the same subject (or unit of analysis) over time or in a particular order. We will commonly refer to “over time” as it is the most frequent repetition in biomedical longitudinal data. RMs data may involve measurements made on the same unit over time, or under changing experimental or observational conditions. Measurements made on the same variable for the same subject are likely to be correlated (e.g., measurements of body weight for a given subject will tend to be similar over time). Models fitted to longitudinal or RMs data involve the estimation of covariance parameters to capture this correlation. Unlike measures taken on different patients, RMs data, however, are not independent. In other

words, repeated observations on the same individual will be more similar to each other than to observations on other individuals. This demands a statistical methodology that can account for this dependency [16, 17].

Repeated-measures data may be defined as data sets in which the dependent variable is measured more than once on the same unit of analysis across levels of a RMs factor (or factors). The RMs factors, which may be time or other experimental or observational conditions, are often referred to as within-subject factors.

- By **longitudinal data**, we mean data sets in which the dependent variable is measured at several points in time (or particular order) for each unit of analysis. We usually conceptualize longitudinal data as involving at least two repeated measurements made over a relatively long period of time. In contrast to RMs data, dropout of subjects is often a concern in the analysis of longitudinal data.

The software procedures (e.g., generalized linear model (GLM)) that were available for fitting models to longitudinal and RMs data prior to the advent of software for fitting LMMs accommodated only a limited range of models (e.g., traditional RM-ANOVA). Software for LMMs has other advantages over software procedures capable of fitting traditional RM-ANOVA models [6]:

- LMM software procedures allow subjects to have missing time points (i.e., unequal numbers of measurements). In contrast, software for traditional RM-ANOVA drops an entire subject from the analysis if the subject has missing data for a single time point (i.e., *complete-case analysis* [18]).
- LMMs allow for the inclusion of time-varying covariates in the model (in addition to a covariate representing time), whereas software for traditional RM-ANOVA does not.
- When analyzing longitudinal data with RM-ANOVA techniques, time is a within-subject factor, where the levels of the time factor are assumed to be the same for all subjects. In contrast, LMMs allow the time points when measurements are collected to vary for different subjects.
- LMMs provide tools for the situation in which the trajectory of the outcome varies over time from one subject to another.

Levels of data in the models

We can understand clustered, RMs, and longitudinal data sets as multilevel data sets. The concept of "levels" of data is based on ideas from the hierarchical linear modeling (HLM) literature [19, 20]. All data sets appropriate for an analysis using LMMs have at least two levels of data. We describe the example data sets that we analyze as two-level or three-level data sets, depending on how many levels of data are present. In a RM or longitudinal data set, we consider data with at most three levels:

- **Level 1** denotes observations at the most detailed level of the data. It represents the RMs made on the same unit of analysis. The continuous dependent variable is always measured at Level 1 of the data.

- **Level 2** represents the next level of the hierarchy. It represents the units of analysis.

- **Level 3** represents the next level of the hierarchy, and generally refers to clusters of units in clustered longitudinal data sets.

The important feature of the different types of data is that the dependent variable is measured more than once for each unit of analysis (level 2), with the RMs likely to be correlated (at level 1). Therefore, clustered longitudinal data sets combine features of both clustered and longitudinal data. More specifically, the units of analysis are nested within clusters, and each unit is measured more than once. We refer to clustered, RMs, and longitudinal data as hierarchical data sets because the observations can be placed into the above-mentioned levels of a hierarchy in the data.

2.4.3 Variables in a LMM

Apart from the dependent variable, in the context of a LMM analysis we have two types of arguments: **fixed component** and **random component**. The distinction between fixed and random factors and their related effects on a dependent variable are critical in the context of LMMs. Unfortunately, the distinction between the two is not always obvious, and is not helped by the presence of multiple, often confusing definitions in the literature [20]. Absolute rules for how to classify something as a fixed or random effect generally are not useful because that decision can change depending on the goals of the analysis [6].

Levels of a **fixed factor** (FF) are chosen so that they represent specific conditions, and they can be used to define contrasts (or sets of contrasts) of interest in the research study. In contrast to the levels of FFs, the levels of random factors do not represent conditions chosen specifically to meet the objectives of the study. RFs are considered in an analysis so that variation in the dependent variable across levels of the RFs can be assessed, and the results of the data analysis can be generalized to a greater population of levels of the RF.

As already stated, the name LMMs comes from the fact that these models are linear in the parameters, and that the covariates, or independent variables, may involve a mix of FEs and REs. Fixed effects (i.e., regression coefficients or fixed-effect parameters) describe the relationships between the dependent variable and continuous covariates (e.g., glucose levels, body weight) which take on values from a continuous range, or with factors (e.g., as gender or treatment group) which are categorical. FEs may describe contrasts or differences between levels of a FF (e.g., between males and females) in terms of mean responses for the continuous dependent variable, or they may describe the effect of a continuous covariate on the dependent variable. FEs are unknown constant parameters associated with either continuous covariates or the levels of categorical factors in an LMM. Estimation of these parameters in LMMs is generally of

intrinsic interest, because they indicate the relationships of the covariates with the continuous outcome variable.

When the levels of a factor can be thought of as having been sampled from a sample space, such that each level is not of intrinsic interest (e.g., classrooms or clinics that are randomly sampled from a larger population of classrooms or clinics), the effects associated with the levels of those factors can be modeled as REs in an LMM. Random effects are random values associated with the levels of a RF (or factors) in an LMM. These values, which are specific to a given level of a RF, usually represent random deviations from the relationships described by FEs. For example, REs associated with the levels of a RF can enter an LMM as **random intercepts** (i.e., representing random deviations for a given subject or cluster from the overall fixed intercept), or as **random coefficients-intercepts** (i.e., representing random deviations for a given subject or cluster from the overall FEs) in the model. In contrast to FEs, which are represented by constant parameters in an LMM, REs are represented by (unobserved) random variables, which are usually assumed to follow a normal distribution.

2.4.4 General specification for an individual observation

The general specification of a LMM presented in this section refers to a model for a longitudinal two-level data set, with the first index, *t*, being used to indicate a time point, and the second index, *i*, being used for subjects. To simplify it, we specify a LMM in the following equation for a hypothetical two level longitudinal data set as was described in by Brady T. West and collaborators [6]. In this specification shown in **Equation 3**, Y_{ti} represents the measure of the continuous response variable Y taken on the *t*-th occasion for the *i*-th subject:

$$\boldsymbol{Y_{ti} = \beta_1 \cdot X_{ti}^{(1)} + \beta_2 \cdot X_{ti}^{(2)} + \cdots + \beta_p \cdot X_{ti}^{(p)}\ ]\ FIXED\ COMPONENT}$$
$$\boldsymbol{+ b_{1i} \cdot Z_{ti}^{(1)} + \cdots + b_{qi} \cdot Z_{ti}^{(q)} + \varepsilon_{ti}\ ]\ RANDOM\ COMPONENT}$$

Equation 3. Equation of LMM for RMs

The value of *t* (*t* = 1, …, n_i), indexes the n_i longitudinal observations on the dependent variable for a given subject, and *i* (*i* = 1, …, *m*) indicates the *i*-th subject (unit of analysis). We assume that the model involves two sets of covariates, namely the X and Z covariates.

- The first set contains *p* covariates, $X^{(1)}$, …, $X^{(p)}$, associated with the FEs (β_1, …, β_p).
- The second set contains *q* covariates, $Z^{(1)}$, …, $Z^{(q)}$, associated with the REs (b_{1i},…, b_{qi}) that are specific to subject *i*.

Importantly, when including an intercept into the fixed and/or random model, the first X and/or Z are a vector of 1's. The X and/or Z covariates may be continuous or indicator variables. The indices for the X and Z covariates are denoted by superscripts.

* For each X covariate, $X^{(1)}$, …, $X^{(p)}$, the terms $X_{ti}^{(1)}$,…, $X_{ti}^{(p)}$ represent the *t*-th observed value of the corresponding covariate for the *i*-th subject. We assume that the *p* covariates may be either time-invariant characteristics of the individual subject (e.g., gender) or time varying for each measurement (e.g., time of measurement, or weight at each time point).

Each β parameter represents the FE of a one-unit change in the corresponding X covariate on the mean value of the dependent variable, Y, assuming that the other covariates remain constant at some value. These β parameters are FEs that we wish to estimate, and their linear combination with the X covariates defines the fixed portion of the model. The effects of the Z covariates on the response variable are represented in the random portion of the model by the q REs, b_{1i}, …, b_{qi}, associated with the i-th subject. In addition, ε_{ti} represents the residual associated with the t-th observation on the i-th subject. The REs and residuals are RVs, with values drawn from defined distributions. We assume that for a given subject, the residuals are independent of the REs.

2.4.5 Estimation in LMMs

In the LMM, we estimate the fixed-effect parameters (β) and the covariance parameters (θ). The main methods commonly used to estimate these parameters in LMM are **maximum likelihood** (ML) and **restricted maximum likelihood** (REML) estimations [21].

a) Maximum likelihood estimation

In general, maximum likelihood (ML) estimation is a method of obtaining estimates of unknown parameters by optimizing a likelihood function. To apply ML estimation, we first construct the likelihood as a function of the parameters in the specified model, based on distributional assumptions. The maximum likelihood estimates (MLEs) of the parameters are the values of the arguments that maximize the likelihood function (i.e., the values of the parameters that make the observed values of the dependent variable most likely, given the distributional assumptions).

Therefore, as a limitation, variance estimates are biased, but tests between two models with differing fixed and random effects are possible.

b) Restricted maximum likelihood estimation

REML estimation is an alternative way of estimating the covariance parameters. The REML estimates of are based on optimization a REML log-likelihood function. REML estimation was introduced in the early 1970s by Patterson and Thompson (1971) as a method of estimating variance components in the context of unbalanced incomplete block designs. REML is often preferred to ML estimation, because it produces unbiased estimates of covariance parameters by considering the loss of degrees of freedom that results from estimating the FEs.

Therefore, as a limitation, can only test between two models that have same FEs, but variance estimates are unbiased.

2.4.6 Tools for model selection

When analyzing clustered and RMs/longitudinal data sets using LMMs, researchers are faced with several competing models for a given data set. These competing models describe sources of variation in the dependent variable and at the same time allow researchers to test hypotheses of interest. It is an important task to select the "best" model. It means a model that is parsimonious in terms of the number of parameters used, and at the same time is best at predicting (or explaining variation in) the dependent variable.

In selecting the best model for a given data set, we consider research objectives, sampling and study design, previous knowledge about important predictors, and important subject matter considerations. We also use analytic tools, such as the hypothesis tests and the information criteria. Importantly, the specific procedure for model selection is shown as **milestone 1** (**M1**) in this FMP and it is a combination of different proposed approaches [6, 22].

2.4.6.1 Likelihood ratio tests

Likelihood ratio tests (LRTs) are a class of tests that are based on comparing the values of likelihood functions for two models (i.e., the nested and reference models) defining a hypothesis being tested [23]. LRTs can be employed to test hypotheses about covariance parameters or FE parameters in the context of LMMs.

2.4.6.2 Information criteria

Another set of tools useful in model selection are referred to as **information criteria** [24]. The information criteria - sometimes referred to as fit criteria - provide a way to assess the fit of a model based on its optimum log-likelihood value, after applying a penalty for the parameters that are estimated in fitting the model. They provide a way to compare any two models fitted to the same set of observations (i.e., the models do not need to be nested). A smaller value of the criterion indicates a better fit.

- The **Akaike information criterion** (AIC, Akaike in 1973) may be calculated based on the (ML or REML) log-likelihood of a fitted model as follows:

$$AIC = -2 \cdot l\left(\hat{\beta}, \hat{\theta}\right) + 2p$$

Equation 4. AIC

In **Equation 4**, *p* represents the total number of parameters being estimated in the model for both the FEs and REs. Note that the AIC in effect "penalizes" the fit of a model for the number of parameters being estimated by adding 2*p* to the −2 log-likelihood.

- The **Bayes information criterion** (BIC, Gideon E. Schwarz in 1978) is also commonly used and may be calculated as in **Equation 5**:

$$BIC = -2 \cdot l\left(\hat{\beta}, \hat{\theta}\right) + p \cdot \ln(n)$$

Equation 5. BIC

The BIC applies a greater penalty for models with more parameters than does the AIC, because we multiply the number of parameters being estimated by the natural logarithm of *n*, where *n* is the total number of observations used in estimation of the model.

In the context of the Shiny app, we have decided to use AIC for model selection. However, we inform the user that both information criteria methods should be combined.

2.4.6.3 Alternative tests for fixed-effect parameters

- A ***t*-test** is often used for testing a single fixed-effect parameter (e.g., $\boldsymbol{H_0}$: β = 0 vs. $\boldsymbol{H_A}$: β ≠ 0) in an LMM. It appears in the summary of each model next to every FE.

- An ***F*-test** can be used to test linear hypotheses about multiple FEs in an LMM. For example, we may wish to test whether any of the parameters associated with the levels of a FF are different from zero.

In the context of a LMM, the null distribution of the t-statistic does not in general follow an exact t distribution [6]. Unlike the case of the standard linear model, the number of degrees of freedom for the null distribution of the test statistic is not equal to n – p (where p is the total number of fixed-effect parameters estimated). Instead, we use approximate methods to estimate the degrees of freedom [25]. The approximate methods for degrees of freedom for both t-tests and F-tests are matter of debate [22].

2.4.7 Model-building strategies

A primary goal of model selection is to choose the simplest model that provides the best fit to the observed data. There are several choices concerning which fixed and random effects should be included in an LMM. There are also many possible choices of variance-covariance structures for the matrix of the REs and the residuals. All these considerations have an impact on the estimations from the model [10, 26]. The process of building an LMM for a given set of longitudinal or clustered data is an iterative one that requires a series of model-fitting steps and investigations, and selection of

appropriate mean and covariance structures for the observed data. Model building typically involves a balance of statistical and subject matter considerations as there is no single strategy that applies to every application.

In this project, we have used the "Top-down strategy" as it has been more widely used in the context of analyzing repeated measures data with LMMs, it was suggested by Verbeke and Molenberghs [27]. The specific procedure will be seen in the Results section. Briefly, it involves starting with a model that includes the maximum number of FEs that we wish to consider in a model, and systematically analyze the structure of the random and fixed components by using AIC.

2.4.8 Checking model assumptions – diagnostics

After fitting an LMM, it is important to carry out model diagnostics to check whether distributional assumptions for the residuals are satisfied and whether the fit of the model is sensitive to unusual observations [26]. Diagnostic methods for standard linear models are well established in the statistics literature. In contrast, diagnostics for LMMs are more difficult to perform and interpret, because the model itself is more complex, due to the presence of random effects and different covariance structures.

In general, model diagnostics should be part of the model-building process throughout the analysis of a longitudinal data set [22]. However, we consider diagnostics only for the final model fitted for simplicity. As happened with section **"2.4.6 Tools for model selection"**, the diagnostic process is explained in the **M1** of the project. However, here we introduce some useful concepts.

2.4.8.1 Residual diagnostics

Informal techniques are commonly used to check residual diagnostics. These techniques depend on the human mind and eye and are used to decide whether a specific pattern exists in the residuals as we will explain in results' section. In the context of the standard linear model, the simplest example is to decide whether a given set of residuals plotted against predicted values represents a random pattern or not. These residual vs fitted plots are used to verify model assumptions and to detect outliers and potentially influential observations.

In general, residuals should be assessed for normality, constant variance, and outliers.
In the context of LMMs, we consider conditional residuals and their standardized versions [20, 28]. A conditional residual is the difference between the observed value and the conditional predicted value of the dependent variable. To alleviate problems with the interpretation of conditional residuals that may have unequal variances, we consider scaling (i.e., dividing) the residuals by their standard deviation to obtain the standardized residuals.

2.4.8.2 Diagnostics for random effects

It is recommended to use standard diagnostic plots (e.g., histograms, Q-Q plots, and scatterplots) to investigate REs predictors (i.e., empirical best linear unbiased predictors (EBLUPs), due to their properties) for potential outliers that may warrant further investigation. In general, checking EBLUPs for normality is of limited value, because their distribution does not necessarily reflect the true distribution of the REs [29].

2.4.9 Other considerations

2.4.9.1 Missing data

In general, analyses using LMMs are carried out under the assumption that missing data in clustered or longitudinal data sets are **missing at random** (MAR). Under the assumption that missing data are MAR, inferences based on methods of ML estimation in LMMs are valid [12, 27]. The MAR pattern means that the probability of having missing data on a given variable may depend on other observed information but does not rest on the data that "would have been observed" but were in fact missing. For example, if subjects in a study do not report their body weight because the actual (unobserved or missing) weights are too small or too large, then the missing body weight data are not MAR. However, if a subject's current body weight does not depend on whether he or she reports it, but the probability of failing to report it depends on other observed information (e.g., illness or previous weight), then the data may be considered MAR.

As we have mentioned before, missing data are frequent in longitudinal studies, often due to dropout. Multivariate RM-ANOVA models are often used in practice to analyze RM or longitudinal data sets, but LMMs offer two primary advantages over these multivariate approaches when there are missing data as we have already commented in section **"2.4.2 Models for longitudinal or repeated-measures data"**. Because of these key differences, LMMs are much more flexible analytic tools for longitudinal data than RM-ANOVA models, under the assumption that any missing data are MAR.

Importantly, if the clear majority of subjects in a longitudinal study have data present at only a single time point, an LMM approach may not be warranted, because there may not be enough information present to estimate all the desired covariance parameters in the model. In this situation, simpler regression models should probably be considered because issues of within-subject dependency in the data may no longer apply [6].

2.4.9.2 Correlations

It is essential to consider the correlation between variables being part of the fixed component of the model. We should not add variables that are highly correlated ($r > 0.70$ based on a Review of methods by Dormann *et al.* [30]) because when we include correlated variables into a parametreic model it will have a problem of coefficients' stability as it will lose accuracy in determining the coefficients. This threshold for correlation coefficient has been found to be an appropriate indicator for when

collinearity begins to severely distort model estimation and subsequent prediction. In case that we have correlation between variables, we should choose one of the correlated variables using biological knowledge/reasoning to select the most meaningful variable or conduct a dimension-reduction analysis (e.g., Principal Components Analysis) leaving a single variable that accounts for most of the shared variance among the correlated variables.

2.4.9.3 Centering covariates

Centering covariates at specific values (i.e., subtracting a specific value, such as the mean, from the observed values of a covariate) has the effect of changing the intercept in the model, so that it represents the expected value of the dependent value at a specific value of the covariate (e.g., the mean), rather than the expected value when the covariate is equal to zero (which is often outside the range of the data). In addition to changing the interpretation of the intercept in a linear model, centered covariates often reduce the amount of collinearity among the covariates with associated fixed effects in the model.

2.4.9.4 Residual covariance structure

In contrast to the standard linear model, the residuals associated with repeated observations on the same subject in an LMM can be correlated. We assume that the n_i residuals in the ε_i vector for a given subject, i, are random variables that follow a multivariate normal distribution with a mean vector 0 and a positive definite symmetric variance-covariance matrix of residuals. We also assume that residuals associated with different subjects are independent of each other. Further, we assume that the vectors of residuals, ε_1, …, ε_m, and random effects, b_1,…, b_m, are independent of each other.

The specific form of this matrix depends upon context. For example, when observations are sampled independently within groups and are assumed to have constant error variance and thus the only free parameter to estimate is the common error variance. The *lmer* function in the **lme4** package handles only models of this form. In contrast, if the observations in a "group" represent longitudinal data on a single individual, then the structure may be specified to capture autocorrelation among the errors, as is common in observations collected over time. In the context of the initial version of our Shiny application we have not included the modulation of the variance-covariance matrix of residuals. However, as it will be further commented, we have used the *lme* function in the **nlme** package because it can handle autocorrelated and heteoscedastic errors that we plan on including in our application in future upgrades.

2.4.10 Why using LMMs?

In many ways, LMMs are difficult to use/interpret, but RM-ANOVA have been lately considered as an 'old-fashioned' approach. However, we can affirm that it is never better or more accurate than LMMs, although a lot simpler. We should use the simplest analysis that gives accurate results and answers the research question. However, it is

rare to find an analysis where the flexibility of LMMs will not be advantageous in either giving accurate results or answering a more sophisticated research question.

Here are some guidelines on similarities and differences based on "*The Analysis Factor*" [31]:

1. Simple design, complete data, normal residuals

If the design is very simple and there are no missing data, you will very likely get identical results from RM-ANOVA and a LMM. By simple, I mean something like a pre-post design (with only two repeats) or an experiment with one between-subjects factor and another within-subjects factor. In that case, RM-ANOVA is usually fine. In fact, the flexibility of mixed models becomes more advantageous the more complicated the design.

2. Non-normal residuals

Both RM-ANOVA and LMMs assume that the dependent variable is continuous, unbounded, and measured on an interval or ratio scale and that residuals are normally distributed. There are, however, generalized LMMs that work for other types of dependent variables (e.g., categorical, ordinal, discrete counts). Thus, if we have one of these outcomes, ANOVA is not an option. There is no RM-ANOVA equivalent for count or logistic regression models. There are generalized estimating equations (GEE) models, but they are closer in many ways to mixed in terms of setting up data, estimation, and how we measure model fit. We cannot calculate sums of squares by hand, for example, the way we can in RM-ANOVA.

3. Clustering

In many designs, there is a RM over time (or space), but subjects are also clustered in some other grouping. For example, students within classroom or patients within hospital. A RM-ANOVA cannot incorporate this extra clustering of subjects in some other clustering, but LMMs can. In fact, this kind of clustering may became really complex.

4. Missing data

As already mentioned, LMMs do a much better job of handling missing data. RM-ANOVA can only use listwise deletion, which can cause bias and reduce power substantially. Therefore, we should use RM-ANOVA only when missing data is minimal.

5. Time as continuous and time-varying covariates

RM-ANOVA cannot take into account time-varying covariates, whereas LMMs can include this information. Moreover, RM-ANOVA can only treat a repeat as a categorical factor. Therefore, if the measurements are made repeatedly over time and we want to treat time as continuous, we cannot do that in RM-ANOVA. E.g., we have measured body weight during months 1, 2, 7 and 12 of a body weight-loss intervention. In LMMs we

have the choice to treat those 4 time points as either 4 discrete categories or as true numbers, which accounts for the different spacing of the weeks. RM-ANOVA can only do the former. This is going to be commented in the **Results** section (page 37) with the example database.

6. Differing number of repeats

RM-ANOVA falls apart when repeats are unbalanced, which is very common in observed data. A common study is to record some repeated behavior for individuals, then compare some aspect of that behavior under different conditions. This kind of studies exists in many fields. One compared the diameter of four species of cherry trees at breast height in areas that were and were not exposed to an invasive pest. Those trees were observed – and not "planted" –, meaning that there was a different number of each of the four species in each plot. Therefore, some plots had many repeated data points for each species, while others had only a few. RM-ANOVA cannot incorporate the fact that each plot has a different number of each type of species. In fact, it can only use one measurement for each type. We can deal with this by averaging multiple measures for each type. However, by doing this we will be under-representing the true variability in the data as those averages are not real data points (i.e., they are averages with variability around them). Fortunately, LMMs can account for this variability and the unbalance design.

Therefore, RM-ANOVA is a good tool for some very specific situations. Once we deviate from those situations, trying to use it may be considered as an inadequate approach.

3. Material and Methods

3.1 Sample dataset

In the program, we have included some example data to allow the user to be trained by using it. Sample dataset is part of the **ADER** package (*data*(fruit)). We have selected this dataset because it has also been used by Cayuela L. 2018 [22] to explain the execution of a RMs analysis in R. Therefore, we have replicated his analysis in our program.

In this dataset, we have data regarding cherry tree fructification. We want to explore whether fruit production (FRUIT) depends on tree size, measured as the diameter at breast height (DBH). For this, 179 trees are measured. The same individuals (TAG) are resampled for 3 years (YEAR). The DBH is measured only once for all three years and, although this can increase by one year for another, the change is so small (they are adult trees) that it has not been taken into account. There are a total of 428 observations.

By using this RM design, we assume that there is an effect of the individual (TAG, random effect) which affects to the relationship between the dependent variable Y (FRUIT) and predictors Xs (DBH & YEAR, fixed effects).

We have additionally created another dataset with missing values in the variables FRUIT, DBH and YEAR. We have used the package **mice** and function *ampute* to create this dataset. After setting a seed (*set.seed*(1234)), the arguments for *ampute* were: **prop = 0.5** and **mech = "MAR"**. Therefore, missing values were generated using the method MAR in a cumulative proportion of 0.5 among variables (i.e., 50%, 0.5/3 = c.a. 16.6% of missing values in each variable). This allowed us to generate the following number of missing values: 70 (DBH), 78 (FRUIT) and 66 (YEAR).

3.2 EDA: Exploratory data analysis

As in all data analysis, it is advisable to examine the data before starting with statistical modeling. For this reason, we have included a specific section for exploratory data analysis (EDA). The seminal work in EDA derived from John Tukey in 1977 [32, 33]**.** It is considered an approach/philosophy for data analysis which specifically refers to the critical process of performing initial investigations on data so as to: discover patterns; extract important variables; spot anomalies; test hypothesis and/or check assumptions with the help of summary statistics and graphical representations. By showing an EDA we are making a “trailer clip” of the data (16).

3.2.1 Which techniques are used?

Most EDA techniques are graphical in nature with a few quantitative techniques. The reason for the heavy reliance on graphics is that by its very nature the main role of EDA is to open-mindedly explore, and graphics gives the analysts incomparable power to do

so, inviting the data to reveal its structural secrets, and being always ready to gain some new, often unsuspected, insight into the data. In combination with the natural pattern-recognition capabilities that we all possess, graphics provides incomparable power to carry this out [34].

The graphical techniques employed in EDA are often quite simple, consisting of various techniques of:

1. Plotting the raw data such as data traces, histograms, probability plots, lag plots and block plots.
2. Plotting simple statistics such as mean plots, standard deviation plots, box plots, and main effects plots of the raw data.

In the development of the Shiny application we have used a specific R package named **DataExplorer** to illustrate the EDA analysis. A specific EDA module will be included in the application as will be explained in the Results section.

3.3 R programming language

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS [35]. In order to generate our R code we have used R version v. 3.5.2. More information regarding R may be found in https://www.r-project.org/.

As many general purpose statistical computer programs, R consists of packages that are physically stored in the library of R. To generate our application we have used the following packages: **car, corrplot, DataExplorer, dichromat, dplyr, DT, editData, ggplot2, htmltools, influence.ME, lme4, lmerTest, MASS, mice, MuMIn, naniar, nlme, plotly, questionr, R.devices, rcompanion, readr, rmarkdown, shiny, shinyjs, shinythemes, shinyWidgets, sjPlot, stringr, summarytools, tools, usdm, xtable**.

There are different packages in R for performing LMMs [36]. However, **lme4** and **nlme** are the most used. In this FMP we have used **nlme** – which appeared before **lme4** - because it has been more widely used in the context of RM analysis compared to **lme4**, access to different objects from the results is easier and allow the definition of matrix structures [37]. We will only go in deep of the package **nlme** because it has been used for conducting the LMMs in our application.

3.3.1 Function lme() from nlme package

As we have already said, there are several facilities in R for fitting mixed models to data, the most ambitious of which is the **nlme** library (an acronym for non-linear mixed effects), described in detail by Pinheiro and Bates [26]. Despite its name, this library includes facilities for fitting linear mixed models (*lme*) along with nonlinear mixed

models and generalized linear mixed models (*nlme*, for example, for logistic and Poisson regression). The following are the main argumentes of *lme*:

Fixed and random component of the function

Function *lme* uses two different arguments "fixed" and "random" to inform about the variables [38]:

- If there are no fixed effects, the nomenclature is: *lme***(Y ~ RE)**.
- If there are both fixed and random effects, we specify random effects: *lme***(Y ~ FE, random = ~ 1|RE)**.

In most LMM it is assumed that RE have a mean of 0 and that what we want to quantify if the variation in the constant (**1|RE**; the vertical line means "given the following distribution of REs") due to the differences between the levels of the factor RE.

- If we want to specify a RE, not only over the constant of the model, but also over some FEs, the model is written as**:** *lme***(Y ~ FE, random = ~ FE | RE)** which is equivalent to *lme***(Y ~ FE, random = ~ 1+FE | RE)**.

- If we have more than one RE we will use the argument: **random = list(~ 1|RE_1, ~ 1|RE_2, …, ~1|RE_i).**

- If there are different REs, but some are nested inside others, then: **random = ~ 1 | RE_1/RE_2/…/RE_i**, meaning that there are "i" REs with "RE_i" nested inside RE_{i-1} that is nested inside RE_{i-2} and so on until the RE_1.

Specifying data

The specific argument of the function is: **data = name_of_database**.

Correlation structure

An optional **corStruct** object describing the within-group correlation structure. By default, *lme* uses an unstructured D matrix for the variance-covariance matrix of the random effects. In this first version of our Shiny application we have used this unstructured correlation structure.

Method

The argument of the function is: **method = "value"**, this value is REML by default but may be changed to ML.

Handling missing data

The argument is: **na.action = "x"**, where x may be:

- **na.fail** (default option) leads to an error in presence of missing values.
- **na.omit** deals with missing values by removing the corresponding lines in the dataset.
- **na.exclude** ignores the missing values but, compared to na.omit, it enables to have outputs with the same number of observations compared to the original dataset.
- **na.pass** will continue the execution of the function without any change. If the function cannot manage missing values, it will lead to an error.

3.4 Shiny code

For the development of this project, we have used the Shiny R-coding format which is part of the package **shiny**. Shiny is a web application framework for R that can help turn your analyses into interactive web applications [1]. Shiny applications have two components, a user interface (UI) object and a server function, that are passed as arguments to the *shinyApp* function that creates a Shiny app object from this UI/server pair. The source code for both components is listed below.

To use Shiny we first created a shinyapps.io profile [39] to publish our program online. Moreover, we have used Rsudio cloud [40] which is an online version of the Rstudio (i.e., a user friendly IDE (integrated development environment) for R, really convenient tool in building shiny apps). Importantly, no HTML, CSS, or JavaScript knowledge are required when using Shiny. Among Shiny's advantages we have: i) easy to learn, easy to use; ii) the development time is minimized; iii) excellent tool for data visualization; iv) it has a very strong backing: R language.

To create our program, we have used a single file shiny app, meaning that the UI and server where written in the same file (app.R). The UI controls the outlook of the web page, whereas the server (a live R session) controls the "logic" (i.e., instructions required to build the app).

3.4.1 How does a Shiny application work?

The "server" keeps monitoring the UI. Whenever there is a change in the UI, the "Server" will follow some instructions (run some R code) accordingly and update the UI's display. This is the basic idea of reactive expression, which is an distinguish feature of Shiny we will talk about later. The following is an example of the parts [1]:

```
library(shiny)
# UI
ui <- fluidPage()
# SERVER
server <- function(input, output){
}
# EXECUTION
shinyApp(ui = ui, server = server)
```

Therefore, in the server we create the functions and outputs that will be invoked in the UI. The UI is *de facto* an HTML file where we include title for the application, modules or sections and in our application, we have included module-dependent left panels to communicate with the program.

When using Shiny, you should consider what you want to do (R-code in the Shiny Server), how you want to do it (Shiny code, Server) and how you want it to look like (Shiny code, UI). Therefore, Shiny requires you not to only have a knowledge about R coding, but also have learned Shiny functions.

3.5 Rmarkdown

R Markdown is a file format for making dynamic documents with R [41]. An R Markdown document is written in markdown (an easy-to-write plain text format) and contains chunks of embedded R code, like the document below. R uses a particular package called **rmarkdown** to process .Rmd format files.

R Markdown files are the source code for rich, reproducible documents. In practice, authors almost always knit and convert their documents at the same time.

- *knit* - You can knit the file. The rmarkdown package will call the **knitr** package. knitr will run each chunk of R code in the document and append the results of the code to the document next to the code chunk. This workflow saves time and facilitates reproducible reports. Therefore, in the R Markdown paradigm, each report contains the code it needs to make its own graphs, tables, numbers, etc. The author can automatically update the report by re-knitting.

- **convert** - You can convert the file. The **rmarkdown** package will use the pandoc program [42] to transform the file into a new format. For example, you can convert your .Rmd file into an HTML, PDF, or Microsoft Word file.

Conversion lets you do your original work in markdown, which is very easy to use. You can include R code to knit, and you can share your document in a variety of formats. In this project, we have created our .Rmd file in the way it may be transformed into PDF, HTML or Microsoft Word file (.DOCX).

4. Results and Discussion

In the present FMP, we have developed a systematic procedure for analyzing repeated-measured data based on LMM in a systematic and supervised fashion. This procedure has been translated into R-code and then integrated into a supervised Shiny application that can be accessed through https://sissrem.shinyapps.io/SISSREM_v1/. One of the milestones (**M1, Annex 1**) of the project was to create a "**Systematic procedure on how to perform LMM**". It has been integrated into this results section. **M2** and **M3** were combined and are part of the R syntax of the program (app.R, **Annex 2**). **M4** was the design of a Rmarkdown report to get the results from the application (**Annex 3**). Finally, **M5** was the recording of a video of our application basics (https://www.youtube.com/watch?v=DpHEpyitzUQ).

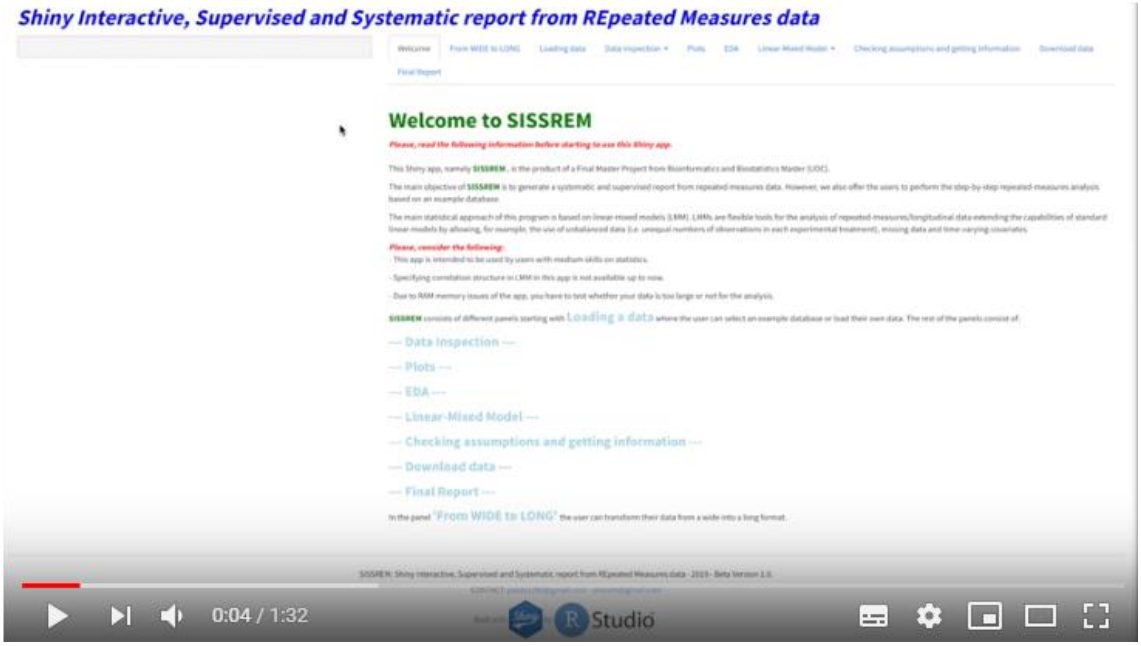


Figure 2. Screenshot of the introduction video

Our Shiny application, SISSREM, works perfectly with the example database. However, the users are advised that due to memory limit, large datasets may have reduced performance.

In this section, we will explain the different modules that integrate SISSREM, together with a brief explanation about our systematic procedure. It is important to consider that this application is intended to be kept up to date by implementing more options (e.g., structure of the variance-covariance residual matrix) and/or including other modules. The results presented here are those related to the Shiny application SISSREM version 1.0 (https://sissrem.shinyapps.io/SISSREM_v1/). Due to the free-hosting characteristics, below you may find the instructions to run the application from a Rstudio Cloud alternative.

How to execute the program based on the app.R file in Rstudio Cloud

The following link https://rstudio.cloud/project/363486 allows you to open the SISSREM public project after you log in with your own credentials. For executing it, you just need to press the "Run" button.

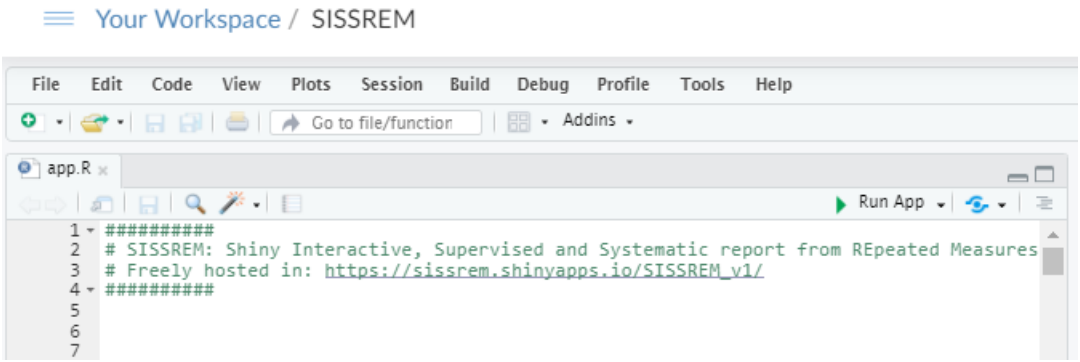


Figure 3. Screenshot of the Workspace for SISSREM in Rstudio Cloud

Importantly, when you access a project created by someone else, RStudio Cloud automatically creates a temporary copy of the original project for you. You can make edits to it, but none of your changes will be reflected in the original. If you want to keep the changes you have made, just save a copy of the project for yourself by pressing the Save a Permanent Copy button.

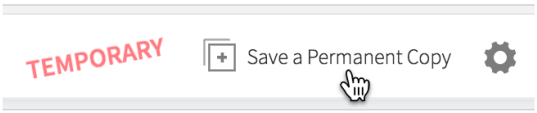


Figure 4. Screenshot of how to save a permanent copy of the project

Modules of SISSREM

1. Welcome
2. From WIDE to LONG
3. Loading data
4. Data inspection
    a. Data summary
    b. Full data view
    c. Missing values
    d. Correlation between variables
    e. Box-cox transformation
    f. Scale, center and other transformations
    g. Update dataset
5. Plots
6. EDA
7. Linear-Mixed Model
    a. What is a linear-mixed model (LMM)?
    b. Step 0. Base models
    c. Step 1. Definition of the model: beyond optimal model
    d. Step 2. Structure of RANDOM effects
    e. Step 3. Structure of FIXED effects
    f. Step 4. Final model
8. Checking assumptions and getting information
9. Download data
10. Final Report

➔ **We will show the appearance of the program by executing "Example Database (complete cases)".**

1. **'Welcome'**

Figure 5. Screenshot of "Welcome" module

2. **'From WIDE to LONG'** → Module for changing from wide to long data

➔ <u>This module is not used in the example.</u>

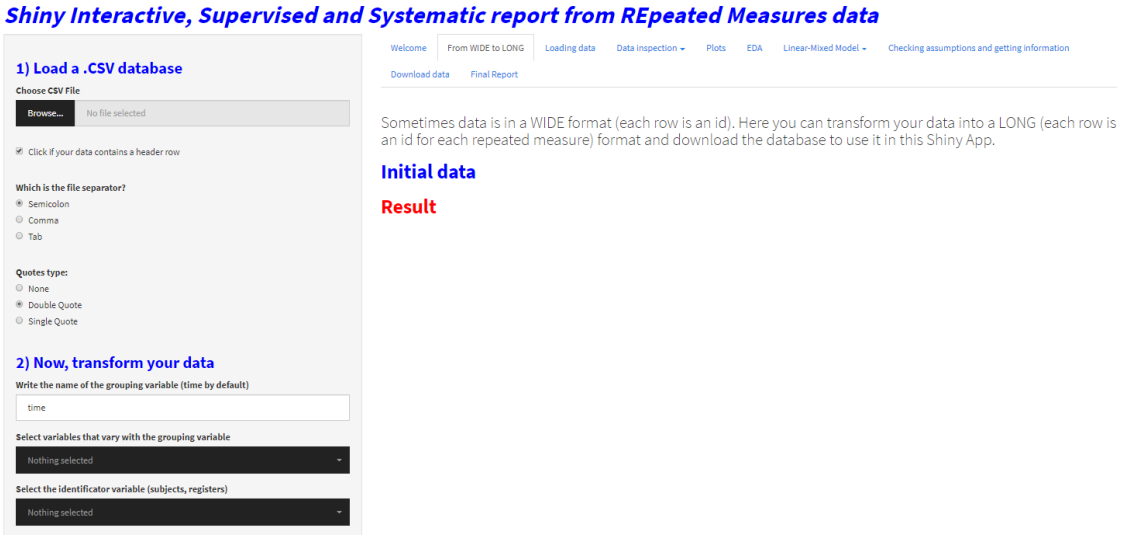


Figure 6. Screenshot of "From WIDE to LONG" module and options

3. **'Loading data' –>** Loading database or selecting an example database

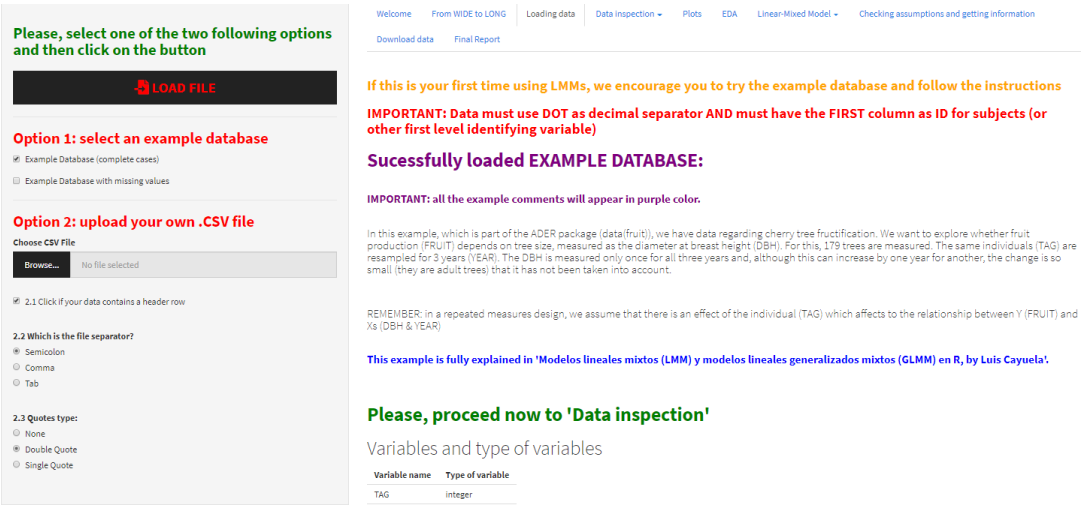


Figure 7. Screenshot of "Loading data" module after clicking 'Example database' and options

One of the main decissions when conducting a LMM analysis is how to include the reapeted-measured factor. In this example, it is time (variable YEAR) and it could have been treated as numerical or as a categorical variable. The main reason is because time is measured discretely (required for using it as categorical) but there are enough numerical values (1, 2 and 3 years) to fit a line to our data. If trees were had been analyzed in different time points - and this is relevant for the analysis -, then we would only have the option of trating YEAR as numeric. An example of this last situation is having data for years: 1, 1.2, 1.5, 2.3 and 2.9.

When we treat time as continuous in a mixed model, we are really fitting a "line" between time (YEAR) and the outcome Y (Ln FRUIT) for each subject (TAG). When we fit a line, it takes into account the distance between each value of the measuring factor. However, when we treat time as categorical, instead of fitting a line, we are fitting a mean of Y (Ln FRUIT) for each category of time (1, 2 and 3 in our data). In the Step 4, we will explain why we chose YEAR as categorical in this particular example.

4. **'Data inspection'** → Exploring the database
    a) **'Data summary'**
        i. Select variables to EXPLORE
        ii. Change between variable types
            1. Are the variables in the correct variable type (i.e., numeric, factor, etc.)? → Change them accordingly
        iii. Transform dependent Y variable
        iv. Center/scale or other transformation of the data
            1. Center/scale the data
            2. Other transformations of a variable

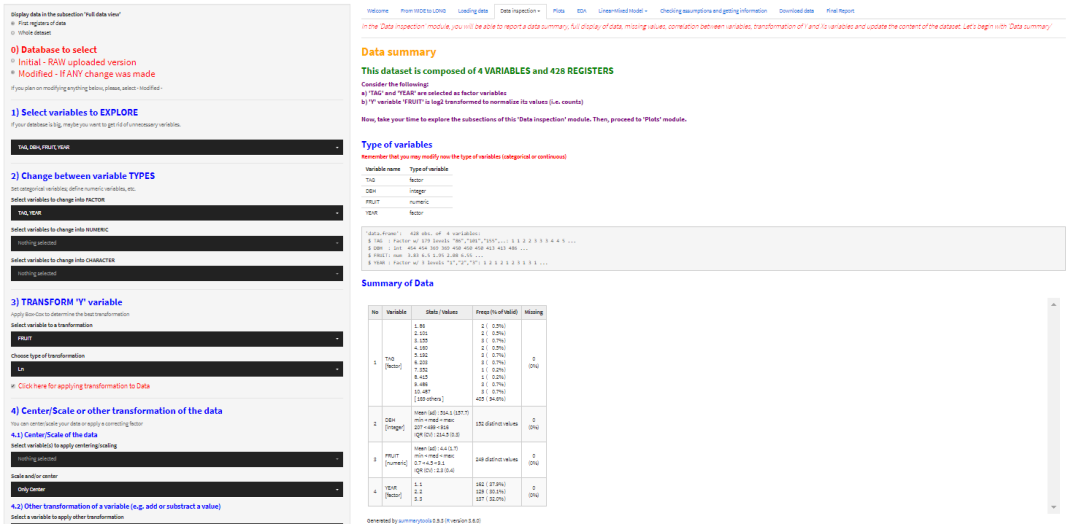


Figure 8. Screenshot of "Data summary" sub-module and options

b) 'Full data view'

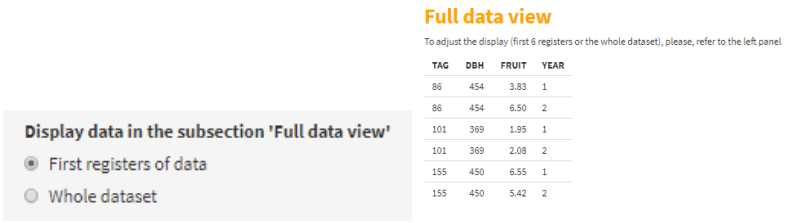


Figure 9. Screenshot of "Full data view" sub-module

c) 'Missing values'

Here we can see the structure of this module (**Figure 10**). However, as there are no missing values in this data, there is no information to plot.

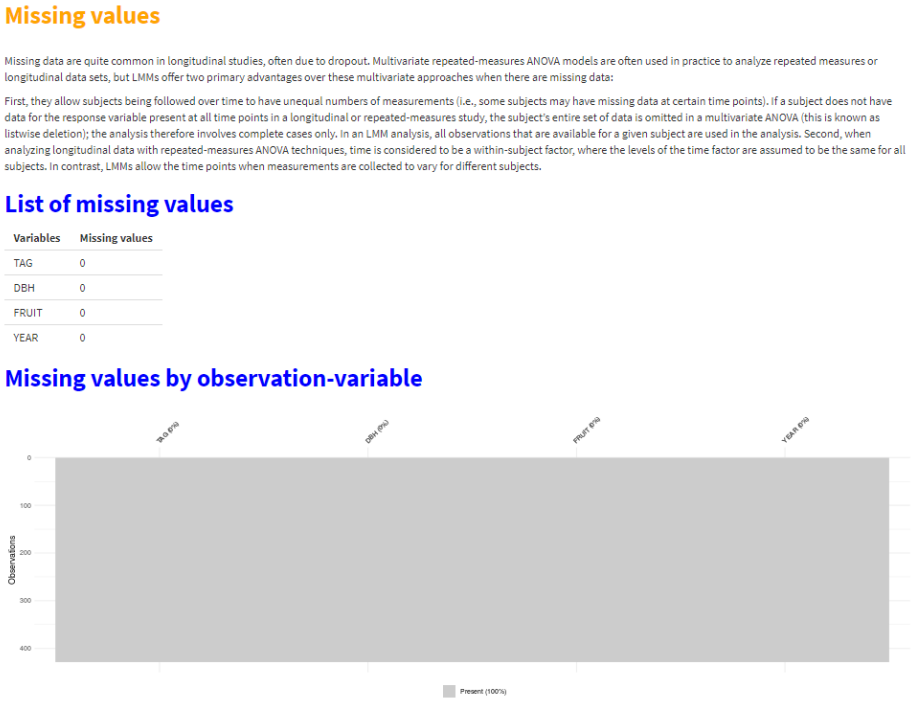


Figure 10. Screenshot of "Missing values" sub-module

d) 'Correlation between variables' → collinearity of continuous variables

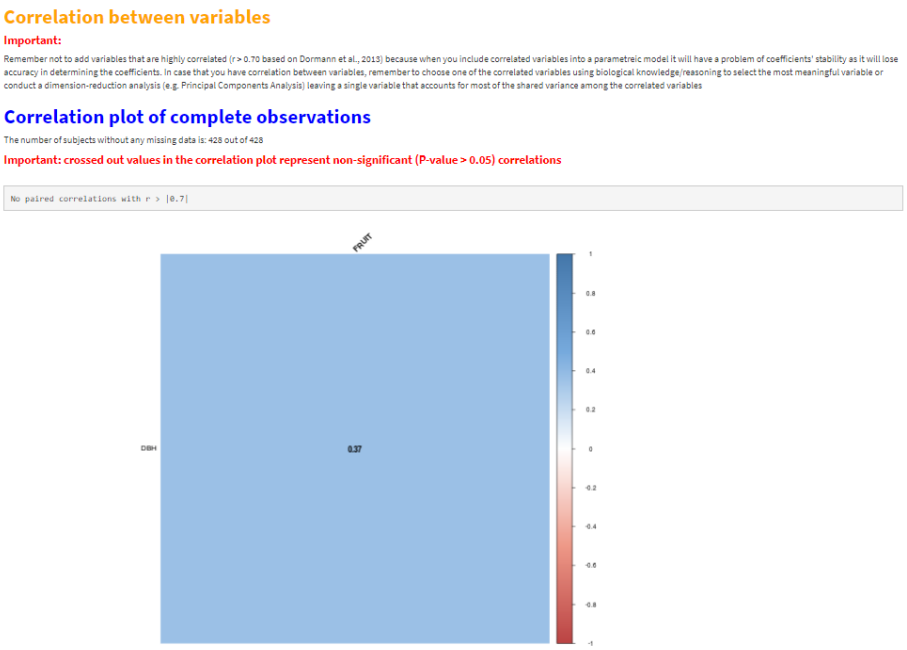
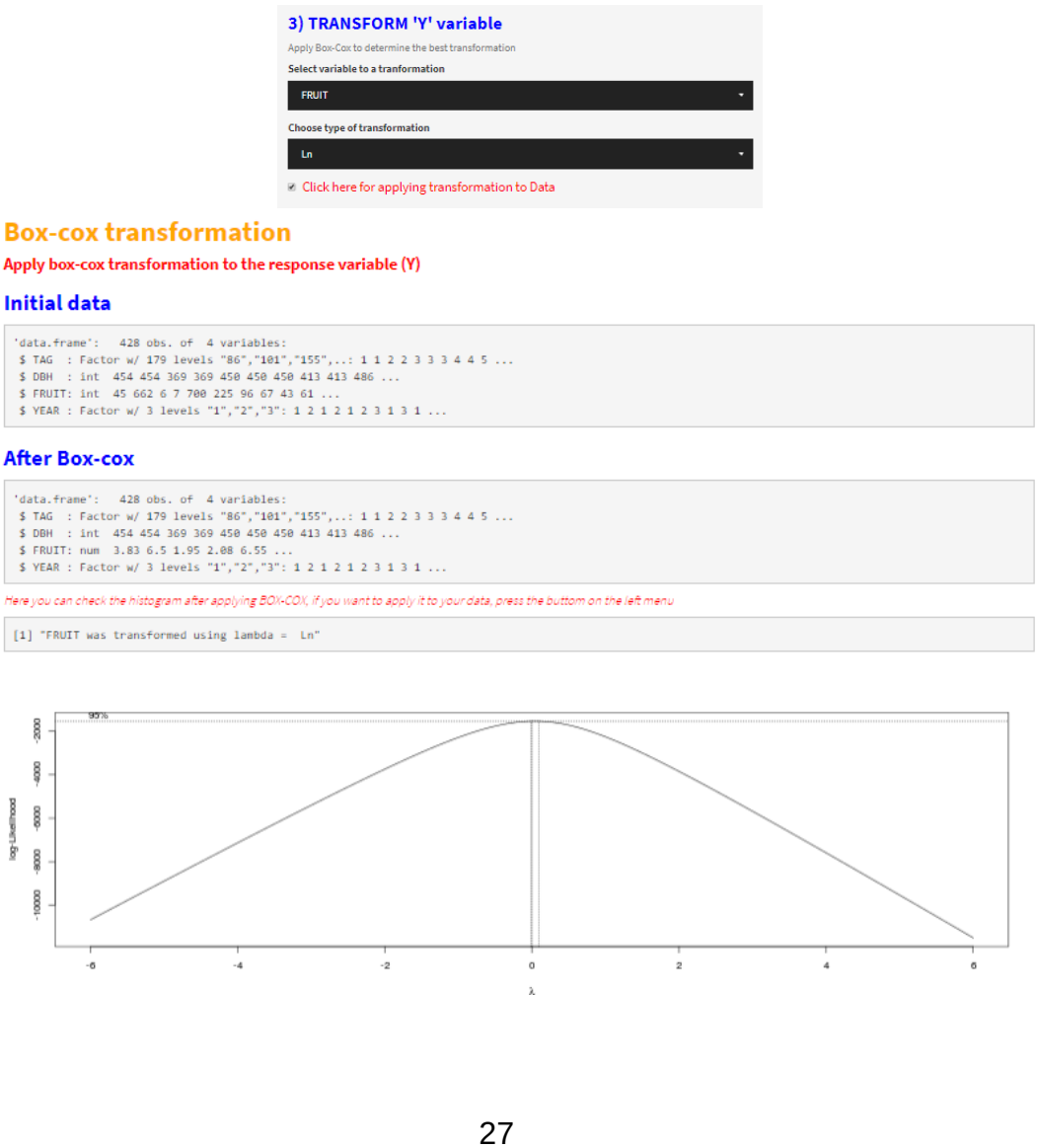


Figure 11. Screenshot of "Correlation between variables" sub-module

e) 'Box-cox transformation' analysis



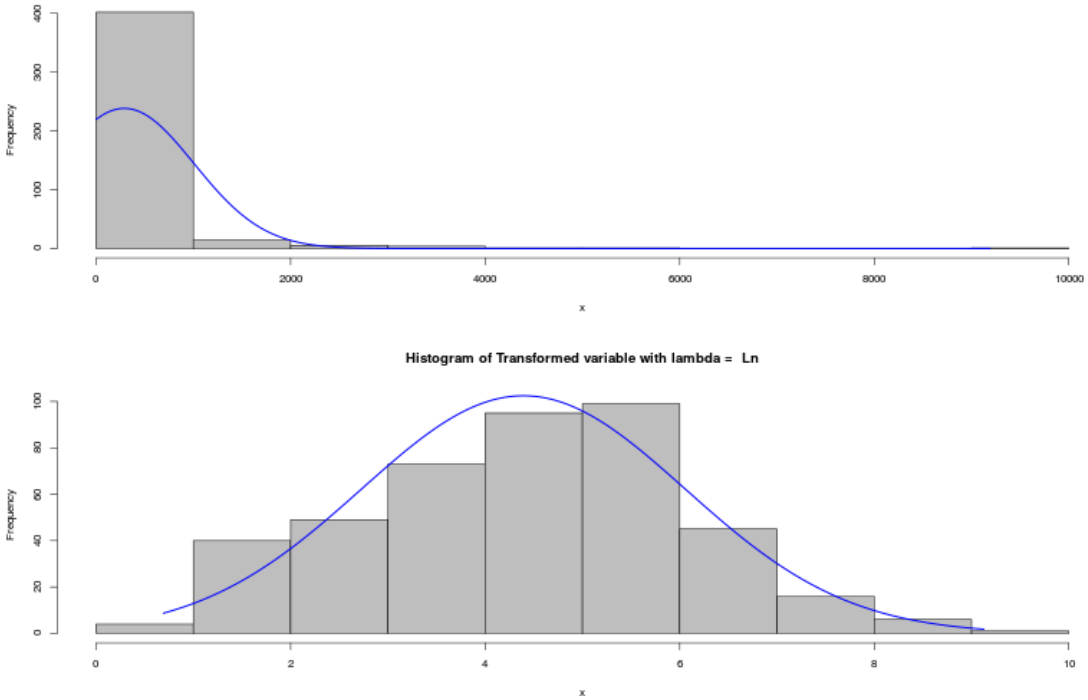
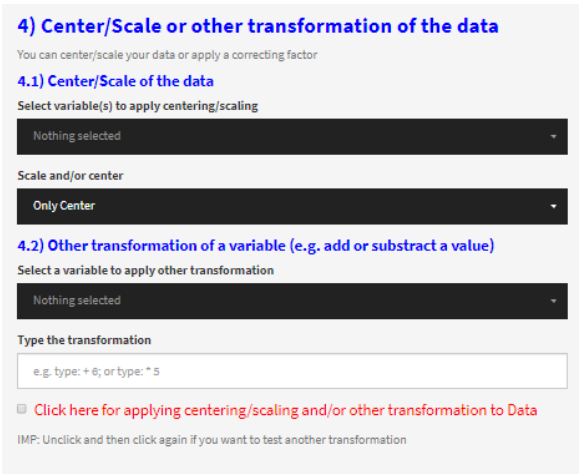


Figure 12. Screenshot of "Box-Cox transformation" sub-module and options

f) 'Scale and Center' the data



Scale, center and other transformations

Centering covariates at specific values (i.e., subtracting a specific value, such as the mean, from the observed values of a covariate) has the effect of changing the intercept in the model, so that it represents the expected value of the dependent value at a specific value of the covariate (e.g., the mean), rather than the expected value when the covariate is equal to zero (which is often outside the range of the data). In addition to changing the interpretation of the intercept in a linear model, centered covariates often reduce the amount of collinearity among the covariates with associated fixed effects in the model.

Initial data

```
'data.frame':   428 obs. of  4 variables:
 $ TAG  : Factor w/ 179 levels "86","101","155",..: 1 1 2 2 3 3 3 4 4 5 ...
 $ DBH  : int  454 454 369 369 450 450 450 413 413 486 ...
 $ FRUIT: num  3.83 6.5 1.95 2.08 6.55 ...
 $ YEAR : Factor w/ 3 levels "1","2","3": 1 2 1 2 1 2 3 1 3 1 ...
```

Transformed data

```
'data.frame':   428 obs. of  4 variables:
 $ TAG  : Factor w/ 179 levels "86","101","155",..: 1 1 2 2 3 3 3 4 4 5 ...
 $ DBH  : int  454 454 369 369 450 450 450 413 413 486 ...
 $ FRUIT: num  3.83 6.5 1.95 2.08 6.55 ...
 $ YEAR : Factor w/ 3 levels "1","2","3": 1 2 1 2 1 2 3 1 3 1 ...
```

Figure 13. Screenshot of "Scale and Center" sub-module and options

g) ‘Update dataset’: Remove/Update/Add registers

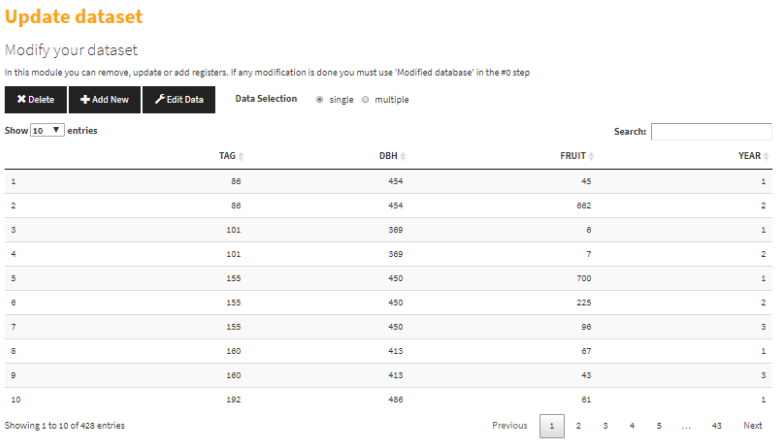


Figure 14. Screenshot of "Update dataset" sub-module

5. **'Plots'**: Bidimensional plots for exploring data

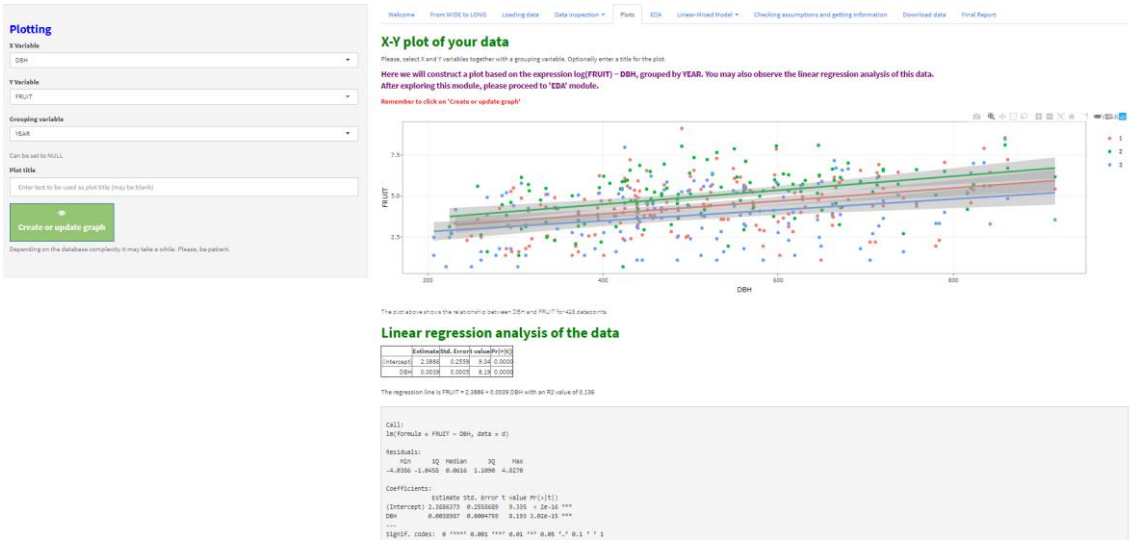


Figure 15. Screenshot of "Plots" module and options

6. **'EDA'**: Systematic EDA report

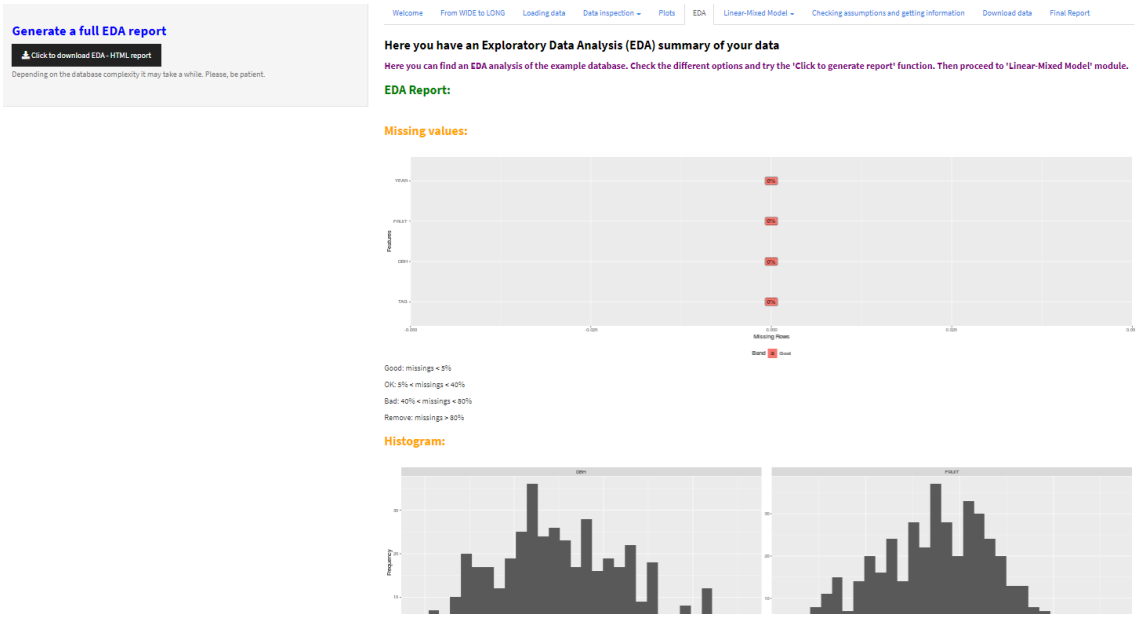


Figure 16. Screenshot of "EDA" module and options

As LMM are worth it regardless of the type of data (e.g., balanced/unbalanced; missing values, etc.), we will use it as our method for analyzing RM data. As mentioned previously, **nlme** package have been chosen due to its continuous updatability and some other aforementioned advatages compared with **lme4** package.

7. ‘Linear-Mixed Model’:

i. What is a linear-mixed model (LMM)?

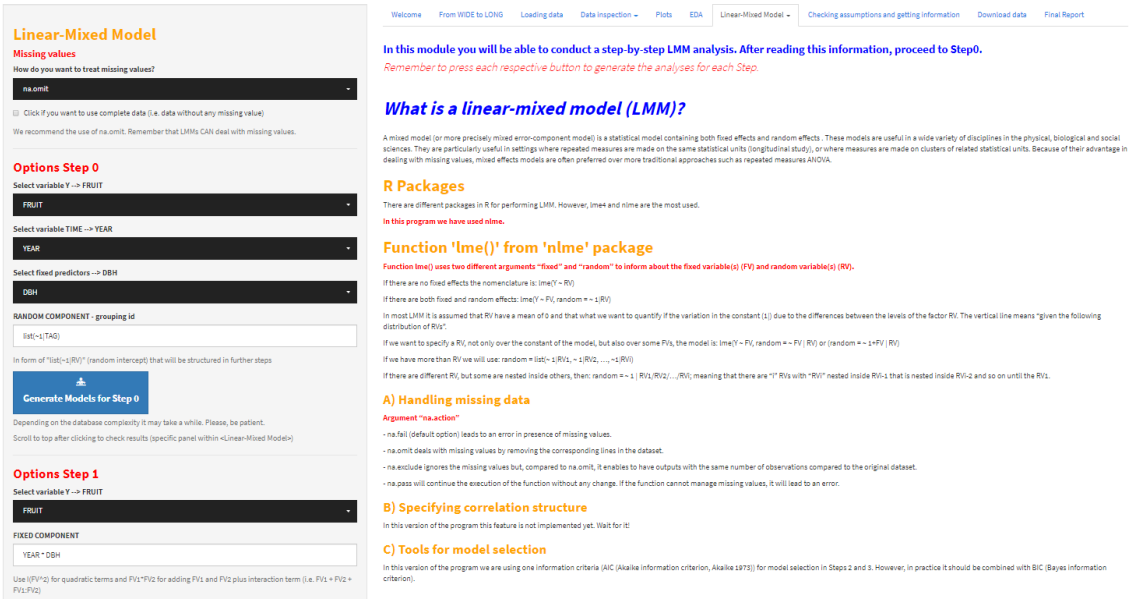


Figure 17. Screenshot of "Linear-mixed model" module and "What is a linear-mixed model (LMM)?" sub-module

ii. **'Step 0. Base models:**

1. Every predictor (fixed component) included in an independent model together with TIME variable and random component
2. Selection of FEs with P-value < 0.2 (based on Maldonado and Greenland, 1993 [43] for further steps
3. Check and solve correlations among variables

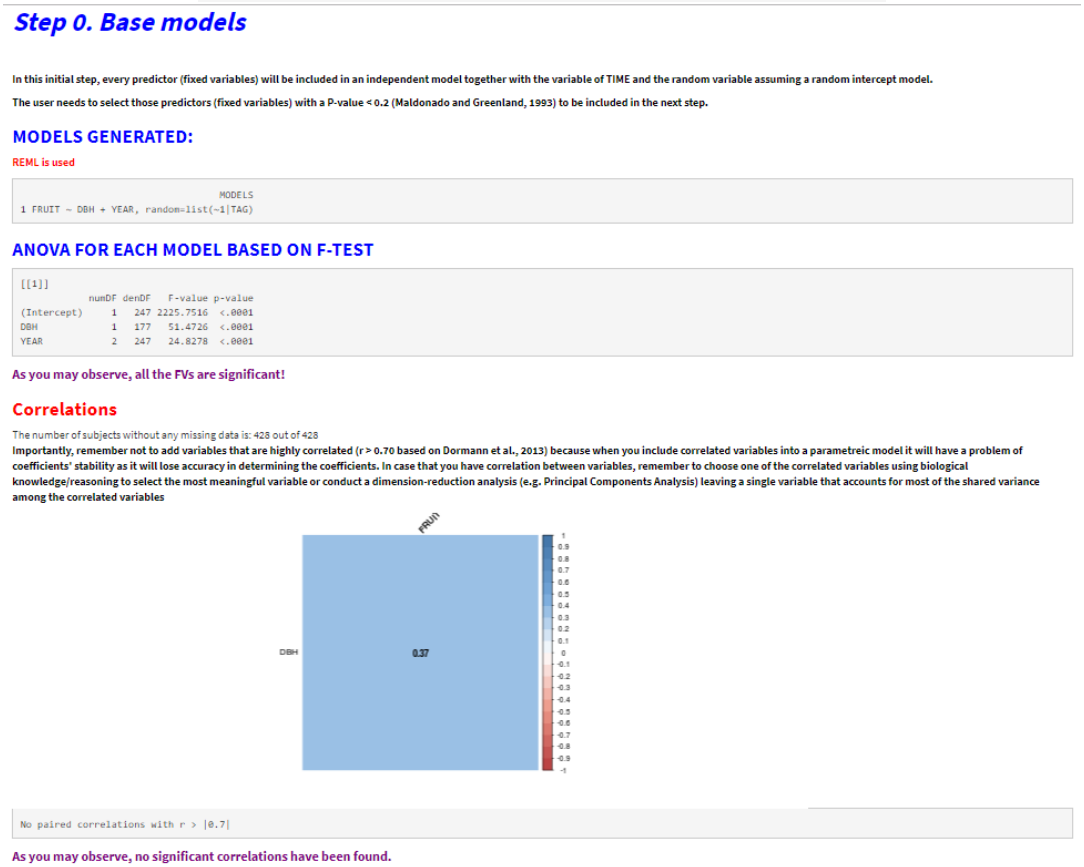


Figure 18. Screenshot of "Step 0. Base models" sub-module and options

Notably, the selection of the random and fixed components was initially defined with panels to select the variables (as happens with the Y variable). However, we then changed it to a text box to incorporate interactions and other transformations of the variables.

iii. 'Step 1' → Options Step 1. Beyond optimal model

1. Fixed component includes all "possible" variables and interaction terms. If it is not possible, select the most important
2. Output of results and ANOVA of fixed effects

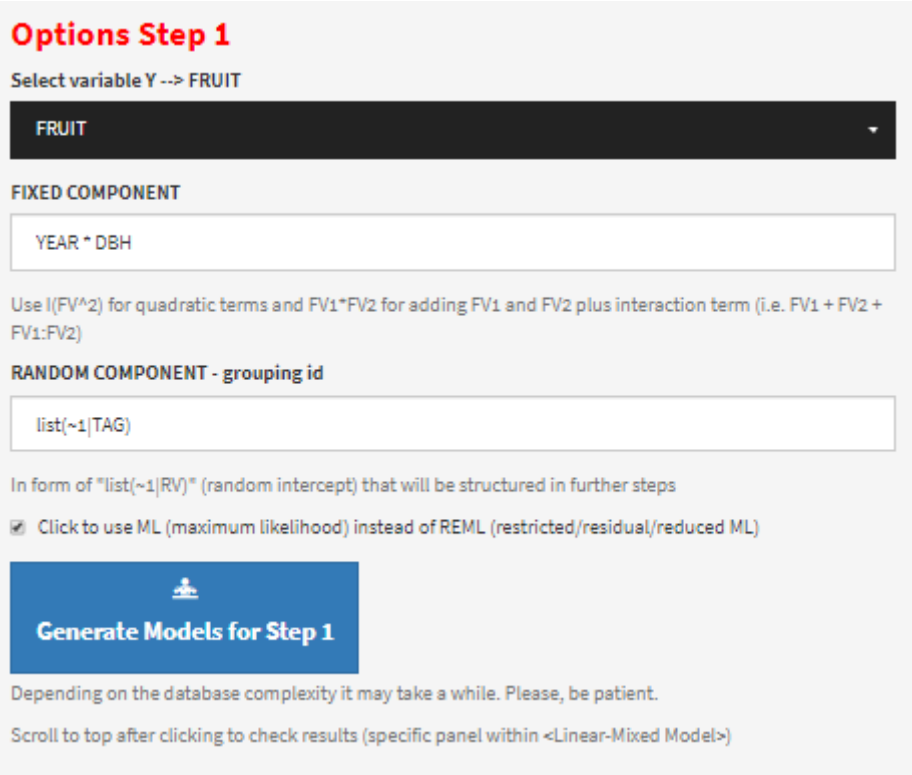


Figure 19. Screenshot of "Options for Step 1"

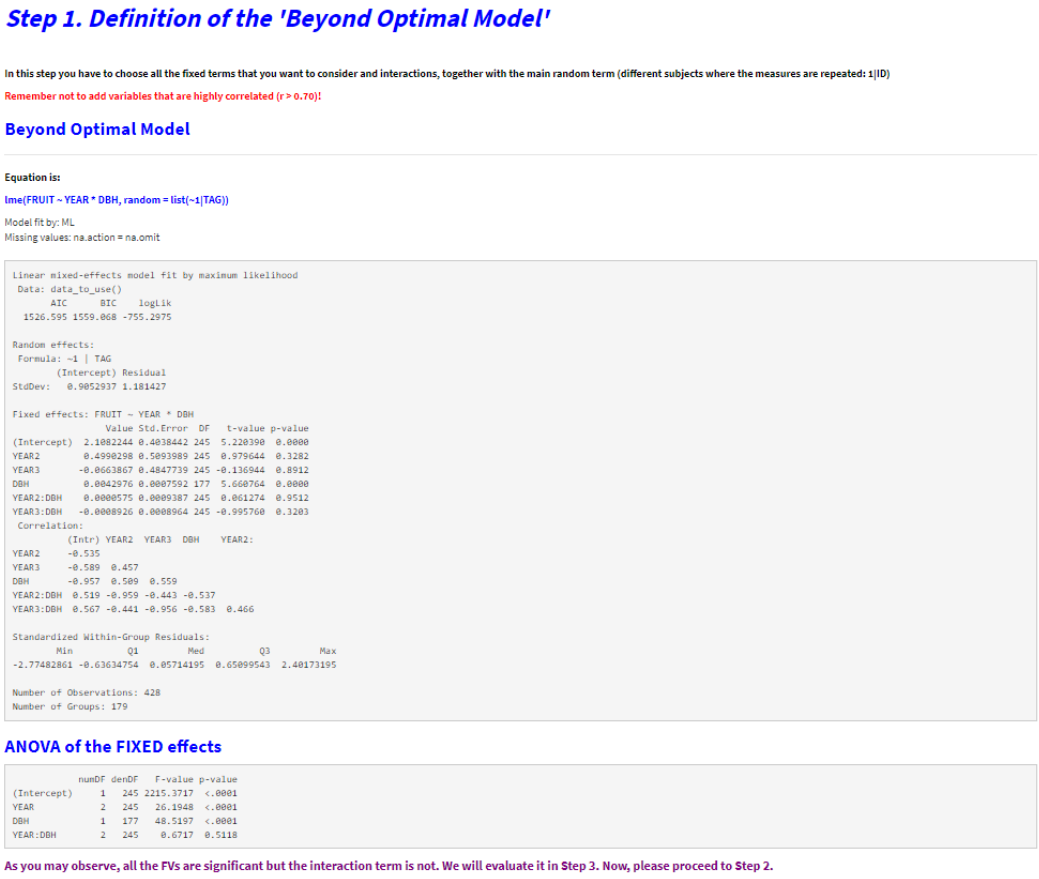


Figure 20. Screenshot of "Step 1. Definition of the model: beyond optimal model" sub-module

iv. 'Step 2' → Options Step 2. Structure of random component

1. Using beyond optimal model
2. Define models with the same fixed component but varying in the random component
3. Using REML
4. Compare models using AIC

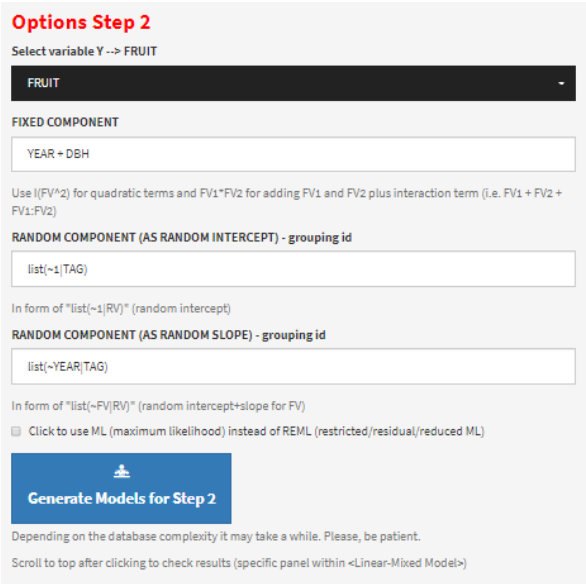


Figure 21. Screenshot of "Options for Step 2"

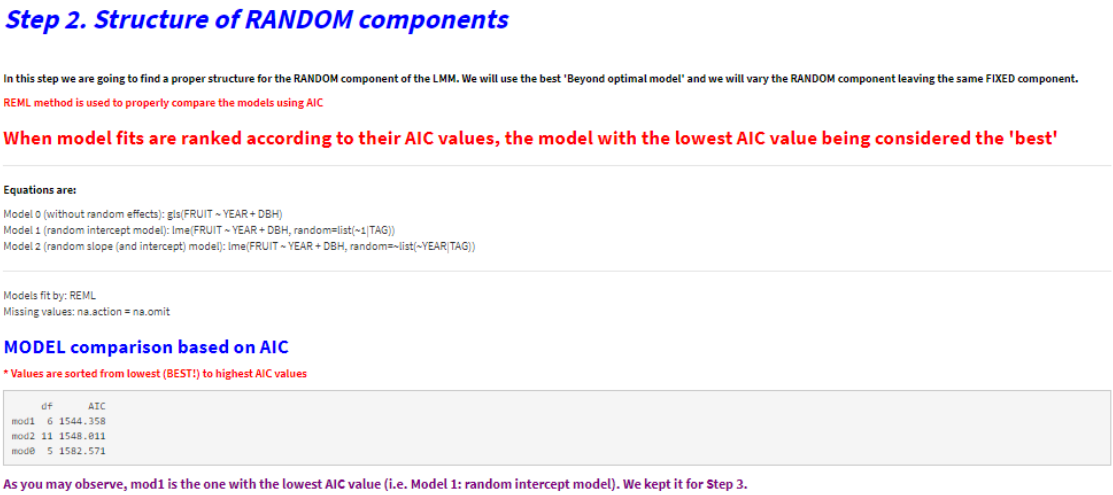


Figure 22. Screenshot of "Step 2. Structure of RANDOM component" sub-module

v. 'Step 3' → Options Step 3. Structure of the fixed component

1. Once we have optimum random component
2. Compare models with same random component but differing in the fixed component
3. Using ML
4. Compare models using AIC

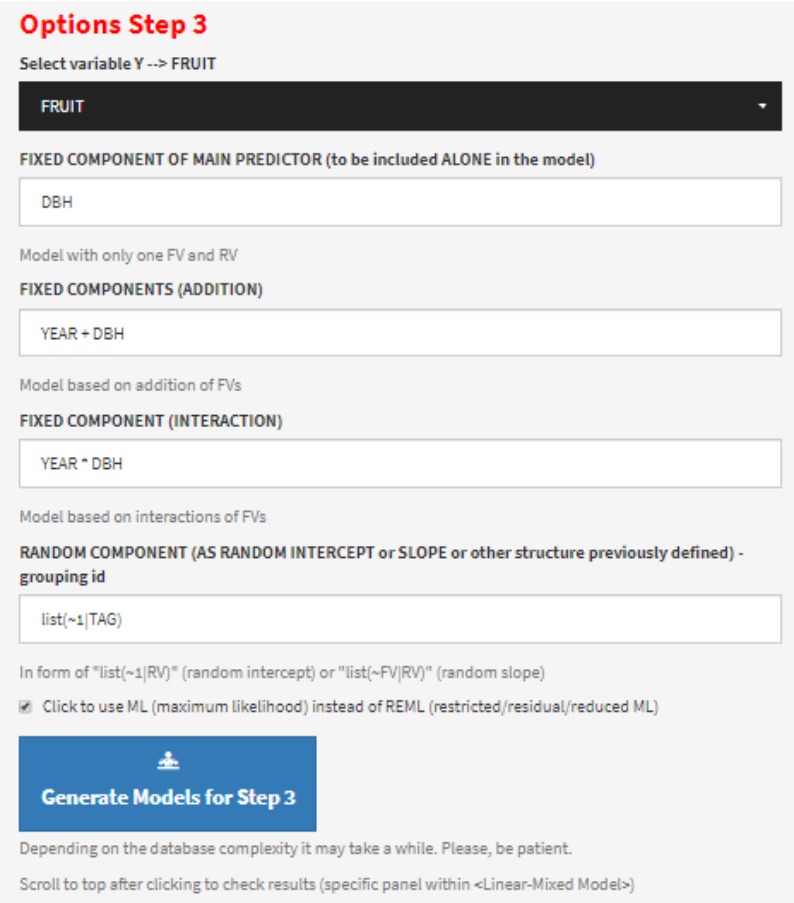


Figure 23. Screenshot of "Options for Step 3"

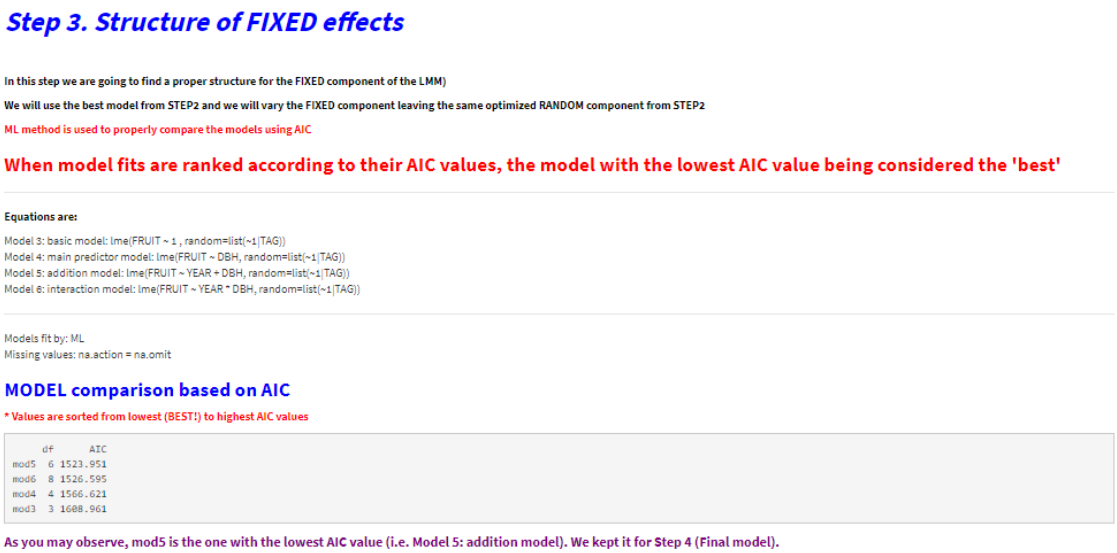


Figure 24. Screenshot of "Step 3. Structure of FIXED effects" sub-module

vi. 'Step 4' → Options Step 4. Final model

1. Using best step 3 model
2. Using REML
3. Show model details
    a. Random effects
    b. Fixed effects
    c. Overall output
    d. ANOVA based on F-value
    e. R-squared
    f. 95% CI

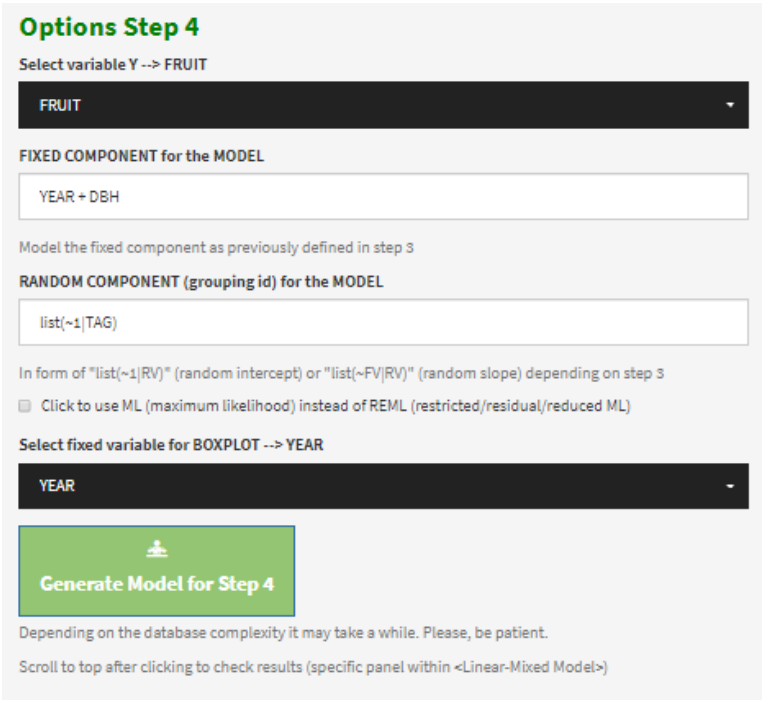


Figure 25. Screenshot of "Options for Step 4"

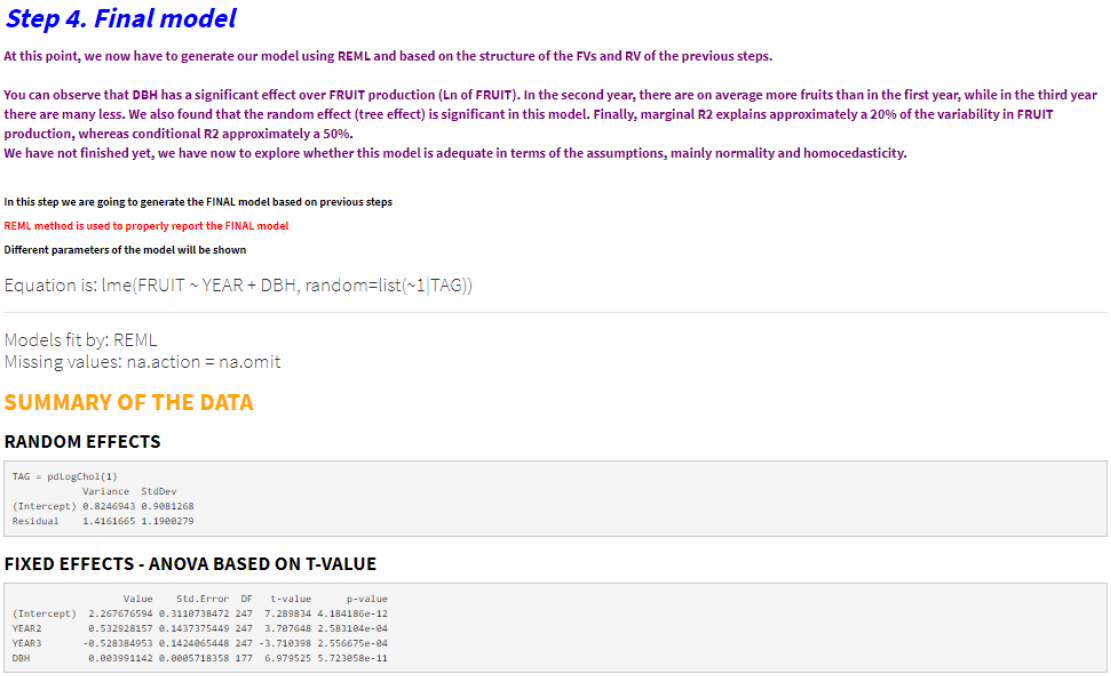


Figure 26. Screenshot of "Step 4. Final model" sub-module – Summary of the data

DBH has a significant effect over FRUIT production (specifically over the Ln of FRUIT). In the second year, there are - on average - more fruits than in the first year, while in the third year there are many less. By treating YEAR as categorical we can observe these specific deviations in year 2 and year 3 versus year 1 that could not have been observed if we had selected YEAR as numeric. In the random effects, it was found that the tree effect is significant by modifying the intercept of the model (Step 2, **Figure 22**). The fixed part of the model contained the variables YEAR and DBH, both of them significant, but not their interaction (Step 3, **Figure 24**). Note that, in case of DBH, as it is previously mentioned it is included with the same repeated value over time as its modulation in this period is scarce. The equation from the model is:

$$\mathbf{Ln\ FRUIT_{ti} = 2.27 + 0.53 \cdot YEAR2_{2i} - 0.53 \cdot YEAR3_{3i} + 0.004 \cdot DBH_{ti}\ \{FIXED\} + b_i + \varepsilon_i\ \{RANDOM\}}$$

$$\mathbf{Where\ b_i \sim N_{ni}(0, 0.82 \cdot I_{ni})\quad \varepsilon_i \sim N_{ni}(0, 1.42 \cdot I_{ni})}$$

We may observe how the individuals' (i.e., TAG) intercepts vary with an SD of 0.91, and the SD of the error not accounted for by TAGs is 1.19.

OVERALL OUTPUT FROM THE MODEL

FULL SUMMARY

```
Linear mixed-effects model fit by REML
 Data: data_to_use()
       AIC      BIC    logLik
  1544.358 1568.657 -766.1792

Random effects:
 Formula: ~1 | TAG
        (Intercept) Residual
StdDev:   0.9081268 1.190028

Fixed effects: FRUIT ~ YEAR + DBH
                 Value  Std.Error  DF   t-value p-value
(Intercept)  2.2676766 0.31107385 247  7.289834   0e+00
YEAR2        0.5329282 0.14373754 247  3.707648   3e-04
YEAR3       -0.5283850 0.14240654 247 -3.710398   3e-04
DBH          0.0039911 0.00057184 177  6.979525   0e+00
 Correlation:
      (Intr) YEAR2  YEAR3
YEAR2 -0.180
YEAR3 -0.214  0.446
DBH   -0.927 -0.020  0.009

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.78647936 -0.60881246  0.03793967  0.65215390  2.38337605

Number of Observations: 428
Number of Groups: 179
```

ANOVA BASED ON F-TEST

```
            numDF denDF   F-value p-value
(Intercept)     1   247 2225.7514  <.0001
YEAR            2   247   26.2072  <.0001
DBH             1   177   48.7138  <.0001
```

R-SQUARED (based on Nakagawa and Schielzeth (2013))

R2m (Marginal R2 (whole model)): it is the marginal R2 for a linear mixed model, meaning that it is concerned with the variance explained by the fixed factors

R2c (Conditional R2 (whole model)): it is the conditional R2 for a linear mixed model, meaning that it is concerned with the variance explained by the fixed and random factors

```
          R2m       R2c
[1,] 0.2072428 0.4989978
```

95% CONFIDENCE INTERVALS FOR THE COEFFICIENTS

```
Approximate 95% confidence intervals

 Fixed effects:
                lower         est.        upper
(Intercept)  1.654980955  2.267676594  2.880372233
YEAR2        0.249820568  0.532928157  0.816035747
YEAR3       -0.808870985 -0.528384953 -0.247898921
DBH          0.002862649  0.003991142  0.005119636
attr(,"label")
[1] "Fixed effects:"

 Random Effects:
  Level: TAG
                lower      est.    upper
sd((Intercept)) 0.7441771 0.9081268 1.108196

 Within-group standard error:
   lower     est.    upper
1.089232 1.190028 1.300152
```

Figure 27. Screenshot of "Step 4. Final model" sub-module - Overall output from the model

8. Checking assumptions and getting information (*see* M1 in the Annex 1):

i. Fitted vs standardized residuals plot
    1. We must observe no pattern
        a. Homoscedasticity?

ii. Explanatory variables vs residuals plot
    1. We must observe no pattern

iii. Histogram of residuals
    1. Normality?

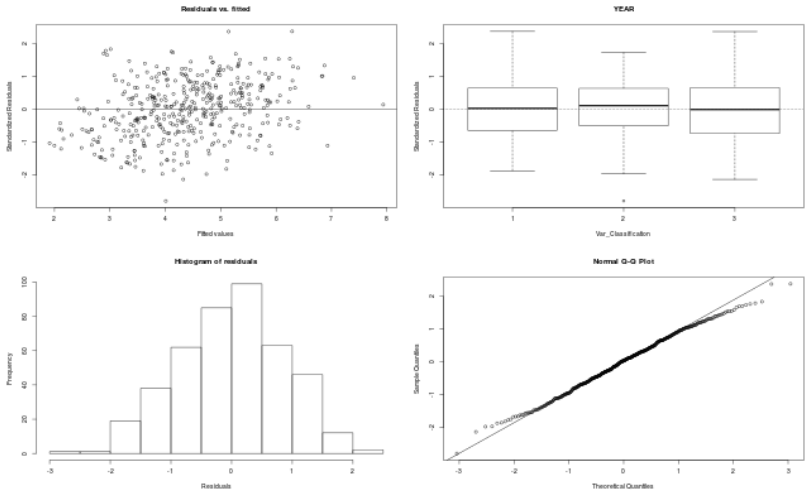
iv. QQ-plot of residuals
    1. Normality?

Assumptions from the LMM

Below you have different graphs to test that the assumptions of our LMM are met, together with general considerations about how it should behave. You may take your time to explore them.
You can observe that the model is quite adequate as the assumptions of normality, homocedasticity and linearity are met. However, we may also observe three outliers in the QQ-plot (one in the lowest theoretical quantiles and two in the highest theoretical quantiles). By using 'Download data' module, we can further explore these observations and/or individuals.

Try now changing the variable FRUIT to its initial value (without the Ln), what do you observe?.

You may now download data from the model in the 'Download data' module, or generate a report in the 'Final report' module.



1) No pattern should be seen in the plots: 'Fitted values vs Standardized residuals' (top left plot) and 'Explanatory variable vs Standardized residuals' (top right plot)

1.a) If we observe increasing or decreasing values in 'Fitted values vs Standardized residuals', it inform us about Heteroscedasticity (i.e. variance of data not approx. equal across range of predicted values) that we should correct (try box-cox transformation alternatives)

2) Histogram of residuals (bottom left) must follow a Gaussian distribution, otherwise it will inform us about non-normality behaviour

3) QQ-Plot (bottom right) must show dots aligned to the diagonal line, otherwise it will inform us about non-normality

4) Normality test: Shapiro-Wilk analysis of residuals

****** if P-value < 0.05 --> Non-normality**

```
        Shapiro-Wilk normality test

data:  Res
W = 0.99555, p-value = 0.2642
```

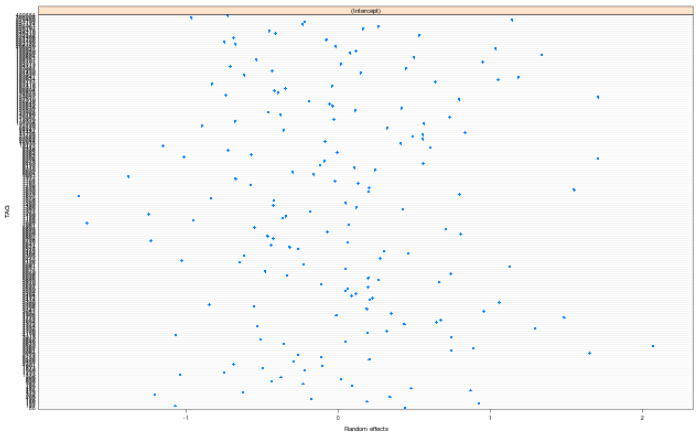
5) Multi-collinearity

```
    YEAR2    YEAR3      DBH
1.249056 1.248677 1.000783
```

Figure 28. Screenshot of "Checking assumptions and getting information" sub-module - Assumptions from the LMM.

Getting information from the model

A) Plot of random effects



B) Predicted values for the data

B1) If the structure is 1|RV then you are constructing a RANDOM INTERCEPT MODEL:

Each random variable (subject) is assigned a different intercept cause by-RV (by subject) variability is taken into account. However, the fixed effects are all the same for all RVs.

In this model, we account for baseline-differences in the dependent variable, but we assume that whatever the effect of the fixed variables, it is the same for all the RVs.

B2) If the structure is FX|RV then you are constructing a RANDOM SLOPE MODEL:

In this case RVs are not only allowed to have differing intercepts, but they are also allowed to have different slopes for the effect of FVs.

```
    (Intercept)     YEAR2     YEAR3        DBH
86     2.707033 0.5329282 -0.528385 0.003991142
101    1.194721 0.5329282 -0.528385 0.003991142
155    3.190287 0.5329282 -0.528385 0.003991142
160    2.456003 0.5329282 -0.528385 0.003991142
192    2.092015 0.5329282 -0.528385 0.003991142
203    2.606860 0.5329282 -0.528385 0.003991142
352    1.059062 0.5329282 -0.528385 0.003991142
415    1.639040 0.5329282 -0.528385 0.003991142
486    3.136129 0.5329282 -0.528385 0.003991142
487    2.745737 0.5329282 -0.528385 0.003991142
```

Figure 29. Screenshot of "Checking assumptions and getting information" module - Getting information from the model

From the plots and data presented, we may affirm that the assumptions for the model are met: there is no heteroscedasticity (i.e., random pattern in the “predicted values” versus “standardized residuals”) and we observe normality in histogram and QQ-plot. Relevantly, we may observe some outliers by checking the QQ-plot. One outlier by considering the lowest theoretical quantiles and two outliers by considering the highest theoretical quantiles. We may take advantage of the ‘**Download data**’ module to get further information from these observations and subjects.

9. **Download data**
    a. Download a .CSV with data used in SISSREM together with fitted values and/or residuals.

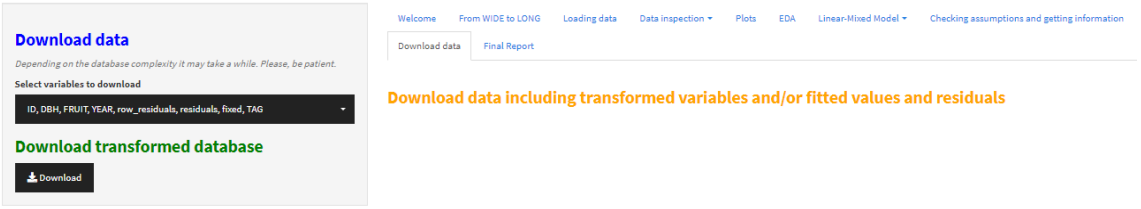


Figure 30. Screenshot of "Download data" module

.CSV File appearance

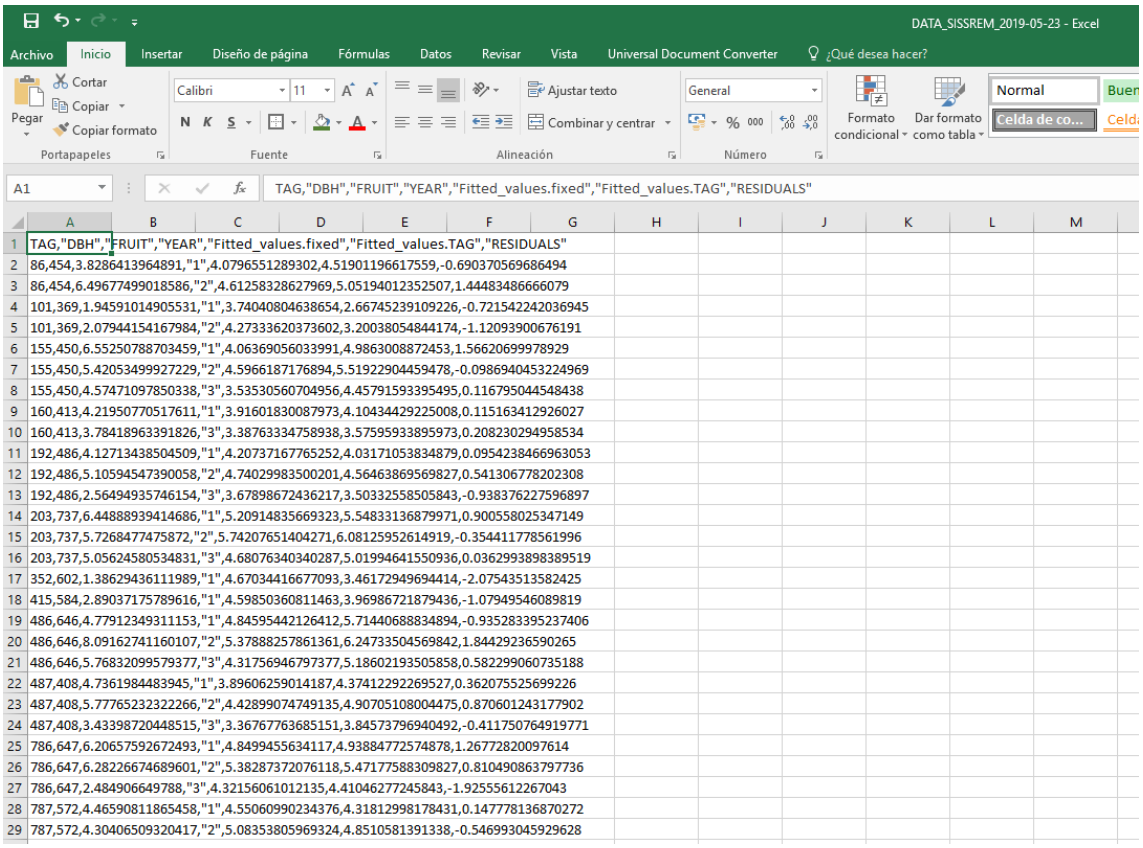


Figure 31. Screenshot of ".CSV file appearance"

10. **Final Report** → Generate a report including the main information from the modules

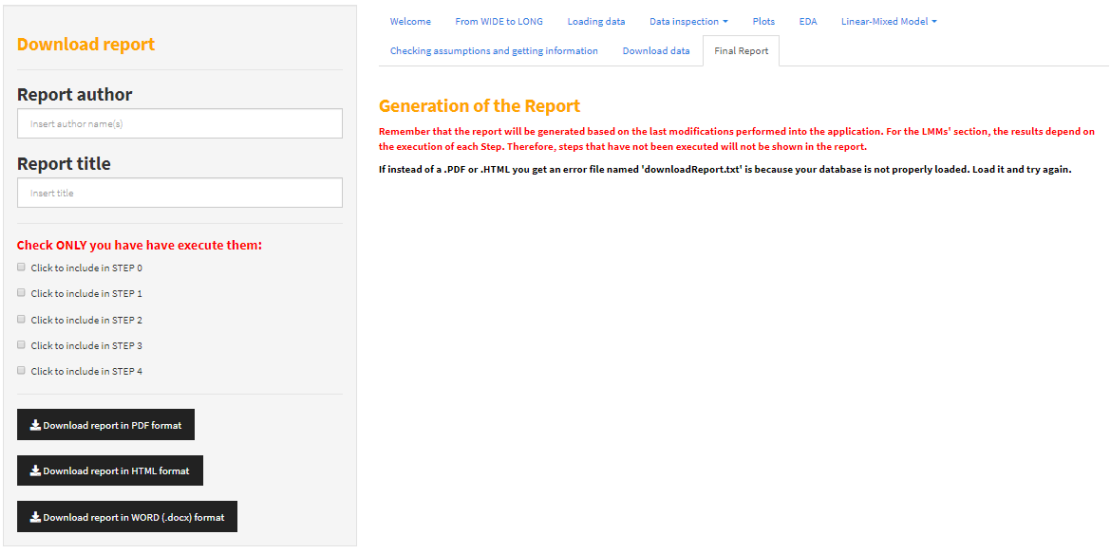
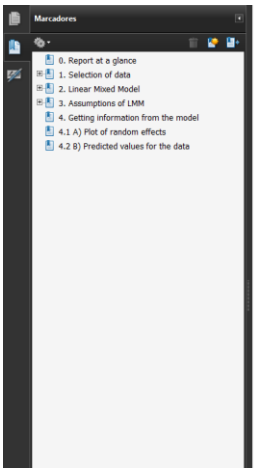


Figure 32. Screenshot of "Final Report" module

A) PDF: <u>Annex 4</u> includes the execution of the .Rmd file using the Example database.



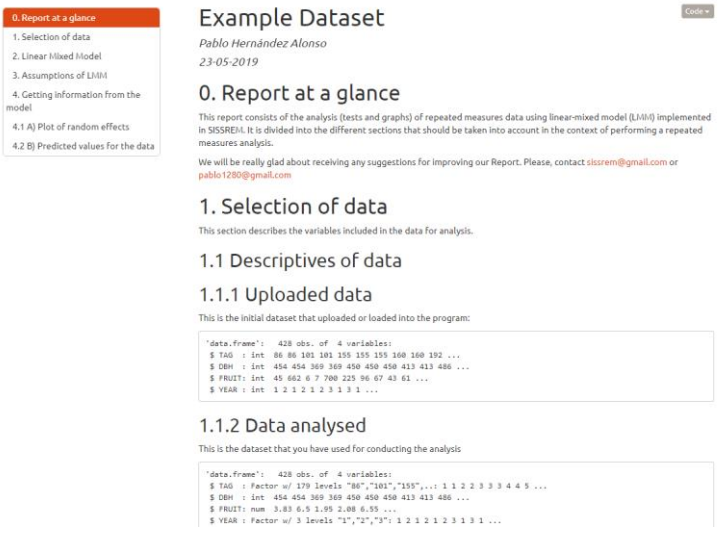

Example Dataset

Pablo Hernández Alonso

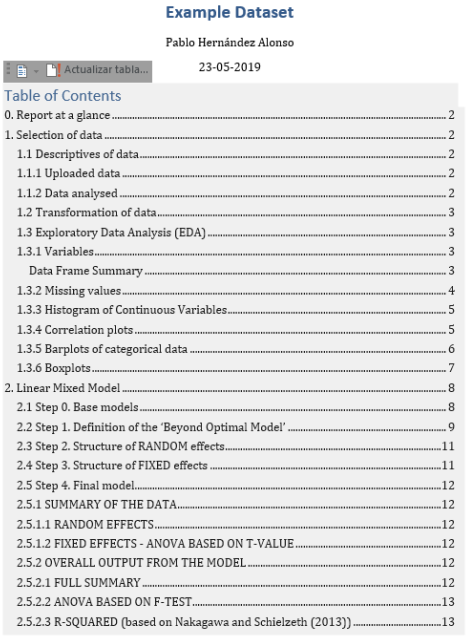
23-05-2019

Contents

B) HTML



C) Microsoft Word (.DOCX)



Example Dataset

Pablo Hernández Alonso

Actualizar tabla...

23-05-2019

Table of Contents

Figure 33. Screenshot of the report in PDF, HTML and Microsoft Word (.DOCX) format

5. Conclusions

5.1 Overall conclusions of the project

In this FMP we have developed a Shiny-R application - entitled SISSREM - to make the procedure of conducting a repeated measures analysis easier, interactive, systematic and supervised. We have included modules for data visualization, data treatment, conducting the LMM analysis, obtaining/downloading data and generating a full report of the process. Therefore, we have fulfilled the initial objectives of the project. The next step, which is beyond the limits of this project - is to divulgate it to test whether it is useful for the scientific community.

5.2 Critic analysis of the overall procedure (planning and methodology)

In order to start with this FMP, I had to learn all the theoretical background behind the analysis of RMs data, and specifically, the use of LMMs for these kind of designs. A great effort was done to explain the main concepts and to combine the relevant information obtained from different references. In this FMP, we have included the analysis – as well as supervised comments – of an example database with a longitudinal design.

The initial Gantt chart of the project have been followed. However, some changes from the main milestones' delivery have been performed. In both PEC2 and PEC3 we included some deviations from the initial procedure. When designing the code in R (**M2**), it was vital to adapt it to the Shiny code (**M3**). This was something that my project supervisor commented from the beginning of the design. Therefore, in PEC2, the R-Shiny code was delivered, so that the deliverables **M2** and **M3** were put together. That code (app.R) contains the structure of the application and all the modules that are intended to be delivered. Moreover, the video of how to use the application (**M5**) was recorded after the delivery of PEC3.

Relevantly, we sent an abstract of our project (**Annex 5**) to the *"XVII Conferencia Española y VII Encuentro Iberoamericano de Biometría"* that will be held in Valencia during June. It was accepted as a 15-minutes oral presentation. This fact will give an external assessment of our work, together with plenty of inputs and suggestions for SISSREM further development.

5.3 Strengths and limitations

One of the main limitations of our application is the need of an internet connection. Moreover, up to now, the language is limited to English. LMM analysis in SISSREM only allows the execution of an uncorrelated matrix for residuals because we have not implemented its modulation. Moreover, we have not implemented the inclusion of generalized LMM.
Unfortunately, due to the free-hosting characteristics, the program gets disconnected if it is inactive for a while (>15 minutes). Also due to this fact, the maximum RAM memory

(1 Gb) may be not enough for large datasets. Moreover, we have not included the option to save the process and load afterwards.

As strengths, it is an easy to follow application which may be acceseed by different devices (e.g., computer, phone, tablet/iPad). There are plenty of modules for exploring the data, filter it, and different EDA analyses to have a "trailer" clip of our data before starting the analysis. This implies to the user that they do not need other intermediate programs between obtaining data and analyzing it with SISSREM. We gave the possibility to download generated data which may be use for conducting alternative or complementary analysis such as GEE. The example database is systematically commented to train the user on the overall procedure. Our application makes easy the extraction of data from it by allowing to generate a report in three different formats. One of the main strengths of our application is that it may – and it is going to – be updated.

5.4 Future perspectives

The main perspectives for the future lie on the removal of many of the limitations. We plan on updating this application by adding more functions such as:

1. Options for saving the status of the session and posibility to load it.
2. A function for labelling factor variables once loaded. E.g., sex (1 and 2; male or female) or treatment (A or B; Control or intervention).
3. To perform some comments about the variables. E.g., whether ID should be set as factor of numeric.
4. To allow the user to use different center/scale for each variable.
5. In Plot module, show grouping options in the equation.
6. To vary the structure of the residuals matrix in *lme*.
7. To combine AIC with BIC in Steps 2 and 3.
8. An option to generate the Final Report based on a very little input information from the user.
9. To add modules to conduct generalized estimating equations (GEE) and update the code to allow the execution of generalized LMMs.
10. To add the equation for the LMM including FIXED and RANDOM component.

6. Glossary

AIC, Akaike information criterion

ANOVA, analysis of variance

BIC, Bayes information criterion

EBLUP, empirical best linear unbiased predictors

EDA, exploratory data analysis

FAQs, frequently asked questions

FE, fixed effect

FF, fixed factor

GEE, generalized estimating equations

GLM, generalized linear model

IDE, integrated development environment

LMM, linear-mixed model

MAR, missing at random

ML, maximum likelihood

NID, normally and independently distributed

REML, restricted/residual/reduced maximum likelihood

RM, repeated measures

RE, random effect

RF, random factor

UI, user interface

7. References

1. Rstudio. Shiny. https://shiny.rstudio.com/. Accessed 20 Mar 2019.

2. Cox DR. Principles of Statistical Inference. Cambridge: Cambridge University Press; 2006. doi:10.1017/CBO9780511813559.

3. Laberge Y. Advising on Research Methods: A Consultant's Companion. J Appl Stat. 2011;38:2991–2991. doi:10.1080/02664763.2011.559375.

4. Sthle L, Wold S. Analysis of variance (ANOVA). Chemom Intell Lab Syst. 1989;6:259–72. doi:10.1016/0169-7439(89)80095-4.

5. Judd CM, McClelland GH, Ryan CS. Repeated-Measures ANOVA. In: Data Analysis. Third Edition. | New York : Routledge, 2017. | Revised edition: Routledge; 2017. p. 260–91. doi:10.4324/9781315744131-11.

6. Brady T. West, Kathleen B. Welch ATG. Linear Mixed Models: A Practical Guide Using Statistical Software. Boca Raton, FL: Taylor and Francis/CRC Press; 2014.

7. Healy K. Book Review: An R and S-PLUS Companion to Applied Regression. Sociol Methods Res. 2005;34:137–40. doi:10.1177/0049124105277200.

8. Wooldrige, Jeffrey M. Front matter. In: Neurology Secrets. Elsevier; 2011. p. i–ii. doi:10.1016/B978-0-323-05712-7.00031-3.

9. Raudenbush SW, Bryk AS. Hierarchical linear models and experimental design. In: Applied analysis of variance in behavioral science. 2nd edition. Newbury Park, CA; 2002.

10. Theory and Computational Methods for Nonlinear Mixed-Effects Models. In: Mixed-Effects Models in S and S-PLUS. New York: Springer-Verlag; 2006. p. 305–36. doi:10.1007/0-387-22747-4_7.

11. Ray BK, Liu Z, Ravishanker N. Dynamic Reliability Models for Software Using Time-Dependent Covariates. Technometrics. 2006;48:1–10. doi:10.1198/004017005000000292.

12. Verbeke G, Molenberghs G. Generalized Linear Mixed Models—Overview. In: The SAGE Handbook of Multilevel Modeling. London (United Kingdom): SAGE Publications Ltd; 2013. p. 127–40. doi:10.4135/9781446247600.n8.

13. Kenyon JR. Statistical Methods for the Analysis of Repeated Measurements. Technometrics. 2003;45:99–100. doi:10.1198/tech.2003.s14.

14. McLean RA, Sanders WL, Stroup WW. A Unified Approach to Mixed Linear Models. Am Stat. 1991;45:54–64. doi:10.1080/00031305.1991.10475767.

15. Ware JH. Linear Models for the Analysis of Longitudinal Studies. Am Stat. 1985;39:95–101. doi:10.1080/00031305.1985.10479402.

16. Fitzmaurice GM, Ravichandran C. A Primer in Longitudinal Data Analysis. Circulation. 2008;118:2005–10. doi:10.1161/CIRCULATIONAHA.107.714618.

17. Verbeke G, Molenberghs G. Selection Models. In: Linear Mixed Models for Longitudinal Data. New York, NY: Springer New York; 2000. p. 231–73. doi:10.1007/b98969.

18. Little RJA, Rubin DB. Statistical Analysis with Missing Data. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2002. doi:10.1002/9781119013563.

19. Goldstein H, Bryk AS, Raudenbush SW. Hierarchical Linear Models: Applications and Data Analysis Methods. J Am Stat Assoc. 1993;88:386. doi:10.2307/2290750.

20. Hilbe JM. Data Analysis Using Regression and Multilevel/Hierarchical Models. J Stat Softw. 2015;30 Book Review 3. doi:10.18637/jss.v030.b03.

21. Pinheiro JCJC, Bates DM. Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model. J Comput Graph Stat. 1995;4:12–35. doi:10.1080/10618600.1995.10474663.

22. Cayuela L. Modelos lineales mixtos (LMM) y modelos lineales genera-lizados mixtos (GLMM) en R (versión 3.1). Spain; 2018.

23. Vuong QH. Likelihood Ratio Tests for Model Selection and Non-Nested Hypotheses. Econometrica. 1989;57:307. doi:10.2307/1912557.

24. Kuha J. AIC and BIC. Sociol Methods Res. 2004;33:188–229. doi:10.1177/0049124103262065.

25. Jiang J, Lahiri P. Mixed model prediction and small area estimation. Test. 2006;15:1–59. doi:10.1007/BF02595419.

26. Carey VJ, Wang Y-G. Mixed-Effects Models in S and S-Plus. J Am Stat Assoc. 2001;96:1135–6. doi:10.1198/jasa.2001.s411.

27. Skrondal A, Rabe-Hesketh S. Multilevel and Related Models for Longitudinal Data. In: Handbook of Multilevel Analysis. New York, NY: Springer New York; 2008. p. 275–99. doi:10.1007/978-0-387-73186-5_7.

28. M. Gad A, B. El Kholy R. Generalized Linear Mixed Models for Longitudinal Data. Int J Probab Stat. 2012;1:41–7. doi:10.5923/j.ijps.20120103.03.

29. Datta GS, Lahiri P. A unified measure of uncertainty of estimated best linear unbiased predictors in small area estimation problems. Stat Sin. 2000;10:613–28.

30. Dormann CF, Elith J, Bacher S, Buchmann C, Carl G, Carré G, et al. Collinearity: A review of methods to deal with it and a simulation study evaluating their performance. Ecography (Cop). 2013;36:027–46. doi:10.1111/j.1600-0587.2012.07348.x.

31. Six Differences Between Repeated Measures ANOVA and Linear Mixed Models - The Analysis Factor. https://www.theanalysisfactor.com/six-differences-between-repeated-measures-anova-and-linear-mixed-models/. Accessed 28 Apr 2019.

32. Friedman JH, Tukey JW. A Projection Pursuit Algorithm for Exploratory Data Analysis. IEEE Trans Comput. 1974;C-23:881–90. doi:10.1109/T-C.1974.224051.

33. Ueda H, Niida K, Usuki T, Hirauchi K, Meschede M, Miura R, et al. Seafloor Geology of the Basement Serpentinite Body in the Ohmachi Seamount (Izu-Bonin Arc) as Exhumed Parts of a Subduction Zone Within the Philippine Sea. In: Modern Approaches in Solid Earth Sciences. 2011. p. 97–128. doi:10.1007/978-90-481-8885-7_5.

34. Bell A, Jones K. Explaining Fixed Effects: Random Effects Modeling of Time-Series Cross-Sectional and Panel Data. Polit Sci Res Methods. 2015;3:133–53. doi:10.1017/psrm.2014.7.

35. R Development Core Team R. R: A Language and Environment for Statistical Computing. R Found Stat Comput. 2011;1 2.11.1:409. doi:10.1007/978-3-540-74686-7.

36. Guide of longitudinal analysis based on lme and lmer. https://rpsychologist.com/r-guide-longitudinal-lme-lmer. Accessed 2 Apr 2019.

37. Kuznetsova A, Brockhoff PB, Christensen RHB. lmerTest Package: Tests in Linear Mixed Effects Models . J Stat Softw. 2017.

38. R-core. nlme - Linear and Nonlinear Mixed Effects Models. https://www.rdocumentation.org/packages/nlme/versions/3.1-140. Accessed 23 May 2019.

39. Rstudio. Shinyapps.io. https://www.shinyapps.io/. Accessed 22 Feb 2019.

40. Rstudio. Rstudio cloud. https://rstudio.cloud. Accessed 22 Feb 2019.

41. Rstudio. R Markdown. https://rmarkdown.rstudio.com/. Accessed 24 May 2019.

42. Krewinkel A, Winkler R. Formatting Open Science: agilely creating multiple document formats for academic manuscripts with Pandoc Scholar. PeerJ Comput Sci. 2017;3:e112.

43. Maldonado G, Greenland S. Simulation Study of Confounder-Selection Strategies. Am J Epidemiol. 1993;138:923–36. doi:10.1093/oxfordjournals.aje.a116813.

8. Acknowledgements

Cuando decidimos estudiar algo podemos hacerlo por múltiples motivos. Este máster ha significado para mi un proceso de aprendizaje “a mi modo”. Sin tener que correr para acabar rápido, con tiempo para asimilar los conceptos y para aplicarlos a mi trabajo. Agradezco profundamente a todos los profesores, en especial a mi tutora, Núria Pérez Álvarez, por toda su ayuda y apoyo durante el proceso del TFM.

Gracias a mis padres y a mi hermano, por su infinito amor. A mi familia canaria y a la andaluza. Gracias a todos mis amigos, los que están cerca o lejos, porque la distancia no supone nada. Simona, Lucía, Denis, Núria, gracias por ser y estar. Juan, gracias por aparecer en mi vida.

9. Annex

Annex 1. Milestone 1 (M1) - Systematic procedure for repeated-measures analysis

Annex 2. M2-M3 - Code of the Shiny app (SISSREM; to be compiled as **app.R**)

Annex 3. M4 - Rmarkdown report file (to be compiled as **report.Rmd** in combination with **app.R**)

Annex 4. Final report in .PDF format using the example database

Annex 5. Congress abstract – Oral communication

9.1 Annex 1. Milestone 1 (M1) - Systematic procedure for repeated-measures analysis

This procedure is adapted to SISSREM

11. **'Welcome'**
12. **'From WIDE to LONG'** → Module for changing from wide to long data
13. **'Loading data'** –> Loading database or selecting an example database
14. **'Data inspection'** Exploring the database
    - h) **'Data summary'**
      - i. Select variables to EXPLORE
      - ii. Change between variable types
        1. Are the variables in the correct variable type (i.e. numeric, factor, etc.)? → Change them accordingly
      - iii. Transform dependent variable
      - iv. Center/scale or other transformation of the data
    - i) **'Full data view'**
    - j) **'Missing values'**
    - k) **'Correlation between variables'** → **collinearity of continuous variables**
    - l) **'Box-cox transformation' analysis**
    - m) **'Scale and Center' the data**
    - n) **'Update dataset': Remove/Update/Add registers**
15. **'Plots'**: Bidimensional plots for exploring data
16. **'EDA (explanatory data analysis)'**: Systematic EDA report

As **LMMs** are worth it regardless of the type of data (balanced/unbalanced; missing values, etc.), we will use it as our method for analyzing RM data:

- **nlme** R package has been chosen because it has been more widely used in the context of RM analysis compared to **lme4**, access to different objects from the results is easier and allow the definition of matrix structures

17. **'Linear-Mixed Model' (*see* also sub-Annex 1):**
    - i. **'What is a linear-mixed model (LMM)?'**
      1. Brief definition
      2. R Packages
      3. Function *lme()* from **nlme**
      4. Handling missing data, specifying correlation structure and test for model selection
    - ii. **'Step 0'** → Options Step 0. Base models
      1. Using REML
      2. Time variable is combined with every putative fixed predictor under a random intercept structure.
      3. Select significant fixed predictors (P<0.2; Maldonado and Greenland, 1993)
    - iii. **'Step 1' → Options Step 1. Beyond optimal model**
      1. Using ML
      2. Fixed component includes all "possible" variables and interaction terms. If it is not possible, select the most important
      3. Output of results and ANOVA of fixed effects
    - iv. **'Step 2' → Options Step 2. Structure of random component**
      1. Using beyond optimal model
      2. Define models with the same fixed component but varying in the random component
      3. Using REML
      4. Compare models using AIC

v. **'Step 3' → Options Step 3. Structure of the fixed component**
    1. Once we have optimum random component (Step 2)
    2. Compare models with same random component but differing in the fixed component
    3. Using ML
    4. Compare models using AIC
vi. **'Step 4' → Options Step 4. Final model**
    1. Using best step 3 model
    2. Using REML
    3. Show model details
        a. **Random effects**
        b. **Fixed effects**
        c. **Overall output**
        d. **ANOVA based on F-value**
        e. **R-squared**
        f. **95% CI**

18. **Checking assumptions and getting information (*see* also sub-Annex 2):**
    i. Fitted vs residuals plot
        1. We have to observe no pattern
            a. Homoscedasticity?
    ii. Explanatory variables vs residuals plot
        1. We have to observe no pattern
    iii. Histogram and QQ-plot of residuals
        1. Normality?
19. **Download data**
    a. Download a .CSV with data used in SISSREM together with fitted values and/or residuals.
20. **Final Report** → Rmarkdown to PDF, HTML or Microsoft Word
    a. Data description and EDA
    b. LMMs
    c. Diagnosis of assumptions

Sub-Annex 1 – Creating our hypothesis for LMM analysis:

Define fixed variables (FV)
- **a.** Do we need any interaction term?

Define random variables (RV)
- **a. random = ~ 1 | subject;** Indicates that each subject will have its own intercept
- **b. random = ~ FV1 | subject;** Indicates that each subject will have its own intercept and its own slope for FV1
- **c. random = ~ 1 | FV1 / subject;** Indicates that each subject-within-FV1 unit will have its own intercept

Sub-Annex 2 – Full explanation of assumptions

Residual plot: Linearity + Homoscedasticity + Normality

- **a. Linearity of residuals:** fitted vs residual values. If it is not OK:
    1. Important fixed effect missing?
    2. Log-transform response variable
    3. Log-transform fixed effects or add var and var^2 (quad terms
    4. If there are "stripes" → categorical data? = consider logistic models

- **b. Homoscedasticity (variance of data approx. equal across range of predicted values):** fitted vs residual values. If it is not OK:
    1. **Transform data (e.g. log)**

c. **Normality of residuals (*paradox. least important assumption):** fitted vs residual values.
    1. Histogram (bell-shaped) or Q-Q plot (line) of residuals → normal distribution

d. **Absence of collinearity**
    ii. **Collinearity produces misinterpretation of the model.** If it is not OK:
        1. Check correlation → if r>0.7 between two variables (Dormann *et al.*, 2013) you have to select one (e.g. by biological plausibility) or perform dimension reduction such as principal component analysis.

e. **Absence of influential data points.**
    1. Leave-one out diagnostics → DFbeta. If it is not OK:

    If there are influential data points → double run the analysis

f. **Independence of observations:**
    iii. Most important → properly define random variables

9.2 Annex 2. M2-M3 - Code of the Shiny app (SISSREM; to be compiled as **app.R**)

```
##########
# SISSREM: Shiny Interactive, Supervised and Systematic report from REpeated Measures data
# Freely hosted in: https://sissrem.shinyapps.io/SISSREM_v1/
##########

###########
# Libraries to be used
###########

library(car)
library(corrplot)
library(DataExplorer)
library(dichromat)
library(dplyr)
library(DT)
library(editData)
library(ggplot2)
library(htmltools)
library(influence.ME)
library(lme4)
library(lmerTest)
library(MASS)
library(MuMIn)
library(naniar)
library(nlme)
library(plotly)
library(questionr)
library(R.devices)
library(rcompanion)
library(readr)
library(shiny)
library(shinyjs)
library(shinythemes)
library(shinyWidgets)
library(sjPlot)
library(stringr)
library(summarytools)
library(tools)
library(usdm) # error?
library(vembedr)
library(xtable)

source("helpers.R") # Load all the code needed to show feedback on a button click
# favicon (web icon): https://stackoverflow.com/questions/30096187/favicon-in-shiny


#########################################################################################################
###################################
############################################################ UI ###################################
###################################
#########################################################################################################
###################################

ui <- fluidPage(theme = shinytheme("cosmo"),
                # favicon
                tags$head(tags$link(rel="shortcut icon", href="https://cdn3.iconfinder.com/data/
icons/finance-152/64/32-512.png")),

                # App title
                titlePanel("SISSREM v1.0 (last updated on 05/06/2019)",
                           windowTitle = "SISSREM v1.0: Shiny Interactive, Supervised and System
atic report from REpeated Measures data"),


                (pageWithSidebar(
                  headerPanel(em("Shiny Interactive, Supervised and Systematic report from REpea
ted Measures data", style = "color:blue; font-weight:bold")),
```

```
                  sidebarPanel(


                  #### USED in "From WIDE to LONG"


                    conditionalPanel(condition="input.tabselected==66",

                                     h3("1) Load a .CSV database", style = "color:blue; font-we
ight:bold"),
                                     fileInput('file_to_transform', 'Choose CSV File',
                                     accept=c('text/csv', 'text/comma-separated-values,text/pla
in', '.csv')),

                                     checkboxInput('header_file_to_transform', 'Click if your d
ata contains a header row', TRUE),
                                     br(),

                                     radioButtons('sep_file_to_transform', 'Which is the file s
eparator?',
                                                                                  c(Semico
lon=';',
                                                                                    Comma=
',',
                                                                                    Tab='\
t'),
                                                                                  ';'),
                                     br(),

                                     radioButtons("quote_file_to_transform", "Quotes type:", ch
oices = c(None = "",

"Double Quote" = '"',

"Single Quote" = "'"), selected = '"'),

                                     hr(),

                                     h3("2) Now, transform your data", style = "color:blue; fon
t-weight:bold"),
                                     textInput("name_var_change_long", "Write the name of the g
rouping variable (time by default)", "time"),
                                     pickerInput(inputId = "vars_that_vary_long",
                                                 label = "Select variables that vary with the g
rouping variable",
                                                  "",
                                                  options = list(`actions-box` = TRUE),multiple
= T),

                                     pickerInput(inputId = "identificator_long",
                                     label = "Select the identificator variable (subjects, regi
sters)",
                                     "",
                                     options = list(`actions-box` = TRUE),multiple = T),

                                     textInput("separator_long", "Separator of grouped variable
s", ""),
                                     actionButton("end_definition_long", "Create LONG data", ic
on = icon("file-export"), style = "color:red; font-weight:bold"),
                                     hr(),

                                     pickerInput(inputId = "sele_variable_to_download_transform
ed",
                                                  label = "Select variables to download",
                                                  "",
                                                   options = list(`actions-box` = TRUE),multipl
e = T),

                                     h3("3) Download transformed database", style = "color:blue
; font-weight:bold"),
                                     downloadButton('download_transformed'),
```

```
                                    br()
                                    ),


                    #### USED in "Loading data"    style = "color:lightblue; font-weight:bold; f
ont-size:24px"

                                conditionalPanel(condition="input.tabselected==0",

                                    h3(p("Please, select one of the two following options and
then click on the button", style = "color:green; font-weight:bold")),
                                    h1(actionButton("load_option", "LOAD FILE", icon = icon("f
ile-import"), style = "color:red; font-weight:bold; font-size:24px", width = "100%")),
                                    tags$hr(),
                                    # bookmarkButton(), # Save a session; not implemented

                                    h3("Option 1: select an example database", style = "color:
red; font-weight:bold"),
                                    checkboxInput('example_database1', 'Example Database (comp
lete cases)', FALSE),
                                    checkboxInput('example_database2', 'Example Database with
missing values', FALSE),
                                    tags$hr(),

                                    h3("Option 2: upload your own .CSV file", style = "color:r
ed; font-weight:bold"),

                                    fileInput('file1', 'Choose CSV File',
                                              accept=c('text/csv',
                                               'text/comma-separated-values,text/plain',
                                               '.csv')),

                                    checkboxInput('header', '2.1 Click if your data contains a
header row', TRUE),
                                    br(),
                                    radioButtons('sep', '2.2 Which is the file separator?',
                                                 c(Semicolon=';',Comma=',',Tab='\t'),';'),
                                    br(),

                                    radioButtons("quote", "2.3 Quotes type:",choices = c(None
= "",
                                                                                         "Double
Quote" = '"',
                                                                                         "Single
Quote" = "'"),
                                                                                        selected
= '"'),

                                    br()
                                    ),



                    #### USED in "Data inspection"


                                                        conditionalPanel(condition="input.tabs
elected==1",

                                    # Input: Select number of rows to display
                                    radioButtons("disp", "Display data in the subsection 'Full
data view'",
                                    choices = c("First registers of data" = "head", "Whole dat
aset" = "all"),
                                                                                        selected
= "head"),
                                    h3(radioButtons('database_to_select', '0) Database to sele
ct',
                                                                         c("Initial - RAW up
loaded version"='data()',
                                                                        "Modified - If ANY c
hange was made"='modified_data()'),
```

```
                                                                    'data()'), style =
"color:red; font-weight:bold"),
                                    helpText('If you plan on modifying anything below, please,
select - Modified -'),
                                    tags$hr(),

                                    h3("1) Select variables to EXPLORE", style = "color:blue;
font-weight:bold"),
                                    helpText('If your database is big, maybe you want to get r
id of unnecessary variables.'),
                                    pickerInput(inputId = "sele_variable",
                                                label = "", "", options = list(`actions-box`
= TRUE),multiple = T),

                                    tags$hr(),
                                    h3("2) Change between variable TYPES", style = "color:blue
; font-weight:bold"),
                                    helpText('Set categorical variables; define numeric variab
les, etc.'),

                                    pickerInput(inputId = "var_factor",
                                                label = "Select variables to change into FAC
TOR",
                                                    "",
                                                options = list(`actions-box` = TRUE),mul
tiple = T),


                                    pickerInput(inputId = "var_numeric",
                                                 label = "Select variables to change into NU
MERIC",
                                                   "", options = list(`actions-box` = TRUE),
multiple = T),

                                    pickerInput(inputId = "var_character",
                                                label = "Select variables to change into CHA
RACTER",
                                                   "", options = list(`actions-box` = TRUE)
,multiple = T),

                                     tags$hr(),

                                   h3("3) TRANSFORM 'Y' variable", style = "color:blue; font-we
ight:bold"),
                                   helpText('Apply Box-Cox to determine the best transformation
'),

                                    pickerInput(inputId = "var_box_cox",
                                                label = "Select variable to apply a tranformati
on",
                                                "",
                                                options = list(`actions-box` = TRUE),multiple =
F),

                                    pickerInput(inputId = "type_transformation",
                                                label = "Choose type of transformation",
                                                "", choices = c("Box-cox", "Log2", "Log10", "Ln
", "Square root", "Inverse"),
                                                options = list(`actions-box` = TRUE),multiple =
F),


                                    h4(checkboxInput('trans_box_buttom', 'Click here for applyi
ng transformation to Data', FALSE), style = "color:red; font-weight:bold"),

                                    tags$hr(),

                                    h3("4) Center/Scale or other transformation of the data", s
tyle = "color:blue; font-weight:bold"),
                                    helpText('You can center/scale your data or apply a correct
ing factor'),
```

```
                                    h4("4.1) Center/Scale of the data", style = "color:blue; fo
nt-weight:bold"),

                                    pickerInput(inputId = "var_center_scale",
                                                label = "Select variable(s) to apply centering/
scaling",
                                                "",
                                                options = list(`actions-box` = TRUE),multiple =
T),

                                    pickerInput(inputId = "center_scale",
                                                label = "Scale and/or center ",
                                                "", choices = c("Only Center", "Only Scale", "C
enter and Scaling (i.e. auto-scaling)",
                                                                "Range Scaling", "Pareto Scalin
g", "Vast Scaling", "Level Scaling"),
                                                options = list(`actions-box` = TRUE),multiple =
F),

                                    h4("4.2) Other transformation of a variable (e.g. add or su
bstract a value)", style = "color:blue; font-weight:bold"),

                                    pickerInput(inputId = "var_other_transform",
                                                label = "Select a variable to apply transformat
ion",
                                                "",
                                                options = list(`actions-box` = TRUE),multiple =
F),

                                    textInput(inputId = "type_other_transform",
                                              label = "Type the transformation",
                                              placeholder = "e.g. type: + 6; or type: * 5"),

                                    h4(checkboxInput('trans_cent_scale_box_buttom', 'Click here
for applying centering/scaling and/or other transformation to Data', FALSE), style = "color:red;
font-weight:bold"),
                                    helpText('IMP: Unclick and then click again if you want to
test another transformation'),

                                    br()


                    ),



                    #### USED in "Plots"


                    conditionalPanel(condition="input.tabselected==2",
                                    h3("Plotting", style = "color:blue; font-weight:bold"),

                                    selectInput('xcol', 'X Variable', ""),
                                    selectInput('ycol', 'Y Variable', "", selected = ""),
                                    selectInput('zcol', 'Grouping variable', "", selected = "")
,
                                    helpText('Can be set to NULL'),

                                    textInput(inputId = "plot_title",
                                              label = "Plot title",
                                              placeholder = "Enter text to be used as plot titl
e (may be blank)"),


                                    actionButton("generate_graph", h4("Create or update graph",
style = "color:white; font-weight:bold"), icon = icon("eye"),
                                                 style="color: #fff; background-color: #93c572
; border-color: #2e6da4"),


                                    helpText('Depending on the database complexity it may take
a while. Please, be patient.'),
```

```
                                    br()

                    ),


                    #### USED in "EDA"
                    conditionalPanel(condition="input.tabselected==3",

                                    h3("Generate a full EDA report", style = "color:blue; font-
weight:bold"),
                                    #h2(checkboxInput('report_EDA', 'Click to generate report',
FALSE), style = "color:red; font-weight:bold"),
                                    downloadButton('download_EDA_html', 'Click to download EDA
- HTML report'),
                                    helpText('Depending on the database complexity it may take
a while. Please, be patient.'),

                                    br()
                    ),


                    #### USED in "Linear-Mixed Model"

                    conditionalPanel(condition="input.tabselected==4",
                                    h2("Linear-Mixed Model", style = "color:orange; font-weight
:bold"),


                                    h4("Missing values", style = "color:red; font-weight:bold")
,

                                    pickerInput(inputId = "type_missing",
                                                label = "How do you want to treat missing value
s?",
                                                "na.omit", choices = c("na.fail", "na.omit", "n
a.exclude", "na.pass"),
                                                options = list(`actions-box` = TRUE),multiple =
F),
                                    checkboxInput('use_complete_data', 'Click if you want to us
e complete data (i.e. data without any missing value)', FALSE),
                                    helpText('We recommend the use of na.omit. Remember that LM
Ms CAN deal with missing values.'),

                                    hr(),



                                    ### Step 0

                                    h3("Options Step 0", style = "color:red; font-weight:bold")
,

                                    pickerInput(inputId = "var_y_step0",
                                                label = "Select variable Y",
                                                "FRUIT",
                                                options = list(`actions-box` = TRUE),multiple =
F),



                                    pickerInput(inputId = "var_time_step0",
                                                label = "Select variable TIME",
                                                "",
                                                options = list(`actions-box` = TRUE),multiple =
F),


                                    pickerInput(inputId = "var_fixed_step0",
                                                label = "Select fixed predictors",
                                                "",
```

```
                                                options = list(`actions-box` = TRUE), multiple
= T),

                                    textInput(inputId = "text_random_step0",
                                              label = "RANDOM COMPONENT - grouping id",
                                              placeholder = "list(~1|RV)", value = "list(~1|RV)
"),
                                    helpText('In form of "list(~1|RV)" (random intercept) that
will be structured in further steps'),


                                    actionButton("Step0_BUTTON", h4("Generate Models for Step 0
", style = "color:white; font-weight:bold"), icon = icon("diagnoses"),
                                                 style="color: #fff; background-color: #337ab7;
border-color: #2e6da4"),
                                    br(),
                                    helpText('Depending on the database complexity it may take
a while. Please, be patient.'),
                                    helpText('Scroll to top after clicking to check results (sp
ecific panel within <Linear-Mixed Model>)'),


                                    #actionButton("Step0_BUTTON", "Generate Models for Step0",
icon = icon("diagnoses")),


                                    hr(),


                                    ### Step 1

                                    h3("Options Step 1", style = "color:red; font-weight:bold")
,

                                    pickerInput(inputId = "var_y",
                                                label = "Select variable Y",
                                                "",
                                                options = list(`actions-box` = TRUE),multiple =
F),


                                    textInput(inputId = "text_fixed_step1",
                                              label = "FIXED COMPONENT",
                                              placeholder = "Enter terms for FIXED component"),
                                    helpText('Use I(FV^2) for quadratic terms and FV1*FV2 for a
dding FV1 and FV2 plus interaction term (i.e. FV1 + FV2 + FV1:FV2)'),


                                    textInput(inputId = "text_random_step1",
                                              label = "RANDOM COMPONENT - grouping id",
                                              placeholder = "list(~1|RV)", value = "list(~1|RV)
"),
                                    helpText('In form of "list(~1|RV)" (random intercept) that
will be structured in further steps'),

                                    checkboxInput('REML', 'Click to use ML (maximum likelihood)
instead of REML (restricted/residual/reduced ML)', TRUE),

                                    actionButton("Step1_BUTTON", h4("Generate Models for Step 1
", style = "color:white; font-weight:bold"), icon = icon("diagnoses"),
                                                 style="color: #fff; background-color: #337ab7;
border-color: #2e6da4"),
                                    helpText('Depending on the database complexity it may take
a while. Please, be patient.'),
                                    helpText('Scroll to top after clicking to check results (sp
ecific panel within <Linear-Mixed Model>)'),


                                    hr(),


                                    ### Step 2
```

```
                                    h3("Options Step 2", style = "color:red; font-weight:bold")
,

                                    pickerInput(inputId = "var_y_step2",
                                                label = "Select variable Y",
                                                "",
                                                options = list(`actions-box` = TRUE),multiple =
F),

                                    textInput(inputId = "text_fixed_step2",
                                              label = "FIXED COMPONENT",
                                              placeholder = "Enter terms for FIXED component"),
                                    helpText('Use I(FV^2) for quadratic terms and FV1*FV2 for a
dding FV1 and FV2 plus interaction term (i.e. FV1 + FV2 + FV1:FV2)'),

                                    textInput(inputId = "text_random_INTERCEPT_step2",
                                              label = "RANDOM COMPONENT (AS RANDOM INTERCEPT) -
grouping id",
                                              placeholder = "list(~1|RV)", value = "list(~1|RV)
"),
                                    helpText('In form of "list(~1|RV)" (random intercept)'),

                                    textInput(inputId = "text_random_SLOPE_step2",
                                              label = "RANDOM COMPONENT (AS RANDOM SLOPE) - gro
uping id",
                                              placeholder = "list(~FV|RV)", value = "list(~FV|R

V)"),
                                    helpText('In form of "list(~FV|RV)" (random intercept+slope
for FV)'),

                                    checkboxInput('REML_step2', 'Click to use ML (maximum likel
ihood) instead of REML (restricted/residual/reduced ML)', FALSE),


                                    actionButton("Step2_BUTTON", h4("Generate Models for Step 2
", style = "color:white; font-weight:bold"), icon = icon("diagnoses"),
                                                 style="color: #fff; background-color: #337ab7;
border-color: #2e6da4"),
                                    helpText('Depending on the database complexity it may take
a while. Please, be patient.'),
                                    helpText('Scroll to top after clicking to check results (sp
ecific panel within <Linear-Mixed Model>)'),




                                    #### STEP 3

                                    h3("Options Step 3", style = "color:red; font-weight:bold")
,

                                    pickerInput(inputId = "var_y_step3",
                                                label = "Select variable Y",
                                                "",
                                                options = list(`actions-box` = TRUE),multiple =
F),


                                    textInput(inputId = "text_fixed_ALONE_step3",
                                              label = "FIXED COMPONENT OF MAIN PREDICTOR (to be
included ALONE in the model)",
                                              placeholder = "Enter terms for FIXED component"),
                                    helpText('Model with only one FV and RV'),

                                    textInput(inputId = "text_fixed_sum_step3",
                                              label = "FIXED COMPONENTS (ADDITION)",
                                              placeholder = "Enter terms for FIXED component"),
                                    helpText('Model based on addition of FVs'),


                                    textInput(inputId = "text_fixed_interactions_step3",
                                              label = "FIXED COMPONENT (INTERACTION)",
```

```
                                          placeholder = "Enter terms for FIXED component"),
                                helpText('Model based on interactions of FVs'),

                                textInput(inputId = "text_random_step3",
                                          label = "RANDOM COMPONENT (AS RANDOM INTERCEPT or
SLOPE or other structure previously defined) - grouping id",
                                          placeholder = "list(~|)", value = "list(~|)"),
                                helpText('In form of "list(~1|RV)" (random intercept) or "l
ist(~FV|RV)" (random slope)'),

                                checkboxInput('REML_step3', 'Click to use ML (maximum likel
ihood) instead of REML (restricted/residual/reduced ML)', TRUE),


                                actionButton("Step3_BUTTON", h4("Generate Models for Step 3
", style = "color:white; font-weight:bold"), icon = icon("diagnoses"),
                                             style="color: #fff; background-color: #337ab7;
border-color: #2e6da4"),
                                helpText('Depending on the database complexity it may take
a while. Please, be patient.'),
                                helpText('Scroll to top after clicking to check results (sp
ecific panel within <Linear-Mixed Model>)'),




                                #### STEP 4

                                h3("Options Step 4", style = "color:green; font-weight:bold
"),

                                pickerInput(inputId = "var_y_step4",
                                            label = "Select variable Y",
                                            "",
                                            options = list(`actions-box` = TRUE),multiple =
F),

                                textInput(inputId = "text_fixed_step4",
                                          label = "FIXED COMPONENT for the MODEL",
                                          placeholder = "Enter terms for FIXED component"),
                                helpText('Model the fixed component as previously defined i
n step 3'),

                                textInput(inputId = "text_random_step4",
                                          label = "RANDOM COMPONENT (grouping id) for the M
ODEL",
                                          placeholder = "list(~|)", value = "list(~|)"),
                                helpText('In form of "list(~1|RV)" (random intercept) or "l
ist(~FV|RV)" (random slope) depending on step 3'),

                                checkboxInput('REML_step4', 'Click to use ML (maximum likel
ihood) instead of REML (restricted/residual/reduced ML)', FALSE),

                                pickerInput(inputId = "var_fixed_for_boxplot",
                                            label = "Select fixed variable for BOXPLOT",
                                            "",
                                            options = list(`actions-box` = TRUE),multiple =
F),


                                actionButton("Step4_BUTTON", h4("Generate Model for Step 4"
, style = "color:white; font-weight:bold"), icon = icon("diagnoses"),
                                             style="color: #fff; background-color: #93c572
; border-color: #2e6da4"),
                                helpText('Depending on the database complexity it may take
a while. Please, be patient.'),
                                helpText('Scroll to top after clicking to check results (sp
ecific panel within <Linear-Mixed Model>)'),



                                br()
```

```
                    ),



                    #### USED in "Download data"

                    conditionalPanel(condition="input.tabselected==18",


                                     h3("Download data", style = "color:blue; font-weight:bold")
,
                                     helpText(em('Depending on the database complexity it may ta
ke a while. Please, be patient.')),

                                     pickerInput(inputId = "variables_LMM_download",
                                                 label = "Select variables to download",
                                                 "",
                                                 options = list(`actions-box` = TRUE),multiple =
T),

                                     h3("Download transformed database", style = "color:green; f
ont-weight:bold"),

                                     downloadButton('download_data_LMM'),

                                     br()

                    ),



                    #### USED in "Final report"

                    conditionalPanel(condition="input.tabselected==5",

                                     h3("Download report", style = "color:orange; font-weight:bo
ld"),
                                     hr(),
                                     #pickerInput(inputId = "vars_report",
                                    #             label = "Select modules to include in report",
                                    #             "",
                                    #             options = list(`actions-box` = TRUE),multiple
= F),

                                     # h3(radioButtons('format', h3('(*) Document format', style
= "color:black; font-weight:bold"), c('PDF', 'HTML', 'Word'),
                                     #              inline = TRUE)),

                                    h3(textInput(inputId = "report_author",
                                                 label = "Report author",
                                                 placeholder = "Insert author name(s)")),

                                    h3(textInput(inputId = "report_name",
                                              label = "Report title",
                                              placeholder = "Insert title")),
                                    hr(),
                                    h4(p("Check ONLY if you have have executed them:", style = "
color:red; font-weight:bold")),
                                    checkboxInput('clic_use_step0', 'Click to include in STEP 0'
, FALSE),
                                    checkboxInput('clic_use_step1', 'Click to include in STEP 1'
, FALSE),
                                    checkboxInput('clic_use_step2', 'Click to include in STEP 2'
, FALSE),
                                    checkboxInput('clic_use_step3', 'Click to include in STEP 3'
, FALSE),
                                    checkboxInput('clic_use_step4', 'Click to include in STEP 4'
, FALSE),
                                    hr(),
```

```
                                    downloadButton('downloadReport_PDF', 'Download report in PDF
format'),
                                    br(),br(),
                                    downloadButton('downloadReport_HTML', 'Download report in HT
ML format'),
                                    br(),br(),
                                    downloadButton('downloadReport_WORD', 'Download report in WO
RD (.docx) format'),


                                     br()
                    )

                  ),









############## ############## ############## ############## ############## MAIN PANEL###########
### ############## ############## ############## ############## ##############

mainPanel(
          tabsetPanel(

           # WELCOME
           tabPanel("Welcome",
br(),

h1("Welcome to SISSREM", style = "color:green; font-weight:bold"),

em("Please, read the following information before starting to use this Shiny app. The following
video is a
   summary of the purpose of SISSREM.",
style = "color:red; font-weight:bold"),
br(),br(),

#embed_url("https://www.youtube.com/watch?v=DpHEpyitzUQ"),
embed_youtube("DpHEpyitzUQ"),
br(),                                br(),

                               p("This Shiny app, named ",span("SISSREM", style = "color:green;
font-weight:bold"),
                                 ", is the product of a Final Master Project from Bioinformatics
and Biostatistics Master (UOC)."),
                               p("The main objective of", span("SISSREM", style = "color:green;
font-weight:bold"), "is to generate a systematic and supervised report from repeated measures da
ta.
                                  However, we also offer the users to perform the step-by-step
repeated-measures analysis based on an
                                   example database."),

                               p("The main statistical approach of this program is based on line
ar-mixed models (LMM). LMMs are flexible tools for the analysis of
                                  repeated-measures/longitudinal data extending the capabilitie
s of standard linear models by allowing, for example, the use of
                               unbalanced data (i.e. unequal numbers of observations in each exp
erimental treatment), missing data and time-varying covariates."),

                               em("Please, consider the following:",
                                  style = "color:red; font-weight:bold"),
                               br(),
                               p("- This app is intended to be used by analists with low-medium
skills on statistics."),
```

```
                                p("- Specifying correlation structure in LMM in this app is not a
vailable up to now."),
                                p("- Due to RAM memory issues of the app, you have to test whethe
r your data is too large or not for the analysis. "),


                                p(span("SISSREM", style = "color:green; font-weight:bold"), "cons
ists of different panels starting with", span("Loading a data", style = "color:lightblue; font-w
eight:bold; font-size:24px"), "where
                                    the user can select an example database or load their own dat
a.
                                      The rest of the panels consist of:"),
                                p("--- Data Inspection ---", style = "color:lightblue; font-weigh
t:bold; font-size:24px"),
                                p("--- Plots ---", style = "color:lightblue; font-weight:bold; fo
nt-size:24px"),
                                p("--- EDA ---", style = "color:lightblue; font-weight:bold; font
-size:24px"),
                                p("--- Linear-Mixed Model ---", style = "color:lightblue; font-we
ight:bold; font-size:24px"),
                                p("--- Checking assumptions and getting information ---", style =
"color:lightblue; font-weight:bold; font-size:24px"),
                                p("--- Download data ---", style = "color:lightblue; font-weight:
bold; font-size:24px"),
                                p("--- Final Report ---", style = "color:lightblue; font-weight:b
old; font-size:24px"),
                                p("In the panel", span("'From WIDE to LONG'", style = "color:ligh
tblue; font-weight:bold; font-size:24px"),
                                  "the user can transform their data from a wide into a long form
at."),

                                #    p("p creates a paragraph of text."),
                                #    p("A new p() command starts a new paragraph. Supply a style
attribute to change the format of the entire paragraph.", style = "font-family: 'times'; font-si
16pt"),
                                #    strong("strong() makes bold text."),
                                #    em("em() creates italicized (i.e, emphasized) text."),
                                #    br(),
                                #    code("code displays your text similar to computer code"),
                                #    div("div creates segments of text with a similar style. This
division of text is all blue because I passed the argument 'style = color:blue' to div", style =
"color:blue"),
                                #    br(),
                                #    p("span does the same thing as div, but it works with",
                                #    span("groups of words", style = "color:blue"),
                                #    "that appear inside a paragraph.")

                                br()
                        ),

# From WIDE to LONG

                        tabPanel("From WIDE to LONG", value=66,
                                br(),

                                h3(p("Sometimes data is in a WIDE format (each row is an id).
                                       Here you can transform your data into a LONG (each row i
s an id for each repeated measure) format and download the database to use it in this Shiny App.
")),

                                h2("Initial data", style = "color:blue; font-weight:bold"),
                                tableOutput('starting_data'),

                                h2("Result", style = "color:red; font-weight:bold"),
                                tableOutput('prueba_show_long_data'),
                                br()
                        ),

# Loading data
                        tabPanel("Loading data", value=0,
                                br(),

                                h3(p("If this is your first time using LMMs, we encourage you to
```

```
try the example database and follow the instructions",
                                    style = "color:orange; font-weight:bold; font-size:24px")),
                                h3(p("IMPORTANT: Data must use DOT as decimal separator AND must
have the FIRST column as ID for subjects (or other first level identifying variable)",
                                    style = "color:red; font-weight:bold; font-size:24px")),
                                h5(htmlOutput("load_confirmation")),

                                h2(p("Variables and type of variables")),
                                tableOutput('contents_loading'),

                                h2(p("Summary of loaded data")),
                                verbatimTextOutput("test_loading"),

                                br()

                        ),




# Data inspection

                        navbarMenu("Data inspection",

                                   #### Data summary
                                   tabPanel("Data summary", value=1,

                                            h4(p("In the 'Data inspection' module, you will be abl
e to report a data summary, full display of data,
                                                 missing values, correlation between variables, tr
ansformation of Y and Xs variables and update the content of
                                                 the dataset. Let's begin with 'Data summary'"), s
tyle = "color:red; font-style: italic"),
                                            hr(),

                                            h2(p("Data summary", style = "color:orange; font-weigh
t:bold")),

                                            h3(htmlOutput("description_dataset", style = "color:gr
een; font-weight:bold")),

                                            p(htmlOutput(outputId = "plot_description_example", st
yle = "color:purple; font-weight:bold; font-size:18px")),


                                            hr(),


                                            h3("Type of variables", style = "color:blue; font-weig
ht:bold"),
                                            p("Remember that you may modify now the type of variab
les (categorical or continuous)", style = "color:red; font-weight:bold"),

                                            tableOutput('contents2'),
                                            verbatimTextOutput("test"),

                                            h3("Summary of Data", style = "color:blue; font-weight
:bold"),

                                            #htmlOutput(outputId = "sum_data"),
                                            htmlOutput(outputId = "sum_data"),

                                            h3("Data display", style = "color:blue; font-weight:bo
ld"),


                                            DT::dataTableOutput(outputId = "virt_data")

                                   ),

#### Full data view
```

```
                                tabPanel("Full data view", value=1,
                                        h2(p("Full data view", style = "color:orange; font-wei
ght:bold")),
                                        p("To adjust the display (first 6 registers or the who
le dataset), please, refer to the left panel"),
                                        tableOutput('contents'),
                                        br()
                                ),

#### Missing values
                                tabPanel("Missing values", value=1,
                                        h2(p("Missing values", style = "color:orange; font-wei
ght:bold")),
                                        br(),
                                        p("Missing data are quite common in longitudinal studi
es, often due to dropout. Multivariate
                                   repeated-measures ANOVA models are often used in practice to
analyze repeated measures
                                   or longitudinal data sets, but LMMs offer two primary advant
ages over these
                                   multivariate approaches when there are missing data:"),

                                        p("First, they allow subjects being followed over time
to have unequal numbers of measurements
                                   (i.e., some subjects may have missing data at certain time p
oints). If a subject
                                   does not have data for the response variable present at all
time points in a longitudinal
                                   or repeated-measures study, the subject's entire set of data
is omitted in a multivariate
                                   ANOVA (this is known as listwise deletion); the analysis the
refore involves complete
                                   cases only. In an LMM analysis, all observations that are av
ailable for a given subject are
                                   used in the analysis.

                                   Second, when analyzing longitudinal data with repeated-measu
res ANOVA techniques,
                                   time is considered to be a within-subject factor, where the
levels of the time factor are
                                   assumed to be the same for all subjects. In contrast, LMMs a
llow the time points when
                                   measurements are collected to vary for different subjects.")
,


                                        h2("List of missing values", style = "color:blue; font
-weight:bold"),
                                        tableOutput("freq_na_values"),

                                        h2("Missing values by observation-variable", style = "
color:blue; font-weight:bold"),

                                        plotOutput("plot_missing1"),

                                        h2("Missing values among variables", style = "color:bl
ue; font-weight:bold"),
                                        plotOutput("plot_missing2"),

                                        #h2("Missing values between two variables", style = "c
olor:blue; font-weight:bold"),
                                        #plotOutput("plot_missing3"),

                                        br()

                                ),

#### Correlation between variables

                                tabPanel("Correlation between variables", value=1,
                                        h2(p("Correlation between variables", style = "color:o
```

```
range; font-weight:bold")),
                                        h4("Important:", style = "color:red; font-weight:bold"
),
                                        p("Remember not to add variables that are highly corre
lated (r > 0.70 based on Dormann et al., 2013) because
                                          when you include correlated variables into a paramet
reic model it will have a problem of coefficients' stability as it will
                                          lose accuracy in determining the coefficients. In ca
se that you have correlation between variables, remember to choose one
                                          of the correlated variables using biological knowled
ge/reasoning to select the most meaningful variable or conduct a
                                          dimension-reduction analysis (e.g. Principal Compone
nts Analysis) leaving a single variable that accounts for most of
                                          the shared variance among the correlated variables")
,

                                        h2("Correlation plot of complete observations", style
= "color:blue; font-weight:bold"),
                                        htmlOutput("corrtext"),
                                        h4("Important: crossed out values in the correlation p
lot represent non-significant (P-value > 0.05) correlations", style = "color:red; font-weight:bo
ld"),

                                        br(),
                                        verbatimTextOutput(outputId = "paired_correlated_vars2
"),

                                        plotOutput(outputId = "correlation_plot_n", width = "7
5%", height = "800px"),



                                        br()

                                ),

#### Box-cox transformation
                                tabPanel("Box-cox transformation", value=1,
                                        h2(p("Box-cox transformation", style = "color:orange;
font-weight:bold")),

                                        h4("Apply box-cox transformation to the response varia
ble (Y)", style = "color:red; font-weight:bold"),

                                        h3("Initial data", style = "color:blue; font-weight:bo
ld"),
                                        verbatimTextOutput("table_pre_transform"),

                                        h3("After Box-cox", style = "color:blue; font-weight:b
old"),
                                        verbatimTextOutput("table_post_transform"),

                                        h5("Here you can check the histogram after applying BO
X-COX, if you want to apply it to your data, press
                                           the buttom on the left menu", style = "color:red; font-st
yle:italic"),

                                        verbatimTextOutput("prueba_transform"),
                                        plotOutput(outputId = "box_lambda"),

                                        plotOutput(outputId = "plot_no_transf"),
                                        plotOutput(outputId = "plot_transf"),

                                        br()

                                ),

#### Scale and Center

                                tabPanel("Scale, center and other transformations", value=1,
                                        h2(p("Scale, center and other transformations", style
= "color:orange; font-weight:bold")),
```

```
                                        p("Centering covariates at specific values (i.e., subt
racting a specific value, such as the mean,
                                          from the observed values of a covariate) has the effect of c
hanging the intercept in the
                                          model, so that it represents the expected value of the depen
dent value at a specific value
                                          of the covariate (e.g., the mean), rather than the expected
value when the covariate is
                                          equal to zero (which is often outside the range of the data)
. In addition to changing the
                                          interpretation of the intercept in a linear model, centered
covariates often reduce the
                                          amount of collinearity among the covariates with associated
fixed effects in the model."),

                                        h3("Initial data", style = "color:blue; font-weight:bo
ld"),
                                        verbatimTextOutput("show_pre_scale_center"),

                                        h3("Transformed data", style = "color:blue; font-weigh
t:bold"),
                                        verbatimTextOutput("show_scale_center"),

                                        br()
                                ),



#### Update dataset
                                tabPanel("Update dataset", value=1,
                                        h2(p("Update dataset", style = "color:orange; font-wei
ght:bold")),

                                        h3("Modify your dataset"),
                                        p("In this module you can remove, update or add regist
ers. If any modification is done you must use 'Modified database' in the #0 step"),

                                        editableDTUI("table1"),

                                        br()
                                )
# CLOSE "Data inspection"
                        ),



# Plots
                        tabPanel("Plots", value=2,
                                h2("X-Y plot of your data", style = "color:green; font-weight:bol
d"),
                                p("Please, select X and Y variables together with a grouping vari
able. Optionally enter a title for the plot."),

                                p(htmlOutput(outputId = "plot_example", style = "color:purple; fo
nt-weight:bold; font-size:18px")),

                                p("Remember to click on 'Create or update graph'", style = "color
:red; font-weight:bold"),



                                plotlyOutput(outputId = "scatterplot"),
                                br(),
                                #plotlyOutput(outputId = "plot_by_slope"),

                                h5(textOutput("description")),

                                h2("Linear regression analysis of the data", style = "color:green
; font-weight:bold"),

                                htmlOutput("sumtable"),
                                br(),
```

```
                                htmlOutput("sumtext"),
                                br(),
                                verbatimTextOutput("sum"),
                                br()

                        ),

# EDA

                        tabPanel("EDA", value=3,
                                br(),
                                p("Here you have an Exploratory Data Analysis (EDA) summary of yo
ur data", style = "color:black; font-weight:bold; font-size:24px"),
                                p(htmlOutput(outputId = "EDA_example", style = "color:purple; fon
t-weight:bold; font-size:18px")),


                                h3("EDA Report:", style = "color:green; font-weight:bold; font-si
ze:24px"),
                                br(),
                                uiOutput("inc", style = "color:red; font-weight:bold; font-size:2
4px"),

                                h3("Missing values:", style = "color:orange; font-weight:bold; fo
nt-size:24px"),

                                br(),
                                plotOutput(outputId = "EDA_missings"),
                                p("Good: missings < 5%"),
                                p("OK: 5% < missings < 40%"),
                                p("Bad: 40% < missings < 80%"),
                                p("Remove: missings > 80%"),



                                h3("Histogram:", style = "color:orange; font-weight:bold; font-si
ze:24px"),
                                br(),
                                plotOutput(outputId = "EDA_histogram"),

                                h3("Density plot:", style = "color:orange; font-weight:bold; font
-size:24px"),
                                br(),
                                plotOutput(outputId = "EDA_density"),

                                h3("Correlations:", style = "color:orange; font-weight:bold; font
-size:24px"),
                                br(),
                                plotOutput(outputId = "EDA_correlation"),

                                h3("Barplots:", style = "color:orange; font-weight:bold; font-siz
e:24px"),
                                br(),
                                plotOutput(outputId = "EDA_barplots"),
                                br()
                        ),



 # Linear-Mixed Model
                        navbarMenu("Linear-Mixed Model",

                                   #### Explanation
tabPanel("What is a linear-mixed model (LMM)?", value=4,
                                            br(),

                                            h4(p("In this module you will be able to conduct a ste
p-by-step LMM analysis. After reading this information, proceed to Step0.",
                                                 style = "color:blue; font-weight:bold; font-size:
22px")),

                                            h4(p("Remember to press each respective button to gene
```

```
rate the analyses for each Step.",
                                                    style = "color:red; font-style:italic; font-size:
22px")),

                                           hr(),

                                           em("What is a linear-mixed model (LMM)?", style = "col
or:blue; font-weight:bold; font-size:36px"),
                                           br(),br(),
                                           p("A mixed model (or more precisely mixed error-compon
ent model) is a statistical model containing both fixed effects and random effects .
These models are useful in a wide variety of disciplines in the physical, biological and social
sciences.
They are particularly useful in settings where repeated measures are made on the same statistica
l units (longitudinal study),
or where measures are made on clusters of related statistical units. Because of their advantage
in dealing with missing values,
mixed effects models are often preferred over more traditional approaches such as repeated measu
res ANOVA."),



                                           h2("R Packages", style = "color:orange; font-weight:bo
ld"),
                                           p("There are different packages in R for performing LM
M. However, lme4 and nlme are the most used."),
                                           p("In this program we have used nlme.", style = "color
:red; font-weight:bold"),



                                           h2(p("Function 'lme()' from 'nlme' package", style = "
color:orange; font-weight:bold")),

                                           p("Function lme() uses two different arguments “fixed”
and “random” to inform about the fixed variable(s) (FV) and random variable(s) (RV).", style = "
color:red; font-weight:bold"),
                                           p("If there are no fixed effects the nomenclature is:
lme(Y ~ RV)"),

                                           p("If there are both fixed and random effects: lme(Y ~
FV, random = ~ 1|RV)"),

                                           p("In most LMM it is assumed that RV have a mean of 0
and that what we want to quantify if the variation in the constant (1|) due to the
differences between the levels of the factor RV. The vertical line means “given the following di
stribution of RVs”."),

                                           p("If we want to specify a RV, not only over the const
ant of the model, but also over some FVs, the model is: lme(Y ~ FV, random = ~ FV | RV) or (rand
om = ~ 1+FV | RV)"),

                                           p("If we have more than RV we will use: random = list(
~ 1|RV1, ~ 1|RV2, …, ~1|RVi)"),

                                           p("If there are different RV, but some are nested insi
de others, then: random = ~ 1 | RV1/RV2/…/RVi; meaning that there are “i” RVs with
                                             “RVi” nested inside RVi-1 that is nested inside RVi-
2 and so on until the RV1."),



                                           h3("A) Handling missing data", style = "color:orange;
font-weight:bold"),

                                           p("Argument “na.action”", style = "color:red; font-wei
ght:bold"),

                                           p("- na.fail (default option) leads to an error in pre
sence of missing values. "),
                                           p("- na.omit deals with missing values by removing the
corresponding lines in the dataset."),
                                           p("- na.exclude ignores the missing values but, compar
ed to na.omit, it enables to have outputs with
```

```
the same number of observations compared to the original dataset."),
                                        p("- na.pass will continue the execution of the functi
on without any change.
If the function cannot manage missing values, it will lead to an error."),

                                        h3("B) Specifying correlation structure", style = "col
or:orange; font-weight:bold"),
                                        p("In this version of the program this feature is not
implemented yet. Wait for it!"),

                                        h3("C) Tools for model selection", style = "color:oran
ge; font-weight:bold"),
                                        p("In this version of the program we are using one inf
ormation criteria (AIC (Akaike information criterion, Akaike 1973)) for model selection in Steps
2 and 3.
                                          However, in practice it should be combined with BIC
(Bayes information criterion)."),


                                        br()

                                    ),
#### Step 0
tabPanel("Step 0. Base models", value=4,
                                        br(),

                                        em("Step 0. Base models", style = "color:blue; font-we
ight:bold; font-size:36px"),
                                        hr(),

                                        p("In this initial step, every predictor (fixed variab
les) will be included in an independent model together with the variable of TIME and the
                                  random variable assuming a random intercept model.", style = "colo
r:black; font-weight:bold"),


                                        p("The user needs to select those predictors (fixed va
riables) with a P-value < 0.2 (Maldonado and Greenland, 1993) to be
                                          included in the next step.", style = "color:black; f
ont-weight:bold"),

                                        #htmlOutput(outputId = "show_equation_step0"),


                                        h3("MODELS GENERATED:", style = "color:blue; font-weig
ht:bold; font-size:26px"),
                                        p("REML is used ", style = "color:red; font-weight:bol
d"),

                                        verbatimTextOutput(outputId = "summary_model_step0"),

                                        h3("ANOVA FOR EACH MODEL BASED ON F-TEST", style = "co
lor:blue; font-weight:bold; font-size:26px"),
                                        verbatimTextOutput(outputId = "anova_model_step0"),

                                        p(htmlOutput(outputId = "step0_example", style = "colo
r:purple; font-weight:bold; font-size:18px")),



                                        h3("Correlations", style = "color:red; font-weight:bol
d; font-size:26px"),
                                        htmlOutput("corrtext2"),

                                        p("Importantly, remember not to add variables that are
highly correlated (r > 0.70 based on Dormann et al., 2013) because when you include correlated v
ariables
                                    into a parametreic model it will have a problem of coeffi
cients' stability as it will lose accuracy in determining the coefficients.
                                    In case that you have correlation between variables, reme
mber to choose one of the correlated variables using biological knowledge/reasoning to select
                                  the most meaningful variable or conduct a dimension-reduction anal
```

```
ysis (e.g. Principal Components Analysis)
                              leaving a single variable that accounts for most of the shared var
iance among the correlated variables", style = "color:black; font-weight:bold"),

                                       plotOutput(outputId = "correlation_plot_n2", width = "
75%", height = "400px"),    ### new syntax added!

                                       verbatimTextOutput(outputId = "paired_correlated_vars"
),

                                       p(htmlOutput(outputId = "step0_example_corr", style =
"color:purple; font-weight:bold; font-size:18px")),


                                       p("Please, proceed to Step 1", style = "color:red; fon
t-weight:bold; font-size:30px"),

                                       #h3("Variance inflation factor (VIF)", style = "color:
green; font-weight:bold; font-size:26px"),


                                       #p("Alternatively you may also evaluate collinearity b
ased on VIF and dropping variables with a VIF higher than a
                              #certain value (e.g. 3, Zuur, Ieno & Elphick, 2010; or 10, Quinn &
Keough, 2002) ", style = "color:red; font-weight:bold"),

                              #            verbatimTextOutput(outputId = "VIF_values_step0"),


                                       br()

                                  ),

#### Step 1
tabPanel("Step 1. Definition of the model: beyond optimal model", value=4,
                                       br(),

                                       em("Step 1. Definition of the 'Beyond Optimal Model'",
style = "color:blue; font-weight:bold; font-size:36px"),
                                       hr(),
                                       p("In this step you have to choose all the fixed terms
that you want to consider and interactions,
                                     together with the main random term (different subjects wh
ere the measures are repeated: 1|ID)", style = "color:black; font-weight:bold"),

                                       p("Remember not to add variables that are highly corre
lated (r > 0.70)!", style = "color:red; font-weight:bold"),



                                       h3("Beyond Optimal Model", style = "color:blue; font-w
eight:bold; font-size:26px"),
                                       htmlOutput(outputId = "show_equation_model_full.nlme")
,
                                       br(),



                                       verbatimTextOutput(outputId = "show_summary_model_full
.nlme"),
                                       h3("ANOVA of the FIXED effects", style = "color:blue;
font-weight:bold; font-size:26px"),
                                       verbatimTextOutput(outputId = "anova_model_full.nlme")
,
                                       p(htmlOutput(outputId = "step1_example", style = "colo
r:purple; font-weight:bold; font-size:18px")),

                                       br()
                                  ),
```

```
#### Step 2
tabPanel("Step 2. Structure of RANDOM effects", value=4,
                                        br(),

                                        em("Step 2. Structure of RANDOM components", style = "
color:blue; font-weight:bold; font-size:36px"),
                                        hr(),
                                        p("In this step we are going to find a proper structur
e for the RANDOM component of the LMM.
                                          We will use the best 'Beyond optimal model' and we w
ill vary the RANDOM component leaving the same FIXED component.", style = "color:black; font-wei
ght:bold"),
                                        p("REML method is used to properly compare the models
using AIC", style = "color:red; font-weight:bold"),

                                        h3(p("When model fits are ranked according to their AI
C values, the model with the lowest AIC value being considered the 'best'", style = "color:red;
font-weight:bold")),



                                        htmlOutput(outputId = "show_equation_step2"),



                                        h3("MODEL comparison based on AIC", style = "color:blu
e; font-weight:bold"),
                                        p("* Values are sorted from lowest (BEST!) to highest
AIC values", style = "color:red; font-weight:bold"),


                                        verbatimTextOutput(outputId = "compare_step2"),
                                        p(htmlOutput(outputId = "step2_example", style = "colo
r:purple; font-weight:bold; font-size:18px")),

                                        br()

                                    ),


#### Step 3
tabPanel("Step 3. Structure of FIXED effects", value=4,
                                        br(),

                                        em("Step 3. Structure of FIXED effects", style = "colo
r:blue; font-weight:bold; font-size:36px"),
                                        hr(),
                                        p("In this step we are going to find a proper structur
e for the FIXED component of the LMM)", style = "color:black; font-weight:bold"),
                                        p("We will use the best model from STEP2 and we will v
ary the FIXED component leaving the same optimized RANDOM component from STEP2", style = "color:
black; font-weight:bold"),
                                        p("ML method is used to properly compare the models us
ing AIC", style = "color:red; font-weight:bold"),

                                        h3(p("When model fits are ranked according to their AI
C values, the model with the lowest AIC value being considered the 'best'", style = "color:red;
font-weight:bold")),



                                        htmlOutput(outputId = "show_equation_step3"),

                                        h3("MODEL comparison based on AIC", style = "color:blu
e; font-weight:bold"),
                                        p("* Values are sorted from lowest (BEST!) to highest
AIC values", style = "color:red; font-weight:bold"),


                                        verbatimTextOutput(outputId = "compare_step3"),
                                        p(htmlOutput(outputId = "step3_example", style = "colo
r:purple; font-weight:bold; font-size:18px")),


                                        br()
```

```
                                ),


#### Step 4
tabPanel("Step 4. Final model", value=4,
                                         br(),

                                         em("Step 4. Final model", style = "color:blue; font-we
ight:bold; font-size:36px"),
                                         p(htmlOutput(outputId = "step4_example", style = "colo
r:purple; font-weight:bold; font-size:18px")),

                                         hr(),

                                         p("In this step we are going to generate the FINAL mod
el based on previous steps", style = "color:black; font-weight:bold"),
                                         p("REML method is used to properly report the FINAL mo
del", style = "color:red; font-weight:bold"),
                                         p("Different parameters of the model will be shown", s
tyle = "color:black; font-weight:bold"),

                                         h3(htmlOutput(outputId = "show_equation_step4")),

                                         h2("SUMMARY OF THE DATA", style = "color:orange; font-
weight:bold"),

                                         h3("RANDOM EFFECTS", style = "color:black; font-weight
:bold"),
                                         verbatimTextOutput(outputId = "sum_ran_effects"),

                                         h3("FIXED EFFECTS - ANOVA BASED ON T-VALUE", style = "
color:black; font-weight:bold"),
                                         verbatimTextOutput(outputId = "sum_fix_effects"),

                                         h2("OVERALL OUTPUT FROM THE MODEL", style = "color:ora
nge; font-weight:bold"),

                                         h3("FULL SUMMARY", style = "color:black; font-weight:b
old"),
                                         verbatimTextOutput(outputId = "show_summary_model_step
4"),

                                         h3("ANOVA BASED ON F-TEST", style = "color:black; font
-weight:bold"),
                                         verbatimTextOutput(outputId = "anova_model_step4"),

                                         h3("R-SQUARED (based on Nakagawa and Schielzeth (2013)
)", style = "color:black; font-weight:bold"),

                                         p("R2m (Marginal R2 (whole model)): it is the marginal
R2 for a linear mixed model, meaning that it is concerned with the variance explained
                                           by the fixed factors", style = "color:blue; font-weigh
t:bold"),

                                         p("R2c (Conditional R2 (whole model)): it is the condi
tional R2 for a linear mixed model, meaning that it is concerned with the variance explained
                                             by the fixed and random factors", style = "color:blu
e; font-weight:bold"),

                                         verbatimTextOutput(outputId = "r.squaredGLMM_step4"),

                                         h3("95% CONFIDENCE INTERVALS FOR THE COEFFICIENTS", st
yle = "color:black; font-weight:bold"),
                                         verbatimTextOutput(outputId = "sum_results_step4"),

                                         br()

                                )

                    ),
```

```
# Checking assumptions
tabPanel("Checking assumptions and getting information", value=4,

                              h2(p("Assumptions from the LMM",
                                   style = "color:orange; font-weight:bold")),

                              p(htmlOutput(outputId = "assumptions_example", style = "color:pur
ple; font-weight:bold; font-size:18px")),



                              # plotOutput(outputId = "qqplot_modifiable"),

                              fluidRow(
                                column(12, align="center",
                                       plotOutput('resi_plot_step4', width = "75%", height = "8
00px")
                                )
                              ),

                              h4(p("1) No pattern should be seen in the plots: 'Fitted values v
s Standardized residuals' (top left plot) and
                                   'Explanatory variable vs Standardized residuals' (top right
plot)", style = "color:black; font-weight:bold")),
                              h4(p("1.a) If we observe increasing or decreasing values in 'Fitt
ed values vs Standardized residuals', it inform us about
                                Heteroscedasticity (i.e. variance of data not approx. equal acr
oss range of predicted values)
                                   that we should correct (try box-cox transformation alternati
ves)", style = "color:black; font-weight:bold")),

                              h4(p("2) Histogram of residuals (bottom left) must follow a Gauss
ian distribution, otherwise it will
                                   inform us about non-normality behaviour", style = "color:bla
ck; font-weight:bold")),
                              h4(p("3) QQ-Plot (bottom right) must show dots aligned to the dia
gonal line, otherwise
                                   it will inform us about non-normality", style = "color:black
; font-weight:bold")),

                              h4(p("4) Normality test: Shapiro-Wilk analysis of residuals", sty
le = "color:black; font-weight:bold")),
                              p("**** If P-value < 0.05 --> Non-normality", style = "color:red;
font-weight:bold"),
                              verbatimTextOutput(outputId = "shapiro_results"),


                              h4(p("5) Multi-collinearity", style = "color:black; font-weight:b
old")),

                              verbatimTextOutput(outputId = "VIF_values"),

                              h2(p("Getting information from the model",
                                   style = "color:orange; font-weight:bold")),

                              h4(p("A) Plot of random effects", style = "color:black; font-weig
ht:bold")),

                              fluidRow(
                                column(12, align="center",
                                       plotOutput('ran_eff_plot', width = "75%", height = "800p
x")
                                )
                              ),

                              h4(p("B) Predicted values for the data", style = "color:black; fo
nt-weight:bold")),

                              h4(p("B1) If the structure is 1|RV then you are constructing a RA
NDOM INTERCEPT MODEL:",
                                   style = "color:blue; font-weight:bold")),
```

```
                                p("Each random variable (subject) is assigned a different interce
pt cause by-RV (by subject) variability is taken into account. However,
                                      the fixed effects are all the same for all RVs.",
                                  style = "color:black; font-weight:bold"),

                                p("In this model, we account for baseline-differences in the depe
ndent variable, but we assume that whatever the effect of the fixed variables,
                                      it is the same for all the RVs.",
                                  style = "color:black; font-weight:bold"),

                                h4(p("B2) If the structure is FX|RV then you are constructing a R
ANDOM SLOPE MODEL:",
                                     style = "color:blue; font-weight:bold")),

                                p("In this case RVs are not only allowed to have differing interc
epts, but they are also allowed to have different slopes for the effect of
                                      FVs.",
                                  style = "color:black; font-weight:bold"),

                                verbatimTextOutput(outputId = "intercep_slope"),


                                br()
                        ),

# Download data
tabPanel("Download data", value=18,
                                br(),
                                h3("Download data including transformed variables and/or fitted v
alues and residuals",
                                   style = "color:orange; font-weight:bold")

                        ),


# Final Report
tabPanel("Final Report", value=5,
                                br(),
                                h3("Generation of the Report", style = "color:orange; font-weight
:bold"),
                                p("Remember that the report will be generated based on the last m
odifications performed into the application. For the
                                  LMMs' section, the results depend on the execution of each Step
. Therefore, steps that have not been executed will not be shown
                                  in the report.", style = "color:red; font-weight:bold"),



                                p("Important: if instead of a .PDF/.HTML/.docx file you get an er
ror file named 'downloadReport.txt' is because
                                  you have checked steps that were not executed from the LMM's mo
dule. It could also be because
                                  your database is not properly loaded. Load it and try again. ",
style = "color:black; font-weight:bold"),

                                br()
                        ),

                        id = "tabselected"
                    )
                )
            )




),

            ##################################################################################
#####################################################
            ################################################### END OF UI #################
```

```
####################################################
                ################################################################################
####################################################


                hr(),

                ################################################################################
####################################################
                ##################################################### FOOT ######################
###############################################
                ################################################################################
####################################################


p("SISSREM: Shiny Interactive, Supervised and Systematic report from REpeated Measures data - 20
19 - Beta Version 1.0.",align = "center"),
uiOutput("tab", align = "center"),

h5("Built with",
img(src = "https://www.rstudio.com/wp-content/uploads/2014/04/shiny.png", height="3%", width="3%
"),
"by",
img(src = "https://www.rstudio.com/wp-content/uploads/2014/07/RStudio-Logo-Blue-Gray.png", heigh
t="9%", width="9%"),
".", align = "center"),

p("We absolutely acknowledge the following R-packages included in this Shiny app:", align = "cen
ter", style = "color:black; font-weight:bold"),
p("car, corrplot, DataExplorer, dichromat, dplyr, DT, editData, ggplot2, htmltools, influence.ME
,
lme4, lmerTest, MASS, mice, MuMIn, naniar, nlme, plotly, questionr, R.devices, rcompanion, readr
, rmarkdown, shiny, shinyjs,
shinythemes, shinyWidgets, sjPlot, stringr, summarytools, tools, usdm, vembedr, xtable", align =
"center", style = "color:blue; font-weight:bold"),
p(" ", align = "center", style = "color:black; font-weight:bold"),
p(" ", align = "center", style = "color:black; font-weight:bold")

)







################################################################################################
####################################
######################################################### SERVER ################################
########################################
################################################################################################
####################################
server <- function(input, output, session) {

#################################### Generation of the database ############################

  data <- eventReactive(input$load_option,{

############################# EXAMPLE DATABASE 1 ####################
    if (input$example_database1 == TRUE)

    {
      df <- read.csv("CSV_FRUIT.csv", header = TRUE, sep = ';',
                     quote = input$quote, na.strings=c("","NA"))

      output$load_confirmation <- renderUI({

        HTML(paste(h2(p("Sucessfully loaded EXAMPLE DATABASE:", style = "color:purple; font-weig
ht:bold")),
                   h4(p("IMPORTANT: all the example comments will appear in purple color.", styl
e = "color:purple; font-weight:bold")),
                   h4("In this example, which is part of the ADER package (data(fruit)), we have
```

```
data regarding cherry tree fructification.
                    We want to explore whether fruit production (FRUIT) depends on tree size, m
easured as the diameter at
                    breast height (DBH). For this, 179 trees are measured.
                    The same individuals (TAG) are resampled for 3 years (YEAR).
                    The DBH is measured only once for all three years and, although this can in
crease by one
                    year for another, the change is so small (they are adult trees) that it has
not been taken into account."),
                  "",
                  h4("REMEMBER: in a repeated measures design, we assume that there is an effec
t of the individual (TAG) which affects to the relationship between Y (FRUIT) and Xs (DBH & YEAR
)"),
                  h4(p("This example is fully explained in 'Modelos lineales mixtos (LMM) y mod
elos lineales generalizados mixtos (GLMM) en R, by Luis Cayuela'.", style = "color:blue; font-we
ight:bold")),
                  "",
                  h2(p("Please, proceed now to 'Data inspection'", style = "color:green; font-w
eight:bold")),
                  sep="<br/>"))

      })



    }


################## EXAMPLE DATABASE 2 ####################
    if (input$example_database2 == TRUE)

    {

      df <- read.csv("CSV_FRUIT_post_NAs_full.csv", header = TRUE, sep = ';',
                     quote = input$quote, na.strings=c("","NA"))

      output$load_confirmation <- renderUI({


        HTML(paste(h2(p("Sucessfully loaded EXAMPLE DATABASE with missing values:", style = "col
or:purple; font-weight:bold")),
                  h4(p("IMPORTANT: all the example comments will appear in purple color.", styl
e = "color:purple; font-weight:bold")),
                  h4("In this example, which is part of the ADER package (data(fruit)), we have
data regarding cherry tree fructification.
                    We want to explore whether fruit production (FRUIT) depends on tree size, m
easured as the diameter at
                    breast height (DBH). For this, 179 trees are measured.
                    The same individuals (TAG) are resampled for 3 years (YEAR).
                    The DBH is measured only once for all three years and, although this can in
crease by one
                    year for another, the change is so small (they are adult trees) that it has
not been taken into account."),
                  "",
                  h4("REMEMBER: in a repeated measures design, we assume that there is an effec
t of the individual (TAG) which affects to the relationship between Y (FRUIT) and Xs (DBH & YEAR
)"),
                  h4(p("This example is fully explained in 'Modelos lineales mixtos (LMM) y mod
elos lineales generalizados mixtos (GLMM) en R, by Luis Cayuela'.", style = "color:blue; font-we
ight:bold")),
                  "",
                  h2(p("Please, proceed now to 'Data inspection'", style = "color:green; font-w
eight:bold")),
                  sep="<br/>"))


      })

    }


################## UPLOADED FILE ####################
```

```
    if (input$example_database1 == FALSE && input$example_database2 == FALSE)

    {
      req(input$file1) ## ?req #  require that the input is available
      inFile <- input$file1
      df <- read.csv(inFile$datapath, header = input$header, sep = input$sep,
                     quote = input$quote, na.strings=c("","NA"))

      output$load_confirmation <- renderUI({
      h2(p("Sucessfully loaded Data. Please, proceed to 'Data inspection'", style = "color:green
; font-weight:bold"))

      })
   }

    if (input$example_database1 == FALSE | input$example_database2 == FALSE){


      # Variables for plot module
      updateSelectInput(session, inputId = 'xcol', label = 'X Variable',
                        choices = names(df), selected = names(df))
      updateSelectInput(session, inputId = 'ycol', label = 'Y Variable',
                        choices = names(df), selected = names(df)[2])
      updateSelectInput(session, inputId = 'zcol', label = 'Grouping variable',
                        choices = c("NULL",names(df)), selected = "NULL")


      # Variables to show
      updateSelectInput(session, inputId = 'sele_variable', label = '',         # Select variabl
es to EXPLORE
                        choices = names(df), selected = names(df))    #selected = names(df)



      # Variables to be modified (Factor, numeric or character)
      updateSelectInput(session, inputId = 'var_factor', label = 'Select variables to change int
o FACTOR',
                        choices = names(df), selected = '')

      updateSelectInput(session, inputId = 'var_numeric', label = 'Select variables to change in
to NUMERIC',
                        choices = names(df), selected = '')

      updateSelectInput(session, inputId = 'var_character', label = 'Select variables to change
into CHARACTER',
                        choices = names(df), selected = '')


      # Variables to be transformed

      updateSelectInput(session, inputId = 'var_box_cox', label = 'Select variable to a tranform
ation',
                        choices = names(df), selected = '')

      updateSelectInput(session, inputId = 'var_center_scale', label = 'Select variable(s) to ap
ply centering/scaling',
                        choices = names(df), selected = '')

      updateSelectInput(session, inputId = 'var_other_transform', label = 'Select a variable to
apply other transformation',
                        choices = names(df), selected = '')
    }

    return(df)
  })




  ################################ Data from WIDE to LONG format ################################
```

```
#

  data_to_transform <- reactive({

    # UPLOADED FILE

    req(input$file_to_transform) ## ?req #  require that the input is available

    inFile <- input$file_to_transform

    # tested with a following dataset: write.csv(mtcars, "mtcars.csv")
    # and                              write.csv(iris, "iris.csv")
    df <- read.csv(inFile$datapath, header = input$header_file_to_transform, sep = input$sep_fil
e_to_transform,
                   quote = input$quote_file_to_transform)

    #### From WIDE to LONG

    #  input$vars_that_vary_long                varying = variables to include. "chol0", "chol1
", "chol2", "hdl0",  "hdl1",  "hdl2"
    #  input$name_var_change_long               name of new variable that differentiates multip
le observations from the same individual
    #  input$identificator_long                 variable in your dataset that identifies multip
le records from the same individual
    #  input$separator_long                     the symbol that separates the name of a varying
column from its number

    updateSelectInput(session, inputId = 'vars_that_vary_long', label = 'Select variables that v
ary with the grouping variable',
                      choices = names(df), selected = '')

    updateSelectInput(session, inputId = 'name_var_change_long', label = 'Write the name of the
grouping variable (time by default)',
                      choices = '', selected = 'time')

    updateSelectInput(session, inputId = 'identificator_long', label = 'Select the identificator
variable (subjects, registers)',
                      choices = names(df), selected = '')

    updateSelectInput(session, inputId = 'separator_long', label = 'Separator of grouped variabl
es (. _ -)',
                      choices = '', selected = '')

    return(df)
  })



  ## Change from WIDE to LONG format

  #### From WIDE to LONG

  df_long <- eventReactive(input$end_definition_long,{

  data_in_long <- reshape(data_to_transform(),
                          varying = input$vars_that_vary_long,
                          timevar = input$name_var_change_long,
                          idvar = input$identificator_long,
                          direction="long",
                          sep = input$separator_long)

        updateSelectInput(session, inputId = 'sele_variable_to_download_transformed', label = 'S
elect variables to DOWNLOAD',
                      choices = names(data_in_long), selected = names(data_in_long))    #selecte
d = names(df)

    return(data_in_long)
  })


  df_to_transform <- reactive({
```

```
    df_to_transform <- df_long()[,input$sele_variable_to_download_transformed]
  })


  output$starting_data <- renderTable({
      return(data_to_transform())
    })

  output$prueba_show_long_data <- renderTable({
    return(df_long())
    })


  output$download_transformed <- downloadHandler(
    filename = function() {
      paste("LONG_data-", substr(input$file_to_transform,1,nchar(input$file_to_transform)-4), "_
", Sys.Date(), ".csv", sep="")
    },
    content = function(file) {
      write.csv(df_to_transform(), file, row.names=FALSE)
    }
  )


  #### DATA EDITING ####
    output$sum_data <- renderUI({

    print(dfSummary(data_to_use(), graph = TRUE, valid.col = FALSE, graph.magnif = 0.75, style =
'grid'),
          max.tbl.height = 800,
          method = "render",
          headings = FALSE,
          bootstrap.css = FALSE)
  })


  # PART 1
  # EDIT TABLE

  # Select variables to show!
  df1 <- reactive({
    req(input$sele_variable)
    df1 <- data()[,input$sele_variable]    # here, data()
  })

  #Alternative with dplyr!
  #df1 <- reactive({
  #  req(input$sele_variable)
  #  df1 <- data() %>% select(input$sele_variable)
  #})

  df2 <- callModule(editableDT,"table1", data=reactive(df1()))
  output$test=renderPrint({
    str(data_to_use())
  })


  # PART 2

  # CHANGE VARIABLE TYPE (as a function)

  contents_change_type_func <- function(data_a_mod){
    mi <- data_a_mod
    selected_factor <- input$var_factor
    selected_numeric <- input$var_numeric
    selected_character <- input$var_character

    mi[selected_factor] <- lapply(mi[selected_factor], factor)

    mi[selected_numeric] <- lapply(mi[selected_numeric], as.character)
    mi[selected_numeric] <- lapply(mi[selected_numeric], as.numeric)
```

```
  mi[selected_character] <- lapply(mi[selected_character], as.character)

  new_database <- mi
  return(new_database)
}



# new_database contains the changed variable TYPES

output$new_data=renderPrint({
  str(contents_change_type_func(df2()))
})

modified_data <- reactive({contents_change_type_func(df2())})


#### BOX-Cox transformation ####

func_to_box_plot <- function(data_load){

  mi <- data_load
  var_to_box_plot <- input$var_box_cox
  y <- mi[,var_to_box_plot]

  if (input$type_transformation=="Log2")
  {T_box = log2(y + 1)   # Transform the original data

  lambda = "Log2"

  Box = MASS::boxcox(y ~ 1,              # Transform Turbidity as a single vector
                     lambda = seq(-6,6,0.1), plotit = TRUE)
  }


  if (input$type_transformation=="Log10")
  {T_box = log10(y + 1)   # Transform the original data

  lambda = "Log10"

  Box = MASS::boxcox(y ~ 1,              # Transform Turbidity as a single vector
                     lambda = seq(-6,6,0.1), plotit = TRUE)

  }

  if (input$type_transformation=="Ln")
  {T_box = log(y + 1)   # Transform the original data

  lambda = "Ln"

  Box = MASS::boxcox(y ~ 1,              # Transform Turbidity as a single vector
                     lambda = seq(-6,6,0.1), plotit = TRUE)

  }


  if (input$type_transformation=="Square root")
  {T_box = sqrt(y)   # Transform the original data

  lambda = "Square root"

  Box = MASS::boxcox(y ~ 1,              # Transform Turbidity as a single vector
                     lambda = seq(-6,6,0.1), plotit = TRUE)

  }


  if (input$type_transformation=="Inverse")
  {T_box = 1/(y+1)   # Transform the original data
```

```
    lambda = "Square root"

    Box = MASS::boxcox(y ~ 1,              # Transform Turbidity as a single vector
                       lambda = seq(-6,6,0.1), plotit = TRUE)

    }


    if (input$type_transformation=="Box-cox")
    {


      Box = MASS::boxcox(y ~ 1,              # Transform Turbidity as a single vector
                         lambda = seq(-6,6,0.1), plotit = TRUE)      # Try values -6 to 6 by 0.1



      Cox = data.frame(Box$x, Box$y)            # Create a data frame with the results

      Cox2 = Cox[with(Cox, order(-Cox$Box.y)),] # Order the new data frame by decreasing y

      # Cox2[1,]                                  # Display the lambda with the greatest log lik
elihood


      lambda = Cox2[1, "Box.x"]                 # Extract that lambda

      T_box = ((y ^ lambda) - 1)/lambda   # Transform the original data

    }


    transf_database <- mi
    transf_database[,var_to_box_plot] <- T_box
    transform_list <- list("transformed_variable" = T_box, "original_variable" = y, "lambda" = l
ambda, "Box_lambda" = Box, "transformed_database" = transf_database)
    return(transform_list)

  }


  valor_transform <- reactive({func_to_box_plot(modified_data())})

  output$table_pre_transform=renderPrint({
    str(modified_data())
  })


  output$table_post_transform=renderPrint({
    str(valor_transform()$transformed_database)
  })

  output$prueba_transform <- renderPrint({

    paste(input$var_box_cox,"was transformed using lambda = ", valor_transform()$lambda)
    # show list
    # valor_transform()

  })


  output$box_lambda <- renderPlot({

    MASS::boxcox(valor_transform()$original_variable ~ 1,              # Transform Turbidity as
a single vector
                 lambda = seq(-6,6,0.1), plotit = TRUE      # Try values -6 to 6 by 0.1
    )

  })
```

```
  output$plot_no_transf <- renderPlot({

    plotNormalHistogram(valor_transform()$original_variable, main = "Histogram of Non-transforme
d variable")

  })




  output$plot_transf <- renderPlot({

    if (valor_transform()$lambda == 0)
    {
      text(x = 0.5, y = 0.5, paste("Transformed plot is not possible due to a null value of lamb
da\n",
                                   "Please, try another transformation\n"),
           cex = 1.6, col = "black")
    }

    if (valor_transform()$lambda != 0)
    {
      plotNormalHistogram(valor_transform()$transformed_variable, main = paste("Histogram of Tra
nsformed variable with lambda = ", valor_transform()$lambda))
    }
  })


  #### Center and Scale data #####
  # input$center_scale
  # database: valor_transform()
  # vars: input$var_center_scale

  func_to_scale_center <- function(data_load){

    scaling_type <- input$center_scale
    var_other_transform2 <- input$var_other_transform
    type_other_transform2 <- input$type_other_transform

    mi <- data_load
    var_to_center_scale <- input$var_center_scale
    var_to_center_scale <- as.factor(var_to_center_scale)

    for (i in levels(var_to_center_scale))

    {
      x <- mi[,i]
      x<-as.matrix(x)


      if (scaling_type=="Only Center")
      {
        #CENTERING but not scaling
        scaledcentered_treatment <- scale(x, center = T, scale = F)
        mi[,i] <- scaledcentered_treatment

      }


      if (scaling_type=="Only Scale")
      {
        #SCALING but not centering
        scaledcentered_treatment <- scale(x, center = F, scale = T)
        mi[,i] <- scaledcentered_treatment

      }


      if (scaling_type=="Center and Scaling (i.e. auto-scaling)")
      {
```

```
        #CENTERING but not scaling
        scaledcentered_treatment <- scale(x, center = T, scale = T)
        mi[,i] <- scaledcentered_treatment

      }



      if (scaling_type=="Range Scaling")
      {
        #CENTERING but not scaling
        scaledcentered_treatment <- apply(x, 2, function(x) ((x - mean(x, na.rm = TRUE))/((max(x
)-min(x)))))
        mi[,i] <- scaledcentered_treatment

      }


      if (scaling_type=="Pareto Scaling")
      {
        #CENTERING but not scaling
        scaledcentered_treatment <- apply(x, 2, function(x) ((x - mean(x, na.rm = TRUE))/sqrt(sd
(x, na.rm = TRUE))))
        mi[,i] <- scaledcentered_treatment

      }


      if (scaling_type=="Vast Scaling")
      {
        #CENTERING but not scaling
        scaledcentered_treatment <- apply(x, 2, function(x) ((x - mean(x, na.rm = TRUE))*(mean(x
, na.rm = TRUE)))/((sd(x, na.rm = TRUE))*(sd(x, na.rm = TRUE))))
        mi[,i] <- scaledcentered_treatment

      }


      if (scaling_type=="Level Scaling")
      {
        #CENTERING but not scaling
        scaledcentered_treatment <- apply(x, 2, function(x) ((x - mean(x, na.rm = TRUE))/(mean(x
, na.rm = TRUE))))
        mi[,i] <- scaledcentered_treatment

      }

      mi[,i] <- as.numeric(mi[,i])


      #### End of Scaling options

    }


    ##  OTHER TRANSFORMATIONS

    if (exists("var_other_transform2")==TRUE)
    {
      y <- mi[,var_other_transform2]
      y<-as.matrix(y)
      expression_transform <- paste0("y ", type_other_transform2)
      scaledcentered_treatment_other <- eval(parse(text=expression_transform))
      #apply(y, 2, function(y) ((y - mean(y, na.rm = TRUE))/(mean(y, na.rm = TRUE))))
      mi[,var_other_transform2] <- scaledcentered_treatment_other

    }


    database_scale_center <- mi
    database_scale_center <- data.frame(database_scale_center)
```

```
  return(database_scale_center)
}



#### Transform SCALE CENTER
valor_database_scale_center <- reactive({

  if (input$trans_box_buttom == TRUE)

  { data_here <- func_to_scale_center(valor_transform()$transformed_database) }

  if (input$trans_box_buttom == FALSE)

  { data_here <- func_to_scale_center(modified_data()) }

  return(data_here)

})




#Output

output$show_pre_scale_center=renderPrint({

  if (input$trans_box_buttom == TRUE)
  {str(valor_transform()$transformed_database)}

  if (input$trans_box_buttom == FALSE)
  {str(modified_data())}

})




output$show_scale_center=renderPrint({
  str(valor_database_scale_center())
})




######################



output$contents <- renderTable({

  if(input$disp == "head") {
    return(head(data_to_use()))
    #return(sapply(df,class))

  }
  else {
    return(data_to_use())
  }

})



output$description_dataset <- renderUI({
```

```
    HTML(paste("This dataset is composed of ",
               dim(data_to_use())[2],
               "VARIABLES and",
               nrow(data_to_use()),
               "REGISTERS"), sep="<br/>")


  })




################# input$database_to_select
  data_to_use <- reactive({

    if (input$database_to_select == "data()")
    {modificada <- data()}       ### used to be data()

    if (input$database_to_select == "modified_data()")
    {
      if (input$trans_box_buttom == TRUE & input$trans_cent_scale_box_buttom == FALSE)
      {modificada <- valor_transform()$transformed_database}


      if (input$trans_box_buttom == FALSE & input$trans_cent_scale_box_buttom == TRUE)
      {modificada <- valor_database_scale_center()}


      if (input$trans_box_buttom == TRUE & input$trans_cent_scale_box_buttom == TRUE)
      {modificada <- valor_database_scale_center()}


      if (input$trans_box_buttom == FALSE & input$trans_cent_scale_box_buttom == FALSE)
      {modificada <- modified_data()}
    }

    # Remove missing values
    if (input$use_complete_data==TRUE)
    {modificada <- modificada[complete.cases(modificada), ]}


    ####
    return(modificada)

  })




  vars_for_LMM <- reactive({
    variab <- names(data_to_use())

    return(variab)
  })




  observe({
    # LMM
    #### LMM: update options ###

    if (input$example_database1 == FALSE | input$example_database2 == FALSE){


      # STEP 0
      updateSelectInput(session, inputId = 'var_y_step0', label = 'Select variable Y',
                        choices = vars_for_LMM(), selected = '')
```

```
      updateSelectInput(session, inputId = 'var_time_step0', label = 'Select variable TIME',
                        choices = vars_for_LMM(), selected = '')

      updateSelectInput(session, inputId = 'var_fixed_step0', label = 'Select fixed predictors',
                        choices = vars_for_LMM(), selected = '')



      # STEP 1
      updateSelectInput(session, inputId = 'var_y', label = 'Select variable Y',
                        choices = vars_for_LMM(), selected = '')

      # STEP 2
      updateSelectInput(session, inputId = 'var_y_step2', label = 'Select variable Y',
                        choices = vars_for_LMM(), selected = '')

      # STEP 3
      updateSelectInput(session, inputId = 'var_y_step3', label = 'Select variable Y',
                        choices = vars_for_LMM(), selected = '')

      # STEP 4
      updateSelectInput(session, inputId = 'var_y_step4', label = 'Select variable Y',
                        choices = vars_for_LMM(), selected = '')

      updateSelectInput(session, inputId = 'var_fixed_step4', label = 'Select FIXED variables',
                        choices = vars_for_LMM(), selected = '')

      updateSelectInput(session, inputId = 'var_random_step4', label = 'Select RANDOM variables'
,
                        choices = vars_for_LMM(), selected = '')

      updateSelectInput(session, inputId = 'var_fixed_with_random_step4', label = 'Select FIXED
variable to nest with RANDOM',
                        choices = vars_for_LMM(), selected = '')


      #### BOXPLOT: variable to add in the boxplot
      updateSelectInput(session, inputId = 'var_fixed_for_boxplot', label = 'Select fixed variab
le for BOXPLOT',
                        choices = vars_for_LMM(), selected = '')
    }
  })



  #### In LOADING DATA ####

  output$contents_loading <- renderTable({
    m <- data()[, sele_variable()]
    cbind("Variable name"=colnames(m), "Type of variable"=sapply(m, class))
    #paste(sapply(m, class))

  })


  output$test_loading=renderPrint({
    str(data())
  })
```

```
  ################

  output$contents2 <- renderTable({
    m <- data_to_use()[, sele_variable()]
    cbind("Variable name"=colnames(m), "Type of variable"=sapply(m, class))

  })




  # Print data table if checked
  output$virt_data <- DT::renderDataTable(
    DT::datatable(data = data_to_use()[, sele_variable()],
                  options = list(pageLength = 10),
                  rownames = FALSE)
  )




  # x and y as reactive expressions
  xcol <- reactive({ toTitleCase(str_replace_all(input$xcol, "_", " ")) })
  ycol <- reactive({ toTitleCase(str_replace_all(input$ycol, "_", " ")) })

  sele_variable <- reactive({input$sele_variable})
  zcol <- reactive({input$zcol})




  ############################################################################## EDA ##################
#########################################################

  # EDA_missings
  output$EDA_missings <- renderPlot({
    plot_missing(data_to_use())
  })


  # EDA_histogram
  output$EDA_histogram <- renderPlot({
    plot_histogram(data_to_use())
  })

  # EDA_density
  output$EDA_density <- renderPlot({
    plot_density(data_to_use())
  })


  # EDA_correlation
  output$EDA_correlation <- renderPlot({
    plot_correlation(data_to_use(), type = 'continuous')
  })

  # EDA_barplots
  output$EDA_barplots <- renderPlot({
    plot_bar(data_to_use())
  })


  # Create report for EDA          NOT USED

  #observeEvent(input$report_EDA, {
  #  if(input$report_EDA == TRUE){
  #
  #   output$create_report <- renderUI({
```

```
#     create_report(data_to_use())
#
#   })
#
#
#   getPage<-function() {
#     create_report(data_to_use())
#     return(includeHTML("report.html"))
#   }
#   output$inc<-renderUI({
#     HTML(paste("REPORT CREATED - CLICK ON THE BUTTON TO DOWNLOAD IT"))
#   })
#
#
#   }
#   })


#### EDA report



output$download_EDA_html <- downloadHandler(
  filename = 'report.html',

  content = function(file) {
    create_report(data_to_use())
    output$inc<-renderUI({
      HTML(paste("A REPORT IN HTML FORMAT HAS BEEN CREATED"))
    })


    src <- normalizePath('report.html')
    # temporarily switch to the temp dir, in case you do not have write
    # permission to the current working directory
    owd <- setwd(tempdir())
    on.exit(setwd(owd))
    file.copy(src, 'report.html', overwrite = TRUE)
    out <- 'report.html'
    file.rename(out, file)
  }
)




  ################################################################################ Missing values #######
#####################################################################


# to table missing values

output$freq_na_values <- renderTable({

  freq_of_na <- data.frame(freq.na(data_to_use()))[,1:2]
  cbind("Variables"=rownames(freq_of_na), "Missing values"=freq_of_na[,1])
})



# Plot with missing values

output$plot_missing1 <- renderPlot({
  vis_miss(data_to_use())
})


# Plot with missing values - Explore patterns

output$plot_missing2 <- renderPlot({
  gg_miss_upset(data_to_use())
```

```
  })


  #output$plot_missing3 <- renderPlot({
  #  ggplot(data_to_use(),
  #         aes(x = xcol(),
  #             y = ycol())) +
  #    geom_miss_point()
  #})




  ######## PLOTS #######

  # Create scatterplot object
  ggstart<-eventReactive(input$generate_graph,{
    x<-input$xcol
    y<-input$ycol
    z<-input$zcol
    d<-data_to_use()
    tx<-sprintf("g0<-ggplot(data=d,aes(x=%s,y=%s, color=%s))",x,y,z)
    eval(parse(text=tx))
    return(g0)
  })

  output$scatterplot <- renderPlotly({
    g0<-ggstart()
    g1<-g0+geom_point() + geom_smooth(method=lm)+theme_bw() + labs(x = xcol(),
                                                                   y = ycol(),
                                                                   color = toTitleCase(str_repla
ce_all(input$zcol, "_", " ")),
                                                                   title = toTitleCase(input$plo
t_title))
    ggplotly(g1)
  })


  # Create description of plot
  output$description <- renderText({
    paste("The plot above shows the relationship between",
          xcol(),
          "and",
          ycol(),
          "for",
          nrow(data_to_use()),
          "datapoints.")
  })



  # Plots linear regression

  linearmodel<-reactive({
    x<-input$xcol
    y<-input$ycol
    d<-data_to_use()
    tx<-sprintf("mod<-lm(data=d,%s~%s)",y,x,x)
    eval(parse(text=tx))
    return(mod)
  })



  # Two outputs at the same time:

  output$sumtable <- renderText({
    mod<-linearmodel()
    a<-summary(mod)
    output$sum<-renderPrint(a)
    print(xtable(a),type="html")
```

```
  } )


  output$sumtext <- renderText({
    x<-input$xcol
    y<-input$ycol
    mod<-linearmodel()
    a<-summary(mod)
    int<-round(coef(mod)[1],4)
    slope<-round(coef(mod)[2],4)
    rsq<-round(a$r.squared,3)
    sprintf("The regression line is %s = %s + %s %s with an R2 value of %s",y,int,slope,x,rsq)

  } )




  # Create scatterplot object by slope - NOT RUN
  # ggstart_slope<-eventReactive(input$generate_graph,{
  #  x<-input$xcol
  #  y<-input$ycol
  #  z<-input$zcol
  #  d<-data_to_use()
  #  tx<-sprintf("g0<-ggplot(data=d,aes(x=%s,y=%s, fill = %s, color=%s))",x,y,z,z)
  #  eval(parse(text=tx))
  #  return(g0)
  # })

  # output$plot_by_slope <- renderPlotly({
  #  g0<-ggstart_slope()
  #  g1<-g0+geom_point() + geom_smooth(method=lm)+theme_bw() + labs(x = xcol(),
  #                                                                 y = ycol(),
  #                                                                 color = toTitleCase(str_repl
ace_all(input$zcol, "_", " ")),
  #                                                                 title = toTitleCase(input$pl
ot_title))
  #  ggplotly(g1)
  # })




  output$plot_example <- renderUI({
    if (input$example_database1 == TRUE || input$example_database2 == TRUE){

      HTML(paste("Here we will construct a plot based on the expression log(FRUIT) ~ DBH, groupe
d by YEAR. You may also observe the linear regression analysis of this data.",
                 sep="<br/>",
                 "After exploring this module, please proceed to 'EDA' module."))
    }
  })




  # Description of the example dataset to include!

  output$plot_description_example <- renderUI({

    if (input$example_database1 == TRUE || input$example_database2 == TRUE){

      HTML(paste("Consider the following:"),
           sep="<br/>",
           paste("a) 'TAG' and 'YEAR' are selected as factor variables"),
           sep="<br/>",
           paste("b) 'Y' variable 'FRUIT' is log2 transformed to normalize its values (i.e. coun
ts)"),
           sep="<br/>",
           sep="<br/>",
           paste("Now, take your time to explore the subsections of this 'Data inspection' modul
```

```
e. Then, proceed to 'Plots' module.")
      )

    }
  })


  # Message EDA for example database
  output$EDA_example <- renderUI({

    if (input$example_database1 == TRUE || input$example_database2 == TRUE){

      HTML(paste("Here you can find an EDA analysis of the example database. Check the different
options and try the 'Click to generate report' function.
                 Then proceed to 'Linear-Mixed Model' module.",
                 sep="<br/>"))

    }
  })









  ########################################################################## LMM ##################
#########################################################


  # REDEFINE REML

  method_REML <- reactive({
    if (input$REML == TRUE){decide<-c("ML")}
    if (input$REML == FALSE){decide<-c("REML")}
    return(decide)
  })


  method_REML_step2 <- reactive({
    if (input$REML_step2 == TRUE){decide<-c("ML")}
    if (input$REML_step2 == FALSE){decide<-c("REML")}
    return(decide)
  })

  method_REML_step3 <- reactive({
    if (input$REML_step3 == TRUE){decide<-c("ML")}
    if (input$REML_step3 == FALSE){decide<-c("REML")}
    return(decide)
  })


  method_REML_step4 <- reactive({
    if (input$REML_step4 == TRUE){decide<-c("ML")}
    if (input$REML_step4 == FALSE){decide<-c("REML")}
    return(decide)
  })







  #### OBSERVE for the examples ####

  observeEvent(input$load_option, {
```

```
    if (input$example_database1 == TRUE || input$example_database2 == TRUE){

      # Database to use

      updateRadioButtons(session, 'database_to_select', '0) Database to select',
                         c("Initial - RAW uploaded version"='data()',
                           "Modified - If ANY change was made"='modified_data()'),
                         'modified_data()')

      # change to factor
      updateSelectInput(session, inputId = 'var_factor', label = 'Select variables to change int
o FACTOR',
                        choices = names(df), selected = c('TAG', 'YEAR'))

      # Box-cox
      updateSelectInput(session, inputId = 'var_box_cox', label = 'Select variable to a tranform
ation',
                        choices = names(df), selected = 'FRUIT')

      updateSelectInput(session, inputId = 'type_transformation', label = "Choose type of transf
ormation",
                        choices = c("Box-cox", "Log2", "Log10", "Ln", "Square root", "Inverse"),
selected = 'Ln')

      updateCheckboxInput(session, inputId = 'trans_box_buttom', value = TRUE)

    }

  })



  observeEvent(input$trans_box_buttom, {

    if (input$example_database1 == TRUE || input$example_database2 == TRUE){

      # Plots
      updateSelectInput(session, inputId = 'xcol', label = 'X Variable',
                        choices = names(df), selected = 'DBH')
      updateSelectInput(session, inputId = 'ycol', label = 'Y Variable',
                        choices = names(df), selected = 'FRUIT')
      updateSelectInput(session, inputId = 'zcol', label = 'Grouping variable',
                        choices = names(df), selected = 'YEAR')

      # LMM


      # Step0
      updateSelectInput(session, inputId = 'var_y_step0', label = 'Select variable Y --> FRUIT',
                        choices = vars_for_LMM(), selected = 'FRUIT')

      updateSelectInput(session, inputId = 'var_time_step0', label = 'Select variable TIME --> Y
EAR',
                        choices = vars_for_LMM(), selected = 'YEAR')

      updateSelectInput(session, inputId = 'var_fixed_step0', label = 'Select fixed predictors -
-> DBH',
                        choices = vars_for_LMM(), selected = 'DBH')

      updateTextInput(session, inputId = "text_random_step0", value = "list(~1|TAG)")

      # STEP 1
      updateSelectInput(session, inputId = 'var_y', label = 'Select variable Y --> FRUIT',
                        choices = vars_for_LMM(), selected = 'FRUIT')

      updateTextInput(session, inputId = "text_fixed_step1", value = "YEAR * DBH")

      updateTextInput(session, inputId = "text_random_step1", value = "list(~1|TAG)")
```

```
      # STEP 2
      updateSelectInput(session, inputId = 'var_y_step2', label = 'Select variable Y --> FRUIT',
                        choices = vars_for_LMM(), selected = 'FRUIT')

      updateTextInput(session, inputId = "text_fixed_step2", value = "YEAR + DBH")

      updateTextInput(session, inputId = "text_random_INTERCEPT_step2", value = "list(~1|TAG)")

      updateTextInput(session, inputId = "text_random_SLOPE_step2", value = "list(~YEAR|TAG)")



      # STEP 3
      updateSelectInput(session, inputId = 'var_y_step3', label = 'Select variable Y --> FRUIT',
                        choices = vars_for_LMM(), selected = 'FRUIT')

      updateTextInput(session, inputId = "text_fixed_ALONE_step3", value = "DBH")
      updateTextInput(session, inputId = "text_fixed_sum_step3", value = "YEAR + DBH")
      updateTextInput(session, inputId = "text_fixed_interactions_step3", value = "YEAR * DBH")
      updateTextInput(session, inputId = "text_random_step3", value = "list(~1|TAG)")

      # STEP 4
      updateSelectInput(session, inputId = 'var_y_step4', label = 'Select variable Y --> FRUIT',
                        choices = vars_for_LMM(), selected = 'FRUIT')
      updateTextInput(session, inputId = "text_fixed_step4", value = "YEAR + DBH")
      updateTextInput(session, inputId = "text_random_step4", value = "list(~1|TAG)")

      #### BOXPLOT: variable to add in the boxplot

      updateSelectInput(session, inputId = 'var_fixed_for_boxplot', label = 'Select fixed variab
le for BOXPLOT --> YEAR',
                        choices = vars_for_LMM(), selected = 'YEAR')
    }
  })







  ####### STEP 0 ######

  model_full.nlme_step0_1 <- eventReactive(input$Step0_BUTTON, {paste0("lme(",input$var_y_step0,
" ~ ", input$var_fixed_step0, "+",input$var_time_step0,", random=", input$text_random_step0, ",
na.action = input$type_missing", ", data=data_to_use(),", "method='REML' ", ")")})

  output$summary_model_step0 <- renderPrint({
    results_list_model  <- list()
    results_list_model[[1]] <- paste0(input$var_y_step0," ~ ", input$var_fixed_step0, " + ",inpu
t$var_time_step0,", random=", input$text_random_step0)

    data.frame("MODELS"=unlist(results_list_model))
  })


  output$anova_model_step0 <- renderPrint({
    results_list  <- list()

    for (i in 1:(length(input$var_fixed_step0))){
      sum_model <- paste("anova(",model_full.nlme_step0_1()[i],")")
      results_list[[i]] <- eval(parse(text=sum_model))
    }

    results_list
  })


  # COMMENTS FROM THE LMM
```

```
  # STEP 0
  observeEvent(input$Step0_BUTTON, {
    if (input$example_database1 == TRUE || input$example_database2 == TRUE){
      output$step0_example <- renderUI({
          HTML(paste("As you may observe, all the FVs are significant!",
                     sep="<br/>"))
      })

      output$step0_example_corr <- renderUI({
          HTML(paste("As you may observe, no significant correlations have been found.",
                     sep="<br/>"))
       })
    }
  })



  # STEP 1
  observeEvent(input$Step1_BUTTON, {
      if (input$example_database1 == TRUE || input$example_database2 == TRUE){
            output$step1_example <- renderUI({
                HTML(paste("As you may observe, all the FVs are significant but the interaction
term is not. We will evaluate it in Step 3. Now, please proceed to Step 2.",
                    sep="<br/>"))
              })
    }
  })




  # STEP 2
  observeEvent(input$Step2_BUTTON, {
        if (input$example_database1 == TRUE || input$example_database2 == TRUE){
            output$step2_example <- renderUI({
                HTML(paste("As you may observe, mod1 is the one with the lowest AIC value (i.e.
Model 1: random intercept model). We kept it for Step 3.",
                    sep="<br/>"))
      })
    }
  })



  # STEP 3
  observeEvent(input$Step3_BUTTON, {
        if (input$example_database1 == TRUE || input$example_database2 == TRUE){
            output$step3_example <- renderUI({
                HTML(paste("As you may observe, mod5 is the one with the lowest AIC value (i.e.
Model 5: addition model). We kept it for Step 4 (Final model).",
                    sep="<br/>"))
              })
    }
  })




  # STEP 4
  observeEvent(input$Step4_BUTTON, {
        if (input$example_database1 == TRUE || input$example_database2 == TRUE){
            output$step4_example <- renderUI({
                HTML(paste("At this point, we now have to generate our model using REML and base
d on the structure of the FVs and RV of the previous steps.",
                    sep="<br/>",
                    " ",
                    "You can observe that DBH has a significant effect over FRUIT production (Ln
of FRUIT). In the second year, there are on average more fruits than in the first year,
                    while in the third year there are many less. We also found that the random ef
fect (tree effect) is significant in this model. Finally, marginal R2 explains approximately
                    a 20% of the variability in FRUIT production, whereas conditional R2 approxim
```

```
ately a 50%.",
                  "We have not finished yet, we have now to explore whether this model is adequ
ate in terms of the assumptions, mainly normality and homocedasticity."))
             })
    }
  })



  # Assumptions, when Step_4_BUTTON

  observeEvent(input$Step4_BUTTON, {
        if (input$example_database1 == TRUE || input$example_database2 == TRUE){
            output$assumptions_example <- renderUI({
               HTML(paste("Below you have different graphs to test that the assumptions of our
LMM are met, together with
                  general considerations about how it should behave. You may take your time to
explore them.",
                  "You can observe that the model is quite adequate as the assumptions of norma
lity, homocedasticity and linearity are met. However,
                   we may also observe three outliers in the QQ-plot (one in the lowest theoret
ical quantiles and two in the highest theoretical
                  quantiles). By using 'Download data' module, we can further explore these obs
ervations and/or individuals.",
                  " ",

                  "Try now changing the variable FRUIT to its initial value (without the Ln), w
hat do you observe?.",
                  "",
                  "You may now download data from the model in the 'Download data' module, or g
enerate a report in the 'Final report' module.",
                  sep="<br/>"))

      })
    }
  })







  ### Correlation plot for step 0 of LMM

  output$correlation_plot_n2 <- renderPlot({
        if (sum(sapply(data_to_use(), is.numeric)) <2){
      cat(paste("You have less than TWO continuous variables"))}
        if (sum(sapply(data_to_use(), is.numeric)) >=2){
    my_num_data <- data_to_use()[, sapply(data_to_use(), is.numeric)]
    my_num_data <- my_num_data
    M<-cor(my_num_data, use="complete.obs")
    p.mat <- cor.mtest(my_num_data)

    col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))

    corrplot(M, method="color", col=col(20),
             type="upper", order="hclust",
             addCoef.col = "black", # Add coefficient of correlation
             tl.col="black", tl.srt=45, #Text label color and rotation
             # Combine with significance
             p.mat = p.mat[[1]], sig.level = 0.05, insig = "pch", addrect = 3,
             # hide correlation coefficient on the principal diagonal
             diag=FALSE)

    subjects <- sum(complete.cases(my_num_data))
    total <- nrow(my_num_data)

    output$corrtext2 <- renderText({

      sprintf("The number of subjects without any missing data is: %s out of %s",subjects, total
)
```

```
    } )

    }
  })


  output$paired_correlated_vars <- renderPrint({
        if (sum(sapply(data_to_use(), is.numeric)) <2){
      cat(paste("You have less than TWO continuous variables"))}
        if (sum(sapply(data_to_use(), is.numeric)) >=2){
        my_num_data <- data_to_use()[, sapply(data_to_use(), is.numeric)]
    my_num_data <- my_num_data
    M<-cor(my_num_data, use="complete.obs")
    matriz_cor <- M

    for (i in 1:nrow(matriz_cor)){
      correlations <-  which((abs(matriz_cor[i,i:ncol(matriz_cor)]) > 0.7) & (matriz_cor[i,i:nco
l(matriz_cor)] != 1))

      if(length(correlations)> 0){
        lapply(correlations,FUN =  function(x) (cat(paste(colnames(my_num_data)[i], "with", coln
ames(my_num_data)[x]), "\n")))

      }
          }

    if(length(correlations)== 0){
      cat(paste("No paired correlations with r > |0.7|"))
    }

    }
      })




  ### VIF values

  output$VIF_values_step0 <- renderPrint({
    results_list  <- list()
        for (i in 1:(length(input$var_fixed_step0))){
      sum_model <- paste("vif.mer(",model_full.nlme_step0_1()[i],")")
      results_list[[i]] <- eval(parse(text=sum_model))
    }

    results_list
  })




  ######### STEP 1 ##########

  model_full.nlme_estruc1 <- eventReactive(input$Step1_BUTTON,{paste0("lme(",input$var_y," ~ ",i
nput$text_fixed_step1,", random=", input$text_random_step1, ", na.action = input$type_missing",
", data=data_to_use(),", "method=method_REML() ", ")")})
  output$show_equation_model_full.nlme <- renderUI({
    HTML(
      "<hr/>",
      paste(p("Equation is:", style = "color:black; font-weight:bold"),
                #model_full.nlme_estruc1(),
                p(paste0("lme(",input$var_y," ~ ",input$text_fixed_step1,", random = ", input$tex
t_random_step1, ")"), style = "color:blue; font-weight:bold"),
                paste0("Model fit by: ", method_REML()),
                "<br/>",
                paste0("Missing values: na.action = ", input$type_missing)))
  })


  output$show_summary_model_full.nlme <- renderPrint({
```

```
    sum_model <- paste("summary(",model_full.nlme_estruc1(),")")
    eval(parse(text=sum_model))
  })


  output$anova_model_full.nlme <- renderPrint({
    sum_model <- paste("anova(",model_full.nlme_estruc1(),")")
    eval(parse(text=sum_model))

  })



  ######### STEP 2 ##########

  model_full.nlme_0 <- eventReactive(input$Step2_BUTTON,{paste0("gls(",input$var_y_step2," ~ ",i
nput$text_fixed_step2, ", na.action = input$type_missing", ", data=data_to_use(),", "method=meth
od_REML_step2() ", ")")})
  model_full.nlme_1 <- eventReactive(input$Step2_BUTTON,{paste0("lme(",input$var_y_step2," ~ ",i
nput$text_fixed_step2, ", random=", input$text_random_INTERCEPT_step2, ", na.action = input$type
_missing", ", data=data_to_use(),", "method=method_REML_step2() ", ")")})}
  model_full.nlme_2 <- eventReactive(input$Step2_BUTTON,{paste0("lme(",input$var_y_step2," ~ ",i
nput$text_fixed_step2, ", random=", input$text_random_SLOPE_step2,", na.action = input$type_miss
ing", ", data=data_to_use(),", "method=method_REML_step2() ", ")")})


  output$show_equation_step2 <- renderUI({
    HTML(
      "<hr/>",
      paste(p("Equations are:", style = "color:black; font-weight:bold"),
              "Model 0 (without random effects):     ",
              #model_full.nlme_0(),
              paste0("gls(",input$var_y_step2," ~ ",input$text_fixed_step2, ")"),
              "<br/>",
              "Model 1 (random intercept model):     ",
              #model_full.nlme_1(),
              paste0("lme(",input$var_y_step2," ~ ",input$text_fixed_step2, ", random=", input$
text_random_INTERCEPT_step2, ")"),
              "<br/>",
              "Model 2 (random slope (and intercept) model):     ",
              #model_full.nlme_2(),
              paste0("lme(",input$var_y_step2," ~ ",input$text_fixed_step2, ", random=~", input
$text_random_SLOPE_step2, ")"),
              "<br/>",  "<hr/>",
              paste0("Models fit by: ", method_REML_step2()),
              "<br/>",
              paste0("Missing values: na.action = ", input$type_missing)))

  })



  output$compare_step2 <- renderPrint({
    mod0 <- eval(parse(text=model_full.nlme_0()))
    mod1 <- eval(parse(text=model_full.nlme_1()))
    mod2 <- eval(parse(text=model_full.nlme_2()))

    sum_model <- paste("AIC(", "mod0,", "mod1,", "mod2", ")")
    res_sum_model <- eval(parse(text=sum_model))
    res_sum_model <- res_sum_model[order(res_sum_model[,2]),]
    res_sum_model
    })



  ######### STEP 3 ##########
  model_full.nlme_3 <- eventReactive(input$Step3_BUTTON, {paste0("lme(",input$var_y_step3," ~ 1
", ", random=", input$text_random_step3, ", na.action = input$type_missing", ", data=data_to_use
(),", "method=method_REML_step3() ", ")")})
  #Add a single fixed
  model_full.nlme_4 <- eventReactive(input$Step3_BUTTON, {paste0("lme(",input$var_y_step3," ~ ",
input$text_fixed_ALONE_step3, ", random=", input$text_random_step3, ", na.action = input$type_mi
```

```
ssing", ", data=data_to_use(),", "method=method_REML_step3() ", ")")})
  model_full.nlme_5 <- eventReactive(input$Step3_BUTTON, {paste0("lme(",input$var_y_step3," ~ ",
input$text_fixed_sum_step3, ", random=", input$text_random_step3, ", na.action = input$type_miss
ing", ", data=data_to_use(),", "method=method_REML_step3() ", ")")})
  model_full.nlme_6 <- eventReactive(input$Step3_BUTTON, {paste0("lme(",input$var_y_step3," ~ ",
input$text_fixed_interactions_step3, ", random=", input$text_random_step3, ", na.action = input$
type_missing", ", data=data_to_use(),", "method=method_REML_step3() ", ")")})


  output$show_equation_step3 <- renderUI({
    HTML(
      "<hr/>",
      paste(p("Equations are:", style = "color:black; font-weight:bold"),
               "Model 3: basic model: ",
               #model_full.nlme_3(),
               paste0("lme(",input$var_y_step3," ~ 1 ", ", random=", input$text_random_step3, ")
"),
            "<br/>",
               "Model 4: main predictor model: ",
               #model_full.nlme_4(),
               paste0("lme(",input$var_y_step3," ~ ",input$text_fixed_ALONE_step3, ", random=",
input$text_random_step3, ")"),
            "<br/>",
               "Model 5: addition model: ",
               #model_full.nlme_5(),
               paste0("lme(",input$var_y_step3," ~ ",input$text_fixed_sum_step3, ", random=", in
put$text_random_step3, ")"),
            "<br/>",
               "Model 6: interaction model: ",
               #model_full.nlme_6(),
               paste0("lme(",input$var_y_step3," ~ ",input$text_fixed_interactions_step3, ", ran
dom=", input$text_random_step3, ")"),
            "<br/>", "<hr/>",
            paste0("Models fit by: ", method_REML_step3()),
            "<br/>",
            paste0("Missing values: na.action = ", input$type_missing)))

  })


  output$compare_step3 <- renderPrint({
    mod3 <- eval(parse(text=model_full.nlme_3()))
    mod4 <- eval(parse(text=model_full.nlme_4()))
    mod5 <- eval(parse(text=model_full.nlme_5()))
    mod6 <- eval(parse(text=model_full.nlme_6()))

    sum_model <- paste("AIC(", "mod3,", "mod4,", "mod5,", "mod6",  ")")
    res_sum_model <- eval(parse(text=sum_model))
    res_sum_model <- res_sum_model[order(res_sum_model[,2]),]
    res_sum_model
    })




  #### STEP 4 - BEST MODEL ####
  model_full.nlme_step4 <- eventReactive(input$Step4_BUTTON, {paste0("lme(",input$var_y_step4,"
~ ",input$text_fixed_step4, ", random=", input$text_random_step4, ", na.action = input$type_miss
ing", ", data=data_to_use(),", "method=method_REML_step4() ", ")")})

  output$show_equation_step4 <- renderUI({
    HTML(paste("Equation is:",
               paste0("lme(",input$var_y_step4," ~ ",input$text_fixed_step4, ", random=", input$
text_random_step4, ")"),
               "<br/>", "<hr/>",
               paste0("Models fit by: ", method_REML_step4()),
               "<br/>",
               paste0("Missing values: na.action = ", input$type_missing)))

  })
  output$sum_ran_effects <- renderPrint({
    sum_model <- paste("VarCorr(",model_full.nlme_step4(),")")
    eval(parse(text=sum_model))
```

```
  })

  output$sum_fix_effects <- renderPrint({
    sum_model <- paste("summary(",model_full.nlme_step4(),")")
    resumen <- eval(parse(text=sum_model))
    resumen$tTable
  })

  output$show_summary_model_step4 <- renderPrint({
    sum_model <- paste("summary(",model_full.nlme_step4(),")")
    eval(parse(text=sum_model))
  })

  output$anova_model_step4 <- renderPrint({
    sum_model <- paste("anova(",model_full.nlme_step4(),")")
    eval(parse(text=sum_model))
  })

  # r.squaredGLMM()

  output$r.squaredGLMM_step4 <- renderPrint({
    modelo_get <- eval(parse(text=model_full.nlme_step4()))
    r.squaredGLMM(modelo_get)
  })



  ################################ Summary of results ###################

  output$sum_results_step4 <- renderPrint({
    modelo_get <- eval(parse(text=model_full.nlme_step4()))
    intervals(modelo_get, level = 0.95)
  })


  ### PLOTS STEP 4

  output$ran_eff_plot <- renderPlot({
    execute_model <- eval(parse(text=model_full.nlme_step4()))
    plot(ranef(execute_model))
  })

  output$resi_plot_step4 <- renderPlot({
    execute_model <- eval(parse(text=model_full.nlme_step4()))

    Res <- residuals(execute_model, type="pearson")           # if "normalized", the normalized
residuals (standardized residuals pre-multiplied by the inverse square-root factor of the estima
ted error correlation matrix) are used
    Fit <- fitted(execute_model)
    par(mfrow=c(2,2))
    plot(Res ~ Fit, xlab="Fitted values", ylab="Standardized Residuals",
         main="Residuals vs. fitted")
    abline(h=0)

    #formula_boxplot <- as.formula(paste("Res"," ~ ", input$var_fixed_for_boxplot))
    #boxplot(formula_boxplot, ylab="Standardized Residuals", main=paste0(input$var_fixed_for_box
plot), data=data_to_use()[complete.cases(data_to_use()),])

    # Modified

    DB_inter <- cbind(Res, "Var_Classification"=data_to_use()[complete.cases(data_to_use()), inp
ut$var_fixed_for_boxplot])
    formula_boxplot <- as.formula(paste("Res"," ~ ", "Var_Classification"))
    boxplot(formula_boxplot, ylab="Standardized Residuals", main=paste0(input$var_fixed_for_boxp
lot), data=DB_inter)


    abline(h=0, lty=3)
    hist(Res, main="Histogram of residuals", xlab="Residuals")
    qqnorm(Res)
    qqline(Res)
```

```
  })

  output$qqplot_modifiable <- renderPlot({

    execute_model <- eval(parse(text=model_full.nlme_step4()))
    Res <- residuals(execute_model, type="normalized")           # if "normalized", the normaliz
ed residuals (standardized residuals pre-multiplied by the inverse square-root factor of the est
imated error correlation matrix) are used

    QQ_y=qqnorm(Res)
    identify(QQ_y)
    qqline(Res)
  })



  #### Random slopes versus random intercepts

  output$intercep_slope <- renderPrint({
    execute_model <- eval(parse(text=model_full.nlme_step4()))
    coef(execute_model)[1:10,]

  })




  ### Correlation plot in Data inspection

  output$correlation_plot_n <- renderPlot({
    if (sum(sapply(data_to_use(), is.numeric)) <2){
      cat(paste("You have less than TWO continuous variables"))}

    if (sum(sapply(data_to_use(), is.numeric)) >=2){
      my_num_data <- data_to_use()[, sapply(data_to_use(), is.numeric)]
      my_num_data <- my_num_data
      M<-cor(my_num_data, use="complete.obs")
      p.mat <- cor.mtest(my_num_data)
      col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
      corrplot(M, method="color", col=col(200),
             type="upper", order="hclust",
             addCoef.col = "black", # Add coefficient of correlation
             tl.col="black", tl.srt=45, #Text label color and rotation
             # Combine with significance
             p.mat = p.mat[[1]], sig.level = 0.05, insig = "pch", addrect = 3,
             # hide correlation coefficient on the principal diagonal
             diag=FALSE)

    subjects <- sum(complete.cases(my_num_data))
    total <- nrow(my_num_data)
    output$corrtext <- renderText({
      sprintf("The number of subjects without any missing data is: %s out of %s",subjects, total
)

    } )

    }
  })



  output$paired_correlated_vars2 <- renderPrint({
    if (sum(sapply(data_to_use(), is.numeric)) <2){
    cat(paste("You have less than TWO continuous variables"))}

    if (sum(sapply(data_to_use(), is.numeric)) >=2){

    my_num_data <- data_to_use()[, sapply(data_to_use(), is.numeric)]
    my_num_data <- my_num_data
    M<-cor(my_num_data, use="complete.obs")
    matriz_cor <- M
```

```
    for (i in 1:nrow(matriz_cor)){
      correlations <-  which((abs(matriz_cor[i,i:ncol(matriz_cor)]) > 0.7) & (matriz_cor[i,i:nco
l(matriz_cor)] != 1))

      if(length(correlations)> 0){
        lapply(correlations,FUN =  function(x) (cat(paste(colnames(my_num_data)[i], "with", coln
ames(my_num_data)[x]), "\n")))

      }
      }

    if(length(correlations)== 0){
      cat(paste("No paired correlations with r > |0.7|"))
    }
    }
  })









  # Assumptions

  output$influence1 <- renderPrint({
    data_for_model <- data_to_use()
    execute_model <- eval(parse(text=model_full.nlme_step4()))
    influence.model <- influence.ME::influence(execute_model, var_ran_comb)
  })




  #########


  # Multi-collinearity

  # VIF values
  # https://jonlefcheck.net/2012/12/28/dealing-with-multicollinearity-using-variance-inflation-f
actors/
  # https://github.com/mrhelmus/ieco/blob/master/R/vif_Lmer.R

  vif.mer <- function (fit) {
    ## adapted from rms::vif

    v <- vcov(fit)
    nam <- names(fixef(fit))

    ## exclude intercepts
    ns <- sum(1 * (nam == "Intercept" | nam == "(Intercept)"))
    if (ns > 0) {
      v <- v[-(1:ns), -(1:ns), drop = FALSE]
      nam <- nam[-(1:ns)]
    }

    d <- diag(v)^0.5
    v <- diag(solve(v/(d %o% d)))
    names(v) <- nam
    v
  }



  output$VIF_values <- renderPrint({
    execute_model <- eval(parse(text=model_full.nlme_step4()))
    vif.mer(execute_model)

  })
```

```
  output$shapiro_results <- renderPrint({

    execute_model <- eval(parse(text=model_full.nlme_step4()))
    Res <- residuals(execute_model, type="normalized")
    shapiro.test(Res)

  })



  #### COOK's distance

  # NOT USED; we would have to use lme4!

  #output$cooksd <- renderPlot({
  #execute_model <- eval(parse(text=model_full.nlme_step4()))
  #cooksd <- CookD(execute_model)
  #})

  #output$cooksd_text <- renderPrint({
  #modelo <- lmer(model_full(), data=data_to_use(), REML=input$REML)
  #cooksd <- cooks.distance(modelo)
  #influential <- as.numeric(names(cooksd)[(cooksd > 4 * mean(cooksd, na.rm = TRUE))])  # influe
ntial row numbers
  #influential
  #head(data()[influential, ])  # influential observations.
  #})







  ##### Download data from LMM

  df_LMM_download <- reactive({
    sum_model <- paste("summary(",model_full.nlme_step4(),")")
    resi_model <- paste("residuals(",model_full.nlme_step4(),")")
    #resumen <- eval(parse(text=sum_model))
    #RESIDUALS <- eval(parse(text=resi_model))

    resu <- data.frame(eval(parse(text=sum_model))$data)
    colnames(resu)[1]<-c("ID")

    fitted_values <- data.frame(eval(parse(text=sum_model))$fitted)

    residuals_values <- eval(parse(text=resi_model))
    row_residuals <- names(residuals_values)
    residuals_comb <- data.frame(cbind("row_residuals"=as.numeric(row_residuals), "residuals"=re
siduals_values))

    #fitted_res <- data.frame(cbind(residuals_comb, fitted_values))
    fitted_res <- do.call(cbind, list(residuals_comb, fitted_values))
    fitted_res$row_residuals <- as.numeric(fitted_res$row_residuals)

    resu2 <- resu[complete.cases(resu),]
    definitive_data <- cbind(resu2, fitted_res[1:dim(resu2)[1],])
    resu3 <- resu[complete.cases(resu)==FALSE,]

    library(data.table)
    definitive_data2 <- rbindlist(list(definitive_data,resu3), fill=TRUE)
    newdata <- definitive_data2[order(definitive_data2[,1]),]
    newdata <- data.frame(newdata)
    return(newdata)
  })


  vars_for_LMM_download <- reactive({
```

```
    variab <- names(df_LMM_download())
    return(variab)
  })


  observe({
    updateSelectInput(session, inputId = 'variables_LMM_download', label = 'Select variables to
download',
                      choices = vars_for_LMM_download(), selected = vars_for_LMM_download())
  })


  df_LMM_to_download <-reactive({
    df_to_down <- df_LMM_download()[,input$variables_LMM_download]
  })


  output$download_data_LMM <- downloadHandler(
    filename = function() {
      paste("DATA_", "SISSREM_", Sys.Date(), ".csv", sep="")
    },
    content = function(file) {
      write.csv(df_LMM_to_download(), file, row.names=FALSE)
    }
  )




  ################################################################################ GENERATE FULL REPO
RT ################################################################################



  # PDF

  output$downloadReport_PDF <- downloadHandler(
    filename = function() {
      paste('my_SISSREM_report_', Sys.Date(), sep = '', switch(
        'PDF', PDF = '.pdf', HTML = '.html', Word = '.docx'
      ))
    },

    content = function(file) {
      src <- normalizePath('report.Rmd')

      # temporarily switch to the temp dir, in case you do not have write
      # permission to the current working directory
      owd <- setwd(tempdir())
      on.exit(setwd(owd))
      file.copy(src, 'report.Rmd', overwrite = TRUE)

      library(rmarkdown)
      out <- render('report.Rmd', switch(
        'PDF',
        PDF = pdf_document(), HTML = html_document(), Word = word_document()
      ))
      file.rename(out, file)
    }
  )


  # HTML

  # This module is different compared to PDF and DOC cause HTML syntax!
  output$downloadReport_HTML <- downloadHandler(
    filename = function() {
      paste('my_SISSREM_report_', Sys.Date(), sep = '', '.html')
    },
```

```
    content = function(file) {
      src <- normalizePath('report.Rmd')

      # temporarily switch to the temp dir, in case you do not have write
      # permission to the current working directory
      owd <- setwd(tempdir())
      on.exit(setwd(owd))
      file.copy(src, 'report.Rmd', overwrite = TRUE)

      library(rmarkdown)
      out <- render('report.Rmd')
      file.rename(out, file)
    }
  )


  # Microsoft Word

  output$downloadReport_WORD <- downloadHandler(
    filename = function() {
      paste('my_SISSREM_report_', Sys.Date(), sep = '', switch(
        'Word', PDF = '.pdf', HTML = '.html', Word = '.docx'
      ))
    },

    content = function(file) {
      src <- normalizePath('report.Rmd')

      # temporarily switch to the temp dir, in case you do not have write
      # permission to the current working directory
      owd <- setwd(tempdir())
      on.exit(setwd(owd))
      file.copy(src, 'report.Rmd', overwrite = TRUE)

      library(rmarkdown)
      out <- render('report.Rmd', switch(
        'Word',
        PDF = pdf_document(), HTML = html_document(), Word = word_document()
      ))
      file.rename(out, file)
    }
  )




  # EMAIL-CONTACT

  url <- a("pablo1280@gmail.com - sissrem@gmail.com", href="pablo1280@gmail.com - sissrem@gmail.
com")
  output$tab <- renderUI({
    h5(tagList("CONTACT:", url))
  })



  ## VIDEO YOUTUBE

  output$video <- renderUI({
    click <- input$plot_click
    if(!is.null(click)){
      link = cases(
        "Gyrfsrd4zK0" = click$x > 40,
        "b518URWajNQ" = click$x > 20,
        "I5Z9WtTBZ_w "= click$x > 0
      )
      HTML(paste0('<iframe width="200" height="100" src="https://www.youtube.com/watch?v=DpHEpyi
tzUQ', link ,'" frameborder="0" allowfullscreen></iframe>'))
    }
  })
```

```
  session$onSessionEnded(stopApp) # To stop Shiny app if the browser tab is closed


}                               #### CLOSE SERVER ####




################################################################################################################
#####################################
############################################################ Create Shiny app object ###############
#############################################################
################################################################################################################
#####################################

# enableBookmarking(store = "url")
shinyApp(ui = ui, server = server)
```

9.3 Annex 3. M4 - Rmarkdown report file (to be compiled as **report.Rmd** in combination with **app.R**)

```
---
title: ``r input$report_name`'
author: ``r input$report_author`'
date: ``r format(Sys.time(), "%d-%m-%Y")`'
geometry: left=2cm,right=2cm,top=1.5cm,bottom=1.5cm
output:
  html_notebook:
    df_print: paged
    toc: yes
    toc_float: true
    theme: united
    highlight: tango
  pdf_document:
    keep_tex: yes
    toc: yes
    df_print: kable
    highlight: zenburn
  html_document:
    df_print: paged
    toc: yes
    toc_float: true
    theme: united
    highlight: tango
toc: true
toc_depth: 3
---


```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)


```
```

````
\newpage

# 0. Report at a glance

This report consists of the analysis (tests and graphs) of repeated measures data using linear-m
ixed model (LMM) implemented in SISSREM. It is divided into the different sections that should b
e taken into account in the context of performing a repeated measures analysis.

We will be really glad about receiving any suggestions for improving our Report.

Please, contact sissrem@gmail.com or pablo1280@gmail.com



\newpage



# 1. Selection of data



This section describes the variables included in the data for analysis.



## 1.1 Descriptives of data



## 1.1.1 Uploaded data



This is the initial dataset that uploaded or loaded into the program:



```{r uploaded_data, echo=FALSE}

str(data())

```

## 1.1.2 Data analysed

This is the dataset that you have used for conducting the analysis



```{r used_data, echo=FALSE}

str(data_to_use())

```

## 1.2 Transformation of data



Data usually needs to be transformed before applying a statistical procedure. In this applicatio
n, we allow the user to transform the "Y" variable after conducting a Box-Cox to determine the o
ptimum transformation. Moreover, SISSREM also allows the user to transform the "X" predictors.

```{r, echo=FALSE, results='asis'}

if (input$trans_box_buttom==TRUE) {cat("You have transformed your Y variable:", input$var_box_co
x, "by applying ", input$type_transformation,". ")}

if (input$trans_box_buttom==FALSE) {cat("You have not transformed your Y variable")}
````
````

```
if (input$trans_cent_scale_box_buttom==TRUE) {cat("You have applied", input$center_scale, "to th
e following variables", input$var_center_scale)}

if (input$trans_cent_scale_box_buttom==FALSE) {cat("You have not further transformed your variab
les.")}

```


\newpage

1.3 Exploratory Data Analysis (EDA)

As in all data analysis, it is advisable to examine the data before embarking upon statistical m
odeling. Exploratory Data Analysis refers to the critical process of performing initial investig
ations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check as
sumptions with the help of summary statistics and graphical representations.

Remember that a full EDA report is also available as an independent module in SISSREM.

1.3.1 Variables

The very first thing that we want to do in your EDA is checking the dimension of the input datas
et and the time of variables.

With that, we can see we've got some Continuous variables and some Categorical variables.


```{r df_sum, echo=FALSE, warning=FALSE, results='asis'}

library(summarytools)

#knitr::is_html_output()

# HTML

if (knitr::is_html_output()==TRUE) {

  print(dfSummary(data_to_use(), plain.ascii = FALSE, varnumbers = FALSE, style = 'grid', graph.
magnif = 0.75))

}

# PDF or DOCX

if (knitr::is_html_output()==FALSE) {

  dfSummary(data_to_use(), plain.ascii = FALSE, varnumbers = FALSE, style = 'grid', graph.magnif
= 0.75)

}

  #dfSummary(data_to_use(), style = "multiline", plain.ascii = TRUE, varnumbers = FALSE, valid.c
ol = FALSE, tmp.img.dir = "./img")

```


\newpage

1.3.2 Missing values

````
Missing values are of important concern in data. LMM can deal with missing values, which is a hu
ge advantage compared to other alternatives such as repeated measures ANOVA.

Here is a summary of the number of missing values in your data

```{r missings, echo=FALSE, warning=FALSE}

plot_missing(data_to_use())

```

\newpage



## 1.3.3 Histogram of Continuous Variables



```{r histo, echo=FALSE, warning=FALSE}



plot_histogram(data_to_use())



```

\newpage

## 1.3.4 Correlation plots

Correlation between variables may generate several problems in the final LMM. Therefore, it is i
mportant to assess the correlation between the continuous variables before including them into t
he model.

```{r correlations, echo=FALSE, warning=FALSE}

plot_correlation(data_to_use(), type = 'continuous')

```

\newpage



## 1.3.5 Barplots of categorical data

```{r barplot, echo=FALSE, warning=FALSE}

testeo <- lapply(data_to_use(), is.factor)

testeo <- unlist(testeo)

testeo2 <- as.factor(testeo)

if (levels(testeo2)=="FALSE")

{

  cat("There are no categorical variables in your data")

}

if (length(levels(testeo2)) == 2 | levels(testeo2)=="TRUE")

{

  plot_bar(data_to_use())

}
````
````

```

\newpage

## 1.3.6 Boxplots

Boxplots by the time variable (used as factor)

```{r boxplot, fig.width = 10, fig.height = 10, fig.fullwidth=TRUE}

plot_boxplot(data_to_use(), by = input$var_fixed_for_boxplot)

```

\newpage

2. Linear Mixed Model

In this section, results regarding the LMM that you have performed will be shown.

2.1 Step 0. Base models

```{r, echo=FALSE, results='asis'}

if (input$clic_use_step0==TRUE) {    #

  cat("Let's check the results for the Step 0.","  \n\n")

  cat("In this initial step, every predictor (fixed variables (FV)) will be included in an indep
endent model together with the variable of TIME and the random variable (RV) assuming a random i
ntercept model. The user should select those FVs with a P-value < 0.2 to be included in the next
step.")

}

if (input$clic_use_step0==FALSE) {cat("You have not run Step 0", "  \n")}

```

```{r, echo=FALSE, results='asis'}

if (input$clic_use_step0==TRUE) {

  results_list_model  <- list()

  for (i in 1:(length(input$var_fixed_step0))){

    cat("These are the models evaluated:", "  \n")

    cat(input$var_y_step0," ~ ", input$var_fixed_step0[i], " + ",input$var_time_step0,

        ", random=", input$text_random_step0, ".\n")

  }

  #data.frame("MODELS"=unlist(results_list_model))


}

```
```

```{r, echo=FALSE}

if (input$clic_use_step0==TRUE) {

  cat("These are the results for each FV and TIME:", "  \n")

  cat("  \n\n")

  results_list  <- list()

    for (i in 1:(length(input$var_fixed_step0))){

    sum_model <- paste("anova(",model_full.nlme_step0_1()[i],")")

    results_list[[i]] <- eval(parse(text=sum_model))

  }



 results_list

  }
```

```{r, echo=FALSE, results='asis'}

if (input$clic_use_step0==TRUE) {

cat("Importantly, remember not to add variables that are highly correlated (r > 0.70 based on Do
rmann et al., 2013) because when you include correlated variables into a parametreic model it wi
ll have a problem of coefficients' stability as it will lose accuracy in determining the coeffic
ients. In case that you have correlation between variables, remember to choose one of the correl
ated variables using biological knowledge/reasoning to select the most meaningful variable or co
nduct a dimension-reduction analysis (e.g. Principal Components Analysis) leaving a single varia
ble that accounts for most of the shared variance among the correlated variables", "  \n")



    my_num_data <- data_to_use()[, sapply(data_to_use(), is.numeric)]

    my_num_data <- my_num_data



    M<-cor(my_num_data, use="complete.obs")

    p.mat <- cor.mtest(my_num_data)



    col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))



    subjects <- sum(complete.cases(my_num_data))

    total <- nrow(my_num_data)



    sprintf("The number of subjects without any missing data is: %s out of %s",subjects, total)



     cat("  \n\n")
```

```
    corrplot(M, method="color", col=col(20),

             type="upper", order="hclust",

             addCoef.col = "black", # Add coefficient of correlation

             tl.col="black", tl.srt=45, #Text label color and rotation

             # Combine with significance

             p.mat = p.mat[[1]], sig.level = 0.05, insig = "pch", addrect = 3,

             # hide correlation coefficient on the principal diagonal

             diag=FALSE)


}
```

```{r, echo=FALSE, results='asis'}
if (input$clic_use_step0==TRUE) {


    my_num_data <- data_to_use()[, sapply(data_to_use(), is.numeric)]

    my_num_data <- my_num_data

    M<-cor(my_num_data, use="complete.obs")

    matriz_cor <- M


    for (i in 1:nrow(matriz_cor)){

      correlations <-  which((abs(matriz_cor[i,i:ncol(matriz_cor)]) > 0.7) & (matriz_cor[i,i:nco
l(matriz_cor)] != 1))


      if(length(correlations)> 0){

        lapply(correlations,FUN =  function(x) (cat(paste(colnames(my_num_data)[i], "with", coln
ames(my_num_data)[x]), "\n")))


      }

    }


    if(length(correlations)== 0){

      cat(paste("No paired correlations with r > |0.7|"))

    }


}
```

````
\newpage

## 2.2 Step 1. Definition of the 'Beyond Optimal Model'

```{r, echo=FALSE, results='asis'}

if (input$clic_use_step1==TRUE) {

  cat("Let's check the results for the Step 1.","  \n\n")

  cat("In this step you have to choose all the fixed terms that you want to consider and interac
tions, together with the main random term (different subjects where the measures are repeated: 1
|ID). Remember not to add variables that are highly correlated (r > 0.70)! ")

}

if (input$clic_use_step1==FALSE) {cat("You have not run Step 1", "  \n")}

```

```{r, echo=FALSE, results='asis'}

if (input$clic_use_step1==TRUE) {



cat("Beyond Optimal Model", "  \n\n")

    cat("Equation is:", "  \n",

              paste0("lme(",input$var_y," ~ ",input$text_fixed_step1,", random=", input$text_ran
dom_step1, ")"),

                "  \n\n",

              paste0("Model fit by: ", method_REML()),

              "  \n",

              paste0("Missing values: na.action = ", input$type_missing)



      )




}
```

Results from the model are:



```{r, echo=FALSE}

if (input$clic_use_step1==TRUE) {

sum_model <- paste("summary(",model_full.nlme_estruc1(),")")

    eval(parse(text=sum_model))

}



```
````

ANOVA of the FIXED effects:

```{r, echo=FALSE}

if (input$clic_use_step1==TRUE) {

    sum_model <- paste("anova(",model_full.nlme_estruc1(),")")

    eval(parse(text=sum_model))



}



```

\newpage

2.3 Step 2. Structure of RANDOM effects

```{r, echo=FALSE, results='asis'}

if (input$clic_use_step2==TRUE) {

  cat("Let's check the results for the Step 2.","  \n\n")

  cat("In this step we are going to find a proper structure for the RANDOM component of the LMM. We will use the best 'Beyond optimal model' and we will vary the RANDOM component leaving the sa me FIXED component. When model fits are ranked according to their AIC values, the model with the lowest AIC value being considered the 'best'. REML method is used to properly compare the models using AIC")

}



if (input$clic_use_step2==FALSE) {cat("You have not run Step 2", "  \n")}

```

```{r, echo=FALSE, results='asis'}

if (input$clic_use_step2==TRUE) {

  cat("Structure of RANDOM components", "  \n")

    cat("Equations are:", "  \n",

            "Model 0 (without random effects):     ",

      paste0("gls(",input$var_y_step2," ~ ",input$text_fixed_step2, ")"),

       "  \n",

      "Model 1 (random intercept model):     ",

      paste0("lme(",input$var_y_step2," ~ ",input$text_fixed_step2, ", random=", input$text_rand
om_INTERCEPT_step2, ")"),

       "  \n",

      "Model 2 (random slope (and intercept) model):     ",

      paste0("lme(",input$var_y_step2," ~ ",input$text_fixed_step2, ", random=~", input$text_ran
dom_SLOPE_step2, ")"),
```

```
     "  \n\n",
                paste0("Models fit by: ", method_REML_step2()),
     "  \n",
                paste0("Missing values: na.action = ", input$type_missing)
      )
  }


```


```{r, echo=FALSE}
if (input$clic_use_step2==TRUE) {


cat("MODEL comparison based on AIC", "  \n")
    mod0 <- eval(parse(text=model_full.nlme_0()))
    mod1 <- eval(parse(text=model_full.nlme_1()))
    mod2 <- eval(parse(text=model_full.nlme_2()))


    sum_model <- paste("AIC(", "mod0,", "mod1,", "mod2", ")")


    eval(parse(text=sum_model))


}


```
\newpage

2.4 Step 3. Structure of FIXED effects


```{r, echo=FALSE, results='asis'}
if (input$clic_use_step3==TRUE) {
  cat("Let's check the results for the Step 3.","  \n\n")
  cat("In this step we are going to find a proper structure for the FIXED component of the LMM).
We will use the best model from STEP2 and we will vary the FIXED component leaving the same opti
mized RANDOM component from STEP2.

When model fits are ranked according to their AIC values, the model with the lowest AIC value be
ing considered the 'best'. ML method is used to properly compare the models using AIC.")
}
```

```
if (input$clic_use_step3==FALSE) {cat("You have not run Step 3", "  \n")}
```

```{r, echo=FALSE, results='asis'}
if (input$clic_use_step3==TRUE) {

  cat("Structure of FIXED effects", "  \n")

    cat("Equations are:", "  \n",



      "Model 3: basic model",

       paste0("lme(",input$var_y_step3," ~ 1 ", ", random=", input$text_random_step3, ")"),



       "  \n",

      "Model 4: main predictor model",

       paste0("lme(",input$var_y_step3," ~ ",input$text_fixed_ALONE_step3, ", random=", input$te
xt_random_step3, ")"),

       "  \n",

       "Model 5: addition model",

       paste0("lme(",input$var_y_step3," ~ ",input$text_fixed_sum_step3, ", random=", input$text
_random_step3, ")"),

       "Model 6: interaction model",

       paste0("lme(",input$var_y_step3," ~ ",input$text_fixed_interactions_step3, ", random=", i
nput$text_random_step3, ")"),



     "  \n\n",

              paste0("Models fit by: ", method_REML_step3()),

     "  \n",

              paste0("Missing values: na.action = ", input$type_missing)



      )


}
```


```{r, echo=FALSE}
if (input$clic_use_step3==TRUE) {


cat("MODEL comparison based on AIC", "  \n")
```

```
    mod3 <- eval(parse(text=model_full.nlme_3()))

    mod4 <- eval(parse(text=model_full.nlme_4()))

    mod5 <- eval(parse(text=model_full.nlme_5()))

    mod6 <- eval(parse(text=model_full.nlme_6()))

    sum_model <- paste("AIC(", "mod3,", "mod4,", "mod5,", "mod6",  ")")

    eval(parse(text=sum_model))

}



```


\newpage

2.5 Step 4. Final model

```{r, echo=FALSE, results='asis'}

if (input$clic_use_step4==TRUE) {

  cat("Let's check the results for the Step 4.","  \n\n")

  cat("In this step we are going to generate the FINAL model based on previous steps. REML metho
d is used to properly report the FINAL model.", "  \n\n")

      cat(paste("Equation is:",

                paste0("lme(",input$var_y_step4," ~ ",input$text_fixed_step4, ", random=", input$
text_random_step4, ")"),

     "  \n\n",

                paste0("Models fit by: ", method_REML_step4()),

     "  \n",

                paste0("Missing values: na.action = ", input$type_missing)))

}



if (input$clic_use_step4==FALSE) {cat("You have not run Step 4", "  \n")}

```

2.5.1 SUMMARY OF THE DATA

2.5.1.1 RANDOM EFFECTS

```{r, echo=FALSE}

if (input$clic_use_step4==TRUE) {

    sum_model <- paste("VarCorr(",model_full.nlme_step4(),")")

    eval(parse(text=sum_model))

}

```

2.5.1.2 FIXED EFFECTS - ANOVA BASED ON T-VALUE

```{r, echo=FALSE}
if (input$clic_use_step4==TRUE) {
    sum_model <- paste("summary(",model_full.nlme_step4(),")")
    resumen <- eval(parse(text=sum_model))
    resumen$tTable
}
```

2.5.2 OVERALL OUTPUT FROM THE MODEL

2.5.2.1 FULL SUMMARY

```{r, echo=FALSE}
if (input$clic_use_step4==TRUE) {
    sum_model <- paste("summary(",model_full.nlme_step4(),")")
    eval(parse(text=sum_model))
}
```

2.5.2.2 ANOVA BASED ON F-TEST

```{r, echo=FALSE}
if (input$clic_use_step4==TRUE) {
    sum_model <- paste("anova(",model_full.nlme_step4(),")")
    eval(parse(text=sum_model))
}
```

2.5.2.3 R-SQUARED (based on Nakagawa and Schielzeth (2013))

R2m (Marginal R2 (whole model)): it is the marginal R2 for a linear mixed model, meaning that it is concerned with the variance explained by the fixed factors.

R2c (Conditional R2 (whole model)): it is the conditional R2 for a linear mixed model, meaning t hat it is concerned with the variance explained by the fixed and random factors.

```{r, echo=FALSE}
if (input$clic_use_step4==TRUE) {
    modelo_get <- eval(parse(text=model_full.nlme_step4()))
    r.squaredGLMM(modelo_get)
}
```

2.5.2.4 95% CONFIDENCE INTERVALS FOR THE COEFFICIENTS

```{r, echo=FALSE}

if (input$clic_use_step4==TRUE) {

    modelo_get <- eval(parse(text=model_full.nlme_step4()))

    intervals(modelo_get, level = 0.95)

}

```

\newpage

3. Assumptions of LMM

3.1 Normality, linearity and Homoscedasticity

We need to check that the assumptions are met:

1) No pattern should be seen in the plots: 'Fitted values vs Standardized residuals' (top left p
lot) and 'Explanatory variable vs Standardized residuals' (top right plot).

1.a) If we observe increasing or decreasing values in 'Fitted values vs Standardized residuals',
it inform us about Heteroscedasticity (i.e. variance of data not approx. equal across range of p
redicted values) that we should correct (try box-cox transformation alternatives).

2) Histogram of residuals (bottom left) must follow a Gaussian distribution, otherwise it will i
nform us about non-normality behaviour.

3) QQ-Plot (bottom right) must show dots aligned to the diagonal line, otherwise it will inform
us about non-normality.

```{r, echo=FALSE, results='asis'}

if (input$clic_use_step4==TRUE) {

    execute_model <- eval(parse(text=model_full.nlme_step4()))

    Res <- residuals(execute_model, type="normalized")           # if "normalized", the normaliz
ed residuals (standardized residuals pre-multiplied by the inverse square-root factor of the est
imated error correlation matrix) are used

    Fit <- fitted(execute_model)

    par(mfrow=c(2,2))

    plot(Res ~ Fit, xlab="Fitted values", ylab="Standardized Residuals",

         main="Residuals vs. fitted")

    abline(h=0)

    #formula_boxplot <- as.formula(paste("Res"," ~ ", input$var_fixed_for_boxplot))

    #boxplot(formula_boxplot, ylab="Standardized Residuals", main=paste0(input$var_fixed_for_box
plot), data=data_to_use()[complete.cases(data_to_use()),])

    DB_inter <- cbind(Res, "Var_Classification"=data_to_use()[complete.cases(data_to_use()), inp
ut$var_fixed_for_boxplot])

    formula_boxplot <- as.formula(paste("Res"," ~ ", "Var_Classification"))

    boxplot(formula_boxplot, ylab="Standardized Residuals", main=paste0(input$var_fixed_for_boxp
lot), data=DB_inter)
```

```
    abline(h=0, lty=3)
    hist(Res, main="Histogram of residuals", xlab="Residuals")
    qqnorm(Res)
    qqline(Res)
}
```

\newpage

4) Shapiro-Wilk analysis of residuals

**** If P-value < 0.05 --> Non-normality

```{r, echo=FALSE}
if (input$clic_use_step4==TRUE) {
    execute_model <- eval(parse(text=model_full.nlme_step4()))
    Res <- residuals(execute_model, type="normalized")
    shapiro.test(Res)
}
```

5) Multi-collinearity

```{r, echo=FALSE}
if (input$clic_use_step4==TRUE) {
    execute_model <- eval(parse(text=model_full.nlme_step4()))
    vif.mer(execute_model)
}
```

4. Getting information from the model

4.1 A) Plot of random effects

```{r, echo=FALSE}
if (input$clic_use_step4==TRUE) {
    execute_model <- eval(parse(text=model_full.nlme_step4()))
    plot(ranef(execute_model))
 }
```

4.2 B) Predicted values for the data

B1) If the structure is 1|RV then you are constructing a RANDOM INTERCEPT MODEL:

Each random variable (subject) is assigned a different intercept cause by-RV (by subject) variab
ility is taken into account. However, the fixed effects are all the same for all RVs.

In this model, we account for baseline-differences in the dependent variable, but we assume that
whatever the effect of the fixed variables, it is the same for all the RVs.

B2) If the structure is FX|RV then you are constructing a RANDOM SLOPE MODEL:

In this case RVs are not only allowed to have differing intercepts, but they are also allowed to
have different slopes for the effect of FVs.

```{r, echo=FALSE}

if (input$clic_use_step4==TRUE) {

    execute_model <- eval(parse(text=model_full.nlme_step4()))

    coef(execute_model)[1:10,]

  }

```

SISSREM: Shiny Interactive, Supervised and Systematic report from REpeated Measures data - 2019
- Beta Version 1.0.

9.4 Annex 4. Final report in .PDF format using the example database

Included as an independent .PDF file.

9.5 Annex 5. Congress abstract – Oral communication

SISSREM: Shiny Interactive, Supervised and Systematic report from REpeated Measures data.

Pablo Hernández-Alonso[1], *Núria Pérez Álvarez*[2]

[1]pablo.hernandez@fimabis.org, Instituto de Investigación Biomédica de Málaga (IBIMA), Málaga University, Málaga, Andalucía, Spain; Human Nutrition Unit, Rovira i Virgili University, Reus, Catalonia, Spain.

[2]nperez@flsida.org, Department of Statistics and Operations Research, Technical University of Catalonia-Barcelona Tech. Fight against AIDS Foundation, HUGTIP.

Longitudinal methods are the procedures of choice for scientists who view their phenomena of interest as dynamic. Linear mixed models (LMM) can be used to describe relationships across time in a longitudinal dataset successfully leading with dependent observations and adding the flexibility of random effects. However, other statistical methods such as the repeated measures ANOVA can perform analysis for dependent observations, but their limitations lead to misuse by part of researchers from biomedical areas due to its statistical simplicity compared with LMM. We have developed a Shiny R code-based application (SISSREM, Shiny Interactive, Supervised and Systematic report from REpeated Measurements data) intended to be used by users in biomedical areas with low-to-medium skills in statistics. Our Shiny app is able to: i) instruct the user in the understanding of an LMM analysis with an example database; ii) allow the user to analyse their own data; and iii) allow the user to create an interactive, supervised and systematic *Rmarkdown* report to be exported from the Shiny app. The main core of the application consists of a guided walk through a default analysis with an example database and the systematic decisions that must be performed in an LMM analysis. Therefore, we have structured the app into three main modules according to their application: i) exploratory data analysis (EDA) to gain insight into data; ii) graphic module to check the relationship between the variables of interest; iii) fitting module to perform the LMM together with evaluating significance of the constructed LMM (e.g. likelihood ratio test). This application will be published online by the end of June. Importantly, its code will be accessible in order to be updated or adapted for other purposes.
SISSREM is a functional Shiny application which is intended to spread the usefulness of LMM into the biomedical research area.

Keywords: SISSREM, linear-mixed model; longitudinal data; shiny app.
AMS: 62M10 Time series, Auto-correlation, Regression, etc.