The University Of Sheffield.

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

IDEAL

Statistical Analysis of Series of N-of-1 Trials

Using SAS

Artur Araujo

September 2018

Acknowledgements

I would like to thank especially Dr. Stephen Senn for giving me the opportunity to work on the IDEAL project with him. I would like to acknowledge with thanks the helpful suggestions made by Dr. Stephen Senn and Dr. Steven A. Julious, during the revision of this document. Many thanks for the Luxembourg Institute of Health for receiving me so well during the first three years of my postgraduate studies and research. I would like to thank Boehringer Ingelheim GmbH for having paid my tuition fees at the University of Sheffield for the first three academic years. Special thanks to the Luxembourgish taxpayers and the Luxembourgish State through the CEDIES (Centre de Documentation et d'Information sur l'Enseignement Supérieur) for providing financial aid during my postgraduate studies at the Luxembourg Institute of Health and at the University of Sheffield. This project has received funding from the European Union’s Seventh Framework Programme for research, technological development, and demonstration under grant agreement no 602552 (IDEAL - Integrated Design and Analysis of small population group trials).

Abstract

The statistical analysis of series of n-of-1 trials in which the treatments are randomized in cycles within individuals is described. Version 9.4 (TS1M3) of the SAS software is used for the statistical analysis. The statistical methods demonstrated assume that the outcome variable is normally distributed. The examples presented are for a comparison between two treatments. Several methods appropriate for the estimation and inference on the overall treatment effect as well as on the individual treatment effects are compared and discussed.

Table of Contents

1 Defining directories

Before starting any statistical analysis within the SAS software, it is often useful to define the working directory. The advantage of this is that relative paths instead of full paths can be used to tell the SAS software where any file is located. A very convenient and portable way is to let the SAS software find the path to the running script and set this path as working directory. Afterwards the path to any file involved in the analysis can be defined relative to the one where the running script is located. This way if you give your SAS programs to anyone else it is expected that they will run as intended.

Program 1.1: Defining the working directory.

```
* Log current folder ;
%xlog(cd);

* Determine path to script ;
%let ScriptDir=%sysfunc(
        tranwrd(
                %sysget(SAS_EXECFILEPATH),
                %sysget(SAS_EXECFILENAME),
        )
);

* Set current folder ;
x cd "&ScriptDir";

* Log current folder ;
%xlog(cd);
```

For informative purposes, the first statement in Program 1.1 outputs the full path to the current working directory to the SAS Log. In the statement that follows the 'SAS_EXECFILEPATH' and 'EXECFILENAME' environment variables are queried and the running script's path is determined from these variables using appropriate macro functions. In the end, the full path to the running script is stored in the "ScriptDir" macro variable. The next step effectively sets the working directory by using the 'x' statement to execute the operating system 'cd' command followed by the value stored in the "ScriptDir" macro variable. The SAS 'x' statement does not output any error or success message to the SAS Log. For this reason, another log of the working directory is requested. If the result is the correct full path to the running script than the operating system command succeeded, otherwise it did not. After submitting the code from Program 1.1 to the SAS system, the following text is written to the SAS Log.

```
1    * Log current folder ;
2    %xlog(cd);
E:\IDEAL_SAS\import\normal
3
4    * Determine path to script ;
5    %let ScriptDir=%sysfunc(
6        tranwrd(
7            %sysget(SAS_EXECFILEPATH),
8            %sysget(SAS_EXECFILENAME),
9        )
10   );
11
12   * Set current folder ;
13   x cd "&ScriptDir"
13 !                 ;
14
15   * Log current folder ;
16   %xlog(cd);
E:\IDEAL_SAS\import\normal
```

The SAS Log informs the user that the full path to the working directory is “E:\IDEAL_SAS\import\normal” before and after defining it. Different results might be obtained depending on the operating system and on the location of the running script.

2 Defining local library and file reference

In this document, the statistical analysis of series of n-of-1 trials where the data is assumed normally distributed is demonstrated by using the SAS software. At first, a reference to a local library for storing permanent SAS datasets is defined. Then a reference to an external file containing the data is defined. Afterwards the external data file is imported and other datasets are created from it by performing several operations on its variables and observations. The SAS base software provides at least two ways of performing operations on datasets, one being data steps and the other being 'proc sql'[1-3]. The later SAS procedure implements the Structured Query Language, a very intuitive programming language designed for managing data. Some programming problems are easier to solve with data steps while other problems are easier to solve with SQL. Some SAS programmers might prefer data steps while others might prefer SQL. Generally, SQL is considered more efficient than data steps. Solutions involving both data steps and SQL are demonstrated for most cases approached throughout this document.

Working with the SAS software often involves importing data from external formats or hardcoding the data directly into a SAS program. Operations on a dataset's variables might be required to proceed with the statistical analysis. The SAS software saves the data to a so-called 'WORK' library by default. Data stored in the 'WORK' library resides in the computer's memory and is lost when the SAS system is terminated. Data stored in the computer's storage devices is maintained even between reboots and shutdowns. Within the SAS software, local libraries can be defined. Any dataset saved in local libraries is permanent. The next program demonstrates the definition of a local library for storage of permanent SAS datasets.

Program 2.1: Defining a local library.

```
libname NORMAL "..\library\normal";
```

The 'libname' statement in Program 2.1 tells SAS to refer to the path "..\library\normal" by the shorter "NORMAL" version. Here a path relative to the location of the running script is used. By defining a local library, an often-longer file system path can be referred by a shorter word chosen by the programmer. This makes it easier for the programmer when a file system path must be referenced many times in a single SAS program. The usage of a library reference can be better understood later as more SAS programs are presented. The purpose of this library, is to store the datasets pertaining to the statistical analysis of normal data collected from series of n-of-1 trials.

As with directory paths, file paths can be referenced by shorter words. File references in SAS can be created through a 'filename' statement as in the following program.

Program 2.2: Defining a file reference.

```
filename nof1 DISK '.. \data\N of 1 example.csv';
```

In this simple ‘filename’ statement the file “N of 1 example.csv” located in the folder ‘’.. \data” is associated to the much smaller word “nof1”. Next time this file needs to be accessed, the smaller word can be used as reference instead of the much longer path to the file.

3 Including macros

The SAS macro language was used to write several macros that are very useful for the analysis of series of n-of-1 trials [4, 5]. The usage of the macros developed for analysing series of n-of-1 trials is demonstrated in the sections below. For the reader and user's convenience, the macros are written in separate files. All the SAS code presented in this document is written across several files, which are provided with the document itself to the reader and potential user's benefit. Before using the macros, you need to include the files containing the actual code for the macros in your SAS programs. The following program demonstrates the inclusion of external files.

Program 3.1: Including macros written in external files.

```
* Define local macros folder ;
filename MACROS '.. \macros';

* Include macros ;
%include MACROS('*.sas');
```

In the SAS programming language text written between an asterisk (*) and a semicolon (;) is considered a comment and is not interpreted. An alternative way of writing comments is to place the text between a forward slash followed by an asterisk (/*) and an asterisk followed by a forward slash (*/). Programs presented in this document contain both types of comment. Not counting with the two comment statements present in Program 3.1, there are two statements of code in the program, which are in fact interpreted by the SAS software. The 'filename' statement defines a reference named "MACRO" to a folder in the local machine. A path relative to the location of the running script is used to define the folder. The '%include' statement includes all the files with extension ".sas" located inside the folder defined in the previous statement in the "MACRO" reference. After submitting the code in Program 3.1, the macros written in the included files can be used, and the code can be submitted for interpretation by the software.

4 Importing and handling data using data steps

Importing data from external files to the SAS software can be accomplished through a data step by means of appropriate statements as in the following program[1, 3].

Program 4.1: Importing data from an external file using a data step.

```
data NORMAL.ndata;
        infile nof1 dsd dlm="," firstobs=2;
        input Patient Treatment $ Cycle $ Pair $ Y;
        drop Pair;
run;
```

Observe that a data step starts with a ‘data’ statement and ends with the ‘run’ statement with some statements in between the two. In the ‘data’ statement the name and location of the resulting dataset is defined. Note the reference to the “NORMAL” library defined previously. The library name precedes the desired dataset name with a dot in between. The dataset “ndata” is then stored in the directory with path “..\library\normal” in the local computer’s storage device. The next statement in the data step presented in Program 4.1 is the ‘infile’ statement. This ‘infile’ statement is composed by the file reference “nof1” previously defined followed by some options that define how the data step handles the file. The ‘dsd’ option indicates that some character delimits the observations in the file. According to the ‘dlm’ option, a comma character separates the observations in the file. While the ‘firstobs=2’ option instructs the data step to ignore the first row in the file which usually is a header containing the variable names. The ‘input’ statement defines the variable names and the data type either numeric or character. Character variables have a dollar symbol ($) after the variable name, while numeric variables do not. This dataset contains five variables: A “Patient” variable identifying the patient. A “Treatment” variable identifying the treatment administered to the patient. A “Cycle” variable identifying the order of administration of the treatment to the patient. A variable named “Pair” identifying the patient cycle association. In addition, an outcome variable named “Y” holding the measurement obtained from the patient under the treatment and cycle identified on the same row. The “Pair” variable is not used by any of the programs exemplified in this document. As such, the “Pair” variable is removed from the “NORMAL.ndata” dataset through a ‘drop’ statement.

Some operations on datasets might require the data to be sorted by a specific order. While for purposes of displaying the data, it might be preferable to have it sorted by increasing or decreasing order of a specific variable. There are several ways to sort data with the SAS software one of them being through the ‘sort’ procedure.

Program 4.2: Sorting the data.

```
proc sort data=NORMAL.ndata;
        by Patient Cycle;
run;
```

In Program 4.2, the dataset “ndata” located in the “NORMAL” library is sorted by increasing order of the “Patient” and “Cycle” variables.

The “ndata” dataset is from a series of n-of-1 trials where the treatments were randomized in cycles within each patient. Each cycle contains two measurements of the outcome variable; consequently, there are two rows for each cycle in the dataset, one row for each treatment. Because there is some time between the measurements, each measurement is said to correspond to a period. For a series of n-of-1 trials with two treatments, measurements are taken under one treatment or the other, but not under both treatments simultaneously. If a measurement is taken when both treatments are present simultaneously on the patient’s body, there is uncertainty as regards to which treatment the effect is attributable. This situation is called a carry-over effect in the literature. See section 1.7 of for a longer discussion of the carry-over phenomenon. For this reason, a certain amount of time is allowed for the previous treatment to leave the body of the patient. In this way, when a measurement is recorded, only one treatment is present on a patient’s body, and the variation observed in the outcome variable is attributable to that treatment. The time required for the previous treatment to leave an individual’s body is designated as washout time. The washout time depends more on the treatment than on the individual, and can vary from a few minutes for psychological treatments, to hours, days or even weeks for some pharmacological treatments. Since n-of-1 trials can have a large number of periods, to reduce the time length of the trial, an active washout strategy can be employed. An active washout consists in commencing the administration of the next treatment when the previous treatment is still present, but measuring the outcome variable when the previous treatment is no longer present and only the desired treatment remains. Another possibility to deal with the carry-over effect is to account for it in the statistical model. These strategies for dealing with the carry-over effect are presented in more detail in section 1.8 of Senn [6]. While a very extensive discussion of mathematical approaches to the carry-over effect can be found on chapter 10 of Senn [6]. In n-of-1 trials, the outcome variable is measured several times in the same individual. Measurements taken from the same individual are certainly correlated between each other, while measurements taken from different individuals are considered uncorrelated. The “NORMAL.ndata” dataset imported to SAS by Program 4.1 does not contain information on the period. Usually the data is stored by the order of collection in time. Therefore, it is considered that for a given patient, the first row appearing in the dataset is relative to the first period, the second row corresponds to the second period, and so on. The next program inserts the period information in the “NORMAL.ndata” dataset. This information allows the future demonstration of statistical models considering time dependence. Obviously, when the period information is already available in the data, this step is not needed. This step is not required also when there is no intention to model any time dependence from the data. A clinical trial should be completely planned from the beginning, including the design of the trial, the collection of the data, and the analysis of the data. One must collect the data that will later be used in the analysis. Obviously, variables that are not measured beforehand cannot take part in the analysis.

Program 4.3: Adding period information to the data using a data step.

```
* Add the Period variable to NORMAL.ndata ;
data NORMAL.ndata;
        set NORMAL.ndata;
        by Patient;
        if first.Patient then Period=1;
        else Period+1;
        charPeriod=put(Period, 8. -L);
        drop Period;
        rename charPeriod=Period;
run;

* Reorder variables in NORMAL.ndata ;
data NORMAL.ndata;
        retain Patient Treatment Cycle Period Y;
        set NORMAL.ndata;
run;

* Make sure NORMAL.ndata is sorted
by increasing order of Patient and Period;
proc sort data=NORMAL.ndata;
        by Patient Period;
run;
```

As you can see, Program 4.3 contains two data steps and finishes with a ‘sort’ procedure. In the first data step, the ‘.first’ temporary variable is used to run through a patient’s data and effectively define the correct value for the “Period” variable. Observe the use of a ‘by’ statement. The ‘by’ statement requires the data to be sorted by increasing order of the “Patient” variable, which was previously accomplished in Program 4.2. This technique is well described in various textbooks. See for example chapter 11 of Cody [7], or section 24.2 of Cody [1], or chapter 3 of Cody [8], or even section 3.1.1 of Carpenter [9]. The purpose of the last three statements in the same data step is to perform a conversion of the data type of the “Period” variable from numeric to character. The next data step in Program 4.3 uses a ‘retain’ statement to alter the order of the variables in the dataset. The results of this data step are merely cosmetic, and a matter of personal preference. The majority of the SAS programs work correctly with the variables on any order. In some cases, it is the order of the observations that can make the difference, not the order of the variables. Finally, the ‘sort’ procedure sorts the observations in the dataset, by increasing order of the “Patient” and “Period” variables.

The imported data is from a series of n-of-1 trials where all the patients receive the same number of treatments in the same number of cycles. Such a trial and dataset are said to be balanced per opposition to an unbalanced trial and dataset where some patients might receive treatments in a different number of cycles. For studying the statistical analysis of unbalanced series of n-of-1 trials, a new dataset is created where some observations are deleted from the imported dataset.

Program 4.4: Creating unbalanced data using a data step.

```
data NORMAL.ndata_unbalanced;
        set NORMAL.ndata;
        if (Patient=11 & Cycle=3) then delete;
        if (Patient=12 & Cycle^=1) then delete;
run;
```

Program 4.4 reads the “ndata” dataset permanently stored in the “NORMAL” library. From this dataset, all the observations from cycle 3 and patient 11 are deleted; and all the observations from patient 12 that are not from cycle 1 are deleted. The resulting dataset is named “ndata_unbalanced” and is stored in the “NORMAL” library. Note that there is no loop explicitly written in Program 4.4, the reason for this being the fact that data steps implicitly loop through the observations of the dataset given in the ‘set’ statement.

One of the objectives of a clinical trial is to assess the difference between two treatments by means of the estimation of a mean difference outcome under the two treatments. The outcome variable can therefore be recoded to represent a difference of outcome under two treatments leading to a new dataset. This new dataset can then be analysed through distinct but equivalent statistical models. The next program demonstrates this operation using a data step.

Program 4.5: Creating a dataset of outcome differences using a data step.

```
data NORMAL.ddata;
        merge
                NORMAL.ndata( drop=Period where=(Treatment="A") rename=(Y=YA) )
                NORMAL.ndata( drop=Period where=(Treatment="B") rename=(Y=YB) );
        by Patient Cycle;
        drop Treatment;
        dY=YB-YA;
run;
```

In plain English Program 4.5 performs the following tasks. The ‘merge’ statement in this data step selects all observations from the “NORMAL.ndata” dataset where treatment “A” was administered; it then renames the outcome variable “Y” to “YA” and drops the variable “Period”. On the same dataset the observations where treatment “B” was administered are selected, the variable “Y” is renamed to “YB” and the variable “Period” is dropped. These observations are then merged by the variables “Patient” and “Cycle”. This means that for each patient and cycle the variables “YA” and “YB” are kept side by side. This procedure can be better understood later when this dataset is printed on screen for visualization. The “Treatment” variable is no longer needed so it is dropped through a ‘drop’ statement. A new variable “dY” is created by differencing the variables “YB” and “YA”. The resulting dataset is permanently stored under the name “ddata” on the “NORMAL” library.

An unbalanced dataset of outcome differences can be created in one of two distinct ways. Either you take the “NORMAL.ddata” dataset as input and write a program similar to Program 4.4, or you take the “NORMAL.ndata_unbalanced” dataset as input and write a program similar to Program 4.5. The next program demonstrates the latter alternative.

Program 4.6: Creating an unbalanced dataset of outcome differences using a data step.

```
data NORMAL.ddata_unbalanced;
        merge
                NORMAL.ndata_unbalanced( drop=Period where=(Treatment="A") rename=(Y=YA) )
                NORMAL.ndata_unbalanced( drop=Period where=(Treatment="B") rename=(Y=YB) );
        by Patient Cycle;
        drop Treatment;
        dY=YB-YA;
run;
```

Program 4.6 is very similar to Program 4.5 with the exception of the dataset names. Detailing the code would involve repeating much of what is written above. Please refer to the paragraph around Program 4.5.if you need to understand Program 4.6.

5 Importing and handling data using SQL

It might be possible to import comma-separated files to the SAS software using ‘proc sql’, however information is either scarce or difficult to find. Data steps provide an easy way of importing data to the SAS software as in Program 4.1. An alternative way is to use ‘proc import’ as in the following piece of code.

Program 5.1: Importing data from an external file using ‘proc import’.

```
proc import
        datafile=nof1
        out=WORK.ndata(
            rename=(VAR1=Patient VAR2=Treatment VAR3=Cycle VAR4=Pair VAR5=Y)
        )
        dbms=dlm
        replace;
    datarow=2;
    delimiter=",";
    getnames=no;
run;
```

The ‘datafile’ option accepts either a path to a file enclosed in quotation marks or a file reference. Here the file reference defined in Program 2.2 is supplied. One disadvantage of using ‘proc import’ instead of a data step is that data types for the variables cannot be specified, so all variables holding numbers are imported as numeric by default, and all variables holding characters are imported as character by default. In this case the variables “Cycle” and “Pair” are desired as character, so the data is imported to the ‘WORK’ library and the conversion is done posteriorly within ‘proc sql’. Unlike with a data step’s input statement the variable names cannot be specified and the variables are imported as VAR1, VAR2… by default, so the variables are renamed in the output dataset through the rename option. An alternative is to rename the variables within ‘proc sql’. The ‘dbms’ option specifies a delimited file, i.e. the observations are delimited by some character within the file specified. Since the ‘replace’ option is specified the “WORK.ndata” dataset is replaced in case it already exists. The data is read from the file starting from row number 2 as specified in the ‘datarow’ statement. The ‘delimiter’ statement specifies the delimiter character in the file supplied to the ‘datafile’ option. In this case, the file is comma-separated. The names of the variables are not obtained from the file hence the ‘getnames=no’ statement.

Unlike with data steps, as many datasets as desired can be created within a single ‘proc sql’ call. In fact, all the SQL code within a given script can be placed inside a single ‘proc sql’ call. However, the code is divided through several ‘proc sql’ calls, rendering explaining and reading the code much easier.

The “ndata” dataset is from a series of n-of-1 trials where the treatments were randomized in cycles within each patient. Each cycle contains two measurements of the outcome variable; consequently, there are two rows for each cycle in the dataset, one row for each treatment. Because there is some time between the measurements, each measurement is said to correspond to a period. For a series of n-of-1 trials with two treatments, measurements are taken under one treatment or the other, but not under both treatments simultaneously. If a measurement is taken when both treatments are present simultaneously on the patient's body, there is uncertainty as regards to which treatment the effect is attributable. This situation is called a carry-over effect in the literature. See section 1.7 of Senn [6] for a longer discussion of the carry-over phenomenon. For this reason, a certain amount of time is allowed for the previous treatment to leave the body of the patient. In this way, when a measurement is recorded, only one treatment is present on a patient's body, and the variation observed in the outcome variable is attributable to that treatment. The time required for the previous treatment to leave an individual's body is designated as washout time. The washout time depends more on the treatment than on the individual, and can vary from a few minutes for psychological treatments, to hours, days or even weeks for some pharmacological treatments. Since n-of-1 trials can have a large number of periods, to reduce the time length of the trial, an active washout strategy can be employed. An active washout consists in commencing the administration of the next treatment when the previous treatment is still present, but measuring the outcome variable when the previous treatment is no longer present and only the desired treatment remains. Another possibility to deal with the carry-over effect is to account for it in the statistical model. These strategies for dealing with the carry-over effect are presented in more detail in section 1.8 of Senn [6]. While a very extensive discussion of mathematical approaches to the carry-over effect can be found on chapter 10 of Senn [6]. In n-of-1 trials, the outcome variable is measured several times in the same individual. Measurements taken from the same individual are certainly correlated between each other, while measurements taken from different individuals are considered uncorrelated. The "WORK.ndata" dataset imported to SAS by Program 5.1 does not contain information on the period. Usually the data is stored by the order of collection in time. Therefore, it is considered that for a given patient, the first row appearing in the dataset is relative to the first period, the second row corresponds to the second period, and so on. The next program creates a new dataset containing the period information from the "WORK.ndata" dataset. This information allows the future demonstration of statistical models considering time dependence. Obviously, when the period information is already available in the data, this step is not needed. This step is not required also when there is no intention to model any time dependence from the data. A clinical trial should be completely planned from the beginning, including the design of the trial, the collection of the data, and the analysis of the data. One must collect the data that will later be used in the analysis. Obviously, variables that are not measured beforehand cannot take part in the analysis.

Program 5.2: Adding period information to the data using SQL.

```
* Determine and count unique Patients ;
proc sql noprint;
        select  distinct Patient
                into :Patient1 - :Patient999999999
        from WORK.ndata;
quit;
%let PatientNumber = &SqlObs;

* Declare %sqlBody212 macro ;
%macro sqlBody212;
        %local i;
        %do i=2 %to &PatientNumber %by 1;
                union
                select  Patient,
                        Treatment,
                        Cycle,
                        monotonic() as Period,
                        Y
                from WORK.ndata
                where Patient eq &&Patient&i
        %end;
%mend sqlBody212;

* Determine Period variable from WORK.ndata,
save the resulting data to WORK.periodata,
and finally delete WORK.ndata;
proc sql;
        create table WORK.periodata as
        select  Patient,
                Treatment,
                Cycle,
                monotonic() as Period,
                Y
        from WORK.ndata
        where Patient eq &Patient1
        %sqlBody212;
        drop table WORK.ndata;
quit;
```

The use of the SAS macro language in Program 5.2 allows for a shorter program, which can indeed be considered simpler than the much longer alternative. Learning the SAS macro language requires some time and patience. The reader interested in learning and mastering this useful and powerful programming language can use the works of Carpenter [5] and or Burlew [4] as reference. There are other perhaps good works on the subject, but these two textbooks were sometimes consulted during the development of the programs presented in this text. Therefore, these references are the obvious recommendations. The intension here is not to teach anyone the SAS macro language. Those more versed in the SAS programming language surely can understand the code very quickly. Program 5.2 is briefly explained next. Program 5.2 consists of one ‘proc sql’ call, a macro definition and a second ‘proc sql’ call, which makes use of the macro defined above in the same program. In the initial ‘proc sql’ call the values of the “Patient” variable are read from the “WORK.ndata” dataset, into several global macro variables, one for each distinct value. Twelve global macro variables are defined, “Patient1”, “Patient2”, and so on, until “Patient12”. By design, the ‘sql’ procedure stores the number of observations processed in a predefined global macro variable named “SqlObs”. Since the number of patients from the sample equals the number of distinct values of the “Patient” variable read by the ‘sql’ procedure, the “PatientNumber” global macro variable is assigned the value of “SqlObs”, after the ‘sql’ procedure is terminated by the ‘quit’ statement. Afterwards the “sqlBody212” macro is defined. The purpose of this macro is detailed below. In the second ‘sql’ procedure of Program 5.2 a new dataset named ‘’WORK.periodata” is created by selecting the desired variables and observations from the “WORK.ndata” dataset. A ‘where’ clause is used to subset the data to the observations from the first patient encountered in the dataset. Note that a new variable “Period” is defined to store the result of the ‘monotonic’ function. The ‘monotonic’ function counts the number of observations processed, and it results essentially in an index. The result of the ‘monotonic’ function in SQL is similar to the result of the ‘_n_’ automatic variable in a data step. In this case, for a given patient an observation corresponds to a period, so it is safe to set the value of the period as the index under that patient. After the data for the first patient is processed, the “sqlBody212” macro is called. When Program 5.2 is executed, the “sqlBody212” macro is expanded and code similar to the one described for the first patient is inserted for all the remaining patients in the sample, with a ‘union’ keyword between each patient’s pieces of code. Essentially the last ‘sql’ procedure in Program 5.2 determines the value of the “Period” variable for each patient separately and puts the resulting rows of all the patients together in a single dataset. The ‘union’ keyword instructs the software to put the data resulting from each patient, together in a single dataset. SAS does not support using ‘%do’ loops in open code. A ‘%do’ loop cannot be placed directly inside the ‘sql’ procedure. To work around this limitation, a macro containing the desired code is defined outside the procedure. Then the macro is invoked inside the procedure. The “WORK.ndata” dataset is not needed anymore, so a ‘drop table’ statement deletes it. A ‘quit’ statement is always used to stop the execution of the ‘sql’ procedure. Note that there is a “Pair” variable in the “WORK.ndata” dataset created by Program 5.1. The “Pair” variable is not present on any of the ‘select’ statements of Program 5.2, which leads to the “WORK.periodata” dataset. Therefore, the “WORK.periodata” dataset does not contain the “Pair” variable. Program 5.2 leaves some global macro variables and a macro, which are not necessary anymore. The following simple program is designed to remove these global macro variables and macro from the SAS system.

Program 5.3: Removing global macro variables from the SAS system.

```
* Declare %DeleteGlobal macro ;
%macro DeleteGlobal;
        %local i;
        %do i=1 %to &PatientNumber %by 1;
                %symdel Patient&i;
        %end;
        %symdel PatientNumber;
%mend DeleteGlobal;

* Delete global macro variables ;
%DeleteGlobal

* Delete macros from WORK.Sasmacr ;
proc catalog cat=WORK.Sasmacr;
        delete sqlBody212 DeleteGlobal / et=macro;
quit;
```

Since ‘%do’ loops cannot be run in open code, the desired ‘%do’ loop is placed inside the “DeleteGlobal” macro definition. The “DeleteGlobal” macro is invoked afterwards. When the “DeleteGlobal” macro is expanded, the macro pre-processor loops through the number of patients in the dataset and deletes the global macro variables that were declared in Program 5.2, one for each patient. The “PatientNumber” is required for the ‘%do’ loop, so it can only be deleted after the ‘%do’ loop is expanded. Finally, the ‘catalog’ procedure is executed to remove both, the “sqlBody212” macro previously defined in Program 5.2, and the “DeleteGlobal” macro variable defined in Program 5.3 itself.

In the next program, the variables “Cycle” and “Period” are converted from numeric to character and the data is sorted by increasing order of both the patient and period indicator variables.

Program 5.4: Converting data types and sorting the data using SQL.

```
proc sql;
        create table NORMAL.ndata as
        select  Patient,
                Treatment,
                left( put(Cycle, 8.) ) as Cycle,
                left( put(Period, 8.) ) as Period,
                Y
        from WORK.periodata
        order by Patient, Period;
        drop table WORK.periodata;
quit;
```

Essentially, what Program 5.4 does is creating a dataset named “ndata” in the “NORMAL” library from the “periodata” dataset that was stored before in the ‘WORK’ library by Program 5.1. Program 5.4 selects the desired variables from “WORK.periodata” while converting variables “Cycle” and “Period” from numeric to character and ordering the observations by increasing order of the variables “Patient” and “Period”. To convert the variables from numeric to character the ‘put’ function is called. While the ‘left’ function is used here to left-align the resulting character string. After the “NORMAL.ndata” dataset is successfully created, the “WORK.periodata” dataset is no longer needed so it is removed through a ‘drop table’ statement. Finally, the ‘quit’ statement terminates the SQL procedure.

The imported data is from a series of n-of-1 trials where all the patients receive the same number of treatments in the same number of cycles. Such a trial and dataset are said to be balanced as opposed to an unbalanced trial and dataset where some patients might receive treatments in a different number of cycles. For studying the statistical analysis of unbalanced series of n-of-1 trials, a new dataset is created where some observations are discarded from the imported dataset.

Program 5.5: Creating unbalanced data using SQL.

```
proc sql;
        create table NORMAL.ndata_unbalanced as
        select *
        from NORMAL.ndata
        where not (Patient eq 11 and Cycle eq "3")
        and not (Patient eq 12 and Cycle ne "1");
quit;
```

There are a number of varied ways to accomplish this using ‘proc sql’. The approach presented in Program 5.5 is not the usual approach where the undesired observations are deleted. Instead, the desired observations are selected from all the variables in the “NORMAL.ndata” dataset to create the new dataset “NORMAL.ndata_unbalanced”. The result is the same as the one obtained from running the data step in Program 4.4.

One of the objectives of a clinical trial is to assess the difference between two treatments through the estimation of a mean difference outcome under the two treatments. The outcome variable can therefore be recoded to represent a difference of outcome under two treatments leading to a new dataset. This new dataset can then be analysed through distinct but equivalent statistical models. The next program demonstrates this recoding operation using ‘proc sql’.

Program 5.6: Creating a dataset of outcome differences using SQL.

```
proc sql;
        create view A as
        select  Patient,
                Cycle,
                Y as YA
        from NORMAL.ndata
        where Treatment eq "A";
        create view B as
        select  Patient,
                Cycle,
                Y as YB
        from NORMAL.ndata
        where Treatment eq "B";
        create table NORMAL.ddata as
        select  A.Patient,
                A.Cycle,
                YA,
                YB,
                YB-YA as dY
        from A join B
        on A.Patient eq B.Patient
        and A.Cycle eq B.Cycle
        order by Patient, Cycle;
        drop view A, B;
quit;
```

Note that Program 5.6 is bigger than Program 4.5 while leading to the same result. Program 5.6 is the first program presented in this document, which uses SQL views. A SQL view contains SQL instructions and it does not perform any modification on the data until some other operation which itself modifies the data runs the SQL view [2, 3]. The same results can be accomplished without resorting to SQL views, however using SQL views might be easier for the programmer writing the code as well as for those interested in reading the same code. Program 5.6 starts by creating two SQL views, one named “A” and the other named “B”. View “A” is created by selecting the observations of the desired variables from the “NORMAL.ndata” dataset where the “Treatment” variable equals “A”. While view “B” is created by selecting the observations of the desired variables from the same dataset where the variable “Treatment” equals “B”. Note how the variable “Y” is renamed becoming “YA” in view “A” and “YB” in view “B”. When no library is specified, the views are stored in the ‘WORK’ library by default. The next step is to create the “NORMAL.ddata” dataset by joining the data from the two views for equal values of the “Patient” and “Cycle” variables present in both views, while sorting the observations by increasing order of “Patient” and “Cycle”. In this step, it is important to note how the variables from view “A” are selected by prefixing them with the view name with a dot in between as in “A.Patient”. The variable is present in both views so if only “Patient” was written the resulting dataset would have two columns referring to that variable, leading to duplicated information. The same would happen for the variable “Cycle”. Also, note how the new variable “dY” is created by differencing the variables “YB” and “YA”. The variables “YB” and “YA” originate from distinct views so there is no need to prefix them with their respective view name. After the “NORMAL.ddata” dataset is created, the views “A” and “B” are removed from the “WORK” library by means of a ‘drop view’ statement.

An unbalanced dataset of outcome differences can be created in a varied number of distinct ways. One way is taking the “NORMAL.ddata” dataset as input and write a program similar to Program 5.5, an alternative is taking the “NORMAL.ndata_unbalanced” dataset as input, and write a program similar to Program 5.6. Due to the relative large size of Program 5.6, an option similar to Program 5.5 is considered.

Program 5.7: Creating an unbalanced dataset of outcome differences using SQL.

```
proc sql;
        create table NORMAL.ddata_unbalanced as
        select *
        from NORMAL.ddata;
        delete from NORMAL.ddata_unbalanced
        where (Patient eq 11 and Cycle eq "3")
        or (Patient eq 12 and Cycle ne "1");
quit;
```

Program 5.7 could have been written very similarly to Program 5.5. Instead, a distinct approach was chosen to provide the reader with yet another example. In this program, first the “NORMAL.ddata” dataset is backed up under the desired target dataset name, in this case “NORMAL.ddata_unbalanced”. Afterwards the undesired observations are effectively deleted from the dataset created in the first statement. Program 5.7 is equivalent to Program 4.6 resulting in the very same dataset.

6 Printing the data

In the earliest stages of the statistical analysis, information on a dataset must be collected. A proper statistical analysis may not be possible without gathering information on the variables present in the dataset. At this stage, the analyst might pose some important questions. What variables are present in the dataset, and what do they represent? How the observations were measured and in what units? Which variables are continuous and which are categorical? Which levels were measured for a specific categorical variable? How many observations are available for a specific level of a categorical variable? The number of possible questions is so immense that it is difficult to enumerate all of them here! In fact, such questions depend on the particular situation. The SQL language can be used within the SAS software to perform queries on a given dataset and answer some of those questions [2]. Other than that, the SAS software provides several distinct procedures that can be applied in a varied number of situations. A small number of examples are presented in this section. These examples are only a fraction of what is possible. A great number of textbooks have been published that might help in other situations. Some but not all of those works are referenced in this document. Besides documentation for the SAS software and programming language is very well done and available on the World Wide Web for anyone to consult [10].

A dataset can be printed for visualization within the SAS software using either the ‘print’ or the ‘sql’ procedures. The ‘sql’ procedure is more powerful than the ‘print’ procedure, so the ‘sql’ procedure is used in the examples presented in this section. The next program prints the dataset created by either Program 4.1 or Program 5.1. To save some space in this document only some observations are printed.

Program 6.1: Printing the first observations of a dataset using SQL.

```
title "First six observations of NORMAL.ndata";
proc sql outobs=6;
        select *
        from NORMAL.ndata;
quit;
```

The first statement of Program 6.1 defines a title for the code and its output. Usually a SAS program is made up of several procedure calls which when submitted sequentially to the SAS software for interpretation lead to the results being laid out together in the SAS software’s output. A title preceding a procedure in the code identifies that procedure’s results in the ensemble of results. In the SQL programming language, the asterisk symbol is used to represent all the variables present in a given dataset, so all the variables from “NORMAL.ndata” are selected for printing. The ‘outobs’ option in ‘proc sql’ defines the number of observations printed. To save space on this document only six observations are requested. The submission of Program 6.1 outputs a table identical to Table 6.1 presented below. The character string used in the 'title' statement in Program 6.1 is used as caption in Table 6.1.

Table 6.1: First six observations of NORMAL.ndata.

Patient	**Treatment**	**Cycle**	**Period**	**Y**
1	A	1	1	2394
1	B	1	2	2686
1	A	2	3	2515
1	B	2	4	2675
1	A	3	5	2583
1	B	3	6	2802

A subset of a given dataset can be easily printed using SQL. The desired variables can be selected and printed by writing the variable names separated by a comma in the 'select statement'. This technique is used in Program 5.2, Program 5.4 and Program 5.6 though not for printing. To select observations of a given dataset that satisfy a given condition the 'where' clause can be used in a 'select' statement. The next program uses the 'where' clause to select observations from patients 11 and 12 in "NORMAL.ndata" for printing.

Program 6.2: Selecting observations from a dataset using SQL.

```
title "Observations from patients 11 and 12 in NORMAL.ndata";
proc sql;
	select *
	from NORMAL.ndata
	where Patient ge 11;
quit;
```

In Program 6.2, all variables from "NORMAL.ndata" are selected where the condition variable "Patient" greater or equal to 11 is met. Here the 'ge' keyword means "greater or equal". Using the operator '>=' instead of the keyword 'ge' works as well. Since there are 12 patients in the dataset, the condition implies that data for patients 11 and 12 is selected for printing. Alternative conditions can be used with the exact same result, such as "Patient eq 11 or Patient eq 12" which also can be written as "Patient = 11 | Patient = 12". Running Program 6.2 within the SAS software writes the data presented in Table 6.2 to the software's output.

Table 6.2: Observations from patients 11 and 12 in NORMAL.ndata.

Patient	**Treatment**	**Cycle**	**Period**	**Y**
11	A	1	1	2617
11	B	1	2	2923
11	A	2	3	2629
11	B	2	4	2832
11	A	3	5	2732
11	B	3	6	2866
12	A	1	1	2627
12	B	1	2	2759
12	A	2	3	2712
12	B	2	4	2698
12	A	3	5	2572
12	B	3	6	2826

Useful information on a SAS dataset can be obtained through the ‘contents’ procedure. Usage of the ‘contents’ procedure is relatively simple, and is demonstrated in the program below.

Program 6.3: Describing a SAS dataset.

```
title "Contents of NORMAL.ndata";
proc contents data=NORMAL.ndata varnum;
run;
```

The dataset reference is supplied to the ‘data’ option of the ‘contents’ procedure. The ‘varnum’ option requests the variables listed by their order in the dataset. The output resulting from Program 6.3 is quite long with four tables being produced one after the other.

Table 6.3: Contents of NORMAL.ndata.

Data Set Name	NORMAL.NDATA	**Observations**	72
Member Type	DATA	**Variables**	5
Engine	V9	**Indexes**	0
Created	12/17/2017 02:45:49	**Observation Length**	40
Last Modified	12/17/2017 02:45:49	**Deleted Observations**	0
Protection		**Compressed**	NO
Data Set Type		**Sorted**	YES
Label			
Data Representation	WINDOWS_64		
Encoding	wlatin1 Western (Windows)		

Table 6.3 is the first one being outputted after the submission of Program 6.3. The information presented in Table 6.3 is self-explanatory so no further comments are made here.

Table 6.4: Contents of NORMAL.ndata - Engine/Host Dependent Information.

Engine/Host Dependent Information	
Data Set Page Size	65536
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	1632
Obs in First Data Page	72
Number of Data Set Repairs	0
ExtendObsCounter	YES
Filename	E:\IDEAL_SAS\library\normal\ndata.sas7bdat
Release Created	9.0401M3
Host Created	X64_8PRO

Table 6.4 follows with information relative to the machine running the SAS software and the SAS software itself. Here one can find information as regards the location of the file holding the dataset in the computer's file system; the operating system over which the SAS software runs; the version of the SAS software's engine; and other information that might prove itself useful.

Table 6.5: Contents of NORMAL.ndata – Variables in Creation Order.

Variables in Creation Order			
#	**Variable**	**Type**	**Len**
1	Patient	Num	8
2	Treatment	Char	8
3	Cycle	Char	8
4	Period	Char	8
5	Y	Num	8

Table 6.5 is the third one outputted by the 'contents' procedure written in Program 6.3. This table displays information on the variables present in the dataset. The information presented in Table 6.5 contains the variable name, its type either numeric or character, and the length of each element of that variable in bytes. Since the 'varnum' option was specified in the 'contents' procedure used in Program 6.3 the variables are laid out in creation order in Table 6.5.

Table 6.6: Contents of NORMAL.ndata – Sort Information.

Sort Information	
Sortedby	Patient Period
Validated	YES
Character Set	ANSI

The fourth and last table created by Program 6.3 is Table 6.6. This table provides information introduced in the dataset by any previous sort procedure. Remember that Program 4.3, Program 4.2 or alternatively Program 5.4 sort "NORMAL.ndata" by increasing order of the variables "Patient" and "Period" also attaching information relative to the sorting procedure to the dataset.

7 Naive estimation of individual treatment effects

Series of n-of-1 trials can be used to estimate both an overall treatment effect as well as individual treatment effects.[11, 12]. This section is devoted to the naive estimation of individual treatment effects. An individual treatment effect can be defined as an average difference of outcome variable registered under two treatments for a specific individual. For the individual treatment effects to be estimable, the subject under study must be administered the two treatments being compared during the course of the trial as is done in n-of-1 trials. Naive methods of estimation of individual treatment effects do not account for any dependency between the observations and use only the data of the subject in question while discarding data from other subjects studied in the series of n-of-1 trials. Due to their use of lesser information, these methods are considered inefficient when compared to other methods, which use more information. Despite the availability of preferable more efficient methods, naive methods have been frequently used to estimate individual treatment effects by the scientific community reporting series of n-of-1 trials in the literature [13-18]. The SAS software is used for naive estimation of individual treatment effects. This way several statistical methods discussed throughout this document can be compared. It is however very important to emphasize that the usage of naive methods of estimation of individual treatment effects should be avoided.

The following program computes and displays a table containing the treatment effects for each patient present in the dataset of outcome variable differences obtained from either Program 4.5 or Program 5.6. Note that in this dataset the n-of-1 trials that make up the series are balanced. This means that the number of observations is the same for every patient available in the dataset.

Program 7.1: Creating and printing a dataset holding the naive individual treatment effects.

```
proc means data=NORMAL.ddata alpha=0.05 noprint;
        by Patient;
        var dY;
        output
                out=NORMAL.sddata( drop=_TYPE_ rename=(_FREQ_=nobs) )
                mean=meanY
                stderr=stdY
                t=tY
                probt=probtY
                lclm=lowY
                uclm=uppY;
run;

proc print data=NORMAL.sddata noobs label;
        label
                Patient='Patient'
                nobs='N Obs'
                meanY='Mean'
                stdY='Std Error'
                tY="t Value"
                probtY="Pr > |t|"
                lowY='Lower 95% CL for Mean'
                uppY='Upper 95% CL for Mean';
        title 'Naive individual treatment effects for balanced data';
run;
```

Two procedures ‘means’ and ‘print’ are used in Program 7.1. Here ‘proc means’ is used to compute several statistics from variable “dY” for each level of the categorical variable “Patient”. The desired dataset is supplied to the ‘data’ option in the ‘proc means’ statement. In this case, the “NORMAL.ddata” dataset of outcome differences obtained from either Program 4.5 or Program 5.6 is used. The level of confidence for the confidence interval of the mean is specified in the ‘alpha’ option. In this case, the purpose of the ‘means’ procedure is to create a dataset holding the results, so the ‘noprint’ option is defined in the ‘proc means’ statement to effectively suppress printing of the results. The ‘by’ statement requests the estimation to be made for each level of the category variable “Patient”. The ‘var’ statement indicates the desired variable for estimation in this case “dY”. So only the values of the variable “dY” available under the given level of the categorical variable “Patient” are used for estimation. The estimates are computed for each level of the categorical variable “Patient”. The last statement under ‘proc means’ is an ‘output’ statement. In the ‘output’ statement, the output dataset is defined in the ‘out’ option, followed by several other options defining the desired statistics. The defined statistics are in this order, the mean, the standard error of the mean, the t-statistic, the p-value, the lower limit of the confidence interval of the mean, and the upper limit of the confidence interval of the mean. In each case, the option defining the desired statistic is set equal to the desired variable name in the output dataset for that statistic's results. The 'means' procedure from Program 7.1 creates and stores the dataset named "sddata" in the "NORMAL" library previously defined in Program 2.1. The next step in Program 7.1 is to print the "NORMAL.sddata" dataset. Here the 'print' procedure is used. Like with many other procedures the dataset is supplied to the 'data' option. The 'noobs' option suppresses the column in the output that identifies each observation number. In addition, the 'label' option requests variable labels as column headings in the output table. The 'label' statement defines the labels of each of the variables present in the dataset. Note that the variable names are set equal to the desired labels under a unique 'label' statement. Finally, the 'title' statement defines a title to identify the table in the output. Submitting Program 7.1 within the SAS software outputs Table 7.1 below.

Table 7.1: Naive individual treatment effects - Balanced data.

Patient	N Obs	Mean	Std Error	t Value	Pr > |t|	Lower 95% CL for Mean	Upper 95% CL for Mean
1	3	223.667	38.176	5.85875	0.02792	59.406	387.927
2	3	84.667	95.325	0.88819	0.46815	-325.482	494.815
3	3	60.000	102.587	0.58487	0.61783	-381.394	501.394
4	3	348.000	68.879	5.05233	0.03701	51.637	644.363
5	3	259.333	168.919	1.53525	0.26450	-467.468	986.135
6	3	50.000	104.510	0.47842	0.67954	-399.670	499.670
7	3	175.000	52.776	3.31589	0.08017	-52.078	402.078
8	3	153.667	91.590	1.67776	0.23539	-240.414	547.748
9	3	324.333	50.854	6.37776	0.02371	105.527	543.140
10	3	247.667	84.924	2.91633	0.10021	-117.732	613.066
11	3	214.333	49.974	4.28886	0.05030	-0.689	429.356
12	3	124.000	77.468	1.60066	0.25060	-209.319	457.319

As mentioned before the data arising from the series of n-of-1 trials is balanced. For each patient in this series of n-of-1 trials three observations were registered. The standard error is estimated for each patient independently using the data available for that particular patient, hence its distinctiveness. The confidence intervals suggest that the treatments are not significantly different for all the patients except for patient number 1, 4 and 9.

The advantage of having saved the results to a dataset in Program 7.1 is that the saved dataset can be used in subsequent programs. Table 7.1 is relatively small so the minimum and maximum individual treatment effects can simply be determined by observation. For larger tables the problem is not that simple. The next program computes the extreme values of the individual treatment effects.

Program 7.2: Computing the extreme values of naive individual treatment effects.

```
title "Extreme values of naive individual treatment effects";
proc sql;
        select  Patient,
                meanY as Extreme
        from NORMAL.sddata
        having meanY eq min(meanY)
        or meanY eq max(meanY);
quit;
```

This simple SQL program determines the minimum and maximum naive individual treatment effects by selecting the “Patient” and “meanY” variables from “NORMAL.sddata” where the values of “meanY” are minimum and maximum. Table 7.2 below is obtained from running Program 7.2.

Table 7.2: Extreme values of naive individual treatment effects.

Patient	Extreme
4	348
6	50

According to Table 7.2, the highest naive individual treatment effect of 348 units was computed for patient number 4, while patient number 6 registered the lowest individual treatment effect of 50 units.

The ‘summary’ procedure is similar to the ‘means’ procedure accepting the same options and statements. The following program uses the ‘summary’ procedure to compute several location measures on the dataset of naive individual treatment effects created by Program 7.1.

Program 7.3: Computing summary statistics of naive individual treatment effects.

```
title "Summary statistics of naive individual treatment effects";
proc summary
                data=NORMAL.sddata
                print
                maxdec=2
                min q1 mean median q3 max;
        var meanY;
run;
```

The “NORMAL.sddata” dataset is supplied to the ‘data’ option of the ‘proc summary’ statement. The ‘summary’ procedure suppresses printing of the results by default, so the ‘print’ option is defined in the statement. The ‘maxdec’ option defines the number of significant figures in the results; in this case, the results are rounded to two decimal places. Afterwards several options defining the desired statistics are written. The requested statistics are in this order, the minimum, the first quartile, the mean, the median, the third quartile and the maximum. The ‘var’ statement defines the variable for computation of the statistics, in this case variable “meanY” holding the naive individual treatment effects. The summary procedure used in Program 7.3 writes Table 7.3 to the SAS software’s output.

Table 7.3: Summary statistics of naive individual treatment effects.

Analysis Variable : meanY					
Minimum	**Lower Quartile**	**Mean**	**Median**	**Upper Quartile**	**Maximum**
50.00	104.33	188.72	194.67	253.50	348.00

Table 7.3 displays several measures of location of the naive individual treatment effects. The mean of the naive individual treatment effects in Table 7.3 is actually the naive overall treatment effect.

Next, the naive individual treatment effects for the unbalanced dataset created by either Program 4.6 or Program 5.7 is computed and presented. This way the results for unbalanced data can be compared with the results for balanced data presented in Table 7.1.

Program 7.4: Computing naive individual treatment effects.

```
proc means
                data=NORMAL.ddata_unbalanced
                alpha=0.05
                maxdec=4
                mean stderr t probt clm;
        class Patient;
        var dY;
        title 'Naive individual treatment effects for unbalanced data';
        ods noproctitle; * remove procedure title from output ;
run;
```

A ‘means’ procedure similar to the one used in Program 7.1 is used. The big difference noticed between Program 7.1 and Program 7.4 is the absence of the ‘output’ statement in the later program. Program 7.4 outputs the results directly without resorting to the creation of a dataset. In Program 7.4 the dataset “NORMAL.ddata_unbalanced”, is used as the ‘data’ option. As in Program 7.1 a level of confidence of 0.05 is defined in the ’alpha’ option. The ‘maxdec’ option defines four decimal places for the results. The following options of the ‘proc means’ statement request the desired statistics. The requested statistics are in this order: the mean, the standard error of the mean, the t-statistic, the p-value, and the confidence limits for the mean. In this case a ‘class’ statement is used in place of a ‘by’ statement which was used before in Program 7.1. The ‘class’ statement leads to a more compact table containing the results of all the patients, while the ‘by’ statement leads to twelve tables one for each patient. The results with a ‘class’ statement are preferred to save some space in this document. Again, the analysis variable is “dY” as written in the ‘var’ statement. A title for the output is defined. Finally, the ‘ods noproctitle’ statement suppresses printing of the procedure title in the output. All text between an asterisk and a semicolon is a comment, and is not interpreted by the SAS software. Program 7.4 produces Table 7.4 below.

Table 7.4: Naive individual treatment effects - Unbalanced data.

Analysis Variable : dY							
Patient	**N Obs**	**Mean**	**Std Error**	**t Value**	**Pr > |t|**	**Lower 95% CL for Mean**	**Upper 95% CL for Mean**
1	**3**	223.6667	38.1765	5.86	0.0279	59.4065	387.9268
2	**3**	84.6667	95.3246	0.89	0.4681	-325.4819	494.8153
3	**3**	60.0000	102.5865	0.58	0.6178	-381.3943	501.3943
4	**3**	348.0000	68.8791	5.05	0.0370	51.6371	644.3629
5	**3**	259.3333	168.9194	1.54	0.2645	-467.4684	986.1350
6	**3**	50.0000	104.5100	0.48	0.6795	-399.6701	499.6701
7	**3**	175.0000	52.7763	3.32	0.0802	-52.0779	402.0779
8	**3**	153.6667	91.5903	1.68	0.2354	-240.4145	547.7478
9	**3**	324.3333	50.8538	6.38	0.0237	105.5270	543.1397
10	**3**	247.6667	84.9241	2.92	0.1002	-117.7325	613.0658
11	**2**	254.5000	51.5000	4.94	0.1271	-399.8695	908.8695
12	**1**	132.0000	.	.	.	.	.

Note that a single observation from patient 11 was deleted, and two observations from patient 12 were deleted when creating the unbalanced dataset used to compute Table 7.4. Since the naive method used to compute the statistics presented in Table 7.1 and Table 7.4 considers only the data from the patient in question, the results for the patients for which the data is not changed do not change either. Only the results for the patients for which observations were deleted are different in Table 7.4 when compared to Table 7.1. Patient number 12 contains only one observation so the mean for that patient is equal to the value of that unique observation, and the other statistics in the table are not estimable for that patient.

8 The paired t-test

The paired t-test is a parametric test that can be used to test the significance of treatment difference both at the overall level and at the individual level in series of n-of-1 trials. The paired t-test assumes the data as independent and normally distributed. The independence assumption is questionable for any measurement taken from the same individual as is done for n-of-1 trials. However, a justification can be provided in terms of testing the point null hypothesis that the treatments are identical for any patient. In that case, under the null hypothesis and given randomization (which will vary the order of the treatments) the observations can be treated as if they were independent. A more elaborate discussion of this is provided by Senn and Hildebrand [19] under section 7 of their paper. The next programs demonstrate the application of the t-test to the "NORMAL.ddata" dataset created from either Program 4.5 or Program 5.6.

Program 8.1: Performing a two-sided paired t-test with 5% significance level.

```
title "Two-sided paired t-test 5% significance level";
proc ttest
                data=NORMAL.ddata
                alpha=0.05
                dist=normal
                h0=0
                sides=2
                test=diff
                ci=none
                plots=none;
        paired YB*YA;
run;
```

A title is defined on top of Program 8.1 to permit the identification of the results in the SAS software's output. Within the SAS software, 'proc ttest' can be used to perform a t-test. As is the case with other procedures, the dataset is supplied to the 'data' option. The significance level for the test is defined in the 'alpha' option. The option 'dist' specifies the distributional assumption, which is "normal" in this case, so the observations are assumed normally distributed. The null value is defined in the 'h0' option. This test considers a null hypothesis of no difference between the means. A two-sided t-test is desired so the 'sides' option is set equal to 2. Specifying the 'test' option equal to "diff" tests the difference of means. A confidence interval for the standard error is not desired so the 'ci' option is set to "none". The last option in the 'proc ttest' statement is 'plots', and setting it to "none" supresses the display of plots in the output. The 'paired' statement is where the variables for the test are defined. Writing "YB*YA" requests a test of the mean of variable "YB" minus the mean of variable "YA". Both variables must be present in the dataset defined in the 'data' option, as is the case here.

Since the dataset “NORMAL.ddata” contains the variable “dY” which is equal to variable “YB” minus variable “YA”, an alternative program with a ‘ttest’ procedure can be written yielding the exact same results. The following program demonstrates an alternative usage of the ‘ttest’ procedure.

Program 8.2: Performing an alternative two-sided paired t-test with 5% significance level.

```
title "Alternative two-sided paired t-test 5% significance level";
proc ttest
                data=NORMAL.ddata
                alpha=0.05
                dist=normal
                h0=0
                sides=2
                ci=none
                plots=none;
        var dY;
run;
```

Program 8.2 is very similar to Program 8.1, the most noticeable difference being the presence of a ‘var’ statement in Program 8.2 instead of a ‘paired’ statement as in Program 8.1. Also note that there is a ‘test=diff’ option in Program 8.1 while in Program 8.2 there is not. The exact same results are obtained from running both Program 8.1 and Program 8.2. In this case, three tables are written to the SAS software’s output.

Table 8.1: Two-sided paired t-test 5% significance level - Mean.

N	Mean	Std Dev	Std Err	Minimum	Maximum
36	188.7	159.2	26.5394	-147.0	592.0

Table 8.2: Two-sided paired t-test 5% significance level - CL Mean.

Mean	95% CL Mean	
188.7	134.8	242.6

Table 8.3: Two-sided paired t-test 5% significance level - t statistic.

DF	t Value	Pr > |t|
35	7.11	<.0001

Table 8.1, the first one resulting from ‘proc ttest’ presents the number of observations, the estimate of the mean, the standard deviation, the standard error of the mean, the minimum and the maximum, from left to right, in this order. Table 8.2 follows with the estimate of the mean and the lower and upper confidence limits for the mean. The last table is Table 8.3, and it contains the degrees of freedom, the t statistic and the

p-value. Note that the confidence interval for the mean does not contain the value 0, which is the null value. In addition, the p-value is much lower than 0.05 the value for the significance level. So the null hypothesis of equality of the means registered under the two treatments shall be rejected. In this case, both the confidence interval for the mean and the p-value suggest that the treatments are significantly different from each other on average at the 5% significance level.

In the previous paragraphs, Program 8.1 and Program 8.2 were presented. These programs demonstrate the usage of the t-test for testing the significance of treatment difference for the whole set of n-of-1 trials. In series of n-of-1 trials there is the possibility and much interest in testing the significance of treatment difference for each subject taking part in each n-of-1 trial of the series. So an individual treatment effect can be estimated, and inferences at the individual level can be made. One is interested in testing the null hypothesis that the individual treatment effect equals zero. The ‘ttest’ procedure offers the possibility of performing a t-test for each level of a categorical variable. To perform a t-test for each patient in the series of n-of-1 trials a ‘by’ statement with the variable “Patient” can be added to either Program 8.1 or Program 8.2. The following programs demonstrate the usage of the ‘by’ statement in ‘proc ttest’.

Program 8.3: Performing a two-sided per patient paired t-test with 5% significance level.

```
title "Two-sided per patient paired t-test 5% significance level";
proc ttest
        data=NORMAL.ddata
        alpha=0.05
        dist=normal
        h0=0
        sides=2
        test=diff
        ci=none
        plots=none;
    paired YB*YA;
    by Patient;
run;
```

Program 8.3 is very similar to Program 8.1 with the addition of the ‘by’ statement. Therefore, there is not much to explain here that is not explained above in respect to Program 8.1. Alternatively, variable “dY” from “NORMAL.ddata” can be used instead of variables “YB” and “YA” in the same dataset. To perform a t-test for each patient while using variable “dY” a ‘by’ statement with variable “Patient” can be added to Program 8.2 leading to the following program.

Program 8.4: Performing an alternative two-sided per patient paired t-test with 5% significance level.

```
title "Alternative two-sided per patient paired t-test 5% significance level";
proc ttest
                data=NORMAL.ddata
                alpha=0.05
                dist=normal
                h0=0
                sides=2
                ci=none
                plots=none;
        var dY;
        by Patient;
run;
```

Program 8.3 and Program 8.4 give the exact same results. In this case, the output is very long with twelve sets of three tables similar to Table 8.1, Table 8.2 and Table 8.3. Due to its extensiveness, the output from either Program 8.3 or Program 8.4 is not be presented here. However, note that the values of the individual means, standard errors, t statistics, p-values and confidence interval of the means obtained from Program 8.3 and Program 8.4 are the same as the ones presented in Table 7.1 resulting from Program 7.1.

Note that the t-test assumes the observations are independent and normally distributed. For measurements taken from the same individual, this assumption is questionable. Once again, it is very important to stress that performing a t-test for each patient is a method that uses only the information of that patient. Alternative methods exist which use more information and are therefore considered more efficient than these are. The alternative more efficient methods are presented on another section of this document.

9 A summary measures approach

A summary measures approach can be accomplished in two steps. In the first step, the individual means are obtained. A t-test is applied on the individual means in the second step. Program 7.1 computes and saves the dataset “NORMAL.sddata”, which contains variable “meanY” holding the individual means. The individual means present in “NOMAL.sddata” are the naive individual treatment effects. In the following program, the t-test is applied on these individual means. Note however that alternative methods can be used to obtain the individual means supplied to the t-test.

Program 9.1: Performing the summary measures approach.

```
title "Summary measures approach for balanced data";
proc ttest
                data=NORMAL.sddata
                alpha=0.05
                dist=normal
                h0=0
                sides=2
                ci=none
                plots=none;
        var meanY;
run;
```

In Program 9.1, the ‘ttest’ procedure is applied on the “NORMAL.sddata” dataset previously created by Program 7.1. The significance level is 0.05 according to the ‘alpha’ option. The ‘dist’ option specifies the distribution of the data, which in this case it is normal. A null value of zero is defined in the ‘h0’ option. It is a two-sided test according to the ‘sides’ option. The ‘ci’ option in this case suppresses the computation and display of the confidence interval of the standard error of the mean. No plots are desired in the output so the ‘plots’ option is set to “none”. Finally, the ‘var’ statement defines variable “meanY” as the variable of interest. Variable “meanY” holds the naive individual treatment effects. The ‘ttest’ procedure in Program 9.1 results in the three tables presented below.

Table 9.1: Summary measures approach 5% significance level – Mean.

N	Mean	Std Dev	Std Err	Minimum	Maximum
12	188.7	98.3242	28.3838	50.0000	348.0

Table 9.2: Summary measures approach 5% significance level- CL Mean.

Mean	95% CL Mean	
188.7	126.2	251.2

Table 9.3: Summary measures approach 5% significance level - t statistic.

DF	t Value	Pr > |t|
11	6.65	<.0001

The summary measures approach can now be compared with the paired t-test presented on section 8. Note that the degrees of freedom in Table 9.3 are equal to 11 instead of 35 as in Table 8.3. This is due to the lower number of observations present in the later dataset. There is one observation per patient in the "NORMAL.sddata" dataset used in the summary measures approach. The number of parameters is still the same in both the paired t-test and the summary measures approach. The overall mean is the only parameter considered in both methods. Table 8.2 and Table 9.2 indicate that the estimate of the mean is the same for both methods. However, the confidence interval for the mean is wider and the standard error of the mean is larger for the summary measures approach. The higher estimate of the standard error of the overall mean observed here does not necessarily indicate that the summary measures approach is less precise than the paired t-test discussed in section 8. Actually, for balanced data both methods are equally precise as regards the estimation of the overall treatment effect, since both methods lead to exactly the same estimate in this situation. The precision of the estimate of the standard error of the overall treatment effect is the one that is put in question. In fact, the estimate of the standard error of the overall mean resulting from the paired t-test presented in section 8 can be downwards biased in certain situations. See Araujo, Julious [20] for a more elaborate discussion of this issue. The t statistic decreased by a small amount and the p-value is still very low when compared to the significance level, suggesting the rejection of the null hypothesis and the same conclusion as the one suggested by the t-test presented in either Program 8.1 or Program 8.2.

An advantage of this approach is as regards the calculation of the confidence interval. As mentioned in section 8 of this document and discussed in section 7 of Senn and Hildebrand [19], under the strict null hypothesis that the treatments are identical for every patient, the matched pairs t-test is valid. However, as soon as we consider the possibility that the treatment effect is not zero, which is what is done by calculating a confidence interval, then the strict null hypothesis is abandoned. It becomes plausible to believe that the treatment effect might vary from patient to patient. The summary measures approach makes allowance for this.

Note that the t-test and hence the summary measures approach estimate the mean by equally weighting the observations. Estimates of the mean that are more efficient can be obtained by weighting the observations according to their relative precision. For balanced data as is the case for the "NORMAL.sddata" dataset used in Program 9.1 the same number of observations is used to obtain the individual means, so each individual mean provides the same information and the same relative precision can be safely assumed for each mean. Therefore, for balanced series of n-of-1 trials the summary measures approach can be considered a valid option. However, for unbalanced series of n-of-1 trials, methods of estimation that consider the value of the information are regarded as more efficient. Statistical methods more adequate for unbalanced series of n-of-1 trials, which are more efficient, are considered on a separate dedicated section.

10 Analysis of variance

The SAS software provides several procedures that can be used to carry out analysis of variance. In this section, the application of the ‘anova’ and the ‘glm’ procedures to data arising form series of n-of-1 trials is demonstrated. The ‘anova’ procedure can be used to perform analysis of variance for balanced designs. While the ‘glm’ procedure can be used to perform analysis of variance for both balanced and unbalanced designs. The following program demonstrates the application of the ‘anova’ procedure to the balanced dataset “NORMAL.ndata” previously created by either Program 4.1 or Program 5.4.

Program 10.1: Performing an analysis of variance using ‘proc anova’.

```
title "ANOVA Cycle within Patient as block structure - proc anova";
proc anova data=NORMAL.ndata plots=none;
        class Treatment Patient Cycle;
        model Y = Treatment Patient Cycle(Patient);
        means Treatment;
run;
quit;
```

The variables of interest in the dataset “NORMAL.ndata” are the outcome variable “Y” and the categorical variables “Treatment”, “Patient” and “Cycle”. The dataset “NORMAL.ndata” arising from the series of n-of-1 trials is balanced, in the sense that for each and every patient the number of observations for the variables “Treatment” and “Cycle” is the same. The dataset is balanced so the ‘anova’ procedure can be applied. In the top line in Program 10.1 lies the ‘title’ statement. The purpose of the title statement is to identify the program’s output. The balanced dataset “NORMAL.ndata” is supplied to the ‘data’ option in the ‘anova’ statement. No plots are desired in the software’s output so the ‘plots’ option is set to “none” in the ‘proc anova’ statement. The ‘class’ statement identifies all the categorical variables used in the analysis, in this case the variables “Treatment”, “Patient” and “Cycle”. The ‘model’ statement defines the actual model. In the ‘model’ statement the dependent variable is written first followed by the equal sign and the explanatory variables. The main purpose is to check the significance of the variable “Treatment” in explaining the variation in the outcome variable “Y”, hence the presence of the “Treatment” variable in the right side of the equation. A variation of the outcome variable between patients is also expected hence the variable “Patient” as explanatory variable in the model. In this design, the measurements of the variable “Cycle” are nested within the variable “Patient” so a final term is added to the model to reflect this. Note how the “Cycle” within “Patient” block structure is written in the ‘model’ statement of the ‘anova’ procedure in Program 10.1. The ‘means’ statement requests the means for each level of the “Treatment” categorical variable. The actual value of the mean of outcome variable estimated under each treatment is a key element in assisting the decision of treatment preference. A more elaborate discussion of this decision process is presented later.

The ‘quit’ statement terminates the procedure. The ‘anova’ procedure outputs six tables. The first two tables contain information relative to the dataset, the following three tables contain information relative to the actual analysis of variance, and the last table presents the estimated means and standard deviation of the outcome variable “Y” under each level of the categorical variable “Treatment”.

Table 10.1: Analysis of variance using ‘proc anova’ – Class level information.

Class Level Information		
Class	**Levels**	**Values**
Treatment	2	A B
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12
Cycle	3	1 2 3

Table 10.2: Analysis of variance using ‘proc anova’ – Number of observations.

Number of Observations Read	72
Number of Observations Used	72

Table 10.3. Analysis of variance using ‘proc anova’ – Model SS.

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	36	2416765.500	67132.375	5.30	<.0001
Error	35	443735.611	12678.160		
Corrected Total	71	2860501.111			

Table 10.4: Analysis of variance using ‘proc anova’ – Model information.

R-Square	Coeff Var	Root MSE	Y Mean
0.844875	4.139439	112.5973	2720.111

Table 10.5: Analysis of variance using ‘proc anova’ – Anova SS.

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Treatment	1	641089.389	641089.389	50.57	<.0001
Patient	11	1458791.444	132617.404	10.46	<.0001
Cycle(Patient)	24	316884.667	13203.528	1.04	0.4479

Table 10.6: Analysis of variance using ‘proc anova’ – Means.

Level of Treatment	**N**	**Y**	
		Mean	**Std Dev**
A	**36**	2625.75000	181.528176
B	**36**	2814.47222	174.525886

Table 10.1 is the first one being outputted and contains information on the categorical variables involved in the analysis. Table 10.1 provides the number of levels for each categorical variable used in the analysis as well as the actual value of each level present in the respective categorical variable. The number of observations read from the dataset and the number of observations used in the analysis is presented on Table 10.2. The analysis of variance results for the model and error sources of variations is presented in Table 10.3. While useful statistics such as the coefficient of determination and the coefficient of variation are presented in Table 10.4. The final table contains the analysis of variance results for the various terms used in the model. Table 10.3 and Table 10.5 contain the most useful information, which allow the assessment of the significance of the various terms used in the model. Observe from Table 10.5 that the “Treatment” and the “Patient” terms are both significant while the “Cycle” within “Patient” term is not. Note that on Table 10.3 the degrees of freedom for the error term are the same as the degrees of freedom resulting from the paired t-test presented on Table 8.3. Also note that the F statistic on Table 10.5 equals the square of the t statistic given by the two-sided paired t test as presented on Table 8.3. The F test considers a null hypothesis of equality of the means estimated under two levels of a category variable against an alternative hypothesis of difference of the means considered. The p-value resulting from the F test is significantly lower than the usual 0.05 significance level, suggesting that the means estimated under the two treatments differ. One treatment should be preferred over the other. Table 10.6 provides the information needed to decide which treatment should be preferred. Observe that the outcome variable is on average higher under treatment “B” than under treatment “A”. Therefore, if a patient feels better when the outcome variable registers higher values, treatment “B” shall be preferred over treatment “A”. If it is the case that the quality of life of the patient is better when lower values of the outcome variable are observed, then the choice shall fall over treatment “A”.

The analysis of variance through the ‘glm’ procedure is demonstrated in the next program. The ‘glm’ procedure was developed for unbalanced designs, so the unbalanced dataset “NORMAL.ndata_unbalanced” created by either Program 4.4 or Program 5.5 is used as example.

Program 10.2: Performing an analysis of variance using 'proc glm'.

```
title "ANOVA Cycle within Patient as block structure - proc glm";
proc glm data=NORMAL.ndata_unbalanced plots=none;
	class Treatment Patient Cycle;
	model Y = Treatment Patient Cycle(Patient);
	means Treatment;
run;
quit;
```

Usage of the 'glm' procedure in Program 10.2 is very similar to the 'anova' procedure presented in Program 10.1. Except for the 'proc' statement and the dataset used the other statements and options remain unaltered. In fact, the statements and options used in the 'glm' procedure presented in Program 10.2 have the same meaning and purpose as the ones used in the 'anova' procedure of Program 10.1, so no further explanations are presented here. Please refer to the paragraph around Program 10.1 or the SAS software's documentation for details about the statements and options used in Program 10.2 [10]. Seven tables are outputted after running Program 10.2. The first two tables contain information relative to the dataset, the following four tables contain information relative to the actual analysis of variance, and the last table presents the estimated means and standard deviation of the outcome variable "Y" under each level of the categorical variable "Treatment".

Table 10.7: Analysis of variance using 'proc glm' – Class level information.

Class Level Information		
Class	**Levels**	**Values**
Treatment	2	A B
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12
Cycle	3	1 2 3

Table 10.8: Analysis of variance using 'proc glm' – Number of observations.

Number of Observations Read	66
Number of Observations Used	66

Table 10.9: Analysis of variance using 'proc glm' - Model SS.

Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	33	2386261.727	72310.961	5.52	<.0001
Error	32	419000.091	13093.753		
Corrected Total	65	2805261.818			

Table 10.10: Analysis of variance using 'proc glm' - Model information.

R-Square	**Coeff Var**	**Root MSE**	**Y Mean**
0.850638	4.208738	114.4279	2718.818

Table 10.11: Analysis of variance using 'proc glm' - Type I SS.

Source	**DF**	**Type I SS**	**Mean Square**	**F Value**	**Pr > F**
Treatment	1	624490.909	624490.909	47.69	<.0001
Patient	11	1448198.902	131654.446	10.05	<.0001
Cycle(Patient)	21	313571.917	14931.996	1.14	0.3607

Table 10.12: Analysis of variance using 'proc glm' - Type III SS.

Source	**DF**	**Type III SS**	**Mean Square**	**F Value**	**Pr > F**
Treatment	1	624490.909	624490.909	47.69	<.0001
Patient	11	1448198.902	131654.446	10.05	<.0001
Cycle(Patient)	21	313571.917	14931.996	1.14	0.3607

Table 10.13: Analysis of variance using 'proc glm' - Means.

Level of Treatment	**N**	**Y**	
		Mean	**Std Dev**
A	**33**	2621.54545	188.010653
B	**33**	2816.09091	181.110699

Table 10.7 contains the exact same information as Table 10.1. The dataset used in Program 10.2 was obtained through the deletion of some observations on the dataset used in Program 10.1, so the class level information does not change. The number of observations however was reduced from 72 in Table 10.2 to 66 in Table 10.8. The degrees of freedom of the “Cycle” within “Patient” term was reduced from 24 in Table 10.5 to 21 in Table 10.11 and Table 10.12 reflecting the deletion of data from 3 cycles while creating the unbalanced

dataset. The results for the type I sums of squares presented in Table 10.11 and the results from the type III sums of squares presented in Table 10.12 are the same in this case. For distinct models and datasets, the type I and type III sums of squares might differ. The p value for the “Treatment” term remains much lower than the usual 0.05 significance level suggesting that the two treatments differ significantly from each other. The conclusions drawn from the results of Program 10.2 are essentially the same as the ones written above in respect to the results of Program 10.1.

The next and final example of the usage of the ‘glm’ procedure uses a term with an interaction in the model. Considering that, the outcome variable varies between both the treatment and the patient simultaneously and between the cycle and the patient simultaneously, a fixed “Treatment” by “Patient” interaction and a fixed “Cycle” by “Patient” interaction are considered in the model. The “NORMAL.ndata” dataset is used in the following program.

Program 10.3: Performing an analysis of variance with interactions using ‘proc glm’.

```
title "ANOVA Treatment by Patient and Cycle by Patient interaction - proc glm";
proc glm data=NORMAL.ndata plots=none;
	class Treatment Patient Cycle;
	model Y = Treatment Patient Treatment*Patient Cycle*Patient;
	means Treatment;
run;
quit;
```

Program 10.3 is very similar to Program 10.2, the most notable difference being the ‘model’ statement. In the ‘model’ statement of Program 10.3, there are two interaction terms. There is a “Treatment” by “Patient” interaction term and a “Cycle” by “Patient” interaction term. Observe that an asterisk is written between the variables to denote an interaction. The submission of Program 10.3 leads to the following seven tables.

Table 10.14: Analysis of variance with interactions using 'proc glm' - Class level information.

Class Level Information		
Class	**Levels**	**Values**
Treatment	2	A B
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12
Cycle	3	1 2 3

Table 10.15: Analysis of variance with interaction using 'proc glm' - Number of observations.

Number of Observations Read	72
Number of Observations Used	72

Table 10.16: Analysis of variance with interactions using 'proc glm' - Model SS.

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	47	2576281.778	54814.506	4.63	<.0001
Error	24	284219.333	11842.472		
Corrected Total	71	2860501.111			

Table 10.17: Analysis of variance with interactions using 'proc glm' - Model information.

R-Square	Coeff Var	Root MSE	Y Mean
0.900640	4.000687	108.8231	2720.111

Table 10.18: Analysis of variance with interactions using 'proc glm' - Type I SS.

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Treatment	1	641089.389	641089.389	54.13	<.0001
Patient	11	1458791.444	132617.404	11.20	<.0001
Treatment*Patient	11	159516.278	14501.480	1.22	0.3241
Patient*Cycle	24	316884.667	13203.528	1.11	0.3960

Table 10.19: Analysis of variance with interactions using 'proc glm' - Type III SS.

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Treatment	1	641089.389	641089.389	54.13	<.0001
Patient	11	1458791.444	132617.404	11.20	<.0001
Treatment*Patient	11	159516.278	14501.480	1.22	0.3241
Patient*Cycle	24	316884.667	13203.528	1.11	0.3960

Table 10.20: Analysis of variance with interactions using 'proc glm' - Means.

Level of Treatment	N	Y	
		Mean	Std Dev
A	**36**	2625.75000	181.528176
B	**36**	2814.47222	174.525886

The type I sums of squares results present on Table 10.18 are the same as the type III sums of squares results present on Table 10.19. The “Treatment” variable remains significant in the model suggesting that the two treatments differ significantly from each other as regards their effect on the outcome. The “Patient” term is significant while the “Treatment” by “Patient” and the “Cycle” by “Patient” interactions are not. However, the coefficient of determination in Table 10.17 increased when compared to the coefficient of determination of the model without the interaction terms presented in Table 10.4. The mean of outcome remains higher under treatment “B” than under treatment “A”. The conclusions are evidently the same as the ones obtained from the results of the two previous programs used as examples in this section.

11 Linear fixed-effects model

In this section, attention is directed to the estimation of overall and individual treatment effects through linear fixed-effects models. Example programs are demonstrated and the results compared with the ones obtained from the methods exemplified above. Linear fixed-effects models contain one or more fixed terms and a single residual error. The normal distribution is assumed for the residual error of the models used in the example programs of this section.

11.1 Estimation and inference on the overall treatment effect

The ‘glm’ procedure can be used to fit linear fixed-effects regression models, and is used in some of the following programs. For starters, attention is given to the estimation of an overall treatment effect, i.e. the difference between the means of an outcome variable estimated under two distinct treatments. An overall treatment effect can be estimated through a linear fixed-effects model by fitting the variable indicating the treatment as explanatory variable. In the following program a model with a “Treatment”, “Patient”, “Treatment” by “Patient” interaction and a “Cycle” by “Patient” interaction is fitted to the “NORMAL.ndata” balanced dataset previously obtained from either Program 4.1 or Program 5.4.

Program 11.1: Fitting a linear fixed-effects model using ‘proc glm’- Balanced data.

```
title "Balanced data";
proc glm
        data=NORMAL.ndata
        alpha=0.05 /* level for CI */
        plots=none; * suppress plots ;
    class Treatment (ref=first) Patient Cycle;
    model Y = Treatment Patient Patient*Treatment Patient*Cycle /
        alpha=0.05 /* level for CI */
        clparm; * CI for parameter estimates ;
    lsmeans Treatment /
        alpha=0.05 /* level for CI */
        cl /* t-type CI for least squares means */
        pdiff /* p-values for differences */
        stderr /* standard error */
        tdiff; * t values ;
    estimate 'B-A'
        Treatment               1 -1;
run;
quit;
```

There are a great number of options written under some of the statements in Program 11.1; as such, each option is not explained in this text. Detailing all the options used in programs would make this document very large, repetitive, and possibly boring for the reader. In the programs presented, comments are kept on the same line as the actual options to indicate the purpose of each option. If there are any doubts as regards the usage of the statements and their options, the SAS software's documentation available on the World Wide Web constitutes a valuable source of information [10]. Whenever a new statement is introduced or whenever it is deemed appropriate further explanations are provided. The 'glm' procedure in Program 11.1 is very similar to the one in Program 10.3. In Program 11.1, an 'lsmeans' statement is used instead of the 'means' statement used in Program 10.3. The 'lsmeans' statement followed by the variable "Treatment" instructs the software to output the overall treatment effect and several related statistics of interest. While the 'estimate' statement also requests the estimation of the overall treatment effect. The 'estimate' statement is a very interesting and flexible statement available in 'proc glm' and some other procedures. The 'estimate' statement permits the computation of linear combinations of the mean of outcome under several levels of categorical variables present in the model. In the 'estimate' statement of Program 11.1 the coefficient under treatment "B" is 1 and the coefficient under treatment "A" is -1. Thus, the average outcome under treatment "B" is multiplied by 1, and the average outcome under treatment "A" is multiplied by -1, being the two averages summed to yield the final statistic. Therefore, the statistic requested by the 'estimate' statement is actually the difference of mean outcome under the two treatments, or in fewer words the overall treatment effect. A label "B-A" is defined in the 'estimate' statement. Multiple 'estimate' statements can be written under the same 'glm' procedure. A label permits the identification of the requested statistic in the software's output. Program 11.1 outputs ten tables. Since the exact same dataset and model is used in Program 11.1 as in Program 10.3, the same tables are outputted except Table 10.20 resulting from the 'means' statement absent from Program 11.1. So Table 10.14, Table 10.15, Table 10.16, Table 10.17, Table 10.18, and Table 10.19 are also obtained from Program 11.1 with the exact same results on them. The following additional tables are outputted after submitting Program 11.1 for execution.

Table 11.1: Least squares means – Hypothesis test.

Treatment	Y LSMEAN	Standard Error	H0:LSMEAN=0	H0:LSMean1=LSMean2	
			Pr > |t|	t Value	Pr > |t|
B	2814.47222	18.13719	<.0001	7.36	<.0001
A	2625.75000	18.13719	<.0001		

Table 11.2: Least squares means - Confidence limits.

Treatment	Y LSMEAN	95% Confidence Limits	
B	2814.472222	2777.038908	2851.905537
A	2625.750000	2588.316685	2663.183315

Table 11.3: Least squares means for effect treatment.

Least Squares Means for Effect Treatment				
i	j	Difference Between Means	95% Confidence Limits for LSMean(i)-LSMean(j)	
1	2	188.722222	135.783521	241.660924

Table 11.4: Overall treatment effect - Balanced data.

Parameter	Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
B-A	188.722222	25.6498562	7.36	<.0001	135.783521	241.660924

The ‘lsmeans’ statement in Program 11.1 leads to Table 11.1, Table 11.2, and Table 11.3, while the ‘estimate’ statement in the same Program 11.1 leads to Table 11.4. In clinical trials, one is interested in estimates of average outcome difference under treatments rather than estimates of average outcome under a specific treatment. Note that an estimate of a mean outcome under treatment “A” or treatment “B” is potentially biased. While the estimate of the mean outcome difference under the two treatments is potentially unbiased. A clinical trial can be designed so that the bias of the mean outcome under all the treatments has the same or identical bias. In such a case, the bias is eliminated by the difference operation, so that the bias of the resulting treatment effect estimate is minimized. The information contained in Table 11.2 is of little interest. Table 11.4 resulting from the ‘estimate’ statement contains all the information needed, and the ‘lsmeans’ statement in Program 11.1 is not required. Both the ‘lsmeans’ and the ‘estimate’ statements are used in Program 11.1 to demonstrate the two alternatives. Note that the overall treatment effect and its confidence limits are present in both Table 11.3 and Table 11.4, while the t statistic and the p-value are present in both Table 11.1and Table 11.4. The standard error of the overall treatment effect is present only on Table 11.4. Also, note that the t statistic in Table 11.4 is equal to the square root of the F statistic from Table 10.18 and Table 10.19. As regards the interpretation of the results, the p-value is much lower than the usual 0.05 significance level and the confidence interval of the overall treatment effect does not contain the origin, suggesting that the treatments differ significantly from each other. The mean outcome under treatment “B” minus the mean outcome under treatment “A”, defines the overall treatment effect. As such, a positive overall treatment effect indicates that the mean outcome is greater under treatment “B” than under treatment “A”. Therefore, if a patient feels better when the outcome variable registers higher values, treatment “B” shall be preferred over treatment “A”. If it is the case that the quality of life of the patient is better when lower values of the outcome variable are observed, then the choice shall fall over treatment “A”.

The next program fits the same model used in Program 11.1 to the unbalanced dataset “NORMAL.ndata_unbalanced” created by either Program 4.4 or Program 5.5. This way the results can be compared to the ones relative to the balanced dataset obtained previously.

Program 11.2: Fitting a linear fixed-effects model using ‘proc glm’- Unbalanced data.

```
title "Unbalanced data";
proc glm
                data=NORMAL.ndata_unbalanced
                alpha=0.05 /* level for CI */
                plots=none; * suppress plots ;
        class Treatment (ref=first) Patient Cycle;
        model Y = Treatment Patient Patient*Treatment Patient*Cycle /
                alpha=0.05 /* level for CI */
                clparm; * CI for parameter estimates ;
        estimate 'B-A'
                Treatment                       1 -1;
run;
quit;
```

Little changes are made from Program 11.1 to Program 11.2. The most notable difference is the dataset which is “NORMAL.ndata_unbalanced” in Program 11.2, being the other difference the absence of the ‘lsmeans’ statement in the later program. There is nothing new worth mentioning that has not been previously mentioned in respect to the used statements and options. Program 11.2 outputs the seven tables presented below.

Table 11.5: Class level information.

Class Level Information		
Class	**Levels**	**Values**
Treatment	2	B A
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12
Cycle	3	1 2 3

Table 11.6: Number of observations.

Number of Observations Read	66
Number of Observations Used	66

Table 11.7: Model SS.

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	44	2543886.568	57815.604	4.65	0.0002
Error	21	261375.250	12446.440		
Corrected Total	65	2805261.818			

Table 11.8: Model information.

R-Square	Coeff Var	Root MSE	Y Mean
0.906827	4.103386	111.5636	2718.818

Table 11.9: Type I SS.

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Treatment	1	624490.909	624490.909	50.17	<.0001
Patient	11	1448198.902	131654.446	10.58	<.0001
Treatment*Patient	11	157624.841	14329.531	1.15	0.3743
Patient*Cycle	21	313571.917	14931.996	1.20	0.3402

Table 11.10: Type III SS.

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Treatment	1	553365.313	553365.313	44.46	<.0001
Patient	11	1448198.902	131654.446	10.58	<.0001
Treatment*Patient	11	157624.841	14329.531	1.15	0.3743
Patient*Cycle	21	313571.917	14931.996	1.20	0.3402

Table 11.11: Overall treatment effect - Unbalanced data.

Parameter	Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
B-A	192.736111	28.9054483	6.67	<.0001	132.623941	252.848282

When there are interactions in the model and the data is unbalanced is when the type I and type III sums of squares tables might differ. Note that the values in the row relative to the “Treatment” term source of variations differ between Table 11.9 and Table 11.10. Also, note that the F statistic in Table 11.10 is equal to the square of the t statistic in Table 11.11. Therefore, the ‘estimate’ statement in Program 11.2 yields results that are equivalent to the type III sums of squares analysis. On both the type I and the type III sums of squares tables the p-value relative to the “Treatment” term is still much lower than the most commonly used 0.05 significance level, suggesting that the treatments “A” and “B” differ significantly one from the other. The same conclusion can be drawn from the p-value and the confidence limits on Table 11.11. Also, note that the standard error of the overall treatment effect estimate on Table 11.11 is higher than the standard error of the overall treatment effect estimate on Table 11.4. This increase in the standard error of the overall treatment effect estimate reflects the decrease in information from deleting the observations when creating the unbalanced dataset with either Program 4.4 or Program 5.5.

The ‘estimate’ statement in Program 11.2 produces an unbiased estimate of the overall treatment effect by equally weighting the means computed from the observations of each individual and treatment. An equally unbiased estimate with lower variance and therefore higher efficiency can be obtained by weighting the individual independent means according to their precision. For a model with only one error term such as the model in question, the precision-based weights only depend on the number of observations available for a particular individual and treatment. Such efficient estimate of the overall treatment effect can be obtained within the SAS software through the definition of an appropriate ‘estimate’ statement under the ‘glm’ procedure. The exact number of observations available under each patient and treatment needs to be determined first. The “NORMAL.ndata_unbalanced” dataset used in Program 11.2 is relatively small so the dataset can be printed and the observations can be counted very easily. For very large datasets printing and counting the observations is extremely difficult. Fortunately, very simple programs can be written to let the computer handle tasks that are more difficult for the common human mind. The next little program determines the number of observations available under both treatments for the unbalanced dataset.

Program 11.3: Determining the number of observations per treatment.

```
title "Unbalanced data";
title2 "Number of observations per Treatment";
proc sql;
        select          Treatment,
                        count(Y) as nobs
        from NORMAL.ndata_unbalanced
        group by Treatment;
quit;
```

In Program 11.3, the ‘sql’ procedure is used to query the “NORMAL.ndata_unbalanced” dataset. The variable “Treatment” is selected from the dataset, and the ‘count’ function is used to determine the number of observations available for variable “Y” present on the same dataset. The number of observations resulting from the ‘count’ function is saved under the variable “nobs”. The usage of the keywords ‘group’ and ‘by’ followed by the variable name in this case “Treatment” implies that the ‘count’ function is computed for each distinct treatment. The table below results from Program 11.3.

Table 11.12: Number of observations per treatment - Unbalanced data.

Treatment	**nobs**
A	33
B	33

Table 11.12 shows that 33 observations are available under both treatments present on the unbalanced dataset. This number determines the denominator of the weights used to compute the estimate of the overall treatment effect. The number of observations available under each patient and treatment determines the numerator of the referred weights. The following program counts the observations available under each patient and treatment on the unbalanced dataset.

Program 11.4: Determining the number of observations per treatment and patient.

```
title "Unbalanced data";
title2 "Number of observations per Treatment and Patient";
proc sql;
        select          Patient,
                        Treatment,
                        count(Y) as nobs
        from NORMAL.ndata_unbalanced
        group by Treatment, Patient;
quit;
```

Program 11.4 is very similar to Program 11.3 so only the differences between the two programs are mentioned. In Program 11.4, the results of the ‘select’ statement are grouped by the two variables “Treatment” and “Patient”. Note the presence of the comma (,) between the two variables placed after the ‘group’ and ‘by’ keywords. It is often useful to enumerate the grouping variables in the ‘select’ statement, so in this case the selection of the variables “Patient” and “Treatment” from the dataset permits the association of the observation count to the respective patient and treatment. The interpretation of Program 11.4 by the SAS software produces the table presented below.

Table 11.13: Number of observations per treatment and patient - Unbalanced data.

Patient	Treatment	nobs
1	A	3
2	A	3
3	A	3
4	A	3
5	A	3
6	A	3
7	A	3
8	A	3
9	A	3
10	A	3
11	A	2
12	A	1
1	B	3
2	B	3
3	B	3
4	B	3
5	B	3
6	B	3
7	B	3
8	B	3
9	B	3
10	B	3
11	B	2
12	B	1

Given the availability of the results displayed in Table 11.12 and Table 11.13, a new program can now be written with the purpose of computing a more efficient estimate of the overall treatment effect. The next program makes use of these results.

Program 11.5: Fitting a linear fixed-effects model using ‘proc glm’- Weighted estimate.

```
title "Unbalanced data";
title2 "Weighted estimate";
proc glm
                data=NORMAL.ndata_unbalanced
                alpha=0.05 /* level for CI */
                plots=none; * suppress plots ;
        class Treatment (ref=first) Patient Cycle;
        model Y = Treatment Patient Patient*Treatment Patient*Cycle /
                alpha=0.05 /* level for CI */
                clparm /* CI for parameter estimates */
                ss1
                ss2
                ss3;
        estimate 'B-A'
                Treatment                33 -33
                Patient*Treatment         3  3  3  3  3  3  3  3  3  3  2  1
                                         -3 -3 -3 -3 -3 -3 -3 -3 -3 -3 -2 -1 /
                divisor=33;
run;
quit;
```

The ‘glm’ procedure in Program 11.5 is very similar to the one in Program 11.2, the most notable differences being located in the ‘estimate’ statement. Three new options are added to the ‘model’ statement in Program 11.5. The ‘ss1’, ‘ss2’ and ‘ss3’ options request the type I, type II and type III sums of squares tables respectively. The ‘estimate’ statement in Program 11.5 is more elaborate than the one in Program 11.2. Immediately after the ‘estimate’ keyword, a label is specified. The label permits the identification of the desired estimate in the SAS software’s output. Here the observation numbers previously determined by Program 11.3 and Program 11.4 and presented in Table 11.12 and Table 11.13 are used to define the weights given to each fixed effect statistic with the purpose of computing the overall treatment effect estimate. All the terms of the model where the treatment factor variable is involved are specified in the ‘estimate’ statement followed by the weights given to each level of the factor. After writing the “Treatment” variable, the numbers defining the weights are specified. In this case, 33 is specified for the number of observations under treatment “B”. The mean outcome under treatment “B” minus the mean outcome under treatment “A” is desired, so the number of observations under treatment “A” preceded by the negative sign is written afterwards, -33 in this case. As regards the treatment by patient interaction term “Patient*Treatment”, the observation numbers previously obtained from Program 11.4 and displayed in Table 11.13 are placed in a position respective to each patient and treatment. The first row of numbers after the “Patient*Treatment” term in Program 11.5 is relative to treatment “B”, while the second row respective to the same “Patient*Treatment” term is relative to treatment “A”. The column defines the position of the patient’s coefficient. The first column refers to the first patient, the second column is relative to the second patient, and so on, until the twelfth column which is respective to the twelfth patient. Note in the second row that a negative sign precedes the observation numbers for each patient under the reference treatment “A”. In fact, the weights must lie between zero and unity, but the ‘estimate’ statement does not support the specification of non-integer numbers as weights. The ‘estimate’ statement supports the specification of the ‘divisor’ option to define a denominator common to all the weights in the statement. So all the observation numbers relative to each term specified in the ‘estimate’ statement are divided by 33, the total number of observations. Note that the weights respective to a specific term in the ‘estimate’ statement sum to zero. Also note that the weights respective to either the “Treatment” term or the “Patient*Treatment” term sum to 1 under treatment “B”, and sum to -1 under treatment “A”. Program 11.5 produces Table 11.5, Table 11.6, Table 11.7, Table 11.8, Table 11.9, and Table 11.10 as obtained from Program 11.2. A new table is obtained for the type II sums of squares, and a table similar to Table 11.11 is obtained. The obtained tables are presented below.

Table 11.14: Type II SS.

Source	DF	Type II SS	Mean Square	F Value	Pr > F
Treatment	1	624490.909	624490.909	50.17	<.0001
Patient	11	1448198.902	131654.446	10.58	<.0001
Treatment*Patient	11	157624.841	14329.531	1.15	0.3743
Patient*Cycle	21	313571.917	14931.996	1.20	0.3402

Table 11.15: Overall treatment effect- Weighted estimate.

Parameter	Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
B-A	194.545455	27.4650637	7.08	<.0001	137.428728	251.662181

In this case the type II sums of squares Table 11.14 contains the exact same results as the type I sums of squares Table 11.9. The weighted estimate presented in Table 11.15 is distinct from the unweighted one displayed in Table 11.11. As expected, the standard error of the weighted estimate is lower than the standard error of the unweighted one. In addition, the confidence interval of the weighted estimate is narrower than the confidence interval of the unweighted estimate. Remember the relationship between the F statistic of the unweighted estimate in the type III sums of squares Table 11.10 and the t statistic of the same estimate in Table 11.11. The t statistic associated to the weighted estimate is no longer related to the type III sums of squares, but related to the type II sums of squares. Therefore, the F statistic in Table 11.14 equals the square of the t statistic in Table 11.15. The weighted estimate displayed in Table 11.15 is more efficient than the unweighted one from Table 11.11 given the assumptions inherent to the model. For this reason, the weighted estimator is recommended over the unweighted one. The p value is still much lower than the conventional 0.05 significance level, so the conclusions previously drawn from Table 11.11 remain valid.

11.2 Estimation and inference on individual treatment effects

Individual treatment effects can be estimated from data arising from series of n-of-1 trials. So that a treatment preference can be determined for each patient, based on its own treatment effect rather than on the overall treatment effect, depending on the information available for that particular patient. For any statistical model, the individual treatment effect is defined as the expectation of the outcome variable conditioned on a given patient and treatment minus the expectation of the outcome variable conditioned on the same patient and another treatment. Given this definition, for a linear model such as the one used in Program 11.1, Program 11.2 and Program 11.5, the individual treatment effects depend only on the "Treatment" term and on the "Treatment" by "Patient" interaction term. The "Treatment" by "Patient" term is distinct for each patient and is the reason for the distinctiveness of the individual treatment effects. A model with a "Treatment" term and without a "Treatment" by "Patient" interaction term results in the same individual treatment effect for every patient in the sample. In such a model, the individual treatment effect coincides with the overall treatment effect. When estimating individual treatment effects form a linear model with the purpose of individualizing medicine it is therefore recommended to include a treatment by patient interaction term in the model. The following program can be considered an extension of Program 11.1, where the estimation of individual treatment effects for each patient present in the dataset is achieved. The code is based on code published by Brown and Prescott [21] for similar linear mixed-effects models applied to the analysis of multicentre trials. Linear mixed-effects models applied to the analysis of series of n-of-1 trials are presented on a separate section of the present document. The balanced data "NORMAL.ndata" created before by either Program 4.1 or Program 5.4 is used in the program below.

Program 11.6: Estimating individual treatment effects using 'proc glm' – The hard way.

```
title "Individual treatment effects";
title2 "Balanced data";
proc glm
                data=NORMAL.ndata
                alpha=0.05 /* level for CI */
                plots=none; * suppress plots ;
        class Treatment (ref=first) Patient Cycle;
        model Y = Treatment Patient Patient*Treatment Patient*Cycle /
                alpha=0.05 /* level for CI */
                clparm; * CI for parameter estimates ;
        estimate 'B-A,1' Treatment 1 -1
                Patient*Treatment       1 0 0 0 0 0 0 0 0 0 0 0
                                        -1 0 0 0 0 0 0 0 0 0 0 0;
        estimate 'B-A,2' Treatment 1 -1
                Patient*Treatment       0 1 0 0 0 0 0 0 0 0 0 0
                                        0 -1 0 0 0 0 0 0 0 0 0 0;
        estimate 'B-A,3' Treatment 1 -1
                Patient*Treatment       0 0 1 0 0 0 0 0 0 0 0 0
                                        0 0 -1 0 0 0 0 0 0 0 0 0;
        estimate 'B-A,4' Treatment 1 -1
                Patient*Treatment       0 0 0 1 0 0 0 0 0 0 0 0
                                        0 0 0 -1 0 0 0 0 0 0 0 0;
        estimate 'B-A,5' Treatment 1 -1
                Patient*Treatment       0 0 0 0 1 0 0 0 0 0 0 0
                                        0 0 0 0 -1 0 0 0 0 0 0 0;
        estimate 'B-A,6' Treatment 1 -1
                Patient*Treatment       0 0 0 0 0 1 0 0 0 0 0 0
                                        0 0 0 0 0 -1 0 0 0 0 0 0;
        estimate 'B-A,7' Treatment 1 -1
                Patient*Treatment       0 0 0 0 0 0 1 0 0 0 0 0
                                        0 0 0 0 0 0 -1 0 0 0 0 0;
        estimate 'B-A,8' Treatment 1 -1
                Patient*Treatment       0 0 0 0 0 0 0 1 0 0 0 0
                                        0 0 0 0 0 0 0 -1 0 0 0 0;
        estimate 'B-A,9' Treatment 1 -1
                Patient*Treatment       0 0 0 0 0 0 0 0 1 0 0 0
                                        0 0 0 0 0 0 0 0 -1 0 0 0;
        estimate 'B-A,10' Treatment 1 -1
                Patient*Treatment       0 0 0 0 0 0 0 0 0 1 0 0
                                        0 0 0 0 0 0 0 0 0 -1 0 0;
        estimate 'B-A,11' Treatment 1 -1
                Patient*Treatment       0 0 0 0 0 0 0 0 0 0 1 0
                                        0 0 0 0 0 0 0 0 0 0 -1 0;
        estimate 'B-A,12' Treatment 1 -1
                Patient*Treatment       0 0 0 0 0 0 0 0 0 0 0 1
                                        0 0 0 0 0 0 0 0 0 0 0 -1;
run;
quit;
```

To identify the output Program 11.6 starts with two ‘title’ statements. Program 11.6 is a variation of Program 11.1 without its ‘lsmeans’ and ‘estimate’ statements. Enclosed in the ‘glm’ procedure in Program 11.6 there are twelve ‘estimate’ statements one for each patient’s individual treatment effect estimate. Each ‘estimate’ statement has an appropriate label to permit the identification of the patient’s individual treatment effect estimate in the software’s output. For each patient’s ‘estimate’ statement, the coefficients of the “Treatment” term are defined as 1 under treatment “B” and as -1 under treatment “A”. The “Treatment” term’s coefficients are common to all the patients in the sample. The “Treatment” by “Patient” interaction term’s coefficients are the ones that differ between the patients. The first row of twelve elements after the “Patient*Treatment” term refers to treatment “B” while the second row refers to treatment “A”. The column defines the position of the patient’s coefficient. The first column refers to the first patient, the second column is relative to the second patient, and so on, until the twelfth column which is respective to the twelfth patient. On each ‘estimate’ statement after the “Patient*Treatment” term there is a 1 for treatment “B” and a -1 for treatment “A” on the position relative to the patient defined on that statement, and there is a 0 on the position relative to the other patients. This way the individual treatment effect is computed according to the definition previously given. Program 11.6 outputs seven tables. The first six tables are Table 10.14, Table 10.15, Table 10.16, Table 10.17, Table 10.18, and Table 10.19. The seventh table presented below, results from the ‘estimate’ statements in Program 11.6, and holds several statistics relative to the individual treatment effects.

Table 11.16: Individual treatment effects – Balanced data.

Parameter	Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
B-A,1	223.666667	88.8537083	2.52	0.0189	40.281626	407.051707
B-A,2	84.666667	88.8537083	0.95	0.3502	-98.718374	268.051707
B-A,3	60.000000	88.8537083	0.68	0.5060	-123.385041	243.385041
B-A,4	348.000000	88.8537083	3.92	0.0007	164.614959	531.385041
B-A,5	259.333333	88.8537083	2.92	0.0075	75.948293	442.718374
B-A,6	50.000000	88.8537083	0.56	0.5788	-133.385041	233.385041
B-A,7	175.000000	88.8537083	1.97	0.0605	-8.385041	358.385041
B-A,8	153.666667	88.8537083	1.73	0.0966	-29.718374	337.051707
B-A,9	324.333333	88.8537083	3.65	0.0013	140.948293	507.718374
B-A,10	247.666667	88.8537083	2.79	0.0102	64.281626	431.051707
B-A,11	214.333333	88.8537083	2.41	0.0239	30.948293	397.718374
B-A,12	124.000000	88.8537083	1.40	0.1756	-59.385041	307.385041

Table 11.16 presents the results in seven columns and twelve rows. The columns represent in this order the label, the estimate, the standard error of the estimate, the t statistic, the p-value, the lower confidence limit of the estimate, and the upper confidence limit of the estimate. The estimate in question is the estimate of the individual treatment effect. There are twelve estimates represented, one for each patient present in the sample. The t statistic and the p-value correspond to testing a null hypothesis of an estimate equal to zero against an alternative hypothesis of difference of the estimate from zero. Note that the estimates displayed on Table 11.16 are the same as the ones present on Table 7.1. In fact, the individual treatment effects resulting from a linear fixed-effects model fitting a treatment by patient interaction are equal to the naive individual treatment effects. This can be shown more rigorously using matrix algebra and linear fixed effects model theory. Both methods use only the data of the patient to estimate the individual treatment effect of that patient. This implies that the linear fixed effects model considered is as efficient as the naive method for estimating individual treatment effects. Also, note that the standard errors of the individual treatment effect estimates in Table 11.16 are all equal, while the standard errors of the naive individual treatment effect estimates on Table 7.1 differ between each other. The naive method estimates the standard error of the estimates separately for each patient leading to a distinct standard error for each patient. While the linear fixed-effects model assumes a residual error common to all the observations and patients. The standard error of the individual treatment effect estimates resulting from the linear fixed effects model considered differs

only when the dataset is unbalanced, as is shown below. Observe that by equally averaging the individual treatment effect estimates from Table 11.16, the overall treatment effect presented on Table 11.3 and Table 11.4 is obtained. As regards the interpretation of the results from Table 11.16, the p-values and confidence intervals suggest that the treatments are significantly different for some patients, while for other patients the treatments do not differ significantly.

The number of n-of-1 trials in as series can be much larger than the twelve present in the datasets used in the previous programs. The procedures developed to fit linear models in the SAS software require writing an 'estimate' statement for each desired individual treatment effect. Writing 'estimate' statements for all the n-of-1 trials in a large series requires a significant amount of work and time from the programmer. A set of macros was written to lessen the effort required to write code for the estimation of individual treatment effects arising from series of n-of-1 trials [4, 5]. Macro A.1, Macro A.2, and Macro A.3 are entirely presented in Appendix A of the present document. The macros can be applied in similar statistical models and problems such as the statistical analysis of multicentre trials. The inner details of these long and complex macros are not exposed here. Detailing the code used in the macros requires a very large textbook, and is not the purpose of this document. The average user does not need to know how to write macros, being more interested in learning how to use the macros. The reader interested in improving his macro writing skills and interested in deciphering the macros used in programs throughout this document can consult the works of Burlew [4] and Carpenter [5] many times referenced in the present document. The usage of the macros developed for the estimation of individual treatment effects is introduced in the next program.

Program 11.7: Estimating individual treatment effects using ‘proc glm’ – The macro way.

```
* Write global macro variable array ;
%estimate_full_global(
        array=vfull_unbalanced,
        subject=Patient,
        treatment=Treatment,
        data=NORMAL.ndata_unbalanced
)

title "Individual treatment effects";
title2 "Unbalanced data";
proc glm
                data=NORMAL.ndata_unbalanced
                alpha=0.05 /* level for CI */
                plots=none; * suppress plots ;
        class Treatment (ref=first) Patient Cycle;
        model Y = Treatment Patient Patient*Treatment Patient*Cycle /
                alpha=0.05 /* level for CI */
                clparm; * CI for parameter estimates ;
        %estimate_full(array=vfull_unbalanced, random=)
run;
quit;

* Delete global macro variable array ;
%estimate_full_free(array=vfull_unbalanced)
```

The ‘glm’ procedure is used in Program 11.7 to fit the same model previously used in Program 11.6 to the unbalanced dataset “NORMAL.ndata_unbalanced” created by either Program 4.4 or Program 5.5. Before running the ‘glm’ procedure, the %estimate_full_global macro is invoked. The %estimate_full_global macro writes a set of global macro variables. All the macro variables created by %estimate_full_global are prefixed by the word defined in the “array” parameter of this macro. When using macros with global macro variables there is the possibility that macro variables with the exact same names as the ones being in use in the main program are declared or reassigned inside the macro definition, leading to unexpected behaviour of the whole program. For this reason, it is considered a good programming practice to declare macro variables in local scope inside the actual macro definitions. Usage of global macro variables shall be avoided inside macro definitions and permitted only in the main program. The development of the %estimate_full_global macro avoids this problem. The final user of the macros, the programmer who is writing the main program can define the prefix of the set of global macros at his own free will. Moreover, by avoiding the definition of other macro variables with the same prefix, conflicts can be prevented. The %estimate_full_global macro variable accepts a “data” argument, which in this case is set as the “NORMAL.ndata_unbalanced” dataset. The “subject” parameter defines the categorical variable indicating the subject or patient in the series of n-of-1 trials. In this case, the “Patient” variable is supplied to the “subject” parameter. The “treatment” parameter of %estimate_full_global defines the categorical variable indicating the treatment in the series of n-of-1 trials, and is set to the “Treatment” variable from “NORMAL.ndata_unbalanced”. The variables supplied to the “subject” and “treatment” parameters must obviously be present in the dataset provided to the “data” parameter of the %estimate_full_global macro. The %estimate_full_global macro accepts a ’ref’ parameter that can be set as “first”, “last” or any level of the treatment category variable. The ‘ref’ parameter can be used to define the reference treatment and is set as “first” by default, where the first treatment level appearing in the dataset is used as the reference treatment. The same value as the one used for the ‘ref’ option in the ‘class’ statement is recommended here. Note that the ‘ref’ option after the “Treatment” variable in the ‘class’ statement is defined as “first”. After writing the set of global macro variables, which can also be considered an array of macro variables, the “vfull_unbalanced” macro variable array is available for posterior use by the %estimate_full macro. Briefly, what %estimate_full_global does is to look into the dataset provided, determine the labels for every patient and treatment pair combination, determine the vectors containing the 0, -1 and 1 in the right place, such as the vectors in the ‘estimate’ statements of Program 11.6, and save this information to a global macro variable array. The %estimate_full macro is placed where the ‘estimate’ statements of the ‘glm’ procedure are usually placed. When the program is submitted for interpretation and the %estimate_full macro is expanded, all the possible ‘estimate’ statements are determined from the “vfull_unbalanced” macro variable array, and are inserted under the ‘glm’ procedure. The “random” parameter of the %estimate_full macro is there for compatibility with the ‘mixed’ procedure. Usage of the ‘mixed’ procedure to fit linear mixed-effects models is demonstrated below on separate sections. For linear fixed-effects models, the “random” parameter must always be left undefined. The macro variables from the “vfull_unbalanced” array are available during the entire SAS session, unless they are deleted. The %estimate_full_free macro can be used to delete all the macro variables previously created by the %estimate_full_global macro. In the bottom of Program 11.7, the “vfull_unbalanced” macro variable array is supplied to the “array” parameter of the %estimate_full_free macro, and when the program is executed, the macro call is expanded with appropriate statements to perform the deletion of the variables prefixed by the “vfull_unbalanced” word. This way the global macro variables can be deleted when no longer needed, providing yet another strategy to avoid possible conflicts while running the main program. A single macro could have been developed to complete the tasks performed by the three different macros introduced in Program 11.7, without resorting to global macro variables. The solution provided in three distinct macro calls is preferred because it can be computationally more efficient under certain situations. The dataset used in Program 11.7 contains a treatment indicator variable with only two treatments; however, the macros exemplified in the same Program 11.7 were designed without considering any limit for the number of treatments. Therefore, the macros presented can in fact be applied to series of n-of-1 trials comparing more than two treatments. Program 11.7 is similar to Program 11.2 referencing the same dataset, using the same ‘glm’ procedure with the same ‘model’ statement, so the output of both programs is identical. Program 11.7 outputs Table 11.5, Table 11.6, Table 11.7, Table 11.8, Table 11.9, Table 11.10 and the table resulting from the ‘estimate’ statements inserted into the program by the %estimate_full macro. The additional table obtained from Program 11.7 holds the individual treatment effect statistics. The table is presented below.

Table 11.17: Individual treatment effects - Unbalanced data.

Parameter	Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
B-A,1	223.666667	91.091311	2.46	0.0229	34.231915	413.101419
B-A,2	84.666667	91.091311	0.93	0.3632	-104.768085	274.101419
B-A,3	60.000000	91.091311	0.66	0.5173	-129.434752	249.434752
B-A,4	348.000000	91.091311	3.82	0.0010	158.565248	537.434752
B-A,5	259.333333	91.091311	2.85	0.0097	69.898581	448.768085
B-A,6	50.000000	91.091311	0.55	0.5889	-139.434752	239.434752
B-A,7	175.000000	91.091311	1.92	0.0684	-14.434752	364.434752
B-A,8	153.666667	91.091311	1.69	0.1064	-35.768085	343.101419
B-A,9	324.333333	91.091311	3.56	0.0018	134.898581	513.768085
B-A,10	247.666667	91.091311	2.72	0.0129	58.231915	437.101419
B-A,11	254.500000	111.563616	2.28	0.0331	22.490759	486.509241
B-A,12	132.000000	157.774779	0.84	0.4122	-196.110615	460.110615

Observe that the individual treatment effect estimates presented in Table 11.17 are the same as the ones presented in Table 7.4. Above in this document, it is noted that the estimates of the individual treatment effects resulting from a linear fixed-effects model with a treatment by patient interaction are equal to the naive individual treatment effects. A linear fixed-effects model with a treatment by patient interaction is as efficient as the naive method for estimating individual treatment effects. Table 11.17 shows that unlike the standard error of the estimates of the individual treatment effects resulting from balanced designs presented in Table 11.16, for unbalanced designs the standard error of the individual treatment effect estimates are not equal between each other. In fact, the linear fixed effects model used to obtain Table 11.17 assumes a residual error term common to all observations, which implies that the standard error of the individual treatment effects differs according to the number of observations used to estimate a particular individual treatment effect. The individual treatment effect estimates for patients 1 through 10 were computed from three observations, and due to the common variance of the residual error assumed by the model, the estimates of the standard errors are equal for the patients in question. The individual treatment effect estimate for patient 11 depends on two observations, so the standard error of the estimate in question shows an increase. For patient 12 there is only one observation available to estimate the individual treatment effect, and the standard error of this estimate increases further in relation to the standard error of the estimate for the previous patients. Also, note that despite the availability of only one observation for patient 12, the linear fixed-effects model permits the estimation of the standard error of the individual treatment effect estimate, while for the naive method such estimation is not possible. It can be observed once more, that the t statistics and p-values from Table 11.17 suggest that the two treatments studied differ significantly for some patients, while for other patients the difference between the treatments is not significant.

12 Linear fixed-effects model of difference

The dataset of outcome differences “NORMAL.ddata” created by either Program 4.5 or Program 5.6 can be analysed through a distinct but equivalent linear fixed-effects regression model. The “dY” outcome variable in this dataset has a meaning that is different from the “Y” outcome variable in the original “NORMAL.ndata” dataset. The “Treatment” variable is absent from “NORMAL.ddata” so only the available “Patient” and “Cycle” variables can be fitted in the model. The linear fixed-effects model of difference used in the programs presented below in this section can be obtained from the one used in the programs presented in section 11. The outcome variable “dY” is defined as the difference between two observations of the variable “Y” registered under two treatments from the same cycle. According to this definition, the “Treatment” by “Patient” interaction in the linear fixed-effects model leads to a single “Patient” fixed effect in the linear fixed-effects model of difference. While the “Cycle” by “Patient” interaction in the former model is eliminated, and is absent from the latter model.

12.1 Estimation and inference on the overall treatment effect

The following program demonstrates fitting a linear fixed-effects model of difference to the “NORMAL.ddata” dataset. The estimation of the overall treatment effect is considered first.

Program 12.1: Fitting a linear fixed-effects model of difference using ‘proc glm’- Balanced data.

```
title "Balanced data";
proc glm
                data=NORMAL.ddata
                alpha=0.05 /* level for CI */
                plots=none; * suppress plots ;
        class Patient;
        model dY = Patient /
                alpha=0.05 /* level for CI */
                clparm; * CI for parameter estimates ;
        estimate 'Overall mean'
                intercept                       1;
run;
quit;
```

The ‘glm’ procedure is used in Program 12.1 to fit a linear fixed effects model on the “NORMAL.ddata” balanced dataset. The ‘model’ statement is a simple one with “dY” as response variable and “Patient” as the only explanatory variable. This model implicitly assumes an intercept, which is equal to the individual treatment effect of the reference patient. The ‘estimate’ statement requests the estimate of the intercept of the model, where the label “Overall mean” is defined. This label allows the identification of the requested estimate in the output. Seven distinct tables are obtained from Program 12.1.

Table 12.1: Class level information.

Class Level Information		
Class	**Levels**	**Values**
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12

Table 12.2: Number of observations.

Number of Observations Read	36
Number of Observations Used	36

Table 12.3: Model SS.

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	319032.5556	29002.9596	1.22	0.3241
Error	24	568438.6667	23684.9444		
Corrected Total	35	887471.2222			

Table 12.4: Model information.

R-Square	Coeff Var	Root MSE	dY Mean
0.359485	81.54797	153.8991	188.7222

Table 12.5: Type I SS.

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Patient	11	319032.5556	29002.9596	1.22	0.3241

Table 12.6: Type III SS.

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Patient	11	319032.5556	29002.9596	1.22	0.3241

Table 12.7: Overall treatment effect - Balanced data.

Parameter	**Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**95% Confidence Limits**	
Overall mean	188.722222	25.6498562	7.36	<.0001	135.783521	241.660924

The p-value of the “Patient” variable source of variation suggests that the term is non-significant at a 0.05 significance level. However, the “Patient” term must be included in the model so that distinct per patient individual treatment effects can be estimated. The results relative to the overall treatment effect presented in Table 12.7 coincide with the results obtained from the linear fixed-effects model presented in Table 11.4. The p-value relative to the overall treatment effect estimate is significantly inferior to the usual 0.05 significance level, and the confidence interval of the same overall treatment effect estimate does not contain the origin, suggesting that on average the treatments differ between each other.

In the following program, the fixed-effects model of difference is fitted to the unbalanced dataset “NORMAL.ddata_unbalanced” created by either Program 4.6 or Program 5.7. The results from the application of the fixed effects model of difference to unbalanced data are compared below against the results of the same model applied to balanced data.

Program 12.2: Fitting a linear fixed-effects model of difference using ‘proc glm’- Unbalanced data.

```
title "Unbalanced data";
proc glm
                data=NORMAL.ddata_unbalanced
                alpha=0.05 /* level for CI */
                plots=none; * suppress plots ;
        class Patient;
        model dY = Patient /
                alpha=0.05 /* level for CI */
                clparm; * CI for parameter estimates ;
        estimate 'Overall mean'
                intercept                       1;
run;
quit;
```

Program 12.2 is very similar to Program 12.1 so only the differences are mentioned. In Program 12.2, the dataset “NORMAL.ddata_unbalanced” is supplied to the ‘data’ option. The other statements and options are kept the same in relation to Program 12.1. To permit a distinction between the outputs of the programs, the ‘title’ statement is modified accordingly. After the interpretation of the code by the SAS software, the following seven tables are outputted.

Table 12.8: Class level information.

Class Level Information		
Class	**Levels**	**Values**
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12

Table 12.9: Number of observations.

Number of Observations Read	33
Number of Observations Used	33

Table 12.10: Model SS.

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	315249.6818	28659.0620	1.15	0.3743
Error	21	522750.5000	24892.8810		
Corrected Total	32	838000.1818			

Table 12.11: Model information.

R-Square	Coeff Var	Root MSE	dY Mean
0.376193	81.09919	157.7748	194.5455

Table 12.12: Type I sums of squares.

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Patient	11	315249.6818	28659.0620	1.15	0.3743

Table 12.13: Type III sums of squares.

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Patient	11	315249.6818	28659.0620	1.15	0.3743

Table 12.14: Overall treatment effect- Unbalanced data.

Parameter	Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
Overall mean	192.736111	28.9054483	6.67	<.0001	132.623941	252.848282

The unbalanced dataset was obtained by deleting some observations from the balanced dataset, so there is less information available in the unbalanced dataset. The standard error of the overall treatment effect estimated from the unbalanced dataset shows a small increase reflecting the lower amount of information available for the same dataset. Considering the usual 0.05 significance level, the p-value associated to the overall treatment effect estimate and displayed in Table 12.14 suggests that the two treatments trialled differ significantly from each other. Also, note that the confidence interval of the overall treatment effect estimate does not include the origin.

The ‘estimate’ statement in Program 12.2 produces an unbiased estimate of the overall treatment effect by equally weighting the means computed from the observations of each individual. An equally unbiased estimate with lower variance and therefore higher efficiency can be obtained by weighting the individual independent means according to their precision. For a model with only one error term such as the model in question, the precision-based weights only depend on the number of observations available for a particular individual. Such efficient estimate of the overall treatment effect can be obtained within the SAS software through the definition of an appropriate ‘estimate’ statement under the ‘glm’ procedure. The exact number of observations available under each patient needs to be determined first. The “NORMAL.ddata_unbalanced” dataset used in Program 12.2 is relatively small so the dataset can be printed and the observations can be counted very easily. For very large datasets printing and counting the observations is extremely difficult. Fortunately very simple programs can be written to let the computer handle tasks that are more difficult for the common human mind. The next small program counts the observations available for the unbalanced dataset.

Program 12.3: Determining the number of observations.

```
title "Unbalanced data";
title2 "Total number of observations";
proc sql;
        select   count(dY) as nobs
        from NORMAL.ddata_unbalanced;
quit;
```

In Program 12.3, the ‘sql’ procedure is used to query the “NORMAL.ddata_unbalanced” dataset. The ‘count’ function is used to determine the number of observations available for variable “dY” present on the same dataset. The number of observations resulting from the ‘count’ function is saved under the variable “nobs”. The table below results from Program 12.3.

Table 12.15: Number of observations - Unbalanced data.

nobs
33

Table 12.15 shows that 33 observations are available on the unbalanced dataset. This number determines the denominator of the weights used to compute the estimate of the overall treatment effect. The number of observations available under each patient determines the numerator of the referred weights. The following program counts the observations available under each patient on the unbalanced dataset.

Program 12.4: Determining the number of observations per patient.

```
title "Unbalanced data";
title2 "Number of observations per Patient";
proc sql;
        select          Patient,
                        count(dY) as nobs
        from NORMAL.ddata_unbalanced
        group by Patient;
quit;
```

As in Program 12.3, the ‘count’ function is used to determine the number of observations available for variable “dY” present on the dataset, and the number of observations resulting from the ‘count’ function is saved under the variable “nobs”. In Program 12.4, the results of the ‘select’ statement are grouped by the “Patient” variable. It is often useful to enumerate the grouping variables in the ‘select’ statement, so in this case the selection of the “Patient” variable from the dataset permits the association of the observation count to the respective patient. The table below is outputted from the submission of Program 12.4.

Table 12.16: Number of observations per patient - Unbalanced data.

Patient	nobs
1	3
2	3
3	3
4	3
5	3
6	3
7	3
8	3
9	3
10	3
11	2
12	1

Given the availability of the results displayed in Table 12.15 and Table 12.16, a new program can now be written with the purpose of computing a more efficient estimate of the overall treatment effect. The next program makes use of these results.

Program 12.5: Fitting a linear fixed-effects model of difference using 'proc glm'- Weighted estimate.

```
title "Unbalanced data";
title2 "Weighted estimate";
proc glm
                data=NORMAL.ddata_unbalanced
                alpha=0.05 /* level for CI */
                plots=none; * suppress plots ;
        class Patient;
        model dY = Patient /
                alpha=0.05 /* level for CI */
                clparm /* CI for parameter estimates */
                ss1
                ss2
                ss3;
        estimate 'Overall mean'
                intercept                   33
                Patient                     3 3 3 3 3 3 3 3 3 3 2 1 /
                divisor=33;
run;
quit;
```

Program 12.5 is similar to Program 12.2. Three additional options are added to the 'model' statement. The 'ss1', 'ss2' and 'ss3' options request the type I, type II and type III sums of squares tables respectively. The 'estimate' statement in Program 12.5 is more elaborate, specifying weights to the "Patient" term in addition to the 'intercept' term. Immediately after the 'estimate' keyword, a label is specified. The label permits the identification of the desired estimate in the SAS software's output. Here the observation numbers previously determined by Program 12.3 and Program 12.4 and presented in Table 12.15 and Table 12.16 are used to define the weights given to each fixed effect statistic with the purpose of computing the overall treatment effect estimate. The total number of observations present in the sample, 33 in this case, is specified as a weight to the 'intercept' term. As regards the "Patient" term, the observation numbers previously obtained from Program 12.4 and displayed in Table 12.16 are placed in a position respective to each patient. The column defines the position of the patient's coefficient. The first column refers to the first patient, the second column is relative to the second patient, and so on, until the twelfth column which is respective to the twelfth patient. In fact, the weights must lie between zero and unity, but the 'estimate' statement does not support the specification of non-integer numbers as weights. The 'estimate' statement supports the specification of the 'divisor' option to define a denominator common to all the weights in the statement. So all the observation numbers relative to each term specified in the 'estimate' statement are divided by 33, the total number of observations. Therefore, the weight relative to the 'intercept' term is in fact 1, and the weights attached to the "Patient" term sum to 1. Program 12.5 produces Table 12.8 Table 12.9 Table 12.10 Table 12.11 Table 12.12 and Table 12.13 exactly as obtained from Program 12.2. A new table is obtained for the type II sums of squares, and a table similar to Table 12.14 is obtained. The obtained tables are presented below.

Table 12.17: Type II sums of squares.

Source	DF	Type II SS	Mean Square	F Value	Pr > F
Patient	11	315249.6818	28659.0620	1.15	0.3743

Table 12.18: Overall treatment effect - Weighted estimate.

Parameter	Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
Overall mean	194.545455	27.4650637	7.08	<.0001	137.428728	251.662181

The same results are contained in Table 12.12, Table 12.13 and Table 12.17. The model contains only one variable so the type I, type II and type III sums of squares cannot be different between each other. Also, note that the same values lie in the 'Model' row of Table 12.10. The weighted estimate presented in Table 12.18 is distinct from the unweighted one displayed in Table 12.14. As expected, the standard error of the weighted estimate is lower than the standard error of the unweighted one. In addition, the confidence interval of the weighted estimate is narrower than the confidence interval of the unweighted estimate. The weighted estimate displayed in Table 12.18 is more efficient than the unweighted one from Table 12.14 given the assumptions inherent to the model. For this reason, the weighted estimator is recommended over the unweighted one. The p value is still much lower than the conventional 0.05 significance level, so the conclusions previously drawn from Table 12.14 remain valid.

12.2 Estimation and inference on individual treatment effects

Individual treatment effects can be estimated from data arising from series of n-of-1 trials. So that a treatment preference can be determined for each patient, based on its own treatment effect rather than on the overall treatment effect, depending on the information available for that particular patient. The linear fixed-effects model of difference in question models the response variable as the difference of a distinct outcome variable taken between two treatments. So for this model the individual treatment effect can be found by taking the expectation of the response variable conditioned on a given patient. Given this definition, for a linear model such as the one used in Program 12.1, Program 12.2 and Program 12.5, the individual treatment effects depend only on the 'intercept' term and on the "Patient" fixed term. The "Patient" term is distinct for each patient and is the reason for the distinctiveness of the individual treatment effects. A fixed-effects model of difference with an 'intercept' term and without a "Patient" term results in the same individual treatment effect for each and every patient in the sample. In such a model, the individual treatment effect coincides with the overall treatment effect. When estimating individual treatment effects with the purpose of individualizing medicine from a linear model applied to a dataset where the outcome variable is already coded as a difference of an outcome variable between treatments, the inclusion in the model of a variable indicating the patient is therefore recommended. The following program can be considered an extension of Program 12.1, where the estimation of individual treatment effects for each patient present in the dataset is achieved. The code is based on code published by Brown and Prescott [21] for similar linear mixed-effects models applied to the analysis of multicentre trials. Linear mixed-effects models applied to the analysis of series of n-of-1 trials are presented on a separate section of the present document. The balanced data "NORMAL.ddata" created before by either Program 4.5 or Program 5.6 is used in the program below.

Program 12.6: Estimating individual treatment effects using 'proc glm' - The hard way.

```
title "Individual treatment effects";
title2 "Balanced data";
proc glm
                data=NORMAL.ddata
                alpha=0.05 /* level for CI */
                plots=none; * suppress plots ;
        class Patient;
        model dY = Patient /
                alpha=0.05 /* level for CI */
                clparm; * CI for parameter estimates ;
        estimate '1' intercept 1
                Patient  1;
        estimate '2' intercept 1
                Patient 0 1;
        estimate '3' intercept 1
                Patient 0 0 1;
        estimate '4' intercept 1
                Patient 0 0 0 1;
        estimate '5' intercept 1
                Patient 0 0 0 0 1;
        estimate '6' intercept 1
                Patient 0 0 0 0 0 1;
        estimate '7' intercept 1
                Patient 0 0 0 0 0 0 1;
        estimate '8' intercept 1
                Patient 0 0 0 0 0 0 0 1;
        estimate '9' intercept 1
                Patient 0 0 0 0 0 0 0 0 1;
        estimate '10' intercept 1
                Patient 0 0 0 0 0 0 0 0 0 1;
        estimate '11' intercept 1
                Patient 0 0 0 0 0 0 0 0 0 0 1;
        estimate '12' intercept 1
                Patient 0 0 0 0 0 0 0 0 0 0 0 1;
run;
quit;
```

To identify the output Program 12.6 starts with two ‘title’ statements. Enclosed in the ‘glm’ procedure in Program 12.6 there are twelve ‘estimate’ statements for each patient’s individual treatment effect estimate. Each ‘estimate’ statement has an appropriate label to permit the identification of the patient’s individual treatment effect estimate in the software’s output. The ‘intercept’ term’s coefficients are common to all the patients in the sample. So each one of the twelve ‘estimate’ statements is given a coefficient of 1 to the ‘intercept’ term. The “Patient” term’s coefficients differ between the patients. The column defines the position of the patient’s coefficient. The first column refers to the first patient, the second column is relative to the second patient, and so on, until the twelfth column which is respective to the twelfth patient. On each ‘estimate’ statement after the “Patient” term, there is a 1 on the position relative to the patient defined on that statement, and there is a 0 on the position relative to the other patients. This way the individual treatment effect is computed according to the definition previously given. Program 12.6 outputs seven tables. The first six tables are Table 12.1, Table 12.2, Table 12.3, Table 12.4, Table 12.5 and Table 12.6. The seventh table presented below, results from the ‘estimate’ statements in Program 12.6 and holds several statistics relative to the individual treatment effects.

Table 12.19: Individual treatment effects - Balanced data.

Parameter	Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
1	223.666667	88.8537083	2.52	0.0189	40.281626	407.051707
2	84.666667	88.8537083	0.95	0.3502	-98.718374	268.051707
3	60.000000	88.8537083	0.68	0.5060	-123.385041	243.385041
4	348.000000	88.8537083	3.92	0.0007	164.614959	531.385041
5	259.333333	88.8537083	2.92	0.0075	75.948293	442.718374
6	50.000000	88.8537083	0.56	0.5788	-133.385041	233.385041
7	175.000000	88.8537083	1.97	0.0605	-8.385041	358.385041
8	153.666667	88.8537083	1.73	0.0966	-29.718374	337.051707
9	324.333333	88.8537083	3.65	0.0013	140.948293	507.718374
10	247.666667	88.8537083	2.79	0.0102	64.281626	431.051707
11	214.333333	88.8537083	2.41	0.0239	30.948293	397.718374
12	124.000000	88.8537083	1.40	0.1756	-59.385041	307.385041

Table 12.19 presents the results in seven columns and twelve rows. The columns represent in this order the label, the estimate, the standard error of the estimate, the t statistic, the p-value, the lower confidence limit of the estimate, and the upper confidence limit of the estimate. The estimate in question is the estimate of the individual treatment effect. There are twelve estimates represented, one for each patient present in the sample. The t statistic and the p-value correspond to testing a null hypothesis of an estimate equal to zero against an alternative hypothesis of difference of the estimate from zero. Observe that the results present on Table 12.19 are the same as the ones previously presented on Table 11.16. The results suggest that the linear fixed-effects model of difference is equivalent to the linear fixed-effects model as regards the estimation of individual treatment effects. A more rigorous demonstration of the equivalence between the two models in question can be obtained through the application of linear algebra and linear fixed-effects model theories. Note that the estimates displayed on Table 12.19 are the same as the ones present on Table

7.1. In fact, the individual treatment effects resulting from a linear fixed-effects model of difference fitting a patient indicator variable are equal to the naive individual treatment effects. Both methods use only the data of the patient to estimate the individual treatment effect of that patient. This implies that the linear fixed effects model of difference considered is as efficient as the naive method for estimating individual treatment effects. Also, note that the standard errors of the individual treatment effect estimates in Table 12.19 are all equal, while the standard errors of the naive individual treatment effect estimates on Table 7.1 differ between each other. The naive method estimates the standard error of the estimates separately for each patient leading to a distinctly distributed residual error for each patient. While the linear fixed-effects model assumes a residual error that is identically distributed across all the observations and patients. The standard error of the individual treatment effect estimates resulting from the linear fixed effects model of difference considered differs only when the dataset is unbalanced, as is shown below. Observe that by equally averaging the individual treatment effect estimates from Table 12.19, the overall treatment effect presented on Table 12.7 is obtained. As regards the interpretation of the results from Table 12.19, the p-values and confidence intervals suggest that the treatments are significantly different for some patients, while for other patients the treatments do not differ significantly.

The number of n-of-1 trials in as series can be much larger than the twelve present in the datasets used in the previous programs. The procedures developed to fit linear models in the SAS software require writing an ‘estimate’ statement for each desired individual treatment effect. Writing ‘estimate’ statements for all the n-of-1 trials in a large series requires a significant amount of work and time from the programmer. A set of macros was developed to lessen the effort required to write code for the estimation of individual treatment effects arising from series of n-of-1 trials [4, 5]. The inner details of these long and complex macros are not exposed here. Detailing the code used in the macros would require a very large textbook. Such extensive exposition is beyond the purpose of this document. The average user does not need to know how to write macros, being more interested in learning how to use the macros. The reader interested in improving his macro writing skills and interested in deciphering the macros used in programs throughout this document can consult the works of Burlew [4] and Carpenter [5] many times referenced in the present document. The ‘estimate’ statements used for the linear fixed-effect model of difference are different from the ones used previously for the linear fixed-effects model, so the macros presented in section 11 cannot be applied, and new sets of macros had to be developed. The reader can observe later that the same macros used to compute the individual treatment effects from linear fixed-effects models, can be applied to compute the same estimates from linear mixed-effects models. The macros developed for the estimation of individual treatment effects from the linear fixed-effects model of difference are introduced in the next program. The macros in question are Macro A.7, Macro A.8, and Macro A.9, wholly presented in Appendix A of this document.

Program 12.7: Estimating individual treatment effects using ‘proc glm’ - The macro way.

```
* Write global macro variable array ;
%estimate_diff_global(
        array=vdiff_unbalanced,
        subject=Patient,
        data=NORMAL.ddata_unbalanced
)

title "Individual treatment effects";
title2 "Unbalanced data";
proc glm
                data=NORMAL.ddata_unbalanced
                alpha=0.05 /* level for CI */
                plots=none; * suppress plots ;
        class Patient;
        model dY = Patient /
                alpha=0.05 /* level for CI */
                clparm; * CI for parameter estimates ;
        %estimate_diff(array=vdiff_unbalanced, random=)
run;
quit;

* Delete global macro variable array ;
%estimate_diff_free(array=vdiff_unbalanced)
```

The ‘glm’ procedure is used in Program 12.7 to fit the same model previously used in Program 12.6 to the unbalanced dataset “NORMAL.ddata_unbalanced” created by either Program 4.6 or Program 5.7. Before running the ‘glm’ procedure, the %estimate_diff_global macro is invoked. The %estimate_diff_global macro writes a set of global macro variables. All the macro variables created by %estimate_diff_global are prefixed by the word defined in the “array” parameter of this macro. When using macros with global macro variables there is the possibility that macro variables with the exact same names as the ones being in use in the main program are declared or reassigned inside the macro definition, leading to unexpected behaviour of the whole program. For this reason, it is considered a good programming practice to declare macro variables in local scope inside the actual macro definitions. Usage of global macro variables shall be avoided inside macro definitions and permitted only in the main program. The development of the %estimate_diff_global macro avoids this problem. The final user of the macros, the programmer who is writing the main program can define the prefix of the set of global macros at his own free will. By avoiding the definition of other macro variables with the same prefix, conflicts can be prevented. The %estimate_diff_global macro variable accepts a “data” argument, which in this case is set as the “NORMAL.ddata_unbalanced” dataset. The “subject” parameter defines the categorical variable indicating the subject or patient in the series of n-of-1 trials. In this case, the “Patient” variable is supplied to the “subject” parameter. The variable supplied to the “subject” parameter must obviously be present in the dataset provided to the “data” parameter of the %estimate_diff_global macro. After writing the set of global macro variables, which can also be considered an array of macro variables, the “vdiff_unbalanced” macro variable array is available for posterior use by the %estimate_diff macro. Briefly, what %estimate_diff_global does is to look into the dataset provided, determine the labels for every patient, determine the vectors containing the 0 and 1 in the right place, such as the vectors in the ‘estimate’ statements of Program 12.6, and save this information to a global macro variable array. The %estimate_diff macro is placed where the ‘estimate’ statements of the ‘glm’ procedure are usually placed. When the program is submitted for interpretation and the %estimate_diff macro is expanded, all the possible ‘estimate’ statements are determined from the “vdiff_unbalanced” macro variable array, and are inserted under the ‘glm’ procedure. The “random” parameter of the %estimate_diff macro is there for compatibility with the ‘mixed’ procedure. Usage of the ‘mixed’ procedure to fit linear mixed-effects models is demonstrated below on a separate subsection. For linear fixed-effects models, the “random” parameter must always be left undefined. The macro variables from the “vdiff_unbalanced” array are available during the entire SAS session, unless they are deleted. The %estimate_diff_free macro can be used to delete all the macro variables previously created by the %estimate_diff_global macro. In the bottom of Program 12.7, the “vdiff_unbalanced” macro variable array is supplied to the “array” parameter of the %estimate_diff_free macro, and when the program is executed, the macro call is expanded with appropriate statements to perform the deletion of the variables prefixed by the “vdiff_unbalanced” word. This way the global macro variables can be deleted when no longer needed, providing yet another strategy to avoid possible conflicts while running the main program. A single macro could have been developed to complete the tasks performed by the three different macros introduced in Program 12.7, without resorting to global macro variables. The solution provided in three distinct macro calls is preferred because it can be computationally more efficient under certain situations. Program 12.7 is similar to Program 12.2 referencing the same dataset, using the same ‘glm’ procedure with the same ‘model’ statement, so the output of both programs is identical. Program 12.7 outputs Table 12.8, Table 12.9, Table 12.10, Table 12.11, Table 12.12, Table 12.13 and the table resulting from the ‘estimate’ statements inserted into the program by the %estimate_diff macro. Below the additional table obtained from Program 12.7 is presented. The table holds the individual treatment effect statistics.

Table 12.20: Individual treatment effects - Unbalanced data.

Parameter	Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
1	223.666667	91.091311	2.46	0.0229	34.231915	413.101419
2	84.666667	91.091311	0.93	0.3632	-104.768085	274.101419
3	60.000000	91.091311	0.66	0.5173	-129.434752	249.434752
4	348.000000	91.091311	3.82	0.0010	158.565248	537.434752
5	259.333333	91.091311	2.85	0.0097	69.898581	448.768085
6	50.000000	91.091311	0.55	0.5889	-139.434752	239.434752
7	175.000000	91.091311	1.92	0.0684	-14.434752	364.434752
8	153.666667	91.091311	1.69	0.1064	-35.768085	343.101419
9	324.333333	91.091311	3.56	0.0018	134.898581	513.768085
10	247.666667	91.091311	2.72	0.0129	58.231915	437.101419
11	254.500000	111.563616	2.28	0.0331	22.490759	486.509241
12	132.000000	157.774779	0.84	0.4122	-196.110615	460.110615

Observe that the individual treatment effect estimates presented in Table 12.20 are the same as the ones presented in Table 7.4. A linear fixed-effects model of difference with a patient fixed effect is as efficient as the naive method for estimating individual treatment effects. Table 12.20 shows that unlike the standard error of the estimates of the individual treatment effects resulting from balanced designs presented in Table 12.19, for unbalanced designs the standard error of the individual treatment effect estimates are not equal between each other. In fact, the linear fixed effects model of difference used to obtain Table 12.20 assumes a residual error term common to all observations, which implies that the standard error of the individual treatment effects differs according to the number of observations used to estimate a particular individual treatment effect. The individual treatment effect estimates for patients 1 through 10 were computed from three observations, and due to the common variance of the residual error assumed by the model, the estimates of the standard errors are equal for the patients in question. The individual treatment effect estimate for patient 11 depends on two observations, so the standard error of the estimate in question shows an increase. For patient 12 there is only one observation available to estimate the individual treatment effect, and the standard error of this estimate increases further in relation to the standard error of the estimate for the previous patients. Also, note that despite the availability of only one observation for patient 12, the linear fixed-effects model permits the estimation of the standard error of the individual treatment effect estimate, while for the naive method such estimation is not possible. It can be observed once more that the t statistics and p-values from Table 12.20 suggest that the two treatments studied differ significantly for some patients, while for other patients the difference between the treatments is not significant.

13 Linear mixed-effects model

In this section, attention is devoted to the estimation of overall and individual treatment effects through linear mixed-effects models. Example programs are demonstrated and the results compared with the ones obtained from the methods exemplified above. Linear mixed-effects models contain both fixed and random terms. The fixed terms are assumed invariable, contributing only to expectation. While the random terms vary according to a distributional assumption and contribute to expectation as well as to the correlation structure of the model. The normal distribution is assumed for all the random terms of the models used in the example programs of this section.

13.1 Estimation and inference on the overall treatment effect

For starters, attention is given to the estimation of an overall treatment effect, i.e. the difference between the means of an outcome variable estimated under two distinct treatments. An overall treatment effect can be estimated through a linear mixed-effects model by fitting the variable indicating the treatment as fixed. In the following program a model with a “Treatment” fixed term, a random “Patient” term, a random “Treatment” by “Patient” interaction and a random “Cycle” by “Patient” interaction is fitted to the “NORMAL.ndata” balanced dataset previously obtained from either Program 4.1 or Program 5.4.

Program 13.1: Fitting a linear mixed-effects model using ‘proc mixed’ – Balanced data.

```
title "Balanced data";
proc mixed
                data=NORMAL.ndata
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Treatment (ref=first) Patient Cycle;
        model Y = Treatment /
                alpha=0.05 /* level for CI */
                cl /* t-type CI for fixed effects */
                covb /* covariance matrix of fixed effects estimates */
                ddfm=KR /* inference based on Kenward and Roger's method */
                htype=2 /* type 2 hypothesis test */
                solution; * solution for fixed effects ;
        random Patient Patient*Cycle Patient*Treatment;
        lsmeans Treatment /
                alpha=0.05 /* level for CI */
                cl /* t-type CI for least squares means */
                diff; * difference between each pair of LS means ;
run;
```

To facilitate the identification of the output, Program 13.1 starts with the definition of a custom title. The ‘mixed’ procedure is the only one being used in this program. The ‘mixed’ procedure supports a greater number of statements and options than the ones being used in programs presented in this document. The statements and options used here are the most pertinent given the questions arising from series of n-of-1 trials. The ‘mixed’ procedure statements in Program 13.1 contain a high number of options followed by comments indicating the purpose of each option, so only the options considered more relevant are mentioned in this text. Additional statements as well as every option available for each statement are very well documented and openly available online for anyone to consult [10]. The ‘data’ option of the ‘mixed’ procedure defines the dataset for analysis, which in this case is the “NORMAL.ndata” balanced dataset previously obtained from either Program 4.1 or Program 5.4. The ‘method’ option defines the residual maximum likelihood method for estimation of the parameters of the model. Residual maximum likelihood provides unbiased estimates of the variance parameters of the model, being recommended over the usual maximum likelihood method [21-24]. The ‘class’ statement enumerates all the categorical variables being fitted in the model. By writing ‘(ref=first)’ after the “Treatment” variable in the ‘class’ statement, the first level of the “Treatment” categorical variable appearing in the dataset is used as the reference treatment. The ‘model’ statement defines the fixed part of the linear mixed-effects model, in this case the outcome variable “Y” is written on the left side of the equal sign (=) and the explanatory variable “Treatment” is placed on the right side. The ‘ddfm=KR’ option requests the Kenward and Roger’s method for the computation of the covariance matrix of the fixed effects coefficients as well as the degrees of freedom [25]. A simple algebraic expression for the covariance matrix of the fixed effects based on asymptotic assumptions is available but can be largely biased for unbalanced datasets with a low number of observations [21]. The Kenward and Roger's method attempts to correct this biasedness, being therefore recommended. The 'random' statement can be used to define all random terms of the linear mixed-effects model. In this case, there is a "Patient" random term, a "Cycle" by "Patient" random interaction and a "Treatment" by "Patient" random interaction. Finally the 'lsmeans' statement requests the estimates of the means under each level of the "Treatment" categorical variable. The 'diff' option of the 'lsmeans' statement is particular important because it requests the estimate of the difference between each treatment mean, which is in fact the much desired estimate of the overall treatment effect. After running Program 13.1, the twelve tables presented below are obtained.

Table 13.1: Model information.

Model Information	
Data Set	NORMAL.NDATA
Dependent Variable	Y
Covariance Structure	Variance Components
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Kenward-Roger
Degrees of Freedom Method	Kenward-Roger

Table 13.2: Class level information.

Class Level Information		
Class	**Levels**	**Values**
Treatment	2	B A
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12
Cycle	3	1 2 3

Table 13.3: Dimensions.

Dimensions	
Covariance Parameters	4
Columns in X	3
Columns in Z	72
Subjects	1
Max Obs per Subject	72

Table 13.4: Number of observations.

Number of Observations	
Number of Observations Read	72
Number of Observations Used	72
Number of Observations Not Used	0

Table 13.5: Iteration history.

Iteration History			
Iteration	**Evaluations**	**-2 Res Log Like**	**Criterion**
0	1	931.31645659	
1	1	893.79244632	0.00000000

Table 13.6: Covariance parameter estimates.

Covariance Parameter Estimates				
Cov Parm	**Estimate**	**Alpha**	**Lower**	**Upper**
Patient	19459	0.05	8999.93	68835
Patient*Cycle	680.53	0.05	63.4043	3.797E24
Treatment*Patient	886.34	0.05	102.22	4.11E13
Residual	11842	0.05	7220.27	22919

Table 13.7: Fit statistics.

Fit Statistics	
-2 Res Log Likelihood	893.8
AIC (Smaller is Better)	901.8
AICC (Smaller is Better)	902.4
BIC (Smaller is Better)	903.7

Table 13.8: Solution for fixed effects.

Solution for Fixed Effects									
Effect	**Treatment**	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Intercept		2625.75	45.2031	13.4	58.09	<.0001	0.05	2528.37	2723.13
Treatment	**B**	188.72	28.3838	11	6.65	<.0001	0.05	126.25	251.19
Treatment	**A**	0	.	.	.	.	.	.	.

Table 13.9: Covariance matrix for fixed effects.

Covariance Matrix for Fixed Effects					
Row	**Effect**	**Treatment**	**Col1**	**Col2**	**Col3**
1	Intercept		2043.32	-402.82	
2	Treatment	B	-402.82	805.64	
3	Treatment	A			

Table 13.10: Type 2 tests for fixed effects.

Type 2 Tests of Fixed Effects				
Effect	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Treatment	1	11	44.21	<.0001

Table 13.11: Least squares means.

Least Squares Means									
Effect	**Treatment**	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Treatment	**B**	2814.47	45.2031	13.4	62.26	<.0001	0.05	2717.10	2911.85
Treatment	**A**	2625.75	45.2031	13.4	58.09	<.0001	0.05	2528.37	2723.13

Table 13.12: Differences of least squares means.

Differences of Least Squares Means										
Effect	**Treatment**	**_Treatment**	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Treatment	**B**	**A**	188.72	28.3838	11	6.65	<.0001	0.05	126.25	251.19

The 'mixed' procedure can produce a large number of tables depending on the statements and the options used. The information contained in some of the tables obtained from Program 13.1 is self-explanatory, while there are other tables that require some knowledge of linear mixed-effects model theory. Attention is given to the more pertinent results given the questions arising form series of n-of-1 trials. Table 13.6 presents the estimates of the variance parameters of the model and their respective confidence intervals. Each random term of the model is identified in the first column of Table 13.6. Note that none of the four confidence intervals in Table 13.6 contain zero, suggesting that all the four random terms in the model contribute significantly to explain the variation in the outcome variable. The estimates of the fixed effects, their standard errors and confidence intervals are displayed in Table 13.8. This table is particularly useful because the coefficient of the "Treatment" fixed effect is equal to the overall treatment effect, which is located in the second row of the table in question. The p-value and the confidence interval relative to the overall treatment effect suggest that the treatments differ significantly from each other. Table 13.11 and Table 13.12 are a result of the 'lsmeans' statement in Program 13.1. The estimate of the mean outcome under a particular treatment can be biased, and is not particular interesting in clinical trials, so Table 13.11 can be disregarded. The bias of the estimate of the difference between the means obtained under the two treatments can be minimized in a properly designed clinical trial. The estimate of this difference is referred shortly as overall treatment effect and is displayed in Table 13.12. Note that the exact same results are present in Table 13.12 and in the second row of Table 13.8. The p-value from *Table* 13.10 confirms the suggestion that the treatments differ significantly from each other. Also, note that the F statistic displayed in *Table* 13.10 is equal to the square of the t statistic form *Table 13.12*. The fit statistics from *Table* 13.7 can be used to compare nested models, being of less importance for the current problem. Interesting observations can be made by comparing the results from the linear mixed-effects model with the ones from the other methods so far presented. So the overall treatment effect results obtained from the various methods when applied to the same or equivalent balanced dataset are compared below. Observe that the exact same overall treatment effect estimate of 188.72 units is obtained from all the methods demonstrated. The estimate in question is presented in Table 8.1 and Table 8.2 for the t-test; in Table 9.1 and Table 9.2 for the summary measures approach; in Table 11.3 and Table 11.4 for the linear fixed-effects model; and in Table 12.7 for the linear fixed-effects model of difference. However, different results are obtained in relation to the standard error of this estimate. It is noted in section 12 that the same results in respect to the statistics associated to the overall treatment effect are obtained from both linear fixed-effects models, regardless of the balance in the data. It can be shown both analytically and by means of simulations, that when the data is balanced, for estimating the overall treatment effect, its standard error, t statistic and p-value; the linear mixed-effects model fitted in Program 13.1 is equivalent to the summary measures approach used in Program 9.1 [20, 26]. Note that the results obtained from the linear mixed-effects model in Table 13.12 are the same as the ones obtained from the summary measures approach in Table 9.1, Table 9.2, and Table 9.3. The lowest standard error of the overall treatment effect is obtained for the linear fixed-effects models displayed in Table 11.4 and Table 12.7. The standard error of the overall treatment effect estimate is higher for the linear mixed-effects model displayed in Table 13.12, and for the summary measures approach presented in Table 9.1. With the standard error of the overall treatment effect resulting from the t-test presented in Table 8.1, lying in between the one resulting from the other methods mentioned. These results seem to suggest that the linear fixed-effects models are the most efficient for estimating the overall treatment effect. However, it can be shown that when the data is balanced and there are additional significant sources of variation; all the methods except the linear mixed-effects model fitted in Program 13.1 and the summary measures approach exemplified in Program 9.1, result in estimates of the standard error of the overall treatment effect estimate that are downwards biased [20]. Observe that the confidence intervals of each of the variances displayed in Table 13.6 do not contain the origin, suggesting that all the random terms fitted are significant at the chosen significance level. In fact, several sources of variation significantly contribute to explain the variation of the outcome variable. Therefore, the standard error and confidence interval of the overall treatment effect resulting from the linear mixed-effects model and summary measures approach respectively demonstrated in Program 13.1 and Program 9.1 are more reliable and should be preferred.

The next program fits the same model used in Program 13.1 to the unbalanced dataset “NORMAL.ndata_unbalanced” created by either Program 4.4 or Program 5.5. The results are opportunely compared to the ones obtained in the sections above.

Program 13.2: Fitting a linear mixed-effects model using ‘proc mixed’ – Unbalanced data.

```
title "Unbalanced data";
proc mixed
                data=NORMAL.ndata_unbalanced
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Treatment (ref=first) Patient Cycle;
        model Y = Treatment /
                alpha=0.05 /* level for CI */
                cl /* t-type CI for fixed effects */
                covb /* covariance matrix of fixed effects estimates */
                ddfm=KR /* inference based on Kenward and Roger's method */
                htype=2 /* type 2 hypothesis test */
                solution; * solution for fixed effects ;
        random Patient Patient*Cycle Patient*Treatment;
        lsmeans Treatment /
                alpha=0.05 /* level for CI */
                cl /* t-type CI for least squares means */
                diff; * difference between each pair of LS means ;
run;
```

Program 13.2 is very similar to Program 13.1 using the same statements and options, the only difference being the datasets used in each program. The reader starting at this point of the document can consult the text following Program 13.1 for further information regarding the purpose of each statement and option used in Program 13.2. For information pertaining to the options not referenced in the text, the SAS software documentation can be consulted [10]. Program 13.2 outputs the twelve tables presented below.

Table 13.13: Model information.

Model Information	
Data Set	NORMAL.NDATA_UNBALANCED
Dependent Variable	Y
Covariance Structure	Variance Components
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Kenward-Roger
Degrees of Freedom Method	Kenward-Roger

Table 13.14: Class level information.

Class Level Information		
Class	**Levels**	**Values**
Treatment	2	B A
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12
Cycle	3	1 2 3

Table 13.15: Dimensions.

Dimensions	
Covariance Parameters	4
Columns in X	3
Columns in Z	69
Subjects	1
Max Obs per Subject	66

Table 13.16: Number of observations.

Number of Observations	
Number of Observations Read	66
Number of Observations Used	66
Number of Observations Not Used	0

Table 13.17: Iteration history.

Iteration History			
Iteration	**Evaluations**	**-2 Res Log Like**	**Criterion**
0	1	856.54072606	
1	3	822.17794569	0.00017945
2	1	822.10899264	0.00000459
3	1	822.10734030	0.00000001

Table 13.18: Covariance parameter estimates.

Covariance Parameter Estimates				
Cov Parm	**Estimate**	**Alpha**	**Lower**	**Upper**
Patient	19398	0.05	8864.10	70891
Patient*Cycle	1366.47	0.05	189.18	7.3298E9
Treatment*Patient	1099.30	0.05	142.07	9.752E10
Residual	12052	0.05	7216.84	24095

Table 13.19: Fit statistics.

Fit Statistics	
-2 Res Log Likelihood	822.1
AIC (Smaller is Better)	830.1
AICC (Smaller is Better)	830.8
BIC (Smaller is Better)	832.0

Table 13.20: Solution for fixed effects.

Solution for Fixed Effects									
Effect	**Treatment**	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Intercept		2621.25	46.2525	14	56.67	<.0001	0.05	2522.07	2720.42
Treatment	**B**	194.51	30.4936	11.3	6.38	<.0001	0.05	127.58	261.44
Treatment	**A**	0	.	.	.	.	.	.	.

Table 13.21: Covariance matrix for fixed effects.

Covariance Matrix for Fixed Effects					
Row	**Effect**	**Treatment**	**Col1**	**Col2**	**Col3**
1	Intercept		2139.29	-464.93	
2	Treatment	B	-464.93	929.86	
3	Treatment	A			

Table 13.22: Type 2 tests for fixed effects.

Type 2 Tests of Fixed Effects				
Effect	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Treatment	1	11.3	40.69	<.0001

Table 13.23: Least squares means.

Least Squares Means									
Effect	**Treatment**	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Treatment	**B**	2815.76	46.2525	14	60.88	<.0001	0.05	2716.58	2914.93
Treatment	**A**	2621.25	46.2525	14	56.67	<.0001	0.05	2522.07	2720.42

Table 13.24: Differences of least squares means.

Differences of Least Squares Means										
Effect	**Treatment**	**_Treatment**	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Treatment	**B**	**A**	194.51	30.4936	11.3	6.38	<.0001	0.05	127.58	261.44

There is a high number of tables in the output. The tables considered more relevant for the questions at hand are subject to a more elaborate analysis. The results displayed in the second row of *Table* 13.20 coincide with the results present on *Table* 13.24, where the statistics pertaining to the overall treatment effect are located. The p-value is much lower than the considered significance level, and the confidence interval of the estimate does not contain zero, suggesting that the treatments differ significantly between each other. The standard error of the overall treatment effect estimate resulting from the unbalanced data in *Table* 13.20 and *Table* 13.24 is higher than the standard error of the said estimate for balanced data displayed in Table 13.8 and Table 13.12, reflecting the deletion of observations while creating the unbalanced dataset from the balanced one. It is mentioned above in the current section that for balanced data, the linear mixed-effects model estimate of the overall treatment effect is equivalent to the summary measures approach estimate presented in section 9. However, for unbalanced data such equivalence is not verified. In general, the summary measures approach is not fully efficient in the following three situations. When the data exhibits a more complex correlation structure; when there is missing data; or when the trial design is unbalanced in some sense. A more elaborate discussion of these three situations is provided by Senn, Stevens [26]. For estimating the overall treatment effect, the linear mixed-effects model weights the individual means according to their precision, which depend on the number of observations available under a particular individual. The summary measures approach estimates the overall treatment effect by equally weighting the individual means. For balanced data, the precisions attached to each individual mean are in fact equal, so the linear mixed-effects model yields an overall treatment effect estimate exactly equal to the one resulting from the summary measures approach. When the number of observations differs between individuals, the precisions of the individual means are distinct under the linear mixed-effects model, resulting in an estimate of the overall treatment effect that is different from the one resulting from the summary measures approach. The summary measures approach application to unbalanced data is not exemplified in this document because its application is considered inappropriate for that case. For estimating the overall treatment effect, the methods that weight the individual information based on the relative precision are more appropriate. It can be shown that such relative precision-based estimators are best linear unbiased predictors under the assumptions considered. From the statistical methods so far presented only the weighted linear fixed-effects model fitted in Program 11.5, the weighted linear fixed-effects model of difference used in Program 12.5, and the linear mixed-effects model exemplified in Program 13.2; consider the relative precision of the individual data to estimate the overall treatment effect. The overall treatment effect estimate resulting from the linear mixed-effects model displayed in *Table* 13.20 and *Table* 13.24 differs, but sits very close to the estimate obtained from the linear fixed-effects models presented in Table 11.15 and Table 12.18. However, the estimate of the standard error of the overall treatment effect is lower for the linear fixed-effects models, as can be noticed from the same tables. This fact is misleading in suggesting that the linear fixed-effects models are more efficient than the linear mixed-effects models for estimating the overall treatment effect. It can be shown that when there are significant sources of variation such as a significant treatment by patient interaction, the estimate of the standard error of the overall treatment effect is downwards biased when such random terms are not fitted in the model [20]. The confidence intervals of the variance parameters of the linear mixed-effects model displayed in *Table* 13.18 suggest that all the random terms are significant. Therefore, the confidence interval and the standard error estimates of the overall treatment effect resulting from the linear mixed-effects model displayed in *Table* 13.24 are more reliable and should be preferred.

13.2 Estimation and inference on individual treatment effects

Individual treatment effects can be estimated from data arising from series of n-of-1 trials. So that a treatment preference can be determined for each patient, based on its own treatment effect rather than on the overall treatment effect, depending on the information available for that particular patient. For any statistical model, the individual treatment effect is defined as the expectation of the outcome variable conditioned on a given patient and treatment minus the expectation of the outcome variable conditioned on the same patient and another treatment. Given this definition, for a linear mixed-effects model such as the one used in Program 13.1 and Program 13.2, the individual treatment effects depend only on the “Treatment” fixed term and on the “Treatment” by “Patient” random interaction term. The “Treatment” by “Patient” random interaction term is distinct for each patient and treatment and is the reason for the distinctiveness of the individual treatment effects. A linear mixed-effects model with a “Treatment” term and without a “Treatment” by “Patient” random interaction term results in the same individual treatment effect for every patient in the sample. In such a model, the individual treatment effect coincides with the overall treatment effect. When estimating individual treatment effects form a linear model with the purpose of individualizing medicine it is therefore recommended to include a treatment by patient interaction term in the model. The following program can be considered an extension of Program 13.1, where the estimation of individual treatment effects for each patient present in the dataset is achieved. The code is based on code published by Brown and Prescott [21] for similar linear mixed-effects models applied to the analysis of multicentre trials. The balanced data “NORMAL.ndata” created before by either Program 4.1 or Program 5.4 is used in the program below. The Kenward and Roger’s method is used for inference on the individual treatment effects [25].

Program 13.3: Estimating individual treatment effects using 'proc mixed' – Kenward and Roger's inference.

```
title "Shrunk individual treatment effects";
title2 "Balanced data";
title3 "Kenward-Roger";
proc mixed
                data=NORMAL.ndata
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Treatment (ref=first) Patient Cycle;
        model Y = Treatment /
                ddfm=KR /* inference based on Kenward and Roger's method */
                htype=2; * type 2 hypothesis test ;
        random Patient Patient*Cycle Patient*Treatment;
        estimate 'B-A,1' Treatment 1 -1 |
                Patient*Treatment       1 0 0 0 0 0 0 0 0 0 0 0
                                        -1 0 0 0 0 0 0 0 0 0 0 0 /
                                                alpha=0.05 /* level for CI */
                                                cl; * t-type CI for estimates ;
        estimate 'B-A,2' Treatment 1 -1 |
                Patient*Treatment       0 1 0 0 0 0 0 0 0 0 0 0
                                        0 -1 0 0 0 0 0 0 0 0 0 0;
        estimate 'B-A,3' Treatment 1 -1 |
                Patient*Treatment       0 0 1 0 0 0 0 0 0 0 0 0
                                        0 0 -1 0 0 0 0 0 0 0 0 0;
        estimate 'B-A,4' Treatment 1 -1 |
                Patient*Treatment       0 0 0 1 0 0 0 0 0 0 0 0
                                        0 0 0 -1 0 0 0 0 0 0 0 0;
        estimate 'B-A,5' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 1 0 0 0 0 0 0 0
                                        0 0 0 0 -1 0 0 0 0 0 0 0;
        estimate 'B-A,6' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 1 0 0 0 0 0 0
                                        0 0 0 0 0 -1 0 0 0 0 0 0;
        estimate 'B-A,7' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 0 1 0 0 0 0 0
                                        0 0 0 0 0 0 -1 0 0 0 0 0;
        estimate 'B-A,8' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 0 0 1 0 0 0 0
                                        0 0 0 0 0 0 0 -1 0 0 0 0;
        estimate 'B-A,9' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 0 0 0 1 0 0 0
                                        0 0 0 0 0 0 0 0 -1 0 0 0;
        estimate 'B-A,10' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 0 0 0 0 1 0 0
                                        0 0 0 0 0 0 0 0 0 -1 0 0;
        estimate 'B-A,11' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 0 0 0 0 0 1 0
                                        0 0 0 0 0 0 0 0 0 0 -1 0;
        estimate 'B-A,12' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 0 0 0 0 0 0 1
                                        0 0 0 0 0 0 0 0 0 0 0 -1;
        ods output estimates=NORMAL.pdata(keep=Label Estimate);
run;
```

Program 13.3 presents some similarities in relation to Program 13.1, with fewer options being used in the 'model' statement, without a 'lsmeans' statement, and the addition of several 'estimate' statements and an 'ods output statement'. The purpose of Program 13.3 is to demonstrate the estimation of the shrunk individual treatment effects for balanced data, so the 'title' statements in the beginning of Program 13.3 reflect the program's purpose and permit the identification of the output. In fact, the individual treatment effects resulting from the linear mixed-effects models are referred as shrunk in the literature. More information on this is provided later. Enclosed in the 'mixed' procedure in Program 13.3 there are twelve 'estimate' statements, one for each patient's individual treatment effect estimate. Each 'estimate' statement has an appropriate label to permit the identification of the patient's individual treatment effect estimate in the software's output. For each patient's 'estimate' statement the coefficients of the "Treatment" fixed term are defined as 1 under treatment "B" and as -1 under treatment "A". The "Treatment" fixed term's coefficients are common to all the patients in the sample. The "Treatment" by "Patient" random interaction term's coefficients are the ones that differ between the patients. The first row of twelve elements after the "Patient*Treatment" term refers to treatment "B" while the second row refers to treatment "A". The column defines the position of the patient's coefficient. The first column refers to the first patient, the second column is relative to the second patient, and so on, until the twelfth column which is respective to the twelfth patient. On each 'estimate' statement after the "Patient*Treatment" term there is a 1 for treatment "B" and a -1 for treatment "A" on the position relative to the patient defined on that statement, and there is a 0 on the position relative to the other patients. This way the individual treatment effect is computed according to the definition previously given. Note that each 'estimate' statement contains a vertical bar (|) between the fixed effects terms and the random effects terms. In this case the "Treatment" term is fixed, and the "Patient*Treatment" term is random. The 'ods output' statement saves data relative to the twelve individual treatment effect estimates to a dataset named "pdata" located on the "NORMAL" library. Remember that Program 2.1 defines the "NORMAL" local library. The "NORMAL.pdata" dataset is saved there for later usage by other programs. The execution of Program 13.3 outputs nine distinct tables. Program 13.3 outputs eight tables previously presented, the tables in question being Table 13.1, Table 13.2, Table 13.3, Table 13.4, Table 13.5, Table 13.6, Table 13.7 and Table 13.10. The new table results from the twelve 'estimate' statements of Program 13.3, and holds the individual treatment effects. The new table holding the individual treatment effects is presented below.

Table 13.25: Shrunk individual treatment effects for balanced data - Kenward and Roger's inference.

Estimates								
Label	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
B-A,1	195.13	71.5475	1.03	2.73	0.2174	0.05	-649.79	1040.04
B-A,2	169.64	71.5475	1.03	2.37	0.2478	0.05	-675.27	1014.56
B-A,3	165.12	71.5475	1.03	2.31	0.2541	0.05	-679.80	1010.03
B-A,4	217.93	71.5475	1.03	3.05	0.1957	0.05	-626.99	1062.84
B-A,5	201.67	71.5475	1.03	2.82	0.2107	0.05	-643.25	1046.58
B-A,6	163.29	71.5475	1.03	2.28	0.2567	0.05	-681.63	1008.20
B-A,7	186.21	71.5475	1.03	2.60	0.2272	0.05	-658.71	1031.12
B-A,8	182.29	71.5475	1.03	2.55	0.2318	0.05	-662.62	1027.21
B-A,9	213.59	71.5475	1.03	2.99	0.1995	0.05	-631.33	1058.50
B-A,10	199.53	71.5475	1.03	2.79	0.2129	0.05	-645.38	1044.45
B-A,11	193.42	71.5475	1.03	2.70	0.2192	0.05	-651.50	1038.33
B-A,12	176.85	71.5475	1.03	2.47	0.2385	0.05	-668.06	1021.77

The standard error and the degrees of freedom in respect to each individual treatment effect represented in Table 13.25 are exactly equal. This is a consequence of the model assuming the same variance parameters for all the patients, and of the fact that the dataset used is balanced, i.e. the same number of observations is available for each patient. Observe that the estimates of the individual treatment effects no longer coincide with the estimates obtained from the methods previously demonstrated. It was previously noted that the estimates of the individual treatment effects resulting from the linear fixed-effects models displayed in Table 11.16 and Table 12.19 coincide with the naïve individual treatment effects presented in Table 7.1. However, the shrunk individual treatment effect estimates displayed in Table 13.25 are distinct from the naïve individual treatment effects in Table 7.1. It can be shown that each shrunk individual treatment effect is a weighted average of two statistics, the overall treatment effect and the respective naïve individual treatment effect, where the weights depend on the variance parameters of the linear mixed-effects model, and on the number of observations available for the individual in question. For a proper demonstration see section 7.5 and section 7.6 of Verbeke and Molenberghs [24], or section 8.6 and section 8.7 of Fitzmaurice, Laird [27]. This is in fact the reason why the individual treatment effect estimates resulting from linear mixed-effect models are said to be shrunk. The naive individual treatment effects are pulled towards the overall treatment effect resulting in the shrunk individual treatment effects, depending on the weights given to each statistic.

The fact that the shrunk individual treatment effects also depend on the overall treatment effect in addition to the naïve individual treatment effect, implies that the shrunk individual treatment effects rely on more information than the naïve individual treatment effects, which only rely on the information of the individual in question. In other words, the shrunk individual treatment effects are estimated from the data of the individual in question and from the data of the other individuals present on the sample. In general, statistics that rely on more information have lower standard error than statistics that rely on less information. Note that the standard error of the shrunk individual treatment effect estimates in Table 13.25 is lower than the standard error of the naïve individual treatment effect estimates from Table 11.16 and Table 12.19. However, the confidence intervals of the shrunk individual treatment effects from Table 13.25 are wider than the confidence intervals of the naïve individual treatment effects, obtained from the naïve method presented in Table 7.1, and obtained from the linear fixed-effects models displayed in Table 11.16 and Table 12.19. Note that the p-value and the confidence interval of each individual treatment effect estimate present in Table 13.25 suggest that there is no significant difference between the treatments for all the patients in the series. However, the p-value and confidence interval of the overall treatment effect in Table 13.12 suggest that the treatments differ significantly between each other on average.

The next program estimates the individual treatment effects while using the asymptotic method of inference, instead of the Kenward and Roger’s method used before in Program 13.3. Below the results are presented and compared with the ones obtained from the programs above.

Program 13.4: Estimating individual treatment effects using 'proc mixed' – Asymptotic inference.

```
title "Shrunk individual treatment effects";
title2 "Balanced data";
proc mixed
                data=NORMAL.ndata
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Treatment (ref=first) Patient Cycle;
        model Y = Treatment /
                htype=2; * type 2 hypothesis test ;
        random Patient Patient*Cycle Patient*Treatment;
        estimate 'B-A,1' Treatment 1 -1 |
                Patient*Treatment       1 0 0 0 0 0 0 0 0 0 0 0
                                        -1 0 0 0 0 0 0 0 0 0 0 0 /
                                                alpha=0.05 /* level for CI */
                                                cl; * t-type CI for estimates ;
        estimate 'B-A,2' Treatment 1 -1 |
                Patient*Treatment       0 1 0 0 0 0 0 0 0 0 0 0
                                        0 -1 0 0 0 0 0 0 0 0 0 0;
        estimate 'B-A,3' Treatment 1 -1 |
                Patient*Treatment       0 0 1 0 0 0 0 0 0 0 0 0
                                        0 0 -1 0 0 0 0 0 0 0 0 0;
        estimate 'B-A,4' Treatment 1 -1 |
                Patient*Treatment       0 0 0 1 0 0 0 0 0 0 0 0
                                        0 0 0 -1 0 0 0 0 0 0 0 0;
        estimate 'B-A,5' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 1 0 0 0 0 0 0 0
                                        0 0 0 0 -1 0 0 0 0 0 0 0;
        estimate 'B-A,6' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 1 0 0 0 0 0 0
                                        0 0 0 0 0 -1 0 0 0 0 0 0;
        estimate 'B-A,7' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 0 1 0 0 0 0 0
                                        0 0 0 0 0 0 -1 0 0 0 0 0;
        estimate 'B-A,8' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 0 0 1 0 0 0 0
                                        0 0 0 0 0 0 0 -1 0 0 0 0;
        estimate 'B-A,9' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 0 0 0 1 0 0 0
                                        0 0 0 0 0 0 0 0 -1 0 0 0;
        estimate 'B-A,10' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 0 0 0 0 1 0 0
                                        0 0 0 0 0 0 0 0 0 -1 0 0;
        estimate 'B-A,11' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 0 0 0 0 0 1 0
                                        0 0 0 0 0 0 0 0 0 0 -1 0;
        estimate 'B-A,12' Treatment 1 -1 |
                Patient*Treatment       0 0 0 0 0 0 0 0 0 0 0 1
```

```
                                         0 0 0 0 0 0 0 0 0 0 0 -1;
run;
```

The differences between Program 13.4 and Program 13.3 are not so many. Under the 'mixed' procedure of the later program there is no 'ddfm=KR' option in the 'model' statement, and there is no 'ods output' statement. Program 13.4 is very similar to programs presented above in section 13 so no further details in respect to the program are provided. If you need more details, please refer to the text following previous programs of section 13, in particular to the text following Program 13.3. Program 13.4 outputs seven tables previously presented, the tables in question being Table 13.2, Table 13.3, Table 13.4, Table 13.5, Table 13.6, Table 13.7 and Table 13.10. The new tables are presented below.

Table 13.26: Model information.

Model Information	
Data Set	NORMAL.NDATA
Dependent Variable	Y
Covariance Structure	Variance Components
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Table 13.27: Shrunk individual treatment effects for balanced data - Asymptotic inference.

Estimates								
Label	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
B-A,1	195.13	44.5523	11	4.38	0.0011	0.05	97.0706	293.19
B-A,2	169.64	44.5523	11	3.81	0.0029	0.05	71.5834	267.70
B-A,3	165.12	44.5523	11	3.71	0.0035	0.05	67.0605	263.18
B-A,4	217.93	44.5523	11	4.89	0.0005	0.05	119.87	315.99
B-A,5	201.67	44.5523	11	4.53	0.0009	0.05	103.61	299.73
B-A,6	163.29	44.5523	11	3.67	0.0037	0.05	65.2269	261.35
B-A,7	186.21	44.5523	11	4.18	0.0015	0.05	88.1470	284.27
B-A,8	182.29	44.5523	11	4.09	0.0018	0.05	84.2353	280.35
B-A,9	213.59	44.5523	11	4.79	0.0006	0.05	115.53	311.65
B-A,10	199.53	44.5523	11	4.48	0.0009	0.05	101.47	297.59
B-A,11	193.42	44.5523	11	4.34	0.0012	0.05	95.3592	291.48
B-A,12	176.85	44.5523	11	3.97	0.0022	0.05	78.7956	274.91

As expected the individual treatment effect estimates are the same in Table 13.27 as in Table 13.25, being the other statistics quite different, reflecting the change in the methods used for the computation of the standard error and the degrees of freedom. In fact, the standard error resulting from the asymptotic method displayed in Table 13.27 is much lower than the Kenward and Roger’s standard error held in Table 13.25. Since the dataset is balanced, and the variance parameters of the model are assumed equal for all the patients, the standard error of the individual treatment effect estimate is equal across the patients. Note that the p-values and the confidence intervals of the individual treatment effect estimates in Table 13.27 suggest a significant difference between the treatments for every patient without exception. A suggestion that contrasts with the one induced by the results of the Kenward and Roger’s method presented in Table 13.25!

In order to perform a comparison of the shrunk individual treatment effects with the naïve individual treatment effects, the next two programs have been developed to compute the extreme values and the summary statistics of the shrunk individual treatment effects. The shrunk individual treatment effects for the balanced dataset “NORMAL.ndata” used as example are contained in both Table 13.25 and Table 13.27. The tables are relatively small so the minimum and maximum individual treatment effects can simply be determined by observation. For larger tables the problem is not that simple. The next program computes the extreme values of the individual treatment effects.

Program 13.5: Computing the extreme values of shrunk individual treatment effects.

```
title "Extreme values of shrunk individual treatment effects";
proc sql;
        select          substring(Label from 5) as Patient,
                        Estimate as Extreme
        from NORMAL.pdata
        having Estimate eq min(Estimate)
        or Estimate eq max(Estimate);
quit;
```

Program 13.5 is very similar to Program 7.2 written to compute the extreme values of the naïve individual treatment effects. Program 13.5 is a very simple SQL program, which determines the minimum, and maximum shrunk individual treatment effects by extracting the patient substring from the “Label” variable and selecting the “Estimate” variable from “NORMAL.pdata” where the values of “Estimate” are minimum and maximum. The table below is obtained from running Program 13.5.

Table 13.28: Extreme values of shrunk individual treatment effects.

Patient	**Extreme**
4	217.93
6	163.29

According to Table 13.28, the highest shrunk individual treatment effect of 217.93 units was computed for patient number 4, while patient number 6 registered the lowest shrunk individual treatment effect of 163.29 units. The minimum and maximum naïve and shrunk individual treatment effects are observed on the same patients. However, a comparison of Table 13.28 with Table 7.2 reveals that the values of the minimum and maximum individual treatment effects are quite different between both methods. Due to shrinkage, the extremes of the individual treatment effects from Table 13.28 are closer to the estimate of the overall treatment effect of 188.72.

The following program uses the ‘summary’ procedure to compute several location measures on the dataset of shrunk individual treatment effects previously created by Program 13.3.

Program 13.6: Computing summary statistics of shrunk individual treatment effects.

```
title "Summary statistics of shrunk individual treatment effects";
proc summary
                    data=NORMAL.pdata
                    print
                    maxdec=2
                    min q1 mean median q3 max;
          var Estimate;
run;
```

The “NORMAL.pdata” dataset is supplied to the ‘data’ option of the ‘proc summary’ statement. The ‘summary’ procedure suppresses printing of the results by default, so the ‘print’ option is defined in the statement. The ‘maxdec’ option defines the number of significant figures in the results; in this case, the results are rounded to two decimal places. Afterwards several options defining the desired statistics are written. The requested statistics are in this order, the minimum, the first quartile, the mean, the median, the third quartile and the maximum. The ‘var’ statement defines the variable for computation of the statistics, in this case variable “Estimate” holding the shrunk individual treatment effects. The summary procedure used in Program 13.6 writes the following table to the SAS software’s output.

Table 13.29: Summary statistics of shrunk individual treatment effects.

Analysis Variable : Estimate					
Minimum	**Lower Quartile**	**Mean**	**Median**	**Upper Quartile**	**Maximum**
163.29	173.25	188.72	189.81	200.60	217.93

Table 13.29 displays several measures of location of the shrunk individual treatment effects. The mean of the shrunk individual treatment effects is equal to the overall treatment effect resulting from the linear mixed-effects model. This is observed for balanced data and generally is not observed for unbalanced data. All the measures of location in Table 13.29 are closer to the overall mean when compared with the naïve results displayed in Table 7.3, suggesting a lower variation between patients of the shrunk individual treatment effects.

The number of n-of-1 trials in a series can be much larger than the twelve present in the datasets used in the previous programs. The procedures developed to fit linear models in the SAS software require writing an ‘estimate’ statement for each desired individual treatment effect. Writing ‘estimate’ statements for all the n-of-1 trials in a large series requires a significant amount of work and time from the programmer. A set of macros was written to lessen the effort required to write code for the estimation of individual treatment effects arising from series of n-of-1 trials [4, 5]. The macros can be applied to similar statistical models and problems such as the statistical analysis of multicentre trials. The inner details of these long and complex macros are not exposed here. Detailing the code used in the macros would require a very large textbook. Such extensive description of the code is not the purpose of this document. The average user does not need to know how to write macros, being more interested in learning how to use the macros. The reader interested in improving his macro writing skills and interested in deciphering the macros used in programs throughout this document can consult the works of Burlew [4] and Carpenter [5]. The macros developed for the estimation of individual treatment effects are demonstrated in the next four programs. The macros in question are Macro A.1, Macro A.2, and Macro A.3 presented in Appendix A of this document.

Program 13.7: Writing a global variable macro array.

```
* Write global macro variable array ;
%estimate_full_global(
          array=vfull_unbalanced,
          subject=Patient,
          treatment=Treatment,
          data=NORMAL.ndata_unbalanced
)
```

In Program 13.7, the %estimate_full_global macro is invoked. The %estimate_full_global macro writes a set of global macro variables. All the macro variables created by %estimate_full_global are prefixed by the word defined in the “array” parameter of this macro. When using macros with global macro variables there is the possibility that macro variables with the exact same names as the ones being in use in the main program are declared or reassigned inside the macro definition, leading to unexpected behaviour of the whole program. For this reason, it is considered a good programming practice to declare macro variables in local scope inside the actual macro definitions. It is recommended to avoid usage of global macro variables inside macro definitions. Global macro variables shall be permitted only in the main program. The development of the %estimate_full_global macro avoids this problem. The final user of the macros, the programmer who is writing the main program can define the prefix of the set of global macros at his own free will. Moreover, by avoiding the definition of other macro variables with the same prefix, conflicts can be prevented. The %estimate_full_global macro variable accepts a “data” argument, which in this case is set as the “NORMAL.ndata_unbalanced” dataset. The “subject” parameter defines the categorical variable indicating the subject or patient in the series of n-of-1 trials. In this case, the “Patient” variable is supplied to the “subject” parameter. The “treatment” parameter of %estimate_full_global defines the categorical variable indicating the treatment in the series of n-of-1 trials, and is set to the “Treatment” variable from “NORMAL.ndata_unbalanced”. The variables supplied to the “subject” and “treatment” parameters must obviously be present in the dataset provided to the “data” parameter of the %estimate_full_global macro. The %estimate_full_global macro accepts a “ref” parameter that can be set as “first”, “last” or any level of the treatment category variable. The “ref” parameter can be used to define the reference treatment and is set as “first” by default, where the first treatment level appearing in the dataset is used as the reference treatment. After writing the set of global macro variables, which can also be considered an array of macro variables, the “vfull_unbalanced” macro variable array is available for posterior use by other macros. Briefly, what %estimate_full_global does is to look into the dataset provided, determine the labels for every patient and treatment pair combination, determine the vectors containing the 0, -1 and 1 in the right place, such as the vectors in the ‘estimate’ statements of Program 13.3 or Program 13.4, and save this information to a global macro variable array.

The following program makes usage of the “vfull_unbalanced” macro variable array, created by Program 13.7.

Program 13.8: Estimating individual treatment effects using 'proc mixed' and KR inference - The macro way.

```
title "Shrunk individual treatment effects";
title2 "Unbalanced data";
title3 "Kenward-Roger";
proc mixed
                data=NORMAL.ndata_unbalanced
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Treatment (ref=first) Patient Cycle;
        model Y = Treatment /
                ddfm=KR /* inference based on Kenward and Roger's method */
                htype=2; * type 2 hypothesis test ;
        random Patient Patient*Cycle Patient*Treatment;
        %estimate_full(array=vfull_unbalanced, options=alpha=0.05 cl)
run;
```

The ‘mixed’ procedure is used in Program 13.8 to fit the same model previously used in Program 13.3 and Program 13.4 to the unbalanced dataset “NORMAL.ndata_unbalanced” created by either Program 4.4 or Program 5.5. The %estimate_full macro is placed where the ‘estimate’ statements of the ‘mixed’ procedure are usually placed. When the program is submitted for interpretation and the %estimate_full macro is expanded, all the possible ‘estimate’ statements are determined from the “vfull_unbalanced” macro variable array previously created in Program 13.7, and are inserted under the ‘mixed’ procedure. The %estimate_full macro supports a “random” parameter which is set to a vertical bar (|) by default. This is the required definition of the parameter for usage with the ’mixed’ procedure. The macro variables from the “vfull_unbalanced” array are available during the entire SAS session, unless they are deleted. The %estimate_full_free macro can be used to delete all the macro variables previously created by the %estimate_full_global macro. The %estimate_full_free macro is invoked on a posterior program in order to delete the “vfull_unbalanced” macro variable array from the running SAS session. Program 13.8 is similar to Program 13.2 referencing the same dataset, using the same ‘mixed’ procedure with a similar ‘model’ statement and the same ‘random’ statement, so the output of both programs is identical. Program 13.8 outputs *Table* 13.13, *Table* 13.14, *Table* 13.15, *Table* 13.16, *Table* 13.17, *Table* 13.18, *Table* 13.19, *Table* 13.22 and the table resulting from the ‘estimate’ statements inserted into the program by the %estimate_full macro. The additional table obtained from Program 13.8 is presented below. The table holds the individual treatment effect statistics.

Table 13.30: Shrunk individual treatment effects for unbalanced data - Kenward and Roger's inference.

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
B-A,1	200.77	73.9441	1.45	2.72	0.1577	0.05	-265.93	667.47
B-A,2	170.91	73.9441	1.45	2.31	0.1930	0.05	-295.79	637.61
B-A,3	165.61	73.9441	1.45	2.24	0.2005	0.05	-301.09	632.31
B-A,4	227.49	73.9441	1.45	3.08	0.1341	0.05	-239.21	694.18
B-A,5	208.44	73.9441	1.45	2.82	0.1503	0.05	-258.26	675.14
B-A,6	163.46	73.9441	1.45	2.21	0.2037	0.05	-303.24	630.16
B-A,7	190.32	73.9441	1.45	2.57	0.1688	0.05	-276.38	657.02
B-A,8	185.73	73.9441	1.45	2.51	0.1740	0.05	-280.96	652.43
B-A,9	222.40	73.9441	1.45	3.01	0.1382	0.05	-244.30	689.10
B-A,10	205.93	73.9441	1.45	2.78	0.1527	0.05	-260.77	672.63
B-A,11	203.76	73.3944	1.14	2.78	0.1945	0.05	-495.59	903.12
B-A,12	189.28	69.1510	1	2.74	0.2230	0.05	-689.36	1067.93

Observe that the shrunk individual treatment effects resulting from the unbalanced data presented in Table 13.30 are distinct from the shrunk individual treatment effects estimated from the balanced data laid out in Table 13.25 and in Table 13.27. The difference is well observed for all the patients without exception. In fact, an increase of the shrunk individual treatment effect estimates is observed going from the balanced data to the unbalanced one. The increase is observed for all the twelve estimates without exception. Remember that the overall treatment effect resulting from the linear mixed-effects model shifted to the right while going from the balanced dataset to the unbalanced one. The linear mixed-effects model overall treatment effect estimated from the balanced dataset as 188.72 units, is displayed in Table 13.8 and Table 13.12. While the linear mixed-effects model overall treatment effect estimated from the unbalanced dataset as 194.51 units, is presented in both *Table* 13.20 and *Table* 13.24. For a particular patient, the shrunk individual treatment effect estimate is defined as a weighted average of the overall treatment effect and the naïve individual treatment effect for that patient. This in turn implies that the shrunk individual treatment effects are pulled towards the overall treatment effect, depending on the values of those two weights. Therefore, a shift of the overall treatment effect in a particular direction implies a shift of the shrunk individual treatment effects on that same direction. The shrunk individual treatment effects go where the overall treatment effect goes! In fact, if the overall treatment effect moves, the shrunk individual treatment effects can either remain stationary, or follow the overall treatment effect. The shrunk individual treatment effects can never move in the opposite direction of the overall treatment effect. In theory, there can be two particular situations where a shrunk individual treatment effect does not follow the overall treatment effect. One is, when the number of observations available for an individual is very big. The other, is when the data varies much more between individuals, than within a given individual. See section 7.5 and section 7.6 of Verbeke and Molenberghs [24], or section 8.6 and section 8.7 of Fitzmaurice, Laird [27], for a mathematical description of this phenomenon. Now, it is important to remember that the difference between the balanced dataset and the unbalanced dataset is located only on patients 11 and 12. Patients 1 to 10 contain the exact same data on both datasets. Program 4.4 or alternatively Program 5.5 create an unbalanced dataset by taking the balanced dataset, and deleting four observations from patient 12 and two observations from patient 11. Despite patients, 1 to 10 containing the exact same data on both datasets, the shrunk individual treatment effects estimated from both datasets are quite different. While for these patients, the naïve individual treatment effects resulting from the balanced and unbalanced datasets remain equal. A visual observation of Table 7.1 and then Table 7.4, or Table 11.16 and then Table 11.17, allows one to make this comparison. Since the shrunk individual treatment effects depend on the data from all the patients in the series as well as on the data for the respective patient, a modification on the data from any of the patients implies a modification on all the shrunk individual treatment effects. A given naïve individual treatment effect depends only on the data from the patient in question, implying that a modification on the data from a patient in the series does not influence the results for a different patient. Regarding the standard error of the estimates, note that it is equal for patients 1 to 10. While the standard error of the estimates differs for patients 11 and 12. This is to be expected, since the same number of observations is available for patients 1 to 10, and the stochastic model assumes the same variance parameters for all the patients. The standard error of the estimate is lower for patient 11 than for patients 1 to 10. In addition, the standard error of the estimate is lower for patient 12 than for patient 11. These results do not make sense at all! Since there is less information available for patient 11 than there is for patients 1 to 10, and the same variance parameters are assumed for all the patients, there should be an increase in the standard error of the estimate. The amount of information available for patient 12 is lower when compared to the amount of information available for patient 11. Consequently, a further increase in the standard error of the shrunk individual treatment effect estimate is to be expected, but the opposite is observed here! It is possible that the SAS implementation of the Kenward and Roger's approach for the estimation of the standard error of individual predictions is not correct. At the time of writing the present document, the Kenward and Roger's approach is not understood well enough, and the underlying code of its implementation within the SAS software is not available, being the software closed source. Thus, an assessment of the correctness of the implementation of the Kenward and Roger's method within the SAS software is not possible. For the time being, it is safer to avoid using the Kenward and Roger's method to make inference on the individual treatment effect estimates. It is recommended to use the Kenward and Roger's method of inference for overall treatment effect estimates only. At the same confidence level, the individual estimate's confidence intervals resulting from the unbalanced dataset are narrower than their respective individual estimate's confidence intervals in the balanced dataset, except for patient 12 where the contrary is verified. Since the shrunk individual treatment effects are obtained from the information of all the patients in the series of n-of-1 trials, and there is less overall information in the unbalanced dataset than in the balanced one, the opposite is expected. This is yet another reason to doubt the results of the Kenward and Roger's method for inference of individual predictions implemented within the SAS software. Table 13.30 shows that, the origin is located inside the confidence interval of each of the twelve estimates of the individual treatment effect; suggesting that the two treatments do not differ significantly between each other, individually for each of the twelve patients. The same suggestion is made by the results of Table 13.25, which is obtained from the application of the Kenward and Roger's inference to the balanced dataset. However, Table 13.27 resulting from the application of the asymptotic method of inference to the balanced dataset suggests otherwise. At the time of writing the present document, the Kenward and Roger's method of inference for individual predictions is not recommended, for the two reasons given above.

The shrunk individual treatment effects are estimated from the unbalanced dataset "NORMAL.ndata_unbalanced" in the program that follows. The asymptotic method is used to estimate the standard error, and the confidence intervals of the estimates. The containment method is used to estimate the degrees of freedom. Below the results are analysed, and compared with the ones obtained from programs presented above.

Program 13.9: Estimating individual treatment effects using ‘proc mixed’ and asymptotic inference – The macro way.

```
title "Shrunk individual treatment effects";
title2 "Unbalanced data";
proc mixed
                data=NORMAL.ndata_unbalanced
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Treatment (ref=first) Patient Cycle;
        model Y = Treatment /
                htype=2; * type 2 hypothesis test ;
        random Patient Patient*Cycle Patient*Treatment;
        %estimate_full(array=vfull_unbalanced, options=alpha=0.05 cl)
run;

* Delete global macro variable array ;
%estimate_full_free(array=vfull_unbalanced)
```

The ‘mixed’ procedure in Program 13.9 is very similar to the one from Program 13.8, with only one observable difference. In Program 13.9 the ‘ddfm’ option is not specified in the ‘model’ statement, so the degrees of freedom method and the standard error method is left to the default assumed by the procedure. In this case, the estimator of the standard error of the predictions is conditioned on the variance parameters of the stochastic model. Such estimator should underestimate the true value of the standard error, and approach the true value as the number of individuals and the number of cycles increase towards infinity. For this reason, the method of inference considered in Program 13.9 is called asymptotic. The %estimate_full macro is used in the same way as above. The “vfull_unbalanced” macro variable array was previously created by Program 13.7 and remains present in the SAS system, unless it is removed from it. After the ‘mixed’ procedure of Program 13.9 is executed, there is no additional intention of accessing the data stored in the “vfull_unbalanced” macro variable array. As such, after the execution of the ‘mixed’ procedure, the %estimate_full_free macro is used to remove the ‘’vfull_unbalanced” macro variable array from the running SAS session. The output is similar to the output from Program 13.2. Therefore, Program 13.9 outputs *Table* 13.14, *Table* 13.15, *Table* 13.16, *Table* 13.17, *Table* 13.18, and *Table* 13.19, presented above. A table equivalent to *Table* 13.13 with distinct model information is produced, and a table equivalent to *Table* 13.22 containing the results of the type 2 tests of fixed effects is made available. The %estimate_full macro leads to a table containing the results for the individual treatment effects. The three tables are presented below.

Table 13.31: Model information.

Model Information	
Data Set	NORMAL.NDATA_UNBALANCED
Dependent Variable	Y
Covariance Structure	Variance Components
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Table 13.32: Type 2 tests for fixed effects.

Type 2 Tests of Fixed Effects				
Effect	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Treatment	1	11	41.06	<.0001

Table 13.33: Shrunk individual treatment effects for unbalanced data – Asymptotic inference.

Estimates								
Label	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
B-A,1	200.77	47.8979	11	4.19	0.0015	0.05	95.3510	306.20
B-A,2	170.91	47.8979	11	3.57	0.0044	0.05	65.4875	276.33
B-A,3	165.61	47.8979	11	3.46	0.0054	0.05	60.1879	271.03
B-A,4	227.49	47.8979	11	4.75	0.0006	0.05	122.06	332.91
B-A,5	208.44	47.8979	11	4.35	0.0012	0.05	103.01	313.86
B-A,6	163.46	47.8979	11	3.41	0.0058	0.05	58.0395	268.88
B-A,7	190.32	47.8979	11	3.97	0.0022	0.05	84.8952	295.74
B-A,8	185.73	47.8979	11	3.88	0.0026	0.05	80.3118	291.16
B-A,9	222.40	47.8979	11	4.64	0.0007	0.05	116.98	327.82
B-A,10	205.93	47.8979	11	4.30	0.0013	0.05	100.51	311.35
B-A,11	203.76	50.1834	11	4.06	0.0019	0.05	93.3116	314.22
B-A,12	189.28	52.8069	11	3.58	0.0043	0.05	73.0571	305.51

The interest of this section is to discuss estimation and inference on individual treatment effects. Therefore, all the attention is directed towards *Table* 13.33. The individual treatment effect estimates presented on *Table* 13.33 are the same as the ones presented above on Table 13.30. The estimation method used in both

Program 13.8 and Program 13.9 is the same. It is the method used to estimate the degrees of freedom and the standard error of the estimates, which differs between the two programs. The estimates of the individual treatment effects being the same as in Table 13.30 are discussed in the text following that table so are not discussed again. Let us concentrate on the standard error of the estimates, the degrees of freedom and the confidence intervals. The unbalanced dataset contains the same amount of information for patients 1 to 10. Since the stochastic model assumes the same variance components for all the patients in the series of n-of-1 trials, the same value of the standard error is expected for those 10 patients, which is observed here. Patient number 11 contains information for 2 cycles instead of 3 cycles, so an increase in the standard error is observed. Finally, for patient number 12, the amount of information is further reduced to 1 cycle with a consequent increase in the standard error. All this makes sense and is to be expected. The estimator of the standard error presented in *Table* 13.33 is conditioned on the value of the variance parameters of the model. Since there are very few individuals and cycles in the series of n-of-1 trials exemplified here, this method should underestimate the standard error. The Kenward and Roger's method used to compute the estimates of the standard error presented in Table 13.30 attempts to correct for the uncertainty in the variance parameters, hence the higher values of the standard error of the individual estimates displayed in Table 13.30, when compared to the ones displayed in *Table* 13.33. The degrees of freedom is the same for all the 12 patients, despite patients 11 and 12 having less information than the other 10 patients. The confidence interval for patient 11 is wider than the confidence interval relative to patients 1 to 10. In addition, the confidence interval for patient 12 is wider than for patient 11. The differences observed in the confidence intervals between patients is expected, given the differences observed in the standard error across the 12 patients. The asymptotic method of inference used to obtain *Table* 13.33 suggests that the treatments differ significantly individually for all the 12 patients. A suggestion that contradicts the one made by Table 13.30, which results from the Kenward and Roger's method of inference. However, the standard error of the individual estimates of the treatment effect may be wrong for both methods. The asymptotic method of inference should be useful for a large series of n-of-1 trials, which is not the case here. On the other hand, the results of the Kenward and Roger's method have been questioned before for several reasons. This very interesting question is left for future research. The standard error and confidence intervals of the shrunk individual treatment effects can be estimated by simulation. Development and testing of code for this purpose is left for the future.

An alternative way of coding the linear mixed-effects model discussed in the current section is demonstrated below. Once more, the code leading to the individual treatments effects is exemplified. The "NORMAL.ndata" dataset previously obtained from either Program 4.1 or Program 5.4 is used for the demonstration. The differences in respect to previous programs are highlighted when appropriate.

Program 13.10: Estimating individual treatment effects using 'proc mixed'- The alternative way.

```
title "Shrunk individual treatment effects";
title2 "Balanced data";
proc mixed
        data=NORMAL.ndata
        alpha=0.05 /* level for CI */
        cl /* CI for covariance parameters */
        method=REML; * REML method for estimation ;
    class Treatment (ref=first) Patient Cycle;
    model Y = Treatment /
        alpha=0.05 /* level for CI */
        cl /* t-type CI for fixed effects */
        htype=2 /* type 2 hypothesis test */
        solution; * solution for fixed effects ;
    random intercept Treatment Cycle /
        subject=Patient;
    estimate 'B-A,1' Treatment 1 -1 | Treatment 1 -1 /
                subject 1
                alpha=0.05 /* level for CI */
                cl; * t-type CI for estimates ;
    estimate 'B-A,2' Treatment 1 -1 | Treatment 1 -1 /
                subject 0 1;
    estimate 'B-A,3' Treatment 1 -1 | Treatment 1 -1 /
                subject 0 0 1;
    estimate 'B-A,4' Treatment 1 -1 | Treatment 1 -1 /
                subject 0 0 0 1;
    estimate 'B-A,5' Treatment 1 -1 | Treatment 1 -1 /
                subject 0 0 0 0 1;
    estimate 'B-A,6' Treatment 1 -1 | Treatment 1 -1 /
                subject 0 0 0 0 0 1;
    estimate 'B-A,7' Treatment 1 -1 | Treatment 1 -1 /
                subject 0 0 0 0 0 0 1;
    estimate 'B-A,8' Treatment 1 -1 | Treatment 1 -1 /
                subject 0 0 0 0 0 0 0 1;
    estimate 'B-A,9' Treatment 1 -1 | Treatment 1 -1 /
                subject 0 0 0 0 0 0 0 0 1;
    estimate 'B-A,10' Treatment 1 -1 | Treatment 1 -1 /
                subject 0 0 0 0 0 0 0 0 0 1;
    estimate 'B-A,11' Treatment 1 -1 | Treatment 1 -1 /
                subject 0 0 0 0 0 0 0 0 0 0 1;
    estimate 'B-A,12' Treatment 1 -1 | Treatment 1 -1 /
                subject 0 0 0 0 0 0 0 0 0 0 0 1;
run;
```

Program 13.10 is similar to Program 13.4 fitting the same linear mixed-effects model to the same dataset. The differences are located mainly on the ‘random’ statement and on the ‘estimate’ statements. Unlike in Program 13.4, the “Patient” variable is absent from the terms of the ‘random’ statement. Instead, the ‘subject’ option of the same ‘random’ statement is set to the “Patient” variable. Then the ‘intercept’ keyword, the “Treatment” and the “Cycle” variables are defined as terms of the ‘random’ statement. The “Patient” variable in the ‘subject’ option indicates that the random effect terms are nested within the patient. As such for each patient there is an ‘intercept’, a “Treatment” term and a “Cycle” term. This is an alternative way of specifying a patient random effect, a treatment by patient random interaction, and a cycle by patient random interaction. The twelve ‘estimate’ statements from Program 13.10 request the individual treatment effects. Note however that the structure of the ‘estimate’ statements from Program 13.10 is somewhat distinct from the one used on the ‘estimate’ statements from Program 13.4. The ‘estimate’ statements from Program 13.10 are simpler and easier to code; therefore some programmers might prefer this alternative. The label and the coefficients of the fixed-effect terms are identical between both programs. As usual, there is a vertical bar separating the fixed effect coefficients from the random effect coefficients. Since only the “Treatment” random effect term within the subject contributes to the individual treatment effect, only the coefficients of the “Treatment” random effect are specified. Here the coefficients are the same as in the fixed effects part of the same ‘estimate’ statement. Each ‘estimate’ statement contains a ‘subject’ option followed by a set of zeros and a single one. Here the coefficients are associated to the levels of the “Patient” variable, which is supplied to the ‘subject’ option of the ‘random’ statement. Therefore, there is a one in the position associated to the patient defining the desired individual treatment effect and a zero in each of the positions associated to the other patients. Note that the order of the patients in the ‘estimate’ statements is the same order the patients occupy in the dataset. The execution of Program 13.10 leads to the tables presented below.

Table 13.34: Model information.

Model Information	
Data Set	NORMAL.NDATA
Dependent Variable	Y
Covariance Structure	Variance Components
Subject Effect	Patient
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Table 13.35: Class level information.

Class Level Information		
Class	**Levels**	**Values**
Treatment	2	B A
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12
Cycle	3	1 2 3

Table 13.36: Dimensions.

Dimensions	
Covariance Parameters	4
Columns in X	3
Columns in Z per Subject	6
Subjects	12
Max Obs per Subject	6

Table 13.37: Number of observations.

Number of Observations	
Number of Observations Read	72
Number of Observations Used	72
Number of Observations Not Used	0

Table 13.38: Iteration history.

Iteration History			
Iteration	**Evaluations**	**-2 Res Log Like**	**Criterion**
0	1	931.31645659	
1	1	893.79244632	0.00000000

Table 13.39: Covariance parameter estimates.

Covariance Parameter Estimates					
Cov Parm	**Subject**	**Estimate**	**Alpha**	**Lower**	**Upper**
Intercept	**Patient**	19459	0.05	8999.93	68835
Treatment	**Patient**	886.34	0.05	102.22	4.11E13
Cycle	**Patient**	680.53	0.05	63.4043	3.797E24
Residual		11842	0.05	7220.27	22919

Table 13.40: Fit statistics.

Fit Statistics	
-2 Res Log Likelihood	893.8
AIC (Smaller is Better)	901.8
AICC (Smaller is Better)	902.4
BIC (Smaller is Better)	903.7

Table 13.41: Solution for fixed effects.

Solution for Fixed Effects									
Effect	**Treatment**	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Intercept		2625.75	45.2031	11	58.09	<.0001	0.05	2526.26	2725.24
Treatment	**B**	188.72	28.3838	11	6.65	<.0001	0.05	126.25	251.19
Treatment	**A**	0	.	.	.	.	.	.	.

Table 13.42: Type 2 tests of fixed effects.

Type 2 Tests of Fixed Effects				
Effect	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Treatment	1	11	44.21	<.0001

Table 13.43: Shrunk individual treatment effects.

Estimates								
Label	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
B-A,1	195.13	44.5523	11	4.38	0.0011	0.05	97.0706	293.19
B-A,2	169.64	44.5523	11	3.81	0.0029	0.05	71.5834	267.70
B-A,3	165.12	44.5523	11	3.71	0.0035	0.05	67.0605	263.18
B-A,4	217.93	44.5523	11	4.89	0.0005	0.05	119.87	315.99
B-A,5	201.67	44.5523	11	4.53	0.0009	0.05	103.61	299.73
B-A,6	163.29	44.5523	11	3.67	0.0037	0.05	65.2269	261.35
B-A,7	186.21	44.5523	11	4.18	0.0015	0.05	88.1470	284.27
B-A,8	182.29	44.5523	11	4.09	0.0018	0.05	84.2353	280.35
B-A,9	213.59	44.5523	11	4.79	0.0006	0.05	115.53	311.65
B-A,10	199.53	44.5523	11	4.48	0.0009	0.05	101.47	297.59
B-A,11	193.42	44.5523	11	4.34	0.0012	0.05	95.3592	291.48
B-A,12	176.85	44.5523	11	3.97	0.0022	0.05	78.7956	274.91

The ten tables obtained from Program 13.10 contain results that can be observed above. The results in question are obtained from both Program 13.1 and Program 13.4. The differences lie mainly in the presentation of the results within the tables. Observe that there is a row indicating the subject effect on Table 13.34, while such a row is absent from model information tables presented above. Table 13.36 is quite different from dimension tables seen above. Twelve subjects are modelled according to Table 13.36. The covariance parameter estimates and related statistics are contained within Table 13.39, where there is a column identified as “Subject”. On the “Subject” column, the “Patient” variable is presented on every row associated to a covariance parameter that is nested on that same “Patient” variable. The covariance parameters and related statistics are the same as the ones displayed in Table 13.6 above. All the other results are the same as the ones presented above. Most notably the table pertaining to the solution for fixed effects contains the same values as Table 13.8 presented on top of the current section. Moreover, the shrunk individual treatment effects and respective statistics are the same as the ones presented above on Table 13.27.

Requesting individual treatment effect estimates through the ‘mixed’ procedure requires writing an ‘estimate’ statement for each individual as desired. One may wish to obtain all the individual treatment effect estimates that can be obtained from a given dataset arising from series of n-of-1 trials. A series of n-of-1 trials may be composed of a high number of single n-of-1 trials. In this case writing an ‘estimate’ statement for each individual may seem a lot of work. A set of three macros developed to lessen the effort of requesting individual treatment effect estimates under the ‘mixed’ procedure are presented above. The %estimate_full_global, %estimate_full, and %estimate_full_free macros have been developed with the ‘estimate’ statements used in Program 13.3 and Program 13.4 in mind. When the linear mixed effects model is requested through the specification of the ‘subject’ option of the ‘random’ statement, the ‘estimate’ statements leading to the individual treatment effect estimates must be specified accordingly, and the referred set of three macros is inadequate. A new set of three macros inspired by the ‘estimate’ statements used in Program 13.10 was developed. Macro A.4, Macro A.5, and Macro A.6 are presented in Appendix A of this document. The usage of these three macros is exemplified in the next program.

Program 13.11: Estimating individual treatment effects using 'proc mixed'- The alternative macro way.

```
* Write global macro variable array ;
%estimate_full_random_global(
            array=vfull_ndata,
            subject=Patient,
            treatment=Treatment,
            data=NORMAL.ndata,
            ref=first
)

title "Shrunk individual treatment effects";
title2 "Balanced data";
proc mixed
                        data=NORMAL.ndata
                        alpha=0.05 /* level for CI */
                        cl /* CI for covariance parameters */
                        method=REML; * REML method for estimation ;
            class Treatment (ref=first) Patient Cycle;
            model Y = Treatment /
                        alpha=0.05 /* level for CI */
                        cl /* t-type CI for fixed effects */
                        htype=2 /* type 2 hypothesis test */
                        solution; * solution for fixed effects ;
            random intercept Treatment Cycle /
                        subject=Patient;
            %estimate_full_random(array=vfull_ndata, options=alpha=0.05 cl)
run;

* Delete global macro variable array ;
%estimate_full_random_free(array=vfull_ndata)
```

The ‘random’ statement is the same as the one used in Program 13.10, with the specification of the “Patient” variable in the ‘subject’ option. In this case, three macros are used to automatize the insertion of the ‘estimate’ statements leading to the individual treatment effect estimates. The %estimate_full_random_global macro is invoked first. This macro reads the necessary information from the

“NORMAL.ndata” dataset, and writes the “vfull_ndata” macro variable array to the system. The “vfull_ndata” macro variable array is in fact a set of macro variables where each macro variable name is prefixed by the word “vfull_ndata”. The variables associated to the macro variable array are declared in global scope, remaining in the computer’s memory until removed, or the SAS system is terminated. When the program is executed, the %estimate_full_random macro is expanded, information is read from the “vfull_ndata” macro variable array and the ‘estimate’ statements for all the individual treatment effects that can be estimated from the dataset are inserted under the ‘mixed’ procedure. Finally, the %estimate_full_random_free macro removes the “vfull_ndata” macro variable array from the system. Program 13.11 outputs the same ten tables as Program 13.10. The results are presented and discussed above, for that reason no further comments are made here.

14 Linear mixed-effects model of difference

The top of section 12, mentions a relationship between the linear fixed-effects model of difference and the linear fixed-effects model discussed in section 11. A similar relationship exists between the linear mixed-effects model of difference presented in this section, and the linear mixed-effects model presented in section 13. Therefore, if you read section 12 and have it in memory, you may feel a sensation of word repetition in this text. The linear mixed-effects model of difference cannot be fitted to the dataset “NORMAL.ndata” used in the previous section, without performing some preliminary arithmetic operations on this dataset. Either Program 4.5 or Program 5.6 take the “NORMAL.ndata” dataset, perform the required arithmetic operations on the outcome variable of this dataset, and create a new “NORMAL.ddata” dataset with a new outcome variable. The “NORMAL.ddata” dataset can be used to fit the linear mixed-effects model of difference discussed in the current section. The same “NORMAL.ddata” dataset is used throughout section 12, where the similar linear fixed-effects model of difference is discussed. The “dY” outcome variable in this dataset has a meaning that is different from the “Y” outcome variable in the original “NORMAL.ndata” dataset. The “Treatment” variable is absent from the “NORMAL.ddata” dataset so only the available “Patient” and “Cycle” variables can be fitted in the model. The linear mixed-effects model of difference used in the programs presented in the current section can be obtained from the one used in the programs presented in section 13. Note that the outcome variable “dY” is defined as the difference between observations of the variable “Y” registered under two treatments in the same cycle. It can be verified by means of elementary algebra, that the “Treatment” by “Patient” random interaction in the linear mixed-effects model leads to a single “Patient” random effect in the linear mixed-effects model of difference. While the “Cycle” by “Patient” random interaction in the former model is eliminated, and is absent from the latter model.

14.1 Estimation and inference on the overall treatment effect

The overall treatment effect is estimated from the dataset of outcome differences “NORMAL.ddata” in the program below. The ‘mixed’ procedure is used to fit the linear mixed-effects model of difference to the data.

Program 14.1: Fitting a linear mixed-effects model of difference using 'proc mixed' – Balanced data.

```
title "Balanced data";
proc mixed
        data=NORMAL.ddata
        alpha=0.05 /* level for CI */
        cl /* CI for covariance parameters */
        method=REML; * REML method for estimation ;
    class Patient;
    model dY = /
        alpha=0.05 /* level for CI */
        cl /* t-type CI for fixed effects */
        covb /* covariance matrix of fixed effects estimates */
        ddfm=KR /* inference based on Kenward and Roger's method */
        htype=2 /* type 2 hypothesis test */
        solution; * solution for fixed effects ;
    random Patient;
run;
```

Like in the programs above, Program 14.1 starts with some 'title' statements. The purpose of the 'title' statements is to insert the text between double quotes in the start of the program's output. It is usual for the SAS output to contain several tables and or plots resulting from several procedures. This way it can be difficult for the analyst to assess where the output from a program starts or ends. The 'title' statement permits the identification of the output. Program 14.1 contains comments in front of each statement's option. The purpose of the comments is to inform the reader of the purpose of each option. Therefore, only the most pertinent options are mentioned here. The 'data' option of the 'mixed' procedure specifies the dataset used to fit the linear mixed-effects model. In this case, the "Restricted Maximum Likelihood" method is used for estimation, as specified in the 'method' option of the 'mixed' procedure. Restricted maximum likelihood produces unbiased estimates of the covariance parameters of the stochastic model, and is therefore preferred over other methods of estimation, such as maximum likelihood. The 'class' statement specifies the categorical variables of the dataset to be included in the model. In the 'model' statement note that the "dY" variable is used as the outcome, and there are no explanatory variables. So no variables as fixed effects in this model. There is only the intercept in the fixed effects part of this linear mixed-effects model. In the linear mixed-effects model of difference, the intercept is equal to the overall treatment effect, so it plays a very important role in the analysis. The 'solution' option requests the solution for the fixed effects, where the overall treatment effect lies, so it must not be missed. Finally, the "Patient" categorical variable belongs to the random part of the model, being specified in the 'random' statement. The "Patient" random term does not contribute directly to the value of the overall treatment effect since its expectation is zero. The variance of the "Patient" error is what matters. Random variables that contribute significantly to explain the variation of the outcome variable lead to desired properties of the variance of the overall treatment effect. Such variables should be included in the linear mixed-effects model. This recommendation is discussed in more detail below. Program 14.1 outputs the following nine tables.

Table 14.1: Model information.

Model Information	
Data Set	NORMAL.DDATA
Dependent Variable	dY
Covariance Structure	Variance Components
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Kenward-Roger
Degrees of Freedom Method	Kenward-Roger

Table 14.2: Class level information.

Class Level Information		
Class	**Levels**	**Values**
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12

Table 14.3: Dimensions.

Dimensions	
Covariance Parameters	2
Columns in X	1
Columns in Z	12
Subjects	1
Max Obs per Subject	36

Table 14.4: Number of observations.

Number of Observations	
Number of Observations Read	36
Number of Observations Used	36
Number of Observations Not Used	0

Table 14.5: Iteration history.

Iteration History			
Iteration	**Evaluations**	**-2 Res Log Like**	**Criterion**
0	1	457.83663221	
1	1	457.67817788	0.00000000

Table 14.6: Covariance parameter estimates.

Covariance Parameter Estimates				
Cov Parm	**Estimate**	**Alpha**	**Lower**	**Upper**
Patient	1772.67	0.05	204.44	8.22E13
Residual	23685	0.05	14441	45838

Table 14.7: Fit statistics.

Fit Statistics	
-2 Res Log Likelihood	457.7
AIC (Smaller is Better)	461.7
AICC (Smaller is Better)	462.1
BIC (Smaller is Better)	462.6

Table 14.8: Solution for fixed effects.

Solution for Fixed Effects								
Effect	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Intercept	188.72	28.3838	11	6.65	<.0001	0.05	126.25	251.19

Table 14.9: Covariance matrix for fixed effects.

Covariance Matrix for Fixed Effects		
Row	**Effect**	**Col1**
1	Intercept	805.64

The information contained in the nine tables presented above is mostly easy to read. There are some exceptions however. The interpretation of some of the tables requires some knowledge of linear mixed-effects model theory. The harder to interpret tables are about to be discussed in more detail. Table 14.1 contains information relative to the linear mixed-effects model. Restricted maximum likelihood is the method used to estimate the parameters of the model. The Kenward and Roger’s method is used to estimate the standard error of the fixed-effects and the degrees of freedom [25]. The Kenward and Roger's method is requested through the 'ddfm' option of the 'model' statement in Program 14.1. The covariance parameter estimates of the model are displayed in Table 14.6. The confidence intervals of the covariance parameters do not contain zero, suggesting the significance of the respective parameters. The covariance parameters define the correlation structure of the model. Fitting a wrong correlation structure in the model may lead to biasedness of the standard error of the fixed effects, which in turn may lead to wrong inferences. A situation that can be dangerous! Araujo, Julious [20] discuss this issue in more detail for the specific case of the statistical analysis of series of n-of-1 trials. The overall treatment effect is equal to the intercept of the linear mixed-effects model of difference. Therefore, Table 14.8 contains the most sought-after statistics, such as the estimate of the overall treatment effect, and associated statistics: the standard error, degrees of freedom, t statistic, p-value, and lower and upper limits of the confidence interval. The confidence interval of the overall treatment effect estimate does not contain the origin, suggesting that the treatments differ significantly at the chosen significance level of 5%. The same suggestion is made by the p-value, which is much lower than the chosen significance level. Table 14.9 displays the variance of the overall treatment effect estimate, which equals the square of the standard error in Table 14.8. This document considers an asymptotic estimator as an alternative to the Kenward and Roger's method of estimation of the standard error of the fixed effects. For balanced data in combination with the current model, the so-called asymptotic estimator is exact, and equal to the Kenward and Roger's method, regardless of the size of the series of n-of-1 trials. For this reason, the application of the asymptotic method of estimation of the standard error of the fixed-effects to the analysis of balanced data is not exemplified here. Let us proceed with the statistical analysis of unbalanced data.

An unbalanced dataset of outcome differences can be obtained by running either Program 4.6 or Program 5.7. The following program fits a linear mixed-effects model of difference to the "NORMAL.ddata_unbalanced" dataset created by either one or the other of those two programs. The results are presented below. Opportunely, a comparison with the results from previous programs is made.

Program 14.2: Fitting a linear mixed-effects model of difference using ‘proc mixed’ - Unbalanced data.

```
title "Unbalanced data";
proc mixed
                data=NORMAL.ddata_unbalanced
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Patient;
        model dY = /
                alpha=0.05 /* level for CI */
                cl /* t-type CI for fixed effects */
                covb /* covariance matrix of fixed effects estimates */
                ddfm=KR /* inference based on Kenward and Roger's method */
                htype=2 /* type 2 hypothesis test */
                solution; * solution for fixed effects ;
        random Patient;
run;
```

The statements and options of the ‘mixed’ procedure used in Program 14.2 are very similar to the ones used in Program 14.1 above. The only difference lies in the dataset supplied to the ‘data’ option of the procedure. In Program 14.2, the “NORMAL.ddata_unbalanced” dataset is used. This dataset is unbalanced and adequate to fit the linear mixed-effects model of difference. Note that the Kenward and Roger’s method is used to compute the standard error of the fixed effects and the degrees of freedom [25]. The submission of Program 14.2 to the SAS system outputs the following nine tables.

Table 14.10: Model information.

Model Information	
Data Set	NORMAL.DDATA_UNBALANCED
Dependent Variable	dY
Covariance Structure	Variance Components
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Kenward-Roger
Degrees of Freedom Method	Kenward-Roger

Table 14.11: Class level information.

Class Level Information		
Class	**Levels**	**Values**
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12

Table 14.12: Dimensions.

Dimensions	
Covariance Parameters	2
Columns in X	1
Columns in Z	12
Subjects	1
Max Obs per Subject	33

Table 14.13: Number of observations.

Number of Observations	
Number of Observations Read	33
Number of Observations Used	33
Number of Observations Not Used	0

Table 14.14: Iteration history.

Iteration History			
Iteration	**Evaluations**	**-2 Res Log Like**	**Criterion**
0	1	419.84577988	
1	3	419.62973862	0.00000003
2	1	419.62973364	0.00000000

Table 14.15: Covariance parameter estimates.

Covariance Parameter Estimates				
Cov Parm	**Estimate**	**Alpha**	**Lower**	**Upper**
Patient	2199.41	0.05	284.37	1.912E11
Residual	24100	0.05	14431	48180

Table 14.16: Fit statistics.

Fit Statistics	
-2 Res Log Likelihood	419.6
AIC (Smaller is Better)	423.6
AICC (Smaller is Better)	424.0
BIC (Smaller is Better)	424.6

Table 14.17: Solution for fixed effects.

Solution for Fixed Effects								
Effect	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Intercept	194.51	30.4924	11.3	6.38	<.0001	0.05	127.59	261.43

Table 14.18: Covariance matrix for fixed effects.

Covariance Matrix for Fixed Effects		
Row	**Effect**	**Col1**
1	Intercept	929.79

The confidence intervals of the covariance parameters on Table 14.15 do not contain the origin, suggesting that the covariance parameters are significant. The standard error of the overall treatment effect depends on the covariance parameters of the model. For this reason, if the “Patient” variable is not fitted as random, the standard error of the overall treatment effect is downwards biased. Such situation would lead to wrong inferences [20]. Therefore, it is important to fit the “Patient” variable as random. The overall treatment effect is equal to the intercept of the fixed part of the linear mixed-effects model of difference. The overall treatment effect and related statistics are presented on Table 14.17. The standard error of the overall treatment effect is greater for the unbalanced dataset, when compared to the one resulting from the balanced dataset presented above on Table 14.8. The increase in the standard error of the overall treatment effect reflects less information contained in the unbalanced dataset. Remember that the unbalanced dataset is obtained by deleting observations from the balanced dataset. Program 14.2 estimates the standard error of the fixed effects and the degrees of freedom through the Kenward and Roger’s method. For unbalanced data, the model-based method of estimation of the standard error of the fixed effects tends to underestimate the actual value. In this particular situation, the standard error of the fixed effects depends on several quantities that are difficult to estimate. The Kenward and Roger’s method adjusts for these quantities, therefore providing better estimates. For this reason, the Kenward and Roger’s method is recommended to estimate the standard error of the overall treatment effect, particularly when the data is unbalanced across the individuals. The p-value being inferior to the significance level, suggests that the treatments differ significantly between each other. The same suggestion is made by the confidence interval, which does not contain the origin. Table 14.18 displays the variance of the overall treatment effect, which equals the square of the standard error of the same overall treatment effect displayed in Table 14.17. In this case, Table 14.18 does not provide any additional information, so for the linear mixed-effects model of difference, the ‘covb’ option can be left out of the ‘model’ statement in the ‘mixed’ procedure.

14.2 Estimation and inference on individual treatment effects

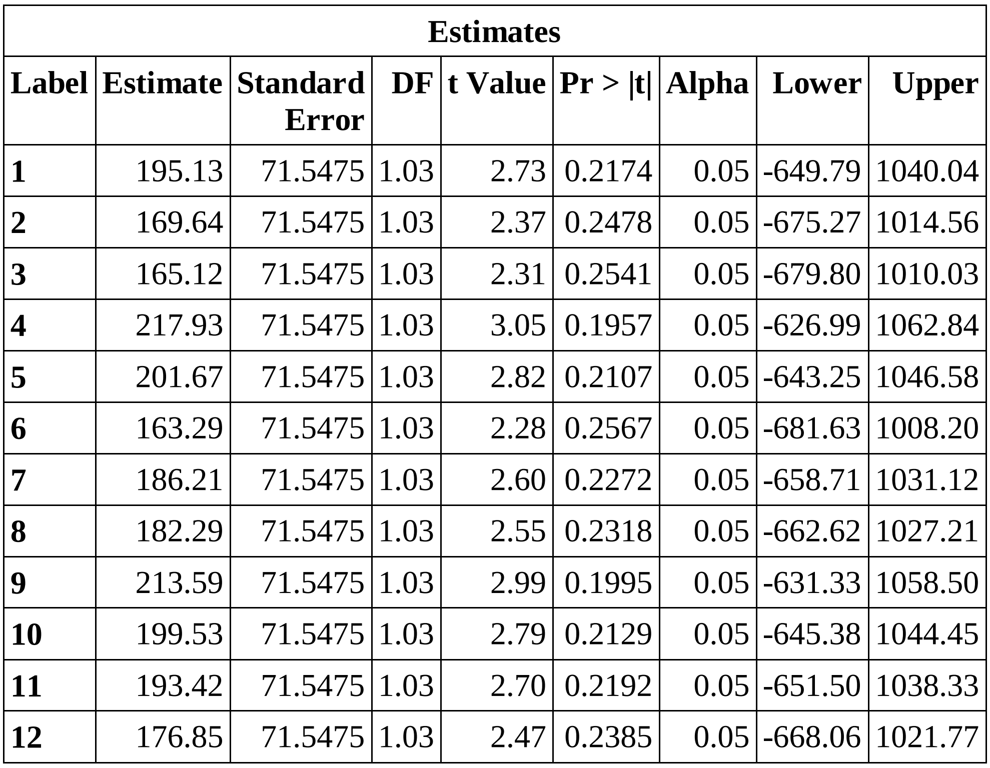
Individual treatment effects can be estimated from data arising from series of n-of-1 trials. So that a treatment preference can be determined for each patient, based on its own treatment effect rather than on the overall treatment effect, depending on the information available for that particular patient. The linear mixed-effects model of difference in question models the outcome variable as the difference of a distinct outcome variable taken between two treatments. So for this model the individual treatment effect can be found by taking the expectation of the outcome variable conditioned on a given patient. Given this definition, for the linear mixed-effects model of difference discussed in the current section, the individual treatment effects depend only on the ‘intercept’ term and on the “Patient” random term. The “Patient” random term is distinct for each patient and is the reason for the distinctiveness of the individual treatment effects. A linear model of difference with an ‘intercept’ term and without a “Patient” term results in the same individual treatment effect for each patient in the sample. In such a model, the individual treatment effect coincides with the overall treatment effect. When estimating individual treatment effects with the purpose of individualizing medicine from a linear model applied to a dataset where the outcome variable is already coded as a difference of an outcome variable between treatments, the inclusion in the model of a variable indicating the patient is therefore recommended. The following program demonstrates the determination of individual treatment effects using the linear mixed-effects model of difference. In this program, the balanced dataset “NORMAL.ddata” is used to fit the model.

Program 14.3: Estimating individual treatment effects using ‘proc mixed’ – Kenward and Roger’s inference.

```
title "Shrunk individual treatment effects";
title2 "Balanced data";
title3 "Kenward-Roger";
proc mixed
        data=NORMAL.ddata
        alpha=0.05 /* level for CI */
        cl /* CI for covariance parameters */
        method=REML; * REML method for estimation ;
    class Patient;
    model dY = / ddfm=KR;
    random Patient;
    estimate '1' intercept 1 |
        Patient 1 /
            alpha=0.05 /* level for CI */
            cl; * t-type CI for estimates ;
    estimate '2' intercept 1 |
        Patient 0 1;
    estimate '3' intercept 1 |
        Patient 0 0 1;
    estimate '4' intercept 1 |
        Patient 0 0 0 1;
    estimate '5' intercept 1 |
        Patient 0 0 0 0 1;
    estimate '6' intercept 1 |
        Patient 0 0 0 0 0 1;
    estimate '7' intercept 1 |
        Patient 0 0 0 0 0 0 1;
    estimate '8' intercept 1 |
        Patient 0 0 0 0 0 0 0 1;
    estimate '9' intercept 1 |
        Patient 0 0 0 0 0 0 0 0 1;
    estimate '10' intercept 1 |
        Patient 0 0 0 0 0 0 0 0 0 1;
    estimate '11' intercept 1 |
        Patient 0 0 0 0 0 0 0 0 0 0 1;
    estimate '12' intercept 1 |
        Patient 0 0 0 0 0 0 0 0 0 0 0 1;
run;
```

Program 14.3 opts for the essential options in the ‘proc mixed’ and ‘model’ statements. The program is similar to the programs presented in section 14.1 with the addition of several ‘estimate’ statements. There are twelve ‘estimate’ statements, one for each patient in the series of n-of-1 trials. The purpose of these ‘estimate’ statements is to request the estimation of the individual treatment effects for each patient. Each ‘estimate’ statement is coded in the following way. After the ‘estimate’ keyword, there is a label between single quotes. The label permits the association of the estimate to the patient, since there is one individual treatment effect for each patient. After the label follow the terms of the model to be added together with the definition of the coefficients of each term. For a linear model with both fixed and random effects there is a “|” character, separating the fixed part from the random part of the model. Since the intercept of the model is common to all the patients, a coefficient of 1 is defined after the ‘intercept’ keyword in all of the twelve ‘estimate’ statements. To define each individual treatment effect appropriately, the “Patient” random term is followed by a series of zeros in the position relative to the other patients, and there is a 1 in the position relative to the patient in question. Only the first ‘estimate’ statement contains the desired options, which are assumed by all the other ‘estimate’ statements. Program 14.3 is similar to Program 14.1 using the same dataset, so some of the tables obtained are equal between both programs. As such, Program 14.3 outputs Table 14.1, Table 14.2, Table 14.3, Table 14.4, Table 14.5, Table 14.6, Table 14.7, and an additional table resulting from the twelve ‘estimate’ statements. The additional table presented below, contains the individual treatment effects.

Table 14.19: Shrunk individual treatment effects for balanced data - Kenward and Roger’s inference.



Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
1	195.13	71.5475	1.03	2.73	0.2174	0.05	-649.79	1040.04
2	169.64	71.5475	1.03	2.37	0.2478	0.05	-675.27	1014.56
3	165.12	71.5475	1.03	2.31	0.2541	0.05	-679.80	1010.03
4	217.93	71.5475	1.03	3.05	0.1957	0.05	-626.99	1062.84
5	201.67	71.5475	1.03	2.82	0.2107	0.05	-643.25	1046.58
6	163.29	71.5475	1.03	2.28	0.2567	0.05	-681.63	1008.20
7	186.21	71.5475	1.03	2.60	0.2272	0.05	-658.71	1031.12
8	182.29	71.5475	1.03	2.55	0.2318	0.05	-662.62	1027.21
9	213.59	71.5475	1.03	2.99	0.1995	0.05	-631.33	1058.50
10	199.53	71.5475	1.03	2.79	0.2129	0.05	-645.38	1044.45
11	193.42	71.5475	1.03	2.70	0.2192	0.05	-651.50	1038.33
12	176.85	71.5475	1.03	2.47	0.2385	0.05	-668.06	1021.77

Note that the individual treatment effect statistics resulting from the linear mixed-effects model of difference displayed in Table 14.19 are the same as the ones resulting from the linear mixed-effects model discussed in section 13, and presented in Table 13.25! In fact, the linear mixed-effects model is equivalent to the linear mixed-effects model of difference. In addition, there is an equivalence between the individual treatment effect estimators arising from both models, which can be verified analytically. Since the statistics from Table

14.19 are equal to the ones from Table 13.25, what is discussed in respect to the latter table is valid for the former. Therefore, no further comments in respect to Table 14.19 are made. If you are starting your read of this document at this point, please refer to the text following Table 13.25 above for additional details.

The default method of estimation of the standard error of the fixed effects and degrees of freedom is used in the next program. The balanced dataset “NORMAL.ddata” is used to fit the linear mixed effects model of difference. Instead of using, several ‘estimate’ statements; the program uses a set of macros developed for estimating the individual treatment effects. Macro A.7, Macro A.8, and Macro A.9 are integrally presented in Appendix A at the bottom of the current document.

Program 14.4: Estimating individual treatment effects using ‘proc mixed’ and asymptotic inference – The macro way.

```
* Write global macro variable array ;
%estimate_diff_global(
        array=vdiff,
        subject=Patient,
        data=NORMAL.ddata
)

title "Shrunk individual treatment effects";
title2 "Balanced data";
proc mixed
                data=NORMAL.ddata
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Patient;
        model dY = ;
        random Patient;
        * insert estimate statements ;
        %estimate_diff(array=vdiff, options=alpha=0.05 cl)
run;

* Delete global macro variable array ;
%estimate_diff_free(array=vdiff)
```

Observe that the ‘ddfm’ option of the ‘model’ statement is not specified in Program 14.4. In this case, the SAS software assumes the default method of estimation of the standard error of the fixed effects and degrees of freedom. The method in question is conditioned on the covariance parameters of the model. The covariance parameters of the model are estimated and hence are random variables. As the number of patients and cycles increases towards infinity, the covariance parameters approach an exact value and their variance approaches zero. Therefore, this estimator of the variance of the fixed effects is asymptotically correct. For this reason, the estimator is called asymptotic estimator in this document. The SAS documentation refers to it as model based estimator [10]. For a small series of n-of-1 trials, the asymptotic estimator should underestimate the variance of the overall treatment effect and individual treatment effects. However, the asymptotic estimator of the variance of the estimates may be useful for a moderately sized series of n-of-1 clinical trials. Before running the ‘mixed’ procedure, the %estimate_diff_global macro is invoked. The %estimate_diff_global macro writes a set of global macro variables. All the macro variables created by %estimate_diff_global are prefixed by the word defined in the “array” parameter of this macro. When using macros with global macro variables there is the possibility that macro variables with the exact same names as the ones being in use in the main program are declared or reassigned inside the macro definition, leading to unexpected behaviour of the whole program. For this reason, it is considered a good programming practice to declare macro variables in local scope inside the actual macro definitions. Global macro variables shall be declared only in the main program. The %estimate_diff_global macro was developed to avoid this problem. The final user of the macros, the programmer who is writing the main program can define the prefix of the set of global macros at his own free will. Moreover, by avoiding the definition of other macro variables with the same prefix, conflicts can be prevented. The %estimate_diff_global macro variable accepts a “data” argument, which in this case is set as the “NORMAL.ddata” dataset. The “subject” parameter defines the categorical variable indicating the subject or patient in the series of n-of-1 trials. In this case, the “Patient” variable is supplied to the “subject” parameter. The variable supplied to the “subject” parameter must obviously be present in the dataset provided to the “data” parameter of the %estimate_diff_global macro. After writing the set of global macro variables, which can also be considered an array of macro variables, the “vdiff” macro variable array is available for posterior use by the %estimate_diff macro. Briefly, what %estimate_diff_global does is to look into the dataset provided, determine the labels for every patient, determine the vectors containing the zeros and ones in the right place, such as the vectors in the ‘estimate’ statements of Program 14.3, and save this information to a global macro variable array. The %estimate_diff macro is placed where the ‘estimate’ statements of the ‘mixed’ procedure are usually placed. When the program is submitted for interpretation and the %estimate_diff macro is expanded, all the possible ‘estimate’ statements are determined from the “vdiff” macro variable array, and are inserted under the ‘mixed’ procedure. The %estimate_diff macro contains a “random” parameter, which by default is set to the character used to separate the fixed part from the random part in the ‘estimate’ statements of Program 14.3. The character in question is the vertical bar (|). The “random” parameter was coded for compatibility with the linear fixed-effects model of difference described in section 12. The default character is correct for linear mixed-effects models, so there is no need to write the “random” parameter of the %estimate_diff macro when invoked under the ‘mixed’ procedure. The macro variables from the “vdiff” array are available during the entire SAS session, unless they are deleted. The %estimate_diff_free macro can be used to delete all the macro variables previously created by the %estimate_diff_global macro. In the bottom of Program 14.4, the “vdiff” macro variable array is supplied to the “array” parameter of the %estimate_diff_free macro, and when the program is executed, the macro call is expanded with appropriate statements to perform the deletion of the variables prefixed by the “vdiff” word. This way the global macro variables can be deleted when no longer needed, providing yet another strategy to avoid possible conflicts while running the main program. A single macro could have been developed to complete the tasks performed by the three different macros invoked in Program 14.4, without resorting to global macro variables. The solution provided in three distinct macro calls is preferred because it can be computationally more efficient under certain situations. Suppose that you need to use the same macro variable array across several ‘mixed’ procedure calls. With the solution designed in three distinct macro invocations, the computer only needs to execute once the determination of the correct position of the zeros and ones for each patient. While with a single macro alternative, the computer would have to perform such task as many times as many ‘mixed’ procedures are requested. In such situation, the three-macro solution provided can run faster at the expense of some additional computer memory. The programmer only needs to invoke the %estimate_diff_global macro once; use the same macro variable array as many times as needed; and in the end invoke the %estimate_diff_free macro to remove the macro variable array from computer memory. Some of the tables outputted from Program 14.4 are the same as the ones outputted from Program 14.1. As such, Program 14.4 outputs Table 14.2, Table 14.3, Table 14.4, Table 14.5, Table 14.6, and Table 14.7. The model information table presented below contains distinct information. A new table holding the individual treatment effects is produced.

Table 14.20: Model information.

Model Information	
Data Set	NORMAL.DDATA
Dependent Variable	dY
Covariance Structure	Variance Components
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Table 14.21: Shrunk individual treatment effects for balanced data - Asymptotic inference.

Estimates								
Label	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
1	195.13	44.5523	11	4.38	0.0011	0.05	97.0706	293.19
2	169.64	44.5523	11	3.81	0.0029	0.05	71.5834	267.70
3	165.12	44.5523	11	3.71	0.0035	0.05	67.0605	263.18
4	217.93	44.5523	11	4.89	0.0005	0.05	119.87	315.99
5	201.67	44.5523	11	4.53	0.0009	0.05	103.61	299.73
6	163.29	44.5523	11	3.67	0.0037	0.05	65.2269	261.35
7	186.21	44.5523	11	4.18	0.0015	0.05	88.1470	284.27
8	182.29	44.5523	11	4.09	0.0018	0.05	84.2353	280.35
9	213.59	44.5523	11	4.79	0.0006	0.05	115.53	311.65
10	199.53	44.5523	11	4.48	0.0009	0.05	101.47	297.59
11	193.42	44.5523	11	4.34	0.0012	0.05	95.3592	291.48
12	176.85	44.5523	11	3.97	0.0022	0.05	78.7956	274.91

The ‘ddfm’ option is not specified in the ‘model’ statement of Program 14.4, in this way the model-based method is used to estimate the standard error of the fixed effects, and the containment method is used to compute the degrees of freedom. Table 14.20 provides this information. The individual treatment effect statistics resulting from the linear mixed-effects model of difference displayed in Table 14.21are the same as the ones resulting from the linear mixed-effects model discussed in section 13, and presented in Table 13.27! The linear mixed-effects model of difference exemplified in this section is equivalent to the linear mixed-effects model demonstrated in section 13. What is written in respect to Table 13.27 is valid for Table 14.21, since the two tables hold the same values for each individual and statistic. Therefore, please refer to the text following Table 13.27 for additional details over the interpretation of the results displayed in Table 14.21.

The following program fits a linear mixed-effects model of difference to the unbalanced dataset of outcome differences previously obtained by running either Program 4.6 or Program 5.7. The set of macros developed to assist the estimation of individual treatment effects introduced in Program 14.4 is used again. The results are presented below. Then a comparison with the results obtained from the methods discussed above is made.

Program 14.5: Estimating individual treatment effects using ‘proc mixed’ and Kenward and Roger’s inference – The macro way.

```
* Write global macro variable array ;
%estimate_diff_global(
        array=vdiff_unbalanced,
        subject=Patient,
        data=NORMAL.ddata_unbalanced
)

title "Shrunk individual treatment effects";
title2 "Unbalanced data";
title3 "Kenward-Roger";
proc mixed
                data=NORMAL.ddata_unbalanced
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Patient;
        model dY = / ddfm=KR;
        random Patient;
        * insert estimate statements ;
        %estimate_diff(array=vdiff_unbalanced, options=alpha=0.05 cl)
run;

* Delete global macro variable array ;
%estimate_diff_free(array=vdiff_unbalanced)
```

Program 14.5 is very similar to Program 14.4 with minor differences. The differences are highlighted next. The “NORMAL.ddata_unbalanced” dataset is used in Program 14.5. The dataset is unbalanced in the sense that some n-of-1 trials contain data for 3 cycles; others contain data for a different number of cycles. The array of global macro variables is named differently, being “vdiff” for Program 14.4 and “vdiff_unbalanced” for Program 14.5. This array holds data associated to the dataset referenced in the %estimate_diff_global macro and the ‘mixed’ procedure. To avoid possible conflicts and to make the code easier to read and understand, it is recommended to use different global macro array names across different datasets. The ‘ddfm’ option of the ‘model’ statement is absent from Program 14.4, but present in Program 14.5. The ‘ddfm’ option requests the Kenward and Roger’s method of estimation of the standard error of the fixed effects and degrees of freedom [25]. The macros %estimate_diff_global, %estimate_diff and %estimate_diff_free are used in both programs. For a description of these three macros, please refer to the text following Program 14.4. The output resulting from Program 14.5 is similar to the output resulting from Program 14.2. As such, Program 14.5 outputs Table 14.10, Table 14.11, Table 14.12, Table 14.13, Table 14.14, Table 14.15, Table 14.16 and the table presented below, which holds the shrunk individual treatment effects.

Table 14.22: Shrunk individual treatment effects for unbalanced data - Kenward and Roger's inference.

Estimates								
Label	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
1	200.78	73.9361	1.45	2.72	0.1575	0.05	-265.14	666.69
2	170.90	73.9361	1.45	2.31	0.1928	0.05	-295.02	636.81
3	165.60	73.9361	1.45	2.24	0.2003	0.05	-300.32	631.51
4	227.50	73.9361	1.45	3.08	0.1340	0.05	-238.41	693.42
5	208.44	73.9361	1.45	2.82	0.1501	0.05	-257.47	674.36
6	163.45	73.9361	1.45	2.21	0.2035	0.05	-302.47	629.36
7	190.32	73.9361	1.45	2.57	0.1686	0.05	-275.60	656.23
8	185.73	73.9361	1.45	2.51	0.1739	0.05	-280.18	651.65
9	222.41	73.9361	1.45	3.01	0.1380	0.05	-243.50	688.33
10	205.93	73.9361	1.45	2.79	0.1525	0.05	-259.98	671.85
11	203.77	73.3900	1.14	2.78	0.1943	0.05	-494.14	901.68
12	189.28	69.1514	1	2.74	0.2230	0.05	-689.37	1067.93

The individual treatment effect estimator arising from the linear mixed-effects model of difference is equivalent to the homologue estimator resulting from the mixed-effects model exemplified in section 13. As expected, the results displayed in Table 14.22 are equal to the ones from Table 13.30. Therefore, to avoid overly repetition of text no additional comments are made in respect to the results presented in Table 14.22. Please refer to the text following Table 13.30 for additional details.

For the estimation of individual treatment effects through the mixed-effects model of difference, the SAS system provides a very accessible alternative to the 'estimate' statements exemplified in Program 14.3, and the macros demonstrated in Program 14.4 and Program 14.5. This alternative is demonstrated in the program below.

Program 14.6: Estimating individual treatment effects using ‘proc mixed’ - Predicted values.

```
title "Shrunk individual treatment effects";
title2 "Unbalanced data";
proc mixed
                data=NORMAL.ddata_unbalanced
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Patient;
        model dY = /
                outp=WORK.pddata(drop=Resid); * output dataset for predicted values;
        id Patient; * variables from input dataset in outp= dataset;
        random Patient;
run;

title "Shrunk individual treatment effect estimates";
proc sql;
        select  distinct *
        from WORK.pddata;
        drop table WORK.pddata;
quit;
```

For the linear mixed-effects model of difference considered in the current section, the individual treatment effects are equal to the predicted values. The predicted values equal the fixed effects plus the random effects. In this case, the random effects are equal to the patient random effects; therefore, as many distinct predicted values as many individuals available in the dataset can be estimated. For the linear mixed-effects model discussed in section 13, the estimation of the individual treatment effects is not as simple! For that particular case, the individual treatment effects can indeed be estimated from the predicted values; however, some algebra is required. The unbalanced dataset previously obtained from either Program 4.6 or Program 5.7 is used in the ‘mixed’ procedure from Program 14.6. Option ‘outp’ in the ‘model’ statement permits the definition of a dataset to store the predicted values. In this case, the new dataset “pddata” is directed towards the “WORK” library. Remember that the “WORK” library resides in the computer memory, consequently being lost between machine reboots and shutdowns. There is no intention in keeping the “pddata” dataset; the interest lies in displaying the dataset to inspect the predicted values. Note that a “Resid” variable is dropped from the “WORK.pddata” dataset. The purpose in dropping this variable is to make the predicted values table look identical to the previous tables holding the individual treatment effects. The “Resid” variable holds the residual errors, and is not relevant for this particular situation. The ‘id’ statement identifies the variables from the input dataset “NORMAL.ddata_unbalanced” to be included in the predicted values output dataset “WORK.pddata”. There is no interest in including variables other than the “Patient” variable. One needs to know which predicted value belongs to which individual. If the “Patient” variable is left out of the ‘id’ statement, the output table holds some statistics pertaining to the individual treatment effects, but the analyst cannot identify the individuals! There is no use for such table. After running the ‘mixed’ procedure from Program 14.6, the statistics are computed, and the “pddata” dataset is found in the “WORK” library. The next step consists in displaying this dataset to inspect the individual treatment effect statistics. For this purpose, the ‘sql’ procedure can be used with very few statements. The ‘sql’ procedure from Program 14.6 contains two very simple statements. The ‘select’ statement selects all the variables from the dataset “WORK.pddata” to be displayed. There is no intension in keeping the “WORK.pddata” dataset, so the dataset is removed from the system through a ‘drop table’ statement. Note the usage of a ‘distinct’ keyword before the asterisk character. The asterisk character represents all the variables contained in the dataset. The reason for using a ‘distinct’ keyword is explained shortly. The input dataset “NORMAL.ddata_unbalanced” contains several rows per patient. Because the “Patient” variable is specified in the ‘id’ statement, the output “WORK.pddata” dataset contains the same number of rows per patient as the input dataset “NORMAL.ddata_unbalanced”. Since the individual treatment effects are unique for an individual, the output “WORK.pddata” contains several repeated rows. One is not interested in repeated information taking extra space in the output and possibly in publications. Furthermore, repeated information can be confusing. The ‘distinct’ keyword instructs the ‘sql’ procedure to select distinct rows from the dataset. The output from the ‘mixed’ procedure in Program 14.6 is similar to the output from Program 14.2 presented above. The ‘mixed’ procedure from Program 14.6 outputs Table 14.10, Table 14.11, Table 14.12, Table 14.13, Table 14.14, Table 14.15 and Table 14.16. The ‘sql’ procedure from the same program outputs the table presented below.

Table 14.23: Shrunk individual treatment effects for unbalanced data and asymptotic inference – Predicted values.

Patient	Predicted	Std Err Pred	DF	Alpha	Lower	Upper
1	200.7764	47.9005	21	0.05	101.1618	300.3909
2	170.8996	47.9005	21	0.05	71.28501	270.5141
3	165.5977	47.9005	21	0.05	65.98313	265.2122
4	227.5007	47.9005	21	0.05	127.8861	327.1152
5	208.4426	47.9005	21	0.05	108.828	308.0571
6	163.4483	47.9005	21	0.05	63.83372	263.0628
7	190.3159	47.9005	21	0.05	90.70134	289.9304
8	185.7305	47.9005	21	0.05	86.11594	285.345
9	222.4137	47.9005	21	0.05	122.7992	322.0283
10	205.9349	47.9005	21	0.05	106.3204	305.5495
11	203.769	50.18703	21	0.05	99.39933	308.1386
12	189.2815	52.81181	21	0.05	79.45336	299.1097

The estimator of the individual treatment effects resulting from the linear mixed-effects model of difference discussed in the current section is equivalent to the homologue estimator resulting from linear mixed-effects model discussed in section 13. Consequently, some of the statistics in Table 14.23 contain identical results to the ones from *Table* 13.33. The notable difference lies in the value for the degrees of freedom, being equal to 11 for the later table, and equal to 21 for the former table. Note that *Table* 13.33 contains a column for the t statistic and another column for the p-value, while Table 14.23 does not. The individual treatment effect estimates from Table 14.23 are not exactly equal to the ones from *Table* 13.33, the differences being minimal. The values can be considered equal up to one or two decimal places. The estimates of the individual treatment effects presented on both tables should be equal. The differences observed are most likely attributed to minor differences in the numerical computation. Likewise, the estimates of the standard error of the predictions displayed in Table 14.23 are not exactly equal to the ones displayed in *Table* 13.33, but very close. Again, the small differences observed may be attributed to differences in the numerical computation used to obtain the estimates on each table. The confidence intervals of the estimates are narrower on Table 14.23 than on *Table* 13.33. This trend is observable for each of the twelve patients. Being the significance level and standard error of the estimates identical between both tables, the differences observed in the confidence intervals can be explained in terms of the degrees of freedom. For the same standard error of an estimate, and for an equal significance level, a higher number of the degrees of freedom generally leads to narrower confidence intervals for that same estimate. However, there is something wrong with the results from either Table 14.23 or *Table* 13.33! The predicted values from Table 14.23 are equivalent to the estimates from *Table* 13.33, so for each individual these estimates are the same realization of the same random variable, and as such follow the same distribution. Since the statistics follow the same distribution, for the same significance level, the lower and upper limits of the confidence interval must be the same. Nonetheless, for a given individual the lower and upper limits of the confidence interval of the estimate on Table 14.23 are quite different from the ones on *Table* 13.33! Still the conclusions do not change from *Table* 13.33 to Table 14.23. Since the origin does not lie within the confidence interval for any of the twelve estimates, there is statistical evidence that the treatments differ significantly individually for each of the twelve patients.

An alternative way of coding the linear mixed-effects model of difference discussed in the current section is demonstrated below. The code leading to the individual treatments effects is exemplified. The “NORMAL.ddata” dataset previously obtained from either Program 4.5 or Program 5.6 is used for the demonstration. Attention is given to the differences in respect to programs presented above in this document.

Program 14.7: Estimating individual treatment effects using 'proc mixed' - The alternative way.

```
title "Shrunk individual treatment effects";
title2 "Balanced data";
proc mixed
                data=NORMAL.ddata
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Patient;
        model dY = /
                alpha=0.05 /* level for CI */
                cl /* t-type CI for fixed effects */
                solution; * solution for fixed effects ;
        random intercept /
                subject=Patient;
        estimate '1' intercept 1 | intercept 1 /
                subject 1
                alpha=0.05 /* level for CI */
                cl; * t-type CI for estimates ;
        estimate '2' intercept 1 | intercept  1 /
                subject 0 1;
        estimate '3' intercept 1 | intercept  1 /
                subject 0 0 1;
        estimate '4' intercept 1 | intercept  1 /
                subject 0 0 0 1;
        estimate '5' intercept 1 | intercept  1 /
                subject 0 0 0 0 1;
        estimate '6' intercept 1 | intercept  1 /
                subject 0 0 0 0 0 1;
        estimate '7' intercept 1 | intercept  1 /
                subject 0 0 0 0 0 0 1;
        estimate '8' intercept 1 | intercept  1 /
                subject 0 0 0 0 0 0 0 1;
        estimate '9' intercept 1 | intercept  1 /
                subject 0 0 0 0 0 0 0 0 1;
        estimate '10' intercept 1 | intercept  1 /
                subject 0 0 0 0 0 0 0 0 0 1;
        estimate '11' intercept 1 | intercept  1 /
                subject 0 0 0 0 0 0 0 0 0 0 1;
        estimate '12' intercept 1 | intercept  1 /
                subject 0 0 0 0 0 0 0 0 0 0 0 1;
run;
```

Program 14.7 is similar to Program 14.3 fitting the same linear mixed-effects model to the same dataset. The differences are located mainly on the ‘random’ statement and on the ‘estimate’ statements. Unlike in Program 14.3, the “Patient” variable is absent from the terms of the ‘random’ statement. Instead, the ‘subject’ option of the same ‘random’ statement is set to the “Patient” variable. Then the ‘intercept’ keyword is defined as a term of the ‘random’ statement. The “Patient” variable in the ‘subject’ option indicates that the random effect terms are nested within the patient. As such for each patient there is an ‘intercept’ term. This is an alternative way of specifying a patient random effect. The twelve ‘estimate’ statements from Program 14.7 request the individual treatment effects. Note however that the structure of the ‘estimate’ statements from Program 14.7 is somewhat distinct from the one used on the ‘estimate’ statements from Program 14.3. The label and the coefficient of the fixed-effect term are identical between both programs. As usual, there is a vertical bar separating the fixed effect coefficients from the random effect coefficients. Since only the ‘intercept’ random effect term within the subject contributes to the individual treatment effects, only the coefficient of the ‘intercept’ random effect is specified. Here the coefficient is the same as in the fixed effects part of the same ‘estimate’ statement. Each ‘estimate’ statement contains a ‘subject’ option followed by a set of zeros and a single one. Here the coefficients are associated to the levels of the “Patient” variable, which is supplied to the ‘subject’ option of the ‘random’ statement. Therefore, there is a one in the position associated to the patient defining the desired individual treatment effect and a zero in each of the positions associated to the other patients. Note that the order of the patients in the ‘estimate’ statements is the same order the patients occupy in the dataset. The execution of Program 14.7 leads to the tables presented below.

Table 14.24: Model information.

Model Information	
Data Set	NORMAL.DDATA
Dependent Variable	dY
Covariance Structure	Variance Components
Subject Effect	Patient
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Table 14.25: Class level information.

Class Level Information		
Class	**Levels**	**Values**
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12

Table 14.26: Dimensions.

Dimensions	
Covariance Parameters	2
Columns in X	1
Columns in Z per Subject	1
Subjects	12
Max Obs per Subject	3

Table 14.27: Number of observations.

Number of Observations	
Number of Observations Read	36
Number of Observations Used	36
Number of Observations Not Used	0

Table 14.28: Iteration history.

Iteration History			
Iteration	**Evaluations**	**-2 Res Log Like**	**Criterion**
0	1	457.83663221	
1	1	457.67817788	0.00000000

Table 14.29: Covariance parameter estimates.

Covariance Parameter Estimates					
Cov Parm	**Subject**	**Estimate**	**Alpha**	**Lower**	**Upper**
Intercept	**Patient**	1772.67	0.05	204.44	8.22E13
Residual		23685	0.05	14441	45838

Table 14.30: Fit statistics.

Fit Statistics	
-2 Res Log Likelihood	457.7
AIC (Smaller is Better)	461.7
AICC (Smaller is Better)	462.1
BIC (Smaller is Better)	462.6

Table 14.31: Solution for fixed effects.

Solution for Fixed Effects								
Effect	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Intercept	188.72	28.3838	11	6.65	<.0001	0.05	126.25	251.19

Table 14.32: Shrunk individual treatment effects.

Estimates								
Label	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
1	195.13	44.5523	11	4.38	0.0011	0.05	97.0706	293.19
2	169.64	44.5523	11	3.81	0.0029	0.05	71.5834	267.70
3	165.12	44.5523	11	3.71	0.0035	0.05	67.0605	263.18
4	217.93	44.5523	11	4.89	0.0005	0.05	119.87	315.99
5	201.67	44.5523	11	4.53	0.0009	0.05	103.61	299.73
6	163.29	44.5523	11	3.67	0.0037	0.05	65.2269	261.35
7	186.21	44.5523	11	4.18	0.0015	0.05	88.1470	284.27
8	182.29	44.5523	11	4.09	0.0018	0.05	84.2353	280.35
9	213.59	44.5523	11	4.79	0.0006	0.05	115.53	311.65
10	199.53	44.5523	11	4.48	0.0009	0.05	101.47	297.59
11	193.42	44.5523	11	4.34	0.0012	0.05	95.3592	291.48
12	176.85	44.5523	11	3.97	0.0022	0.05	78.7956	274.91

The nine tables obtained from Program 14.7 contain results that are presented above in the present document. The results in question are obtained from both Program 14.1 and Program 14.4. The differences lie mainly in the presentation of the results within the tables. Observe that there is a row indicating the subject effect on *Table* 14.24; while such a row is absent from previous model information tables. *Table* 14.26 is quite different from previous dimension tables. Twelve subjects are modelled according to *Table* 14.26. The covariance parameter estimates and related statistics are contained within *Table* 14.29, where there is a column identified as "Subject". On the "Subject" column, the "Patient" variable is presented on every row associated to a covariance parameter that is nested on that same "Patient" variable. For the model in question, only the 'intercept' is nested on the "Patient" variable. The covariance parameters and related statistics are the same as the ones displayed in Table 14.6 above. All the other results are the same as the ones presented above. Most notably the table pertaining to the solution for fixed effects contains the same values as Table 14.8 presented above. Moreover, the shrunk individual treatment effects and respective statistics are the same as the ones presented above on Table 14.21.

Requesting individual treatment effect estimates through the ‘mixed’ procedure requires writing an ‘estimate’ statement for each individual as desired. One may wish to obtain all the individual treatment effect estimates that can be obtained from a given dataset arising from series of n-of-1 trials. A series of n-of-1 trials may be composed of a high number of single n-of-1 trials. In this case writing an ‘estimate’ statement for each individual may seem a lot of work. A set of three macros was developed to lessen the effort of requesting individual treatment effect estimates under the ‘mixed’ procedure. The %estimate_diff_global, %estimate_diff, and %estimate_diff_free macros were developed with the ‘estimate’ statements used in Program 14.3 in mind. When the linear mixed effects model of difference is requested through the specification of the ‘subject’ option of the ‘random’ statement, the ‘estimate’ statements leading to the individual treatment effect estimates must be specified accordingly, and the referred set of three macros is inadequate. A new set of three macros inspired by the ‘estimate’ statements used in Program 14.7 was developed. The macros in question are Macro A.10, Macro A.11, and Macro A.12 integrally presented on Appendix A at the bottom of this document. The usage of these three macros is exemplified in the following program.

Program 14.8: Estimating individual treatment effects using 'proc mixed' - The alternative macro way.

```
* Write global macro variable array ;
%estimate_diff_random_global(
        array=vdiff,
        subject=Patient,
        data=NORMAL.ddata
)

title "Shrunk individual treatment effects";
title2 "Balanced data";
proc mixed
                data=NORMAL.ddata
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Patient;
        model dY = /
                alpha=0.05 /* level for CI */
                cl /* t-type CI for fixed effects */
                solution; * solution for fixed effects ;
        random intercept /
                subject=Patient;
        * insert estimate statements ;
        %estimate_diff_random(array=vdiff, options=alpha=0.05 cl)
run;

* Delete global macro variable array ;
%estimate_diff_random_free(array=vdiff)
```

The ‘random’ statement is the same as the one used in Program 14.7, with the specification of the “Patient” variable in the ‘subject’ option. In this case, three macros are used to automatize the insertion of the ‘estimate’ statements leading to the individual treatment effect estimates. The %estimate_diff_random_global macro is invoked first. This macro reads the necessary information from the “NORMAL.ddata” dataset, and writes the “vdiff” macro variable array to the system. This macro variable array is in fact a set of macro variables where each macro variable name is prefixed by the word “vdiff”. The variables associated to the macro variable array are declared in global scope, remaining in the computer’s memory until removed, or the SAS system is terminated. When the program is executed, the %estimate_diff_random macro is expanded, information is read from the “vdiff” macro variable array and the ‘estimate’ statements for all the individual treatment effects that can be estimated from the dataset are inserted under the ‘mixed’ procedure. Finally, the %estimate_diff_random_free macro removes the “vdiff” macro variable array from the system. Program 14.8 outputs the same nine tables as Program 14.7. The results are presented and discussed above, for that reason no further discussions are presented.

15 Modelling heterogeneous variances

The linear models considered in the above sections assume that the variance of the residual error is equal among the groups, be it the patients, the cycles or the treatments. These models are frequently referred as homoscedastic or homogeneous variance models. Per opposition, models that consider distinct residual variances are called heteroscedastic or heterogeneous variance models. The purpose of this section is to demonstrate the process of fitting linear mixed-effect models with different residual variances per group by using the SAS system. Fitting distinct residual variances per patient may not be practical with linear mixed-effects models applied to series of n-of-1 trials data. If you try such an approach, you may find that the estimation algorithm does not converge to a solution. In that case, there is a large number of parameters and the number of degrees of freedom is significantly reduced. Your dataset may not have enough observations to estimate a model with such a large number of parameters. If you try fitting linear mixed-effects models with different residual variances per patient on any of the datasets introduced in either section 4 or section 5, you will find that the estimation algorithm does not converge to a meaningful solution. However, different residual variances per treatment, per cycle, or both simultaneously, can be fitted on the "NORMAL.ndata" dataset introduced by either Program 4.1 or Program 5.1, or on the "NORMAL.ndata_unbalanced" dataset resulting from either Program 4.4 or Program 5.5. On the datasets of outcome difference "NORMAL.ddata" obtained from either Program 4.5 or Program 5.6; or the respective unbalanced dataset "NORMAL.ddata_unbalanced" created by either Program 4.6 or Program 5.7, different residual variances per cycle can be fitted. Note that the "Treatment" variable is not available in the datasets of outcome difference. The "NORMAL.ndata" dataset is used in the current section for demonstration purposes.

You may choose one of two approaches. You may start by fitting a heterogeneous variance linear mixed-effects model, which due to the complexity inherent to the larger number of parameters may not be the best choice for your data. Your data may fit well with a simpler homogeneous variance linear mixed-effects model. You may go for the second approach, where you start by fitting a homogeneous variance linear mixed-effects model and then you use one of several techniques available to validate the homoscedastic assumption. If you find that the homoscedastic assumption is not reasonable for your data, you then opt for a heteroscedastic linear mixed-effects model; otherwise, you opt for the homoscedastic linear mixed-effects model. The goal is to choose the simpler model that fits your data reasonably well. This second approach is about to be demonstrated through a simple and very intuitive graphical method.

Program 15.1: Fitting a linear mixed-effects model using 'proc mixed' and saving influence statistics.

```
ods exclude all;
proc mixed data=NORMAL.ndata
                method=REML;
        class Treatment (ref=first) Patient Cycle;
        model Y = Treatment /
                influence;
        random Patient Patient*Cycle Patient*Treatment;
        ods output influence=WORK.influence;
run;
ods exclude none;
```

The linear mixed-effects model introduced in section 13 is fitted in Program 15.1. The purpose of this program is to obtain a dataset with influence statistics. There is no intention in obtaining anything other than the influence statistics dataset. The ‘ods exclude all’ statement in the top of Program 15.1 instructs SAS to suppress all the output resulting from the statements written afterwards. All the output is supressed until an ‘ods exclude none’ statement is encountered. Since in Program 15.1 the ‘mixed’ procedure is enclosed between an ‘ods exclude all’ statement and an ‘ods exclude none’ statement, nothing is outputted from the procedure. The ‘influence’ option of the ‘model’ statement requests influence and case-deletion diagnostics. A “WORK.influence” dataset is defined to hold the influence data resulting from the model fitting process according to the ‘ods output’ statement in Program 15.1. After running Program 15.1, the “influence” dataset becomes available in the “WORK” library. There is no other output from the program. The “WORK.influence” dataset contains several variables, but only the variable holding the residual errors is relevant for this demonstration. In addition, the “WORK.influence” dataset does not contain any variable from the input dataset, in this case “NORMAL.ndata”. To validate the homoscedasticity of the linear mixed-effects model, one needs the residual errors and the grouping variables from the input dataset, such as the “Treatment” and “Cycle” variables, preferably in the same dataset. The next program was designed to achieve just this.

Program 15.2: Obtaining a dataset with the residual errors and grouping variables.

```
data WORK.ndata;
        set NORMAL.ndata;
        index=_n_;
run;

data WORK.residual;
        merge
                WORK.ndata
                WORK.influence;
        by index;
        keep Treatment Cycle Residual;
run;

proc datasets lib=WORK nolist;
        delete influence ndata;
quit;
```

Two data steps are contained in Program 15.2. The first data step creates an “ndata” dataset in the “WORK” library, by taking the “NORMAL.ndata” dataset and creating a new variable named “index” to hold the values of the ‘_n_’ automatic variable. The purpose of this data step is to index the observations in the dataset “NORMAL.ndata” inputted to the ‘mixed’ procedure in Program 15.1. Since the “WORK.influence” dataset already contains an “index” variable, the two datasets “WORK.ndata” and “WORK.influence” can be merged, and the residuals associated to the grouping variables in the input dataset “NORMAL.ndata”. This is exactly what the second data step in Program 15.2 does. The second data step in Program 15.2 creates a dataset “residual” in the “WORK” library, by merging the “ndata” and “influence” datasets found in the same library. The merge is defined by the “index” variable present in both “WORK.ndata” and “WORK.influence” datasets. The resulting dataset “WORK.residual” has the rows of those two datasets side by side. Only the variables “Residual”, “Cycle”, and “Treatment” are pertinent to the analysis, so these variables are written in the ‘keep’ statement, and remain in the “WORK.residual” dataset, while all the other variables are discarded. Finally, the “influence” dataset obtained from Program 15.1, and the “ndata” dataset obtained from the first data step in Program 15.2, are deleted from the “WORK” library through the “datasets” procedure. Alternatives to Program 15.1 and Program 15.2 are possible with the SAS software. For example, a program inspired by Program 14.6 can be designed, where the residuals and the grouping variables are obtained in the predicted values dataset. There is a dataset with the residual errors associated to the treatment; therefore, the residual errors can be plotted against the levels of the “Treatment” categorical variable. The variance of the residual errors under treatment “A” can be visually compared with the variance of the residuals under treatment “B”. The following program can be used to obtain a plot, which permits such a comparison to be made.

Program 15.3: Creating boxplots of residual error for each treatment.

```
* Set graph size in pixels ;
ods graphics on /
        border=off /* suppress graphics border */
        width=720px
        height=720px;

proc sgpanel
                data=WORK.residual
                description="Boxplot of residual error vs treatment"
                noautolegend;
        panelby Treatment;
        vbox Residual /
                group=Treatment
                nofill
                percentile=1;
        rowaxis label="Residual error";
run;

ods graphics off;

proc datasets lib=WORK nolist;
        delete residual;
quit;
```

To obtain a plot that looks as desired, an ‘ods graphics on’ statement is used in the top of Program 15.3. The ‘border=off’ option is used to suppress the plot’s border, while the ‘width’ and ‘height’ options define the size of the plot in pixels. The ‘sgpanel’ procedure is used with some statements defining the plot. The dataset “WORK.residual” contains the variables desired for the plot, so it is supplied to the ‘data’ option of the ‘proc sgpanel’ statement. A legend is not needed for the plot’s interpretation, hence the use of the ‘noautolegend’ option in the ‘proc sgpanel’ statement. The ‘panelby’ statement can be used with a categorical variable to group the data by this variable. In this case, the “Treatment” variable is used, so the data is split between the levels of this categorical variable. In this case two panels are drawn, one with a plot of the data associated to treatment “A”, and the other with a plot of the data associated to treatment “B”. The ‘sgpanel’ procedure supports several statements according to the type of plot desired. For boxplots, the ‘vbox’ statement is available for vertical boxplots, and the ‘hbox’ statement can be used to draw horizontal boxplots. The ‘vbox’ statement is used in Program 15.3. In this statement, the desired variable “Residual” is written immediately after the ‘vbox’ keyword. The ‘group’ option defines the grouping variable. As such, in Program 15.3, the values of the “Residual” variable are plotted separately for the levels of the “Treatment” categorical variable. After drawing the plot, Program 15.3 disables graphics output through a ‘ods graphics off’ statement. Finally, the ‘datasets’ procedure permanently removes the “residual” dataset from the “WORK” library. The result of Program 15.3 is two boxplots of the residual errors resulting from the homoscedastic linear mixed-effects model fitted in Program 15.1, one under treatment "A", and the other under treatment "B". The plot is presented below.

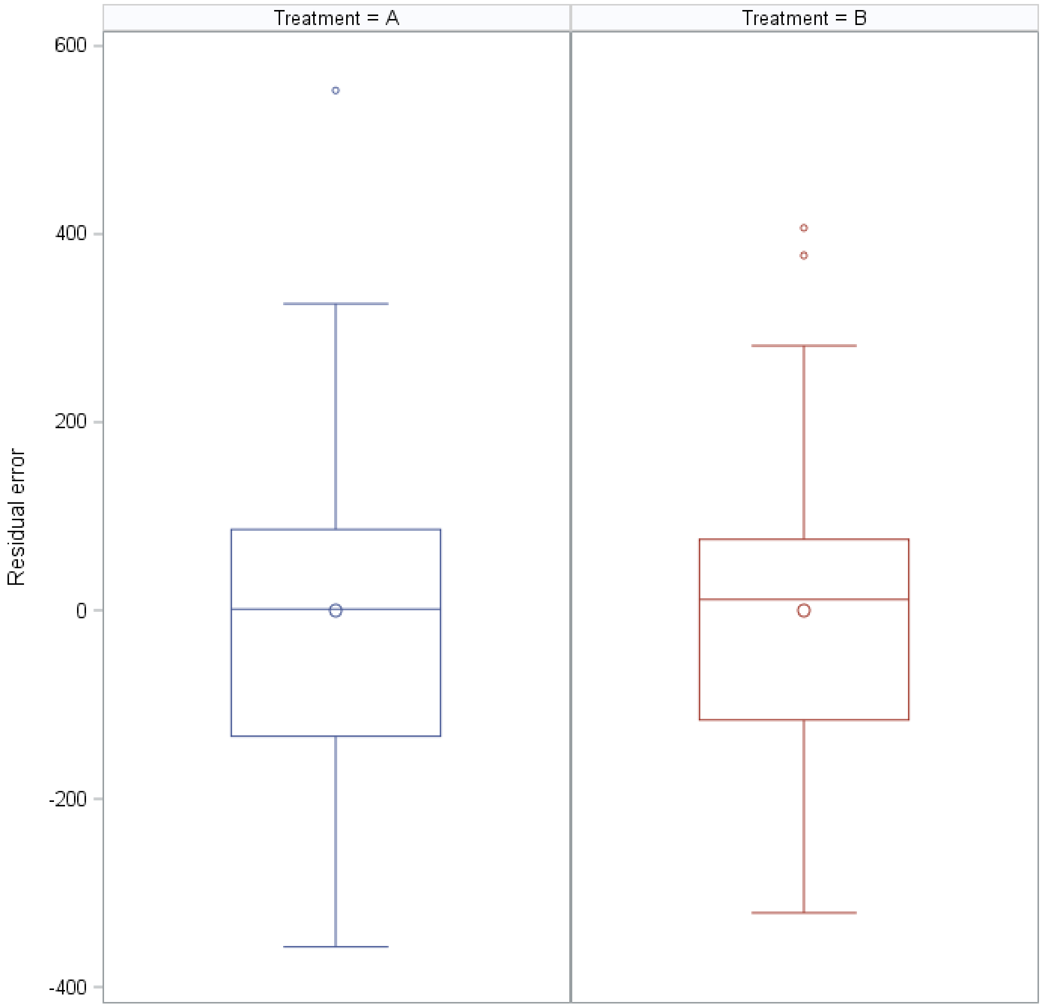


Figure 15.1: Boxplots of residual error for each treatment.

Introduced by Tukey [28], boxplots are graphical representations where the quartiles determine the limits of a box, and the centre line can be determined by either the mean or the median. Therefore, the size of the box is an indicative of how big or small the variance of the data is. As such, Figure 15.1 suggests a very small difference between the variance of the residual errors under the two treatments. A difference such small does not constitute justification to fit a linear model with distinct variances per treatment. Despite these results, for demonstrations purposes, a linear mixed-effects model with distinct variances per treatment is fitted to the "NORMAL.ndata" dataset in a program below. There are several graphical alternatives to the homogeneous variance validation method demonstrated here. Alternatively, statistical hypothesis tests can be used to test the equality of variances among two or more groups, such as the Bartlett's test or the Brown-Forsythe's test [29, 30]. Due to space and time constraints, not all the available homogeneous variance validation methods can be demonstrated on this text.

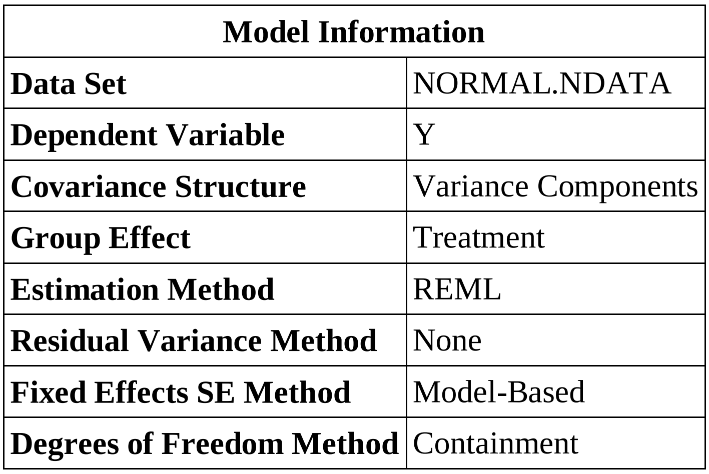
The next program fits a linear mixed-effects model with different residual variances per treatment to the "NORMAL.ndata" dataset previously obtained through either Program 4.1 or Program 5.1.

Program 15.4: Fitting a heteroscedastic linear mixed-effects model using 'proc mixed'.

```
* Write global macro variable array ;
%estimate_full_global(
        array=vfull_balanced,
        subject=Patient,
        treatment=Treatment,
        data=NORMAL.ndata
)

title "Heterocedastic linear mixed-effects model";
proc mixed
                data=NORMAL.ndata
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Treatment (ref=first) Patient Cycle;
        model Y = Treatment /
                alpha=0.05 /* level for CI */
                cl /* t-type CI for fixed effects */
                covb /* covariance matrix of fixed effects estimates */
                htype=2 /* type 2 hypothesis test */
                solution; * solution for fixed effects ;
        random Patient Patient*Cycle Patient*Treatment;
        repeated /
                group=Treatment
                type=vc;
        %estimate_full(array=vfull_balanced, options=alpha=0.05 cl)
run;

* Delete global macro variable array ;
%estimate_full_free(array=vfull_balanced)
```

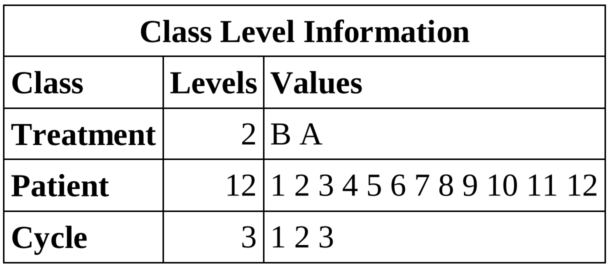
The statements and options of the 'mixed' procedure used in Program 15.4 are very similar to the ones used in the programs presented in section 13. Note the usage of the same variables in the 'model' and 'random' statements as Program 15.1. The macros, %estimate_full_global, %estimate_full and %estimate_full_free, developed for the estimation of individual treatment effects are used. Usage of the macros is discussed in section 13.2. Please refer to that section if you need further details. The 'repeated' statement is used to specify the covariance structure of the residual error part of the linear mixed-effects model. On the ‘repeated’ statement, the ‘type’ option requests a variance components structure. While the ‘group’ option specifies the grouping variable, “Treatment” in this case. To request distinct residual variances per cycle, the “Cycle” variable can be specified in the ‘group’ option of the ‘repeated’ statement. Different residual variances per both cycle and treatment can be requested by writing “Treatment*Cycle” in the same option of the ‘repeated’ statement. In Program 15.4, a linear mixed-effects model with different residual variances per treatment is fitted to the “NORMAL.ndata” balanced dataset. The execution of Program 15.4 outputs the tables presented below.

Table 15.1: Model information.



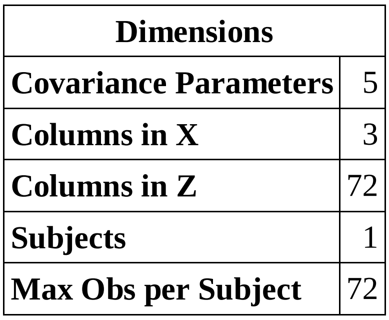
Model Information	
Data Set	NORMAL.NDATA
Dependent Variable	Y
Covariance Structure	Variance Components
Group Effect	Treatment
Estimation Method	REML
Residual Variance Method	None
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Table 15.2: Class level information.



Class Level Information		
Class	**Levels**	**Values**
Treatment	2	B A
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12
Cycle	3	1 2 3

Table 15.3: Dimensions.



Dimensions	
Covariance Parameters	5
Columns in X	3
Columns in Z	72
Subjects	1
Max Obs per Subject	72

Table 15.4: Number of observations.

Number of Observations	
Number of Observations Read	72
Number of Observations Used	72
Number of Observations Not Used	0

Table 15.5: Iteration history.

Iteration History			
Iteration	**Evaluations**	**-2 Res Log Like**	**Criterion**
0	1	931.31645659	
1	2	892.95203518	0.00002226
2	1	892.94327725	0.00000005
3	1	892.94325855	0.00000000

Table 15.6: Covariance parameter estimates.

Covariance Parameter Estimates					
Cov Parm	**Group**	**Estimate**	**Alpha**	**Lower**	**Upper**
Patient		19626	0.05	9084.92	69270
Patient*Cycle		677.92	0.05	63.7036	5.634E23
Treatment*Patient		942.27	0.05	113.96	2.364E12
Residual	**Treatment B**	9512.88	0.05	5010.93	24568
Residual	**Treatment A**	14081	0.05	8047.46	30738

Table 15.7: Fit statistics.

Fit Statistics	
-2 Res Log Likelihood	892.9
AIC (Smaller is Better)	902.9
AICC (Smaller is Better)	903.9
BIC (Smaller is Better)	905.4

Table 15.8: Solution for fixed effects.

Solution for Fixed Effects									
Effect	**Treatment**	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Intercept		2625.75	46.0872	11	56.97	<.0001	0.05	2524.31	2727.19
Treatment	**B**	188.72	28.5034	11	6.62	<.0001	0.05	125.99	251.46
Treatment	**A**	0	.	.	.	.	.	.	.

Table 15.9: Covariance matrix for fixed effects.

Covariance Matrix for Fixed Effects					
Row	**Effect**	**Treatment**	**Col1**	**Col2**	**Col3**
1	Intercept		2124.03	-469.67	
2	Treatment	B	-469.67	812.44	
3	Treatment	A			

Table 15.10: Type 2 tests of fixed effects.

Type 2 Tests of Fixed Effects				
Effect	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Treatment	1	11	43.84	<.0001

Table 15.11: Estimates.

Estimates								
Label	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
B-A,1	194.76	45.2557	11	4.30	0.0012	0.05	95.1533	294.37
B-A,2	168.66	45.2557	11	3.73	0.0033	0.05	69.0524	268.27
B-A,3	162.96	45.2557	11	3.60	0.0042	0.05	63.3513	262.57
B-A,4	218.86	45.2557	11	4.84	0.0005	0.05	119.25	318.47
B-A,5	204.53	45.2557	11	4.52	0.0009	0.05	104.92	304.13
B-A,6	161.76	45.2557	11	3.57	0.0044	0.05	62.1569	261.37
B-A,7	186.22	45.2557	11	4.11	0.0017	0.05	86.6169	285.83
B-A,8	183.08	45.2557	11	4.05	0.0019	0.05	83.4709	282.69
B-A,9	215.34	45.2557	11	4.76	0.0006	0.05	115.73	314.95
B-A,10	198.46	45.2557	11	4.39	0.0011	0.05	98.8537	298.07
B-A,11	193.99	45.2557	11	4.29	0.0013	0.05	94.3861	293.60
B-A,12	176.04	45.2557	11	3.89	0.0025	0.05	76.4310	275.65

Two residual variance estimates are displayed on Table 15.6. On this table, there is a row for the residual variance estimate under treatment “A”, and another row for the residual variance estimate under treatment “B”. The heteroscedastic linear mixed-effects model fitted in Program 15.4 contains only one additional parameter in relation to the homoscedastic model discussed in section 13. The residual variance estimated under treatment “A” is slightly superior to the residual variance estimated under treatment “B”. The overall treatment effect estimate presented on Table 15.8 is equal to the one resulting from the homoscedastic model and presented on Table 13.8. The dataset being balanced, the overall treatment effect does not depend on the covariance parameters of the model, so this result is expected. Nonetheless, the standard error of the overall treatment effect estimate does depend on the covariance parameters of the model regardless of the balance in the data. The value of standard error of the overall treatment effect estimate resulting from the heteroscedastic model and displayed in Table 15.8 is marginally higher than the one resulting from the homoscedastic model and displayed in Table 13.8. There is a small difference between the individual treatment effect estimates resulting from the heteroscedastic model (Table 15.11), and the individual treatment effect estimates resulting from the homoscedastic model (Table 13.27). The individual treatment effect estimates also depend on the covariance parameters of the model, regardless of the data being balanced or not. A visual comparison of Table 15.11 with Table 13.27 reveals a higher value of the standard error of the individual treatment effect estimates under the heteroscedastic model. There is a reduction in the degrees of freedom induced by the introduction of an additional parameter. This can explain the increase of the uncertainty associated to the estimation of the overall and individual treatment effects. Despite the small differences observed between the results from the heteroscedastic model and the homoscedastic model, the conclusions pertaining to the significance of the difference between the treatments remain the same. The confidence intervals of the overall treatment effect estimate and the twelve individual treatment effect estimates do not contain zero, suggesting the overall and individual significance of the difference between the treatments. In addition, the p-values relative to the overall treatment effect and the individual treatment effects are much lower than the chosen significance level of 0.05, suggesting the significance of the difference between the treatments on overage and individually for each patient.

16 Modelling time dependency

The datasets considered in the previous sections are from a series of n-of-1 trials where the treatments were randomized in cycles within each patient. The linear fixed-effects models do not consider a correlation between observations measured from the same individual, while the linear mixed-effects models do consider such a correlation. Observations that have more in common are more correlated between each other, while observations that have less in common are less correlated. Therefore, it is reasonable to assume that observations taken from the same individual are more correlated between each other. Observations taken from distinct individuals are certainly less correlated. Usually Statisticians model observations taken from different individuals as independent, meaning absence of any correlation. Although a correlation between different individuals may exist, and in some situations, the existence of such a correlation may be particularly evident. In n-of-1 trials, observations are repeatedly taken from an individual several units of time apart. Measurements that are recorded closer in time have more in common between each other, than measurements observed farther apart in time. In other words, measurements observed closer in time are more correlated between each other than measurements observed farther away in time. The linear mixed-effects model discussed in section 13 assumes that observations made in the same cycle are more correlated between each other, than observations measured under different cycles. Although this may be reasonable, the model assumes that the correlation between observations belonging to distinct cycles is the same irrespective of how far apart in time the cycles are. An assumption that may not be correct. Fitting the wrong correlation structure to the data may lead to biased standard error estimates of the overall and individual treatment effects, in turn leading to wrong conclusions. Therefore, it is important to fit the correct correlation structure to the data. Several correlation structure models are possible with the 'mixed' procedure of the SAS software. It is not possible to demonstrate all the available correlation structure models in this text. Two distinct time dependent correlation structure models are demonstrated in the current section. Then the results are analysed and discussed. The correlation structures demonstrated here require observations equally distanced in time. Observations unequally distanced in time can be modelled with the 'mixed' procedure by means of spatial correlation structures. These spatial correlation structures model the Euclidean distance between the observations. The Euclidean distance can be obtained for a single dimension such as time, one, two or three space dimensions, the space and time dimensions together, or any number of dimensions. Due to the limitations of the data available, spatial correlation structures cannot be demonstrated in this text. For information regarding the SAS implementation of spatial correlation structures, you can consult the SAS documentation available online [10]. The "NORMAL.ndata" dataset introduced by either Program 4.1 or Program 5.1 is used in this section for demonstration purposes. Several methods to select the most suitable correlation structure for a particular dataset are known. However, the series of n-of-1 trial datasets used throughout this text are inappropriate for such demonstrations. Littell, Stroup [31] discuss several correlation structure models in section 5.3.1 of chapter 5, and present methods for selecting appropriate correlation models with SAS in section 5.3.2 of the same chapter 5.

The following program demonstrates fitting a first-order autoregressive covariance structure to the “NORMAL.ndata” dataset introduced by either Program 4.1 or Program 5.1. With only two parameters regardless of the number of observations; the residual variance, and the autoregressive parameter; the autoregressive covariance model can be considered a simple one. This correlation structure is only appropriate for observations that are equally (or approximately equal) distanced in time. According to this covariance model, the correlation decreases in time as the separation between measurements increases. To deepen your understanding on this and other correlation models, you can read section 7.4 of Fitzmaurice, Laird [27], and or section 6.2 of Brown and Prescott [21]. Since in the “NORMAL.ndata” n-of-1 trials dataset the period is associated to the moment in time when an observation is measured, the “Period” variable can be used to identify the time in the model. Note that in the “NORMAL.ndata” dataset, for a given individual, a period corresponds to an observation. For each individual there are six periods. The data for patient 1 can be visualized on Table 6.1, and the data for patients 11 and 12 can be visualized on Table 6.2. To save space in this document, the whole “NORMAL.ndata” dataset is not presented. You can print the whole “NORMAL.ndata” dataset with a simple modification to Program 6.1. By removing the ‘outobs’ option from the ‘sql’ procedure in Program 6.1 all the observations from the dataset can be displayed. It is unknown whether the periods from “NORMAL.ndata” are equally distanced in time. Nonetheless, a first-order autoregressive model is fitted as an example. Remember that the “Period” variable is originally absent from the “NORMAL.ndata” dataset. Either Program 4.3 or Program 5.2 can be used to insert the period information in the “NORMAL.ndata” dataset. Afterwards the “Period” variable can be used in subsequent code.

Program 16.1: Fitting a first-order autoregressive linear mixed-effects model with 'proc mixed'.

```
* Write global macro variable array ;
%estimate_full_global(
        array=vfull_balanced,
        subject=Patient,
        treatment=Treatment,
        data=NORMAL.ndata
)

title "First-order autoregressive linear mixed-effects model";
proc mixed
                data=NORMAL.ndata
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Treatment (ref=first) Patient Period;
        model Y = Treatment /
                alpha=0.05 /* level for CI */
                cl /* t-type CI for fixed effects */
                htype=2 /* type 2 hypothesis test */
                solution; * solution for fixed effects ;
        random Patient Patient*Treatment;
        repeated Period /
                subject=Patient
                type=ar(1)
                r
                rcorr;
        %estimate_full(array=vfull_balanced, options=alpha=0.05 cl)
run;
```

The %estimate_full_global and %estimate_full macros introduced in section 11.2, and applied to linear mixed-effects models in section 13.2, are invoked in Program 16.1. The %estimate_full_global macro gets the necessary information to estimate the individual treatment effects from the "NORMAL.ndata" dataset, and saves this information to the "vfull_balanced" macro variable array. All the possible 'estimate' statements required to estimate the individual treatment effects are computed from the "vfull_balanced" macro variable array and inserted under the 'mixed' procedure when the %estimate_full macro is expanded. The "vfull_balanced" macro variable array is not yet removed from the SAS system. This macro variable array is used in a posterior program, which performs its removal. A more elaborate discussion of the %estimate_full_global macro can be found in the text following Program 13.7. While more details on the %estimate_full macro can be found in the text following Program 13.8. The fixed-effects part of the model is identical to the one on previous linear mixed-effects models. The outcome variable "Y" followed by the equal character and the "Treatment" variable are written in the 'model' statement. The coefficient of the "Treatment" variable is equal to the overall treatment effect, so it is important to include this variable as a fixed effect in the model. With the absence of the “Patient*Cycle” interaction, the ‘random’ statement is distinct from the one used in previous programs. The “Patient*Cycle” random interaction is included in previous programs for the following reason; both the “Patient” and the “Cycle” variables are part of the blocking structure in those n-of-1 trial designs, as such a “Patient*Cycle” random interaction induces a correlation between the observations of the same cycle measured on a given patient. This is a form of time dependency as explained in the top of the current section. Since other correlation structures using the “Period” can better account for time dependency, it makes more sense not to use a “Patient*Cycle” random interaction when those correlation structures are used. As written above, a “Patient*Cycle” random interaction assumes that the correlation between observations belonging to different cycles is the same regardless of how far apart in time the cycles are. This assumption may not be correct. However it may be best to use a “Patient*Cycle” random interaction when a better time dependent correlation alternative is unknown, and provided that the treatments were randomized in cycles within individuals. This may be better than not accounting for any time dependency at all. The “Patient” random effect induces a correlation between the observations belonging to the same individual. While the “Patient*Treatment” random interaction plays two very important roles in the model. Due to the “Patient*Treatment” random interaction, the model assumes that observations belonging to the same treatment and patient are more correlated between each other than observations recorded under distinct treatments for that same patient. Remember that observations that have more in common are more correlated between each other, while observations that have less in common are less correlated. The other role the “Patient*Treatment” random interaction plays is related to the estimation of the individual treatment effects. Distinct individual treatment effects can only be estimated from a linear model when some form of treatment by patient interaction is included in the model. A random treatment by patient interaction leads to estimates of the individual treatment effects with statistical properties that are considered superior to the naïve individual treatment effect estimates discussed in sections 7, 11, and 12. The ‘repeated’ statement is used to specify the correlation structure of the residual error term of the linear mixed-effects model. After the ‘repeated’ keyword, a variable indicating the repeated effect can be specified. The “Period” variable indicates the repeated effect in this case. In n-of-1 trials as in large scale cross-over trials, the interval of time an individual is being assessed under one treatment or the other is called a period. In n-of-1 trials, measurements are taken from an individual under one treatment or the other repeatedly over time. N-of-1 trials can be designed with a large number of periods. For this reason, n-of-1 trials have been defined in very few words as multi-period crossover trials. In fact, in n-of-1 trials the treatments are crossed over multiple times (or periods) on an individual. For this reason, in n-of-1 trials the period is associated to time and can be used to identify the repeated effect. The ‘subject’ option can be used to specify a variable indicating the physical location where the measurements are repeatedly measured. In n-of-1 trials, measurements are recorded repeatedly over time on an individual.

Therefore, the ‘subject’ option equals the “Patient” variable in the ‘repeated’ statement of Program 16.1. The ‘type’ option can be used to define the desired correlation structure for the residual term of the model. Several correlation structure types have been implemented in the ‘repeated’ statement of the ‘mixed’ procedure. A first-order autoregressive stochastic process can be specified by supplying the ‘ar(1)’ keyword to the ‘type’ option. Generally, the ‘r’ option requests the estimated covariance matrix of the residual errors for the first level of the category variable supplied to the ‘subject’ option. So in this case, the estimated covariance matrix of the residual errors for the first patient is outputted by SAS. Similarly, the ‘rcorr’ option requests the estimated correlation matrix of the residual errors for the first patient. Usually the correlation matrix is easier to interpret than the covariance matrix. The submission of Program 16.1 to the SAS system produces the twelve tables presented below.

Table 16.1: Model information.

Model Information	
Data Set	NORMAL.NDATA
Dependent Variable	Y
Covariance Structures	Variance Components, Autoregressive
Subject Effect	Patient
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Table 16.2: Class level information.

Class Level Information		
Class	**Levels**	**Values**
Treatment	2	B A
Patient	12	1 2 3 4 5 6 7 8 9 10 11 12
Period	6	1 2 3 4 5 6

Table 16.3: Dimensions.

Dimensions	
Covariance Parameters	4
Columns in X	3
Columns in Z	36
Subjects	1
Max Obs per Subject	72

Table 16.4: Number of observations.

Number of Observations	
Number of Observations Read	72
Number of Observations Used	72
Number of Observations Not Used	0

Table 16.5: Iteration history.

Iteration History			
Iteration	**Evaluations**	**-2 Res Log Like**	**Criterion**
0	1	931.31645659	
1	2	893.89385234	0.00008445
2	1	893.86130284	0.00000010
3	1	893.86126531	0.00000000

Table 16.6: Estimated R matrix for patient 1.

Estimated R Matrix for Patient 1						
Row	**Col1**	**Col2**	**Col3**	**Col4**	**Col5**	**Col6**
1	12521	114.18	1.0413	0.009496	0.000087	7.897E-7
2	114.18	12521	114.18	1.0413	0.009496	0.000087
3	1.0413	114.18	12521	114.18	1.0413	0.009496
4	0.009496	1.0413	114.18	12521	114.18	1.0413
5	0.000087	0.009496	1.0413	114.18	12521	114.18
6	7.897E-7	0.000087	0.009496	1.0413	114.18	12521

Table 16.7: Estimated R correlation matrix for patient 1.

Estimated R Correlation Matrix for Patient 1						
Row	**Col1**	**Col2**	**Col3**	**Col4**	**Col5**	**Col6**
1	1.0000	0.009119	0.000083	7.584E-7	6.916E-9	6.31E-11
2	0.009119	1.0000	0.009119	0.000083	7.584E-7	6.916E-9
3	0.000083	0.009119	1.0000	0.009119	0.000083	7.584E-7
4	7.584E-7	0.000083	0.009119	1.0000	0.009119	0.000083
5	6.916E-9	7.584E-7	0.000083	0.009119	1.0000	0.009119
6	6.31E-11	6.916E-9	7.584E-7	0.000083	0.009119	1.0000

Table 16.8: Covariance parameter estimates.

Covariance Parameter Estimates					
Cov Parm	**Subject**	**Estimate**	**Alpha**	**Lower**	**Upper**
Patient		19642	0.05	9126.71	68626
Treatment*Patient		723.64	0.05	68.7578	6.386E22
AR(1)	**Patient**	0.009119	0.05	-0.3756	0.3938
Residual		12521	0.05	8707.28	19542

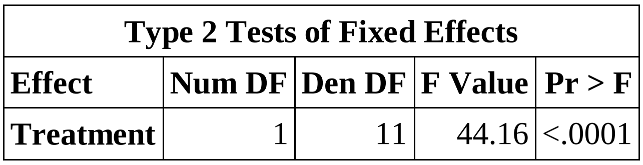
Table 16.9: Fit statistics.

Fit Statistics	
-2 Res Log Likelihood	893.9
AIC (Smaller is Better)	901.9
AICC (Smaller is Better)	902.5
BIC (Smaller is Better)	903.8

Table 16.10: Solution for fixed effects.

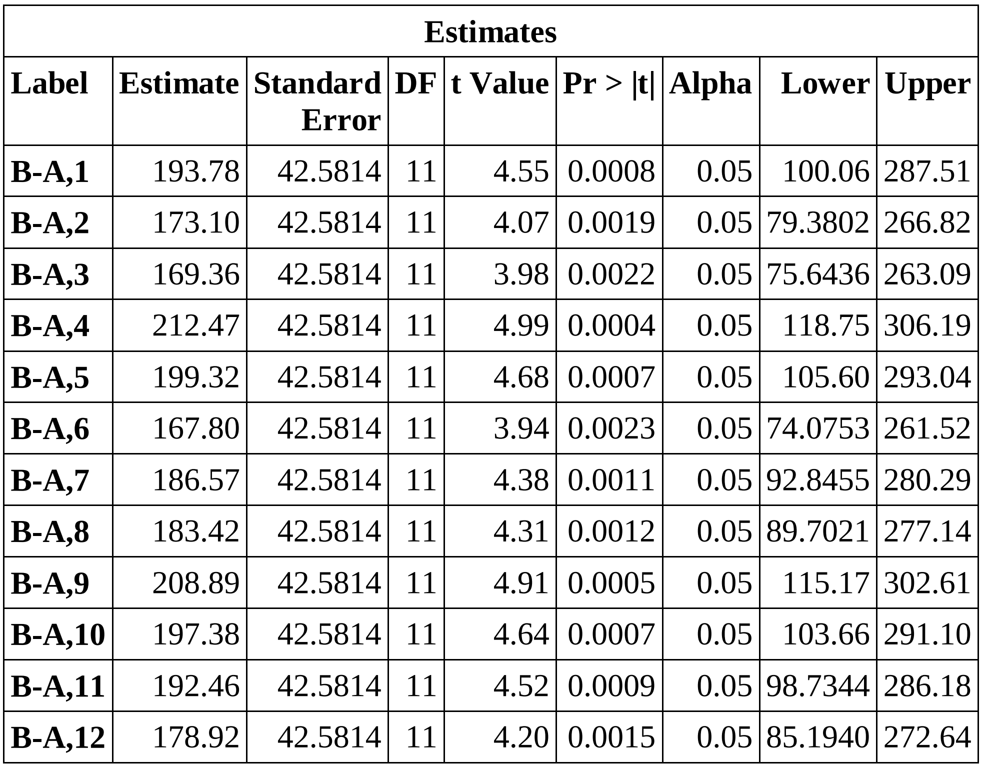
Solution for Fixed Effects									
Effect	**Treatment**	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Intercept		2625.87	45.2211	11	58.07	<.0001	0.05	2526.34	2725.40
Treatment	**B**	188.62	28.3849	11	6.65	<.0001	0.05	126.15	251.10
Treatment	**A**	0	.	.	.	.	.	.	.

Table 16.11: Type 2 tests of fixed effects.



Type 2 Tests of Fixed Effects				
Effect	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Treatment	1	11	44.16	<.0001

Table 16.12: Shrunk individual treatment effect estimates.



Estimates								
Label	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
B-A,1	193.78	42.5814	11	4.55	0.0008	0.05	100.06	287.51
B-A,2	173.10	42.5814	11	4.07	0.0019	0.05	79.3802	266.82
B-A,3	169.36	42.5814	11	3.98	0.0022	0.05	75.6436	263.09
B-A,4	212.47	42.5814	11	4.99	0.0004	0.05	118.75	306.19
B-A,5	199.32	42.5814	11	4.68	0.0007	0.05	105.60	293.04
B-A,6	167.80	42.5814	11	3.94	0.0023	0.05	74.0753	261.52
B-A,7	186.57	42.5814	11	4.38	0.0011	0.05	92.8455	280.29
B-A,8	183.42	42.5814	11	4.31	0.0012	0.05	89.7021	277.14
B-A,9	208.89	42.5814	11	4.91	0.0005	0.05	115.17	302.61
B-A,10	197.38	42.5814	11	4.64	0.0007	0.05	103.66	291.10
B-A,11	192.46	42.5814	11	4.52	0.0009	0.05	98.7344	286.18
B-A,12	178.92	42.5814	11	4.20	0.0015	0.05	85.1940	272.64

The model information displayed in Table 16.1 contains some differences in relation to the model information tables presented in the sections above, such as Table 13.26. In Table 16.1 two covariance structures are reported, variance components, and autoregressive. There is also an additional row reporting the subject effect, which is absent from most model information tables above presented. While, Table 16.2, and Table 16.4 contain information on the dataset used, being equal to tables previously presented, where the same "NORMAL.ndata" dataset is supplied to the 'data' option of the 'mixed' procedure. The information presented in Table 16.3 is relative to the dimensions of the linear mixed-effects model fitted to the data. According to Table 16.5, the estimation algorithm converged to a solution matching the convergence criterion in three iterations. The covariance matrix of the residual errors is presented in Table 16.6. This covariance matrix is relative to patient 1 according to the output. All the twelve patients share an equal covariance matrix of the residual errors according to the model fitted to the data. Any differences observed usually are due to a difference in the number of periods. However, the dataset used is balanced, so each of the twelve patients has data for the same number of periods. It is possible to fit a model with distinct covariance matrices per patient by specifying the “Patient” variable in the ‘group’ option of the ‘repeated’ statement. However, such a model has an excessive number of parameters, requiring an identically excessive amount of data for each patient. Fitting such a model to series of n-of-1 trials data is generally a very bad idea. Note that the diagonal elements of the covariance matrix are all equal. The diagonal elements contain the variance of the residual error pertaining to each of the six periods. In fact, the covariance matrix of the residual errors is a symmetric one. The off-diagonal elements contain the covariance between the residual errors associated to the periods. It can be observed that the covariance between the residual errors is positive and decreases as the time between the residual errors increases. The correlation matrix of the residual errors presented in Table 16.7 can be obtained from the covariance matrix of the residual errors from Table 16.6, by using elementary linear algebra. Therefore, like the covariance matrix, the correlation matrix of the residual errors is equal for the twelve patients. The correlation matrix can be considered easier to interpret than the covariance matrix. The correlation of a random variable between itself is equal to unity, so the diagonal elements of the correlation matrix equal unity. Like any covariance matrix, any correlation matrix is symmetric. One can observe from Table 16.7 that the correlation between the residual errors is positive and decreases as the associated period or time increases. This is a direct consequence of the autoregressive stochastic process fitted to the data. You will observe this behaviour every time you fit an autoregressive correlation structure to your data. See for example section 7.4 of Fitzmaurice, Laird [27]. The covariance parameter estimates of the model and respective statistics lie in Table 16.8. Four rows are contained in Table 16.8, one for each covariance parameter. The covariance parameters are; the variance of the patient random term, the variance of the treatment by patient interaction random term, the autoregressive parameter, and the variance of the residual error. The autoregressive parameter can be any real number between minus unity and unity, and is essentially a correlation. Note that the autoregressive parameter equals the correlation between any residual error from a period and the residual error associated to the adjacent closest periods. Look attentively at Table 16.7 and see where you can find the autoregressive parameter estimate from Table 16.8. An autoregressive stochastic process with an autoregressive parameter equal to zero implies a correlation between the residual errors equal to zero, and is essentially equivalent to a “variance components” correlation structure. Remember that a “variance components” correlation structure like the one considered by the linear mixed-effects models discussed in previous sections, assumes that the residual errors are independent. Look at Table 16.8 and observe that zero is contained within the confidence interval of the autoregressive parameter. This result suggests that the autoregressive parameter is not significant. There is little statistical evidence that the residual errors are dependent. A simpler “variance components” correlation structure is sufficient to model the data. Therefore, a simpler “variance components” correlation structure is preferable for the dataset over the more complex first-order autoregressive one. For other datasets, an autoregressive correlation structure or maybe other correlation structures might fit better than a "variance components" correlation structure. Each case is unique and must be considered distinctively. Additional methods for selecting an appropriate correlation structure are presented by Littell, Stroup [31] in section 5.3.2 of chapter 5. The confidence intervals of the other covariance parameters of the model do not contain zero, and as such are significant at the chosen significance level. The fit statistics contained in Table 16.9 can be useful in comparing several models as regards their goodness of fit, and make a choice between those models. The first fit statistic from Table 16.9, minus two times the residual log likelihood, is not appropriate to perform a comparison between models. Instead, the likelihood ratio test can be used to compare two nested models. Two models are said to be nested when the smaller model is contained within the bigger model. The likelihood ratio statistic is not directly available from the 'mixed' procedure's output. It can be computed from the log likelihood statistics from two nested models estimated using maximum likelihood instead of residual maximum likelihood. The fit statistics from the smaller model, which is nested within the model fitted by Program 16.1, are not available. As such, the likelihood ratio test is not demonstrated here. However, the likelihood ratio test is well covered in any statistics textbook that discusses regression methods, see for example section 6.2.2 of Brown and Prescott [21]. The statistics that follow in the same Table 16.9; the AIC, AICC and BIC; account for the likelihood and the number of parameters of the model. These three statistics can directly be used to compare distinct regression models. The model that fits the data the best, and that is the simplest, is the preferable model. Therefore, as Table 16.9 informs, smaller is better, for the three statistics! Looking back at the fit statistics from the linear mixed-effects model fitted to the same "NORMAL.ndata" dataset displayed in Table 13.7, one can observe that the fit statistics; AIC, AICC and BIC; are slightly smaller for the previously fitted model. This observation is consistent with the non-significance of the autoregressive parameter suggested by Table 16.8, and mentioned above. The first-order autoregressive correlation structure does not fit the data better than the simpler "variance components" correlation structure. The overall treatment effect statistics can be found in the second row of Table 16.10. The overall treatment effect estimate and associated statistics from Table 16.10 are quite close to the ones resulting from the model with the cycle by patient random interaction from Table 13.8. The p-value is much lower than the chosen significance level, and the confidence interval of the overall treatment effect estimate does not contain the origin, suggesting that the treatments differ significantly between each other. The individual treatment effect estimates and related statistics are presented in Table 16.12, the last table resulting from Program 16.1. A few differences are observed while comparing Table 16.12, which results from the linear mixed-effects model with a first-order autoregressive correlation structure, with Table 13.27 resulting from the linear mixed-effects model with a cycle by patient random interaction. Note that in Table 16.12 the estimates are more shrunk towards the overall mean, than the estimates in Table 13.27. A shrunk individual treatment effect is a weighted average of the overall treatment effect and the respective naïve individual treatment effect, where the weights depend on the covariance parameters of the model. See particularly sections 7.5 and 7.6 of Verbeke and Molenberghs [24], or sections 8.6 and 8.7 of Fitzmaurice, Laird [27]. For the models in question, the weights of the shrunk individual treatment effect estimates depend only on the variance of the treatment by patient random term and on the covariance structure of the residual error. A lower variance of the treatment by patient random term gives more weight to the overall treatment effect, and brings the shrunk individual treatment effects closer to it. Note on Table 16.8, that the variance of the treatment by patient random interaction resulting from the model with first-order autoregressive correlation structure is lower than the one resulting from the model with cycle by patient random term on Table 13.6. This explains the differences observed in the individual treatment effect estimates resulting from both models. The same standard error of the individual treatment effect is estimated for the twelve individuals. Note however, that when a time dependent correlation structure is modelled through a linear mixed-effects model; the overall treatment effect estimate, the individual treatment effect estimates, and their respective standard error estimates; may differ between individuals, even when the series of n-of-1 trials is balanced. When such differences do occur, can be attributed to differences in the sequence of treatments across individuals, which arise from the randomization of treatments within individuals. For the balanced dataset in question, the standard error of the individual treatment effect estimated from the autoregressive linear mixed-effects model is equal across individuals, because the sequence of treatments is the same for the twelve individuals. By printing and looking carefully at the “NORMAL.ndata” dataset, one can observe the same ABABAB sequence of treatments for the twelve individuals. The whole “NORMAL.ndata” is too big to fit in a single page of this document, so it is not presented here. By removing the ‘outobs’ option of the ‘sql’ procedure from Program 6.1, the whole “NORMAL.ndata” dataset can be printed and visualized. The fact that the same sequence of treatments is observed in the “NORMAL.ndata” dataset does not necessarily imply that the treatments were not randomized in cycles within individuals! Note that the order of administration of the treatments within a given cycle can only be inferred by the presence of the “Period” variable on that dataset. Remember that the “Period” variable is not originally present in the “NORMAL.ndata” dataset. The “Period” variable is inserted in the “NORMAL.ndata” dataset by a previous execution of either Program 4.3, or Program 5.2. The order of administration of the treatments in unknown, and both programs assume the order of the observations in the dataset to determine the “Period” variable. This is done to allow for the demonstration of the programs of this section. Moreover, this should not be done under any circumstance in practice with real clinical trials. In practice, only information that is known should be registered in the datasets. In practice, a clinical trial is well planned before being carried out. Both the trial design and the statistical models are decided before collecting the data. As such, variables that do not take part in the statistical models do not need to be measured and registered. While those variables that play a role in the statistical analysis must surely be measured and recorded. The standard error of the individual treatment effect estimates resulting from the autoregressive model is lower. A lower standard error of an estimate may indicate a higher precision of that same estimate. However, a lower standard error does not necessarily mean that an estimate is better than the other is. Even when both estimates are unbiased. In fact the estimate of the standard error of an estimate can be biased itself and induce the analyst in error. In that case, wrong inferences and wrong decisions can be made! To minimize the possibility of wrong inferences it is important to find a model with a correlation structure that fits the data reasonably well. The standard error of the individual treatment effect estimates being lower and the degrees of freedom the same, the confidence intervals of those estimates are narrower for the model with autoregressive correlation structure. The p-value of each of the twelve estimates is much lower than the significance level, also the confidence interval of the same estimates does not contain zero, suggesting that the treatments differ significantly individually. The overall treatment effect being positive indicates that the outcome variable is higher under treatment “B” than under treatment “A”. Therefore, if the patient prefers higher values of the outcome variable, treatment “B” is recommended, otherwise treatment “A” shall be the choice.

The demonstration of a second time dependent correlation structure model is promised in the first paragraph of the present section. A first-order autoregressive moving-average correlation structure was chosen for the alternative demonstration. A first-order autoregressive moving-average stochastic process is a combination of two processes, a first-order autoregressive process, and a first-order moving-average process. These models are typically better discussed on textbooks covering time series statistical methods, than on textbooks that cover linear mixed-effects regression. Chapter 2 and chapter 3 of Brockwell and Davis [32], and or chapter 3 of Brockwell and Davis [33] are suggested as references on this subject. Like the first-order autoregressive covariance structure demonstrated above, the first-order autoregressive moving-average correlation structure is appropriate to model observations equally distanced in time. The ‘mixed’ procedure models a first-order autoregressive moving-average correlation structure in three parameters, the autoregressive parameter (Rho), a parameter modelling the moving-average component (Gamma), and the variance of the residual error. The “NORMAL.ndata” balanced dataset introduced by either Program 4.1 or Program 5.1 is used in the next program to fit a first-order autoregressive moving-average covariance structure. The “Period” categorical variable is used to identify the time, indicating the repeated effect in the model. Again, remember that the “Period” variable is originally absent from the “NORMAL.ndata” dataset. A previous run of either Program 4.3 or Program 5.2 insert the period information in the “NORMAL.ndata” dataset. It is assumed that the time between the periods is constant during each n-of-1 trial in the series.

Program 16.2: Fitting a first-order autoregressive moving-average linear mixed-effects model with 'proc mixed'.

```
title "First-order autoregressive moving-average linear mixed-effects model";
proc mixed
                data=NORMAL.ndata
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Treatment (ref=first) Patient Period;
        model Y = Treatment /
                alpha=0.05 /* level for CI */
                cl /* t-type CI for fixed effects */
                htype=2 /* type 2 hypothesis test */
                solution; * solution for fixed effects ;
        random Patient Patient*Treatment;
        repeated Period /
                subject=Patient
                type=arma(1,1)
                r
                rcorr;
        %estimate_full(array=vfull_balanced, options=alpha=0.05 cl)
run;

* Delete global macro variable array ;
%estimate_full_free(array=vfull_balanced)
```

The statements and options within the 'mixed' procedure of Program 16.2 are very similar to the ones from the 'mixed' procedure used in Program 16.1. Actually, there is only a minor difference between the two 'mixed' procedures. The 'type' option is set to the 'arma(1,1)' keyword in Program 16.2. This way a linear mixed-effects model with a first-order autoregressive moving-average correlation structure is fitted to the "NORMAL.ndata" dataset. Remember that the "vfull_balanced" macro variable array is filled with data during the invocation of the %estimate_full_global macro in Program 16.1. The "vfull_balanced" macro variable array is required by the %estimate_full macro of the same Program 16.1, which does not remove the macro variable array from the SAS system. When Program 16.2 runs, the "vfull_balanced" macro variable array is still present in the computer's memory, and is used again by the %estimate_full macro under the 'mixed' procedure of this program. Finally, the %estimate_full_free macro in Program 16.2 removes the "vfull_balanced" macro variable array from the SAS system. Due to the similarities between Program 16.2 and Program 16.1, no additional details are given here in relation to Program 16.2. It is assumed that each section of this document is read continuously and completely. If you are commencing your read at this point, and desire more details, please read the text following Program 16.1. The 'mixed' procedure in Program 16.2 uses the same "NORMAL.ndata" dataset as the 'mixed' procedure from Program 16.1. Hence, the class level information table and number of observations table is the same for both programs. As such, Program 16.2 outputs Table 16.2, Table 16.4, and the ten tables presented below.

Table 16.13: Model information.

Model Information	
Data Set	NORMAL.NDATA
Dependent Variable	Y
Covariance Structures	Variance Components, Autoregressive Moving Average
Subject Effect	Patient
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

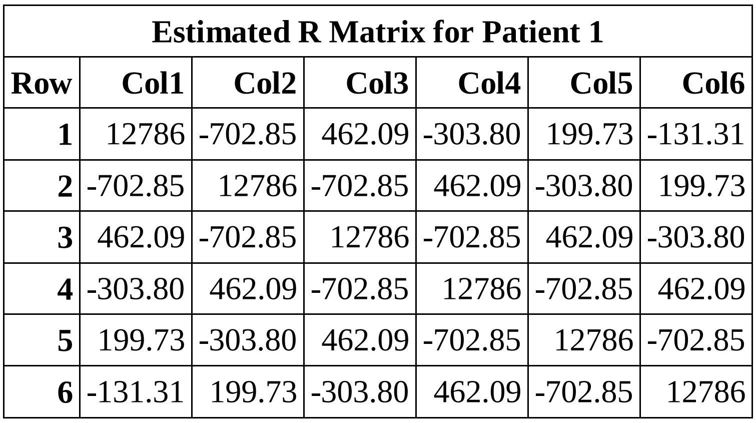
Table 16.14: Dimensions.

Dimensions	
Covariance Parameters	5
Columns in X	3
Columns in Z	36
Subjects	1
Max Obs per Subject	72

Table 16.15: Iteration history.

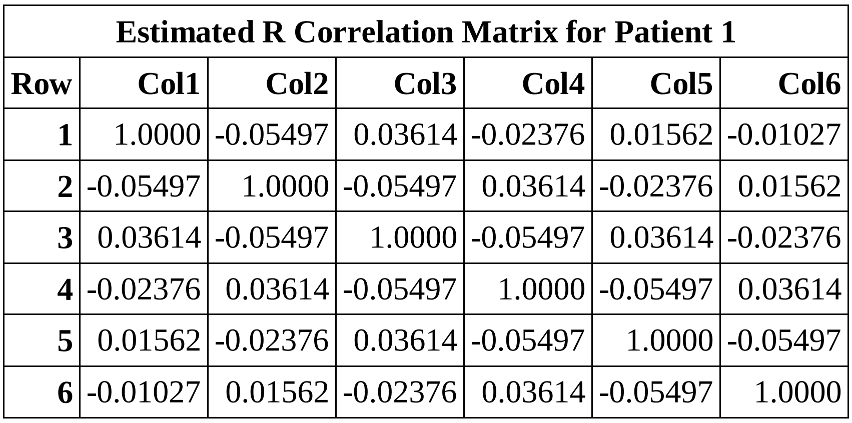
Iteration History			
Iteration	**Evaluations**	**-2 Res Log Like**	**Criterion**
0	1	931.31645659	
1	3	893.83806227	0.00019706
2	2	893.75918355	0.00000299
3	1	893.75805235	0.00000001

Table 16.16: Estimated R matrix for patient 1.



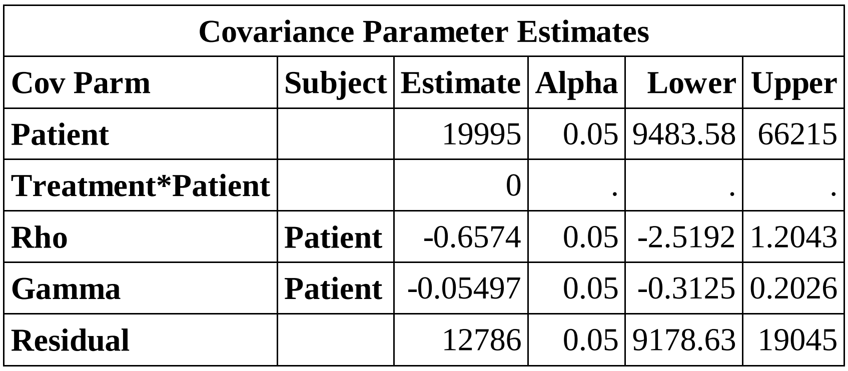
Estimated R Matrix for Patient 1						
Row	**Col1**	**Col2**	**Col3**	**Col4**	**Col5**	**Col6**
1	12786	-702.85	462.09	-303.80	199.73	-131.31
2	-702.85	12786	-702.85	462.09	-303.80	199.73
3	462.09	-702.85	12786	-702.85	462.09	-303.80
4	-303.80	462.09	-702.85	12786	-702.85	462.09
5	199.73	-303.80	462.09	-702.85	12786	-702.85
6	-131.31	199.73	-303.80	462.09	-702.85	12786

Table 16.17: Estimated R correlation matrix for patient 1.



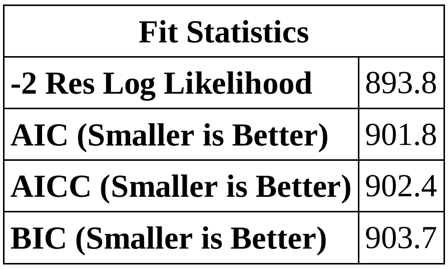
Estimated R Correlation Matrix for Patient 1						
Row	**Col1**	**Col2**	**Col3**	**Col4**	**Col5**	**Col6**
1	1.0000	-0.05497	0.03614	-0.02376	0.01562	-0.01027
2	-0.05497	1.0000	-0.05497	0.03614	-0.02376	0.01562
3	0.03614	-0.05497	1.0000	-0.05497	0.03614	-0.02376
4	-0.02376	0.03614	-0.05497	1.0000	-0.05497	0.03614
5	0.01562	-0.02376	0.03614	-0.05497	1.0000	-0.05497
6	-0.01027	0.01562	-0.02376	0.03614	-0.05497	1.0000

Table 16.18: Covariance parameter estimates.



Covariance Parameter Estimates					
Cov Parm	**Subject**	**Estimate**	**Alpha**	**Lower**	**Upper**
Patient		19995	0.05	9483.58	66215
Treatment*Patient		0	.	.	.
Rho	**Patient**	-0.6574	0.05	-2.5192	1.2043
Gamma	**Patient**	-0.05497	0.05	-0.3125	0.2026
Residual		12786	0.05	9178.63	19045

Table 16.19: Fit statistics.



Fit Statistics	
-2 Res Log Likelihood	893.8
AIC (Smaller is Better)	901.8
AICC (Smaller is Better)	902.4
BIC (Smaller is Better)	903.7

Table 16.20: Solution for fixed effects.

Solution for Fixed Effects									
Effect	**Treatment**	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Intercept		2625.13	45.1891	11	58.09	<.0001	0.05	2525.67	2724.59
Treatment	**B**	189.19	28.9106	11	6.54	<.0001	0.05	125.56	252.82
Treatment	**A**	0	.	.	.	.	.	.	.

Table 16.21: Type 2 tests of fixed effects.

Type 2 Tests of Fixed Effects				
Effect	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Treatment	1	11	42.82	<.0001

Table 16.22: Shrunk individual treatment effect estimates.

Estimates								
Label	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
B-A,1	189.19	28.9106	11	6.54	<.0001	0.05	125.56	252.82
B-A,2	189.19	28.9106	11	6.54	<.0001	0.05	125.56	252.82
B-A,3	189.19	28.9106	11	6.54	<.0001	0.05	125.56	252.82
B-A,4	189.19	28.9106	11	6.54	<.0001	0.05	125.56	252.82
B-A,5	189.19	28.9106	11	6.54	<.0001	0.05	125.56	252.82
B-A,6	189.19	28.9106	11	6.54	<.0001	0.05	125.56	252.82
B-A,7	189.19	28.9106	11	6.54	<.0001	0.05	125.56	252.82
B-A,8	189.19	28.9106	11	6.54	<.0001	0.05	125.56	252.82
B-A,9	189.19	28.9106	11	6.54	<.0001	0.05	125.56	252.82
B-A,10	189.19	28.9106	11	6.54	<.0001	0.05	125.56	252.82
B-A,11	189.19	28.9106	11	6.54	<.0001	0.05	125.56	252.82
B-A,12	189.19	28.9106	11	6.54	<.0001	0.05	125.56	252.82

The variance components and autoregressive moving-average covariance structures are indicated in Table 16.13, which provides information pertaining to the model. An additional covariance parameter in relation to Table 16.3 is indicated in Table 16.14. The estimated covariance matrix of the residual error is represented in Table 16.16, while the related estimated correlation matrix of the residual error is displayed in Table 16.17. The estimated covariance and correlation matrices are for patient 1. There are twelve patients in the series of n-of-1 trials. Note that for the model in question the covariance and correlation matrices are identical for all the individuals in the sample. Therefore, there is no need to display a covariance and correlation matrix for each patient. Any correlation matrix is directly related to the respective covariance matrix, and vice-versa. One can be computed from the other through elementary matrix algebra. Either one or the other matrix can be looked at to make similar observations. Due to the more limited set of numbers a correlation can assume, a correlation matrix can be easier to interpret. Remember that by definition, a correlation can assume any real number between minus unity and unity inclusive, while a covariance can assume any real number. A correlation matrix is a standardization of the covariance matrix. Each element is obtained by dividing the covariance between two variables by the product of the respective standard deviations. A covariance and a variance can have units, such as pascal, volt, ampere, meter, etc. Due to the referred standardization, the correlation has no units, being dimensionless by definition. The correlation of a random variable with itself equals unity, so the diagonal elements of a correlation matrix are equal to unity. In addition, covariance and correlation matrices are symmetric by definition. Look at Table 6.1 and note that the sequence of treatments for patient 1 is ABABAB. Now look at Table 16.17 and observe that the correlation between two residual errors belonging to the same treatment is positive, while the correlation between two residual errors associated to different treatments is negative. This result suggests that when there is an increase in the residual error under one treatment, there is a decrease under the other treatment. Since the model assumes the same correlation structure for all the patients, and the correlation matrix of the residual error is estimated with the data from all the patients, this trend can only be understood while going from one patient to the next, and so on... So suppose that you observed a patient, and then you observe another patient and you note an increase in the residual error under one treatment, you did not yet observe the residual error under the other treatment for this patient, then you should expect a decrease in the residual error under the other treatment for this patient. Table 16.18 holds the covariance parameter estimates of the linear mixed-effects model. The results contained within Table 16.18 are striking! Note that the estimate of the variance of the treatment by patient interaction error is zero, and there are no estimates of the lower and upper limits of the associated confidence interval, suggesting that the estimate is exact. Both the autoregressive parameter (Rho), and the moving-average component (Gamma) estimates are negative and non-significant. Observe that zero lies within the estimated confidence interval for each of these two parameter estimates. The non-significance of these two parameter estimates suggests that the first-order autoregressive moving-average correlation structure does not fit the data better than the simpler variance components correlation structure. Note that when both the autoregressive parameter (Rho), and the moving-average component (Gamma) are equal to zero, the first-order autoregressive moving-average covariance structure becomes a simpler variance components correlation structure. To better understand this, look for information regarding the 'repeated' statement in the online documentation of the 'mixed' procedure[10]. The fit statistics from Table 16.19 are close to the fit statistics from the less complex variance components model presented in

Table 13.7. These results suggest that both models explain a similar amount of variation of the outcome variable with a similar number of parameters, despite being quite different! Still, the overall and individual treatment effect estimates resulting from both models are quite distinct. The overall treatment effect estimate and associated statistics are found on Table 16.20. Here the associated p-value and confidence interval suggest the average significance of the difference of the treatments being trialled. The same suggestion is made by the very low p-value resulting from the type 2 test of the "Treatment" fixed effect on Table 16.21. Note that the F value from Table 16.21 equals the square of the t value associated to the overall treatment effect from Table 16.20. This is a consequence of the well-known result that any F random variable with a single numerator degree of freedom, and a given denominator degrees of freedom, equals the square of a t random variable with that same denominator degrees of freedom. The two overall treatment effect statistical hypothesis tests are essentially equivalent. The estimated individual treatment effects displayed on Table 16.22 are the same for the twelve patients! The individual treatment effect estimates are equal to the estimate of the overall treatment effect. In fact, distinct individual treatment effects can only be estimated from a linear model in the presence of a significant treatment by patient interaction either fixed or random. As noted previously while looking at Table 16.18, the model estimates an exact treatment by patient interaction. Since the individual treatment effect estimates are all equal to the overall treatment effect, the treatment by patient interaction is not only exact but also equal to zero. Remember that for the linear model in question, the individual treatment effect equals the sum of the overall treatment effect to that individual's random treatment by patient interaction error. The results suggest that the first-order autoregressive moving-average correlation structure is explaining the treatment by patient random interaction variation. Additionally, the results suggest that a linear mixed-effects model with such a correlation structure may not be adequate to estimate individual treatment effects. One of the reasons for running n-of-1 trials is a suspected variation in the individual response to the treatments. The identification of an individual treatment response requires the estimation of individual treatment effects and appropriate inferences on these estimates. However, note that the purpose of a first-order autoregressive moving-average correlation structure is to model time dependency. This correlation structure, which is adequate to model observations that are equally, distanced in time, models in fact the ordered time observations. It has been noted previously that the sequence of treatments for patient 1 is ABABAB. If you modify Program 6.1 by removing the 'outobs' option from the 'sql' procedure, you can print and visualize the whole "NORMAL.ndata" dataset. The whole dataset is not presented here due to space constraints. Then you can look at the table holding the data and note that all the twelve patients share the same ABABAB sequence of treatments. Again, note that the same sequence of treatments for all the individuals in the dataset "NORMAL.ndata" does not necessarily imply that the treatments were not randomized within the individuals. Remember that the "Period" variable, which indicates the order of administration of the treatments within

a given individual, is absent from the original dataset. A previous run of either Program 4.3 or Program 5.2 insert the "Period" variable in the dataset. Both programs assume the order of the observations in the original dataset to determine the "Period" variable. This operation on the data may not reflect the actual order of administration of the treatments within the patients. Both Program 4.3 and Program 5.2 were developed to permit the demonstration of programs where time dependency is modelled. It is important to stress that such operations on a dataset should never be done in practice with real clinical trials. In practice with actual clinical trials, only known and true information should be registered in a dataset. The same sequence of treatments for all the individuals is very improbable in a properly randomized series of n-of-1 trials. Since the first-order autoregressive moving-average covariance structure models the order of the observations in time, it is very unlikely that it will explain all the treatment by patient random variation in a properly randomized series of n-of-1 trials. A very important lesson can be learned from these results. If the order of the treatments is the same over time and for all the individuals in a series of n-of-1 trials, a correlation structure that models the order of the observations in time may explain all the variation of the treatment by patient random interaction error, rendering the estimation of distinct individual treatment effects impossible. In n-of-1 trials, whether in series or single, the treatments under study are administered to individuals alternately over time, so there is the risk of some confusion between treatment effect and time effect. The confusion between treatment effect and time effect can be minimized by preventing the same order of treatments consistently over time. A proper randomization of treatments within individuals in n-of-1 trials, which is useful for other reasons that are not relevant to the issue here, can minimize the confusion between treatment effect and time effect.

The next and the last two programs of the present section, were designed to assess the effect of the within individual randomization of treatments procedure in series of n-of-1 trials, when estimating individual treatment effects. Randomization of treatments within individuals in n-of-1 trials has been widely reported in the scientific literature. However, the reasons for the application of a randomization procedure in n-of-1 trials are rarely provided. One of the obvious reasons, which is shared with other types of randomized clinical trials, is related to the minimization of a treatment preference bias introduced by the people involved in the trial; be it the patient, the investigator, or whomever may pose a threat of such bias. For the particular case of n-of-1 trials, the randomization of treatments within individuals over time minimizes the confusion between treatment effect and time effect. In n-of-1 trials, treatments are administered to individuals alternately over time, so there is the risk of confusion between treatment effect and time effect. It is verified in the previous paragraph that a model, which considers time dependency is not useful for the estimation of distinct individual treatment effects, if there is confusion between time effect and treatment effect. The purpose of the following program is to provide proof that distinct individual treatment effects can still be estimated from a linear mixed-effects model with a time dependent correlation structure. This is in contrast to what is observed in the previous paragraph! To begin with the proof; consider the dataset “NORMAL.ndata” used as example throughout the current section. Now suppose that the same realizations of the outcome variable of this dataset are observed for the same patient, the same treatment, and the same cycle, but in a random order. For a particular patient and cycle, where before there is an AB sequence of treatments, now there is the possibility of observing the reverse sequence BA. The purpose is to simulate a proper randomization of treatments in cycles within individuals. Then obtain a new dataset with the results of the simulated randomization procedure. Finally, use a ‘mixed’ procedure similar to the one used in Program 16.2, fitting a first-order autoregressive moving-average linear mixed-effect model to the dataset obtained previously. Run the code and assess the results. To avoid having a large amount of code spanning across several pages, the demonstration is divided in two programs. After each program, a brief explanation of the code is provided. The programs are presented below.

Program 16.3: Simulating randomized series of n-of-1 trials.

```
%let seed=314159;

proc iml;
    varNames={'Patient' 'Treatment' 'Cycle' 'Period' 'Y'};
    use NORMAL.ndata;
    read all var varNames; * read variables into vectors ;
    close NORMAL.ndata;
    uPatient=unique(Patient); * get unique patients ;
    uCycle=unique(Cycle); * get unique cycles ;
    call randseed(&seed); * set seed for RNG;
    do i=1 to ncol(uPatient) by 1; * loop through the patients ;
        do j=1 to ncol(uCycle) by 1; * loop through the cycles ;
            index=loc(Patient=uPatient[i] & Cycle=uCycle[j]); * locate patient and cycle ;
            call randgen(x, "bernoulli", 0.5); * toss the coin ;
            if x then do; * if true then switch order of treatments ;
                p1=Period[index[1]];
                Period[index[1]]=Period[index[2]];
                Period[index[2]]=p1;
            end;
        end;
    end;
    create WORK.rndata var varNames;
    append;
    close WORK.rndata;
quit;

%symdel seed;

proc sort data=WORK.rndata;
    by Patient Period;
quit;
```

Program 16.3 opts for the SAS/IML programming language to handle the data. IML stands for Interactive Matrix Language, and is an alternative to the data steps introduced in section 4, and the ‘sql’ procedure introduced in section 5. The SAS/IML programming language can be easier to understand if you are already familiar with programming languages that share at least some similarities, such C, Fortran or R. A comparison between data steps and the SAS/IML language is provided by Wicklin [34] in section 1.2 and or by the same Wicklin [35] in section 1.5. The SAS/IML language can be easier to code when simulations are required. The SAS/IML programming techniques used in Program 16.3 are described in detail in section 3.2 and section 3.3 of Wicklin [34]; and section 2.6, and section 2.7 of Wicklin [35]. Note that on Program 16.3 all the SAS/IML code is enclosed within the ‘iml’ procedure, and the ‘quit’ statement terminates the same ‘iml’ procedure. There are comments throughout the program to help the reader understand the code. The code is explained here briefly in plain English. Program 16.3 starts by assigning the “seed” macro variable to an integer number, which has the first six digits of the number pi. Then all the variables from “NORMAL.ndata” are read into separate vectors. Afterwards the unique values of the “Patient” and “Cycle” vectors are determined. These unique values permit the program to loop over each patient and cycle, instead of looping over each observation of the “Patient” and “Cycle” variables, as a data step typically does. On the following statement, the ‘randseed’ function is called with the parameter equal to the value of the “seed” macro variable defined above. This function sets the seed for the random number generator, which generates pseudorandom numbers. The numbers are not truly random, therefore being referred as pseudorandom. If the same seed is maintained across different runs of the program, the same results are obtained, despite the simulation. The numbers are not truly random but are designed to look like random. After setting the seed for the random number generator, Program 16.3 loops over the unique patients and the unique cycles, and for each loop, it finds the index of the observations for that patient and cycle in the original dataset. It is known from previous visualizations of the “NORMAL.ndata” dataset that there are only two observations for each patient and cycle, each observation corresponding to a different period. Within the loop, a Bernoulli pseudorandom number with probability equal to one-half is generated, by calling the ‘randgen’ function. There are two possibilities for this pseudorandom number, either zero or one, both being equally likely. If unity is obtained from the ‘randgen’ function, then the order of the periods is switched, by switching the elements of the “Period” vector indexed at the values previously determined at the beginning of the loop. The ‘iml’ procedure is finalized by creating a new dataset “WORK.rndata” with the modified “Period” variable, and the same “Patient”, “Treatment”, “Cycle”, and “Y” variables as the “NORMAL.ndata” dataset. After quitting the ‘iml’ procedure, Program 16.3 removes the “seed” macro variable from the SAS system. Note that the observations from “NORMAL.ndata” are sorted by the “Patient” and “Period” variables. The sorting of “NORMAL.ndata” has been previously accomplished by running either Program 4.3 or Program 5.4. The “WORK.rndata” dataset results from pseudo randomly switching the periods within the cycles for each patient, so it is not sorted by the “Period” variable. The ‘sort’ procedure is used in the bottom of Program 16.3, to sort “WORK.rndata” by increasing order of the elements of the “Patient” and “Period” variables. After Program 16.3 is executed, the “rndata” dataset becomes available in the “WORK” library, and its use is then possible in subsequent programs. The second program in the demonstration makes use of the “WORK.rndata” dataset, and is presented next.

Program 16.4: Fitting a first-order autoregressive moving-average linear mixed-effects model with ‘proc mixed’- Simulated randomized series of n-of-1 trials.

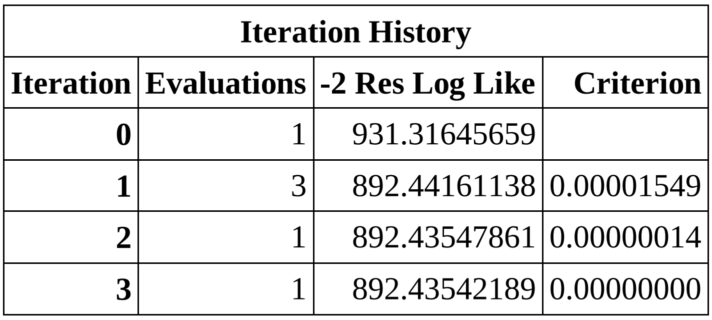
```
title "First-order autoregressive moving-average linear mixed-effects model";
title2 "Randomized series of n-of-1 trials";

* Write global macro variable array ;
%estimate_full_global(
        array=vfull_rndata,
        subject=Patient,
        treatment=Treatment,
        data=WORK.rndata
)

proc mixed
                data=WORK.rndata
                alpha=0.05 /* level for CI */
                cl /* CI for covariance parameters */
                method=REML; * REML method for estimation ;
        class Treatment (ref=first) Patient Period;
        model Y = Treatment /
                alpha=0.05 /* level for CI */
                cl /* t-type CI for fixed effects */
                htype=2 /* type 2 hypothesis test */
                solution; * solution for fixed effects ;
        random Patient Patient*Treatment;
        repeated Period /
                subject=Patient
                type=arma(1,1)
                r
                rcorr;
        %estimate_full(array=vfull_rndata, options=alpha=0.05 cl)
run;

* Delete global macro variable array ;
%estimate_full_free(array=vfull_rndata)

proc datasets lib=WORK nolist;
        delete rndata;
quit;
```

The specification of two titles in the top of Program 16.4 facilitates the localization of the program’s results in the SAS output. The %estimate_full_global macro is used to obtain information from the “WORK.rndata” dataset, which is required later by the %estimate_full macro. The %estimate_full_global macro saves this information into the “vfull_rndata” macro variable array, which is later supplied as a parameter to the %estimate_full macro. The ‘mixed ’procedure from Program 16.4 is very similar to the one from Program 16.2. Only two differences are evident, the usage of a different dataset and a different macro variable array name in the %estimate_full macro. When Program 16.4 runs, the %estimate_full macro is expanded into all the possible ‘estimate’ statements leading to the individual treatment effect estimates in the output. Note the same ‘model’, ‘random’ and ‘repeated’ statements are present in both Program 16.2 and Program 16.4. Therefore, the same first-order autoregressive moving-average linear mixed-effects model is fitted in both programs. However, Program 16.4 fits the model to the “WORK.rndata” dataset obtained before by running Program 16.3. After the ‘mixed’ procedure is executed, the %estimate_full_free macro is invoked to remove the “vfull_rndata” macro variable array from the system. There is no intention in accessing the data from the “vfull_rndata” macro variable array, past the execution of the program, so there is no need in keeping it in the computer’s memory. The ‘datasets’ procedure is executed lastly to remove the “rndata” dataset from the “WORK” library. The “WORK.rndata” dataset is created for this demonstration only, so keeping it afterwards is unnecessary. The “WORK.rndata” dataset is obtained from minor modifications done to “WORK.ndata”, therefore the class level information and the number of observations tables are still the same as the ones outputted from the ‘mixed’ procedures used in Program 16.1 and Program 16.2. As such, Program 16.4 outputs Table 16.2 and Table 16.4, presented above. The same model as Program 16.2 is fitted in Program 16.4, so the dimensions table resulting from both programs is the same. Thus, Table 16.14 is obtained once more. The tables with distinct information are presented below.

Table 16.23: Model information.

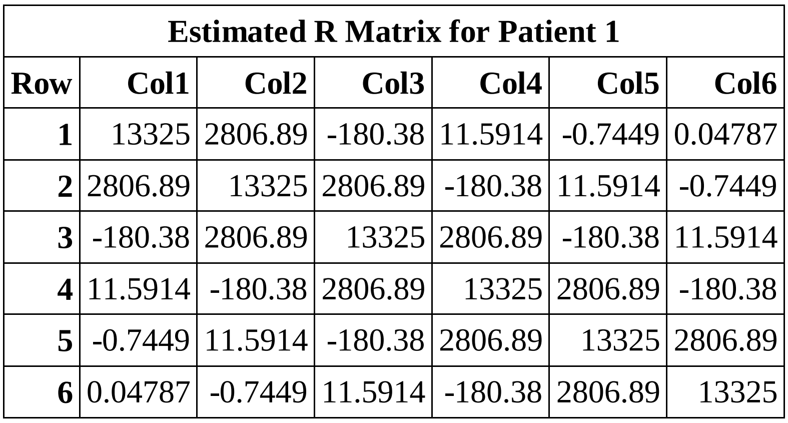
Model Information	
Data Set	WORK.RNDATA
Dependent Variable	Y
Covariance Structures	Variance Components, Autoregressive Moving Average
Subject Effect	Patient
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Table 16.24: Iteration history.



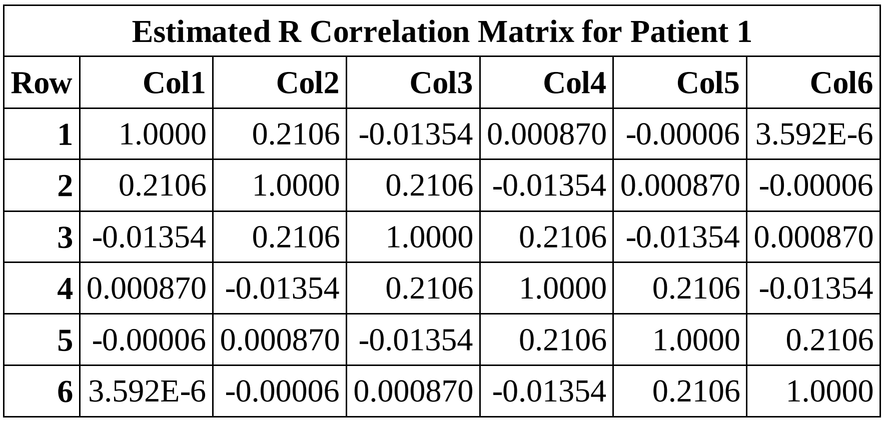
Iteration History			
Iteration	**Evaluations**	**-2 Res Log Like**	**Criterion**
0	1	931.31645659	
1	3	892.44161138	0.00001549
2	1	892.43547861	0.00000014
3	1	892.43542189	0.00000000

Table 16.25: Estimated R matrix for patient 1.



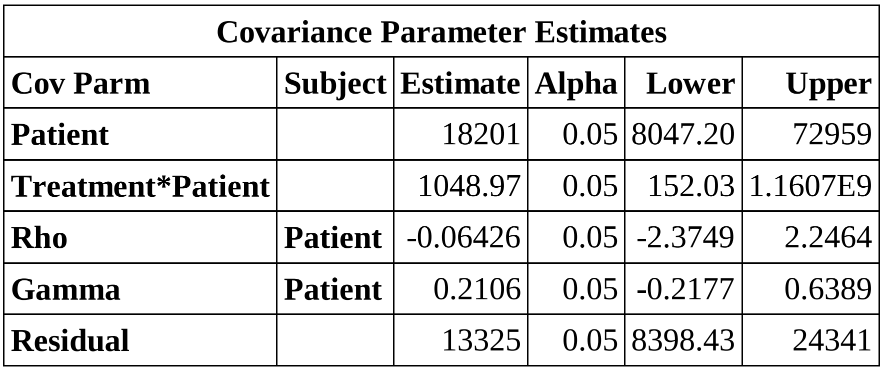
Estimated R Matrix for Patient 1						
Row	**Col1**	**Col2**	**Col3**	**Col4**	**Col5**	**Col6**
1	13325	2806.89	-180.38	11.5914	-0.7449	0.04787
2	2806.89	13325	2806.89	-180.38	11.5914	-0.7449
3	-180.38	2806.89	13325	2806.89	-180.38	11.5914
4	11.5914	-180.38	2806.89	13325	2806.89	-180.38
5	-0.7449	11.5914	-180.38	2806.89	13325	2806.89
6	0.04787	-0.7449	11.5914	-180.38	2806.89	13325

Table 16.26: Estimated R correlation matrix for patient 1.



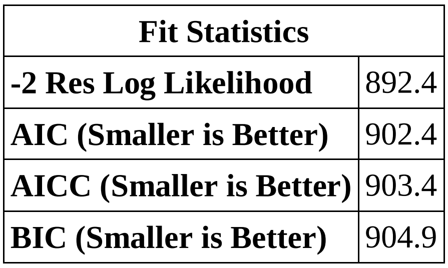
Estimated R Correlation Matrix for Patient 1						
Row	**Col1**	**Col2**	**Col3**	**Col4**	**Col5**	**Col6**
1	1.0000	0.2106	-0.01354	0.000870	-0.00006	3.592E-6
2	0.2106	1.0000	0.2106	-0.01354	0.000870	-0.00006
3	-0.01354	0.2106	1.0000	0.2106	-0.01354	0.000870
4	0.000870	-0.01354	0.2106	1.0000	0.2106	-0.01354
5	-0.00006	0.000870	-0.01354	0.2106	1.0000	0.2106
6	3.592E-6	-0.00006	0.000870	-0.01354	0.2106	1.0000

Table 16.27: Covariance parameter estimates.



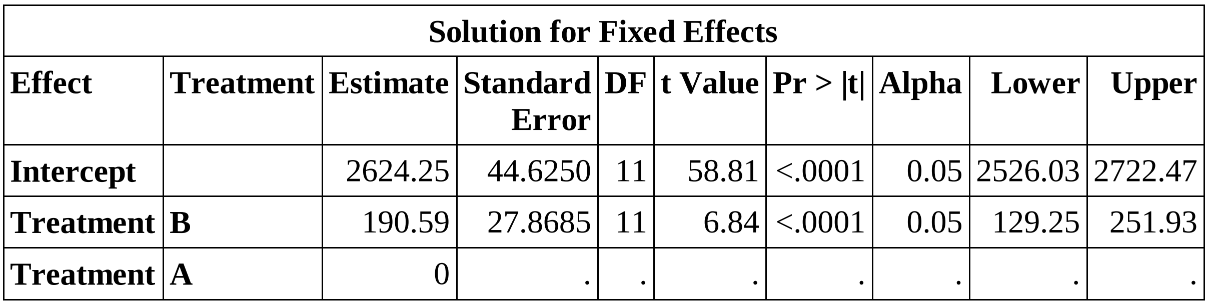
Covariance Parameter Estimates					
Cov Parm	**Subject**	**Estimate**	**Alpha**	**Lower**	**Upper**
Patient		18201	0.05	8047.20	72959
Treatment*Patient		1048.97	0.05	152.03	1.1607E9
Rho	**Patient**	-0.06426	0.05	-2.3749	2.2464
Gamma	**Patient**	0.2106	0.05	-0.2177	0.6389
Residual		13325	0.05	8398.43	24341

Table 16.28: Fit statistics.



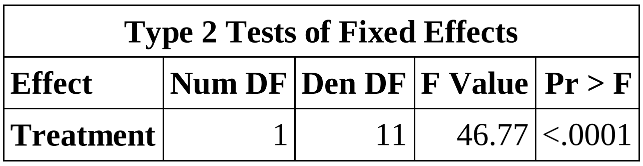
Fit Statistics	
-2 Res Log Likelihood	892.4
AIC (Smaller is Better)	902.4
AICC (Smaller is Better)	903.4
BIC (Smaller is Better)	904.9

Table 16.29: Solution for fixed effects.



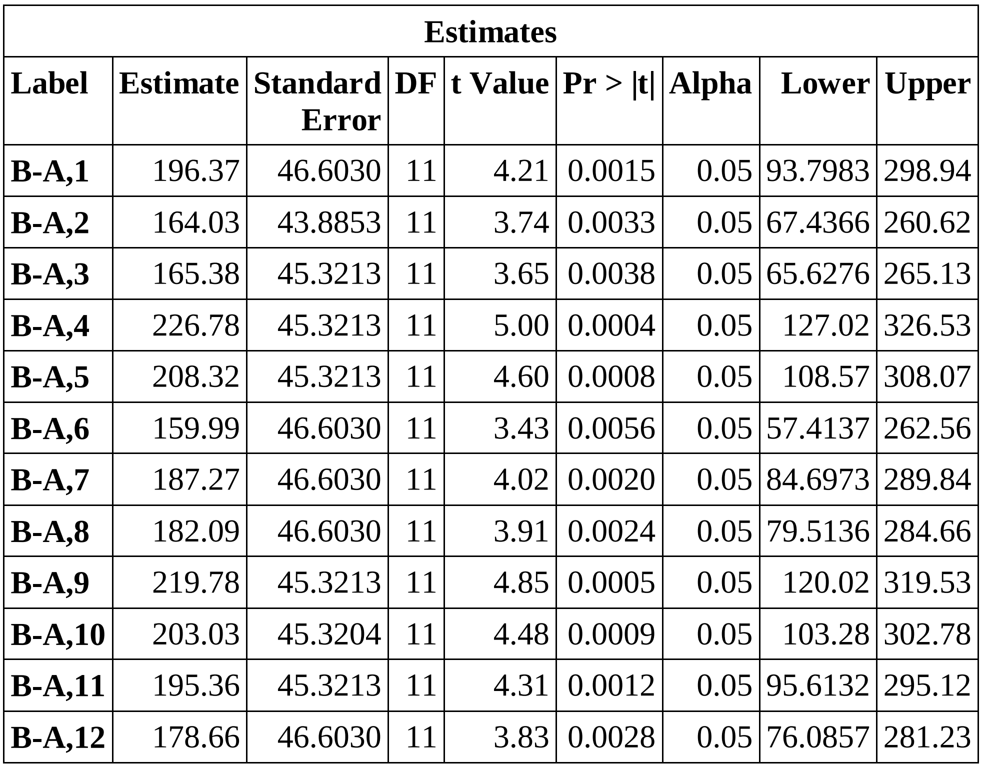
Solution for Fixed Effects									
Effect	**Treatment**	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Alpha**	**Lower**	**Upper**
Intercept		2624.25	44.6250	11	58.81	<.0001	0.05	2526.03	2722.47
Treatment	**B**	190.59	27.8685	11	6.84	<.0001	0.05	129.25	251.93
Treatment	**A**	0	.	.	.	.	.	.	.

Table 16.30: Type 2 tests of fixed effects.



Type 2 Tests of Fixed Effects				
Effect	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Treatment	1	11	46.77	<.0001

Table 16.31: Shrunk individual treatment effect estimates.



Estimates								
Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
B-A,1	196.37	46.6030	11	4.21	0.0015	0.05	93.7983	298.94
B-A,2	164.03	43.8853	11	3.74	0.0033	0.05	67.4366	260.62
B-A,3	165.38	45.3213	11	3.65	0.0038	0.05	65.6276	265.13
B-A,4	226.78	45.3213	11	5.00	0.0004	0.05	127.02	326.53
B-A,5	208.32	45.3213	11	4.60	0.0008	0.05	108.57	308.07
B-A,6	159.99	46.6030	11	3.43	0.0056	0.05	57.4137	262.56
B-A,7	187.27	46.6030	11	4.02	0.0020	0.05	84.6973	289.84
B-A,8	182.09	46.6030	11	3.91	0.0024	0.05	79.5136	284.66
B-A,9	219.78	45.3213	11	4.85	0.0005	0.05	120.02	319.53
B-A,10	203.03	45.3204	11	4.48	0.0009	0.05	103.28	302.78
B-A,11	195.36	45.3213	11	4.31	0.0012	0.05	95.6132	295.12
B-A,12	178.66	46.6030	11	3.83	0.0028	0.05	76.0857	281.23

Information pertaining to the model is displayed in Table 16.23, where one can see that the “WORK.rndata” dataset was used, and the autoregressive moving average and variance components covariance structures are modelled. The estimation algorithm converged to a solution matching the convergence criteria in three iterations, according to Table 16.24. The covariance matrix of the residual errors for patient 1 is represented by Table 16.25, while the related correlation matrix of the residual errors for the same patient 1 is displayed on Table 16.26. For the model in question, the covariance matrix of the residual errors is the same for all the patients in the series of n-of-1 trials. Likewise, the correlation matrix of the residual errors is the same for all the patients. For that reason, displaying the covariance and correlation matrices of the residual errors for only one patient is sufficient. Table 16.27 reveals a nonzero estimate of the variance of the treatment by patient random term. While the estimate of the variance of the treatment by patient random term for the non-randomized dataset is zero, according to Table 16.18. Remember that the “WORK.rndata” dataset used in the ‘mixed’ procedure from Program 16.4 and leading to Table 16.27 contains the same information as the “NORMAL.ndata” dataset, which is used in Program 16.2 and results in Table 16.18. The difference lies in the order of the treatments, which is observed through time. Program 16.3 takes the “NORMAL.ndata” dataset, and simulates a randomization of treatments in cycles within patients leading to the “WORK.rndata” dataset. Also, remember that the “NORMAL.ndata” dataset contains the same ABABAB sequence of treatments for all the twelve patients in the series of n-of-1 trials. Because of the repeated AB sequence of treatments over time in the “NORMAL.ndata” dataset, the treatment effect is confused with the time effect. The first-order autoregressive moving average linear mixed-effects model when fitted to the “NORMAL.ndata” dataset completely explains the random treatment by patient variation in the time dependent covariance structure of the residual error. Due to the simulated randomization of treatments in cycles within patients, the same AB sequence of treatments is not always repeated in time for the twelve patients in the series of n-of-1 trials represented by the “WORK.rndata” dataset. In this dataset the repetition of the AB sequence over time is prevented by the within patients treatment randomization procedure. The within patients treatment randomization procedure avoids confusion between treatment effect and time effect. When a first-order autoregressive moving average linear mixed-effects model is fitted to the “WORK.rndata” dataset, the random treatment by patient variation cannot be explained in the time dependent correlation structure of the residual error and the estimation of a treatment by patient random term is then possible. Look at Table 16.27 and observe that zero lies within the estimated confidence interval of the autoregressive parameter (Rho), and the moving-average component (Gamma) estimates, therefore the two parameter estimates are non-significant. The non-significance of these two parameter estimates suggests that the first-order autoregressive moving-average correlation structure does not fit the data better than the simpler variance components correlation structure. The same Table 16.27 suggests that the patient random term, the treatment by patient random term, and the residual error are significant. Note that zero is not contained within the confidence interval of none of these three covariance parameter estimates. The overall treatment effect estimate and associated statistics are found on the second row of Table 16.29. The p-value of the overall treatment effect estimate is very low when compared to the significance level, and zero is not contained within the confidence interval of the same estimate, suggesting that the treatments differ significantly between each other. The individual treatment effect estimates and respective statistics are presented in Table 16.31. Note that the individual treatment effect estimates are distinct across patients. A consequence of the nonzero estimate of the variance of the treatment by patient random term. Remember from Table 16.20 and Table 16.22 that the individual treatment effect estimates resulting from the similar first-order autoregressive moving-average linear mixed-effects model fitted to the non-randomized series of n-of-1 trials, are equal to the overall treatment effect for the twelve patients. The within patients treatment randomization procedure prevents confusion between treatment effect and time effect, and permits the estimation of distinct individual treatment effects from a linear mixed-effect model with a time dependent covariance structure and a treatment by patient random term. The estimates of the standard error of the individual treatment effect are not the same for the twelve patients, unlike what is observed in the results from the linear mixed-effects models discussed in sections 13, 14, and 15. When a time dependent correlation is modelled through an appropriate linear mixed-effects model, the estimates of the individual treatment effects and their respective standard error, depend on the treatment sequence resulting from the within patients treatment randomization procedure. In fact, when a time dependent correlation is modelled though a linear mixed-effect model, the estimates of the overall treatment effect and its standard error also depend on the sequence of treatments for each individual in the series of n-of-1 trials. If you try changing the value of the seed macro variable in Program 16.3, and run both Program 16.3 and Program 16.4, several times, you will get different sequences of treatments and different results across runs of the two programs. Provided that you use different values for the seed macro variable each time. The p-value attached to each individual treatment effect estimate is inferior to the chosen significance level, suggesting that the treatments differ significantly between each other individually for each patient. Also note that zero is not contained within none of the twelve confidence intervals of the individual treatment effect estimates, once more suggesting the individual significance of the difference between the treatments

Appendix A SAS macros

Macro A.1: estimate_full_global.

```
%macro estimate_full_global(
    array, /* macro variable array name */
    subject, /* variable indicating the subject */
    treatment, /* variable indicating the treatment */
    data, /* dataset name */
    ref=first /* reference treatment */
);
    %if not %length(&array) %then %do;
        %put ERROR: Value for macro parameter 'array' is missing with no default.;
        %return;
    %end;
    %else %if not %length(&subject) %then %do;
        %put ERROR: Value for macro parameter 'subject' is missing with no default.;
        %return;
    %end;
    %else %if not %length(&treatment) %then %do;
        %put ERROR: Value for macro parameter 'treatment' is missing with no default.;
        %return;
    %end;
    %else %if not %length(&data) %then %do;
        %put ERROR: Value for macro parameter 'data' is missing with no default.;
        %return;
    %end;
    %else %if not %sysfunc( exist(&data) ) %then %do;
        %put ERROR: Dataset %tslit(&data) does not exist.;
        %return;
    %end;
    %else %if not %length(&ref) %then %do;
        %put ERROR: Value for macro parameter 'ref' is missing with no default.;
        %return;
    %end;
    %global &array._s &array._t;
    %global dim_&array._s dim_&array._t;
    %let &array._s=&subject;
    %let &array._t=&treatment;
    proc sql noprint;
        select  distinct &&&array._s
                into :&array.p1 - :&array.p999999999
        from &data;
    quit;
    %if (&SQLRC ne 0) and (&SQLRC ne 4) %then %goto finish;
    %let dim_&array._s = &SqlObs; * save number of subjects ;
    proc sql noprint;
        select  distinct &&&array._t
                into :&array.t1 - :&array.t999999999
        from &data;
    quit;
    %if (&SQLRC ne 0) and (&SQLRC ne 4) %then %goto finish;
    %let dim_&array._t=&SqlObs; * save number of treatments ;
    %local i;
    %if &ref eq first %then %let i=1;
    %else %if &ref eq last %then %let i=&&dim_&array._t;
    %else %if &ref eq %sysfunc( dequote(&ref) ) %then %do;
        %put ERROR: Syntax error, expecting one of the following: a quoted string, FIRST, LAST.;
        %goto finish;
    %end;
    %else %do;
        %let ref=%sysfunc( dequote(&ref) );
        %do i=1 %to &&dim_&array._t %by 1;
            %if &&&array.t&i eq &ref %then %goto leave;
        %end;
        %leave:
        %if &i > &&dim_&array._t %then %do;
            %put ERROR: Value %tslit(&ref) not found in variable %tslit(&treatment) within
dataset %tslit(&data).;
            %goto finish;
        %end;
    %end;
    %local j k m;
    %let j=&&&array.t&i;
    %do k=&i %to %eval(&&dim_&array._t-1) %by 1;
        %let m=%eval(&k+1);
        %let &array.t&k=&&&array.t&m;
    %end;
    %let &array.t&k=&j;
    %local f0 f1;
    %let k=1;
    %let i=&&dim_&array._t;
    %let f0=%sysfunc( cats(-1) );
    %do j=1 %to %eval(&i-1) %by 1;
        %local l&k;
        %let l&k=&&&array.t&j-&&&array.t&i;
        %let f1=&f0;
        %do m=&i %to %eval(&j+2) %by -1;
            %let f1=0 &f1;
        %end;
        %let f1=1 &f1;
        %do %while (&m > 2);
            %let f1=0 &f1;
            %let m=%eval(&m-1);
        %end;
        %global &array._f&k;
        %let &array._f&k=&f1;
        %let k=%eval(&k+1);
    %end;
    %do i=1 %to %eval(&&dim_&array._t-1) %by 1;
```

```
        %let f0=%sysfunc( cats(-1) );
        %do m=1 %to %eval(&i-1) %by 1;
            %let f0=0 &f0;
        %end;
        %do j=%eval(&i+1) %to %eval(&&dim_&array._t-1) %by 1;
            %local l&k;
            %let l&k=&&&array.t&j-&&&array.t&i;
            %let f1=&f0;
            %do m=&i %to %eval(&j-2) %by 1;
                %let f1=&f1 0;
            %end;
            %let f1=&f1 1;
            %do %while( &m < %eval(&&dim_&array._t-1) );
                %let f1=&f1 0;
                %let m=%eval(&m+1);
            %end;
            %global &array._f&k;
            %let &array._f&k=&f1;
            %let k=%eval(&k+1);
        %end;
    %end;
    %local s r0 r1 r2;
    %let r0=%sysfunc( cats(0) );
    %do i=2 %to &&dim_&array._s %by 1;
        %let r0=0 &r0;
    %end;
    %do s=1 %to &&dim_&array._s %by 1;
        %let r1=%sysfunc( cats(1) );
        %let r2=%sysfunc( cats(-1) );
        %do i=2 %to &s %by 1;
            %let r1=0 &r1;
            %let r2=0 &r2;
        %end;
        %do %while(&i <= &&dim_&array._s);
            %let r1=&r1. 0;
            %let r2=&r2. 0;
            %let i=%eval(&i+1);
        %end;
        %let k=1;
        %let i=&&dim_&array._t;
        %let f0=&r2;
        %do j=1 %to %eval(&i-1) %by 1;
            %global &array._l%eval(&s+(&k-1)*&&dim_&array._s);
            %let &array._l%eval(&s+(&k-1)*&&dim_&array._s) = &&l&k,&&&array.p&s;
            %let f1=&f0;
            %do m=&i %to %eval(&j+2) %by -1;
                %let f1=&r0 &f1;
            %end;
            %let f1=&r1 &f1;
            %do %while (&m > 2);
```

```
                %let f1=&r0 &f1;
                %let m=%eval(&m-1);
            %end;
            %global &array._r%eval(&s+(&k-1)*&&dim_&array._s);
            %let &array._r%eval(&s+(&k-1)*&&dim_&array._s)=&f1;
            %let k=%eval(&k+1);
        %end;
        %do i=1 %to %eval(&&dim_&array._t-1) %by 1;
            %let f0=&r2;
            %do m=1 %to %eval(&i-1) %by 1;
                %let f0=&r0 &f0;
            %end;
            %do j=%eval(&i+1) %to %eval(&&dim_&array._t-1) %by 1;
                %global &array._l%eval(&s+(&k-1)*&&dim_&array._s);
                %let        &array._l%eval(&s+(&k-1)*&&dim_&array._s)        =
&&l&k,&&&array.p&s;
                %let f1=&f0;
                %do m=&i %to %eval(&j-2) %by 1;
                    %let f1=&f1 &r0;
                %end;
                %let f1=&f1 &r1;
                %do %while( &m < %eval(&&dim_&array._t-1) );
                    %let f1=&f1 &r0;
                    %let m=%eval(&m+1);
                %end;
                %global &array._r%eval(&s+(&k-1)*&&dim_&array._s);
                %let &array._r%eval(&s+(&k-1)*&&dim_&array._s)=&f1;
                %let k=%eval(&k+1);
            %end;
        %end;
    %end;
    %let dim_&array._t=%eval(&k-1); * save number of treatment comparisons ;
    %return;
    %finish:
    %symdel &array._s &array._t;
    %symdel dim_&array._s dim_&array._t;
    %return;
%mend estimate_full_global;
```

Macro A.2: estimate_full.

```
%macro estimate_full(
        array, /* macro variable array name */
        options=, /* estimate statement options */
        random=| /* leave undefined for fixed effects models */
);
        %if not %length(&array) %then %do;
                %put ERROR: Value for macro parameter 'array' is missing with no default.;
                %abort;
        %end;
        %else %if ( %length(&random) and %superq(random) ne %str(|) ) %then %do;
                %put ERROR: Syntax error, expecting one of the following: |, missing.;
                %abort;
        %end;
        %local i j;
        %do i=1 %to &&dim_&array._s %by 1;
                %do j=1 %to &&dim_&array._t %by 1;
                        %let k=%eval(&i+(&j-1)*&&dim_&array._s);
                        estimate "&&&array._l&k" &&&array._t &&&array._f&j &random
                                &&&array._s*&&&array._t &&&array._r&k / &options;
                %end;
        %end;
        %return;
%mend estimate_full;
```

Macro A.3: estimate_full_free.

```
%macro estimate_full_free(
        array /* macro variable array name */
);
        %if not %length(&array) %then %do;
                %put ERROR: Value for macro parameter 'array' is missing with no default.;
                %return;
        %end;
        %symdel &array._s &array._t;
        %local i;
        %do i=1 %to &&dim_&array._t %by 1;
                %symdel &array._f&i;
        %end;
        %local j;
        %do i=1 %to &&dim_&array._s %by 1;
                %do j=1 %to &&dim_&array._t %by 1;
                        %symdel &array._l%eval(&i+(&j-1)*&&dim_&array._s);
                        %symdel &array._r%eval(&i+(&j-1)*&&dim_&array._s);
                %end;
        %end;
        %symdel dim_&array._s dim_&array._t;
        %return;
%mend estimate_full_free
```

Macro A.4: estimate_full_random_global.

```
%macro estimate_full_random_global(
    array, /* macro variable array name */
    subject, /* variable indicating the subject */
    treatment, /* variable indicating the treatment */
    data, /* dataset name */
    ref=first /* reference treatment */
);
    %if not %length(&array) %then %do;
        %put ERROR: Value for macro parameter 'array' is missing with no default.;
        %return;
    %end;
    %else %if not %length(&subject) %then %do;
        %put ERROR: Value for macro parameter 'subject' is missing with no default.;
        %return;
    %end;
    %else %if not %length(&treatment) %then %do;
        %put ERROR: Value for macro parameter 'treatment' is missing with no default.;
        %return;
    %end;
    %else %if not %length(&data) %then %do;
        %put ERROR: Value for macro parameter 'data' is missing with no default.;
        %return;
    %end;
    %else %if not %sysfunc( exist(&data) ) %then %do;
        %put ERROR: Dataset %tslit(&data) does not exist.;
        %return;
    %end;
    %else %if not %length(&ref) %then %do;
        %put ERROR: Value for macro parameter 'ref' is missing with no default.;
        %return;
    %end;
    %global &array._s &array._t;
    %global dim_&array._s dim_&array._t;
    %let &array._s=&subject;
    %let &array._t=&treatment;
    proc sql noprint;
        select    distinct &&&array._s
                into :&array.p1 - :&array.p999999999
        from &data;
    quit;
    %if (&SQLRC ne 0) and (&SQLRC ne 4) %then %goto finish;
    %let dim_&array._s = &SqlObs; * save number of subjects ;
    proc sql noprint;
        select    distinct &&&array._t
                into :&array.t1 - :&array.t999999999
        from &data;
    quit;
    %if (&SQLRC ne 0) and (&SQLRC ne 4) %then %goto finish;
    %let dim_&array._t=&SqlObs; * save number of treatments ;
    %local i;
    %if &ref eq first %then %let i=1;
    %else %if &ref eq last %then %let i=&&dim_&array._t;
    %else %if &ref eq %sysfunc( dequote(&ref) ) %then %do;
        %put ERROR: Syntax error, expecting one of the following: a quoted string, FIRST, LAST.;
        %goto finish;
    %end;
    %else %do;
        %let ref=%sysfunc( dequote(&ref) );
        %do i=1 %to &&dim_&array._t %by 1;
            %if &&&array.t&i eq &ref %then %goto leave;
        %end;
        %leave:
        %if &i > &&dim_&array._t %then %do;
            %put ERROR: Value %tslit(&ref) not found in variable %tslit(&treatment) within dataset %tslit(&data).;
            %goto finish;
        %end;
    %end;
    %local j k m;
    %let j=&&&array.t&i;
    %do k=&i %to %eval(&&dim_&array._t-1) %by 1;
        %let m=%eval(&k+1);
        %let &array.t&k=&&&array.t&m;
    %end;
    %let &array.t&k=&j;
    %local f0 f1;
    %let k=1;
    %let i=&&dim_&array._t;
    %let f0=%sysfunc( cats(-1) );
    %do j=1 %to %eval(&i-1) %by 1;
        %local l&k;
        %let l&k=&&&array.t&j-&&&array.t&i;
        %let f1=&f0;
        %do m=&i %to %eval(&j+2) %by -1;
            %let f1=0 &f1;
        %end;
        %let f1=1 &f1;
        %do %while (&m > 2);
            %let f1=0 &f1;
            %let m=%eval(&m-1);
        %end;
        %global &array._f&k;
        %let &array._f&k=&f1;
        %let k=%eval(&k+1);
    %end;
    %do i=1 %to %eval(&&dim_&array._t-1) %by 1;
        %let f0=%sysfunc( cats(-1) );
        %do m=1 %to %eval(&i-1) %by 1;
```

```
                        %let f0=0 &f0;
                %end;
                %do j=%eval(&i+1) %to %eval(&&dim_&array._t-1) %by 1;
                        %local l&k;
                        %let l&k=&&&array.t&j-&&&array.t&i;
                        %let f1=&f0;
                        %do m=&i %to %eval(&j-2) %by 1;
                                %let f1=&f1 0;
                        %end;
                        %let f1=&f1 1;
                        %do %while( &m < %eval(&&dim_&array._t-1) );
                                %let f1=&f1 0;
                                %let m=%eval(&m+1);
                        %end;
                        %global &array._f&k;
                        %let &array._f&k=&f1;
                        %let k=%eval(&k+1);
                %end;
        %end;
        %local s r;
        %let r=%sysfunc( cats(1) );
        %do s=1 %to &&dim_&array._s %by 1;
                %let k=1;
                %let i=&&dim_&array._t;
                %do j=1 %to %eval(&i-1) %by 1;
                        %global &array._l%eval(&s+(&k-1)*&&dim_&array._s);
                        %let &array._l%eval(&s+(&k-1)*&&dim_&array._s) = &&l&k,&&&array.p&s;
                        %let k=%eval(&k+1);
                %end;
                %do i=1 %to %eval(&&dim_&array._t-1) %by 1;
                        %do j=%eval(&i+1) %to %eval(&&dim_&array._t-1) %by 1;
                                %global &array._l%eval(&s+(&k-1)*&&dim_&array._s);
                                %let            &array._l%eval(&s+(&k-1)*&&dim_&array._s)               =
&&l&k,&&&array.p&s;
                                %let k=%eval(&k+1);
                        %end;
                %end;
                %global &array._r&s;
                %let &array._r&s=&r;
                %let r=0 &r;
        %end;
        %let dim_&array._t=%eval(&k-1); * save number of treatment comparisons ;
        %return;
        %finish:
        %symdel &array._s &array._t;
        %symdel dim_&array._s dim_&array._t;
        %return;
%mend estimate_full_random_global;
```

Macro A.5: estimate_full_random.

```
%macro estimate_full_random(
    array, /* macro variable array name */
    options= /* estimate statement options */
);
    %if not %length(&array) %then %do;
        %put ERROR: Value for macro parameter 'array' is missing with no default.;
        %abort;
    %end;
    %local i j;
    %do i=1 %to &&dim_&array._s %by 1;
        %do j=1 %to &&dim_&array._t %by 1;
            %let k=%eval(&i+(&j-1)*&&dim_&array._s);
            estimate "&&&array._l&k" &&&array._t &&&array._f&j |
                                    &&&array._t &&&array._f&j /
                                        subject &&&array._r&i
                                        &options;
        %end;
    %end;
    %return;
%mend estimate_full_random;
```

Macro A.6: estimate_full_random_free.

```
%macro estimate_full_random_free(
    array /* macro variable array name */
);
    %if not %length(&array) %then %do;
        %put ERROR: Value for macro parameter 'array' is missing with no default.;
        %return;
    %end;
    %symdel &array._s &array._t;
    %local i;
    %do i=1 %to &&dim_&array._t %by 1;
        %symdel &array._f&i;
    %end;
    %local j;
    %do i=1 %to &&dim_&array._s %by 1;
        %do j=1 %to &&dim_&array._t %by 1;
            %symdel &array._l%eval(&i+(&j-1)*&&dim_&array._s);
        %end;
        %symdel &array._r&i;
    %end;
    %symdel dim_&array._s dim_&array._t;
    %return;
%mend estimate_full_random_free;
```

Macro A.7: estimate_diff_global.

```
%macro estimate_diff_global(
        array, /* macro variable array name */
        subject, /* variable indicating the subject */
        data /* dataset name */
);
        %if not %length(&array) %then %do;
                %put ERROR: Value for macro parameter 'array' is missing with no default.;
                %return;
        %end;
        %else %if not %length(&subject) %then %do;
                %put ERROR: Value for macro parameter 'subject' is missing with no default.;
                %return;
        %end;
        %else %if not %length(&data) %then %do;
                %put ERROR: Value for macro parameter 'data' is missing with no default.;
                %return;
        %end;
        %else %if not %sysfunc( exist(&data) ) %then %do;
                %put ERROR: Dataset %tslit(&data) does not exist.;
                %return;
        %end;
        %global &array._p dim_&array._p; /* declare global macro variables */
        %let &array._p=&subject; /* read variable name into macro variable */
        proc sql noprint;
                select  distinct &&&array._p
                                into :&array.p1 - :&array.p999999999
                from &data; /* select unique individuals */
        quit;
        %if (&SQLRC ne 0) and (&SQLRC ne 4) %then %do;
                %symdel &array._p dim_&array._p;
                %return;
        %end;
        %let dim_&array._p = &SqlObs; /* count unique individuals */
        %local i r;
        %let i=1; %let r=%sysfunc( cats(1) ); /* initialize macro variables */
        %do %until(&i >= &&dim_&array._p); /* loop through the individuals */
                %global &array._l&i &array._r&i; /* declare global macro variables */
                %let &array._l&i=&&&array.p&i; /* read individual into macro variable */
                %let &array._r&i=&r; /* read string into macro variable */
                %let r=0 &r; /* concatenate string */
                %let i=%eval(&i+1); /* increase loop counter */
        %end;
        %global &array._l&i &array._r&i; /* declare global macro variables */
        %let &array._l&i=&&&array.p&i; /* read individual into macro variable */
        %let &array._r&i=&r; /* read string into macro variable */
        %return;
%mend estimate_diff_global;
```

Macro A.8: estimate_diff.

```
%macro estimate_diff(
    array, /* macro variable array name */
    options=, /* estimate statement options */
    random=|, /* leave undefined for fixed effects models */
    noint=0 /* no intercept */
);
    %if not %length(&array) %then %do;
        %put ERROR: Value for macro parameter 'array' is missing with no default.;
        %abort;
    %end;
    %else %if ( %length(&random) and %superq(random) ne %str(|) ) %then %do;
        %put ERROR: Syntax error, expecting one of the following: |, missing.;
        %abort;
    %end;
    %else %if not %length(&noint) %then %do;
        %put ERROR: Value for macro parameter 'noint' is missing with no default.;
        %abort;
    %end;
    %local i;
    %do i=1 %to &&dim_&array._p %by 1;
        estimate "&&&array._l&i" intercept %eval(^&noint) &random
            &&&array._p &&&array._r&i / &options;
    %end;
    %return;
%mend estimate_diff;
```

Macro A.9: estimate_diff_free.

```
%macro estimate_diff_free(
    array /* macro variable array name */
);
    %if not %length(&array) %then %do;
        %put ERROR: Value for macro parameter 'array' is missing with no default.;
        %return;
    %end;
    %local i;
    %do i=1 %to &&dim_&array._p %by 1;
        %symdel &array._l&i &array._r&i;
    %end;
    %symdel &array._p dim_&array._p;
    %return;
%mend estimate_diff_free;
```

Macro A.10: estimate_diff_random_global.

```
%macro estimate_diff_random_global(
    array, /* macro variable array name */
    subject, /* variable indicating the subject */
    data /* dataset name */
);
    %estimate_diff_global(
        array=&array,
        subject=&subject,
        data=&data
    )
    %return;
%mend estimate_diff_random_global;
```

Macro A.11: estimate_diff_random.

```
%macro estimate_diff_random(
    array, /* macro variable array name */
    options= /* estimate statement options */
);
    %if not %length(&array) %then %do;
        %put ERROR: Value for macro parameter 'array' is missing with no default.;
        %abort;
    %end;
    %local i;
    %do i=1 %to &&dim_&array._p %by 1;
        estimate "&&&array._l&i" intercept 1 |
                                        intercept 1 /
                                            subject &&&array._r&i
                                            &options;
    %end;
    %return;
%mend estimate_diff_random;
```

Macro A.12: estimate_diff_random_free.

```
%macro estimate_diff_random_free(
    array /* macro variable array name */
);
    %estimate_diff_free(array=&array)
    %return;
%mend estimate_diff_random_free;
```

References

1. Cody, R., *Learning SAS by Example: A Programmer's Guide*. 2007, Cary, North Carolina: SAS Institute.
2. Schreier, H., *PROC SQL by Example: Using SQL Within SAS*. 2008, Cary, North Carolina: SAS Institute.
3. Burlew, M.M., *Combining and Modifying SAS Data Sets: Examples*. 2nd ed. 2009, Cary, North Carolina: SAS Institute.
4. Burlew, M.M., *SAS Macro Programming Made Easy*. 3rd ed. 2014, Cary, North Carolina: SAS Institute.
5. Carpenter, A., *Carpenter's Complete Guide to the SAS Macro Language*. 3rd ed. 2016, Cary, North Carolina: SAS Institute.
6. Senn, S., *Cross-over Trials in Clinical Research*. 2nd ed. 2002, Chichester, England: John Wiley & Sons.
7. Cody, R., *Cody's Collection of Popular SAS Programming Tasks and How to Tackle Them*. 2012, Cary, North Carolina: SAS Institute.
8. Cody, R., *Longitudinal Data and SAS: A Programmer's Guide*. 2001, Cary, North Carolina: SAS Institute.
9. Carpenter, A., *Carpenter's Guide to Innovative SAS Techniques*. 2012, Cary, North Carolina: SAS Institute.
10. SAS Institute. *SAS Product Documentation*. [cited 2016 July 25]; Available from: http://support.sas.com/documentation/.
11. Zucker, D.R., et al., *Combining single patient (N-of-1) trials to estimate population treatment effects and to evaluate individual patient responses to treatment.* J Clin Epidemiol, 1997. **50**(4): p. 401-10.
12. Zucker, D.R., R. Ruthazer, and C.H. Schmid, *Individual (N-of-1) trials can be combined to give population comparative treatment effect estimates: methodologic considerations.* J Clin Epidemiol, 2010. **63**(12): p. 1312-23.
13. Rostoker, G., et al., *A pilot study of routine colloid infusion in hypotension-prone dialysis patients unresponsive to preventive measures.* J Nephrol, 2011. **24**(2): p. 208-17.
14. Rascol, O., et al., *A proof-of-concept, randomized, placebo-controlled, multiple cross-overs (n-of-1) study of naftazone in Parkinson's disease.* Fundam Clin Pharmacol, 2012. **26**(4): p. 557-64.
15. Lee, S., et al., *Validation Study of Kim's Sham Needle by Measuring Facial Temperature: An N-of-1 Randomized Double-Blind Placebo-Controlled Clinical Trial.* Evid Based Complement Alternat Med, 2012. **2012**: p. 507937.
16. Sniehotta, F.F., et al., *Testing self-regulation interventions to increase walking using factorial randomized N-of-1 trials.* Health Psychol, 2012. **31**(6): p. 733-7.
17. Tison, F., et al., *Simvastatin decreases levodopa-induced dyskinesia in monkeys, but not in a randomized, placebo-controlled, multiple cross-over ("n-of-1") exploratory trial of simvastatin*

against levodopa-induced dyskinesia in Parkinson's disease patients. Parkinsonism Relat Disord, 2013. **19**(4): p. 416-21.

18. Yuhong, H., et al., *An n-of-1 Trial Service in Clinical Practice: Testing the Effectiveness of Liuwei Dihuang Decoction for Kidney-Yin Deficiency Syndrome.* Evid Based Complement Alternat Med, 2013. **2013**: p. 827915.
19. Senn, S.J. and H. Hildebrand, *Crossover trials, degrees of freedom, the carryover problem and its dual.* Stat Med, 1991. **10**(9): p. 1361-74.
20. Araujo, A., S. Julious, and S. Senn, *Understanding Variation in Sets of N-of-1 Trials.* PLoS One, 2016. **11**(12): p. e0167167.
21. Brown, H. and R. Prescott, *Applied Mixed Models in Medicine*. 3rd ed. 2015, Chichester, England: John Wiley & Sons.
22. Pinheiro, J.C. and D.M. Bates, *Mixed-Effects Models in S and S-PLUS*. 2000, New York: Springer.
23. Diggle, P., et al., *Analysis of Longitudinal Data*. 2nd ed. 2002, Oxford, England: Oxford University Press.
24. Verbeke, G. and G. Molenberghs, *Linear Mixed Models for Longitudinal Data*. 2009, New York: Springer.
25. Kenward, M.G. and J.H. Roger, *Small sample inference for fixed effects from restricted maximum likelihood.* Biometrics, 1997. **53**(3): p. 983-97.
26. Senn, S., L. Stevens, and N. Chaturvedi, *Repeated measures in clinical trials: simple strategies for analysis using summary measures.* Stat Med, 2000. **19**(6): p. 861-77.
27. Fitzmaurice, G.M., N.M. Laird, and J.H. Ware, *Applied Longitudinal Analysis*. 2nd ed. 2011: John Wiley & Sons.
28. Tukey, J.W., *Exploratory Data Analysis*. 1977: Addison-Wesley Publishing Company.
29. Bartlett, M.S., *Properties of Sufficiency and Statistical Tests.* Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 1937. **160**(901): p. 268-282.
30. Brown, M.B. and A.B. Forsythe, *Robust Tests for the Equality of Variances.* Journal of the American Statistical Association, 1974. **69**(346): p. 364-367.
31. Littell, R.C., et al., *SAS for Mixed Models*. 2nd ed. 2006, Cary, North Carolina: SAS Institute.
32. Brockwell, P.J. and R.A. Davis, *Introduction to Time Series and Forecasting*. 2nd ed. 2002, New York: Springer.
33. Brockwell, P.J. and R.A. Davis, *Time Series: Theory and Methods*. 2nd ed. 2009, New York: Springer.
34. Wicklin, R., *Statistical Programming with SAS/IML Software*. 2010, Cary, North Carolina: SAS Institute.
35. Wicklin, R., *Simulating Data with SAS*. 2013, Cary, North Carolina: SAS Institute.