# Reproducible Research using R - A template for analysing behavioural experiments

*Tuomas Eerola* tuomas.eerola@durham.ac.uk, Durham University, UK

19/1/2015, update 12/12/2016

## Why reproducible research?

1. To comply with the increased demands for transparency and open access data, originally formulated within the computer sciences but later spread to biosciences and currently taking hold in social sciences (see Asendorpf et al., 2013; Tomasello & Call, 2011). We should aim to do all our statistical analyses transparently and in a reproducible fashion.
2. To collaborate more easily and effectively. This also helps to spot mistakes and encourage learning and trying out new things (Sandve et al., 2013).
3. To communicate your research more effectively by writing clear analysis paths that are able to produce the key statistics, figures and tables effortlessly. This also helps to gain visibility to research if shared fully (see Piwowar et al., 2007).

## Why R (and not Matlab or SPSS)?

Here I have chosen *R* to be the chosen tool for reproducible analyses, although of course statistical software could also be used. However, there are several good arguments to support R as the best choice. And I have long personal experience of SPSS and Matlab, both powerful but hampered by various design issues, but R has several advantages over these:

1. R is the most accessible software. R is free, open source, available for all operating systems. Matlab is great for certain type of work, but expensive, fussy about the operating systems, not to mention SPSS in this regard.
2. R is completely programming driven (thus fully transparent). Matlab is equaly so, but since it is essentially a *MATrix LABoratory*, it is very good for numerical analyses, but R is a little more versatile for strings and data structures more commonly used in statistics. SPSS also has the syntax option, but it is much more cryptic and unwieldy than R and Matlab. Clear syntax driven operation makes the analyses easily human readable, which is important for collaborations.
3. R has excellent coverage of statistical modelling tools. Thousands of R packages exist for any state-of-the-art statistical technique (bayesian, structural equation modelling, rare regression analytics, all machine-learning algorithms with effective implementations.
4. R is rational and even pedagogical in many of its functionalities (it warns about calculating means for categorical variables, is much more explicit about the outputs, and data.frames, etc.)
5. Good support for producing reports (R Markdown, knitr, Sweave) or interactive websites (Shiny).

## Analysis Template

I have prepared an analysis template, which contains examples of the whole process of data analysis; from loading to preprocessing data and analysing and reporting the results. I suggest a certain folder structure to keep the different parts of the processes tightly in different folders. I have been influenced by existing templates^fn-ProjectTemplate and style guides[^fn-acknowledgments], but it is basically the cleaned up version of the structures that I have for each project.

[^fn-acknowledgments]: Style Guide

### Suggested folder structure:

A project should have a dedicated folder with a descriptive name it. Within the folder, there is a master file called the `contents.R` which contains the necessary commands to load the data, pre-process it, analyse, and produce figures and tables. For clarity, it is good idea to keep things organised in particular special subfolders.

**config** : Configuration file ( `libraries.R` ), which define custom functions and loads the R libraries needed in the project.

**data** : Stores the original data, preferably in read-only format. Can be Excel or ascii files, or folders of separate files exported from experiment data collection interface.

**munge** : Munger refers to "data munging", which means cleaning and transforming the data to a suitable format for the analysis. If you have to recode (e.g. inverse) and combine items into instruments, this is the place to do it.

**scr** : Scripts used in the analysis. This is the main folder for the analysis. Examples of diagnostics, statistical testing, plotting, generating tables are given.

**figures** : All figures and graphs produced by analysis scripts (preferably in pdf format)

**reports** : Reports summarising the analysis, either manually or automatically. The latter can be done with R markdown.

## Using the Template

Once you have the template including the data and folder structure as well as *R* installed (or *RStudio*), it should be straightforward to proceed to using the template. The following example loads the first part of the Exp. 1 data in the Sweet Sorrow project containing background survey responses and self-reports given in the experiment from 42 participants (this data is a subset of the study that came out as

- Eerola, T., Vuoskoski, J. K., & Kautiainen, H. (2016). Being Moved by Unfamiliar Sad Music Is Associated with High Empathy. Frontiers in Psychology, 7, 1176. http://journal.frontiersin.org/article/10.3389/fpsyg.2016.01176

### 1. Initialise the analysis

Start R and open up the `contents.R` file using your preferred editor. Check that the directory after the first command `setwd` is pointing the location of your analysis directory and run the first lines of the code:

```
## INITIALISE: SET PATH, CLEAR MEMORY AND LOAD LIBRARIES
setwd('/Users/ptee/data_analysis/')  # set working directory to the project
rm(list=ls(all=TRUE))                # cleans the R memory, just in case
source('config/libraries.R')         # loads the extra R libraries needed
```

If you get errors at this stage with new installation of R, they might refer to the special libraries that were loaded or installed in `libraries.R` . This script should install the required libraries for you such as `ggplot2` , but there might be issues with your particular setup.

### 2. Load, preprocess and diagnose the data

Next, it is time to load the data with two scripts, the first one `read_data_survey.R` is simply reading an excel file stored in data folder whereas the `read_data_behavioural.R` reads separate ascii files exported from Matlab (and stored in a sub-folder under the data).

```
## READ DATA (Exp. 1 survey)
source('scr/read_data_survey.R')         # OUTPUT IS IN VARIABLE v
source('scr/read_data_behavioural.R')    # OUTPUT IS IN VARIABLE B
```

This should get variables v and B, which are data frames containing all the data. Next the raw responses are munged, that is, preprocessed in terms of the instruments (some items are reversed and then several items are combined). Missing values are imputed (with means), and when the items are combined, the reliability of the instruments is printed (Cronbach alpha). Finally the survey and self-report responses are combined into a single dataframe called `df` .

```
## MUNGE DATA (preprocess, recode, etc.)
source('munge/recode_instruments.R')     # This produces S data frame (Survey)
source('munge/merge_data.R')             # This combines Behavioural & Survey
```

After the munging, it is prudent to check some aspects of the data, N, age, gender in order to remind of the dataset properties. It is also useful plot the distributions of the collected data in order to learn whether one needs to use certain operations (transformation or non-parametric statistics) in the subsequent analyses. Correlations between selected instruments is also shown.

```
## DIAGNOSE data
source('scr/diagnose_simple.R')          # describe the data (N, etc.)
source('scr/diagnostic_plots.R')         # display histograms
```

If everything seems to be fine, it is time to proceed into the actual analysis.

### 3. Analyse the data

Finally we get to test the planned hypotheses of the experiment. Here the first analysis script builds a linear model for explaining the appreciation of sad music ( `ASM_appreciation` ) with other background variables (emotional contagion, nostalgia-proneness, quality of life, musicianship, etc.) in a shape of linear regression. It also gives a simple example of exploring whether this same variable is different across gender and education. These are just examples which do not actually deliver deep answers to the questions.

```
source('scr/linear_models.R')            # simple regression and ANOVA examples
```

Another idea is to attempt to predict those who have been highly influenced by the experiment in terms of giving high sadness ratings and high liking for these as well, whom we can call these "sad music lovers". The opposite group, who disliked the music and do not report any sadness could be called "sad music haters". We first define these by quantiles of the self-reports, then try to predict the members into three groups (lovers, neutrals, and haters) from the background variables using logistic regression.

```
source('scr/categorical_models.R')       # predict groupings
```

Let's visualise the values of the liking and feeling sad for the three groups. For clarity, it is a good idea to put the plots in separate scripts.

```
source('scr/figure1.R',print.eval=TRUE)  # create Figure 1 for manuscript
```

One can also produce tables in the same way using a simple script. Here is an example of how to report the means and standard deviations of the background variables grouped according to the simple median split of the Liking for Sad Music (LSMS) variable. P-values from ANOVAs are thrown in as a bonus.

```
source('scr/table1.R')            # create Table 1 for manuscript
```

Happy exploring. The intention is carrying out the analysis this way is to get a clear sense of the process and deliver outputs of the analyses that are easy to bring to the manuscript, and which also should be transparent for the other readers (supervisors and collaborators).

### 4. Combine report and analysis (optional)

It is also possible to combine the reporting of the analysis and the actual analysis to make the process even more transparent. An example of this can be found in `report.Rmd` , which basically runs the steps in the example tempate in sequence within a particular syntax (R md, using `knitr` , and this also creates a pdf or htlm report to the same folder (take a loot at the `report.pdf` ), which can contain all sorts of written arguments, comments and so on. In order to use this option, the `report.Rmd` should be on the project folder (not within reports subfolder) and your machine should have additional libraries installed (at least *LaTeX* and *knitr*, see knitr in a nutshell).

*Update in December 2016*: RStudio now supports **R Notebooks**, which combine the scripts and reports in a very nice fashion. See `report_NB.Rmd` for an example.

# Help for statistics with R

## Online tutorials

An Introduction to R : The official guifance from *The Comprehensive R Archive Network (CRAN)*. May not be always the most compelling introduction but exhaustive at least.

Quick-R : Really good source of R examples for almost all operations (manipulation, representation, functions, syntax, stats, figures, etc.).

R tutorials : Another fairly clear collection of tutorials.

RStudio online learning pages : R Studio is fancy and great visual GUI on top of the R for all platforms and they have released very useful documentations, tutorials, demos, etc.

Advanced R : Author of the best packages, Hadley Wickham, has created this resource (nook and online version).

## Statistics Handbooks with complete R scripts (online)

Practical Regression and Anova using R : A handbook of the basic statistical operations written by Julian Faraway.

Data Analysis and Graphics Using R - An Example-Based Approach : Handbook in 3rd printing, written by John Maindonald and John Braun. This source contains exercises, slides, the scripts for all graphs of the book, etc.

## Other Online Resources

R blogger : Multipurpose source for news and latest issues in R.

Collection of Resources at CRAN : Large collection of different resources (e.g. R for Matlab-minded, Fitting Distributions with R, Reference Cards, Data-mining with R, and so on).

R Documentation : Searchable online documentation

StackOverflow : Forum of questions and answers about computer programming, including R. Contains over 40,000 questions related to R.

## References

- Asendorpf, J. B., Conner, M., De Fruyt, F., De Houwer, J., Denissen, J. J., Fiedler, K., Fiedler, S., Funder, D. C., Kliegl, R., Nosek, B. A., & others (2013). Recommendations for increasing replicability in psychology. European Journal of Personality, 27(2), 108--119.
- Piwowar, H. A., Day, R. S., & Fridsma, D. B. (2007). Sharing detailed research data is associated with increased citation rate. PloS one, 2(3), e308.
- Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. PLoS computational biology, 9(10), e1003285.
- Tomasello, M. & Call, J. (2011). Methodological challenges in the study of primate cognition. Science, 334(6060), 1227--1228.

This document is available in GitHub: https://github.com/tuomaseerola/ReproR