R_template - A Template for Reproducible Analysis of Behavioural Data

Tuomas Eerola, Durham University

1/2/2021

Contents

Release notes

Created: 1/2/2021.

These files contains R template for analysing data from experiments and surveys and justification to follow certain conventions and structure. This document is available at https://github.com/tuomaseerola/R_template.

This can also be started as an independent process at Binder, [[Binder]](https://mybinder.org/v2/gh/tuomaseerola/R_template/main?urlpath=rstudio.

This repository and the documents are not a quick *R tutorial* nor *statistics tutorial* but simply a way to explain to PG students and collaborators of how clear analyses schemes can be created, followed, and shared.

1 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).

2 Why *R*?

Here I have chosen *R* (and *RStudio* as the smooth and handy front-end) to be the chosen tool for reproducible analyses, although of course any statistical software could also be used. However, there are several good arguments to support *R* as a good 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 these issues. Often the problem is not the price, universities can afford to have the licenses, but the skills learned through the software need to be used often in a different environment that might not have the same resources (arts organisation, startup company, etc.).

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, and many more).

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.) and allows to produce really easily understandable code with some extra libraries (`tidyr`, `ggplot2`,`psych`).

5. *R* has excellent support for producing reports in R Markdown or even for creating interactive websites using Shiny.

3 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[1] and style guides[2], but it is basically the cleaned up version of the structures that I have for each project.

3.1 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 a brief summary of the project, owner, status, and 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.

`/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 or from *Qualtrics* or *PsyToolkit*. Be wary of different encodings (UTF-8, Western, UTF-16, etc.) which is the usual cause of problems when reading in data. I would stress that we do not carry out any edits on the data, even if you find out that there are typos or mistakes in the data. Handling these in the next step makes the process replicable and transparent and documents what issues were fixed and how. If there are several versions of the data (original survey, and a small top-up), it is useful to use the date of the data retrieval or the N as a part of an informative filename (`Emotion_Identification_N119_noheader.tsv`).

`/munge` Munge folder 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., invert the scale for a question that has been asked with a reverse wording in comparison to other items, or relabel cryptic variable names from surveys) or combine items into indices of instruments, this is the place to do it. Sometimes filtering participants out that do not fit the criteria (missed too many questions, trials, or did not consent to the study etc.) can be done at this stage.

`/scr` R Scripts used in the analysis. This is the main folder that keeps all the interesting elements of the analysis. Examples of diagnostics, statistical testing, plotting, and generating tables are given.

[1] ProjectTemplate
[2] Style Guide

/figures All figures and graphs produced by analysis scripts (preferably in pdf format) can be stored here. The example scripts always write the graphics in this folder.

/reports This is often option, but if you create manual or automated reports summarising the analysis, this is the place to store them. The latter can be done with R markdown.

3.2 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. A copy of the template can be found at https://github.com/tuomaseerola/R_template.

3.2.1 Example data

The following example loads one dataset from Annaliese Micallef-Grimaud's study about perceived emotions in music. This was a experiment where 119 participants rated a small number of music examples using different emotion scales (Anger, Calmness, Fear, Joy, Power, Sadness, and Surprise). The ratings were done using likert scale of 1 to 5 (1=minimal and 5=maximal) and the music excerpts were composed to portray different emotions (angry, calm, scary, joyful, power, sad, surprising). The data was collected via Qualtrics and there is quite a lot of tidying up to do before this data can be analysed. The typical research questions would revolve around whether the tracks representing different emotions are differently in terms of the emotions and there are variants of the pieces from the past experiments that have either been created by the Annaliese (Exp. 1) or modified by participants in production study (Exp. 2), so another question is whether the two sources for the same piece differ in terms of their ratings. This is a validation part of the study that we have submitted to a journal together with some other data from production experiments (where people adjust the musical cues to produce different emotional expressions of the same pieces). We hope to get an actual reference for this data in near future.

You can grab the whole template (folder structures, R scripts, and Report.Rmd notebook and the data) from https://github.com/tuomaseerola/R_template.

3.2.2 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
rm(list=ls(all=TRUE))                  # Cleans the R memory, just in case
source('scr/load_libraries.R')         # Loads the necessary R libraries
```

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.

3.2.3 Load, preprocess and diagnose the data

Next, it is time to load the data with a scripts, the first one `read_data_survey.R` is simply reading an TSV file exported from Qualtrics stored in data folder. I've taken the second, descriptive header row out of the data to simply the process, but different datasets will have slightly different structures.

```
## READ data
source('scr/read_data_survey.R')      # Produces data frame v
```

```
## N x Variables:119 131
```

This should retrieve a data frame into a variable called `v` in *R*, which contains a complex data frame. In the next step this raw data will be munged, that is, pre-processed in several ways. Pre-processing can have multiple steps, here these have broken into two:

1. First operation carries out a long list of renaming the variables (columns in the data, `rename_variables.R`). This can be avoided if the data has these names already, and it is quite useful to try to embed meaningful variables names to the data collection (experiment or survey or manual coding).
2. Recoding instruments (`recode_instruments.R`) has several steps and it might be useful to study the steps separately. Finally the responses are reshaped into a form called long-form that is better suited for the analyses. This dataframe will be called `df`.

```
## MUNGE data (preprocess, recode, etc.)
source('munge/rename_variables.R')        # Renames the columns of the v
source('munge/recode_instruments.R')      # Produces df (long-form) from v
```

After the munging, it is prudent to check various aspects of the data.

1. Descriptives such as the N, age, gender are echoed in order to remind us of the dataset properties (`demographics_info.R`).
2. We can also explore the consistency of the ratings across the people to check whether people agreed on the ratings and generally understood the task (`interrater_reliability.R`).
3. We also want to look at the distributions of the collected data in order to learn whether one needs to use certain operations (transformations or resort to non-parametric statistics) in the subsequent analyses (`visualise.R`). This step will also include displaying correlations between the emotion scales which is a useful operation to learn about the overlap of the concepts used in the tasks.

```
## DIAGNOSE and VISUALISE data
source('scr/demographics_info.R')     # Reports N, Age and other details
```

```
## [1] "N = 91"
## [1] "Mean age 34.99"
## [1] "SD age 15.86"
## [1] "Youngest 18 years"
## [1] "Oldest 71 years"
##
##   Male Female  Other
##     23     67      1
##
##              NonMusician Music-Loving NonMusician                    Amateur
##                        13                        44                         15
## Serious Amateur Musician                 Semi-Pro                        Pro
##                        11                         6                          2
##
## Nonmusician    Musician
##          57          34
```

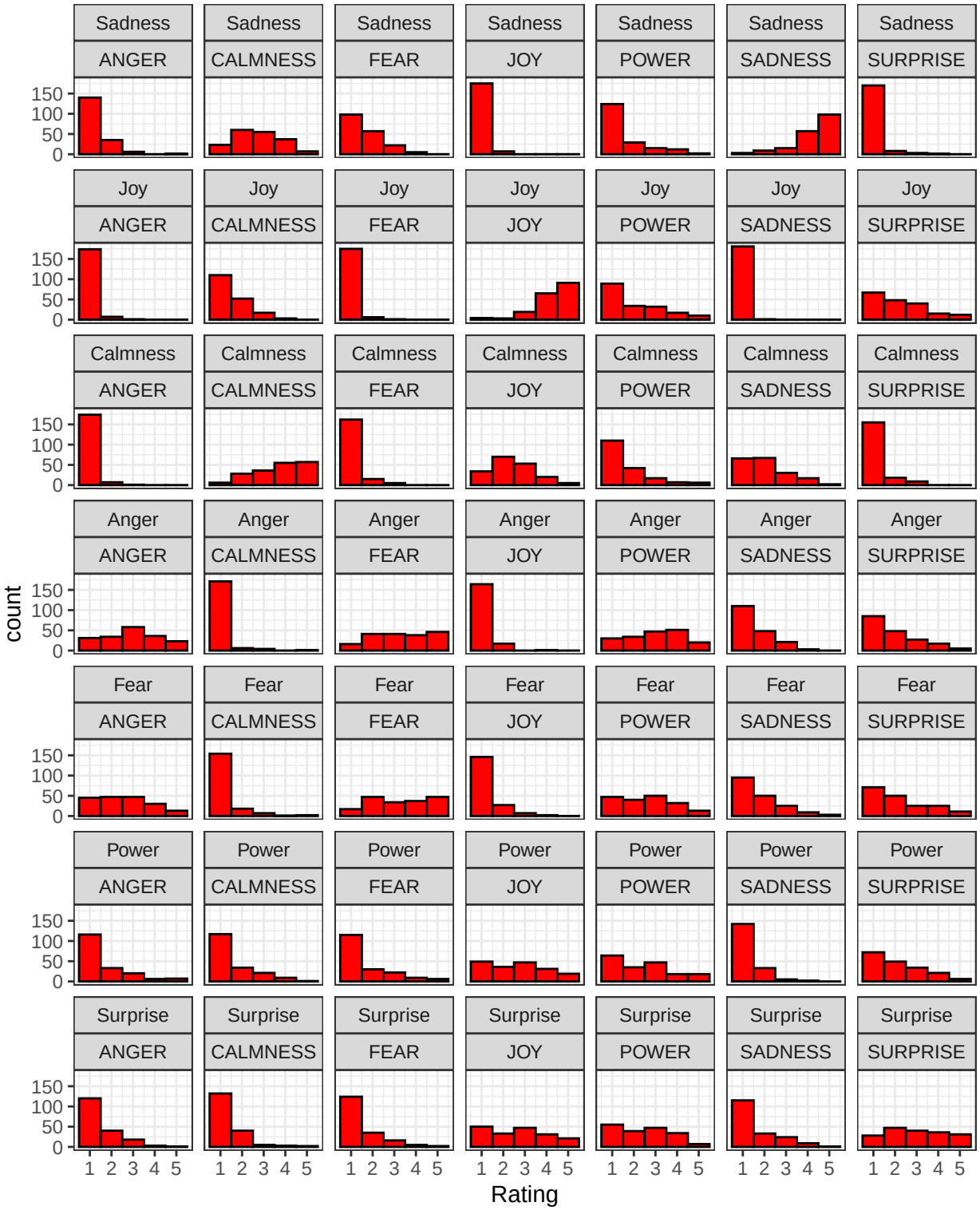
```
source('scr/interrater_reliability.R')# Quality checks, consistency check
```

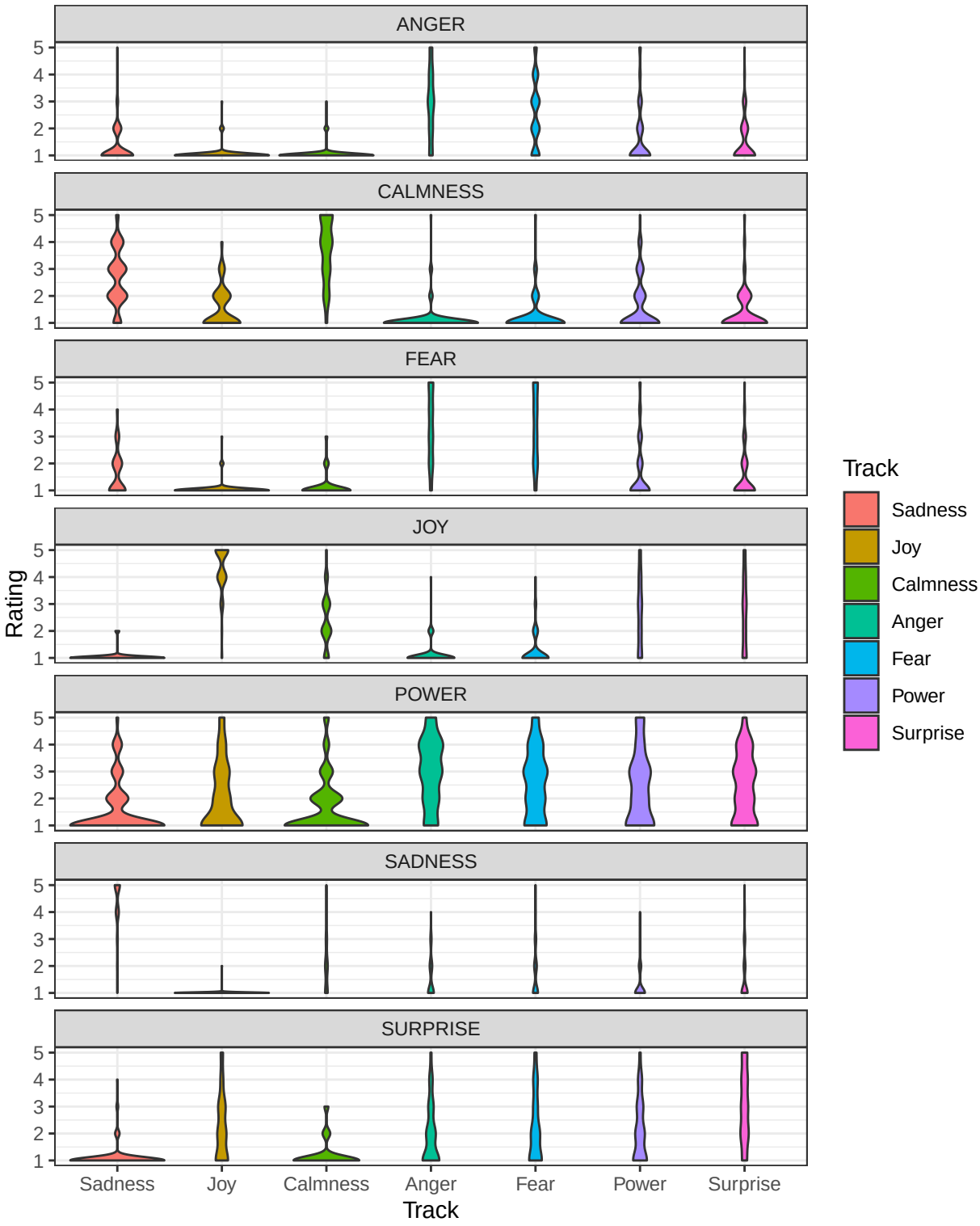
```
## [1] "Fastest response 7.17 mins"
## [1] "Slowest response 8291.48 mins"
## [1] "Median response 14.9 mins"
```

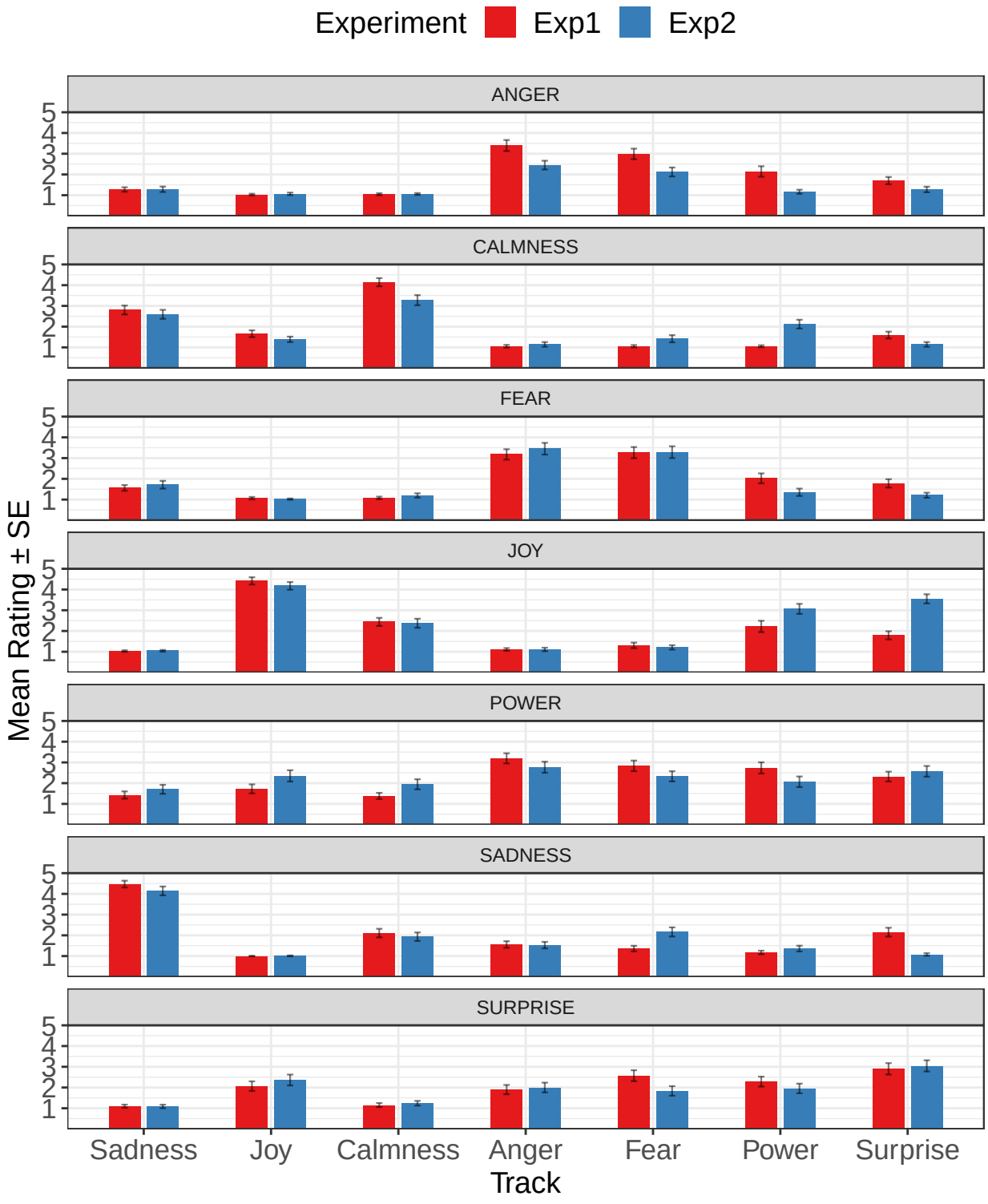

```
##
##
## Table: Inter-reliability ratings (Cronbach alphas)
##
## | SADNESS| CALMNESS|   JOY| ANGER| FEAR| POWER| SURPRISE|
```

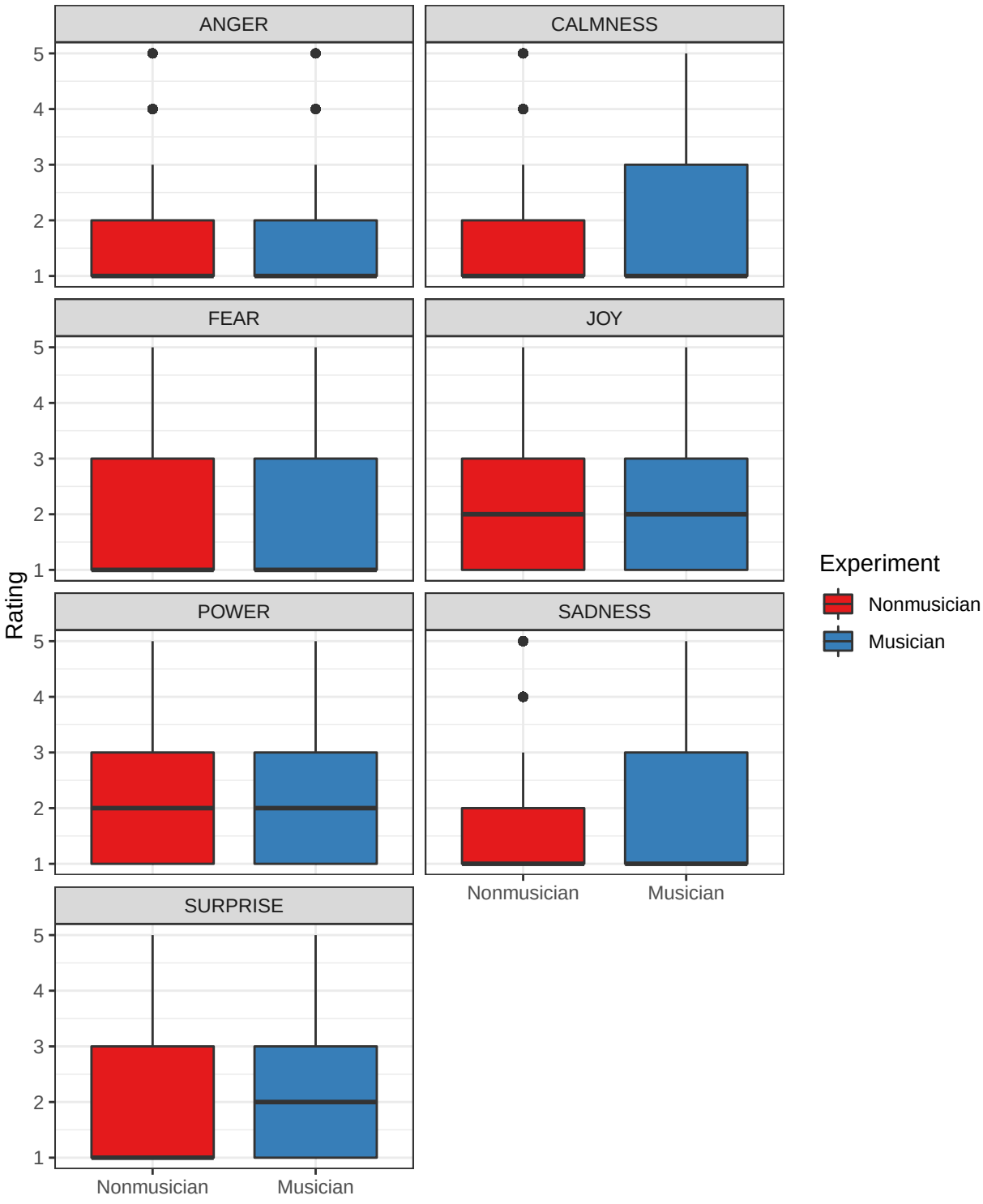
```
## |-------:|--------:|-----:|-----:|----:|-----:|--------:|
## |   0.995|    0.994| 0.995|  0.99| 0.99| 0.962|    0.978|
```

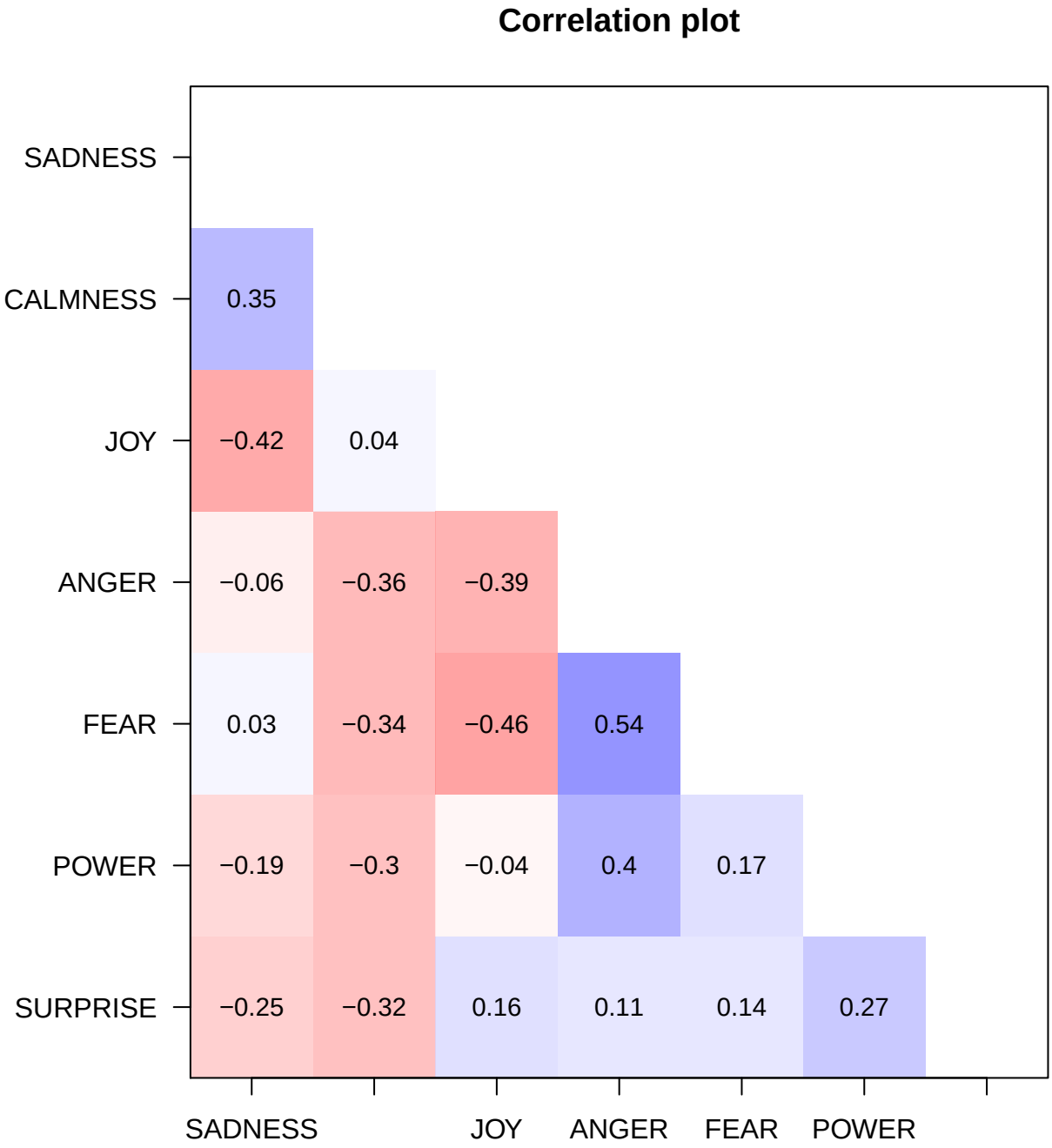
```
source('scr/visualise.R')          # Visualise few aspects of the data
```




ANGER
CALMNESS
FEAR
JOY
POWER
SADNESS
SURPRISE
Rating
1
2
3
4
5
Sadness
Joy
Calmness
Anger
Fear
Power
Surprise
Track
Track
Sadness
Joy
Calmness
Anger
Fear
Power
Surprise


Experiment
Exp1
Exp2
ANGER
CALMNESS
FEAR
JOY
POWER
SADNESS
SURPRISE
Mean Rating ± SE
1
2
3
4
5
Sadness
Joy
Calmness
Anger
Fear
Power
Surprise
Track


ANGER
CALMNESS
FEAR
JOY
POWER
SADNESS
SURPRISE
Rating
5
4
3
2
1
Nonmusician
Musician
Experiment
Nonmusician
Musician



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

3.2.4 Analyse the data

Finally we get to test the planned hypotheses of the experiment. Here we simply test whether the emotion ratings different between the sources and emotions. We do this by applying a Linear Mixed Model, which is a fancy name for a versatile within-subject anova in this case, where we have one random factor (participants) and we test the manipulated factors (Source, Track) and perhaps some non-manipulated group-level descriptors (e.g., Gender and Musical

Expertise) have an effect on ratings of specific emotions expressed by the tracks.

```
source('scr/compare_means.R')    # Compare Sources & Tracks for one emotion
```

Table 1: LMM results for Sadness.

	Estimate	Std. Error	df	t value	Pr(>|t|)
(Intercept)	2.784	0.295	460.816	9.430	0.000
as.numeric(Track)	-0.230	0.051	1180.000	-4.474	0.000
as.numeric(Source)	0.042	0.145	1180.000	0.292	0.771
as.numeric(MusicalExpertiseBinary)	0.096	0.078	88.000	1.232	0.221
as.numeric(Gender)	0.039	0.083	88.000	0.472	0.638
as.numeric(Track):as.numeric(Source)	-0.033	0.033	1180.000	-1.002	0.317

Table 2: Confidence Intervals for the coefficients (Sadness)

	2.5 %	97.5 %
.sig01	0.00	0.27
.sigma	1.11	1.21
(Intercept)	2.21	3.36
as.numeric(Track)	-0.33	-0.13
as.numeric(Source)	-0.24	0.33
as.numeric(MusicalExpertiseBinary)	-0.06	0.25
as.numeric(Gender)	-0.12	0.20
as.numeric(Track):as.numeric(Source)	-0.10	0.03

Table 1 is a raw summary of the LMM analysis, suggesting that there is one main effect (Track) whereas the other factor do not really contribute to the differences. Only one interaction was tested (Track and Source) You would normally report this in text, but that's a different topic (statistics and reporting). Table 2 is related to Table 1 as it shows the confidence intervals of the beta coefficients (model estimates).

One can also produce tables in the same way using a simple script. Here's an example of the sadness ratings across the key variables, showing the N, mean, SD, SE (Standard errors), and lower (LCI) and upper boundaries (UCI) of the 95% confidence intervals.

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

Table 3: Ratings of Sadness across Tracks and Source.

Track	Source	n	m	sd	se	LCI	UCI
Sadness	Exp1	91	4.47	0.79	0.08	4.31	4.64
Sadness	Exp2	91	4.14	1.04	0.11	3.93	4.36
Joy	Exp1	91	1.00	0.00	0.00	1.00	1.00
Joy	Exp2	91	1.01	0.10	0.01	0.99	1.03
Calmness	Exp1	91	2.11	1.00	0.11	1.90	2.32
Calmness	Exp2	91	1.93	1.00	0.10	1.73	2.14
Anger	Exp1	91	1.56	0.76	0.08	1.40	1.72
Anger	Exp2	91	1.53	0.77	0.08	1.37	1.68
Fear	Exp1	91	1.36	0.66	0.07	1.23	1.50
Fear	Exp2	91	2.16	1.08	0.11	1.94	2.39
Power	Exp1	91	1.18	0.41	0.04	1.09	1.26
Power	Exp2	91	1.36	0.68	0.07	1.22	1.50
Surprise	Exp1	91	2.15	1.03	0.11	1.94	2.37
Surprise	Exp2	91	1.08	0.31	0.03	1.01	1.14

```
source('scr/figure1.R')           # create Figure 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, and other readers now that analysis routines can be routinely shared in *Github* and *OSF*, see https://osf.io).

3.2.5 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.

It is actually possible to write the whole manuscript in *RStudio* using *RMarkdown*, which handles citations nicely with a build-in citation manager, and has an excellent APA compatible

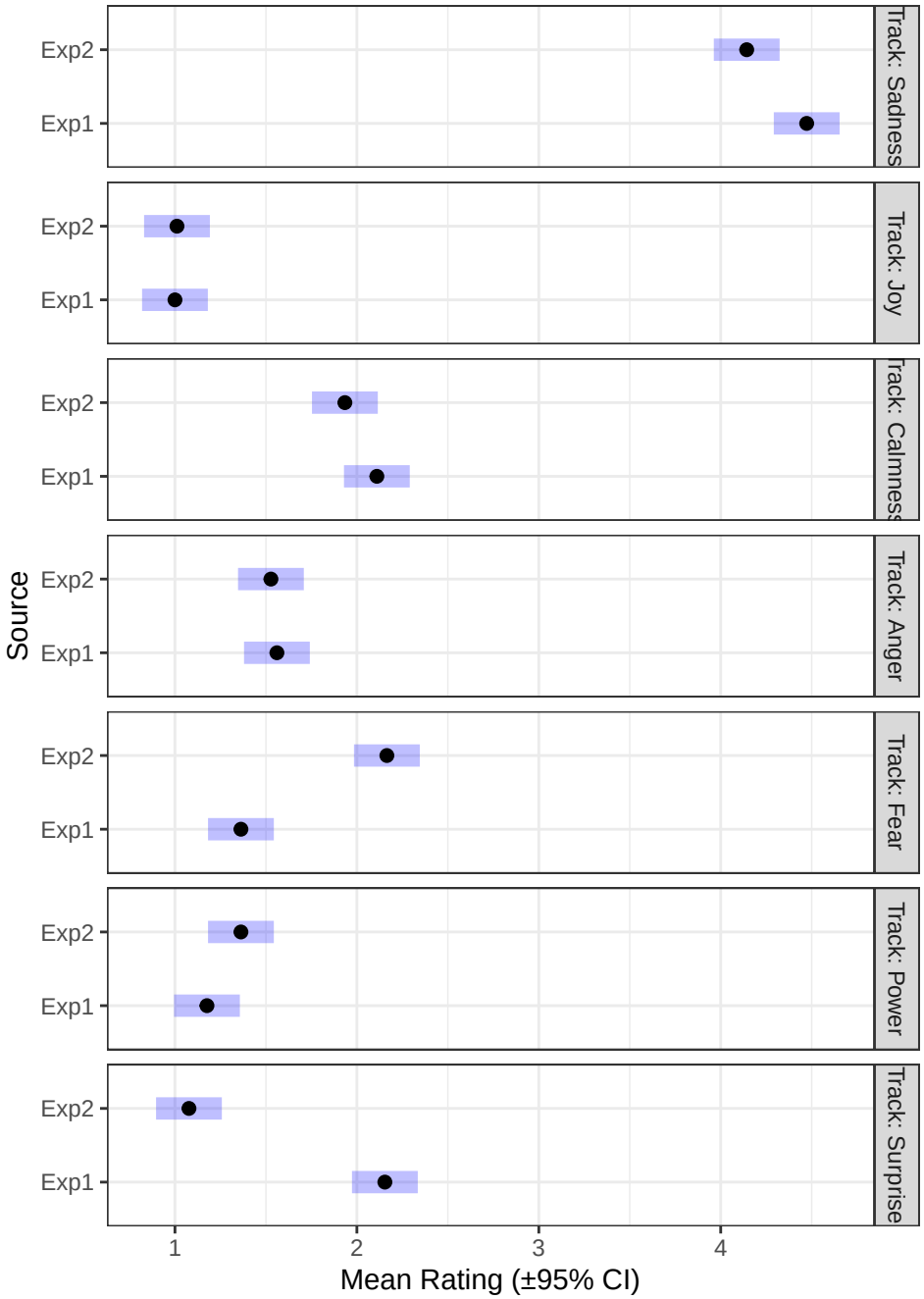


Figure 1: Ratings of Sadness across tracks and sources.

reporting tool (`papaja` library) that allows to weave every detail from the data, analysis, statistics to manuscript. Anyway, that's for an advanced tutorial.

4 Help for statistics with R

4.1 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).

4.1.1 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.

4.2 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.

5 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/template_R