Duke
GLOBAL HEALTH
INSTITUTE

Research Design
& Analysis Core

Core Guide: Reproducible Research Software Tutorial – Part 2: R Markdown

Part of a series addressing common issues in statistical and epidemiological design and analysis

Background and overview

In statistics, reproducibility refers to the ability of an independent researcher to reproduce the same results as the original study if we apply identical methods to the same dataset. Statistical programming tools are available to enhance reproducibility, allowing users to summarize the code and results of an analysis in a single document so that others can easily execute the same analysis and obtain the same results. The scope of Part 2 of this Core Guide is to provide technical and programming support for reproducible research using the software package R Markdown.[1]

R Markdown provides a unified framework for statistical analyses, combining the code, its results, and commentary of the analysts. R Markdown files can be useful for communication with decision makers, whose focus is on the results. It can also be used for collaborating with other analysts, where both the conclusions and the corresponding code are of interest. Using R Markdown in this way facilitates reproducibility and manuscript version control by allowing the analyst to output results in an easily readable format and document the analytical code that produces the results.

In this core guide, we show how to use R Markdown to easily output statistical analysis results. Familiarity with the basics of R will be needed to read this core guide.

Tutorial

Data – the "Informed Health Choices" Podcast Study

The data used for this tutorial come from a randomized controlled trial assessing the effects of the "Informed Health Choices" podcast on the ability of parents of primary school children in Uganda to assess claims about the benefits and harms of treatments (Semakula et al., 2017). The paper was published in volume 390 of *The Lancet* (July 2017). The raw data (in .xlsx format) for the study are available in Supplementary Appendix 2 on the journal website, and are provided with this core guide in Supplemental Material 1. In the tutorial, we demonstrate the use of R Markdown to facilitate reproducing the results in the paper.

The randomized controlled trial was conducted in central Uganda. A total of 675 parents of children in their fifth year of school at 35 primary schools were enrolled and 561 of them were followed up. The parents were randomly allocated to listen to the Informed Health Choices podcast (intervention arm), or typical public service announcements about health issues (control arm). The primary outcome, measured after listening to the entire podcast, was the mean score and the proportion of parents with passing scores (correctly answering at least 11 questions) for an 18-question test measuring competency to assess the

[1] Read more on the context and thought process of reproducible research in another Core Guide – Reproducing the Results of a Published Trial

accuracy of claims about the benefits of health treatments. The secondary outcome was the proportion of parents achieving a mastery score (correctly answering 15 or more questions). The analysis was based on the intention-to-treat principle (Platt and Turner, 2019).

R Markdown

We will walk through the use of R Markdown in the context of analyzing the “Informed Health Choices” Podcast data. To help users go through the details interactively, we provide the following in Supplemental Material 1:

- ***mmc2_data.xlsx*** – “Informed Health Choices” Podcast Study dataset
- ***mmc1.pdf*** – data dictionary for mmc2_data.xlsx
- ***Statistical_Report_R.Rmd*** – R markdown file for this tutorial

Installing R Markdown

R Markdown is an open resource and can be installed by typing `install.packages("rmarkdown").` We recommend using RStudio to work with R Markdown, though it is not required. The use of R Markdown with base R is described in Supplemental Material 2. If you plan to generate PDF output, installation of LaTeX (https://www.latex-project.org/get/) is required.

Basic instructions

We will walk through the use of R Markdown in the context of analyzing the “Informed Health Choices” Podcast data. Key features are described below, with the full analysis code provided in Supplemental Material 1 (***Statistical_Report_R.Rmd***) and the output provided in Supplemental Material 3.

An R Markdown file has the extension .Rmd. When compiled, R Markdown generates a new file that contains the text, code, and results of the code from the .Rmd file. The new file can be a finished web page, PDF, Microsoft Word document, slide show, or other format (Xie et al., 2018).

The .Rmd file contains three basic types of content, as shown in *R code snippets 1-3*:

- An optional YAML header surrounded by `---` as shown in *R code snippet 1*. The title, author, and date included in this section will be displayed at the top of the output document. `output:` determines the default output file type, which can be word_document, html_document, or pdf_document, among others. A template of the YAML header will be automatically provided by RStudio when you start a new .Rmd file.

```
---
title: "Statistical Analysis Report"
subtitle: "Effects of the Informed Health Choices podcast on the ability of parents of
primary school children in Uganda to assess claims about treatment effects: a randomized
controlled trial"
author: "Research Design and Analysis Core"
date: "January 2022"
output: word_document
---
```

R code snippet 1

- R code surrounded by delimiters ```{r} and ```. The methods to customize the R output with the arguments in {r} will be discussed later. *R code snippet 2* shows some example R code embedded in ```, which will load the R packages needed for our analysis and read in the data.

~~~~
```{r, echo=FALSE, warning=FALSE, include=FALSE}
# Packages needed –
# make sure you have already installed all of them using the install.packages() command
library(openxlsx)
library(epitools)
library(tableone)
library(knitr)
library(dplyr)

# read in data - need to change to your own path
test <- read.xlsx("~/mmc2_data.xlsx", sheet=1)
```
~~~~

R code snippet 2

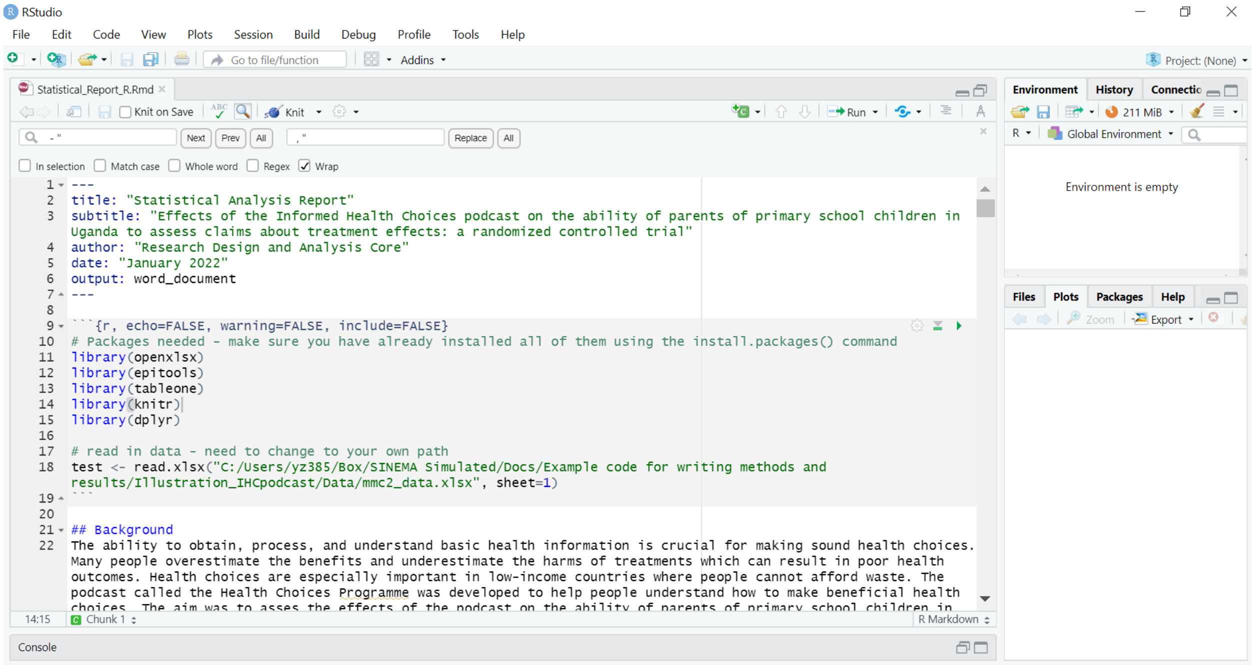
- Text mixed with simple text formatting. More formatting details will be discussed later. In addition, mathematical symbols with LaTeX syntax are supported.

~~~~
## Background

The ability to obtain, process, and understand basic health information is crucial for
making sound health choices. Many people overestimate the benefits and underestimate the
harms of treatments which can result in poor health outcomes. Health choices are especially
important in low-income countries where people cannot afford waste. The podcast called the
Health Choices Programme was developed to help people understand how to make beneficial
health choices. The aim was to assess the effects of the podcast on the ability of parents
of primary school children in Uganda to assess claims about the effects of treatments.
~~~~

R code snippet 3

An R Markdown file has a notebook interface in RStudio (Figure 1). It allows users to run each code chunk individually by clicking the ▶ icon, without compiling the whole .Rmd file and generating the final output document. RStudio will execute the code chunk and display the results inline under the code chunk. This feature allows users to build and debug code interactively, instead of waiting to run the .Rmd file until all analysis code is written. In addition, users can also click the ▼ icon to run all code chunks above the current one.


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Statistical_Report_R.Rmd
Knit on Save
Knit
Run
Next Prev All Replace All
□ In selection □ Match case □ Whole word □ Regex ☑ Wrap
14:15
Chunk 1
R Markdown
Console
Environment History Connectio
211 MiB
Global Environment
Environment is empty
Files Plots Packages Help
Zoom Export

~~~
---
title: "Statistical Analysis Report"
subtitle: "Effects of the Informed Health Choices podcast on the ability of parents of primary school children in Uganda to assess claims about treatment effects: a randomized controlled trial"
author: "Research Design and Analysis Core"
date: "January 2022"
output: word_document
---

```{r, echo=FALSE, warning=FALSE, include=FALSE}
# Packages needed - make sure you have already installed all of them using the install.packages() command
library(openxlsx)
library(epitools)
library(tableone)
library(knitr)
library(dplyr)

# read in data - need to change to your own path
test <- read.xlsx("C:/Users/yz385/Box/SINEMA Simulated/Docs/Example code for writing methods and results/Illustration_IHCpodcast/Data/mmc2_data.xlsx", sheet=1)
```

## Background
The ability to obtain, process, and understand basic health information is crucial for making sound health choices. Many people overestimate the benefits and underestimate the harms of treatments which can result in poor health outcomes. Health choices are especially important in low-income countries where people cannot afford waste. The podcast called the Health Choices Programme was developed to help people understand how to make beneficial health choices. The aim was to asses the effects of the podcast on the ability of parents of primary school children in
~~~

Figure 1. R Markdown interface in RStudio

Empty R chunks (```{r} and ```) can be quickly inserted into the .Rmd file in any of the following ways:

- the keyboard shortcut Ctrl + Alt + I (OS X: Cmd + Option + I)
- the Insert command in the editor toolbar
- or by typing the chunk delimiters ```{r} and ```.

Once an empty R chunk is created, you can embed R code in it. In a .Rmd file, you can create multiple R chunks and compile them together in the final output. Chunk output can be customized with arguments set in the chunk header, for example `{r, include = FALSE}`. Here are five commonly used arguments:

- `include = FALSE` prevents code and results from appearing in the finished file. R Markdown still runs the code in the chunk, and the results can be used by other chunks.
- `echo = FALSE` prevents code from appearing in the finished file, but the results will still appear.
- `message = FALSE` prevents messages that are generated by code from appearing in the finished file.
- `warning = FALSE` prevents warnings that are generated by code from appearing in the finished file.
- `fig.cap = "..."` adds a caption to graphical results.

Code chunks can also be named, for example `{r data_cleaning, include = FALSE}`, which allows easier navigation of the .Rmd file using the drop-down menu in the bottom left corner of the RStudio editor window, as well as other benefits (Wickham and Grolemund, 2016). A specific chunk named "setup" can be included, which will set chunk options that apply to all other chunks. These global chunk options can be overridden by options set within a specific chunk header. For example, including the setup chunk shown in *R code snippet 4* will cause the code in all other chunks to be hidden in the output file, except for any chunks with a header that includes `{r, echo = TRUE}`.

~~~
```{r setup}
knitr::opts_chunk$set(echo = FALSE)
```
~~~

R code snippet 4

Text to be included in the final output should be added outside the R chunks and will always be displayed in the final output. A variety of formatting options are available in R markdown to make the text more informative.

R code snippet 5 shows how to create headers with the pound symbol (#) and lists of items with asterisks (*) and plus signs (+). One # indicates a top-level header. Adding more # indicates lower level headers. A series of * indicates an unordered list of items, while + is for sub-items.

```
## Primary Objectives and Hypotheses

* Objective 1: Determine if mean test score is different between study arms.
  + Hypothesis 1: Mean test score is not different between study arms.
* Objective 2: Determine if the proportion of passing score is different between study arms.
  + Hypothesis 1: The proportion of passing score is not different between study arms.
* Objective 3: Determine if the proportion of mastery score is different between study arms.
  + Hypothesis 1: The proportion of mastery score is not different between study arms.
```

R code snippet 5

R code snippet 6 shows how to create a table in plain text. Columns of the table are separated by |. The relative space between | reflects the column width in the final output. Horizontal lines can be added using the dash (---). Another feature demonstrated in *R code snippet 6* is that the text surrounded by ** will be bolded in the final output. When compiled, *R code snippet 6* will generate the table shown in Table 1 below.

```
### Primary Outcomes

**Outcome**    | **Specifications**                     | **Variables**
------------- | -------------------------------------|----------------
1.Mean score    | Number of correct answers on the test |  score
2.Passing score| Proportion of participants with a passing score (11 or more correct) | pass
```

R code snippet 6

Table 1. Example table of primary outcomes

Outcome	**Specifications**	**Variables**
1.Mean score	Number of correct answers on the test	score
2.Passing score	Proportion of participants with a passing score (11 or more correct)	pass

R code snippet 7 includes a histogram of the first co-primary outcome. We choose to hide the code using echo=FALSE argument and add a caption using the fig.cap argument.

~~~
```{r, echo=FALSE, fig.cap="Distribution of Score"}
hist(test$score, breaks = 13, main="",xlab = "SCORE", xlim = c(0,20))
```
~~~

R code snippet 7

There are multiple ways to create a table in R Markdown. We have shown one of the ways, using plain text in *R code snippet 6*. However, this method does not make full use of the results and objects created by the R program. A commonly used package to conveniently convert R objects into tables is knitr, a free resource which can be installed with the code install.packages("knitr").

R code snippet 8 demonstrates how to use the kable() function from the knitr package to create the baseline characteristics table. We first generate an R object Table1 using the CreateTableOne() function (the R package "tableone" is required to call this function). We then store it as a data frame because the kable() function typically works with a matrix or data frame. Some formatting details are omitted here for simplicity. The final step is to call the kable() function and output the data frame into a nicely formatted table.

~~~
```{r, echo=FALSE}
# Create Table one
Table1 <- CreateTableOne(vars =
c("qn1part3","qn1part4","qn1part5","qn1part6","govt","pnfps","pfps","altmed","frelatives","h
workers","comleaders","radiotv","altmedpract","internet"), strata = "groupF", data = test,
test = F)

# Save table one into a data frame
Table1 <- data.frame(print(Table1, explain = T, varLabels = T, dropEqual = F, nonnormal =
c("qn1part3","qn1part4","qn1part5","qn1part6","govt","pnfps","pfps","altmed","frelatives","h
workers","comleaders","radiotv","altmedpract","internet"),showAllLevels = F))

# Print out table one in R Markdown
kable(Table1, caption = "Baseline and demographic characteristics")
```
~~~

R code snippet 8

To make the final report fully reproducible, numbers/statistics in the text should be inserted with inline R code surrounded by `r and `. Inline R code can be viewed as a simpler version of an R code chunk.

R code snippet 9 includes a paragraph describing the regression results with inline R code, such as `r round(coef(score.model)[2],1)`. We insert the results in the text, displaying the regression coefficients and confidence intervals rounded to the first decimal point.

```
We can see that the adjusted difference in mean scores (podcast group – control group) is `r
round(coef(score.model)[2],1)`%, with a confidence interval (CI) from `r
round(score.model.ci[1],1)`% to `r round(score.model.ci[2],1)`%. This means, conditional on
their level of education, parents who listened to the podcast answered, on average, `r
round(coef(score.model)[2],1)`% more of the questions correctly that parents who only
listened to the public service announcements.
```

R code snippet 9

Finally, additional syntax commonly used in R Markdown can be found in Supplemental Material 4.

Running R Markdown

To generate the final output, click the Knit ▾ button in editor toolbar. You can specify your preferred file format (which can be different from the default format set in the YAML header), including HTML, PDF, Word, etc., from the drop-down list. The final output of the tutorial is shown in Supplemental Material 3.

Conclusion

We have walked through simple reproducible analysis examples using R Markdown. A "cheat sheet" for R Markdown can be found at https://raw.githubusercontent.com/rstudio/cheatsheets/main/rmarkdown.pdf. Additional instructions on how to use R Markdown can also be found at https://rmarkdown.rstudio.com/lesson-1.html.

References

Platt, A.C., and Turner, E.L. (2019). The implications of non-compliance: Randomised controlled trials: the intention-to-treat principle. BJOG *126*, 1337.

Semakula, D., Nsangi, A., Oxman, A.D., Oxman, M., Austvoll-Dahlgren, A., Rosenbaum, S., Morelli, A., Glenton, C., Lewin, S., Kaseje, M., et al. (2017). Effects of the Informed Health Choices podcast on the ability of parents of primary school children in Uganda to assess claims about treatment effects: a randomised controlled trial. Lancet *390*, 389–398.

Wickham, H., and Grolemund, G. (2016). R for Data Science: Import, Tidy, Transform, Visualize, and Model Data (O"Reilly Media).

Xie, Y., Allaire, J.J., and Grolemund, G. (2018). R markdown: the definitive guide (Chapman and Hall/CRC).

Prepared by: John Gallis, MSc and Yunji Zhou, MB
Reviewed by: Duke Global Health Institute | Research Design & Analysis Core
Version: 4.0, last updated 09/01/2022