

Reproducible Documents

Why do we need reproducible documents?

Traditional workflows within science use many different kinds of tools. This requires careful recording of what happens at each step of the scientific process. Good laboratory notebook practices are a feature of many sciences but are not typically taught as part of a psychology curriculum. This combination within psychology of a lack of clear recording of data collection and analysis practices can lead to difficulties in retracing the steps that led to any given outcome in science. A traditional workflow in psychology may involve the design of research and preparation for ethical review using a word processor and perhaps some diagramming tools. This is followed by data collection on a different platform perhaps using pen and paper in a laboratory, but increasingly using automated platforms such as Qualtrics, PsychoPy, or jsPsych. Following this data is manipulated from the shape it comes from the platform and is wrangled into the right shape for analysis sometimes in a statistical program, but often using cut and paste techniques in Microsoft Excel or a similar spreadsheet program. The data is then analysed using some statistical software platform such as SPSS, SAS, JASP, Jamovi or R. When the results are found these are extracted and the whole sequence is written up using some combination of a word processor—typically Microsoft Word—and perhaps some citation software such as Endnote or RefWorks.

The problem with this approach is that it is not very reproducible, the steps are difficult to trace unless the researcher is inordinately meticulous with their recording and it is prone to untraceable errors.

The solution is to create a reproducible document.

What are reproducible documents?

Reproducible documents are a way of eliminating as many of these disparate steps as possible and maximising the recording of the steps that are taken along the research process. The goal is to streamline the research process while making it more transparent and reproducible. Ideally these documents will be made available for the scrutiny of other scientific researchers and the public, but there are many advantages to using reproducible documents as they make the process of conducting science more efficient and they make research much easier to pick up after the inevitable six month to year hiatuses that occur throughout the scientific process as

projects go through their different stages and the myriad competing attention drawing factors of a professional researchers life fragment our time.

We are not yet at a stage where all of the various platforms and processes can be accumulated into one work flow but we can do much better than the traditional approach.

Typically a reproducible document is a combination of some word processing tools to create the document with mechanisms to insert computer code chunks that allow us to load, store, and manipulate data; conduct statistical and mathematical analysis; visualise the analysis producing tables and plots as part of the document output; and integrating with citation software to allow the easy creation of references.

The tools of reproducible documents

Document preparation tools

There are a range of preexisting computational technologies that can be brought together to enable these documents. The TeX and LaTeX systems are well established document preparation and typesetting systems ([latex-project](https://www.latex-project.org/)), these are aided by the use of Markdown ([markdown-guide.org](https://www.markdown-guide.org/)) which provides a very simple document markup language which the researcher uses to create the documents and pandoc (pandoc.org) which allows the same markdown document to be output to many different formats. One markdown document can be used to create websites, PDFs, Word documents, ePub books and many more. These represent the word processing part.

Code chunks

The computer chunks are comprised of various different languages that can be mixed and matched, the most commonly used languages are R and Python, but languages like Julia, Stan, Javascript, SQL, C, and Fortran.

The current document is being written as a reproducible document using the RStudio software and I can include a code chunk in R

```
data <- c(1,2,3,4,5,6,7,8,9,10)
x <- mean(data)
print(x)
```

```
[1] 5.5
```

or in python

```
import statistics
data = [1,2,3,4,5,6,7,8,9,10]
x = statistics.mean(data)
print(x)
```

5.5

Code chunks can be hidden and only the output produced which is useful when including any data wrangling code that you do not want to be seen in a finished output, of for creating tables and plots. The important thing is that the code remains visible in the source form of the document so it can be inspected if necessary and it stays in the one place with the end product document. You can see this process at work in Figure ??, which contains code included in the source document but hidden in the final document.

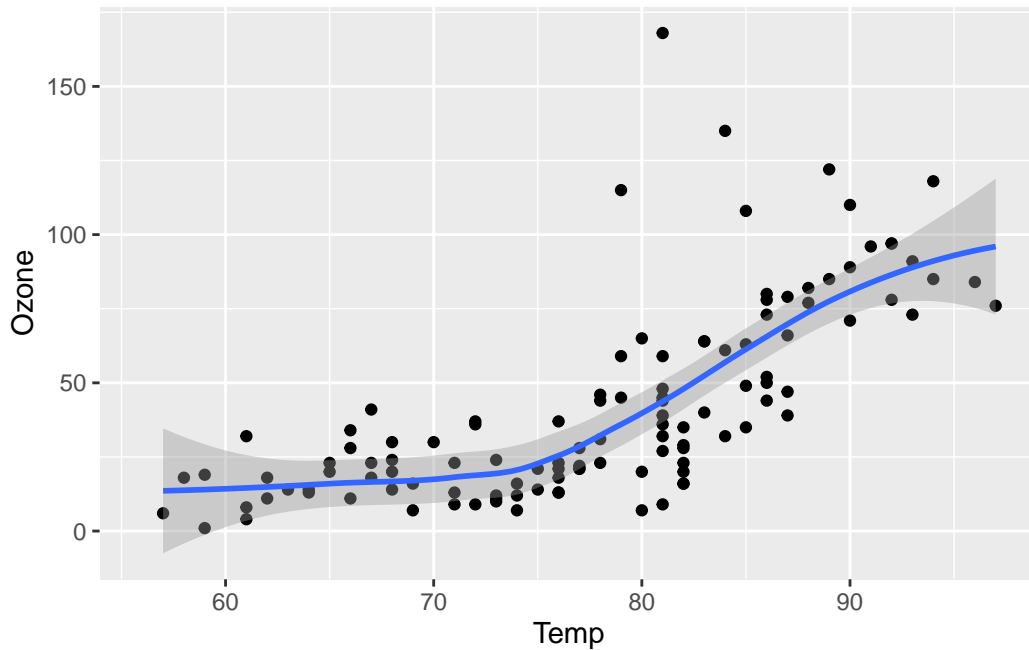


Figure 1: Temperature and ozone level.

Citation software

Citation management and the automatic creation of a references section are important components of reproducible document systems. Most systems interact with pandoc, BibTeX, Zotero,

Crossref, PubMed, and the DOI system to allow in text citations with automated References sections (e.g. McKeown, 2013). This reference will appear in the References section automatically without any more work.

Version control

An important but optional companion technology to the creation of reproducible documents is the variety of version control software such as git or svn. Git is the currently most popular type of version control. Version control systems were developed to keep track of the various changes that were made to computer code and to allow variations of code to be made and merged into a single document and to allow collaboration by multiple coders without the danger of overwriting each other. This turns out to be handy when it comes to writing documents too, especially documents that contain code. Think of version control as a high powered and rigorous combination of “track changes” and backing up files. It is a solution to folders full of final final final versions of the same document. Git is a piece of software that sits on your local computer and controls the versions of your documents, but it has a companion in github that offers remote hosting of documents and enables collaboration between multiple contributors. You can see the code for this document at (github.com/QUBPsy/QUBPsyOpenScience).

Reproducible Document Platforms

There are a variety of ways of creating reproducible documents. Here we will look at some of the options.



RStudio and R Markdown

One of the two major pioneers of mechanisms to produce reproducible documents came from the combination of the R focused Integrated Development Environment (IDE) RStudio. RStudio started in 2011 to create an easy environment in which to write R code and install and manage R packages. In 2012 Yihui Xie introduced the knitr package to enable the creation of documents with code chunks embedded in them. Markdown became the most popular language to develop these documents and the rmarkdown package was introduced in 2014, and it became well embedded within the RStudio IDE where a lot of effort was made to make the production of RMarkdown documents simple and easy. The kinds of documents that could be produced in RMarkdown, webpages and websites, pdf documents, Shiny web apps, journal articles, slides and presentations, and books.