

# How to write reproducible academic papers

## Basic Workflow

This workflow covers many features we want to incorporate to our paper writing:

1. **Write R-markdown file**

A markdown-file (.rmd) consists of **both** R-code and your paper prosa. That means, both analyses and text are kept in the same master file (you can load stuff from external files, too)

2. **Compile Markdown file to Markdown**

Using knitr, you can easily convert your R-ladden Markdown-file (.rmd) into pure markdown text (.md)

3. **Compile Markdown file to Latex file**

Use pandoc to convert your Markdown file to a Latex-file (.tex)

4. **Compile Latex file to pdf**

Using your favorite Tex compiler (eg., pdflatex) you can easily convert your tex file to pdf

Done!

## Step-by-step walkthrough

Ok, let's look at the steps in more detail.

### Step 1: Write R-markdown file

R-Markdown is a mixture of R and markdown (somehow not very surprisingly). Once you learned the syntax (that's easy), you can use each text editor you like for typing your text (it's just a regular text file). Of course, a text editor with syntax highlighting makes sense. I like **RStudio** because it has a number of useful features, and it's free.

Here is a minimal working example for a R-markdown file (.rmd):

Here goes your normal text. Include any Markdown you can think of.

Now assume you want to include some R code:

You can also embed plots, for example:

```
```r
plot(cars)
```
```

Note that the three backticks indicate the beginning (or end) of some code. Here, "{r}" explains to the compiler that it is R code to come.

That's basically it!

BTW: When compiled with Pandoc (see below), inline L<sup>A</sup>T<sub>E</sub>X is supported!

:smile:

## Step 2: Compile Markdown file to Markdown

Ok, now what? We have our easy-to-read Markdown plain text enriched with our R analyses. What to do next? In the next step, we want a “pure” Markdown text file. That means that the R code must be executed and its results (eg., numbers of plots) need be pasted right at their place. There exists an incredible cool tool, called knitr. That's an R package which will execute the R code and will return a Markdown file.

So, how does it work? The easiest way is to execute the following R code:

```
#don't forget to install these packages if not yet done:
library(knitr)
library(markdown)

knit('Untitled.Rmd')
```

So, in essence a one-liner will do the job of translating .rmd to .md. A more advanced way would consist of building a make-file for this job. Then, the make file calls R and hands over the three lines above as a parameter. But we save that for another day.

To recap: You handed in your .rmd file to knitr. knitr said thanks and returned a .md file to you.

### Step 3: Compile Markdown file to Latex file

At this point, you might lean back and just feel happy. The markdown file can easily be to html – a ready-to-post format suitable for your website, your blog, and social media. However, scholarly work is mostly published in form of “papers”. Papers are thought of a collection of limited-size pieces of paper, often stringently formatted (even for submitted papers). Unfortunately, the layout of a html document is quite distinct from the layout of a sheet of paper. That’s why we need an additional tools that formats your text (.md file so far) to fit to paper size. Here’s where **Pandoc** comes into play. We need to convert the .md-file to Latex (.tex-file).

Pandoc comes as a command line tool, and it has quite a number of parameters that you can feed into it. The basic syntax, however, is quite simple:

```
pandoc example.md -o example.tex
```

The syntax basically says: “Hey, I want to convert some document (pandoc.md) to a file (-o for output a file), and the output file should be of pdf format.” Of course, Pandoc needs to installed upfront (same for Latex). If you have RStudio installed, then easy, because Pandoc comes with it.

Now check your folder where you placed the source file. You should find the according Latex file (.tex). Note that Pandoc create a “piece” or a “fragment” of a Latex file. If you want a complete .tex file including the preamble (all the information on how your document should look like), you need to tell Pandoc. You do so using the “-s” parameter; “-s” means “standalone”. So you tell Pandoc that you want a complete, standalone file:

```
pandoc example.md -o example.tex -s
```

Note that it is also possible to convert directly to .pdf from .md. However, in cases where you want more fine-grained control over the file, you will want to have a .tex-file in order to adjust some hidden levers.

### Step 4: Compile Latex file to pdf

Now that you have your Latex file, you can convert it to pdf, the “end product” in many cases for academic writing. Note that you can also use Pandoc to convert your .md file to .doc. That way it will be easier for you to communicate with Word-Lovers (better convince those guy to use text files). Simple enough:

```
pandoc example.md -o example.docx  
pandoc example.md -o example.odt
```

To convert the .tex file to .pdf, open your Latex editor (such as TexShop), open the tex file and click on the appropriate conversion button. From the command line you can type:

```
pdflatex example.tex
```

That's it – you have your pdf file!

## Citations and References

### Step A: Create your library

For academics, it is essential to include references in a paper. In Markdown, this is accomplished by citing a paper using a unique ID, so that it which paper out of your library you want to cite. This ID is the bibtex ID. So your first step is to get list of bibtex entries in a text file. This file is referred to as a .bib file. For example, a bibtex entry might look like this:

```
@article{fyfe_digital_2011,
  title = {Digital Pedagogy Unplugged},
  volume = {5},
  url = {http://digitalhumanities.org/dhq/vol/5/3/000106/000106.html},
  number = {3},
  urldate = {2013-09-28},
  author = {Fyfe, Paul},
  year = {2011},
  file = {fyfe_digital_pedagogy_unplugged_2011.pdf}
}
```

Taken from here.

If this looks confusing to you, no worries. Your reference manager will likely be able to spit out your references in this format. For example, I use Mendeley, and it comes with the option to export references in bib-format. You can even convert your whole library to a bib-file. Don't forget to place your bib file in the working folder, so that Pandoc will find it.

### Step B: Put citations in your R-Markdown file

Now cite some paper in your text, using the ID, in this way:

- A reference formatted like this will render properly as inline- or footnote- style citations
- "For citations within quotes, put the comma outside the quotation mark" [@fyfe\_digital\_2011]

Taken from here.

One more thing: You need to tell the name of the bibliography file, somewhere in your main text file. As you may have expected, there is a certain place where to do so, namely in the header at the very beginning of the .rmd-file:

```
---
title: "Example with References"
author: "Sebastian Sauer"
date: "23 Dezember 2015"
output: html_document
bibliography: mybib.bib
---
```

Next, you will “knitr” your .rmd file to .md, as explained above as first step. Note that Knitr will “see” your references as simple, plain, boring plain text, and will do nothing with it. So, at least, it won’t do any harm or will not interfere.

### Step C: Compile text with citations

Pandoc has the feature to read bibtex-style citations and convert them to “real” citations including a bibliography. Pandoc understands this parameter in order to digest your citations: `--filter pandoc-citeproc`. Let’s see:

```
pandoc example.md -o example.pdf -s --filter pandoc-citeproc -S
```

The “-S” parameter stands for “smart” output, rendering the output a bit more beautiful (see Pandoc documentation for more details). Let’s see... Worked fine for me.

### Step D: Choose your citation style

For some insane reasons, there exists literally thousands of citation styles. Choose the one your journal loves. For psychologists, for example, this will often be APA6. How to get the styles?

Details on your citation style (e.g, whether numbers only in Text or last name plus year, etc.) are quite efficiently and quite commonly stored using the CSL format (Chicago style language). This is also some plain text, markup style “language”. Luckily, you can download thousands of styles from here. For example, I searched on that site for the APA citation style, and downloaded it to my project directory.

Now inform your header in the .rmd-file that you want to use (here `apa.csl`):

```
---
title: "Example with References"
author: "Sebastian Sauer"
date: "23 Dezember 2015"
output: html_document
bibliography: mybib.bib
csl: apa.csl
---
```

Then, again, do your compilations: `.rmd -> .md -> .pdf`.

## Latex lovers long for Latex, even in Markdown

Ok, Latex lovers, you can get what you want: seamless Latex integration to Markdown, see [here](#) for a nice intro.

Good news first: Pandoc has a built-in Latex-Markdown mixture feature, so everything runs quite automatically. See for example this md-file. Is it really *that* easy? Yes, it is!

## APA manuscript layout

Now we are having pretty much of the functionality we would want to. However, human desire is boundless, and we certainly want more. Pressingly, it would be nice if the document would automatically be formatted to suit the requirements of the APA, from a psychologist's point of view. Similar things may be said for other fields. Here, we explore next how to automatically format your document to APA (6ed.) style. See [here](#) for more details.