Writing an IEEE CS paper with R Markdown

Marco Torchiano[*]
[*]Politecnico di Torino
Torino, Italy
Email: marco.torchiano@polito.it

***Abstract*—*Context*: paper based on empirical studies are becoming more and more common in Software Engineering (SE) research. In such studies a relevant part of the work consists in data analysis. Sharing the data and the analysis method (or even code) is considered a good practice, the goal is to make research as reproducible as possible.**
A widely used tool for performing statistical analysis is R, often associated with the RStudio development environment. Such environment provide provides R Markdown as a blend of scripting and documentation with powerful mechanisms.

***Goal*: the objective of this work analyze the main issues in writing a paper in markdown and generating a document compliant with the guidelines for IEEE CS conferences.**

***Method*: a case study was conducted by writing a paper for an ICSE workshop, during the task several issues were encountered and possible solutions are put forward and validated.**

***Results*: a kit comprising a template and guidelines has been put together which allows producing a paper formatted properly.**

***Conclusions*: starting from an R Markdown document we can generate a compliant paper that allows reproducible data analys one cornerstone of an open scientific method. Actually this very paper has been written using the presented approach.**

I. Introduction

We are (gladly) observing and increasing number of paper based on empirical studies in Software Engineering (SE) research venues.

In such studies a relevant part of the work consists in data analysis. This fact sparks off two distinct needs:

- from the public perspective the demand to have access to both the data and the analysis procedures.
  In the empirical software engineering community (as well as in many other scientific communities) it is a recommended good practice to make the data from experiments available to the readers. Typically a replication package is provided containing all the materials required to replicate the whole study. What is typically missing from the replication package are the scripts and methods used to anlyze the data.
- from an author perspective the requirement to produce the paper with a seamless and streamlined process.
  A widely used tool for performing statistical analysis is **R** [1]. In addition to providing a powerful processing environment, R enables – through the use of different additional packages – a literate programming [2] approach.

II. Background

I will assume hera a basic knowledge of what the R statistical package [1] is and how it works.

A. R markdown

R Markdown[3] is an authoring format that enables easy creation of dynamic documents, presentations, and reports from R. It combines the core syntax of markdown[4] (an easy-to-write plain text format) with embedded R code chunks that are run so their output can be included in the final document. R Markdown documents are fully reproducible (they can be automatically regenerated whenever underlying R code or data changes).

R Markdown is integrated in the R Studio [5] and is based on three main components:

- the *R* statistical environment: the engine used to perform statistical computations;
- the *knitr* package [6]: a templating tool based on R, used to process the R Markdown files by executing the R code chunks an generating pure markdown;
- the *pandoc* tool [7]: that is used to process the markdown generated by knitr and produce several kind of documents such as LaTeX, PDF (through LaTeX), HTML, and Word.

A typical R Markdown document consists of two main parts:

- a metadata section that specify general properties of the document, e.g. the title and the output format;
- a markdown document, that contains the actual document contents.

The advantage of R Markdown is that its syntax is extremely more simple and easier to read than HTML or LaTeX. In addition, through the flexibility of pandoc, it can be converted in virtually any kind of output format; starting from the same document you can generate a PDF document or an HTML page.

B. IEEE CS guidelines

IEEE Computer Society conference – e.g. ICSE [8] – mandate a specific format for the contributions that will be published in the proceedings of the event.

The strict adherence to the formatting guidlines is compulsory to have the paper accepted and published.

III. Guidelines

Here are the basic guidelines to write an R Markdown document that can be converted into a PDF conforming to the IEEE CS publication guidelines.

The metadata section must specify that the output will be a PDF file and the generation will use a specific template:

```
output:
  pdf_document:
    template: IEEEtran_template.tex
```

The template file contains most of the tricks I developed to obtain a IEEE CS guidelines compliant file.

There are several options to be added for the `pdf_document`, they should be added, properly aligned, after the `template` option.

In addition, since a few details cannot be implemented using only markdown features, we need to use directly within the markdown some LaTeX commands; for this purpose we need to enable the preservation of the raw latex commands using the following directive in the metadata section:

```
raw_tex: true
```

A. Title

The title is defined in the metadata section, e.g. as:

```
title: "Papers with R MarkDown"
```

B. Authors

The authors need to be specified in two parts:

- the names of the authors with an affiliation code, e.g.

  ```
  author:
    - name: Marco Torchiano
      affiliation: 1
  ```

  Please not that the affiliation codes *must* be integer numbers that are then used in the affiliation description part.
- the affiliation description, e.g.

  ```
  affiliation:
    - code: 1
      organization:"Politecnico Torino"
      address: |
        Torino, Italy
      email: marco.torchiano@polito.it
  ```

C. Abstract

The abstract is written in the metadata section. e.g. as

```
abstract: |
  the abstract can span
  several lines.

  Include paragraphs and
  _emphasized_ words.
```

D. Bibiographic citations

Citations can be written using the standard markdown syntax, e.g.

```
[@Rstat]
```

The citation codes and the relative information are taken from a BibTeX database file that must be specified in the metadata section, using the following directive:

```
biblio-files: biblio.bib
```

The BibTeX database contains all the information about the cited refrences; for instance information about the previous citation is:

```
@manual{Rstat,
Address = {Vienna, Austria},
Author = {{R Core Team}},
Organization =
{R Foundation for Statistical Computing},
Title =
{R: A Language and Environment for
    Statistical Computing},
Url = {http://www.R-project.org/},
Year = {2013},
```

For more information on the BiTeX syntax and tools please refer to any of the numerous online resources.

By default – at least with version 0.98 of RStudio – the LaTeX file is not automatically processes with the Bibtex tool, therefore the bibliography is missing from the file generated with the "*Knit PDF*" command.

Fortunately there is a simple workaround:

- instruct the converter to keep the `.tex` file (by default is removed after compilation to PDF), by inserting the following line in the metadata block:

  ```
  output:
    pdf_document:
      template: IEEEtran_template.tex
      keep_tex: true
  ```
- after you generate the PDF just open the `.tex` file that has been generated and use the "*Compile PDF*" command. This steps just processes the references and adds the *References* section at the end of the paper.

E. Figures

It is possible to include figures in the paper using the normal markdown syntax or generate diagrams in an R code chunk.

By default the conversion from markdown does not include figure captions that are typically mandatory in a paper. Therefore we need to enable them using the following directive in the metadata section:

```
output:
  pdf_document:
    fig_caption: true
```

The typical way of referring to a figure in a paper is by number, which is assigned automatically and progressively by the LaTeX compiler. Therfore to refer to a figure we need to use the basic LaTeX mechanism:

- within the figure caption add a `\label` macro, e.g.

  ```
  Sample figure caption.\label{fig:sample}
  ```

  – remember that you need to leave an empty line before and after the image in order to have it embedded in the `\figure` environment. Otherwise you'll end up with an inline figure mixed with the text and without either a caption or a reference label;

  - beware, if you put the `\label` inside a string – e.g. in the `fig.cap` parameter of a code chunk –, you need to quote the backslash as in `fig.cap="Caption \\label{fig:sample}"`.
- anytime you need to refer to the figure use a `\ref` macro, e.g.

  ```
  As shown in figure \ref{fig:sample}
  ```
- if you have a wide figure and want to fit its size to the whole page and not limit it to one column width, you can use the macros `\widefig` to enable page-wide figures (i.e. using the `figure*` LaTeX environment) and `\normfig` to switch back to column-wide figures.

For instance we can show in figure 1 the qqplot a random normal sample. Please note that the figure is generated by a few lines of R when the source R Markdown of this article is processed.

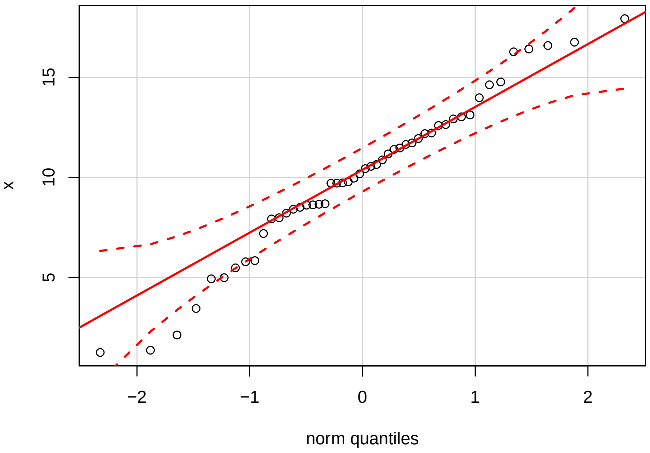

Fig. 1. Sample figure caption

F. Tables

For tables we need to use an approach similar to figures as far as references are concerned.

To refer a table therefore we need to use the LaTeX mechanism:

- within the table caption add a `\label` macro, e.g.

  ```
  Sample table caption.\label{tab:sample}
  ```
- anytime you need to refer to the table use a `\ref` macro, e.g.

  ```
  As shown in table \ref{tab:sample}
  ```

IV. Issues and solutions

A. Author block

B. Abstract

C. Bibliography and references

Problems

- the predefined *pandoc* reference formatting is not cusomizable enough, in particular it generates (at the end of the document) the reference list which is impossible to format according to the guidelines.
- the option `--biblatex` generates latex-friendly references that can be formatted properly. Though most people use *BibTeX* which is not fully compatible with *BibLaTeX*.

The solution:

- a normal '.bib' file can be used by specifying it in the meta data block:

  ```
  biblio-files: biblio.bib
  ```
- we specify the `--biblatex` option to generate LaTeX-friendly citations, in the meta-data block:

  ```
  output:
    pdf_document:
      keep_tex: true
      pandoc_args: ["--biblatex"]
  ```

  The argument forces the generation of the BibLaTeX `\autocite` directives in correspondence to the bibliography citations in the document.
- since we want to use the usual *BibTeX* we need to convert the biblatex citations into BibTeX citations by means of a redefinition in the LaTeX template:

  ```
  \newcommand{\autocite}[1]{\cite{#1}}
  ```

  This LaTeX directive implicitly replaces the BibLaTeX `\autocite` macros – generated in the conversion to LaTeX – with the BibTeX `\cite` directive.

D. Internal references

E. Tables

F. Figures

References

[1] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013. [Online]. Available: http://www.R-project.org/

[2] D. E. Knuth, "Literate programming," *The Computer Journal*, vol. 27, no. 2, pp. 97–111, 1984.

[3] R Markdown - dynamic documents for R. [Online]. Available: http://rmarkdown.rstudio.com

[4] J. Gruber. Markdown. [Online]. Available: http://daringfireball.net/projects/markdown/basics

[5] (2015) R. [Online]. Available: http://www.rstudio.com

[6] Y. Xie, *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, 2013.

[7] J. MacFarlane. Pandoc - a universal document converter. [Online]. Available: http://johnmacfarlane.net/pandoc/index.html

[8] Formatting and submission guidelines. [Online]. Available: http://2015.icse-conferences.org/submission-guidelines