SIGCHI Conference Proceedings Format

Ben Trovato
Institute for Clarity in Documentation
Oxford, UK
trovato@corporation.com

Lars Thørväld
The Thørväld Group
Hekla, Iceland
larst@affiliation.org

Valerie Palmer
Palmer Research Laboratories
San Antonio, US
cpalmer@prl.com

ABSTRACT

This sample paper describes the formatting requirements for SIGCHI conference proceedings, and offers recommendations on writing for the worldwide SIGCHI readership. Please review this document even if you have submitted to SIGCHI conferences before, as some format details have changed relative to previous years. Abstracts should be about 150 words and are required.

Author Keywords

Authors' choice; of terms; separated; by semicolons; include commas, within terms only; this section is required.

CCS Concepts

•**Human-centered computing** → **Human computer interaction (HCI);** *Haptic devices;* User studies; Use the 2012 Classifiers from here: https://dl.acm.org/ccs/ccs_flat.cfm. They should be put in the file **ccsxml.tex**

Introduction

Using a tool like R Markdown to write scientific papers makes your work more transparent and reproducible. It also reduces the risk of errors, because you can dynamically insert tables, figures, and summary statistics directly from the data they are generated from insted of transferring results manually from statistical software to manuscript.

This example illustrates how to use the Paper template for CHI'20 submissions, a variation on the CHI'18 template. The content in this example is adapted and adjusted from content in the **sample-sigchi.tex** template included with the ACM template, to illustrate how to create the same content through the R Markdown workflow as well as to showcase additional features enabled by R Markdown.

PAPER META DATA

Set meta data (copyright, authors, keywords, title, keywords, etc.) in the YAML header of the .Rmd file in which you write the manuscript. This is done in the form of `key: value` pairs, e.g. `title: Writing CHI Proceedings Papers With R Markdown`. When compiling to a PDF (in RStudio, just click the 'Knit' button), the information in the YAML header is plugged into the CHI Extended Abstracts LaTeX template. (If you were to take a look at this template file inside of the rticles package, you would see e.g. `\def\plaintitle{$title$}` where stuff between dollar signs is interpreted as a variable to be searched for in the YAML header and plugged into the template when generating a PDF).

Note the sole exception for adding paper meta data: The CCS Concepts are messy to insert from the YAML header, so you should manually insert this into the **ccsxml.tex** file from which it will be included in your manuscript.

CHI'20, April 25–30, 2020, Honolulu, HI, USA

ISBN 978-1-4503-6708-0/20/04…$15.00

DOI: https://doi.org/10.1145/3313831.XXXXXXX

THE BODY OF THE PAPER

Typically, the body of a paper has a hierarchical structure, with numbered or unnumbered headings for sections, subsections, sub-subsections, and paragraphs. Whereas in LaTeX you use the command `\section` for main sections, in R Markdown you simply use `#`, as in `# The Body of The Paper`. For subsections, or sub-subsections, use additional hashes, as in `## This Become a Subsection`, and `#### This Becomes a Paragraph Heading`.[1]

If you want some section to be unnumbered in the output, add `{-}` after the section name, as in `# Unnumbered Section{-}`.

Indicate the start of a new paragraph with a blank line in your input file; that is why this sentence forms a separate paragraph. This line, however, does not form a separate paragraph.

Type Changes and *Special* Characters

Make words or phrases *italicized* by surrounding them with a single *; **embolden** them by surrounding them with **two**. `Typewriter-style` (for instance, for computer code) you create by surrounding text with `` `backticks` ``.[2]

Citations

Citations to articles [@bowman:reasoning; @braams:babel; @Cohen07], conference proceedings [@clark:pct] or maybe books [@lamport:latex; @salas:calculus] listed in the Bibliography section of your article will occur throughout the text of your article. To insert a reference in the R Markdown syntax, type @ followed by the citation key. The key is a short

[1] By the way, this is how to insert footnotes.

[2] Another footnote here. Let's make this a rather long one to see how it looks.

reference uniquely identifying each entry in in the `.bib` file for your article, in which your references are listed in BibTex format.

For example, to cite the article "Deciding equivalances among conjunctive aggregate queries" from our `.bib` file, write `[@Cohen07]`. If you drop the `[]`'s, you get author names, as well as the citation: @Cohen07. See this short guide for more.

DYNAMIC REPORTING

One of the most important benefits of writing in R Markdown (aside from being able to compile to other formats than PDF, such as HTML or even Microsoft Word), is the ability to insert results dynamically into your manuscript using code chunks or inline code. This means that you can do analyses **directly** in your manuscript or, probably better, read file(s) with data, summaries, or results directly into your manuscript and refer to them dynamically.

This is important for two (related) reasons: 1. You avoid initial manual transfer of results from statistical software to manuscript, which reduces the risk of error. 2. If at a later stage you update the analysis files, the results reported in your manuscript are automatically also updated - this again reduces the risk of mistakes, because you don't need to manually update figures and tables.

In R Markdown syntax, **code chunks** have the following form (cf. *R Markdown: The Definitive Guide*):

````
```{coding_language chunk-label, chunk_options}
# your code goes here
```
````

Inline code has the form `` `coding_language #code here` ``.

Setup chunk

The first chunk in an R Markdown document is usually used to load packages and set default chunk options, for example like so (we normally add the chunk option `include=FALSE` to not include output from this chunk in the manuscript; here we just add `message=FALSE` to suppress the message that the tidyverse package has been loaded):

```
library(tidyverse)
knitr::opts_chunk$set(echo = FALSE,
  message = FALSE, warning = FALSE)
# these options will exclude code output,
# messages, or warnings in knitted manuscript
```

Inline results

You might read in a made-up data set of goals scored by basketball players like so:

```
data <- read_csv("data/fakeBasketData.csv")
```

We can use inline code to dynamically report properties of this data set. For example, "there are a total of 270 observations of goals scored. The mean number of goals made by any player in a given game is: 17.2555556".

Table 1. Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
π	1 in 5	Common in math
$	4 in 5	Used in business
Ψ_1^2	1 in 40,000	Unexplained usage

Table 2. The first 5 rows of some made-up basket data.

Player	goals
Carmelo Anthony	4
Carmelo Anthony	2
Carmelo Anthony	10
Carmelo Anthony	3
Carmelo Anthony	1

Tables

For tables, you could use LaTeX syntax directly. This might be useful if your table itself contains LaTeX syntax, as in Table 1.

However, the power of writing in R Markdown is that you can read in data and automatically create corresponding LaTeX tables. The easiest way is probably to use the `kable` function. For example, Table 2 shows the first 5 rows in our basket data set.

You can reference Table 2 with `\@ref(tab:basket-data)`.

You can also do arbitrary transformations and analyses of the data before creating a table, as in Table 3.

To set a wider table, which takes up the whole width of the page's live area, put it in a `\table*` environment by adding the parameter `table.env = 'table*'` to the `kable` function, like in Table 4.

Figures

Static figures

Figures are similarly included via code chunks. You can include arbitrary image files, as in Figure 1.

If you don't give it a caption in the chunk options (with something like `fig.cap="My caption"`), the figure does not float:

Figure 1. Here's a little pretty fly.

Table 3. Summary statistics of goals scored by top players in made-up basketball season.

Player	Total goals scored
Blake Griffin	406
Brook Lopez	776
Carmelo Anthony	166
Damian Lillard	808
David Lee	362
David West	492
Demar Derozan	972
Deron Williams	365
Dwyane Wade	312

Figure 2. A sample black and white graphic that has been resized with the `out.height` and `out.width` chunk options.

You can resize the figures with the chunk options `out.height` and `out.width`, as in Figure 2. If you only care about LaTeX output, you can resize e.g. in inches or relative to the column width (`out.height = '1in'` or `out.height = '0.50\\columnwidth'`), but if you want to get maximum value out of R Markdown and be able to output also to html formats, set it with a percentage (`out.height = '50%'` - when outputting to PDF via LaTeX, this will be translated into `out.height = '.5\linewidth'`, see the bookdown reference).

If you need to style text in a caption, or include references in the caption, you have two options (see bookdown on 'text references'):

1. set the caption with the chunk option `fig.cap` and use LaTeX rather than markdown syntax. As the figure caption is a string, you must escape the LaTeX syntax's \ with another \. The caption for Figure 2 would then have been written like this: `fig.cap="A sample black and white graphic that has been resized with the \\texttt{out.height} and \\texttt{out.width} chunk options."`.

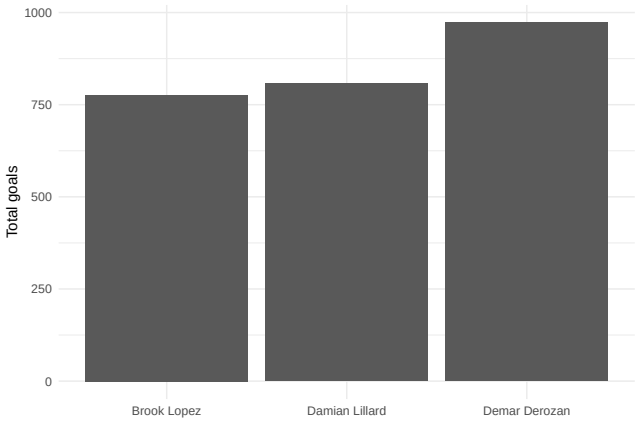

Figure 3. Total number of goals by the top 3 players in made-up basketball season

2. write the caption in the body text with the syntax `(ref:chunk_label) My caption here.` and then refer to it in the chunk options with `fig.cap='(ref:chunk_label)` as we did for the resized fly caption.

Dynamic figures

Again, the power of R Markdown is that you can include e.g. plots that are dynamically generated from the underlying data. For example, Figure 3 is a simple visualisation of the basket data.

As with tables, you may want a figure to span two columns. To do this, set the environment to `figure*` with the chunk option `fig.env = 'figure*'`. You can fiddle around with the size and aspect ratio of the generated plot with the chunk options `fig.height` and `fig.width`. If your image is very large, you may want to restrict its width with `out.width`.

Math Equations

You may want to display math equations in three distinct styles: inline, numbered or non-numbered display. Each of the three are discussed in the next sections. You can use usual LaTeX syntax directly, or R Markdown.

Inline (In-text) Equations

A formula that appears in the running text is called an inline or in-text formula. In LaTeX it is produced by the **math** environment, which can be invoked by surrounding text with dollar signs: $. You can use any of the symbols and structures, from α to ω, available in LaTeX. For example, here's a nice equation inline: $\lim_{n\to\infty} x = 0$. If you're writing in RStudio, you can even hover over it to see the rendered output displayed!

Display Equations

A numbered display equation—one set off by vertical space from the text and centered horizontally—is produced by using LaTeX syntax directly to put the content in an `equation`

Table 4. Bigger display of more summary statistics of goals scored by top players in made-up basketball season.

Player	Total goals scored	Goals per game
Blake Griffin	406	13.533333
Brook Lopez	776	25.866667
Carmelo Anthony	166	5.533333
Damian Lillard	808	26.933333
David Lee	362	12.066667
David West	492	16.400000
Demar Derozan	972	32.400000
Deron Williams	365	12.166667
Dwyane Wade	312	10.400000

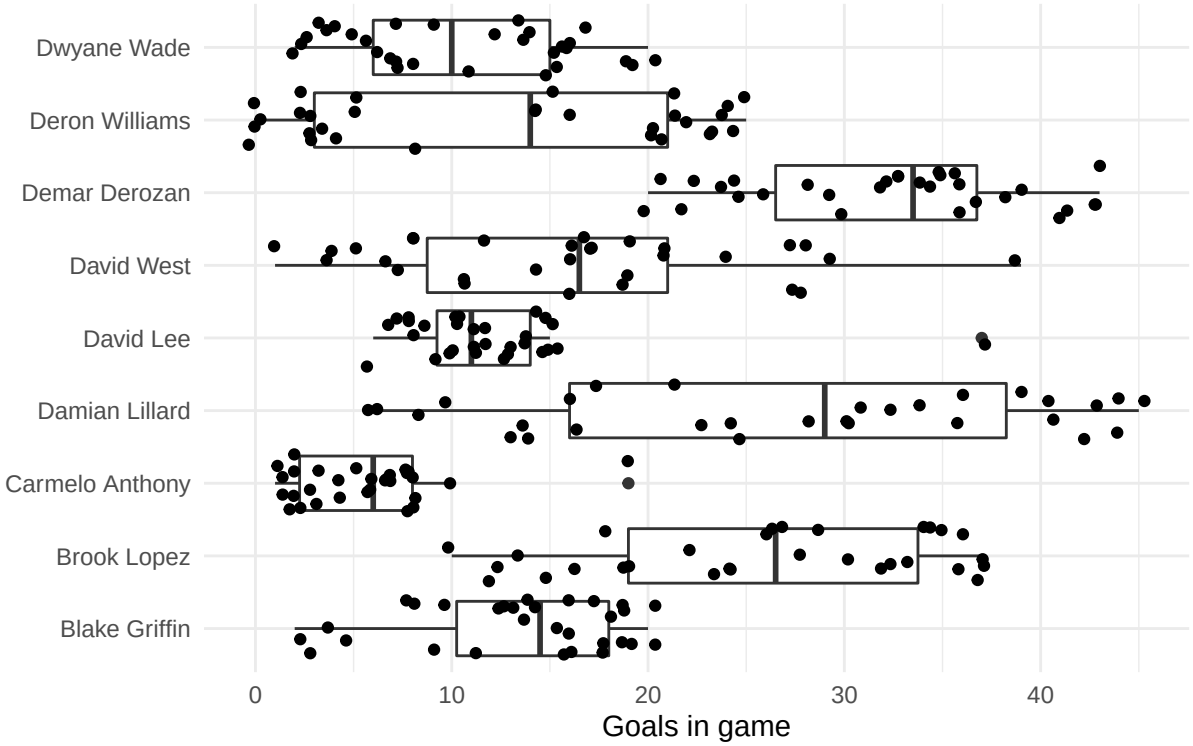

Figure 4. Distribution of goals scored by game for players in made-up basketball season

environment[3]. So here's that nice equation from above:

$$\lim_{n \to \infty} x = 0 \tag{1}$$

They can be assigned labels with the syntax `(\#eq:label)`. Refer to the equation with `\@ref(eq:display-equation)`, e.g. see Equation (1).

To make an unnumbered display equation, surround the expression with two dollar signs:

$$\lim_{n \to \infty} x = 0$$

[3]In fact, you can use any arbitrary LaTeX syntax directly in your .Rmd document.

CONCLUSIONS

This paragraph ends the body of this sample document. Remember that you might still have Acknowledgments or Appendices; brief samples of these follow. There is still the Bibliography to deal with; and we will make a disclaimer about that here: with the exception of the reference to the LaTeX book, the citations in this paper are to articles which have nothing to do with the present subject and are used as examples only.

APPENDIX

HEADINGS IN APPENDICES

The rules about hierarchical headings discussed above for the body of the article are different in the appendices. You being the **appendix** section with the special header `# (APPENDIX) Appendix {-}`. Then, any subsequent top level headers (`#`) indicates the start of each Appendix, with alphabetic order

designation (i.e., the first is A, the second B, etc.). So, if you need hierarchical structure *within* an Appendix, start with **subsection** (`##`) as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

Introduction

Paper meta data

The Body of the Paper

Type Changes and Special Characters

Citations

Dynamic reporting

Inline results

Tables

Figures

Math Equations

Inline (In-text) Equations

Display Equations

Conclusions

References

MORE HELP FOR THE HARDY

For acknowledgements, you may want to use the LaTeX syntax for this from the ACM template example, in which case you'll put acknowledgement text in between `\begin{acks}` and `\end{acks}`. Alternatively, just start an unnumbered heading `# Acknowledgements{-}` and write your text, like this:

Acknowledgements

The authors would like to thank Dr. Yuhua Li for providing the MATLAB code of the *BEPS* method.

The authors would also like to thank the anonymous referees for their valuable comments and helpful suggestions. The work is supported by the National Natural Science Foundation of China under Grant No. 61273304 and Young Scientists' Support Program (http://www.nnsf.cn/youngscientists).