

GRAPH NEURAL NETWORKS - DRAFT

LEARNING GRAPH EMBEDDINGS FOR LINK PREDICTION AND NODE CLASSIFICATION

THROUGH HYPERGRAPH CONVOLUTIONS, MARKOVIAN NETWORKS, AND VARIATIONAL INFERENCE

By

MV

A thesis submitted to
Harrisburg University of Science and Technology
for the degree of
DOCTOR OF PHILOSOPHY



Department of Analytics
Harrisburg University of Science and Technology
April 2020

ABSTRACT

DEDICATION

To my family

Ph.D. COMMITTEE APPROVAL

To the Faculty of Harrisburg University of Science and Technology:

The members of the Committee appointed to examine the dissertation of MV find it satisfactory and recommend that it be accepted.

K P, Ph.D., Chair

RF, Ph.D.

RS, Ph.D.

KS, Ph.D.

Glenn Mitchell, MD

ACCEPTANCE PAGE

Harrisburg University of Science and Technology

As chair of the candidate's graduate committee, I have read the thesis of MV in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

K P, Ph.D.
Chair, Graduate Committee
Harrisburg University of Science and Technology

Accepted for the Department of Analytics

RF, Ph.D.
Chair, Department of Analytics
Harrisburg University of Science and Technology

Accepted for the University

Bilita Mattes, D.Ed,
Provost
Harrisburg University of Science and Technology

ACKNOWLEDGMENTS

My deepest gratitude to Dr. KP for his mentoring, and to my committee members: Dr. RS, Dr. RF, and Dr. KH.

Table of Contents

	Page
Introduction	1
Why use it?	1
Who should use it?	2
1 R Markdown Basics: The Markdown syntax	3
1.1 Markdown basic syntax	3
1.1.1 Italics and bold	3
1.1.2 Inline code	4
1.1.3 Sub and superscript	4
1.1.4 Strikethrough	4
1.1.5 ‘Escaping’ (aka “What if I need an actual asterisk?”)	4
1.1.6 Endash (—), emdash (—)	4
1.1.7 Blockquotes	4
1.1.8 Headings	4
1.1.9 Lists	5
1.1.10 Line breaks	6
1.1.11 Hyperlinks	6
1.1.12 Footnotes	6
1.1.13 Comments	7
1.1.14 Math	7
1.2 Additional resources	7
2 Adding code	9
2.1 Code chunks	9
2.1.1 Setup chunks	10
2.1.2 Including images	10
2.1.3 Including plots	12

2.1.4	Including tables	13
2.1.5	A note on content positioning	14
2.2	Inline code	15
2.2.1	Referring to results computed in other languages than R	15
3	Citations and cross-references	17
3.1	Citations	17
3.1.1	PDF output	18
3.1.2	Gitbook output	19
3.1.3	Insert references easily with the <code>citr</code> add-in	20
3.2	Cross-referencing	20
3.2.1	Section references	21
3.2.2	Figure (image and plot) references	21
3.2.3	Table references	23
3.2.4	Including page numbers	23
3.3	Customising your thesis' front matter 'n stuff	24
3.3.1	Shorten captions shown in the list of figures (PDF)	24
3.3.2	Shorten captions shown in the list of tables (PDF)	24
3.3.3	Shorting the running header (PDF)	25
4	Final Notes on The OxThesis template and on collaboration	26
4.1	Beginning chapters with quotes	26
4.2	Highlighting corrections	27
4.2.1	Short, inline corrections	27
4.2.2	Blocks of added or changed material	27
4.2.3	Stopping corrections from being highlighted in the output	27
4.3	Diving in to the OxThesis LaTeX template	28
4.4	Collaborative writing	28
5	Customisations and extensions	29
5.1	Embedding PDF documents as chapters	29
5.2	Customizing referencing	33
5.2.1	Using a .csl file with pandoc instead of biblatex	33
5.2.2	Customizing biblatex and adding chapter bibliographies	33
5.3	Customizing the page headers and footers	36
	Conclusion	39

References	39
Appendices	
A The First Appendix	42
B The Second Appendix, for Fun	43

List of Figures

2.1	Code chunk syntax	10
2.2	Oxford logo	11
2.3	Oxford logo, rotated	12
2.4	A ggplot of car stuff	13
3.1	The ‘citr’ add-in	20
3.2	A marvel-lous meme	22

List of Tables

2.1	A knitr kable table	14
3.1	Stopping cars	23

Introduction

Welcome to the *R Markdown* Oxford University thesis template. This sample content is adapted from `thesisdown` and the formatting of PDF output is adapted from the OxThesis LaTeX template. Hopefully, writing your thesis in R Markdown will provide a nicer interface to the OxThesis template if you haven't used TeX or LaTeX before. More importantly, using *R Markdown* allows you to embed chunks of code directly into your thesis and generate plots and tables directly from the underlying data, avoiding copy-paste steps. This will get you into the habit of doing reproducible research, which benefits you long-term as a researcher, but also will greatly help anyone that is trying to reproduce or build upon your results down the road.

Using LaTeX together with *Markdown* is more consistent than the output of a word processor, much less prone to corruption or crashing, and the resulting file is smaller than a Word file. While you may never have had problems using Word in the past, your thesis is likely going to be about twice as large and complex as anything you've written before, taxing Word's capabilities.

Why use it?

R Markdown creates a simple and straightforward way to interface with the beauty of LaTeX. Packages have been written in **R** to work directly with LaTeX to produce nicely formatting tables and paragraphs. In addition to creating a user friendly interface to LaTeX, *R Mark-*

down allows you to read in your data, analyze it and to visualize it using **R**, **Python** or other languages, and provide documentation and commentary on the results of your project. Further, it allows for results of code output to be passed inline to the commentary of your results. You'll see more on this later, focusing on **R**. If you are more into **Python** or something else, you can still use *R Markdown* - see 'Other language engines' in Yihui Xie's *R Markdown: The Definitive Guide*.

Who should use it?

Anyone who needs to use data analysis, math, tables, a lot of figures, complex cross-references, or who just cares about reproducibility in research can benefit from using *R Markdown*. If you are working in 'softer' fields, the user-friendly nature of the *Markdown* syntax and its ability to keep track of and easily include figures, automatically generate a table of contents, index, references, table of figures, etc. should still make it of great benefit to your thesis project.

Chapter One

R Markdown Basics: The Markdown syntax

Here is a brief introduction to using *R Markdown*. *Markdown* is a simple formatting syntax for authoring HTML, PDF, and MS Word documents and much, much more. *R Markdown* provides the flexibility of *Markdown* with the implementation of **R** input and output. For more details on using *R Markdown* see <http://rmarkdown.rstudio.com>.

Be careful with your spacing in *Markdown* documents. While whitespace largely is ignored, it does at times give *Markdown* signals as to how to proceed. As a habit, try to keep everything left aligned whenever possible, especially as you type a new paragraph. In other words, there is no need to indent basic text in the Rmd document (in fact, it might cause your text to do funny things if you do).

1.1 Markdown basic syntax

1.1.1 Italics and bold

- *Italics* are done like `*this*` or `_this_`
- **Bold** is done like `**this**` or `__this__`
- ***Bold and italics*** is done like `***this***`, `___this___`, or (the most transparent

solution, in my opinion) **`**_this_`**

1.1.2 Inline code

- `Inline code` is created with backticks like ``this``

1.1.3 Sub and superscript

Sub₂ and super² script is created like `this~2~` and `this^2^`

1.1.4 Strikethrough

- ~~Strikethrough~~ is done `~~like this~~`

1.1.5 ‘Escaping’ (aka “What if I need an actual asterisk?”)

- To include an actual `*`, `_` or `\`, add another `\` in front of them: `*`, `_`, `\\`

1.1.6 Endash (–), emdash (—)

- `–` and `—` with `--` and `---`

1.1.7 Blockquotes

Do like this:

Put a `>` in front of the line.

1.1.8 Headings

- are done with `#`’s of increasing number, i.e.

– `#` First-level heading

- ## Second-level heading
- ### Etc.

In PDF output, a level-five heading will turn into a paragraph heading, i.e. `\paragraph{My level-five heading}`, which appears as bold text on the same line as the subsequent paragraph.

1.1.9 Lists

Unordered list by starting a line with an `*` or a `-`:

- Item 1
- Item 2

Ordered lists by starting a line with a number:

1. Item 1
2. Item 2

Notice that you can mislabel the numbers and *Markdown* will still make the order right in the output.

To create a sublist, indent the values a bit (at least four spaces or a tab):

1. Item 1
2. Item 2
3. Item 3
 - Item 3a
 - Item 3b

1.1.10 Line breaks

The official *Markdown* way to create line breaks is by ending a line with more than two spaces.

Roses are red. Violets are blue.

This appears on the same line in the output, because we didn't add spaces after red.

Roses are red.

Violets are blue.

This appears with a line break because I added spaces after red.

I find this is confusing, so I recommend the alternative way: Ending a line with a backslash will also create a linebreak:

Roses are red.

Violets are blue.

To create a new paragraph, you put a blank line.

Therefore, this line starts its own paragraph.

1.1.11 Hyperlinks

- This is a hyperlink created by writing the text you want turned into a clickable link in [square brackets followed by a](https://hyperlink-in-parentheses)

1.1.12 Footnotes

- Are created¹ by writing either `^[my footnote text]` for supplying the footnote content inline, or something like `[^a-random-footnote-label]` and supplying the text elsewhere in the format shown below ²:

`[^a-random-footnote-label]: This is a random test.`

¹my footnote text

²This is a random test.

1.1.13 Comments

To write comments within your text that won't actually be included in the output, use the same syntax as for writing comments in HTML. That is, `<!-- this will not be included in the output -->`.

1.1.14 Math

The syntax for writing math is stolen from LaTeX. To write a math expression that will be shown **inline**, enclose it in dollar signs. - This: `$A = \pi*r^{2}$` Becomes: $A = \pi * r^2$

To write a math expression that will be shown in a block, enclose it in two dollar signs.

This: `$$A = \pi*r^{2}$$`

Becomes:

$$A = \pi * r^2$$

To create numbered equations, put them in an 'equation' environment and give them a label with the syntax `(\#eq:label)`, like this:

```
\begin{equation}
  f\left(k\right) = \binom{n}{k} p^k\left(1-p\right)^{n-k}
  \label{eq:binom}
\end{equation}
```

Becomes:

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{1.1}$$

For more (e.g. how to theorems), see e.g. the documentation on bookdown.org

1.2 Additional resources

- *R Markdown: The Definitive Guide* - <https://bookdown.org/yihui/rmarkdown/>

- *R for Data Science* - <https://r4ds.had.co.nz>

Chapter Two

Adding code

The magic of R Markdown is that we can add code within our document to make it dynamic.

We do this either as *code chunks* (generally used for loading libraries and data, performing calculations, and adding images, plots, and tables), or *inline code* (generally used for dynamically reporting results within our text).

2.1 Code chunks

The syntax of a code chunk is shown in Figure 2.1.

Common chunk options include (see e.g. bookdown.org):

- **echo**: whether or not to display code in knitted output
- **eval**: whether or to run the code in the chunk when knitting
- **include**: wheter to include anything from the from a code chunk in the output document
- **fig.cap**: figure caption
- **fig.scap**: short figure caption, which will be used in the ‘List of Figures’ in the PDF front matter

IMPORTANT: Do *not* use underscoores in your chunk labels - if you do, you are likely to get an error in PDF output saying something like “! Package caption Error: \caption

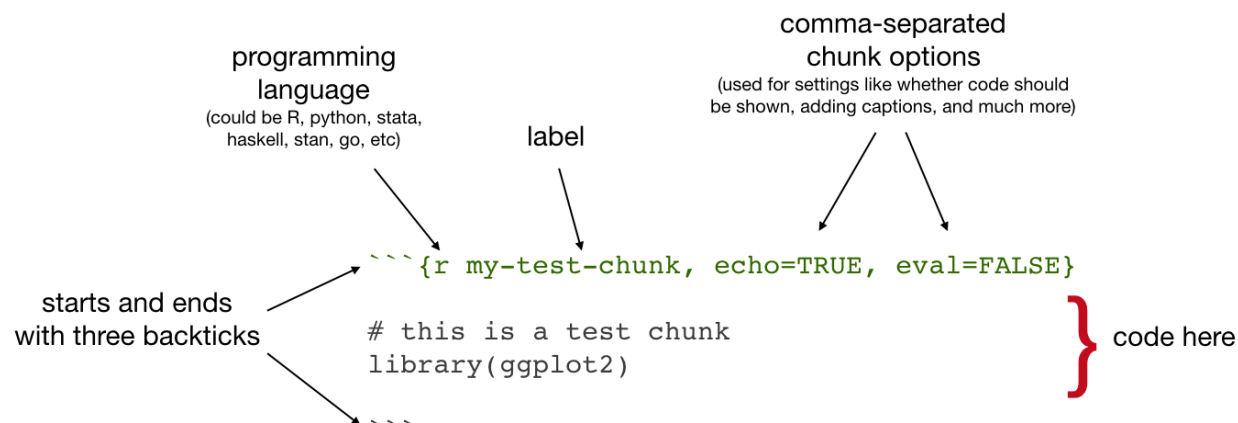


Figure 2.1 Code chunk syntax

outside float”.

2.1.1 Setup chunks

An R Markdown document usually begins with a chunk that is used to **load libraries**, and to **set default chunk options** with `knitr::opts_chunk$set`.

In your thesis, this will probably happen in **index.Rmd** and/or as opening chunks in each of your chapters.

```

““{r setup, include=FALSE}
# don't show code unless we explicitly set echo = TRUE
knitr::opts_chunk$set(echo = FALSE)

library(tidyverse)
““

```

2.1.2 Including images

Code chunks are also used for including images, with `include_graphics` from the `knitr` package, as in Figure 2.2



Figure 2.2 Oxford logo

```
knitr::include_graphics("figures/beltcrest.png")
```

Useful chunk options for figures include:

- `out.width` (use with a percentage) for setting the image size
- if you've got an image that gets waaay to big in your output, it will be constrained to the page width by setting `out.width = "100%"`

Figure rotation

You can use the chunk option `out.extra` to rotate images.

The syntax is different for LaTeX and HTML, so for ease we might start by assigning the right string to a variable that depends on the format you're outputting to:

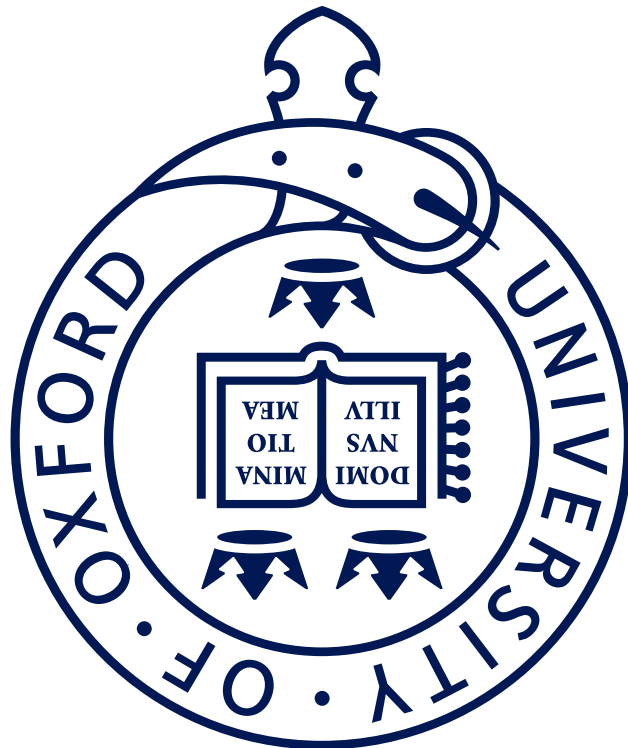


Figure 2.3 Oxford logo, rotated

```
if (knitr::opts_knit$get('rmarkdown.pandoc.to') == 'latex'){
  rotate180 <- "angle=180"
} else {
  rotate180 <- "style='transform:rotate(180deg);'"
}
```

Then you can reference that variable as the value of `out.extra` to rotate images, as in Figure 2.3.

2.1.3 Including plots

Similarly, code chunks are used for including dynamically generated plots. You use ordinary code in R or other languages - Figure 2.4 shows a plot of the `cars` dataset of stopping distances for cars at various speeds (this dataset is built in to **R**).

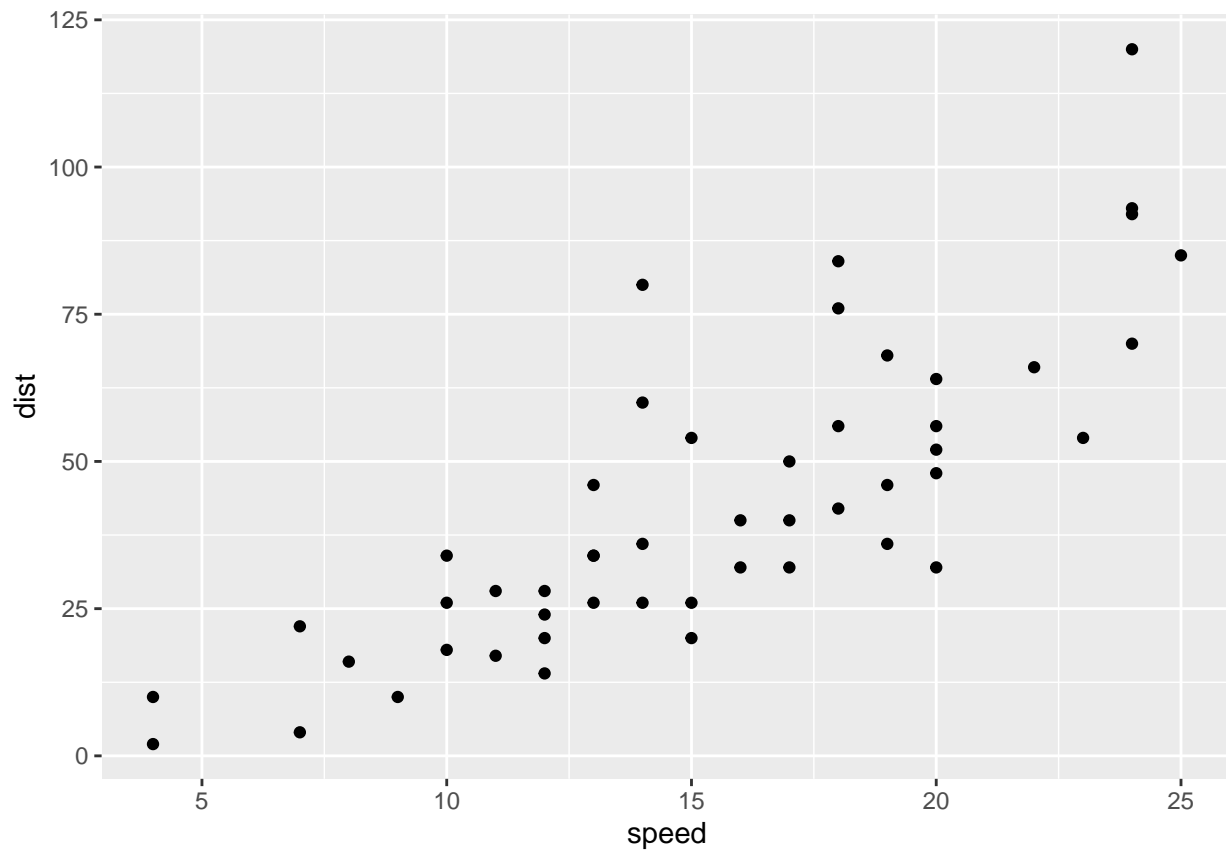


Figure 2.4 A ggplot of car stuff

```
cars %>%  
  ggplot() +  
    aes(x = speed, y = dist) +  
    geom_point()
```

Under the hood, plots are included in your document in the same way as images - when you build the book or knit a chapter, the plot is automatically generated from your code, saved as an image, then included into the output document.

2.1.4 Including tables

Tables are usually included with the `kable` function from the `knitr` package.

Table 2.1 A knitr kable table

speed	dist
4	2
4	10
7	4
7	22
8	16
9	10

Table 2.1 shows the first rows of that cars data - read in your own data, then use this approach to automatically generate tables.

```
cars %>%
  head() %>%
  knitr::kable(caption = "A knitr kable table")
```

- Gotcha: when using `kable`, captions are set inside the `kable` function
- The `kable` package is often used with the `kableExtra` package

2.1.5 A note on content positioning

One thing that may be annoying is the way *R Markdown* handles “floats” like tables and figures.

In your PDF output, LaTeX will try to find the best place to put your object based on the text around it and until you’re really, truly done writing you should just leave it where it lies.

When the time comes for you to make final tweaks to content positioning, read the relevant R Markdown documentation to see if there are easy ways to do what you want.

If you have very specific needs, you might have to read up on LaTeX (https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions) for your PDF output and/or on how to style HTML documents with CSS for your gitbook output.

2.2 Inline code

‘Inline code’ simply means inclusion of code inside text.

The syntax for doing this is ``r R_CODE``

For example, ``r 4 + 4`` would output 8 in your text.

You will usually use this in parts of your thesis where you report results - read in data or results in a code chunk, store things you want to report in a variable, then insert the value of that variable in your text.

For example, we might assign the number of rows in the `cars` dataset to a variable:

```
num_car_observations <- nrow(cars)
```

We might then write:

“In the `cars` dataset, we have ``r num_car_observations`` observations.”

Which would output:

“In the `cars` dataset, we have 50 observations.”

2.2.1 Referring to results computed in other languages than R

I’ve commented the below section out, to avoid compilation errors from the `reticulate` package being unable to find a python installation (after I installed MacOS Catalina, `reticulate` was unable to select a python version on my system, and I had to set it manually with `use_python()`).

If you need to use other languages, have a look at the content I commented out by the

end of the **02-rmd-basics-code.Rmd** file, which gives an example of using Python in your R Markdown file.

Chapter Three

Citations and cross-references

3.1 Citations

The usual way to include citations in an *R Markdown* document is to put references in a plain text file with the extension **.bib**, in **BibTeX** format.¹ Then reference the path to this file in **index.Rmd**'s YAML header with **bibliography:** **example.bib**.

Most reference managers can create a **.bib** file with your references automatically. However, the **by far** best reference manager to use with *R Markdown* is Zotero with the Better BibTeX plug-in, because the **citr** plugin for RStudio (see below) can read references directly from your Zotero library!

Here is an example of an entry in a **.bib** file:

```
@article{Shea2014,  
  author =      {Shea, Nicholas and Boldt, Annika},  
  journal =      {Trends in Cognitive Sciences},  
  pages =        {186--193},  
  title =        {{Supra-personal cognitive control}},  
  volume =       {18},
```

¹The bibliography can be in other formats as well, including EndNote (**.enl**) and RIS (**.ris**), see rmarkdown.rstudio.com/authoring_bibliographies_and_citations.

```
year =      {2014},  
doi =      {10.1016/j.tics.2014.01.006},  
}
```

In this entry highlighted section, ‘Shea2014’ is the **citation identifier**. To default way to cite an entry in your text is with this syntax: `[@citation-identifier]`.

So I might cite some things (Shea et al., 2014; Lottridge et al., 2012).

3.1.1 PDF output

In PDF output, the bibliography is handled by the OxThesis LaTeX template. If you set `bib-humanities: true` in `index.Rmd`, then in-text references will be formatted as author-year; otherwise references will be shown as numbers.

If you choose author-year formatting, a number of variations on the citation syntax are useful to know:

- Put author names outside the parenthesis
 - This: `@Shea2014 says blah`.
 - Becomes: Shea et al. (2014) says blah.
- Include only the citation-year (in parenthesis)
 - This: `Shea et al. says blah [-@Shea2014]`
 - Becomes: Shea et al. says blah (2014)
- Add text and page or chapter references to the citation
 - This: `[see @Shea2014, pp. 33-35; also @Wu2016, ch. 1]`
 - Becomes: Blah blah (see Shea et al., 2014, pp. 33-35; also Wu, 2016, ch. 1).

3.1.2 Gitbook output

In gitbook output, citations are by default inserted in the Chicago author-date format.

To change the format, add `csl: some-other-style.csl` in **index.Rmd**'s YAML header.

You can browse through and download styles at zotero.org/styles.

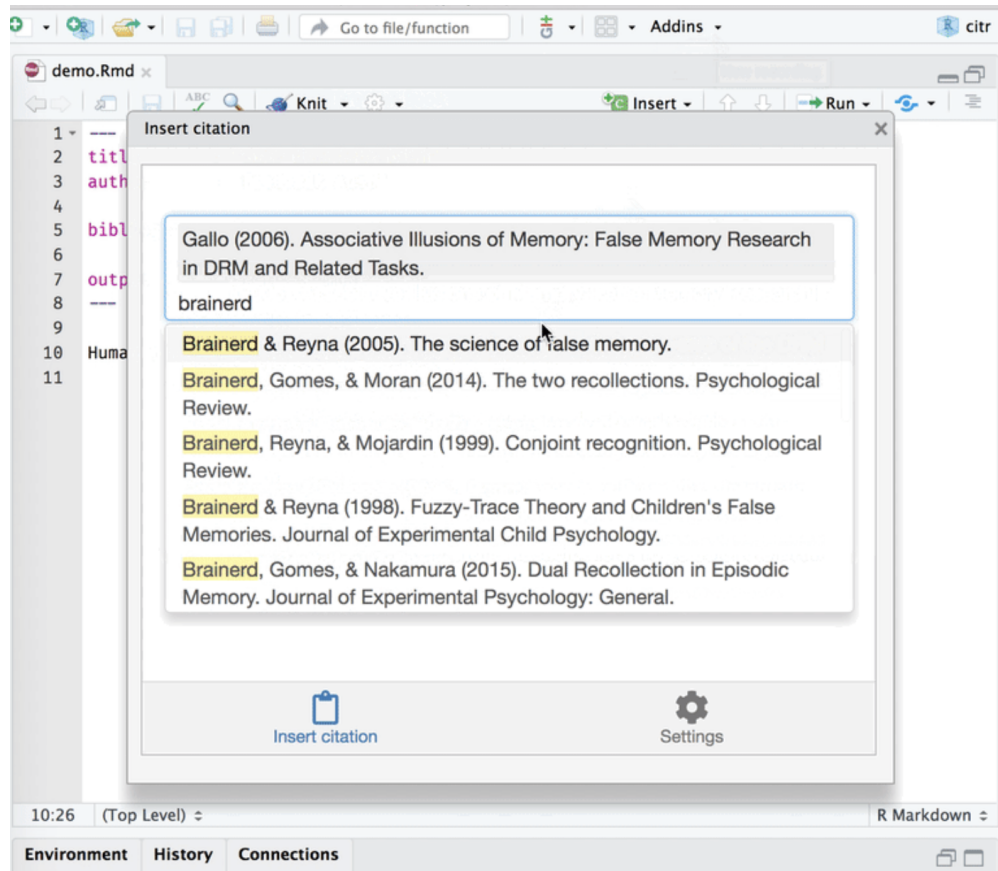


Figure 3.1 The ‘citr’ add-in

3.1.3 Insert references easily with the citr add-in

For an easy way to insert citations, try the `citr` RStudio add-in (Figure 3.1). You can install this add-in by typing `install.packages("citr")` in the R Console.

3.2 Cross-referencing

We can make cross-references to **sections** within our document, as well as to **figures** (images and plots) and **tables**.

The general cross-referencing syntax is `\@ref(label)`

3.2.1 Section references

Headers are automatically assigned a reference label, which is the text in lower caps separated by dashes. For example, `# My header` is automatically given the label `my-header`. So `# My header` can be referenced with `\@ref(my-section)`

Remember what we wrote in section 3.1?

We can also use **hyperlink syntax** and add `#` before the label, though this is only guaranteed to work properly in HTML output:

- So if we write `Remember what we wrote up in [the previous section](#citations)?`
- It becomes `Remember what we wrote up in the previous section?`

Creating custom labels

It is a very good idea to create **custom labels** for our sections. This is because the automatically assigned labels will change when we change the titles of the sections - to avoid this, we can create the labels ourselves and leave them untouched if we change the section titles.

We create custom labels by adding `{#label}` after a header, e.g. `# My section {#my-label}`. See our chapter title for an example. That was section 3.

3.2.2 Figure (image and plot) references

- To refer to figures (i.e. images and plots) use the syntax `\@ref(fig:label)`
- **GOTCHA:** Figures and tables must have captions if you wish to cross-reference them.

Let's add an image:

```
knitr::include_graphics("figures/captain.jpeg")
```

We refer to this image with `\@ref(fig:captain)`. So Figure 3.2 is this image.

And in Figure 2.4 we saw a cars plot.



Figure 3.2 A marvel-lous meme

Table 3.1 Stopping cars

speed	dist
4	2
4	10
7	4
7	22
8	16

3.2.3 Table references

- To refer to tables use the syntax `\@ref(tab:label)`

Let's include a table:

```
knitr::kable(cars[1:5,],
             caption="Stopping cars")
```

We refer to this table with `\@ref(tab:cars-table2)`. So Table 3.1 is this table.

And in Table 2.1 we saw more or less the same cars table.

3.2.4 Including page numbers

Finally, in the PDF output we might also want to include the page number of a reference, so that it's easy to find in physical printed output. LaTeX has a command for this, which looks like this: `\pageref{fig/tab:label}` (note: curly braces, not parentheses)

When we output to PDF, we can use raw LaTeX directly in our .Rmd files. So if we wanted to include the page of the cars plot we could write:

- This: Figure `\@ref(fig:cars-plot)` on page `\pageref(fig:cars-plot)`
- Becomes: Figure 2.4 on page 13

Include page numbers only in PDF output

A problem here is that LaTeX commands don't display in HTML output, so in the gitbook output we'd see simply "Figure 2.4 on page".

One way to get around this is to use inline R code to insert the text, and use an `ifelse` statement to check the output format and then insert the appropriate text.

- So this: ``r ifelse(knitr::is_latex_output(), "Figure \@ref(fig:cars-plot) on page \pageref{fig:cars-plot}", "")``
- Inserts this (check this on both PDF and gitbook): Figure 2.4 on page 13

Note that we need to escape the backslash with another backslash here to get the correct output.

3.3 Customising your thesis' front matter 'n stuff

3.3.1 Shorten captions shown in the list of figures (PDF)

You might want your list of figures (which follows the table of contents) to have shorter (or just different) figure descriptions than the actual figure captions.

Do this using the chunk option `fig.scap` ('short caption'), for example `{r captain-image, fig.cap="A very long and descriptive (and potentially boring) caption that doesn't fit in the list of figures, but helps the reader understand what the figure communicates" fig.scap="A concise description for the list of figures"}`

3.3.2 Shorten captions shown in the list of tables (PDF)

You might want your list of tables (which follows the list of figures in your thesis front matter) to have shorter (or just different) table descriptions than the actual table captions.

If you are using `knitr::kable` to generate a table, you can do this with the argument `caption.short`, e.g.:

```
knitr::kable(mtcars,
             caption = "A very long and descriptive (and potentially
             boring) caption that doesn't fit in the list of figures,
             but helps the reader understand what the figure
             communicates.",
             caption.short = "A concise description for the list of tables")
```

3.3.3 Shorting the running header (PDF)

You might want a chapter's running header (i.e. the header showing the title of the current chapter at the top of page) to be shorter (or just different) to the actual chapter title.

Do this by adding the latex command `\chaptermark{My shorter version}` after your chapter title.

For example, this chapter's running header is simply 'Cites and cross-refs', because it begins like this:

```
# Citations and cross-references {#cites-and-refs}
\chaptermark{Cites and cross-refs}
```

Chapter Four

Final Notes on The OxThesis template and on collaboration

4.1 Beginning chapters with quotes

The OxThesis LaTeX template lets you inject some wittiness into your thesis by including a block of type `savequote` at the beginning of chapters. To do this, use the syntax ````\block type='savequote'}`.¹

Add the reference for the quote with the chunk option `quote_author="my author name"`. You will also want to add the chunk option `include=knitr::is_latex_output()` so that quotes are only included in PDF output.

It's not possible to use markdown syntax inside chunk options, so if you want to e.g. italicise a book name in the reference use a 'text reference': Create a named piece of text with `'(ref:label-name) My text'`, then point to this in the chunk option with `quote_author='(ref:label-name)'`

¹For more on custom block types, see the relevant section in *Authoring Books with R Markdown*.

4.2 Highlighting corrections

For when it comes time to do corrections, you may want to highlight changes made when you submit a post-viva, corrected copy to your examiners so they can quickly verify you've completed the task. You can do so like this:

4.2.1 Short, inline corrections

Highlight **short, inline corrections** by doing `[like this]{.correction}` — the text between the square brackets will then **be highlighted in blue** in the output.

4.2.2 Blocks of added or changed material

Highlight entire **blocks of added or changed material** by putting them in a block of type `correction`, using the syntax ````{block type='correction'}`.² Like so:

For larger chunks, like this paragraph or indeed entire figures, you can use the ‘correc

4.2.3 Stopping corrections from being highlighted in the output

For **PDF** output, go to `index.Rmd` and (i) set `corrections: false` under `params` in the YAML header (stops block of corrections from being highlighted), (ii) comment out `pandoc_args: ["--lua-filter=scripts_and_filters/correction_filter.lua"]` (stops inline corrections from being highlighted).

For **gitbook** output, go to `style.css` and comment out the styling for `.correction`.

²In the `.tex` file for PDF output, this will put the content between `\begin{correction}` and `\end{correction}`; in gitbook output it will be put between `<div class="correction">` and `</div>`.

4.3 Diving in to the OxThesis LaTeX template

For LaTeX minded people, you can read through `templates/template.tex` to see which additional customisation options are available as well as `templates/ociamthesis.cls` which supplies the base class. For example, `template.tex` provides an option for master's degree submissions, which changes identifying information to candidate number and includes a word count. At the time of writing, you must set this directly in `template.tex` rather than from the YAML header in `index.Rmd`.

4.4 Collaborative writing

Best practices for collaboration and change tracking when using R Markdown are still an open question. In the blog post **One year to dissertate** by Lucy D'Agostino, which I highly recommend, the author notes that she knits `.Rmd` files to a `word_document`, then uses the `googledrive` R package to send this to Google Drive for comments / revisions from co-authors, then incorporates Google Drive suggestions *by hand* into the `.Rmd` source files. This is a bit clunky, and there are ongoing discussions among the *R Markdown* developers about what the best way is to handle collaborative writing (see issue #1463 on GitHub, where CriticMarkup is among the suggestions).

For now, this is an open question in the community of R Markdown users. I often knit to a format that can easily be imported to Google Docs for comments, then go over suggested revisions and manually incorporate them back in to the `.Rmd` source files. For articles, I sometimes upload a near-final draft to Overleaf, then collaboratively make final edits to the \LaTeX file there. I suspect some great solution will be developed in the not-to-distant future, probably by the RStudio team.

Chapter Five

Customisations and extensions

This chapter describes a number of possible customizations to the `oxforddown` thesis.

5.1 Embedding PDF documents as chapters

You may want to embed existing PDF documents into the thesis, for example if your department allows a ‘portfolio’ style thesis and you need to include an existing typeset publication as a chapter.

In gitbook output, you can simply use `knitr::include_graphics` and it should include a scrollable (and downloadable) PDF. You will probably want to set the chunk options `out.width='100%'` and `out.height='1000px'`:

```
knitr::include_graphics("figures/pdf_example/Lyngs2020_FB.pdf")
```

In LaTeX output, however, this approach can cause odd behaviour. Therefore, when you build your thesis to PDF, split the PDF into an alphanumerically sorted sequence of **single-page** PDF files (you can do this automatically with the package `pdftools`). You can then use the appropriate LaTeX command to insert them, as shown below (for brevity, in the `oxforddown` PDF sample content we’re only including two pages) *Note that the chunk option `results='asis'` must be set.* You may also want to remove margins from the PDF

files, which you can do with Adobe Acrobat (paid version) and likely other software.

```
# install.packages(pdftools)
# split PDF into pages stored in figures/pdf_example/split/
# pdftools::pdf_split("figures/pdf_example/Lyngs2020_FB.pdf",
# output = "figures/pdf_example/split/")

# grab the pages
pages <- list.files("figures/pdf_example/split", full.names = TRUE)

# set how wide you want the inserted PDFs to be:
# 1.0 is 100 per cent of the oxforddown PDF page width;
# you may want to make it a bit bigger
pdf_width <- 1.2

# for each PDF page, insert it nicely and
# end with a page break
cat(stringr::str_c("\\newpage \\begin{center}
\\makebox[\\linewidth][c]{\\includegraphics[width=", pdf_width,
"\\linewidth]{", pages, "}} \\end{center}"))
```

‘I Just Want to Hack Myself to Not Get Distracted’: Evaluating Design Interventions for Self-Control on Facebook

Ulrik Lyngs¹, Kai Lukoff², Petr Slovak³, William Seymour¹, Helena Webb¹,
Marina Jirotko¹, Jun Zhao¹, Max Van Kleek¹, Nigel Shadbolt¹

¹Department of Computer Science, University of Oxford, UK, {first.last}@cs.ox.ac.uk

²Human Centered Design & Engineering, University of Washington, Seattle, US, kail@uw.edu

³Department of Informatics, King’s College London, UK, petr.slovak@kcl.ac.uk

ABSTRACT

Beyond being the world’s largest social network, Facebook is for many also one of its greatest sources of digital distraction. For students, problematic use has been associated with negative effects on academic achievement and general wellbeing. To understand what strategies could help users regain control, we investigated how simple interventions to the Facebook UI affect behaviour and perceived control. We assigned 58 university students to one of three interventions: goal reminders, removed newsfeed, or white background (control). We logged use for 6 weeks, applied interventions in the middle weeks, and administered fortnightly surveys. Both goal reminders and removed newsfeed helped participants stay on task and avoid distraction. However, goal reminders were often annoying, and removing the newsfeed made some fear missing out on information. Our findings point to future interventions such as controls for adjusting types and amount of available information, and flexible blocking which matches individual definitions of ‘distraction’.

Author Keywords

Facebook; problematic use; self-control; distraction; ICT non-use; addiction; focus; interruptions

CCS Concepts

•Human-centered computing → Empirical studies in HCI;

INTRODUCTION

Research on ‘Problematic Facebook Use’ (PFU) has investigated correlations between Facebook use and negative effects on outcomes such as level of academic achievement [35] and subjective wellbeing [58, 57]. A cross-cutting finding is that negative outcomes are associated with difficulty at exerting self-control over use, as well as specific use patterns including viewing friends’ wide-audience broadcasts rather than receiving targeted communication from strong ties [13, 58].

Much of this work has focused on self-control over Facebook use in student populations [2, 44, 46], with media multitasking research finding that students often give in to use which provides short-term ‘guilty pleasures’ over important, but aversive academic tasks [76, 88, 60]. In the present paper, we present a mixed-methods study exploring how two interventions to Facebook — goal reminders and removing the newsfeed — affect university students’ patterns of use and perceived control over Facebook use. To triangulate self-report with objective measurement, our study combined usage logging with fortnightly surveys and post-study interviews.

We found that both interventions helped participants stay on task and use Facebook more in line with their intentions. In terms of use patterns, goal reminders led to less scrolling, fewer and shorter visits, and less time on site, whereas removing the newsfeed led to less scrolling, shorter visits, and less content ‘liked’. However, goal reminders were often experienced as annoying, and removing the newsfeed made some participants fear missing out on information. After the study, participants suggested a range of design solutions to mitigate self-control struggles on Facebook, including controls for filtering or removing the newsfeed, reminders of time spent and of use goals, and removing features that drive engagement. As an exploratory study, this work should be followed by confirmatory studies to assess whether our findings replicate, and how they may generalise beyond a student population.

RELATED WORK

Struggles with Facebook use

Whereas many uses of Facebook offer important benefits, such as social support, rapid spread of information, or facilitation of real-world interactions [78], a substantial amount of research has focused on negative aspects [58]. For example, studies have reported correlations between patterns of Facebook use and lower academic achievement [77, 86], low self-esteem, depression and anxiety [51], feelings of isolation and loneliness [2], and general psychological distress [15]. Such ‘Problematic Facebook Use’ (PFU) has been studied under various names (including ‘Facebook dependence’ [87] and ‘Facebook addiction’ [5]), but a recent review summarised a common definition as ‘problematic behaviour characterised by addictive-like symptoms and/or self-regulation difficulties related to Facebook use leading to negative consequences in personal and social life’ [58].

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
CHI ’20, April 25–30, 2020, Honolulu, HI, USA.
© 2020 Copyright is held by the author/owner(s).
ACM ISBN 978-1-4503-6708-0/20/04.
<http://dx.doi.org/10.1145/3313831.3376672>

REFERENCES

- [1] Alexander T. Adams, Jean Costa, Malte F. Jung, and Tanzeem Choudhury. 2015. Mindless Computing: Designing Technologies to Subtly Influence Behavior. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 719–730. DOI: <http://dx.doi.org/10.1145/2750858.2805843>
- [2] Sami Abdo Radman Al-Dubai, Kurubaran Ganasegeran, Mustafa Ahmed Mahdi Al-Shagga, Hematram Yadav, and John T. Arokiasamy. 2013. Adverse Health Effects and Unhealthy Behaviors among Medical Students Using Facebook. <https://www.hindawi.com/journals/tswj/2013/465161/>. (2013). DOI: <http://dx.doi.org/10.1155/2013/465161>
- [3] All Party Parliamentary Group on Social Media and Young People's Mental Health and Wellbeing. 2019. *#NewFilters to Manage the Impact of Social Media on Young People's Mental Health and Wellbeing*. Technical Report. UK Parliament.
- [4] Hunt Allcott, Luca Braghieri, Sarah Eichmeyer, and Matthew Gentzkow. 2019. *The Welfare Effects of Social Media*. Working Paper 25514. National Bureau of Economic Research. DOI: <http://dx.doi.org/10.3386/w25514>
- [5] Cecilie Schou Andreassen, Torbjørn Torsheim, Geir Scott Brunborg, and Staale Pallesen. 2012. Development of a Facebook Addiction Scale. *Psychological Reports* 110, 2 (apr 2012), 501–517. DOI: <http://dx.doi.org/10.2466/02.09.18.PR0.110.2.501-517>
- [6] Yummy Apps. 2019. Todobook. (May 2019).
- [7] Albert Bandura. 1982. Self-efficacy mechanism in human agency. *American Psychologist* 37, 2 (1982), 122–147. DOI: <http://dx.doi.org/10.1037/0003-066x.37.2.122>
- [8] Fanni Bányai, Ágnes Zsila, Orsolya Király, Aniko Maraz, Zsuzsanna Elekes, Mark D. Griffiths, Cecilie Schou Andreassen, and Zsolt Demetrovics. 09-Jan-2017. Problematic Social Media Use: Results from a Large-Scale Nationally Representative Adolescent Sample. *PLOS ONE* 12, 1 (09-Jan-2017), e0169839. DOI: <http://dx.doi.org/10.1371/journal.pone.0169839>
- [9] Elliot T Berkman, Cendri A Hutcherson, Jordan L Livingston, Lauren E Kahn, and Michael Inzlicht. 2017. Self-Control as Value-Based Choice. *Current Directions in Psychological Science* 26, 5 (2017), 422–428. DOI: <http://dx.doi.org/10.1177/0963721417704394>
- [10] Walter R. Boot, Daniel J. Simons, Cary Stothart, and Cassie Stutts. 2013. The Pervasive Problem with Placebos in Psychology. *Perspectives on Psychological Science* 8, 4 (jul 2013), 445–454. DOI: <http://dx.doi.org/10.1177/1745691613491271>
- [11] Amara Brook. 2011. Ecological Footprint Feedback: Motivating or Discouraging? *Social Influence* 6, 2 (April 2011), 113–128. DOI: <http://dx.doi.org/10.1080/15534510.2011.566801>
- [12] Gharad Bryan, Dean Karlan, and Scott Nelson. 2010. Commitment Devices. *Annual Review of Economics* 2, 1 (Sept. 2010), 671–698. DOI: <http://dx.doi.org/10.1146/annurev.economics.102308.124324>
- [13] Moira Burke and Robert E. Kraut. 2016. The Relationship Between Facebook Use and Well-Being Depends on Communication Type and Tie Strength. *Journal of Computer-Mediated Communication* 21, 4 (2016), 265–281. DOI: <http://dx.doi.org/10.1111/jcc4.12162>
- [14] Moira Burke, Cameron Marlow, and Thomas Lento. 2010. Social Network Activity and Social Well-Being. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 1909–1912. DOI: <http://dx.doi.org/10.1145/1753326.1753613>
- [15] Wenhong Chen and Kye-Hyoung Lee. 2013. Sharing, Liking, Commenting, and Distressed? The Pathway between Facebook Interaction and Psychological Distress. *Cyberpsychology, Behavior and Social Networking* 16, 10 (oct 2013), 728–734. DOI: <http://dx.doi.org/10.1089/cyber.2012.0272>
- [16] Justin Cheng, Moira Burke, and Elena Goetz Davis. 2019. Understanding Perceptions of Problematic Facebook Use: When People Experience Negative Life Impact and a Lack of Control. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 199:1–199:13. DOI: <http://dx.doi.org/10.1145/3290605.3300429>
- [17] Jacob Cohen. 1992. A Power Primer. *Psychological Bulletin* 112, 1 (1992), 155–159. DOI: <http://dx.doi.org/10.1037/0033-2909.112.1.155>
- [18] Anna L Cox, Sandy J J Gould, Marta E Cecchinato, Ioanna Iacovides, and Ian Renfree. 2016. Design Frictions for Mindful Interactions: The Case for Microboundaries. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 1389–1397. DOI: <http://dx.doi.org/10.1145/2851581.2892410>
- [19] Helen Creswick, Liz Dowthwaite, Ansgar Koene, Elvira Perez Vallejos, Virginia Portillo, Monica Cano, and Christopher Woodard. 2019. "... They don't really listen to people". *Journal of Information, Communication and Ethics in Society* 17, 2 (May 2019), 167–182. DOI: <http://dx.doi.org/10.1108/jices-11-2018-0090>
- [20] Angela L. Duckworth, Katherine L. Milkman, and David Laibson. 2018. Beyond Willpower: Strategies for Reducing Failures of Self-Control. *Psychological Science in the Public Interest* 19, 3 (Dec. 2018), 102–129. DOI: <http://dx.doi.org/10.1177/1529100618821893>

5.2 Customizing referencing

5.2.1 Using a .csl file with pandoc instead of biblatex

The `oxforddown` package uses biblatex in latex for referencing. It is also possible to use pandoc for referencing by providing a .csl file in the YAML header of `index.Rmd` (likely requiring commenting out the biblatex code in `templates/template.tex`). This may be helpful for those who have a .csl file describing the referencing format for a particular journal. However, note that this approach does not support chapter bibliographies (see Section 5.2.2).

```
csl: ecology.csl
```

5.2.2 Customizing biblatex and adding chapter bibliographies

This section provides one example of customizing biblatex. Much of this code was combined from searches on Stack Exchange and other sources (e.g. [here](#)).

In `templates/template.tex`, one can replace the existing biblatex calls with the following to achieve referencing that looks like this:

(Charmantier and Gienapp 2014)

Charmantier, A. and P. Gienapp (2014). Climate change and timing of avian breeding and migration: evolutionary versus plastic changes. *Evolutionary Applications* 7(1):15–28. doi: 10.1111/eva.12126.

```
\usepackage[backend=biber,
  bibencoding=utf8,
  refsection=chapter, % referencing by chapter
  style=authoryear,
  firstinits=true,
```

```
isbn=false,
doi=true,
url=false,
eprint=false,
related=false,
dashed=false,
clearlang=true,
maxcitenames=2,
mincitenames=1,
maxbibnames=10,
abbreviate=false,
minbibnames=3,
uniquelist=minyear,
sortcites=true,
date=year
]{biblatex}
\AtEveryBibitem{%
  \clearlist{language}%
  \clearfield{note}
}

\DeclareFieldFormat{titlecase}{\MakeTitleCase{#1}}

\newrobustcmd{\MakeTitleCase}[1]{%
  \ifthenelse{\ifcurrentfield{booktitle}\OR\ifcurrentfield{booksubtitle}%
    \OR\ifcurrentfield{maintitle}\OR\ifcurrentfield{mainsubtitle}%
```

```

\OR\ifcurrentfield{journaltitle}\OR\ifcurrentfield{journalsubtitle}%
\OR\ifcurrentfield{issuetitle}\OR\ifcurrentfield{issuesubtitle}%
\OR\ifentrytype{book}\OR\ifentrytype{mvbook}\OR\ifentrytype{bookinbook}%
\OR\ifentrytype{booklet}\OR\ifentrytype{suppbook}%
\OR\ifentrytype{collection}\OR\ifentrytype{mvcollection}%
\OR\ifentrytype{suppcollection}\OR\ifentrytype{manual}%
\OR\ifentrytype{periodical}\OR\ifentrytype{suppperiodical}%
\OR\ifentrytype{proceedings}\OR\ifentrytype{mvproceedings}%
\OR\ifentrytype{reference}\OR\ifentrytype{mvreference}%
\OR\ifentrytype{report}\OR\ifentrytype{thesis}}
{#1}
{\MakeSentenceCase{#1}}}

% \renewbibmacro{in:}{}
% suppress "in" for articles
%
\renewbibmacro{in:}{%
  \ifentrytype{article}{}{\printtext{\bibstring{in}\intitlepunct}}}
%-- no "quotes" around titles of chapters/article titles
\DeclareFieldFormat{article, inbook, incollection, inproceedings, misc, thesis, unpublis
{title}{#1}
%-- no punctuation after volume
\DeclareFieldFormat{article}
{volume}{{#1}}
%-- puts number/issue between brackets
\DeclareFieldFormat{article, inbook, incollection, inproceedings, misc, thesis, unpublis

```

```

{number}{\mkbibparens{#1}}

%-- and then for articles directly the pages w/o any "pages" or "pp."

\DeclareFieldFormat[article]
{pages}{#1}

%-- for some types replace "pages" by "p."

\DeclareFieldFormat[inproceedings, incollection, inbook]
{pages}{p. #1}

%-- format 16(4):224--225 for articles

\renewbibmacro*{volume+number+eid}{
  \printfield{volume}%
  \printfield{number}%
  \printunit{\addcolon}
}

```

If you would like chapter bibliographies, in addition insert the following code at the end of each chapter, and comment out the entire REFERENCES section at the end of `template.tex`.

```
\printbibliography[segment=\therefsection,heading=subbibliography]
```

5.3 Customizing the page headers and footers

The following code, when it replaces the existing corresponding code block in **ociamthesis.cls**, puts chapter number and title centered in the header and page number in the footer, centered. This may be desirable particularly when inserting PDF chapters, as the margins of the PDF may not exactly align with the left and right margins of the page, demarcated by the existing header and footer text. In the following code block, the original code is commented out where replaced.

```

\usepackage{fancyhdr}
\setlength{\headheight}{15pt}
\fancyhf{} % clear the header and footers
\pagestyle{fancy}
\renewcommand{\chaptermark}[1]{\markboth{\thechapter. #1}{\thechapter. #1}}
% \renewcommand{\chaptermark}[1]{\markboth{\thechapter. #1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection. #1}}
\renewcommand{\headrulewidth}{0pt}
\fancyhead[C0]{\emph{\leftmark}}
\fancyhead[CE]{\emph{\rightmark}}
% \fancyhead[LO,RE]{}
% \fancyhead[LE,RO]{}
\fancyfoot[C0,CE]{\emph{\thepage}}

\fancypagestyle{plain}{\fancyhf{}\fancyfoot[C]{\emph{\thepage}}}

% JEM fix header on cleared pages for openright
\def\cleardoublepage{\clearpage\if@twoside \ifodd\c@page\else
  \hbox{}
  % \fancyhead[RE,LO]{}
  \fancyhead[CE,C0]{}
  \newpage
  \if@twocolumn\hbox{}\newpage
  \fi
  % \fancyhead[LO]{\emph{\leftmark}}
  % \fancyhead[RE]{\emph{\rightmark}}

```

```
\fancyhead[CO]{\emph{\leftmark}}  
\fancyhead[CE]{\emph{\rightmark}}  
\fi\fi}
```

Conclusion

If we don't want Conclusion to have a chapter number next to it, we can add the `{-}` attribute. (Goethe, 1829)

More info

And here's some other random info: the first paragraph after a chapter title or section head *shouldn't be* indented, because indents are to tell the reader that you're starting a new paragraph. Since that's obvious after a chapter or section title, proper typesetting doesn't add an indent there.

References

- Goethe, Johann Wolfgang von (1829). *Wilhelm Meisters Wanderjahre oder die Entsagenden*. de. Cotta.
- Lottridge, Danielle et al. (2012). “Browser design impacts multitasking”. In: *Proceedings of the Human Factors and Ergonomics Society 56th Annual Meeting*. ISBN: 9780945289418. DOI: 10.1177/1071181312561289.
- Shea, Nicholas et al. (2014). “Supra-personal cognitive control and metacognition”. In: *Trends in Cognitive Sciences* 18.4, pp. 186–193. ISSN: 1364-6613. DOI: 10.1016/j.tics.2014.01.006. URL: <http://dx.doi.org/10.1016/j.tics.2014.01.006>.
- Wu, Tim (2016). *The Attention Merchants: The Epic Scramble to Get Inside Our Heads*. Knopf Publishing Group.

APPENDICES

Appendix One

The First Appendix

This first appendix includes an R chunk that was hidden in the document (using `echo = FALSE`) to help with readability:

In 02-rmd-basics-code.Rmd

```
library(tidyverse)
knitr::include_graphics("figures/chunk-parts.png")
```

And here's another one from the same chapter, i.e. Chapter 2:

```
knitr::include_graphics("figures/beltcrest.png")
```

Appendix Two

The Second Appendix, for Fun