

# 1

## Citations, cross-references, and collaboration

### 1.1 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.<sup>1</sup> 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 you 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}},
```

---

<sup>1</sup>The bibliography can be in other formats as well, including EndNote (**.enl**) and RIS (**.ris**), see [rmarkdown.rstudio.com/authoring\\_bibliographies\\_and\\_citations](https://rmarkdown.rstudio.com/authoring_bibliographies_and_citations).

```

volume =      {18},
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 (**Shea2014**; **Lottridge2012**).

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

### 1.1.2 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](https://zotero.org/styles).

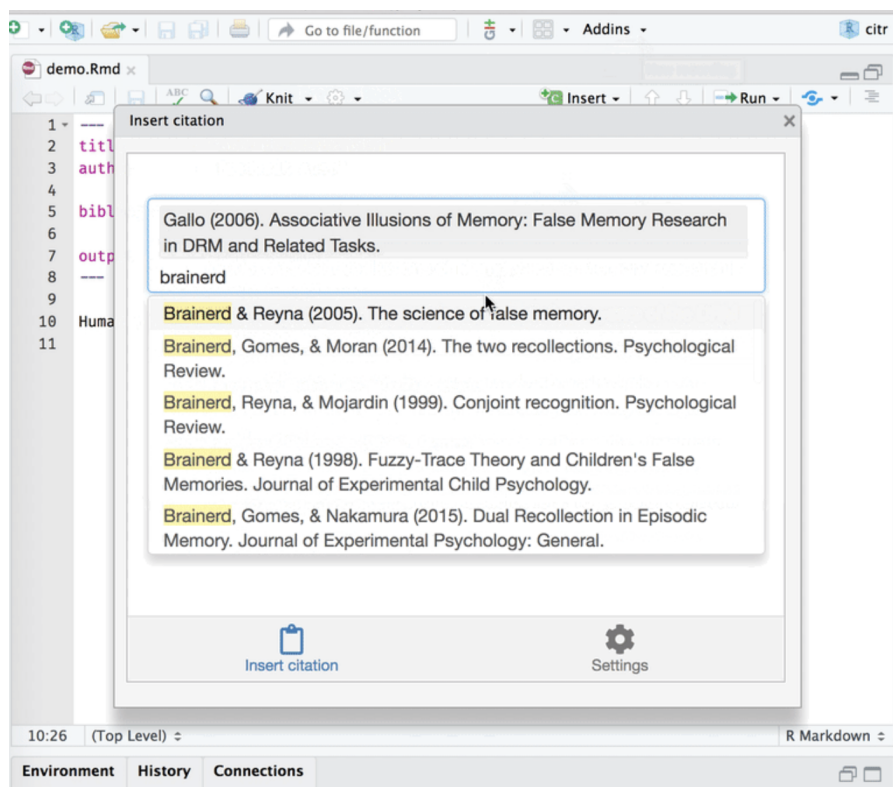


Figure 1.1: The ‘citr’ add-in

### 1.1.3 Insert references easily with the citr add-in

For an easy way to insert citations, try the `citr` RStudio add-in (Figure 1.1). You can install this add-in by typing `install.packages("citr")` in the R Console.

## 1.2 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)`

### 1.2.1 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 1.1?

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 1.

### 1.2.2 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/sample-content/captain.jpeg")
```

We refer to this image with `\@ref(fig:captain)`. So Figure 1.2 is this image. And in Figure ?? we saw a cars plot.

### 1.2.3 Table references

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

Let's include a table:



**Figure 1.2:** A marvel-lous meme

**Table 1.1:** Stopping cars

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

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

We refer to this table with `\@ref(tab:cars-table2)`. So Table 1.1 is this table. 1.1 magic reference

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

## 1.2.4 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 ?? on page ??

### 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 ?? 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 ?? on page ??

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

## 1.3 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.

## 1.4 Additional resources

- *R Markdown: The Definitive Guide* - <https://bookdown.org/yihui/rmarkdown/>
- *R for Data Science* - <https://r4ds.had.co.nz>