SI Carpentries Brown Bags - rmarkdown tutorial

Paula Pappalardo

2021-03-24

This is how we create titles

And subtitles

And so on...

An index is created based on our headings

Setup

First, we load some libraries we need, and create a subset of the *penguins* dataset that will be used for some examples. You can find more information about this dataset here.

```
library(knitr)
library(dplyr)
library(palmerpenguins)

# subset the penguins dataset for some rmarkdown examples

minipenguins <- penguins %>%
  dplyr::select(species, island, flipper_length_mm, body_mass_g) %>%
  dplyr::distinct(species, island, .keep_all = T)
```

Resources consulted for this tutorial

This is just my own summary to give you a little taste of the wonders of rmarkdown. Most of this information has been compiled from available tutorials or thanks to awesome people that share their knowledge on *stackoverflow.com* to solve specific problems.

Note how you can wrap word or sentences with one *asterisk* or a *dash* to make them *italics*. And when you use **two** asterisks or **two** dashes, you will make the words or sentence **bold**.

Some of the sources I have consulted:

- https://bookdown.org/yihui/rmarkdown-cookbook/how-to-read-this-book.html
- https://bookdown.org/yihui/rmarkdown/basics.html
- https://r4ds.had.co.nz/r-markdown.html

- https://rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf
- Slides showing other uses of rmarkdown: https://slides.yihui.org/2019-rmarkdown-RaukR-Yihui-Xie.html#18
- Interactive tutorial with markdown basics: https://commonmark.org/help/

Why do we want to learn rmarkdown?

Because it can combine text, citations, data, plots, code, and render into a professional looking document. For that, it uses:

markdown for the plain text formatting syntax and *R* for the code chunks to analyze and visualize data.

Below I used a chunk of text, so that the text will appear different when we render the document. Chunks of text start and end with three backticks.

```
code + narrative + plots + citations = nice and reproducible report
```

And you can also do slides, and more with rmarkdown, but today I will focus on the HTML, Word, and PDF outputs.

To me, the most important thing is that **everything is dynamic**, which gives clear advantages for reproducibility:

```
* your tables and plots will change when your data changes
* even the text can have inline dynamic variables that will change as well
```

Introduction

Getting started: what do you need

You need the latest version of R studio and R for the rmarkdown to work. The first time you try to open an "R markdown" file, it may ask you to download more packages, to help you set up with everything you need. From rmarkdown you can transform ("knit") your documents into different formats (e.g., Word, PDF, HTML). And then the first time you try to knit a document, it may prompt you to install a few more packages. After following the suggestions, knitting in the three formats should work.

Basic markdown functionality

The good thing of rmarkdown, is that if we want to write text... we just write! And a recent addition with the new R studio is the spell check functionality.

Using the R Studio tabs, if you go to "Help" you will find *Markdown Quick Reference*, if you go to "Help/Cheatsheets" you will also find *R markdown Cheat Sheet* and *R markdown Reference Guide*.

I showed earlier how to do a bulleted list, if you want to do numbered list, you directly put the numbers, and you can nest items using 2 TAB or 4 spaces:

1. Easy enumerated lists
2. Simple syntax

- nested lists
  - super nested!

3. You can add chunks of code by:

   - manually typing the chunk delimiters (three back ticks)
   - Using the "Insert" button icon in the editor toolbar.
   - The keyboard shortcut Cmd/Ctrl + Alt + I

The argument after {r} in the first line will indicate the chunk options, for example look at the same script with and without a trick to make it look nicer. The first word after the **r** is the label, and that is going to be very helpful to navigate between chunks.

After the chunk label {r mylabel,...}, you can add other chunk options, for example, if you want to evaluate the code or not, if you want to show your results or not, or some edition tricks as *comment* that is useful to avoid the # signs in the output.

Here the basic look if we want to show the minipenguins object:

```
minipenguins
```

```
## # A tibble: 5 x 4
##   species   island    flipper_length_mm body_mass_g
##   <fct>     <fct>                 <int>       <int>
## 1 Adelie    Torgersen               181        3750
## 2 Adelie    Biscoe                  174        3400
## 3 Adelie    Dream                   178        3250
## 4 Gentoo    Biscoe                  211        4500
## 5 Chinstrap Dream                   192        3500
```

Here a nicer look:

```
minipenguins
```

```
# A tibble: 5 x 4
  species   island    flipper_length_mm body_mass_g
  <fct>     <fct>                 <int>       <int>
1 Adelie    Torgersen               181        3750
2 Adelie    Biscoe                  174        3400
3 Adelie    Dream                   178        3250
4 Gentoo    Biscoe                  211        4500
5 Chinstrap Dream                   192        3500
```

Here a way to not display this chunk in the final report (*echo = FALSE*, you will not see the commands in the output, only the table):

```
## # A tibble: 5 x 4
##   species   island    flipper_length_mm body_mass_g
##   <fct>     <fct>                 <int>       <int>
## 1 Adelie    Torgersen               181        3750
## 2 Adelie    Biscoe                  174        3400
## 3 Adelie    Dream                   178        3250
## 4 Gentoo    Biscoe                  211        4500
## 5 Chinstrap Dream                   192        3500
```

Here a way of displaying the chunk but hiding the results (*results = 'hide'*)

```
minipenguins
```

You can also choose to show it in the final document without running it (*eval = FALSE*).

```
minipenguins
```

And you can write inline code that is visible such as `x <- 1:10`. Or calculate something directly from your dataset, such as there were 3 penguin species studied. This is how if your data changes, you can rerun your document and the text with your calculations will change automatically.

So, there are many ways to personalize your output, depending who is your audience. If there is someone that doesn't use R, you can hide your code and only show your results. You can also set up your preferred chunk options at the beginning of the document using **opts_chunk**:

```
knitr::opts_chunk$set(echo = F, warning = F, message = F, comment = "")
```

And if you need one chunk to be different, setting the option in that particular chunk will be counted over the global setup made at the begginng of the document.

Have in mind that if you have duplicated chunk labels, you will have an error when trying to knitt your document. The error will tell you which one is duplicated so you can find it. It happens **very** frequently because we tend to copy-paste chunks when we only want to tweak something. I just added this paragraph after my third time getting this error :)

Citations

Bibliography management can be also done with rmarkdown.

Set up to insert bibliography

You need two things:

1. A **.bib** file with your reference library

For example, I am now using Zotero, that is a free reference manager software. In Zotero, you can do *Edit/Select All* (or select the specific references you want), and then go to *File/Export Library*, selecting BibTeX from the dropdown menu. If you use Endnote, this link can help you to export a library in the bib.tex format. This is how a reference will look in the **.bib** file, in the first line, after { and before the comma, you can see the label for that particular reference, that you will use to cite it:

```
@article{fortin_species_2005,
    title = {Species' geographic ranges and distributional limits: pattern analysis and statistical issu
    volume = {108},
    issn = {00301299, 16000706},
    shorttitle = {Species' geographic ranges and distributional limits},
    url = {http://doi.wiley.com/10.1111/j.0030-1299.2005.13146.x},
    doi = {10.1111/j.0030-1299.2005.13146.x},
    language = {en},
    number = {1},
```

```
    urldate = {2018-06-01},
    journal = {Oikos},
    author = {Fortin, M.-J. and Keitt, T. H. and Maurer, B. A. and Taper, M. L. and Kaufman, Dawn M. and
    month = jan,
    year = {2005},
    pages = {7--17},
    file = {Fortin et al. - 2005 - Species' geographic ranges and distributional limi.pdf:C\:\\Users\\Pa
}
```

2. A **.cls** file that will indicate the journal style for the references. In this zotero link you can find hundreds of styles in *cls* format.

In the first lines of the markdown document (the YAML code), there is a section called *Bibliography* that allows you to indicate your **.bib** and **.cls** files. In this case I downloaded the **.csl** file for the ICES Journal of Marine Sciences style (*ices-journal-of-marine-science.csl*), and saved my references in a file called *References.bib*.

How to cite?

```
To add one citation, you use [@mycitation].

To separate multiple citations you use ';': [@burgman_bias_2003; @fortin_species_2005].
```

This will look like this (Burgman and Fox, 2003; Fortin *et al.*, 2005). To know which code corresponds to each of your citations, you will have to check your **.bib** file and look at the label for the reference you want to cite. You can also open the **.bib** file in R studio and have it on hand.

```
When you need to add comments within your citation, you can do so inside the square brackets:

Multiple citations with comments [see @cote_meta-analysis_2012, pp. 1-3; also see @pappalardo_comparing
```

Comments within citations will look like this (see Côté and Reynolds, 2012, pp. 1–3; also see Pappalardo *et al.*, 2020).

```
Sometimes we need to cite authors within the text, we can do that removing the square brackets:

  @gurevitch_meta_2001 said this
```

Which will look like Gurevitch *et al.* (2001) said this.

When you knit the document, the references will appear in the correct format, at the end of your document. This is why I ended my rmarkdown file with the header *# References*.

To get a **.bib** file for all the R packages you are using, you can use:

```
knitr::write_bib(c(.packages()), "packages.bib")
```

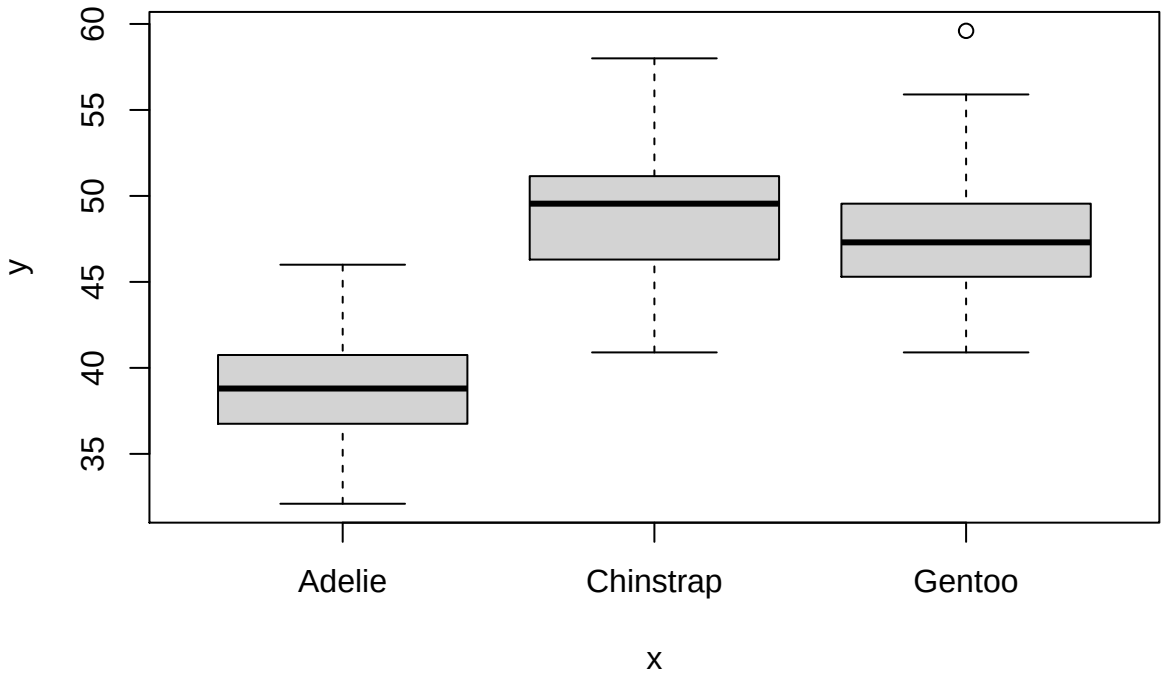
This will compile the citations for all the packages that are loaded into the file *packages.bib*. Note that many packages are loaded in the background, as dependencies of packages you called directly, so it may return a much bigger list than you thought, but they will be handy for you to cite them. You can have more than one **.bib** file in your YAML header, separated by commas.

Plots

Images generated by `knitr` are saved in a figures folder. However, they also appear to be represented in the HTML output using a data URI scheme. This means that you can paste the HTML into a blog post or discussion forum and you don't have to worry about finding a place to store the images; they're embedded in the HTML.

Here is a basic plot using base graphics:

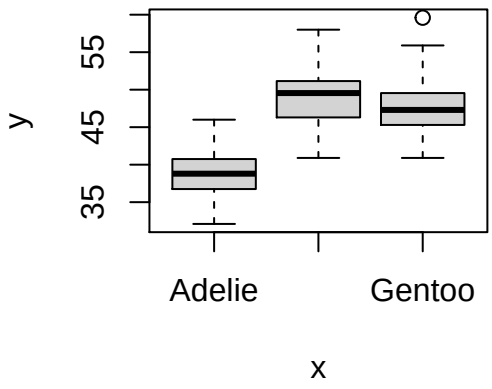
```
plot(penguins$species, penguins$bill_length_mm)
```



You can set up the plot size using the chunk options, for this example I added:

```
fig.width= 3, fig.height= 3
```

```
plot(penguins$species, penguins$bill_length_mm)
```



Equations

Equations are included by using LaTeX notation and including them either between single dollar signs (inline equations) or double dollar signs (displayed equations).

```
There are inline equations such as $y_i = \alpha + \beta x_i + e_i$.

And displayed formulas:

$$\frac{1}{1+\exp(-x)}$$
```

There are inline equations such as $y_i = \alpha + \beta x_i + e_i$.

And displayed formulas:

$$\frac{1}{1+\exp(-x)}$$

Here you can find information on how to write the symbols you may need.

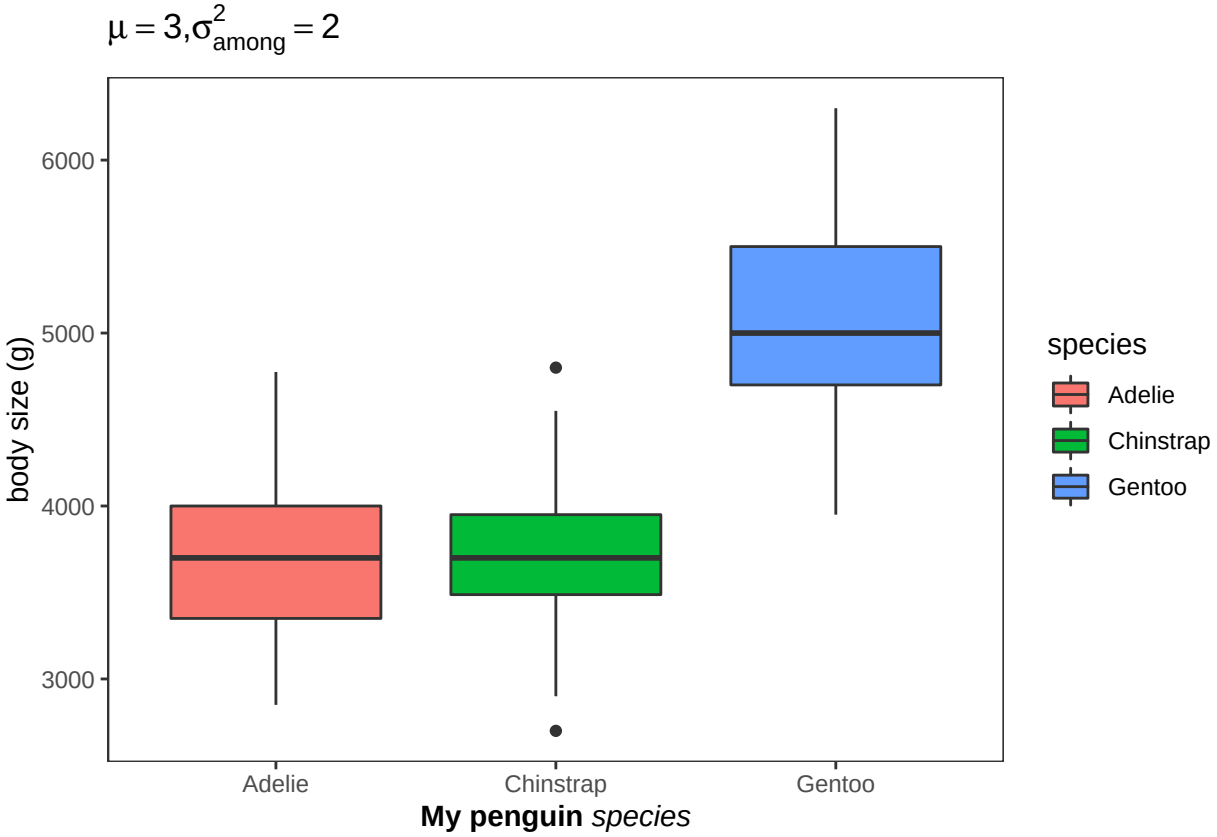
There are packages that can provide additional help for other types of scientific notation. For example the R package *latex2exp* helps to write mathematical expressions in plots, using the function *TeX()*. Using the function *expression()* from the base package we can add **bold** or *italics* to our plots as well.

Here for an example, I made up a title that include symbols:

```
library(ggplot2)
library(latex2exp)

ggplot(penguins, aes(x = species, y = body_mass_g, fill = species)) +
  theme_bw() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) +
  geom_boxplot() +
```

```
  xlab(expression(bold(paste("My penguin ", italic("species"))))) +
  ylab("body size (g)") + ggtitle(TeX("$\\mu = 3,  \\sigma^2_{among} = 2$"))
```



Tables

Tables can be included using the following notation

```
A   | B   | C
--- | --- | ---
1   | Dog | German Sheperd
2   | Cat | Siamese
```

A	B	C
1	Dog	German Sheperd
2	Cat	Siamese

Which is not very practical! Luckily we can directly use dataframes we have in the environment and display them as nice tables using functions from different packages.

For example, we can use **kable()** from the *knitr* package:

```
kable(minipenguins)
```

species	island	flipper_length_mm	body_mass_g
Adelie	Torgersen	181	3750
Adelie	Biscoe	174	3400
Adelie	Dream	178	3250
Gentoo	Biscoe	211	4500
Chinstrap	Dream	192	3500

Or we can use **flextable()** from the *flextable* package, you can find more information here:

```
library(flextable)

flextable(minipenguins)
```

```
## Warning: Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘
## engine is used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning
## by using the ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a
## compatible engine by defining ‘latex_engine: xelatex‘ in the YAML header of the
## R Markdown document.
```

species	island	flipper_length_mm	body_mass_g
Adelie	Torgersen	181	3,750
Adelie	Biscoe	174	3,400
Adelie	Dream	178	3,250
Gentoo	Biscoe	211	4,500
Chinstrap	Dream	192	3,500

For HTML outputs, we can use **kable_styling()** from the *kableExtra* to create an "interactive" table where you can scroll up and down. Not that useful if the table is small:

```
library(kableExtra)

knitr::kable(minipenguins) %>%
    kable_styling(bootstrap_options =
                    c("striped", "hover",
                      "condensed", "responsive"),
                  stripe_color = "blue") %>%
    scroll_box(height = "250px")
```

Tweaking some options we can make it look different:

```
knitr::kable(minipenguins) %>%
    kable_styling(font_size = 7,
                  html_font = "Arial Narrow",
```

species	island	flipper_length_mm	body_mass_g
Adelie	Torgersen	181	3750
Adelie	Biscoe	174	3400
Adelie	Dream	178	3250
Gentoo	Biscoe	211	4500
Chinstrap	Dream	192	3500

species	island	flipper_length_mm	body_mass_g
Adelie	Torgersen	181	3750
Adelie	Biscoe	174	3400
Adelie	Dream	178	3250
Gentoo	Biscoe	211	4500
Chinstrap	Dream	192	3500

```
                  bootstrap_options =
                    c("bordered"))
```

Adding a scroll box (avaiable in HTML) is very useful for a big table:

```
library(kableExtra)

knitr::kable(penguins) %>%
    kable_styling(bootstrap_options =
                    c("striped", "hover",
                      "condensed", "responsive")) %>%
    scroll_box(height = "300px")
```

Please note that for us to be able to knit this document with the HTML specific commands into Word or PDF, we need to add this to the YAML metadata:

```
always_allow_html: true
```

Also note that for the previous example that focused on the HTML output, this table will look aweful in the Word file, and will be incomplete in the PDF file.

Other useful things to add to a document

Hyperlinks

Create simple links by wrapping square brackets around the link text and round brackets around the URL. Example:

```
To find species occurrence data, you can check  [GBIF](http://www.gbif.org/).
```

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007
Adelie	Torgersen	38.9	17.8	181	3625	female	2007
Adelie	Torgersen	39.2	19.6	195	4675	male	2007
Adelie	Torgersen	34.1	18.1	193	3475	NA	2007
Adelie	Torgersen	42.0	20.2	190	4250	NA	2007
Adelie	Torgersen	37.8	17.1	186	3300	NA	2007
Adelie	Torgersen	37.8	17.3	180	3700	NA	2007
Adelie	Torgersen	41.1	17.6	182	3200	female	2007
Adelie	Torgersen	38.6	21.2	191	3800	male	2007
Adelie	Torgersen	34.6	21.1	198	4400	male	2007
Adelie	Torgersen	36.6	17.8	185	3700	female	2007
Adelie	Torgersen	38.7	19.0	195	3450	female	2007
Adelie	Torgersen	42.5	20.7	197	4500	male	2007
Adelie	Torgersen	34.4	18.4	184	3325	female	2007
Adelie	Torgersen	46.0	21.5	194	4200	male	2007
Adelie	Biscoe	37.8	18.3	174	3400	female	2007
Adelie	Biscoe	37.7	18.7	180	3600	male	2007
Adelie	Biscoe	35.9	19.2	189	3800	female	2007
Adelie	Biscoe	38.2	18.1	185	3950	male	2007
Adelie	Biscoe	38.8	17.2	180	3800	male	2007
Adelie	Biscoe	35.3	18.9	187	3800	female	2007
Adelie	Biscoe	40.6	18.6	183	3550	male	2007
Adelie	Biscoe	40.5	17.9	187	3200	female	2007
Adelie	Biscoe	37.9	18.6	172	3150	female	2007
Adelie	Biscoe	40.5	18.9	180	3950	male	2007
Adelie	Dream	39.5	16.7	178	3250	female	2007
Adelie	Dream	37.2	18.1	178	3900	male	2007
Adelie	Dream	39.5	17.8	188	3300	female	2007
Adelie	Dream	40.9	18.9	184	3900	male	2007
Adelie	Dream	36.4	17.0	195	3325	female	2007

To find species occurrence data, you can check GBIF.

Or another example:

```
Penguin artwork provided by @allison_horst. You can get it [here](https://allisonhorst.github.io/palmer
```

Penguin artwork provided by @allison_horst. You can get it here.

Notice that in the sentence above we had to use "\" to avoid the at symbol to be interpreted as a reference. We can also use"\" before an asterisk or lower dash to stop it from considering as in use for bold or italics.

Images

Here are @allison_horst cute penguins. You can put your Figure legend between the square brackets:

```
![your figure legend](pictures/penguins.png)
```

Quotes

With the ">" symbol at the beginning of your sentences, they will appear as a quote. You can also use bold and italics within a quote. For example:

```
> "It is not the strongest or the most intelligent who
> will survive but those who can best manage change".
> _Charles Darwin_
```

> "It is not the strongest or the most intelligent who will survive but those who can best manage change". *Charles Darwin*

References

Burgman, M. A., and Fox, J. C. 2003. Bias in species range estimates from minimum convex polygons: Implications for conservation and options for improved planning. Animal Conservation, 6: 19–28. http://doi.wiley.com/10.1017/S1367943003003044 (Accessed 1 June 2018).

Côté, I. M., and Reynolds, J. D. 2012. Meta-analysis at the intersection of evolutionary ecology and conservation. Evolutionary Ecology, 26: 1237–1252. http://link.springer.com/10.1007/s10682-012-9568-0 (Accessed 25 August 2020).

Fortin, M.-J., Keitt, T. H., Maurer, B. A., Taper, M. L., Kaufman, D. M., and Blackburn, T. M. 2005. Species' geographic ranges and distributional limits: Pattern analysis and statistical issues. Oikos, 108: 7–17. http://doi.wiley.com/10.1111/j.0030-1299.2005.13146.x (Accessed 1 June 2018).

Gurevitch, J., Curtis, P. S., and Jones, M. H. 2001. Meta analysis in ecology. Advances in Ecological Research, 32: 199–247.

Pappalardo, P., Ogle, K., Hamman, E. A., Bence, J. R., Hungate, B. A., and Osenberg, C. W. 2020. Comparing traditional and Bayesian approaches to ecological meta-analysis. Methods in Ecology and Evolution: 2041–210X.13445. https://onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.13445 (Accessed 25 August 2020).