Tips and Tricks about R and Quarto

Minh-Anh Huynh

9/9/24

Table of contents

Preface

This is my homecooked R and Quarto tips and tricks that I found while debugging my stuff.

1 Tips and Trick to write your Thesis in Quarto

Below are some tips and tricks that I needed to use alongside the regular Quarto settings, which were unfortunately not enough to fully format my thesis just like how I wanted. I had to scour the internet and stackoverflow alongside some latex documentation, and I hope this will help someone one day. Or else it serves as a nice little documentation for future usage.

2 Formatting

2.1 Add short captions to List of Tables with pandoc filters and LaTeX

AKA how to add only part of figure caption in list of figures ### Pandoc filter : short-captions

Based on this discussion which leads to the usage of the short-captions pandoc filter.

First add the filter to the yaml header:

```
filters:
  - "pandoc-filters/short-captions.lua"
```

```
![**Bold caption.** Additional caption. End with a citation. [@citation]](figures/figure.p
```

Cross-reference in-line with this syntax : `@fig-label`

2.1.1 captioner package

Based on this thread, or this one from stackoverflow, you can run this LaTeX code to do so :

```
\begin{figure}
\includegraphics{figure.jpg}
\caption[\textbf{Bold caption.}]
{\textbf{Figure or table description. (\cite{Meza-Meza.2020}) or Adapted from \textcite{Au
\label{fig-label}
\end{figure}
```

Note that I return to a new line to make it easier to read.

Write `\listoffigures` to generate the list of figures.

Then you can crossreference the figure using a LaTeX command : `**(Figure \ref{label})**`

I originally used this method because I didn't know about the pandoc filters. I'd have to say the pandoc way is easier to write, and you can have a miniature image in source or visual mode inside R Studio.

One minor hiccup of the latex way is that if you ever switch to the source pane, any bolded cross-reference with latex syntax will be undone. Such as : From `**(Figure \ref{fig:vitd3})**` to `**(Figure** \ref{fig:vitd3})` when going to source and back. This is quite annoying. So I'd rather use pandoc from now.

Update : The cleveref package Section 2.2 solves this problem.

2.1.2 Built-in Quarto attribute using fig-scap

Unknowingly, the figure attribute fig-scap solves the problem.

```
![Long caption](R_logo.png){#fig-label fig-scap="A short caption"}
```

Figure 2.1: Long caption

List of Figures

2.2 Automatically bold cross-reference (Figure, Table #)

Instead of using `(**Figure \ref{figure:stuff}**)`, use `\cref{figure:stuff}` which will automatically put "Figure" or "Table" or something else as appropriate.

To add automatic bold, add to the preambule :

```
header-includes:
- \usepackage[capitalise,noabbrev, nameinlink]{cleveref} # Allows \cref{} to cite latex ta
# Specify which cross-reference should automatically be bolded : Tables and Figures
# Use \cref{} ; For some reason this only works with this exact disposition :
# Only #1, nameinlink and each of the reference specified. namelink + #1#2#3 would give an
- \crefdefaultlabelformat{#2\textbf{#1}#3} # <-- Only #1 in \textbf
- \crefname{table}{\textbf{Table}}{\textbf{Tables}}
- \Crefname{table}{\textbf{Table}}{\textbf{Tables}}
- \crefname{figure}{\textbf{Figure}}{\textbf{Figures}}
- \Crefname{figure}{\textbf{Figure}}{\textbf{Figures}}
```

The parameter `nameinlink` could be removed, but it allows the link to span both the number and the cross-referenced material (Table + #) and not just the number, which I find more practical.

Stackoverflow reference

2.3 Add your table of content to the pdf bookmark

The generated pdf document has a convenient bookmark function for ease of navigation. The bookmark automatically includes pandoc headers, except your table of content.
Add the bookmark package to include headers, and then use the following command:

```
header-includes:
- \usepackage{setspace} # Example of another package used. This syntax will not work with
- \usepackage{bookmark}
```

```
% Add the command just before the toc.
\pdfbookmark[section]{\contentsname}{toc}
% Next command is to rename the table of content
\renewcommand{\contentsname}{Table des matières}
\tableofcontents{}
\newpage
```

2.4 Remove a section from your table of contents in pandoc

You need to combine .unlisted with .unnumbered to achieve this, as stated in Pandoc documentation.

(I have looked inside the Pandoc documentation however and I have no idea where that is stated).

Also, I discovered it was literally marked in the Quarto documentation itself !

```
# Abstract {.unnumbered .unlisted}

# Acknowledgements {.unnumbered .unlisted}

# Chapter 1
```

2.5 Continuous figure numbering

If you want something like Figure 1, Figure 2 instead of Figure 1.1, Figure 1.2, you need to use the following latex command :

```
- \counterwithout{figure}{chapter}
- \counterwithout{figure}{section}
```

This is working for me, using scrreport as the document class (KOMA class).

2.5.1 Explanation :

Changing the numbering of (e.g.) figures involves two modifications:

- Redefining whether or not the figure counter will be reset whenever the chapter/section counter is incremented;
- Redefining the "appearance" of the figure counter (2.1), i.e., removing (or adding) the chapter/section prefix.

`\counterwithout{somecounter}{anothercounter}`

`\counterwithout{somecounter}{anothercounter}` removes the link between *somecounter* and *anothercounter* so that they are independent. For any pair of counters, you can switch between using `\counterwithout` and `\counterwithin`, as the following example shows for the `example` and `section` counters—you can open this example in Overleaf using the link provided below the code.

3 Fix bugs

3.1 Delete the biber cache

If for some unknown reason, you're trying to generate your bibliography and Quarto hits you with

```
generating bibliography
  Couldn't load any math lib(s), not even fallback to Calc.pm at C:\Users\Minh-Anh\AppData\L
etc.
```

Go to the specified file path and delete the folders. What a headache.

Also, `biber --cache` shows you where it is (and if it's bugged, you'll be greeted with this awesome bug error).

4 Bibliography

4.1 Include an organization name in citeproc

Just include double brackets around your {{organization}} and citeproc will format correctly.

```
@article{IOM.2011.org,
year = {2011},
title = {{Dietary Reference Intakes for Calcium and Vitamin D}},
author = {{Institute of Medicine}},
journal = {The National Academies Press},
doi = {10.17226/13050},
pages = {1115},
keywords = {},
}
```

5 ggplot2

5.1 Non Standard Evaluation - Programming with ggplot2

5.1.1 Problem with programming color inside

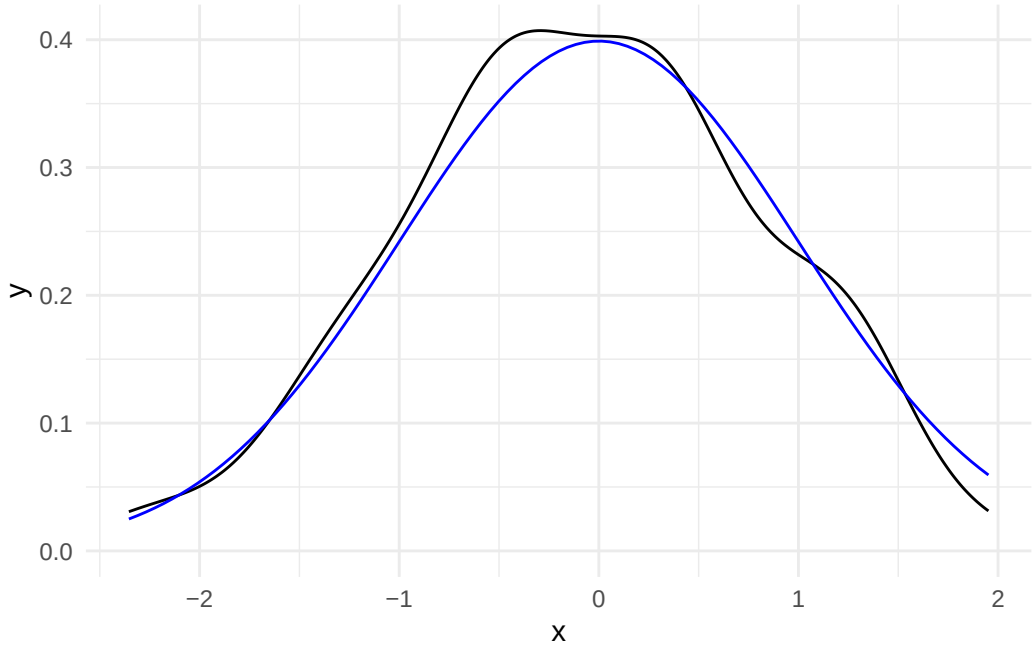
`aes` uses tidy-evaluation

```
librarian::shelf(ggplot2, ragg, quiet = TRUE)
# Sample data
df <- data.frame(
  a = seq(0, 100, by = 10),
  b = seq(100, 200, by = 10)
)

# Your base plot
base_plot <- ggplot(data.frame(x = rnorm(100)), aes(x)) +
  geom_density() +
  theme_minimal()

# Create the plot
plot <- base_plot + geom_function(
  fun = dnorm,
  show.legend = TRUE,
  aes(color = "ATK"),
  colour = "blue"
) + scale_colour_manual(name = "Legend", values = c("Line" = "red"))

# Display the plot
print(plot)
```



Why is the plot not displaying the legend with a red color ?

> ChatGPT: The issue you're encountering with the legend not displaying the color red and the legend being removed when you include the colour = "blue" argument in the geom_function is related to how aesthetics are mapped in ggplot2. When you use aes(color = "line"), you are mapping the color aesthetic to a constant string "line", which means that all the points or lines will have the same color, and that color will be determined by the color scale associated with the "line" category. However, when you include colour = "blue" within the geom_function, you are effectively overriding the color aesthetic that you set with aes(color = "line"). This means that all elements drawn by this specific geom_function will be colored in blue, and ggplot2 will not create a legend because there's no mapping of aesthetics that varies. If you want to specify a different color for this specific geom_function and still have a legend, you can do the following:

```
base_plot +
  geom_function(
    fun = dnorm,
    aes(color = "line"),
    show.legend = TRUE
  ) +
  scale_colour_manual(
```

```
    name = "Legend",
    values = c("line" = "red")
  )
```

More information : Indirectly referring to a variable

5.2 Correct automatic brackets

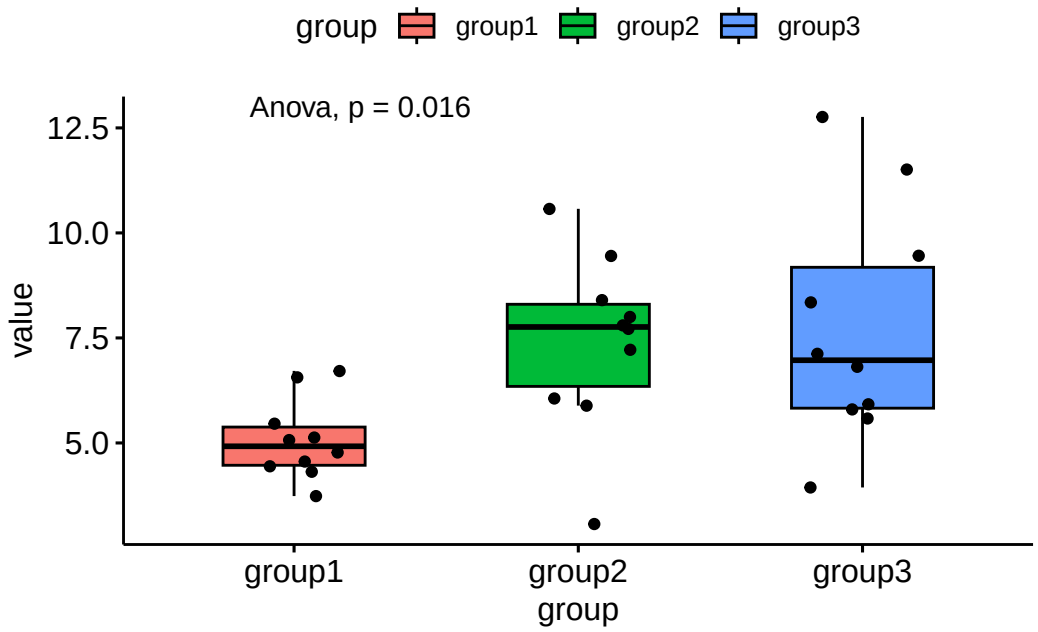
```
# Install ggpubr package
librarian::shelf(ggpubr, tidyr, quiet = TRUE)

# Create some dummy data
set.seed(123)
group1 <- rnorm(10, mean = 5, sd = 1)
group2 <- rnorm(10, mean = 7, sd = 2)
group3 <- rnorm(10, mean = 9, sd = 3)

# Combine the data into a data frame
data <- data.frame(group1, group2, group3) %>%
  pivot_longer(cols = everything(), names_to = "group")
```

```r
# Note that ggpubr works for tidy data, hence using pivot_longer()

# Create the plot
plot <- ggboxplot(data,
  x = "group",
  y = "value",
  width = 0.5,
  fill = "group",
  add = "jitter"
)
plot + stat_compare_means(method = "anova")
```

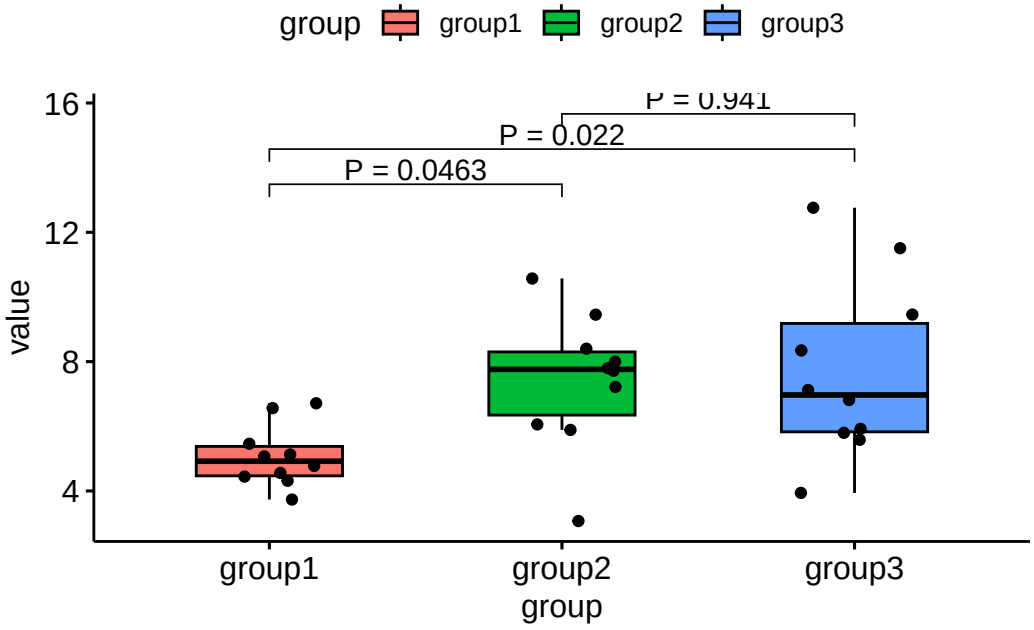


Now let's add some brackets:

```r
# Note that ggpubr seems to also load rstatix
librarian::shelf(rstatix, quiet = TRUE)

# Here is how you can add brackets with P values in your plot:
aov_results <- suppressWarnings(anova_test(value ~ group, data = data))
if (aov_results$p <= 0.05) {
```

```
  tukey_test <- tukey_hsd(data, value ~ group) %>% add_y_position()
  plot + stat_pvalue_manual(tukey_test, label = "P = {p.adj}")
}
```




```
# Note that it is recommended to use an italic *P* in uppercase. I don't think
# this is possible in an R code, so a simple uppercase P should suffice. However
# now the problem is that the automatic p for the anova test is in lowercase.
```

In the datanovia example, you see `add_xy_position()` used, however that can mess up the order of the brackets. Instead, stick with add_y_position, as the x positions can be already determined in some functions. Here both functions work however. Perhaps rstatix got updated ? This was documented in this git issue.

```
tukey_hsd(data, value ~ group) -> test
```

```
# Alternatively with P symbols (not recommended anymore):
    # From ?stat_compare_means()
    symnum.args <-
      list(
        cutpoints = c(0, 0.0001, 0.001, 0.01, 0.05, Inf),
```

```
        symbols = c("****", "***", "**", "*", "ns")
      )
    # Brackets for anova would not work, so you need another test
    my_comparisons <-
      list(
        c("group1", "group2"),
        c("group2", "group3"),
        c("group1", "group3")
      )
    plot + stat_compare_means(
      method = "wilcox.test",
      comparisons = my_comparisons,
      symnum.args = symnum.args
    )
```

```
# Note that the following code doesn't work:
aov_results <- anova_test(value ~ group, data = data) %>%
  tukey_hsd() %>%
  add_xy_position()
# Instead, don't start from anova and use the test directly:
tukey_test <- tukey_hsd(data, value ~ group) %>% add_y_position()
```

References