Workshop 3 Instructions: Rmd reports

2024-01-23

Introduction

Welcome!

In this week's exercise you will apply your R knowledge from the three prework lessons for this week:

Refer to the following lesson notes during the workshop:

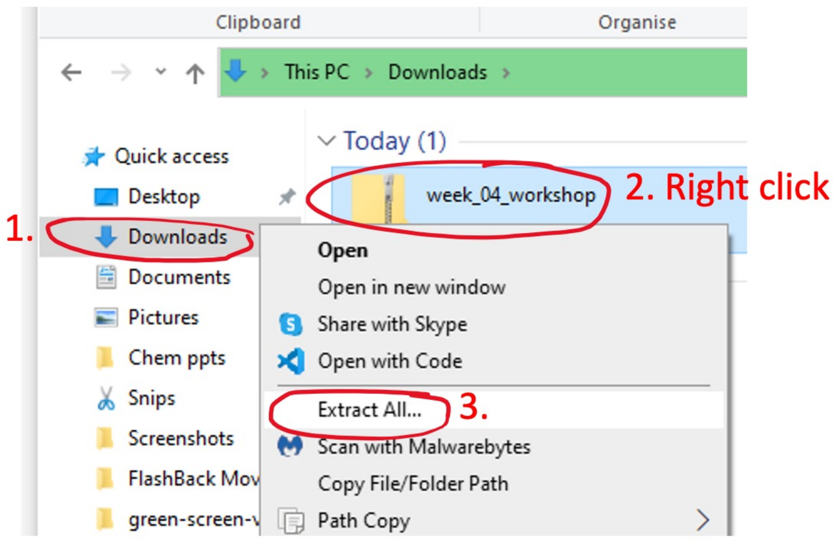
- Data Structures
- R Markdown

The due date for the assignment is Friday, January 26th at 23:59 PM UTC.

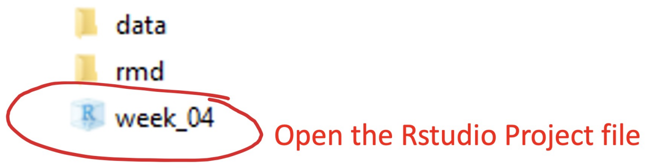
Downloading and Opening the Assignment

1. **Locate the email** we sent you with the workshop folder attached. The subject line is "RBB January 2024: Workshop 3 Materials".
2. **Download the attached week_03_workshop.zip folder -**

   Alternatively: you can download the folder by clicking "Download workshop code" button on the assignment page from the WORKSHOP 3 topic of the course page.
3. **Unzip the week_03_workshop.zip folder**. If you are on a PC, go into your Downloads folder, right click on the zip file downloaded, and click on "**Extract all**".



4. **IMPORTANT**: click on the RStudio Project file called "week_03_workshop.Rproj" in the unzipped folder to open the project in RStudio.



Steps to complete the assignment

Part A: Practice troubleshooting R Markdown

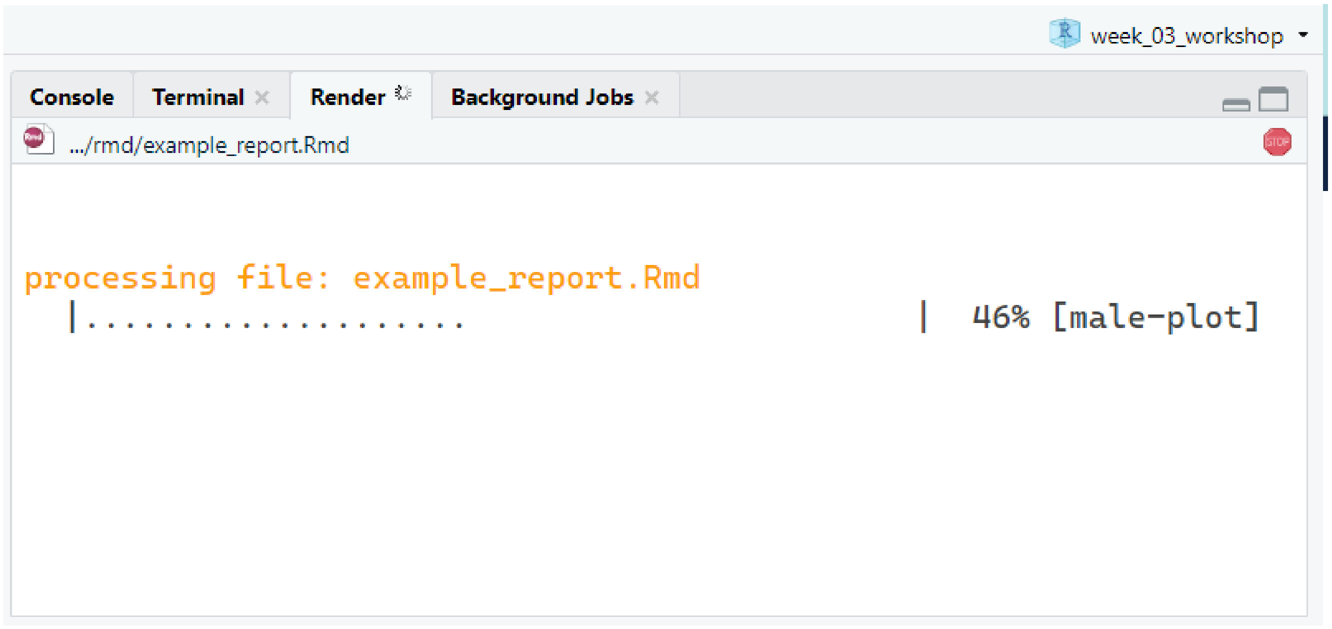
- In the "rmd" sub-folder, there is an example HTML report called "colombia_example_report.html". Open and quickly examine the report.
- This report was created with a source Rmd file, but we have changed the Rmd to introduce 6 small mistakes that prevent the Rmd from successfully knitting to produced the desired HTML.
- Your mission is to find and fix these errors, so that you can produce the correct output!

Work with your team to explore the Example Report to find and correct mistakes!

Steps to complete Part A:

1) Open and try to **knit the example report Rmd** to HTML.

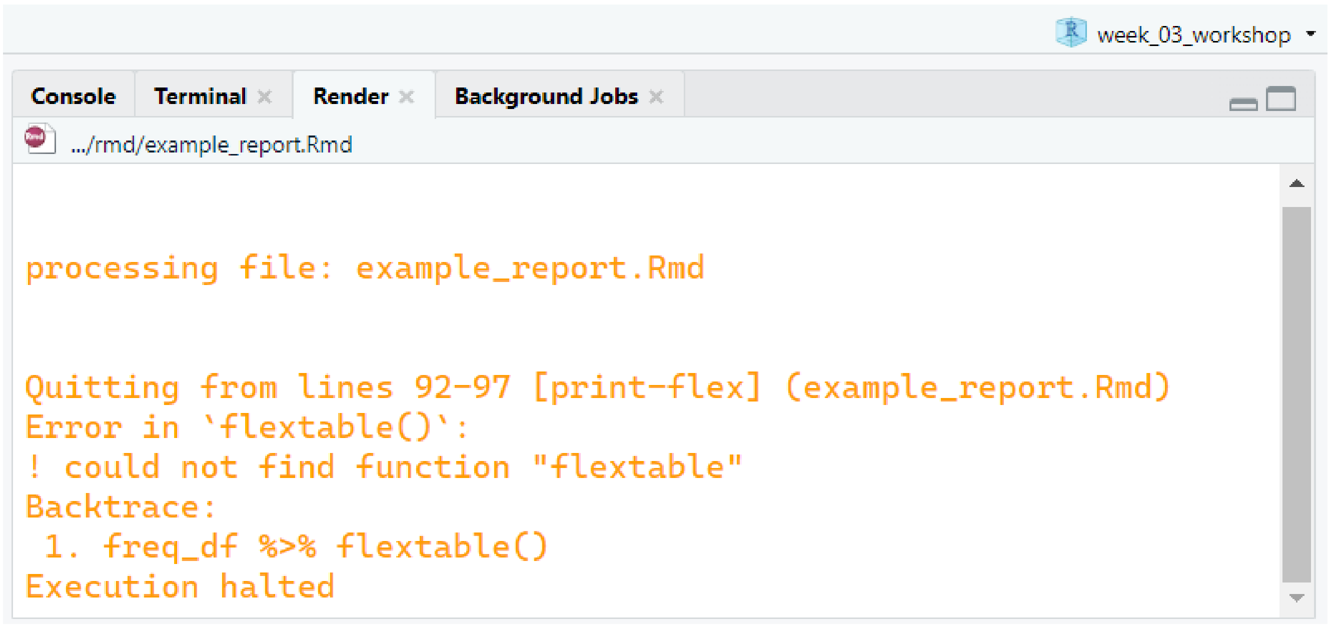
- Problem 1: it will open esquisse and get stuck there. The HTML won't finish knitting. look at your "Render" pane - it will probably look like this. It's stuck on the code chunk that is generating the plot for male victims. To quit, press the red stop button at the top right of the Render pane.



```
processing file: example_report.Rmd
  |.....................                           |  46% [male-plot]
```

Now identify and fix the part of the code that caused this issue. Then, try knitting again.

- Problem 2: The document is no longer stuck on esquisse like last time, but still won't knit! However, you should get a helpful error message like this:



```
processing file: example_report.Rmd


Quitting from lines 92-97 [print-flex] (example_report.Rmd)
Error in `flextable()`:
! could not find function "flextable"
Backtrace:
 1. freq_df %>% flextable()
Execution halted
```

Figure 1: Quitting from lines 92-97.

Find the source of this problem and fix the code. Then, try knitting again. This time, you should be able to get a knitted HTML. Yay!

- Problem 3: Code output. As you scroll through your HTML report, you will notice that there are two code outputs in the document, which is not what we want. Edit the code chunk options in your Rmd to hide the unncessary output. Then, knit again!

- Problem 4: Missing plot. Note that the age group distribution for female victims is missing in the HTML report. Fix the issue in the code, and then knit once more.
- Problem 5: Incorrect display of inline code. The the section on "Age group distribution per group", there are two instances of inline code. One of them renders correctly in the final document, and one of them does not. Fix the issue in the code, and then knit again.
- Problem 6: Finally, we have one last issue. We want the title of the report "Road Injuries of Motorcyclists in Colombia" to be a Level 1 heading. However, it is showing up as a bullet point. Fix the syntax in the Rmd to make it a Level 1 heading.

Final knit: Now all the issues should be resolved! Knit the example report Rmd, and you should now get an HTML that has no more errors. Congrats! Now it's time for Part B.

Part B: Creating your own report

Your task will be to create a short R Markdown-based report comparing the distribution of ANY **TWO CATEGORICAL variables** in the India TB pathways dataset. You could compare for example, education level and smoking habit, or employment and visit location, or alcohol consumption and diabetes—any two variables of your choice

The report must contain these three things:

a. A plot created with {ggplot2}/{esquisse}
b. A table created with {flextable} or another table package in R.
c. At least one use of *inline R code* within the Rmd.

Steps for Part B:

2) **Create a new Rmd** file (not R script!) in the "rmd" folder and give it an appropriate name (e.g., tb_report_YOUR_NAME.Rmd).
3) **Copy the YAML** header from the example report into your new file.
4) Make a new code chunk for package loading and **load your needed packages**. (Refer to the example report for clues about what packages you might need.) Make sure to use the correct chunk options!
5) Under a new **section heading**, add a short description of the India TB dataset, similar to the one about Colombia motor accidents in the example report. Info about the dataset can be found in the "data" folder.
    - Remember that *section headings* will appear in large text of your final report, so avoid using names like "Data cleaning". Keep these notes hidden in code chunks instead.
6) Make a **new code chunk** in which you import the India TB dataset. Give the data frame a short, descriptive title. Remember to use the `here()` function when loading the CSV, otherwise, the data loading might not work.
    - Tip: You can use Ctrl+Alt+I to insert a new code chunk in your rmd.
7) Once your dataset is loaded, inspect the dataframe and **choose two categorical variables for your analysis**. Once you and your group have chosen your desired variables, you can proceed with creating the four requirements for the assignment.

8) **Explore data in esquisse**. Remember that variable names in esquisse are color-coded by data type. Categorical variables are *orange*.

   - Tip: A good way to visualize two categorical variables is with a *stacked bar chart*. You can create on in esquisse by adding one variable to "X" or "Y" and the second variable to "Fill".

9) Another way to compare the distribution of two categorical variables is with a table. You can **make a frequency table** using the `tabyl()` function, however this creates a data frame that is not very presentable in a report.

10) Create a **better-looking table** using {flextable} or {gt}. You should summarise the SAME two variables that you chose for plotting above. Remember to use the appropriate chunk options so only the nice table is printed (see example report for hints).

11) You can highlight specific numbers from the data using **inline code**. See example report Rmd for hints!

12) The final and most important part - **knit your report**! Before you knit, make sure you delete or comment out any instances of `esquisser()` or `View()`. This is because functions that open new windows will interrupt the knitting process. You may have to try several times before successfully knitting your report. Be patient and ask instructors for help if you get stuck.

To submit your work, please upload JUST YOUR Rmd to the submission page. But make sure the Rmd can be knit successfully before uploading.

Bonus Challenge (Optional)

Finished early? Try one or both of the challenge exercises below.

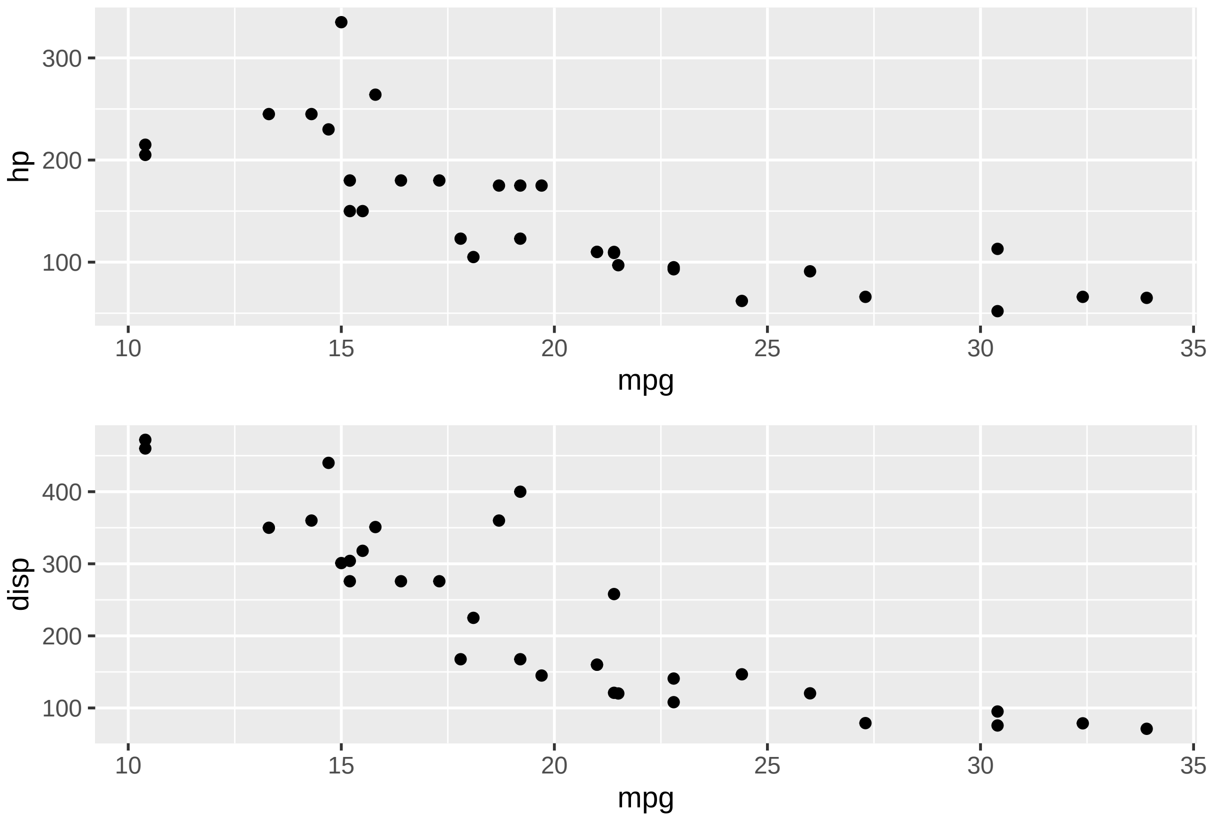
Challenge 1: Using patchwork to combine images

GOAL: Use `{patchwork}` operations in your report Rmd and stitch together two plots!

The `{patchwork}` package allows you to combine multiple plots into a single plot, which can make it easier to compare plots and keep your data visualization tidy and organized.

Let's look at a simple example. Assume we have two plots called `plot1` and `plot2`:

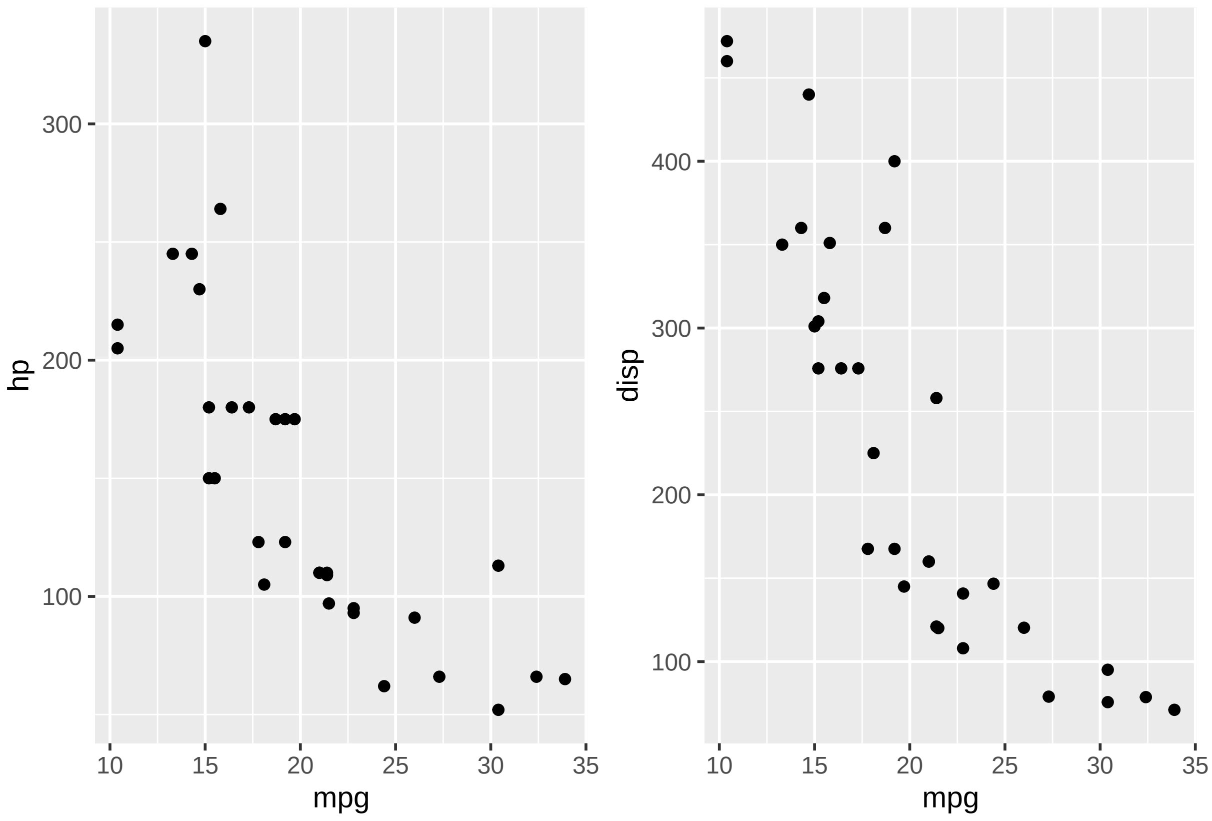
```r
# load the required packages
pacman::p_load(tidyverse, patchwork)

# Create plot1
plot1 <- ggplot(mtcars, aes(x = mpg, y = hp)) +
         geom_point()

# Create plot2
plot2 <- ggplot(mtcars, aes(x = mpg, y = disp)) +
         geom_point()

# Combine plots vertically with patchwork
combined_plot <- plot1 / plot2
combined_plot
```



In this example, the `/` operator is used to arrange the plots vertically, with `plot1` on top and `plot2` at the bottom.

If you wanted the plots to be arranged horizontally, you would use the `+` operator instead:

```r
# Combine plots vertically with patchwork
combined_plot_2 <- plot1 + plot2
combined_plot_2
```



Challenge 2: Use square brackets in inline code

GOAL: Extract numbers from a data frame using square bracket notation, and display it in your knitted report with inline code.

In R markdown, we use specific syntax to include inline R code, which is R code that runs within the text of your document. To use inline R code, we simply include the R code within backticks and precede it with an `r` like so

```
We can found that there were `r your_code_here` cases in Category B.
```

One common use of inline R code is to access and display specific elements from a data object such as a matrix or dataframe. To access a specific element, we use square brackets `[]` with the row and column indices. For instance, if we have a matrix `m`, we can access the element at row 2, column 3 as follows: `m[2, 3]`.

Let's take an example with a Markdown table. Suppose we have a table as follows:

Category	Count
A	10
B	20
C	30

If we want to display the count for Category B in our text, we could write something like this in our Rmarkdown script (not in a code chunk):

```
We found that there were `r counts[2, 2]` cases in Category B.
```

When this R Markdown document is knitted, it will display as:

“We found that there were 20 cases in Category B.”