# Workshop 3 Instructions: Rmd reports

2024-04-23

## Introduction

Welcome!

In this week's exercise you will apply your R knowledge from the two pre-work lessons for this week (linked below):

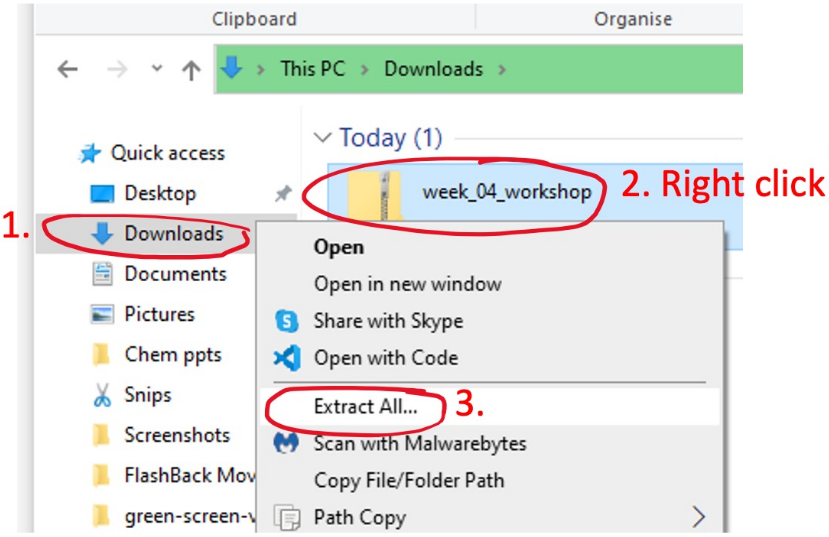
- **Data Structures**
- **R Markdown**

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

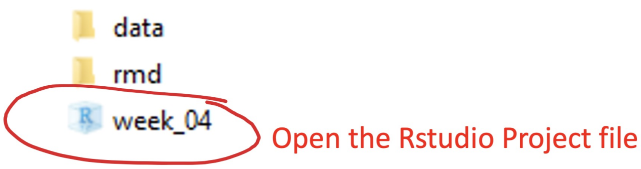
---

## Downloading and Opening the Assignment

1. **Access the WORKSHOP 3 page** from the course page.
2. **Download the attached week_03_workshop.zip folder** by clicking the "*Download workshop code*" button on the assignment 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.



## Read BEFORE Completing the Assignment

Today's workshop will be broken down into **TWO** parts (Part A and Part B) with two optional bonus questions.

- **PART A**
  - In Part A, you will be creating an R Markdown report from scratch based on the **India TB pathways dataset**. For more information about this data, please review the ***"info_about_the_datasets.pdf"*** document located in the ***"data"*** sub-folder.
  - Read the detailed step-by-step instructions below carefully!!! As a reference document, we recommend you to use the ***"example_report_exercise.Rmd"*** located in the ***"rmd"*** sub-folder to help you in the process of creating your own RMD report.
  - **** Note: The example_report_exercise.Rmd* ***CONTAINS MISTAKES*** *that will be addressed in PART B of today's workshop. When using this document for reference in Part A, make sure to keep this in mind!!!*
- **PART B**
  - In Part B, you will practice troubleshooting some common R Markdown errors. The aim in this section is to reproduce the ***"colombia_example_report.html"*** report located in the ***"rmd"*** sub-folder. For this part, you should correct the code issues directly in the ***"example_report_exercise.Rmd"*** file located in the ***"rmd"*** sub-folder.

# Directions

## Part A : 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. For example, you could compare education level and smoking habit, employment and visit location, or alcohol consumption and diabetes – any two variables of your choice.

**The report must contain these three things:**

a. One plot created with {ggplot2}/{esquisse}
b. One table created with {flextable} or another table package in R (e.g., {gt}, {reactable}).
c. At least one use of *inline R code* within the Rmd.

**Steps for Part A**

1) **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).

2) **Copy the YAML** header from the example report (*"example_report_exercise.Rmd"*) into your new Rmd file.

3) 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.)

   - Tip: Make sure to use the correct chunk options!

4) 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. Information about the dataset can be found in the *"data"* folder.

5) Make a **new code chunk** in which you import the India TB dataset. Give the data frame object 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.

6) 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.

   - Tip: Inspect your dataframe using functions taught in our previous lessons such as `head()`, `summary()` and `glimpse()` .

7) **Create an informative graph using 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 one in esquisse by adding one variable to "X" or "Y" and the second variable to "Fill".

8) **Create a table** using {flextable}, {gt} or {reactable} to compare the distribution of 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).*

   - Side-note: You can **make a frequency table** using the `tabyl()` function, however this creates a data frame that is not very presentable in a report; therefore, we recommend you to use the other previously mentioned packages.

9) **Use inline code** to highlight specific numbers from the data. *Remember that you can do so by embedding code chunks in the text.*

   - Hint: You can refer to the *"example_report_exercise.Rmd"* file or the R Markdown lesson for hints!

10) The final and most important part - **knit your report**!

    - Tip: 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.

---

## Part B: Practice troubleshooting R Markdown

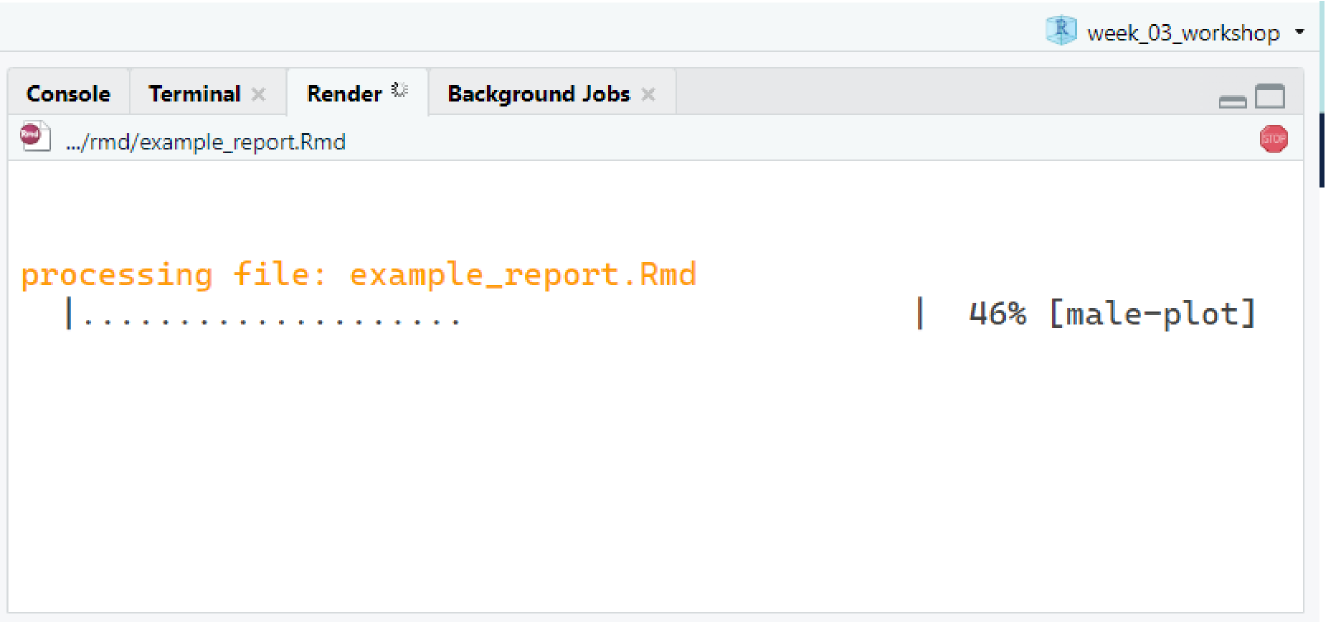
- In the “rmd” sub-folder, there is an example HTML report called *“colombia_example_report.html”*. Open and quickly examine the report.
- This report was created with a source Rmd file, but we have changed the Rmd to introduce 6 small mistakes that prevent the Rmd from successfully knitting to produced 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 for Part B

1) Open the *“example_report_exercise.Rmd”* located in the sub-folder *“rmd”* and try to **knit the file into an HTML.**

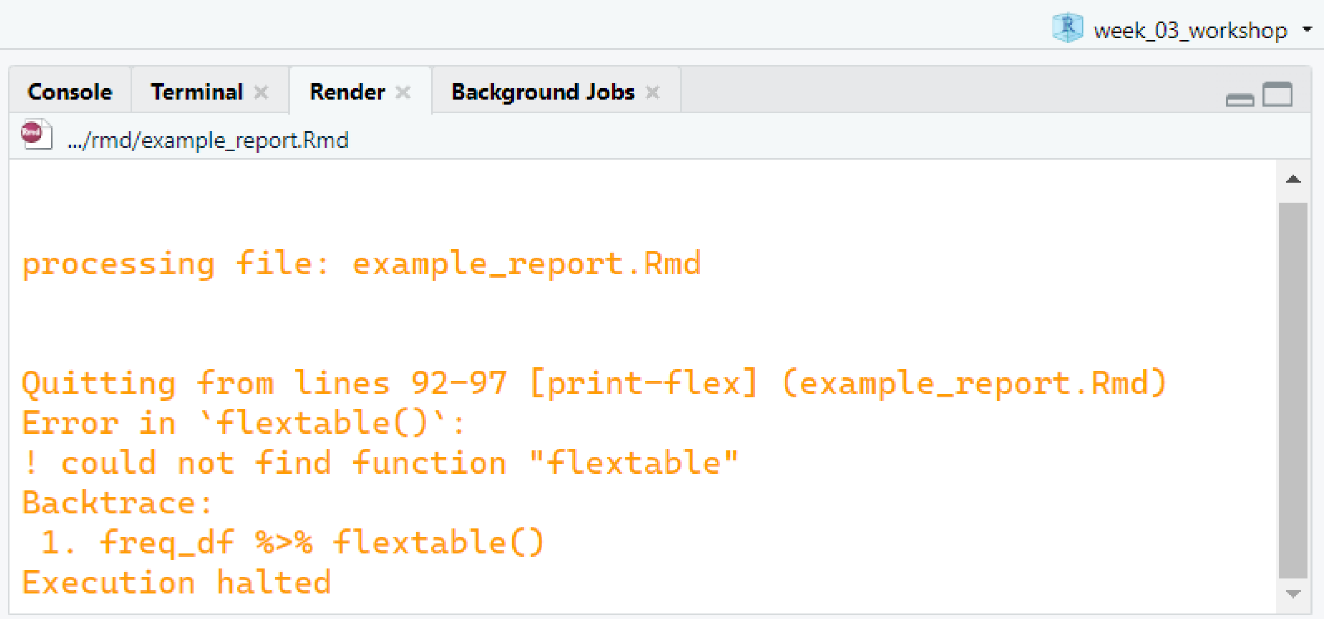
   ***Problem 1 : Render Stuck***. The HTML won’t finish knitting because Rstudio will open **esquisse** and get stuck there. If you look at your *“Render”* pane it should indicate that it is stuck on the code chunk named *male-plot*.



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

2) **Stop knitting** your Rmd file and **fix the part of the code** that caused this issue.
   - Tip: To stop knitting your Rmd file, press the red stop button at the top right of the Render pane, as seen in the above image.
   - Hint: Locate the male-plot code chunk by reviewing the name of each code chunk.
3) **Try knitting** your Rmd again.

   ***Problem 2: Error message***. The document is no longer stuck on esquisse like last time, but still won’t knit! However, you should get a helpful error message.
4) **Find the source of this error message** and **fix the code**.
   - Hint: Read the error message carefully! It should indicate **where** the file is stopping and **what** is missing in your Rmd file.



```
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.

5) **Try knitting again**. This time around, you should be able to get a knitted HTML. *Yay!*

   ***Problem 3: Code output***. When creating reports, we only want to include relevant and important text and graphs and *hide* unnecessary code chunks. 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!

6) **Edit the two code chunk options** in your Rmd to **hide the unnecessary code output**.

   - Hint: Consider adding code chunk options to appropriately hide the code outputs.

7) **Try knitting once more!**

   ***Problem 4: Missing plot***. Note that the plot illustrating the age group distribution for female victims is missing in the HTML report.

8) **Fix the issue** in the code by making sure the plot shows up in the knitted HTML.

9) Try to **knit again**!

   ***Problem 5: Incorrect display of inline code***. The section on *"Age group distribution per group"* contains two instances of inline code. One of them renders correctly in the final document, and one of them does not.

10) **Fix the issue** in the code by correcting the syntax of one of the inline codes.

11) Try to **knit your Rmd file one more time**!

    ***Problem 6: Title syntax.*** 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.

12) **Fix the syntax** in the Rmd to make the title a Level 1 heading.

13) **Knit your Rmd file one last time**! All the issues should have been resolved and you should have an HTML document similar to the *"colombia_example_report.html"*.

# Submitting Workshop 3 Exercise

For this workshop, you will **ONLY** submit the *Rmd file* you created in **Part A** to report on two categorical variables from the India TB dataset.

To submit your work, please upload your own **Rmd file** to the submission page. *Only submit the Part A RMD FILE!*

*Note: Before uploading and submitting your file, make sure the Rmd knits successfully!*

---

# 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."