

7. Integrating it all together: Paired exercise

Recall that our goal is to generate a report which analyses how environmental conditions change microbial communities in Lake Ontario.

Discussion

How do you usually share data analyses with your collaborators? Add your usual workflow to the Etherpad.

Missing a period, if you care about that

Starting the report

[Back to top](#)

Let's return to the new R Markdown file you created and delete everything below the setup code chunk. (That stuff is just examples and reminders of how to use R Markdown.)

Next, let's save our R markdown file to the `reports` directory. You can do this by clicking the save icon in the top left or using `control` + `S` (`command` + `S` on a Mac).

There's one other thing that we need to do before we get started with our report. To render our documents into html format, we can "knit" them in R Studio. Usually, R Markdown renders documents from the directory where the document is saved (the location of the `.Rmd` file), but we want it to render from the main project directory where our `.Rproj` file is. This is because that's where all of our relative paths are from and it's good practice to have all of your relative paths from the main project directory. To change this default, click on the down arrow next to the "Knit" button at the top left of R Studio, go to "Knit Directory" and click "Project Directory". Now it will assume all of your relative paths for reading and writing files are from the `ontario-report` directory, rather than the `reports` directory.

Now that we have that set up, let's start on the report!

Might it be helpful to say the file should be named something ending in `.Rmd`?

Before we finalize our report, let's look at a few other cool features. Sometimes, you want to describe your data or results (like our plot) to the audience in text but the data and results may still change as you work things out. R Markdown offers an easy way to do this dynamically, so that the text updates as your data or results change. Here is how to do this.

First, let's create a code chunk that summarizes features of our data that we can use to describe our plot to our audience. Note that we set `include=FALSE` because we only want this step to happen in the background. For our purposes, we will calculate how many countries were included in the analysis, as well as the minimum and maximum GDP per capita values:

Output

```
```{r data_summary, include=FALSE}
nSamples <- sample_and_taxon %>%
 select(sample_id) %>%
 n_distinct()
```

## Countries and GDP from old lesson

Here's the text that we need to include to create a summary table of our data:

### Output

```

|Summary of Data|
|-----|-----|
|Number of Samples|`r nSamples`|
|Minimum Temperature|`r minTemp`|
|Maximum Temperature|`r maxTemp`|

```

This will render like this:

I needed to run this without the three backticks at the beginning and the end for it to match the rendered table and not just be a bunch of dashes and pipes.

This is useful if we are reporting a few values, but can get tedious for larger tables. Another way we can add tables to our reports is using an R function called `kable()`. Since this is an R function, we will use it within a code chunk. We can give the `kable()` function a data table and it will format it to a nice looking table in the report. For example, we could use the following code to generate a table of all the Deep samples. [The rendered version should look almost exactly as it does on this webpage.](#)

```
R
Load library
library(knitr)

print kable
sample_and_taxon %>%
 filter(env_group == "Deep") %>%
 select(sample_id, env_group, cells_per_ml, temperature) %>%
 kable()
```

#### Output

sample_id	env_group	cells_per_ml	temperature
May_12_B	Deep	2058864	4.07380
May_29_B	Deep	2153086	4.66955
May_33_B	Deep	2293177	3.87050
May_41_B	Deep	2422141	3.76370
May_55_B	Deep	1847686	3.66830

1. “The rendered version should look almost exactly as it does on this webpage.” What do you mean? Where is it on this webpage?
2. Why are we going back to the `##` for denoting descriptions? Is that something you can do in Rmd? Would it make more sense to write it like what I put below so that it is consistent with what we were doing before? (Except I meant “echo” instead of “include.”)

```
54 {r print kable, include=FALSE}
55 sample_and_taxon %>%
56 filter(env_group == "Deep") %>%
57 select(sample_id, env_group, cells_per_ml, temperature) %>%
58 kable()
59 }
```

#### ★ Using the “Visual” view for easy formatting

In newer versions of RStudio, we can switch to the “Visual” view when editing our documents. This makes the experience much more similar to writing in software like Microsoft Word or Google Docs. We can use formatting tools (like bolding and italicizing), insert pictures, and create tables without manually typing out the markdown syntax. The best part? If you then switch back to the “Source” view, you can see the markdown syntax RStudio has automatically created for you.

Missing punctuation at the end