CHESTER ISMAY

LEARNING R IN CHEMISTRY AT REED COLLEGE

Contents

Introduction

What is this?

In the HTML version of this book, you can also download the PDF version of the book by clicking on PDF button in the top toolbar of the page. HTML is the preferred format but the PDF format may be preferred for some readers. Links to the different GIFs directly found in the HTML version are provided in the PDF version.

This resource is designed to assist students in CHEM 101/102 in using RStudio and R Markdown to complete their labs. (A more general reference to the specifics of R, RStudio and R Markdown is available in a different free book here). This resource will show you GIFs explaining how to do some common procedures you'll need to do to complete labs. In addition, it will provide more details about the specifics of the different lab templates available in the `chemistr` R package. Each chapter of this book will correspond to each of the different labs.

Lastly, if you are interested, you'll find instructions on how to complete the labs using RStudio Desktop instead of the RStudio Server in the Appendix. The Appendix will also include descriptions of the code that is sitting "behind the scenes" in many of the functions created in the `chemistr` package: `chem_table`, `chem_scatter`, `chem_clustered.bar`, etc. These functions were written to ease a beginner into working with R. Those interested in customizing their lab reports further should look over the R code in the Appendix. Additionally, many of the needed packages are automatically loaded when the `chemistr` package is loaded using `library(chemistr)`. You'll see how each of these packages is used in the Appendix.

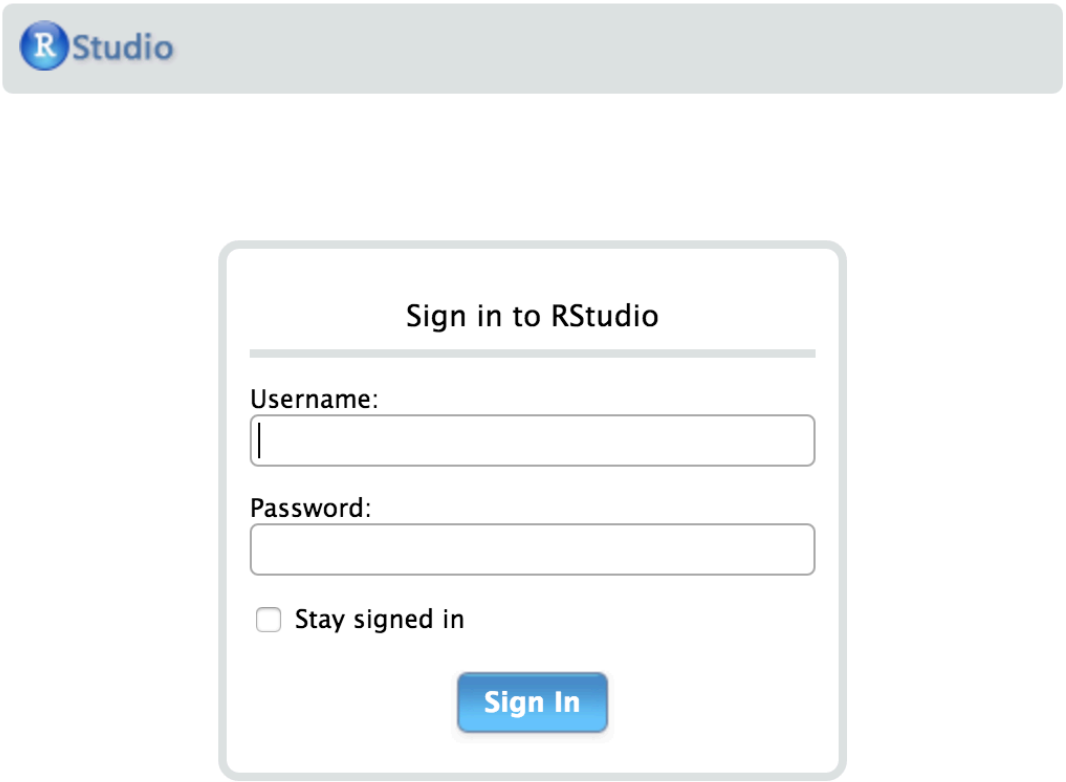
If further clarification is needed on any other aspect of the book, please email me with a reference to the error/area where more guidance is necessary. More advanced users are encouraged to create a GitHub issue here. Pull requests on GitHub for typos or improvements are also welcome and you can easily do so by clicking on the Edit button near Search at the top of the HTML version of the book.

This book will evolve and be updated as needed based on feedback. You can always check the date at the end of the chapter to see when the book was last updated. It is recommended that you use Google Chrome as your browser since GIFs sometimes do not load automatically with other browsers.

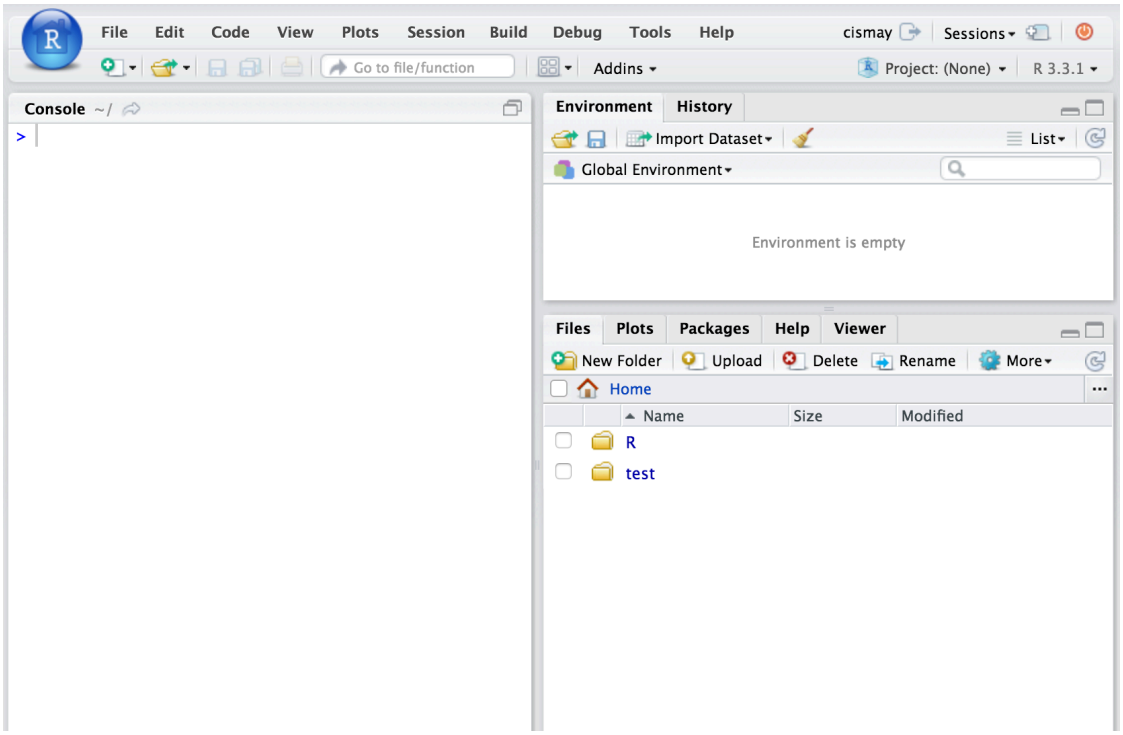
Starting the labs

Logging in and initial screen

The RStudio Server provides a web-based way to run analyses in R. This means that you will only need an internet connection and a web browser to run your analyses. You can be running a Windows machine, a Mac, a Linux, or pretty much any other device that has access to the internet and a web browser. If you are interested in using the Reed College RStudio Server off campus, you'll need to request access in the form here. The initial login page will not load for you unless you are connected to the VPN from off campus locations. After entering the link (`http://rstudio-dev.reed.edu`) into your browser, you'll see a page that looks something like:



After logging in with your Reed College username (mine is `cismay`, for example) and password, you should see a layout similar to what follows.



Creating and Sharing Projects using RStudio Server

A good habit to get into whenever you start a new project following a common theme (such as your CHEM 101 labs) with R code is to create a new RStudio project to go along with it. RStudio project files have the extension `.Rproj` and store metadata that goes along with the documents you've saved and information about the R environment you are working in. More information about RStudio projects is available from RStudio, Inc. here.

You'll be sharing your Chem 101 labs with your Danielle Cass (`dcass`) and Chester Ismay (`cismay`) using the RStudio Server. This will allow both of us to help you with errors you may be having in writing your lab report without having to email files back and forth. You'll just need to email us with the name of the lab you are having troubles with and what you named your RStudio project and we'll be able to go in and look over your R Markdown file. You can think of this as working somewhat like Google Docs in that both Danielle and I can provide feedback and walk-through your analyses.

To share the project with us, you should follow along with the example below. Instead of naming your RStudio project `CHEM101_ChesterIsmay` as the GIF shows, you should name it `CHEM101_YourName` where `YourName` is replaced with your first and last names. Spaces can cause problems in file names so it is recommended that you avoid them. If you've already shared your project and you have spaces in the name, you'll probably be OK, but try to avoid spaces in your file names going forward. Calling your project `CHEM101_YourName` will allow Danielle and I to track down your project more easily than if everyone named their project `initial` or something similar.

`https://raw.githubusercontent.com/ismayc/chemistr-book/master/gifs/proj_share.gif`

You'll need to also add `cismay` as someone you'd like to share the project with. You can do so by entering my username immediately after you have clicked **Add** to add Danielle or you can go back to **Share Project** and enter my name there just as you have with `dcass`. If you see that your project is shared with both `dcass` and `cismay`, you have completed this stage successfully.

You'll see at the beginning of each chapter corresponding to the labs how you can access the different R Markdown template files for each lab. These template files are built into the `chemistr` package that I have already loaded on the Reed RStudio Server for you.[^If you are completing the labs using RStudio Desktop, you'll see instructions on how to install this package onto your computer in the Appendix. After installing the package, you'll be able to use the R Markdown templates in much the same way students using the RStudio Server can. Note that if you are using the RStudio Desktop, you won't have access to the Shared Projects feature and it will be more difficult for Danielle and I to help you.]

Requesting help

It's nearly impossible for Danielle and I to help you unless we know that you are having troubles. If you find yourself unable to complete a lab because of problems with RStudio and you've spent more than 30 minutes trying to figure it out, it is time for you to stop and ask for help. We encourage you to try to figure it out on your own as much as possible because this helps you learn, but if you are deeply entrenched in not getting R to work, that could be taking away from your learning of Chem 101/102 material.

We encourage you to reach out to both of us and carefully explain what is going wrong with your R Markdown file. Sending us an email saying "I can't get it to work, please help." is not insightful and won't help you learn in the long-run. Be as descriptive as possible. Something like what follows is much preferred:

> "I'm receiving an error on line 46 of my Light lab about not finding an object. I've tried Googling it and found a couple references about it including **link1** and **link2** but it's not working for me."

This tells us that you have tried to figure it out for yourself, but that you require our assistance. That is to be expected and you shouldn't be discouraged if you don't understand something about R Markdown immediately.

We are here to help and please don't give up on this. Learning how to work with R even at the most basic of levels will help you long-term in keeping track of analyses and getting you on the path to reproducible research. This will help you in your academic careers and whatever you do after your academic career ends.

If you need assistance, please email Danielle and Chester with your concise error in which lab and what the name of your RStudio project is. We are happy to help you learn how to use R!

Book was last updated:

```
## [1] "By Chester on Sunday, September 04, 2016 13:32:00 PDT"
```

1

Lab 1: Light

1.1 Getting the lab template

In RStudio, you'll find the R Markdown template that you will update with your own results by following along with the GIF below:

`https://raw.githubusercontent.com/ismayc/chemistr-book/master/gifs/light_template.gif`

I've also created a new folder called `Lab1-Light` in my `CHEM101_ChesterIsmay` folder and named the resulting R Markdown file that was loaded in from the R Markdown template in `chemistr` the same thing: `Lab1-Light.Rmd`. You are highly encouraged to follow this same convention for all of your labs. (Note that as I mentioned earlier in this manual I do not have any spaces in the folder names or in the names of files here.) This will allow you to remember exactly where you stored the files and allow Danielle and I to easily find your files on the RStudio Server. Note that these folders were also created in the `CHEM101_ChesterIsmay` folder where the `Rproj` file resides. If you put your lab in the `Home` directory instead, Danielle and I won't be able to access it since `Home` is the parent directory of `CHEM101_YourName`.

1.2 The parts of the lab template

1.2.1 The YAML header

I've already entered some values into my R Markdown file. First I gave a title to the R Markdown file: "Lab 1 - Light". I then entered my name and the current date. All of these are in quotation marks since they all correspond to character strings and not numbers. It will likely still work if you don't put quotation marks around the entries here, but it is recommended that you use the quotation marks to help you understand the use of quotation marks when you begin working with R code.

This portion of the R Markdown file is called the YAML header. It stores metadata about your document. It is separated by `---` before and `---` after. RStudio will warn you by coloring

the `---` red if you accidentally do this.

The last line here corresponds to the type of output that will be produced. In this case, I've created a nice way for you to output this R Markdown file to a DOCX (Word document) format. The `chem_lab_word` document format in `chemistr` creates a Word document with figure captions enabled by default and images with size 3 inches in height and 6 inches in width by default.

1.2.2 Initial R Chunk

On line 8 of the Light R Markdown template, you'll see three backticks starting the line. This tells R Markdown to expect a block/chunk of code. This code is R code by the next designation `{r`. Next we give this R chunk a name `setup` and then specify chunk options immediately after the name and a comma. Here we have specified `include=FALSE`. This chunk option will run the R code when we press **Knit** button near the top of the code editor will not put the R code or its output into the document. We'll press the **Knit** button in a bit to understand what it accomplishes.

You'll see this `setup` chunk in all of the lab templates. It's good practice to load any package you'll need for your analysis at the top in a chunk like this. You may not initially understand why `chemistr` is needed but it will be needed when we create a table using the `chem_table` function, which resides in the `chemistr` package, in the `table` chunk beginning on line 21.

1.2.3 Headers

Immediately below this first R chunk is two hash tags. This corresponds to creating a header in our resulting document. The font immediately following the hash-tags will be larger and act as a divider in our document. The number of hash-tags you enter corresponds to how large you'd like to make the header. One hash tag is the largest and six is the smallest. It is recommended that you use between one and three hash-tags for headers.

1.2.4 Enter data chunk

If you press the **Knit** button at this moment, you'll receive an error. Let's look into what this error means:

`https://raw.githubusercontent.com/ismayc/chemistr-book/master/gifs/knit_error1.gif`

Danielle has created lab templates that might not necessarily build without errors immediately for you. You should carefully read over the directions in each chunk (The comments after the `#` in the R chunks.) before pressing **Knit**. Here we see that the error occurs near line 15 of our file.

The `Color` and `WavelengthRange` vectors need to have some values entered. These results will correspond to what you observed in lab. I will enter some demo results in here and also show you how to use the R Console to check your results. You'll see that the `Color` and `WavelengthRange` vectors must be of the same length when we create a `data_frame` to combine them together into "spreadsheet" format in the variable named `Q1_data`:

https://raw.githubusercontent.com/ismayc/chemistr-book/master/gifs/enter_data.gif

You should use the R Console to check your results by either entering the R code line by line into the Console or by pressing the **Run Current Chunk** button (the green Play button). We see here that `Q1_data` is 6 rows by 2 columns and stores our results in "spreadsheet" form as a `data_frame`.

1.2.5 A nice table R chunk

The `chem_table` function will result in a nicely formatted table being produced in your resulting Word document. You'll need to specify a caption inside the parentheses instead of `"A lovely caption for the table"` default. You'll also see that RStudio will create a second quotation mark when you might not want one. Carefully look over your document if you have errors and make sure there isn't two quotation marks when you only wanted one. You'll see this below:

https://raw.githubusercontent.com/ismayc/chemistr-book/master/gifs/two_quotes.gif

After fixing our error, we have now created a DOCX file that will have our nice table showing the `Q1_data`. Feel free to open up this `Lab1_Light.docx` file now and see that the table was created with the caption "Light Source Table". Feel free to enter a different caption instead.

If you scroll down the R Markdown file, you'll see that there are many more spots where we will be inserting pictures. R Markdown has nicely skipped over these chunks when we pressed **Knit** and we see that in the R Console window under the R Markdown tab.

1.3 Uploading and inserting external pictures

You'll frequently be asked to add pictures into your lab reports that weren't necessarily created using R. We will see how to include plots created inside R in later labs though. If you look over **Q2** in the lab it is asking you to include a picture of the "Absorbance spectrum of water in a 1 cm cuvette". Note that this is specified as the figure caption here.

Mathematical text is also included by surrounding the mathematical expression with dollar signs. This is using LaTeX to produce nicely formatted math text. Here we see the expression H_20 is given using `$H_{2}0$`.

In order to include a picture, you'll need to first upload it to the RStudio Server into the directory where the R Markdown file is located. The GIF below shows you how to do so. Remember that you'll need to know where you saved your image on your computer to specify it here. This example is done on a Mac but similar procedures should be done on a Linux or Windows machine.

`https://raw.githubusercontent.com/ismayc/chemistr-book/master/gifs/upload_photo.gif`

The `Water.jpg` file has been uploaded to my `Lab1-Light` folder. The `include_graphics` function requires that you add the name of the file and its extension here. We now press the **Knit** button to see the resulting Word document:

`https://raw.githubusercontent.com/ismayc/chemistr-book/master/gifs/knitted.gif`

1.3.1 The remainder of the lab

In the remaining portions of the lab you'll be asked to include other pictures from your lab sessions and also include a discussion of your results. This discussion will proceed in much the same way as you would add a discussion into a Word document or Google Doc. You'll be able to immediately discuss what occurred in your R chunks immediately below them. This allows for the reader to easily follow your work.

1.4 Note on white space

As you look over the R Markdown document you'll see that there is always a new line of white space between the discussion and the R chunks and also between each of the R chunks. It is highly recommended that you also follow this workflow. You'll receive some strange errors at times if you try to stack everything together and it's also much harder to follow for another reader of your document if you have everything bunched together. **White space is your friend!**

1.5 Spell-check

Just as I'm sure your English teachers have told you to spell check your documents before submitting, you are also encouraged to do so here. There is a built-in spell check option found near the **Knit** button.

Please run this and carefully read over your lab report before converting it to a PDF and submitting it to Moodle.

1.6 Converting your Word document to PDF

The directions for each lab on Moodle say to upload a PDF version of your lab. You'll see how to create this PDF from inside Microsoft Word for Mac. A similar procedure can be done using OpenOffice (`http://www.openoffice.org/download/`) or Microsoft Word on a PC. You may also have the option to **Save As** a PDF there and you can get to this option by going to **File -> Save As -> File Format: -> PDF** on the Mac if you prefer.

`https://raw.githubusercontent.com/ismayc/chemistr-book/master/gifs/word_pdf.gif`

A

RStudio Desktop

More advanced users of RStudio may be interested in using the RStudio Desktop version installed on their own computer instead of using the server. It is encouraged that all students do this after they have some familiarity with how RStudio works. The RStudio Server is a great way to learn how this works while providing the ability for more advanced users to give support to learning through Shared Projects. Downloading the RStudio Desktop allows for resources to be freed up on the RStudio Server though and if you are performing more advanced calculations it makes more sense to download your own version of RStudio instead of doing so on the RStudio Server.

A.1 Downloading R and RStudio Desktop

It is worth noting that you can't just install RStudio Desktop without installing R as RStudio needs to have R installed in order to run. A step-by-step guide to installing R and RStudio Desktop with screenshots can be found

- here for the Mac and
- here for a PC.

Unless you plan to create PDF directly from R Markdown documents (which requires a multiple gigabyte download of LaTeX) you can skip some of the later steps of the installation.

A.2 Downloading and installing `chemistr`

I created the `chemistr` R package as a way to introduce students of Chem 101/102 at Reed College to R without the sometimes intimidating amounts of code needed to produce specific plots and tables. The package also includes R Markdown lab report templates for each of the labs.

Intermediate users of R and RStudio may be used to using the `install.packages` function to download and install R packages from CRAN (The Comprehensive R Archive Network).

This network is the standard for R packages and, to be on CRAN, packages there go through a series of tests to make sure they are working well. With the `chemistr` package being developed recently (and still under development), it does not currently reside on CRAN. But you can still download a developmental version of it.

Hadley Wickham has created an R package called `devtools` that allows for R packages to be more easily created and shared with others. We will use this package to install the `chemistr` package from my GitHub page. You'll need to enter these commands into the R Console of your RStudio Desktop:

```
install.packages("devtools")
devtools::install_github("ismayc/chemistr")
```

After running these two commands you can check that `chemistr` was installed correctly by entering `library(chemistr)` into the R Console. If you get back to `>` without any error messages, you should be good to go. Now you'll follow the same steps given in each chapter of this book to get the R Markdown template for that specific lab.

File -> **New File** -> **R Markdown** -> **From Template** -> NameOfLab

You are encouraged to run the two lines of code above before you begin each lab as there may have been slight corrections/changes made to the lab templates. You can find a description of each change to the lab templates here with dates given.

B

Lab 1: Light (More Details)

B.1 `chem_table` function

The `chem_table` function is essentially a wrapper function to the `pandoc.table` function in the `pander` package. Here is the code for `chem_table`:

```
chem_table <- function(data, caption){
  names(data) <- pandoc.strong.return(names(data))
  pandoc.table(data, caption = caption, style = "multiline",
               split.tables = Inf)
  cat("\\newline")
}
```

We see here that `chem_table` expects two arguments:

- `data` is a data frame containing the variables you'd like to appear in the table
- `caption` is the caption we'd like to correspond to this table

Remember that you can run `?chemistr::chem_table` to bring up the help documentation for the function.

We first use the `pandoc.strong.return` function that bolds the column names. We then call the `pandoc.table` function in the `pander` package with `data` as our argument and then our entered `caption` argument as the `caption` argument to `pandoc.table`. The last two argument for `style` and `split.tables` are used to ensure the outputted table appears as you might expect it to in the Word document:

- `style` set to `multiline` allow headers and table rows to span multiple lines of text. This may be helpful if you have long column names.

- `split.tables` set to `Inf` ensures that wide tables will not be split into multiple tables

B.2 The `include_graphics` function

The `include_graphics` function enables you to include pictures that are stored as image files (`*.png` or `*.jpg`, for example) into your Word document via R Markdown. It is a function in the `knitr` package. If you'd like more information on how to use `include_graphics` run `?knitr::include_graphics` in the R Console. The important argument here is `path` which tells R where to look for the file you want to include.

If you include your pictures in the same directory as your Rmd file you need only specify the name of the file here. If you have a `figure` folder in the same folder as your R Markdown file, you'll need to specify that in something like:

```
knitr::include_graphics("figure/myimage.png")
```

Whatever you specify as the chunk option `fig.cap` will appear as the figure caption. You are also encouraged to name the R chunk, which will allow you to reference the figure later in your document:

````
```{r myimage, echo=FALSE, fig.cap="Here is my picture"}

knitr::include_graphics("figure/myimage.png")

```
````

We could then reference our picture in the text of our document by using `\@ref(fig:myimage)`.

C

Bibliography