

# PSTAT 5LS Lab 0

TA Name Here

Summer 2025

# Section 1

## Introduction to PSTAT 5LS

# Section Logistics

- Sections meet twice a week
- Section attendance is worth 10% of your overall grade in PSTAT 5LS
- We understand that sometimes you'll need to miss. As part of the "life happens" policy, your lowest 2 attendance scores (absences) will be dropped
- Use these 2 drops wisely! Save them for a rainy day!

# Homework Logistics

- There will be 8 homework assignments during this summer session
- Homework will be available in Gradescope and will be due 11:59 p.m. Pacific on most Tuesdays and Fridays
- Late work is not accepted and extensions are not permitted because the solutions are posted at the deadline
- Your lowest 1 homework score will be dropped
- Use this 1 HW drop wisely! Don't just blow off homework one week and then wish you had that drop later in the quarter!

## Section 2

### Learning Objectives

# R Learning Objectives

- 1 Learn the difference between R, RStudio, and R Markdown
- 2 Become familiar with the RStudio interface
- 3 Understand key components of an R Markdown document
- 4 Learn how to use R to do basic calculations

# Statistical Learning Objectives

- 1 Understand that technology can be helpful in statistics

## Section 3

### Lab Tutorial



# Getting Started: What is R?

In Statistics, we often use computers to analyze data. There are a lot of “statistical computing environments” that can help you do statistical analyses. One of the most popular (and powerful) is called R. R is a programming language that is designed for manipulating data, doing calculations, and making graphical displays. R works by writing **R code**.

That might sound scary, but *don't worry*: this is not a programming class. Over the course of the quarter, you'll learn how to edit and write some basic R code to help you analyze data to answer research questions. Our goal with PSTAT 5LS labs is to help you learn the basics of R and R coding, but through the lens of answering statistical questions.

# What are all these “R” terms?

There are a lot of “R” words floating around. What’s going on?

- *R* is a “statistical computing environment” that’s designed for manipulating data, generating plots, and performing analyses. It’s also a programming language. You’ll be *using R* this quarter.
- *RStudio* is an “integrated development environment (IDE)” for R (you’ll never have to hear the term IDE again in this class). Basically, RStudio is a pretty interface that makes working with R easier. You use R inside of RStudio. If R is ice cream, RStudio is the cone or cup.
- *R Markdown* is a way to write pretty analysis reports that combines R code, R output (plots, analysis results, etc.) and text in one document. This lab document is an R Markdown report!

# JupyterHub

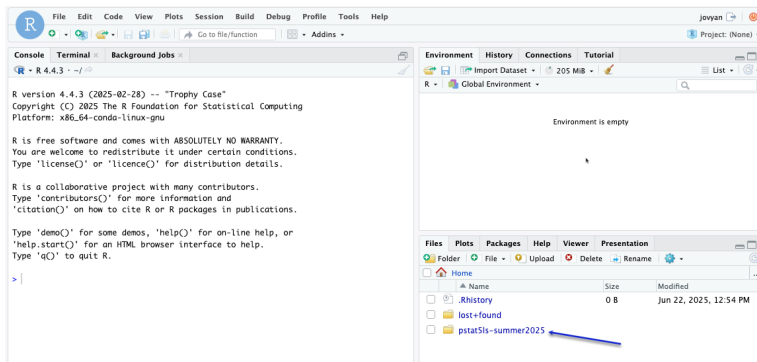
We will be accessing R and RStudio through JupyterHub. JupyterHub is a web-based multiuser environment that will allow us to share with you the files you need for lab. LSIT has provided us with access to JupyterHub.

Each time we begin a lab or when you work on a data analysis question for homework, navigate to <https://bit.ly/pstat5ls-m25>.

This will open your “instance” of JupyterHub and copy all necessary files for your PSTAT 5LS work that week.

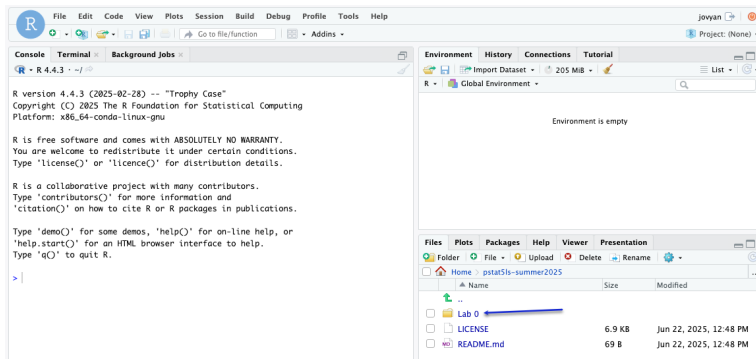
# Initial View of RStudio

Click on the 'pstat5ls-summer2025' folder.



# Accessing the Lab

After you have clicked on the 'pstat-5ls-summer2025' folder, click on the folder for this lab (lab0) to open it:



# RStudio Project Screen

The screenshot displays the RStudio Project Screen with the following components:

- Source Editor:** Contains an R Markdown file named "lab0-notes-drmiller.Rmd". The file includes a title, author, date, and output format, followed by sections for Lab Notes and Lab Code.
- Environment Pane:** Shows the current environment, which is empty.
- Console:** Displays the R version (4.3.3) and copyright information.
- Files & Data Sets:** A list of files and folders in the project, including .Rhistory, assets, lab0-intro-to-R.Rproj, lab0-notes-drmiller.Rmd, lab0-slides.Rmd, penguins.csv, and tuition\_sample.csv.

**R Markdown file**

```

1 ---
2 title: "PSTAT 5LS Lab 0"
3 author: "Your Name Here"
4 date: "Tuesday, August 6, 2024"
5 output: pdf_document
6 ---
7
8 # Lab Notes
9
10 Use this place to take any notes during your lab section.
11
12 - Type any notes here
13 - Add new notes by using the dash '-' to create a list
14
15 # Lab Code
16
17 You can follow along with the TA's tutorial by running the chunks below.
18 1:1 PSTAT 5LS Lab 0
  
```

**environment pane**

Environment is empty

**R console**

```

R version 4.3.3 (2024-02-29) -- "Angel Food Cake"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-conda-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
  
```

**files & data sets**

Name	Size	Modified
..		
.Rhistory	0 B	Aug 6, 2024, 2:47 PM
assets		
lab0-intro-to-R.Rproj	205 B	Aug 6, 2024, 2:47 PM
lab0-notes-drmiller.Rmd	2 KB	Aug 6, 2024, 2:47 PM
lab0-slides.Rmd	11.2 KB	Aug 6, 2024, 2:47 PM
penguins.csv	16.4 KB	Aug 6, 2024, 2:47 PM
tuition_sample.csv	27.1 KB	Aug 6, 2024, 2:47 PM

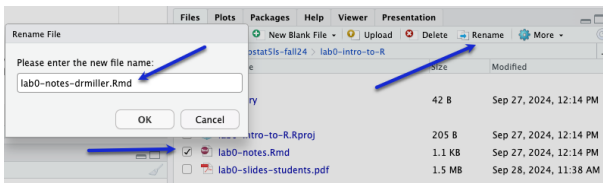
# Working with R in JupyterHub

When you navigate to <https://bit.ly/pstat5ls-m25> you are telling JupyterHub to bring in all of the files that Professor Miller has prepared for PSTAT 5LS. This means that any files you have worked on will be overwritten.

To avoid your work being overwritten, **rename** both the notes and report files. For Lab 0, you want to rename lab0-notes.Rmd. You could add your name to the file name (e.g., lab0-notes-drmiller.Rmd).

# Renaming Files

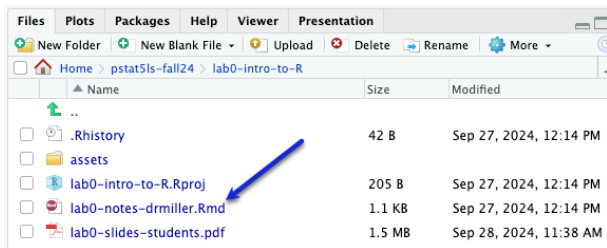
To rename a file, check the box next to the file you want to rename. Click “Rename” in the files pane, rename the file, then click OK.





# Renaming Files

Your file will then appear with its new name.



# R Markdown

R Markdown lets you combine text, R code, and plots in one pretty, reproducible report. If you're curious about this, you can find more details on using R Markdown at <http://rmarkdown.rstudio.com>.

R Markdown runs code contained in “chunks”. A chunk looks like this:

```
```${r helloWorld}  
print("Hello world!")  
```
```



```
print("Hello world!")
```

```
## [1] "Hello world!"
```

# R Markdown Chunk

```
```${r helloWorld}  
print("Hello world!")  
```
```



Notice that the code, `print("Hello world!")` is contained between three backticks (`````, below the `esc` key on a US English keyboard; the same key as the tilde) and some stuff in curly braces `{ }`. This is how R Markdown knows where your chunks start and stop. We will almost always provide pre-made chunks for you to use.

# Running a Chunk

You can just run a single chunk by clicking the green “play” button in the upper right corner of the chunk. It’s usually a good idea to click the “Run All Chunks Above” button immediately to the left of the play button first. So you should click Run All Chunks Above, then the play button.

```
```${r helloWorld}  
print("Hello world!")  
```
```

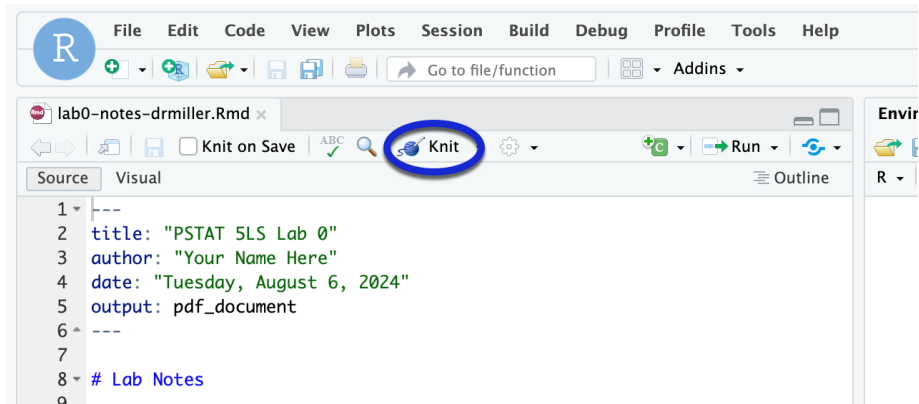


If you can’t find the play button, just highlight the code you want to run and click the Run button in the top right corner of the editor pane.

Let’s try this code together in the tryIt1 chunk in your notes document. Note that the code is already entered for you in this first code chunk.

# Knitting a Document

When you click the **Knit** button in RStudio, a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.



# Tips for R Markdown

- 1 **Knit and knit often:** Frequently knitting your document will help you make sure that all your code works and that the document looks the way you want.
- 2 **Don't be afraid to experiment:** Nobody gets things exactly right the first time, and we all forget how things work sometimes. Keep trying, and you'll eventually get what you want!
- 3 **Formatting:** You can make text **bold** by surrounding it with two asterisks (**\*\***) and *italic* by surrounding it with one asterisk (\*). You can make text look like `code` by surrounding it with single backticks.

# Using R as a Calculator

At it's most basic, R is a fancy calculator.

Let's try this example together in the tryIt2 chunk in your lab notes!

Remember, you can run the code in this chunk by clicking the green “play” button in the upper right corner of the chunk.

```
5 * 7
```

```
## [1] 35
```

When you run the chunk, you'll see a [1] before the output of 35. *Just ignore this. The result is 35.*

All of the symbols you think you'd use to do math work: + for addition, - for subtraction, \* for multiplication, / for division, ^ for exponentiation.

# R Function Syntax

We'll be using R primarily through the use of *functions*. A function generally looks something like this:

```
function.name(argument1, argument2, ...)
```

Notice that we have the name of the function, followed immediately by an open parenthesis '(' without a space, then a sequence of *arguments* to the function, separated by commas ',', then a close parenthesis ')'.



# R “Assignment” Syntax

Often, we want R to remember the results of a calculation so that we can use it later. We can give the result of some code a name by **assigning** it to something.

```
x <- 36 / 6
```

We read that code as “x gets 36 / 6”. The arrow is made using the less-than symbol (<, shift + comma on a US English keyboard) and a hyphen.

Fill out the tryIt3 code chunk in your notes to assign 36 divided by 6 to the variable x.

# R “Assignment” Syntax

Now, we’ve stored the result as `x`, and R will remember that `x` is 6. You can see in the environment pane in RStudio (top right) that there’s now a “value” called `x` and it’s 6. You can also access the value of `x` by typing `x` into R. Check it out:

```
x
```

```
## [1] 6
```

Use the `tryIt4` code chunk in your notes to see the value that R stored for the variable `x`.

# R is Case-Sensitive

R is “case-sensitive”, which means that upper-case letters are *different from* lower-case letters. Notice what happens when we ask R for the value of X:

```
X
```

```
## Error: object 'X' not found
```

Use the tryIt5 code chunk in your notes to see what happens if you type in a capital X.

When giving things names in R, you can only use a combination of letters, numbers, periods, and underscores, and the **names have to start with a letter or a period**. People tend to use underscores or periods instead of spaces.

# Common Errors

Watch what happens when you try to assign something to a “bad” name:

```
tik tok <- 12
```

```
## Error in parse(text = input): <text>:1:5: unexpected symbol
## 1: tik tok
##      ^
```

```
4eva <- 4 * 2
```

```
## Error in parse(text = input): <text>:1:1: unexpected input
## 1: 4ev
##      ^
```

```
_hi_mom <- 5^2
```

```
## Error in parse(text = input): <text>:1:2: unexpected symbol
## 1: _hi_mom
##      ^
```

# Common Errors

The errors saying “unexpected symbols” or “unexpected input” are R’s way of telling you that these names are not allowed, and that you should use a different name. Here’s how we’d correct these:

```
tiktok <- 12
```

```
forever <- 4 * 2
```

```
hi_mom <- 5^2
```

# Knit Your Document One Final Time

At the end of any of the labs we do, be sure to knit one more time. This is so you have a finalized version of your work just in case you have made any changes.

Be sure that a popup window shows you the file! Otherwise there is an error in the creation of the document. If this happens, you will need to troubleshoot the R markdown file. I am happy to help you with this.

Also be sure that you read over your file and make sure that it looks right! If pieces are missing, then you have an error somewhere in the creation of the document. You should troubleshoot the R Markdown file.

# Where Do We Go from Here?

During our next section meeting, we will use R for data analysis. You'll learn how to

- load a data file
- examine the data file
- create tables, graphical displays, and summary statistics