

# Introduction to R and RStudio

Ecological Systems Modeling

Jan 8-10, 2024

# About me



## Current position

- Snr. Research Associate, Oregon IPM Center & Hort.

## Research interests

- Modeling, ecology, biogeography, conservation, biological invasions, and genetics

## Hobbies

- Running, hiking, biking, moving!
- Gardening, native habitat restoration
- Artistic and creative endeavors

# Learning objectives

- Explain what R and RStudio are, what they are used for, and how they relate to each other
- Describe the purpose of the RStudio Script, Console, Environment, and Plots panes
- Create an R project and understand the purpose of the working directory
- Copy a folder of files on Jupyter Hub and open them in RStudio

# Learning objectives

- Explain what R and RStudio are, what they are used for, and how they relate to each other
- Describe the purpose of the Script, Console, Environment, and Plots panes in RStudio
- Create an R project and understand the purpose of the working directory
- Copy a folder of files on Jupyter Hub and open them in RStudio

# What is R?

- First implemented beginning in early 1990s (based on S language)
- First designed for statistics but can now do much more



Source: [blog.internshala.com](http://blog.internshala.com)

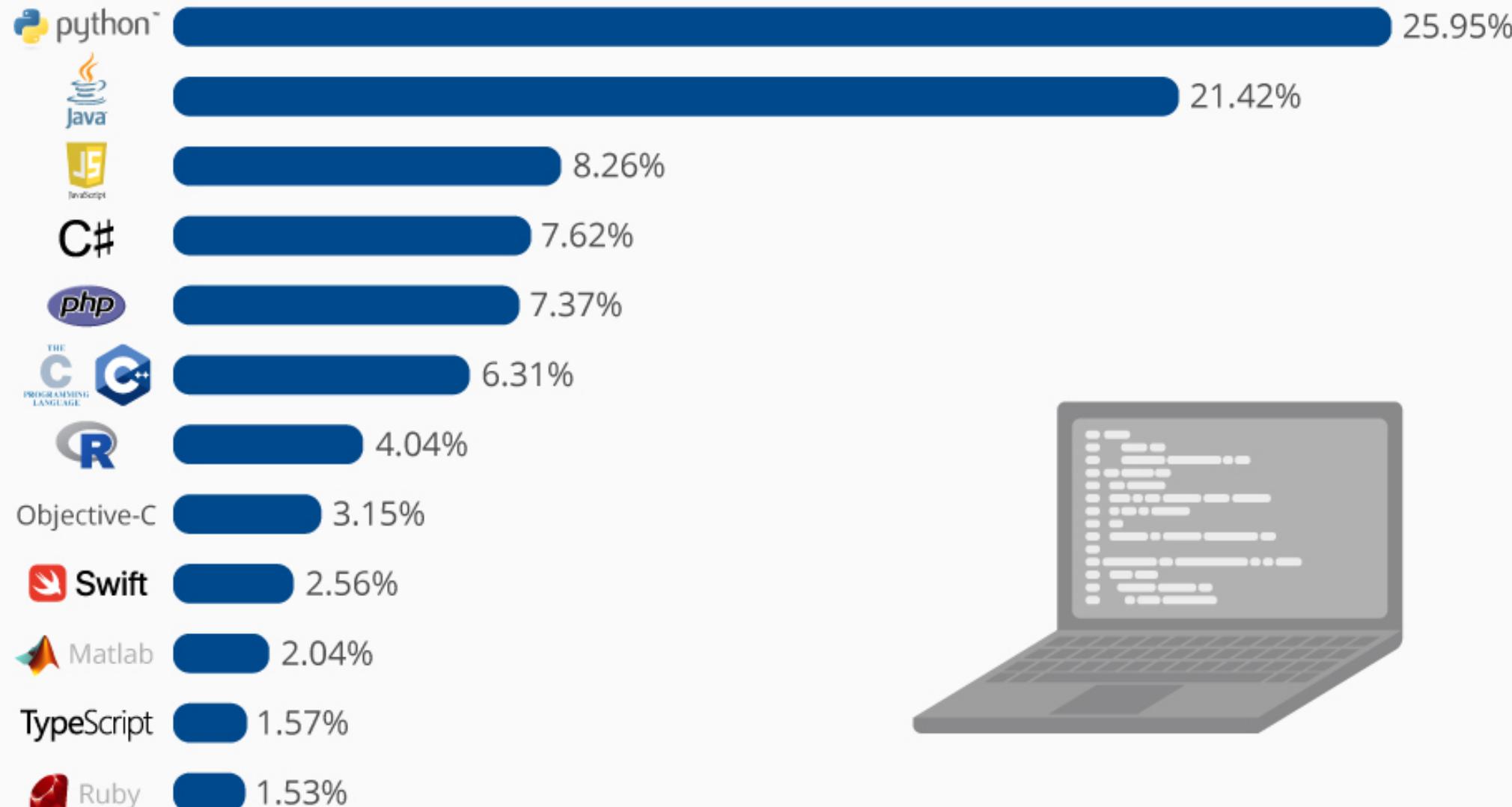
# Why learn R?

- Commonly used programming language
- Does more than just statistics
- Great for reproducibility
- Interdisciplinary and extensible
- Large and friendly user community
- Facilitates learning and comprehension

# R is a commonly used programming language

## The Most Popular Programming Languages

Share of the most popular programming languages in the world\*



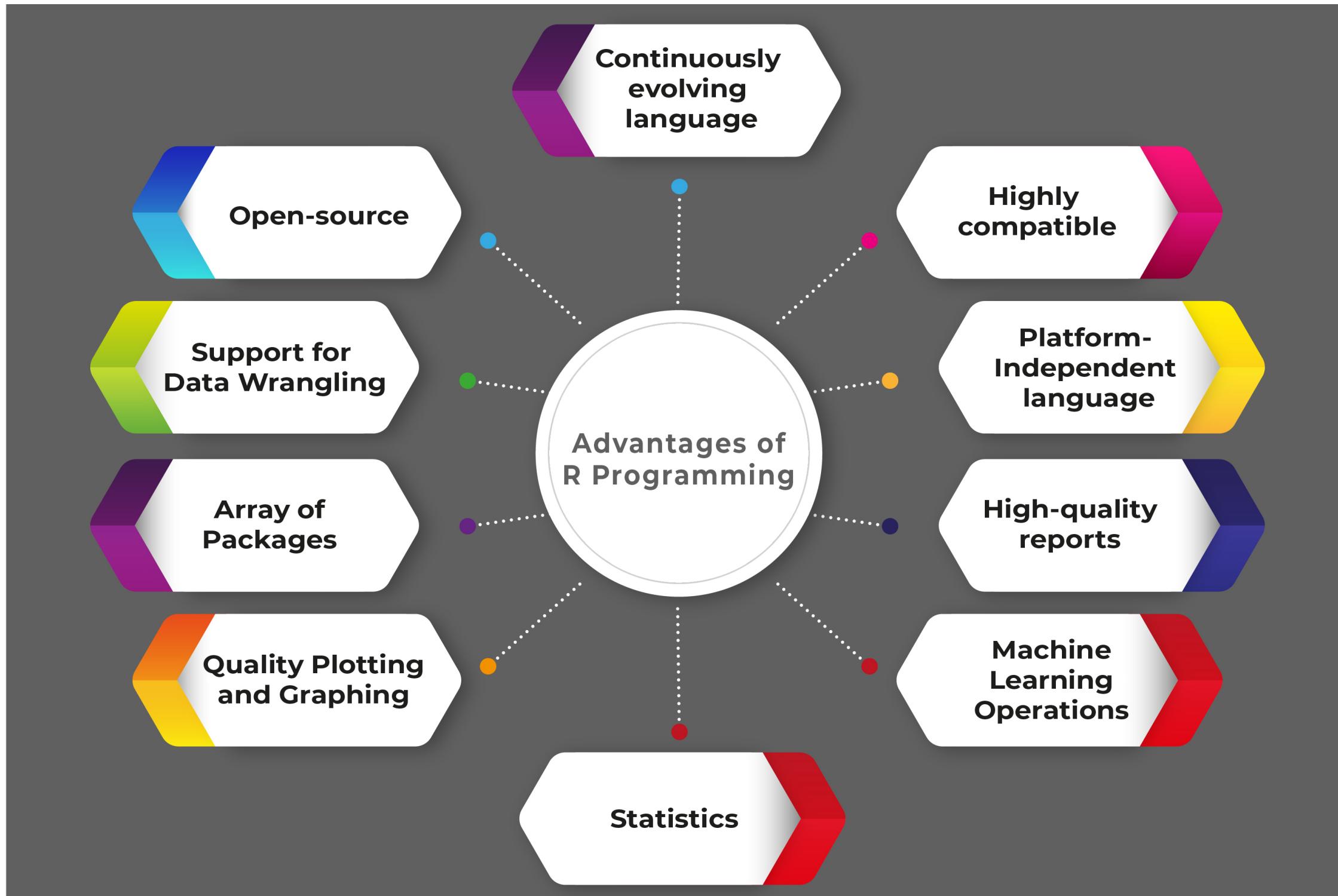
\* Based on the PYPL-Index, an analysis of Google search trends  
for programming language tutorials.



@StatistaCharts

statista

# R is more than just statistics



Source: [www.icertglobal.com](http://www.icertglobal.com)

# R is great for reproducibility

- No need to remember a series of points and clicks
- Quickly re-create plots, re-run models and analyses, etc.
- Makes steps used in your analysis clear
- Can be inspected by someone for feedback



Source: [www.cos.io](http://www.cos.io)

# Example: R scripts in GitHub for one of my pubs

The screenshot shows a GitHub repository page for 'Cps-climSuit-modeling'. The repository is public and has 60 commits. The main branch is 'main'. The repository description is: 'Project and R scripts used in climate suitability modeling study of *Calonectria pseudonaviculata*'. The README file contains a brief description of the project's purpose and its relation to a published paper by Barker et al. (2022). The Languages section shows that the code is written in R.

**Code**

**Commits**

**bbarker505 Merge branch 'main' of https://github.com/bbarker505/Cps-climSuit-mod... ad371ab · 2 years ago 60 Commits**

**CLIMEX/Final\_outfls** Final R1 updates 2 years ago

**CliMond\_raw** Update 4-15-22 2 years ago

**Climate\_pt\_analysis** Update 9/28/21 3 years ago

**ENMTML** Updates to gitignore 2 years ago

**Records** Delete Records/Subsampled directory 2 years ago

**Scripts** Final updates 5-31-22 2 years ago

**.gitattributes** Initial commit 3 years ago

**.gitignore** Update .gitignore 2 years ago

**Boxwood\_blight.Rproj** Initial commit 3 years ago

**README.md** Update README.md 2 years ago

**README**

**Climate suitability modeling study of *Calonectria pseudonaviculata***

Files in this repository replicate the majority of results presented in "Potential distribution of invasive boxwood blight pathogen (*Calonectria pseudonaviculata*) as predicted by process-based and correlative models" by Barker et al. (2022). The repository contents include one R project, 11 R scripts, raw and processed CliMond data used for modeling (27 variables), the full and subsampled occurrence record datasets used for modeling, and the CLIMEX outfile.

**About**

Project and R scripts used in climate suitability modeling study of *Calonectria pseudonaviculata*

**Readme**

**Activity**

**1 star**

**1 watching**

**0 forks**

**Releases**

No releases published

[Create a new release](#)

**Packages**

No packages published

[Publish your first package](#)

**Languages**

R 100.0%

Barker, B. S., L. Coop, and C. Hong. 2022. Biology 11:849

# R is interdisciplinary and extensible

- 10,000+ packages to extend its capabilities
  - Use statistical approaches from many scientific disciplines
  - e.g., image analysis, GIS, time series, population genetics, and much more

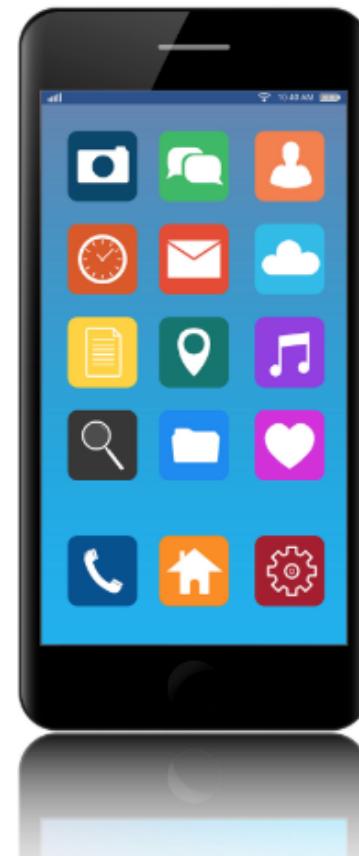


# R packages

- Libraries of functions, code, and documentation
- Analogy: apps that you download on your phone

---

R: A new phone



R Packages: Apps you can download



---

Source: ModernDive

# Wrangle and manipulate data

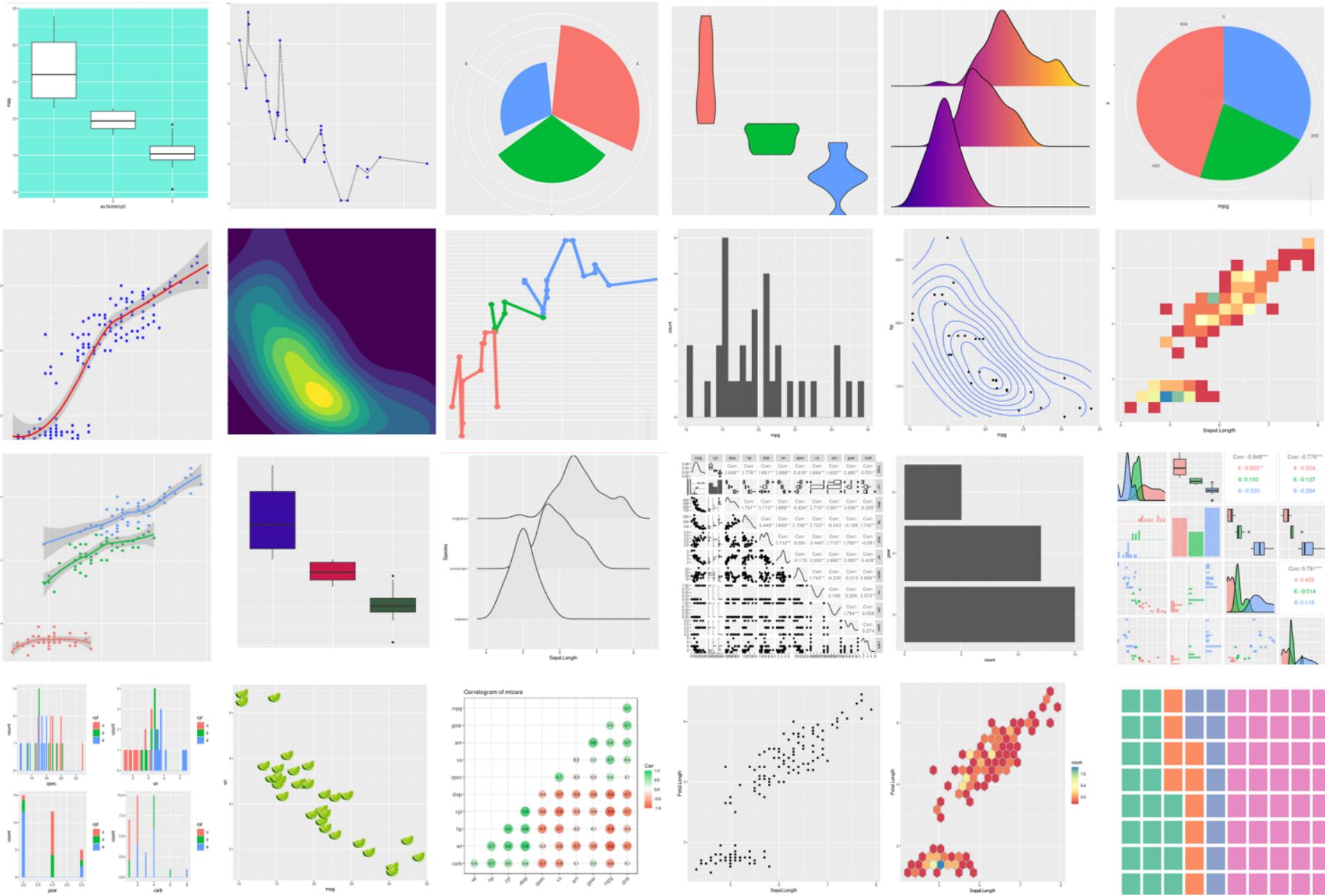
- tidyverse packages



Source: Allison Horst

# Make beautiful plots

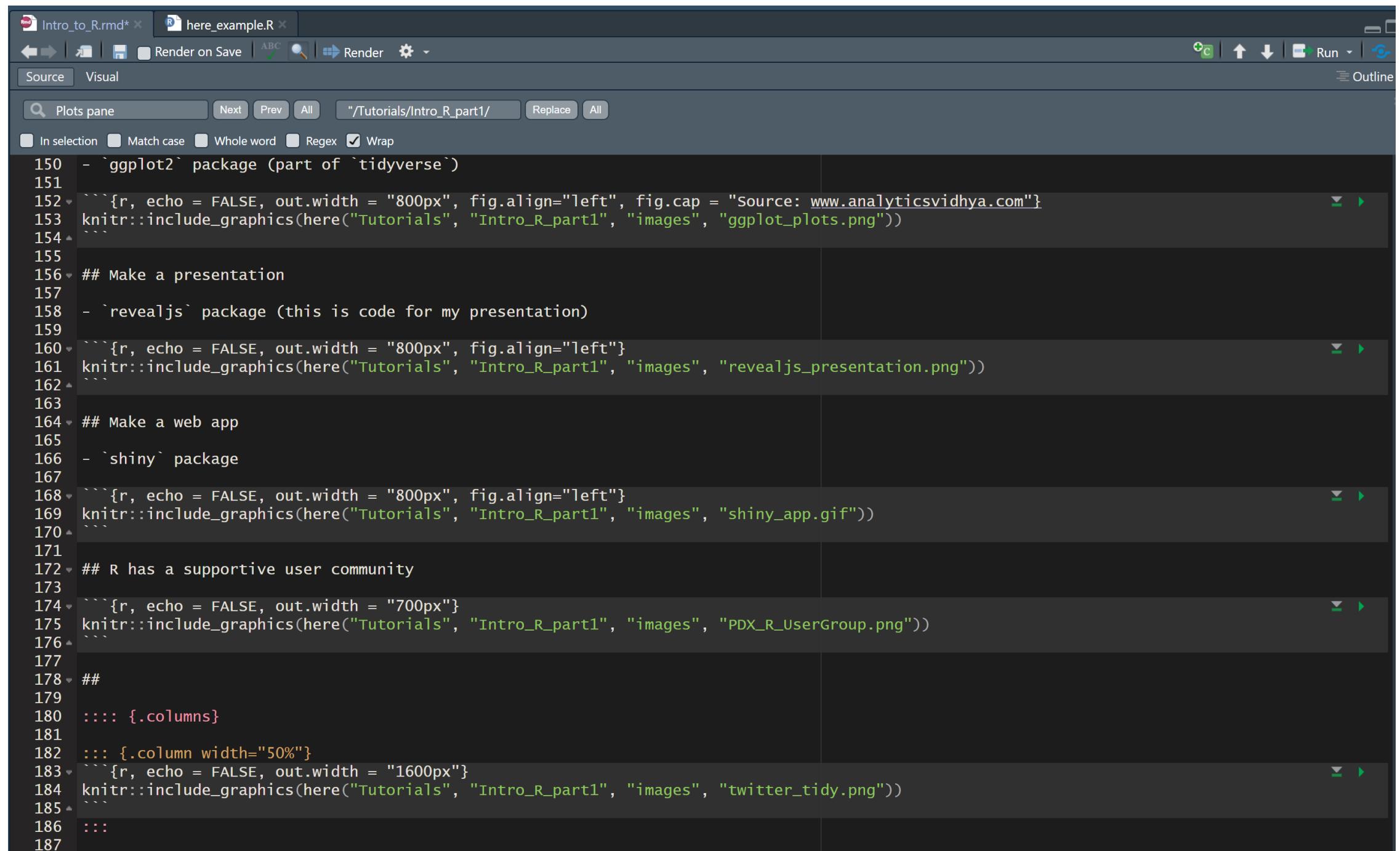
- `ggplot2` package (part of `tidyverse`)



Source: [www.analyticsvidhya.com](http://www.analyticsvidhya.com)

# Make a presentation

- `revealjs` package (this is code for my presentation)



The screenshot shows the RStudio interface with two tabs open: "Intro\_to\_R.rmd\*" and "here\_example.R". The "Source" tab is selected, displaying R code. The code includes several code blocks (indicated by triple backticks) that render images and graphics. The code is organized into sections like "ggplot2" package, "revealjs" package, "shiny" package, and "R has a supportive user community". The code uses the `knitr::include\_graphics` function to include images from the "Tutorial" directory.

```
150 - `ggplot2` package (part of `tidyverse`)
151
152 ````{r, echo = FALSE, out.width = "800px", fig.align="left", fig.cap = "Source: www.analyticsvidhya.com"}
153 knitr::include_graphics(here("Tutorials", "Intro_R_part1", "images", "ggplot_plots.png"))
154 ```
155
156 ## Make a presentation
157
158 - `revealjs` package (this is code for my presentation)
159
160 ````{r, echo = FALSE, out.width = "800px", fig.align="left"}
161 knitr::include_graphics(here("Tutorials", "Intro_R_part1", "images", "revealjs_presentation.png"))
162 ```
163
164 ## Make a web app
165
166 - `shiny` package
167
168 ````{r, echo = FALSE, out.width = "800px", fig.align="left"}
169 knitr::include_graphics(here("Tutorials", "Intro_R_part1", "images", "shiny_app.gif"))
170 ```
171
172 ## R has a supportive user community
173
174 ````{r, echo = FALSE, out.width = "700px"}
175 knitr::include_graphics(here("Tutorials", "Intro_R_part1", "images", "PDX_R_UserGroup.png"))
176 ```
177
178 ##
179
180 :::: {.columns}
181
182 ::: {.column width="50%"}
183 ````{r, echo = FALSE, out.width = "1600px"}
184 knitr::include_graphics(here("Tutorials", "Intro_R_part1", "images", "twitter_tidy.png"))
185 ```
186 :::
```

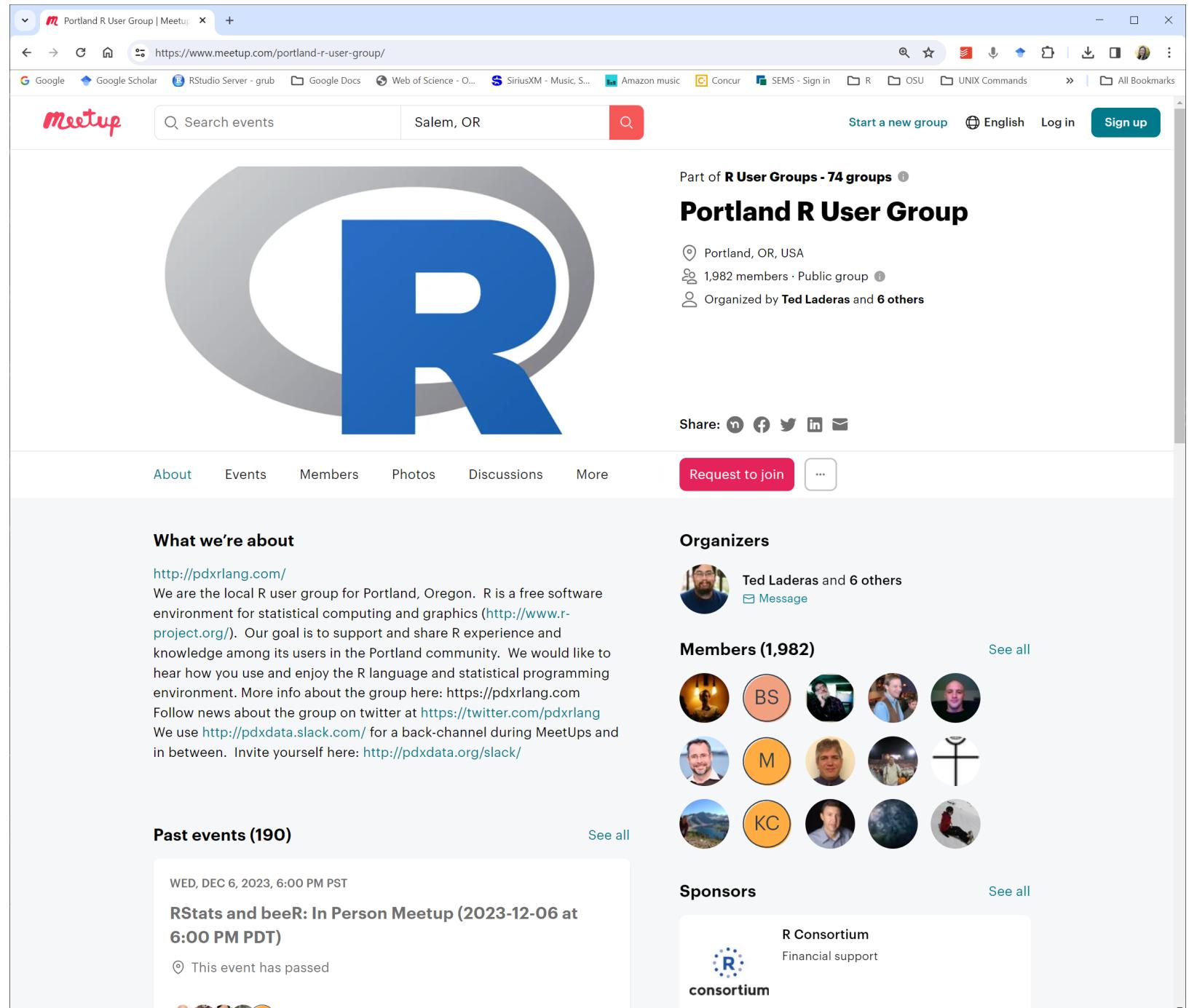
# Make a web app

- shiny package

The screenshot shows a web browser window for RStudio Connect at the URL <https://colorado.rstudio.com/rsc/connect/#/apps/0287f7d9-4d55-4813-8852-680f54beaad1/access>. The page title is "Content / Example Palmer Penguins Shiny Dashboard". The dashboard itself has a yellow header bar with the title "Example Palmer Penguins Dashboard". On the left, there's a sidebar with the heading "Palmer Penguins dataset" and a paragraph explaining its purpose: "The goal of palmerpenguins is to provide a great dataset for data exploration & visualization, as an alternative to `iris`." Below this is a link to the GitHub repository: <https://allisonhorst.github.io/palmerpenguins/>. The main content area contains three cards: "Number of Penguins" (344), "Number of Species" (3), and "Number of Islands" (3). Below these are two plots: a scatter plot titled "Flipper and bill length Dimensions for Adelie, Chinstrap and Gentoo Penguin" showing Bill length (mm) vs. Flipper length (mm) for three species, and a histogram titled "Penguin flipper lengths" showing Frequency vs. Flipper length (mm) for the same three species. At the bottom, there are three cartoon penguins labeled "CHINSTRAP!", "GENTOO!", and "ADELIE!". The right side of the screen shows the RStudio Connect application settings, including "Sharing settings" (set to "Anyone - no login required"), "Who can view or change this Application" (listing "Isabella Velasquez" and "Andrie de Vries"), and "Who runs this content on the server" (set to "The default user"). There are also sections for "Content URL" and "Customize".

# R has a supportive user community

<https://www.meetup.com/portland-r-user-group>



The screenshot shows the homepage of the Portland R User Group on Meetup.com. The page features a large blue 'R' logo with a grey outline. Key statistics are displayed: 1,982 members, organized by Ted Laderas and 6 others. A 'Request to join' button is prominent. Below the main header, sections include 'What we're about' (describing the group's purpose and links to pdxrlang.com), 'Past events (190)' (listing a recent event: RStats and beeR: In Person Meetup on Dec 6, 2023), 'Organizers' (Ted Laderas and 6 others), 'Members (1,982)' (with a grid of member profiles), and 'Sponsors' (R Consortium).

Portland R User Group | Meetup

Part of **R User Groups - 74 groups**

**Portland R User Group**

Portland, OR, USA  
1,982 members · Public group  
Organized by **Ted Laderas** and **6 others**

Share: [n](#) [f](#) [t](#) [in](#) [e](#)

[Request to join](#)

**What we're about**

<http://pdxrlang.com/>  
We are the local R user group for Portland, Oregon. R is a free software environment for statistical computing and graphics (<http://www.r-project.org/>). Our goal is to support and share R experience and knowledge among its users in the Portland community. We would like to hear how you use and enjoy the R language and statistical programming environment. More info about the group here: <https://pdxrlang.com>. Follow news about the group on twitter at <https://twitter.com/pdixrlang>. We use <http://pdxdata.slack.com/> for a back-channel during Meetups and in between. Invite yourself here: <http://pdxdata.org/slack/>

**Past events (190)**

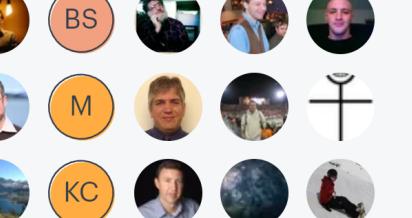
WED, DEC 6, 2023, 6:00 PM PST  
**RStats and beeR: In Person Meetup (2023-12-06 at 6:00 PM PDT)**  
This event has passed

**Organizers**

 **Ted Laderas** and 6 others [Message](#)

**Members (1,982)**

See all



**Sponsors**

See all

**R Consortium**  
Financial support

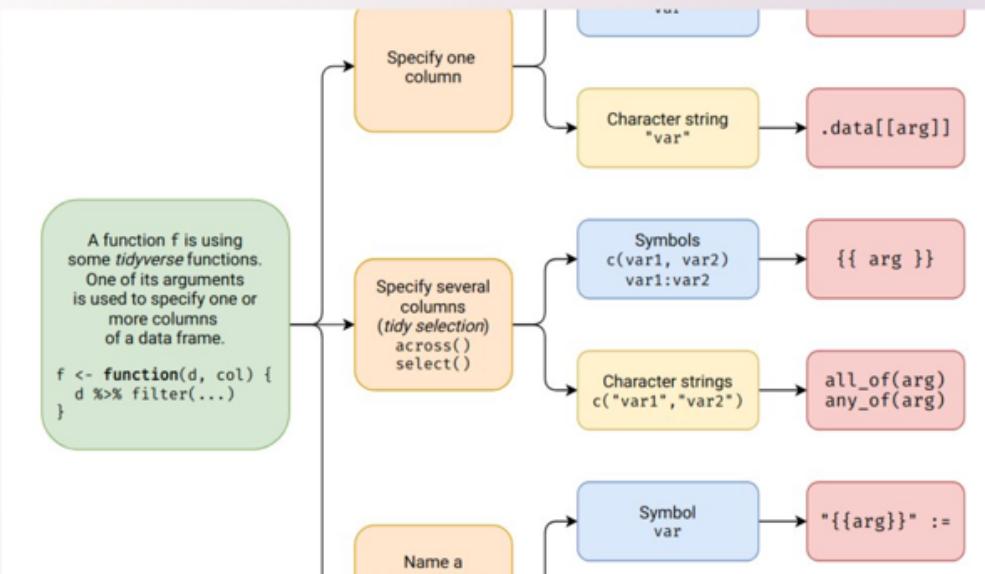
 **consortium**



Brittany Barker  
@BrittanySBarker



I had a great time presenting examples of `#tidyverse` functions in `#RStats` during my demo on tidy evaluation last night at the Aggregate meetup for [@pdxrlang](#). Check it out to learn more! [github.com/smithjd/demo-t...](https://github.com/smithjd/demo-t...)



8:31 AM · 23 Mar, 2022

9 shares 30 likes

PDX R Retweeted

Brittany Barker  
@BrittanySBarker

...

In case anyone wants to do a simple demo to better learn the 'purrr' package R, here is the GitHub repo where I deposited my demo that I gave last night to [@pdxrlang](#) Aggregate #41 Meetup. Stop over-using 'for' loops and purrrrr. [github.com/bbarker505/pur...](https://github.com/bbarker505/pur...)



# Check out upcoming PDX R User Group events

 Ted Laderas (He/Him) • 1st  
I make complex topics in computing, bioinformatics, and data science ap...  
2d • 

Hey PDX #rstats people and beyond! We're doing an online lightning talk session on February 6.

You should give a talk! It's a super friendly audience. You do not have to be from Portland. Just join our meetup as a member.

<https://lnkd.in/gdN2B4XU>



PDX-R Lightning Talks!, Tue, Feb 6, 2024, 6:30 PM | Meetup  
[meetup.com](https://www.meetup.com/PDX-R/events/25780000/) • 1 min read

 You and 12 others      3 reposts

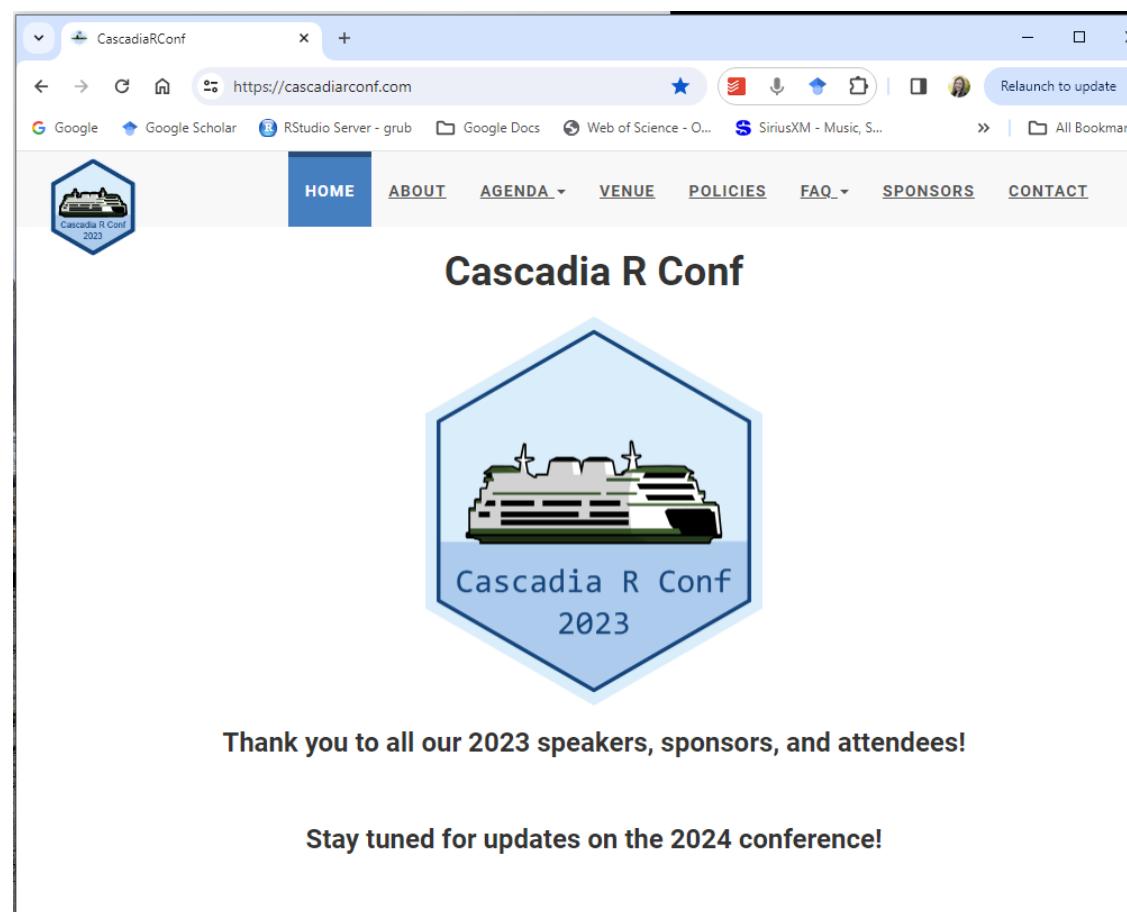
Reactions



 Love     Comment     Repost     Send

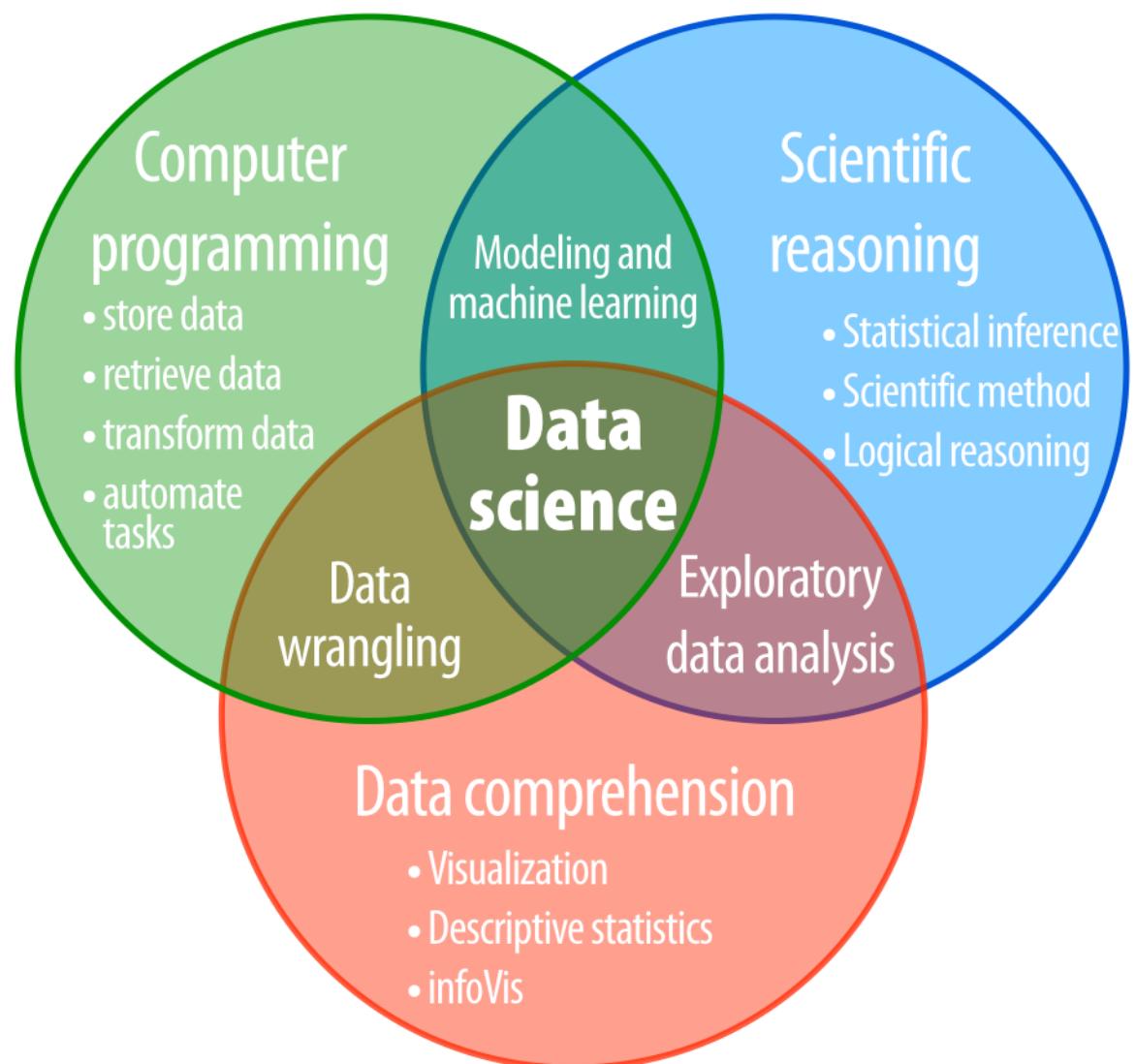
# Cascadia R Conf 2024

- Conference that brings together professionals and students who use R
- Various fields represented: life sciences, medicine, stats, teaching, etc.
- Next conference will be in Seattle in June 2024 (date TBD)
- Stay tuned at: <https://cascadiarconf.com/>



# Facilitate learning and comprehension

- Computer programming is a key skill set for data science
- Hands-on work with your data, analyses, etc., provides more in depth understanding

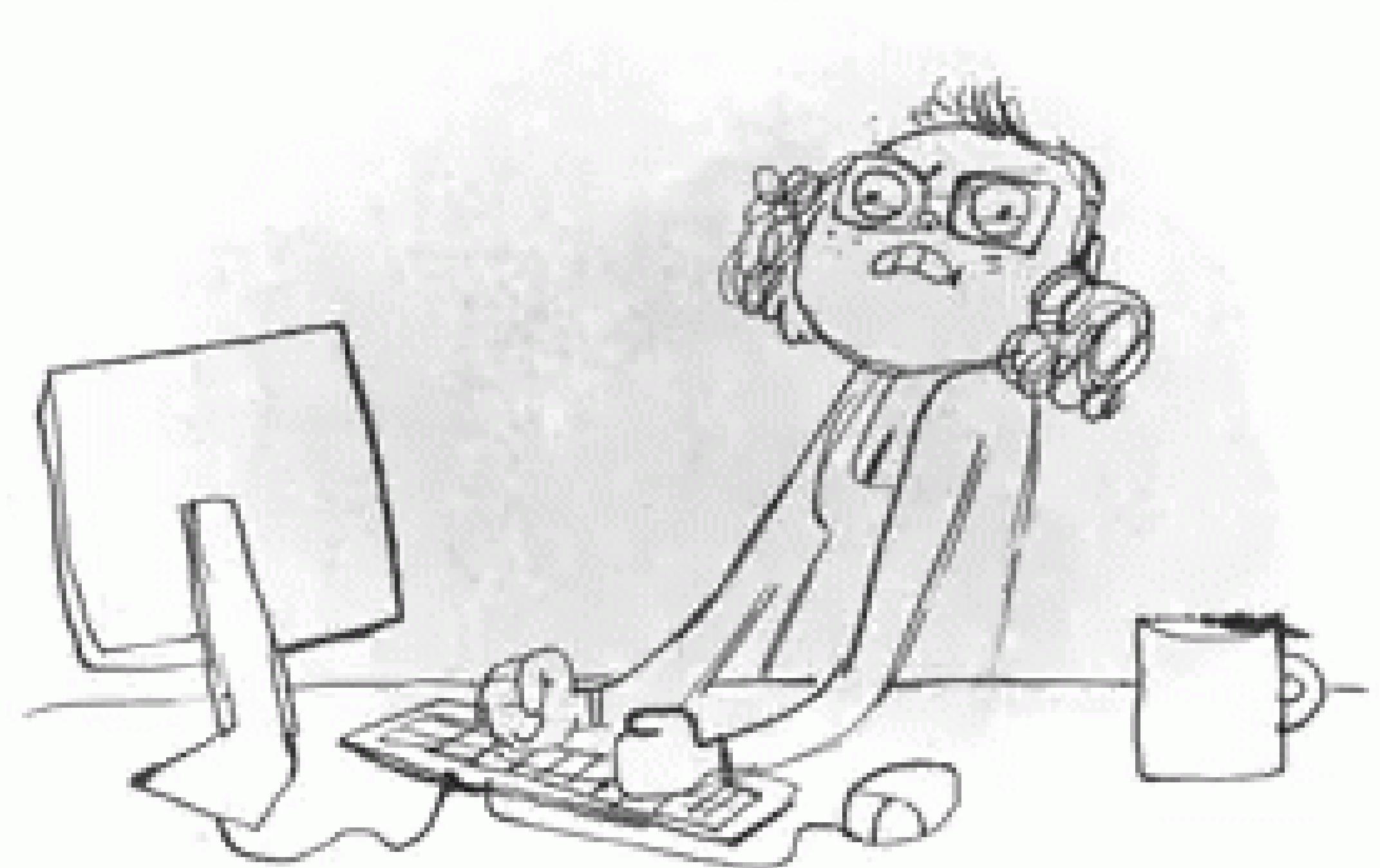


# Limitations of R

- Steep learning curve, but [RStudio](#) can help overcome this limitation
- Hungry for physical memory: stores all data in physical memory
- Slower execution: sometimes not as fast as MATLAB and Python



Even after years of experience with R, I have days where I....



# RStudio

- Re-branded as Posit (more inclusive of other programming langs)
- Free and open source IDE (integrated development environment)
- Many convenient features and tools that make it easier to
  - write and troubleshoot code
  - interact with objects in the R environment
  - simultaneously view graphs, data tables, code, and outputs



# Analogy of difference between R and RStudio

R: Engine



RStudio: Dashboard



Source: moderndive.com

# Learning objectives

- Explain what R and RStudio are, what they are used for, and how they relate to each other
- **Describe the purpose of the Script, Console, Environment, and Plots panes in RStudio**
- Create an R project and understand the purpose of the working directory
- Copy a folder of files on Jupyter Hub and open them in RStudio

# We will use RStudio on Jupyter Hub at OSU

- Everyone log on to [HORT499X](#) in Canvas

# We will use RStudio on Jupyter Hub at OSU

- Navigate to ([Modules](#) -> [Jupyter Hub](#))
- Click on [Hub-HORT-499X](#)

The screenshot shows the Oregon State University Canvas LMS interface. The left sidebar has a dark theme with orange highlights for selected items. The 'Modules' item is selected, indicated by a vertical bar. The main content area shows the 'ST/ECOLOGICAL SYSTEMS MODE...' course. The 'Modules' tab is active, showing the 'Winter 2024' term. A navigation bar includes 'Collapse All', 'View Progress', a checked 'Publish All' button, a '+ Module' button, and a three-dot menu. Below this, the 'Jupyter Hub' module is expanded, showing the 'Hub-HORT-499X' instance, which is also checked. Further down, the 'Week 1 - Conceptual models & Intro to R' module is expanded, showing '(1) Lab assignment'. The top right corner shows '6d Student View' and a back arrow.

# We will use RStudio on Jupyter Hub at OSU

- Navigate to ([Modules](#) -> [Jupyter Hub](#))
- Click on [Hub-HORT-499X](#)
- Click on [Load Hub-HORT-499X in a new window](#)

ST/ECOL... > Modules > Jupyter Hub  
> Hub-HORT-499X

This tool needs to be loaded in a new browser window

Load Hub-HORT-499X in a new window

Winter 2024

Home

Modules

Discussions

Announcements

Syllabus

Assignments

Pages

Quizzes

Zoom

Account

Dashboard

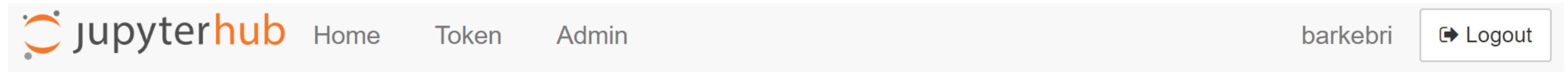
Courses

Calendar

Inbox

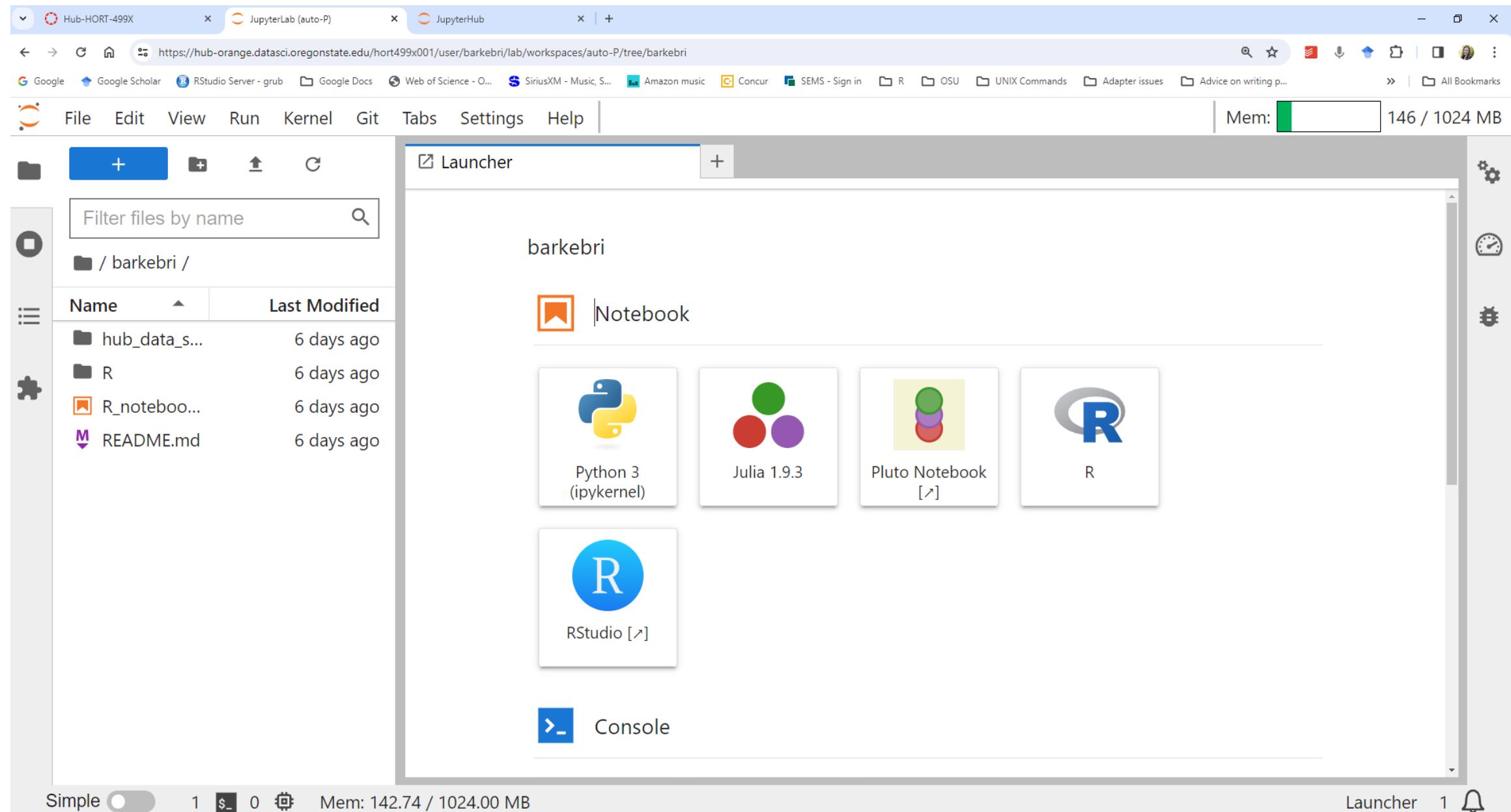
Resources

- Click Start My Server



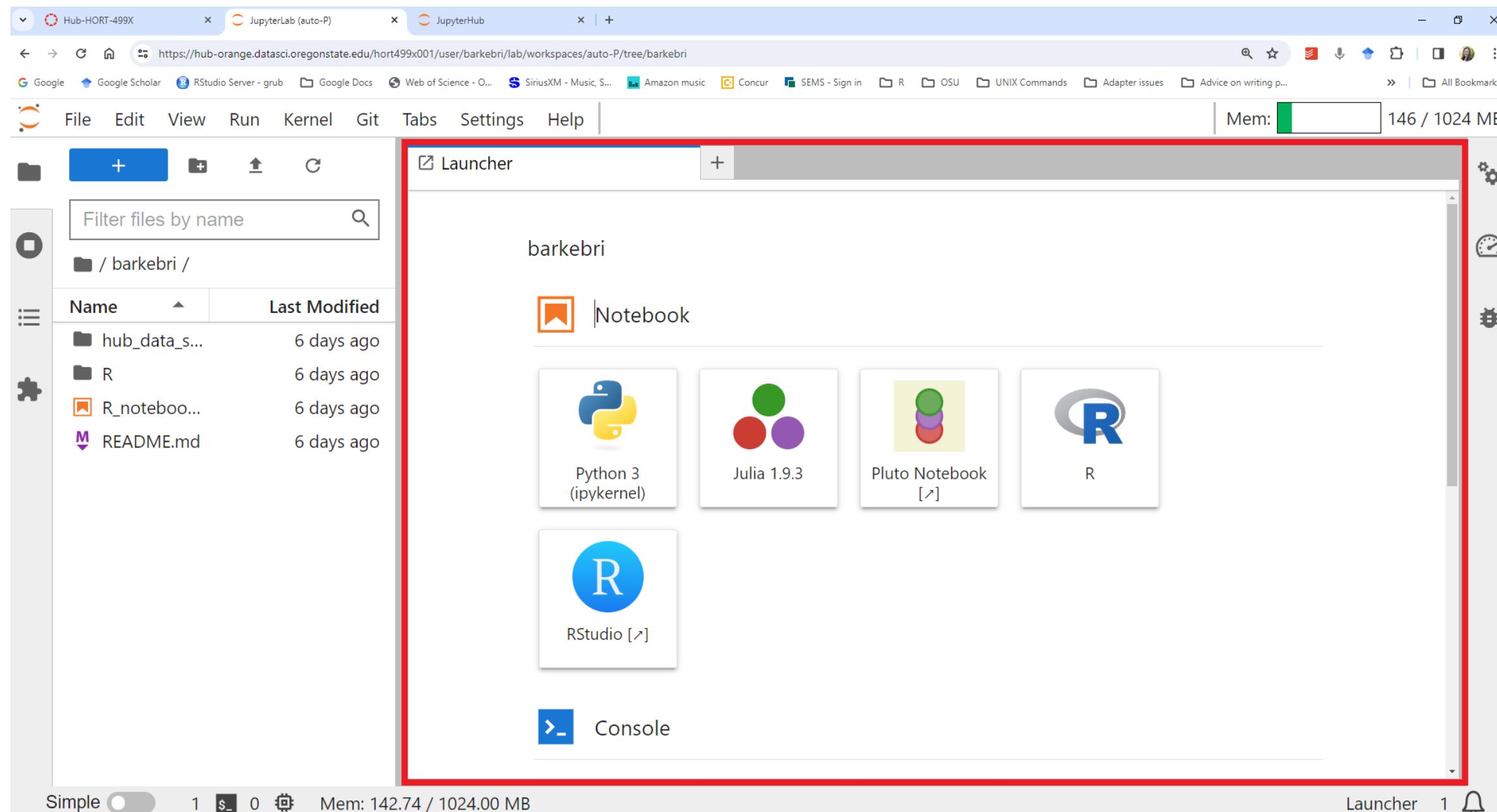
# Jupyter Hub

- You should see your Jupyter Hub workspace (mine is below)
- Other students cannot modify anything in your workspace



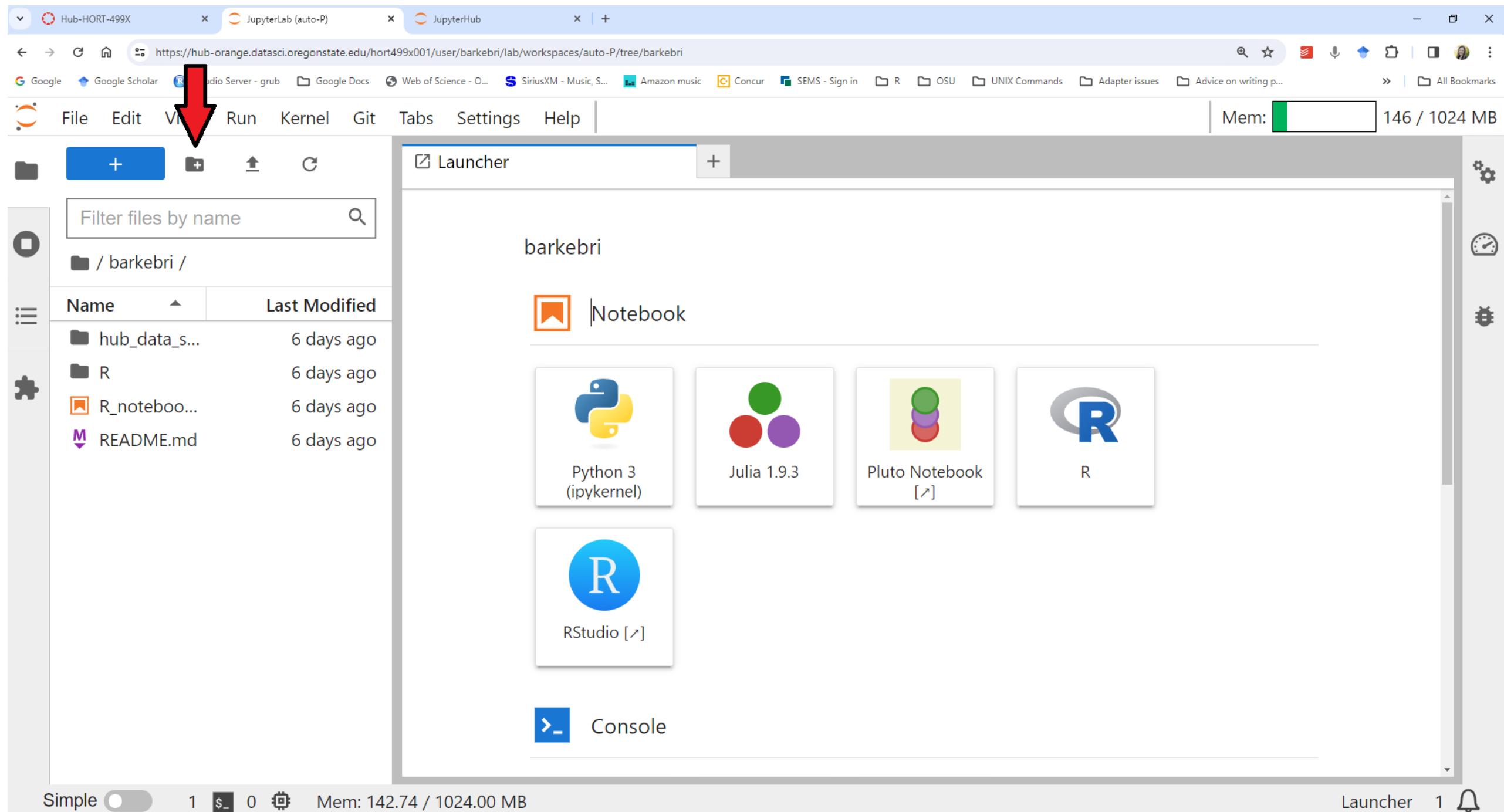
# Jupyter Notebook

- The **Launcher** tab contains your Jupyter Notebook for this course
- For this course, we will only be utilizing RStudio



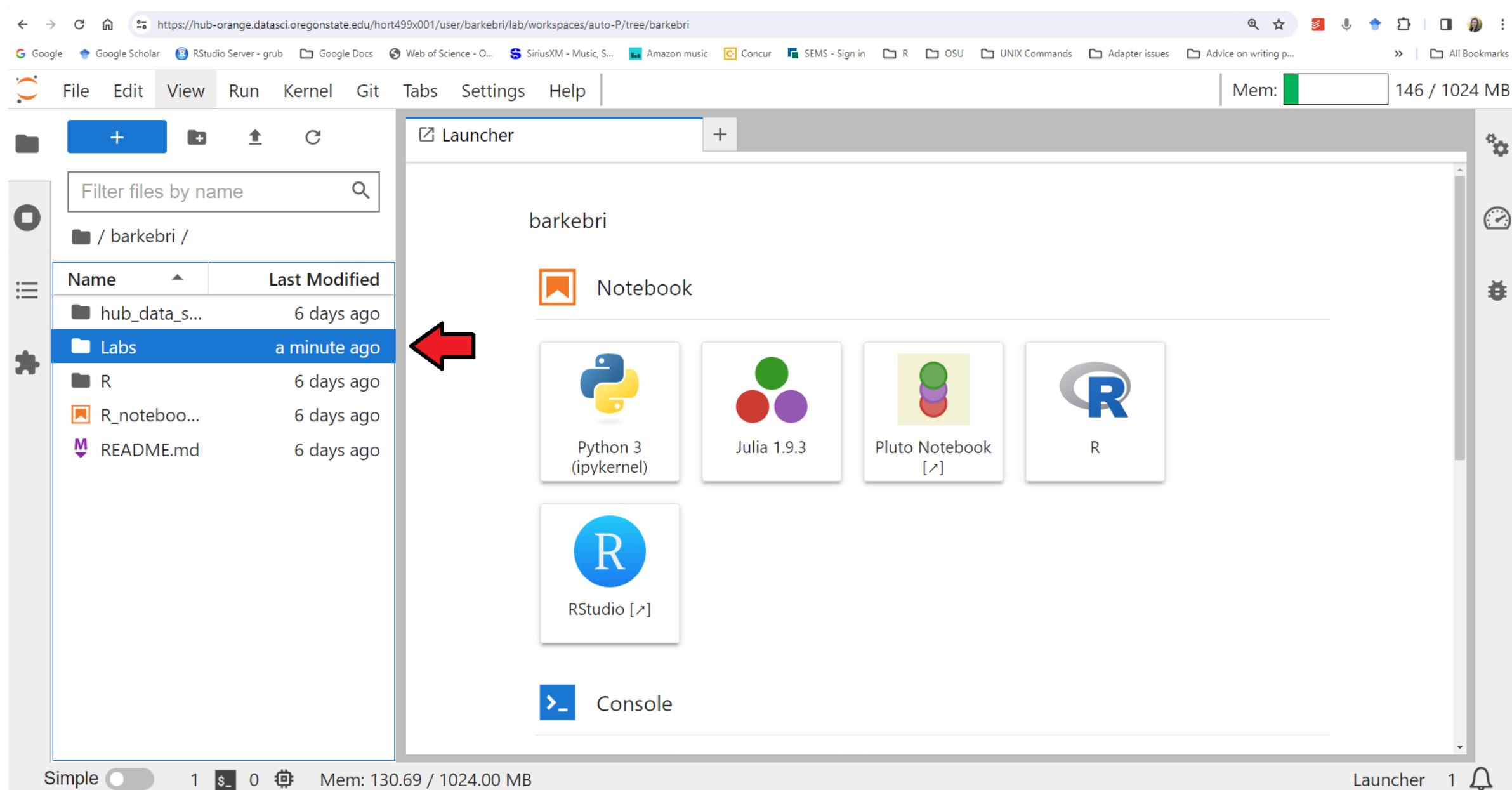
# In Jupyter Hub

- Click on the new folder icon



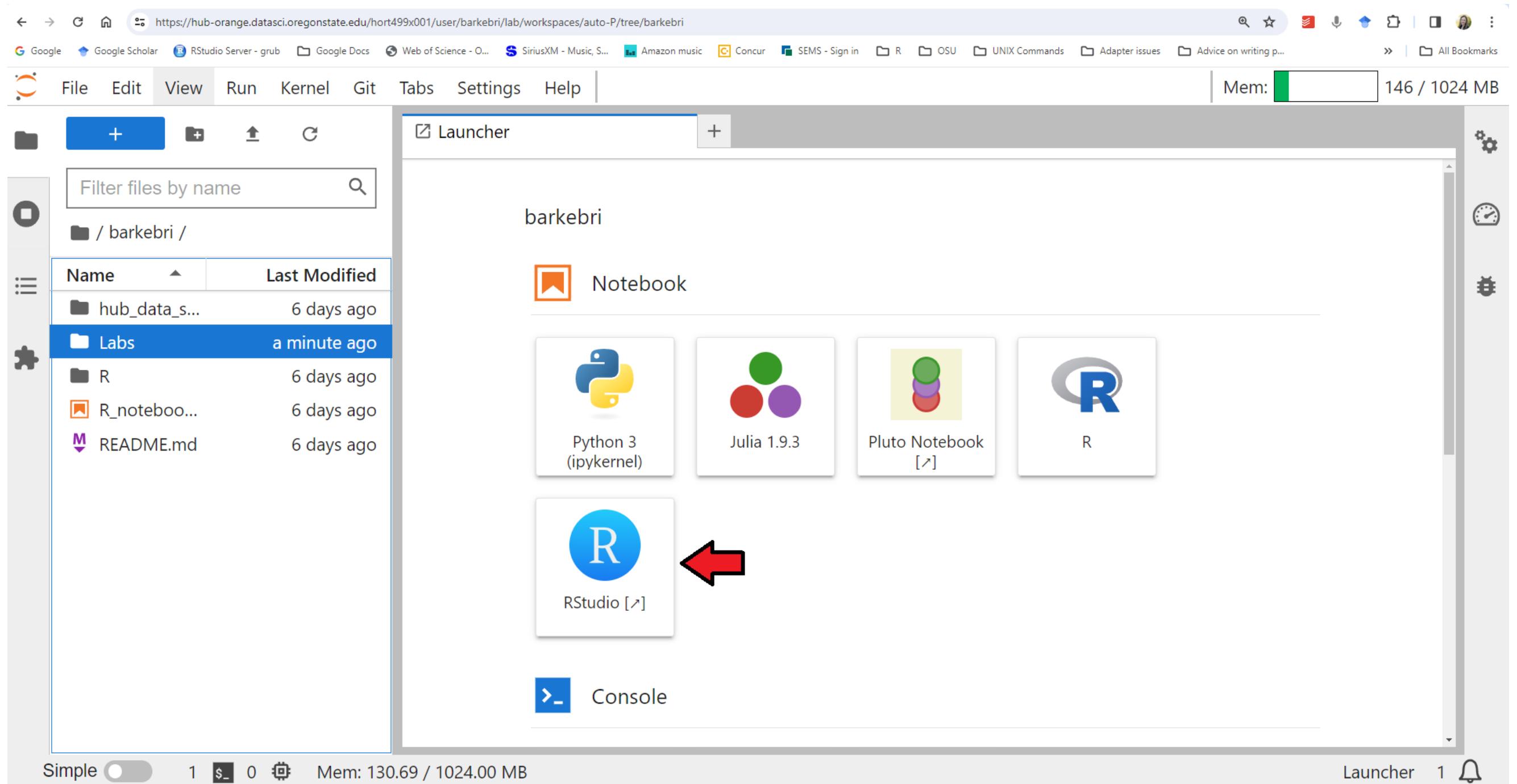
# In Jupyter Hub

- Click on the new folder icon
- Create a directory (folder) named **Labs**



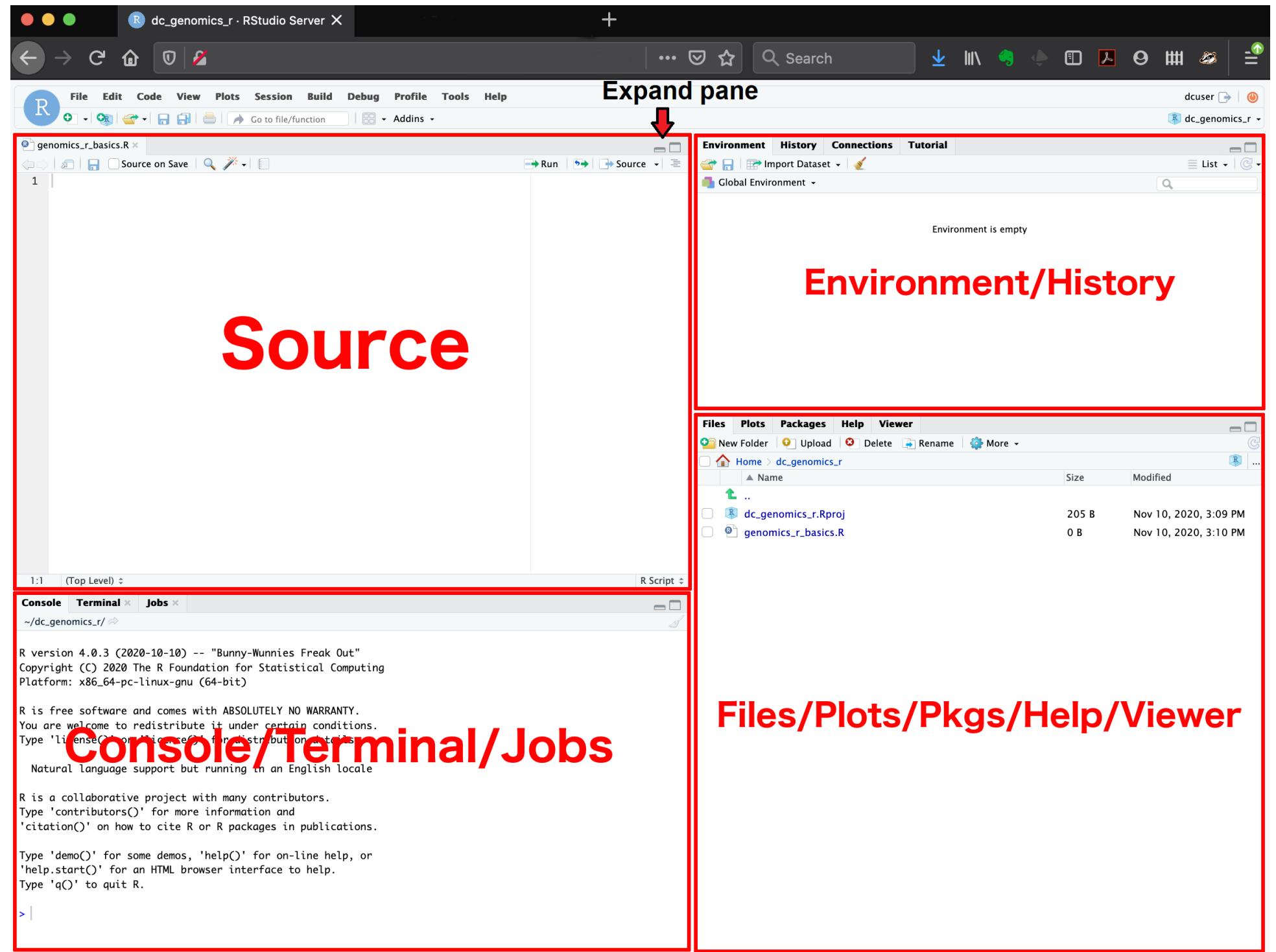
# In Jupyter Hub

- Click the RStudio icon in your Notebook (in Launcher tab)



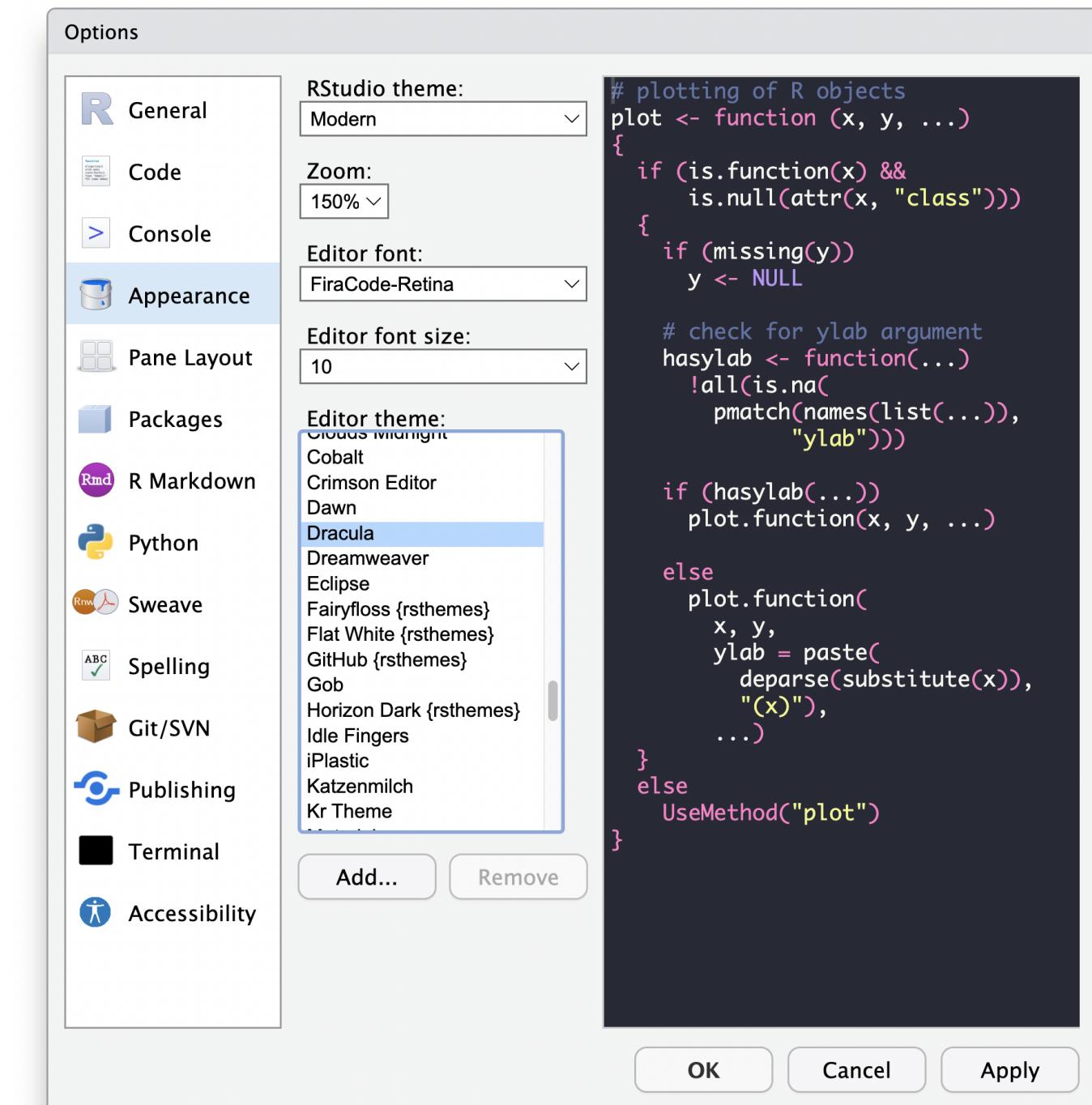
# RStudio is divided into 4 “panes”

Not seeing the entire Source pane? Click the first icon in the top upper-right corner



# Preferences

- The placement of the 4 panes and their content can be customized
- See menu, Tools -> Global Options -> Pane Layout
- Background color, font color, font size, and zoom level are in (Global Options -> Appearance)



# RStudio: Source (top-left) pane

- Your scripts or other files (e.g., R Markdown files)

The screenshot displays the RStudio interface with the following components:

- Source (Top Left):** The `flights-example.R` script is shown, containing R code to load packages, filter data, and create a boxplot. A red box highlights this pane.
- Environment (Top Right):** Shows the `daily` dataset with 365 observations and 3 variables: date, n, and wday.
- Console (Bottom Left):** Displays the R session history, including the creation of the `daily` tibble and its structure.
- Plots (Bottom Right):** A boxplot titled "Number of 2013 New York Flights Each Weekday" showing flight counts by weekday (Sun to Sat).

# RStudio: Environment/History (top-right) pane

- Objects in your working space (Environment) and your command history (History)

The screenshot shows the RStudio interface with several panes visible. The top-left pane contains an R script named "flights-example.R". The script loads packages (nycflights13, lubridate, dplyr, ggplot2), processes the flights dataset by date and weekday, and creates a boxplot titled "Number of 2013 New York Flights Each Weekday". The bottom-left pane is the Console, displaying the R code and its output, including the structure of the "daily" tibble and the resulting boxplot. The top-right pane, which is the focus of the question, is the Environment pane. It lists objects in the Global Environment, specifically the "daily" tibble, which is described as having 365 observations and 3 variables: date (Date), n (int), and wday (Ord.factor). The bottom-right pane is the Plots pane, showing the generated boxplot.

```
library(nycflights13) ## package containing flights dataset
library(lubridate)
library(dplyr)
library(ggplot2)

head(flights, n = 3)
daily <- flights %>%
  mutate(date = make_date(year, month, day)) %>%
  count(date) %>%
  mutate(wday = wday(date, label = TRUE))
head(daily, n = 3)
ggplot(daily, aes(wday, n)) +
  geom_boxplot(outlier.colour = "hotpink") +
  labs(x = "Weekday", y = "Flights",
       subtitle = "Number of 2013 New York Flights Each Weekday")

# A tibble: 3 x 19
# ... with 9 more variables: flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>,
#   distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
> daily <- flights %>%
+   mutate(date = make_date(year, month, day)) %>%
+   count(date) %>%
+   mutate(wday = wday(date, label = TRUE))
> head(daily, n = 3)
# A tibble: 3 x 3
# ... with 2 variables: date <date>, n <int>, wday <ord>
1 2013-01-01 842 Tue
2 2013-01-02 943 Wed
3 2013-01-03 914 Thu
> ggplot(daily, aes(wday, n)) +
+   geom_boxplot(outlier.colour = "hotpink") +
+   labs(x = "Weekday", y = "Flights",
+        subtitle = "Number of 2013 New York Flights Each Weekday")
>
```

Environment | History | Connections | Tutorial

Global Environment

Data

daily 365 obs. of 3 variables

\$ date: Date[1:365], format: "2013-01-01" "2013-01-02" ...  
\$ n : int [1:365] 842 943 914 915 720 832 933 899 902...  
\$ wday: Ord.factor w/ 7 levels "Sun" <"Mon" <"Tue" <...: 3 ...

Number of 2013 New York Flights Each Weekday

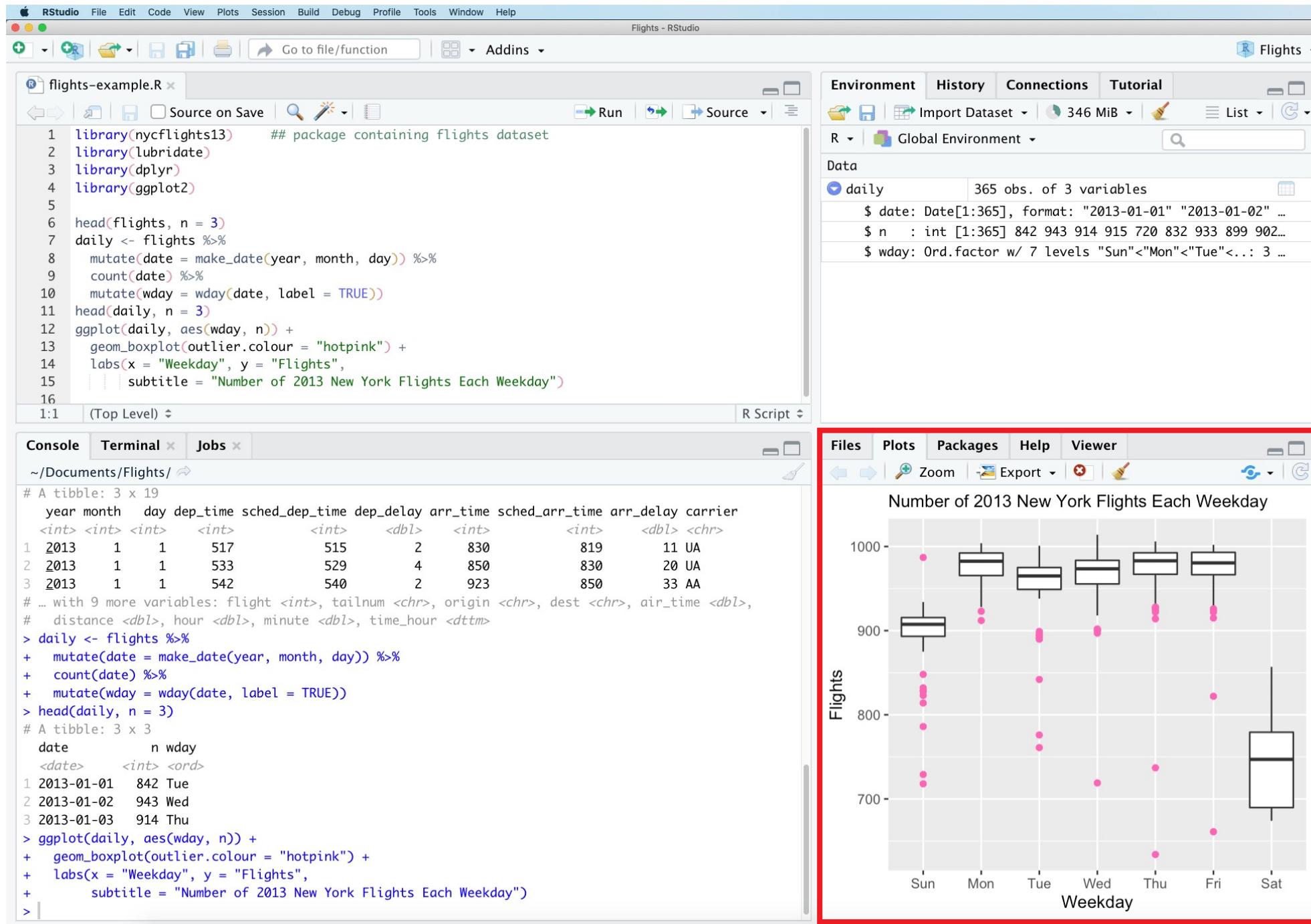
Flights

Weekday

Sun Mon Tue Wed Thu Fri Sat

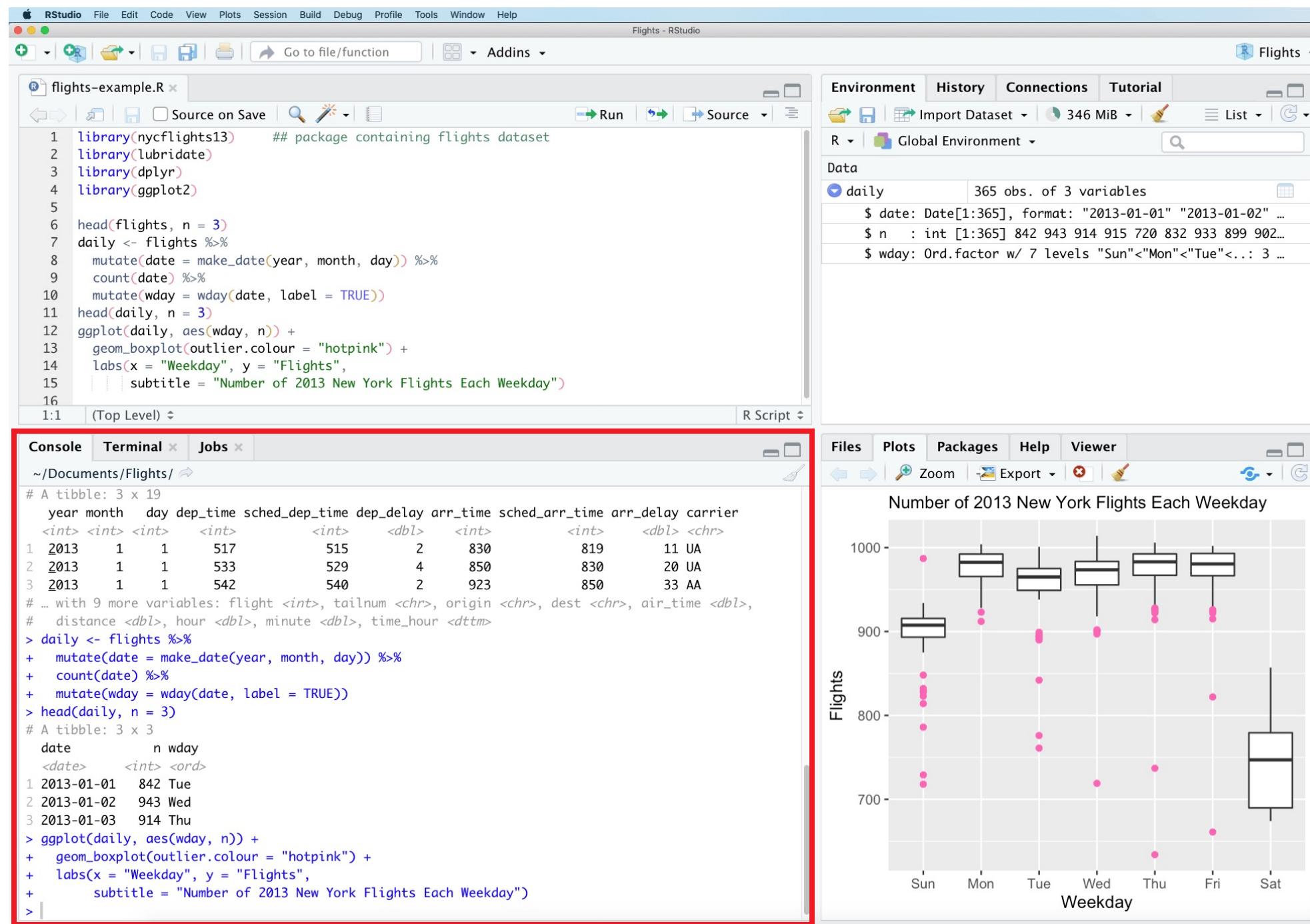
# RStudio: Files/Plots/Packages/etc. pane (bottom-right)

- Files, plots, etc. in your directories (in Jupyter Hub)



# RStudio: R Console (bottom-left)

- Provides an area to interactively execute code (but won't be saved!)



The screenshot shows the RStudio interface with the following components:

- Top Bar:** RStudio, File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Window, Help.
- Left Sidebar:** Addins dropdown.
- Script Editor:** flights-example.R file open, showing R code for data manipulation and visualization.
- Environment Tab:** Shows the 'daily' dataset with 365 observations and 3 variables: date, n, and wday.
- Console Tab:** Contains the R command history and output. A red box highlights the console area.
- Output:** Displays the results of the R code execution, including the structure of the 'daily' dataset and a boxplot titled "Number of 2013 New York Flights Each Weekday".

Console output (highlighted by red box):

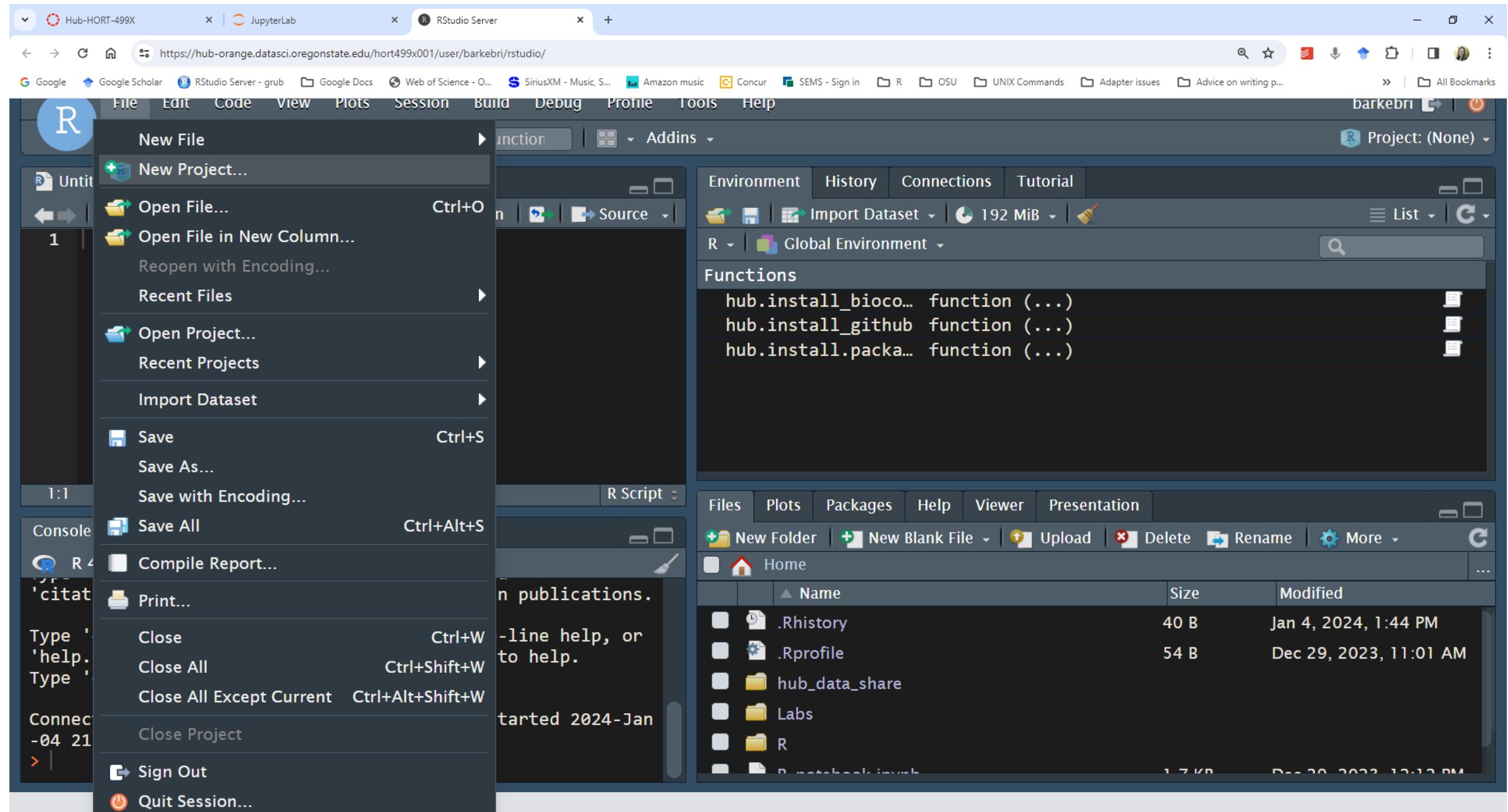
```
# A tibble: 3 x 19
  year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay carrier
  <int> <int> <int>     <int>      <dbl>    <int>     <dbl>      <int>      <dbl> <chr>
1 2013     1     1     517       515        2     830       819        11  UA
2 2013     1     1     533       529        4     850       830        20  UA
3 2013     1     1     542       540        2     923       850        33  AA
# ... with 9 more variables: flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>,
# distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
> daily <- flights %>%
+   mutate(date = make_date(year, month, day)) %>%
+   count(date) %>%
+   mutate(wday = wday(date, label = TRUE))
> head(daily, n = 3)
# A tibble: 3 x 3
  date       n wday
  <date> <int> <ord>
1 2013-01-01  842 Tue
2 2013-01-02  943 Wed
3 2013-01-03  914 Thu
> ggplot(daily, aes(wday, n)) +
+   geom_boxplot(outlier.colour = "hotpink") +
+   labs(x = "Weekday", y = "Flights",
+        subtitle = "Number of 2013 New York Flights Each Weekday")
>
```

# Learning objectives

- Explain what R and RStudio are, what they are used for, and how they relate to each other
- Describe the purpose of the Script, Console, Environment, and Plots panes in RStudio
- **Create an R project and understand the purpose of the working directory**
- Copy a folder of files on Jupyter Hub and open them in RStudio

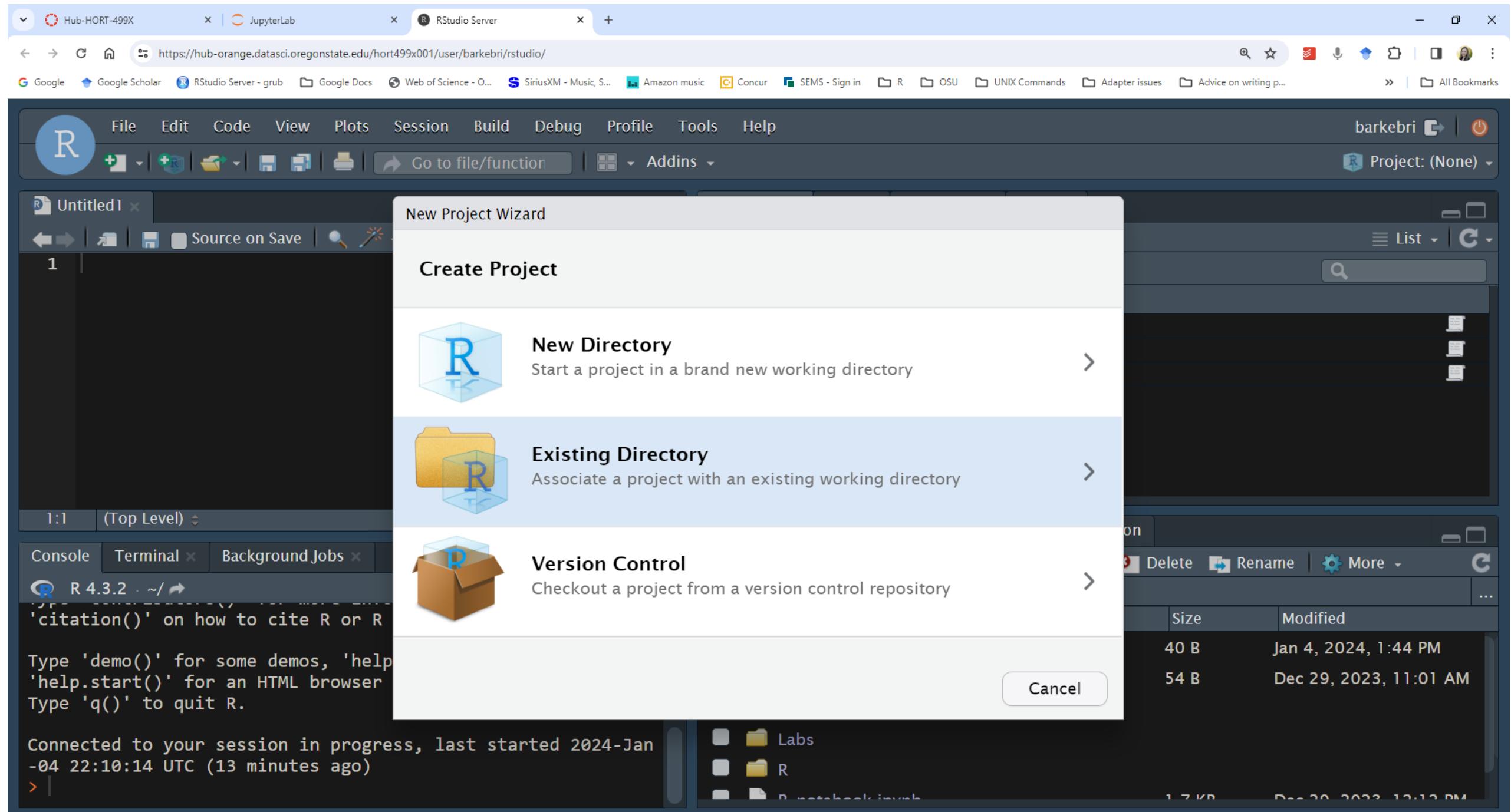
# Create an R project for this course

## 1. Under the File menu, click on New Project



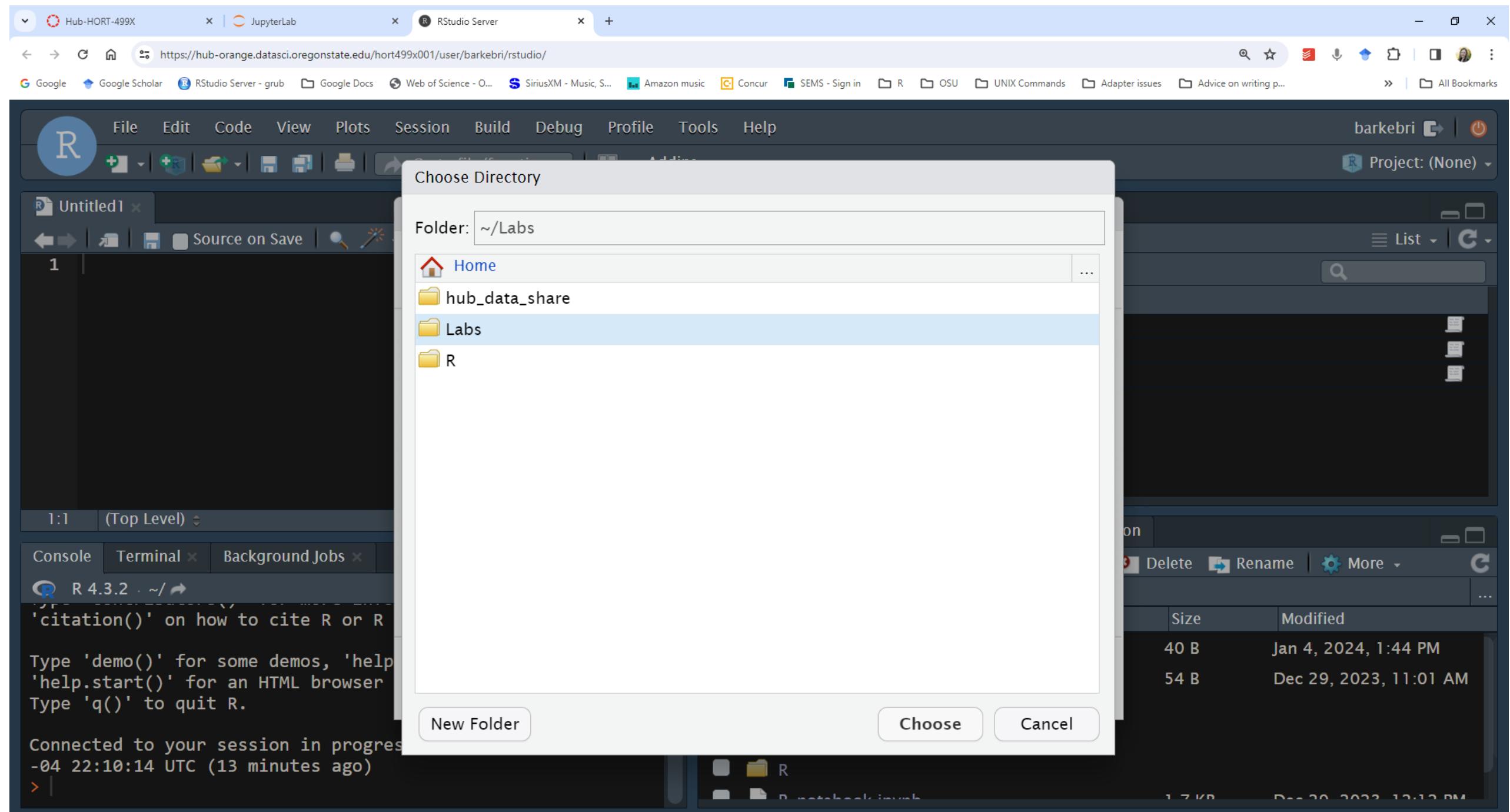
# Create an R project for this course

## 2. Choose Existing Directory, then New Project



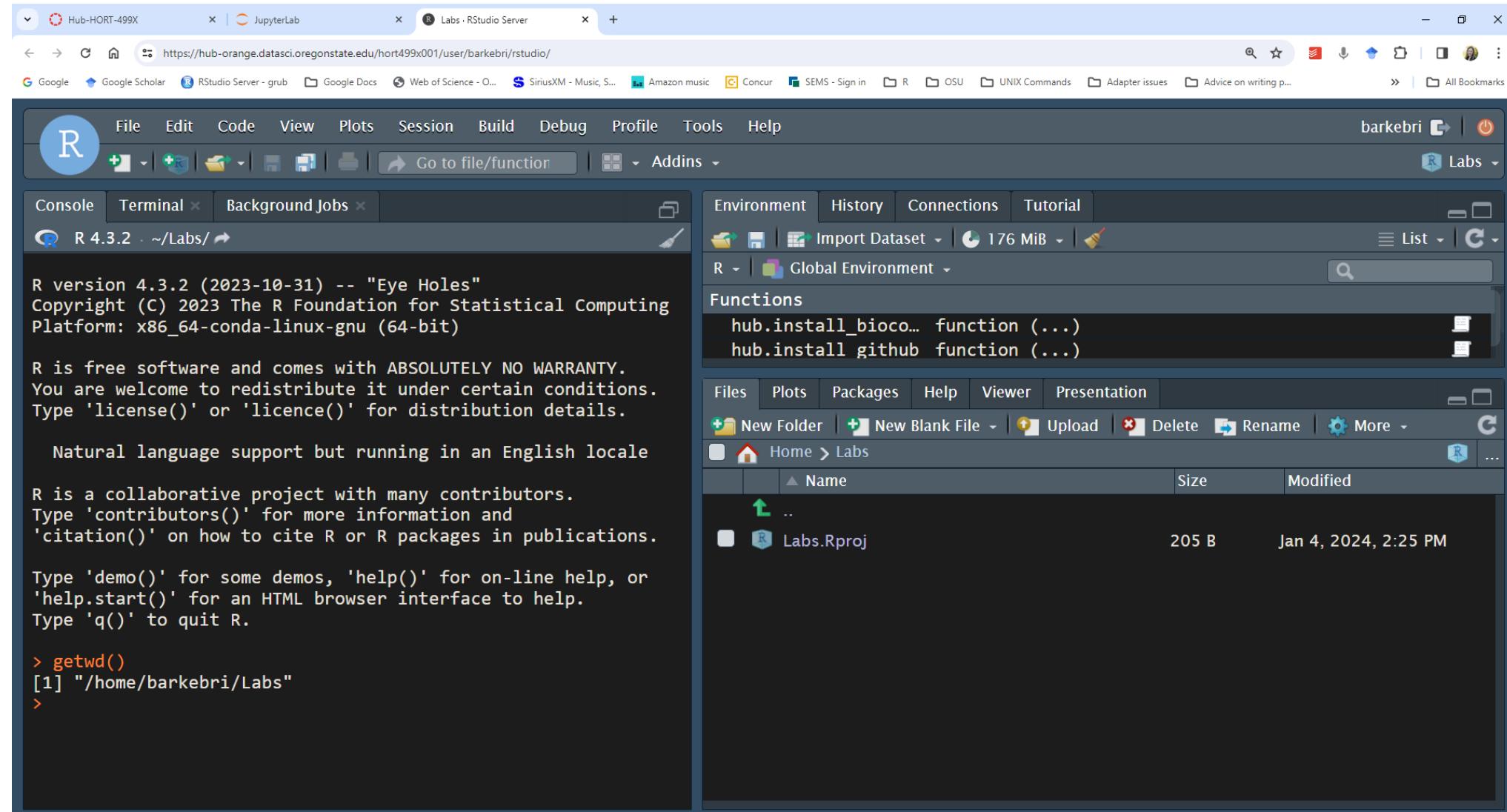
# Create an R project for this course

## 3. Select the **Labs** folder that you recently created



# The working directory

- The **Labs** folder, where you put your R project, will be your **working directory**
- To see this, type in `getwd()` in the R console
- It should return `/home/username/Labs` (username is your ONID)



A screenshot of an RStudio interface running in a web browser. The browser tab is titled "Labs - RStudio Server". The RStudio window shows the following:

- Console:** Displays the R startup message and the command `> getwd()` followed by the output `[1] "/home/barkebri/Labs"`.
- Environment:** Shows the Global Environment with two functions: `hub.install_bioco...` and `hub.install_github`.
- Files:** Shows a file named `Labs.Rproj` in the "Labs" directory.

# Why are we doing this?

- You must open your R project prior to opening a lab assignment (more on this later)
- All code within the `Labs` directory can use *relative paths* to files that indicate where inside the project a file is located
- The use of projects and relative paths makes sharing code with others easier
- It's also less laborious than typing out *absolute paths* (i.e., the full path to your directory)

# Learning objectives

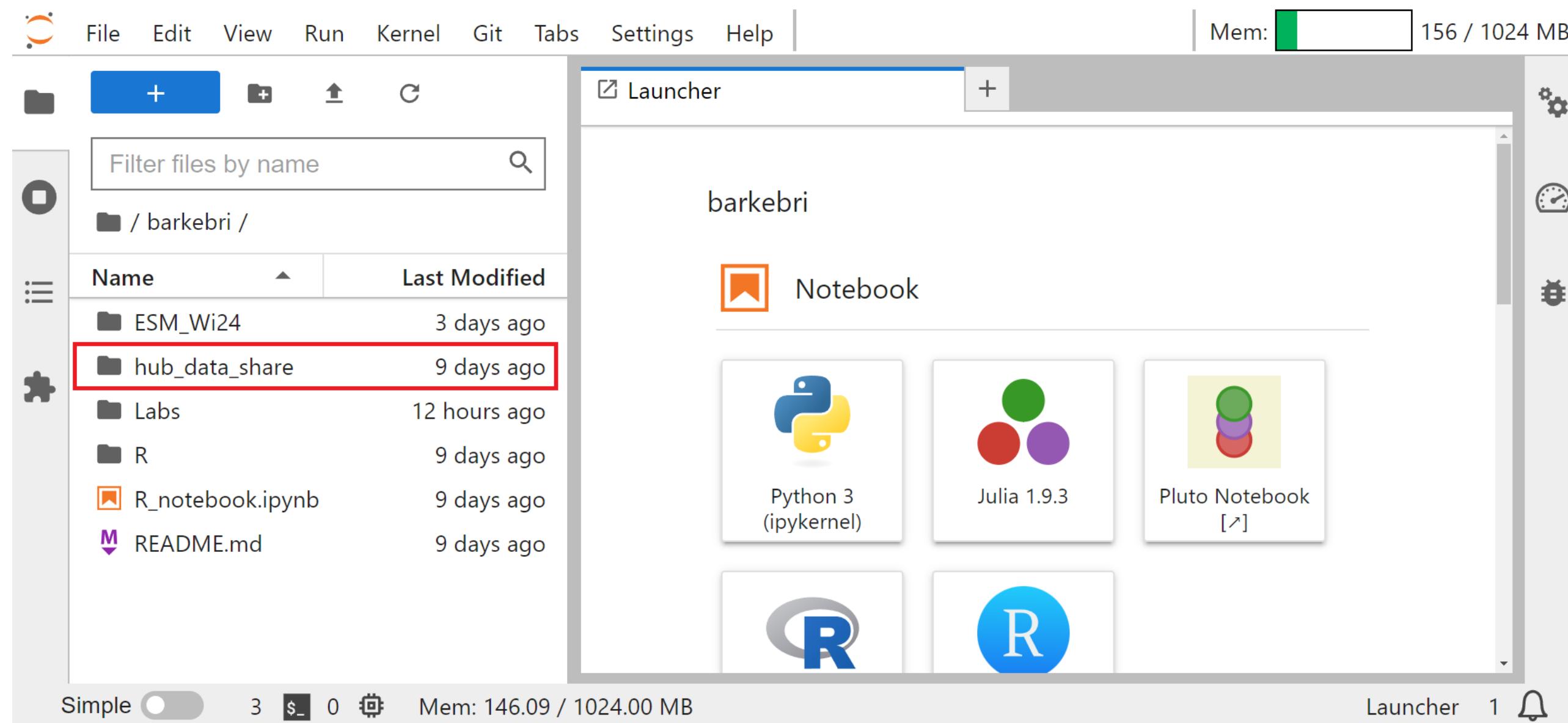
- Explain what R and RStudio are, what they are used for, and how they relate to each other
- Describe the purpose of the Script, Console, Environment, and Plots panes in RStudio
- Create an R project and understand the purpose of the working directory
- **Copy a folder of files on Jupyter Hub and open them in RStudio**

# Copying tutorials/lab assignments

- NOTE: Instructions presented here are available in Canvas
- Go to [Modules](#) -> [Jupyter Hub](#) -> [Copying\\_files.pdf](#)

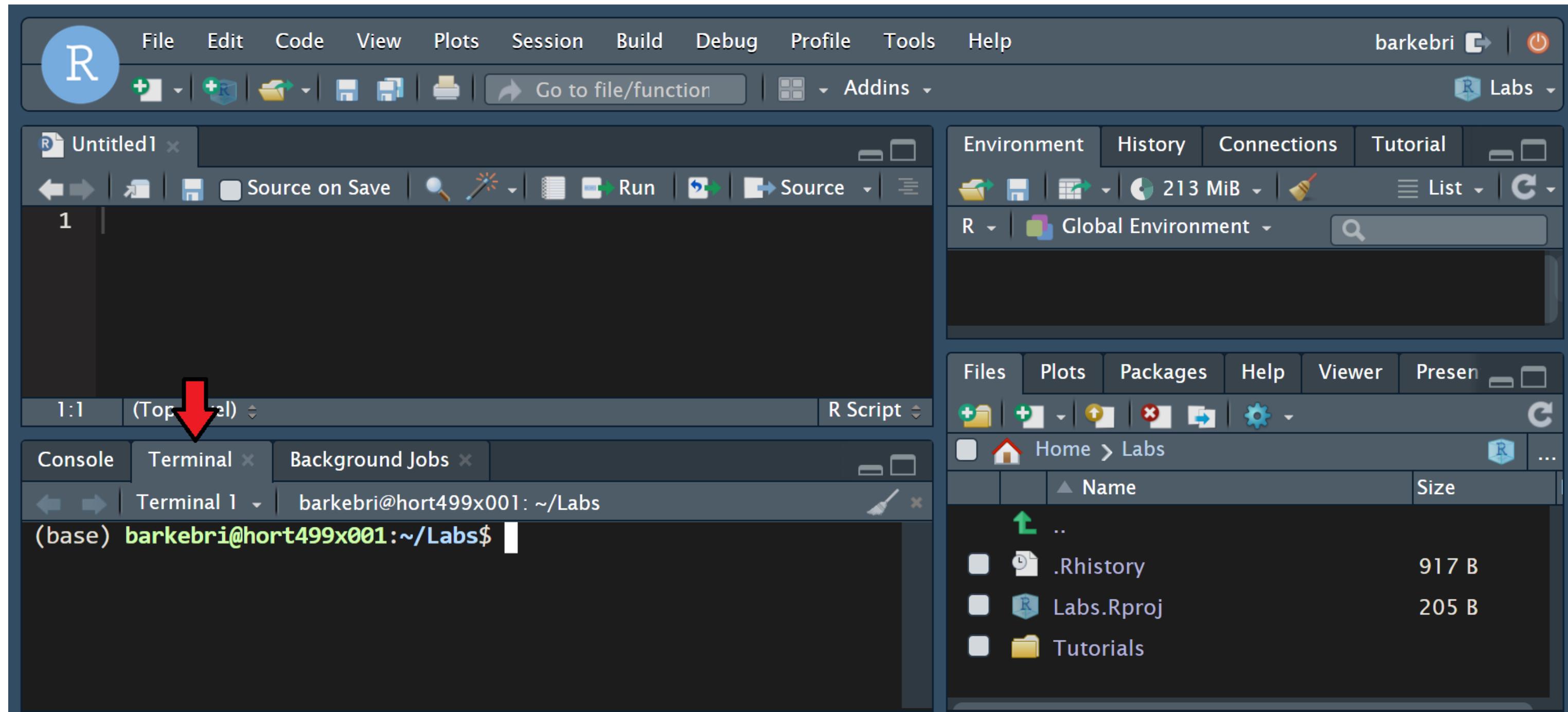
# Copying tutorials/lab assignments

- All files that involved the use of R will be in `hub_data_share` in Jupyter Hub
- You will need to copy folders with these files to your own account



# Method 1: copy folder using Terminal in RStudio

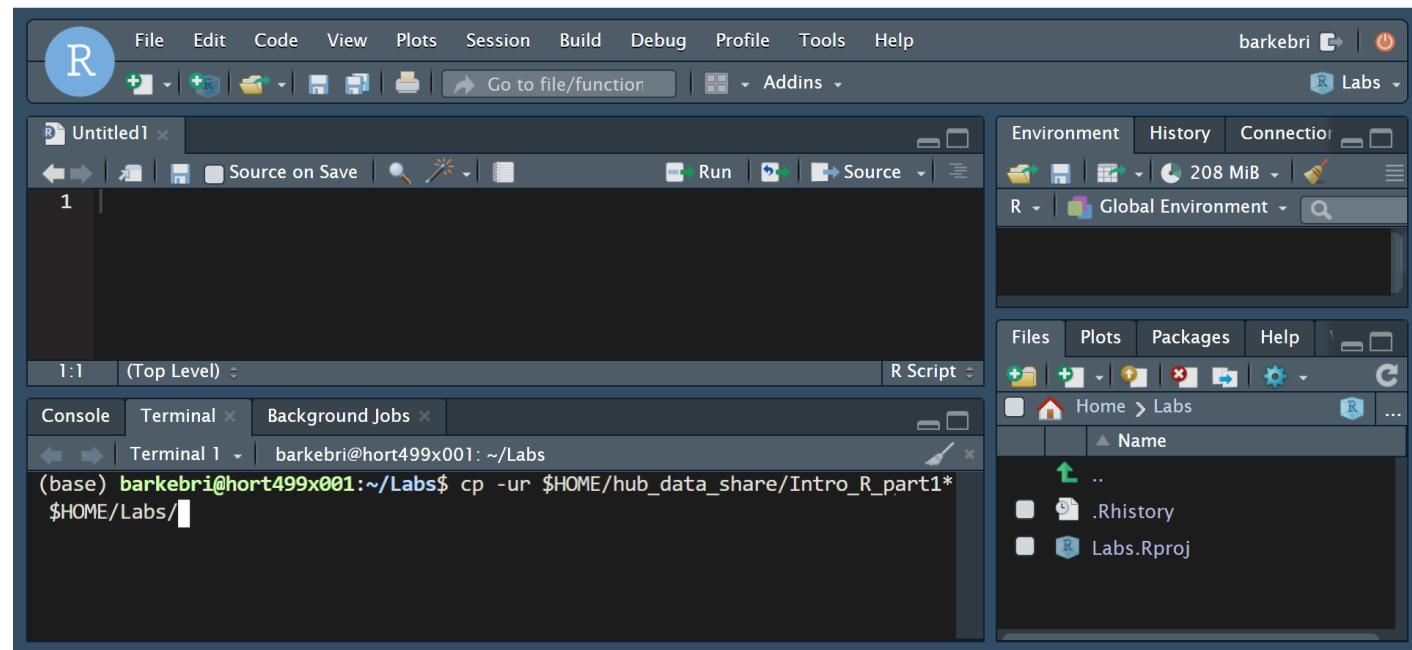
Open up RStudio and click on the Terminal tab in the Console pane (bottom-left)



# Method 1: copy folder using Terminal in RStudio

Type in the following command in the terminal:

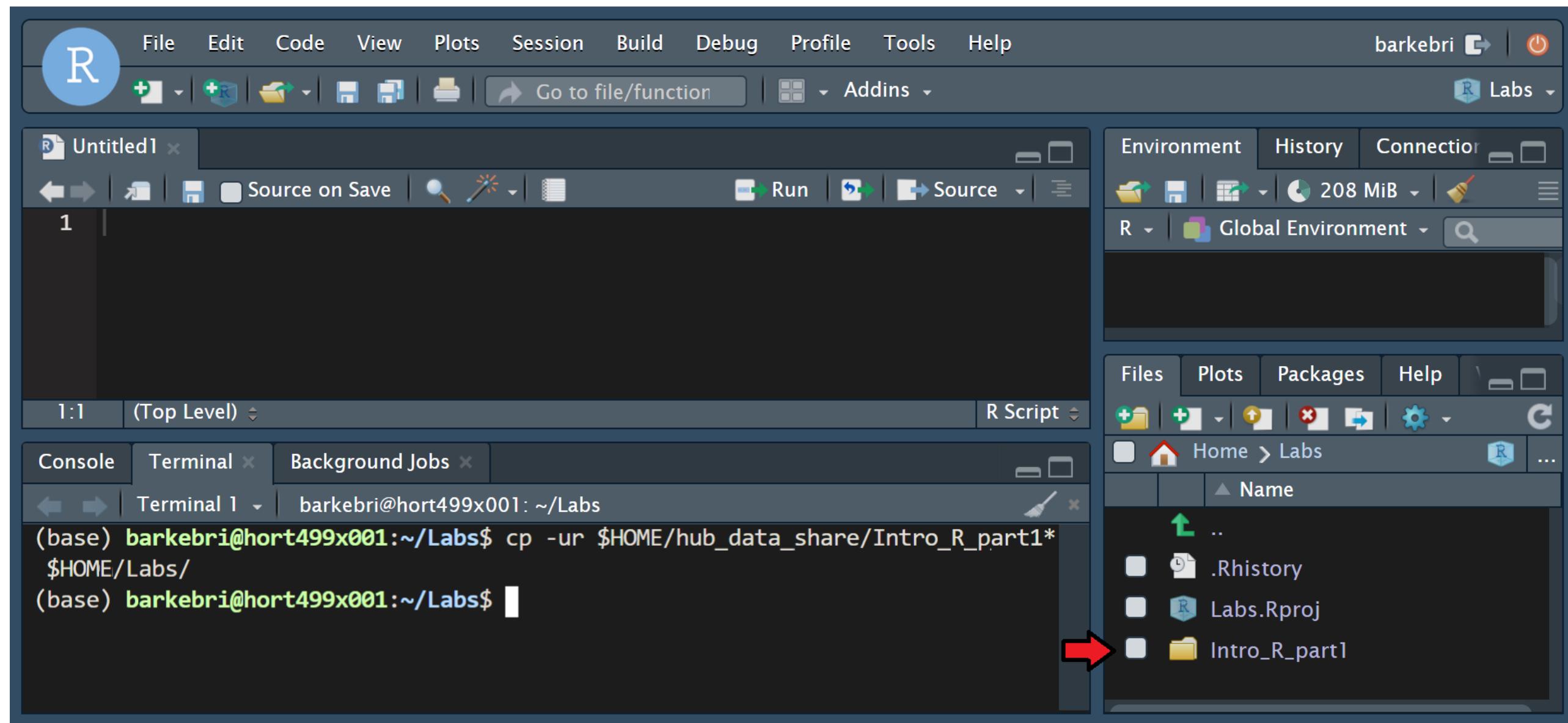
```
cp -ur $HOME/hub_data_share/Intro_to_R_part1* $HOME/Labs/
```



- `cp` copies files and directories
- `-u` option indicates copying only new and missing files to the destination
- `-r` option indicates drilling down through the directory structure
- `$HOME` is the path to your home directory, which is `/home/your_onid`

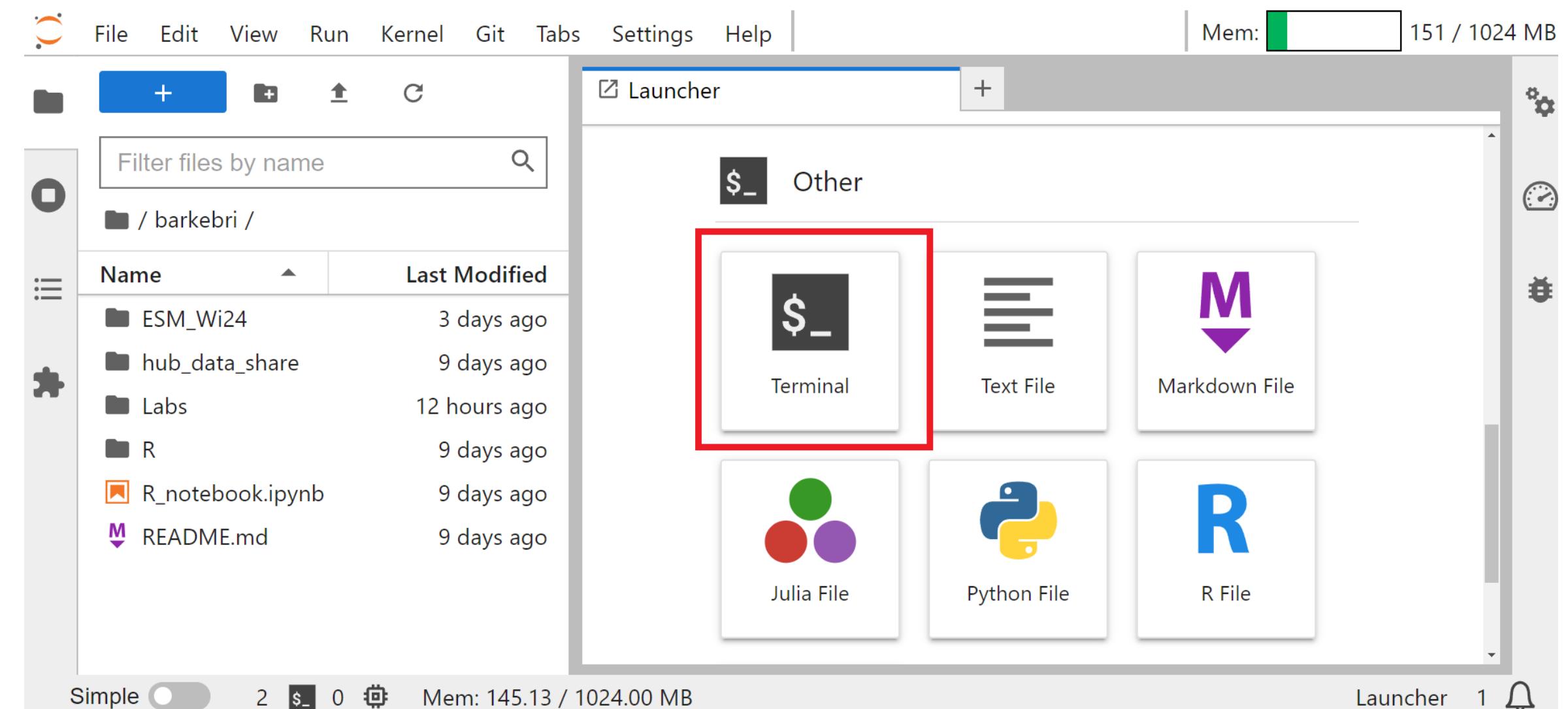
# Method 1: copy folder using Terminal in RStudio

- If it worked correctly, you should see the `Intro_R_Part1` folder
- (In the `Files` tab of the `Files/Plots/Packages/etc.` pane)



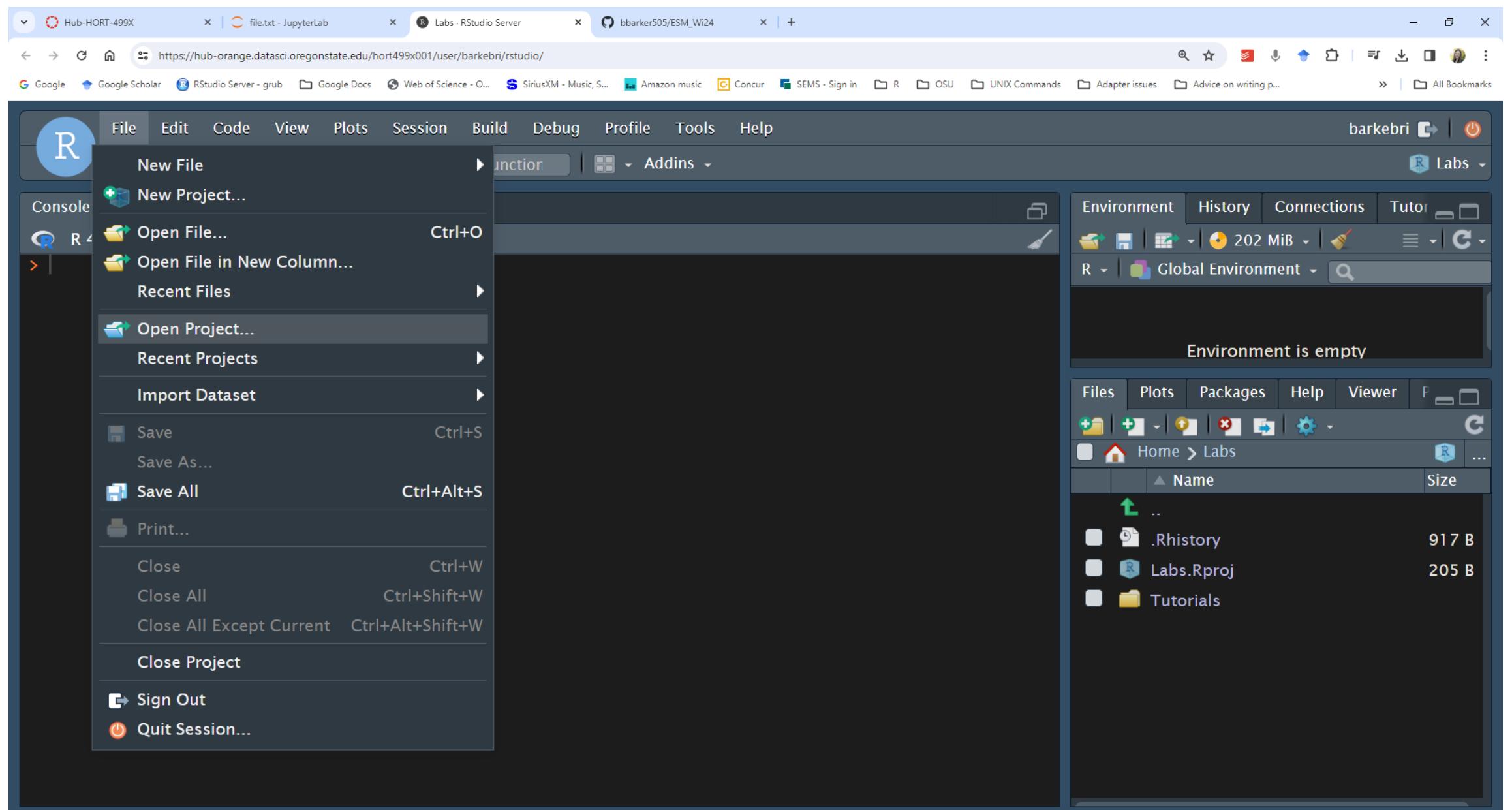
# Method 2: Using the Terminal in Jupyter Hub

1. In the [Launcher](#) tab, scroll down to the [Other](#) section
2. Click on [Terminal](#) (black box with money symbol) to start a session
3. Type `cp -ur $HOME/hub_data_share/Intro_to_R_part1* $HOME/Labs/`



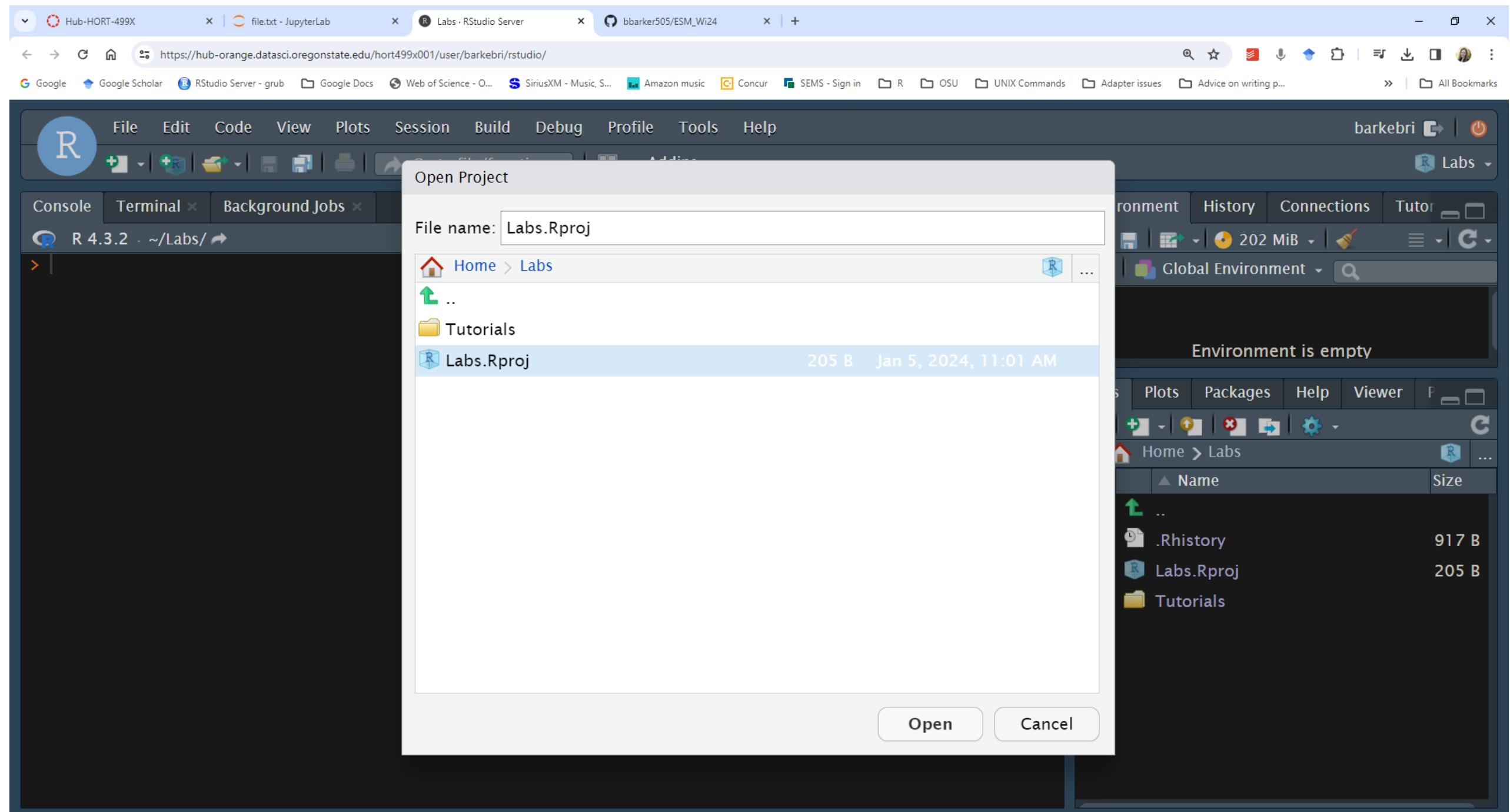
# Open your project and load some files

- We will be using the **Labs** project (`Labs.proj`) for all tutorials and lab assignments
- In RStudio, if you're project isn't already open, click on **File** -> **Open Project**



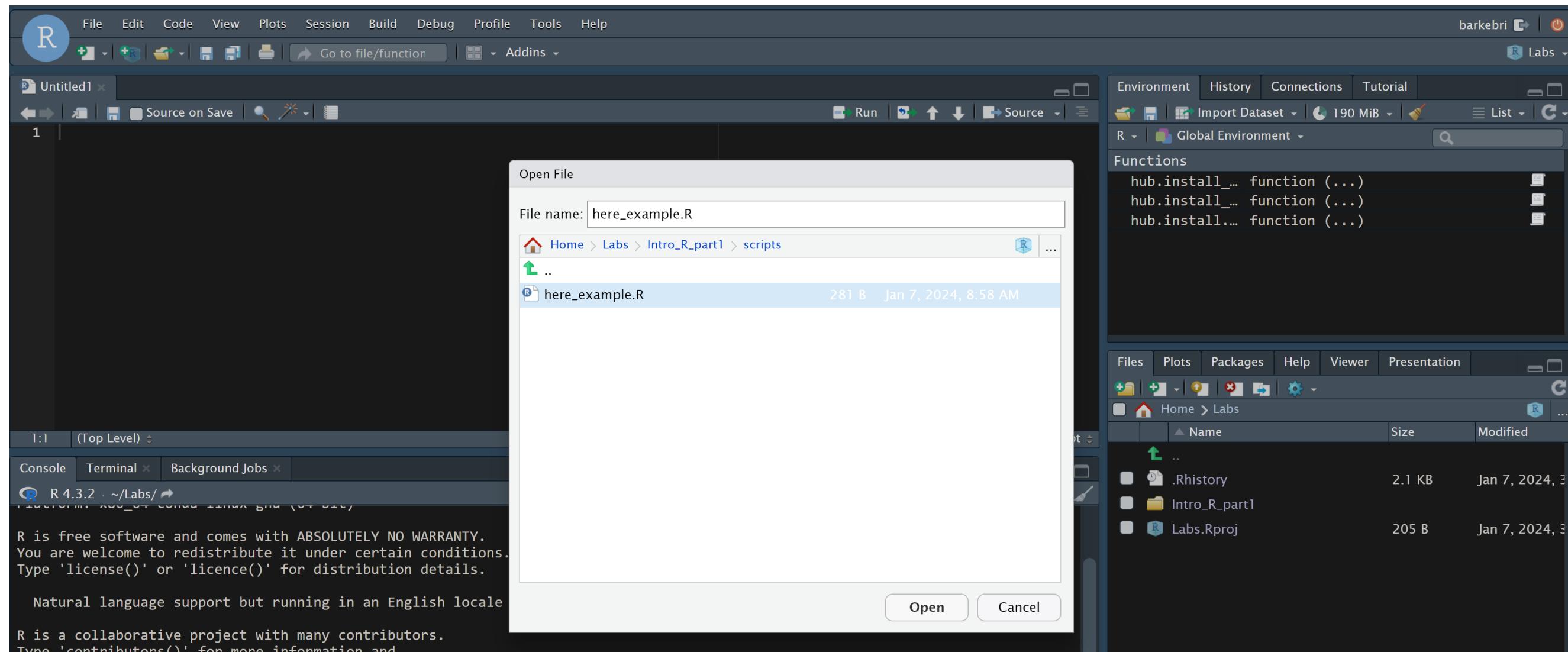
# Open your project and load some files

- Open your project (`Labs.proj`)



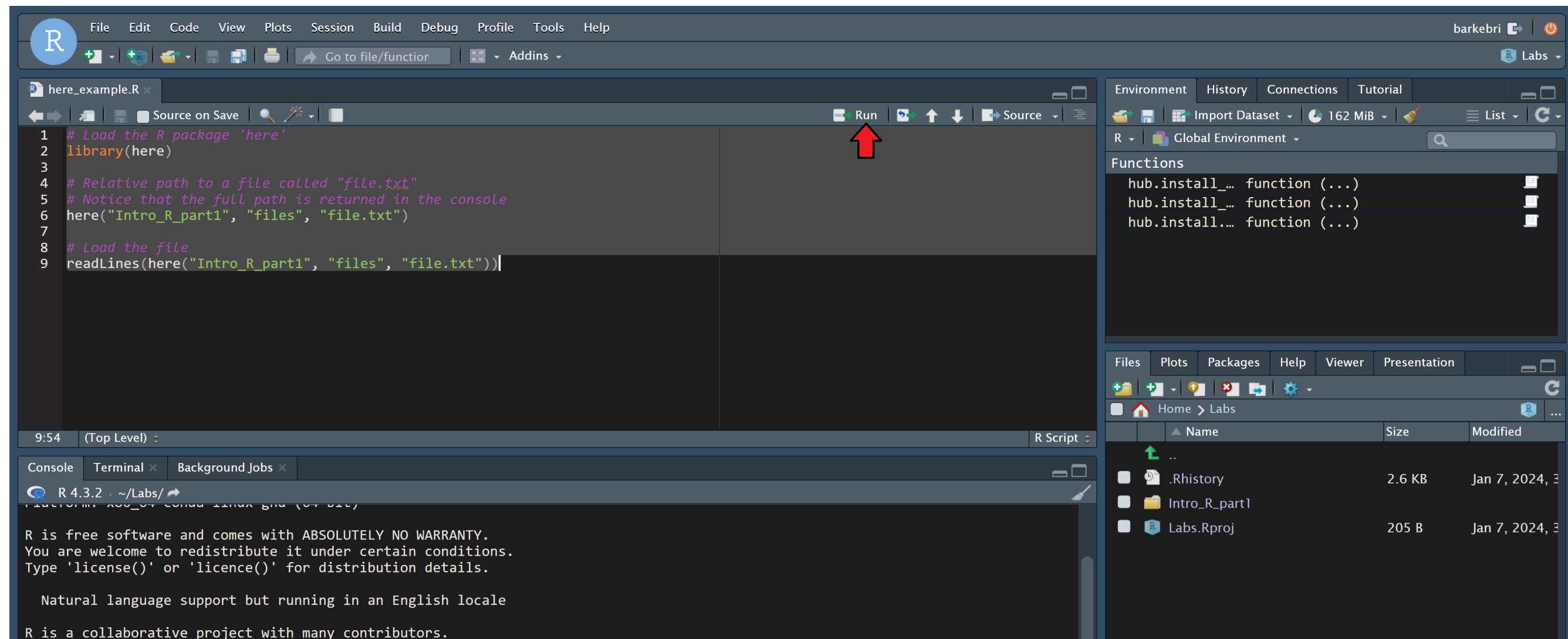
# Open an R script

- Click on [Open File](#)
- Navigate to [Home -> Labs -> Tutorials -> Intro\\_R\\_part1 -> scripts](#)
- Select the [here\\_example.R](#) script



# The here package

- For this class, we will use the `here` package to use project-relative paths
- Select all text in the script by pressing `Ctrl + A`
- Click on the `Run` button on the Source pane



# Output in the Console pane

- Line 6 of the code was the project-relative path to `file.txt`
- Line 9 read the lines of `file.txt` using `readLines()` (see result in console)

The screenshot shows the RStudio interface with the following components:

- Code Editor:** Displays the script `here_example.R` containing R code.
- Console:** Shows the R session output. It includes the R startup message, the loading of the `here` package, the determination of the project root, the creation of a relative file path, and the execution of `readLines` which prints the file content "Hi, welcome to Ecological Systems Modeling!".
- Environment:** Shows the global environment with functions like `hub.install...`.
- Files:** Shows the project structure with files `.Rhistory`, `Intro_R_part1`, and `Labs.Rproj`.

