Workshop Series:
Reusable Research Data Made Shiny

Ontario Dairy Research Centre | Online
February 21st - 24th, 2023

University of Guelph
Improve Life.

Compute Ontario

ADC

First things first

House-keeping rules

- Webcam
  - It's up to you ;)
- Microphone
  - OFF, unless when needed
- Questions?
  - Yes, please! Just raise your hand

Day 3 Overview

Session		Topic	Speaker (s)
Morning	1	Quick Posit Cloud Recap Introduction to the R Programming Language and R Studio IDE	Busayo Kodaolu Marjan Asgari
	2	Introduction to the Tidyverse with Hands-on exercises Tidy verbs: filter(), select(), arrange(), summarize()	Lucas Alcantara
Afternoon	3	Introduction to the Tidyverse with Hands-on exercises Tidy verbs: mutate(), group_by(), pivot_longer(), pivot_wider()	Lucas Alcantara
	4	Introduction to R Shiny	Lucas Alcantara

Quick Posit Cloud Recap (RStudio Cloud)

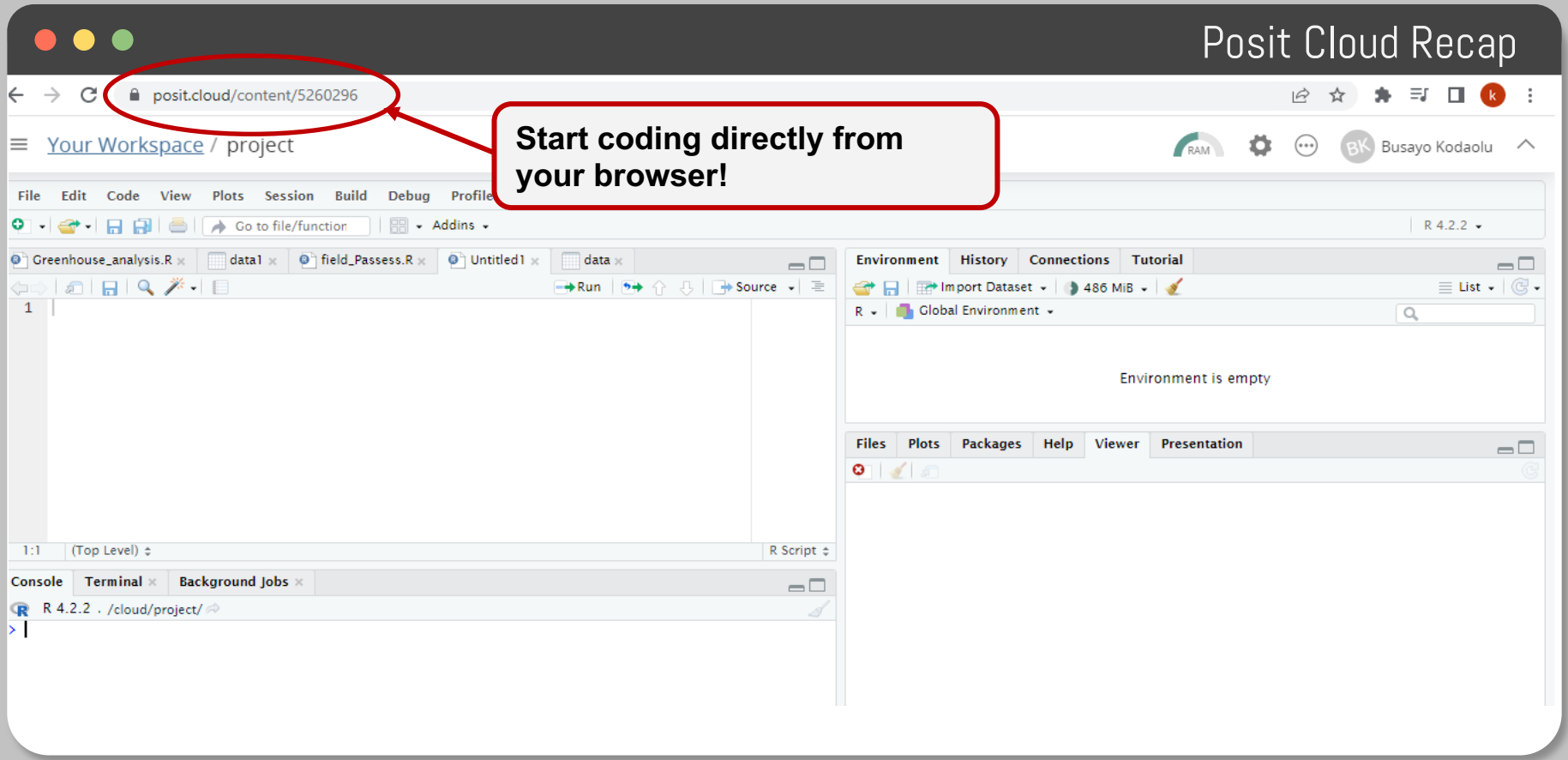
Busayo Kadaolu

What is Posit cloud?

R Studio IDE in the cloud!

Using server hosted on the internet rather than a local computer or server

Posit Cloud Recap



Posit Cloud set-up

Work along (if you haven't activated RStudio)

Remember to use your sticky notes to indicate your progress/you need help

Welcome email

You should have received a welcome email

Hello,

We are excited to have you for our 4-day Reusable Research Data Made
help you make your research data reusable and shiny!

Quick notes about the workshop:

The workshop will run from Feb 21st to Feb 24th from 9:00 AM EDT to 4:

If you registered to attend **in-person**:

- The Ontario Dairy Research Centre is located at: 6185 2nd Line Ea
- Breakfast will be served daily at the Research Centre at 8:30 AM.

If you registered to attend **on-line**:

- Click here to attend the meeting on MS Teams
- Note that the first session of the day starts at 9:00 AM, but feel fr

Click here to get started on the workshop, learn where to find the mater

Have a great long weekend and see you at the workshop!

- Workshop Organizing Committee

Welcome email

You should have received a welcome email

Hello,

We are excited to have you for our 4-day Reusable Research Data Made
help you make your research data reusable and shiny!

Quick notes about the workshop:

The workshop will run from Feb 21st to Feb 24th from 9:00 AM EDT to 4:

If you registered to attend **in-person**:

- The Ontario Dairy Research Centre is located at: 6185 2nd Line Ea
- Breakfast will be served daily at the Research Centre at 8:30 AM.

If you registered to attend **on-line**:

- Click here to attend the meeting on MS Teams
- Note that the first session of the day starts at 9:00 AM, but feel fr

Click here to get started on the workshop, learn where to find the mater

Have a great long weekend and see you at the workshop!

- Workshop Organizing Committee

Workshop GitHub page

Welcome email

You should have received a welcome email

Hello,

We are excited to have you for our 4-day Reusable Research Data Made
help you make your research data reusable and shiny!

Quick notes about the workshop:

The workshop will run from Feb 21st to Feb 24th from 9:00 AM EDT to 4:

If you registered to attend **in-person**:

- The Ontario Dairy Research Centre is located at: 6185 2nd Line Ea
- Breakfast will be served daily at the Research Centre at 8:30 AM.

If you registered to attend **on-line**:

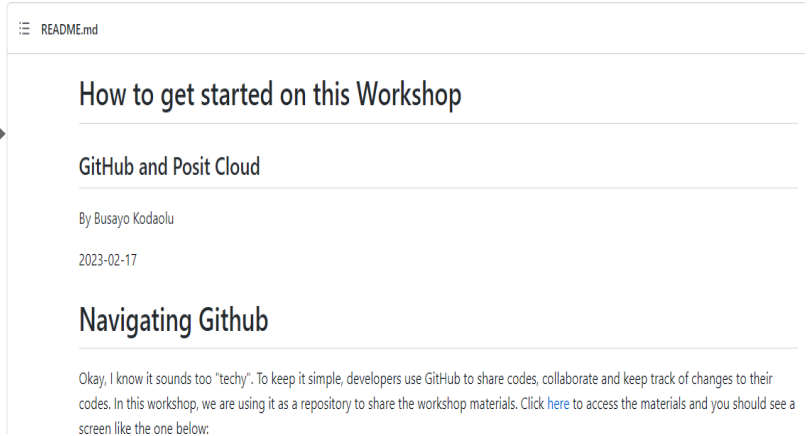
- Click here to attend the meeting on MS Teams
- Note that the first session of the day starts at 9:00 AM, but feel fr

Click here to get started on the workshop, learn where to find the mater

Have a great long weekend and see you at the workshop!

- Workshop Organizing Committee

Workshop GitHub page


README.md

How to get started on this Workshop

GitHub and Posit Cloud

By Busayo Kodaolu

2023-02-17

Navigating Github

Okay, I know it sounds too "techy". To keep it simple, developers use GitHub to share codes, collaborate and keep track of changes to their codes. In this workshop, we are using it as a repository to share the workshop materials. Click here to access the materials and you should see a screen like the one below:

Welcome email

You should have received a welcome email

Instructions on how to download workshop materials and setup the free Posit Cloud account

Hello,

We are excited to have you for our 4-day Reusable Research Data Made
help you make your research data reusable and shiny!

Quick notes about the workshop:

The workshop will run from Feb 21st to Feb 24th from 9:00 AM EDT to 4:

If you registered to attend **in-person**:

- The Ontario Dairy Research Centre is located at: 6185 2nd Line Ea
- Breakfast will be served daily at the Research Centre at 8:30 AM.

If you registered to attend **on-line**:

- Click here to attend the meeting on MS Teams
- Note that the first session of the day starts at 9:00 AM, but feel fr

Click here to get started on the workshop, learn where to find the mater

Have a great long weekend and see you at the workshop!

- Workshop Organizing Committee

Workshop GitHub page

README.md

How to get started on this Workshop

GitHub and Posit Cloud

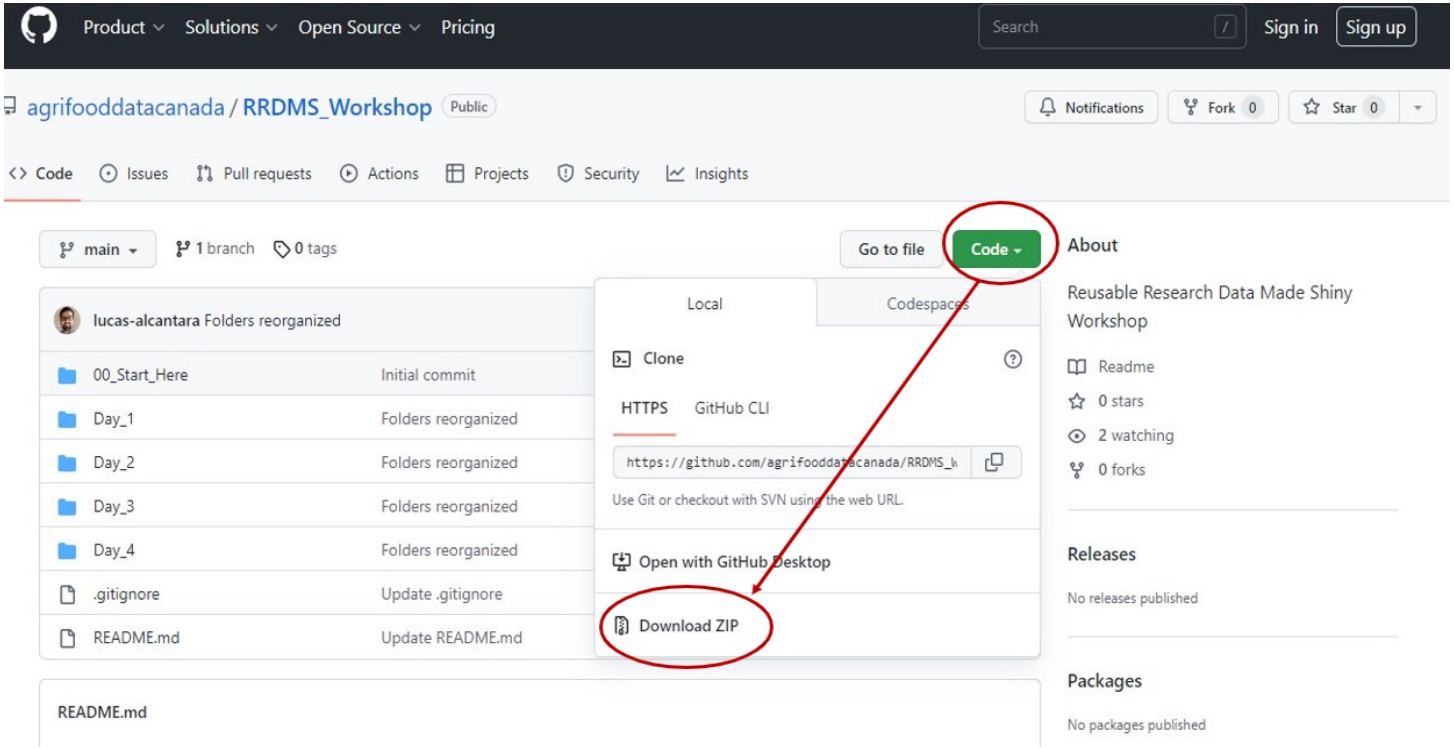
By Busayo Kodaolu

2023-02-17

Navigating Github

Okay, I know it sounds too "techy". To keep it simple, developers use GitHub to share codes, collaborate and keep track of changes to their codes. In this workshop, we are using it as a repository to share the workshop materials. Click here to access the materials and you should see a screen like the one below:

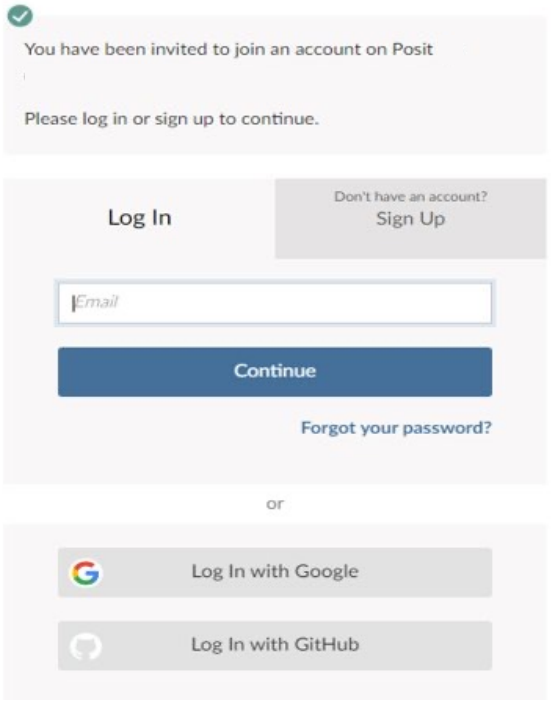
Downloading workshop materials



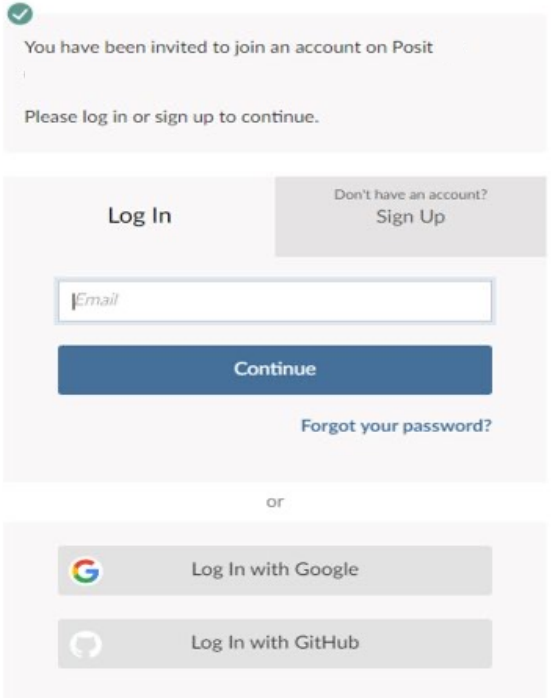
Posit cloud set-up --> Invitation email

Email inviting you into the workshop space

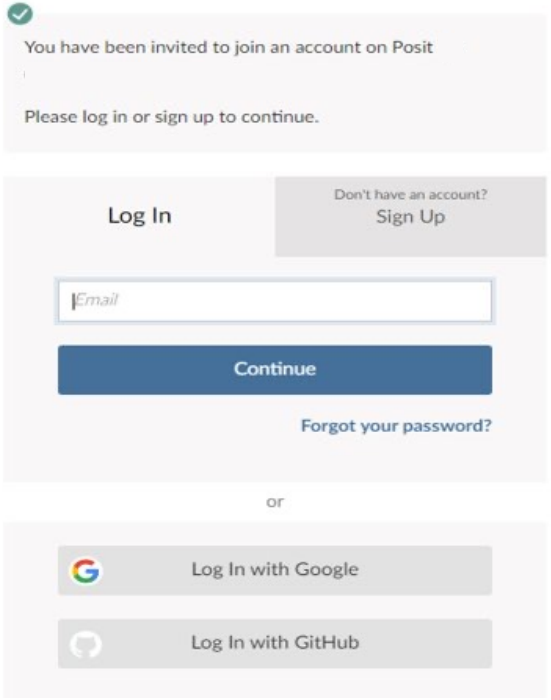
Log In or Sign Up

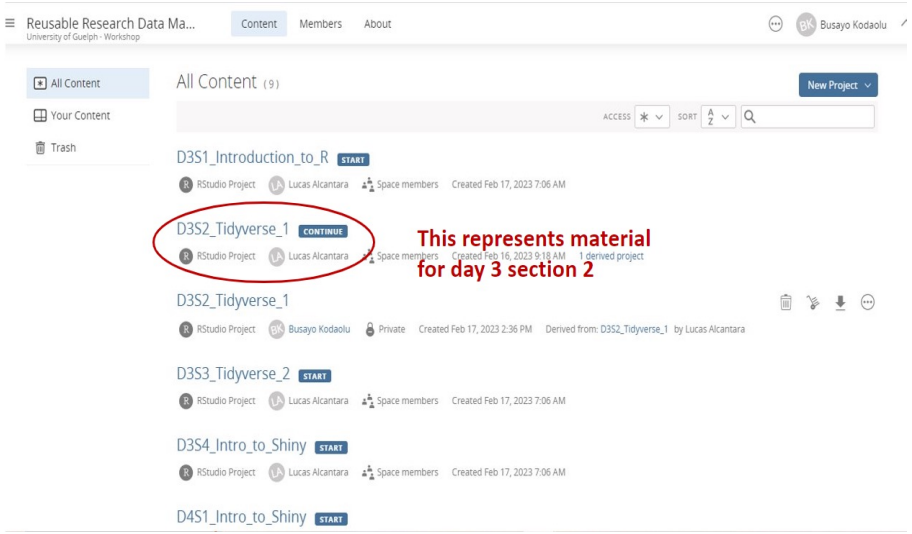


Log In or Sign Up

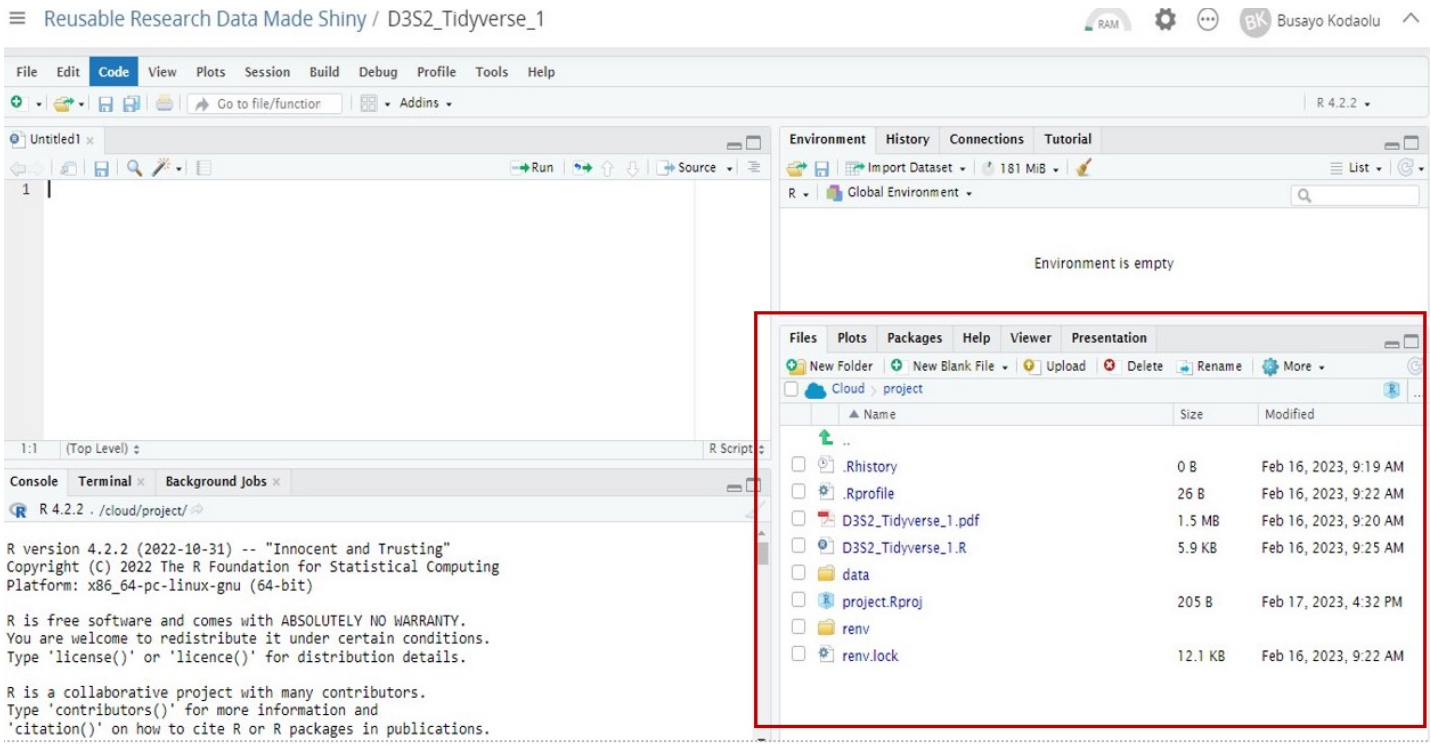


Log In or Sign Up

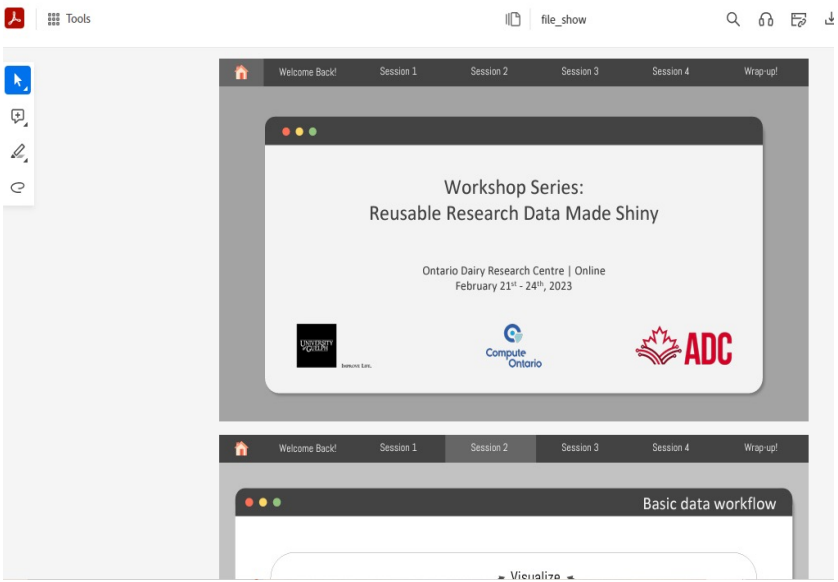


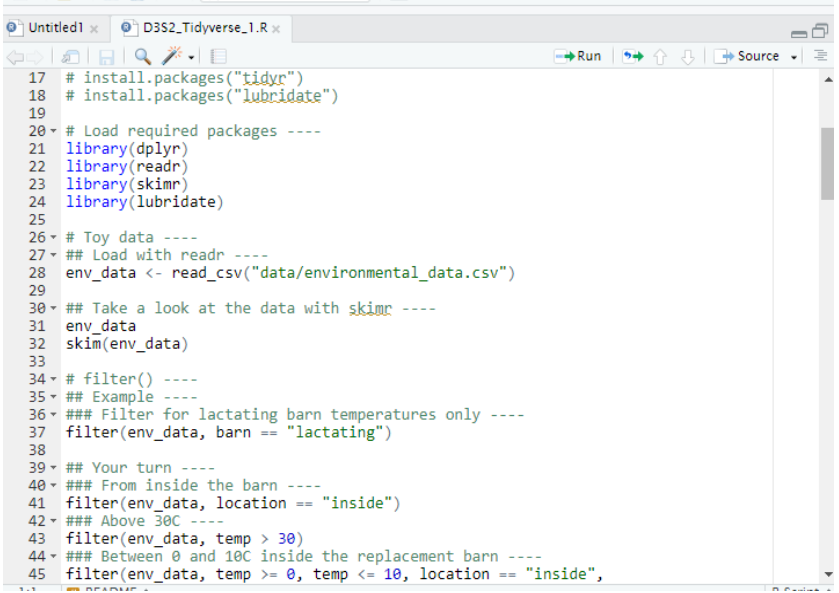


D3S2_Project files preloaded



Slides and R-script for that section





Workspaces?

Shared workspace/classroom

Workspaces?

Shared workspace/classroom

Personal workspace/office

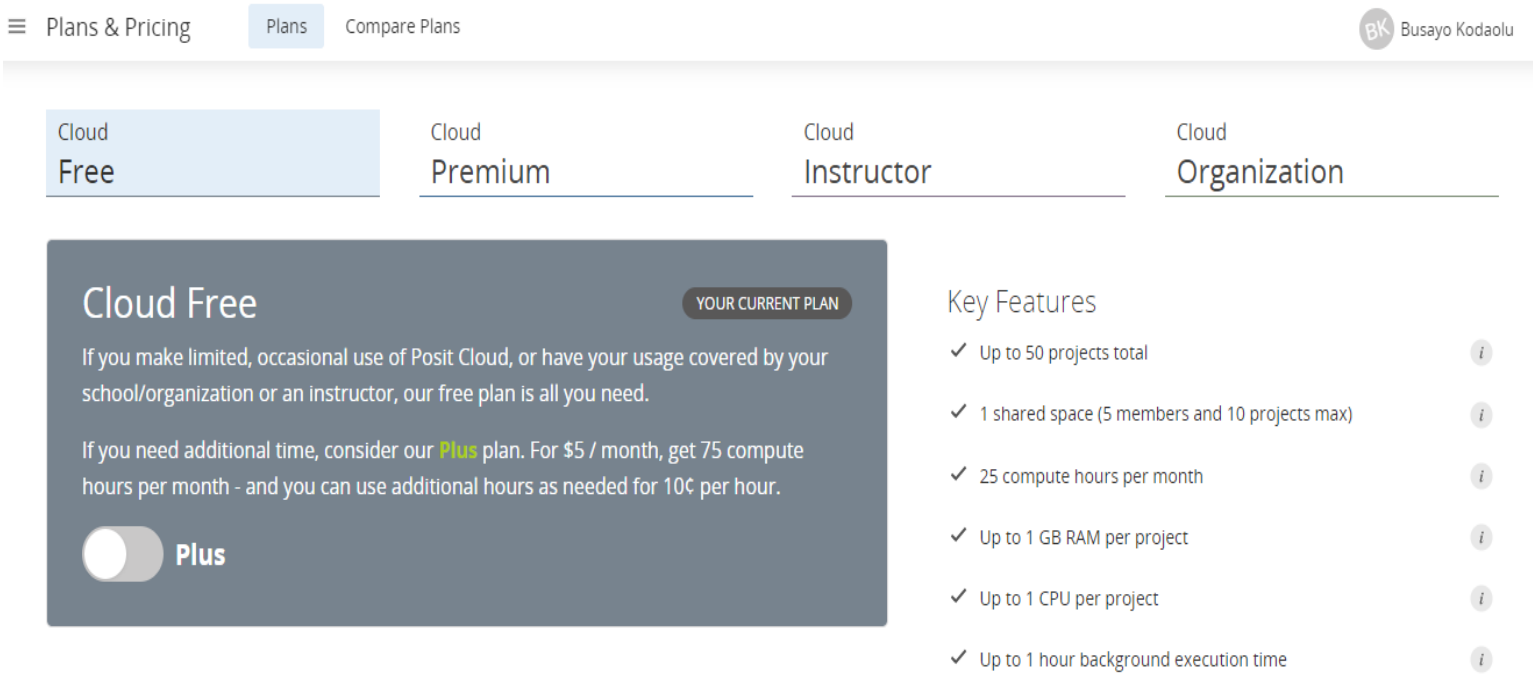
Workspaces?

Shared workspace/classroom

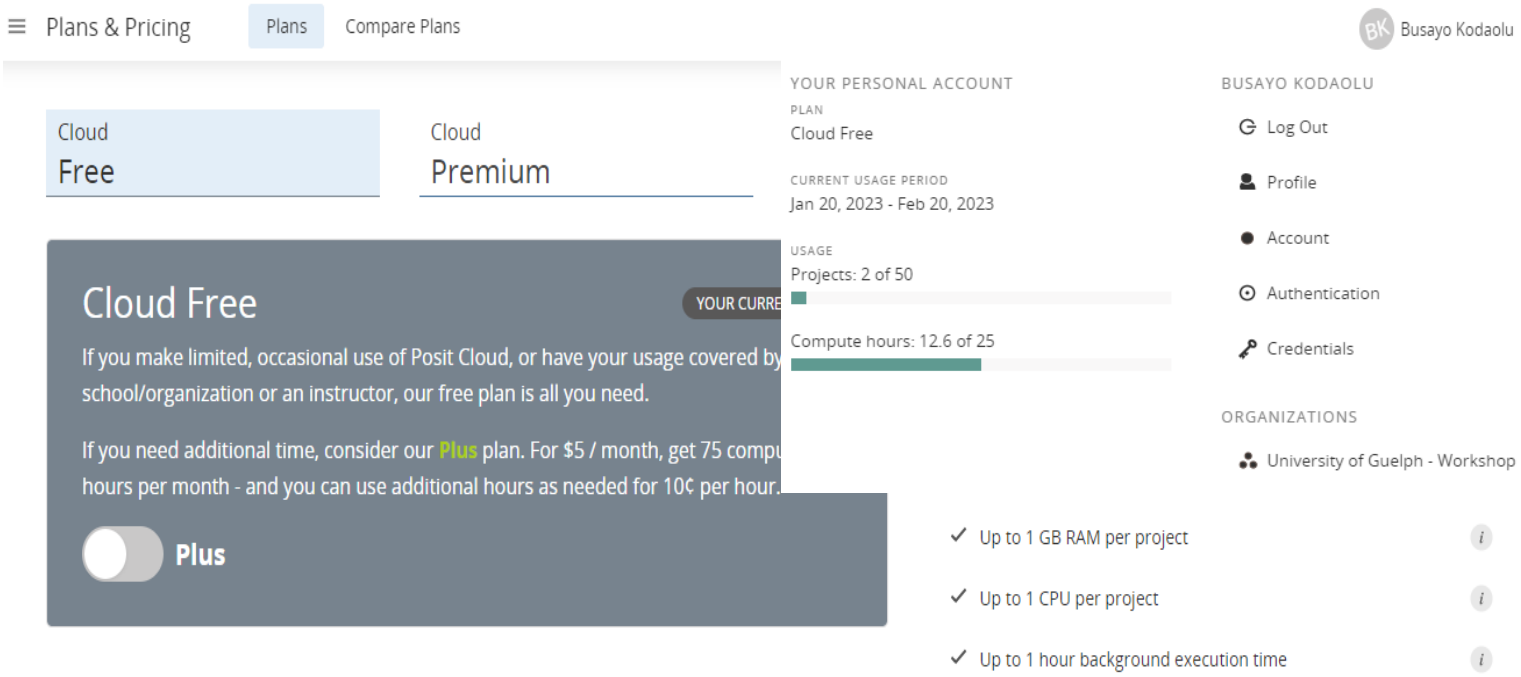
Both can be made public or private

Personal workspace/office

Your workspaces?

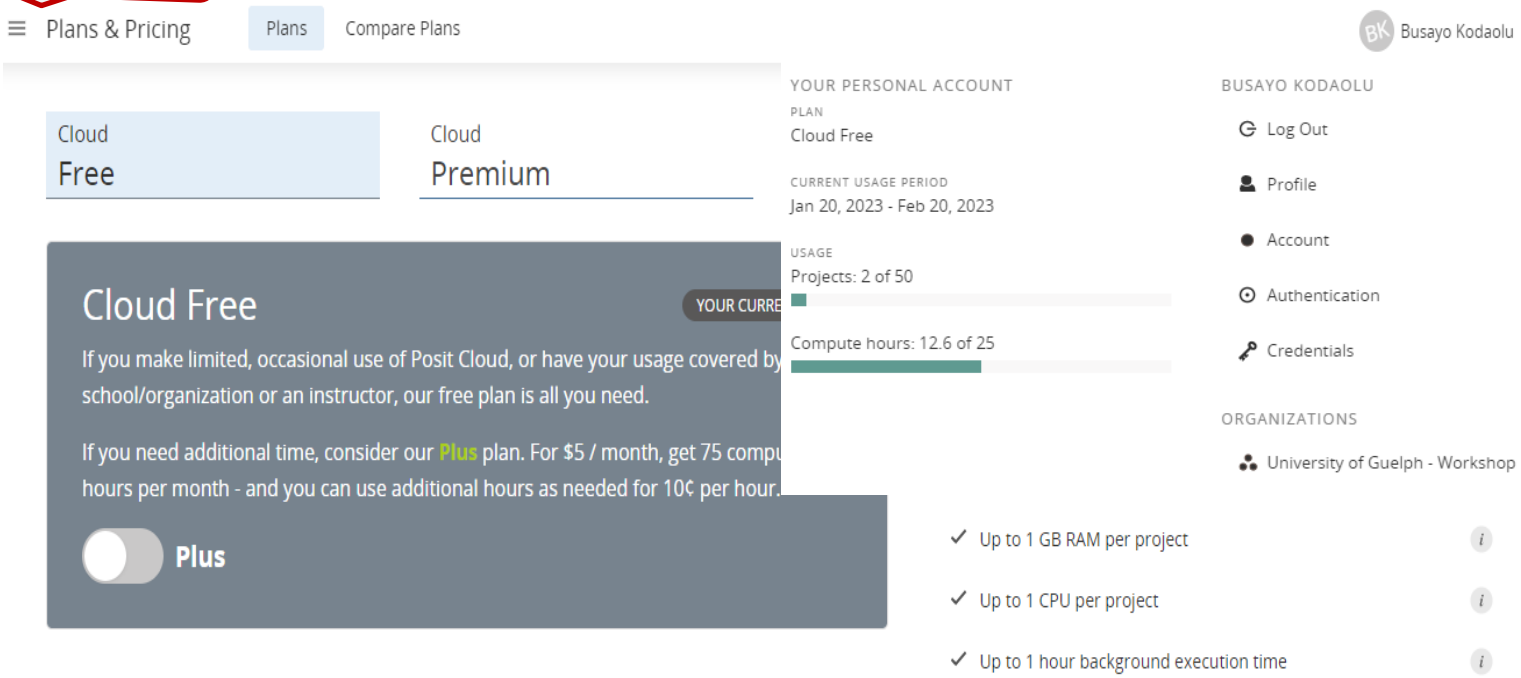


Your workspaces?



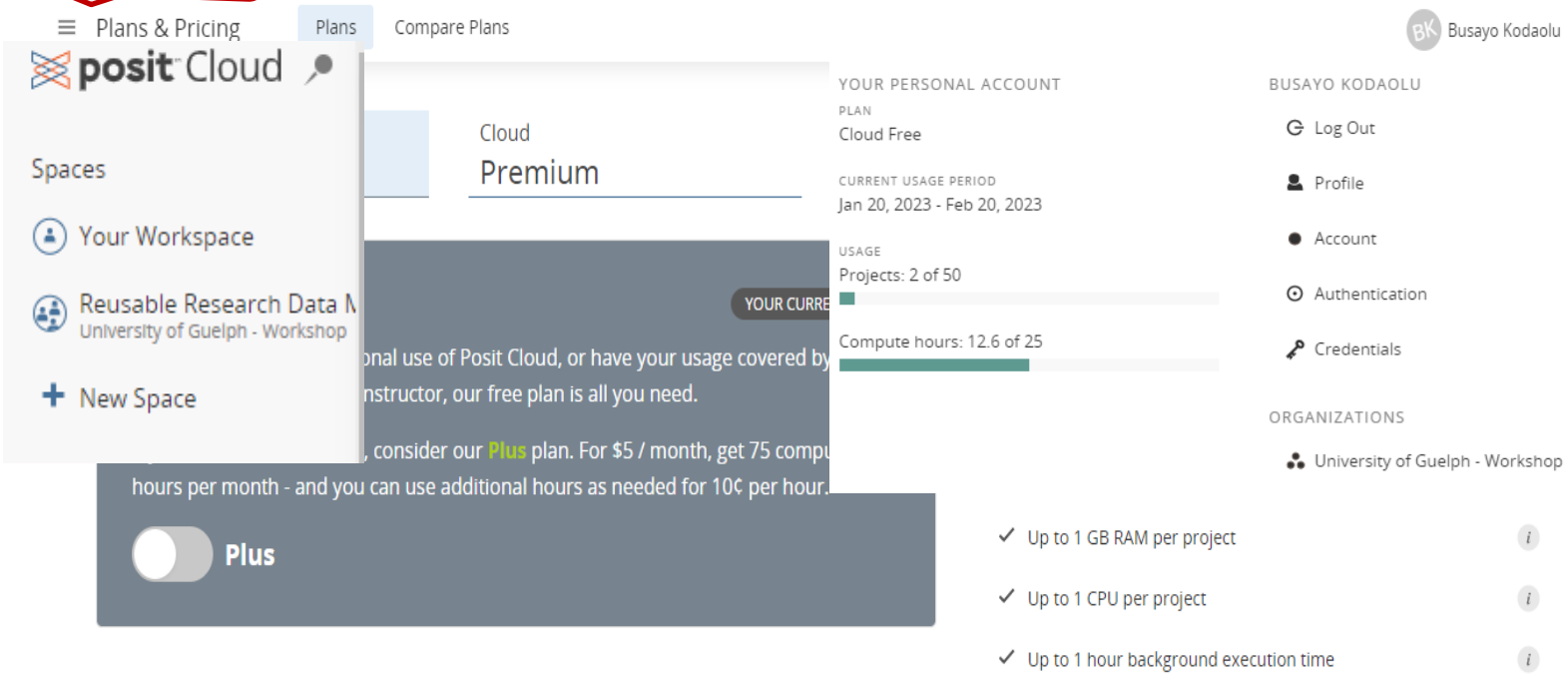
Your workspaces?

Click here

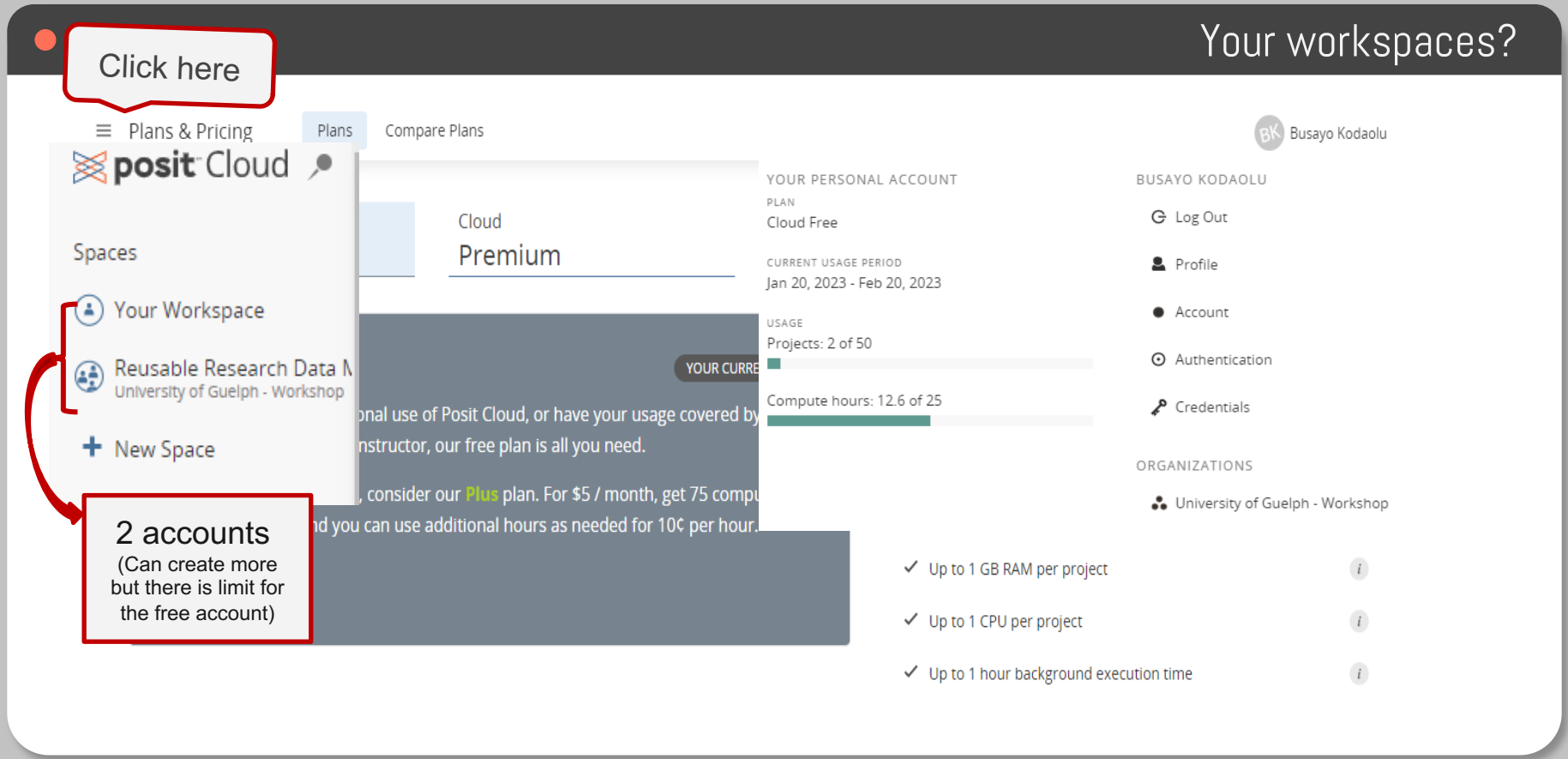


Your workspaces?

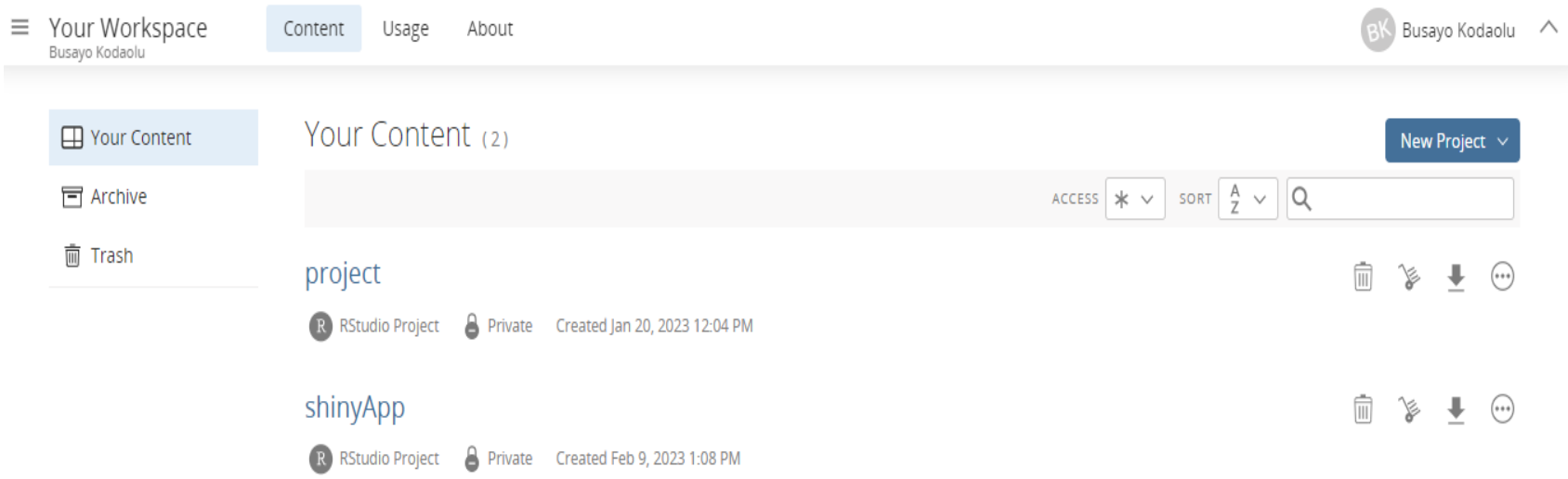
Click here



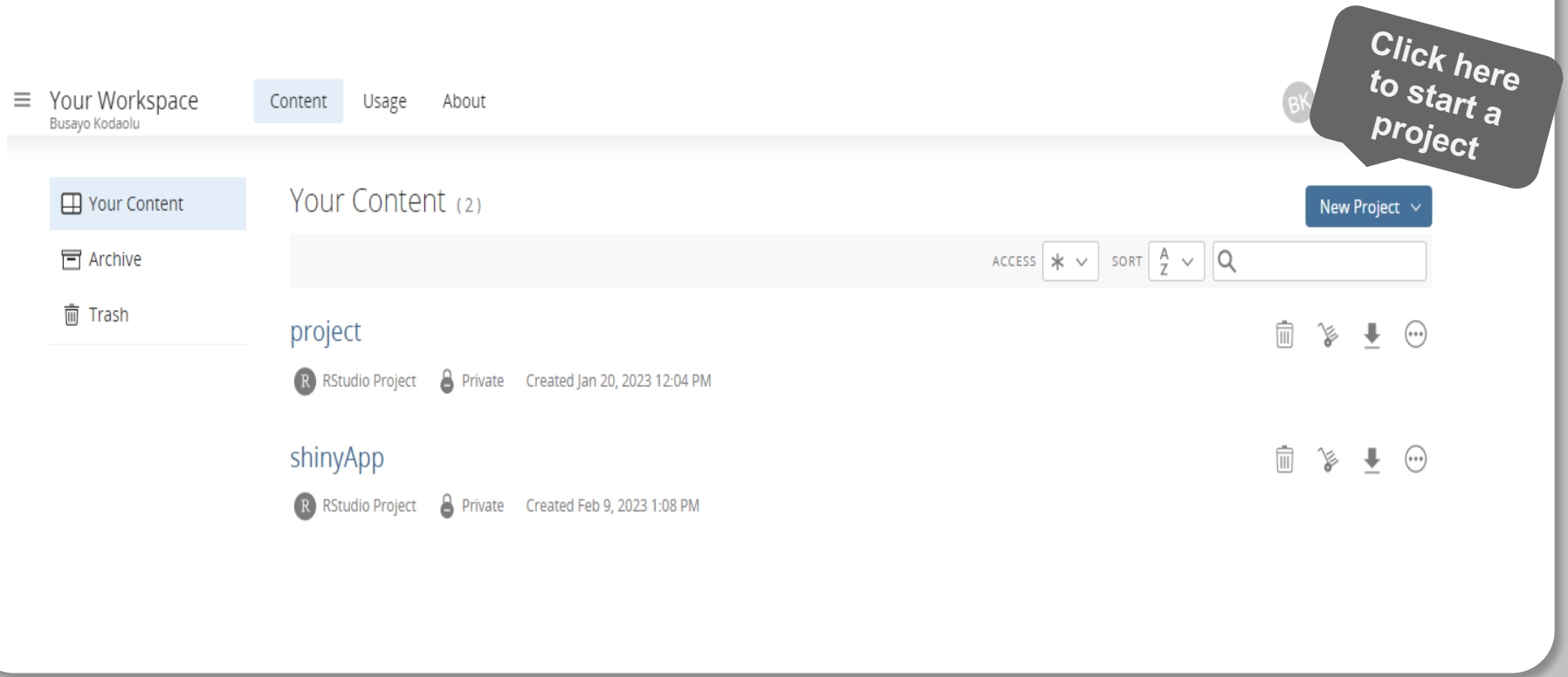
Your workspaces?

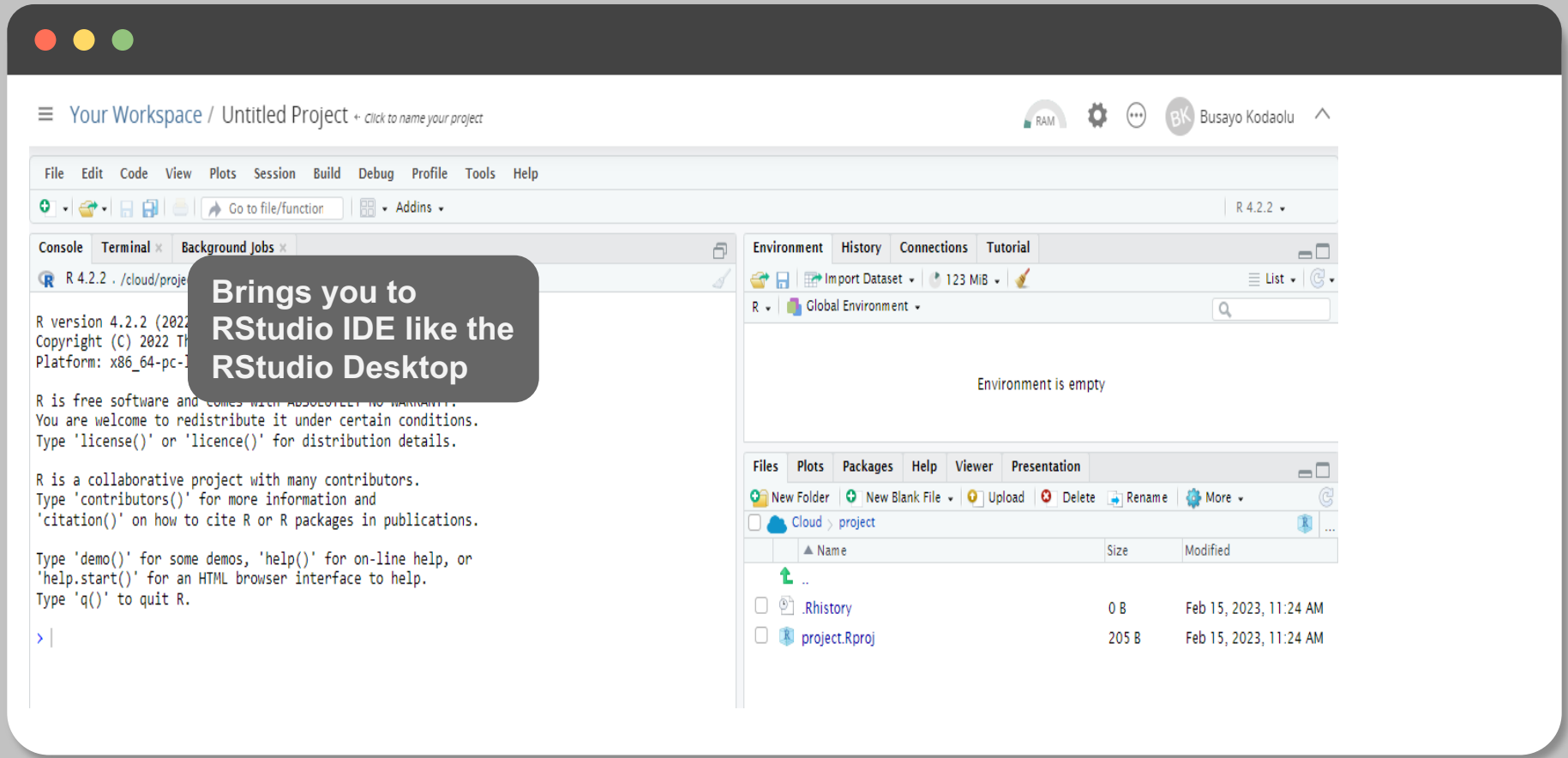


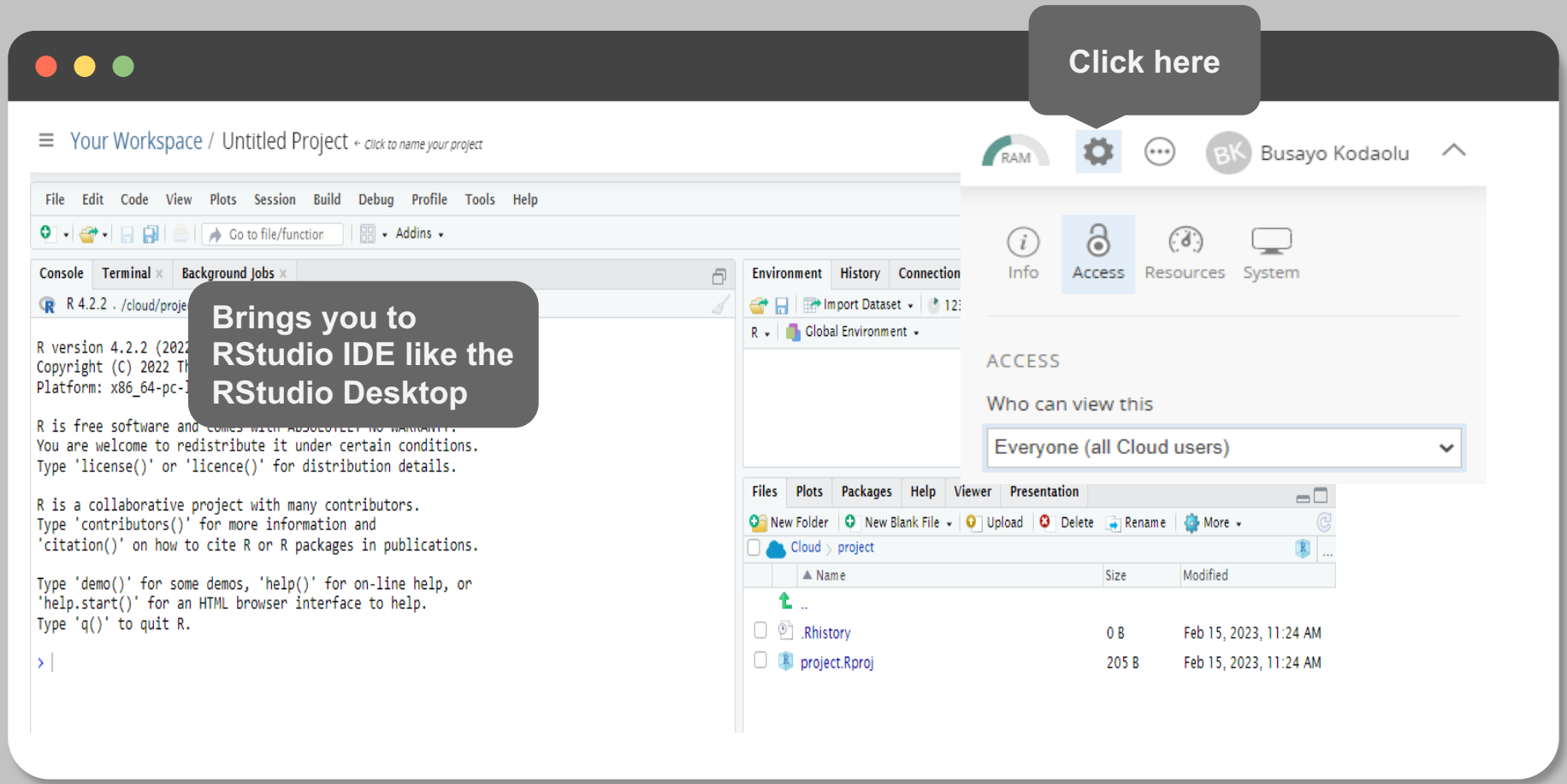
Your workspaces?



Your workspaces?




Your Workspace / Untitled Project
Busayo Kodaolu
File Edit Code View Plots Session Build Debug Profile Tools Help
Addins
R 4.2.2
Console Terminal Background Jobs
Brings you to RStudio IDE like the RStudio Desktop
Environment History Connections Tutorial
Import Dataset
123 MiB
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer Presentation
New Folder
New Blank File
Upload
Delete
Rename
More
Cloud project
Name
Size
Modified
.Rhistory
0 B
Feb 15, 2023, 11:24 AM
project.Rproj
205 B
Feb 15, 2023, 11:24 AM


Click here
Your Workspace / Untitled Project
Busayo Kodaolu
Info
Access
Resources
System
ACCESS
Who can view this
Everyone (all Cloud users)
Brings you to RStudio IDE like the RStudio Desktop
File Edit Code View Plots Session Build Debug Profile Tools Help
Console Terminal Background Jobs
Environment History Connection
Files Plots Packages Help Viewer Presentation
New Folder New Blank File Upload Delete Rename More
Cloud project
Name Size Modified
.Rhistory 0 B Feb 15, 2023, 11:24 AM
project.Rproj 205 B Feb 15, 2023, 11:24 AM

Introduction to R and Rstudio

Marjan Asgari

R Programming Language

R, developed by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand in the mid-1990s, is a programming language for mostly statistical computing and graphics.

IEEE publishes a list of the most popular programming languages each year. R was ranked 5th in 2016, up from 6th in 2015. Due to its expressive syntax and easy-to-use interface, it has grown in popularity in recent years.

R provides tools and techniques for

- Linear and nonlinear modeling
- Time-series analysis
- Classification and Clustering
- Graphical Representation of data
- etc.

Adapted from: https://www.datamentor.io/r-programming

Why R Language?

Why R?

- R is open-source and free
- R runs on all platforms (Windows, Linux and Mac)
- R has lots of packages
  - ✓ R language has more than 10,000 packages stored in the CRAN repository, and the number is continuously increasing.
- R facilitates quality plotting and graphing.
  - ✓ The popular libraries like *ggplot2* and *plotly* are used for visually appealing graphs that makes R outstanding from other programming languages.
- R is Highly Compatible
  - ✓ R can be paired with many other programming languages like C, C++, Java, and Python.

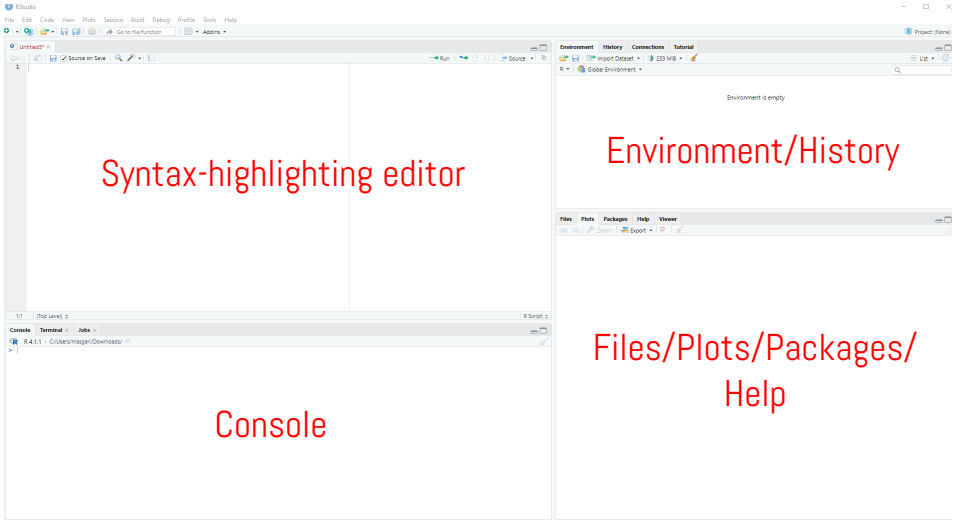
Adapted from: https://www.simplilearn.com/what-is-r-article

What is RStudio?

RStudio is a free and open-source Integrated Development Environment (IDE) for R

RStudio is available in two editions:

- **RStudio Desktop:** the program is run locally as a regular desktop application;
- **RStudio Server**: allows accessing RStudio using a web browser while it is running on a remote Linux server.



Adapted from: https://www.r-project.org

RStudio Projects

Creating a project in RStudio is a good practice for several reasons:

- **Organization:** By creating a project, you can keep all your R scripts, R markdown documents, R functions and data together in one place. This makes it easier to keep track of your progress, collaborate with others, and share your work.
- **Isolation:** R projects are self-contained. It means the libraries, packages, and environment variables that you use in one project will not interfere with other projects.
- **Reproducibility:** Projects in RStudio allow you to save your workspace and the state of your environment, so that you can easily reproduce your analysis later.

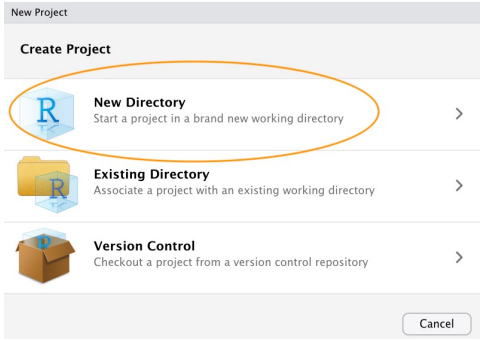
All in all, creating a project in RStudio provides a streamlined workflow that helps you stay organized, focused, and productive while working on your data analysis projects.

Adapted from: https://intro2r.com

RStudio Projects

How to create an RStudio Project?

- Open RStudio and click on "File" from the top menu bar, then click "New Project".
- In the "New Project" window, select the type of project you want to create. You can choose from a variety of project types, such as "New Directory", "Existing Directory", "Version Control", or "Package".

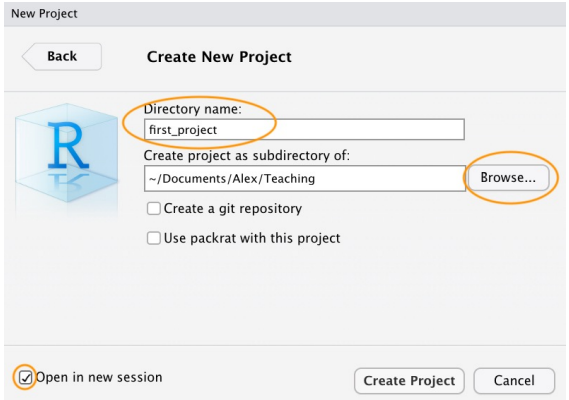


Adapted from: https://intro2r.com

RStudio Projects

How to create an RStudio Project?

- Next, you'll need to specify the project directory and location. Choose a name for your project and select the directory where you want to save your project files. You can create a new directory or choose an existing one.

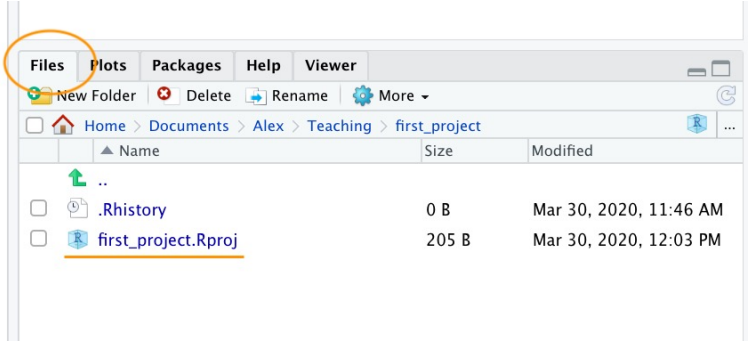


Adapted from: https://intro2r.com

RStudio Projects

How to create an RStudio Project?

- Now, you have a new folder on your computer containing an RStudio project file called first_project.Rproj. This .Rproj file contains various project options and can also be used as a shortcut for opening the project directly from the file system (just double click on it).
- You can check this out in the 'Files' tab in RStudio.



RStudio Projects

How to create an RStudio Project?

- Once your project is created, you can start coding by opening a new R script file or opening an existing R script file.
- You can also use RStudio's other features, such as the console, the environment tab, and the plot tab, to analyze and visualize your data.

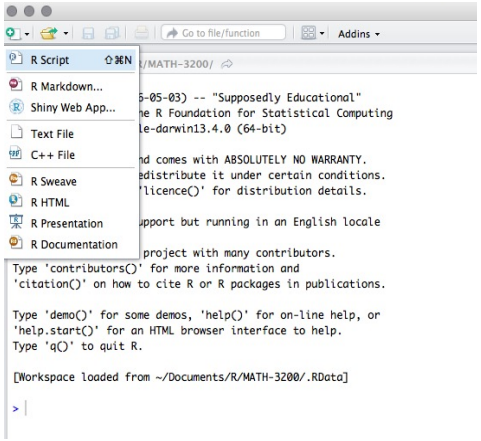
Adapted from: https://intro2r.com

R Scripts

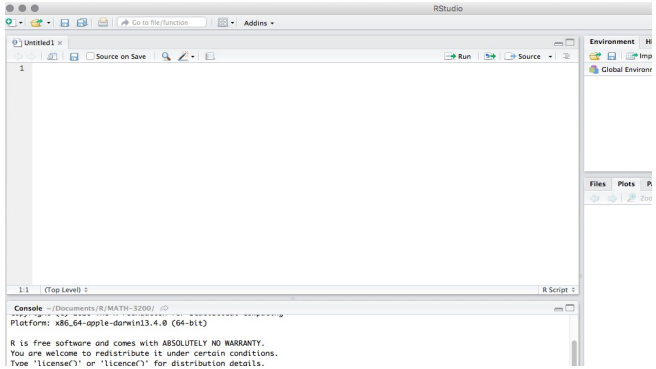
Creating, opening, and running R scripts are important tasks in R programming.

Creating a new R script:

In RStudio, click on "File" from the top menu bar and select "New File". From the dropdown menu, select "R Script" and click "OK".



A new blank R script will appear in the editor window. You can start writing your code in this file

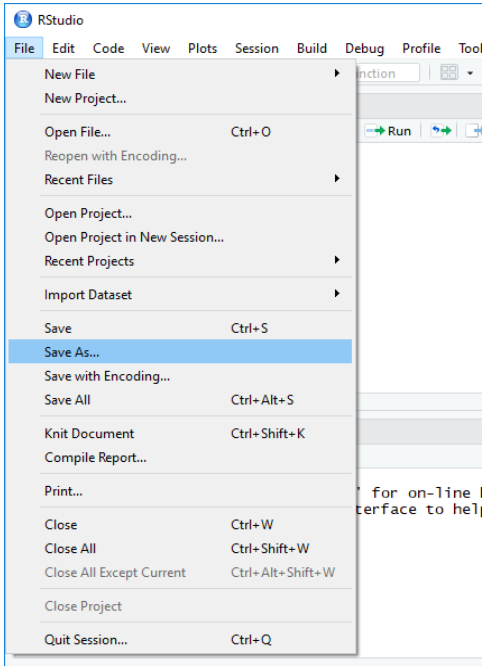


Adapted from: https://rforhr.com/g

R Scripts

Creating a new R script:

If you want to save the changes you made to an existing R script, simply click on "File" from the top menu bar and select "Save".

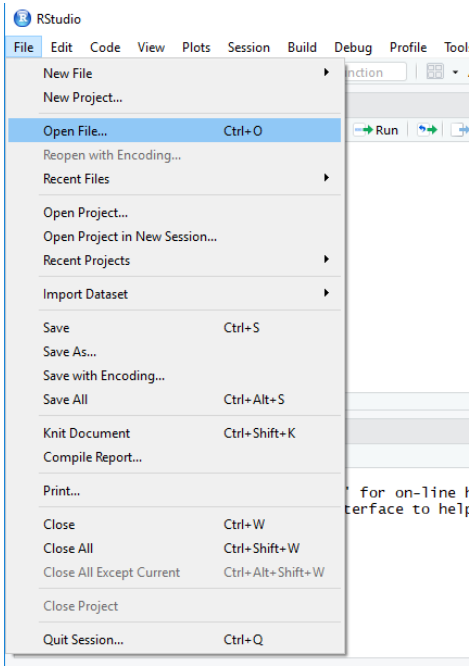


Adapted from: https://rforhr.com/g

R Scripts

Opening an existing R script:

- In RStudio, click on "File" from the top menu bar and select "Open File".
- Navigate to the location of the R script on your computer and select it.
- The R script will open in a new tab in the editor window, and you can start working with the code.

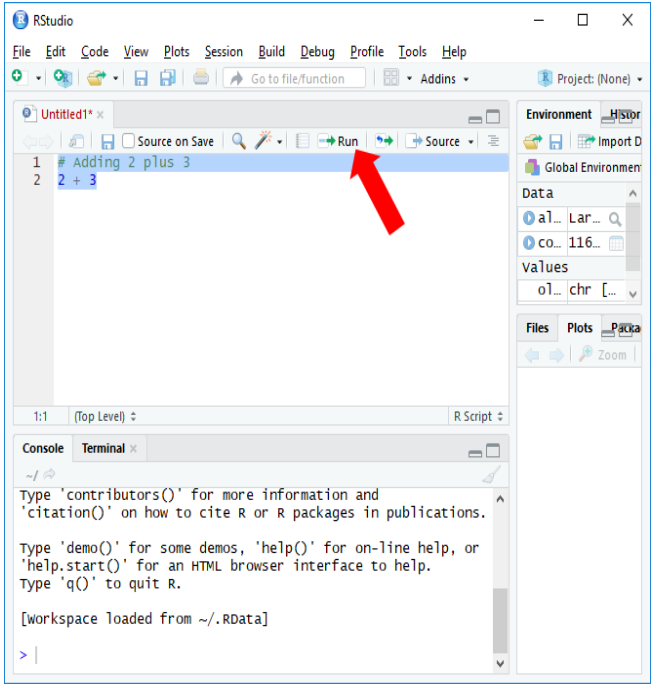


Adapted from: https://intro2r.com

R Scripts

Running an R script:

- In the editor window of RStudio, select the code you want to run.
- Click on the "Run" button located in the top-right corner of the editor window.
- The output of your code will appear in the console window at the bottom
- of the screen.



Adapted from: https://intro2r.com

Navigating the tree of directory

Get the path of the current directory with **getwd()** (get working directory)

- getwd()

Change working directory with **setwd()** (set working directory)

- setwd("~/Rcourse")

Variables in Programming

In programming languages, a **Variable** is a named location in memory that can hold a value. Variables are used to store data that can be manipulated and processed by the program.

Variables are defined using a **Name** and a **Data Type**.

- The name of the variable is used to refer to the data stored in the memory location, and the data type specifies the type of data that can be stored in the variable.

Variables can be assigned a value using an assignment operator (=) (and/or (<-) (leftward) in R programming language).

For example, in R programming, you can define a variable named "x" to hold a numeric value using the following syntax:

x <- 10 or x = 10

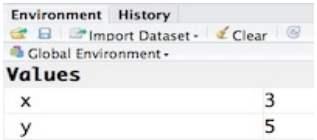
Adapted from: https://www.geeksforgeeks.org/r-variables/

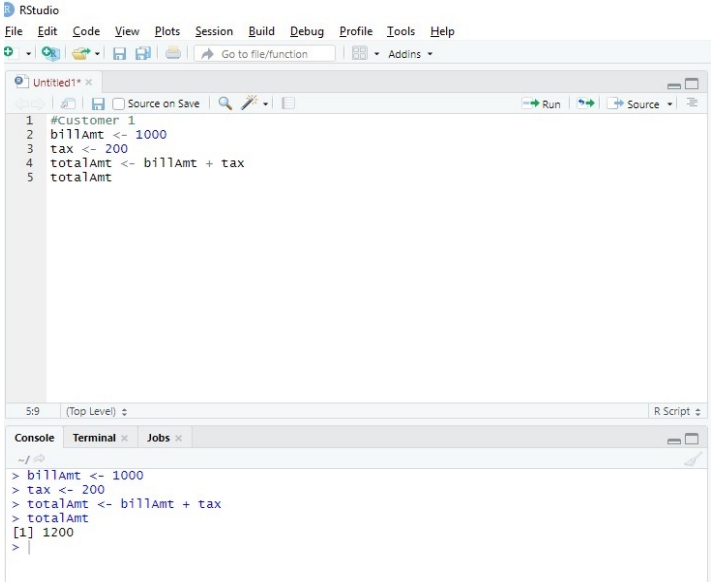
Variables in Programming

Variables can be used in a program to

- Store Input Data
- Perform various operations, such as arithmetic, comparison, and logical operations.
- Store the results of a computation or to pass data between functions or modules within the program.

You can also view information on the variable by looking in your Environment window in the upper right-hand corner of the RStudio interface.





Adapted from: https://biocorecrg.github.io/CRG_RIntroduction/

Variables in Programming

Tips on Variable Names

- Make your names explicit and not too long
- Avoid names starting with a number (2x is not valid, but x2 is)
- Avoid names of fundamental functions in R (*e.g.,* if, else, for). When in doubt check the help to see if the name is already in use
- Avoid dots (.) within a variable name as in my.dataset
- Use nouns for object names
- Keep in mind that R is case sensitive

Basic Data Types

R supports various data types. Some of the basic data types in R are:

1. **Numeric**: This data type represents numeric values, including integers and real numbers.
    - Numeric values can be positive, negative, or zero
    - They are represented as double precision floating-point values by default.
2. **Character**: This data type represents textual data, such as names, sentences, and paragraphs. In R, character values are enclosed in quotes, either single ('') or double ("").
3. **Logical**: This data type represents Boolean values, which are either TRUE or FALSE.

Adapted from: https://www.geeksforgeeks.org/r-data-types/

Programming Operators

Operators are used for operations on data. We have Arithmetic, Comparison, Logical, and Assignment operations.

- **Arithmetic Operators**: These are used to perform **Mathematical Calculations** on numeric values.

+	Addition →	X <- 2+2
-	Subtraction →	X <- 2-2
*	Multiplication →	X <- 2*2
/	Division →	X <- 2/2
^	Exponentiation →	X <- 2^2
%%	Remainder after division →	X <- 2%%2

Adapted from: https://www.w3schools.com/r/

Programming Operators

Operators are used for operations on data. We have Arithmetic, Comparison, Logical, and Assignment operations.

- **Comparison Operators**: These are used to compare two values and return a logical value (TRUE or FALSE).

< Less than →	2<3	return: TRUE
> Greater than →	2>3	return: FALSE
<= Less than or equal to →	2<=3	return: TRUE
>= Greater than or equal to →	2>=3	return: TRUE
== Equal to →	2==3	return: FALSE
!= Not equal to →	2!=3	return: TRUE

Adapted from: https://www.w3schools.com/r/

Programming Operators

Operators are used for operations on data. We have Arithmetic, Comparison, Logical, and Assignment operations.

- **Logical Operators**: These are used to combine or negate logical values.

  & Element-wise logical AND → (X > 30 & Y<15)
  | Element-wise logical OR → (X > 30 | Y<15)
  ! Element-wise logical NOT
  && Short-circuiting logical AND
  || Short-circuiting logical OR

Adapted from: https://www.w3schools.com/r/

DataFrames in R

In R, a data frame is a two-dimensional object for storing data tables in rows and columns. The columns should have the same size, but they can store different data types.

A data frame in R can be created using the **data.frame()** function. For example, to create a data frame with three columns, “Name“, “Age", and “ClassA”, we can use the following code:

```
Source on Save
1  Students <- data.frame (
2    Name = c("Lily", "Ben", "Tony"),
3    Age = c(20, 16, 31),
4    ClassA = c(TRUE, TRUE, FALSE)
5  )
```

```
> Students
  Name Age ClassA
1 Lily  20   TRUE
2  Ben  16   TRUE
3 Tony  31  FALSE
```

- c() function: Creates a vector of objects with the same datatypes.
- All columns have the same length.
- The name of data frame is “Students”.
- Students have columns with Numeric, character and logical datatypes.

Adapted fromhttps://www.w3schools.com/r

DataFrames in R

You can access the elements of a data frame using the $ operator or the square bracket [] operator.

- For example, to access "Name" column of the "Students" data frame we can use the following code:

  Students $Name

- To access the second row of the "Students" data frame , we can use the following code:

  Students[2,]

You can also use logical operators to select specific rows or columns based on a condition. For example, to select all rows where the Age is greater than 15, you can use the following code:

Students[Students$Age > 15,]

Get first rows of a DataFrame: **head()** → head(Students)
Get last rows of a DataFrame: **tail()** → tail(Students)

Adapted fromhttps://www.w3schools.com/r

Basic Built-in Functions in R

Some basic Functions in R:

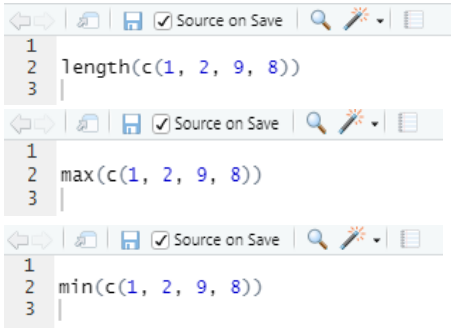
- print() - Prints the specified input to the console.
- sum() - Returns the sum of numeric data.
- mean() - Returns the mean of a numeric vector.
- sd() - Returns the standard deviation of a numeric vector.
- seq() - Generates a sequence of numbers.

Adapted fromhttps://www.w3schools.com/r

Basic Built-in Functions in R

Some basic Functions in R:

- length() - Returns the length of a vector.
- max() - Returns the maximum value of a vector.
- min() - Returns the minimum value of a vector.



Adapted fromhttps://www.w3schools.com/r

Calling Functions in R

Built-in functions: In R, you can call a function by typing the name of the function followed by parentheses containing any necessary arguments.

Functions in Packages

1 - Explicitly loading the package:

In R, you can call a function from a package by **first loading the package into your R session** using the library() or require() functions, and then calling the function using the function name.

```
library(ggplot2)
MyGraph <- ggplot(....)
```

Calling Functions in R

Built-in functions: In R, you can call a function by typing the name of the function followed by parentheses containing any necessary arguments.

Functions in Packages

2 - Without loading the package:

In R, you can call a function from a package without explicitly loading the package by using the double colon operator ::

The double colon operator allows you to access functions and objects in a package without loading the entire package into the current R session.

```
MyGraph <- ggplot2::ggplot(....)
```

Commenting in R

Comments are generic English sentences, mostly written in a program to explain what it does or what a piece of code is supposed to do. More specifically, information that programmer should be concerned with, and it has nothing to do with the logic of the code. They are completely ignored by the compiler and are thus never reflected on to the input.

Comments are generally used for the following purposes:

- Code Readability
- Explanation of the code or Metadata of the project
- Prevent execution of code
- To include resources

Adapted fromhttps://www.w3schools.com/r

Commenting in R

There is one type of comment in R: **single-line comments**
Single-line comments start with the **# symbol** and continue until the end of the line.

```
# This is a single-line comment
```

It is good practice to include comments in your code to make it easier to understand and maintain. It is also important to keep your comments up to date as you make changes to your code.

Adapted from https://www.w3schools.com/r

Some RStudio Useful Shortcuts

Description	Windows & Linux	Mac
Clear console	Ctrl+L	Ctrl+L
Change working directory	Ctrl+Shift+H	Ctrl+Shift+H
Save all documents	Ctrl+Alt+S	Cmd+Option+S
Undo	Ctrl+Z	Cmd+Z
Redo	Ctrl+Shift+Z	Cmd+Shift+Z
Indent	Tab (at beginning of line)	Tab (at beginning of line)
Outdent	Shift+Tab	Shift+Tab
Search R Help	Ctrl+Alt+F1	Ctrl+Option+F1
Save RScripts	Ctrl+S	Cmd+S
Comment/uncomment a Line	Ctrl+Shift+C	Cmd+Shift+C
Run Codes	Ctrl+Enter	Cmd+Enter

Adapted from: https://support.posit.co/

Coffee Break!