Statistical Computing with R: Masters in Data Sciences 503 (S29) Second Batch, SMS, TU, 2023

Shital Bhandary

Associate Professor

Statistics/Bio-statistics, Demography and Public Health Informatics

Patan Academy of Health Sciences, Lalitpur, Nepal

Faculty, Data Analysis and Decision Modeling, MBA, Pokhara University, Nepal

Faculty, FAIMER Fellowship in Health Professions Education, India/USA.

Review Preview:

- R script
- R markdown
- R notebook
  - Like jupyter notebook
- Project in R studio
- Working with Project
- Version control of Project
- R Shiny application

Chapter 8 (Workflow:projects) of the course text book: R for Data Science

- One day you will need to quit R, go do something else and return to your analysis the next day.
- One day you will be working on multiple analyses simultaneously that all use R and you want to keep them separate.
- One day you will need to bring data from the outside world into R and send numerical results and figures from R back out into the world.

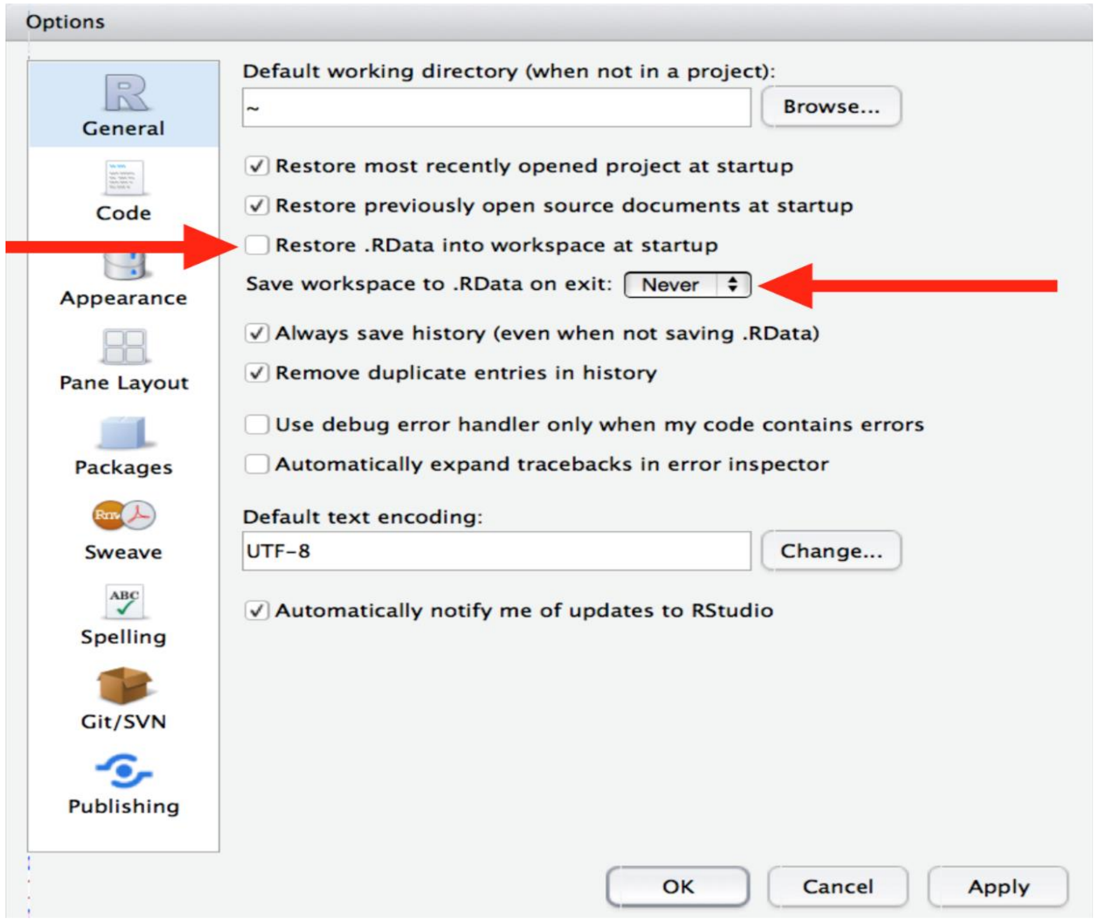
- To handle these real life situations, you need to make two decisions:
  - What about your analysis is "real", i.e. what will you save as your lasting record of what happened?
  - Where does your analysis "live"?

Chapter 8 (Workflow:projects) of the course text book: R for Data Science

- As a beginning R user, it's OK to consider your environment (i.e. the objects listed in the environment pane) "real".
- However, in the long run, you'll be much better off if you consider your R scripts as "real".
- With your R scripts (and your data files), you can recreate the environment. It's much harder to recreate your R scripts from your environment!
- You'll either have to retype a lot of code from memory (making mistakes all the way) or you'll have to carefully mine your R history.

Chapter 8 (Workflow:projects) of the course text book: R for Data Science

- To foster this behaviour, it is highly recommended to instruct RStudio not to preserve your workspace between sessions.



Default working directory (when not in a project): ~

✓ Restore most recently opened project at startup
✓ Restore previously open source documents at startup
☐ Restore .RData into workspace at startup
Save workspace to .RData on exit: Never
✓ Always save history (even when not saving .RData)
✓ Remove duplicate entries in history
☐ Use debug error handler only when my code contains errors
☐ Automatically expand tracebacks in error inspector

Default text encoding: UTF-8

✓ Automatically notify me of updates to RStudio

Chapter 8 (Workflow:projects) of the course text book: R for Data Science

- This will cause you some short-term pain, because now when you restart RStudio it will not remember the results of the code that you ran last time.
- But this short-term pain will save you long-term agony because it forces you to capture all important interactions in your code.
- There's nothing worse than discovering three months after the fact that you've only stored the results of an important calculation in your workspace, not the calculation itself in your code.
- Therefore we need "Project" and R Studio provides this feature to us!

R Studio Project:

https://support.posit.co/hc/en-us/articles/200526207-Using-RStudio-Projects

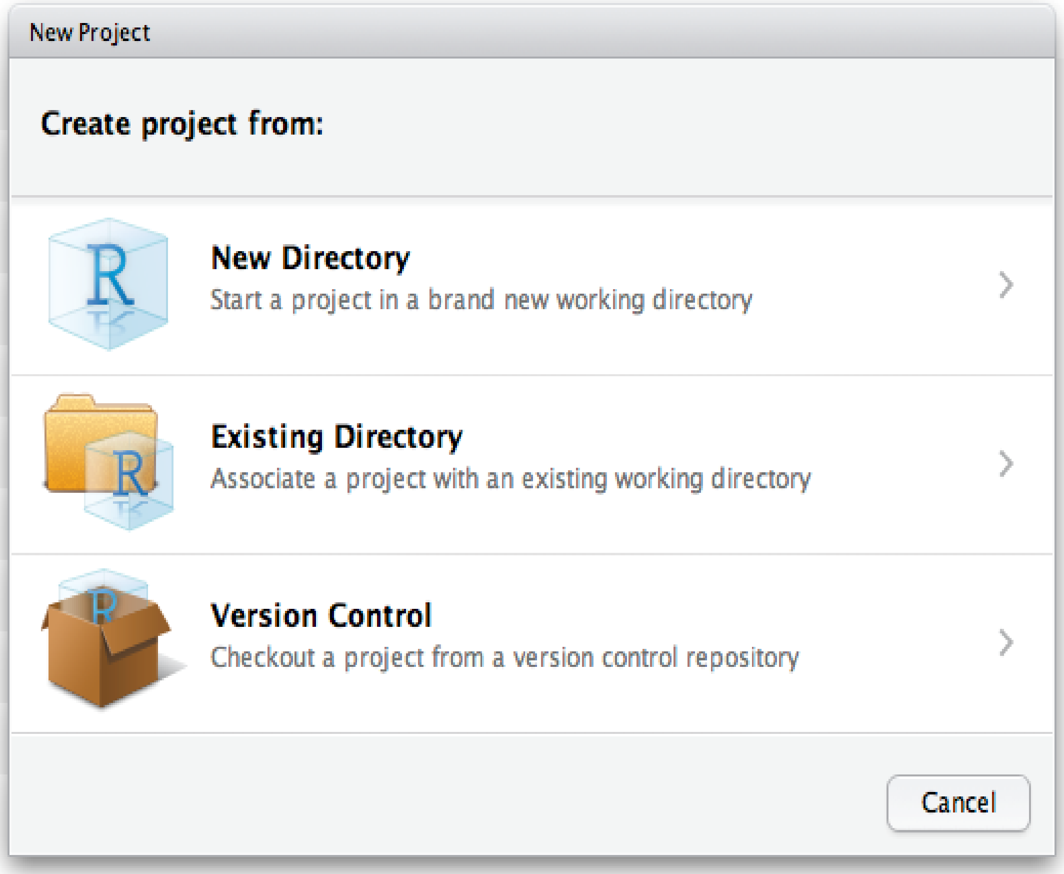
- R **experts keep all the files associated with a project together** — input data, R scripts, analytical results, figures.
- This is such a wise and common practice that RStudio has built-in support for this via **projects**.
- RStudio projects make it straightforward to divide your work into multiple contexts, each with their own working directory, workspace, history, and source documents.

#Creating Projects

- RStudio projects are associated with R working directories.
- You can create an RStudio project:
  - In a brand new directory
  - In an existing directory where you already have R code and data
  - By cloning a version control (Git or Subversion) repository

R Studio project: Let’s do it

- To create a new project in the RStudio IDE, use the **Create Project** command (available on the Projects menu and on the global toolbar):


New Project

Create project from:

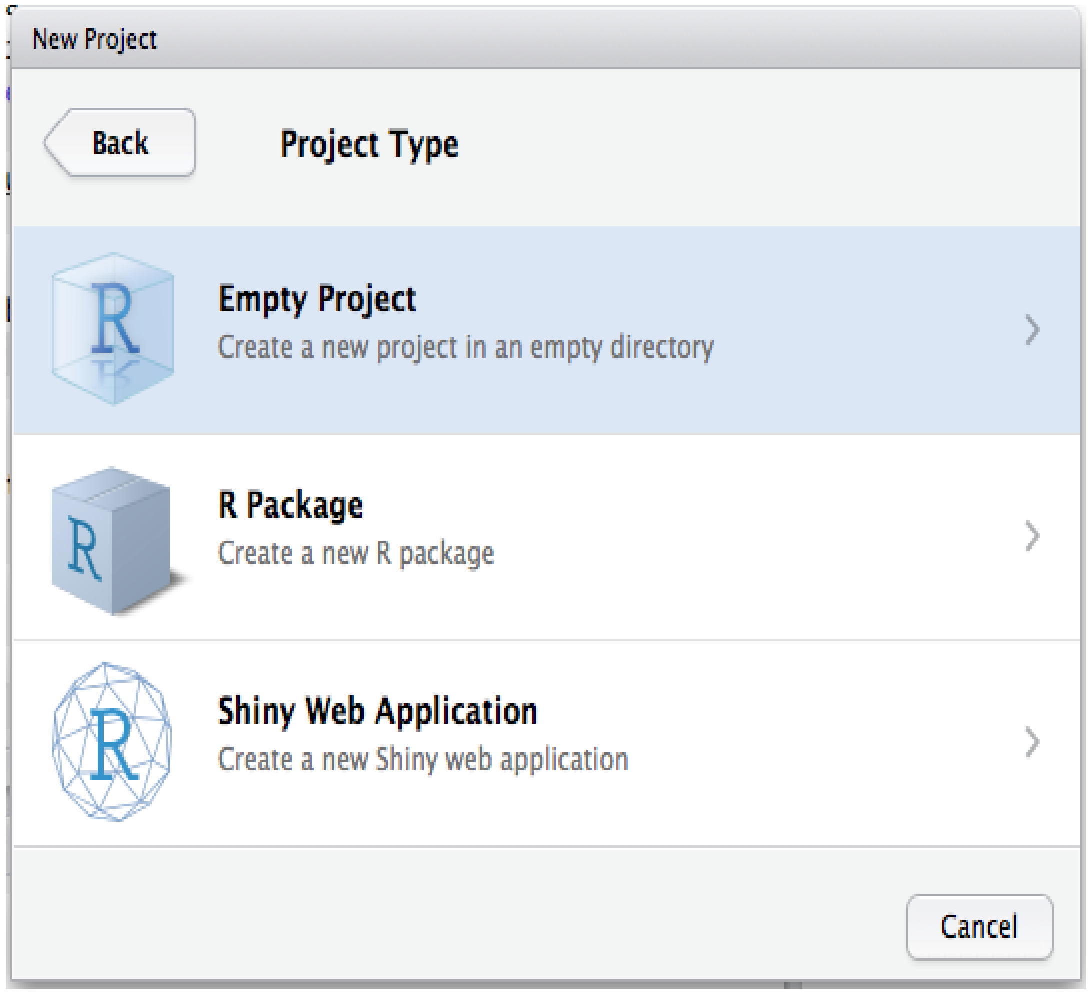
New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Cancel

Working with Project: Let's do it



New Project

Back

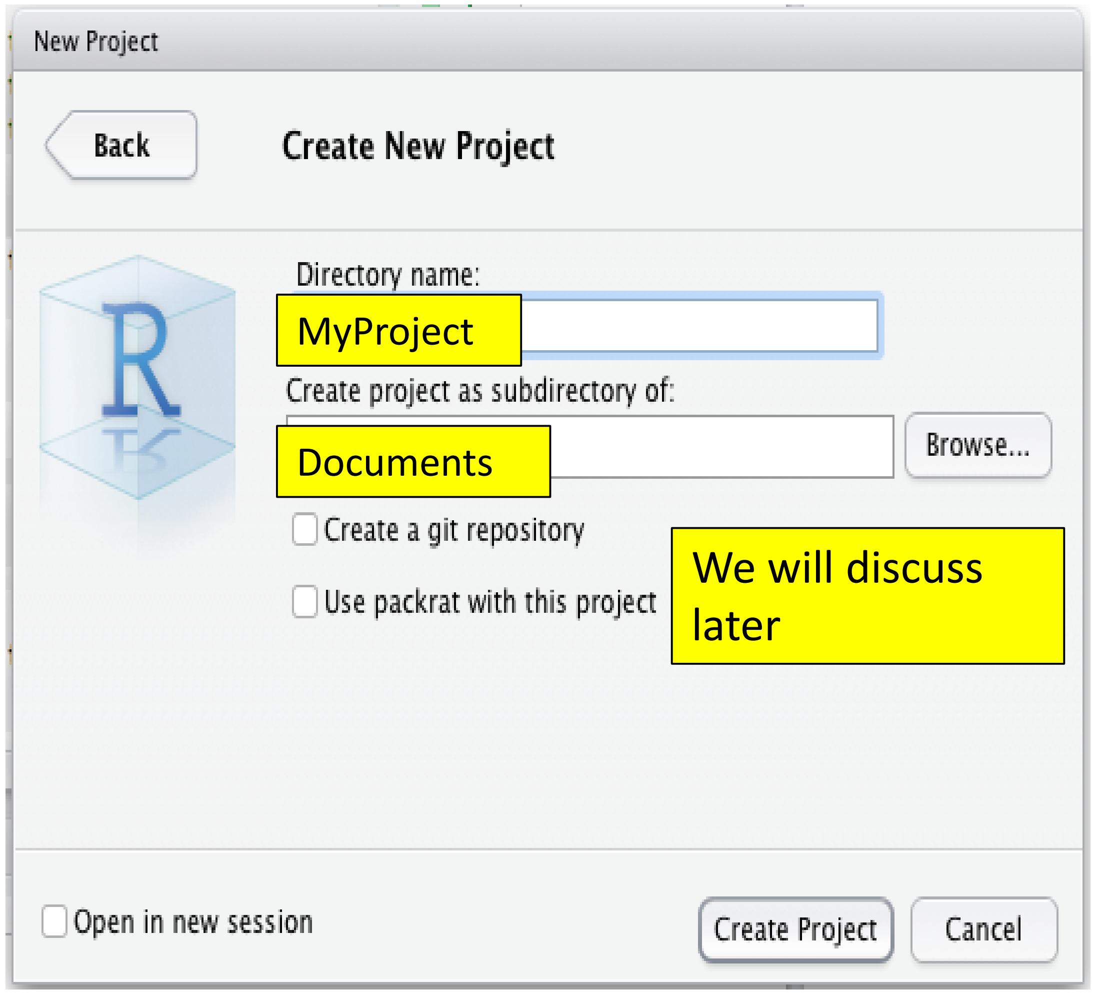
Project Type

Empty Project
Create a new project in an empty directory

R Package
Create a new R package

Shiny Web Application
Create a new Shiny web application

Cancel



New Project

Back

Create New Project

Directory name:

MyProject

Create project as subdirectory of:

Documents

Browse...

☐ Create a git repository

☐ Use packrat with this project

We will discuss later

☐ Open in new session

Create Project

Cancel

Working with Project:

- When a new project is created RStudio:

- Creates a project file (**with an .Rproj extension e.g. MyProject.Rproj**) within the project directory.

- This file contains various project options (discussed below) and can also be used as a shortcut for opening the project directly from the file system.

- Creates a **hidden directory** (named .Rproj.user) where project-specific temporary files (e.g. auto-saved source documents, window-state, etc.) are stored. This directory is also automatically added to .Rbuildignore, .gitignore, etc. if required.

- Loads the project into RStudio and display its name in the Projects toolbar (**which is located on the far right side of the main toolbar**)

Working with Project:

- Call your project “MyProject” and think carefully about which *subdirectory* you put the project in.
- If you don’t store it somewhere sensible, it will be hard to find it in the future!
- Once this process is complete, you’ll get a new RStudio project.
- Check that the “home” directory of your project is the current working directory with:
  - getwd()
  - "C:/Users/Dell/Documents/MyProject"

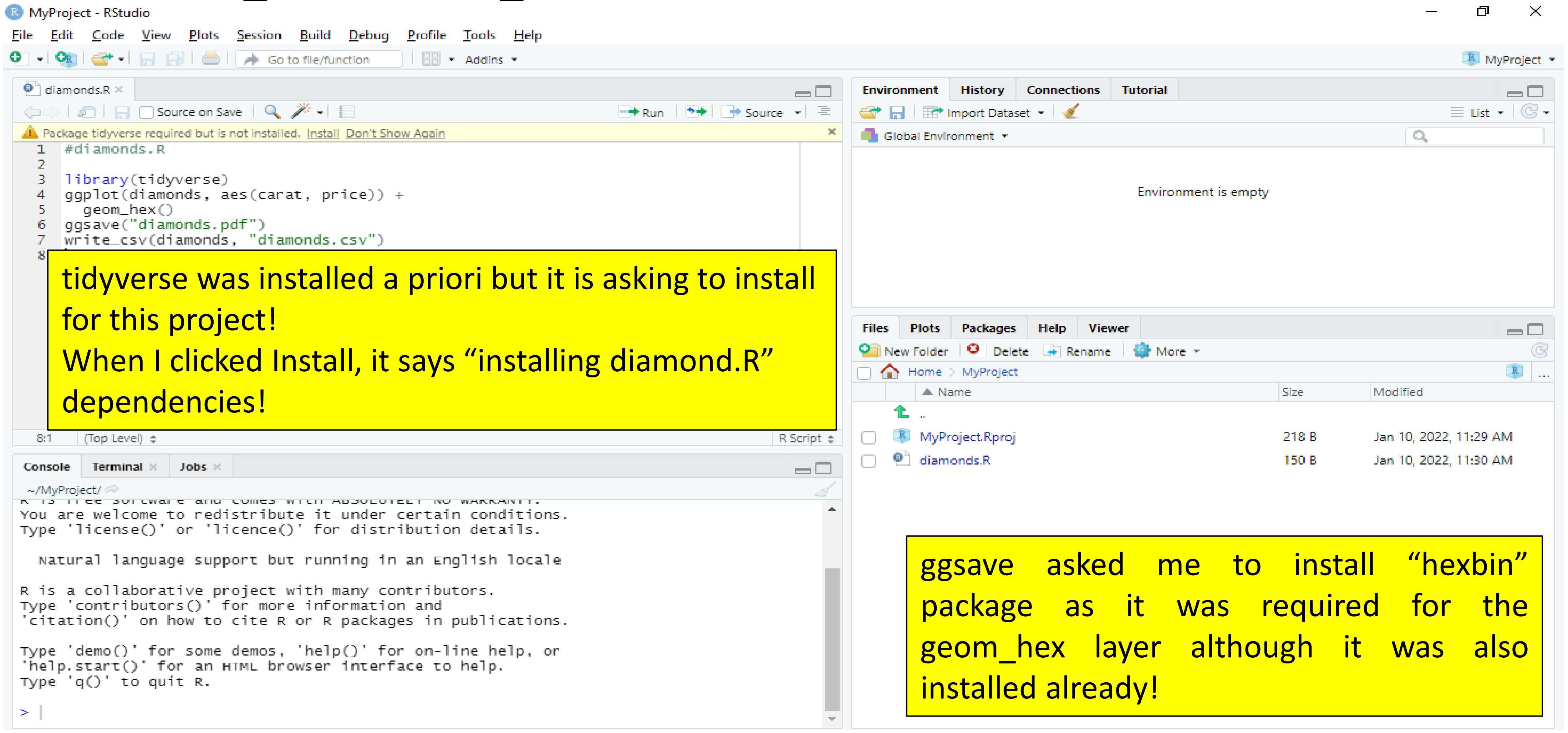
Working with Project:

- Now enter the following commands in the script editor, and **save the file**, calling it "diamonds.R" **inside/in the project directory** →
- Next, run the complete script which will save a PDF and CSV file into **your project directory**.
- Quit RStudio & Inspect the folder associated with your project — notice the MyProject**.Rproj** file!

```
#diamonds.R


library(tidyverse)
ggplot(diamonds, aes(carat, price)) +
geom_hex()
ggsave("diamonds.pdf")
write_csv(diamonds, "diamonds.csv")
```

I got this in Dell Optiplex 3010: Windows 10

You might not get error!



Package tidyverse required but is not installed. Install Don't Show Again

```
#diamonds.R

library(tidyverse)
ggplot(diamonds, aes(carat, price)) +
  geom_hex()
ggsave("diamonds.pdf")
write_csv(diamonds, "diamonds.csv")
```

tidyverse was installed a priori but it is asking to install for this project!
When I clicked Install, it says "installing diamond.R" dependencies!

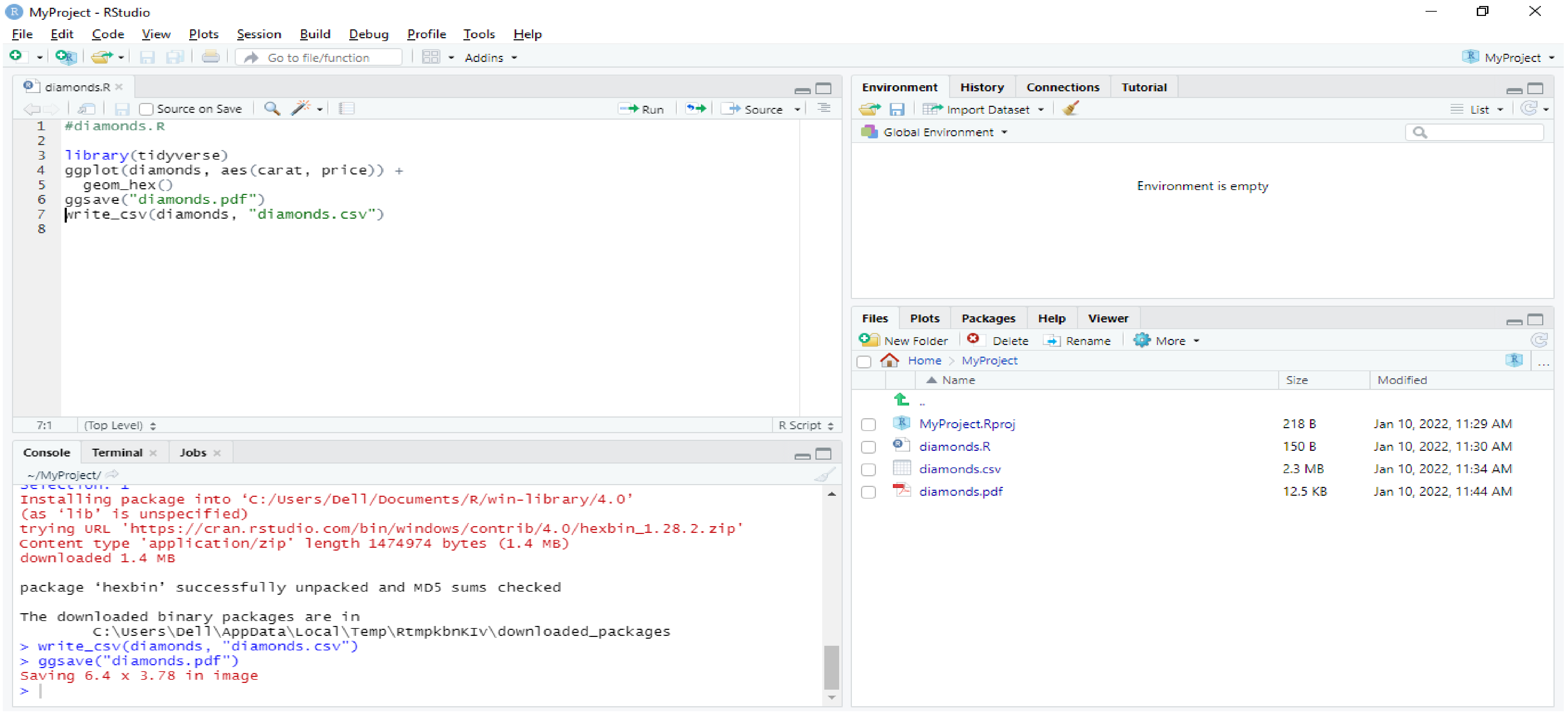
```
~/MyProject/
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

Environment is empty

Name	Size	Modified
..		
MyProject.Rproj	218 B	Jan 10, 2022, 11:29 AM
diamonds.R	150 B	Jan 10, 2022, 11:30 AM

ggsave asked me to install "hexbin" package as it was required for the geom_hex layer although it was also installed already!

Finally, I got this:
Make sure that pdf is not empty!



MyProject - RStudio

diamonds.R

```r
#diamonds.R

library(tidyverse)
ggplot(diamonds, aes(carat, price)) +
  geom_hex()
ggsave("diamonds.pdf")
write_csv(diamonds, "diamonds.csv")

```

Environment is empty

```
~/MyProject/
Installing package into ‘C:/Users/Dell/Documents/R/win-library/4.0’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/hexbin_1.28.2.zip'
Content type 'application/zip' length 1474974 bytes (1.4 MB)
downloaded 1.4 MB

package ‘hexbin’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Dell\AppData\Local\Temp\RtmpkbnKIv\downloaded_packages
> write_csv(diamonds, "diamonds.csv")
> ggsave("diamonds.pdf")
Saving 6.4 x 3.78 in image
> 
```

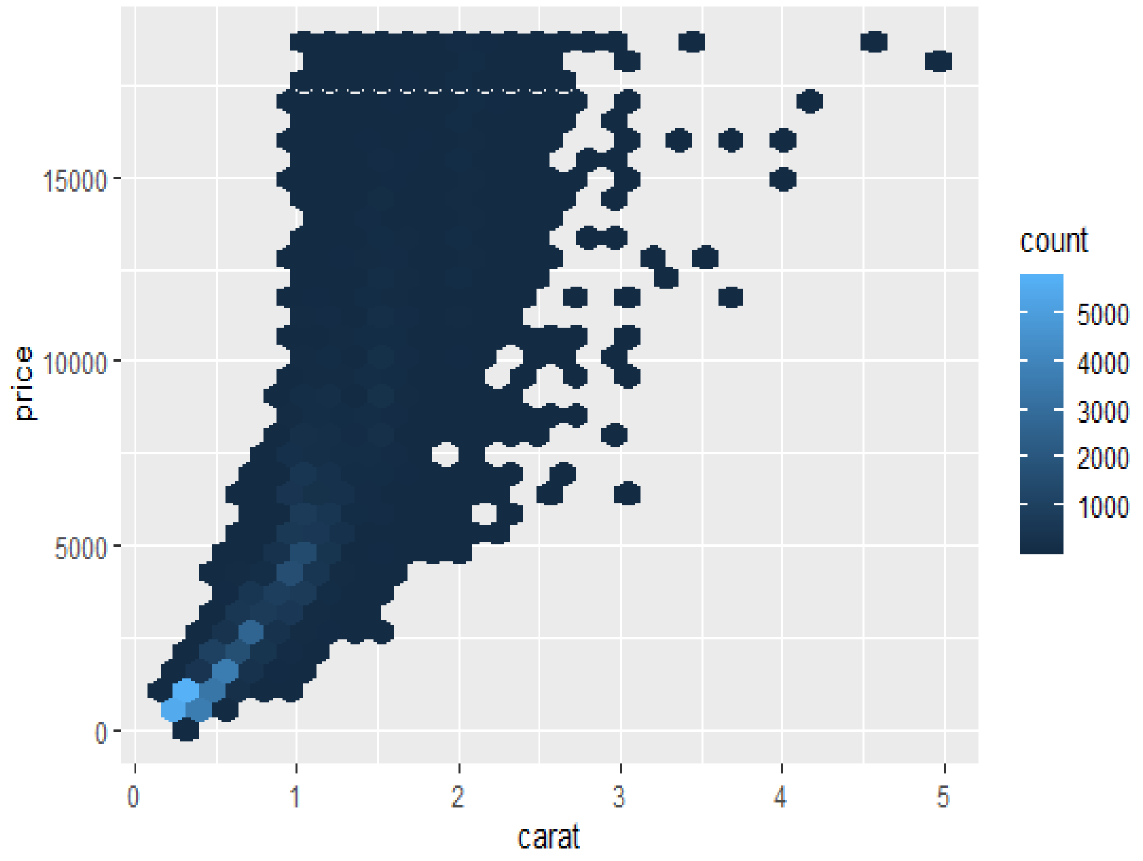
Home > MyProject

Name	Size	Modified
..		
MyProject.Rproj	218 B	Jan 10, 2022, 11:29 AM
diamonds.R	150 B	Jan 10, 2022, 11:30 AM
diamonds.csv	2.3 MB	Jan 10, 2022, 11:34 AM
diamonds.pdf	12.5 KB	Jan 10, 2022, 11:44 AM

The plot: Saved as PDF and also seen in the “plots” tab

#It was produced due to this code:

- library(tidyverse)
- ggplot(diamonds, aes(carat, price)) +
- geom_hex()



Working with Project:

# Opening Projects

- There are several ways to open a project:
- Using the **Open Project** command (available from both the Projects menu and the Projects toolbar) to browse for and select an existing project file
  - e.g. MyProject.Rproj
- Selecting a project from the list of most recently opened projects (also available from both the Projects menu and toolbar).
- Double-clicking on the project file within the system shell (e.g. Windows Explorer, OSX Finder, etc.).

Working with Project:

- When a project is opened within RStudio the following actions are taken:

- A new R session (process) is started

- The **.Rprofile** file in the project's main directory (if any) is sourced by R

- The **.RData** file in the project's main directory is loaded (if project options indicate that it should be loaded).

- The **.Rhistory file in the project's main directory is loaded** into the RStudio History pane (and used for Console Up/Down arrow command history).

- The current working directory is set to the project directory.

- Previously edited source documents are restored into editor tabs

- Other **RStudio settings** (e.g. active tabs, splitter positions, etc.) **are restored to where they were the last time** the project was closed.

Working with Project:

# Quitting a Project

- When you are within a project and choose to either Quit, close the project, or open another project the following actions are taken:
- **.RData and/or .Rhistory are written to the project directory** (if current options indicate they should be)
- The list of open source documents is saved (so it can be restored next time the project is opened)
- Other RStudio settings (as described above) are saved.
- The R session is terminated.

Working with Project:

# Working with Multiple Projects at Once

- You can work with more than one RStudio project at a time by simply opening each project in its own instance of RStudio.

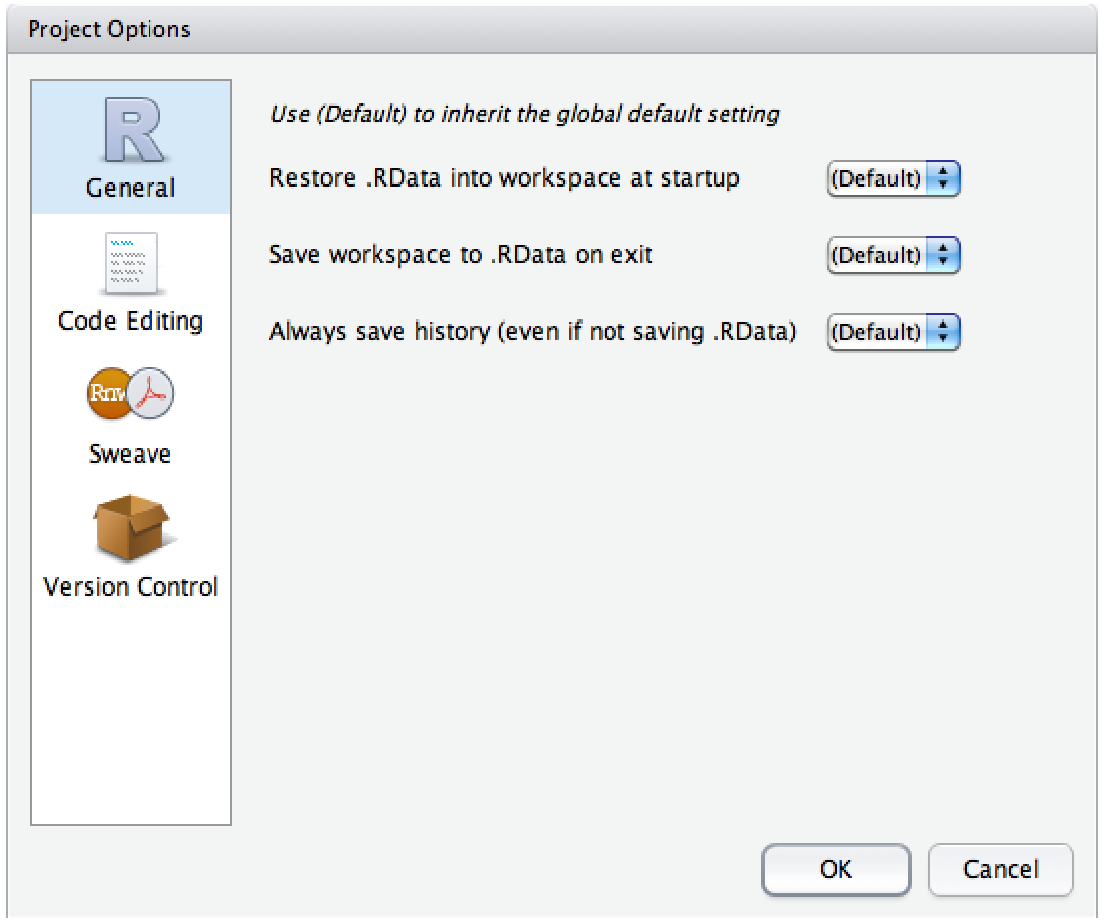
- There are two ways to accomplish this:

- Use the **Open Project in New Window** command located on the Project menu.

- Opening **multiple project files via the system shell** (i.e. double-clicking on the project file).

Working with Project:

- Tools → Project Options
- There are several options that can be set on a per-project basis to customize the behavior of RStudio. You can edit these options using the **Project Options** command on the **Project** menu:



Use (Default) to inherit the global default setting

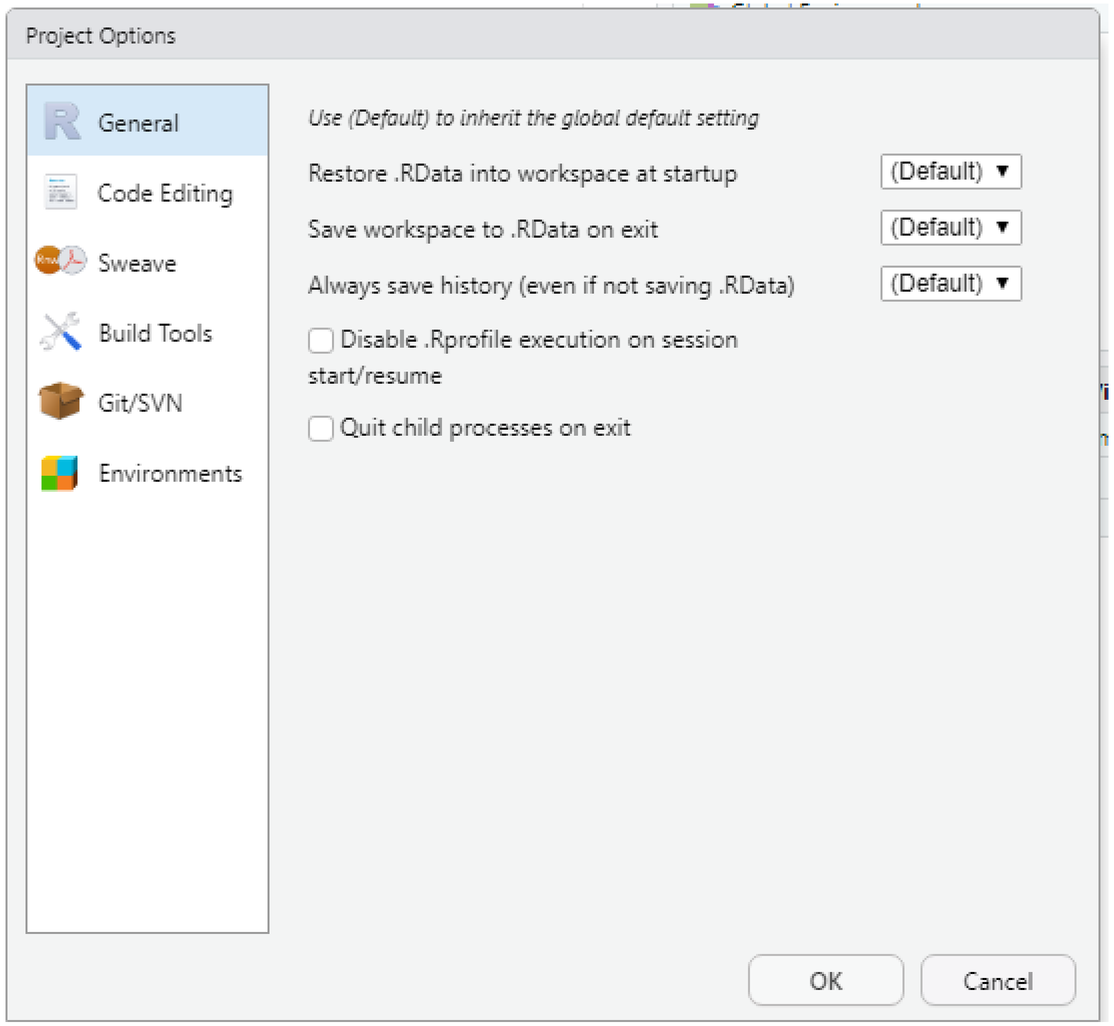
Restore .RData into workspace at startup (Default)

Save workspace to .RData on exit (Default)

Always save history (even if not saving .RData) (Default)

You can explore each of the options seen:

- **General *(Do not change!)***
- Code Editing
- Sweave
- Build Tools
- **Git/SVN**
- Environments
  - packrat would appear if it was selected while creating the project



Use (Default) to inherit the global default setting

Restore .RData into workspace at startup (Default)

Save workspace to .RData on exit (Default)

Always save history (even if not saving .RData) (Default)

☐ Disable .Rprofile execution on session start/resume

☐ Quit child processes on exit

Working with Project: Code editing

- **Index R source files** — Determines whether R source files within the project directory are indexed for code navigation (i.e. go to file/function, go to function definition).
- Normally this should remain enabled, however if you have a project directory with thousands of files and are concerned about the overhead of monitoring and indexing them you can disable indexing here.
- **Insert spaces for tab** — Determine whether the tab key inserts multiple spaces rather than a tab character (soft tabs). Configure the number of spaces per soft-tab.
- **Text encoding** — Specify the default text encoding for source files.
- Note that source files which don't match the default encoding can still be opened correctly using the **File : Reopen with Encoding** menu command.

Working with Project: Sweave

#Project options: Sweave

PDF Generation

- Weave Rnw (or Sweave) file using: Sweave
- Typeset LaTeX into PDF using: pdfLaTeX

PDF Preview

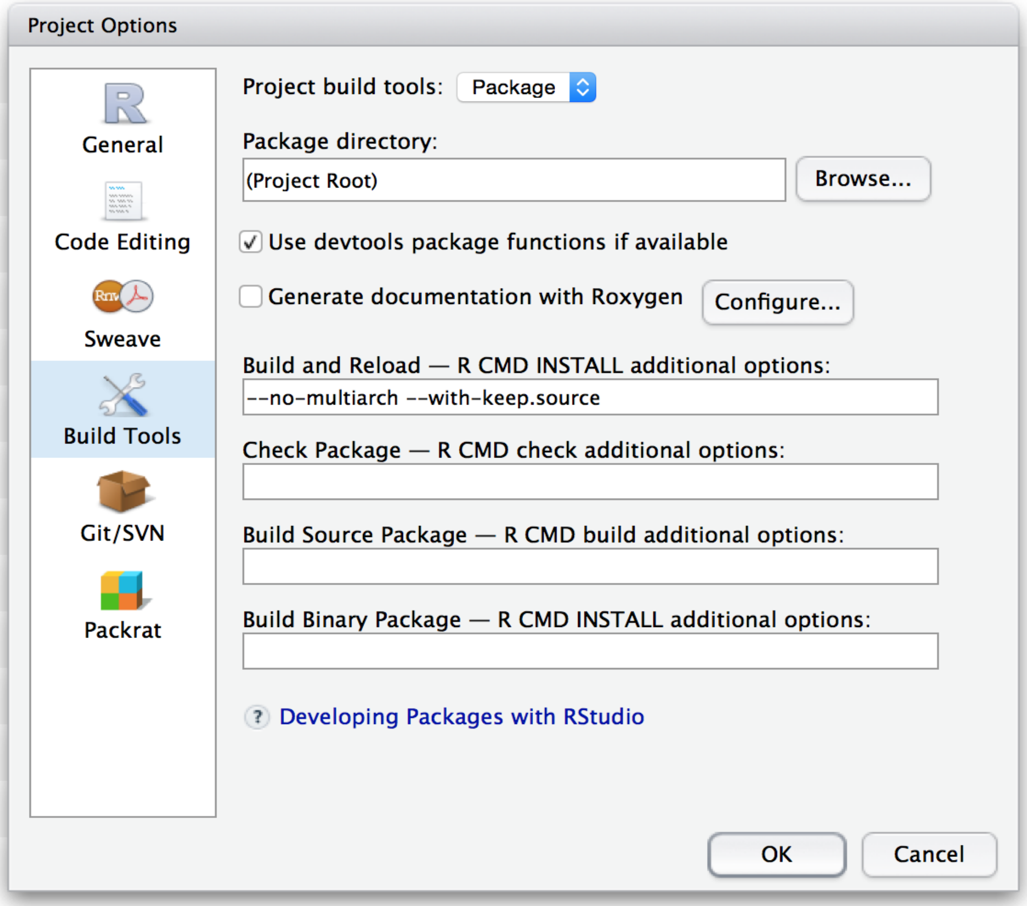
- Compile PDF root document:

- More here: https://support.posit.co/hc/en-us/articles/200532257
- https://support.posit.co/hc/en-us/articles/200532247
- https://support.posit.co/hc/en-us/articles/200486298

Working with Project: Build Tool (package)

#Project options: Build Tools

- There are three R package build commands used by the package development tools in the RStudio IDE:
  - R CMD check
  - R CMD build
  - R CMD INSTALL
- It's possible to customize the options passed to each of these commands using **Project Options : Build Tools**



Project build tools: Package

Package directory: (Project Root)

[✓] Use devtools package functions if available

[ ] Generate documentation with Roxygen

Build and Reload — R CMD INSTALL additional options: --no-multiarch --with-keep.source

Check Package — R CMD check additional options:

Build Source Package — R CMD build additional options:

Build Binary Package — R CMD INSTALL additional options:

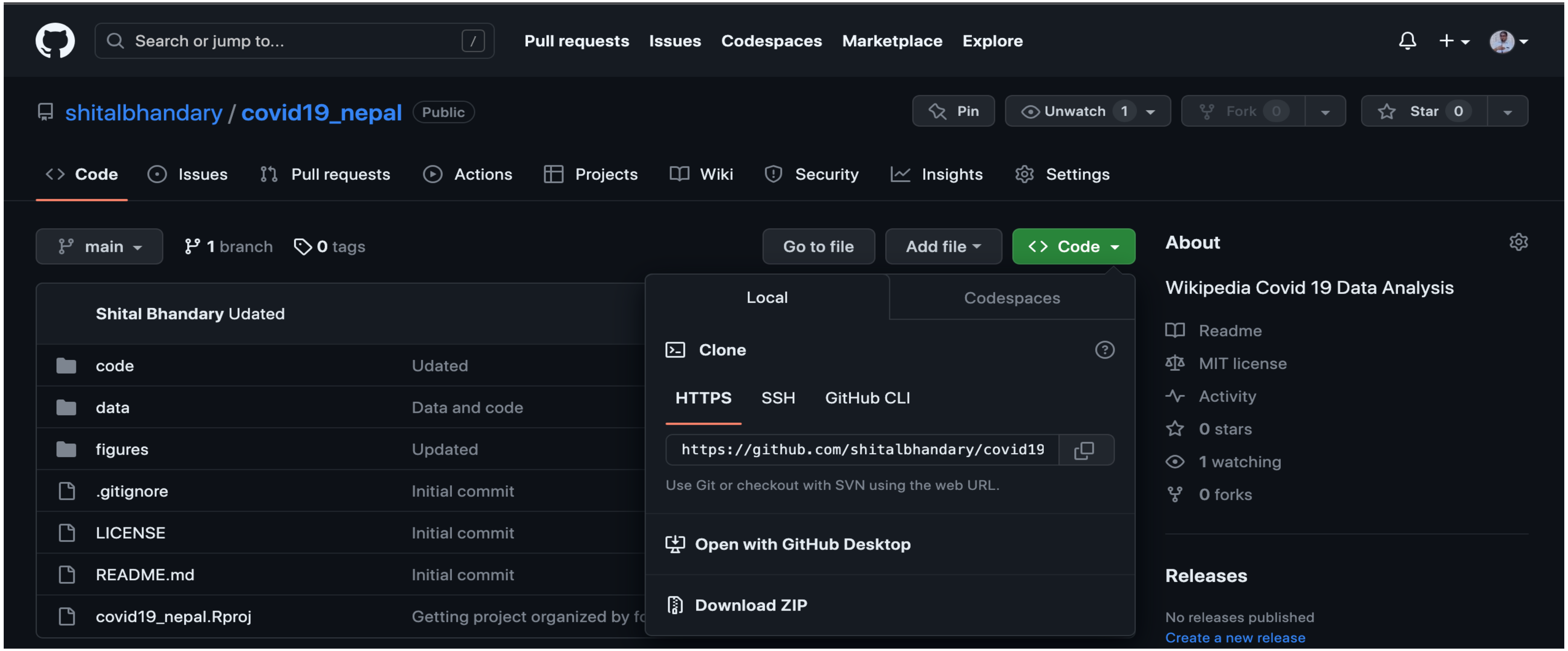
More here: https://support.posit.co/hc/en-us/articles/200486518-Customizing-Package-Build-Options-in-the-RStudio-IDE

Working with Project: Version control

- **Version control system -** Specify the version control system to use with this project.
- Note that **RStudio automatically detects** the presence of version control for projects by scanning for a .git or .svn directory.
- **Therefore it isn't normally necessary to change this setting.**
- You may want to change the setting for the following reasons:
  - You have both a .git and .svn directory within the project and wish to specify **which version control system RStudio should bind** to.
  - You have no version control setup for the project and you **want to add a local git repository** (equivalent to executing git init from project root directory)
- **Origin** — Read-only display of the remote origin (if any) for the project version control repository

More here: https://support.rstudio.com/hc/en-us/articles/200532077?version=1.3.1093&mode=desktop

Github account and a repository: Example - **Getting a public repo in R studio!**



Shital Bhandary Udated

code	Udated
data	Data and code
figures	Updated
.gitignore	Initial commit
LICENSE	Initial commit
README.md	Initial commit
covid19_nepal.Rproj	Getting project organized by f

```
https://github.com/shitalbhandary/covid19
```

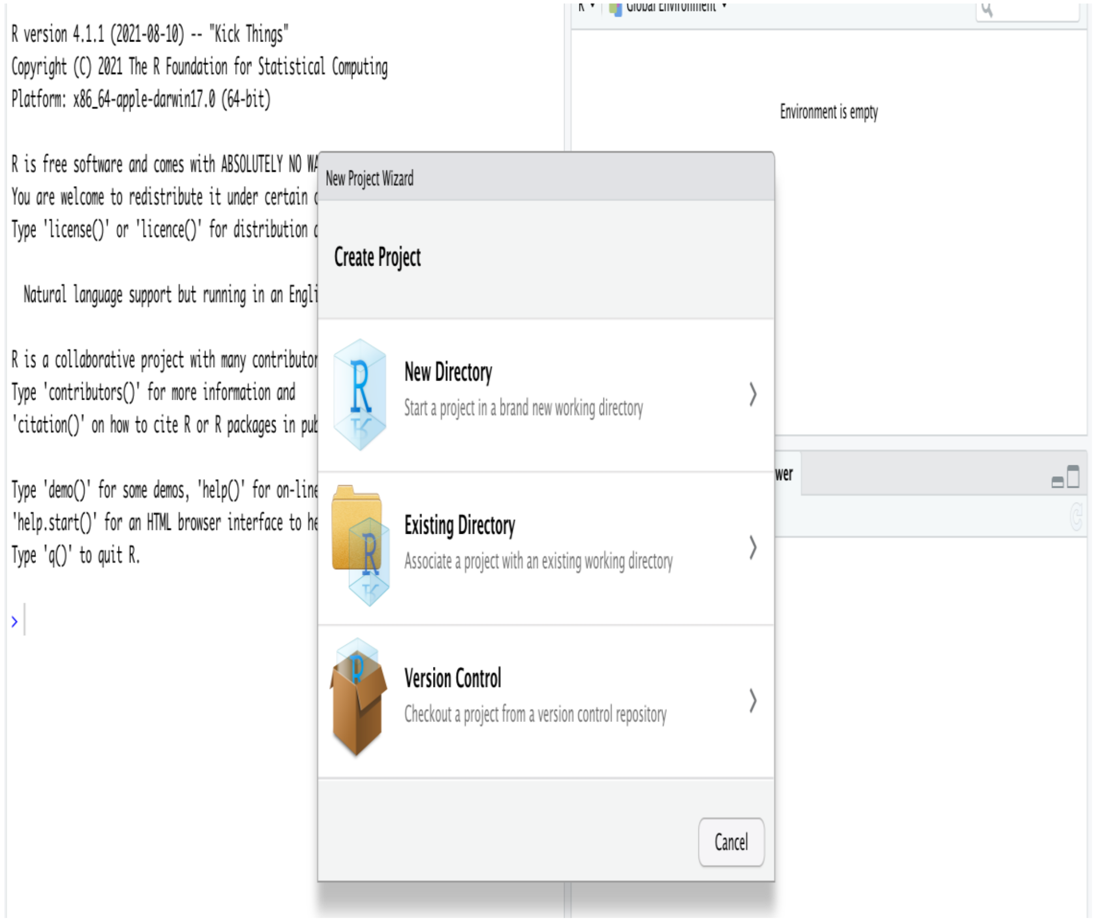
Use Git or checkout with SVN using the web URL.

Wikipedia Covid 19 Data Analysis

No releases published

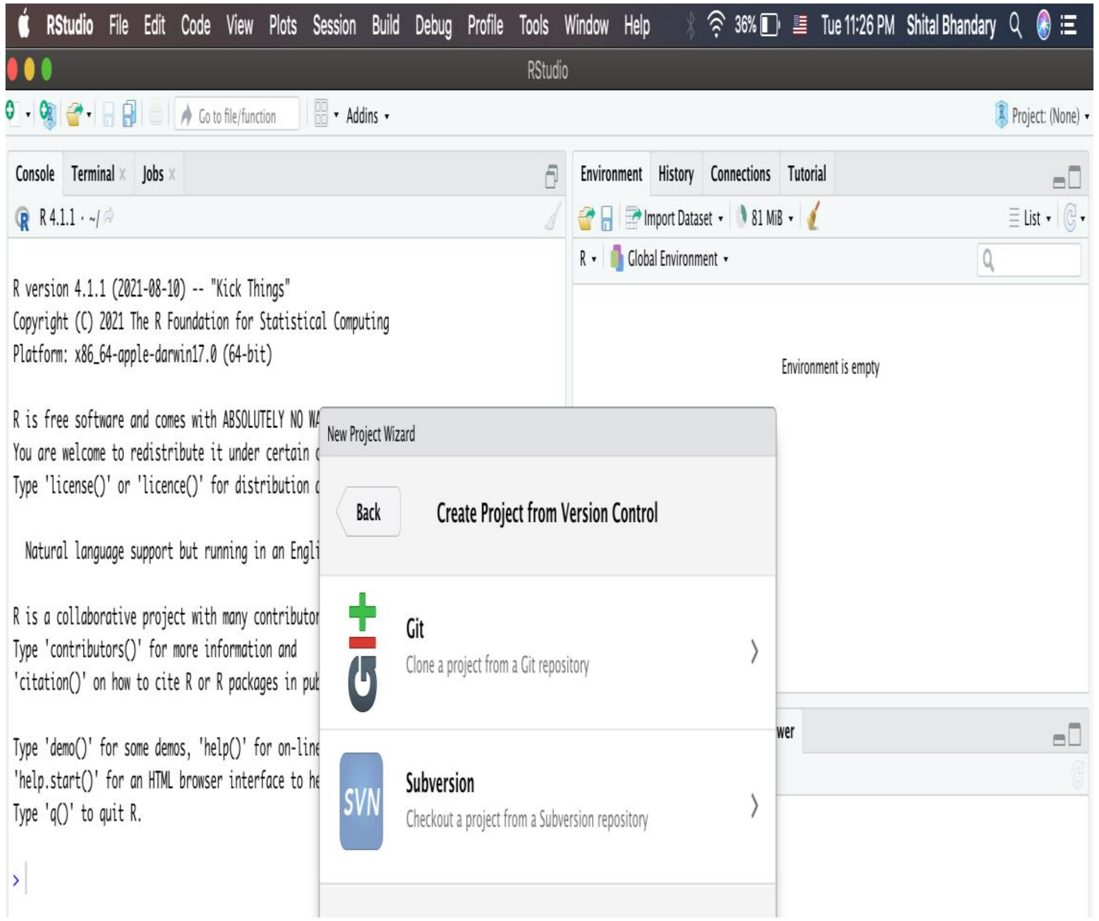
R Studio and Github: Working together!

Create new project → Version Control → Git



```
R version 4.1.1 (2021-08-10) -- "Kick Things"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WA
You are welcome to redistribute it under certain c
Type 'license()' or 'licence()' for distribution d

  Natural language support but running in an Engli

R is a collaborative project with many contributor
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in pub

Type 'demo()' for some demos, 'help()' for on-line
'help.start()' for an HTML browser interface to he
Type 'q()' to quit R.

>
```

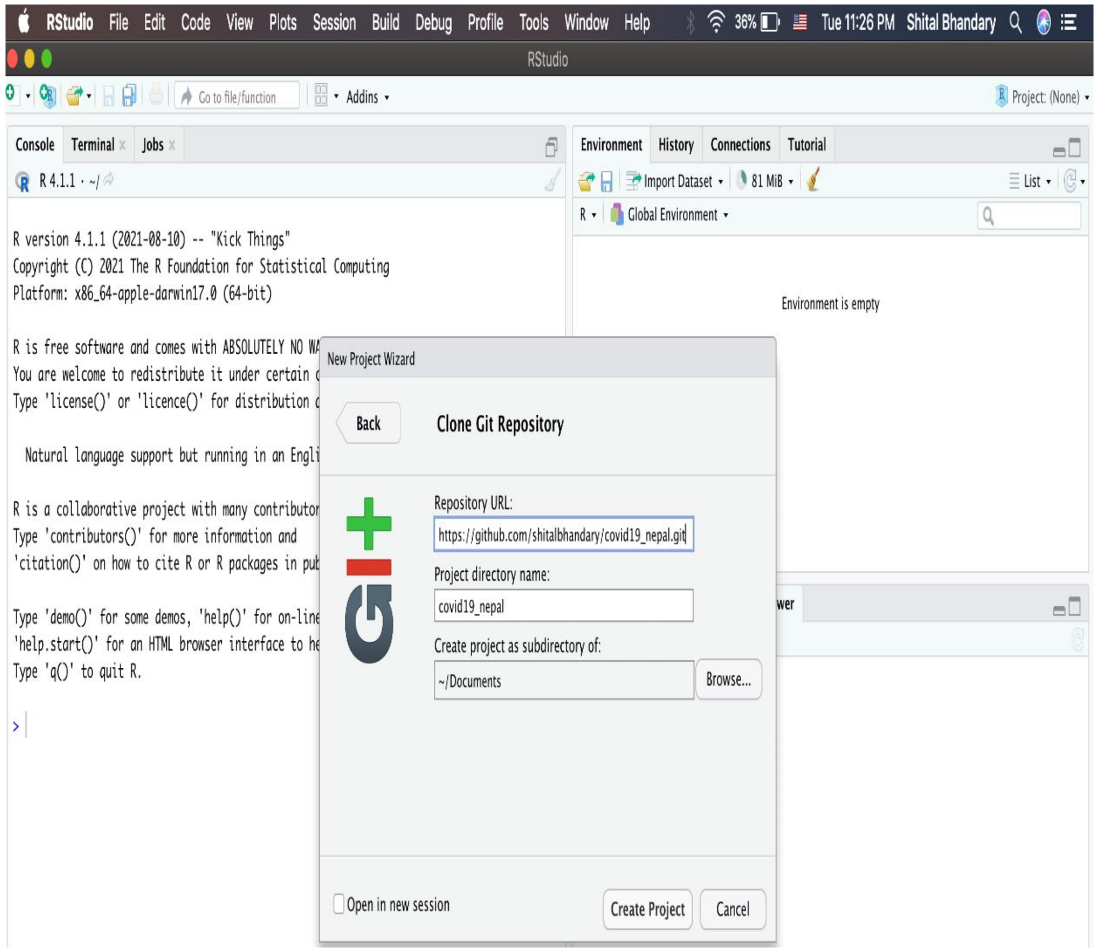
Environment is empty
New Project Wizard
Create Project
New Directory
Start a project in a brand new working directory
Existing Directory
Associate a project with an existing working directory
Version Control
Checkout a project from a version control repository
Cancel



```
R version 4.1.1 (2021-08-10) -- "Kick Things"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WA
You are welcome to redistribute it under certain c
Type 'license()' or 'licence()' for distribution d

  Natural language support but running in an Engli

R is a collaborative project with many contributor
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in pub

Type 'demo()' for some demos, 'help()' for on-line
'help.start()' for an HTML browser interface to he
Type 'q()' to quit R.

>
```

RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
36%
Tue 11:26 PM Shital Bhandary
RStudio
Go to file/function
Addins
Project: (None)
Console Terminal Jobs
R 4.1.1 · ~/
Environment History Connections Tutorial
Import Dataset
81 MiB
List
R
Global Environment
Environment is empty
New Project Wizard
Back
Create Project from Version Control
Git
Clone a project from a Git repository
SVN
Subversion
Checkout a project from a Subversion repository

R Studio and Github: Git → Copied link of the github repository from the github web

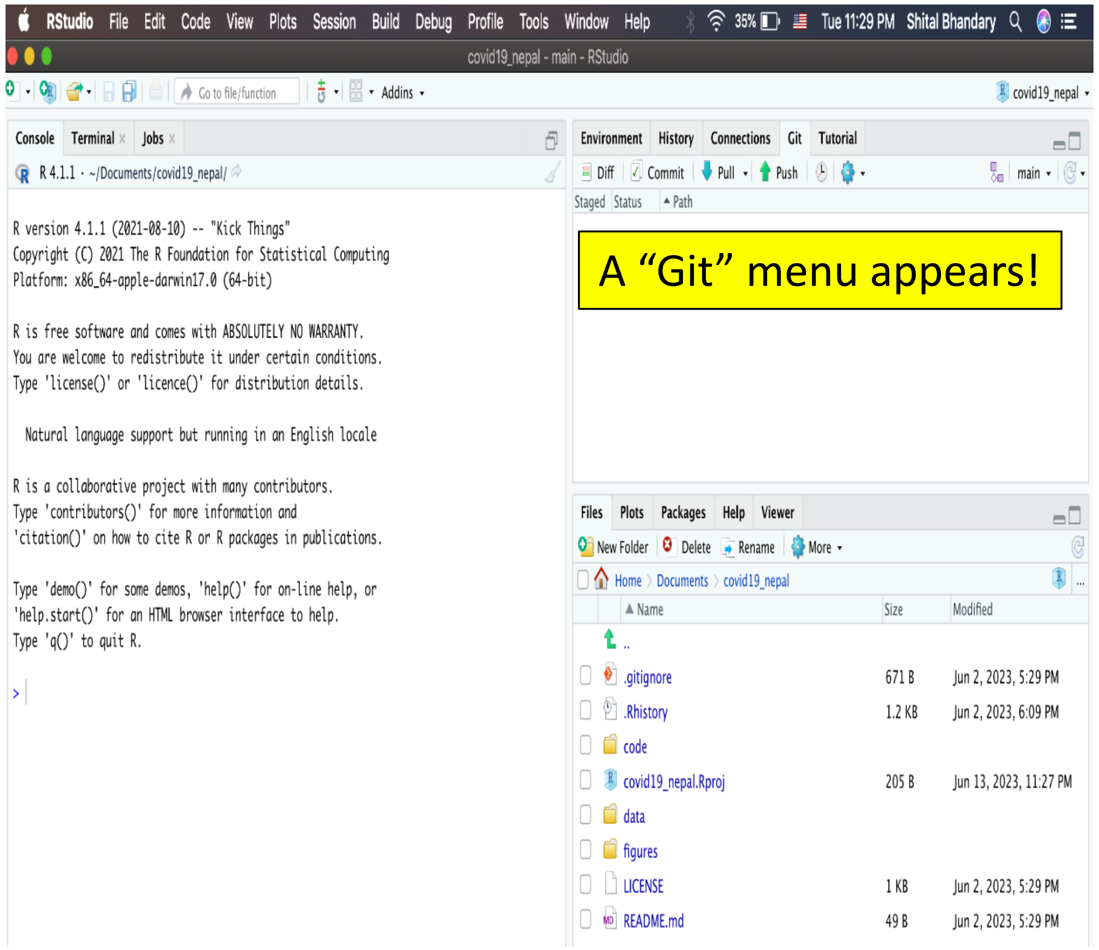


```
R version 4.1.1 (2021-08-10) -- "Kick Things"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WA
You are welcome to redistribute it under certain c
Type 'license()' or 'licence()' for distribution d

  Natural language support but running in an Engli

R is a collaborative project with many contributor
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in pub

Type 'demo()' for some demos, 'help()' for on-line
'help.start()' for an HTML browser interface to he
Type 'q()' to quit R.

>
```

Repository URL: https://github.com/shitalbhandary/covid19_nepal.git

Project directory name: covid19_nepal

Create project as subdirectory of: ~/Documents


RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
35% Tue 11:29 PM Shital Bhandary
covid19_nepal - main - RStudio
Go to file/function
Addins
covid19_nepal
Console Terminal Jobs
R 4.1.1 · ~/Documents/covid19_nepal/
Environment History Connections Git Tutorial
Diff Commit Pull Push
main
Staged Status Path
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home › Documents › covid19_nepal

```
R version 4.1.1 (2021-08-10) -- "Kick Things"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (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.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

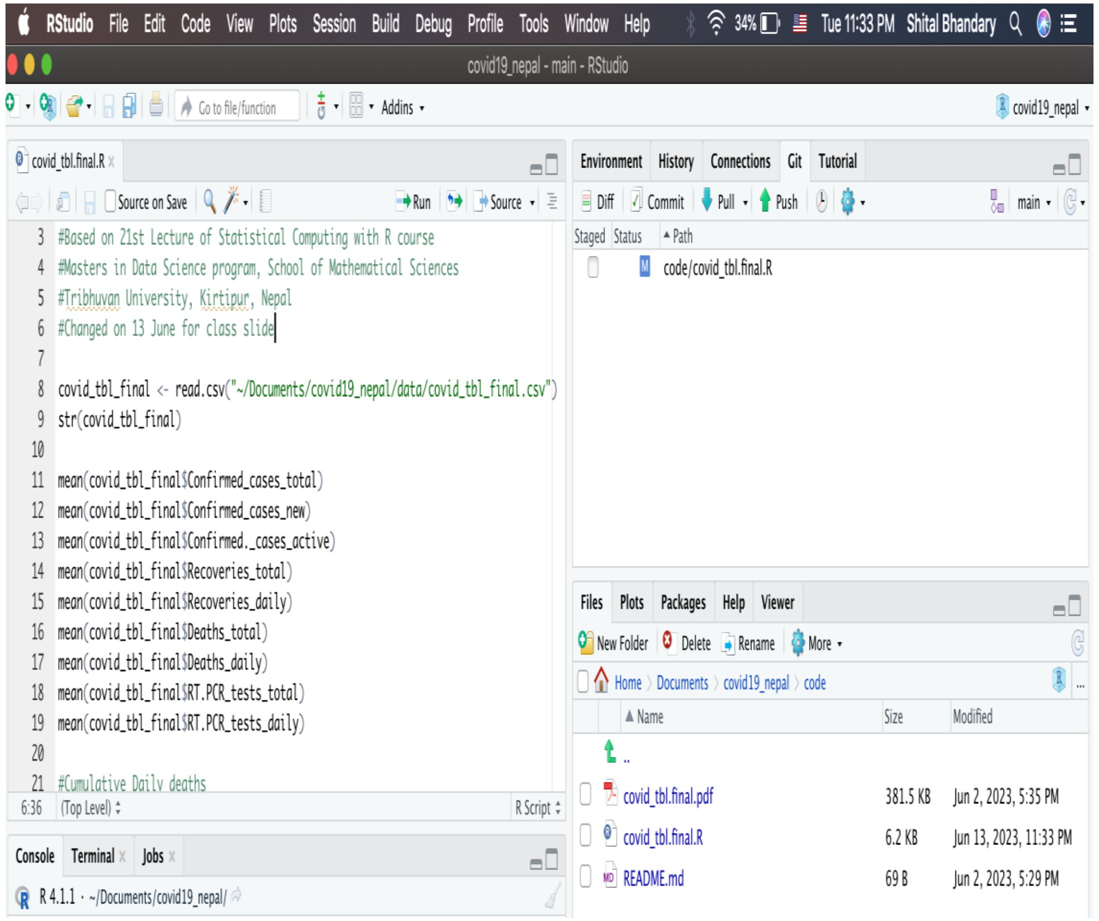
A “Git” menu appears!

Name	Size	Modified
..		
.gitignore	671 B	Jun 2, 2023, 5:29 PM
.Rhistory	1.2 KB	Jun 2, 2023, 6:09 PM
code		
covid19_nepal.Rproj	205 B	Jun 13, 2023, 11:27 PM
data		
figures		
LICENSE	1 KB	Jun 2, 2023, 5:29 PM
README.md	49 B	Jun 2, 2023, 5:29 PM

We can work with this repository in R Studio:

- When we make changes (**in line 6**) and hit save button then the changed file appears in the Git environment e.g. changes in the code/covid_tbl.final.R file is shown there →

- We need to check the highlighted change, hit the “commit” button and then type a commit message as a simple text e.g. “code updated” (see next slide)


RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
34% Tue 11:33 PM Shital Bhandary
covid19_nepal - main - RStudio
Go to file/function
Addins
covid19_nepal
covid_tbl.final.R
Source on Save
Run
Source
Environment History Connections Git Tutorial
Diff Commit Pull Push
main
Staged Status Path
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home › Documents › covid19_nepal › code
Console Terminal Jobs
R 4.1.1 · ~/Documents/covid19_nepal/

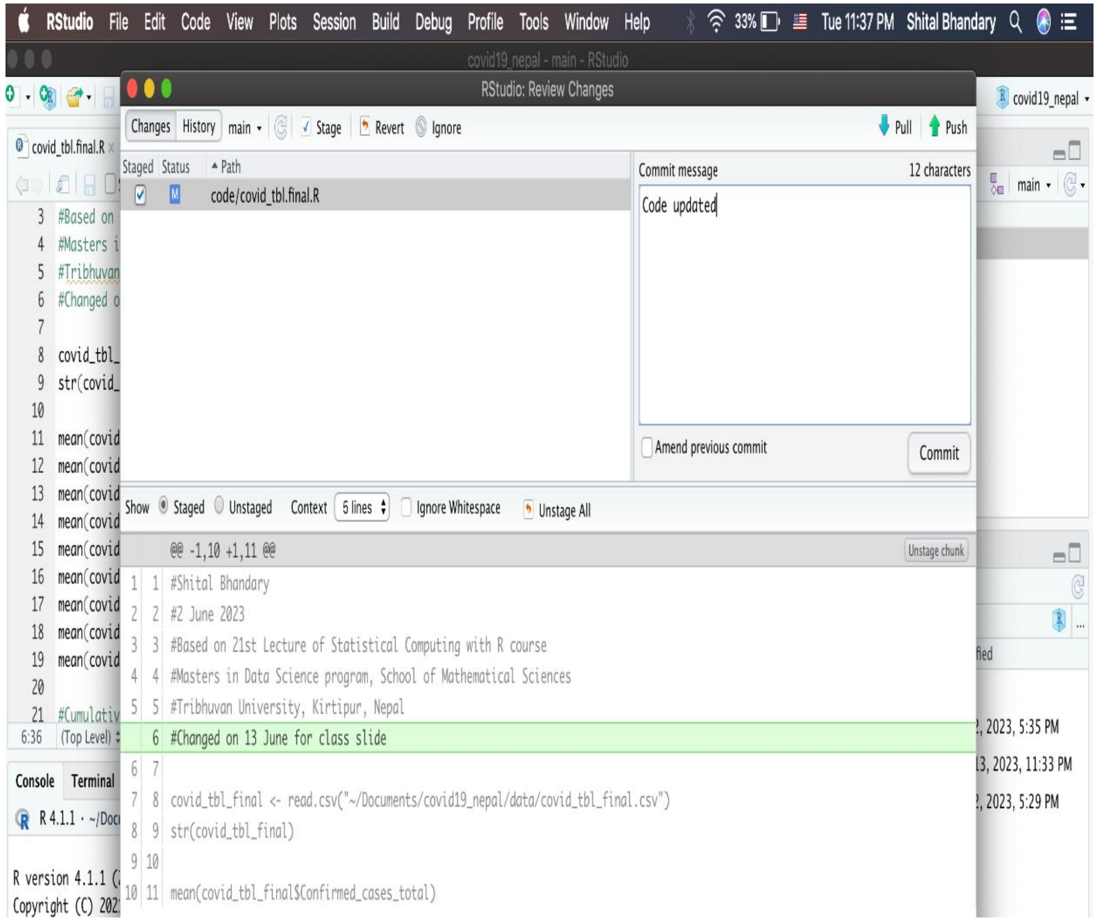
```
3  #Based on 21st Lecture of Statistical Computing with R course
4  #Masters in Data Science program, School of Mathematical Sciences
5  #Tribhuvan University, Kirtipur, Nepal
6  #Changed on 13 June for class slide
7
8  covid_tbl_final <- read.csv("~/Documents/covid19_nepal/data/covid_tbl_final.csv")
9  str(covid_tbl_final)
10
11 mean(covid_tbl_final$Confirmed_cases_total)
12 mean(covid_tbl_final$Confirmed_cases_new)
13 mean(covid_tbl_final$Confirmed._cases_active)
14 mean(covid_tbl_final$Recoveries_total)
15 mean(covid_tbl_final$Recoveries_daily)
16 mean(covid_tbl_final$Deaths_total)
17 mean(covid_tbl_final$Deaths_daily)
18 mean(covid_tbl_final$RT.PCR_tests_total)
19 mean(covid_tbl_final$RT.PCR_tests_daily)
20
21 #Cumulative Daily deaths
```

6:36 (Top Level) R Script

Staged	Status	Path
☐	M	code/covid_tbl.final.R

Name	Size	Modified
..		
covid_tbl.final.pdf	381.5 KB	Jun 2, 2023, 5:35 PM
covid_tbl.final.R	6.2 KB	Jun 13, 2023, 11:33 PM
README.md	69 B	Jun 2, 2023, 5:29 PM

Once the “commit” is done then we need to “push” it to the github repo to save the changes!

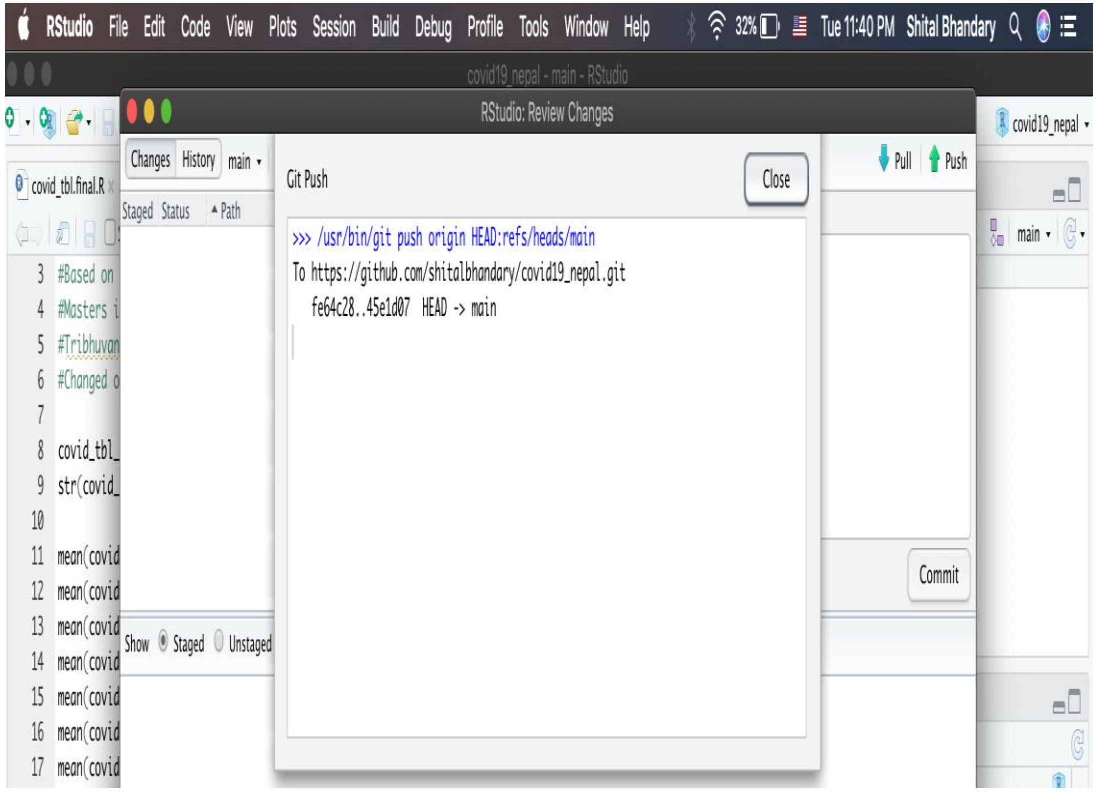


Staged: ✓ M code/covid_tbl_final.R

Commit message (12 characters):

```
Code updated
```

```
@@ -1,10 +1,11 @@
#Shital Bhandary
#2 June 2023
#Based on 21st Lecture of Statistical Computing with R course
#Masters in Data Science program, School of Mathematical Sciences
#Tribhuvan University, Kirtipur, Nepal
#Changed on 13 June for class slide

covid_tbl_final <- read.csv("~/Documents/covid19_nepal/data/covid_tbl_final.csv")
str(covid_tbl_final)

mean(covid_tbl_final$Confirmed_cases_total)
```



```
>>> /usr/bin/git push origin HEAD:refs/heads/main
To https://github.com/shitalbhandary/covid19_nepal.git
   fe64c28..45e1d07  HEAD -> main
```

This type of message will be seen if all goes well!

Working with Project: Environment

- Project options: Environments

- Rstudio uses the "renv" package to give your projects their own privately managed library, making your R code more isolated, portable and reproducible
- Click/check: "Use "renv" with this project" to use this feature!

More here: https://rstudio.github.io/renv/

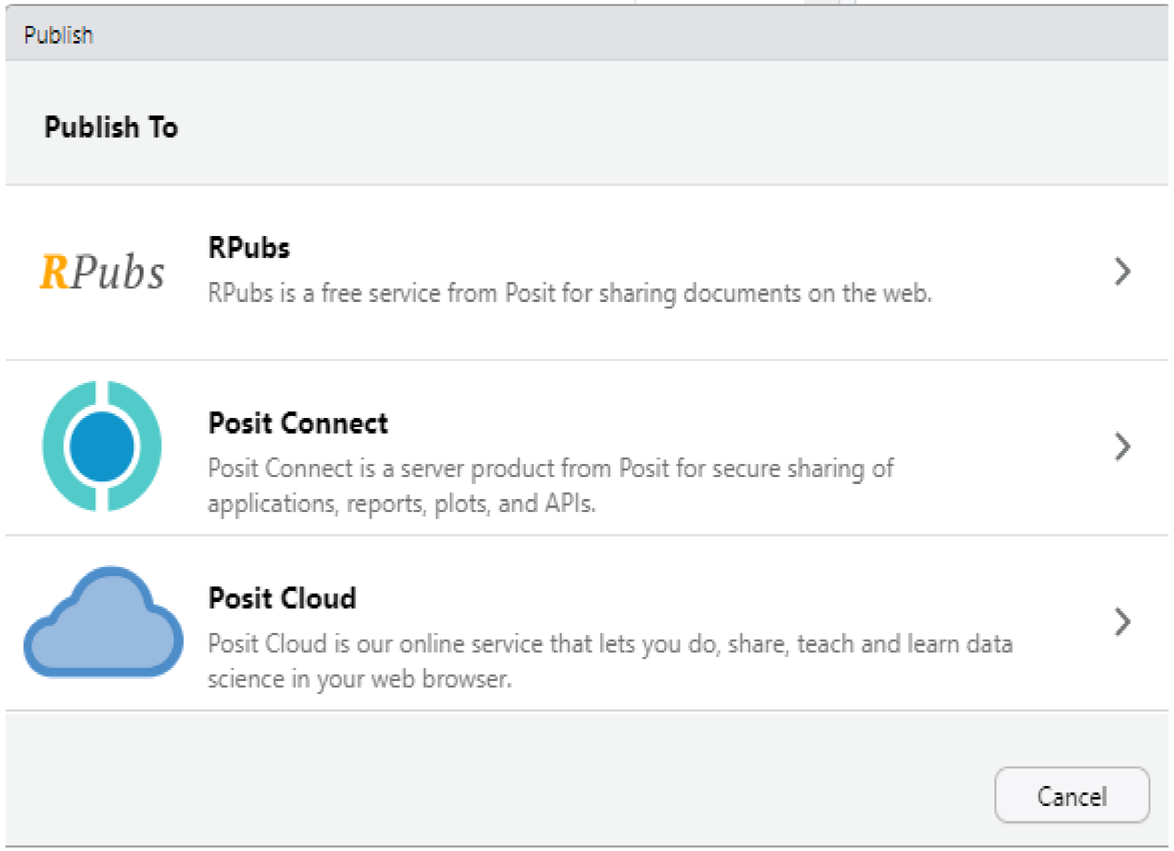
#How to use:

- Use **renv::init()** to initialize renv with a new or existing project.
- This will set up your project with a private library, and also make sure to install all of the packages you're using into that library.
- The packages used in your project will be recorded into a *lockfile*, called renv.lock

Alternatively, we can use "packrat" too (define while creating project): https://rstudio.github.io/packrat/rstudio.html

Working with Project: Publishing your work!

- Publishing with R studio:
- Markdown/R notebook files using:
- **Rpubs (free!)**
- R Studio/Posit connect



RPubs is a free service from Posit for sharing documents on the web.

Posit Connect is a server product from Posit for secure sharing of applications, reports, plots, and APIs.

Posit Cloud is our online service that lets you do, share, teach and learn data science in your web browser.

More here: https://support.posit.co/hc/en-us/articles/228270928-Push-button-publishing-to-RStudio-Connect

More here: https://posit.co/products/enterprise/connect/

R Shiny web app intro: https://shiny.posit.co/ Details here: https://mastering-shiny.org/index.html

- We need to define
  - ui
  - server
  - shinyApp(ui, server)
- to get an interactive web app of data science projects for "free"
- First Shiny app: https://mastering-shiny.org/basic-app.html

# A complex example from posit:

- app.R (R Script) that contains:

-
  ```
  library(shiny)
  library(bslib)
  library(dplyr)
  library(ggplot2)
  ```

Getting the data and defining user interface (ui)

Data: https://raw.githubusercontent.com/jcheng5/simplepenguins.R/main/penguins.csv

- `User interface – defined`

- `ui <- page_fillable(theme = bs_theme(bootswatch = "minty"), layout_sidebar(fillable = TRUE, sidebar( varSelectInput("xvar", "X variable", df_num, selected = "Bill Length (mm)"), varSelectInput("yvar", "Y variable", df_num, selected = "Bill Depth (mm)"),`

- `checkboxGroupInput("species", "Filter by species", choices = unique(df$Species), selected = unique(df$Species)), hr(), # Add a horizontal rule checkboxInput("by_species", "Show species", TRUE), checkboxInput("show_margins", "Show marginal plots", TRUE), checkboxInput("smooth", "Add smoother"),), plotOutput("scatter")))`

R Shiny app server and output parameters:

- `server <- function(input, output, session) { subsetted <- reactive({ req(input$species) df |> filter(Species %in% input$species) })`

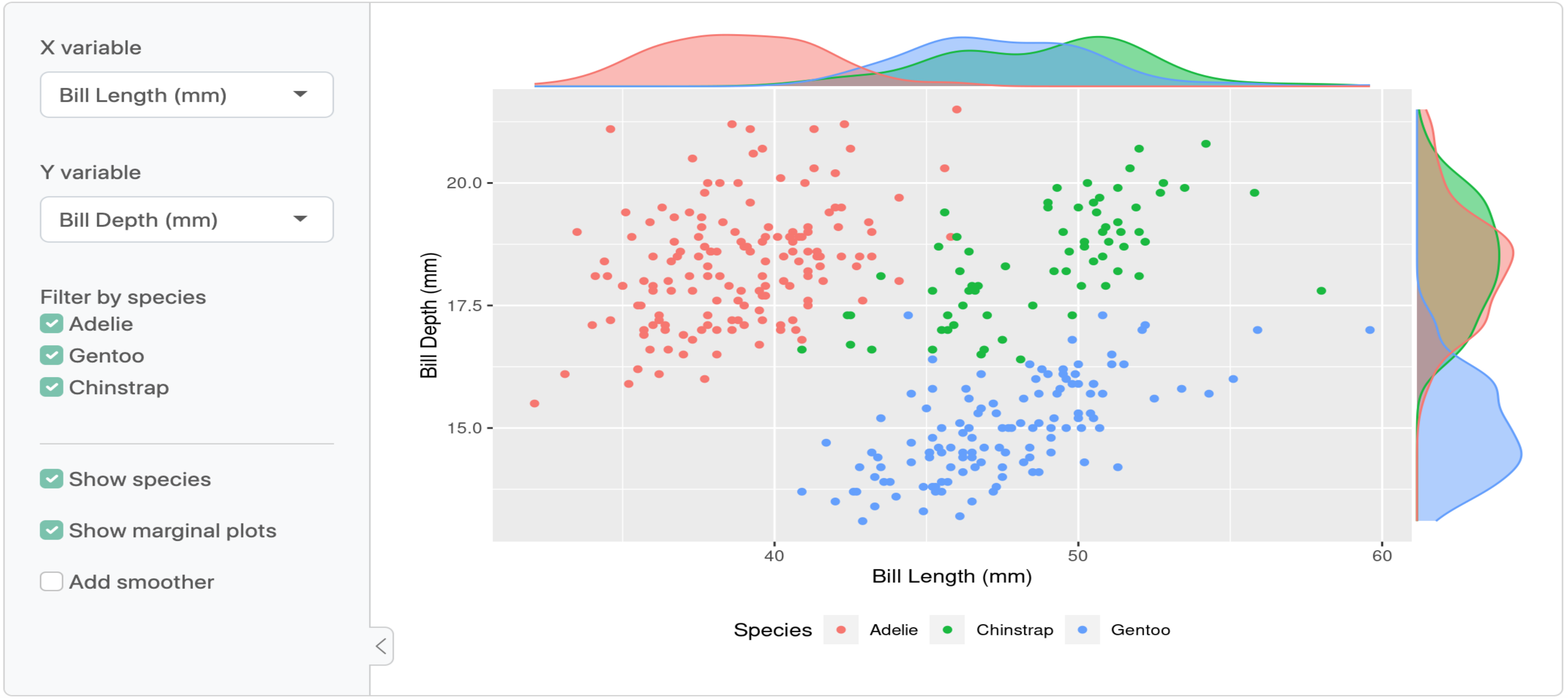
- `output$scatter <- renderPlot({ p <- ggplot(subsetted(), aes(!!input$xvar, !!input$yvar)) + list( theme(legend.position = "bottom"), if (input$by_species) aes(color=Species), geom_point(), if (input$smooth) geom_smooth() )`

- `if (input$show_margins) { margin_type <- if (input$by_species) "density" else "histogram" p <- p |> ggExtra::ggMarginal(type = margin_type, margins = "both", size = 8, groupColour = input$by_species, groupFill = input$by_species) } p }, res = 100) }`

- `shinyApp(ui, server)`

- See next page for the Web app

Output: We can put our Shiny app on the web by using our own servers or posit's hosting service!



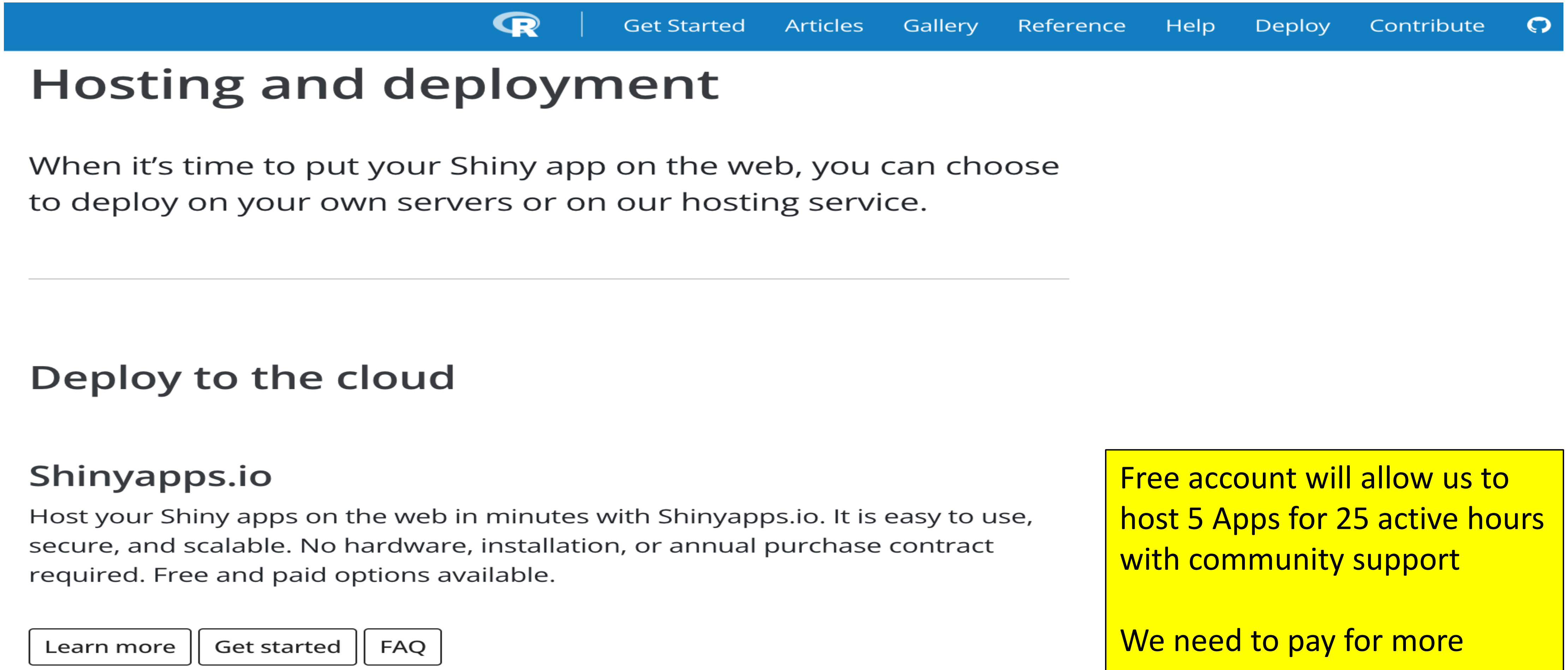
X variable
Bill Length (mm)

Y variable
Bill Depth (mm)

Filter by species
- [x] Adelie
- [x] Gentoo
- [x] Chinstrap

- [x] Show species
- [x] Show marginal plots
- [ ] Add smoother

Hosting and Deployment R Shiny web app:



Hosting and deployment

When it’s time to put your Shiny app on the web, you can choose to deploy on your own servers or on our hosting service.

Deploy to the cloud

Shinyapps.io

Host your Shiny apps on the web in minutes with Shinyapps.io. It is easy to use, secure, and scalable. No hardware, installation, or annual purchase contract required. Free and paid options available.

Free account will allow us to host 5 Apps for 25 active hours with community support

We need to pay for more

Details are at Shinyapps.io

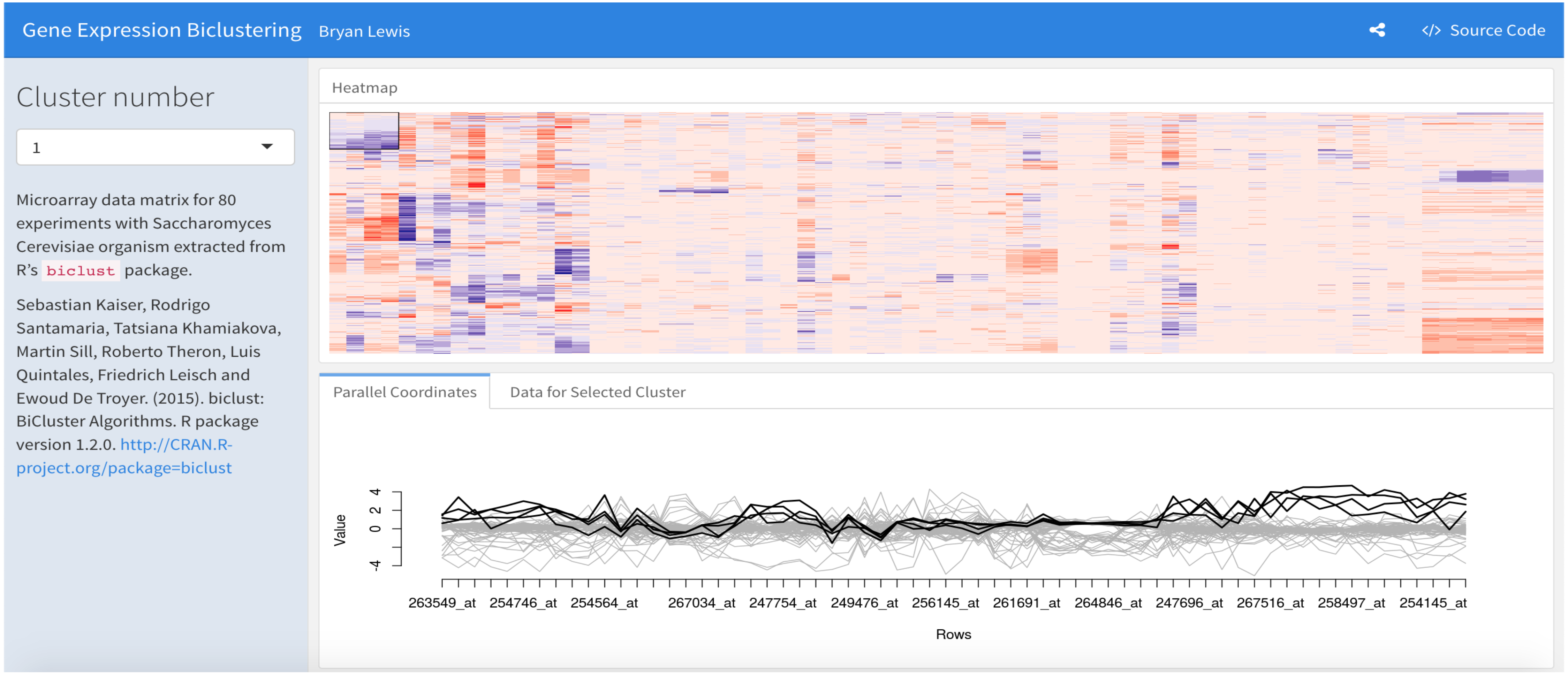
R “Flexdashboard”: Interactive dashboard https://pkgs.rstudio.com/flexdashboard/

- The goal of **flexdashboard** is to make it easy to create interactive dashboards for R, using R Markdown (can work without Shiny!).

- If you are not using RStudio, you can create a new flexdashboard R Markdown file from the R console.

- Currently there are two templates to do so:
  - "flex_dashboard" (basic)
  - "flex_dashboard_bslib"

Example:



Gene Expression Biclustering   Bryan Lewis

</> Source Code

Cluster number

1

Microarray data matrix for 80 experiments with Saccharomyces Cerevisiae organism extracted from R's `biclust` package.

Sebastian Kaiser, Rodrigo Santamaria, Tatsiana Khamiakova, Martin Sill, Roberto Theron, Luis Quintales, Friedrich Leisch and Ewoud De Troyer. (2015). biclust: BiCluster Algorithms. R package version 1.2.0. http://CRAN.R-project.org/package=biclust

Question/queries?

- Next classes

- Presentations
- 5 minutes each
- Details are already posted in the Google Classroom

Thank you!

@shitalbhandary