



Week 4: Coding Workflows

SESSION 2: ALL THAT'S FIT TO GIT!

STARFISH SCHOOL 2021

Markdown

Getting down with Markdown

- Common, simple, structured way of writing in plain text files
- Easily interpretable, supports images, tables, links, code
- Natively supported in GitHub, and even Slack!

#Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipisicing elit, quis **nostrud exercitation** ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in *voluptate velit*.

###Code

```
```javascript
var foo = 'bar';
if(true) foo = 'foo';
```
```

###Tables

| First Header | Second Header |
|-----------------------------|------------------------------|
| Content from cell 1 | Content from cell 2 |
| Content in the first column | Content in the second column |

###Lists

- [x] @mentions, #refs, [links](#)(), **formatting**, and ~~tags~~ supported
- [x] list syntax required (any unordered or ordered list supported)
- [x] this is a complete item
- [] this is an incomplete item

Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipisicing elit, quis **nostrud exercitation** ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in *voluptate velit*.

Code

```
var foo = 'bar';
if(true) foo = 'foo';
```

Tables

| First Header | Second Header |
|-----------------------------|------------------------------|
| Content from cell 1 | Content from cell 2 |
| Content in the first column | Content in the second column |

Lists

- ☒ @mentions, #refs, [links](#), **formatting**, and ~~tags~~ supported
- ☒ list syntax required (any unordered or ordered list supported)

Getting down with Markdown

- Common, simple, structured way of writing in plain text files
- Easily interpretable, supports images, tables, links, code
- Natively supported in GitHub, and even Slack!

#Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipisicing elit, quis **nostrud exercitation** ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in *voluptate velit*.

###Code

```
```javascript
var foo = 'bar';
if(true) foo = 'foo';
```
```

###Tables

| First Header | Second Header |
|-----------------------------|------------------------------|
| Content from cell 1 | Content from cell 2 |
| Content in the first column | Content in the second column |

###Lists

- [x] @mentions, #refs, [links](), **formatting**, and `code` del> supported
- [x] list syntax required (any unordered or ordered list supported)
- [x] this is a complete item
- [] this is an incomplete item

Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipisicing elit, quis **nostrud exercitation** ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in *voluptate velit*.

Code

```
var foo = 'bar';
if(true) foo = 'foo';
```

Tables

| First Header | Second Header |
|-----------------------------|------------------------------|
| Content from cell 1 | Content from cell 2 |
| Content in the first column | Content in the second column |

HOT TIP

Putting a “readme.md” file in the base level of your GitHub repository makes the contents of that file appear when you look at the repository on the web



Some Basic Markup

This is a heading

This is a smaller heading

This is an even smaller heading

* This is list item 1. I am underlined

* This is list item 2. ****I Can Be Bold****

* I want to put a [Link In
Here] (<http://www.link.com>)

! [Image] (image.png)

A Little More Markup

```
> This is something that's in a quote
```

```
...
```

```
A little bit of code can go in here
```

```
...
```

```
# A Table
```

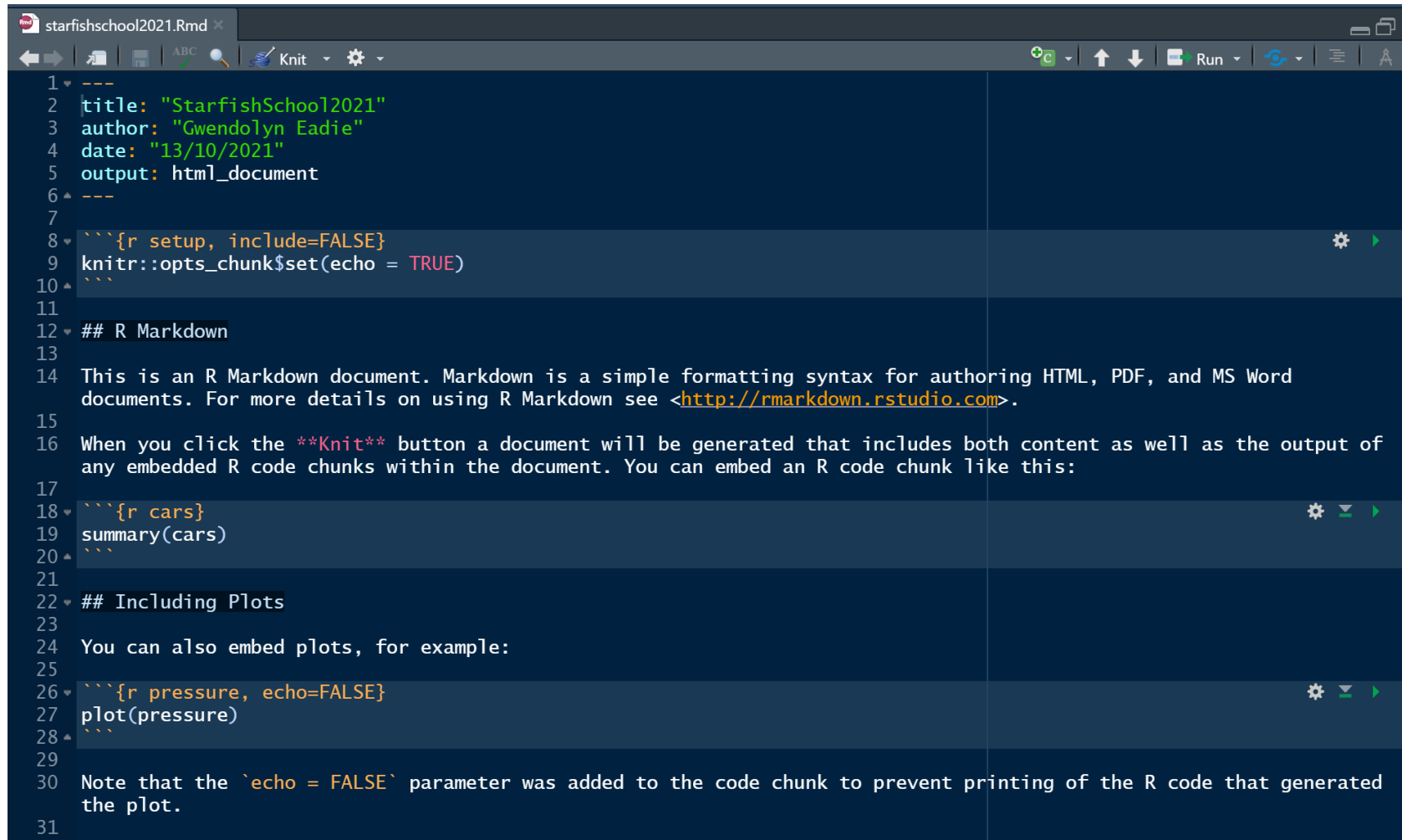
```
| Col 1 | Col2 | Col 3 |
```

```
| --- | --- | --- |
```

```
| a | b | c |
```

```
| a | b | c |
```

R Markdown in R Studio




```
1 ---
2 title: "StarfishSchool2021"
3 author: "Gwendolyn Eadie"
4 date: "13/10/2021"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word
15 documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
16
17 When you click the Knit button a document will be generated that includes both content as well as the output of
18 any embedded R code chunks within the document. You can embed an R code chunk like this:
19
20 ```{r cars}
21 summary(cars)
22 ```
23
24 ## Including Plots
25
26 You can also embed plots, for example:
27
28 ```{r pressure, echo=FALSE}
29 plot(pressure)
30 ```
31
32 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated
33 the plot.
```



Demo Time!
R Markdown

Cheat Sheets

[Markdown Cheat Sheet](#) | [Markdown Guide](#)

 **Markdown Guide**

[Get Started](#) [Cheat Sheet](#) [Basic Syntax](#) [Extended Syntax](#) [Tools](#) [Book](#)

Search

Markdown Cheat Sheet

A quick reference to the Markdown syntax.

Overview

This Markdown cheat sheet provides a quick overview of all the Markdown syntax elements. It can't cover every edge case, so if you need more information about any of these elements, refer to the reference guides for [basic syntax](#) and [extended syntax](#).

- Overview
- Basic Syntax
- Extended Syntax
- Downloads

Basic Syntax

These are the elements outlined in John Gruber's original design document. All Markdown applications support these elements.

| Element | Markdown Syntax |
|----------------------------|--|
| Heading | <code># H1</code>
<code>## H2</code>
<code>### H3</code> |
| Bold | <code>**bold text**</code> |
| Italic | <code><i>*italicized text*</i></code> |
| Blockquote | <code>> blockquote</code> |

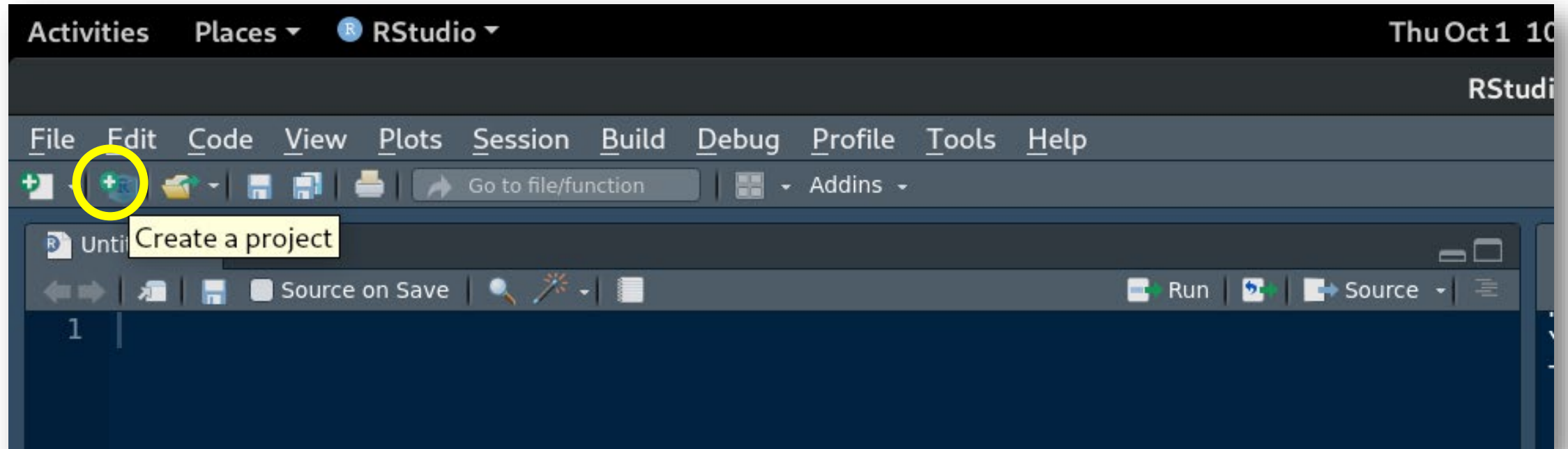
Git and R

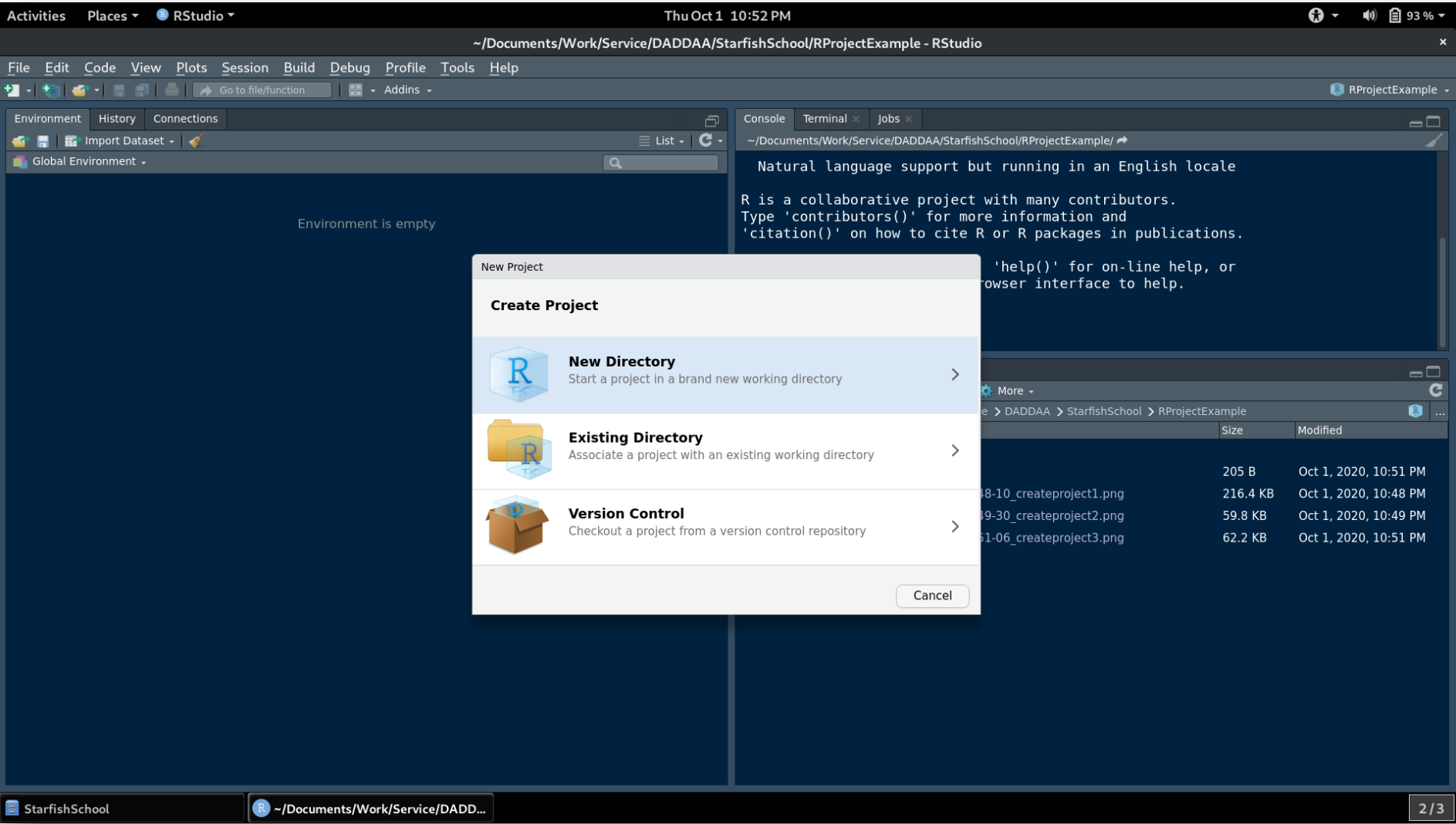
RStudio, R Projects, and Git

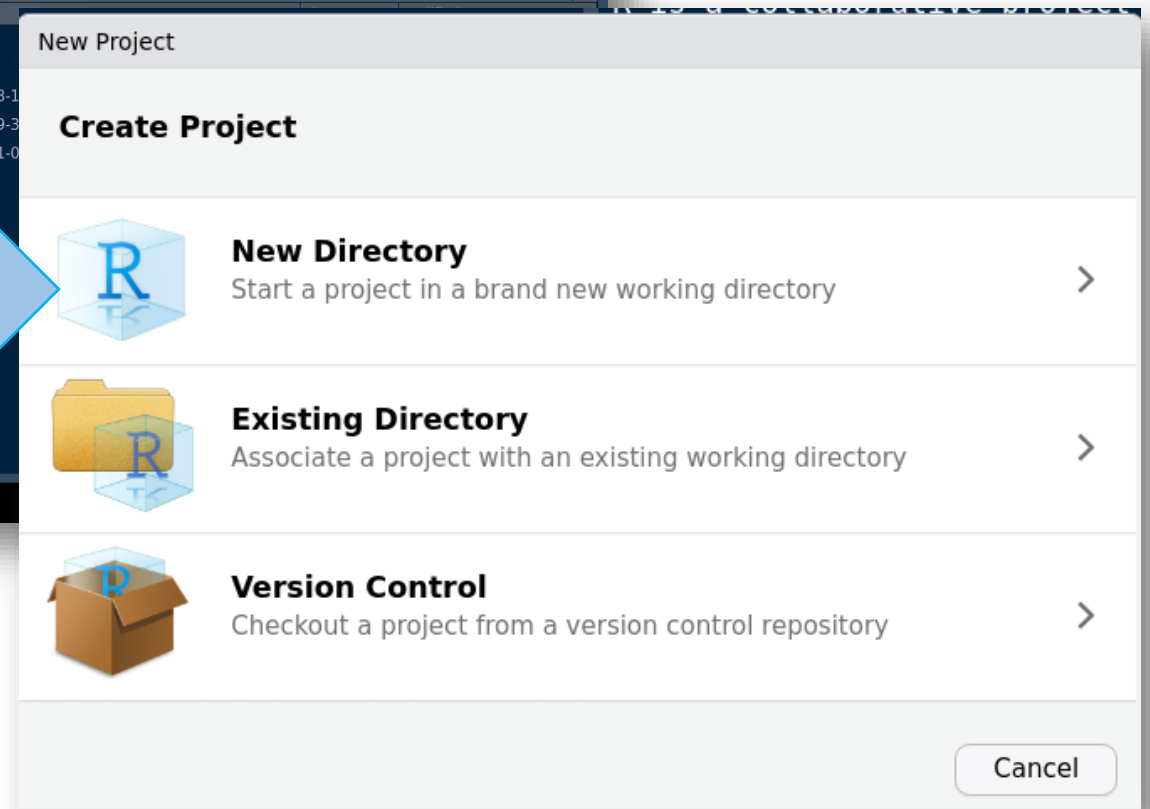
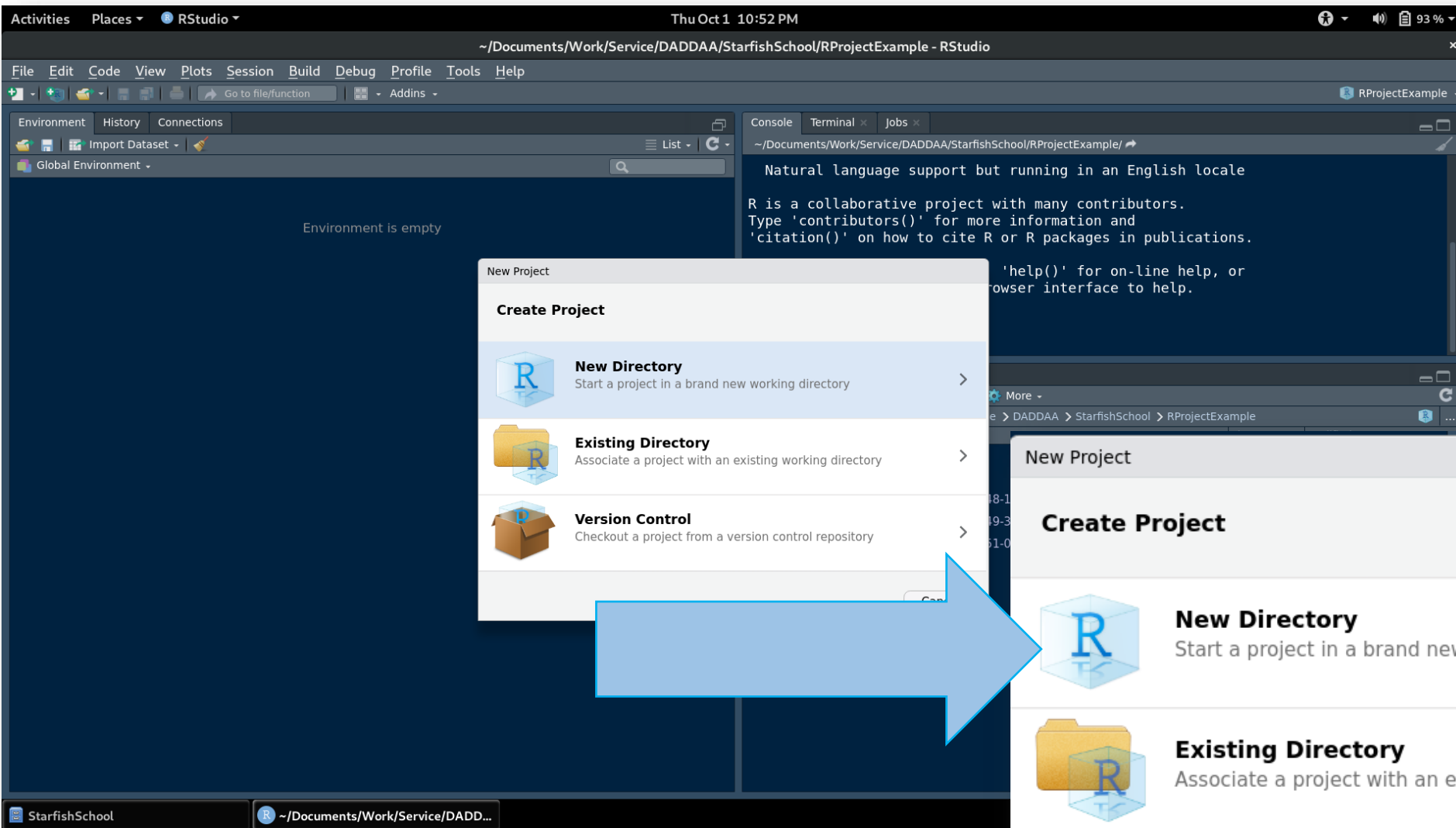
- An R project is a great way to keep track of your R scripts and other files
- RProjects work with or without git, all within R Studio.
- When you open an R Project, it will open all the files you previously had open in RStudio (ie., it will pick up where you left off)
- The next few slides will show you how to set up a new project and initialize a git repository for that project

RStudio, R Projects, and Git

- An R project is a great way to keep track of your rscripts and other files you want to version control.







~/Documents/Work/Service/DADDAA/StarfishSchool/RProjectExample - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

RProjectExample

Environment History Connections

Import Dataset

Global Environment

Environment is empty

Console Terminal Jobs

~/Documents/Work/Service/DADDAA/StarfishSchool/RProjectExample/

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.

'help()' for on-line help, or
browser interface to help.

New Project

Back

Project Type



New Project



R Package



Shiny Web Application

R Package using Rcpp

R Package using RcppArmadillo

R Package using RcppEigen

R Package using RcppParallel

Create a new
project in an empty
directory

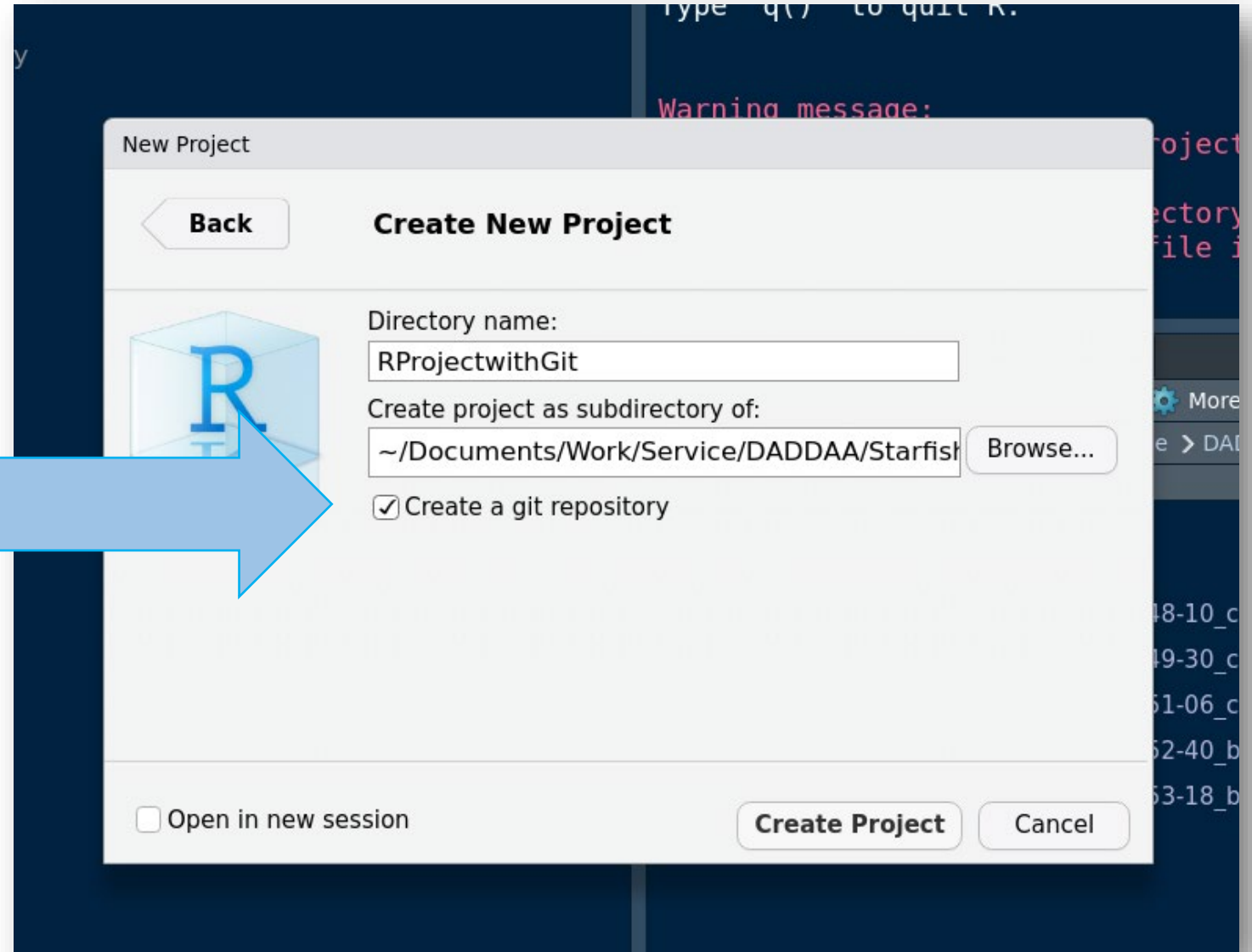
Cancel

A > StarfishSchool > RProjectExample

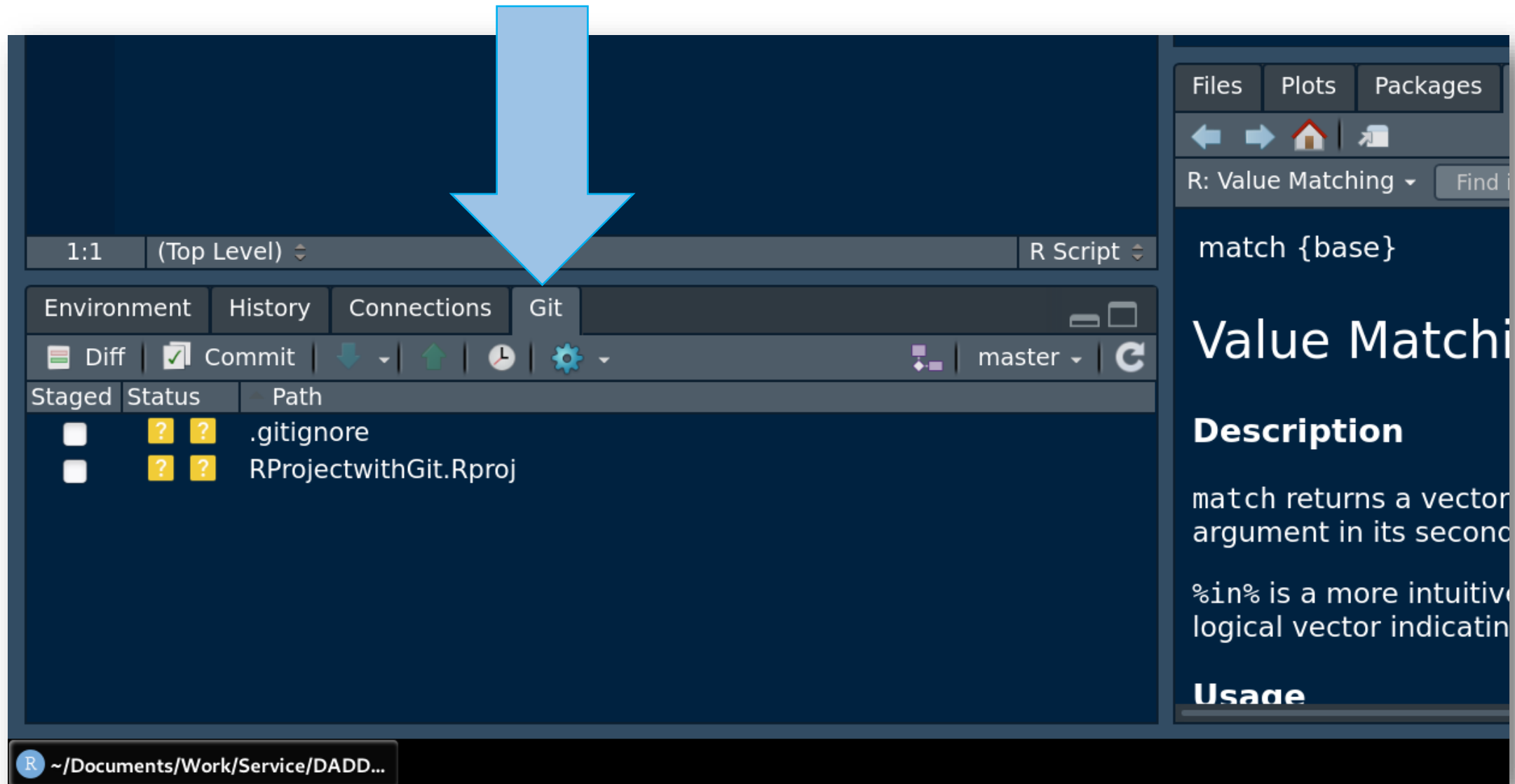
| | Size | Modified |
|--------------------------|----------|-----------------------|
| | 205 B | Oct 1, 2020, 10:51 PM |
| 48-10_createproject1.png | 216.4 KB | Oct 1, 2020, 10:48 PM |
| 49-30_createproject2.png | 59.8 KB | Oct 1, 2020, 10:49 PM |
| 51-06_createproject3.png | 62.2 KB | Oct 1, 2020, 10:51 PM |
| 52-40_brandnew1.png | 221.7 KB | Oct 1, 2020, 10:52 PM |

Directory Name for project

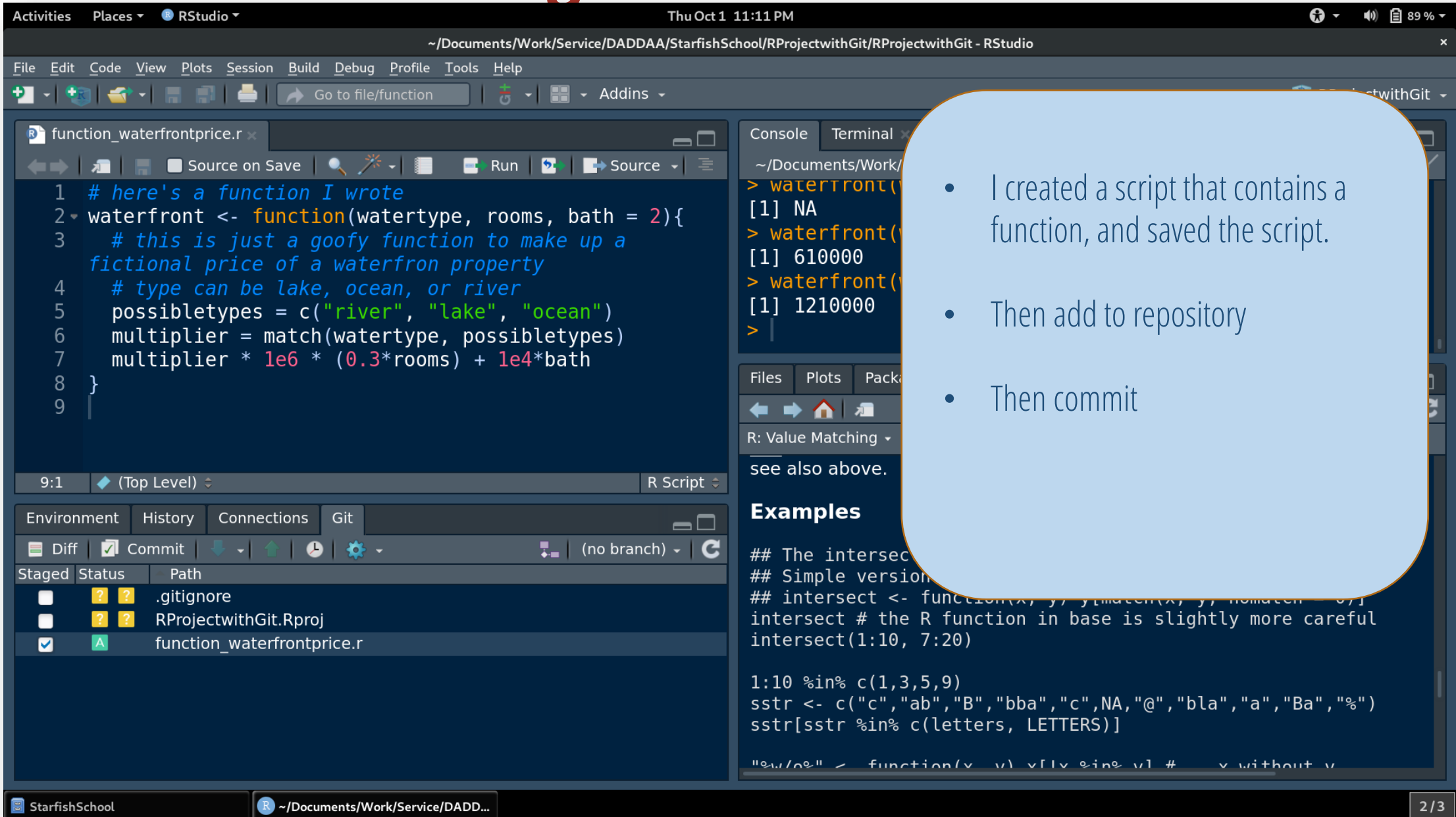
Check this box to create a git repository when you start a new project



- Once the RProject with a git repository is set up, you will see Git options appear in RStudio



User Interface for git in RStudio



The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains a script named `function_waterfrontprice.r` with the following R code:

```
1 # here's a function I wrote
2 waterfront <- function(watertype, rooms, bath = 2){
3   # this is just a goofy function to make up a
4   # fictional price of a waterfron property
5   # type can be lake, ocean, or river
6   possibletypes = c("river", "lake", "ocean")
7   multiplier = match(watertype, possibletypes)
8   multiplier * 1e6 * (0.3*rooms) + 1e4*bath
9 }
```
- Console:** Shows the execution of the `waterfront` function with the following output:

```
> waterfront(
[1] NA
> waterfront(
[1] 610000
> waterfront(
[1] 1210000
>
```
- Git Panel:** Located at the bottom, it shows the status of the repository. The `function_waterfrontprice.r` file is staged for commit, indicated by a green 'A' in the status column.

A light blue rounded rectangle is overlaid on the right side of the interface, containing the following steps:

- I created a script that contains a function, and saved the script.
- Then add to repository
- Then commit

Below the Git panel, the **Examples** section shows the following R code:

```
## The intersect function in base R is slightly more careful
## Simple version
intersect <- function(x, y) {
  match(x, y)
}
intersect # the R function in base is slightly more careful
intersect(1:10, 7:20)

1:10 %in% c(1,3,5,9)
sstr <- c("c","ab","B","bba","c",NA,"@","bla","a","Ba","%")
sstr[sstr %in% c(letters, LETTERS)]

"ew/oe" <- function(x, y) {
  x[y %in% y] # y without y
```

User Interface for git in RStudio

Activities Places RStudio Thu Oct 1 11:11 PM

~/Documents/Work/Service/DADDAA/StarfishSchool/RProjectwithGit/RProjectwithGit - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

RStudio: Review Changes

Changes History (no branch) Stage Revert Ignore Pull Push

staged Status Path

☐ .gitignore

☐ RProjectwithGit

☒ function_wat

Commit message

my message h

☐ Amend previous commit

Commit

Show Staged Unstaged Context 5 line Ignore Whitespace Unstage All

@@ -0,0 +1,8 @@

```
1 # here's a function I wrote
2 waterfront <- function(watertype, rooms, bath = 2){
3   # this is just a goofy function to make up a fictional
4   # price of a waterfron property
5   # type can be lake, ocean, or river
6   possibletypes = c("river", "lake", "ocean")
7   multiplier = match(watertype, possibletypes)
8   multiplier * 1e6 * (0.3*rooms) + 1e4*bath
9 }
```

Console

```
~/Documents/Work/Service/DADDAA/StarfishSchool/RProjectwithGit/RProjectwithGit
> waterfront(
[1] NA
> waterfront(
[1] 610000
> waterfront(
[1] 1210000
>
```

Files Plots Packages

R: Value Matching

see also above.

Examples

```
## The intersect function
## Simple version
## intersect <- function(x, y, ymatch = FALSE, nomatch = FALSE) {
intersect # the R function in base is slightly more careful
intersect(1:10, 7:20)

1:10 %in% c(1,3,5,9)
sstr <- c("c","ab","B","bba","c",NA,"@","bla","a","Ba","%")
sstr[sstr %in% c(letters, LETTERS)]

"ew/oe" <- function(x, y) x[!x %in% y] # x without y
```

- I created a script that contains a function, and saved the script.
- Then add to repository
- Then commit

StarfishSchool ~/Documents/Work/Service/DADD... 2/3

User Interface for git in RStudio

Activities Places RStudio Thu Oct 1 11:19 PM

RStudio: Review Changes

Changes History master Stage Revert Ignore Pull Push

Staged Status Path

Commit message

I decided to add another comment

☐ Amend previous commit Commit

Show Staged Unstaged Context 5 line Ignore Whitespace Unstage All

@@ -1,8 +1,9 @@

```
1 1 # here's a function I wrote
2 2 waterfront <- function(watertype, rooms, bath = 2){
3 3   # this is just a goofy function to make up a fictional
4 4   # type can be lake, ocean, or river
5 5   # basically, a lake house is twice as expensive as a
6 6   # house by the river, and an oceanfront house is three
7 7   # times as expensive as a riverside house.
8 8   possibletypes = c("river", "lake", "ocean")
9 9   multiplier = match(watertype, possibletypes)
10 10 multiplier * 1e6 * (0.3*rooms) + 1e4*bath
11 11 }
```

function_waterfrontprice.r

```
1 # here's a function I wrote
2 waterfront <- function(watertype, rooms, bath = 2){
3   # this is just a goofy function to make up a fictional
4   # type can be lake, ocean, or river
5   # basically, a lake house is twice as expensive as a
6   # house by the river, and an oceanfront house is three
7   # times as expensive as a riverside house.
8   possibletypes = c("river", "lake", "ocean")
9   multiplier = match(watertype, possibletypes)
10  multiplier * 1e6 * (0.3*rooms) + 1e4*bath
11 }
```

5:149 waterfront(watertype, rooms, bath)

Environment History Connections Git

Diff Commit

Staged Status Path

.gitignore

RProjectwithGit.Rproj

function_waterfrontprice.r

StarfishSchool ~/Documents/Work/Service/DADD... RStudio: Review Changes 2/3

- If you make a change, then you can look at the difference between versions

Exercise

Teams of 2-3

One member of the team should fork the exercise repository and give the other team member access to the repository.

1. Each of the members should clone the repository and edit a file called "load_data.py"
2. In the file, everyone should create a function that produces the Fibonacci sequence to a certain input number, and commit it to their local repository
3. Push your changes to the remote repository, and deal with the conflicts.
4. Edit the file named "readme.md" with your description of the repository and push to GitHub

Exercise

- Create a new folder called "TestProject"
- Open R Studio and start a new R Project within this folder
- Make an R script with a function, or a few commands, etc. (whatever you like!)
- Add an R markdown document to the Project describing your functions.
- Save the R script and R markdown and add it to the git repository
- Commit the file.
- Make a change to the R script, and then look at the difference between the changes and the previous commit.