

R Packages / Git / GitHub

Jacob Strunk

3/27/2018



Setup / some details

Download / install:

- R
- RStudio
- Git
- Miktex
- GitHub Account





GitHub



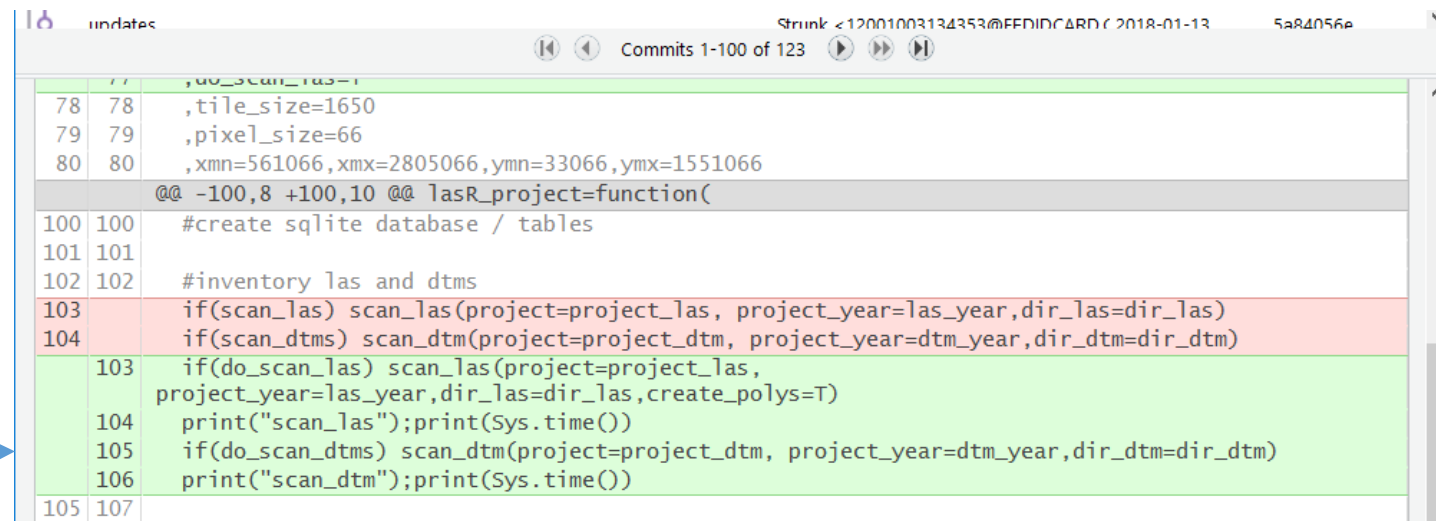
- Versioning software
(e.g. track changes in MS Word)
- Free / open source

- Online code / version hosting
- Works with GIT
- Collaboration
- Free
- Paid (private repo)

Why **git** is Awesome(!!):

- a) You wrote some GREAT code!!! Yippee
- b) You fixed that code, and made it better!!! More Yippee
- c) Turns out in **b)** you broke **a)** 😞
- d) Git: Find / fix errors 😊

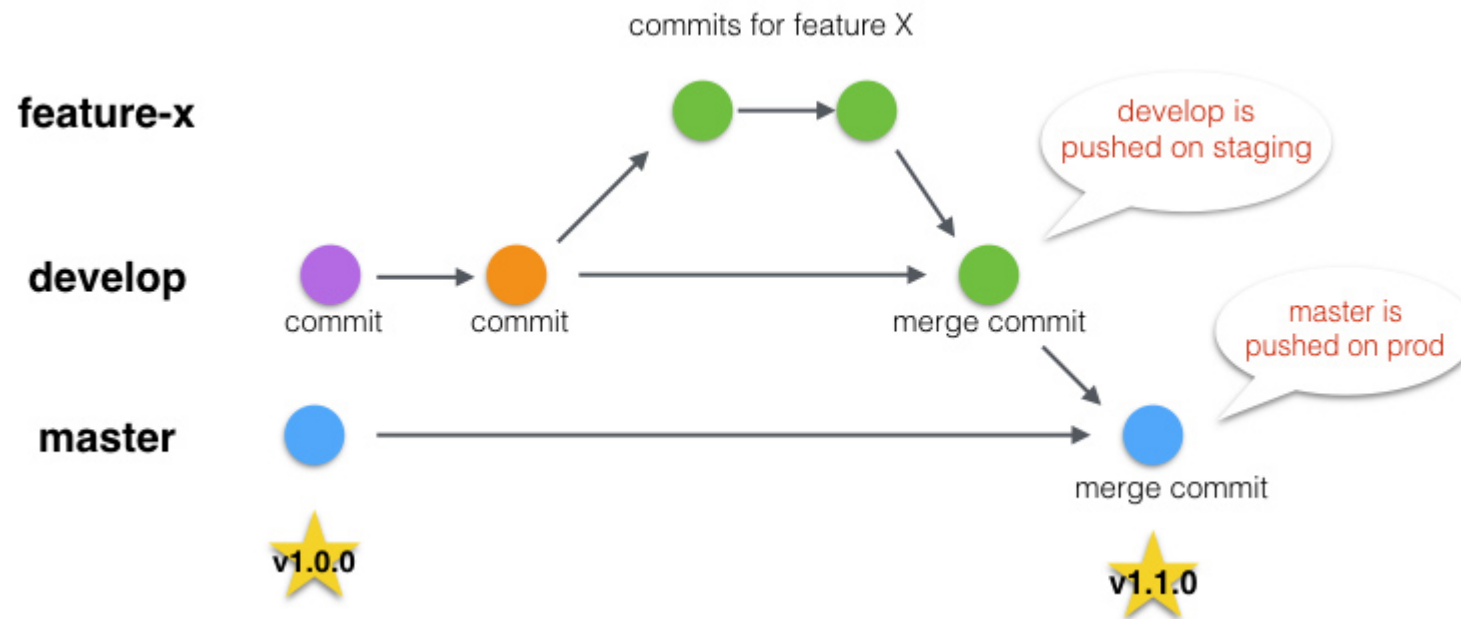
New, bad code →



```
updates Strunk <12001003134353@FFD1DC4RD (2018-01-13) 5a84056e
Commits 1-100 of 123

77 77 ,do_scan_las=T
78 78 ,tile_size=1650
79 79 ,pixel_size=66
80 80 ,xmn=561066,xmx=2805066,ymn=33066,ymx=1551066
@@ -100,8 +100,10 @@ lasR_project=function(
100 100 #create sqlite database / tables
101 101
102 102 #inventory las and dtms
103 103 if(scan_las) scan_las(project=project_las, project_year=las_year,dir_las=dir_las)
104 104 if(scan_dtms) scan_dtm(project=project_dtm, project_year=dtm_year,dir_dtm=dir_dtm)
103 103 if(do_scan_las) scan_las(project=project_las,
project_year=las_year,dir_las=dir_las,create_polys=T)
104 104 print("scan_las");print(Sys.time())
105 105 if(do_scan_dtms) scan_dtm(project=project_dtm, project_year=dtm_year,dir_dtm=dir_dtm)
106 106 print("scan_dtm");print(Sys.time())
105 107
```

git workflow





git

Simplified

- Git for tracks changes, backuo
- Work on some code, commit , push
- Repeat

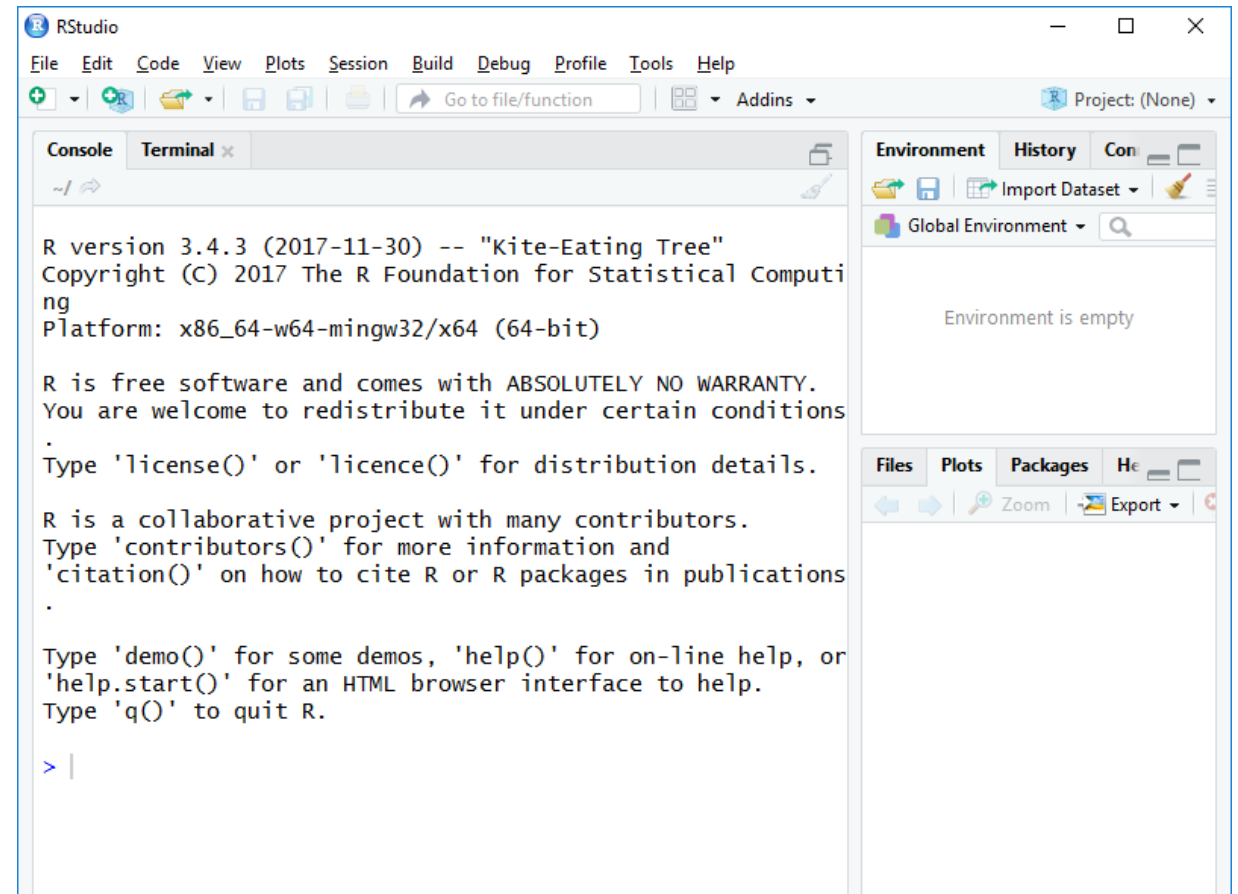




- A statistical programming language
- Flexible
- Powerful
- “Easy”
- Extensible!!!
- Many users
- Many Contributors
- Many Packages
- Free / open source



- Integrated development environment (IDE)
- Free / Open source
- Syntax highlighting
- Debugging
- Code completion
- **Package development !!**
- **Versioning (e.g. Git) !!**
- Etc.





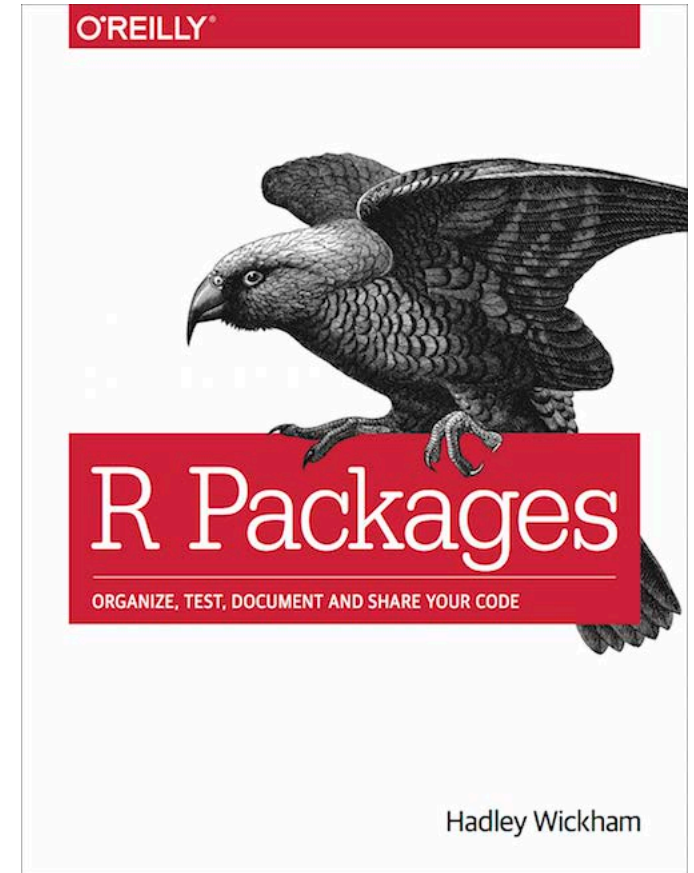
- Coding / package building / git –
 - Rstudio not “required”
 - Rstudio **Recommended**

R Packages

Package Building in R

“... the fundamental unit of shareable code is the package. A package bundles together code, data, documentation, and tests, and is easy to share with others.”

<http://r-pkgs.had.co.nz/intro.html>



Why build packages?

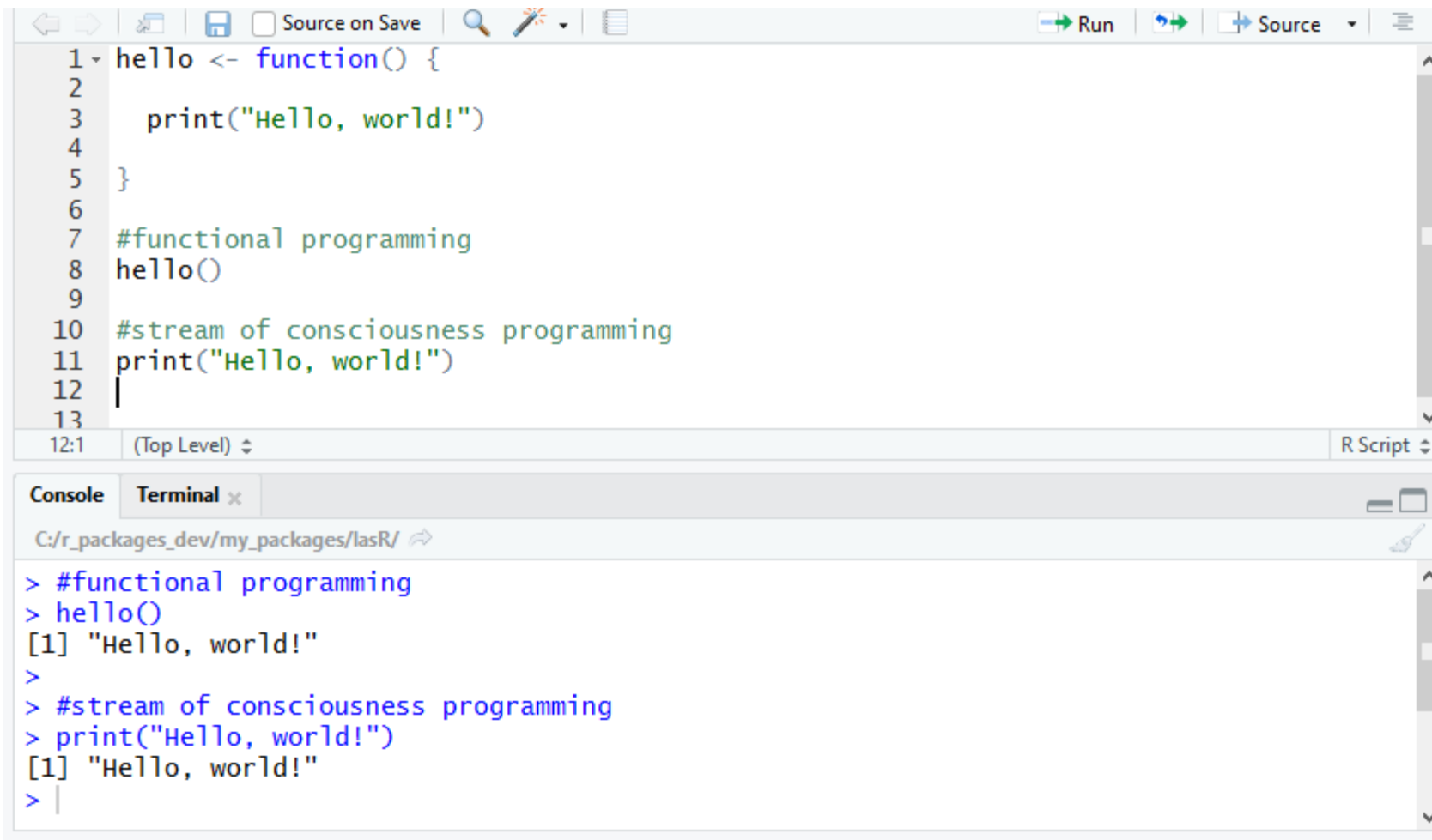
- Portable
- Testable
- Extensible
- Facilitate collaboration
- Reduce redundancy
- Encourage more work on your topic of interest
 - People are freed up to do other cool things (and hopefully create packages)
- You will never go back...



(I think I wrote this code before)

Functional programming

- R packages very suited to functional programming:



The screenshot shows an R IDE window with a script editor and a console. The script editor contains the following code:

```
1 hello <- function() {  
2  
3   print("Hello, world!")  
4  
5 }  
6  
7 #functional programming  
8 hello()  
9  
10 #stream of consciousness programming  
11 print("Hello, world!")  
12 |  
13
```

The console shows the output of the script execution:

```
> #functional programming  
> hello()  
[1] "Hello, world!"  
>  
> #stream of consciousness programming  
> print("Hello, world!")  
[1] "Hello, world!"  
> |
```

A handy function

- readClipboard()
 - Use instead of paste (ctrl + V)
 - Automatically adds double backslashes

```
> readClipboard()  
[1] "C:\\box\\BoxSync\\sync\\data\\javad\\2018 March 21 OLY FSL"  
> |
```

Another Handy Function: forward()

```
1
2
3 forward=function(x=NA,writeClipboard=T,return=T){
4
5   #get data from clipboard or from x
6   if(is.na(x)) x=readClipboard()
7
8   #substitute forwardslash for backslash
9   y=gsub("\\\\", "/", x)
10
11  #send string somewhere
12  writeClipboard(y)
13  return(y)
14
15 }
16
17 #C:\box\BoxSync\sync\docs\USFS\Ppresentations\2017 04 20 OLi\OLI 2017 Presentations
18 forward()
19
20
```

20:1 (Top Level) R Script

Console Terminal x

C:/r_packages_dev/my_packages/lasR/

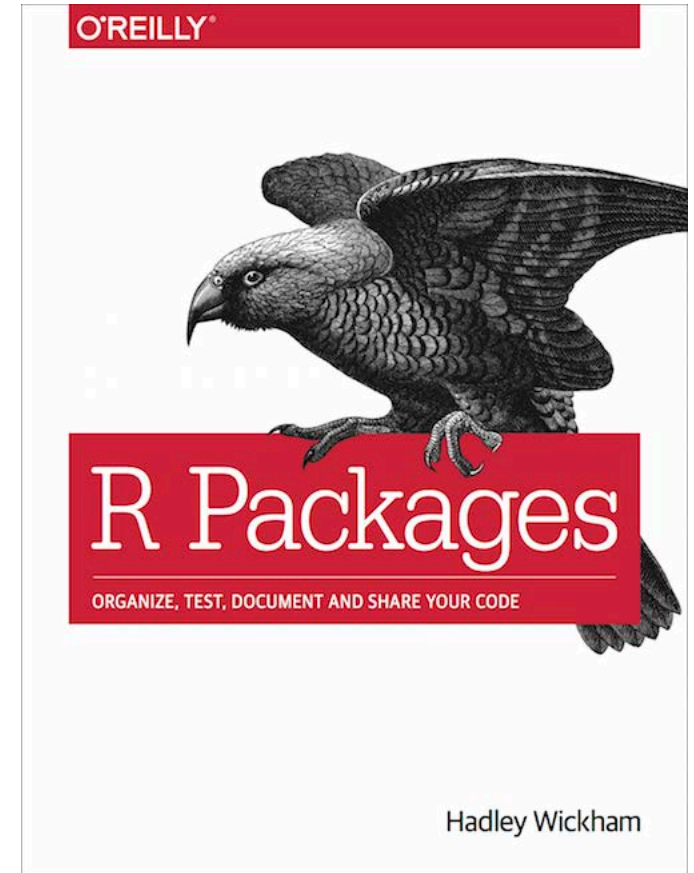
```
+ }
> forward()
[1] "C:/box/BoxSync/sync/docs/USFS/Ppresentations/2017 04 20 OLi/OLI 2017 Presentations"
> |
```


R Packages

Package Building in R

“... the fundamental unit of shareable code is the package. A package bundles together code, data, documentation, and tests, and is easy to share with others.”

<http://r-pkgs.had.co.nz/intro.html>



Why build packages?

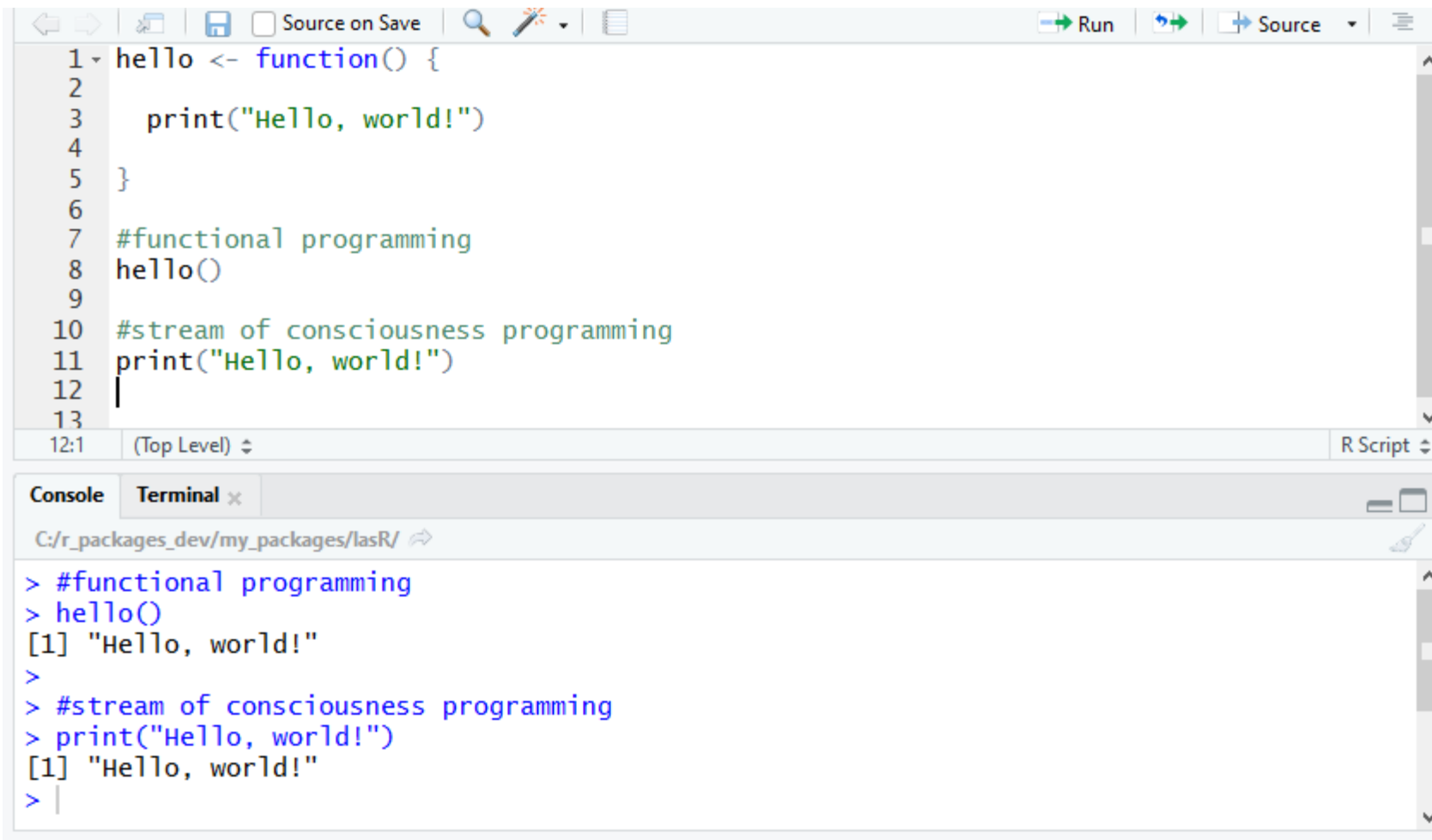
- Portable
- Testable
- Extensible
- Facilitate collaboration
- Reduce redundancy
- Encourage more work on your topic of interest
 - People are freed up to do other cool things (and hopefully create packages)
- You will never go back...



(I think I wrote this code before)

Functional programming

- R packages very suited to functional programming:



The screenshot shows an R IDE window with a script editor and a console. The script editor contains the following code:

```
1 hello <- function() {  
2  
3   print("Hello, world!")  
4  
5 }  
6  
7 #functional programming  
8 hello()  
9  
10 #stream of consciousness programming  
11 print("Hello, world!")  
12 |  
13
```

The console shows the output of the script execution:

```
> #functional programming  
> hello()  
[1] "Hello, world!"  
>  
> #stream of consciousness programming  
> print("Hello, world!")  
[1] "Hello, world!"  
> |
```

A handy function

- readClipboard()
 - Use instead of paste (ctrl + V)
 - Automatically adds double backslashes

```
> readClipboard()  
[1] "C:\\\\box\\BoxSync\\sync\\data\\javad\\2018 March 21 OLY FSL"  
> |
```

Another Handy Function: forward()

```
1
2
3 forward=function(x=NA,writeClipboard=T,return=T){
4
5   #get data from clipboard or from x
6   if(is.na(x)) x=readClipboard()
7
8   #substitute forwardslash for backslash
9   y=gsub("\\\\", "/", x)
10
11  #send string somewhere
12  writeClipboard(y)
13  return(y)
14
15 }
16
17 #C:\box\BoxSync\sync\docs\USFS\Ppresentations\2017 04 20 OLi\OLI 2017 Presentations
18 forward()
19
20
```

20:1 (Top Level) R Script

Console Terminal x

C:/r_packages_dev/my_packages/lasR/

```
+ }
> forward()
[1] "C:/box/BoxSync/sync/docs/USFS/Ppresentations/2017 04 20 OLi/OLI 2017 Presentations"
> |
```

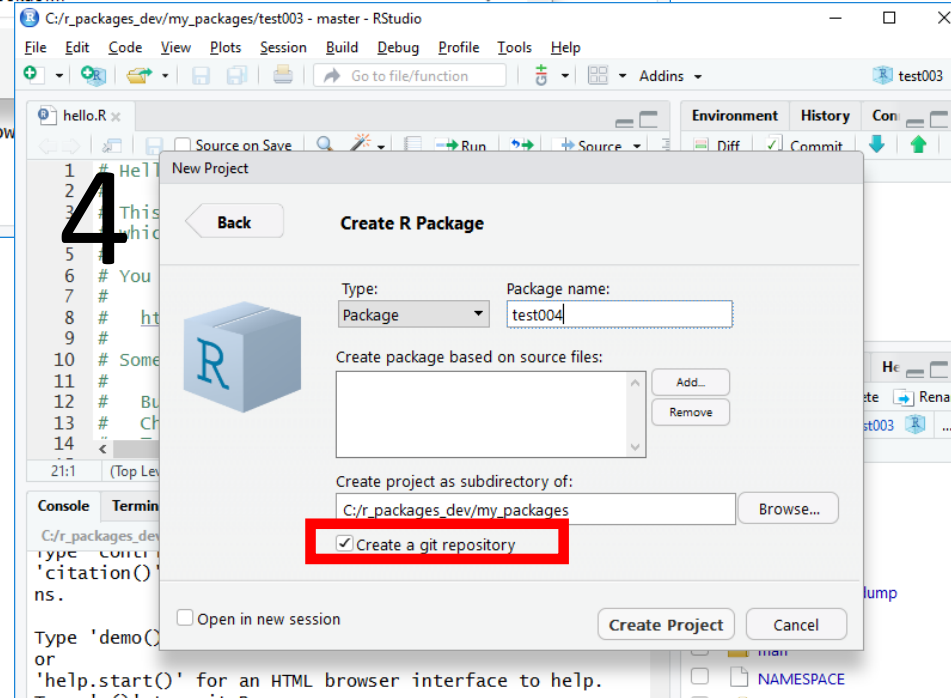
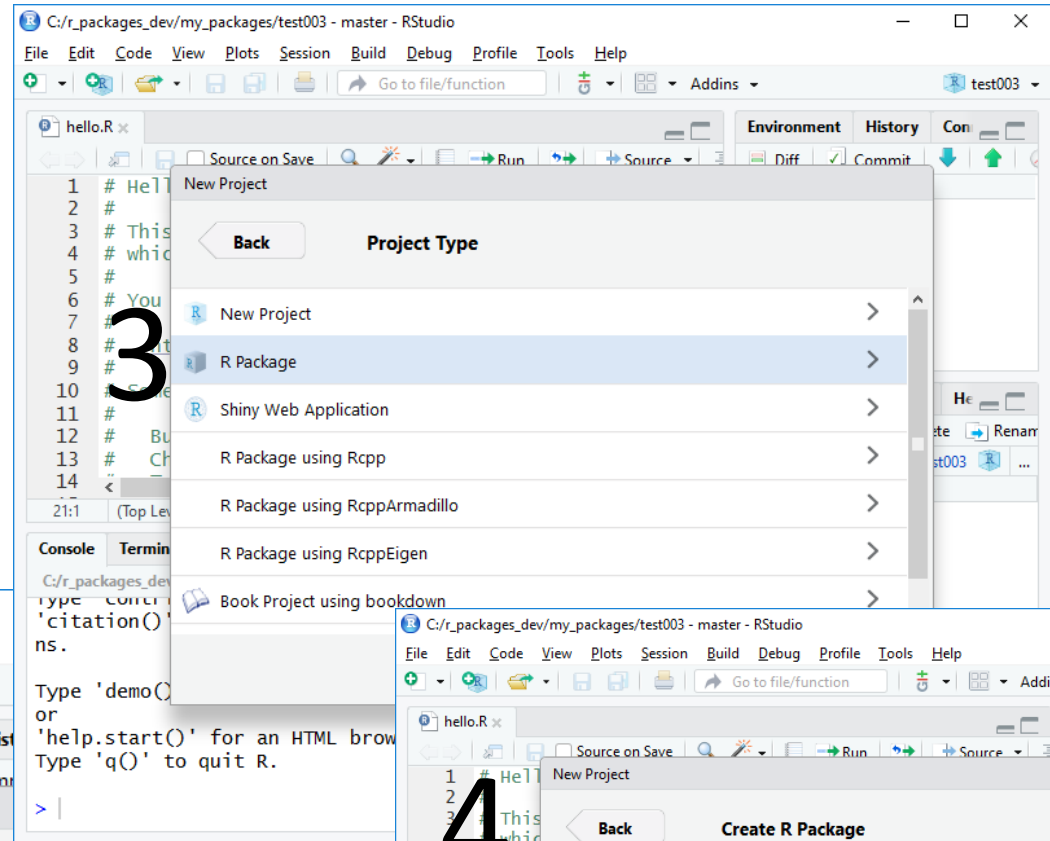
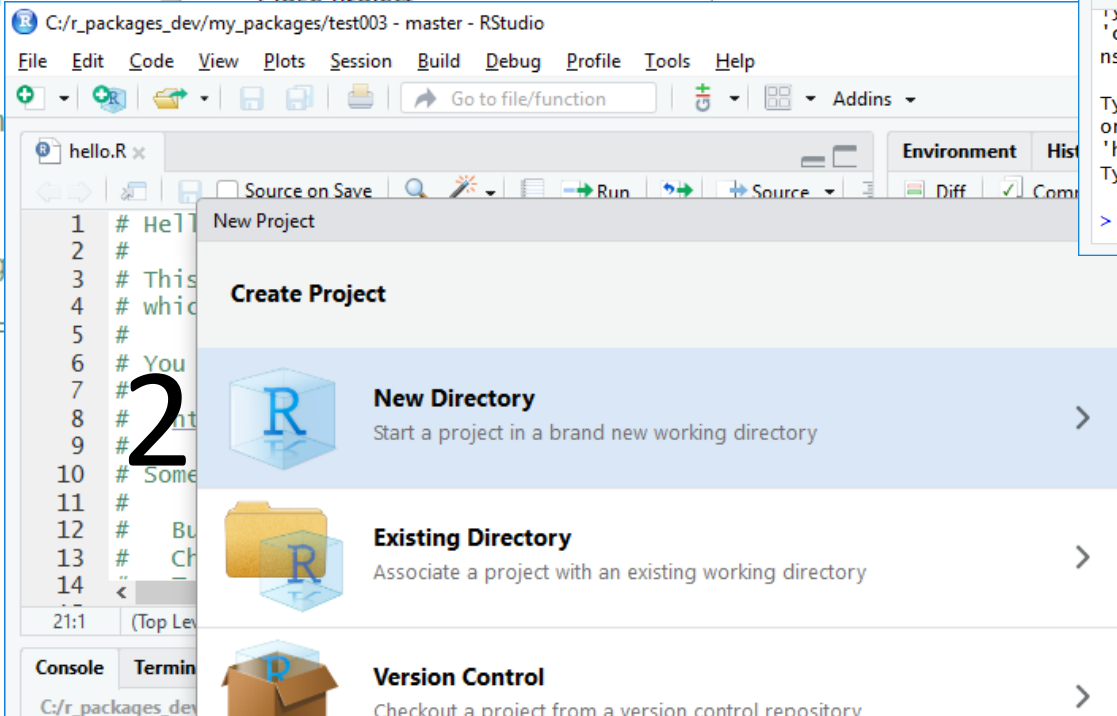
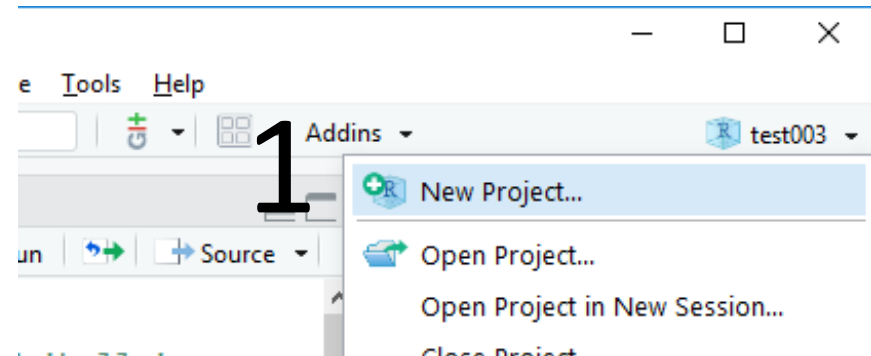
Create R Package on GitHub



&

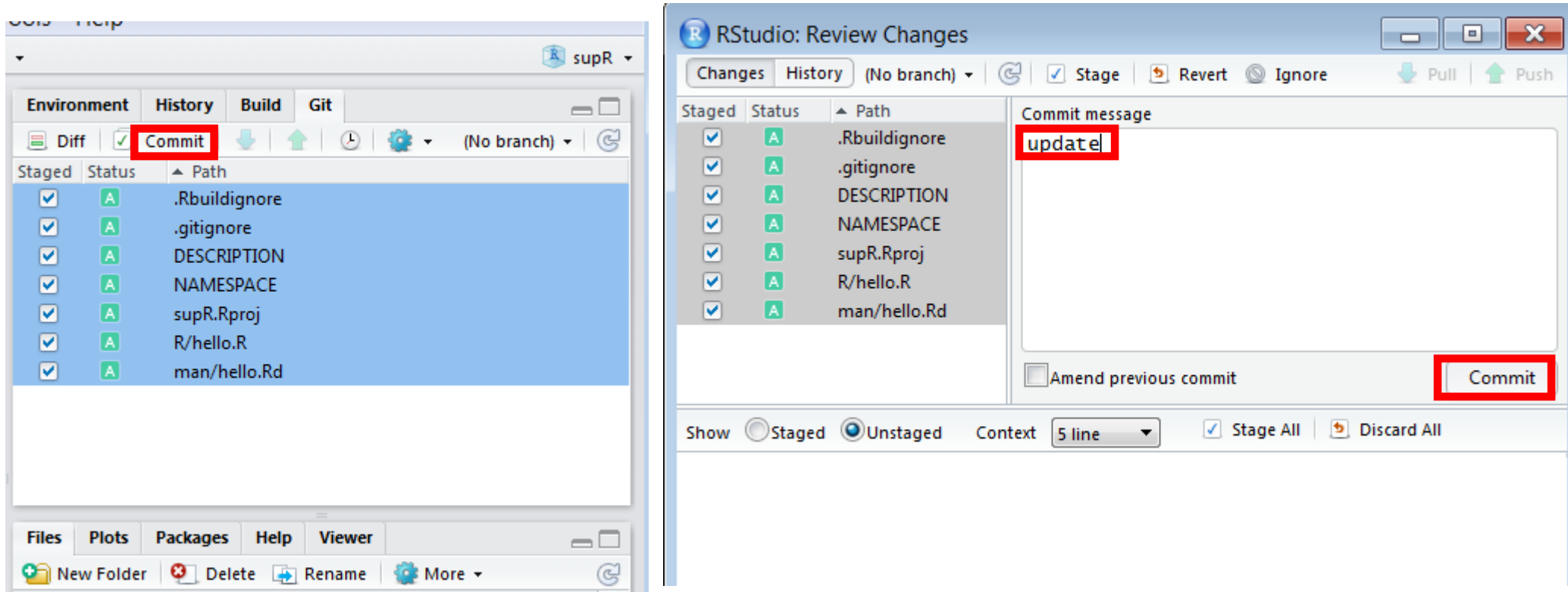


1) “Create new package” in RStudio

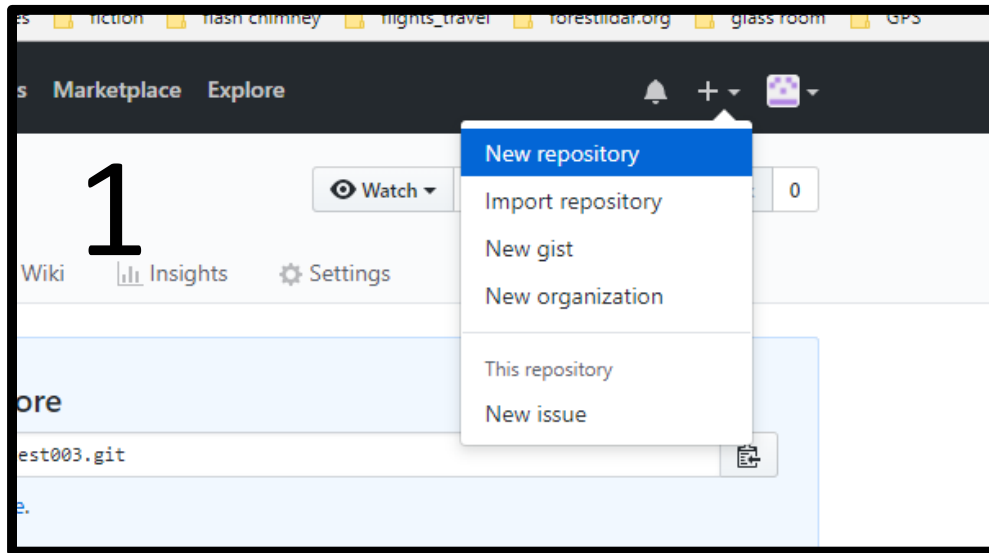


Git commit (first time)

- Shift-click to select all items



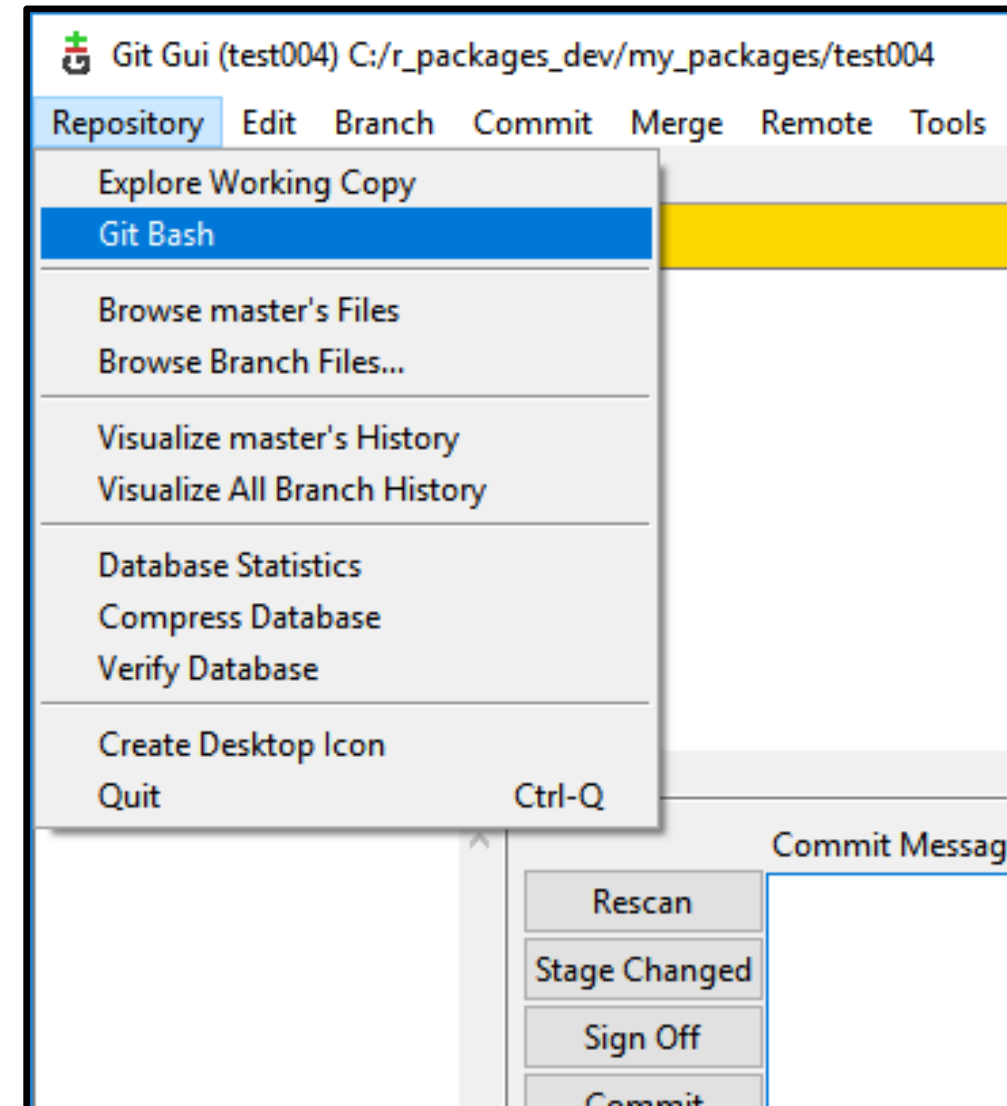
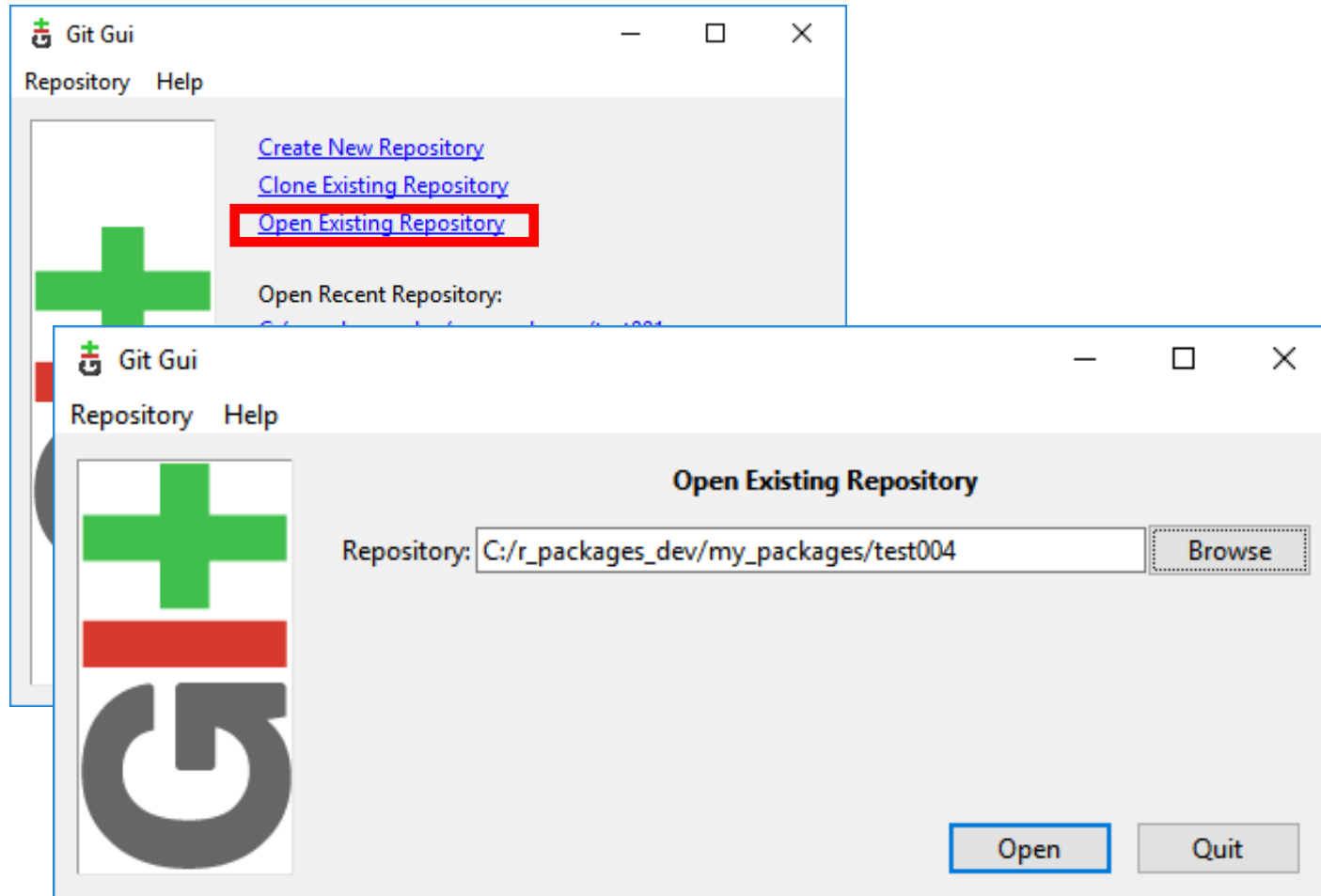
2) Create GitHub Repository (Assuming you have an account already)



A screenshot of the 'Create a new repository' form on GitHub. A large white number '2' is overlaid on the left. The form includes fields for 'Owner' (jstrunk001) and 'Repository name' (test004, with a green checkmark). Below these are options for 'Public' (selected) or 'Private', a 'Description (optional)' field, and a checkbox for 'Initialize this repository with a README'. At the bottom are dropdowns for 'Add .gitignore: None' and 'Add a license: None', and a green 'Create repository' button.

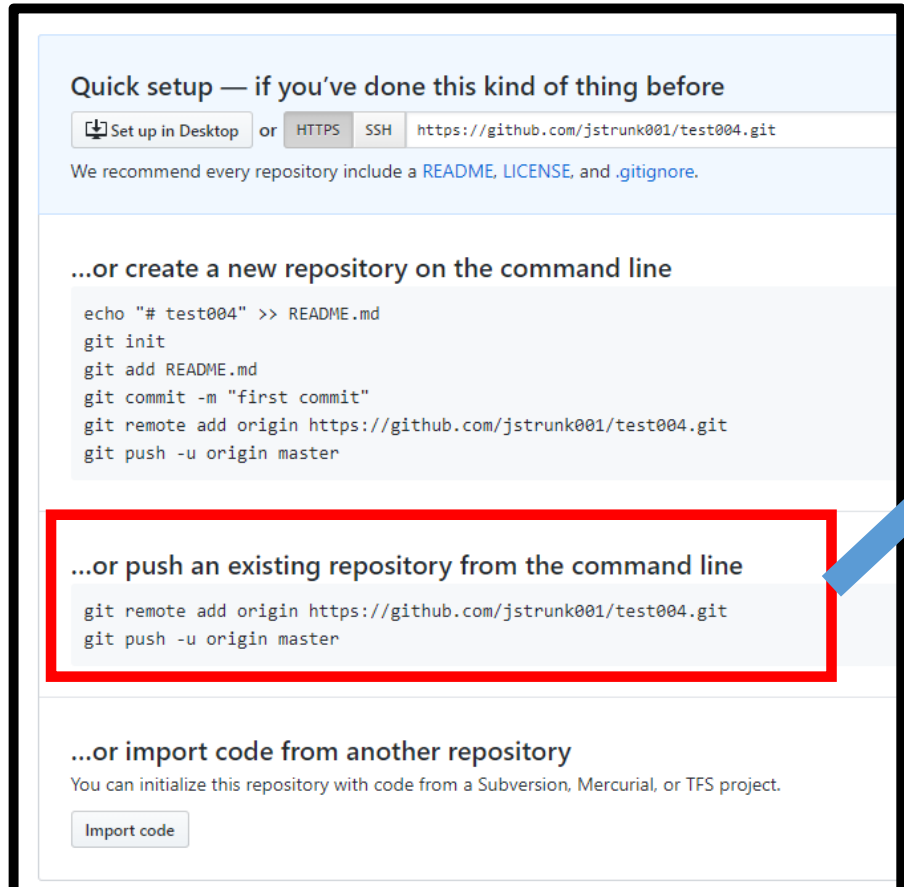


3) Open git GUI, git BASH




4) Link git & GitHub

Recall



Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** **SSH** <https://github.com/jstrunk001/test004.git>

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# test004" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/jstrunk001/test004.git
git push -u origin master
```

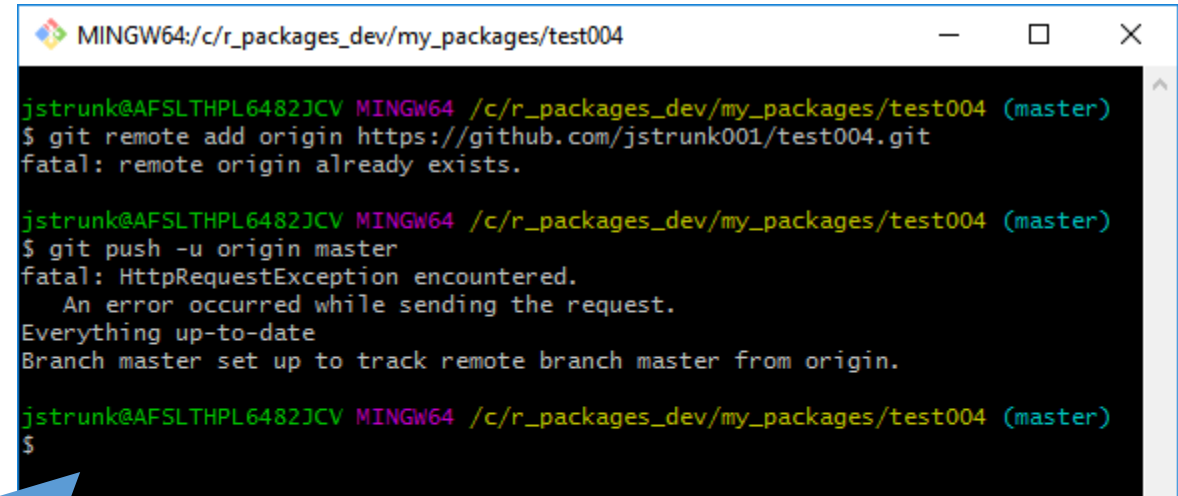
...or push an existing repository from the command line

```
git remote add origin https://github.com/jstrunk001/test004.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

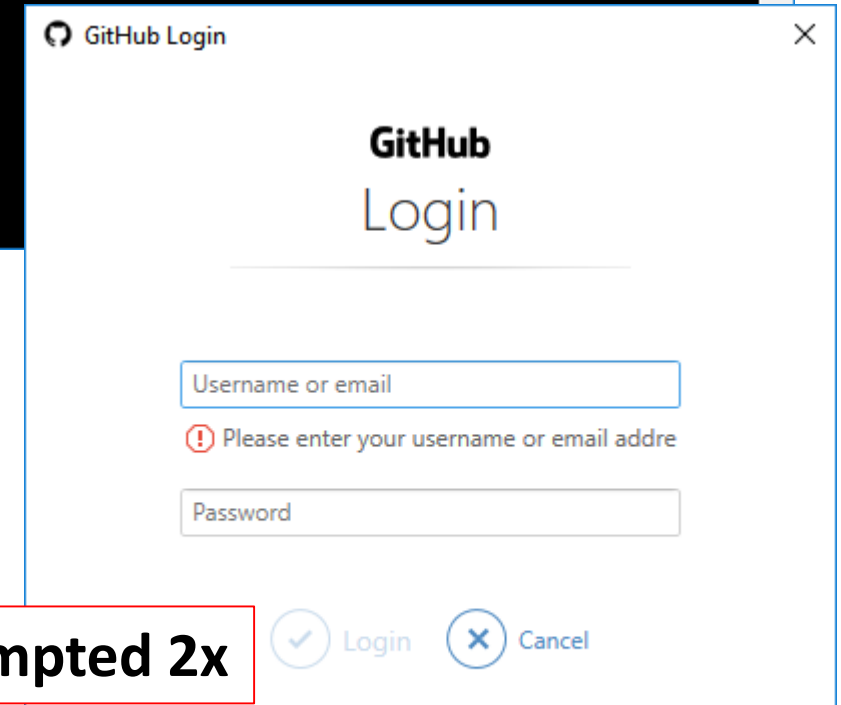


```
MINGW64:/c/r_packages_dev/my_packages/test004

jstrunk@AFSLTHPL6482JCV MINGW64 /c/r_packages_dev/my_packages/test004 (master)
$ git remote add origin https://github.com/jstrunk001/test004.git
fatal: remote origin already exists.

jstrunk@AFSLTHPL6482JCV MINGW64 /c/r_packages_dev/my_packages/test004 (master)
$ git push -u origin master
fatal: HttpRequestException encountered.
An error occurred while sending the request.
Everything up-to-date
Branch master set up to track remote branch master from origin.

jstrunk@AFSLTHPL6482JCV MINGW64 /c/r_packages_dev/my_packages/test004 (master)
$
```



GitHub Login

Username or email

❗ Please enter your username or email address

Password

[Login](#) [Cancel](#)

May be prompted 2x

Connection Successful

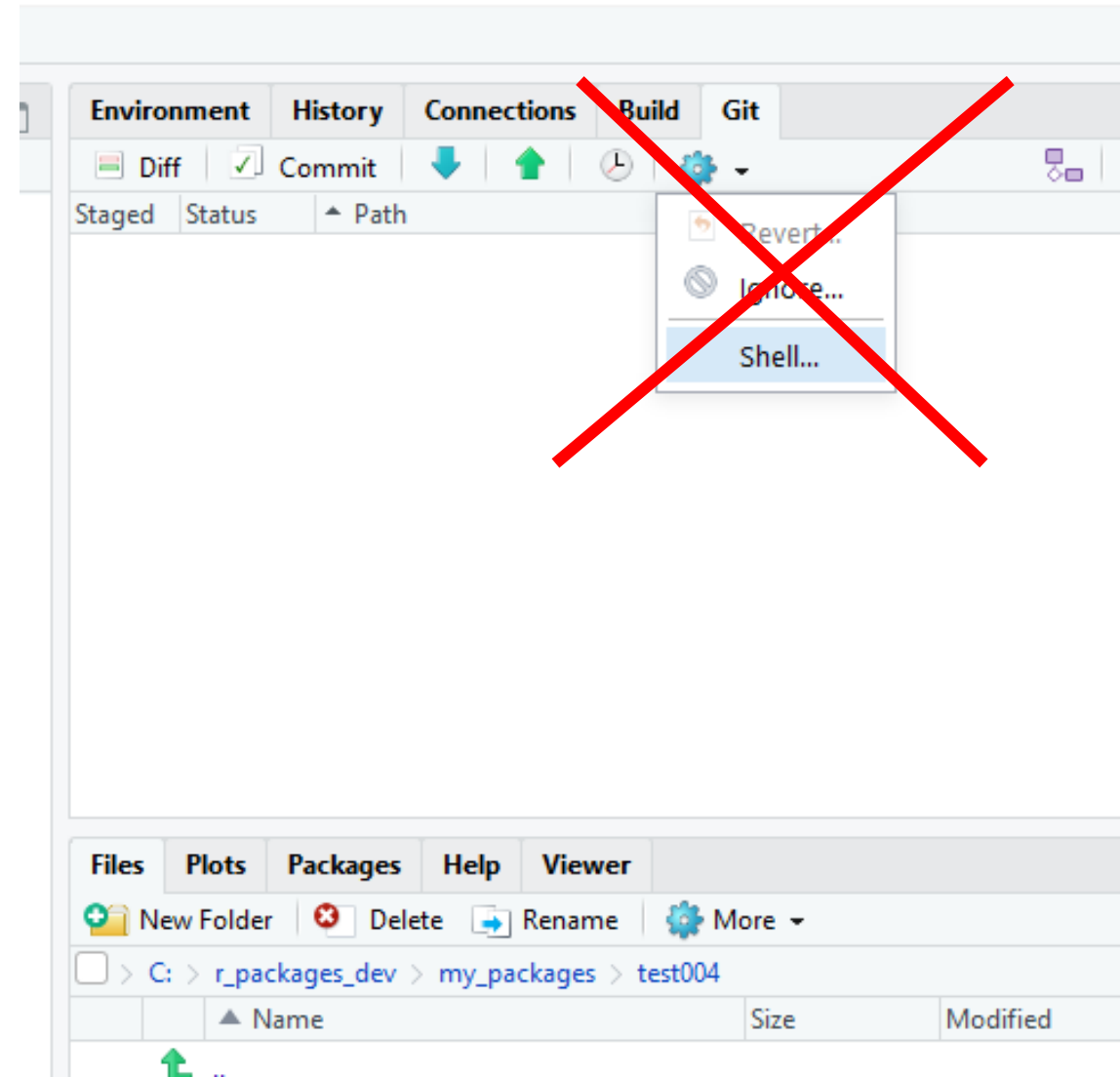
- Arrows not grayed out



Note: GitHub Connection

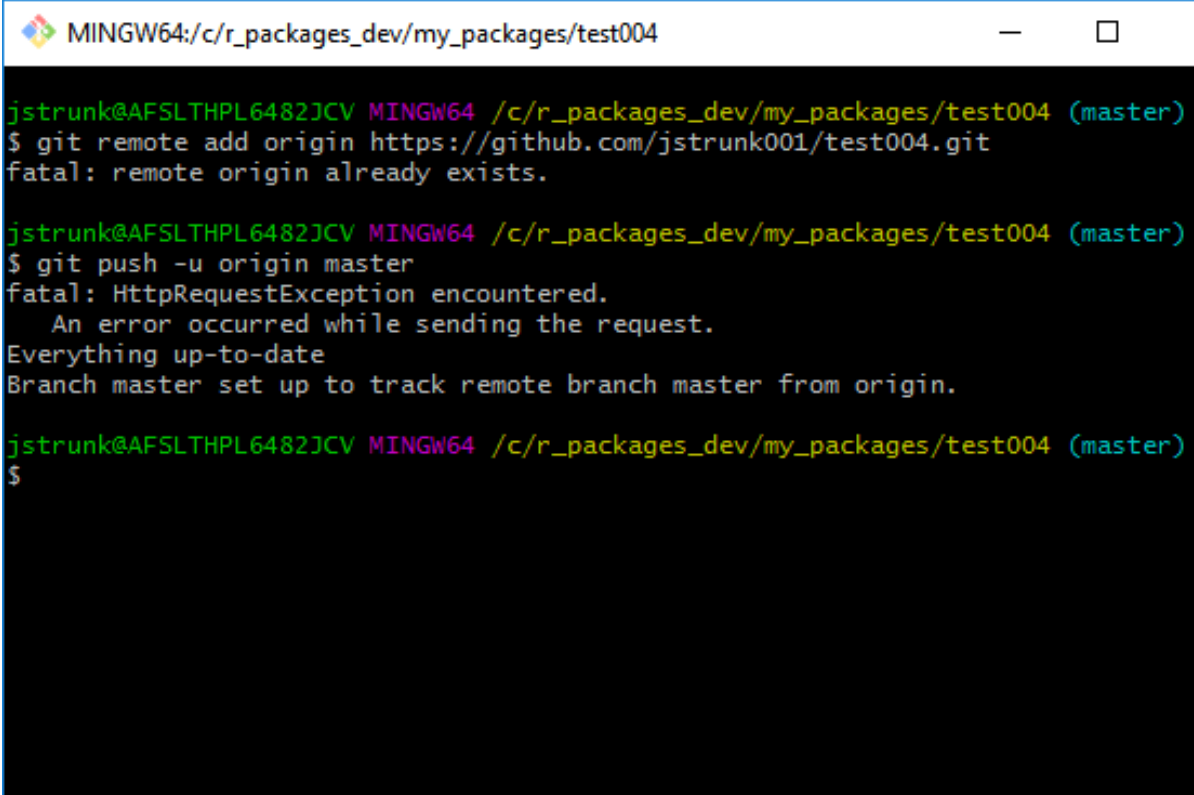
Rstudio Git Shell

- **GitHub Initiation – shell fail**
- **After GitHub – shell OK!!**



Aside: Bash Shell

- Evil command line box ?

A screenshot of a Windows command prompt window titled "MINGW64:/c/r_packages_dev/my_packages/test004". The window shows a series of Git commands and their outputs. The user is in the "master" branch. They attempt to add a remote origin, which fails with the message "fatal: remote origin already exists." They then attempt to push the master branch to the origin, which fails with the message "fatal: HttpRequestException encountered. An error occurred while sending the request. Everything up-to-date". Finally, they set up the branch master to track the remote branch master from origin.

```
jstrunk@AFSLTHPL6482JCV MINGW64 /c/r_packages_dev/my_packages/test004 (master)
$ git remote add origin https://github.com/jstrunk001/test004.git
fatal: remote origin already exists.

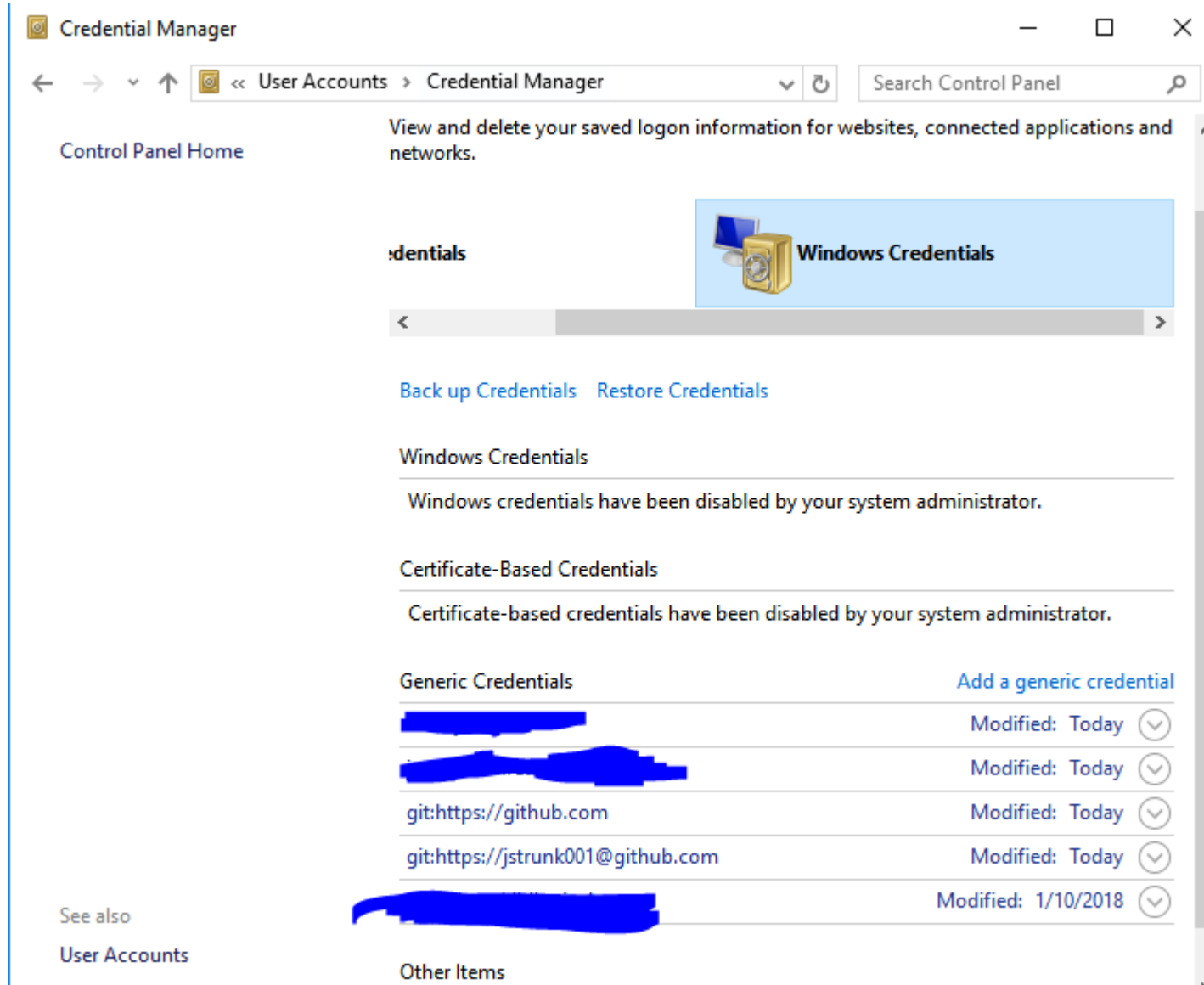
jstrunk@AFSLTHPL6482JCV MINGW64 /c/r_packages_dev/my_packages/test004 (master)
$ git push -u origin master
fatal: HttpRequestException encountered.
  An error occurred while sending the request.
Everything up-to-date
Branch master set up to track remote branch master from origin.

jstrunk@AFSLTHPL6482JCV MINGW64 /c/r_packages_dev/my_packages/test004 (master)
$
```

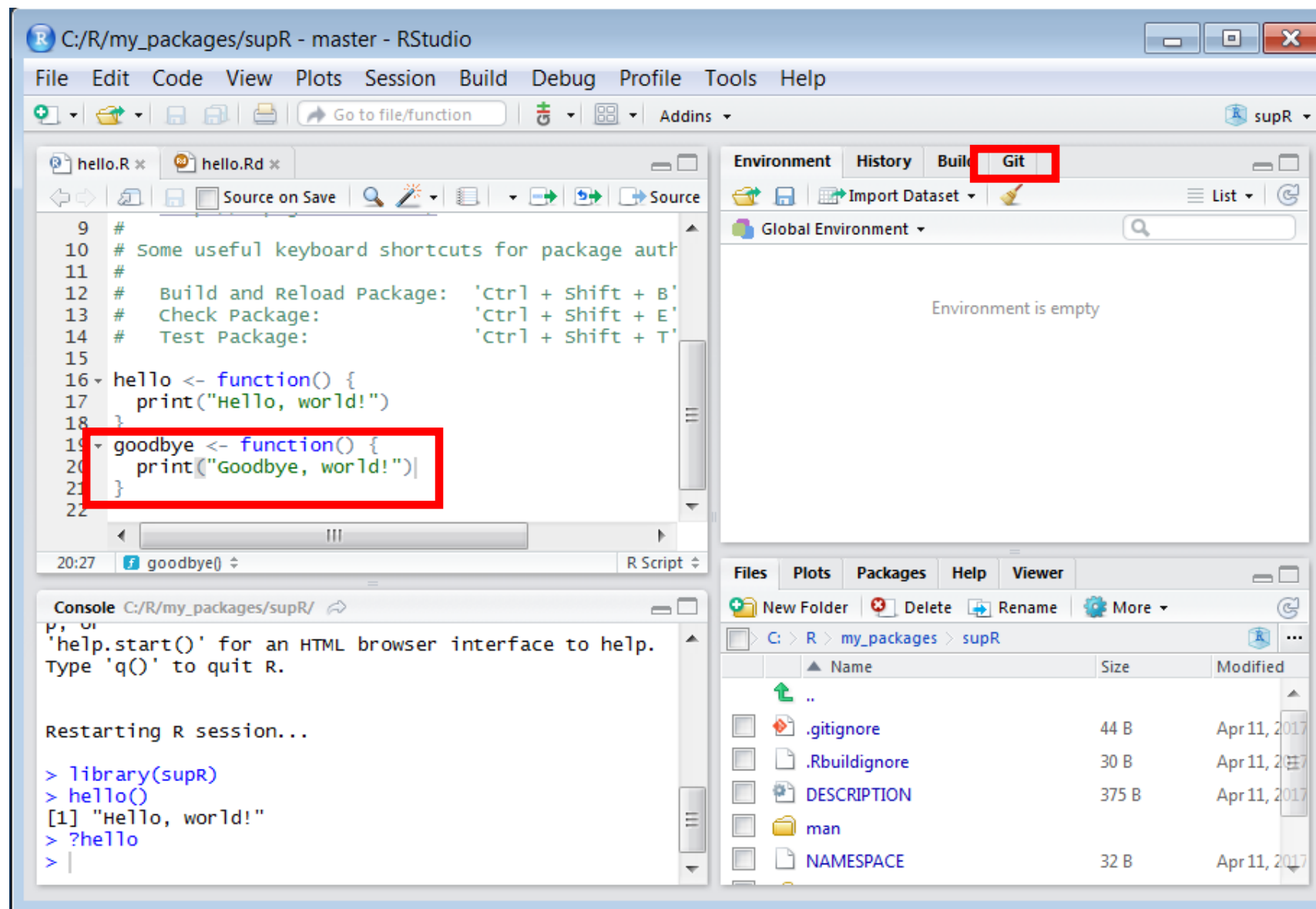
- **RARELY** (never?) absolutely needed for Rstudio / Git / Github / package!!
- **GIT GUI** – for git novices like us
- **Google** is your friend

Hint: Windows Credential Manager

- Why doesn't Git Prompt a password anymore ?
- How do I log out?



Git



The screenshot shows the RStudio interface with the following components:

- Title Bar:** C:/R/my_packages/supR - master - RStudio
- Menu Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help
- Toolbar:** Includes icons for file operations and a search bar labeled "Go to file/function".
- Source Editor:** Displays the file `hello.R` with the following code:

```
9 #  
10 # Some useful keyboard shortcuts for package author  
11 #  
12 # Build and Reload Package: 'Ctrl + Shift + B'  
13 # Check Package: 'Ctrl + Shift + E'  
14 # Test Package: 'Ctrl + Shift + T'  
15  
16 hello <- function() {  
17   print("Hello, world!")  
18 }  
19  
20 goodbye <- function() {  
21   print("Goodbye, world!")  
22 }
```

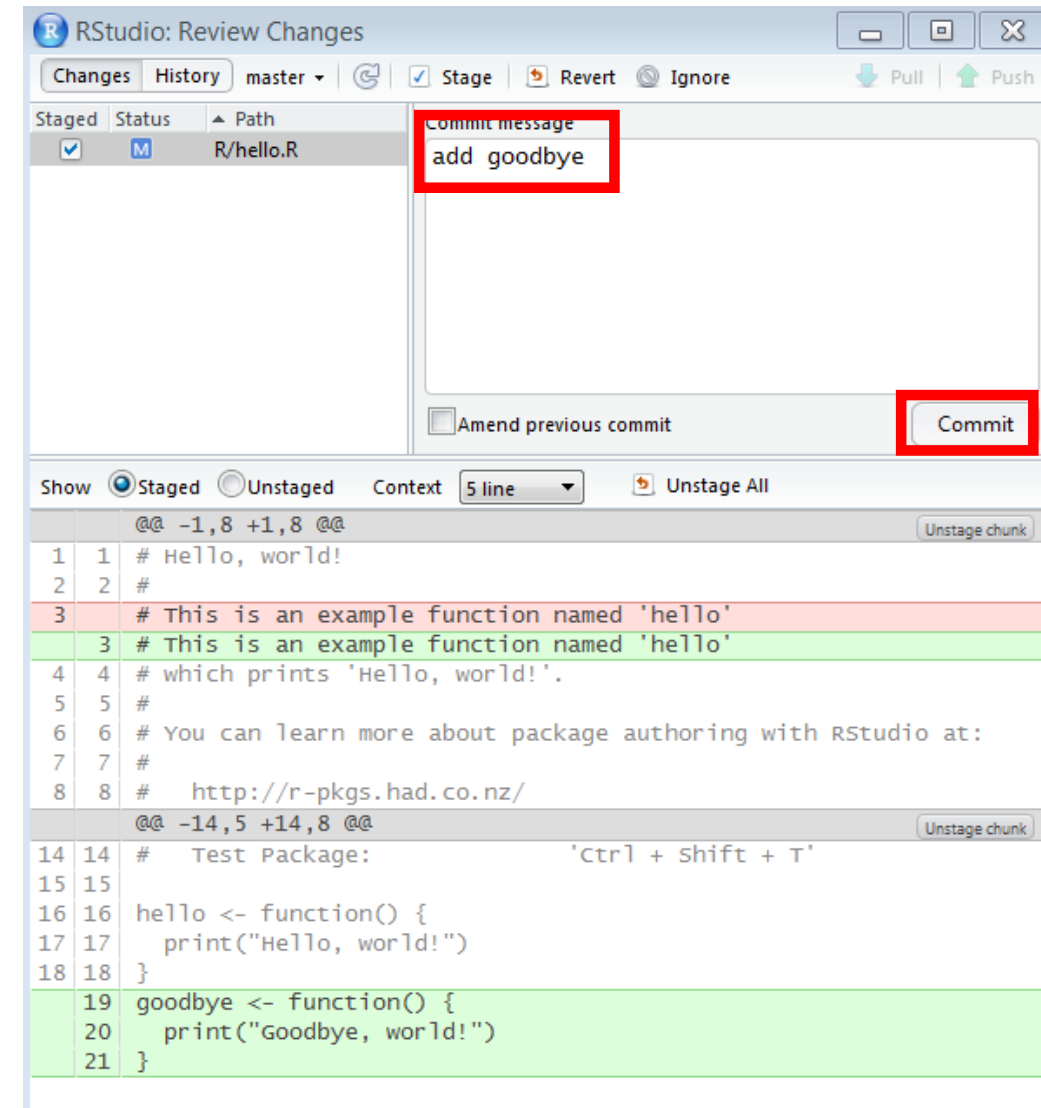
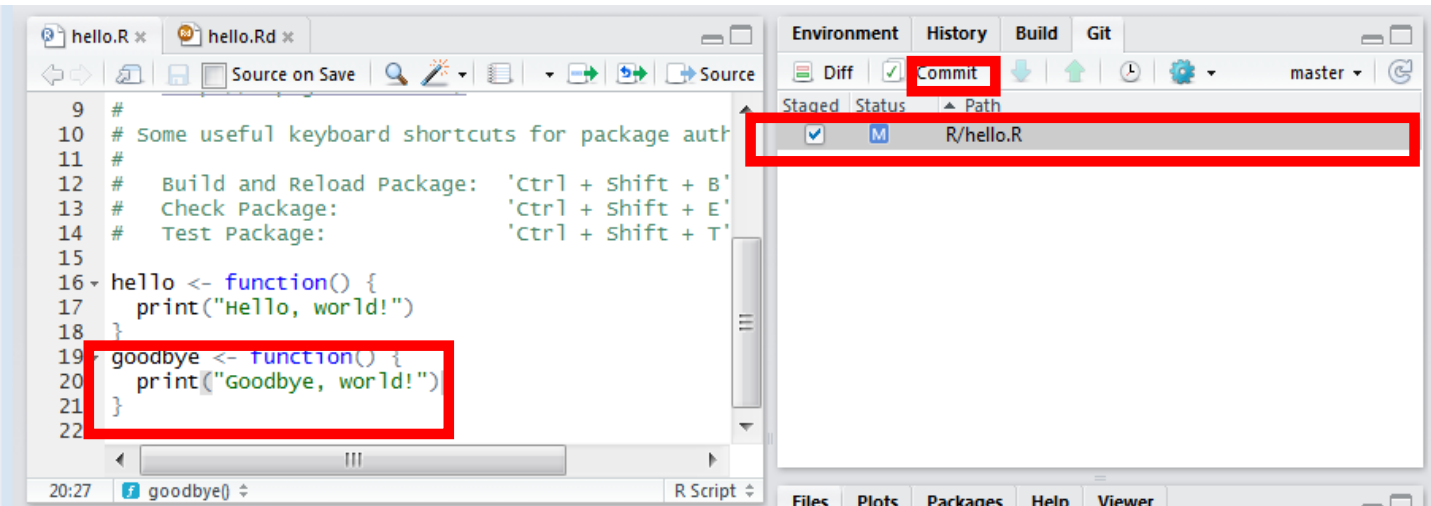
The `goodbye` function definition (lines 19-22) is highlighted with a red box.
- Environment Pane:** Located in the top right, it has tabs for Environment, History, Build, and Git. The **Git** tab is highlighted with a red box. Below the tabs, it says "Global Environment" and "Environment is empty".
- Console:** Located at the bottom, it shows the following output:

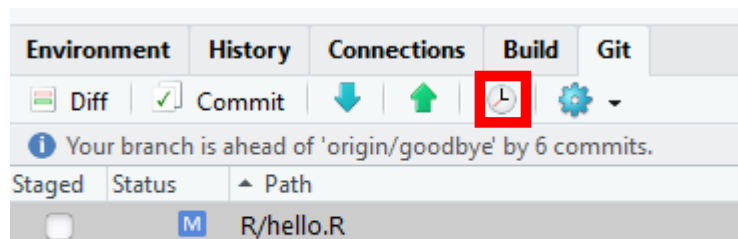
```
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
Restarting R session...  
  
> library(supR)  
> hello()  
[1] "Hello, world!"  
> ?hello  
> |
```
- Files Pane:** Located in the bottom right, it shows the file structure of the `supR` package:

Name	Size	Modified
..		
.gitignore	44 B	Apr 11, 2017
.Rbuildignore	30 B	Apr 11, 2017
DESCRIPTION	375 B	Apr 11, 2017
man		
NAMESPACE	32 B	Apr 11, 2017

Git commit (add function)

- Shift-click to select all items
 - commit





RStudio: Review Changes

Changes **History** master (all commits) Search Pull

	Subject	Author	Date	SHA
○	master asdfsa	jacobstrunk001	<120010031 2018-03-23	cf9a0f2e
○	test	jacobstrunk001	<120010031 2018-03-23	7bc14951
○	first commit	jacobstrunk001	<120010031 2018-03-23	de3677cb

Commits 1-3 of 3

SHA cf9a0f2e
Author jacobstrunk001 <12001003134353@FEDIDCARD.GOV>
Date 2018-03-23 23:02
Subject asdfsa
Parent 7bc14951

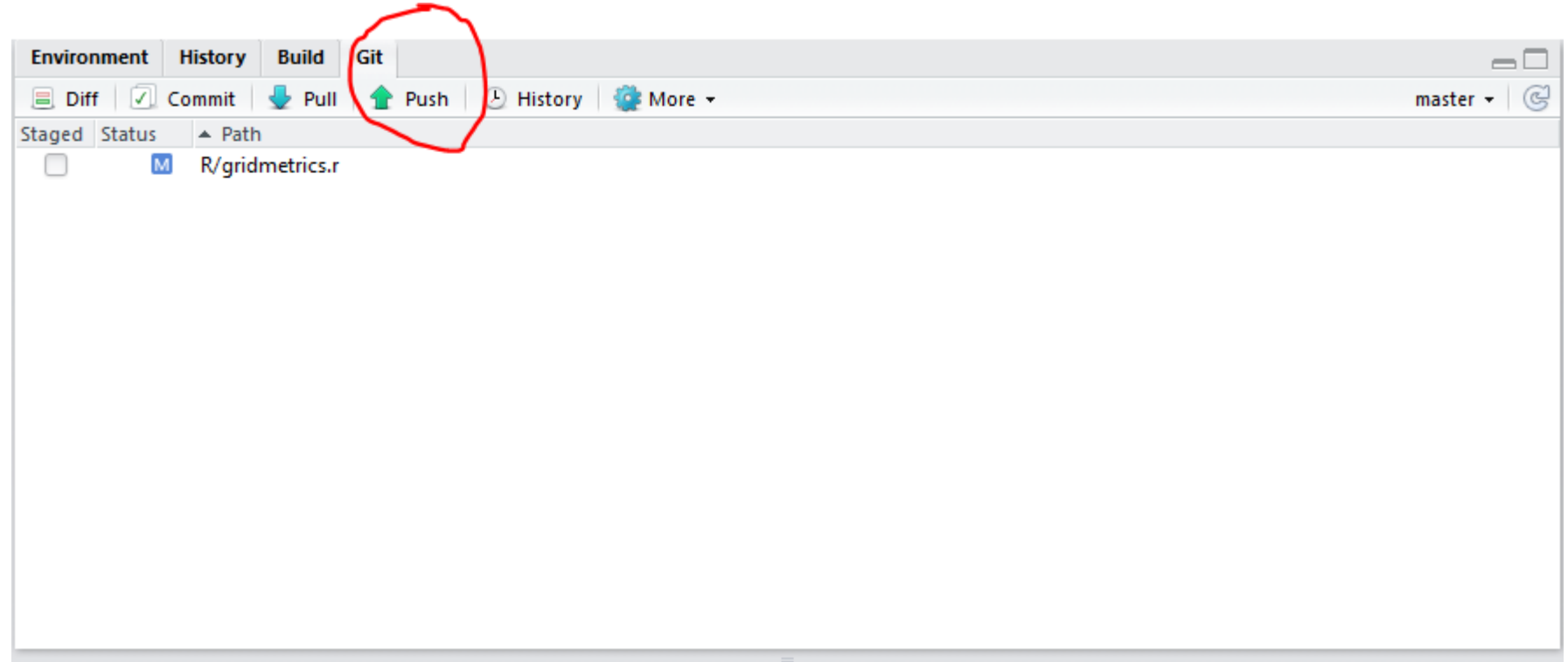
[R/hello.R](#)

R/hello.R [View file @ cf9a0f2e](#)

```
@@ -1,6 +1,6 @@
1 1 # Hello, world!
2 2 #
3 3 # This is an example function named 'hello'
3 3 # This is an example function named 'hello'
4 4 # which prints 'Hello, world!'.
5 5 #
6 6 # You can learn more about package authoring with RStudio at:
@@ -16,3 +16,4 @@
16 16 hello <- function() {
17 17   print("Hello, world!")
18 18 }
19 19 print("")
```

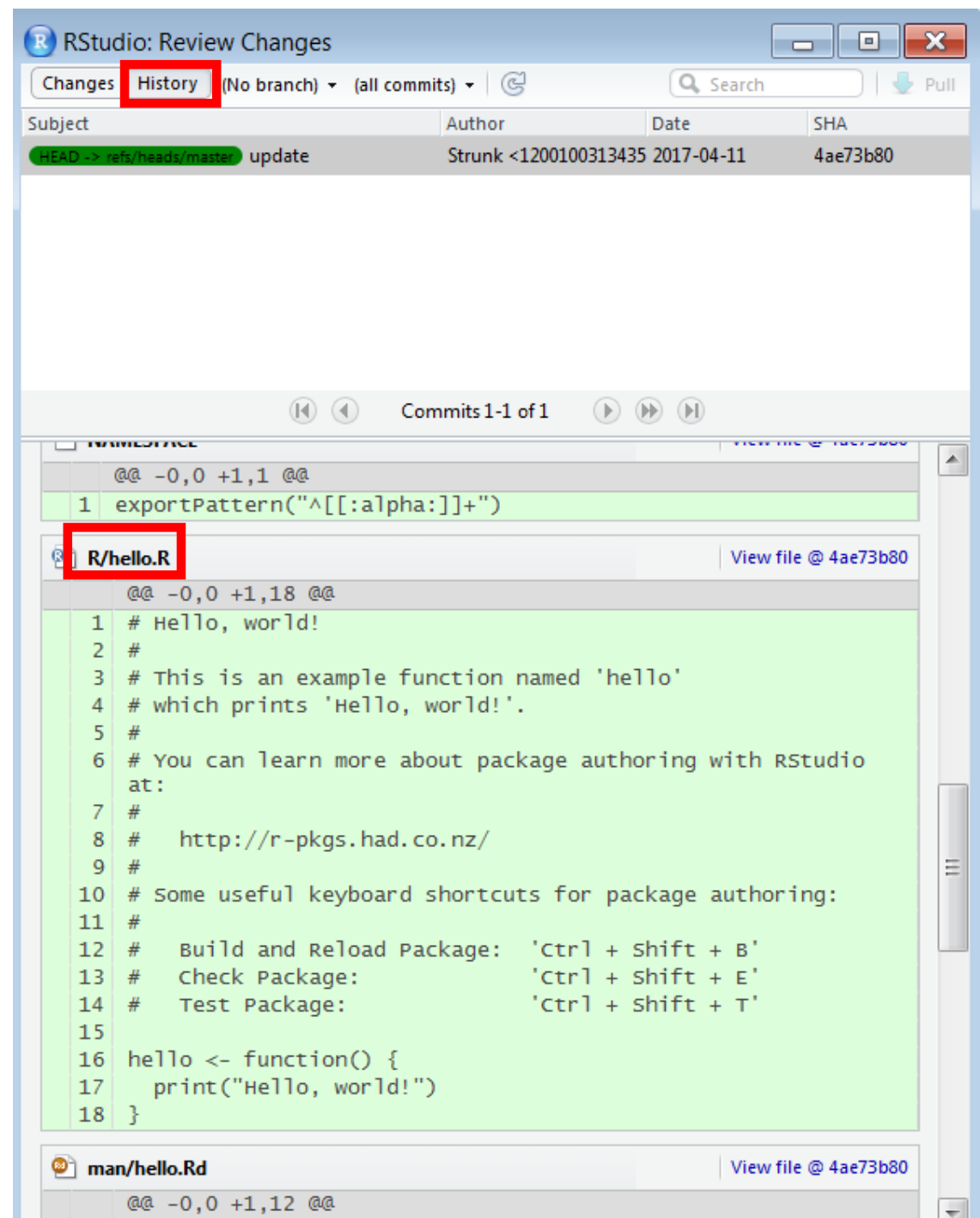
Git Push (to GitHub)

- Will need to log in first time

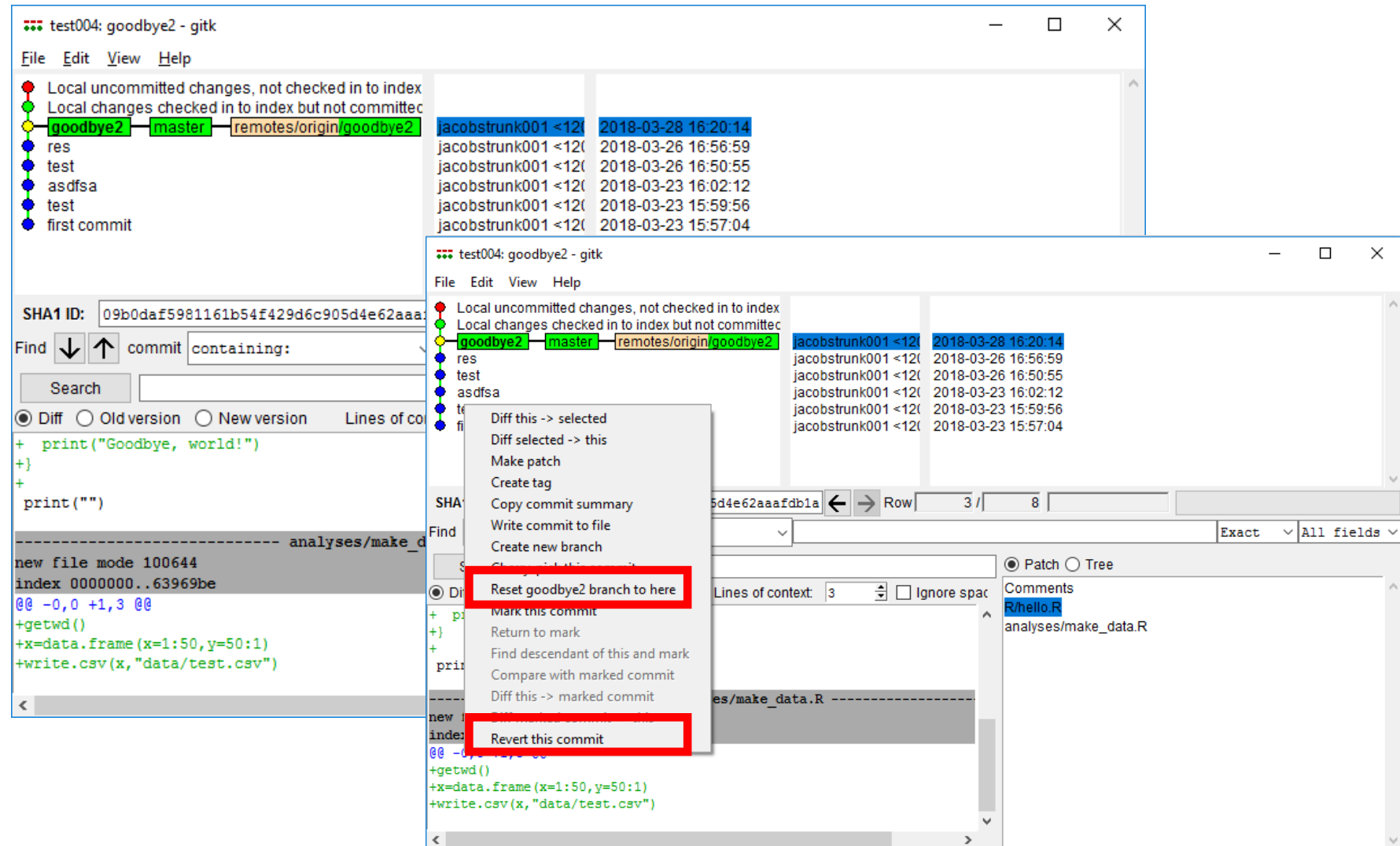


Rstudio: History

- See old versions of scripts
- Hard to copy / paste
 - Line numbers included

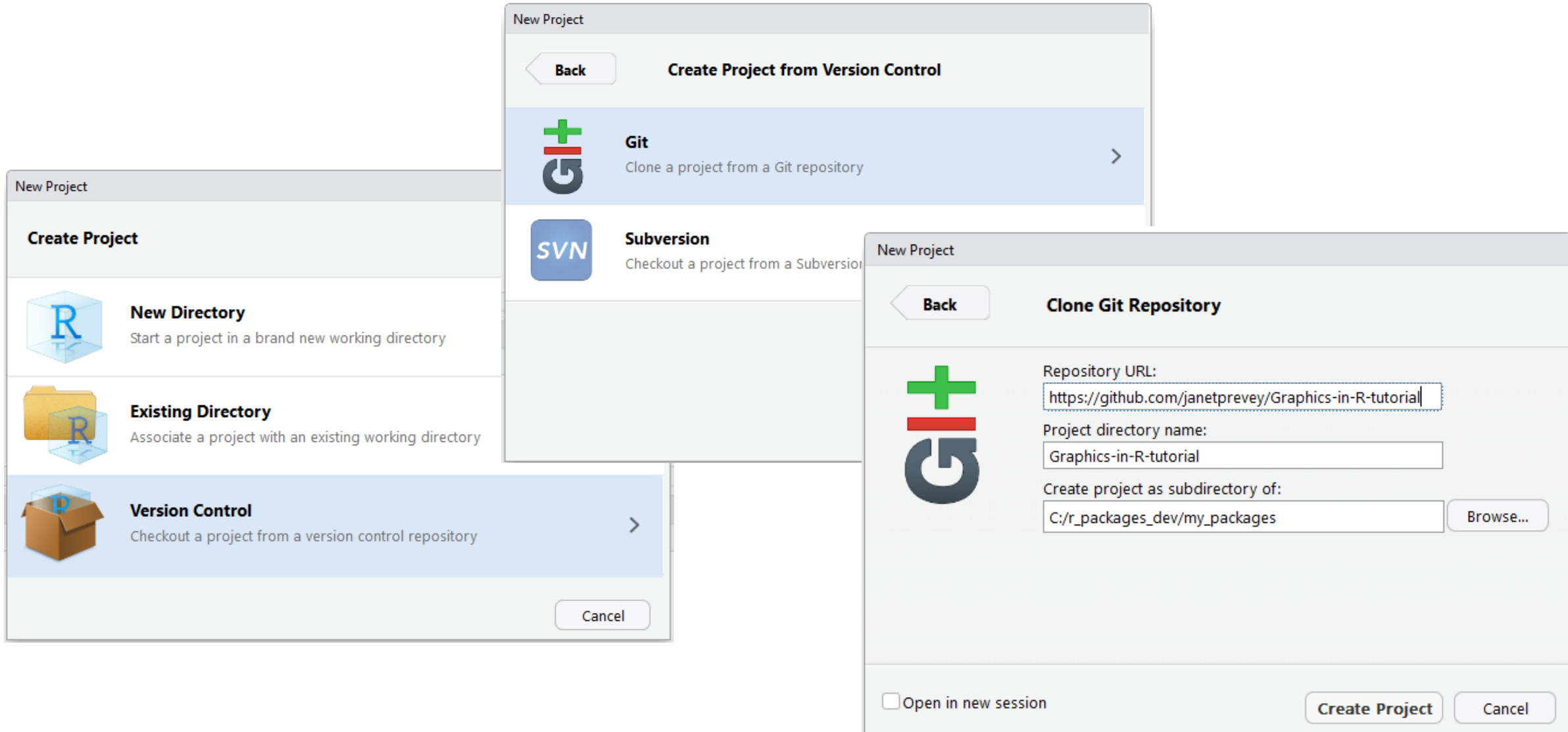


Git Gui: History



Or copy/paste

GitHub: Clone an existing repository



Roxygen Headers

- maintain one set of code
- No need to manage help files separately from code...
- Roxygen Keys -> Help Files

```
#'@name write_dtm
#'@title write USDA FUSION format raster
#'
#'@description
#'write USDA FUSION format raster
#
#
#'@details
#
# the DEM format is an unusual raster format used only by (and required by) FUSION gridmetrics. Since it is
# required and we extensively use FUSION (esp gridmetrics) we need the ability to interact with this file format.
#
# \cr
#Revision History
# \tabular{ll}{
#0.1 \tab 2017 Dec 12 - Operational \cr
#
```


Roxygen: Auto-populate Help Files (Awesome)

Just paste template into header of code and the appropriate files are auto-created

```
clean_path.R x
Source on Save
1 #'@name clean_path
2 #'@title
3 #'standardize path to file
4 #'
5 #'@description
6 #'
7 #'paths often have extra slashes, or in opposing directions.
8 #'
9 #'@details
10 #'
11 #'standardizes slashes - forward or back
12 #'
13 #'\cr
14 #'
15 #'Revision History
16 #' \tabular{ll}{
17 #'1.0 \tab 2015 Jan 29 Roxygen header prepared \cr
18 #' }
19 #'
20 #'
21 #'@author
22 #'Jacob Strunk <jstrunk@fs.fed.us>
23 #'
24 #'@param path supply a path
25 #'@param backslash T/F backslashes or forwardslashes?
26 #'@param force_endslash T/F cap path with slash?
27 #'
28 #'@return
29 #'updated path
30 #'
31 #'
```

```
Files Plots Packages Help Viewer
clean_path.Rd Find in Topic
clean_path {larsR}

standardize path to file

Description
paths often have extra slashes, or in opposing directions. This function standardizes slashes

Usage
clean_path(path, backslash = T, force_endslash = F)

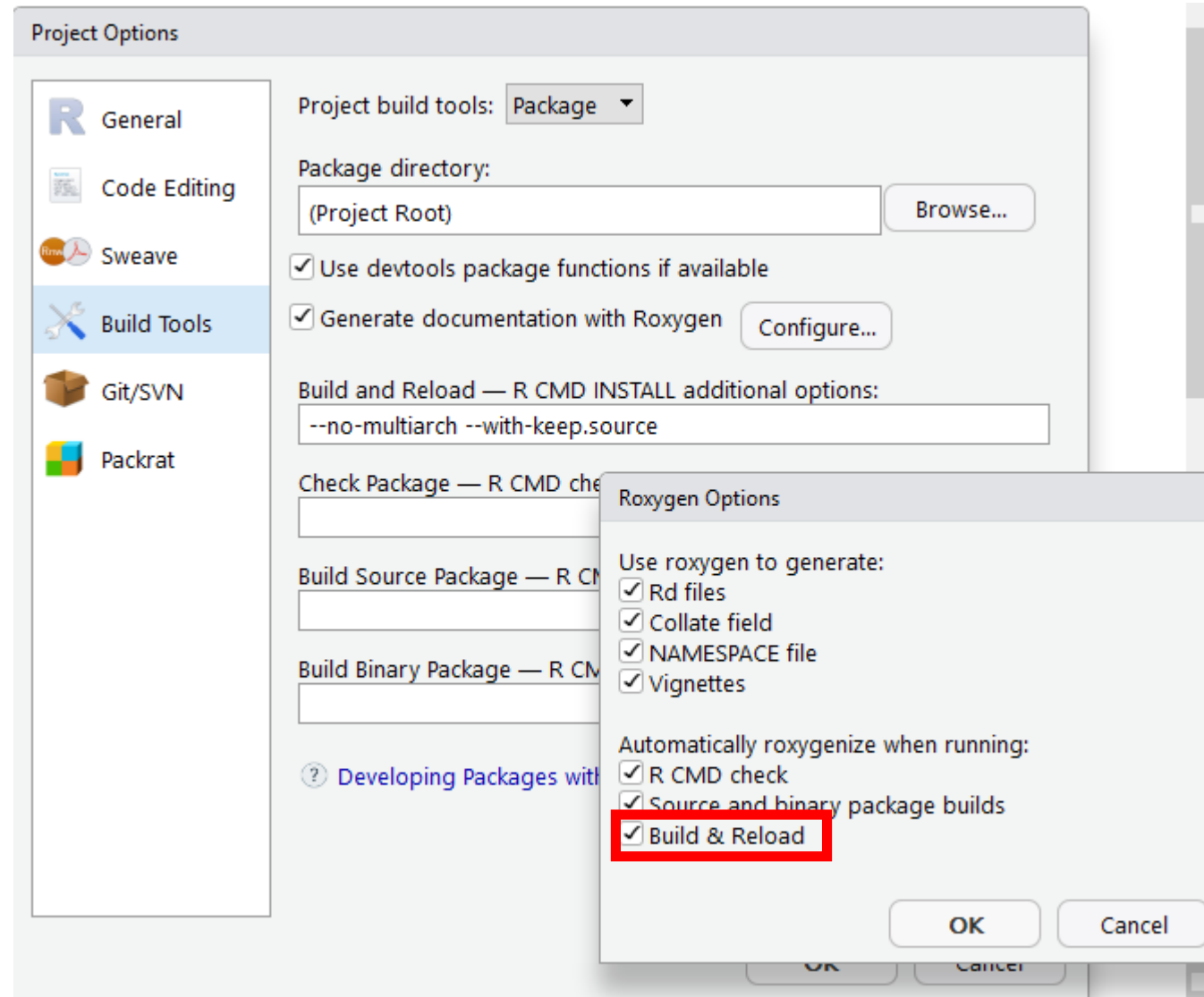
backslash(path)

forwardslash(path)

Arguments
path          supply a path
backslash     T/F backslashes or forwardslashes?
force_endslash T/F cap path with slash?

Details
```

Hint



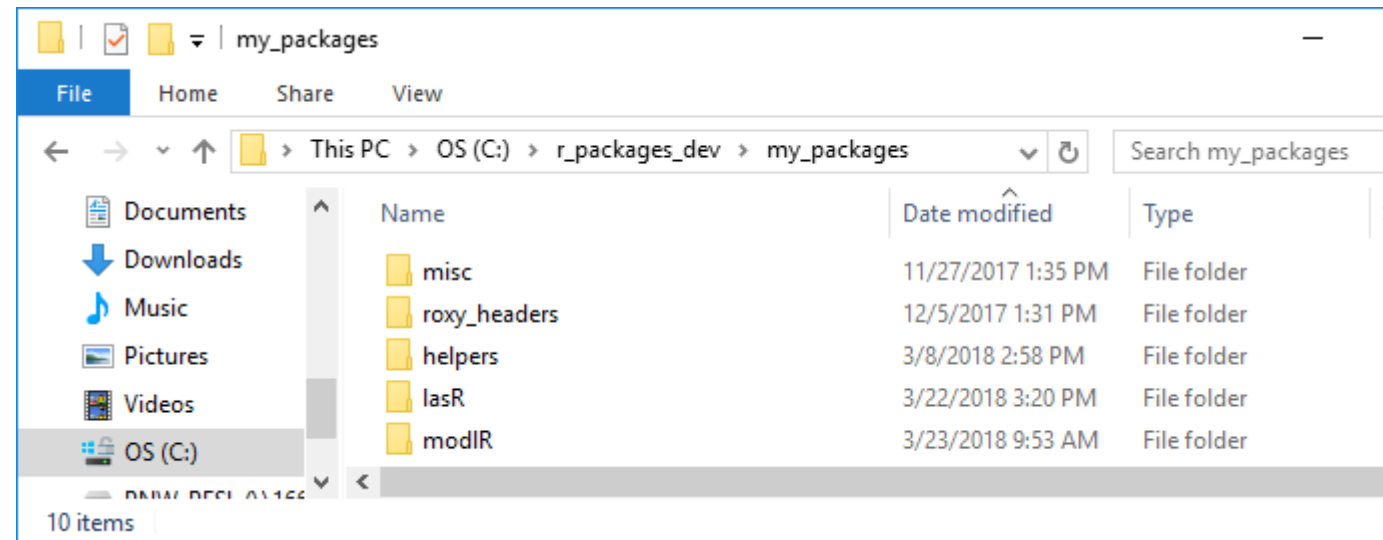
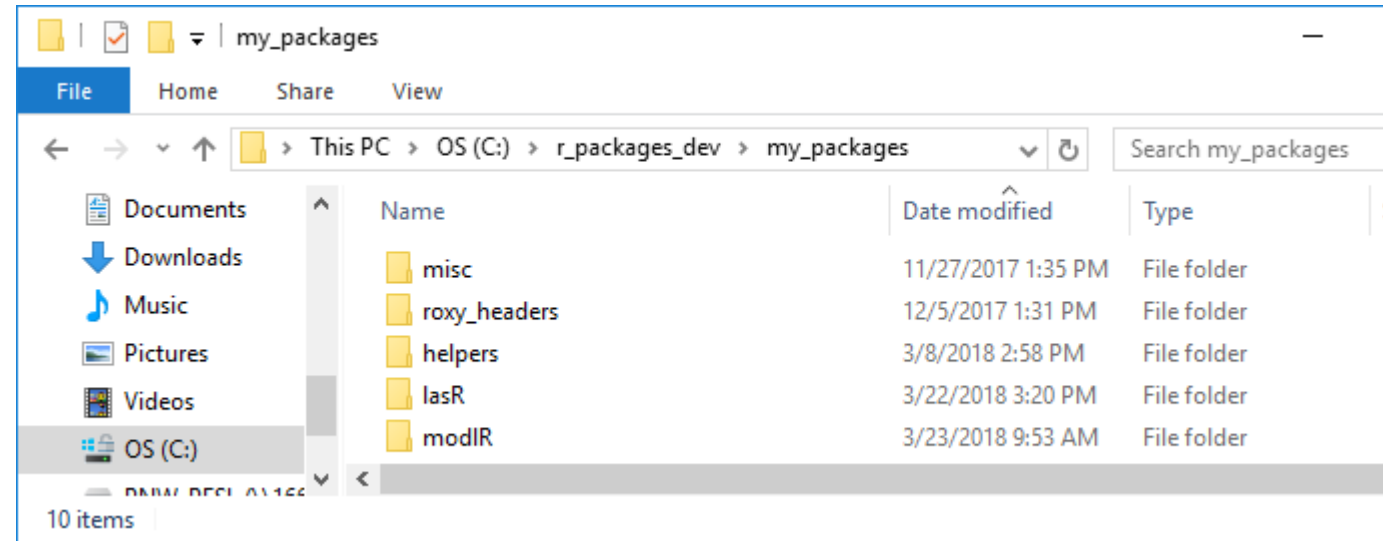
Aside: Data Backup

- Code is my most valuable resource
 - Hdd's fail
 - Accidental deletion etc.
- Nightly backups (manual) 😞
- Sync programs provide hands-off backup 😊
 - [Box](#) / [dropbox](#) / [google drive](#) / [ms onedrive](#)
 - Super convenient
 - Background processes
 - Minimal performance penalty*



Helpful Hint:

- Git doesn't play nicely with sync programs ☹
 - box/dropbox/drive/onedrive
- Approach #1
 - GitHub Push often!
- Approach #2
 - Put git directory outside of sync folder
 - Use SyncToy (etc.) to copy sync folder






r_packages_dev

All Folder Pairs

r_packages_dev

**Left Folder** C:\r_packages_dev**Right Folder** C:\box\BoxSync\sync\R\r_packages_devSyncToy Action: **Echo**[Change action...](#)

Options for this folder pair:

- ☐ All files included
- ☐ No files excluded
- ☐ Save overwritten files in the Recycle Bin
- ☐ Do not check file contents
- ☐ All subfolders included
- ☐ Active for run all
- ☐ No file attributes excluded

[Change options...](#)

Last run at: 12/15/2017 12:08 PM

Rename Folder Pair

Delete Folder Pair

Create New Folder Pair

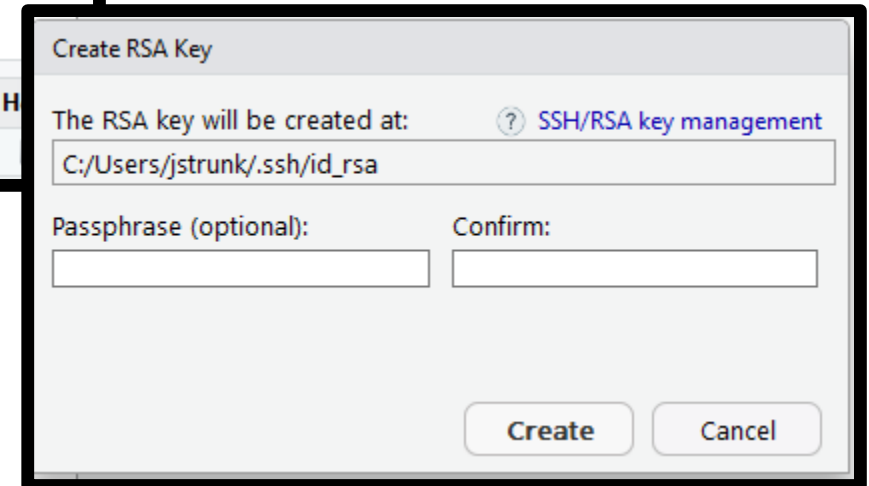
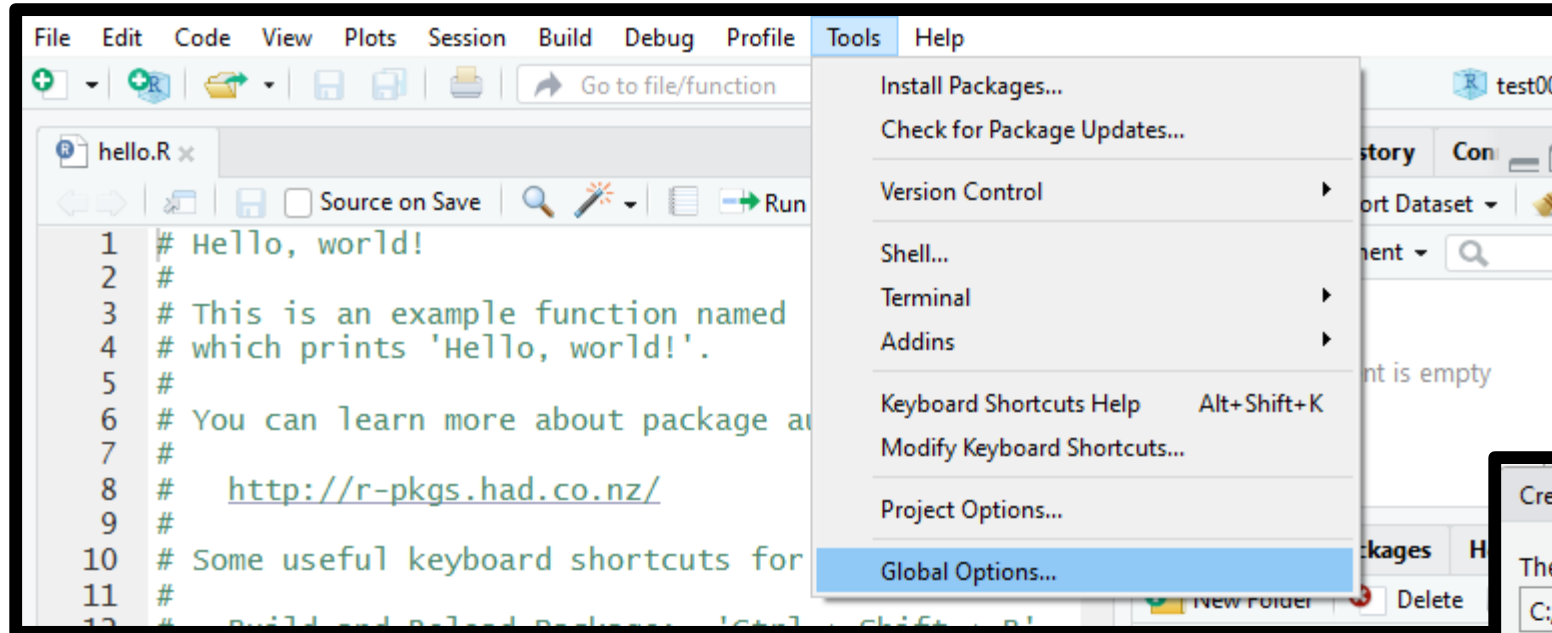
Preview

Run

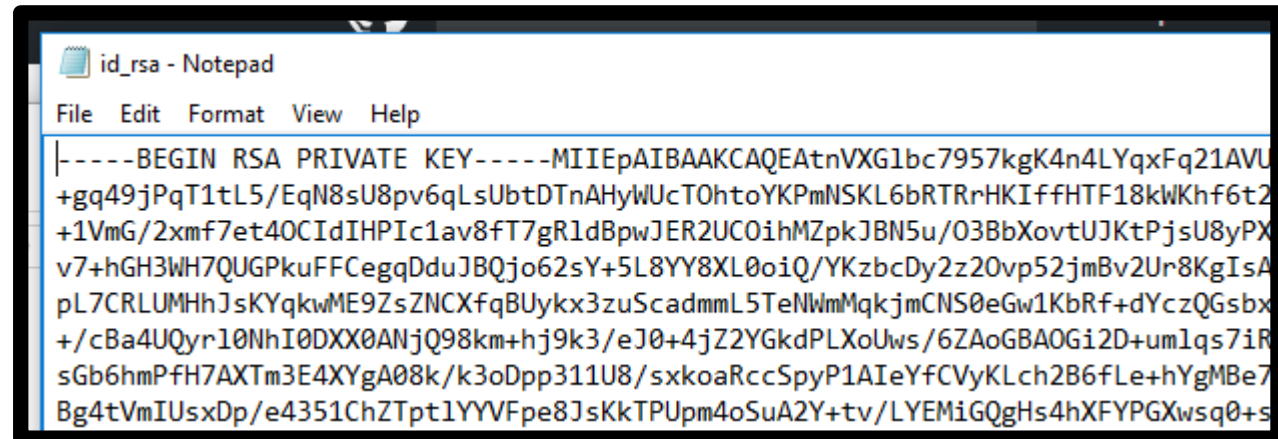
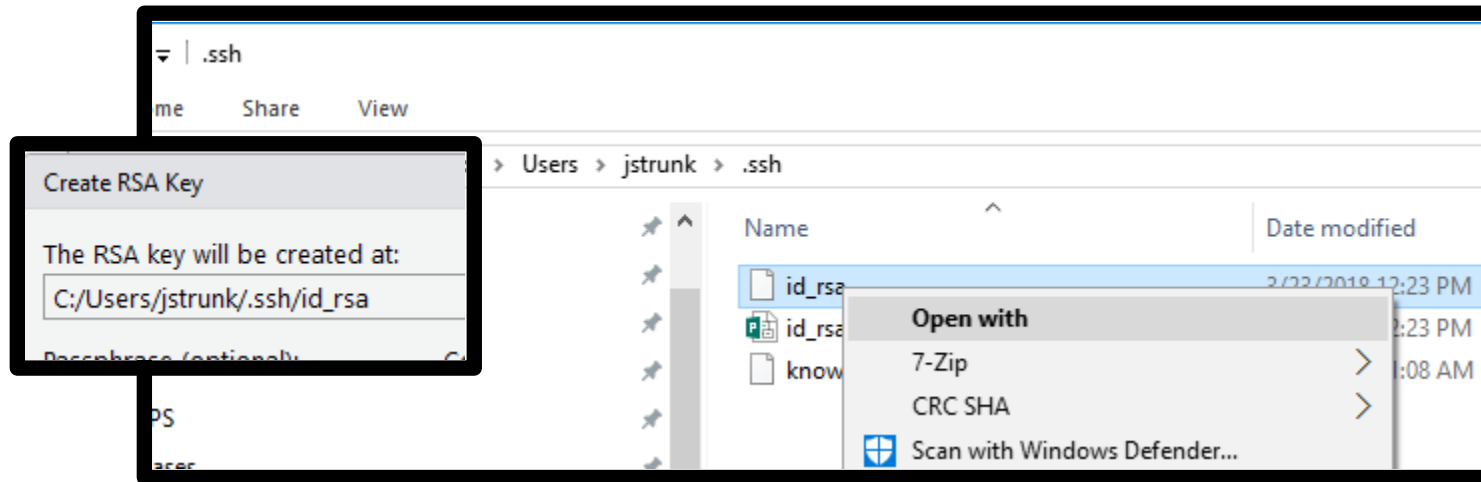
END

RSA Key?

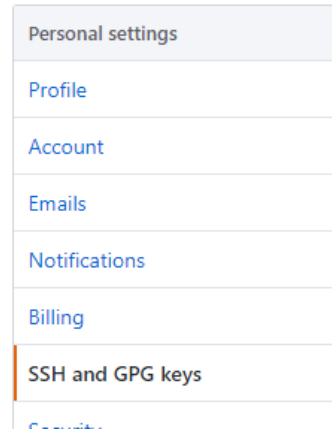
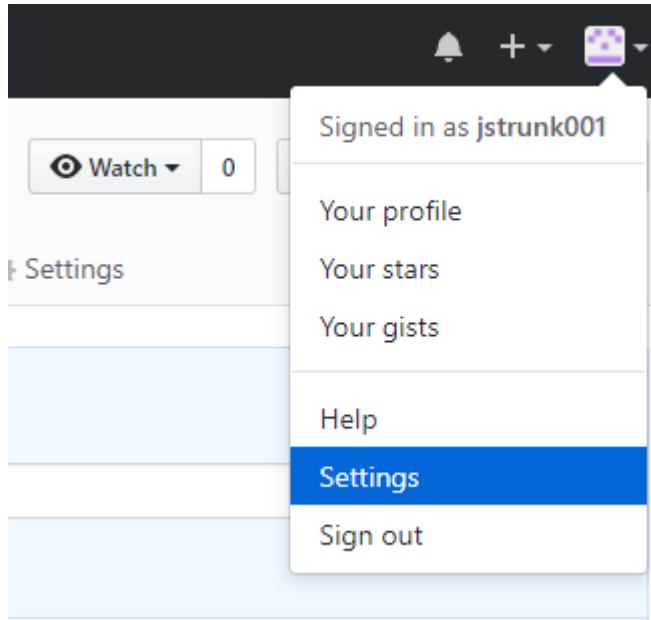
2) Create RSA Key



3) Add key to GitHub



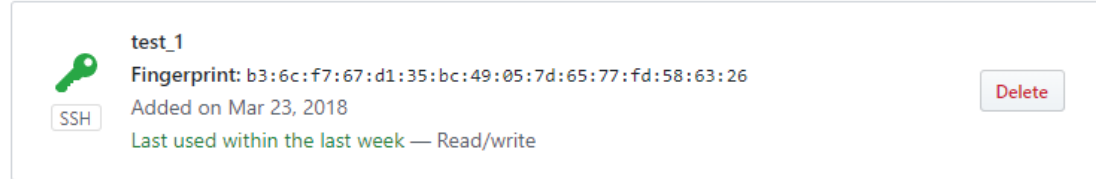
3) Add key to GitHub



SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

.ssh

me Share View

↑ This PC > OS (C:) > Users > jstrunk > .ssh

loads

PS

area

Name

Date modified

id_rsa	2/22/2019 12:23 PM
id_rsa	2:23 PM
known_hosts	1:08 AM
Open with	
7-Zip	
CRC SHA	
Scan with Windows Defender...	

Recall

