

Git Hub And R Packages

J. Lambert, *M.S., M.S.*¹

¹Department of Statistics, University of Kentucky



February 9, 2016

Overview

- GitHub
- R packages in RStudio with devtools, and roxygen2
- Using Rstudio and GitHub

What is GitHub?

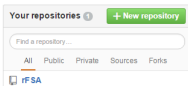
- Open source and private code/document publishing web platform.
- Cloud and desktop based version control system
- Cloud based collaboration system built in

Why use GitHub now?

- Organization
- Collaboration
- Networking
- Resume

How do I use it?

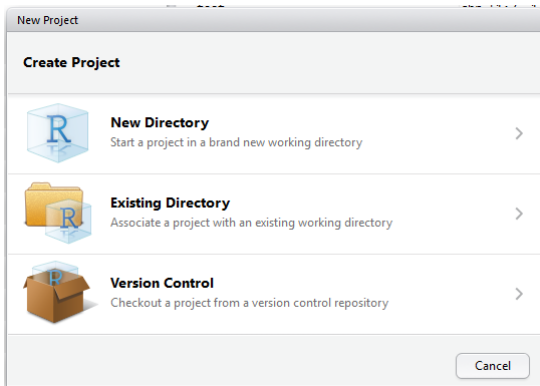
- First go to <https://github.com/> and make an account. Public version is free! Be sure to download and install git. <https://git-scm.com/downloads>.
- Make a “New Repository”. Kind of like a “New Project”.



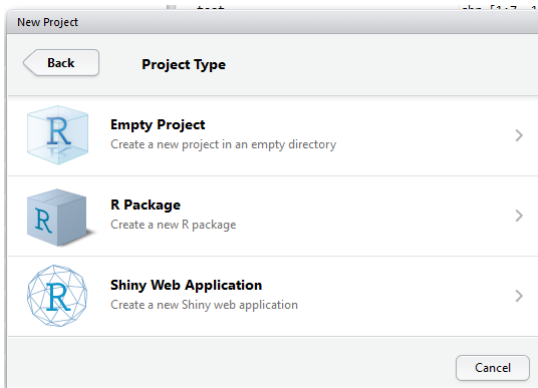
- Adding new code or updated code to current Repository is easy from Linux, Mac, or Windows.
 - Command line environments for Linux, Mac, or Windows (recommended).
 - Windows GUI
 - R package, and in Rstudio (more later)
- YouTube: Learncode.academy “Github Tutorial For Beginners.”

Getting Started

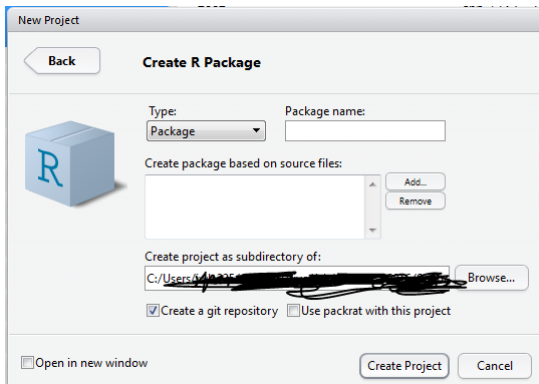
- First install two packages: Roxygen2, and devtools.
- Create a new project in Rstudio: File>New Project



New R Package



Package Information



- Be sure to select “Create a git repository”.

Configuring RStudio and GitHub

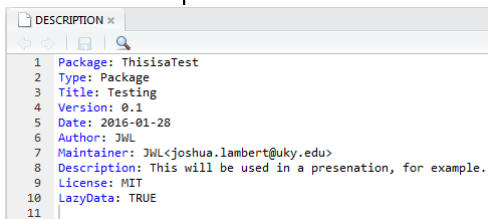
- Next, we need to configure Rstudio to use Roxogyn2 to generate documentation files.
- To Configure: Build>Configure Build Tools>Build Tools>Check “Generate documentation with Roxygen”>Check all boxes>OK.
- Next, Go to Tools>Global Options>Git/SVN and create RSA key. Enter a password you can remember. After that, click View public key and copy it.
- Go to GitHub. Go into Edit Profile on your main page. Select SSH Keys. Add SSH Key. Title it and paste the RSA key that you copied into the key field on github. Click “Add Key”.

More GitHub Configuration

- Now to connect our new package with a GitHub repository.
 - First make sure that the repository is created. You do this on GitHub.
 - Next, we will connect GitHub and our project. In RStudio, Select Git by the Environment window. Select More, then Shell. Type the following commands:
 - `git remote add origin git@github.com:githubusername/repositoryname.git`
 - `git push -u origin master` (be sure to commit changes before trying to push)
 - **Only remove origin, githubusername, and repositoryname with your a name of the repository, specific github username ,and the specific repository name.**

Building a Package

- Next, delete the hello.R and hello.rmd files from the man and R folder.
- Edit the description file

A screenshot of a text editor window titled 'DESCRIPTION x'. The window shows the content of a DESCRIPTION file, which is a standard format for R packages. The text is as follows:

```
1 Package: ThisisaTest
2 Type: Package
3 Title: Testing
4 Version: 0.1
5 Date: 2016-01-28
6 Author: JWL
7 Maintainer: JWL<joshua.lambert@uky.edu>
8 Description: This will be used in a presenation, for example.
9 License: MIT
10 LazyData: TRUE
11 |
```

The text is color-coded: 'Package:', 'Type:', 'Title:', 'Version:', 'Date:', 'Author:', 'Maintainer:', 'Description:', 'License:', and 'LazyData:' are in blue, while the values are in black. The cursor is at the end of line 11.

Using Roxygen2

- <https://cran.r-project.org/web/packages/roxygen2/vignettes/rd.html>

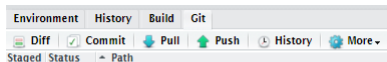
```

1 #' Square a number
2 #' @description A function to square a number
3 #' @param num Is a number.
4 #'
5 #' @details PLEASE NOTE:
6 #' @return a square of a number
7 #' @export
8 #'
9 #' @examples
10 #' square(2) #4
11 #' square(4) #16
12
13 square<-function(num){
14   num^2
15 }
16
17 #' cube a number
18 #' @description A function to cube a number
19 #' @param num a number.
20 #'
21 #' @details PLEASE NOTE:
22 #' @return a cube of a number
23 #' @export
24 #'
25 #' @examples
26 #' cube(2) #4
27 #' cube(4) #16
28
29 cube<-function(num){
30   num^3
31 }
32

```

- Test and Build the package to make sure it works correctly and the help files are showing up. Roxygen2 relies on Latex so make sure that is installed on your system before proceeding. Other errors can be combatted with google searching.

Finishing Touches



- We should now be able to make changes to our package and commit and push changes to our GitHub account.
- First, we commit pending changes with the Commit Button. Select the files you want to publish or change on your Github Repo.
- Then, when we are ready to actually change them, click Push. It will come up with a dialog box asking you for your RSA password. Enter it and click submit.
- Download your own package with `devtools::install_github("username/reponame")` .

Summary

- Check out : <http://r-pkgs.had.co.nz/> for more information about building R packages in Rstudio with Roxygen2.