

How to Build an R Package

Jihui Lee

WCM Computing Club

January 17, 2019



Weill Cornell
Medicine

Today's Objective

- ① Create a new R package: step by step
- ② Add documentation: each function in the R package
- ③ Upload the R package on GitHub for sharing

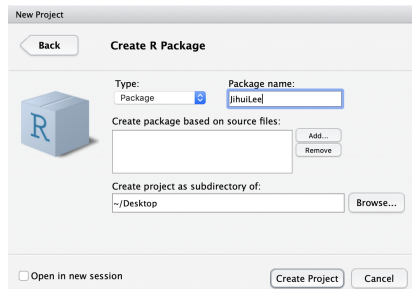
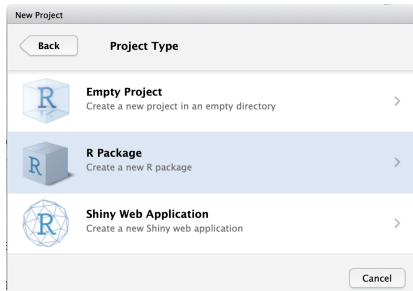
Why Your Own R Package?

- Set of frequently used R functions for one or more projects
- Repetitive plotting, hypothesis testing, model fitting. . .
- Local (or publicly available) R package

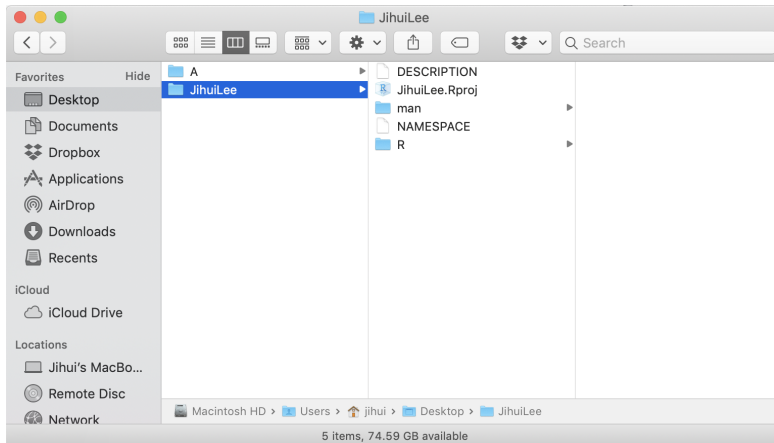
Step 0: Install Required R Packages

- *devtools*: Building and installing a package
- *roxygen2*: Documenting each function in the package

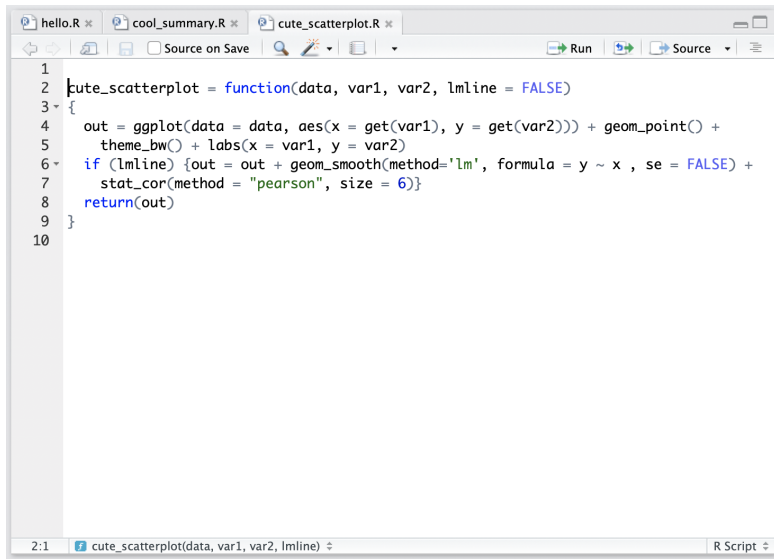
Step 1: Create an R Project




Step 1: Create an R Project



Step 2: Write a Function or More



```
1  
2 cute_scatterplot = function(data, var1, var2, lmline = FALSE)  
3 {  
4   out = ggplot(data = data, aes(x = get(var1), y = get(var2))) + geom_point() +  
5     theme_bw() + labs(x = var1, y = var2)  
6   if (lmline) {out = out + geom_smooth(method='lm', formula = y ~ x , se = FALSE) +  
7     stat_cor(method = "pearson", size = 6)}  
8   return(out)  
9 }  
10
```

2:1 |  cute_scatterplot(data, var1, var2, lmline) | R Script

Step 3: Build and Install the Package

```
> library(devtools)
> library(roxygen2)
>
> document()
Updating JihuiLee documentation
Loading JihuiLee
First time using roxygen2. Upgrading automatically...
Warning: The existing 'NAMESPACE' file was not generated by roxygen2, and will not be overwritten.
>
> setwd("../")
>
> install("JihuiLee")
Installing JihuiLee
'/Library/Frameworks/R.framework/Resources/bin/R' --no-site-file --no-environ --no-save --no-restore --quiet CMD INSTALL \
  '/Users/jihui/Desktop/JihuiLee' --library='/Library/Frameworks/R.framework/Versions/3.4/Resources/library' --install-tests

* installing *source* package 'JihuiLee' ...
** R
** preparing package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded
* DONE (JihuiLee)
Reloading installed JihuiLee
```

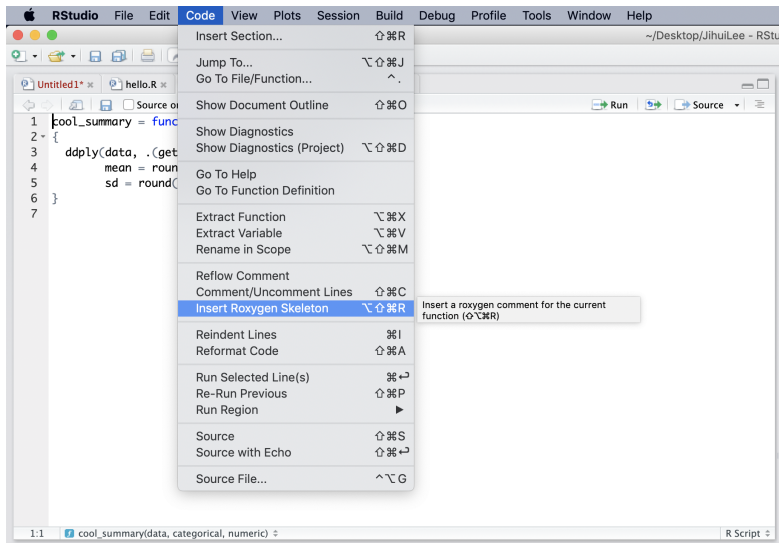

Step 3: Build and Install the Package

```
> cool_summary
function (data, categorical, numeric)
{
  ddply(data, .(categorical), summarize, mean = round(mean(get(numeric)),
    2), sd = round(sd(get(numeric)), 2))
}
<environment: namespace:JihuiLee>
```

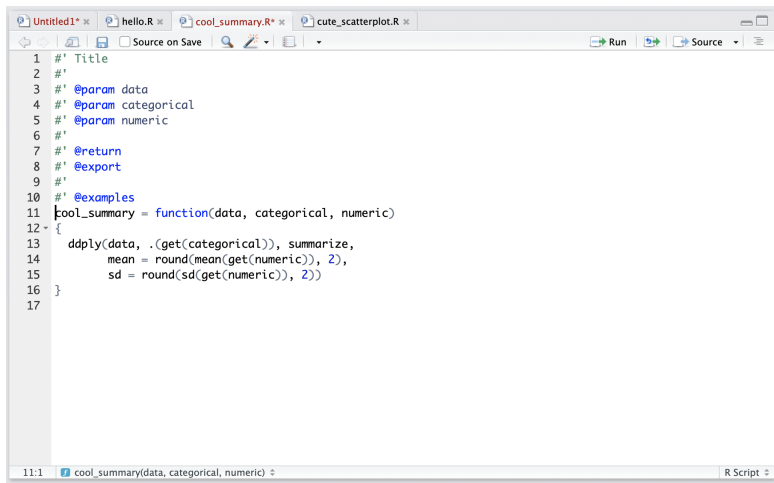
```
> ?cool_summary
No documentation for 'cool_summary' in specified packages and libraries:
you could try '??cool_summary'
```

The R package is installed, but no documentation!!

Step 4: Add Documentation



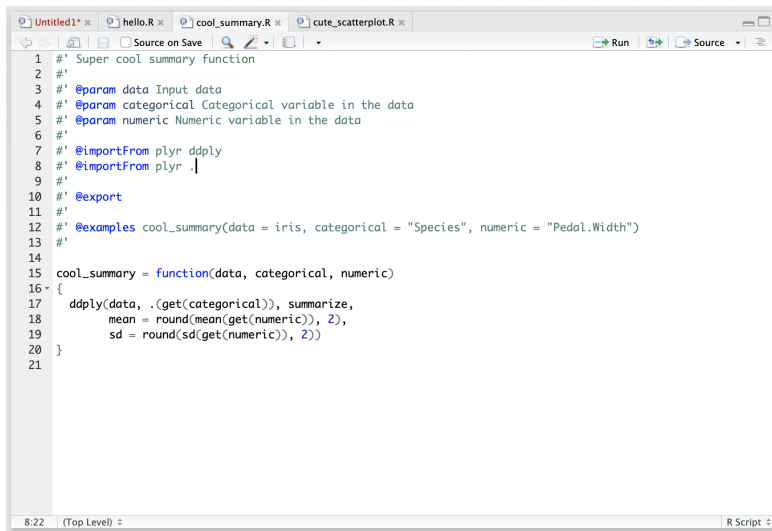
Step 4: Add Documentation (Skeleton)



```
1 #' Title
2 #'
3 #' @param data
4 #' @param categorical
5 #' @param numeric
6 #'
7 #' @return
8 #' @export
9 #'
10 #' @examples
11 cool_summary = function(data, categorical, numeric)
12 {
13   ddply(data, .(get(categorical)), summarize,
14     mean = round(mean(get(numeric)), 2),
15     sd = round(sd(get(numeric)), 2))
16 }
17
```

11:1 cool_summary(data, categorical, numeric) R Script

Step 4: Add Description



```
1 #' Super cool summary function
2 #'
3 #' @param data Input data
4 #' @param categorical Categorical variable in the data
5 #' @param numeric Numeric variable in the data
6 #'
7 #' @importFrom plyr ddply
8 #' @importFrom plyr .|
9 #'
10 #' @export
11 #'
12 #' @examples cool_summary(data = iris, categorical = "Species", numeric = "Petal.Width")
13 #'
14
15 cool_summary = function(data, categorical, numeric)
16 {
17   ddply(data, .(get(categorical)), summarize,
18     mean = round(mean(get(numeric)), 2),
19     sd = round(sd(get(numeric)), 2))
20 }
21
```

8:22 (Top Level) R Script

Step 4: Add Description

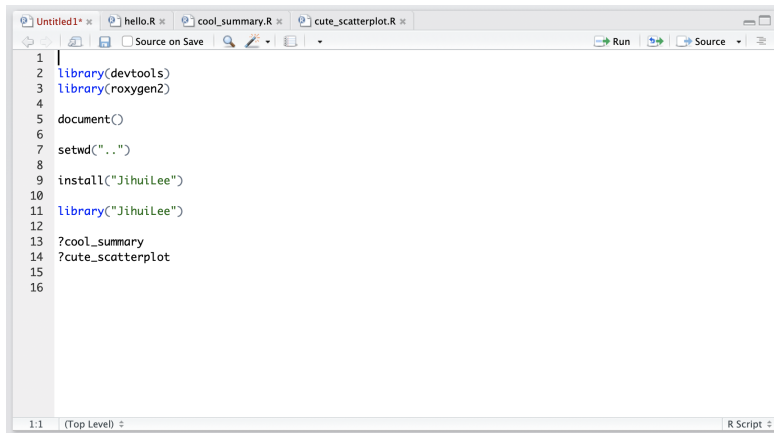
```
Untitled1* x | hello.R x | cool_summary.R x | cute_scatterplot.R x
Source on Save Run Source
1 #' Super cute scatterplot
2 #'
3 #' @param data Input data
4 #' @param var1 Numeric variable 1
5 #' @param var2 Numeric variable 2
6 #' @param lmline TRUE if you'd like to add the fitted line of linear regression. Default is FALSE.
7 #'
8 #' @importFrom ggplot2 ggplot
9 #' @importFrom ggplot2 geom_point
10 #' @importFrom ggplot2 theme_bw
11 #' @importFrom ggplot2 labs
12 #' @importFrom ggplot2 geom_smooth
13 #' @importFrom ggplot2 aes
14 #' @importFrom ggpubr stat_cor
15 #'
16 #' @export
17 #'
18 #' @examples cute_scatterplot(data = iris, var1 = "Petal.Length", var2 = "Petal.Width", lmline = TRUE)
19
20 cute_scatterplot = function(data, var1, var2, lmline = FALSE)
21 {
22   out = ggplot(data = data, aes(x = get(var1), y = get(var2))) + geom_point() +
23     theme_bw() + labs(x = var1, y = var2)
24   if (lmline) {out = out + geom_smooth(method='lm', formula = y ~ x , se = FALSE) +
25     stat_cor(method = "pearson", size = 6)}
26   return(out)
27 }
28
8:30 (Top Level) R Script
```

Step 5: Install and Check

```
> document()
Updating JihuiLee documentation
Loading JihuiLee
Writing NAMESPACE
Writing NAMESPACE
> setwd("..")
>
> install("JihuiLee")
Installing JihuiLee
'/Library/Frameworks/R.framework/Resources/bin/R' --no-site-file --no-enviro --no-save --no-restore \
--quiet CMD INSTALL '/Users/jihui/Desktop/JihuiLee' \
--library='/Library/Frameworks/R.framework/Versions/3.4/Resources/library' --install-tests

* installing *source* package 'JihuiLee' ...
** R
** preparing package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded
* DONE (JihuiLee)
Reloading installed JihuiLee
```

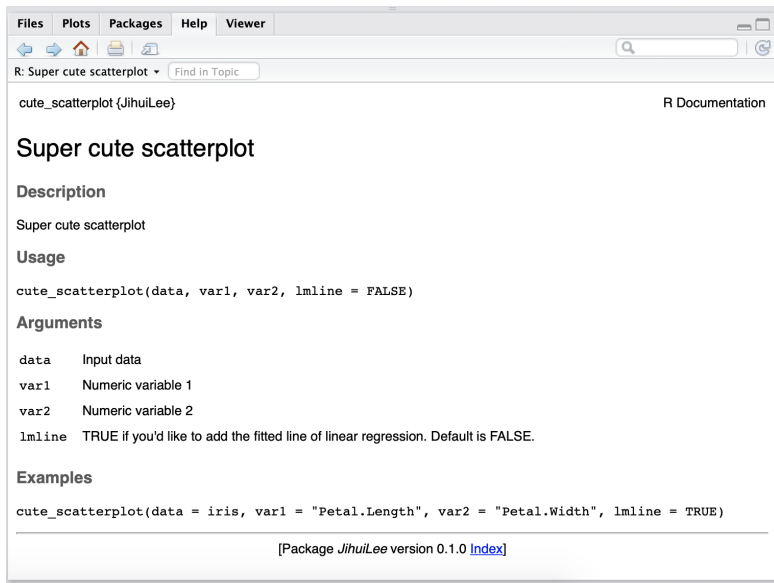
Step 5: Install and Check



```
1 |
2 library(devtools)
3 library(roxygen2)
4
5 document()
6
7 setwd("../")
8
9 install("JihuiLee")
10
11 library("JihuiLee")
12
13 ?cool_summary
14 ?cute_scatterplot
15
16
```

The screenshot shows an R Studio window with four tabs: 'Untitled1*', 'hello.R', 'cool_summary.R', and 'cute_scatterplot.R'. The 'Untitled1*' tab is active, displaying the R script. The script includes comments for 'cool_summary' and 'cute_scatterplot'. The status bar at the bottom indicates '1:1 (Top Level)' and 'R Script'.

Step 5: Install and Check



The screenshot shows the RStudio interface with the 'Packages' tab selected. The 'R: Super cute scatterplot' package is highlighted in the left sidebar. The main pane displays the documentation for the 'cute_scatterplot' function. The documentation includes a title, a description, usage instructions, arguments, and examples. The package version is noted as 0.1.0.

Files Plots Packages Help Viewer

R: Super cute scatterplot Find in Topic

cute_scatterplot {JihuiLee} R Documentation

Super cute scatterplot

Description

Super cute scatterplot

Usage

```
cute_scatterplot(data, var1, var2, lmline = FALSE)
```

Arguments

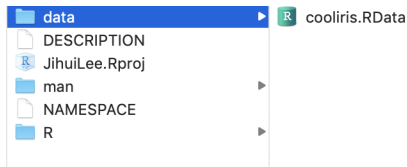
data	Input data
var1	Numeric variable 1
var2	Numeric variable 2
lmline	TRUE if you'd like to add the fitted line of linear regression. Default is FALSE.

Examples

```
cute_scatterplot(data = iris, var1 = "Petal.Length", var2 = "Petal.Width", lmline = TRUE)
```

[Package *JihuiLee* version 0.1.0 [Index](#)]

Side step: Add data sets




..Installing the package again...

```
> data("cooliris")
> head(cooliris)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

Step 6: Create a Repository

 Search or jump to... Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

Repository name


jihuilee

/ JihuiLee


Great repository names are short and memorable. Need inspiration? How about [psychic-waddle](#).

Description (optional)

My Cool Package

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None**

Add a license: **None**

Create repository




Step 7: Upload Everything

JihuiLee /



Drag additional files here to add them to your repository

Or [choose your files](#)

 /data/cooliris.RData	×
 DESCRIPTION	×
 JihuiLee.Rproj	×
 /man/cool_summary.Rd	×
 /man/cute_scatterplot.Rd	×
 /man/hello.Rd	×
 NAMESPACE	×
 /R/cute_scatterplot.R	×
 /R/hello.R	×
 /R/cool_summary.R	×

Step 7: Install the Package from GitHub

```
> devtools::install_github("jihuiLee/JihuiLee", force = TRUE)
Downloading GitHub repo jihuiLee/JihuiLee@master
from URL https://api.github.com/repos/jihuiLee/JihuiLee/zipball/master
Installing JihuiLee
'/Library/Frameworks/R.framework/Resources/bin/R' --no-site-file --no-enviro --no-save --no-restore --quiet CMD INSTALL \
  '/private/var/folders/w3/dlvly41s291gvqlfch8zj53h0000gn/T/Rtmpc0AALN/devtools3e2d24227f21/jihuiLee-JihuiLee-8781d59' \
  --library='/Library/Frameworks/R.framework/Versions/3.4/Resources/library' --install-tests

* installing *source* package 'JihuiLee' ...
** R
** data
*** moving datasets to lazyload DB
** preparing package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded
* DONE (JihuiLee)
Reloading installed JihuiLee
```

Now you can share your package with everyone!

Let's Practice!

- 1 Create a GitHub account
- 2 Build a local R package
- 3 Upload the package in GitHub
- 4 Install one another's R packages from GitHub

- Not So Standard Deviations
- Developing Packages with RStudio