

Writing R packages in Rstudio

https://stirlingcodingclub.github.io/SCC_R_package/notebook/Rpackage_notes.html

Brad Duthie

4 May 2022

Introduction: What is an R package?

- ▶ Bundles of code and data that can be written by anyone in the R community

Introduction: What is an R package?

- ▶ Bundles of code and data that can be written by anyone in the R community
- ▶ Held in online repositories
 - ▶ Comprehensive R Archive Network (CRAN)
`(install.packages())`
 - ▶ GitHub (`devtools::install_github()`)

Introduction: What is an R package?

- ▶ Bundles of code and data that can be written by anyone in the R community
- ▶ Held in online repositories
 - ▶ Comprehensive R Archive Network (CRAN)
(`install.packages()`)
 - ▶ GitHub (`devtools::install_github()`)
- ▶ CRAN currently holds over 13000 R packages

Introduction: What is an R package?

- ▶ Bundles of code and data that can be written by anyone in the R community
- ▶ Held in online repositories
 - ▶ Comprehensive R Archive Network (CRAN)
(`install.packages()`)
 - ▶ GitHub (`devtools::install_github()`)
- ▶ CRAN currently holds over 13000 R packages
- ▶ R packages are wide ranging
 - ▶ Massive multi-person development efforts for large scale coding projects
 - ▶ Small joke packages with minimal code
 - ▶ Personal packages customised for an individual user

Two packages that need to be installed

The `devtools` package:

Devtools makes package development a breeze: it works with R's existing conventions for code structure, adding efficient tools to support the cycle of package development. With devtools, developing a package becomes so easy that it will be your default layout whenever you're writing a significant amount of code.

Two packages that need to be installed

The **devtools** package:

Devtools makes package development a breeze: it works with R's existing conventions for code structure, adding efficient tools to support the cycle of package development. With devtools, developing a package becomes so easy that it will be your default layout whenever you're writing a significant amount of code.

The **roxygen2** package:

The goal of roxygen2 is to make documenting your code as easy as possible. R provides a standard way of documenting packages: you write .Rd files in the man/ directory.

The most basic R package

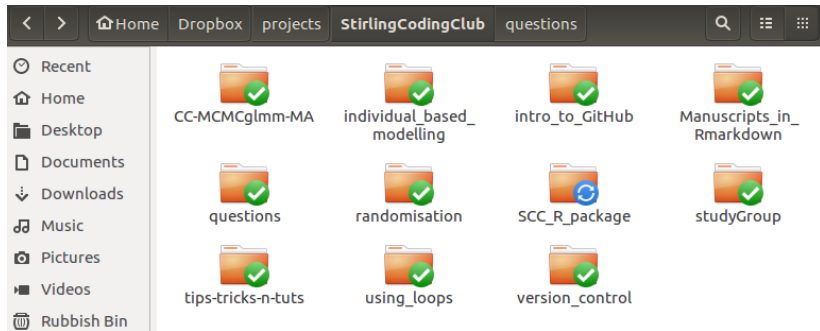
Goal: Create an R package that includes two functions.

```
# Convert Fahrenheit to Celsius
F_to_C <- function(F_temp){
  C_temp <- (F_temp - 32) * 5/9;
  return(C_temp);
}

# Convert Celsius to Fahrenheit
C_to_F <- function(C_temp){
  F_temp <- (C_temp * 9/5) + 32;
  return(F_temp);
}
```

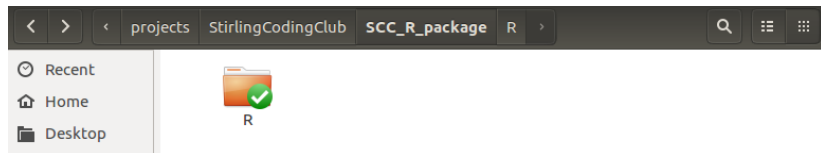

The most basic R package

First, create a new folder in your computer



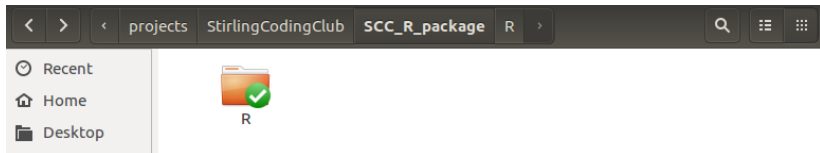
The most basic R package

Next, add a folder called 'R'

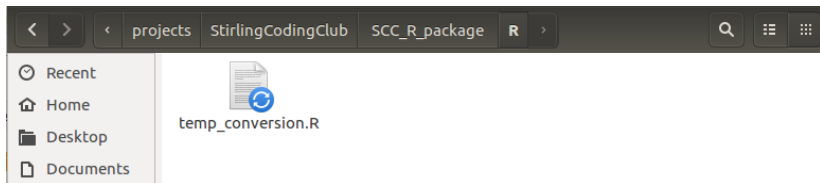


The most basic R package

Next, add a folder called 'R'



Then drop the R scripts into the 'R' folder



The most basic R package

Add a file called DESCRIPTION to the outer directory. This is a plain text file that can just be a few lines of code.

Package: SCCTempConverter

Type: Package

Title: Temperature Conversion Package for Demonstration

Version: 0.0.1.0

RoxygenNote: 6.1.0

It can also include [a lot more information](#) (e.g., authors, maintainers, extended description, website, etc.) if need be.

The most basic R package

This is now *already* an R package, which we can load.

```
# Working dir should be SCC_R_package  
library(devtools);  
load_all(".");
```

The most basic R package

This is now *already* an R package, which we can load.

```
# Working dir should be SCC_R_package  
library(devtools);  
load_all(".");
```

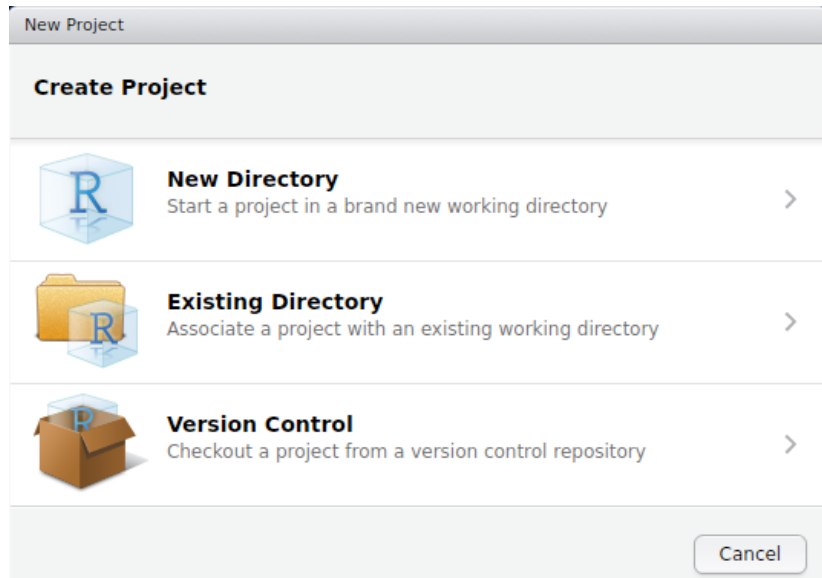
Once loaded, we can start using the R functions in the package.

```
C_to_F(20);
```

```
## [1] 68
```

Creating a new R project

To do this in Rstudio, go to `File > New Project...`




Creating a new R project

To do this in Rstudio, go to File > New Project...

New Project

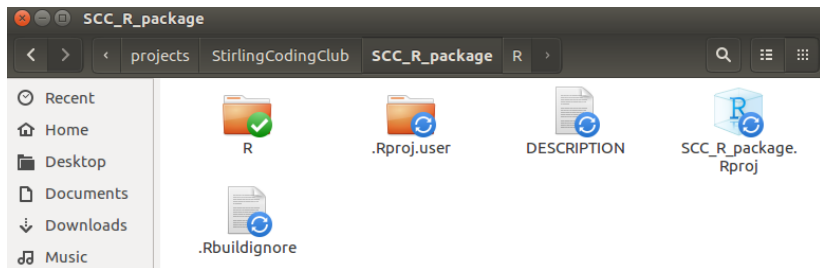
Back **Create Project from Existing Directory**

 Project working directory:
~/Dropbox/projects/StirlingCodingClub/SCC_R_pac **Browse...**

☐ Open in new session **Create Project** **Cancel**

Creating a new R project

To do this in Rstudio, go to `File > New Project...`



Adding some minimal documentation

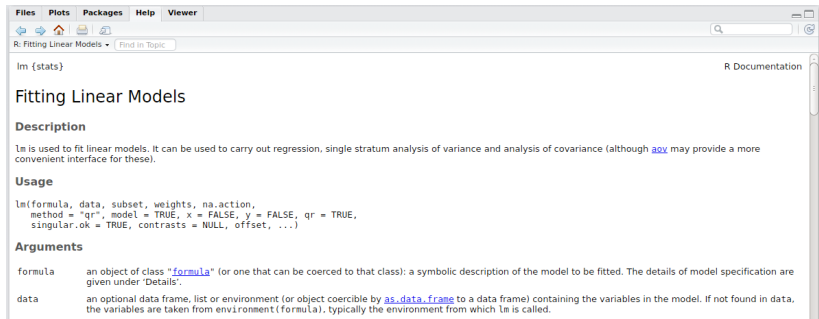
Documentation shows up in the 'Help' tab of RStudio when running the function `help`

```
help(lm);
```

Note that the code below does the same thing as the code above.

```
?lm
```

Adding some minimal documentation



The screenshot shows the R Documentation window for the `lm` function. The window has a menu bar with 'Files', 'Plots', 'Packages', 'Help', and 'Viewer'. Below the menu bar is a toolbar with icons for back, forward, home, search, and print. The main content area displays the title 'lm {stats}', the subtitle 'Fitting Linear Models', and the text 'R Documentation'. The 'Description' section explains that `lm` is used to fit linear models. The 'Usage' section shows the function signature: `lm(formula, data, subset, weights, na.action, method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE, contrasts = NULL, offset, ...)`. The 'Arguments' section lists the parameters: `formula` (an object of class "formula") and `data` (an optional data frame, list or environment).

Files Plots Packages Help Viewer

R: Fitting Linear Models Find in Topic

lm {stats} R Documentation

Fitting Linear Models

Description

`lm` is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance (although [aov](#) may provide a more convenient interface for these).

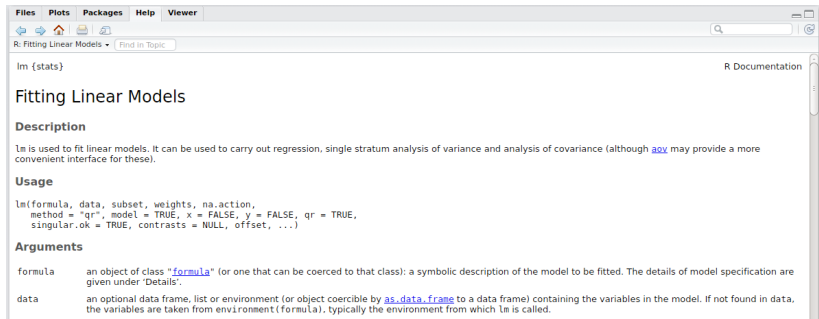
Usage

```
lm(formula, data, subset, weights, na.action,
    method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,
    singular.ok = TRUE, contrasts = NULL, offset, ...)
```

Arguments

formula	an object of class " formula " (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in <code>data</code> , the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>lm</code> is called.

Adding some minimal documentation



Create help files in Rstudio using the roxygen2 package

```
install.packages("roxygen2");  
library(roxygen2);
```

Adding some minimal documentation

```
#' Fahrenheit conversion  
#'  
#' Convert degrees Fahrenheit temp to degrees Celsius  
#' @param F_temp The temperature in degrees Fahrenheit  
#' @return The temperature in degrees Celsius  
#' @examples  
#' temp1 <- F_to_C(50);  
#' temp2 <- F_to_C( c(50, 63, 23) );  
#' @export  
F_to_C <- function(F_temp){  
  C_temp <- (F_temp - 32) * 5/9;  
  return(C_temp);  
}
```

Adding some minimal documentation

If we load our package and type the following:

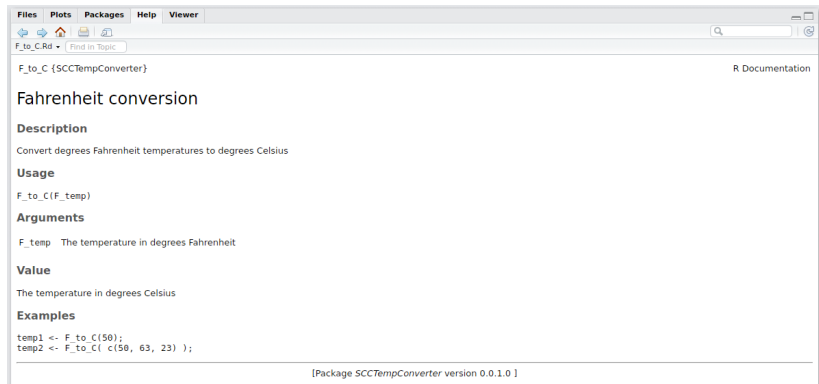
```
help(F_to_C); # ?F_to_C also works
```

Adding some minimal documentation

If we load our package and type the following:

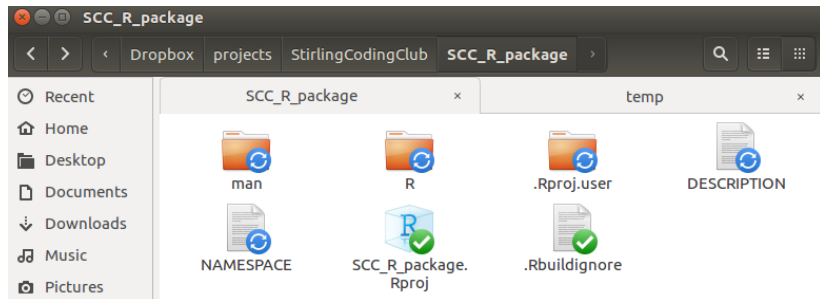
```
help(F_to_C); # ?F_to_C also works
```

We get the below in RStudio



Adding some minimal documentation

Note that a new folder has been added to the directory



Upload the package to GitHub

See notes on [version control](#) for help

StirlingCodingClub / SCC_R_package

Unwatch

7

Star

0

Fork

0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

An example R package

Edit

Manage topics

2 commits

1 branch

0 releases

1 contributor

Branch: master


New pull request

Create new file

Upload files








Find file

Clone or download

 bradduthie

Add the build ignore

Latest commit 05bda4 3 minutes ago

 R	Initialise everything	4 minutes ago
 man	Initialise everything	4 minutes ago
 notebook	Initialise everything	4 minutes ago
 .Rbuildignore	Add the build ignore	3 minutes ago
 DESCRIPTION	Initialise everything	4 minutes ago
 NAMESPACE	Initialise everything	4 minutes ago
 SCC_R_package.Rproj	Initialise everything	4 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README

Upload the package to GitHub

Anyone can download it by using the `install_github` function in `thedeveloptools`

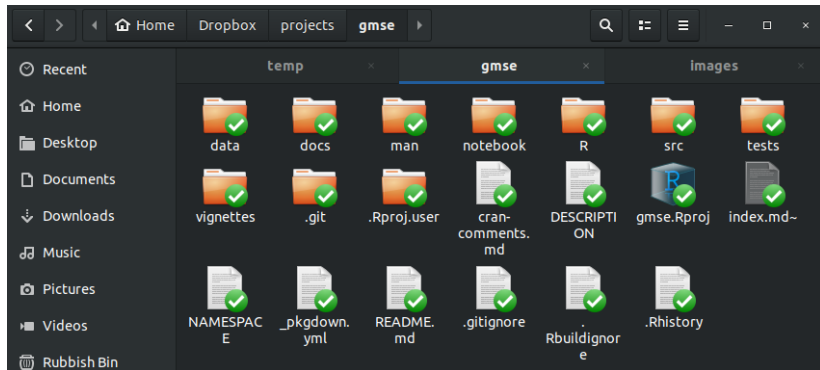
```
library(devtools);  
install_github("StirlingCodingClub/SCC_R_package");
```

Our R package is now installed. We can start using it by reading it in as a normal package.

```
library(SCCTempConverter);  
F_to_C(30);
```

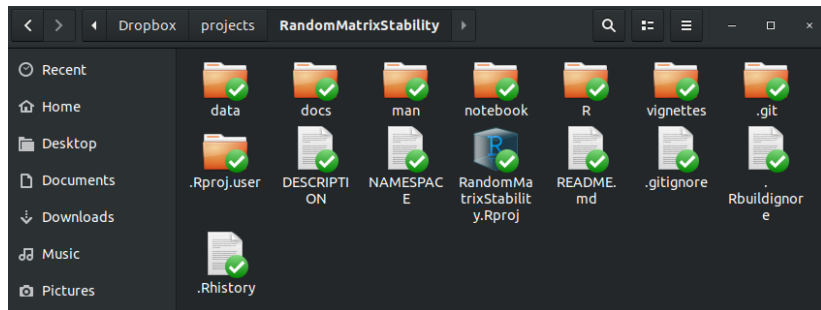
The R package design is very useful

A reasonably large R package directory



The R package design is very useful

A project for a manuscript in the R package style



Today's resources and additional help

Slides: https://stirlingcodingclub.github.io/SCC_R_package/notebook/Rpackage_slides.pdf

Notes: https://stirlingcodingclub.github.io/SCC_R_package/notebook/Rpackage_notes.html

From Karl Broman

- ▶ The minimal R package
- ▶ Building and installing an R package
- ▶ Writing documentation with Roxygen2

From RStudio

- ▶ R packages (free online book).
- ▶ pkgdown (automatically builds package website)