Lecture 14: Creating an R Package

Stevenson Bolivar

DATA WRANGLING AND HUSBANDRY

Introduction to R Packages

What is an R Package?

- A structured collection of R functions, data, and documentation.
- Organized with files like `DESCRIPTION`, `R/`, `man/`, `data/`.
- Allows easy sharing, installation, and reuse of code.

How to Create an R Package

- Set up folder structure manually or with `usethis::create_package()`.
- Write and organize functions under `R/`.
- Add metadata in the `DESCRIPTION` file.
- Document functions using `roxygen2` or `.Rd` files.
- Test, build, and install using tools like `devtools`.

Advantages

- Promotes code reuse and standardization.
- Improves collaboration and reproducibility.
- Enhances visibility of research and tools.
- Facilitates maintenance and scaling of codebases.

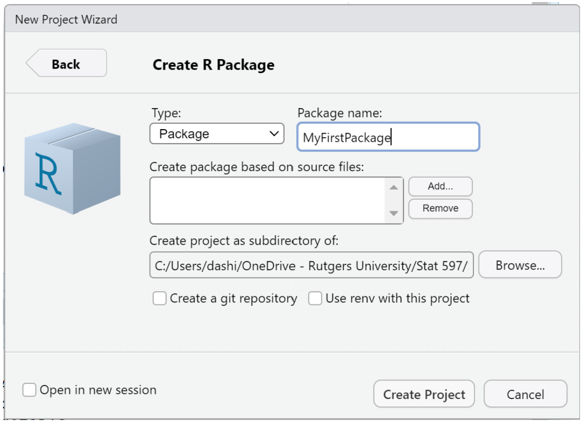
Disadvantages

- Requires time and effort to learn best practices.
- Maintenance over time is necessary.
- CRAN submission can be demanding.
- Poor maintenance can lead to user errors and reputational risks.

First Steps in Package Development

Setting Up the Package Structure

```r
# Load necessary packages
library(usethis)
library(devtools)
# Create a new package
create_package("myPackage")
# Create a new package with a specific path
create_package("path/to/myPackage")
# Create a new package with a specific path and
description
create_package("path/to/myPackage", description =
list(Title = "My Package",
    Description = "A package for demonstration
    purposes.", Version = "0.1.0",
    Author = "Your Name", License = "Rutgers
    University"))
```


New Project Wizard

Create R Package

Type: Package

Package name: MyFirstPackage

Create package based on source files:

Create project as subdirectory of: C:/Users/dashi/OneDrive - Rutgers University/Stat 597/

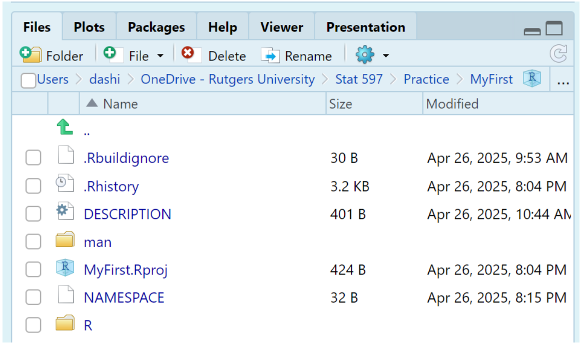
Create a git repository ☐ Use renv with this project ☐

Open in new session ☐

Figure 1: Create a new project

Setting Up the Package Structure

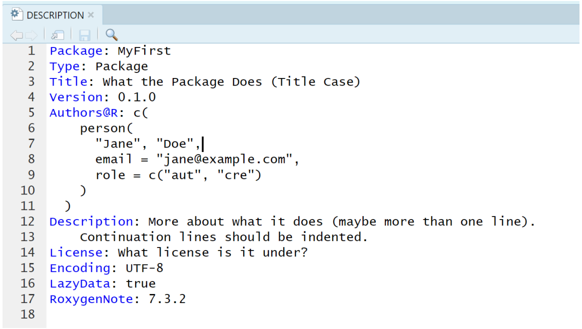
```r
# Load necessary packages
library(usethis)
library(devtools)
# Create a new package
create_package("myPackage")
# Create a new package with a specific path
create_package("path/to/myPackage")
# Create a new package with a specific path and
description
create_package("path/to/myPackage", description =
list(Title = "My Package",
    Description = "A package for demonstration
    purposes.", Version = "0.1.0",
    Author = "Your Name", License = "Rutgers
    University"))
```


Files
Plots
Packages
Help
Viewer
Presentation
Folder
File
Delete
Rename
Users > dashi > OneDrive - Rutgers University > Stat 597 > Practice > MyFirst

```
Name             Size     Modified
..
.Rbuildignore    30 B     Apr 26, 2025, 9:53 AM
.Rhistory        3.2 KB   Apr 26, 2025, 8:04 PM
DESCRIPTION      401 B    Apr 26, 2025, 10:44 AM
man
MyFirst.Rproj    424 B    Apr 26, 2025, 8:04 PM
NAMESPACE        32 B     Apr 26, 2025, 8:15 PM
R
```

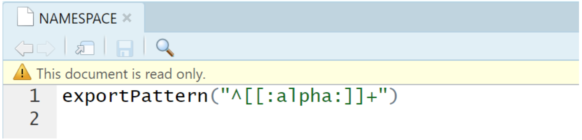
Figure 2: The project looks like this

Setting the files inside the package



```
Package: MyFirst
Type: Package
Title: What the Package Does (Title Case)
Version: 0.1.0
Authors@R: c(
    person(
      "Jane", "Doe",
      email = "jane@example.com",
      role = c("aut", "cre")
    )
  )
Description: More about what it does (maybe more than one line).
    Continuation lines should be indented.
License: What license is it under?
Encoding: UTF-8
LazyData: true
RoxygenNote: 7.3.2
```

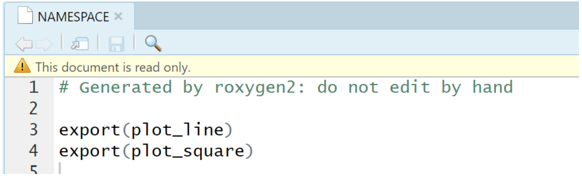
Figure 3: Description file



This document is read only.

```
exportPattern("^[[:alpha:]]+")
```

Figure 4: NAMESPACE file, exportPattern



This document is read only.

```
# Generated by roxygen2: do not edit by hand

export(plot_line)
export(plot_square)
```

Figure 5: NAMESPACE file, export

Create the functions inside the foldet R

```r
# Function to plot a line
plot_line <- function(m, b, x = c(1:100)) {
  y <- m * x + b
  plot(x, y, type = "l", col = "blue", xlab = "x", ylab = "y")
  title("Line Plot")
}

# Function to plot a square
plot_square <- function(a, c, x = c(1:100)) {
  y <- (a + x)^2 + c
  plot(x, y, type = "l", col = "red", xlab = "x", ylab = "y")
  title("Square Plot")
}
```

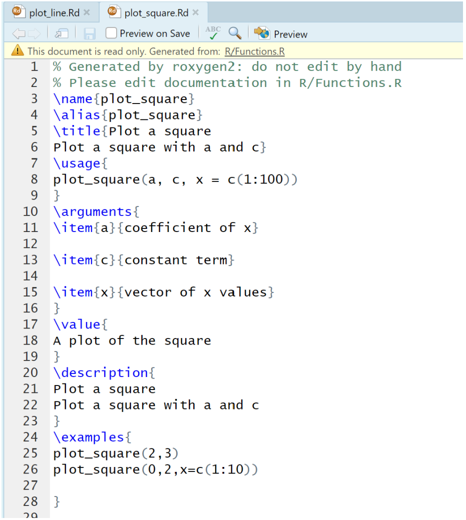
Create the documentation

```r
#'Plot a line
#'
#'@description
#' Plot a line with slope m and intercept b
#'@param m slope of the line
#'@param b intercept of the line
#'@param x vector of x values
#'
#'@examples
#' plot_line(2,3)
#' plot_line(0,2,x=c(1:10))
#'
#'@return A plot of the line
#'
#'@export
plot_line <- function(m, b, x = c(1:100)) {
  y <- m * x + b
  plot(x, y, type = "l", col = "blue", xlab = "x",
  ylab = "y")
  title("Line Plot")
}
```

```r
#'Plot a square
#'Plot a square with a and c
#'@param a coefficient of x
#'@param c constant term
#'@param x vector of x values
#'
#'@examples
#' plot_square(2,3)
#' plot_square(0,2,x=c(1:10))
#'
#' @return A plot of the square
#'
#'
#'@export
plot_square <- function(a, c, x = c(1:100)) {
  y <- (a + x)^2 + c
  plot(x, y, type = "l", col = "red", xlab = "x",
  ylab = "y")
  title("Square Plot")
}
```

Create the documentation in the Rd files

```r
# Load the devtools package, which helps to create the
# package
library(devtools)

# Load the roxygen2 package, which helps to create the
# documentation
library(roxygen2)

# Load the package
load_all()

# Create the documentation
roxygenise()
```



This document is read only. Generated from: R/Functions.R

```
% Generated by roxygen2: do not edit by hand
% Please edit documentation in R/Functions.R
\name{plot_square}
\alias{plot_square}
\title{Plot a square
Plot a square with a and c}
\usage{
plot_square(a, c, x = c(1:100))
}
\arguments{
\item{a}{coefficient of x}

\item{c}{constant term}

\item{x}{vector of x values}
}
\value{
A plot of the square
}
\description{
Plot a square
Plot a square with a and c
}
\examples{
plot_square(2,3)
plot_square(0,2,x=c(1:10))

}
```

Data inside the package

```r
# Create a data frame
my_data <- data.frame(x = c(1, 2, 3, 4, 5), y =
c(2, 4, 6, 8,
  10))

# or load it from a csv file
my_data <- read.csv("data/my_data.csv")

# Save the data frame as an R object, .rda file
usethis::use_data(my_data, overwrite = TRUE)
```

```r
#' @title Example Data: my_data
#' @name my_data
#' @docType data
#' @usage data(my_data)
#' @format A data frame with 10 rows and 2 columns:
#' \describe{
#'   \item{x}{A numeric vector of values 1 to 10.}
#'   \item{y}{A numeric vector of random values.}
#' }
#' @source Created with data.frame function.
#' @export
#' @examples
#' # Load the data
#' data(my_data)
#' # View the first few rows
#' head(my_data)
"my_data"
```

Create the package

Two main alternatives

Create and install a tar.gz file

```r
# Create the package
build()

# Install the package
install.packages("../MyFirstPackage_0.1.0.tar.gz",
repos = NULL,
  type = "source")

# Load the package
library(MyFirstPackage)

# Check the package documentation
help(package = "MyFirstPackage")
```

Using GitHub

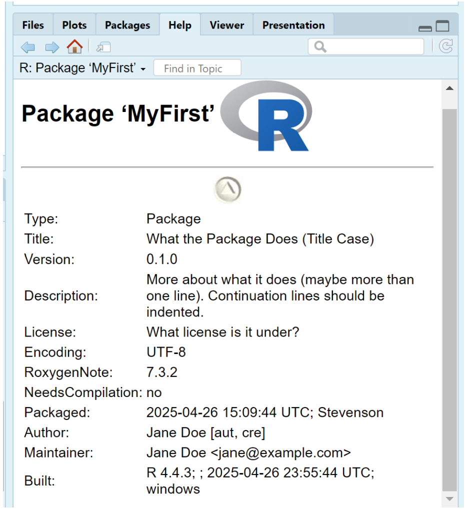
Projects can be created in GitHub and then downloaded to your computer

```r
# Load to your Github account your package
(project)
library(usethis)
use_git()
use_github()

# Install the package from your Github account
library("devtools")

#
install_github('github_user_name/github_repo_name')
install_github("StevensonBolivar/MyFirstPackage")

# Check the package documentation
help(package = "MyFirstPackage")
```

Use the package

Load the package

```r
# Load the package
library(MyFirstPackage)
# Check the package documentation
help(package = "MyFirstPackage")
```

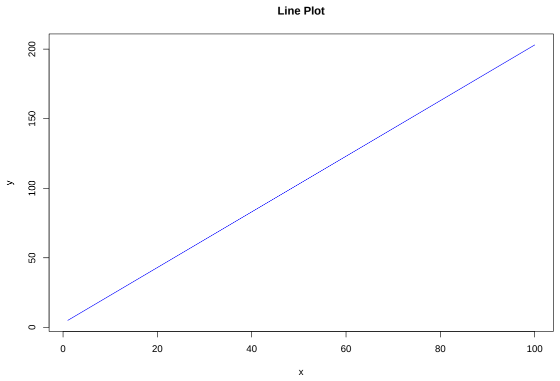


Package 'MyFirst'

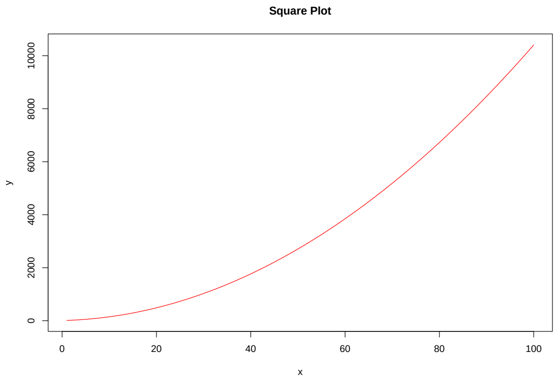
Type:	Package
Title:	What the Package Does (Title Case)
Version:	0.1.0
Description:	More about what it does (maybe more than one line). Continuation lines should be indented.
License:	What license is it under?
Encoding:	UTF-8
RoxygenNote:	7.3.2
NeedsCompilation:	no
Packaged:	2025-04-26 15:09:44 UTC; Stevenson
Author:	Jane Doe [aut, cre]
Maintainer:	Jane Doe <jane@example.com>
Built:	R 4.4.3; ; 2025-04-26 23:55:44 UTC; windows

Use the functions

```r
# Use the functions
plot_line(2, 3)
```



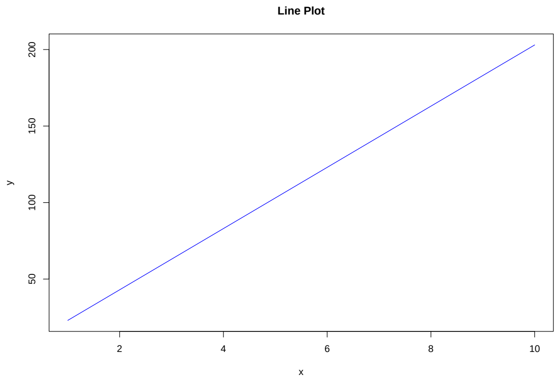
```r
# Use the functions
plot_square(2, 3)
```

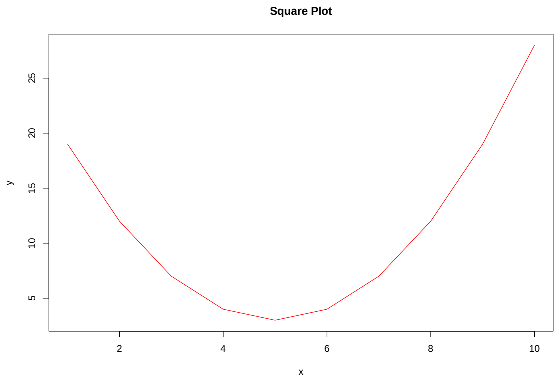


Use the functions

```r
# Use the functions
plot_line(20, 3, x = 1:10)
```

```r
# Use the functions
plot_square(-5, 3, x = 1:10)
```

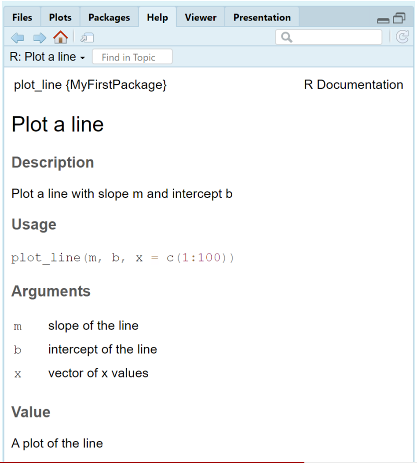




Help for the functions

```r
# Help for the functions
`?`(plot_line)
```

```r
# Help for the functions
`?`(plot_square)
```



plot_line {MyFirstPackage} R Documentation

Plot a line

Description

Plot a line with slope m and intercept b

Usage

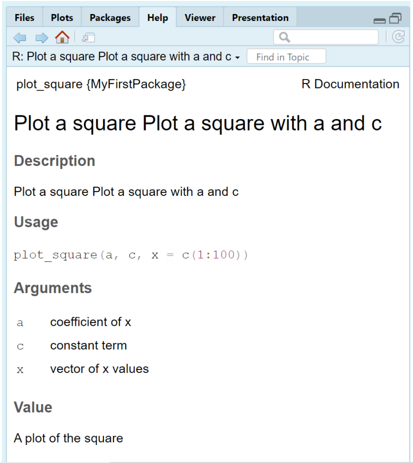
```
plot_line(m, b, x = c(1:100))
```

Arguments

m	slope of the line
b	intercept of the line
x	vector of x values

Value

A plot of the line



plot_square {MyFirstPackage} R Documentation

Plot a square Plot a square with a and c

Description

Plot a square Plot a square with a and c

Usage

```
plot_square(a, c, x = c(1:100))
```

Arguments

a	coefficient of x
c	constant term
x	vector of x values

Value

A plot of the square

References

References

- Support Posit. Developing packages with the RStudio IDE. Retrieved from https://support.posit.co/hc/en-us/articles/200486488-Developing-Packages-with-the-RStudio-IDE