Code as Data - R Packages and Git

Jared Edgerton

Code as Data - R Packages and Git

- Understanding R Packages
- Creating R Packages with devtools
- Documentation with roxygen2
- Version Control with Git
- Collaborating with GitHub

Understanding R Packages

- Organize and share code efficiently
- Ensure reproducibility
- Facilitate collaboration
- Contribute to the R community

Basic Structure of an R Package

```
my_package/
├── DESCRIPTION
├── NAMESPACE
├── R/
│   └── functions.R
├── man/
├── data/
└── tests/
```

Creating R Packages with devtools

```r
# Install devtools if not already installed
install.packages("devtools")
library(devtools)

# Create a new package
create_package("poliscitools")

# Navigate to the package directory
setwd("poliscitools")

# Add a function file
use_r("data_cleaning")
```

Writing Functions for Your Package

```r
# In R/data_cleaning.R

#' Clean political science data
#'
#' @param data A data frame containing political science data
#' @return A cleaned data frame
#' @export
clean_political_data ← function(data) {
  # Remove rows with any NA values
  data ← data[complete.cases(data), ]

  # Convert character columns to factors
  data ← data.frame(lapply(data, function(x) {
    if(is.character(x)) as.factor(x) else x
  }))

  return(data)
}
```

Documentation with roxygen2

- Use roxygen2 comments to document your functions
- Start each line with #'
- Use tags like @param, @return, @export, @examples

```
#' Analyze voter turnout
#'
#' @param data A data frame containing voter data
#' @return A summary of voter turnout
#' @export
#'
#' @examples
#' data ← data.frame(turnout = c(0, 1, 1, 0, 1))
#' analyze_turnout(data)
analyze_turnout ← function(data) {
  summary(data$turnout)
}
```

Adding Package Dependencies

```
# Add package dependencies
use_package("dplyr")
use_package("ggplot2")

# This will add the packages to the DESCRIPTION file
```

Including Data in Your Package

```r
# Create example data
example_data ← data.frame(
  voter_id = 1:1000,
  age = sample(18:90, 1000, replace = TRUE),
  party = sample(c("Democrat", "Republican", "Independent"), 1000, replace = TRUE),
  turnout = sample(c(0, 1), 1000, replace = TRUE, prob = c(0.4, 0.6))
)

# Add data to your package
use_data(example_data)
```

Creating a Vignette

```
# Create a vignette
use_vignette("introduction", "Introduction to poliscitools")

# This creates a template vignette file that you can edit
```

Adding Unit Tests

```r
# Set up testing infrastructure
use_testthat()

# Create a test file for data_cleaning.R
use_test("data_cleaning")

# In tests/testthat/test-data_cleaning.R
test_that("clean_political_data removes NA values", {
  dirty_data ← data.frame(a = c(1, 2, NA), b = c("x", "y", "z"))
  clean ← clean_political_data(dirty_data)
  expect_equal(nrow(clean), 2)
  expect_true(all(complete.cases(clean)))
})
```

Building and Checking Your Package

```
# Document your package
document()

# Build your package
build()

# Check your package
check()
```

Version Control with Git

Basic Git commands:

```
# Initialize a repository
git init

# Add files to staging
git add .

# Commit changes
git commit -m "Initial commit"

# Check status
git status

# View commit history
git log
```

Collaborating with GitHub

1. Create a GitHub repository
2. Clone the repository locally
3. Push changes to GitHub
4. Pull changes from GitHub
5. Create and merge pull requests

This Week's Assignment

1. Develop an R package for a specific political science analysis task
    - Include at least 3 functions
    - Write proper documentation using roxygen2
    - Include a relevant dataset
2. Use Git for version control
    - Create a GitHub repository for your package
    - Make regular commits as you develop
    - Use branches for different features

Resources

- R Packages (2e) by Hadley Wickham and Jenny Bryan
- Git and GitHub for Data Science
- CRAN - Writing R Extensions

Questions?

Thank you for your attention!