

# **Advanced Data Analysis with R**










Build Your Own R Package



# Advanced Data Analysis with R



# Build Your Own R Package

-  Introduction and Goals (15 min)
-  Setting Up the Package Structure (30 min)
-  Adding a Function and Documentation (45 min)
-  Break (15 min)
-  Testing Your Package with **testthat** (30 min)
-  Package Checking, Installation, and Usage (30 min)
-  (Optional) Advanced Topics and Extensions (15 min)
-  (Optional) Publishing Your Package on GitHub (15 min)
-  Wrap-up and Q&A (15 min)



# Introduction and Goals



# Why Build Your Own R Package?

```
1 final_script.R
2 final_script_v2.R
3 final_script_v2_final.R
4 landscape_calc_v3.R
```

- Where did this code come from?
- What does this function do again?
- Which R-packages and versions did I use?
- Why did it work 3 months ago but not now?

# Organize Your Code as a Package

```
1 install.packages("forestMetrics") # or from GitHub
2 library(forestMetrics)
3
4 load(file='landscape.rda')
5
6 landscape_class_area <- calc_area(landscape)
```

- ✓ Clear structure
- ✓ Reusable functions
- ✓ Auto-generated help pages
- ✓ Built-in documentation and tests

You don't need to publish on CRAN to benefit.

 **Create a Research Tool, not a  
Collections of Scripts**



# Setting Up the Package Structure



# Tools We Use

- **devtools**: build, install, test
- **usethis**: automate setup tasks
- **roxygen2**: documentation
- **testthat**: testing
- **RStudio**: integrated development

# Hands-On: Start Your Package

```
1 library(devtools)
2 library(usethis)
3
4 create_package("forestMetrics") # adjust if you want a different name
5 use_package("terra")             # add dependencies
6 use_git()                        # initialize git repository
```



# Adding a Function and Documentation

# Write a Function

```
1 # Create new R script for the function
2 use_r("calc_area")
3
4 # Add function
5 total_area <- function(landscape) {
6   stopifnot(inherits(landscape, "SpatRaster"))
7   # calculate patch areas (simplified)
8   area_ha <- terra::cellSize(landscape, mask = TRUE, unit = "ha")
9   total <- sum(terra::values(area_ha), na.rm = TRUE)
10
11   return(total)
12 }
```

# Document with roxygen2

```
1 #' Compute total area of a landscape
2 #'
3 #' @param landscape SpatRaster, the landscape raster (terra SpatRaster object)
4 #'
5 #' @return Total area in hectares (ha), omitting NA values.
6 #' @examples
7 #' library(terra)
8 #' landscape <- terra::rast(matrix(1, 10, 10)) # example raster
9 #' total_area(landscape)
10 #' @export
```

```
1 # Generate documentation
2 devtools::document()
```

**Break (15 min) 🚀 Grab a coffee,  
stand up and stretch!**

# Testing Your Package with testthat

# Software Testing: Why?

- Catch bugs early <- save time in the long run
- Improves reliability, confidence, and trust
- Testable code is inherently better code



# Software Testing: What?

- **Unit tests:** test individual functions
- **Integration tests:** test how functions work together

# Unit Tests

- Functions to test functions, usually named `test-functionname()`
- Have no parameters
- Fixed data and expected results
- returns a boolean value - `TRUE` or `FALSE`
- Use `testthat` to write unit tests

# Integration Tests

- Not our focus today because they are more complex
- `testthat` can also handle these by looping through multiple inputs

# Software Testing: How?

- Use the `testthat` package
- Have at least one test for each function in your `R/` directory
- When you find a bug, write a test that fails first, then fix the bug
- Use `usethis::use_test()` to write tests in `tests/testthat/`
- Run tests with `devtools::test()`

# Package Checking, Installation, and Usage

- Check your package: `devtools::check()`
- Install your package: `devtools::install()`
- Load your package: `library(forestMetrics)`




# (Optional) Advanced Topics and Extensions

# (Optional) Publishing Your Package on GitHub

```
1 library(usethis)  
2 use_github()
```

```
1 devtools::install_github("yourname/forestMetrics")
```

# Wrap-up and Q&A - What You've Built

-  A reusable R package
-  With functions, documentation, and tests
-  That you can share and build on

You're no longer just writing scripts —  
You're building research tools.