# *Outline for GSC Projects*

## Week 1: Familiarization and Initial Setup

**Day 1-3: Understanding the Requirements**

- Review the project instructions to clearly understand the required functionalities of the package.
- Take note of any necessary components such as functions, data inputs, output expectations, and documentation needs.

**Day 4-5: Set Up Your R Package Structure**

- Use usethis::create_package() to create a new package project in RStudio.
- Create folders for **R scripts**, **documentation** (man folder), **tests** (tests/testthat folder), and **data** (if needed).

**Day 6-7: Define Core Functions**

- List the main functions the package needs based on the problem, such as functions to:
  - Generate data.
  - Apply the GSC (generate, sort, correlate) algorithm.
  - Compute correlation bounds and implement trivariate correlation.
- Start writing pseudo-code or comments in each function to outline steps.

## Week 2: Developing Functions and Documentation

**Day 8-12: Develop and Test Core Functions**

- Begin coding each function based on your outline from Week 1.
- After coding, test each function independently to ensure it works as expected.
- Create simple tests in tests/testthat for key functionality and edge cases (use testthat::test_that()).

**Day 13-14: Add Documentation**

- For each function, add roxygen2 documentation above the function code (use #' to describe each argument and the function's purpose).
- Run devtools::document() to generate .Rd files in the man folder.

## Week 3: Package Testing, Improvement, and Finalization

**Day 15-17: Internal Testing and Refinement**

- Test the package as a whole by calling functions in a new R script or console session.
- Address any errors, warnings, or unexpected behavior. Verify that tests cover possible use cases.

**Day 18-19: Write a Vignette**

- Write a vignette to demonstrate usage. Use usethis::use_vignette("package_name") to create the vignette.
- Show example workflows, function calls, and expected outputs.

**Day 20-21: Final Check and Submission Preparation**

- Check the package with devtools::check() to identify issues.
- Package your project, ensuring everything is well-documented and functions as expected.