Programming Tools in Data Science
Exercise 3

31 October 2024
Correction: 7 November 2024

Lecture 10: Functional Programming

Suppose you work for a retail company, and you have a list of products with their daily sales recorded for each day of the month. The data is represented as follows:

```r
product_sales <- list(
  product1 = c(50, 45, 60, 55, 70, 80, 75, 90, 85, 60, 70, 65, 70, 75, 80,
               85, 90, 95, 85, 70, 75, 80, 60, 45, 55, 50, 45, 60, 65),
  product2 = c(30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100,
               105, 110, 115, 120, 125, 130, 135, 140, 145, 150, 155, 160,
               165, 170, 175),
  product3 = c(20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48,
               50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78)
)
```

1. Using a `for` loop, calculate the total monthly sales for each product.
2. Repeat 1 using `map`.
3. Repeat 1 using `lapply`.
4. Repeat 1 using `sapply`.
5. Repeat 1 using `vapply`.
6. Repeat 1 using `mclapply` or `parLapply`.
7. Compare these six approaches with `microbenchmark`. Which approach is the most efficient?

Lecture 11: R Package

1. Create a package from an RStudio new project with the following function:

   ```r
   `%r%` <- function(y, x) {
     fit <- lm(y ~ x)
     coef(fit)
   }
   ```

2. Modify the `DESCRIPTION` file: Add an author, a license, dependencies, and any other relevant metadata.
3. Document the function using `roxygen2`. Ensure that your `Build tools` options are set to use `roxygen2` for documentation.
4. Create and add a `snipes` dataset (from the class website) to the package. Keep the raw data, create an `.rda` file, and document the dataset accordingly.
5. Construct a vignette for the package to provide an in-depth explanation of its usage.

6. Add examples to demonstrate how to use the function in various scenarios.
7. Add tests using the `testthat` package to ensure the function behaves as expected.
8. Set up automated checks with GitHub Action to continuously test your package.
9. Create a website for your package using `pkgdown` and add a GitHub Action to build and deploy the website automatically.

Lecture 12: Advanced Shiny Applications

1. Develop a Shiny app with two tabs: the first to draw an histogram and the second to report summary statistics. Inputs are: number of cells, label for x-axis, title for the graph and an action button.
2. Using the 'mtcars' dataset, allow the user to select a variable.
3. Rewrite the app using Shiny modules.
4. Try the same exercise using ShinyUiEditor.
5. Structure the app into a package.