PHB 228: Statistical Computing

Homework Assignment 2: Data Structures and Mapping Functions

Assigned: Wednesday, April 9, 2025 Due: Wednesday, April 16, 2025, 11:59 PM Points: 100

Overview

This assignment will reinforce your understanding of R's data structures, mapping functions, and version control concepts covered in Lectures 2-4. You will practice working with different data structures, apply mapping functions, and implement memory-efficient code. The Palmer Penguins dataset will serve as our primary dataset for this assignment.

Technical Requirements

- R (version 4.3.0 or later)
- RStudio Desktop
- Git and GitHub account
- Required R packages: `palmerpenguins`, `purrr`, `dplyr`, `ggplot2`

Instructions

Part 1: Version Control Setup (15 points)

1. **(5 points)** Create a new Git repository for this assignment:
   - Initialize a local Git repository named `hw2_yourlastname`
   - Create a README.md file with a brief description of the assignment
   - Make an initial commit with the README.md file
2. **(5 points)** Create an R script named `hw2_solutions.R` with header comments including:
   - Your name
   - Course name and assignment number
   - Date
   - Brief description of the script's purpose
3. **(5 points)** Load the required packages and commit this change with an appropriate commit message.
   - Load `palmerpenguins`, `purrr`, `dplyr`, and `ggplot2`
   - If the `palmerpenguins` package is not installed, install it using `install.packages("palmerpenguins")`

Part 2: Data Structures (30 points)

1. **List Operations (10 points)**
   - **(3 points)** Extract the unique species from the `penguins` dataset

- **(3 points)** Create a list where each element contains data for one species
- **(2 points)** Add an attribute to each list element showing the sample size
- **(2 points)** Demonstrate how to access the sample size attribute

2. **Matrix vs. Data Frame (10 points)**
   - **(3 points)** Create a matrix containing only the numeric measurements from the `penguins` dataset (bill length, bill depth, flipper length, body mass)
   - **(3 points)** Perform the same operation as a data frame
   - **(2 points)** Compare the results and explain any differences
   - **(2 points)** Describe when you would prefer each data structure (2-3 sentences)
3. **Copy-on-Modify (10 points)**
   - **(3 points)** Create a vector `x` with values 1:5
   - **(3 points)** Create a reference `y` pointing to the same vector
   - **(2 points)** Modify `y` and explain what happens to `x`
   - **(2 points)** Explain how this behavior differs from languages like Python or Java (2-3 sentences)

Part 3: Map Functions (40 points)

4. **Base R Apply Functions (15 points)**
   - **(5 points)** Use `lapply()` to calculate the mean of each numeric variable in the penguins dataset (excluding NAs)
   - **(5 points)** Use `tapply()` to find the mean body mass by species
   - **(3 points)** Use `tapply()` again to find the mean body mass by both species and sex
   - **(2 points)** Compare the output types from each function (1-2 sentences)
5. **Purrr Map Functions (15 points)**
   - **(5 points)** Rewrite the first task from question 4 using `map_dbl()`
   - **(5 points)** Use `map2()` to calculate the ratio of bill length to bill depth for each species
   - **(3 points)** Create a list where each element contains a different statistic (mean, median, sd) for each measurement variable
   - **(2 points)** Explain which approach you prefer (base R vs purrr) and why (2-3 sentences)
6. **Practical Application (10 points)**
   - **(7 points)** Use the map pattern to create a separate histogram for each numeric variable in the penguins dataset. Each histogram should:
     - Have an appropriate title based on the variable name
     - Use faceting to show separate histograms by species
     - Use a color palette that distinguishes between species
   - **(3 points)** Write a brief explanation (2-3 sentences) of how this approach is more efficient than writing separate code for each plot

Part 4: Memory Management (15 points)

7. **Efficient Code (8 points)**
    - **(3 points)** The following code is inefficient. Rewrite it using pre-allocation:

      ```
      result <- numeric(0)
      for(i in 1:10000) {
        result <- c(result, i^2)
      }
      ```
    - **(3 points)** Compare the execution time of your version vs. the original using `system.time()`
    - **(2 points)** Explain why your version is more efficient in terms of R's memory model (2-3 sentences)
8. **Data Structure Selection (7 points)**
    - For each of the following scenarios, identify the most appropriate data structure and explain why:
        - **(2 points)** a. Storing patient IDs and their blood pressure readings
        - **(2 points)** b. Representing a correlation matrix between 5 variables
        - **(1 point)** c. Organizing multiple statistical models applied to different subsets of data
        - **(2 points)** d. Storing latitude and longitude coordinates for map plotting

Grading Rubric

Component	Points	Description
Version Control Setup	15	Repository creation, script setup, package loading
Data Structures	30	List operations, matrix vs. data frame comparison, copy-on-modify
Map Functions	40	Base R apply functions, purrr map functions, practical application
Memory Management	15	Efficient code implementation, data structure selection
Total	**100**	

Late Penalty: 10% deduction per day, up to 3 days. Assignments more than 3 days late will not be accepted.

Academic Integrity

Your submission must be your own work. You may discuss concepts with classmates, but all code and analysis must be completed individually. Please cite any resources you used, including AI tools, if applicable.