Programming I

Introduction

- Important Reading
  - Ch. 15-17 in R4DS
  - Ch. 10-11 in RPZP
  - Ch. 2.3,2.4,9,11 in HOPR

- Programming Steps
  - Understand the Problem
  - Inputs and Outputs
  - Create Code
  - Test the Code (Simple Case)
  - Generalize the Code
  - Test Problematic Cases
  - Edit Code to Handle Issues
  - Consider Efficiency

Setup for Lecture

- Open Supplement

- Packages Required:
  - Tidyverse
  - Ecdat

- Knit Document As You Go

- Read Introduction

- Prepare Your Minds for the Matrix

Part 1: If-Else

- General Construction:

  - "If"

    ```
    if (CONDITION) {
        ACTION
    }
    ```

  - "If-Else"

    ```
    if (CONDITION) {
        ACTION 1
    } else {
        ACTION 2
    }
    ```

  - ifelse()

    ```
    ifelse(CONDITION,ACTION1,ACTION2)
    ```

Part 1: If-Else

- Run Chunk 1
  - Check if Larger than 0
  - If True, Take Log
  - Result When x = 3?
  - Result When x = -3?

- Run Chunk 2
  - Notice the Difference
  - If-Else to Handle Errors

- Run Chunk 3
  - Situation Not Considered
  - Replace *BLANK* to Lead to Potential Problem

Part 1: If-Else

- Run Chunk 4
  - Replace BLANK with Different Options and Check
  - How Would You Explain this Code to Your Granny?

- Run Chunk 5
  - What is the Difference Between y1 and y2?
  - Always Look for a Vectorized Solution for Efficiency

- Run Chunk 6
  - Nested ifelse() Statements
  - How Would You Explain this to your Mother?

Part 2: Loops

- General Construction

  - "for" Loop

    ```
    for (INDEX in VECTOR) {
          ACTION FOR EACH INDEX
    }
    ```

  - "while" Loop

    ```
    while (CONDITION) {
          ACTION UNTIL CONDITION = FALSE
    }
    ```

  - Nested "for" Loops

    ```
    for (INDEX1 in VECTOR1) {
          for (INDEX2 in VECTOR2) {
                ACTION
          }
    }
    ```

Part 2: Loops

- Mental Process

  - I Want to Do _____ for Every _____ until _____

  - What Type of Object Do You Want Returned?

  - Initiate a Starting Point Based on the Desired Output

  - Try R Code on Single Instance

  - Create the Loop

Part 2: Loops

- Geometric Series

$$\sum_{k=0}^{\infty} ar^k = \frac{a}{1-r}, \text{ for } |r| < 1$$

- Run Chunk 1
  - What *a* did you choose?
  - What *r* did you choose?
  - What is the theoretical limit?
  - What pattern exists?

- Run Chunk 2
  - Choose *a* and *r* that work?
  - Choose *a* and *r* that don't work?
  - Modify: `if(k>100) break`

Part 2: Loops

- Geometric Series (Cont.)

$$\sum_{k=0}^{\infty} ar^k = \frac{a}{1-r}, \text{ for } |r| < 1$$

- Run Chunk 3
  - Suppose We Want to Save at Every Step
  - Why? Picture to Examine the Path of the Summation
  - Choose Small K<15
  - Choose Large K>50
  - What do You Observe?
  - How Would You Explain This Code to Your Stranged Brother?

Closing

Disperse and Make Reasonable Decisions