SPEC REU R Resources: Intro to R - Groupwork

Alix Ziff, Gaea Morales and Claudia Salas Gimenez

Summer 2024

Welcome to the final group assignment for Module 1: Introduction to R. Throughout this module, we've covered arithmetic in R, basic R operations, vectors, and tibbles. The goal of this assignment is to reinforce these foundational concepts before we move on to more advanced tasks in data management, regression analysis, and data visualization. You'll practice subsetting vectors and accessing specific elements within them, using larger vectors where we can't simply eyeball the answers.

By the end of this assignment, you will have practiced a range of operations using pre-packaged data in R, and you'll be well-prepared for the upcoming modules on data management and data visualization.

Initial Setup

To start, load the data needed to complete this assignment. We will be working with pre-packaged data in R using the `EuStockMarkets` dataset. Pre-packaged data refers to datasets that come built-in with R, ready for you to use without the need for external imports. `EuStockMarkets` contains daily closing stock prices for major European stock markets from 1991 to 1998. Specifically, you'll focus on the Swiss Market Index (SMI), which is the second column in this dataset.

To load the data on SMI, we will extract the second column from the `EuStockMarkets` dataset and convert it into a numeric vector, which will then be stored in an object called `stocks`.

```
# Extracting the Swiss Market Index (SMI) data from the EuStockMarkets dataset
stocks <- as.numeric(EuStockMarkets[,2])

## EuStockMarkets[,2] selects the second column from the EuStockMarkets dataset,
## which corresponds to the Swiss Market Index (SMI).

## as.numeric() converts the selected column into a numeric vector to make sure
## the data are stored as numeric data
```

The following exercises will guide your group through various operations with the `stocks` data, helping you collaboratively build confidence in manipulating and analyzing vectors in R.

Calculations, Logical Operators, and Vectors

Now that we have the data, work through the following exercises. Assign each task to different group members, but ensure that everyone participates in the discussion and solution of each exercise.

Exercise 1: Access the 90th Element

Get the 90th element of the `stocks` vector. Save it to an object named `nintey`.

```
# Retrieve 90th element from stocks vector
nintey <- stocks[90]
```

Exercise 2: Retrieve the Last Element

Get the last element of the vector `stocks`. Save it to an object named `last`.

Helpful hint: You will need to determine the size of the vector first and use that information to access the last element.

```
# Find the last element of the vector stocks
## Approach 1: Find the length of the vector and access the last element in
## one line of code
last <- stocks[length(stocks)]

## Approach 2: Find the length of the vector and store it in object 'n'. Then
## retrieve the nth element from stocks vector
n <- length(stocks)
last <- stocks[n]

## Approach 3: Look at the global environment under "Values" to see the length
## of stocks vector and then retrieve that number
last <- stocks[1860]
```

Exercise 3: Create a Subset by Removing Elements

Make a copy of the `stocks` vector, name it `copy`, and then delete the first five elements of `copy`.

```
# Create vector 'copy'
## Approach 1: Create a copy of stocks vector, excluding the first five element
copy <- stocks[-c(1,2,3,4,5)]

## Approach 2: Creates a copy of stocks vector, excluding the numbers in the
## range from the first to the fifth element
copy <- stocks[-(1:5)]

## Approach 3: Creates a copy of stocks vector and then remove the first five
## elements from the 'copy' vector
copy <- stocks
copy <- copy[-c(1,2,3,4,5)]
```

Exercise 4: Filter Entries Above and Below the Mean

Get all the entries from `stocks` that are above the mean value of `stocks`. Save this new vector as `above`. Then, get all the entries from `stocks` that are below the mean. Save this new vector as `below`.

```
# Filters the stocks vector for values above the mean
above <- stocks[stocks > mean(stocks)]

# Filters the stocks vector for values below the mean
below <- stocks[stocks < mean(stocks)]

## Approach 2: Calculate the mean of stocks and then create the 'above' and
## 'below' vectors
mu <- mean(stocks)

above <- stocks[stocks > mu]
below <- stocks[stocks < mu]
```

Exercise 5: Count Days with Closing Prices Above 6,000

On how many days were the closing prices greater than 6,000?

Helpful Hint: Calculate how many elements of the vector `stocks` are larger than 6,000.

```
# Calculate how many elements of the vector stocks are larger than 6,000
## Approach 1: Calculate the length of the elements in stocks vector that are
## greater than 6000
length(stocks[stocks > 6000])
```

```
## [1] 180
```

```
## Approach 2: Create vector 'large' with all the elements from stocks that are
## greater than 6000, and then calculate the number of elements in large vector
large <- stocks[stocks > 6000]
length(large)
```

```
## [1] 180
```

```
## Approach 3: Count the number of elements in stocks vector that are greater
## than 6000
sum(stocks > 6000)
```

```
## [1] 180
```

Bonus Question

Save a time vector (from 1991 to 1998) for the `stocks` data and then create and name a tibble consisting of the vectors `year` and `stocks`.

Helpful Hint: Use the `time()` command on the original `EuStockMarkets` dataset to extract the year variables.

```
# Extract time data from the EuStockMarkets dataset and store it in object 'year'
year <- time(EuStockMarkets)

# Create tibble combining 'year' and 'stocks' vectors
stock_year <- tibble::tibble(year, stocks)
```

This groupwork assignment is intended to strengthen your foundational skills in R, so use this opportunity to collaborate with each other and discuss any challenges you faced. In the next assignment, we'll move on to the basics of data management and start working with datasets from published research papers. This will give you an early glimpse into the work of a social scientists.