

SPEC REU R Resources: Intro to R - Group Work

Alix Ziff and Gaea Morales

Summer 2022

1 Getting Started

The purpose of these exercises is to get you more comfortable working with arithmetic operations and vectors, particularly subsetting vectors and accessing elements of data in vectors. This time, we're going to work with larger vectors - ones where we can't just eyeball the answers. The following materials were adapted from early materials by Miriam Barnum and Therese Anders.

We will be working with pre-packaged data in R, using the `EuStockMarkets` function. Run the code below, which will save a vector of Swiss Market Index (daily) closing stock prices from 1991 to 1998 to the object `stocks`.

```
stocks <- as.numeric(EuStockMarkets[,2]) # this is telling `R` that we want
# only the second column in the EuStockMarkets tibble, and we want to save it
# as a numeric vector we also wrap the line in the as.numeric() function
# to make sure the data are stored as numeric data
```

2 Calculations, Logical Operators, and Vectors

Exercise 1 Get the 90th element of the vector `stocks`. Save it to an object named `nintey`.

```
nintey <- stocks[90]
```

Exercise 2 Get the last element of the vector `stocks`. Save it to an object named `last`. *Helpful hint:* You will need to figure out the size of the vector.

```
last <- stocks[length(stocks)]

# Or
n <- length(stocks)
last <- stocks[n]

# Or, you can look at the global environment under "Values" and
# see that the stocks vector has 1860 elements
last <- stocks[1860]
```

Exercise 3 Make a copy of the vector `stocks`, and name it `copy`. Then delete the first five elements of `copy`.

```
copy <- stocks[-c(1,2,3,4,5)]

# Or
copy <- stocks[-(1:5)]

# Or
copy <- stocks
copy <- copy[-c(1,2,3,4,5)]
```

Exercise 4 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`.

```
above <- stocks[stocks > mean(stocks)]
below <- stocks[stocks < mean(stocks)]

# Or
mu <- mean(stocks)
above <- stocks[stocks > mu]
below <- stocks[stocks < mu]
```

Exercise 5 On how many days were the closing prices greater than 6,000? (How many elements of the vector `stocks` are larger than 6000?)

```
length(stocks[stocks > 6000])
```

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

```
# Or
large <- stocks[stocks > 6000]
length(large)
```

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

```
# Or
sum(stocks > 6000)
```

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

BONUS CHALLENGE: Save a time vector (1990 to 1998) for the stocks data and then create and name a tibble consisting of the vectors `year` and `stocks`. *Helpful hint:* You can use the `time()` command on the original `EuStockMarkets` tibble to extract year variables.

```
year <- time(EuStockMarkets)
stock_year <- tibble::tibble(year, stocks)
```