Vectors in R: Takeaways

Syntax

GENERATING A VECTOR

- Use a colon (:) to generate a range of value:

```
vector <- 1:4
```

- Use the function seq() to generate a sequence of values following a rule

```
vector <- seq(from = 2, to = 10, by = 3)
```

- Use the function rep() to generate repeated values

```
vector <- rep(4, times = 10)
```

CREATING A VECTOR

- Use the c() function:

```
vector <- c(14, 24, 34)
```

CREATING A NAMED VECTOR

- Use the c() function:

```
vector <- c("name_1" = 14, "name_2" = 24, "name_3" = 34)
```

- Assign name attributes to a vector:

```
names(vector) <- name_vector
```

INDEXING VECTORS BY POSITION

- Extract a single element:

```
vector[1]
```

- Extract a range of elements:

```
vector[3:7]
```

- Extract multiple elements:

```
vector[c(2,5,7)]
```

INDEXING VECTORS BY LOGICALS

- Index into a numeric vector using a logical vector:

```
numeric_vector[logical_vector]
```

INDEXING VECTORS BY NAME

- Extract a single element:

```
vector["name_2"]
```

- Extract multiple elements:

```
vector[c("name_1", "name_2")]
```

APPENDING ELEMENTS TO A VECTOR

- Append a single element to a vector:

```
vector_1  <- c(5, 10, 15)

vector_2  <- c(vector_1, 20)
```

- Append a vector to another vector:

```
vector_1  <- c(5, 10, 15)

extra_values  <- c(20, 25)

vector_2  <- c(vector_1, extra_values)
```

REMOVING ELEMENTS FROM A VECTOR

- Remove a single element:

```
vector[-1]
```

- Remove multiple elements:

```
vector[c(-2, -5, -7)]
```

PERFORMING ARITHMETIC ON VECTORS

- Add, divide, or multiply vectors:

```
vector_1 + vector_2

vector_1 / vector_2

vector_1 * vector_2

vector_1 + vector_2 * vector_3
```

Concepts

- In this course we will learn the following data structures:
    - **Vector: one-dimensional structure for storing values of SAME TYPE.**
    - Matrix: two-dimensional structure for storing values of SAME TYPE.
    - Lists: multi-dimensional stucture for storing values of ANY DATA TYPE/OBJECT.
    - Dataframe: multi-dimensional structure for storing values of ANY DATA TYPE/OBJECT like datasets.

- R is a **1-indexed** programming language, which means that the first element in a vector is assigned a position of one.
- When performing operations on vectors of unequal length, R "recycles" values of the shorter vector until the two vectors are the same length.

Resources

- Documentation on indexing vectors in R
- Documentation on R's "recycling rule"