Concepts in Computing with Data

Stat 133, Fall 2024

Lecture 1 08/30/2024

1. Data Paradigms

i. How do we think of data?

ii. How do computers treat data?

iii. How do data sets get stored/formatted?

iv. How do programs/languages handle data?

2. Vectors

2.1. Data Types in R:

i. Logical (boolean)

ii. Integer

iii. Double (float)

iv. Char (String)

v. Complex, raw (*we don't deal with this in R too much*)

2.2. Consider the sample data set below:

name	**height (cm)**	**force**
Leia	150	True
Luke	170	True
Han	180	False

name = c(”Leia”, ”Luke”, ”Han”) where c(., ., .) is the combine function
height = c(150, 170, 180) (note that R treats this vector of values as doubles implicitly)
force = c(TRUE, TRUE, FALSE)

Remark: Alternatively, you could write `force = c(1, 1, 0)` but these are NOT logical values by default. There is a way to do this in R, however. We'll talk about this later! Since in our problem we have integers, the way to tell R explicitly that we're dealing with integer values, is as follows:

`ht_int = c(150L, 170L, 180L)` where the `L` specifies these are integer values.

3. Vector Properties

3.1. Introduction to Vectors and Indexing

Vectors are fundamental data structures in R that contain elements of the same type. Here, we'll consider two types of vectors: a vector of strings and vector of doubles. Note that in R, vectors are indexed starting at 1, meaning the first element of a vector is accessed using index 1, the second element using index 2, and so on.

Example 1: name vector

Value	Leia	Luke	Han
Index	1	2	3

Example 2: height vector

Value	150	170	180
Index	1	2	3

The tables above show two vectors—a name vector and a height vector—along with their corresponding indices. In R, each element of a vector is accessed using its index. For example, the first element of the name vector, Leia, is accessed using `name[1]`, and the first element of the height vector, 150, is accessed using `height[1]`.

3.2. Functions for Vectors

Several functions are available in R to manipulate and retrieve information about vectors.

- **Finding the Length of a Vector:** Use the `length()` function to find the number of elements in a vector.
  - **Example 3:** To find the length of the name vector: `length(name)`
- **Determining Data Type:** The `typeof()` function returns the data type of elements within a vector.
  - **Example 4:** To determine the data type of the name vector: `typeof(name)`

3.3. Naming Elements in Vectors

We will discuss two common methods to assign names to elements in vectors:

- **Assigning Names to an Existing Vector:** This method is preferred when you already have a vector and want to add names to its elements. It scales well to large vectors.
  - **Example 5:** Consider the height vector:

    ```
    height = c(150, 170, 180)
    names(height) = name
    ```

    This approach is advantageous because it allows you to dynamically assign names to elements without redefining the vector.
- **Creating a Named Vector:** This method is useful for small vectors when you want to define the values and their corresponding names simultaneously.
  - **Example 6:** Define a named height vector:

    ```
    height = c("Leia" = 150, "Luke" = 170, "Han" = 180)
    ```

    This approach is simpler and more readable for small vectors but can become cumbersome with larger datasets.

Both methods have their advantages:

- The first method is more flexible and can handle large vectors effectively.
- The second method is straightforward and useful for small datasets.

4. Special Values

In R, special values are used to represent: missing data, undefined operations, and infinities.

- **Null:** Empty or undefined value.
- **NA:** Missing data.
- **NaN (not a number):**
  - **Remark:** This appears when an operation results in an undefined numerical result e.g., `0/0`, `sqrt(-6)`, or `log(-10)`.
- **Inf, -Inf:** Represents positive and negative infinity in R.
  - **Remark:** Occurs when dividing by zero, e.g., `k/0` where `k > 0` results in `Inf`, and `k/0` where `k < 0` results in `-Inf`.

Warning:

Avoid using the combine function `c()` for a single value e.g., `stat = c(133)`. While this is valid R code, it is redundant since `stat = 133` achieves the same result.

Using `c()` in this manner might lead to unexpected results or confusion in your code, especially when the intent is not to create a vector but to assign a single value. Reserve the use of `c()` for combining multiple values into a vector.

5. Numeric Sequences

Numeric sequences in R can be created using the colon `:` operator:

- **Example 7:** `1:5` is the sequence `1, 2, ..., 5`.
- **Example 8:** `-5:-1` is the sequence `-5, -4, ..., -1`.
- **In general:** `seq(from = x, to = y, by = h)` where `x` is the starting point, `y` is the end point, and `h` is the step size.