Lists in R

Sean Hellingman ©

Data Visualization and Manipulation through Scripting (ADSC1010)

shellingman@tru.ca

Fall 2024

THOMPSON RIVERS UNIVERSITY

Topics

Introduction

- A **list** is a one-dimensional collection of data.
- Unlike vectors, lists can store mixtures of data types.
- We can even store vectors and arrays within lists.
- Useful for storing irregular or non-rectangular data.

List Creation in R

- Elements in lists may be *tagged* for easy retrieval.
- Therefore, may be useful for organising data.
- Tags are not required, but are recommended.

`list()` Function

- The `list()` function may be used to create a list.
- Include labels within the function.
- May also store lists within lists.

Example 1

- Create a list with the following information:

- Name: Haaland
- Position: Striker
- Club: City
- Number: 9
- Clubs: City, Dortmund, Salzburg, Molde, and Bryne

- The sublist called 22/23_Stats:
  - Goals: 36
  - Assists: 8

List Elements

- Use `names()` to check the data structures within a list.
- When tagged, use `$` to access list elements by name.
- Also use `[[]]` to access list elements by their numerical index.
- Or, use `[["variable_name"]]` to access list elements by name.

Example 2

- Use the following to examine the list elements from Example 1:
  - `names()`
  - `$`
  - `[[]]`
  - `[["variable_name"]]`

Complex List Elements

- Lists may contain (nested) vectors and/or lists.
- We may use two $ or [] to extract specific elements.

Example 3

- Using the list from Example 1:
  - Extract Haaland's second professional club.
  - How many assists Haaland got in the 22/23 season.

List Modification

- Can assign new elements using the $.
- Can modify existing values using $.
- Numeric operators work within lists.

Example 4

- Using the list from Example 1:
  - Change Haaland's name to *Erling Haaland*
  - Add Haaland's Champions League goals (12) and assists (1) to his totals.

More Information

- Single brackets `[]` return a list (filtered sublist).
- Double brackets return `[[]]` vector.
- Can use a vector of column names to obtain a filtered sublist.
  - `Player[c("Name","Club","Number")]`

`lapply()` Function

- The *list apply* `lapply()` function applies a function to each element of the list.
- Apply the `toupper()` function to each element:
  - `Player_Upper <- lapply(Player, toupper)`
- `toupper()` will convert everything to a *character*.

Exercise 1

- Create a list about yourself with the following elements:
  - Name
  - University
  - Places lived (as a vector)
  - The sublist called favourites:
    - Favourite food
    - Favourite colour
    - Favourite animal

Exercise 2

- Using the list from Exercise 1:
  - Change your favourite colour to Orange.
  - Capitalize all of the elements in your list.
  - Add your age to the list.

References & Resources

1. Douglas, A., Roos, D., Mancini, F., Couto, A., & Lusseau, D. (2023). *An introduction to R*. Retrieved from https://intro2r.com/

2. Freeman, M., Ross, J. (2018). *Programming Skills for Data Science.* Addison-Wesley Professional. ISBN-13: 978-0-13-513310-1