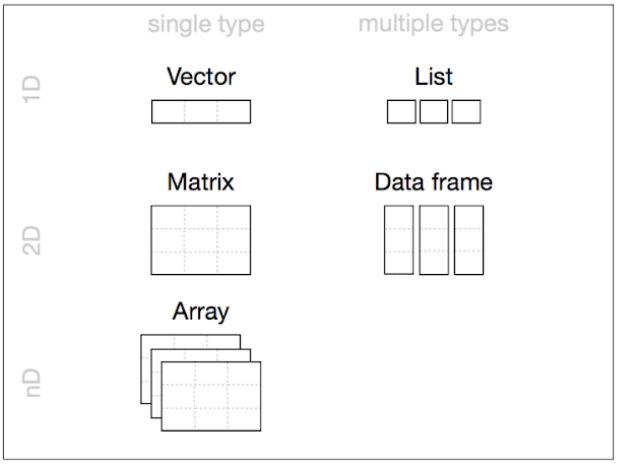
4 Feb 2023, Saturday

R Programming Lab
Assignment 4 with Concepts
Dataframe

Dimension	Homogenous	Heterogenous
1	Atomic vectors	Lists
2	Matrix	Data frame
≥ 1	Array	

- Data frames are rectangular 2-dimensional heterogeneous objects. Share some properties with matrices (the form) and lists (heterogeneity).



- Data frames look like matrices but are based on lists. This allows for a higher degree of flexibility than matrices as it now allows storing different types of data into different columns/variables, e.g., characters in the first column, integers in the second, and logical in the third.
- Data frames are constructed using the function data.frame()

1. Try the usage of **?data.frame** or **help("data.frame")**

2. Create a data frame with two vectors, x, and y. Vector x is an integer sequence of length 3, and vector y is a character vector of length 3 (‘a’,’b’,’c’).

3. Create a data frame with names by using a series of key = value arguments.
   df <- data.frame(age = c(35, 21, 12), height = c(1.72, 1.65, 1.39))

4. Try to create the following data frames:
    a. df1

##	name	manufacturer	release
## 1	Nintendo Switch	Nintendo	2017
## 2	Atari 2600	Atari	1977
## 3	Xbox	Microsoft	2001

    b. df2 (use vectors of different lengths (3 and 1 resp.))

##	month	season
## 1	Dec	Winter
## 2	Jan	Winter
## 3	Feb	Winter

5. Given the following two character vectors:
   state = "Vienna", "Lower Austria", "Upper Austria", "Styria", "Tyrol", "Carinthia", "Salzburg", "Vorarlberg", "Burgenland"

   country = "Austria"

    a. create data frame states using state and country vectors.
    b. create a list() with the same two vectors. What's the difference?
    c. create a data frame states first, and then coerce it to a list? Use as.list(states)

Subsetting Data Frames:

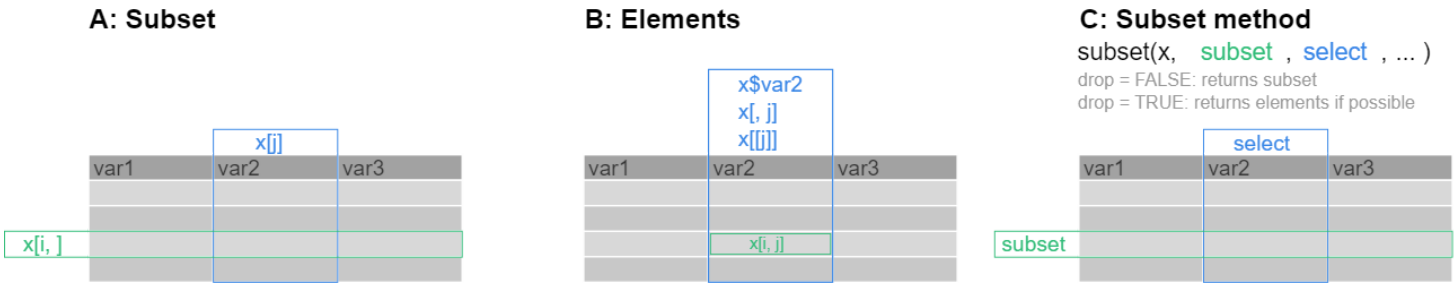
List-alike subsetting: [...], [[...]] and $

- [...] returns a reduced object of the same class.
- [[...]] and $ returns the content of the variable.

Matrix-alike subsetting: Subsetting by index, character, and logical vectors.

Subset() function:

- Argument subset: subsetting observations (rows).
- Argument select: subsetting variables (columns).



6. Given this data frame,
   dino <- data.frame(name = c("Tyrannosaurus", "Velociraptor", "Stegosaurus", "Ultrasauros", "MAN Lion's Coach"),
   height_m = c(7.0, 0.6, 3.4, 16.2, 3.87),
   length_m = c(15.2, 1.8, 9.1, 30.5, 12.101),
   weight_kg = c(6350, 113, 2722, 63500, 19700))

   Try to answer the following questions.

   a. What does dino["length_m"] return?
   b. What does dino[["length_m"]] return?
   c. What does dino$length_m return?

      Also, find the class, typeof, and length.

   d. Using indices: extract the first row of the data frame.
   e. Using characters: extract the column "name".
   f. Using indices: extract the length of the Tyrannosaurus and the Ultrasauros.
   g. Create a logical vector (use some logical expression on dino$height_m) which contains TRUE if the dinosaur was longer than 10 meters and FALSE else.
      With this logical vector,
      (i) extract all rows which belong to these large dinosaurs, and
      (ii) extract the names of these large (long) dinosaurs.

7. Write a loop that loops over all rows of the data frame dino and prints something like "The Tyrannosaurus was typically 7 meters tall" (requires one for loop and some matrix-alike subsetting).

<u>Loop replacement functions</u>: sapply() and lapply()

- lapply() always returns a list: applies a function on each element of an object.
  lapply(X, FUN)
- sapply() tries to simplify the return (vector/matrix): works the very same as lapply()
  sapply(X, FUN)

8. Try to call lapply() and sapply() on the object dino and return the maximum of each variable (max()).

9. Create 2 data frames:
   dataframe1 <- data.frame (Name = c("Juan", "Alcaraz"),Age = c(22, 15))
   dataframe2 <- data.frame (Name = c("Yiruma", "Bach"),Age = c(46, 89))
   dataframe3 <- data.frame (Hobby = c("Tennis", "Piano"))

a. combine data frames 1 and 2 vertically using rbind()
b. combine data frames 1 and 3 horizontally using cbind()

Note: The number of items on each vector of two or more combining data frames must be equal; otherwise, we will get an error.