Icahn
School of
Medicine at
Mount
Sinai

Introduction to coding with R

Part III

Joselyn C. Chávez Fuentes

01/28/2025

Let's recap

- How do we access vector elements?
- How do we modify vector contents?
- How do we create a matrix?
- What operations can we make with a matrix?
- How do we select/extract data from a matrix?

Data structures in R

- Vectors
- Matrices
- Lists
- Data frames
- Functions

Lists

Lists

- A list is an ordered collection of objects, known as components.
- Components can be of different size and class.
- Lists can contain vectors, matrices, strings, data frames, functions, etc.

How do we create a list?

```
fruits <- list(name = "apple",
               quantity = 5,
               properties = c("small", "red", "rounded"))

fruits
```

```
## $name
## [1] "apple"
##
## $quantity
## [1] 5
##
## $properties
## [1] "small"   "red"     "rounded"
```

How does the list look like using View( )?

How do we access list elements?

Using index

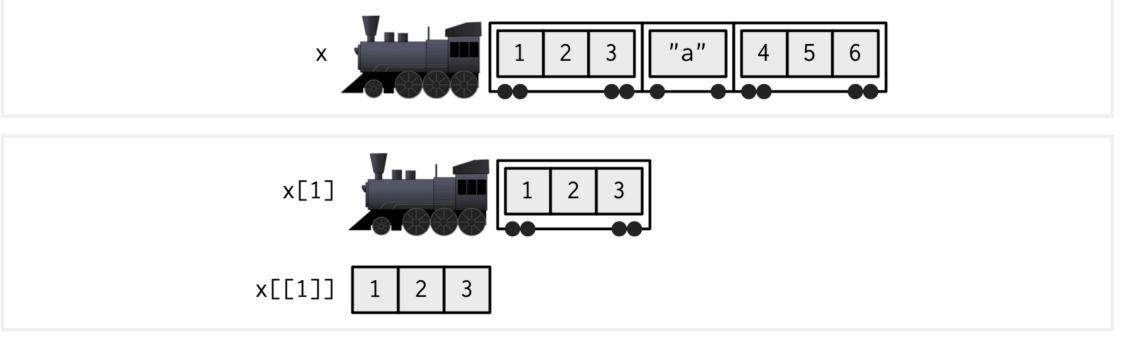

Image from Chapter 4 Subsetting. Advanced R. Second ed. by Hadley Wickham

How do we access list elements?

Using index

```
fruits[1]
```

```
## $name
## [1] "apple"
```

```
fruits[[1]]
```

```
## [1] "apple"
```

```
class(fruits[1])
```

```
## [1] "list"
```

```
class(fruits[[1]])
```

```
## [1] "character"
```

Your turn!

- How would you extract the number of apples?
- How would you extract the word "red" from the properties of the apple?

How do we access list elements?

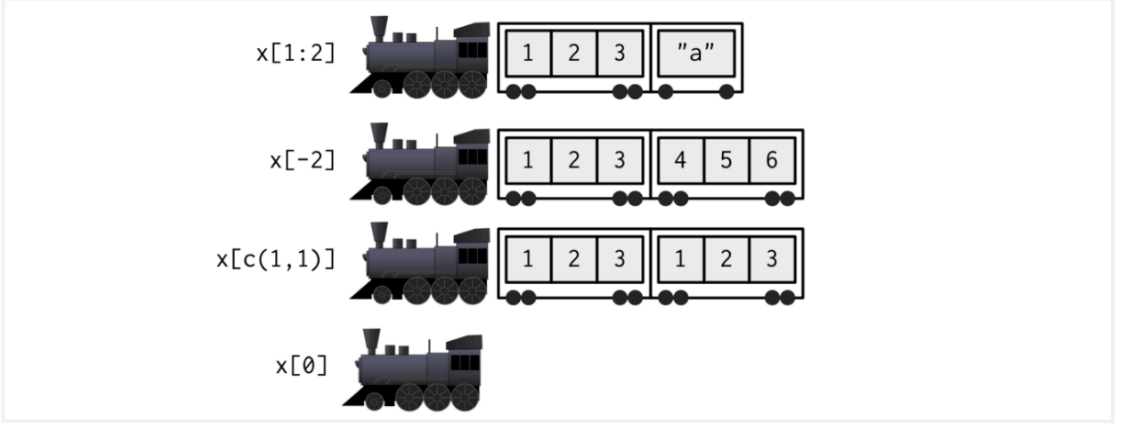

Image from [Chapter 4 Subsetting. Advanced R. Second ed.](#) by Hadley Wickham

- Extract the name and number of apples

```
fruits[1:2]
```

```
## $name
## [1] "apple"
##
## $quantity
## [1] 5
```

- Extract all elements except the number of apples

```
fruits[-2]
```

```
## $name
## [1] "apple"
##
## $properties
## [1] "small"   "red"     "rounded"
```

How do we access list elements?

Using the name

```
fruits["name"]
```

```
## $name
## [1] "apple"
```

```
fruits["properties"]
```

```
## $properties
## [1] "small"   "red"     "rounded"
```

Access elements using $

```
fruits
```

```
## $name
## [1] "apple"
##
## $quantity
## [1] 5
##
## $properties
## [1] "small"   "red"     "rounded"
```

```
fruits$name
```

```
## [1] "apple"
```

Only one element can be extracted with $

Your turn!

- Use $ to extract the number of apples
- How would you extract the word "rounded" from the apple properties using $?

How do we add a new element to the list?

```
fruits["edible_shell"] <- TRUE
fruits
```

```
## $name
## [1] "apple"
##
## $quantity
## [1] 5
##
## $properties
## [1] "small"   "red"     "rounded"
##
## $edible_shell
## [1] TRUE
```

How do we add a new element to the list?

```
fruits$hard_shell <- FALSE
fruits
```

```
## $name
## [1] "apple"
##
## $quantity
## [1] 5
##
## $properties
## [1] "small"   "red"     "rounded"
##
## $edible_shell
## [1] TRUE
##
## $hard_shell
## [1] FALSE
```

Your turn!

Create a list with a patient information

```
patient_1 <- list(name = "John",
                  last_name = "Doe",
                  weight = 170,
                  glucose = c(85, 90, 87, 89, 91))
```

- Extract the patient's name and last name
- Extract the glucose levels of days 3 to 5
- Remove the weight
- Add a new element to the list containing blood pressure levels

Data Frames

What is a data frame?

- Two-dimensional arranged data (tables)
- rows and columns
- All columns must be the same length
- Data frames can have different type of data
- All elements in a column must be the same type (vector)

Creating a data frame

```
fruits <- data.frame(

  name = c("apples","berries","mangos","bananas"),
  number = c(1, 10, 7, 2),
  edible_shell = c(TRUE,TRUE,FALSE,FALSE) )

fruits
```

```
##      name number edible_shell
## 1  apples      1         TRUE
## 2 berries     10         TRUE
## 3  mangos      7        FALSE
## 4 bananas      2        FALSE
```

Properties of data frames

- nrow

```
nrow(fruits)
```

```
## [1] 4
```

- ncol

```
ncol(fruits)
```

```
## [1] 3
```

- dim

```
dim(fruits)
```

```
## [1] 4 3
```

- rownames

```
rownames(fruits)
```

```
## [1] "1" "2" "3" "4"
```

- colnames

```
colnames(fruits)
```

```
## [1] "name"         "number"       "edible_shell"
```

Let's practice

Create a data frame (patients) that contains the following information:

```
##   first_name last_name age co_morbidity
## 1        Ava     Smith  65         TRUE
## 2    Richard     Brown  20        FALSE
## 3     Olivia  Williams  47        FALSE
```

- How many rows and columns does the data frame have?
- Print the columns and rows names

How do we access data frame elements?

Using row and column index

Syntaxis: df[row, column]

```
fruits
```

```
##      name number edible_shell
## 1  apples      1         TRUE
## 2 berries     10         TRUE
## 3  mangos      7        FALSE
## 4 bananas      2        FALSE
```

```
fruits[2,3]
```

```
## [1] TRUE
```

Select rows 1 to 2 from column 3

```
fruits[1:2,3]
```

```
## [1] TRUE TRUE
```

Your turn!

Using the patients data frame

```
##   first_name last_name age co_morbidity
## 1        Ava     Smith  65         TRUE
## 2    Richard     Brown  20        FALSE
## 3     Olivia  Williams  47        FALSE
```

- Extract the last name and age from Ava and Richard

- Select all rows from column 2

```
fruits[,2]
```

```
## [1]  1 10  7  2
```

- Select all columns from row 2

```
fruits[2,]
```

```
##      name number edible_shell
## 2 berries     10         TRUE
```

Your turn!

Using the patients data frame

```
##   first_name last_name age co_morbidity
## 1        Ava     Smith  65         TRUE
## 2    Richard     Brown  20        FALSE
## 3     Olivia  Williams  47        FALSE
```

- Extract all the information (columns) from Olivia.
- Extract the age from all patients

Using the $ symbol

Syntaxis: df$column_name

- Extract the fruits name

```
fruits$name
```

```
## [1] "apples"  "berries" "mangos"  "bananas"
```

- What type of structure has the result?

```
class(fruits$name)
```

```
## [1] "character"
```

Your turn!

Using the patients data frame

```
##   first_name last_name age co_morbidity
## 1        Ava     Smith  65         TRUE
## 2    Richard     Brown  20        FALSE
## 3     Olivia  Williams  47        FALSE
```

- Extract the last_name column using the $ symbol
- Extract the age column. What's the class of the result?
- How do you extract the age of Richard using the previous result?

How do we add a column?

```
cbind(fruits,
      "hard_shell" = c(FALSE, FALSE, FALSE, FALSE))
```

```
##      name number edible_shell hard_shell
## 1  apples      1         TRUE      FALSE
## 2 berries     10         TRUE      FALSE
## 3  mangos      7        FALSE      FALSE
## 4 bananas      2        FALSE      FALSE
```

```
fruits
```

```
##      name number edible_shell
## 1  apples      1         TRUE
## 2 berries     10         TRUE
## 3  mangos      7        FALSE
## 4 bananas      2        FALSE
```

How do we add a column?

```
fruits$hard_shell <- c(FALSE, FALSE, FALSE, FALSE)

fruits
```

```
##      name number edible_shell hard_shell
## 1  apples      1         TRUE      FALSE
## 2 berries     10         TRUE      FALSE
## 3  mangos      7        FALSE      FALSE
## 4 bananas      2        FALSE      FALSE
```

How do we add a new row?

```
rbind(fruits, c("coconut", 3, FALSE, TRUE))
```

```
##      name number edible_shell hard_shell
## 1  apples      1         TRUE      FALSE
## 2 berries     10         TRUE      FALSE
## 3  mangos      7        FALSE      FALSE
## 4 bananas      2        FALSE      FALSE
## 5 coconut      3        FALSE       TRUE
```

```
fruits
```

```
##      name number edible_shell hard_shell
## 1  apples      1         TRUE      FALSE
## 2 berries     10         TRUE      FALSE
## 3  mangos      7        FALSE      FALSE
## 4 bananas      2        FALSE      FALSE
```

Your turn!

```
patients
```

```
##   first_name last_name age co_morbidity
## 1        Ava     Smith  65         TRUE
## 2    Richard     Brown  20        FALSE
## 3     Olivia  Williams  47        FALSE
```

- Add a new column ("glucose") with glucose levels
- Add a new patient (row)

Thanks!