Icahn
School of
Medicine at
Mount
Sinai

Introduction to coding with R

Part II

Joselyn C. Chávez Fuentes

01/16/2024

Let's recap

- How many dimensions does a vector have?
- What types of data can we store in a vector?
- What happens if we store different types of data in a vector?
- How do we create a new vector?
- How do we access the vector elements?

Using the name as index

```
x <- c(1,3,10)
names(x)
```

```
## NULL
```

```
x <- c("number_a" = 10, "number_b" = 50,
       "number_c" = 100)
x
```

```
## number_a number_b number_c
##       10       50      100
```

```
names(x)
```

```
## [1] "number_a" "number_b" "number_c"
```

```r
x <- c(10, 50, 100)

names(x) <- c("number_a", "number_b", "number_c")
x
```

```
## number_a number_b number_c 
##       10       50      100
```

```r
x["number_b"]
```

```
## number_b 
##       50
```

```r
x[c("number_a","number_c")]
```

```
## number_a number_c 
##       10      100
```

Using logical evaluation as index

```
x <- seq(1:10)
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
x < 5
```

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

```
x[x < 5]
```

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

```r
x <- c("a","a","b","c","c","c")
x == "c"
```

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

```r
x[x == "c"]
```

```
## [1] "c" "c" "c"
```

Your turn!

The following vector shows the temperature in New York during a week:

```r
temperature_nyc <- c("Monday" = 55,
                     "Tuesday" = 55,
                     "Wednesday" = 66,
                     "Thursday" = 50,
                     "Friday" = 48,
                     "Saturday" = 45,
                     "Sunday" = 47)
```

- Select the temperature of the first 3 days.
- Select the temperature of Tue, Thu and Sat.
- Which days had temperatures above average?

How do we modify a vector?

- Adding a new element

```
x <- c("a","b","c")
x
```

```
## [1] "a" "b" "c"
```

```
x[4] <- "d"
x
```

```
## [1] "a" "b" "c" "d"
```

```r
x <- c(x, "e")
x
```

```
## [1] "a" "b" "c" "d" "e"
```

- Removing an element

```r
x
```

```
## [1] "a" "b" "c" "d" "e"
```

```r
x[-2]
```

```
## [1] "a" "c" "d" "e"
```

```r
x <- x[-2]
x
```

```
## [1] "a" "c" "d" "e"
```

Your turn!

Using the vector:

```r
fruits <- c("apples" = 1,
            "cherries" = 10,
            "mangos" = 7)
```

- Add a new fruit to the vector
- Remove the cherries

- Replacing an element by index

```r
x
```

```
## [1] "a" "c" "d" "e"
```

```r
x[1] <- "m"
x
```

```
## [1] "m" "c" "d" "e"
```

- Replacing an element by logical evaluation

```r
x
```

```
## [1] "m" "c" "d" "e"
```

```r
x[x == "d"] <- "e"
x
```

```
## [1] "m" "c" "e" "e"
```

Excercise

Using the vector:

```
fruits <- c("apples" = 1,
            "cherries" = 10,
            "mangos" = 7)
```

- Select all fruits with values bigger than 5
- Replace the number of apples with a 4

Matrices

Creating a matrix

Matrices are objects with elements arranged in a two-dimensional layout.

```
my_matrix <- matrix(data = 1:12, nrow = 4)
my_matrix
```

```
##      [,1] [,2] [,3]
## [1,]    1    5    9
## [2,]    2    6   10
## [3,]    3    7   11
## [4,]    4    8   12
```

- rows and columns
- All elements must be the same type

```
my_matrix <- matrix(data = 1:12, ncol = 4)

my_matrix
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
```

```r
my_matrix <- matrix(data = 1:12,
                    ncol = 4,
                    byrow = TRUE)

my_matrix
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    5    6    7    8
## [3,]    9   10   11   12
```

Operations with matrices

Arithmetic operations

```
my_matrix + 10
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   11   12   13   14
## [2,]   15   16   17   18
## [3,]   19   20   21   22
```

```
my_matrix * 2
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    4    6    8
## [2,]   10   12   14   16
## [3,]   18   20   22   24
```

Arithmetic operations

```
matrix1 <- matrix(1:6, nrow = 2, ncol = 3)
matrix2 <- matrix(7:12, nrow = 2, ncol = 3)
```

```
matrix1
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
matrix2
```

```
##      [,1] [,2] [,3]
## [1,]    7    9   11
## [2,]    8   10   12
```

```
matrix1 + matrix2
```

```
##      [,1] [,2] [,3]
## [1,]    8   12   16
## [2,]   10   14   18
```

Your turn!

- Create a matrix with numbers 101:125, arrange them in 5 rows.
- Add 10 units, then multiply by 3

How do we access matrices elements?

```
matrix1 <- matrix(101:125, nrow = 5)
matrix1
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  101  106  111  116  121
## [2,]  102  107  112  117  122
## [3,]  103  108  113  118  123
## [4,]  104  109  114  119  124
## [5,]  105  110  115  120  125
```

```
matrix1[1:7]
```

```
## [1] 101 102 103 104 105 106 107
```

How do we access matrices elements?

```
matrix1[1,2]
```

```
## [1] 106
```

```
matrix1[1:2,1:2]
```

```
##      [,1] [,2]
## [1,]  101  106
## [2,]  102  107
```

Your turn!

- How would you select the number 120 from the matrix?
- How would you select the last two columns?

Some extra information about packages

Two main repositories

CRAN

- 20272 available packages
- Topics: Statistics, machine learning, plotting, economy, spatial data, databases, phylogenetics, natural language processing, ...

- https://cran.rstudio.com > Packages

- How do we install packages from CRAN?

```
install.packages("ggplot2")
```

Two main repositories

Bioconductor

- Software, Annotation, and Experiment packages
- 2266 software packages

- https://bioconductor.org

- How do we install packages from Bioconductor?

```
install.packages("BiocManager")
BiocManager::install("Biostrings")
```

Installing packages from the source

- Main source of code: GitHub
- Under development packages, small packages, in preparation for submitting to CRAN or Bioconductor...

- How do we install packages from GitHub?

```r
install.packages("remotes")
remotes::install_github("nstrayer/datadrivencv")
```

Thanks!