

Examples R

Useful commands

1. Check which is the working directory
2. Set the working directory for the workbench
3. List environment variables
4. Create a variable (ex: x=1)
5. List environment variables
6. Delete the created variable
7. List environment variables

Variables

1. Check that it is possible to create the following variables:
 - 1.1. division with value 3
 - 1.2. Square with the value "blue"
 - 1.3. .sub .multiplication with value "a"
 - 1.4. accumulative_sum with value 2
 - 1.5. Sum5 with the value 10
2. Check that it is not possible to create the following variables, regardless of the value
 - 2.1. tot@l
 - 2.2. 5um
 - 2.3. _fine
 - 2.4. FALSE
 - 2.5. .0three

constants

1. Check the type of the following constants:
 - 1.1. two
 - 1.2. 2L
 - 1.3. 2i
 - 1.4. 'example'
 - 1.5. 'two'

operators

1. Assignment: assign the following values to the variable x
 - 1.1. Value 20 (operator <-)
 - 1.2. Value 30 (operator =)
 - 1.3. Value 5 (operator ->)
2. arithmetic
 - 2.1. Create a variable x with the value 20 and a variable y with the value 15 and perform the following operations between the variables:
 - 2.1.1. Sum
 - 2.1.2. Subtraction
 - 2.1.3. Multiplication
 - 2.1.4. Division
 - 2.1.5. Whole part of the division
 - 2.1.6. Rest of Division
 - 2.1.7. exponentiation
 - 2.2. Relational: Create a variable x with the value 20 and a variable y with the value 15 and perform the following operations:
 - 2.2.1. Check if x is less than y
 - 2.2.2. Check if x is greater than y
 - 2.2.3. Check if x is less than or equal to y
 - 2.3. Logical: Create the vector x with the elements (TRUE, FALSE, 0, 3) and the vector y with the elements (FALSE, TRUE, FALSE, TRUE) and perform the following operations:
 - 2.3.1. no x
 - 2.3.2. x E y (element-wise)
 - 2.3.3. x OR y (logical)

Precedence

Check precedence in the following operations

1. $3 + 4 / 2$
2. $(3 + 4) / 2$
3. $3 / 4 / 2$
4. $3 / (4 / 2)$

vectors

1. Creating vectors using the `c()` function:
 - 1.1. Elements: 1, 5, 4, 9, 0
 - 1.2. Elements: 1, 5.4, TRUE, "hello"
2. Creating vectors using the ":" operator
 - 2.1. Create a vector with the integers from 1 to 7
 - 2.2. Create a vector with the integers from 2 to -2
3. Creating vectors using the `seq()` function
 - 3.1. Create a vector with values from 1 to 3 with a range of 0.2 (ex: 1, 1.2, 1.4, ...)
 - 3.2. Create a vector with values from 1 to 5 with 4 elements
4. Functions for vectors
 - 4.1. Create the following vector: `a <- c(4, 3, 2, 5, 2, 4, 3, 1, 2, 4)`
 - 4.2. Get vector size
 - 4.3. Sort the vector ascending
 - 4.4. Get the unique values of the vector
 - 4.5. Get the frequency of vector elements
5. Reading using logical vector as index
 - 5.1. Create the vector `b <- seq(-3, 3, 2)`
 - 5.2. Show only the first and fourth elements of the vector
 - 5.3. Show negative vector elements
 - 5.4. Show the positive elements of the vector
6. Reading using character vector as index
 - 6.1. Create vector `d <- c(first=3, second=0, third=9)`
 - 6.2. View element names
 - 6.3. Get the element whose name is "second"
 - 6.4. Get the elements named "first" and "third"
7. modify a vector
 - 7.1. Create a vector and `<- seq(-3, 9, 1)`
 - 7.2. Replace the second element of the vector with the value 0
 - 7.3. Replace negative elements with the value 12
 - 7.4. Create a vector `f` with the first 4 elements of the vector and

matrices

1. Create a matrix using the `matrix()` function
 - 1.1. Values from 1 to 9, with 3 rows and 3 columns
 - 1.2. Values from 1 to 9, with 3 rows and 3 columns, spread over rows
 - 1.3. Change row names to L1, L2, L3 and column names to C1, C2, C3
2. Array creation using `cbind()` and `rbind()` functions
 - 2.1. Create a matrix by column joining the vectors `c(1,2,3)` and `c(4,5,6)`
 - 2.2. Create a matrix by row joining the vectors `c(7,8,9)` and `c(10,11,12)`
3. Reading using an array of integers as index (using array from 1.1)
 - 3.1. Show only the first row
 - 3.2. Show only the first column
 - 3.3. Show the entire array except the first row
 - 3.4. Show the intersection of rows 1 and 2 with columns 2 and 3
 - 3.5. Show all columns of rows 1 and 2
4. Reading using an array of logical values as an index
 - 4.1. Show the existing even values in the array (note: the remainder of dividing an even number by 2 is 0)
5. Reading using an array of characters as an index (using the array from 1.3)
 - 5.1. Show columns whose names are C1 and C2
 - 5.2. Show the lines whose names are L2 and L3
 - 5.3. Show the intersection of the second and third rows with columns C1 and C2
6. modify an array
 - 6.1. Replace element in second row and second column with 10
 - 6.2. Replace elements less than 5 with 0
 - 6.3. Get the matrix transpose
 - 6.4. Add the column from the vector `c(20,30,40)` to the matrix
 - 6.5. Add to the matrix the row from the vector `c(500,600,700,800)`
 - 6.6. Remove the third row from the matrix

lists

1. Create a list using the `list()` function
 - 1.1. Create a list with 3 elements: element "a" with value 2.5; element "b" with value TRUE, and element "c" with integers from 1 to 3
 - 1.2. Check list structure
2. Read elements from a list
 - 2.1. Get the elements in positions 1 to 2 of the list
 - 2.2. Get all elements of the list except the second

- 2.3. Using a logical vector, get the first element of the list
- 2.4. Create a list with the following values: "name" with the value "John", "age" with the value 19, "speaks" with the values "English" and "French"
- 2.5. Get the "age" element
- 2.6. Get the "age" and "speaks" elements
- 2.7. Get the "name" element using the \$ operator
- 2.8. Get the first element of "speaks" using the \$ operator
3. modify a list
 - 3.1. Change the "name" from the previous list to "Clair"
 - 3.2. Add to the list the element "married" with the value FALSE

data frames

1. Create a data frame using the data.frame () function
 - 1.1. Create a data frame with the following content

	SN	Age	Name
1	21	John	
2	15	Dora	
 - 1.2. View the data frame structure
2. read data frames
 - 2.1. View Names with three operators: [, [[]] and \$
 - 2.2. See the name of the second record
 - 2.3. View the iris dataset
 - 2.4. See the first 3 lines of the iris dataset
 - 2.5. View the records of the iris dataset whose Species is "setosa"
3. Modify a data frame
 - 3.1. In the data frame of the first exercise, change John's age to 20
 - 3.2. Add the "State" column to the data frame of the first exercise with the values "NY" and "FL" for records 1 and 2, respectively

factors

1. Create a factor using the factor() function
 - 1.1. Create a factor with the following values: "single", "married", "married", "single"
 - 1.2. Create a factor with the following values: "single", "married", "married", "single", defining the levels as: "single", "married", "divorced"
2. modify a factor
 - 2.1. Modify the second element from 1.2 to "single"
 - 2.2. Modify second element from 1.2 to "widowed" (impossible)



predefined functions

1. math functions
 - 1.1. Calculate the square root of 2
 - 1.2. Calculate the cosine of pi
 - 1.3. Calculate the absolute value of -3
2. text functions
 - 2.1. Create the variable x with the value "abcdef"
 - 2.2. Show the letters in positions 2 to 4
 - 2.3. Find "A" in c("B", "A", "C")
 - 2.4. Create the vector [x1 x2 x3] by concatenating the letter x with the numbers 1 to 3
3. statistical functions
 - 3.1. Create the vector x with elements 2, 5 and 7
 - 3.2. Calculate the mean of x
 - 3.3. Calculate the maximum value of x
4. Useful functions
 - 4.1. Create a vector with 5 repetitions of the value 20