

Basics of R

Laika Jean Paculba

March 15, 2023

`<-` operator to store an object

`#`-can be used to put a comment

```
my_sum <- 3 + 3
my_sum + 3
```

`my_sum` => *show the value of my_sum*

`new_sum <- my_sum + 3` => *assign my_sum to new_sum and show the value*

INTEGERS

- can be signified by adding an **'L'** to the end

```
my_integer <- 1L
my_double <- 6.38
```

typeof() function

- used to see the type of a single scalar value

```
typeof(my_integer)
typeof(my_double)
```

NUMERIC DATA

- can be in integer form or double (decimal) form.

```
my_integer <- 1L
my_double <- 6.38
typeof(my_integer)
typeof(my_double)
```

```
my_integer <- 9L
typeof(my_integer)
my_double <- 7.98
typeof(double)
```

CHARACTER DATA

- text data surrounded by single or double quotes

```
my_character <- "This is text"
typeof(my_character)
```

LOGICAL DATA

- takes the form TRUE or FALSE

```
my_logical <- TRUE
typeof(my_logical)
my_logical <- TRUE
typeof(TRUE)
```

`typeof(my_logical) => verify if its indeed logical`

HOMOGENOUS DATA STRUCTURES

VECTORS

- one-dimensional structures containing data of the same type and are notated by using `c()`.
- `typeof()` - a function used to view the type of vector
- `str()` - a function used to display both the contents of the vector and its type.

```
my_double_vector <- c(2.3, 6.8, 4.5, 65, 6)
str(my_double_vector) #verify
type of(my_double_vector)
```

CATEGORICAL DATA

- which takes only a finite number of possible values
- can be stored as a factor vector to make it easier to perform grouping and manipulation.

define a vector of the factor which is not

```
categories <- factor(c("A", "B", "V", "A", "C"))
str(categories) # verify
categories <- factor(c("A","B","C","A","C"))
```

Factors can be given order

character vector:

```
ranking <- c("Medium", "High", "Low")
str(ranking)
```

turn it to order factor:

```
ranking_factors <- ordered(
  ranking, levels = c("Medium", "High", "Low")
)
str(ranking_factors)
```

contents and type of the vector which is ordered type:

```
((ranking <- c("Medium", "High", "Low")))
ranking_order <- order(ranking, levels(c("Medium", "High", "Low")))

ranking_order <- order(ranking, levels(c("Medium", "High", "Low")))
str(ranking_order)
```

contents and type of the vector which is ordered type

link(categories)

COERCION

- result of poor design
- one or more of the types are forced into a different type to ensure homogeneity

simple numeric sequence vectors:

```
(my_sequence <- 1:10)
(my_sequence_2 <- seq(from=1, to=10)) #same above, but flexible than above
(seq_five <- seq(from=5, to=500, by=5)) #example
```

numeric sequence vector:

```
vec <- 1:5
str(vec)
```

create a new vector containing vec and the character hello:

```
new_vec <- c(vec, "hello") #creates a vector from the previous vector
```

numeric value have been coerced into their character equivalences:

```
str(new_vec) # the new elements become a type elements
```

```
vec[1] + vec[2] # result is 3
new_vec[1] + new_vec[2]
```

- result to error
- keep in mind to know what is the type of data

attempt to a mixed logical and numeric:

```
mix <- c(TRUE, 6)
```

logical has been converted to binary numeric (TRUE = 1):

```
str(mix)
```

try to add a numeric to our previous categories factor vector:

```
new_categories <- c(categories, 1)
```

categories have been coerced to background integer representations:

```
str(new_categories)
```

MATRICES

- two dimensional data structures of the same type and are built from a vector by defining the number of rows and columns.
- you can convert a vector to matrix

Create a 2x2 matrix with the first four integers

```
(m <- matrix( c(1, 2, 3, 4), nrow = 2, ncol = 2 ))
```

```
(m <- matrix(c(1,2,3,4), nrow=2, ncol=3))
```

```
(m <- matrix(c(1,2,3,4), nrow=3, ncol=3))
```

ARRAYS

- are n-dimensional data structure with the same data type and are not used extensively by most r users
- not really important