

LABWORK – 3

Variables declaration in R

1. Declaring a variable of type “numeric”

```
> myVariable <- numeric()           # Creating a numeric variable
> class(myVariable)
[1] "numeric"
> myVariable1 = as.numeric()
> class(myVariable1)
[1] "numeric"
```

2. Declaring a variable of type “character”

```
> myVariable2 <- character()        # Creating a character variable
> class(myVariable2)
[1] "character"
> myVariable3 = as.character()
> class(myVariable3)
[1] "character"
```

3. Declaring a variable of type “logical”, “complex” & “integer”

> myVariable4 = as.logical ()	> myVariable7 <- logical ()
> class (myVariable4)	> class (myVariable7)
[1] "logical"	[1] "logical"
> myVariable5 = as.complex ()	> myVariable8 <- complex ()
> class (myVariable5)	> class (myVariable8)
[1] "complex"	[1] "complex"
> myVariable6 = as.integer ()	> myVariable9 <- integer ()
> class (myVariable6)	> class (myVariable9)
[1] "integer"	[1] "integer"

Variables assignment in R

1. Assigning a decimal value to a variable

```
> myVariable = 3.14  
> class (myVariable)  
[1] "numeric"
```

2. Assignment of an integer value

```
> myInteger = as.integer (1)  
> myInteger  
[1] 1  
> class (myInteger)  
[1] "integer"
```

3. Assignment of a complex number

```
> z = 2+3i  
> class (z)  
[1] "complex"
```

4. Assignment of a logical value

```
> myMoodIsHappy = TRUE  
> class (myMoodIsHappy)  
[1] "logical"  
> !myMoodIsHappy  
[1] FALSE
```

5. Assignment of an integer value

```
> myName = "beu"  
> class (myName)  
[1] "character"
```

Data type of a variable in R

```
> myVariable
[1] 3.14
> typeof(myVariable)
[1] "double"

> typeof(TRUE)
[1] "logical"

> typeof(0)
[1] "double"

> typeof(0i)
[1] "complex"

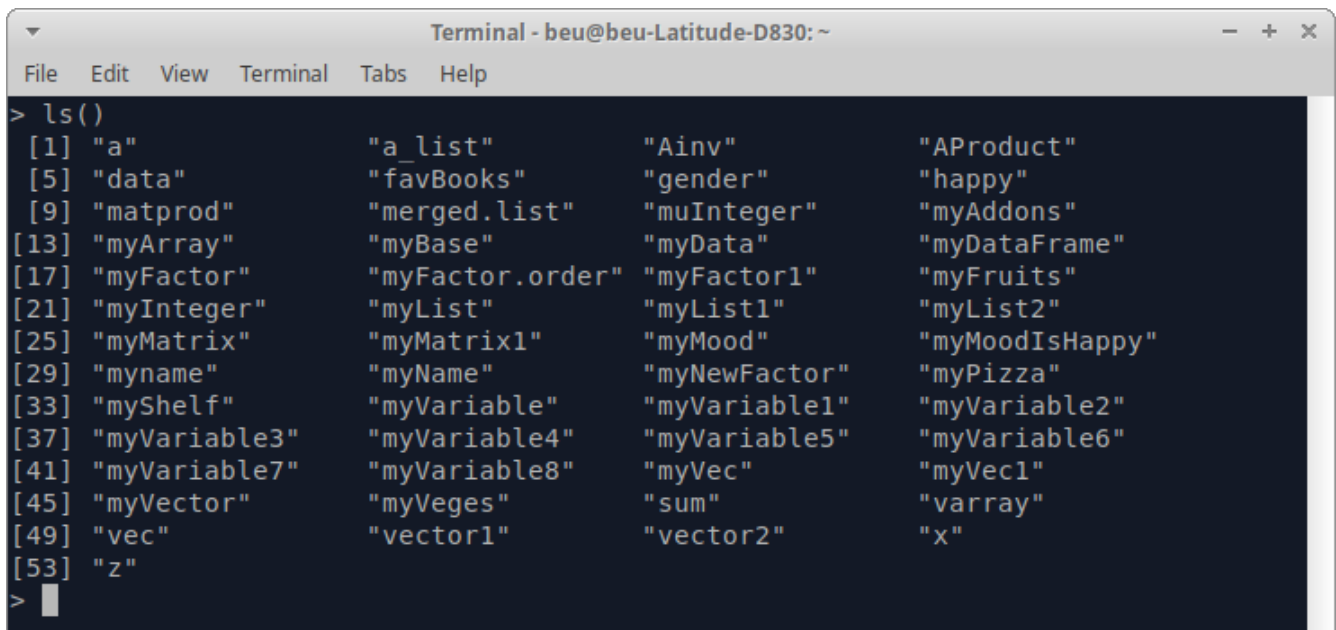
> x <- as.numeric()
> typeof(x)
[1] "double"

> typeof(1)
[1] "double"

> typeof(0L)
[1] "integer"

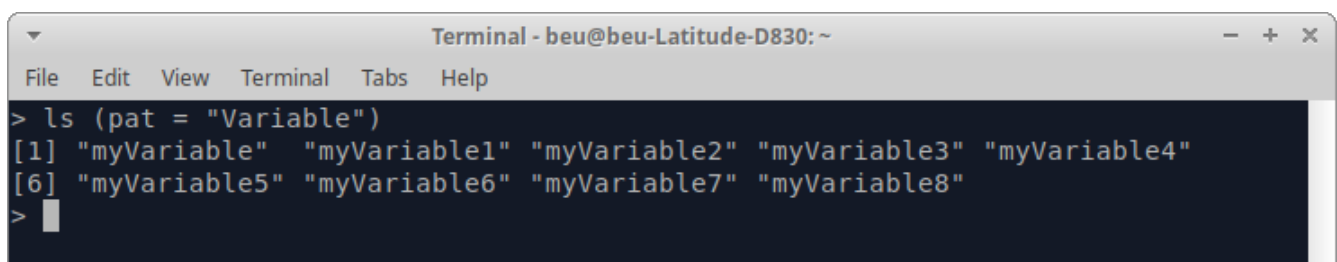
> typeof("beu")
[1] "character"
```

Listing all variables



```
Terminal - beu@beu-Latitude-D830: ~
File Edit View Terminal Tabs Help
> ls()
[1] "a" "a_list" "Ainv" "AProduct"
[5] "data" "favBooks" "gender" "happy"
[9] "matprod" "merged.list" "muInteger" "myAddons"
[13] "myArray" "myBase" "myData" "myDataFrame"
[17] "myFactor" "myFactor.order" "myFactor1" "myFruits"
[21] "myInteger" "myList" "myList1" "myList2"
[25] "myMatrix" "myMatrix1" "myMood" "myMoodIsHappy"
[29] "myname" "myName" "myNewFactor" "myPizza"
[33] "myShelf" "myVariable" "myVariable1" "myVariable2"
[37] "myVariable3" "myVariable4" "myVariable5" "myVariable6"
[41] "myVariable7" "myVariable8" "myVec" "myVec1"
[45] "myVector" "myVegetables" "sum" "varray"
[49] "vec" "vector1" "vector2" "x"
[53] "z"
>
```

Listing all variables which contain the word “Variable”



```
Terminal - beu@beu-Latitude-D830: ~
File Edit View Terminal Tabs Help
> ls(pat = "Variable")
[1] "myVariable" "myVariable1" "myVariable2" "myVariable3" "myVariable4"
[6] "myVariable5" "myVariable6" "myVariable7" "myVariable8"
>
```

Listing all variables which contain “.”

```
Terminal - beu@beu-Latitude-D830: ~
File Edit View Terminal Tabs Help
> ls (pat = "\\.", all.names = FALSE)
[1] "merged.list"      "my.name"          "myFactor.order"
> ls (pat = "\\.")
[1] "merged.list"      "my.name"          "myFactor.order"
> ls (pat = "\\.", all.names = TRUE)
[1] ".name"            "merged.list"      "my.name"          "myFactor.order"
> █
```

all.names – is a logical value. If it is TRUE, all object names are returned. If FALSE, names which begin with a . are omitted.

Deleting variables in R

```
> favBooks
[1] "Discrete_Mathematics" "Linear_Algebra"
[3] "Graph_Theory"        "Mathematical_Statistics"
[5] "Abstract_Algebra"    "Algorithm_Design"
[7] "Computer_Networking"
> remove(favBooks)          # using remove
> favBooks
Error: object 'favBooks' not found
```

```
> myData
[1] 1 2 2 4 6 3 1 2 3 4 6 6 1 3 1 2 3 3 1
> rm(myData)                # using rm
> myData
Error: object 'myData' not found
```

```
> rm(list = ls())           # removes all objects from the current workspace
> ls()
character(0)
```

Relational Operators in R

FUNCTION	R EXPRESSION	WORKING EXAMPLE
Equality	==	<pre>> x = 7 > x == 7 [1] TRUE > x == 3 [1] FALSE</pre>
Inequality	!=	<pre>> y = -3 > y != -3 [1] FALSE > y != 4 [1] TRUE</pre>
Less than	<	<pre>> 5 < 2 [1] TRUE > 5 < 5 [1] FALSE > 5 < 10 [1] FALSE</pre>
Greater than	>	<pre>> 5 < 2 [1] FALSE > 5 < 5 [1] FALSE > 5 < 10 [1] TRUE</pre>
Less than or equal to	<=	<pre>> a = 10 > b = 5 > a <= b [1] FALSE</pre>
Greater than or equal to	>=	<pre>> a = 1 > b = 0 > a >= b [1] TRUE</pre>

Logical Operators in R

FUNCTION	R EXPRESSION	WORKING EXAMPLE
Logical NOT	!	> myMoodIsHappy [1] TRUE > ! myMoodIsHappy [1] FALSE
Element-wise logical AND	&	> TRUE & TRUE [1] TRUE > TRUE & FALSE [1] FALSE > FALSE & FALSE [1] FALSE > FALSE & TRUE [1] FALSE
Element-wise logical OR		> TRUE TRUE [1] TRUE > TRUE FALSE [1] TRUE > FALSE TRUE [1] TRUE > FALSE FALSE [1] FALSE
Logical AND	&&	> x <- c(TRUE, FALSE, FALSE) > y <- c(FALSE, FALSE, FALSE) > x && y [1] FALSE
Logical OR		> x <- c(TRUE, FALSE, FALSE) > y <- c(FALSE, FALSE, FALSE) > x y [1] TRUE

Assignment Operators in R

FUNCTION	R EXPRESSION	WORKING EXAMPLE
Leftwards assignment	<-, <<-, =	<pre>> myNum <- 7 > myNum [1] 7 > myNum1 = 10 > myNum1 [1] 10 > myNum2 <<- 13 > myNum2 [1] 13</pre>
Rightwards assignment	->, ->>	<pre>> 1 -> newNum1 > newNum1 [1] 1 > 2 ->> newNum2 > newNum2 [1] 2</pre>