

Attributes 1

<code>attr()</code>	to retrieve or set single attributes
<code>structure()</code>	to set all the attributes in one shot
<code>attributes()</code>	to list all the attributes in one shot

(**Basics)

Matrices 1

```
mat2<-matrix(c(1,2,3, TRUE, FALSE, FALSE), 3, 2)
typeof(mat2)
[1] "double"
+ Subsetting matrices
The element in 1 row and 3column: mat[1,3]
Extract the second row: mat[2, ]
Extract the second column: mat[, 2]
Erase the second column: mat[, -2]
Select a subset: mat[1: 2, 1:2]
```

(**Basics)

Matrices 2

<code>t(A)</code>	transposed
<code>diag(A), diag(c(.))</code>	diagonal
<code>solve(.)</code>	inverse
<code>matrix(1:6,2,3)+ matrix(6:1,2,3)</code>	sum
<code>matrix(1:6,2,3)- matrix(6:1,2,3)</code>	subtract
<code>matrix(1/(1:6),2,3)* matrix(1:6,2,3)</code>	product elem
<code>matrix(1/(1:6),2,3)%*% matrix(1:6,2,3)</code>	product
<code>crossprod(A,B)</code>	cross-- product
<code>outer(A,B)</code>	outer product
<code>rbind() cbind()</code>	bind by rows, by columns

(**Basics)(**Advanced)

Attributes 2

+ Set two different attributes

```
x<-1:10
attr(x,"att1") <- "AttV"
attr(x,"att2") <- 14
+ Same with structure:
x <- structure(1:10, att1 = "AttV", att2=14)
+ Names:
names(x) <- c(.)
setNames(.,c(.))
+ Unnames:
unname(x)
```

```
names(x)<-NULL
+Identical?
identical(mat1,mat2)
+Yes, not the SAME
lobstr::obj_addr(mat1)
[1] "0xf3107a0"
lobstr::obj_addr(mat2)
[1] "0xf310610"
```

(**Basics)

Lists 1 - generic vector

```
(lis1 <- list(1:5, c("a","b"),
c(TRUE,FALSE,TRUE), 2.3))
[[1]]
[1] 1 2 3 4 5
[[2]]
[1] "a" "b"
[[3]]
[1] TRUE FALSE TRUE
[[4]]
[1] 2.3
```

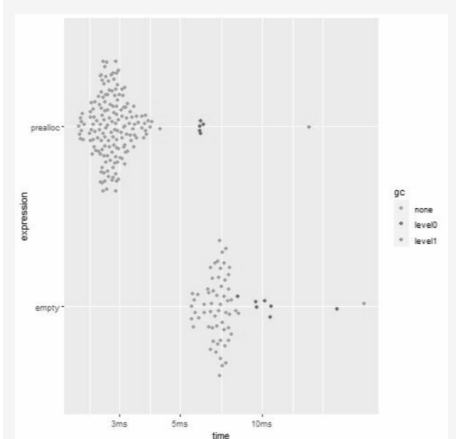
A list can contain lists.

Lists 2

<code>typeof(lis1)</code>	"list"
<code>class(lis)</code>	"list"
<code>attributes(lis1)</code>	NULL
<code>is.object(lis1)</code>	FALSE
<code>is.list(lis1)</code>	TRUE
<code>str(lis1)</code>	List of 4 \$: int [1:5] 1 \$: chr [1:2] "a" \$: logi [1:3] T \$: num 2.3
<code>lobstr::ref()</code>	compare changes in t
<code>unlist(lis1)</code>	"1" "2" "3" "a" "b" "1" "
<code>list_name[c(.)]</code>	Subset elements
<code>list_name[[]]</code>	
<code>list\$name</code>	

(**Basics)(**Advanced)

Lists 3: Preallocation vs Empty



vector ("li st",5) vs list()
(**Advanced)

