

Data Structures in R: Lists

Stat 133 with Gaston Sanchez

Creative Commons Attribution Share-Alike 4.0 International CC BY-SA

Lists

single data type

multiple data types

Vector

List

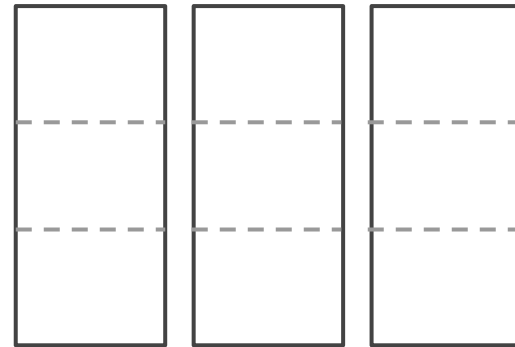
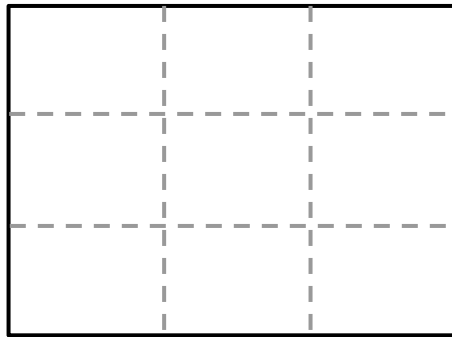
1D



Matrix

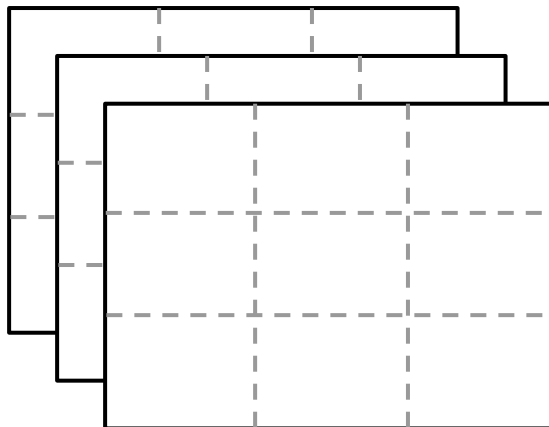
Data Frame

2D



Array

nD



non-atomic
structures

dimensions

R lists

A list is the most general data structure in R

Lists can contain any other type of data structure

Lists can even contain other lists

List of Vectors (of equal length)

--	--	--

--	--	--

--	--	--

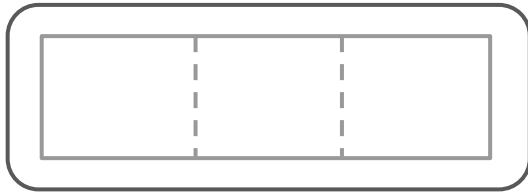
List of Vectors (of different length)

--	--	--

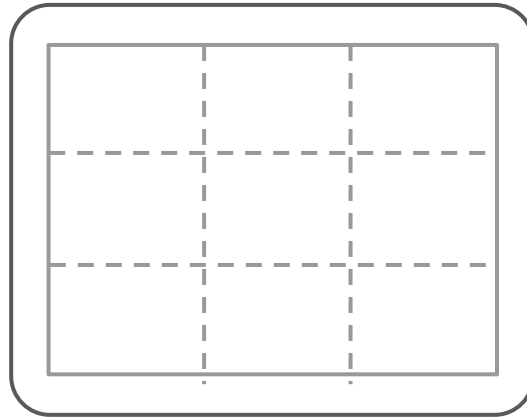
--	--

--	--	--	--

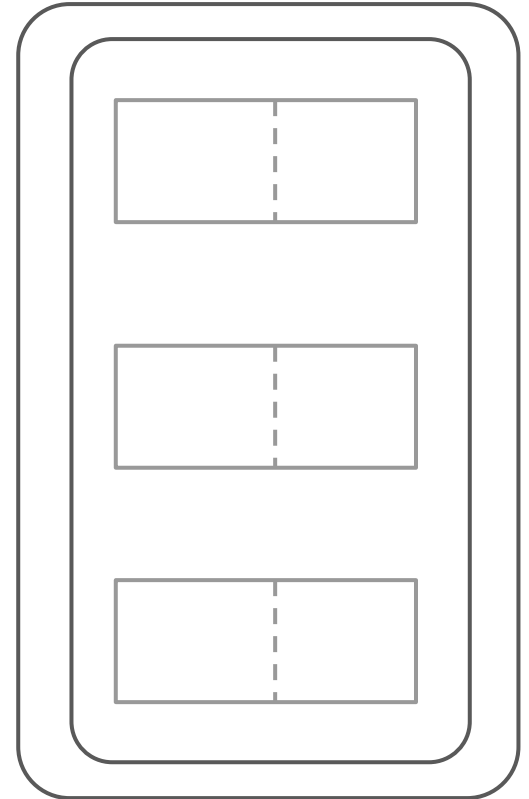
List of various objects



vector



matrix



Other lists

R lists

Lists are a special type of vector

```
lst <- vector(mode = "list")
```

Lists are vectors in the sense of being a one-dimensional object

Lists are NOT atomic structures

Subsetting and Indexing

Bracket Notation System

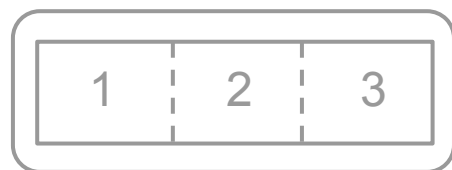
To extract values from R objects use brackets: []

Inside the brackets specify vector(s) of indices

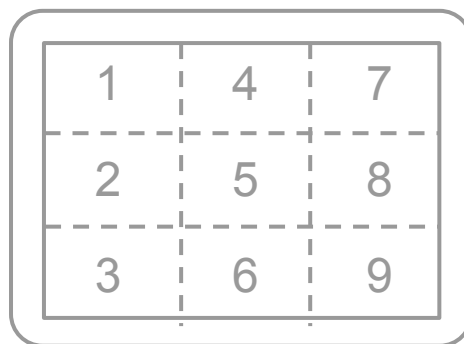
Use as many indices, separated by commas, as dimensions in the object

Vector(s) of indices can be numbers, logicals, and sometimes names

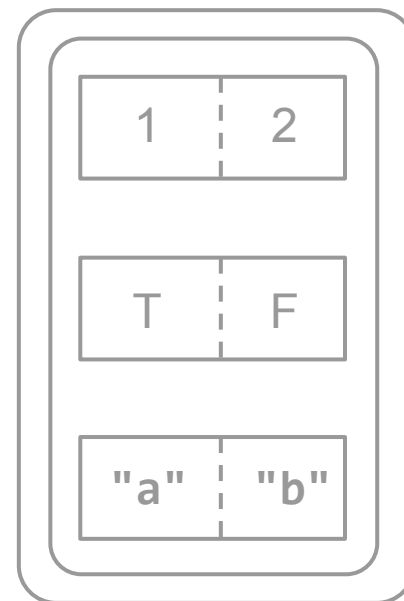
```
lst <- list(  
  c(1, 2, 3),  
  matrix(1:9, nrow = 3, ncol = 3),  
  list(1:2, c(TRUE, FALSE), c("a", "b"))  
)
```



vector



matrix

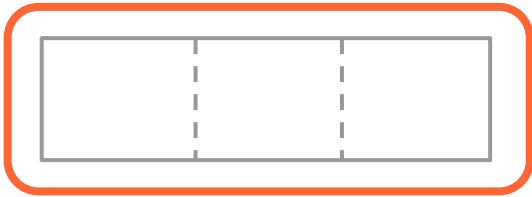


list of vectors

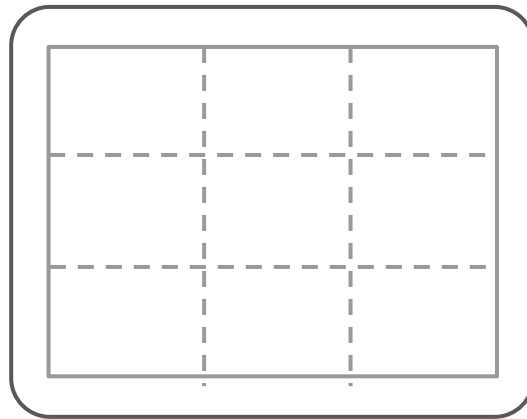
access list element(s)

`list[elem]`

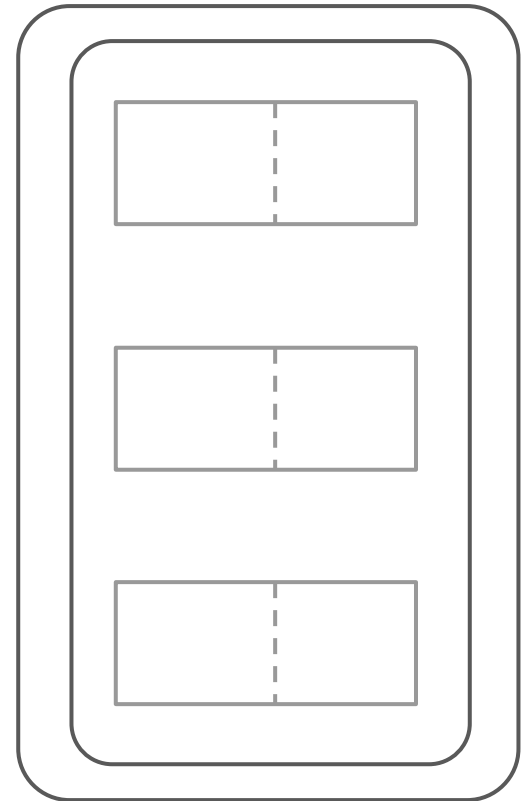
`lst[1]`



vector

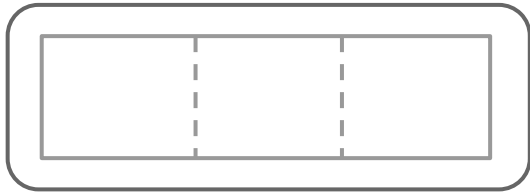


matrix

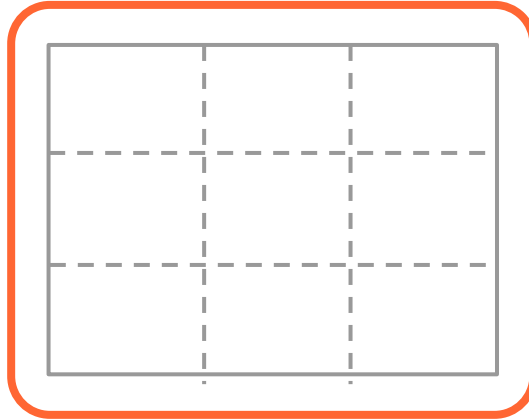


another list

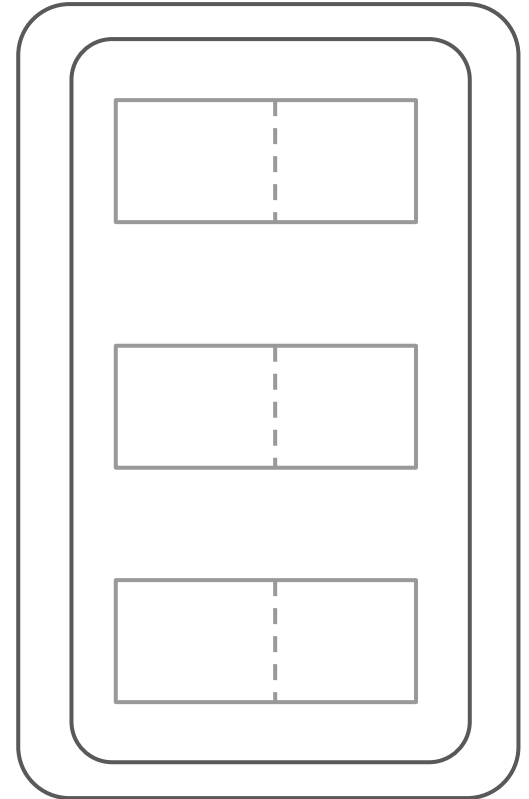
1st[2]



vector

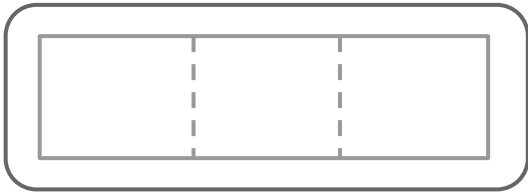


matrix

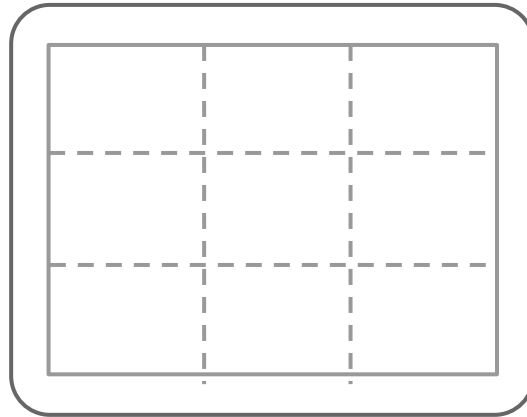


another list

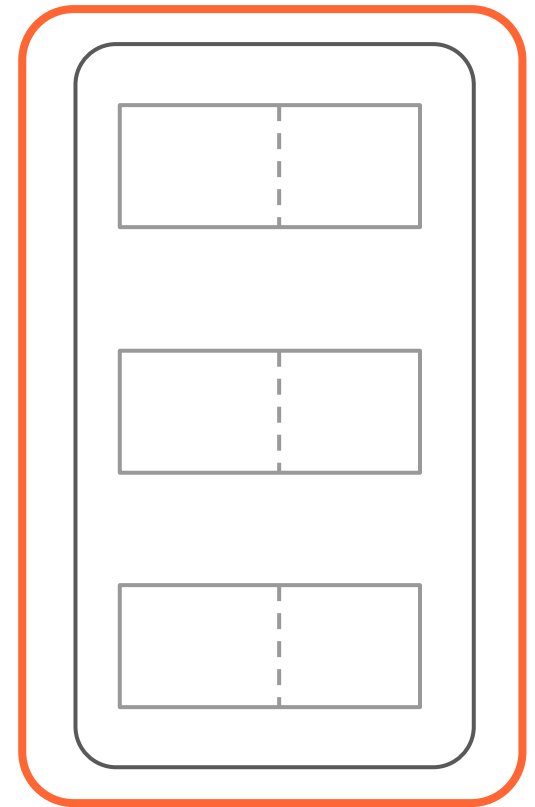
1st[3]



vector



matrix

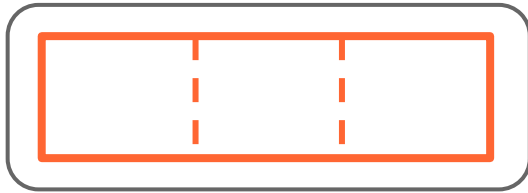


another list

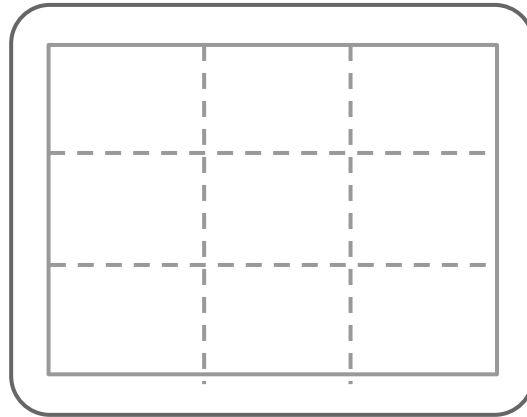
access object of list element

```
list[[elem]]
```

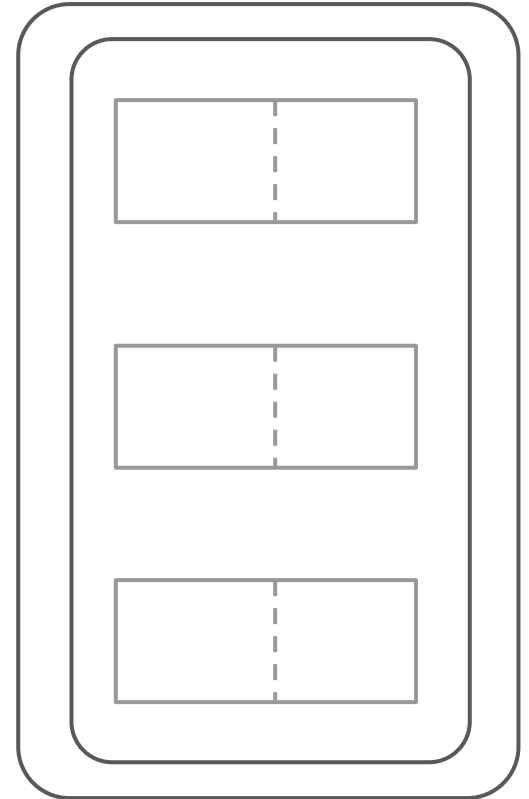
`1st[[1]]`



vector

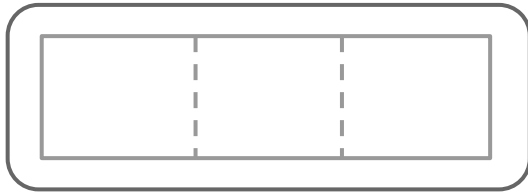


matrix

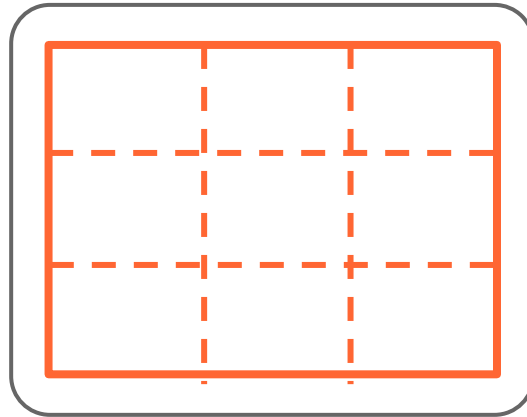


another list

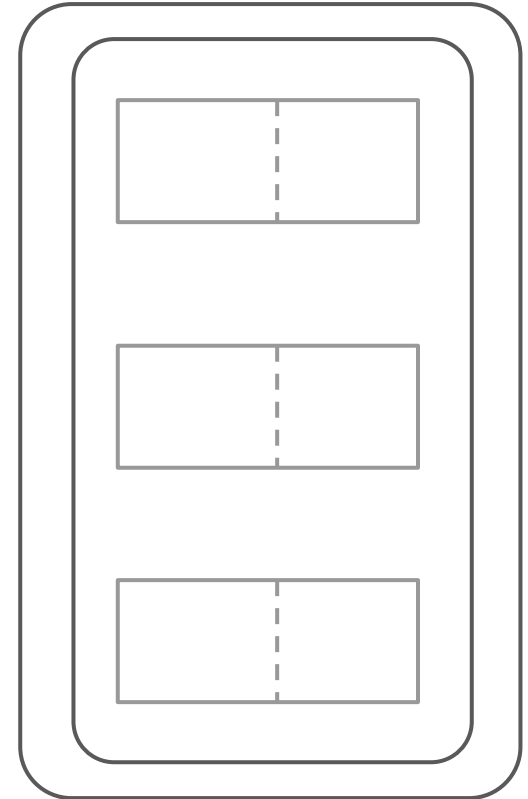
`1st[[2]]`



vector

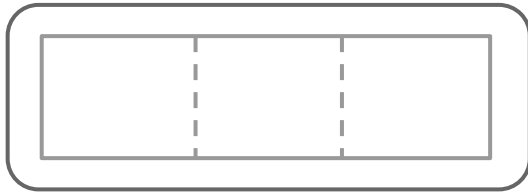


matrix

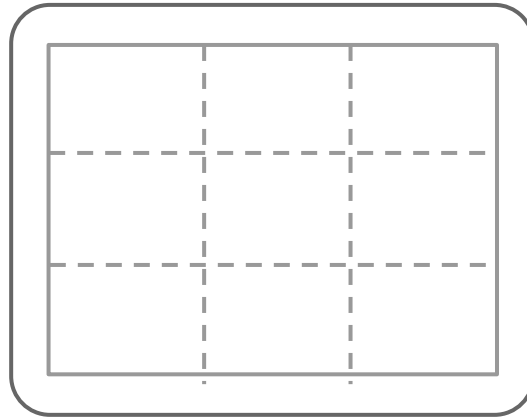


another list

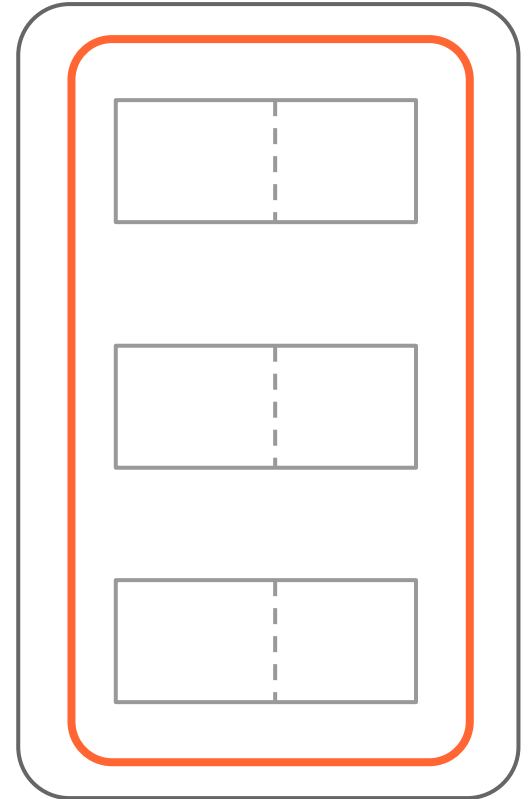
`1st[[3]]`



vector



matrix



another list

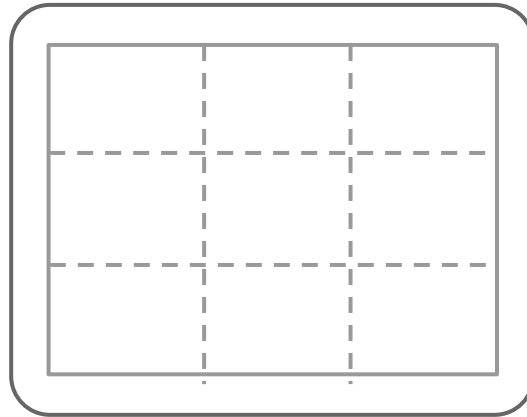
access object's elements,
of list element

```
list[[elem]][obj]
```

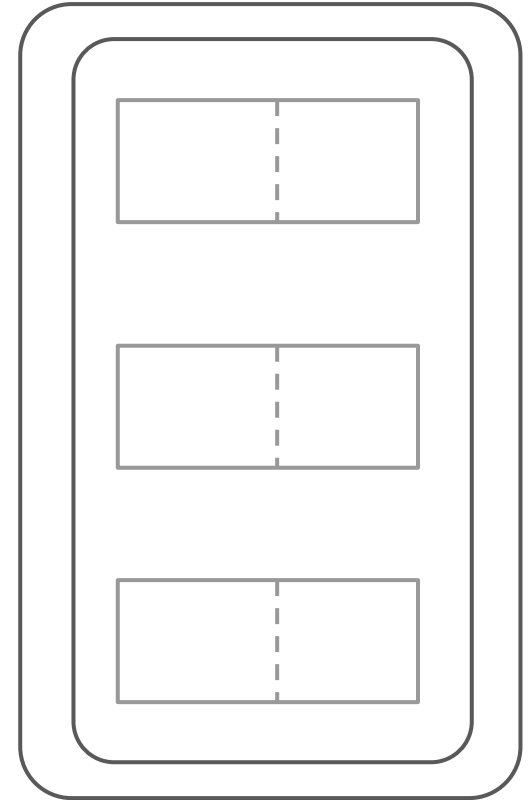
1st[[1]][2]



vector

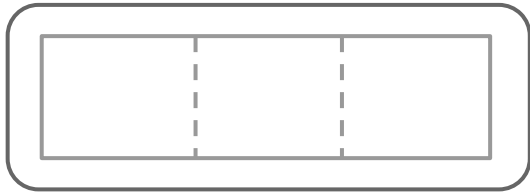


matrix

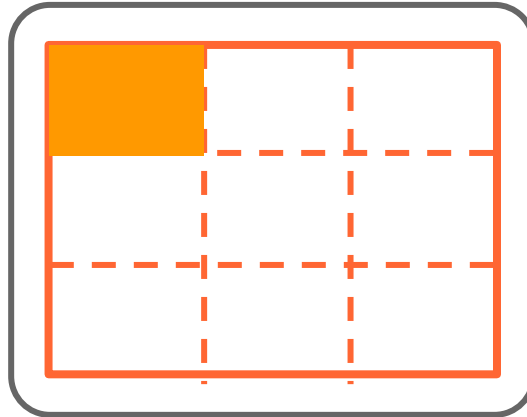


another list

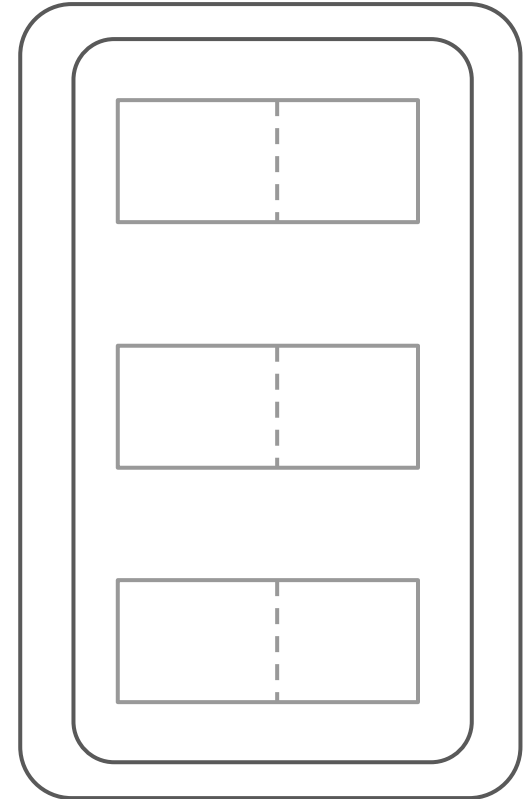
`1st[[2]][1,1]`



vector

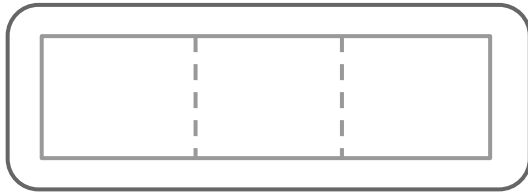


matrix

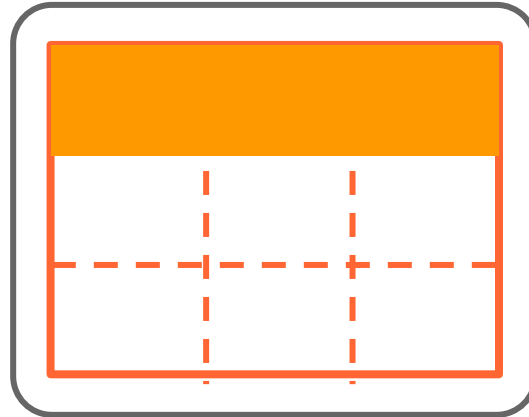


another list

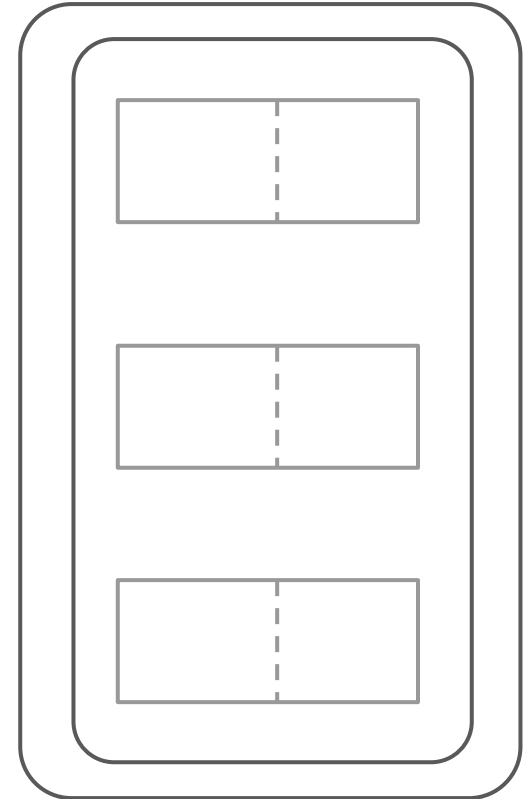
`1st[[2]][1,]`



vector

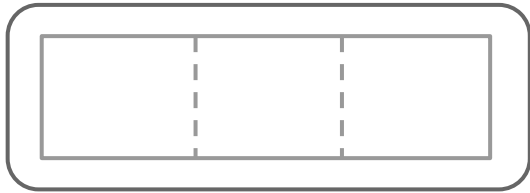


matrix

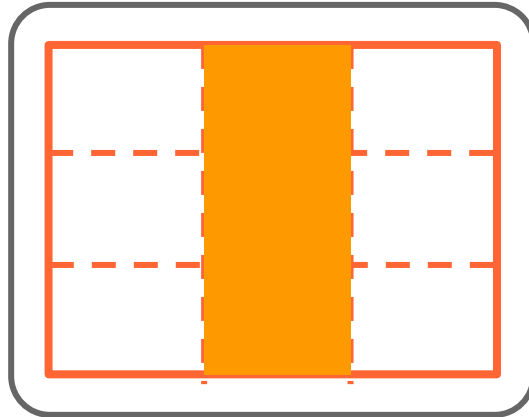


another list

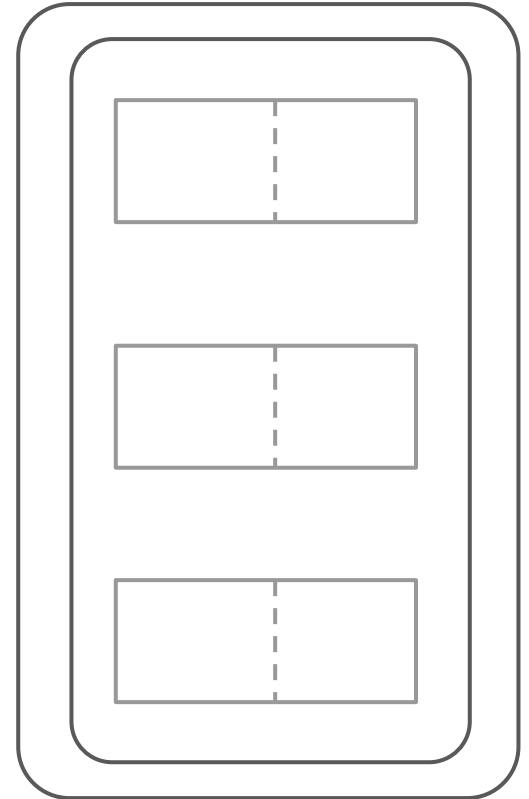
`1st[[2]][,2]`



vector

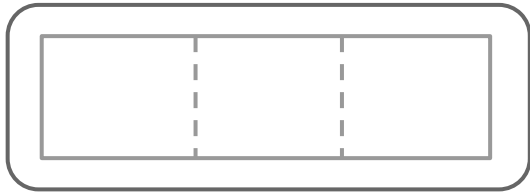


matrix

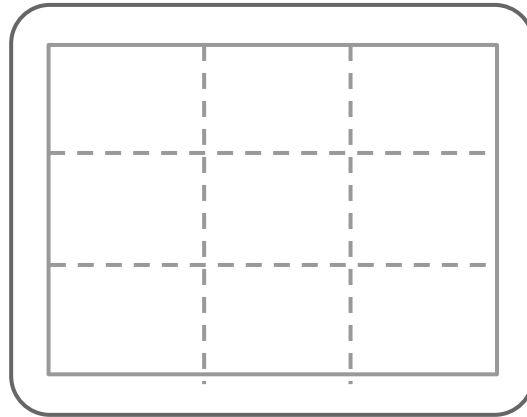


another list

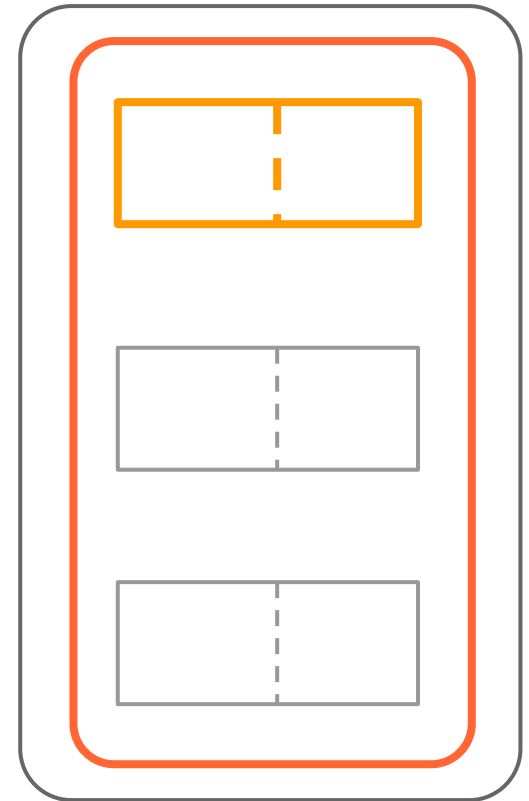
`1st[[3]][1]`



vector



matrix

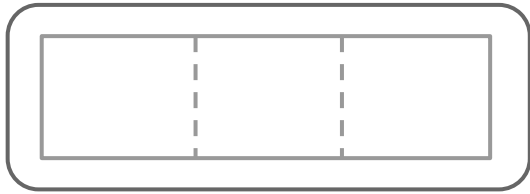


another list

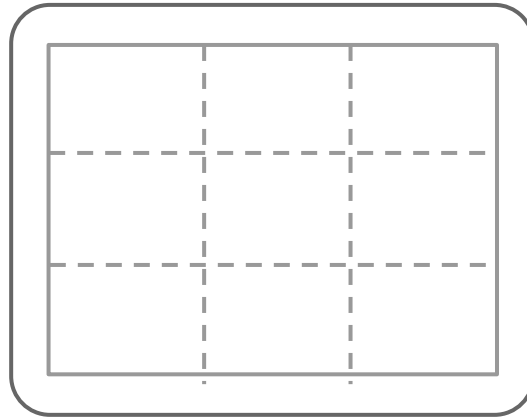
access object's elements,
of list element

```
list[[elem]][[obj]]
```

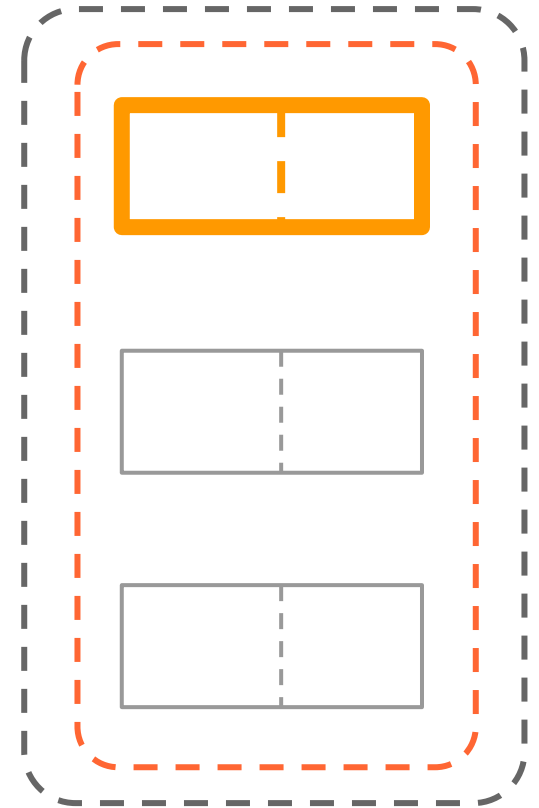
1st[[3]][[1]]



vector



matrix

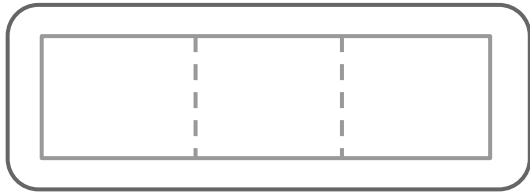


another list

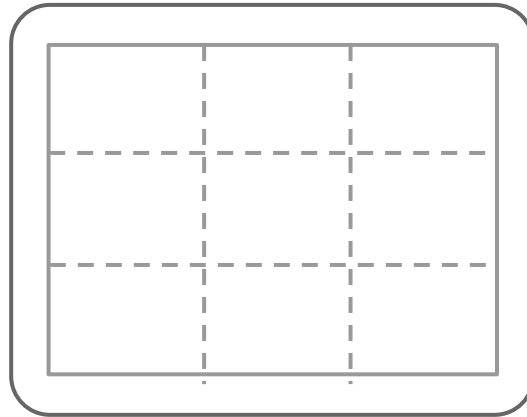
access element of object's elements,
of list element

```
list[[elem]][[obj]][ind]
```

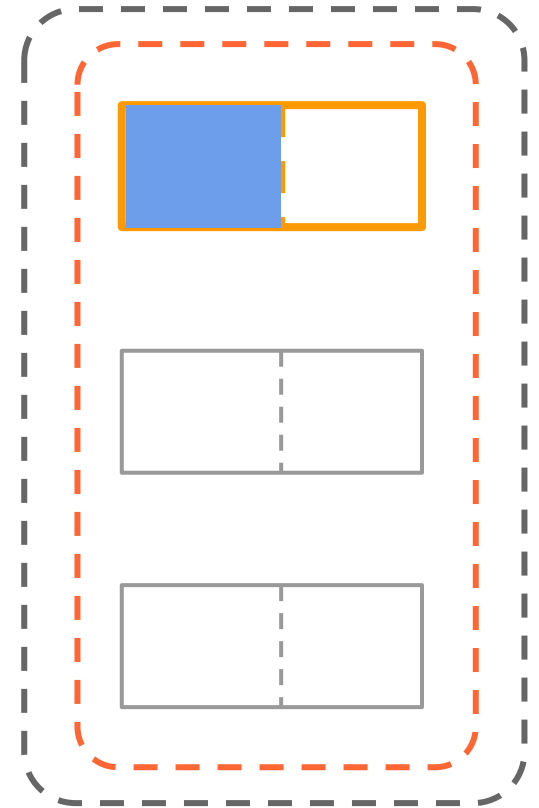
1st[[3]][[1]][1]



vector

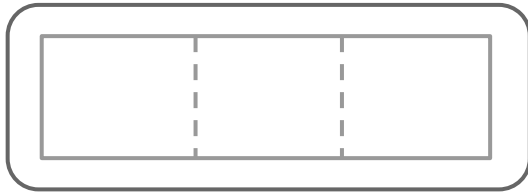


matrix

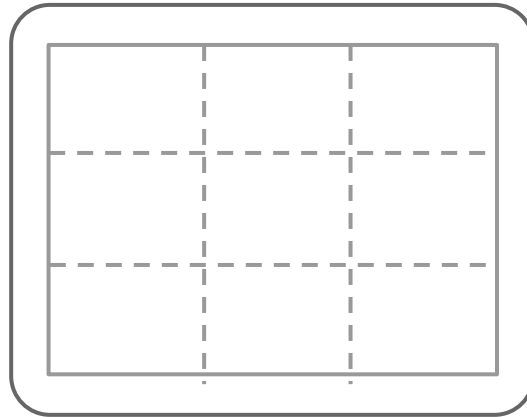


another list

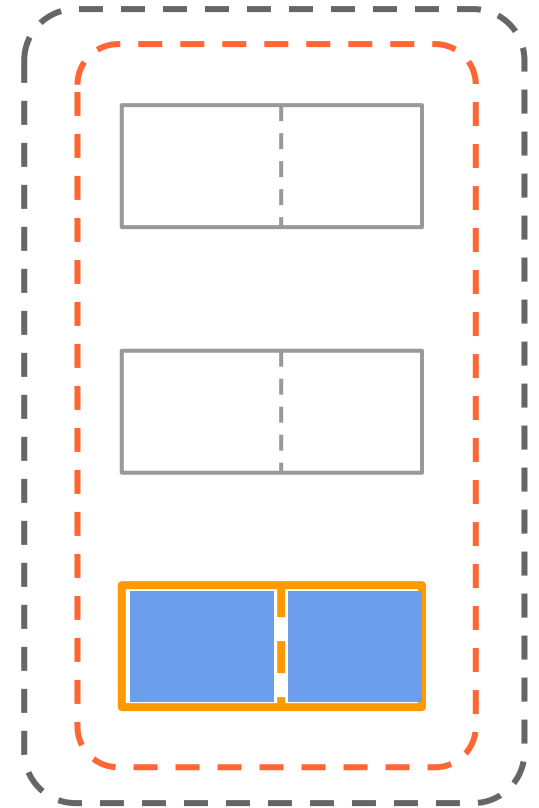
1st[[3]][[3]][c(1,2)]



vector



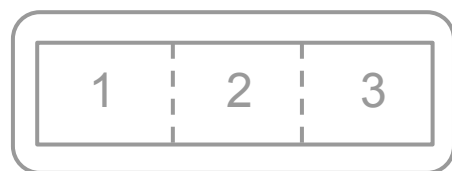
matrix



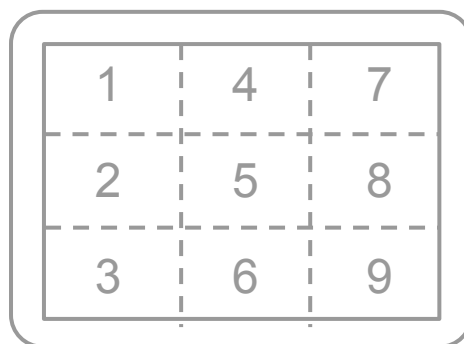
another list

Dollar Notation

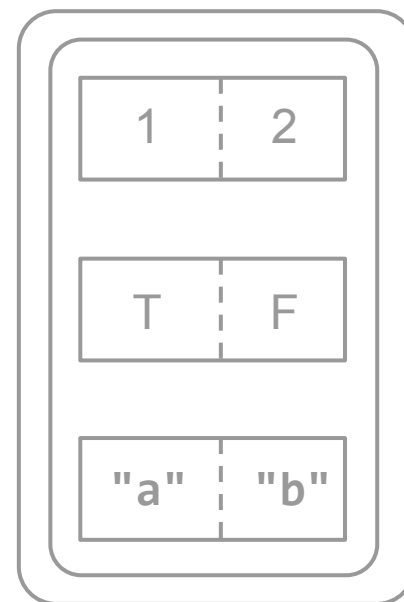
```
lst <- list(  
  vec = c(1, 2, 3),  
  mat = matrix(1:9, nrow = 3, ncol = 3),  
  lis = list(1:2, c(TRUE, FALSE), c("a", "b"))  
)
```



"vec"



"mat"

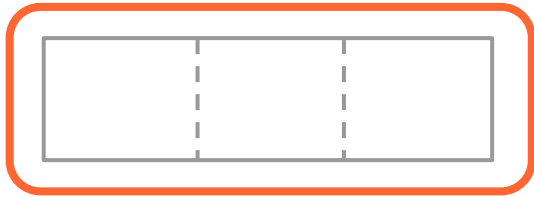


"lis"

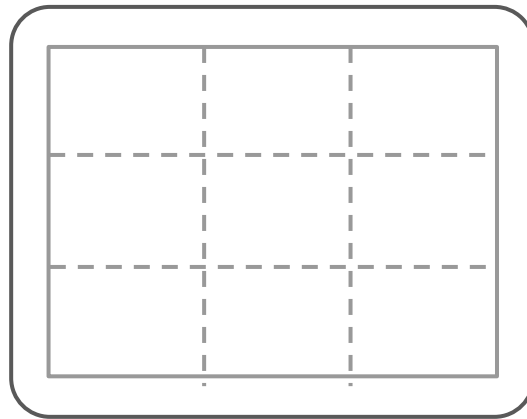
access list named element(s)

`list$name`

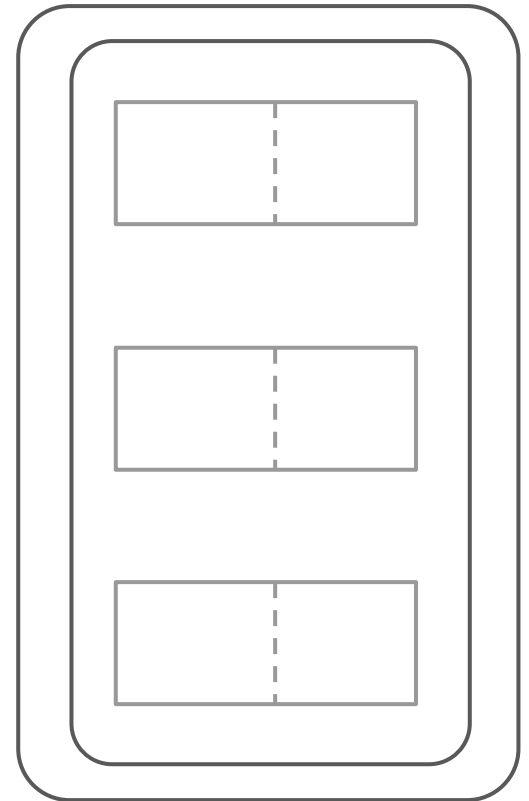
1st\$vec



"vec"

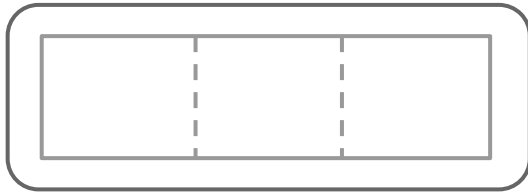


"mat"

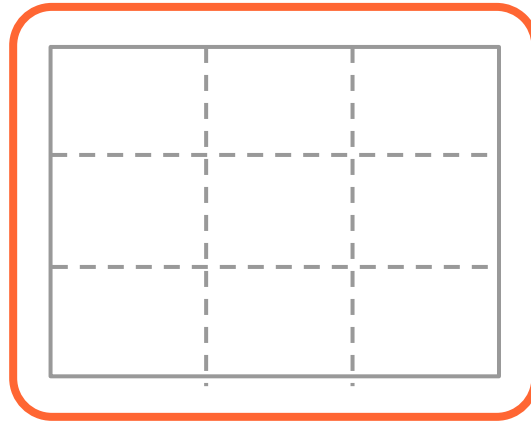


"lis"

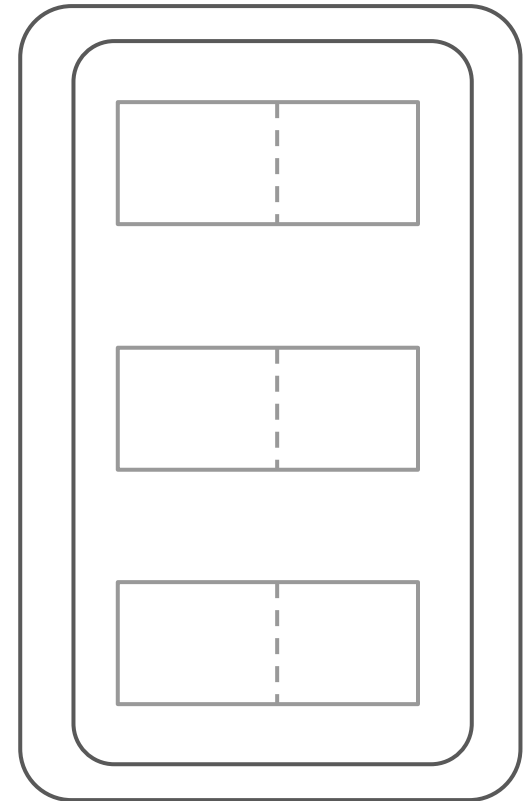
1st\$mat



"vec"

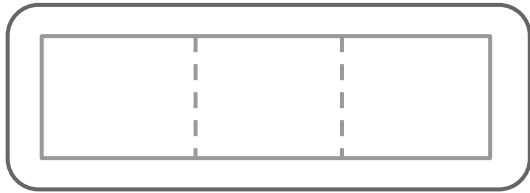


"mat"

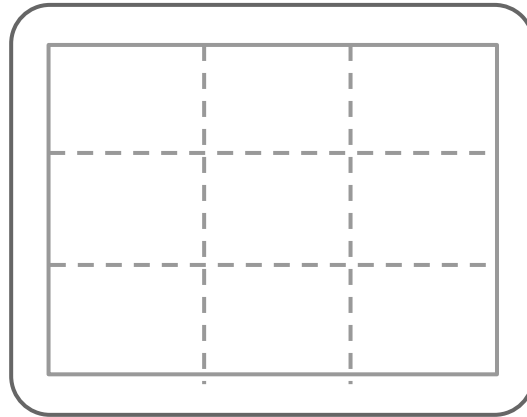


"lis"

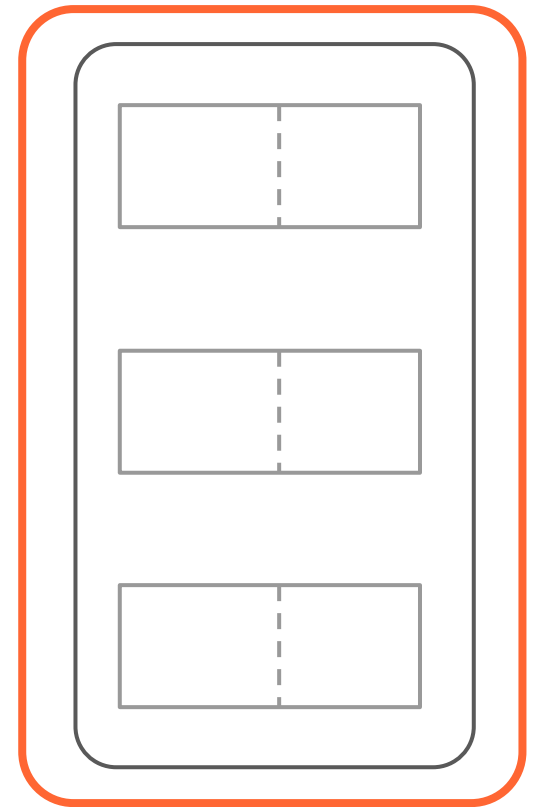
1st\$lis



"vec"



"mat"



"lis"

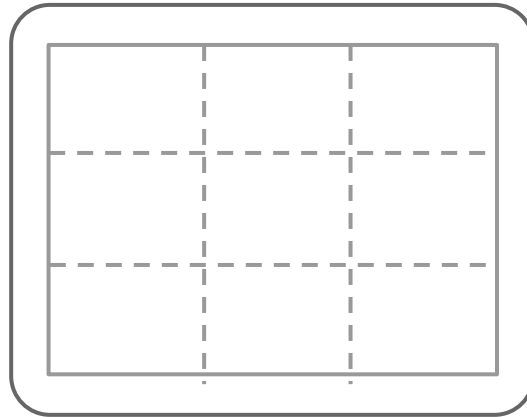
access list named element(s)

`list$name[ind]`

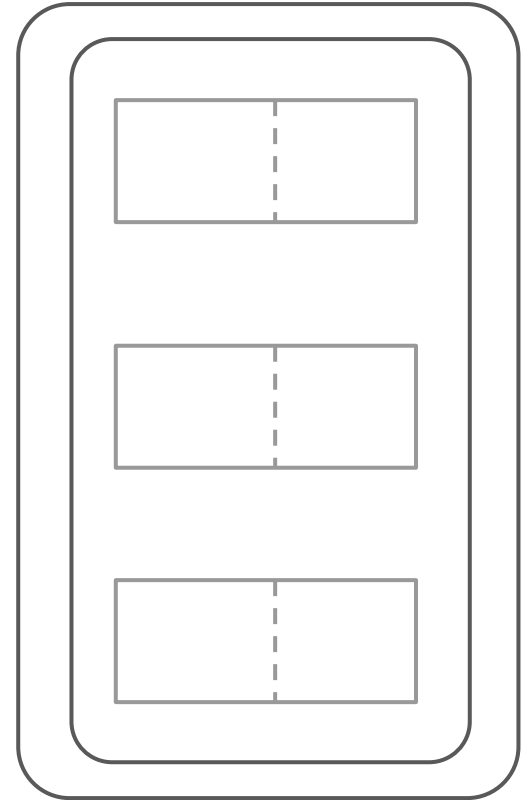
1st\$vec[2]



"vec"

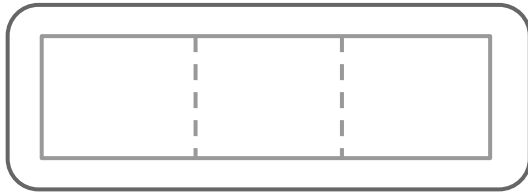


"mat"

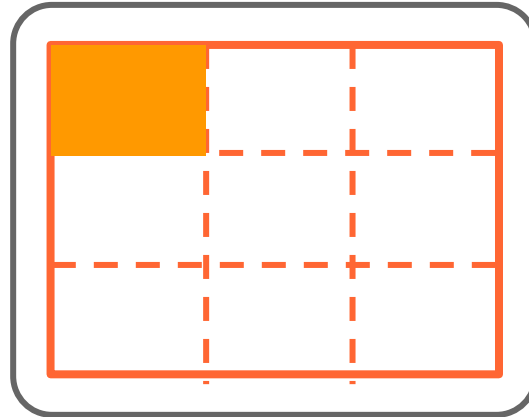


"lis"

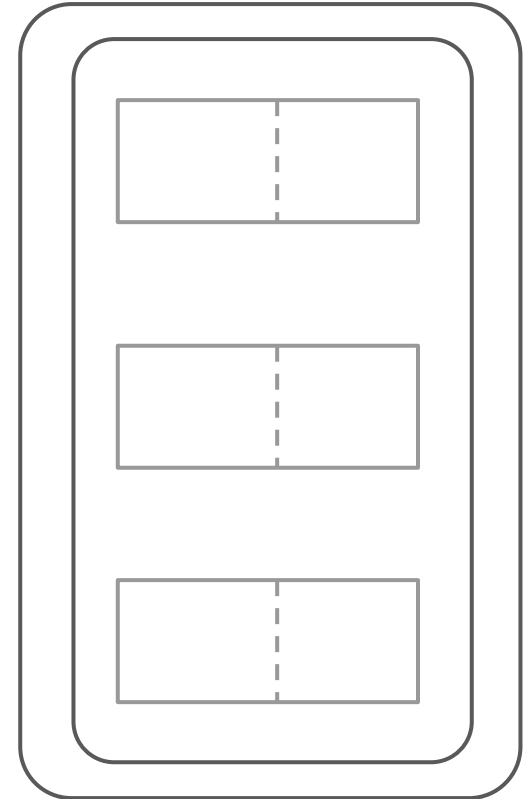
1st\$mat[1,1]



"vec"

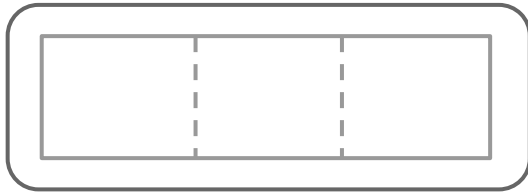


"mat"

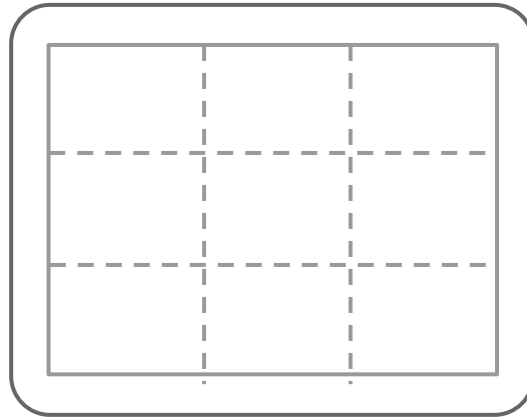


"lis"

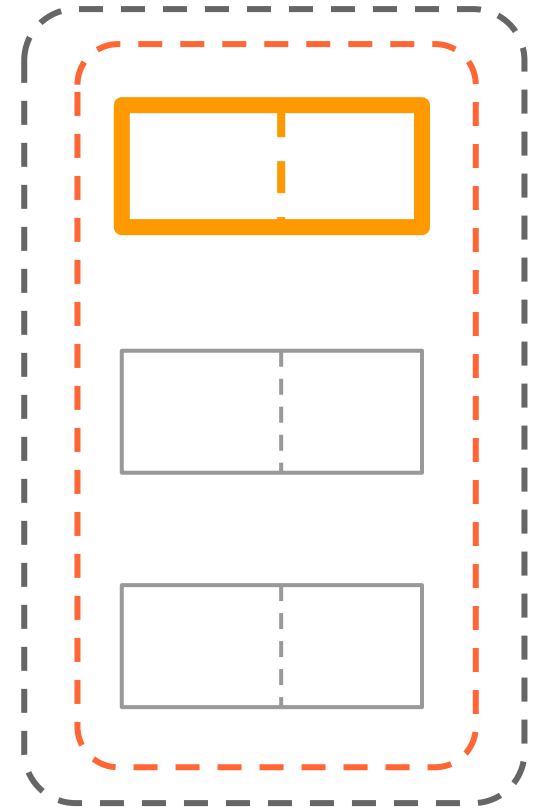
1st\$lis[[1]]



"vec"



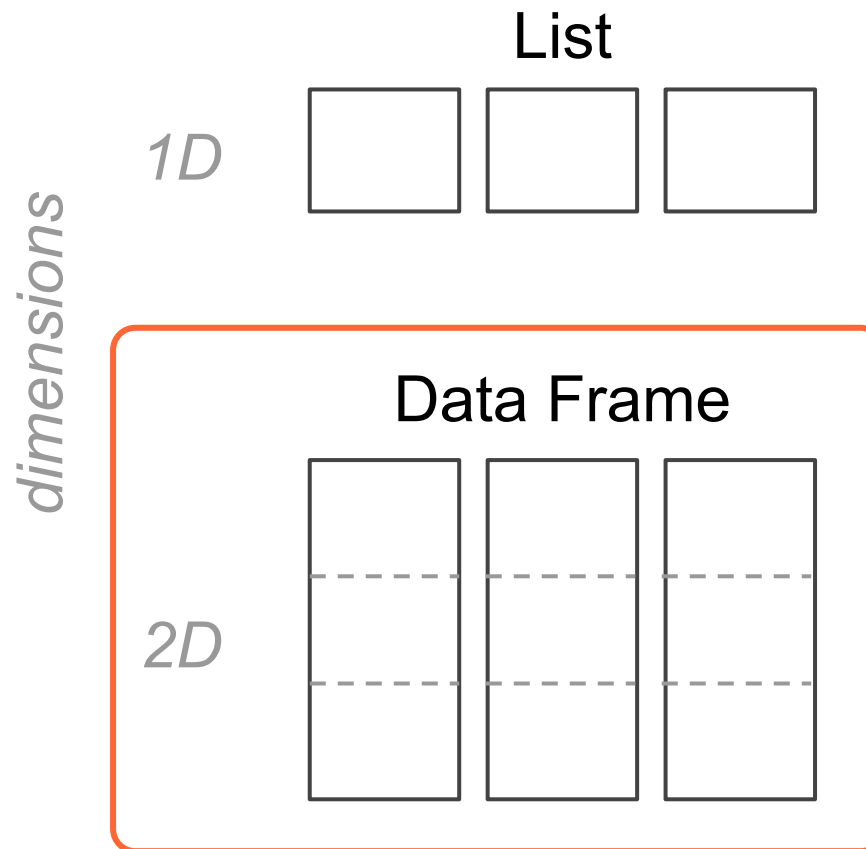
"mat"



"lis"

Special list: Data Frames

multiple data types



We'll talk about data frames later