R
Review Session

Paleobiology

February 22, 2016

The R WorkFlow

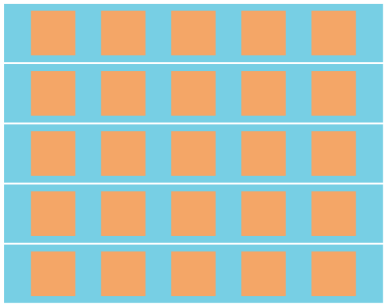
- Step 1: What data do you currently have?
- Step 2: What data do you want to have?
- Step 3: What function do you use to get the data you want?
- Step 4: How do you get the data you have into that function?

The R Workflow

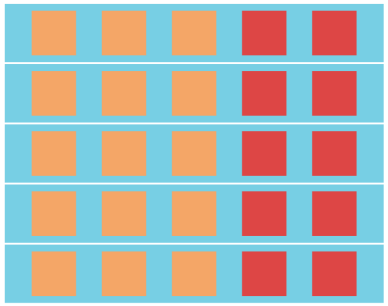
What shape is the data in?

Vector

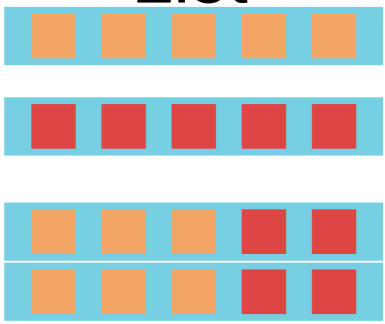
Matrix



Data Frame



List

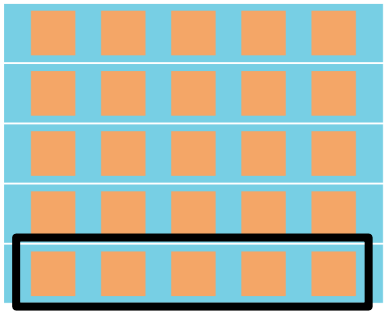


Referencing data from different shapes

Vector

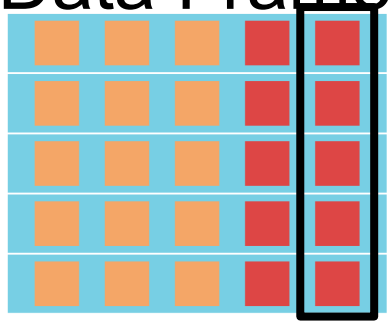
Vector[5]

Matrix



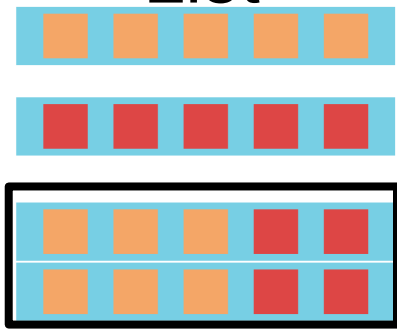
Matrix[5,]

Data Frame



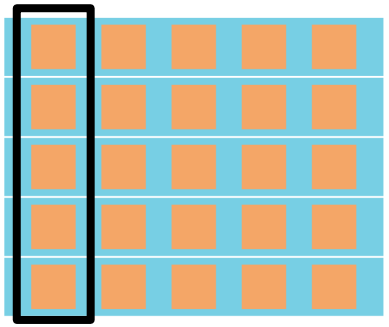
Frame[,5]

List

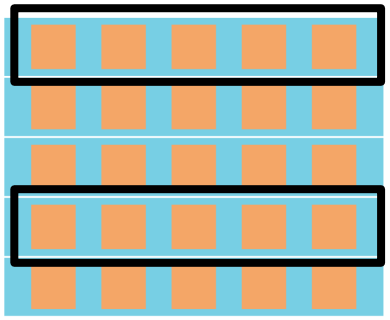


List**[[3]]**

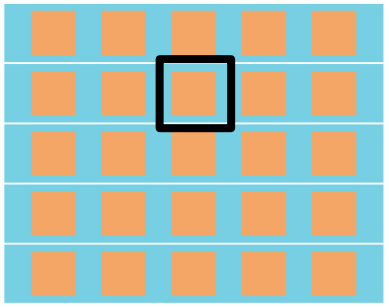
How do I reference these elements



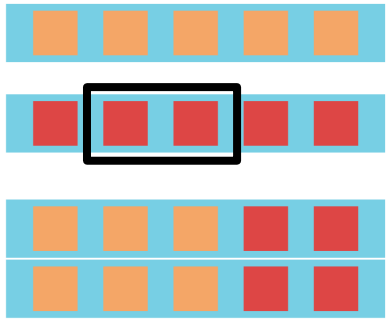
Matrix[,1]



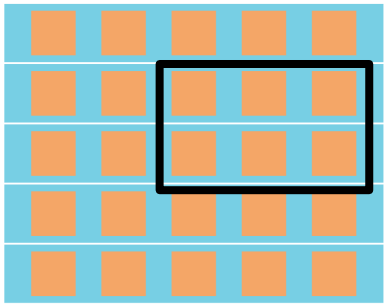
Matrix[c(1,4),]



Matrix[2,3]

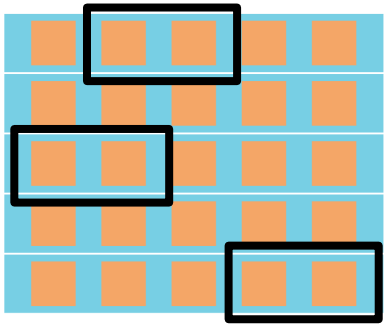


(List[[2]])[c(2,3)]



Matrix[2:3,3:5]

Matrix[c(2,3),c(3,4,5)]



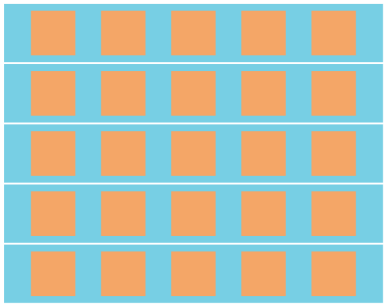
Don’t subscript, use which or subset

What size is the data?

Vector

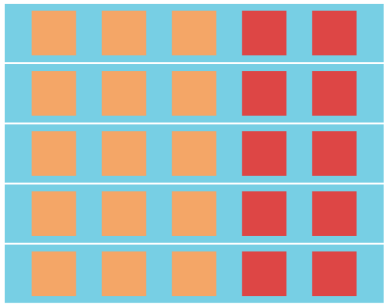
length(Vec)

Matrix



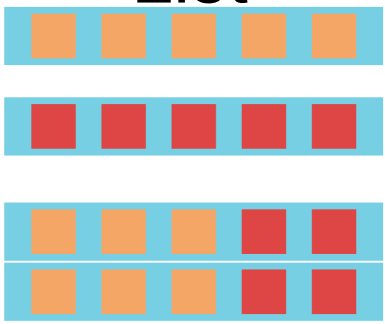
dim(Matrix)

Data Frame



dim(Frame)

List



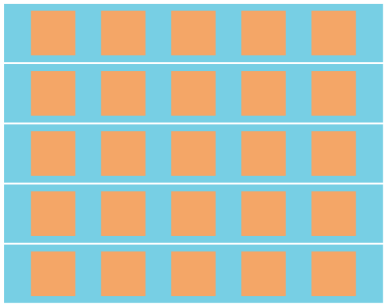
length(List)

Other information in your data

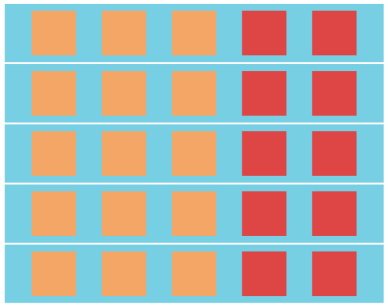
Vector

names(Vec)

Matrix



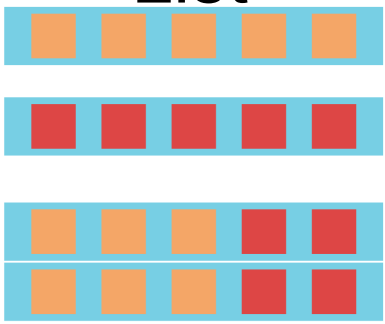
Data Frame



dimnames(Matrix)

dimnames(Frame)

List



names(List)

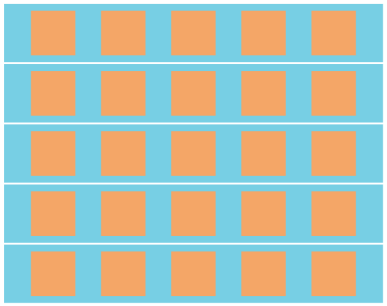
Other information in your data

Vector

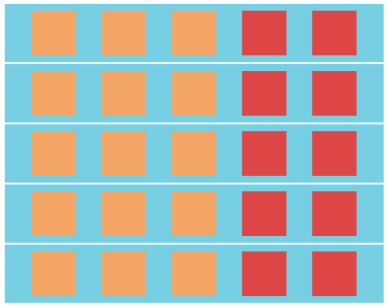
names(Vec)

Output is a ***vector*** of names

Matrix



Data Frame

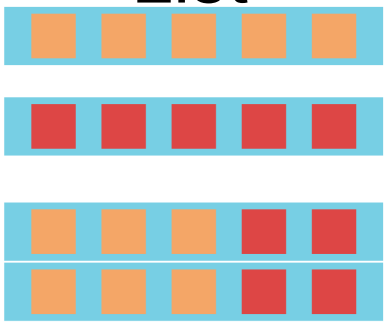


dimnames(Matrix)

dimnames(Frame)

Output is ***list*** of names

List

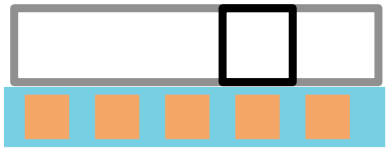


names(List)

Output is a ***vector*** of names

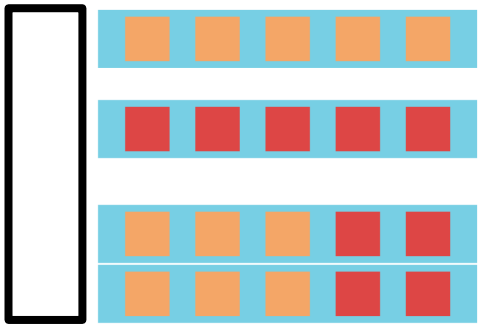
Other information in your data

names(Vec)

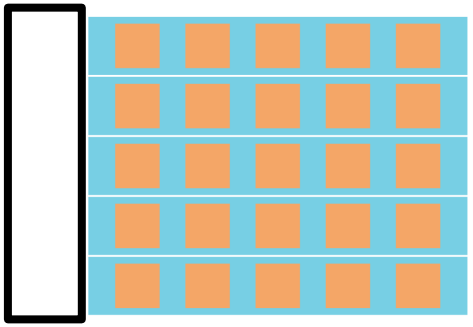


names(Vec)[4]

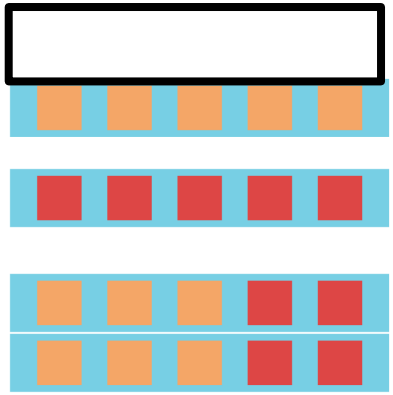
names(Vec[4])



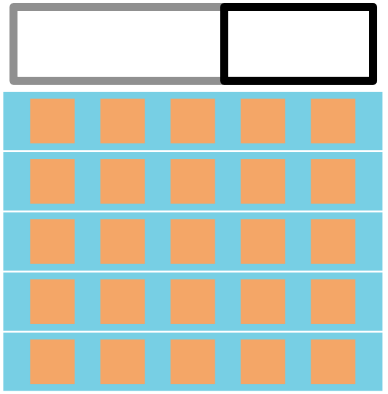
names(List)



Dimnames(Matrix)[[1]]



names(List[[1]])



Dimnames(Matrix)[[2]][c(3,4)]

Dimnames(Matrix)[[2]][3:4]

What I am getting at

- Every time you perform an action in R, a new object is created (but not necessarily saved)
- This new object, can also be subscripted.
- Therefore, you want to understand the shape of ***everything*** that you produce in R

What I am getting at

- Every time you perform an action in R, a new object is created (but not necessarily saved)
- This new object, can also be subscripted.
- Therefore, you want to understand the shape of ***everything*** that you produce in R

A simple example.

```
> c(1, 2, 3, 4, 5)^2
[1] 1  4  9  16  25
```

What I am getting at

- Every time you perform an action in R, a new object is created (but not necessarily saved)
- This new object, can also be subscripted.
- Therefore, you want to understand the shape of ***everything*** that you produce in R

A simple example.

```
> c(1, 2, 3, 4, 5)^2
[1] 1  4  9  16  25
```

A simple example.

```
> (c(1, 2, 3, 4, 5)^2) [5]
[1] 25
```

Parting tips

- Always check the data shape and size before you do anything else
- Try subscripting before you reach for something more advanced like subset( ) or which( ).
- Remember your answers can also be subscripted.
- Different functions take different shaped data. 9 times out of 10, if your function is giving you an error, it is because you gave it the wrong shape.

Parting tips

- R reads words without “ ” as objects. You need to use “ ” whenever you are not referencing an object.

```
Object<-5
Object                    “Object”
[1] 5                     [1] “Object”
```

- You can add/subtract/multiply/divide two objects of identical size and shape

```
c(1,2,3,4,5) + c(1,2,3,4,5)
[1] 2, 4, 6, 8, 10
```

  - In other words, you may not need to do any subscripting at all.