

R Data Structure

Young W. Lim

2018-02-12 Mon

1 Introduction

- References
- Data Structures
- Vectors
- Data Frames
- Lists
- Matrices
- Arrays
- Factors and Levels

"R for Everyone - Advanced Analytics and Graphic" J. P. Lander

I, the copyright holder of this work, hereby publish it under the following licenses: GNU head Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled GNU Free Documentation License.

CC BY SA This file is licensed under the Creative Commons Attribution ShareAlike 3.0 Unported License. In short: you are free to share and make derivative works of the file under the conditions that you appropriately attribute it, and that you distribute it only under a license compatible with this one.

- Vectors
- data.frames
- Lists
- Matrices
- Arrays

Vectors, Arrays, Lists, and Data Frames

Vectors	a collection of elements of the same type no mixed type is allowed different from mathematical row / col vectors
Lists	hold arbitray objects of either the same type or varying type recursive inclusion
Matrices	2-dimensional array with rows and colums of the same type (different from data.frame) no mixed typeis allowd
Arrays	a multidimensional vector must be of the same type
Data Frames	just like Excel spreadsheet each column is a vector, each has the same length each element in a column must be of the same type

Making Vectors, Arrays, Lists, and Data Frames

Vectors	<code>c(1, 2, 3)</code>
Lists	<code>list(1, 2, c(3, 4, 5))</code>
Matrices	<code>matrix(1:10, nrow=5)</code>
Arrays	<code>array(1:12, dim=c(2,3,2))</code>
Data Frames	<code>data.frame(3:1, -1:1, c("A0", "A1", "A2"))</code>

Results (1)

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

[[2]]
[1] 2

[[3]]
[1] 3 4 5

> matrix(1:10, nrow=5)
      [,1] [,2]
[1,]    1    6
[2,]    2    7
[3,]    3    8
[4,]    4    9
[5,]    5   10
```

Results (2)

```
> array(1:12, dim=c(2,3,2))
, , 1

      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

, , 2

      [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12

> data.frame(3:1, -1:1, c("A0", "A1", "A2"))
  X3.1 X.1.1 c..A0....A1....A2..
1     3   -1                    A0
2     2    0                    A1
3     1    1                    A2
>
```


Vector Operations (1)

```
x <- c(1, 2, 3, 4, 5)
x
x * 3
x + 2
x - 3
x / 4
x ^ 2
sqrt(x)
1:10
10:1
-2:3
5:-7
```

Vector Operations (2)

```
x <- 1:10  
y <- -5:4  
x + y  
x - y  
x * y  
x / y  
x ^ y  
length(x)  
length(y)  
length(x+y)
```

Vector Operations (3)

```
x + c(1, 2)
x + c(1, 2, 3)
x <= 5
x > y
x < y
x <- 10:1
y <- -4:5
any(x<y)
all(x>y)
```

Vector Operations (4)

```
q <- c("AAA", "BBBB", "CCCCC")
nchar(q)
nchar(y)
x[1]
x[1:2]
x[c(1,4)]
c(One="a", Two="y", Three="z")
w <- 1:3
names(w) <- c("a", "b", "c")
w
```

```
q2 <- c(q, "AAA", "BBBB" , "DDDD" )
q2Factor <- as.factor(q2)
q2Factor
as.numeric(q2Factor)
factor(x=c("High School", "College", "Masters", "Doctorate"),
      levels=c("High School", "College", "Masters", "Doctorate"),
      ordered=TRUE)
```

data.frame (1)

```
x <- 10:1
y <- -4:5
q <- c("A1", "A2", "A3", "A4", "A5",
      "A6", "A7", "A8", "A9", "A10")
DF <- data.frame(x, y, q)

DF <- data.frame(First=x, Second=y, Sport=q)
nrow(DF)
ncol(DF)
dim(DF)
names(DF)
names(DF)[3]
```

data.frame (2)

```
rownames(DF) <- c("B1", "B2", "B3", "B4", "B5",  
                 "B6", "B7", "B8", "B9", "B10")  
rownames(DF)  
rownames(DF) <- NULL  
rownames(DF)  
head(DF)  
head(DF, n=7)  
tail(DF)  
class(DF)
```

data.frame (3)

```
DF$Sport
DF[3, 2]
DF[3, 2:3]
DF[c(3,5), 2]
DF[C(3,5), 2:3]
DF[, 3]
DF[, 2:3]
DF[2, ]
DF[2:4, ]
```


data.frame (4)

```
DF[, c("First", "Sport")]  
DF[, "Sport"]  
class(DF[, "Sport"])  
DF["Sport"]  
class(DF["Sport"])  
DF[["Sport"]]  
class(DF[["Sport"]])
```

data.frame (5)

```
DF[, "Sport", drop=FALSE]
class(DF[, "Sport", drop=FALSE])
DF[, 3, drop=FALSE]
class(DF[, 3, drop=FALSE])
NF <- factor(c("N1", "N2", "N3", "N4",
               "N5", "N6", "N7", "N8"))
model.matrix(NF-1)
attr("assign")
attr("contrasts")
attr("contrasts")$NF
```

Lists (1)

```
list(1, 2, 3)
list(c(1, 2, 3))
(list3 <- list(c(1,2,3), 3:7))
list(DF, 1:10)
list5 <- list(DF, 1:10, list3)
list5
```

Lists (2)

```
names(list5)
names(list5) <- c("data.frame", "vectors", "list")
names(list5)
list5
$vector
$list
$list[[1]]
```

Lists (3)

```
list6 <- list(DFrame=DT, Vector=1:10, List=list3)
names(list6)
list6
$Vector
$List
$List[[1]]
$List[[2]]
```

Lists (4)

```
(emptyList <- vector(mode="list", length=4))  
list5[[1]]  
list[["data.frame"]]  
list5[[1]]$Sport  
list5[[1]][, "Second"]  
list5[[1]][, "Second", drop=FALSE]
```

Lists (5)

```
length(list5)
list5[[4]] <- 2
length(list5)
list5["NewElement"] <- 3:6
length(list5)
names(list5)
```

Lists (6)

```
list5
$data.frame
$vector
$list
$list[[1]]
$list[[2]]
$NewElement
```


Matrices (1)

```
A <- matrix(1:10, nrow=5)
B <- matrix(21:30, nrow=5)
C <- matrix(21:40, nrow=2)
```

```
A
```

```
B
```

```
C
```

```
nrow(A)
```

```
ncol(A)
```

```
dim(A)
```

```
A+B
```

```
A*B
```

```
A == B
```

```
A %*% t(B)
```

Matrices (2)

```
colnames(A)
rownames(A)
colnames(A) <- c("Left", "Right")
rownames(A) <- c("1", "2", "3", "4", "5")
colnames(B)
rownames(B)
colnames(B) <- c("First", "Second")
rownames(B) <- c("1", "2", "3", "4", "5")
colnames(C)
rownames(C)
colnames(C) <- LETTERS[1:10]
rownames(C) <- c("Top", "Bottom")
```

Arrays

```
A <- array(1:12, dim=c(2, 3, 2))  
A  
A[1, , ]  
A[1, , 1]  
A[ , , 1]
```

Definitions of Factors and Levels

- A **factor** is a categorical variable
 - that can take only one of a fixed, finite set of possibilities.
 - these possible categories are the **levels**
- factors in R are stored as a vector of integer values
 - with a corresponding set of character values (levels)
 - to use when the factor is displayed
- both numeric and character variables can be made into factors
- a factor's levels will always be character values.

<https://www.stat.berkeley.edu/classes/s133/factors.html>

<https://stackoverflow.com/questions/20314318/what-are-r-levels>

Functions: factor and level

- `factor(vector)` returns a vector of factor values
 - `x <- factor(c("male", "female", "female", "male"))`
 - 1 is assigned to the level "female"
 - 2 is assigned to the level "male"
 - female < male in lexicographical order
- `level` shows the possible levels for a factor
 - `levels(x)`
 - `[1] "female" "male"`
 - `nlevels(x)`
 - `[1] 2`

<https://www.stat.berkeley.edu/classes/s133/factors.html>

<http://monashbioinformaticsplatform.github.io/>

2015-09-28-rbioinformatics-intro-r/01-supp-factors.html

- to change the default sorted order of levels
use the `levels=` with a vector of all the possible values
of the variable in the desired order
- to keep the ordering in comparison,
use the optional `ordered=TRUE` argument
an ordered factor

<https://www.stat.berkeley.edu/classes/s133/factors.html>

Changing levels of a factor

- the levels of a factor are used when displaying the factor's values
- can change these levels when creating a factor
by `rdata = factor(data, labels=c("I", "II", "III"))`
where `data = c(1,2,2,3,1,2,3,3,1,2,3,3,1)`
- note that this actually changes the internal levels of the factor
- to change the labels of a factor after it has been created
use `levels(fdata) = c('I', 'II', 'III')`

<https://www.stat.berkeley.edu/classes/s133/factors.html>

Factor example codes

```
> data = c(1,2,2,3,1,2,3,3,1,2,3,3,1)
> fdata = factor(data)
> fdata
[ 1] 1 2 2 3 1 2 3 3 1 2 3 3 1
Levels: 1 2 3
> rdata = factor(data,labels=c("I","II","III"))
> rdata
[ 1] I  II  II  III I  II  III III I  II  III III I
Levels: I II III

> levels(fdata) = c('I','II','III')
> fdata
[1] I  II  II  III I  II  III III I  II  III III I
Levels: I II III
```

<https://www.stat.berkeley.edu/classes/s133/factors.html>