4. Even More Data structures in R

Principles of Data Science with R

Dr. Uma Ravat

PSTAT 10

Announcement

1. Quiz 1 will be next Monday (Week 3 Monday) 30 minutes between noon and 11:59pm. (No make ups)

   - Material will include everything till Week 2.

2. HW 3 will be out this evening.

3. Use office hours (TA, ULA) to review material you are having difficulty with.

4. **Email Policy:** : Check canvas Week 0 for detailed email policy.

   - Emails must be formatted correctly for fastest response.
   - Your TA is first person to contact and will reach out to HeadTA and instructor as needed.

Summary:

More data structures

- matrices and arrays. (Textbook Ch3)
- Factors (Textbook Chapter 4)
- Logical datatye and operator (Textbook Chapter 4)
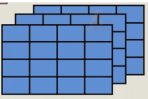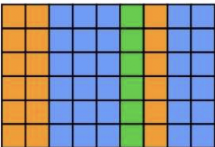
Maintain a glossary of functions used.

- Even more Data structures
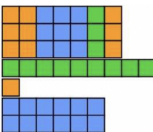   - list
    - data frame
    - Working with strings

	1 D	2 D	Multi-D
Homogeneous	Vector	Matrix	Array
Heterogeneous	List	Dataframe	

- allows for different data types to be included
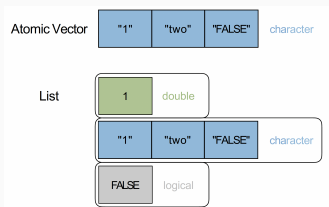
- allows for different data structures (even lists!) to be included
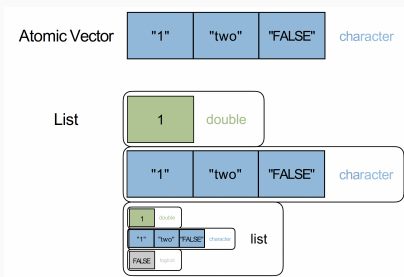
List


```r
lst <- list(num = c(8,9), log = TRUE, cha = c("a", "b", "c"))
```


← **Tweet**

One R Tip a Day
@RLangTip

•••

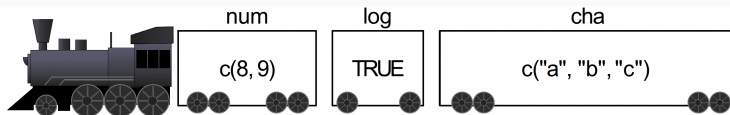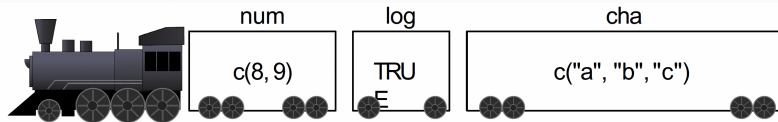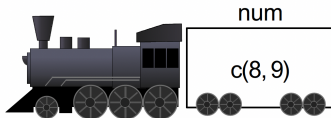If the #rstats list "x" is a train carrying objects, then x[[5]] is the object in car 5; x[4:6] is a train of cars 4-6.
bit.ly/p6epvV

7:32 AM · Nov 13, 2012 · TweetDeck

| | num | log | cha |
| | c(8, 9) | TRUE | c("a", "b", "c") |

| num |
| c(8, 9) |

`lst["num"]`

`lst[["num"]]` $c(8, 9)$

`lst$num` $c(8, 9)$

What we did

- create using list() function
- Access list objects, item/member
    - [[ ]] or $
        - pull out a single object of the list using index or name.
        - can't be used for extracting multiple objects
        - for list x: x[[y]] or x$y
    - [ ]
        - returns members/items in the list
        - returns a list
        - can be used for extracting multiple members in the list
        - mylist[2:4]
- Assigning new values to list objects
- working with lists

Data frames

- natural way of representing a data set.
    - data is in rows(observations of the data set) and columns (variables of the data set)
    - each column is a vector (could be a factor for categorical data)
    - all columns are of same length
    - each column can be of a different type

Vector	**Matrix**	**Data Frame**

- 1 column or row of data
- 1 type (numeric or text)
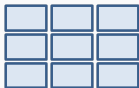
- multiple columns and/or rows of data
- 1 type (numeric or text)

- multiple columns and/or rows of data
- multiple types

Creating, extracting, assigning

- data.frame()
- [,] , $
- <-
- other functions

Special values in R

- Inf : Infinity

```
100/0
```

```
## [1] Inf
```

- NaN : Not a number

```
Inf - Inf
```

```
## [1] NaN
```

- NA : Not applicable (is.na(), na.rm())

```r
a_vec <- c("a", 1, NA, 2, "cat", NA, 100)
a_vec
```

```
## [1] "a"   "1"   NA    "2"   "cat" NA    "100"
```

```r
is.na(a_vec)
```

```
## [1] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
```

```
any(is.na(a_vec))
```

```
## [1] TRUE
```

```
any(a_vec == 2 )
```

```
## [1] TRUE
```

```
which(a_vec == 2)
```

```
## [1] 4
```

```r
all(is.na(a_vec ))
```

```
## [1] FALSE
```

```r
b <- seq(-5 , 5, 1)
b
```

```
##  [1] -5 -4 -3 -2 -1  0  1  2  3  4  5
```

```r
all(b > 2)
```

```
## [1] FALSE
```

```r
any(b > 2)
```

```
## [1] TRUE
```

```
which(b > 2 )
```

```
## [1]  9 10 11
```

```
b[b>2]
```

```
## [1] 3 4 5
```

- NULL : an empty entity

```
x <- c()
x
```

```
## NULL
```

```
is.null(x)
```

```
## [1] TRUE
```

Strings

```
my_string <- c("This is a string")
my_string

## [1] "This is a string"

nchar(my_string)

## [1] 16

substr(my_string, start = 3, stop = 9)

## [1] "is is a"
```

```r
sub(pattern="is",replacement="was",x=my_string)
```

```
## [1] "Thwas is a string"
```

```r
gsub(pattern="is",replacement="was",x=my_string)
```

```
## [1] "Thwas was a string"
```

Predefined Constants in R

```
letters
```

```
## [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
```

```
LETTERS
```

```
## [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S"
## [20] "T" "U" "V" "W" "X" "Y" "Z"
```

```
month.abb
```

```
## [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"
```

```
month.name
```

```
## [1] "January"   "February"  "March"     "April"     "May"       "June"
## [7] "July"      "August"    "September" "October"   "November"  "December"
```

questions you should be able to answer

- What are the different data types in R?
- What are the different data structures in R?
    - Which data structures require elements to be of same data type and which allow for elements of different data types?
- How do I create, access(retrieve, subset), update data within the various data structures?
- What functions are available to work with the different data structures?

Post-Lecture To DO

1. Review the lecture again
2. Write down a summary of today's lecture. Include all functions we went over and a short description of what each function does.

You will be asked to do this to your homework.

Summary:

- Factors (Textbook Chapter 4)
- Logical values (Textbook Chapter 4)

Even More data structures

- Lists and Data frames (Textbook Chapter 5)
- Special values (Textbook Chapter 6)
- strings (Non-numeric values Chapter 4)

Maintain a glossary of functions used.

Learning Programming is HARD!

E. Kale Edmiston PhD
@EKaleEdmiston

Follow

A friend/colleague who is an excellent programmer offhandedly told me the other day that coding is 90% googling error messages & 10% writing code. Until this point, I thought that all the time I spent googling error messages meant I was bad at coding. What a perspective change!

8:12 AM - 4 Jan 2019

151 Retweets **1,069** Likes

27   151   1.1K