FALL 2018

TEXAS
The University of Texas at Austin

APPLIED ENGINEERING DATA ANALYSIS, OPTIMIZATION AND VISUALIZATION

Data structures and other exciting things

JOSHUA RHODES, PHD
Research Fellow/Adjunct Professor, The University of Texas at Austin

Homework 1 has been assigned and is on Canvas.

Note that a grading rubric is on Canvas too.

Now that we have got into the weeds, lets step back and go over some important R concepts.

This lecture will go over the important data structures in R*

	Homogeneous	Heterogeneous
1d	Atomic vector	List
2d	Matrix	Data frame
nd	Array	

* http://adv-r.had.co.nz/Data-structures.html

Something that is a bit different about R vs. other languages

- R has no 0-dimensional, or scalar types. Individual numbers or strings, which you might think would be scalars, are actually vectors of length one.
- The function str() gives you the structure of an object

Atomic vectors

- Usually created with c(), short for combine
- There are 4 main types
  - Integer
  - Logical
  - Double (often called numeric)
  - Character

Integer vectors save space

- Integers are 4 bytes in size
- ½ the size of doubles
- Denoted with a “L” designation

Doubles are double the size, double the fun

- Integers are 8 bytes in size
- Can include decimals
- Are more flexible than integers

Logical vectors hold the truth

- Logicals are the least flexible
  - Either true or false, no gray areas with computers
- Can be coerced into numerics
  - TRUE = 1, FALSE = 0

Character vectors are the most versatile

- If all else fails, R goes with characters
- Anything can be a character
  - Just like you

Lists

- Lists are different from atomic vectors because their elements can be of any type, including lists.
- You construct lists by using list() instead of c()
- Lists can contain other lists as elements

Matrices and arrays

- An **array** is a multidimensional vector
- A special case of the array is the **matrix**, which has two dimensions

Matrix =

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

y

x

Array =

z

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

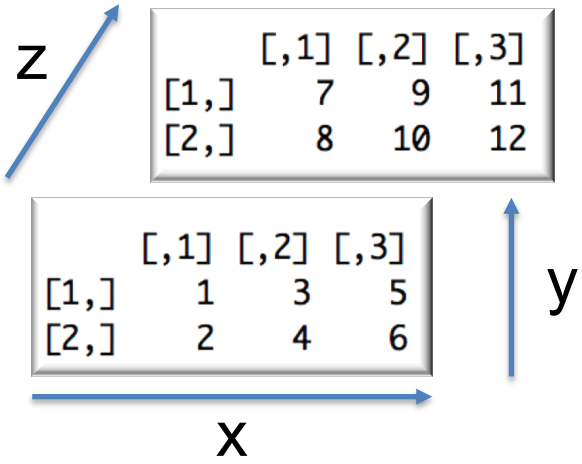
```
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

y

x

An array is like a matrix with more dimensions

- Here, d = 3
- Could be more…



Data frames

- A data frame is a list of equal-length vectors
- 2-dimensional structure, so it shares properties of both the matrix and the list
- Columns can be different types
- Most common data analysis structure

Let's talk about factors

- Can contain only predefined values, and is used to store categorical data
- Factors are built on top of integer vectors using two attributes: the class, "factor", and the levels, which defines the set of allowed values.