Tutorial 1
Introduction to R
Statistical Methods in Research
COSC 6323
Spring 2018

Ioannis Pavlidis
Dinesh Majeti
George Panagopoulos

Computational Physiology Lab

ipavlidis@uh.edu
dmajeti@uh.edu
gpanagopoulos@uh.edu

January 22, 2018

Introduction to R

Introduction to R

1. Basics
2. Workspace and Files
3. Data Types
4. Vectors
5. Sequences
6. Missing Values
7. Matrices
8. Data Frames
9. Basic Statistics

Basics

- Variable assignment

```
> a = 1
> a
[1] 1
```

- Functions - invoked by name

```
> c(1, 2, 3)
[1] 1 2 3
```

- Install packages

```
> install.packages()
```

- Comments

```
> 1 + 1       # this is a comment
[1] 2
```

Basics

- Create vector

```
> z = c(1:10)
```

- Read from keyboard

```
y = scan()
1: 10
2: 11
3: 12
y
[1] 1 2 3
```

- help

```
> ?c
> help(c)
```

- Case sensitive

Workspace and Files

```
olddir = getwd()
dir.create("testdir")
list.files()
setwd("testdir")
setwd(olddir)
unlink("testdir")
```

If recursive = FALSE directories are not deleted, not even empty ones.

```
unlink("testdir", recursive = TRUE)
dir.create("testdir2/testdir3", recursive = TRUE)
unlink("testdir2", recursive = TRUE)
```

Basic Data Types - Numeric

- Numeric
- Integer
- Complex
- Logical
- Character

Basic Data Types - Numeric

- Numeric
  - decimal values are called numerics in R.
  - default computational data type

```
> x = 10.5          # assign a decimal value
> x                 # print the value of x
[1] 10.5
> class(x)          # print the class name of x
[1] "numeric"

> k = 1
> k                 # print the value of k
[1] 1
> class(k)          # print the class name of k
[1] "numeric"

> is.integer(k)  # is k an integer?
[1] FALSE
```

Basic Data Types - Integer

- Integer

```
> y = as.integer(3)
> is.integer(y)  # is y an integer?
[1] TRUE
> as.integer("5.27")  # coerce a decimal string
[1] 5

> as.integer("abc")   # coerce an non-decimal string
[1] NA
Warning message:
NAs introduced by coercion
```

Basic Data Types - Complex

- Complex

```
> z = 1 + 2i      # create a complex number

> sqrt(-1)        # square root of -1
[1] NaN
Warning message:
In sqrt(-1) : NaNs produced

> sqrt(-1+0i)     # square root of -1+0i
[1] 0+1i
# or use as.complex()
```

Basic Data Types - Logical

- Logical

```
> x = 4; y = 10; z = 6  # sample values
> comparison1 = x > y      # is x larger than y?
> comparison1             # print the logical value
[1] FALSE
> comparison2 = z > x      # is z larger than x?
> comparison1 & comparison2 # are both comparisons true?
[1] FALSE
> comparison1 | comparison2 # is at least one true?
[1] TRUE
> bool = T
> bool
[1] TRUE
```

Basic Data Types - Character

- Character - used to represent string values in R

```
> x = as.character(5.16)
> x                 # print the character string
[1] "5.16"
> class(x)          # print the class name of x

> fname = "John"; lname ="Doe"
> paste(fname, lname)
[1] "John Doe"
> paste0(fname, lname,2)
[1] "JohnDoe2"
> sprintf("%s has %d dollars", "Sam", 100)
[1] "Sam has 100 dollars"
```

Sequences

```
>s1 = seq(0, 10)
>seq(-3, 3, by=.2)  -> s2
>s3 = seq(length=10, from=-2, by=.2)
>s4 = rep(c(1,2), times=3)
>s4
[1] 1 2 1 2 1 2
>s5 = rep(c(1,2), each=3)
>s5
[1] 1 1 1 2 2 2
>sort(s4) == s5
[1] TRUE TRUE TRUE TRUE TRUE TRUE
```

Vectors

```
>assign("x", c(5.4, 3.6, 3.2, 1.4, 22.7))
>c(5.4, 3.6, 3.2, 1.4, 22.7) -> x
>x = c(5.4, 3.6, 3.2, 1.4, 22.7)

Logical vector
>temp = x > 13

>x = x * 2 + 100
>sqrt(x)
>y = c(1,2,3,4) + c(0,10)

>my_char = c("My", "name", "is")
>paste(my_char, collapse = " ")
```

Vectors - Indexes

```
> s = c("aa", "bb", "cc", "dd", "ee")

> s[c(2, 3, 3)]
[1] "bb" "cc" "cc"

> s[c(FALSE, TRUE, FALSE, TRUE, FALSE)]
[1] "bb" "dd"
```

Missing values

- NA

```
x = c(1:5,NA);
indices = is.na(x)
```

- NaN

```
0/0
Inf - Inf
is.na() for both NA and NaN
is.nan()
```

Matrices

- rectangular data type
- matrix - single class of data

```
>z = 1:20
>dim(z) = c(4,5)
>z
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
my_mat = matrix( c(1:20), nrow=4, ncol=5)

>identical(z, my_mat)
[1] TRUE
```

Data Frames

- for storing tabular data

```
patients = c("David", "Tina", "Joseph", "Ann")
cbind(patients, my_mat)

my_data = data.frame(patients, my_mat)

class(my_data)

cnames = c("patient", "age", "weight", "bp", "rating", "te
colnames(my_data) = cnames
```

List

- generic vector containing other objects

```
> n = c(2, 3, 5)
> s = c("aa", "bb", "cc", "dd", "ee")
> b = c(TRUE, FALSE, TRUE, FALSE, FALSE)
> x = list(n, s, b, 3)   # x contains copies of n, s, b

# list slicing - returns a list
> x[2]
[[1]]
[1] "aa" "bb" "cc" "dd" "ee"

# list member reference
> x[[2]]
[1] "aa" "bb" "cc" "dd" "ee"
```

List

- named members

```
> v = list(alan=c(1,2,3), james=c("ab", "cd"))
> v
$alan
[1] 1 2 3

$james
[1] "ab" "cd"
```


YO MAMA IS SO MEAN
SHE HAS NO STANDARD DEVIATION!
memegenerator.net

Basic Statistics

- basic functions

  `mean()`
  `sd()`
  `summary()`
  `boxplot()`

  `Linear regression`
  `lm()`

Boxplot

```
x=c(1,2,3,3,4,5,5,7,9,9,15)
boxplot(x, xlab="x label", ylab="y label")
title("My First Boxplot")
```

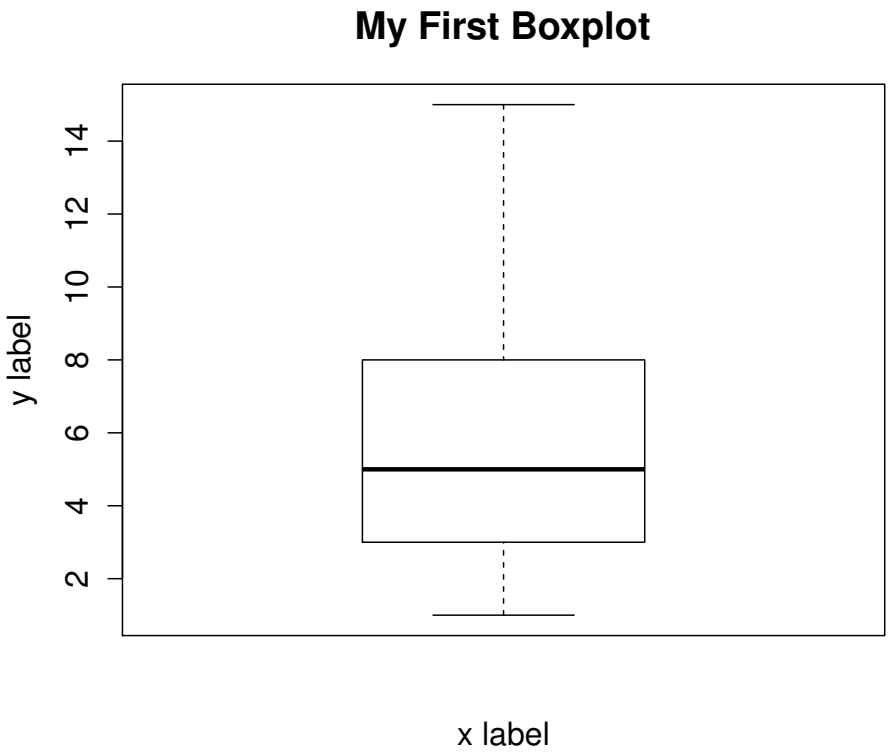

Figure 1: Sample boxplot in R

Puzzle

If you choose an answer to this question at random, what is the chance you will be correct?

A) 25%

B) 50%

C) 60%

D) 25%

For more on R

- https://www.r-bloggers.com/why-r-is-the-best-data-science-language-to-learn-today/
- http://www.r-project.org/
- http://swirlstats.com/students.html
- http://cran.r-project.org/doc/manuals/r-release/R-intro.html
- http://www.r-tutor.com/r-introduction
- R Studio
- http://www.rstudio.com/