

Rbootcamp

Michael Kleinsasser

8/8/2019

Mike's Personal Introduction

- R programmer for the Department of Biostatistics
- Write and maintain R packages for faculty and students
- Consult faculty and students on writing R packages



Rbootcamp Introduction

Goals:

- R basics: syntax, common functions, etc., via Rstudio
- R functions and for loops (functions and advanced control structures)
- Basics of R on the cluster (non-interactive R)
- Simulation analysis with comment on efficiency

Materials

- All bootcamp materials online at <https://github.com/umich-biostatistics/Rbootcamp>
- Handouts for each topic with examples to work through
- R scripts of our examples

R Basics: Goals

- Data types and functions
 - Create object having data types
 - Combine those into data structures
 - Write basic R function
- Learn parts of R most useful to statisticians
 - How do most modeling functions work in R, and
 - How to inspect structure and content of objects

R data types/structures

Five fundamental data types

- character, numeric, integer, logical, complex

Combine to form data structures

- atomic vector (atomic - vector of single type)
- list
- matrix
- data.frame
- factor

Assignment in R

```
# x gets the number 3.14
```

```
x <- 3.14
```

```
x      # print x
```

```
## [1] 3.14
```

```
# equivalently
```

```
x = 3.14
```

```
x      # print x
```

```
## [1] 3.14
```

Data types examples

3 ways to create numeric vector:

```
# empty numeric vector  
y1 <- numeric(6)  
y1      # print y1
```

```
## [1] 0 0 0 0 0 0
```

```
y2 <- vector(mode = "numeric", length = 6)  
y2      # print y2
```

```
## [1] 0 0 0 0 0 0
```

```
y3 <- c(5, 13.222, 2, 0.001, 77.4, 31.9)  
y3      # print y3
```

```
## [1] 5.000 13.222 2.000 0.001 77.400 31.900
```


Data structures examples

Create a data.frame out of the following “class” data:

- Has Master's (logical): TRUE FALSE FALSE TRUE
- GPA (numeric): 3.1 4.0 2.9 3.6
- First Name (character): Mike Dan Sara Karen

```
# store data
has_ms <- c(TRUE, FALSE, FALSE, TRUE)
gpa <- c(3.1, 4.0, 2.9, 3.6)
name <- c("Mike", "Dan", "Sara", "Karen")
# Create data.frame
dat <- data.frame(has_MS = has_ms, GPA = gpa, Name = name)
dat      # print data.frame
```

```
##   has_MS GPA  Name
## 1   TRUE 3.1  Mike
## 2  FALSE 4.0   Dan
## 3  FALSE 2.9  Sara
## 4   TRUE 3.6 Karen
```

Inspect an object

- `Class()` - what kind of object is it (high-level)?
- `typeof()` - what is the data type (low-level)?
- `length()` - how long is it?
- `attributes()` - does it have meta-data?

R functions

R function syntax:

```
NAME <- function(ARG1, ARG2, ARG3) {  
  DO SOMETHING  
  STORE RESULT  
  return(RESULT)  
}
```

```
pow <- function(base, expon) { # power function  
  prod(rep(base, expon)) # base^(expon)  
}  
# Use power function  
pow(5, 2)
```

```
## [1] 25
```

```
pow(10, 3)
```

```
## [1] 1000
```

Common R functions

R has a huge collection of packages:

- 6,000+ packages for data analysis build (on CRAN alone)

Example: `lm` (linear models)

- Use `?lm` to read help documentation

`lm {stats}`

R Documentation

Fitting Linear Models

Description

`lm` is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance (although [aov](#) may provide a more convenient interface for these).

Usage

```
lm(formula, data, subset, weights, na.action,  
   method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,  
   singular.ok = TRUE, contrasts = NULL, offset, ...)
```

Arguments

- | | |
|----------------------|--|
| <code>formula</code> | an object of class " formula " (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'. |
| <code>data</code> | an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in <code>data</code> , the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>lm</code> is called. |

S3 classes

- S3 is most common class type of R objects
- Know how to inspect it:

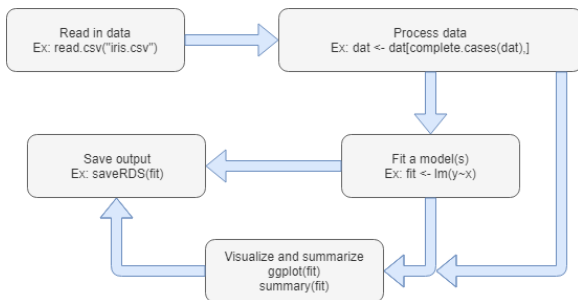
R on cluster (non-interactive R)

Cluster Computation

- Dan Barker danbarke at umich.edu
- Cluster System Administrator

R workflow for statistical analysis

- Bootcamps throughout semester on each area



Slide with R Output

```
summary(cars)
```

##	speed	dist
##	Min. : 4.0	Min. : 2.00
##	1st Qu.:12.0	1st Qu.: 26.00
##	Median :15.0	Median : 36.00
##	Mean :15.4	Mean : 42.98
##	3rd Qu.:19.0	3rd Qu.: 56.00
##	Max. :25.0	Max. :120.00

Slide with Plot

