Introduction to R

Augustin Luna
20 January, 2016

Research Fellow
Department of Biostatistics and Computational Biology
Dana-Farber Cancer Institute

Topics to be Covered

- R Language Basics
- Vectors and Selections
- Matrices and Data Frames
- Writing and Reading Data
- Plotting
- Control Flows
  - for Loops
  - if Statements
- Using Packages
  - Installing
  - Loading
  - Viewing Help
- Additional Common Functions

Basics

- From: https://github.com/Data-Camp/introduction_to_R/blob/master/chapter1.Rmd

Simple calculations

```
2 + 2
```

```
[1] 4
```

Commenting Code

```
# This is a comment
2 + 2
```

```
[1] 4
```

```
# Addition
5 + 5
```

```
[1] 10
```

```
# Subtraction
5 - 5
```

```
[1] 0
```

```
# Multiplication
3 * 5
```

```
[1] 15
```

```
# Division
(5 + 5) / 2
```

```
[1] 5
```

Variable Assignments

```
my_variable <- 4
my_variable
```

```
[1] 4
```

Basic Data Types

```r
# What is the answer to the universe?
my_numeric <- 42

# The quotation marks indicate that the
variable is of type character
my_character <- "some text"

# Change the value of my_logical
my_logical <- TRUE
```

Help

```
?mean
```

Vectors

- From: https://github.com/Data-Camp/introduction_to_R/blob/master/chapter2.Rmd

Creating a vector

```r
numeric_vector <- c(1, 2, 3)
character_vector <- c("a", "b", "c")
boolean_vector <- c(TRUE, FALSE, FALSE)
```

Selection by index

```r
numeric_vector[c(1, 3)]
```

```
[1] 1 3
```

Selection by logical

```r
my_variable <- 2
result <- numeric_vector[numeric_vector > my_variable]
result
```

```
[1] 3
```

Matrices

- From: https://github.com/Data-Camp/introduction_to_R/blob/master/chapter3.Rmd

Matrices from vectors

```r
first_row <- c(6,8,7,9,9,10)
second_row <- c(6,8,7,5,9,6)
third_row <- c(5,4,6,6,7,8)
fourth_row <- c(4,5,3,4,6,8)

# Combine multiple vectors to form a matrix
theater <- rbind(first_row, second_row,
third_row, fourth_row)
row_scores <- rowSums(theater)
scores <- cbind(theater, row_scores)
```

Naming a Matrix

```r
rownames(scores) <- c("row1", "row2", "row3", "row4")
colnames(scores) <- c("col1", "col2", "col3","col4",
"col5", "col6", "total")
scores
```

```
     col1 col2 col3 col4 col5 col6 total
row1    6    8    7    9    9   10    49
row2    6    8    7    5    9    6    41
row3    5    4    6    6    7    8    36
row4    4    5    3    4    6    8    30
```

Size of Matrix

```
ncol(scores)
```

```
[1] 7
```

```
nrow(scores)
```

```
[1] 4
```

```
dim(scores)
```

```
[1] 4 7
```

Selecting Elements

Select rows and columns

```
i <- 1
j <- 1

scores[i,]
```

```
 col1  col2  col3  col4  col5  col6 total 
    6     8     7     9     9    10    49 
```

```
scores[,j]
```

```
row1 row2 row3 row4 
   6    6    5    4 
```

```
scores[i,j]
```

```
[1] 6
```

Data Frames

```
data(iris)

# See the first 6 rows of a data.frame
head(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width
Species
1          5.1         3.5          1.4         0.2
setosa
2          4.9         3.0          1.4         0.2
setosa
3          4.7         3.2          1.3         0.2
setosa
4          4.6         3.1          1.5         0.2
setosa
5          5.0         3.6          1.4         0.2
setosa
6          5.4         3.9          1.7         0.4
setosa
```

```
# See the last 6 rows of a data.frame
tail(iris)
```

Rename data.frame Columns

```r
numeric_vector <- c(1, 2, 3)
character_vector <- c("a", "b", "c")
boolean_vector <- c(TRUE, FALSE, FALSE)

df <- data.frame(numbers=numeric_vector,
characters=character_vector, boolean=boolean_vector)

df
```

```
  numbers characters boolean
1       1          a    TRUE
2       2          b   FALSE
3       3          c   FALSE
```

Selecting Columns by Name

```
iris[,"Sepal.Length"]
```

```
  [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4
 [18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5
 [35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0
 [52] 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8
 [69] 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4
 [86] 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8
[103] 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7
[120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7
[137] 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9
```

```
iris$Sepal.Length
```

```
  [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4
 [18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5
 [35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0
 [52] 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8
 [69] 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4
 [86] 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8
[103] 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7
[120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7
[137] 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9
```

Exporting Data

Writing files

```
write.table(iris, file="iris.txt", sep="\t",
row.names=TRUE, col.names=TRUE, quote=FALSE)
```

Reading files

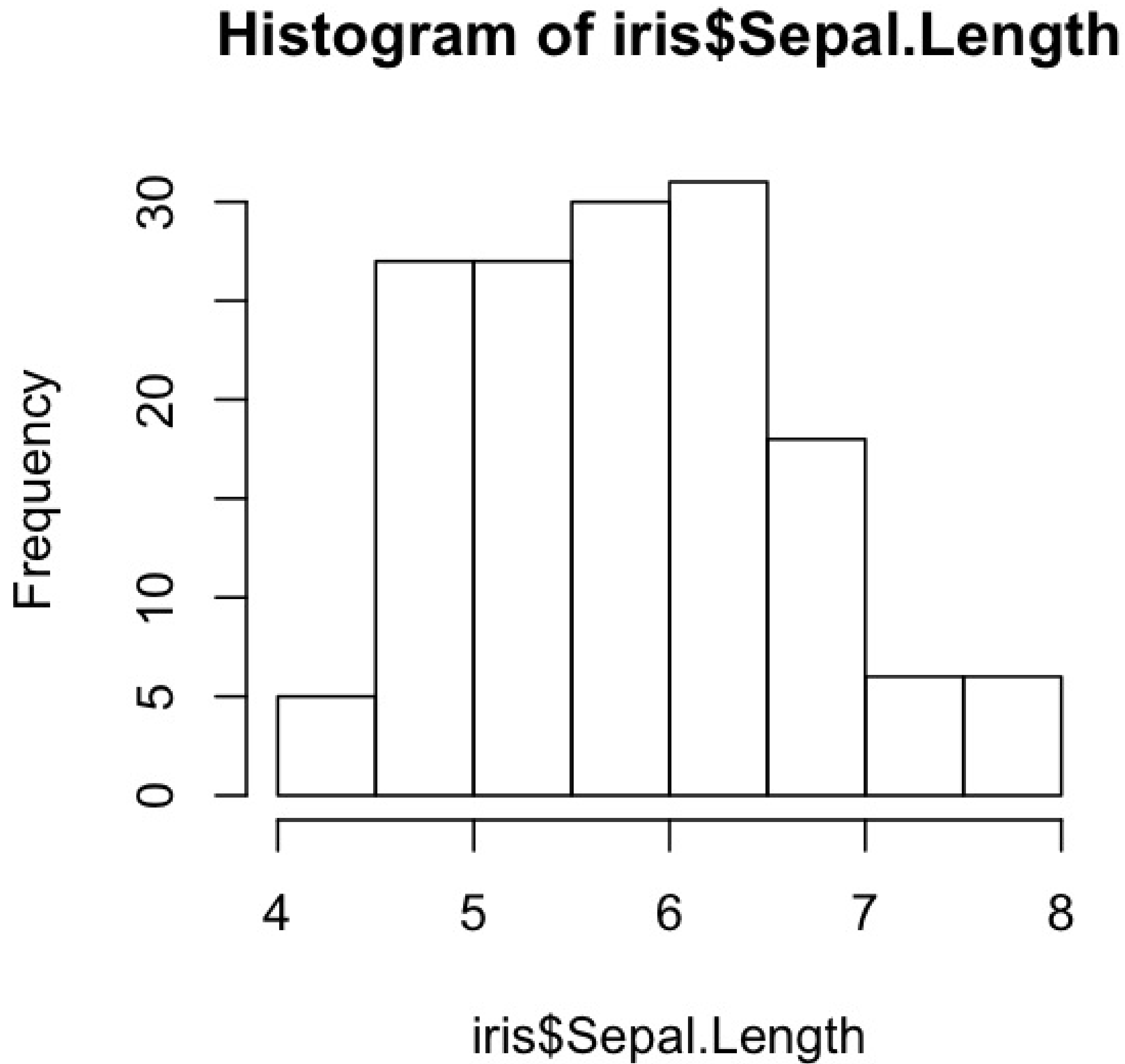
```
df <- read.table("iris.txt", sep="\t",
header=TRUE)
```

Plotting

- From: https://github.com/Data-Camp/introduction_to_R/blob/master/chapter7.Rmd

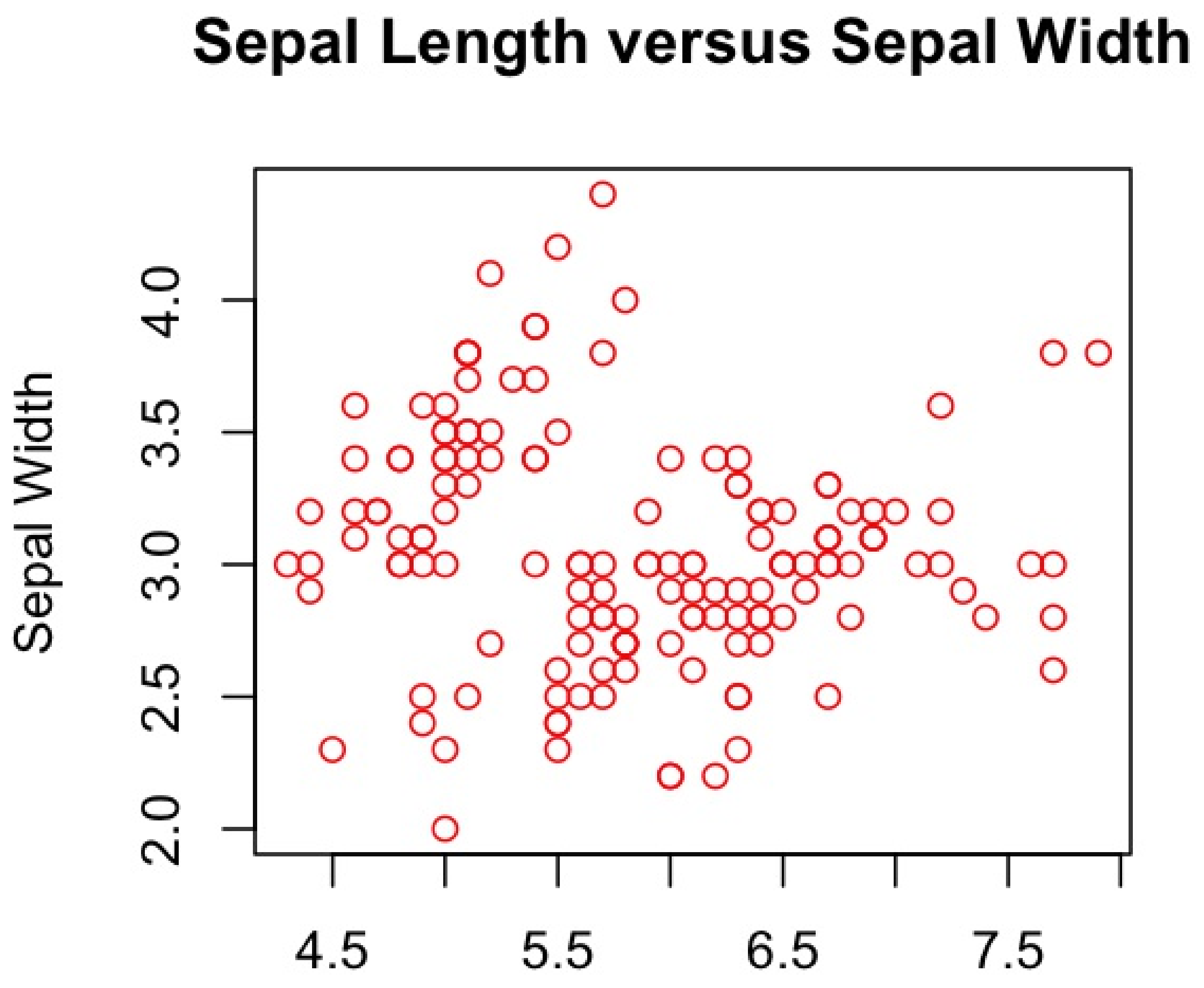
Histogram

```r
hist(iris$Sepal.Length)
```



Scatterplot

```
plot(x=iris$Sepal.Length,
     y=iris$Sepal.Width,
     main = "Sepal Length versus Sepal Width",
     xlab = "Sepal Length",
     ylab = "Sepal Width",
     col = "red")
```



for Loops

```r
hellos <- c("Hello World!",
            "Hola Mundo",
            "Ola Mundo!")

for(hello in hellos) {
  cat(hello, "\n")
}
```

```
Hello World!
Hola Mundo
Ola Mundo!
```

if Statements

```r
a <- 5

# Greater than or equal to
if(a >= 5) {
  cat("Greater than or equal to")
} else {
  cat("Not greater than or equal to")
}
```

```
Greater than or equal to
```

```r
# Equivalent
if(a == 5) {
  cat("Equal")
}
```

```
Equal
```

```r
# Not equivalent
if(a != "Hello World!") {
  cat("Not equal")
}
```

```
Not equal
```

R Packages

- From: http://www.jkarreth.net/files/RPOS517_Day1_IntroR.pdf

Install packages from repositories

- NOTE: These commands are commented out since these packages are already installed

```
# From CRAN (for general packages)
install.packages("httr")

# From Bioconductor (for biology-related packages)
source("https://bioconductor.org/biocLite.R")

biocLite("rcellminer")
```

Load Package

```r
library(rcellminer)

# Check if package was loaded
sessionInfo()
```

Package Help

```r
help(package="rcellminer")
```

length Function

```r
# Find the length of a vector
my_variable <- runif(100)
length(my_variable)
```

```
[1] 100
```

min, max, summary Functions

```
# Find the minimum
min(my_variable)
```

```
[1] 0.001824665
```

```
# Find the maximum
max(my_variable)
```

```
[1] 0.9447014
```

```
# Output a summary statistics of vector
summary(my_variable)
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
0.001825 0.173600 0.451800 0.448800 0.692900 0.944700 
```

cat, paste Functions

```
hello <- c("hello", "hola", "ola")
world <- c("world", "mundo")

# Make a new string from multiple variable and
separated by "sep"
helloWorld <- paste(hello[1], world[2], sep=" ")
cat(helloWorld)
```

```
hello mundo
```

names Function

```r
indicies <- 1:10

randNum <- runif(max(indicies))
vectorNames <- letters[indicies]

# Name the randNum vector according to
vectorNames
names(randNum) <- vectorNames
```

list Function

```
# Make a list variable; each list element has a different
length
my_list <- list(a=1:5, b=1:10, c=1:100)

names(my_list)
```

```
[1] "a" "b" "c"
```

```
my_list$a
```

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

```
my_list[[1]]
```

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

```
my_list[["a"]]
```

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

```
length(my_list)
```

is.na, which Function and not Operator

```
my_vector <- c(1, 2, NA, 4, 5, 6, 7, 8, NA, 10)

# Is each element in my_vector an NA
is.na(my_vector)
```

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

```
# Which indicies in my_vector are NA
which(is.na(my_vector))
```

```
[1] 3 9
```

```
# Which indicies in my_vector are not NA
which(!is.na(my_vector))
```

```
[1]  1  2  4  5  6  7  8 10
```

is.null Function

```
# NULL variables have undefined values
my_vector <- NULL
my_vector
```

```
NULL
```

```
is.null(my_vector)
```

```
[1] TRUE
```

```
my_vector <- c(my_vector, 5)
my_vector <- c(my_vector, 6)
my_vector
```

```
[1] 5 6
```

```
is.null(my_vector)
```

```
[1] FALSE
```

```
is.vector(my_vector)
```

```
[1] TRUE
```

is.nan Function

```r
my_variable <- NaN

is.nan(my_variable)
```

```
[1] TRUE
```

unique Function

```r
my_vector <- c(1, 1, 2, 3, 3, 4, 4, 5)

# Find the unique values in a vector
unique(my_vector)
```

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

sort Function

```r
my_vector <- c(1, 4, 3, 6, 7, 10, 9, 5, 2, 8)

# Sort values in vector
sort(my_vector)
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

```r
sort(my_vector, decreasing=TRUE)
```

```
 [1] 10  9  8  7  6  5  4  3  2  1
```

%in% Function

```
restaurant_foods <- c("mango", "chicken", "pork", "chips",
                      "cookies", "cake", "muffins",
"cupcakes")

favorite_foods <- c("mango", "orange", "cake", "chicken")

# Does the restaurant have my favorite foods?
restaurant_foods %in% favorite_foods
```

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

```
# What are the indicies of my favorite foods
which(restaurant_foods %in% favorite_foods)
```

```
[1] 1 2 6
```

```
# Return my favorite foods
restaurant_foods[which(restaurant_foods %in% favorite_foods)]
```

```
[1] "mango"   "chicken" "cake"
```

Getting Help

- Stack Overflow
  - http://stackoverflow.com/
- Cross-Validated Stats Exchange
  - Part of Stack Overflow
  - http://stats.stackexchange.com/
- Biostars
  - https://www.biostars.org