Very basic statistical computing in R

Malachy Campbell

11/9/2018

Outline

- Very basics of R
  - Types of objects and classes
  - Input and output (setwd, read/write)
- Manipulating data
  - manipulating datasets with reshape2 and tidyr
  - Applying functions over datasets (for loops, apply and plyr)
- Data visualization
  - Base graphics and ggplot2)
- Presenting data
  - Reproducible reports and presentations in Rmarkdown

Basics of R: Objects and object types

- Programming in R is centered around the use of objects (data structures with attributes)
  - Apply some method or function on object
- These objects can be of several different types (character, factor, dataframe, list, etc.)

```
someObject <- 5

#What type of object is this?
str(someObject)
```

```
##  num 5
```

```
is.numeric(someObject)
```

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

```
anotherObject <- "5"
str(anotherObject)
```

```
##  chr "5"
```

```
is.character(anotherObject)
```

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

```
#Must quote strings or characters or else they are treated as objects
finalObject <- "five"
str(finalObject)
```

Basics of R: Manipulating objects

- The object type will restrict the functions that can be used on the object

```
#Try to add some number to 'anotherObject', a character vector containing '5'
try(anotherObject + 5)

#as.numeric forces some object to be numeric
anotherObject <- as.numeric(anotherObject)
anotherObject + 5
```

```
## [1] 10
```

```
#finalObject contains the string "five". Try and convert "five" to some number. R doesn't like this.
as.numeric(finalObject)
```

```
## Warning: NAs introduced by coercion
```

```
## [1] NA
```

Basics of R: Vectors, matrices, data frames, and lists

- All of these store more than one element

```
#c() combines
aVector <- c(anotherObject, 16, 20, 22, 27, 30)
aVector
```

```
## [1]  5 16 20 22 27 30
```

```
mean(aVector)
```

```
## [1] 20
```

```
#Access the 5th element of aVector
aVector[5]
```

```
## [1] 27
```

```
aMatrix <- matrix(aVector, nrow = 2, ncol = 3, byrow = T)
aMatrix
```

```
##      [,1] [,2] [,3]
## [1,]    5   16   20
## [2,]   22   27   30
```

```
#Access the element in the second row and third column of aMatrix [x,y]
aMatrix[2,3]
```

```
## [1] 30
```

Basics of R: Data frames

- A data frame is an object that can handle multiple types of data

```
#Rep(X, N): repeat some value(s) N times
aDataframe <- data.frame(Obs = rep(c("A", "B"), 3), Trt = rep(c("T1", "T2"), 3), Val = aVector)
str(aDataframe)
```

```
## 'data.frame':    6 obs. of  3 variables:
##  $ Obs: Factor w/ 2 levels "A","B": 1 2 1 2 1 2
##  $ Trt: Factor w/ 2 levels "T1","T2": 1 2 1 2 1 2
##  $ Val: num  5 16 20 22 27 30
```

```
#show only the first three rows. Rows are indicated by [x,], columns are [,x]
aDataframe[1:3,]
```

```
##   Obs Trt Val
## 1   A  T1   5
## 2   B  T2  16
## 3   A  T1  20
```

```
#access specific columns using object$'column name' or object[,colnumber]
aDataframe$Obs[1:3]
```

```
## [1] A B A
## Levels: A B
```

```
mean(aDataframe$Val)
```

```
## [1] 20
```

Basics of R: Lists

- Lists can store any type of data in each of its elements

```
firstList <- list(aVector, aMatrix, aDataframe)
firstList[1]
```

```
## [[1]]
## [1]  5 16 20 22 27 30
```

```
nestedList <- list(c(1,2,3,4), firstList)
nestedList
```

```
## [[1]]
## [1] 1 2 3 4
##
## [[2]]
## [[2]][[1]]
## [1]  5 16 20 22 27 30
##
## [[2]][[2]]
##      [,1] [,2] [,3]
## [1,]    5   16   20
## [2,]   22   27   30
##
## [[2]][[3]]
##   Obs Trt Val
## 1   A  T1   5
## 2   B  T2  16
## 3   A  T1  20
## 4   B  T2  22
## 5   A  T1  27
## 6   B  T2  30
```

Input and output

- R can read various types of text files (.csv, .txt, .xlsx (try to avoid), etc.)

```
setwd("/Users/malachycampbell/Documents/Dropbox/Work/Presentations/Japan/IntroToR/")
#Data <- read.csv("/Users/malachycampbell/Documents/Dropbox/Work/Presentations/Japan/IntroToR/SeriousData.c
Data <- read.csv("SeriousData.csv")
head(Data)
```

```
##   Obs Plot Line  Loc Height
## 1   1    A   L1 Home     15
## 2   2    A   L2 Home     12
## 3   3    A   L3 Home     NA
## 4   4    A   L4 Home     14
## 5   5    A   L5 Home     20
## 6   6    B   L1 Home     22
```

```
str(Data)
```

```
## 'data.frame':    20 obs. of  5 variables:
##  $ Obs   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Plot  : Factor w/ 2 levels "A","B": 1 1 1 1 1 2 2 2 2 2 ...
##  $ Line  : Factor w/ 5 levels "L1","L2","L3",..: 1 2 3 4 5 1 2 3 4 5 ...
##  $ Loc   : Factor w/ 2 levels "Away","Home": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Height: int  15 12 NA 14 20 22 NA 18 12 19 ...
```

```
write.csv(Data, "SomeData.csv", row.names = F)
```

Input and output

- R can read various types of text files (.csv, .txt, .xlsx (try to avoid), etc.)

```
str(Data)
```

```
## 'data.frame':    20 obs. of  5 variables:
##  $ Obs   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Plot  : Factor w/ 2 levels "A","B": 1 1 1 1 1 2 2 2 2 2 ...
##  $ Line  : Factor w/ 5 levels "L1","L2","L3",..: 1 2 3 4 5 1 2 3 4 5 ...
##  $ Loc   : Factor w/ 2 levels "Away","Home": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Height: int  15 12 NA 14 20 22 NA 18 12 19 ...
```

```
write.csv(Data, "SomeData.csv", row.names = F)
```

Cleaning up data

```
## Here's that data you asked for!,,,,
## It starts now!,,,,
## Obs,Plot,Line,Loc,Height
## 1,A,L1,Home,15
## 2,A,L2,Home,12
## 3,A,L3,Home,Missing
## 4,A,L4,Home,14
## 5,A,L5,Home,20
## 6,B,L1,Home,22
## 7,B,L2,Home,Missing
```

```
Data <- read.csv("SillyData.csv")
head(Data)
```

```
##   Here.s.that.data.you.asked.for.    X  X.1  X.2     X.3
## 1                  It starts now!
## 2                             Obs Plot Line  Loc  Height
## 3                               1    A   L1 Home      15
## 4                               2    A   L2 Home      12
## 5                               3    A   L3 Home Missing
## 6                               4    A   L4 Home      14
```

Cleaning up data

```
Data <- read.csv("SillyData.csv", header = T, skip = 2)
str(Data)
```

```
## 'data.frame':    20 obs. of  5 variables:
##  $ Obs   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Plot  : Factor w/ 2 levels "A","B": 1 1 1 1 1 2 2 2 2 2 ...
##  $ Line  : Factor w/ 5 levels "L1","L2","L3",..: 1 2 3 4 5 1 2 3 4 5 ...
##  $ Loc   : Factor w/ 2 levels "Away","Home": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Height: Factor w/ 11 levels "12","14","15",..: 3 1 11 2 6 8 11 4 1 5 ...
```

```
#Replace 'Missing' with NA
Data[Data$Height == "Missing", ]$Height <- NA

#Convert the height column to numeric
Data$Height <- as.numeric(as.character(Data$Height) )
```

Cleaning up data

```
head(Data)
```

```
##   Obs Plot Line  Loc Height
## 1   1    A   L1 Home     15
## 2   2    A   L2 Home     12
## 3   3    A   L3 Home     NA
## 4   4    A   L4 Home     14
## 5   5    A   L5 Home     20
## 6   6    B   L1 Home     22
```

```
str(Data)
```

```
## 'data.frame':    20 obs. of  5 variables:
##  $ Obs   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Plot  : Factor w/ 2 levels "A","B": 1 1 1 1 1 2 2 2 2 2 ...
##  $ Line  : Factor w/ 5 levels "L1","L2","L3",..: 1 2 3 4 5 1 2 3 4 5 ...
##  $ Loc   : Factor w/ 2 levels "Away","Home": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Height: num  15 12 NA 14 20 22 NA 18 12 19 ...
```