Data Handling and Data Mining

Fixing The Sparrow Data Set

-

BASIC STATISTICS FOR BIOLOGISTS

Erik Kusch
PhD Student
Aarhus University
Department of Bioscience
Section for Ecoinformatics & Biodiversity
Center for Biodiversity and Dynamics in a Changing World (BIOCHANGE)
Ny Munkegade 116, Building 1540
8000 Aarhus
Denmark
email: erik@i-solution.de

Summary:

These are the solutions to the exercises contained within the handout to Introduction to R which walks you through the basics of the R machinery. R is a coding language that can be highly individualised and hence there are often multiple solutions to the same problem. Within these solutions, I shall only present you with one solution for every given task. However, do keep in mind that there is probably a myriad of other ways to achieve your goal.

Contents

1. Creating and Inspecting Objects

1.1 Vector

- A vector reading: “A”, “B”, “C”

```
Letters_vec <- c("A", "B", "C")
Letters_vec
```

```
## [1] "A" "B" "C"
```

```
length(Letters_vec)
```

```
## [1] 3
```

- A vector reading: 1, 2, 3

```
Numbers_vec <- c(1, 2, 3)
Numbers_vec
```

```
## [1] 1 2 3
```

```
length(Numbers_vec)
```

```
## [1] 3
```

- A vector reading: `TRUE`, `FALSE`

```
Logic_vec <- c(TRUE, FALSE)
Logic_vec
```

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

```
length(Logic_vec)
```

```
## [1] 2
```

- A vector of the elements of the first three vectors

```
Big_vec <- c(Letters_vec, Numbers_vec, Logic_vec)
Big_vec
```

```
## [1] "A"     "B"     "C"     "1"     "2"     "3"     "TRUE"  "FALSE"
```

```
length(Big_vec)
```

```
## [1] 8
```

- A vector reading as a sequence of full numbers from 1 to 20

```
Seq_vec <- c(1:20)
Seq_vec
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```
length(Seq_vec)
```

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

1.2 Factor

- A factor reading: "A", "B", "C"

```
Letters_fac <- factor(x = c("A", "B", "C"))
Letters_fac
```

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

```
length(Letters_fac)
```

```
## [1] 3
```

- A factor reading: 1, 2, 3

```
Numbers_fac <- factor(x = c(1, 2, 3))
Numbers_fac
```

```
## [1] 1 2 3
## Levels: 1 2 3
```

```
length(Numbers_fac)
```

```
## [1] 3
```

- A factor reading: 1, 2, 3 but only levels 1 and 2 are allowed

```
Constrained_fac <- factor(x = c(1, 2, 3), levels = c(1, 2))
Constrained_fac
```

```
## [1] 1    2    <NA>
## Levels: 1 2
```

```
length(Constrained_fac)
```

```
## [1] 3
```

- A factor reading: 1, 2, 3 levels 1 - 4 are allowed

```
Expanded_fac <- factor(x = c(1, 2, 3), levels = c(1, 2, 3, 4))
Expanded_fac
```

```
## [1] 1 2 3
## Levels: 1 2 3 4
```

```
length(Expanded_fac)
```

```
## [1] 3
```

1.3 Matrix

- The first two vectors we established in distinct columns of a matrix

```
Combine_mat <- matrix(data = c(Numbers_vec, Letters_vec), ncol = 2)
Combine_mat
```

```
##      [,1] [,2]
## [1,] "1"  "A" 
## [2,] "2"  "B" 
## [3,] "3"  "C"
```

```
dim(Combine_mat)
```

```
## [1] 3 2
```

- The first two vectors we established in distinct rows of a matrix

```
Pivot_mat <- matrix(data = c(Numbers_vec, Letters_vec), nrow = 2, byrow = TRUE)
Pivot_mat
```

```
##      [,1] [,2] [,3]
## [1,] "1"  "2"  "3" 
## [2,] "A"  "B"  "C"
```

```
dim(Pivot_mat)
```

```
## [1] 2 3
```

- The above matrix with meaningful names

```
Names_mat <- Pivot_mat
dimnames(Names_mat) <- list(c("Numbers", "Letters"))
Names_mat
```

```
##         [,1] [,2] [,3]
## Numbers "1"  "2"  "3" 
## Letters "A"  "B"  "C"
```

```
dim(Names_mat)
```

```
## [1] 2 3
```

1.4 Data Frame

- The first matrix we established as a data frame

```
Combine_df <- data.frame(Combine_mat)
Combine_df
```

```
##   X1 X2
## 1  1  A
## 2  2  B
## 3  3  C
```

```
dim(Combine_df)
```

```
## [1] 3 2
```

- The previous data frame with meaningful names

```
Names_df <- Combine_df
colnames(Names_df) <- c("Numbers", "Letters")
Names_df
```

```
##   Numbers Letters
## 1       1       A
## 2       2       B
## 3       3       C
```

```
dim(Names_df)
```

```
## [1] 3 2
```

1.5 List

- The first two vectors we created

```
Vectors_ls <- list(Numbers_vec, Letters_vec)
Vectors_ls
```

```
## [[1]]
## [1] 1 2 3
##
## [[2]]
## [1] "A" "B" "C"
```

```
length(Vectors_ls)
```

```
## [1] 2
```

2. Statements and Loops

2.1 Statements

- Numbers_vec contains more elements than Letters_fac?

```
length(Numbers_vec) > length(Letters_fac)
```

```
## [1] FALSE
```

- The first column of Combine_df is shorter than Vectors_ls?

```
length(Combine_df[, 1]) < length(Vectors_ls)
```

```
## [1] FALSE
```

- The elements of Letters_vec are the same as the elements of Letters_fac?

```
Letters_vec == Letters_fac
```

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

2.2 Loops

- Print each element of Vectors_ls

```
for (i in 1:length(Vectors_ls)) {
    print(Vectors_ls[[i]])
}
```

```
## [1] 1 2 3
## [1] "A" "B" "C"
```

- Print each element of Numbers_vec + 1

```
Numbers_veca <- Numbers_vec + 1
for (i in 1:length(Numbers_veca)) {
    print(Numbers_veca[i])
}
```

```
## [1] 2
## [1] 3
## [1] 4
```

- Subtract 1 from each element of the first column of Combine_mat and print each element separately

```
Mat_column <- Combine_mat[, 1]  # extract data
Mat_column <- as.numeric(Mat_column)  # convert to numeric
Mat_column <- Mat_column - 1  # substract 1
for (i in 1:length(Mat_column)) {
    print(Mat_column[i])
}
```

```
## [1] 0
## [1] 1
## [1] 2
```

3. Useful Commands

- Read out your current working directory (not showing you the result as it is different on every machine, it should start like this "C:/Users/. . . .")

```
getwd()
```

- Inspect the `Vectors_ls` object using the `View()` function (again, I am not showing you the result as this only works directly in `R` or Rstudio)

```
View(Vectors_ls)
```

- Inspect the `Combine_df` object using the `View()` function (again, I am not showing you the result as this only works directly in `R` or Rstudio)

```
View(Combine_df)
```

- Get the help documentation for the `as.matrix()` function (again, I am not showing you the result as this only works directly in `R` or Rstudio)

```
`?`(as.matrix)
```

- Install and load the `dplyr` package

```
install.packages("dplyr")
library(dplyr)
```

- Remove the `Logic_vec` object from your working environment

```
rm(Logic_vec)
```

- Clear your entire working environment

```
ls()  # this command shows you all the object in the environment
```

```
##  [1] "Big_vec"         "Combine_df"      "Combine_mat"     "Constrained_fac"
##  [5] "Expanded_fac"    "i"               "Letters_fac"     "Letters_vec"
##  [9] "Mat_column"      "Names_df"        "Names_mat"       "Numbers_fac"
## [13] "Numbers_vec"     "Numbers_veca"    "Pivot_mat"       "Seq_vec"
## [17] "Vectors_ls"
```

```
rm(list = ls())
```