Data in R

Juan Camilo Rivera. [1] Hugo Andres Dorado. [1]

[1] Big data and site-specific agriculture
Decision and Policy Analysis
Centro Internacional de Agricultura Tropical

Data analysis course, 2018

Outline

1. Basic concepts and data organization
   - Basic Syntax
   - Data Types
   - R objects
   - Operations on Matrices and Data.Frames

Basic Syntax

- **R prompt**. The symbol is the greater than >.

```
> 2 * 3
[1] 6
```

- **Continuation prompt**. A continuation prompt (+) appears.

```
> 2 *
+  3
[1] 6
```

- **Assignment operator**. The left arrow assigns the value of the object on the right to the object on the left.

```
> value <- 3*2
```

- **Last Expression**.

```
> value <-.Last.value
> value
[1] 8
```

- **Removing Objects**. `rm` or `remove`

```
> rm(value)
```

- **Main functions**

```
> attach()
> search()
> objects()
> find()
> ?function help(function)
```

Data type

There are four atomic types in R.

- Numeric

```
> value <- 12
```

- Character

```
> string <- "Hello World"
```

- Logical

```
> 2 < 4
[1] TRUE
```

- Complex number

```
> cn <- 3 + 4i
[1] 3 + 4i
```

Functions

```
>mode()
>names()
>length()
```

R objects

Vector is a set of elements of the same mode.

Creating vectors

```
> value_num  <- c(1,2,3,5)
> value_char <- c("koala", "kangaroo", "elephant")
> value_logical <- c(F, T, T,T)
> value_logical2 <- c(FALSE, TRUE, TRUE,TRUE)
```

Functions rep and seq

```
> value  <- rep(1,5)
> value  <- 1:5
> value_seq <- seq(from = 2, to = 10, by =2 )
> value_seq2 <- seq(along = value)
> value_logical <- c(1,2,3, rep(3,4), seq(from=1, to=6, by=2))
```

R objects

Matrix is a collection data elements arranged in a two-dimensional rectangular layout.

Creating matrices

The dim function can be used to convert a vector to a matrix

```
> value <- rnorm(6)
> dim (value) <- c(2,3)
```

Functions matrix rbind and cbind

```
> matrix(value, 2, 3)
> matrix(value, 2, 3, byrow= T)
> rbind(value, c(1,1,2))
> cbind(value, c(1,2,2))
```

R objects

Data frame It is a matrix with each line corresponding to an individual and each column corresponding to a variable measured on the individuals.

Creating data.frame

```
> BMI <- data.frame(Gender = c("M", "F", "M", "F"),
+  Height = c(1.83, 1.76, 1.82, 1.60)
+  Weight = c(67, 58, 48, 76)
+  row.names = c("Jack", "Julia", "Henry", "Emma")
```

Functions

```
> data.frame()
> is.data.frame()
> class()
> str()
```

Accesing Elements of a Vector or Matrix

- Indexing vectors

```
> x <- sample(1:5, 20, T)
> x == 1
> ones <- x == 1
> x[ones]
> other <- x > 1
> x[other]
> which(x > 1)
```

- Indexing data frame

```
> # index by column
> BMI[,"Gender"] <- 0
> # index by row
> BMI[1,] <- 0
> BMI[] <- 1:12
> BMI[1:2,]
> BMI[,1:2]
```

Operations on Matrices and Data.Frames

These are most important functions which give information on a matrix or a data.frame:

- dim(): size of the matrix or data.frame
- nrow(): number of rows
- ncol(): number of columns
- dimnames(): names of rows and columns (as a list)
- names(), colnames(): names of columns
- rownames(): names of rows

Operations on Matrices and Data.Frames

Merging columns. The basic functions are **cbind()** and **rbind()** .

```
> cbind(1:4,5:8)
> X1 <- data.frame(Id=1:4,GENDER=c("M","F","F","M"),
+
Weight=c(75,68,48,72))
> X2 <- data.frame(Id=1:4,GENDER=c("M","F","F","M"),
+
Height=c(182,165,160,178))
>cbind(X1,X2)
```

Operations on Matrices and Data.Frames

Merge

```
> merge(X1,X2)
> X3 <- data.frame(Id=c(2,1,4,3),GENDER=c("F","M","M","F"), He
> merge(X1, X3)
> merge(X,Y,by=c("GENDER","Weight"))
> merge(X,Y,by=c("GENDER","Weight"),all=TRUE)
```

Operations on Matrices and Data.Frames

Merging lines. The generic function is rbind() and the function **apply()** applies a given function (`FUN`) to all rows (`MARGIN = 1`) or to all columns (`MARGIN = 2`).

```
> rbind(1:4,5:8)
> X <- matrix(c(1:4, 1, 6:8), nr = 2)
> apply(X, MARGIN=1, FUN=mean)
> apply(X, MARGIN=2, FUN=sum)
```

R objects

Lists is a object that incorporate mixture of modes into one list and each component can be of a different length or size

Creating lists

```
> L1 <- list(x = sample (1:5, 20, rep =T),
+ y = rep(letters[1:5],4), z = rpois(20,1))
```

Accessing

```
> L1[["x"]]
> L1$x
> L1[[1]]
> L1[1] #the first component
> names(L1) <- c("Item1","Item2","Item3")
> L1$Item1[L1$Item1 > 2]
```

R objects

Dates it is a structure the data representing dates.

```
> dates <- c("92/27/02", "92/02/27", "92/01/14",
+"92/02/28", "92/02/01")
> dates <- as.Date(dates, "%y/%m/%d")
> dates
[1] NA, "1992-02-27" "1992-01-14" "1992-02-28"
[5] "1992-02-01"
> class(dates)
[1] "Date"
```

R objects

Time Series ts is a function that organize them into an structure that reflects the temporal.

```
>ts(1:10, frequency = 4, start = c(1959, 2))
```

Factor. It is used to store quantitative variables.

```
>x <- factor(c("blue","green","blue","red",
"blue","green","green"))
>levels(x)
[1] "blue" "green" "red"
> class(x)
>Poids <- c(55,63,83,57,75,90,73,67)
>cut(Poids,3)
```

Importing and exporting

- scan(). The low - level Input function.It is useful when the data are not organized as a rectangular table.

```
# Reading variable names:
variable.names <- scan("intima_media.txt",skip=4,nlines=1,what="")
# Reading data:
data <- scan("intima_media.txt",skip=7,dec=",")
mytable <- as.data.frame(matrix(data,ncol=9,byrow=TRUE))
colnames(mytable) <- variable.names
```

Importing and exporting

- read.table(). can be used to read data frames from formatted text files.

```
# Download info
# http://www.biostatisticien.eu/springeR/Intima_Media_Thickness.txt
mydata <- read.table("Intima_Media_Thickness.txt",sep=" ",
header=TRUE,dec=",")
mydata
head(mydata)
```

Importing and exporting

Others types:

- read.csv(). To read comma-separated data
- read.csv2(). To read semi-colon-separated data
- read.delim() with a . as decimal mark
- read.delim2() with a , as decimal mark

For Further Reading I

- Pierre Lafaye de Micheaux.
  The R Software,
  Fundamentals of programming and statistical analysis
  Springer, 2013.

- William Sullivan
  Machine Learning for Beginners Guide Algorithms

- Giuseppe Ciaburro
  Balaji Venkateswaran
  Neural Networks with R