R Basics and Examples - A short introduction

Markus Baaske

Faculty of Mathematics and Computer Science (FSU Jena)

April 30, 2018

The R Project for Statistical Computing

The R project `http://www.r-project.org` develops a free software environment for statistical computing and graphics. R compiles and runs on a wide variety of UNIX platforms, Windows and MacOS, is mostly used for statistics but can also be used as a programming (script) language alone.

R is organized as a core distribution of base packages which can be extended by further packages loaded into the a user workspace (or interpreter global environment). Some useful links are

- Tutorials on using R can be found at `http://www.r-tutor.com/`
- Meta search and package documentation `https://www.rdocumentation.org/`
- R CRAN repository for contributed packages: `https://cran.r-project.org/`
- A short reference card `https://cran.r-project.org/doc/contrib/Short-refcard.pdf`

R Basics

```
R> PATH <- getwd()          # get working directory
R> INFO <- Sys.info()       # get system info
R> objects()                # show all loaded variables
```

```
[1] "INFO" "PATH"
```

```
R> ls()                     # objects in your workspace
```

```
[1] "INFO" "PATH"
```

Whats is in these objects?

```
R> PATH
```

```
[1] "/home/baaske/workspace/RIntro/doc"
```

```
R> INFO[c("sysname","nodename","user")]
```

```
     sysname           nodename               user
     "Linux" "baaskelap.rdm.de"           "baaske"
```

Important: On quitting, R offers the option of saving the workspace image, by default in the file "*.RData". Use before ending the R session:

```
R> rm(list=ls())
R> q()
```

R Help and vectors

Getting help:

```
R> help()                  # general help
R> ?length                 # help for `length`
R> help.search(lapply) # help for function `lapply`
R> help.start()            # start html help system
```

Vectors:

```
R> 2+2
[1] 4
R> round(pi,3)
[1] 3.142
R> sqrt(10)
[1] 3.162
R> 1000*(1+0.075)^5-1000
[1] 435.6
R> sin(c(30,60,90)*pi/180)
[1] 0.500 0.866 1.000
```

R variables and subsetting

```
R> a <- 2*3
R> a
[1] 6
R> a^2
[1] 36
R> b <- a^2
R> a <- c(17,1,3,9)
R> a
[1] 17  1  3  9
R> a[2]
[1] 1
R> a[c(1,3)]
[1] 17  3
R> a[-2]
[1] 17  3  9
R> a[2] <- 1
R> a
[1] 17  1  3  9
```

Characters and categories

```
R> (x <- "Hallo")                      # character vector
[1] "Hallo"
R> (y <- factor(c("C","A","C","B")))   # characters as categories
[1] C A C B
Levels: A B C
R> (z <- factor(c(1,1,2)))             # numbers as factors
[1] 1 1 2
Levels: 1 2
R> (x <- c(1,2,3))                     # distroy x and overwrite
[1] 1 2 3
R> x[4]                                # NA = Not Available
[1] NA
R> try(x[4])                           # catch error
[1] NA
```

R object classes

```
R> class(1.7) # "numeric"
[1] "numeric"
R> class(x)   # "character" = character vector
[1] "numeric"
R> class(y)   # "factor" categories
[1] "factor"
R> class(z)
[1] "factor"
R> mode(1.7)
[1] "numeric"
R> x <- as.integer(x)
R> class(x)
[1] "integer"
R> z <- as.character(z)
R> class(z)
[1] "character"
```

Characters and categories

```
R> # Save contents of workspace, into the file .RData
R> save.image()
R> # Save into the file archive.RData
R> save.image(file="archive.RData")
R> # save single objects
R> save(x, y,z, file="tmpobj.RData")
R> # save as RDS (could be big data)
R> saveRDS(list(x,y,z),file="myfile.rds")
R> # read as RDS
R> XYZ <- readRDS(file="myfile.rds")

R> # attach (reload) to current workspace
R> attach("tmpobj.RData")
R> ls()

[1] "a"    "b"    "INFO" "PATH" "x"    "y"    "z"
```

R vector repetitions

```
R> # vectors and repeating components
R> 10:5
[1] 10  9  8  7  6  5
R> -1:2
[1] -1  0  1  2
R> # a sequence
R> seq(5,10)
[1]  5  6  7  8  9 10
R> seq(5,10,by=0.1)
 [1]  5.0  5.1  5.2  5.3  5.4  5.5  5.6  5.7  5.8  5.9  6.0  6.1
[16]  6.5  6.6  6.7  6.8  6.9  7.0  7.1  7.2  7.3  7.4  7.5  7.6
[31]  8.0  8.1  8.2  8.3  8.4  8.5  8.6  8.7  8.8  8.9  9.0  9.1
[46]  9.5  9.6  9.7  9.8  9.9 10.0
R> # repeat
R> rep(1:3,times=3)
[1] 1 2 3 1 2 3 1 2 3
R> rep(1:3,each=3)
[1] 1 1 1 2 2 2 3 3 3
```

R vector repetitions

```
R> # replicate
R> x <- 1:5
R> y <- 3:1
R> # multiply elements
R> x*y
```

```
[1]  3  4  3 12 10
```

```
R> (M <- matrix(sample(1:10),nr=5))
```

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

```
R> as.numeric(M%*%c(2,2))
```

```
[1] 18 22 34 24 12
```

R data frame object

```
R> ?data.frame      # help on data frames
R> example(data.frame)    # some examples
```

Construct a data frame of study courses

```
R> g <- data.frame(StG=c("GTB","MPV","BGM"),Anz=c(75,11,62))
R> g
  StG Anz
1 GTB  75
2 MPV  11
3 BGM  62
R> class(g)    # "data.frame" = Datenmatrix
[1] "data.frame"
R> names(g)    # categories in g
[1] "StG" "Anz"
R> g[,2]             # 2nd column
[1] 75 11 62
R> g[3,]             # 3rd row
  StG Anz
3 BGM  62
R> g[3,2]             # single element
```

R data frame object

```
R> g[c(2,3),] # select 2nd and 3rd row
  StG Anz
2 MPV  11
3 BGM  62
R> g$Anz      # select category `Anz`
[1] 75 11 62
R> g$Anz[1]   # select first element of `Anz`
[1] 75
R> # Extending the  data frame
R> (g <- rbind(g,c("GTB",26)))
  StG Anz
1 GTB  75
2 MPV  11
3 BGM  62
4 GTB  26
```

R data frame object

```
R> # add category
R> (g <- cbind(g,Sem=c(3,1,3,5)))
  StG Anz Sem
1 GTB  75   3
2 MPV  11   1
3 BGM  62   3
4 GTB  26   5
R> # add factor level
R> (g <- rbind(g,data.frame(StG=factor("BGOK"),Anz=57,Sem=3)))
   StG Anz Sem
1  GTB  75   3
2  MPV  11   1
3  BGM  62   3
4  GTB  26   5
5 BGOK  57   3
```

R statistics

```
R> runif(n=10) # uniform on [0,1]

 [1] 0.20949 0.30704 0.61871 0.63192 0.09593 0.49546 0.38556 0.7
[10] 0.48235

R> rnorm(n=20,mean=5,sd=2)  # normal distribution

 [1] 8.487 5.968 4.237 6.551 9.362 4.210 5.725 3.062 9.080 5.094
[13] 4.835 1.563 2.030 5.365 4.805 6.113 5.957 7.456

R> rnorm(n=20)   # standard normal (mean=0, sd=1)

 [1] -0.70798  0.33993 -0.57505 -1.00112 -0.08003 -0.17630 -1.13
 [9] -0.11669  0.09420  0.38246 -1.39561  0.25327 -1.35219  0.95
[17] -2.95086 -0.22754  1.27911 -0.01134

R> rpois(n=20,lambda=5) # Poisson with parameter lambda

 [1] 3 3 4 5 2 7 5 7 4 7 5 6 4 3 6 5 7 8 7 4

R> rexp(n=20,rate=5) # exponential distribution

 [1] 0.509424 0.312712 0.349580 0.005942 0.129564 0.026428 0.026
 [9] 0.019561 0.505199 0.153799 0.026285 0.153606 0.300117 0.126
[17] 0.025016 0.104827 0.384372 0.074233
```

R statistical characteristics

```
R> x <- runif(100)
R> min(x)      # Minimum
[1] 0.01077
R> max(x)      # Maximum
[1] 0.9868
R> mean(x)     # Mean
[1] 0.5204
R> median(x) # Median
[1] 0.5335
R> var(x)      # variance
[1] 0.07077
R> sd(x)            # standard deviation/error
[1] 0.266
R> summary(x) # overview
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0108  0.3219  0.5335  0.5204  0.7300  0.9868
```

R loops

```
R> for(i in 7:9) {print(i)}
[1] 7
[1] 8
[1] 9
R> x <- 0
R> while(x < 3){print(x); x <- x+1}
[1] 0
[1] 1
[1] 2
R> fun <- function(x) x^2
R> fun(2.3)
[1] 5.29
```

Logical ifs

```
R> x <- sample(1:10,size=1)
R> if(x < 5) {print("Yeah!")} else {print("Oh, noo!")}
[1] "Oh, noo!"
R> print(x)
[1] 5
```

R Practice

1. Generate standard normal random variables, check characteristics and use different sample sizes!
2. Construct a numeric vector and sort this in ascending order using function *sort()*.
3. Delete first row of data frame ‘g‘. Add some new category. Change number of students ‘Anz‘ in in ‘MPV‘.
4. Load data set *airquality*:

   ```
   R> data(airquality)
   R> head(airquality)
   ```

   Check for ‘NAs‘ and omit these by functions *is.na()* and *na.omit()* What is it about? Write a function which calculates the means and standard deviations of categories ‘Ozone‘,‘Solar.R‘, ‘Wind‘ and ‘Temp‘ from data set *airquality* for each month separately. THe function returns a matrix of dimensions 5×2 where the first columnstores the means and the second the standard errors of all month.