Introduction to R

Øystein H. Opedal

24 June 2019

What is R

R is a software package for statistical computing, freely available under a General Public License. It exists for most major operating systems (Windows, Unix, Linux, Mac). In ecology and evolutionary biology, R is by far the most widely used package for performing statistical analyses, performing computations, and producing graphics. It is designed to be very flexible, with a limited set of base functions, and a nearly unlimited number of 'packages' tailored for specific tasks.

Useful R help websites:

http://www.personality-project.org/r/r.commands.html

http://www.statmethods.net/

Where to find R and RStudio

R is freely available from

http://www.r-project.org/

Several programs provide a nicer interface and make R more user-friendly. My preferred one is RStudio:

http://www.rstudio.com/

Basic operations

One of the simplest possible tasks in R is to enter an arithmetic expression and receive a result. For a first impression of what R can do, type the following in the console:

```
2 + 2
```

```
## [1] 4
```

The second value (i.e., the `4`) is the result. The first value (i.e., the `[1]`) indicates its position in the results vector.

R also deals with symbolic variables, which are names that represent values. If you type:

```
x <- 2
```

There is no immediate result, but from now on, `x` will have the value of 2 and you can use it in subsequent expressions:

```
x
```

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

```
x + x
```

```
## [1] 4
```

R reads the characters "<-" as a single symbol that points to the variable to which it will assign the value. Names of variables can be chosen quite freely in R. They can include letters, digits, underscores, and the period (dot) symbol. There is, however, the limitation that the name must not start with a digit or a period followed by a digit. Names are case-sensitive (`WT` is thus different from `wt`).

Vectors can also be assigned to objects. The normal `()` brackets are used for function calls, here to the function `c` (for concatenate).

```
x <- c(1:5, 10:15)
```

Square brackets `[]` are used to index vectors, matrices, arrays, and dataframes.

```
x
```

```
##  [1]  1  2  3  4  5 10 11 12 13 14 15
```

```
x[1:3]
```

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

We can then perform vectorized operations

```
x*2
```

```
##  [1]  2  4  6  8 10 20 22 24 26 28 30
```

Sampling from statistical distributions

To demonstrate some simple operations, let us simulate some data from the normal distrubution, using the `rnorm` function. We set the seed of the random number generator to make the results reproducible.

```
set.seed(1)
x <- rnorm(n = 200, mean = 10, sd = 2)
y <- 1.3*x + rnorm(200, 0, 2) #R knows the order of arguments
```

Getting help

We can access help files for any R function by typing `?function`

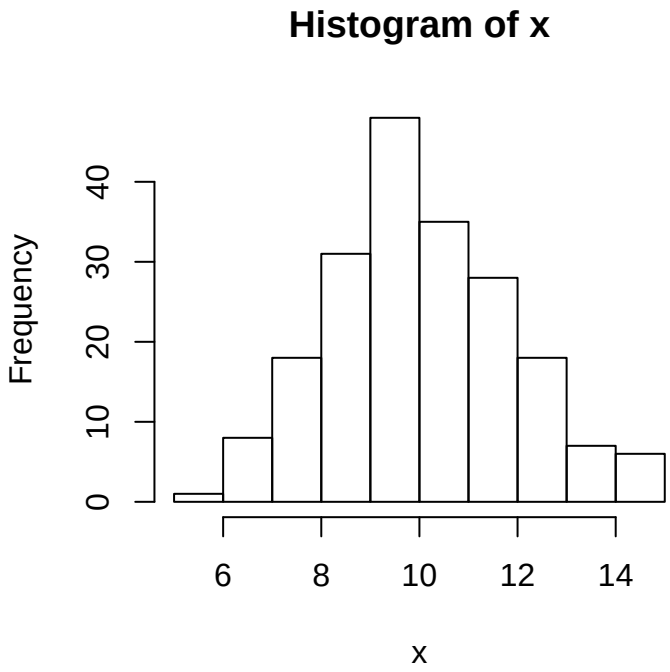
```
?rnorm
```

```
## starting httpd help server ... done
```

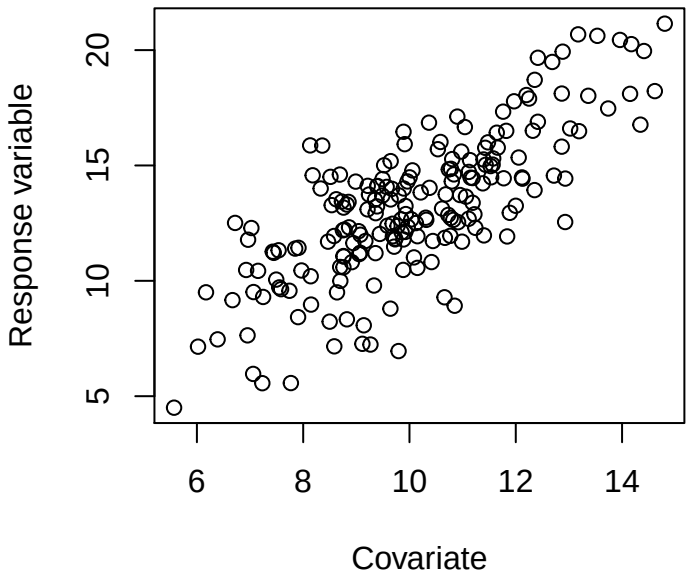
Graphics

A key advantage of R is its flexible and powerful tools for producing publication-quality graphics

```
hist(x)
```




```
plot(x, y, xlab = "Covariate", ylab = "Response variable")
```



Statistical analysis

R comes with built-in functions for mainstream statistical analyses such as t-tests, correlation tests, and linear models.

```
t.test(x, y)
```

```
##
##  Welch Two Sample t-test
##
## data:  x and y
## t = -12.092, df = 324.6, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.607379 -2.597820
## sample estimates:
## mean of x mean of y
##  10.07108  13.17368
```

```
cor.test(x, y)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x and y
## t = 16.521, df = 198, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6960689 0.8140491
## sample estimates:
##       cor
## 0.7612897
```

```
mod <- lm(y ~ x)
summary(mod)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8655 -1.0855 -0.0505  1.3367  5.1723
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.31369    0.79148   0.396    0.692
## x            1.27692    0.07729  16.521   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.026 on 198 degrees of freedom
## Multiple R-squared:  0.5796, Adjusted R-squared:  0.5774
## F-statistic: 272.9 on 1 and 198 DF,  p-value: < 2.2e-16
```

Programming

R can do most basic programming operations. Note the use of the `{}` brackets in functions and loops.

```
out <- NULL
for(i in 1:length(x)){
  if(x[i]>y[i]){
    out[i] <- 1
  }
  else{
    out[i] <- 0
  }
}

sum(out)
```

```
## [1] 15
```

R Packages

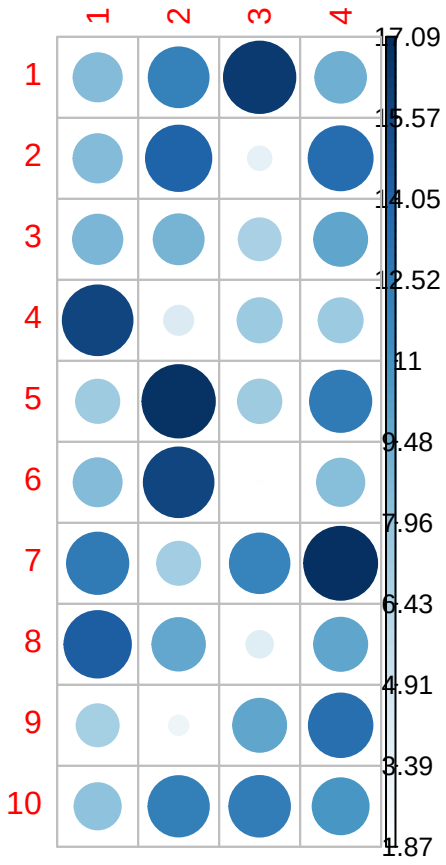
While base R includes many standard functions as demonstrated above, more specialized tasks require installing additional 'packages'.

```
#install.packages("corrplot")

mat = matrix(rnorm(10*4, 10, 4), nrow=10)

library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.5.3
```

```
## corrplot 0.84 loaded
```

```
corrplot(mat, is.cor = FALSE)
```


1
2
3
4
5
6
7
8
9
10
17.09
15.57
14.05
12.52
11
9.48
7.96
6.43
4.91
3.39
1.87