R Tutorial

Simple operations and variable assignments

Erik Hintz

erik.hintz@uwaterloo.ca
Department of Statistics and Actuarial Science
University of Waterloo

Agenda

- Defining vectors
- Working with vectors (sums, products, changing a value, ...)
- Defining matrices
- Working with matrices (sums, products, ...)
- Basic functions (`exp()`, `log()`,...)
- Basic plotting

Defining vectors

- We assign, to a variable `a`, a value `val` via `a <- val`
- The function `c()` (for "concatenate") can be used to build vectors, e.g., via `vec <- c(1, 2, 4.5)`
- Useful is also the shorthand `vec <- a:b`, which is equivalent to `vec <- c(a, a+1, ....., b-1, b)`.
- See `?seq` for another function to build sequences of numbers, e.g., `vec <- seq(from = 1, to = 10, by = 0.1)`
- We can print all elements of an object by merely typing its name, e.g., `vec`.
- The *i*th element of a vector can be accessed as `vec[i]`, e.g., `vec[3] <- -1/2` or `vec[1:2]`

Working with vectors

- The length of `vec` is `length(vec)`
- Vectors can be added, e.g. `vec1 + vec2` gives the componentwise sum
- Similarly, `vec1 * vec2` gives the componentwise product
- When the lengths don't match, Rwill repeat the shorter object so often such that its length matches the longer object.
  Thus, be careful!
- We can apply functions to vectors, e.g., `exp(vec)`, `10^ vec`, `sqrt(vec) + log(vec)`
- There are other useful operations, e.g., `length(vec)`, `rev(vec)`, `sum(vec)`, `mean(vec)`

Exercise 1.1

a) Define a vector `a` with the natural numbers between 1 and 10, and a vector `b` with the numbers $10^{-1}, 10^{-2}, \ldots, 10^{-10}$.

b) Store the result of `a*b` in a vector `c`.

c) Change every other element in `c` to 0, starting with the first one.

d) Print the result of `a+c`

Defining and working with matrices

- The function `matrix()` can be used to define matrices; see `?matrix`.
- By default, matrices are filled column-wise.
- Example: `M <- matrix( 1:4, ncol = 2)` gives $\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$
- We can add and multiply elementwise via `+` and `*`
- The matrix product is obtained via `%*%` and the inverse of `M` is `solve(M)`, if existent.
- We can apply functions to matrices, such as `exp(mat)`, `log(mat)`,...
- Other useful operations include `sum(mat)`, `colSums(mat)`, `rowSums(mat)`, `t(mat)`,... as well as `diag()`.

Exercise 1.2

a) Define the matrix with entries $\begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$ and call it `M`

b) Define `N` as the componentwise product of `M`

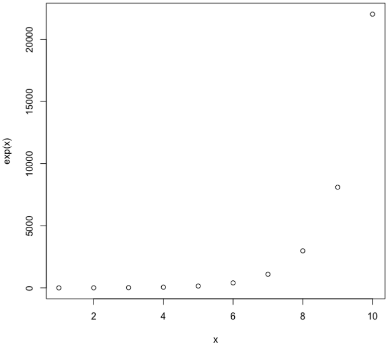
c) Compute the matrix inverse of `N`.

d) Verify that the result in c) is indeed the inverse.

Basic functions and basic plotting

- R comes with tons of useful standard functions, e.g.,
  - `exp()`, `sqrt()`, `log()`, `binom()`, `factorial`
- We can also compare elements via `==`
- Plotting is provided via the function `plot()` and lines/points can be added with `lines()` and `points()`, e.g.,

```
plot(1:25, exp(1:25), xlab = "x", ylab = "exp(x)")
```



Exercise 1.3

In this exercise, we will create a plot illustrating Stirlings formula, which was

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n, \quad n \to \infty.$$

Create a plot which shows the function $n!$ and the function $\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$, using two colours, for $n \in \{1, 2, \ldots, 40\}$. How can you improve this plot?

Hint: type `?plot`