

Chapter 3

Getting help in R

3.1 Help in R itself

- Open an html search page using `help.start()`
- If you *do not know* the function name, try, for example, `RSiteSearch("regression")`. This function searches for key words or phrases in the R-help mailing list archives, or R manuals and help pages, using the search engine at <http://search.r-project.org> and enables them to be viewed in a web browser.
- If you *do not know* the function name, another way is to try, for example, `help.search("regression")`.
- If you *do know* the function name, try, for example, `help("lm")` or `?lm`
- You can also try using `example`; for example, `example(lm)`.
- Demonstration of R capabilities are given using `demo()`; for example, `demo(graphics)`. Type `demo()` for a list of available demonstrations.

3.2 Help outside R

- You can search the web pages:
 - The R Project at <http://www.r-project.org/>
 - The Comprehensive R Archive Network (CRAN) at <http://cran.au.r-project.org/>
 - Search the mailing list of three R groups: <http://cran.au.r-project.org/search.html>

- The library has books on R. (Most books about S-Plus are also relevant; see Section 1.2). Some useful books include (in subjective rank order from very useful to useful):
 - Dalgaard, Peter. *Introductory statistics with R*, New York: Springer, 2002. (519.5 Dal)
 - Maindonald, John and Braun, John. *Data Analysis and Graphics Using R: An Example-based Approach*, Cambridge University Press, 2003. (519.50285 Mai)
 - Krause, Andreas and Olson, Melvin. *The basics of S and S-Plus*, New York: Springer, 1997. (519.50285 Spl/Kra). This book is also available as an e-book through the library catalogue.
 - Venables, W. N. and Ripley, B. D. *Modern applied statistics with S-PLUS*, New York: Springer, 1999. (519.5028553 Ven)

3.3 Using R functions

3.3.1 An initial look

Suppose we wish to create a sequence (like $-1, 0, 1, 2, \dots$). To find such a function, type

```
help.search("seq")
```

`seq` appears to be the appropriate function.

Part of an example help page for `seq`, obtained by typing `?seq` at the R prompt, is shown in Figure 3.1.

Here is a brief explanation:

- Line 5: A brief description of what the function does
- Lines 9–18: How to use the function.
- Lines 20–32: The argument to the function, and what they mean.
- Lines 38–42: Other function that might be relevant.
- Lines 44–63: Some examples of using the function. When the command `example(seq)` is given, these commands are executed. Sometimes, these example are for non-beginners.

We learn that `seq` is for generating regular sequences, like: 0, 1, 2, 3, 4, 5. Consider the following:

```
1  seq                package:base                R Documentation
2
3  Sequence Generation
4
5  Description:
6
7      Generate regular sequences.
8
9  Usage:
10
11      from:to
12          a:b
13
14      seq(from, to)
15      seq(from, to, by= )
16      seq(from, to, length.out= )
17      seq(along.with= )
18      seq(from)
19
20 Arguments:
21
22     from: starting value of sequence.
23
24     to: (maximal) end value of the sequence.
25
26     by: increment of the sequence.
27
28 length.out: desired length of the sequence.
29
30 along.with: take the length from the length of this argument.
31
32     a,b: 'factor's of same length.
33
34 ---      <snip> ---
35
36 See Also:
37
38     The method 'seq.POSIXt'.
39
40     'rep', 'sequence', 'row', 'col'.
41
42     As an alternative to using ':' for factors, 'interaction'.
43
44 Examples:
45
46     1:4
47     pi:6 # float
48     6:pi # integer
49
50     seq(0,1, length=11)
51     seq(rnorm(20))
52     seq(1,9, by = 2) # match
53     seq(1,9, by = pi)# stay below
54     seq(1,6, by = 3)
55     seq(1.575, 5.125, by=0.05)
56     seq(17) # same as 1:17
57
58     <snip>
```

- Generally, the input arguments are *named*:

```
> seq(from = 0, to = 5, by = 1)
```

```
[1] 0 1 2 3 4 5
```

- If the arguments appear in the order given, they need not be named:

```
> seq(0, 5, by = 1)
```

```
[1] 0 1 2 3 4 5
```

- If the arguments are named, they can appear in *any* order:

```
> seq(to = 5, by = 1, from = 0)
```

```
[1] 0 1 2 3 4 5
```

- A variation is:

```
> seq(0, 5, length = 6)
```

```
[1] 0 1 2 3 4 5
```

- Since this quite a common command, we can also use:

```
> 0:5
```

```
[1] 0 1 2 3 4 5
```

3.3.2 Another look

Consider the help for the R function `rnorm`, for generating random numbers from a Normal distribution. This help page also discusses the help for other functions related to the Normal distribution; see Figure 3.2.

Observe the following¹:

- To generate two random numbers from $N(0, 1)$, use:

```
> rnorm(n = 2, mean = 0, sd = 1)
```

```
[1] -0.549638 1.091054
```

¹Note that, since random numbers are used, each time the function is run produces a different value, and you will see different values if you type these commands in yourself. If you want to set the random number seed (and so have reproducible ‘random’ numbers), use `set.seed()`, and place some integer in the parentheses; for example `set.seed(518768)`.

```
1 Normal                package:stats                R Documentation
2
3 The Normal Distribution
4
5 Description:
6
7     Density, distribution function, quantile function and random
8     generation for the normal distribution with mean equal to 'mean'
9     and standard deviation equal to 'sd'.
10
11 Usage:
12
13     dnorm(x, mean=0, sd=1, log = FALSE)
14     pnorm(q, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
15     qnorm(p, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
16     rnorm(n, mean=0, sd=1)
17
18 Arguments:
19
20     x,q: vector of quantiles.
21
22     p: vector of probabilities.
23
24     n: number of observations. If 'length(n) > 1', the length is
25     taken to be the number required.
26
27     mean: vector of means.
28
29     sd: vector of standard deviations.
30
31 log, log.p: logical; if TRUE, probabilities p are given as log(p).
32
33 lower.tail: logical; if TRUE (default), probabilities are P[X <= x],
34     otherwise, P[X > x].
35
36 Details:
37
38     If 'mean' or 'sd' are not specified they assume the default values
39     of '0' and '1', respectively.
40
41 --- <snip> ---
42 Value:
43
44     'dnorm' gives the density, 'pnorm' gives the distribution
45     function, 'qnorm' gives the quantile function, and 'rnorm'
46     generates random deviates.
47
```

Figure 3.2: Part of the help for the function rnorm

- In the definition, the default values for the mean and standard deviation are specified as 0 and 1 (see Line 16) respectively; so we could use:

```
> rnorm(n = 2)

[1] 0.6397777 1.0425760
```

- Since arguments need not be named if given in the specified order, we could use:

```
> rnorm(2)

[1] 0.1697033 1.1378004
```

- We could, of course, type:

```
> rnorm(mean = 0, n = 2, sd = 1)

[1] -0.9705545 -0.1318273
```

If we produce a lot of random numbers you see how R displays long vectors:

```
> rnorm(30)

[1] 0.14622623 1.44129222 -2.94138513 -0.24283860 -0.14058509
[6] -0.03265396 0.27981588 0.59014791 1.02429531 2.10732266
[11] 0.15463424 0.91306984 -0.25422449 1.51943184 1.78126982
[16] -0.87884444 -1.52921818 0.13591141 -0.70889985 -1.40957238
[21] 1.83093656 1.29019398 -2.36243538 -0.55097480 -0.30466297
[26] -0.74996129 0.14372175 -0.54943140 0.15992394 -0.08767391
```

The numbers in brackets on the far-left indicate the element number in the vector.