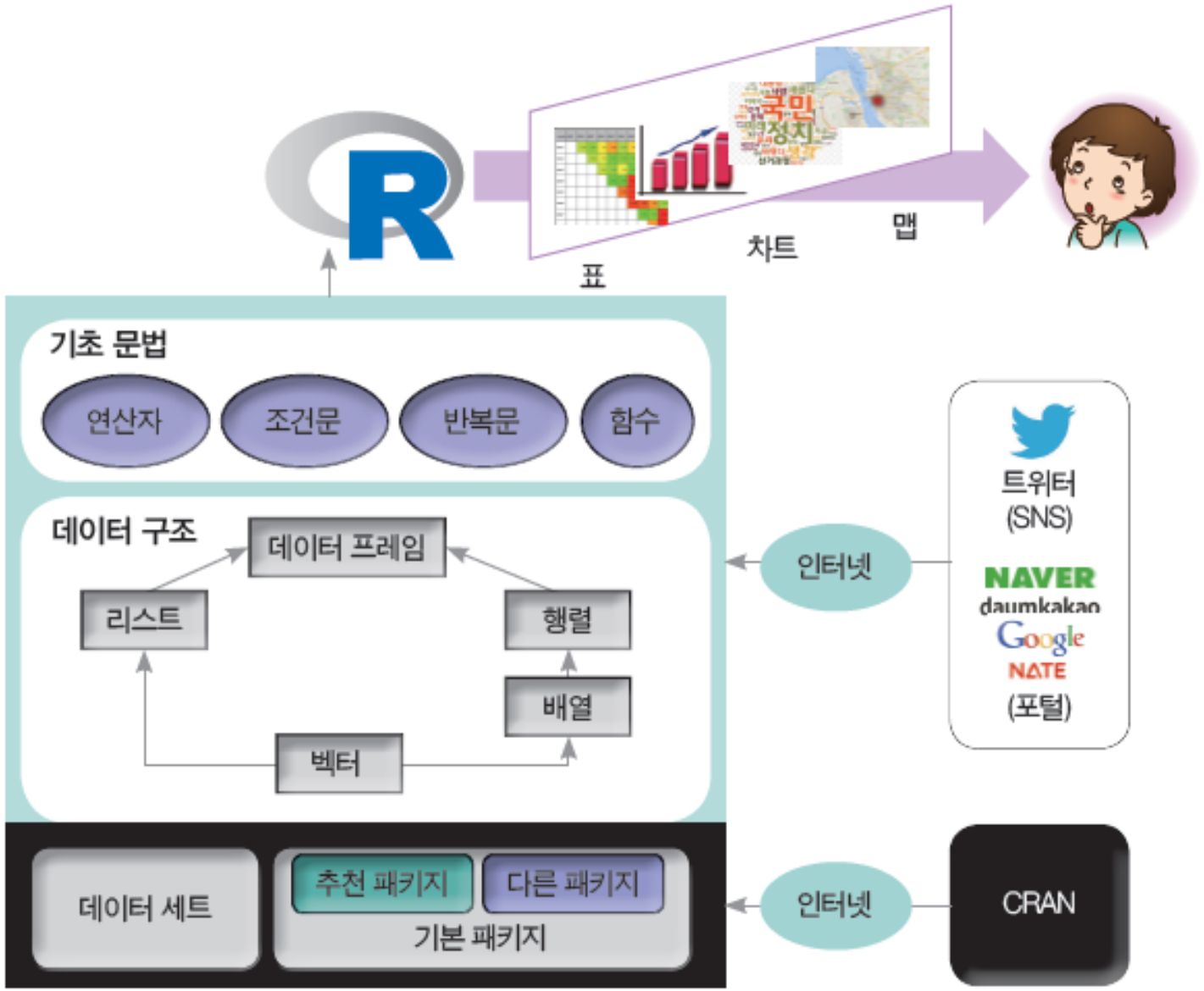
Getting Started

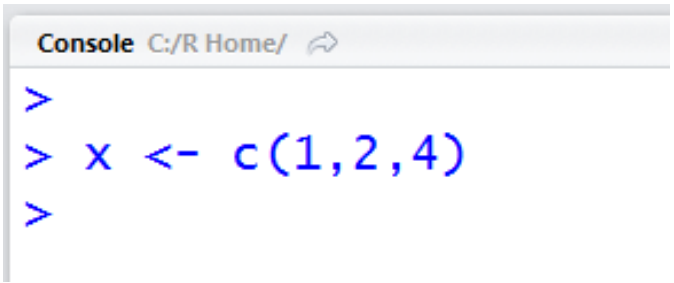
Bok, Jong Soon
javaexpert@nate.com
https://github.com/swacademy/R

R Usage Concepts



A First R Session

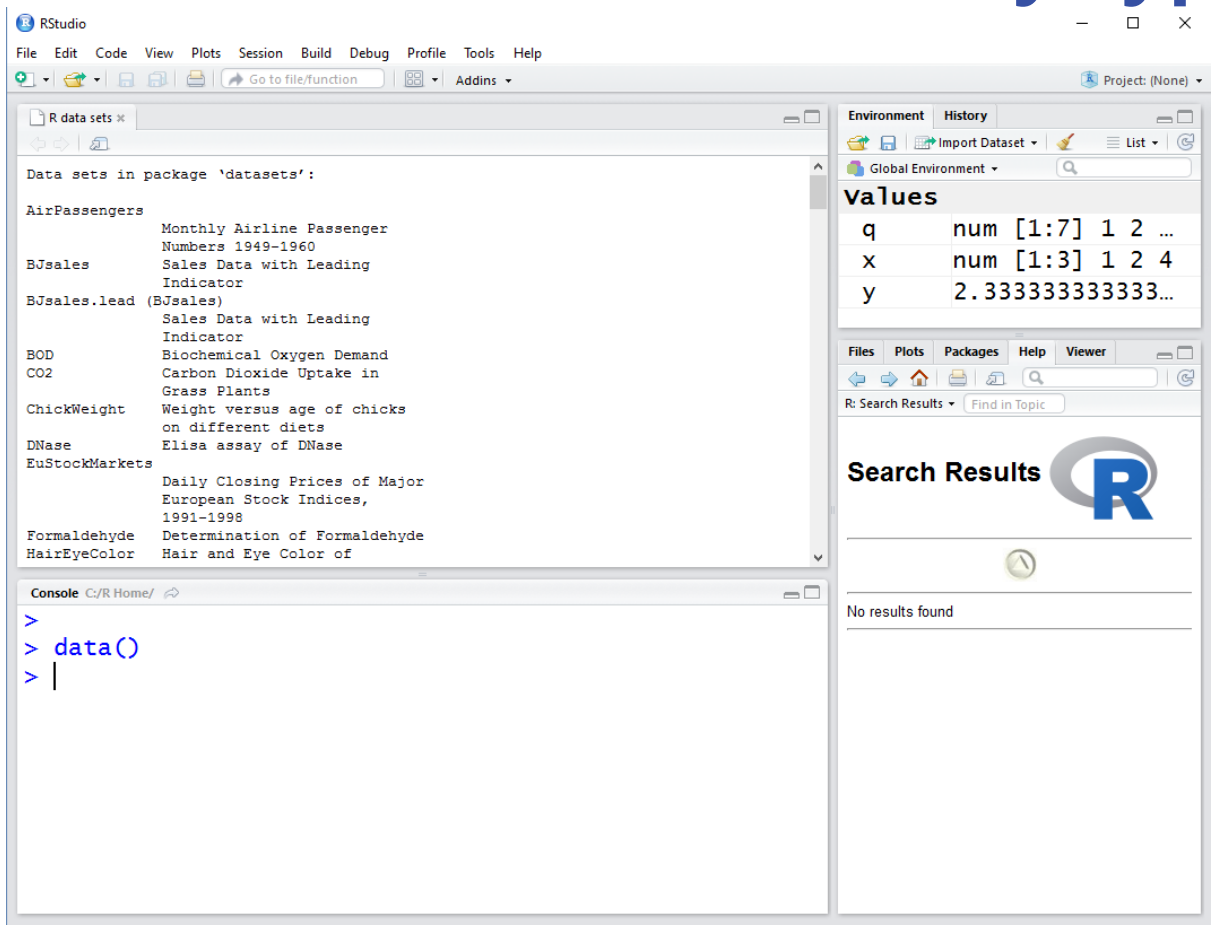
- Let's make a simple data set (in R parlance, a **vector**) consisting of the numbers **1**, **2**, and **4**, and name it **x**:



- The standard assignment operator in R is **<-**.
- You can also use **=**, but this is discouraged, as it does not work in some special situations.
- Note that there are no fixed types associated with variables.
- The **c** stands for concatenate.

A First R Session (Cont.)

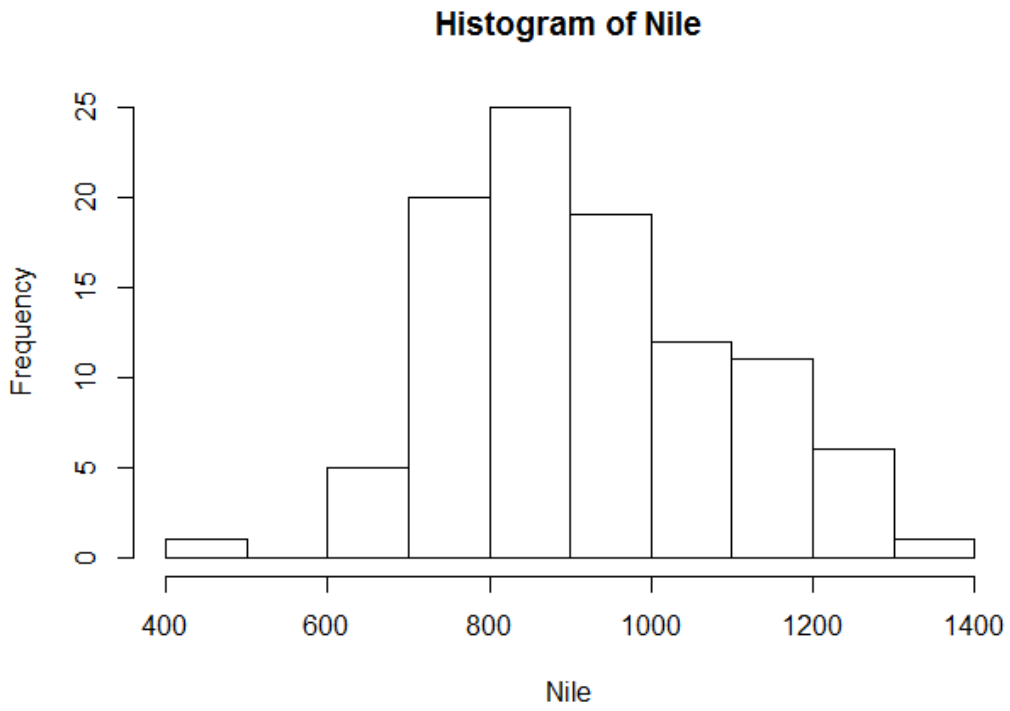
- Let's do something with one of R's internal data sets.
- You can get a list of these data sets by typing the following:



A First R Session (Cont.)

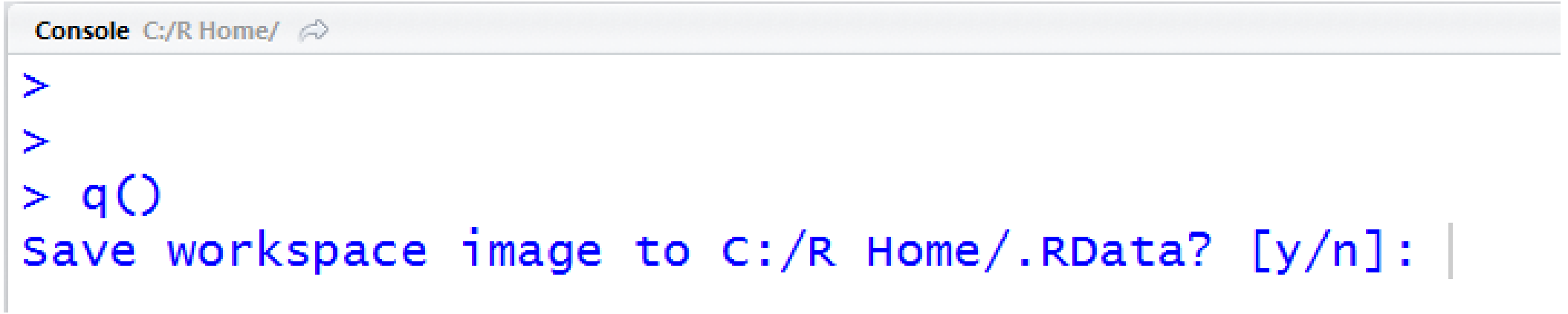
- One of the data sets is called **Nile** and contains data on the flow of the Nile River.
- Let's find the mean and standard deviation of this data set:

```
>
> mean(Nile)
[1] 919.35
>
> sd(Nile)
[1] 169.2275
>
> hist(Nile)
>
```



A First R Session (Cont.)

- Quit R by calling the `q()` function (or alternatively by pressing `CTRL-D` in Linux or `CMD-D` on a Mac):



In other hand...

Command Prompt

```
C:₩R Home>copy con test.R
# My first program in R Programming
myString <- "Hello, World!"

print( myString )
^Z
        1 file(s) copied.

C:₩R Home>"C:₩Program Files₩R₩R-3.4.1₩bin₩RScript.exe" test.R
[1] "Hello, World!"

C:₩R Home>_
```

Introduction to Functions

- As in most programming languages, the heart of R programming consists of writing functions.
- A function is a group of instructions.

```
>
> # counts the number of odd intergers in x
> oddcount <- function(x){
+     k <- 0    # assign 0 to k
+     for(n in x) {
+         if(n %% 2 == 1) k <- k + 1   #%% is the modulo operator
+     }
+     return(k)
+ }
>
> oddcount(c(1,3,5))
[1] 3
>
> oddcount(c(1,2,3,7,9))
[1] 4
>
```

Variable Scope

- A variable that is visible only within a function body is said to be *local* to that function.
- In `oddcount()`, `k` and `n` are *local* variables.

```
>
> oddcount(c(1,2,3,7,9))
[1] 4
>
> n
Error: object 'n' not found
>
>
> z <- c(2,6,7)
> oddcount(z)
[1] 1
>
```

Variable Scope (Cont.)

- Variables created outside functions are *global* and are available within functions as well.

```
>
> f <- function(x) return(x + y)
> y <- 3
> f(5)
[1] 8
>
```

- Here `y` is a *global* variable.

Some Important R Data Structures

- R has a variety of data structures.
- We will sketch some of the most frequently used structures.
- Scalars
- Character Strings
- Matrices
- Lists
- Data Frames
- Classes

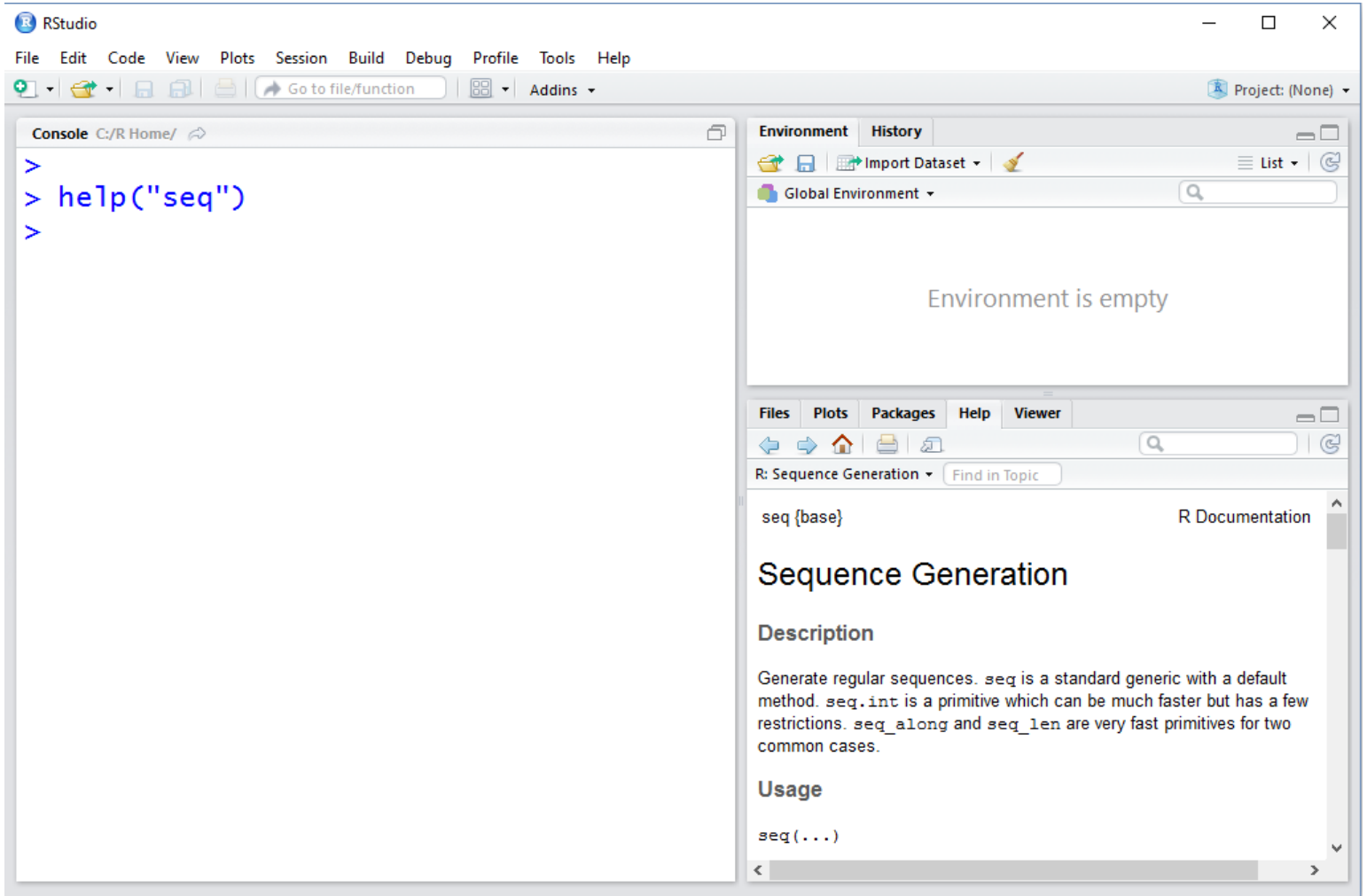
Comment in R

```
1
2  # 2017년 7월 6일
3
4  x <- 9     #x라는 변수에 9라는 정수형 스칼라값을 할당
5  y <- 5     #y라는 변수에 5라는 정수형 스칼라값을 할당
6
7  x          #x변수의 값 출력
8  y          #y변수의 값 출력
9
10
```

Getting Help

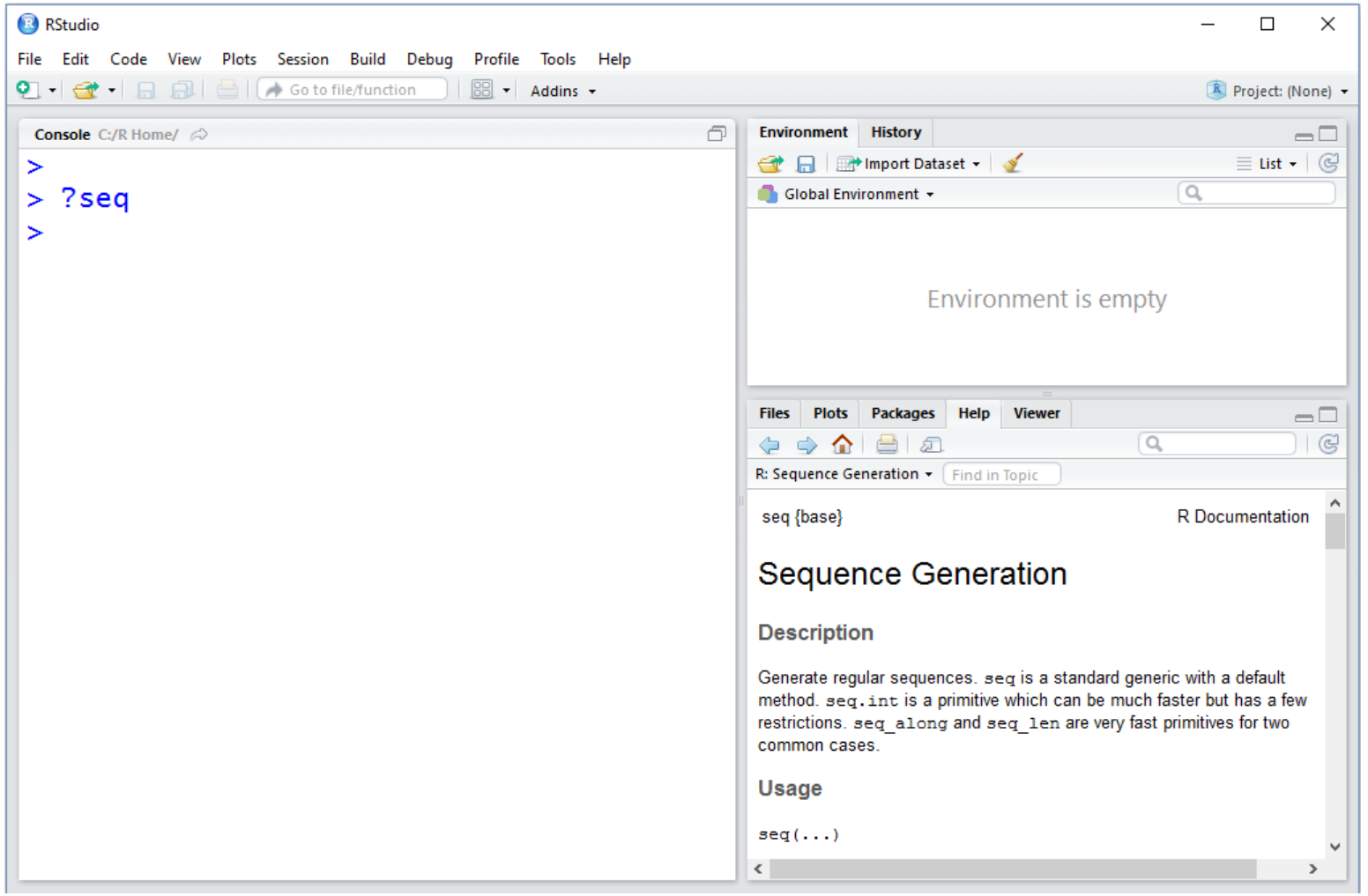
The help() Function

- To get online help, invoke **`help()`**.



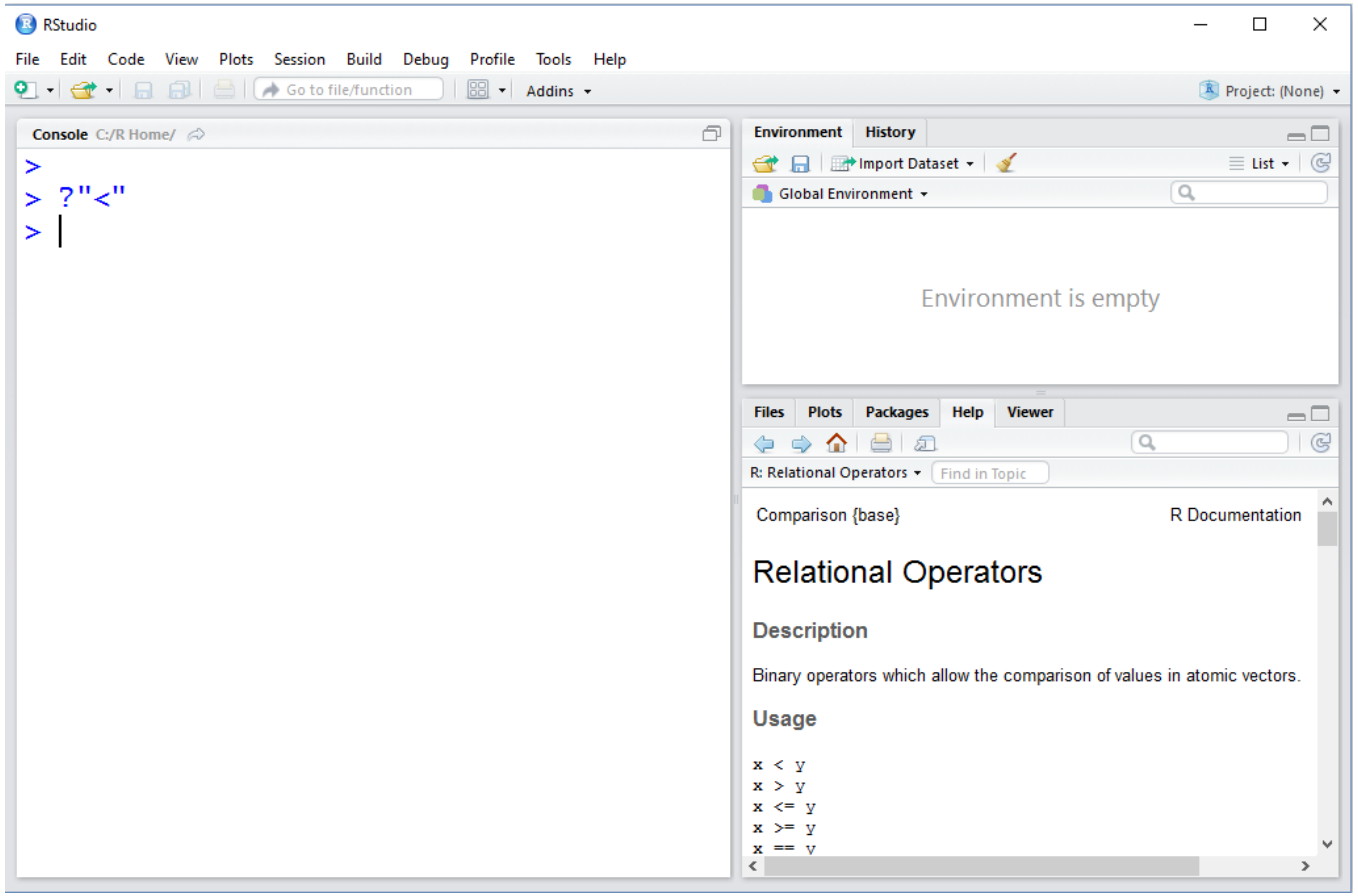
The help() Function

- The shortcut to help() is a question mark (?).



The help() Function

- Special characters and some reserved words must be quoted(‘’) when used with the help() function.



The example() Function

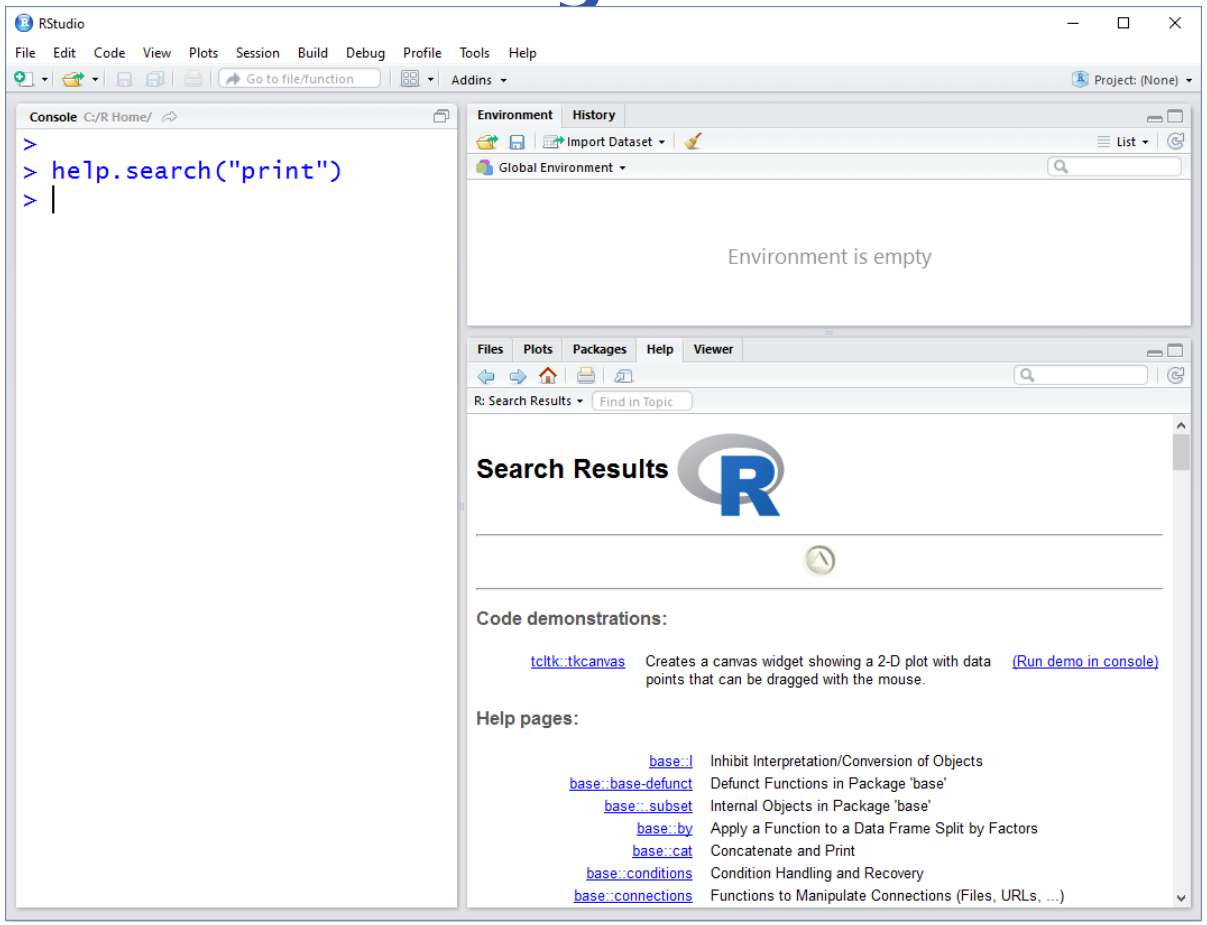
- Each of the help entries comes with examples.
- One really nice feature of R is that the **example()** function will actually run those examples for you.

Console C:/R Home/

```
>
> example("seq")

seq> seq(0, 1, length.out = 11)
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7
 [9] 0.8 0.9 1.0

seq> seq(stats::rnorm(20)) # effectively 'along'
 [1]  1  2  3  4  5  6  7  8  9 10 11
[12] 12 13 14 15 16 17 18 19 20

seq> seq(1, 9, by = 2)     # matches 'end'
[1] 1 3 5 7 9

seq> seq(1, 9, by = pi)    # stays below 'end'
[1] 1.000000 4.141593 7.283185

seq> seq(1, 6, by = 3)
[1] 1 4

seq> seq(1.575, 5.125, by = 0.05)
 [1] 1.575 1.625 1.675 1.725 1.775
 [6] 1.825 1.875 1.925 1.975 2.025
[11] 2.075 2.125 2.175 2.225 2.275
[16] 2.325 2.375 2.425 2.475 2.525
[21] 2.575 2.625 2.675 2.725 2.775
[26] 2.825 2.875 2.925 2.975 3.025
[31] 3.075 3.125 3.175 3.225 3.275
```

If You Don't Know Quite What You're Looking For

- You can use the function `help.search()` to do a Google-style search through R's documentation.



Help on the Internet

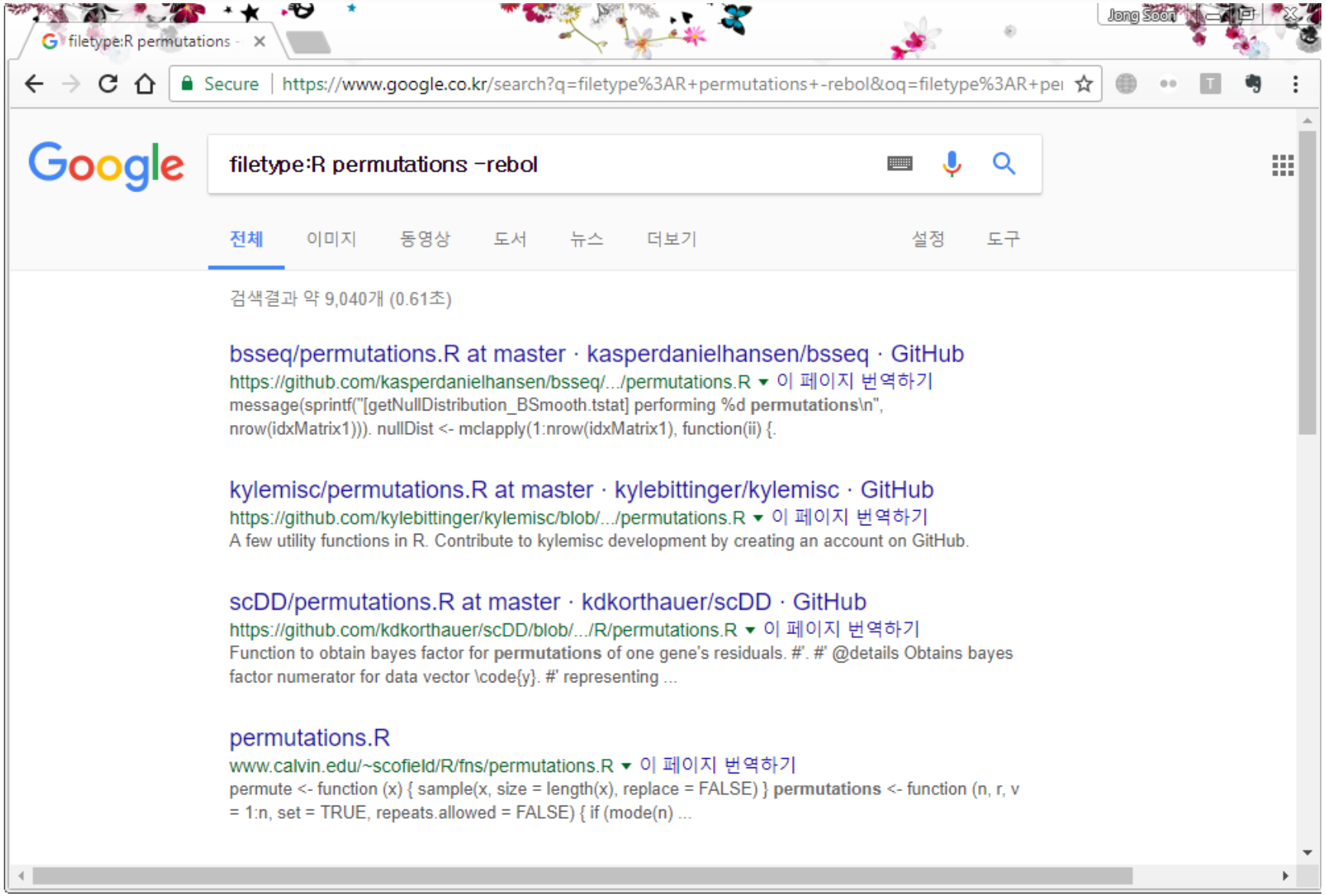
- **The R Project's own manuals are available from the R home page, http://www.r-project.org/. → Click `Manuals`.**
- **Various R search engines are listed on the R home page. → Click `Search`.**
- **The `sos` package offers highly sophisticated searching of R materials.**
  - See Appendix B for instructions on how to install R packages.
- **RSeek search engine : http://www.rseek.org/.**

Help on the Internet (Cont.)

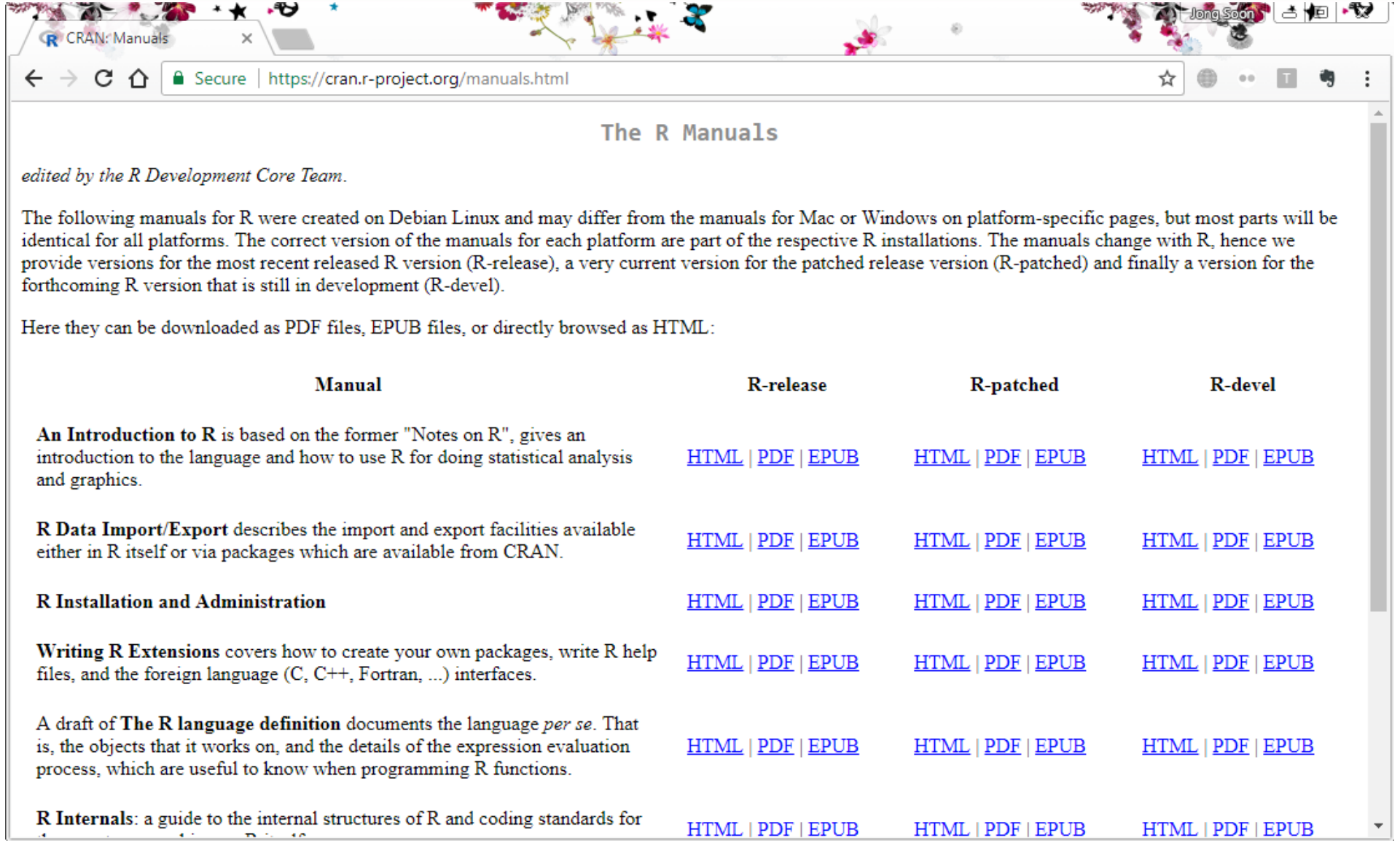
- Because of its single-letter name, R is difficult to search for using general purpose search engines such as Google.
- But there are tricks you can employ.
- One approach is to use Google's `filetype` criterion.

```
filetype:R permutations -rebol
```

Help on the Internet (Cont.)



Help on the Internet (Cont.)



The R Manuals

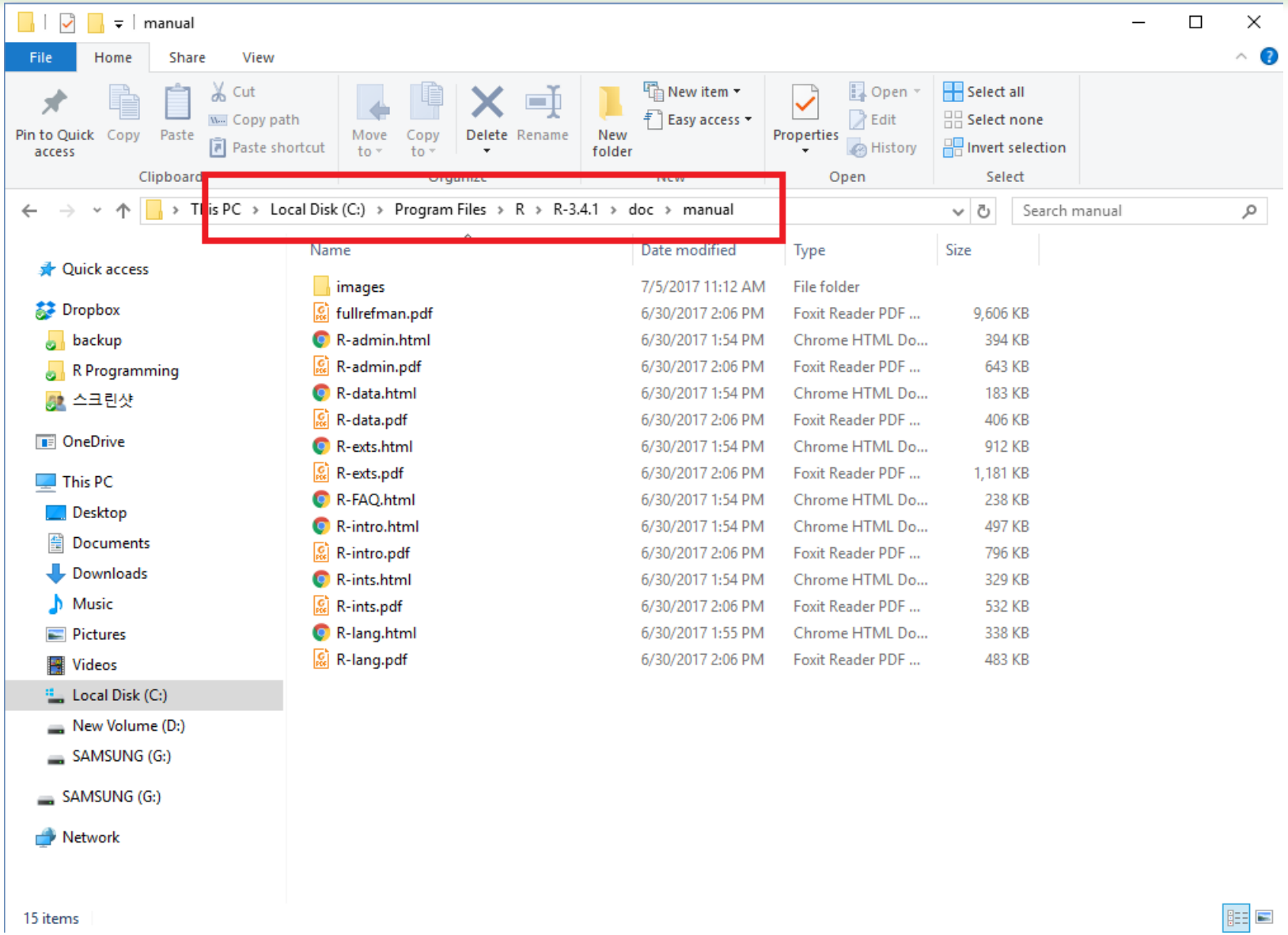
edited by the R Development Core Team.

The following manuals for R were created on Debian Linux and may differ from the manuals for Mac or Windows on platform-specific pages, but most parts will be identical for all platforms. The correct version of the manuals for each platform are part of the respective R installations. The manuals change with R, hence we provide versions for the most recent released R version (R-release), a very current version for the patched release version (R-patched) and finally a version for the forthcoming R version that is still in development (R-devel).

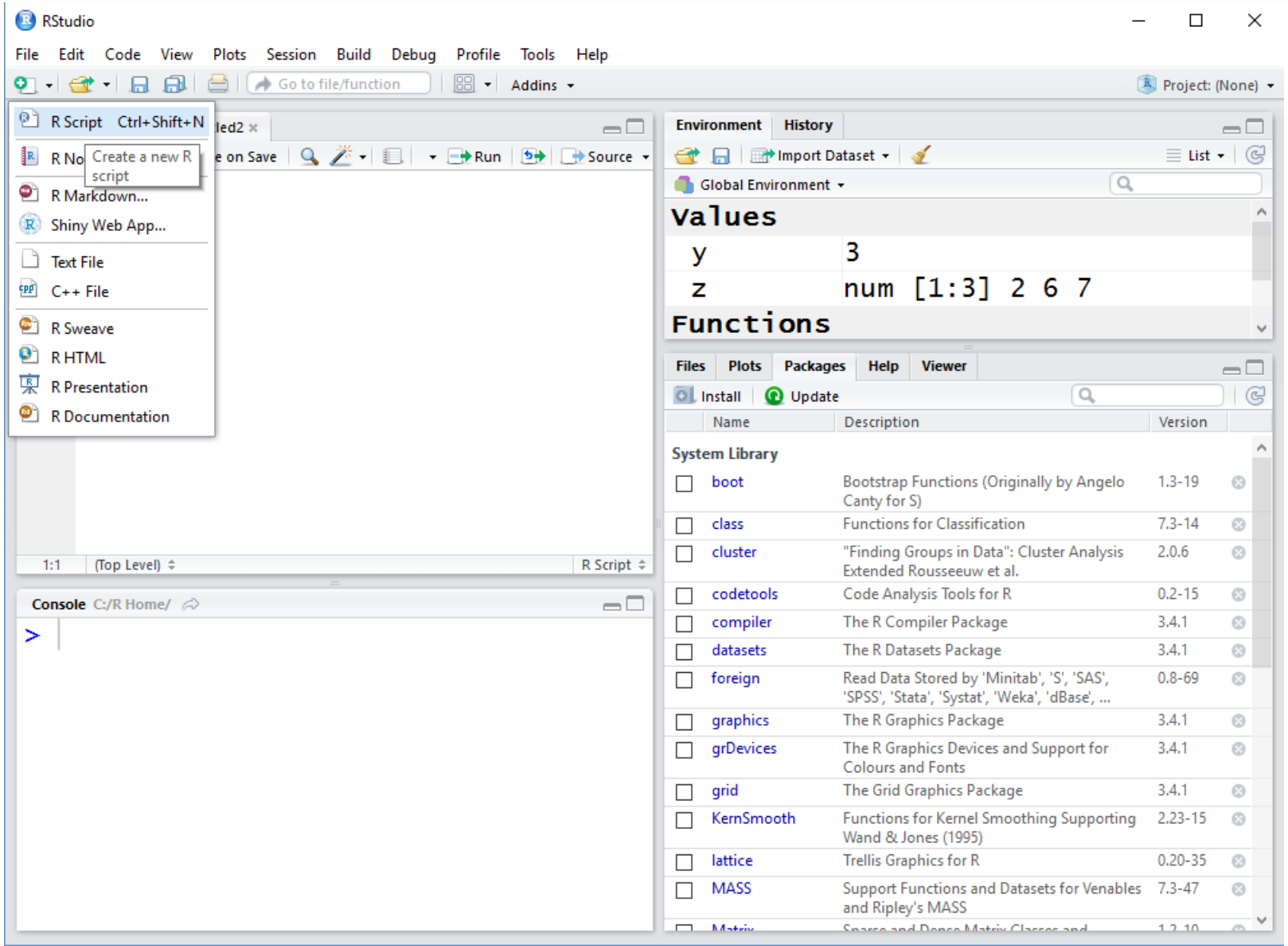
Here they can be downloaded as PDF files, EPUB files, or directly browsed as HTML:

Manual	R-release	R-patched	R-devel
An Introduction to R is based on the former "Notes on R", gives an introduction to the language and how to use R for doing statistical analysis and graphics.	HTML | PDF | EPUB	HTML | PDF | EPUB	HTML | PDF | EPUB
R Data Import/Export describes the import and export facilities available either in R itself or via packages which are available from CRAN.	HTML | PDF | EPUB	HTML | PDF | EPUB	HTML | PDF | EPUB
R Installation and Administration	HTML | PDF | EPUB	HTML | PDF | EPUB	HTML | PDF | EPUB
Writing R Extensions covers how to create your own packages, write R help files, and the foreign language (C, C++, Fortran, ...) interfaces.	HTML | PDF | EPUB	HTML | PDF | EPUB	HTML | PDF | EPUB
A draft of **The R language definition** documents the language *per se*. That is, the objects that it works on, and the details of the expression evaluation process, which are useful to know when programming R functions.	HTML | PDF | EPUB	HTML | PDF | EPUB	HTML | PDF | EPUB
R Internals: a guide to the internal structures of R and coding standards for	HTML | PDF | EPUB	HTML | PDF | EPUB	HTML | PDF | EPUB

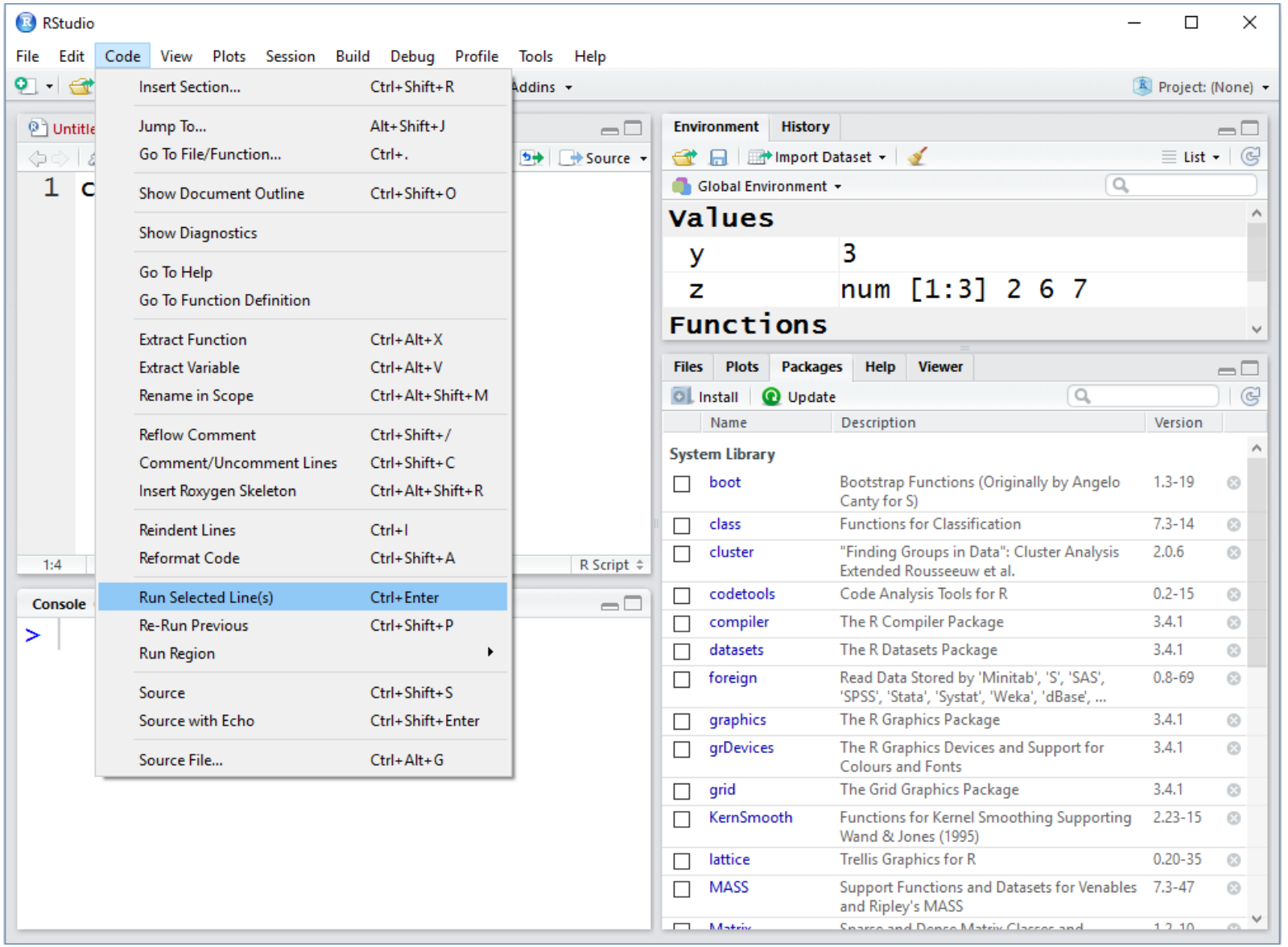
Local Document



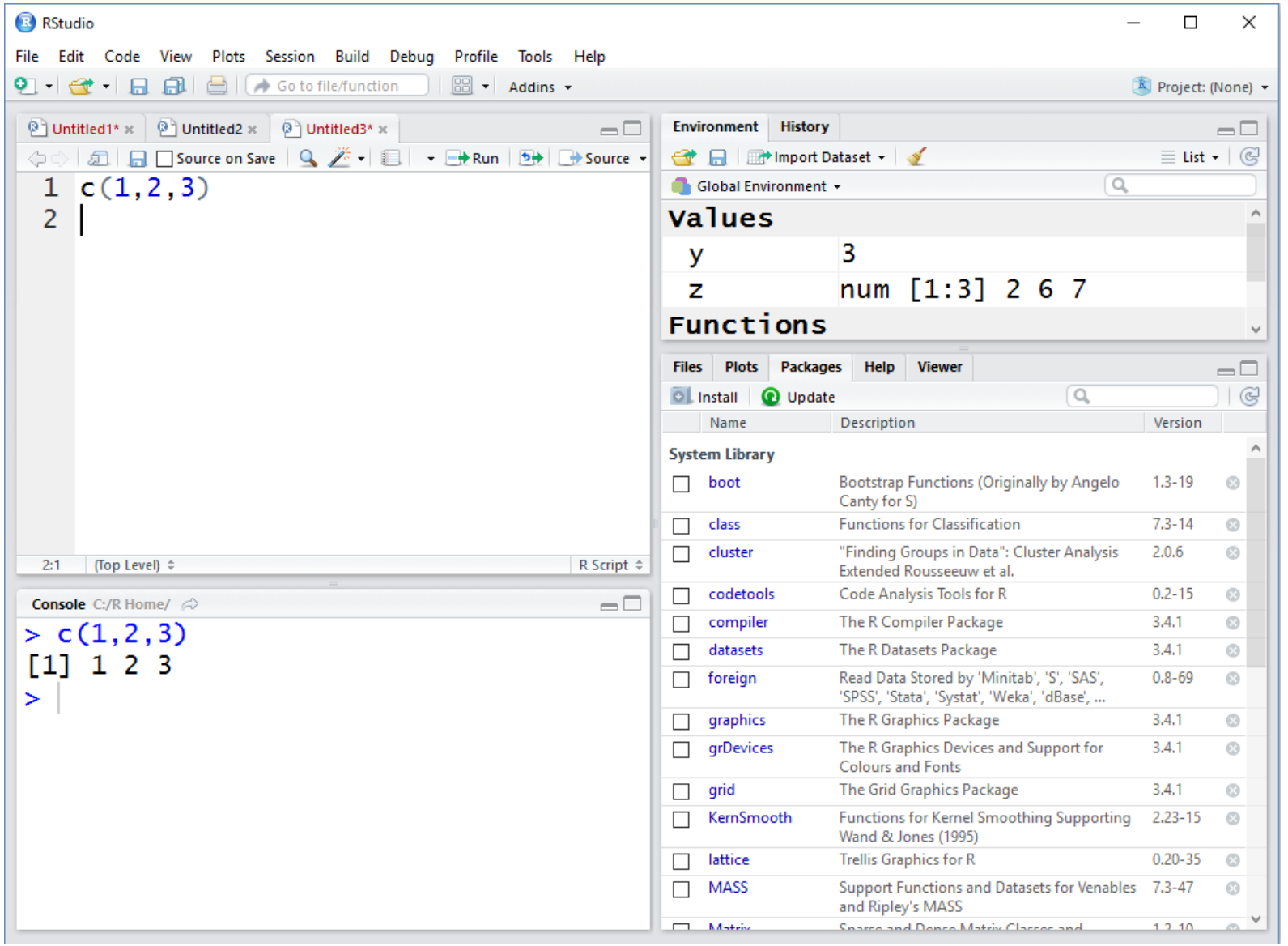
R Script Window



R Script Window (Cont.)



R Script Window (Cont.)



R Packages

- Are a collection of R functions, complied code and sample data.
- Are stored under a directory called *library* in the R environment.
- By default, R installs a set of packages during installation.
- More packages are added later, when they are needed for some specific purpose.
- When start the R console, only the default packages are available by default.
- Other packages which are already installed have to be loaded explicitly to be used by the R program that is going to use them.
- All the packages available in R language are listed at R Packages.

R Packages (Cont.)

- **Check Available R Packages**
  - Get library locations containing R packages

```
.libPaths()
```

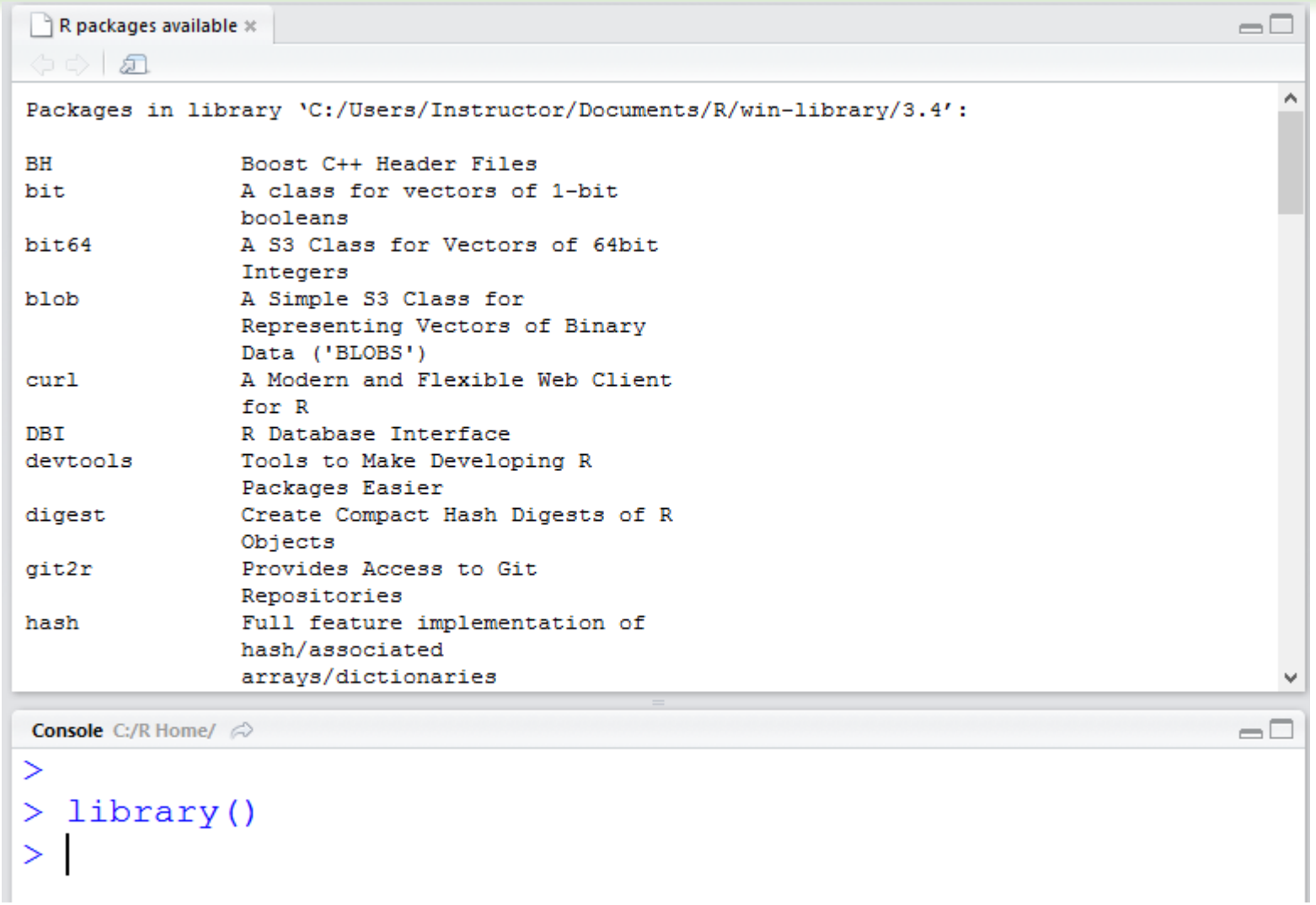
Console C:/R Home/

```
> .libPaths()
[1] "C:/Users/Instructor/Documents/R/win-library/3.4"
[2] "C:/Program Files/R/R-3.4.1/library"
>
```

R Packages (Cont.)

- Get the list of all the packages installed

```
library()
```



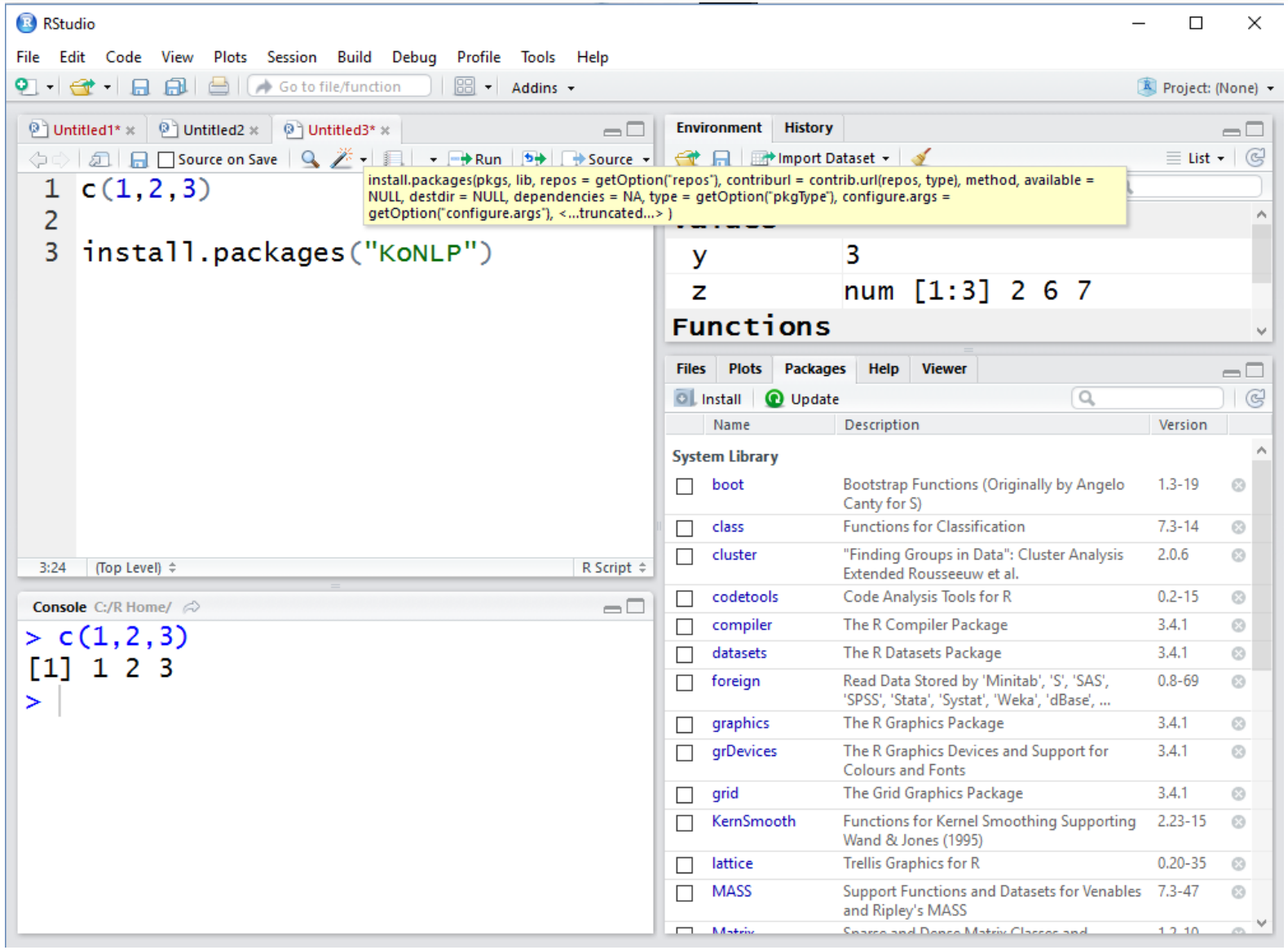
R Packages (Cont.)

- **Get all packages currently loaded in the R environment**

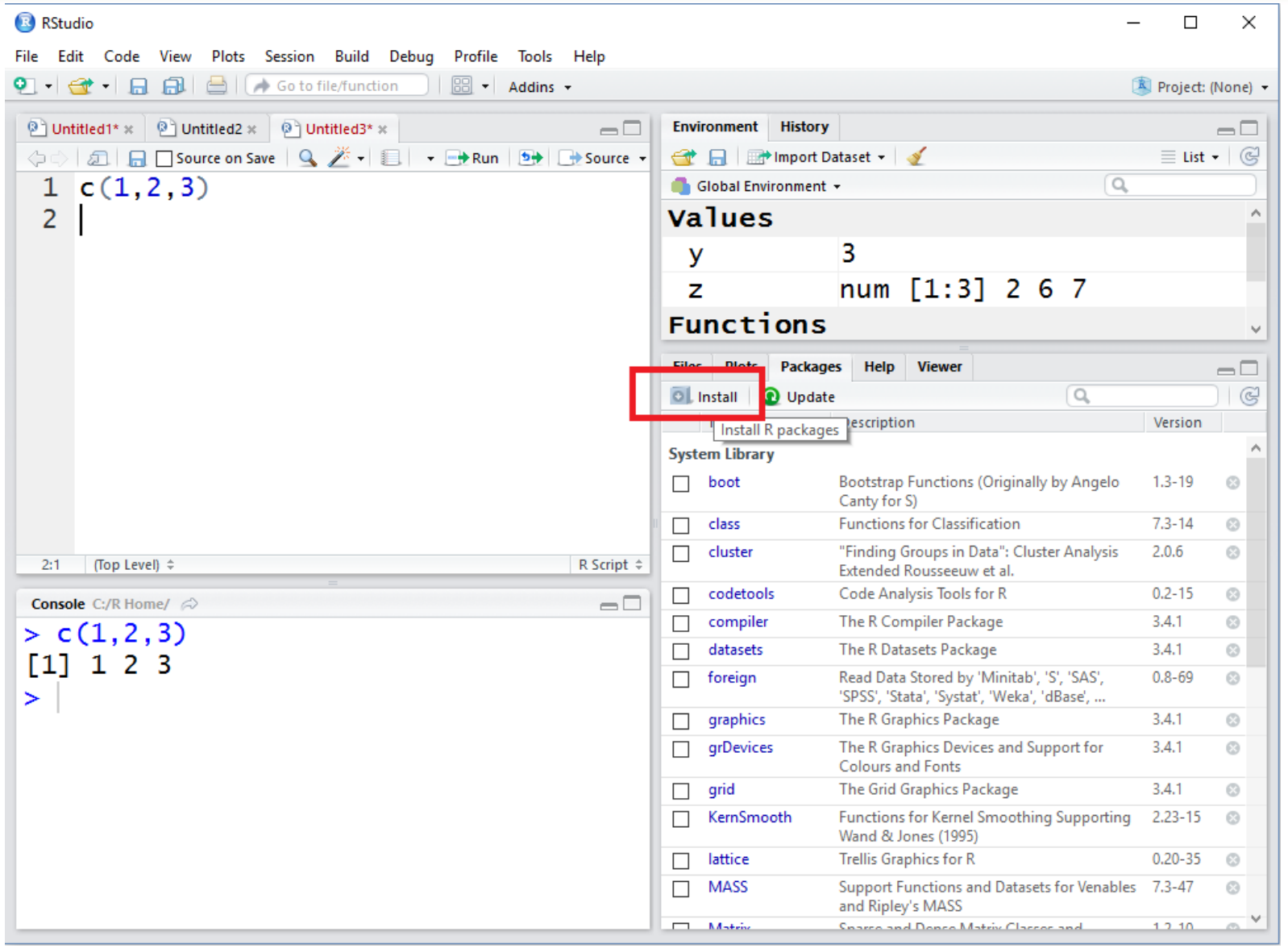
```
search()
```

```
Console C:/R Home/
>
> search()
 [1] ".GlobalEnv"        "tools:rstudio"
 [3] "package:stats"     "package:graphics"
 [5] "package:grDevices" "package:utils"
 [7] "package:datasets"  "package:methods"
 [9] "Autoloads"         "package:base"
>
```

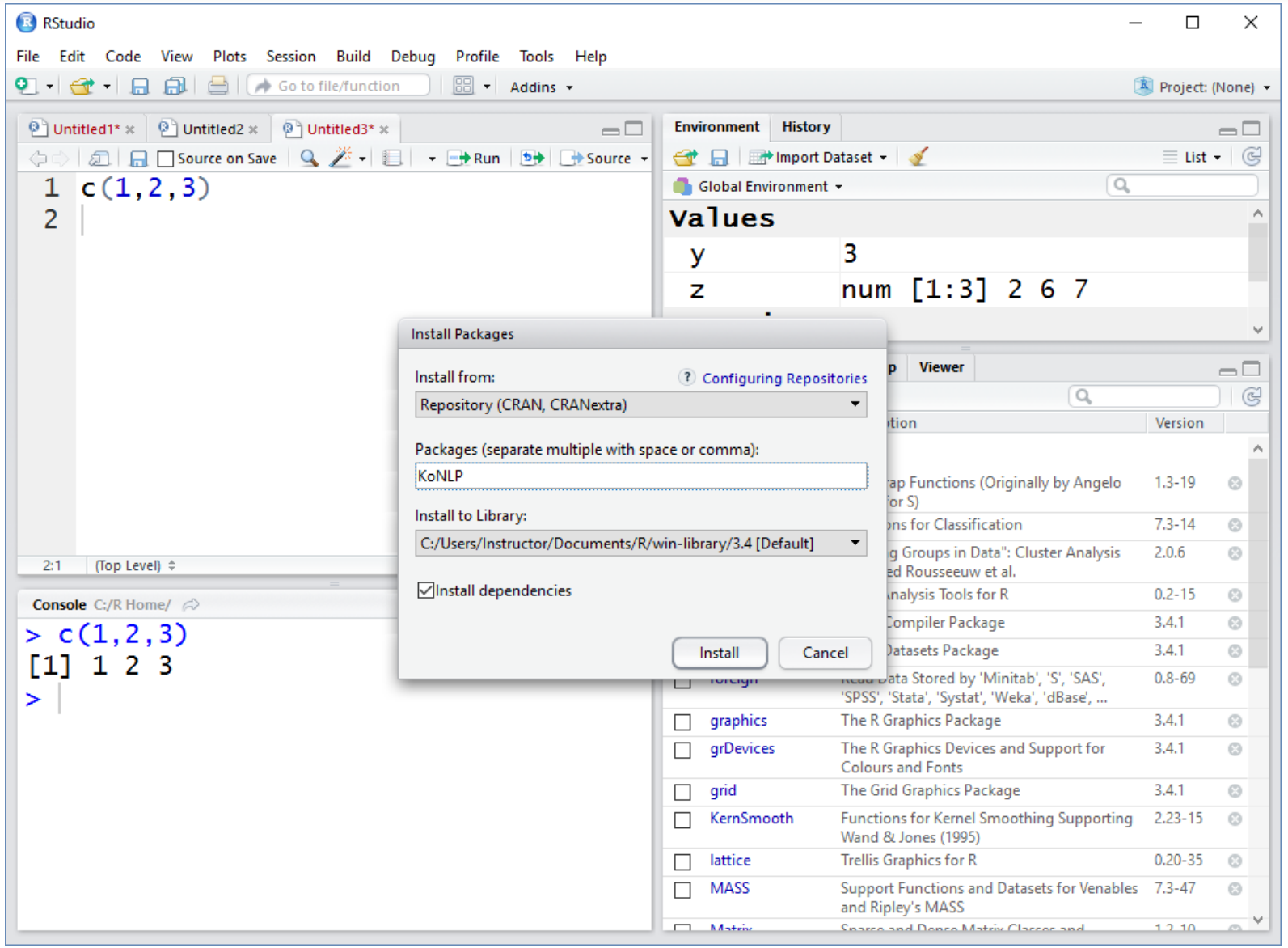
Package Installation & Usage



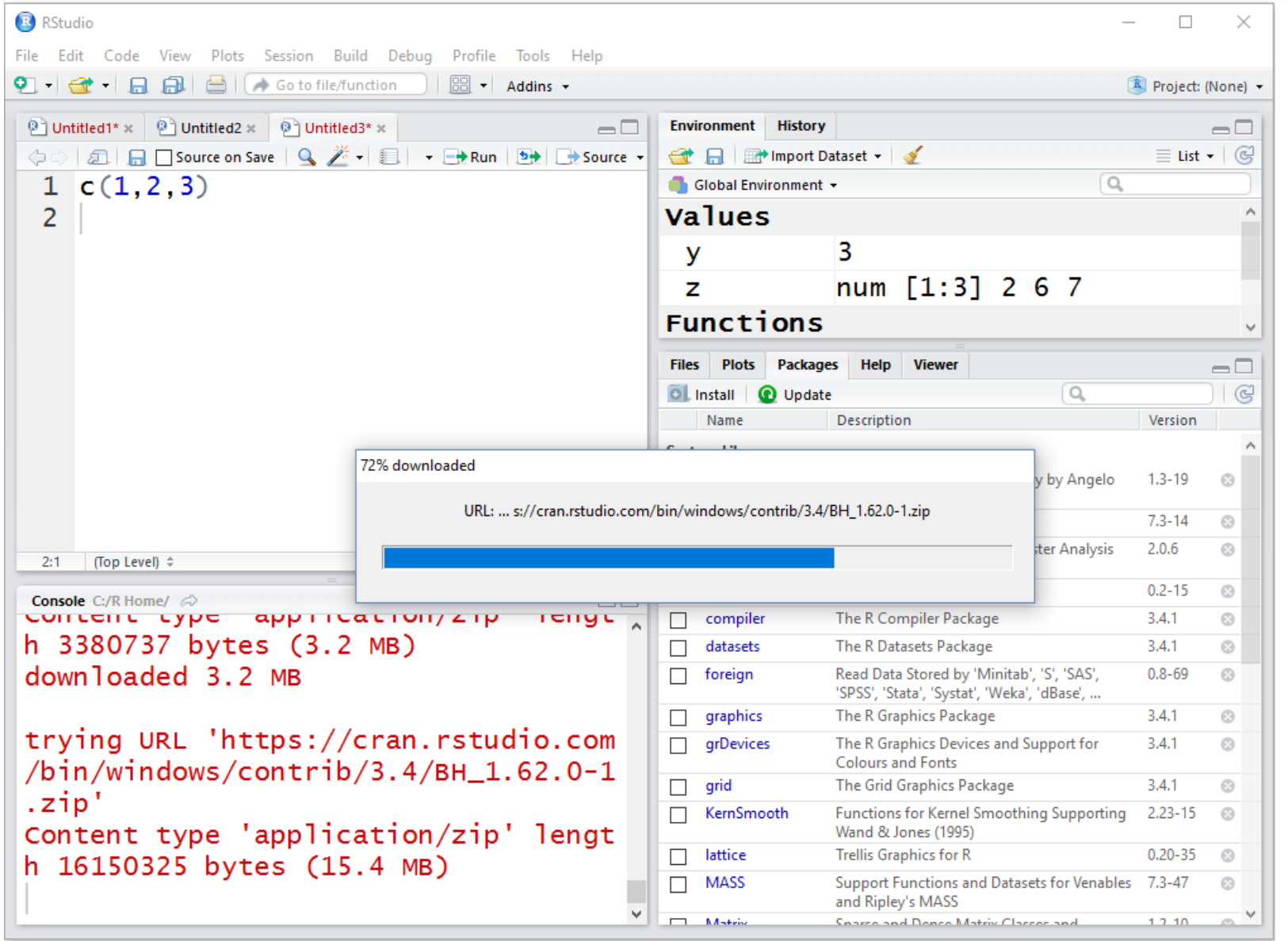
Package Installation & Usage (Cont.)



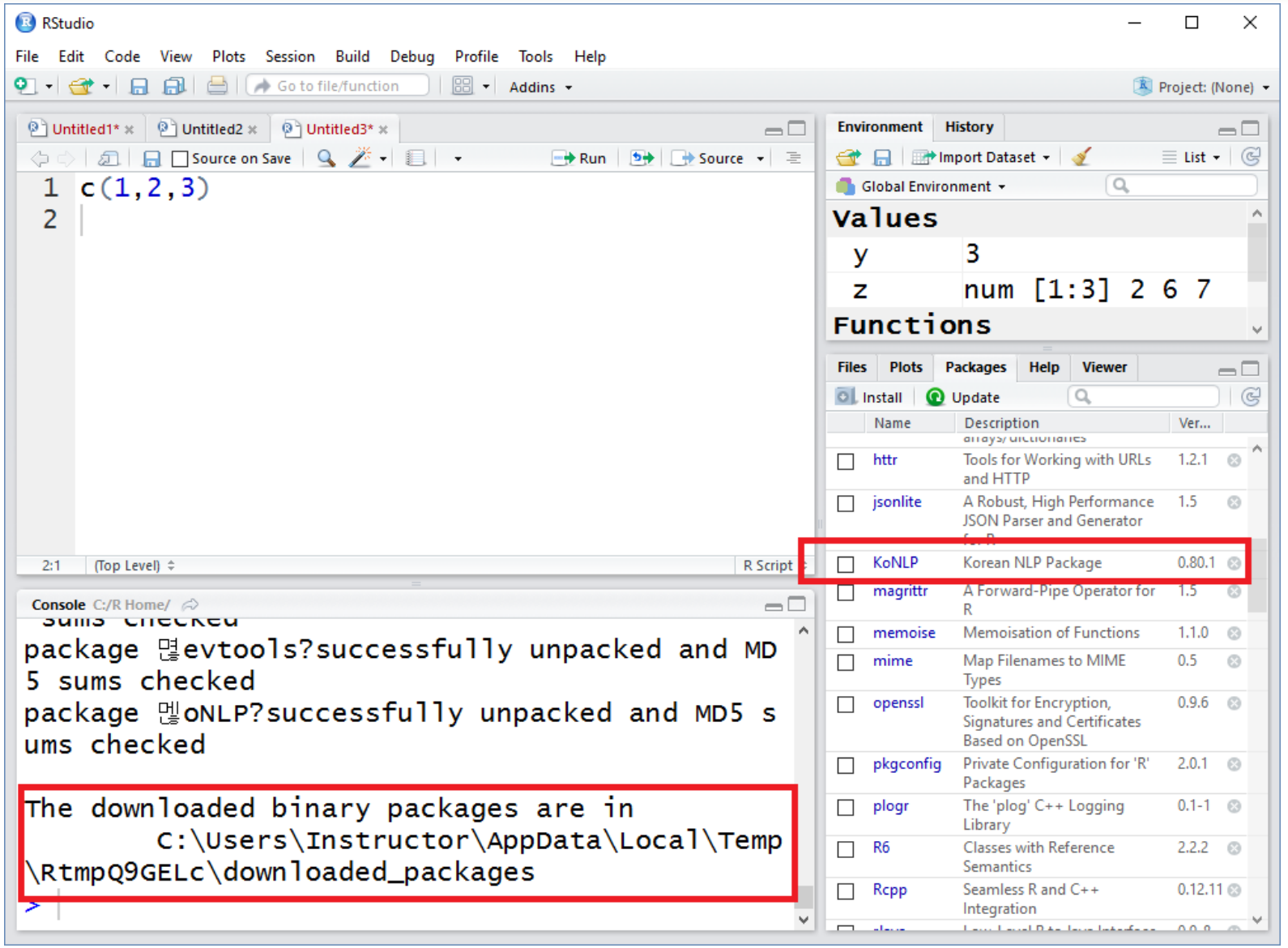
Package Installation & Usage (Cont.)



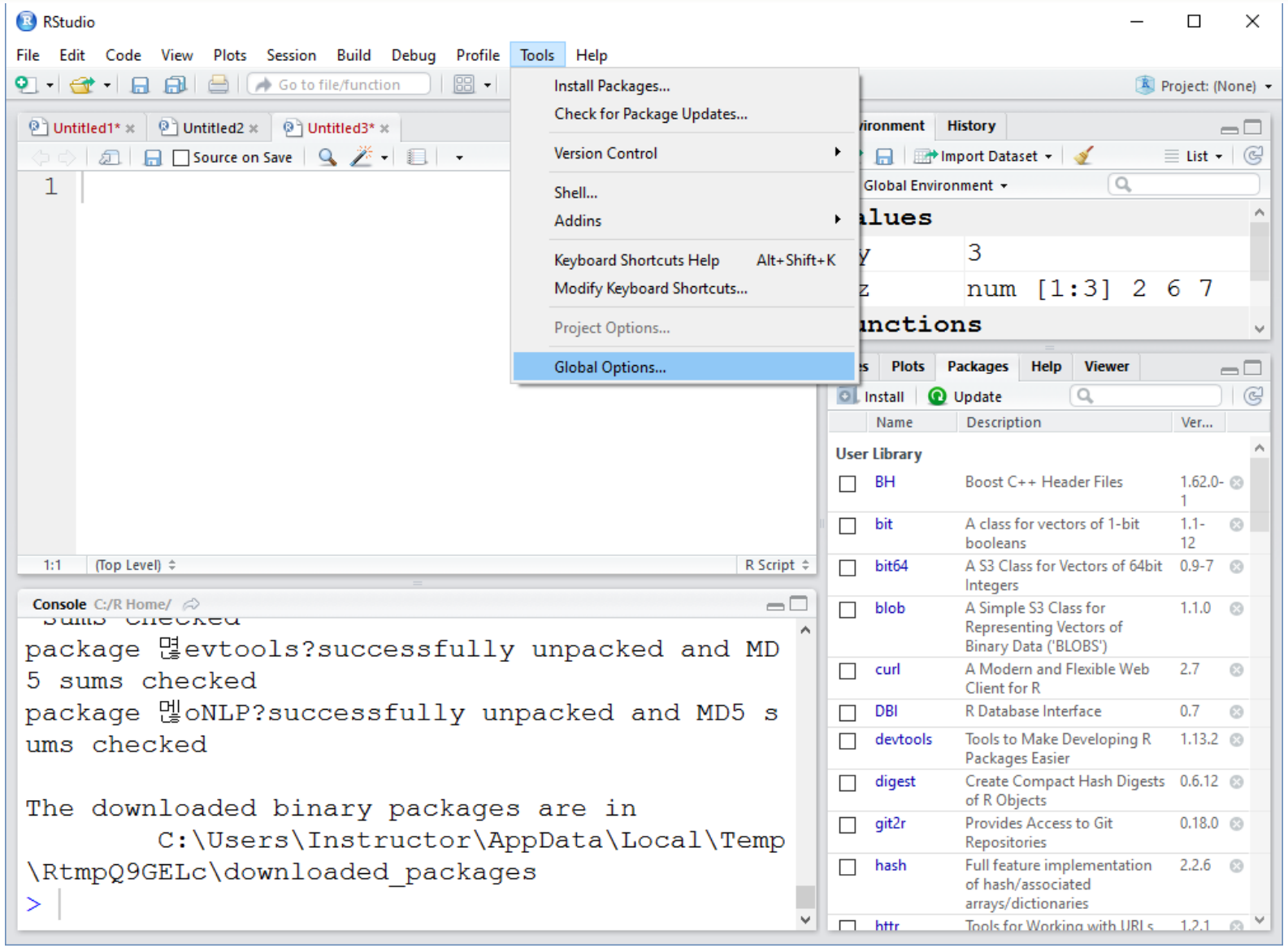
Package Installation & Usage (Cont.)



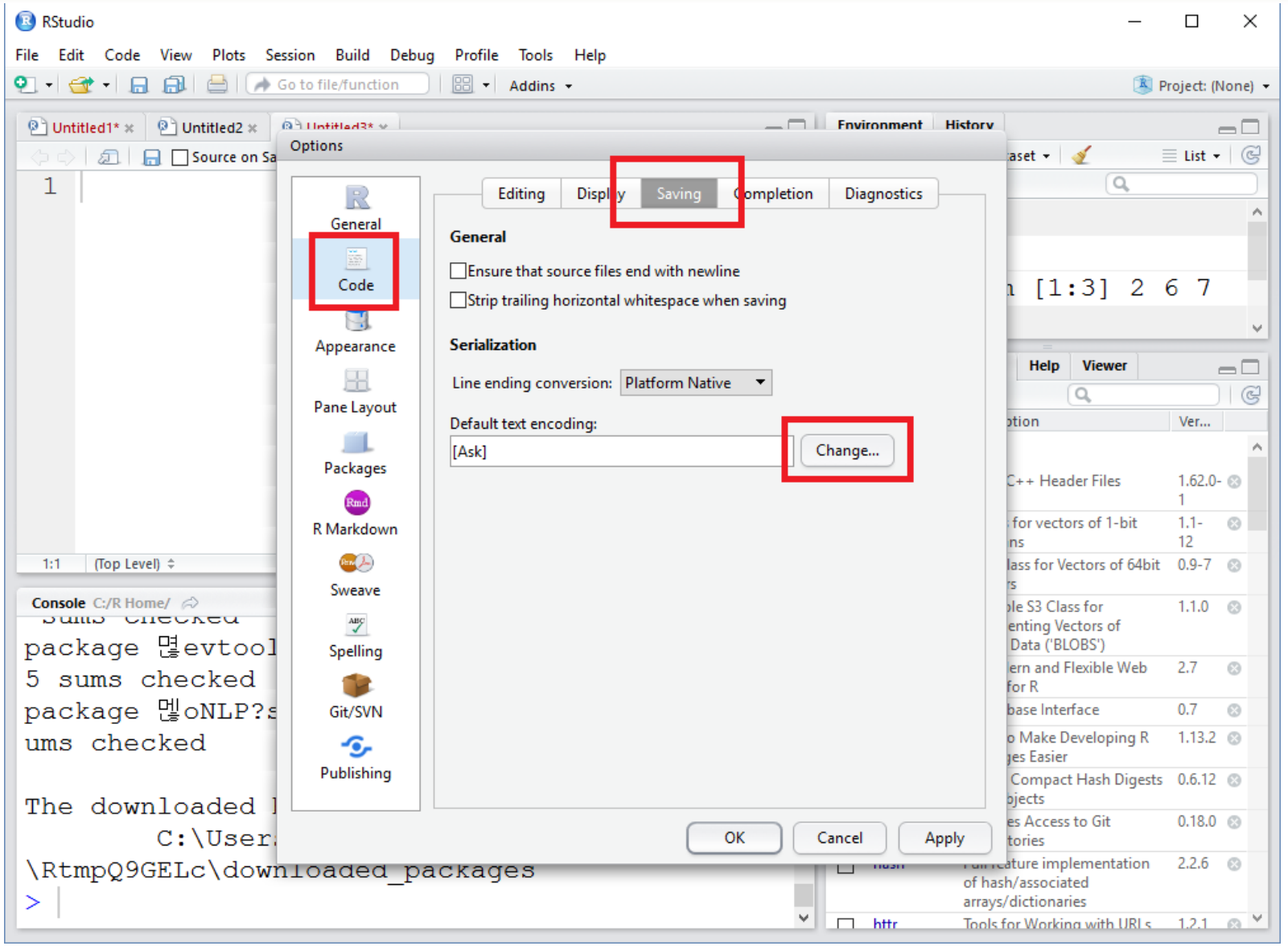
Package Installation & Usage (Cont.)



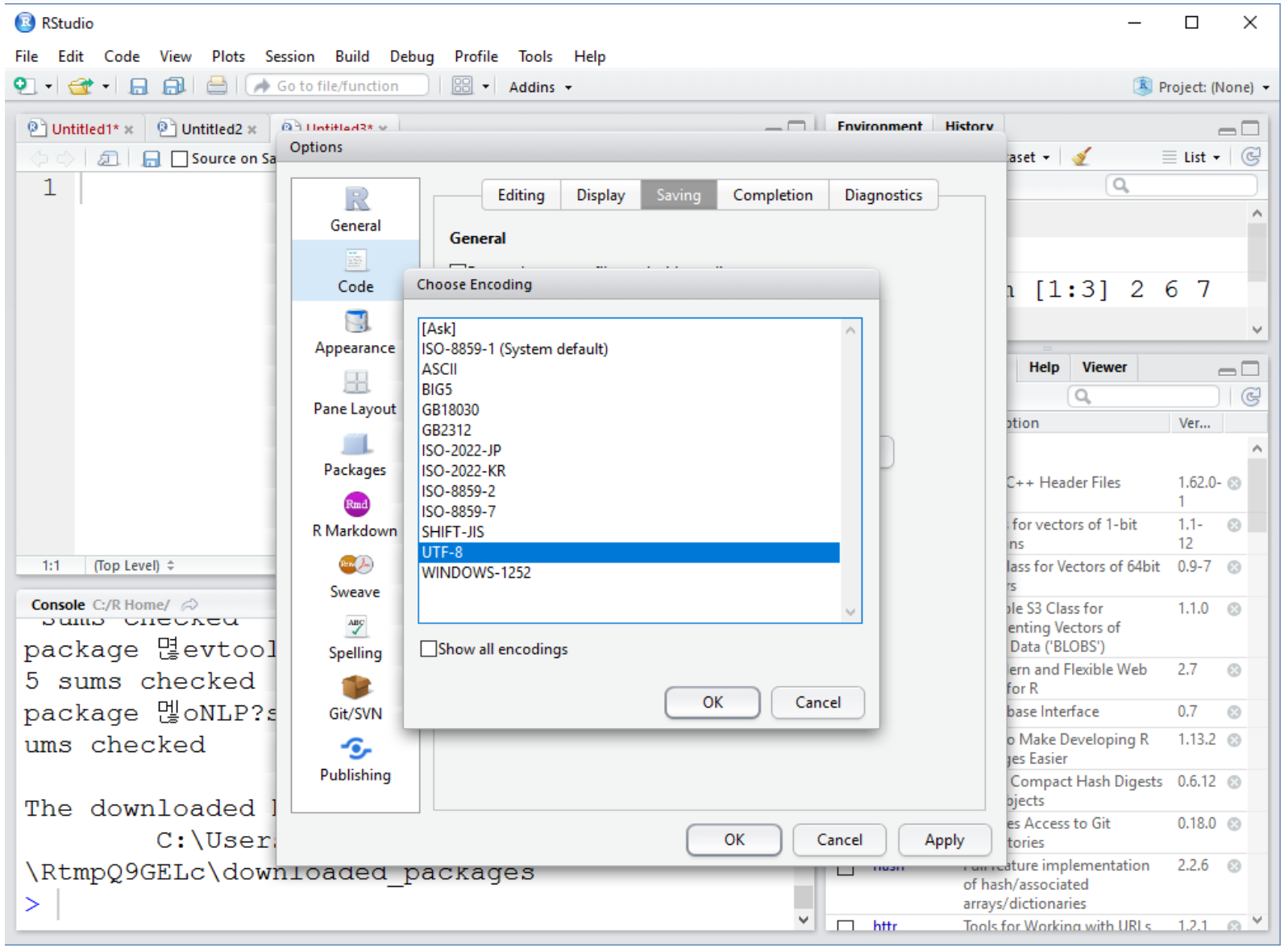
Package Installation & Usage (Cont.)



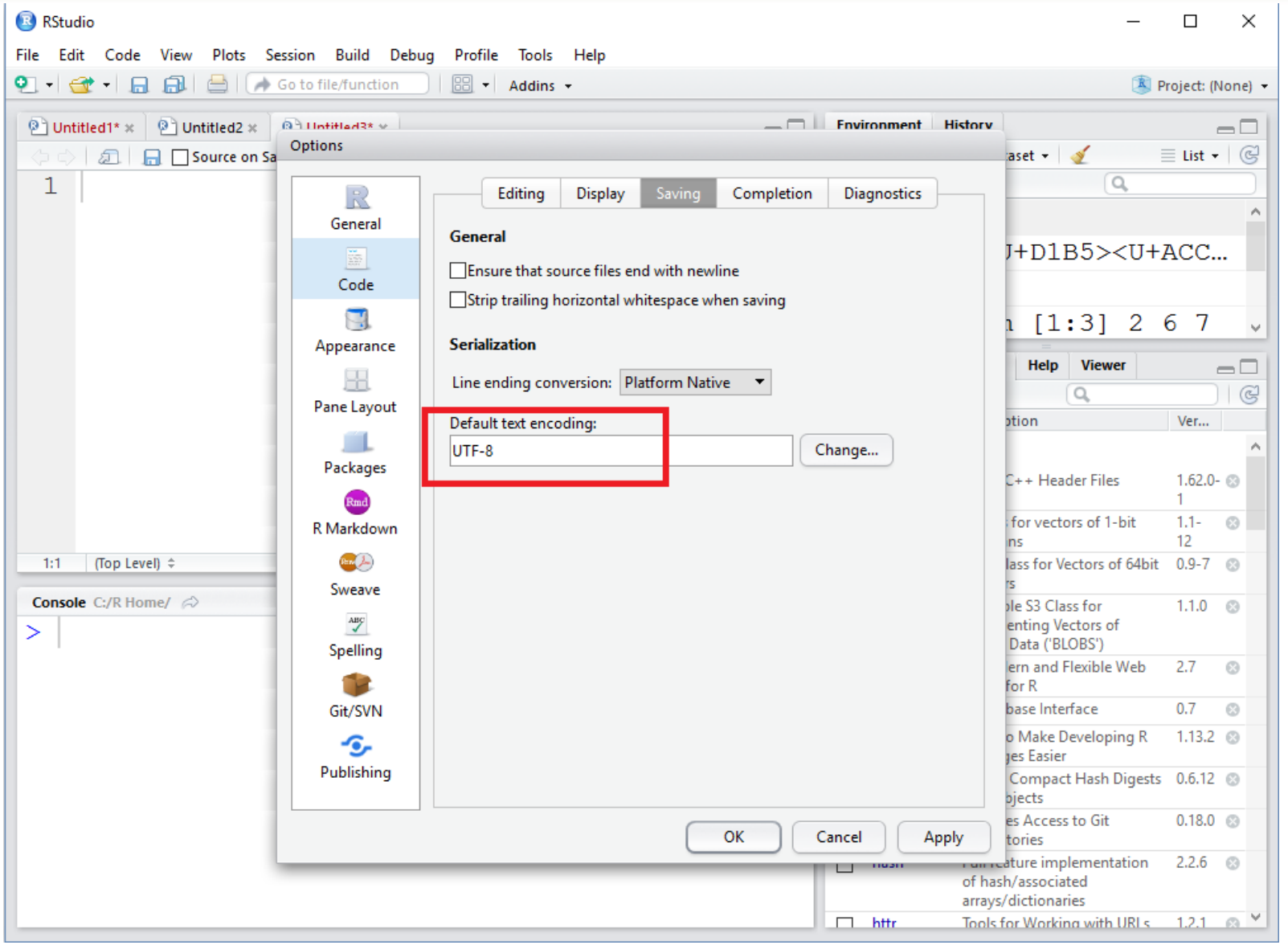
Package Installation & Usage (Cont.)



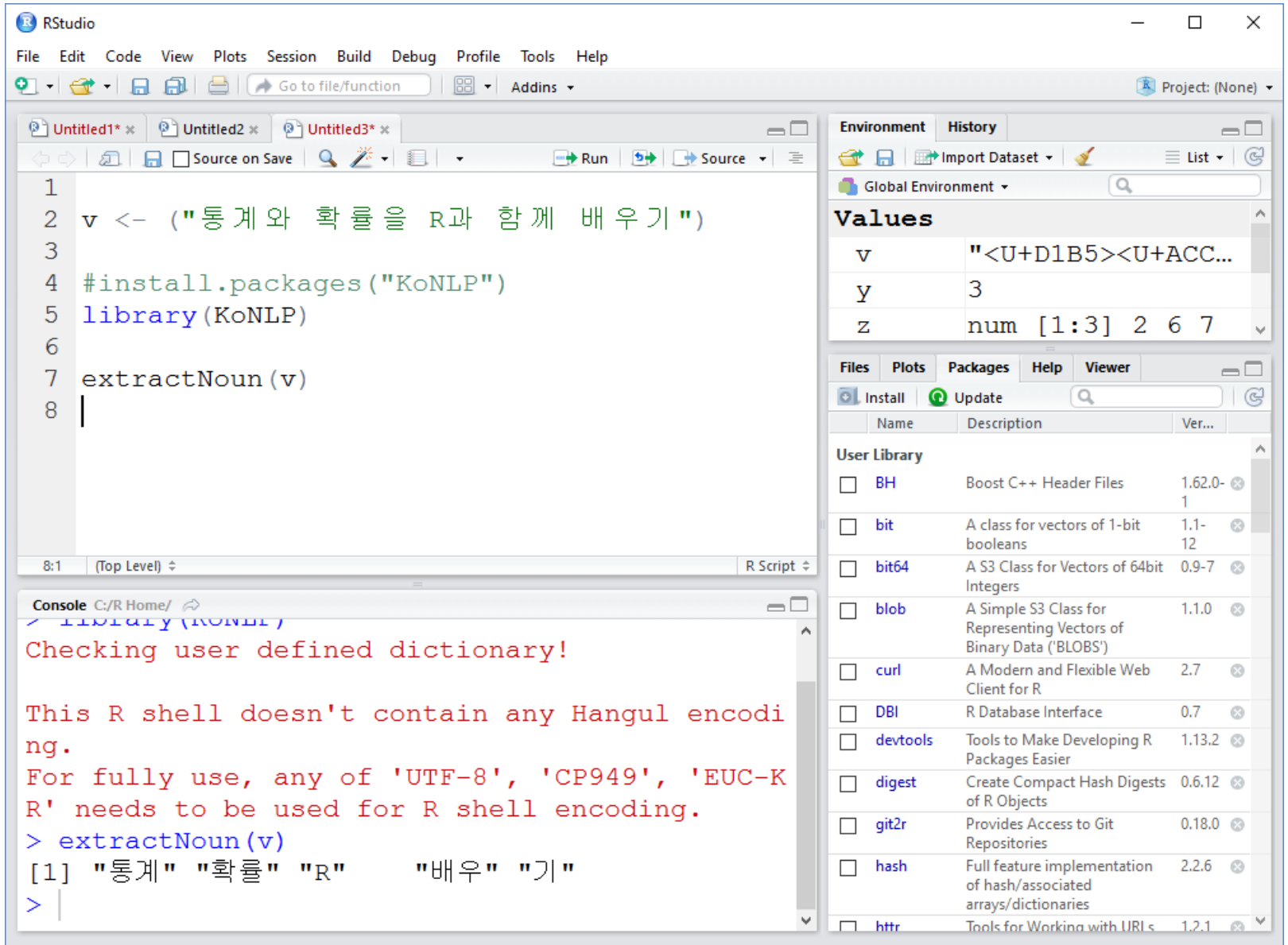
Package Installation & Usage (Cont.)



Package Installation & Usage (Cont.)



Package Installation & Usage (Cont.)



Package Remove

- To remove installed packages, do like below:

```
remove.packages("package_name")
```