

R

2019-09-06

Contents

1	Introduction	5
1.1	Goal	5
1.2	References	6
1.3	Evaluation	6
1.4	Schedule	6
1.5	References	7
2	R basics	9
2.1	What is R / Rstudio	9
2.2	R / Rstudio installation	9
2.3	Rstudio interface	16
2.4	Keyboard shortcuts	16
2.5	R programming basics and terminology	17
2.6	Set working directory	17
2.7	R coding practice	18
2.8	Variables and values	18
2.9	Variable type of (storage) mode	19
2.10	Variable - Vectors	19
2.11	Functions	20
2.12	Vectorized functions	20
2.13	Help	20
2.14	RStudio workspace	23
2.15	R packages	24
2.16	Data sets	27
2.17	Cheatsheet	29

Chapter 1

Introduction

- : 1213 (13:00~16:00)
- :
- : 042-860-4372, haseong@kribb.re.kr (1143)
- site: <https://greendaygh.github.io/Rstat2019/>

1.1 Goal

- . R

1.2 References



- Using R for Introductory Statistics by John Verzani
 - Free version of 1st Edition
 - * <https://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf>
 - * <http://cbb.sjtu.edu.cn/~mywu/bi217/usingR.pdf>
 - Second edition
 - * <https://www.crcpress.com/Using-R-for-Introductory-Statistics-Second-Edition/Verzani/p/book/9781466590731>
- R for Data Science (<https://r4ds.had.co.nz>, <https://github.com/hadley>)
- <https://resources.rstudio.com/>
- (,)

1.3 Evaluation

- 50% / 50% / 80 S, 80 U

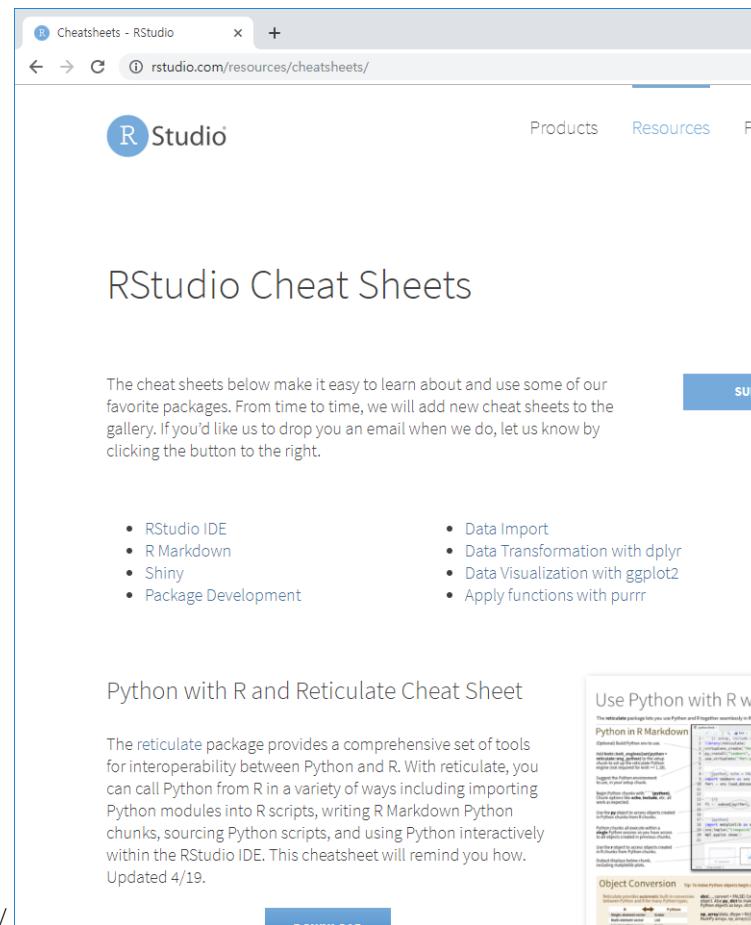
1.4 Schedule

- 1 - R basics / introduction of data

- 2 - Univariate data – Summary statistics (, ,)
- 3 - Bivariate data – Correlation / Independence (, , ,)
- 4 - Multivariate data – R data structure (, R , R)
- 5 - Populations – Families of distributions
- 6 - Sampling – Distribution and CLT ,
- 7 - Statistical inference
- 8 - Confidence intervals
- 9 - Significance test - parameteric ()
- 10 - Significance test – non parametric ()
- 11 - Goodness of fit - parametric ()
- 12 - Goodness of fit – non parametirc ()
- 13 - Linear regression – basics & simple LR
- 14 - Multiple linear regression
- 15 - Analysis of variance
- 16 - Logistic / Non-linear regression /
- 9/25 ()

1.5 References

- R <https://www.r-project.org/>
- Rstudio <https://www.rstudio.com/>
- Packages for biologists <https://www.bioconductor.org/>
- R (, , ,)
 - <https://cran.r-project.org/doc/manuals/r-release/R-intro.html>
 - <https://cran.r-project.org/doc/manuals/r-release/R-data.html>
 - <https://cran.r-project.org/doc/manuals/r-release/R-admin.html>
- R ebooks
 - <https://bookdown.org/>
- Cheat Sheets



The screenshot shows the RStudio Cheat Sheets page at rstudio.com/resources/cheatsheets/. The page features the R Studio logo and navigation links for Products and Resources. A section titled "RStudio Cheat Sheets" is displayed, with a note about the availability of new cheat sheets and a sign-up button. Below this, two columns of bullet points list various R-related topics: RStudio IDE, R Markdown, Shiny, Package Development on the left; and Data Import, Data Transformation with dplyr, Data Visualization with ggplot2, and Apply functions with purrr on the right. A large preview image of the "Use Python with R" cheat sheet is shown on the right side of the page.

RStudio Cheat Sheets

The cheat sheets below make it easy to learn about and use some of our favorite packages. From time to time, we will add new cheat sheets to the gallery. If you'd like us to drop you an email when we do, let us know by clicking the button to the right.

- RStudio IDE
- R Markdown
- Shiny
- Package Development

- Data Import
- Data Transformation with dplyr
- Data Visualization with ggplot2
- Apply functions with purrr

Python with R and Reticulate Cheat Sheet

The reticulate package provides a comprehensive set of tools for interoperability between Python and R. With reticulate, you can call Python from R in a variety of ways including importing Python modules into R scripts, writing R Markdown Python chunks, sourcing Python scripts, and using Python interactively within the RStudio IDE. This cheatsheet will remind you how. Updated 4/19.

Use Python with R

Download

– <https://www.rstudio.com/resources/cheatsheets/>

Chapter 2

R basics

2.1 What is R / Rstudio



- R is a programming language that runs computations (<https://www.r-project.org/>)
- RStudio is an integrated development environment (IDE) that provides an interface for the programming (<https://www.rstudio.com/>)

2.2 R / Rstudio installation

- Install R first and then install RStudio second

The screenshot shows two browser windows side-by-side. The top window displays the main homepage of the R Project for Statistical Computing ([r-project.org](https://www.r-project.org)). The bottom window displays the 'CRAN - Mirrors' page (cran.r-project.org/mirrors.html).

Main R Project Page:

- Logo:** The R logo.
- Navigation:** [Home], Download, CRAN, R Project, About R, Logo, Contributors, What's New?, Reporting Bugs, Conferences, Search, Get Involved: Mailing Lists, Developer Pages, R Blog.
- R Foundation:** Foundation, Board, Members, Donors, Donate.
- Help With R:** Getting Help.
- News:** News via Twitter.
- Getting Started:** A paragraph explaining R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. A link to download R is provided.
- FAQs:** If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

CRAN Mirrors Page:

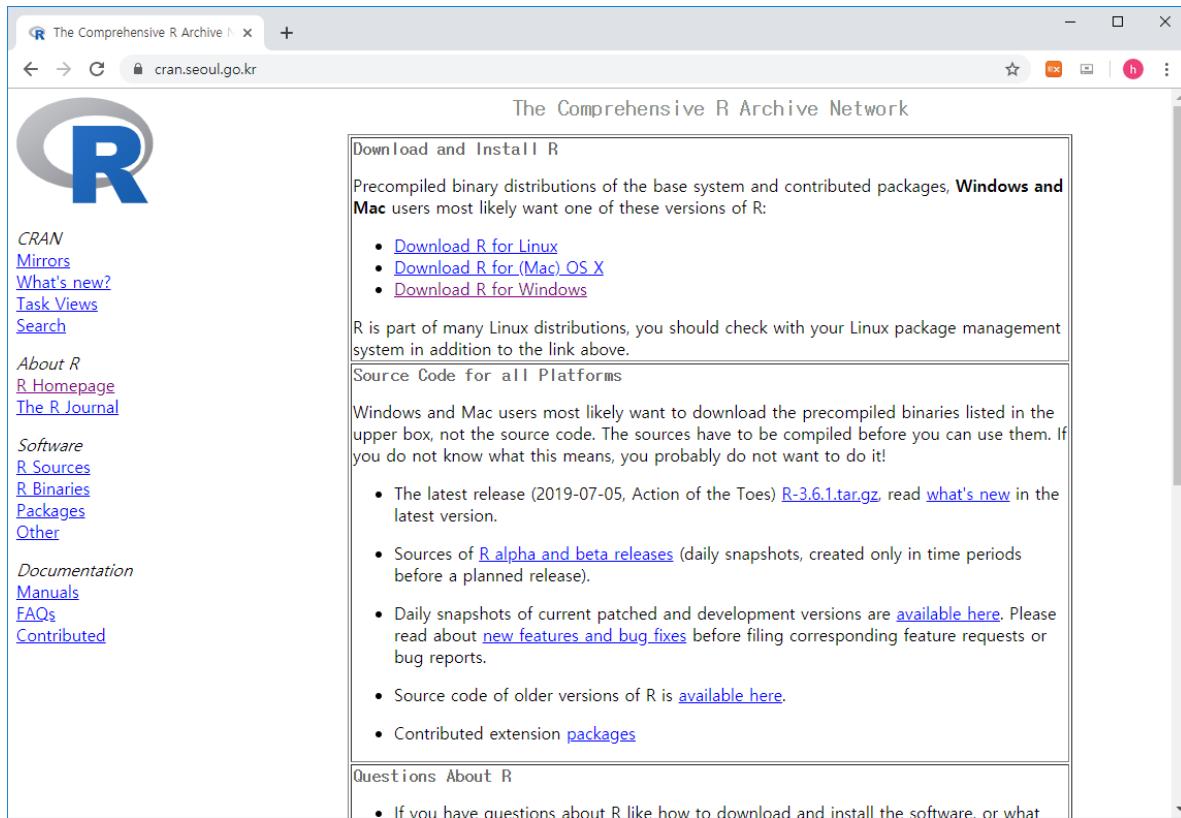
CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: [main_page](#), [windows_release](#), [windows_old_release](#).

If you want to host a new mirror at your institution, please have a look at the [CRAN Mirror HOWTO](#).

0-Cloud	https://cloud.r-project.org/	Automatic redirection to servers worldwide, currently sponsored by Rstudio
Algeria	https://cran.usthb.dz/	University of Science and Technology Houari Boumediene
Argentina	http://mirror.fcaglp.unlp.edu.ar/CRAN/	Universidad Nacional de La Plata
Australia	https://cran.csiro.au/ https://mirror.aarnet.edu.au/pub/CRAN/ https://cran.ms.unimelb.edu.au/ https://cran.curtin.edu.au/	CSIRO AARNET School of Mathematics and Statistics, University of Melbourne Curtin University of Technology
Austria	https://cran.wu.ac.at/	Wirtschaftsuniversität Wien
Belgium	https://www.freestatistics.org/cran/ https://lib.ugent.be/CRAN/	Patrick Wessa Ghent University Library
Brazil	https://cran-rc3sl.ufpr.br/ https://cran.fiocruz.br/ https://vps.fmvz.usp.br/CRAN/ https://brieger.esalq.usp.br/CRAN/	Universidade Federal do Paraná Oswaldo Cruz Foundation, Rio de Janeiro University of São Paulo, São Paulo University of São Paulo, Piracicaba
Bulgaria	https://cran.math.bas.bg/	

The Comprehensive R Archive Network



Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

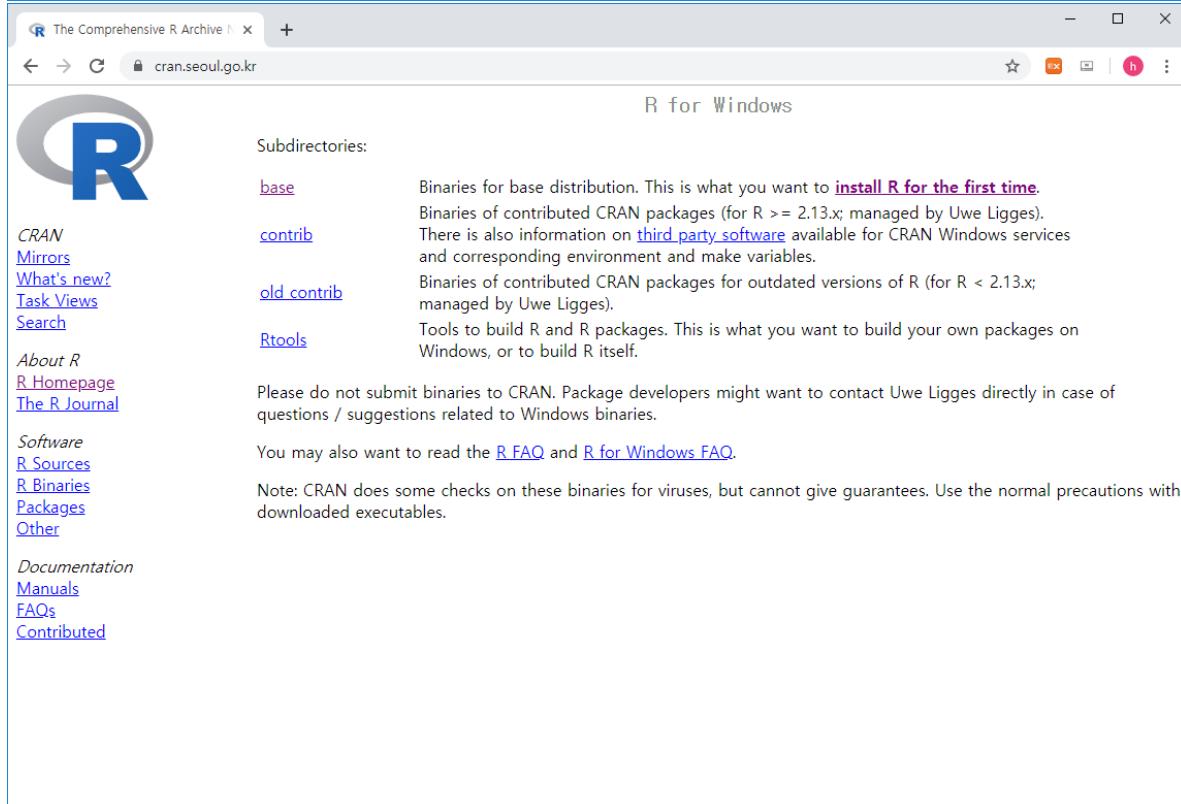
Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2019-07-05, Action of the Toes) [R-3.6.1.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- [If you have questions about R like how to download and install the software, or what](#)

R for Windows



Subdirectories:

base	Binaries for base distribution. This is what you want to install R for the first time .
contrib	Binaries of contributed CRAN packages (for R >= 2.13.x; managed by Uwe Ligges). There is also information on third party software available for CRAN Windows services and corresponding environment and make variables.
old_contrib	Binaries of contributed CRAN packages for outdated versions of R (for R < 2.13.x; managed by Uwe Ligges).
Rtools	Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

The screenshot shows the CRAN (Comprehensive R Archive Network) website for Windows. The main content is titled "R-3.6.1 for Windows (32/64 bit)". It features the R logo and navigation links for CRAN, Mirrors, What's new?, Task Views, and Search. Under "About R", links include R Homepage, The R Journal, Software, R Sources, R Binaries, Packages, and Other. Documentation links include Manuals, FAQs, and Contributed. A prominent section is "Download R 3.6.1 for Windows" (81 megabytes, 32/64 bit), with links for Installation and other instructions and New features in this version. Below this, a note explains how to verify package integrity using md5sum and provides links for graphical and command line versions. A "Frequently asked questions" section lists links for Does R run under my version of Windows?, How do I update packages in my previous version of R?, and Should I run 32-bit or 64-bit R?. A note for webmasters mentions a stable link to the current Windows binary release. The page footer indicates the last change was on 2019-07-05.

The screenshot shows the RStudio website. The header includes the R logo and navigation links for Products, Resources, Pricing, About Us, and Blogs. The main content highlights "New RStudio Server Pro Standard and Enterprise" and "Launch remote interactive sessions and jobs on your compute cluster of choice". A "Learn More" button is present. To the right, there is a large image of a monitor displaying RStudio's interface with multiple windows open. Below this, the section "Introducing RStudio Team" is shown, with a note that RStudio's recommended professional data science solutions for every team. RStudio Team includes: R-3.6.1-win.exe. The bottom right corner has a "전체 보기" (View All) button.

- Rstudio

The screenshot shows a web browser displaying the RStudio website at rstudio.com/products/rstudio/. The page header includes the RStudio logo and navigation links for Products, Resources, Pricing, About Us, Blogs, and a search icon.

RStudio

Take control of your R code

RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. Click here to see more RStudio features.

RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian/Ubuntu, RedHat/CentOS, and SUSE Linux).

Desktop

Run RStudio on your desktop

[RStudio Desktop >](#)

Server

Centralize access and computation

[RStudio Server >](#)

Open Source Edition

Commercial License

Overview

- Access RStudio locally
- Syntax highlighting, code completion, and smart indentation
- Execute R code directly from the source editor
- Quickly jump to function definitions
- Easily manage multiple working directories using projects
- Integrated R help and documentation
- Interactive debugger to diagnose and fix errors quickly
- Extensive package development tools

All of the features of open source; plus:

- A commercial license for organizations not able to use AGPL software
- Access to priority support

Support

Community forums only

- Priority Email Support
- 8 hour response during business hours (ET)

License

AGPL v3

[RStudio License Agreement](#)

Pricing

Free

\$995/year

[DOWNLOAD RSTUDIO DESKTOP](#)

[BUY NOW](#)

<https://www.rstudio.com/pricing/>

The screenshot shows the RStudio download page at rstudio.com/products/rstudio/download/. The page features the RStudio logo and navigation links for Products, Resources, Pricing, About Us, and Blogs. A search bar is also present. The main heading is "Choose Your Version of RStudio". Below it, a paragraph describes RStudio as a set of integrated tools for R, mentioning its console, editor, and various tools for plotting and managing workspaces. It links to "Learn More about RStudio features". To the right, there's a section for "RStudio Team" with a graphic of overlapping colored lines and text about RStudio's new solution for professional data science teams. Below this, four product options are listed:

RStudio Desktop	RStudio Desktop	RStudio Server	RStudio Server Pro
Open Source License	Commercial License	Open Source License	Commercial License
FREE	\$995 per year	FREE	\$4,975 per year (5 Named Users)
DOWNLOAD	BUY	DOWNLOAD	BUY

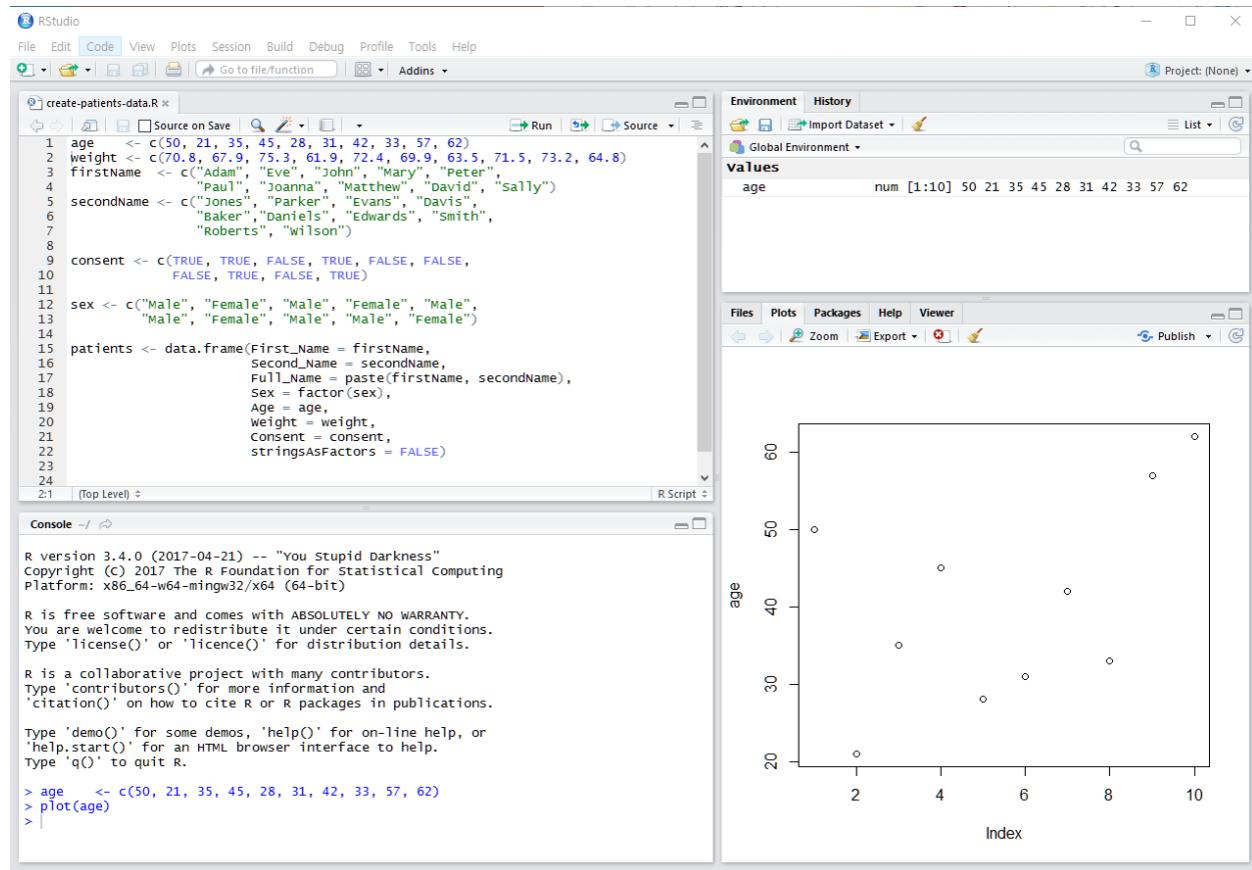
At the bottom left, there's a link to "R-3.6.1-win.exe" with a "취소됨" status. On the far right, there are "전체 보기" and "X" buttons.

The screenshot shows a web browser window displaying the RStudio download page at rstudio.com/products/rstudio/download/#download. The page header includes the RStudio logo and navigation links for Products, Resources, Pricing, About Us, Blogs, and a search icon.

The main content area displays the "RStudio Desktop 1.2.1335 — Release Notes". It contains several paragraphs of text providing system requirements and instructions for Linux users regarding code-signing keys. Below this, a section titled "Installers for Supported Platforms" lists various installer files with their sizes, dates, and MD5 checksums. At the bottom, there is a section for "Zip/Tarballs" with a single entry for "R-3.6.1-win.exe".

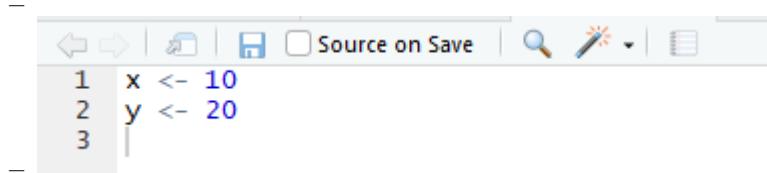
Installers	Size	Date	MD5
RStudio 1.2.1335 - Windows 7+ (64-bit)	126.9 MB	2019-04-08	d0e2470f1f8ef4cd35a669aa323a2136
RStudio 1.2.1335 - macOS 10.12+ (64-bit)	121.1 MB	2019-04-08	6c570b0e2144583f7c48c284ce299eef
RStudio 1.2.1335 - Ubuntu 14/Debian 8 (64-bit)	92.2 MB	2019-04-08	c1b07d0511469abfe582919b183eee83
RStudio 1.2.1335 - Ubuntu 16 (64-bit)	99.3 MB	2019-04-08	c142d69c210257fb10d18c045fff13c7
RStudio 1.2.1335 - Ubuntu 18/Debian 10 (64-bit)	100.4 MB	2019-04-08	71a8d1990c0d97939804b46cfb0aea75
RStudio 1.2.1335 - Fedora 19/RedHat 7 (64-bit)	114.1 MB	2019-04-08	296b6ef88969a91297fab6545f256a7a
RStudio 1.2.1335 - Debian 9 (64-bit)	100.6 MB	2019-04-08	1e32d4d6f6e216f086a81ca82ef65a91
RStudio 1.2.1335 - OpenSUSE 15 (64-bit)	101.6 MB	2019-04-08	2795a63c7efdb8e2aa2dae86ba09a81e5
RStudio 1.2.1335 - SLES/OpenSUSE 12 (64-bit)	94.4 MB	2019-04-08	c65424b06ef6737279d982db9eefcae1

2.3 Rstudio interface



2.4 Keyboard shortcuts

- - <https://support.rstudio.com/hc/en-us/articles/200711853-Keyboard-Shortcuts>
 - Tools → Keyboard shortcut Quick Reference (Alt + Shift + K)
- (Ctrl+1) (Ctrl+2)
- (Ctrl+Enter)
- (Ctrl + Shift + C)
 - Starting with a hashmark ('#'), everything to the end of the line is a comment
-



- Ctrl + enter
- Ctrl + 2
- x x+y
- Ctrl + 1
- Ctrl + Shift + C

```
# x <- 10
# y <- 20
```

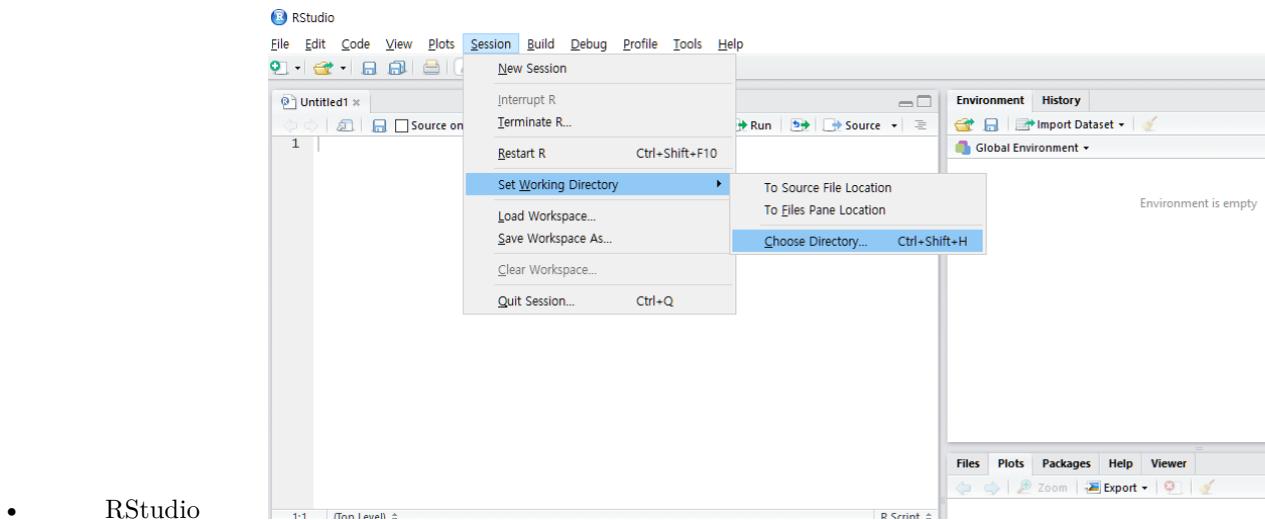
2.5 R programming basics and terminology

- Console:
- Code: R /
- Objects (, variable): ()
- Data types: Integers, doubles/numerics, logicals, and characters.
- Object (Variable) types:
 - Vectors: combine function c() EX: c(6, 11, 13, 31, 90, 92)
 - Factors:
 - Data frames: 2D matrix
- Conditionals (,):
 - if: ==, & (AND), | (OR) Ex: (2 + 1 == 3) & (2 + 1 == 4)
 - for, while:
- Functions (, commands): , - (arguments) - (output)

2.6 Set working directory

-
- c: rstat01

```
getwd()
dir()
setwd("C:\\rstat01")
getwd()
dir()
```



- RStudio

2.7 R coding practice

-

```
2 + 2
((2 - 1)^2 + (1 - 3)^2 )^(1/2)
2 + 2; 2 - 2
```

- :

2.8 Variables and values

- R is a programming language
- Assignment operator (<- OR =)
 - Valid object name <- value
 - : Alt + - (the minus sign)
- Built-in variables

```
x <- 2
y <- x^2 - 2*x + 1
y
x <- "two"
some_data <- 9.8
pi
```

-

- Characters (letters), numbers, “_”, “.”
- A and a are different symbols
- Names are effectively unlimited in length

```
i_use_snake_case <- 1
otherPeopleUseCamelCase <- 2
some.people.use.periods <- 3
And_aFew.People_RENOUNCEconvention <- 4
```

- (Tab completion) in RStudio

2.9 Variable type of (storage) mode

Type	Explanation
285	
34.67	Numeric (Integer)
4.23E-4	
TRUE, T	
FALSE, F	Logical
'B'	
"Hello" or 'Hello'	Character
NULL	NULL

2.10 Variable - Vectors

- Combine function `c()`: Concatenating elements end to end

```
x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
y <- c("X1", "Y2", "X3", "Y4")
```

- : Subsets of the elements of a vector

```
x[1]
x[1:3]
x[c(1,2,4)]
y[3]
```

2.11 Functions

- Function define

```
my_sine <- function(x){
  y <- sin(x)
  return(y)
}
```

- Usage

```
my_sine(pi)
```

- Terminology
 - function name: `my_sine`
 - parameter: `x`
 - argument: `pi`
 - return value: `y`
- Built-in functions
 - Arguments separated by commas
 - Tab completion

```
x <- pi
sin(x)
sqrt(x)
log(x)
log(x, 10)
x <- c(10, 20, 30)
x + x
mean(x)
sum(x)/length(x)
```

2.12 Vectorized functions

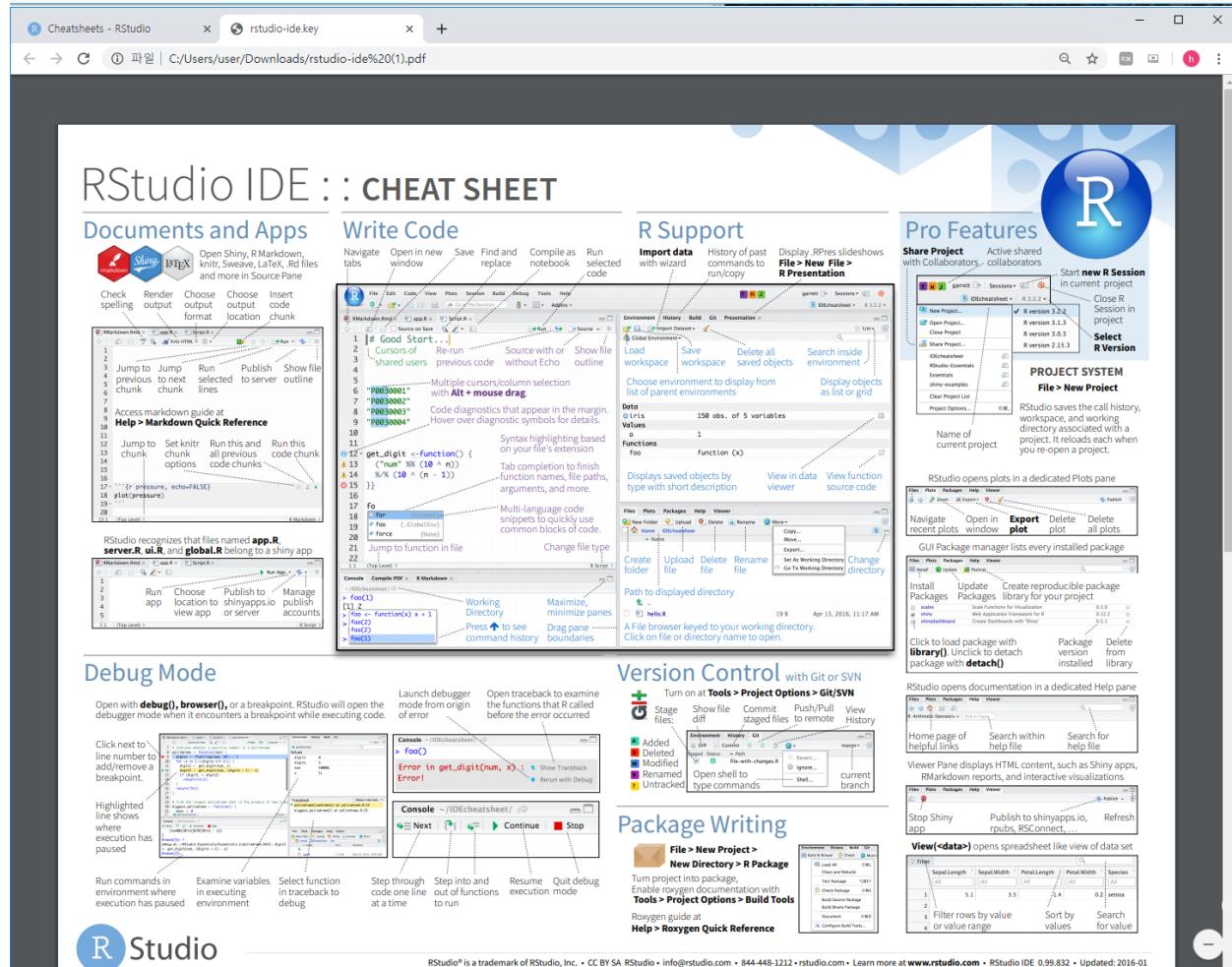
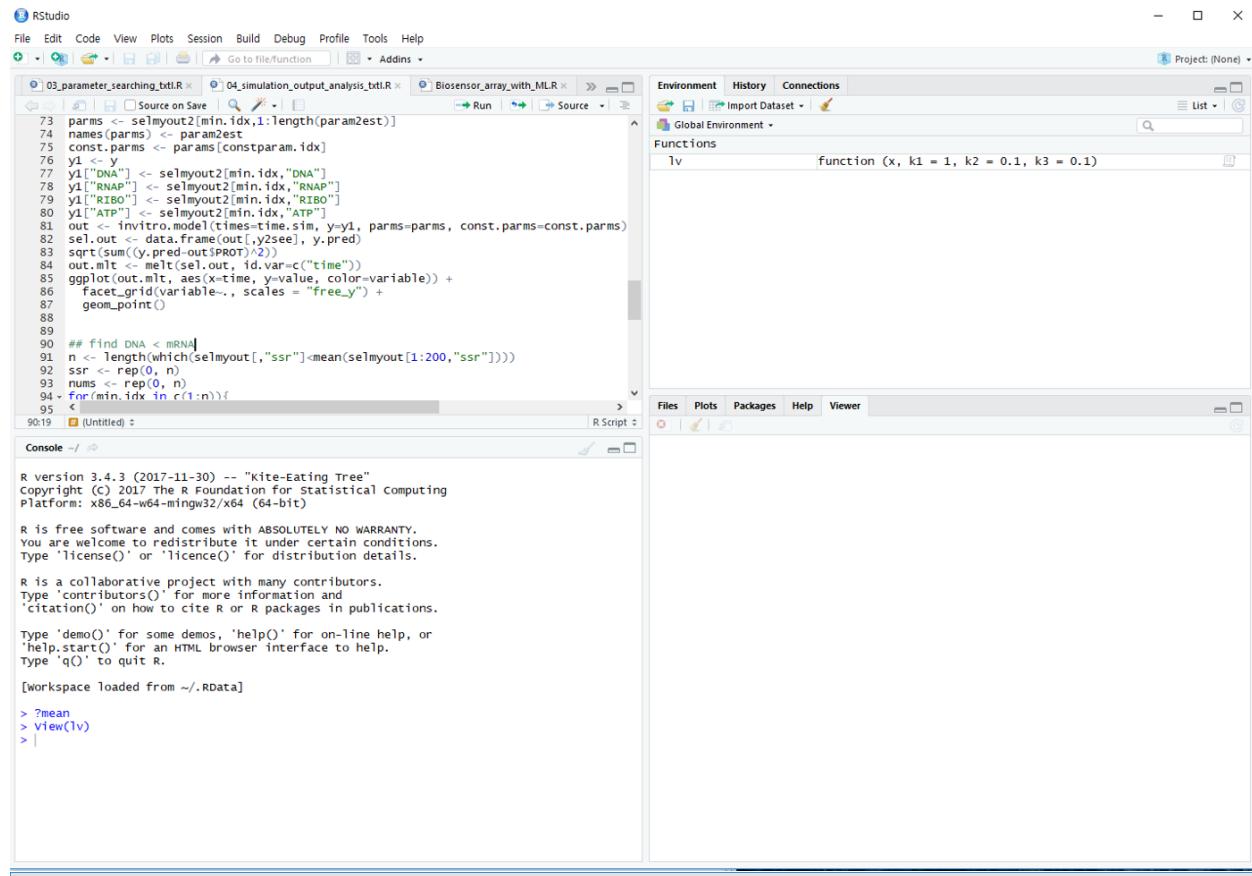
```
x <- c(10, 20, 30)
x + x
sqrt(x)
sin(x)
log(x)
x-mean(x)
```

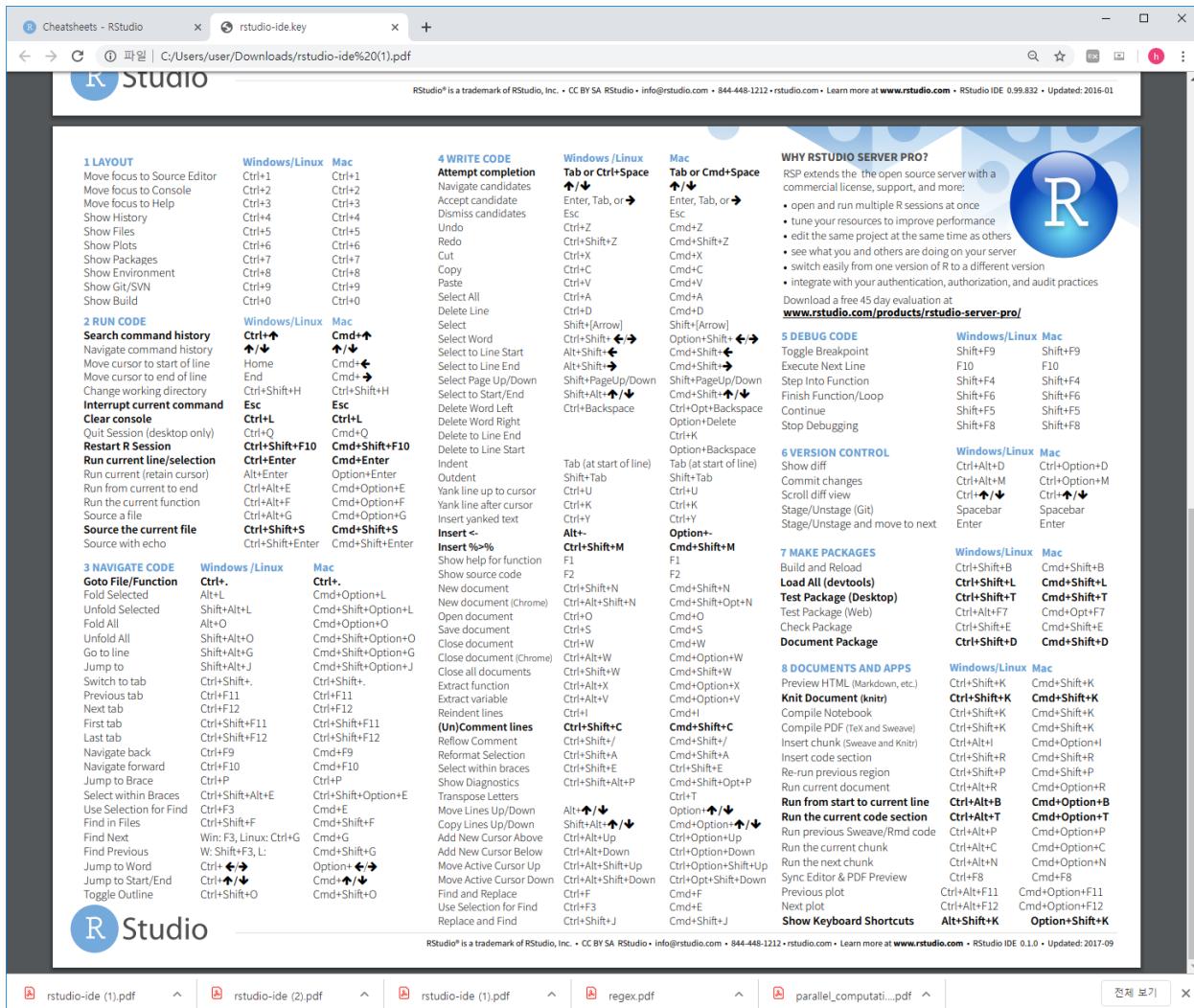
2.13 Help

- R ()

```
?  
?mean  
help("mean")  
example("mean")  
help.search("mean")  
help(package="MASS")
```

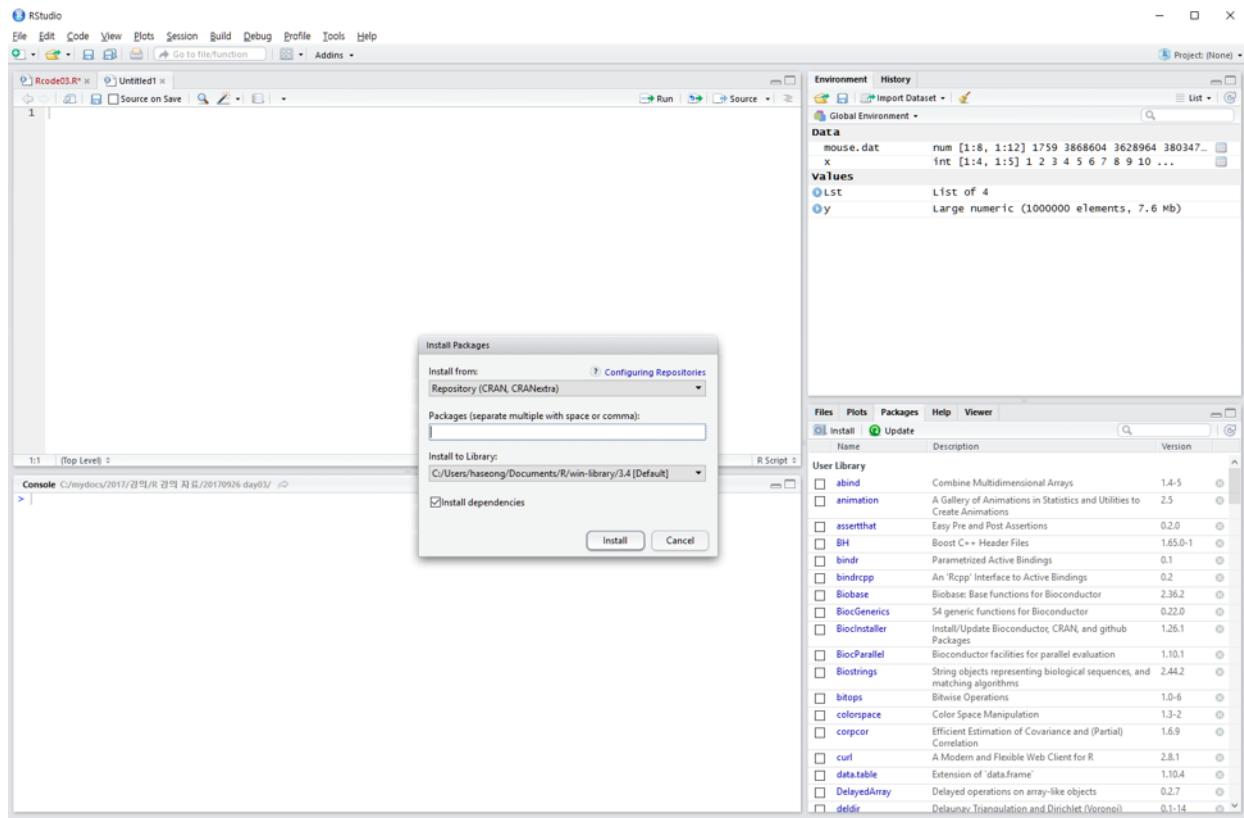

2.14 RStudio workspace





2.15 R packages

- R comes ready loaded with various libraries of functions called packages
- ex) sum() is in the “base” package and sd() in the “stats” package
- The packages can be found in numerous server locations on the web called repositories
- The Comprehensive R Archive Network (CRAN) <http://cran.r-project.org/web/views/>
- Bioconductor specialised in genomics <http://www.bioconductor.org/packages/release/bioc/>



- UsingR package installation

Packages → Install

The screenshot shows the RStudio interface. On the left, the 'Install Packages' dialog is open, showing the user has selected 'CRAN, CRANextra' as the repository and entered 'UsingR' as the package name. The 'Install dependencies' checkbox is checked. Below the dialog, the R console window displays the command and output for installing the 'UsingR' package, including the download of 'HistData_0.8-4.zip' and 'UsingR_2.0-6.zip', and the successful unpacking of both packages.

Install Packages

Install from: [Configuring Repositories](#)

Repository (CRAN, CRANextra)

Packages (separate multiple with space or comma): UsingR

Install to Library: C:/Users/user/Documents/R/win-library/3.4 [Default]

Install dependencies

Install Cancel

```
> install.packages("UsingR")
Installing package into 'C:/Users/user/Documents/R/win-library/3.4'
(as 'lib' is unspecified)
also installing the dependency 'HistData'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/HistData_0.8-4.zip'
Content type 'application/zip' length 359785 bytes (351 KB)
downloaded 351 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/UsingR_2.0-6.zip'
Content type 'application/zip' length 2081603 bytes (2.0 MB)
downloaded 2.0 MB

package 'HistData' successfully unpacked and MD5 sums checked
package 'UsingR' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:/Users/user/AppData/Local/Temp/RtmpwdxEQ7/downloaded_packages
> |
```

Packages

Files Plots Packages Help Viewer

Install Update

Name	Description
UsingR	Data Sets, Etc. for the Text "Using R for Introductory Statistics", Second Edition
ggplot2	Create Elegant Data Visualisations Using the Grammar of Graphics
munsell	Utilities for Using Munsell Colours
rsbml	R support for SBML, using libsbml
stats4	Statistical Functions using S4 Classes

```
> library("UsingR", lib.loc="~/R/win-library/3.4")
필요한 패키지를 로딩중입니다: MASS
필요한 패키지를 로딩중입니다: HistData
필요한 패키지를 로딩중입니다: Hmisc
필요한 패키지를 로딩중입니다: lattice
필요한 패키지를 로딩중입니다: survival
필요한 패키지를 로딩중입니다: Formula
필요한 패키지를 로딩중입니다: ggplot2

다음의 패키지를 부착합니다: 'Hmisc'

The following objects are masked from 'package:base':

  format.pval, units

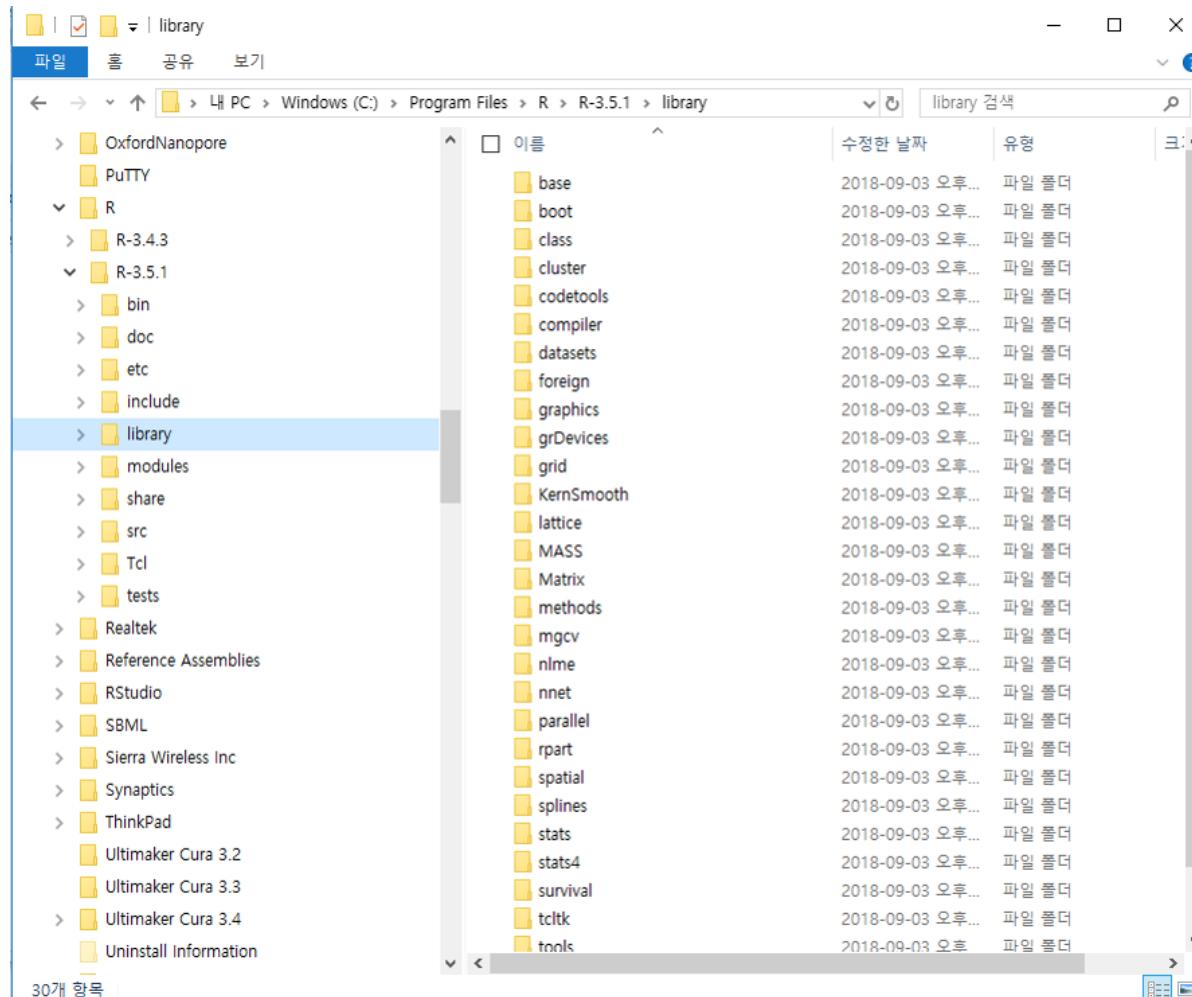
다음의 패키지를 부착합니다: 'UsingR'
```

- UsingR package loading

```
library(UsingR)
```

- R
- R

```
.libPaths()
path.package()
```



2.16 Data sets

- Packages include accompanying data sets
- R has a datasets package that is loaded automatically
- The data function produces a copy of dataset in user's workspace

```
head(rivers)
length(rivers)
class(rivers)
data(rivers)
data(package="UsingR")
library(HistData)
head(Cavendish)
str(Cavendish)
head(Cavendish$density2)
```


2.17 Cheatsheet

Base R Cheat Sheet

Getting Help

Accessing the help files

?mean
Get help of a particular function.
help.search('weighted mean')
Search the help files for a word or phrase.
help(package = 'dplyr')
Find help for a package.

More about an object

str(iris)
Get a summary of an object's structure.
class(iris)
Find the class an object belongs to.

Using Packages

install.packages('dplyr')
Download and install a package from CRAN.

library(dplyr)
Load the package into the session, making all its functions available to use.

dplyr::select
Use a particular function from a package.

data(iris)
Load a built-in dataset into the environment.

Working Directory

getwd()
Find the current working directory (where inputs are found and outputs are sent).

setwd('C://file/path')
Change the current working directory.

Use projects in RStudio to set the working directory to the folder you are working in.

Vectors			Programming		
Creating Vectors		For Loop		While Loop	
c(2, 4, 6)	2 4 6	Join elements into a vector.	for (variable in sequence){ Do something }	while (condition){ Do something }	Example
2:6	2 3 4 5 6	An integer sequence	Example	Example	
seq(2, 3, by=0.5)	2.0 2.5 3.0	A complex sequence	for (i in 1:4){ j <- i + 10 print(j) }	while (i < 5){ print(i) i <- i + 1 }	
rep(1:2, times=3)	1 2 1 2 1 2	Repeat a vector			
rep(1:2, each=3)	1 1 1 2 2 2	Repeat elements of a vector			
Vector Functions			If Statements		
sort(x)	rev(x)	Return x sorted. Return x reversed.	if (condition){ Do something } else { Do something different }	function_name <- function(var){ Do something return(new_variable) }	Example
table(x)	unique(x)	See counts of values. See unique values.	Example	Example	
Selecting Vector Elements			Functions		
By Position			Reading and Writing Data		
x[4]	The fourth element.		Input	Output	Description
x[-4]	All but the fourth.		df <- read.table('file.txt')	write.table(df, 'file.txt')	Read and write a delimited text file.
x[2:4]	Elements two to four.		df <- read.csv('file.csv')	write.csv(df, 'file.csv')	Read and write a comma separated value file. This is a special case of read.table/write.table.
x[-(2:4)]	All elements except two to four.		load('file.RData')	save(df, file = 'file.Rdata')	Read and write an R data file, a file type special for R.
x[c(1, 5)]	Elements one and five.				
By Value					
x[x == 10]	Elements which are equal to 10.				
x[x < 0]	All elements less than zero.				
x[x %in% c(1, 2, 5)]	Elements in the set 1, 2, 5.				
Named Vectors					
x['apple']	Element with name 'apple'.				
Conditions					
a == b	Are equal	a > b	Greater than	a >= b	Greater than or equal to
a != b	Not equal	a < b	Less than	a <= b	Less than or equal to
				is.na(a)	Is missing
				is.null(a)	Is null

RStudio® is a trademark of RStudio, Inc. • CC BY Mhairi McNeill • mhairi@mcneill@gmail.com

Learn more at [web page](#) or [vignette](#) • package version • Updated: 3/15

Types		
Converting between common data types in R. Can always go from a higher value in the table to a lower value.		
as.logical	TRUE, FALSE, TRUE	Boolean values (TRUE or FALSE).
as.numeric	1, 0, 1	Integers or floating point numbers.
as.character	'1', '0', '1'	Character strings. Generally preferred to factors.
as.factor	'1', '0', '1', levels: '1', '0'	Character strings with present levels. Needed for some statistical models.

Matrices		
<code>m <- matrix(x, nrow = 3, ncol = 3)</code>		Create a matrix from x.
<code>m[2,]</code>	- Select a row	<code>t(m)</code>
<code>m[, 1]</code>	- Select a column	Transpose
<code>m[2, 3]</code>	- Select an element	<code>m %*% n</code>
		Matrix Multiplication
		<code>solve(m, n)</code>
		Find x in m: x = n

Strings		
Also see the string package.		Also see the string package.
<code>paste(x, y, sep = ' ')</code>	Join multiple vectors together.	
<code>paste(x, collapse = ' ')</code>	Join elements of a vector together.	
<code>grep(pattern, x)</code>	Find regular expression matches in x.	
<code>gsub(pattern, replace, x)</code>	Replace matches in x with a string.	
<code>toupper(x)</code>	Convert to uppercase.	
<code>tolower(x)</code>	Convert to lowercase.	
<code>nchar(x)</code>	Number of characters in a string.	

Maths Functions		
<code>log(x)</code>	Natural log.	<code>sum(x)</code>
<code>exp(x)</code>	Exponential.	<code>mean(x)</code>
<code>max(x)</code>	Largest element.	<code>Median.</code>
<code>min(x)</code>	Smallest element.	<code>Percentage quantiles.</code>
<code>round(x, n)</code>	Round to n decimal places.	<code>rank(x)</code>
<code>signif(x, n)</code>	Round to n significant figures.	<code>var(x)</code>
<code>cor(x, y)</code>	Correlation.	<code>sd(x)</code>

Lists		
<code>l <- list(x = 1:5, y = c('a', 'b'))</code>		
A list is a collection of elements which can be of different types.		
<code>l[1:2]</code>	New list with only the first element.	<code>l[[1]]</code>
<code>l[[1]]</code>	Second element of l.	Element named x.
		New list with only element named y.

Factors		
<code>factor(x)</code>		Turn a vector into a factor. Can set the levels of the factor and the order.
<code>cut(x, breaks = 4)</code>	Turn a numeric vector into a factor by 'cutting' into sections.	

Variable Assignment		
<code>> a <- 'apple'</code>		
<code>> a</code>		
<code>[1] 'apple'</code>		

Data Frames		
<code>df <- data.frame(x = 1:3, y = c('a', 'b', 'c'))</code>		
A special case of a list where all elements are the same length.		
<code>df\$x</code>	<code>df\$y</code>	List subsetting
<code>df[[2]]</code>		

Statistics		
Also see the dplyr package.		
<code>lm(y ~ x, data=df)</code>	Linear model.	
<code>glm(y ~ x, data=df)</code>	Generalised linear model.	
<code>summary</code>	Get more detailed information out a model.	
<code>t.test(x, y)</code>	Perform a t-test for difference between means.	
<code>pairwise.t.test</code>	Perform a t-test for paired data.	
<code>prop.test</code>	Test for a difference between proportions.	
<code>aov</code>	Analysis of variance.	

The Environment		
<code>ls()</code>	List all variables in the environment.	
<code>rm(x)</code>	Remove x from the environment.	
<code>rm(list = ls())</code>	Remove all variables from the environment.	
You can use the environment panel in RStudio to		

Distributions			
Random Variates	Density Function	Cumulative Distribution	Quantile
Normal	<code>rnorm</code>	<code>dnorm</code>	<code>pnorm</code>
Poisson	<code>rpois</code>	<code>dpois</code>	<code>ppois</code>
Binomial	<code>rbinom</code>	<code>dbinom</code>	<code>pbinom</code>
Uniform	<code>runif</code>	<code>dunif</code>	<code>qunif</code>

Plotting		
Also see the ggplot2 package.		
<code>plot(x)</code>	Values of x in order.	
<code>plot(x, y)</code>	Values of x against y.	
<code>hist(x)</code>	Histogram of x.	