



Introduction to **R** Programming

Ali Ahani

(a_ahani@sbu.ac.ir)

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



R

- ▶ A language and environment for statistical computing and graphics
- ▶ Similar to the **S** language and environment: some important differences, but much code written for **S** runs unaltered under **R**
- ▶ Available as Free Software under the terms of the Free Software Foundation's GNU
- ▶ It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

(<https://www.r-project.org/>)

R

- A wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques
- Highly extensible
- The ease with well-designed publication-quality

(<https://www.r-project.org/>)

R Environment

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes:

- ▶ an effective data handling and storage facility,
- ▶ a suite of operators for calculations on arrays, in particular matrices,
- ▶ a large, coherent, integrated collection of intermediate tools for data analysis,

(<https://www.r-project.org/>)

R Environment

- ▶ graphical facilities for data analysis and display either on-screen or on hardcopy, and
- ▶ a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The term “environment” is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools.

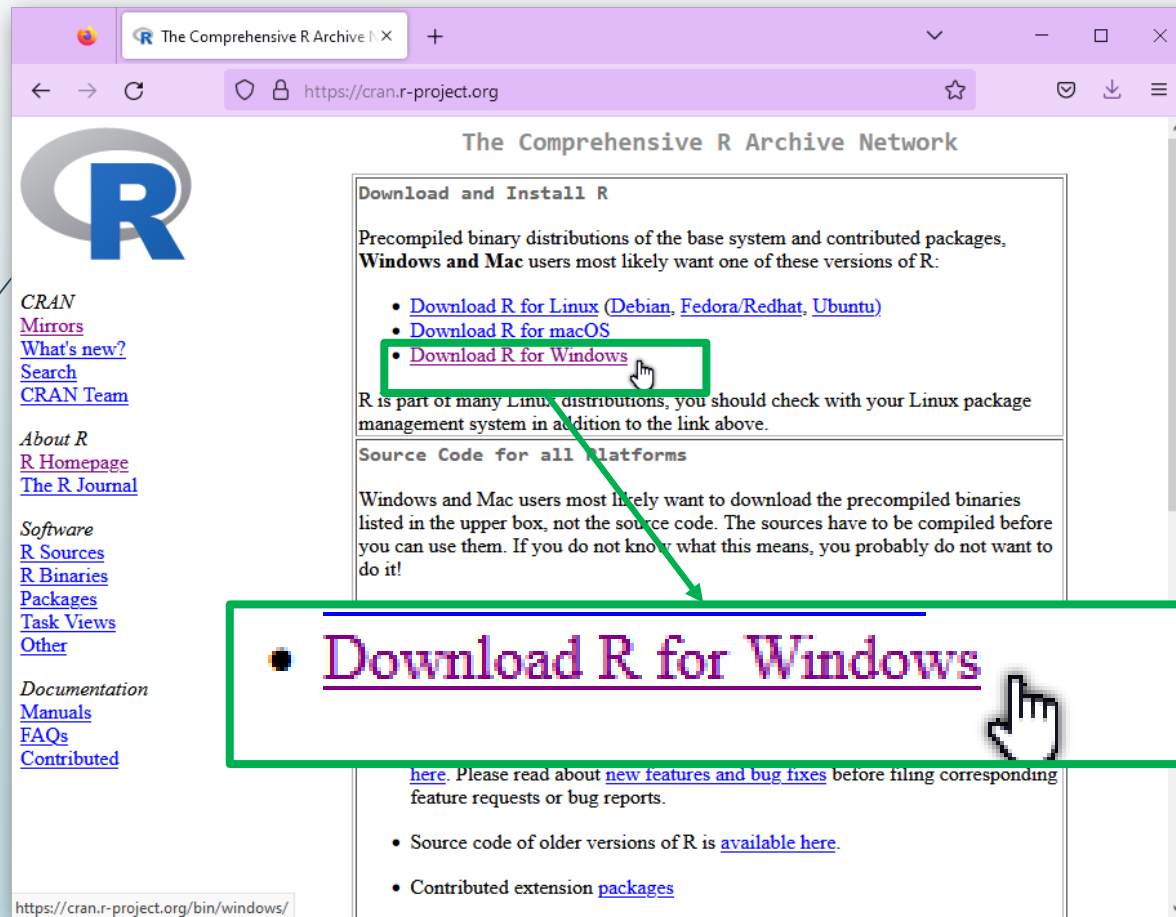
(<https://www.r-project.org/>)

R Contributors

- The current **R** is the result of a collaborative effort with contributions from all over the world. **R** was initially written by **Robert Gentleman** and **Ross Ihaka** also known as “**R & R**” of the Statistics Department of the University of Auckland. Since mid-1997 there has been a core group, the **R** Core Team, with write access to the **R** source. (<https://www.r-project.org/>)

Downloading the Installation File

➡ <https://cran.r-project.org/>



The screenshot shows the CRAN website with the following content:

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 \(Debian, Fedora/Redhat, Ubuntu\)](#)
- [Download R for macOS](#)
- [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!

Download R for Windows

[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](#)

<https://cran.r-project.org/bin/windows/>

Downloading the Installation File

➡ <https://cran.r-project.org/>

The screenshot shows the CRAN website for Windows binaries. The page title is "R for Windows". Under "Subdirectories:", there are links for "base", "contrib", "old contrib", and "Rtools". The "base" link is circled in green, and a green arrow points from it to a larger green circle at the bottom of the page containing the text "install R for the first time".

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 >= 3.4.x).
- [old contrib](#) Binaries of contributed CRAN packages for outdated versions of R (for R < 3.4.x).
- [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.

install R for the first time

Downloading the Installation File

➡ <https://cran.r-project.org/>

The screenshot shows a web browser window with the address bar displaying <https://cran.r-project.org/>. The page title is "R-4.2.2 for Windows". The main content area features a large blue "R" logo on the left and a central box containing the text "Download R-4.2.2 for Windows (76 megabytes, 64 bit)". Below this box are links for "README on the Windows binary distribution" and "New features in this version". The page also includes a section for "Frequently asked questions" with links such as "Does R run under my version of Windows?" and "How do I update...". A green oval highlights the "Download R-4.2.2 for Windows (76 megabytes, 64 bit)" link in the main content area, and another green oval highlights the same link at the bottom of the page. A hand cursor is visible over the bottom link.

Download R-4.2.2 for Windows (76 megabytes, 64 bit)

[README on the Windows binary distribution](#)
[New features in this version](#)

This build requires UCRT, which is part of Windows since Windows 10 and Windows Server 2016. On older systems, UCRT has to be installed manually from [here](#).

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server.

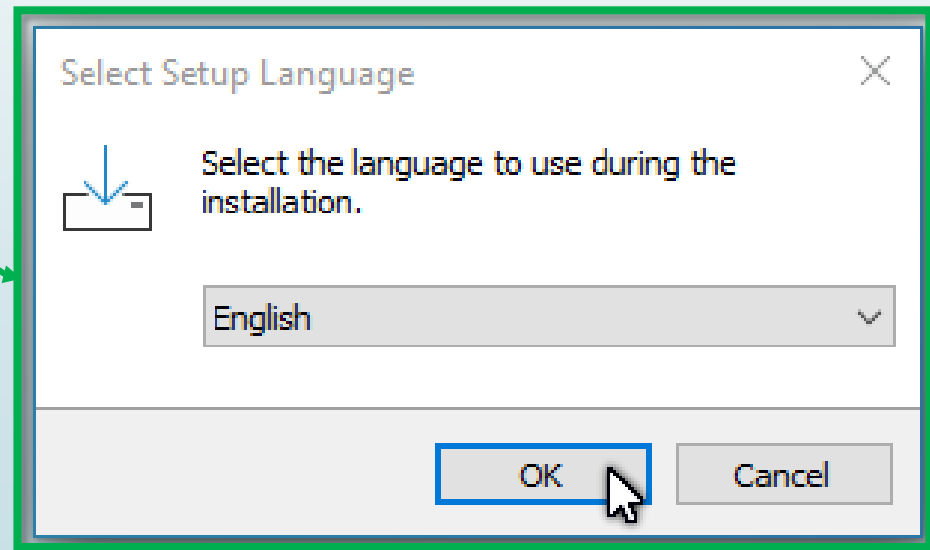
Frequently asked questions

- [Does R run under my version of Windows?](#)
- [How do I update...](#)

Download R-4.2.2 for Windows (76 megabytes, 64 bit)

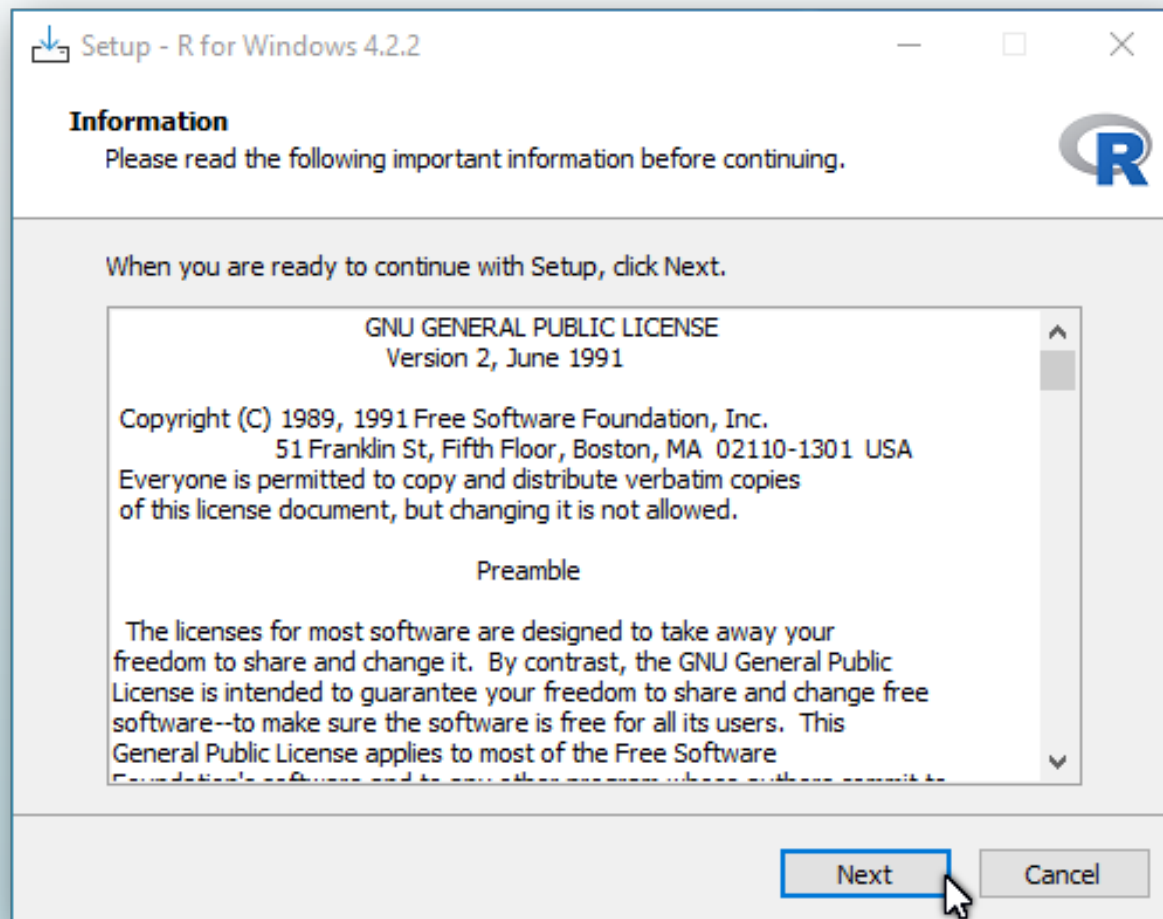
Downloading the Installation File

➤ <https://cran.r-project.org/>



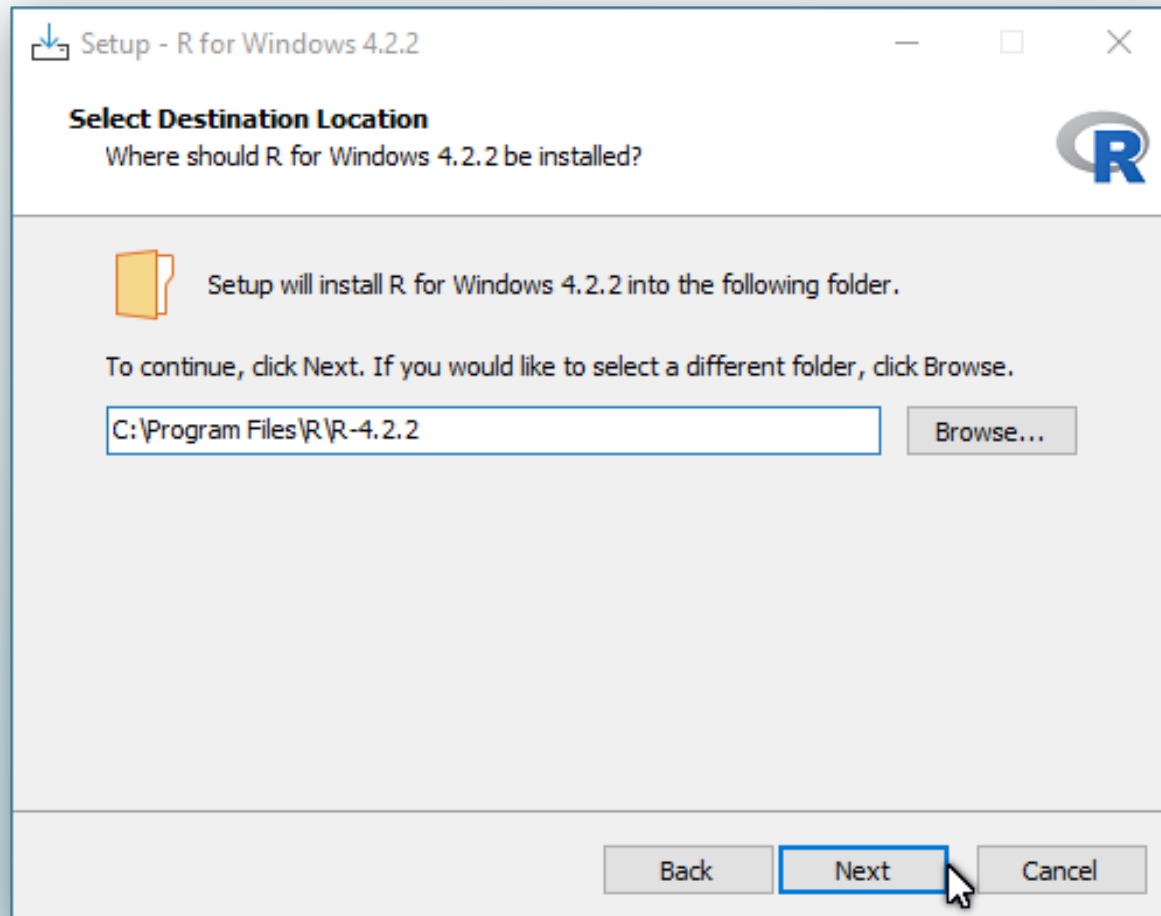
Downloading the Installation File

➤ <https://cran.r-project.org/>



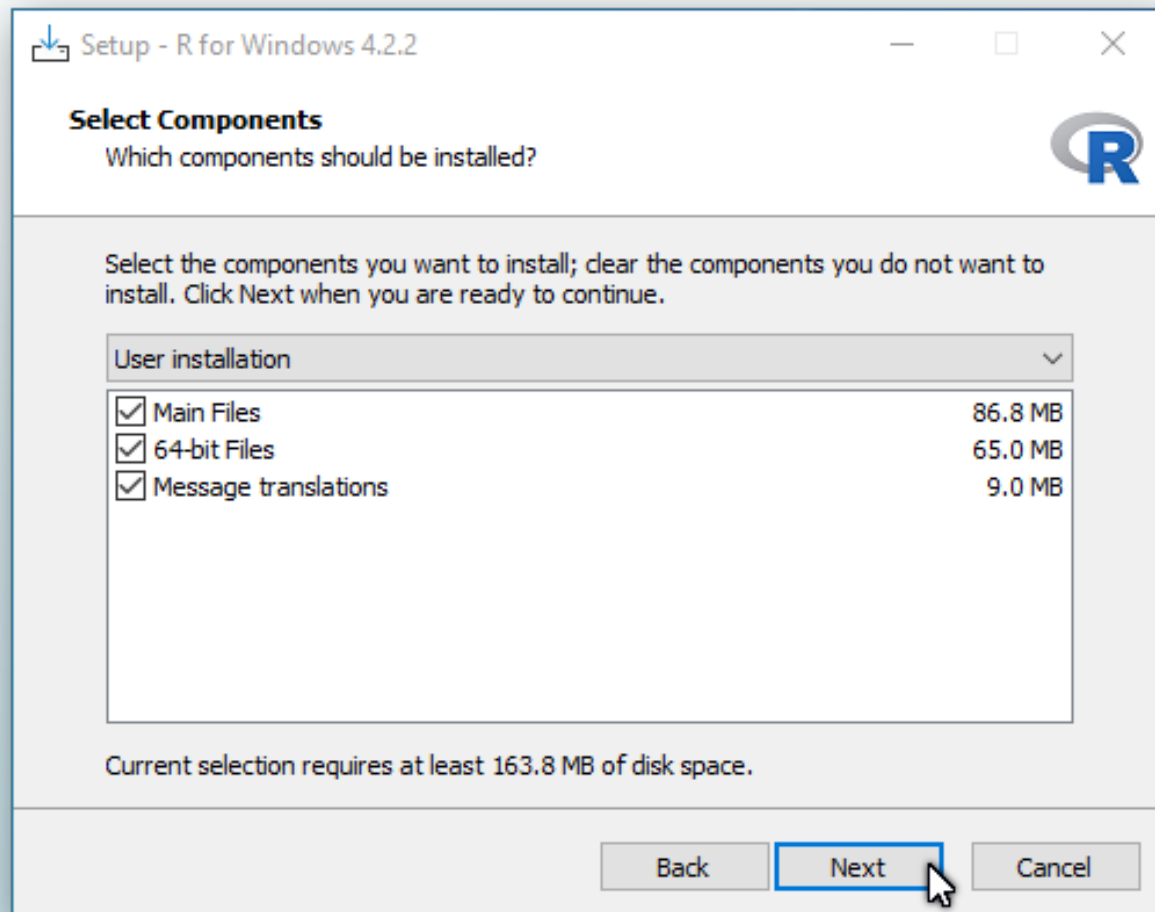
Downloading the Installation File

➤ <https://cran.r-project.org/>



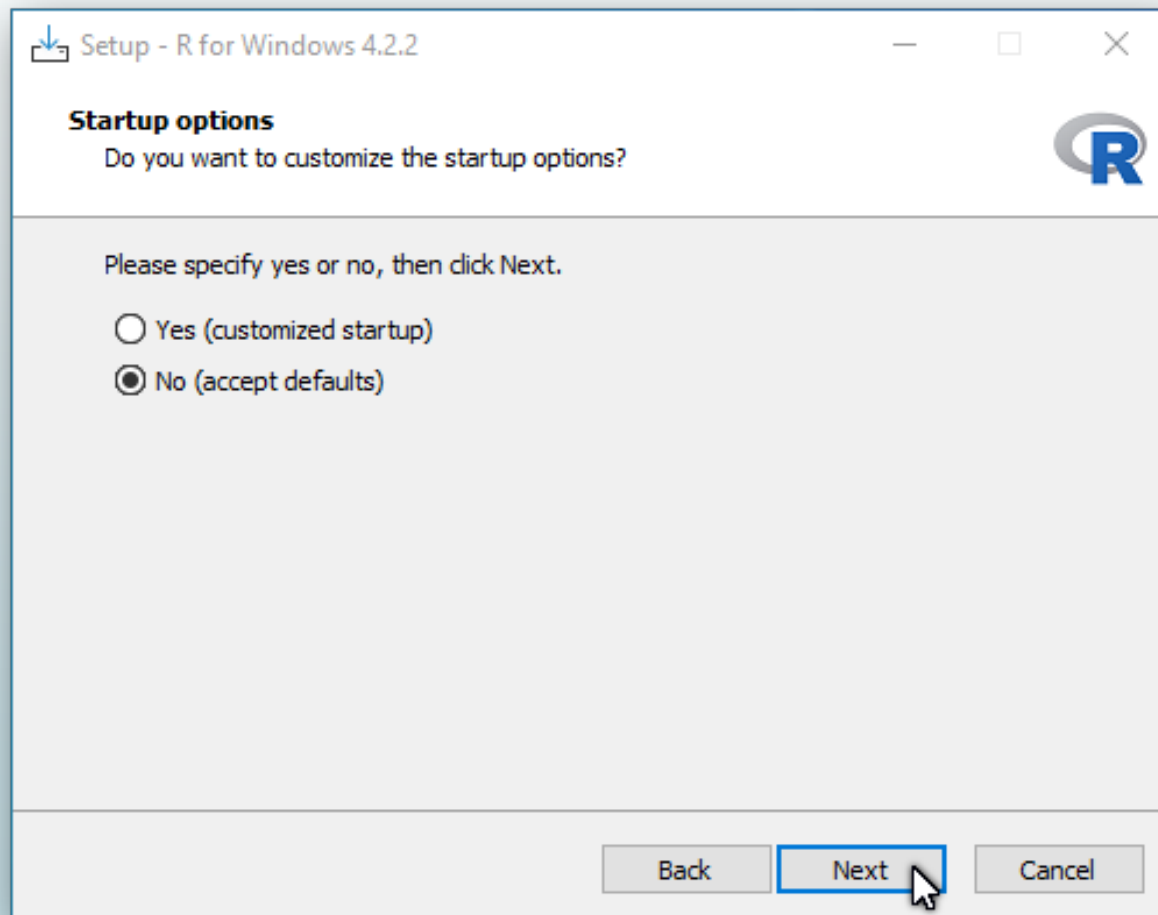
Downloading the Installation File

➤ <https://cran.r-project.org/>



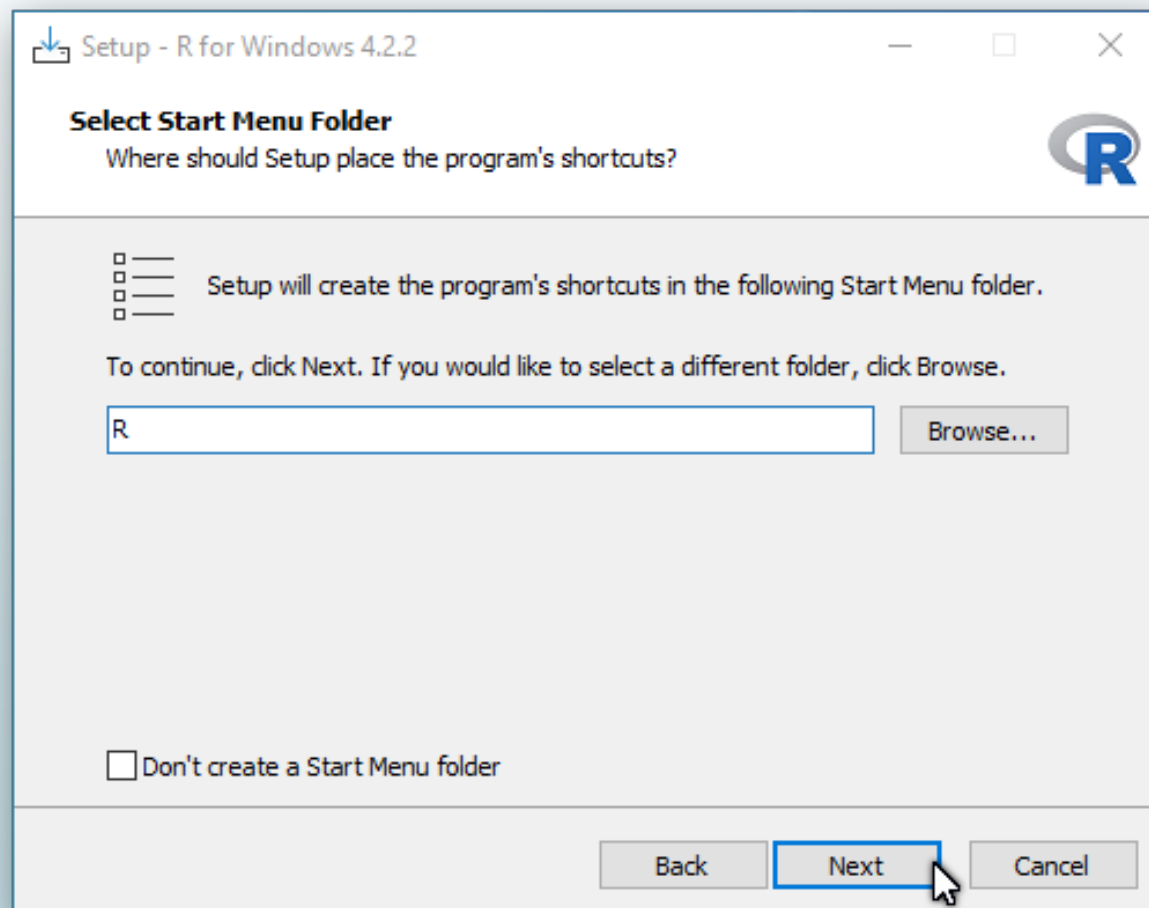
Downloading the Installation File

➤ <https://cran.r-project.org/>



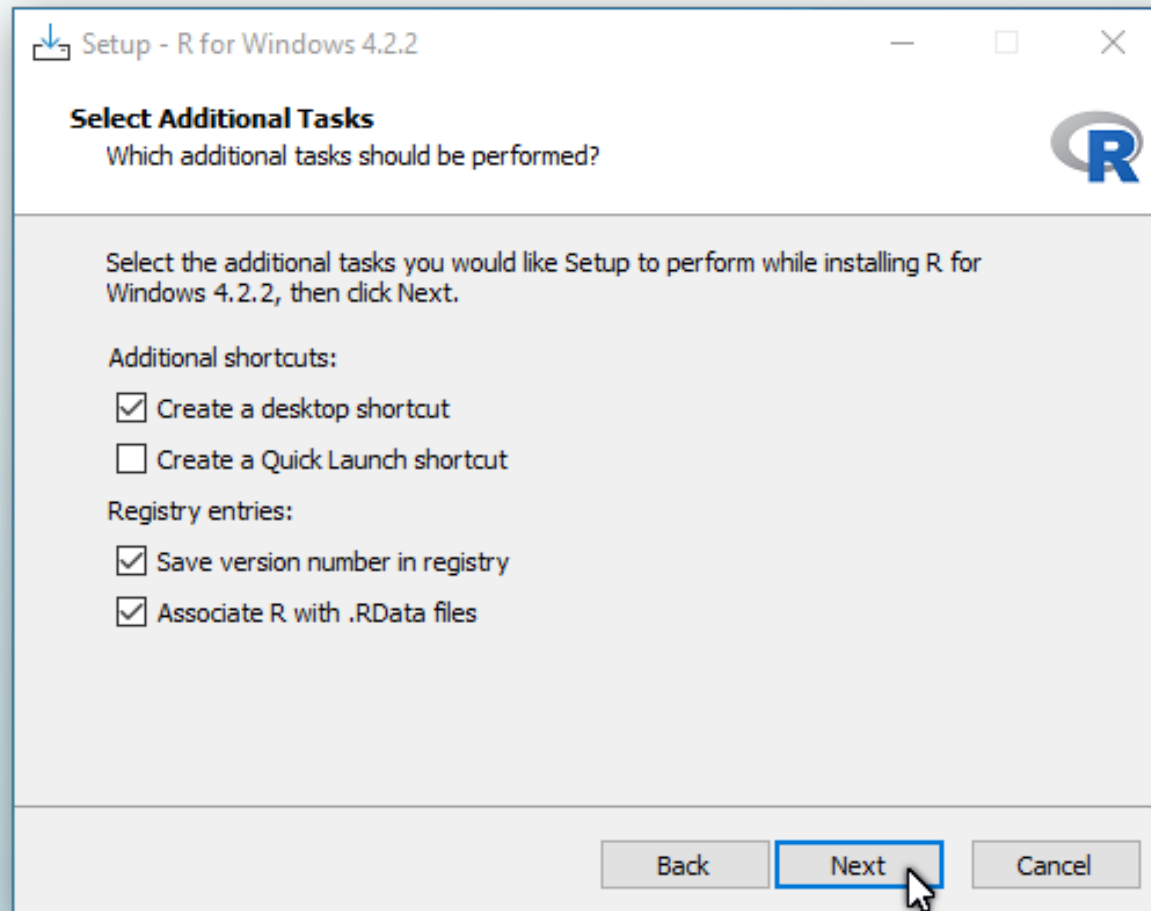
Downloading the Installation File

➤ <https://cran.r-project.org/>



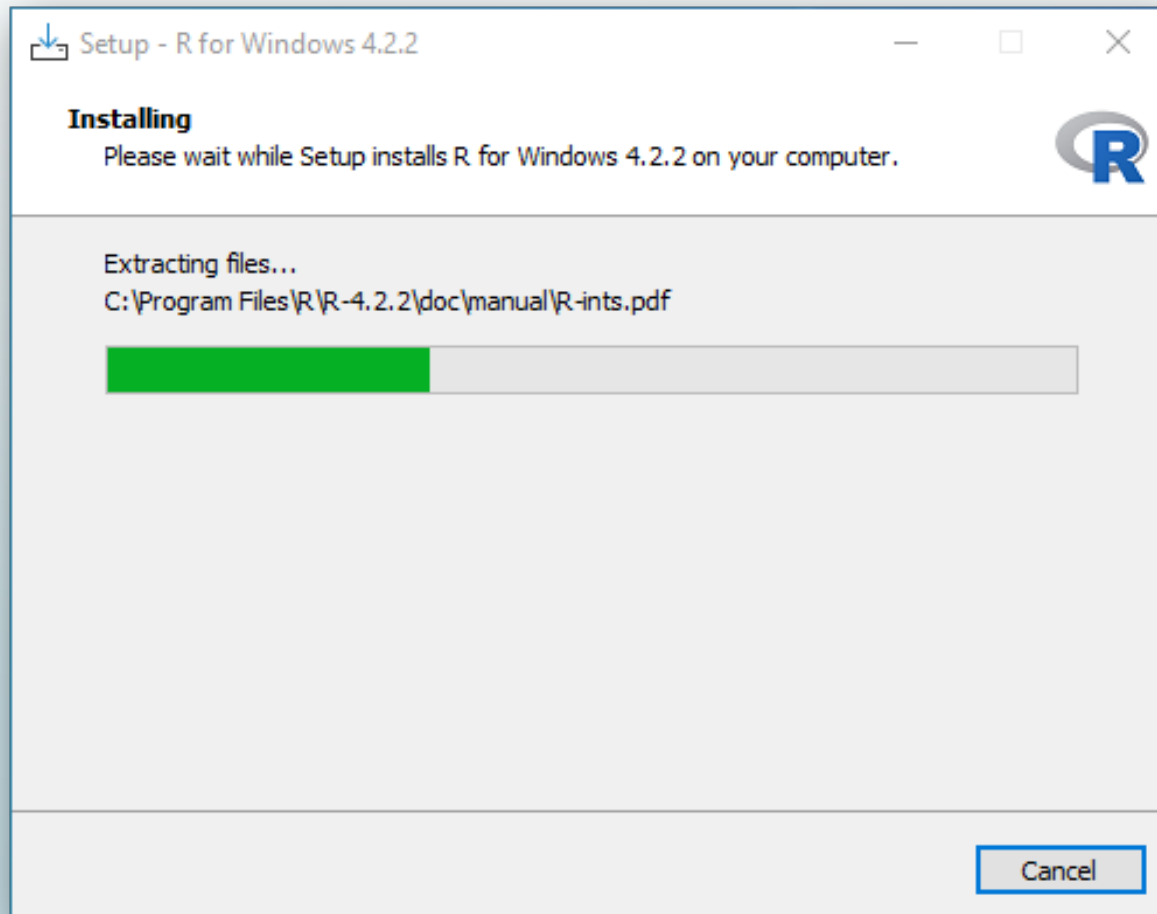
Downloading the Installation File

➤ <https://cran.r-project.org/>



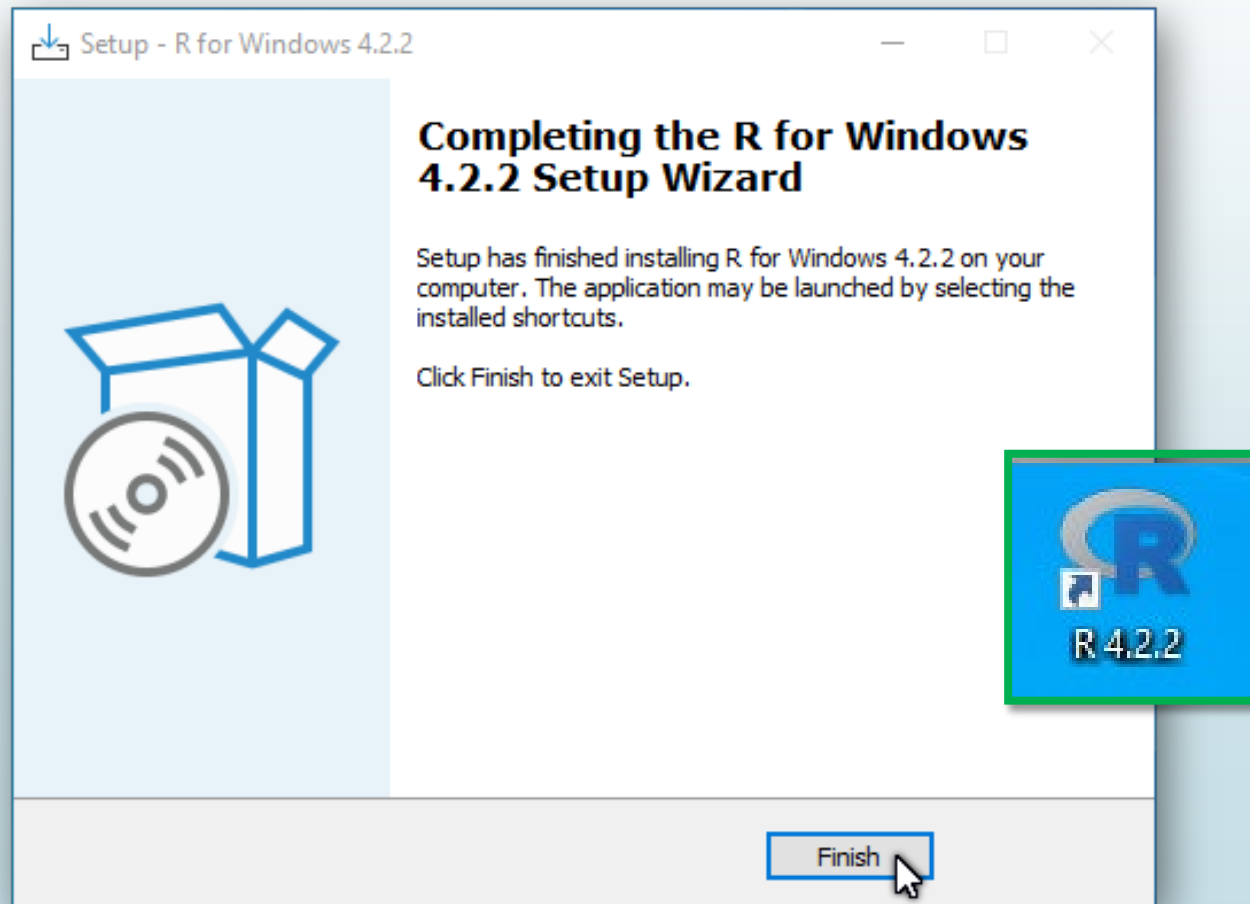
Downloading the Installation File

➡ <https://cran.r-project.org/>

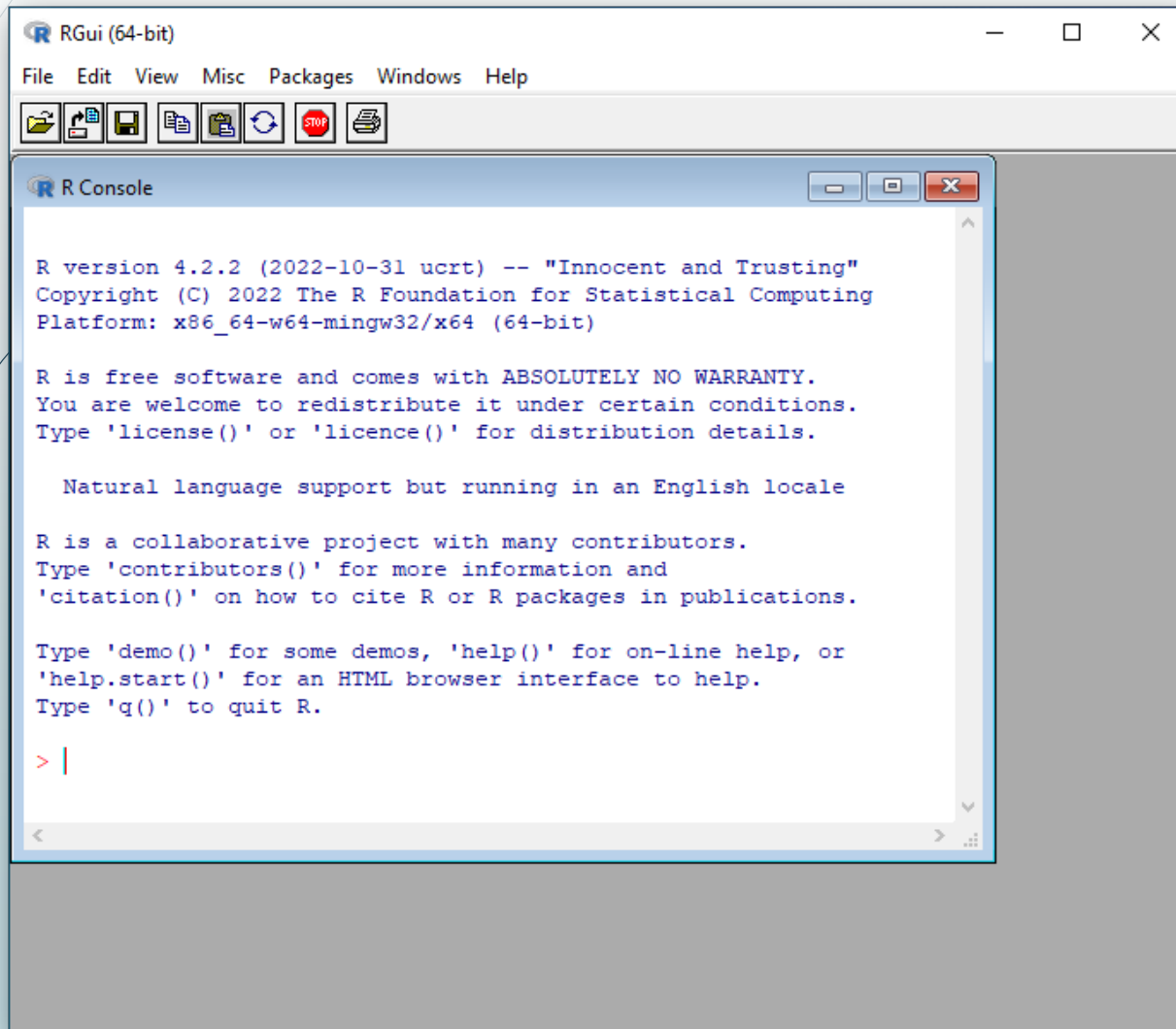


Downloading the Installation File

► <https://cran.r-project.org/>

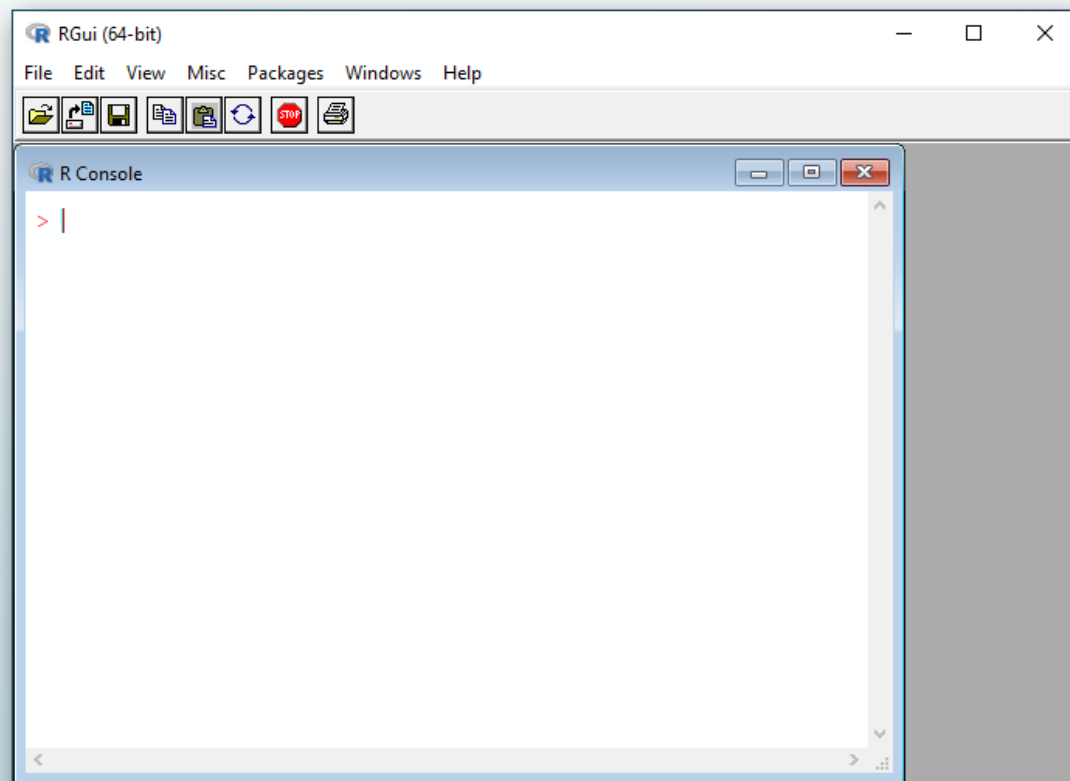


R GUI



R GUI

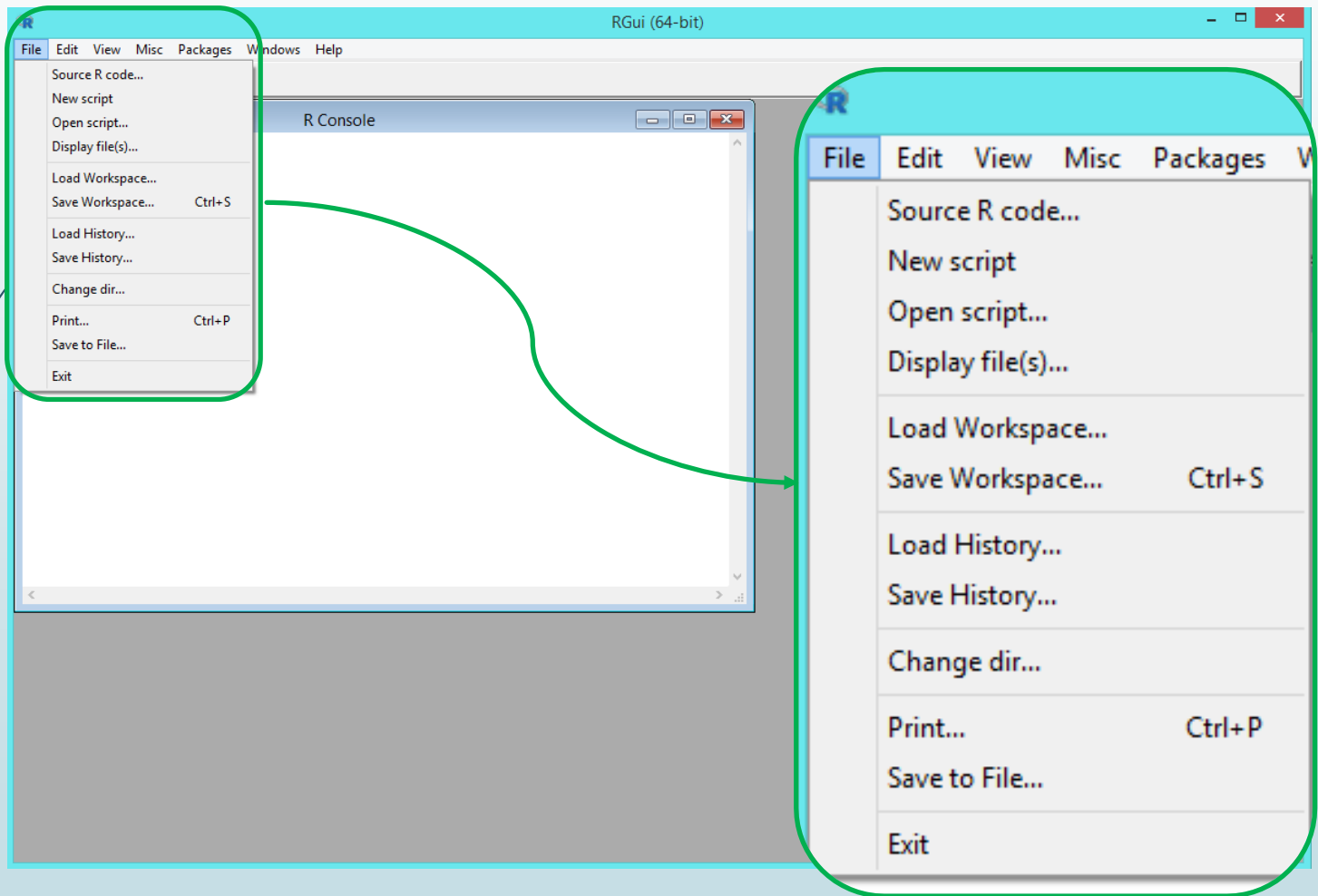
➤ R Console



Clearing R Console by Ctrl+L

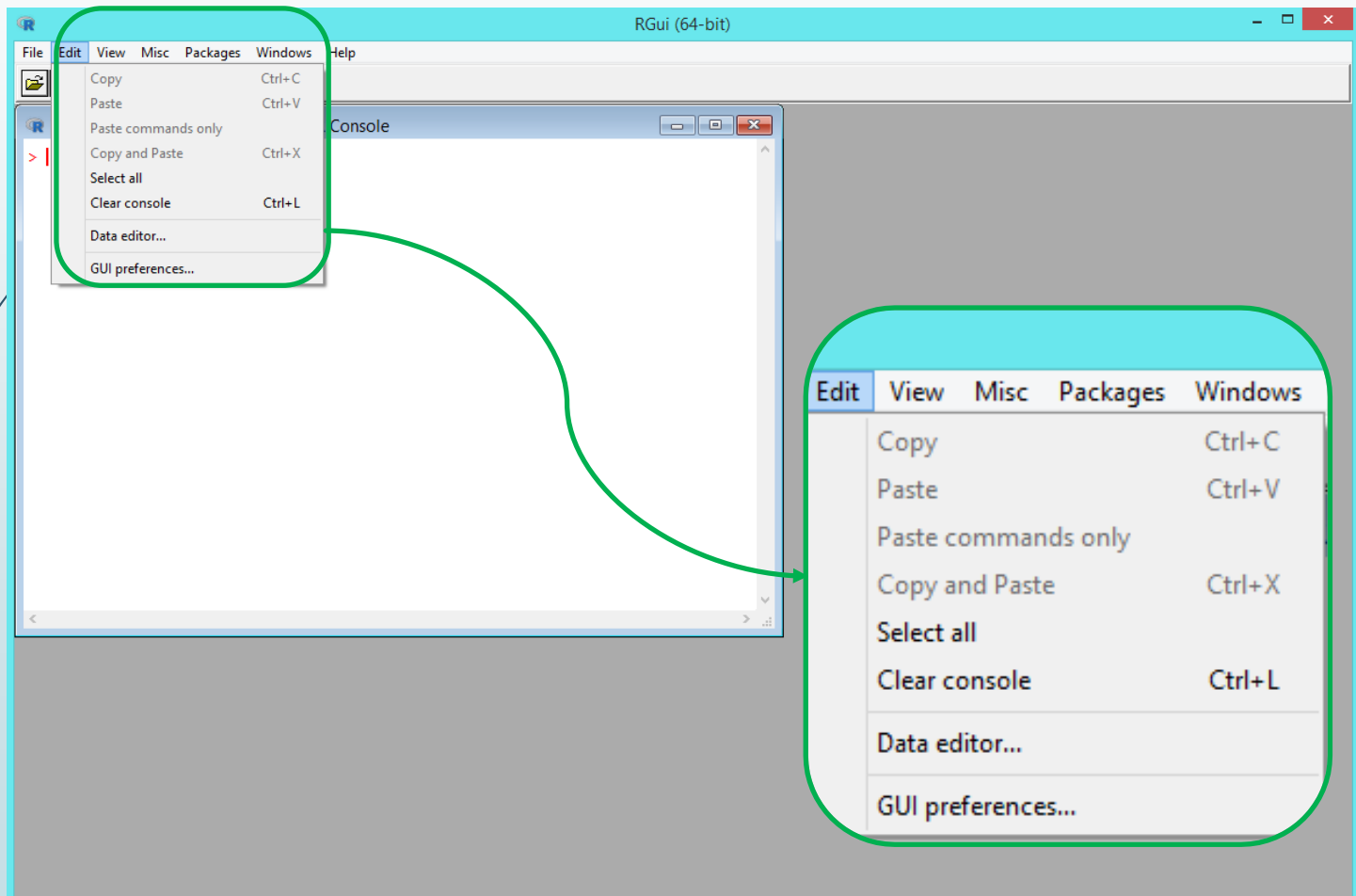
R GUI

► Menu: File



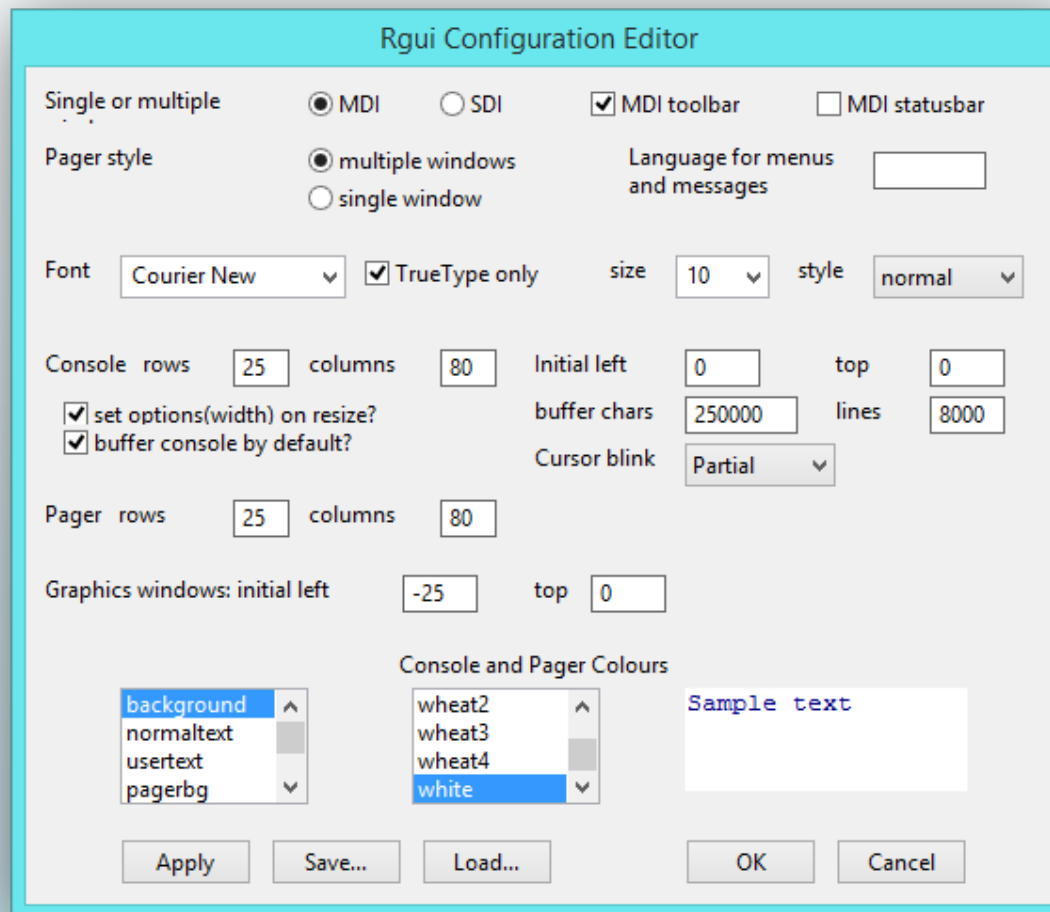
R GUI

Menu: Edit



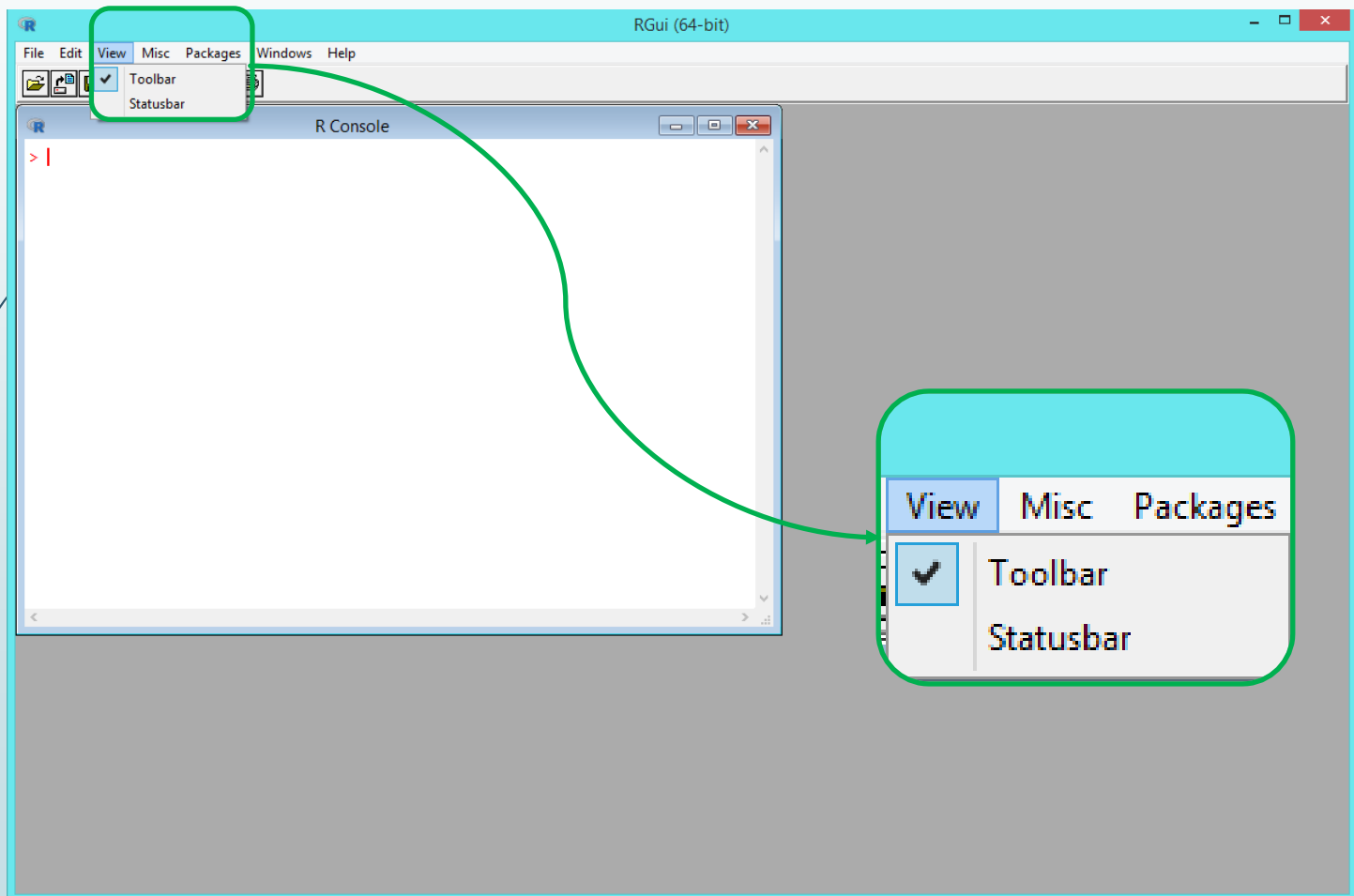
R GUI

➤ Menu: Edit → GUI preferences ...



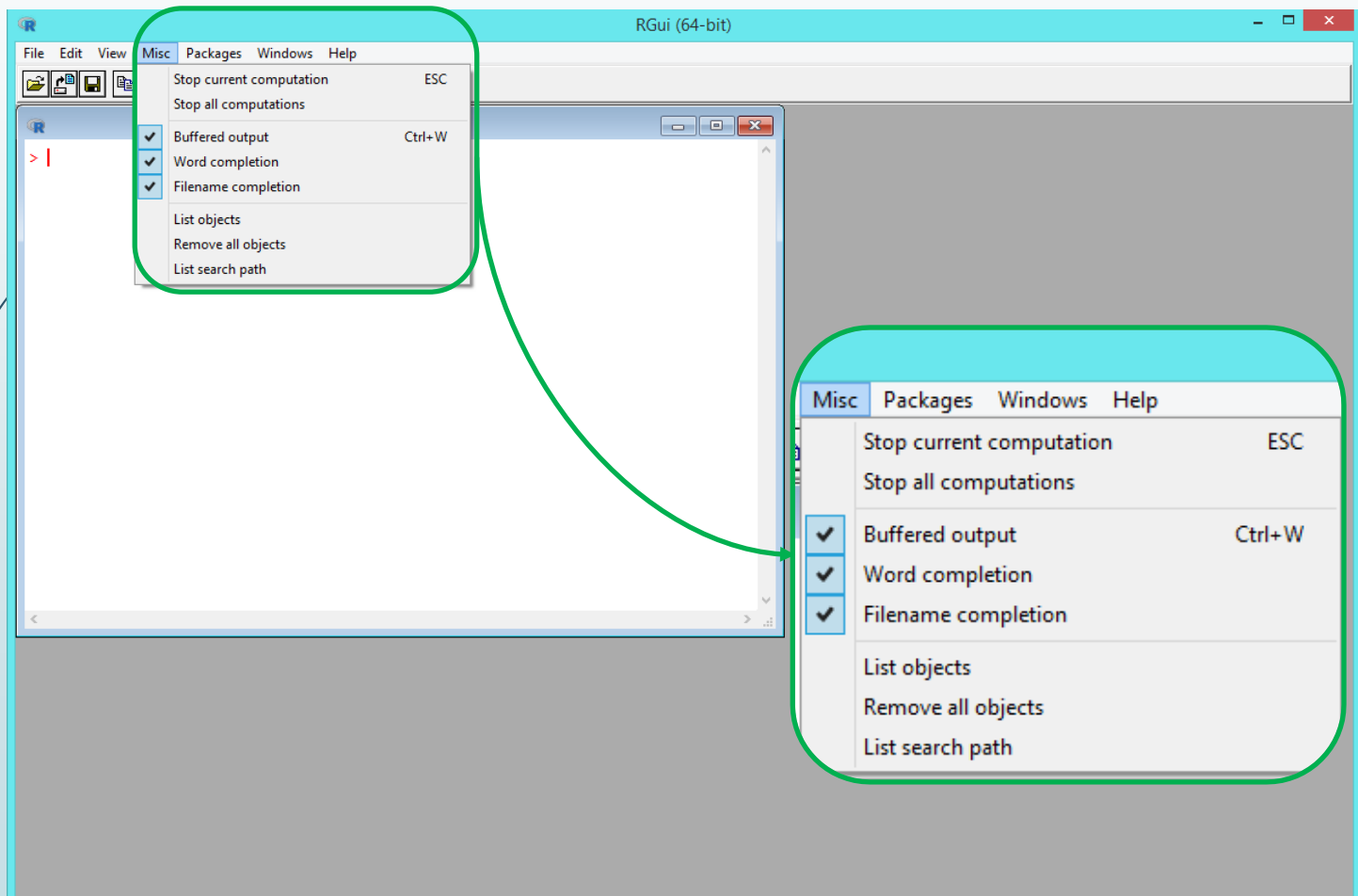
R GUI

► Menu: View



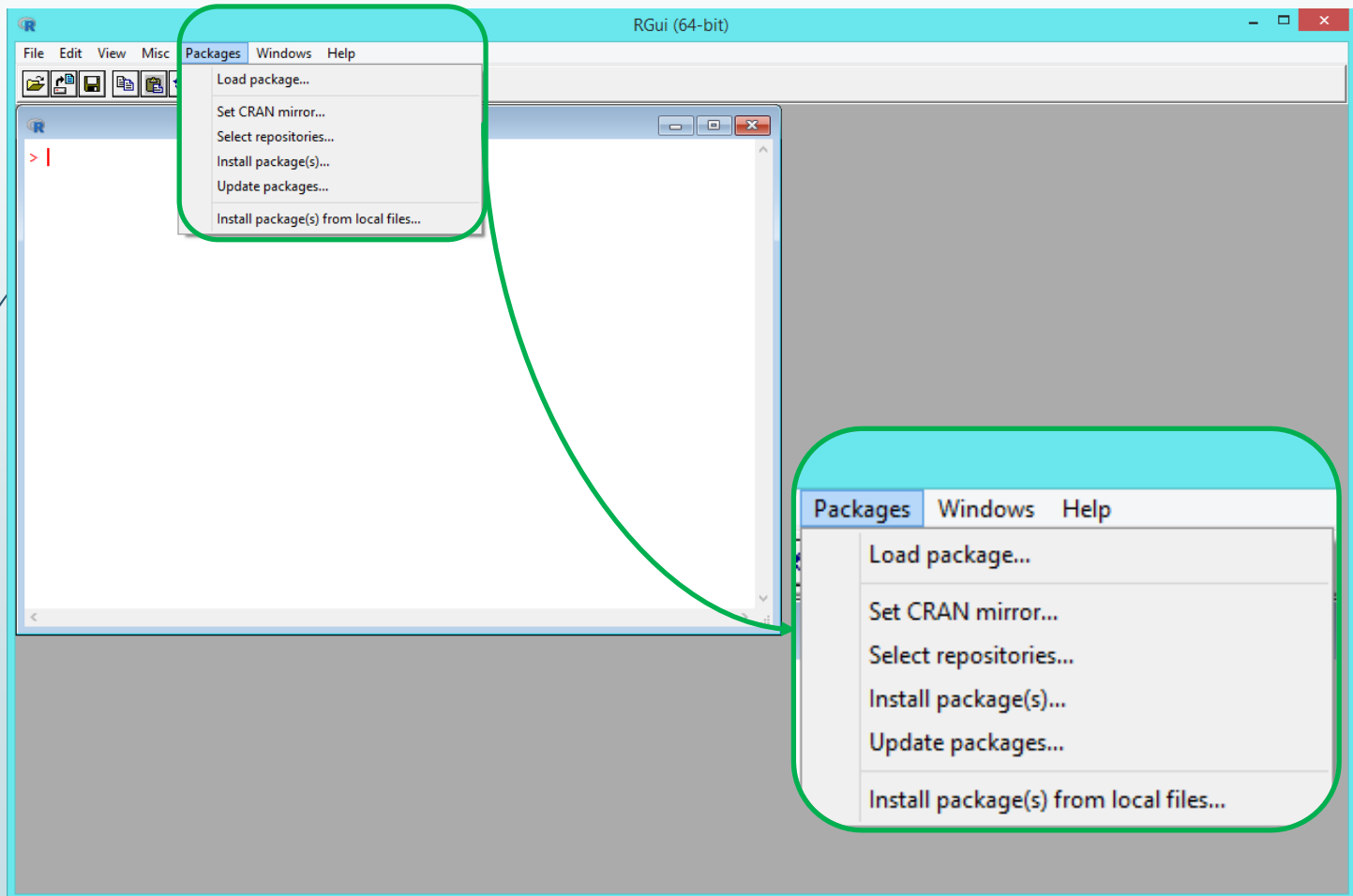
R GUI

Menu: Misc



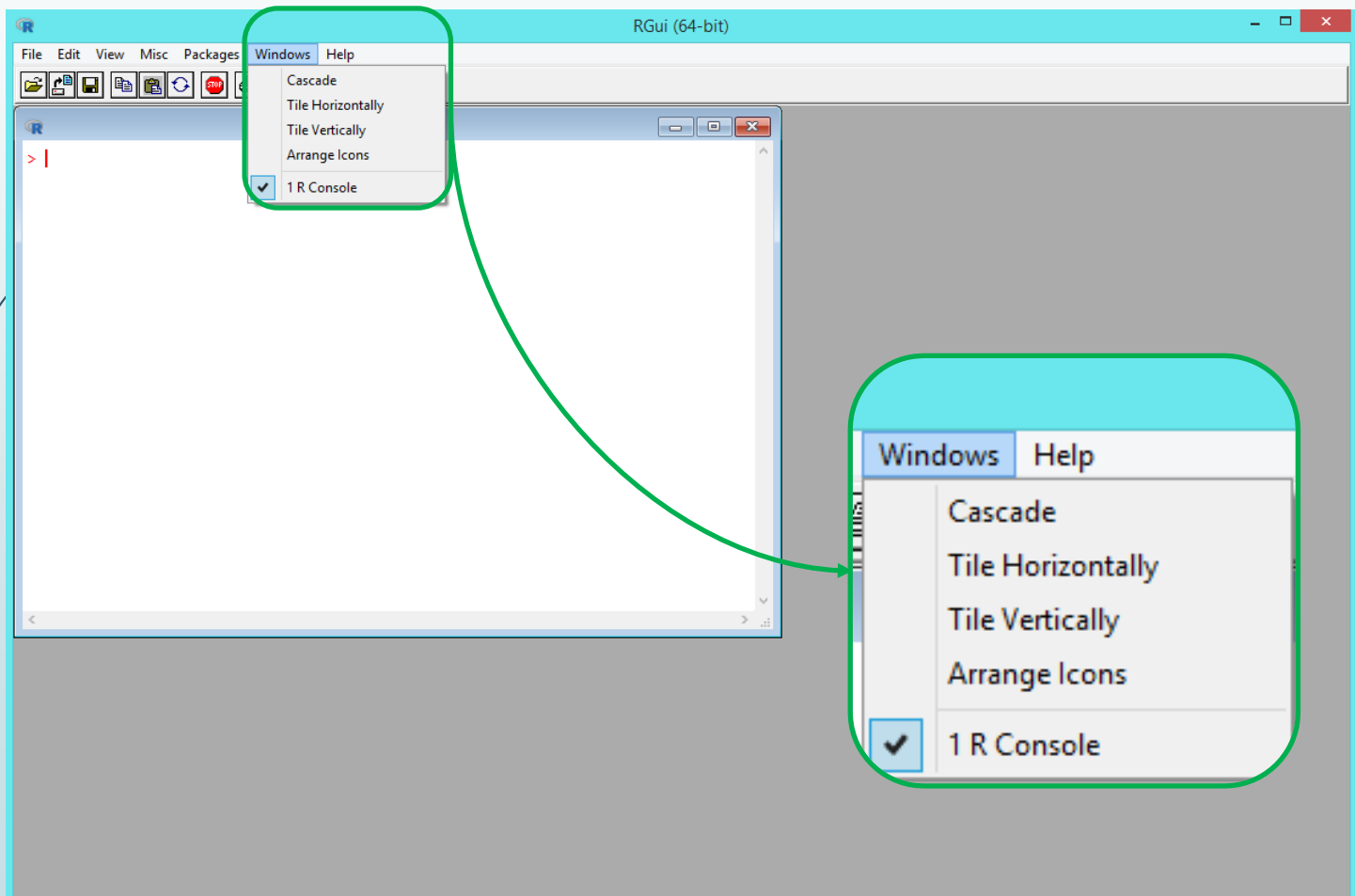
R GUI

► Menu: Packages



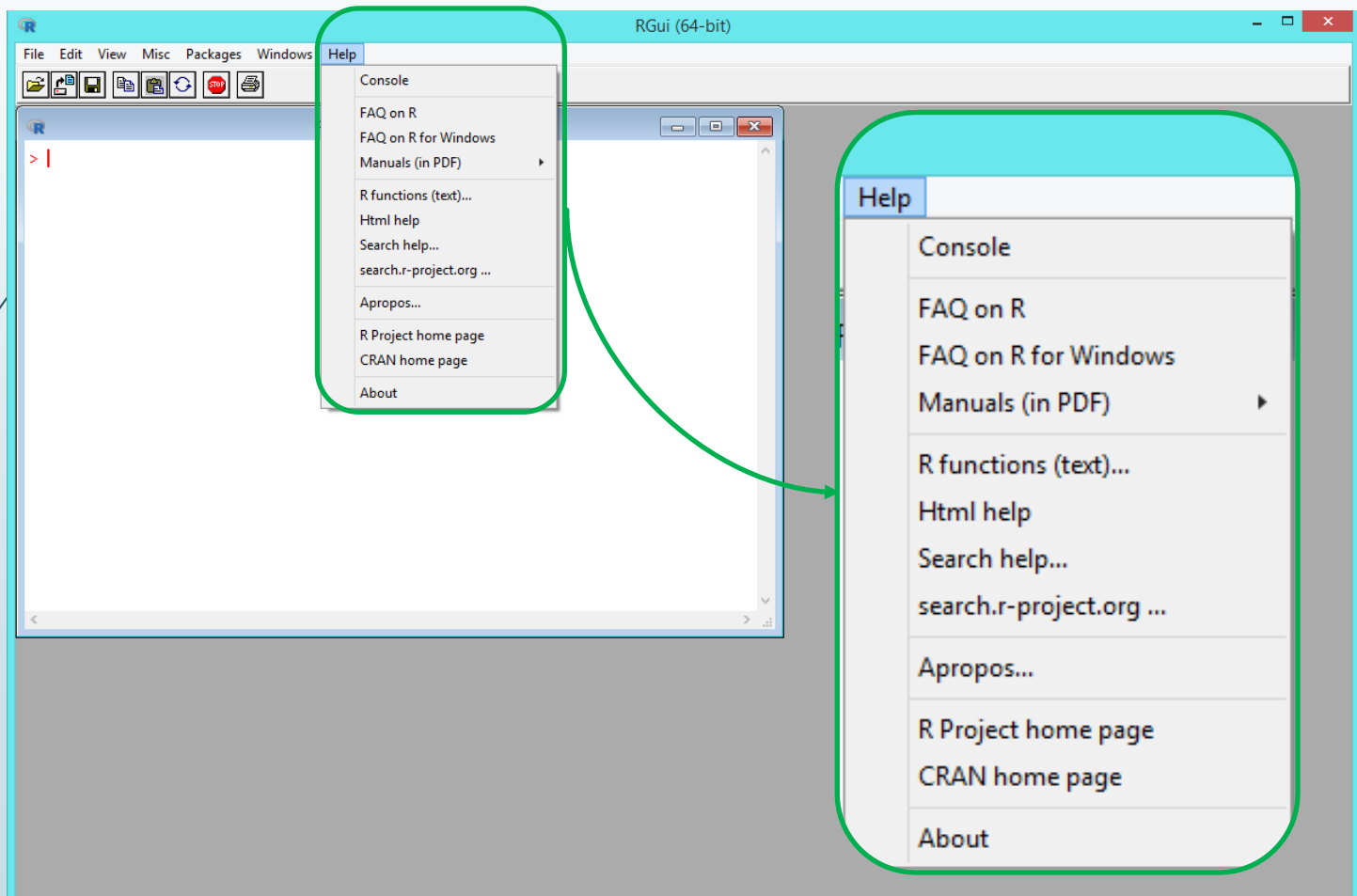
R GUI

► Menu: Windows



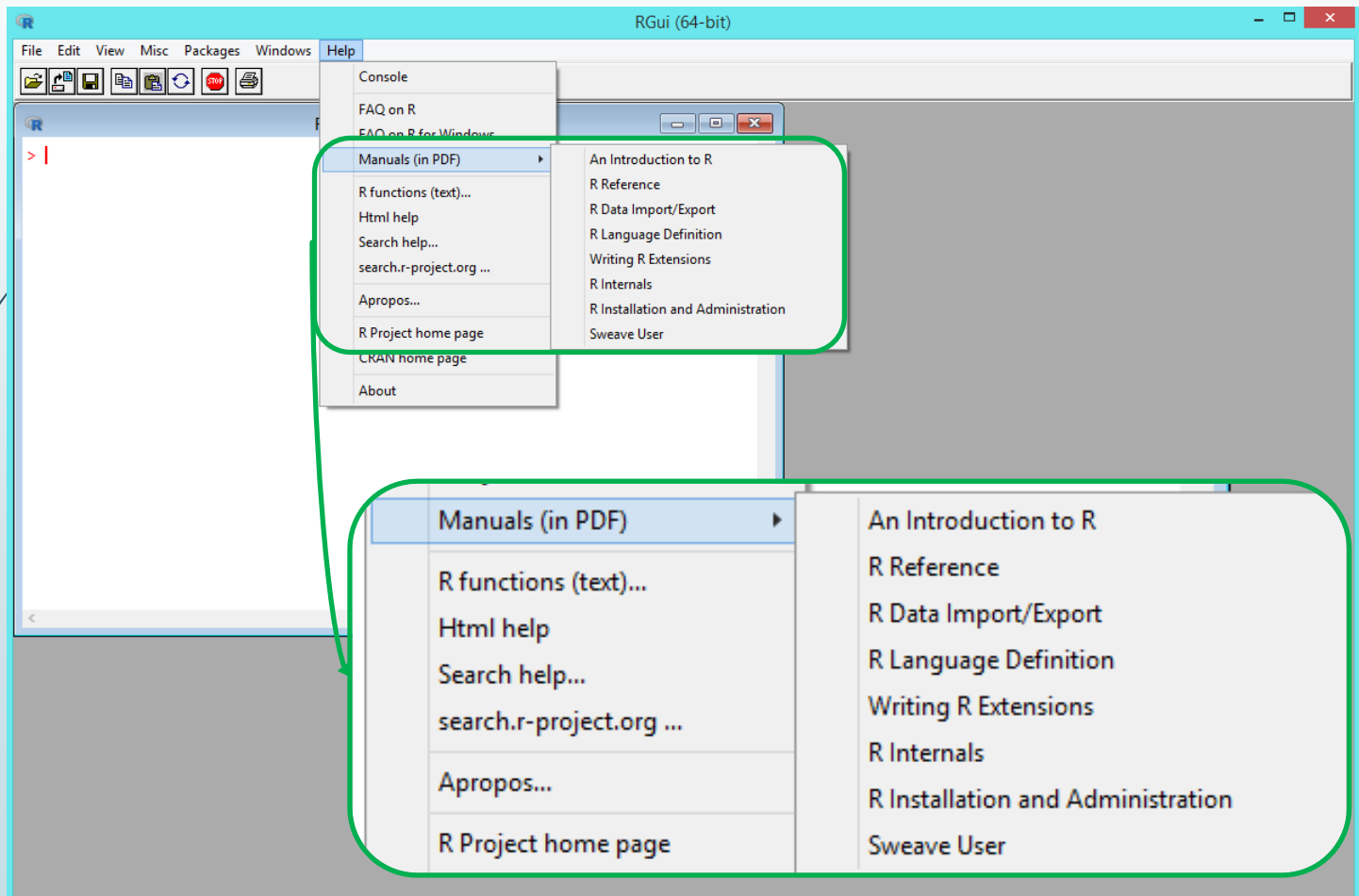
R GUI

Menu: Help



R GUI

➤ Menu: Help → Manuals (in PDF)



R IDEs

► Integrated Development Environments

- Rstudio
- JupyterLab
- Rattle
- RTVS (R Tools for Visual Studio)
- displayr
- Architect
- RKWard
- Tinn-R
- R AnalyticFlow
- ...

R IDEs

The screenshot displays the RStudio IDE interface. The main editor window shows the following R code:

```

1 ## Prepare for analyses
2 {r}
3 set.seed(1234)
4 library(ggplot2)
5 library(lattice)
6 {r}
7
8 {r}
9 x <- 1:10
10 y <- round(rnorm(10, x, 1), 2)
11 df <- data.frame(x, y)
12 df
13 {r}
14
15
16

```

The R Graphics window shows a scatter plot of 'dist' (y-axis, 0 to 125) versus 'speed' (x-axis, 5 to 20). The plot includes black data points and a blue smoothed line representing a linear model fit.

The Console window shows the following output:

```

Embedded R [R Console] Embedded R Server / RJ (Jun 20, 2017, 4:36:46 PM) · C:\Users\... \workspaceStarters2\reporting
> ggplot(cars, aes(speed, dist)) +
+   geom_point() +
+   geom_smooth(method = "lm", formula = y ~ splines::bs(x, 3), se = FALSE)
>
> lm(dist ~ speed, data = cars)

Call:
lm(formula = dist ~ speed, data = cars)

Coefficients:
(Intercept)      speed
-17.579         3.932

```

The Outline window on the left shows a project structure with folders for 'exampleProject', 'myFirstProject', 'plotting', 'refactoring', and 'reporting'. The 'reporting' folder is expanded, showing 'rmdSample.Rmd' and 'rnwSample.Rnw'. The Outline also shows a list of chunks: 'Prepare for analyses', '#1', '#2', '#3', 'Basic console output', 'basicconsole', 'Plots', 'R Code chunk features', and 'Basic functionality'.

R IDEs

The screenshot displays the Tinn-R R IDE interface. The main editor window shows a LaTeX document with the following code:

```

19 \usepackage{geometry}
20
21 \usepackage{sectsty}
22 \allsectionsfont{\sffamily\raggedright}
23
24 \geometry{verbose, a4paper,
25 tmargin=1.5cm, bmargin=1.5cm,
26 lmargin=1.5cm, rmargin=1.0cm,
27 headsep=5mm, footskip=0cm}
28
29 \usepackage{Sweave}
30 \SweaveOpts{concordance=TRUE}
31
32 \begin{document}
33
34 <<echo=F>>=
35 alunos <- read.table('dados/alunos.txt',
36                      header=TRUE,
37                      fileEncoding='utf-8'
38 #alunos <- read.table('dados/alunos_ficti
39 #
40 #                      header=TRUE,
41 #                      fileEncoding='utf-8'
42 aluno <- alunos[6,
43                 1]
44
45 # Matricula
46 #1 201620294      CLEONE JÚNIO LELIS SA
47 #2 201420562      JOSÉ TALMON MELO JÚ
48 #3 201320310      LUCAS DE SOUZA GA
49 #4 201520229      MARCEL SILVA HENR

```

The R console on the right shows the following output:

```

> (x <- rnorm(16))
[1] -0.1847793  1.1031598 -0.1817066
[4]  0.5238915  1.5089383  1.2939852
[7]  0.6151822 -0.4550306 -1.1959774
[10] -0.1192176  2.2176692  0.6228209
[13] -1.0513555 -0.6785117 -0.6853487
[16] -0.6907645

> mean(x)
[1] 0.1651847

> var(x)
[1] 0.994795

> sd
function (x, na.rm = FALSE)
sqrt(var(if (is.vector(x) || is.factor(x)) x
          na.rm = na.rm))
<bytecode: 0x00000001b520830>
<environment: namespace:stats>

> sd(1e3, 10, )

```

The console also shows a small dialog box with the following text:

```

x=
na.rm=FALSE

```

The status bar at the bottom indicates the file is in UTF-8 encoding, is 11.48 KB in size, and is being edited in the C: drive.

The screenshot illustrates the RStudio IDE environment. The main window displays a rendered R Markdown document titled "Managing R Packages with roxyPackage" by m.ik michalke, dated 2019-11-14. The document content includes an abstract and a main body of text discussing R packages and the roxyPackage tool.

The left pane shows the file explorer and environment. The environment pane displays the following data frame:

Name	Label	Type	Class
sleep		data.frame	
ID		Factor	factor
extra		Numeric	numeric
group		Factor	factor
test.data			matrix

The bottom pane shows the following R code:

```

local({
  ## Prepare
  require(psych)
  ## Print result
  rk.header("Parallel analysis (Horn) results")
  rk.graph.on()
  try({
    parallel.data <- fa.parallel(
      test.data,
      n.obs=245)
  })
  rk.graph.off()
})

```

The right pane shows the "Configure Packages" dialog and a "Parallel Analysis Scree Plots" plot. The plot displays eigenvalues of principal components and factor analysis, with the x-axis labeled "Factor/Component Number" (0 to 20) and the y-axis labeled "eigenvalues of principal components and factor analysis" (0 to 8). The plot includes four data series: PC Actual Data (blue line with 'x' markers), PC Simulated Data (red dashed line with 'x' markers), FA Actual Data (blue line with 'x' markers), and FA Simulated Data (red dashed line with 'x' markers).

R IDEs

In Depth: Linear Regression

Just as naive Bayes (discussed earlier in [In Depth: Naive Bayes Classification](#)) is a good starting point for classification tasks, linear regression models are a good starting point for regression tasks. Such models are popular because they can be fit very quickly, and are very interpretable. You are probably familiar with the simplest form of a linear regression model (i.e., fitting a straight line to data) but such models can be extended to model more complicated data behavior.

In this section we will start with a quick intuitive walk-through of the mathematics behind this well-known problem, before seeing how before moving on to see how linear models can be generalized to account for more complicated patterns in data.

We begin with

```

[1]: %matplotlib inline
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

```

Simple

We will start with a simple example where y is a function of x .

```

[2]: rng = np.random.RandomState(1)
x = 10 * rng.rand(100)
y = 2 * x + 1 + rng.rand(100)
plt.scatter(x, y)

```

Consider the following

```

[3]: from sklearn.linear_model import LinearRegression
lin_regr = LinearRegression()
lin_regr.fit(x.reshape(-1, 1), y)

```

We can use

```

lin_regr.predict(x.reshape(-1, 1))

```

Launcher

- Python 3
- C++11
- C++14
- C++17
- Julia 1.1.0
- phylogenetics (Python 3.7)
- R

Console

```

Python 3
C++11
C++14
C++17

```

Seattle Weather: 2012-2015

Maximum Daily Temperature (C)

Date

weather

Number of Records

Julia

```

[10]: using RDatasets, Gadfly
plot(datasets("datasets", "iris"), x=:Sepal.Length, y=:Petal.Length)

```

python notebook

```

[1]: %matplotlib inline
from ipywidgets import interactive, fixed

We explore the Lorenz system of differential equations:

x-dot = sigma*(y - x)
y-dot = px - y - xz
z-dot = -beta*z + xy

Let's change (sigma, beta, p) with ipywidgets and examine the trajectories.

[2]: from lorenz import solve_lorenz
w = interactive(solve_lorenz, sigma=(0.0, 50.0), beta=10.0, description='sigma', max=50.0), FloatSlider(value=2.6666666666666666, min=0.0, max=10.0)

```

R

```

[3]: ggplot(data=iris, aes(x=Sepal.Length, y=Petal.Length)) + geom_point()

```

head(iris)

Sepal.Length	Sepal.Width	Petal.Length
5.1	3.5	1.4
4.9	3.0	1.4

R IDEs

The screenshot displays an R IDE interface with the following components:

- Menu Bar:** File, Edit, View, Run, Project, Preferences, Help.
- Toolbar:** Input, Edit, Chart, Summary, Model, Output, Script, Custom, Projects...
- Main Canvas:** A scatter plot of Sepal.Width (Y-axis, 2.0 to 4.5) versus Sepal.Length (X-axis, 5 to 8). Data points are colored by species: setosa (blue), versicolor (pink), and virginica (green). A legend on the right identifies the colors.
- R Console:**

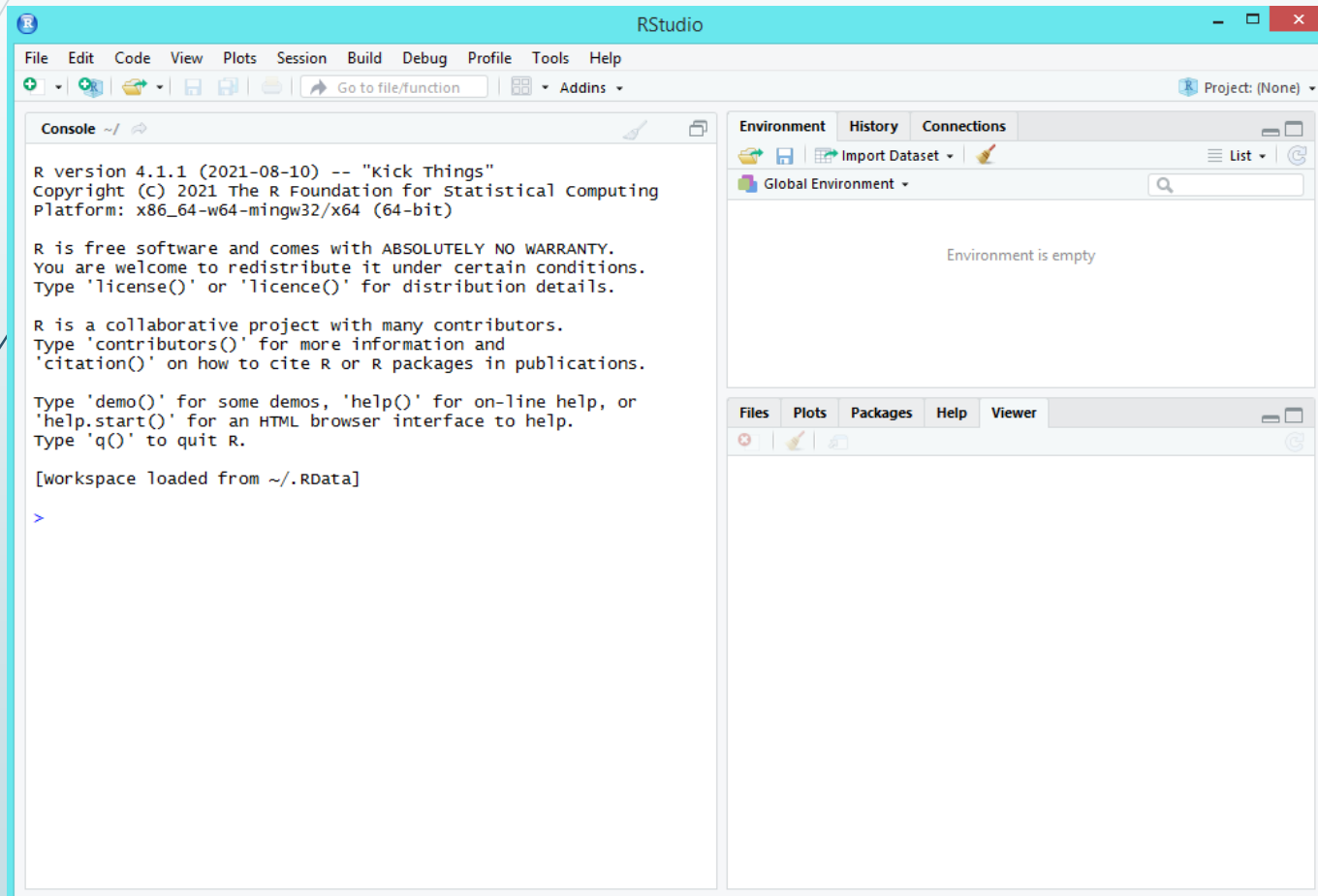
```
> data(iris)
> print(lattice::xyplot(x = Sepal.Width ~ Sepal.Length, data = iris,
auto.key = list(space = "right", groups = Species))
>
```
- AnalysisFlow Panel:** A workflow diagram showing the process: Load Sample Data → Sampling → Build Predictiv... → Predict → Cross Tabulation. A branch from Build Predictiv... leads to Draw Tree Model. A branch from Load Sample Data leads to Histogram and X-Y Plot.
- Configuration Panel:**

Main	Title	Axis	Advanced
Data	iris		
X	Sepal.Length		
Y	Sepal.Width		<input checked="" type="checkbox"/>
Conditioning			
Grouping	Species		
Legends	Right		

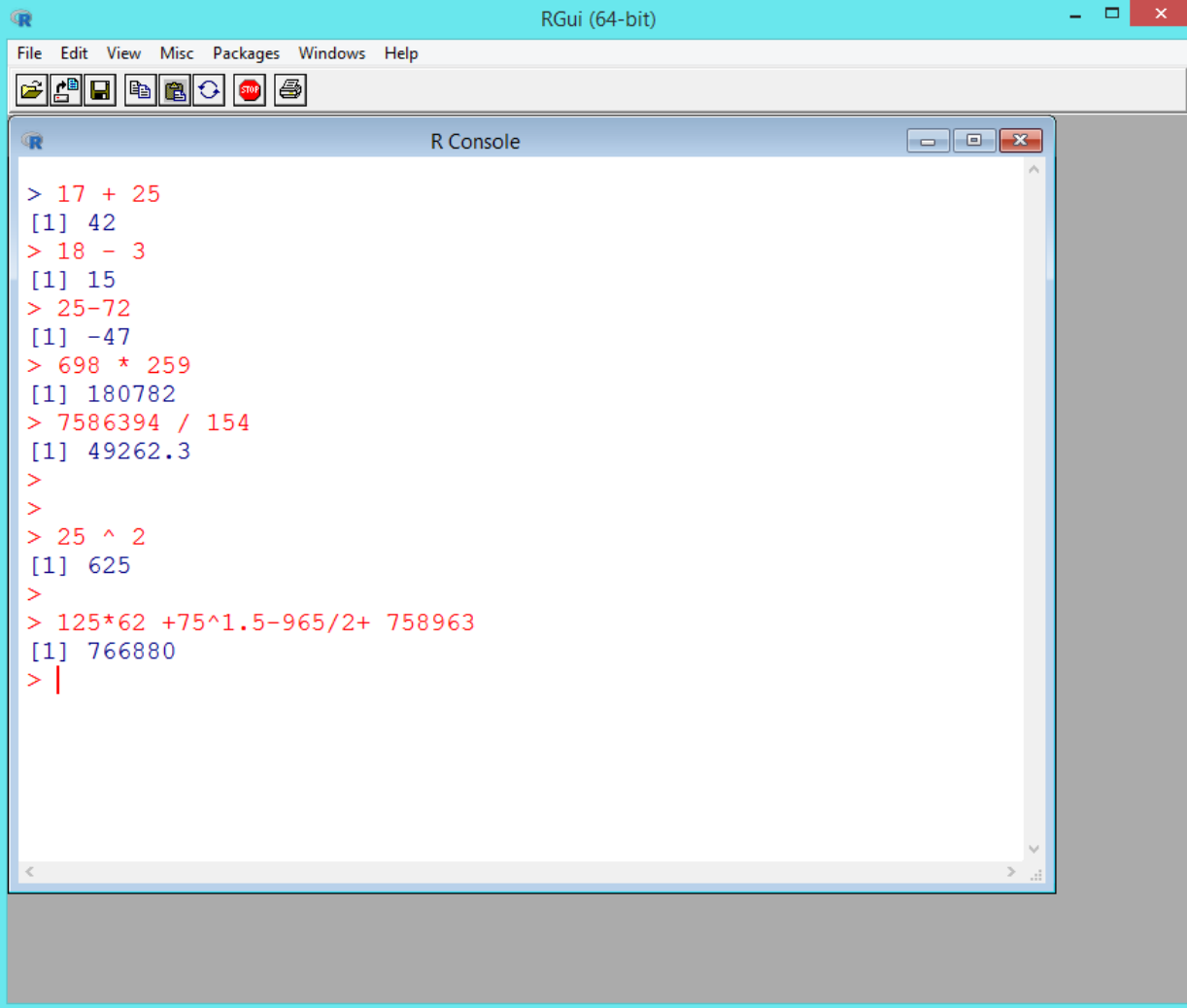
RStudio



► Popular **R** IDE



R Console as a Calculator

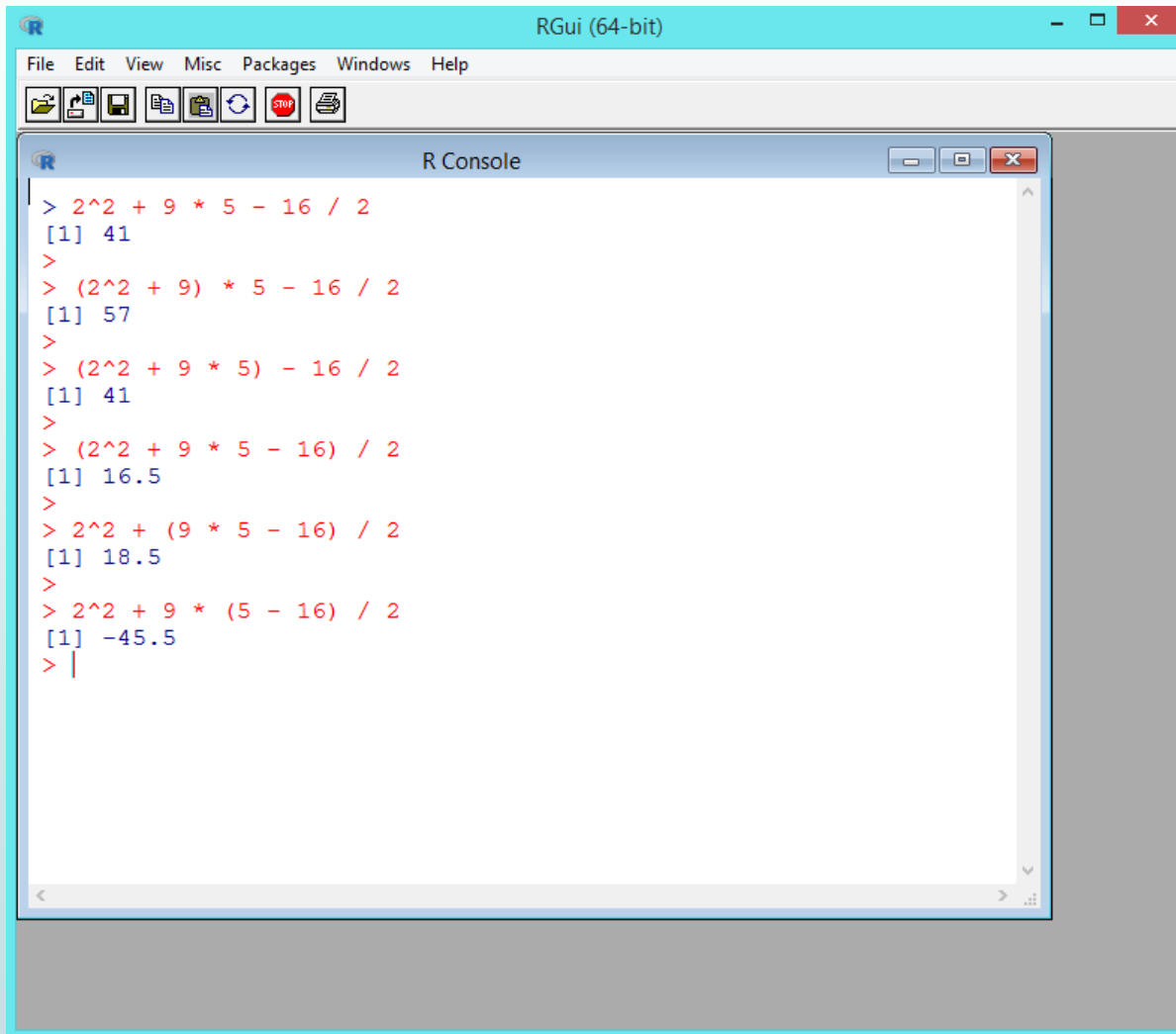


```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> 17 + 25
[1] 42
> 18 - 3
[1] 15
> 25-72
[1] -47
> 698 * 259
[1] 180782
> 7586394 / 154
[1] 49262.3
>
>
> 25 ^ 2
[1] 625
>
> 125*62 +75^1.5-965/2+ 758963
[1] 766880
> |
```

Order of Precedence

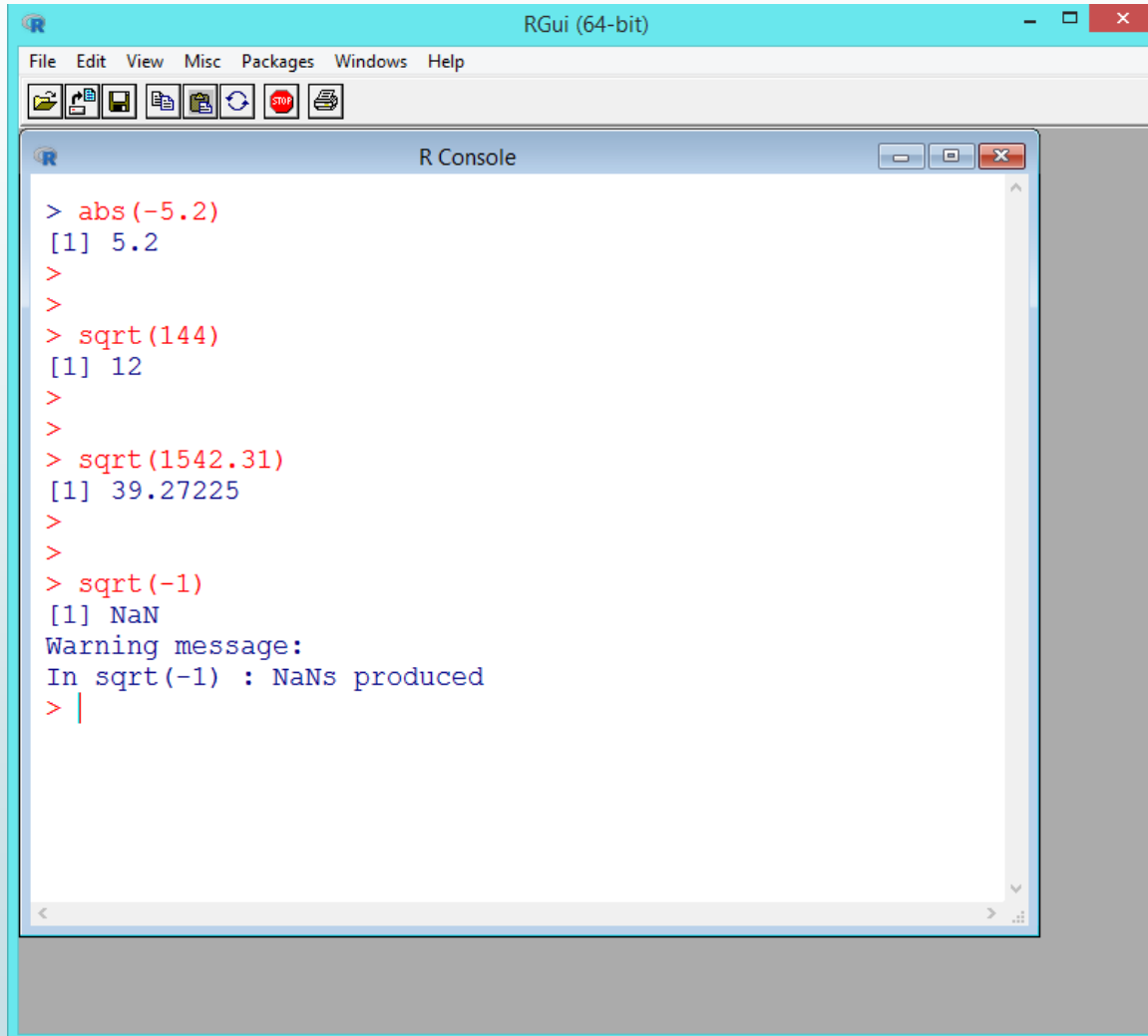
Order	Symbol
1	Parentheses. For nested parentheses, the innermost are executed first.
2	Exponentiation.
3	Multiplication, division (equal precedence).
4	Addition and subtraction.

Order of Precedence



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> 2^2 + 9 * 5 - 16 / 2
[1] 41
>
> (2^2 + 9) * 5 - 16 / 2
[1] 57
>
> (2^2 + 9 * 5) - 16 / 2
[1] 41
>
> (2^2 + 9 * 5 - 16) / 2
[1] 16.5
>
> 2^2 + (9 * 5 - 16) / 2
[1] 18.5
>
> 2^2 + 9 * (5 - 16) / 2
[1] -45.5
> |
```

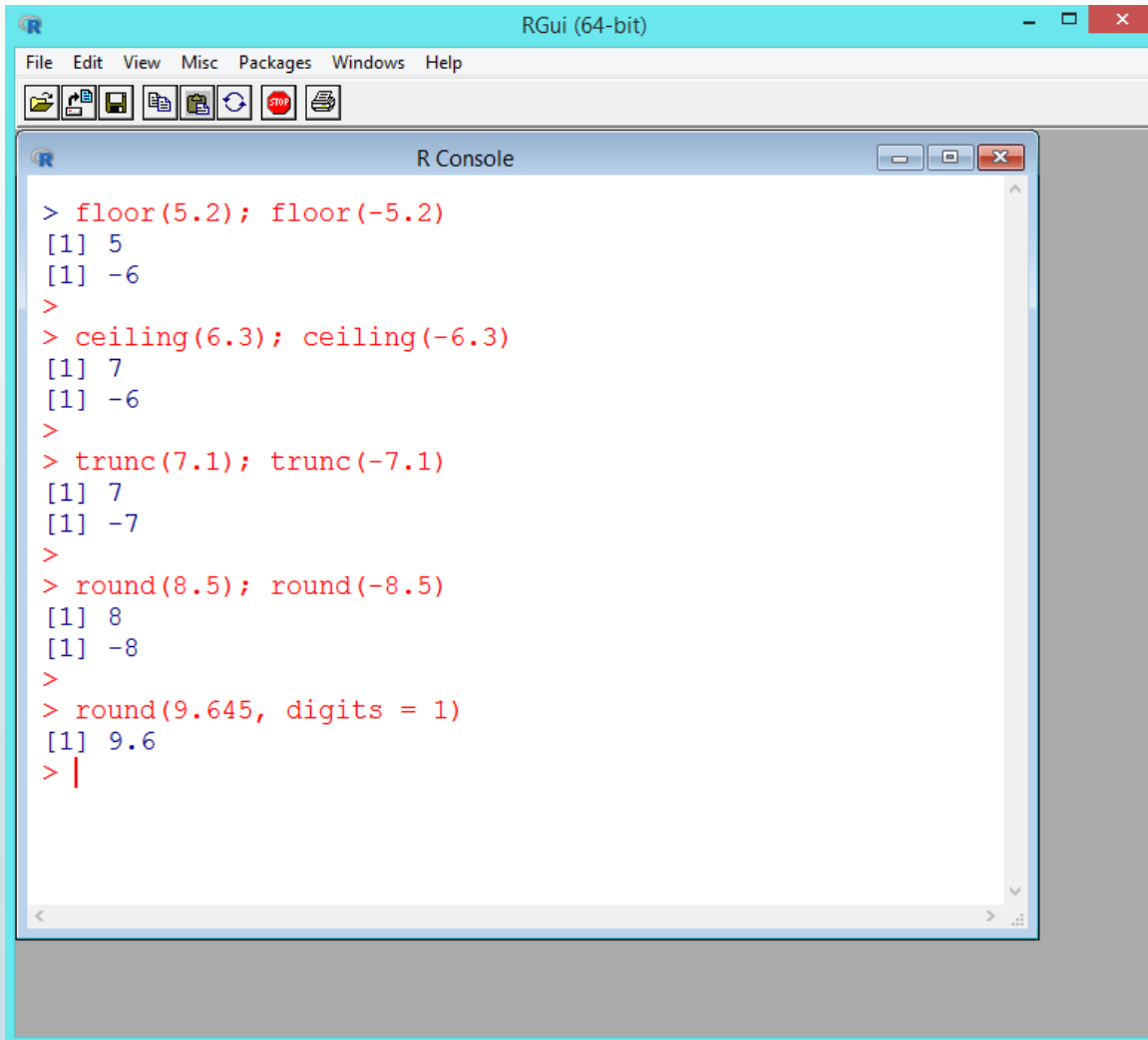
Some Base Functions



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> abs(-5.2)
[1] 5.2
>
>
> sqrt(144)
[1] 12
>
>
> sqrt(1542.31)
[1] 39.27225
>
>
> sqrt(-1)
[1] NaN
Warning message:
In sqrt(-1) : NaNs produced
> |
```

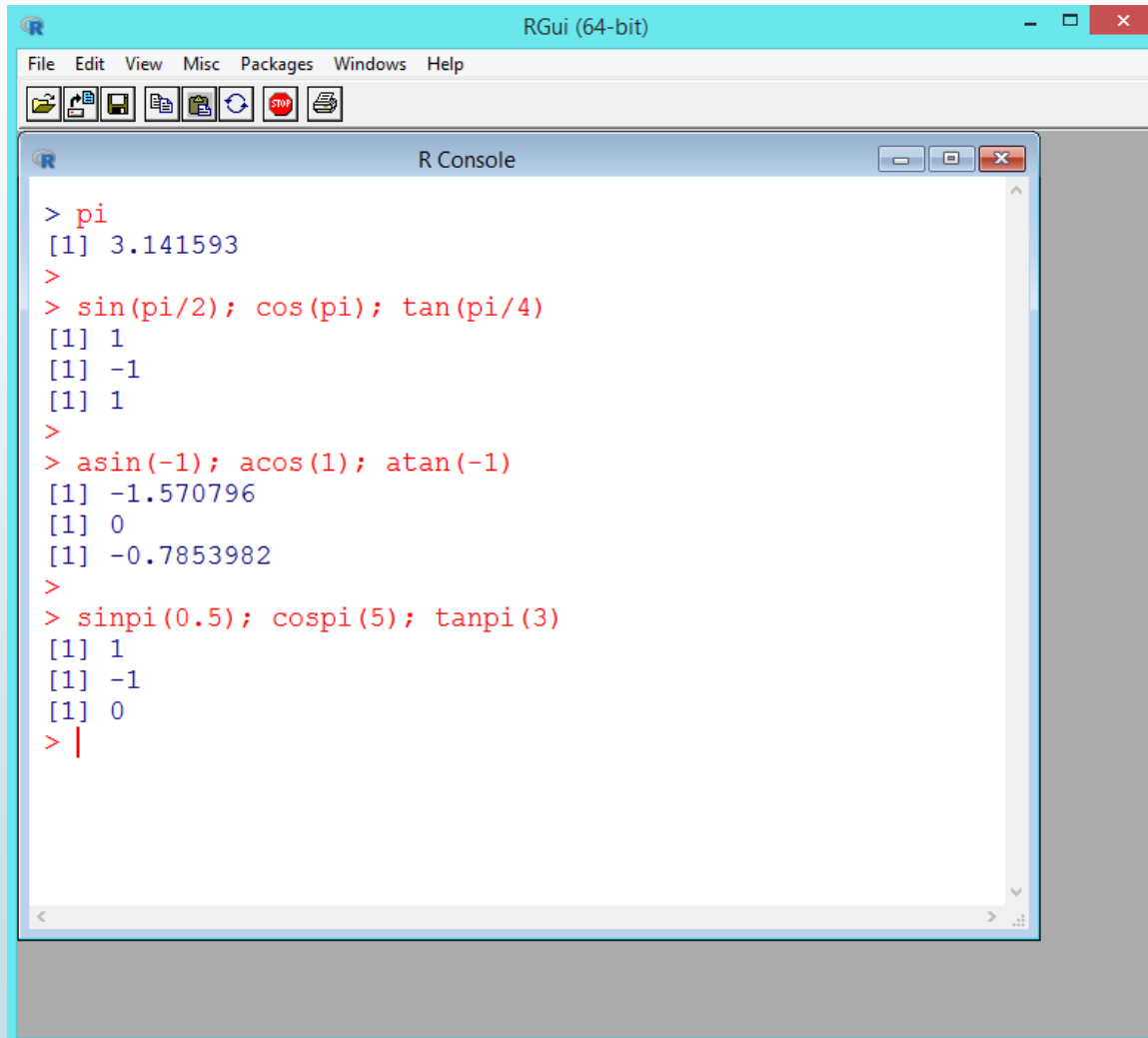
Rounding Functions



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> floor(5.2); floor(-5.2)
[1] 5
[1] -6
>
> ceiling(6.3); ceiling(-6.3)
[1] 7
[1] -6
>
> trunc(7.1); trunc(-7.1)
[1] 7
[1] -7
>
> round(8.5); round(-8.5)
[1] 8
[1] -8
>
> round(9.645, digits = 1)
[1] 9.6
> |
```

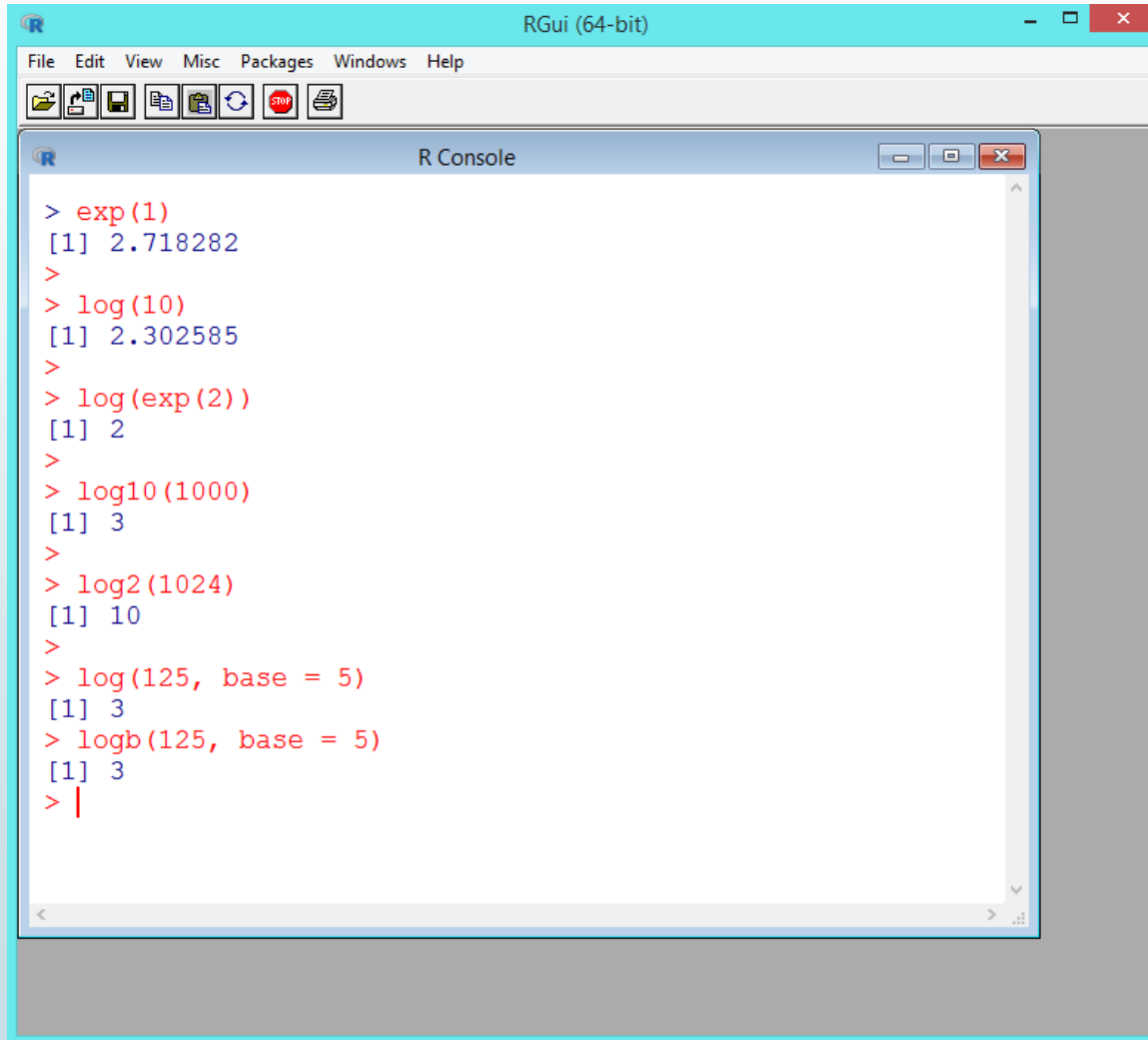
Trigonometric Functions



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> pi
[1] 3.141593
>
> sin(pi/2); cos(pi); tan(pi/4)
[1] 1
[1] -1
[1] 1
>
> asin(-1); acos(1); atan(-1)
[1] -1.570796
[1] 0
[1] -0.7853982
>
> sinpi(0.5); cospi(5); tanpi(3)
[1] 1
[1] -1
[1] 0
> |
```

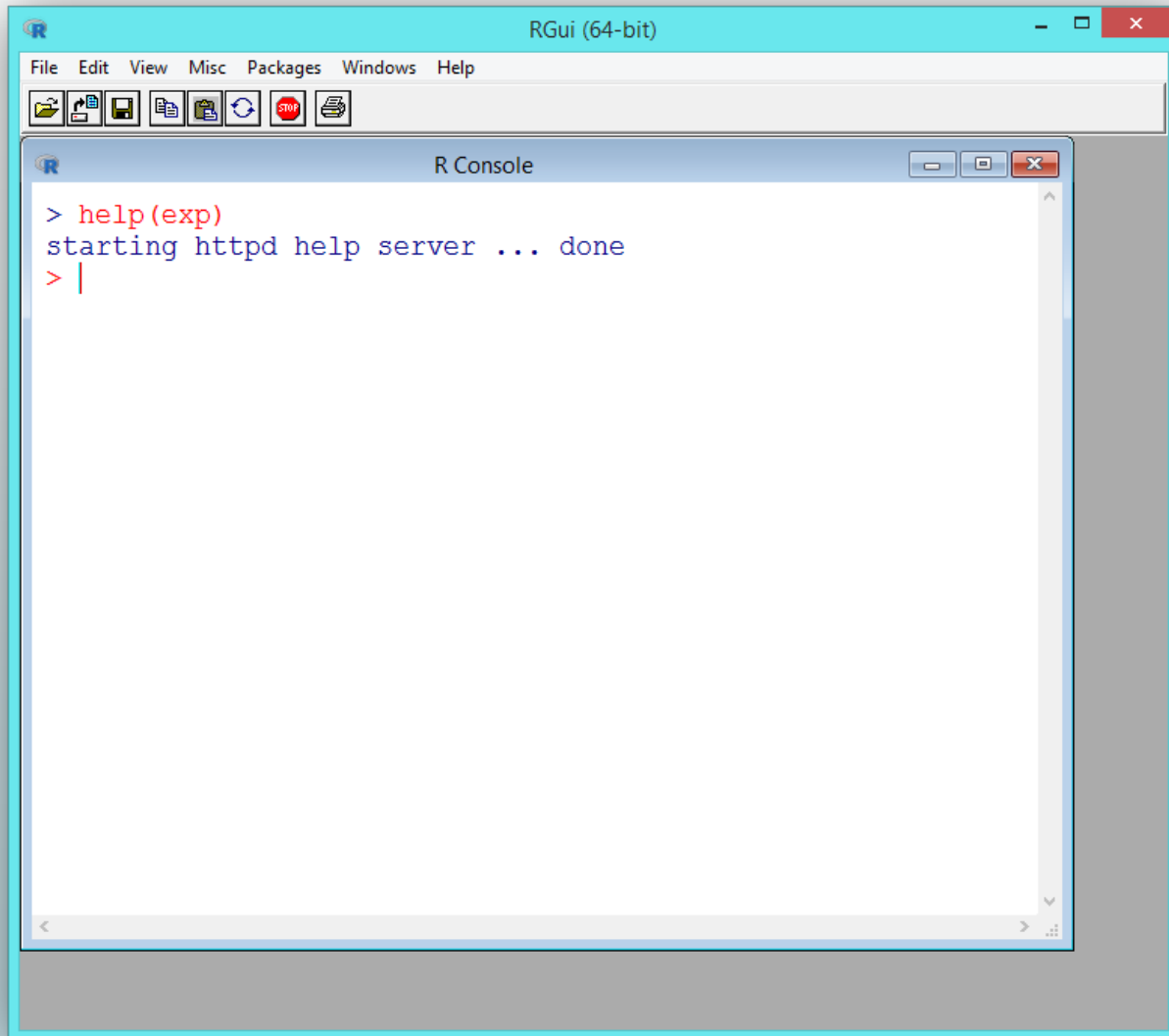
Exponential and Logarithmic Functions



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

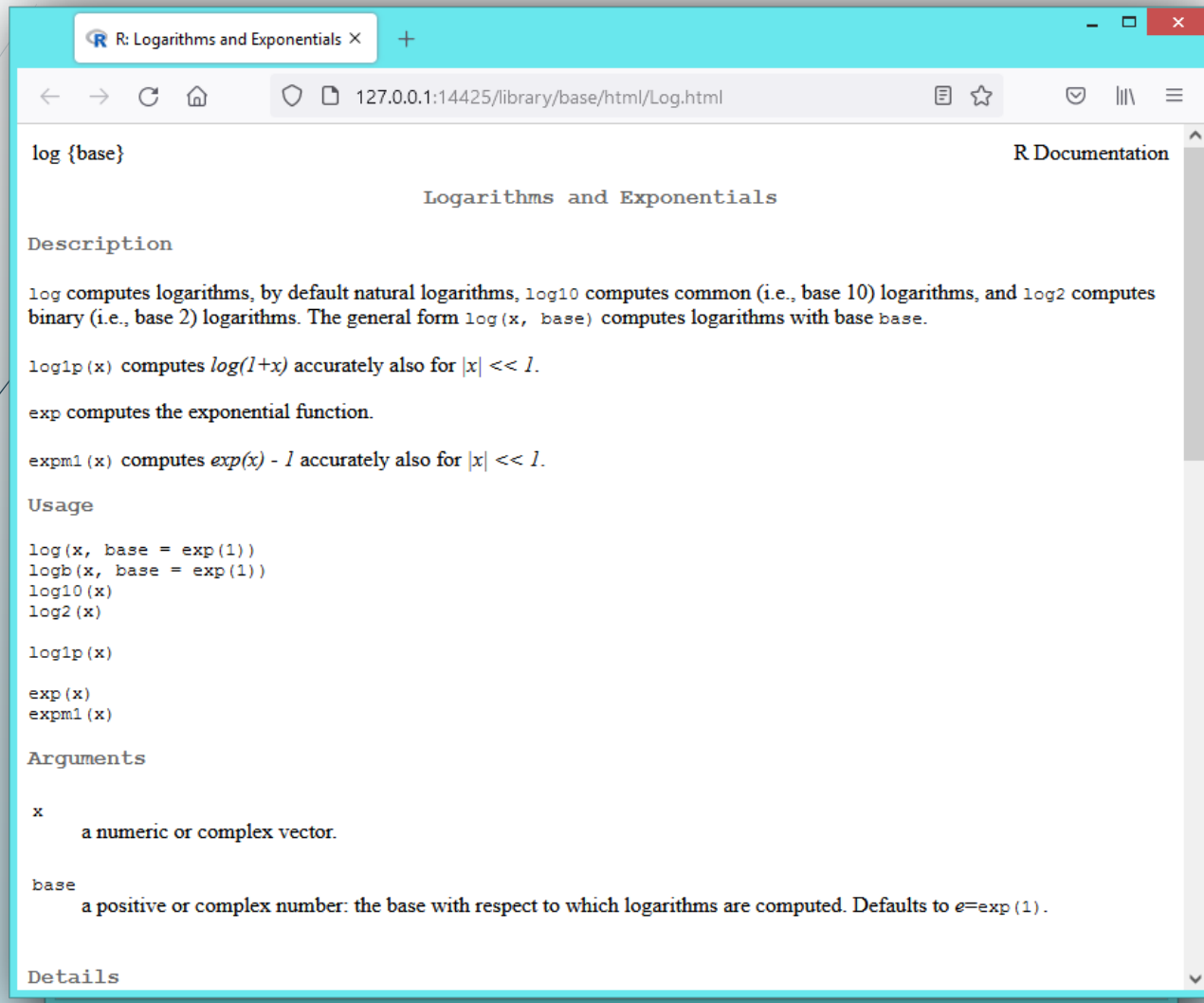
R Console
> exp(1)
[1] 2.718282
>
> log(10)
[1] 2.302585
>
> log(exp(2))
[1] 2
>
> log10(1000)
[1] 3
>
> log2(1024)
[1] 10
>
> log(125, base = 5)
[1] 3
> logb(125, base = 5)
[1] 3
> |
```


help ()



```
> help(exp)
starting httpd help server ... done
> |
```

help()



The screenshot shows a web browser window displaying the R documentation for logarithmic and exponential functions. The browser's address bar shows the URL `127.0.0.1:14425/library/base/html/Log.html`. The page title is "R Documentation" and the main heading is "Logarithms and Exponentials".

Description

`log` computes logarithms, by default natural logarithms, `log10` computes common (i.e., base 10) logarithms, and `log2` computes binary (i.e., base 2) logarithms. The general form `log(x, base)` computes logarithms with base `base`.

`log1p(x)` computes $\log(1+x)$ accurately also for $|x| \ll 1$.

`exp` computes the exponential function.

`expm1(x)` computes $\exp(x) - 1$ accurately also for $|x| \ll 1$.

Usage

```
log(x, base = exp(1))
logb(x, base = exp(1))
log10(x)
log2(x)

log1p(x)

exp(x)
expm1(x)
```

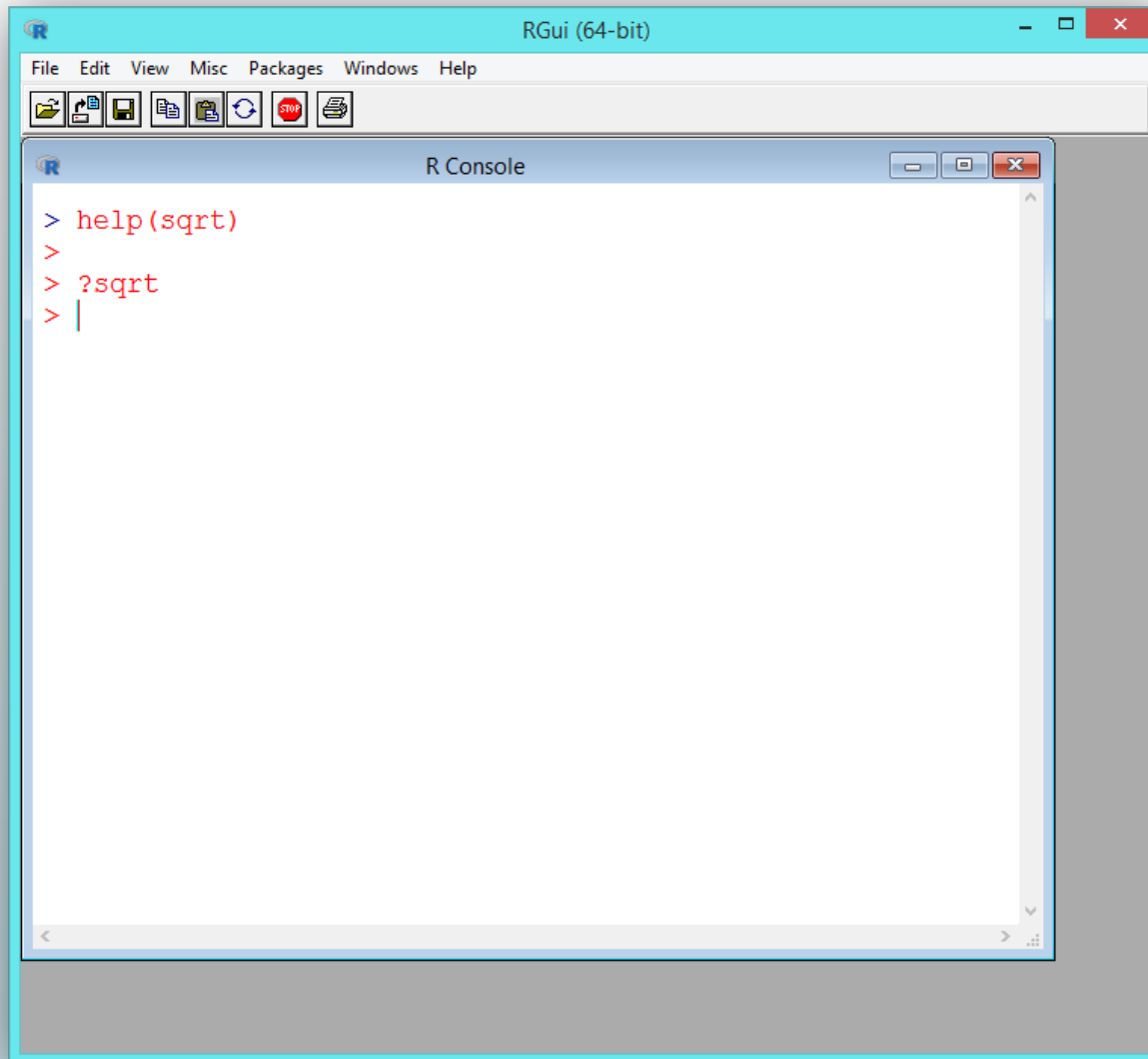
Arguments

`x`
a numeric or complex vector.

`base`
a positive or complex number: the base with respect to which logarithms are computed. Defaults to $e = \exp(1)$.

Details

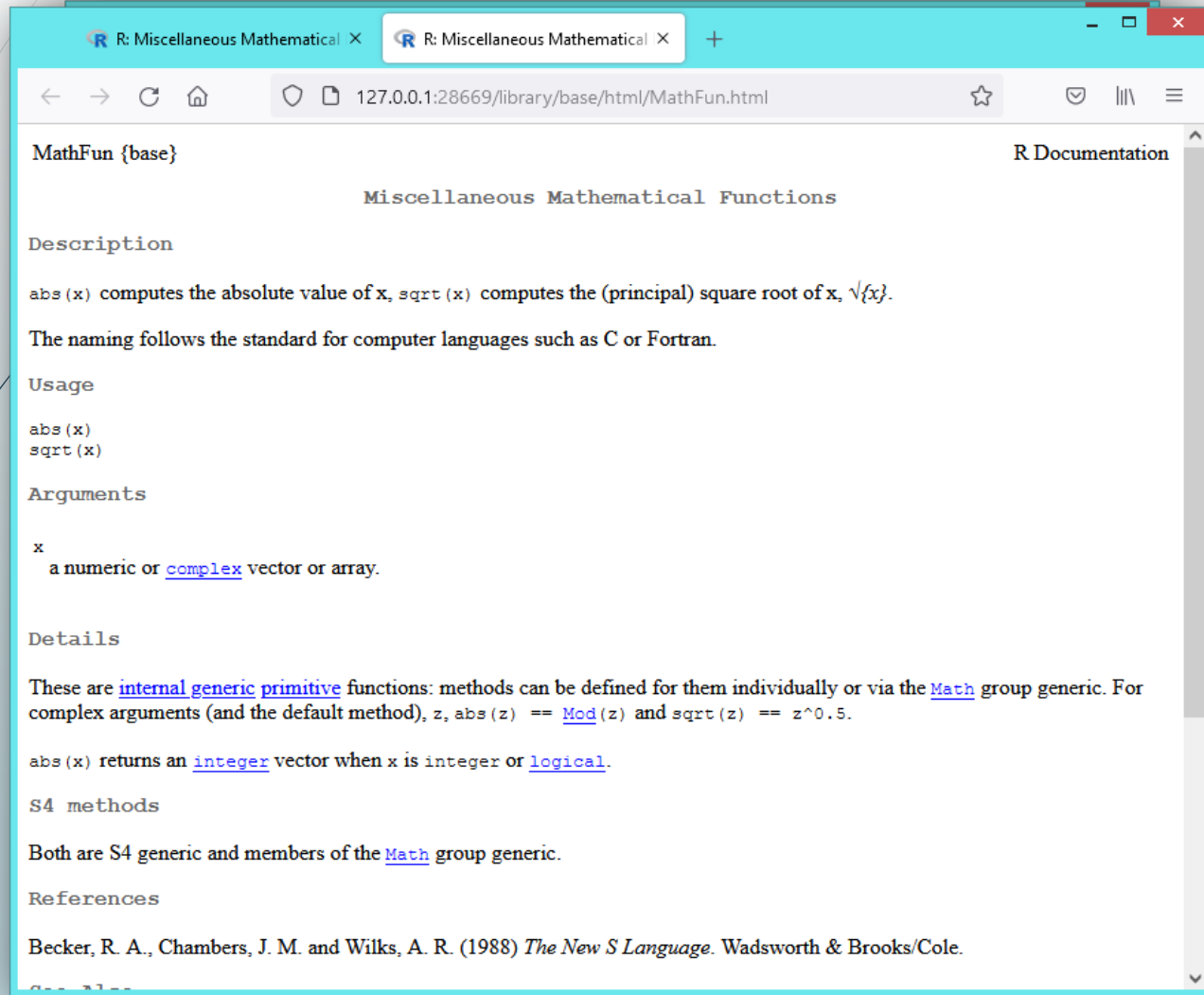
help () and ?



The image shows a screenshot of the RGui (64-bit) window. The window has a menu bar with 'File', 'Edit', 'View', 'Misc', 'Packages', 'Windows', and 'Help'. Below the menu bar is a toolbar with icons for file operations. The main area is the R Console, which contains the following text:

```
> help(sqrt)
>
> ?sqrt
> |
```

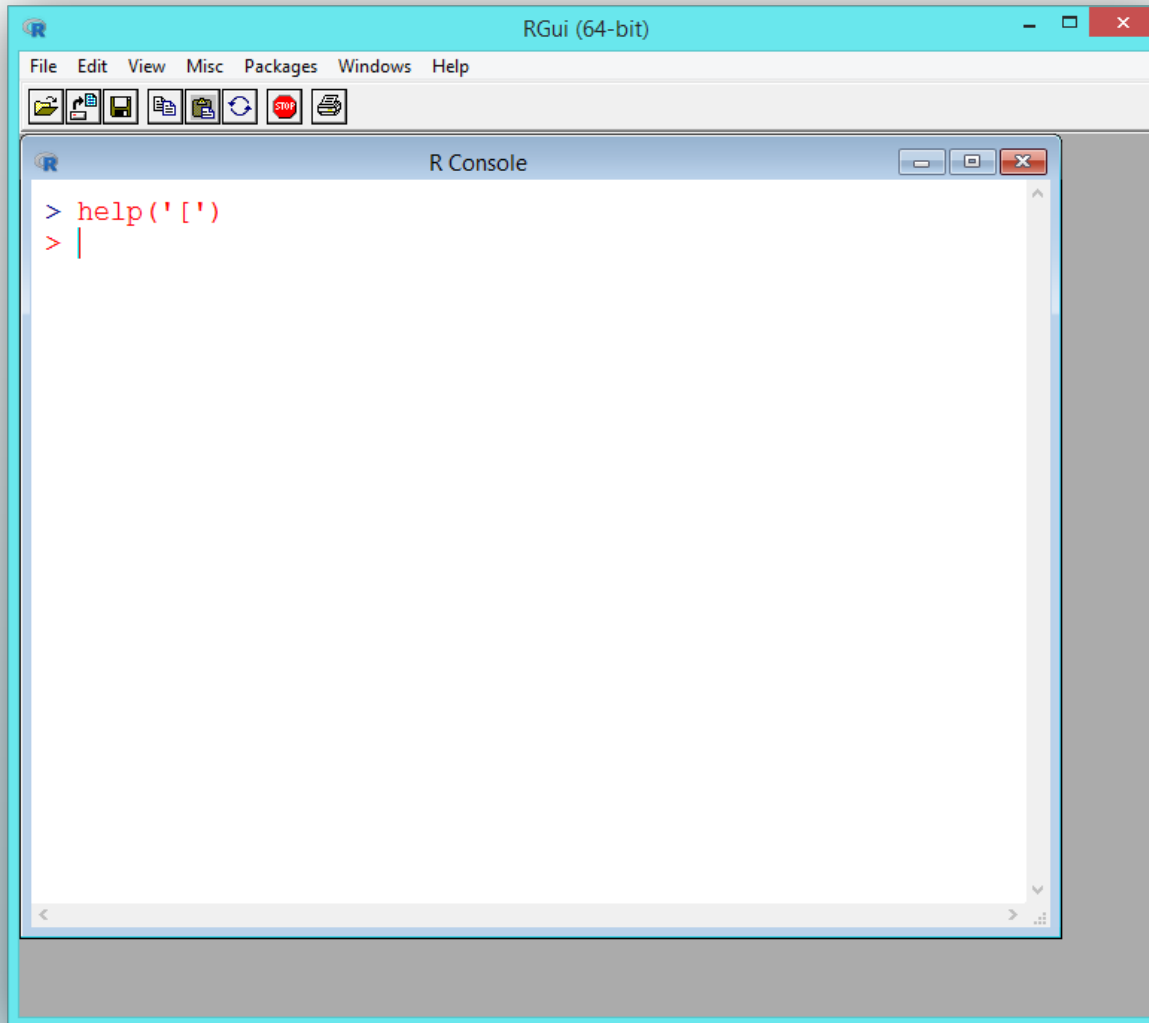
help() and ?



The screenshot shows a web browser window displaying the R documentation for the `MathFun` package. The browser tabs are labeled "R: Miscellaneous Mathematical". The address bar shows the URL `127.0.0.1:28669/library/base/html/MathFun.html`. The page content includes the following sections:

- MathFun {base}** (R Documentation)
- Miscellaneous Mathematical Functions**
- Description**
 - `abs(x)` computes the absolute value of `x`, `sqrt(x)` computes the (principal) square root of `x`, \sqrt{x} .
 - The naming follows the standard for computer languages such as C or Fortran.
- Usage**
 - `abs(x)`
 - `sqrt(x)`
- Arguments**
 - `x`
a numeric or [complex](#) vector or array.
- Details**
 - These are [internal generic primitive](#) functions: methods can be defined for them individually or via the [Math](#) group generic. For complex arguments (and the default method), `z`, `abs(z) == Mod(z)` and `sqrt(z) == z^0.5`.
 - `abs(x)` returns an [integer](#) vector when `x` is integer or [logical](#).
- S4 methods**
 - Both are S4 generic and members of the [Math](#) group generic.
- References**
 - Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

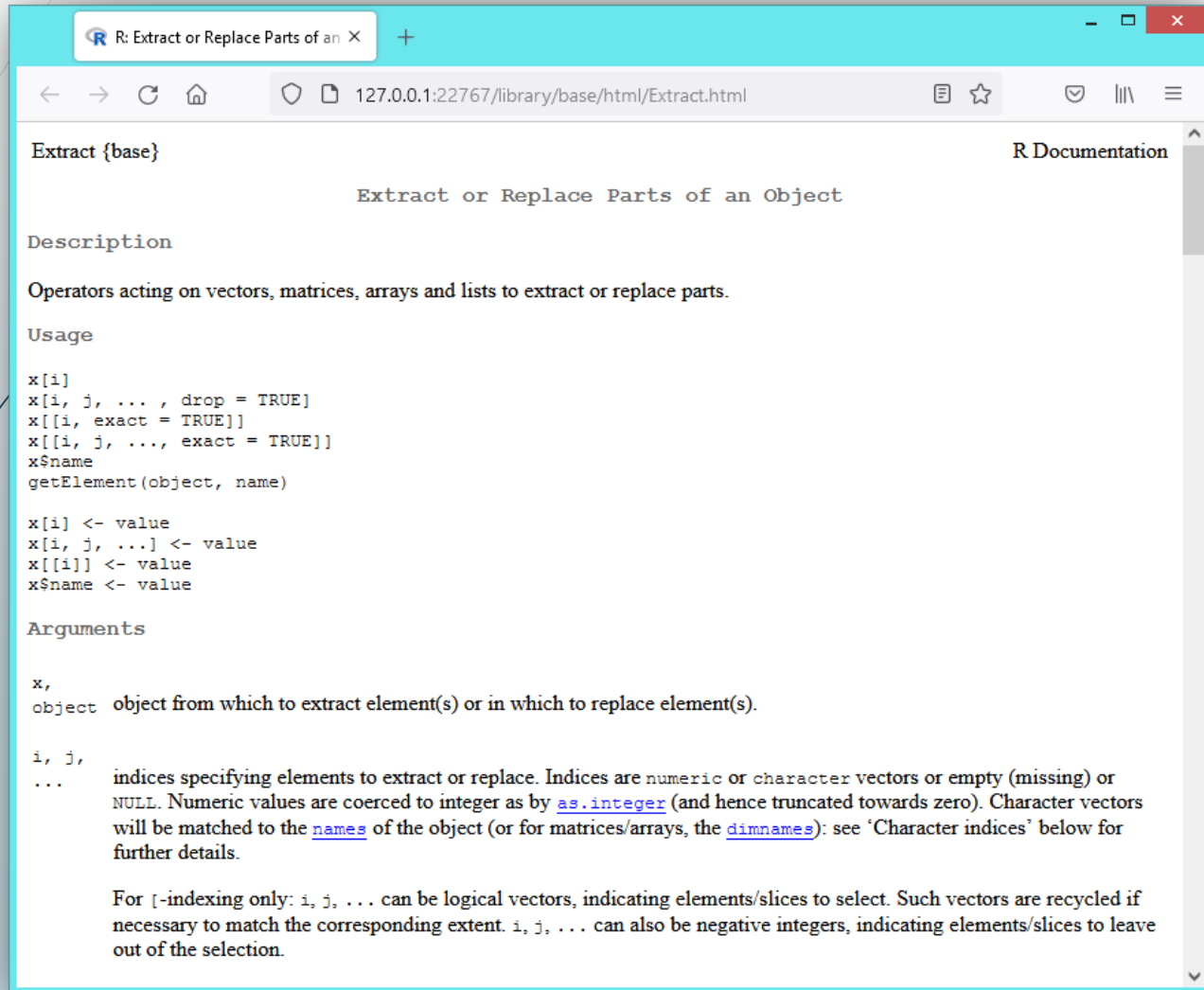
help() and ?



The image shows a screenshot of the RGui (64-bit) application window. The window has a menu bar with 'File', 'Edit', 'View', 'Misc', 'Packages', 'Windows', and 'Help'. Below the menu bar is a toolbar with icons for file operations and execution. The main area is the R Console, which displays the following text:

```
> help('[')  
> |
```

help() and ?



The screenshot shows a web browser window displaying the R documentation for the `Extract` function. The browser's address bar shows the URL `127.0.0.1:22767/library/base/html/Extract.html`. The page title is "Extract {base}" and the subtitle is "Extract or Replace Parts of an Object". The content includes a description, usage examples, and arguments.

Extract {base} R Documentation

Extract or Replace Parts of an Object

Description

Operators acting on vectors, matrices, arrays and lists to extract or replace parts.

Usage

```
x[i]
x[i, j, ... , drop = TRUE]
x[[i, exact = TRUE]]
x[[i, j, ..., exact = TRUE]]
x$name
getElement(object, name)

x[i] <- value
x[i, j, ...] <- value
x[[i]] <- value
x$name <- value
```

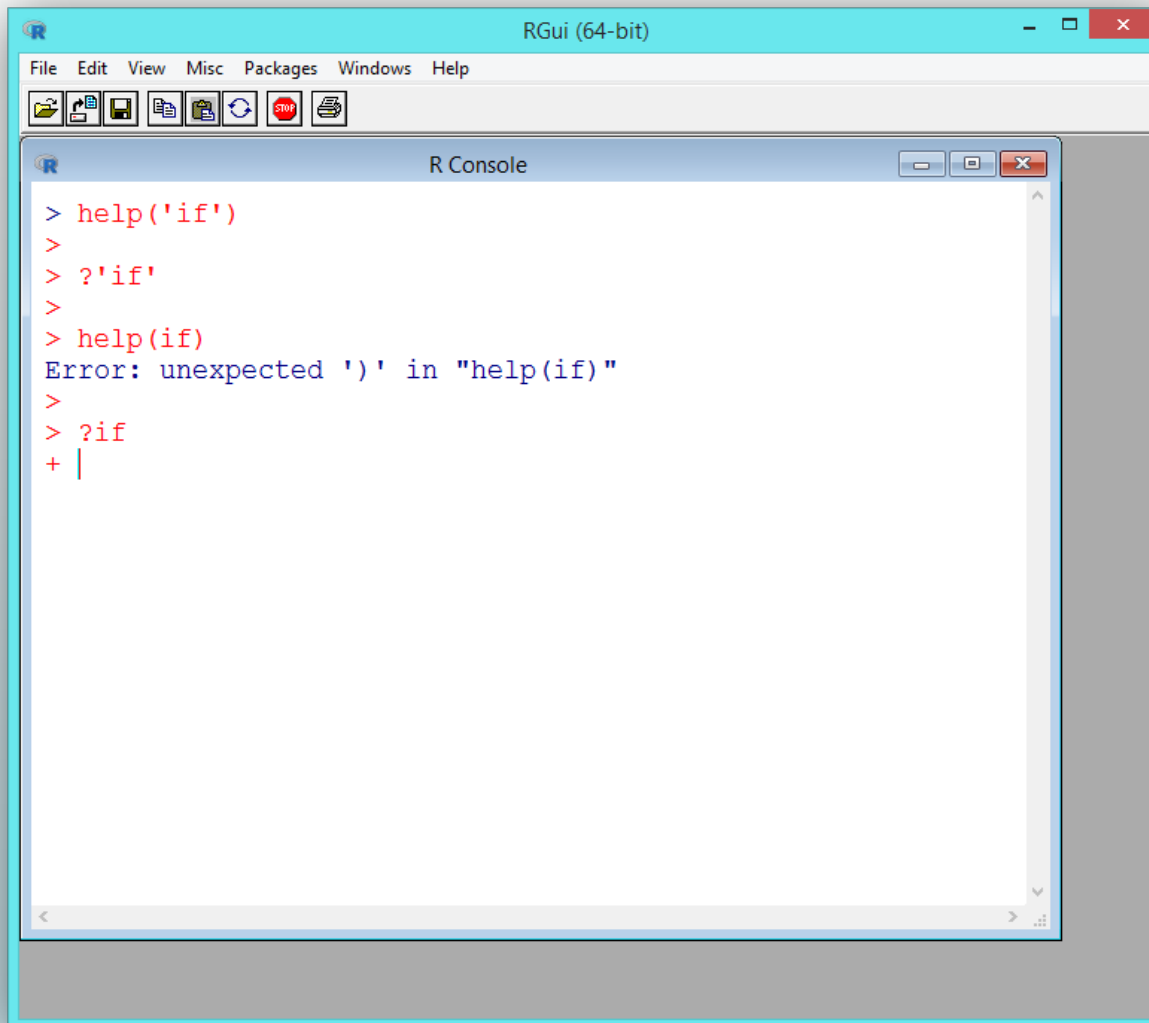
Arguments

`x`,
`object` object from which to extract element(s) or in which to replace element(s).

`i`, `j`,
`...` indices specifying elements to extract or replace. Indices are numeric or character vectors or empty (missing) or NULL. Numeric values are coerced to integer as by [as.integer](#) (and hence truncated towards zero). Character vectors will be matched to the [names](#) of the object (or for matrices/arrays, the [dimnames](#)): see 'Character indices' below for further details.

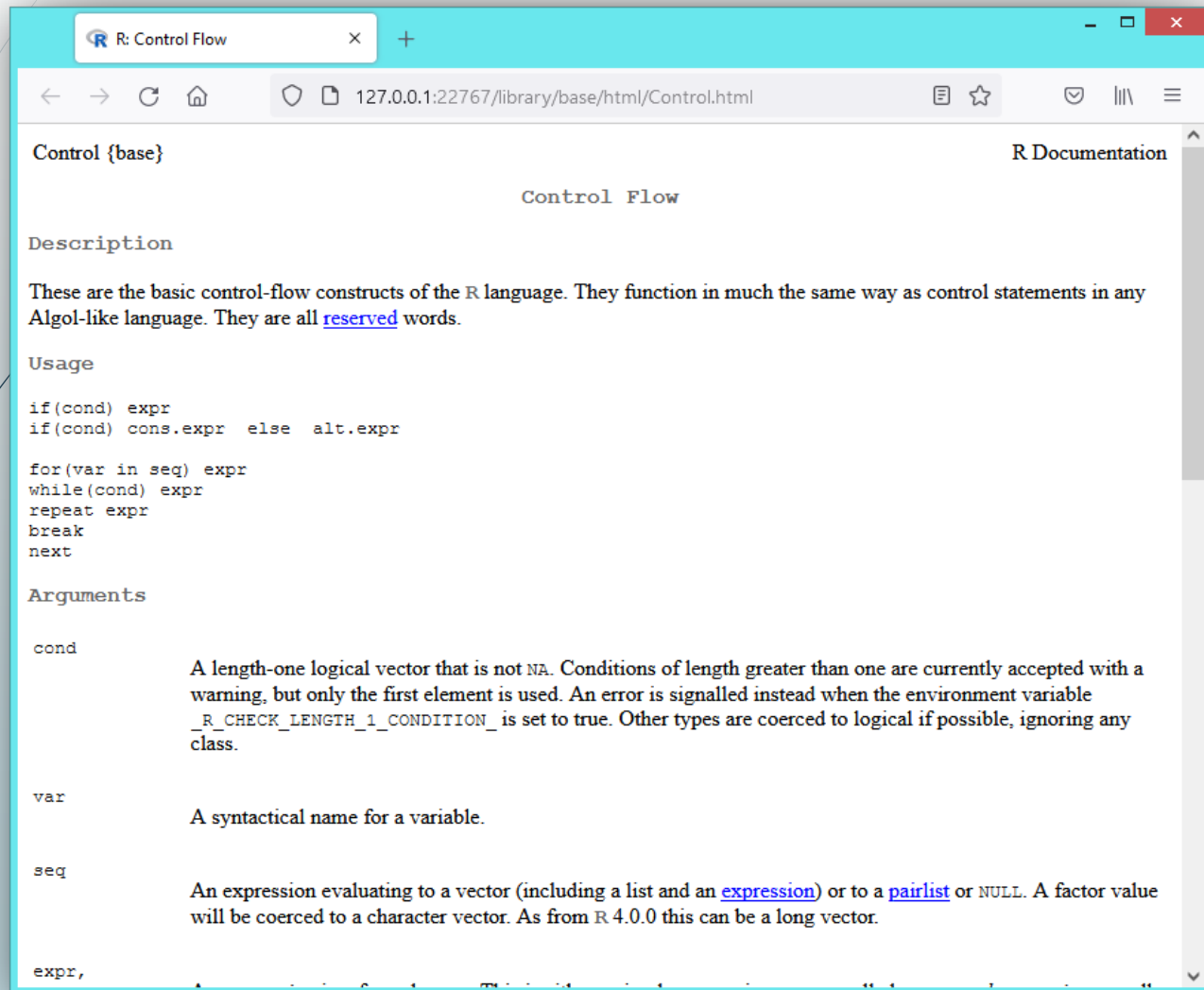
For [-indexing only: `i`, `j`, `...` can be logical vectors, indicating elements/slices to select. Such vectors are recycled if necessary to match the corresponding extent. `i`, `j`, `...` can also be negative integers, indicating elements/slices to leave out of the selection.

help() and ?



```
> help('if')
>
> ?'if'
>
> help(if)
Error: unexpected ')' in "help(if)"
>
> ?if
+ |
```

help() and ?



The screenshot shows a web browser window displaying the R documentation for Control Flow constructs. The browser's address bar shows the URL `127.0.0.1:22767/library/base/html/Control.html`. The page title is "Control {base}" and the subtitle is "Control Flow". The content is organized into sections: "Description", "Usage", and "Arguments".

Description

These are the basic control-flow constructs of the R language. They function in much the same way as control statements in any Algol-like language. They are all [reserved](#) words.

Usage

```
if(cond) expr
if(cond) cons.expr else alt.expr

for(var in seq) expr
while(cond) expr
repeat expr
break
next
```

Arguments

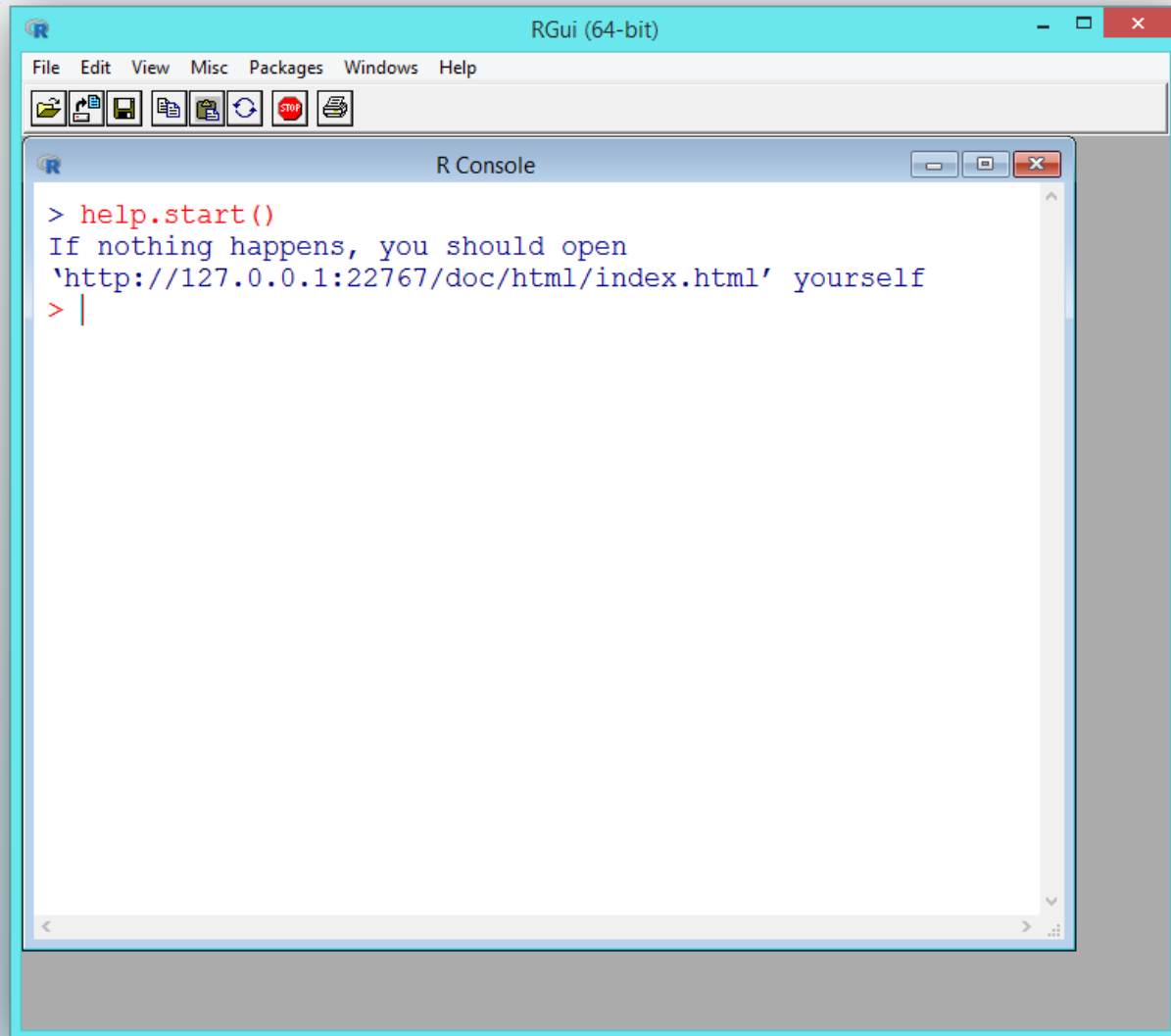
cond A length-one logical vector that is not NA. Conditions of length greater than one are currently accepted with a warning, but only the first element is used. An error is signalled instead when the environment variable `_R_CHECK_LENGTH_1_CONDITION_` is set to true. Other types are coerced to logical if possible, ignoring any class.

var A syntactical name for a variable.

seq An expression evaluating to a vector (including a list and an [expression](#)) or to a [pairlist](#) or NULL. A factor value will be coerced to a character vector. As from R 4.0.0 this can be a long vector.

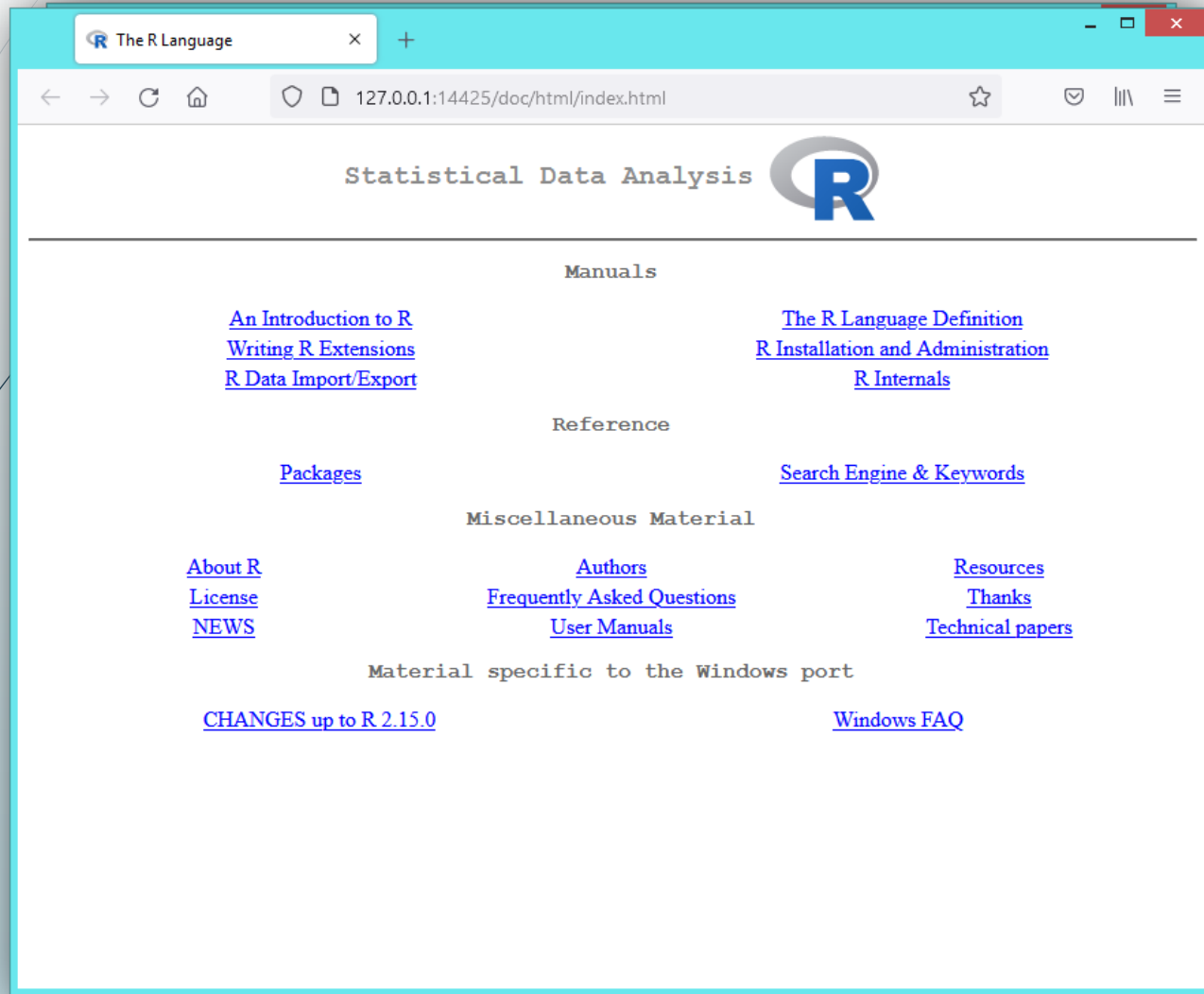
expr,

help.start()

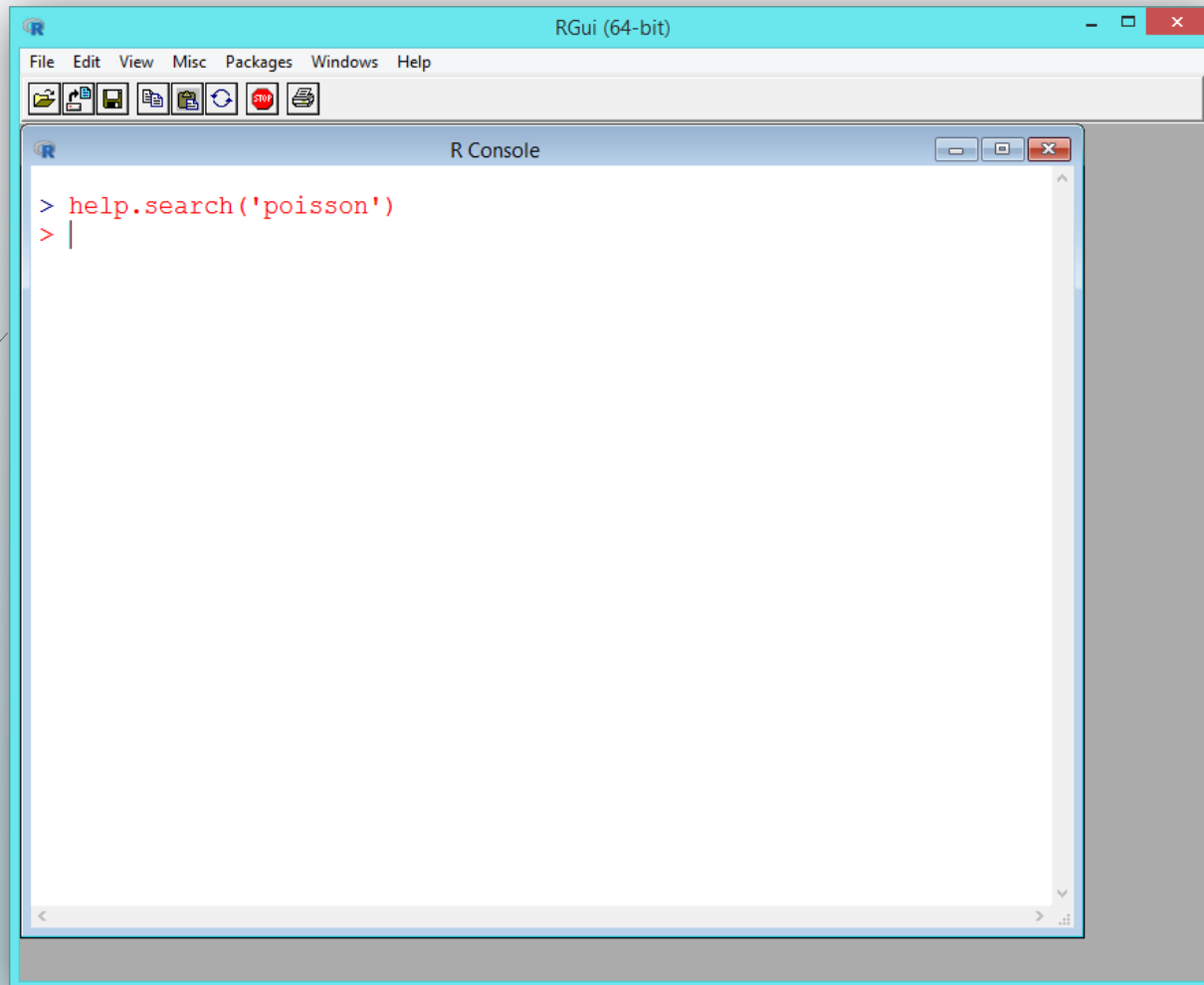


```
> help.start()
If nothing happens, you should open
'http://127.0.0.1:22767/doc/html/index.html' yourself
> |
```

help.start()



help.search()



```
> help.search('poisson')
> |
```

help.search()

R: Search Results

127.0.0.1:14425/doc/html/Search?objects=1&port=14425

Search Results

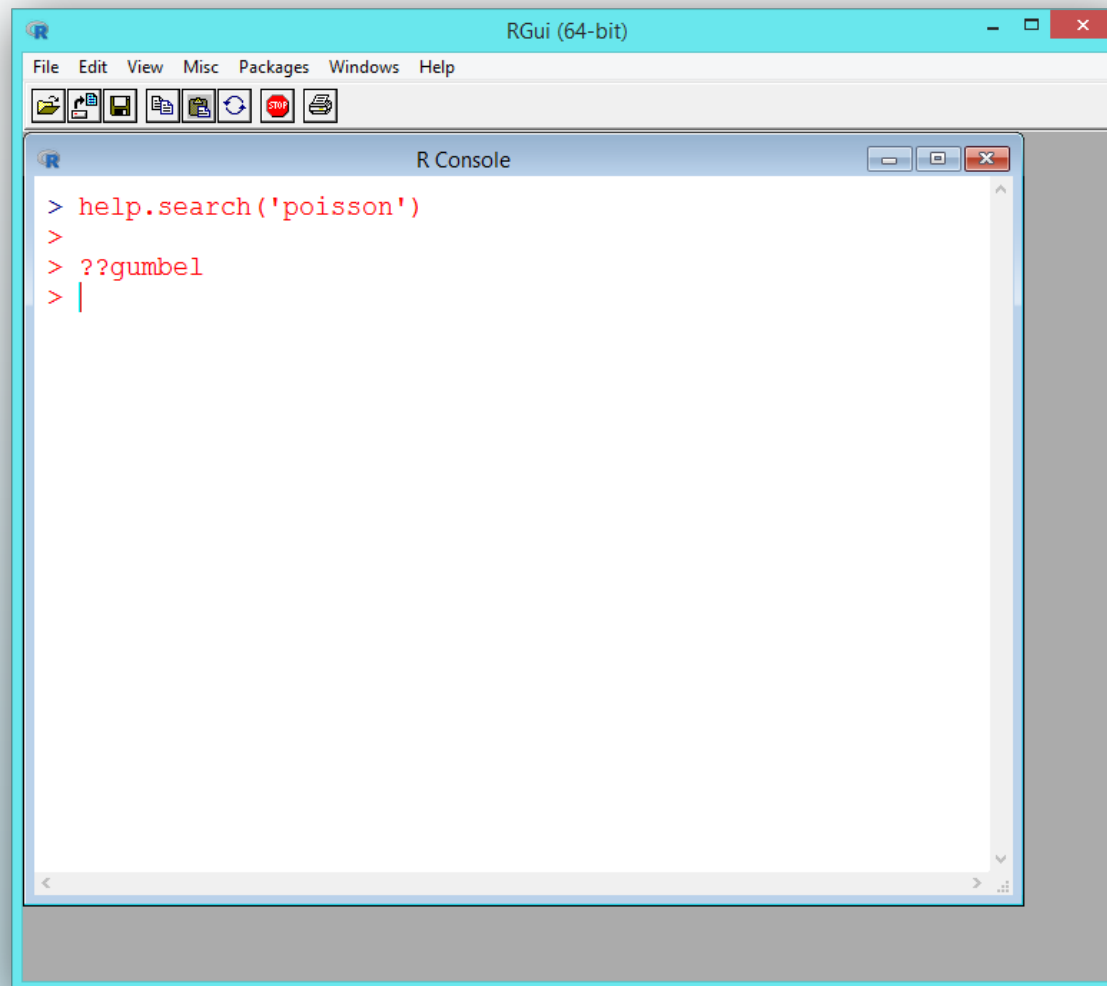
Code demonstrations:

SQUAREM::poissonmix	Examples of EM acceleration for Poisson mixture estimation using Hasselblad 1969 data.	(Run demo in console)
xgboost::poisson_regression	Poisson Regression on count data	(Run demo in console)

Help pages:


actuar::PoissonInverseGaussian	The Poisson-Inverse Gaussian Distribution
actuar::ZeroModifiedPoisson	The Zero-Modified Poisson Distribution
actuar::ZeroTruncatedPoisson	The Zero-Truncated Poisson Distribution
boot::poisons	Animal Survival Times
broom::augment.mfx	Augment data with information from a(n) mfx object
broom::glance.mfx	Glance at a(n) mfx object
broom::tidy.mfx	Tidy a(n) mfx object
extraDistr::BivPoiss	Bivariate Poisson distribution
extraDistr::GammaPoiss	Gamma-Poisson distribution
extraDistr::PoissonMix	Mixture of Poisson distributions
extraDistr::TruncPoisson	Truncated Poisson distribution
extraDistr::ZIP	Zero-inflated Poisson distribution
FactoMineR::poison	Poison
FactoMineR::poison.text	Poison
glmnet::PoissonExample	Synthetic dataset with count response

help.search() and ??



```
> help.search('poisson')
>
> ??gumbel
> |
```

help.search() and ??

Search Results 

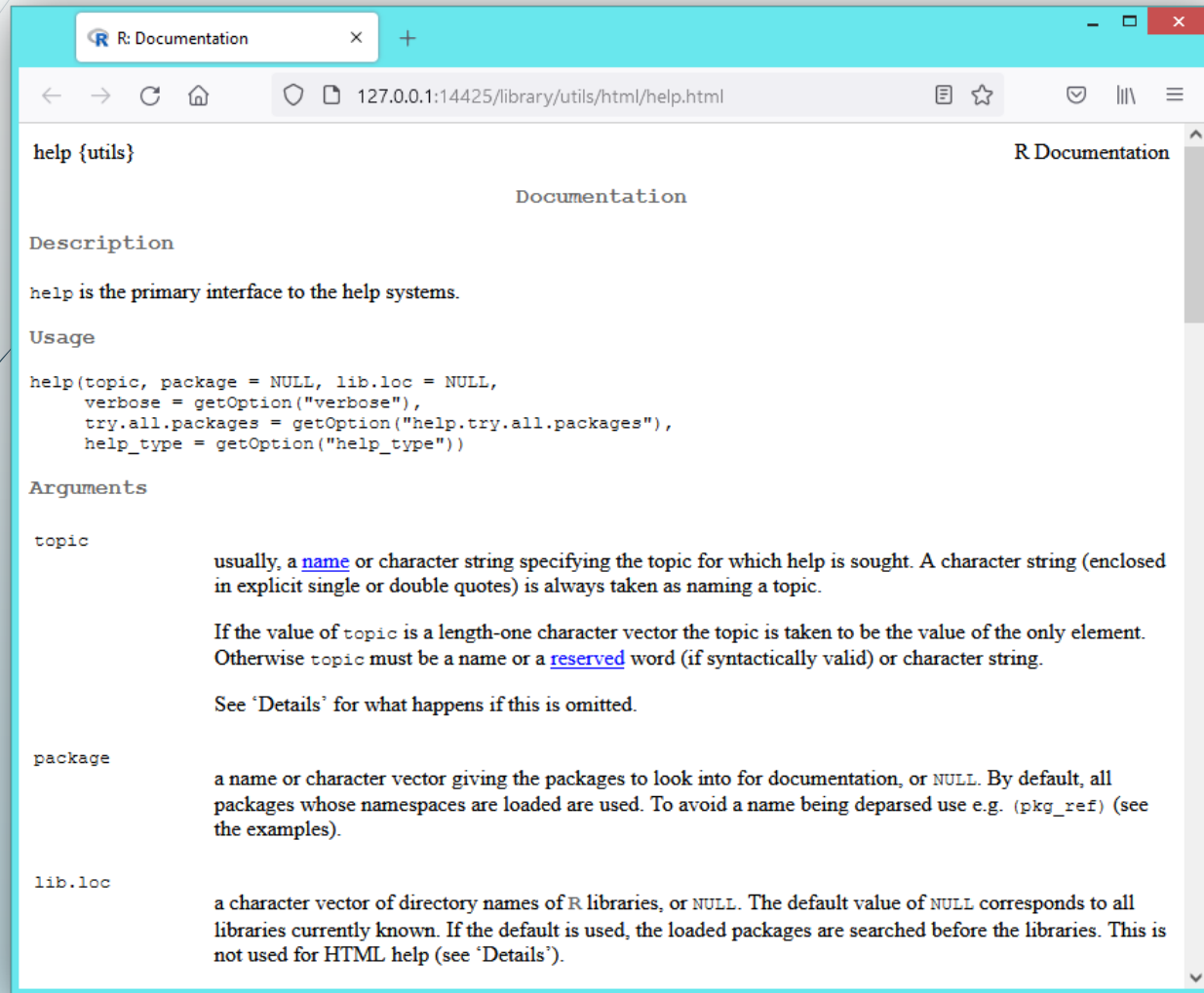
Code demonstrations:

[copula::G ak](#) Coefficients a_k for Gumbel's density derivatives, MLE, etc [\(Run demo in console\)](#)

Help pages:

- [actuar::Gumbel](#) The Gumbel Distribution
- [copula::Copula](#) Density, Evaluation, and Random Number Generation for Copula Functions
- [copula::archmCopula-class](#) Class "archmCopula"
- [copula::archmCopula](#) Construction of Archimedean Copula Class Object
- [copula::tau](#) Dependence Measures for Bivariate Copulas
- [copula::coeffG](#) Coefficients of Polynomial used for Gumbel Copula
- [copula::acopula-families](#) Specific Archimedean Copula Families ("acopula" Objects)
- [copula::fgmCopula-class](#) Class "fgmCopula" - Multivariate Multiparameter Farlie-Gumbel-Morgenstern Copulas
- [copula::iPsi](#) Generator Functions for Archimedean and Extreme-Value Copulas
- [extraDistr::Gumbel](#) Gumbel distribution
- [lmom::cdfgum](#) Gumbel (extreme-value type I) distribution
- [lmomco::are.pargum.valid](#) Are the Distribution Parameters Consistent with the Gumbel Distribution
- [lmomco::are.parrevgum.valid](#) Are the Distribution Parameters Consistent with the Reverse Gumbel Distribution
- [lmomco::cdfgum](#) Cumulative Distribution Function of the Gumbel Distribution
- [lmomco::cdfrevgum](#) Cumulative Distribution Function of the Reverse Gumbel Distribution
- [lmomco::grv2prob](#) Convert a Vector of Gumbel Reduced Variates to Annual Nonexceedance Probabilities
- [lmomco::is.gum](#) Is a Distribution Parameter Object Tuned as Gumbel

?help.search and ?help



The screenshot shows a web browser window displaying the R Documentation page for the `help` function in the `utils` package. The browser's address bar shows the URL `127.0.0.1:14425/library/utils/html/help.html`. The page content includes the following sections:

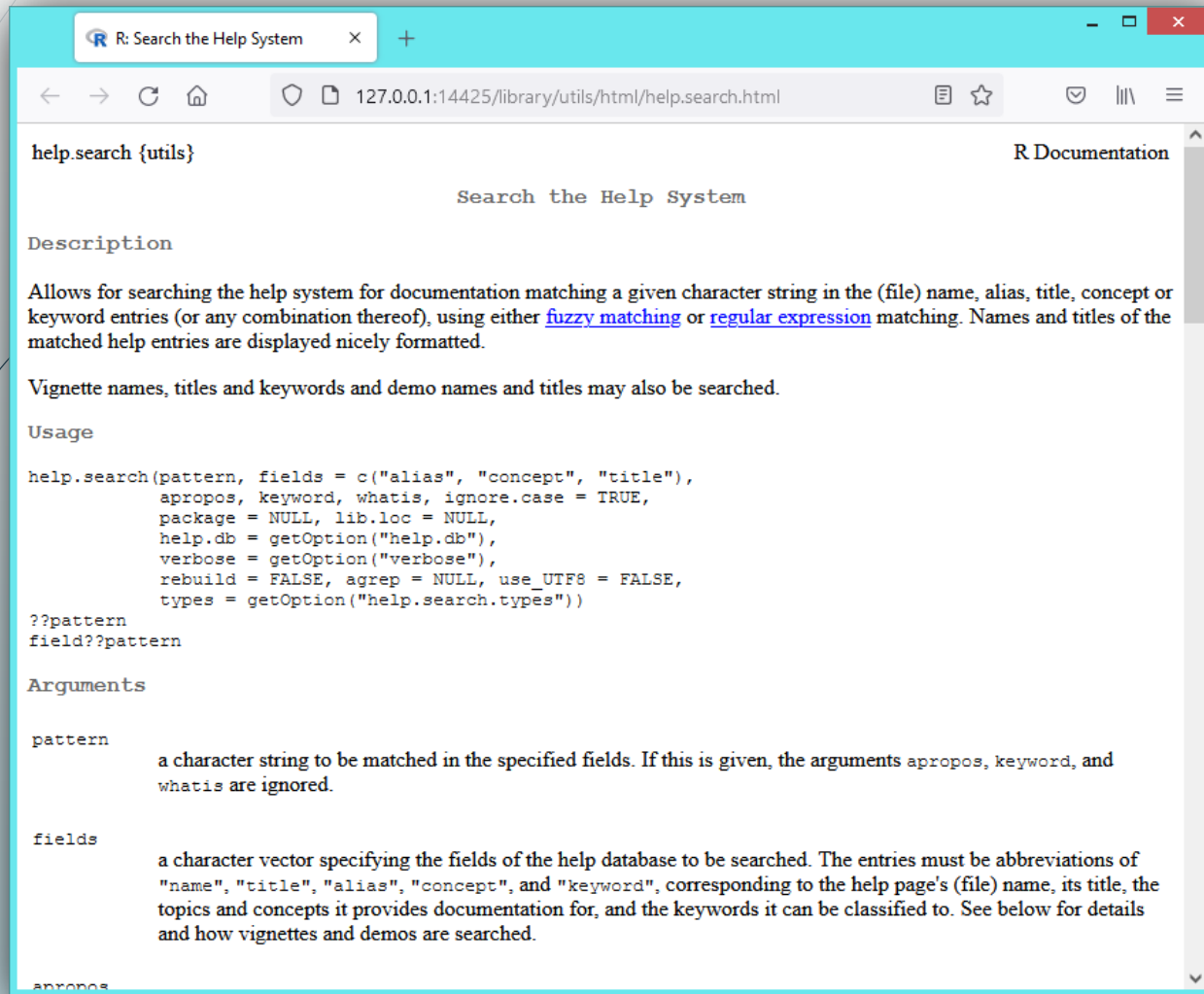
- help {utils}** (top right)
- Description**
 - `help` is the primary interface to the help systems.
- Usage**

```
help(topic, package = NULL, lib.loc = NULL,  
      verbose = getOption("verbose"),  
      try.all.packages = getOption("help.try.all.packages"),  
      help_type = getOption("help_type"))
```
- Arguments**
 - topic**: usually, a [name](#) or character string specifying the topic for which help is sought. A character string (enclosed in explicit single or double quotes) is always taken as naming a topic.

If the value of `topic` is a length-one character vector the topic is taken to be the value of the only element. Otherwise `topic` must be a name or a [reserved](#) word (if syntactically valid) or character string.

See 'Details' for what happens if this is omitted.
 - package**: a name or character vector giving the packages to look into for documentation, or `NULL`. By default, all packages whose namespaces are loaded are used. To avoid a name being deparsed use e.g. `(pkg_ref)` (see the examples).
 - lib.loc**: a character vector of directory names of R libraries, or `NULL`. The default value of `NULL` corresponds to all libraries currently known. If the default is used, the loaded packages are searched before the libraries. This is not used for HTML help (see 'Details').

?help.search and ?help



R: Search the Help System

127.0.0.1:14425/library/utils/html/help.search.html

help.search {utils}

R Documentation

Search the Help System

Description

Allows for searching the help system for documentation matching a given character string in the (file) name, alias, title, concept or keyword entries (or any combination thereof), using either [fuzzy matching](#) or [regular expression](#) matching. Names and titles of the matched help entries are displayed nicely formatted.

Vignette names, titles and keywords and demo names and titles may also be searched.

Usage

```
help.search(pattern, fields = c("alias", "concept", "title"),
            apropos, keyword, whatis, ignore.case = TRUE,
            package = NULL, lib.loc = NULL,
            help.db = getOption("help.db"),
            verbose = getOption("verbose"),
            rebuild = FALSE, agrep = NULL, use_UTF8 = FALSE,
            types = getOption("help.search.types"))
```

??pattern
field??pattern

Arguments

pattern
a character string to be matched in the specified fields. If this is given, the arguments `apropos`, `keyword`, and `whatis` are ignored.

fields
a character vector specifying the fields of the help database to be searched. The entries must be abbreviations of "name", "title", "alias", "concept", and "keyword", corresponding to the help page's (file) name, its title, the topics and concepts it provides documentation for, and the keywords it can be classified to. See below for details and how vignettes and demos are searched.

apropos

example ()

```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> example(floor)

floor> round(.5 + -2:4) # IEEE / IEC rounding: -2 0 0 2 2 4 4
[1] -2 0 0 2 2 4 4

floor> ## (this is *good* behaviour -- do *NOT* report it as bug !)
floor>
floor> ( x1 <- seq(-2, 4, by = .5) )
[1] -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0 2.5 3.0
[12] 3.5 4.0

floor> round(x1) #-- IEEE / IEC rounding !
[1] -2 -2 -1 0 0 0 1 2 2 2 3 4 4

floor> x1[trunc(x1) != floor(x1)]
[1] -1.5 -0.5

floor> x1[round(x1) != floor(x1 + .5)]
[1] -1.5 0.5 2.5

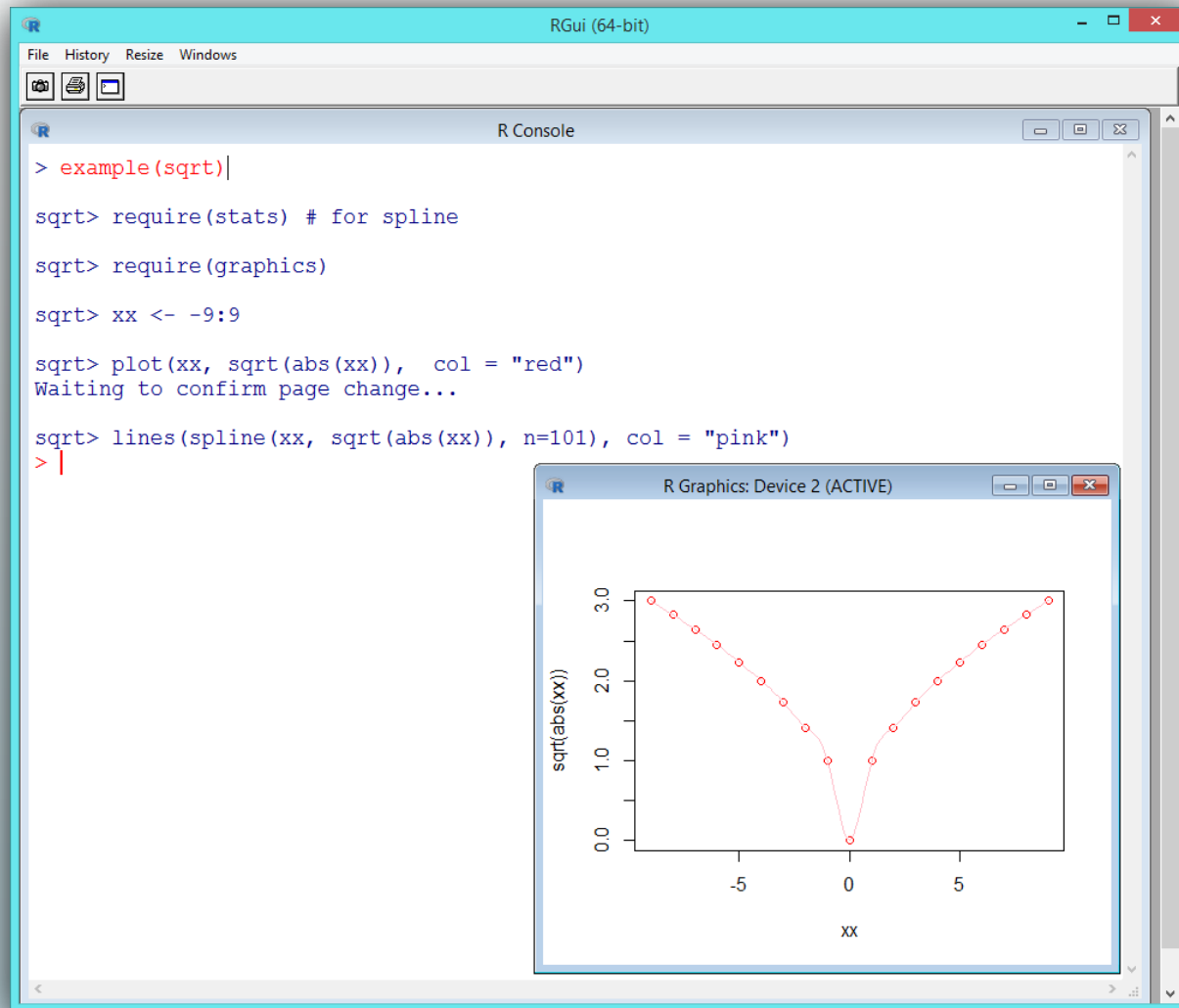
floor> (non.int <- ceiling(x1) != floor(x1))
[1] FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE
[10] TRUE FALSE TRUE FALSE

floor> x2 <- pi * 100^(-1:3)

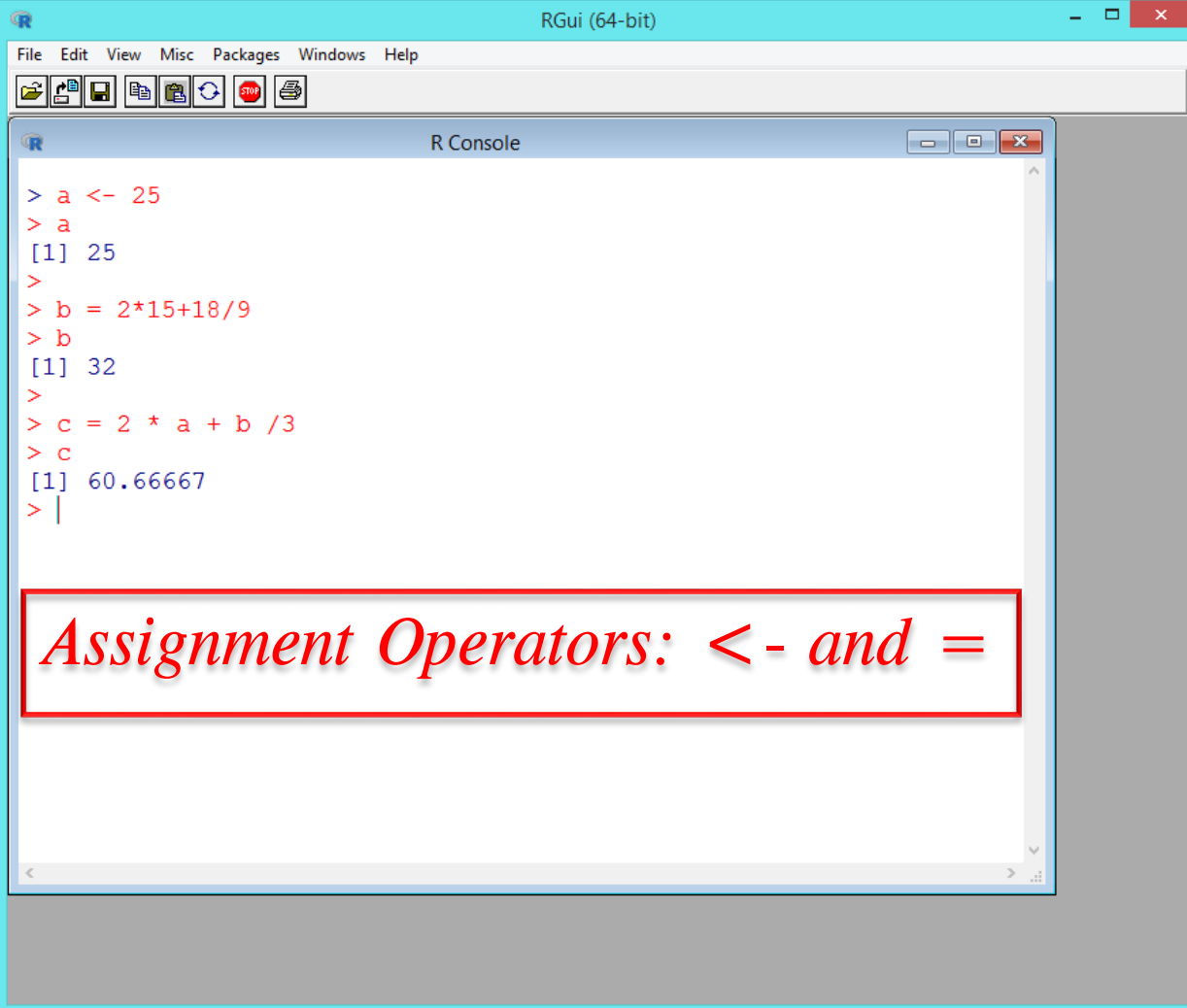
floor> round(x2, 3)
[1] 0.031 3.142 314.159 31415.927
[5] 3141592.654

floor> signif(x2, 3)
[1] 3.14e-02 3.14e+00 3.14e+02 3.14e+04 3.14e+06
> |
```

example ()



Assignment



The screenshot shows the RGui (64-bit) window with the R Console open. The console displays the following commands and their outputs:

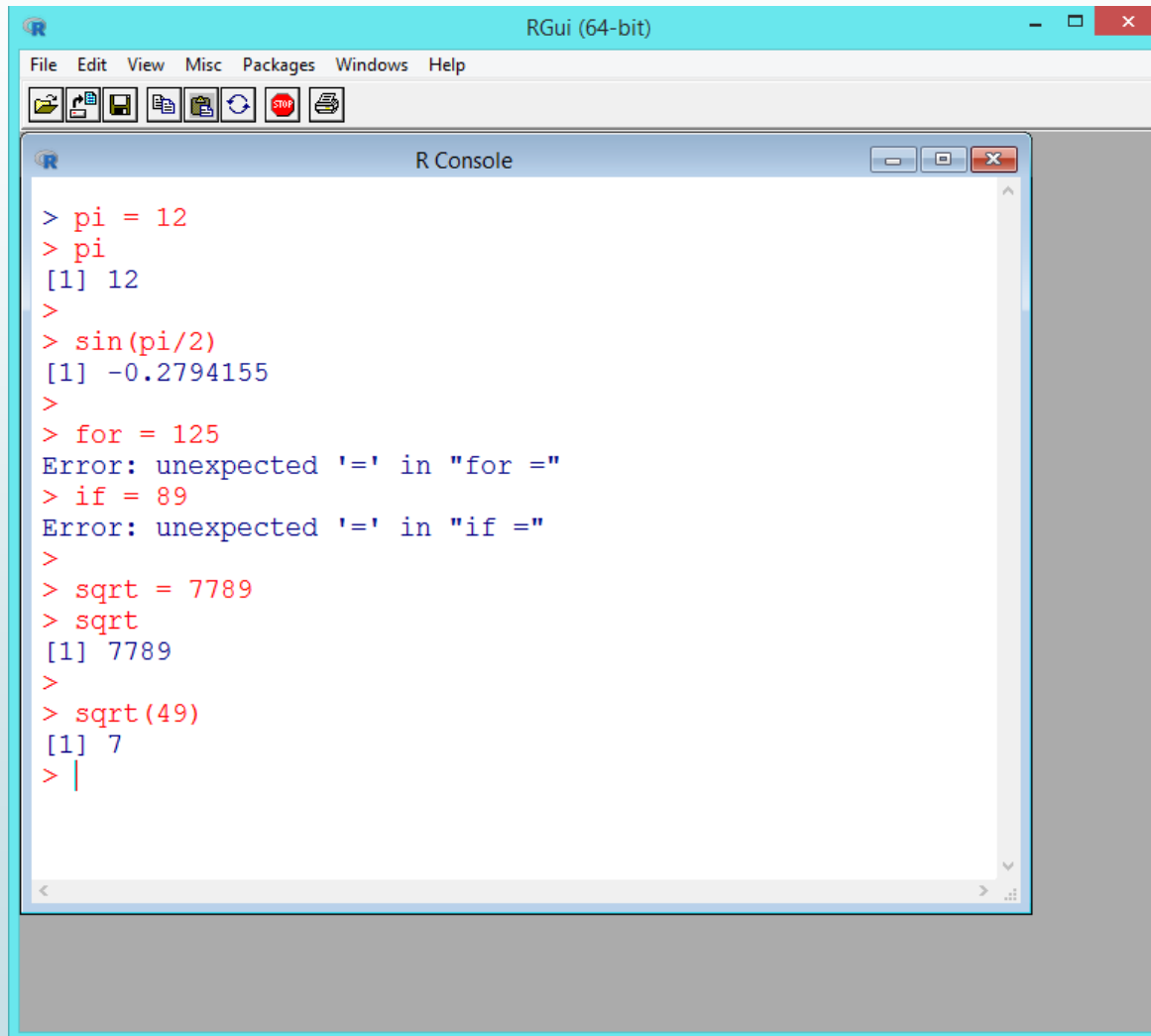
```
> a <- 25
> a
[1] 25
>
> b = 2*15+18/9
> b
[1] 32
>
> c = 2 * a + b /3
> c
[1] 60.66667
> |
```

Below the console, a red-bordered box contains the text: *Assignment Operators: <- and =*

Assignment

- All alphanumeric symbols are allowed¹ (and in some countries this includes accented letters) plus ‘.’ and ‘_’.
- A name must start with ‘.’ or a letter, and if it starts with ‘.’ the second character must not be a digit.
- Names are effectively unlimited in length.

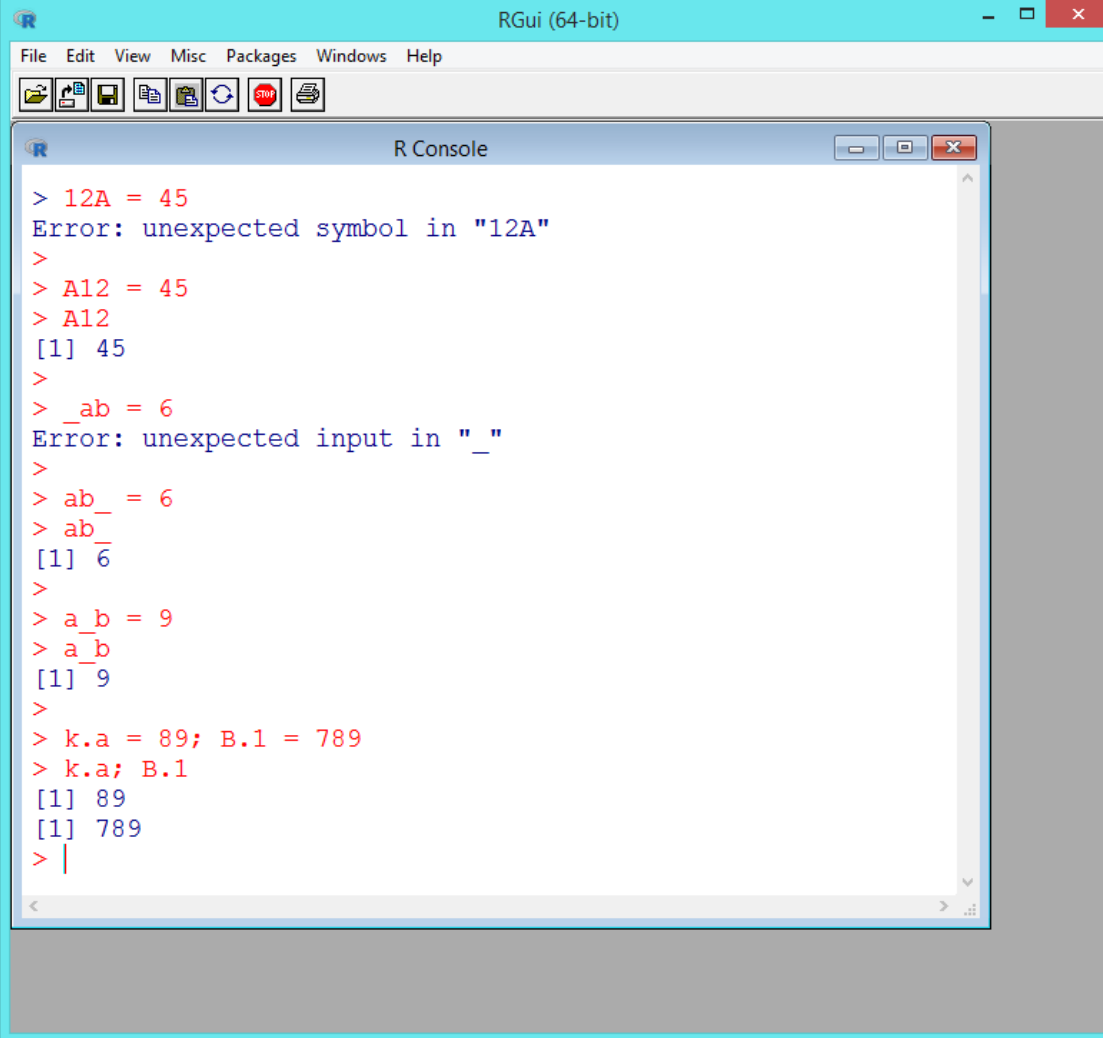
Assignment



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> pi = 12
> pi
[1] 12
>
> sin(pi/2)
[1] -0.2794155
>
> for = 125
Error: unexpected '=' in "for ="
> if = 89
Error: unexpected '=' in "if ="
>
> sqrt = 7789
> sqrt
[1] 7789
>
> sqrt(49)
[1] 7
> |
```

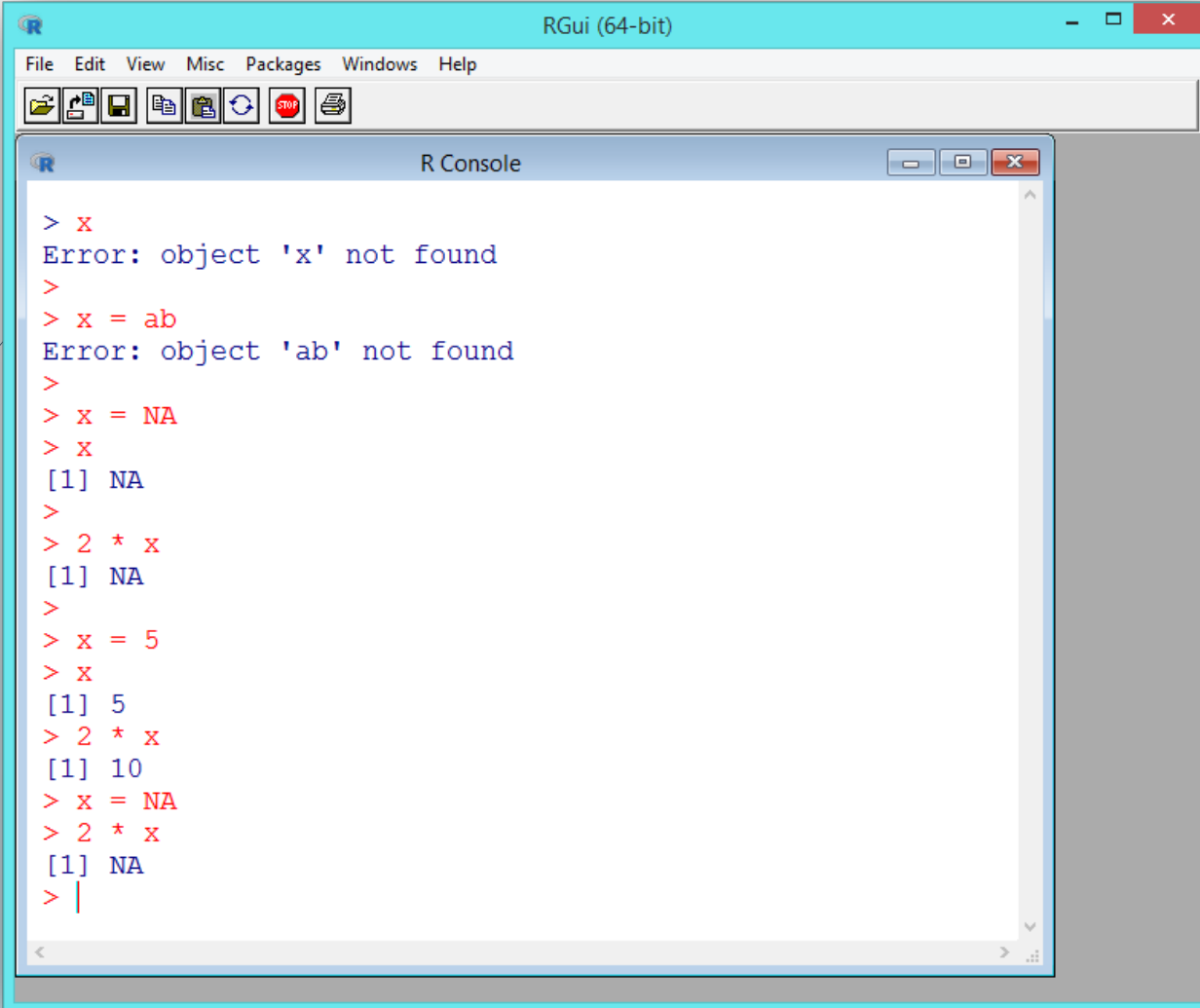
Assignment



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

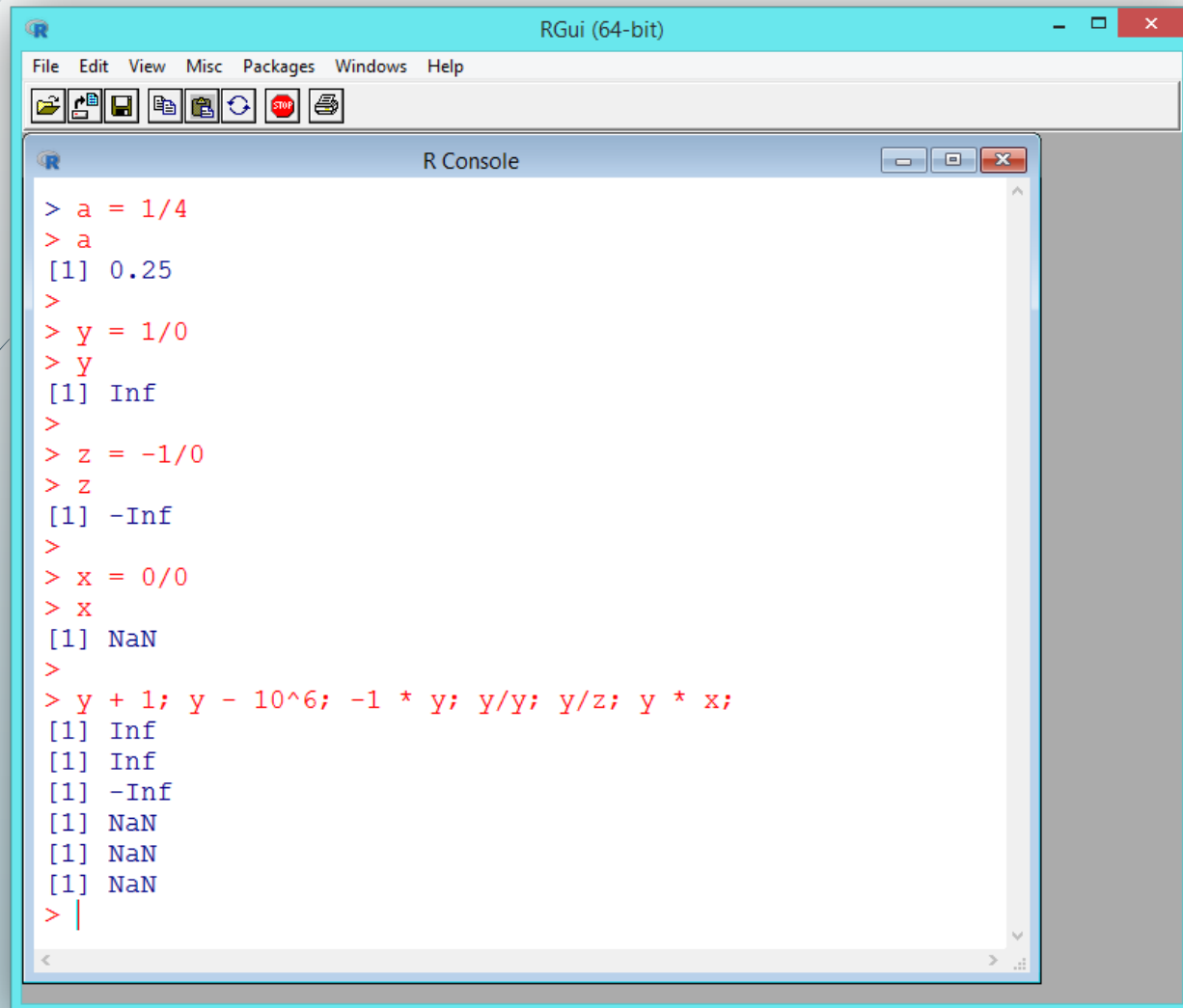
R Console
> 12A = 45
Error: unexpected symbol in "12A"
>
> A12 = 45
> A12
[1] 45
>
> _ab = 6
Error: unexpected input in "_"
>
> ab_ = 6
> ab_
[1] 6
>
> a_b = 9
> a_b
[1] 9
>
> k.a = 89; B.1 = 789
> k.a; B.1
[1] 89
[1] 789
> |
```

Missing values or NA



```
> x
Error: object 'x' not found
>
> x = ab
Error: object 'ab' not found
>
> x = NA
> x
[1] NA
>
> 2 * x
[1] NA
>
> x = 5
> x
[1] 5
> 2 * x
[1] 10
> x = NA
> 2 * x
[1] NA
> |
```

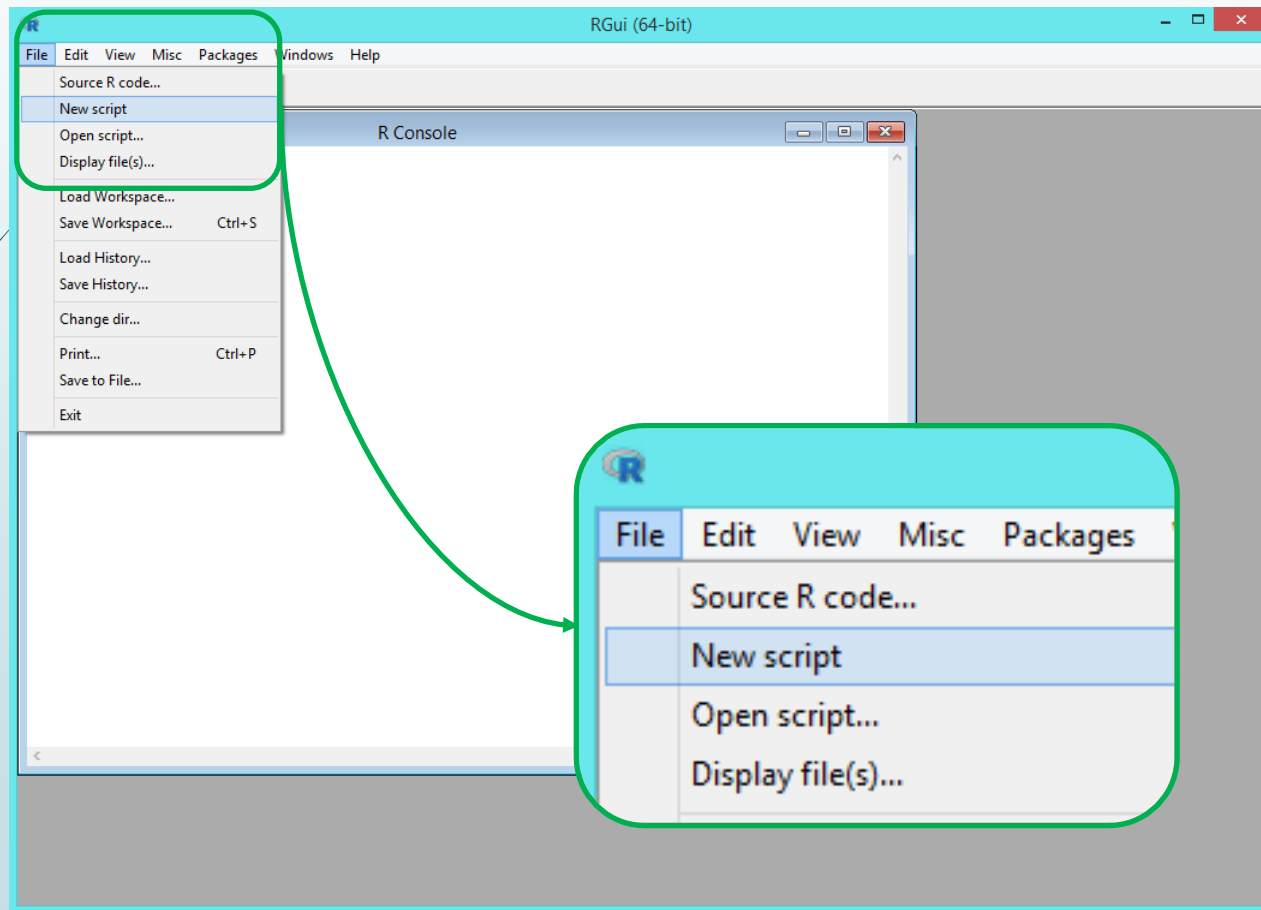
NaN and Inf



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> a = 1/4
> a
[1] 0.25
>
> y = 1/0
> y
[1] Inf
>
> z = -1/0
> z
[1] -Inf
>
> x = 0/0
> x
[1] NaN
>
> y + 1; y - 10^6; -1 * y; y/y; y/z; y * x;
[1] Inf
[1] Inf
[1] -Inf
[1] NaN
[1] NaN
[1] NaN
> |
```

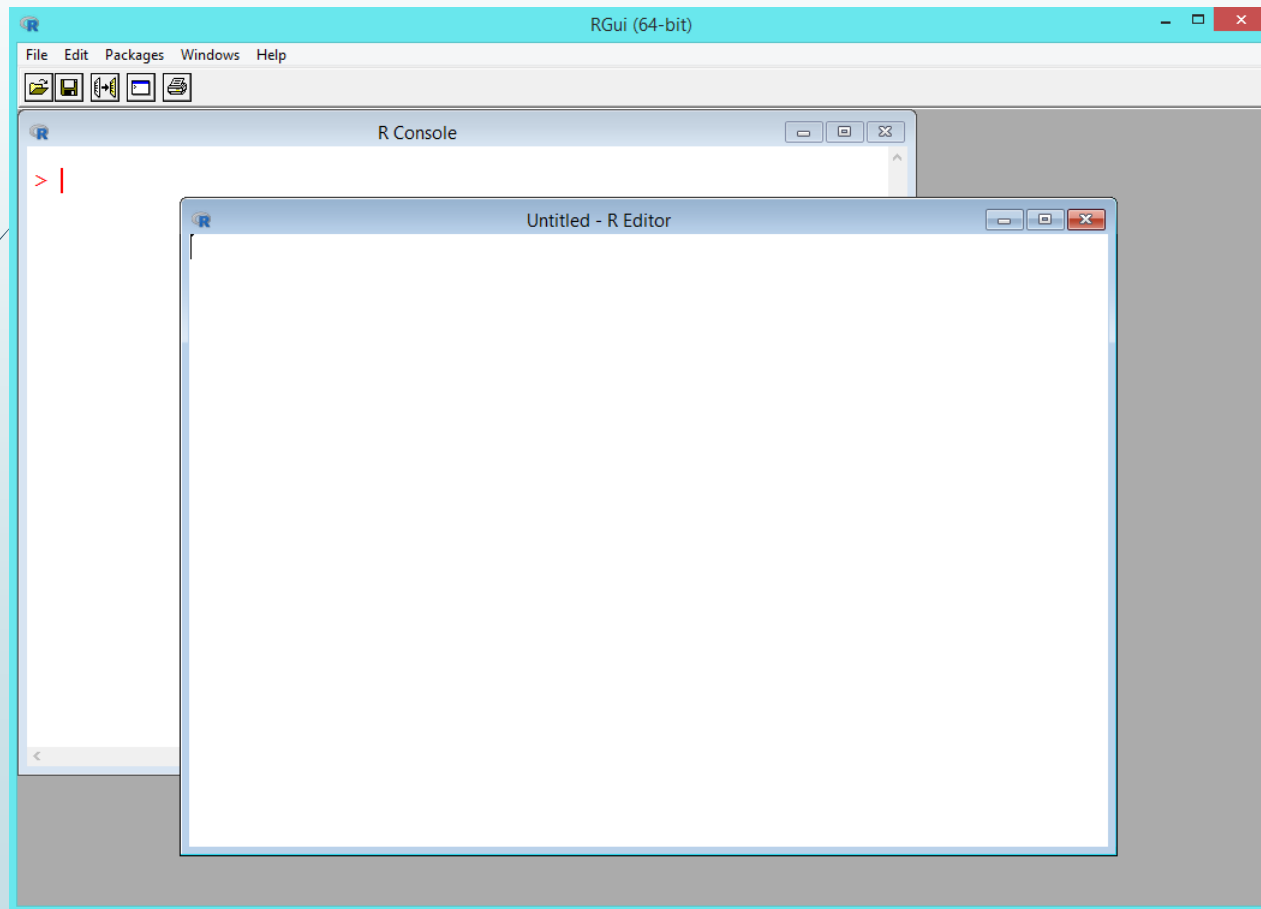

R Editor and Scripts

► Creating a New Script



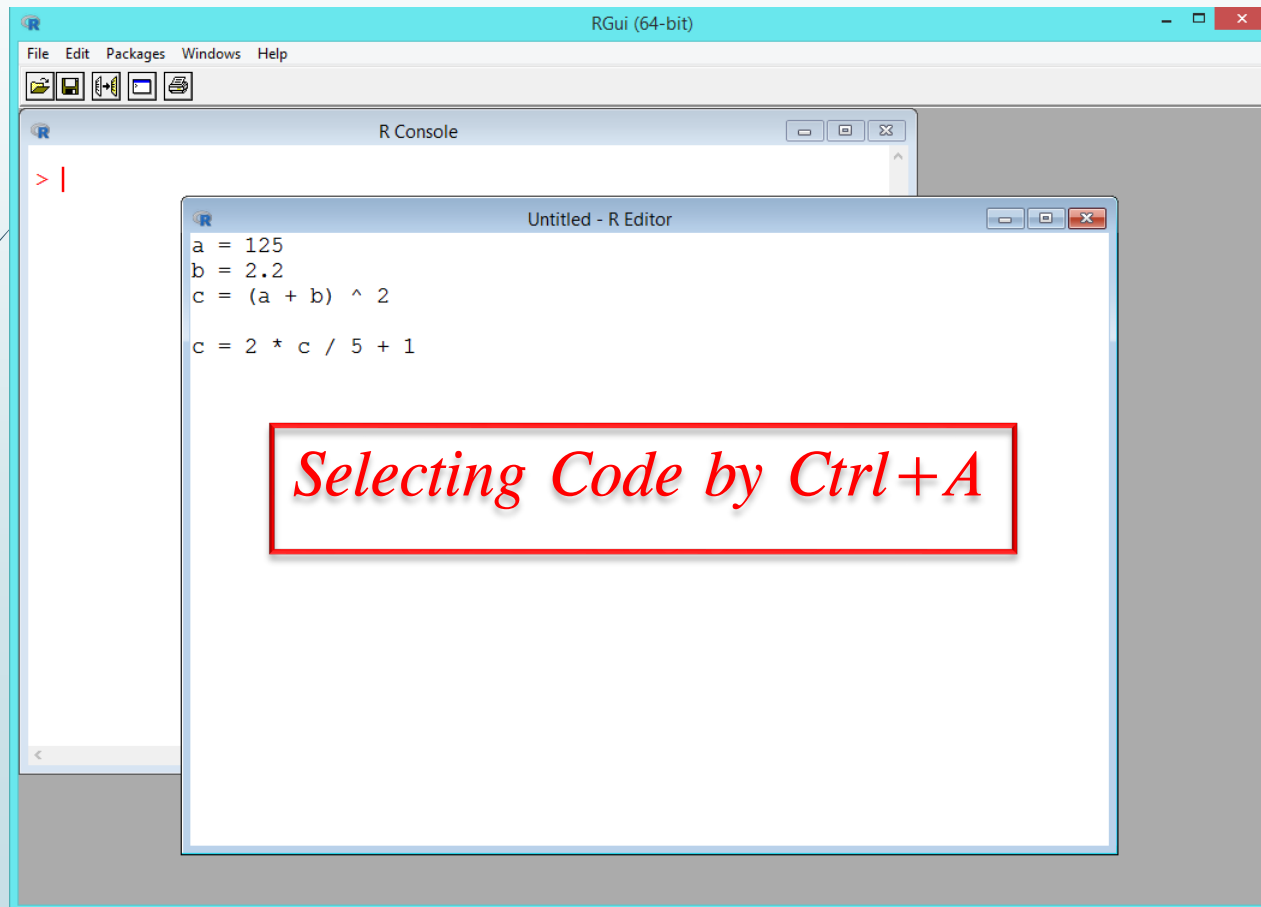
R Editor and Scripts

► Creating a New Script (1)



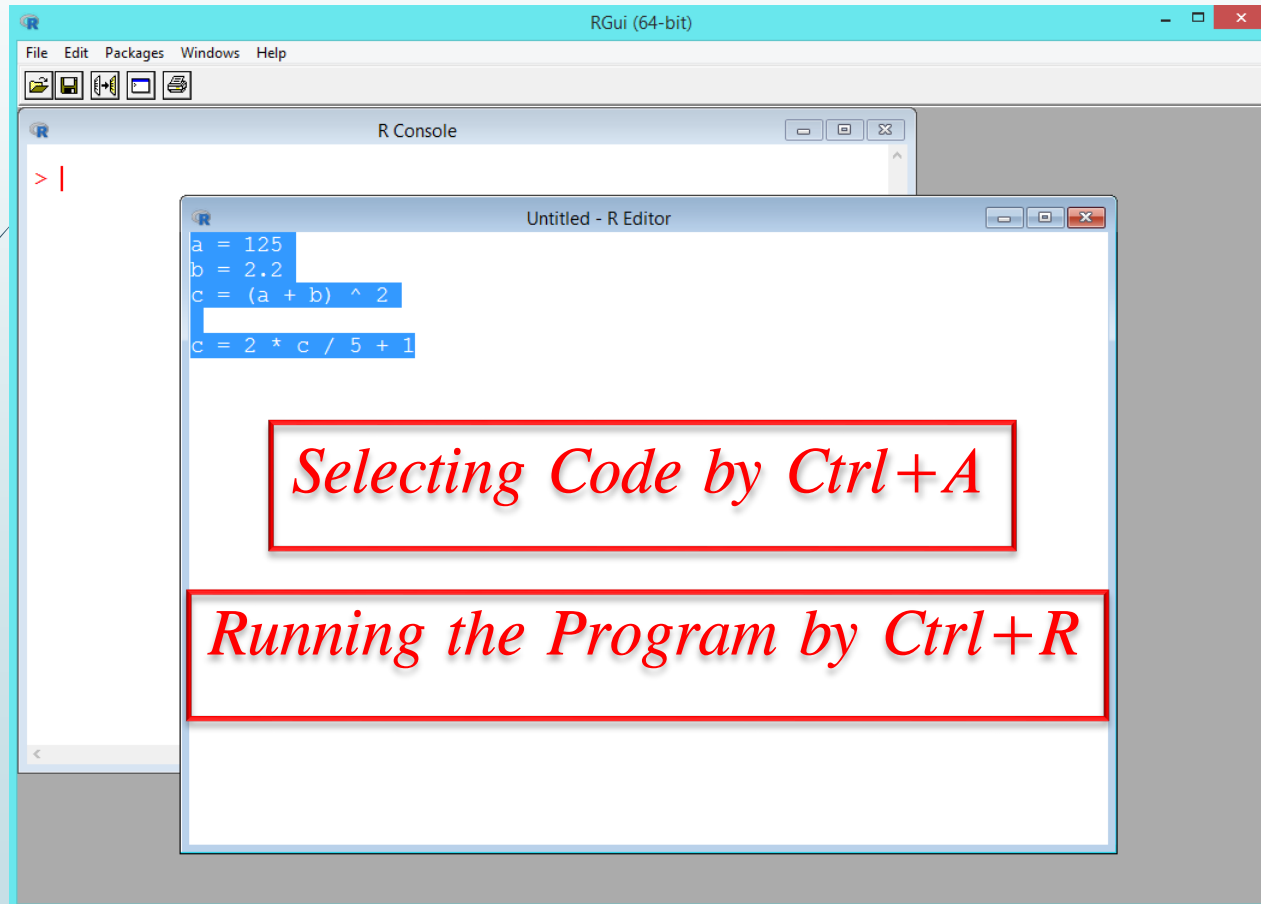
R Editor and Scripts

► Creating a New Script (2)



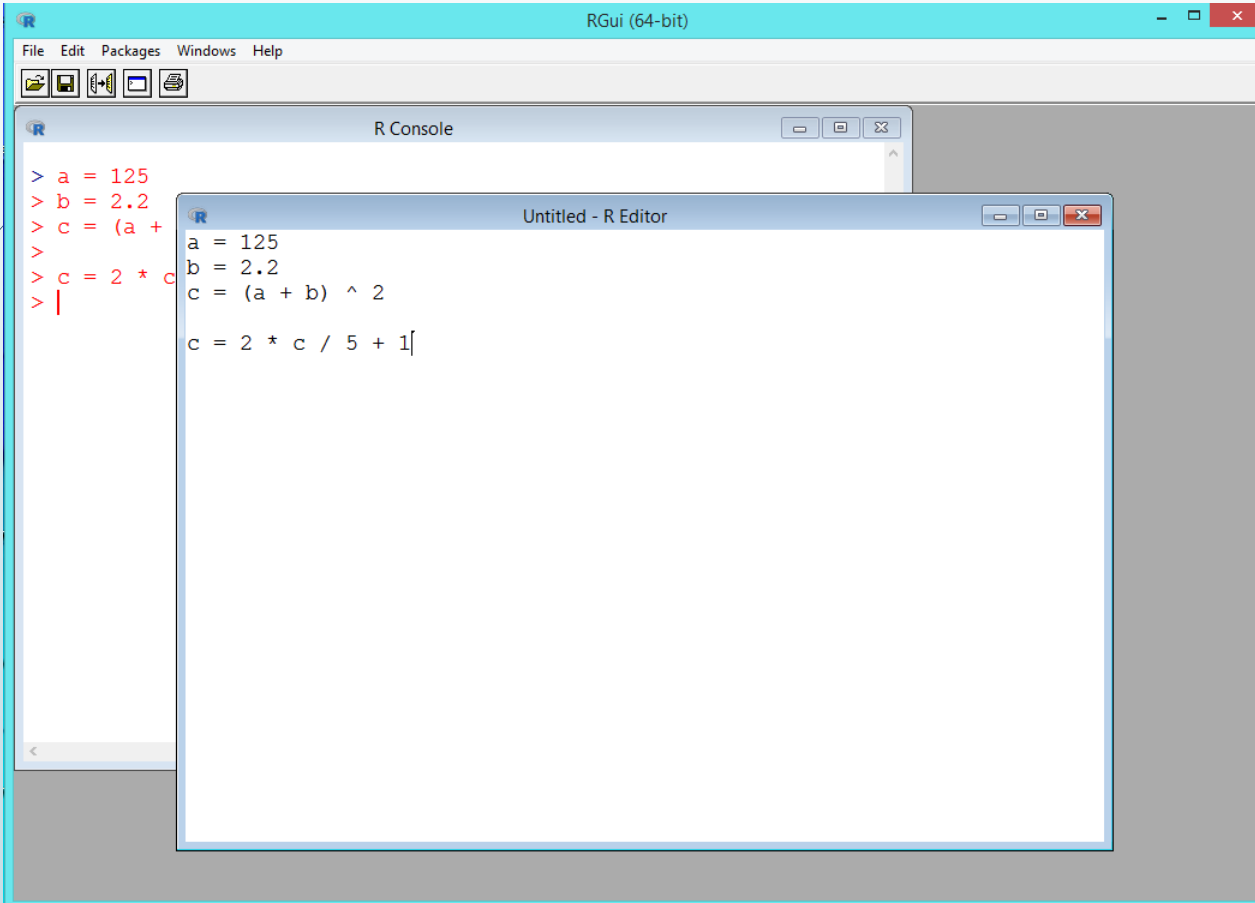
R Editor and Scripts

▶ Running a Program (1)



R Editor and Scripts

▶ Running a Program (2)

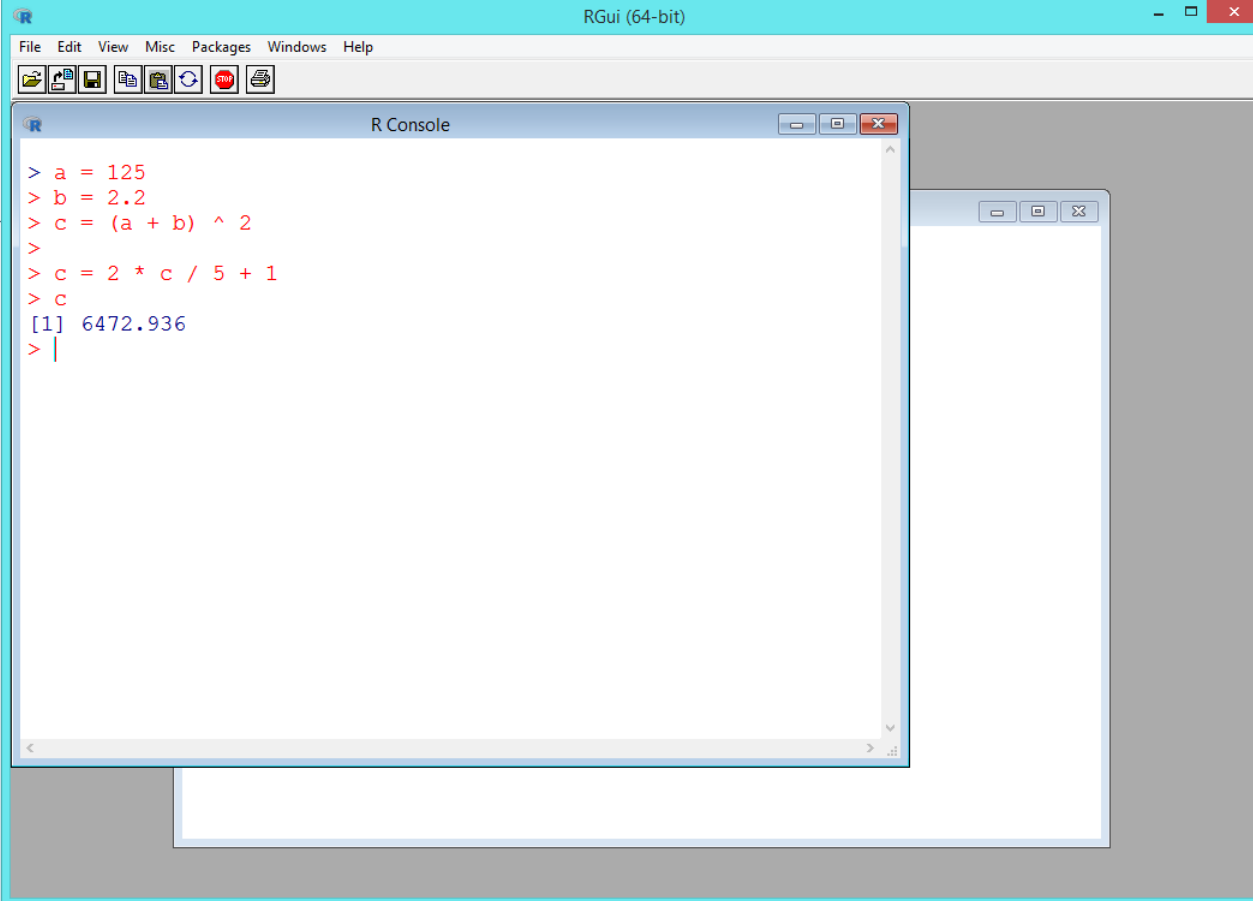


```
RGui (64-bit)
File Edit Packages Windows Help
R Console
> a = 125
> b = 2.2
> c = (a +
>
> c = 2 * c
> |
>

Untitled - R Editor
a = 125
b = 2.2
c = (a + b) ^ 2
c = 2 * c / 5 + 1[
```

R Editor and Scripts

▶ Running a Program (3)

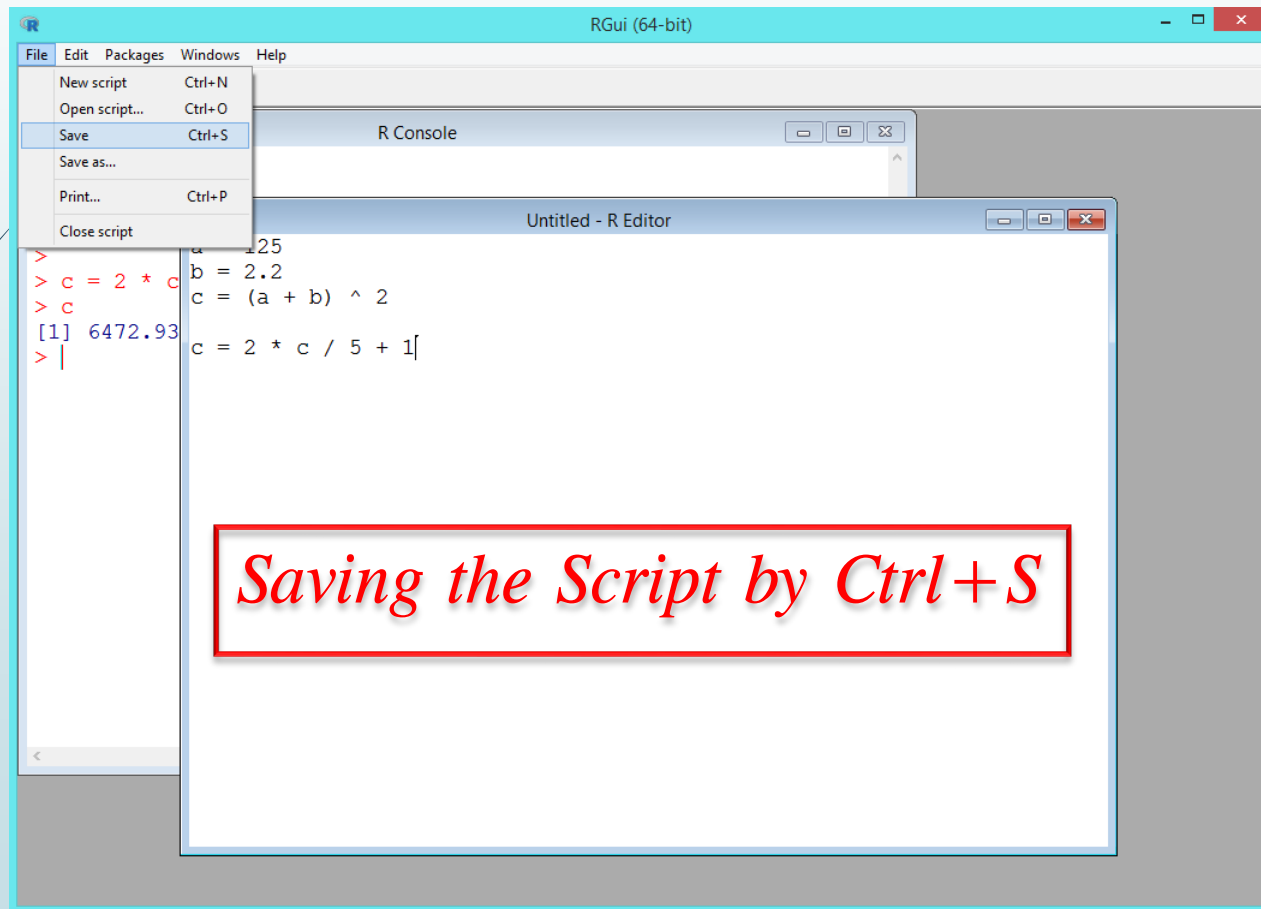


The screenshot shows the RGui (64-bit) window with a menu bar (File, Edit, View, Misc, Packages, Windows, Help) and a toolbar. The R Console window is open, displaying the following R code and output:

```
> a = 125
> b = 2.2
> c = (a + b) ^ 2
>
> c = 2 * c / 5 + 1
> c
[1] 6472.936
> |
```

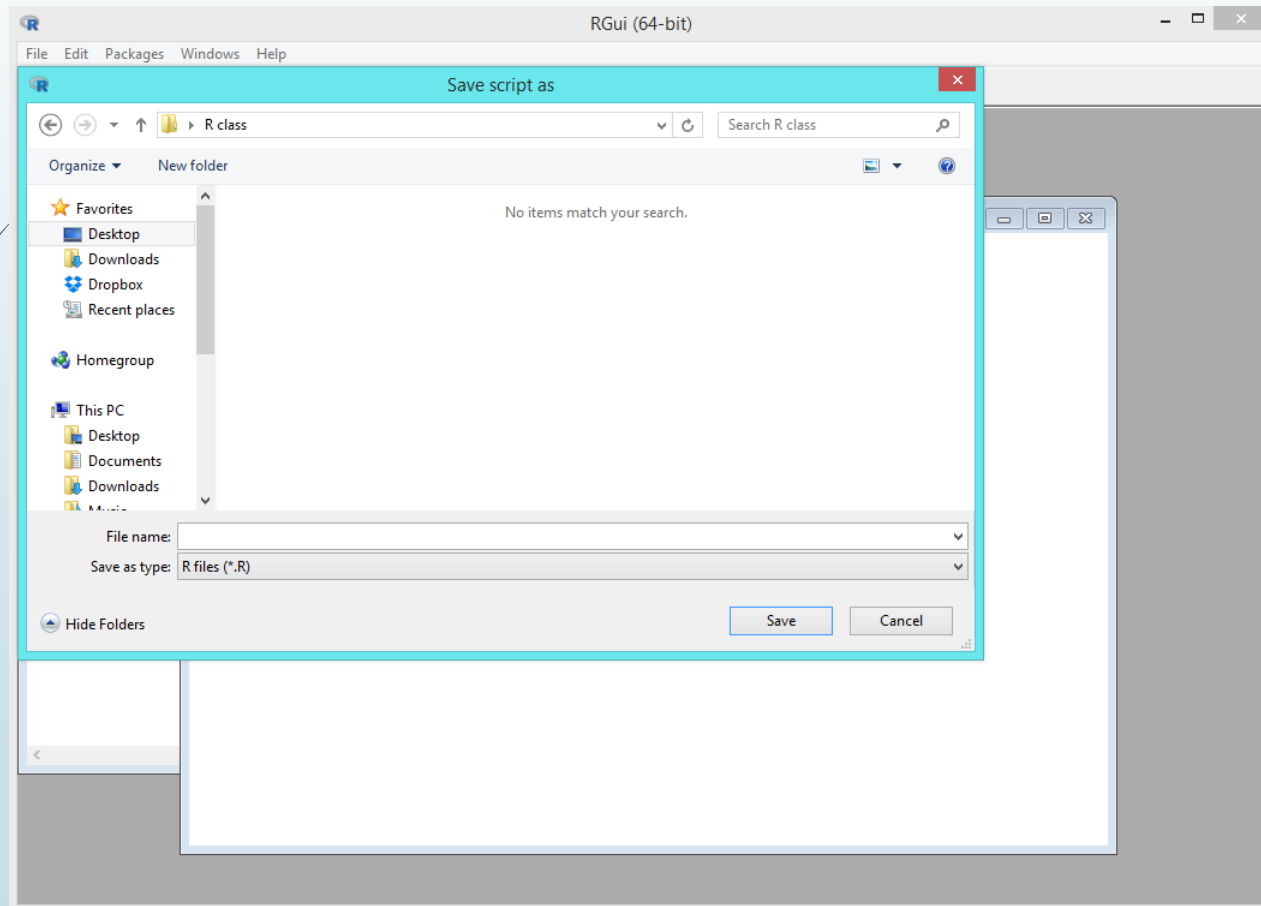
R Editor and Scripts

► Saving a Script (1)



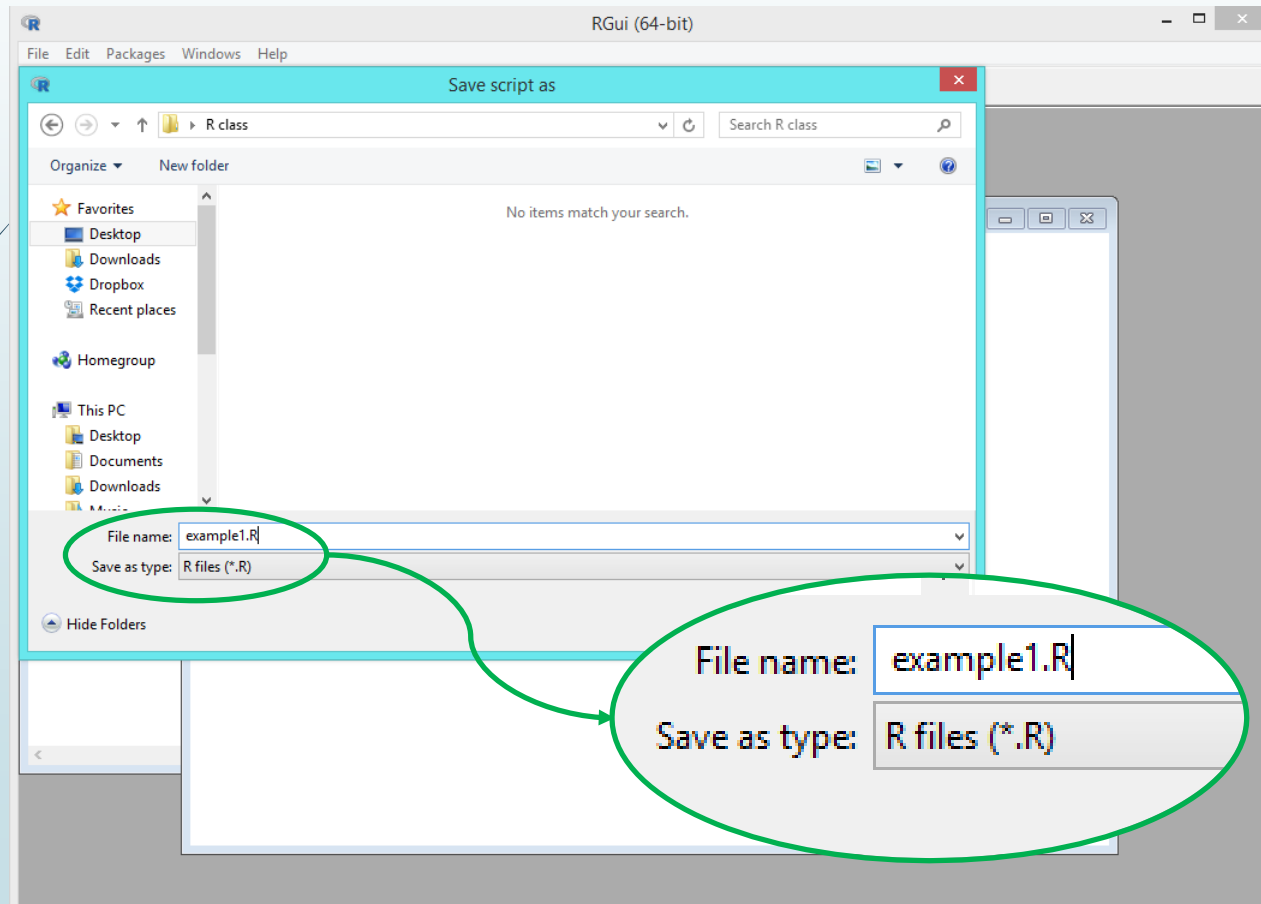
R Editor and Scripts

► Saving a Script (2)



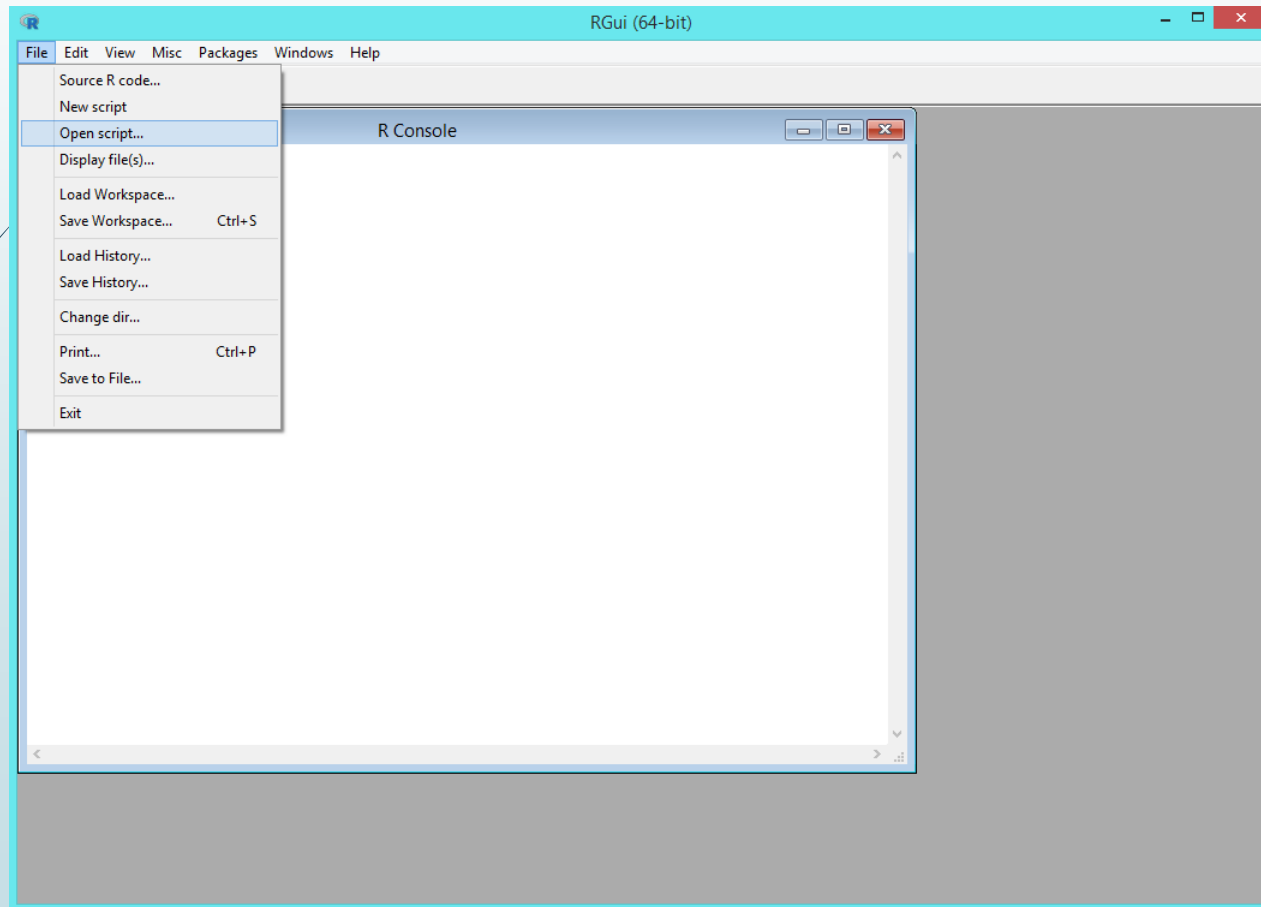
R Editor and Scripts

► Saving a Script (3)



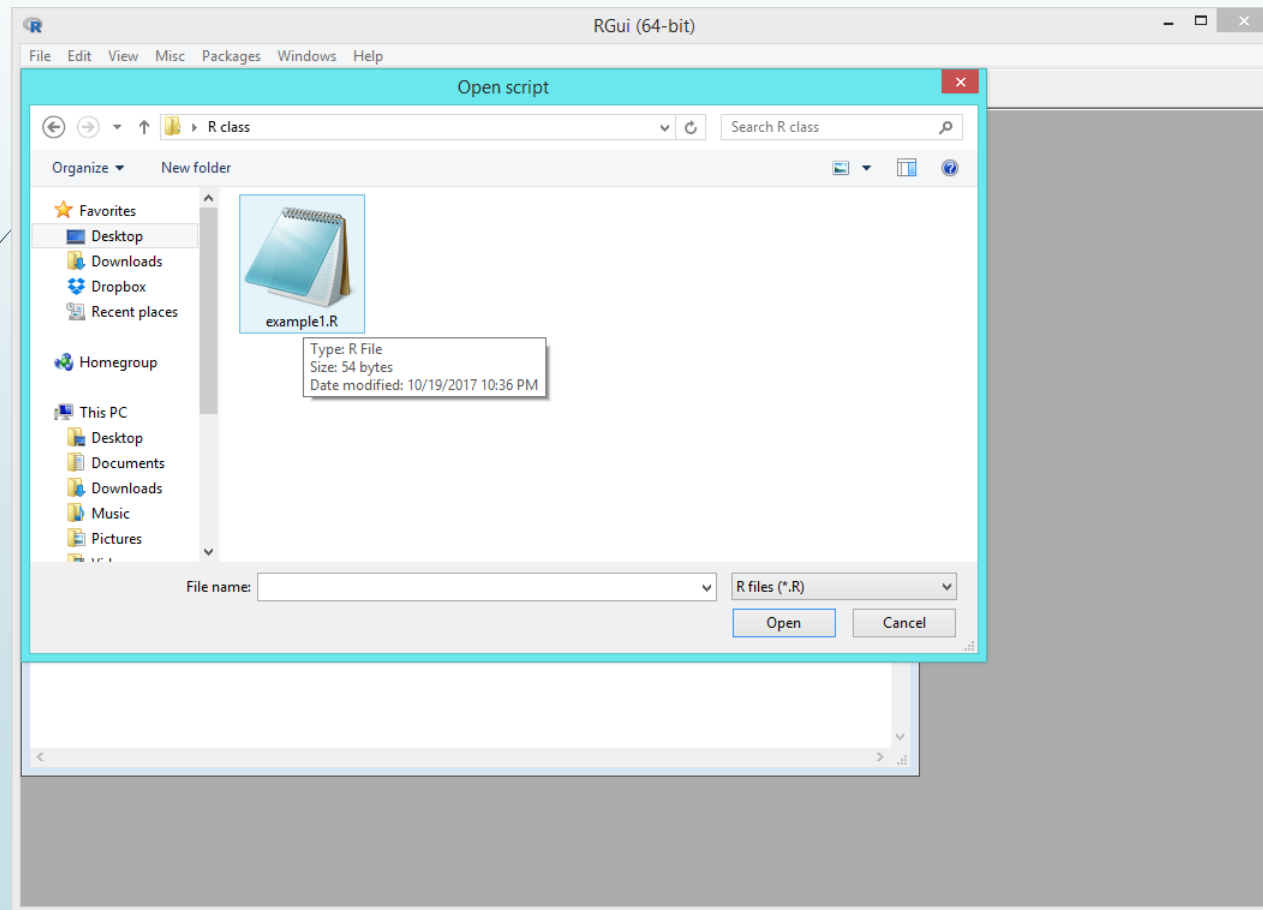
R Editor and Scripts

➤ Opening a Script (1)



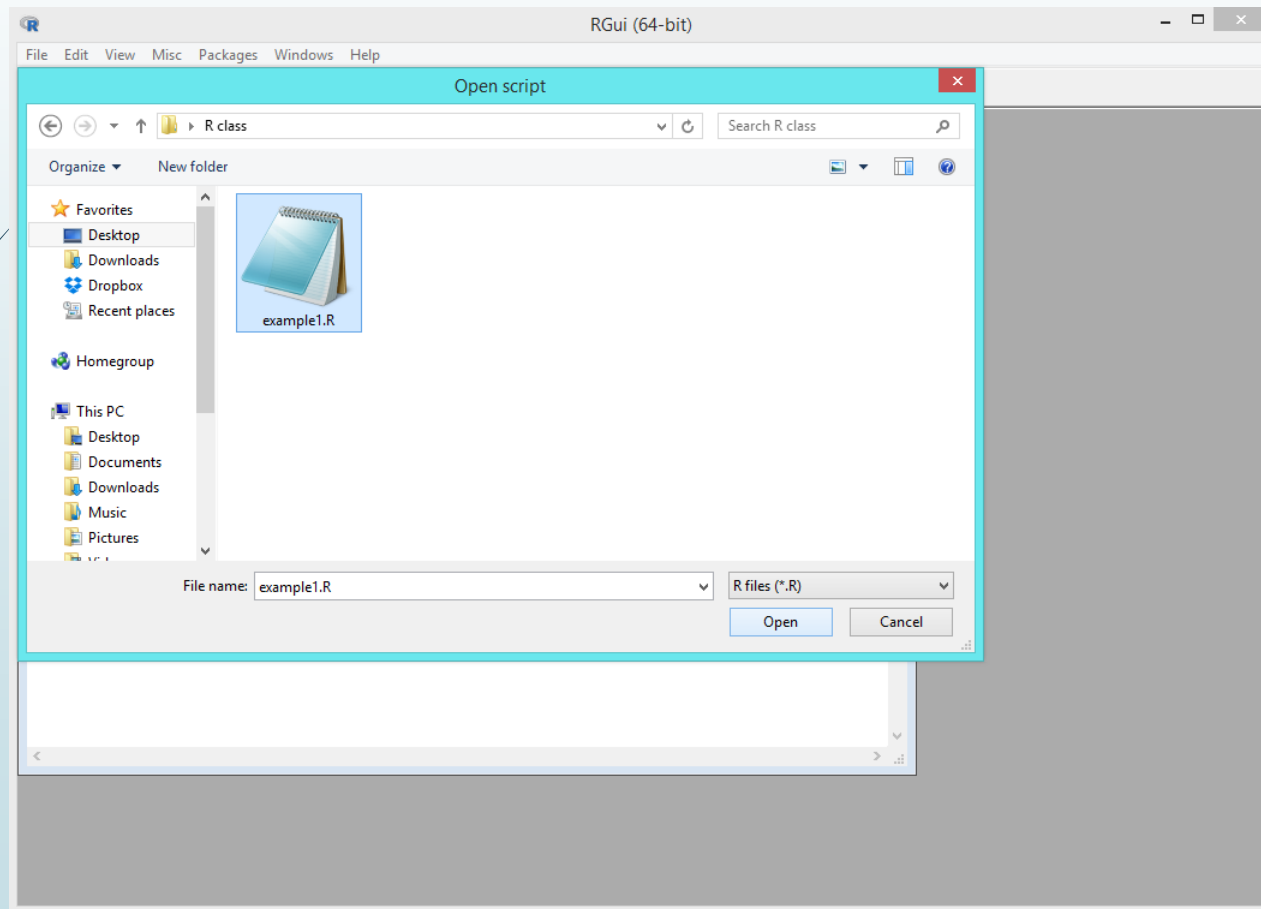
R Editor and Scripts

➤ Opening a Script (2)



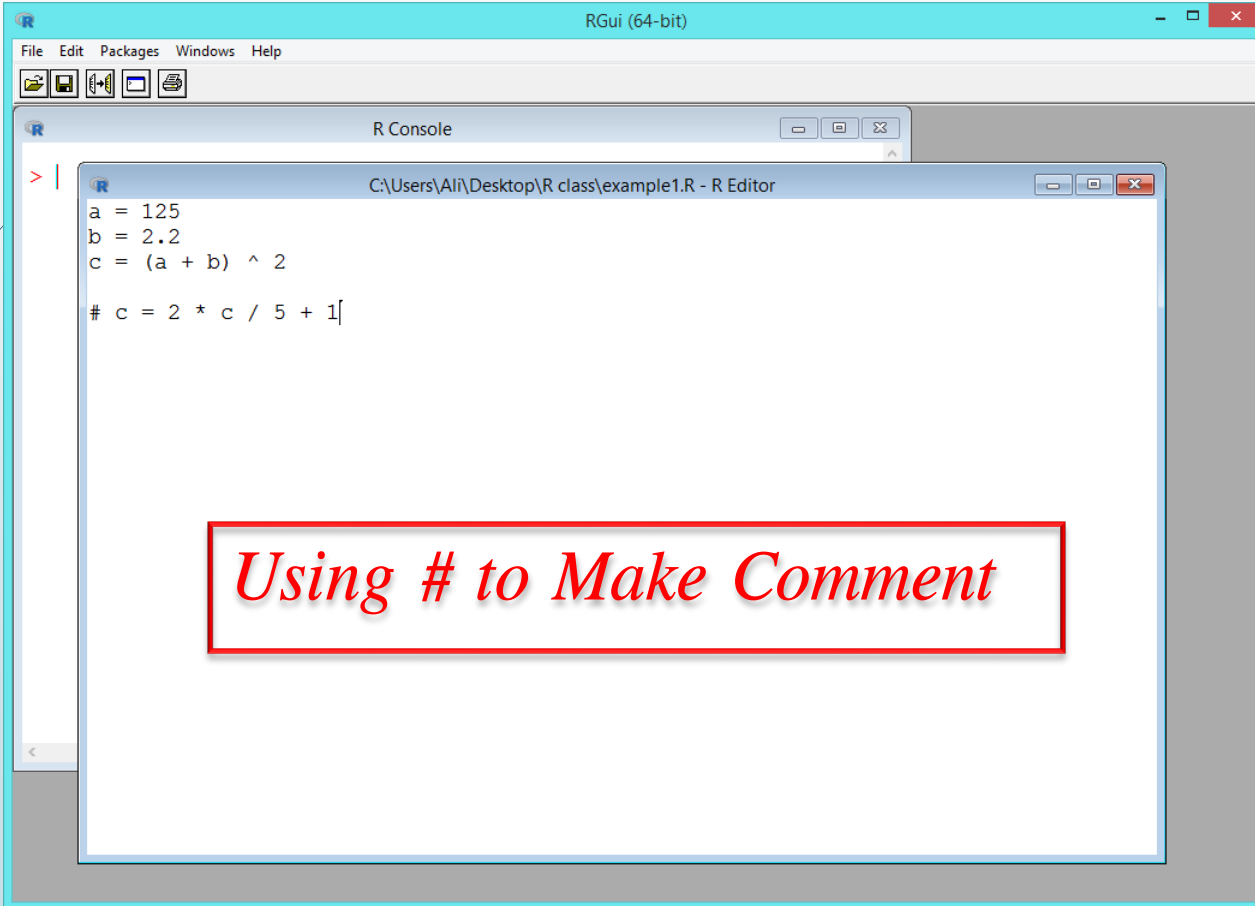
R Editor and Scripts

➤ Opening a Script (3)



Comment

► Making a Command Inactive (1)



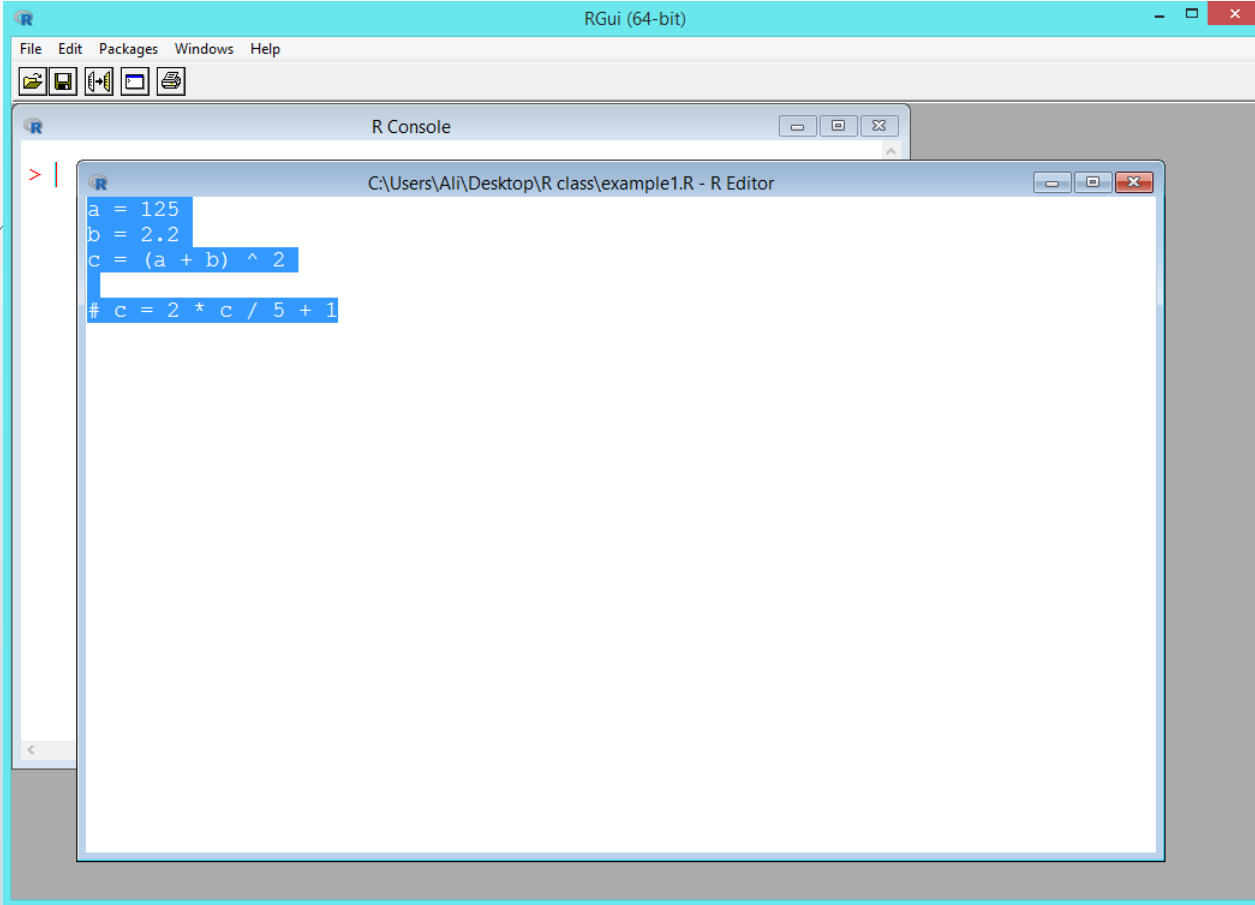
The screenshot shows the RGui (64-bit) window with a menu bar (File, Edit, Packages, Windows, Help) and a toolbar. The main window is titled "R Console" and contains the following R code:

```
> |  
a = 125  
b = 2.2  
c = (a + b) ^ 2  
# c = 2 * c / 5 + 1[
```

The code is displayed in a window titled "C:\Users\Ali\Desktop\R class\example1.R - R Editor". A red-bordered box highlights the text "Using # to Make Comment" in red, italicized font.

Comment

► Making a Command Inactive (2)

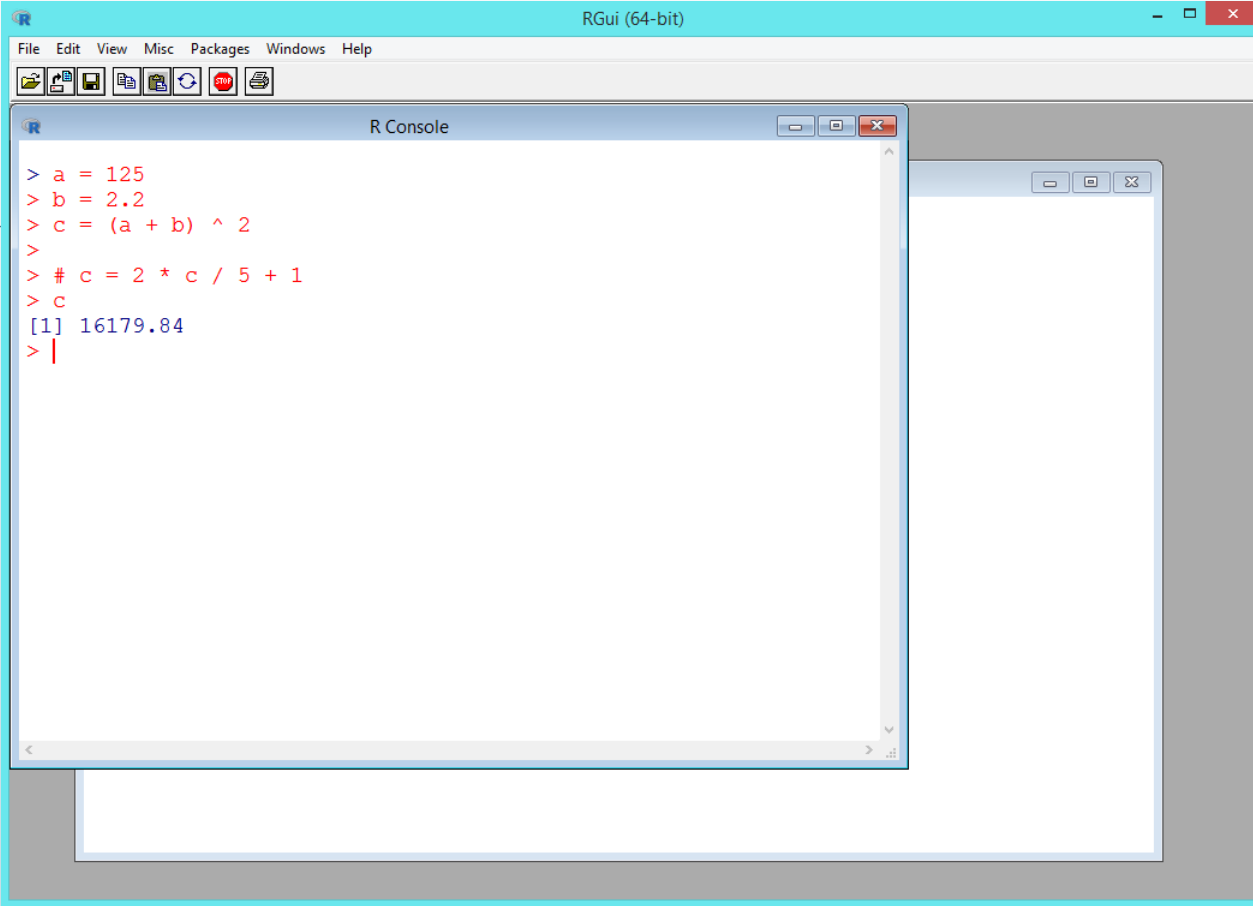


The screenshot displays the RGui (64-bit) window. The main window is titled "R Console" and shows the R prompt "> |". A smaller window titled "C:\Users\Ali\Desktop\R class\example1.R - R Editor" is open, showing the following R code:

```
a = 125  
b = 2.2  
c = (a + b) ^ 2  
# c = 2 * c / 5 + 1
```

Comment

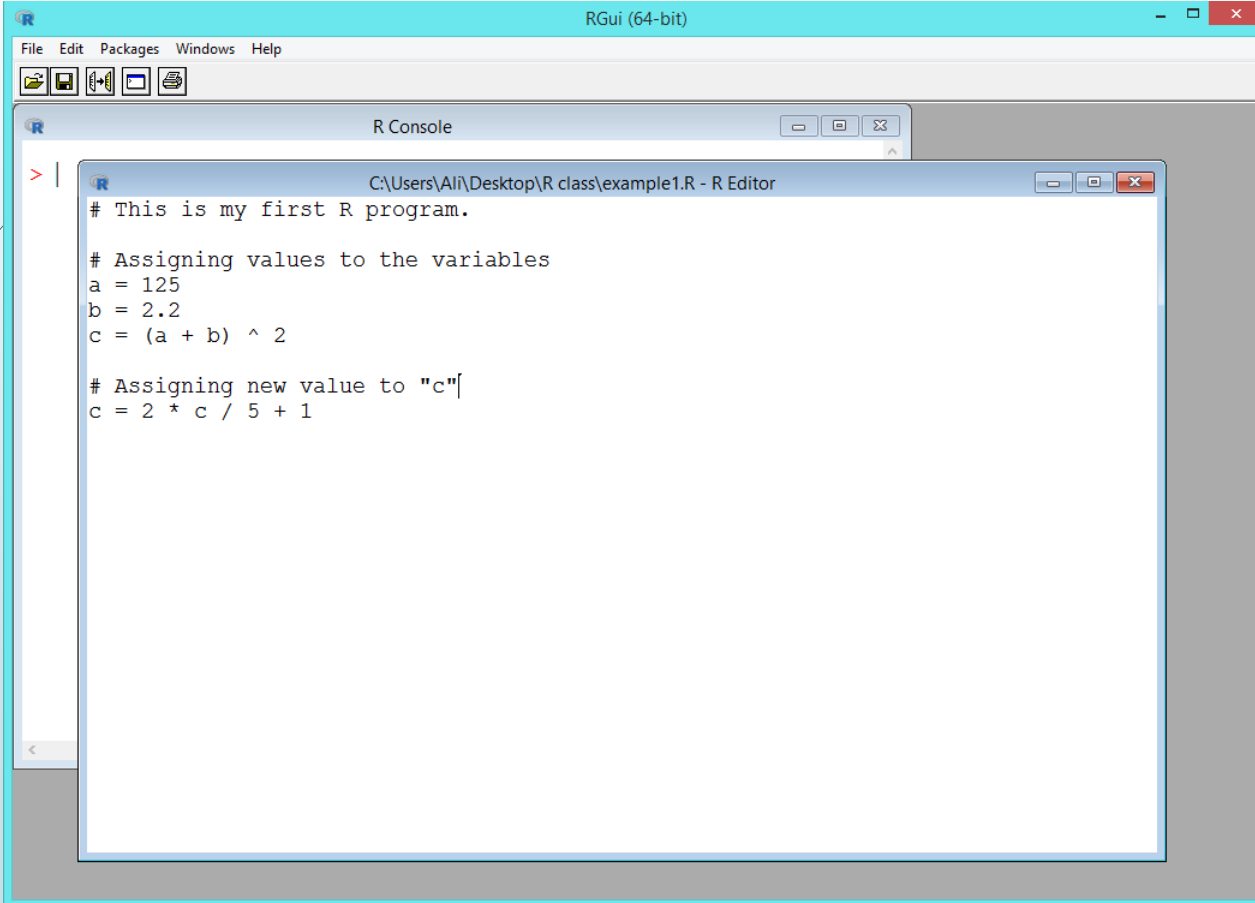
► Making a Command Inactive (3)



```
> a = 125
> b = 2.2
> c = (a + b) ^ 2
>
> # c = 2 * c / 5 + 1
> c
[1] 16179.84
> |
```

Comment

► Adding Descriptions and Explanations (1)

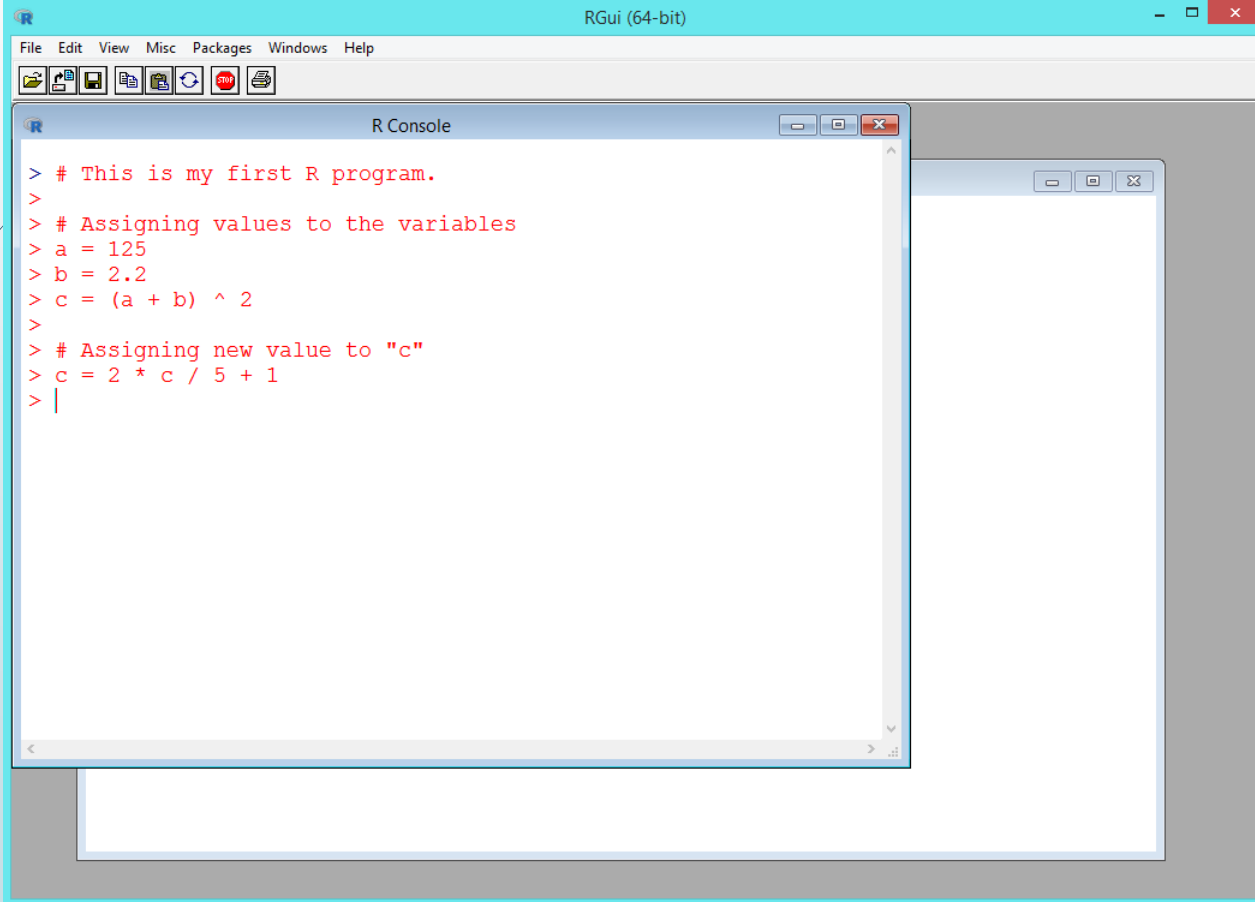


The screenshot displays the RGui (64-bit) window. The main window is titled "RGui (64-bit)" and has a menu bar with "File", "Edit", "Packages", "Windows", and "Help". Below the menu bar is a toolbar with icons for file operations. The main area shows an "R Console" window with a prompt "> |". Overlaid on this is an "R Editor" window titled "C:\Users\Ali\Desktop\R class\example1.R - R Editor". The editor contains the following R code:

```
# This is my first R program.  
  
# Assigning values to the variables  
a = 125  
b = 2.2  
c = (a + b) ^ 2  
  
# Assigning new value to "c"  
c = 2 * c / 5 + 1
```


Comment

► Adding Descriptions and Explanations (2)



The screenshot shows the RGui (64-bit) window with the R Console open. The console contains the following R code:

```
> # This is my first R program.  
>  
> # Assigning values to the variables  
> a = 125  
> b = 2.2  
> c = (a + b) ^ 2  
>  
> # Assigning new value to "c"  
> c = 2 * c / 5 + 1  
> |
```

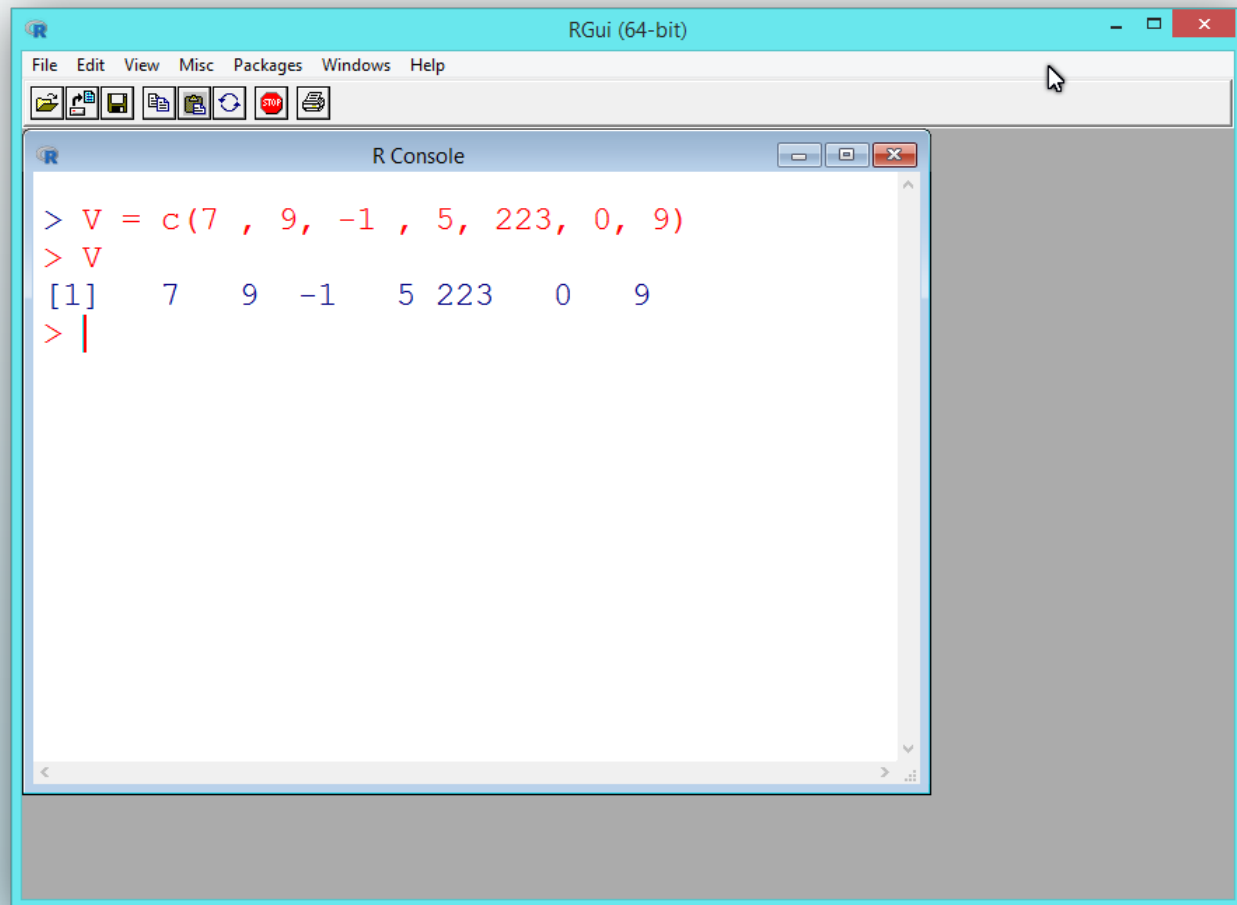


Data Structures

Vectors, Matrices, Data Frames, Arrays, Lists, Strings, Factors

Vector

► A Simple Vector

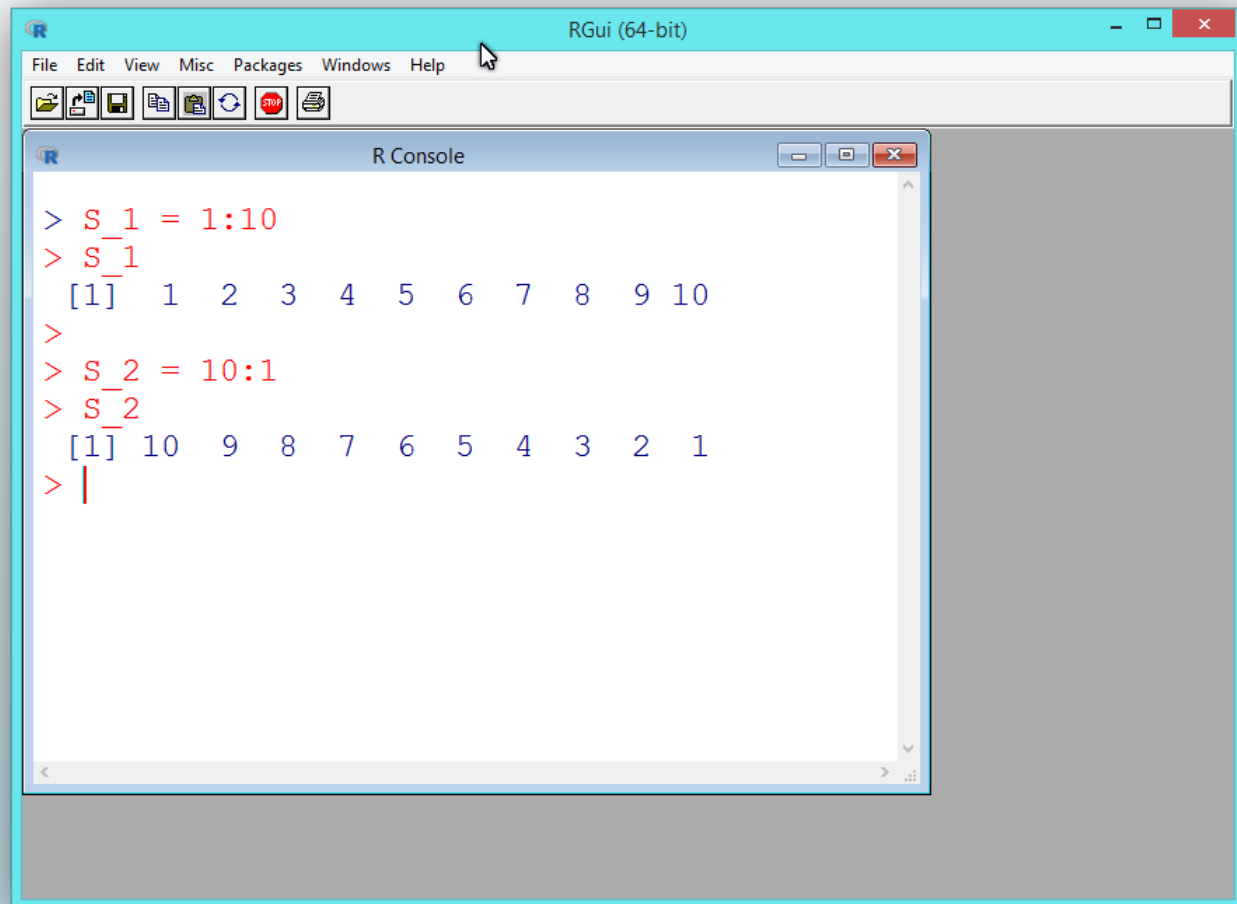


```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> V = c(7 , 9, -1 , 5, 223, 0, 9)
> V
[1] 7 9 -1 5 223 0 9
> |
```

Vector

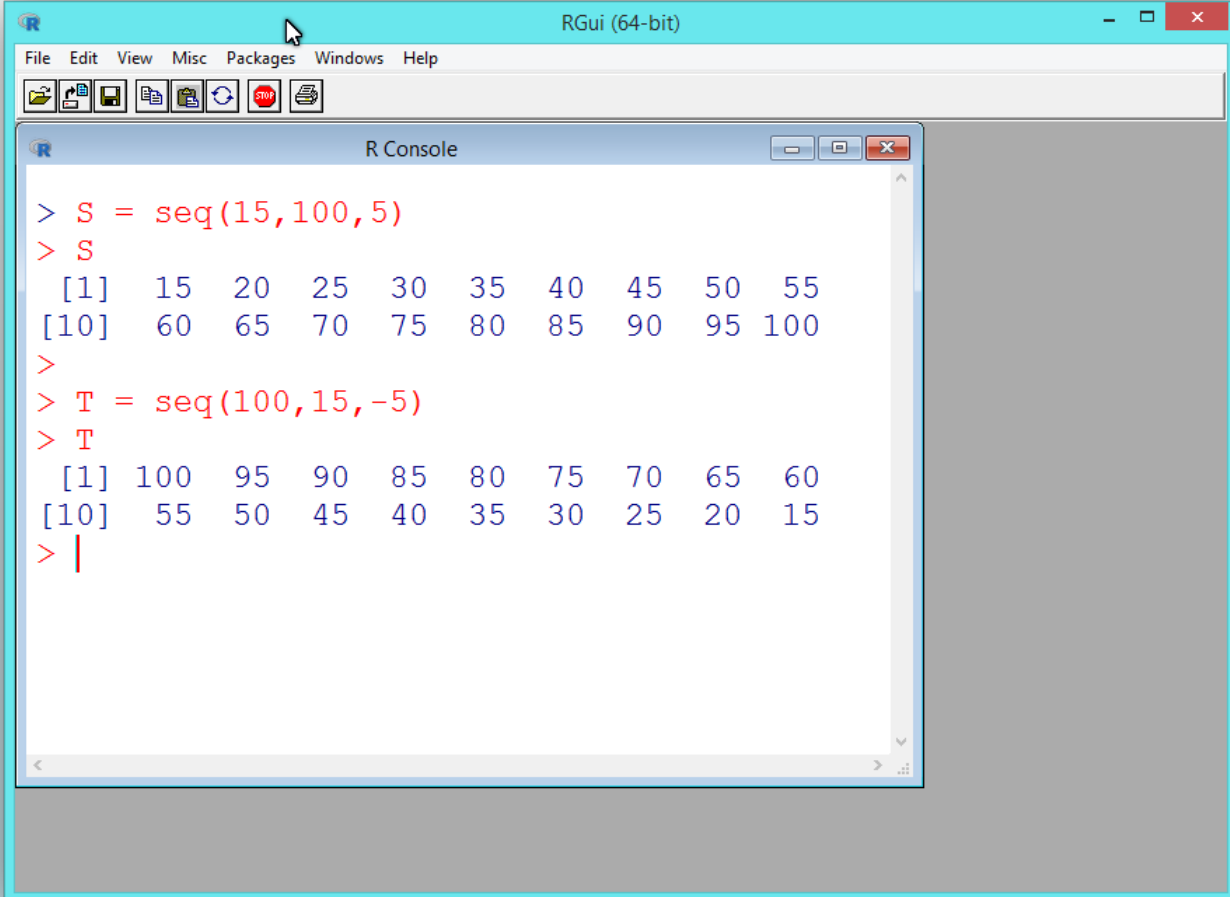
Simple Sequences



```
> S_1 = 1:10
> S_1
[1] 1 2 3 4 5 6 7 8 9 10
>
> S_2 = 10:1
> S_2
[1] 10 9 8 7 6 5 4 3 2 1
> |
```

Vector

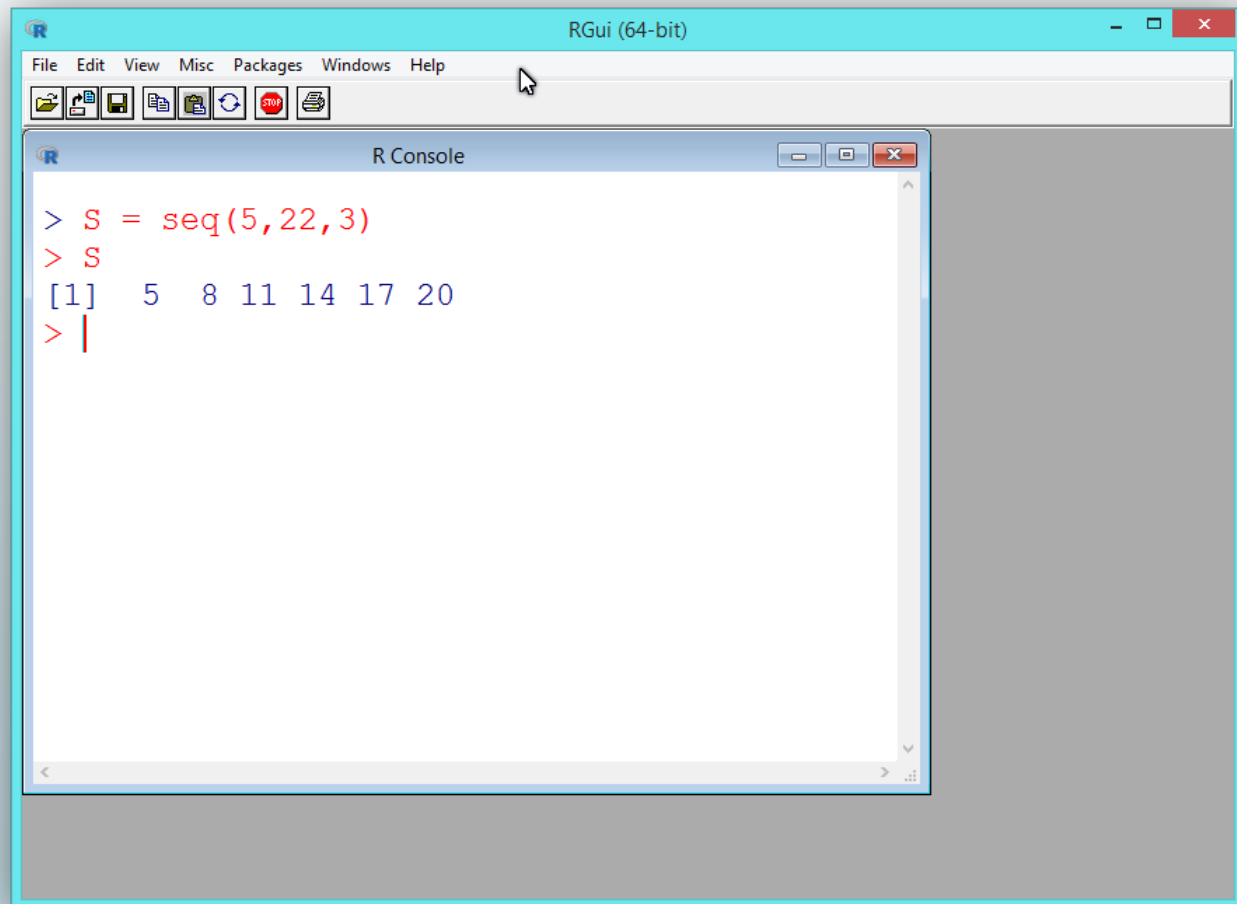
► Sequences (1)



```
> S = seq(15,100,5)
> S
 [1] 15 20 25 30 35 40 45 50 55
[10] 60 65 70 75 80 85 90 95 100
>
> T = seq(100,15,-5)
> T
 [1] 100 95 90 85 80 75 70 65 60
[10] 55 50 45 40 35 30 25 20 15
> |
```

Vector

► Sequences (2)

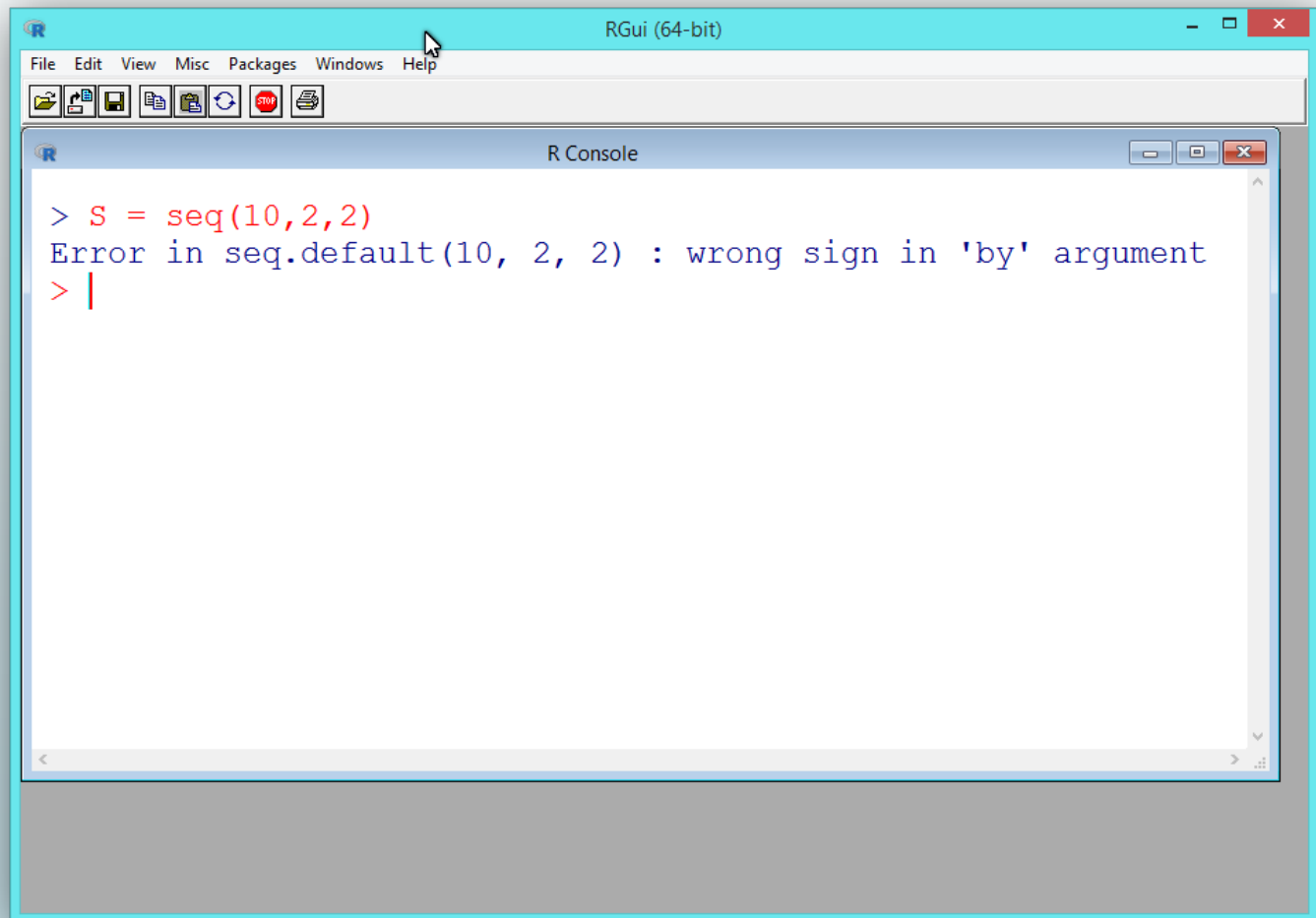


The screenshot shows the RGui (64-bit) window with the R Console open. The console displays the following R code and output:

```
> s = seq(5, 22, 3)
> s
[1] 5 8 11 14 17 20
> |
```

Vector

► Sequences (3)

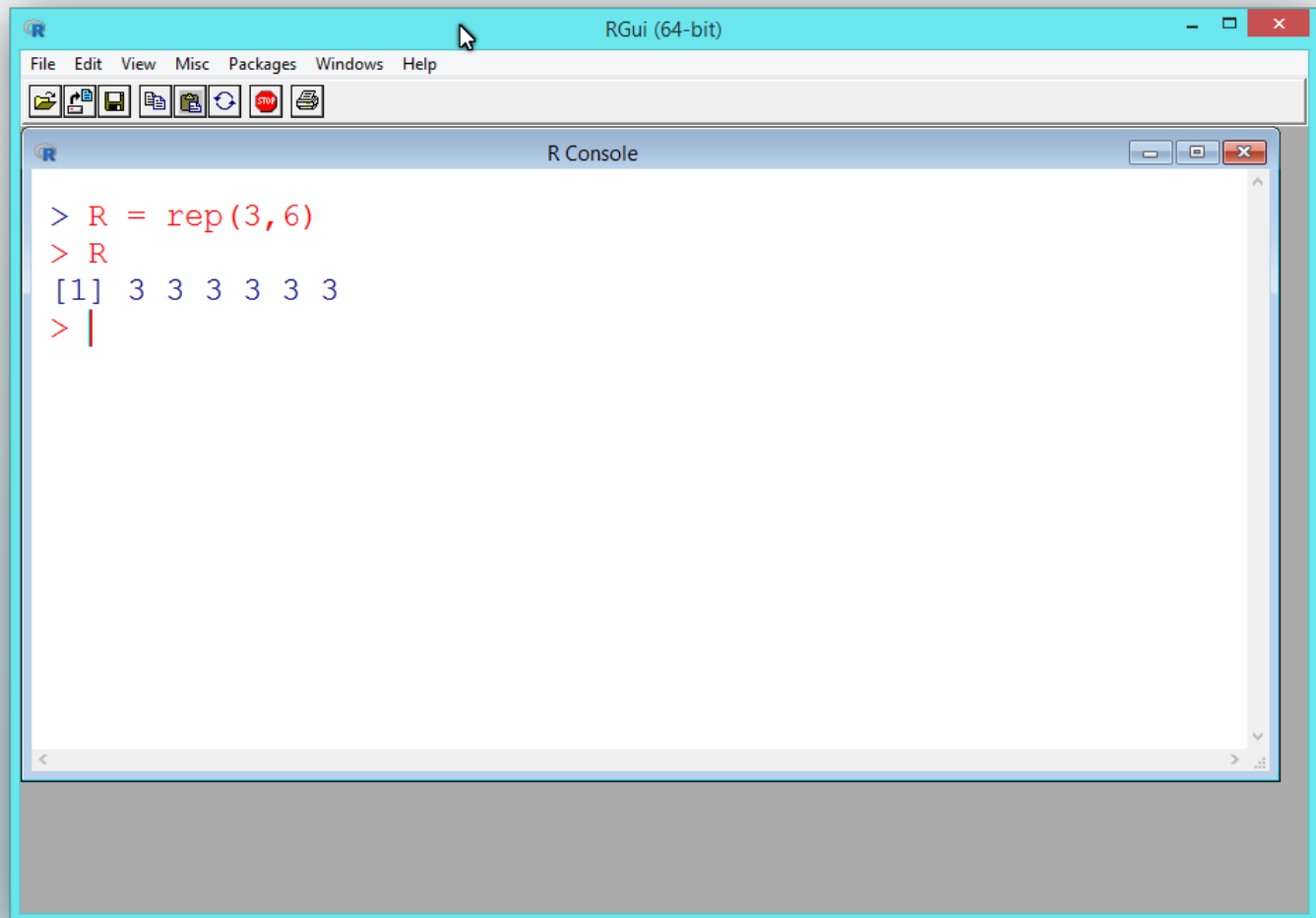


The screenshot shows the RGui (64-bit) interface. The R Console window displays the following text:

```
> S = seq(10,2,2)
Error in seq.default(10, 2, 2) : wrong sign in 'by' argument
> |
```

Vector

► Sequences with Repeated Elements (1)

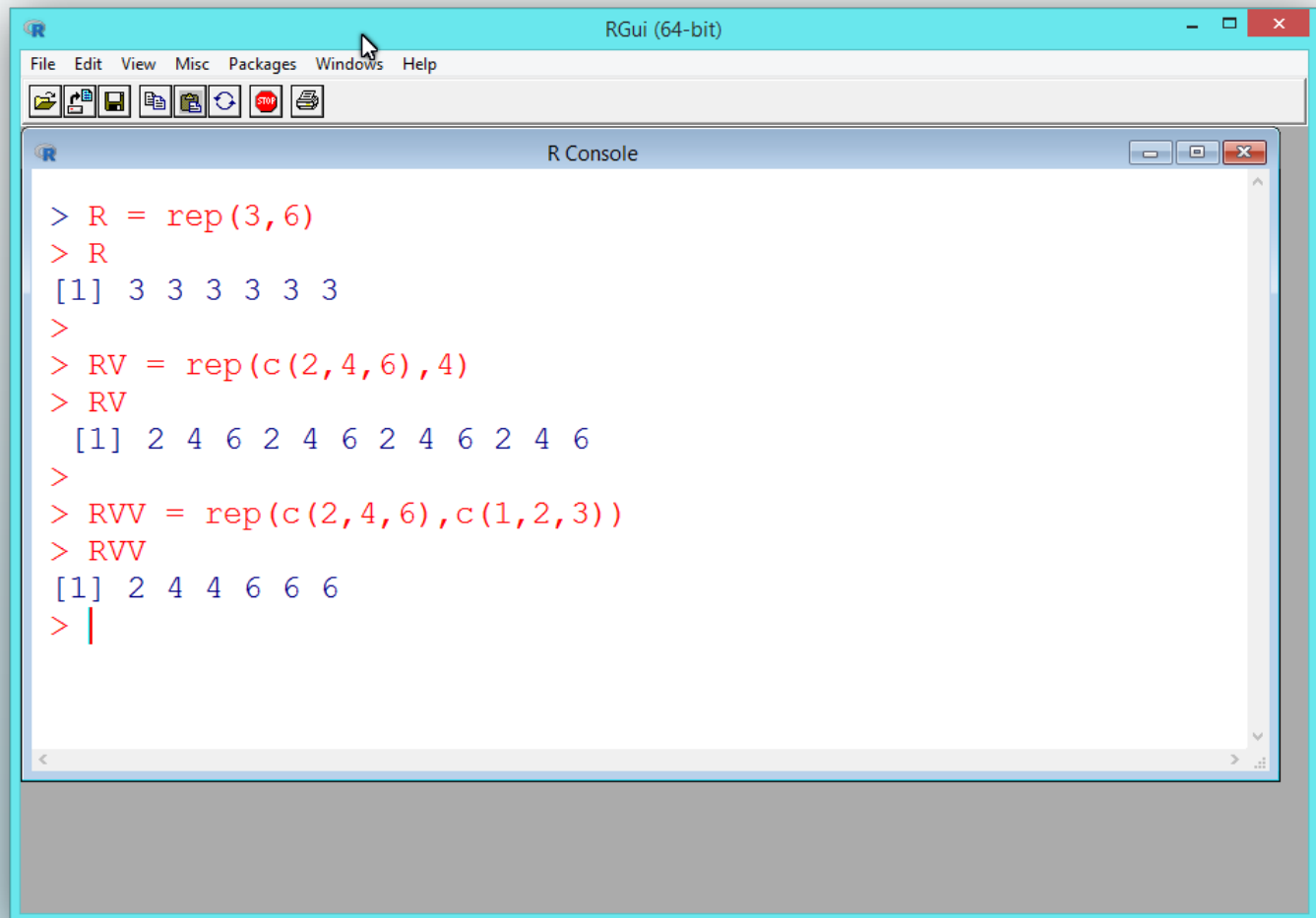


```
> R = rep(3, 6)
> R
[1] 3 3 3 3 3 3
> |
```

The screenshot shows the RGui (64-bit) interface. The R Console window displays the following code and output:

Vector

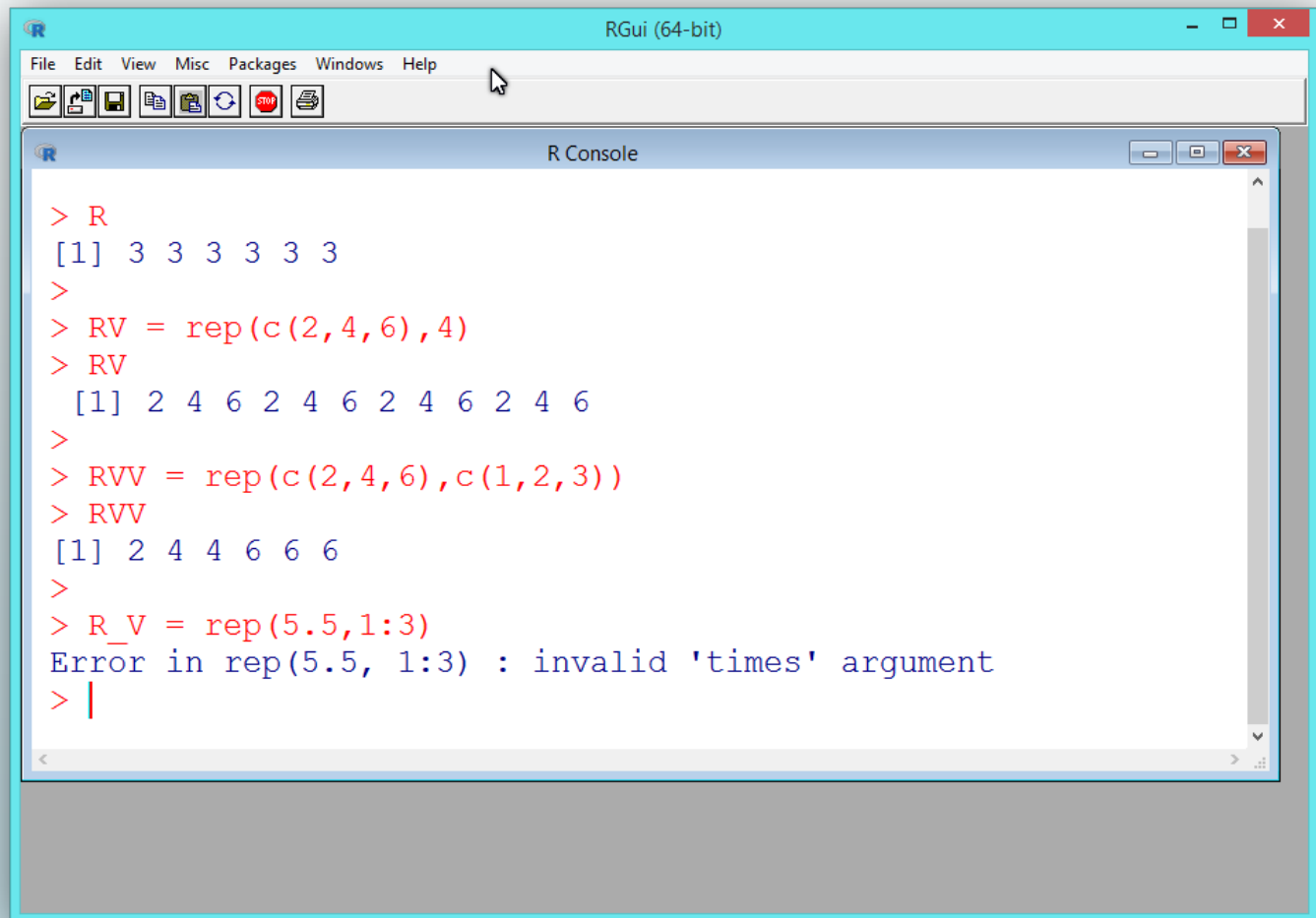
► Sequences with Repeated Elements (2)



```
> R = rep(3,6)
> R
[1] 3 3 3 3 3 3
>
> RV = rep(c(2,4,6),4)
> RV
[1] 2 4 6 2 4 6 2 4 6 2 4 6
>
> RVV = rep(c(2,4,6),c(1,2,3))
> RVV
[1] 2 4 4 6 6 6
> |
```

Vector

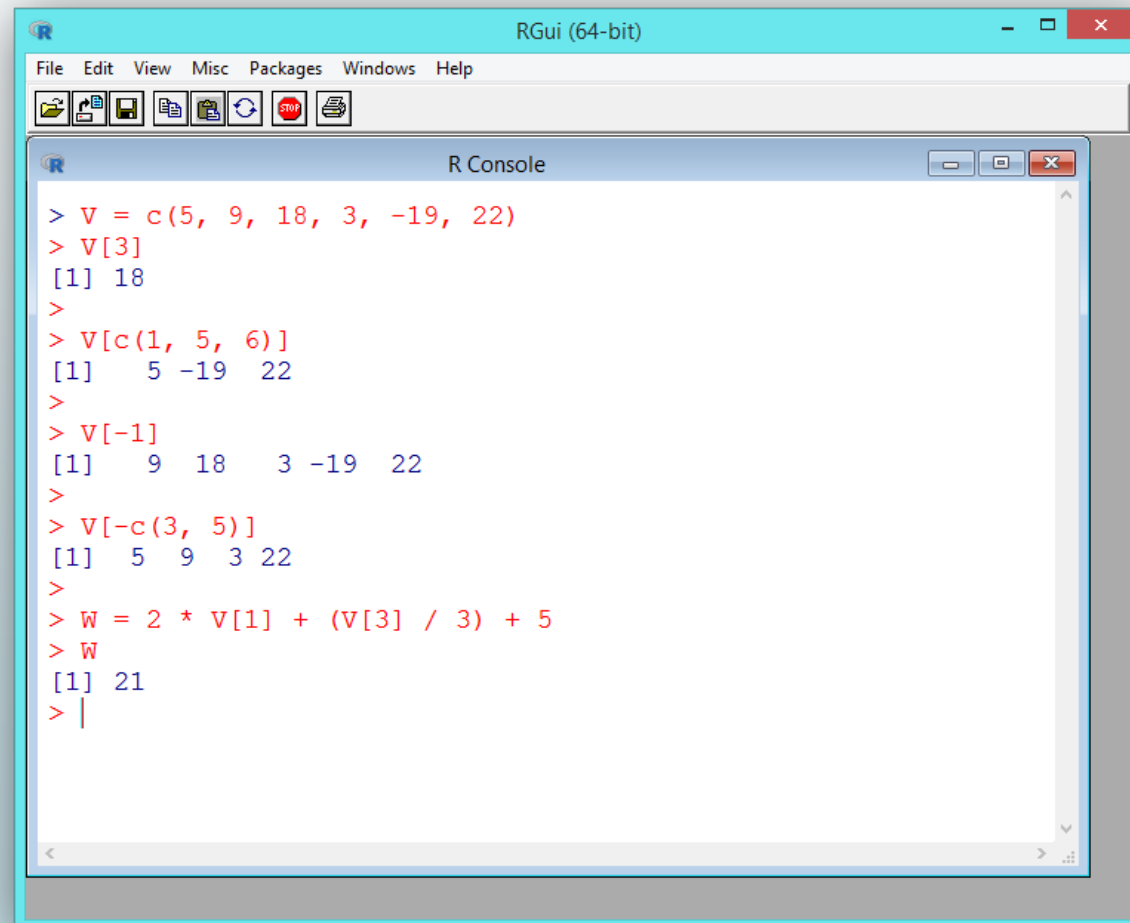
► Sequences with Repeated Elements (3)



```
> R
[1] 3 3 3 3 3 3
>
> RV = rep(c(2,4,6),4)
> RV
[1] 2 4 6 2 4 6 2 4 6 2 4 6
>
> RVV = rep(c(2,4,6),c(1,2,3))
> RVV
[1] 2 4 4 6 6 6
>
> R_V = rep(5.5,1:3)
Error in rep(5.5, 1:3) : invalid 'times' argument
> |
```

Vector

➤ Addressing Vector Elements

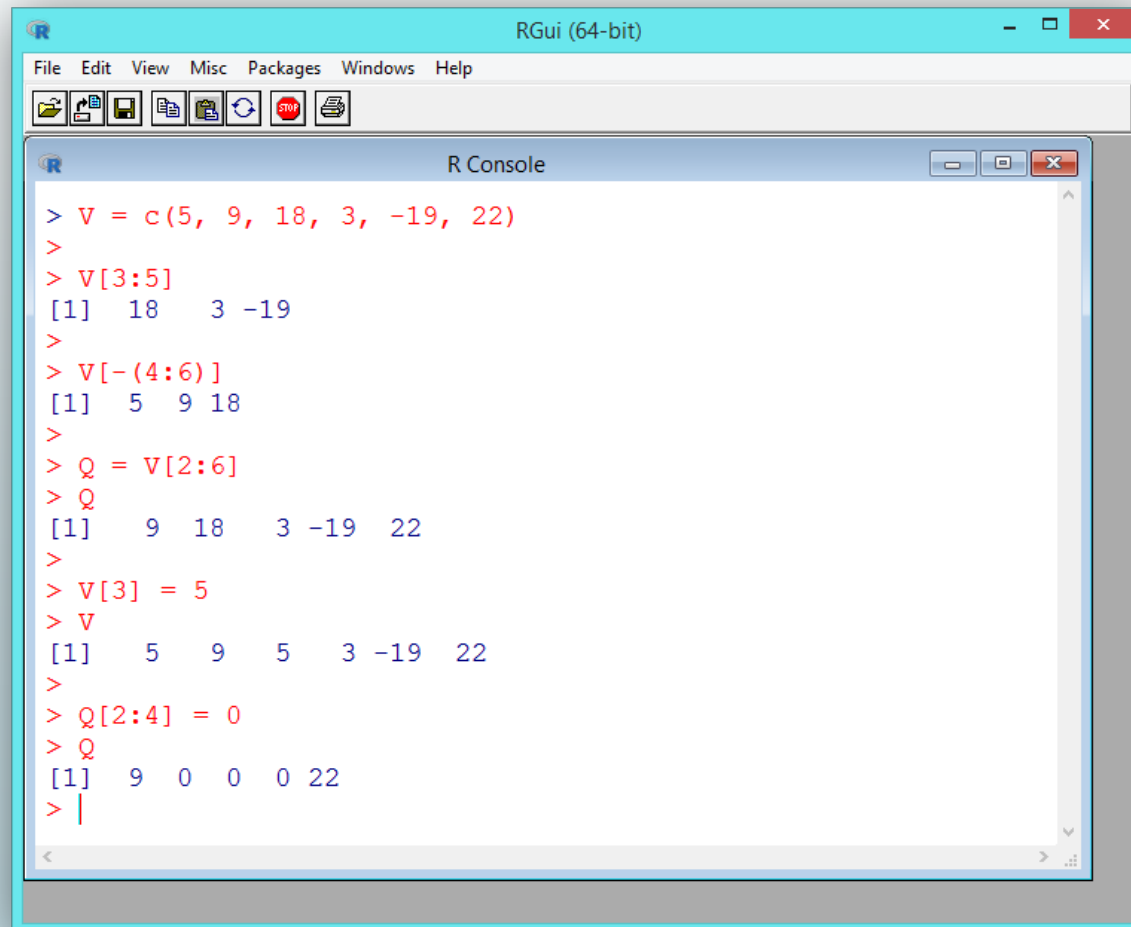


```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> V = c(5, 9, 18, 3, -19, 22)
> V[3]
[1] 18
>
> V[c(1, 5, 6)]
[1] 5 -19 22
>
> V[-1]
[1] 9 18 3 -19 22
>
> V[-c(3, 5)]
[1] 5 9 3 22
>
> W = 2 * V[1] + (V[3] / 3) + 5
> W
[1] 21
> |
```

Vector

► Vector Slice

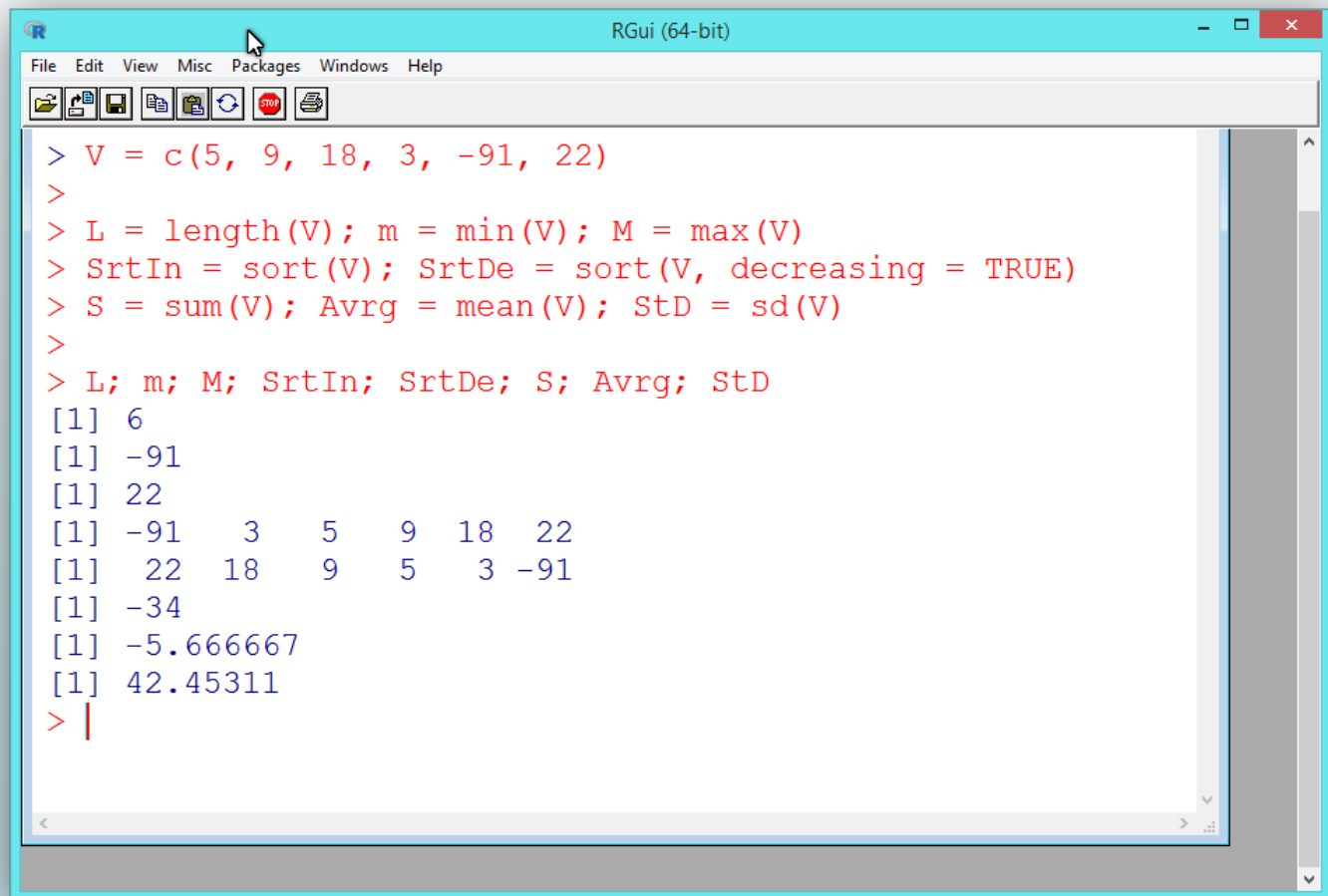


```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> v = c(5, 9, 18, 3, -19, 22)
>
> v[3:5]
[1] 18 3 -19
>
> v[-(4:6)]
[1] 5 9 18
>
> Q = v[2:6]
> Q
[1] 9 18 3 -19 22
>
> v[3] = 5
> v
[1] 5 9 5 3 -19 22
>
> Q[2:4] = 0
> Q
[1] 9 0 0 0 22
> |
```

Vector

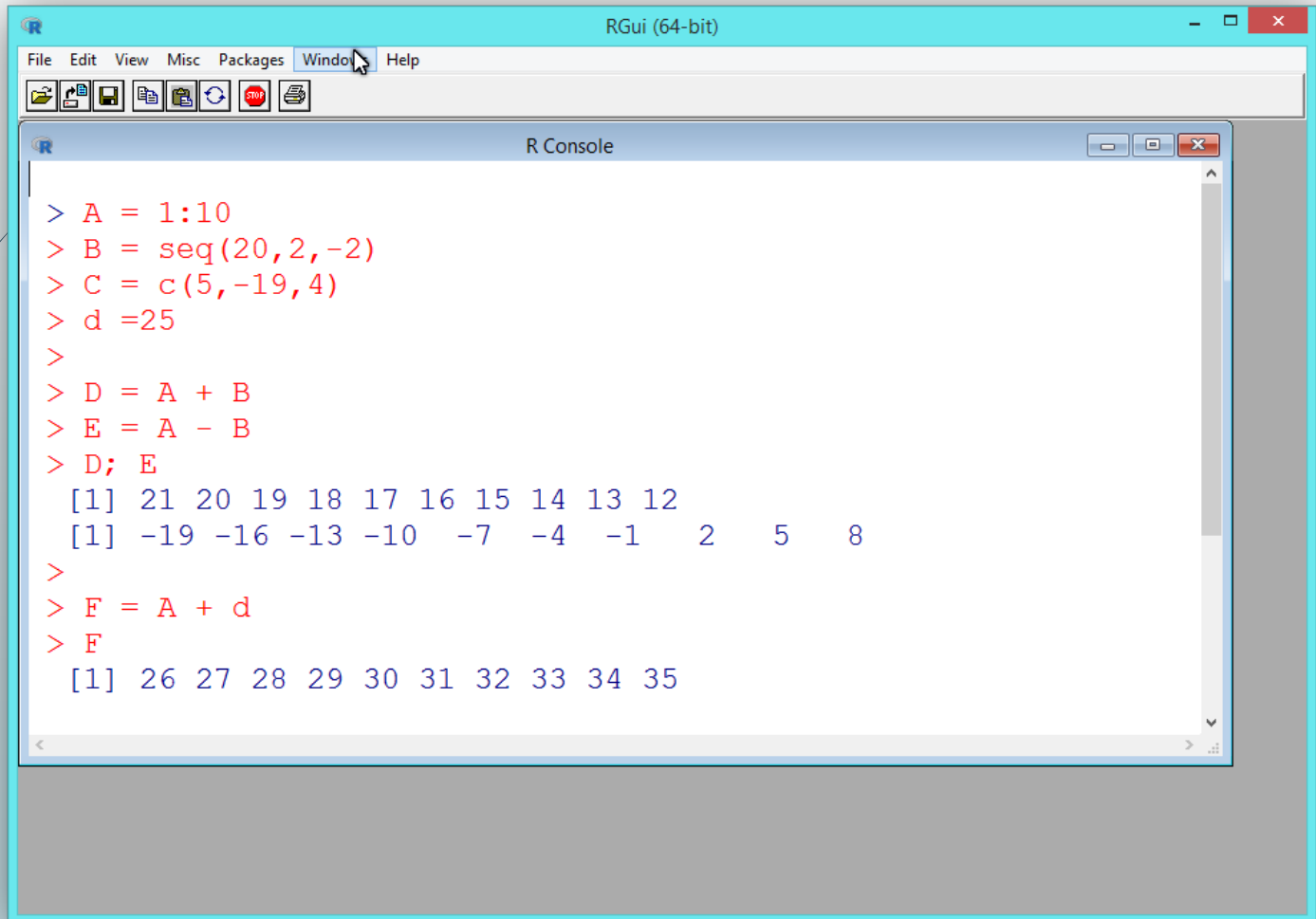
► Some Built-in Functions for Vectors



```
> V = c(5, 9, 18, 3, -91, 22)
>
> L = length(V); m = min(V); M = max(V)
> SrtIn = sort(V); SrtDe = sort(V, decreasing = TRUE)
> S = sum(V); Avrg = mean(V); StD = sd(V)
>
> L; m; M; SrtIn; SrtDe; S; Avrg; StD
[1] 6
[1] -91
[1] 22
[1] -91  3  5  9 18 22
[1] 22 18 9 5 3 -91
[1] -34
[1] -5.666667
[1] 42.45311
> |
```

Vector

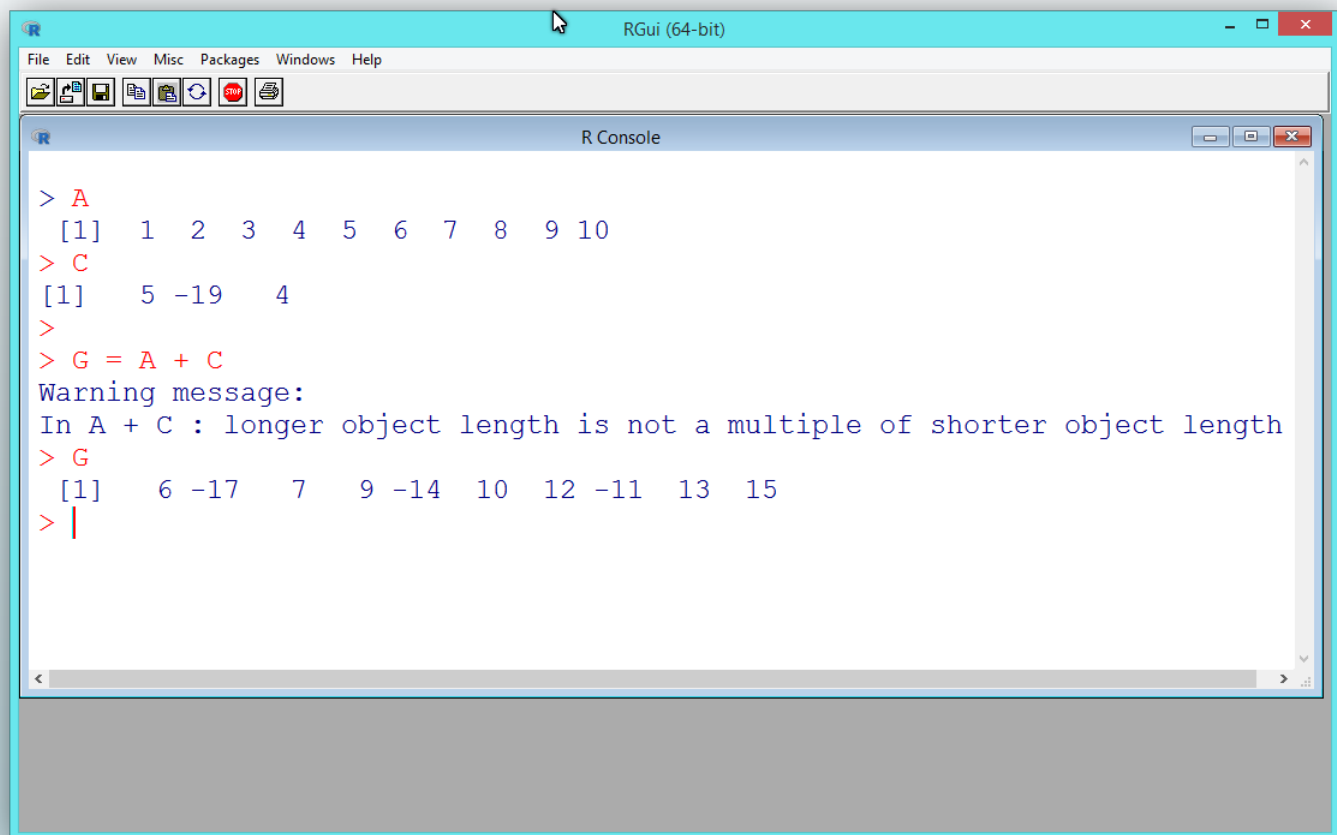
► Some Mathematical Operations (1)



```
RGui (64-bit)
File Edit View Misc Packages Window Help
[Icons]
R Console
> A = 1:10
> B = seq(20,2,-2)
> C = c(5,-19,4)
> d =25
>
> D = A + B
> E = A - B
> D; E
[1] 21 20 19 18 17 16 15 14 13 12
[1] -19 -16 -13 -10 -7 -4 -1 2 5 8
>
> F = A + d
> F
[1] 26 27 28 29 30 31 32 33 34 35
```

Vector

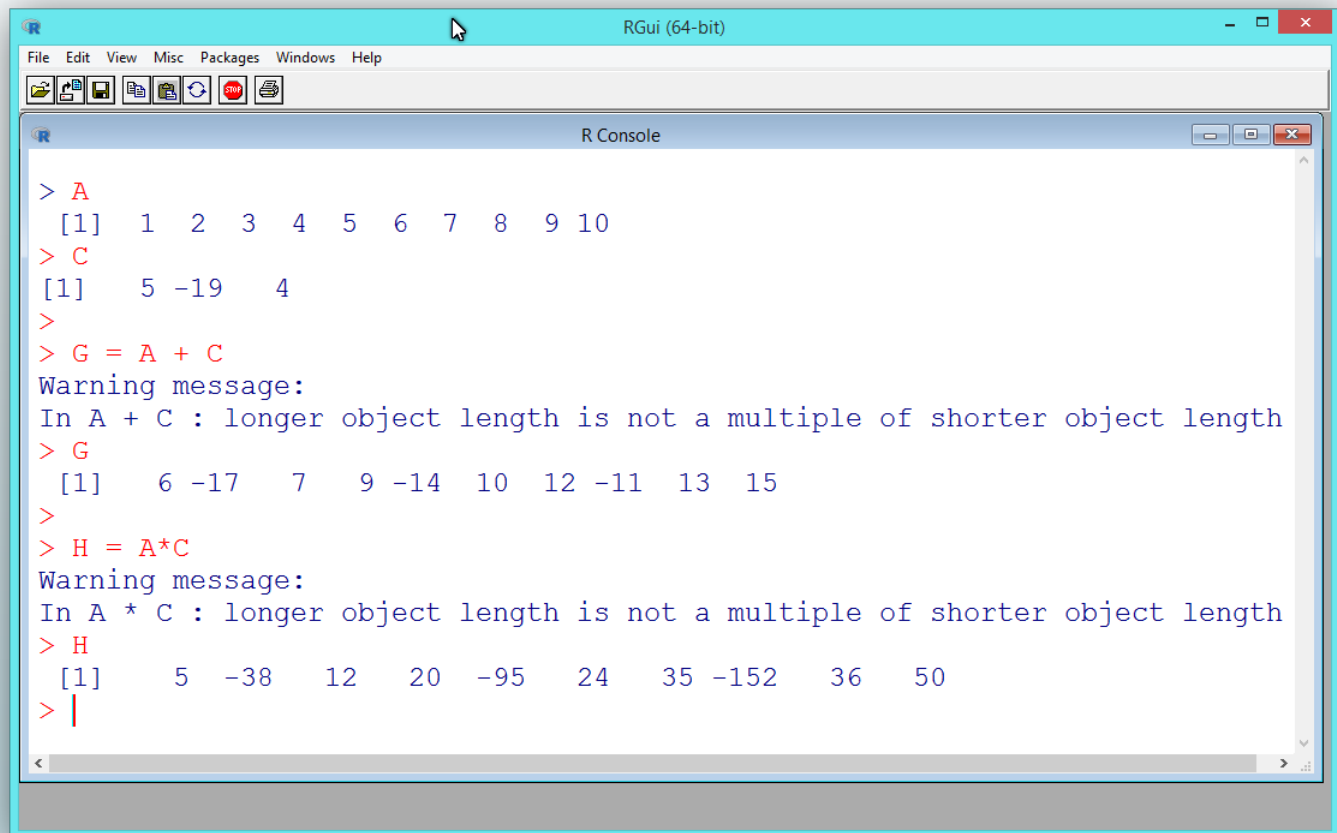
► Some Mathematical Operations (2)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> A
[1] 1 2 3 4 5 6 7 8 9 10
> C
[1] 5 -19 4
>
> G = A + C
Warning message:
In A + C : longer object length is not a multiple of shorter object length
> G
[1] 6 -17 7 9 -14 10 12 -11 13 15
> |
```

Vector

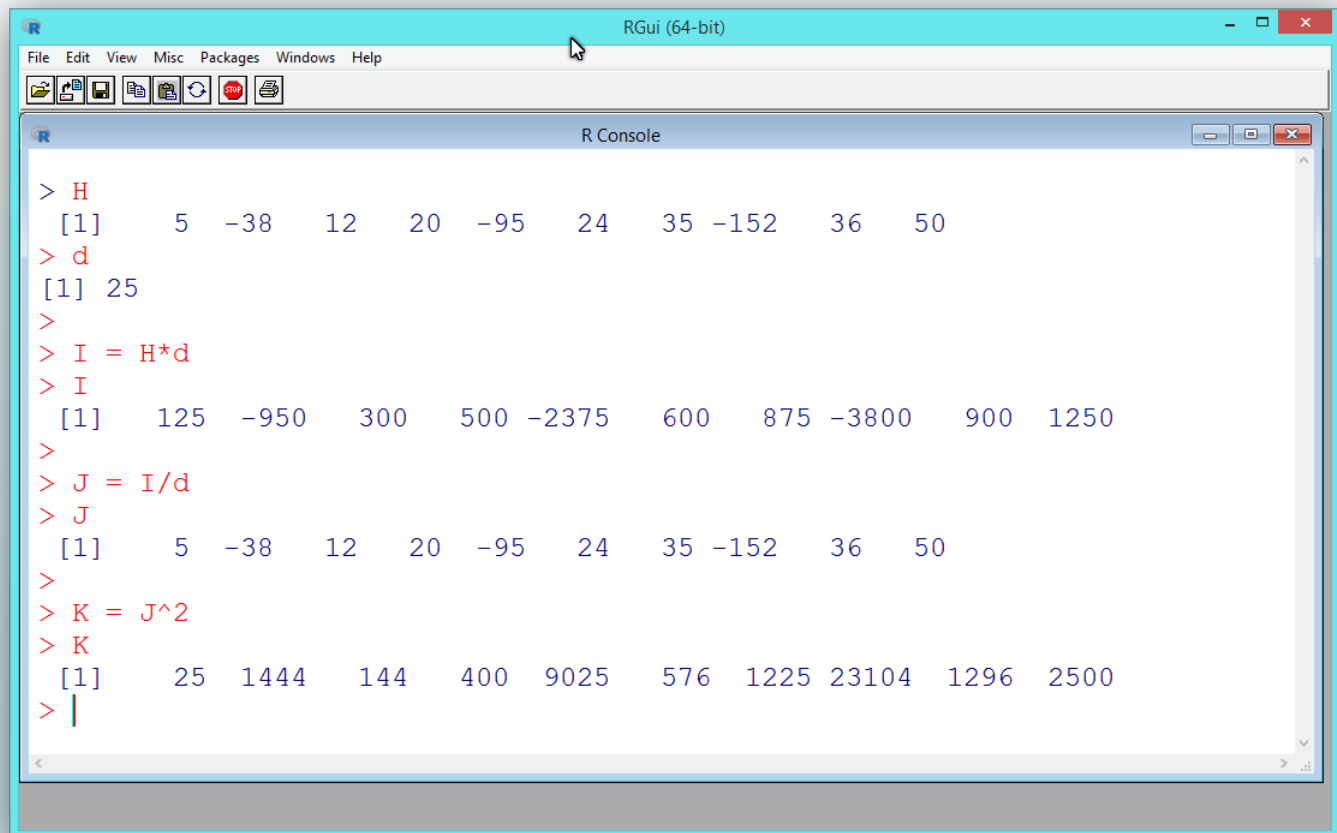
► Some Mathematical Operations (3)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> A
[1] 1 2 3 4 5 6 7 8 9 10
> C
[1] 5 -19 4
>
> G = A + C
Warning message:
In A + C : longer object length is not a multiple of shorter object length
> G
[1] 6 -17 7 9 -14 10 12 -11 13 15
>
> H = A * C
Warning message:
In A * C : longer object length is not a multiple of shorter object length
> H
[1] 5 -38 12 20 -95 24 35 -152 36 50
> |
```


Vector

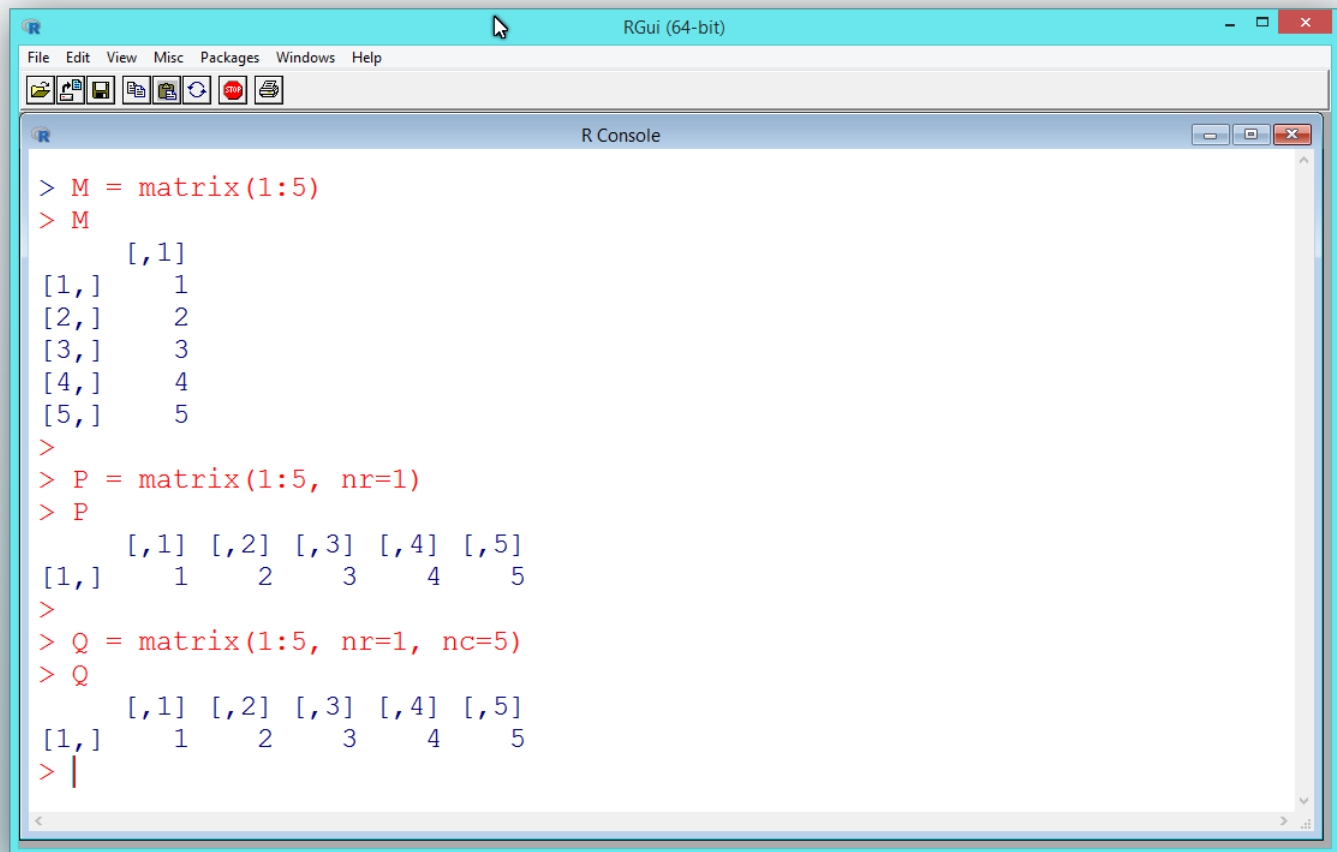
► Some Mathematical Operations (4)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> H
[1] 5 -38 12 20 -95 24 35 -152 36 50
> d
[1] 25
>
> I = H*d
> I
[1] 125 -950 300 500 -2375 600 875 -3800 900 1250
>
> J = I/d
> J
[1] 5 -38 12 20 -95 24 35 -152 36 50
>
> K = J^2
> K
[1] 25 1444 144 400 9025 576 1225 23104 1296 2500
> |
```

Matrix

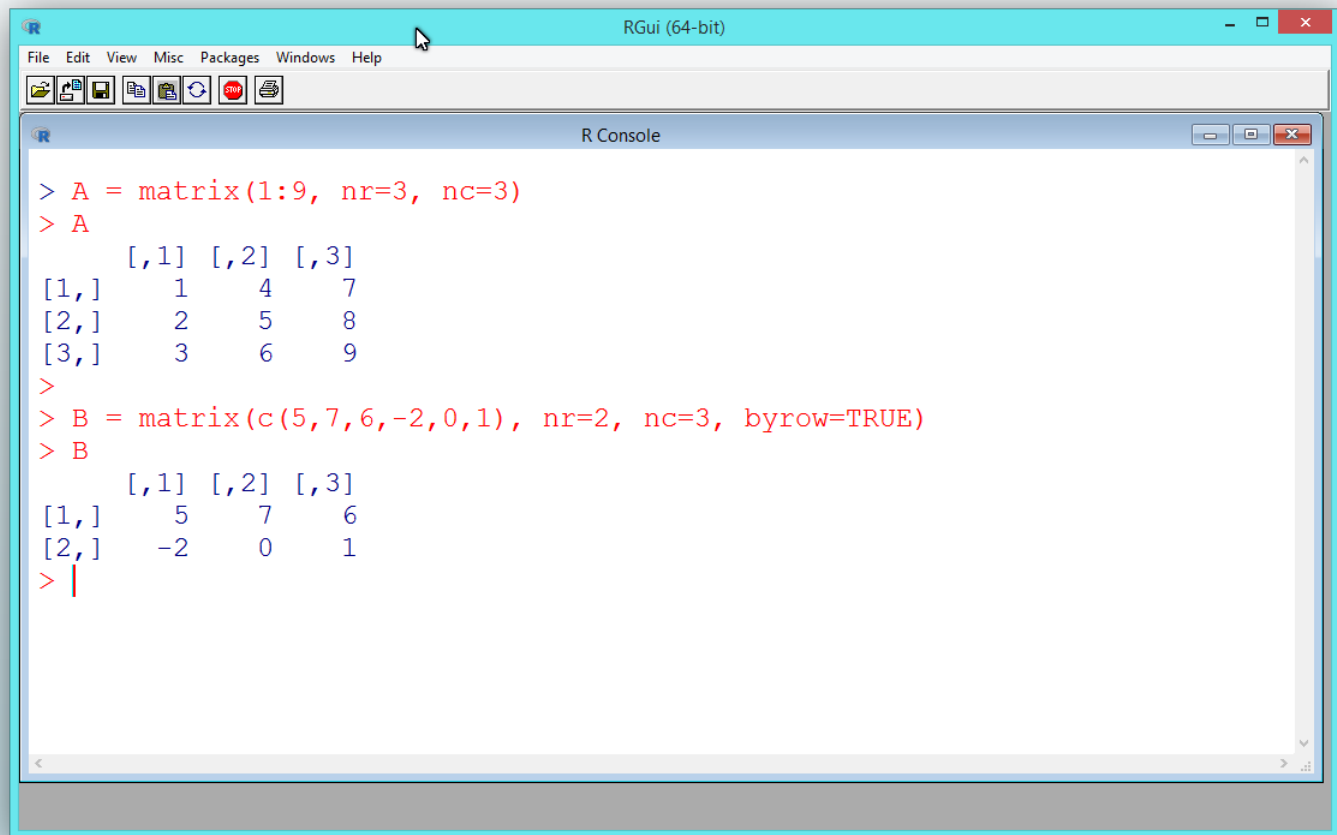
► Assignment and Arguments (1)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> M = matrix(1:5)
> M
      [,1]
 [1,]    1
 [2,]    2
 [3,]    3
 [4,]    4
 [5,]    5
>
> P = matrix(1:5, nr=1)
> P
      [,1] [,2] [,3] [,4] [,5]
 [1,]    1    2    3    4    5
>
> Q = matrix(1:5, nr=1, nc=5)
> Q
      [,1] [,2] [,3] [,4] [,5]
 [1,]    1    2    3    4    5
> |
```

Matrix

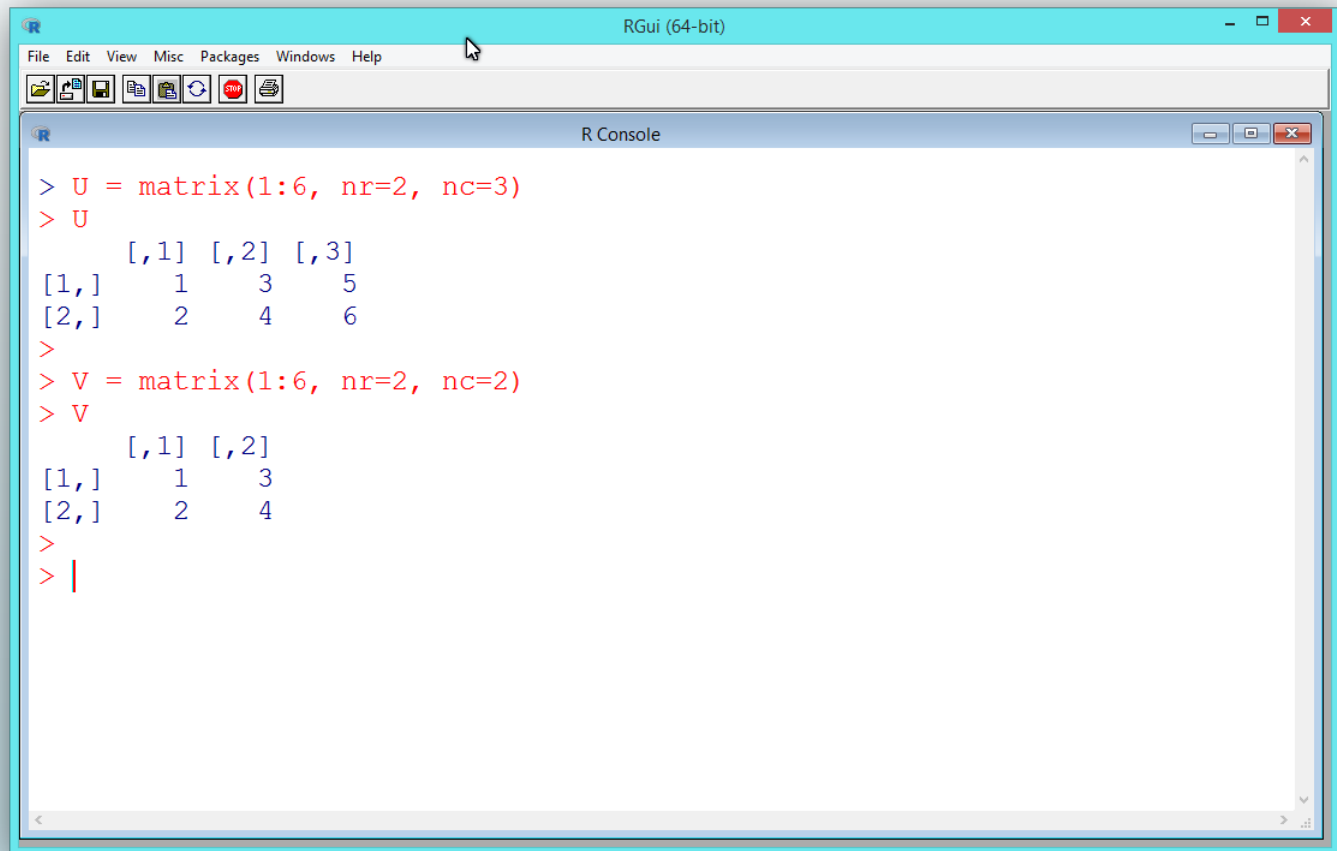
► Assignment and Arguments (2)



```
> A = matrix(1:9, nr=3, nc=3)
> A
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
>
> B = matrix(c(5,7,6,-2,0,1), nr=2, nc=3, byrow=TRUE)
> B
      [,1] [,2] [,3]
[1,]    5    7    6
[2,]   -2    0    1
> |
```

Matrix

► Assignment and Arguments (3)

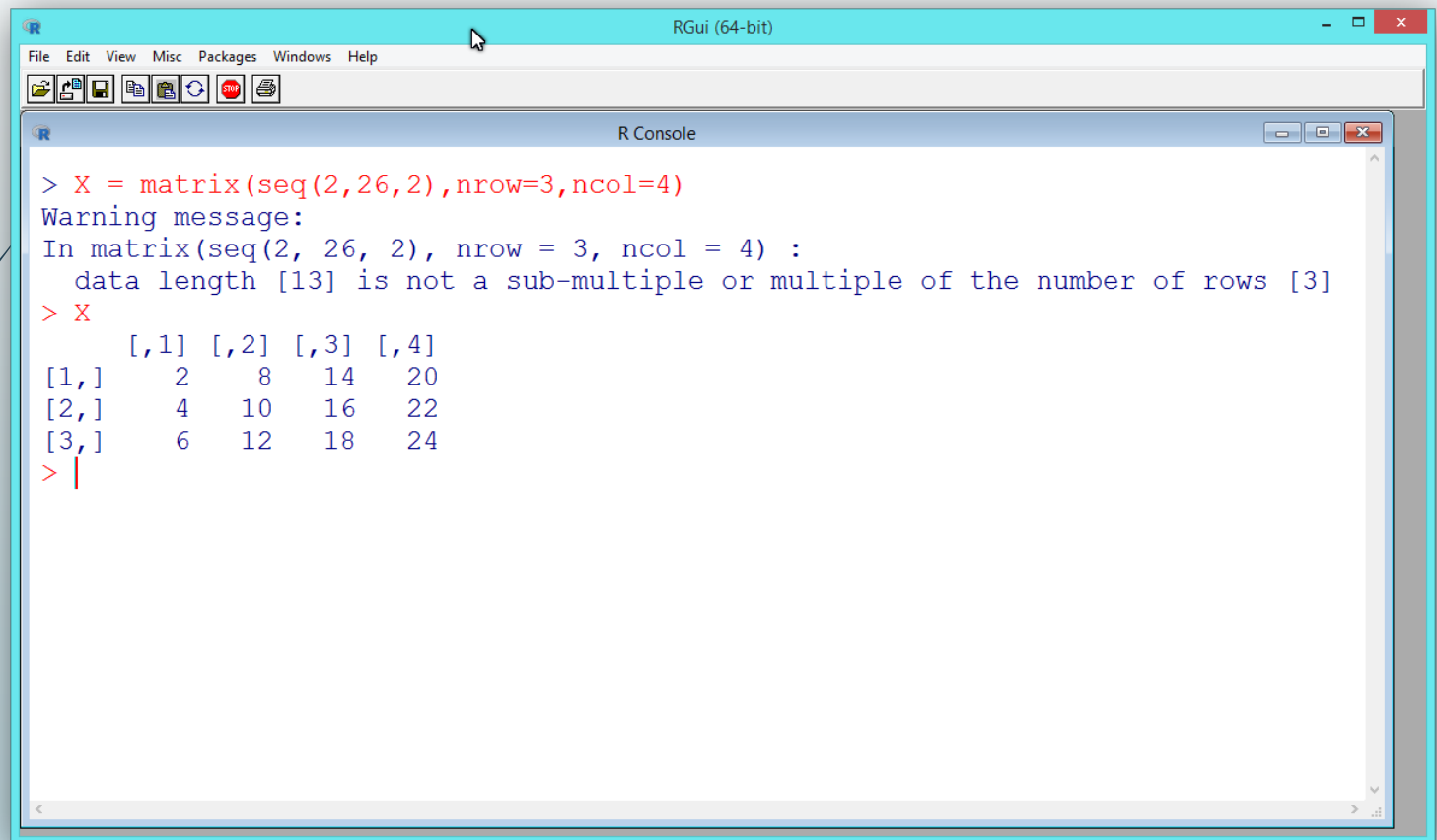


```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> U = matrix(1:6, nr=2, nc=3)
> U
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
>
> V = matrix(1:6, nr=2, nc=2)
> V
      [,1] [,2]
[1,]    1    3
[2,]    2    4
>
> |
```

Matrix

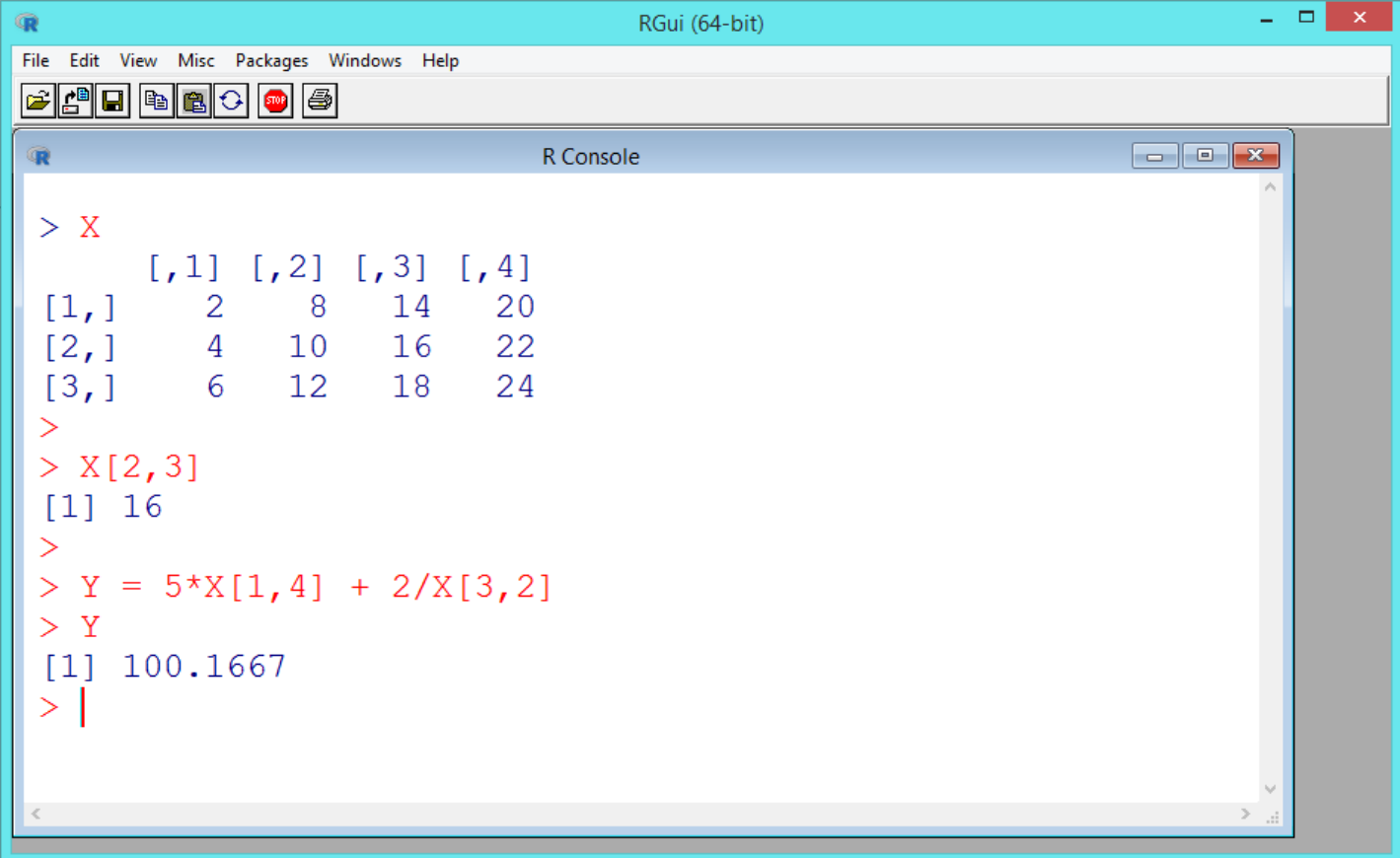
► Assignment and Arguments (4)



```
> X = matrix(seq(2,26,2),nrow=3,ncol=4)
Warning message:
In matrix(seq(2, 26, 2), nrow = 3, ncol = 4) :
  data length [13] is not a sub-multiple or multiple of the number of rows [3]
> X
     [,1] [,2] [,3] [,4]
[1,]    2    8   14   20
[2,]    4   10   16   22
[3,]    6   12   18   24
> |
```

Matrix

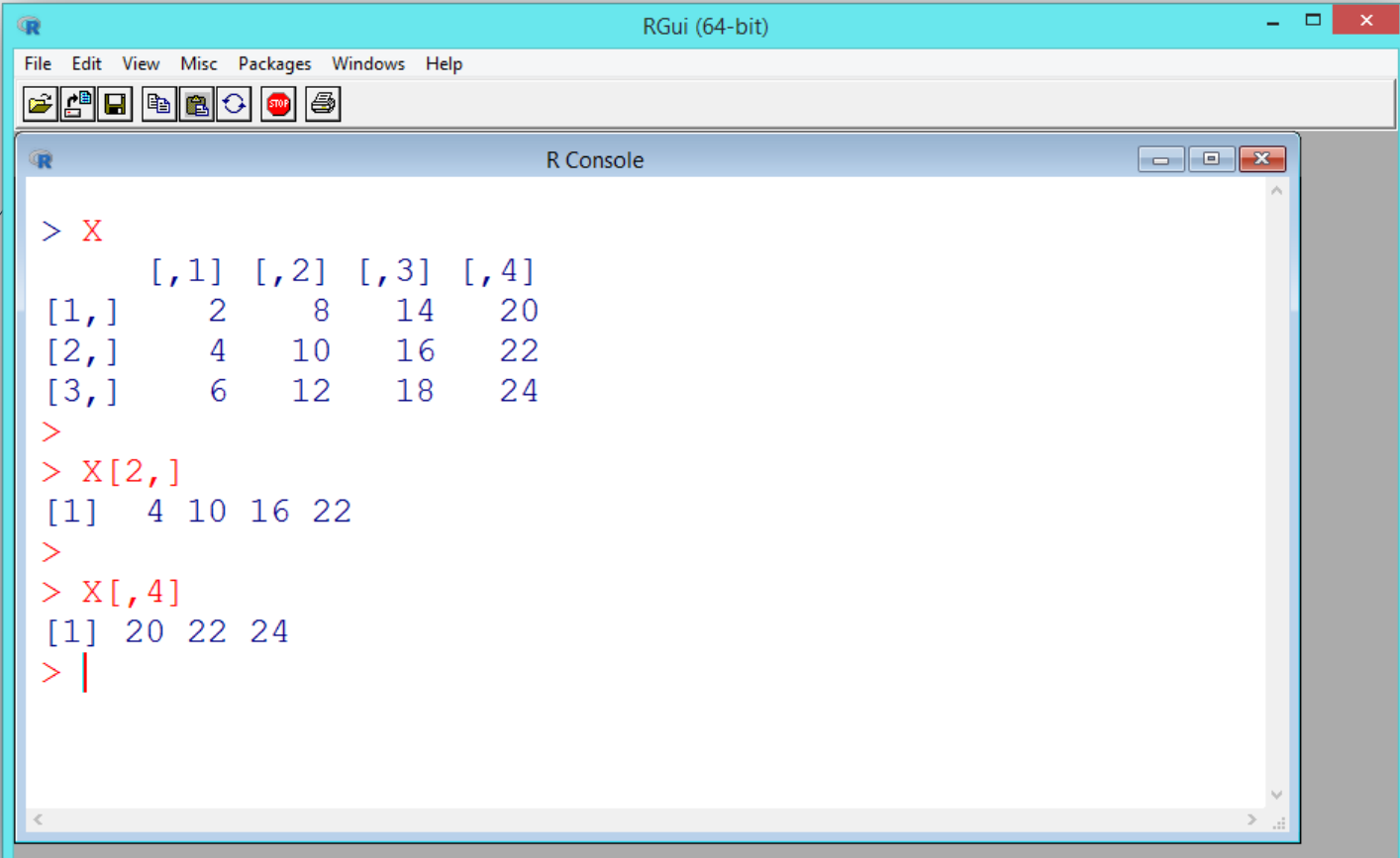
➤ Addressing Matrix Elements



```
> X
      [,1] [,2] [,3] [,4]
[1,]    2    8   14   20
[2,]    4   10   16   22
[3,]    6   12   18   24
>
> X[2,3]
[1] 16
>
> Y = 5*X[1,4] + 2/X[3,2]
> Y
[1] 100.1667
> |
```

Matrix

➤ Addressing Matrix Rows and Columns (1)

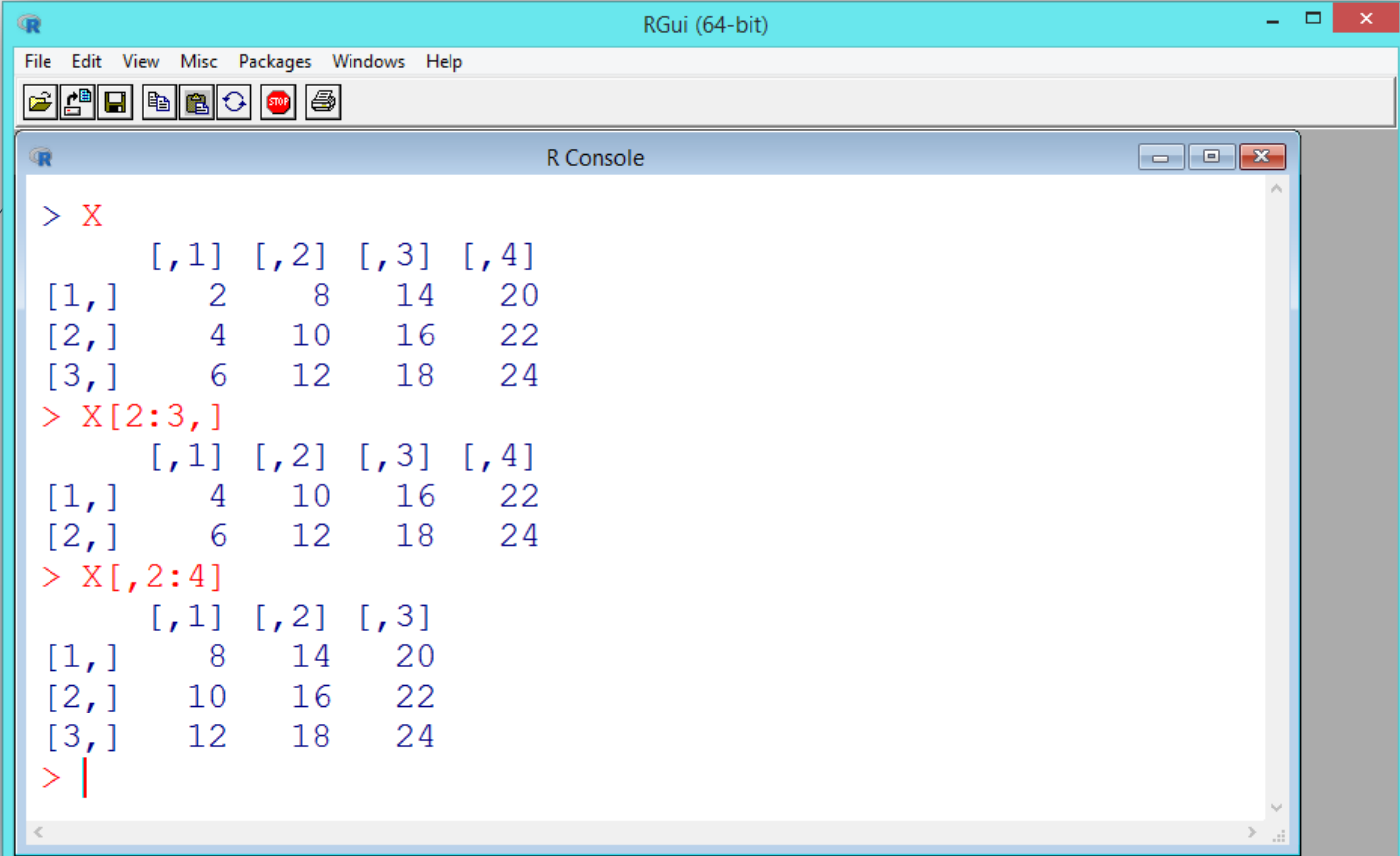


The screenshot shows the RGui (64-bit) window with the R Console. The console displays the following code and output:

```
> X
      [,1] [,2] [,3] [,4]
[1,]    2    8   14   20
[2,]    4   10   16   22
[3,]    6   12   18   24
>
> X[2,]
[1]  4 10 16 22
>
> X[,4]
[1] 20 22 24
> |
```

Matrix

► Addressing Matrix Rows and Columns (2)

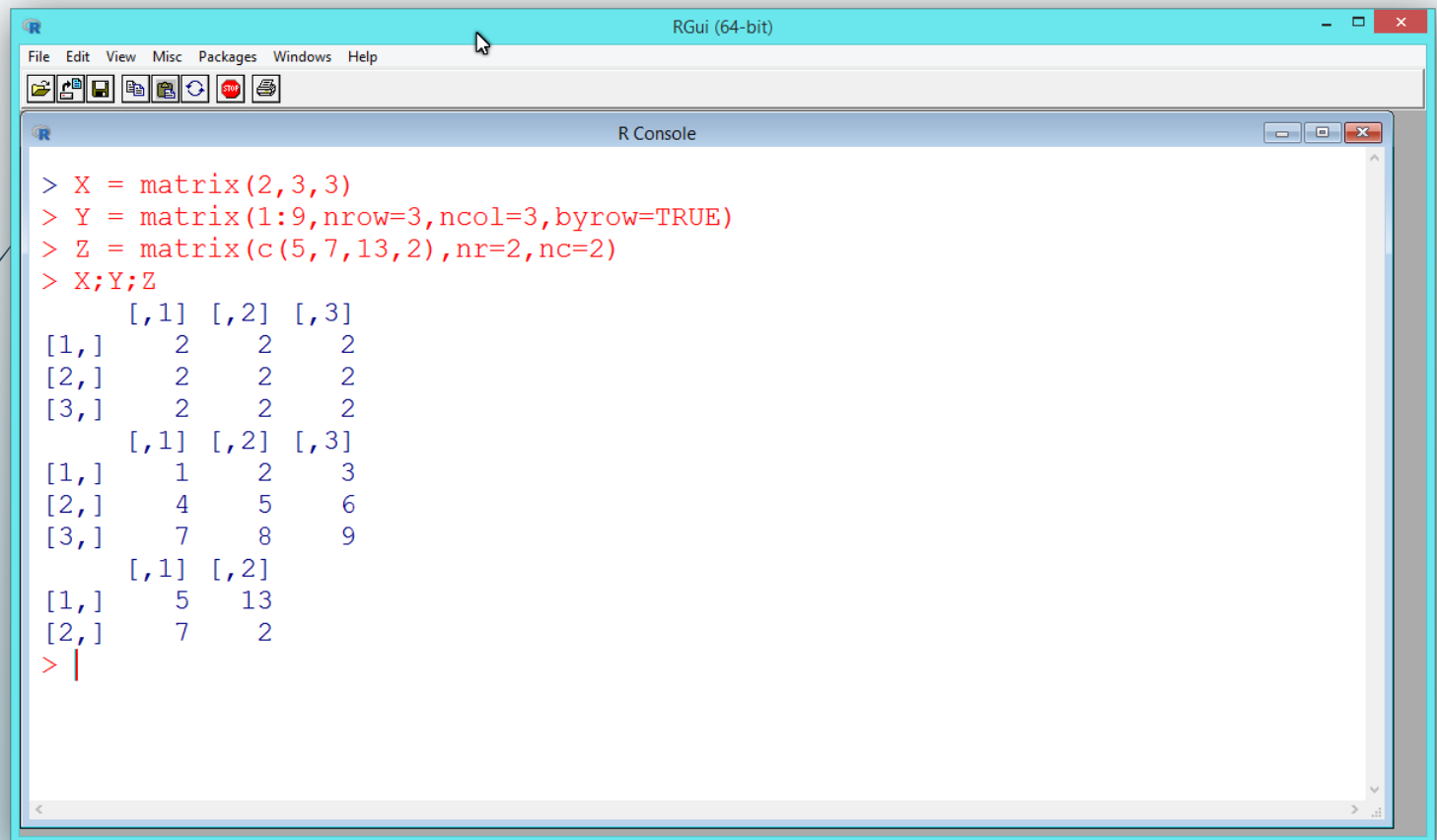


The screenshot shows the RGui (64-bit) interface with the R Console window open. The console displays the following R code and its output:

```
> X
      [,1] [,2] [,3] [,4]
[1,]    2    8   14   20
[2,]    4   10   16   22
[3,]    6   12   18   24
> X[2:3,]
      [,1] [,2] [,3] [,4]
[1,]    4   10   16   22
[2,]    6   12   18   24
> X[,2:4]
      [,1] [,2] [,3]
[1,]    8   14   20
[2,]   10   16   22
[3,]   12   18   24
> |
```


Matrix

► Some Mathematical Operations (1)



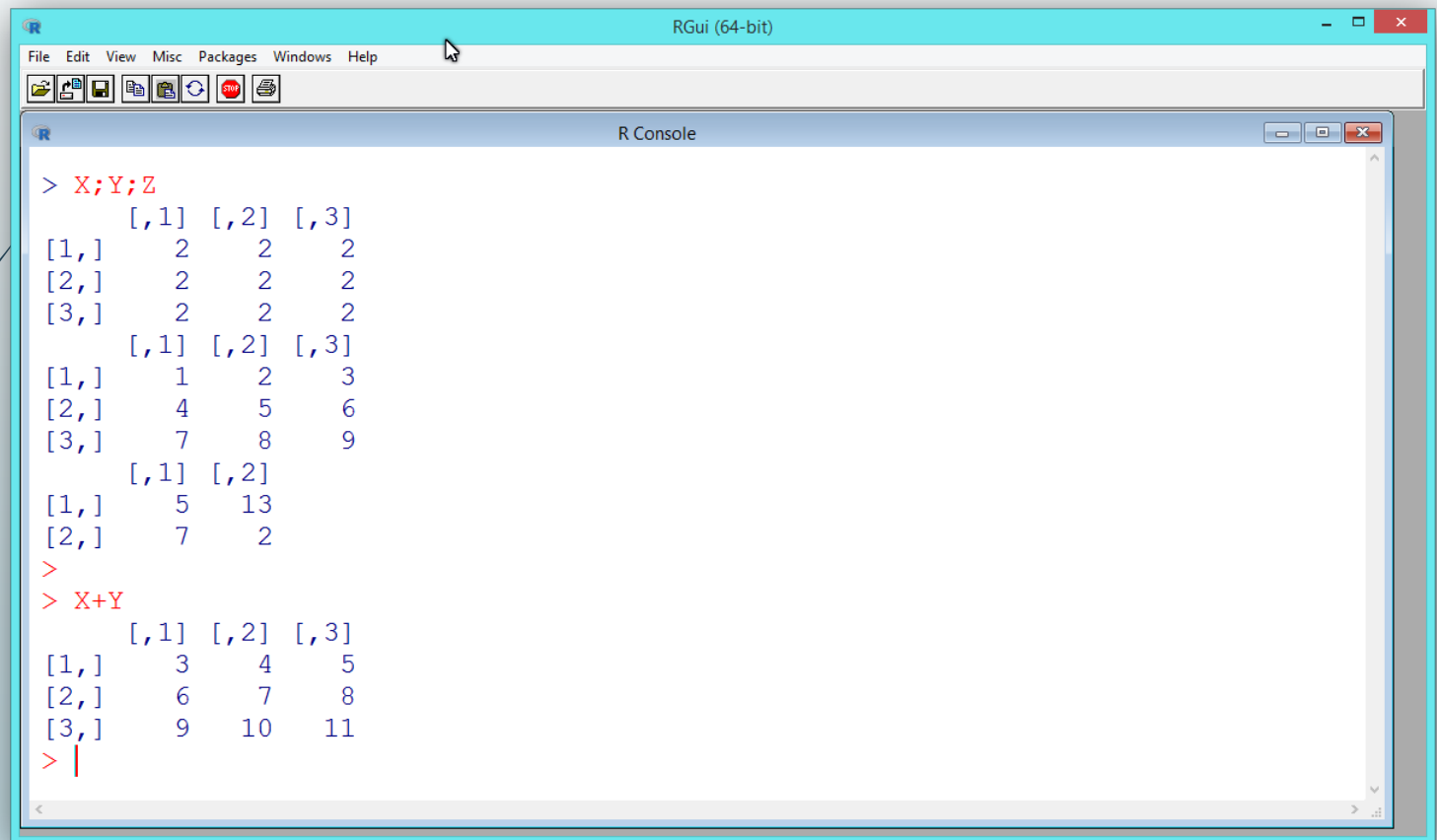
```
> X = matrix(2,3,3)
> Y = matrix(1:9,nrow=3,ncol=3,byrow=TRUE)
> Z = matrix(c(5,7,13,2),nr=2,nc=2)
> X;Y;Z
      [,1] [,2] [,3]
[1,]    2    2    2
[2,]    2    2    2
[3,]    2    2    2

      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9

      [,1] [,2]
[1,]    5   13
[2,]    7    2
> |
```

Matrix

► Some Mathematical Operations (2)

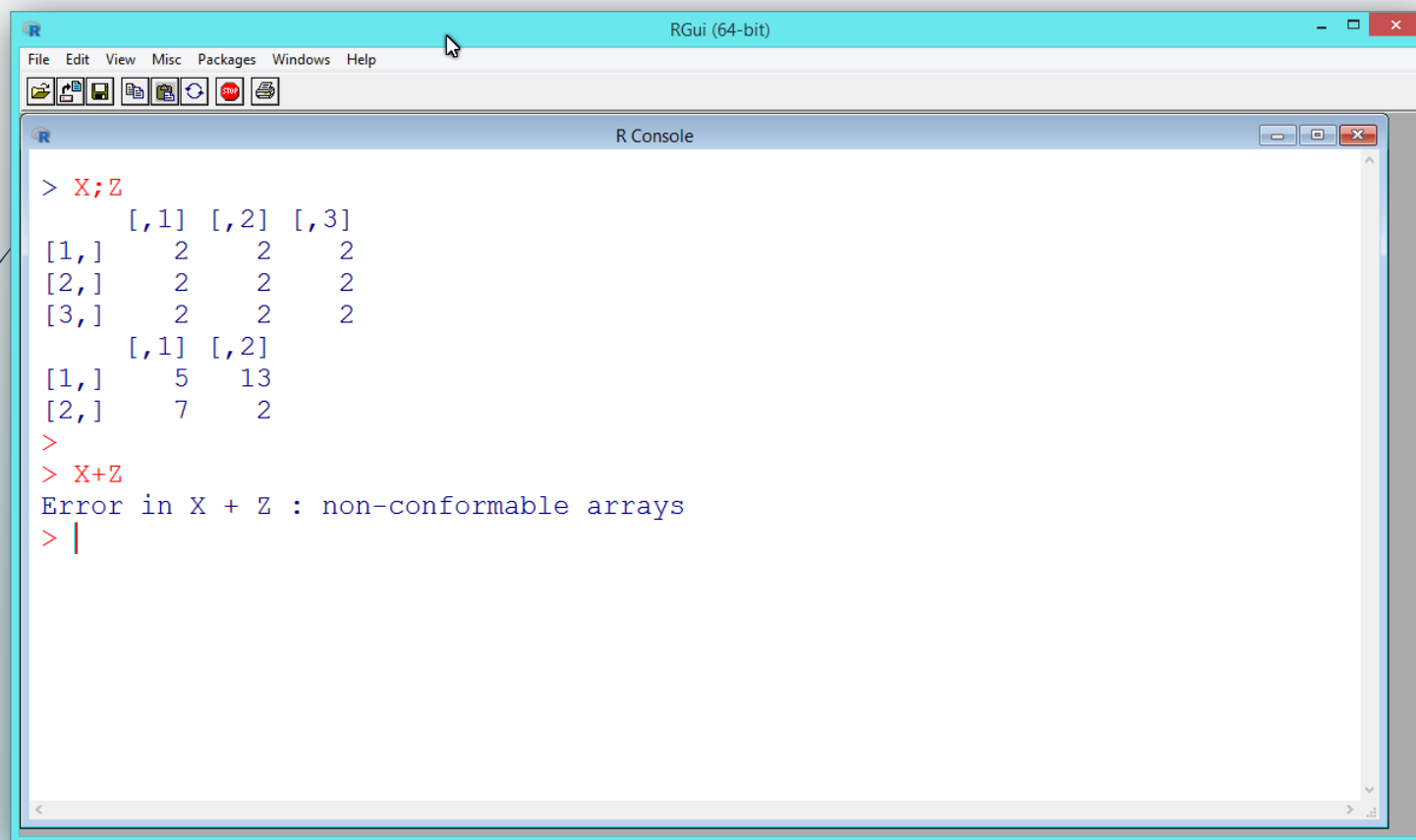


The screenshot shows the RGui (64-bit) R Console window. The console displays the following R code and its output:

```
> X;Y;Z
      [,1] [,2] [,3]
[1,]    2    2    2
[2,]    2    2    2
[3,]    2    2    2
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
      [,1] [,2]
[1,]    5   13
[2,]    7    2
>
> X+Y
      [,1] [,2] [,3]
[1,]    3    4    5
[2,]    6    7    8
[3,]    9   10   11
> |
```

Matrix

► Some Mathematical Operations (3)

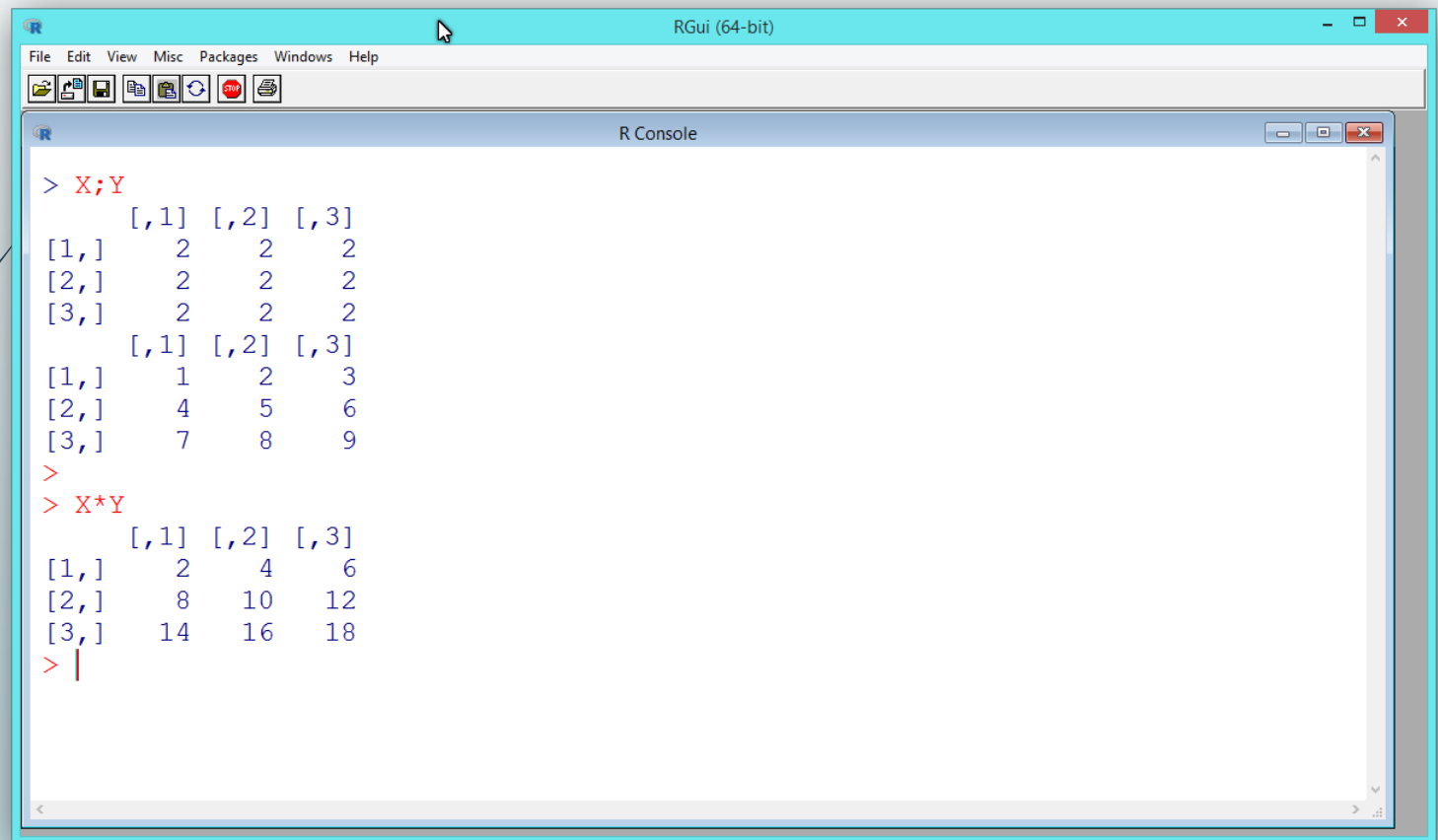


```
> X;Z
      [,1] [,2] [,3]
[1,]    2    2    2
[2,]    2    2    2
[3,]    2    2    2

      [,1] [,2]
[1,]    5   13
[2,]    7    2
>
> X+Z
Error in X + Z : non-conformable arrays
> |
```

Matrix

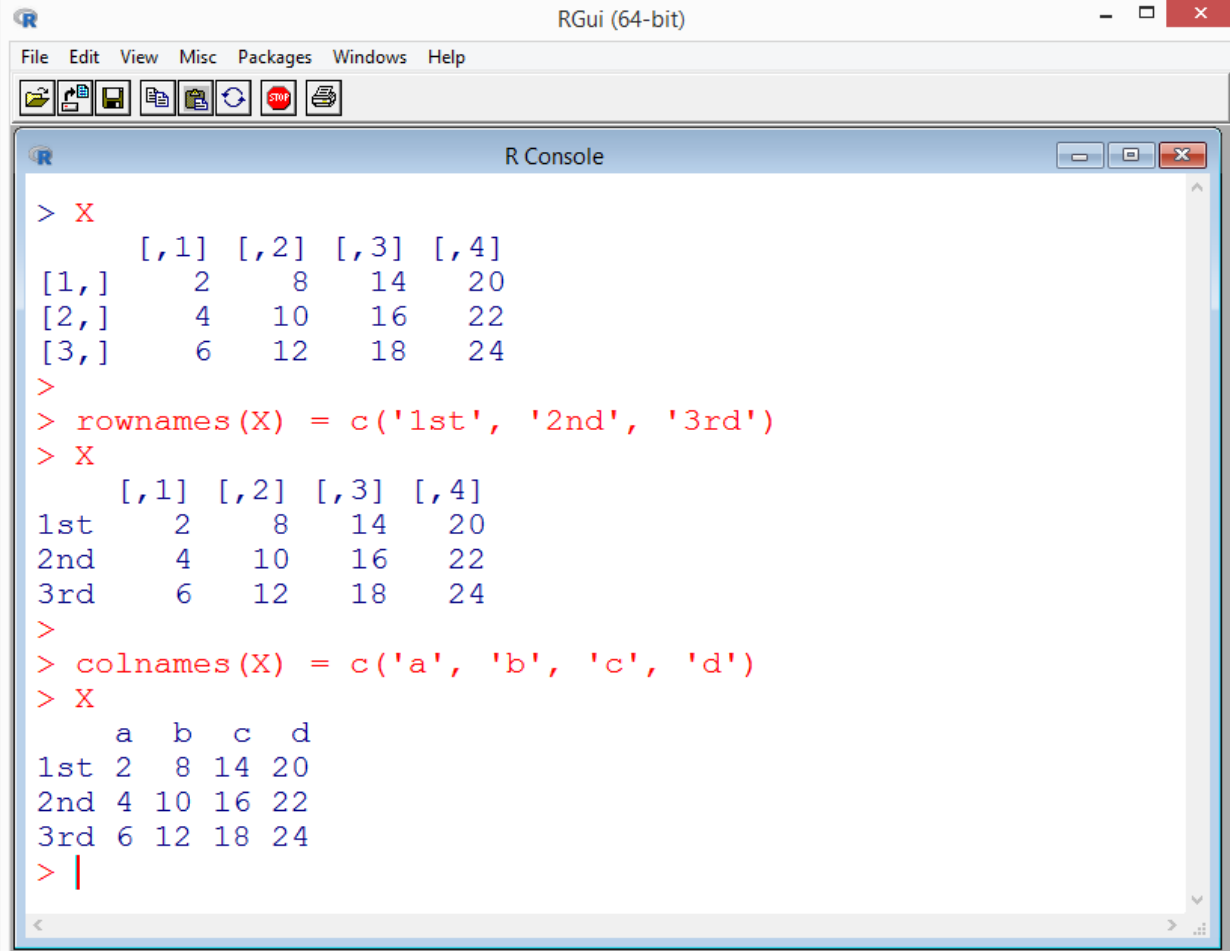
► Some Mathematical Operations (4)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> X;Y
      [,1] [,2] [,3]
[1,]    2    2    2
[2,]    2    2    2
[3,]    2    2    2
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
>
> X*Y
      [,1] [,2] [,3]
[1,]    2    4    6
[2,]    8   10   12
[3,]   14   16   18
> |
```

Matrix

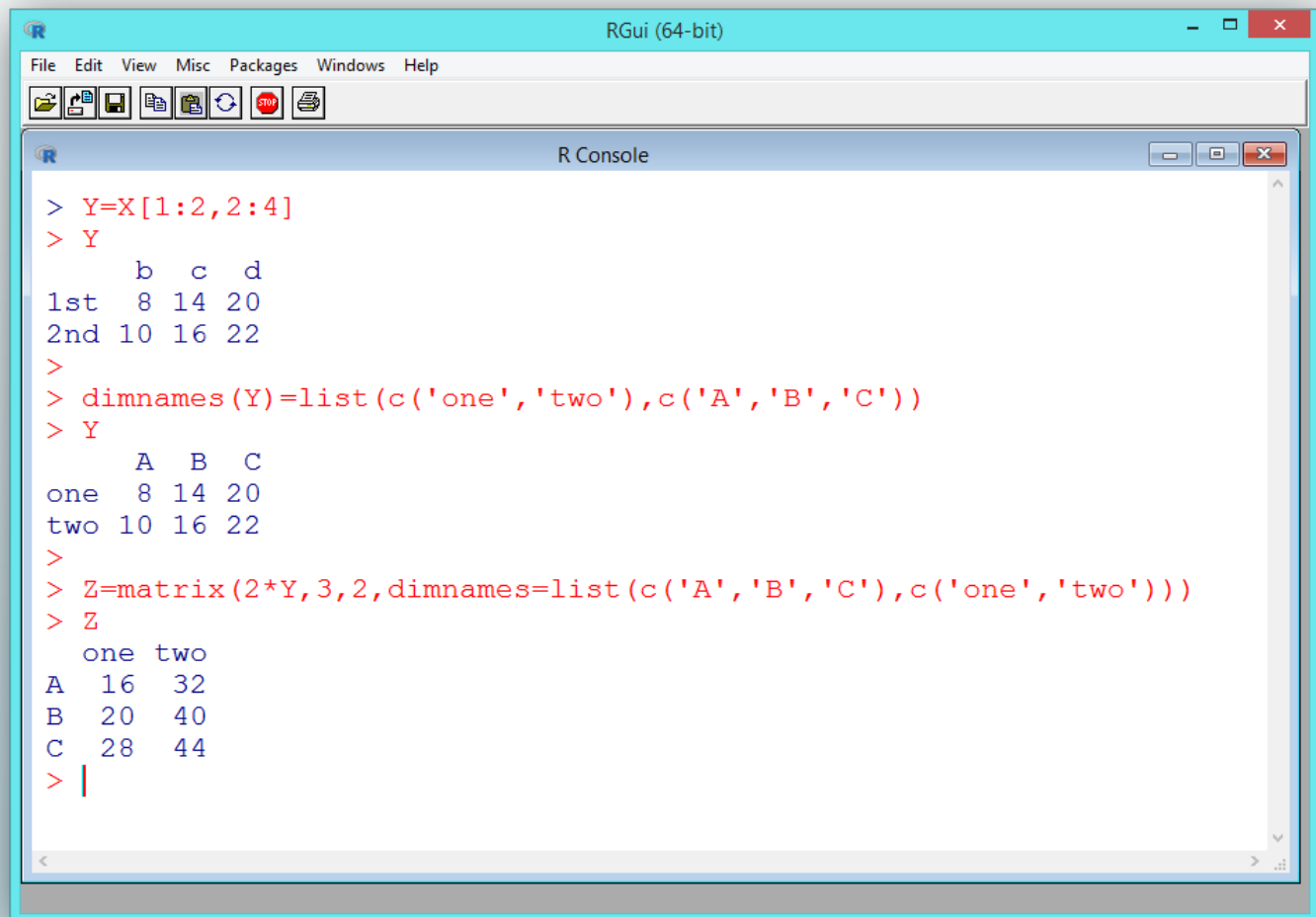
► Naming Matrix Rows and Columns (1)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]
R Console
> X
  [,1] [,2] [,3] [,4]
[1,]  2   8  14  20
[2,]  4  10  16  22
[3,]  6  12  18  24
>
> rownames(X) = c('1st', '2nd', '3rd')
> X
  [,1] [,2] [,3] [,4]
1st   2   8  14  20
2nd   4  10  16  22
3rd   6  12  18  24
>
> colnames(X) = c('a', 'b', 'c', 'd')
> X
   a  b  c  d
1st 2  8 14 20
2nd 4 10 16 22
3rd 6 12 18 24
> |
```

Matrix

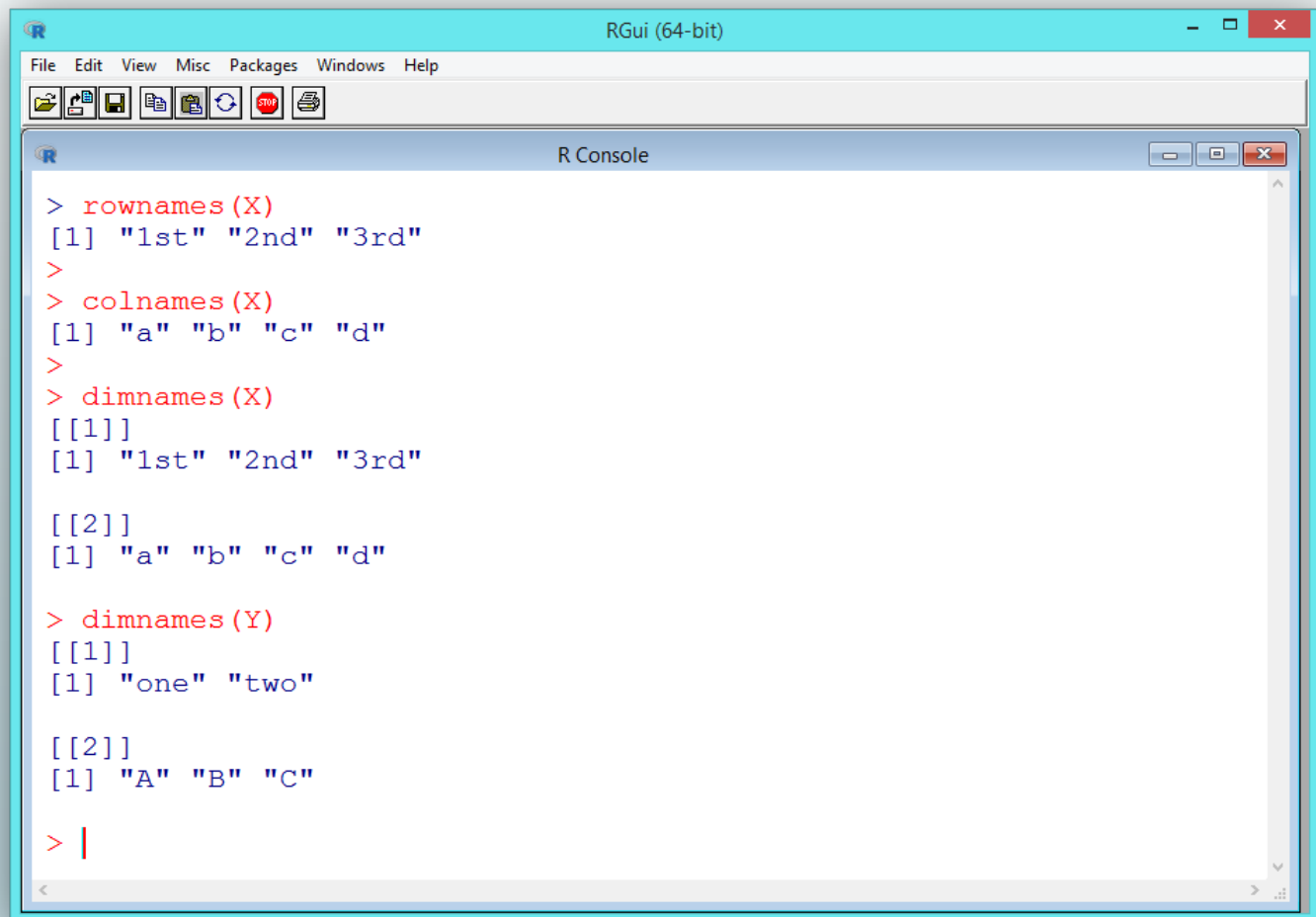
► Naming Matrix Rows and Columns (2)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]
R Console
> Y=X[1:2,2:4]
> Y
      b c d
1st  8 14 20
2nd 10 16 22
>
> dimnames(Y)=list(c('one','two'),c('A','B','C'))
> Y
      A B C
one  8 14 20
two 10 16 22
>
> Z=matrix(2*Y,3,2,dimnames=list(c('A','B','C'),c('one','two')))
> Z
      one two
A    16  32
B    20  40
C    28  44
> |
```

Matrix

► Naming Matrix Rows and Columns (3)



```
> rownames(X)
[1] "1st" "2nd" "3rd"
>
> colnames(X)
[1] "a" "b" "c" "d"
>
> dimnames(X)
[[1]]
[1] "1st" "2nd" "3rd"

[[2]]
[1] "a" "b" "c" "d"

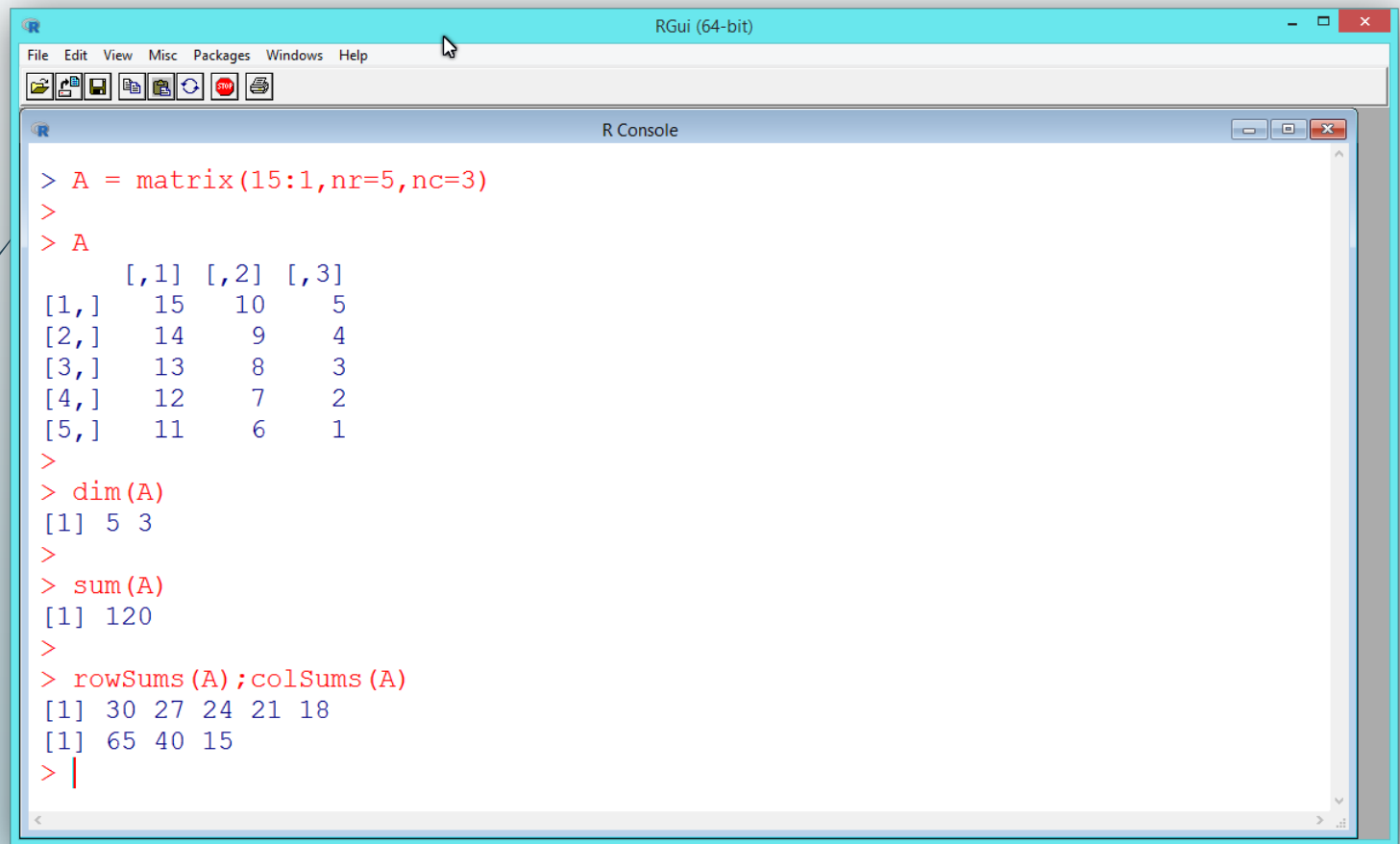
> dimnames(Y)
[[1]]
[1] "one" "two"

[[2]]
[1] "A" "B" "C"

> |
```

Matrix

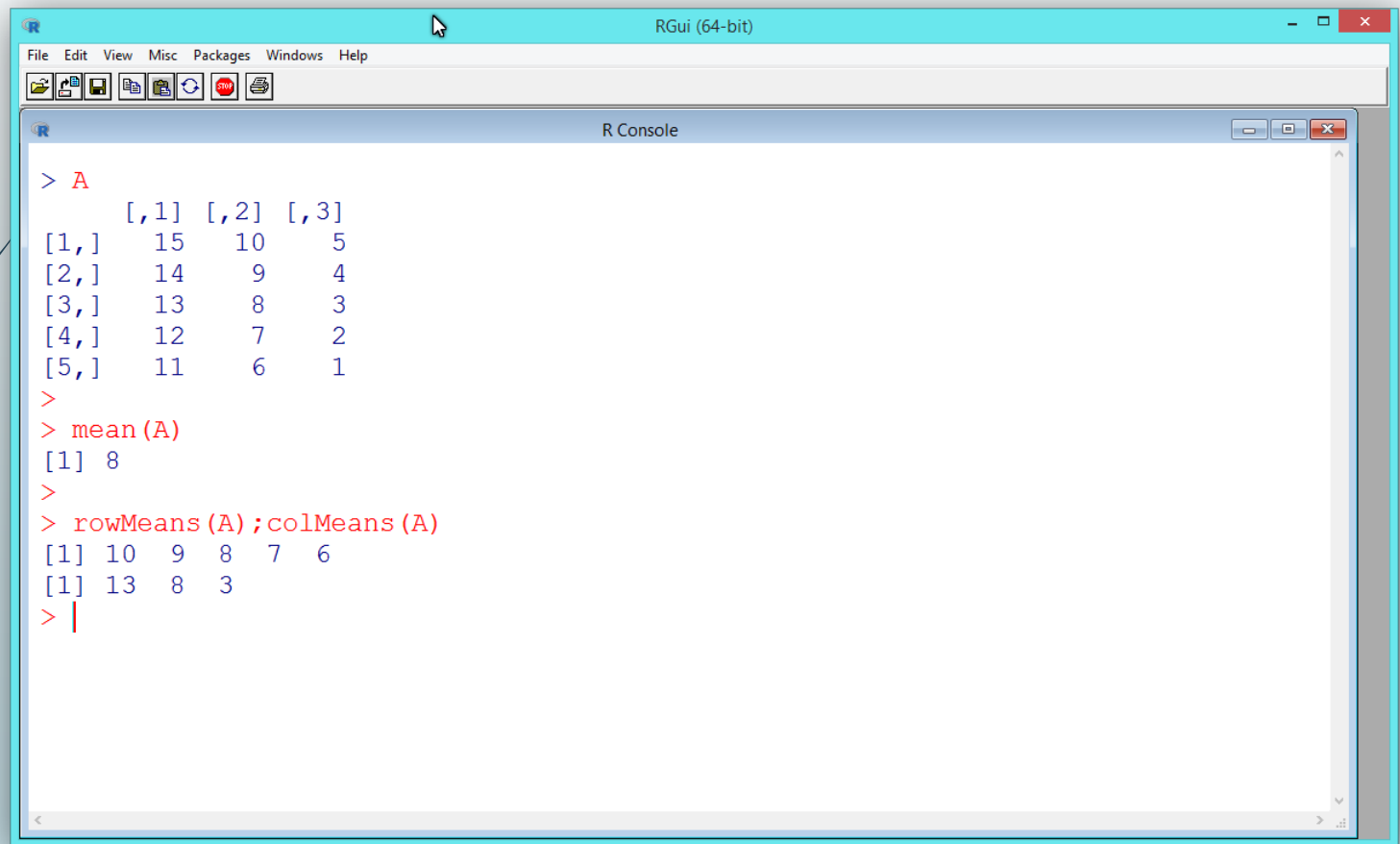
► Some Built-in Functions for Matrices (1)



```
> A = matrix(15:1,nr=5,nc=3)
>
> A
      [,1] [,2] [,3]
[1,]  15   10   5
[2,]  14   9   4
[3,]  13   8   3
[4,]  12   7   2
[5,]  11   6   1
>
> dim(A)
[1] 5 3
>
> sum(A)
[1] 120
>
> rowSums(A);colSums(A)
[1] 30 27 24 21 18
[1] 65 40 15
> |
```


Matrix

► Some Built-in Functions for Matrices (2)

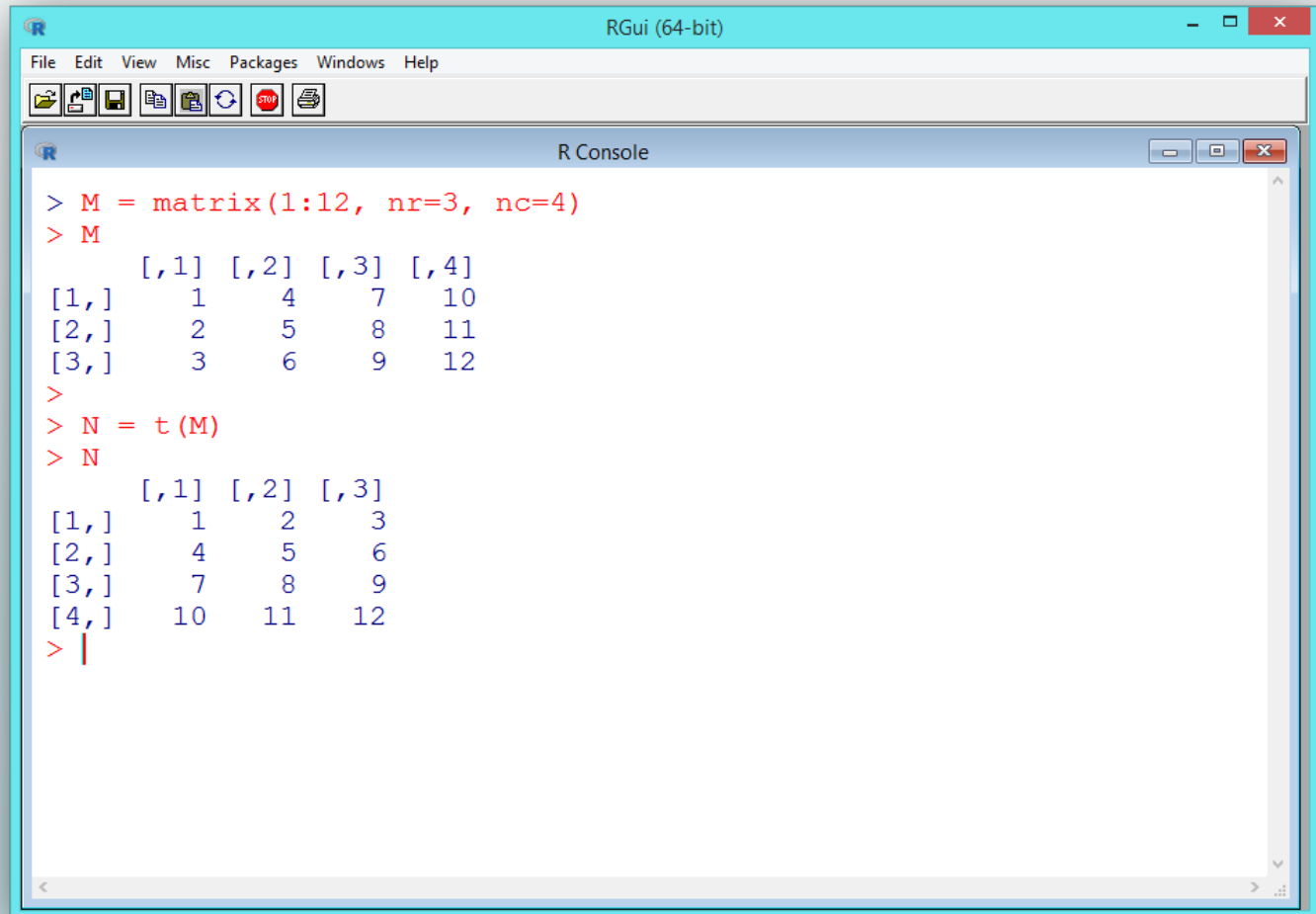


The screenshot shows the RGui (64-bit) interface with the R Console window open. The console displays the following R code and its output:

```
> A
      [,1] [,2] [,3]
[1,]  15  10   5
[2,]  14   9   4
[3,]  13   8   3
[4,]  12   7   2
[5,]  11   6   1
>
> mean(A)
[1] 8
>
> rowMeans(A); colMeans(A)
[1] 10  9  8  7  6
[1] 13  8  3
> |
```

Matrix

► Some Built-in Functions for Matrices (3)

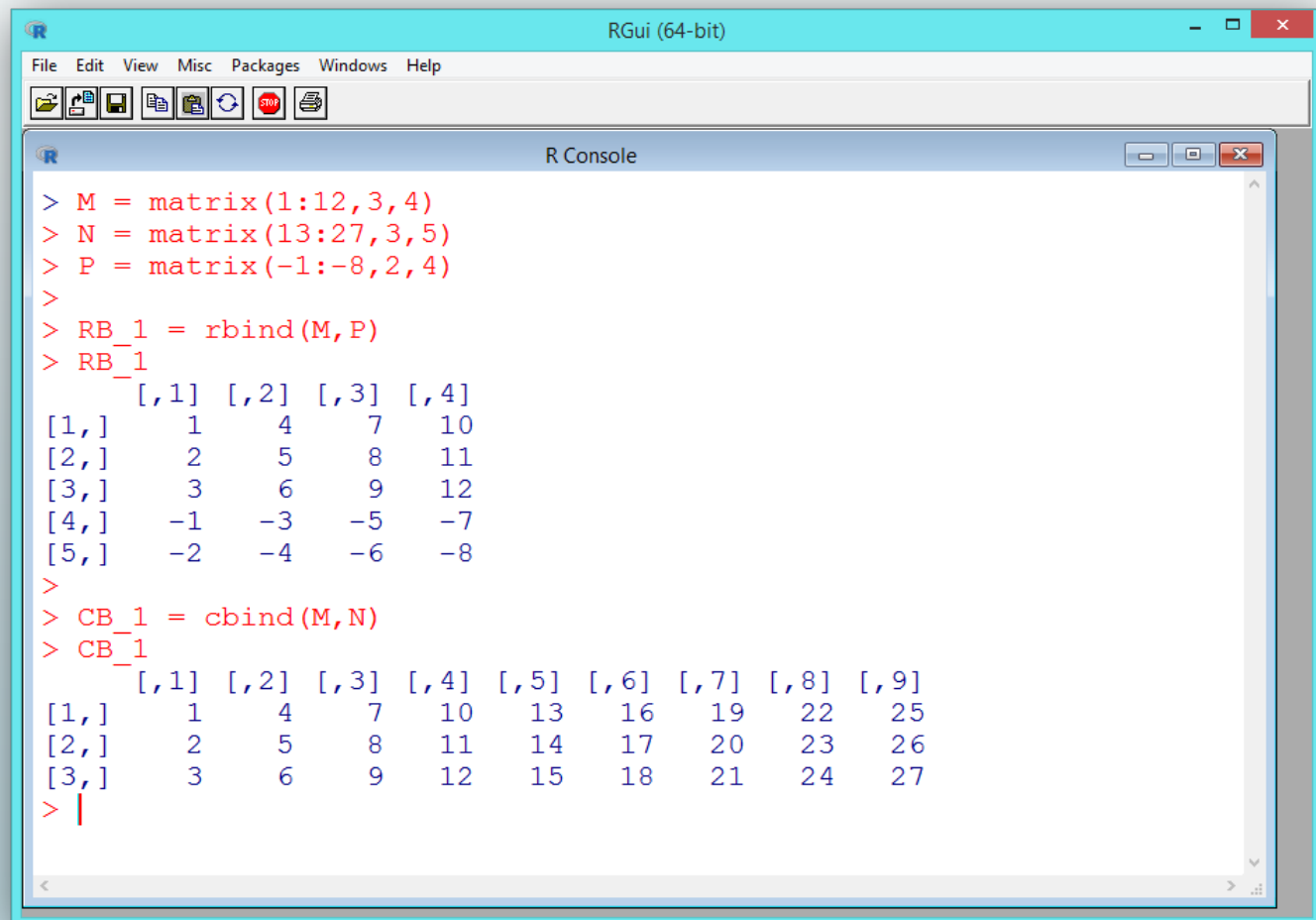


```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> M = matrix(1:12, nr=3, nc=4)
> M
  [,1] [,2] [,3] [,4]
[1,]  1   4   7  10
[2,]  2   5   8  11
[3,]  3   6   9  12
>
> N = t(M)
> N
  [,1] [,2] [,3]
[1,]  1   2   3
[2,]  4   5   6
[3,]  7   8   9
[4,] 10  11  12
> |
```

Matrix

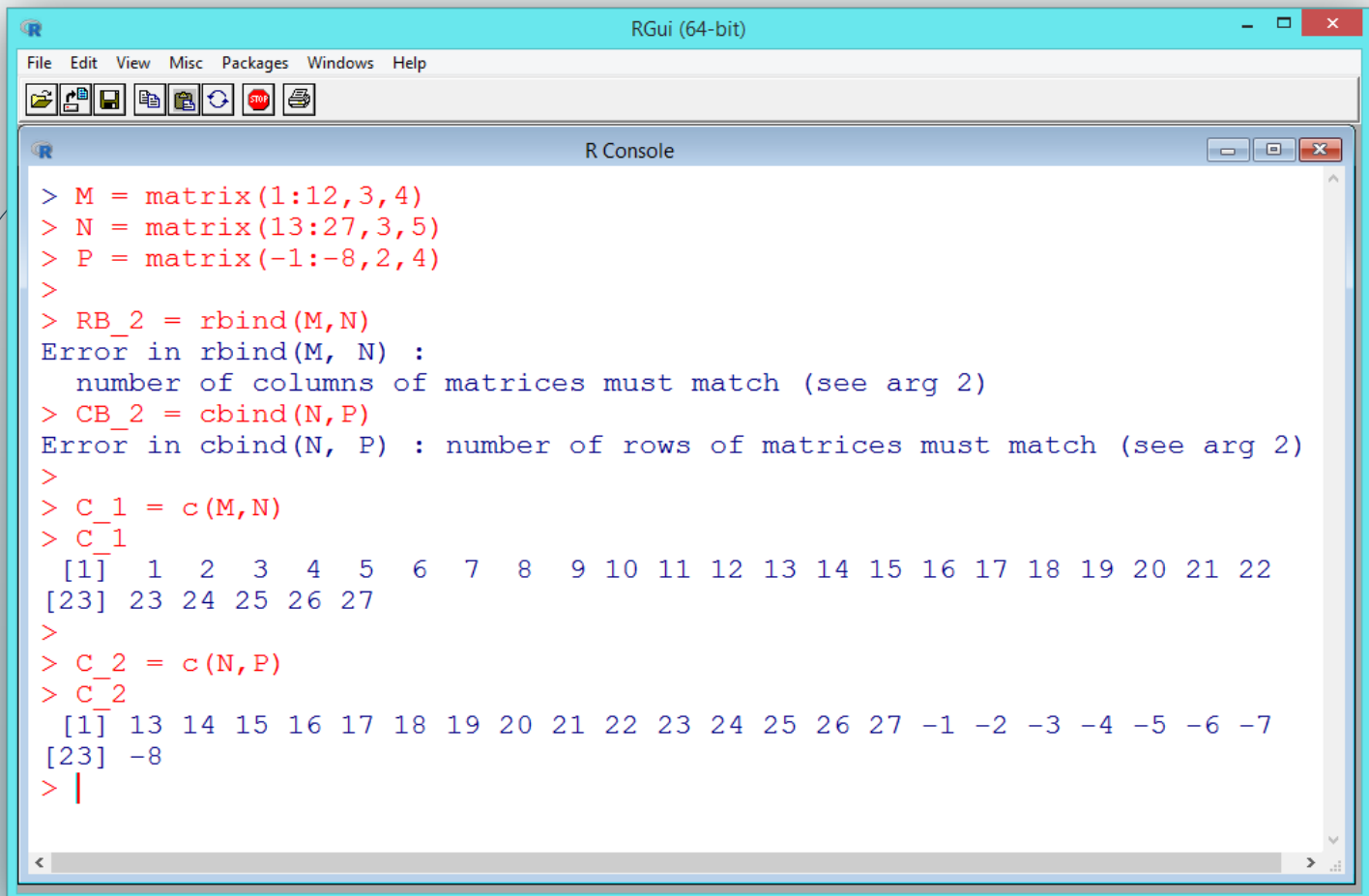
► Some Built-in Functions for Matrices (4)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]
R Console
> M = matrix(1:12,3,4)
> N = matrix(13:27,3,5)
> P = matrix(-1:-8,2,4)
>
> RB_1 = rbind(M,P)
> RB_1
      [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
[4,]   -1   -3   -5   -7
[5,]   -2   -4   -6   -8
>
> CB_1 = cbind(M,N)
> CB_1
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
[1,]    1    4    7   10   13   16   19   22   25
[2,]    2    5    8   11   14   17   20   23   26
[3,]    3    6    9   12   15   18   21   24   27
> |
```

Matrix

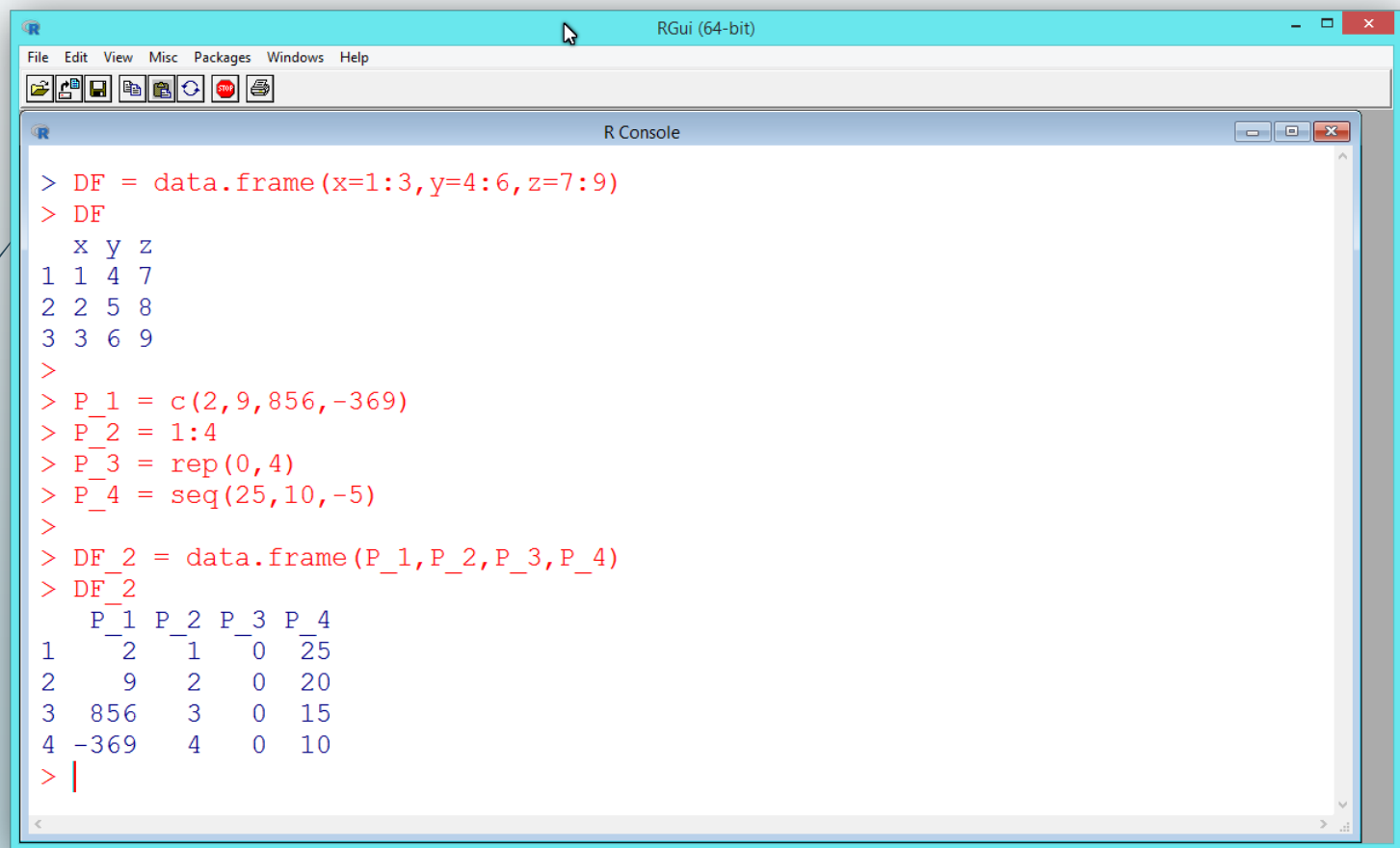
► Some Built-in Functions for Matrices (5)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> M = matrix(1:12,3,4)
> N = matrix(13:27,3,5)
> P = matrix(-1:-8,2,4)
>
> RB_2 = rbind(M,N)
Error in rbind(M, N) :
  number of columns of matrices must match (see arg 2)
> CB_2 = cbind(N,P)
Error in cbind(N, P) : number of rows of matrices must match (see arg 2)
>
> C_1 = c(M,N)
> C_1
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
[23] 23 24 25 26 27
>
> C_2 = c(N,P)
> C_2
[1] 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 -1 -2 -3 -4 -5 -6 -7
[23] -8
> |
```

Data Frame

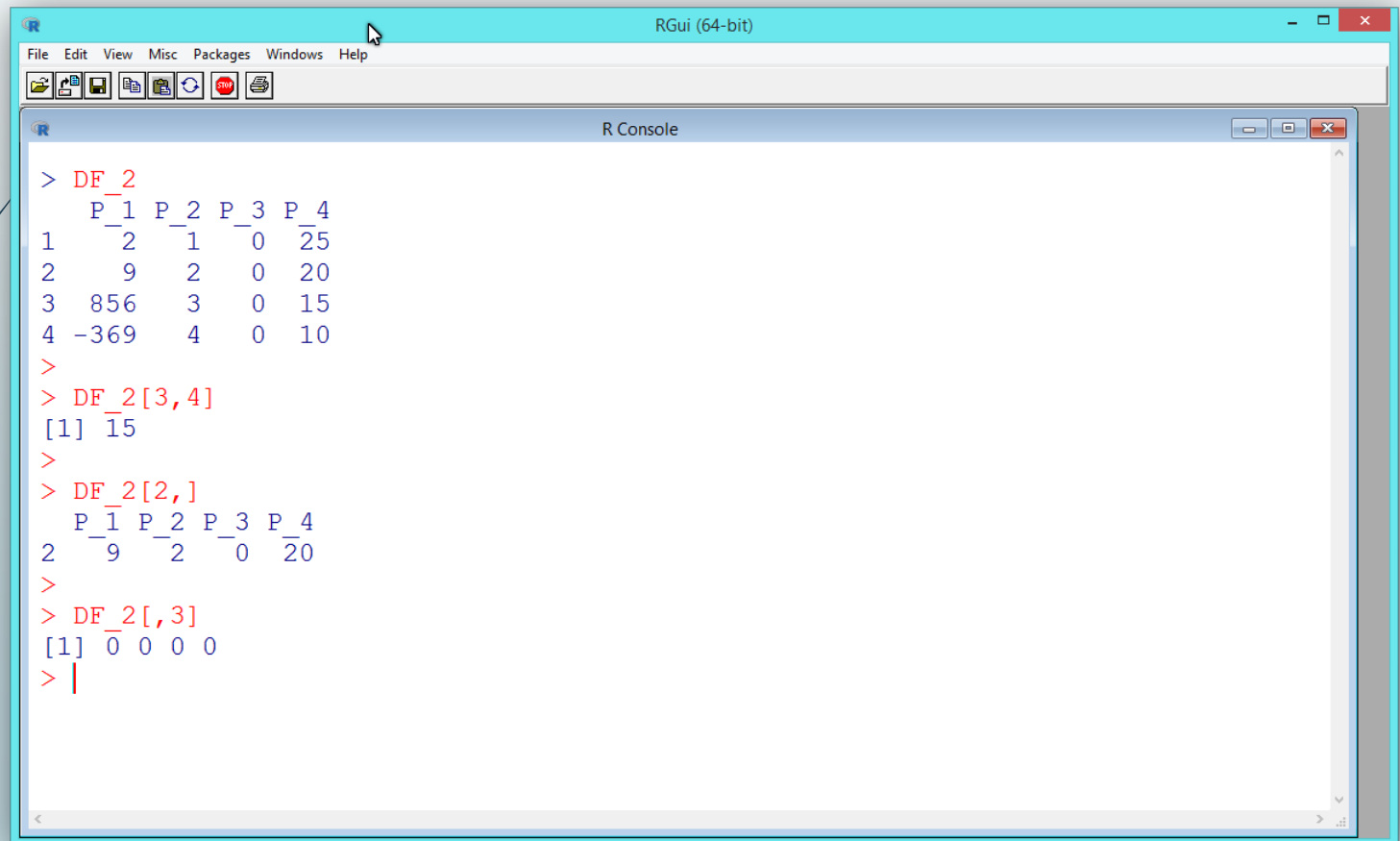
► Assignment



```
> DF = data.frame(x=1:3,y=4:6,z=7:9)
> DF
  x y z
1 1 4 7
2 2 5 8
3 3 6 9
>
> P_1 = c(2,9,856,-369)
> P_2 = 1:4
> P_3 = rep(0,4)
> P_4 = seq(25,10,-5)
>
> DF_2 = data.frame(P_1,P_2,P_3,P_4)
> DF_2
  P_1 P_2 P_3 P_4
1    2  1  0  25
2    9  2  0  20
3 856  3  0  15
4-369  4  0  10
> |
```

Data Frame

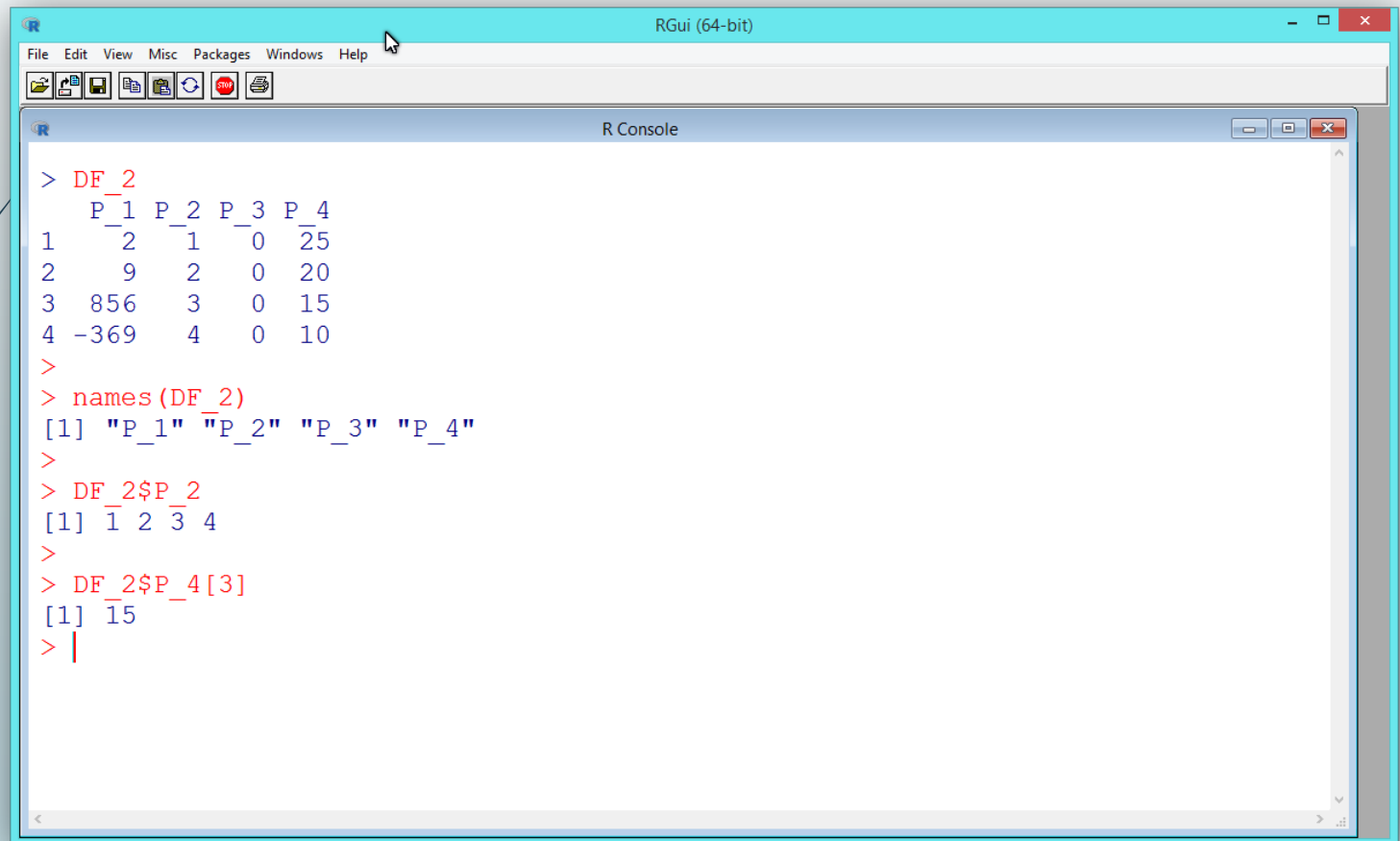
- Addressing Data Frame Elements, Rows and Columns (1)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> DF_2
  P_1 P_2 P_3 P_4
1   2   1   0  25
2   9   2   0  20
3 856   3   0  15
4 -369  4   0  10
>
> DF_2[3,4]
[1] 15
>
> DF_2[2,]
  P_1 P_2 P_3 P_4
2   9   2   0  20
>
> DF_2[,3]
[1] 0 0 0 0
> |
```

Data Frame

Addressing Data Frame Elements, Rows and Columns (2)



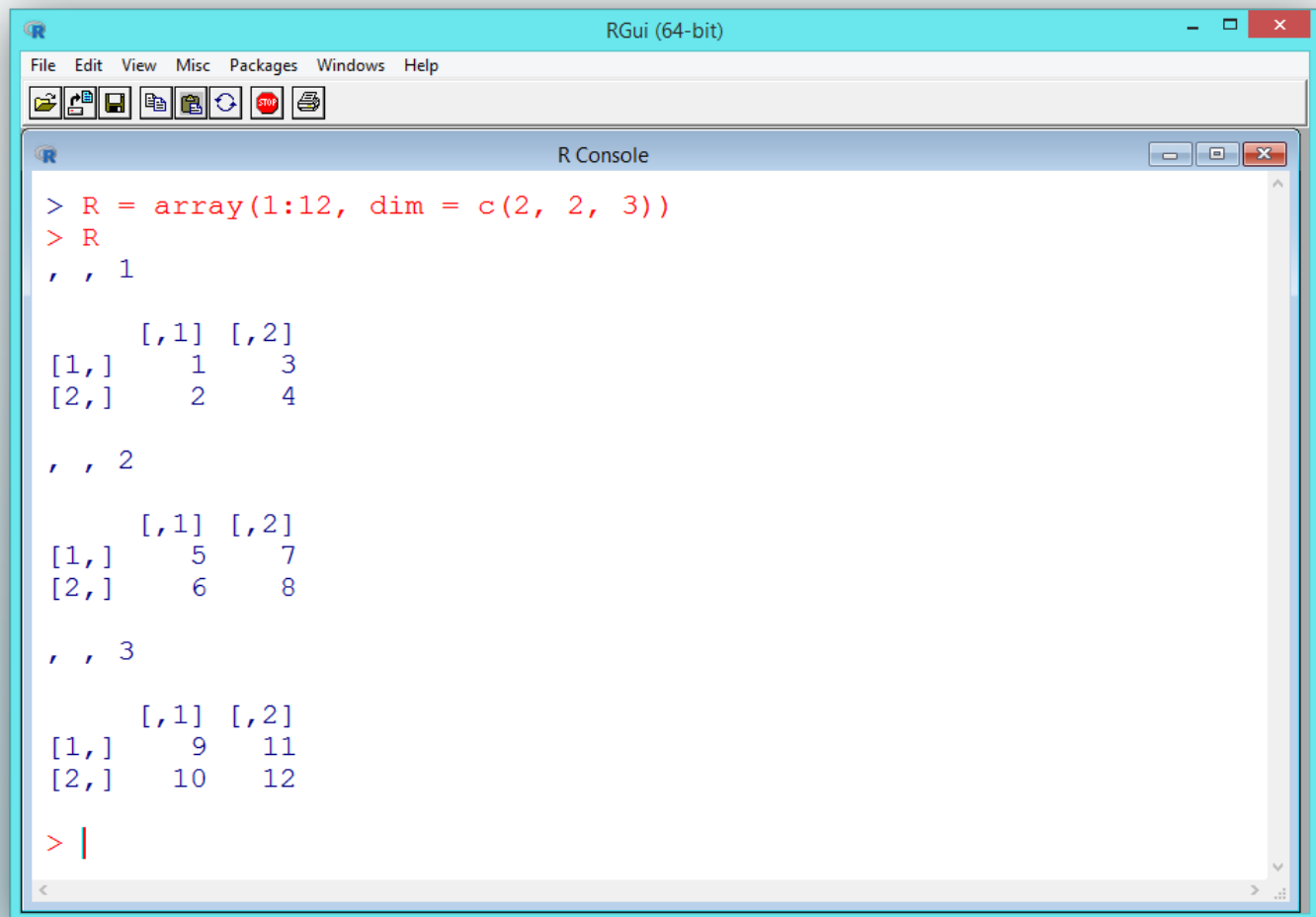
The screenshot shows the RGui (64-bit) interface with the R Console window open. The console displays the following R code and its output:

```
> DF_2
  P_1 P_2 P_3 P_4
1    2  1  0  25
2    9  2  0  20
3  856  3  0  15
4 -369  4  0  10
>
> names(DF_2)
[1] "P_1" "P_2" "P_3" "P_4"
>
> DF_2$P_2
[1] 1 2 3 4
>
> DF_2$P_4[3]
[1] 15
> |
```

The data frame DF_2 is a 4x4 matrix with columns P_1, P_2, P_3, and P_4. The output shows the names of the columns, the values of the second column (P_2), and the value of the third row in the fourth column (P_4).

Array

► Assignment (1)

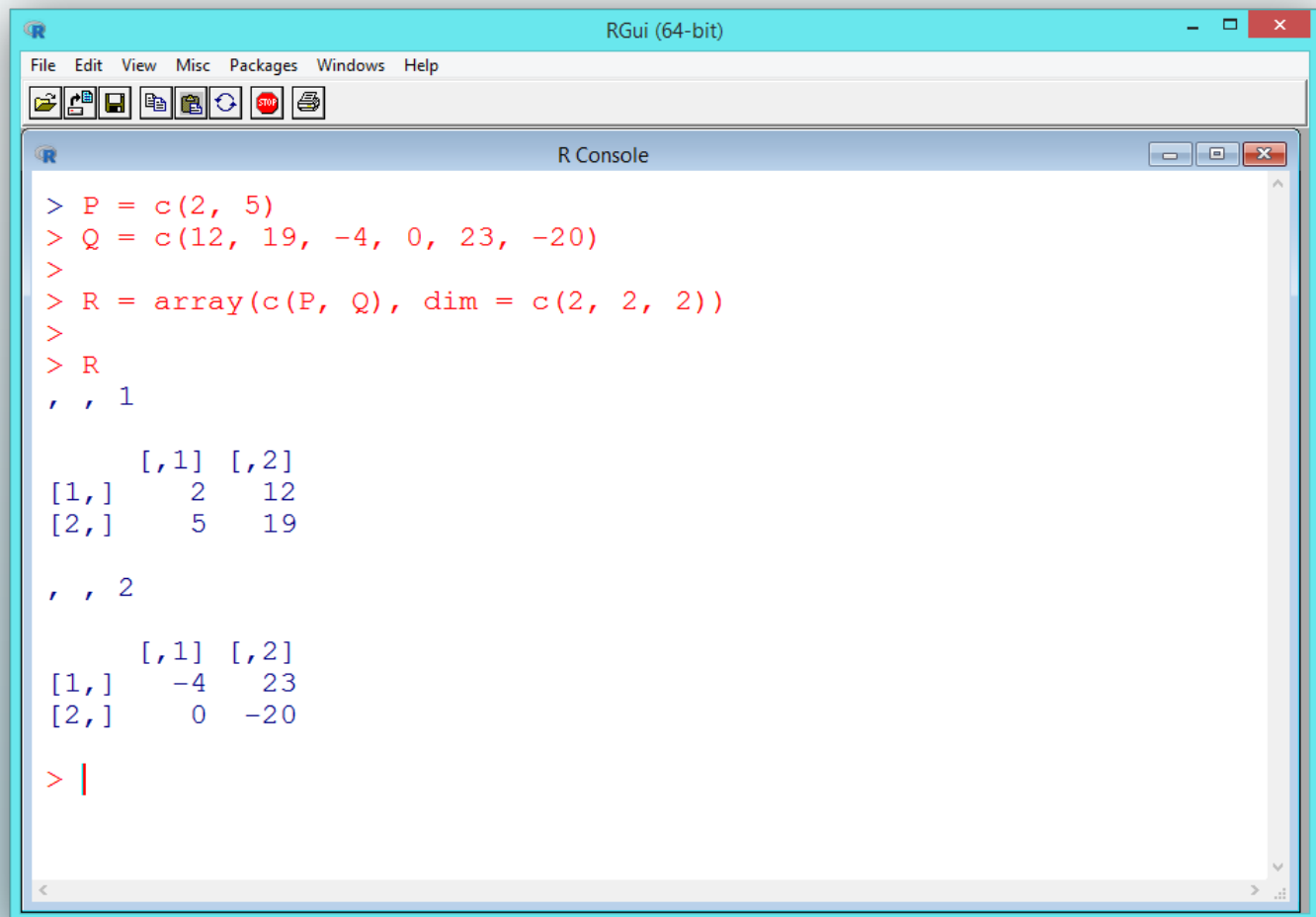


```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> R = array(1:12, dim = c(2, 2, 3))
> R
, , 1
     [,1] [,2]
[1,]    1    3
[2,]    2    4
, , 2
     [,1] [,2]
[1,]    5    7
[2,]    6    8
, , 3
     [,1] [,2]
[1,]    9   11
[2,]   10   12
> |
```


Array

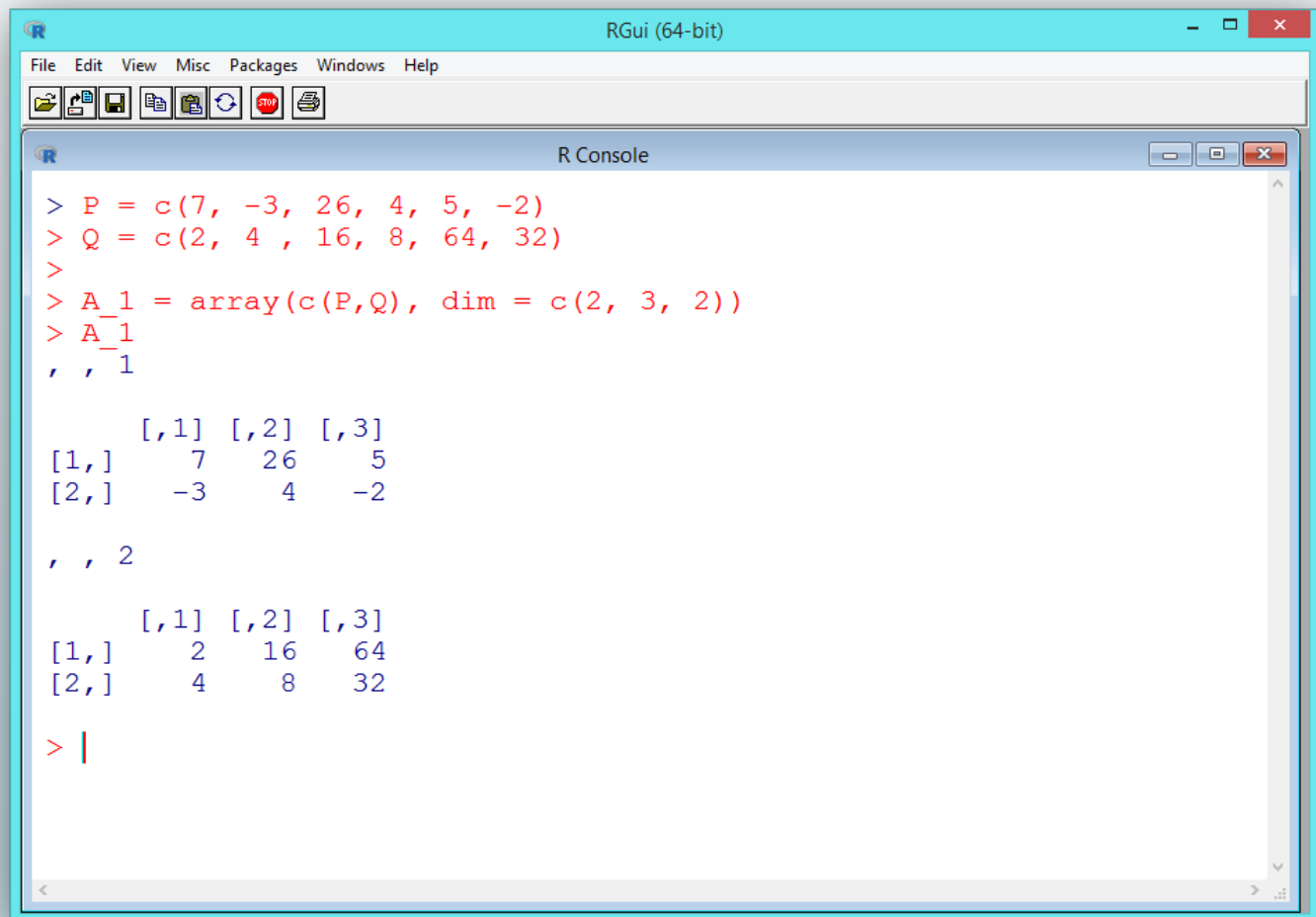
► Assignment (2)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]
R Console
> P = c(2, 5)
> Q = c(12, 19, -4, 0, 23, -20)
>
> R = array(c(P, Q), dim = c(2, 2, 2))
>
> R
, , 1
      [,1] [,2]
[1,]    2   12
[2,]    5   19
, , 2
      [,1] [,2]
[1,]   -4   23
[2,]    0  -20
> |
```

Array

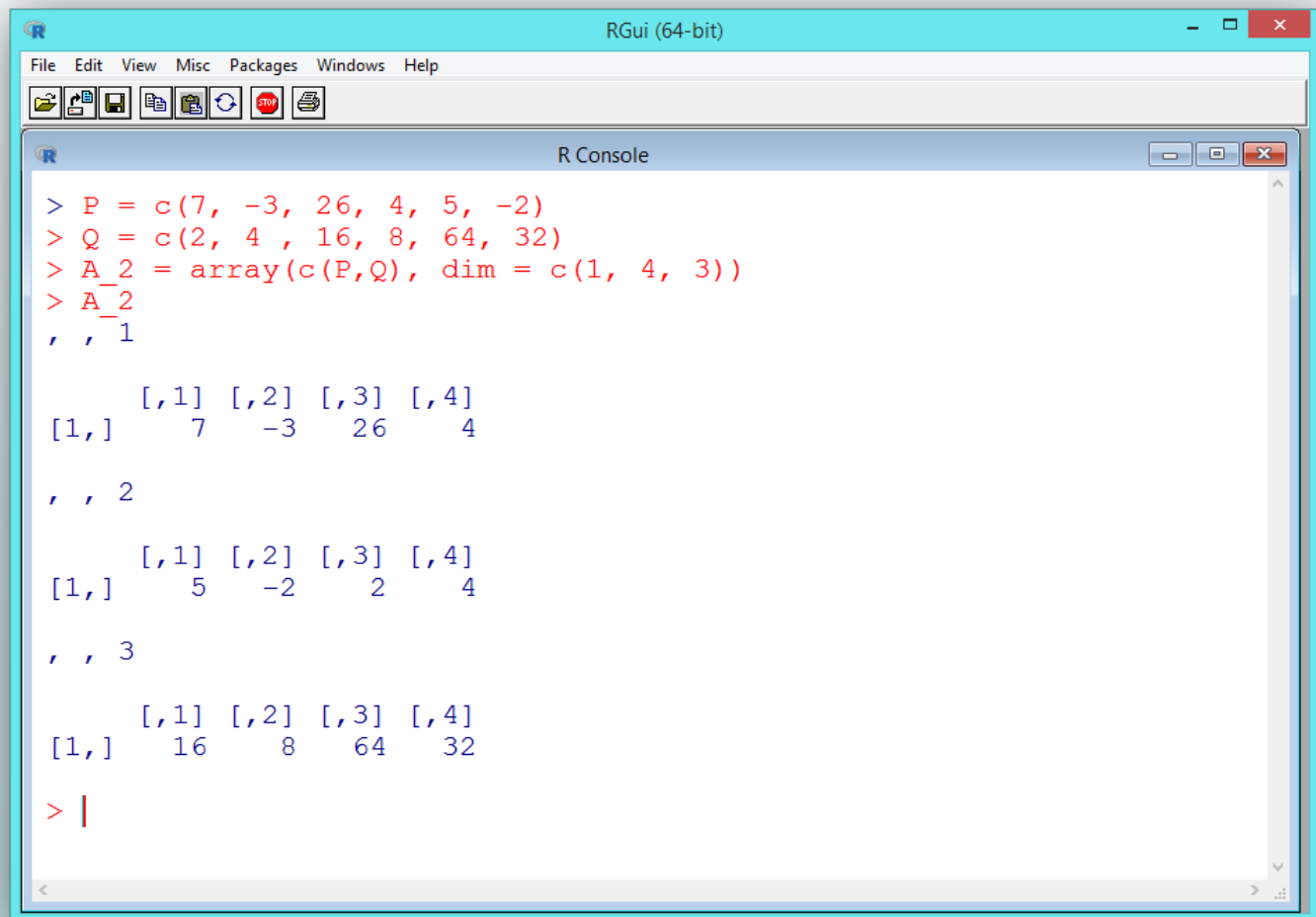
► Assignment (3)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]
R Console
> P = c(7, -3, 26, 4, 5, -2)
> Q = c(2, 4, 16, 8, 64, 32)
>
> A_1 = array(c(P,Q), dim = c(2, 3, 2))
> A_1
, , 1
      [,1] [,2] [,3]
[1,]    7   26    5
[2,]   -3    4   -2
, , 2
      [,1] [,2] [,3]
[1,]    2   16   64
[2,]    4    8   32
> |
```

Array

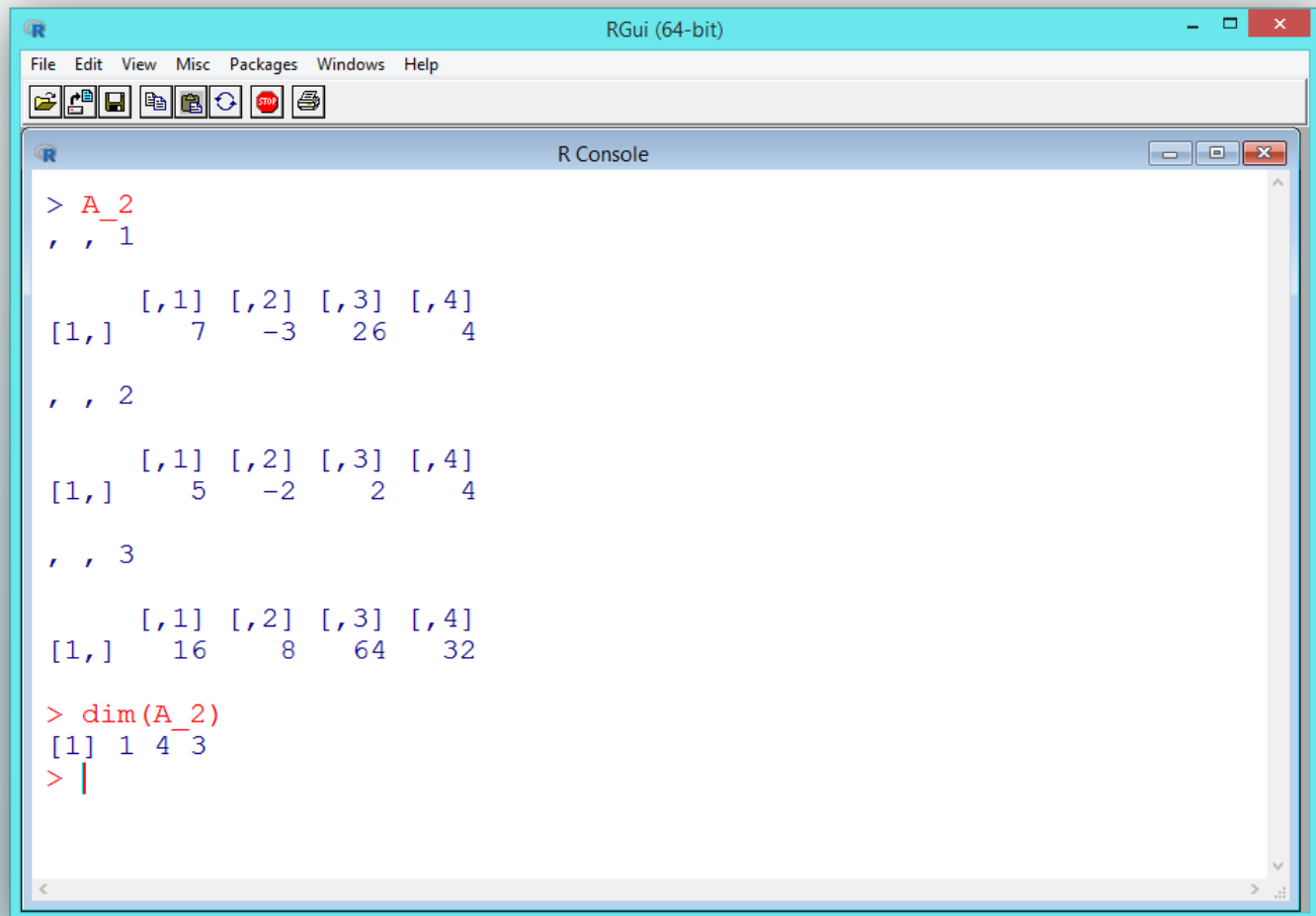
► Assignment (4)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]
R Console
> P = c(7, -3, 26, 4, 5, -2)
> Q = c(2, 4, 16, 8, 64, 32)
> A_2 = array(c(P,Q), dim = c(1, 4, 3))
> A_2
, , 1
      [,1] [,2] [,3] [,4]
[1,]    7   -3   26    4
, , 2
      [,1] [,2] [,3] [,4]
[1,]    5   -2    2    4
, , 3
      [,1] [,2] [,3] [,4]
[1,]   16    8   64   32
> |
```

Array

► Array Dimensions

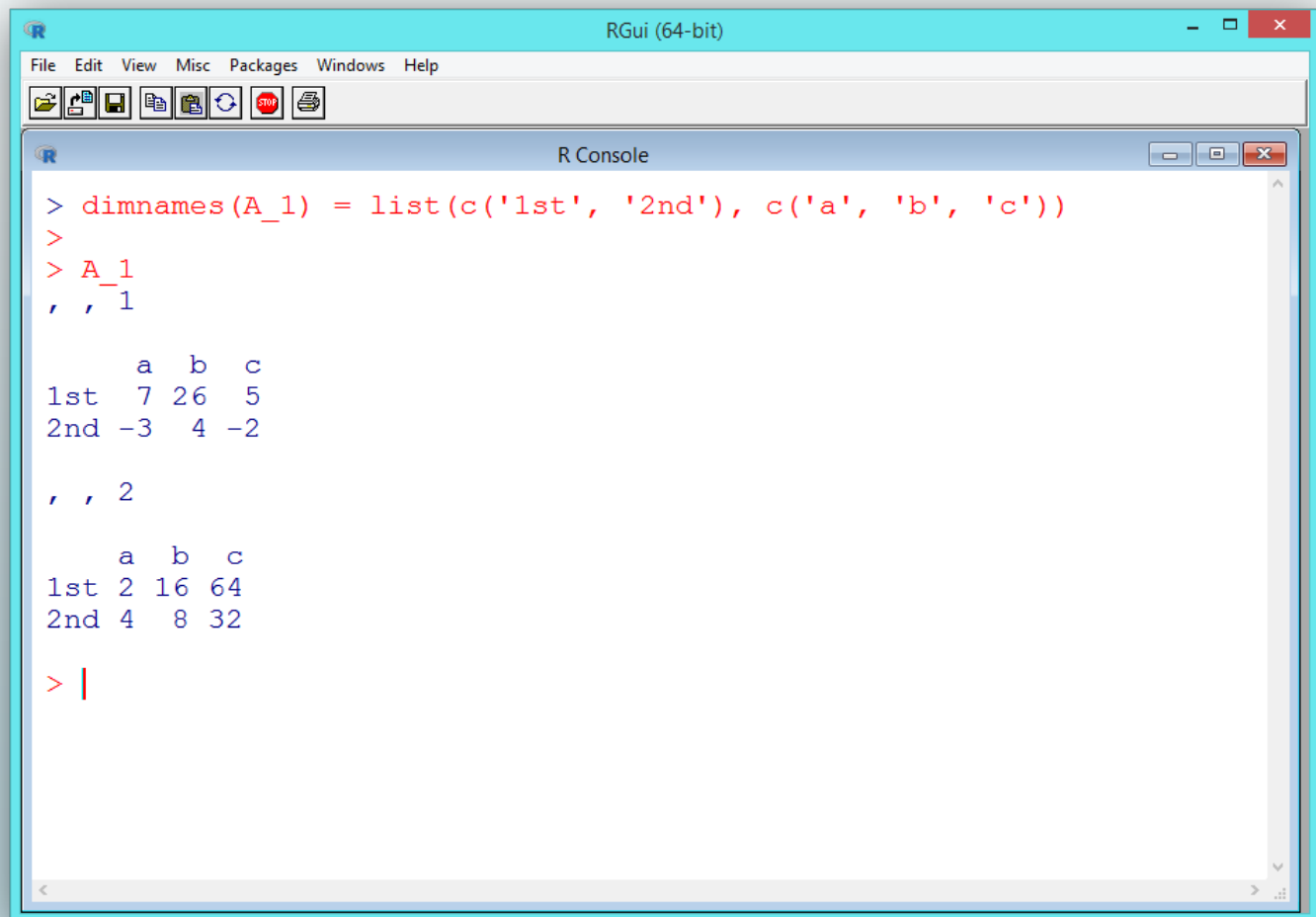


```
> A_2
, , 1
     [,1] [,2] [,3] [,4]
[1,]    7   -3   26    4
, , 2
     [,1] [,2] [,3] [,4]
[1,]    5   -2    2    4
, , 3
     [,1] [,2] [,3] [,4]
[1,]   16    8   64   32

> dim(A_2)
[1] 1 4 3
> |
```

Array

► Naming the Array Dimensions



```
> dimnames(A_1) = list(c('1st', '2nd'), c('a', 'b', 'c'))
>
> A_1
, , 1

      a  b  c
1st  7 26  5
2nd -3  4 -2

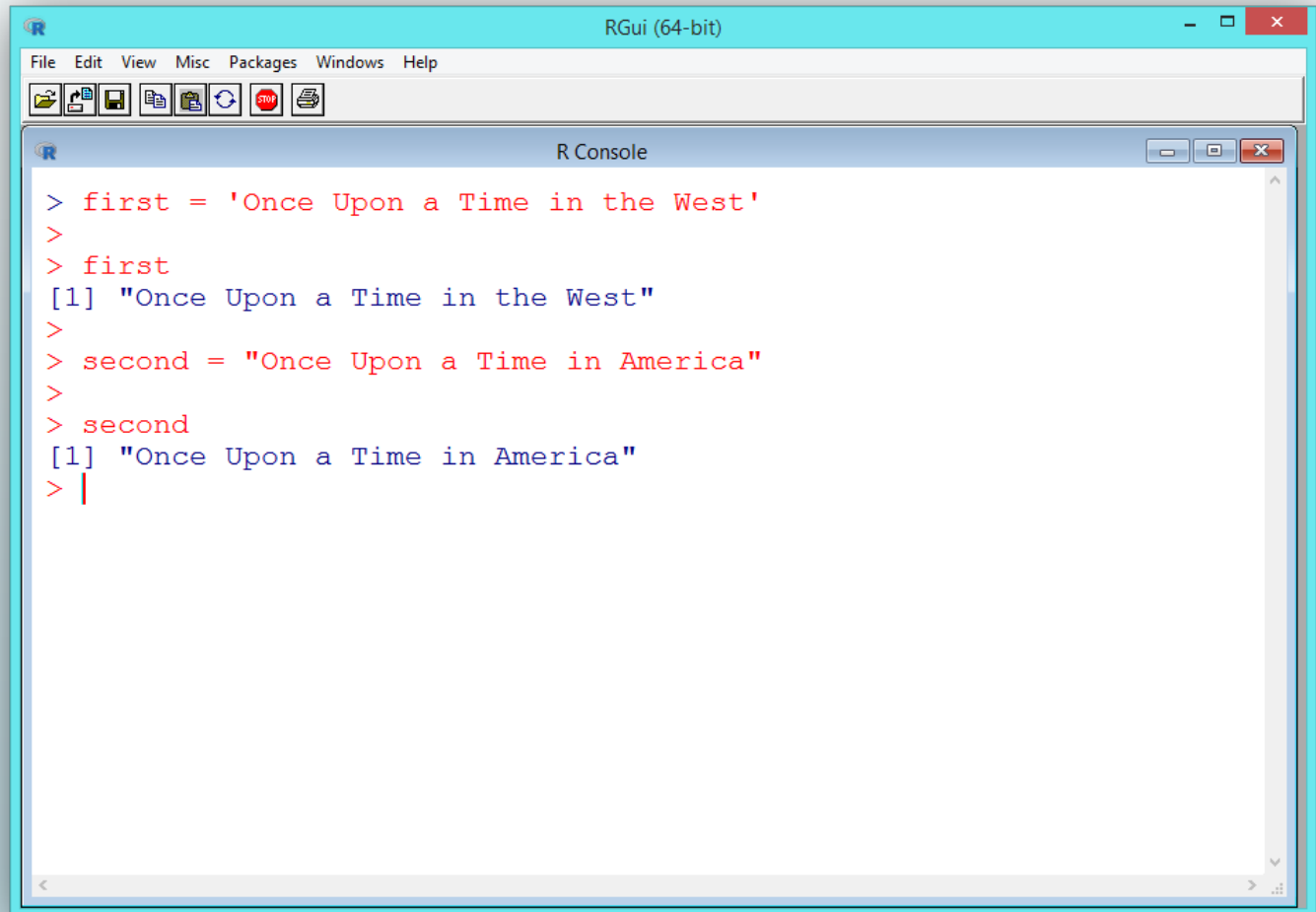
, , 2

      a  b  c
1st  2 16 64
2nd  4  8 32

> |
```

String

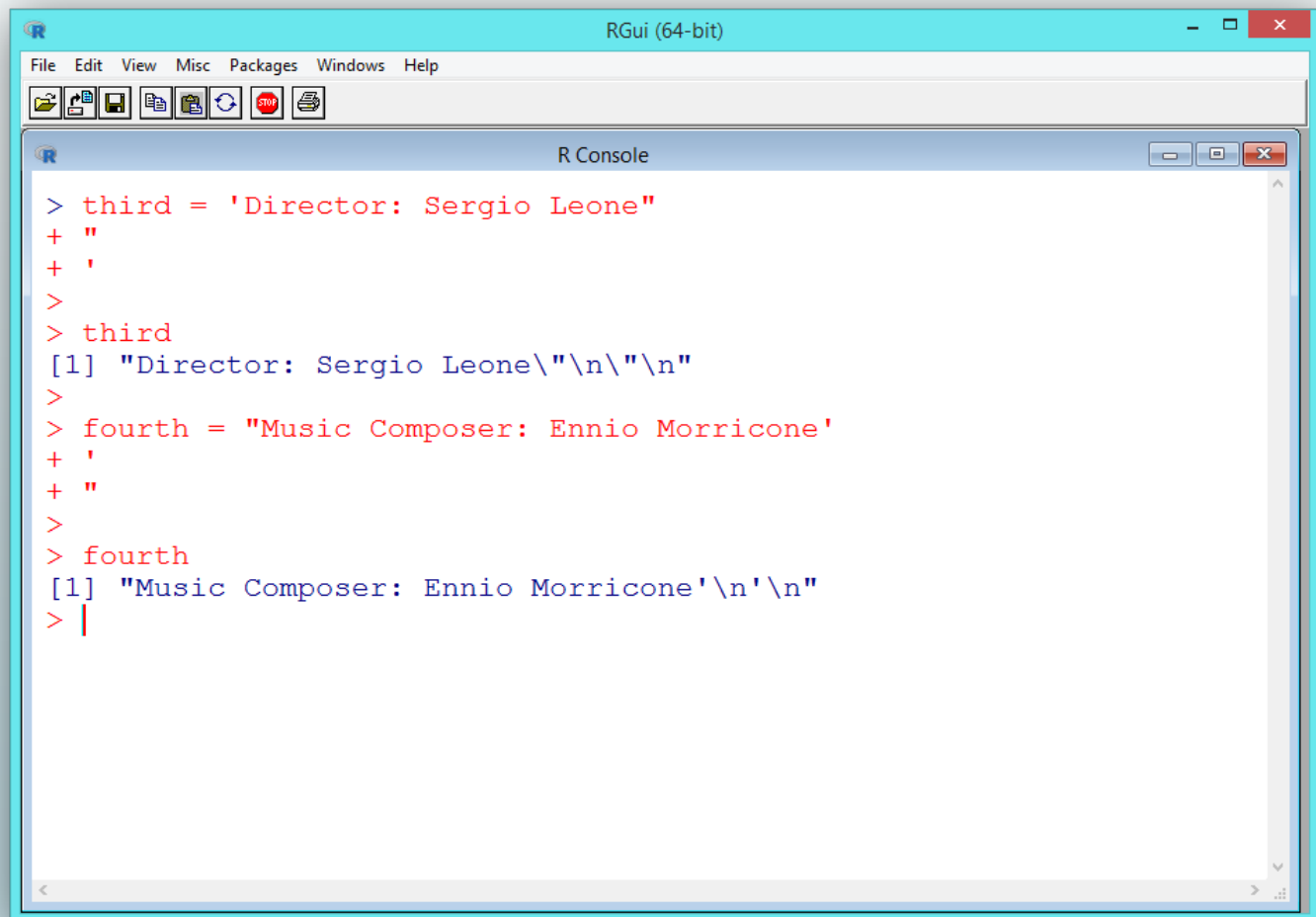
► Assignment (1)



```
> first = 'Once Upon a Time in the West'
>
> first
[1] "Once Upon a Time in the West"
>
> second = "Once Upon a Time in America"
>
> second
[1] "Once Upon a Time in America"
> |
```

String

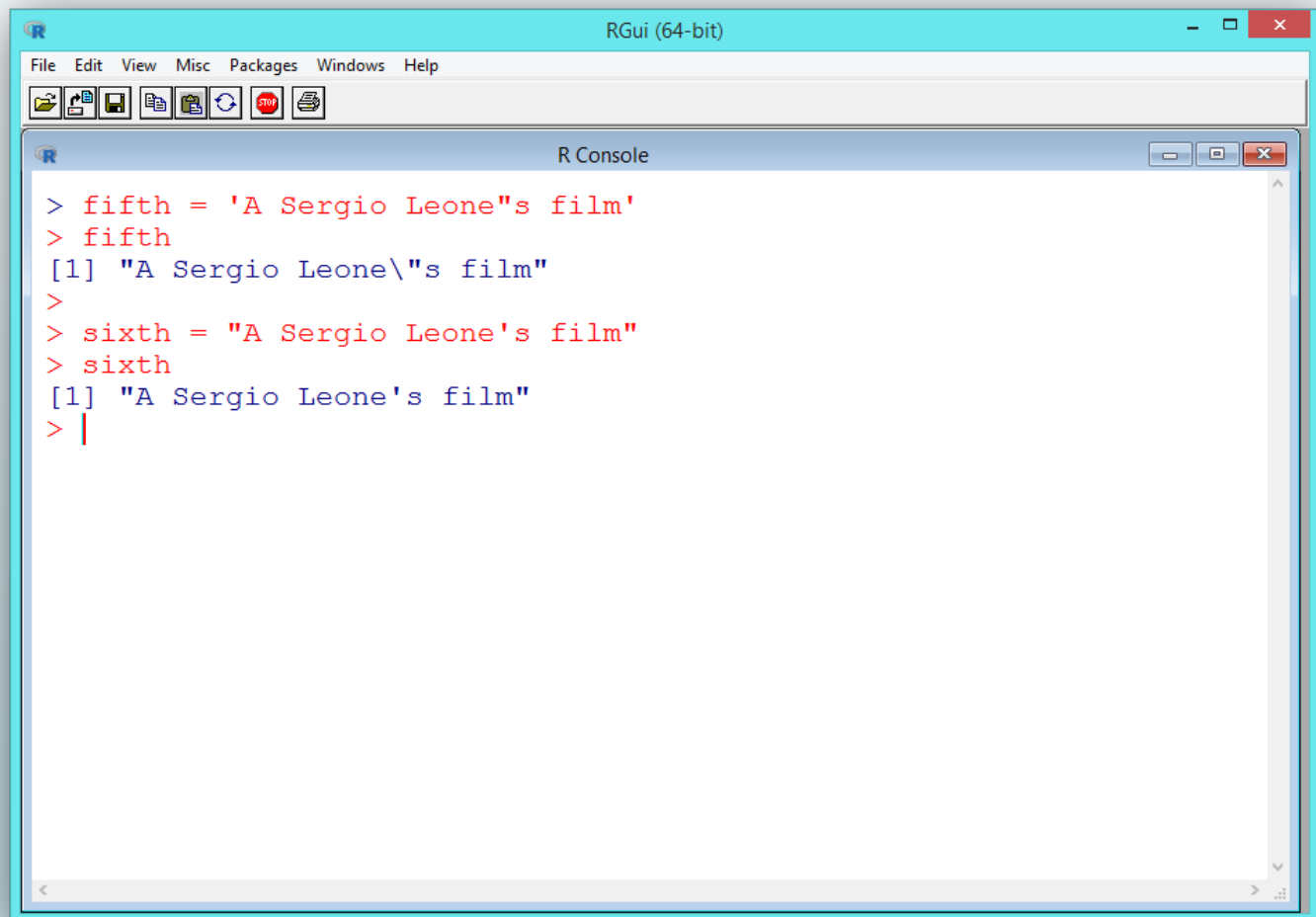
► Assignment (2)



```
> third = 'Director: Sergio Leone'
+ "
+ '
>
> third
[1] "Director: Sergio Leone\\n\\n"
>
> fourth = "Music Composer: Ennio Morricone"
+ '
+ "
>
> fourth
[1] "Music Composer: Ennio Morricone'\n'\n"
> |
```

String

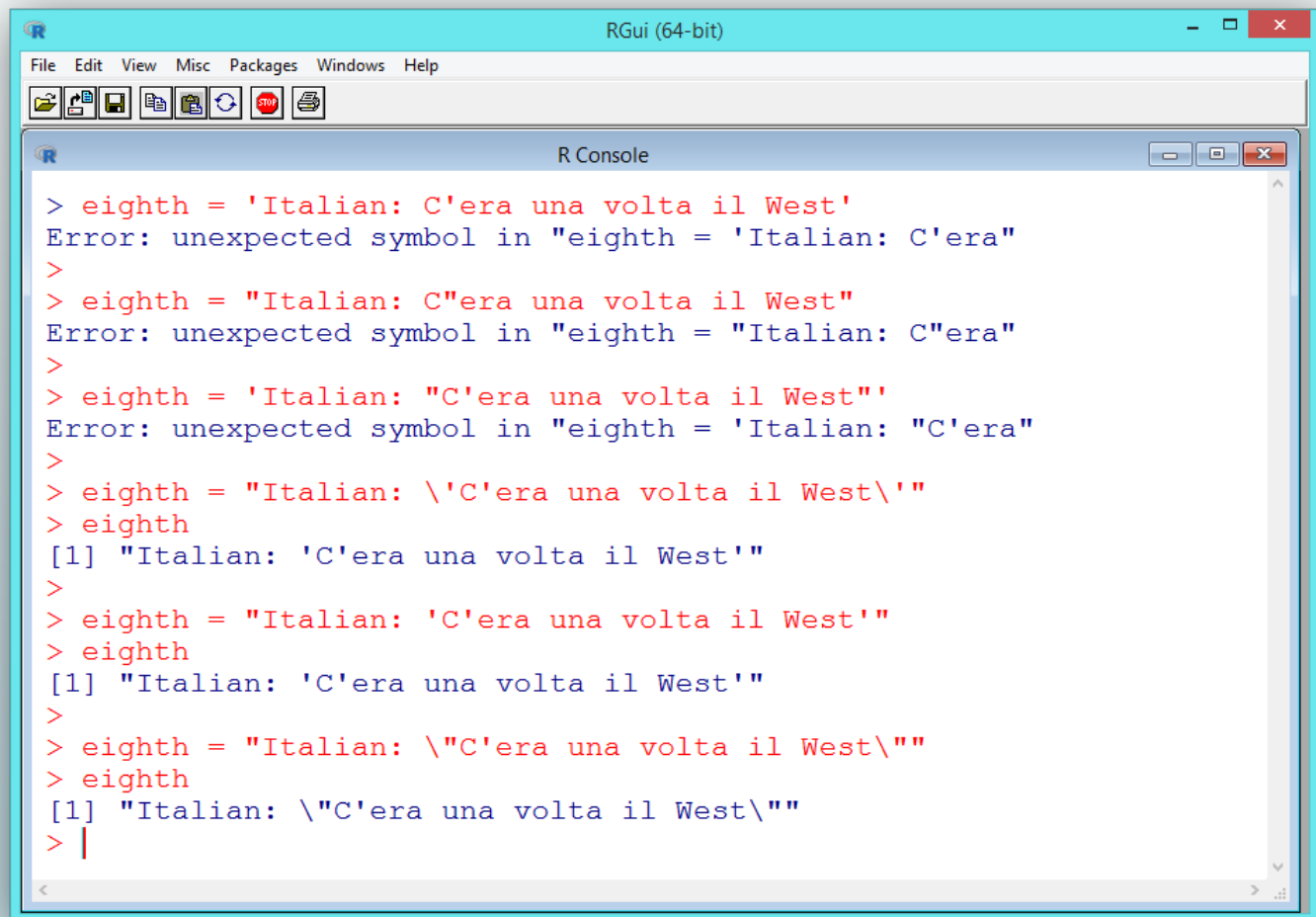
► Assignment (3)



```
> fifth = 'A Sergio Leone"s film'
> fifth
[1] "A Sergio Leone\"s film"
>
> sixth = "A Sergio Leone's film"
> sixth
[1] "A Sergio Leone's film"
> |
```


String

► Assignment (4)

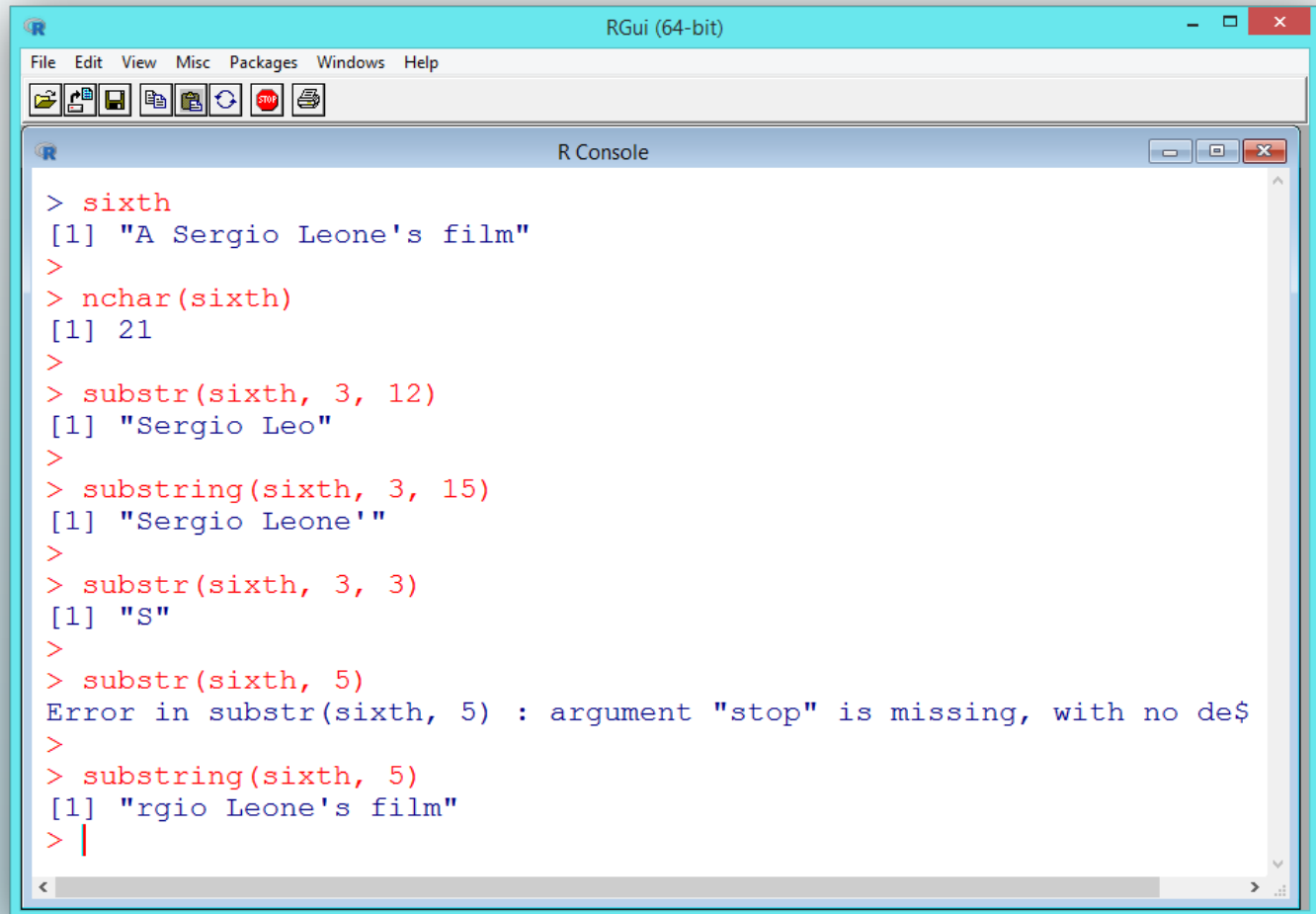


```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> eighth = 'Italian: C'era una volta il West'
Error: unexpected symbol in "eighth = 'Italian: C'era"
>
> eighth = "Italian: C"era una volta il West"
Error: unexpected symbol in "eighth = "Italian: C"era"
>
> eighth = 'Italian: "C'era una volta il West"'
Error: unexpected symbol in "eighth = 'Italian: "C'era"
>
> eighth = "Italian: \'C'era una volta il West\'"
> eighth
[1] "Italian: 'C'era una volta il West'"
>
> eighth = "Italian: 'C'era una volta il West'"
> eighth
[1] "Italian: 'C'era una volta il West'"
>
> eighth = "Italian: \"C'era una volta il West\""
> eighth
[1] "Italian: \"C'era una volta il West\""
> |
```

String

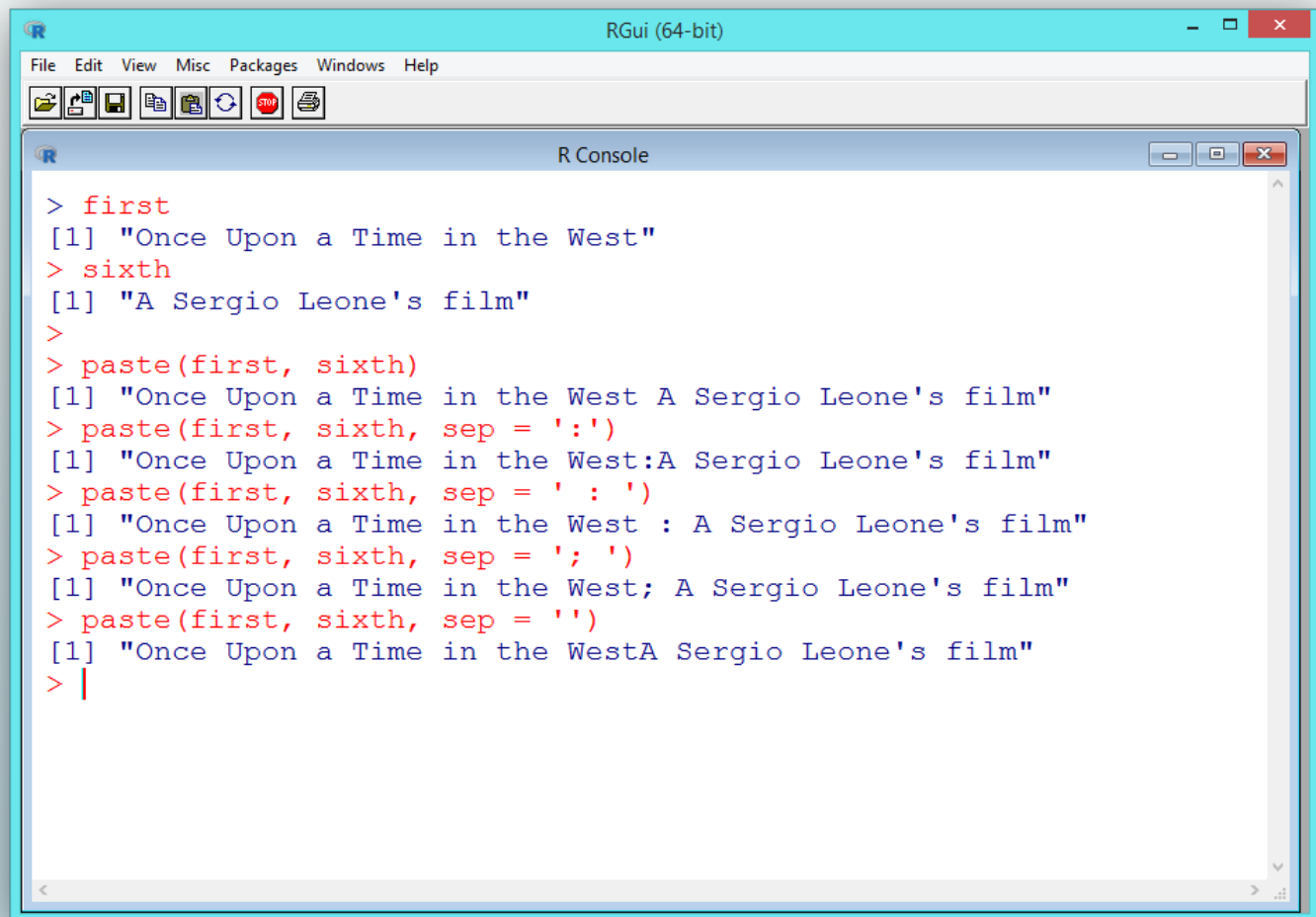
► Substrings



```
> sixth
[1] "A Sergio Leone's film"
>
> nchar(sixth)
[1] 21
>
> substr(sixth, 3, 12)
[1] "Sergio Leo"
>
> substring(sixth, 3, 15)
[1] "Sergio Leone'"
>
> substr(sixth, 3, 3)
[1] "S"
>
> substr(sixth, 5)
Error in substr(sixth, 5) : argument "stop" is missing, with no de$
>
> substring(sixth, 5)
[1] "rgio Leone's film"
> |
```

String

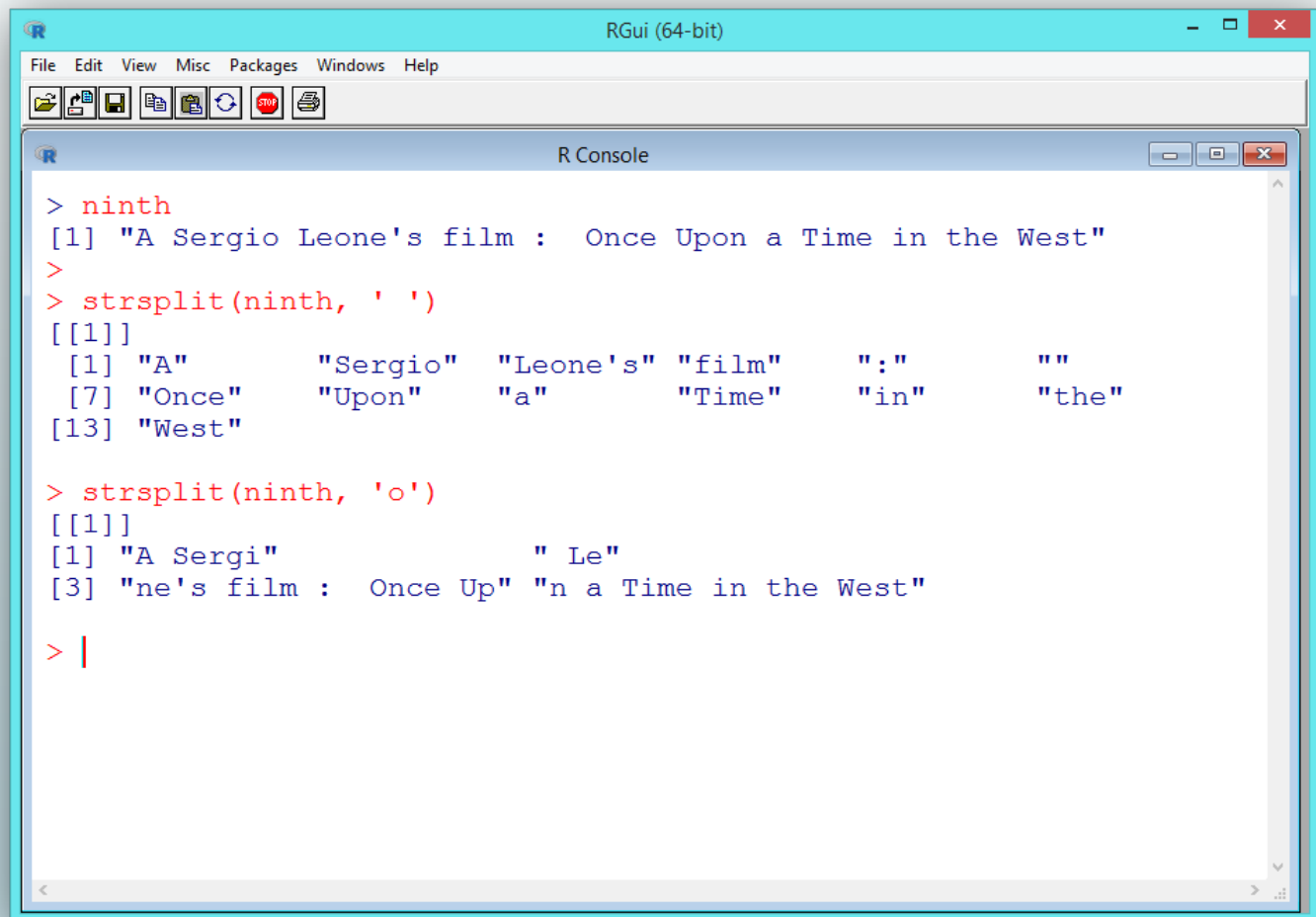
► Pasting Strings



```
> first
[1] "Once Upon a Time in the West"
> sixth
[1] "A Sergio Leone's film"
>
> paste(first, sixth)
[1] "Once Upon a Time in the West A Sergio Leone's film"
> paste(first, sixth, sep = ':')
[1] "Once Upon a Time in the West:A Sergio Leone's film"
> paste(first, sixth, sep = ' : ')
[1] "Once Upon a Time in the West : A Sergio Leone's film"
> paste(first, sixth, sep = '; ')
[1] "Once Upon a Time in the West; A Sergio Leone's film"
> paste(first, sixth, sep = '')
[1] "Once Upon a Time in the WestA Sergio Leone's film"
> |
```

String

► Splitting a String



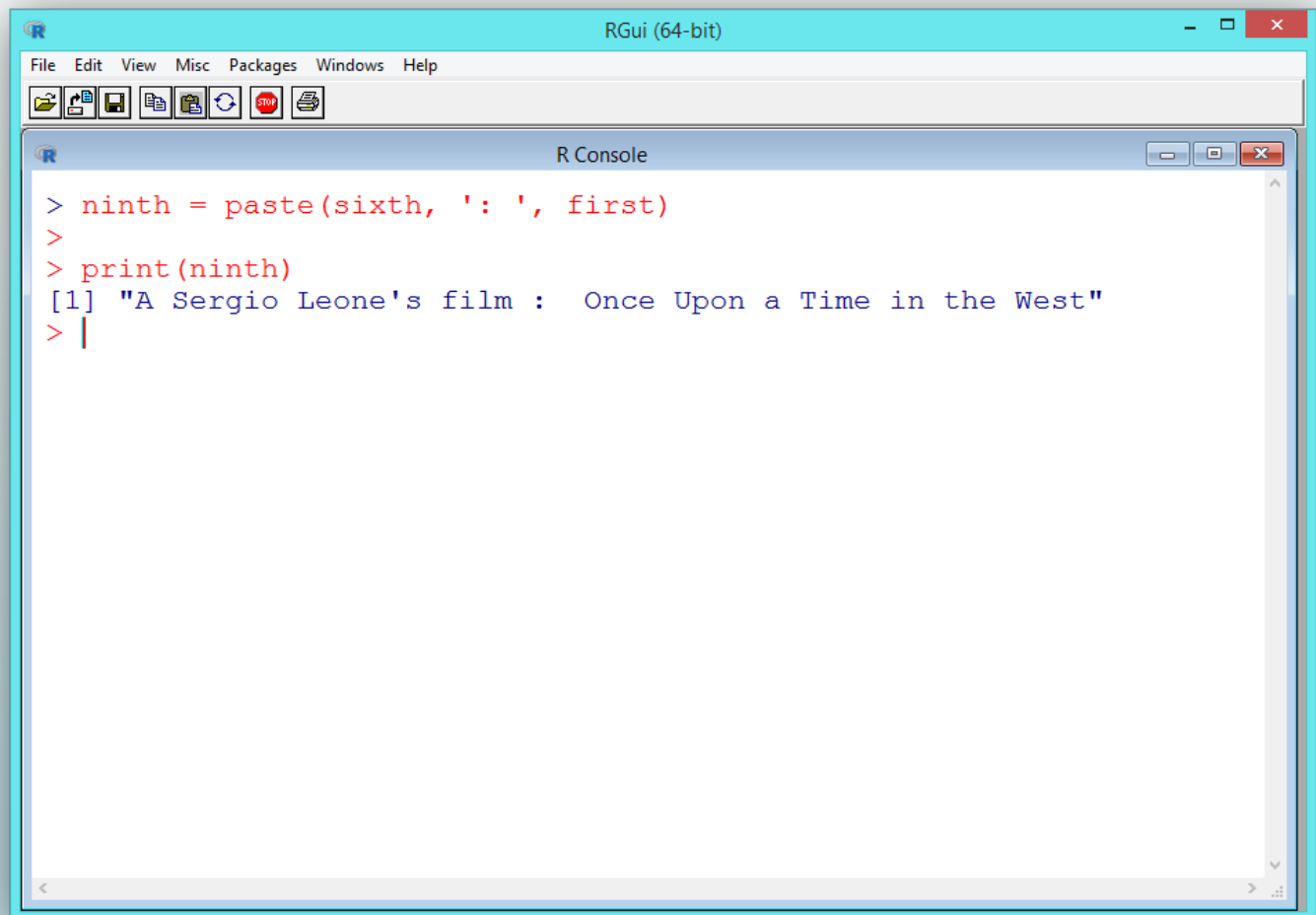
```
> ninth
[1] "A Sergio Leone's film : Once Upon a Time in the West"
>
> strsplit(ninth, ' ')
[[1]]
 [1] "A"      "Sergio"  "Leone's" "film"    ":"      ""
 [7] "Once"   "Upon"    "a"        "Time"   "in"     "the"
[13] "West"

> strsplit(ninth, 'o')
[[1]]
 [1] "A Sergi"          " Le"
 [3] "ne's film : Once Up" "n a Time in the West"

> |
```

String

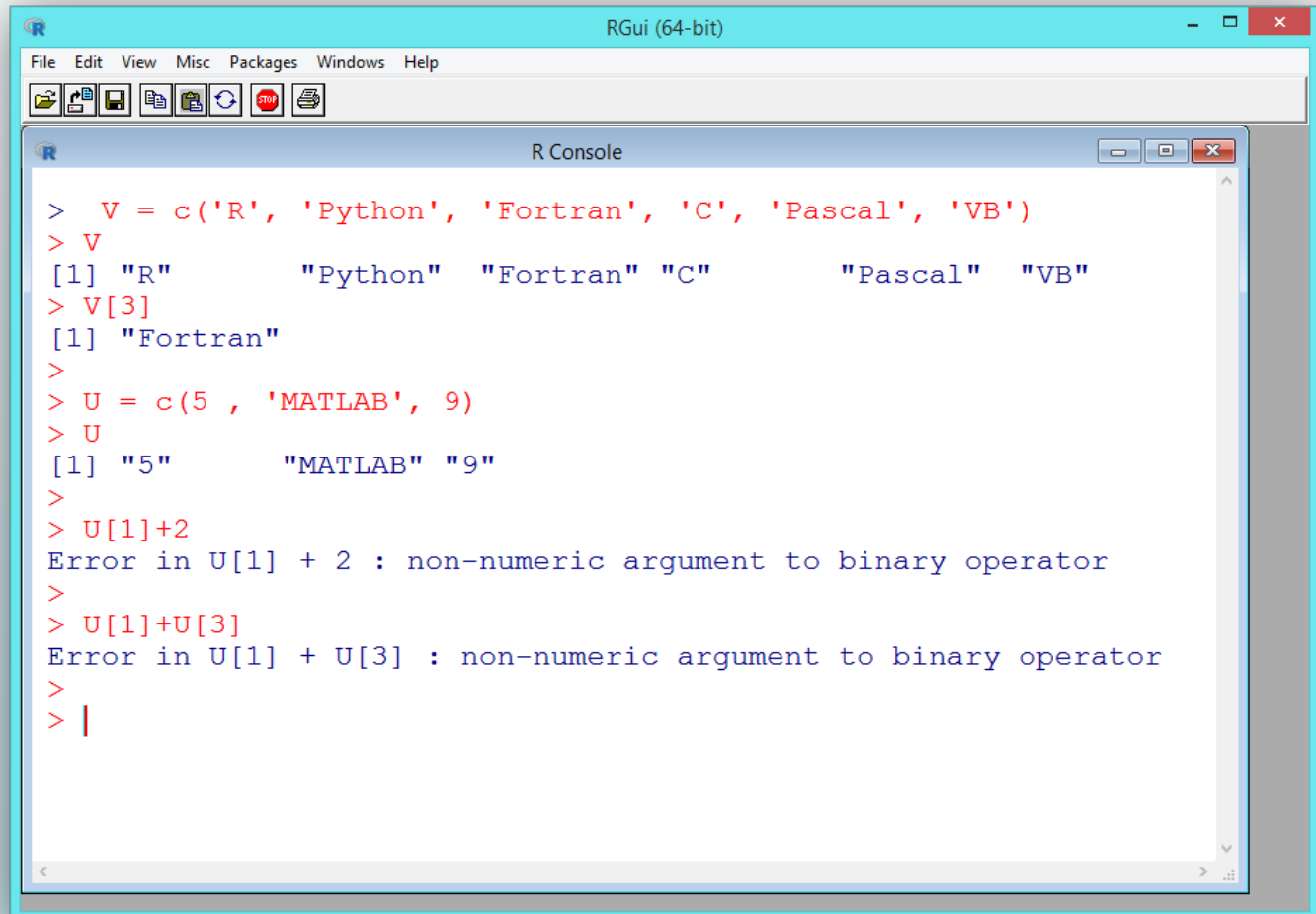
► Printing a String



```
> ninth = paste(sixth, ': ', first)
>
> print(ninth)
[1] "A Sergio Leone's film : Once Upon a Time in the West"
> |
```

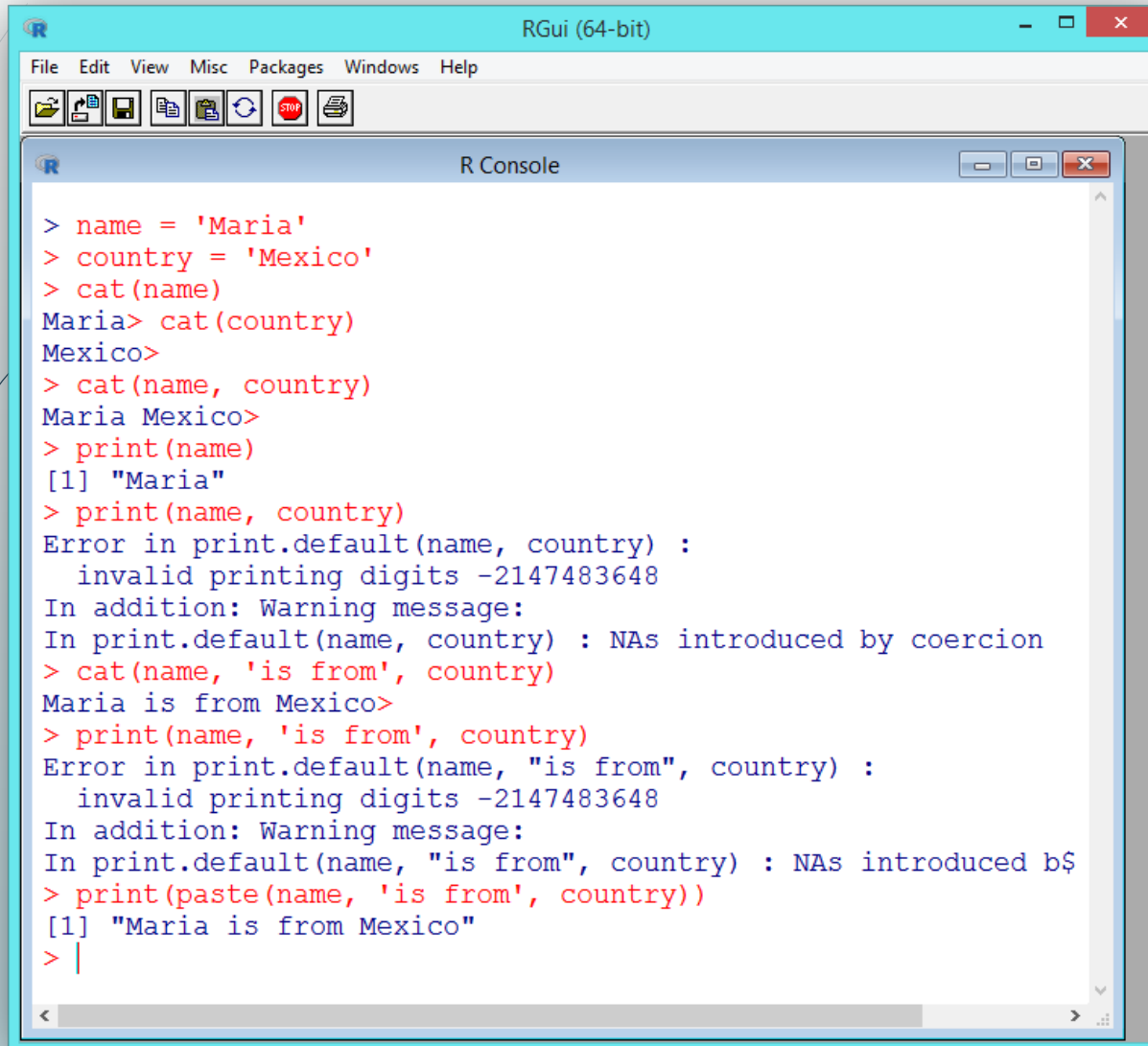
String

► String Vector



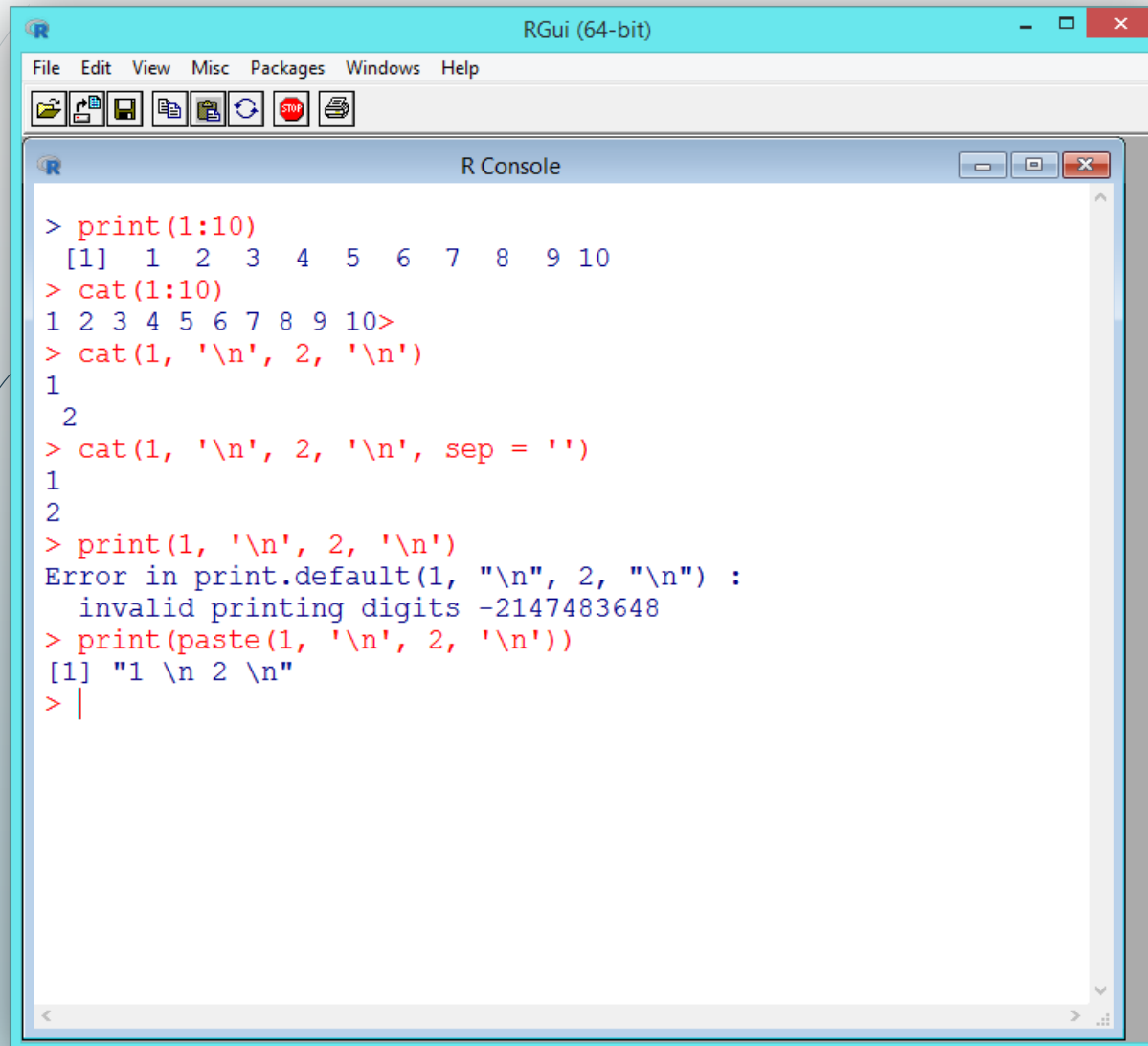
```
> V = c('R', 'Python', 'Fortran', 'C', 'Pascal', 'VB')
> V
[1] "R"          "Python"    "Fortran"   "C"         "Pascal"    "VB"
> V[3]
[1] "Fortran"
>
> U = c(5, 'MATLAB', 9)
> U
[1] "5"          "MATLAB"    "9"
>
> U[1]+2
Error in U[1] + 2 : non-numeric argument to binary operator
>
> U[1]+U[3]
Error in U[1] + U[3] : non-numeric argument to binary operator
>
> |
```

cat and print (1)



```
> name = 'Maria'
> country = 'Mexico'
> cat(name)
Maria> cat(country)
Mexico>
> cat(name, country)
Maria Mexico>
> print(name)
[1] "Maria"
> print(name, country)
Error in print.default(name, country) :
  invalid printing digits -2147483648
In addition: Warning message:
In print.default(name, country) : NAs introduced by coercion
> cat(name, 'is from', country)
Maria is from Mexico>
> print(name, 'is from', country)
Error in print.default(name, "is from", country) :
  invalid printing digits -2147483648
In addition: Warning message:
In print.default(name, "is from", country) : NAs introduced by coercion
> print(paste(name, 'is from', country))
[1] "Maria is from Mexico"
> |
```

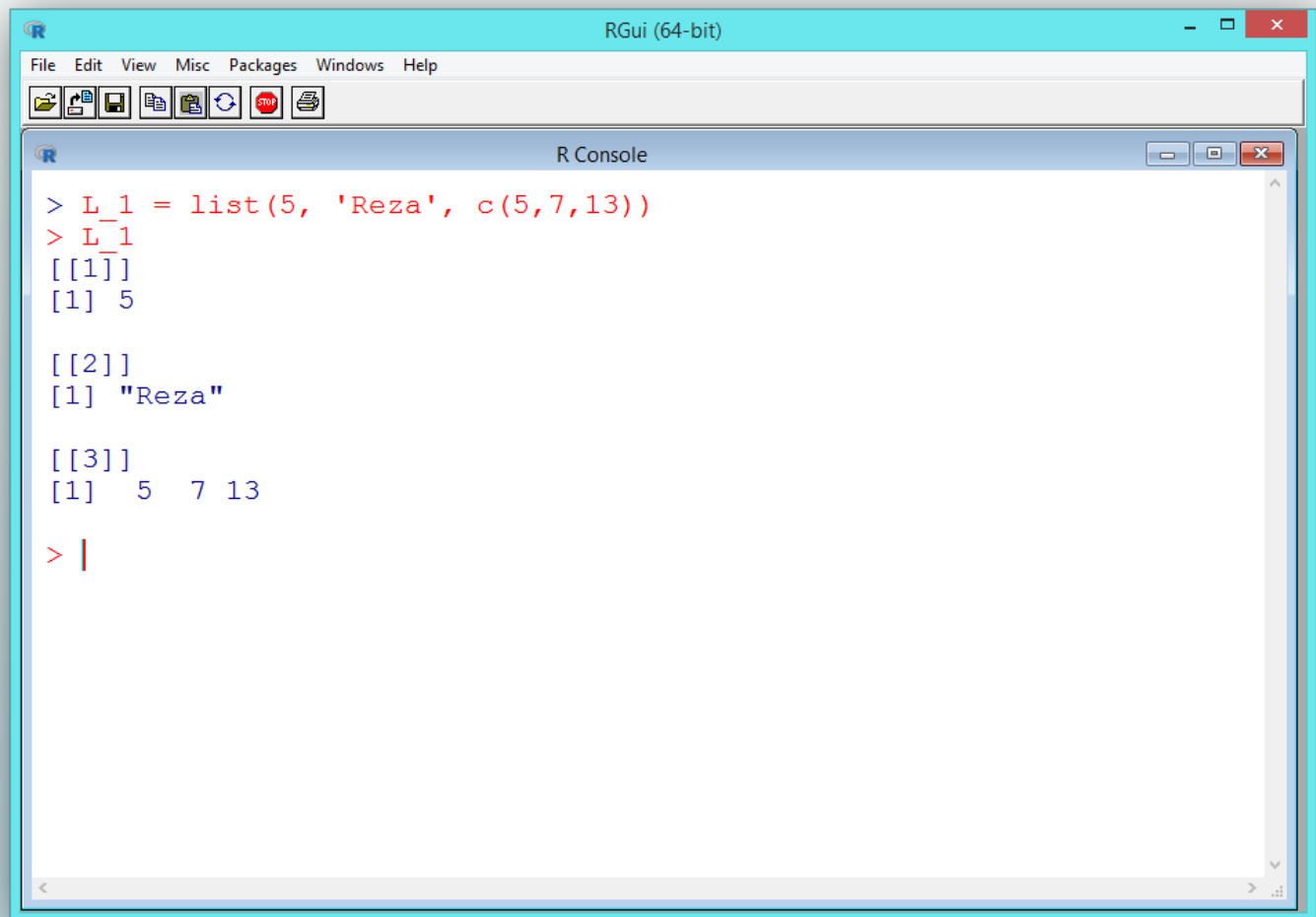
cat and print (2)



```
> print(1:10)
[1] 1 2 3 4 5 6 7 8 9 10
> cat(1:10)
1 2 3 4 5 6 7 8 9 10>
> cat(1, '\n', 2, '\n')
1
2
> cat(1, '\n', 2, '\n', sep = '')
1
2
> print(1, '\n', 2, '\n')
Error in print.default(1, "\n", 2, "\n") :
  invalid printing digits -2147483648
> print(paste(1, '\n', 2, '\n'))
[1] "1 \n 2 \n"
> |
```


List

► Assignment (1)



```
> L_1 = list(5, 'Reza', c(5,7,13))
> L_1
[[1]]
[1] 5

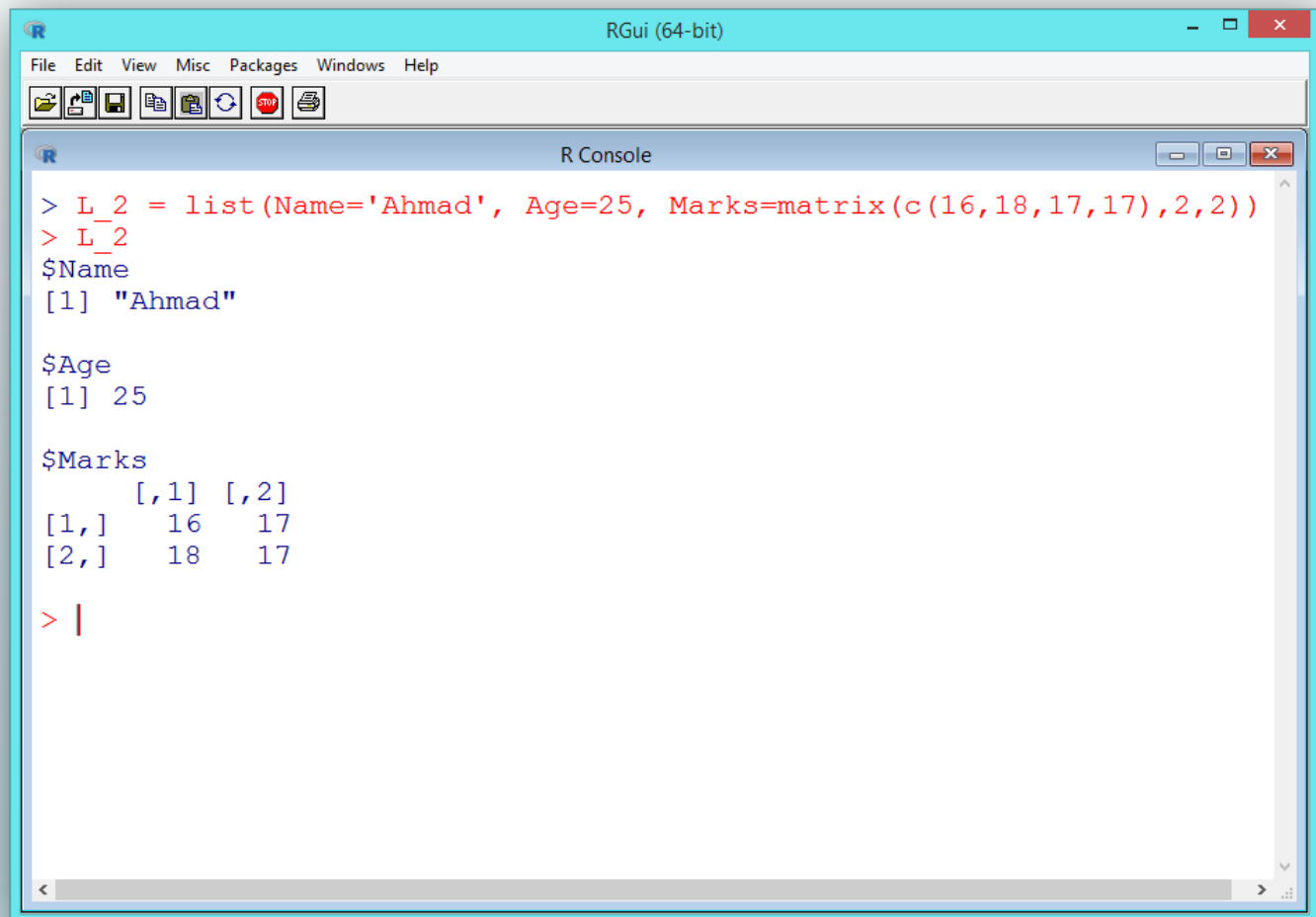
[[2]]
[1] "Reza"

[[3]]
[1] 5 7 13

> |
```

List

► Assignment (2)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> L_2 = list(Name='Ahmad', Age=25, Marks=matrix(c(16,18,17,17),2,2))
> L_2
$Name
[1] "Ahmad"

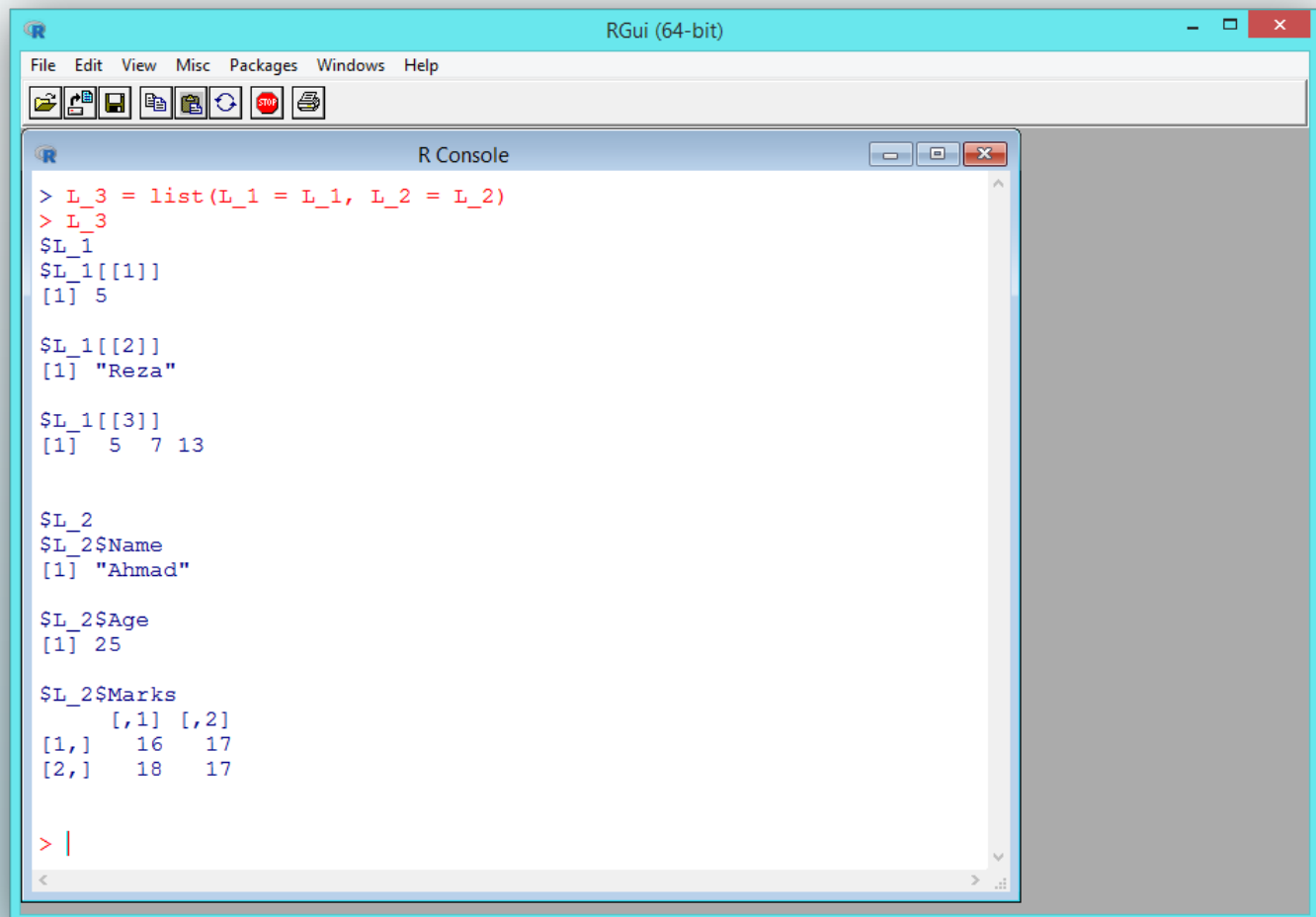
$Age
[1] 25

$Marks
      [,1] [,2]
[1,]   16   17
[2,]   18   17

> |
```

List

► Assignment (3)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> L_3 = list(L_1 = L_1, L_2 = L_2)
> L_3
$L_1
$L_1[[1]]
[1] 5

$L_1[[2]]
[1] "Reza"

$L_1[[3]]
[1] 5 7 13

$L_2
$L_2$Name
[1] "Ahmad"

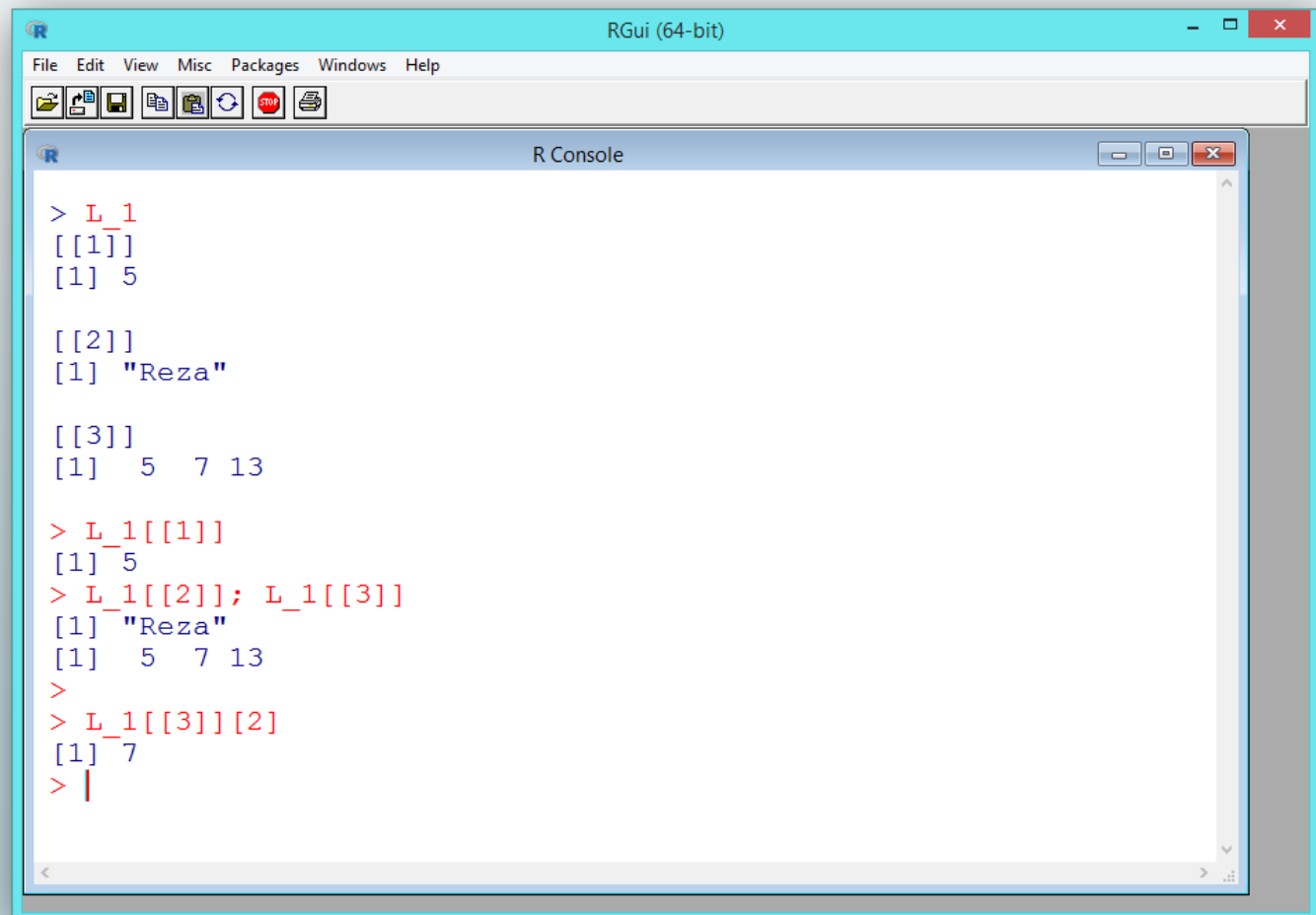
$L_2$Age
[1] 25

$L_2$Marks
      [,1] [,2]
[1,]  16  17
[2,]  18  17

> |
```

List

➤ Addressing the List Members (1)



```
> L_1
[[1]]
[1] 5

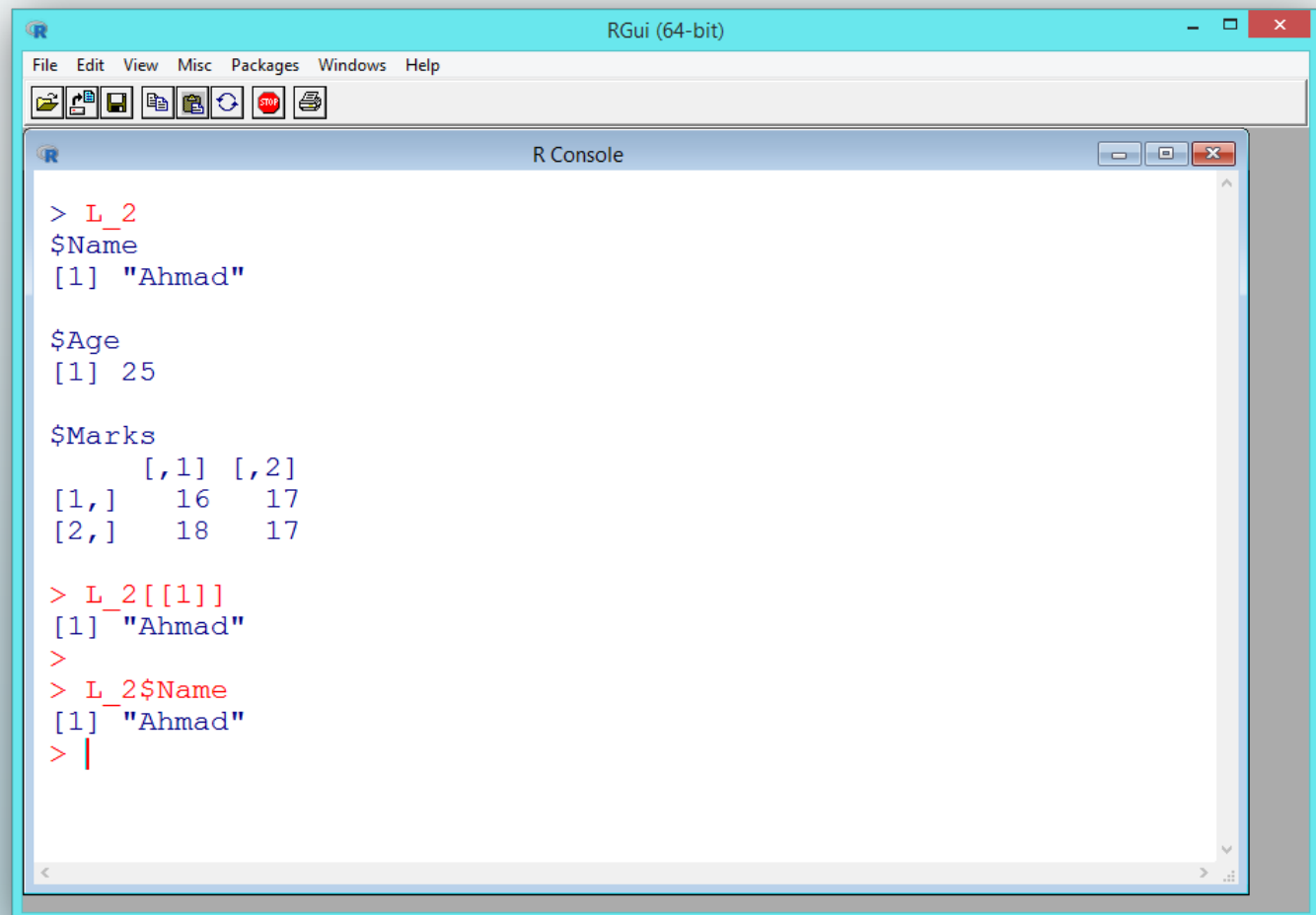
[[2]]
[1] "Reza"

[[3]]
[1] 5 7 13

> L_1[[1]]
[1] 5
> L_1[[2]]; L_1[[3]]
[1] "Reza"
[1] 5 7 13
>
> L_1[[3]][2]
[1] 7
> |
```

List

➤ Addressing the List Members (2)



```
> L_2
$Name
[1] "Ahmad"

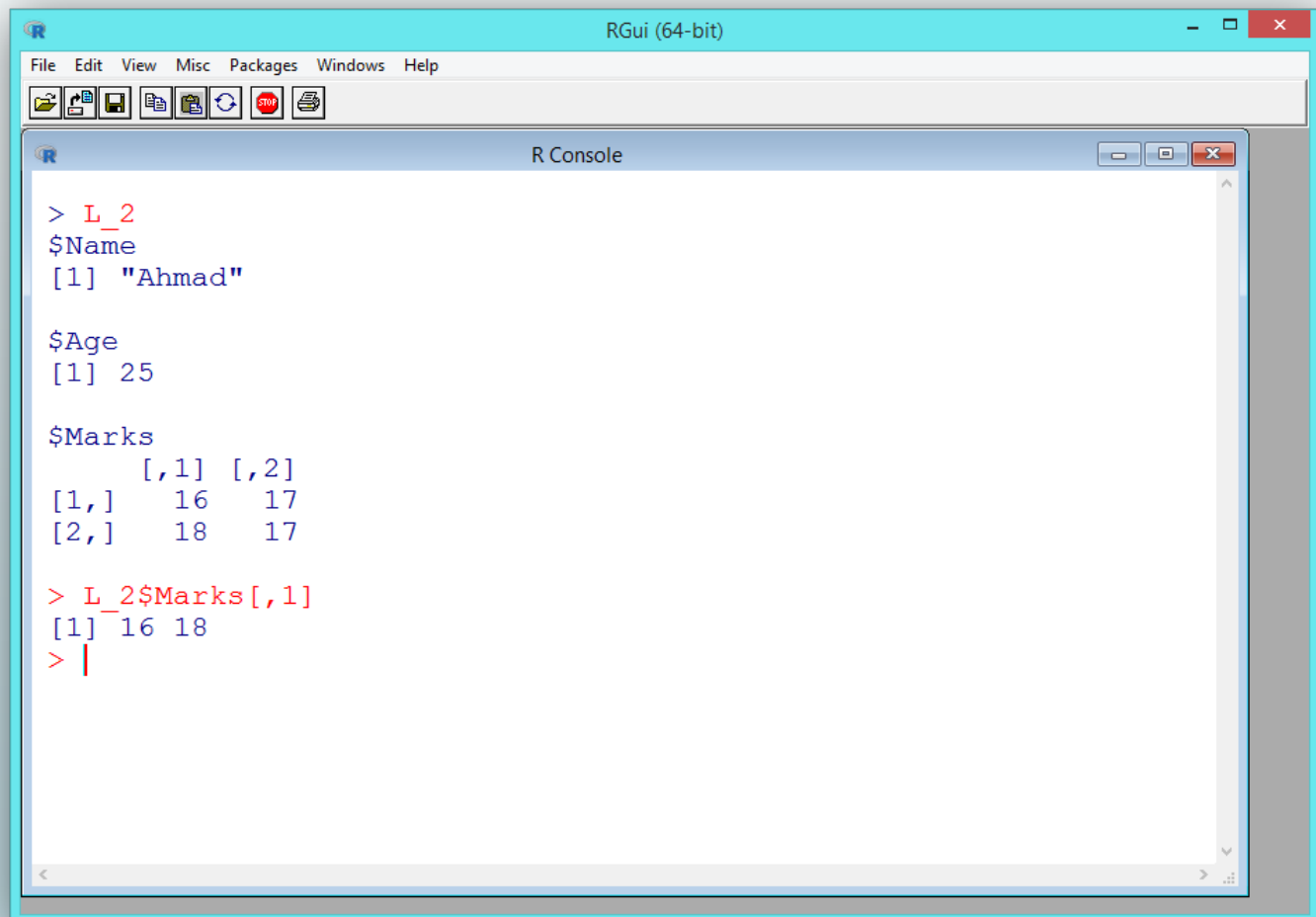
$Age
[1] 25

$Marks
      [,1] [,2]
[1,]  16  17
[2,]  18  17

> L_2[[1]]
[1] "Ahmad"
>
> L_2$name
[1] "Ahmad"
> |
```

List

➤ Addressing the List Members (3)



```
> L_2
$Name
[1] "Ahmad"

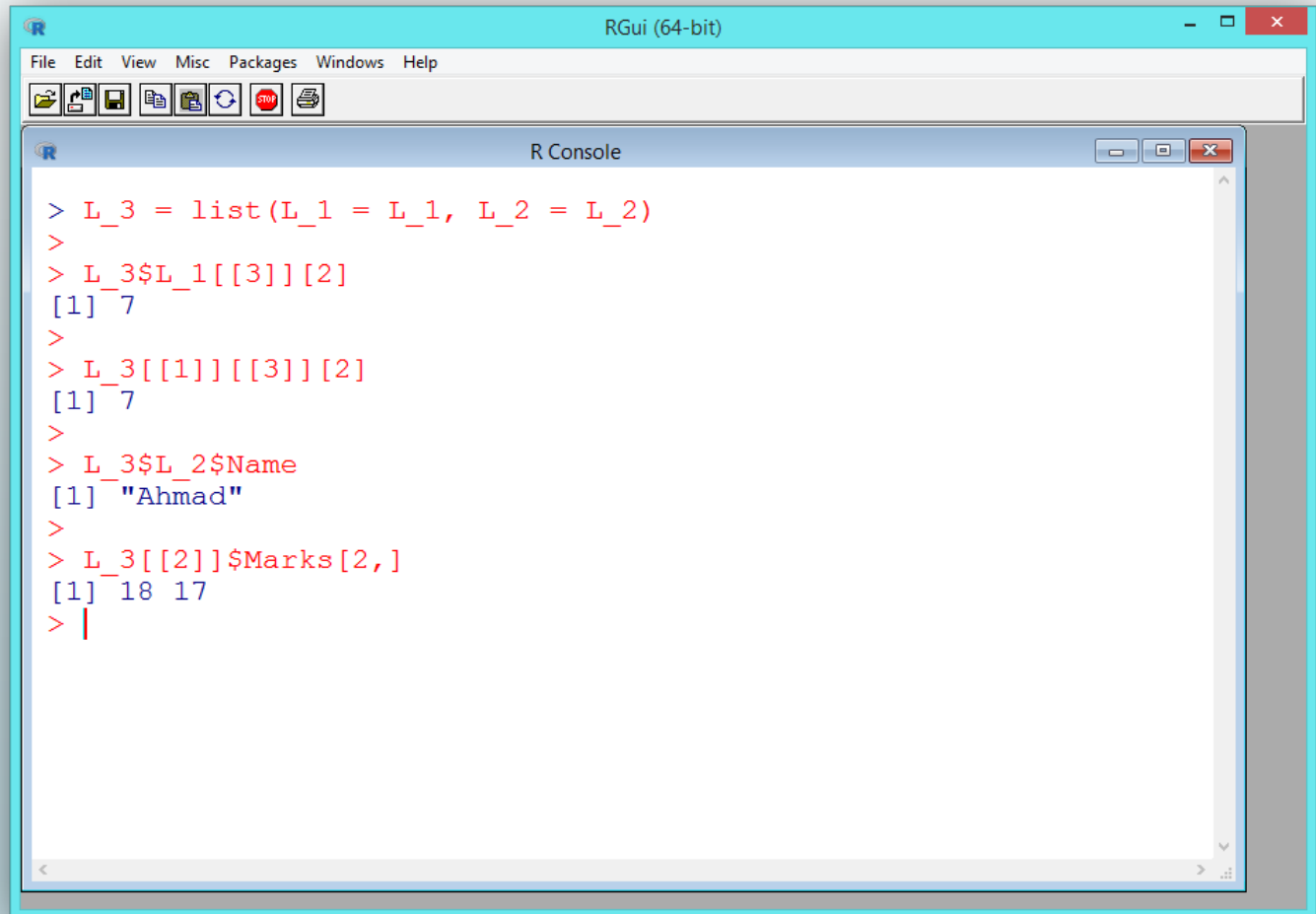
$Age
[1] 25

$Marks
  [,1] [,2]
[1,]  16  17
[2,]  18  17

> L_2$Marks[,1]
[1] 16 18
> |
```

List

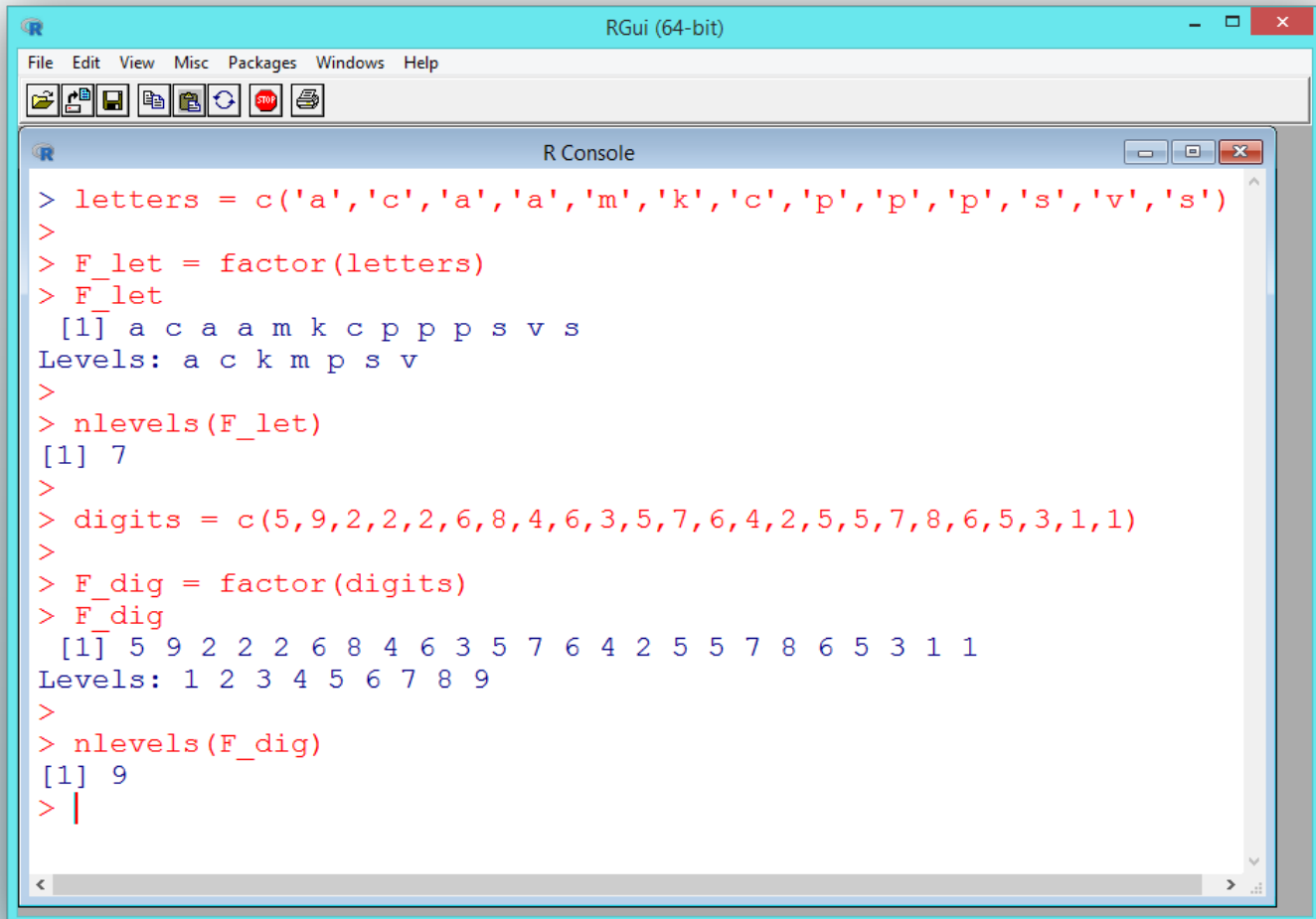
➤ Addressing the List Members (4)



```
> L_3 = list(L_1 = L_1, L_2 = L_2)
>
> L_3$L_1[[3]][2]
[1] 7
>
> L_3[[1]][[3]][2]
[1] 7
>
> L_3$L_2$Name
[1] "Ahmad"
>
> L_3[[2]]$Marks[2,]
[1] 18 17
> |
```

Factor

► Assignment

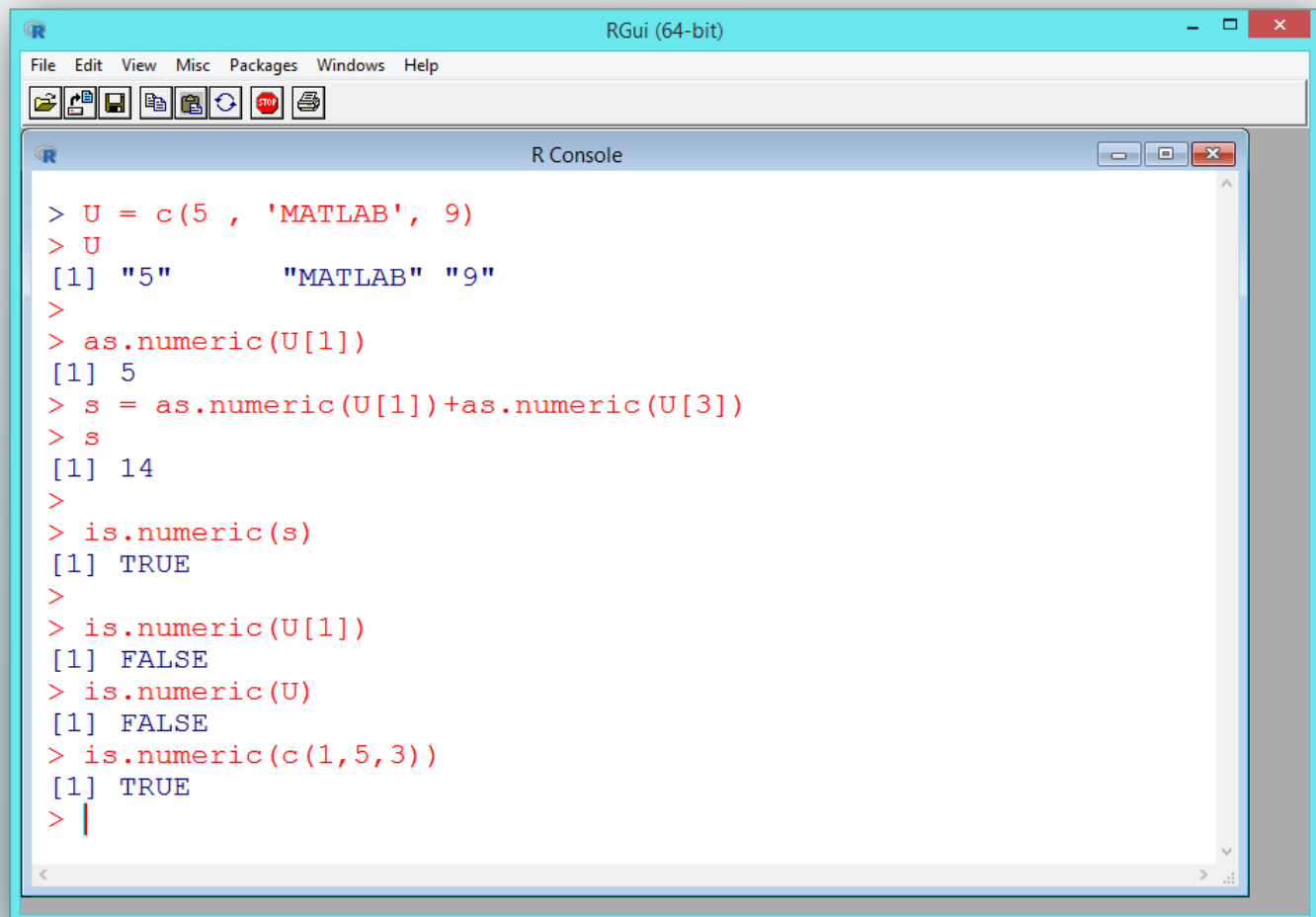


```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> letters = c('a','c','a','a','m','k','c','p','p','p','s','v','s')
>
> F_let = factor(letters)
> F_let
[1] a c a a m k c p p p s v s
Levels: a c k m p s v
>
> nlevels(F_let)
[1] 7
>
> digits = c(5,9,2,2,2,6,8,4,6,3,5,7,6,4,2,5,5,7,8,6,5,3,1,1)
>
> F_dig = factor(digits)
> F_dig
[1] 5 9 2 2 2 6 8 4 6 3 5 7 6 4 2 5 5 7 8 6 5 3 1 1
Levels: 1 2 3 4 5 6 7 8 9
>
> nlevels(F_dig)
[1] 9
> |
```


Identification and Transformation

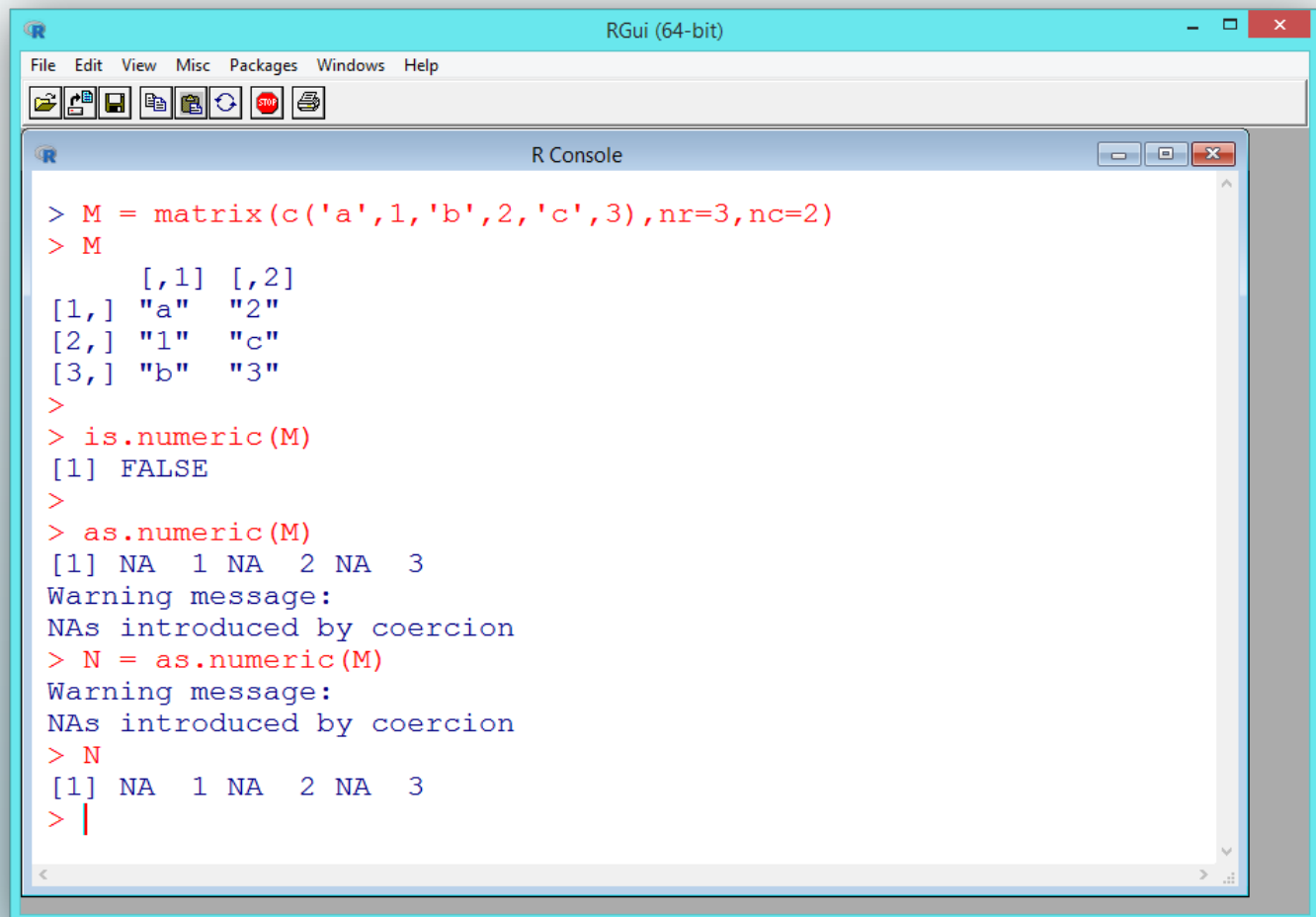
► `as.numeric` and `is.numeric` (1)



```
> U = c(5 , 'MATLAB', 9)
> U
[1] "5"      "MATLAB" "9"
>
> as.numeric(U[1])
[1] 5
> s = as.numeric(U[1])+as.numeric(U[3])
> s
[1] 14
>
> is.numeric(s)
[1] TRUE
>
> is.numeric(U[1])
[1] FALSE
> is.numeric(U)
[1] FALSE
> is.numeric(c(1,5,3))
[1] TRUE
> |
```

Identification and Transformation

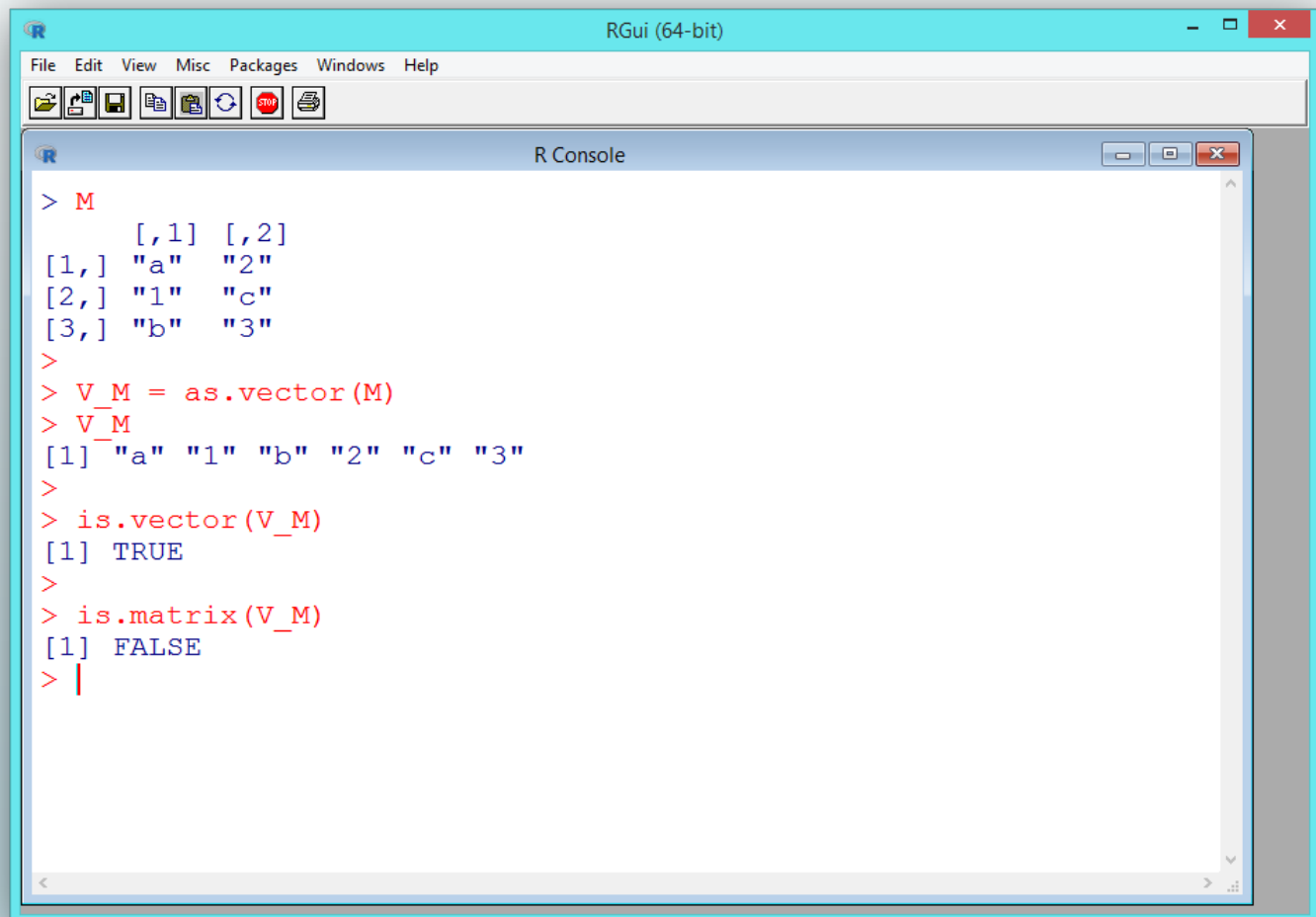
➤ `as.numeric` and `is.numeric` (2)



```
> M = matrix(c('a',1,'b',2,'c',3),nr=3,nc=2)
> M
      [,1] [,2]
[1,] "a"  "2"
[2,] "1"  "c"
[3,] "b"  "3"
>
> is.numeric(M)
[1] FALSE
>
> as.numeric(M)
[1] NA  1 NA  2 NA  3
Warning message:
NAs introduced by coercion
> N = as.numeric(M)
Warning message:
NAs introduced by coercion
> N
[1] NA  1 NA  2 NA  3
> |
```

Identification and Transformation

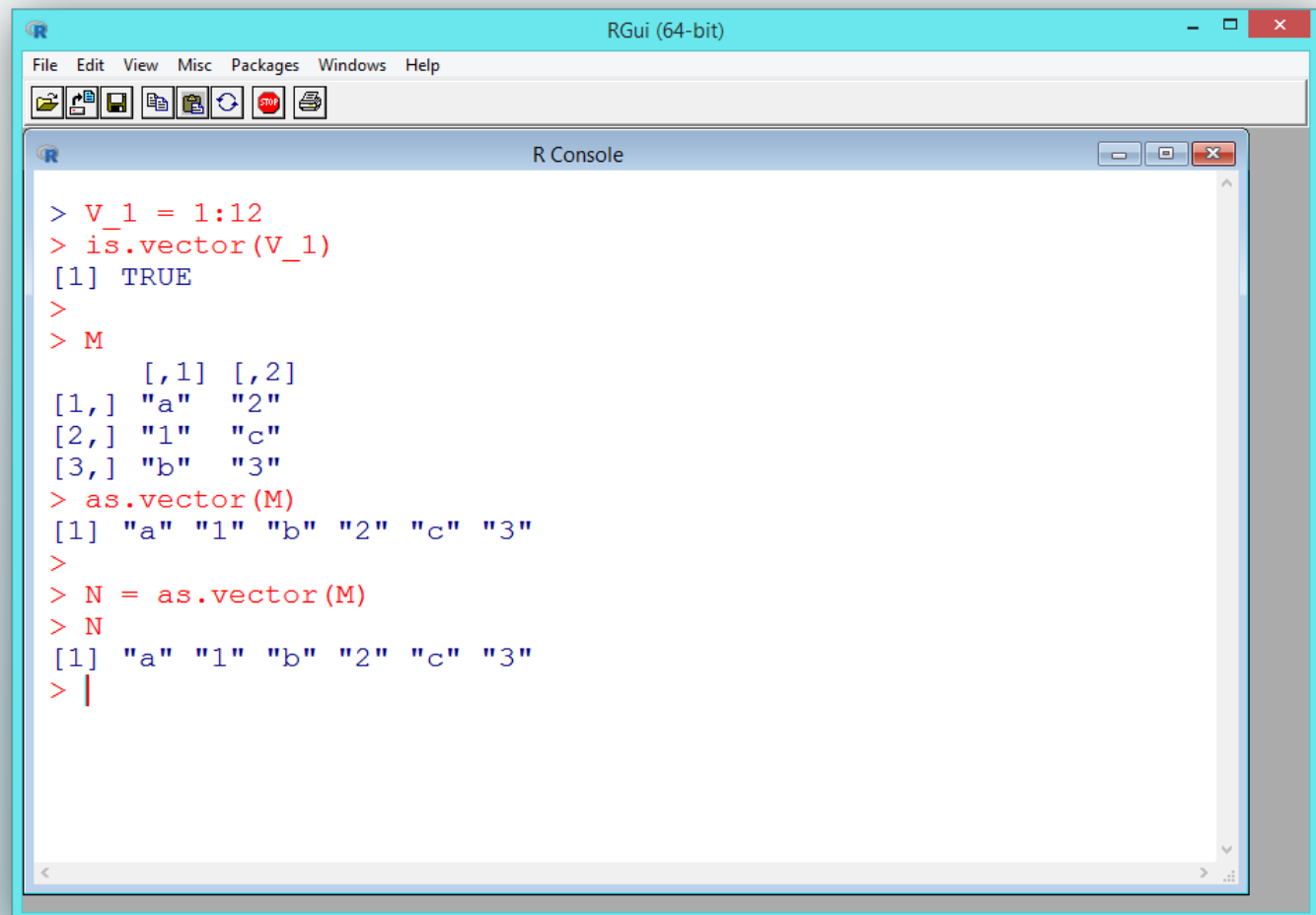
► `as.vector` and `is.vector` (1)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]
R Console
> M
      [,1] [,2]
[1,] "a"  "2"
[2,] "1"  "c"
[3,] "b"  "3"
>
> V_M = as.vector(M)
> V_M
[1] "a" "1" "b" "2" "c" "3"
>
> is.vector(V_M)
[1] TRUE
>
> is.matrix(V_M)
[1] FALSE
> |
```

Identification and Transformation

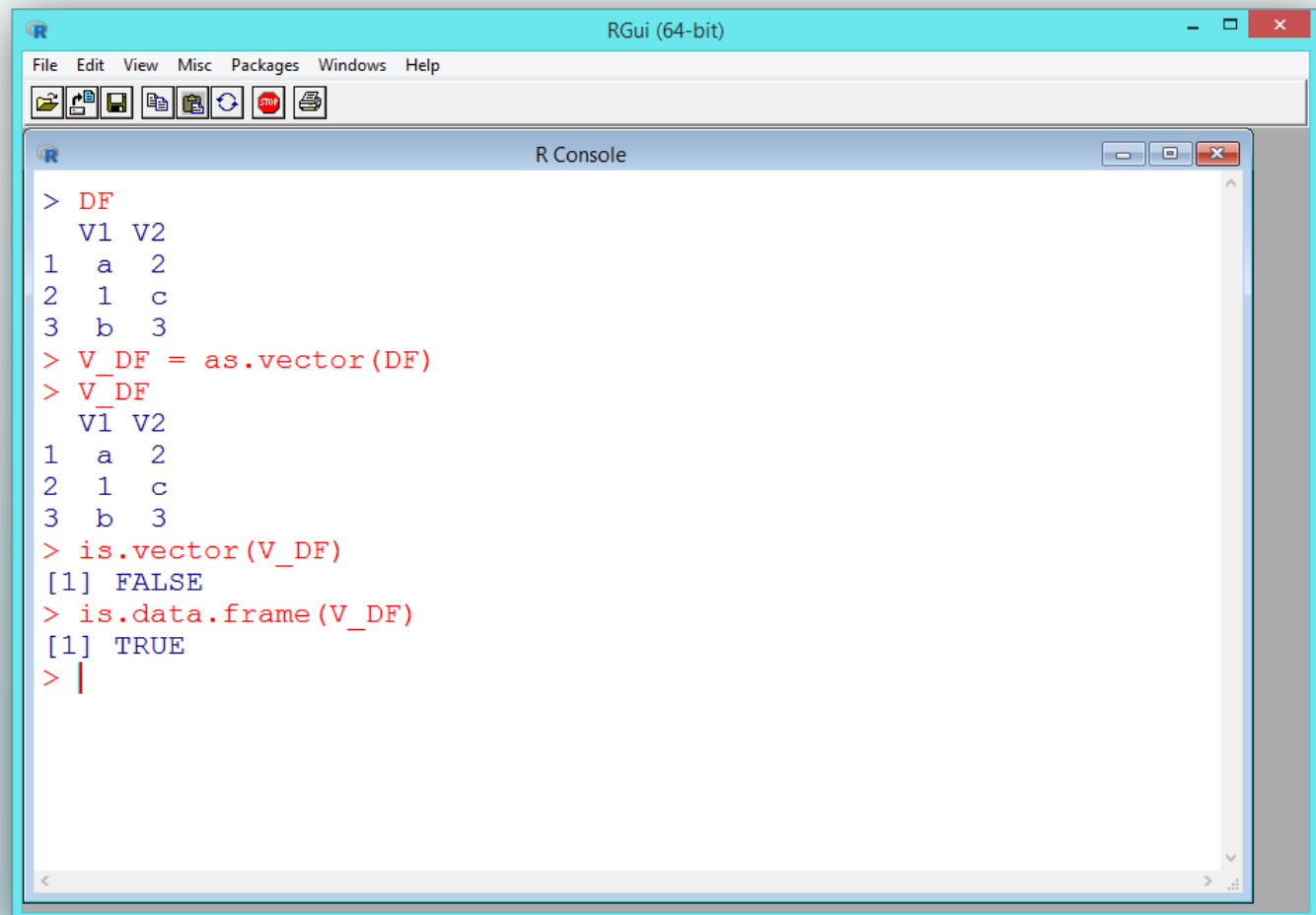
► `as.vector` and `is.vector` (2)



```
> V_1 = 1:12
> is.vector(V_1)
[1] TRUE
>
> M
      [,1] [,2]
[1,] "a"  "2"
[2,] "1"  "c"
[3,] "b"  "3"
> as.vector(M)
[1] "a" "1" "b" "2" "c" "3"
>
> N = as.vector(M)
> N
[1] "a" "1" "b" "2" "c" "3"
> |
```

Identification and Transformation

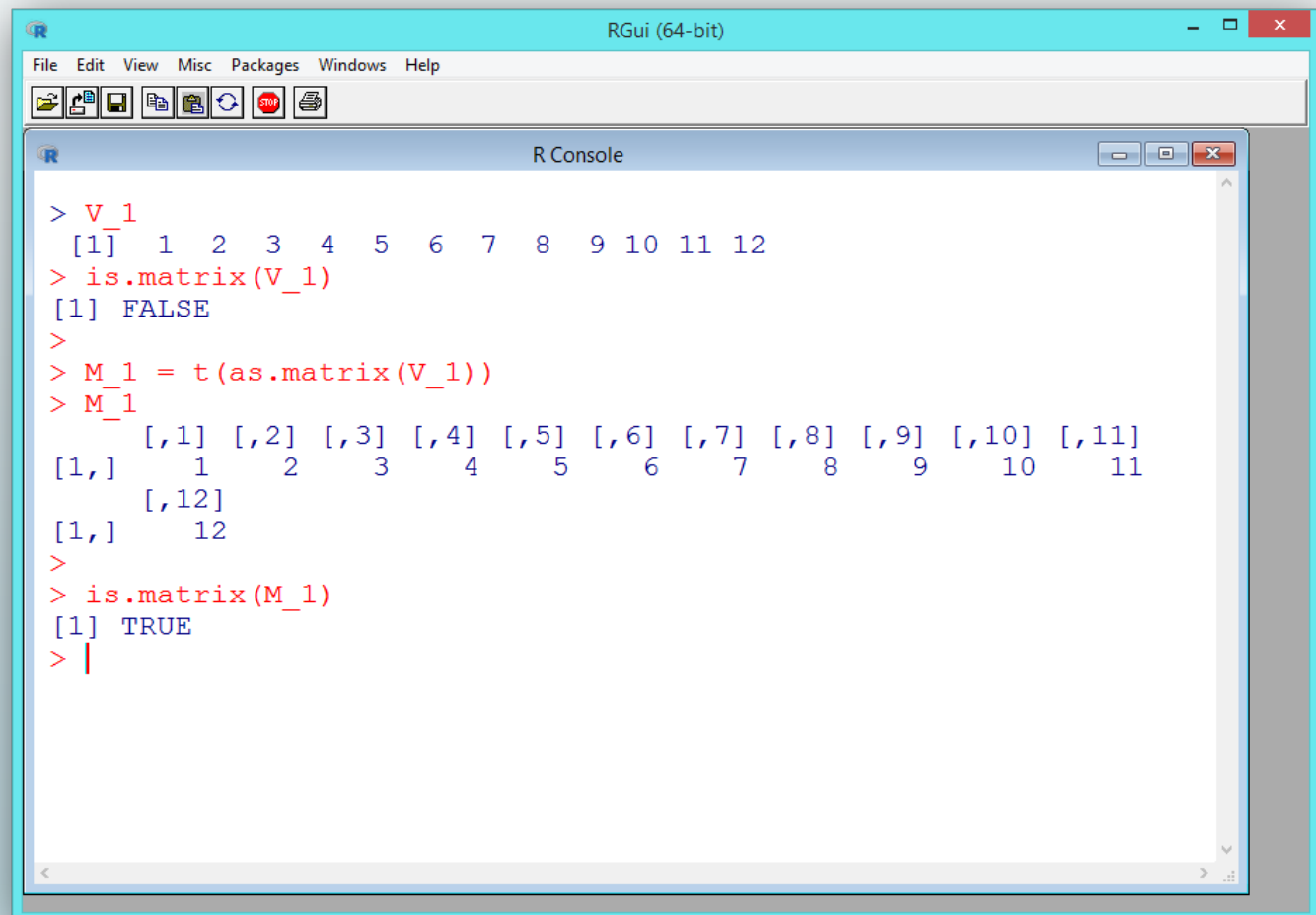
► `as.vector` and `is.vector` (3)



```
> DF
  V1 V2
1  a  2
2  1  c
3  b  3
> V_DF = as.vector(DF)
> V_DF
  V1 V2
1  a  2
2  1  c
3  b  3
> is.vector(V_DF)
[1] FALSE
> is.data.frame(V_DF)
[1] TRUE
> |
```

Identification and Transformation

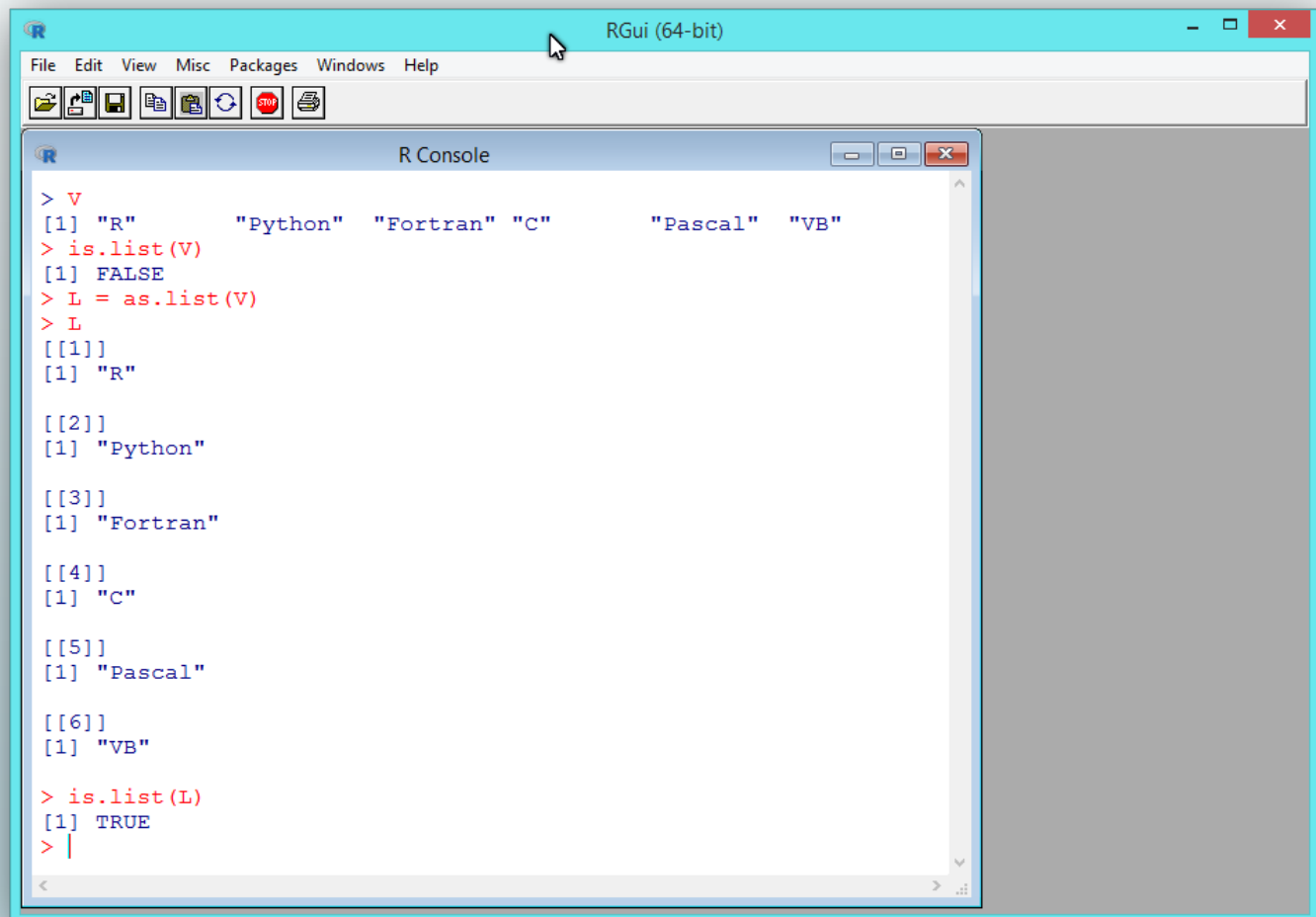
► `as.matrix` and `is.matrix`



```
> V_1
 [1] 1 2 3 4 5 6 7 8 9 10 11 12
> is.matrix(V_1)
 [1] FALSE
>
> M_1 = t(as.matrix(V_1))
> M_1
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
[1,]    1    2    3    4    5    6    7    8    9   10   11
      [,12]
[1,]    12
>
> is.matrix(M_1)
 [1] TRUE
> |
```

Identification and Transformation

► `as.list` and `is.list`



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> V
[1] "R"      "Python" "Fortran" "C"      "Pascal" "VB"
> is.list(V)
[1] FALSE
> L = as.list(V)
> L
[[1]]
[1] "R"

[[2]]
[1] "Python"

[[3]]
[1] "Fortran"

[[4]]
[1] "C"

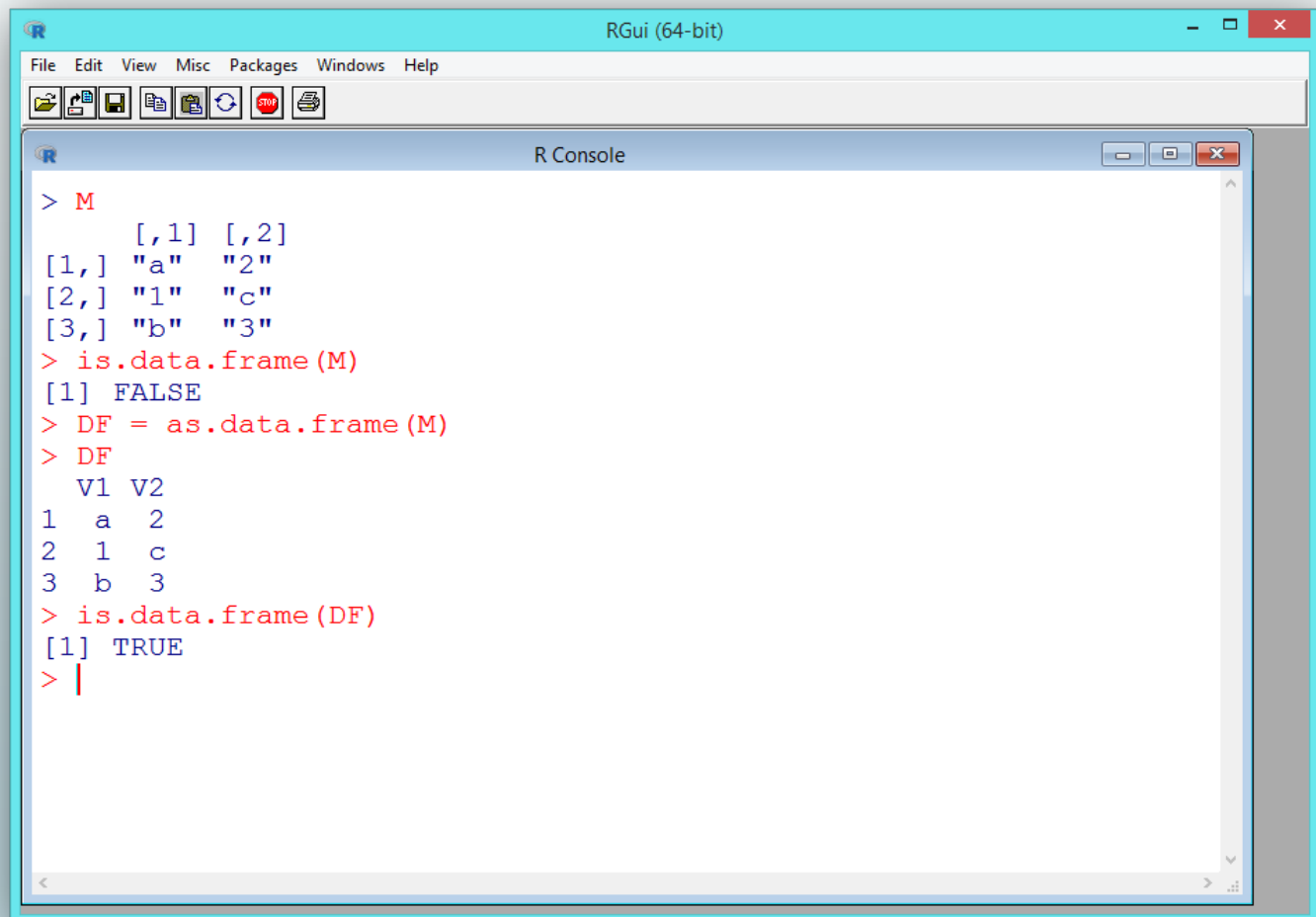
[[5]]
[1] "Pascal"

[[6]]
[1] "VB"

> is.list(L)
[1] TRUE
> |
```

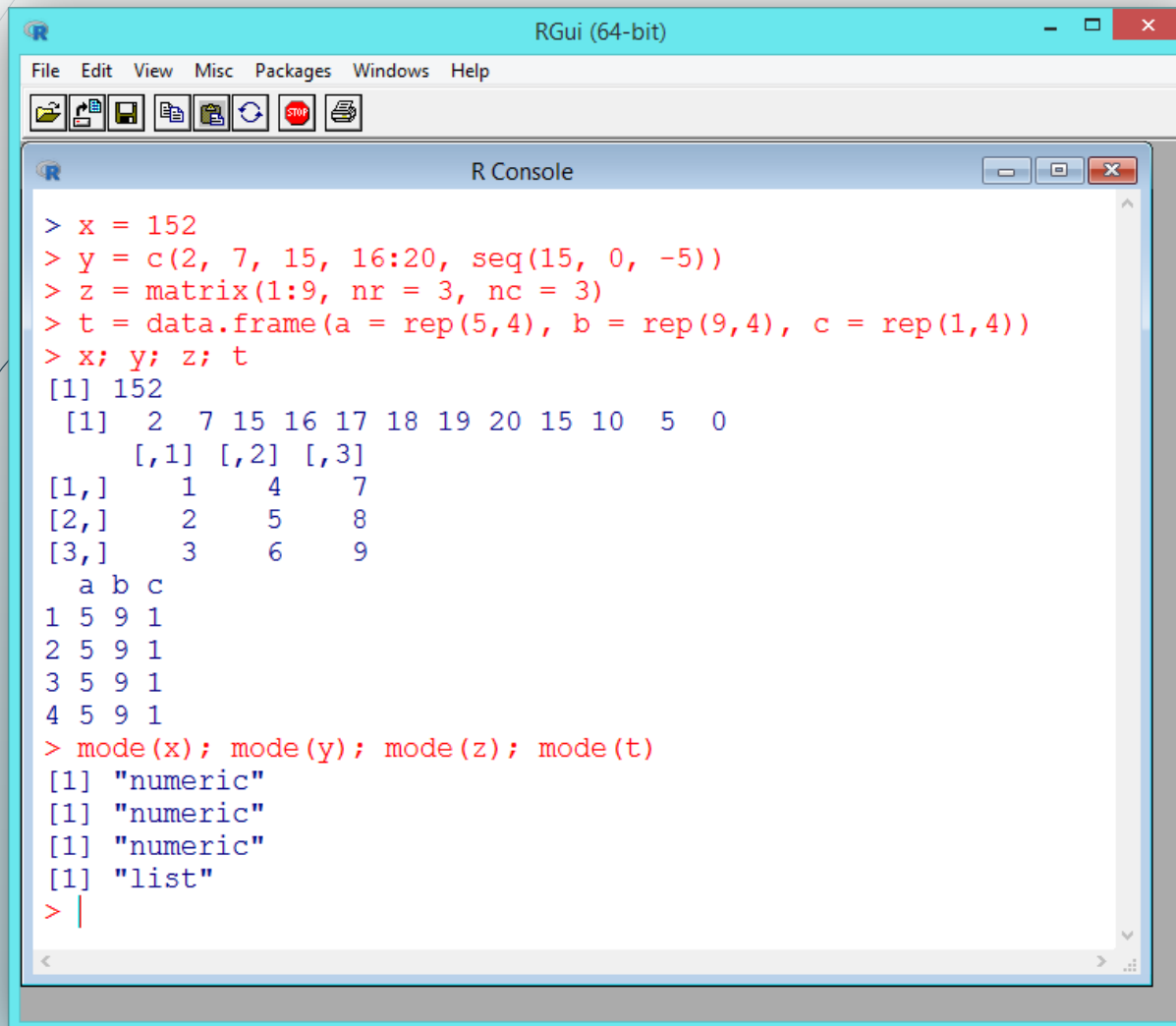
Identification and Transformation

➤ `as.data.frame` and `is.data.frame`



```
> M
      [,1] [,2]
[1,] "a"  "2"
[2,] "1"  "c"
[3,] "b"  "3"
> is.data.frame(M)
[1] FALSE
> DF = as.data.frame(M)
> DF
  V1 V2
1  a  2
2  1  c
3  b  3
> is.data.frame(DF)
[1] TRUE
> |
```

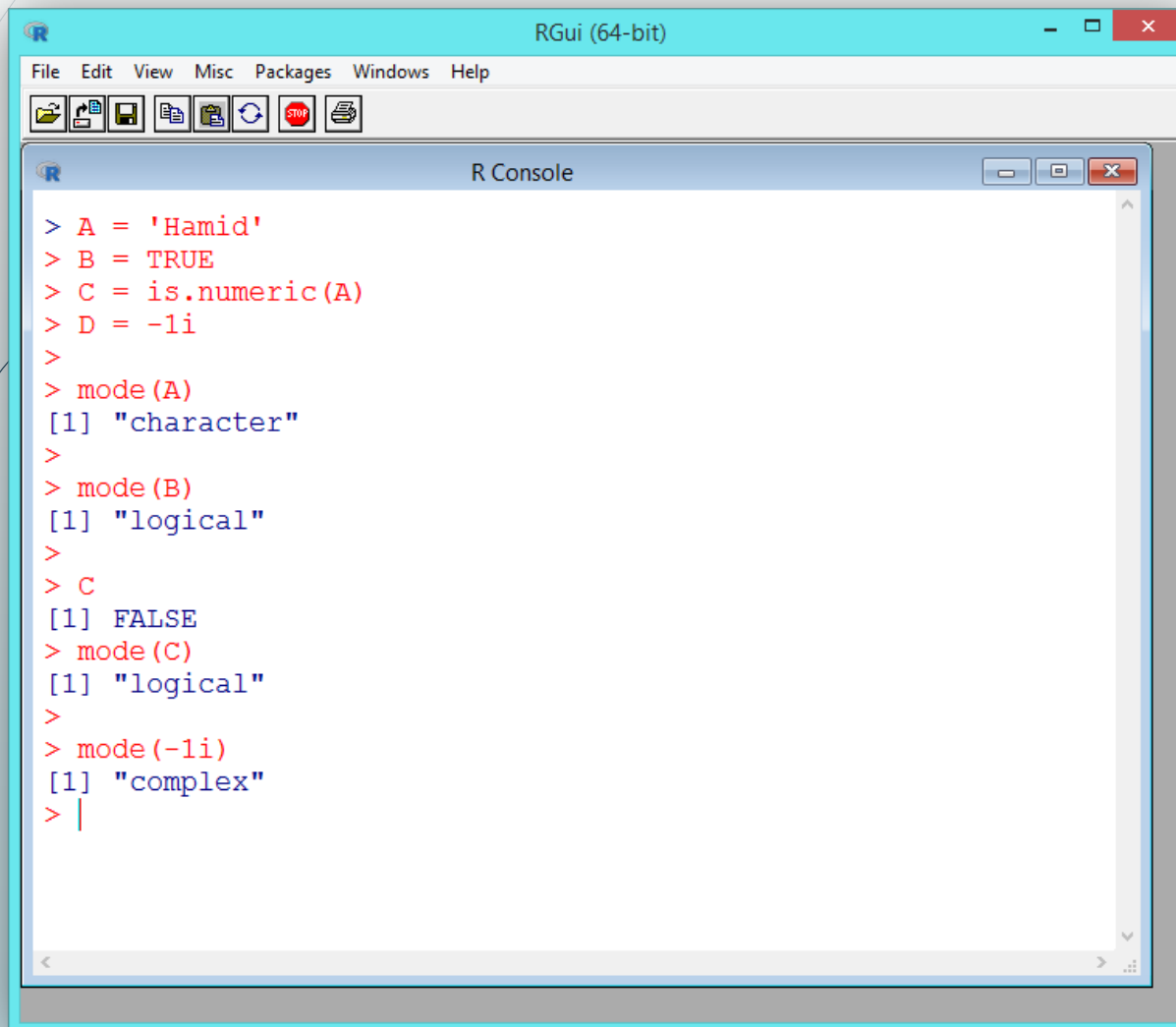

mode



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

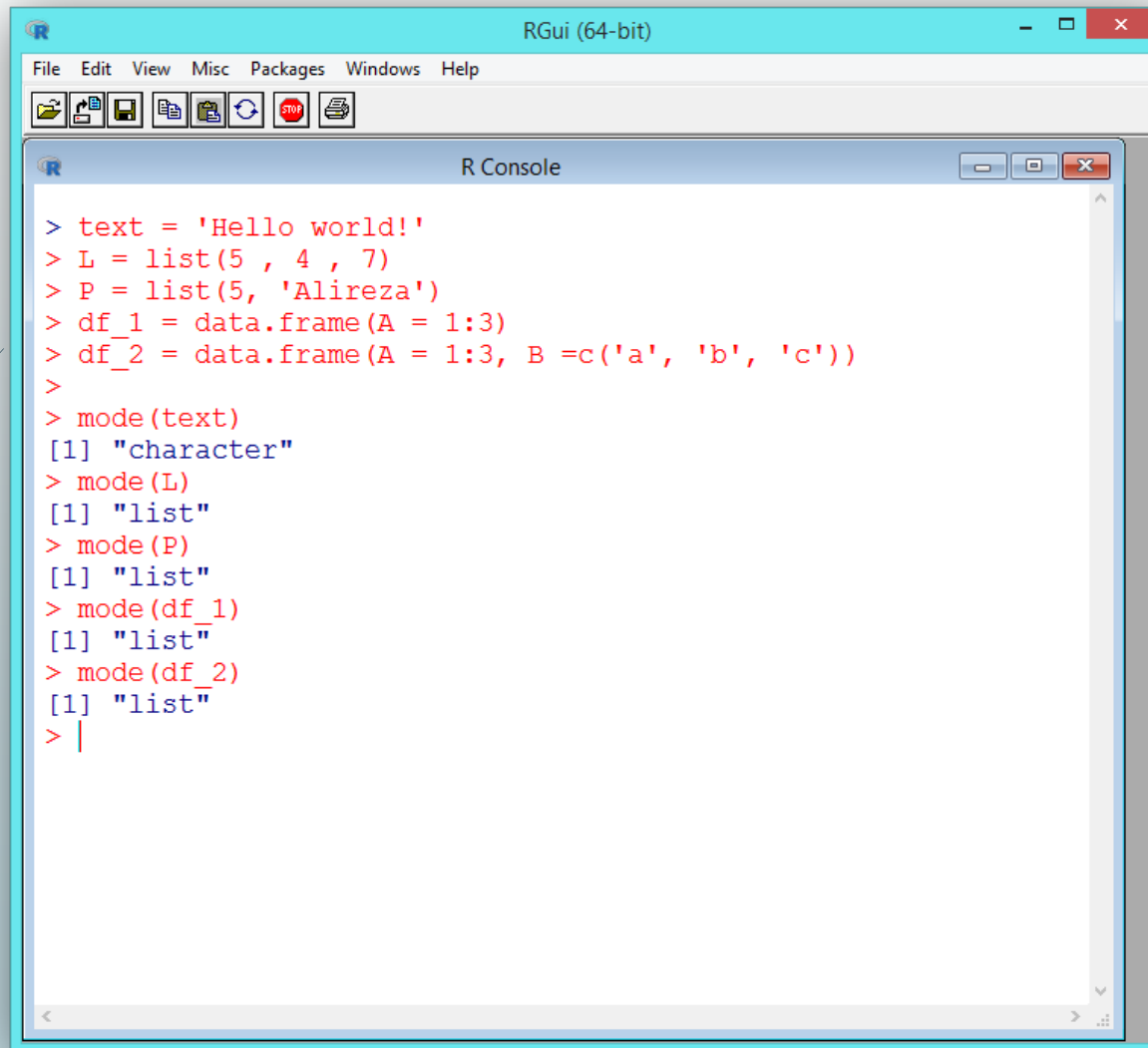
R Console
> x = 152
> y = c(2, 7, 15, 16:20, seq(15, 0, -5))
> z = matrix(1:9, nr = 3, nc = 3)
> t = data.frame(a = rep(5,4), b = rep(9,4), c = rep(1,4))
> x; y; z; t
[1] 152
 [1]  2  7 15 16 17 18 19 20 15 10  5  0
      [,1] [,2] [,3]
 [1,]    1    4    7
 [2,]    2    5    8
 [3,]    3    6    9
   a b c
1 5 9 1
2 5 9 1
3 5 9 1
4 5 9 1
> mode(x); mode(y); mode(z); mode(t)
[1] "numeric"
[1] "numeric"
[1] "numeric"
[1] "list"
> |
```

mode



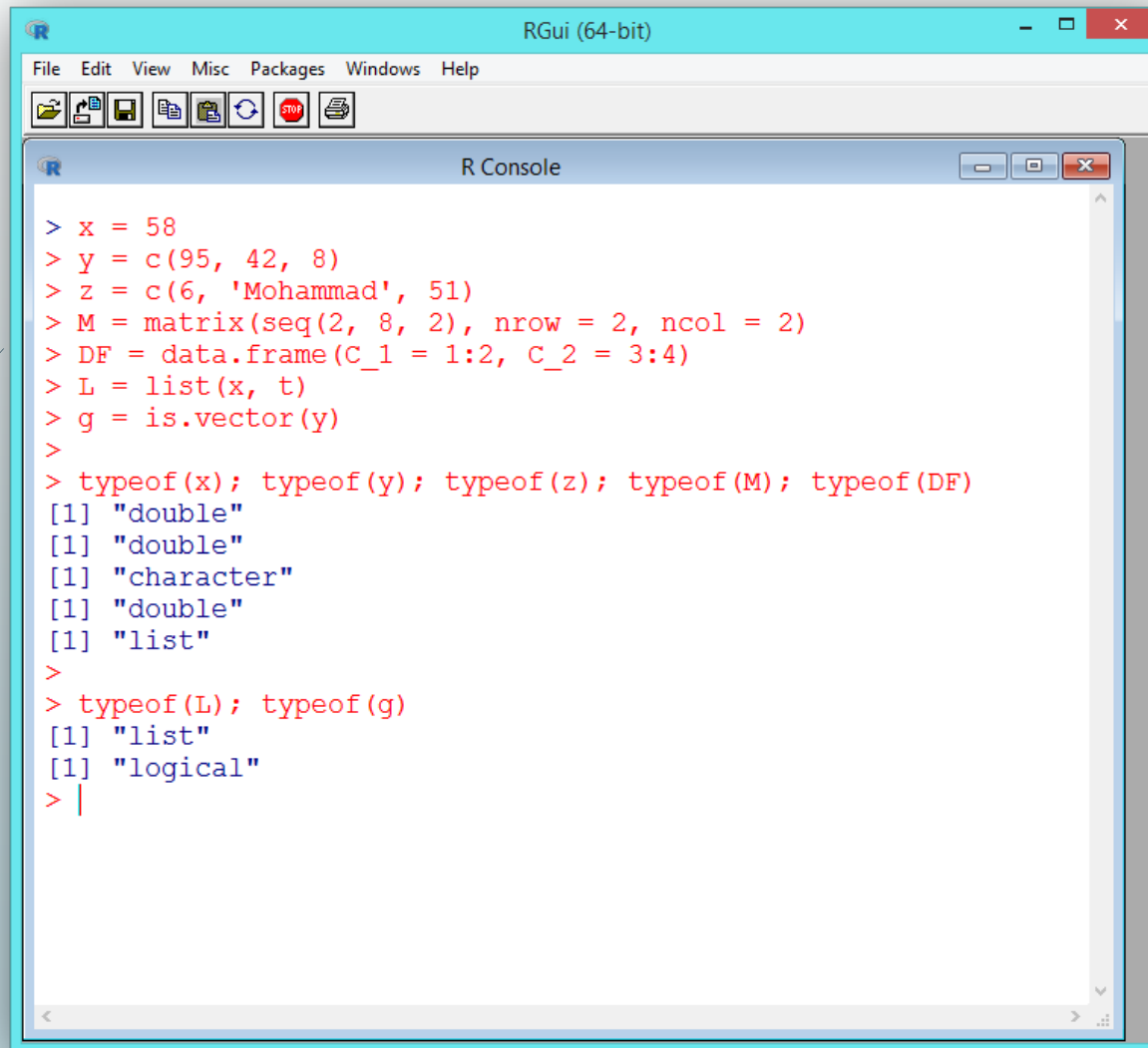
```
> A = 'Hamid'
> B = TRUE
> C = is.numeric(A)
> D = -1i
>
> mode(A)
[1] "character"
>
> mode(B)
[1] "logical"
>
> C
[1] FALSE
> mode(C)
[1] "logical"
>
> mode(-1i)
[1] "complex"
> |
```

mode



```
> text = 'Hello world!'
> L = list(5 , 4 , 7)
> P = list(5, 'Alireza')
> df_1 = data.frame(A = 1:3)
> df_2 = data.frame(A = 1:3, B =c('a', 'b', 'c'))
>
> mode(text)
[1] "character"
> mode(L)
[1] "list"
> mode(P)
[1] "list"
> mode(df_1)
[1] "list"
> mode(df_2)
[1] "list"
> |
```

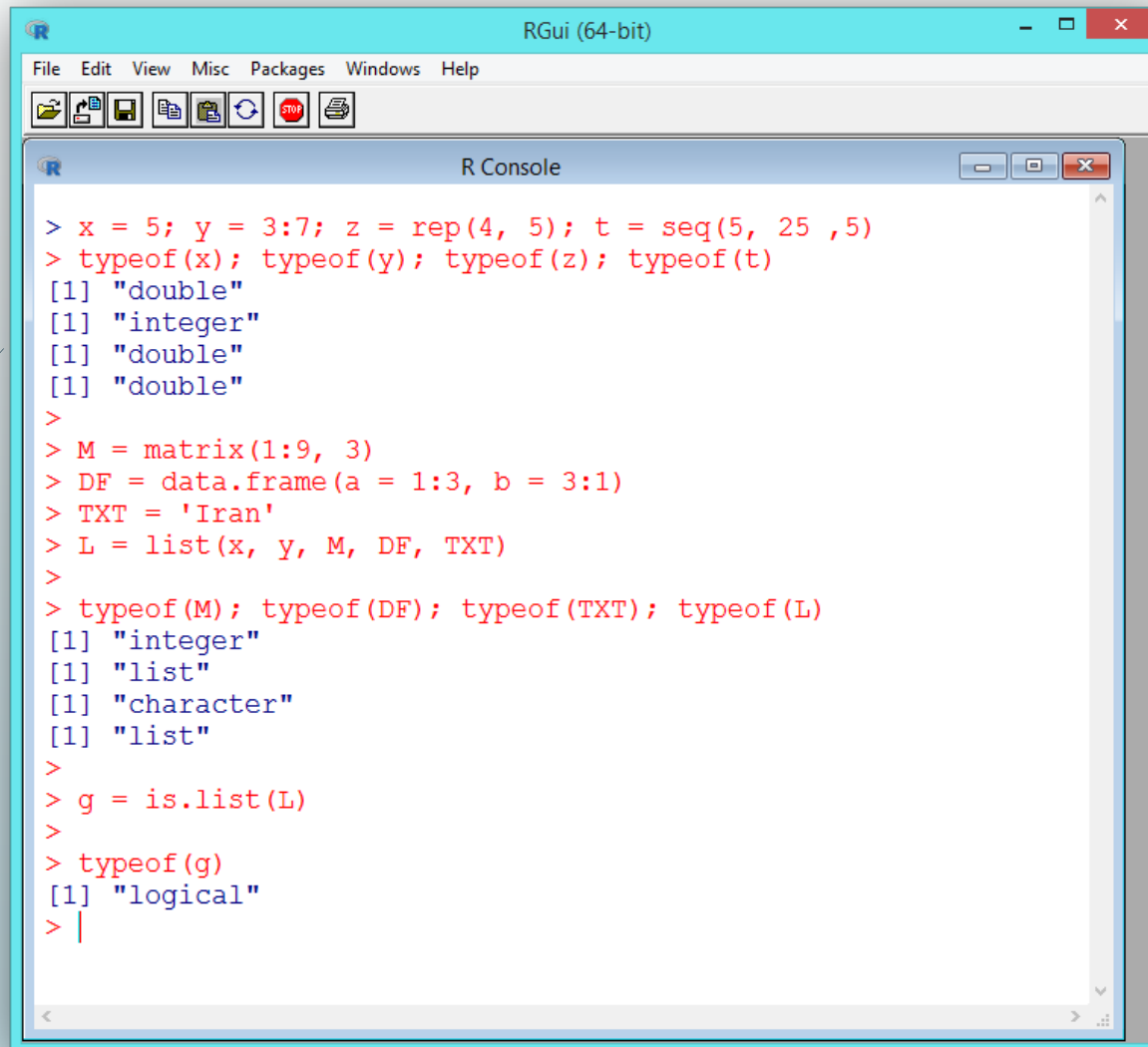
typeof



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

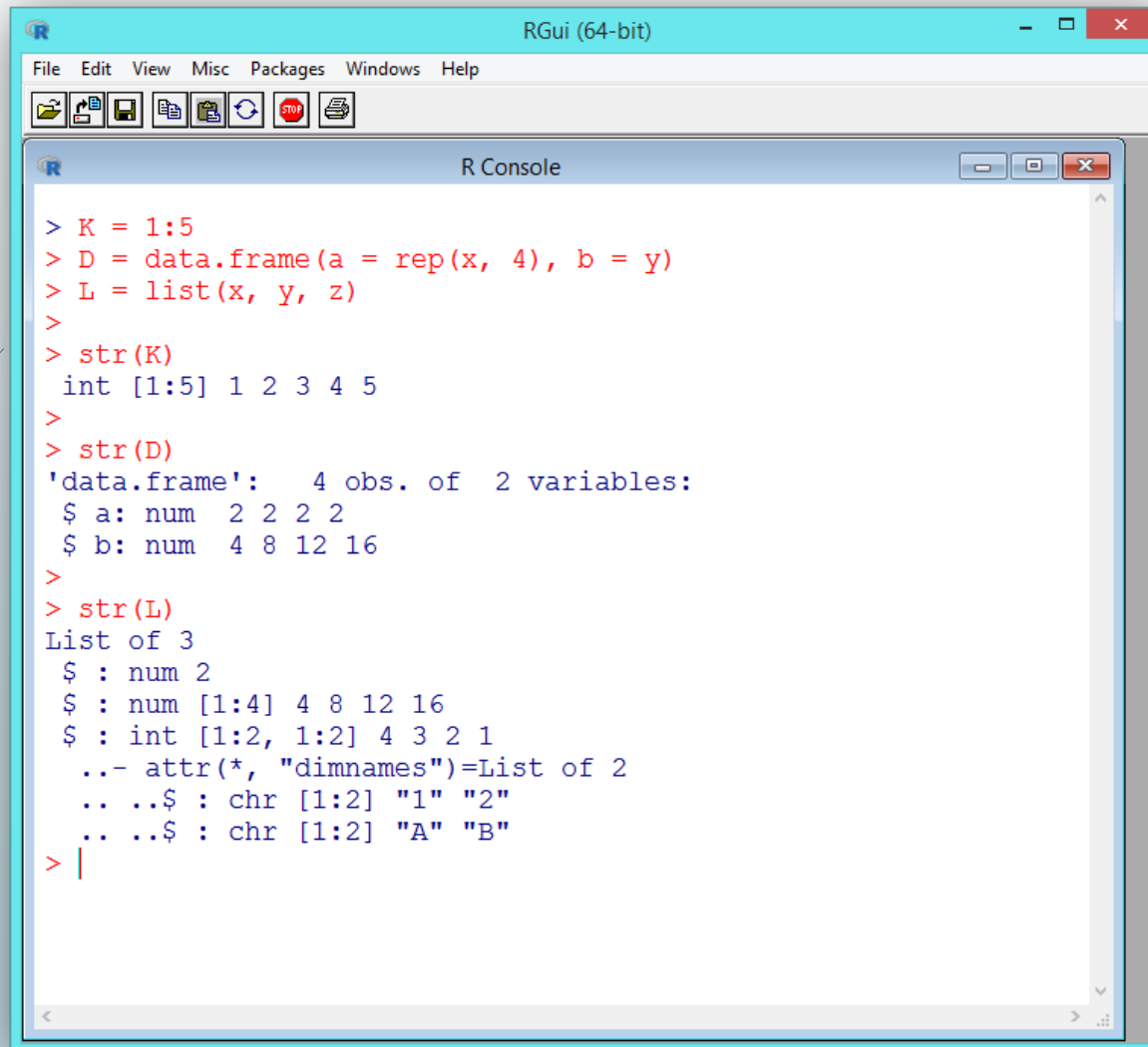
R Console
> x = 58
> y = c(95, 42, 8)
> z = c(6, 'Mohammad', 51)
> M = matrix(seq(2, 8, 2), nrow = 2, ncol = 2)
> DF = data.frame(C_1 = 1:2, C_2 = 3:4)
> L = list(x, t)
> g = is.vector(y)
>
> typeof(x); typeof(y); typeof(z); typeof(M); typeof(DF)
[1] "double"
[1] "double"
[1] "character"
[1] "double"
[1] "list"
>
> typeof(L); typeof(g)
[1] "list"
[1] "logical"
> |
```

typeof



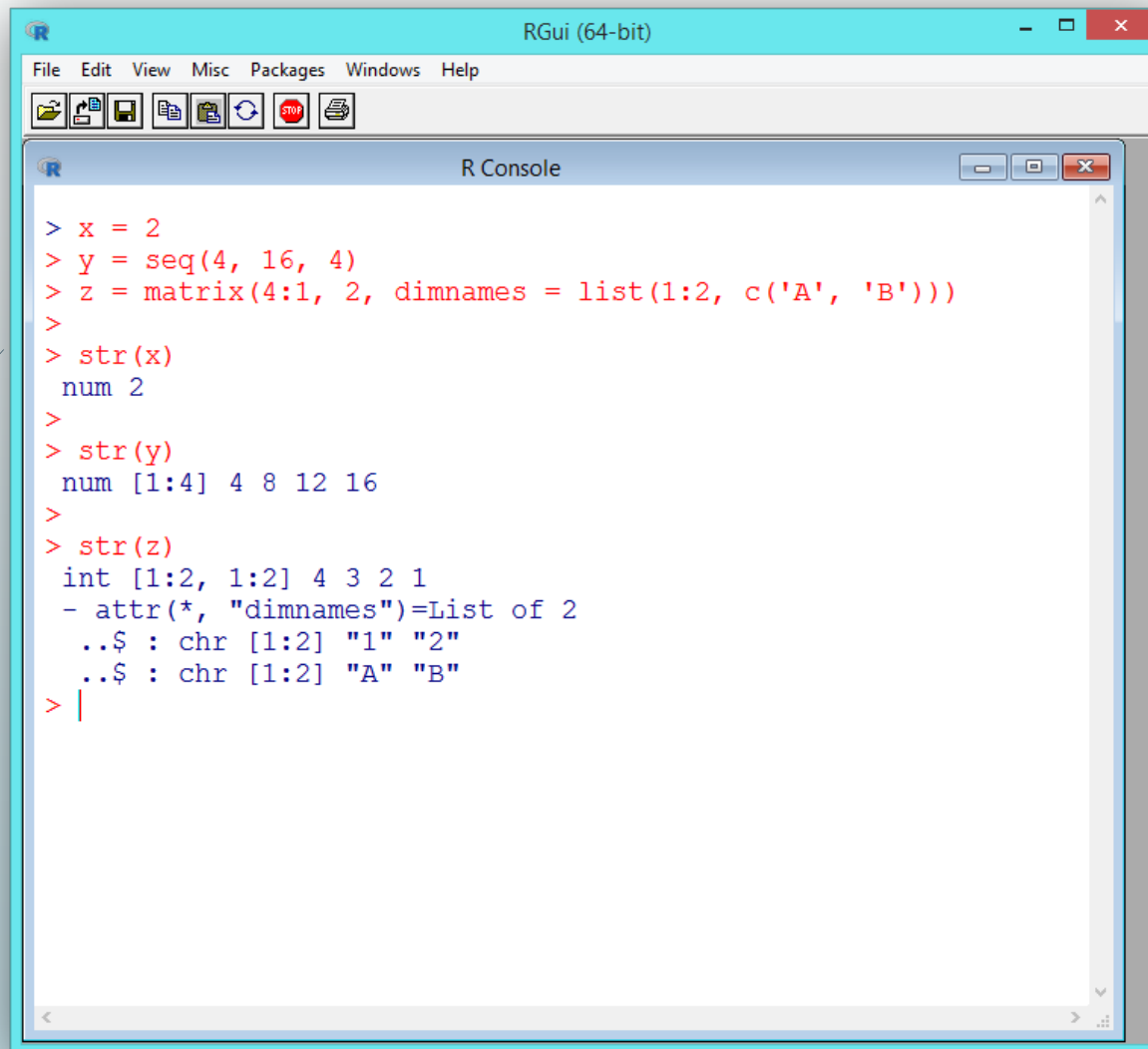
```
> x = 5; y = 3:7; z = rep(4, 5); t = seq(5, 25 ,5)
> typeof(x); typeof(y); typeof(z); typeof(t)
[1] "double"
[1] "integer"
[1] "double"
[1] "double"
>
> M = matrix(1:9, 3)
> DF = data.frame(a = 1:3, b = 3:1)
> TXT = 'Iran'
> L = list(x, y, M, DF, TXT)
>
> typeof(M); typeof(DF); typeof(TXT); typeof(L)
[1] "integer"
[1] "list"
[1] "character"
[1] "list"
>
> g = is.list(L)
>
> typeof(g)
[1] "logical"
> |
```

str



```
> K = 1:5
> D = data.frame(a = rep(x, 4), b = y)
> L = list(x, y, z)
>
> str(K)
int [1:5] 1 2 3 4 5
>
> str(D)
'data.frame':  4 obs. of  2 variables:
 $ a: num  2 2 2 2
 $ b: num  4 8 12 16
>
> str(L)
List of 3
 $ : num 2
 $ : num [1:4] 4 8 12 16
 $ : int [1:2, 1:2] 4 3 2 1
 ..- attr(*, "dimnames")=List of 2
 .. ..$ : chr [1:2] "1" "2"
 .. ..$ : chr [1:2] "A" "B"
> |
```

str



```
> x = 2
> y = seq(4, 16, 4)
> z = matrix(4:1, 2, dimnames = list(1:2, c('A', 'B')))
>
> str(x)
num 2
>
> str(y)
num [1:4] 4 8 12 16
>
> str(z)
int [1:2, 1:2] 4 3 2 1
- attr(*, "dimnames")=List of 2
..$ : chr [1:2] "1" "2"
..$ : chr [1:2] "A" "B"
> |
```

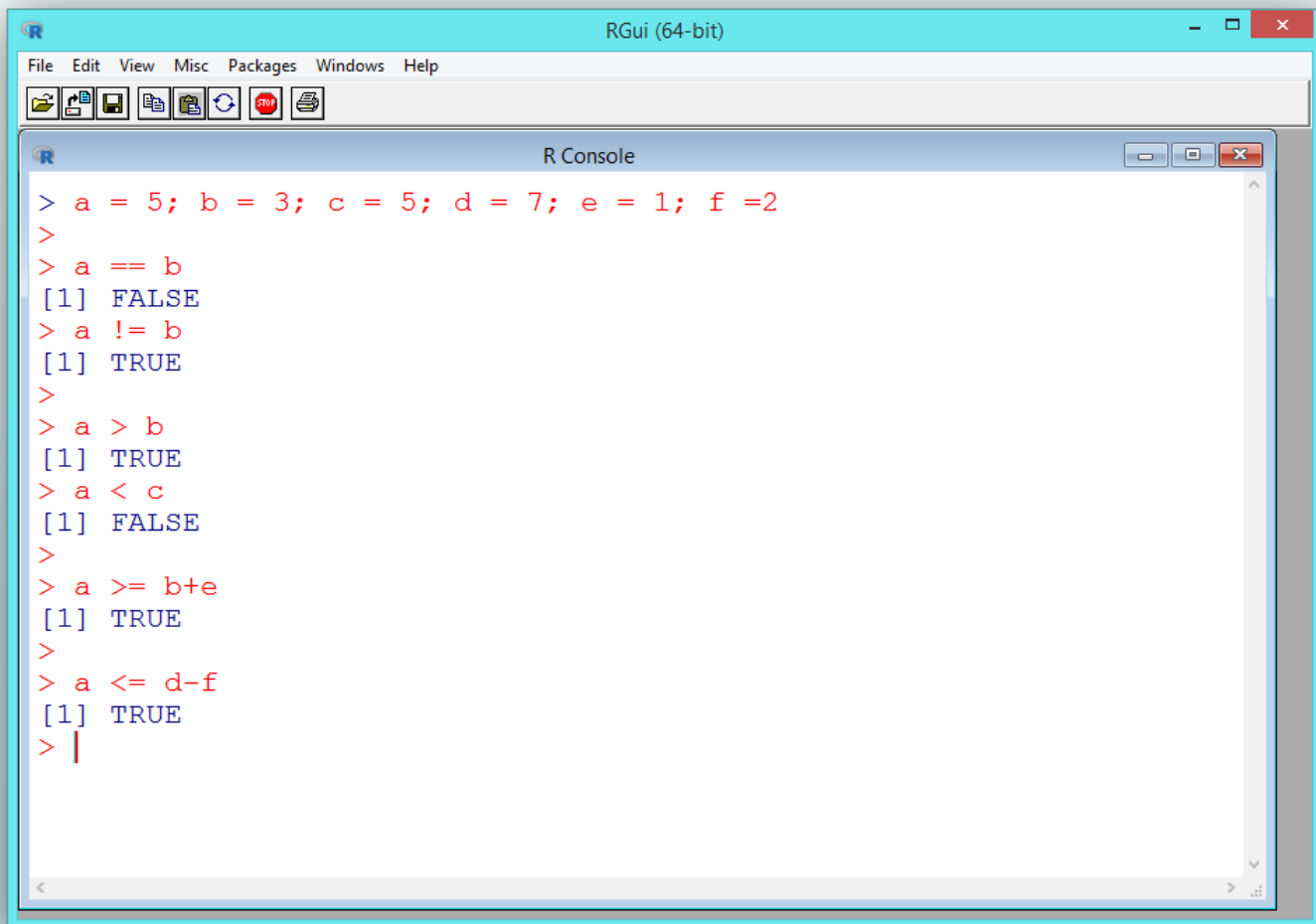


Control Structures

Conditional Statements and Loops

Relational Operators

► Relational Operators

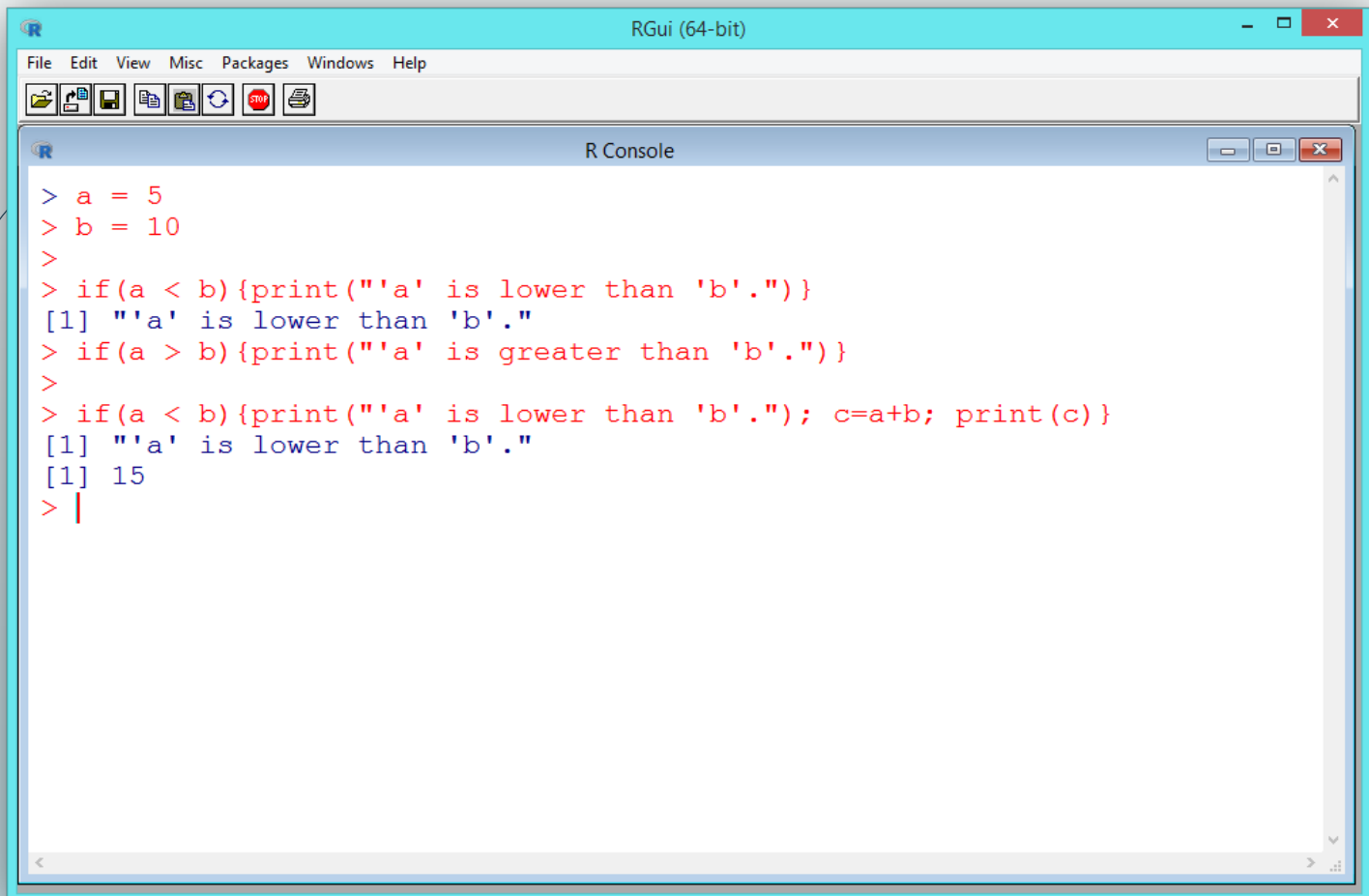


```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> a = 5; b = 3; c = 5; d = 7; e = 1; f = 2
>
> a == b
[1] FALSE
> a != b
[1] TRUE
>
> a > b
[1] TRUE
> a < c
[1] FALSE
>
> a >= b+e
[1] TRUE
>
> a <= d-f
[1] TRUE
> |
```

Conditional Statements

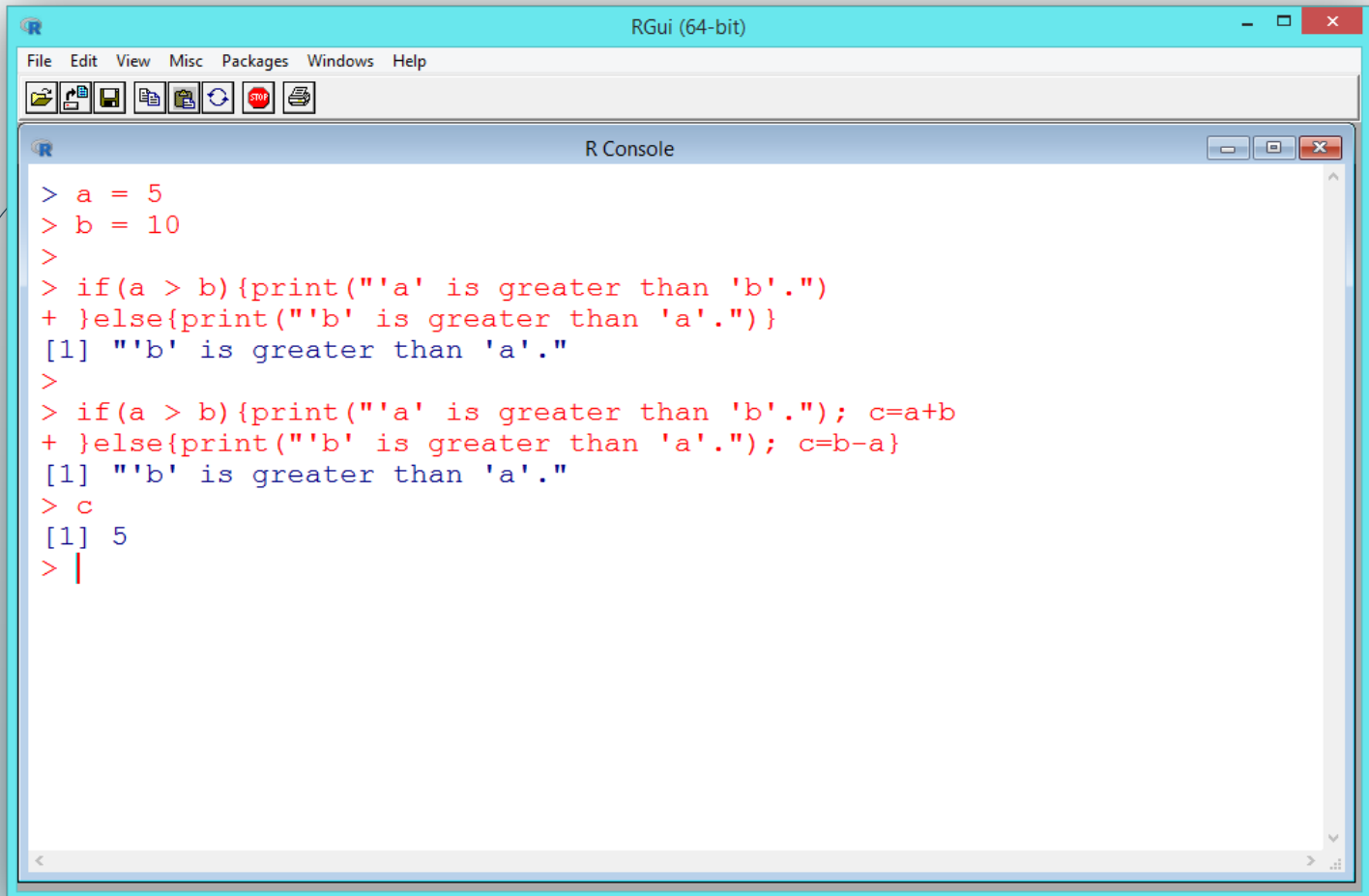
► `if ...` Statements



```
> a = 5
> b = 10
>
> if(a < b){print("'a' is lower than 'b'.")}
[1] "'a' is lower than 'b'."
> if(a > b){print("'a' is greater than 'b'.")}
>
> if(a < b){print("'a' is lower than 'b'."); c=a+b; print(c)}
[1] "'a' is lower than 'b'."
[1] 15
> |
```

Conditional Statements

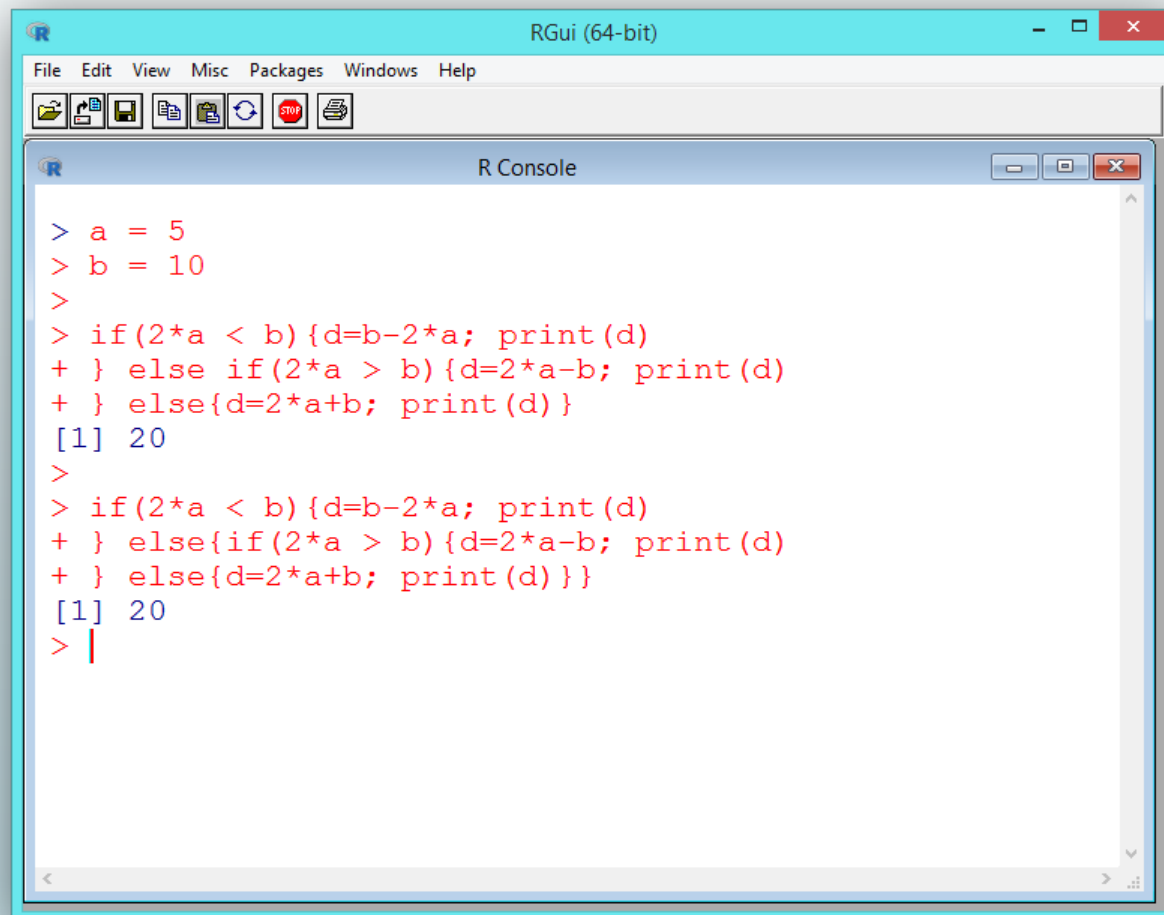
➤ `if ... else` Statements (1)



```
> a = 5
> b = 10
>
> if(a > b){print("'a' is greater than 'b'.")}
+ }else{print("'b' is greater than 'a'.")}
[1] "'b' is greater than 'a'."
>
> if(a > b){print("'a' is greater than 'b'."); c=a+b
+ }else{print("'b' is greater than 'a'."); c=b-a}
[1] "'b' is greater than 'a'."
> c
[1] 5
> |
```

Conditional Statements

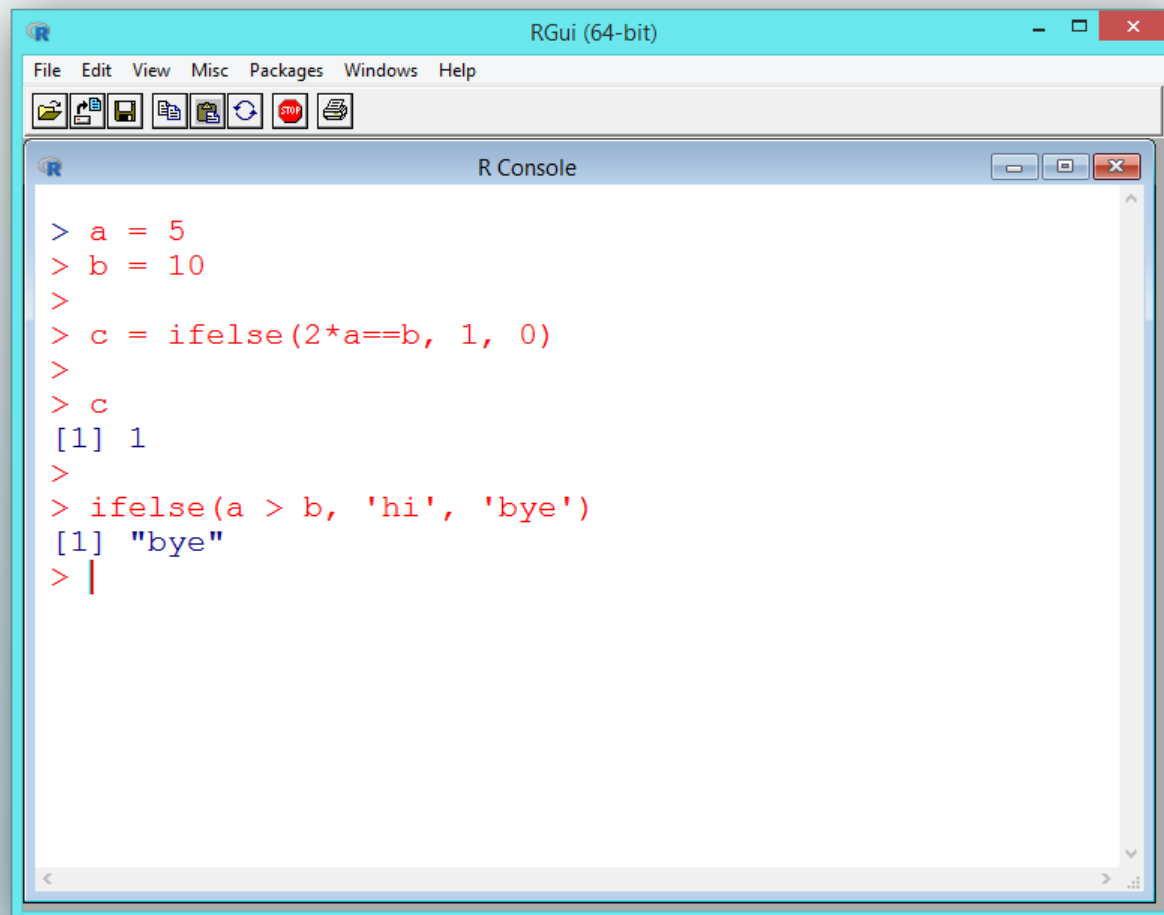
► if ... else Statements (2)



```
> a = 5
> b = 10
>
> if(2*a < b){d=b-2*a; print(d)
+ } else if(2*a > b){d=2*a-b; print(d)
+ } else{d=2*a+b; print(d)}
[1] 20
>
> if(2*a < b){d=b-2*a; print(d)
+ } else{if(2*a > b){d=2*a-b; print(d)
+ } else{d=2*a+b; print(d)}}
[1] 20
> |
```

Conditional Statements

► ifelse Statements

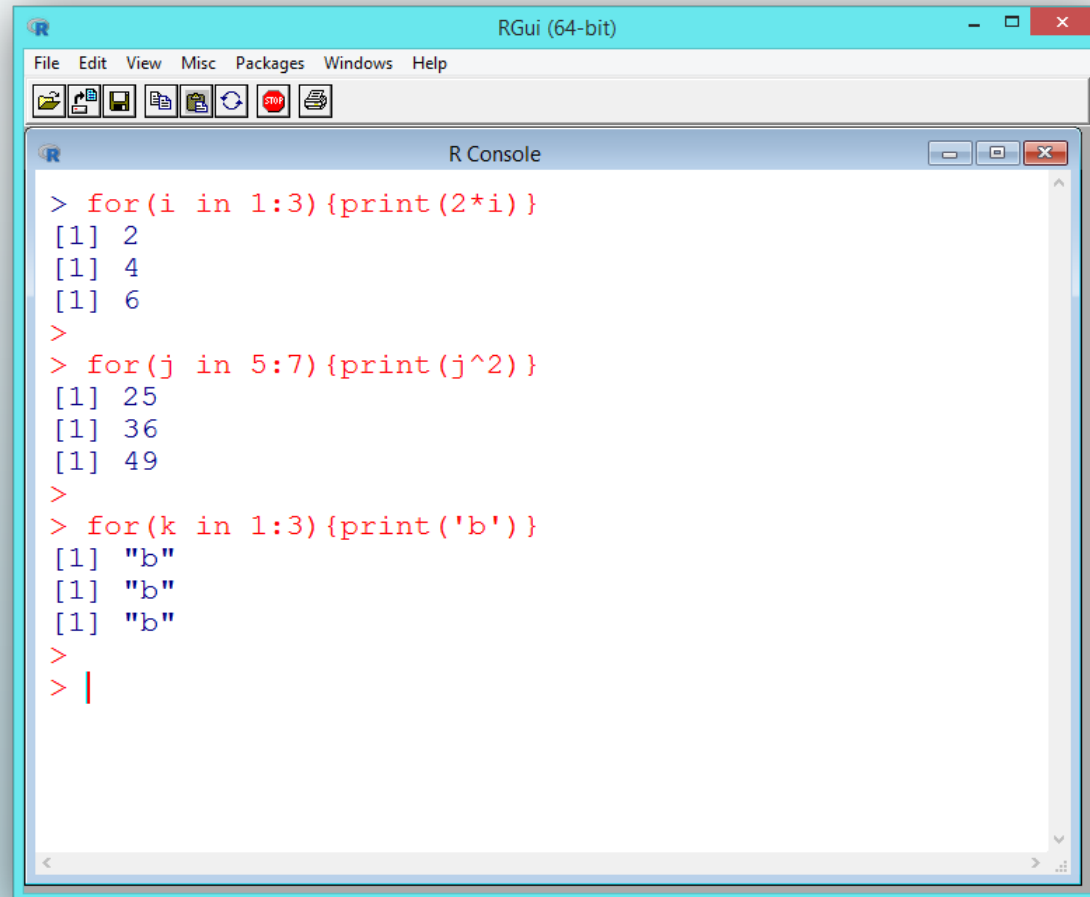


```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> a = 5
> b = 10
>
> c = ifelse(2*a==b, 1, 0)
>
> c
[1] 1
>
> ifelse(a > b, 'hi', 'bye')
[1] "bye"
> |
```

Loop

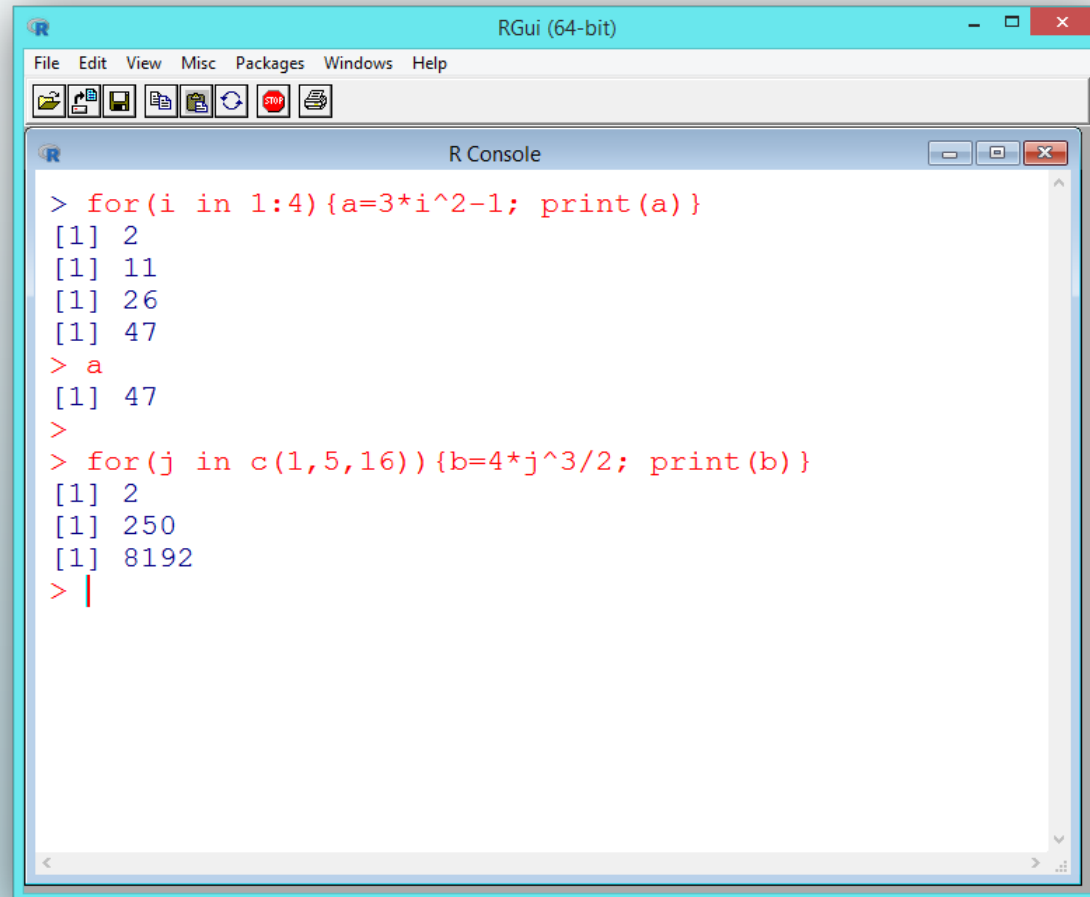
► for Loops (1)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]
R Console
> for(i in 1:3){print(2*i)}
[1] 2
[1] 4
[1] 6
>
> for(j in 5:7){print(j^2)}
[1] 25
[1] 36
[1] 49
>
> for(k in 1:3){print('b')}
[1] "b"
[1] "b"
[1] "b"
>
> |
```

Loop

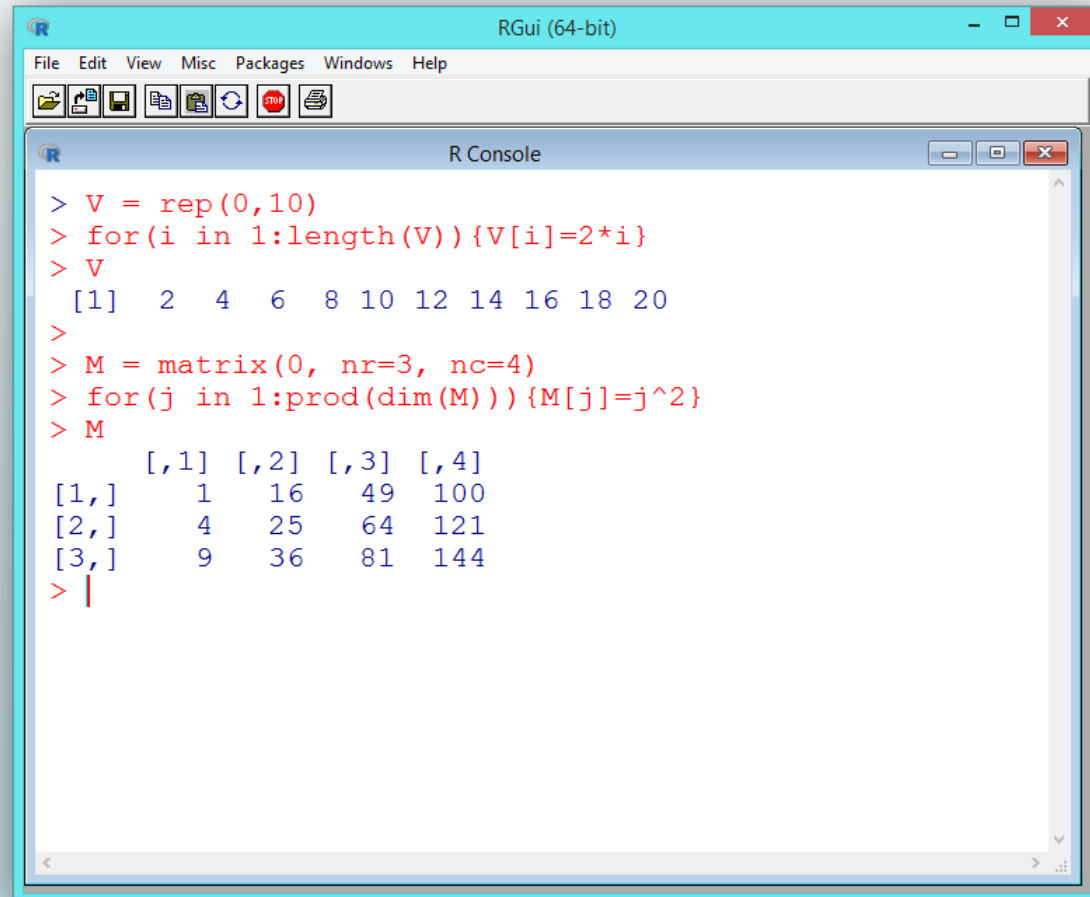
► for Loops (2)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]
R Console
> for(i in 1:4){a=3*i^2-1; print(a)}
[1] 2
[1] 11
[1] 26
[1] 47
> a
[1] 47
>
> for(j in c(1,5,16)){b=4*j^3/2; print(b)}
[1] 2
[1] 250
[1] 8192
> |
```

Loop

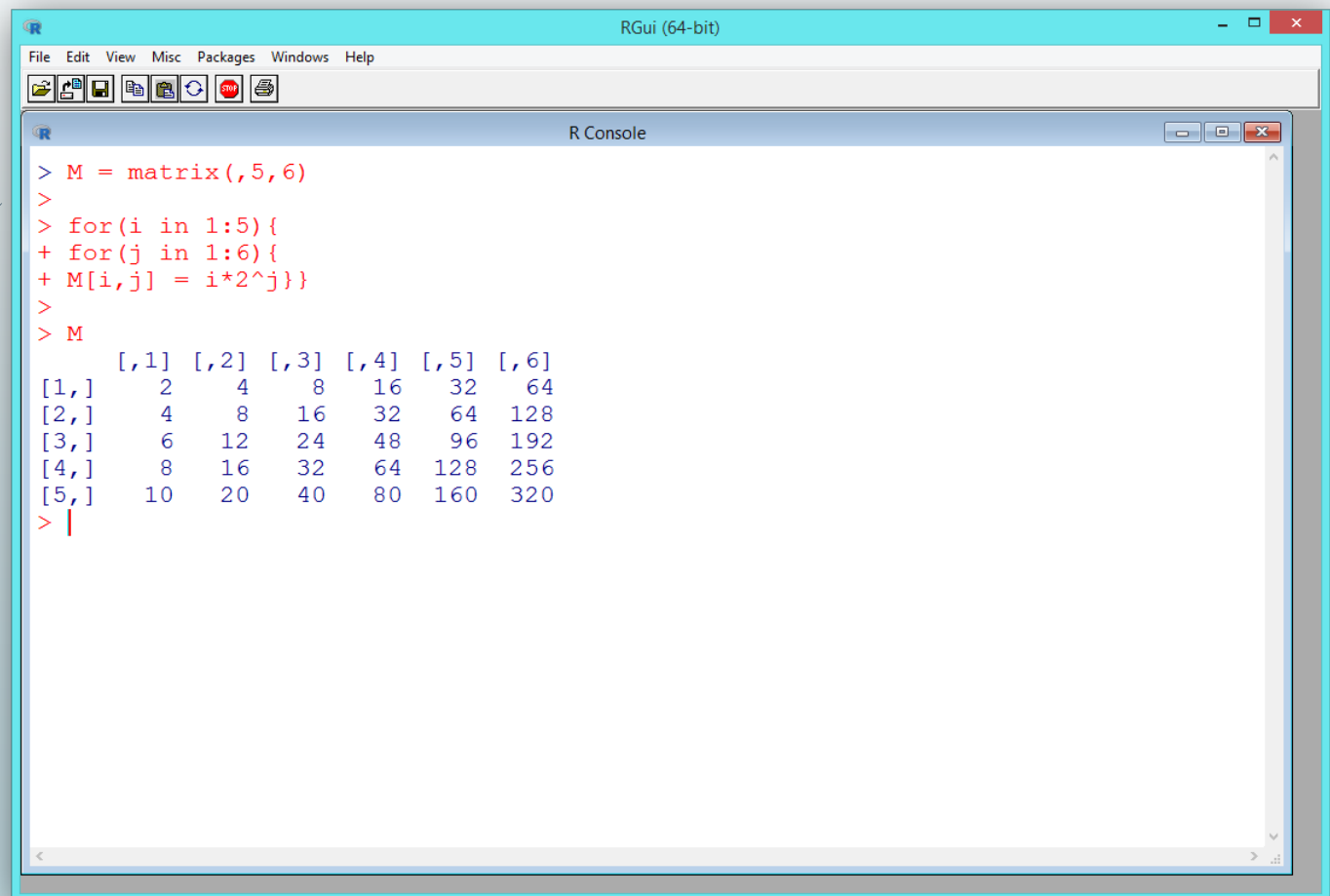
► for Loops (3)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> V = rep(0,10)
> for(i in 1:length(V)){V[i]=2*i}
> V
[1] 2 4 6 8 10 12 14 16 18 20
>
> M = matrix(0, nr=3, nc=4)
> for(j in 1:prod(dim(M))){M[j]=j^2}
> M
      [,1] [,2] [,3] [,4]
[1,]    1   16   49  100
[2,]    4   25   64  121
[3,]    9   36   81  144
> |
```


Loop

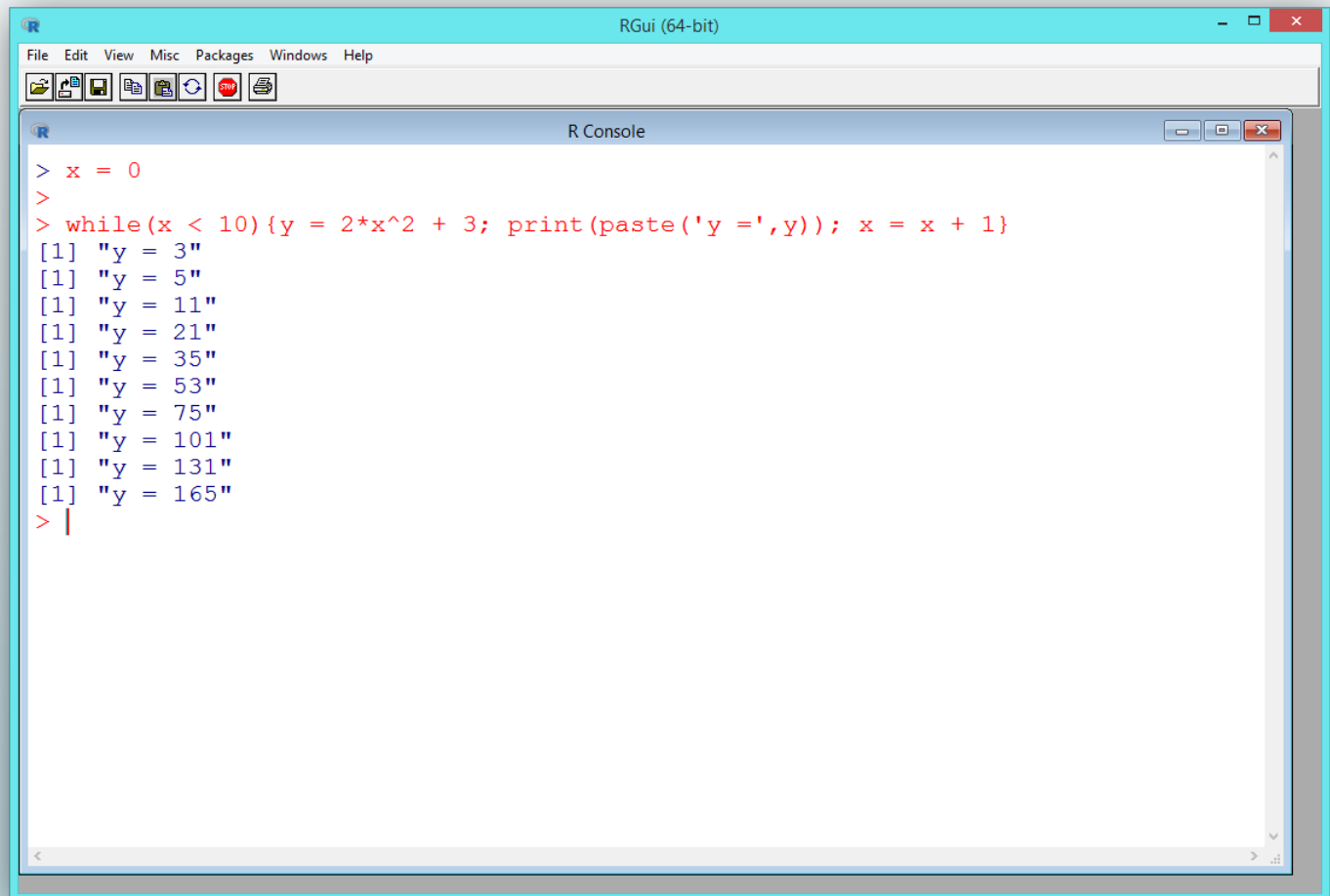
► Nested for Loops



```
> M = matrix(,5,6)
>
> for(i in 1:5){
+ for(j in 1:6){
+ M[i,j] = i*2^j}}
>
> M
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]     2     4     8    16    32    64
[2,]     4     8    16    32    64   128
[3,]     6    12    24    48    96   192
[4,]     8    16    32    64   128   256
[5,]    10    20    40    80   160   320
> |
```

Loop

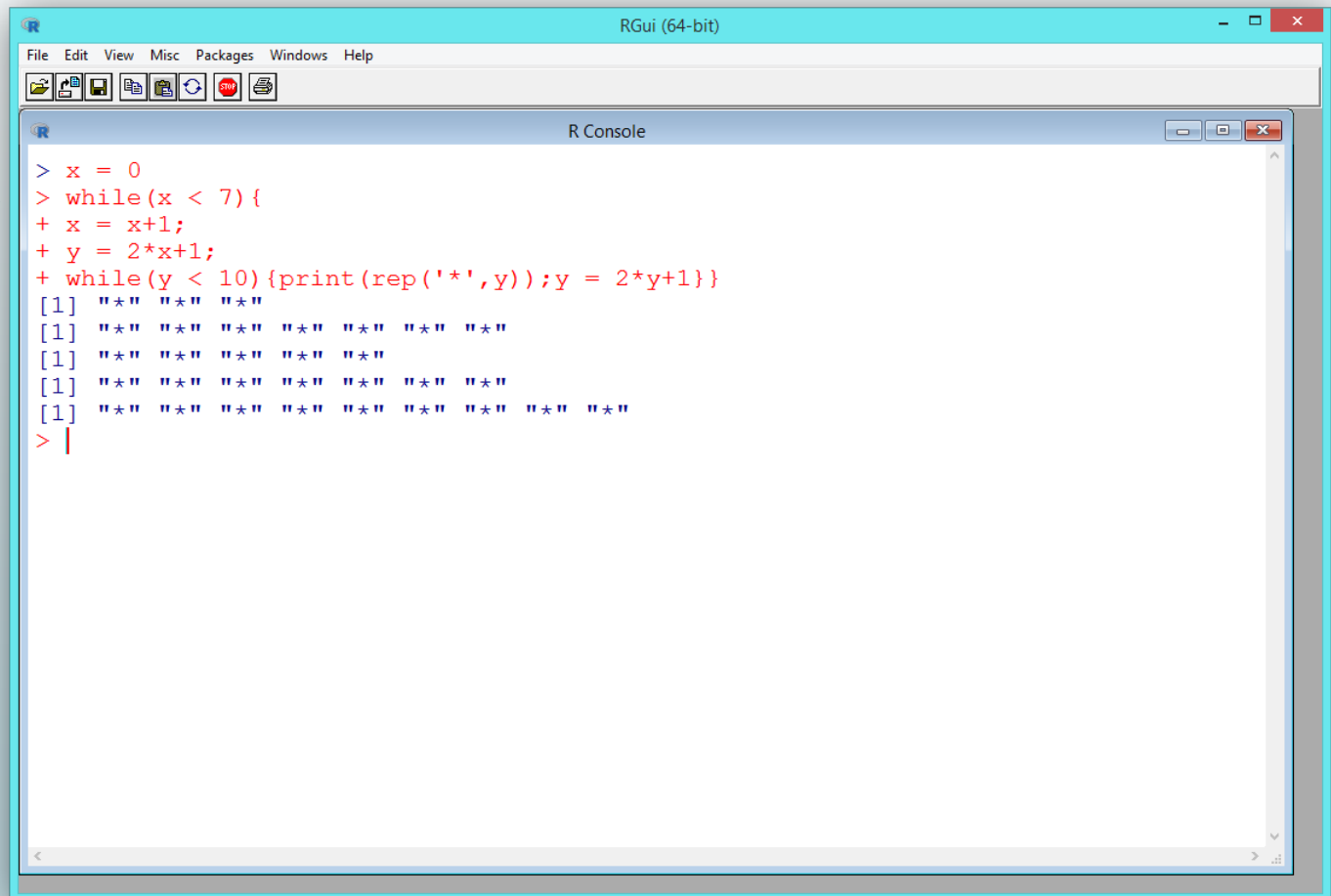
► while Loops



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> x = 0
>
> while(x < 10){y = 2*x^2 + 3; print(paste('y =',y)); x = x + 1}
[1] "y = 3"
[1] "y = 5"
[1] "y = 11"
[1] "y = 21"
[1] "y = 35"
[1] "y = 53"
[1] "y = 75"
[1] "y = 101"
[1] "y = 131"
[1] "y = 165"
> |
```

Loop

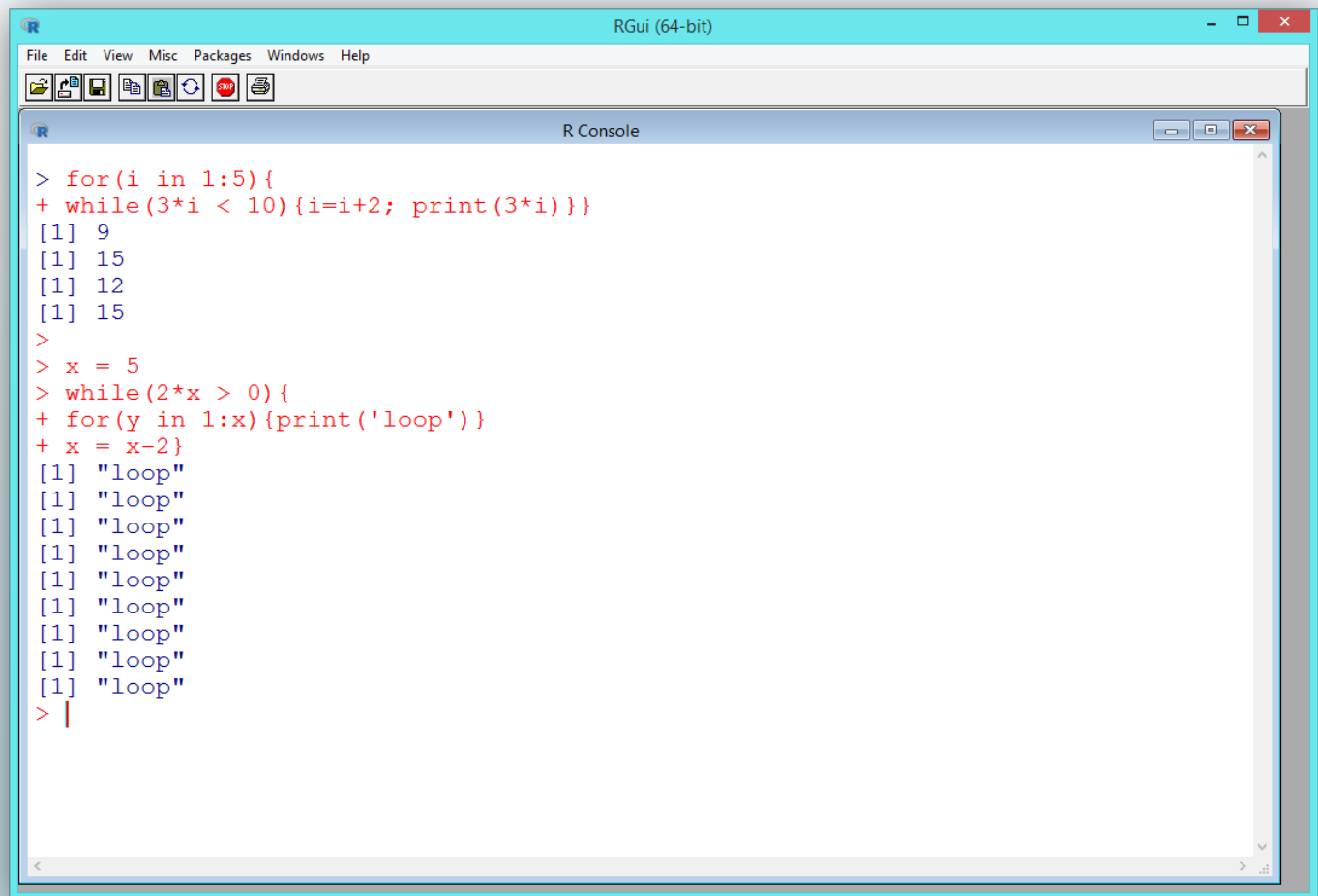
➡ Nested while Loops



```
> x = 0
> while(x < 7){
+ x = x+1;
+ y = 2*x+1;
+ while(y < 10){print(rep('*',y));y = 2*y+1}}
[1] "*" "*" "*"
[1] "*" "*" "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*" "*" "*" "*" "*" "*"
> |
```

Loop

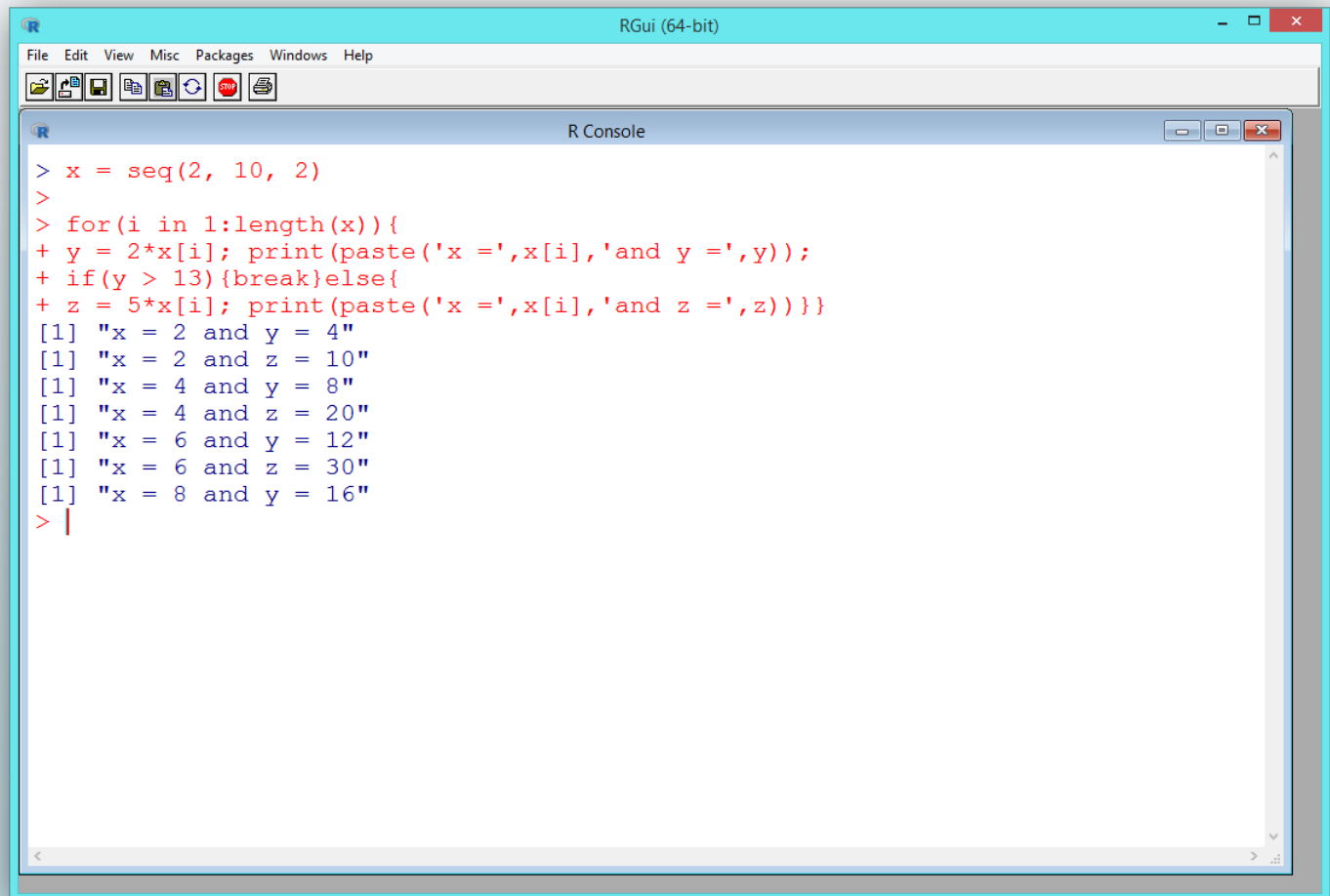
► Nested for and while Loops



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> for(i in 1:5){
+ while(3*i < 10){i=i+2; print(3*i)}}
[1] 9
[1] 15
[1] 12
[1] 15
>
> x = 5
> while(2*x > 0){
+ for(y in 1:x){print('loop')}
+ x = x-2}
[1] "loop"
[1] "loop"
[1] "loop"
[1] "loop"
[1] "loop"
[1] "loop"
[1] "loop"
[1] "loop"
[1] "loop"
[1] "loop"
> |
```

Loop

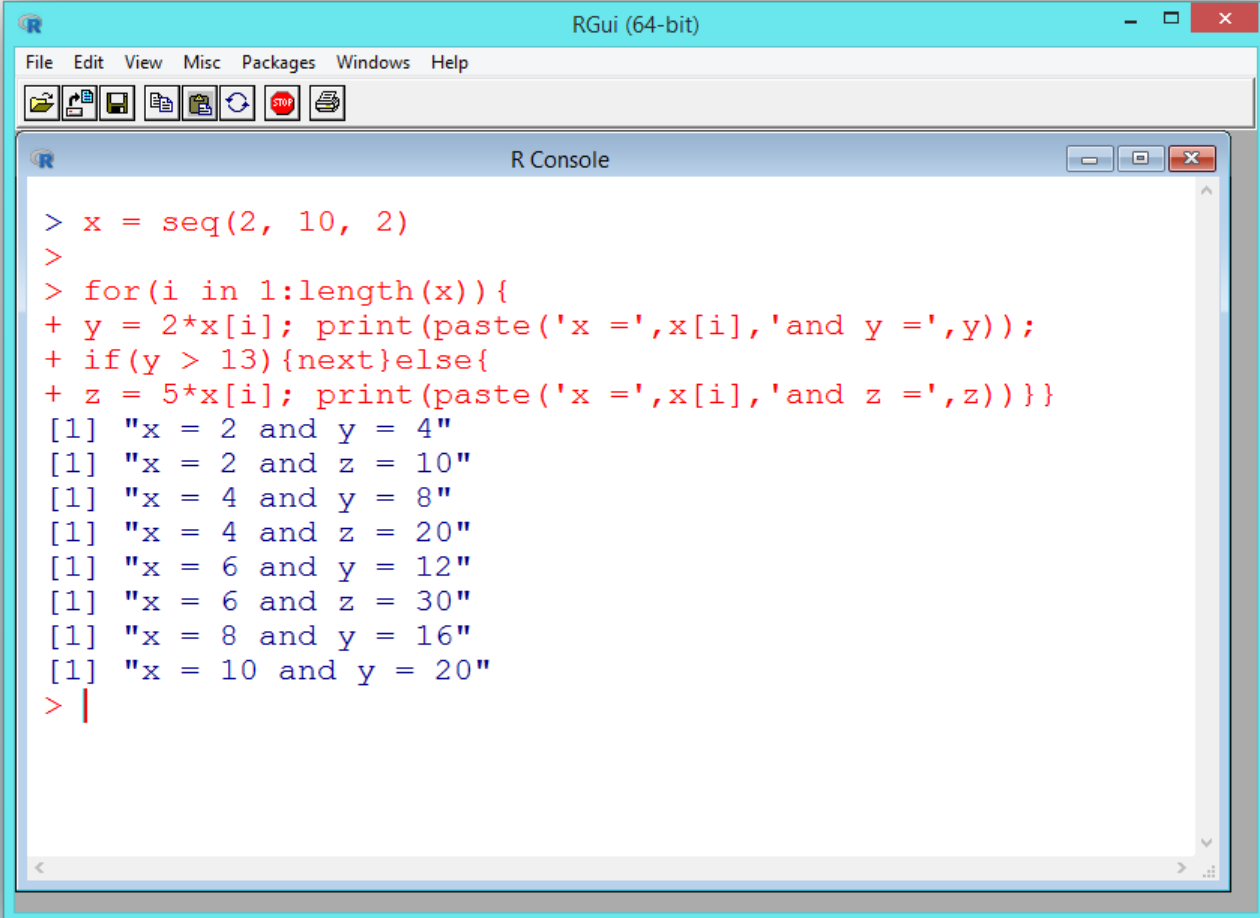
► break Control Command



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> x = seq(2, 10, 2)
>
> for(i in 1:length(x)){
+ y = 2*x[i]; print(paste('x =',x[i],'and y =',y));
+ if(y > 13){break}else{
+ z = 5*x[i]; print(paste('x =',x[i],'and z =',z))}
[1] "x = 2 and y = 4"
[1] "x = 2 and z = 10"
[1] "x = 4 and y = 8"
[1] "x = 4 and z = 20"
[1] "x = 6 and y = 12"
[1] "x = 6 and z = 30"
[1] "x = 8 and y = 16"
> |
```

Loop

► next Control Command



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

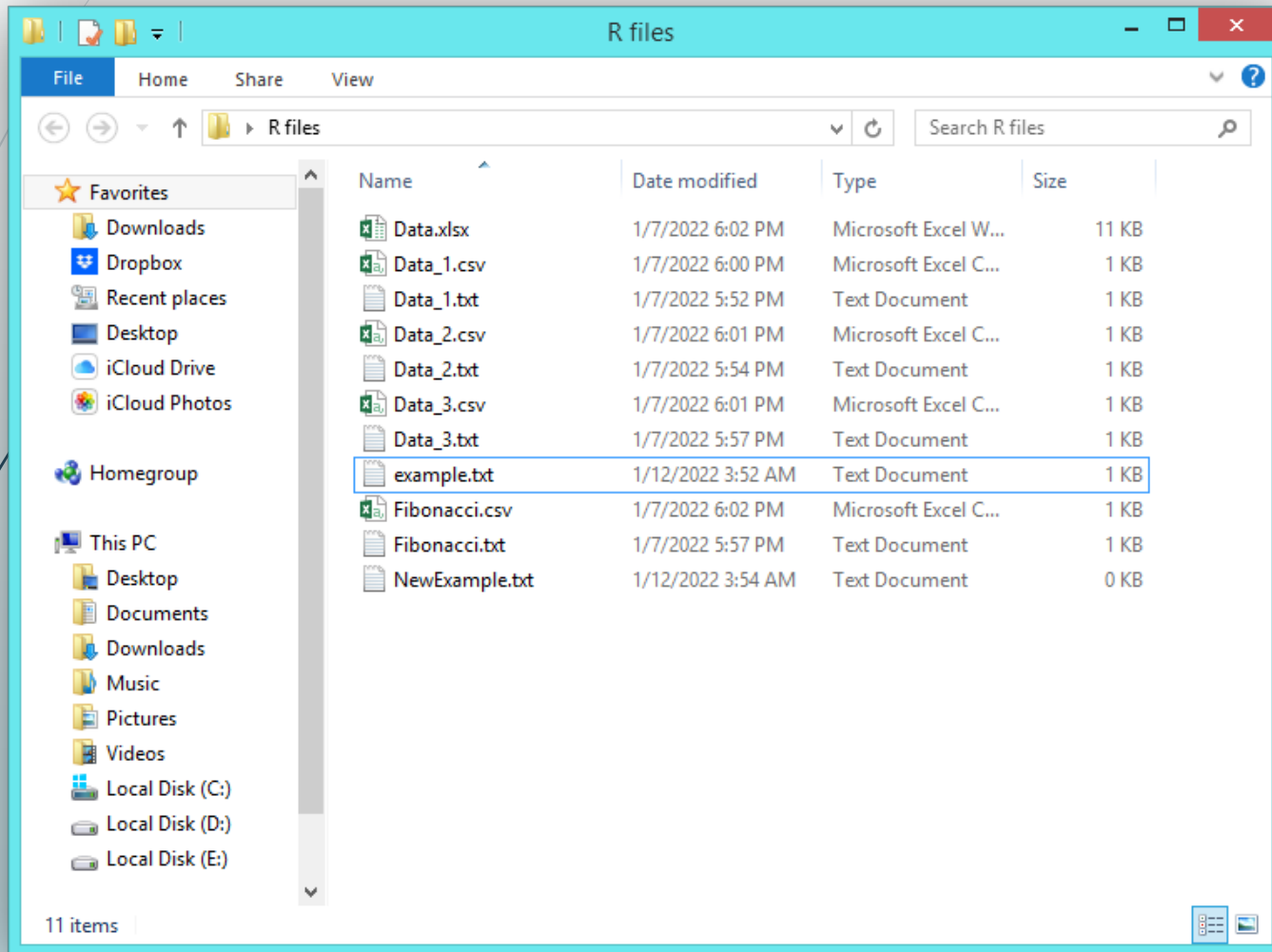
R Console
> x = seq(2, 10, 2)
>
> for(i in 1:length(x)){
+ y = 2*x[i]; print(paste('x =',x[i],'and y =',y));
+ if(y > 13){next}else{
+ z = 5*x[i]; print(paste('x =',x[i],'and z =',z))}
[1] "x = 2 and y = 4"
[1] "x = 2 and z = 10"
[1] "x = 4 and y = 8"
[1] "x = 4 and z = 20"
[1] "x = 6 and y = 12"
[1] "x = 6 and z = 30"
[1] "x = 8 and y = 16"
[1] "x = 10 and y = 20"
> |
```



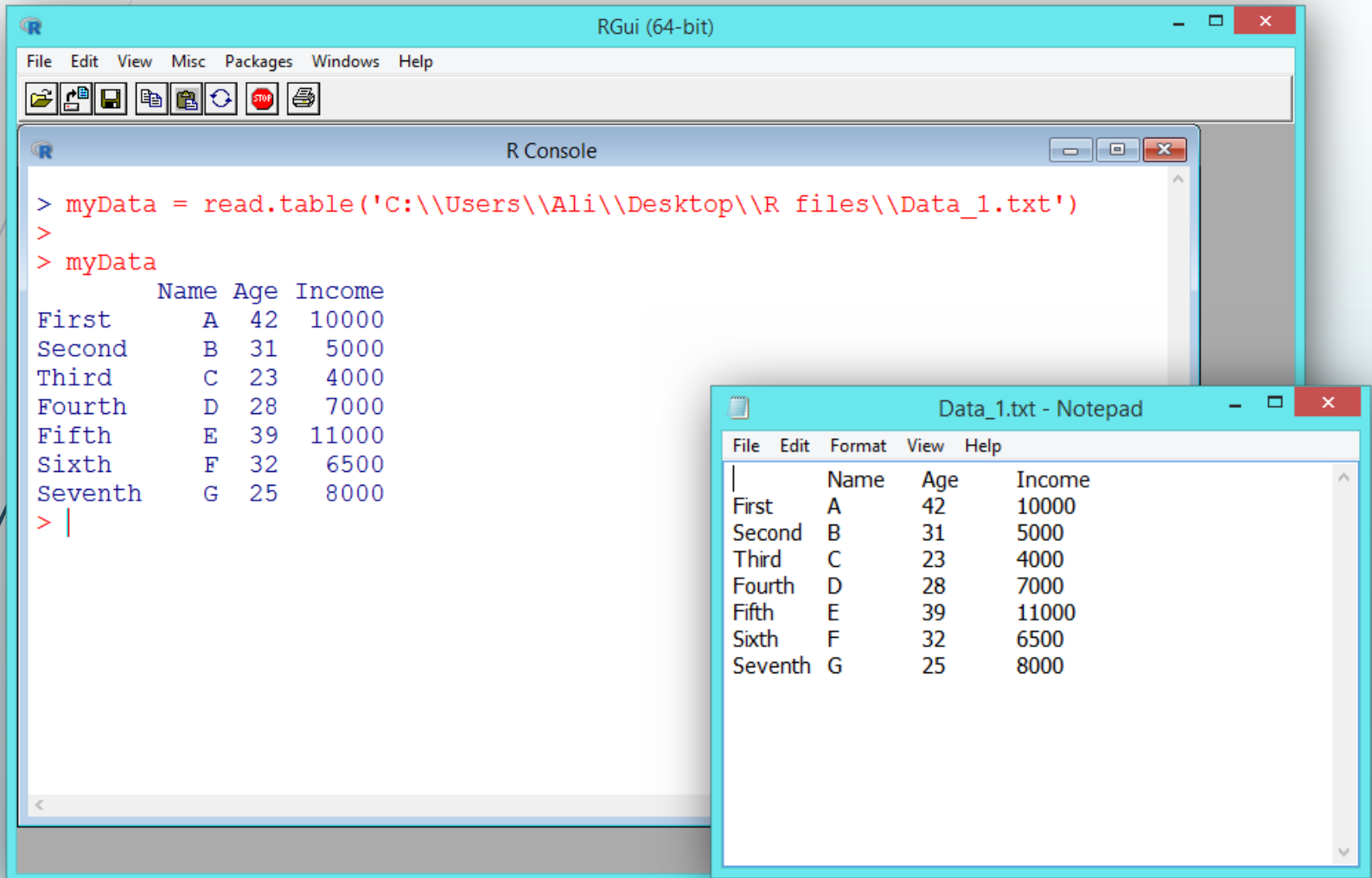
Reading and Writing

Working with data files

read.table()



read.table



The screenshot shows the RGui (64-bit) interface. The R Console window displays the following code and output:

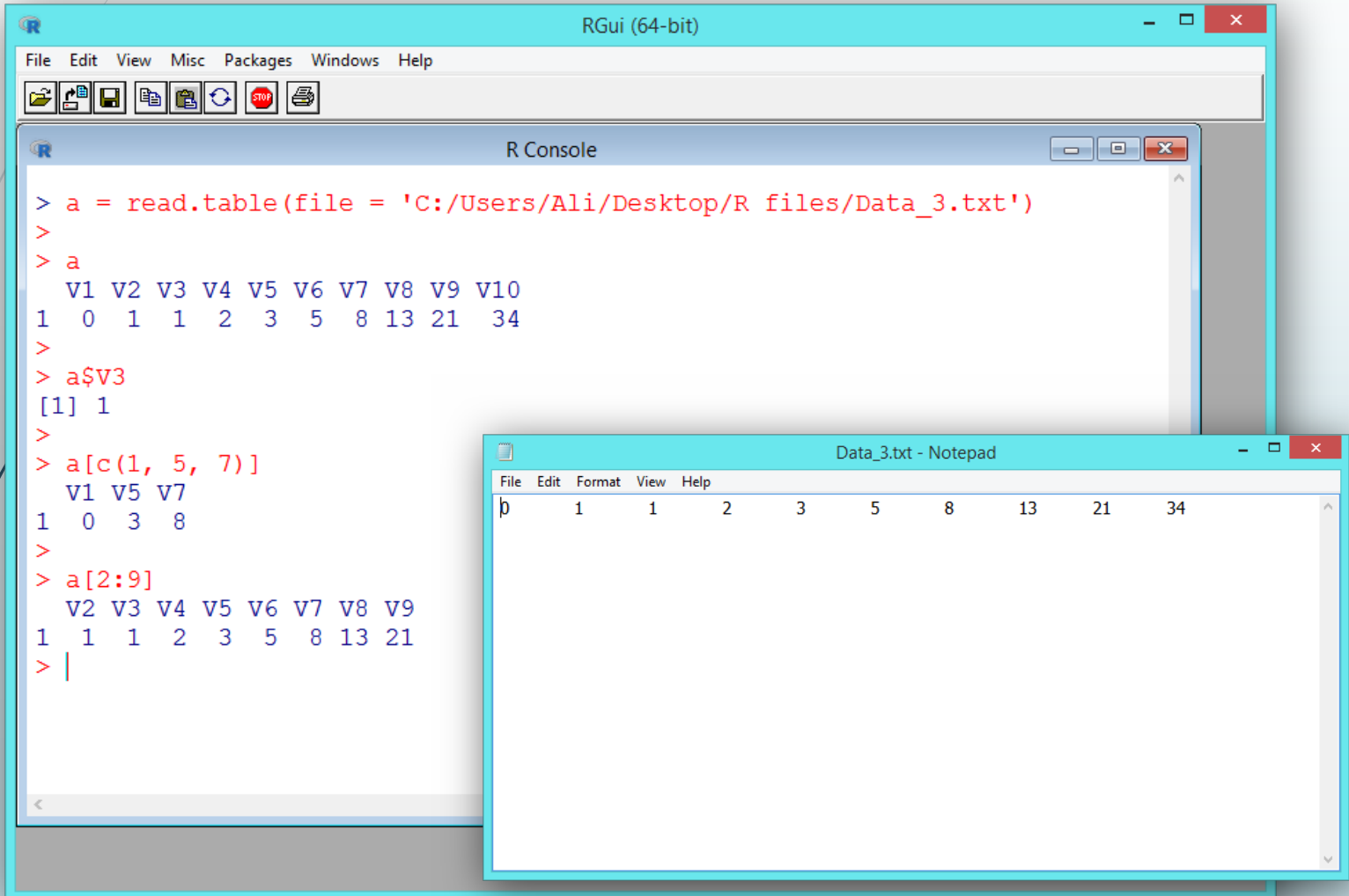
```
> myData = read.table('C:\\Users\\Ali\\Desktop\\R files\\Data_1.txt')
>
> myData
```

	Name	Age	Income
First	A	42	10000
Second	B	31	5000
Third	C	23	4000
Fourth	D	28	7000
Fifth	E	39	11000
Sixth	F	32	6500
Seventh	G	25	8000

The Notepad window shows the original content of the file 'Data_1.txt':

	Name	Age	Income
First	A	42	10000
Second	B	31	5000
Third	C	23	4000
Fourth	D	28	7000
Fifth	E	39	11000
Sixth	F	32	6500
Seventh	G	25	8000

read.table()



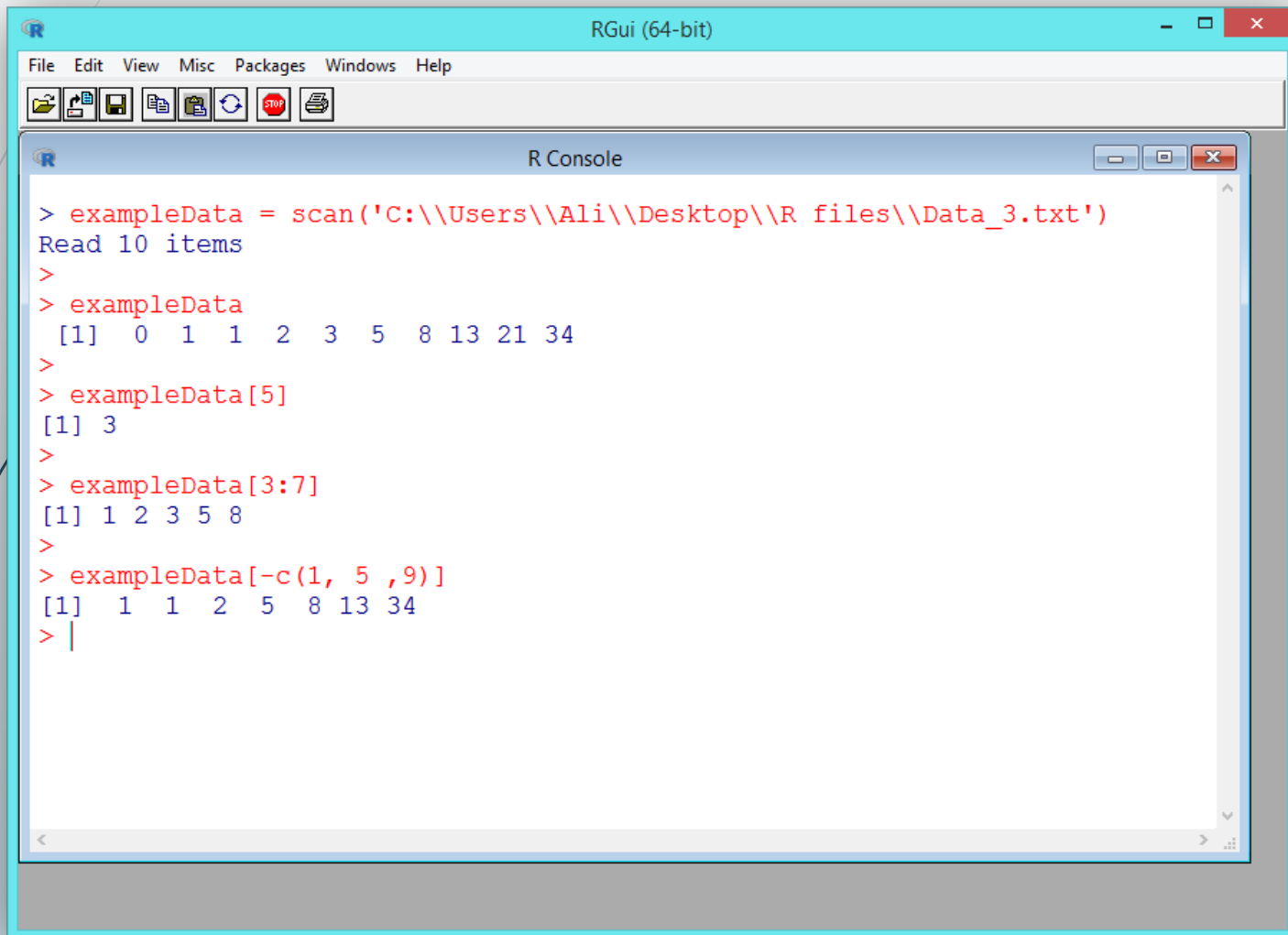
The screenshot displays the RGui (64-bit) interface. The R Console window shows the following commands and output:

```
> a = read.table(file = 'C:/Users/Ali/Desktop/R files/Data_3.txt')
>
> a
  V1 V2 V3 V4 V5 V6 V7 V8 V9 V10
1  0  1  1  2  3  5  8 13 21  34
>
> a$V3
[1] 1
>
> a[c(1, 5, 7)]
  V1 V5 V7
1  0  3  8
>
> a[2:9]
  V2 V3 V4 V5 V6 V7 V8 V9
1  1  1  2  3  5  8 13 21
> |
```

The Notepad window, titled 'Data_3.txt - Notepad', shows the contents of the file:

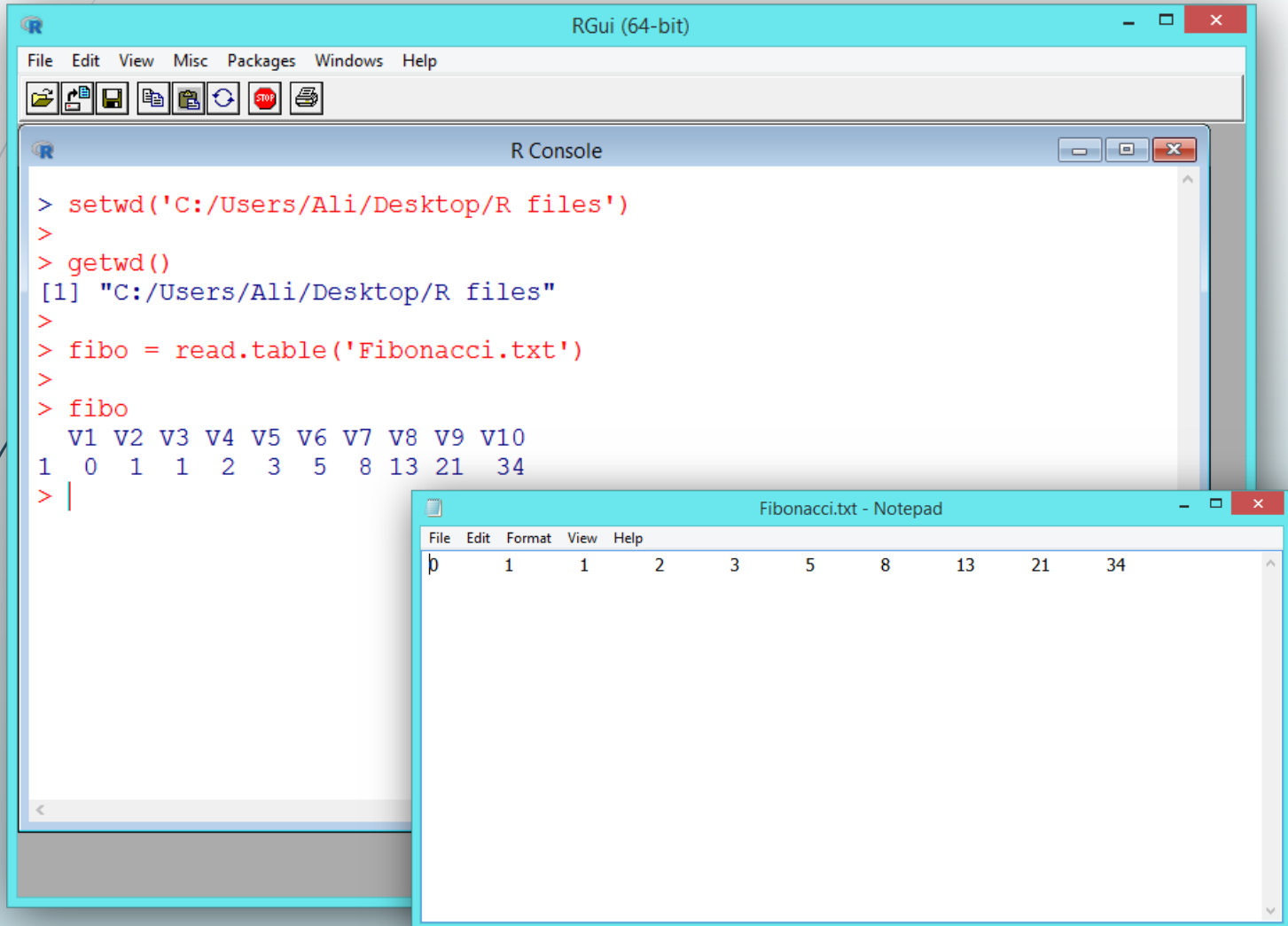
```
0      1      1      2      3      5      8      13      21      34
```

Scan ()



```
> exampleData = scan('C:\\\\Users\\Ali\\Desktop\\R files\\Data_3.txt')
Read 10 items
>
> exampleData
[1] 0 1 1 2 3 5 8 13 21 34
>
> exampleData[5]
[1] 3
>
> exampleData[3:7]
[1] 1 2 3 5 8
>
> exampleData[-c(1, 5, 9)]
[1] 1 1 2 5 8 13 34
> |
```

Setwd() & getwd()



The screenshot shows the RGui (64-bit) interface. The R Console window displays the following commands and output:

```
> setwd('C:/Users/Ali/Desktop/R files')
>
> getwd()
[1] "C:/Users/Ali/Desktop/R files"
>
> fibo = read.table('Fibonacci.txt')
>
> fibo
  V1 V2 V3 V4 V5 V6 V7 V8 V9 V10
1  0  1  1  2  3  5  8 13 21  34
```

The Fibonacci.txt - Notepad window shows the following content:

	1	1	2	3	5	8	13	21	34
0									

read.table()

```
RGui (64-bit)
File Edit View Misc Packages Windows Help

R Console
> Y = read.table(file = 'NewExample.txt', header = TRUE, sep = '-')
> Y
  A B C D E
1 4 5 8 9 h
2 2 95 66 32 m
3 9 1 7 2 n
>
> Z = read.table(file = 'NewExample.txt', header = TRUE, sep = '-', row.names = 5)
> Z
  A B C D
h 4 5 8 9
m 2 95 66 32
n 9 1 7 2
>
> W = read.table(file = 'NewExample.txt', header = TRUE, sep = '-', row.names = 1)
> W
  B C D E
4 5 8 9 h
2 95 66 32 m
9 1 7 2 n
> |

NewExample.txt - Notepad
File Edit Format View Help
A-B-C-D-E
4-5-8-9-h
2-95-66-32-m
9-1-7-2-n
```

read.table()

```

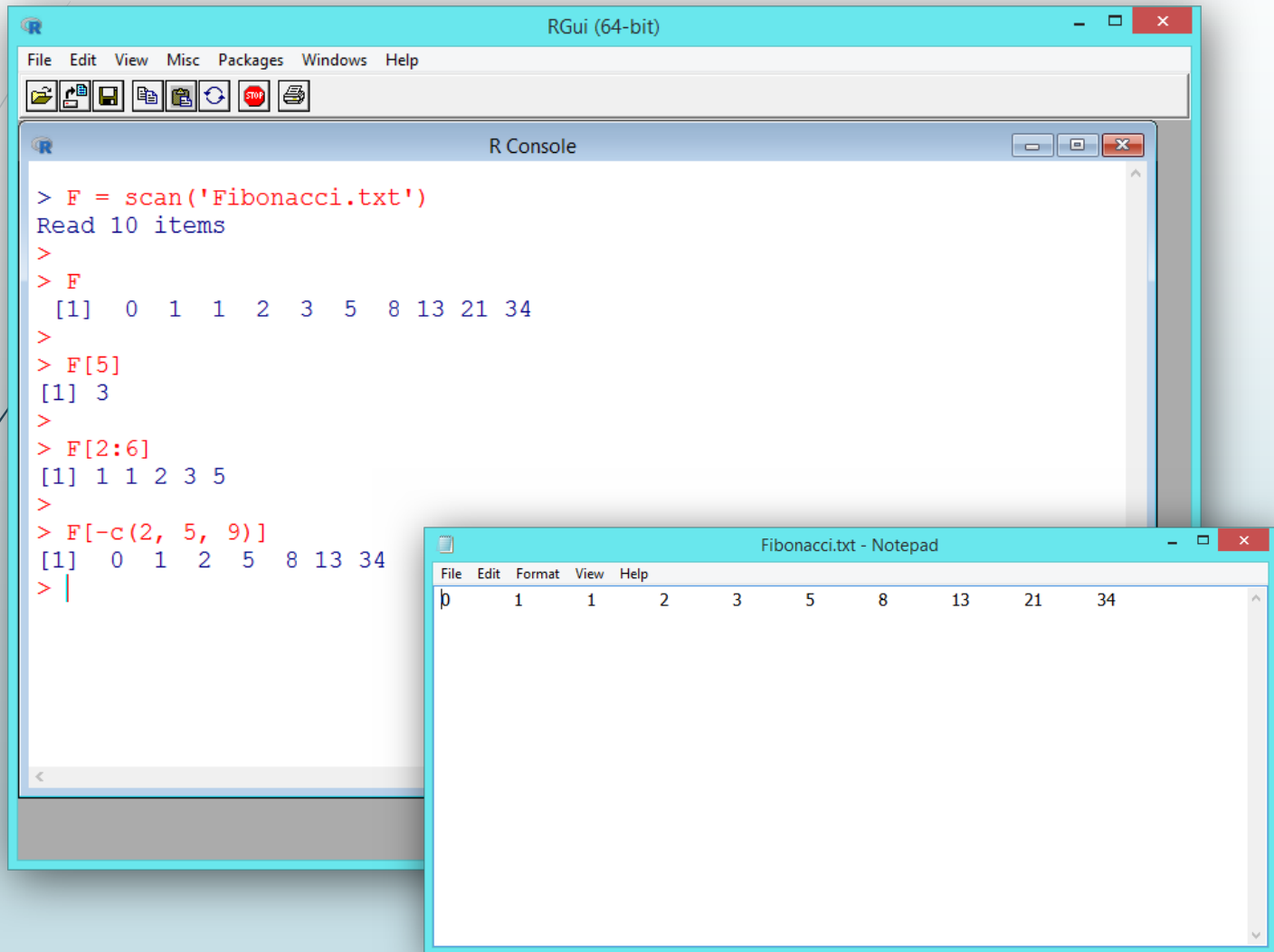
RGui (64-bit)
File Edit View Misc Packages Windows Help

R Console
> ex = read.table('example.txt')
Error in scan(file = file, what = what, sep = sep, quote = quote, dec $
  line 1 did not have 5 elements
> ex = read.table('example.txt', skip = 2)
> ex
      V1
1  A,B,C,D,E
2  a,5,5,8,9,h
3 b,95,66,32,0,m
4  c,1,7,16,2,n
> ex = read.table('example.txt', skip = 2, header = TRUE)
> ex
      A.B.C.D.E
1  a,5,5,8,9,h
2 b,95,66,32,0,m
3  c,1,7,16,2,n
> ex = read.table('example.txt', skip = 2, header = TRUE, sep = ',')
> ex
  A B C D E
a 5 5 8 9 h
b 95 66 32 0 m
c 1 7 16 2 n
> |

example.txt - Notepad
File Edit Format View Help
Hi guys!
This is an example file.
A,B,C,D,E
a,5,5,8,9,h
b,95,66,32,0,m
c,1,7,16,2,n

```

Scan ()



```
> F = scan('Fibonacci.txt')
Read 10 items
>
> F
[1] 0 1 1 2 3 5 8 13 21 34
>
> F[5]
[1] 3
>
> F[2:6]
[1] 1 1 2 3 5
>
> F[-c(2, 5, 9)]
[1] 0 1 2 5 8 13 34
> |
```

Fibonacci.txt - Notepad

File	Edit	Format	View	Help					
0	1	1	2	3	5	8	13	21	34

read.csv()

```

RGui (64-bit)
File Edit View Misc Packages Windows Help

R Console
> csvdata = read.csv(file = 'Data_1.csv')
> csvdata
  Name Age Income
1    A  42 10000
2    B  31  5000
3    C  23  4000
4    D  28  7000
5    E  39 11000
6    F  32  6500
7    G  25  8000
>
> CSVDATA = read.csv(file = 'Data_2.csv')
> CSVDATA
  X Name Age Income
1 First  A  42 10000
2 Second B  31  5000
3 Third  C  23  4000
4 Fourth D  28  7000
5 Fifth  E  39 11000
6 Sixth  F  32  6500
7 Seventh G  25  8000
>

```

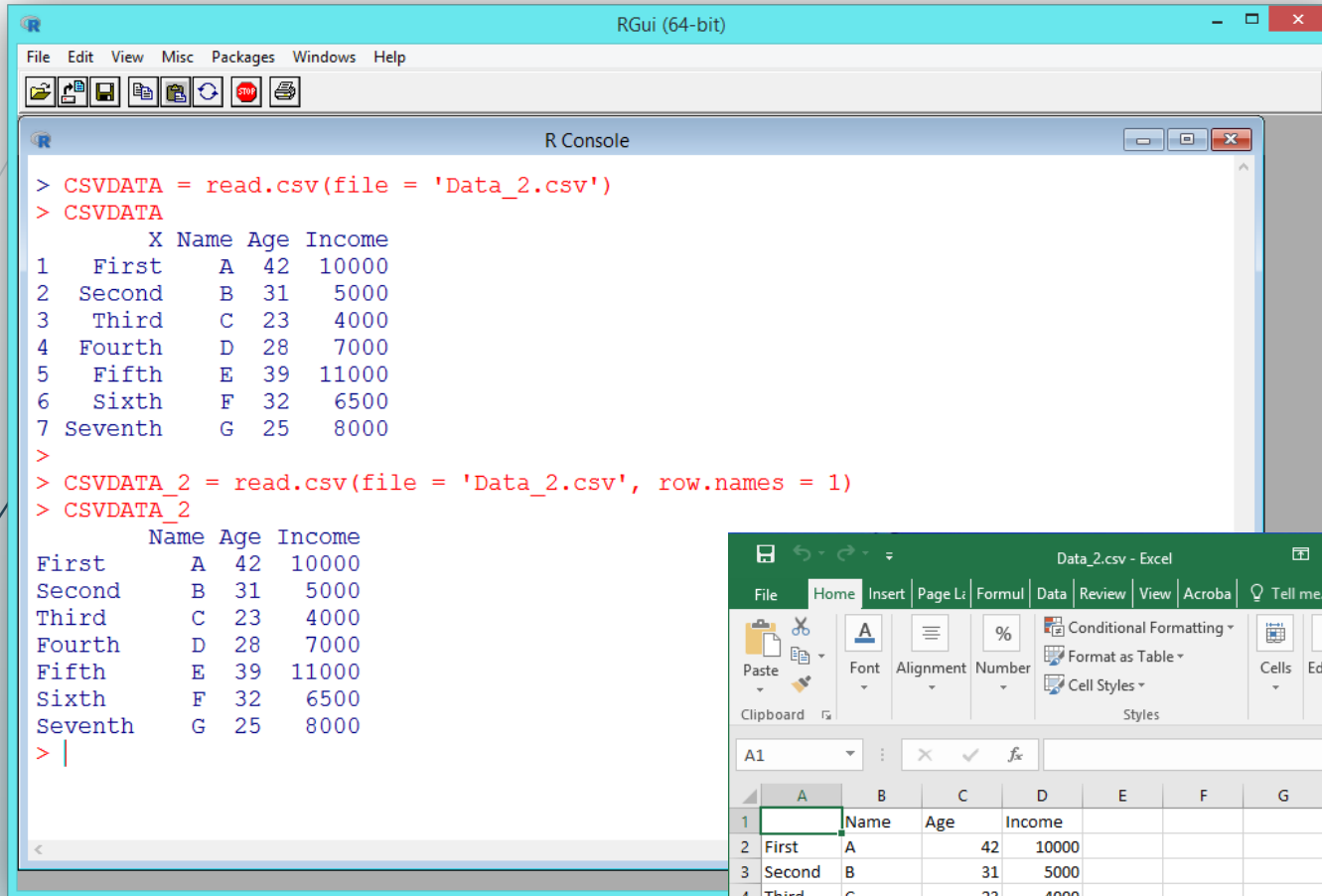
Data_1.csv - Excel

Name	Age	Income
A	42	10000
B	31	5000
C	23	4000
D	28	7000
E	39	11000
F	32	6500
G	25	8000

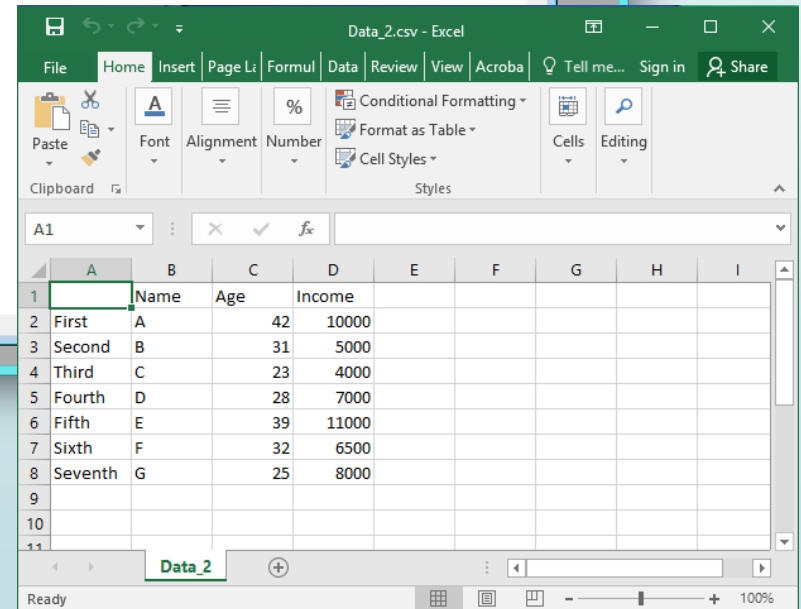
Data_2.csv - Excel

X	Name	Age	Income	
1	First	A	42	10000
2	Second	B	31	5000
3	Third	C	23	4000
4	Fourth	D	28	7000
5	Fifth	E	39	11000
6	Sixth	F	32	6500
7	Seventh	G	25	8000

read.csv()

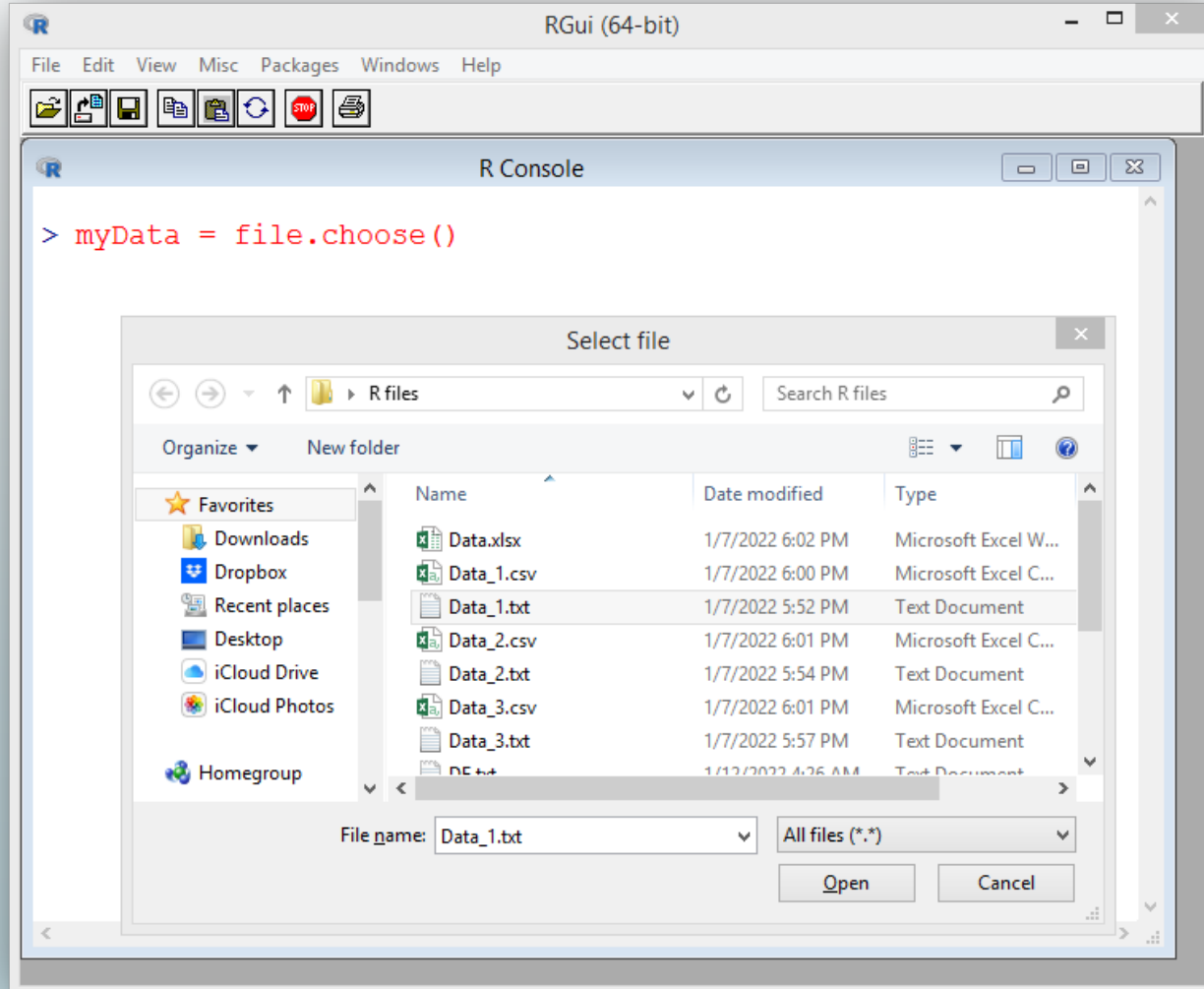


```
> CSVDATA = read.csv(file = 'Data_2.csv')
> CSVDATA
  X Name Age Income
1  First  A  42 10000
2  Second B  31  5000
3  Third  C  23  4000
4  Fourth D  28  7000
5  Fifth  E  39 11000
6  Sixth  F  32  6500
7  Seventh G  25  8000
>
> CSVDATA_2 = read.csv(file = 'Data_2.csv', row.names = 1)
> CSVDATA_2
  Name Age Income
First  A  42 10000
Second B  31  5000
Third  C  23  4000
Fourth D  28  7000
Fifth  E  39 11000
Sixth  F  32  6500
Seventh G  25  8000
> |
```

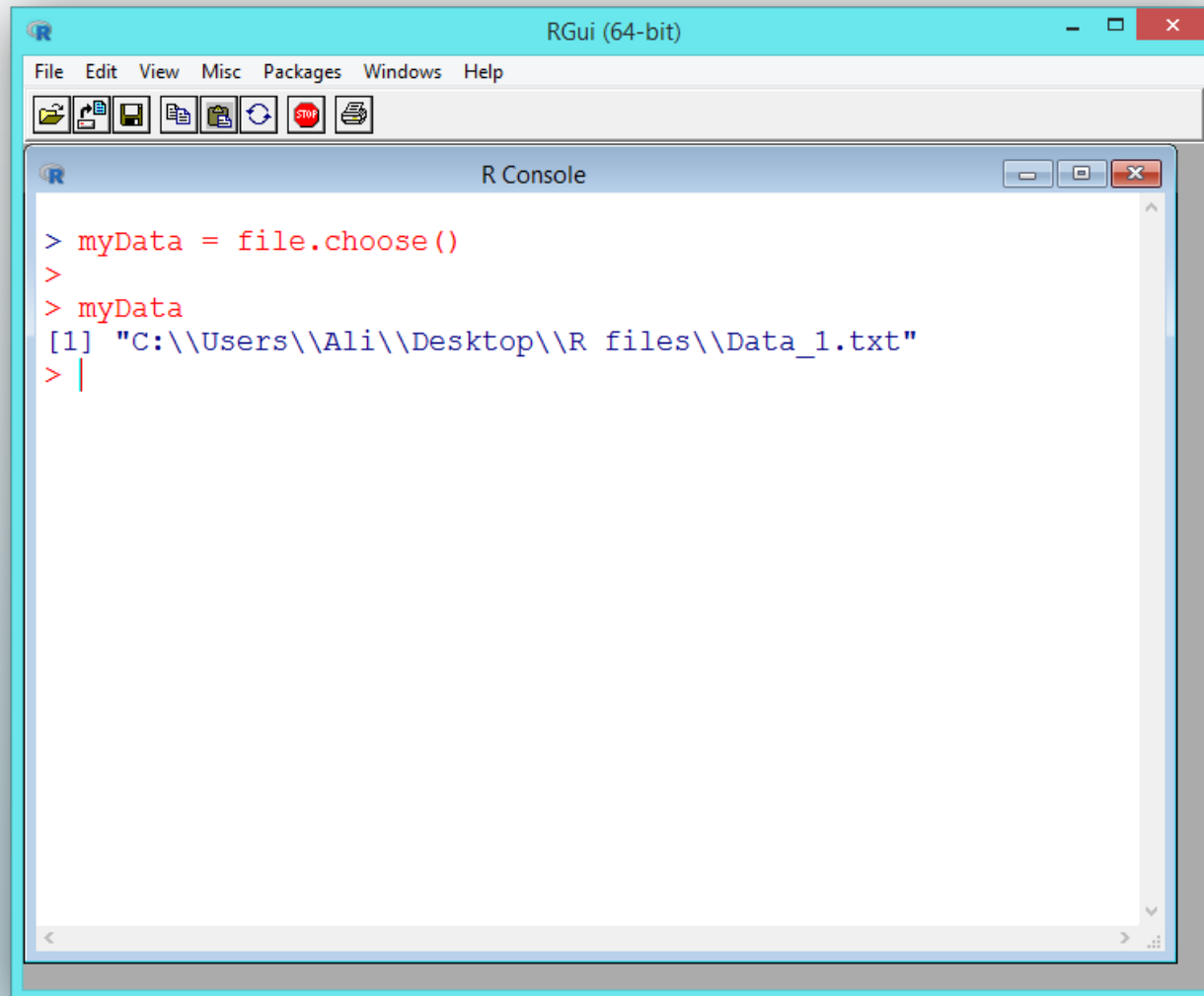


	A	B	C	D	E	F	G	H	I
1		Name	Age	Income					
2	First	A	42	10000					
3	Second	B	31	5000					
4	Third	C	23	4000					
5	Fourth	D	28	7000					
6	Fifth	E	39	11000					
7	Sixth	F	32	6500					
8	Seventh	G	25	8000					
9									
10									
11									

file.choose()



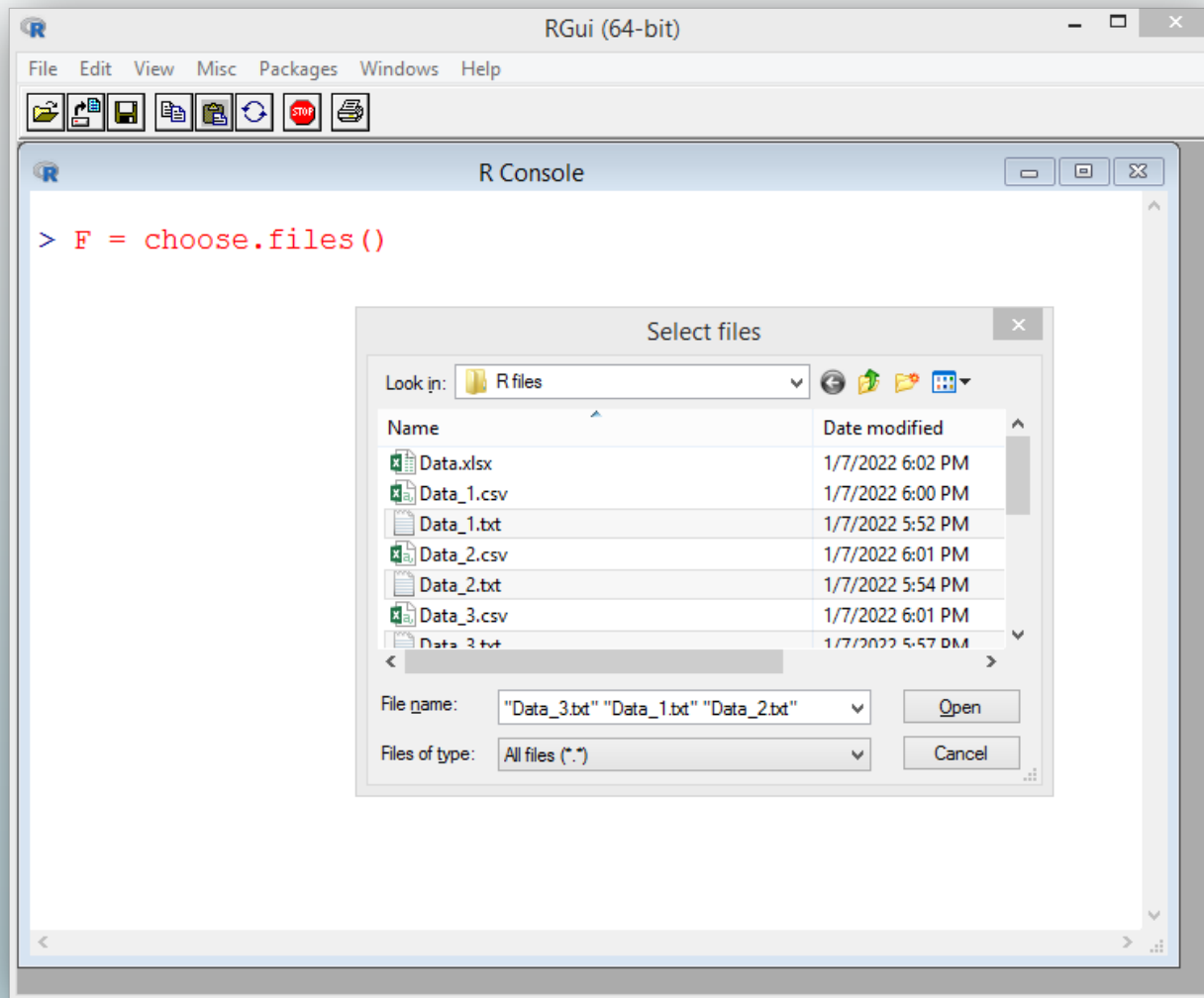
file.choose()



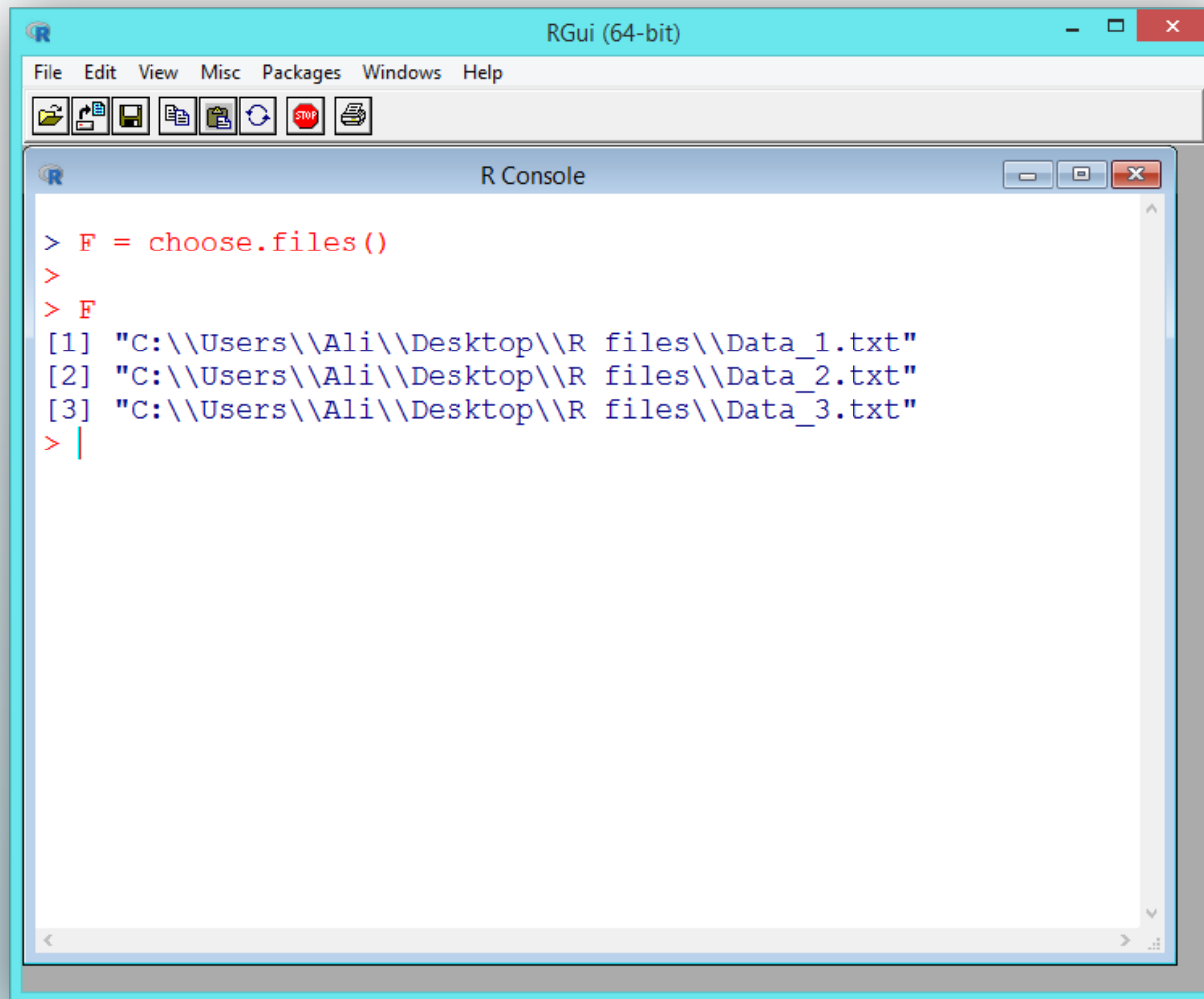
The screenshot displays the RGui (64-bit) window. The menu bar includes File, Edit, View, Misc, Packages, Windows, and Help. The toolbar contains icons for file operations. The R Console window shows the following code and output:

```
> myData = file.choose()
>
> myData
[1] "C:\\Users\\Ali\\Desktop\\R files\\Data_1.txt"
> |
```

choose.files()



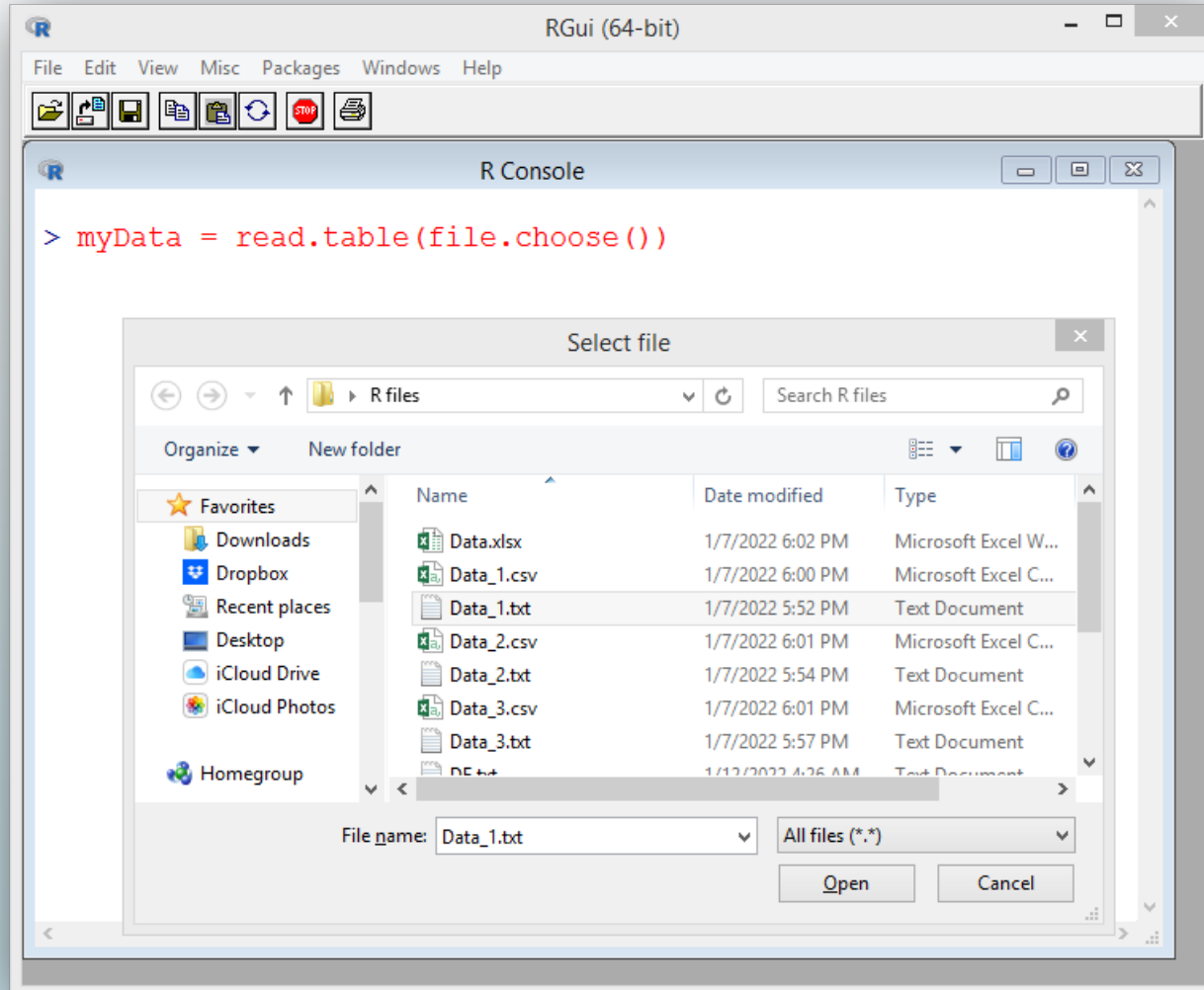
choose.files()



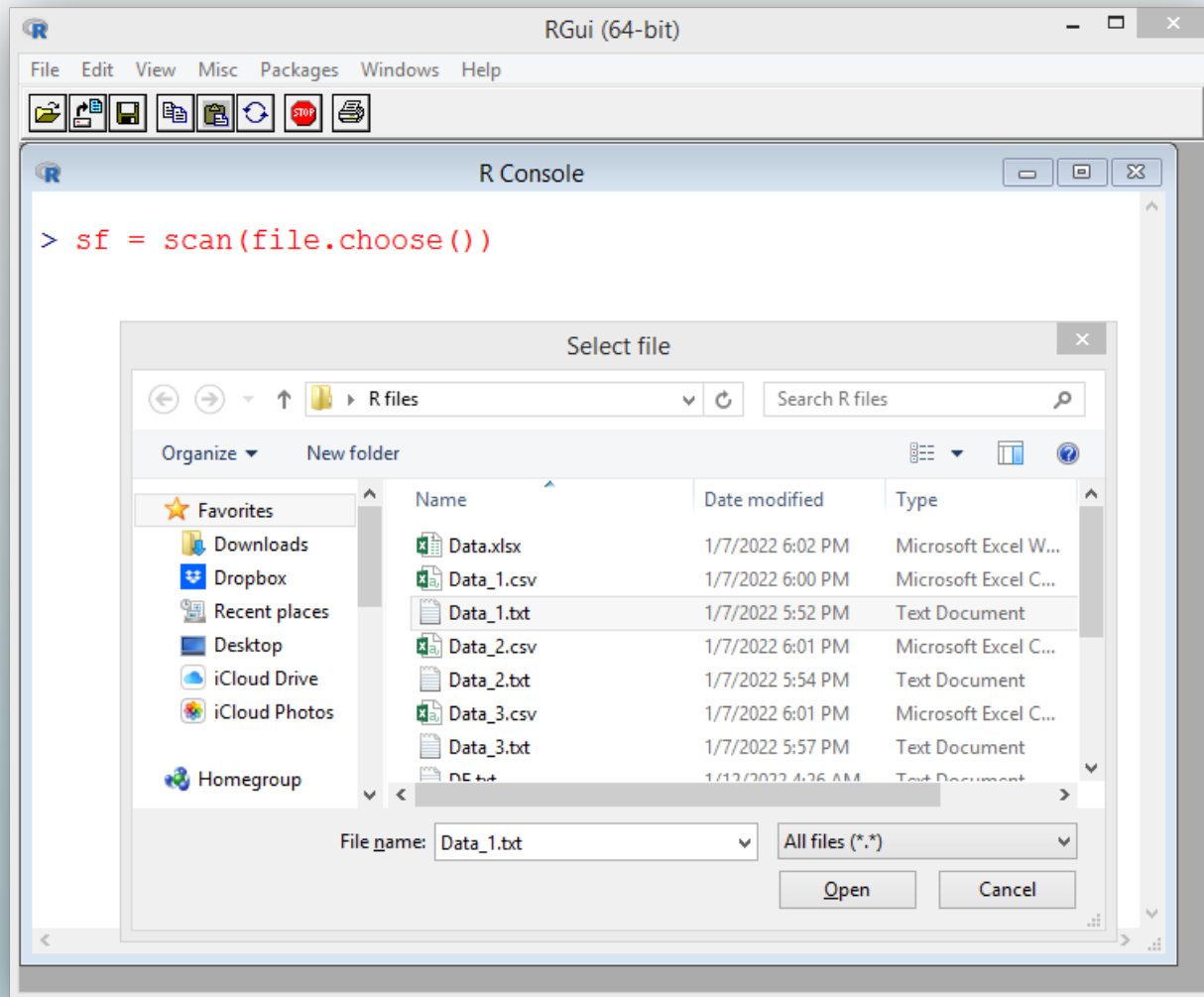
The screenshot shows the RGui (64-bit) interface. The R Console window displays the following code and output:

```
> F = choose.files()
>
> F
[1] "C:\\Users\\Ali\\Desktop\\R files\\Data_1.txt"
[2] "C:\\Users\\Ali\\Desktop\\R files\\Data_2.txt"
[3] "C:\\Users\\Ali\\Desktop\\R files\\Data_3.txt"
> |
```

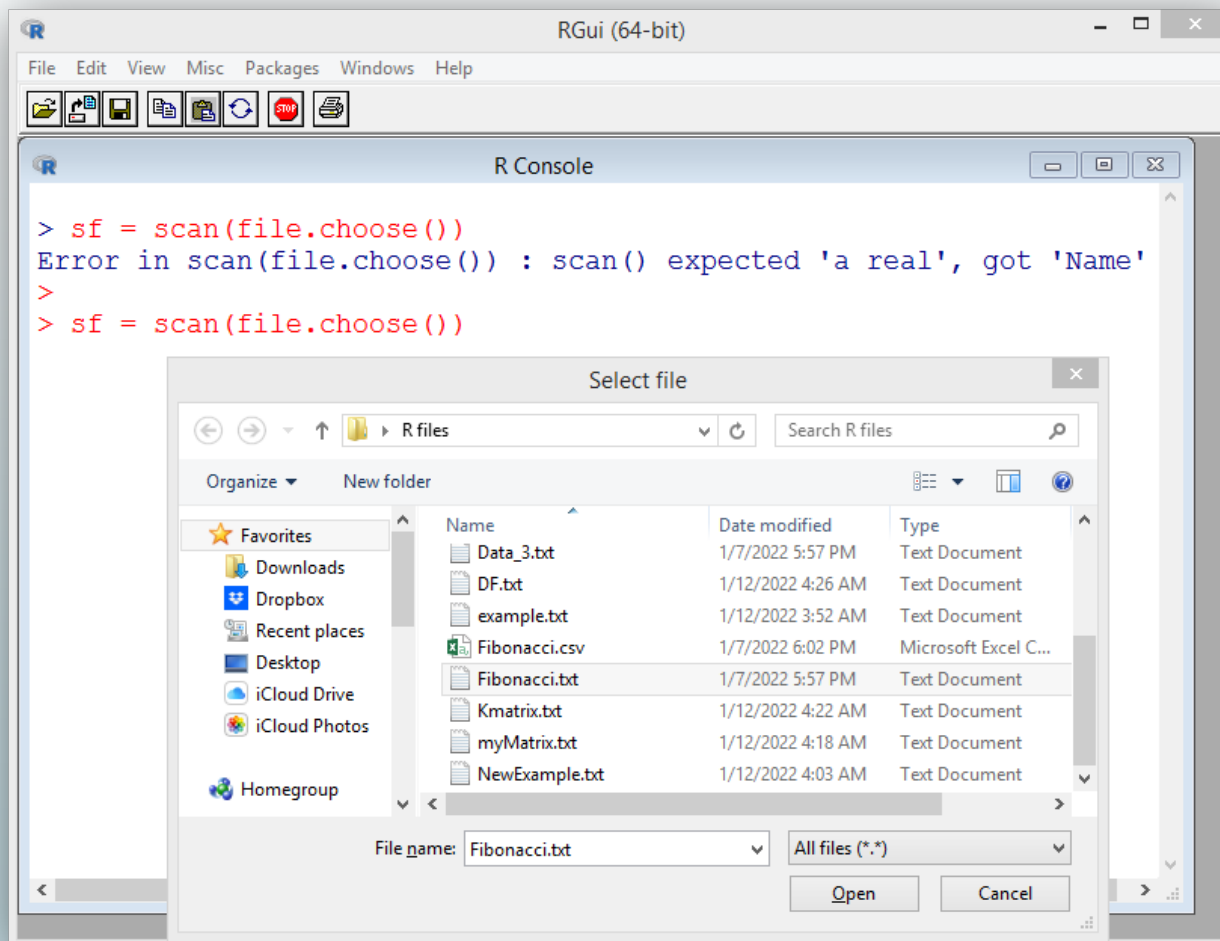
```
read.table(file.choose())
```



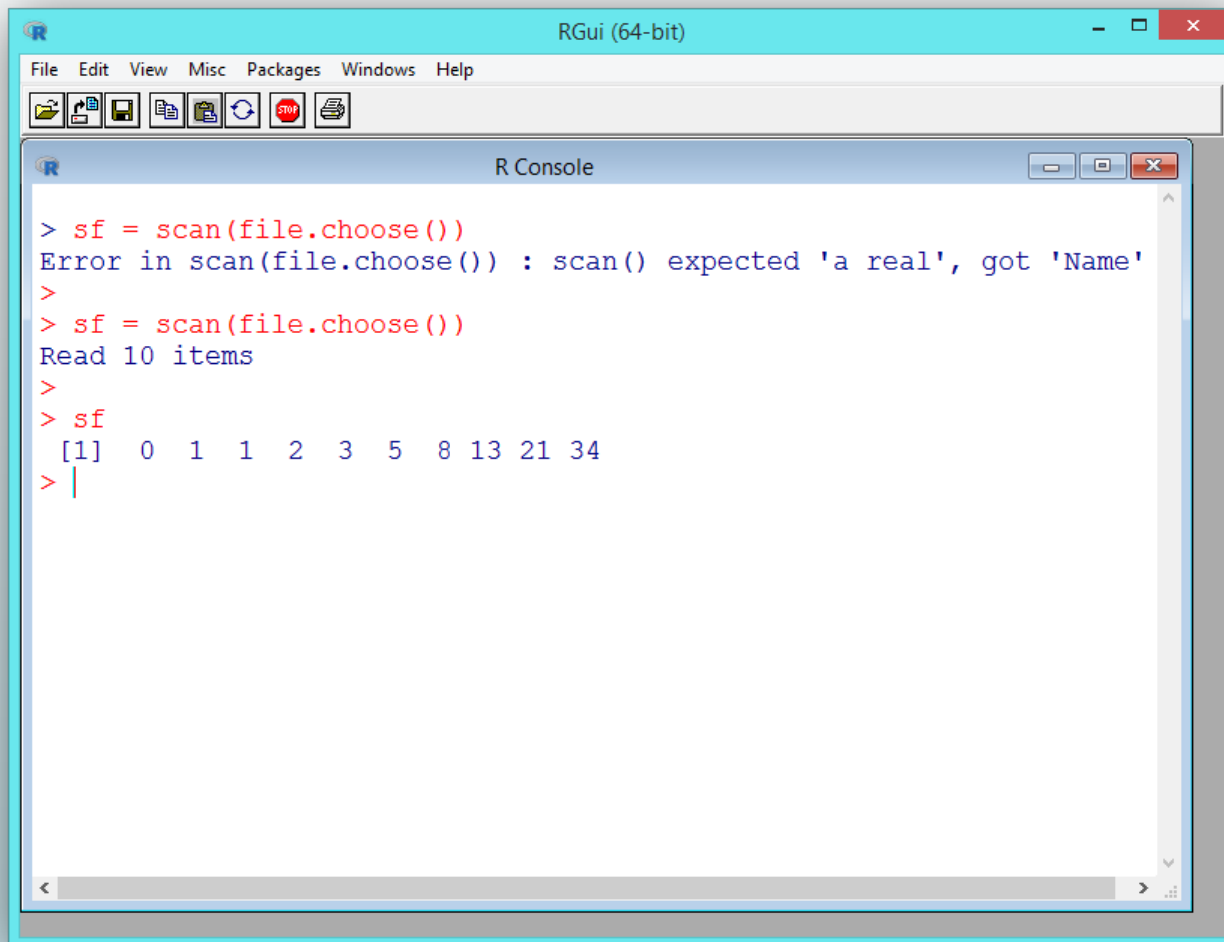
scan(file.choose())



scan(file.choose())



scan(file.choose())



```
> sf = scan(file.choose())
Error in scan(file.choose()) : scan() expected 'a real', got 'Name'
>
> sf = scan(file.choose())
Read 10 items
>
> sf
[1] 0 1 1 2 3 5 8 13 21 34
> |
```

write.table()

```
> M = matrix(1:9, nr = 3)
> M
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
>
> write.table(M, file = 'myMatrix.txt')
> |
```

```
File Edit Format View Help
|'V1' "V2" "V3"
"1" 1 4 7
"2" 2 5 8
"3" 3 6 9
```

write.table()

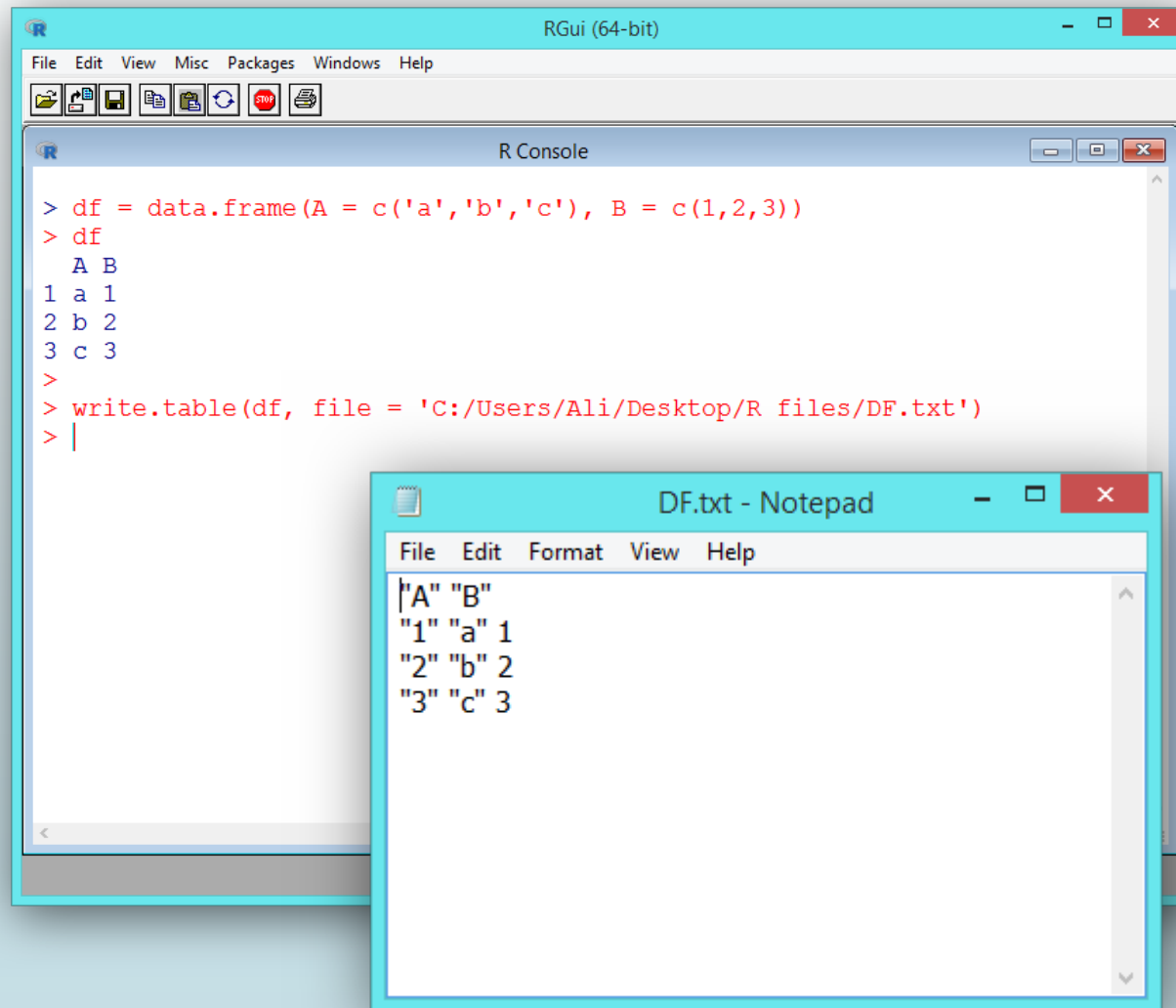
The image shows two windows from an R environment. The top window is the R Console, titled 'R Console', with a menu bar (File, Edit, View, Misc, Packages, Windows, Help) and a toolbar. The console contains the following R code and output:

```
> K = matrix(c(2,5,7,9), 2, 2, dimnames = list(c('a','b'),c('A','B')))
> K
  A B
a 2 7
b 5 9
>
> write.table(K, 'Kmatrix.txt')
> |
```

The bottom window is a Notepad window titled 'Kmatrix.txt - Notepad', with a menu bar (File, Edit, Format, View, Help). The text in the Notepad window is:

```
"A" "B"
"a" 2 7
"b" 5 9
```

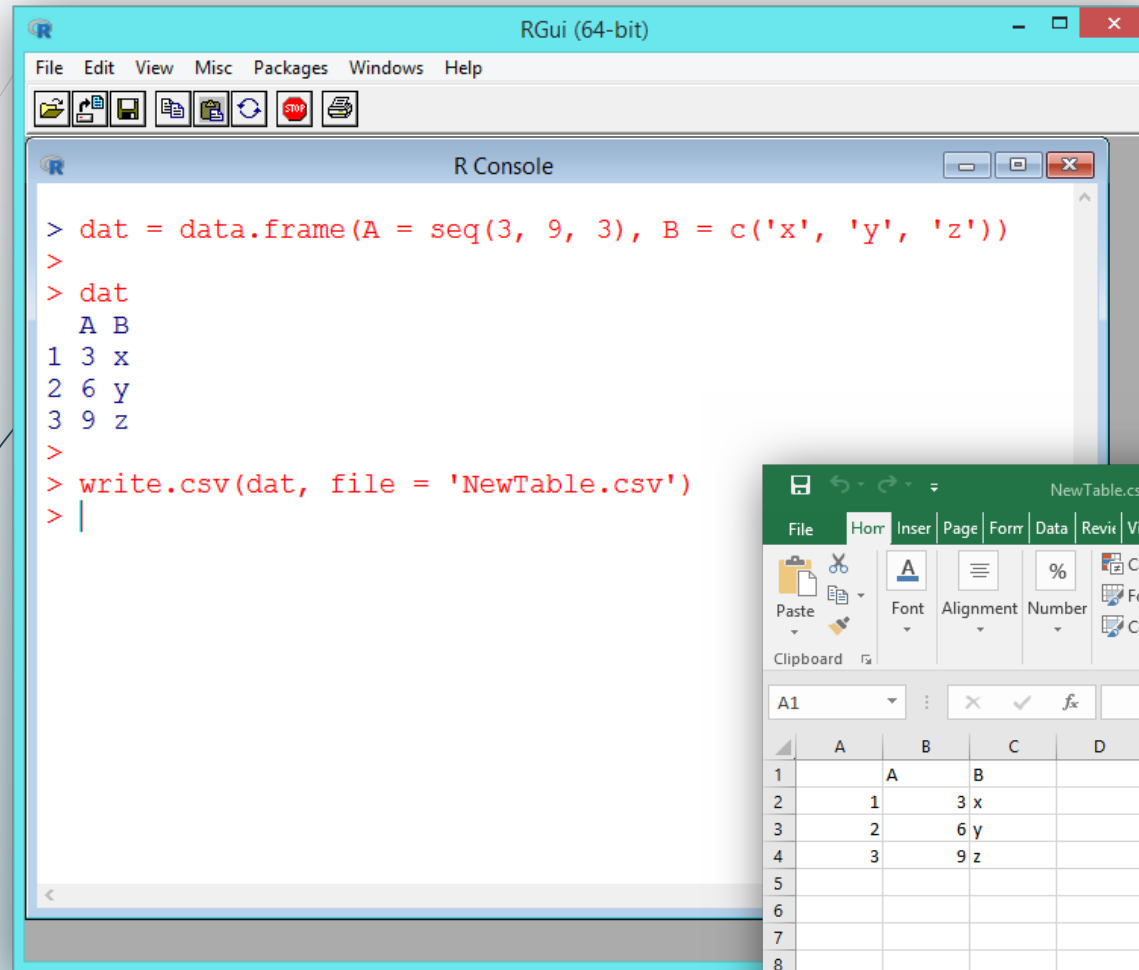
write.table()



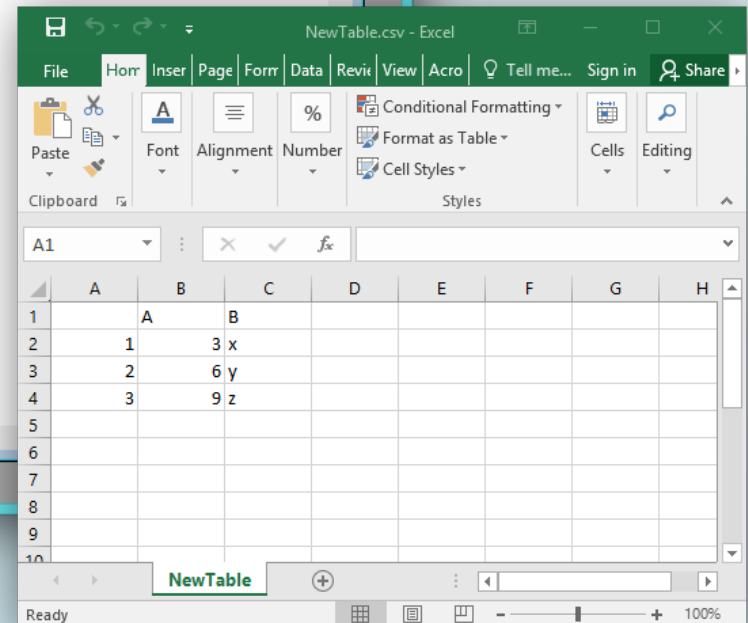
```
> df = data.frame(A = c('a','b','c'), B = c(1,2,3))
> df
  A B
1 a 1
2 b 2
3 c 3
>
> write.table(df, file = 'C:/Users/Ali/Desktop/R files/DF.txt')
> |
```

```
File Edit Format View Help
|A" "B"
|1" "a" 1
|2" "b" 2
|3" "c" 3
```

write.csv()



```
> dat = data.frame(A = seq(3, 9, 3), B = c('x', 'y', 'z'))
>
> dat
  A B
1 3 x
2 6 y
3 9 z
>
> write.csv(dat, file = 'NewTable.csv')
> |
```

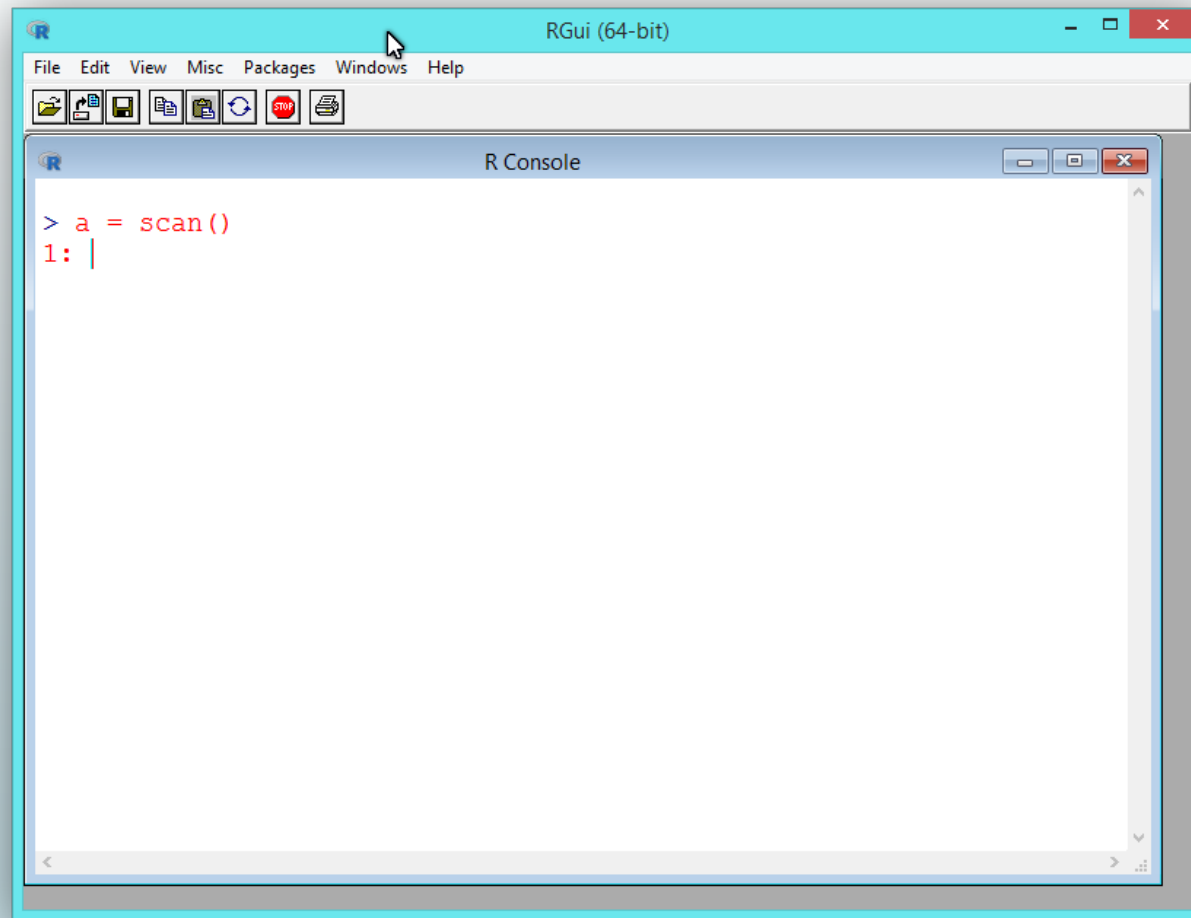


NewTable.csv - Excel

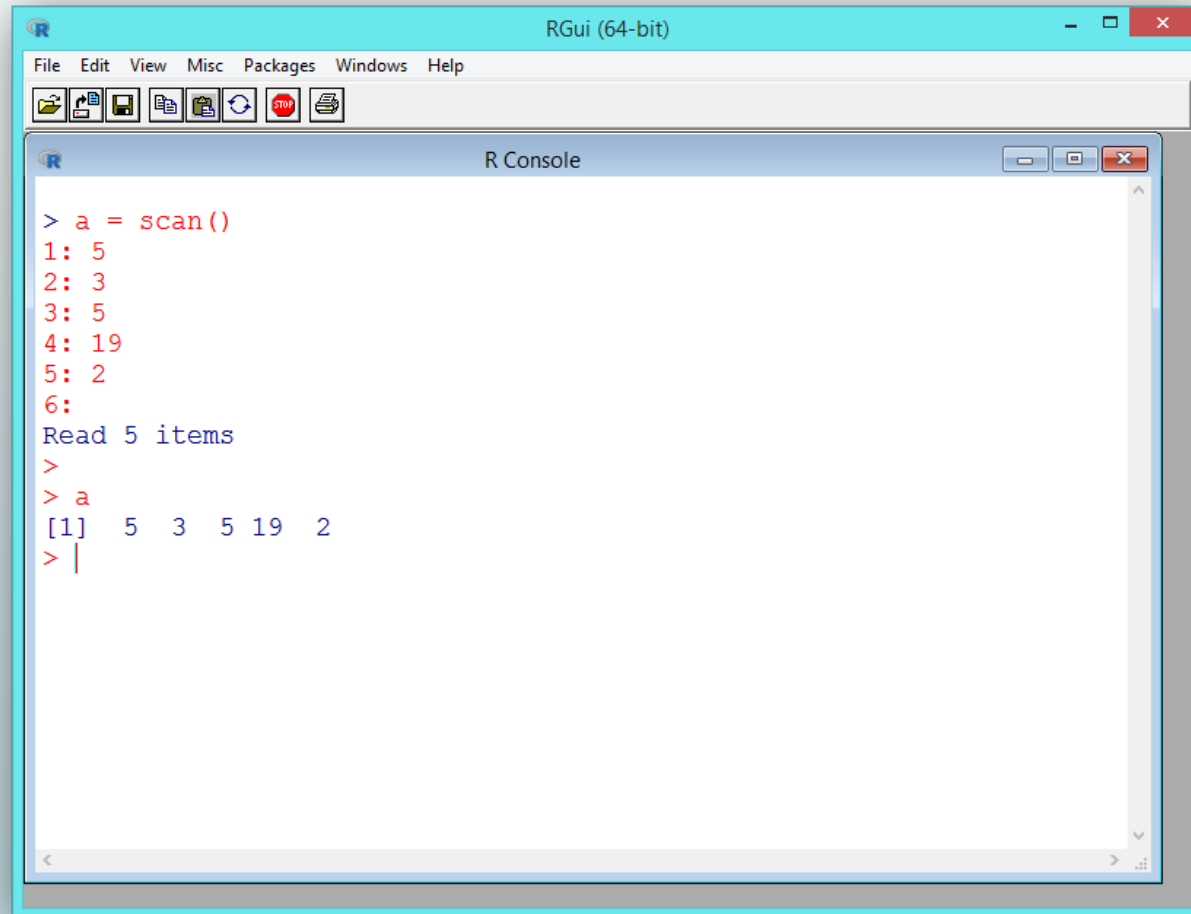
	A	B	C	D	E	F	G	H
1		A	B					
2	1	3	x					
3	2	6	y					
4	3	9	z					
5								
6								
7								
8								
9								
10								

202

scan ()



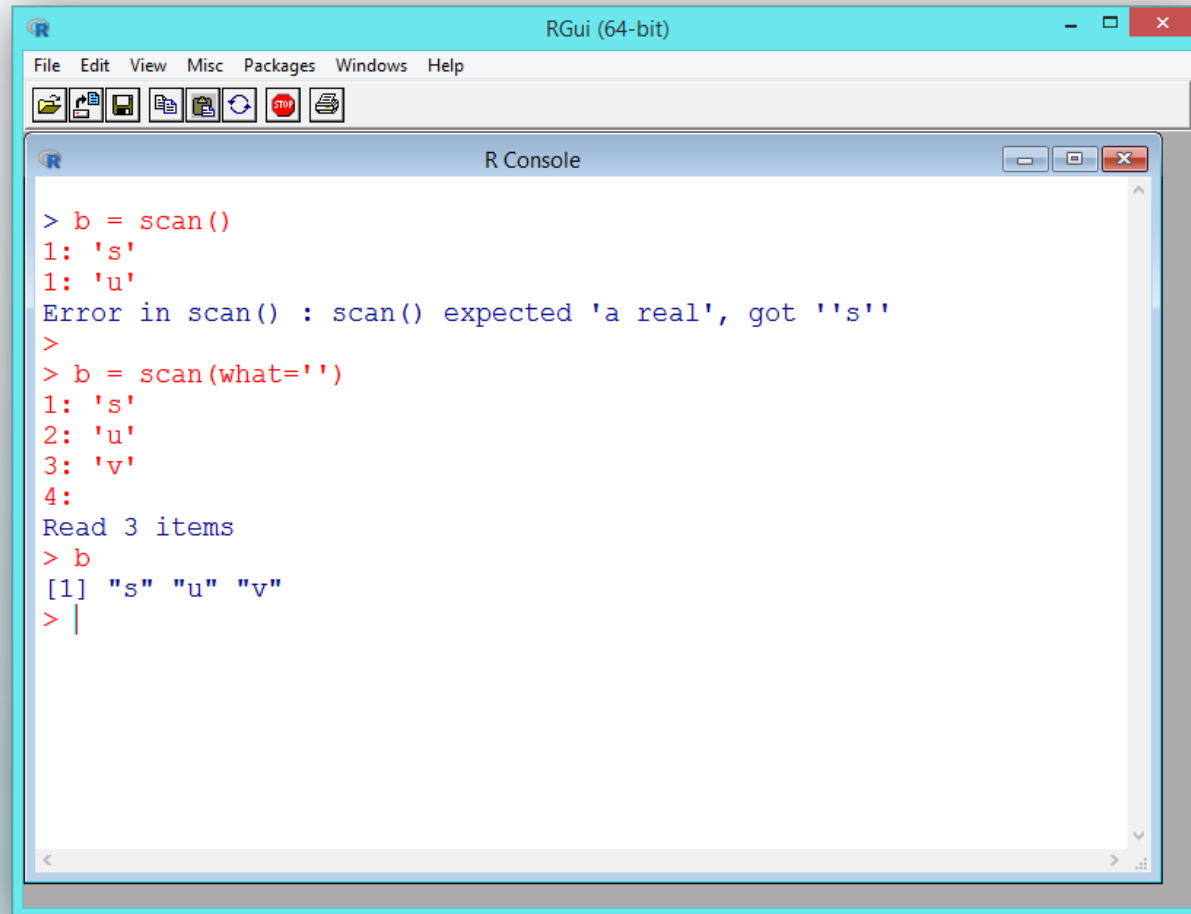
scan ()



The screenshot shows the RGui (64-bit) window with the R Console. The console displays the following output:

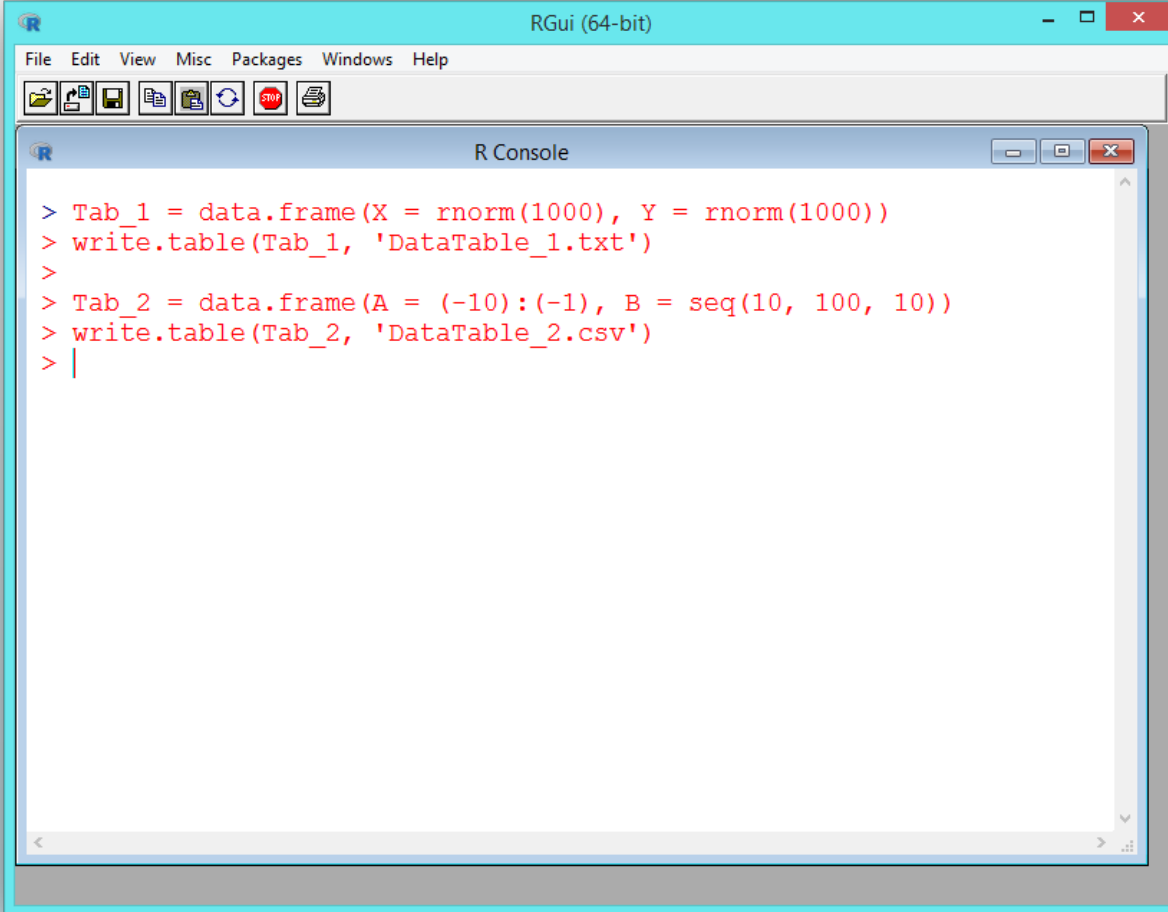
```
> a = scan()  
1: 5  
2: 3  
3: 5  
4: 19  
5: 2  
6:  
Read 5 items  
>  
> a  
[1] 5 3 5 19 2  
> |
```

scan ()



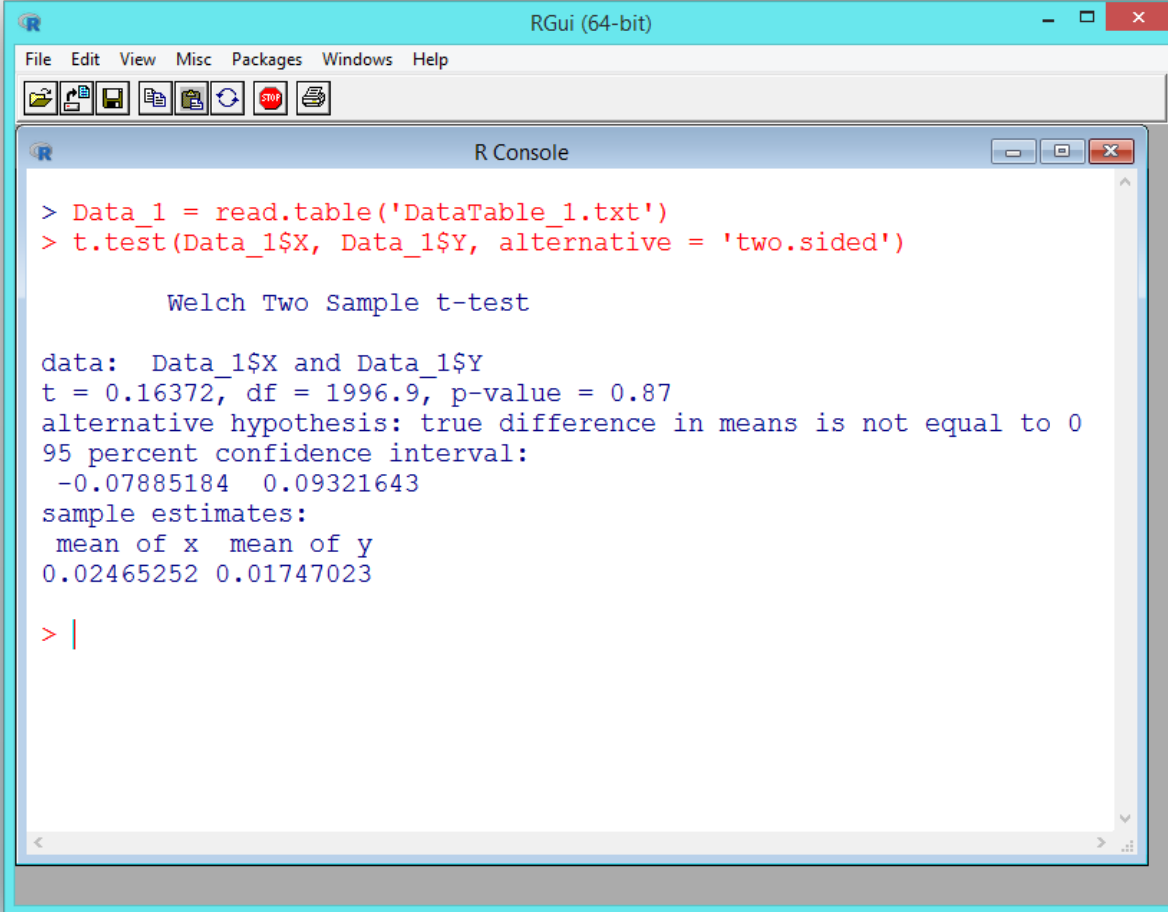
```
> b = scan()
1: 's'
1: 'u'
Error in scan() : scan() expected 'a real', got ''s''
>
> b = scan(what='')
1: 's'
2: 'u'
3: 'v'
4:
Read 3 items
> b
[1] "s" "u" "v"
> |
```


Read and Write Data - T-test Example



```
> Tab_1 = data.frame(X = rnorm(1000), Y = rnorm(1000))
> write.table(Tab_1, 'DataTable_1.txt')
>
> Tab_2 = data.frame(A = (-10):(-1), B = seq(10, 100, 10))
> write.table(Tab_2, 'DataTable_2.csv')
> |
```

Read and Write Data - T-test Example



The screenshot shows the RGui (64-bit) window with the R Console open. The console displays the following R code and output:

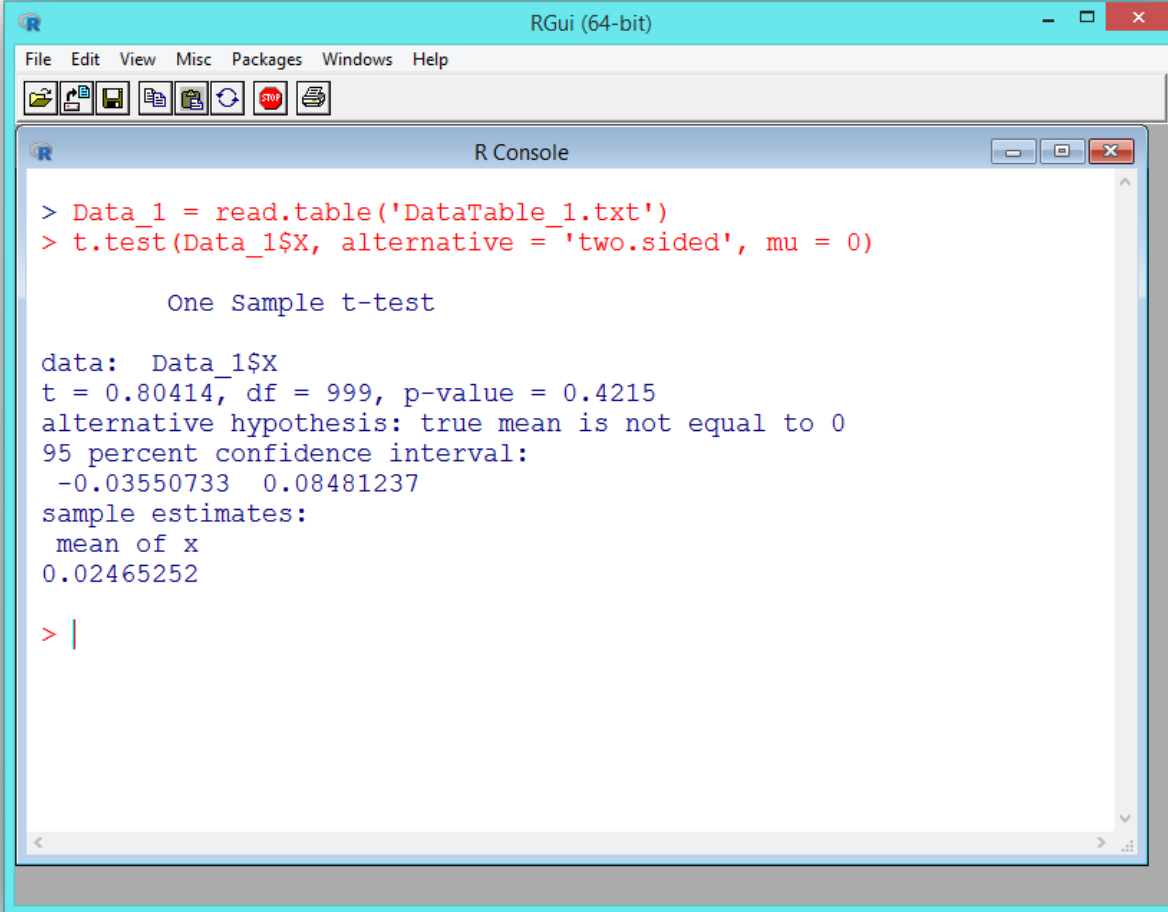
```
> Data_1 = read.table('DataTable_1.txt')
> t.test(Data_1$X, Data_1$Y, alternative = 'two.sided')

Welch Two Sample t-test

data: Data_1$X and Data_1$Y
t = 0.16372, df = 1996.9, p-value = 0.87
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.07885184  0.09321643
sample estimates:
 mean of x  mean of y
0.02465252 0.01747023

> |
```

Read and Write Data - T-test Example



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

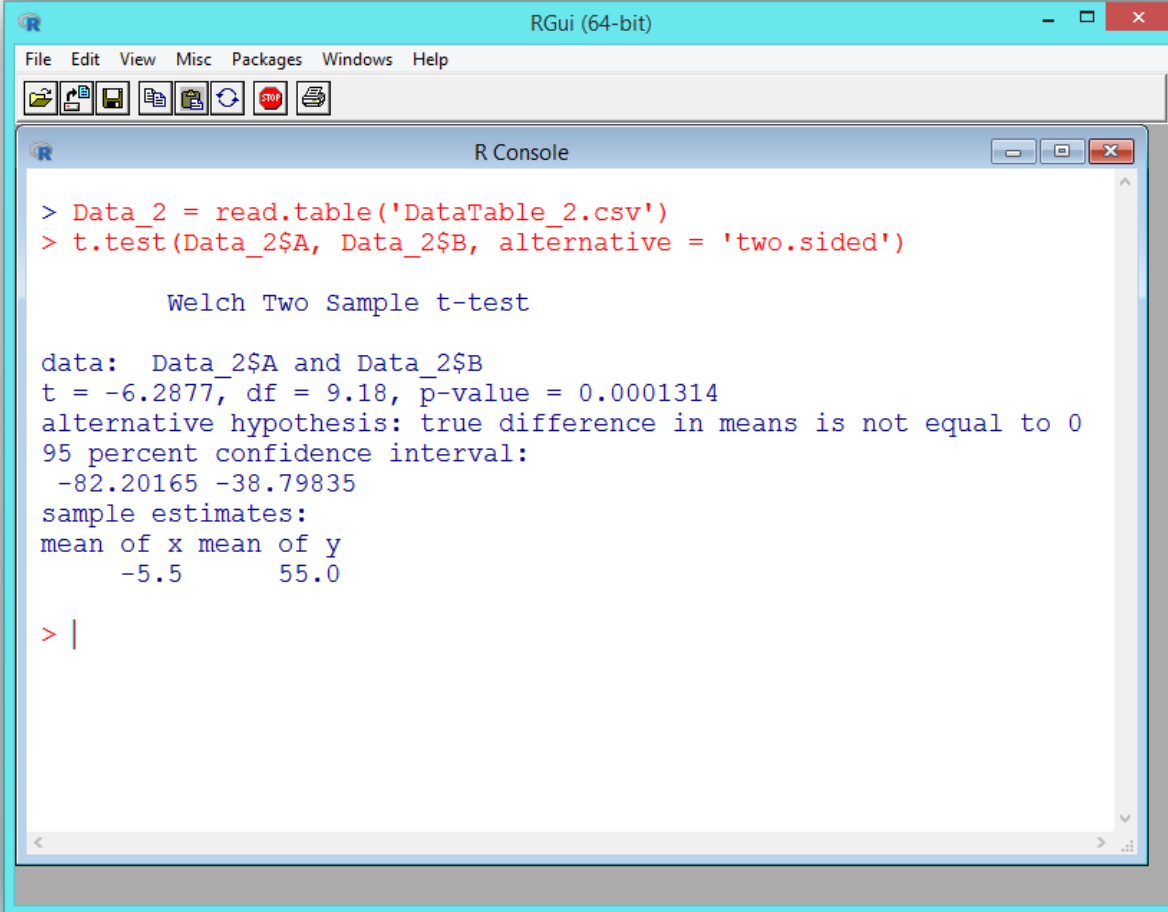
R Console
> Data_1 = read.table('DataTable_1.txt')
> t.test(Data_1$X, alternative = 'two.sided', mu = 0)

One Sample t-test

data: Data_1$X
t = 0.80414, df = 999, p-value = 0.4215
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.03550733  0.08481237
sample estimates:
mean of x
0.02465252

> |
```

Read and Write Data - T-test Example



The screenshot shows the RGui (64-bit) window with the R Console open. The console displays the following output for a Welch Two Sample t-test:

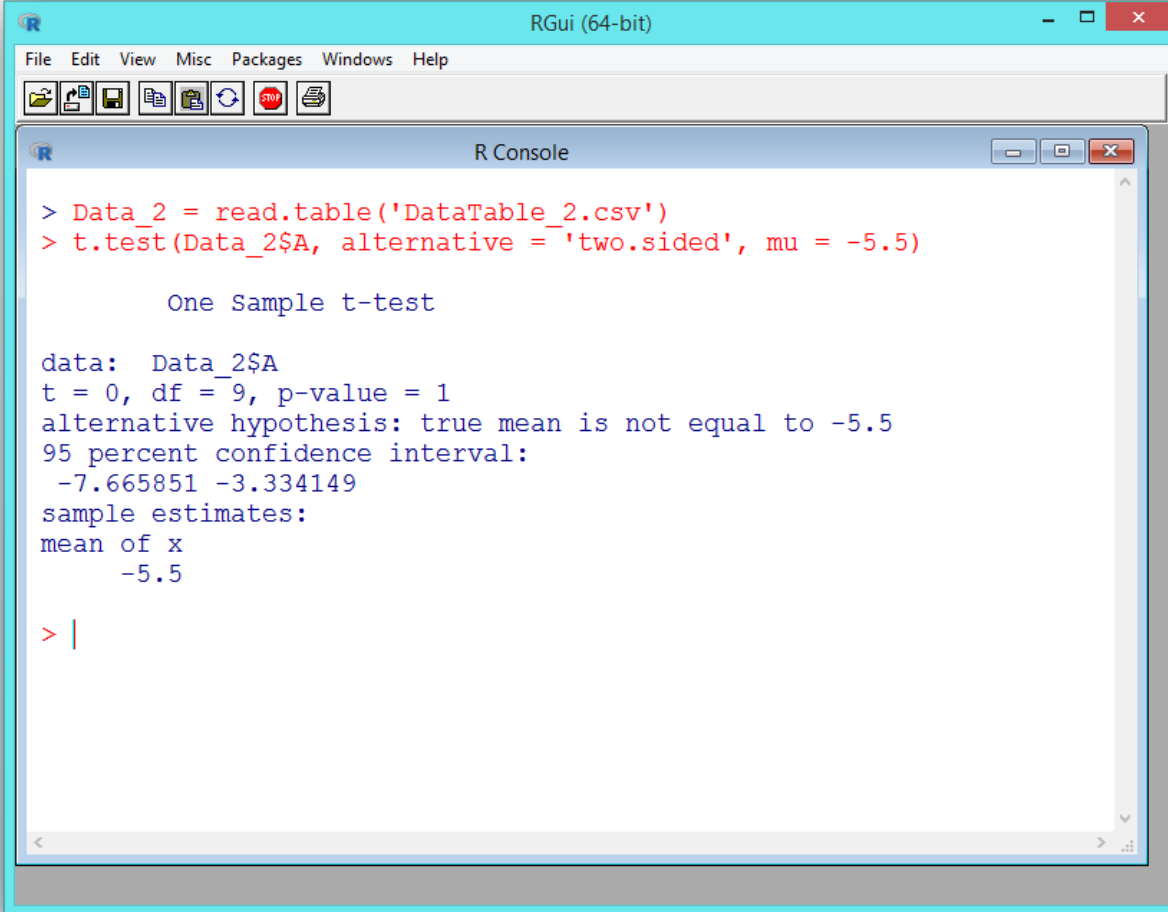
```
> Data_2 = read.table('DataTable_2.csv')
> t.test(Data_2$A, Data_2$B, alternative = 'two.sided')

Welch Two Sample t-test

data: Data_2$A and Data_2$B
t = -6.2877, df = 9.18, p-value = 0.0001314
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -82.20165 -38.79835
sample estimates:
mean of x mean of y
  -5.5      55.0

> |
```

Read and Write Data - T-test Example



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

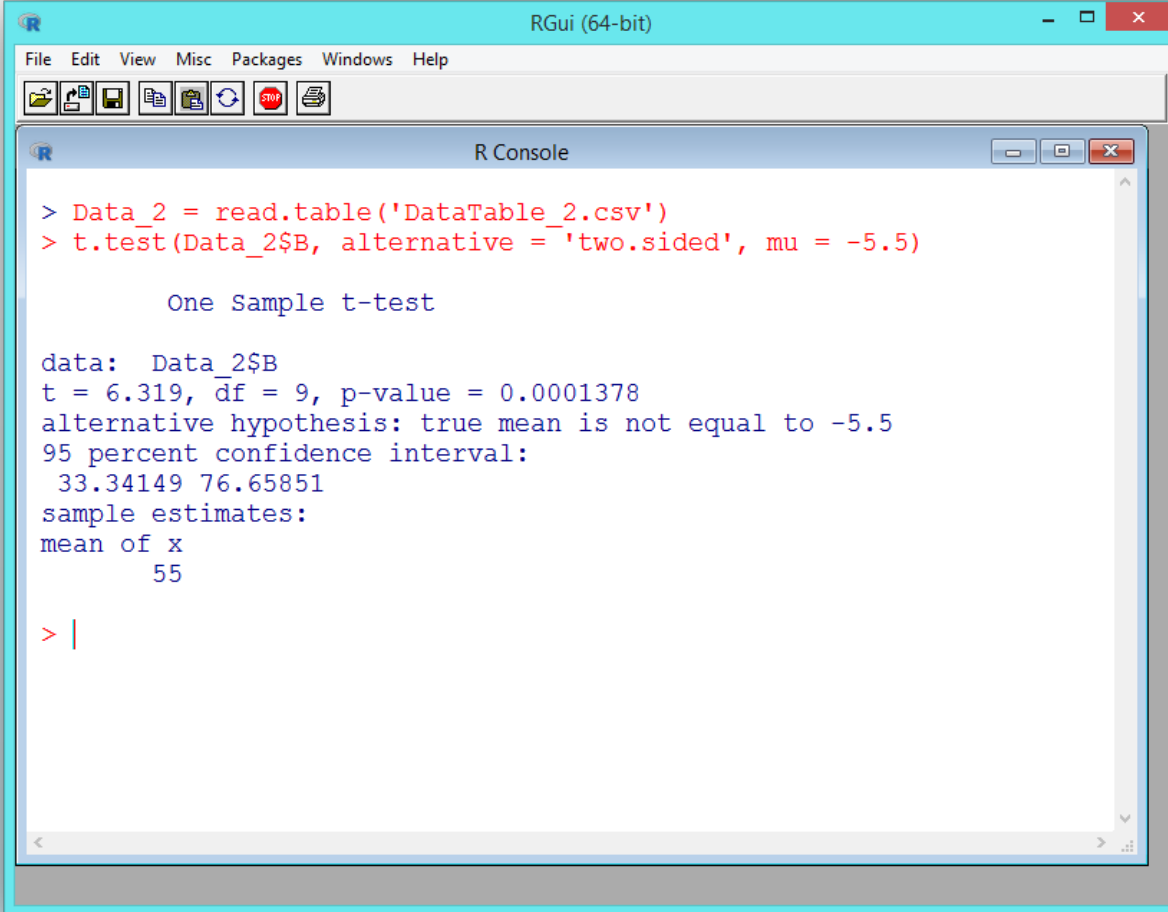
R Console
> Data_2 = read.table('DataTable_2.csv')
> t.test(Data_2$A, alternative = 'two.sided', mu = -5.5)

      One Sample t-test

data:  Data_2$A
t = 0, df = 9, p-value = 1
alternative hypothesis: true mean is not equal to -5.5
95 percent confidence interval:
 -7.665851 -3.334149
sample estimates:
mean of x
 -5.5

> |
```

Read and Write Data - T-test Example



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> Data_2 = read.table('DataTable_2.csv')
> t.test(Data_2$B, alternative = 'two.sided', mu = -5.5)

      One Sample t-test

data:  Data_2$B
t = 6.319, df = 9, p-value = 0.0001378
alternative hypothesis: true mean is not equal to -5.5
95 percent confidence interval:
 33.34149 76.65851
sample estimates:
mean of x
      55

> |
```

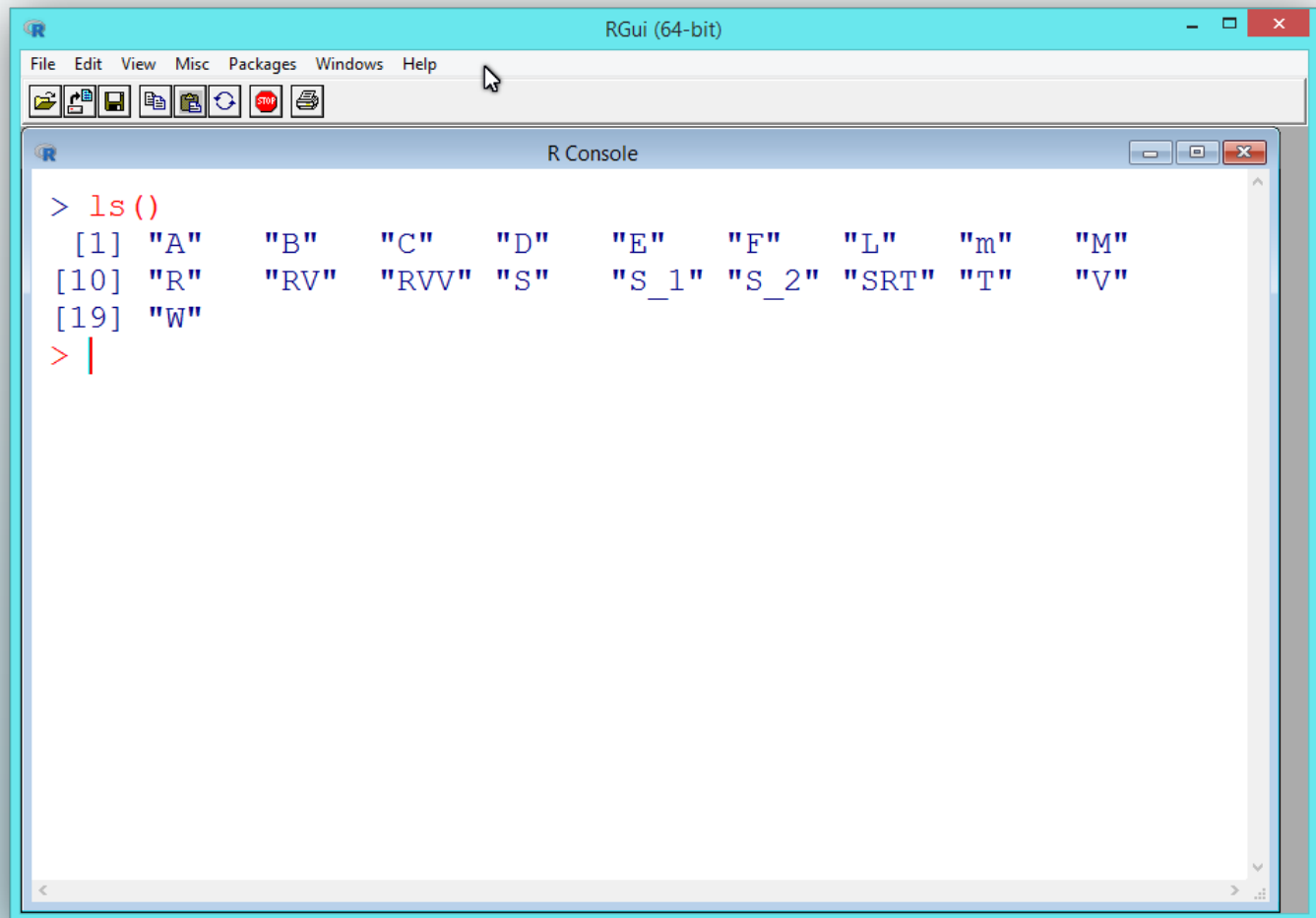


Workspace

Workspace and Objects

Workspace

► List of the Saved Objects

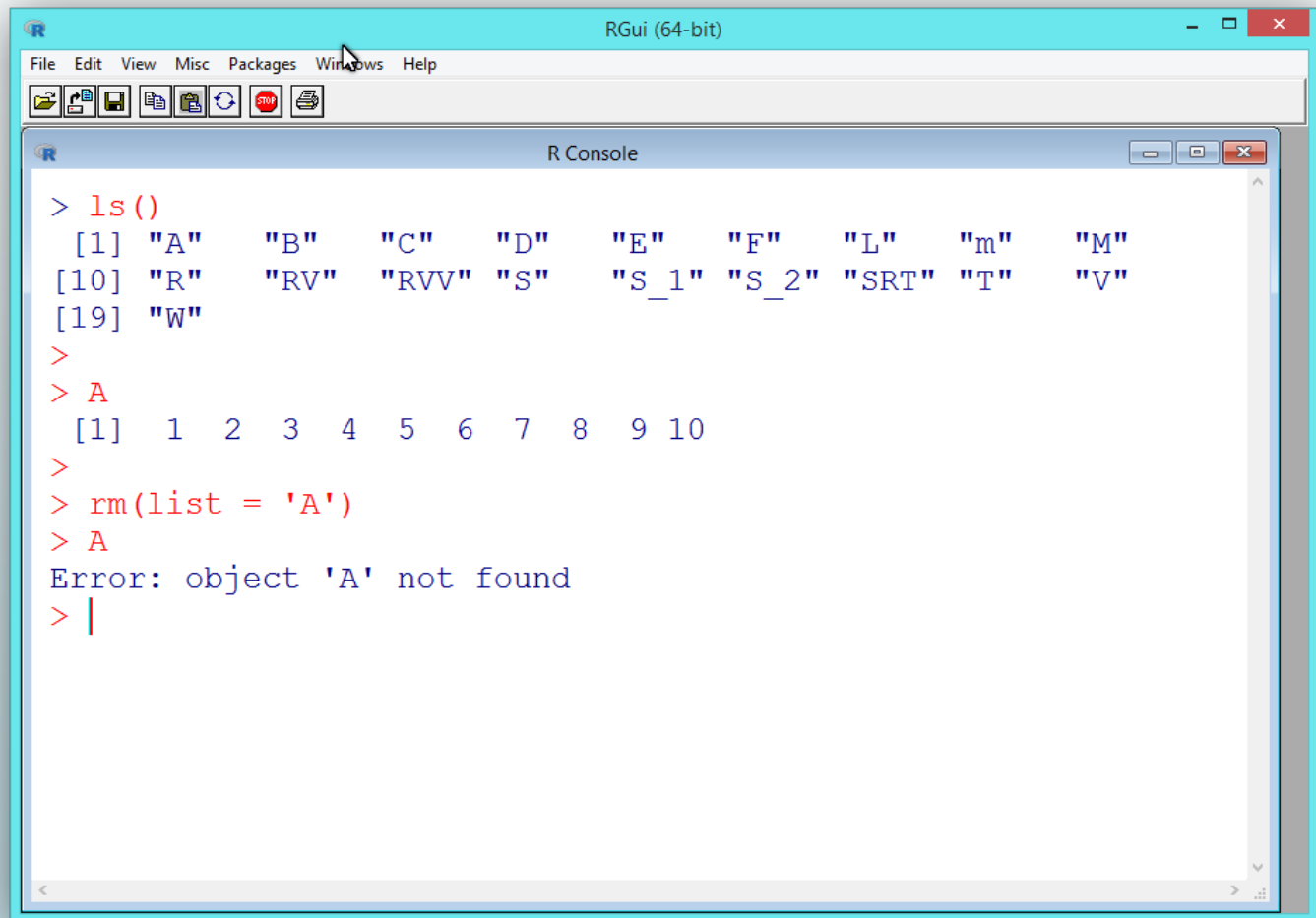


The screenshot shows the RGui (64-bit) interface. The R Console window displays the following output for the `ls()` command:

```
> ls()  
[1] "A"      "B"      "C"      "D"      "E"      "F"      "L"      "m"      "M"  
[10] "R"      "RV"     "RVV"    "S"      "S_1"    "S_2"    "SRT"    "T"      "V"  
[19] "W"  
> |
```


Workspace

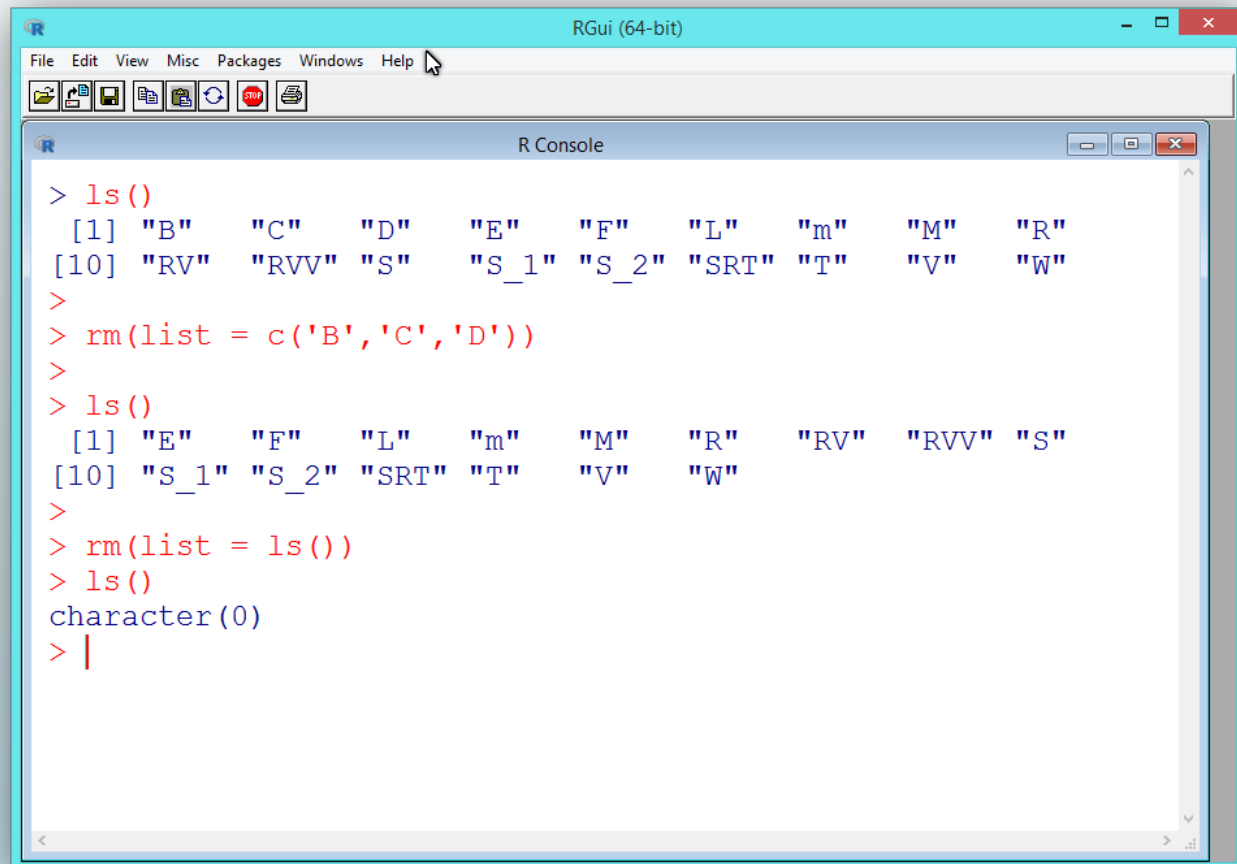
➤ Removing an Object from the Workspace



```
> ls()
 [1] "A"  "B"  "C"  "D"  "E"  "F"  "L"  "m"  "M"
 [10] "R"  "RV" "RVV" "S"  "S_1" "S_2" "SRT" "T"  "V"
 [19] "W"
>
> A
 [1] 1 2 3 4 5 6 7 8 9 10
>
> rm(list = 'A')
> A
Error: object 'A' not found
> |
```

Workspace

- Removing a List of the Objects from the Workspace



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> ls()
[1] "B" "C" "D" "E" "F" "L" "m" "M" "R"
[10] "RV" "RVV" "S" "S_1" "S_2" "SRT" "T" "V" "W"
>
> rm(list = c('B', 'C', 'D'))
>
> ls()
[1] "E" "F" "L" "m" "M" "R" "RV" "RVV" "S"
[10] "S_1" "S_2" "SRT" "T" "V" "W"
>
> rm(list = ls())
> ls()
character(0)
> |
```

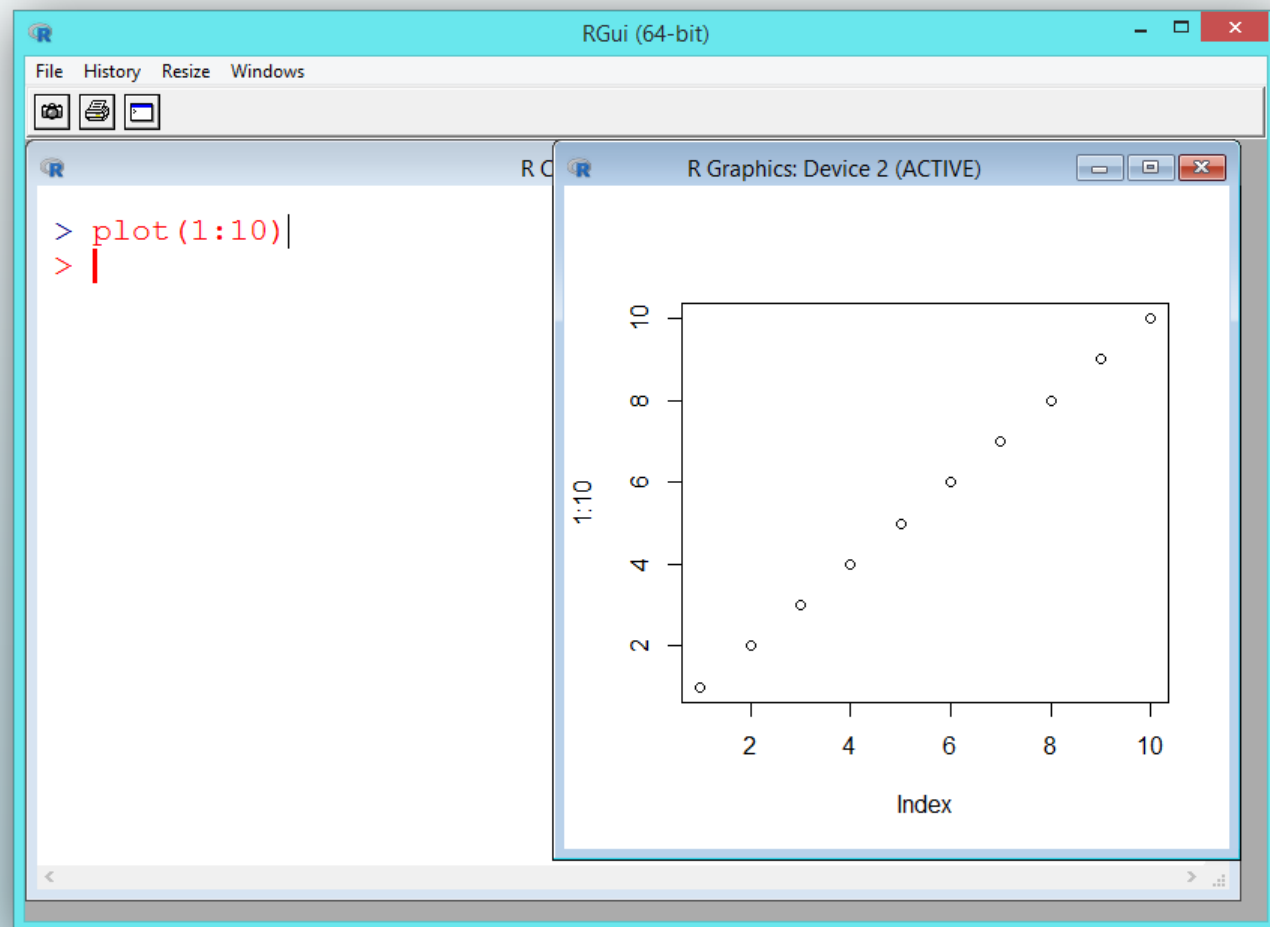


Plots

Two-Dimensional Plots

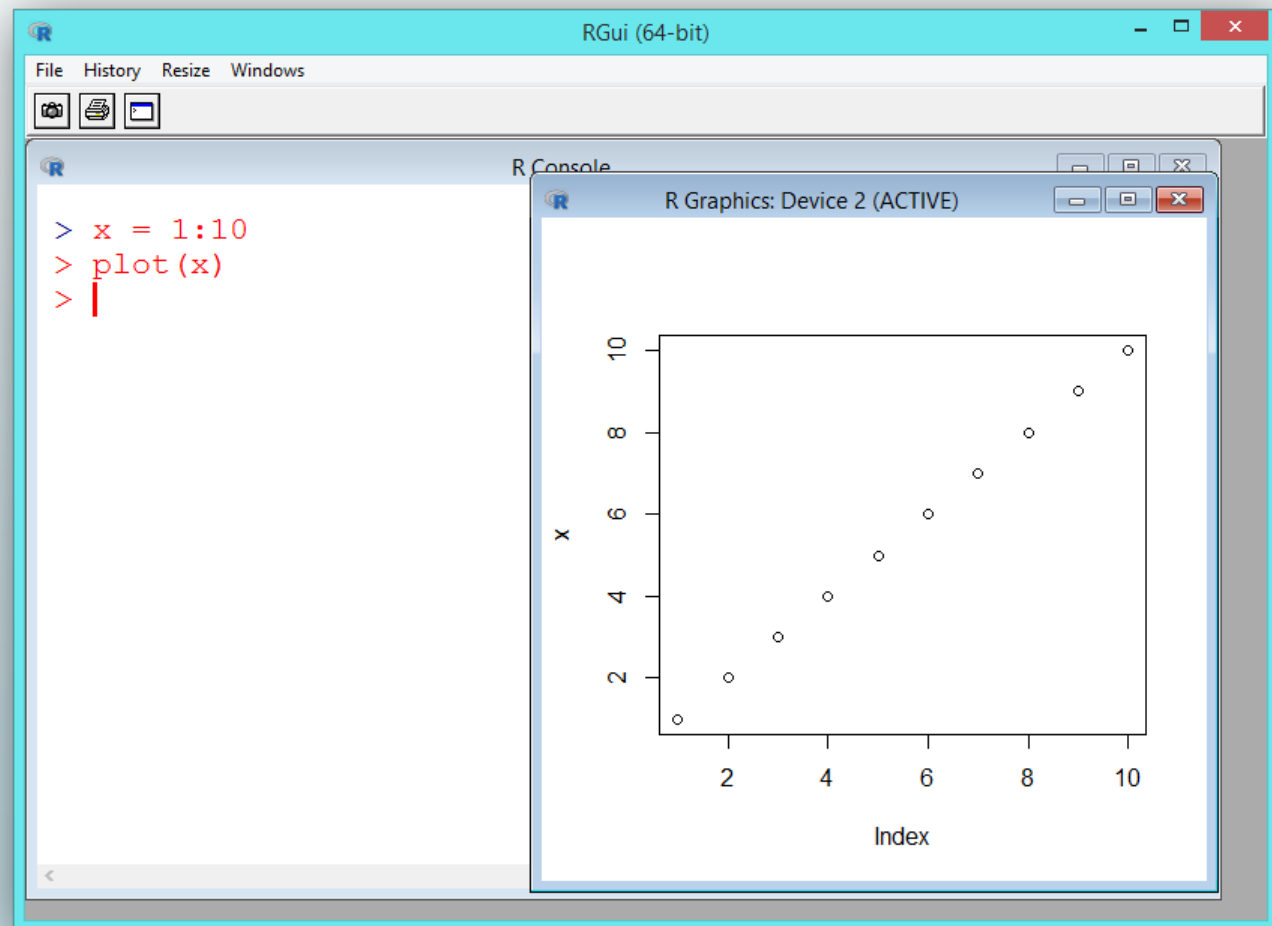
Plot

► plot Function (1)



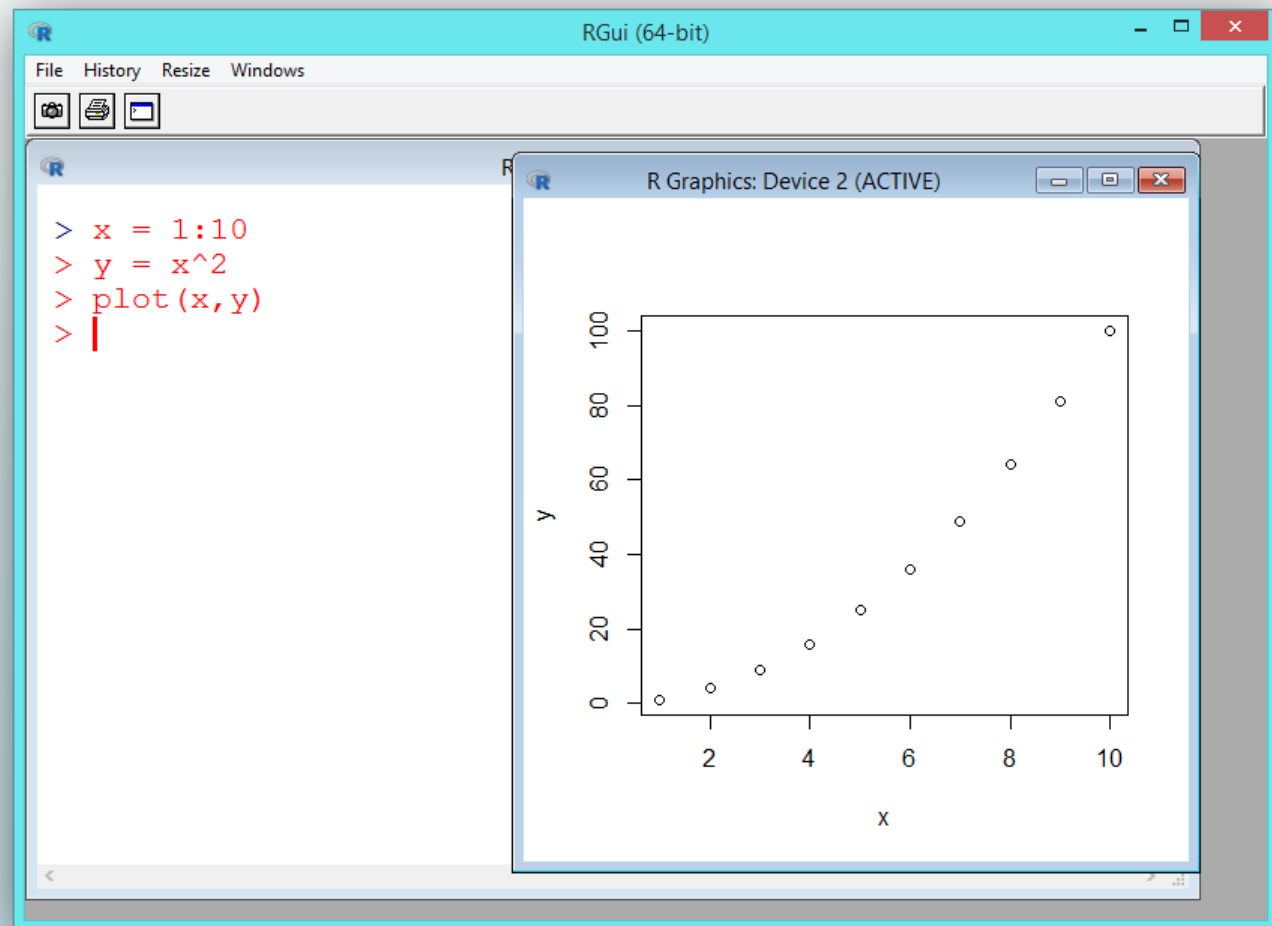
Plot

plot Function (2)



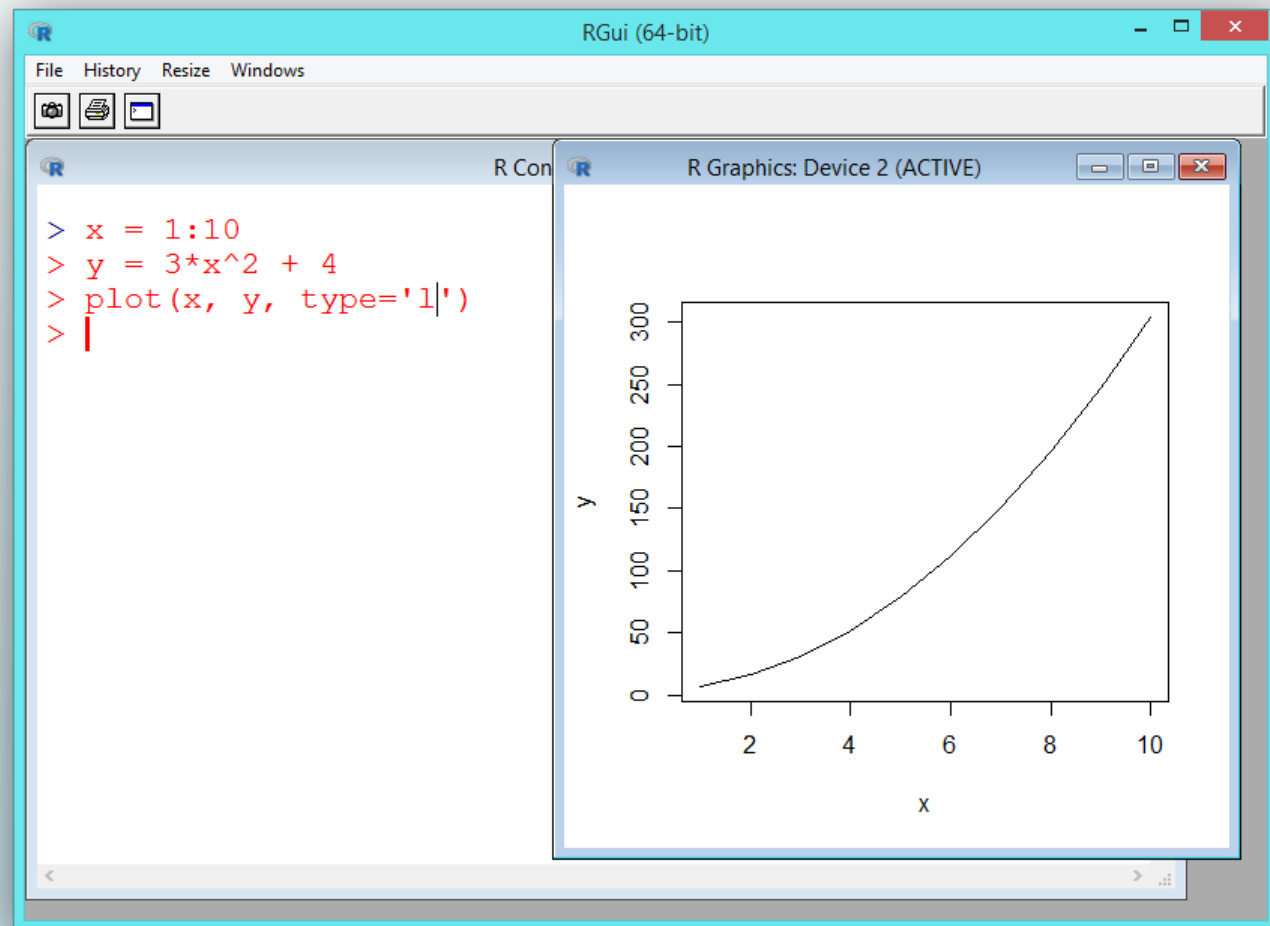
Plot

plot Function (3)



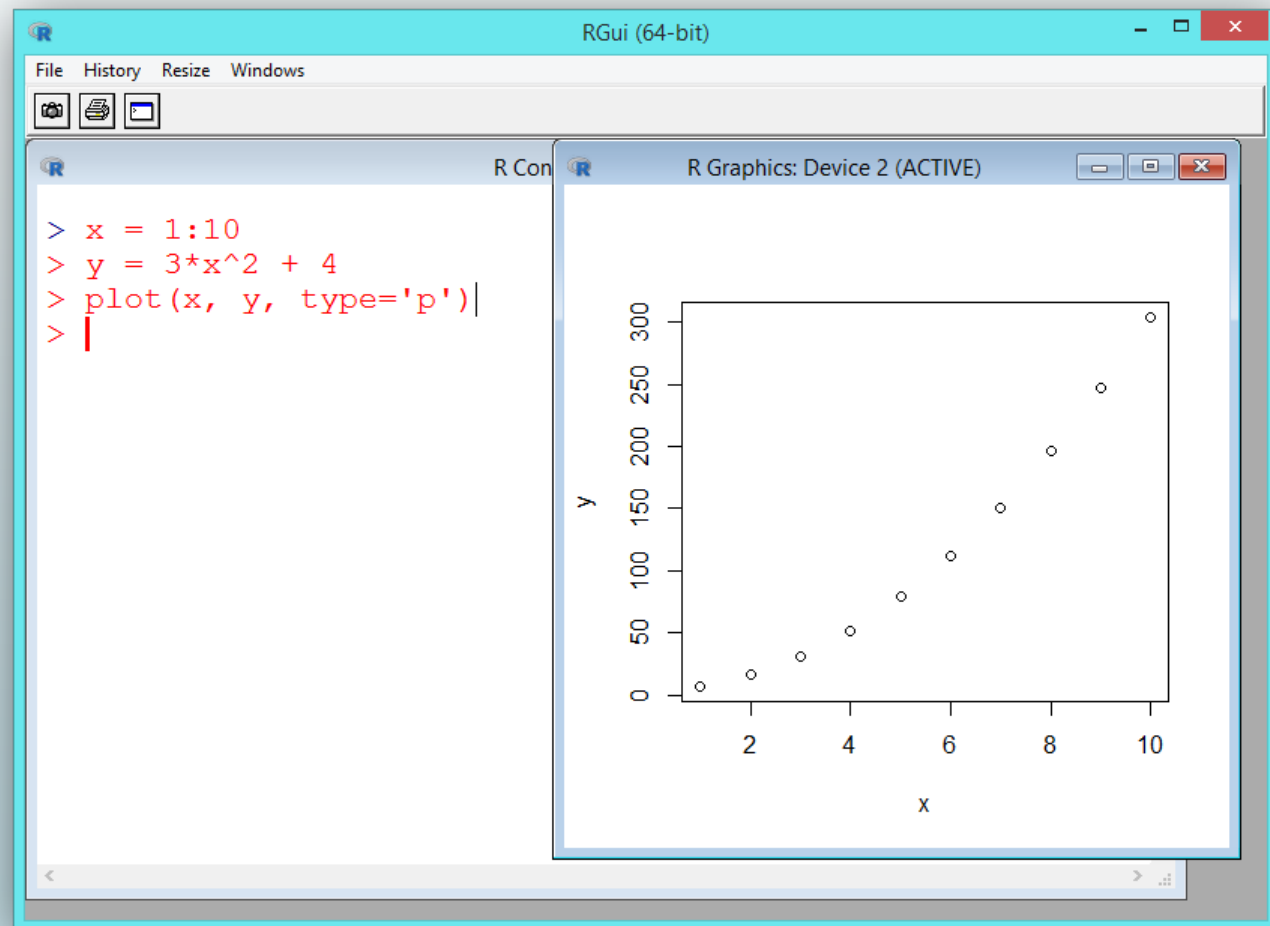
Plot

► `type='l'` (lines)



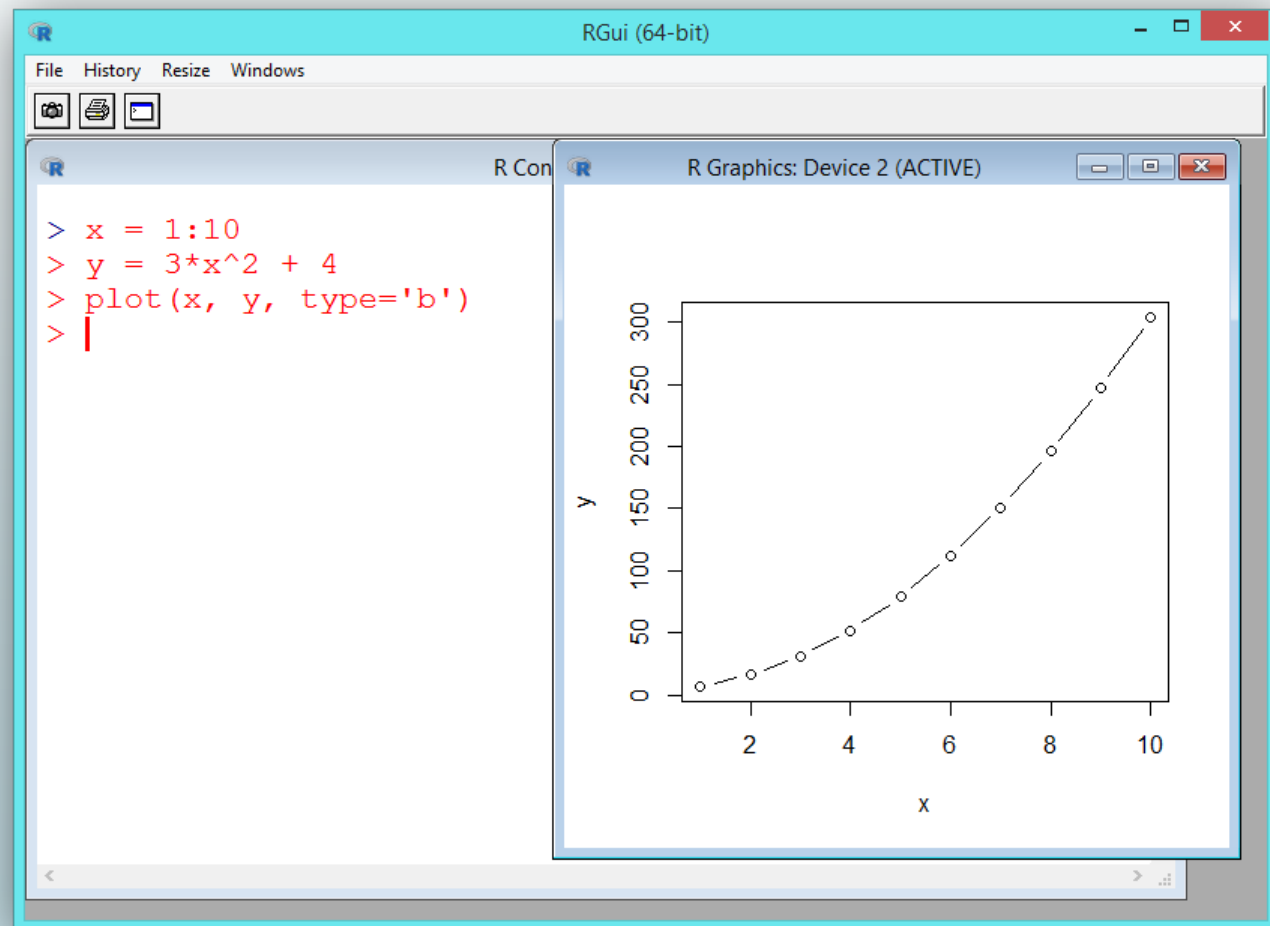
Plot

► `type='p'` (points)



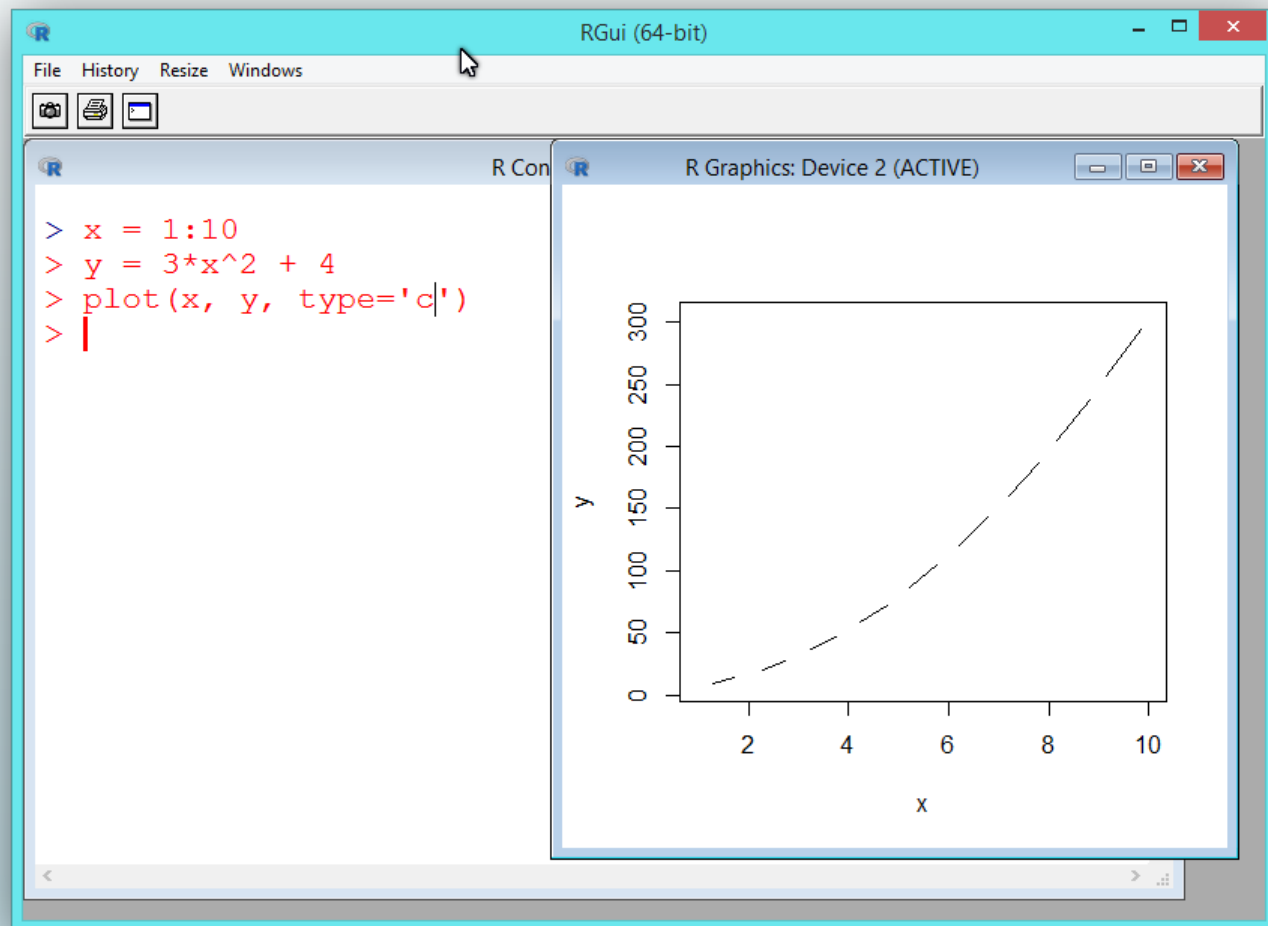
Plot

► `type='b'` (both)



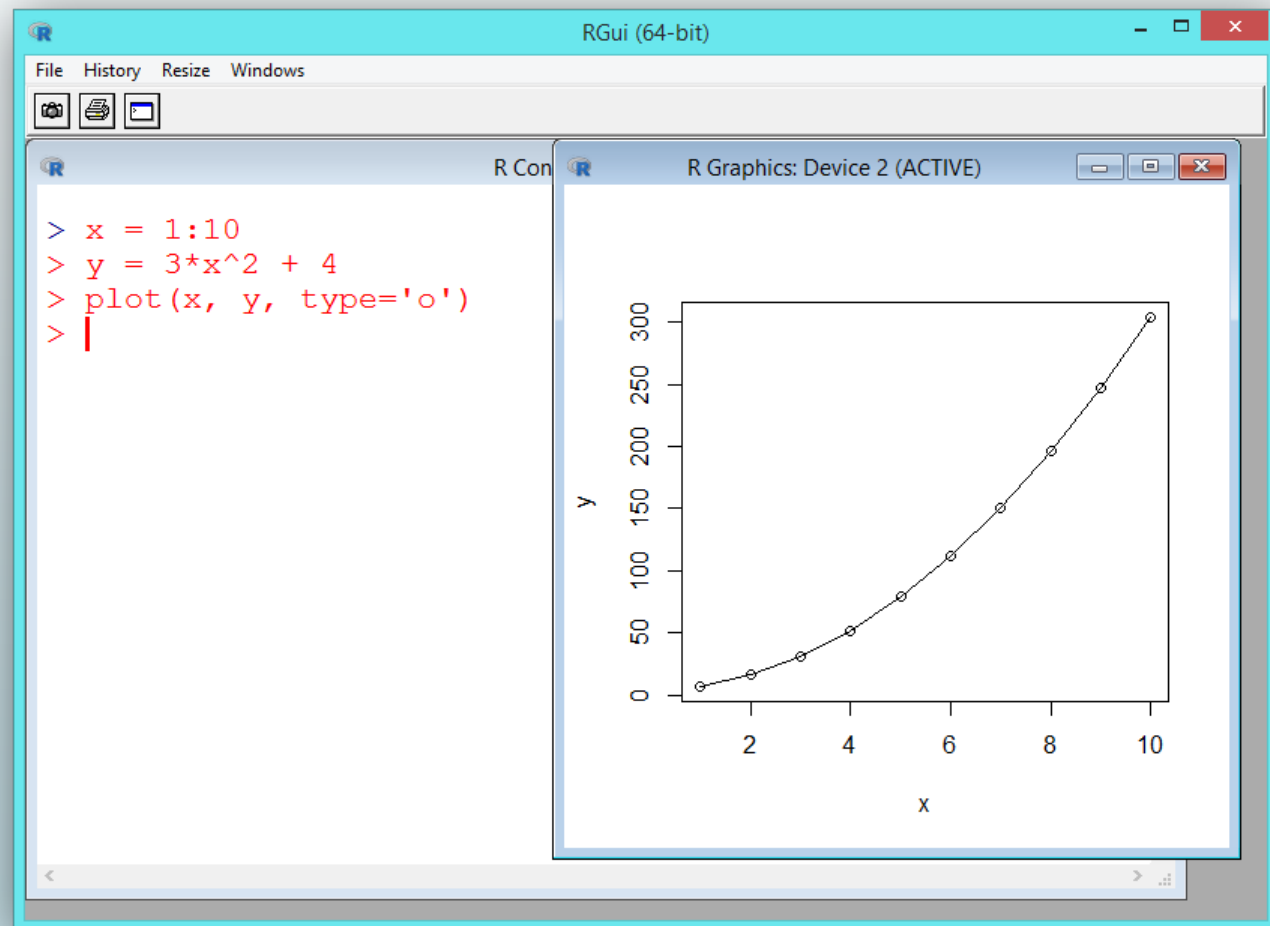
Plot

► `type='c'` (the lines part alone of 'b')



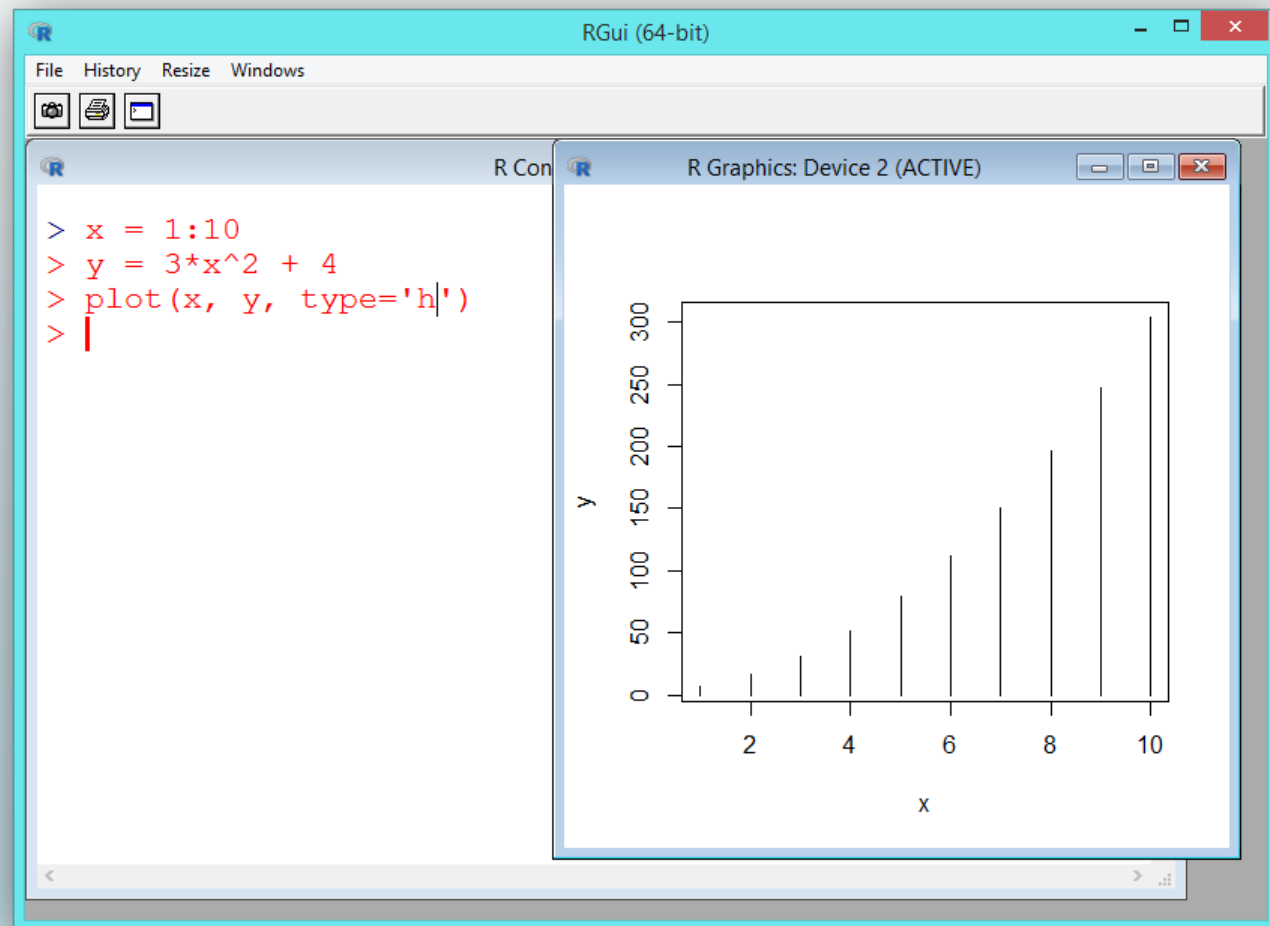
Plot

► `type='o'` (overplotted)



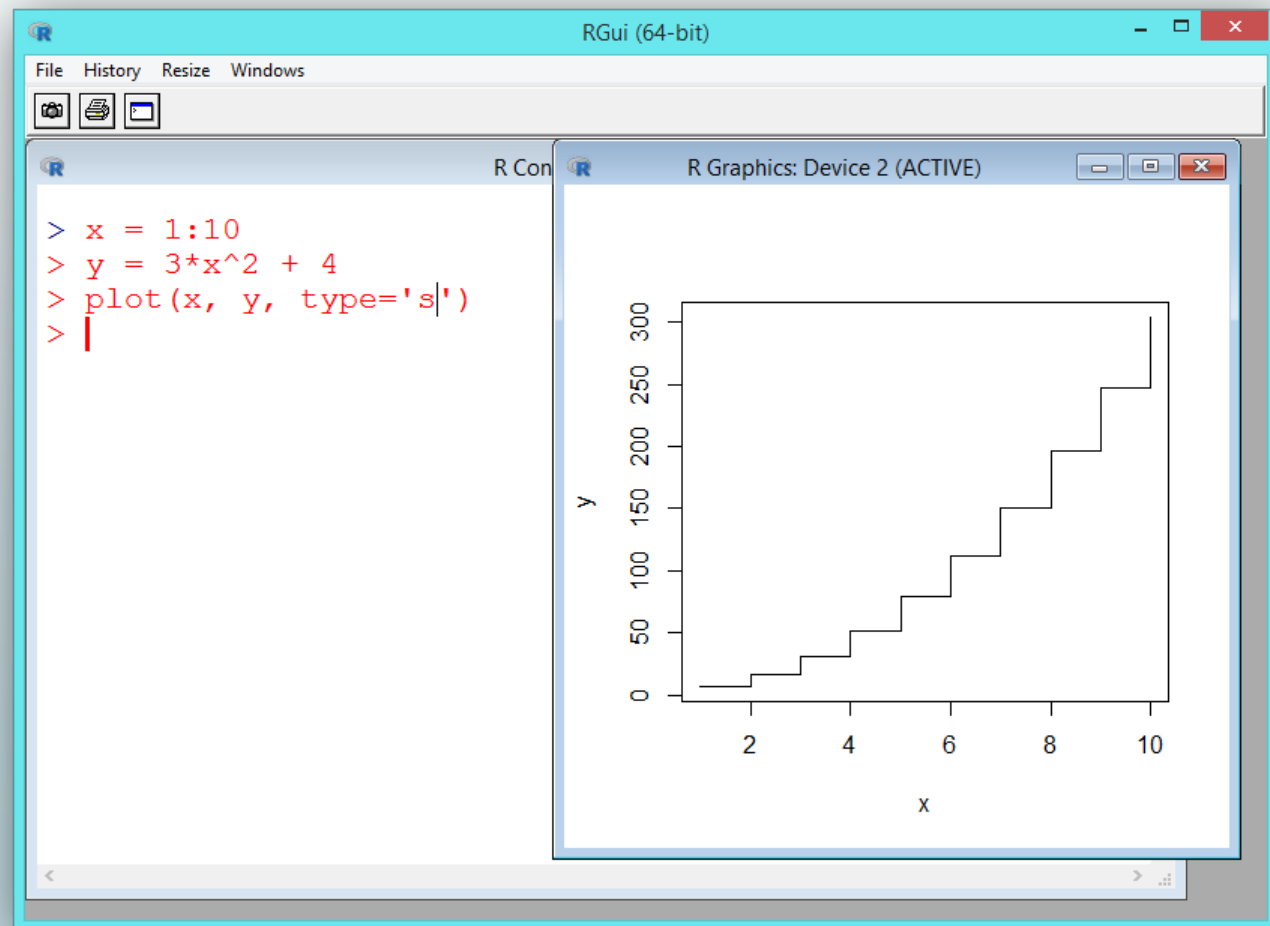
Plot

► `type='h'` ('histogram' like)



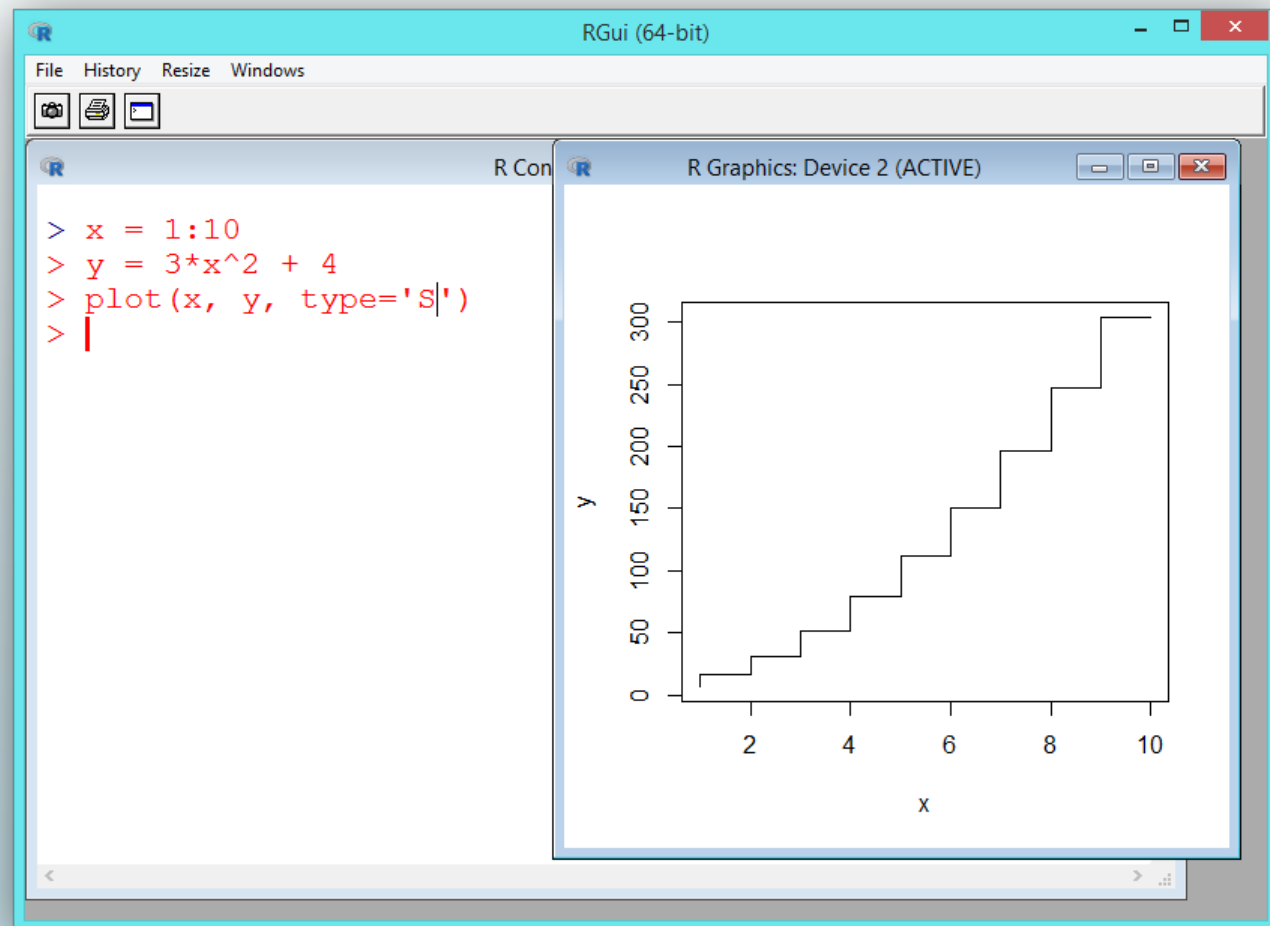
Plot

► `type='s'` (stair steps)



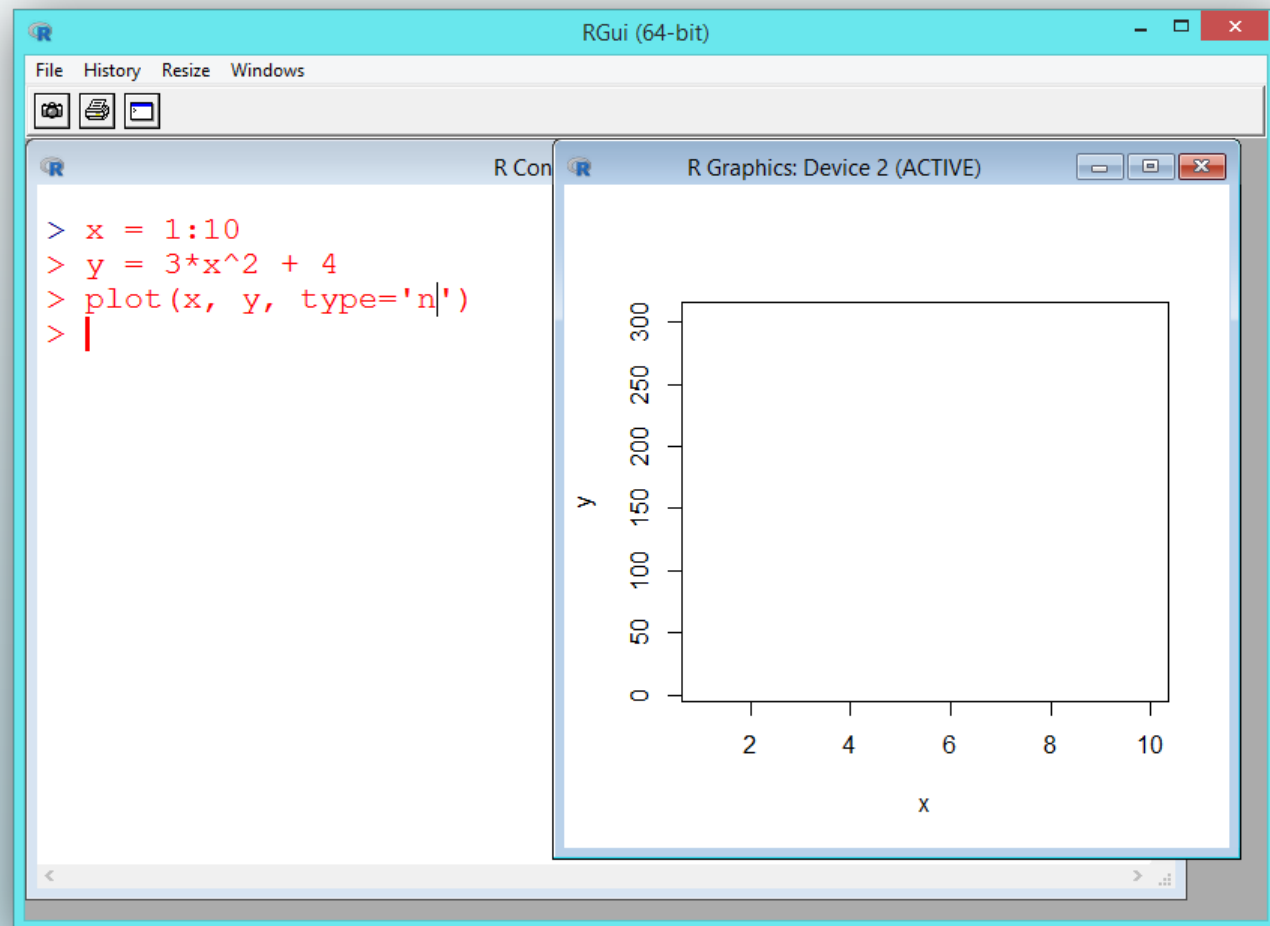
Plot

► `type='s'` (other steps)



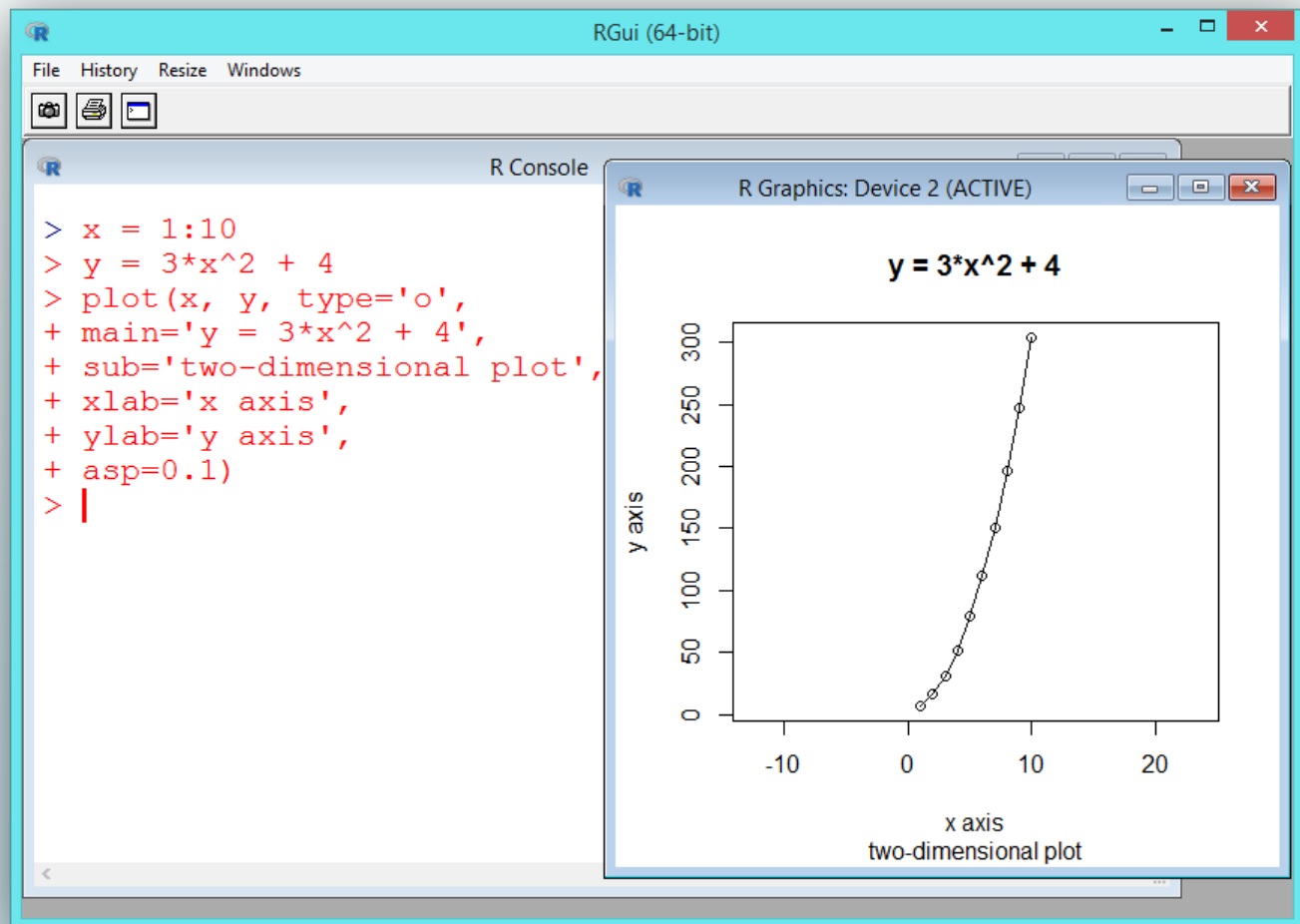
Plot

► `type='n'` (no plotting)



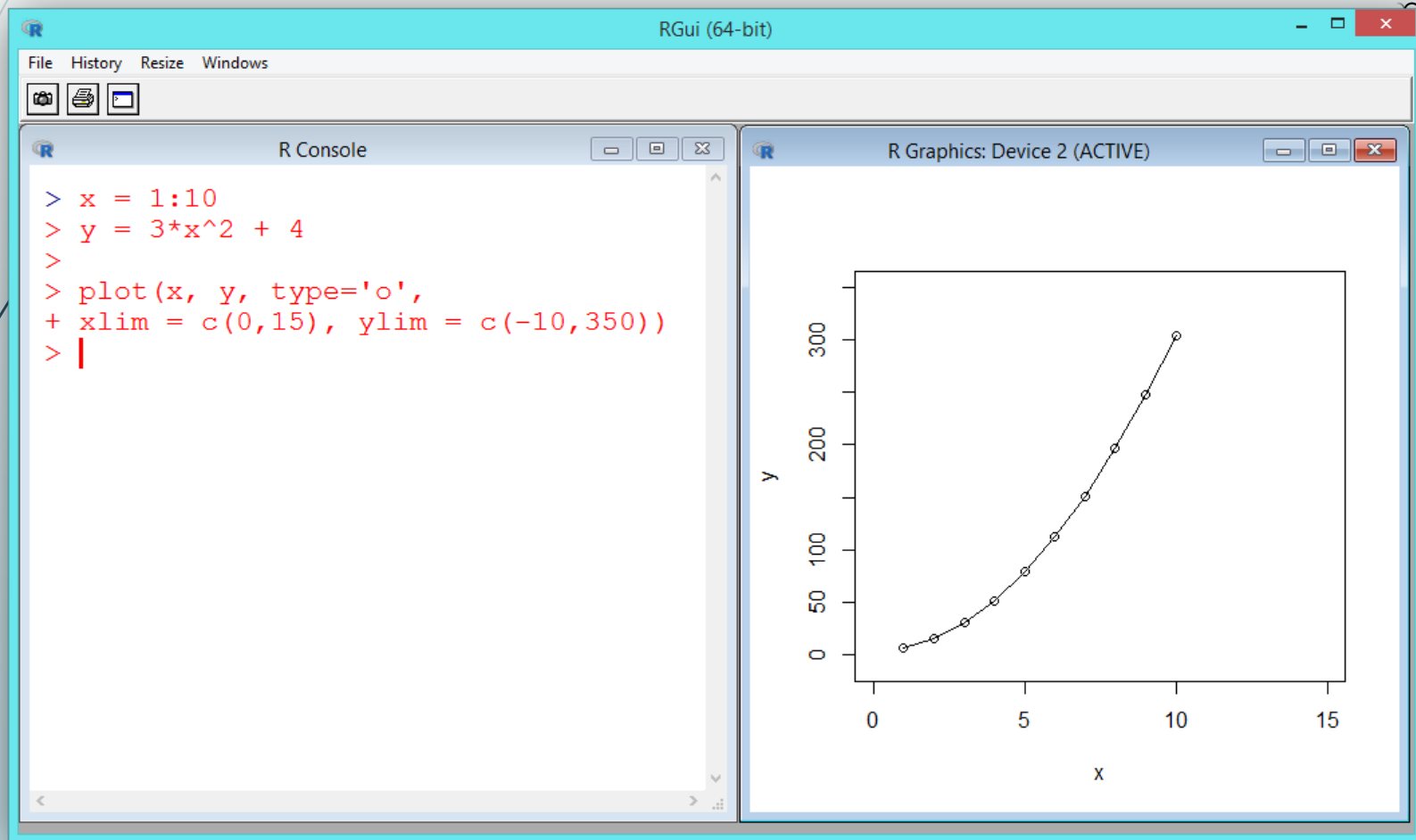
Plot

plot Arguments (1)



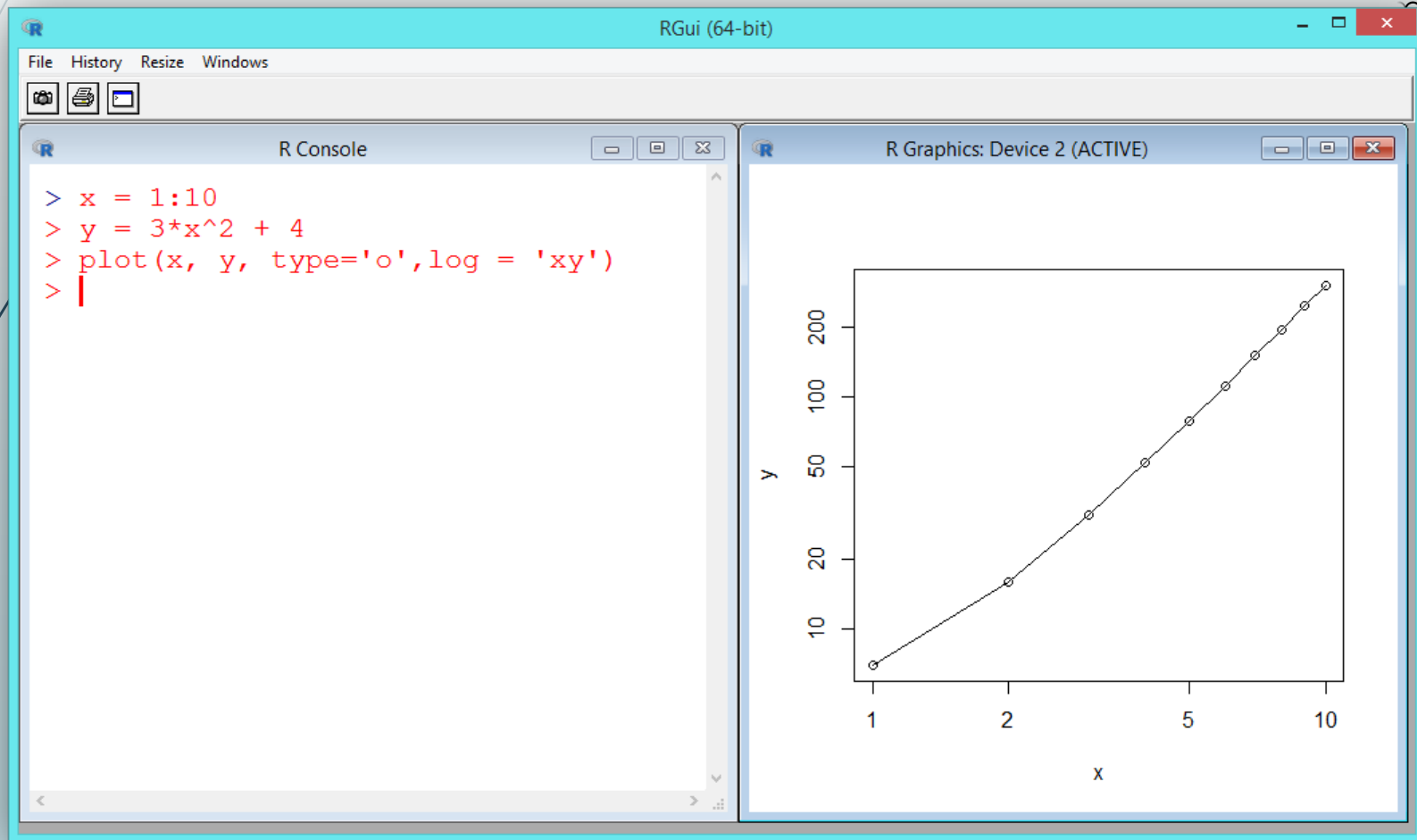
Plot

plot Arguments (2)



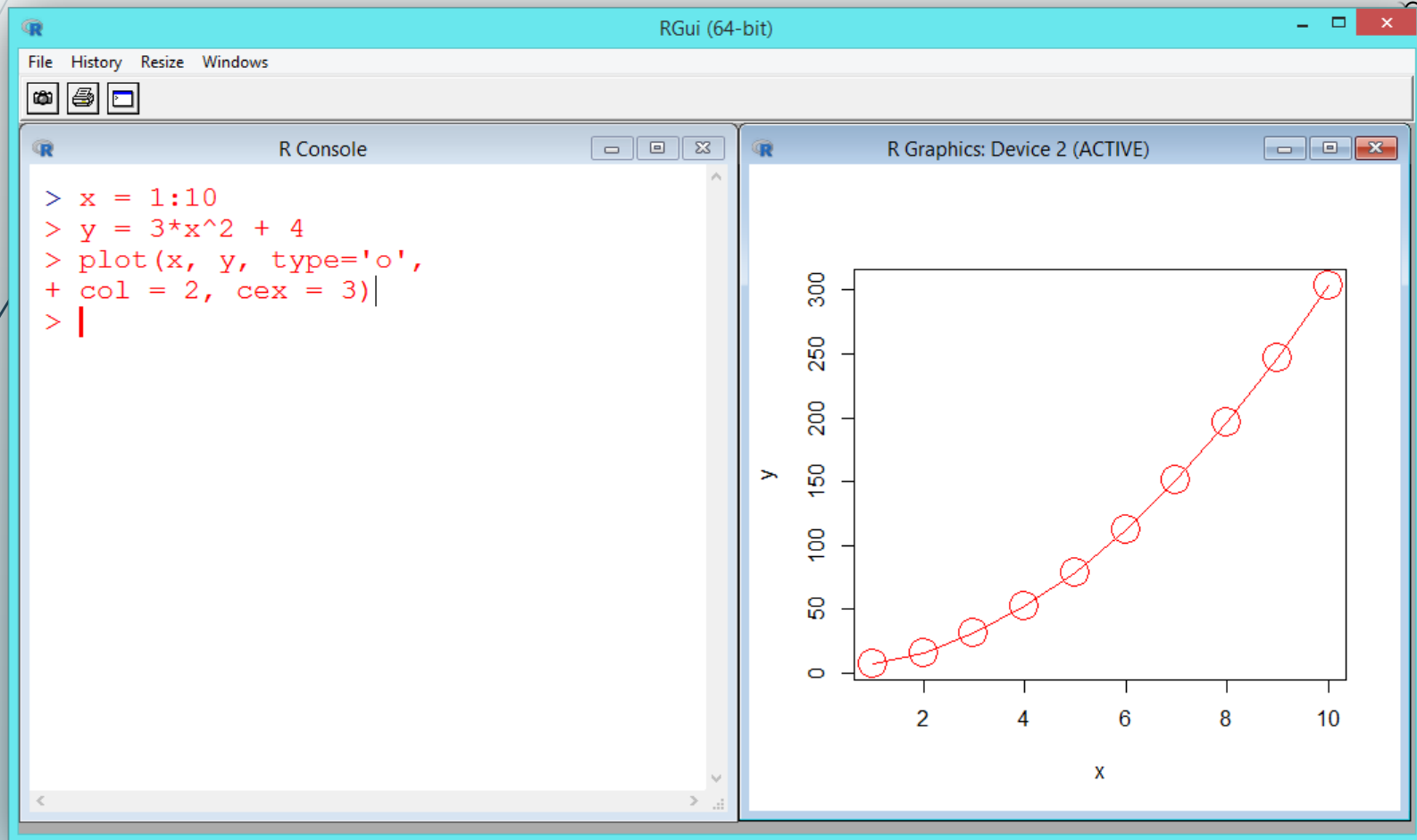
Plot

plot Arguments (3)



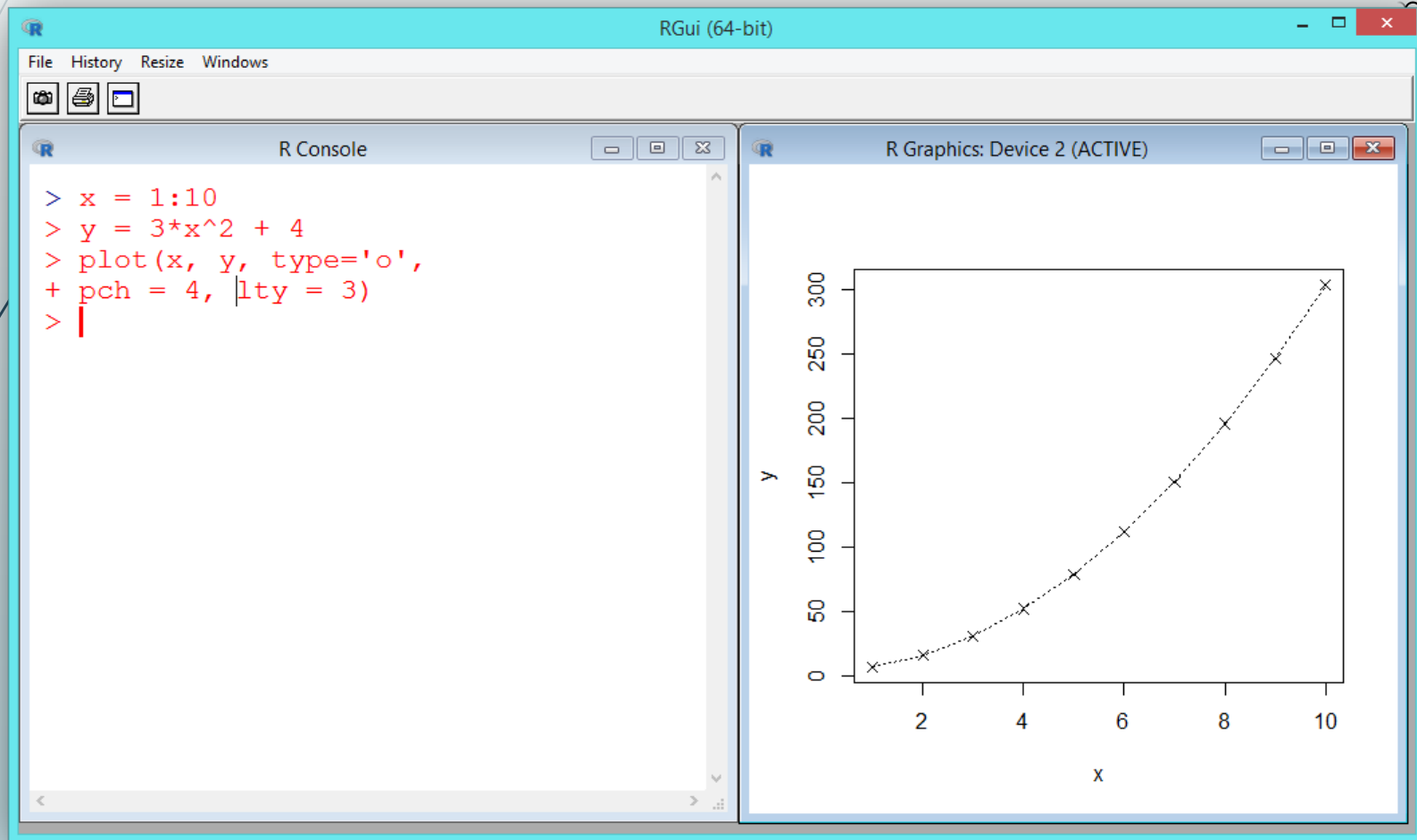
Plot

plot Arguments (4)



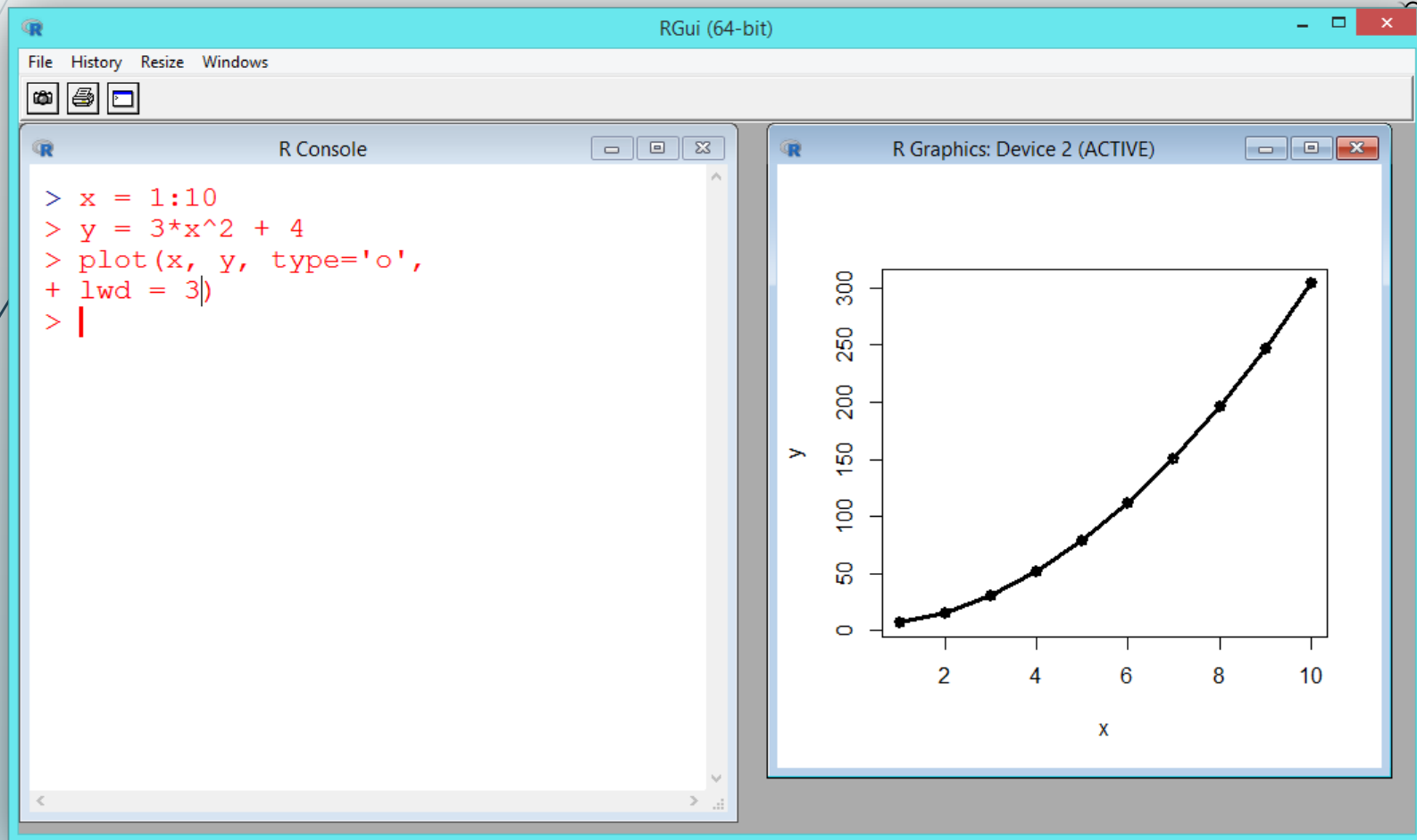
Plot

plot Arguments (5)



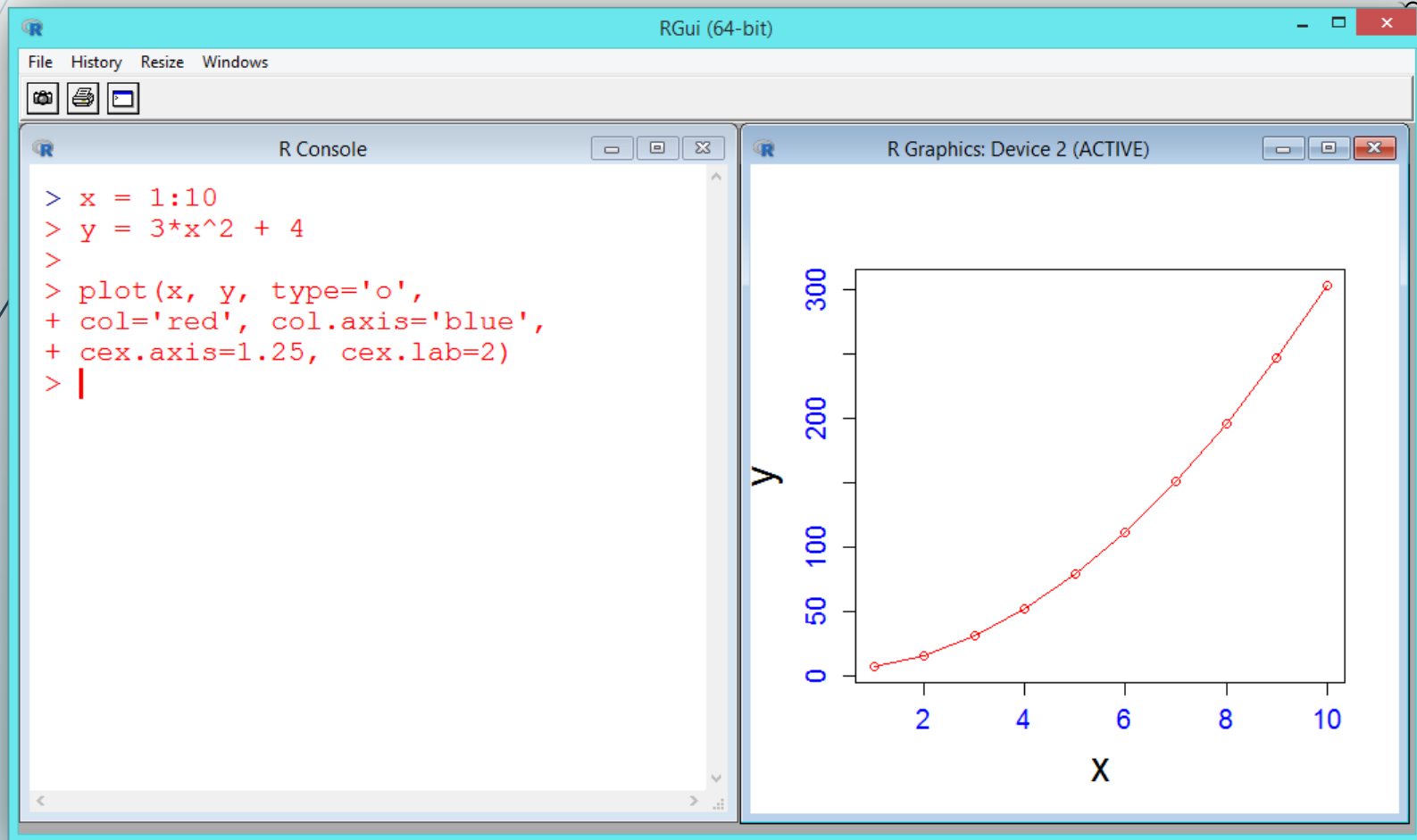
Plot

plot Arguments (6)



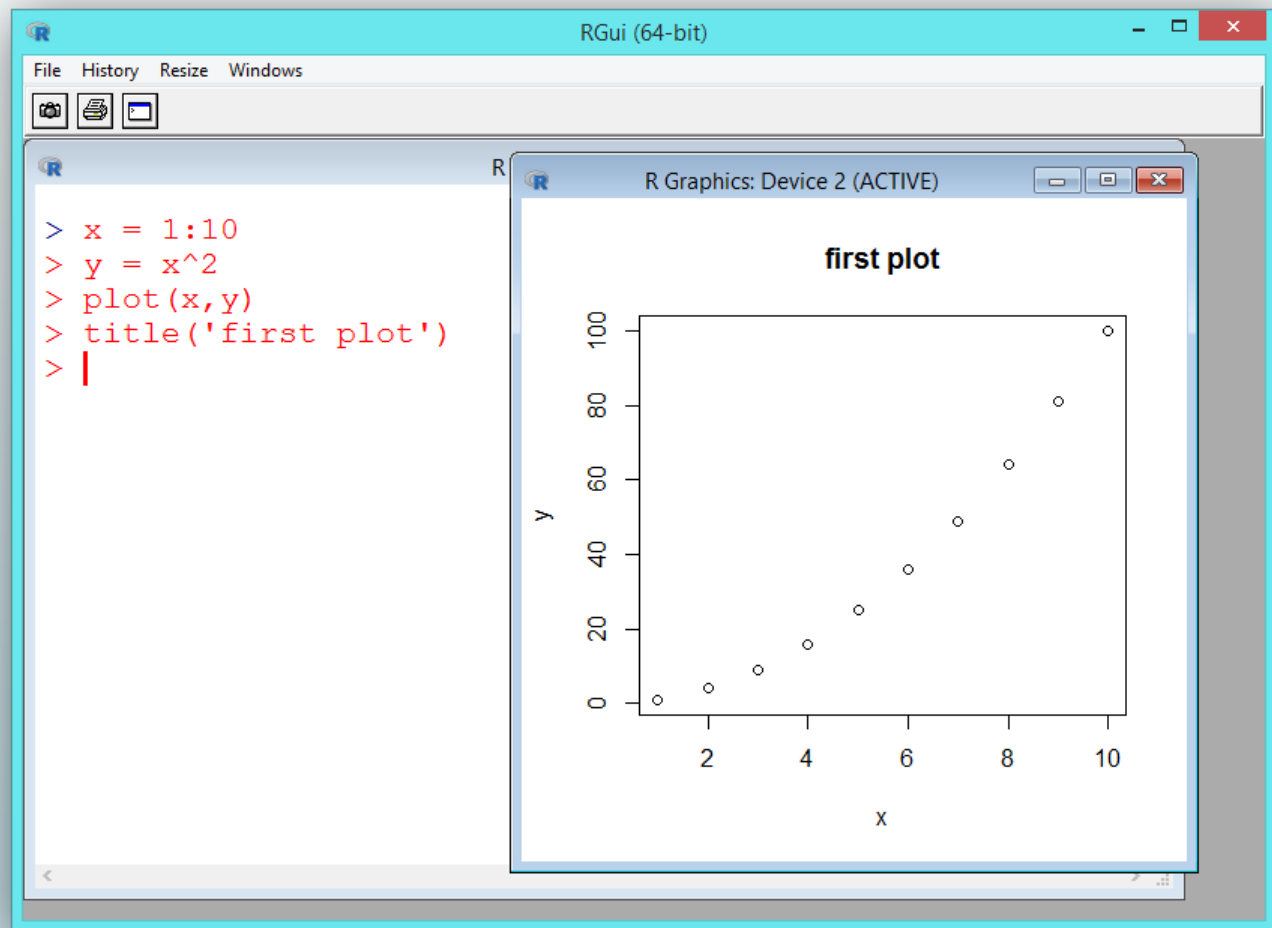
Plot

plot Arguments (7)



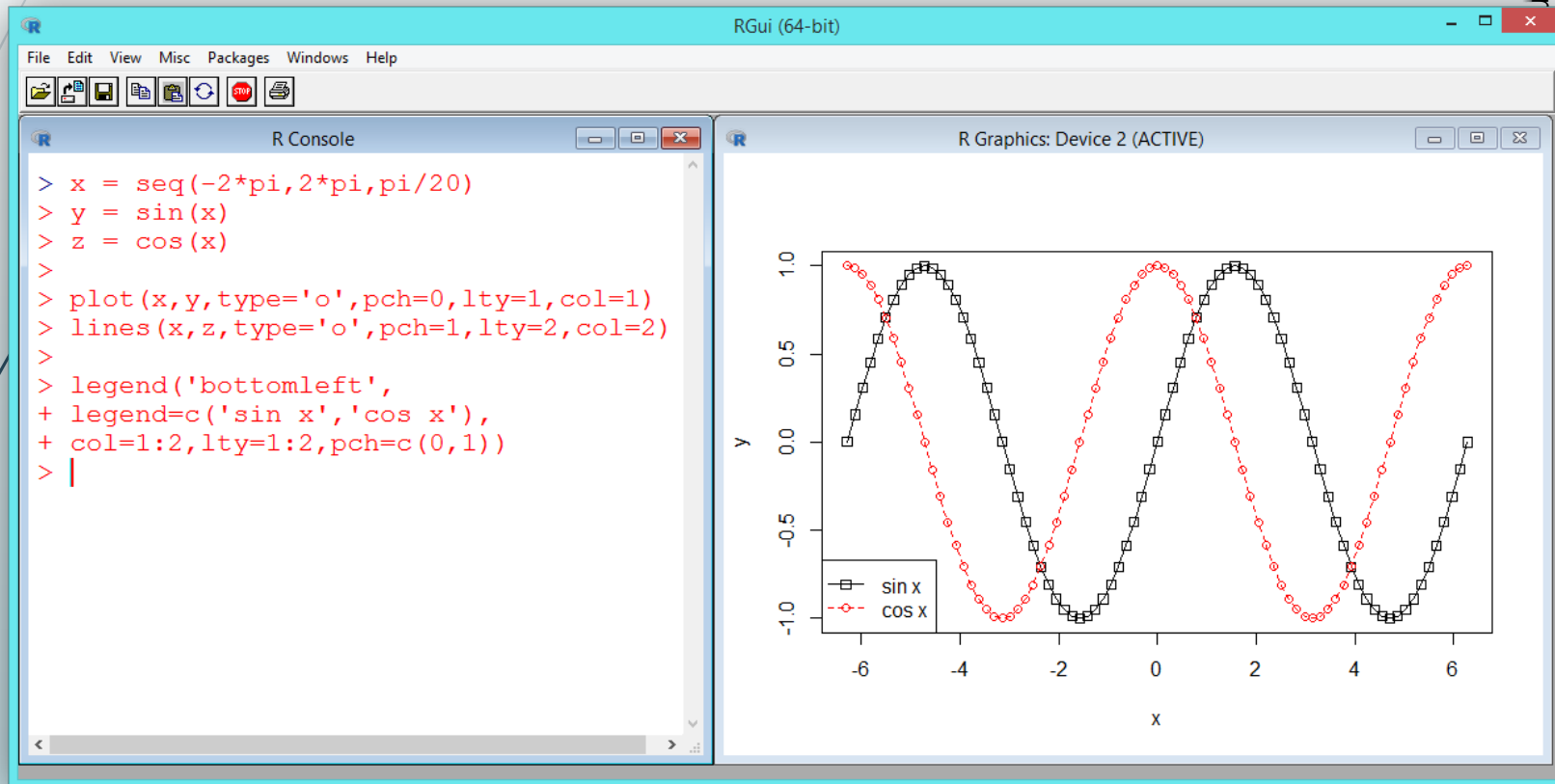
Plot

► title Function



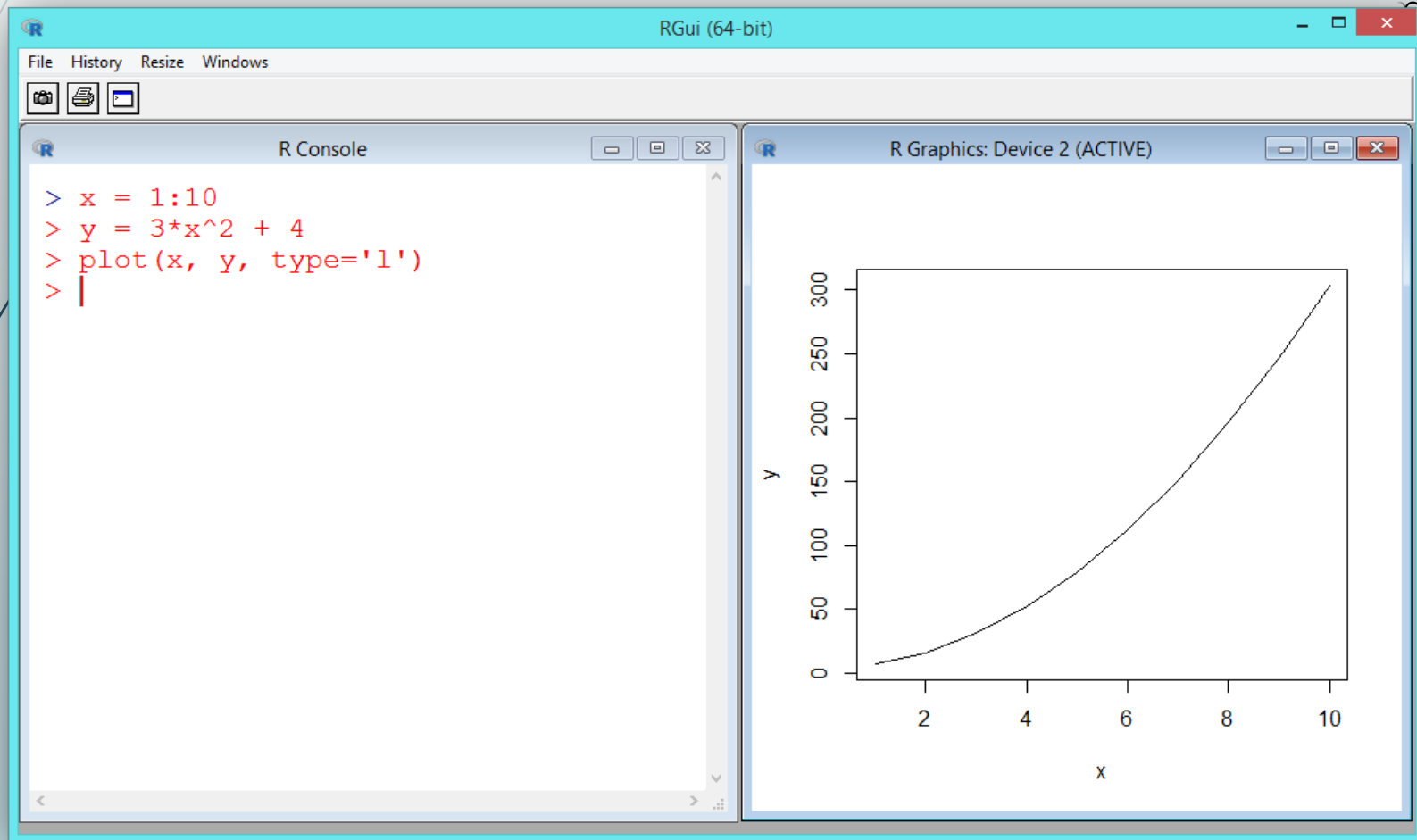
Plot

► legend Function



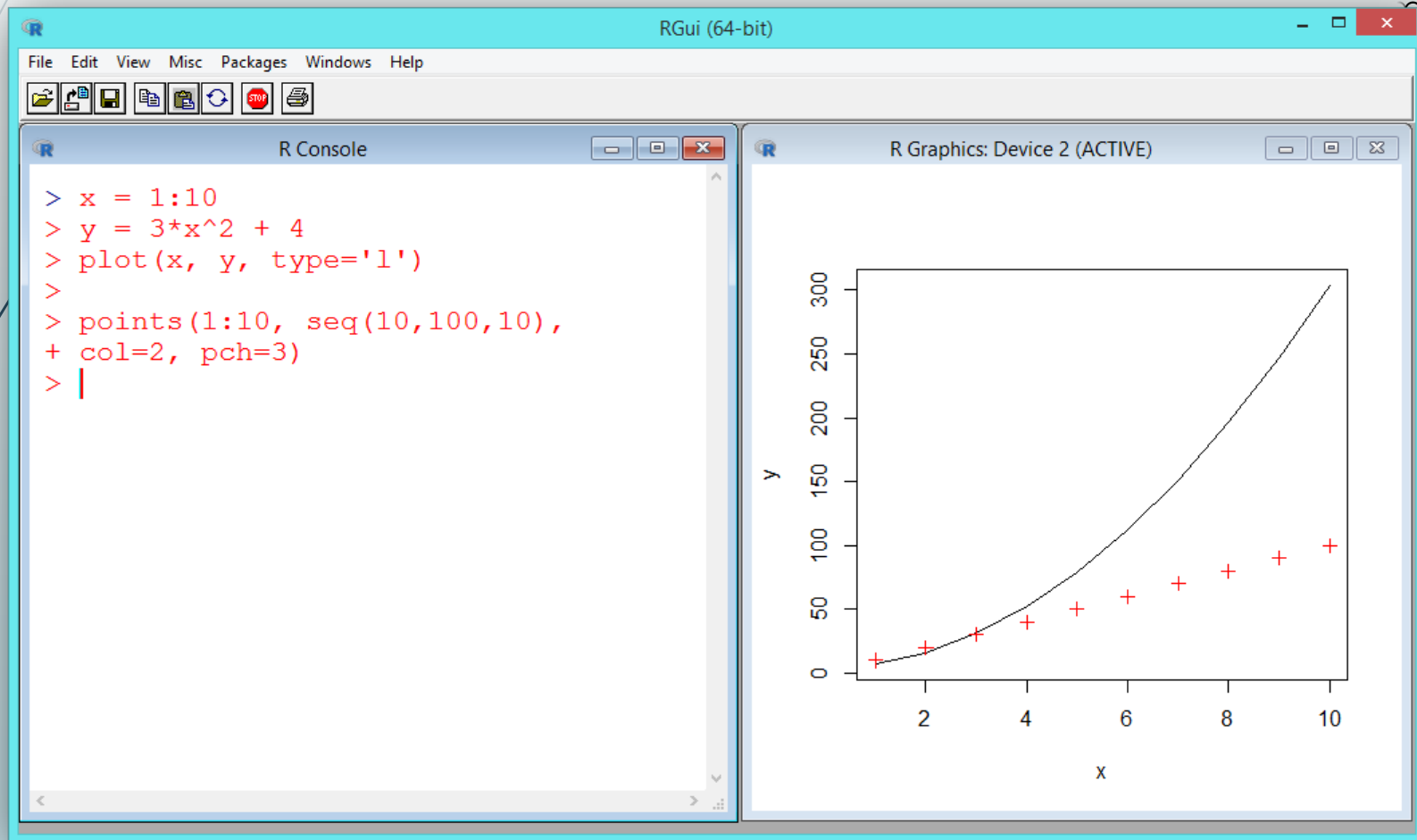
Plot

► points Function



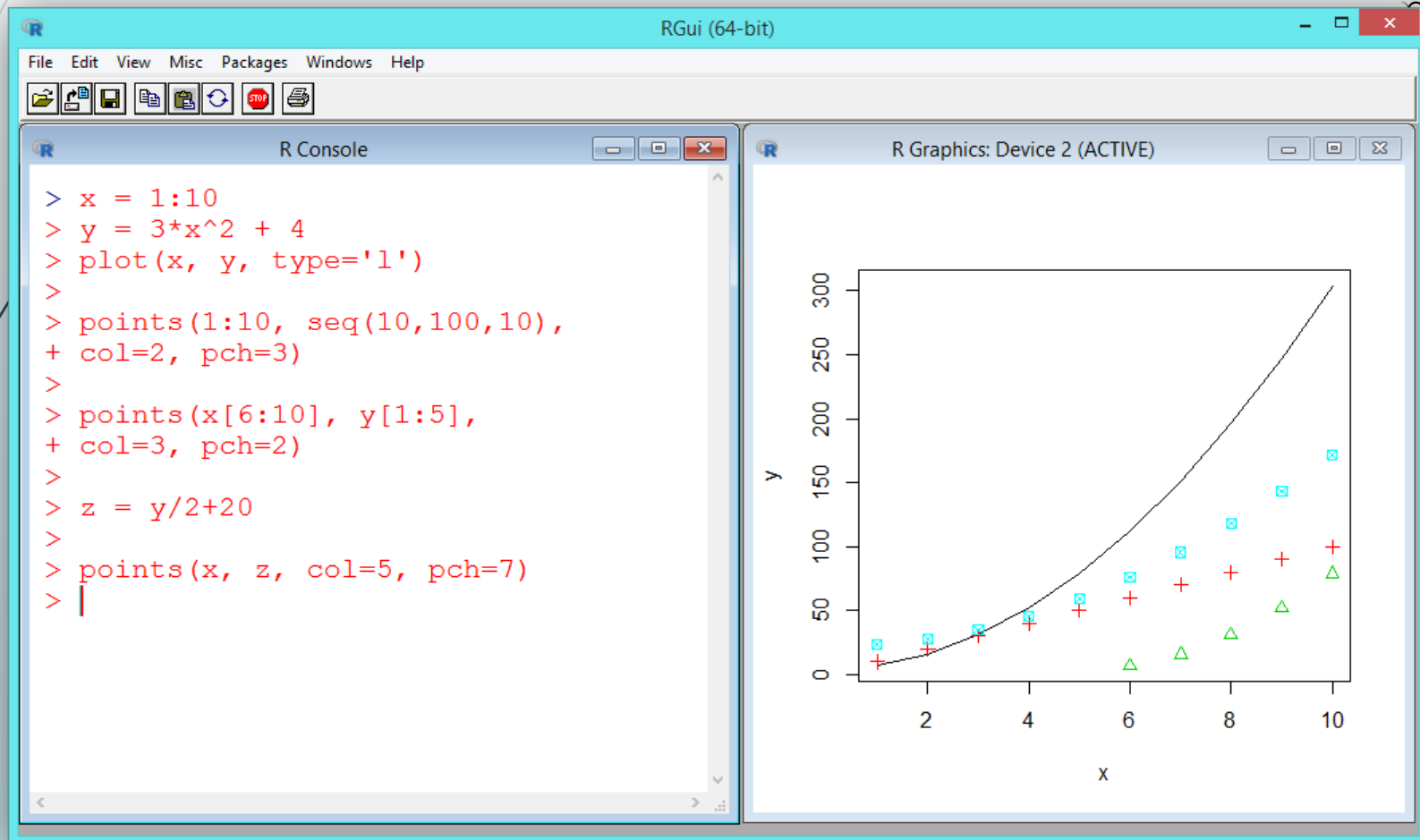
Plot

► points Function



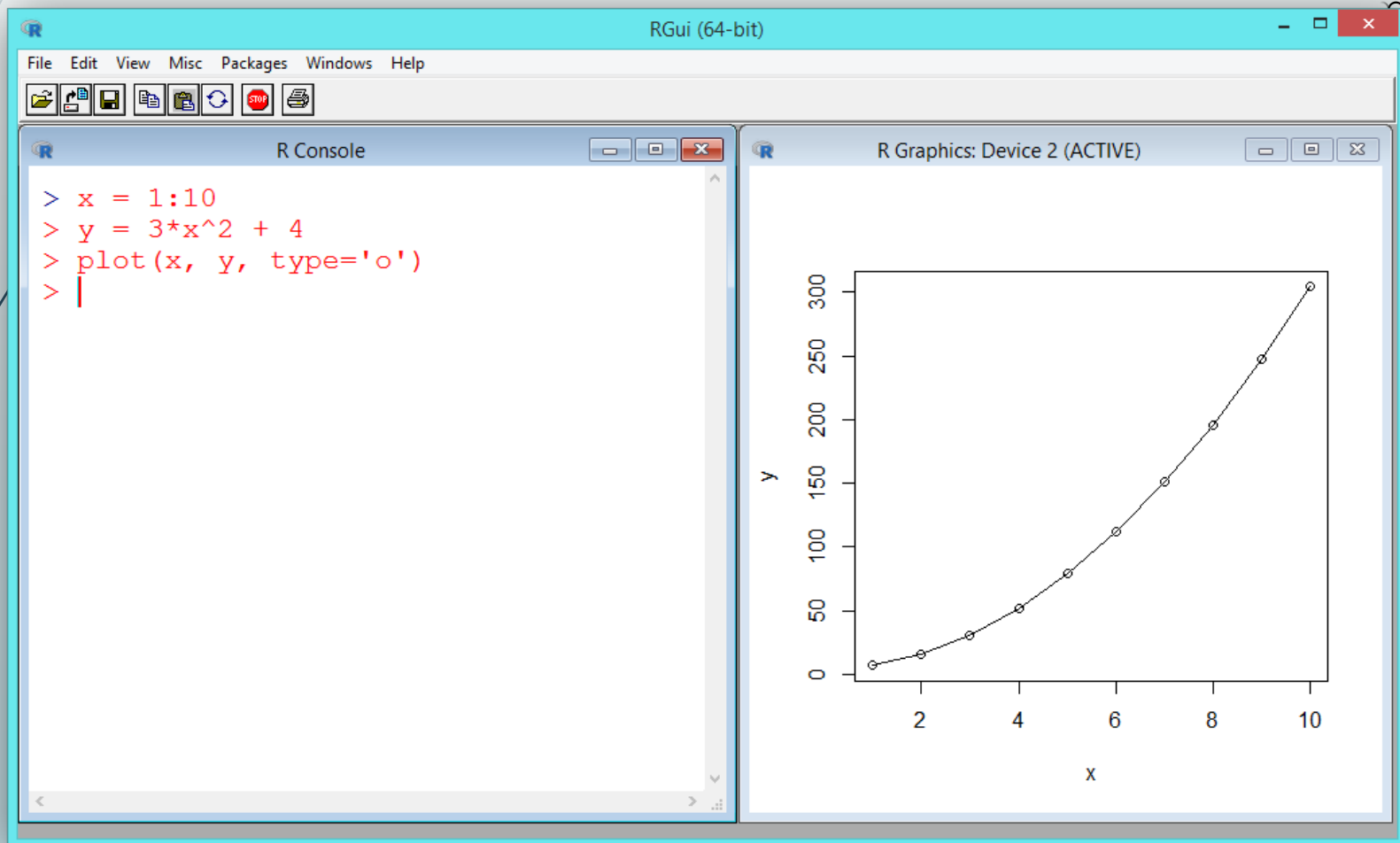
Plot

► points Function



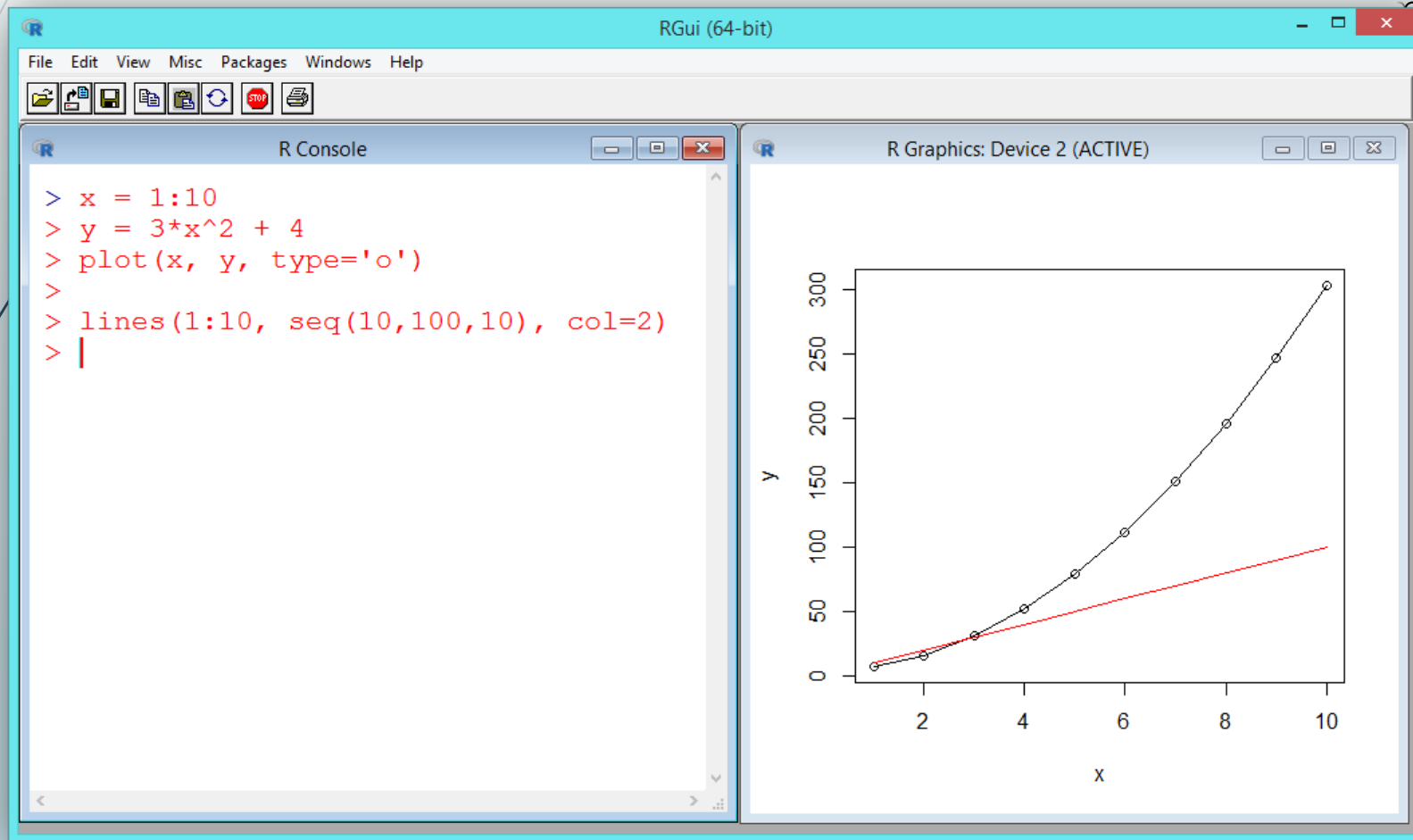
Plot

► `lines` Function



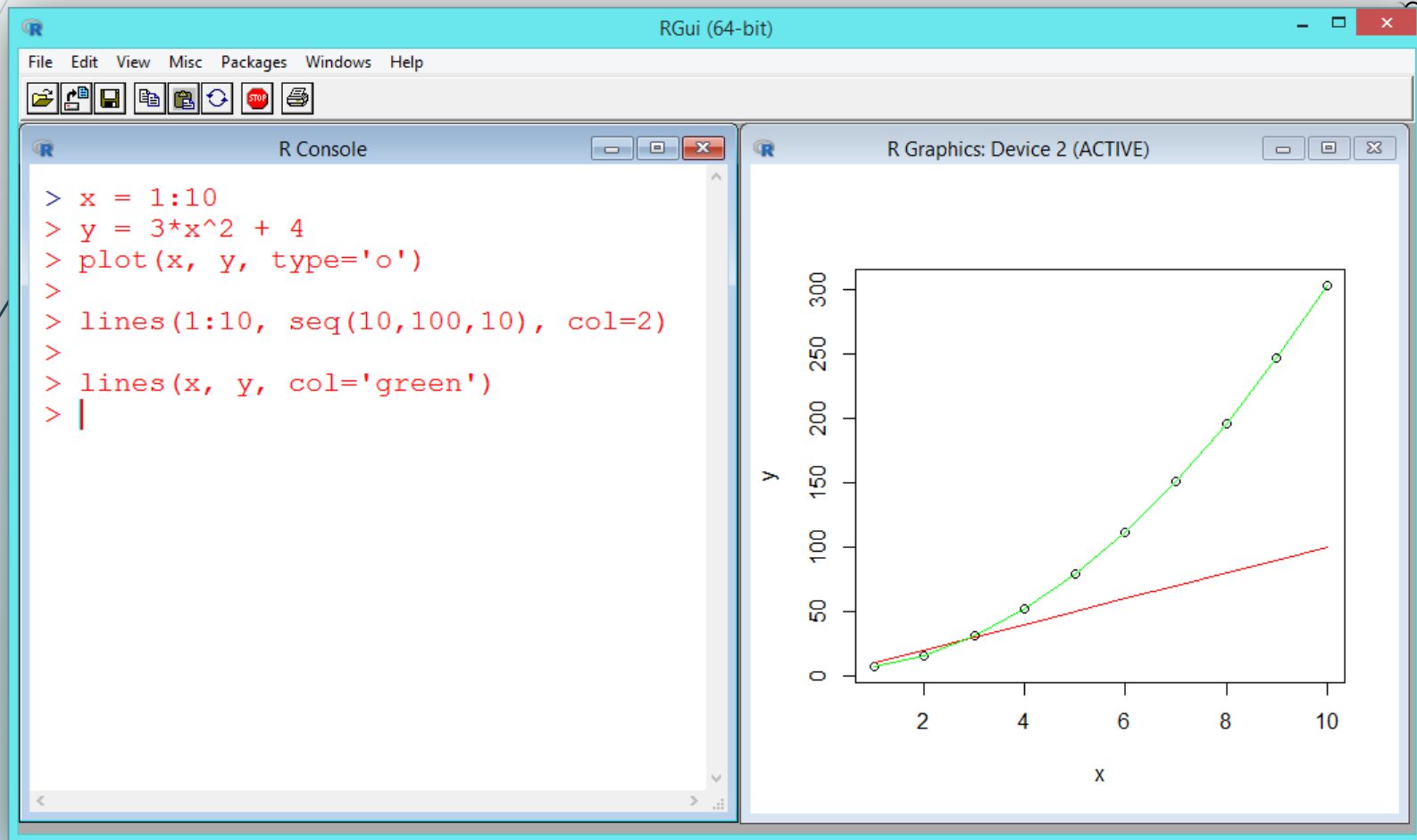
Plot

► lines Function



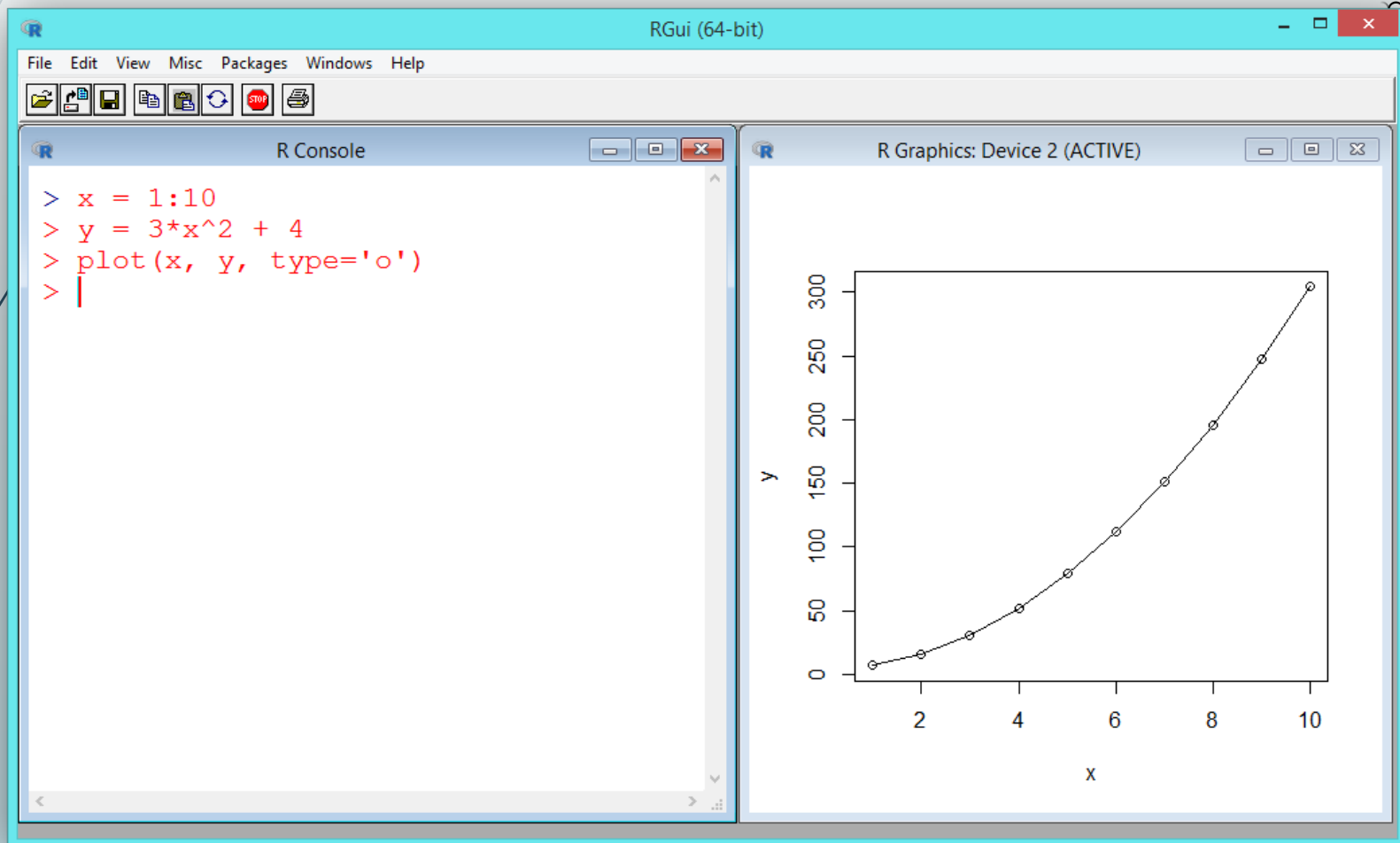
Plot

► lines Function



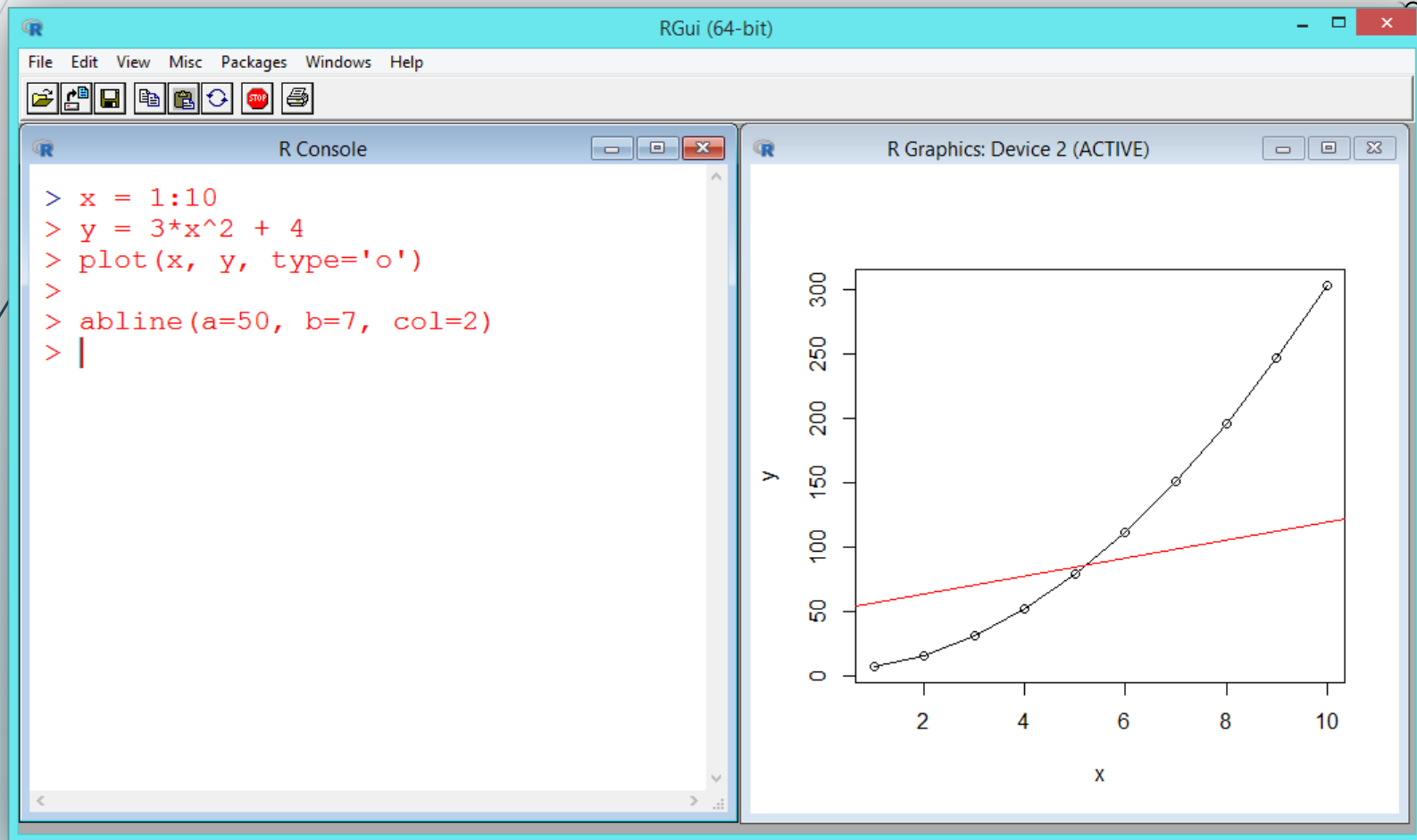
Plot

► abline Function



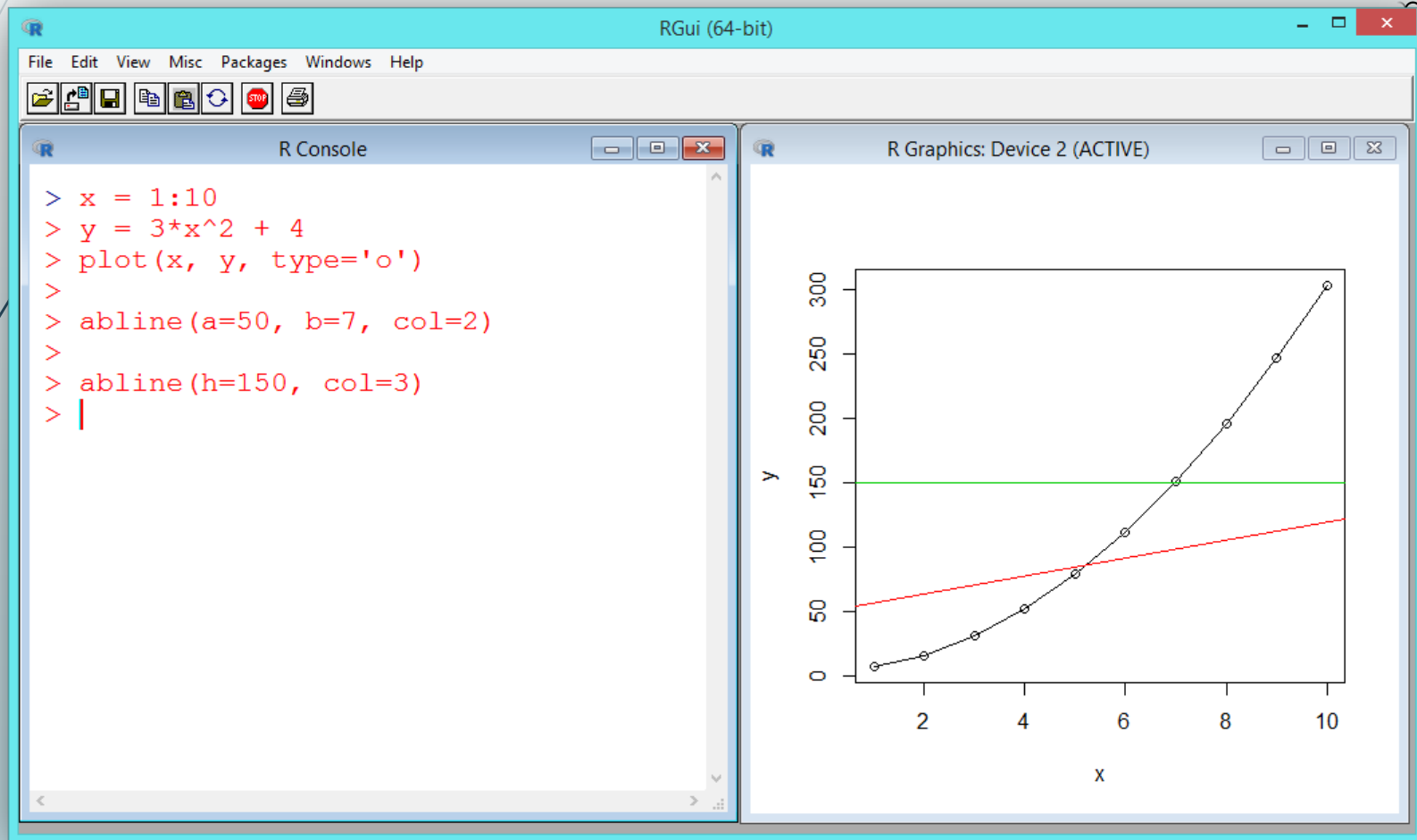
Plot

► abline Function



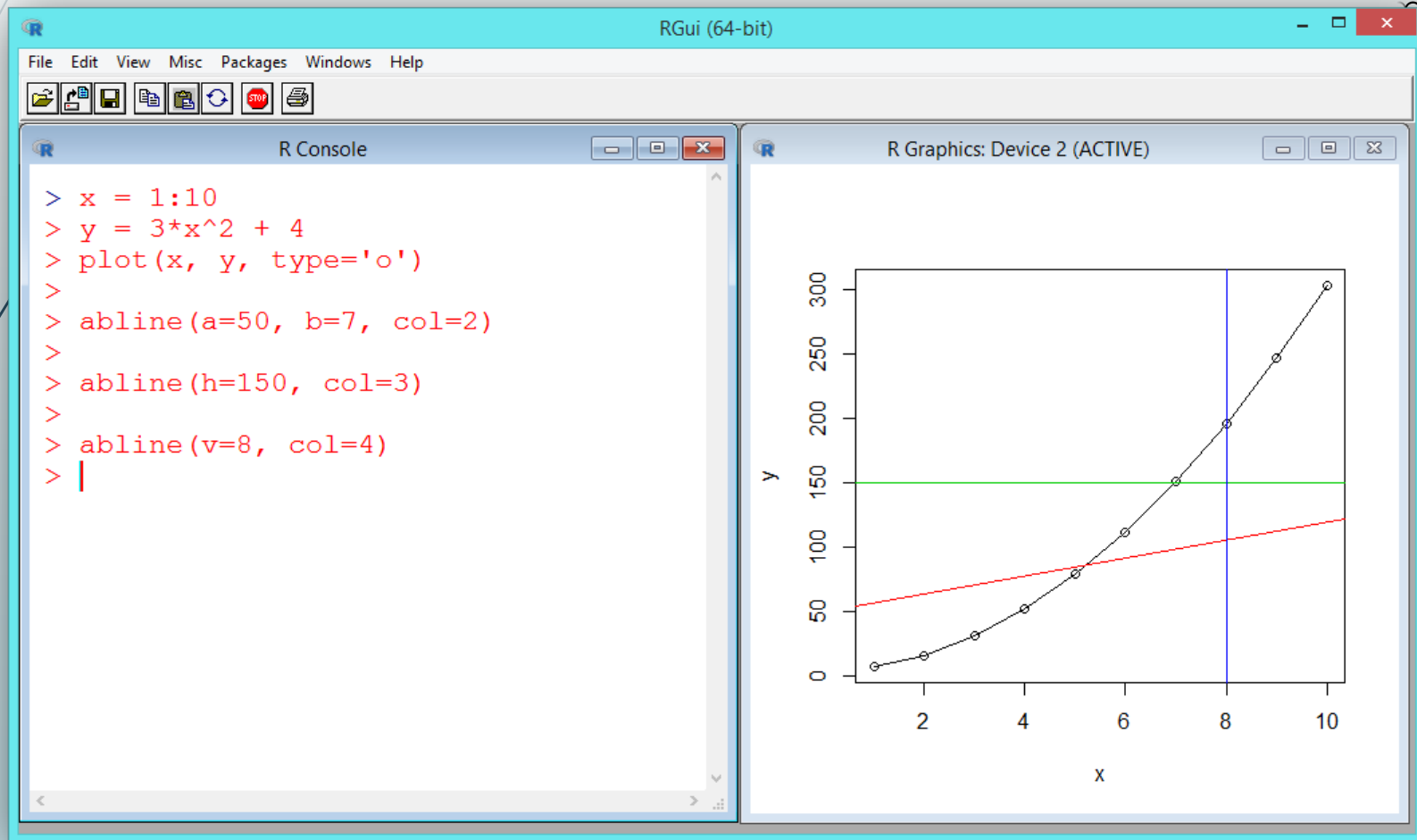
Plot

► abline Function



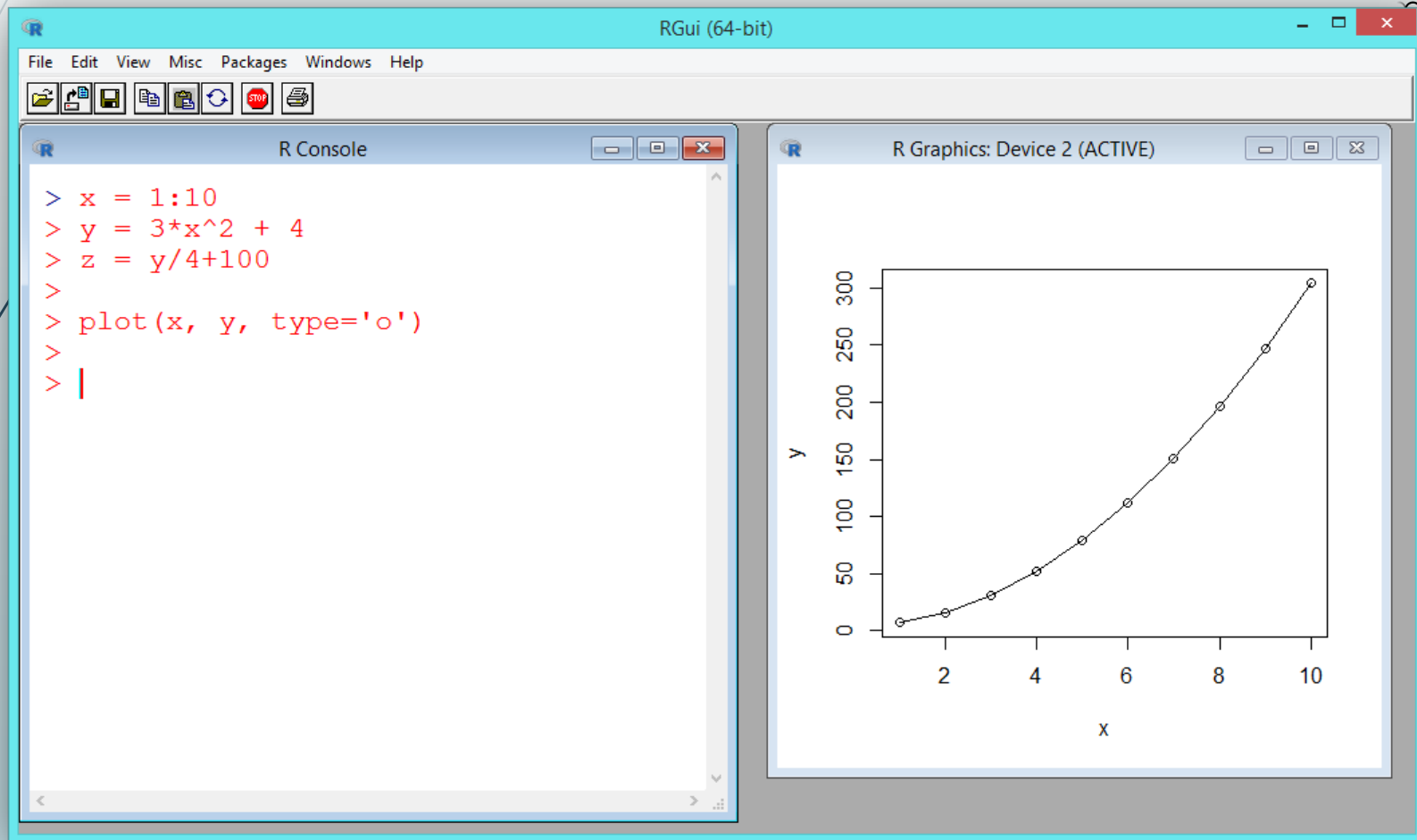
Plot

► abline Function



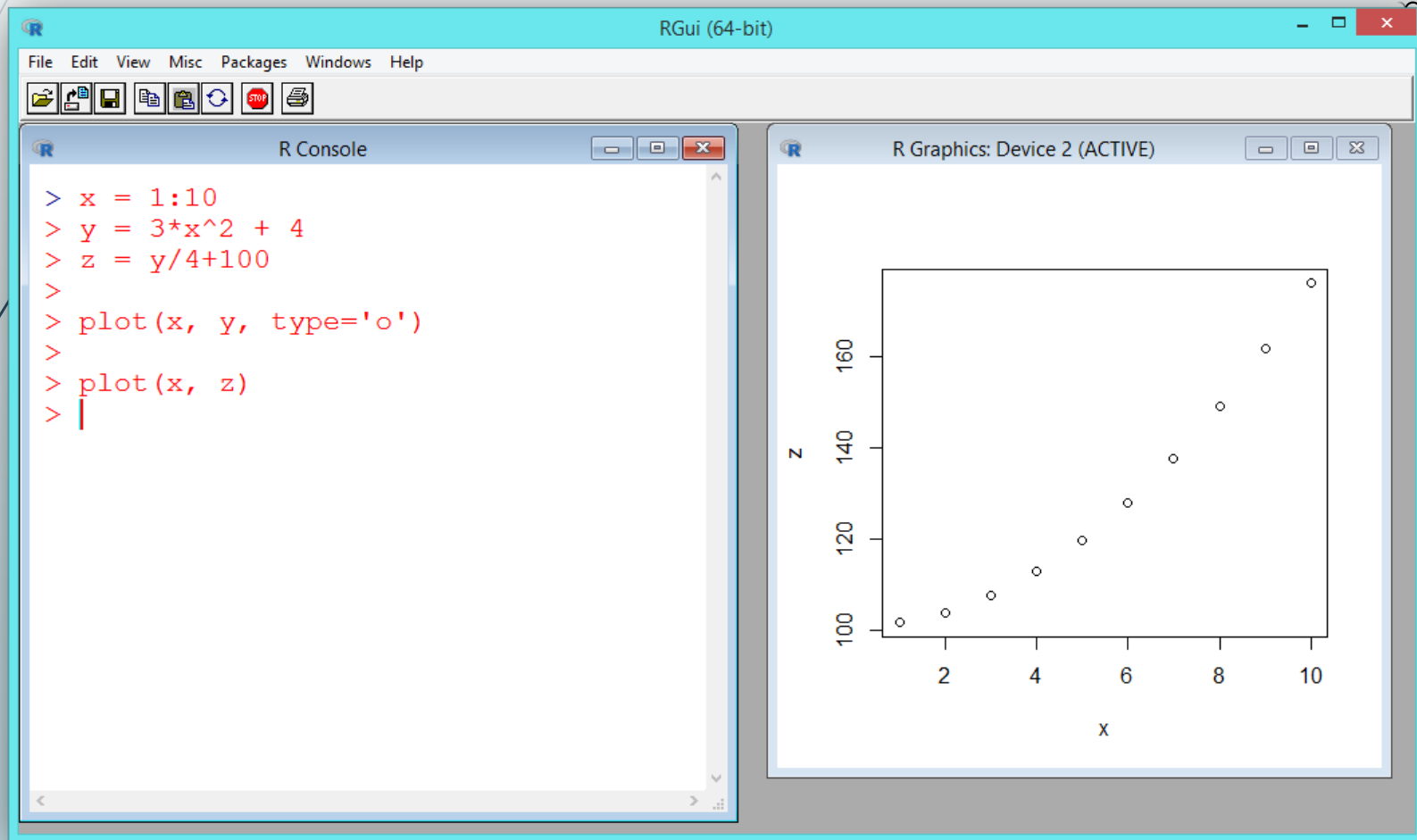
Plot

► windows Function



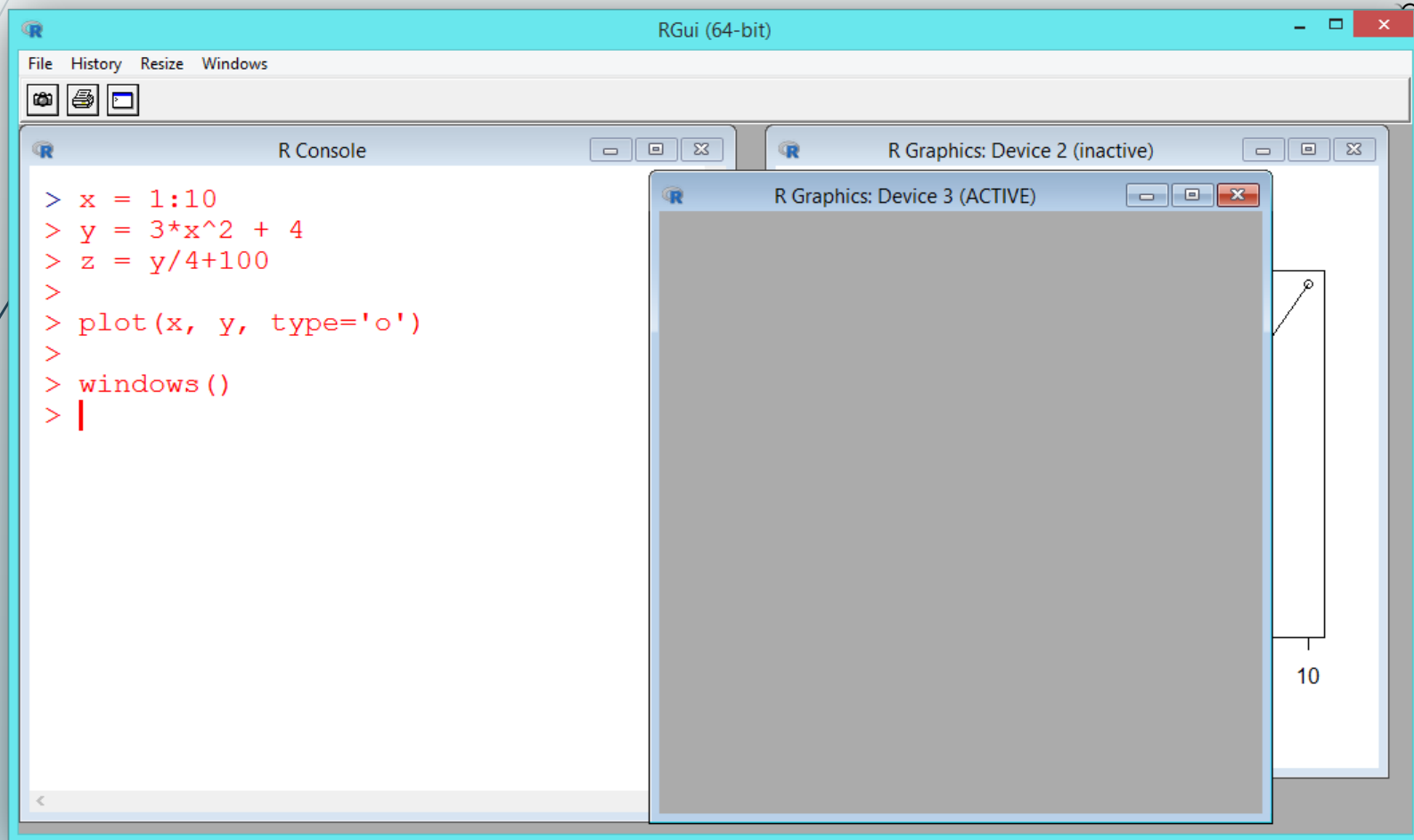
Plot

► windows Function



Plot

► windows Function



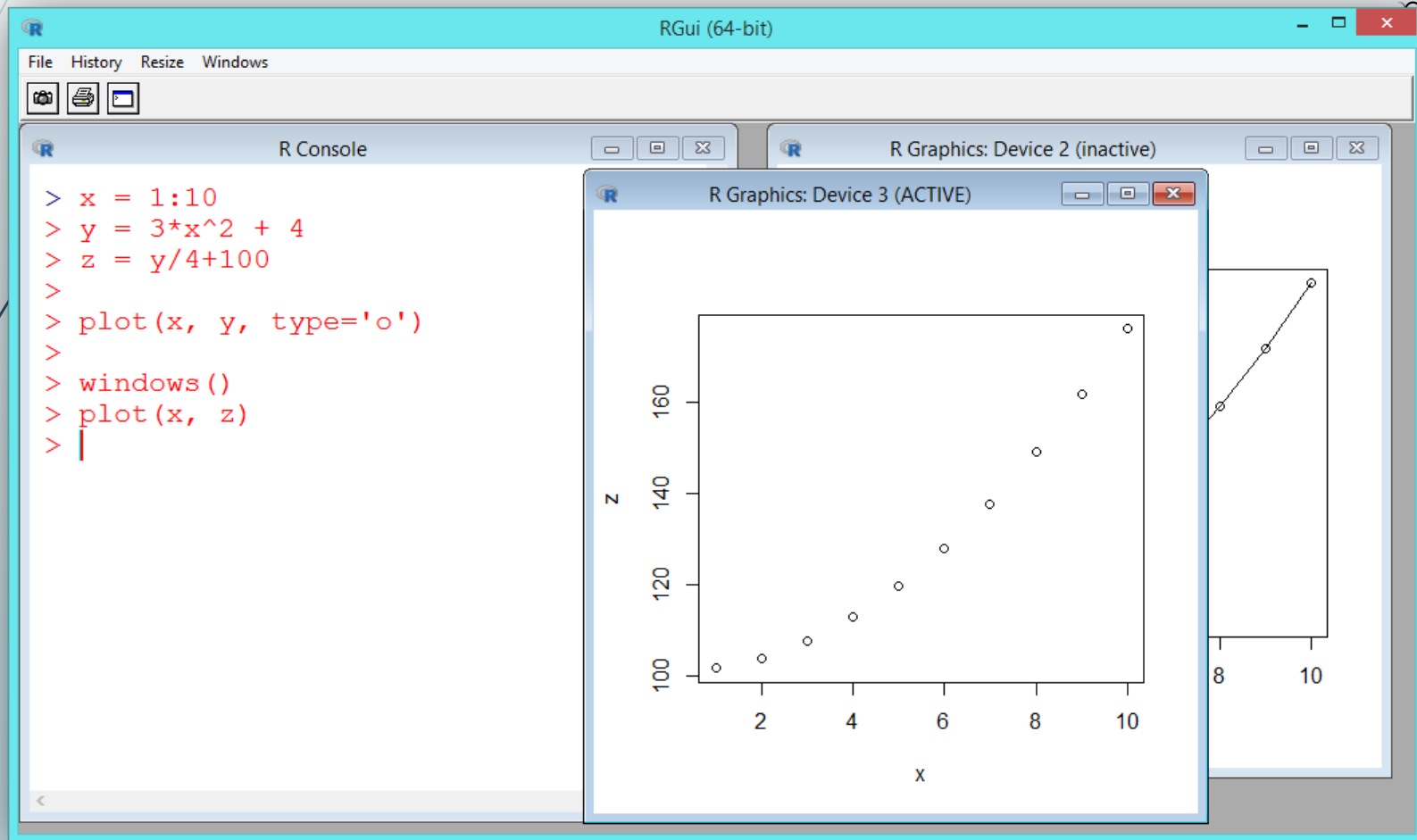
The screenshot displays the RGui (64-bit) interface. The main window contains three panes:

- R Console:** Shows the following R code:

```
> x = 1:10
> y = 3*x^2 + 4
> z = y/4+100
>
> plot(x, y, type='o')
>
> windows()
> |
```
- R Graphics: Device 2 (inactive):** This pane is currently inactive and shows a grey background.
- R Graphics: Device 3 (ACTIVE):** This pane is active and shows a plot of the data generated in the console. The plot displays a series of open circles representing the data points. The x-axis is labeled with the value 10, and the y-axis is labeled with the value 10.

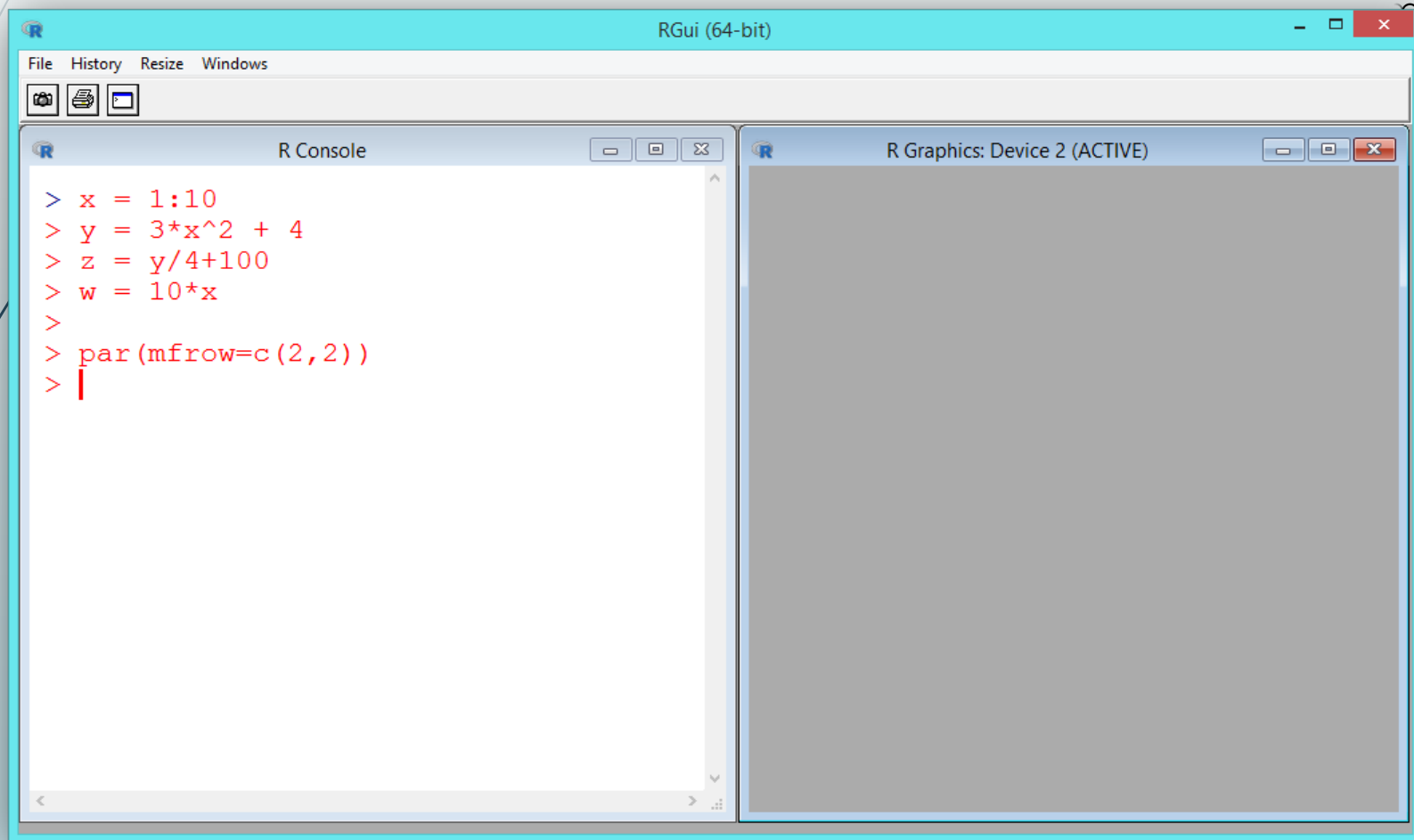
Plot

► windows Function



Plot

► Multiple Plots



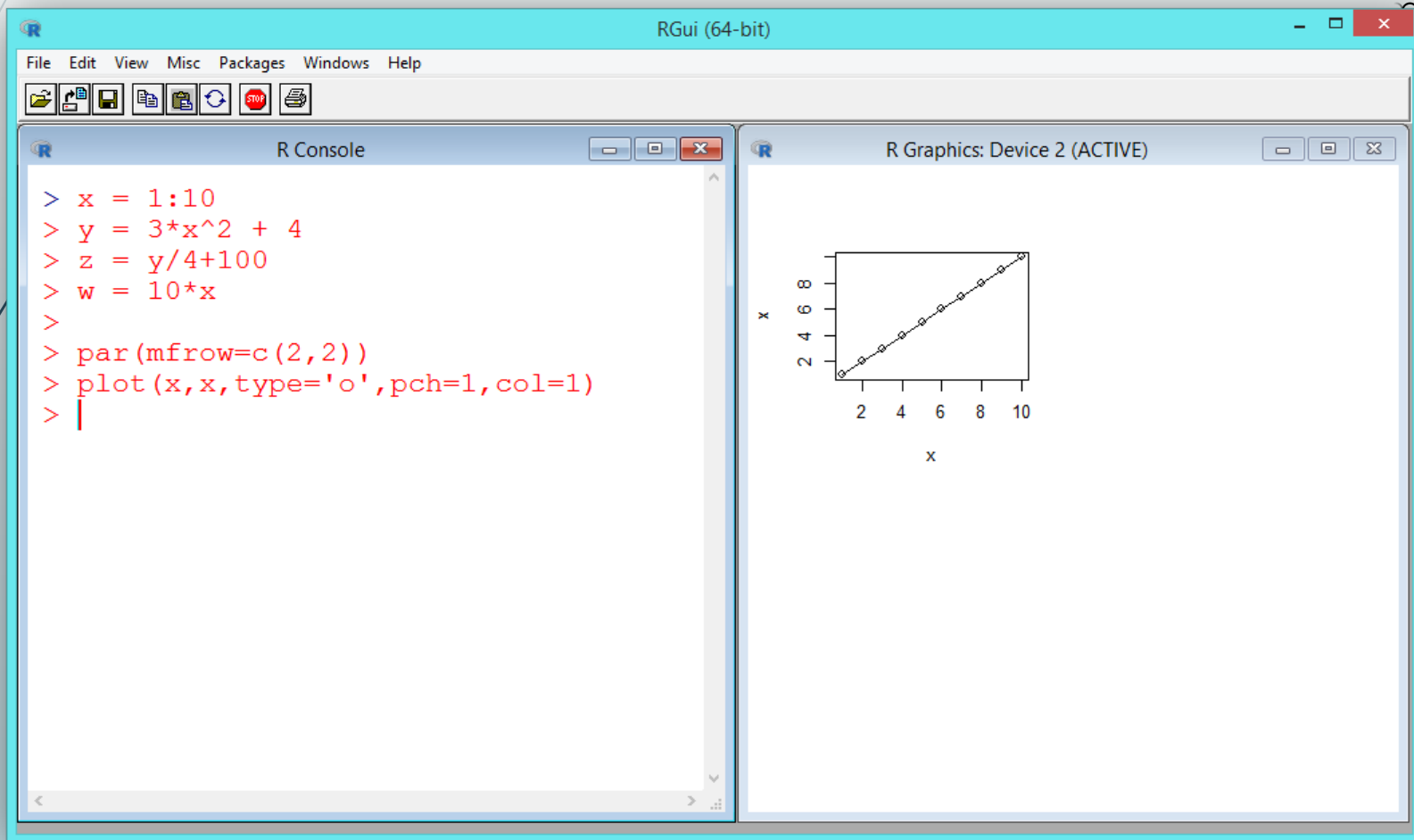
The screenshot displays the RGui (64-bit) interface. The main window is divided into two panes. The left pane, titled "R Console", contains the following R code:

```
> x = 1:10
> y = 3*x^2 + 4
> z = y/4+100
> w = 10*x
>
> par(mfrow=c(2,2))
> |
```

The right pane, titled "R Graphics: Device 2 (ACTIVE)", is currently blank, indicating that the plots have not yet been rendered.

Plot

► Multiple Plots: `par(mfrow=...)`



The screenshot displays the RGui (64-bit) interface. The R Console window shows the following code being executed:

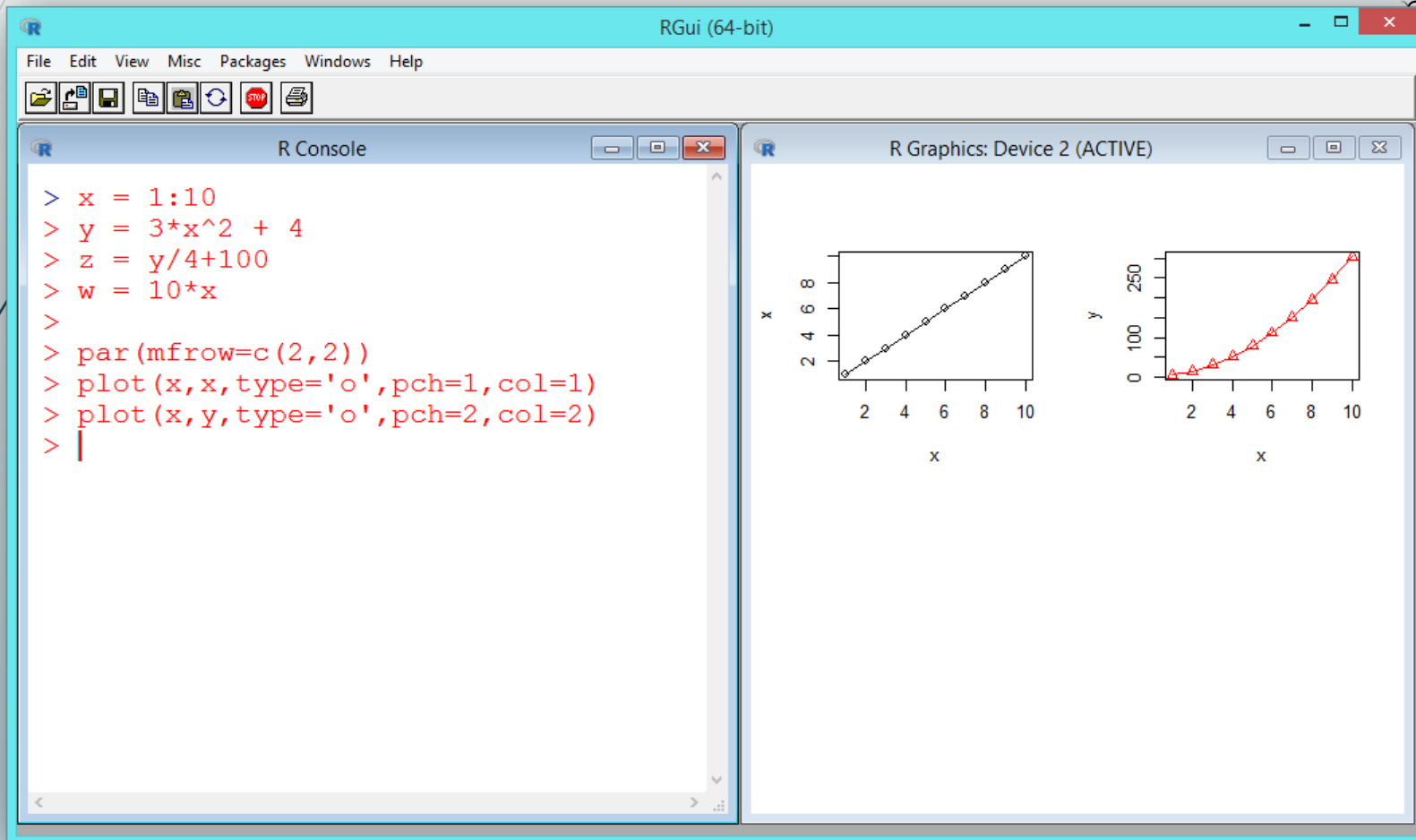
```
> x = 1:10
> y = 3*x^2 + 4
> z = y/4+100
> w = 10*x
>
> par(mfrow=c(2,2))
> plot(x,x,type='o',pch=1,col=1)
> |
```

The R Graphics: Device 2 (ACTIVE) window shows a plot of x versus x . The plot displays a series of points connected by a line, forming a straight line. The x-axis is labeled 'x' and ranges from 0 to 10 with major ticks at 2, 4, 6, 8, and 10. The y-axis is labeled 'x' and ranges from 0 to 10 with major ticks at 2, 4, 6, and 8. The data points are represented by open circles (type='o', pch=1, col=1).

x	y
1	7
2	16
3	27
4	40
5	55
6	72
7	91
8	112
9	135
10	160

Plot

► Multiple Plots: `par(mfrow=...)`



The screenshot shows the RGui (64-bit) interface. The R Console window contains the following code:

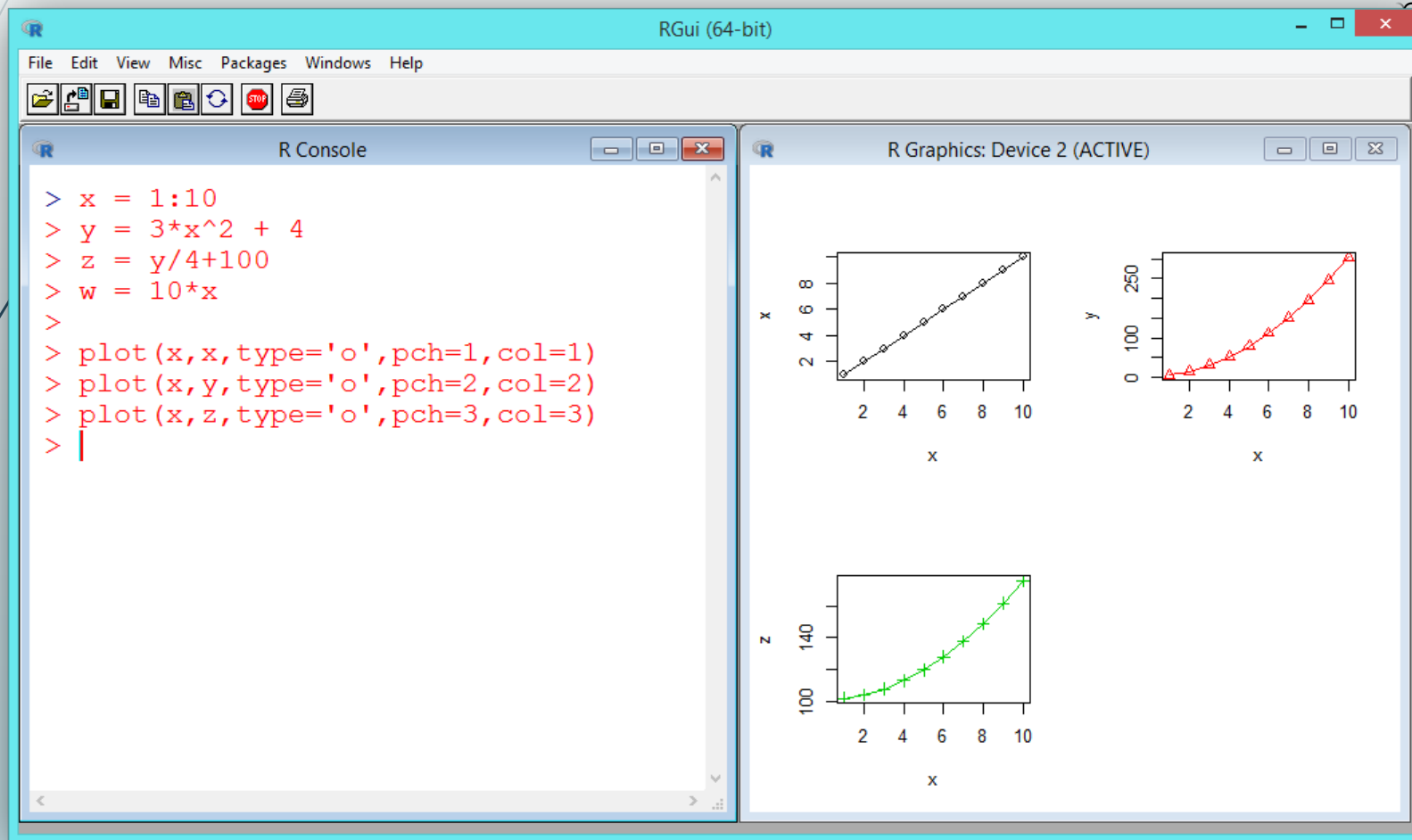
```
> x = 1:10
> y = 3*x^2 + 4
> z = y/4+100
> w = 10*x
>
> par(mfrow=c(2,2))
> plot(x,x,type='o',pch=1,col=1)
> plot(x,y,type='o',pch=2,col=2)
> |
```

The R Graphics: Device 2 (ACTIVE) window displays two plots side-by-side. The left plot shows a linear relationship between x and x, with data points represented by open circles (pch=1) connected by a line. The right plot shows a quadratic relationship between x and y, with data points represented by open triangles (pch=2) connected by a line. Both plots have x-axes ranging from 0 to 10.

x	x	y
1	1	7
2	2	16
3	3	27
4	4	40
5	5	55
6	6	72
7	7	91
8	8	112
9	9	135
10	10	160

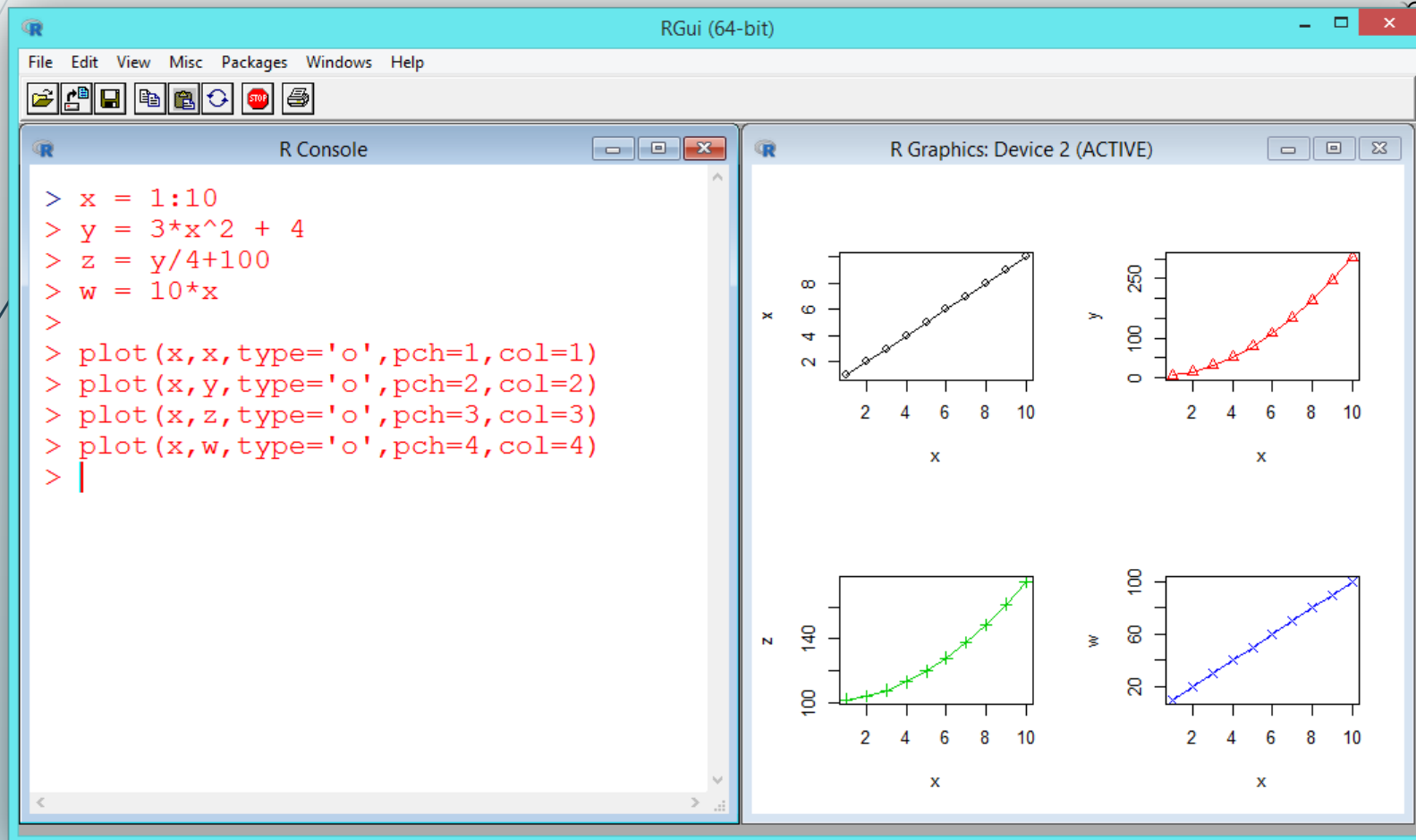
Plot

► Multiple Plots: `par(mfrow=...)`



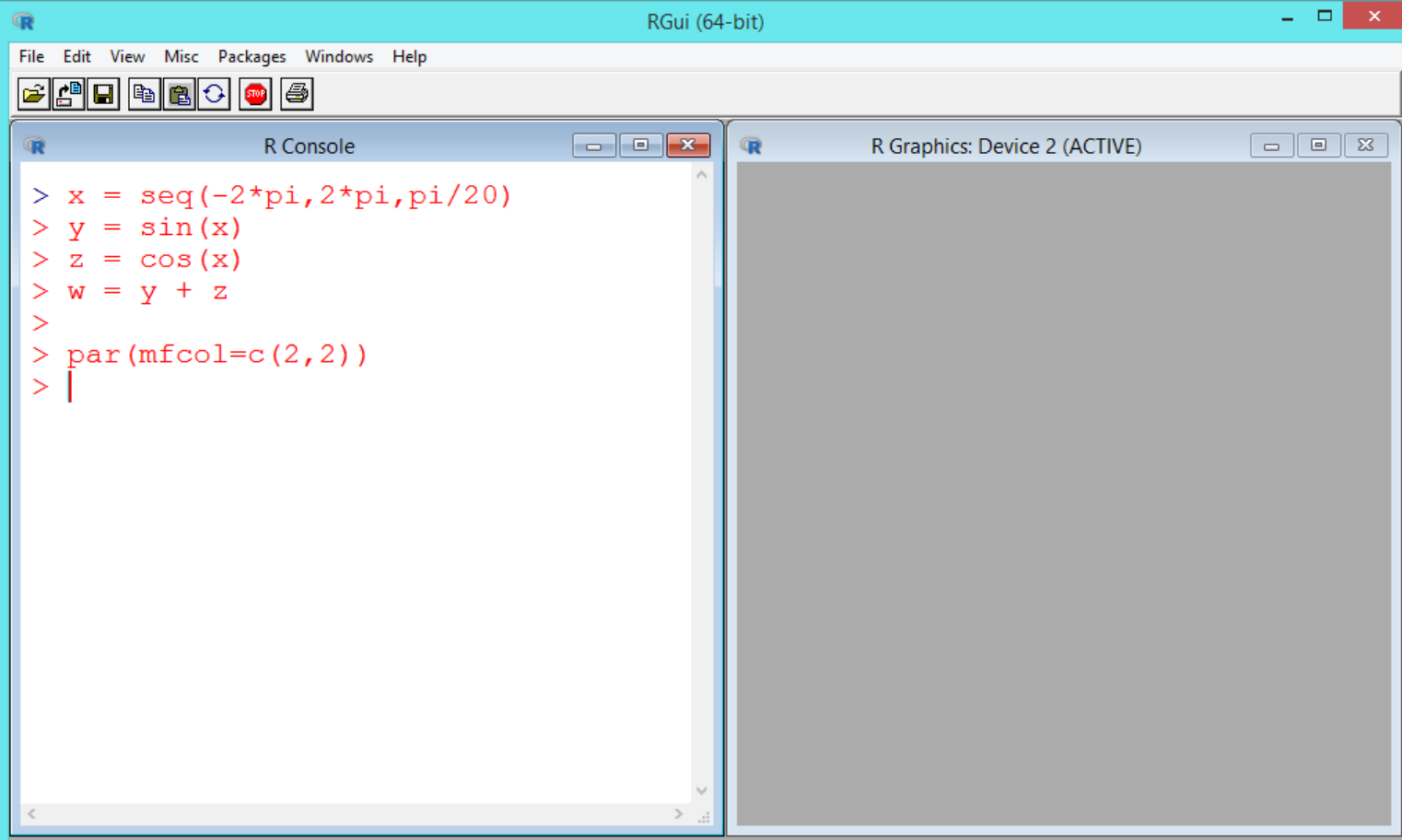
Plot

► Multiple Plots: `par(mfrow=...)`



Plot

➔ Multiple Plots: `par(mfcol=...)`



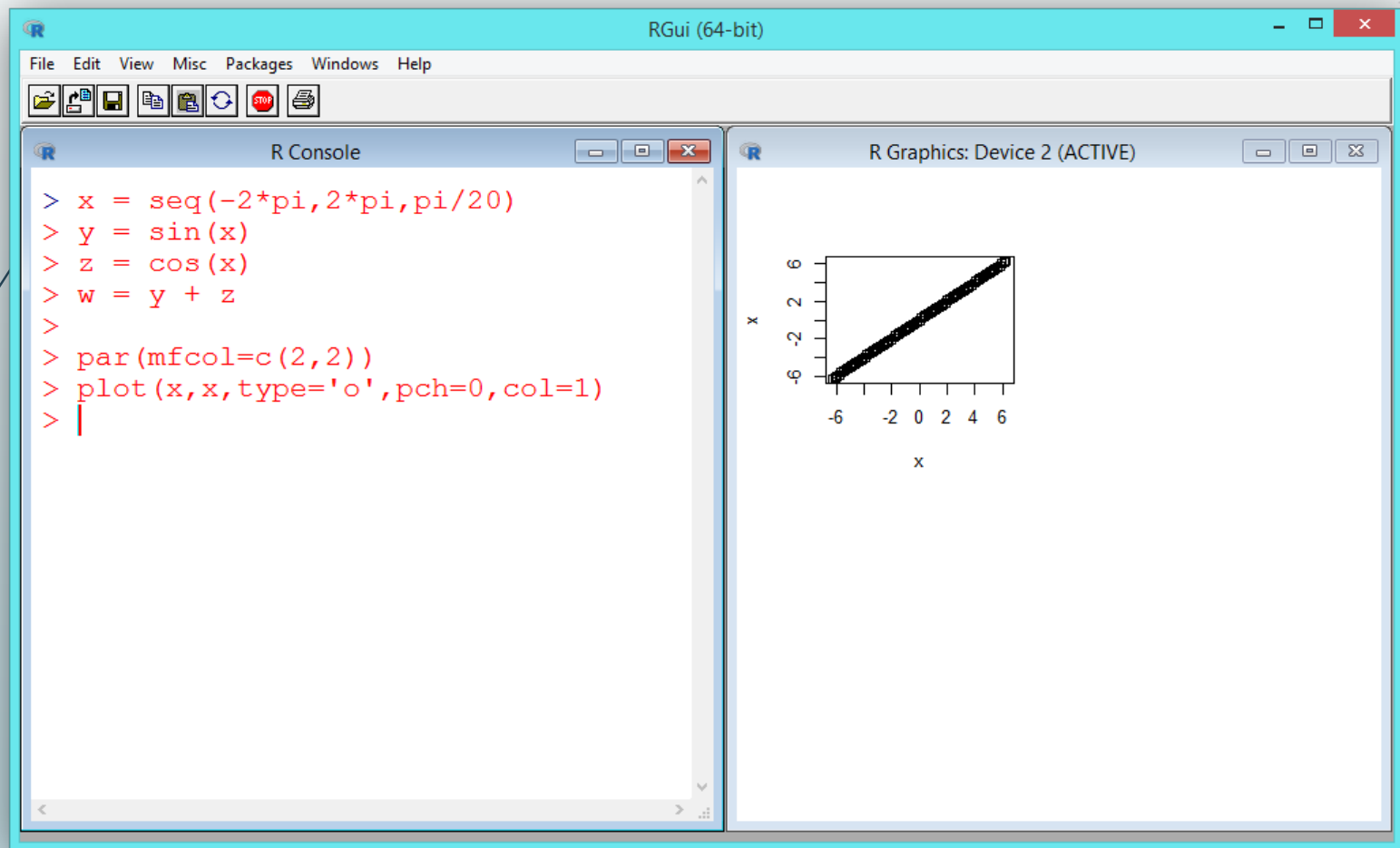
The screenshot shows the RGui (64-bit) interface. The R Console window contains the following code:

```
> x = seq(-2*pi, 2*pi, pi/20)
> y = sin(x)
> z = cos(x)
> w = y + z
>
> par(mfcol=c(2,2))
> |
```

The R Graphics window (Device 2 (ACTIVE)) is currently blank.

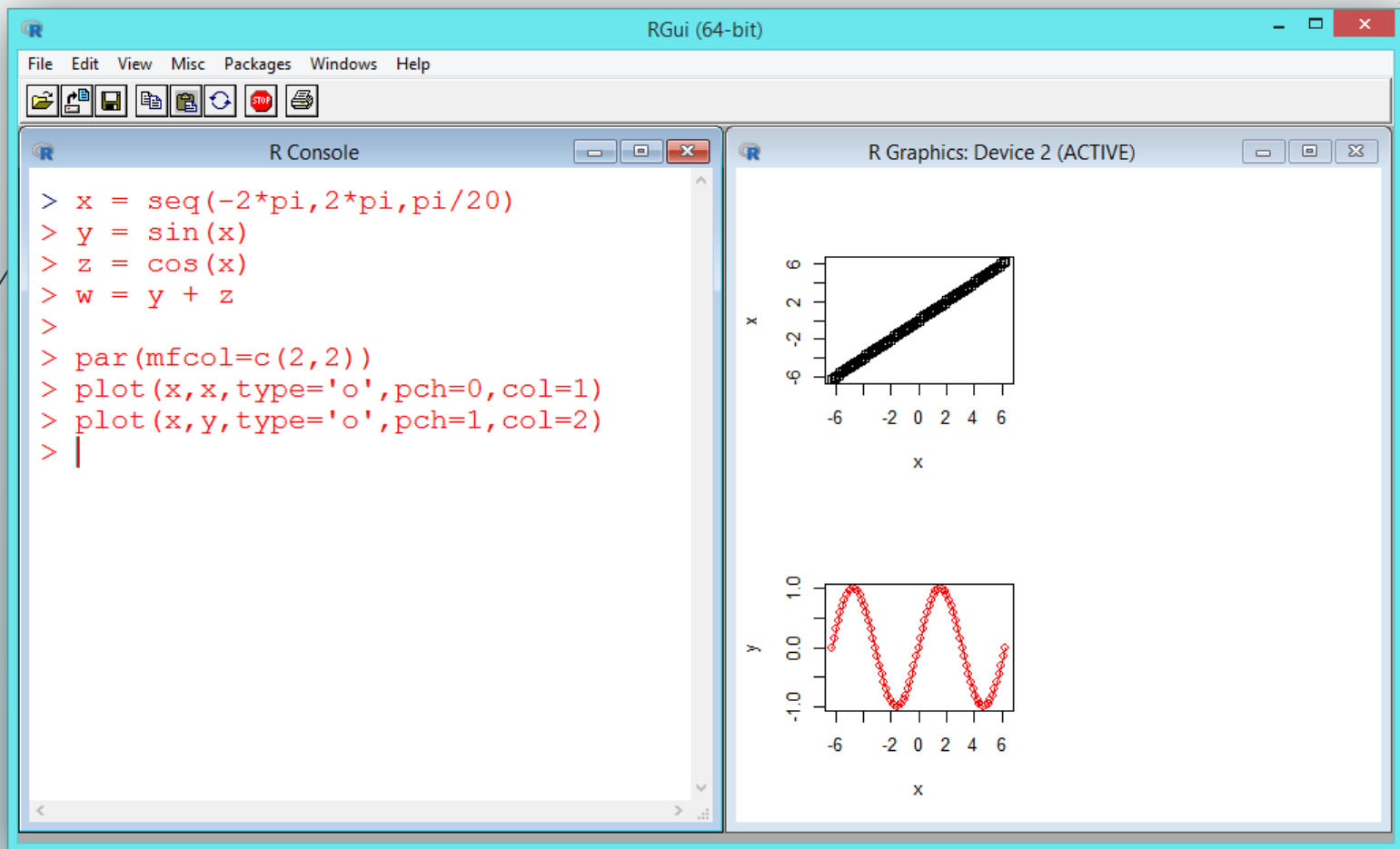
Plot

► Multiple Plots: `par(mfcol=...)`



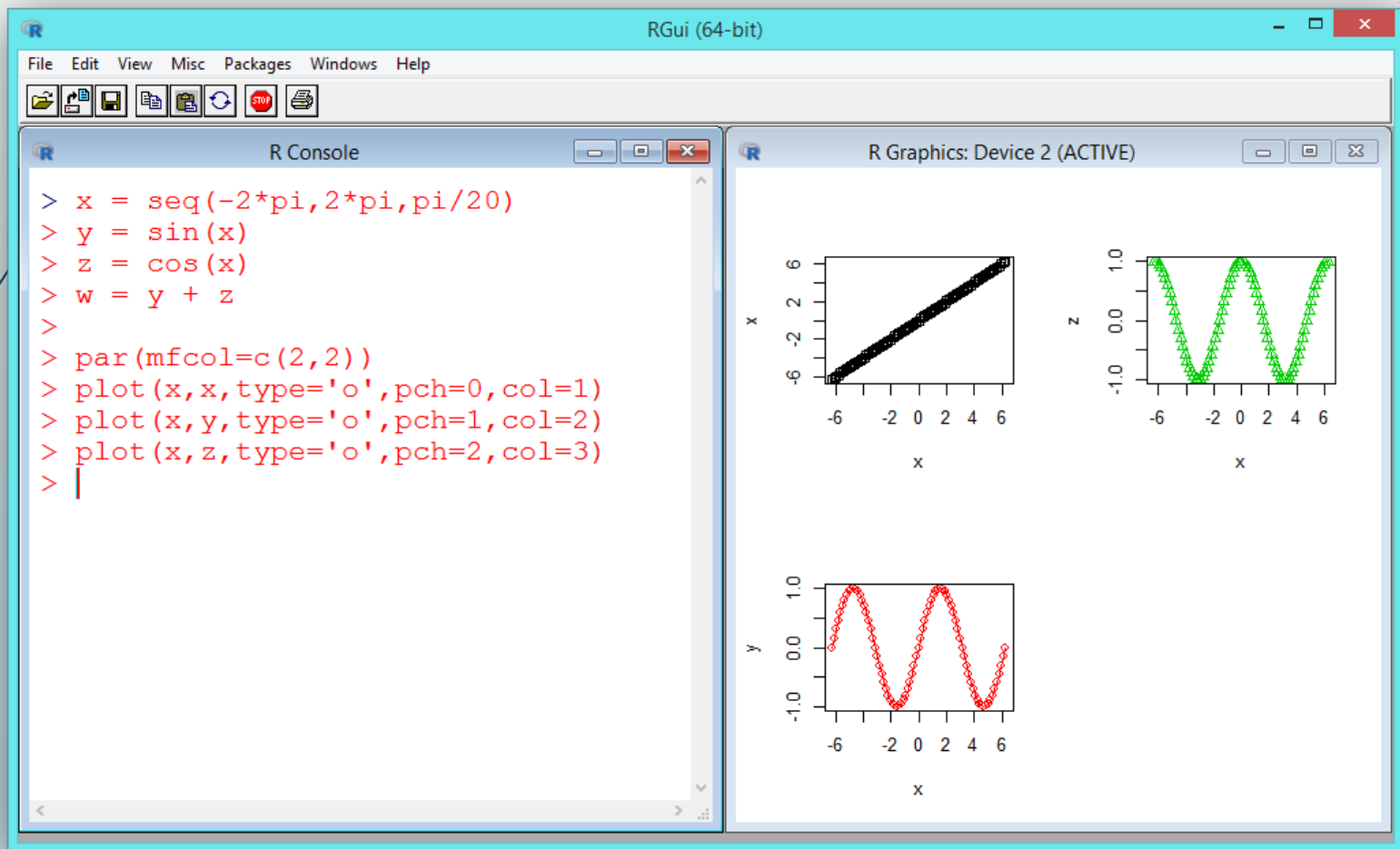
Plot

► Multiple Plots: `par(mfcol=...)`



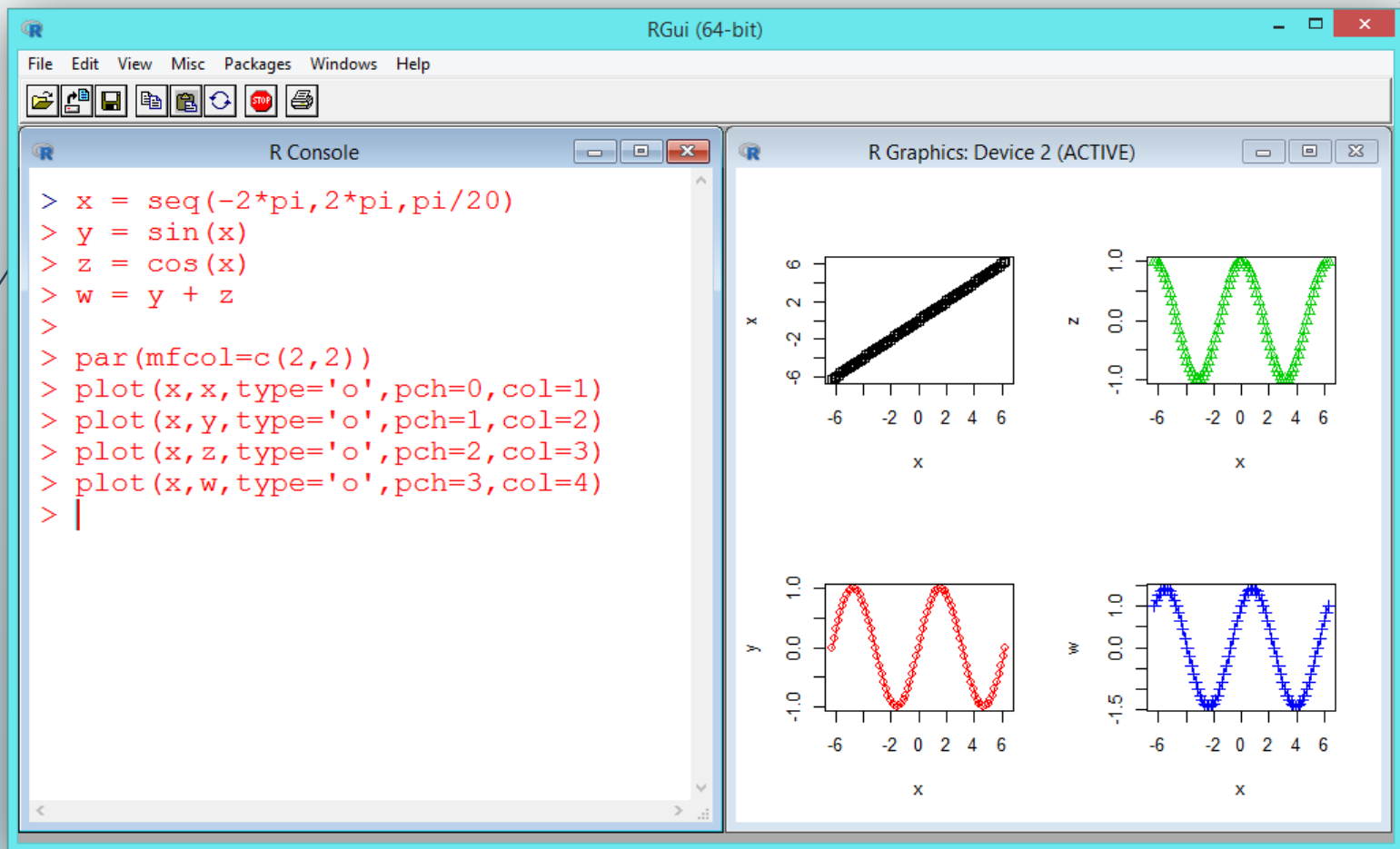
Plot

► Multiple Plots: `par(mfcol=...)`



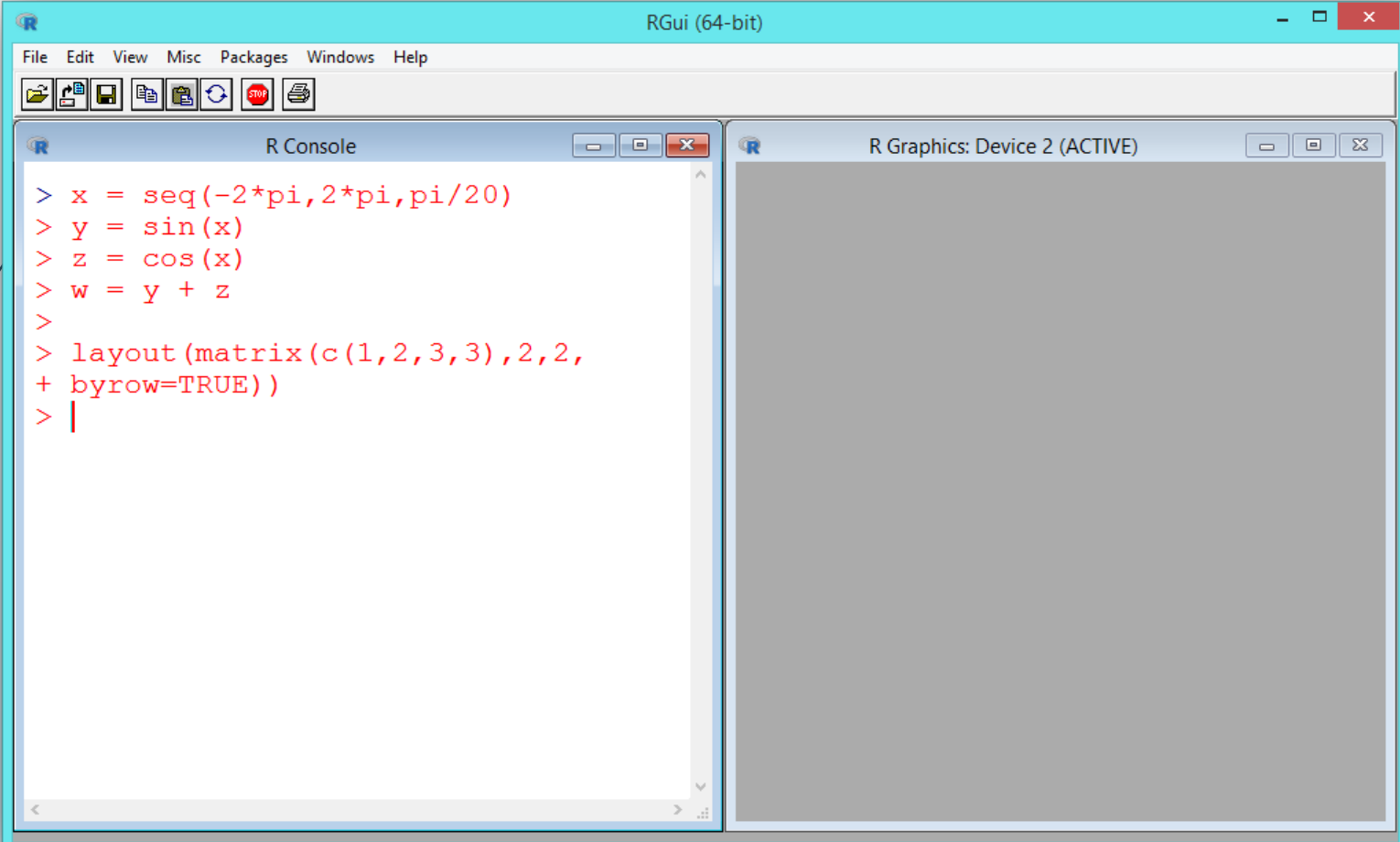
Plot

► Multiple Plots: `par(mfcol=...)`



Plot

► Multiple Plots: layout Function



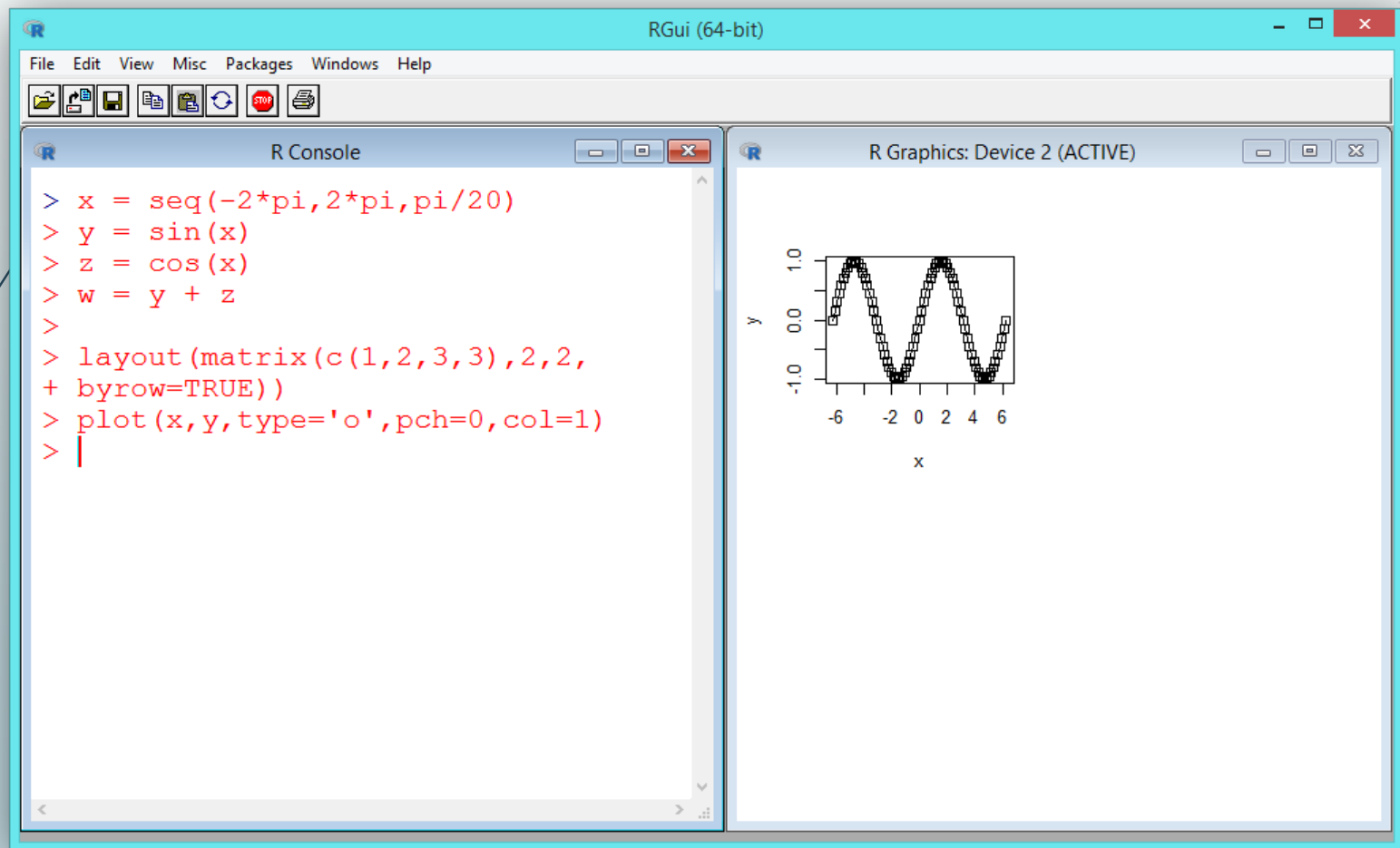
The screenshot shows the RGui (64-bit) interface. The R Console window contains the following code:

```
> x = seq(-2*pi, 2*pi, pi/20)
> y = sin(x)
> z = cos(x)
> w = y + z
>
> layout(matrix(c(1, 2, 3, 3), 2, 2,
+ byrow=TRUE))
> |
```

The R Graphics window is currently blank, indicating that the plots have not yet been rendered.

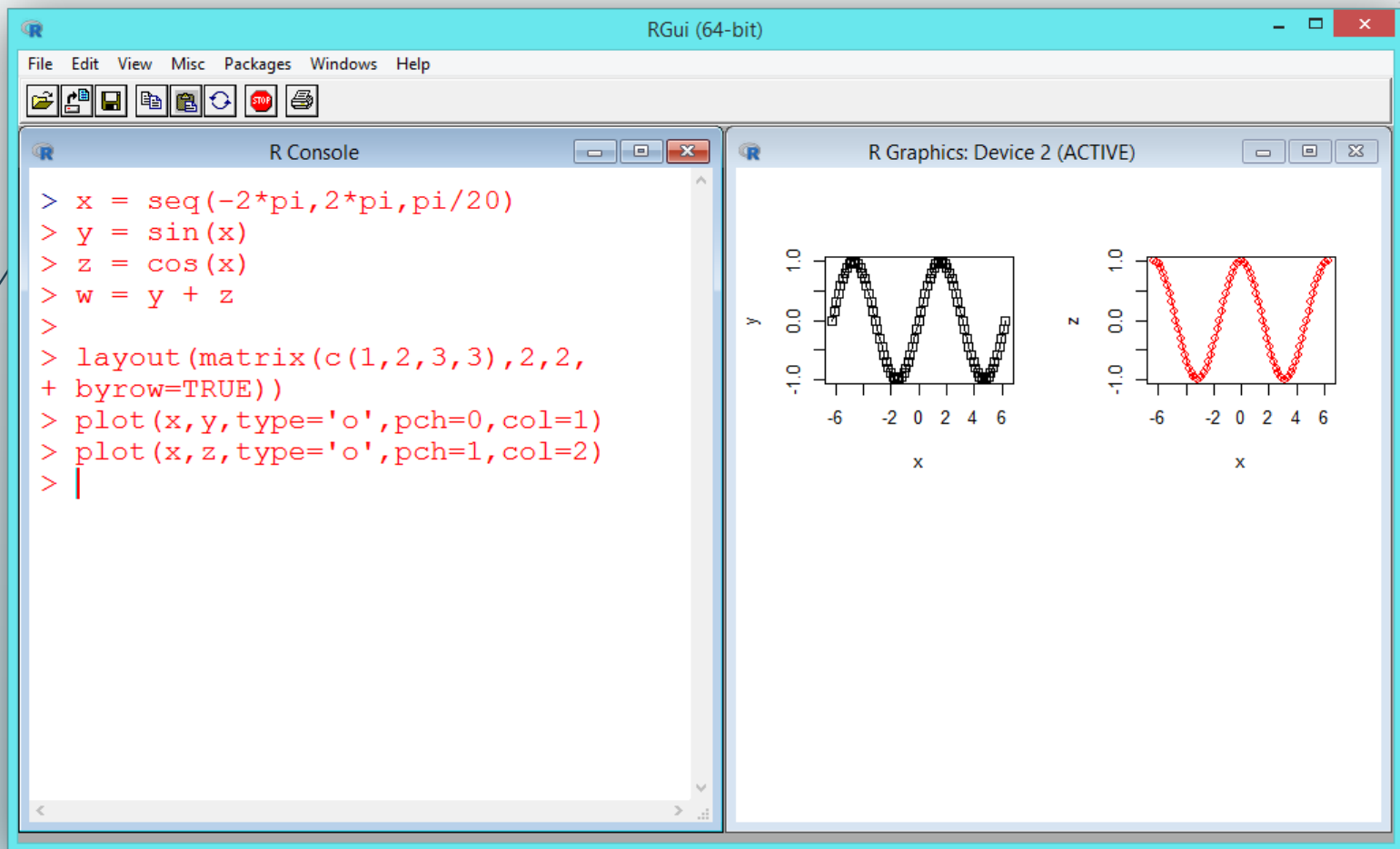
Plot

► Multiple Plots: layout Function



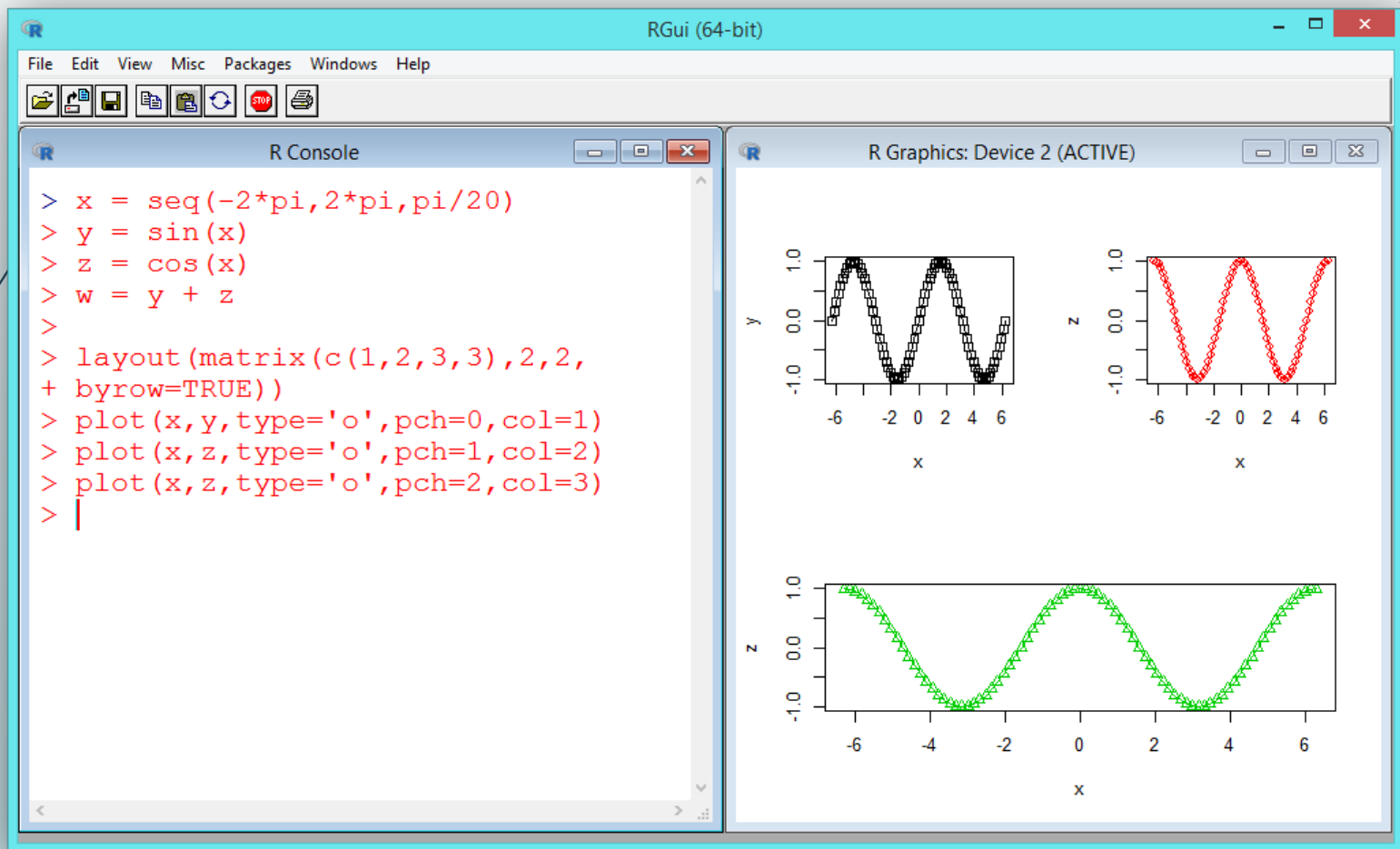
Plot

► Multiple Plots: layout Function

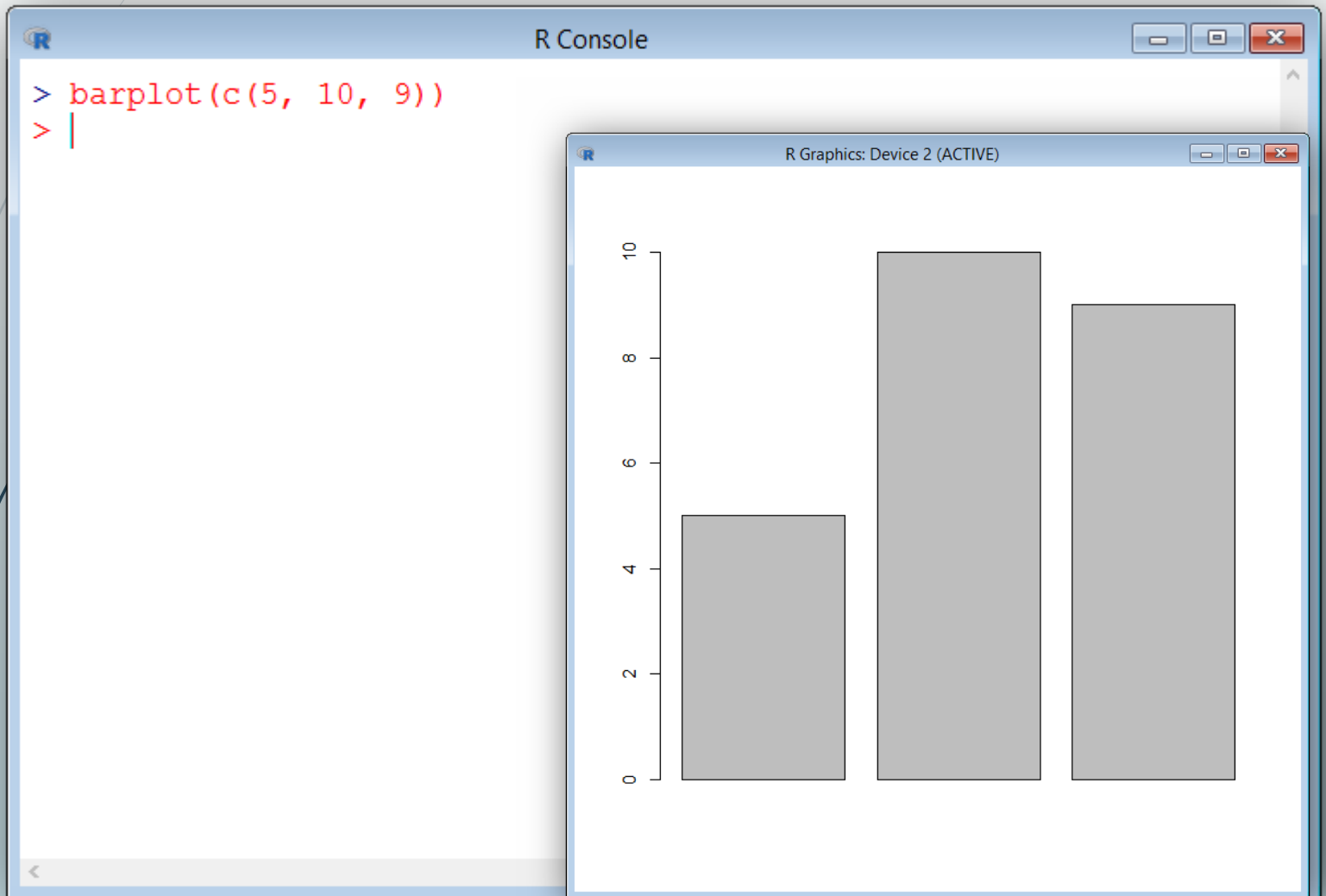


Plot

► Multiple Plots: layout Function

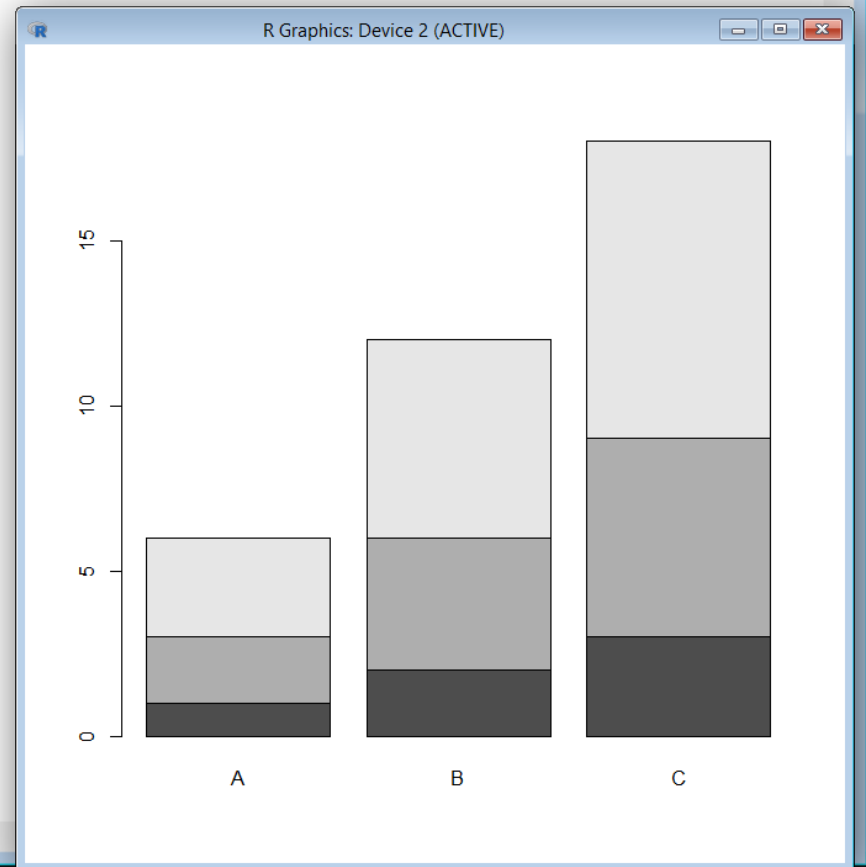


barplot()



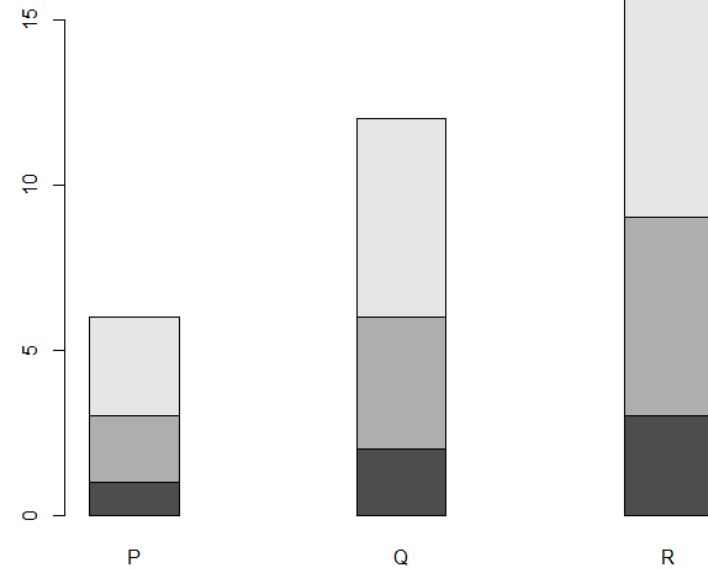
barplot()

```
R Console
> DF = data.frame(A = 1:3, B = c(2, 4, 6), C = seq(3, 9, 3))
> DF
  A B C
1 1 2 3
2 2 4 6
3 3 6 9
>
> barplot(DF)
Error in barplot.default(DF) :
>
> barplot(as.matrix(DF))
> |
```

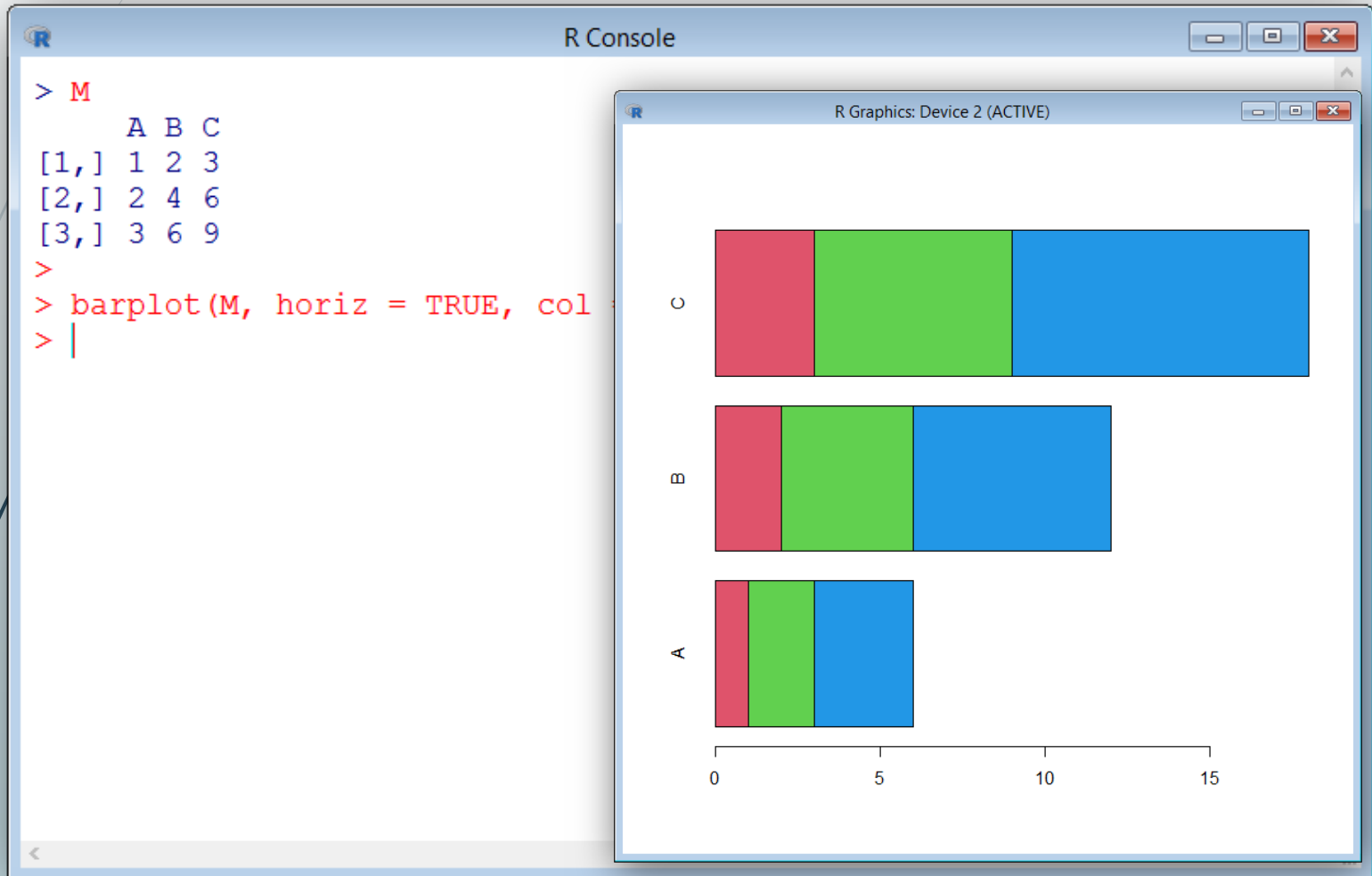


barplot()

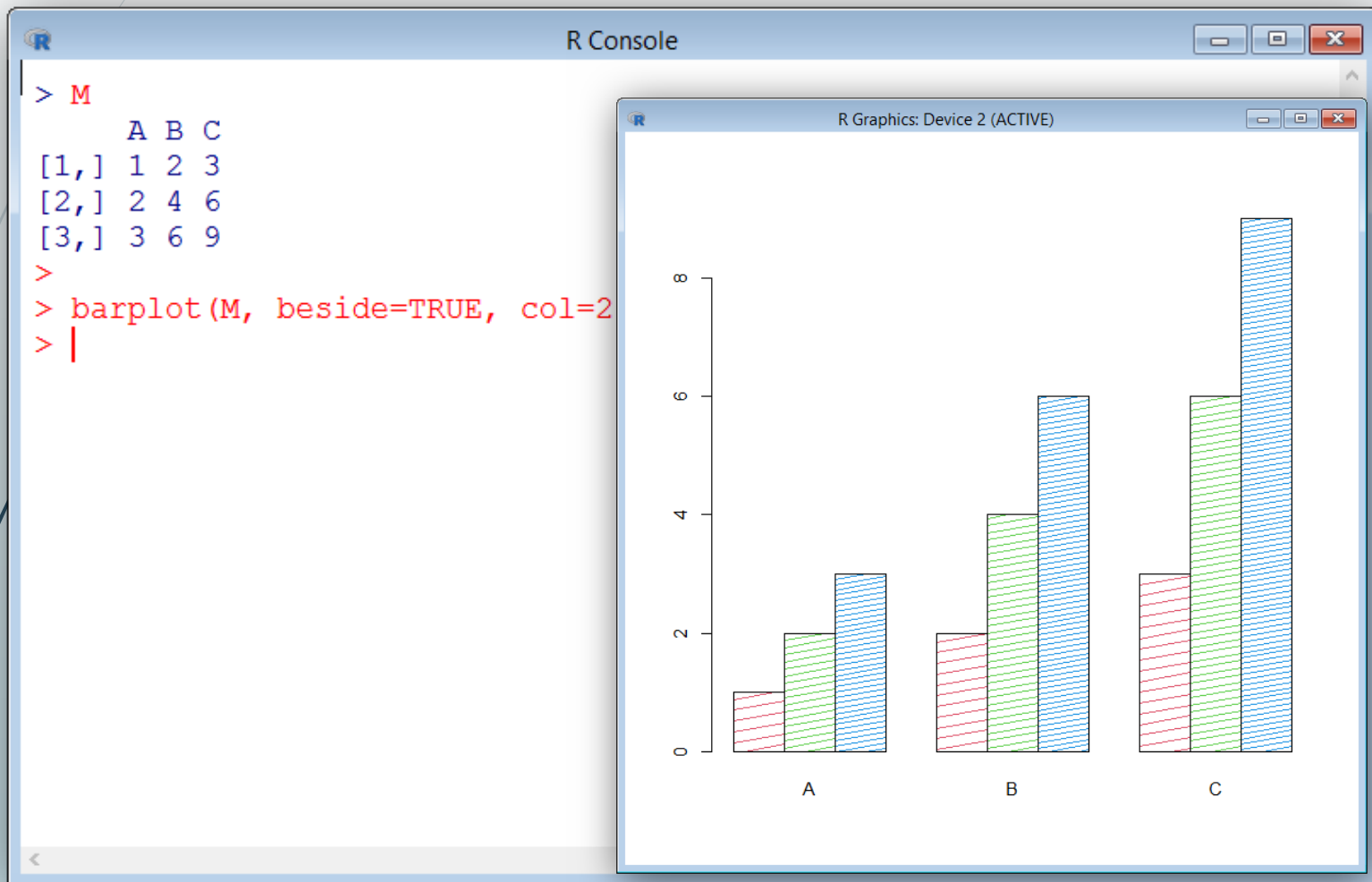
```
> DF
  A B C
1 1 2 3
2 2 4 6
3 3 6 9
> M = as.matrix(DF)
>
> barplot(M, width = 1, space =
> |
```



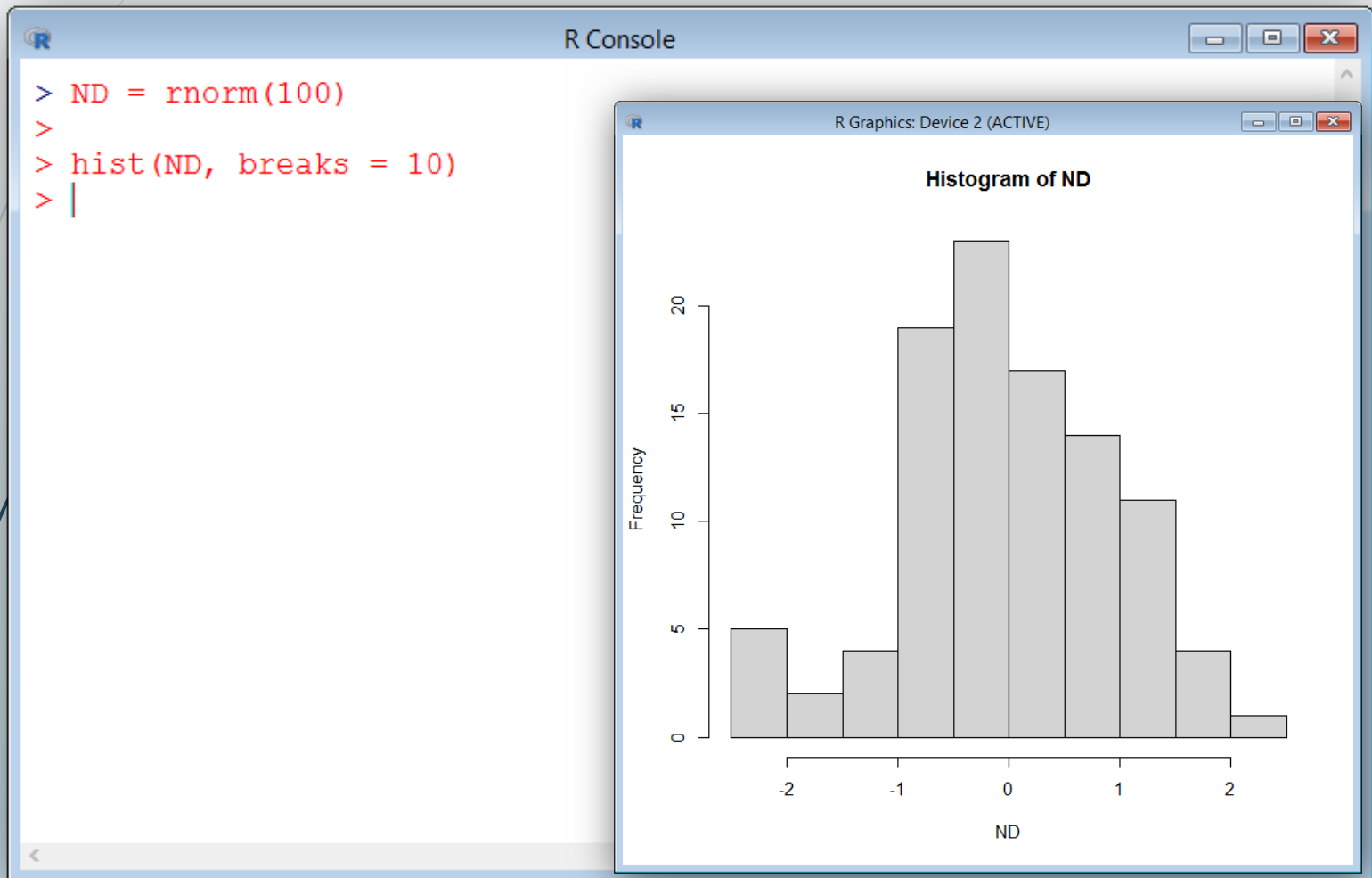
barplot()



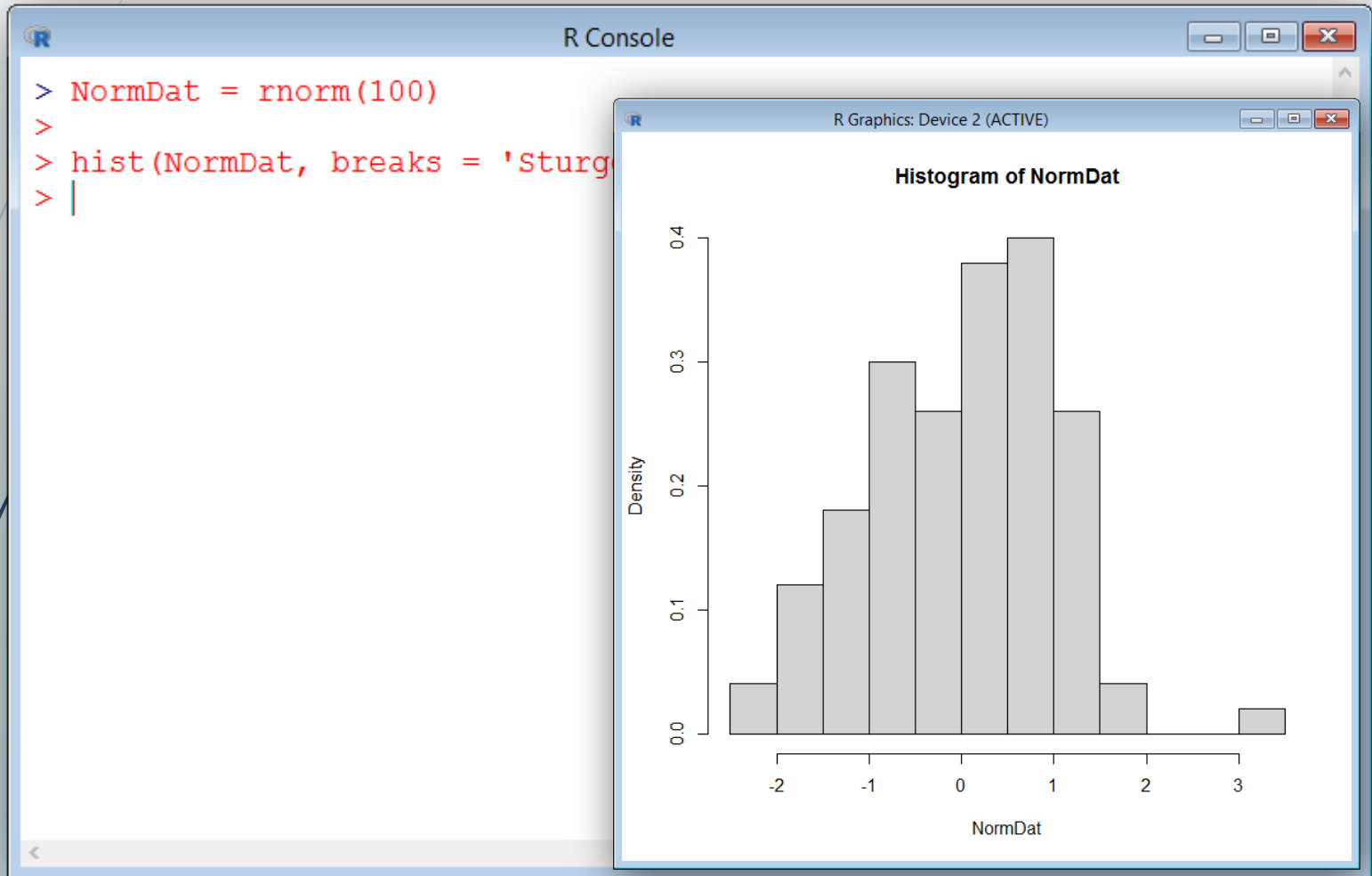
barplot()



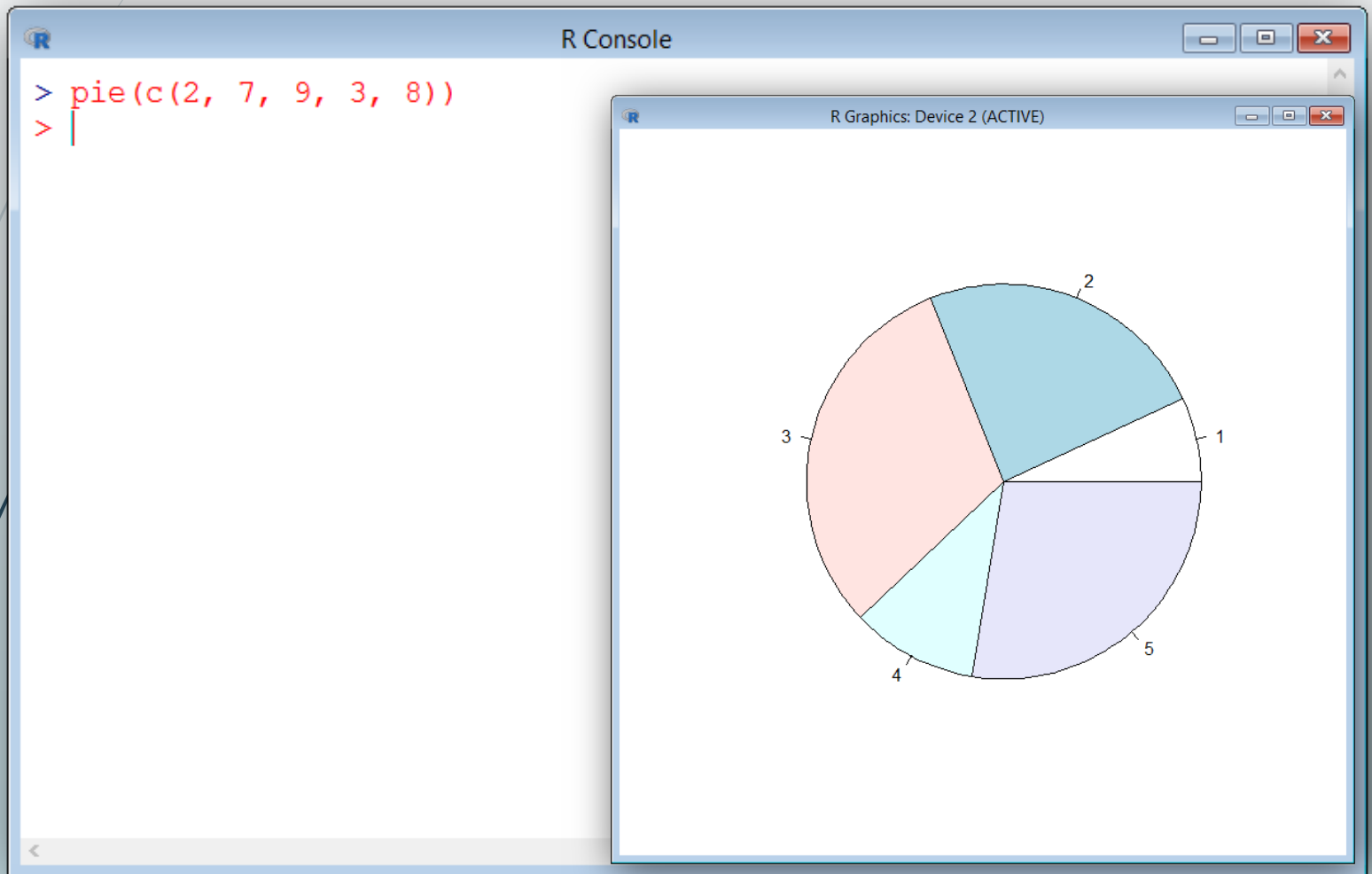
hist()



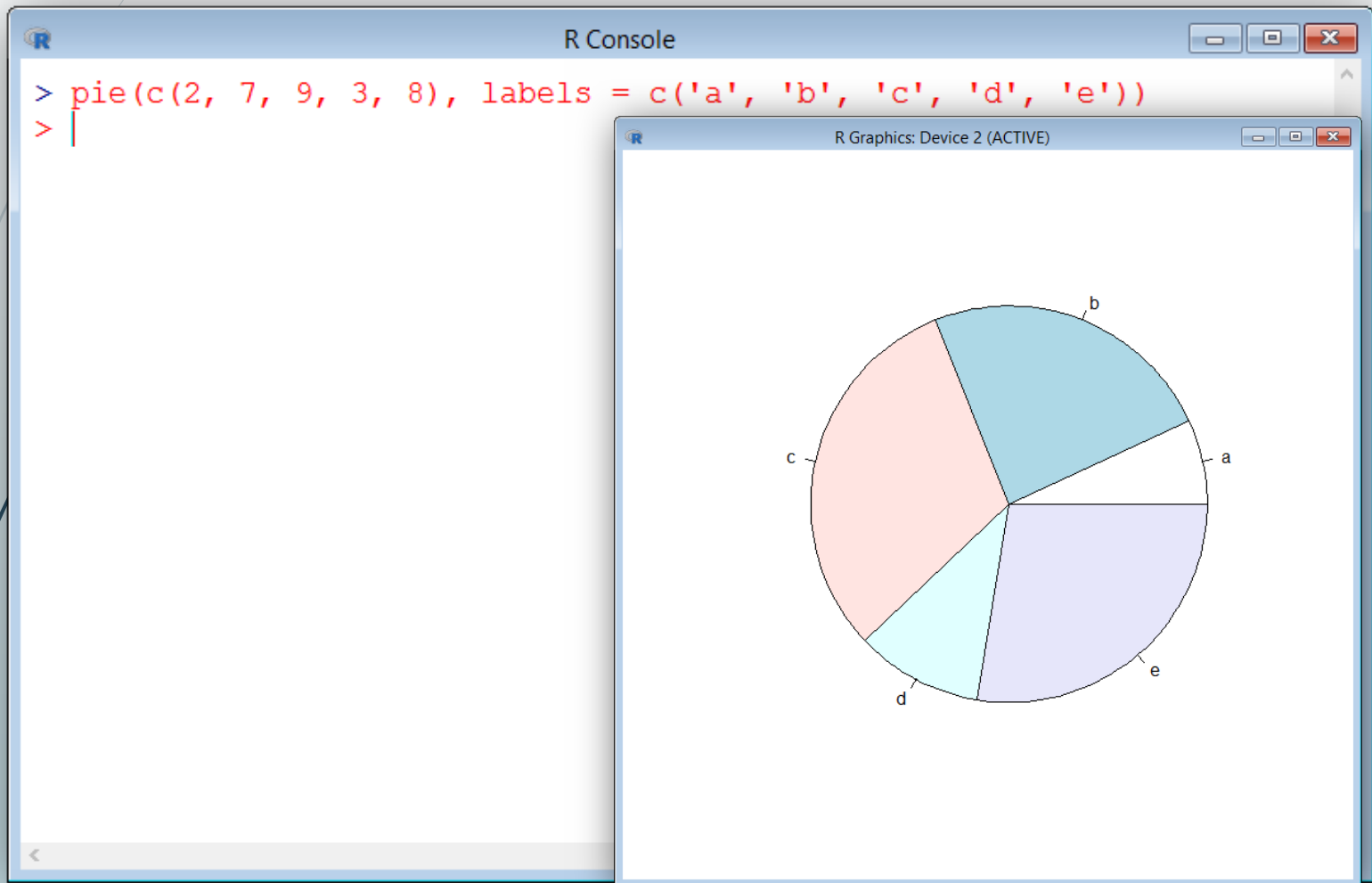
hist()



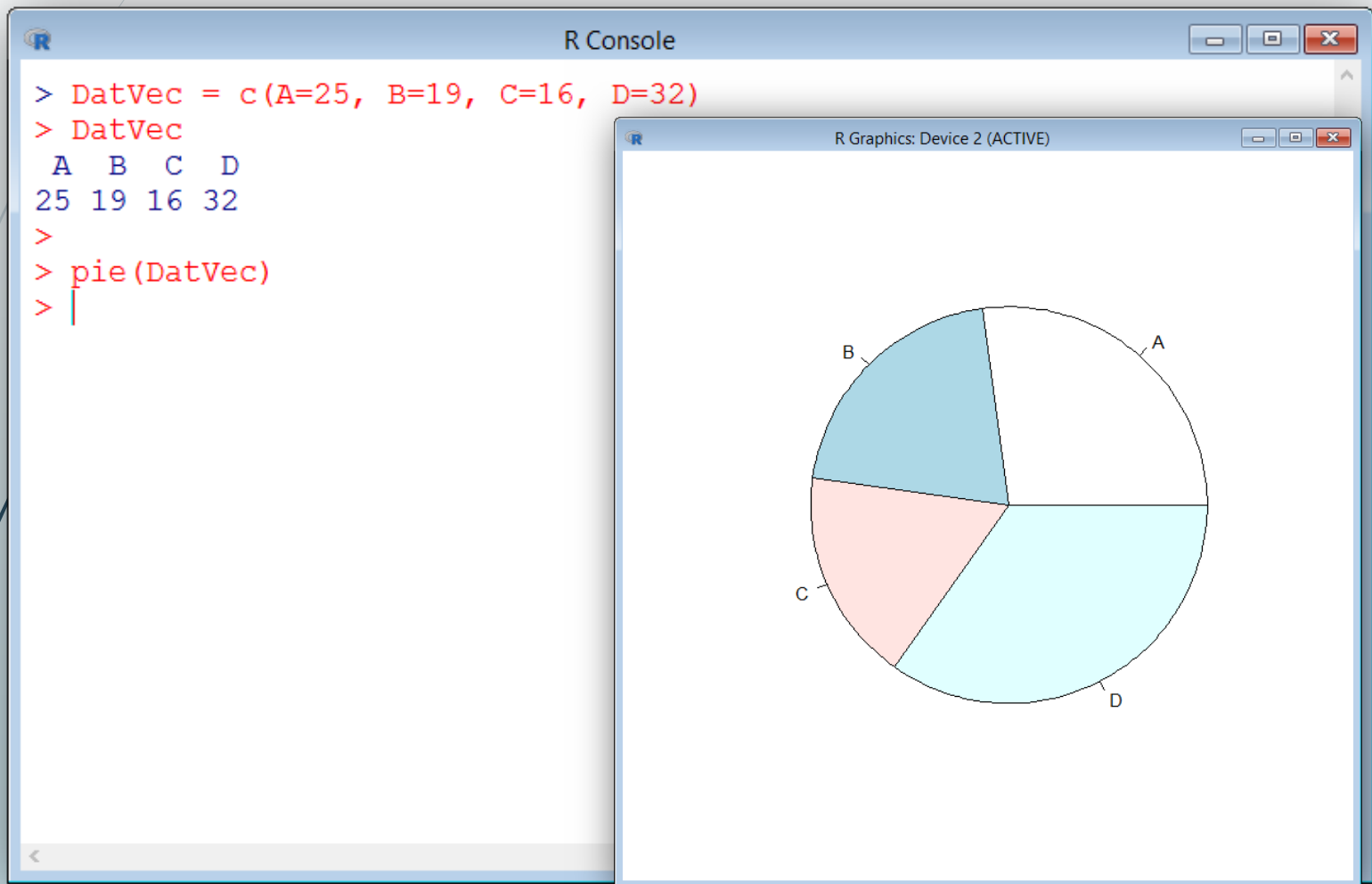
pie()



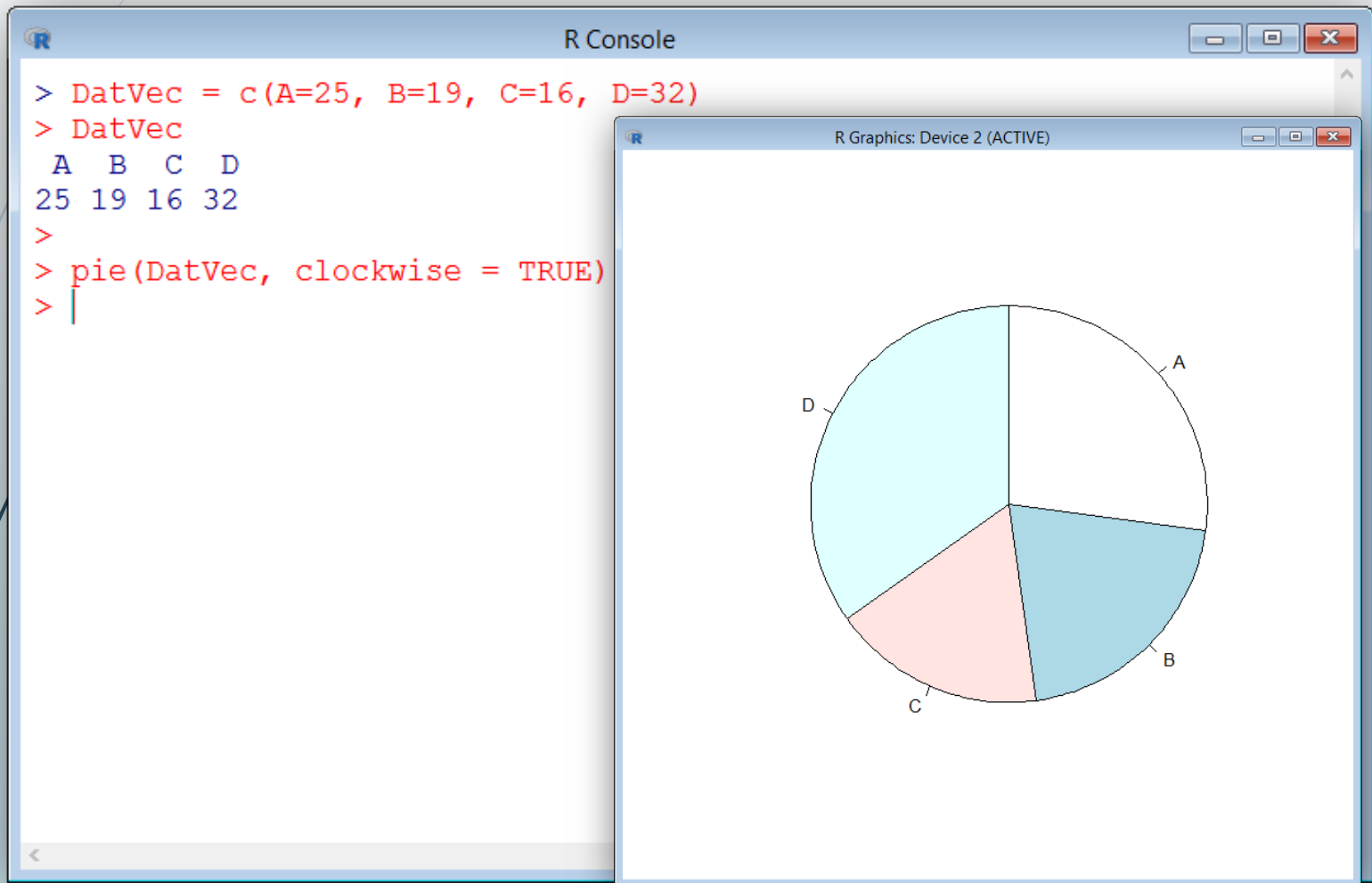
pie()



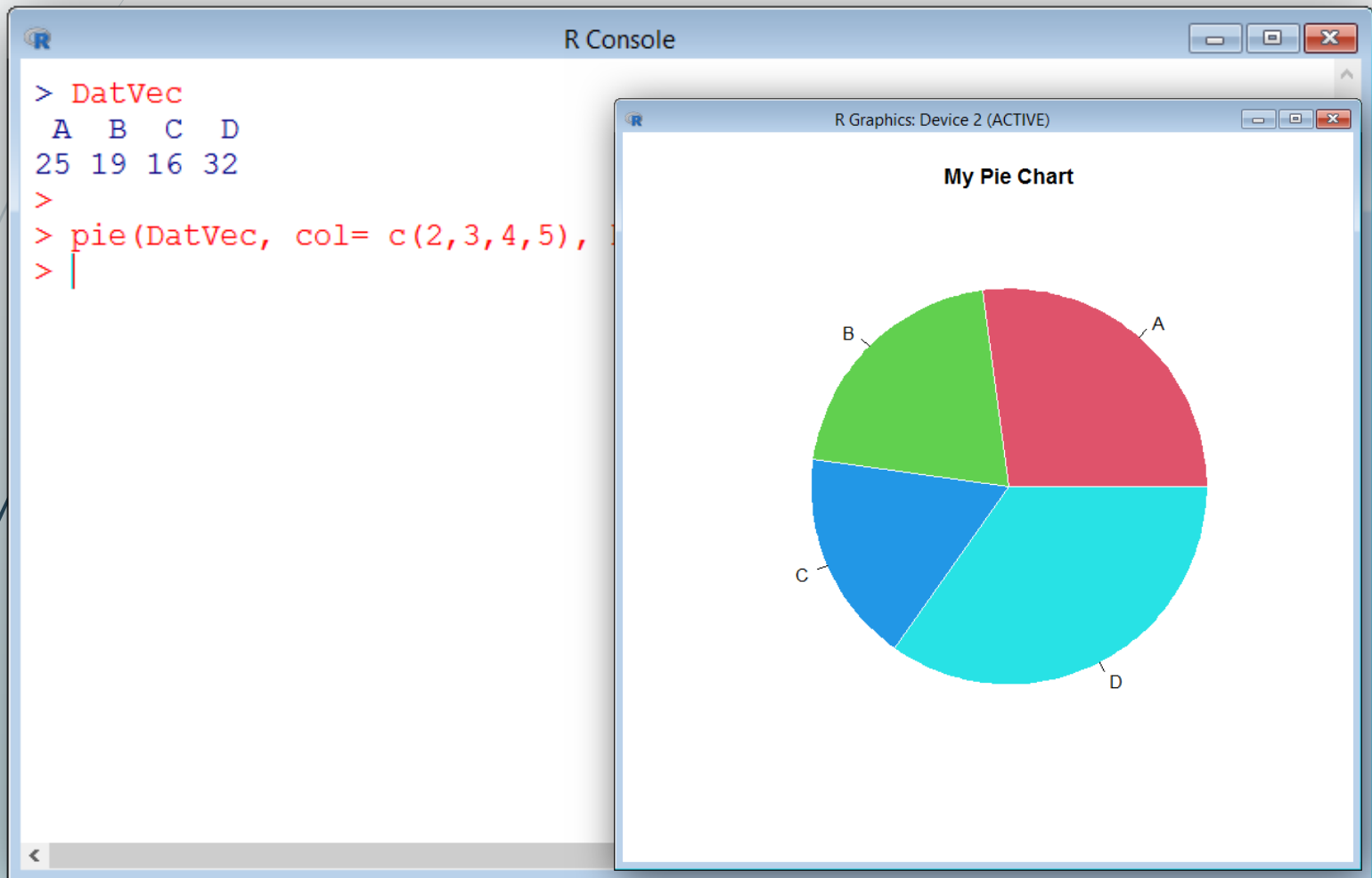
pie()



pie()



pie()



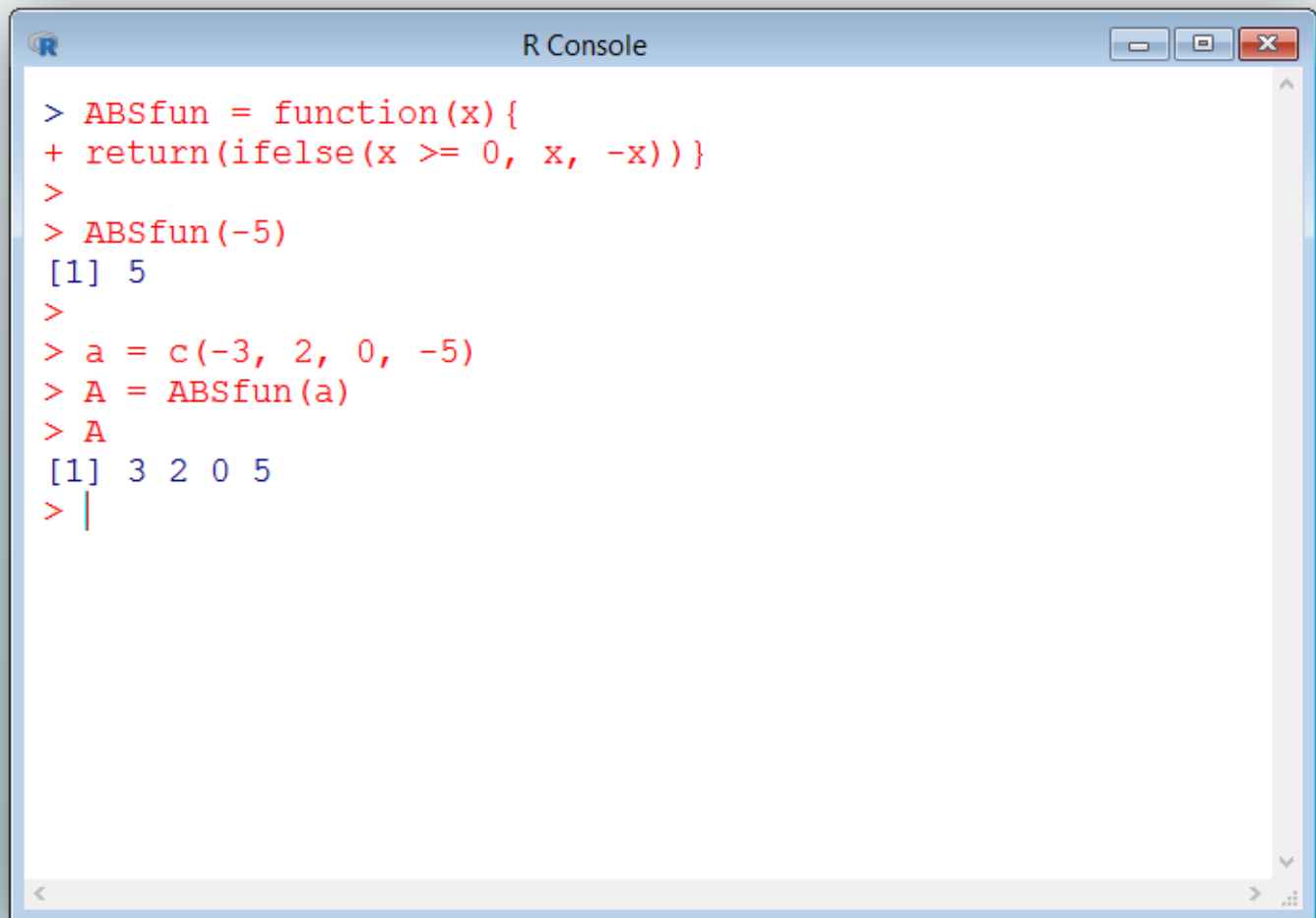


User-Defined Functions

Functional Programming

function() {}

► Example: absolute value



```
R Console
> ABSfun = function(x){
+ return(ifelse(x >= 0, x, -x))}
>
> ABSfun(-5)
[1] 5
>
> a = c(-3, 2, 0, -5)
> A = ABSfun(a)
> A
[1] 3 2 0 5
> |
```

function() {}

► Example: sign function

```
R Console
> SGN = function(x) {
+ if(x < 0){y = -1}
+ if(x == 0){y = 0}
+ if(x > 0){y = 1}
+ return(y)}
>
> SGN(32)
[1] 1
>
> V = seq(-10, 10, 2)
> V
[1] -10 -8 -6 -4 -2 0 2 4 6 8 10
>
> V_sgn = SGN(V)
Warning messages:
1: In if (x < 0) { :
  the condition has length > 1 and only the first element will be used
2: In if (x == 0) { :
  the condition has length > 1 and only the first element will be used
3: In if (x > 0) { :
  the condition has length > 1 and only the first element will be used
> V_sgn
[1] -1
> |
```

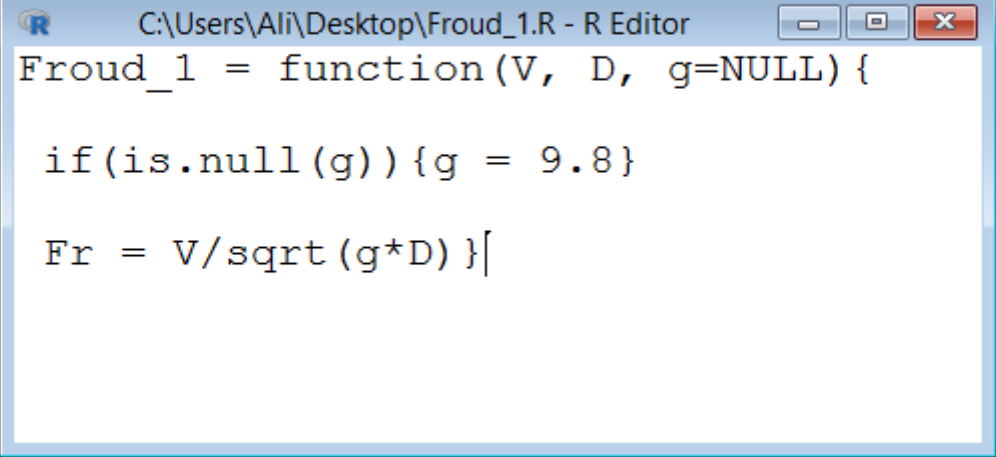
function() {}

► Example: sign function

```
R Console
> SGN = function(x){
+ ifelse(x > 0, 1, ifelse(x < 0, -1, 0))}
>
> SGN(0)
[1] 0
> SGN(-5)
[1] -1
> SGN(5)
[1] 1
>
> V = seq(-10, 10, 2)
> V
[1] -10 -8 -6 -4 -2 0 2 4 6 8 10
>
> V_sgn = SGN(V)
> V_sgn
[1] -1 -1 -1 -1 -1 0 1 1 1 1 1
> |
```

function() {}

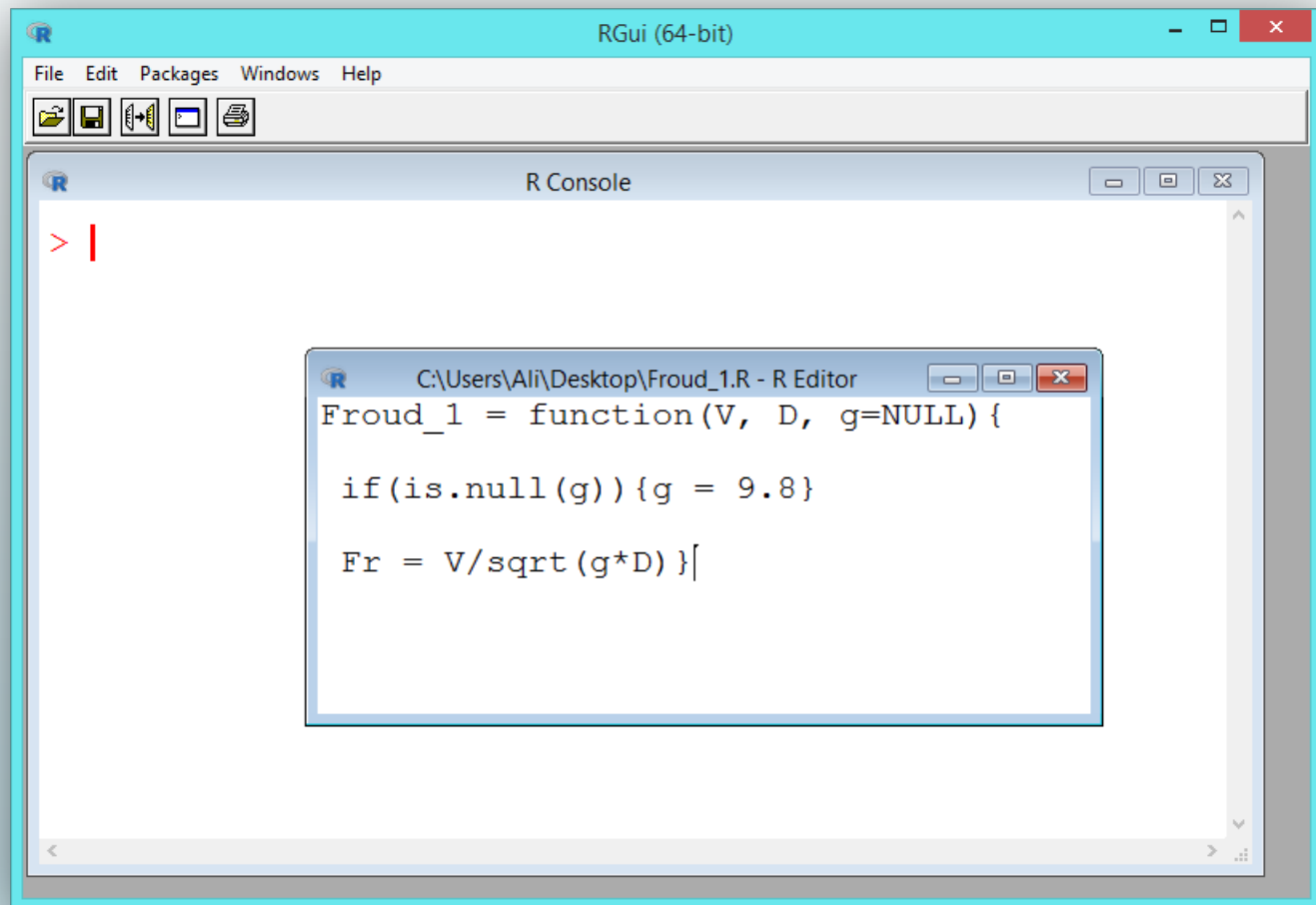
► Example: Froud Number



```
Froud_1 = function(V, D, g=NULL) {  
  if(is.null(g)) {g = 9.8}  
  Fr = V/sqrt(g*D) }|
```

function() {}

► Example: Froud Number



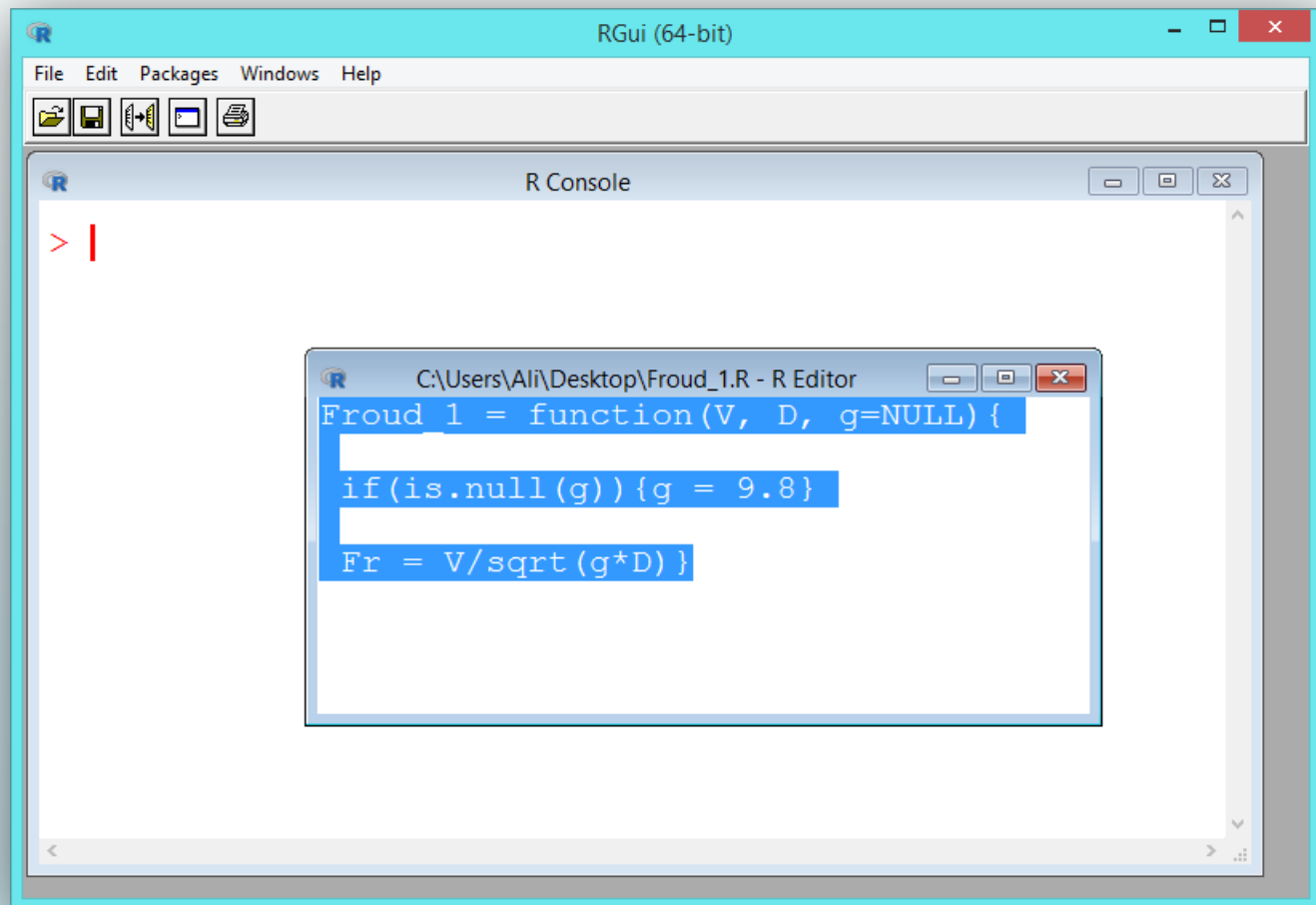
```
File Edit Packages Windows Help
[Icons]

R Console
> |

C:\Users\Ali\Desktop\Froud_1.R - R Editor
Froud_1 = function(V, D, g=NULL) {
  if(is.null(g)) {g = 9.8}
  Fr = V/sqrt(g*D) }|
```

function() {}

► Example: Froud Number

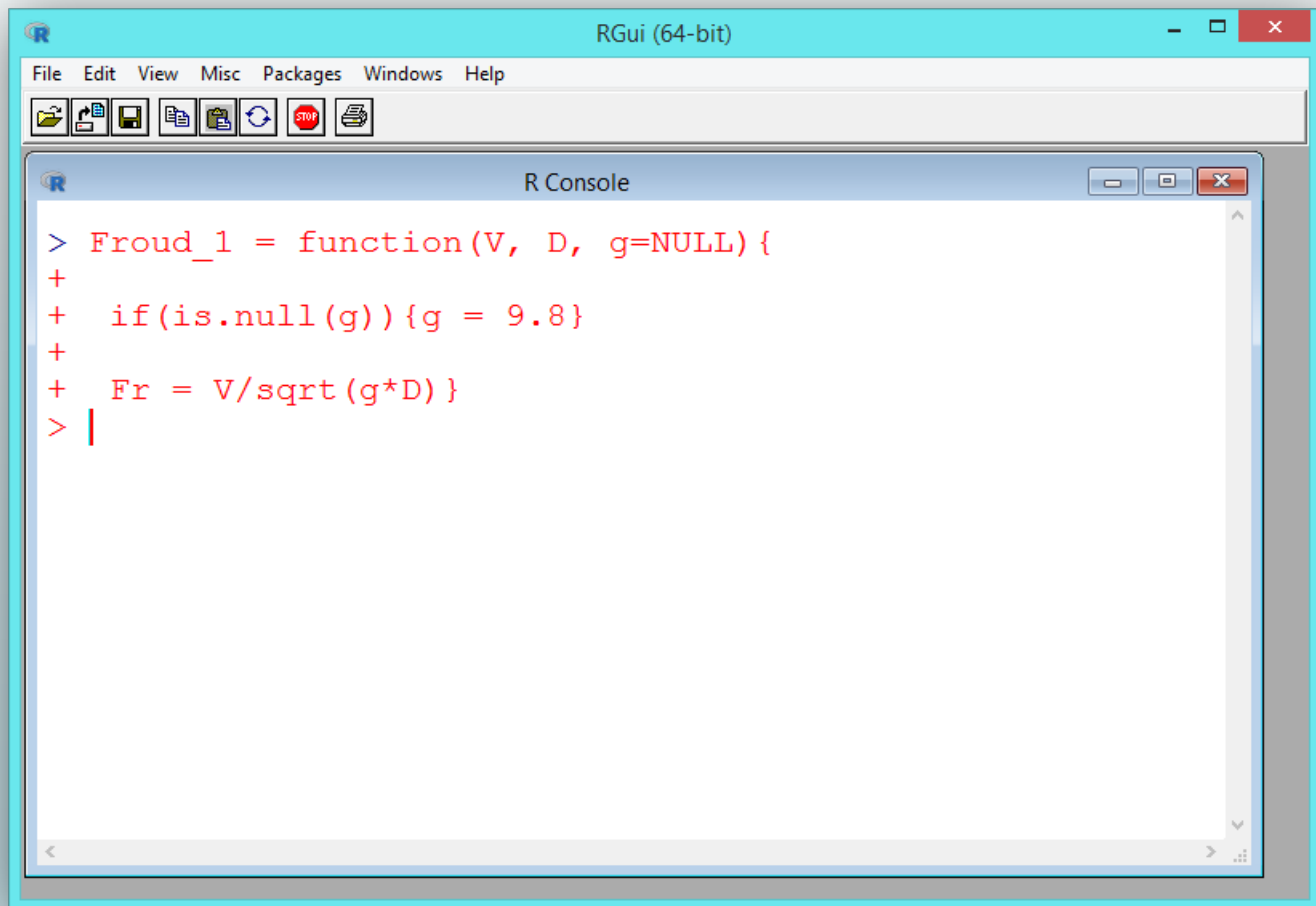


```
> |
```

```
C:\Users\Ali\Desktop\Froud_1.R - R Editor  
Froud_1 = function(V, D, g=NULL) {  
  if(is.null(g)) {g = 9.8}  
  Fr = V/sqrt(g*D) }
```

```
function() {}
```

► Example: Froud Number

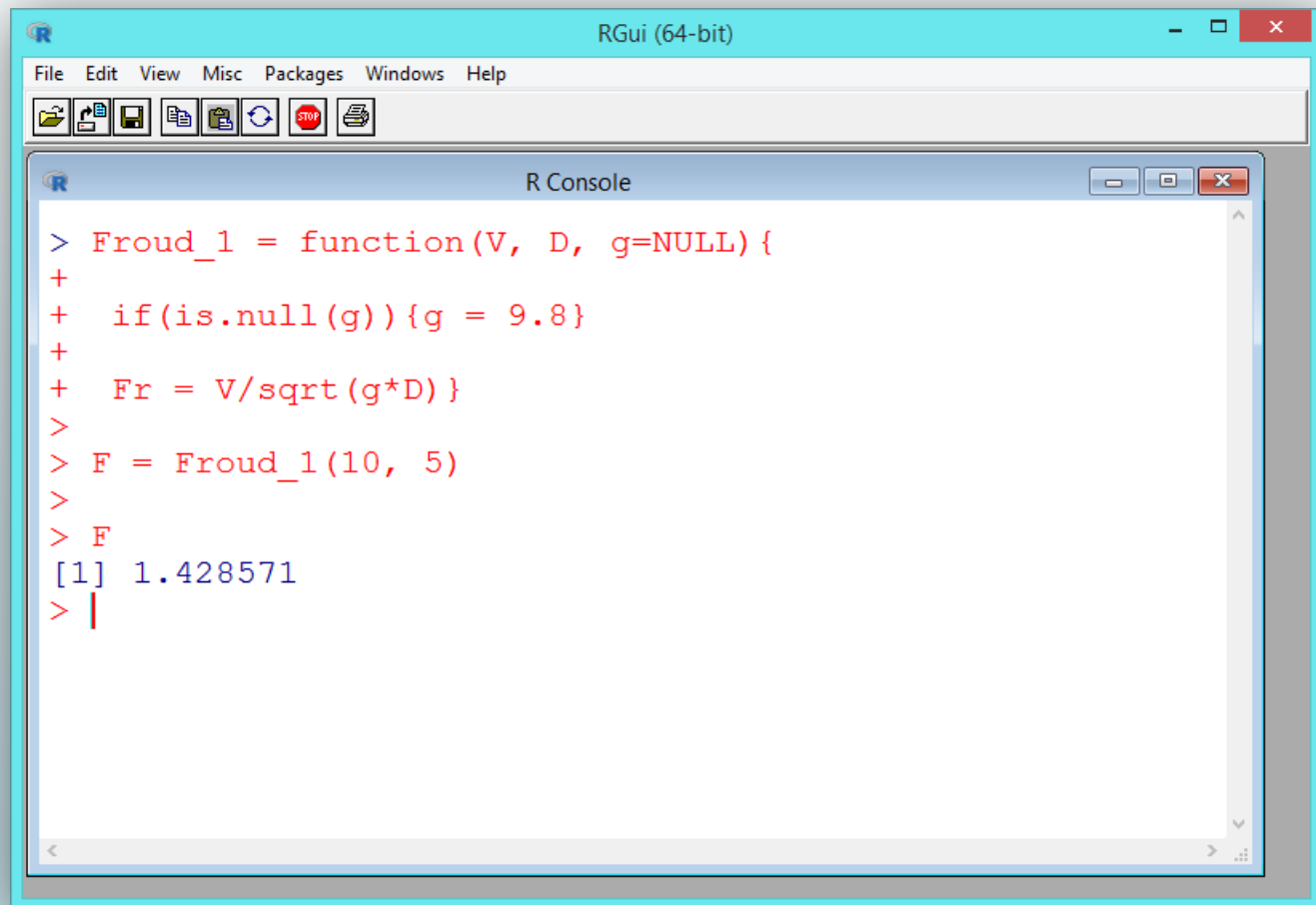


```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> Froud_1 = function(V, D, g=NULL){
+   if(is.null(g)){g = 9.8}
+   Fr = V/sqrt(g*D)}
> |
```


function() {}

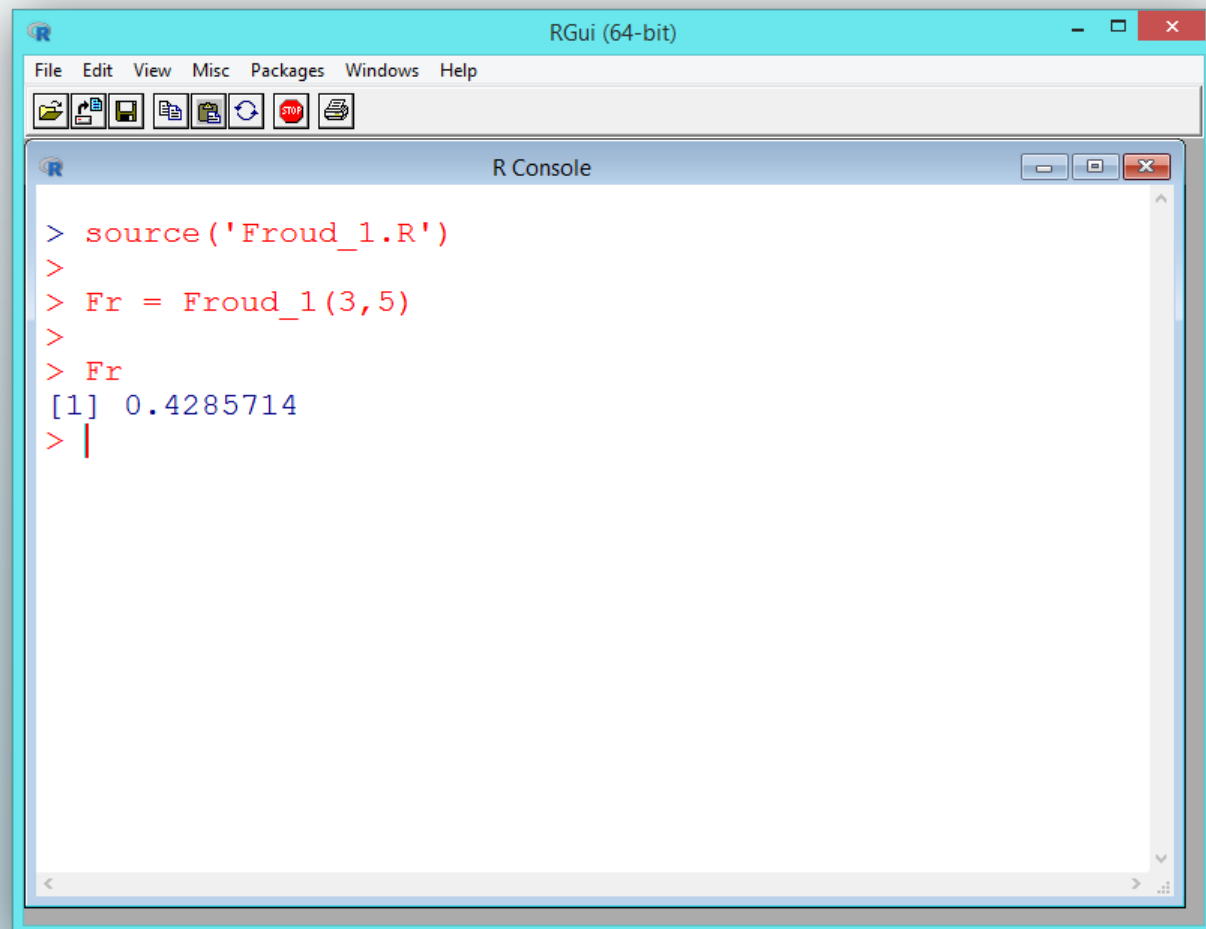
► Example: Froud Number



```
> Froud_1 = function(V, D, g=NULL){
+   if(is.null(g)){g = 9.8}
+   Fr = V/sqrt(g*D)}
>
> F = Froud_1(10, 5)
>
> F
[1] 1.428571
> |
```

function() {} and source()

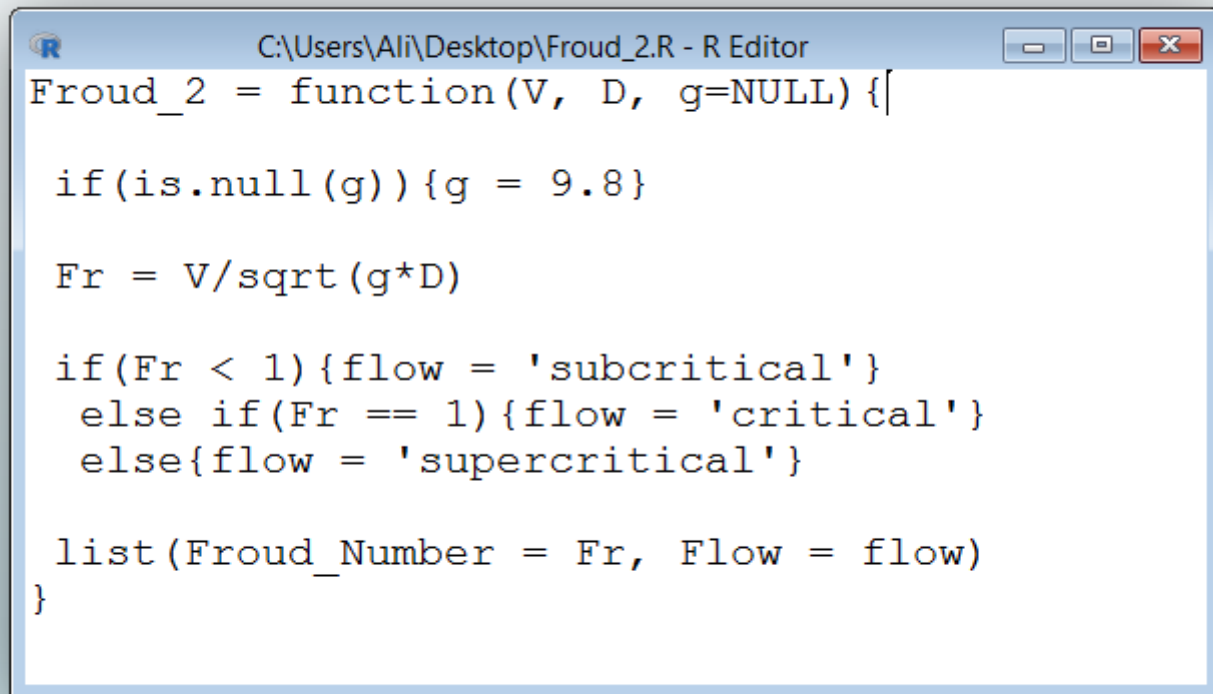
► Example: Froud Number



```
> source('Froud_1.R')
>
> Fr = Froud_1(3,5)
>
> Fr
[1] 0.4285714
> |
```

function() {} and source()

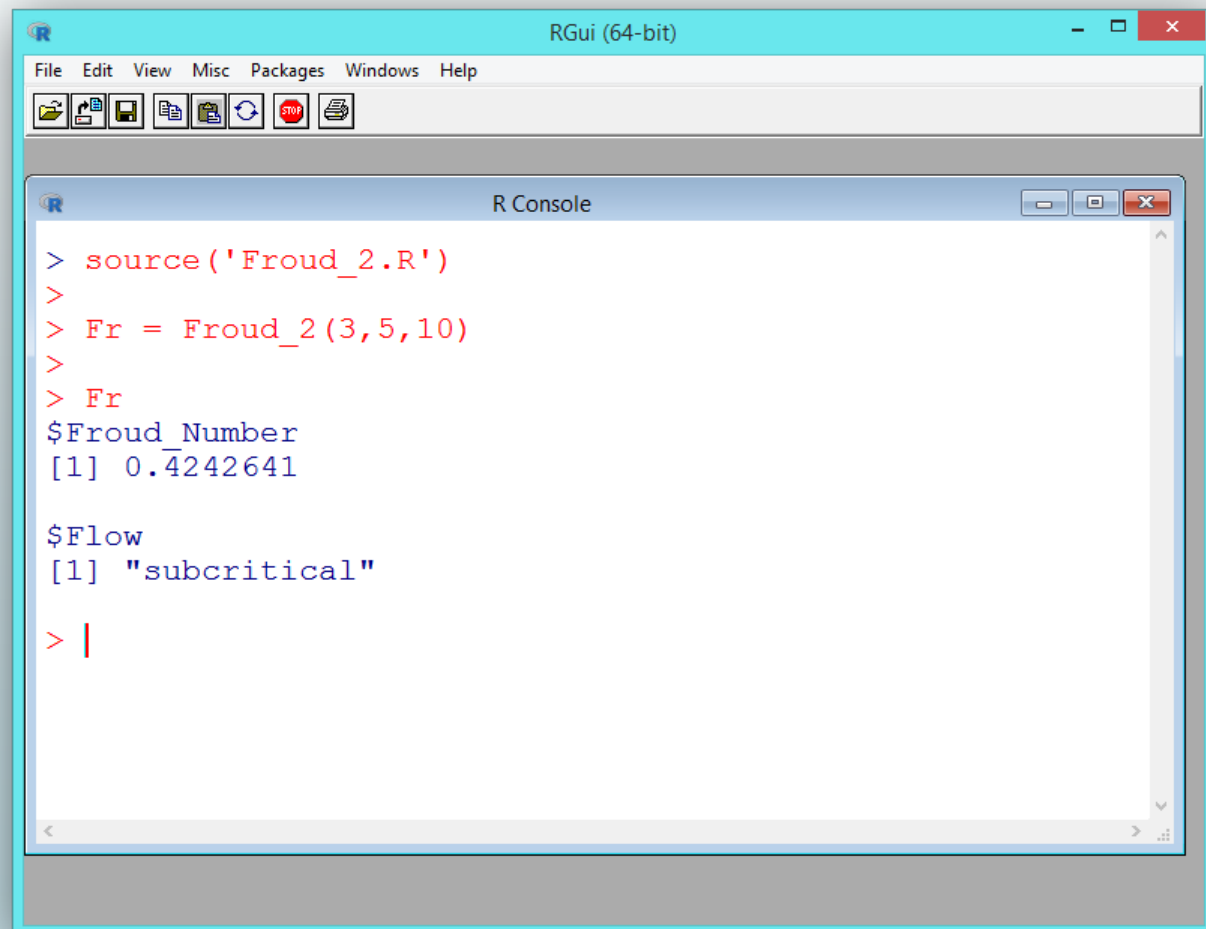
► Example: Froud Number



```
C:\Users\Ali\Desktop\Froud_2.R - R Editor
Froud_2 = function(V, D, g=NULL) {
  if(is.null(g)) {g = 9.8}
  Fr = V/sqrt(g*D)
  if(Fr < 1) {flow = 'subcritical'}
  else if(Fr == 1) {flow = 'critical'}
  else {flow = 'supercritical'}
  list(Froud_Number = Fr, Flow = flow)
}
```

function() {} and source()

► Example: Froud Number



```
> source('Froud_2.R')
>
> Fr = Froud_2(3,5,10)
>
> Fr
$Froud_Number
[1] 0.4242641

$Flow
[1] "subcritical"

> |
```

function() {} and source()

► Example: Free Fall

```

C:\Users\Ali\Desktop\FreeFall.R - R Editor
FreeFall_1 = function(y_0, v_0=NULL, g=NULL, y=NULL){

  if(is.null(v_0)){v_0 = 0}
  if(is.null(g)){g = -9.8}
  if(is.null(y)){y = 0}

  dy = y-y_0

  delta = v_0^2 - 4*0.5*g*(-dy)
  t_1 = (-v_0 + sqrt(delta)) / (2*0.5*g)
  t_2 = (-v_0 - sqrt(delta)) / (2*0.5*g)
  t = max(t_1, t_2)

  v_1 = c()
  y_1 = c()
  for(i in 1:t){
    v_1[i] = g*i + v_0
    y_1[i] = 0.5*g*i^2 + v_0*i + y_0}

  data.frame(time = 1:t, height = y_1, velocity = v_1)}

#####
FreeFall_2 = function(y_0, v_0=NULL, g=NULL, y=NULL){

  if(is.null(v_0)){v_0 = 0}
  if(is.null(g)){g = -9.8}
  if(is.null(y)){y = 0}

  dy = y-y_0

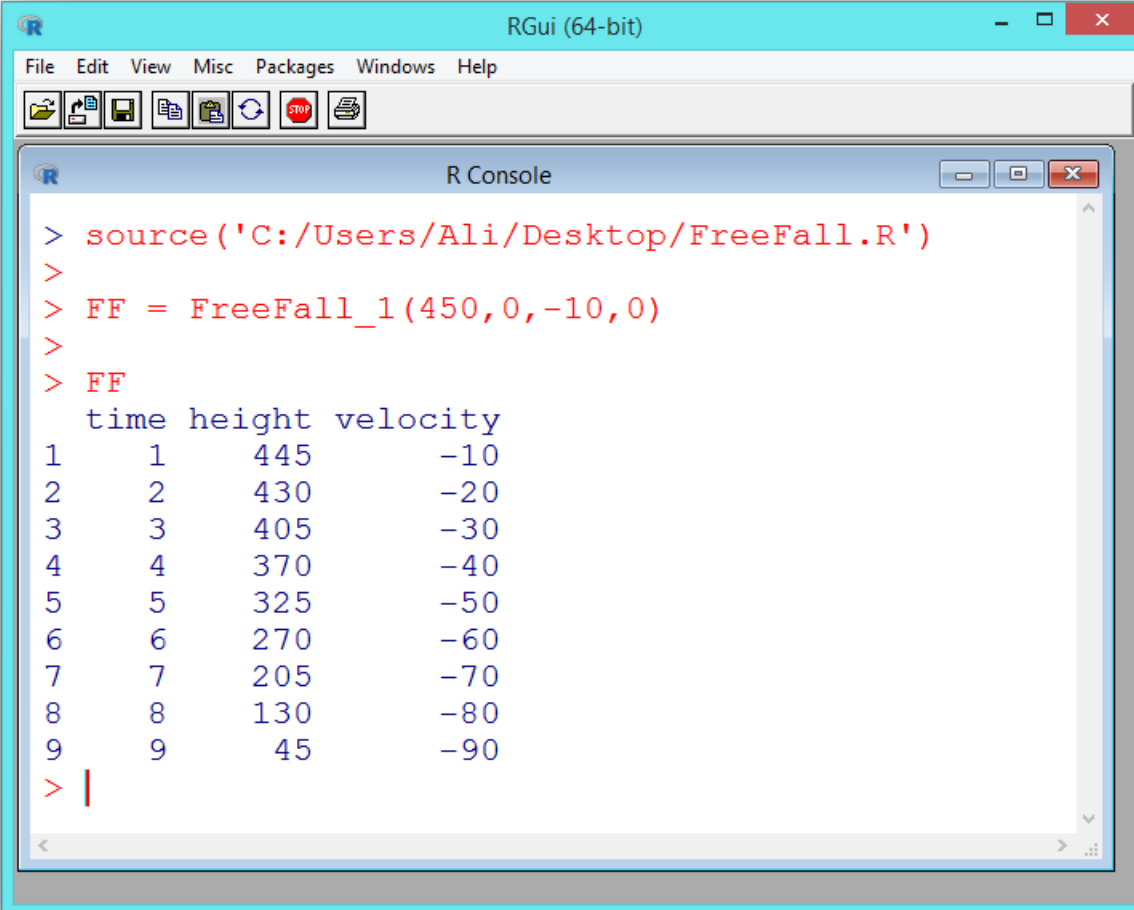
  y_2 = seq(y_0, y, -10)
  v_2 = -sqrt(2*g*(y_2-y_0) + v_0^2)[
  T = (v_2 - v_0)/g

  data.frame(Height = y_2, Velocity = v_2, Time = T)}

```

function() {} and source()

► Example: Free Fall



```
> source('C:/Users/Ali/Desktop/FreeFall.R')
>
> FF = FreeFall_1(450,0,-10,0)
>
> FF
  time height velocity
1     1     445     -10
2     2     430     -20
3     3     405     -30
4     4     370     -40
5     5     325     -50
6     6     270     -60
7     7     205     -70
8     8     130     -80
9     9      45     -90
> |
```

apply()

```
R Console
> X = matrix(seq(2, 30, 2), nr = 5, byrow = TRUE)
>
> X
      [,1] [,2] [,3]
[1,]    2    4    6
[2,]    8   10   12
[3,]   14   16   18
[4,]   20   22   24
[5,]   26   28   30
>
> SD_X_r = apply(X, MARGIN = 1, FUN = sd)
> SD_X_r
[1] 2 2 2 2 2
>
> SD_X_c = apply(X, MARGIN = 2, FUN = sd)
> SD_X_c
[1] 9.486833 9.486833 9.486833
> |
```

apply()

```
R Console
> M = matrix(seq(5, 45, 5), nc = 3)
> M
      [,1] [,2] [,3]
[1,]    5   20   35
[2,]   10   25   40
[3,]   15   30   45
> colMeans(M)
[1] 10 25 40
>
> CMSbtr = apply(M, MARGIN = 2, function(a){a - mean(a)})
> CMSbtr
      [,1] [,2] [,3]
[1,]   -5   -5   -5
[2,]    0    0    0
[3,]    5    5    5
> |
```

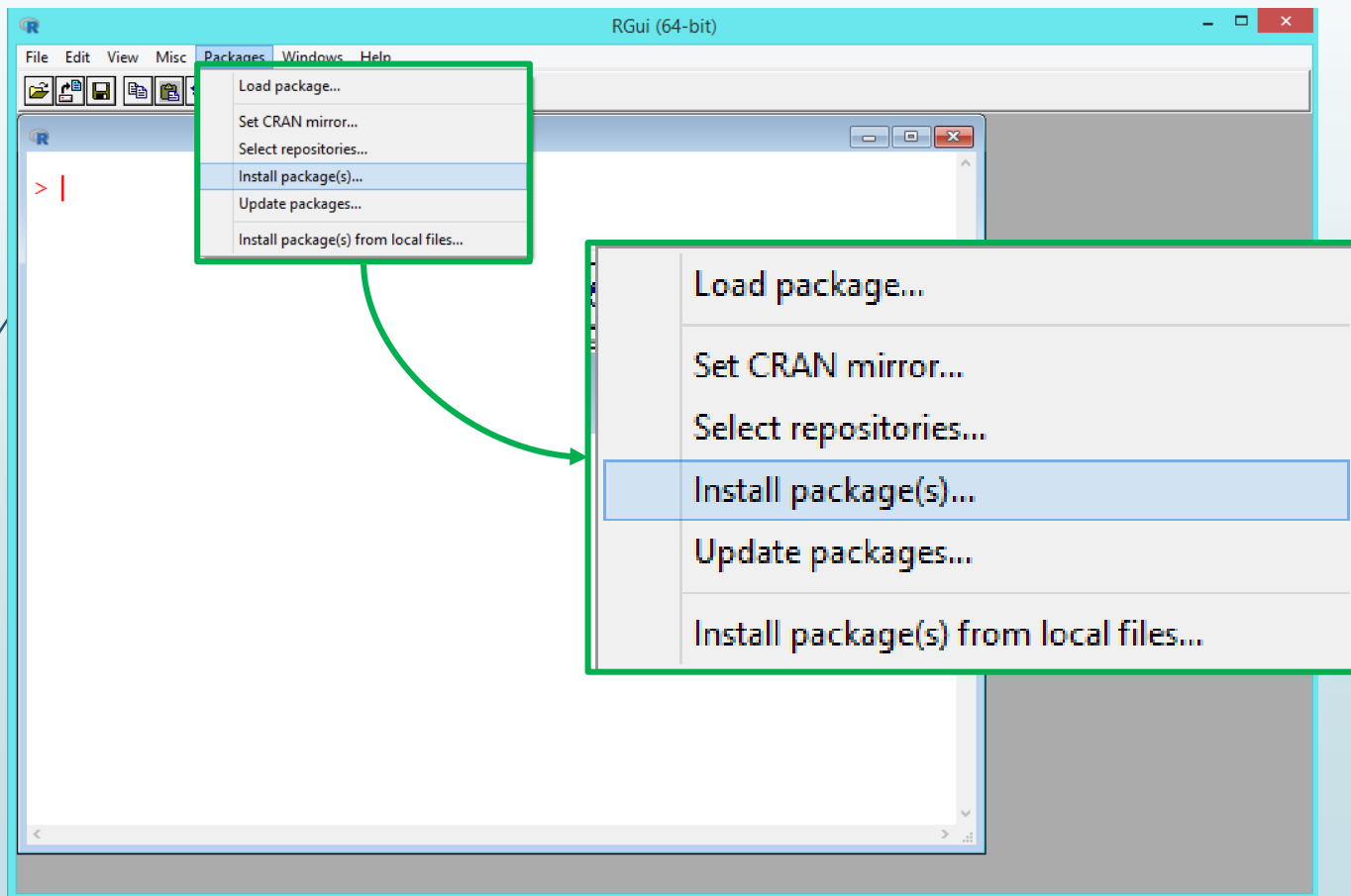



Packages

Downloading and Installing R Packages

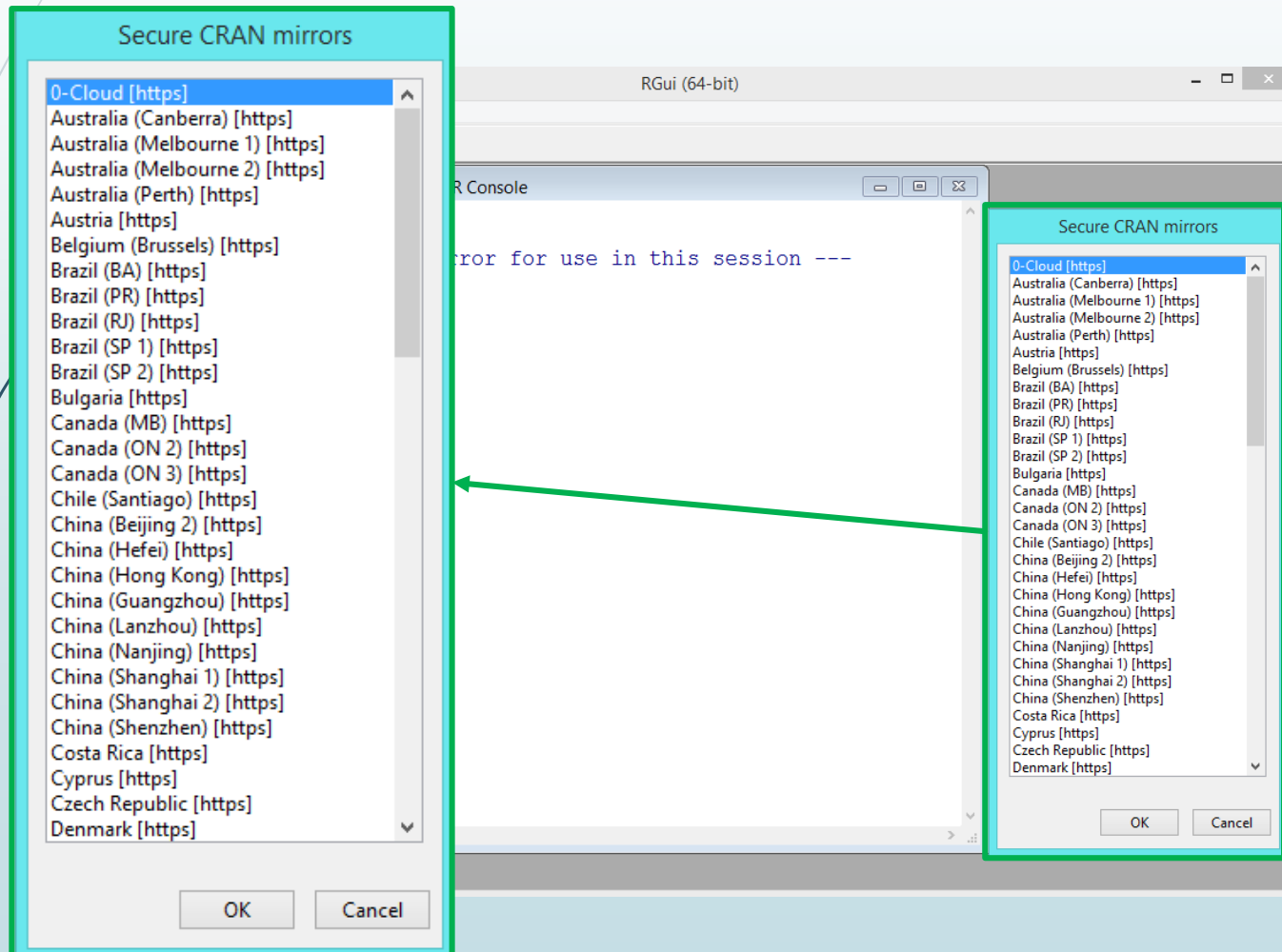
Downloading and Installing a Package

► First method (example 1)



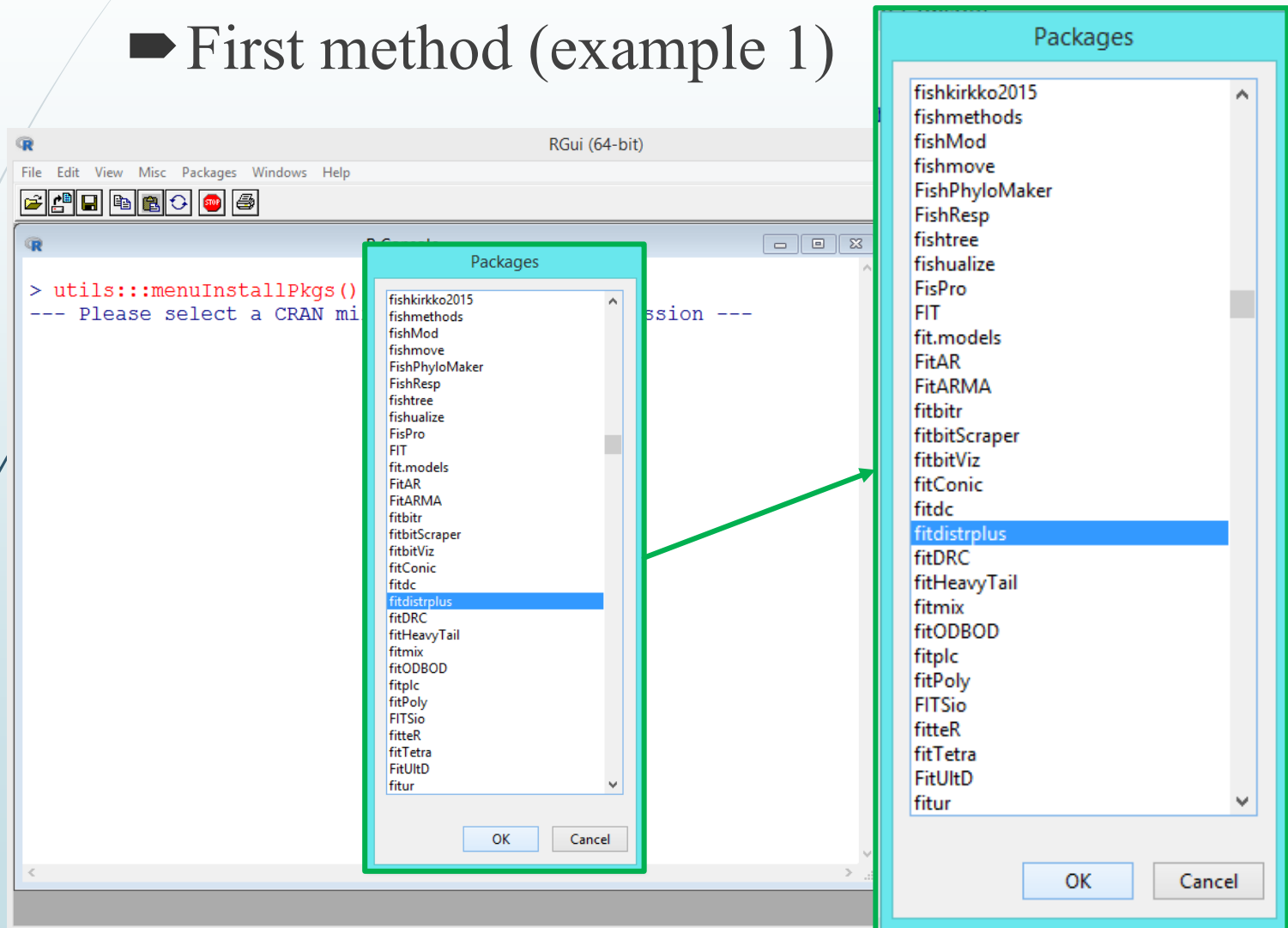
Downloading and Installing a Package

► First method (example 1)



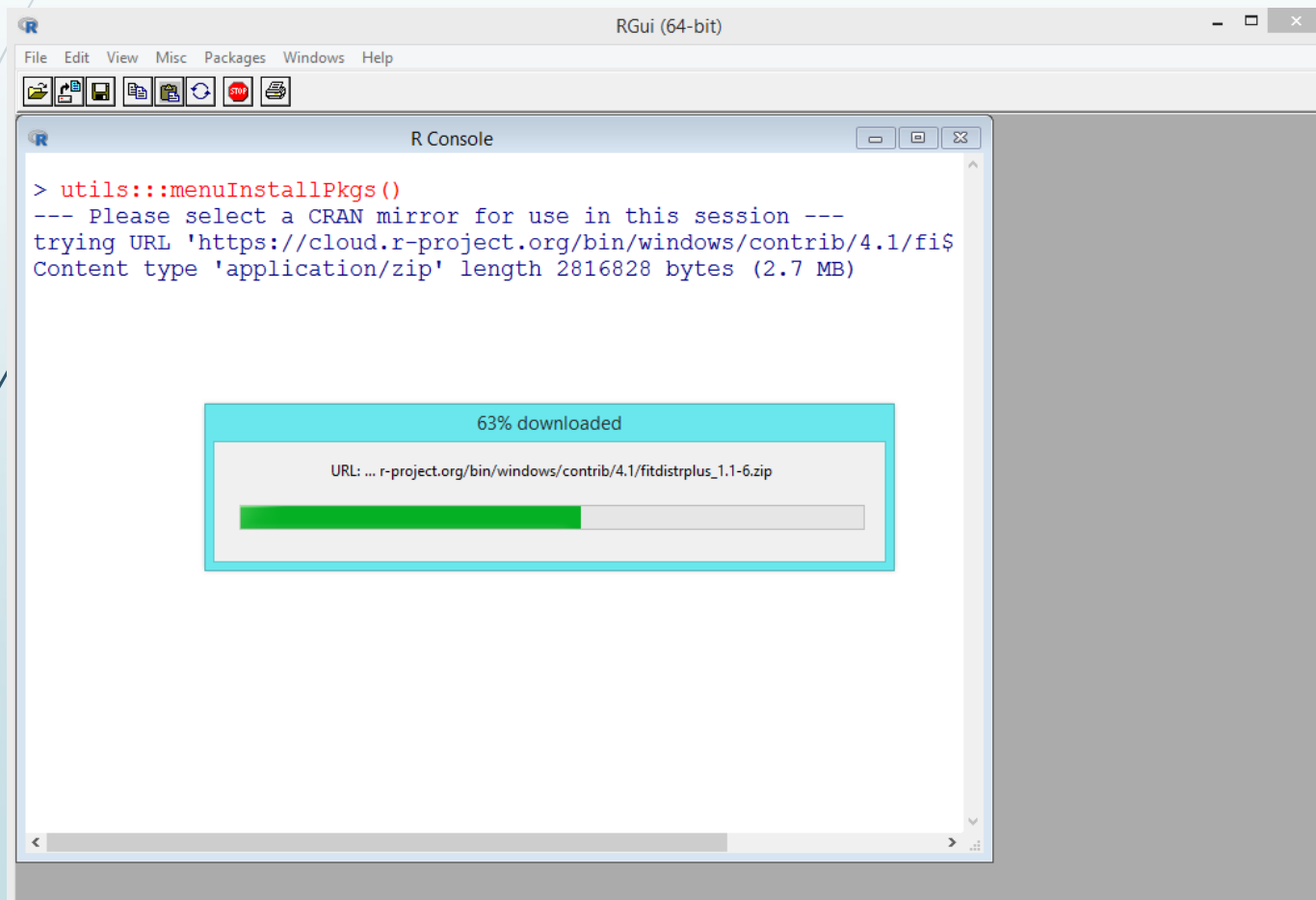
Downloading and Installing a Package

➤ First method (example 1)



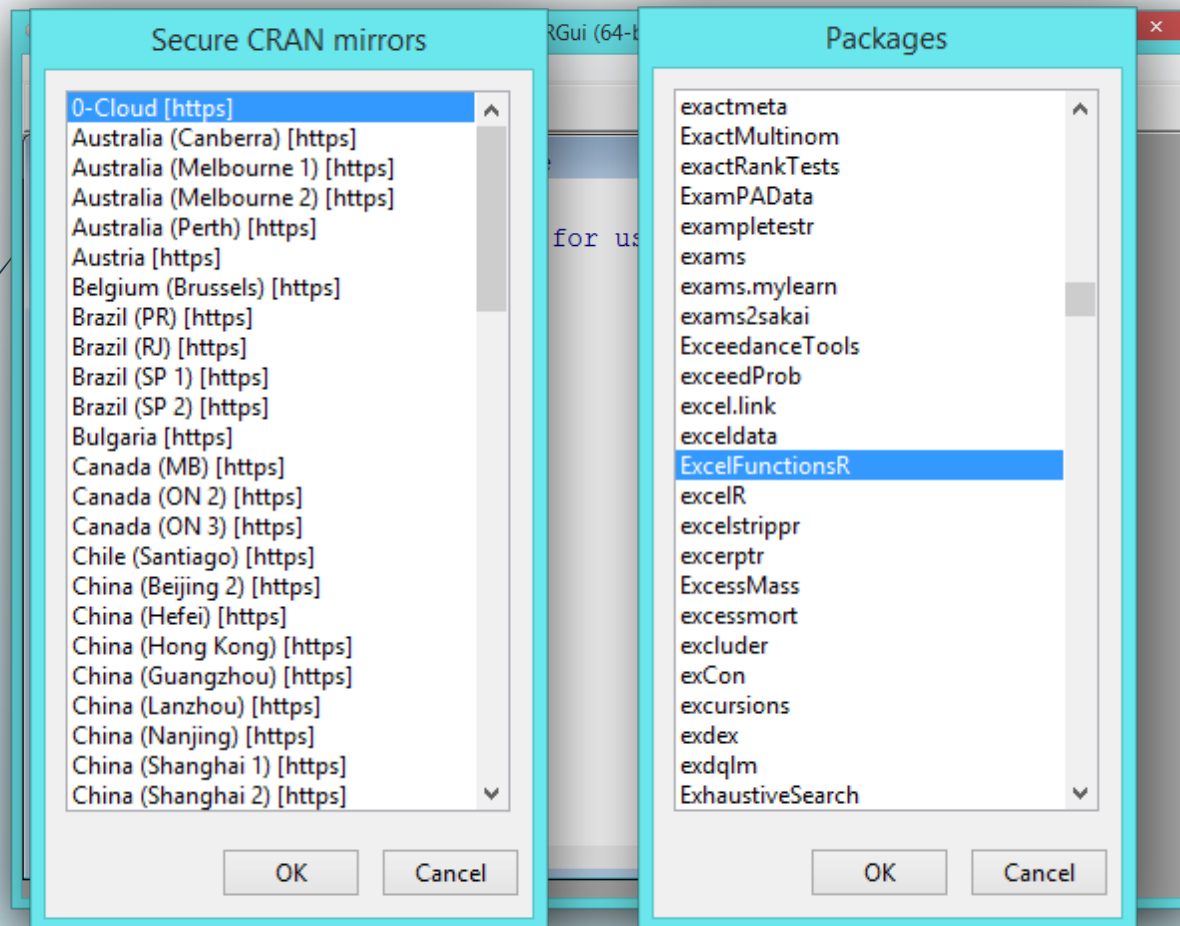
Downloading and Installing a Package

► First method (example 1)



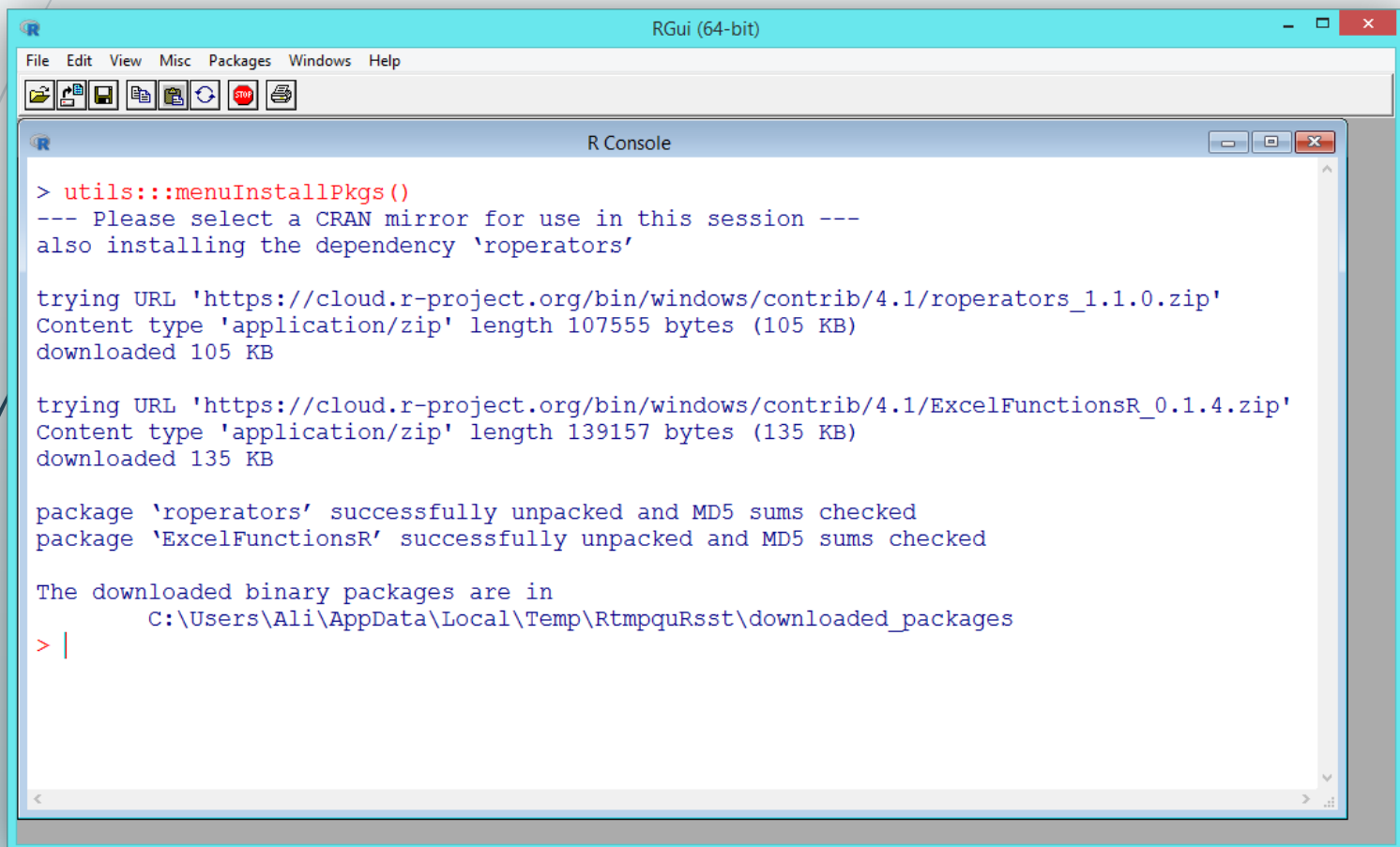
Downloading and Installing a Package

► First method (example 2)



Downloading and Installing a Package

► First method (example 2)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> utils:::menuInstallPkgs()
--- Please select a CRAN mirror for use in this session ---
also installing the dependency 'roperators'

trying URL 'https://cloud.r-project.org/bin/windows/contrib/4.1/roperators_1.1.0.zip'
Content type 'application/zip' length 107555 bytes (105 KB)
downloaded 105 KB

trying URL 'https://cloud.r-project.org/bin/windows/contrib/4.1/ExcelFunctionsR_0.1.4.zip'
Content type 'application/zip' length 139157 bytes (135 KB)
downloaded 135 KB

package 'roperators' successfully unpacked and MD5 sums checked
package 'ExcelFunctionsR' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\Ali\AppData\Local\Temp\RtmpqRsst\downloaded_packages
> |
```

Downloading and Installing a Package

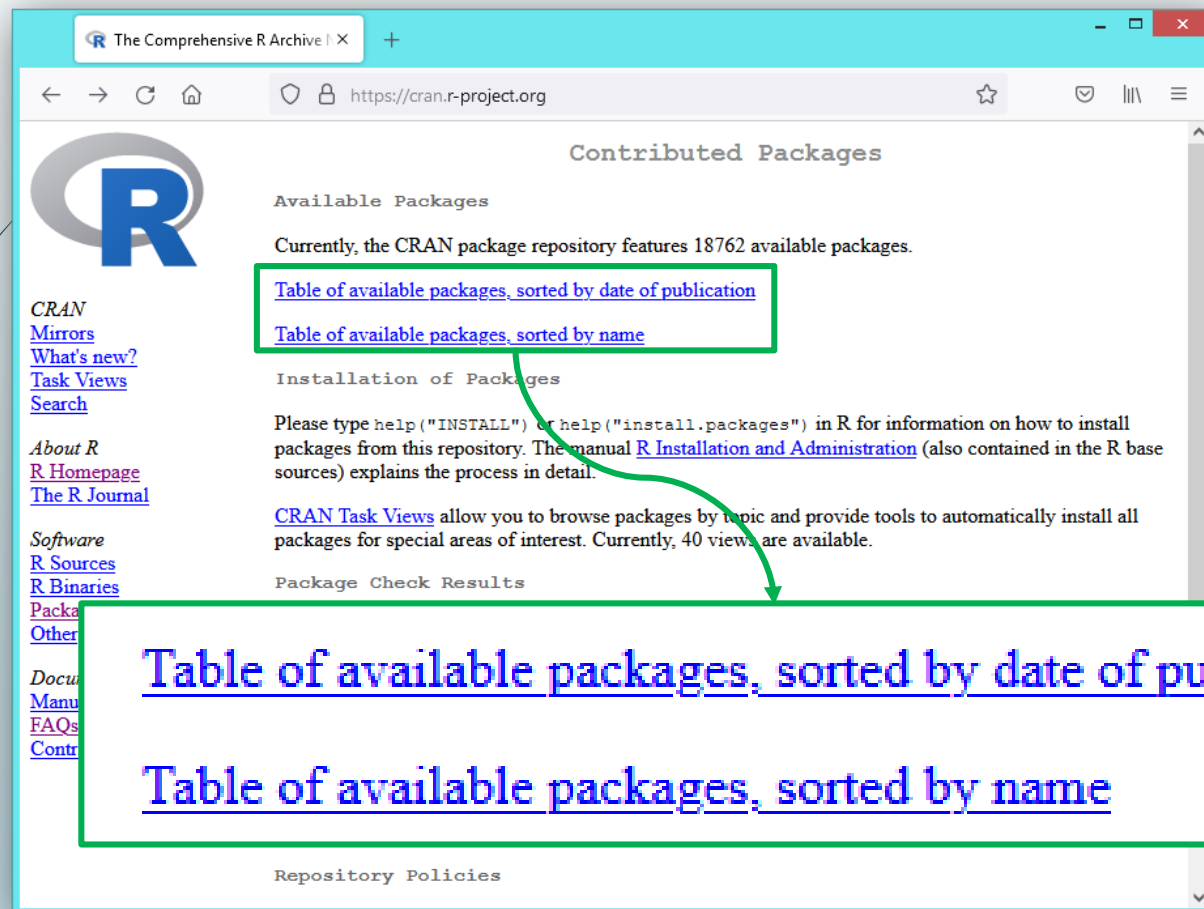
➤ Second method

The screenshot shows the CRAN website with the following elements:

- Address Bar:** `https://cran.r-project.org` (highlighted with a green box and a green arrow pointing to the main content area).
- Header:** The Comprehensive R Archive Network
- Left Sidebar:**
 - CRAN
 - [Mirrors](#)
 - [What's new?](#)
 - [Task Views](#)
 - [Search](#)
 - About R
 - [R Homepage](#)
 - [The R Journal](#)
 - Software
 - [R Sources](#)
 - [R Binaries](#)
 - [Packages](#)** (highlighted with a green box and a green arrow pointing to the main content area)
 - [Other](#)
 - Documentation
 - [Manuals](#)
 - [FAQs](#)
 - [Contributed](#)
- Main Content Area:**
 - Download and Install R**
 - Precompiled binary distributions of R are available for Windows and Mac users most likely:
 - [Download R for Linux \(Debian\)](#)
 - [Download R for macOS](#)
 - [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 (2021-11-01, Bird Hippie) [R-4.1.2.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 packages are available [here](#). Please read [about new packages](#) for feature requests or bug reports.
 - Source code of older versions of R
 - Contributed extension [packages](#)

Downloading and Installing a Package

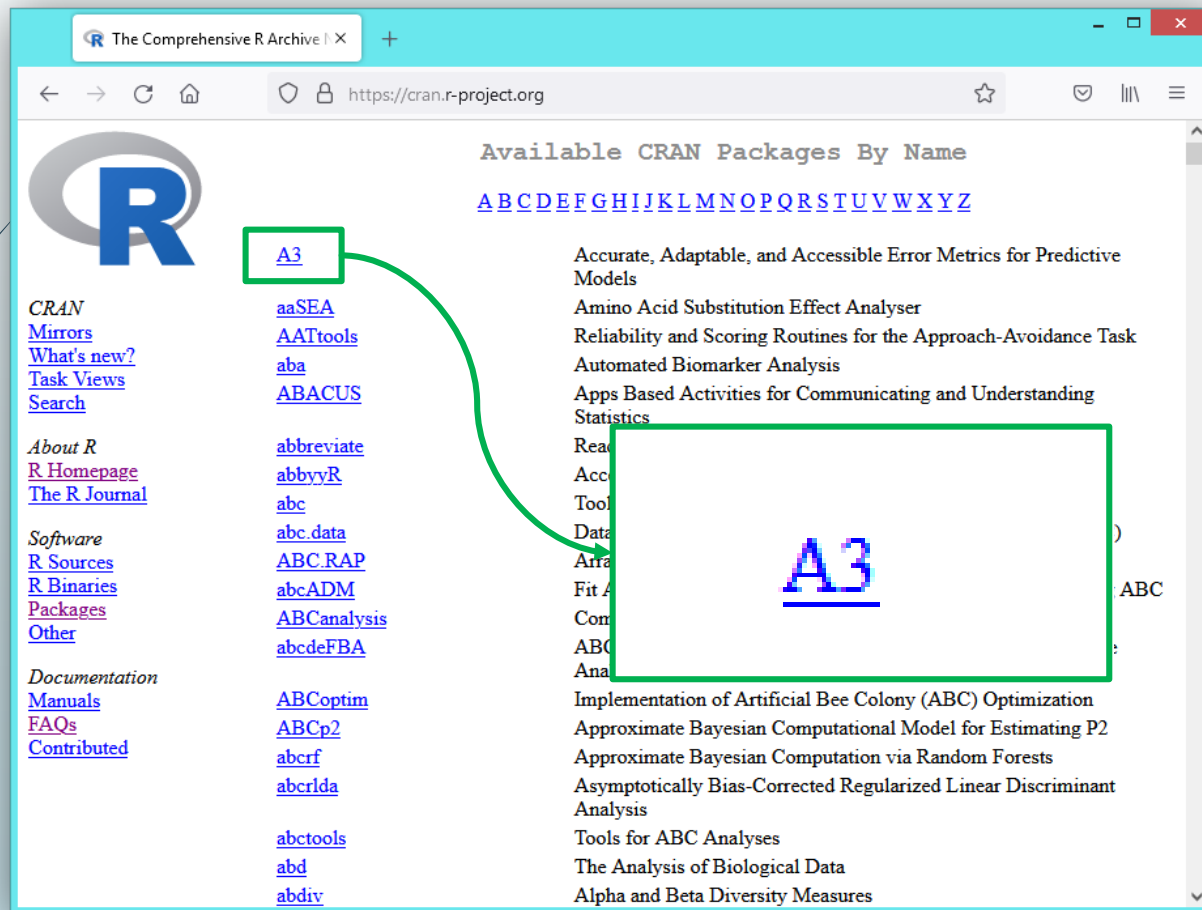
Second method



The screenshot shows the CRAN website at <https://cran.r-project.org>. The page title is "Contributed Packages". Under the "Available Packages" section, it states: "Currently, the CRAN package repository features 18762 available packages." Two links are highlighted with a green box: [Table of available packages, sorted by date of publication](#) and [Table of available packages, sorted by name](#). Below this, the "Installation of Packages" section provides instructions on how to install packages from the repository. A green arrow points from the highlighted links to a larger green box at the bottom of the screenshot, which contains the same two links: [Table of available packages, sorted by date of publication](#) and [Table of available packages, sorted by name](#).

Downloading and Installing a Package

Second method



The screenshot shows the CRAN website interface. The browser address bar displays <https://cran.r-project.org>. The page title is "Available CRAN Packages By Name". Below the title, there is a navigation bar with letters A through Z. The letter 'A' is highlighted, and a list of packages starting with 'A' is displayed. The package 'A3' is highlighted with a green box. A green arrow points from the 'A3' link in the list to the 'A3' package description. The package description for 'A3' is: "Accurate, Adaptable, and Accessible Error Metrics for Predictive Models".

Available CRAN Packages By Name

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

[A3](#)

[aaSEA](#)
[AATtools](#)
[aba](#)
[ABACUS](#)

[abbreviate](#)
[abbyR](#)
[abc](#)
[abc.data](#)
[ABC.RAP](#)
[abcADM](#)
[ABCanalysis](#)
[abcdeFBA](#)

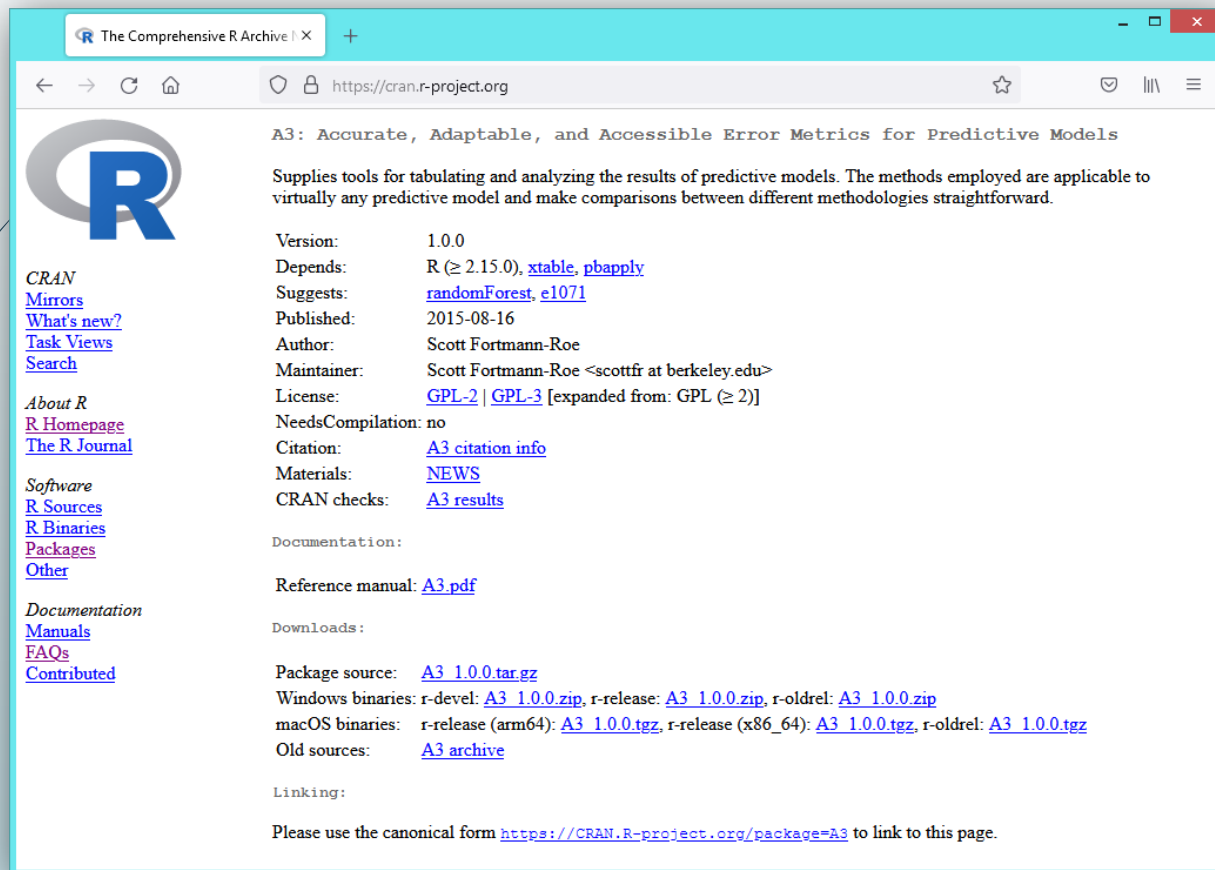
[ABCOptim](#)
[ABCp2](#)
[abcrf](#)
[abcrlda](#)

[abctools](#)
[abd](#)
[abdiv](#)

Accurate, Adaptable, and Accessible Error Metrics for Predictive Models
Amino Acid Substitution Effect Analyser
Reliability and Scoring Routines for the Approach-Avoidance Task
Automated Biomarker Analysis
Apps Based Activities for Communicating and Understanding Statistics
Read
Acc
Tool
Data
Arra
Fit A
Com
ABC
Ana
Implementation of Artificial Bee Colony (ABC) Optimization
Approximate Bayesian Computational Model for Estimating P2
Approximate Bayesian Computation via Random Forests
Asymptotically Bias-Corrected Regularized Linear Discriminant Analysis
Tools for ABC Analyses
The Analysis of Biological Data
Alpha and Beta Diversity Measures

Downloading and Installing a Package

➤ Second method



The screenshot shows a web browser window with the URL <https://cran.r-project.org>. The page displays the details for the R package 'A3: Accurate, Adaptable, and Accessible Error Metrics for Predictive Models'. The package description states: 'Supplies tools for tabulating and analyzing the results of predictive models. The methods employed are applicable to virtually any predictive model and make comparisons between different methodologies straightforward.'

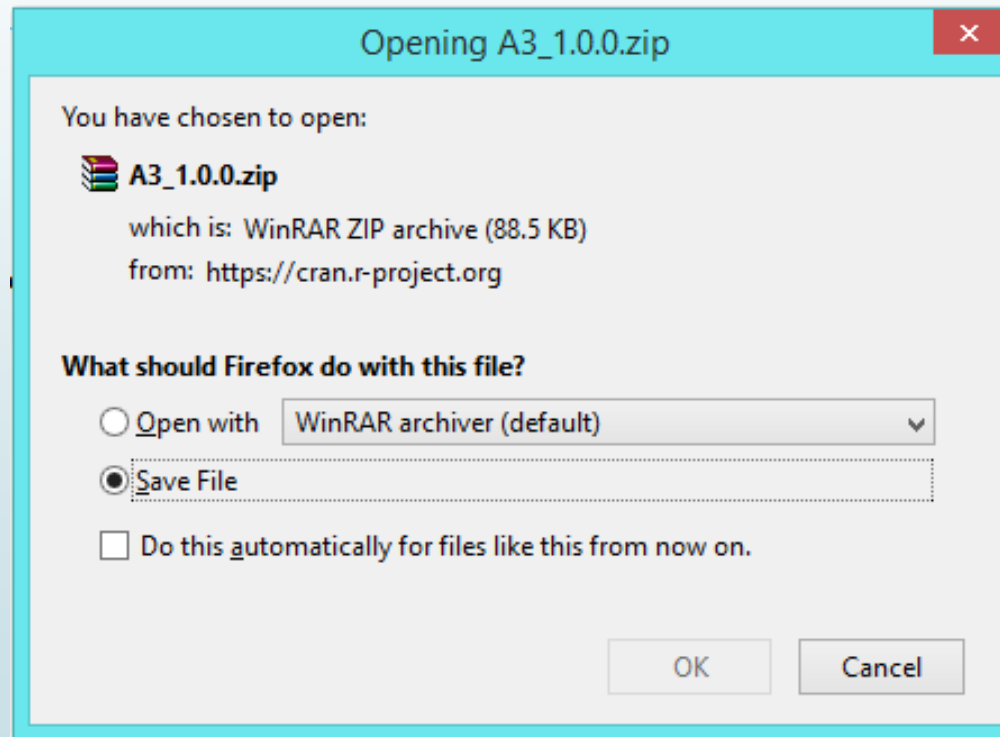
Key details from the screenshot include:

- Version:** 1.0.0
- Depends:** R ($\geq 2.15.0$), [xtable](#), [pbapply](#)
- Suggests:** [randomForest](#), [e1071](#)
- Published:** 2015-08-16
- Author:** Scott Fortmann-Roe
- Maintainer:** Scott Fortmann-Roe <[scottfr at berkeley.edu](mailto:scottfr@berkeley.edu)>
- License:** [GPL-2](#) | [GPL-3](#) [expanded from: GPL (≥ 2)]
- NeedsCompilation:** no
- Citation:** [A3 citation info](#)
- Materials:** [NEWS](#)
- CRAN checks:** [A3 results](#)

Additional information provided includes a reference manual ([A3.pdf](#)), download links for various operating systems (e.g., [A3 1.0.0.tar.gz](#), [A3 1.0.0.zip](#)), and a canonical URL for linking to the page: <https://CRAN.R-project.org/package=A3>.

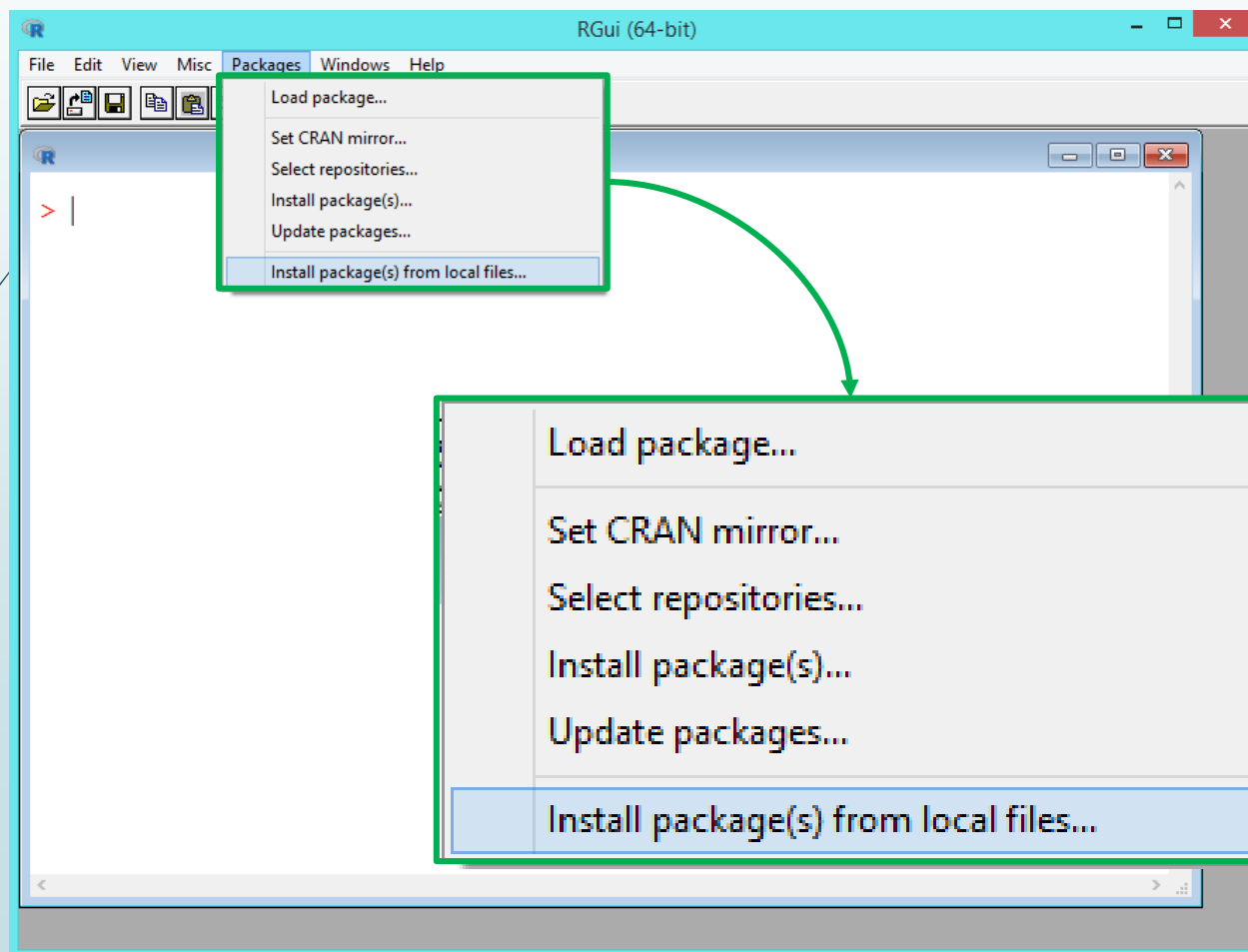
Downloading and Installing a Package

➤ Second method



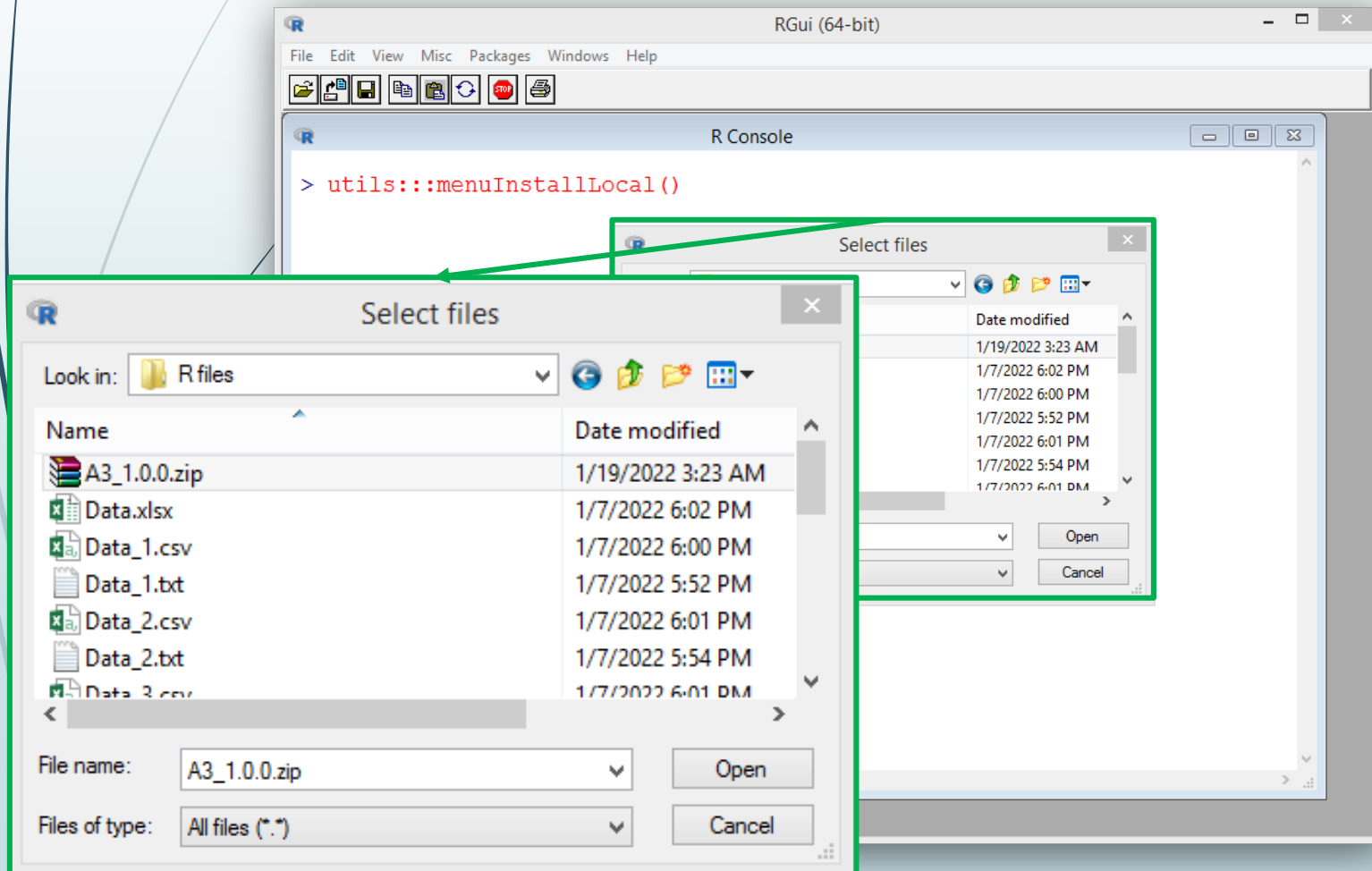
Downloading and Installing a Package

➤ Second method



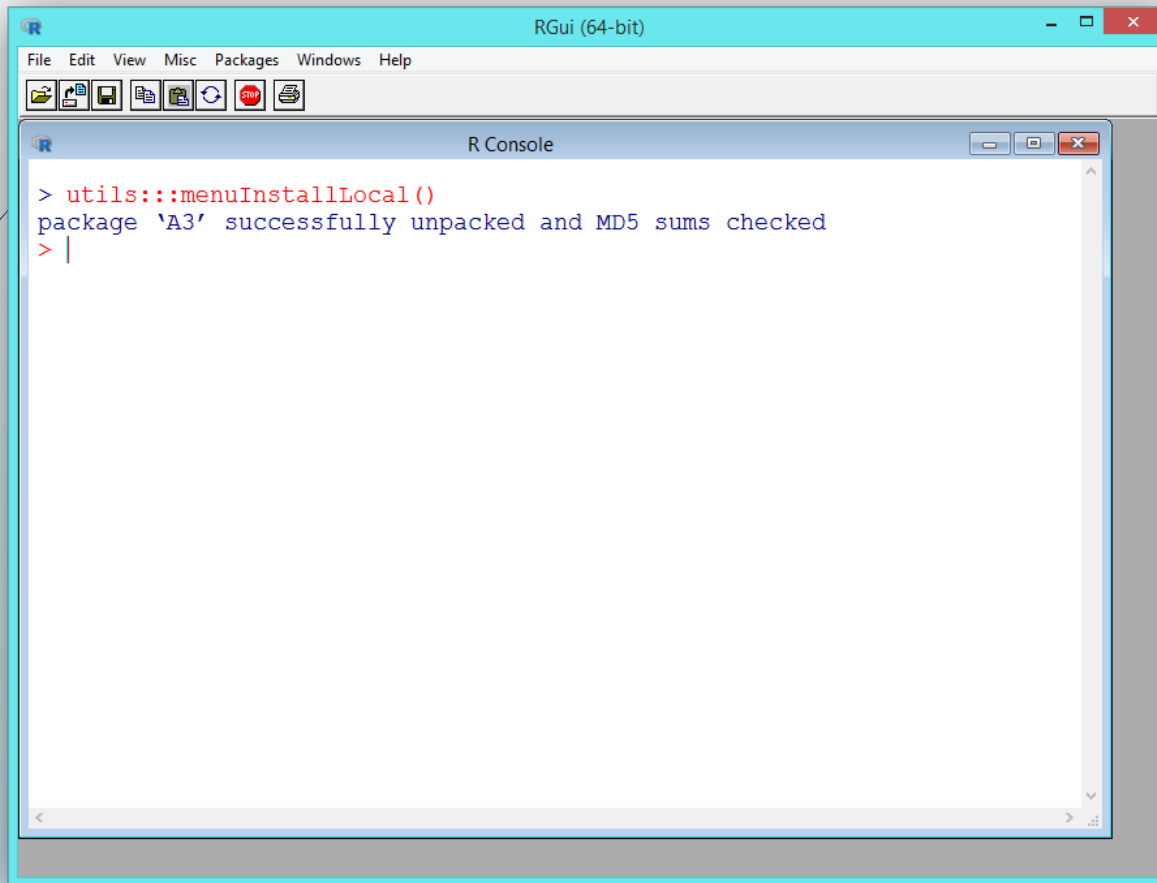
Downloading and Installing a Package

➤ Second method



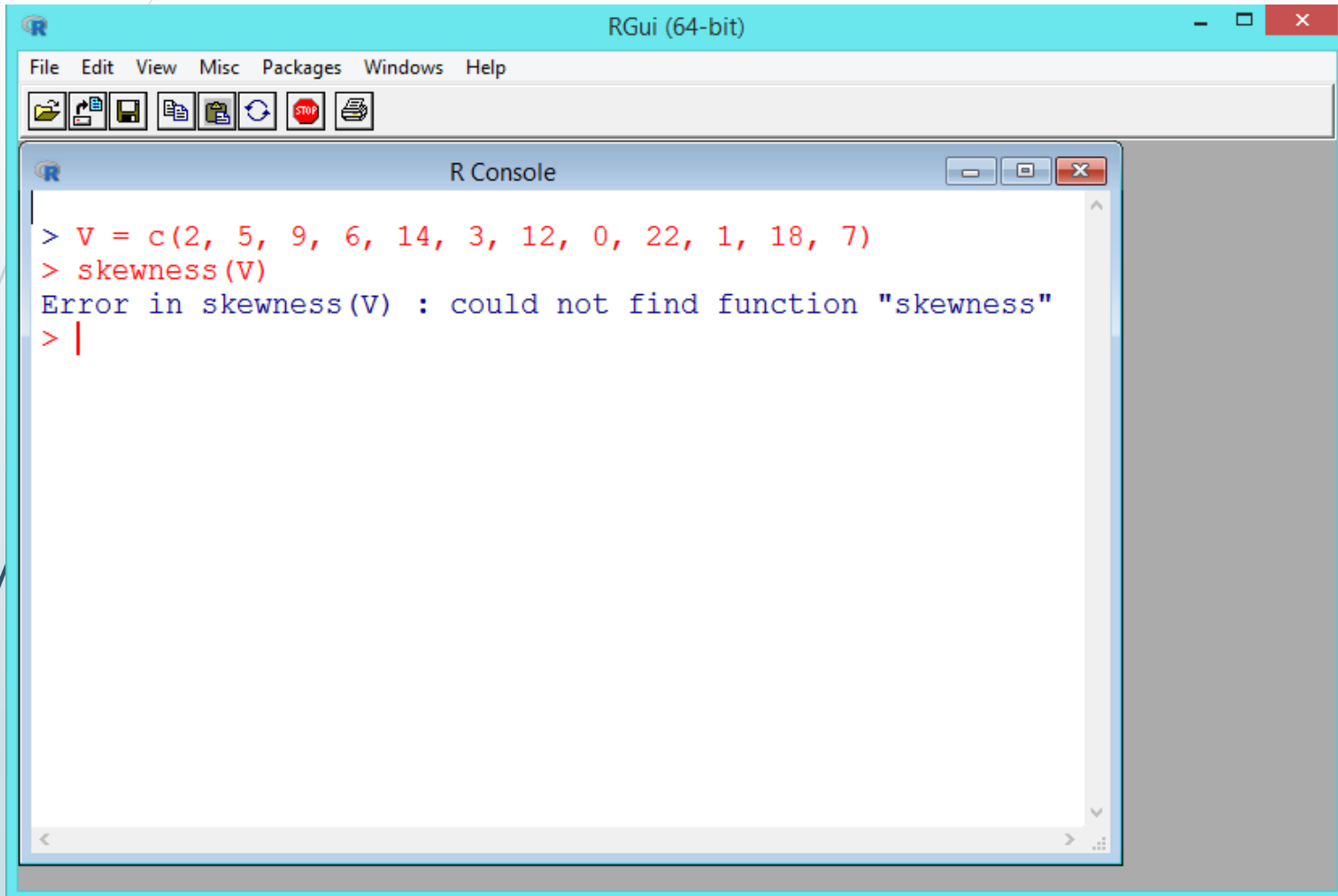
Downloading and Installing a Package

➤ Second method



```
> utils:::menuInstallLocal()
package 'A3' successfully unpacked and MD5 sums checked
> |
```

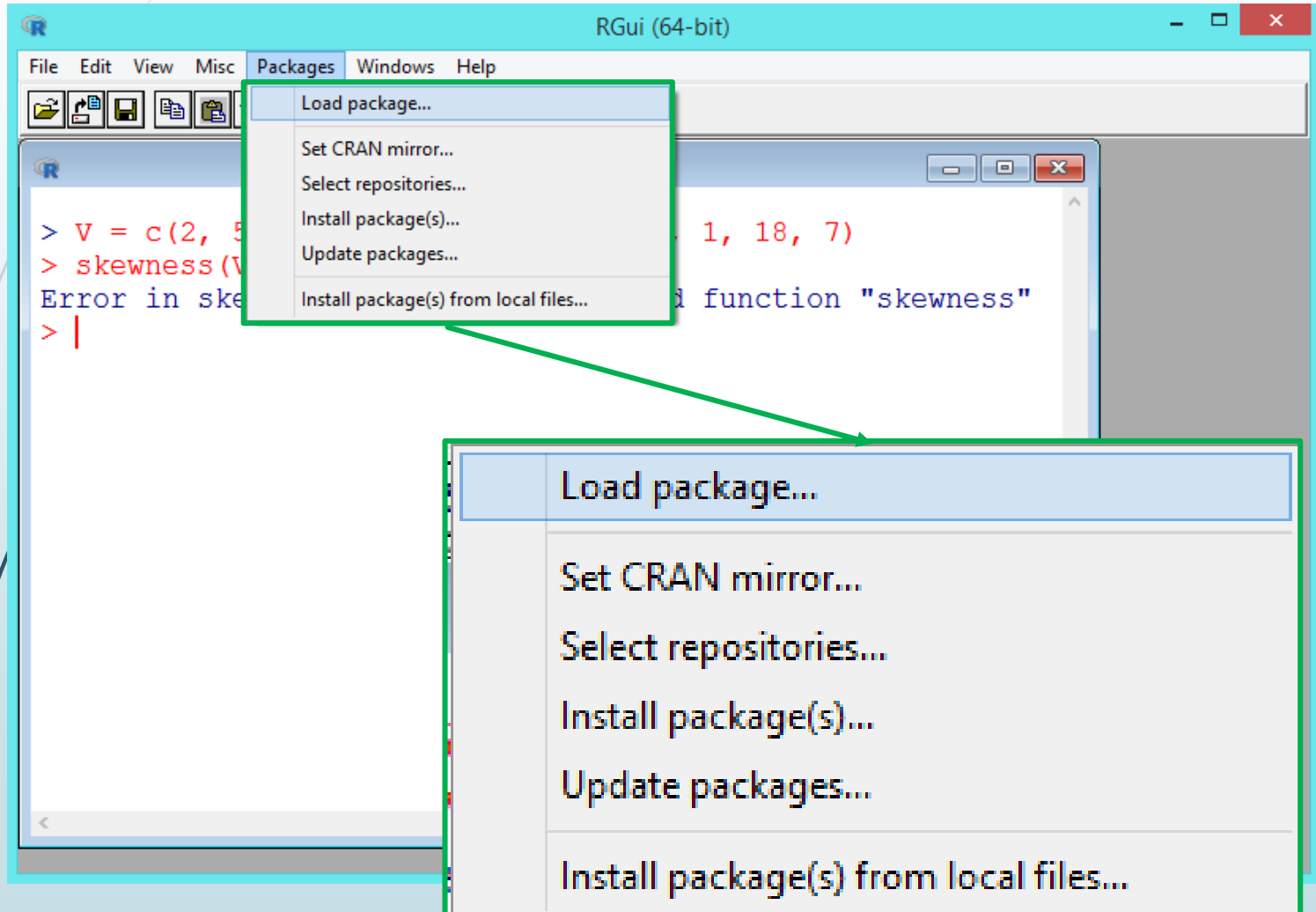
Loading a Package



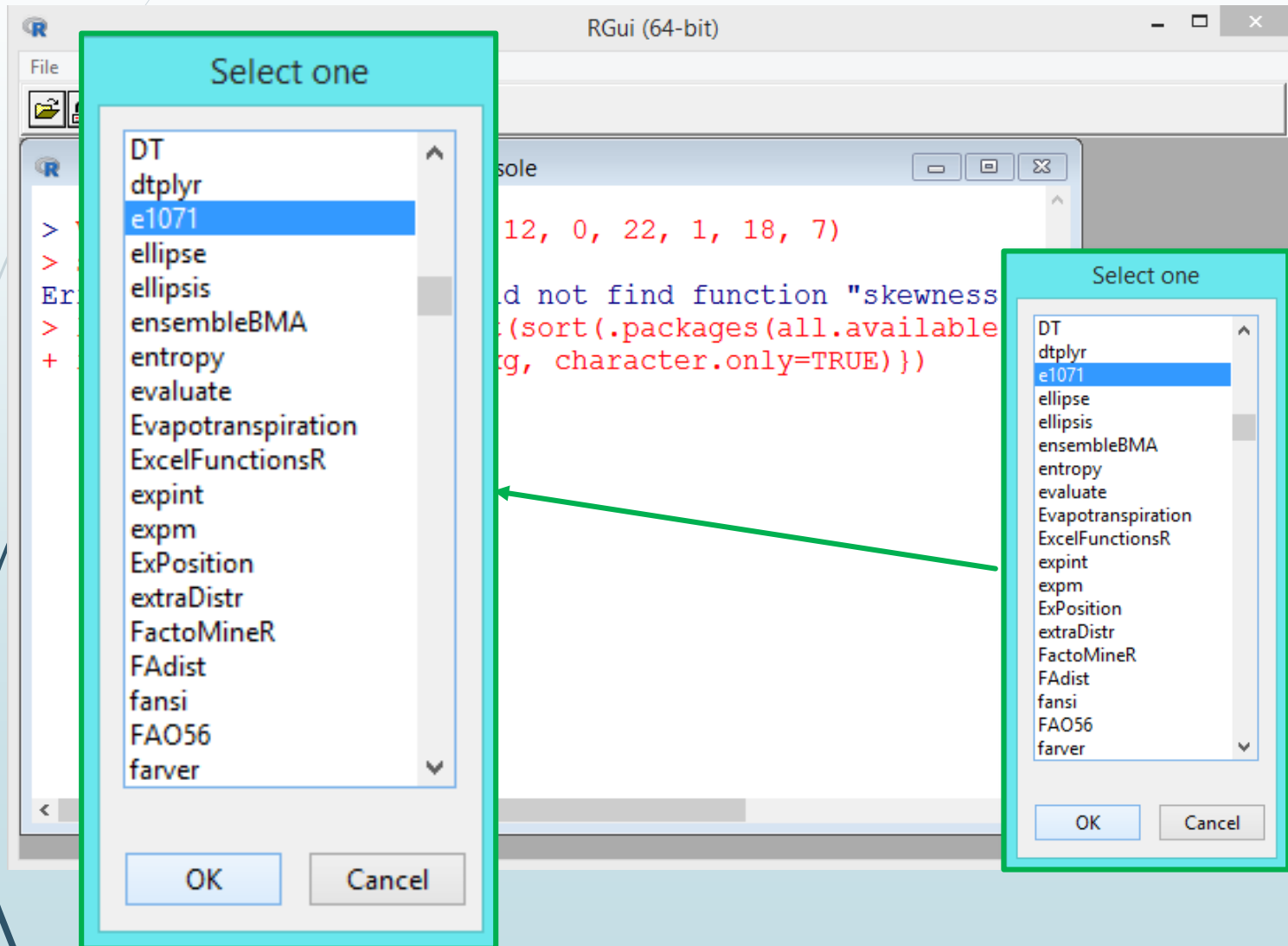
The screenshot shows the RGui (64-bit) interface. The main window has a menu bar with 'File', 'Edit', 'View', 'Misc', 'Packages', 'Windows', and 'Help'. Below the menu bar is a toolbar with icons for file operations and execution. The 'R Console' window is open, displaying the following R code and error message:

```
> V = c(2, 5, 9, 6, 14, 3, 12, 0, 22, 1, 18, 7)
> skewness(V)
Error in skewness(V) : could not find function "skewness"
> |
```

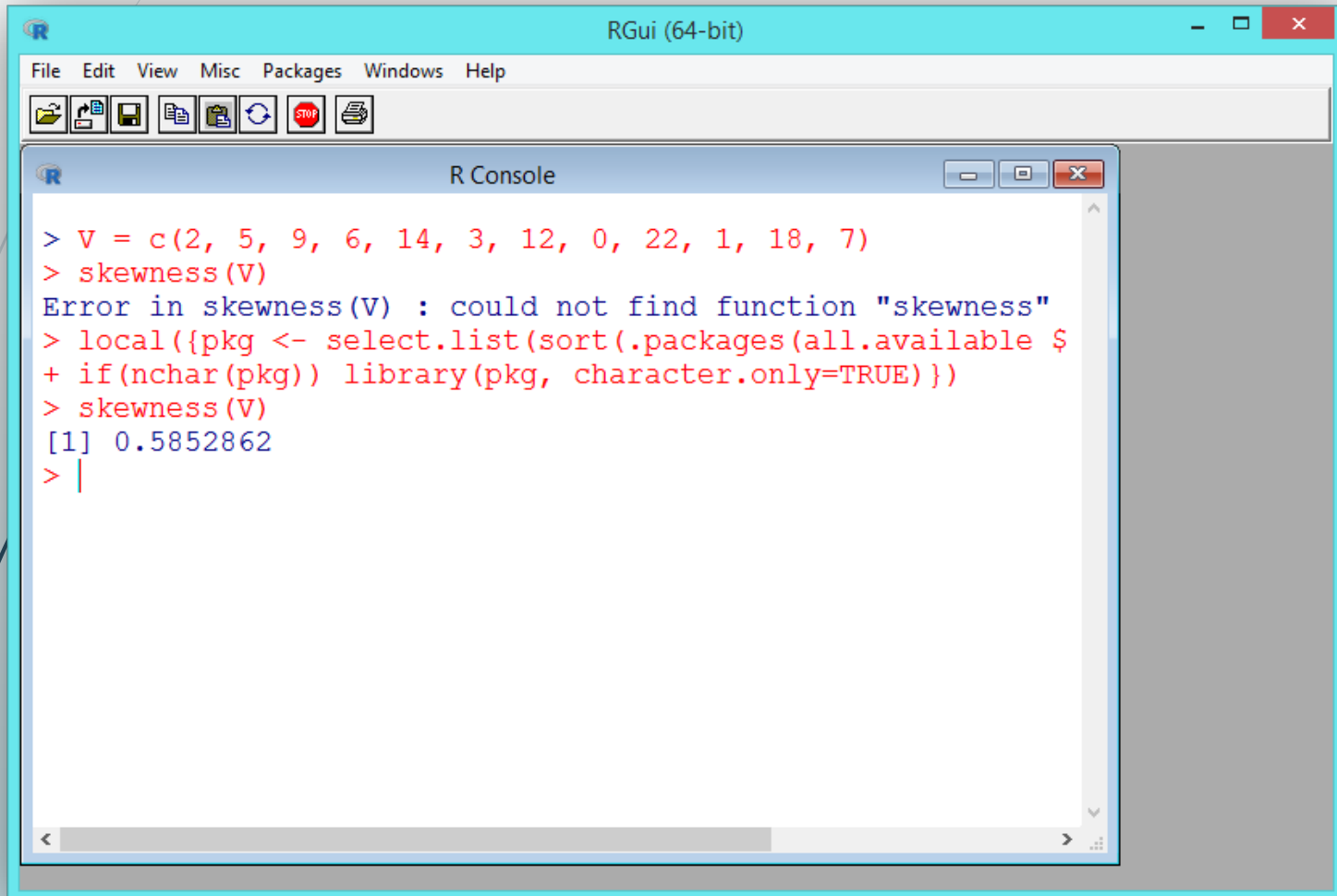

Loading a Package



Loading a Package



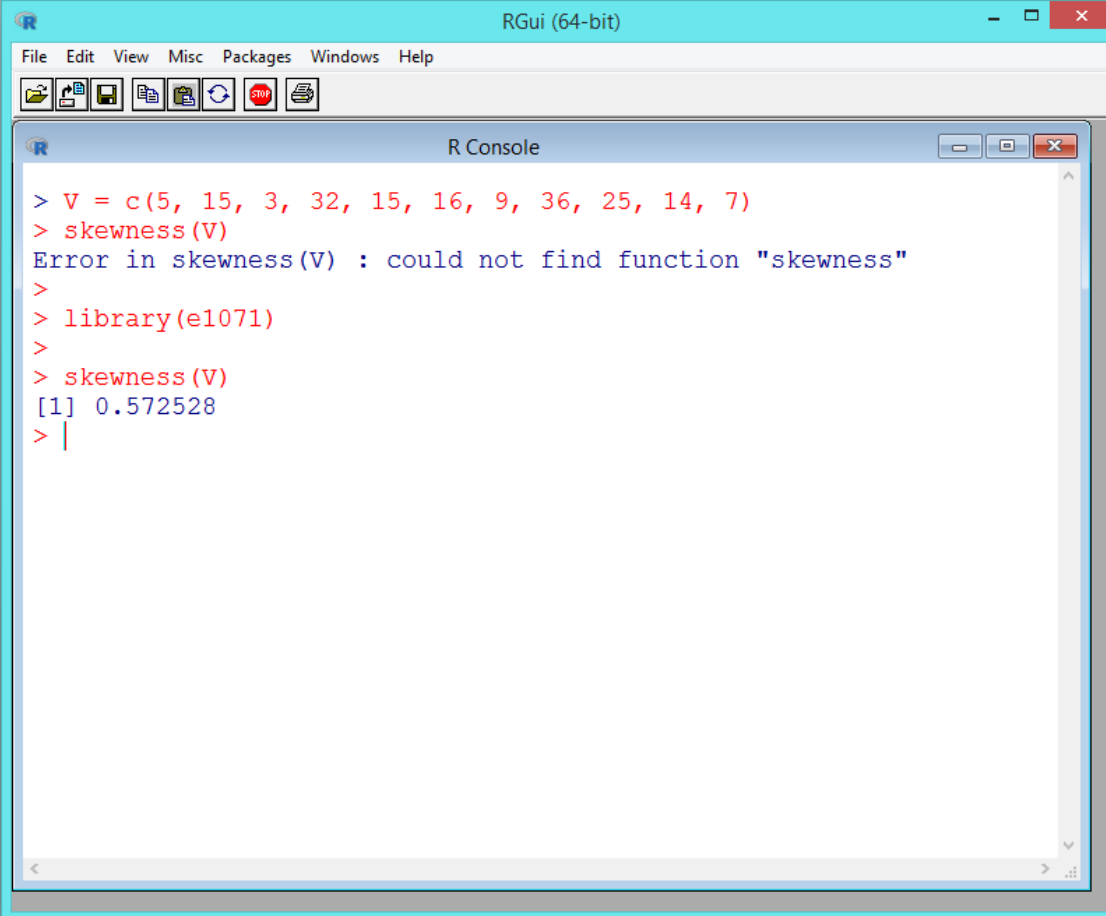
Loading a Package



```
> V = c(2, 5, 9, 6, 14, 3, 12, 0, 22, 1, 18, 7)
> skewness(V)
Error in skewness(V) : could not find function "skewness"
> local({pkg <- select.list(sort(.packages(all.available $
+ if(nchar(pkg)) library(pkg, character.only=TRUE))})
> skewness(V)
[1] 0.5852862
> |
```

Loading a Package

➤ `library()`

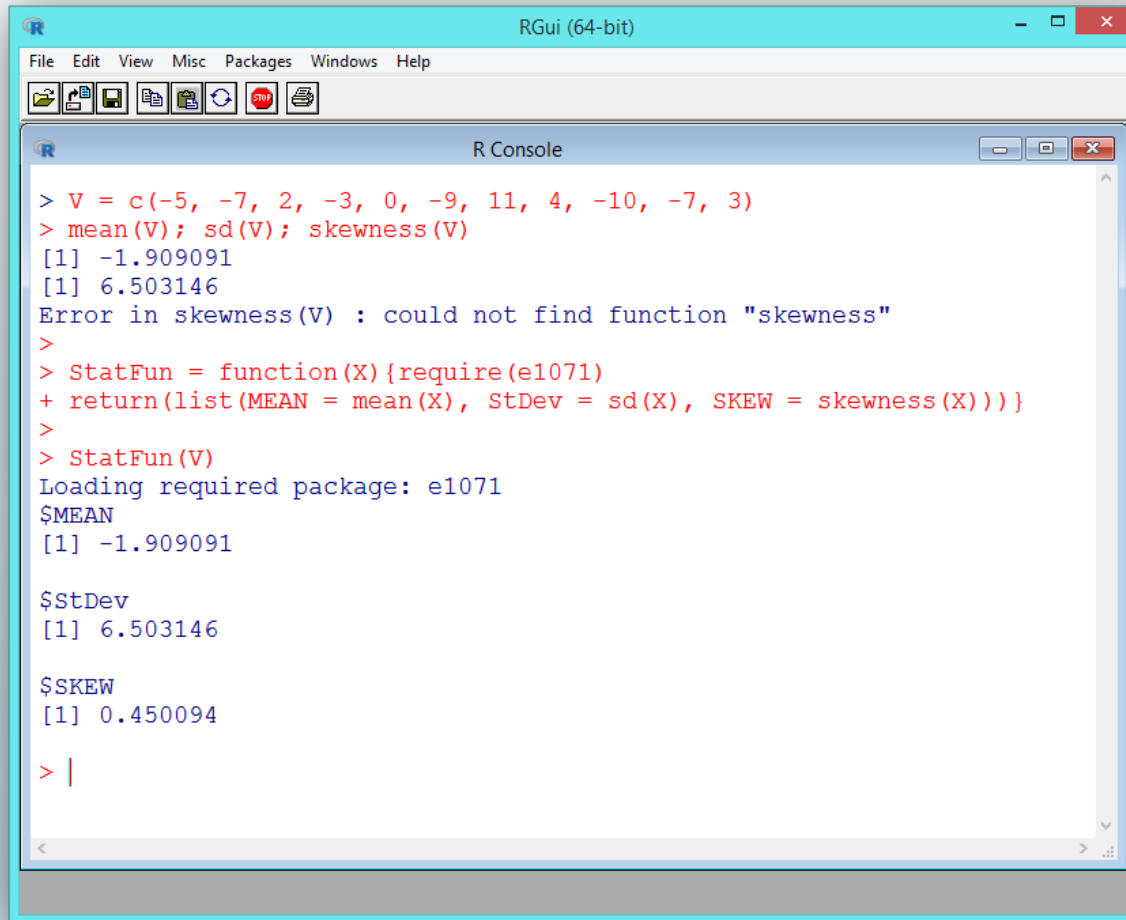


```
RGui (64-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console
> V = c(5, 15, 3, 32, 15, 16, 9, 36, 25, 14, 7)
> skewness(V)
Error in skewness(V) : could not find function "skewness"
>
> library(e1071)
>
> skewness(V)
[1] 0.572528
> |
```

Loading a Required Package

➤ `require()`



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> V = c(-5, -7, 2, -3, 0, -9, 11, 4, -10, -7, 3)
> mean(V); sd(V); skewness(V)
[1] -1.909091
[1] 6.503146
Error in skewness(V) : could not find function "skewness"
>
> StatFun = function(X){require(e1071)
+ return(list(MEAN = mean(X), StDev = sd(X), SKEW = skewness(X)))}
>
> StatFun(V)
Loading required package: e1071
$MEAN
[1] -1.909091

$StDev
[1] 6.503146

$SKEW
[1] 0.450094

> |
```

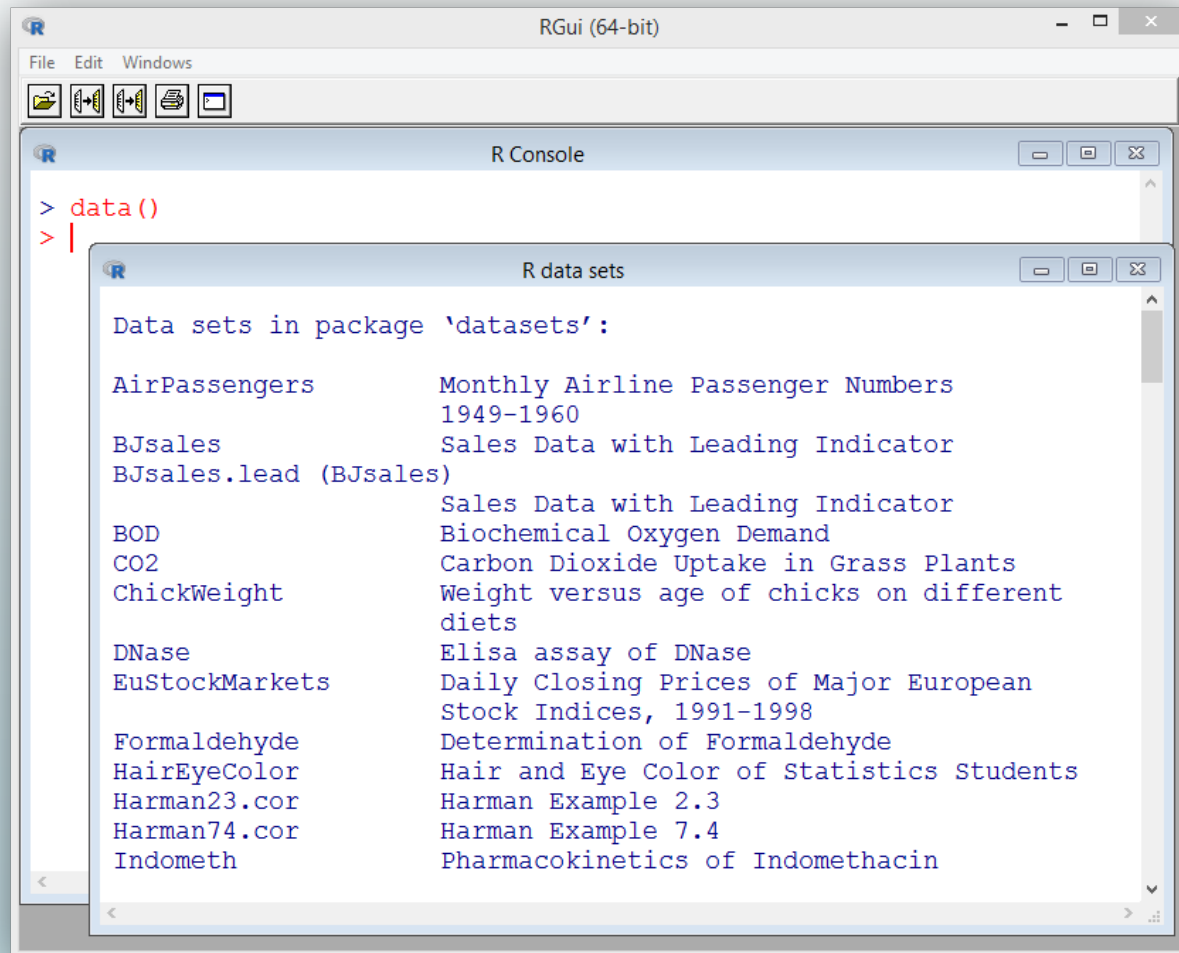


R Data Sets

Built-in Data Sets

R Data Sets

➔ data ()

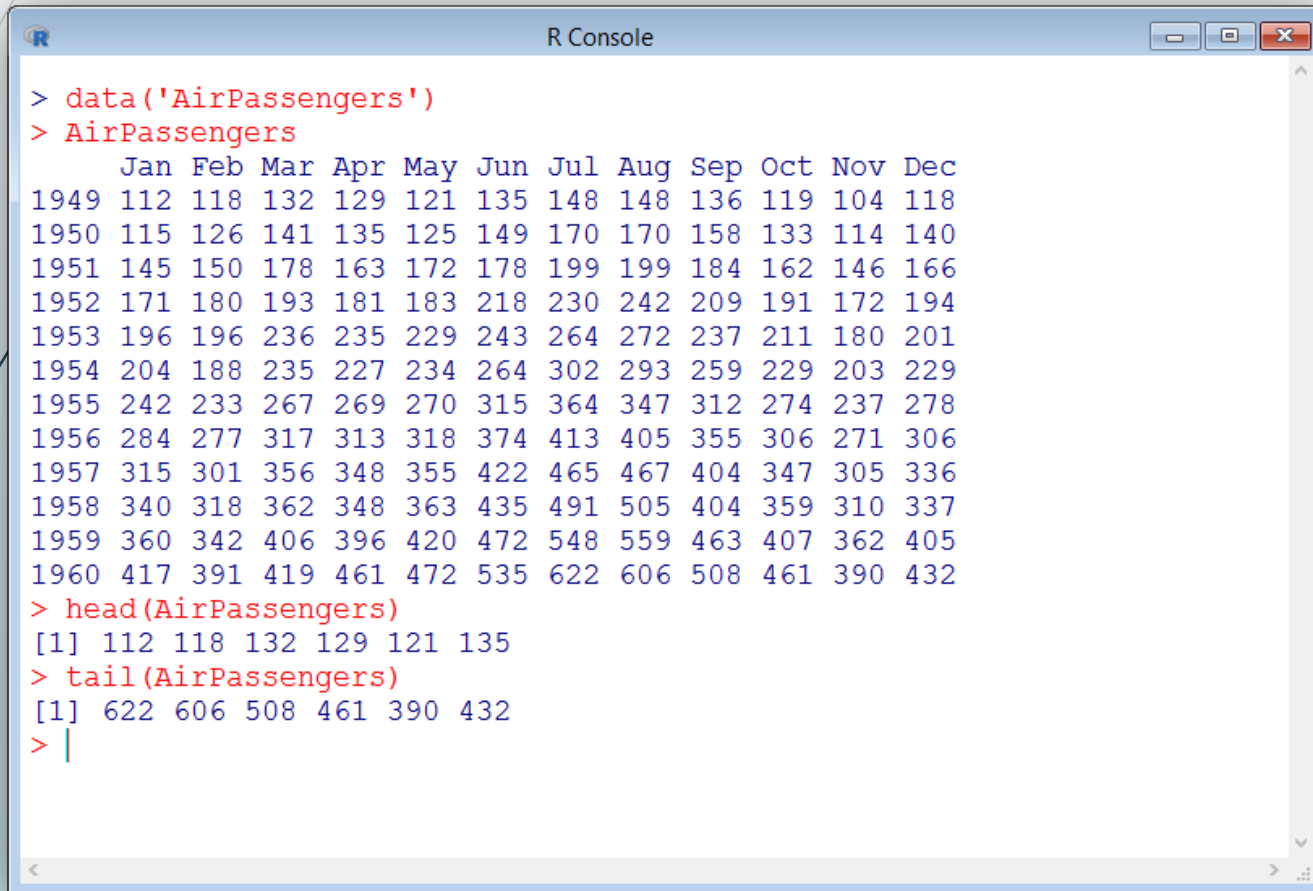


The screenshot shows the R GUI (64-bit) with the R Console and R data sets windows. The R Console shows the command `> data()` and the R data sets window displays the following output:

```
Data sets in package 'datasets':  
  
AirPassengers      Monthly Airline Passenger Numbers  
                    1949-1960  
BJsales            Sales Data with Leading Indicator  
BJsales.lead (BJsales) Sales Data with Leading Indicator  
  
BOD                Biochemical Oxygen Demand  
CO2                Carbon Dioxide Uptake in Grass Plants  
ChickWeight       Weight versus age of chicks on different  
                    diets  
  
DNase              Elisa assay of DNase  
EuStockMarkets    Daily Closing Prices of Major European  
                    Stock Indices, 1991-1998  
  
Formaldehyde       Determination of Formaldehyde  
HairEyeColor      Hair and Eye Color of Statistics Students  
Harman23.cor       Harman Example 2.3  
Harman74.cor       Harman Example 7.4  
Indometh           Pharmacokinetics of Indomethacin
```

R Data Sets

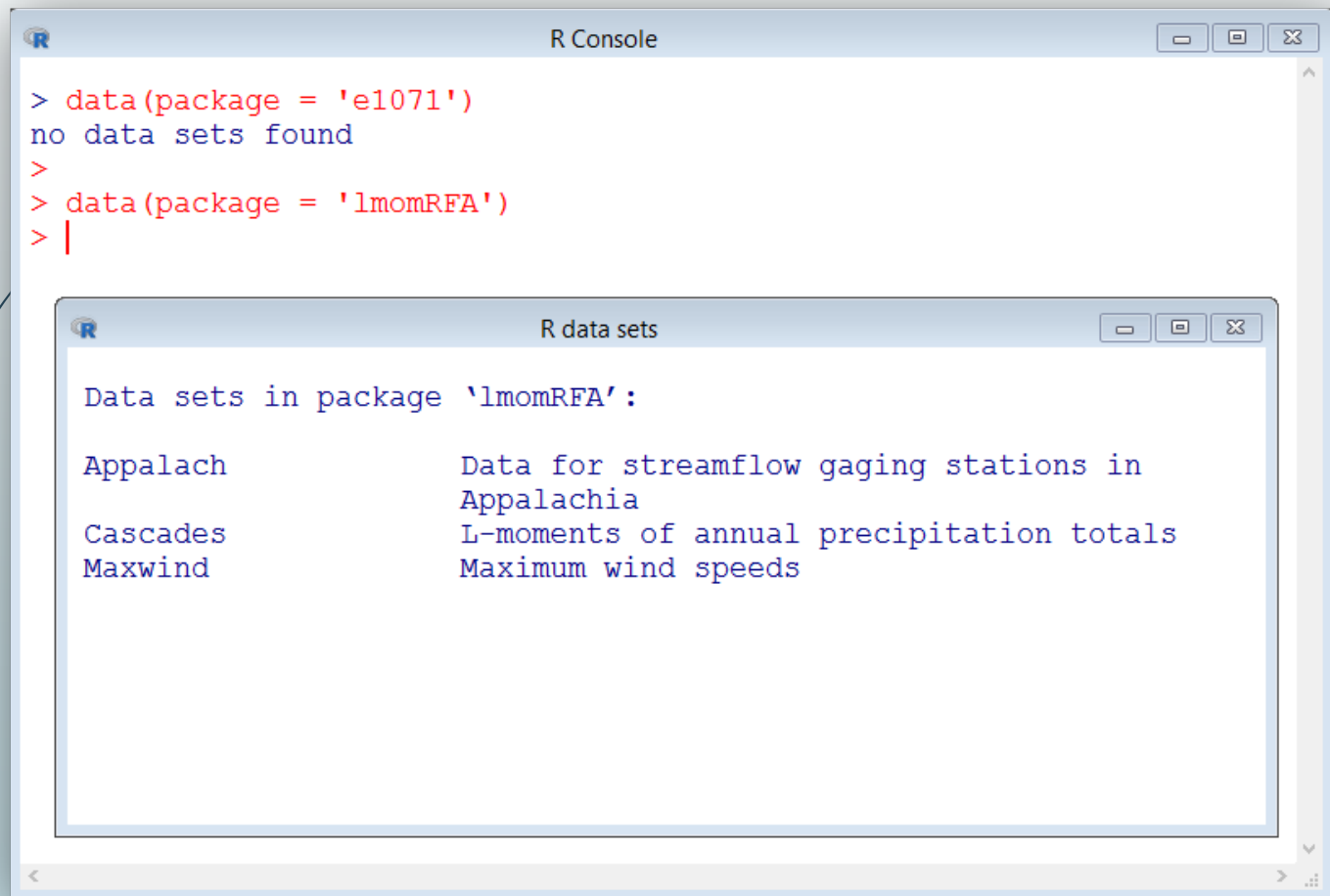
► `data()`, `head()`, `tail()`



```
R Console
> data('AirPassengers')
> AirPassengers
      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949 112 118 132 129 121 135 148 148 136 119 104 118
1950 115 126 141 135 125 149 170 170 158 133 114 140
1951 145 150 178 163 172 178 199 199 184 162 146 166
1952 171 180 193 181 183 218 230 242 209 191 172 194
1953 196 196 236 235 229 243 264 272 237 211 180 201
1954 204 188 235 227 234 264 302 293 259 229 203 229
1955 242 233 267 269 270 315 364 347 312 274 237 278
1956 284 277 317 313 318 374 413 405 355 306 271 306
1957 315 301 356 348 355 422 465 467 404 347 305 336
1958 340 318 362 348 363 435 491 505 404 359 310 337
1959 360 342 406 396 420 472 548 559 463 407 362 405
1960 417 391 419 461 472 535 622 606 508 461 390 432
> head(AirPassengers)
[1] 112 118 132 129 121 135
> tail(AirPassengers)
[1] 622 606 508 461 390 432
> |
```


Data Sets in the R Packages

➔ `data()`



The image shows two overlapping R windows. The top window, titled 'R Console', displays the following commands and output:

```
> data(package = 'e1071')
no data sets found
>
> data(package = 'lmomRFA')
> |
```

The bottom window, titled 'R data sets', displays the following output:

```
Data sets in package 'lmomRFA':

Appalach      Data for streamflow gaging stations in
               Appalachia
Cascades      L-moments of annual precipitation totals
Maxwind       Maximum wind speeds
```

References

- Venables, W. N., Smith, D. M., and the R Core Team. 2021. An Introduction to R - Notes on R: A Programming Environment for Data Analysis and Graphics Version 4.1.1 (2021-08-10).

- موسوی ندوشنی، س. س. ۱۳۹۱. آشنایی با زبان محاسبات آماری R (ویراست نوشتار ۱/۸/۰).



**Thank you for your
attention**