

R/Python programming & ChatGPT



```
> set.seed(1);  
> (mat.1 <- matrix(sample(12), nrow=3))  
      [,1] [,2] [,3] [,4]  
[1,]    4    9   10    1  
[2,]    5    2   12   11  
[3,]    6    7    3    8  
> set.seed(2);  
> (mat.2 <- matrix(sample(12), nrow=3))  
      [,1] [,2] [,3] [,4]  
[1,]    3    2    1    4  
[2,]    8   11    5   12  
[3,]    6    7    9   10
```

```
> mat.1 + 3  
      [,1] [,2] [,3] [,4]  
[1,]    7   12   13    4  
[2,]    8    5   15   14  
[3,]    9   10    6   11
```

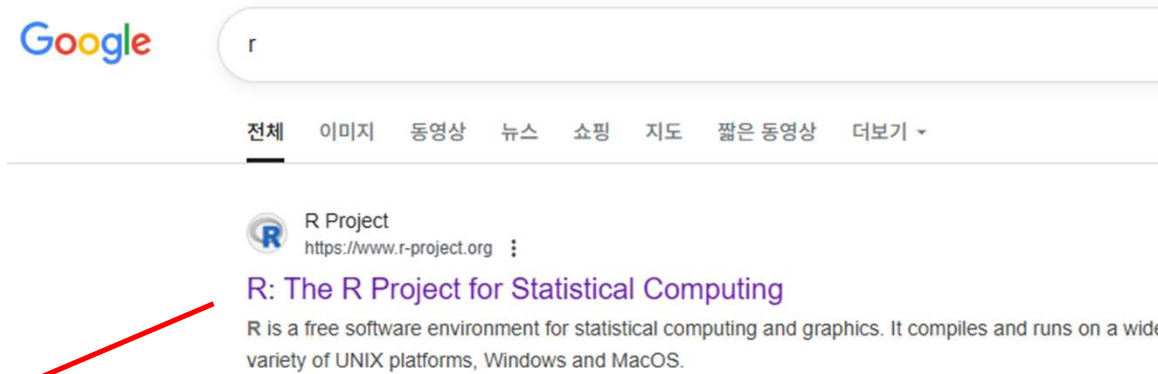
```
> c(2,3,5) + mat.1  
      [,1] [,2] [,3] [,4]  
[1,]    6   11   12    3  
[2,]    8    5   15   14  
[3,]   11   12    8   13
```

```
> mat.2 - 3  
      [,1] [,2] [,3] [,4]  
[1,]    0   -1   -2    1  
[2,]    5    8    2    9  
[3,]    3    4    6    7
```

```
> mat.2 - 1:2  
      [,1] [,2] [,3] [,4]  
[1,]    2    0    0    2  
[2,]    6   10    3   11  
[3,]    5    5    8    8
```

<http://dic1224.blog.me>

R 설치



[Home]

Download

CRAN

R Project

About R

Logo

Contributors

What's New?

Reporting Bugs

Conferences

Search

Get Involved: Mailing Lists

Get Involved: Contributing

Developer Pages

R Blog

R Foundation

Foundation

Board

Members

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- **R version 4.5.0 (How About a Twenty-Six) prerelease versions** will appear starting Tuesday 2025-03-11. Final release is scheduled for Friday 2025-04-11.
- **R version 4.4.3 (Trophy Case)** has been released on 2025-02-28.
- The **useR! 2025** conference will take place at Duke University, in Durham, NC, USA, August 8-10.
- We are deeply sorry to announce that our friend and colleague Friedrich (Fritz) Leisch has died. [Read our tribute to Fritz here.](#)
- **R version 4.3.3 (Angel Food Cake)** (wrap-up of 4.3.x) was released on 2024-02-29.
- You can support the R Foundation with a renewable subscription as a [supporting member](#).

News via Mastodon

Italy

<https://cran.mirror.garr.it/CRAN/>

<https://cran.stat.unipd.it/>

Japan

<https://ftp.yz.yamagata-u.ac.jp/pub/cran/>

Mexico

<https://cran.itam.mx/>

<https://est.colpos.mx/>

Morocco

<https://mirror.marwan.ma/cran/>

Netherlands

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:

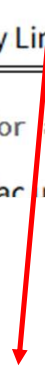
- [Download R for Linux](#) ([Debian](#), [Fedora/Redhat](#), [Ubuntu](#))
- [Download R for macOS](#)
- [Download R for Windows](#)

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

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 $\geq$ 4.0.x).
old contrib	Binaries of contributed CRAN packages for outdated versions of R (for R $<$ 4.0.x).
Rtools	Tools to build R and R packages. This is what you want to build your own packages

[Download R-4.6.0 for Windows](#) (88 megabytes, 64 bit)

[README on the Windows binary distribution](#)

[New features in this version](#)

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

For Apple silicon (M1,2,...) Macs:

[R-4.6.0-arm64.pkg](#)

SHA1-hash: 85775cbe3ba6242d2a2f0693a76422e3303bda9c
(ca. 101MB, notarized and signed)

R 4.6.0 binary for macOS 14 (**Sonoma**) and higher for arm64 Macs, signed and notarized.

Contains R 4.6.0 framework, R.app GUI 1.83, Tcl/Tk 8.6.12 X11 libraries and Texinfo. Some components are optional and can be omitted when choosing "custom install", the user can also want to use the tcltk R package or build package documentation from sources.

If installing from the *Downloads* folder you must allow the Apple Installer access to the access will result in you not being able to install any software from there and the file to a location that is not private such as your home folder.

Note: the use of X11 (including tcltk) requires [XQuartz](#) (version 2.8.5 or later). Always upgrading your macOS to a new major version.

This release uses Xcode 26.0 (the same LLVM toolchain up to 26.3) and GNU Fortran to compile R packages which contain Fortran code, you may need to download the compiler from <https://mac.R-project.org/tools>. Any external libraries and tools are in `/opt/R/arm64`.

For older Intel Macs:

[R-4.6.0-x86_64.pkg](#)

SHA1-hash: e6430cd1b7b27e83597edcdf95d0a828228b701a
(ca. 104MB, notarized and signed)

R 4.6.0 binary for macOS 11 (**Big Sur**) and higher, signed and notarized packages.

Contains R 4.6.0 framework, R.app GUI 1.83, Tcl/Tk 8.6.12 X11 libraries and Texinfo. Some components are optional and can be omitted when choosing "custom install", the user can also want to use the tcltk R package or build package documentation from sources.

Rstudio installation

<https://posit.co/download/rstudio-desktop/>

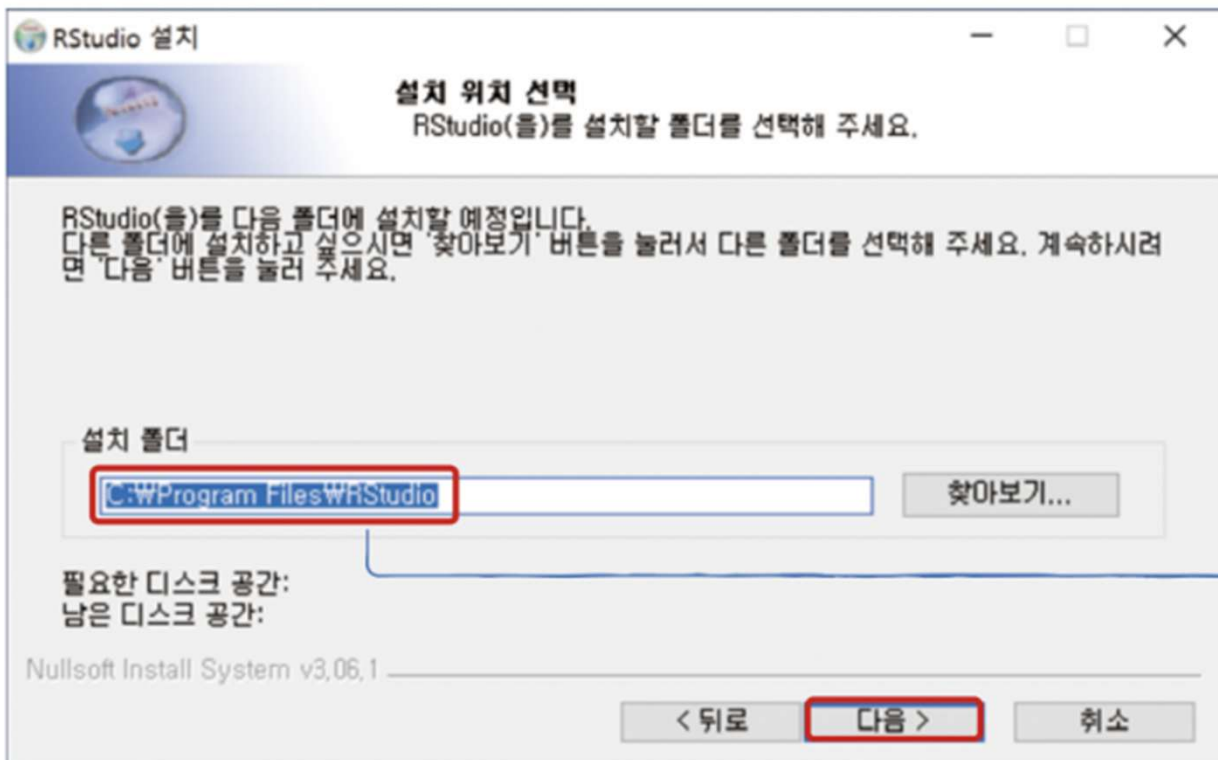


Navigation links: [About](#) [Partners](#) [Download RStudio](#) [Download Positron](#)

Secondary navigation: [Solutions](#) [Products](#) [Open Source](#) [Customers](#) [Resources](#) [GET STARTED](#) [DEMO GALLERY](#) [🔍](#)

More resources

OS	Download	Size	SHA-256
Windows 11	RSTUDIO-2026.04.0-526.EXE 	319.84 MB	8B5DA756
macOS 14/15/26	RSTUDIO-2026.04.0-526.DMG 	694.56 MB	F49B0016



경로와 폴더 이름에는
한글이 없어야 합니다.

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function

Addins

Project: (None)

Console Terminal x Background Jobs x

R 4.4.3 · ~/

R version 4.4.3 (2025-02-28 ucrt) -- "Trophy Case"
Copyright (c) 2025 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

R은 자유 소프트웨어이며, 어떠한 형태의 보증없이 배포됩니다.
또한, 일정한 조건하에서 이것을 재배포 할 수 있습니다.
배포와 관련된 상세한 내용은 'license()' 또는 'licence()'을 통하여 확인할 수 있습니다.

R은 많은 기여자들이 참여하는 공동프로젝트입니다.
'contributors()'라고 입력하시면 이에 대한 더 많은 정보를 확인할 수 있습니다.
그리고, R 또는 R 패키지들을 출판물에 인용하는 방법에 대해서는 'citation()'을 통해 확인할 수 있습니다.

'demo()'를 입력하신다면 몇가지 데모를 보실 수 있으며, 'help()'를 입력하시면 온라인 도움말을 이용하실 수 있습니다.
또한, 'help.start()'의 입력을 통하여 HTML 브라우저에 의한 도움말을 이용하실 수 있습니다.
R의 종료를 원하시면 'q()'을 입력해주세요.

> |

Environment History Connections Tutorial

Import Dataset 92 MiB

Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

Folder File Delete Rename

Home

	Name	Size	Modified
☐	.Rhistory	20 B	Mar 13, 2025, 11:21 AM
☐	'fmLj7g.jpg	228 KB	Mar 13, 2025, 8:47 AM
☐	사용자 지정 Office 서식 파일		
☐	카카오톡 받은 파일		
☐	Default.rdp	0 B	Feb 28, 2025, 3:32 PM
☐	desktop.ini	402 B	Mar 4, 2025, 8:34 AM
☐	My Music		
☐	My Pictures		
☐	My Videos		
☐	Welcome to Hwp.hwp	26.5 KB	Jun 13, 2023, 7:56 AM
☐	Welcome to Word.hwd	14.8 KB	Jun 13, 2023, 7:56 AM

-Downloading R packages

R

```
install.packages('Seurat')
```

```
# https://satijalab.org/seurat/
```



Seurat v5

We are excited to release Seurat v5! To install, please follow the instructions in our [install page](#). This update brings the following new features and functionality:

예제 1:

```
# Declare variables of different types
# Numeric
x <- 28
class(x)
```

결과:

```
## [1] "numeric"
```

예제 2:

```
# String
y <- "R is Fantastic"
class(y)
```

결과:

```
## [1] "character"
```

예제 3:

```
# Boolean
z <- TRUE
class(z)
```

결과:

```
## [1] "logical"
```

-Class: variable type

<https://misdb.github.io/R/R-Tutorial/>

예제 4:

```
# Declare variables of different types
# Numeric
x <- 28
str(x)
```

결과 :

```
## num 28
```

예제 5:

```
# String
y <- "R is Fantastic"
str(y)
```

결과:

```
## chr "R is Fantastic"
```

예제 6:

```
# Boolean
z <- TRUE
str(z)
```

결과:

```
## logi TRUE
```

-Class change
str
numeric
character
as.~

-Variable declaration

예제 1:

```
# Print variable x
x <- 42
x
```

결과:

```
## [1] 42
```

예제 2:

```
y <- 10
y
```

결과:

```
## [1] 10
```

예제 3:

```
# We call x and y and apply a subtraction
x-y
```

결과:

```
## [1] 32
```

-Vector

예제 1: (정수형)

```
# Numerical
vec_num <- c(1, 10, 49)
vec_num
```

결과:

```
## [1]  1 10 49
```

예제 2:(문자형)

```
# Character
vec_chr <- c("a", "b", "c")
vec_chr
```

결과:

```
## [1] "a" "b" "c"
```

예제 3: (논리형)

```
# Boolean
vec_bool <- c(TRUE, FALSE, TRUE)
vec_bool
```

결과:

```
## [1] TRUE FALSE TRUE
```

예제 1:

```
# Create the vectors
vect_1 <- c(1, 3, 5)
vect_2 <- c(2, 4, 6)
# Take the sum of A_vector and B_vector
sum_vect <- vect_1 + vect_2

# Print out total_vector
sum_vect
```

결과:

```
## [1] 3 7 11
```

예제 2:

```
# An addition
3 + 4
```

결과:

```
## [1] 7
```

예제 3:

```
# A multiplication
3*5
```

결과:

```
## [1] 15
```

예제 4:

```
# A division
(5+5)/2
```

결과:

```
## [1] 5
```

예제 5:

```
# Exponentiation
2^5
# or
2**5
```

결과:

```
## [1] 32
```

-Operator

연산자	설명
+	덧셈(Addition)
-	뺄셈(Subtraction)
*	곱셈(Multiplication)
/	나눗셈(Division)
^ or **	제곱(Exponentiation)
%%	나머지(Modulo)
%/%	몫(Quotient)

예제 6:

```
# Modulo : %% (나머지)
27%%6
```

결과:

```
## [1] 3
```

예제 7:

```
# Quotient : %/% (몫)
27%/%6
```

결과:

```
## [1] 4
```

연산자	설명
<	작다
<=	작거나 같다
>	크다
>=	크거나 같다
==	같다
!=	다르다
!x	not x
x	x
x y	논리합 (or)
x&y	논리곱 (and)
isTRUE (x)	x가 TRUE인가?

예제 1:

```
# Create a vector from 1 to 10
logical_vector <- 1:10

# logical statement
logical_vector > 5
```

결과:

```
## [1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
```

예제 3:

```
# Print 5 and 6
logical_vector <- 1:10
logical_vector[(logical_vector>4) & (logical_vector<7)]
```

결과:

```
## [1] 5 6
```

-Vector slicing (subsetting)

```
# Slice the first five rows of the vector
slice_vector <- c(1,2,3,4,5,6,7,8,9,10)
slice_vector[1:5]
```

결과:

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

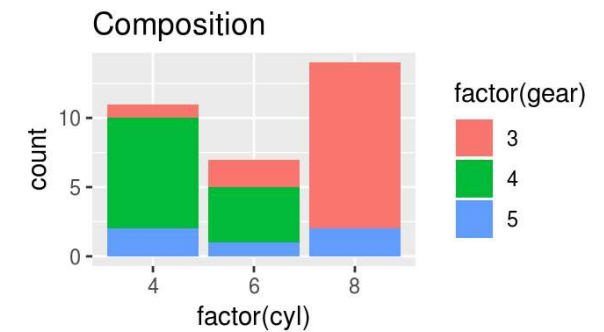
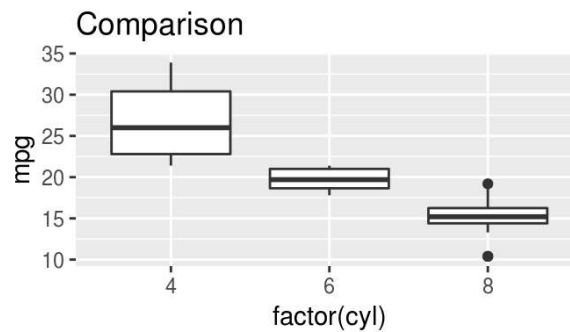
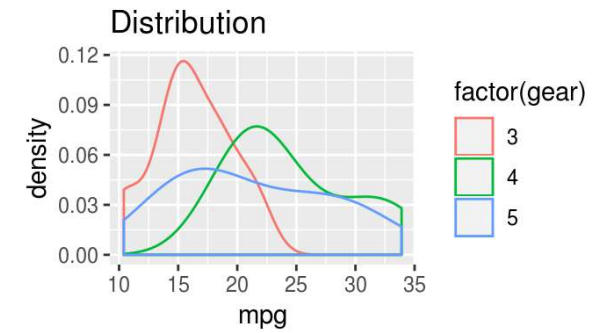
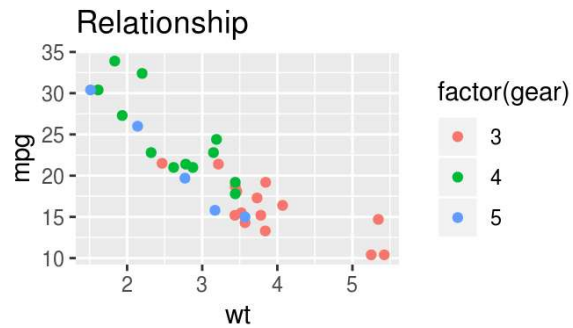
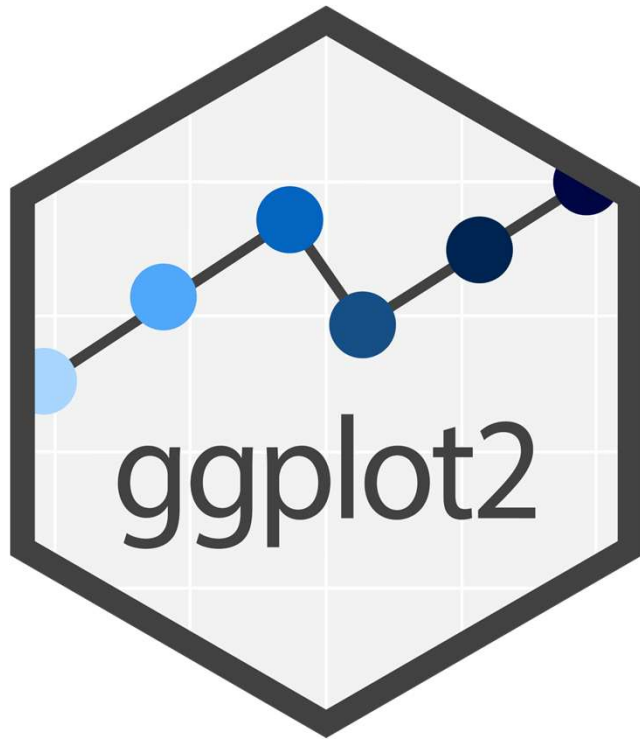
예제 2: `which()` 함수의 이용

```
slice_vector <- c(1,2,3,4,5,6,7,8,9,10)
slice_vector[which(slice_vector %% 3 == 0)] # 3의 배수만 출력
```

결과:

```
## [1] 3 6 9
```

-Data visualization



```
library(ggplot2)  
input: data.frame
```

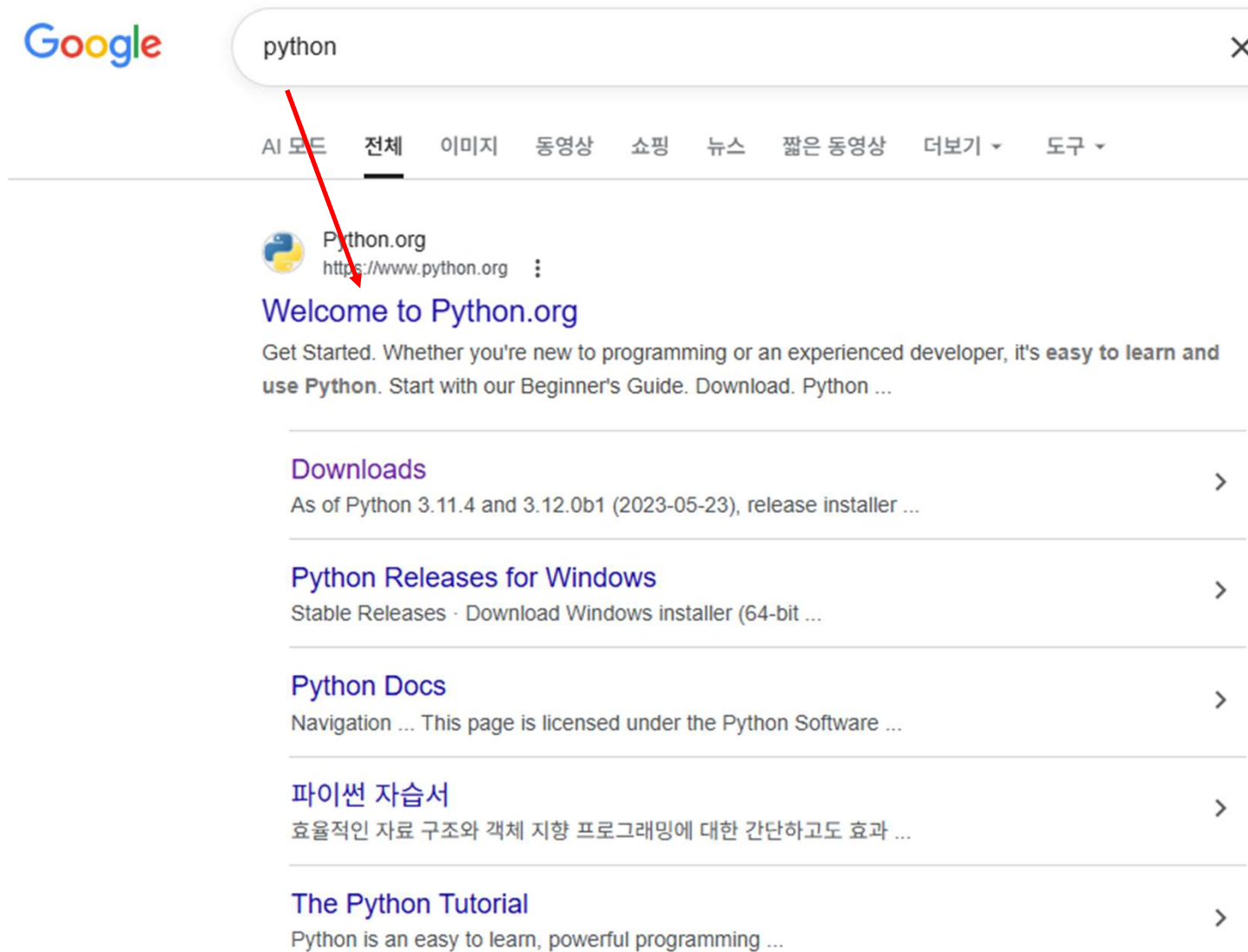





```
membershipoperator.py > ...
1  str="Hello world"
2  print("H" in str)
3  print('s' in str)
4  print("a" not in str)

PS F:\Python\operators> & C:/Users/Lenovo/AppData/Local/Programs
True
False
True
PS F:\Python\operators>
```

Python 설치



A screenshot of a Google search for 'python'. The search bar at the top contains the word 'python' with a red arrow pointing to it. Below the search bar, the first search result is for 'Python.org', with the URL 'https://www.python.org' and a three-dot menu icon. The result title is 'Welcome to Python.org'. Below the title, there is a paragraph: 'Get Started. Whether you're new to programming or an experienced developer, it's **easy to learn and use Python**. Start with our Beginner's Guide. Download. Python ...'. Below this paragraph, there are five links, each with a right-pointing chevron icon:

- Downloads**: As of Python 3.11.4 and 3.12.0b1 (2023-05-23), release installer ...
- Python Releases for Windows**: Stable Releases · Download Windows installer (64-bit ...
- Python Docs**: Navigation ... This page is licensed under the Python Software ...
- 파이썬 자습서**: 효율적인 자료 구조와 객체 지향 프로그래밍에 대한 간단하고도 효과 ...
- The Python Tutorial**: Python is an easy to learn, powerful programming ...

[Donate](#)[GO](#)[Socialize](#)[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

```
# Python 3: Fibonacci sequence
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

[All releases](#)[Source code](#)[Windows](#)[macOS](#)[Android](#)[Other Platforms](#)[License](#)[Alternative Implementations](#)

Download for Windows

[Python install manager](#)

Or get the standalone installer for

[Python 3.14.5](#)

Not the OS you are looking for? Python can be used on many operating systems and environments. [View the full list of downloads.](#)

and integrate systems more effectively. [>>> Learn More](#)

Python Releases for macOS

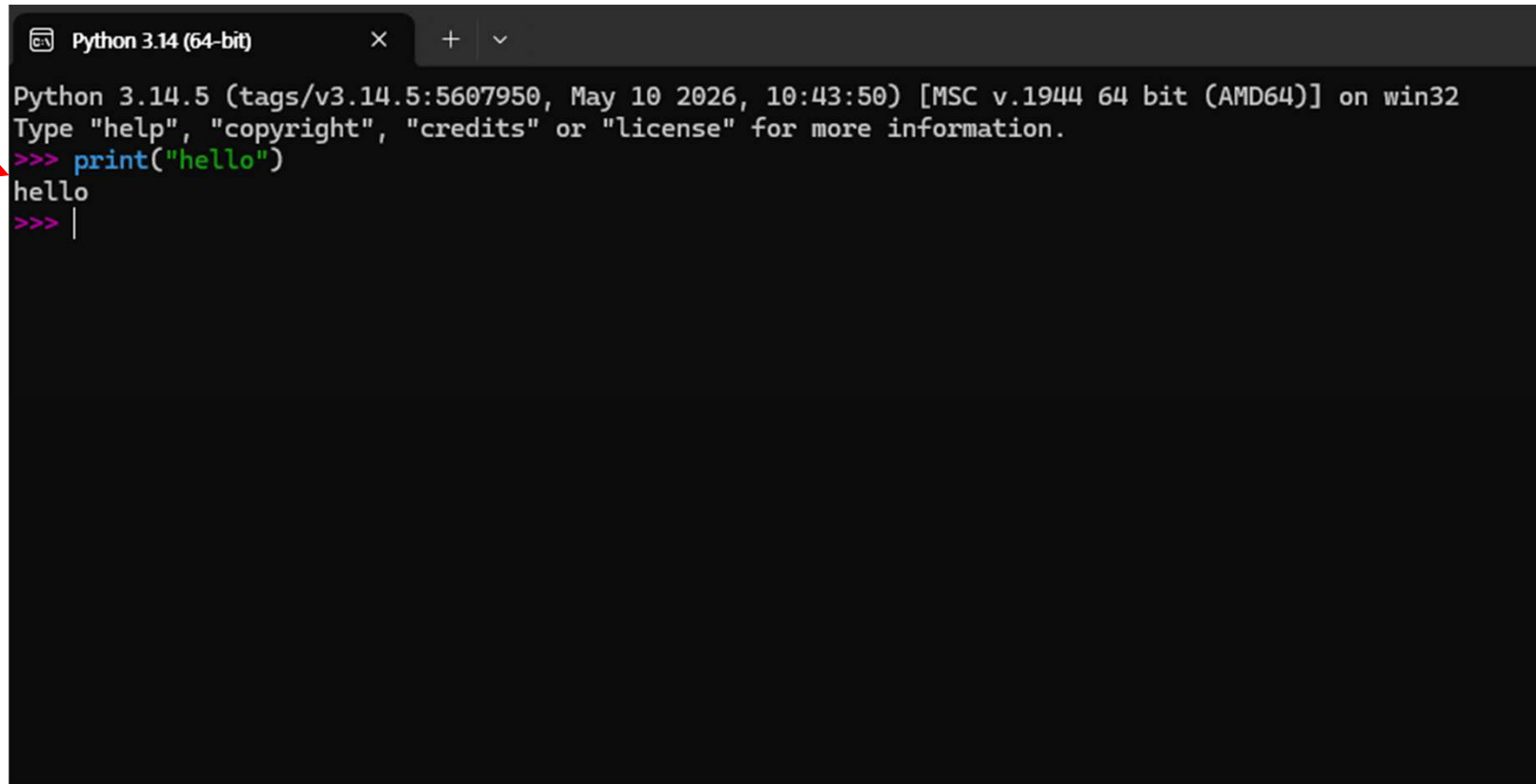
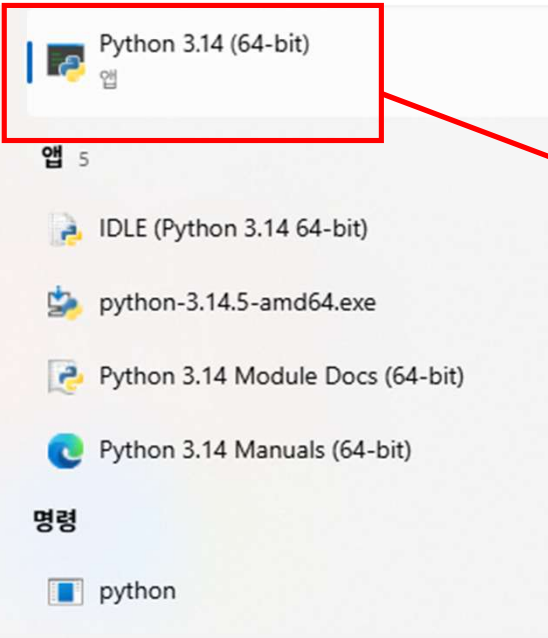
- [Latest Python 3 Release - Python 3.14.5](#)

Stable Releases

- [Python 3.14.5 - May 10, 2026](#)
 - [Download macOS installer](#)
- [Python 3.13.13 - April 7, 2026](#)
 - [Download macOS installer](#)
- [Python 3.14.4 - April 7, 2026](#)

Pre-releases

- [Python 3.15.0b1 - May 7, 2026](#)
 - [Download macOS installer](#)
- [Python 3.14.5rc1 - May 4, 2026](#)
 - [Download macOS installer](#)
- [Python 3.15.0a8 - April 7, 2026](#)



```
# this is the first comment
spam = 1 # and this is the second comment
        # ... and now a third!
text = "# This is not a comment because it's inside quotes."
```

-Numeric calculation

2 + 2

50 - 5*6

(50 - 5*6) / 4

8 / 5 # division always returns a floating-point number

17 / 3 # classic division returns a float

17 // 3 # floor division discards the fractional part

17 % 3 # the % operator returns the remainder of the division

5 * 3 + 2 # floored quotient * divisor + remainder

`5 ** 2 # 5 squared`

`2 ** 7 # 2 to the power of 7`

```
width = 20  
height = 5 * 9  
width * height
```

Cautious for “float”, “int”, ...

-Text

'spam eggs' # single quotes

"Paris rabbit got your back :)! Yay!" # double quotes

'1975' # digits and numerals enclosed in quotes are also strings

'doesn't' # use \ to escape the single quote...

"doesn't" # ...or use double quotes instead

"""Yes," they said."

\"Yes,\" they said."

\"Isn't,\" they said."

s = 'First line.\nSecond line.' # \n means newline

s # without print(), special characters are included in the string

print(s) # with print(), special characters are interpreted, so \n produces new line

```
print('C:\this\name') # here \t means tab, \n means newline
```

```
print(r'C:\this\name') # note the r before the quote
```

```
print("""\nUsage: thingy [OPTIONS]\n    -h                Display this usage message\n    -H hostname       Hostname to connect to\n""")
```

```
# 3 times 'un', followed by 'ium'\n3 * 'un' + 'ium'
```

```
text = ('Put several strings within parentheses '\n        'to have them joined together.')\ntext
```

```
word = 'Python'  
word[0] # character in position 0
```

```
word[5] # character in position 5
```

```
word[-1] # last character
```

```
word[-2] # second-last character
```

```
word[-6]
```

```
word[0:2] # characters from position 0 (included) to 2 (excluded)
```

```
word[2:5] # characters from position 2 (included) to 5 (excluded)
```

```
squares = [1, 4, 9, 16, 25]  
squares
```

-List

```
squares[0] # indexing returns the item
```

```
squares[-1]
```

```
squares[-3:] # slicing returns a new list
```

```
squares + [36, 49, 64, 81, 100]
```

```
cubes.append(216) # add the cube of 6  
cubes.append(7 ** 3) # and the cube of 7  
cubes
```

```
rgb = ["Red", "Green", "Blue"]  
rgba = rgb  
id(rgb) == id(rgba) # they reference the same object
```

```
rgba.append("Alph")  
rgb
```

```
correct_rgba = rgba[:]  
correct_rgba[-1] = "Alpha"  
correct_rgba
```

```
rgba
```

```
letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
```

```
letters
```

```
# replace some values
```

```
letters[2:5] = ['C', 'D', 'E']
```

```
letters
```

```
# now remove them
```

```
letters[2:5] = []
```

```
letters
```

```
# clear the list by replacing all the elements with an empty list
```

```
letters[:] = []
```

```
letters
```

```
letters = ['a', 'b', 'c', 'd']
```

```
len(letters)
```

-Data visualization

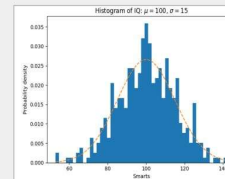
matplotlib

```
python -m pip install -U matplotlib
import matplotlib.pyplot as plt
```

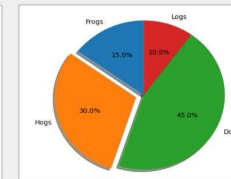
Matplotlib: Visualization with Python

laboputer.github.io

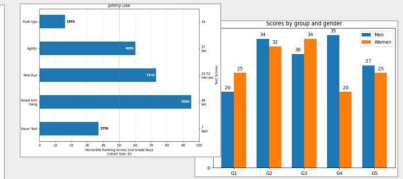
1 Dimension



Histogram

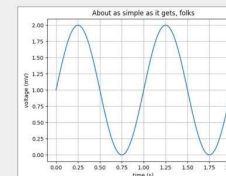


Pie Chart

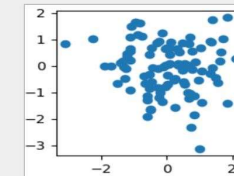


Bar Chart

2 Dimension

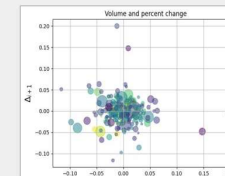


Line Plot

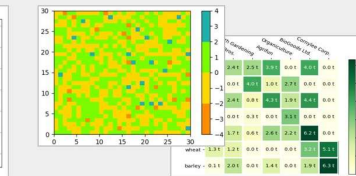


Scatter Chart

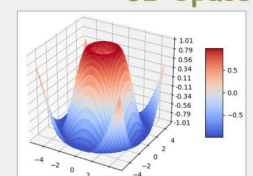
3 Dimension



Bubble Chart



Color Map



Surface Plot

3D Space

<https://wikidocs.net/book/5011>

What is ChatGPT?



Gemini



ChatGPT ▾

어디서부터 시작할까요?

+ 무엇이든 물어보세요



음성

R에서
boxplot 어떻게 그리냐?



R에서 `boxplot()` 함수 쓰면 됩니다.

가장 기본 형태:

```
<> R  
  
data <- c(1, 2, 3, 4, 5, 100)  
  
boxplot(data)
```



여러 그룹 비교:

```
<> R  
  
groupA <- c(1,2,3,4,5)  
groupB <- c(2,3,4,5,6)  
  
boxplot(groupA, groupB,  
        names=c("A", "B"),  
        col=c("skyblue", "pink"))
```



데이터프레임에서 변수 기준으로 그리기:

<> R



```
df <- data.frame(  
  score = c(80, 85, 90, 70, 95, 88),  
  class = c("A", "A", "B", "B", "A", "B")  
)  
  
boxplot(score ~ class, data=df)
```

score ~ class 의미:

- score: 숫자값
- class: 그룹 기준

옵션 자주 쓰는 것들:

<> R



```
boxplot(score ~ class,  
  data=df,  
  main="클래스별 점수",  
  xlab="Class",  
  ylab="Score",  
  col="lightgreen",  
  horizontal=FALSE)
```



↗ R

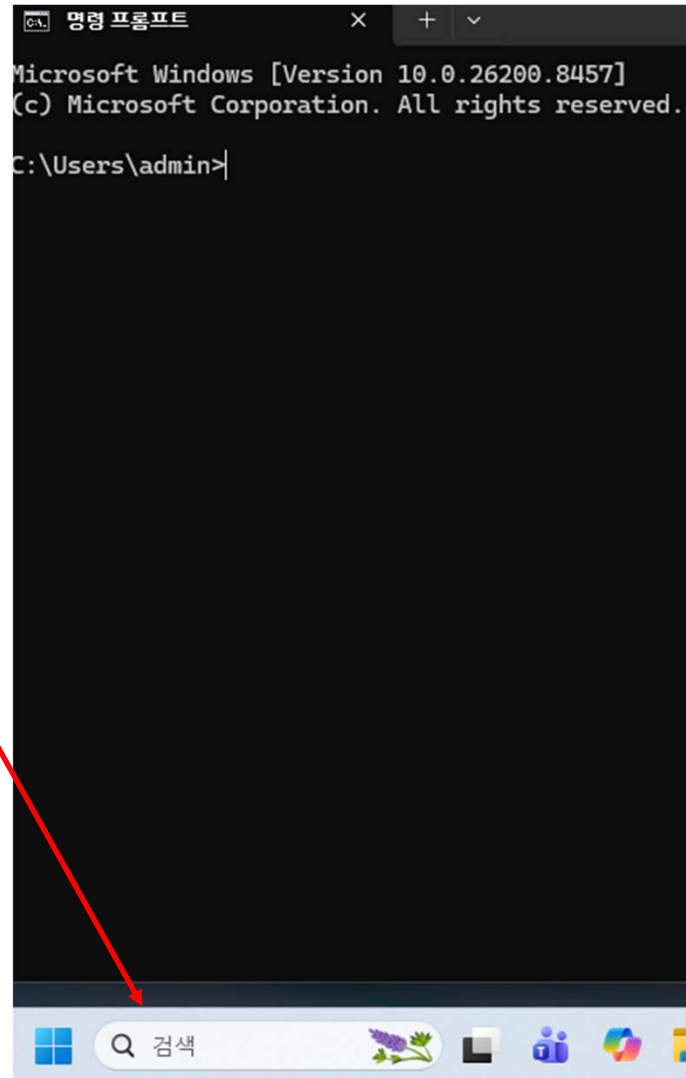


```
library(ggplot2)
```

```
ggplot(df, aes(x=class, y=score)) +  
  geom_boxplot(fill="skyblue")
```


-Module installation

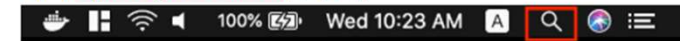
cmd
pip install numpy
pip install pandas



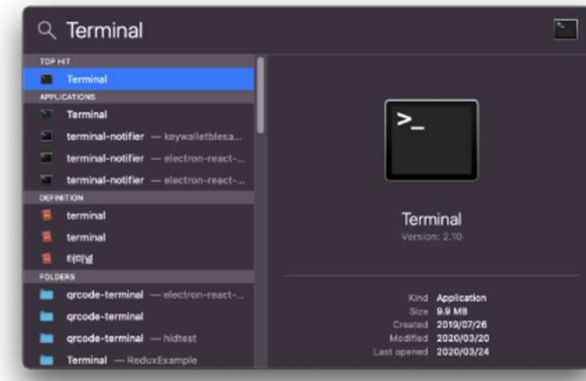
Mac:

1) Spotlight Search 사용하기

(1) 메뉴바에서 돋보기 모양의 아이콘을 클릭하거나 또는 단축키 **Command + Space** 를 입력하여 **Spotlight Search** 를 연다.



(2) "Teminal" 또는 "터미널"을 입력한다.



Python에서는 보통 `matplotlib` 또는 `seaborn` 으로 boxplot을 그립니다.

1. matplotlib 기본 boxplot

</> Python



```
import matplotlib.pyplot as plt

data = [1, 2, 3, 4, 5, 100]

plt.boxplot(data)
plt.title("Boxplot")
plt.show()
```

2. 여러 그룹 비교

</> Python



```
import matplotlib.pyplot as plt

groupA = [1,2,3,4,5]
groupB = [2,3,4,5,6]

plt.boxplot([groupA, groupB],
            labels=["A", "B"])

plt.show()
```