

강원대학교
AI소프트웨어학과

재난안전 프로그래밍

Chapter 1. R 소개 및 설치



Search or jump to...



[Pull requests](#) [Issues](#) [Codespaces](#) [Marketplace](#) [Explore](#)



[Overview](#) [Repositories](#) 7 [Projects](#) [Packages](#) [Stars](#) 1



Kimchanggyun

Edit profile

0 followers · 1 following

Organizations



Popular repositories

Customize your pins

[MLPB](#)

Forked from ben519/MLPB

Machine Learning Problem Bible | Problem Set Here >>

R

Public

[facerecognition](#)

Forked from bwcho75/facerecognition

Jupyter Notebook

Public

[Tensorflow-101](#)

Forked from sjchoi86/Tensorflow-101

TensorFlow Tutorials

Jupyter Notebook

Public

[models](#)

Forked from tensorflow/models

Models and examples built with TensorFlow

Python

Public

[soynlp](#)

Forked from lovit/soynlp

한국어 자연어처리를 위한 파이썬 라이브러리입니다. 단어 추출/ 토큰나이저 / 품사판별/ 전처리의 기능을 제공합니다.

Jupyter Notebook

Public

3 contributions in the last year

Contribution settings

Overview Repositories 7 Projects Packages Stars 1

Popular repositories

MLPB

Forked from ben519/MLPB

Machine Learning Problem Bible | Problem Set Here >>

R

Public

facerecognition

Forked from bwcho75/facerecognition

Jupyter Notebook

Public

Customize your pins

Kimchanggyun

Type to search

Overview Repositories 8 Projects Packages Stars 1

Popular repositories

MLPB

Forked from ben519/MLPB

Machine Learning Problem Bible | Problem Set Here >>

R

Public

facerecognition

Forked from bwcho75/facerecognition

Jupyter Notebook

Customize your pins

New repository

Import repository

New codespace

New gist

New organization

New project

3 contributions in the last year


Contribution settings

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 Kimchanggyun

 /

Repository name *


R

✔ R is available.


Great repository names are short and memorable. Need inspiration? How about **laughing-garbanzo** ?

Description (optional)

- ☒



Public
Anyone on the internet can see this repository. You choose who can commit.
- ☐



Private
You choose who can see and commit to this repository.

Initialize this repository with:

- ☒

Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: None


Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

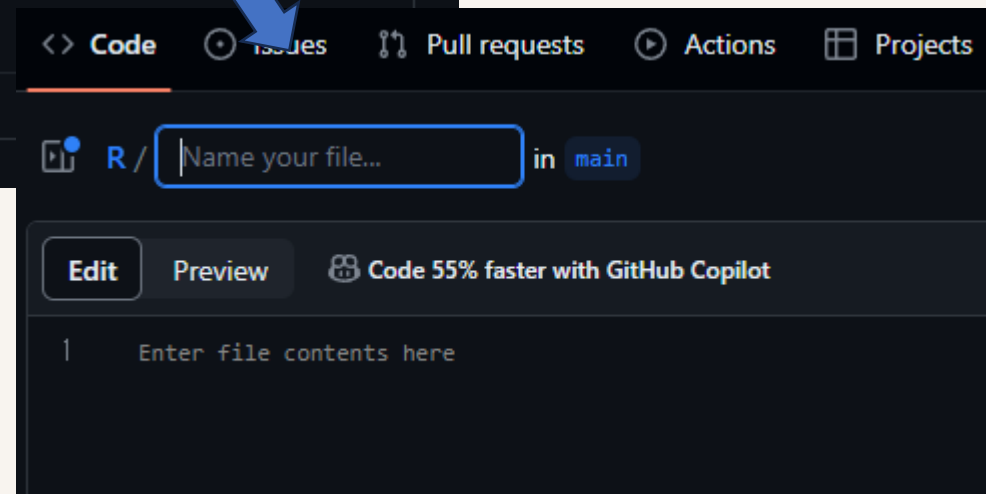
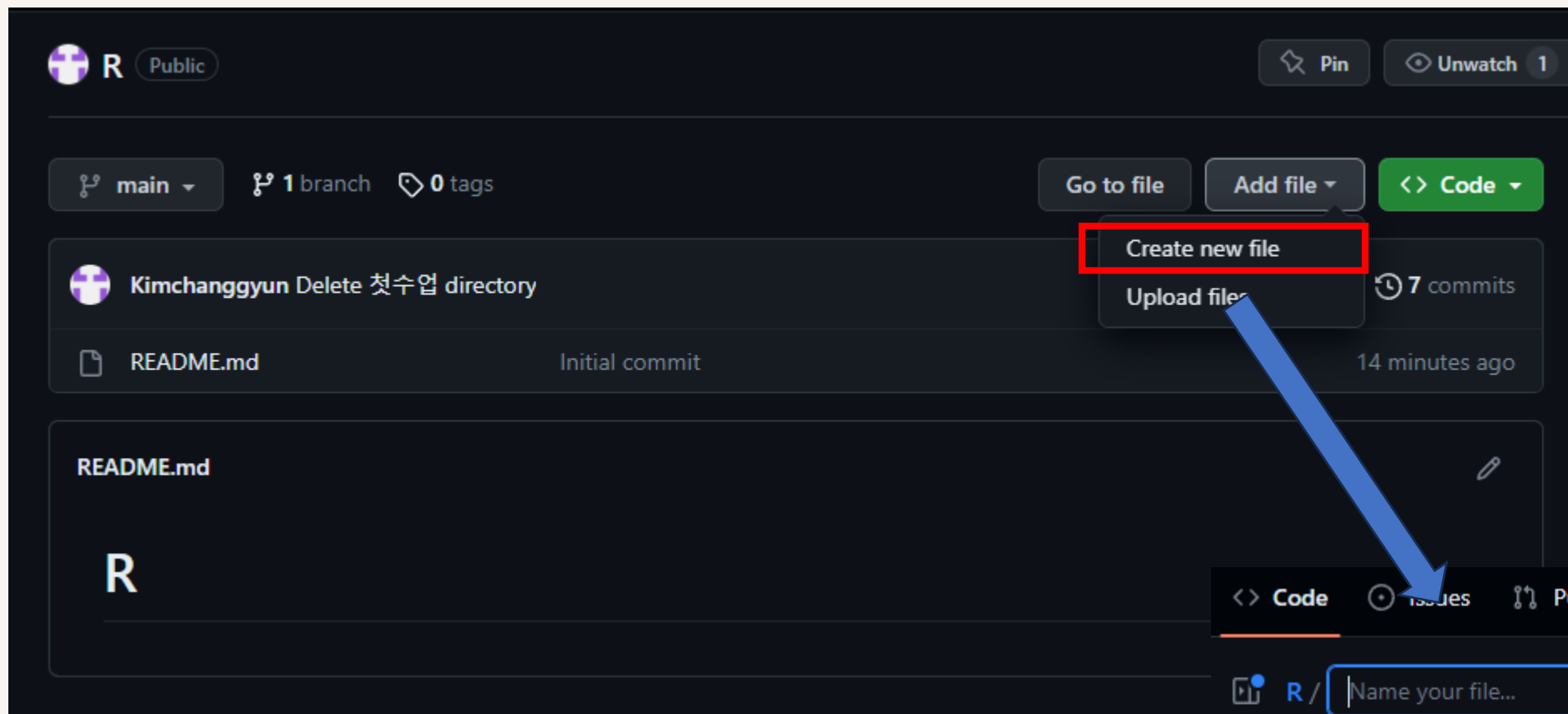
License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).


 You are creating a public repository in your personal account.

Create repository




Enhance your workflow with extensions


Tools from the community and partners to simplify tasks and automate processes

 Search for Copilot extensions, apps, actions, and models




 Featured

 Copilot

 Models

 Apps

 Actions

+ Create a new extension

Models

Try, test, and deploy from a wide range of model types, sizes, and specializations. [Learn more.](#)

Beta

[Give feedback](#)

By: All providers

Capability: All

Tag: All



AI21 Jamba 1.5 Large

Model

A 398B parameters (94B active) multilingual model, offering a 256K long context window, function calling, structured output, and grounded...



AI21 Jamba 1.5 Mini

Model

A 52B parameters (12B active) multilingual model, offering a 256K long context window, function calling, structured output, and grounded...



AI21-Jamba-Instruct

Model

A production-grade Mamba-based LLM model to achieve best-in-class performance, quality, and cost efficiency.



Cohere Command R

Model

Command R is a scalable generative model targeting RAG and Tool Use to enable production-scale AI for enterprise.



Cohere Command R+

Model

Command R+ is a state-of-the-art RAG-optimized model designed to tackle enterprise-grade workloads.



Cohere Embed v3 English

Model

Cohere Embed English is the market's leading text representation model used for semantic search, retrieval-augmented generation (RAG),...

Danger Zone

Change repository visibility
This repository is currently public.

Change visibility

Transfer ownership
Transfer this repository to another user or to an organization where you have the ability to create repositories.



Transfer


Archive this repository
Mark this repository as archived and read-only.


Archive this repository


Delete this repository
Once you delete a repository, there is no going back. Please be certain.


Delete this repository


susanli2016 / Data-Analysis-with-R


Type  to search

+














<> Code


 Issues 3


 Pull requests



 Actions



 Projects



 Security



 Insights


 Data-Analysis-with-R Public


 Watch 47 


 Fork 370 

 Star 380 


 master 

 1 Branch


 0 Tags


Go to file 

+



<> Code 


About

 susanli2016 Add data


d8fbc5b · 6 years ago  76 Commits


Using ggplot2, tidyr, dplyr, ggmap, choroplethr, shiny, logistic regression, clustering models and more


susanli2016 / Data-Analysis-with-R


Type  to search


<> Code


 Issues 3

 Pull requests

 Actions

 Projects


 Security


 Insights

Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).


Owner *
 Kimchanggyun

Repository name *
Data-Analysis-with-R
 Data-Analysis-with-R is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)
Using ggplot2, tidyr, dplyr, ggmap, choroplethr, shiny, logistic regression, clustering models and more

☒ Copy the master branch only
Contribute back to susanli2016/Data-Analysis-with-R by adding your own branch. [Learn more.](#)

 You are creating a fork in your personal account.

Create fork

Chapter 1. R 소개 및 설치

1.1 R 소개



- R은 AT&T Bell Laboratory 에서 만든 S 언어를 모태로 하여 1996년에 Ross Ihaka와 Robert Gentleman이 만들었다
- R은 객체지향언어와 함수형 언어의 특징을 가지고 있다
- R은 GNU General Public License에 의해 무료로 배포된다
- R은 기본적인 통계계산과 그래픽 기능이 내장되어 있으며 사용자들이 만든 다양한 패키지를 자유롭게 이용할 수 있다

Chapter 1. R 소개 및 설치


1.1 R 소개

- R은 통합 개발 환경은 오픈소스 프로그램으로 누구나 설치해서 사용할 수 있음
- R과 R Studio는 계속 발전하며 업그레이드 되고 있음
- 딥러닝 기반 분석이 가능하도록 지속적인 업데이트

구분	설명
R	오픈소스 프로그램으로 통계/데이터 마이닝 및 그래프를 위한 프로그래밍 언어 또는 R 통합 개발 환경
R Studio	R 프로그램을 편리하게 사용할 수 있도록 만든 통합 개발 환경

Chapter 1. R 소개 및 설치

1.2 R 설치하기



Don't want to download or install anything? Get started with RStudio on [Posit Cloud for free](#). If you're a professional data scientist looking to download RStudio and also need common enterprise features, don't hesitate to [book a call with us](#).

1: Install R


RStudio requires R 3.3.0+. Choose a version of R that matches your computer's operating system.

DOWNLOAD AND INSTALL R

2: Install RStudio

DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS

Size: 212.78 MB | [SHA-256: BCF6B866](#) | Version: 2023.06.2+561 | Released: 2023-08-30



CRAN
[Mirrors](#)
[What's new?](#)
[Search](#)
[CRAN Team](#)

About R
[R Homepage](#)
[The R Journal](#)

Software
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Task Views](#)
[Other](#)

Documentation
[Manuals](#)
[FAQs](#)
[Contributed](#)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed package

- [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 packa

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binary before you can use them. If you do not know what this means, you probably

- The latest release (2023-06-16, Beagle Scouts) [R-4.3.1.tar.gz](#), read [what's](#)
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in ti
- Daily snapshots of current patched and development versions are [availa](#) feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the soft [questions](#) before you send an email.

① R 홈페이지 접속

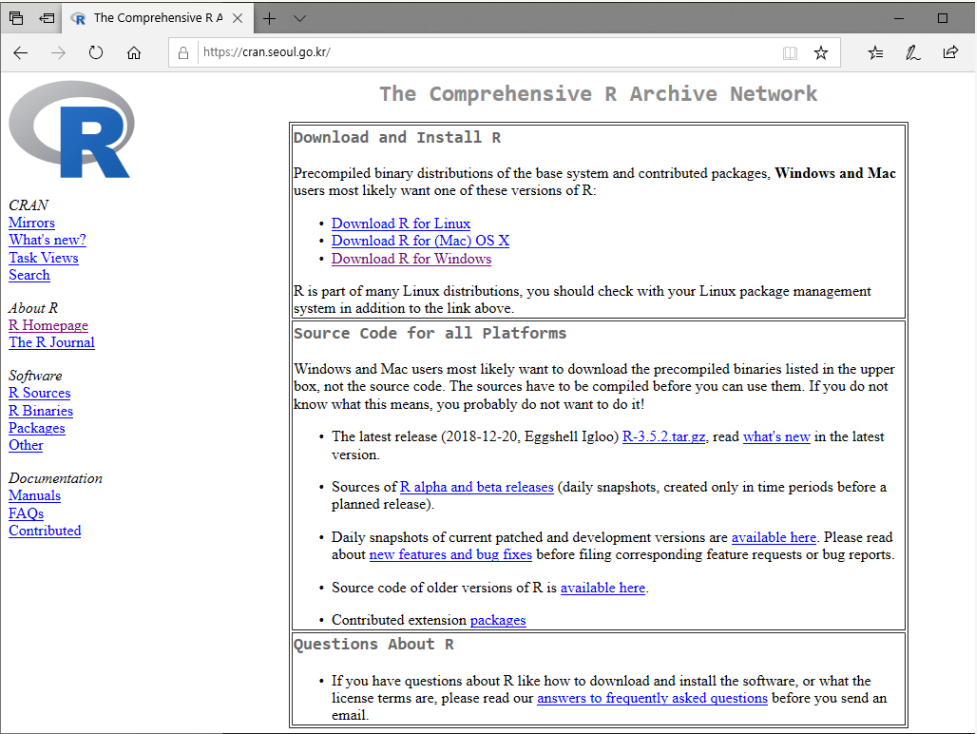
② <https://posit.co/download/rstudio-desktop/>

③ Download 메뉴의 CRAN 클릭

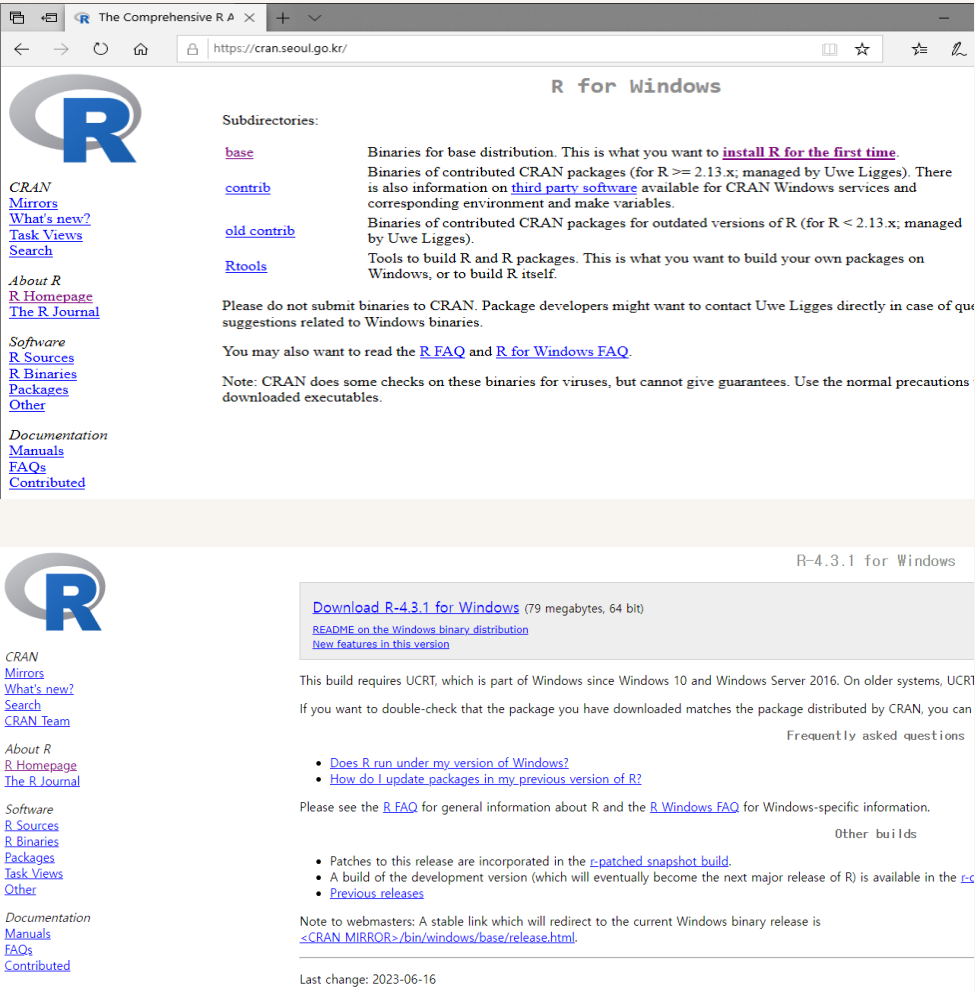
③ 해당하는 국가(Korea)의 링크 클릭

Chapter 1. R 소개 및 설치

1.2 R 설치하기



④ Download R for Windows 클릭

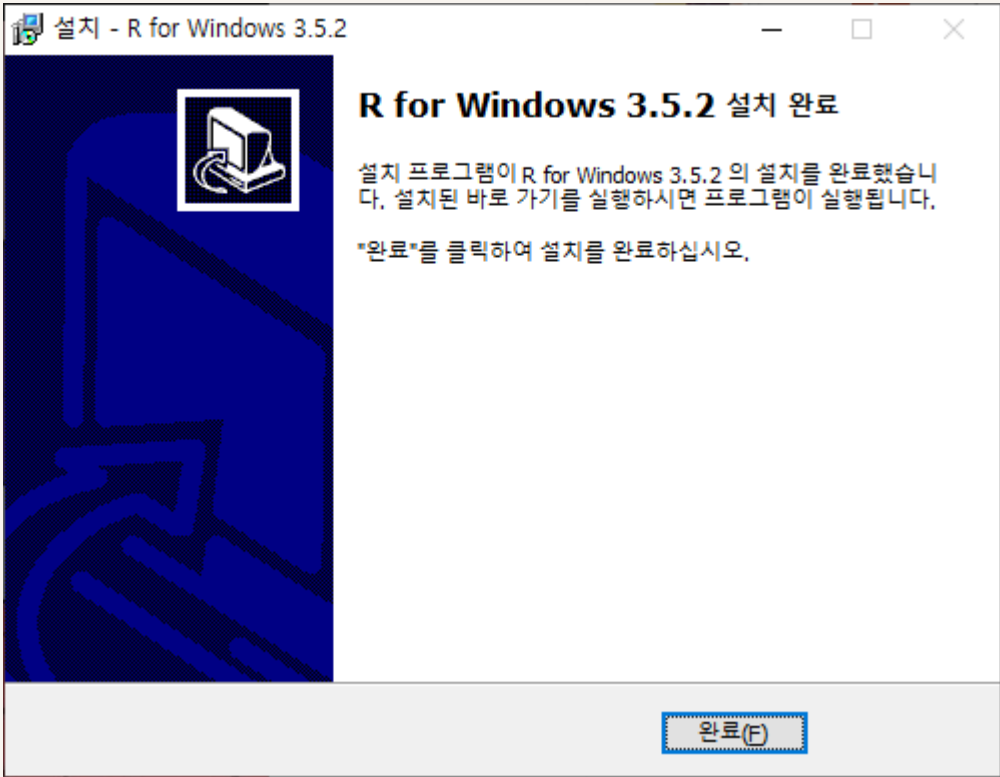


⑤ 'base' 클릭

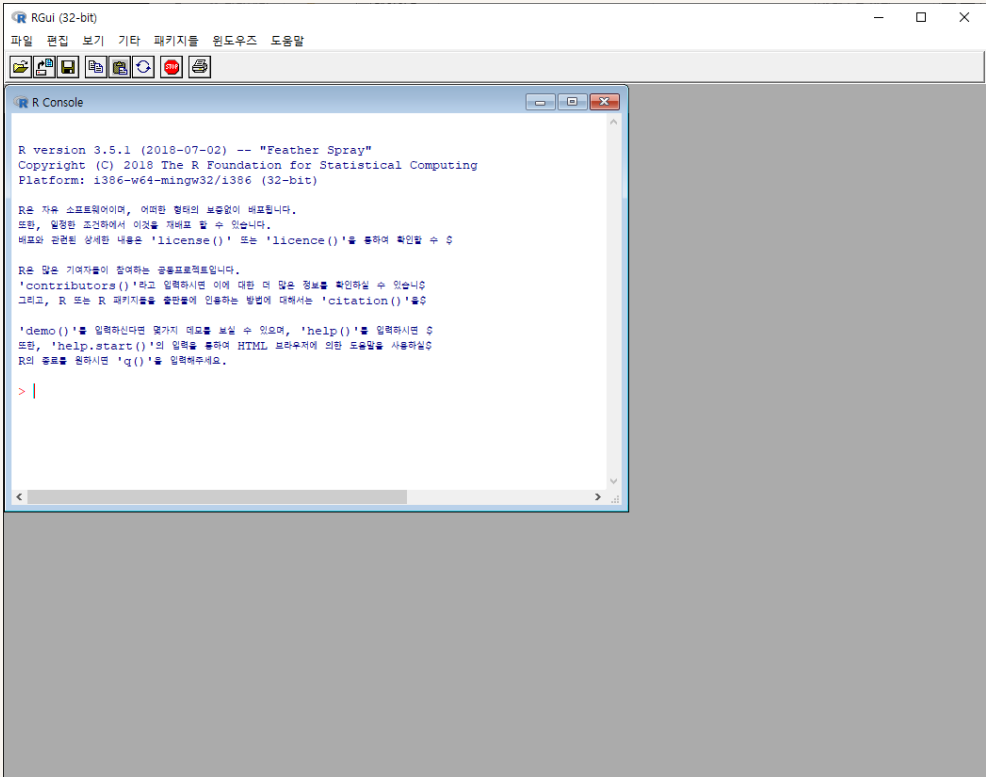
⑥ Download R 3.5.2 for Window를 클릭

Chapter 1. R 소개 및 설치

1.2 R 설치하기



⑦ R 설치완료

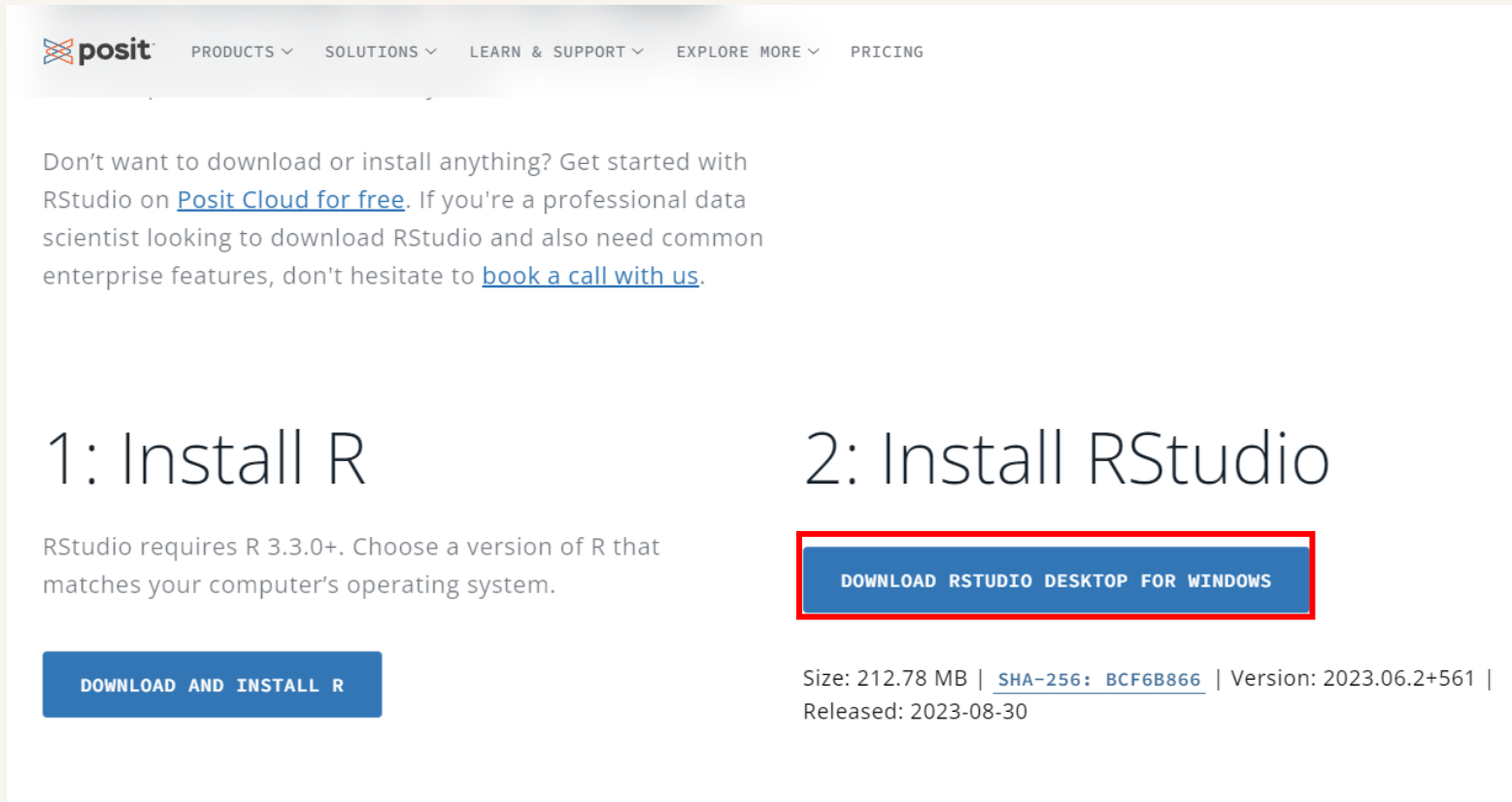


⑧ R 실행해보기

Chapter 1. R 소개 및 설치

1.3 R - Studio 설치하기

※ R Studio는 R을 설치하지 않으면 실행 되지 않음



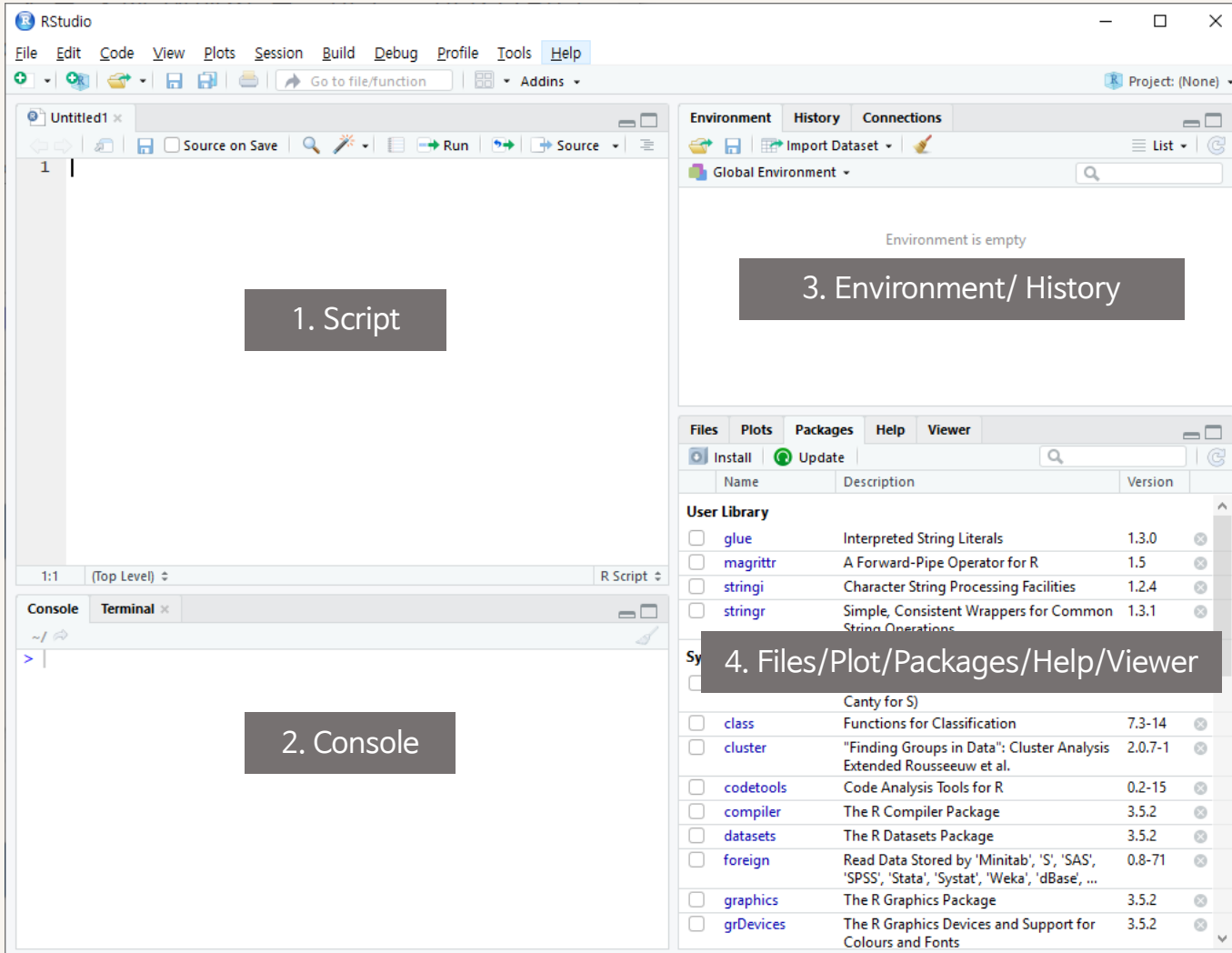
① R Studio 홈페이지 접속

<https://posit.co/products/open-source/rstudio/>

‘Download Rstudio’ 클릭

Chapter 1. R 소개 및 설치

1.3 R - Studio 설치하기



1. Script 창

- Batch 모드로 R Script를 작성하고 실행 가능
- 'Ctrl + R' or 'Ctrl + Enter' 로 실행

2. Console 창

- R Script 창 or Console 창에서 작성한 프로그램의 결과 볼 수 있음
- 패키지 설치, 오류 메시지 볼 수 있음

3. Environment/ History 창

- 데이터 셋의 이름, 관측치 개수, 변수 개수 볼 수 있음

4. Files/Plot/Packages/Help/Viewer 창

- 패키지, 디렉토리 및 파일 관리
- 이미지 뷰어

Chapter 1. R 소개 및 설치

1.3 R - Studio 설치하기

패키지 설치하고 로드하기

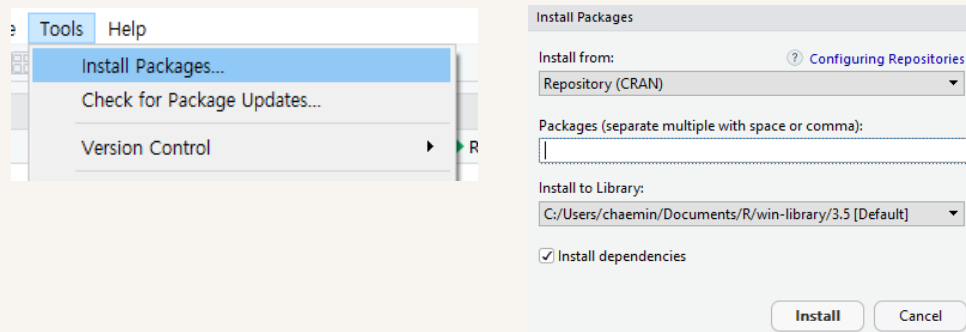
- R은 오픈소스로 수천개가 넘는 패키지를 가지고 있음
- `install.packages("패키지명")`을 쳐서 설치한 후 `library(패키지명)`으로 실행시켜주면 패키지 내의 함수를 사용할 수 있음
- `help(packages=)`를 통해 패키지를 검색할 수 있음

```
> install.packages("stringr")
Installing package into 'C:/Users/chaemin/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/stringr_1.4.0.zip'
Content type 'application/zip' length 214115 bytes (209 KB)
downloaded 209 KB

package 'stringr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\chaemin\AppData\Local\Temp\RtmpUTYf6N\downloaded_packages
> library(stringr)
```

- Rstudio에서는 설치 이후 한번만 설정하면 됨



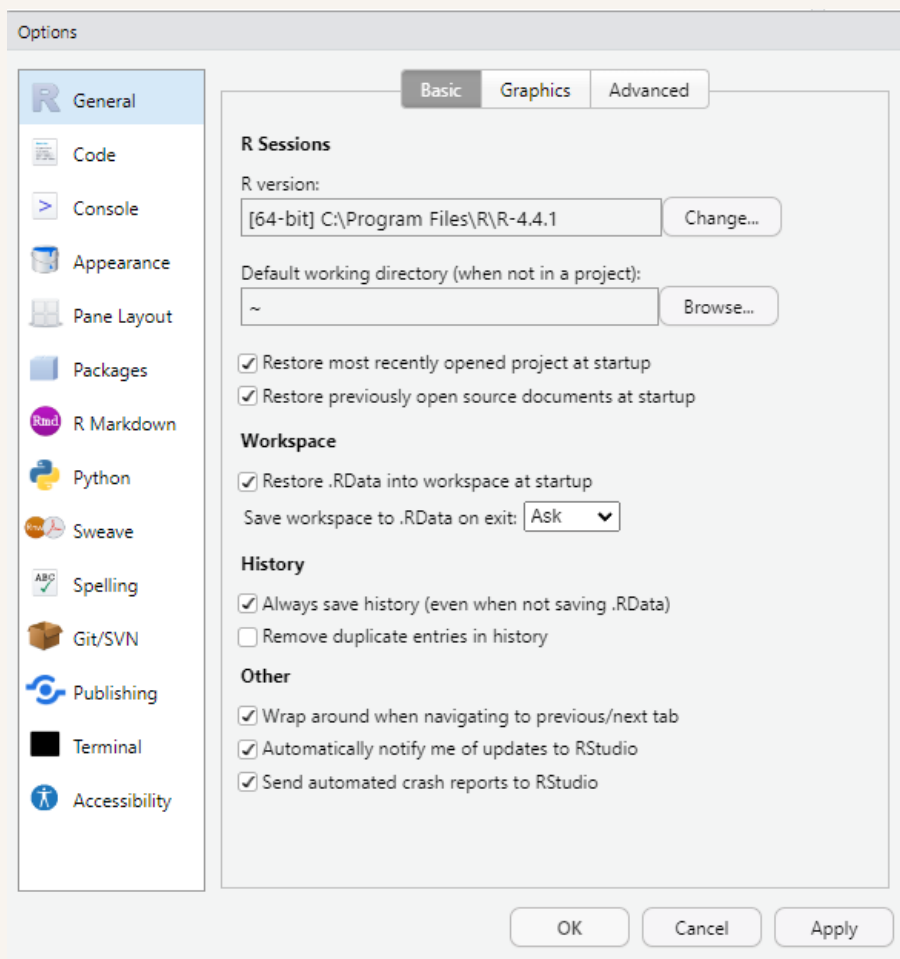
Tool → Install Packages → 원하는 패키지 명 입력 Install 버튼을 누름
패키지 설치하면 항상 Library 호출을 해줘야 함

Chapter 1. R 소개 및 설치

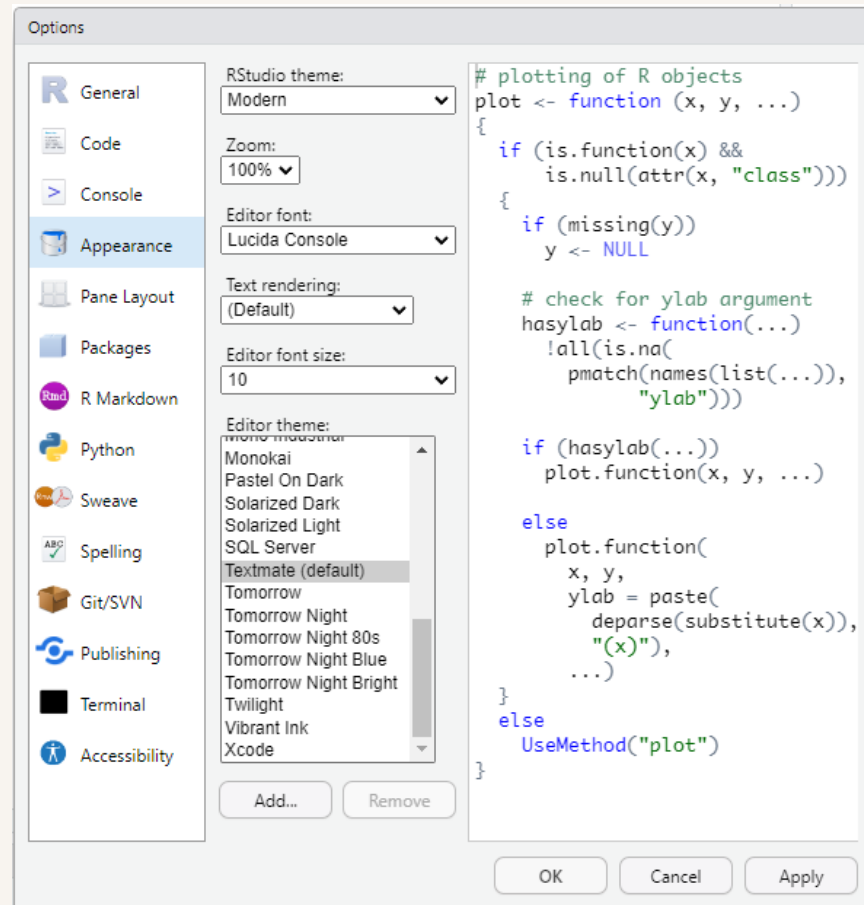
1.3 R - Studio 설치하기

패키지 설치하고 로드하기

- Tools → Global options에서 다양한 Option들을 조정 가능



- Appearance에서 글자 크기 및 폰트, 배경을 조정 가능

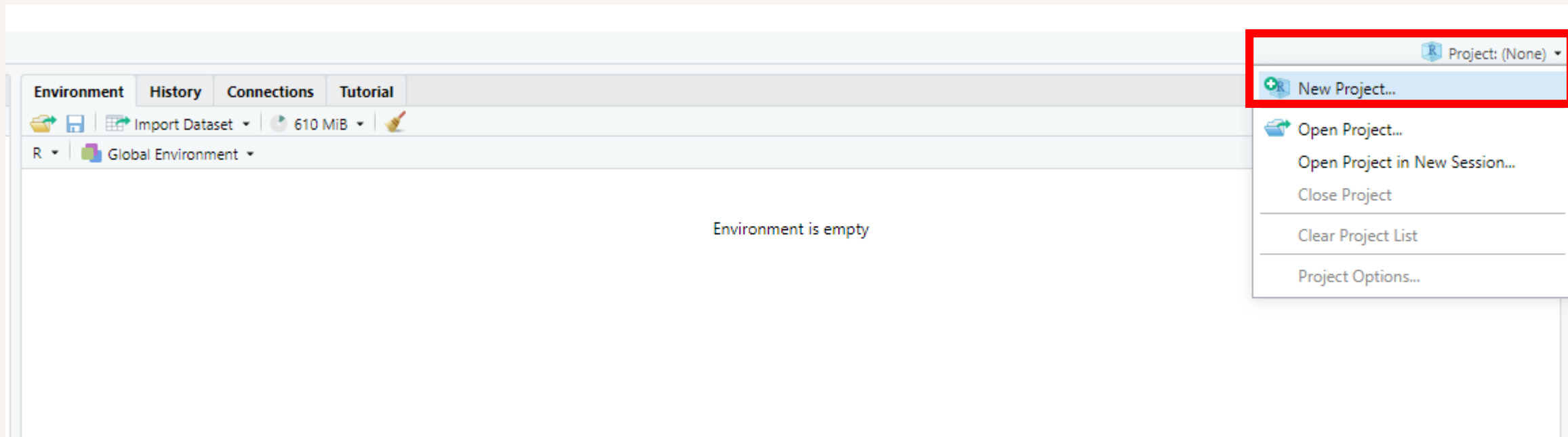


Chapter 1. R 소개 및 설치

1.4 R 활용

New Project 생성하기

- 단순 R 파일을 저장하게 되면 디렉토리, library와 같은 작업환경 고려 X
- Environment 환경에서 Project 클릭

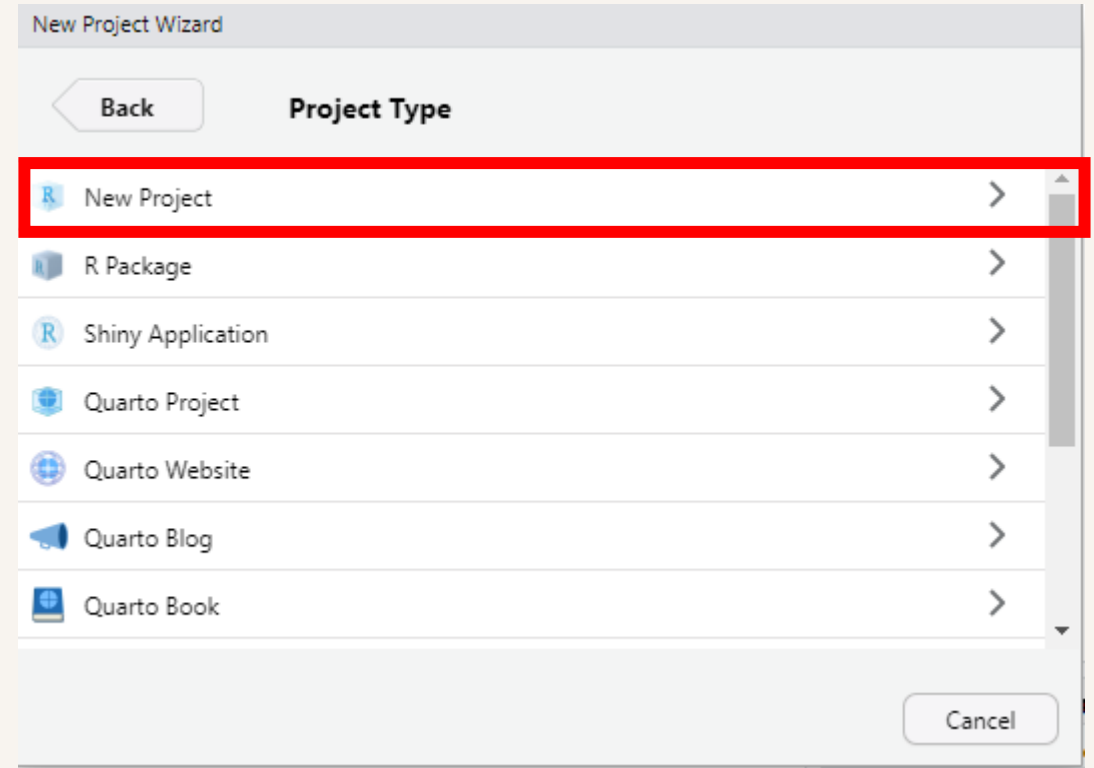
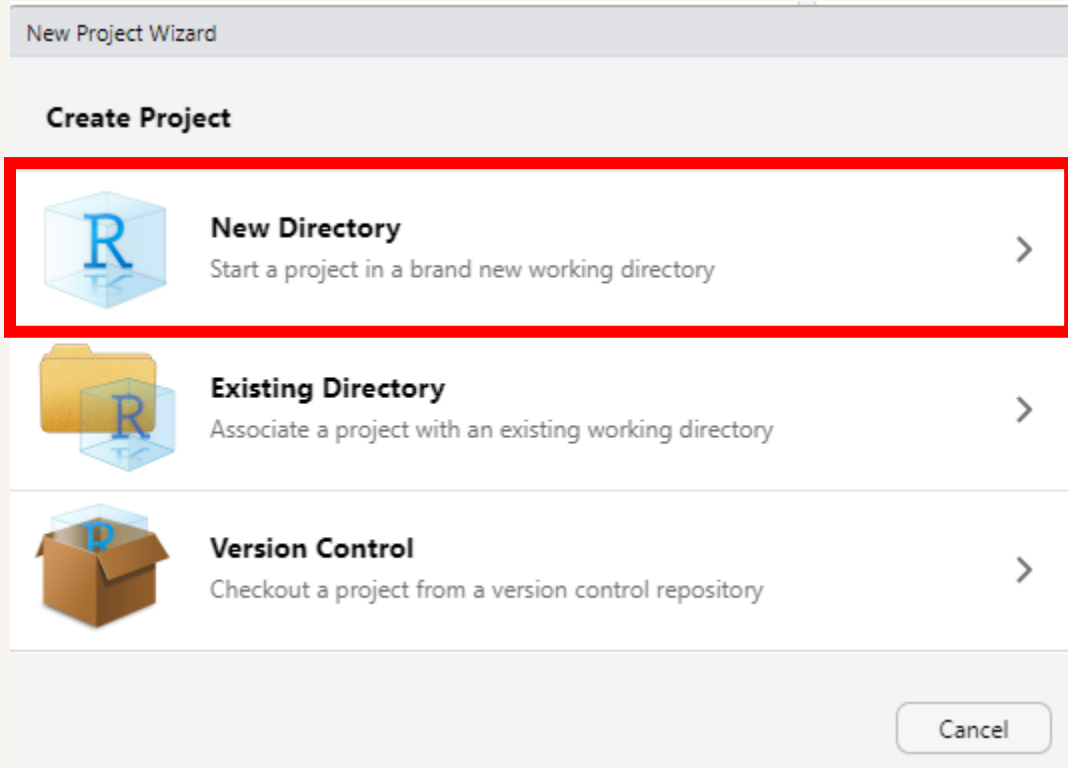


Chapter 1. R 소개 및 설치

1.4 R 활용

New Project 생성하기

- Environment 환경에서 Project 클릭

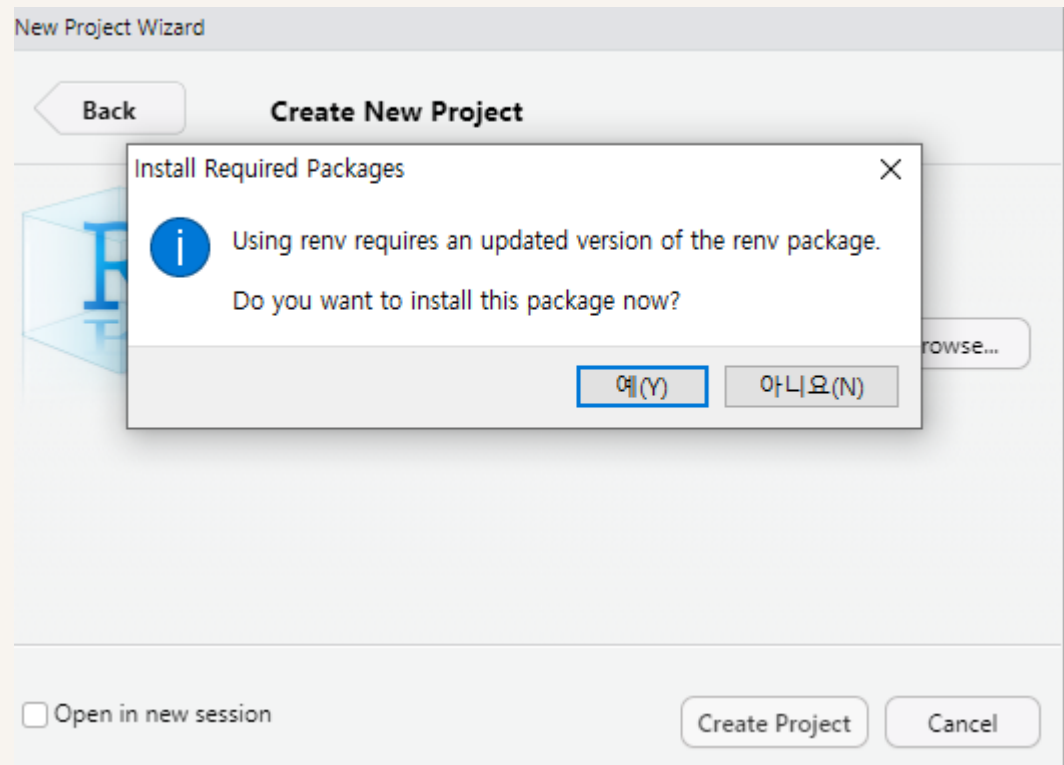
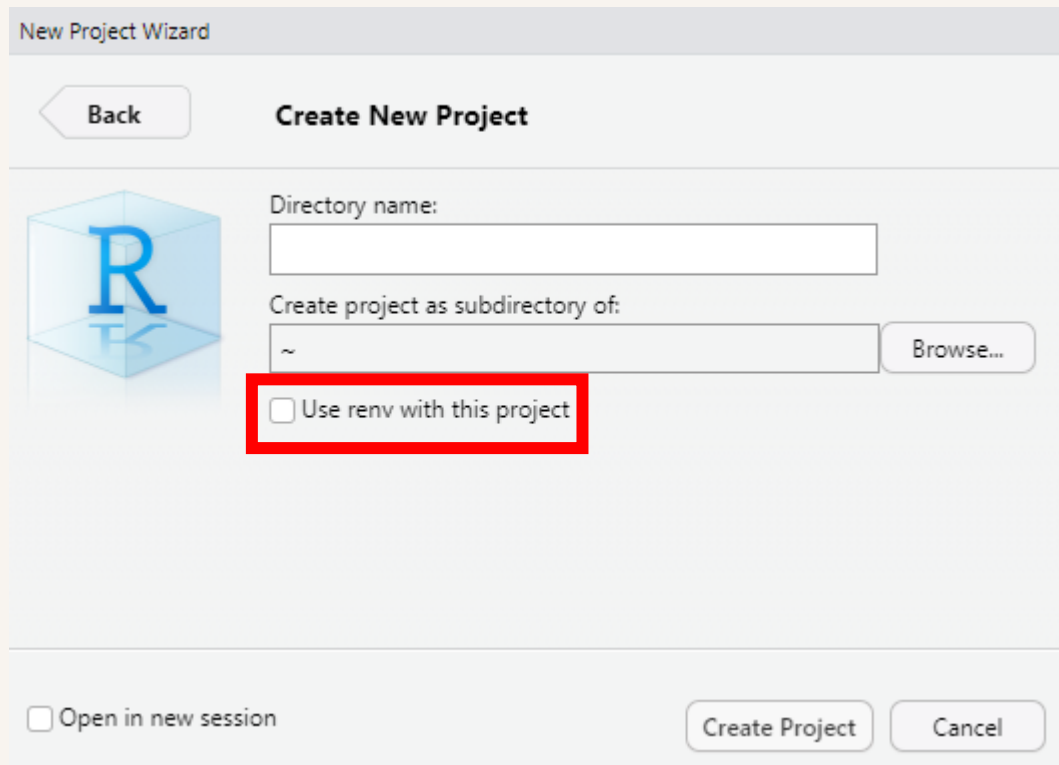


Chapter 1. R 소개 및 설치

1.4 R 활용

New Project 생성하기

- Environment 환경에서 Project 클릭



Chapter 1. R 소개 및 설치

1.4 R 활용

New Project 생성하기

- New Project 환경에서 패키지 설치하기

```
install.packages("installr")  
library(installr)
```

```
check.for.updates.R()  
#install.R()
```

Chapter 1. R 소개 및 설치

1.4 R 활용

- 할당연산자를 이용하여 변수 생성하기
 - R에서 객체(object)의 형태 name <- object(x) #특정 객체(object)를 name에 할당

연산자	설명	예시
<- = <<-	왼쪽에 있는 변수에 값을 할당 변수 <- 수치, 계산식, 함수, 데이터 등 * <-를 주로 사용	<pre>> x1 <- 2 > x2 <<- c(1,2,3) > x3 = c("a","b","c") > x1 [1] 2 > x2 [1] 1 2 3 > x3 [1] "a" "b" "c"</pre>
-> ->>	오른쪽에 있는 변수에 값을 할당 수치, 계산식, 함수, 데이터 등 -> 변수	<pre>> c("a","b","c") -> y1 > 2 ->> y2 > y1 [1] "a" "b" "c" > y2 [1] 2</pre>

- 변수 지정 특징
 - ① 변수를 지정할 때 대소문자를 명확히 구분해야 한다
 - ② 변수명에는 영문자, 숫자, 마침표(.), 언더라인(_)을 사용할 수 있다
 - ③ 변수명을 숫자나 숫자로 시작하는 이름으로 지정할 수 없다
 - ④ R 시스템에서 사용하는 이름은 변수명으로 사용할 수 없다 (예 : if, else, NULL, NA 등)

- ① ↑ 방향키로 이전에 했던 작업 수행 가능
 - ② q() 사용하면 R 종료
 - ③ #: 주석
 - ④ R studio 스크립트 창에서 코드 입력 후 해당 부분을 선택하여 Ctrl + Enter 로 실행

Chapter 1. R 소개 및 설치

1.4 R 활용

객체를 변수로 저장하고, 출력하는 방법

Script창에서 작성해 Console 창에 결과를 띄어 줄 수 있음 : Ctrl+Enter

```
a<-3
```

```
b=4
```

```
c=7
```

```
d=a+b
```

```
print(a)
```

```
print(d)
```

따옴표, 쌍따옴표 : ‘, “ 로 문자 처음과 끝

```
print(“Hello”)
```

Chapter 1. R 소개 및 설치

1.4 R 활용

Script창에서 작성해 Console 창에 결과를 띄어 줄 수 있음 : Ctrl+Enter

연산자	설명	예제	결과
+	더하기	3 + 2	5
-	빼기	3 - 2	1
*	곱하기	3 * 2	6
/	나누기	3 / 2	1.5
^ or **	승수	3 ^ 2	9
x %% y	X 를 y 로 나눈 나머지 값 반환	3 %% 2	1
x %/% y	나누기의 결과를 정수로	3 %/% 2	1

Chapter 1. R 소개 및 설치

1.4 R 활용

Script창에서 작성해 Console 창에 결과를 띄어 줄 수 있음 : Ctrl+Enter

	연산자	설명	예제	결과
비교 연산자	<	좌변이 보다 작은	5 < 5	FALSE
	<=	좌변 이하	5 <= 5	TRUE
	>	좌변이 보다 큰	5 > 5	FALSE
	>=	좌변 이상	5 >= 5	TRUE
	==	값이 같은	5 == 5	TRUE
	!=	값이 다른	5 != 5	FALSE
논리 연산자	!x	부정형 연산	!TRUE	FALSE
	x y	x OR y	TRUE FALSE	TRUE
	x & y	x AND y	TRUE & FALSE	FALSE
	isTRUE(x)	X 의 TRUE 여부조사	isTRUE(TRUE)	TRUE

Chapter 1. R 소개 및 설치

1.4 R 활용

Script창에서 작성해 Console 창에 결과를 띄어 줄 수 있음 : Ctrl+Enter

- 기본 연산자

(1) 산술연산자 예시

```
> 1+2+3+4+5+6+7+8+9+10 #합
[1] 55
> (1+2+3+4+5+6+7+8+9+10)/10 #평균
[1] 5.5
> 1+2+3+4+5 #합
[1] 15
> (1+2+3+4+5)/5 #평균
[1] 3
> (1-3)^2+(2-3)^2+(3-3)^2+(4-3)^2+(5-3)^2 #편차제곱합
[1] 10
> 10/4 #분산
[1] 2.5
> 13 %/% 4 #몫
[1] 3
> 13%%4 #나머지
[1] 1
```

(2) 비교연산자 예시

```
> 3>4 #3이 4보다 큰지 판단
[1] FALSE
> 3>=4 #3이 4보다 크거나 같은지 판단
[1] FALSE
> 3<4 #3이 4보다 작은지 판단
[1] TRUE
> 3<=4 #3이 4보다 작거나 같은지 판단
[1] TRUE
> 3==4 #3이 4와 같은지 판단
[1] FALSE
> 3!=4 #3과 4가 다른지 판단
[1] TRUE
> !(3==4) #3과 4가 같다는 것을 부정
[1] TRUE
```

(3) 논리연산자 예시

```
> (5<=6)&(8<=7) #두개(또는 이상)의 조건을 동시에 만족할때 TRUE
[1] FALSE
> (5<=6)&(8>=7)
[1] TRUE
> (5<=6)|(8<=7) #두개(또는 이상)의 조건 중에서 하나만 만족해도 TRUE
[1] TRUE
> (5<=6)|(8>=7)
[1] TRUE
> (5==6)|(8>=7)
[1] TRUE
> (5==6)|(8==7)
[1] FALSE
```

Chapter 1. R 소개 및 설치

1.4 R 활용

R의 기본 자료형 : 자료 저장을 위해 R에서 사용하는 기본 자료형은 하나의 객체이며 그 종류는 다음과 같음

- 숫자형
 - 정수(integer)
 - 실수(numeric)
 - 복소수(complex)
- 문자형(character)
 - 큰따옴표로 둘러 쌓아 표현 : "abc", "123" 등
- 논리형(logical)
 - TRUE(T), FALSE(F)
- 특수한 상태를 나타내는 상수
 - NULL : 정의되지 않은 값
 - NA : Missing Value
 - -Inf, Inf : 음과 양의 무한대
 - NaN : 수의 연산에서 불능의 경우 표현 0/0, Inf/Inf 등

Chapter 1. R 소개 및 설치

1.4 R 활용

R의 기본 자료형 : 자료 저장을 위해 R에서 사용하는 기본 자료형은 하나의 객체이며 그 종류는 다음과 같음

- 숫자형
 - 정수(integer) → 1, -5, 100
 - 실수(numeric) → 1.5, -0.75, 100.0
 - 복소수(complex) → $2 + 3i$, $-1 - 2i$ → 주파수, 전기회로, 양자 역학 및 크기와 관련된 분야에서 사용
- 문자형(character)
 - 큰따옴표로 둘러 쌓아 표현 → "abc", "123"
- 논리형(logical)
 - TRUE(T), FALSE(F) → TRUE, FALSE
- 특수한 상태를 나타내는 상수
 - NULL : 정의되지 않은 값 → NULL
 - NA : Missing Value → NA
 - -Inf, Inf : 음과 양의 무한대 → -Inf, Inf
 - NaN : 수의 연산에서 불능의 경우 표현 $0/0$, Inf/Inf 등 → 0/0

Chapter 1. R 소개 및 설치

1.4 R 활용

R의 기본 자료형 : 자료 저장을 위해 R에서 사용하는 기본 자료형은 하나의 객체이며 그 종류는 다음과 같음

```
integer_var <- 5
numeric_var <- 3.14
complex_var <- 2 + 3i
character_var <- "Hello, world!"
logical_var <- TRUE
another_logical_var <- FALSE
NULL_var <- NULL
NA_var <- NA
negative_inf <- -Inf
positive_inf <- Inf
nan <- 0/0
```

Chapter 1. R 소개 및 설치

1.4 R 활용

R의 기본 자료형 : 자료 저장을 위해 R에서 사용하는 기본 자료형은 하나의 객체이며 그 종류는 다음과 같음

```
typeof(integer_var) [1] "double"  
typeof(numeric_var) [1] "double"  
typeof(complex_var) [1] "complex"  
typeof(character_var) [1] "character"  
typeof(logical_var) [1] "logical"  
typeof(another_logical_var) [1] "logical"  
typeof(NULL_var) [1] "NULL"  
typeof(NA_var) [1] "logical"  
typeof(negative_inf) [1] "double"  
typeof(positive_inf) [1] "double"  
typeof(nan) [1] "double"
```

자극사용

→ 연속적인 값

```
mode(integer_var) [1] "numeric"  
mode(numeric_var) [1] "numeric"  
mode(complex_var) [1] "complex"  
mode(character_var) [1] "character"  
mode(logical_var) [1] "logical"  
mode(another_logical_var) [1] "logical"  
mode(NULL_var) [1] "NULL"  
mode(NA_var) [1] "logical"  
mode(negative_inf) [1] "numeric"  
mode(positive_inf) [1] "numeric"  
mode(nan) [1] "numeric"
```

Chapter 1. R 소개 및 설치

1.4 R 활용

R의 기본 자료형 : 자료 저장을 위해 R에서 사용하는 기본 자료형은 하나의 객체이며 그 종류는 다음과 같음

- 기본 데이터 형태

① 숫자형 (numeric) : 숫자로만 이루어진 것으로 벡터 안에 문자가 들어있으면 전체가 문자형으로 바뀜

② 문자형 (character) : 문자를 처리할 때는 (“ ”), (‘ ’) 로 표시함

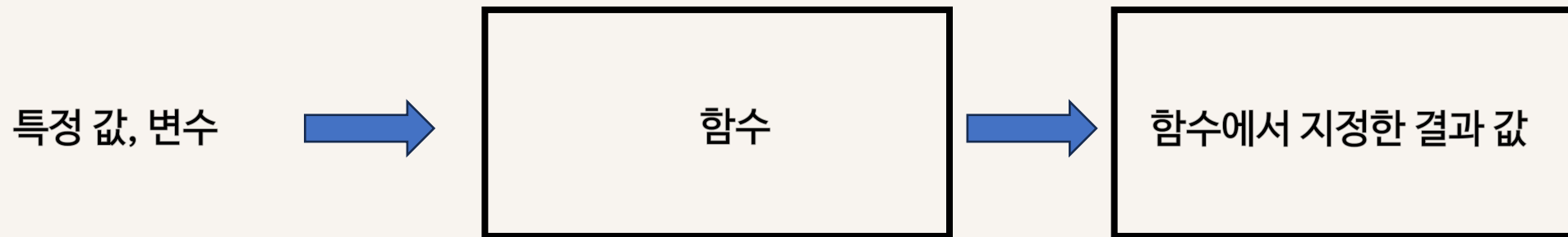
③ 논리값 (logic value : TRUE, FALSE) : 내부적으로 TRUE는 1, FALSE는 0 의미함

④ NA형 & NULL형 : NA형은 연산 대상에 포함, NULL형은 연산 대상에서 제외됨

Chapter 1. R 소개 및 설치

1.4 R 활용

함수 : 변수는 자료를 저장하는 역할을 한다면 함수는 행위를 나타냄



Chapter 1. R 소개 및 설치

1.4 R 활용

함수 : 변수는 자료를 저장하는 역할을 한다면 함수는 행위를 나타냄

```
함수명 <- function(인자1, 인자2, ...){
    실행될 프로그램
    return(반환 값) #생략가능
}
```

반복작업이 있을 때 프로그래밍을 간결하고 깔끔하게 만들어줌

함수	내용
sin()	$\sin x$
cos()	$\cos x$
tan()	$\tan x$
sinh()	$\sinh x$
exp()	e^x
log()	\log
log(x, base=a)	\log_a^x
loglp()	$\log_2^{(1+x)}$
sqrt()	\sqrt{x}
abs()	$ x $
round(x, digits)	반올림
floor()	내림
ceiling()	올림
min()	최소
max()	최대
sum()	합
choose()	C
factorial()	n!
which.max()	벡터내의 최대값 인덱스
which.min()	벡터내의 최소값 인덱스

Chapter 1. R 소개 및 설치

1.4 R 활용

함수 : 변수는 자료를 저장하는 역할을 한다면 함수는 행위를 나타냄 → 변수값 생성

```
a<-3  
print(a)  
print(3)  
print("Hello")
```

```
paste(3,4,5,6,7)  
print(paste(3,4,5,6,7))
```

```
paste0(3+4,4,5,6,7)  
print(paste0(3+4,4,5,6,7))
```

```
paste(3,4,5,6,7, sep = "-")  
paste0(3+4,4,5,6,7, sep = "-")
```

Chapter 1. R 소개 및 설치

1.4 R 활용

함수 : 변수는 자료를 저장하는 역할을 한다면 함수는 행위를 나타냄 → 변수값 생성

```
a=c(1,2,3,4,5,6)
```

```
sum(c(1,2,3,4,5,6,7,8,9,10))
```

```
seq(from=1, to=5, by=2) #1부터 5까지 2개씩 증가  
seq(1,5, by=2)
```

```
seq(0,1, length.out=1000) #0부터 1까지 1000개가 되는 자료 생성
```

```
rep(c(1,2,3), times=2) 1,2,3을 두 번 반복하는 vector
```

```
rep(c(1,2,3), each=2) 1,2,3의 개별 원소를 두 번 반복하는 vector
```

Chapter 1. R 소개 및 설치

1.4 R 활용

함수 : 변수는 자료를 저장하는 역할을 한다면 함수는 행위를 나타냄 → 변수값 확인

```
x<- 1:100
```

```
head(x)
```

```
head(x, n=7)
```

```
tail(x)
```

```
tail(x, n=7)
```

Chapter 1. R 소개 및 설치

1.4 R 활용

함수 : 변수는 자료를 저장하는 역할을 한다면 함수는 행위를 나타냄 → 임의 값 생성

```
sample(x, size, replace =FALSE, prob = NULL)
```

x : 선택할 집단에 해당하는 양의 값

size : 추출할 개수

replace : 복원여부(기본값 FALSE)

prob : x의 각 원소별 추출 확률(기본값은 NULL)

Chapter 1. R 소개 및 설치

1.4 R 활용

함수 : 변수는 자료를 저장하는 역할을 한다면 함수는 행위를 나타냄 → 임의 값 생성

```
sample(x, size, replace =FALSE, prob = NULL)
```

```
sample(10) #1부터 10까지를 10개를 임의로 섞어 추출
```

```
sample(45, 6) # 1부터 45에서 6개를 임의로 추출함
```

```
sample(10, 3, replace=TRUE) #동일한 확률 (1/10)로 복원 추출
```

```
sample(10, 3, prob=(1:10)/55) #10개중 3개를 임의로 복원 추출하는데 추출될 확률은 (1:10)/55로 각각 다름
```

```
sample(10, 3, prob = c(0.1, 0.2, 0.3, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05))
```

Chapter 1. R 소개 및 설치

1.4 R 활용

함수 : 변수는 자료를 저장하는 역할을 한다면 함수는 행위를 나타냄 → 값에 대한 조건

```
x<-c(2,4,-1,3)
```

```
which(x>2) #x의 값들 중 2보다 큰 값 출력
```

```
names(x)<-c("1번", "2번", "3번", "4번") x의 값에 이름을 각각 부여
```

```
which(x>2)
```

Chapter 1. R 소개 및 설치

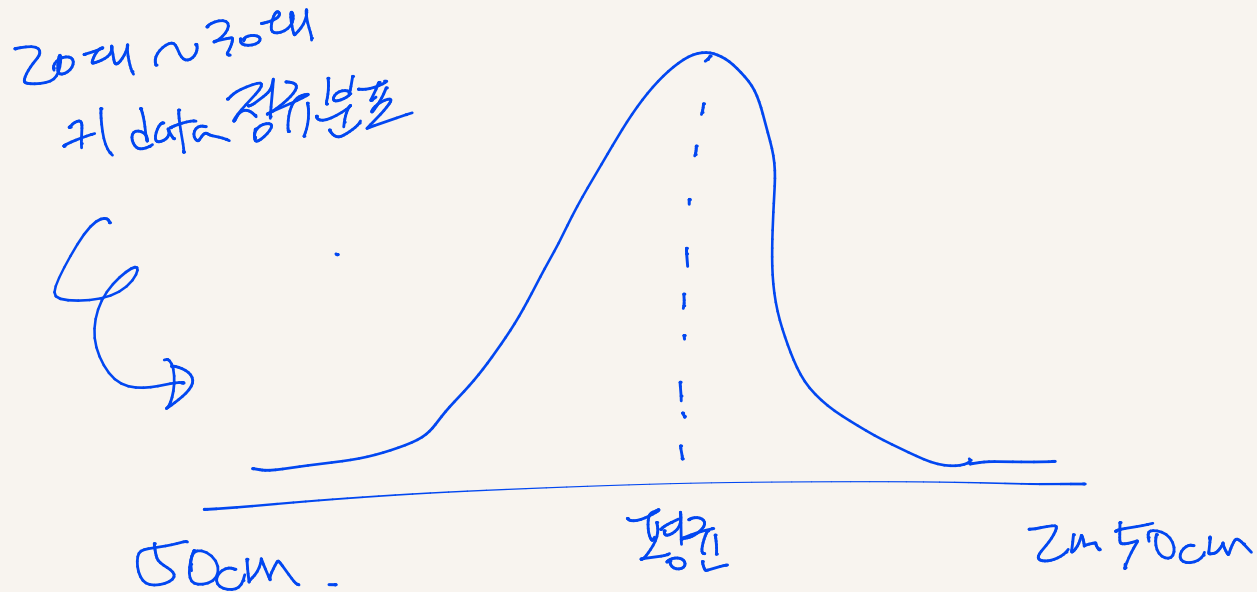
1.4 R 활용

함수 : 변수는 자료를 저장하는 역할을 한다면 함수는 행위를 나타냄 → 난수 발생

`runif(n, min, max)` #최소와 최대 사이의 균일 분포에서 n개의 연속형 난수를 생성

`rnorm(n, 평균, sd)` #주어진 평균과 표준편차 sd를 갖는 정규분포에서 n개의 난수를 생성

`rbinom(n, size, prob)` #주어진 시행 횟수와 성공 가능성 prob를 사용하여 이항 분포에서 n개의 난수를 생성함



Chapter 1. R 소개 및 설치

1.4 R 활용

특정 변수 제거

```
a=10
```

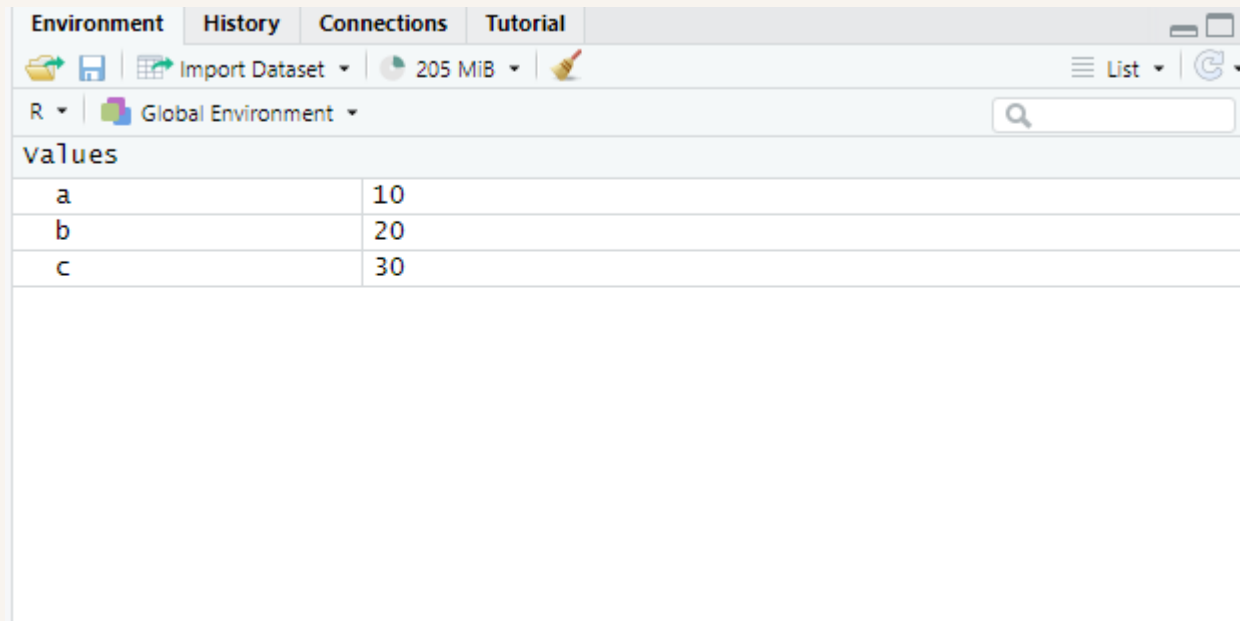
```
b=20
```

```
c=30
```

```
rm(a) #a라는 변수를 삭제
```

```
rm(list=c("a", "c")) #a,b 변수 삭제
```

```
rm(list=ls()) #모든 값 삭제
```



The screenshot shows the R Studio Environment pane. At the top, there are tabs for Environment, History, Connections, and Tutorial. Below the tabs, there are icons for Import Dataset, a memory usage indicator (205 MiB), and a search icon. The main area is titled 'Global Environment' and contains a table of variables. The table has two columns: 'values' and 'values'. The first column lists the variables 'a', 'b', and 'c', and the second column lists their corresponding values: 10, 20, and 30.

values	values
a	10
b	20
c	30

Chapter 1. R 소개 및 설치

1.4 R 활용

특정 변수 제거

- 기본적인 작업 디렉터리를 변경하여, 저장 및 불러오기를 할 수 있음
- `getwd()` 함수로 작업 디렉터를 확인
- `setwd("원하는 디렉터리 경로")` 함수로 작업 디렉터를 변경

```
#현재 나의 작업 경로 확인  
getwd()
```

```
#나의 작업 경로 설정 1  
setwd("c:/경로1/경로2/경로3")
```

```
#나의 작업 경로 설정 2  
setwd("c:\\경로1\\경로2\\경로3")
```

Chapter 1. R 소개 및 설치

문제

1부터 100까지 5씩 증가하는 값을 생성하시오.

seq

1,2,3,4,5의 벡터 값을 만들어 이를 100번 반복해 도출하는 벡터를 만드시오.
(예 : 1,2,3,4,5,1,2,3,4,5,1,2,3,4,5 ...)

times = 100

1,2,3,4,5의 벡터 값을 만들어 개별 값이 각각 100번 반복해 도출하는 벡터를 만드시오.
(예 : 1,1,1,...,2,2,2,...,3,3,3,...,4,4,4,...,5,5,5,...)

each = 100

Chapter 1. R 소개 및 설치

문제

1부터 10까지 5개의 숫자를 무작위로 섞어서 프린트 하시오.

Sample.

숫자 범위가 1부터 49까지인 복권에서 6개의 숫자를 뽑으시오.

sample.

1부터 10까지의 숫자 3개를 무작위로 선택하시오. (복원추출)

sample

replace True.

1부터 10까지의 숫자 3개를 무작위로 뽑으시오. (각 숫자 'i'의 가중치는 'i/55')

prob = (1:10)/55.

확률 벡터를 사용하여 1부터 10까지의 숫자 3개를 무작위로 뽑으시오. (0.1, 0.2, 0.3, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)

prob = c(.)

Chapter 1. R 소개 및 설치

문제

50에서 100 사이에 균일하게 분포된 100명의 학생 점수를 생성하시오.

`runif(100, 50, 100)`

IQ의 평균이 100이고 표준편차가 15인 정규분포에서 IQ 점수 200개를 생성하시오.

`rnorm(200, 100, 15)`

인구 10명 중 1명에게 영향을 미치는 질병을 rbinom을 사용하여 500명을 대상으로 실시했을 때, 총 걸린 사람의 수는?

`rbinom`