

# 데이터분석기초(R)

## 2주차

IDE와 변수

# R과 R 스튜디오의 설치 및 사용

## 4. R과 R 스튜디오의 설치 및 사용

---

### 1. R과 R 스튜디오의 소개

- Python : 프로그래밍 언어로서의 특성이 강함
- R : 데이터 분석을 목적으로 개발, R 로 SW 를 만들지는 못함

R studio 라는 훌륭한 작업환경 제공

풍부한 패키지 제공

미려한 데이터 시각화 패키지 제공



그림 1-10 R과 파이썬

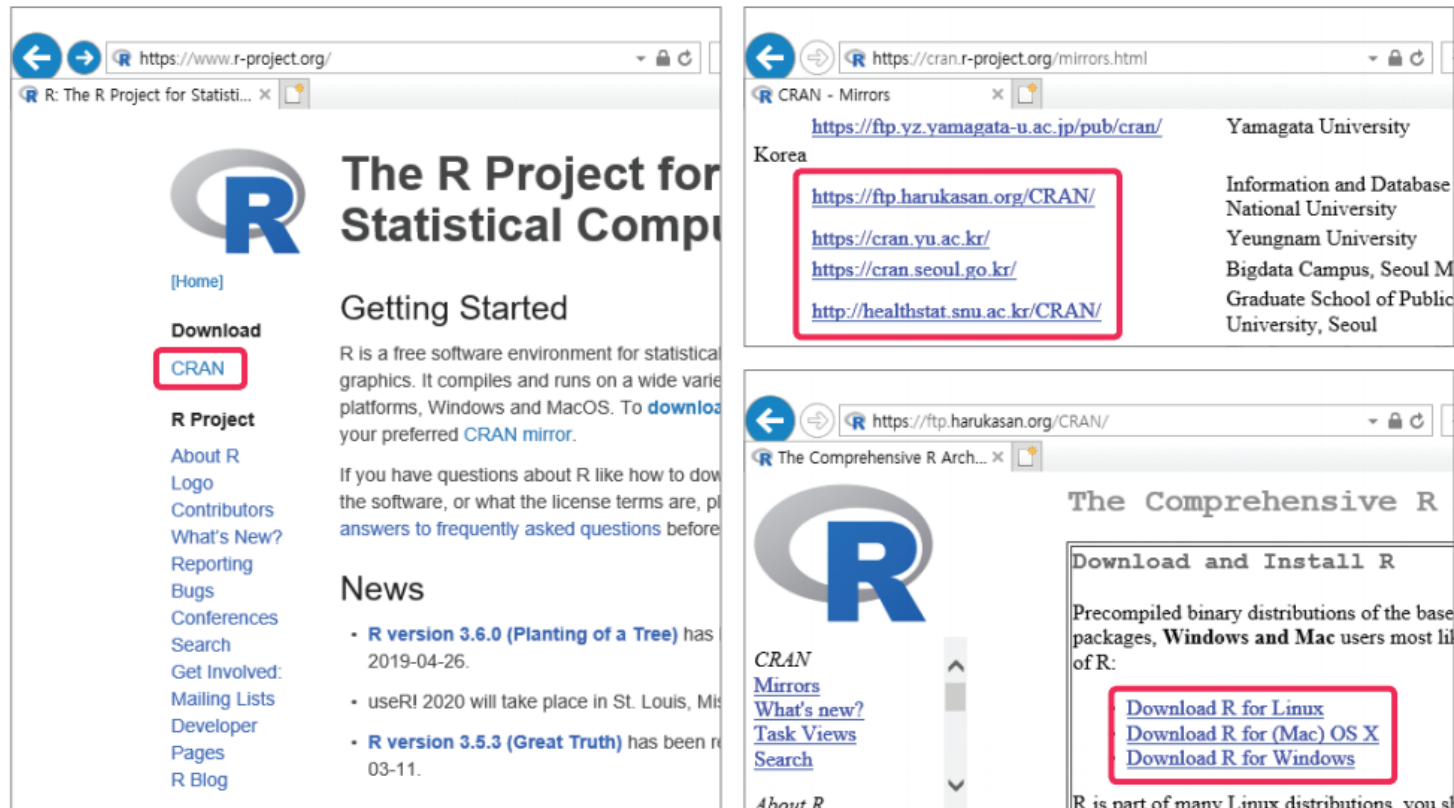


그림 1-11 R을 쉽게 사용할 수 있는 R 스튜디오

## 4. R과 R 스튜디오의 설치 및 사용

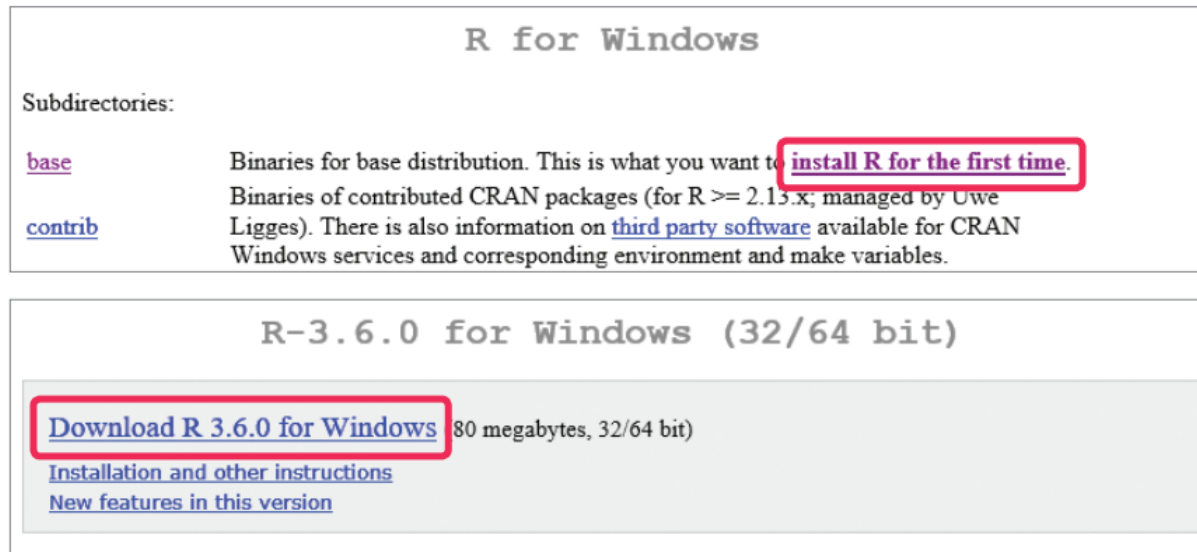
### 2. R의 설치

01 <https://www.r-project.org/> 에 접속하여 설치 진행

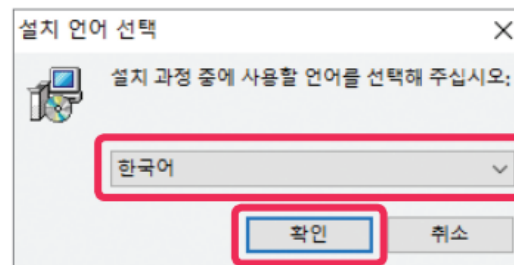


## 4. R과 R 스튜디오의 설치 및 사용

02 [install R for the first time] 링크 클릭 → [Download 3.6.0 for Windows] 클릭

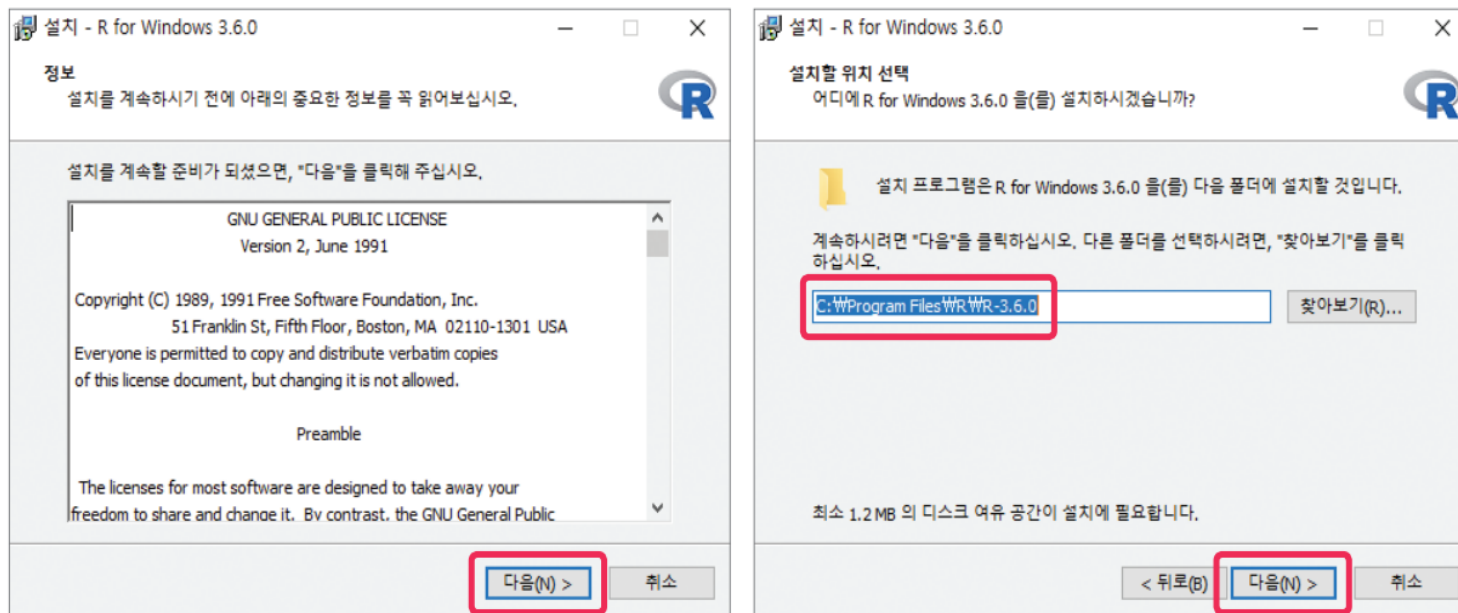


03 [한국어] 선택하고 [확인] 클릭



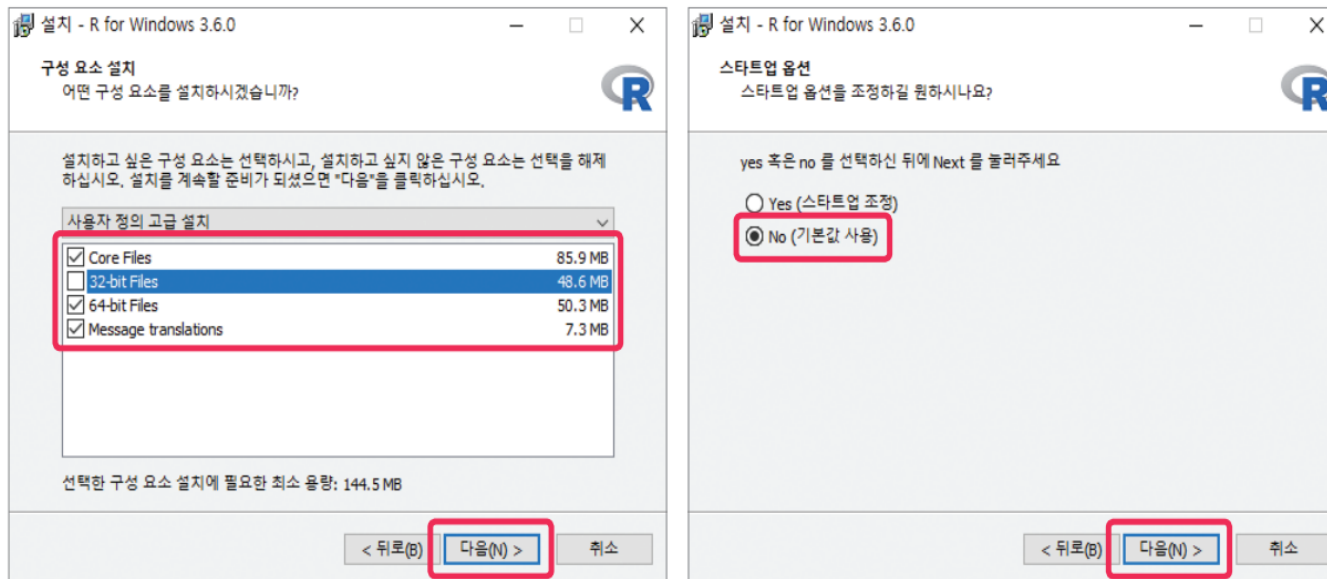
## 4. R과 R 스튜디오의 설치 및 사용

04 설치 정보가 나타나면 내용 확인하고 [다음] 버튼 클릭 → 설치할 위치 선택  
에서 경로를 변경하거나 유지한 채로 [다음] 버튼 클릭



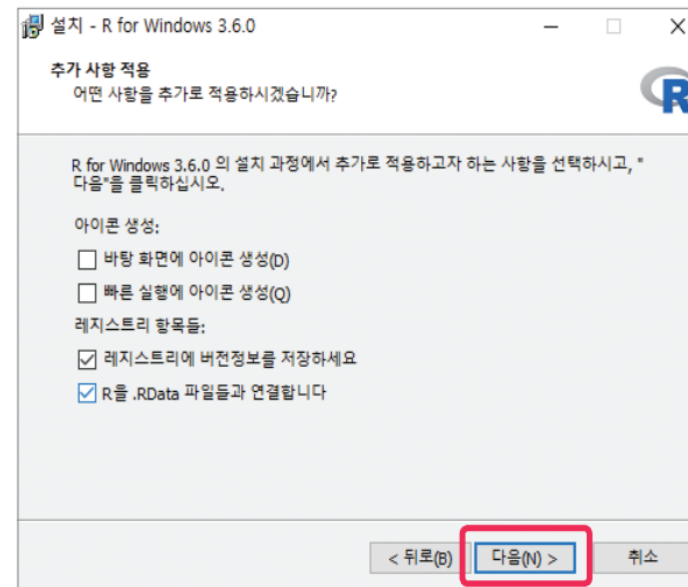
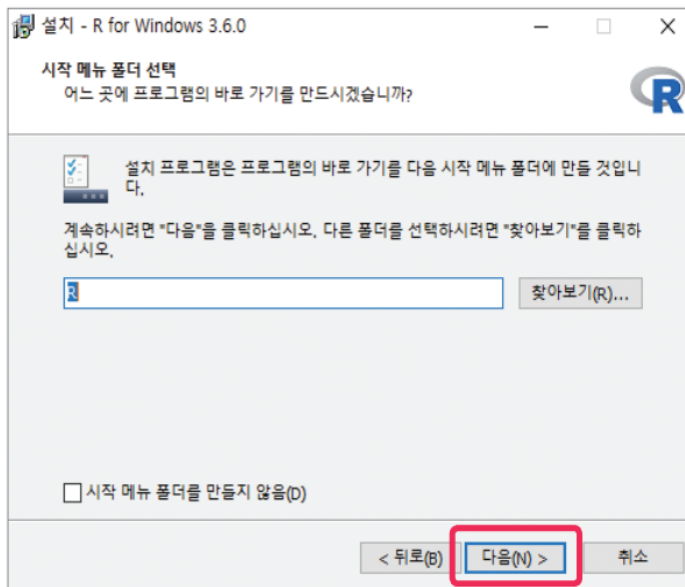
## 4. R과 R 스튜디오의 설치 및 사용

05 구성 요소 설치에서 필요한 항목 체크하고 [다음] 버튼 클릭 → 스타트업 옵션에서 [No]를 선택 후, [다음] 버튼 클릭



## 4. R과 R 스튜디오의 설치 및 사용

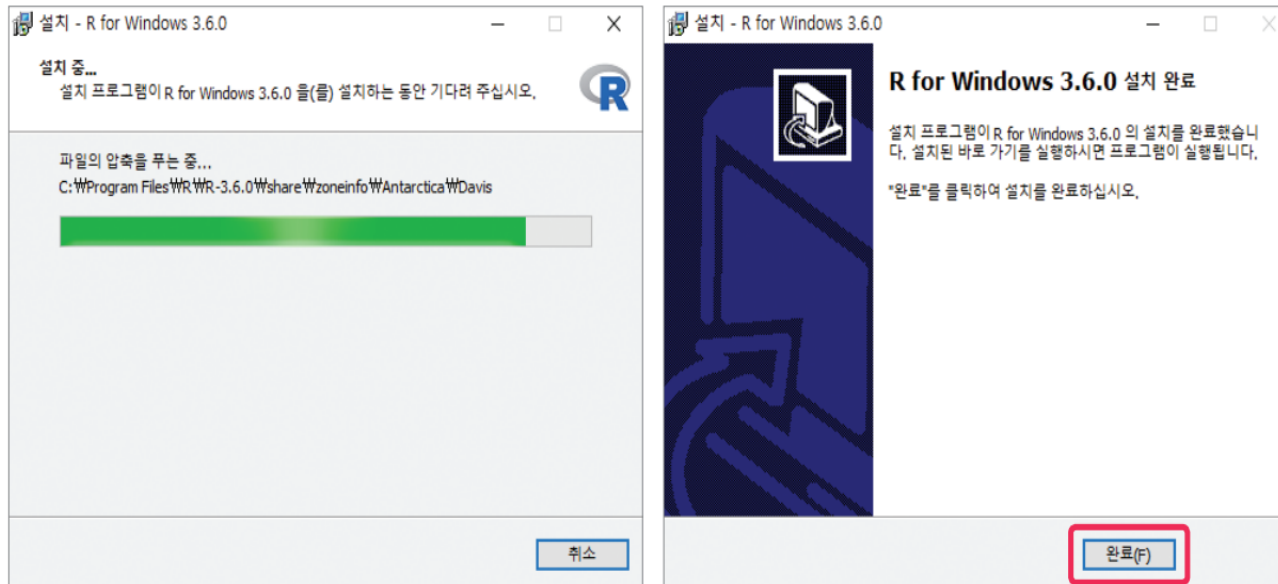
06 시작 메뉴 폴더 선택은 내용 변경 없이 [다음] 버튼 클릭 → 추가 사항 적용도 내용 변경 없이 [다음] 버튼 클릭





## 4. R과 R 스튜디오의 설치 및 사용

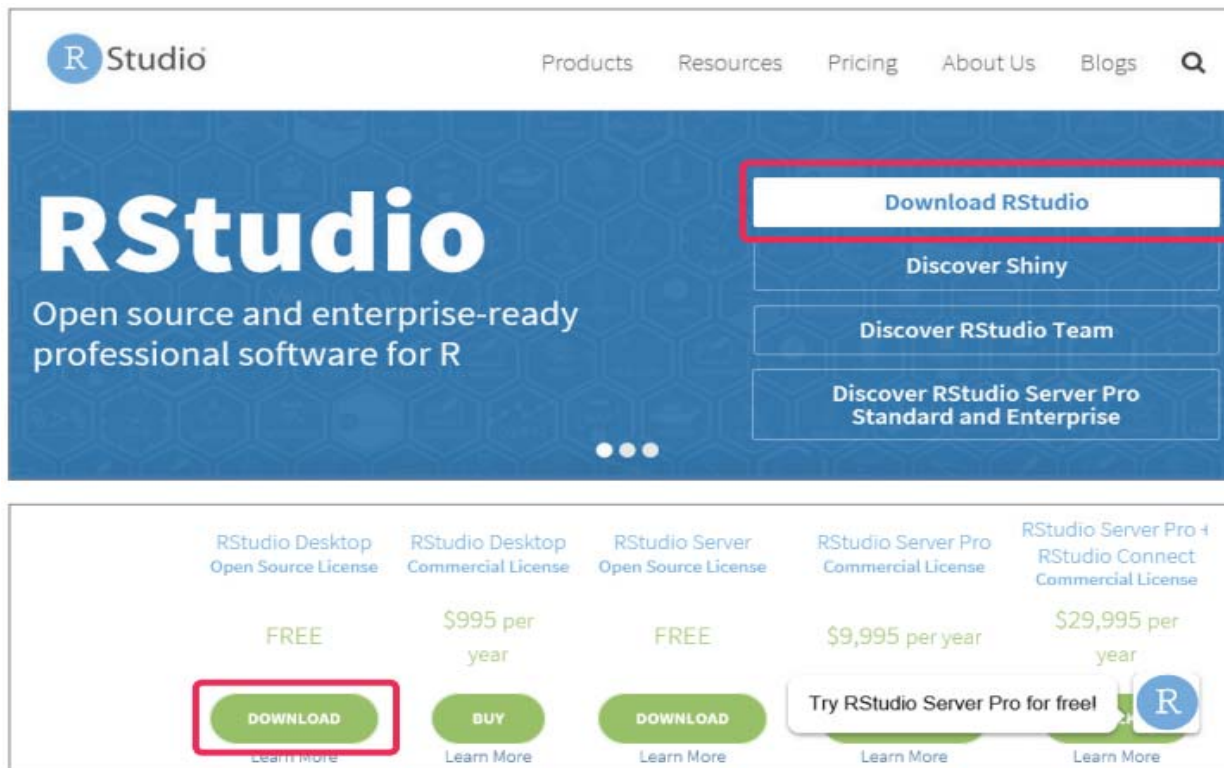
### 07 설치 완료 창 열리면 [완료] 버튼을 눌러 설치 완료



## 4. R과 R 스튜디오의 설치 및 사용

### 3. R 스튜디오의 설치

01 <https://www.rstudio.com/>에 접속 → [Download RStudio]를 클릭 → [RStudio Desktop Open Source License]의 [DOWNLOAD] 버튼 클릭



## 4. R과 R 스튜디오의 설치 및 사용

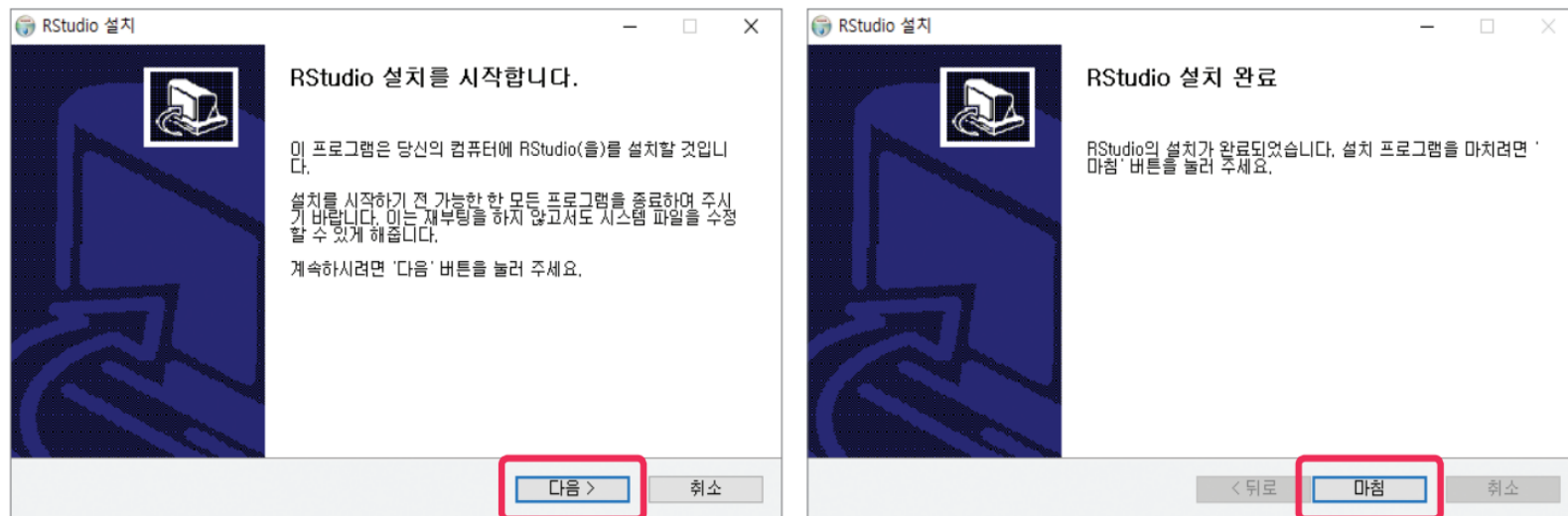
02 운영체제별 설치 파일 다운로드 목록이 나타나면 사용자 환경에 맞는 링크를 클릭하여 설치 파일 다운로드

### Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.2.1335 - Windows 7+ (64-bit)	126.9 MB	2019-04-08	d0e2470f1f8ef4cd35a669aa323a2136
RStudio 1.2.1335 - Mac OS X 10.12+ (64-bit)	121.1 MB	2019-04-08	6c570b0e2144583f7c48c284ce299eef
RStudio 1.2.1335 - Ubuntu 14/Debian 8 (64-bit)	92.2 MB	2019-04-08	c1b07d0511469abfe582919b183eee83
RStudio 1.2.1335 - Ubuntu 16 (64-bit)	99.3 MB	2019-04-08	c142d69c210257fb10d18c045fff13c7
RStudio 1.2.1335 - Ubuntu 18 (64-bit)	100.4 MB	2019-04-08	71a8d1990c0d97939804b46cfb0aea75
RStudio 1.2.1335 - Fedora 19+/RedHat 7+ (64-bit)	114.1 MB	2019-04-08	296b6ef88969a91297fab6545f256a7a
RStudio 1.2.1335 - Debian 9+ (64-bit)	100.6 MB	2019-04-08	1e32d4d6f6e216f086a81ca82ef65a91
RStudio 1.2.1335 - OpenSUSE 15+ (64-bit)	101.6 MB	2019-04-08	2795a63c7efd8e2aa2dae86ba09a81e5
RStudio 1.2.1335 - SLES/OpenSUSE 12+ (64-bit)	94.4 MB	2019-04-08	c65424b06ef6737279d982db9eefcae1

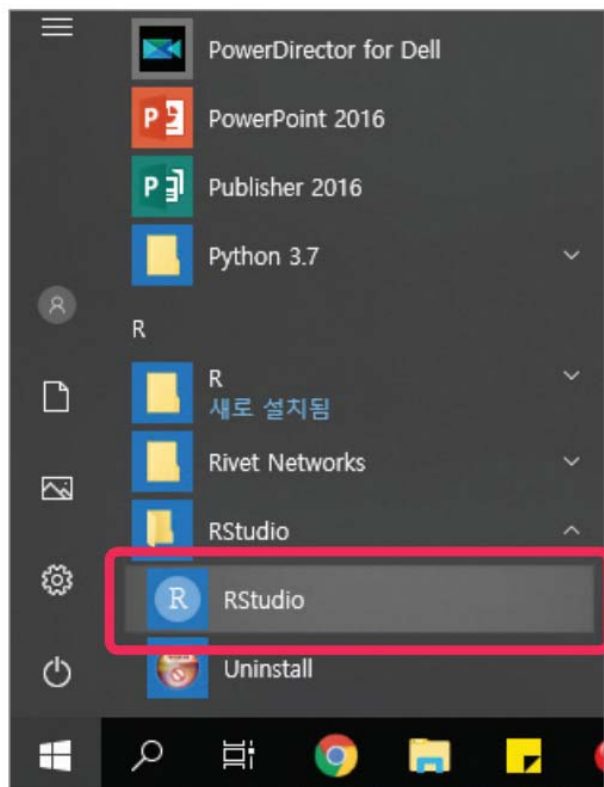
## 4. R과 R 스튜디오의 설치 및 사용

03 설치 파일 더블클릭 → 계속해서 [다음] 버튼을 클릭 → R 스튜디오 설치가 완료되면 [마침] 버튼 클릭



## 4. R과 R 스튜디오의 설치 및 사용

04 설치가 완료되면 윈도우 시작 메뉴에서 [RStudio]-[Rstudio] 클릭하여  
R 스튜디오 실행



## 4. R과 R 스튜디오의 설치 및 사용

### 4. R 스튜디오의 화면 구성

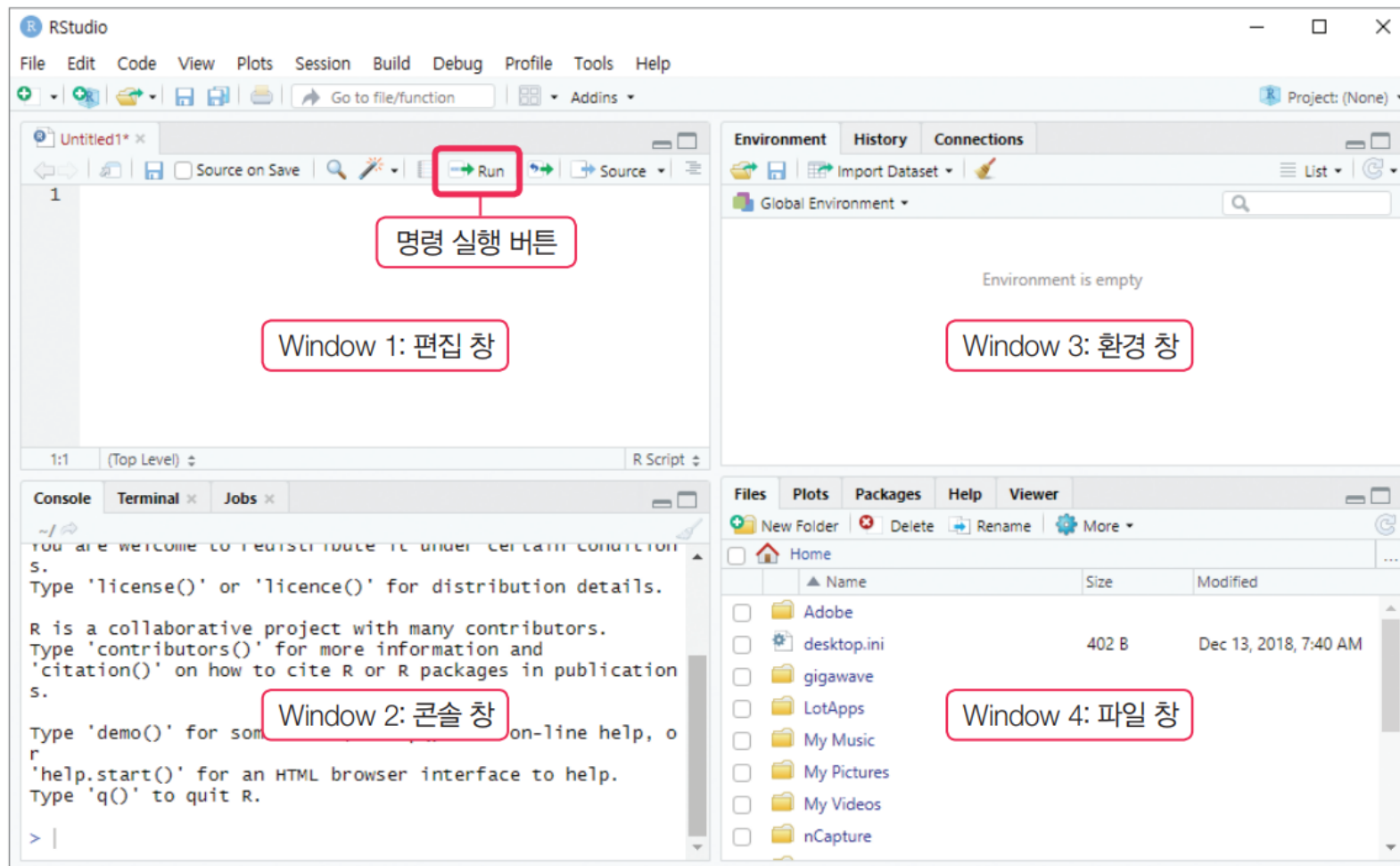


그림 1-12 R 스튜디오 초기 화면

## 4. R과 R 스튜디오의 설치 및 사용

---

### 4.1 편집(Script) 창

- R 명령문('R 스크립트' 라고도 한다.)들을 작성하고 실행하는 영역

### 4.2 콘솔(Console) 창

- 편집 창에서 R 명령문을 편집하고 실행 버튼을 클릭했을 때, 명령문의 실행 과정 및 결과를 표시하는 창

### 4.3 환경(Environment) 창

- R 명령문이 실행하는 동안 만들어지는 각종 변수나 자료구조의 내용을 보여주는 영역

### 4.4 파일(Files) 창

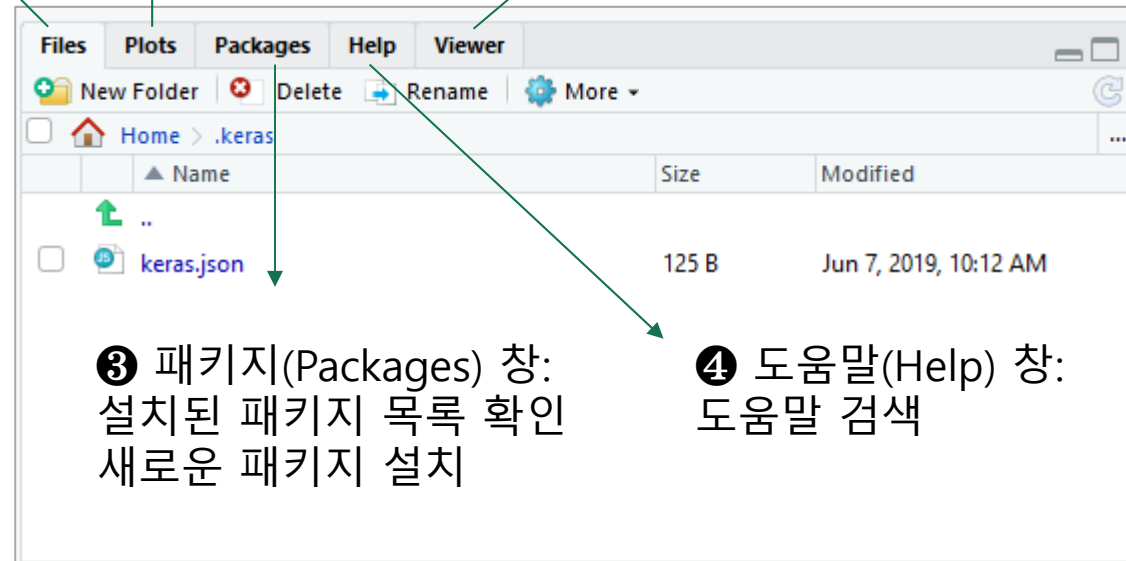
- 도움말, 패키지 설치 및 조회, 그래프 실행 내용 조회 등 유용한 기능을 제공하는 창

## 4. R과 R 스튜디오의 설치 및 사용

❶ 파일(Files) 창:  
현재 작업폴더의 내용을  
탐색기처럼 보여준다

❷ 플롯(Plots) 창:  
그래프가 표시되는 영역

❸ 뷰어(Viewer) 창:  
결과가 웹브라우저에  
나타나는 경우 여기에 표시



❹ 패키지(Packages) 창:  
설치된 패키지 목록 확인  
새로운 패키지 설치

❺ 도움말(Help) 창:  
도움말 검색



## 4. R과 R 스튜디오의 설치 및 사용

### 5. R 스튜디오 다루기

#### 5.1 R 스튜디오 화면 재구성하기

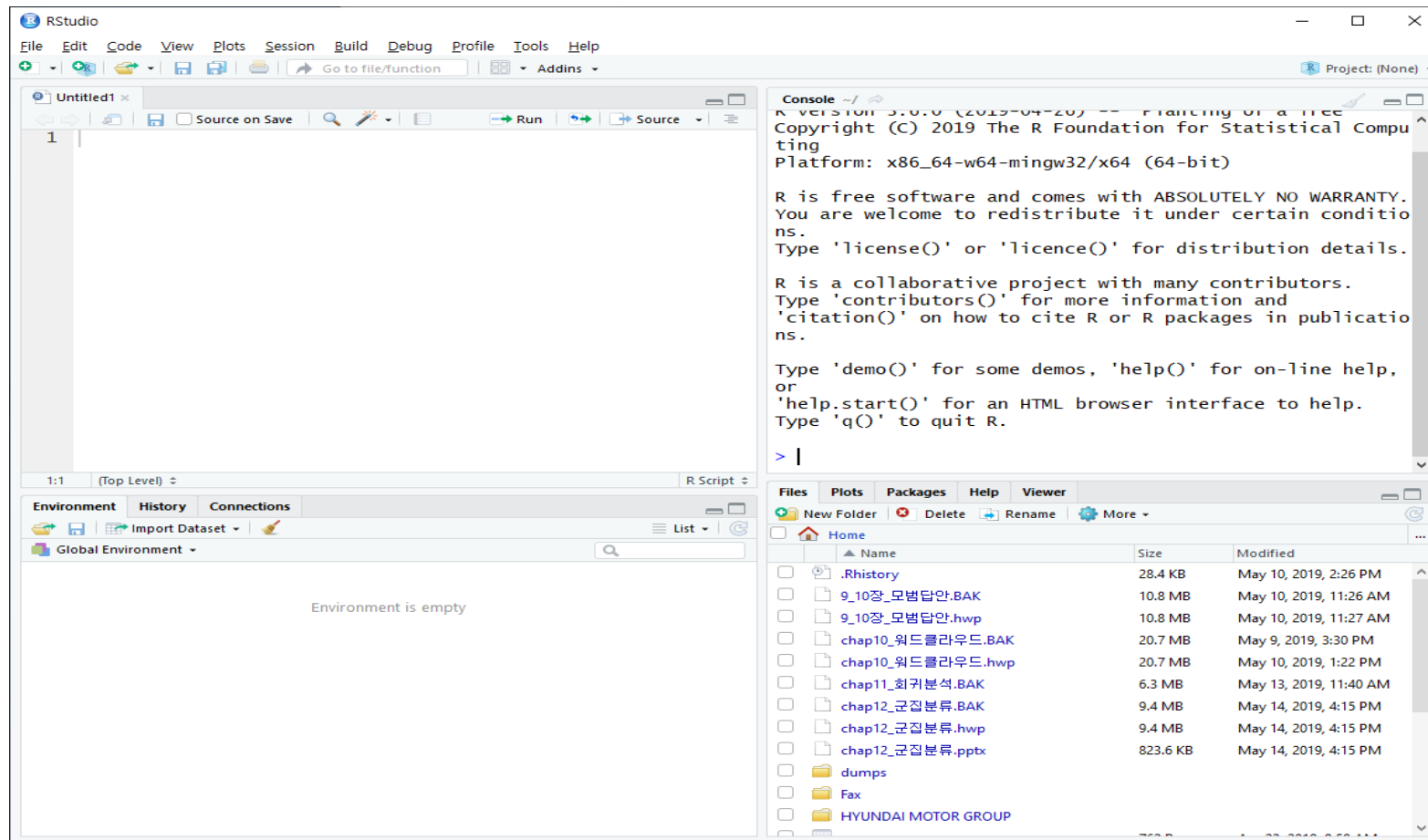


그림 1-13 콘솔 창 재배치 후의 R 스튜디오

## 4. R과 R 스튜디오의 설치 및 사용

### 5.2 R 스튜디오에서 명령문의 실행

```
5+8  
3+(4*5)  
a <- 10  
print(a)
```

```
> 5+8  
[1] 13
```

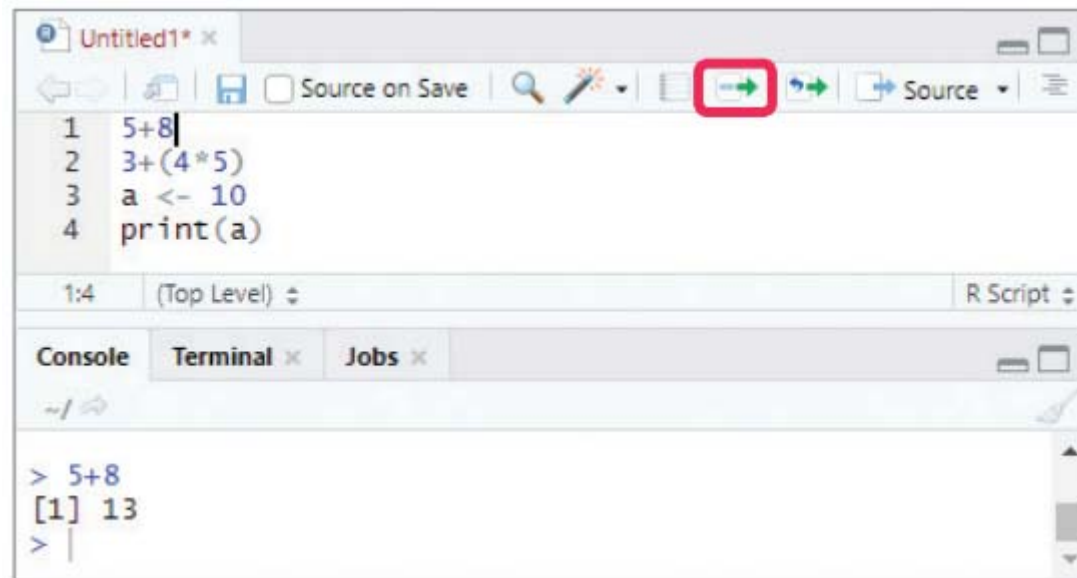


그림 1-14

편집 창 1행에 커서를 놓고  
실행 아이콘을 클릭했을 때  
콘솔 창에서의 실행 결과

## 4. R과 R 스튜디오의 설치 및 사용

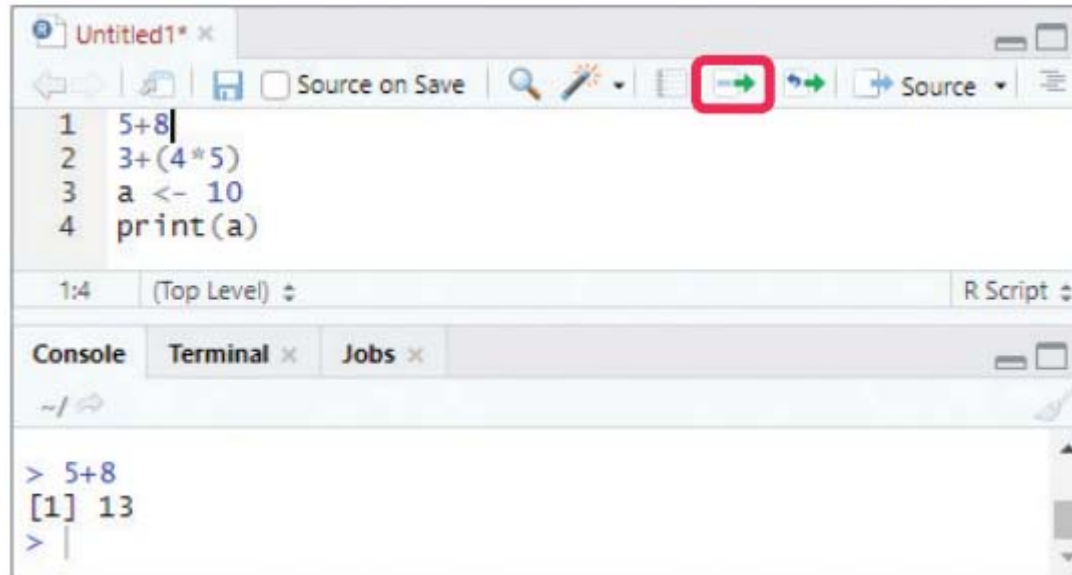


그림 1-15

편집 창 1~4행을 블록 선택하고

실행 아이콘을 눌렀을 때 콘솔 창에서의  
실행 결과

명령어 실행	단축키
한 줄만 실행할 때	명령어가 있는 줄에서 <b>Ctrl</b> + <b>Enter</b>
여러 줄 실행할 때	명령어들을 드래그하여 블록을 만든 후 <b>Ctrl</b> + <b>Enter</b>
편집된 모든 명령문들을 실행할 때	<b>Ctrl</b> + <b>Alt</b> + <b>R</b>
바로 직전에 실행한 명령을 다시 실행할 때	<b>Ctrl</b> + <b>Shift</b> + <b>P</b>

표 1-1 명령어 실행과 관련된 단축키

## 4. R과 R 스튜디오의 설치 및 사용

---

### 5.3 R 스튜디오에서의 저장과 종료

- 메뉴에서 [File]-[Save] 또는 [File]-[Save As]
- R 스크립트 파일의 확장자 이름은 일반적으로 'test.R'과 같이 '.R'을 붙임
- 아래와 같은 메시지가 출력되면 [Save] 클릭

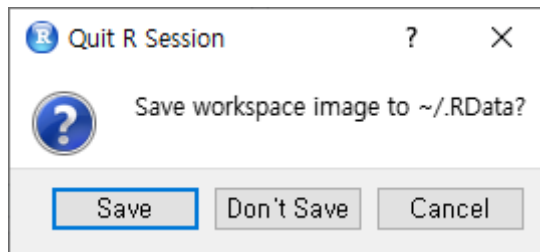


그림 1-16 R 스튜디오 종료 대화상자

## 4. R과 R 스튜디오의 설치 및 사용

### 5.4 패키지의 설치

- R에서는 데이터 분석을 위해서 매우 다양한 함수들을 제공
- 패키지(package) 는 이러한 함수들을 기능별로 묶어놓은 '꾸러미'
- 어떤 함수를 이용하기 위해서는 그 함수를 포함하고 있는 패키지를 사전에 설치해야 함

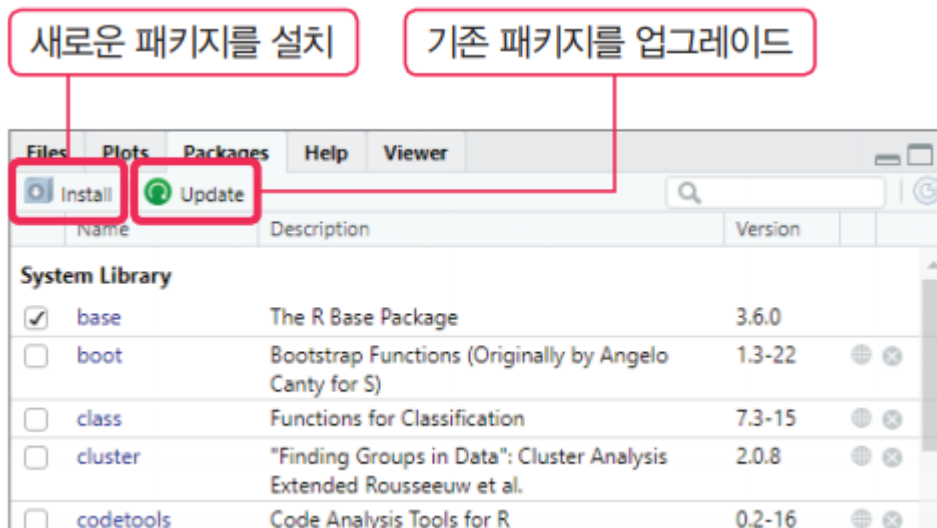


그림 1-17 현재 설치된 패키지의 목록

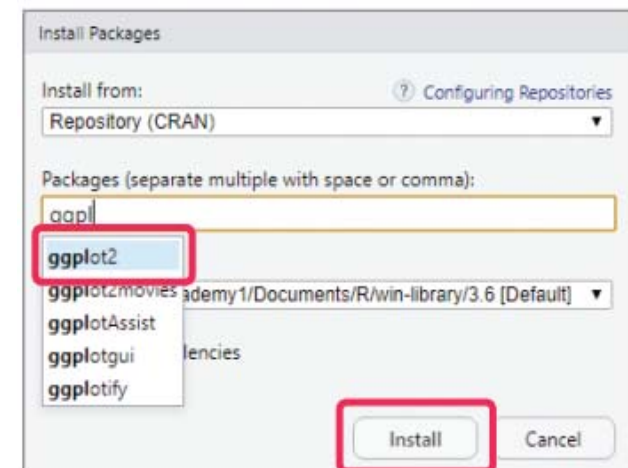
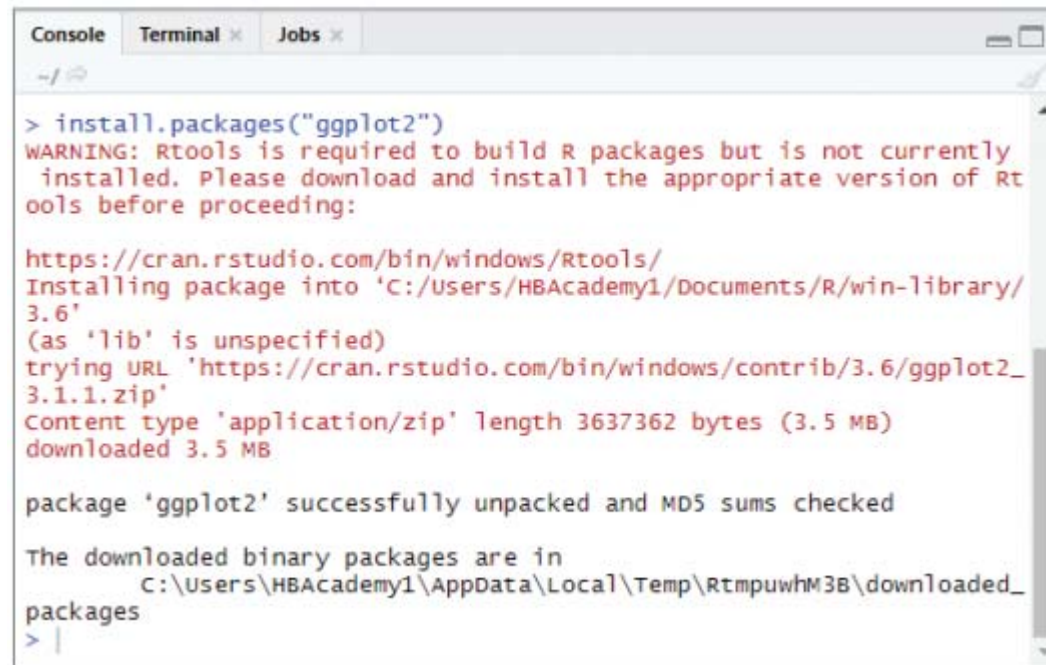


그림 1-18 패키지 설치 윈도우 화면

## 4. R과 R 스튜디오의 설치 및 사용



```
Console Terminal x Jobs x
~/
> install.packages("ggplot2")
WARNING: Rtools is required to build R packages but is not currently
installed. Please download and install the appropriate version of R t
ools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/HBAcademy1/Documents/R/win-library/
3.6'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/ggplot2_
3.1.1.zip'
Content type 'application/zip' length 3637362 bytes (3.5 MB)
downloaded 3.5 MB

package 'ggplot2' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:\Users\HBAcademy1\AppData\Local\Temp\RtmpuwhM3B\downloaded_
packages
> |
```

그림 1-19

패키지 설치가 성공한 경우의 일반적 화면

- 설치한 패키지 불러오기

```
library(ggplot2)
```

# 변수와 벡터

01. R의 기본 연산

02. 변수

03. 벡터의 이해

04. 벡터의 연산

05. 리스트와 팩터

# Section 01

## R의 기본 연산



# 1. R의 기본연산

## 1. 산술연산과 주석

코드 2-1

```
2+3  
(3+6)*8  
2^3
```

# 2의 세제곱

```
> 2+3  
[1] 5  
> (3+6)*8  
[1] 72  
> 2^3  
[1] 8
```

# 2의 세제곱

- 일반적으로 R에서는 한 줄에 하나의 명령문을 입력한다.
- 한 줄 내에서 # 이후의 내용은 주석으로 간주하여 실행하지 않는다.

## 1. R의 기본연산

연산자	의미	사용 예
+	덧셈	3+5+8
-	뺄셈	9-3
*	곱셈	7*5
/	나눗셈	8/3
%%	나눗셈의 나머지	8%%3
^	제곱	2^3

표 2-1 산술연산자

코드 2-2

```
7+4  
# 2^3
```

```
> # 2^3
```

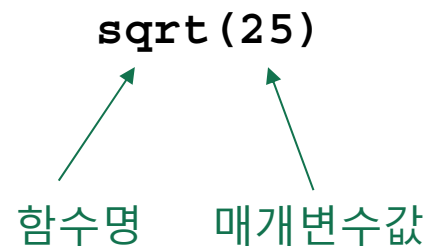
# 1. R의 기본연산

## 2. 산술연산 함수

### 코드 2-3

```
log(10)+5      # 로그함수  
sqrt(25)       # 제곱근  
max(5,3,2)     # 가장 큰 값
```

```
> log(10)+5      # 로그함수  
[1] 7.302585  
> sqrt(25)       # 제곱근  
[1] 5  
> max(5,3,2)     # 가장 큰 값  
[1] 5
```



## 1. R의 기본연산

함수	의미	사용 예
<code>log()</code>	로그함수	<code>log(10)</code> , <code>log(10, base=2)</code>
<code>sqrt()</code>	제곱근	<code>sqrt(36)</code>
<code>max()</code>	가장 큰 값	<code>max(3,9,5)</code>
<code>min()</code>	가장 작은 값	<code>min(3,9,5)</code>
<code>abs()</code>	절대값	<code>abs(-10)</code>
<code>factorial()</code>	팩토리얼	<code>factorial(5)</code>
<code>sin()</code> , <code>cos()</code> , <code>tan()</code>	삼각함수	<code>sin(pi/2)</code>

표 2-2 함수

# Section 02

변수

## 2. 변수

---

### 코드 2-4

```
a <- 10  
b <- 20  
c <- a+b  
print(c)
```

```
> a <- 10  
> b <- 20  
> c <- a+b  
> print(c)  
[1] 30
```

## 2. 변수

### 1. 변수의 개념

- 프로그램에서 어떤 값을 저장하는 저장소나 보관 박스

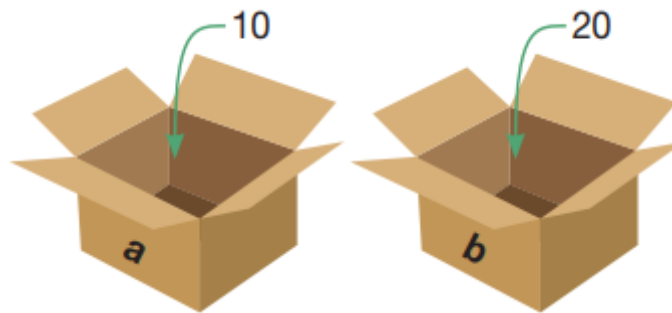


그림 2-1 변수의 개념: 변수, 변수명, 값

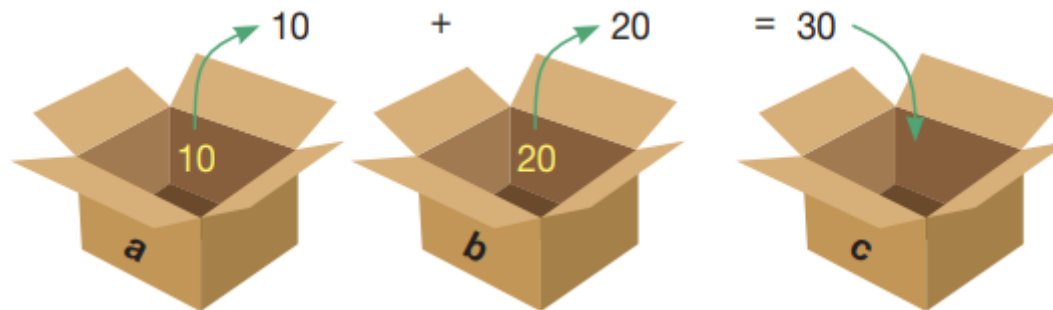


그림 2-2  $c = a + b$ 의 실행 과정

## 2. 변수

---

- `a <- 10`  
10을 변수 a에 저장
- `b <- 20`  
20을 변수 b에 저장
- `c <- a+b`  
변수 a의 값과 변수 b의 값을 더하여 변수 c에 저장
- `print(c)`  
변수 c의 값을 출력('c'만 입력해도 출력)



## 2. 변수

### 2. 변수명 지정

- ❶ 첫 글자는 영문자(알파벳)나 마침표(.)로 시작하는데, 일반적으로 영문자로 시작

ex) avg, .avg

ex) 12th는 숫자로 시작했기 때문에 변수명 사용 불가

- ❷ 두 번째 글자부터는 영문자, 숫자, 마침표(.), 밑줄(\_) 사용 가능

ex) v.1, a\_sum, d10

ex) this-data, this@data은 변수명 사용 불가(@과 - 같은 특수문자 사용 불가)

- ❸ 대문자와 소문자를 구분

ex) var\_A 와 var\_a는 서로 다른 변수

- ❹ 변수명 중간에 빈칸을 넣을 수 없음

ex) first ds는 변수명 사용 불가

### 3. 변수에 값 저장 및 확인

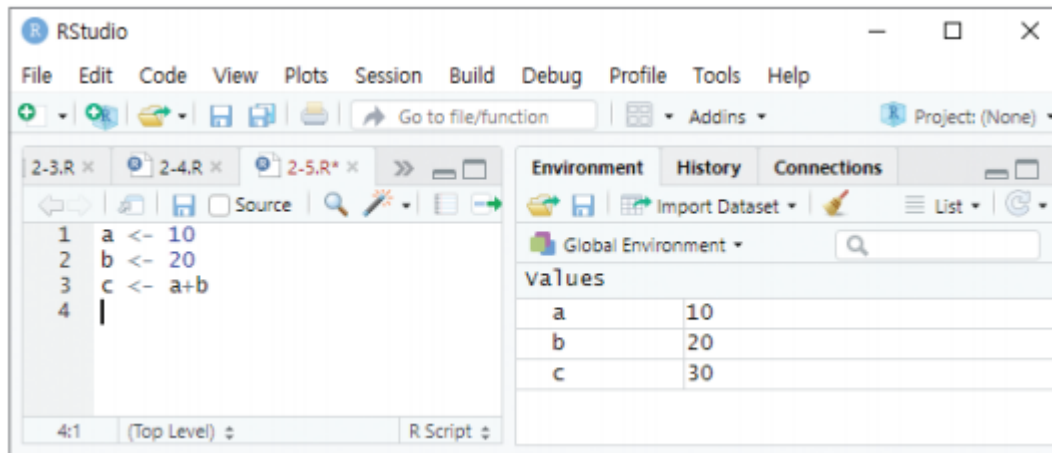
```
a <- 10      # 권장
b = 20      # 권장하지 않음
```

## 2. 변수

### 코드 2-5

```
a <- 125  
a  
print(a)
```

```
> a <- 125  
> a  
[1] 125  
> print(a)  
[1] 125
```



Alt와 - 를 누르면  
'<-'가 입력

그림 2-3 R 스튜디오의 환경 창에서 변수의 내용 확인하기

## 2. 변수

### 4. 변수의 자료형

자료형	사용 예	비고
숫자형	1, 2, 3, -4, 12.8	정수와 실수 모두 가능
문자형	'Tom', "Jane"	작은 따옴표나 큰 따옴표로 묶어서 표현
논리형	TRUE, FALSE	반드시 따옴표가 없는 대문자로 표기하며, T나 F로 줄여서 사용하는 것도 가능
특수값	NULL	정의되어 있지 않음을 의미하며, 자료형도 없고 길어도 0임
	NA	결측값(missing value)
	NaN	수학적으로 정의가 불가능한 값 예 sqrt(-3)
	Inf, -Inf	양의 무한대(Inf), 음의 무한대(-Inf)

표 2-3 R에서 사용할 수 있는 값들의 자료형

## 2. 변수

### 5. 변수의 값 변경

- 변수에 저장된 값은 언제라도 변경 가능
- 변수의 자료형은 어떤 값을 저장하는가에 따라 유동적으로 바뀜

#### 코드 2-6

```
a <- 10          # a에 숫자 저장
b <- 20
a+b              # a+b의 결과 출력
a <- "A"         # a에 문자 저장
a+b              # a+b의 결과 출력. 에러 발생
```

```
> a <- 10          # a에 숫자 저장
> b <- 20
> a+b              # a+b의 결과 출력
[1] 30
> a <- "A"         # a에 문자 저장
> a+b              # a+b의 결과 출력. 에러 발생
Error in a + b : non-numeric argument to binary operator
```

# Section 03

## 벡터의 이해

### 3. 벡터의 이해

#### 1. 벡터의 개념

- 1차원 배열 데이터

- 1학년 학생들의 몸무게 자료
- 2학년 학생들의 영어 성적 자료
- 3학년 학생들의 선호하는 색 자료

몸무게

56	67	84	63	65	56	49	90
----	----	----	----	----	----	----	----

그림 2-4 1차원 배열 데이터의 예: 몸무게

- 2차원 배열 데이터

- 4학년 학생들의 전 과목 성적
- 국가별 GDP

### 3. 벡터의 이해

전 과목 성적

이름	국어	영어	수학	물리	음악	체육	미술
김철수	90	85	60	75	65	100	90
전혜련	95	90	85	85	90	85	85
강재국	85	70	75	90	85	65	75
최원식	70	65	80	100	75	85	65
정대열	80	80	85	90	100	90	80

그림 2-5 2차원 배열 데이터의 예: 전 과목 성적



(a) 하나의 값이 저장된 변수 (b) 벡터가 저장된 변수

그림 2-6 변수와 벡터

### 3. 벡터의 이해

#### 2. 벡터 만들기

##### 코드 2-7

```
x <- c(1,2,3)           # 숫자형 벡터
y <- c("a","b","c")     # 문자형 벡터
z <- c(TRUE,TRUE, FALSE, TRUE) # 논리형 벡터
x                        # x에 저장된 값을 출력
y
z
```

```
> x <- c(1,2,3)          # 숫자형 벡터
> y <- c("a","b","c")    # 문자형 벡터
> z <- c(TRUE,TRUE, FALSE, TRUE) # 논리형 벡터
> x                      # x에 저장된 값을 출력
[1] 1 2 3
> y
[1] "a" "b" "c"
> z
[1] TRUE TRUE FALSE TRUE
```



### 3. 벡터의 이해

---

코드 2-8

```
w <- c(1,2,3, "a","b","c")  
w
```

```
> w <- c(1,2,3, "a","b","c")  
> w  
[1] "1" "2" "3" "a" "b" "c"
```

### 3. 벡터의 이해

---

#### 2.1 연속적인 숫자로 이루어진 벡터의 생성

##### 코드 2-9

```
v1 <- 50:90  
v1  
v2 <- c(1,2,5, 50:90)  
v2
```

```
> v1 <- 50:90  
> v1  
[1] 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71  
[23] 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90  
> v2 <- c(1,2,5, 50:90)  
> v2  
[1] 1 2 5 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68  
[23] 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
```

### 3. 벡터의 이해

---

#### 2.2 일정한 간격의 숫자로 이루어진 벡터 생성

코드 2-10

```
v3 <- seq(1,101,3)
v3
v4 <- seq(0.1,1.0,0.1)
v4
```

```
> v3 <- seq(1,101,3)
> v3
 [1]  1  4  7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52
[19] 55 58 61 64 67 70 73 76 79 82 85 88 91 94 97 100
> v4 <- seq(0.1,1.0,0.1)
> v4
 [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

### 3. 벡터의 이해

#### 2.3 반복된 숫자로 이루어진 벡터 생성

##### 코드 2-11

```
v5 <- rep(1,times=5)          # 1을 5번 반복
v5
v6 <- rep(1:5,times=3)        # 1에서 5까지 3번 반복
v6
v7 <- rep(c(1,5,9), times=3)  # 1, 5, 9를 3번 반복
v7
```

```
> v5 <- rep(1,times=5)          # 1을 5번 반복
> v5
[1] 1 1 1 1 1
> v6 <- rep(1:5,times=3)        # 1에서 5까지 3번 반복
> v6
[1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
> v7 <- rep(c(1,5,9), times=3)  # 1, 5, 9를 3번 반복
> v7
[1] 1 5 9 1 5 9 1 5 9
```