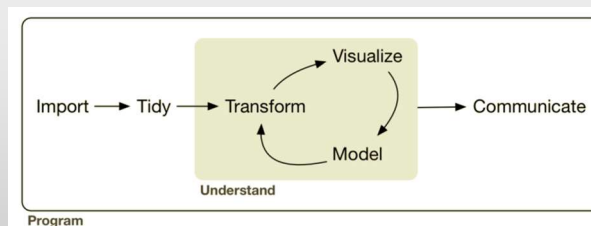


1장. R 시작하기

1.1 R의 소개

- 통계분석과 그래프 작성 등을 위한 프로그래밍 언어이자 개발 환경
- 통계 소프트웨어 중 전세계적으로 가장 많이 사용되고 있음
- 최초 개발자: Ross Ihaka, Robert Gentleman
- 오픈 소스 소프트웨어
- R Development Core Team에 의해 유지 관리
- 수많은 R 사용자들의 활발한 참여에 의한 꾸준한 발전
 - 다양한 분석기법 보유
 - 과도하게 많은 R 패키지 수로 인한 문제점

- Base R
 - 1993년 Ross Ihaka와 Robert Gentleman: R 기본 내용 공개
 - 1997년 R Core team 결성 및 CRAN 사이트 개설
 - 2000년 R 버전 1.0.0 출시
 - 2020년 R 버전 4.0.0 출시 (2023년 R 버전 4.3.1 출시)
- Tidyverse
 - 2010년부터 Hadley Wickham이 ggplot2, dplyr 등을 개발
 - R은 tidyverse 이전과 이후로 구분된다고 할 수 있음
 - 2020년 dplyr 1.0.0 출시 (2023년 현재 dplyr 1.1.2)



3

1.2 R의 설치

- R = 'base 시스템' + '패키지'
- R의 설치는 base 시스템의 설치를 의미
 - 설치 파일: Comprehensive R Archive Network (CRAN)
 - R-Project 사이트: <https://www.r-project.org>
 - CRAN 사이트: <https://www.cran.r-project.org>
 - 2023년 6월 R 버전 4.3.1 출시
- 일년에 몇 차례 업데이트 버전 출시

4

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

1: CRAN 선택

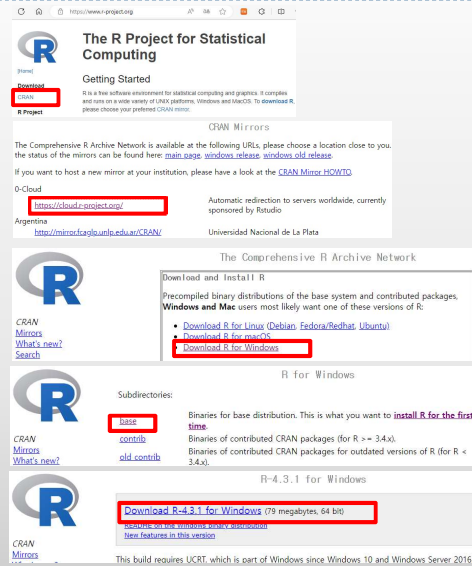
2: Mirrors 사이트 선택

3: Window용 설치파일 선택

4: base선택


5: 설치파일 다운로드

6: 설치파일 실행



5

1.3 R Studio의 설치 및 R의 실행

- R Studio 
 - R을 위한 통합 개발 환경
 - 2011년 최초 출시
 - 무료로 사용할 수 있음
 - 모든 R 작업은 R Studio에서 실행
 - 2022 10월 POSIT으로 변환
- 설치 방법
 - <https://posit.co/> 이동
 - 일년에 몇 차례 업데이트 버전 출시

6

● R Studio 설치

The most popular coding environment for R, built with love by Posit.

Used by millions of people weekly, the RStudio integrated development environment (IDE) is built to help you be more productive with R and Python. It includes a syntax-highlighting editor that supports direct code execution, a viewer for viewing history, debugging, and managing your workspace.

If you're a professional data scientist and want guidance on adapting open source software, check out our [open source adaptation guide](https://posit.co/2022/05/05/).

[Download RStudio](#) [Download RStudio Desktop](#) [Download RStudio Server](#)

RStudio Desktop

Find out more about RStudio Desktop and RStudio Desktop Pro below.

[Download RStudio](#)

RStudio requires a 64-bit operating system. If you are on a 32-bit system, you can use [an older version of RStudio](#).

Linux users may need to import [Posit's public code signing key](#) prior to installation, depending on the operating system's security policy.

OS	Download	Size	SHA-256
Windows 10/11	RSTUDIO-2022.06.1-609.EXE	212.77 MB	A8320A95
macOS 11+	RSTUDIO-2022.06.1-609.DMG	303.02 MB	2848045A

7

● R Studio 첫 화면

R version 4.2.1 (2022-06-23 ucrt) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[workspace loaded from ~/.RData]

>

Console 창
- 명령문의 입력
- 실행
- 결과 출력

8

R의 실행

- R은 기본적으로 대화형 처리방식

R Console 창에서

- 프롬프트 '>' 다음에 명령문 입력하고 Enter 키를 치면 명령문 실행
- 결과는 다음 줄에 나타남

```
> 2+3
[1] 5
> sqrt(2)
[1] 1.414214
> pi
[1] 3.141593
```

- 대화형 처리방식
 - 복잡한 구조의 데이터를 분석하는 경우
 - 중간 단계를 거쳐야 하는 분석의 경우
 - SAS 등에서 사용되는 일괄처리방식보다 더 효과적인 처리방식

9

예제 프로그램의 실행

- 예제 데이터

speed	dist	speed	dist
4	2	11	17
7	4	12	24
8	16	13	34
9	10	13	26
10	18	14	26

- 실습문제
 1. 두 변수 speed와 dist 생성 (자료 입력)
 2. 두 변수의 기술통계량 계산
 3. 두 변수의 상관계수 계산 및 산점도 작성

10

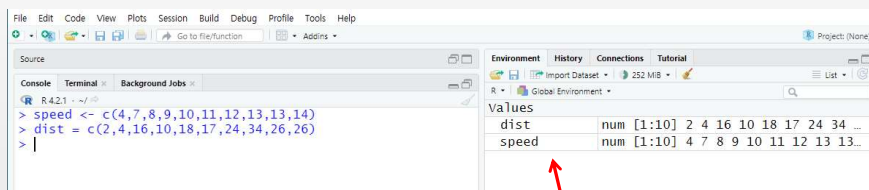
실습문제 1: 변수 생성

- 변수에 데이터 입력 → R 객체에 데이터 할당
 - 객체: R의 기본 요소
R에서 다루지는 모든 것이 객체
다양한 구조의 데이터 객체가 있음
- 데이터 입력: 함수 c()
 - R에서 사용되는 모든 명령문은 내장함수 혹은 사용자정의 함수
- 할당 기호: <- 또는 =
 - R Studio에서 할당 기호 '<-' 단축키: Alt + 마이너스 키(-)

11

- 예제 프로그램 R Studio에서 실행

```
> speed <- c(4,7,8,9,10,11,12,13,13,14)
> dist = c(2,4,16,10,18,17,24,34,26,26)
```



Environment / History 창
- 생성된 객체 정보 확인
- 입력된 명령문 확인

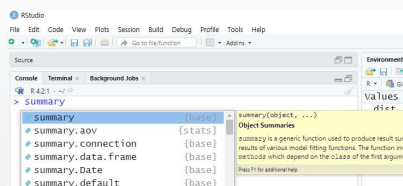
12

실습문제 2: 변수의 기술통계량 계산

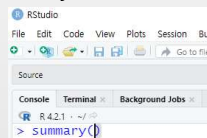
- 기술통계량 계산에 관련된 R 함수
 - `summary()` : 최소, 최대, 평균, 중앙값, 1사분위수, 3사분위수
 - `mean()` : 평균
 - `sd()` : 표준편차
- 변수 `speed`를 함수 `summary()`에 입력하고 실행
- 변수 `dist`를 함수 `mean()`과 `sd()`에 입력하고 실행

13

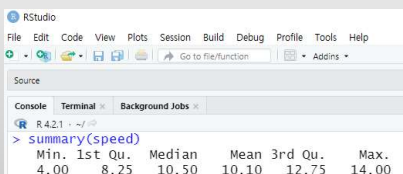
- 1) 함수 `summary` 입력
- `summary`로 시작하는 함수 나열 (선택 가능) 및 도움말 표시



- 2) `summary`에 이어서 왼쪽 괄호 '(' 입력하면 오른쪽 괄호는 자동으로 생성



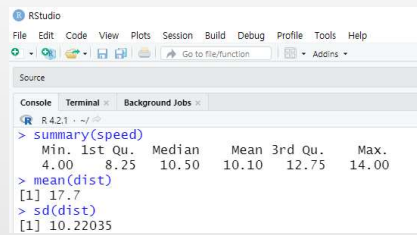
- 3) 빈 괄호에 `speed` 입력 후 실행



14

4) 아래의 명령문도 입력하고 실행

```
> mean(dist)
> sd(dist)
```



The screenshot shows the RStudio interface with the console pane active. The following commands and their outputs are visible:

```
R 4.2.1 ~ /
> summary(speed)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   4.00   8.25   10.50   10.10   12.75   14.00
> mean(dist)
[1] 17.7
> sd(dist)
[1] 10.22035
```

15

실습문제 3: 두 변수 상관관계수 계산 및 산점도 작성

- 두 변수의 상관관계수: 함수 cor()
- 두 변수의 산점도: 함수 plot()

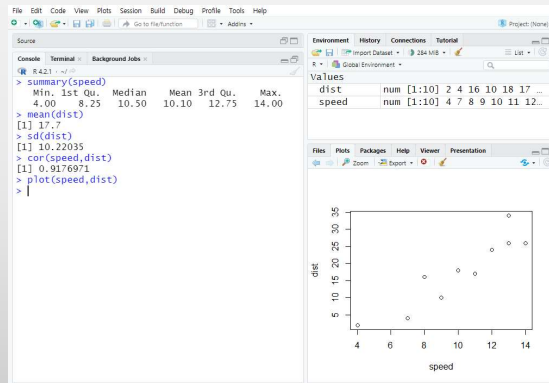
16

1) 함수 cor()로 상관계수 계산

```
> cor(speed, dist)
```

2) 함수 plot()으로 산점도 작성

```
> plot(speed, dist)
```



← 산점도 생성

17

1. 영문자 소문자와 대문자를 서로 다른 문자로 인식
2. 명령문이 완성되지 않은 상태에서 Enter 키를 치면 프롬프트가 '>' 기호에서 '+' 기호로 바뀐.
3. '#' 기호 이후의 명령문은 무시됨

```
> mean(speed)
[1] 10.1

> Mean(speed)
Error in Mean(speed) : could not find function "Mean"

> cor(dist,
+ speed)
[1] 0.9176971

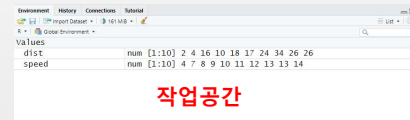
> sd(dist) # standard deviation of dist
[1] 10.22035
```

18

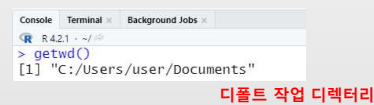
1.4 작업공간

- 작업공간: R의 실행과정에서 생성되는 모든 객체 및 입력되는 명령문 등이 보관되는 가상의 공간
 - 작업 디렉터리(폴더)에 임시로 보관

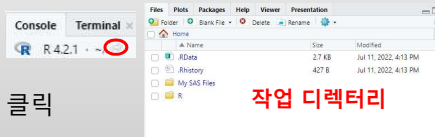
- 작업공간:
 - 임시 저장된 객체: Environment 창
 - 앞서 실행된 명령문: History 창



- 현재의 작업 디렉토리 확인:
 - 함수 `getwd()` 실행



- 작업 디렉터리의 파일 확인:
 - R Studio의 Files 창 보기
 - 또는 Console 창의 화살표 클릭

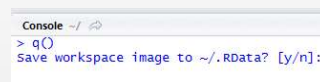


19

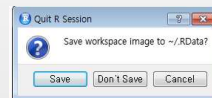
• 작업 공간의 저장

- R 세션 중 생성된 객체를 작업 디렉터리에 파일 .RData로 저장

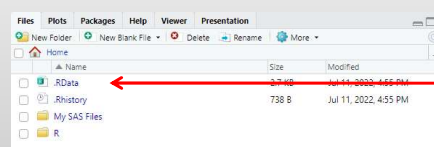
- 1) 함수 `save.image()` 실행
- 2) R 종료 시
 - 함수 `q()`로 종료 시



- 메뉴 File > Quit Session 으로 종료 시

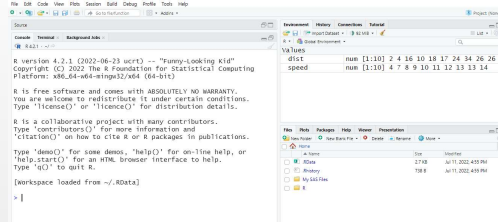


- R 세션 중 입력한 명령문은 파일 .Rhistory 로 저장되는 것이 디폴트
- 작업공간을 디폴트 작업 디렉터리에 저장한 후의 상태

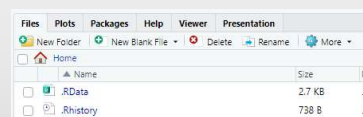


20

- 파일로 저장된 작업공간
 - 디폴트 작업 디렉토리에 저장 된 경우: R이 다시 시작되면 자동으로 올라옴



- 작업공간의 저장: 실제 상황에서는 사용하지 않는 것이 바람직함
- 현재 저장된 작업공간은 제거하기 바람
- R 종료 시 저장 여부를 묻는 과정 생략
 - Tools > Global options... > General : Workspace
 - 'Save workspace to .RData on exit'을 [Never]로 선택



21

- Windows에서의 경로 표시
C:\Users\KU\Documents
또는 C:\Users\W\KU\Documents

- R에서의 경로 표시
 - 1) C:/Users/KU/Documents
 - 2) C:\\Users\\W\\KU\\Documents

22

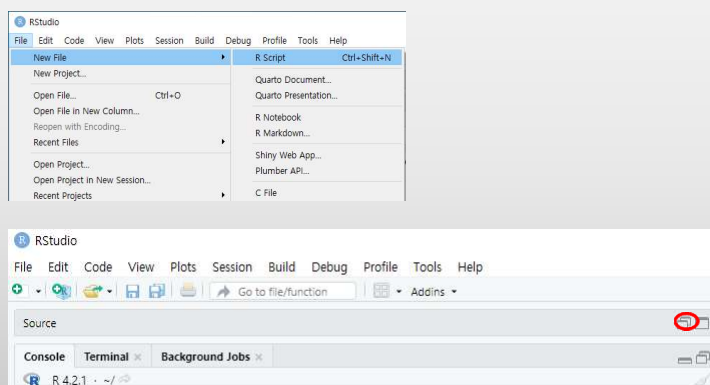
1.5 스크립트 파일의 활용

- Console 창에 명령문을 직접 입력하여 작업하는 방식
번거롭고 오류를 범할 가능성이 많다
- 스크립트 파일: R 명령문이 입력된 파일
- 스크립트 파일을 활용한 작업 방식
 1. R Script 창 활용
 2. 함수 `source()`의 활용 (생략)

23

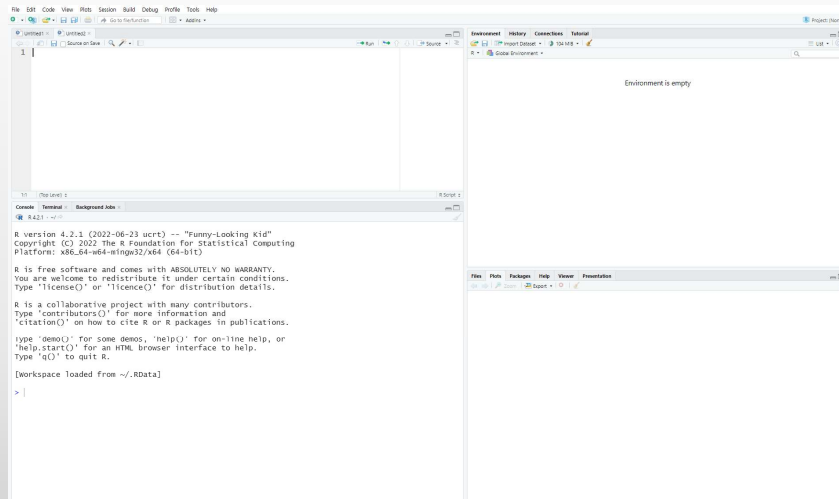
1. R Script 창 활용

- 매우 효율적이고 융통성 있는 방식
- R Script 창을 여는 방법
 - 메뉴 바에서 'File > New File > R Script' 선택
 - Ctrl+Shift+N



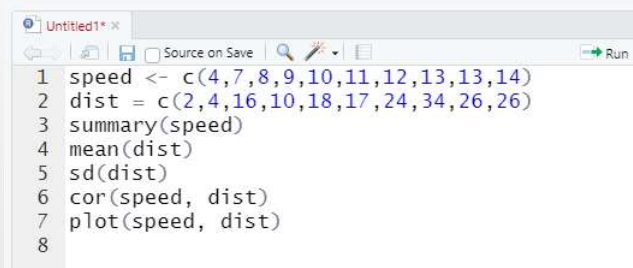
24

- R Script 창이 열린 상태




25

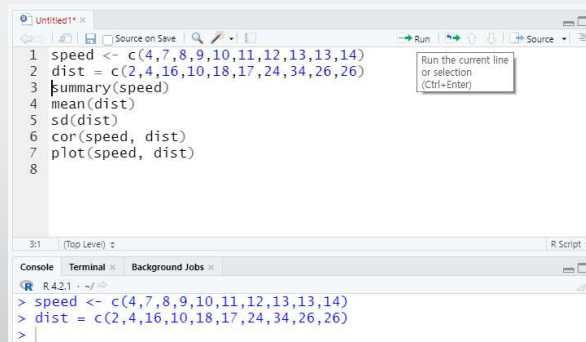
1) R Script 창에 스크립트 파일 작성



26

2) R Script 창에 입력된 명령문 실행 방법

- 한 명령문씩 차례로 실행해야 하는 경우
 - 커서를 해당되는 명령문에 놓고 Ctrl + Enter 키를 누른다.
또는 아이콘  이용.
 - 해당 명령문이 실행되고 커서는 다음 명령문으로 이동




The screenshot shows the RStudio interface. The script editor contains the following code:

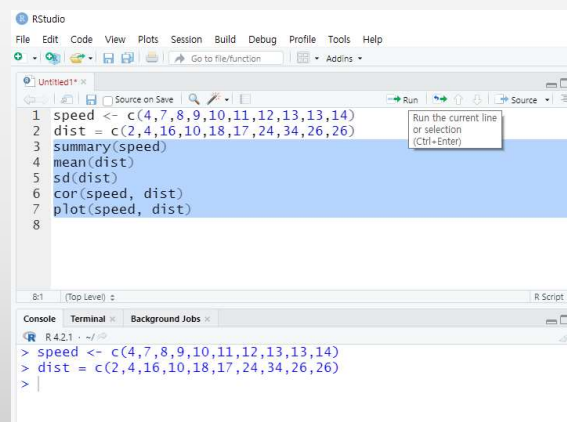
```
1 speed <- c(4,7,8,9,10,11,12,13,13,14)
2 dist = c(2,4,16,10,18,17,24,34,26,26)
3 summary(speed)
4 mean(dist)
5 sd(dist)
6 cor(speed, dist)
7 plot(speed, dist)
8
```

A tooltip indicates: "Run the current line or selection (Ctrl+Enter)". The console shows the output of the first two lines:

```
> speed <- c(4,7,8,9,10,11,12,13,13,14)
> dist = c(2,4,16,10,18,17,24,34,26,26)
>
```

27

- 몇 개의 명령문을 한꺼번에 실행해야 하는 경우
 - 마우스로 해당 명령문들을 선택하고 Ctrl + Enter 키를 누른다.
(또는 아이콘  이용.)
 - 선택된 명령문들이 모두 실행된다.

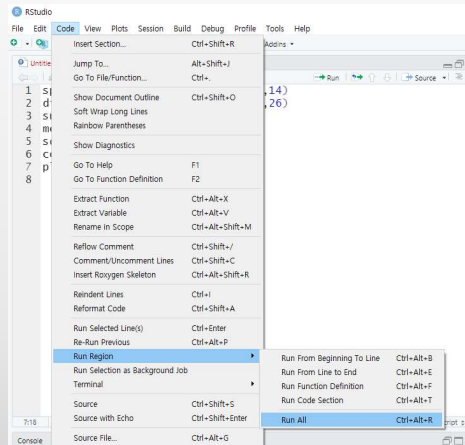


The screenshot shows the RStudio interface. The script editor contains the same code as the previous slide. Lines 1 through 7 are selected (highlighted in blue). A tooltip indicates: "Run the current line or selection (Ctrl+Enter)". The console shows the output of all selected lines:

```
> speed <- c(4,7,8,9,10,11,12,13,13,14)
> dist = c(2,4,16,10,18,17,24,34,26,26)
> summary(speed)
> mean(dist)
> sd(dist)
> cor(speed, dist)
> plot(speed, dist)
>
```

28

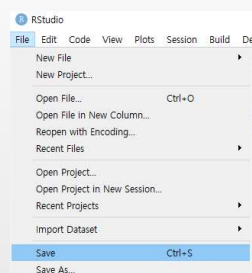
- R Script 창의 모든 명령문을 실행해야 하는 경우
 - Ctrl + Alt + R 키



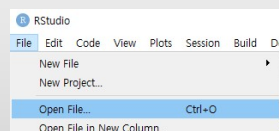
29

3) 작성된 스크립트 파일의 저장 및 열기

- 스크립트 파일의 저장
 - 메뉴에서 'File > Save' (Ctrl + S 키) 혹은 'File > Save As' 선택
 - 원하는 디렉터리 선택 후 파일 이름 지정
 - 파일 이름에 점(.)을 포함시키지 않음
 - R 스크립트 파일의 디폴트 확장자 : .R



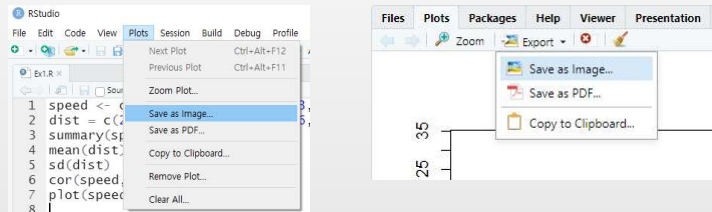
- 저장된 스크립트 파일의 열기
 - 메뉴에서 'File > Open File' (Ctrl + O 키) 선택한 후, 저장된 폴더로 이동하여 파일열기를 선택



30

Plots 창에 그려진 그래프 복사 혹은 저장 방법

1. 메뉴 Plots > Save as Image, Save as PDF, Copy to Clipboard 중 선택
2. Plots 창 메뉴 Export 이용



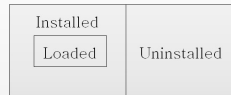
31

1.7 R의 확장: 패키지

- R 막강 파워의 원천
 - 패키지: 사용자가 작성한 R 함수, 데이터 등을 모아 놓은 것
- CRAN 사이트에 현재 15300여 개의 패키지가 있음
 - 계속 증가하고 있음
 - 다양한 분석도구 제공
 - 필요한 분석기법이 어떤 패키지에 있는지 알아내는 것이 쉽지 않음
 - 간혹 문제가 있는 패키지가 있을 수도 있다는 점을 항상 유의

32

1.7.1 패키지의 종류



- R을 설치할 때 자동으로 설치되는 패키지
 - base 패키지: R의 base 시스템의 일부분처럼 작동 (install & load)
base, datasets, graphics, grid, methods, stats, utils 등이 포함
 - recommended 패키지: 자동으로 설치(install)되지만, 사용하기 위해서는 R 세션으로 따로 불러들여야(load) 함.
MASS, foreign, lattice 등이 포함
- 그 외의 다른 패키지:
 - 사용자가 직접 설치하고 R 세션으로 불러들여야 사용할 수 있음
 - tidyverse(dplyr, ggplot2, ...)

33

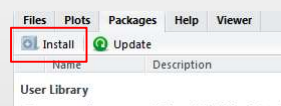
1.7.2 패키지의 설치 및 사용

- 패키지 설치: 예) 패키지 UsingR 설치

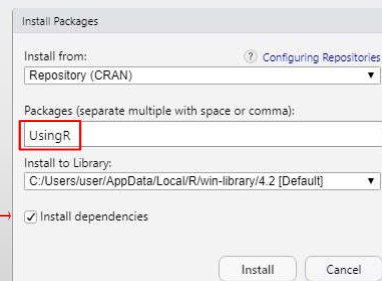
- 1) install.packages() 사용

`> install.packages("UsingR")` 패키지 이름 양쪽에 인용부호

- 2) Packages 창의 Install 이용



해당 패키지를 사용하기 위해
필요한 다른 패키지도 설치 →



34

- 패키지의 사용(load): 함수 `require()` 혹은 `library()`

```
> require(UsingR)

혹은

> library(UsingR)
```

인용부호 없이 패키지 이름을 지정

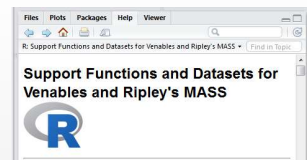
Name	Description	Version
<input type="checkbox"/> stats4	Statistical functions using 54 classes	4.2.1
<input type="checkbox"/> stringi	Character String Processing Facilities	1.7.6
<input type="checkbox"/> stringr	Simple, Consistent Wrappers for Common String Operations	1.4.0
<input checked="" type="checkbox"/> survival	Survival Analysis	3.3-1
<input type="checkbox"/> tcltk	Tcl/Tk Interface	4.2.1
<input type="checkbox"/> tibble	Simple Data Frames	3.1.7
<input type="checkbox"/> tools	Tools for Package Development	4.2.1
<input type="checkbox"/> translations	The R Translations Package	4.2.1
<input checked="" type="checkbox"/> UsingR	Data Sets, Etc. for the Text "Using R for Introductory Statistics", Second Edition	2.0-7

패키지 이름의 box 체크

35

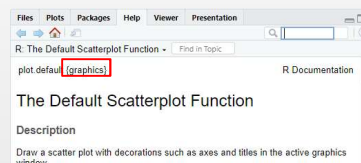
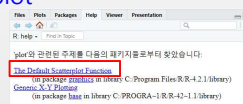
- 패키지에 대한 도움말 얻기: `help()` 사용

```
> help(package = MASS)
```



- 함수(명령문)에 대한 도움말 얻기

```
> help(plot) 또는 ? plot
```

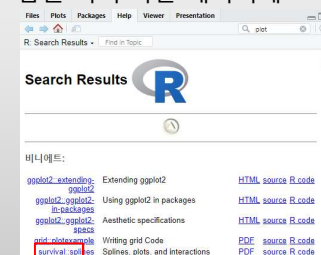


- 조금 더 포괄적인 도움말 얻기: 문자열 `plot`이 포함된 여러 다른 패키지에 대한 다양한 함수들에 대한 도움

```
> help.search("plot")
```

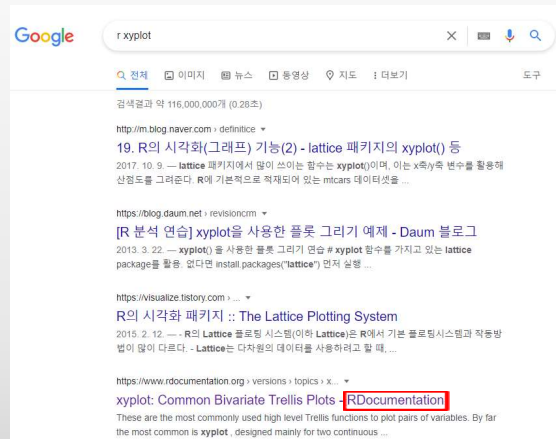
```
> ?? plot
```

- Help 탭에서 search 창에 "plot" 입력



36

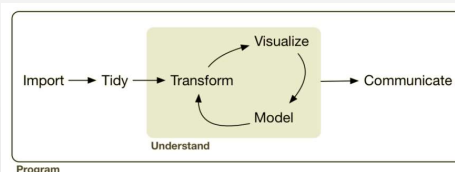
Googling에 의한 도움말 찾기



37

1.7.3 패키지 tidyverse의 소개

- Data science를 위해 개발된 패키지들의 묶음
 - 공통된 분석 방식 공유
 - 자료분석에 필요한 모든 과정이 망라되어 있음



- Core tidyverse
 - 자료 불러오기: **readr**
 - 분석하기 편리한 형태의 자료 만들기: **tibble**, **tidyr**
 - 자료 다루기: **dplyr**, **stringr**, **forcats**
 - 자료 시각화: **ggplot2**
 - 함수형 프로그래밍: **purrr**

38

tidyverse의 설치 및 사용

- tidyverse의 모든 패키지의 설치: `install.packages("tidyverse")`
- tidyverse의 패키지 로딩: `library(tidyverse)`

```
> library(tidyverse)
— Attaching packages — tidyverse 1.3.1 —
✓ tibble 3.1.7      ✓ dplyr 1.0.9
✓ tidyr 1.2.0      ✓ stringr 1.4.0
✓ readr 2.1.2      ✓ forcats 0.5.1
✓ purrr 0.3.4
— Conflicts — tidyverse_conflicts() —
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag() masks stats::lag()
✖ dplyr::select() masks MASS::select()
✖ dplyr::src() masks Hmisc::src()
✖ dplyr::summarize() masks Hmisc::summarize()
```

- Core tidyverse 패키지가 모두 세션에 올라옴
- Core tidyverse에 속하지 않은 패키지는 따로 함수 `library()`로 불러와야 함

39

연습

두 가지 교육방법(기존방법, 새방법) 자료를 생성하시오

사원 방법	1	2	3	4	5	6	7
기존방법	78.5	60.3	81.7	69.0	64.0	62.6	86.7
새방법	82.0	74.9	88.1	62.1	78.5	79.9	94.4

1. 기존방법과 새 방법의 평균, 분산, 최고치를 구하시오
2. 기존방법의 5% 절사평균을 구하시오
3. x축: 기존방법 y축: 새방법 산점도를 그리시오
4. 함수 `xyplot`을 이용하여 산점도를 그리시오.
5. 문제1,2,3,4의 결과를 담는 한글파일을 만드시오.
6. 문제1,2,3,4의 결과를 수행할 수 있는 파일을 생성, 저장하고 수행하시오.

40