



FMBA 2018 사전 교육

프로그래밍(R)

KAIST

경영공학과 석박사통합과정 유승현

rambor12@business.kaist.ac.kr

목 차

1. 개요
2. R 설치
3. RStudio 인터페이스
4. R 연산
5. R 프로그래밍
6. R 응용 : 회귀분석

Github Repository

□ 프로그래밍 교육 자료

https://github.com/Seung-hyeon/KAIST_FMBA_Programming_2018

This repository Search Pull requests Issues Marketplace Explore

Seung-hyeon / KAIST_FMBA_Programming_2018 Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

KAIST FMBA Programming Prior-Education Materials Edit

Add topics

5 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Seung-hyeon some files		Latest commit 871ad68 2 minutes ago
Excel VBA	some files	2 minutes ago
Matlab	some files	2 minutes ago
R	some files	2 minutes ago
README.md	some files	2 minutes ago

README.md

KAIST FMBA Programming 2018

This is KAIST FMBA Programming Prior-Education Materials.

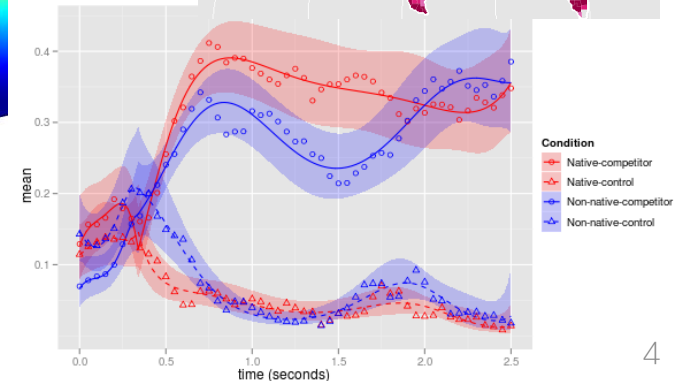
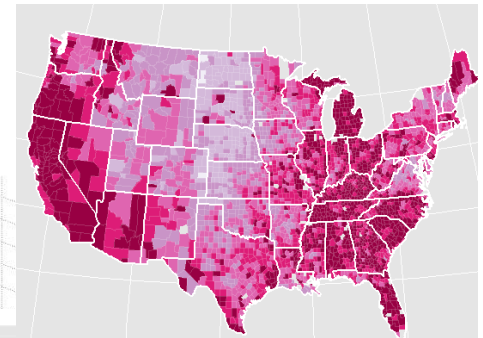
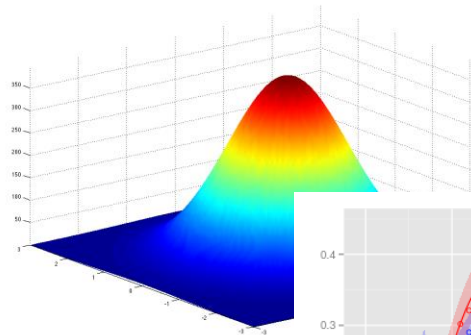
This materials cover Excel VBA , Matlab , and R .

자료 다운로드

1. 개요

□ (R)

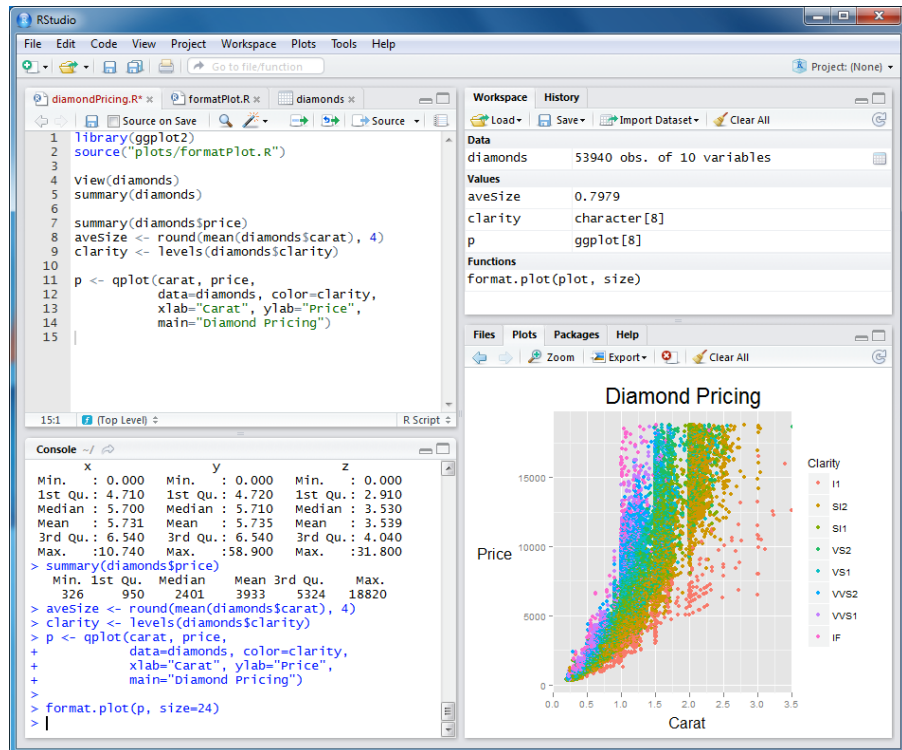
- ✓ GNU GPL(일반 공중 사용 허가서)하에 배포되는 무료 통계 프로그래밍 언어
- ✓ 사용자의 입력에 대한 결과 즉시 확인 가능 (VBA와 다름)
- ✓ 강력한 시각화 그래프 제공
- ✓ 다양한 패키지 제공 (CRAN)
 - ggplot2
 - polynom
 - tseries
 - quantmod
 - stringr
 - ...
- ✓ 주로 통계학, 데이터 분석에 많이 사용



1. 개요

□ R Studio® (RStudio)

- ✓ R 기본편집기의 부족한 부분 보완
- ✓ 실시간으로 프로그램 실행 결과, 변수 값 확인 가능
- ✓ 자동 명령어 완성
- ✓ 코드 구문 강조(색깔)
- ✓ 버전, 프로젝트 관리



2. R 설치

□ R 설치 방법

① R 다운로드 홈페이지 접속

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

② "Download R X.X.X for Windows" 클릭, 바탕화면에 저장

R-3.3.2 for Windows (32/64 bit)

Download R 3.3.2 for Windows (62 megabytes, 32/64 bit)

[Installation and other instructions](#)
[New features in this version](#)

If you want to double-check that the package you have downloaded exactly matches the [command line versions](#) are available.

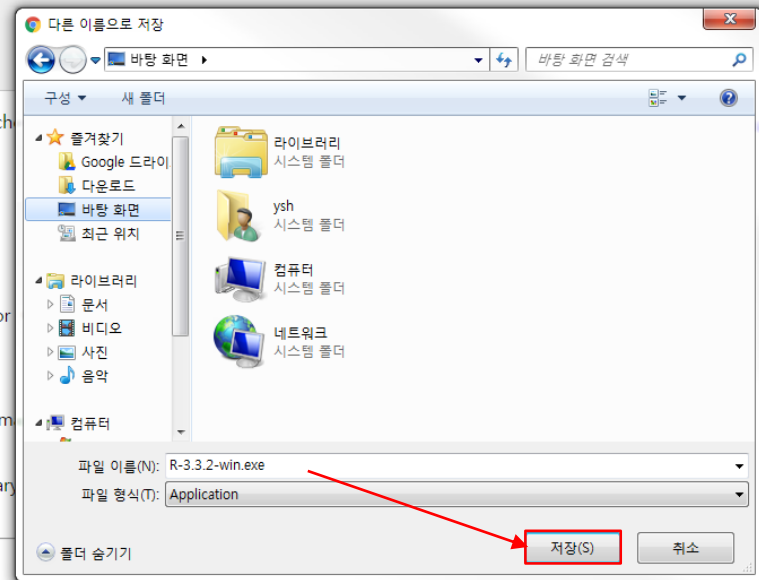
- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for

- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary is <https://cran.r-project.org/bin/windows/base/release.htm>.

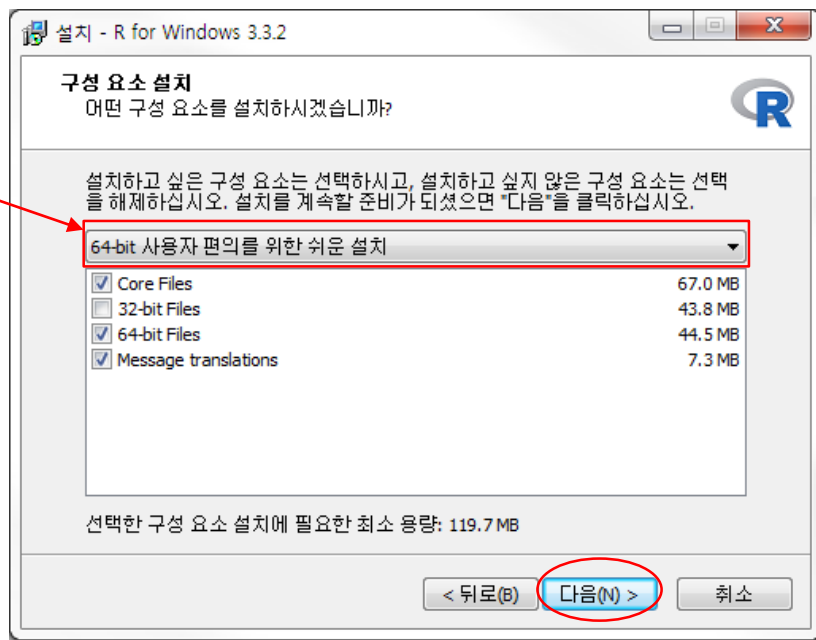
Last change: 2016-10-31, by Duncan Murdoch



2. R 설치

□ R 설치 방법

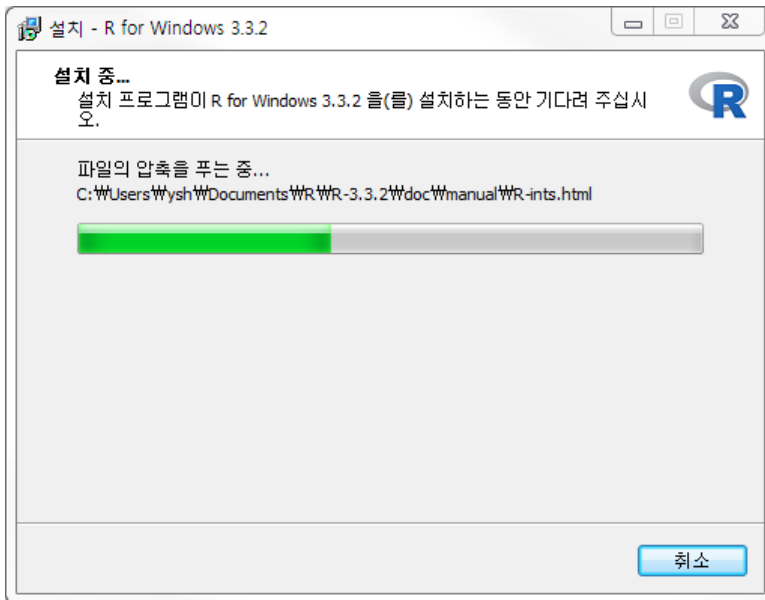
- ③ 다운받은 파일 실행, "한국어"로 설치
- ④ 운영체제 환경에 맞게 64-bit or 32-bit 맞춰서 설치



2. R 설치

□ R 설치 방법

- ⑤ 계속 "다음" 버튼 눌러 설치
- ⑥ 설치 끝난 후 "완료" 클릭



2. R 설치


□ RStudio 설치 방법 (주의: 계정 이름이 한글이면 에러 발생)

① RStudio 다운로드 홈페이지 접속

<https://www.rstudio.com/products/rstudio/download/>

② 맨 왼쪽 "DOWNLOAD" 클릭 후 윈도우버전 Installer 다운로드

	RStudio Desktop Open Source License	RStudio Desktop Commercial License	RStudio Server Open Source License	RStudio Commercial License
	FREE	\$995 per year	FREE	\$9,99
Integrated Tools for R	●	●	●	
Priority Support		●		
Access via Web Browser			●	
Enterprise Security				
Project Sharing				
Manage Multiple R Sessions & Versions				
Admin Dashboard				
Load Balancing				
License	AGPL	Commercial	AGPL	Commercial
Pricing	FREE	\$995/yr	FREE	\$9,995
	DOWNLOAD	BUY NOW	DOWNLOAD	DOWNLOAD

rstudio::conf Products Resources

RStudio Desktop 1.0.136 — Release Notes

RStudio requires R 2.11.1+. If you don't already have R, download it [here](#).

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.0.136 - Windows Vista/7/8/10	81.9 MB	2016-12-21	938f...
RStudio 1.0.136 - Mac OS X 10.6+ (64-bit)	71.2 MB	2016-12-21	12c...
RStudio 1.0.136 - Ubuntu 12.04+/Debian 8+ (32-bit)	85.5 MB	2016-12-21	0a2...
RStudio 1.0.136 - Ubuntu 12.04+/Debian 8+ (64-bit)	92.1 MB	2016-12-21	2a7...
RStudio 1.0.136 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	84.7 MB	2016-12-21	fa6...
RStudio 1.0.136 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	85.7 MB	2016-12-21	2b0...

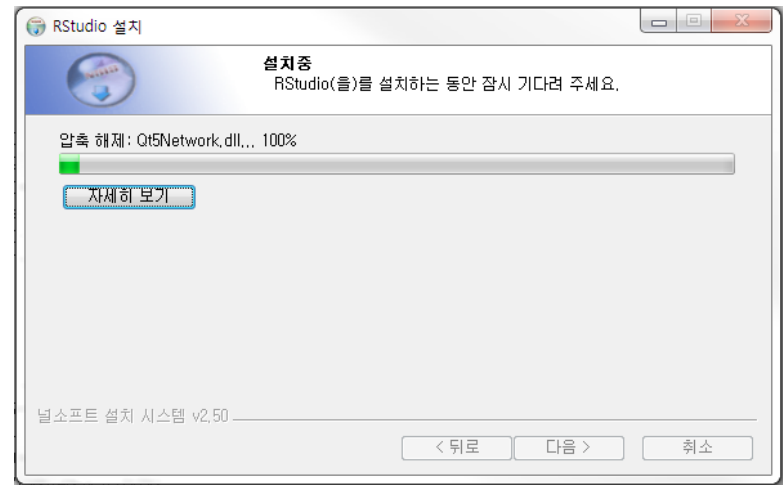
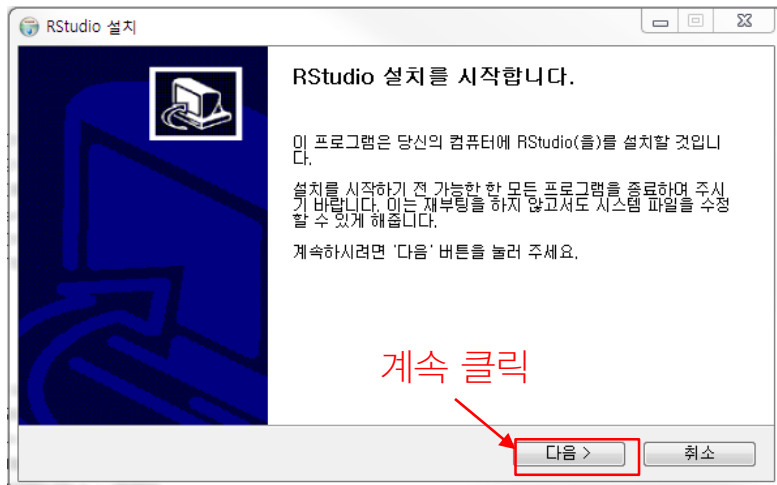
Zip/Tarballs

Zip/tar archives	Size	Date	MD5
RStudio 1.0.136 - Windows Vista/7/8/10	117.5 MB	2016-12-21	f4...
RStudio 1.0.136 - Ubuntu 12.04+/Debian 8+ (32-bit)	86.2 MB	2016-12-21	fc...
RStudio 1.0.136 - Ubuntu 12.04+/Debian 8+ (64-bit)	93.2 MB	2016-12-21	7c...

2. R 설치

□ RStudio 설치 방법

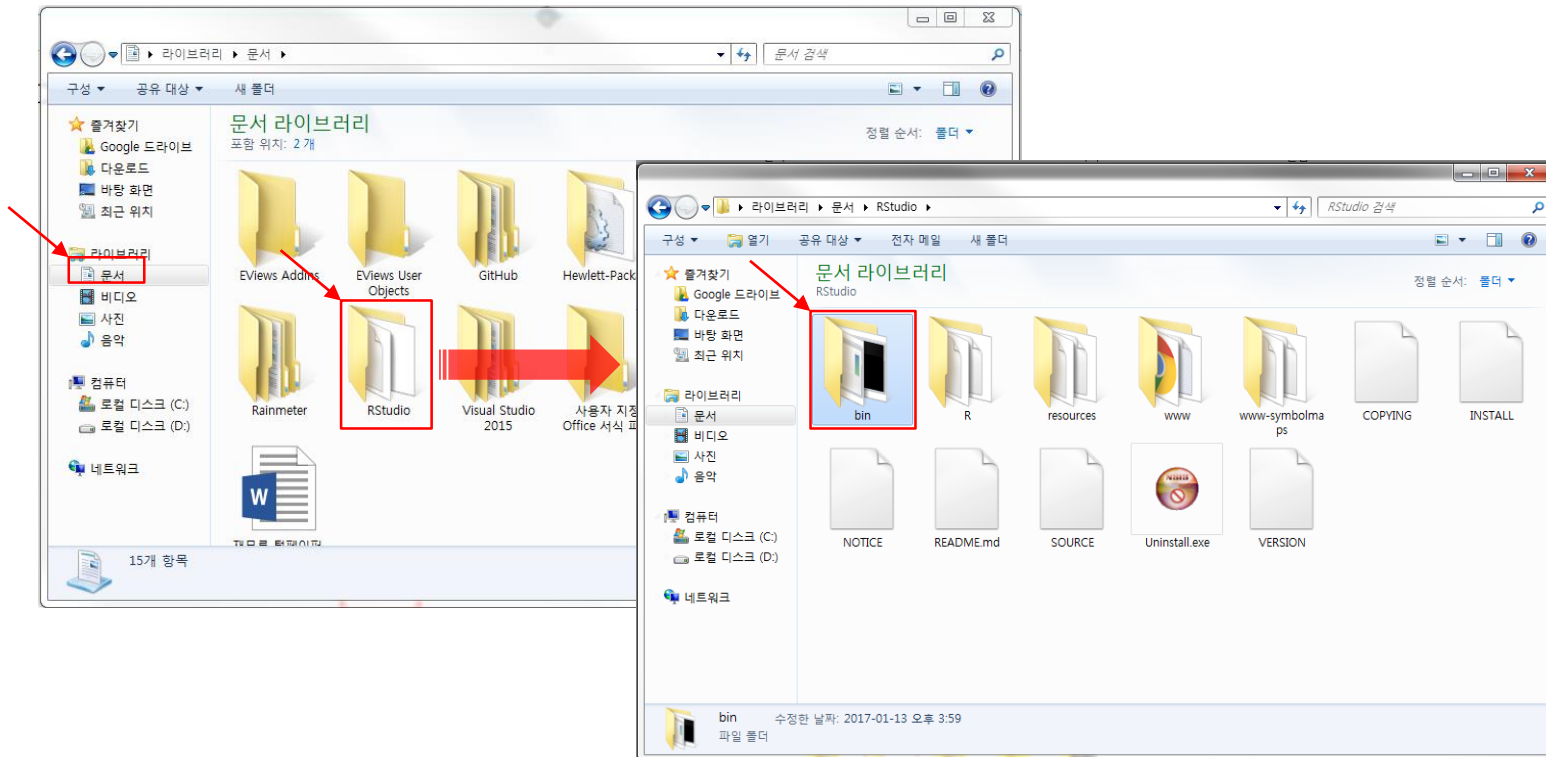
- ④ 다운로드 받은 파일 실행
- ⑤ 계속 "다음" 버튼 눌러 설치
- ⑥ 마지막에 "마침" 클릭



2. R 설치

□ RStudio 설치 방법

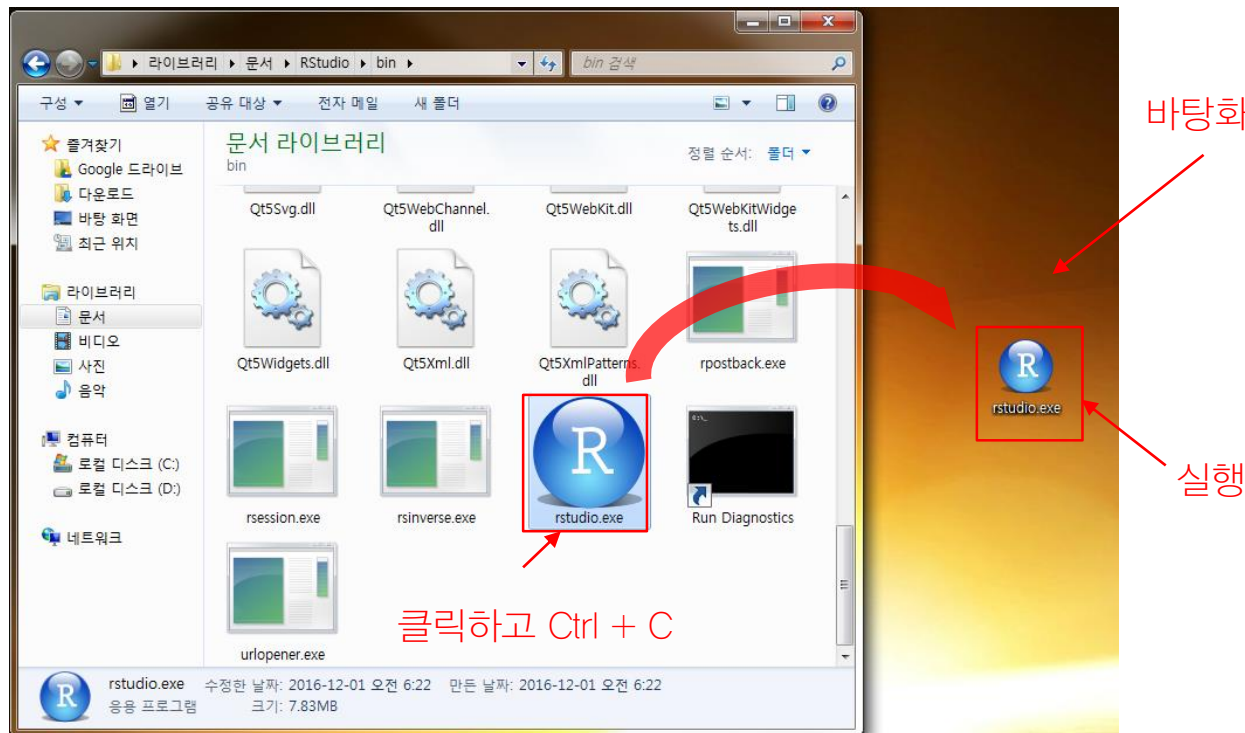
- ⑦ "문서" 폴더로 들어가기
- ⑧ "RStudio" -> "bin" 폴더 들어가기



2. R 설치

□ RStudio 설치 방법

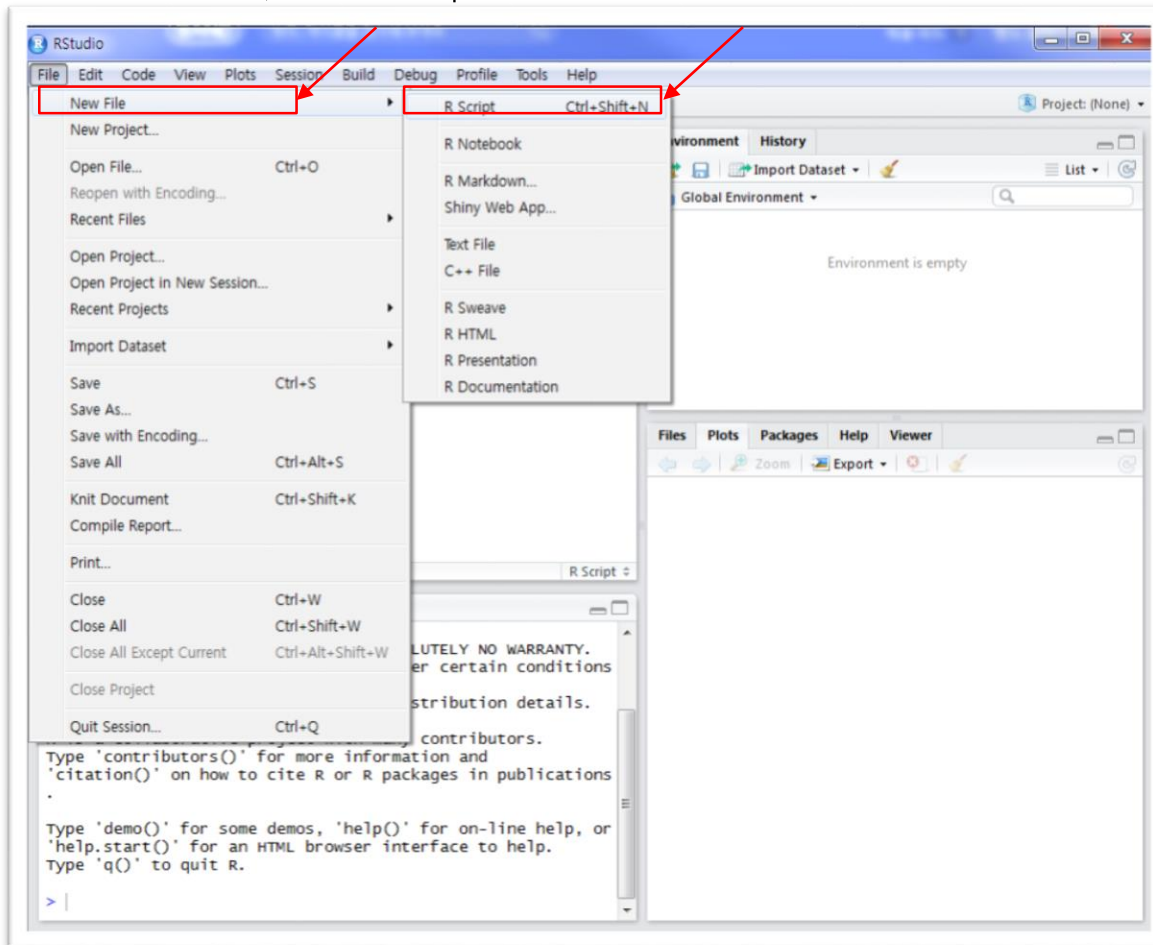
- ⑨ "bin"폴더 맨 아래 "rstudio.exe"를 바탕화면에 복사하기
- ⑩ 바탕화면의 "rstudio.exe" 실행 (앞으로 여기서 계속 실행하면 됨)



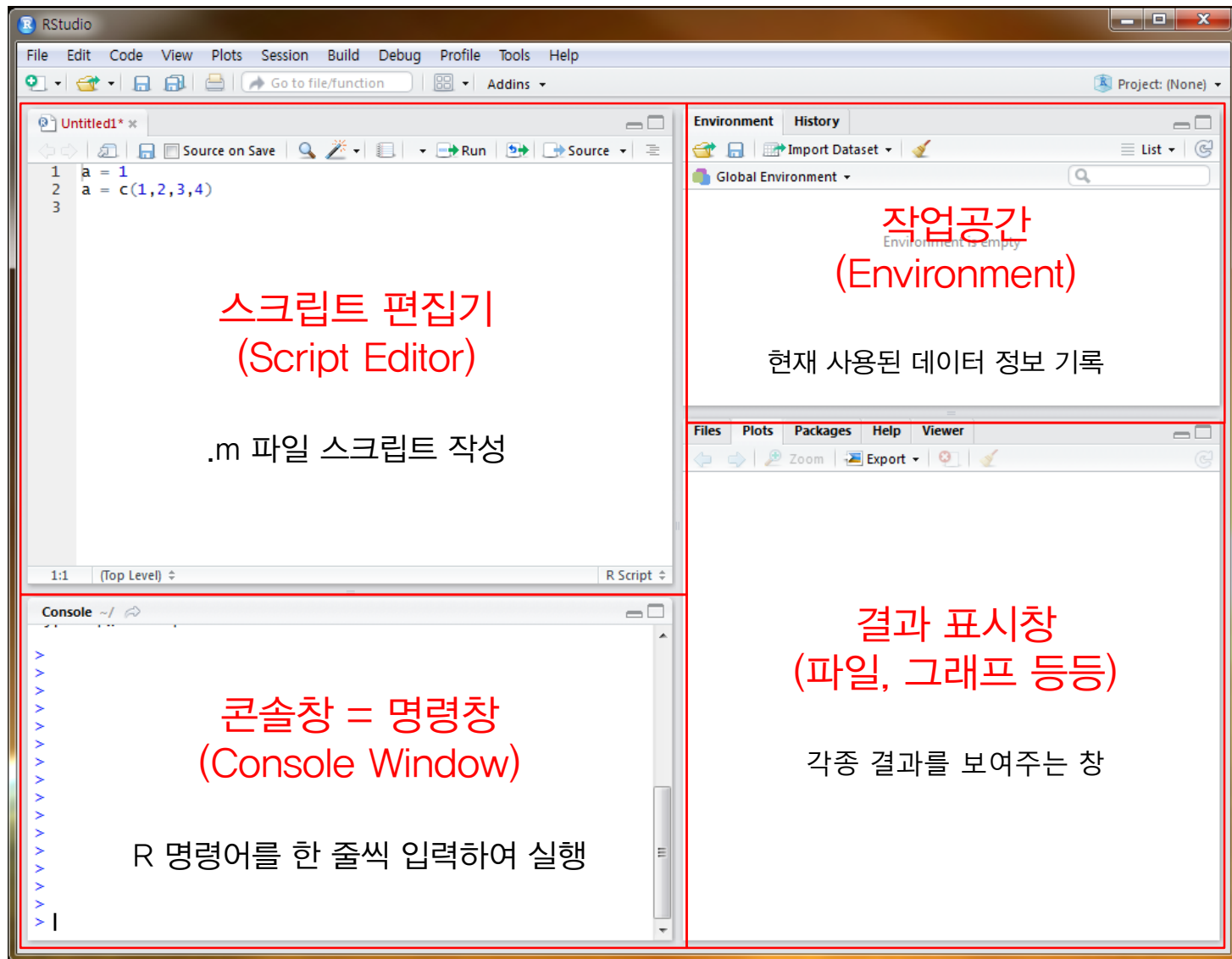
2. R 설치

□ RStudio 설치 방법

- ⑪ "New File" → "R Script" 클릭하여 스크립트 만들기



3. RStudio 인터페이스



3. R 인터페이스

□ 기본 데이터셋

data()

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
InsectSprays	Effectiveness of Insect Sprays
JohnsonJohnson	Quarterly Earnings per Johnson & Johnson Share
LakeHuron	Level of Lake Huron 1875-1972
LifeCycleSavings	Intercountry Life-Cycle Savings Data
Loblolly	Growth of Loblolly pine trees
Nile	Flow of the River Nile
Orange	Growth of Orange Trees
OrchardSprays	Potency of Orchard Sprays
PlantGrowth	Results from an Experiment on Plant Growth
Puromycin	Reaction Velocity of an Enzymatic Reaction
Seatbelts	Road Casualties in Great Britain 1969-84
Theoph	Pharmacokinetics of Theophylline
Titanic	Survival of passengers on the Titanic
ToothGrowth	The Effect of Vitamin C on Tooth Growth in Guinea Pigs
UCBAdmissions	Student Admissions at UC Berkeley
UKDriverDeaths	Road Casualties in Great Britain 1969-84
UKgas	UK Quarterly Gas Consumption
USAccDeaths	Accidental Deaths in the US 1973-1978
USArrests	Violent Crime Rates by US State
USJudgeRatings	Lawyers' Ratings of State Judges in the US Superior Court
USPersonalExpenditure	Personal Expenditure Data
UScitiesD	Distances Between European Cities and Between US Cities

3. R 인터페이스

□ 기본 명령어

대입연산자(<-) : 오른쪽 값을 왼쪽에 대입 ('='과 동일한 기능)

```
# 값 대입  
a = 5 #or  
a <- 5
```

```
# R 도움말 이용하기  
help([물어볼 대상])
```

```
# R 종료하기  
quit() #or  
q()
```

```
# 정의했던 변수 나열하기  
ls()
```

```
# 타입 확인하기  
typeof([확인할 대상])
```


4. R 연산

□ 수식 표현

$$8 + \frac{6 \times (1 + \sqrt{5})}{2} \quad \gg 8 + (6 * (1 + \text{sqr t}(5))) / 2$$

$$\frac{4}{3} \times \pi \times 10^3 \quad \gg 4 / 3 * \text{pi} * 10^3$$

$$e^{\sqrt{3}} + \pi \times 10^3 \quad \gg \text{exp}(\text{sqr t}(3)) + \text{pi} * 10^3$$

$$e^{\pi \times \sqrt{-1}} \quad \gg \text{exp}(\text{pi} * 1i)$$

R에서는 허수를 **1i** 라고 씀

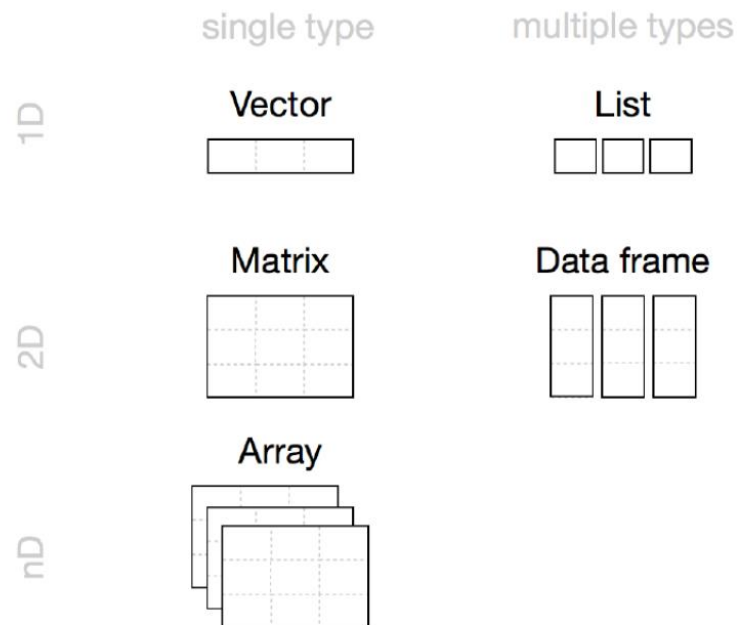
- ① π 는 pi로 기술
- ② 자연상수 e 는 exp(1)로 기술, e^x 는 exp(x)로 기술
- ③ 1i는 허수 단위(imaginary number unit)

4. R 연산

□ R 기초 문법

1. 변수의 선언 : (Declaration)

다양한 데이터 타입 : 리스트, Data Frame



4. R 연산

벡터(배열) 생성

'c()' 함수로 벡터를 생성하면 행, 열 구분 없음!
계산시 필요할 때 행, 열 자동 변환

```
> # 벡터(배열) 생성
> a = c(1, 2, 3)
> a
[1] 1 2 3
>
> # 열(column) 벡터 생성
> cbind(a)
      a
[1,] 1
[2,] 2
[3,] 3
>
> # 행(row) 벡터 생성
> rbind(a)
      [,1] [,2] [,3]
a       1    2    3
```

벡터 변환(Transpose) : 't()' 사용하기

```
> # 벡터 transpose
> t(a)
      [,1] [,2] [,3]
[1,]     1     2     3
```

벡터 원소 참조 브라켓[] 사용

```
> # 벡터 원소 참조
> a[1]
[1] 1
>
> a[2]
[1] 2
>
> # 여러 원소 참조
> a[2:3]
[1] 2 3
>
> # 해당 원소 제외하고 참조
> a[-1]
[1] 2 3
```

다양한 원소 추출 방법

```
> # 1. TRUE, FALSE를 이용한 추출
> a[c(TRUE,FALSE,TRUE)]
[1] 1 3
>
> a[c(FALSE,TRUE,FALSE)]
[1] 2
>
> # 2. 인덱스 번호 배열을 이용한 추출
> a[c(1,3)]
[1] 1 3
>
> # 3. 논리 조건을 이용한 추출
> a[a/2+1 > 2]
[1] 3
>
> a[a%%3 == 1]
[1] 1
```

나머지 연산자

4. R 연산

순차적인 벡터 생성

```
> # 순차적인 벡터 생성
> a = 1:3
> a
[1] 1 2 3
>
>
> # 간격이 정해져있는 벡터
> a = seq(1, 2, 0.2)
> a
[1] 1.0 1.2 1.4 1.6 1.8 2.0
```

시작 값

마지막 값

간격의 길이

4. R 연산

□ 단일 벡터 연산

```
> exp(a)
[1] 2.718282 7.389056 20.085537
>
> log(a)
[1] 0.0000000 0.6931472 1.0986123
>
> sqrt(a)
[1] 1.0000000 1.414214 1.732051
>
> abs(a)
[1] 1 2 3
>
> a^2
[1] 1 4 9
```

각 벡터의 원소 x 에 대해
 $e^{\sqrt{3}x} + \pi \times 10^{-1} + x$ 계산

```
> exp(sqrt(3)*a) + pi*10^(-1) + a
[1] 6.966393 34.261905 183.890282
```

벡터 길이 확인

```
> length(a)
[1] 3
```

기타 : min, max, sum, mean, var, sd 등도 가능!

4. R 연산

□ 벡터 간 연산

```
> a = c(1, 2, 3)
> b = c(4, 5, 6)
>
> # 벡터의 합과 차
> a + b
[1] 5 7 9
>
>
> a - b
[1] -3 -3 -3
```

재사용규칙(Recycling Rule)
벡터 길이가 다를때 짧은 길이의
벡터를 반복적으로 사용

```
> # 재사용규칙(Recycling Rule)
> c(1, 2, 4) + c(3, 5, 8, 2, 0)
[1] 4 7 12 3 2
Warning message:
In c(1, 2, 4) + c(3, 5, 8, 2, 0) :
  longer object length is not a multiple of shorter object length
```

```
> # 벡터의 내적
> a %*% b
      [,1]
[1,]    32
> rbind(a) %*% cbind(b)
      b
a 32
>
> # 벡터의 외적(Cross Product)
> crossprod(a, b) # 이게 아님!
```

```
      [,1]
[1,]    32
```

주의! R에선 Cross Product 구하는
기본 함수가 없음

```
> # 벡터의 외적(Outer Product)
> a %o% b #or
      [,1] [,2] [,3]
[1,]    4    5    6
[2,]    8   10   12
[3,]   12   15   18
> cbind(a) %*% rbind(b)
      [,1] [,2] [,3]
[1,]    4    5    6
[2,]    8   10   12
[3,]   12   15   18
```

4. R 연산

□ 행렬 생성

행렬 생성

```
> # 방법 1
> M = matrix(c(1,2,3,4,5,6,7,8,9),
+           nrow=3, ncol=3, byrow=TRUE)
```

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

TRUE면 row 먼저 채움

```
> # 방법 2 row 개수 하나만 지정
> M = matrix(c(1,2,3,4,5,6,7,8,9), nrow=3)
```

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

```
> # 행렬 원소 참조
```

```
> M[1,3]
[1] 7
```

→ 인덱스는 1부터 시작

특수 행렬 생성

```
> m = 5
> n = 3
> # 대각선 원소가 1로 구성된 mxm 정방행렬
> diag(m)
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    0    1    0    0    0
[3,]    0    0    1    0    0
[4,]    0    0    0    1    0
[5,]    0    0    0    0    1
```

```
> # 모든 원소가 0인 mxn 행렬 생성
> matrix(0L, nrow=m, ncol=n)
```

```
      [,1] [,2] [,3]
[1,]    0    0    0
[2,]    0    0    0
[3,]    0    0    0
[4,]    0    0    0
[5,]    0    0    0
```

```
> # 모든 원소가 1인 mxn 행렬 생성
> matrix(1L, nrow=m, ncol=n)
```

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

4. R 연산

□ 행렬 연산

```
> # 행렬 차원 확인
> dim(M)
[1] 3 3
```

역행렬 계산

```
> # 역행렬 계산
> solve(M)
Error in solve.default(M) :
  Lapack routine dgesv: system is exactly singular: U[3,3] = 0
> M
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

```
> M
      [,1] [,2] [,3]
[1,]    5    1    2
[2,]    7    3    5
[3,]    1   84   12
> # 역행렬 계산
> solve(M)
      [,1]      [,2]      [,3]
[1,] 0.46320869 -0.18817853 0.001206273
[2,] 0.09529554 -0.06996381 0.013268999
[3,] -0.70566948 0.50542823 -0.009650181
```


3. R 인터페이스

□ **예제1** 다음 행렬의 곱을 구하여라.


$$\begin{pmatrix} 2 & 4 & 9 \\ 1 & 2 & 1 \\ 3 & 0 & 2 \end{pmatrix} \times \begin{pmatrix} 5 & 5 & 2 \\ 2 & 2 & 1 \\ 1 & 3 & 3 \end{pmatrix} = ?$$

$= A \qquad \qquad \qquad = B$

Sample Code

```
> # 행렬 A 생성
> A = matrix(c(2,4,9,1,2,1,3,0,2), nrow=3, byrow = TRUE)
>
> # 행렬 B 생성
> B = matrix(c(5,5,2,2,2,1,1,3,3), nrow=3, byrow = TRUE)
>
> # 행렬 곱 실행
> A %*% B
      [,1] [,2] [,3]
[1,]   27  45  35
[2,]   10  12   7
[3,]   17  21  12
```

행렬 생성



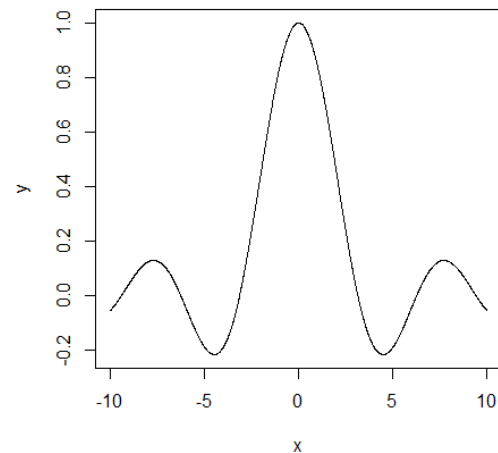
5. R 프로그래밍

□ 함수 만들기

```
# 함수 만들기  
[함수명] = function(인수1, 인수2...){  
  [명령문]  
  return(결과)  
}
```

예제2 $\sin(x)/x$ 그리는 'myfunction' 함수를 만들고 실행하라.

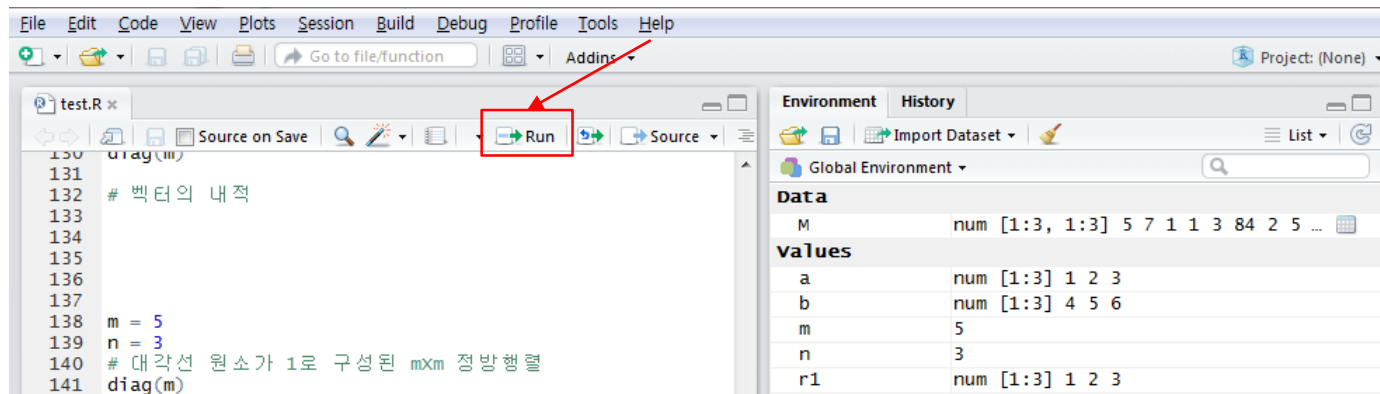
```
# 함수 정의  
myfunction = function(x){  
  y = sin(x)/x  
  plot(x, y, type='l')  
  return(y)  
}  
  
# 함수 실행  
x = seq(-10,10,0.01)  
myfunction(x)
```



5. R 프로그래밍

□ R-file

- ① R 명령어들을 포함하는 텍스트 파일
- ② 명령어가 너무 길거나 반복적으로 써야할 때 사용
- ③ 파일의 확장자는 반드시 ".R"이어야 함
- ④ Run 을 눌러 실행할 수 있음



이름	수정한 날짜	유형	크기
test.R	2017-01-15 오후...	R 파일	2KB

5. R 프로그래밍

□ R-file (R 콘솔창에서 실행하기)

- ① R-file이 저장된 디렉토리로 이동후에 다음과 같이 실행가능

이름	수정한 날짜	유형	크기
 test.R	2017-01-15 오후...	R 파일	2KB

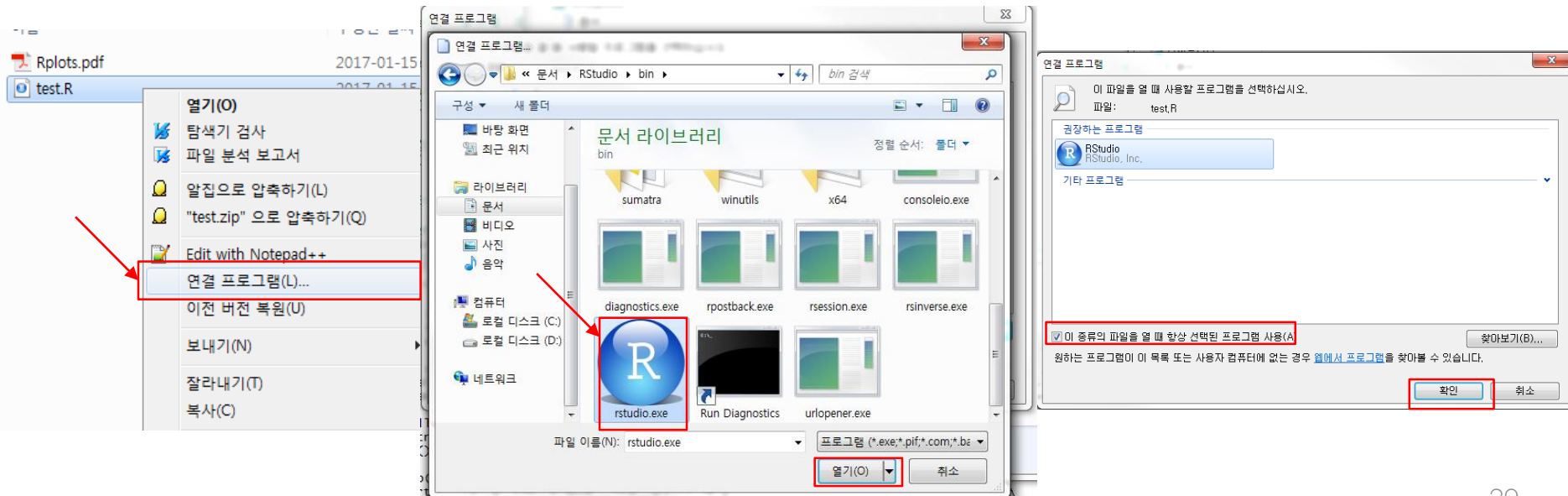
```
> source("test.R")
```

5. R 프로그래밍

□ R-file (R 콘솔창에서 실행하기)

#참고: R-file을 RStudio로 열기

- ① R-file 우측클릭 -> 연결프로그램 클릭
- ② 찾아보기 클릭 : 문서->RStudio->bin->rstudio.exe 열기
- ③ "이 종류의 파일을 열 때 항상 선택된 프로그램 사용" 체크, 확인



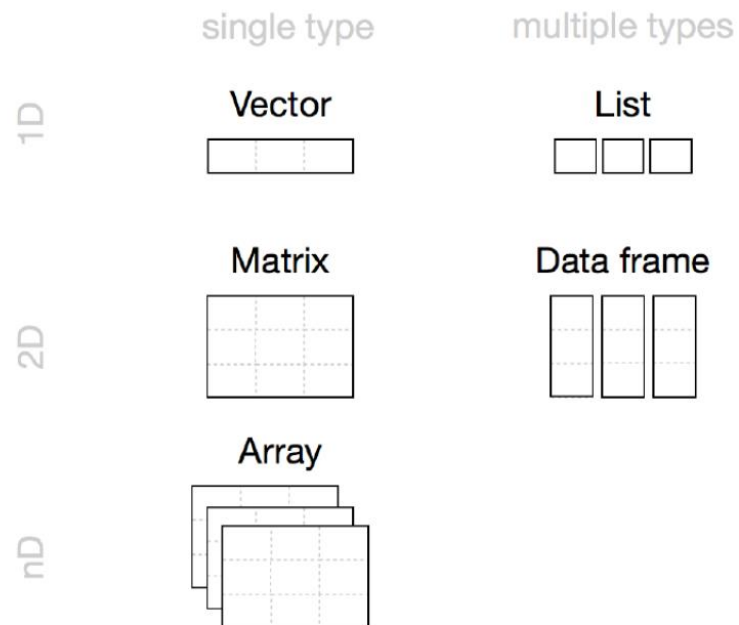
5. R 프로그래밍

□ R 기초 문법

1. 변수의 선언 : (Declaration)

숫자, 벡터, 행렬의 생성 => 이미 했음.

다양한 데이터 타입 : 리스트, Data Frame



5. R 프로그래밍

□ R 기초 문법

2. 조건문 : (Selection)

조건에 따라 어떤 코드를 실행할 것인가?

실행 조건 ← if([조건1]){
[명령문] → 실행할 명령문
} else if([조건2]){
[명령문]
} else{
[명령문]
}

의! ←
쓰고 "}"와 If 문 종료

주의!
else 쓸 꼭 "}"와
같은 줄에 쓰기

```
# Example
a = 1

if (a<0){
    b = -1
} else if (a>1){
    b = 1
} else{
    b = 0
}

# 실행하면 b=0
```

5. R 프로그래밍

□ R 기초 문법

3. 반복문 : (Iteration)

이 코드를 언제까지 반복할 것인가?

반복 조건

```
# for 문 : 반복적인 작업 수행
for (변수 in 범위){
  [명령문] → 실행할 명령문
  if( [탈출조건] ){
    break
  }
}
```

탈출 조건

```
# Example
j = 0
for (i in 0 : 100){
  j = j + i
  if(i>4){
    break
  }
}

# 실행하면 j=15 이 실행 됨
```


5. R 프로그래밍

□ R 기초 문법

3. 반복문 : (Iteration)

이 코드를 언제까지 반복할 것인가?

```
# while 문 : 반복회수가 정해지지 않을 때 사용
while([반복조건]){
  [명령문] → 실행할 명령문
}
```

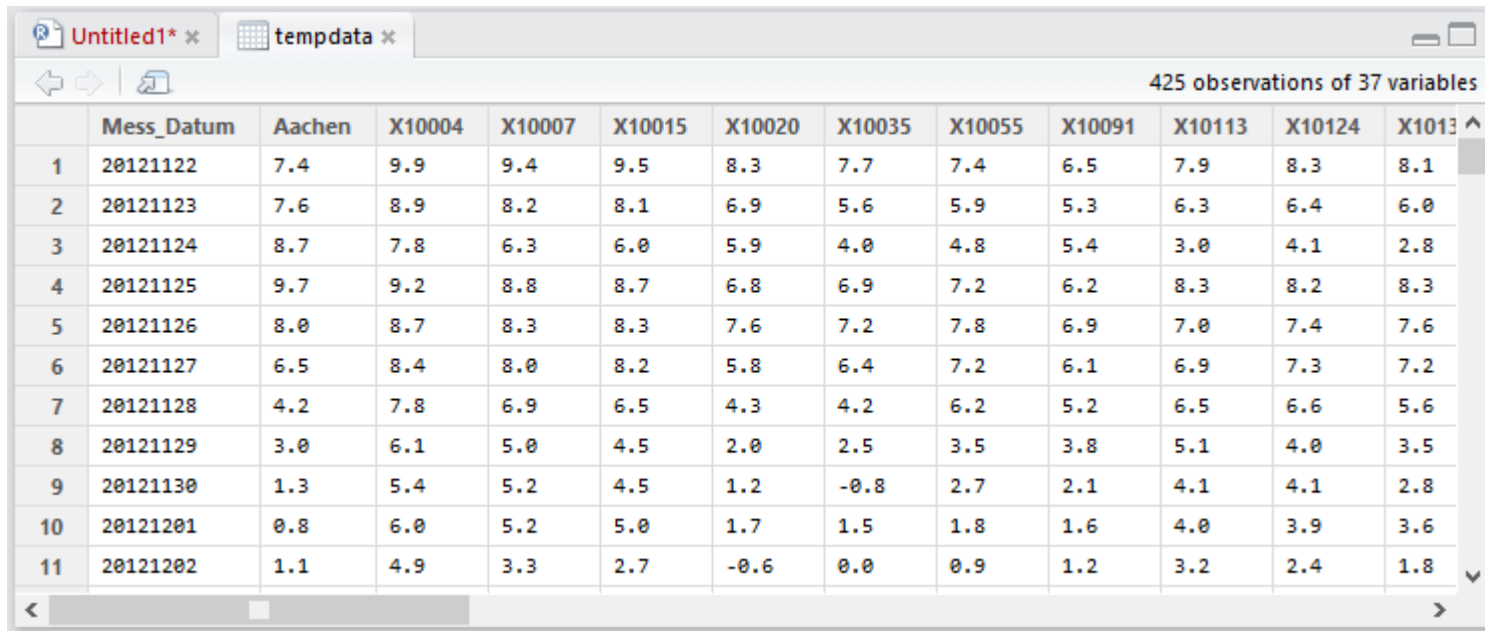
반복 조건 : 이 조건이 맞지 않으면 탈출함

```
# Example (for 문과 정확히 동등한 예)
j = 0
i = 0
while(i<=5){
  j = j + i
  i = i + 1
}
# 실행하면 j=15 이 실행됨
```

5. R 프로그래밍

□ R 데이터 분석 : Data Frame

- ① 대부분의 데이터는 엑셀과 같은 표로 이루어져 있음
- ② R에서 표 형식의 데이터가 Data Frame (2차원)
- ③ 변수 이름에 따라 수많은 데이터 존재



425 observations of 37 variables

	Mess_Datum	Aachen	X10004	X10007	X10015	X10020	X10035	X10055	X10091	X10113	X10124	X1013
1	20121122	7.4	9.9	9.4	9.5	8.3	7.7	7.4	6.5	7.9	8.3	8.1
2	20121123	7.6	8.9	8.2	8.1	6.9	5.6	5.9	5.3	6.3	6.4	6.0
3	20121124	8.7	7.8	6.3	6.0	5.9	4.0	4.8	5.4	3.0	4.1	2.8
4	20121125	9.7	9.2	8.8	8.7	6.8	6.9	7.2	6.2	8.3	8.2	8.3
5	20121126	8.0	8.7	8.3	8.3	7.6	7.2	7.8	6.9	7.0	7.4	7.6
6	20121127	6.5	8.4	8.0	8.2	5.8	6.4	7.2	6.1	6.9	7.3	7.2
7	20121128	4.2	7.8	6.9	6.5	4.3	4.2	6.2	5.2	6.5	6.6	5.6
8	20121129	3.0	6.1	5.0	4.5	2.0	2.5	3.5	3.8	5.1	4.0	3.5
9	20121130	1.3	5.4	5.2	4.5	1.2	-0.8	2.7	2.1	4.1	4.1	2.8
10	20121201	0.8	6.0	5.2	5.0	1.7	1.5	1.8	1.6	4.0	3.9	3.6
11	20121202	1.1	4.9	3.3	2.7	-0.6	0.0	0.9	1.2	3.2	2.4	1.8

5. R 프로그래밍

□ R 데이터 분석 : Data Frame

	LungCap	Age	Height	Smoke	Gender	Caesarean
1	6.475	6	62.1	no	male	no
2	10.125	18	74.7	yes	female	no
3	9.550	16	69.7	no	female	yes
4	11.125	14	71.0	no	male	no
5	4.800	5	56.9	no	male	no
6	6.225	11	58.7	no	female	no
7	4.950	8	63.3	no	male	yes
8	7.325	11	70.4	no	male	no
9	8.875	15	70.5	no	male	no
10	6.800	11	59.2	no	male	no
11	11.500	19	76.4	no	male	yes
12	10.925	17	71.7	no	male	no
13	6.525	12	57.5	no	male	no
14	6.000	10	61.1	no	female	no
15	7.825	10	61.2	no	male	no
16	9.525	13	63.5	no	male	yes
17	7.875	15	59.2	no	male	no
18	5.050	8	56.1	no	male	no
19	7.025	11	61.2	yes	female	no

```
> df = read.csv(file.choose())
```

```
> names(df)
[1] "LungCap" "Age"      "Height"   "Smoke"    "Gender"
```

```
> dim(df)
[1] 725 6
```

```
> summary(df)
      LungCap      Age      Height      Smoke
Min.   : 0.507   Min.   : 3.00   Min.   :45.30   no :648
1st Qu.: 6.150   1st Qu.: 9.00   1st Qu.:59.90   yes: 77
Median : 8.000   Median :13.00   Median :65.40
Mean   : 7.863   Mean   :12.33   Mean   :64.84
3rd Qu.: 9.800   3rd Qu.:15.00   3rd Qu.:70.30
Max.   :14.675   Max.   :19.00   Max.   :81.80

      Gender      Caesarean
female:358   no :561
male :367    yes:164
```

```
> nrow(df)
[1] 725
```

```
> ncol(df)
[1] 6
```

5. R 프로그래밍

□ R 데이터 분석 : Data Frame

	LungCap	Age	Height	Smoke	Gender	Caesarean
1	6.475	6	62.1	no	male	no
2	10.125	18	74.7	yes	female	no
3	9.550	16	69.7	no	female	yes
4	11.125	14	71.0	no	male	no
5	4.800	5	56.9	no	male	no
6	6.225	11	58.7	no	female	no
7	4.950	8	63.3	no	male	yes
8	7.325	11	70.4	no	male	no
9	8.875	15	70.5	no	male	no
10	6.800	11	59.2	no	male	no
11	11.500	19	76.4	no	male	yes
12	10.925	17	71.7	no	male	no
13	6.525	12	57.5	no	male	no
14	6.000	10	61.1	no	female	no
15	7.825	10	61.2	no	male	no
16	9.525	13	63.5	no	male	yes
17	7.875	15	59.2	no	male	no
18	5.050	8	56.1	no	male	no
19	7.025	11	61.2	yes	female	no

```

> class(df)
[1] "data.frame"
>
> df$Age
[1] 6 18 16 14 5 11 8 11 15 11 19 17 12 10 10 13 15 8 11 14
[21] 6 8 16 11 11 12 12 9 4 18 4 13 13 13 12 10 6 9 11 17
[41] 14 17 8 12 6 11 11 12 17 7 15 15 11 10 18 6 13 19 9 12
[61] 12 14 8 12 12 12 11 11 11 12 14 11 11 13 13 12 14 8 17 11

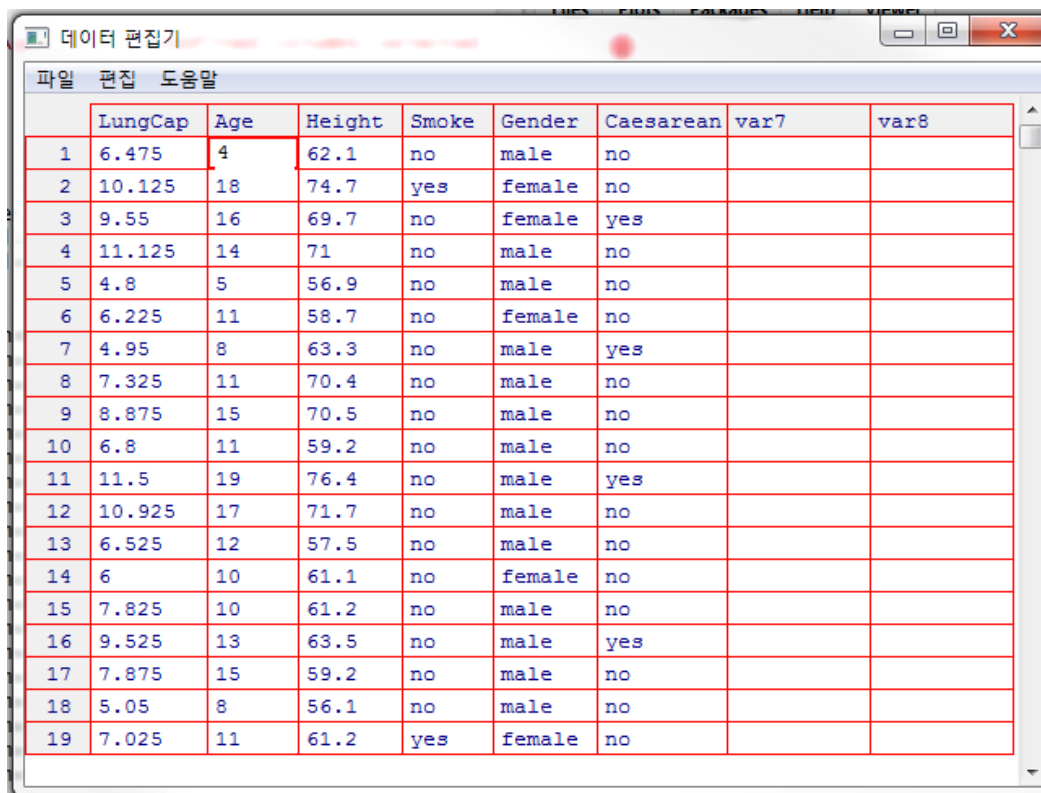
> df[1,3]
[1] 62.1
>
> df[5:6,]
  LungCap Age Height Smoke Gender Caesarean
5   4.800  5   56.9   no   male         no
6   6.225 11   58.7   no  female         no
>
> df[df$Age < 5,]
  LungCap Age Height Smoke Gender Caesarean
29   1.125  4   48.7   no  female         no
31   4.650  4   53.7   no  female         no
114  5.875  3   55.9   no   male         no
150  0.507  3   51.6   no  female         yes
190  2.875  4   55.4   no   male         no
222  1.175  3   51.9   no   male         no
229  4.700  3   52.7   no   male         no
293  5.475  3   52.9   no   male         no
318  1.025  3   47.0   no  female         no
390  3.400  4   55.6   no   male         no
401  2.000  3   51.0   no  female         no
405  3.225  4   52.8   no  female         no
487  2.375  4   51.7   no  female         yes
514  1.675  3   51.9   no   male         no
531  4.075  3   53.6   no   male         yes
596  1.450  3   45.3   no  female         no
621  3.675  3   54.2   no   male         yes
660  3.250  3   52.0   no   male         no
713  3.425  3   51.0   no   male         yes

```

5. R 프로그래밍

□ R 데이터 분석 : Data Frame

```
> df_edit = edit(df)
```



	LungCap	Age	Height	Smoke	Gender	Caesarean	var7	var8
1	6.475	4	62.1	no	male	no		
2	10.125	18	74.7	yes	female	no		
3	9.55	16	69.7	no	female	yes		
4	11.125	14	71	no	male	no		
5	4.8	5	56.9	no	male	no		
6	6.225	11	58.7	no	female	no		
7	4.95	8	63.3	no	male	yes		
8	7.325	11	70.4	no	male	no		
9	8.875	15	70.5	no	male	no		
10	6.8	11	59.2	no	male	no		
11	11.5	19	76.4	no	male	yes		
12	10.925	17	71.7	no	male	no		
13	6.525	12	57.5	no	male	no		
14	6	10	61.1	no	female	no		
15	7.825	10	61.2	no	male	no		
16	9.525	13	63.5	no	male	yes		
17	7.875	15	59.2	no	male	no		
18	5.05	8	56.1	no	male	no		
19	7.025	11	61.2	yes	female	no		

5. R 프로그래밍

□ R 데이터 분석 : `apply(m, dimcode, f, fargs)`

`m` : matrix

`dimcode` : 1 = x축, 2 = y축

`f` : 적용할 함수

`fargs` : `f`에 추가로 들어가는 arguments

5. R 프로그래밍

□ R 패키지 사용하기 (ggplot2)

- ① 전세계의 유능한 사람들이 다양한 패키지를 무료로 제공
- ② 필요한 패키지 찾아서 사용하면 됨
- ③ ggplot2 : R의 대표적인 그래프 그리기 패키지

#패키지 관련 명령어

현재 설치된 패키지 목록 확인

```
installed.packages()
```

패키지 설치

```
install.packages([패키지 이름])
```

ggplot2 패키지 설치

```
install.packages("ggplot2")
```

CRAN mirror는 Korea로 선택

ggplot2 패키지 사용하기

```
library(ggplot2)
```

5. R 프로그래밍

□ R 패키지 사용하기 (ggplot2)

2차원 그래프

```
# x값 설정  
x = seq(-50,50,0.01)
```

```
# x에 따른 y값 계산  
y = sin(x)/x
```

```
# ggplot2로 그래프 그리기  
qplot(x,y,
```

x값

main='sin(x)/x',

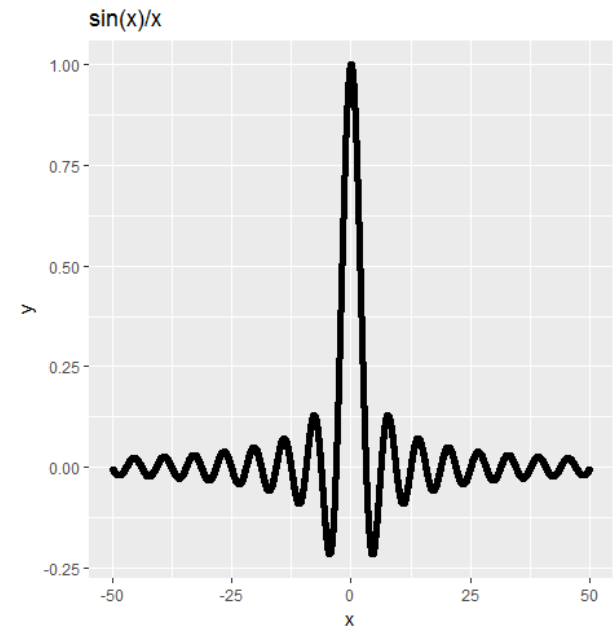
xlab='x',

ylab='y')

그래프 제목

x축 이름

y축 이름



5. R 프로그래밍

□ R 패키지 사용하기 (plotly)

3차원 곡면 그래프

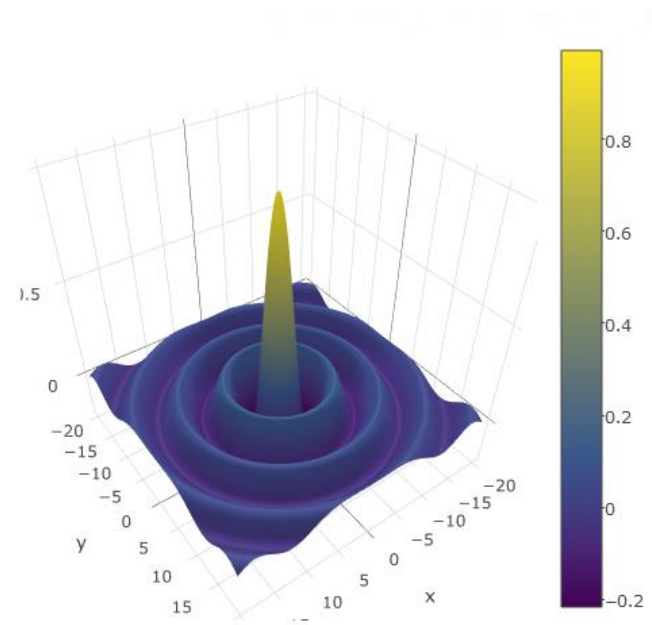
```
# plotly 패키지 설치
install.packages(plotly)
# plotly 불러오기
library(plotly)

# x값 설정
x = seq(-20,20,0.2)

# y값 설정
y = seq(-20,20,0.2)

# z값 계산
z = outer(x, y,
          function(x, y) sin(sqrt(x^2 + y^2))/sqrt(x^2 + y^2))

# plotly surface 그리기
plot_ly(x=x,y=y,z=z) %>% add_surface()
```



왼쪽클릭 드래그 : 돌려보기

우측클릭 드래그 : 시점 이동

스크롤 클릭 드래그: 확대 및 축소

6. R 응용 : 회귀 분석

□ 회귀분석

- ✓ 회귀분석이란 통계학에서 연속된 변수간의 관계를 찾는 것
- ✓ Y가 종속변수이고 X가 설명변수일 때 회귀계수는 다음 식으로 구할 수 있음

$$b = (X'X)^{-1}X'Y$$

- ✓ 가상의 데이터를 다음과 같이 생성한다.

$$X = 1:100$$

$$Y = 3 + 2 * X + 4 * \varepsilon$$

(여기서 ε 는 표준정규분포 확률변수이다.)

예제3 위와 같은 조건에서 가상의 데이터를 생성하고 회귀분석 결과를 그래프로 그려라. (rnorm(), lm(y~x)를 사용할 것)

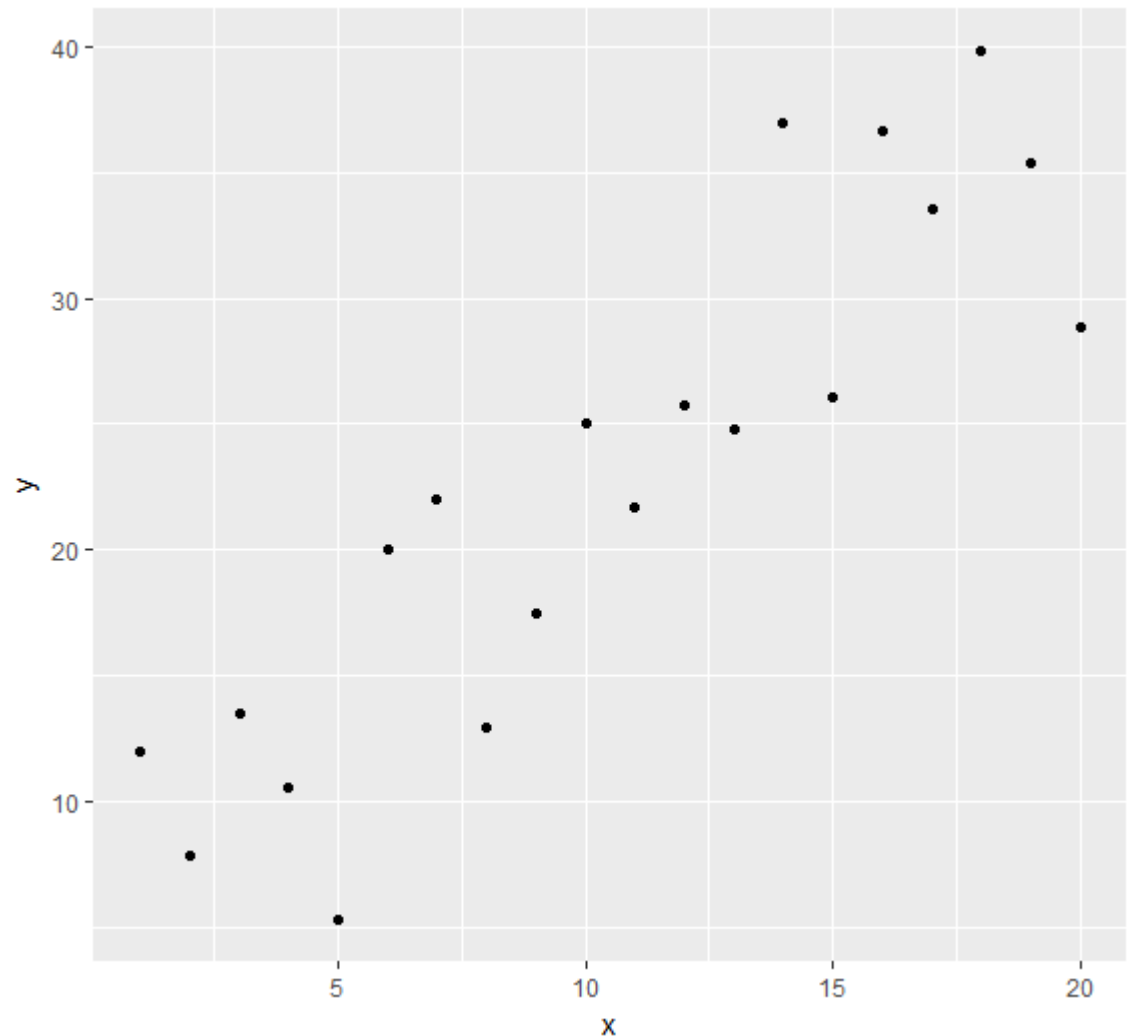
6. R 응용 : 회귀 분석

□ 예제3 회귀 분석

```
# 가상의 데이터 생성하기
x=1:20
y=3+2*x + 4*norm(20)

data = data.frame(x=x, y=y)

# ggplot을 이용하여 그래프 그리기
library(ggplot2)
qplot(x=x, y=y, data=data)
```

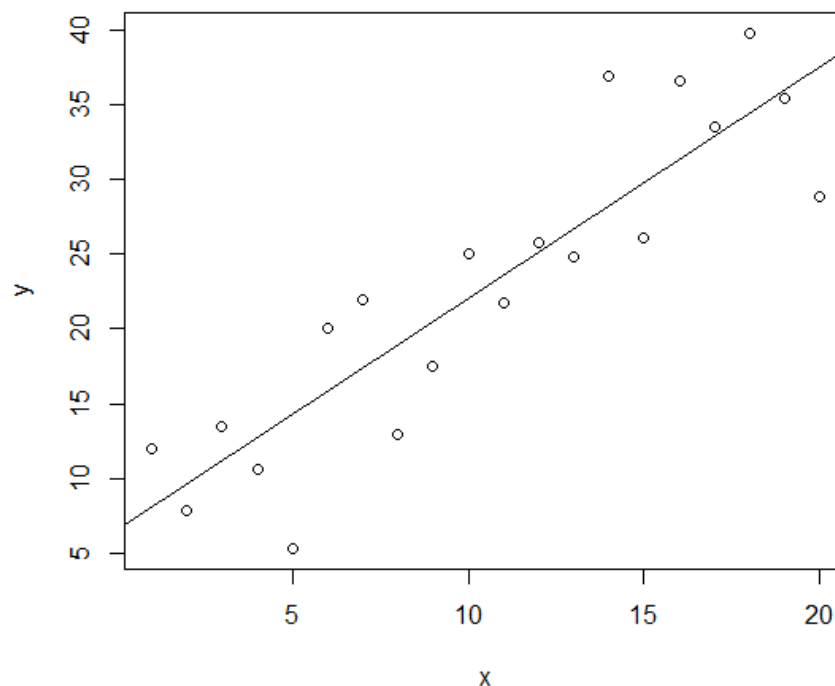


6. R 응용 : 회귀 분석

□ 예제3 회귀 분석

```
# lm을 이용한 회귀분석
fit = lm(y~x, data=data)
summary(fit)

# 그래프 그리기
plot(data)
abline(fit)
```



Residuals:

Min	1Q	Median	3Q	Max
-8.9415	-2.3544	0.0051	3.9887	8.6953

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	6.4907	2.2747	2.853	0.0106 *
x	1.5548	0.1899	8.188	1.76e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.897 on 18 degrees of freedom

Multiple R-squared: 0.7883, Adjusted R-squared: 0.7766

F-statistic: 67.04 on 1 and 18 DF, p-value: 1.758e-07

6. R 응용 : 회귀 분석

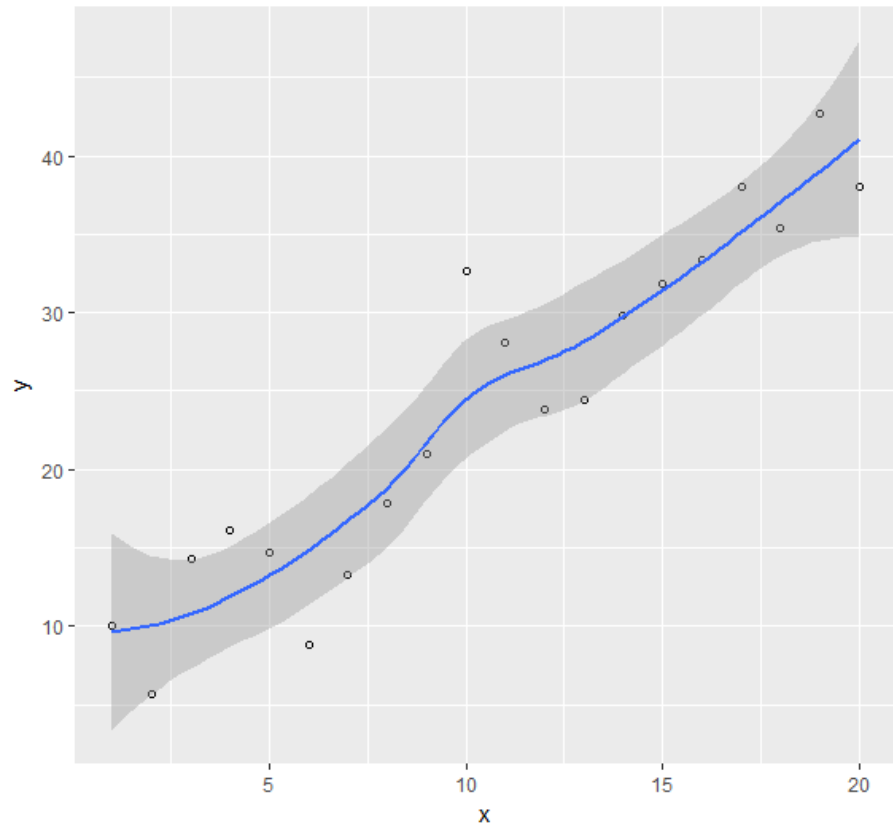
□ 예제3 회귀 분석

```
# 가상의 데이터 생성하기
x=1:20
y=3+2*x + 4*rnorm(20)

data = data.frame(x=x, y=y)

# ggplot을 이용하여 그래프 그리기
library(ggplot)

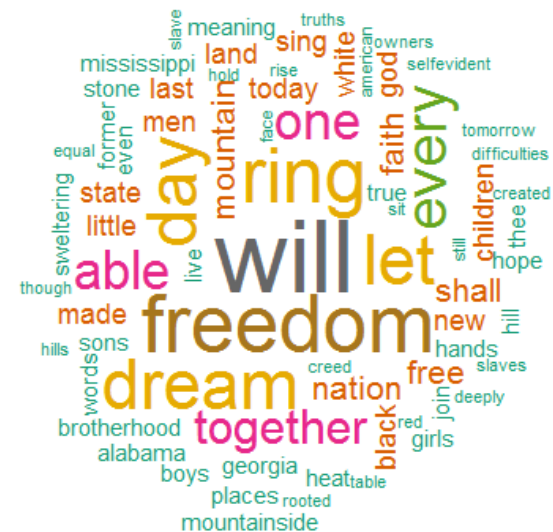
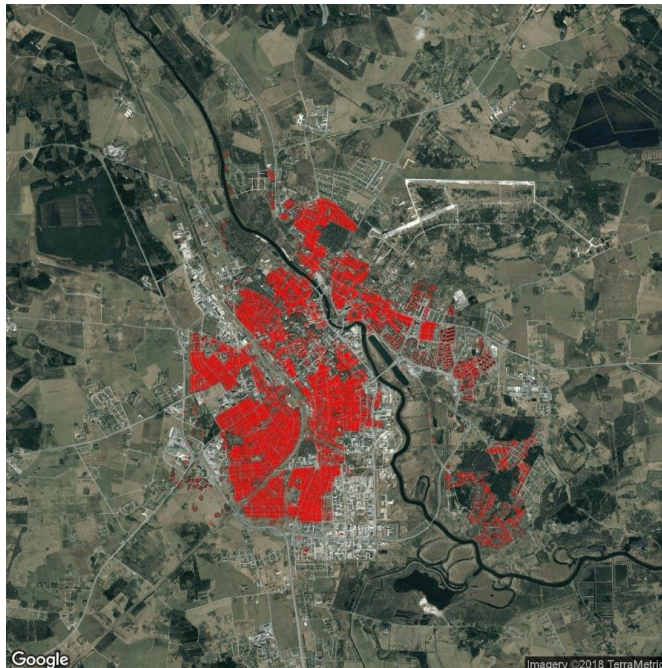
ggplot(data, aes(x=x, y=y))+
  geom_point(shape=1)+
  geom_smooth(method=lm)
```



5. R 프로그래밍

□ 예제4 R 패키지 추가 응용

- ① ggmap : 인공위성 지도 이미지
- ② ggplot2 : 그래프 그리기
- ③ wordcloud : word cloud 만들기

$$\frac{\square}{\bigcirc} \frac{\square}{\bigcirc} \dots$$


6. R 응용 : 회귀 분석

□ 예제5 LungCapData : Multiple Regression & Visualization

데이터 불러오기

```
LungCapData = read.csv(file.choose(), header=T, sep='\t')
```

데이터 표시하기

```
library(plotly)
```

```
plot_ly(LungCapData, x=~Age, y=~Height, z=~LungCap,  
        type='scatter3d', marker=list(size=2, color='black'))
```

fitting surface 만들기

```
x_axis = seq(min(LungCapData$Age), max(LungCapData$Age), 1)
```

```
y_axis = seq(min(LungCapData$Height), max(LungCapData$Height), 1)
```

```
fit = lm(LungCap~Age+Height, LungCapData)
```

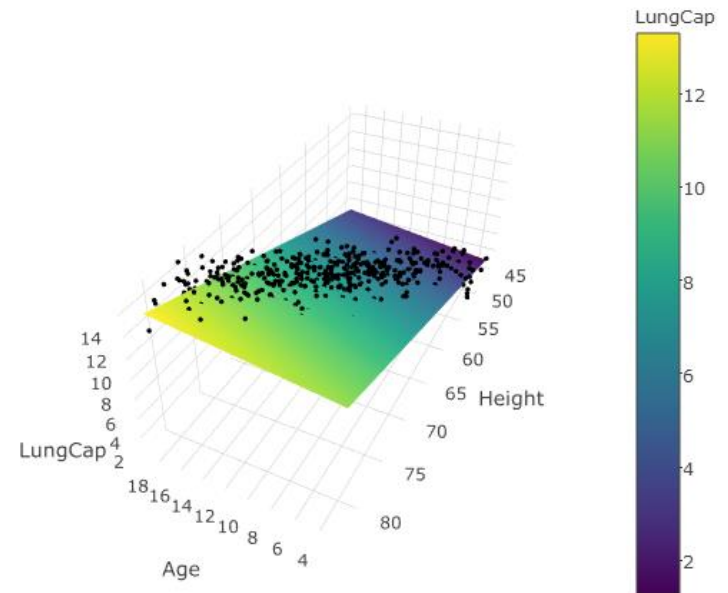
```
fit_surface = expand.grid(Age=x_axis, Height=y_axis)
```

```
fit_z = matrix(predict.lm(fit, fit_surface), nrow=length(x_axis))
```

그래프 그리기

```
plot_ly(LungCapData, x=~Age, y=~Height, z=~LungCap, type='scatter3d',  
        marker=list(size=2, color='black')) %>%
```

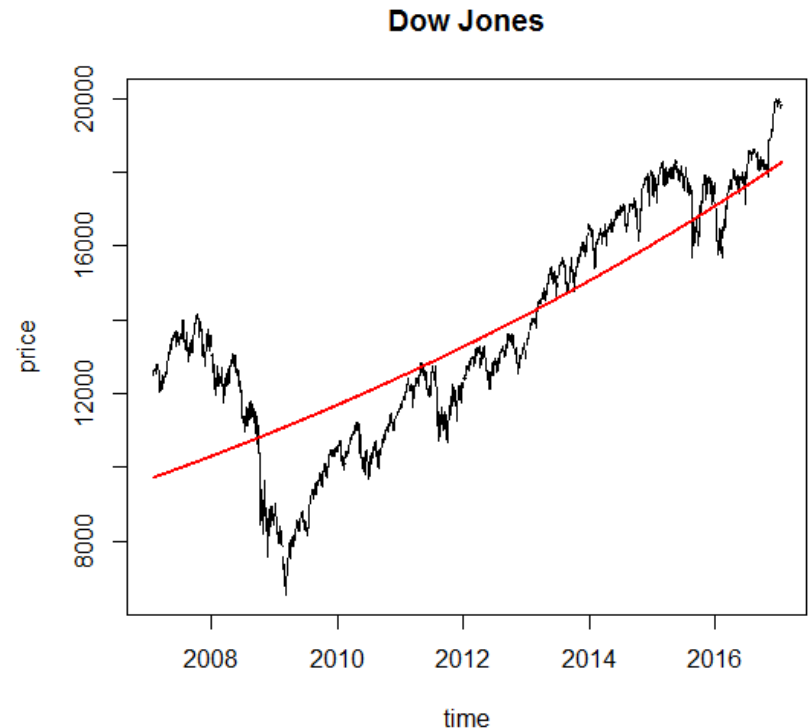
```
add_trace(z=t(fit_z), x=x_axis, y=y_axis, type='surface')
```



6. R 응용 : 금융 데이터 분석

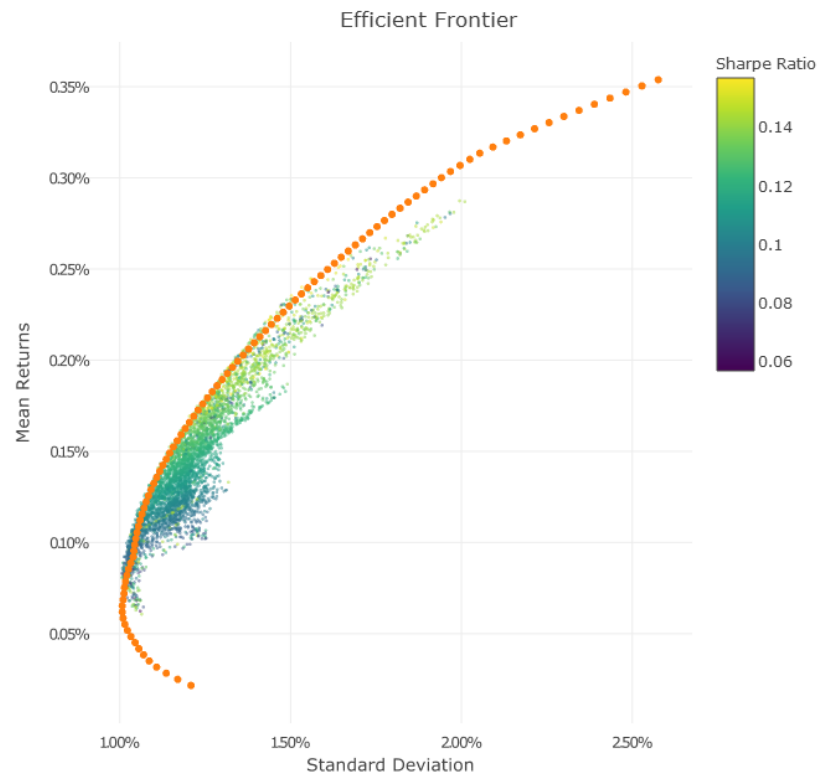
- **예제6** 다우 존스 추세 분석
- quantmod package 설치
- 다우 존스 추세 분석

```
# 다우 존스 추세 분석  
getSymbols("DJIA", src="FRED")  
price = as.numeric(DJIA)  
time = index(DJIA)  
  
x = 1:length(price)  
fit = lm(log(price)~x)  
fit_exp = exp(fit$coef[1] + fit$coef[2]*x)  
  
plot(x=time, y=price, main="Dow Jones", type="l")  
lines(time, fit_exp, col=2, lwd=2)
```



6. R 응용 : 금융 데이터 분석

□ 예제7 포트폴리오 최적화 (Mean Variance Optimization)



End

Question & Comment