

# Basic Statistics with R

---

## 01 R이란 무엇인가

---

## R 소개

- R 프로그래밍 언어(줄여서 R)는 통계 계산과 그래픽을 위한 프로그래밍 언어이자 소프트웨어 환경이다.  
뉴질랜드 오클랜드 대학의 로스 이하카와 로버트 젠틀만에 의해 시작되어 현재는 R 코어 팀이 개발하고 있다.
- R의 문법과 통계처리 부분은 AT&T 벨 연구소가 개발했던 S를 참고했고, 데이터 처리부분은 스킴에 영향을 받았다.
- R은 다양한 통계 기법과 수치 해석 기법을 지원한다.  
R은 사용자가 제작한 패키지를 추가하여 기능을 확장할 수 있다.  
핵심적인 패키지는 R과 함께 설치되며, CRAN(the Comprehensive R Archive Network)을 통해 2020년 현재 16,000개 이상의 패키지를 내려 받을 수 있다.
- R의 또다른 강점은 그래픽 기능으로 수학 기호를 포함할 수 있는 출판물 수준의 그래프를 제공한다.
- R은 통계 계산과 소프트웨어 개발을 위한 환경이 필요한 통계학자와 연구자들 뿐만 아니라, 행렬 계산을 위한 도구로서도 사용될 수 있으며 이 부분에서 GNU Octave나 MATLAB에 견줄 만한 결과를 보여준다.
- R은 윈도, 맥 OS 및 리눅스를 포함한 UNIX 플랫폼에서 이용 가능하다.

## Why R?

- R의 장점

- 무료

- 자유로운 데이터 분석이 가능

- > Graphical User Interface (GUI)를 이용한 분석 환경은 처음에는 접근이 쉬우나 확장성에 한계가 있음.

- > 스스로 새로운 기능을 추가하는 것도 자유로움

- 그래픽이 예쁘다

- R의 단점

- 커맨드라인(command line) 기반 환경

- > 명령어들 다수가 정해진 문법을 따라야 함

- > 결과가 잘못되었을 때 따라오는 피드백이나 에러메시지가 그다지 친절하지 않음

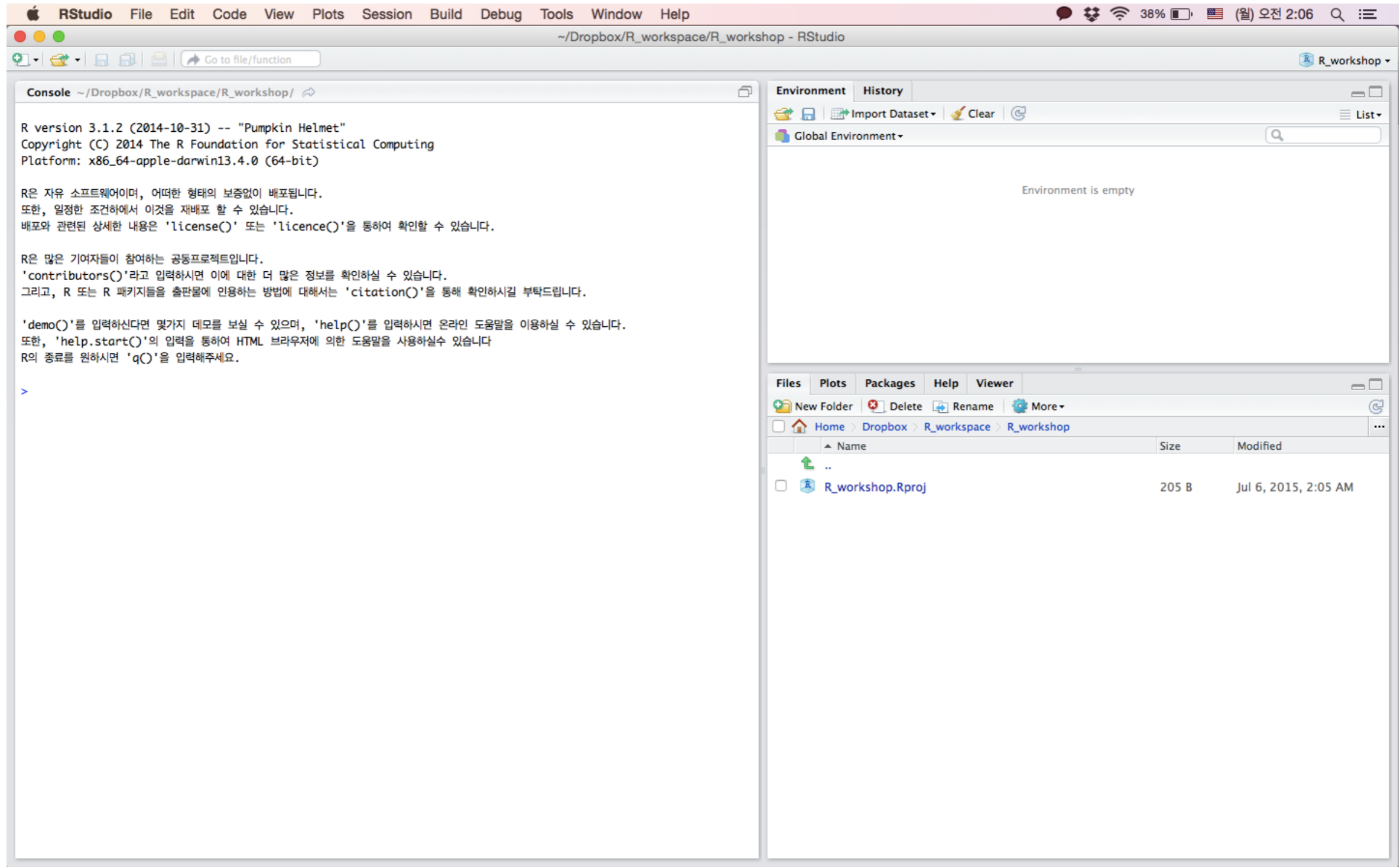
- 한글 표기가 잘 작동하지 않음

---

## 02 R 시작하기

---

# R Studio



# R Studio

## Command와 결과가 보여짐

## 변수(object), 데이터, 히스토리 열람

```
Console ~/Dropbox/R_workspace/Newspaper_2017/
Natural language support but running in an English locale

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 ~/Dropbox/R_workspace/Newspaper_2017/.RData]
> |
```

Environment	History	Connections
Global Environment		
Data		
both	791 obs. of 589 variables	
by_cyl	32 obs. of 11 variables	
cor.data	1099 obs. of 17 variables	
cpi1	790 obs. of 6 variables	
cpi1.g	46 obs. of 3 variables	
cpi2	720 obs. of 6 variables	
cpi2.g	48 obs. of 3 variables	
data	1099 obs. of 275 variables	
data2	1201 obs. of 322 variables	
demo	791 obs. of 10 variables	

```
Basic_result.Rmd x Basic_result4.Rmd x Untitled3* x Untitled4* x Untitled5 x total x
1 |
```

Files Plots Packages Help Viewer

Zoom Export

## Script 명령어 편집기

## Files, Plots, Packages, Help

## R Studio 설명

- **Console 창**

- 모든 명령어와 결과가 나타남.
- 1회성으로 명령어가 흘러감. 명령어 편집 등을 할 수 없으나, 간단한 명령어 실행해 볼 때는 사용해도 됨

- **Source창**

- 명령어 작성과 편집은 여기서
- 명령어 입력 후, 선택하여 'Run'하거나 "Ctrl(mac의 경우 Command) + Enter" 해야 실행
- 명령어 자동 완성 기능 사용 - 명령어 치고 "tab 키"를 누르면 파일명이나 변수명을 자동으로 찾아 줌 - 문자, 숫자, 연산자, 명령어가 색깔이 다르게 표기되므로 구분이 쉬움

- **History**

- 이제까지 console에서 실행한 명령어들을 볼 수 있음
- 과거 실행 명령어를 선택하여 console로 보내거나, source창으로 보낼 수 있음

- **Environment:**

- 현재 작업 중에 생성 한 변수, 데이터 등을 볼 수 있음
- object의 간단한 구조도 보여줌

- **Packages:** 현재 설치한 package들을 보여줌

- **Plots:** 그래프를 그릴 경우 표시되는 창

- **Files:** Working Directory에 있는 파일들 보여줌



## R 그리고 데이터 분석을 시작하는 마음가짐

- 사람은 실수한다. 그러나 컴퓨터는 실수하지 않는다.
- 데이터 정리에 시간을 아끼지 말라.
- 일정한 규칙에 따라 정리 - 규칙은 간단하게
- 내가 만들지 않은 데이터는 절대 그대로 쓰지 않는다.
- 데이터를 정리할 때는 컴퓨터처럼 생각한다.
- Detail, detail, detail
- 사소한 것도 기록하고, 저장한다.
- 에러를 두려워하지 말라.

## 시작하기 전에...

- R의 기본 명령어 구조는 “명령어( )” 형태
  - ( )속에는 명령어가 적용될 변수, 데이터 등을 넣는다
- object 이름은 숫자로 시작해서는 안된다
- object 이름에는 공백이 있어서는 안된다 → “.”나 “\_”를 사용
- 대문자, 소문자를 구분한다
- 띄어쓰기는 하는 것과 하지 않는 것이 동일하다
  - coding etiquette  
(<https://ourcodingclub.github.io/2017/04/25/etiquette.html>)
- R studio에서 명령어와 파일 쉽게 입력하기 위해서는 작성 중에 tab을 눌러보자
  - 명령어의 옵션, Working Directory에 있는 파일, 사용 가능한 object들을 보여줌

## Basic Operators

연산자	뜻
+, -, *, /	더하기, 빼기, 곱하기, 나누기
log(), log10()	로그
exp()	exponential
sqrt()	제곱근( $\sqrt{\quad}$ )
==, !=	같다, 같지 않다
>, <, >=, <=	크다, 작다, 크거나 같다, 작거나 같다
round()	반올림
ceiling()	올림
floor()	버림
%%	몫
%%	나머지
:	수열
&	and
	or

## 기본 함수들

명령어	기능	예제
ls()	생성된 object 리스트를 보여줌	
rm()	object 삭제	rm(x), rm(list = ls())
gc()	garbage collection 호출하여 메모리 효율성을 올림	
c( , )	2개 이상의 원소 결합	c(x, y), c(1:3, 5), c("Dad", "Mom", "Dog")
rep( , )	원소를 반복	rep(5, 5), rep(0, 10), rep(1:3, 5)
seq( , , )	규칙이 있는 수열 생성	seq(1, 10, 2), seq(-10, 10, 3)
sort()	정렬	sort(x), sort(x, decreasing = TRUE)
order()	첨자정렬	order(x)
sample( , )	데이터 섞기, 랜덤추출	sample(1:10, 5), sample(1:10, 5, replace = TRUE)
paste( , , sep = )	문자열 연결	paste("Hi", "everyone", sep = " "), paste("A", 1:10, sep="")
gsub	문자열 대치, 변경	gsub("a", "z", "abc, cba, xya"), gsub(pattern, replacement, x)

## 패키지(Package)

- R에서 사용하는 다양한 명령어, 기능들은 패키지의 형태로 제공된다
- 이용자들의 필요에 의해 개발된 특수한 목적들의 로직과 코드의 집합을 패키지라고 부른다
- 기본 함수 외에 많은 기능은 패키지를 설치해야 사용할 수 있다
- 패키지 설치: `install.packages("package")`
- 패키지를 이용하기 위해서는 `library` 또는 `require` 함수를 통해 R 세션에 로드해 두어야 한다.

`library("package")`

`require("package")`

- `library`나 `require`은 R 세션을 새로 시작할 때마다 해주어야 함

- 시작 시, 1회 하면 계속 사용할 수 있음

---

## 03 데이터 보기

---

## Working directory 설정

- getwd함수로 작업 디렉터리 확인, setwd 함수로 변경

```
> getwd()
[1] "/Users/hjhwang/Dropbox/R_workspace/R_workshop"
> setwd("/Users/hjhwang/Dropbox/R_workspace/practice")
> getwd()
[1] "/Users/hjhwang/Dropbox/R_workspace/practice"
```

- 내가 지금 어디에 있는지 확인하기
- Working Directory에 데이터 파일들을 저장하면 이용하기 쉽다
- 결과를 저장할때, 별도로 지정하지 않으면 Working Directory에 저장된다.
- R studio에서 변경 및 확인
  - file pane에서 확인하고 변경하기
- R project 생성하기
  - 데이터 분석을 시작할 때, 작업 공간을 세팅하고 시작하기
  - 새로운 분석을 만들 때마다, 프로젝트를 새로 생성해서 관리하면 좋다.
  - 프로젝트 안에 데이터 파일, 코드, 결과를 함께 관리

## R의 데이터 구조

- R object에는 벡터(Vectors), 행렬(Matrices), 리스트(Lists), 데이터프레임(Dataframes), 배열(Arrays)가 있다.
- 동질적 객체(Homogeneous Objects): 벡터, 행렬, 배열
- 이질적 객체(Heterogeneous Objects): 리스트, 데이터 프레임
- 벡터와 리스트는 1차원 객체, 행렬과 데이터프레임은 2차원 객체, 배열은 필요한 만큼의 차원을 가질 수 있다.

## 벡터

- 같은 형의 원소로 구성된 objects

```
> v <- 1:10
> v
[1] 1 2 3 4 5 6 7 8 9 10
> v <- c(1, 3, 5, 7, 9)
> v
[1] 1 3 5 7 9
> v <- c("Fred", "Mary", "David")
> v
[1] "Fred" "Mary" "David"
>
```

- [ ]를 통해서 indexing
- c( ), rep( ), seq( ) 등으로 생성



## 행렬

- matrix 함수를 사용하여 생성: `matrix(values, ncol, nrow)`
- dim 함수로 벡터를 행렬로 변경 가능
- 행렬 내 원소에 접근하려면 행과 열 번호를 인덱스로 이용 `[row, col]`
- 벡터와 같은 방식으로 인덱싱 가능

## 리스트

- 이질적인 원소들로 구성되는 object
- `list()` 로 생성

```
> lst <- list(3.14, "Moe", c(1, 1, 2, 3))
```

```
> lst
```

```
[[1]]
```

```
[1] 3.14
```

```
[[2]]
```

```
[1] "Moe"
```

```
[[3]]
```

```
[1] 1 1 2 3
```

## 리스트(Cont.)

- [], [[ ]] 로 인덱싱 (이거 좀 헷갈림!!)

```
> lst <- list(name = "Fred", wife = "Mary", c(1, 3, 5, 10))
> lst
$name
[1] "Fred"

$wife
[1] "Mary"

[[3]]
[1] 1 3 5 10

> lst[[1]]
[1] "Fred"
> lst$name
[1] "Fred"
> lst[[3]]
[1] 1 3 5 10
> lst[1]
$name
[1] "Fred"
```

## 데이터 프레임

- 행과 열로 구성된 행렬 형식의 object
  - list의 한 종류.
  - 모든 열의 길이가 같은 리스트의 조합
  - 행은 case, 열은 variable 또는 attribute
  - data frame의 예

Name	Age	Gender	Weight
Dad	43	Male	76
Mom	42	Female	56
Sister	12	Female	42
Brother	8	Male	25
Dog	5	Female	5

- 표의 첫 행은 메타데이터. 데이터에 대한 데이터
- Data frame 속성
  - 행(row): case
  - 열쇠속성(key attribute): "Name" 각 행을 유일하게 특정하는 사례이름
  - 열(column): variable
    - > 한 열은 위아래로 모두 동일한 종류의 값을 가지고 있다  
(Name: character, Age: numeric, Gender: factor, Weight: numeric)
  - 모든 열은 같은 수의 엔트리를 가지고 있다 -> 전체 데이터는 직사각형 형태

## 데이터 프레임

- Data frame 내 indexing
  - 데이터프레임명\$리스트 key 형태로 얻어낼 수 있다
- [ ]를 사용하여, 행번호 또는 열번호 지정 가능  
(순서는 [행(row), 열(column)])

```
> Family$Age
[1] 43 42 12  8  5
> Family$Gender
[1] Male  Female Female Male  Female
Levels: Female Male
```

```
> Family[1,]
  Name Age Gender Weight
1  Dad  43   Male     76
> Family[,3]
[1] Male  Female Female Male  Female
Levels: Female Male
> Family[4,2]
[1] 8
> Family[[2]]
[1] 43 42 12  8  5
> Family[2]
  Age
1  43
2  42
3  12
4   8
5   5
```

## 데이터 불러오기

- `read.csv(file = "")`
  - 1) data가 들어 있는 파일을 .csv로 바꾸어 저장한다: 엑셀에서 다른 이름으로 저장
  - 2) csv 파일을 R의 working directory에 넣는다: `getwd()`에서 확인한 폴더에 저장
  - 3) `read.csv` 명령어로 파일을 R에 불러온다
  - 4) 꼭 object 에 따로 저장한다: `data <- read.csv(file = "")`의 형태로 명령어 입력
  - 5) environment에 data가 제대로 불러졌는지 확인
- package 를 활용
  - csv: `readr`
    - `read_csv`
  - excel: `readxl`
    - `read_xls`
    - `read_xlsx`
    - `read_excel`
  - SPSS, SAS, STAT: `haven` / `foreign`
    - `read_spss` / `read_sav`
    - `read_sas`
    - `read_stata` / `read_dta`

## 데이터 저장하기

- `write.csv(x, file = “ “)`  
`write_csv(x, file = “ “)`  
`write_sav(x, file = “ “)`  
`write_dta(x, file = “ “)`
- `save(x, file = “ “)`  
save를 사용할 경우, Rdata 파일로 저장됨
- 저장된 데이터 파일은 working directory에서 찾을 수 있다.

---

## 04 R 로 기술통계량 구하기

## 데이터의 분포 확인

- `summary(data)`: 데이터의 기본적인 정보를 제공.
  - Numeric한 변수일 경우 `quantile`(최대, 최소, 25%, 50%, 75%)과 평균 출력
  - Character나 factor 변수 경우 빈도표 출력
- 기본 기술 통계량 구하는 함수

```

> mean(data$AGE)
[1] 44.237
> median(data$AGE)
[1] 44
> quantile(data$AGE)
 0%  25%  50%  75% 100%
 20  33  44  55  69
> quantile(data$AGE, .3)
30%
 36
> quantile(data$AGE, c(.3, .9))
30% 90%
 36  62
> min(data$AGE)
[1] 20
> max(data$AGE)
[1] 69
> range(data$AGE)
[1] 20 69
> var(data$AGE)
[1] 171.4002
> sd(data$AGE)
[1] 13.09199
> |

```



## 데이터의 분포 확인

- 기본 기술 통계량 구하는 함수(cont.)

```
> library("psych")
> describe(data$AGE)
  vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
X1     1 1000 44.24 13.09    44   44.43 16.31  20  69   49 -0.1    -1.13 0.41
```

- 빈도표 (frequency table) 그리기

```
> library("descr")
> freq(data$Q31)
응답자 최종학력
  Frequency Percent
1           2     0.2
2          11     1.1
3         191    19.1
4         105    10.5
5         583    58.3
6         108    10.8
Total      1000   100.0
```

- 히스토그램 그리기

```
> hist(data$AGE)
```



## 변수 간의 상관관계

- 두 변수의 상관관계 출력 함수

- '나는 행복하다' (Q1\_1)와 '5년 전에 비해 나의 가정 경제 상황은 좋아졌다' (Q1\_8)간의 상관관계

```
> cor.test(data$Q1_1, data$Q1_8)
```

Pearson's product-moment correlation

```
data: data$Q1_1 and data$Q1_8
t = 16.417, df = 998, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4108660 0.5085693
sample estimates:
      cor
0.4611141
```

- 여러 변수의 상관관계 출력 함수

- Q1\_1 ~ Q1\_8까지 8개의 변수들의 상관관계 출력

```
> data %>%
+   select(Q1_1:Q1_8) %>%
+   corr.test(.)
```

Call:corr.test(x = .)

Correlation matrix

	Q1_1	Q1_2	Q1_3	Q1_4	Q1_5	Q1_6	Q1_7	Q1_8
Q1_1	1.00	0.55	0.50	0.47	0.42	0.47	0.34	0.46
Q1_2	0.55	1.00	0.68	0.63	0.60	0.58	0.46	0.46
Q1_3	0.50	0.68	1.00	0.61	0.56	0.51	0.45	0.46
Q1_4	0.47	0.63	0.61	1.00	0.63	0.59	0.47	0.43
Q1_5	0.42	0.60	0.56	0.63	1.00	0.70	0.41	0.44
Q1_6	0.47	0.58	0.51	0.59	0.70	1.00	0.37	0.47
Q1_7	0.34	0.46	0.45	0.47	0.41	0.37	1.00	0.60
Q1_8	0.46	0.46	0.46	0.43	0.44	0.47	0.60	1.00

Sample Size

[1] 1000

Probability values (Entries above the diagonal are adjusted for multiple tests.)

	Q1_1	Q1_2	Q1_3	Q1_4	Q1_5	Q1_6	Q1_7	Q1_8
Q1_1	0	0	0	0	0	0	0	0
Q1_2	0	0	0	0	0	0	0	0
Q1_3	0	0	0	0	0	0	0	0
Q1_4	0	0	0	0	0	0	0	0
Q1_5	0	0	0	0	0	0	0	0
Q1_6	0	0	0	0	0	0	0	0
Q1_7	0	0	0	0	0	0	0	0
Q1_8	0	0	0	0	0	0	0	0

To see confidence intervals of the correlations, print with the short=FALSE option

---

## 0 5 데이터 정리하기

---

## 변수 리코드

- Ex) AREA 변수 중, 1) 서울, 2) 경기, 3) 인천을 수도권 1로, 나머지 지역을 비수도권 0으로 리코드 하고자 한다.
- 인덱싱 방법을 사용

```
> data$met_area <- 0
> data$met_area[data$AREA == 1 | data$AREA == 2 | data$AREA == 3] <- 1
> table(data$AREA, data$met_area)
```

	0	1
1	0	194
2	0	258
3	0	59
4	29	0
5	31	0
6	40	0
7	28	0
8	46	0
9	65	0
10	47	0
11	24	0
12	68	0
13	34	0
14	33	0
15	28	0
16	11	0
17	5	0

- data\$met\_area[data\$AREA < 4] <- 1 도 같은 결과

## 변수 리코드 (cont.)

- Ex) AREA 변수 중, 1) 서울, 2) 경기, 3) 인천을 수도권 1로, 나머지 지역을 비수도권 0으로 리코드 하고자 한다.
- “tidyverse” 식 방식

```
> data <- data %>%
+   mutate(met_area2 = ifelse(AREA < 4, 1, 0))
> table(data$AREA, data$met_area2)
```

	0	1
1	0	194
2	0	258
3	0	59
4	29	0
5	31	0
6	40	0
7	28	0
8	46	0
9	65	0
10	47	0
11	24	0
12	68	0
13	34	0
14	33	0
15	28	0
16	11	0
17	5	0

- mutate는 변수의 계산에서도 사용될 수 있다. 새로운 변수를 만들 때 사용

## 데이터 조작 with “tidyverse”

- R studio의 chief scientist이자, R 이용자들의 아이돌(?)인 Hadley Wickham 이 개발한 데이터 정리 패키지
- tidy data를 만들고, 데이터를 요약, 정리하는 데 탁월한 패키지
- 기존에 R의 불규칙한 문법들을 한번에 정리해 버림
- tidyverse 패키지 - dplyr, tidyr, ggplots, tibble 등의 패키지를 묶은 패키지
- dataframe을 tibble이라는 직관적인 구조로 변경해서 조작을 용이하게 해줌

함수	기능	유사함수
filter()	지정한 조건에 맞는 데이터 추출	subset()
select()	column, 변수의 추출	data[ , row number/변수명], subset()
mutate()	변수 계산하여 열을 추가	apply
arrange()	정렬	order(), sort()
summarise()	변수의 요약, 집계	aggregate()
group_by()	그룹을 나누어 줌	
join	데이터 결합	merge()

---

## 06 통계 분석

---

## T test

- 독립표본 t-test

- 가설: 남녀 간에 “나는 사회 경제적으로 안전한 사회에 살고 있다” (Q1\_3) 라고 인식하는 정도의 차이가 있다.

```
> bartlett.test(Q1_3 ~ SEX, data = data)
```

```
Bartlett test of homogeneity of variances
```

```
data: Q1_3 by SEX
```

```
Bartlett's K-squared = 0.019111, df = 1, p-value = 0.89
```

```
> t.test(Q1_3 ~ SEX, data = data, var.equal = TRUE)
```

```
Two Sample t-test
```

```
data: Q1_3 by SEX
```

```
t = 2.0958, df = 998, p-value = 0.03635
```

```
alternative hypothesis: true difference in means is not equal to 0
```

```
95 percent confidence interval:
```

```
 0.01654691 0.50298434
```

```
sample estimates:
```

```
mean in group 1 mean in group 2
```

```
 6.384766      6.125000
```

- 만약 bartlett.test에서 두 집단간 분산이 동일하지 않다고 나올 경우(p value가 <.05일 경우),

```
t.test(Q1_3 ~ SEX, data = data, var.equal = FALSE)
```



## T test

- 대응표본 t-test

- 가설: 환경 및 기후 변화 문제가 현재 삶에 영향을 주는 정도(Q3\_8)과 10년 후 삶에 영향을 주는 정도(Q5\_8)에 차이가 있다.

```
> t.test(data$Q3_8, data$Q5_8, paired = TRUE)
```

```
Paired t-test
```

```
data: data$Q3_8 and data$Q5_8
t = -6.0341, df = 999, p-value = 2.249e-09
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.3896113 -0.1983887
sample estimates:
mean of the differences
                -0.294
```

## ANOVA

- 가설: 교육수준(고졸 이하, 대학재학 이상, 대학원 이상)에 따라 “나는 행복하다”고 생각하는 정도에 차이가 있다
- 분석 순서
  - 집단 변수(교육수준) 리코드
  - Run “aov”함수
  - 사후검정

```
> result <- aov(Q1_1 ~ factor(educ), data = data)
> summary(result)
              Df Sum Sq Mean Sq F value    Pr(>F)
factor(educ)  2     61   30.579    7.982 0.000364 ***
Residuals   997   3820    3.831
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> TukeyHSD(result)
Tukey multiple comparisons of means
 95% family-wise confidence level

Fit: aov(formula = Q1_1 ~ factor(educ), data = data)

$`factor(educ)`
      diff      lwr      upr      p adj
2-1 0.4870041 0.12074473 0.8532635 0.0052647
3-1 0.8730937 0.32637243 1.4198149 0.0005513
3-2 0.3860896 -0.08942743 0.8616066 0.1375379
```

## Regression

- “나는 행복하다” (Q1\_1) 인식에 영향을 미치는 변수는 무엇인가?

- DV: Q1\_1

- ID: SEX, AGE, met\_area, Q31(교육수준), Q32(월평균 가구 소득),  
Q1\_8(5년 전에 비해 나의 가정 경제 상황은 좋아졌다)

```
> reg.result <- lm(Q1_1 ~ factor(SEX) + AGE + met_area + Q31 + Q32 + Q1_8, data = data)
> summary(reg.result)
```

Call:

```
lm(formula = Q1_1 ~ factor(SEX) + AGE + met_area + Q31 + Q32 +
    Q1_8, data = data)
```

Residuals:

```
   Min       1Q   Median       3Q      Max
-7.877 -1.077  0.133  1.127  5.785
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	3.594554	0.379422	9.474	<2e-16 ***
factor(SEX)2	0.201253	0.110730	1.818	0.0694 .
AGE	-0.004814	0.004261	-1.130	0.2588
met_area	0.118589	0.111140	1.067	0.2862
Q31	0.110445	0.059409	1.859	0.0633 .
Q32	0.065978	0.023861	2.765	0.0058 **
Q1_8	0.352244	0.023479	15.002	<2e-16 ***

---

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

Residual standard error: 1.737 on 993 degrees of freedom

Multiple R-squared: 0.2283, Adjusted R-squared: 0.2236

F-statistic: 48.95 on 6 and 993 DF, p-value: < 2.2e-16

## Regression

- “나는 행복하다” (Q1\_1) 인식에 영향을 미치는 변수는 무엇인가? (Cont.)

- standardized coefficients 구하기

```
> library("lm.beta")
> lm.beta(reg.result)
```

Call:

```
lm(formula = Q1_1 ~ factor(SEX) + AGE + met_area + Q31 + Q32 +
    Q1_8, data = data)
```

Standardized Coefficients::

(Intercept)	factor(SEX)2	AGE	met_area	Q31	Q32	Q1_8
0.00000000	0.05106555	-0.03197742	0.03009182	0.05411301	0.08261439	0.43369754

- 분산팽창계수(VIF 구하기)

```
> library("car")
> vif(reg.result)
```

factor(SEX)	AGE	met_area	Q31	Q32	Q1_8
1.015726	1.030791	1.023354	1.090160	1.148630	1.075324

- coefficients의 confident interval 구하기

```
> confint(reg.result)
```

	2.5 %	97.5 %
(Intercept)	2.849992273	4.339115428
factor(SEX)2	-0.016038336	0.418545183
AGE	-0.013175842	0.003547643
met_area	-0.099507303	0.336684889
Q31	-0.006136404	0.227026913
Q32	0.019153475	0.112802471
Q1_8	0.306169182	0.398319411

**Thank you**

email: [hjhwang@snu.ac.kr](mailto:hjhwang@snu.ac.kr)