Basic Statistics with R

서울대학교 언론정보학과
황현정

01 R 이란 무엇인가

R 소개

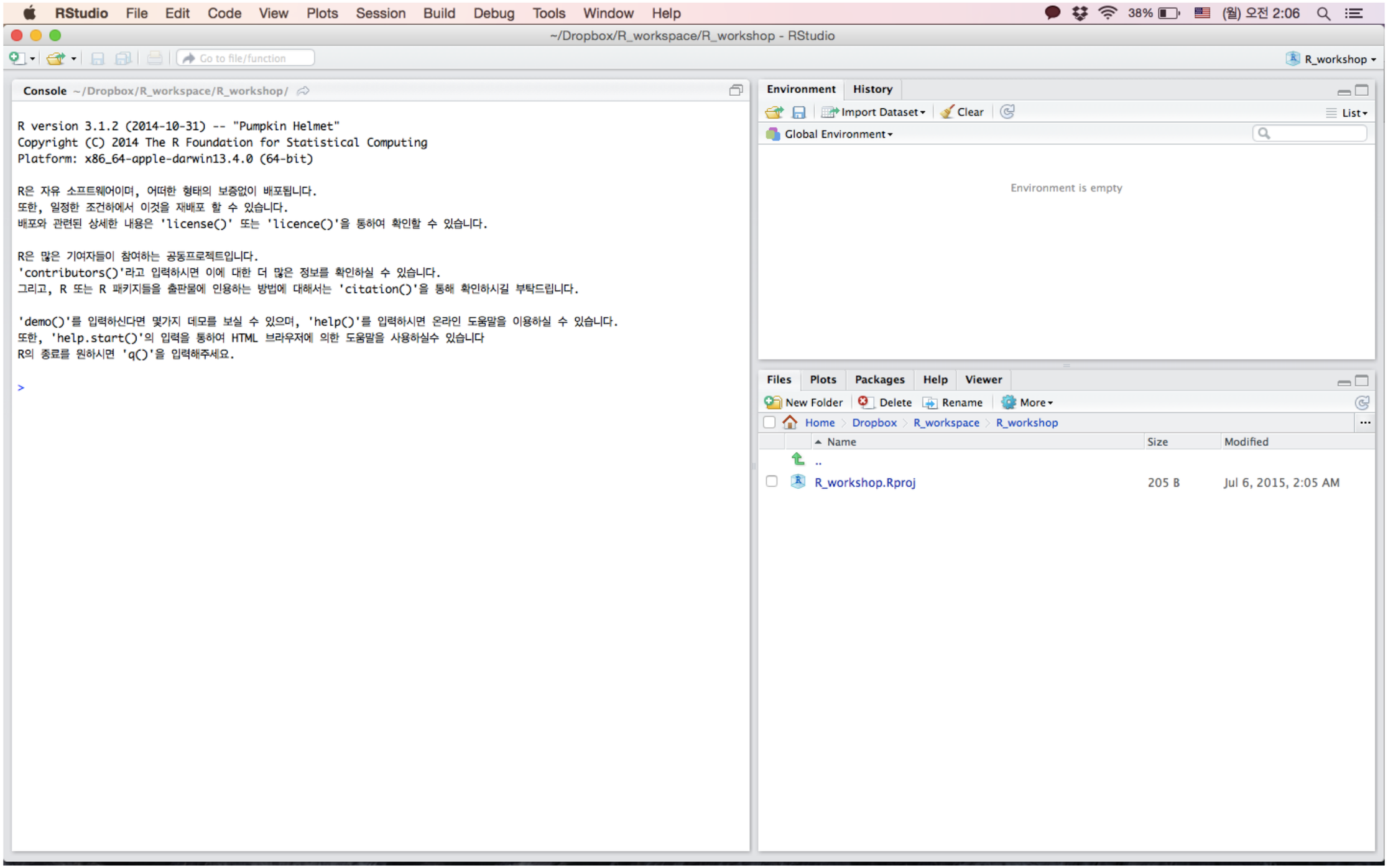
- R 프로그래밍 언어(줄여서 R)는 통계 계산과 그래픽을 위한 프로그래밍 언어이자 소프트웨어 환경이다.
  뉴질랜드 오클랜드 대학의 로스 이하카와 로버트 젠틀맨에 의해 시작되어 현재는 R 코어 팀이 개발하고 있다.
- R의 문법과 통계처리 부분은 AT&T 벨 연구소가 개발했던 S를 참고했고, 데이터 처리부분은 스킴에 영향을 받았다.
- R은 다양한 통계 기법과 수치 해석 기법을 지원한다.
  R은 사용자가 제작한 패키지를 추가하여 기능을 확장할 수 있다.
  핵심적인 패키지는 R과 함께 설치되며, CRAN(the Comprehensive R Archive Network)을 통해 2021년 현재 17,000개 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**

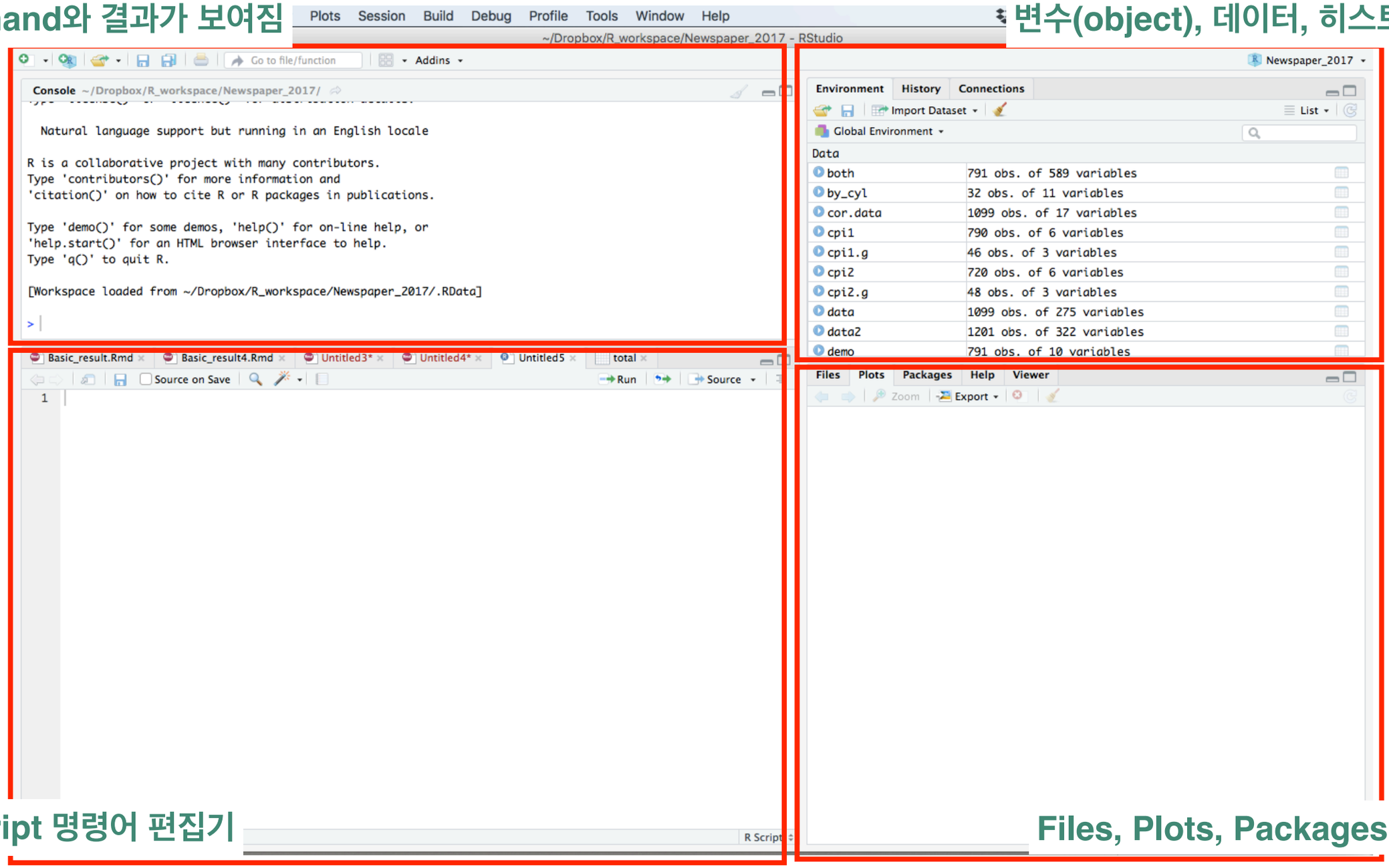
R Studio


RStudio File Edit Code View Plots Session Build Debug Tools Window Help
38% (월) 오전 2:06
~/Dropbox/R_workspace/R_workshop - RStudio
Go to file/function
R_workshop
Console ~/Dropbox/R_workspace/R_workshop/
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)
R은 자유 소프트웨어이며, 어떠한 형태의 보증없이 배포됩니다.
또한, 일정한 조건하에서 이것을 재배포 할 수 있습니다.
배포와 관련된 상세한 내용은 'license()' 또는 'licence()'을 통하여 확인할 수 있습니다.
R은 많은 기여자들이 참여하는 공동프로젝트입니다.
'contributors()'라고 입력하시면 이에 대한 더 많은 정보를 확인하실 수 있습니다.
그리고, R 또는 R 패키지들을 출판물에 인용하는 방법에 대해서는 'citation()'을 통해 확인하시길 부탁드립니다.
'demo()'를 입력하신다면 몇가지 데모를 보실 수 있으며, 'help()'를 입력하시면 온라인 도움말을 이용하실 수 있습니다.
또한, 'help.start()'의 입력을 통하여 HTML 브라우저에 의한 도움말을 사용하실수 있습니다
R의 종료를 원하시면 'q()'을 입력해주세요.
>
Environment History
Import Dataset Clear List
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home Dropbox R_workspace R_workshop
Name Size Modified
..
R_workshop.Rproj 205 B Jul 6, 2015, 2:05 AM

R Studio

Command와 결과가 보여짐

변수(object), 데이터, 히스토리 열람



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()	exponetial
sqrt()	제곱근(√)
==, !=	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)

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 방식으로도 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

  - row은 case, column은 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 불려졌는지 확인

- package 를 활용

  - csv: readr

    read_csv

  - excel: readxl

    read_xls

    read_xlsx

    read_excel

  - SPSS, SAS, STATA: haven / foreign

    read_spss / read_sav (read.spss / read.sav)

    read_sas (read.sas)

    read_stata / read_dta (read.stata / read.dta)

데이터 저장하기

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

03 R 데이터 보기

데이터 구조 보기

- str(data)
  - 데이터의 전체적인 구조를 보여줌
  - 데이터의 형태, 원소, 갯수 등을 한 눈에 파악 가능
- View(data)
  - 엑셀처럼 표의 형태로 보여줌. source 창 옆에 데이터를 열람할 수 있는 창이 열림
  - 전체 변수, 전체 개체수를 보여주지는 않음
- names(data)
  - 데이터의 변수 이름을 보여줌
- dim(data)
  - 데이터의 행과 열의 수를 확인
- head(data)
  - 데이터의 시작 6행을 보여줌
  - 6행이 아닌 지정된 숫자의 행을 보고 싶을 경우, head(data, 10)의 형태로 숫자를 넣어주면 됨
- tail(data)
  - 데이터의 끝 6행을 보여줌
  - 6행이 아닌 지정된 숫자의 행을 보고 싶을 경우, tail(data, 10)의 형태로 숫자를 넣어주면 됨
- summary(data)
  - 데이터의 통계적 요약

데이터 구성 원소의 종류

- str(data)를 입력하면, 데이터의 구조가 나옴
- $가 앞에 붙은 각 변수들의 원소가 어떠한 형태인지 확인할 수 있다
- R에서 찾아볼 수 있는 원소의 형태는 아래와 같다

  - numeric, double integer는 계산할 수 있으나, 나머지는 계산할 수 없다

  - 숫자도 “ ”로 묶어줄 경우, character로 인식 (R에게는 1은 숫자이나, “1”은 문자)
- 측정척도와 연결해서 생각하자
- 기타 형태:

  - Inf/-Inf: 무한대 / NA: 결측값 / NaN: 숫자가 아닌 값(Not a Number) / NULL: 미정

Mode	형태
numeric	3.14, 9, 5, -2, 0
double	3.14, 2.00, -2.3
integer	0, 1, 2, 3, 4, 5
character	“Mary”, “Dog”, “I”
factor	F, M / 1, 2, 3
logical	TRUE, FALSE
date/time	2015-07-07

데이터 속성 확인 및 변경

- is.*( ) 함수를 사용해서 데이터 속성을 확인할 수 있다
  - is.numeric(data)
  - is.integer(data)
  - is.character(data)
- as.*( ) 함수를 사용해서 데이터 속성을 변경할 수 있다
  - as.numeric(data)
  - as.integer(data)
  - as.factor(data)

인덱싱 연습

- data[row number, column number]
- data$variable name
- data$variable name[row number]
- data$variable name[조건]

기술 통계학(Descriptive Statistics)

- 자료의 요약
- 통계학은 기술통계학과 추론통계학(Inferential Statistics)로 나뉜다.
- 통계학의 꽃은 추론통계학

  - 그러나 추론통계에 들어가기 전에 반드시 기술통계에 대해 이해를 해야 한다.
- 기술통계는 내가 무엇을 가지고 있는지, 재료에 대한 탐구

일원분석(Univariate Analysis)

- 하나의 변수의 속성에 대해 탐구하는 것

  - 분포 (Distribution)
  - 중심경향 (Central Tendency)
  - 산포도(Dispersion or Spread)
  - 기울기(Skewness)

이원분석(Bivariate Analysis)

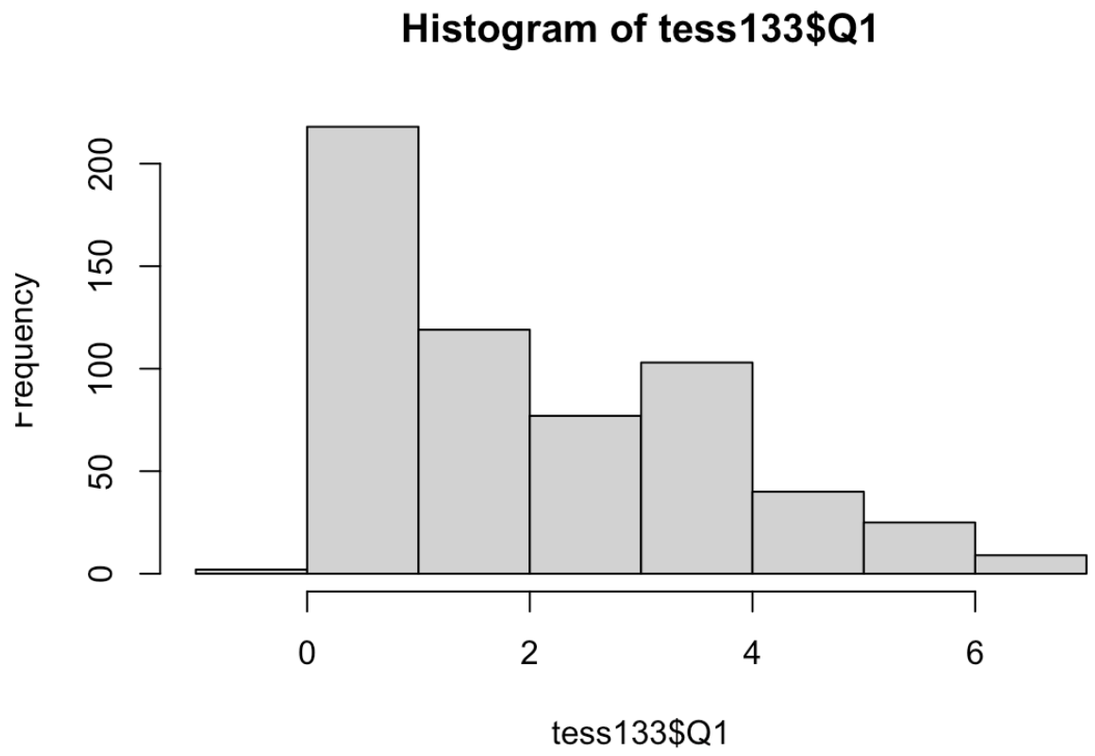
- 두 변수 간의 관계에 대한 요약

  - 교차 분석표 (Cross table)
  - 산점도 (Scatter plot)
  - 평균비교

일원분석 - 분포

- 변수의 종류, 척도, 변수값의 수에 따른 분포
- 빈도표(Frequency table): 변수의 속성들이 관찰되는 수에 대한 기술
- 막대그래프, 파이그래프
- 히스토그램

```
> library("descr")
> freq(tess133$Q1)
How about you, how much do you know about student loan forgiveness policies?
      Frequency  Percent
-1            2   0.3373
1           218  36.7622
2           119  20.0675
3            77  12.9848
4           103  17.3693
5            40   6.7454
6            25   4.2159
7             9   1.5177
Total       593 100.0000
```



일원분석 - 중심경향치

- 데이터의 전형적인(typical) 값 또는 대푯(representative)값
- 평균(mean)

  - 모든 변수 값을 더한 후 총 사례 수로 나누어 구한 값 (∑X/n)

  - mean(변수, na.rm = TRUE)

- 중위수 또는 중앙값(median)

  - 분포를 50대 50으로 가르는 값 즉, 50분위수(50th Percentile)

  - median(변수, na.rm = TRUE)

  - quantile(변수, na.rm = TRUE)

    quantile(변수, prob = .5)

- 최빈값(mode)

  - 가장 빈번하게 관찰되는 값이나 속성을 보여주는 값

  - 별도 함수를 만들어야 함

일원분석 - 산포도

- 데이터가 얼마나 퍼져있는가
- 범위(range)

  - 최대값 - 최소값

  - range(변수, na.rm = TRUE)

  - max(변수, na.rm = TRUE)

  - min(변수, na.rm = TRUE)

- 분산(variance)

  - 각 값이 평균적으로 평균으로부터 얼마나 떨어져 있는가?

  - 편차(평균 각 값)을 제곱하여 더한 후, 사례 수로 나눈 값

  - var(변수, na.rm = TRUE)

- 표준편차(standard deviation)

  - 분산의 제곱근

  - sd(변수, na.rm = TRUE)

일원분석 - 기울기

- 왜도(Skewness)
  - 양의 기울기: 긴 꼬리가 오른쪽
  - 음의 기울기: 긴 꼬리가 왼쪽
  - 0이면 양쪽이 대칭
- 첨도(Kurtosis)
  - 얼마나 표족한 분포인가
  - 숫자가 클수록 뾰족한(중심에 급격히 몰려있는 분포)

- 첨도와 왜도를 구하는 함수는 moments라는 패키지에 잇음
  - skewness(변수, na.rm = TRUE)
  - kurtosis(변수, na.rm = TRUE)

이원분석 - 교차분석(Crosstab analysis)

- Contingency table (교차표 또는 분할표)

  - 두 범주형 변수 (명목, 순서 척도 변수)의 관계를 검증하는 데 쓰임.

- 이변량 결합 빈도표(bivariate joint frequency table)

  - Crosstab은 cross-tabulation의 약어로서 테이블 내의 각 cell 안에는 두 변수의 각 값에 공통으로 들어가는 사례 수가 들어감.

- 교차분석표 만들기

  1)독립변수는 열(column)에 위치

  2)종속변수는 행(row)에 위치

  3)독립변수 기준으로 퍼센트 계산 - 독립변수 방향으로 100%가 되도록

  4)종속변수의 한 값에 초점을 맞추어 독립변수 값 변화 시 가설에 부합하는 변화가 있는지 살펴본다

- crosstab(종속변수, 독립변수, prop.c = TRUE, chisq = TRUE)

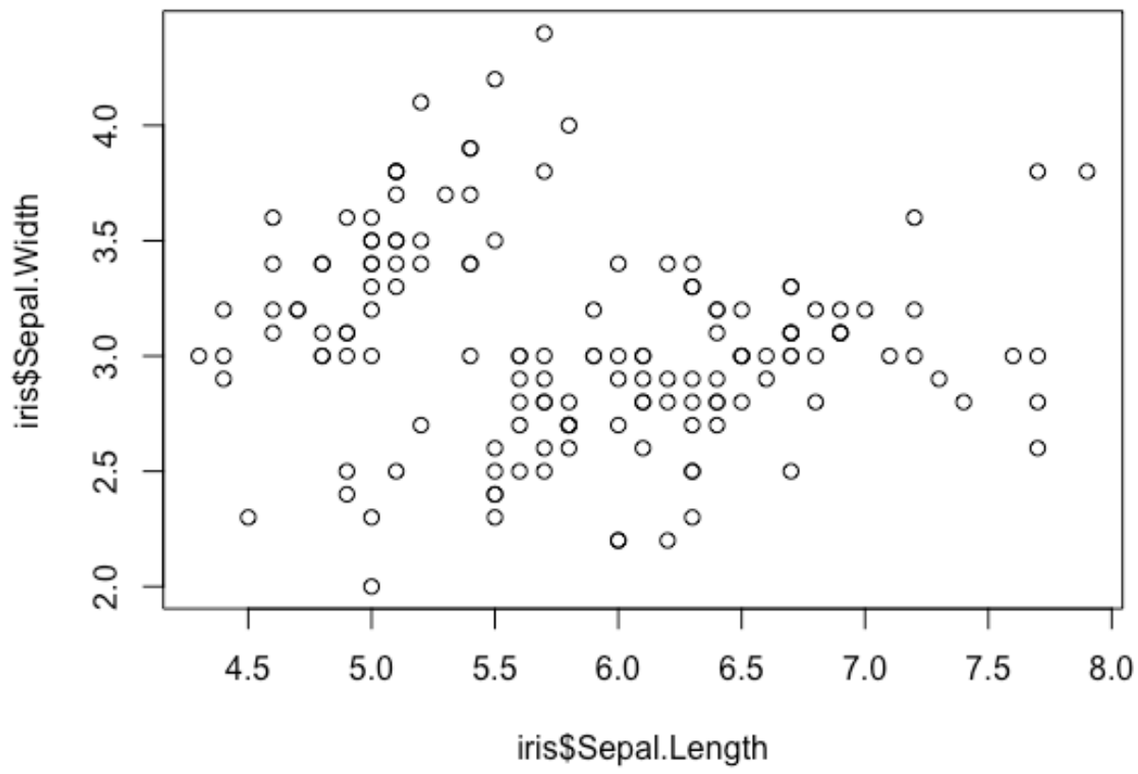
  *descr 패키지에 있음

이원분석 - 산포도

- 두 변수 모두 구간 혹은 비율 척도로 측정된 변수인 경우
- 순위 척도라도 측정값(value)이 많은 경우 사용할 수 있다.
- 산점도내 한 점은 한 사례(case)를 나타낸다.
- 독립변수는 X축에 종속변수는 Y축에 배치
- 상관계수(correlation coefficient)로 요약할 수 있다.

  -우상향 혹은 우하향의 선형(linear)관계
- cor.test(x, y, method = “pearson")
- corr.test(data, method = “pearson”)

  * psych 패키지에 있음



05 기초 통계 분석

t test

- 두 집단의 평균 비교
- t test 절차

  1) 변수 정리

  2) t test의 종류 확인: 단일 표본 / 독립 표본 / 대응 표본

  3) 독립 표본일 경우, 분산 동질성 test 하기: bartlett.test(종속변수 ~ 독립변수, data = data)
- 문법

1) 단일표본: t.test(변수, mu = mu)

2) 독립표본: t.test(종속변수 ~ 독립변수, data = data, var.equl = FALSE)

3) 대응표본: t.test(변수1, 변수2, data = data, paired = T)

ANOVA

- 셋 이상의 집단 간 평균 비교
- ANOVA 절차

  1) 변수 정리: 독립변수(집단을 구분하는 변수)는 factor의 형태여야 한다 (numeric 안됨)

  2) 집단 별 평균 확인하기

  3) aov 실행:

  result <- aov(종속변수 ~ 독립변수, data = data) **result*는 임의의 *object*

  summary(result)

  4) 사후 검정:

  TukeyHSD(result, ordered = FALSE)

- 이원 분산 분석 (two-way ANOVA)

  - 상호작용 가정하지 않을 때: aov(종속변수 ~ 독립변수1 + 독립변수2, data = data)

  - 상호작용 가정할 때: aov(종속변수 ~ 독립변수1 * 독립변수2, data = data)

OLS 회귀분석

- 종속변수와 독립변수 간의 선형 관계의 예측 —> 상관관계의 추정
- OLS 회귀분석 절차

  1) 변수정리: factor 변수들 처리에 주의.

  2) 변수들의 분포 확인: 종속변수가 정규분포가 아닐 경우, 다른 모형을 고려하기

  3) 회귀모형의 추정

  result <- lm(종속변수 ~ 독립변수1 + 독립변수2 + …, data = data) **result*는 임의의 *object*

  summary(result)

  4) standardized coefficient 확인

  lm.beta(result) **lm.beta*는 *"lm.beta" library*에 있음

  5) vif(분산팽창계수) 확인

  vif(result) **vif*와 *confint*는 *"car" library*에 있음

  6) confident interval 확인

  confint(result)

Thank you

email: hjhwang@snu.ac.kr