1장 R 개요

https://en.wikipedia.org/wiki/R_(programming_language)

R 설치
https://www.r-project.org/

CRAN (Comprehensive R Archive Network)

Install r ver. 4.0.0

Rstudio 설치
https://www.rstudio.com/products/rstudio/download/

Rstudio setting

```
# 실습: R 패키지 보기
dim(available.packages())

# a<-available.packages()
# head(a)

# 실습: R 패키지 목록 보기
available.packages()

# 실습: R Session 보기
# R Session: 사용자가 R 프로그램을 기동한 이후 R 콘솔 시작과 종료 전까지의 기간에 수행된 정보

sessionInfo()

# 실습: stringr 패키지 설치
```

```
install.packages(("stringr"))
# 패널 내 Packages 에서 install로 설치


# 실습: 설치된 패키지 확인
installed.packages()

# 실습: 패키지 로드
library(stringr)
search() # 현재 load된 패키지 확인

require(stringr)

# 실습: 패키지 제거
remove.packages("stringr")

# 실습: 기본 데이터 셋 보기
data()


# 실습: 기본 데이터 셋으로 히스토그램 그리기
# 단계 1: 빈도수를 기준으로 히스로그램 그리기
hist(Nile)
# 단계 2: 밀도를 기준으로 히스토그램 그리기
hist(Nile, freq = F)
# 단계 3: 단계 2의 결과에 분포 곡선을 추가
lines(density(Nile))


# working directory 설정
getwd()
setwd('C:/Temp2/Rwork')
getwd()
```

4. 변수와 자료형

4.1 변수

변수이름 명명 방법

(1) 변수 이름 명명 규칙 in R
1) 첫 글자는 영문자로 시작
2) 2 번째 단어부터는 숫자, _, . 등을 사용 가능
3) 대문자와 소문자를 구별한다.
4) 변수의 이름은 의미에 맞도록 작성한다.
5) 한 번 정의된 변수는 재사용이 가능하고, 가장 최근의 값이 저장되어 있다.
6) Camel case 선호

```
# 실습: 변수 사용 예
var1 <- 0
var1
var1 <- 1
var1
var2 <- 2
var2
var3 <- 3
var3
```

```
# 실습: '변수.멤버' 형식의 변수 선언
goods.code <- 'a001'
goods.name <- '냉장고'
goods.price <- 850000
goods.des <- '최고 사양, 동급 최고 품질'
```

(2) 스칼라(scalar)변수
스칼라(scalar)변수: 한 개의 값만 갖는 변수
벡터(vector)변수: 복수의 자료를 저장할 수 있는 1차원의 선형 자료 구조

```
# 실습: 벡터 변수 사용 예
```

```
age <- 35
names <- "홍길동"
age
names
# 실습: 벡터 변수 사용 예
age <- 35
names <- c("홍길동", "이순신", "유관순")
age
names
```

4.2 자료형

R은 변수를 선언할 때 별도의 자료형(type)을 선언하지 않음.

[표1-1] 기본자료형
숫자형: 124.123
문자형: "홍길동"
논리형: TRUE, T, FALSE, F
결측 데이터: NA(Not Available), NaN(Not a number)

실습

```
# 스칼라 변수 사용 예
int <- 20
int
string <- "홍길동"
string
boolean <- TRUE
boolean
sum(10, 20, 20)
sum(10, 20, 20, NA)
sum(10, 20, 20, NA, na.rm = TRUE)
ls()
```

sum() 함수: 주어진 인수를 이용하여 합계를 구하는 함수
https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/sum

option 'na.rm = T' : NA 제거
ls() 함수: 현재 메모리에 할당된 변수(객체) 확인하는 함수
https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/ls

(1) 자료형 확인
자료형 확인 함수

[표1.2]
is.numeric(), is.logical(), is.character(), is.data.frame(), is.na(), is.integer(), is.double(), is.complex(), is.factor(), is.nan()

실습 (자료형 확인)
is.character(string)

x <- is.numeric(int)
x

is.logical(boolean)
is.logical(x)
is.na(x)

(2) 자료형 변환

다른 자료형으로 변환하는 함수

[표 1-3]
as.numeric(), as.logical(), as.character(), as.data.frame(), as.list(), as.array(), as.integer()., as.double(),. as.complex(), as.factor(), as.vector(), asDate()

실습 (문자 원소를 숫자 원소로 형 변환)

```
x <- c(1, 2, "3")
x

result <- x * 3
result <- as.numeric(x) * 3
#result <- as.integer(x) * 3
result
```

c() 함수: 벡터(여러 개의 자료를 저장할 수 있는 1차원의 선형 자료구조) 생성

실습 (복소수)

```
z <- 5.3 - 3i
Re(z)
Im(z)
is.complex(x)
as.complex(5.3)
```

(3) 자료형과 자료구조 보기

mode()함수: 자료형 확인
https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/mode
mode()함수 in other packages
https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/class

class()함수: 자료구조(메모리 구조) 확인
https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/class

*스칼라 변수인 때는 mode()와 class() 결과 동일

# 실습: 스칼라 변수의 자료형과 자료구조 확인
mode(int)
mode(string)
mode(boolean)

class(int)
class(string)
class(boolean)

(4) 요인(Factor)형 변환
요인(Factor): 값의 목록을 범주(category)로 갖는 벡터 자료

Nominal: 범주의 순서는 알파벳 순서로 정렬
Ordinal: 범주의 순서는 사용자가 지정한 순서대로 정렬

실습(문자 벡터와 그래프 생성)
gender <- c("man", "woman", "woman", "man", "man")
plot(gender)
➔ error

plot() 함수: 막대그래프 그리기. 숫자 데이터만을 대상으로 그래프 생성 가능.
https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/plot

Factor Nominal
벡터 원소를 요인형으로 변환한 경우 범주의 순서가 알파벳순으로 정렬

P45 실습(요인형 변환)
Ngender <- as.factor(gender)
table(Ngender)

as.factor(): 요인형 변환

table()함수: 빈도수 계산

p46 실습(factor형 변수로 차트 그리기)

```
plot(Ngender)
mode(Ngender)
class(Ngender)
is.factor(Ngender)
```

is.factor()함수: 요인형 확인

```
# 실습: Factor Nominal 변수 내용 보기
Ngender
```

Factor Ordinal

factor(): 범주의 순서를 사용자가 지정한 순서대로 정렬하는 기능

형식:

```
factor(x, levels, ordered)
```

* levels에서 사용자가 정한 순서대로 정렬

P47 실습 factor() 함수를 이용한 Factor형 변환

```
args(factor)
Ogender <- factor(gender, levels = c("woman", "man"), ordered = T)
Ogender
```

args(name): 함수의 argument 이름과 관련한 default값 표시(cf. p51)

p47 실습(순서가 없는 요인과 순서가 있는 요인형 변수로 차트 그리기)

```
par(mfrow = c(1, 2))
plot(Ngender)
plot(Ogender)
```

par()함수: 그래프 파라미터의 지정
https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/par

(5) 날짜형 변환

요인형 또는 문자형으로 인식되는 날짜를 날짜형으로 변환

실습(날짜형 변환)

```
as.Date("20/02/28", "%y/%m/%d")
class(as.Date("20/02/28", "%y/%m/%d"))
dates <- c("02/28/20", "02/30/20", "03/01/20")
as.Date(dates, "%m/%d/%y") # 해당 날짜가 없는 경우 NA
```

as.Date()함수: 날짜형 변환
형식:
as.Date(변수, format)
https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/as.Date

[표 1.4] 날짜와 시간 표현을 위한 제어문자
%Y: 년도 4자리, %y: 년도 2자리
%m: 월, %d: 일
%H: 24시간, %I: 12시간
%M: 분, %S: 초

실습 (시스템 로케일 정보 확인)

```
Sys.getlocale(category = "LC_ALL")
Sys.getlocale(category = "LC_COLLATE")
```

로케일(Locale)

SW현지화 자료

실습 (현재 날짜와 시간 확인)

Sys.time()

Sys.time() 함수: 로케일 정보에 따른 현재 날짜와 시간
https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/Sys.time

실습 (날짜형 변환)

```
sdate <- "2019-11-11 12:47:5"
class(sdate)

today <- strptime(sdate, format = "%Y-%m-%d %H:%M:%S")
class(today)
```

Strptime()함수 이용
형식:
strptime(x, format)

https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/strptime

as.Date()함수와 strptime()함수는 문자 상수를 날짜형으로 변환할 경우 사용
as.Date()함수는 날짜 자료만 형변환이 가능

strptime()함수에서 날짜형과 시간형은 POSIXlt방식을 이용

==========

The details of the formats are platform-specific, but the following are likely to be widely available: most are defined by the POSIX standard.
https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/DateTimeClasses

*POSIX: 이식가능 운영체제 인터페이스(Portable Operating System Interface),
IEEE가 ㅊ설정한 애플리케이션 인터페이스 규격.
마지막 X는 유닉스 호환 운영체제에 보통 X가 붙는 것에서 유래.
Windows 2003 R2부터 POSIX 2.0에 준하는 Subsystem for UNIX-based Application(SUA)를 통해 POSIX지원

A *conversion specification* is introduced by %, usually followed by a single letter or O or E and then a single letter. Any character in the format string not part of a conversion specification is interpreted literally (and %% gives %). Widely implemented conversion specifications include

%a
Abbreviated weekday name in the current locale on this platform. (Also matches full name on input: in some locales there are no abbreviations of names.)

%A
Full weekday name in the current locale. (Also matches abbreviated name on input.)

%b
Abbreviated month name in the current locale on this platform. (Also matches full name on input: in some locales there are no abbreviations of names.)

%B
Full month name in the current locale. (Also matches abbreviated name on input.)

%c
Date and time. Locale-specific on output, "%a %b %e %H:%M:%S %Y" on input.

%C
Century (00--99): the integer part of the year divided by 100.

%d
Day of the month as decimal number (01--31).

%D
Date format such as %m/%d/%y: the C99 standard says it should be that exact format (but not all OSes comply).

%e
Day of the month as decimal number (1--31), with a leading space for a single-digit number.

%F
Equivalent to %Y-%m-%d (the ISO 8601 date format).

%g
The last two digits of the week-based year (see %V). (Accepted but ignored on input.)

%G
The week-based year (see %V) as a decimal number. (Accepted but ignored on input.)

%h
Equivalent to %b.

%H
Hours as decimal number (00--23). As a special exception strings such as 24:00:00 are accepted for input, since ISO 8601 allows these.

%I
Hours as decimal number (01--12).

%j
Day of year as decimal number (001--366).

%m
Month as decimal number (01--12).

%M
Minute as decimal number (00--59).

%n
Newline on output, arbitrary whitespace on input.

%p
AM/PM indicator in the locale. Used in conjunction with %I and not with %H. An empty string in some locales (for example on some OSes, non-English European locales including Russia). The behaviour is undefined if used for input in such a locale.

Some platforms accept %P for output, which uses a lower-case version (%p may also use lower case): others will output P.

%r
For output, the 12-hour clock time (using the locale's AM or PM): only defined in some locales, and on some OSes misleading in locales which do not define an AM/PM indicator. For input, equivalent to %I:%M:%S %p.

%R
Equivalent to %H:%M.

%S
Second as integer (00--61), allowing for up to two leap-seconds (but POSIX-compliant implementations will ignore leap seconds).

%t
Tab on output, arbitrary whitespace on input.

%T
Equivalent to %H:%M:%S.

%u
Weekday as a decimal number (1--7, Monday is 1).

% see https://en.wikipedia.org/wiki/Week_number#Week_number%U
Week of the year as decimal number (00--53) using Sunday as the first day 1 of the week (and typically with the first Sunday of the year as day 1 of week 1). The US convention.

%V
Week of the year as decimal number (01--53) as defined in ISO 8601. If the week (starting on Monday) containing 1 January has four or more days in the new year, then it is considered week 1. Otherwise, it is the last week of the previous year, and the next week is week 1. (Accepted but ignored on input.)

%w
Weekday as decimal number (0--6, Sunday is 0).

%W
Week of the year as decimal number (00--53) using Monday as the first day of week (and typically with the first Monday of the year as day 1 of week 1). The UK convention.

%x
Date. Locale-specific on output, "%y/%m/%d" on input.

%X
Time. Locale-specific on output, "%H:%M:%S" on input.

%y
Year without century (00--99). On input, values 00 to 68 are prefixed by 20 and 69 to 99 by 19 -- that is the behaviour specified by the 2004 and 2008 POSIX standards, but they do also say ‘it is expected that in a future version the default century inferred from a 2-digit year will change’.

%Y
Year with century. Note that whereas there was no zero in the original Gregorian calendar, ISO 8601:2004 defines it to be valid (interpreted as 1BC): see https://en.wikipedia.org/wiki/0_(year). Note that the standards also say that years before 1582 in its calendar should only be used with agreement of the parties involved.

For input, only years 0:9999 are accepted.

%z
Signed offset in hours and minutes from UTC, so -0800 is 8 hours behind UTC. Values up to +1400 are accepted. (Standard only for output. For input R currently supports it on all platforms.)

%Z
(Output only.) Time zone abbreviation as a character string (empty if not available). This may not be reliable when a time zone has changed abbreviations over the years.
==========

실습 (4자리 연도와 2자리 연도 표기)
strptime()함수 이용
위의 %제어문자로 실습

```
strptime("30-11-2019", format = ("%d-%m-%Y"))
strptime("30-11-19", format = ("%d-%m-%y"))
```

p49 실습 (국가별 로케일 설정)
setlocale()함수: 로케일 설정

형식:
Sys.setlocale(category = 'LC_ALL, locale="언어_국가")

설정된 로케일로 문자, 숫자, 날짜/시간 형식 사용 가능

```
# 실습: 국가별 로케일 설정
Sys.setlocale(category = "LC_ALL", locale = "")

Sys.setlocale(category = "LC_ALL", locale = "Korean_Korea")
```

```
Sys.setlocale(category = "LC_ALL", locale = "English_US")

Sys.setlocale(category = "LC_ALL", locale = "Japanese_Japan")

Sys.getlocale()

# 실습: 미국식 날짜 표현을 한국식 날짜 표현으로 변환
strptime("01-nov-19", format = "%d-%b-%y") # 날짜 형식을 인식 못해서 NA 출력

Sys.setlocale(category = "LC_ALL", locale = "English_US")

strptime("01-nov-19", format = "%d-%b-%y")
day <- strptime("tuesday, 19 nov 2019", format = "%A,%d %b %Y")
day <- strptime("Tue, 19 nov 2019", format = "%a,%d %b %Y")
weekdays(day)
strptime("19 Nov 19", format = "%d %b %y")
day <- c("1may99", "2jun01", "28jul15")
strptime(day, format = "%d%b%y")
```

5. 기본함수와 작업공간

5.1 함수 사용

(1) 함수 도움말

형식:
help(함수명)
?함수명
함수명 in r at google.com
검색 in RStudio
# help(함수명, package="패키지명")
help(rlm, package="MASS")

(2) 함수 파라미터 보기

arg(함수명)함수: 특정함수를 대상으로 사용 가능한 함수의 파라미터를 보여준다

실습 (함수 파라미터 확인)

```
args(max)    # max 함수의 파라미터 확인
max(10, 20, NA, 30)
```

(3) 함수 사용 예제

example()함수: R에서 제공하는 기본 함수의 사용예제
형식:
example(함수명)

실습(example()함수 사용)

```
example(seq)
```

seq()함수: 일반 시퀀스 생성

example(max)

p53 실습(mean()함수 사용)

example(mean)

mean(10:20)

x <- c(0:10, 50)
mean(x)

mean()함수: 평균 계산
*NA값 포함 여부에 따른 계산결과 확인!

5.2 작업공간

(1) 작업공간 보기
getwd()함수: 기본 작업공간 확인

p53 실습 (현 작업공간 보기)
getwd()

(2) 작업공간 변경
작업 디렉토리 변경

setwd("~/Rwork/")
data <- read.csv("test.csv", header = T)
data

5.3 스크립트 파일 저장하기

UTF-8 설정

5.4 스크립트 파일 불러오기

File>>Open File...

Reference:

김진성, 빅데이터분석을 위한 R 프로그래밍, 2nd edit, 2020

https://intro2r.com/

https://rstudio-education.github.io/hopr/index.html