

R 기초 문법 ① (R특징, 데이터형, 변수)

2014.11.27 15:22

1. R 관련 특징

- R은 인터프리터 언어 (cf.컴파일방식=C언어 등)
- 대소문자 구분
- ↑ 방향키로 이전에 했던 작업 수행 가능
- q() 사용하면 R 종료
- 작업하는 내용을 저장하거나 작업용 데이터를 보관하는 작업 디렉토리를 지정하는것이 좋음
- 스크립트창을 열어서 코드를 입력한 후 해당 부분을 선택한 후 ctrl+R을 눌러서 실행

: 주석

> : 명령 프롬프트

+ : 여러줄에 명령을 칠때 다음줄의 가장 왼쪽에 생성됨

2. 작업디렉토리 지정하기 & 화면에 결과 출력

① setwd("디렉토리명")

- R로 작업을 할때 필요한 데이터들을 미리 가져다 두는 약속된 디렉토리
- 작업 후 나오는 결과물들도 기본적으로 저장됨
- 작업하기전 디렉토리 생성 후 분석할 소스 데이터들을 생성한 디렉토리로 옮겨놓고 작업

② print() / cat()

- R 콘솔 화면에서 print명령을 이용하여 내용 출력
- print() : 1번에 1가지만 출력가능 (단점)
- cat() : 여러개 출력 가능, but, 복잡한 데이터 형태는 출력 불가(행렬, 리스트 등)
- 여러개의 명령을 연속적으로 실행하고 싶을경우 세미콜론(;)사용

3. 기본 데이터 형

① 숫자형과 주요 산술연산자

기호	의미	사용 예 -> 결과
+	더하기	1+2 -> 3
-	빼기	2-1 -> 1
*	곱하기	1*2 -> 2
/	나누기(실수 가능)	4/2 -> 2
%/%	정수나누기	
%%	나머지	5%%4 -> 1
^, **	승수	3^2 -> 9
		3**3 -> 27

^ 접기

```
> 1+2
```

```
[1] 3
```

```
> 1+2*3
```

```
[1] 7
```

```
> (1+2)*3
```

```
[1] 9
```

```
>
```

```
> #0이 5개부터는 e로 표시
```

```
> 1000
```

```
[1] 1000
```

```
> 100000
```

```
[1] 1e+05
```

```
> 1000000
```

```
[1] 1e+06
```

```
>
```

```
> 1e2
```

```
[1] 100
```

```
> 3e2 #3*10^2=3*100
```

```
[1] 300
```

```
> #-1은 소수점 1자리까지 표시
```

```
> 3e-1
```

```
[1] 0.3
```

```
> #-2는 소수점 2자리까지 표시
```

```
> 3e-2
```

```
[1] 0.03
```

```
>
```

```
> # 3은 숫자, "3", '3'은 문자
```

```
> '1'+2'
```

다음에 오류가 있습니다"1" + "2" : 이항연산자에 수치가 아닌 인수입니다

```
>
```

```
> as.numeric('1') + as.numeric('2') #숫자로 강제변환
```

```
[1] 3
```

② 문자형

- 문자를 처리할때는 쌍따옴표(" ")나 홑따옴표(' ')를 사용해야함
- 1글자인 문자형과 여러글자인 문자열의 구분없이 전부 문자열 형태
- 어떤 데이터가 숫자형인지 문자형인지 알수 없다면 class()함수 사용

^ 접기

```
> # 따옴표를 사용 -> 문자로인식
```

```
> 'First'
```

```
[1] "First"
```

```
> "Second"
```

```
[1] "Second"
```

```
> # 따옴표없이 그냥 사용하면 변수이름으로 인식 -> 에러발생
```

```
> First
```

에러: 객체 'First'를 찾을 수 없습니다

```
> # 데이터의 형을 알고싶다면 class()
```

```
> class('1') <- 문자
```

```
[1] "character"
```

```
> class(1) <- 숫자
```

```
[1] "numeric"
```

```
>
```

③ TRUE / FALSE 값 (진리값)

- 데이터들을 비교하여 참일 경우 true, 거짓일 경우 false 반환
- & : 양쪽데이터가 모두 참일 경우만 true, 곱하기로 간주(and)
- | : 한가지만 참이어도 결과를 false, 더하기로 간주(or)
- ! : 해당 데이터가 아닌것, not
- 0(false)을 제외한 나머지는 참(true)

^ 접기

```
> # 참 x 거짓=거짓
```

```
> 3&0
```

```
[1] FALSE
```

```
> # 참 x 참 = 참
```

```
> 3&1
```

```
[1] TRUE
```

```
> # 참 + 거짓 = 참
```

```
> 3|0
```

```
[1] TRUE
```

```
> # 참 + 참 = 참
```

```
> 3|1
```

```
[1] TRUE
```

```
> # 거짓이 아닌것 = 참
```

```
> !0
```

```
[1] TRUE
```

```
> # 참이 아닌것 = 거짓
```

```
> !2
```

```
[1] FALSE
```

④ NA / NULL 형

- NA (Not Applicable 또는 Not Available) : 정해진 범위 안에 있는 값이 아니라서 사용할수 없는경우
- NA 값인지 아닌지 여부 확인 : **is.na(변수명)**
- NA값이 포함된 연산의 결과는 무조건 NA
- NA 값은 연산 대상에 포함되지만 but, NULL 값은 연산 대상에서 제외
- NA 값을 수동으로 제거 : **na.rm** **파라미터** 사용
- NULL : 값이 정해지지 않아서 얼마인지 모르는 경우

^ 접기

```
> # NA는 그대로 NA로 출력
```

```
> cat(1,NA,2)
```

```
1 NA 2
```

```
> # NULL값은 제외되고 출력됨
```

```
> cat(1,NULL,2)
```

```
1 2
```

```
> # NA가 포함된 연산 결과는 NA
```

```
> sum(1,NA,2)
```

```
[1] NA
```

```
> # NULL이 포함된 연산은 NULL을 제외한 연산결과 출력
```

```
> sum(1,NULL,2)
```

```
[1] 3
```

```
> # na.rm=T 파라미터로 NA값을 제거하고 연산 -> 올바른 결과 출력
```

```
> sum(1,2,NA, na.rm=T)
```

```
[1] 3
```

⑤ Factor 형

- 여러번 중복으로 나오는 데이터들을 각 값으로 모아서 대표값을 출력해줌
- c() 함수는 combine value라는 의미로 여러개의 값을 한꺼번에 처리할때 사용되는 함수

^ 접기

```
> c('a','a','b','b','b','c')
[1] "a" "a" "b" "b" "b" "c"
>
> factor(c('a','a','b','b','b','c'))
[1] a a b b b c
Levels: a b c
```

⑥ 날짜와 시간

- R에서는 날짜나 시간같은 데이터가 아주 중요하게 생각됨
- 데이터를 분석할때 특정 시간 또는 특정 기간을 지정하여 분석하는 경우가 많기때문
- Sys.Date() : 날짜만 보여주는 함수
- Sys.time() : 날짜와 시간
- date() : (미국식) 날짜와 시간
- as.Date() : 문자형태로 저장된 날짜를 날짜로 변경 + format사용가능

FORMAT

```
%d : 일자를 숫자로 인식
%m : 월을 숫자로 인식
%b : 월을 영어 약어로 인식
%B : 월을 영어로 인식
%y : 년도를 숫자 두자리로 인식
%Y : 년도를 숫자 네자리로 인식
```

^ 접기

```
> # 날짜만 보여주는 함수
> Sys.Date()
```

```
[1] "2014-11-27"
```

```
> # 시간만 보여주는 함수
```

```
> Sys.time()
```

```
[1] "2014-11-27 15:36:37 KST"
```

```
> # 날짜와 시간
```

```
> date()
```

```
[1] "Thu Nov 27 15:36:49 2014"
```

```
> # 문자형태로 저장된 날짜를 날짜로 변경
```

```
> as.Date('2014-11-27')
```

```
[1] "2014-11-27"
```

```
> as.Date("2014/11/27")
```

```
[1] "2014-11-27"
```

```
> # as.Date()의 기본포맷 : YYYY-MM-DD
```

```
> as.Date("01-11-2014")
```

```
[1] "0001-11-20"
```

```
> as.Date("01-11-2014", format="%d-%m-%Y")
```

```
[1] "2014-11-01"
```

```
> # 특정날짜를 기준으로 며칠 후를 알고싶을때 origin
```

```
> as.Date(10,origin="2014-11-27") <-지정날짜의 10일 후
```

```
[1] "2014-12-07"
```

```
> as.Date(-10,origin="2014-11-27") <-지정날짜의 10일 전
```

```
[1] "2014-11-17"
```

- **as.POSIXct()** : 시간단위까지 연산이 필요할때 사용 cf) as.Date()함수는 날짜까지만 연산가능

- **POSIXlt** : 날짜를 년,월,일로 표시하는 리스트형 클래스

- **POSIXct** : 날짜를 연속적인 데이터로 인식해서 1970년을 기준으로 초단위계산

- 날짜 - 날짜 = 일수

- 날짜 + 숫자 = 날짜 / 날짜 - 숫자 = 날짜

- **as.numeric()** : 문자를 숫자로 바꿀때 사용

^ 접기

> 날짜 관련 연산

> "2014-11-27" - "2014-11-01"

다음에 오류가 있습니다"2014-11-27" - "2014-11-01" :

이항연산자에 수치가 아닌 인수입니다

> " "쌍따옴표로 묶은 날짜는 문자로 인식

> class("2014-11-27")

[1] "character"

> as.Date() 함수를 사용하여 날짜 연산

> as.Date("2014-11-27") - as.Date("2014-11-01")

Time difference of 26 days

> as.Date("2014-11-27") + 5

[1] "2014-12-02"

> as.Date()함수는 날짜단위까지만 연산

> as.Date("2014-11-27 20:00:00") - as.Date("2014-11-27 18:30")

Time difference of 0 days

> as.POSIXct()함수는 시간단위까지 연산 가능

> as.POSIXct("2014-11-27 20:00:00") - as.POSIXct("2014-11-27 18:30")

Time difference of 1.5 hours

※ lubridate 패키지

^ 접기



> install.packages("lubridate")

Installing package into 'C:/Users/stu/Documents/R/win-library/3.1'

(as 'lib' is unspecified)

also installing the dependency 'memoise'

URL 'http://cran.nexr.com/bin/windows/contrib/3.1/memoise_0.2.1.zip'을 시도합니다

Content type 'application/zip' length 16862 bytes (16 Kb)

URL을 열었습니다

downloaded 16 Kb

URL 'http://cran.nexr.com/bin/windows/contrib/3.1/lubridate_1.3.3.zip'을 시도합니다

Content type 'application/zip' length 665638 bytes (650 Kb)

URL을 열었습니다

downloaded 650 Kb

패키지 'memoise'를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다

패키지 'lubridate'를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다

다운로드된 바이너리 패키지들은 다음의 위치에 있습니다

C:\Users\stu\AppData\Local\Temp\Rtmp8uOhOR\downloaded_packages

>

> library(lubridate)

^ 접기

> # 현재 날짜와 시간 넣기

> date <- now()

> date

[1] "2014-11-27 16:05:51 KST"

> # 년도만 출력하기

> year(date)

[1] 2014

> # 월을 출력하되 영문이름으로 출력

> month(date,label=T)

```
[1] Nov
```

```
Levels: Jan < Feb < Mar < Apr < May < Jun < Jul < Aug < Sep < Oct < Nov < Dec
```

```
> # 월을 출력하되 숫자로 출력
```

```
> month(date,label=F)
```

```
[1] 11
```

```
> # 요일을 출력하되 영문이름으로 출력
```

```
> wday(date,label=T)
```

```
[1] Thurs
```

```
Levels: Sun < Mon < Tues < Wed < Thurs < Fri < Sat
```

```
> # 요일을 출력하되 가중치 숫자로 출력하기(1부터 일요일)
```

```
> wday(date,label=F)
```

```
[1] 5
```

```
> # 2일전 날짜 출력하기
```

```
> date <-date-days(2)
```

```
> date
```

```
[1] "2014-11-25 16:05:51 KST"
```

```
> # 2개월뒤 날짜 출력하기
```

```
> month(date) <-2
```

```
> date
```

```
[1] "2014-02-25 16:05:51 KST"
```

```
> # 시간 분 지정하기
```

```
> date <- hm("22:30")
```

```
> date
```

```
[1] "22H 30M 0S"
```

```
> # 시간 분 초 지정하기
```

```
> date <-hms("22:30:15")
```

```
> date
```

```
[1] "22H 30M 15S"
```

① 변수설정시 주의사항

- 대소문자를 구분함
- 영어와 숫자모두 사용가능, 그러나 시작은 반드시 문자로
- 예약어는 사용불가 (break else FALSE for function if in Inf NA NaN next NULL repeat TRUE while 등)

② 변수에 데이터 담기

- R은 다른언어와는 다르게 변수를 선언할 때 어떤데이터를 담을 것인지 미리 지정하지 않고 바로 사용 가능
- 변수에 값을 넣을때는 <- 기호 사용
- 문자와 문자열을 구분하지 않고 모두 문자열로 생각하고 저장함
- 하나의 변수에 여러개의 데이터를 담을 때 데이터형은 동일해야 함

^ 접기

> # 변수에 데이터 담기

> var1 <- "aaa"

> var2 <- 111

> var3 <- Sys.Date()

> var4 <- c("a","b","c")

> # 출력

> var1

[1] "aaa"

> var2

[1] 111

> var3

[1] "2014-11-27"

> var4

[1] "a" "b" "c"

> # 같은 값을 연속적으로 할당 가능

> 111->var5->var6

> var5

[1] 111

> var6

```
[1] 111
```

```
> var8<-var9<-222
```

```
> var8
```

```
[1] 222
```

```
> var9
```

```
[1] 222
```

```
> # 문자와 문자열을 구분하지 않음
```

```
> string1 <- "Hello, R World"
```

```
> string1
```

```
[1] "Hello, R World"
```

```
> # 하나의 변수에 여러개의 데이터를 담을때는 데이터형이 동일해야함
```

```
> comp <- c(1,2)
```

```
> comp
```

```
[1] 1 2
```

```
> class(comp)
```

```
[1] "numeric"
```

```
> # 숫자와 문자가 동시에 담는다면 숫자가 자동으로 문자로 변경됨
```

```
> comp2 <-c(1,"4")
```

```
> comp2
```

```
[1] "1" "4"
```

```
> class(comp2)
```

```
[1] "character"
```

③ 변수값을 사용하여 산술연산하기

- 변수에 데이터들을 담아서 변수끼리 연산가능
- 한개라도 숫자데이터가 아닐경우 산술연산 불가

^ 접기



```
> # 숫자와 숫자의 연산
```

```
> num1 <-1
> num2 <-2
> num1 + num2
[1] 3
```

> # 숫자와 데이터의 연산 -> 오류발생

```
> num1 <-1
> char1 <- "a"
> num1 + char1
```

다음에 오류가 있습니다 num1 + char1 : 이항연산자에 수치가 아닌 인수입니다

④ 변수에 연속적인 값 대입하기

- 연속적인 숫자만 대입가능
- 문자는 연속적으로 할당 불가

^ 접기

```
> seq1 <- 1:5
> seq1
[1] 1 2 3 4 5
```

```
> # 문자는 연속적으로 할당되지 않음
> seq2 <- "a":"f"
```

다음에 오류가 있습니다 "a":"f" : NA/NaN 인자입니다

추가정보: 경고메시지:

- 1: 강제형변환에 의해 생성된 NA 입니다
- 2: 강제형변환에 의해 생성된 NA 입니다

⑤ 생성한 변수 확인 및 삭제

- **objects()** : 자신이 생성한 모든 변수 확인
- **objects(all.names=T)** : 숨김 속성의 변수까지 모두 확인가능
- **rm(변수명)** : 특정변수 삭제가능
- **rm(list=ls())** : 모든 변수 삭제 (조심해서사용)

^ 접기

> # 생성한 변수 모두 확인하기

> objects()

```
[1] "char1" "comp" "comp2" "date" "num1" "num2" "seq1" "string1"
[9] "var1" "var2" "var3" "var4" "var5" "var6" "var8" "var9"
```

> # 숨김변수까지 모두 확인하기

> objects(all.names=T)

```
[1] ".Random.seed" "char1" "comp" "comp2" "date"
[6] "num1" "num2" "seq1" "string1" "var1"
[11] "var2" "var3" "var4" "var5" "var6"
[16] "var8" "var9"
```

> # 변수 제거하기

> string1

```
[1] "Hello, R World"
```

> rm(string1)

> string1

에러: 객체 'string1'를 찾을 수 없습니다

> objects()

```
[1] "char1" "comp" "comp2" "date" "num1" "num2" "seq1" "var1" "var2" "var3"
[11] "var4" "var5" "var6" "var8" "var9"
```

> # 모든 변수 삭제

> rm(list=ls())

> objects()

```
character(0)
```

공감

구독하기

