

Big Data



About..

컴퓨터소프트웨어공학과
김 원 일



목차



- R 자료형
- R 변수 연산
- 콘솔 활용
- R 스크립트와 콘솔
- 패키지 설치
- 도움말 사용하기
- 작업공간
- vector in R



- 기본적으로 4가지 자료형을 제공

- 일반적인 프로그래밍 언어와 유사
- 자료형에 대한 변수 형식은 자유롭기 때문에 주의해서 사용해야 함

자료형	사용 예	비고
숫자형	1, 2, -4, 12.8	음/양수의 정수와 실수
문자형	'ycs', "Yuhan"	작은/큰 따옴표로 표시. 연산불가
논리형	TRUE / T, FALSE / F	부울 값
특수 자료형	NULL	정의되지 않은 값
	NA	vector에서 missing value
	NaN	정의 불가능한 값
	Inf, -Inf	양과 음의 무한대



R 자료형 - 2

- 문자열은 `'`, `"` 모두 동일하게 저장
 - 선언되지 않은 변수와 연산은 불가능

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말
[Icons]

> a <- 3
> b <- -2
> a+b
[1] 1
> str <- "char"
> str + a
str + a에서 다음과 같은 에러가 발생했습니다:$
> str1 <- 'string'
> str
[1] "char"
> str1
[1] "string"
> z
에러: 객체 'z'를 찾을 수 없습니다
> |
```

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말
[Icons]

> str
[1] "char"
> str1
[1] "string"
> 0 / 0
[1] NaN
> 3 / 0
[1] Inf
> -3 / 0
[1] -Inf
> z
에러: 객체 'z'를 찾을 수 없습니다
> z + a
에러: 객체 'z'를 찾을 수 없습니다
> |
```



R 변수 연산 - 1

- 값 입력을 통한 변수 선언

- 상수와 상수의 2항 연산
- 변수와 상수의 2항 연산
- 변수와 변수의 2항 연산

- 변수 값 출력

- "변수명 [ENTER]"
- print(변수명) [ENTER]

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말

> a <- 65535
> a
[1] 65535
> print(a)
[1] 65535
> a + 35
[1] 65570
> a * 89
[1] 5832615
> a / 16
[1] 4095.938
> 1 - 60000
[1] -59999
> a - 60000
[1] 5535
> b <- 4096
> a / b
[1] 15.99976
> a * b
[1] 268431360
> |
```



R 변수 연산 - 2

- 다양한 4칙 연산 수행

- 변수 값 변경 없이 진행
- 변수 간 연산 연습

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지를 윈도우즈 도움말

[1] 65570
> a * 89
[1] 5832615
> a / 16
[1] 4095.938
> 1 - 60000
[1] -59999
> a - 60000
[1] 5535
> b <- 4096
> a / b
[1] 15.99976
> a * b
[1] 268431360
> ls()
[1] "a" "b"
> b
[1] 4096
> print(b)
[1] 4096
> |
```



- 변수 연산을 통한 초기화

- 연산 가능한 변수와 상수 값은 언제나 연산이 가능
- 연산 결과로 변수 초기화 가능

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말

> ls( )
[1] "a" "b" "c"
> c
[1] 77
> a
[1] 65535
> b
[1] 4096
> a + b + c
[1] 69708
> d <- a + b + c
> d
[1] 69708
> a <- a * b
> a
[1] 268431360
> |
```



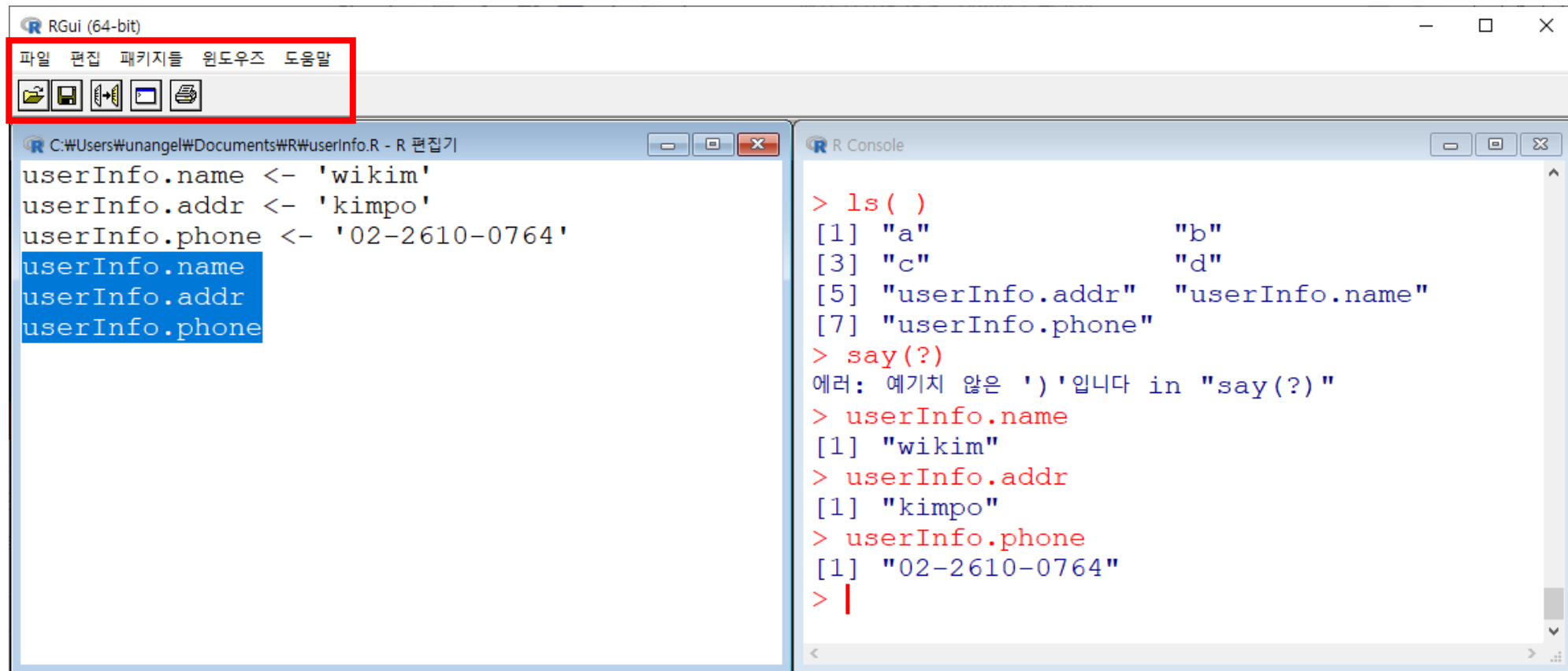
- 다중 연산은 ';' 로 분리
 - 한 문장으로 출력되는 모든 문장은 ';'로 분리하여 적용이 가능
 - 콘솔에서 #으로 시작하면 주석 처리되어 출력 없음

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지를 윈도우즈 도움말

> print( a + b ); print( b + c ); print( c + d )
[1] 268435456
[1] 4173
[1] 69785
> a
[1] 268431360
> b
[1] 4096
> c
[1] 77
> d
[1] 69708
> b %% c
[1] 15
> # b * c * d * a
> |
```



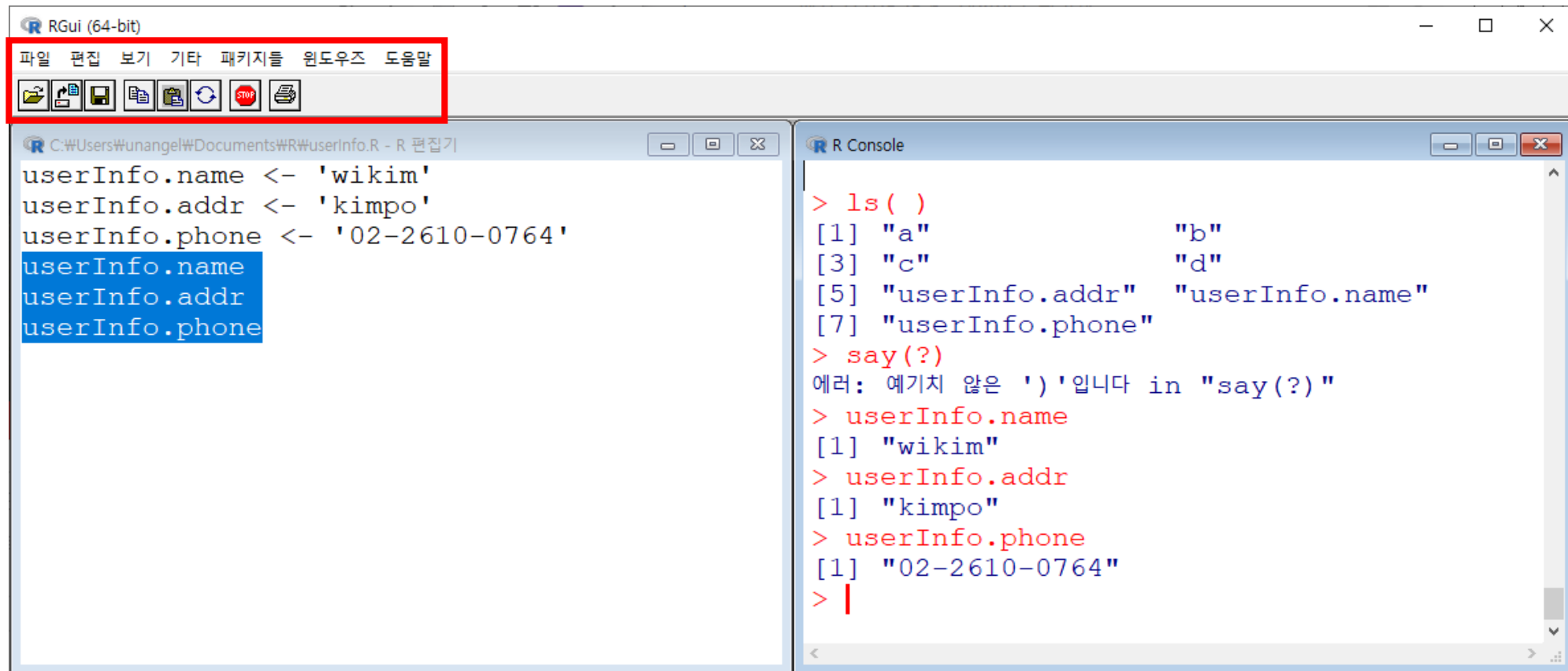

- 스크립트 파일과 콘솔 동시 사용이 기본
 - 스크립트는 일련의 R 스크립트를 파일로 기록





R 스크립트와 콘솔 - 2

- 콘솔과 스크립트 파일 선택에 따라 메뉴가 변경
 - 상단 아이콘들도 내용이 달라지는 것을 확인할 수 있음





- 콘솔에서 수행한 결과를 스크립트로 옮겨 보관하는 형식
 - 스크립트를 저장한 다음 콘솔에서 실행하여 지속적인 개발 수행

```
RGui (64-bit)
파일 편집 패키지 윈도우즈 도움말

C:\Users\Wunangel\Documents\WR\userInfo.R - R 편집기
userInfo.name <- 'wikim'
userInfo.addr <- 'kimpo'
userInfo.phone <- '02-2610-0764'
userInfo.name
userInfo.addr
userInfo.phone

R Console
[5] "userInfo.addr" "userInfo.name"
[7] "userInfo.phone"
> say(?)
에러: 예기치 않은 ')'입니다 in "say(?)"
> userInfo.name
[1] "wikim"
> userInfo.addr
[1] "kimpo"
> userInfo.phone
[1] "02-2610-0764"
> userInfo.phone <- '02-2610-0763'
> userInfo.phone
[1] "02-2610-0763"
> |
```



- 스크립트와 콘솔은 서로 통신하며 수행
 - 스크립트에서 블록 한 내용은 "Ctrl + Enter"로 콘솔에서 실행 가능

```
RGui (64-bit)
파일 편집 패키지들 윈도우즈 도움말

C:\Users\... R\UserInfo.R - R 편집기
userInfo.name <- 'wikim'
userInfo.addr <- 'kimpo'
userInfo.phone <- '02-2610-0764'
userInfo.name
userInfo.addr
userInfo.phone

R Console
에러: 예기치 않은 ')'입니다 in "say(?) "
> userInfo.name
[1] "wikim"
> userInfo.addr
[1] "kimpo"
> userInfo.phone
[1] "02-2610-0764"
> userInfo.phone <- '02-2610-0763'
> userInfo.phone
[1] "02-2610-0763"
> userInfo.phone <- '02-2610-0764'
> userInfo.phone
[1] "02-2610-0764"
> |
```



패키지 설치 - 1



- R은 다양한 패키지를 지원

- 필요에 따라 패키지를 설치하여 원하는 동작을 수행할 수 있음
- 통계 및 빅데이터와 관련된 대부분의 패키지는 이미 존재
- 패키지를 설치하고, 사용하는 방법만 익히면 되는 형태
- 패키지 이름만으로는 내용을 확인하기 어려움
- 필수 패키지들은 기본적으로 설치되어 있음
- 패키지 이름으로 설치 가능
- “install.packages(‘패키지명’) [ENTER]” 으로 설치



패키지 설치 - 2

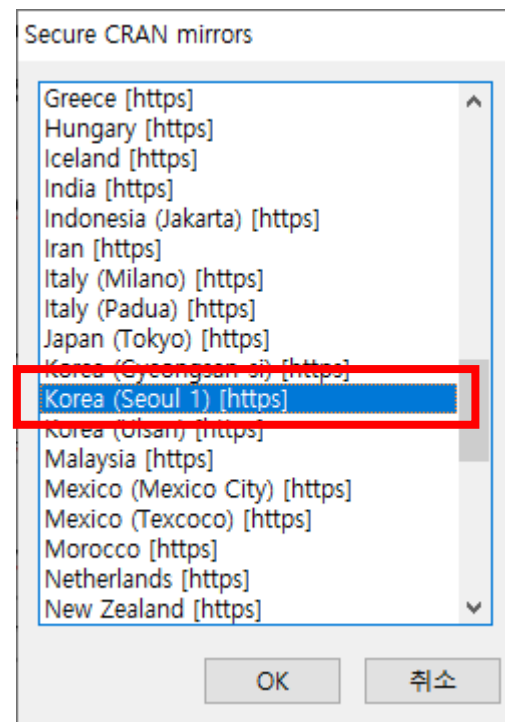
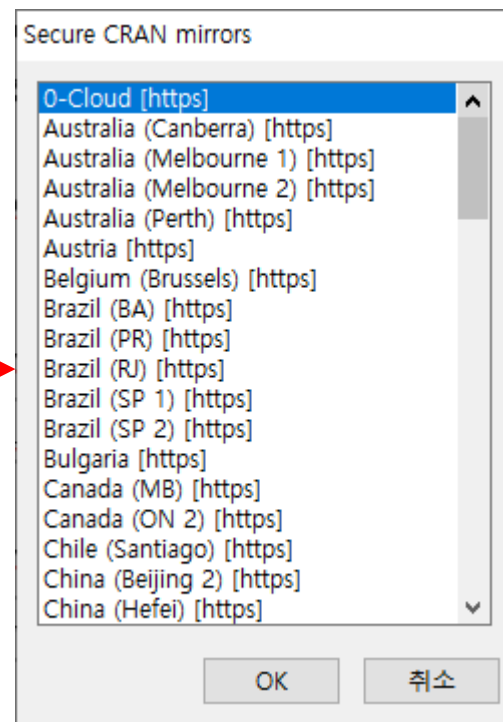
• 패키지 설치 명령어 실행

- 패키지를 다운로드 받을 위치를 설정하면 설치가 진행

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말

> max(a, b, c, d)
[1] 268431360
> min(a, b, c, d)
[1] 77
> abs( log(10 ) + 3 )
[1] 5.302585
> abs( 77 - 187)
[1] 110
> factorial(10)
[1] 3628800
> factorial(3)
[1] 6
> install.packages('cowsay')
```

'C:/Users/unange1/Documents/R/win-library/4.1'의 \$
(왜냐하면 'lib'가 지정되지 않았기 때문입니다)
--- 현재 세션에서 사용할 CRAN 미러를 선택해 주세요 \$





패키지 설치 - 3

- 패키지 설치 중 필요한 추가 패키지도 자동으로 설치 진행

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말

> install.packages('cowsay')
'C:/Users/unangel/Documents/R/win-library/4.1'의 $
(왜냐하면 'lib'가 지정되지 않았기 때문입니다)
--- 현재 세션에서 사용할 CRAN 미러를 선택해 주세요 $
'fortunes', 'rmsfact' (들)을 또한 설치합니다.

URL 'https://cran.seoul.go.kr/bin/windows/contrib/4$
Content type 'application/zip' length 210258 bytes $
downloaded 205 KB

URL 'https://cran.seoul.go.kr/bin/windows/contrib/4$
Content type 'application/zip' length 21452 bytes ($
downloaded 20 KB

URL 'https://cran.seoul.go.kr/bin/windows/contrib/4$
Content type 'application/zip' length 405673 bytes $
downloaded 396 KB

package 'fortunes' successfully unpacked and MD5 $
package 'rmsfact' successfully unpacked and MD5 s$
package 'cowsay' successfully unpacked and MD5 su$

The downloaded binary packages are in
      C:\Users\unangel\AppData\Local\Temp\RtmpS22$
> |
```



패키지 설치 - 4

- **설치된 패키지는 설치만 진행**
 - 실제 사용을 위해서는 로드해야 함
 - “library(패키지명)” 으로 로드
 - 설치한 패키지는 say() 함수를 지원
 - 간단한 이미지 표현 패키지

[illegible]



패키지 설치 - 5

- 새로 콘솔을 연 경우, 기본 패키지만 로드
 - 사용하려는 패키지를 사용 전에 항상 로드하고 실행해야 함

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말
> say('i am tired', by = 'cow')
say("i am tired", by = "cow")에서 다음과 같은$
> library(cowsay)
> say('i am tired', by = 'cow')

-----
i am tired
-----
  \      ^__^
   (oo)\_____)
    (__)  )
       ||----w |
       ||     ||

> |
```

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말
> say('아 춥다....', by = 'snowman')

-----
아 춥다....
-----
  \      ^__^
   (oo)\_____)
    (__)  )
       ||----w |
       ||     ||

> |
```



도움말 사용하기 - 1

- “?키워드” 입력으로 도움말이 브라우저로 자동 연결

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지를 윈도우즈 도움말

say("i am tired", by = "cow")에서 다음과 같은$
> library(cowsay)
> say('i am tired', by = 'cow')

-----
i am tired
-----
      \      ^__^
       (oo)\_____)
            (_____)
                ||----w |
                ||----w |

> ?sin
> |
```

R: Trigonometric Functions

127.0.0.1:28958/library/base/html/Trig.html

Trig {base}

R Documentation

Trigonometric Functions

Description

These functions give the obvious trigonometric functions. They respectively compute the cosine, sine, tangent, arc-cosine, arc-sine, arc-tangent, and the two-argument arc-tangent.

cospi(x), sinpi(x), and tanpi(x), compute cos(pi*x), sin(pi*x), and tan(pi*x).

Usage

```
cos(x)
sin(x)
tan(x)

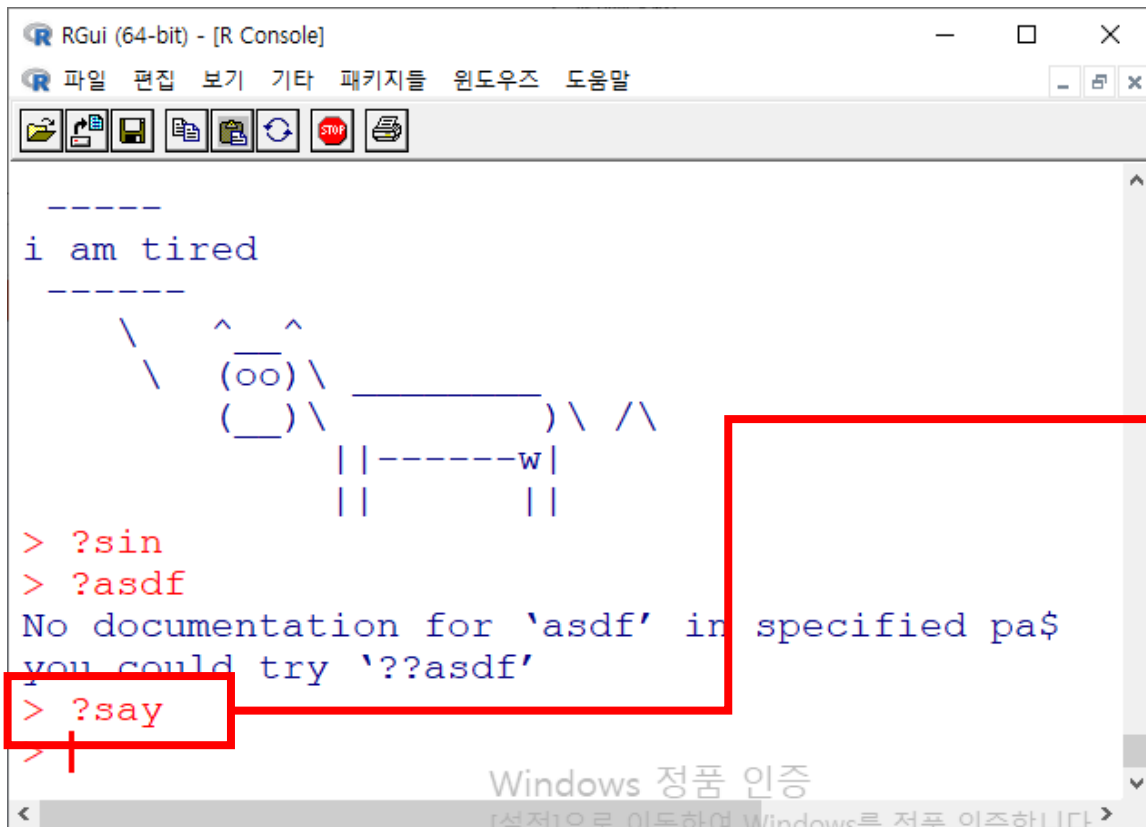
acos(x)
asin(x)
atan(x)
atan2(y, x)

cospi(x)
sinpi(x)
tanpi(x)
```

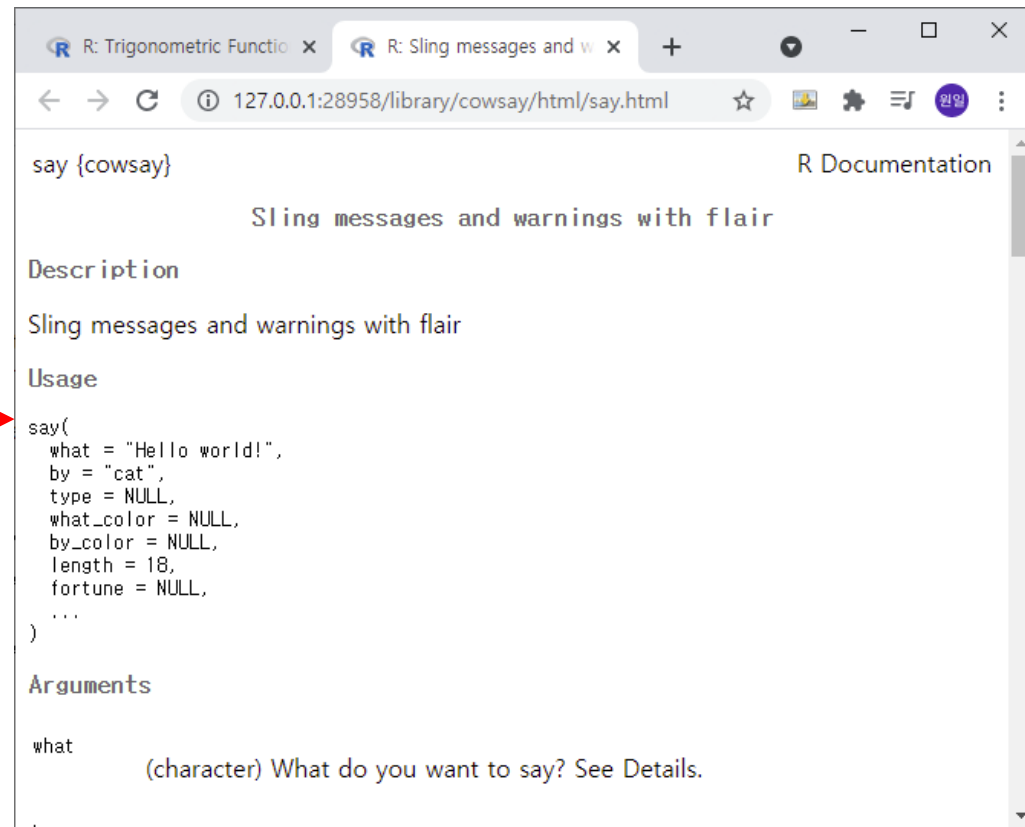


도움말 사용하기 - 2

- 정확한 함수 명이 아니면 도움말을 찾을 수 없음
 - "??키워드"를 이용하여 온라인 검색도 가능



```
-----  
i am tired  
-----  
  \      ^  ^  
   \    (oo)\_____  
      (__)\\       )\/\  
           ||----w |  
           ||     ||  
  
> ?sin  
> ?asdf  
No documentation for 'asdf' in specified pa$  
you could try '??asdf'  
> ?say  
> |
```



```
R: Trigonometric Functio x R: Sling messages and w x +  
127.0.0.1:28958/library/cowsay/html/say.html ☆  
say {cowsay} R Documentation  
Sling messages and warnings with flair  
Description  
Sling messages and warnings with flair  
Usage  
say(  
  what = "Hello world!",  
  by = "cat",  
  type = NULL,  
  what_color = NULL,  
  by_color = NULL,  
  length = 18,  
  fortune = NULL,  
  ...  
)  
Arguments  
what  
  (character) What do you want to say? See Details.
```



작업공간



- **작업 공간**

- 현재 작업 중인 메모리 공간을 작업 공간이라 지칭
- 메모리에 할당된 변수와 변수가 갖는 정보를 그대로 보관
- 일종의 스냅샷과 같은 형태로 작업 공간을 저장
- 현재 작업 공간을 다음 번에 그대로 사용하고자 하는 경우에 저장 후 사용 가능



vector in R – 1



- R에서는 array(배열)와 동일한 의미로 사용
- 일반적인 배열과 동일한 사용과 속성을 가짐
 - 동일한 자료형 다수 개를 동시에 선언
 - 다차원 배열 선언 가능
 - 대표 이름을 이용하여 요소들을 접근하는 방법
 - 일반 배열과 다르게 첨자(인덱스)는 1부터 시작
- 배열 생성 및 접근
 - 배열 대표명 <- c(배열 요소 리스트) : 배열 요소 리스트의 vector 생성
 - 배열 대표명[1] : 첫 번째 배열 요소 접근
 - 배열 대표명[n] : n 번째 배열 요소 접근



vector in R – 2

- vector 숫자 배열 사용 예
 - 1부터 5까지의 숫자 입력
 - vector 요소 전체 출력
 - 요소 별 접근과 연산

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말

> vec <- c( 1, 2, 3, 4, 5 )
> vec
[1] 1 2 3 4 5
> print( vec )
[1] 1 2 3 4 5
> vec[0]
numeric(0)
> vec[1]
[1] 1
> vec[5]
[1] 5
> vec[1] + vec[3]
[1] 4
> |
```

Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증합니다 >



vector in R – 3

- vector 문자열 사용 예
 - 4개의 문자열로 초기화
 - vector 요소 전체 출력
 - 요소 별 접근

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말

> strV <- c( "abc", "asdf", "move", "left" )
> strV
[1] "abc" "asdf" "move" "left"
> print( strV )
[1] "abc" "asdf" "move" "left"
> strV[ 0 ]
character(0)
> strV[ 1 ]
[1] "abc"
> strV[ 3 ]
[1] "move"
> |
```



vector in R – 4

- vector Bool 자료형 사용 예
 - 5개의 참/거짓으로 초기화
 - vector 요소 전체 출력
 - 요소 별 접근과 연산
 - TRUE는 1, FALSE는 0 확인 가능

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말

> vecB <- c( T, F, F, T, T )
> vecB
[1]  TRUE FALSE FALSE  TRUE  TRUE
> print( vecB )
[1]  TRUE FALSE FALSE  TRUE  TRUE
> vecB[ 1 ]
[1] TRUE
> vecB[ 4 ]
[1] TRUE
> vecB[ 1 ] + vecB[ 2 ]
[1] 1
> vecB[ 2 ] + vecB[ 3 ]
[1] 0
> |
```

Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증합니다.



• 다양한 자료형 입력 문제

- 다른 자료형을 표현할 수 있는 대표적인 자료형으로 자동 변환되어 입력
- 하나라도 다른 자료형이 확인되면 자동 변환 수행
- 문자열 포함 시, 모두 문자열
- 숫자 표현 가능 시, 모두 숫자

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말

> vec1 <- c( 1, 2, 3, 'a', 'b', 'c' )
> vec1
[1] "1" "2" "3" "a" "b" "c"
> vec2 <- c( 'a', 1, 2, 3, 4, 5 )
> vec2
[1] "a" "1" "2" "3" "4" "5"
> vec3 <- c( TRUE, 'a', 1, 2, 3 )
> vec3
[1] "TRUE" "a" "1" "2" "3"
> vec4 <- c( F, 1, 2, 3 )
> vec4
[1] 0 1 2 3
> |
```



vector in R – 6

• 연속 숫자 입력 vector 생성 – 1

- ":"을 이용하여 범위 설정
- 지정된 범위 값이 자동으로 입력되며, 순차적으로 1씩 증가
- 메모리가 허용되는 만큼 입력 가능할 것으로 판단

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말

> value1 <- 1:10
> value1
[1] 1 2 3 4 5 6 7 8 9 10
> value2 <- 10:21
> value2
[1] 10 11 12 13 14 15 16 17 18 19 20 21
> value3 <- 100:90
> value3
[1] 100 99 98 97 96 95 94 93 92 91 90
> value4 <- 100:300
> value4
[1] 100 101 102 103 104 105 106 107 108 109
[11] 110 111 112 113 114 115 116 117 118 119
[설정]으로 이동하여 Windows를 정품 인증합니다.
```



vector in R – 7

• 연속 숫자 입력 vector 생성 – 2

- vector 생성에도 ":" 사용 가능
- 유연하게 범위 지정 가능
- 입력 순서대로 데이터 저장

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지를 윈도우즈 도움말

> value5 <- c( 1, 3, 5, 35:77 )
> value5
 [1]  1  3  5 35 36 37 38 39 40 41 42 43 44 45
[15] 46 47 48 49 50 51 52 53 54 55 56 57 58 59
[29] 60 61 62 63 64 65 66 67 68 69 70 71 72 73
[43] 74 75 76 77
> value6 <- c( 1:20, 3:10, 11:25, 77, 78 )
> value6
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14
[15] 15 16 17 18 19 20  3  4  5  6  7  8  9 10
[29] 11 12 13 14 15 16 17 18 19 20 21 22 23 24
[43] 25 77 78
> |
```

Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증합니다.



vector in R – 8

• 연속 숫자 입력 vector 생성 – 3

- seq()를 통해 값 입력 가능
- for()문장과 동일하게 사용
- seq(초기값, 종료값, 증감값)

```
RGU (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말

> value7 <- seq( 1, 100, 3 )
> value7
 [1] 1 4 7 10 13 16 19 22 25 28 31
[12] 34 37 40 43 46 49 52 55 58 61 64
[23] 67 70 73 76 79 82 85 88 91 94 97
[34] 100
> value9 <- seq( 100, 80, -1 )
> value9
 [1] 100 99 98 97 96 95 94 93 92 91 90
[12] 89 88 87 86 85 84 83 82 81 80
> |
```