



---

## 2장. R Language 기초



# 도움말

## 2. R Language 기초

물음표 기호(?) 또는 `help()` 함수를 사용하면 함수, 데이터셋 등에 대한 도움말을 보여줍니다.

```
? 이름
```

```
help("이름")
```

다음 코드는 `iris` 데이터의 도움말을 보여줍니다. RStudio에서 도움말은 `help` 탭에 출력됩니다. `?iris`와 `help(iris)`의 결과는 같습니다.

```
> ?iris  
> help(iris)
```

# 도움말 검색

## 2. R Language 기초

??와 `help.search()`는 함수 등의 이름을 검색할 때 사용합니다.

```
?? 이름
```

```
help.search("이름")
```

`help.search()` 함수의 인자는 반드시 인용부호(따옴표, ‘ ’ 또는 “ ”)로 묶어야 합니다.

```
> ??iris  
> help.search("iris")
```

# 패키지 도움말

## 2. R Language 기초

`library()` 함수에 `help` 파라미터를 이용하면 패키지 도움말과 패키지에서 제공하는 함수 목록을 볼 수 있습니다. 설치되어 있지 않는 패키지는 도움말 기능을 사용할 수 없습니다.

```
library(help="패키지명")
```

다음 구문은 `stats` 패키지의 함수 목록을 출력합니다.

```
> library(help="stats")
```

# 함수 도움말

## 2. R Language 기초

methods()는 이름으로 시작하는 함수의 목록을 조회합니다.

```
methods(이름)
```

아래의 구문은 as로 시작하는 모든 함수 목록을 출력합니다.

```
> methods(as)
[1] as.array          as.array.default
[3] as.call           as.character
[5] as.character.condition as.character.Date
... 생략
```

args()는 함수의 인자(파라미터) 정보를 조회합니다.

```
args(함수명)
```

아래 구문은 data 함수의 파라미터 목록을 출력합니다.

```
> args(data)
function (..., list = character(), package = NULL, lib.loc = NULL,
  verbose = getOption("verbose"), envir = .GlobalEnv)
NULL
```

# 정보 조회

## 2. R Language 기초

`attributes()`는 객체의 속성들을 조회합니다.

```
attributes(이름)
```

`iris` 데이터의 속성들을 출력합니다.

```
> attributes(iris)
$names
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"
[5] "Species"

$row.names
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
[17] 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
... 생략
[129] 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
[145] 145 146 147 148 149 150
```

# 사용 사례 표시

## 2. R Language 기초

examples()는 사용 사례를 표시합니다.

```
example(이름)
```

다음 구문은 mean 함수의 사용 예를 출력합니다.

```
> example(mean)

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

mean> xm <- mean(x)

mean> c(xm, mean(x, trim = 0.10))
[1] 8.75 5.50
```

# 주석과 자동완성

## 2. R Language 기초

#은 주석을 표시할 때 사용합니다. #이후의 내용은 모두 주석으로 처리되어 R 엔진은 실행 시 해당 내용을 모두 무시합니다.

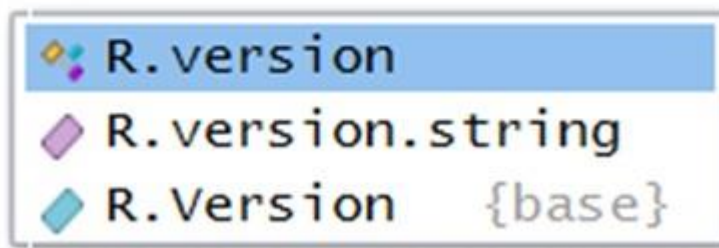
#로 시작하는 라인은 주석입니다.

TAB 키를 이용하면 자동 완성기능을 사용할 수 있습니다.

TAB키

만일 RStudio를 사용한다면 Ctrl+Spacebar를 사용하여 입력 값 제안기능을 사용할 수 있습니다. RStudio의 스크립트 창에서 R.v 까지만 입력하고 컨트롤키와 스페이스바를 같이 누르면(Ctrl+Spacebar) 아래의 그림처럼 R.v로 시작하는 변수, 함수 등의 이름을 보여줍니다.

R.v





# 패키지

## 2. R Language 기초

패키지 설치에 `install.packages()` 함수를 이용합니다. `lib.loc` 파라미터는 패키지가 설치된 경로를 지정합니다.

```
install.packages("패키지명", lib.loc="패키지설치경로")
```

설치된 패키지를 사용하기 위해 `library()` 함수를 사용합니다.

```
library(패키지명)
```

`require()` 함수는 `library()` 함수와 유사합니다. 주로 함수 안에서 사용합니다.

```
require(패키지명)
```

패키지를 언로드하기 위해서 `detach()` 함수를 사용합니다.

```
detach("package:패키지명", unload=TRUE)
```

# 패키지

## 2. R Language 기초

data() 함수는 패키지에 포함된 데이터셋을 메모리로 로딩합니다. R에 기본으로 포함된 데이터는 datasets 패키지에 있고, datasets 패키지에 있는 데이터셋은 data(데이터셋명) 형태로 로딩할 수 있습니다. 패키지가 로딩되어 있으면 data(데이터셋명) 형태로 간단하게 사용할 수 있습니다.

```
data(list="데이터셋명", package="패키지명")
```

다음 코드는 datasets 패키지를 언로드 한 후 iris 데이터를 메모리에 로드할 때 데이터셋을 찾을 수 없다는 에러메시지를 보이고 있습니다.

```
> detach("package:datasets", unload=TRUE)
> data(iris)
Warning message:
In data(iris) : 데이터셋 'iris'을 찾을 수 없습니다
```

패키지를 로드하면 정상적으로 데이터셋이 로드되는 것을 확인할 수 있습니다.

```
> library("datasets", lib.loc="C:/Program Files/R/R-3.4.1/library")
> data(iris)
```

## # 패키지

```
# 1. 라이브러리 설치 : install.packages("패키지명")
# 2. 라이브러리를 메모리에 로딩 : library("패키지명") require("패키지명")
library(MASS)
head(MASS::Cars93)
head(Cars93)
# 3. 설치된 라이브러리 확인 : installed.packages()
# 4. 메모리에 로딩된 패키지 : search()
# 5. 메모리에 로딩된 패키지 메모리에서 내리기
detach("package:MASS", unload=TRUE);
head(Cars93)
```



# 변수

## 2. R Language 기초

R에서 변수를 만들 때 몇 가지 규칙이 있습니다.

- 변수명은 알파벳, 숫자, 밑줄 문자(underscore, '\_'), 점('.')으로 구성됩니다.
- 첫 글자는 알파벳 또는 점('.')으로 시작합니다.
- 점('.')으로 시작 시 바로 뒤에는 숫자가 올 수 없으며, 히든(hidden, 숨겨진)변수가 됩니다.

R 1.9.0 이전에는 밑줄 문자(underscore, '\_')가 변수명에 사용될 수 없었습니다. 이런 이유로 다른 언어에서 흔히 밑줄 문자(underscore, '\_')를 사용할만한 상황에서 R은 점('.')을 사용하곤 합니다. 예를 들어 `training_data`, `validation_data` 같은 변수명 대신 `data.training`, `data.validation` 과 같이 마치 객체의 속성을 접근하는 것처럼 명명 규칙을 사용하곤 합니다.

R에서 변수 선언을 위한 데이터타입 키워드를 지정하지 않습니다. C언어 또는 Java에서 존재하는 `int`, `long`, `float`, `double`, `boolean` 등의 키워드는 존재하지 않습니다.

# 변수에 데이터 할당

## 2. R Language 기초

R에서 변수에 데이터 할당하기 위한 연산자는 `<-`, `->`, `=`, `<<-`, `->>`가 있습니다.

- 대부분의 경우 할당은 '`<-`'를 이용합니다. '`->`'는 왼쪽의 값을 오른쪽 변수에 할당합니다.
- '`=`'은 주로 함수 파라미터의 값을 지정할 때 사용합니다. 예를 들면 `plot(cars, col="red")`처럼 `col` 파라미터 값에 "red"를 할당할 때 `=(equal)`을 사용합니다.
- '`<<-`'와 '`->>`'는 함수 안에서 전역변수에 값을 할당할 때 사용합니다.
- `(varA <- varB)`처럼 할당문 앞/뒤를 ( )로 묶을 경우, 할당과 동시에 할당된 데이터를 화면에 표시합니다.

# 변수 목록 출력

## 2. R Language 기초

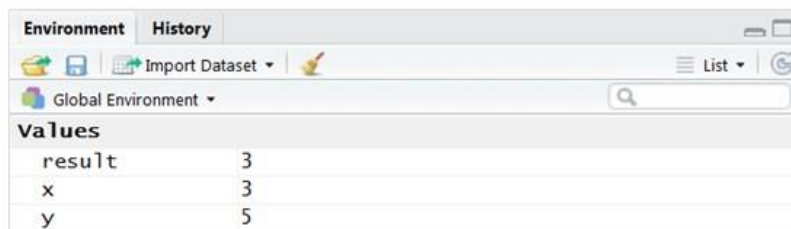
ls()는 현재 메모리에 있는 변수의 목록을 출력합니다. all.names=TRUE 일 경우 히든변수도 출력해 줍니다.

```
ls(all.names=FALSE)
```

ls.str() 함수를 이용하면 변수의 이름과 구조를 함께 보여줍니다.

```
> x <- 3
> y <- 5
> (z <- x+y)
[1] 8
> ls()
[1] "add"      "result" "x"      "y"      "z"
> ls.str()
add : function (a, b)
result : num 3
x : num 3
y : num 5
z : num 8
```

RStudio의 Environment 탭에서 변수 목록을 확인할 수 있습니다.



Environment		History
Global Environment		
Values		
result	3	
x	3	
y	5	

# 변수 출력하기

## 2. R Language 기초

다음 구문은 변수의 값을 출력하는 예입니다.

```
> result  
[1] 3  
> print(result)  
[1] 3  
> (z <- x+y)  
[1] 8
```

cat() 함수는 여러 개의 항목을 묶어서 출력해 줍니다. 항목과 항목 사이에 공백이 추가되며, 행렬이나 리스트 등 복합 데이터 구조 출력 못합니다.

다음 구문은 cat을 이용하여 여러 개 항목을 묶어 출력하는 예입니다.

```
> fib <- c(0,1,1,2,3,5,8,13,21,34)  
> cat("피보나치 수열 몇 개 :", fib, "...\\n")  
피보나치 수열 몇 개 : 0 1 1 2 3 5 8 13 21 34 ...
```

여러 문자열을 이어 출력하고 싶다면 paste() 함수를 사용하여 문자열을 이어줄 수 있습니다. paste0() 함수는 문자열을 이어줄 때 구분자 없이 이어줍니다.

```
paste(..., sep=" ", collapse=NULL)
```

```
paste0(..., collapse=NULL)
```

# paste 함수의 속성

## 2. R Language 기초

paste() 함수의 collapse 속성은 연결되는 데이터셋의 구분자입니다. collapse 파라미터를 이용하면 각 항목들의 구분자를 지정한 단일 문자열로 출력합니다.

```
> month.name
[1] "January" "February" "March" "April" "May" "June"
[7] "July" "August" "September" "October" "November" "December"

> (nth <- paste0(1:12, c("st", "nd", "rd", rep("th", 9))))
[1] "1st" "2nd" "3rd" "4th" "5th" "6th" "7th" "8th" "9th" "10th" "11th"
[12] "12th"

> paste(month.name, nth, sep=": ", collapse="; ")
[1] "January: 1st; February: 2nd; March: 3rd; April: 4th; May: 5th; June: 6th;
July: 7th; August: 8th; September: 9th; October: 10th; November: 11th; December:
12th"

> paste(month.name, nth, sep=": ")
[1] "January: 1st" "February: 2nd" "March: 3rd" "April: 4th"
[5] "May: 5th" "June: 6th" "July: 7th" "August: 8th"
[9] "September: 9th" "October: 10th" "November: 11th" "December: 12th"
```



# 변수 삭제

## 2. R Language 기초

```
rm(변수명)
```

다음 구문은 변수 `z`를 삭제합니다.

```
> rm(z)
> ls()
[1] "add"      "c"        "result"   "x"        "y"
```

다음 구문은 `ls()`함수의 결과를 이용하여 메모리에 있는 모든 변수를 삭제합니다. 이 경우 히든 변수는 삭제되지 않습니다.

```
> rm(list=ls())
> ls()
character(0)
```

히든변수까지 삭제하려면 `rm(list=ls(all.names=TRUE))` 구문을 이용합니다.

```
> .hiddenVar=10
> .hiddenVar
[1] 10
> rm(list=ls(all.names=TRUE))
> .hiddenVar
Error: object '.hiddenVar' not found
```

# R 확장자

## 2. R Language 기초

파일	형식	설명
R 스크립트	~.R	실행할 R 코드를 저장한 파일 rscript ~.R # Linux 터미널에서 실행 source(~.R, echo=FALSE) # Console에서 실행
R 작업공간	.RData ~.RData	Console에서 실행하여 생성한 R 객체(데이터셋) 저장 load("~.RData") # Rdata 파일 로딩 save(var1, file=~.RData) # var1 객체를 저장 save(var1, var2, file=~.RData) save.image() # .RData에 작업 공간 저장 save.image("~.RData") # 작업 공간 전체 저장 unlink("~.RData") # ~.RData 파일 삭제
R 작업기록	.Rhistory	Console에서 실행한 R 명령어 저장 savehistory() # .Rhistory 저장 loadhistory() # .Rhistory 로딩

# R Language 기초 실습

## 2. R Language 기초

### 1. 도움말 기능을 사용해 봅니다.

- 1) `stat` 패키지의 도움말을 사용합니다.
- 2) `mean` 함수의 도움말을 사용합니다.
- 3) `iris` 데이터셋의 도움말을 사용합니다.

### 2. 아래 패키지를 설치합니다.

- 1) `data.table`
- 2) `plyr`, `dplyr`
- 3) `reshape`, `reshape2`
- 4) `ggplot2`