Package 설치하기

▣ 패키지의 개념

R에서는 데이터 분석을 위해 다양한 함수를 제공한다. **패키지**(package)는 이러한 함수들을 기능별로 묶어 놓은 일종의 꾸러미이다.

▣ 패키지의 설치[1)]

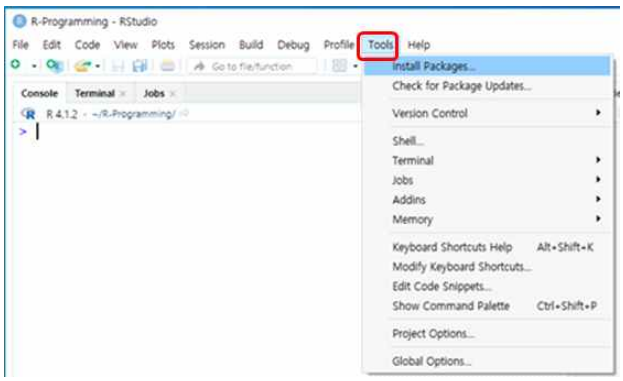
R에서는 어떤 함수를 이용하려면 원칙적으로 그 함수가 속해 있는 패키지를 불러와야 한다. 지금부터 **패키지를 설치**하고 프로그램에서 사용할 수 있도록 **불러오는(loading) 방법**에 대해서 알아보자.

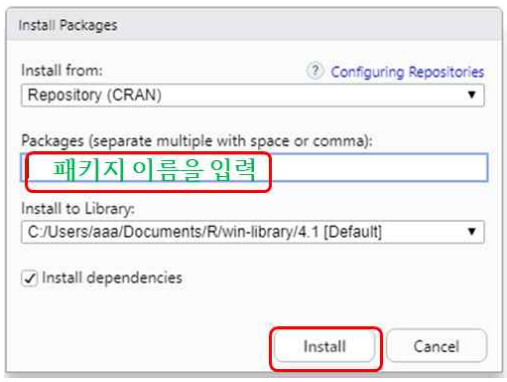
(1) 명령문으로 설치하기

install.packages("패키지명")

- 설치할 때 패키지 명에 큰따음표가 필요하다.
- package를 설치하기 위해서는 인터넷 연결이 필요하며, 컴퓨터에서 package 설치는 한 번만 하면 되지만 패키지 불러오기는 RStudio가 새로 시작할 때마다 필요하다.

(2) RStudio 메뉴로 설치하기





- 메뉴 상단의

Tools > Install Packages…

를 클릭하면 Install Packages 팝업 메뉴가 나오고 원하는 패키지명을 입력하면 된다.

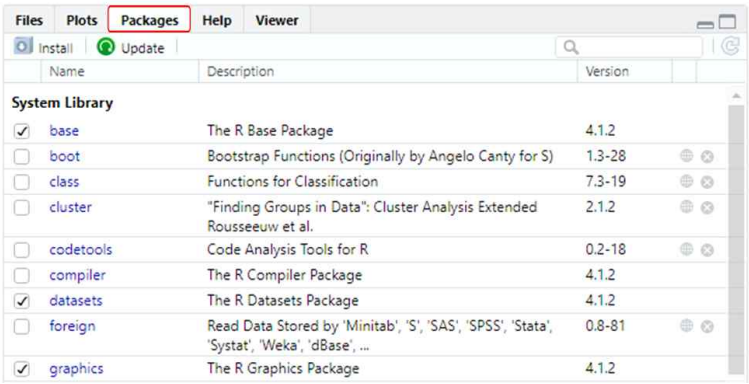
---

1) 패키지의 공유는 CRAN(The Comprehensive R Archive Network) 을 통해 공유되고 있고, 현존하는 패키지만 2만개 이상 있다. 어떤 패키지들이 CRAN에 등재되어 있는지 확인하고 싶다면 available.packages( ) 함수를 통해 확인이 가능하다.

• 패키지 이름의 일부만 입력해도 설치 가능한 패키지 목록이 나타난다.

그러면 패키지 설치 과정이 콘솔 창에 표시된 후 설치 완료 메시지가 나타난다.

• 설치된 패키지 확인하기 **패키지 창**에 하단을 보면 컴퓨터에 설치된 패키지 목록을 확인할 수 있다. 방금 설치한 패키지 외에도 다른 패키지를 보이는데 이것은 R을 설치할 때 자동으로 설치된 패키지들이다.



▣ 패키지 불러오기

라이브러리는 도서관이라고 생각하면 되고, 패키지는 도서관에서 대여하는 책이라고 생각할 수 있다. 즉, 책(package)이 필요하면 library( ) 함수로 불러와서 데이터 분석과 작업에 사용한다. 참고로 R의 내장데이터들과 달리 패키지의 데이터들은 Load를 해주어야 사용할 수 있다.

```
library(package)
```

[예제1]

```
# "MASS" 패키지 설치하기
install.packages("MASS")

# "MASS" 패키지 불러오기
library(MASS)
# "MASS" 패키지 활용하기 : 20번째 데이터인 "Cars93"을 불러오는 방법
data.frame(Cars93, package="MASS")
```

[예제2]

```
# "prob" 패키지 설치하기
```

```
install.packages("prob", repos="http://R-Forge.R-project.org")
```

"prob" 패키지 불러오기

```
library(prob)
```

"prob" 패키지 활용하기 : tosscoin( )을 이용하여 표본공간을 생성

```
tosscoin(3)
tosscoin(3, makespace=T)
```

rolldie( )를 이용하여, 표본공간 S를 구하자.

```
rolldie(2)
rolldie(2, makespace=T)
```

▣ 패키지에 포함된 함수 목록 보는 방법

패키지에서 제공하는 모든 함수 목록을 볼 수는 없을까?

예를들어 지금 설치한 MASS 패키지에서 제공하는 모든 함수를 보고 싶다면 아래와 같은 명령어를 입력한다. (명령어 형태로 출력한다.)

```
ls("package:MASS")
ls("package:base")
```

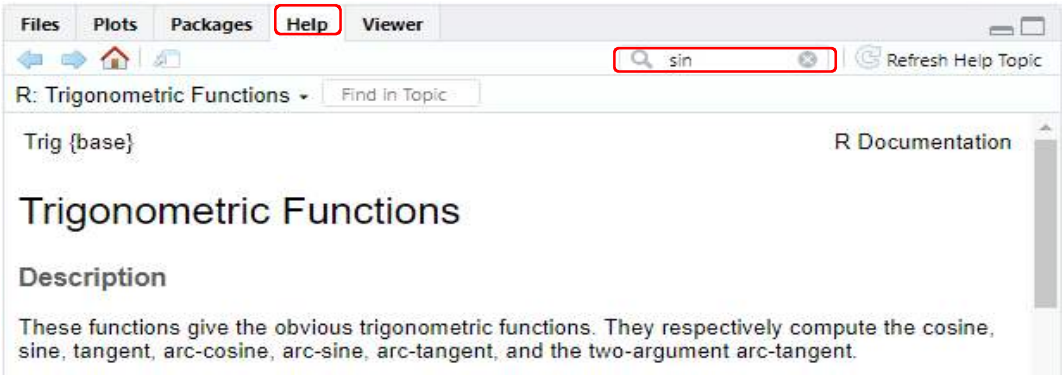
참고. 패키지에서 제공하는 함수의 개수를 알고 싶다면...

```
length(ls("package:MASS"))
length(ls("package:base"))
```

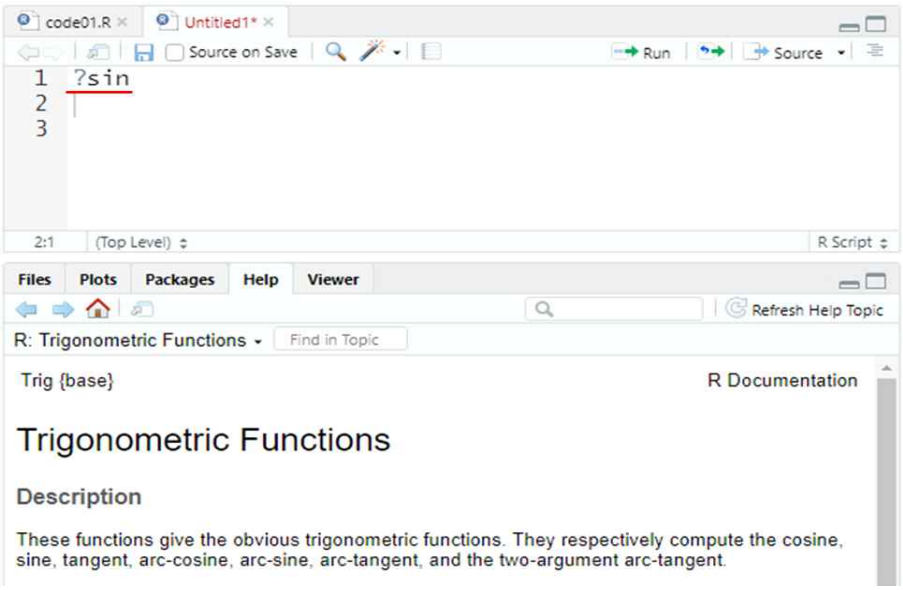
도움말 사용방법

▣ 도움말 사용 방법

[방법1] RStudio에서 Help(도움말) 창에서 검색하고자 하는 함수명을 입력한다.



[방법2] Source 창에서 함수 이름 앞에 물음표를 입력하고 실행한다.



[방법3] Source 창에서 help( ) 함수의 괄호 안에 함수인 이름을 입력하고 실행한다.

▣ 도움말의 구성 항목

도움말은 일반적으로 다음과 같이 구성되어 있다.

구분	설명
설명(Description)	함수에 대한 간단한 소개
사용법(Usage)	함수 호출 방법
인수(Arguments)	입력값(매개변수)들에 대한 설명
세부사항(Details)	구체적인 적용 방식과 주의사항
값(Value)	함수 실행 시 반환되는 결과값
참조(See Also)	관련된 다른 R 함수
예제(Examples)	실제로 적용되는 예제 코드

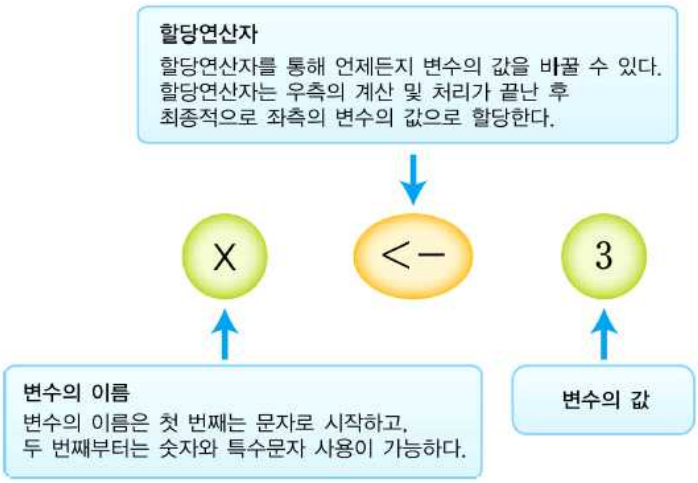
- 함수에 어떤 값을 입력해야 하는지 알려주는 '인수(**Arguments**)',
- 함수 실행 시 어떤 값이 반환되는지 알려주는 '값(**Value**)
- 실제로 작동하는 예제 코드가 제시된 '예제(**Examples**)'

변수

일반적으로 프로그래밍에서는 값들을 저장하기 위하여 다양한 변수(variables)들이 필요하다. 변수는 값을 저장한 메모리의 위치를 나타낸다. 즉, 변수는 프로그램 내에서 어떤 값을 저장해 놓을 수 있는 '**보관 상자**'의 역할을 한다.

(1) 변수 만들기

- x 라는 변수에 3이라는 값을 저장하고 싶다면 그림과 같은 명령문 실행.
- 3을 x 로 보내라는 모양을 하고 있는데, 이것이 '**3을 x 에 저장하라**'는 명령임.
- x 는 **변수명**, 3은 변수에 저장할 값, <- 는 값을 변수에 저장하는 **할당연산자**.



- 변수에 값을 할당[2]할 때는 <-, <<- 또는 = 연산자를 사용한다.

(2) 변수명 규칙

변수의 이름은 사용자가 임의로 만들어 사용할 수 있지만 다음과 같은 규칙을 지켜야 한다.

① 문자와 숫자 모두 사용 가능

② 영어의 대, 소문자를 구분하여 A와 a는 다른 변수로 취급한다.

③ 한글 문자사용이 가능

④ 특수문자는 '_'(언더스코어)와 '.'(마침표)만 사용 가능. -(하이픈)은 사용할 수 없다.

⑤ 첫 글자는 알파벳 또는 . 으로 시작하며, 숫자로 시작할 수 없다.

⑥ 특수문자(_, .)로 시작하는 경우 바로 숫자가 올 수 없음

---

2) 변수를 생성하고 값을 넣는 것을 변수 할당이라고 한다.

⑦ 변수명 중간에 빈칸(공백)을 사용할 수 없다.

올바른 변수명	올바르지 않은 변수명
a	
b	2a
a1	.2
a2	a-b
.x	

[예제1]

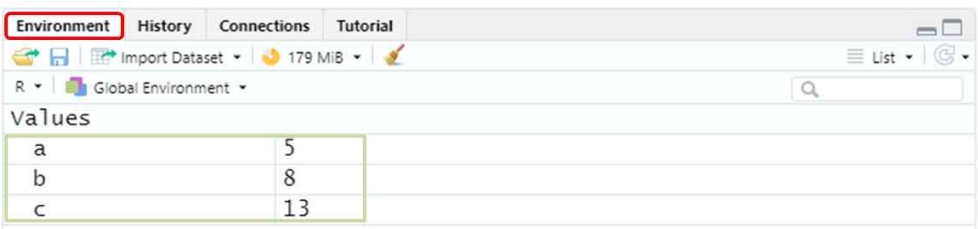
```
total <- 234  # 234을 변수 total에 저장
total = 234
total <<- 234
total   # 방법 1
print(total)    # 방법 2, print( ) 함수 사용
( total <- 234 )  # 방법 3
```

(3) RStudio 에서 변수의 내용을 확인하는 방법

R 스튜디오에서 **환경창**(Environment Pane)을 보면 현재 만들어진 변수와 저장된 값을 확인할 수 있다.

[예제2]

```
a <- 5        # 변수 a에 5를 저장하는 명령문
b <- 8        # 변수 b에 8을 저장
c <- a + b    # 변수 a, b에 저장된 값을 더하여 변수 c에 저장
print(a)      # 변수 a의 내용 출력
print(b)      # 변수 b의 내용 출력
print(c)      # 변수 c의 내용 출력
```



[예제3]

```
a <- 5
print(a)
a <- 10     # 변수의 값은 바뀔 수 있다
print(a)
```

하나의 변수에는 하나의 값만 저장 가능

- 다음과 같이 number 라는 변수에 두 개의 값 3, 7를 저장하려고 하면 에러 발생.
- R에서는 여러 개의 값을 한 곳에 저장하는 방법으로 **벡터**(vector)를 제공.

```
number <- 3, 7      # 에러 발생
```