
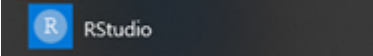


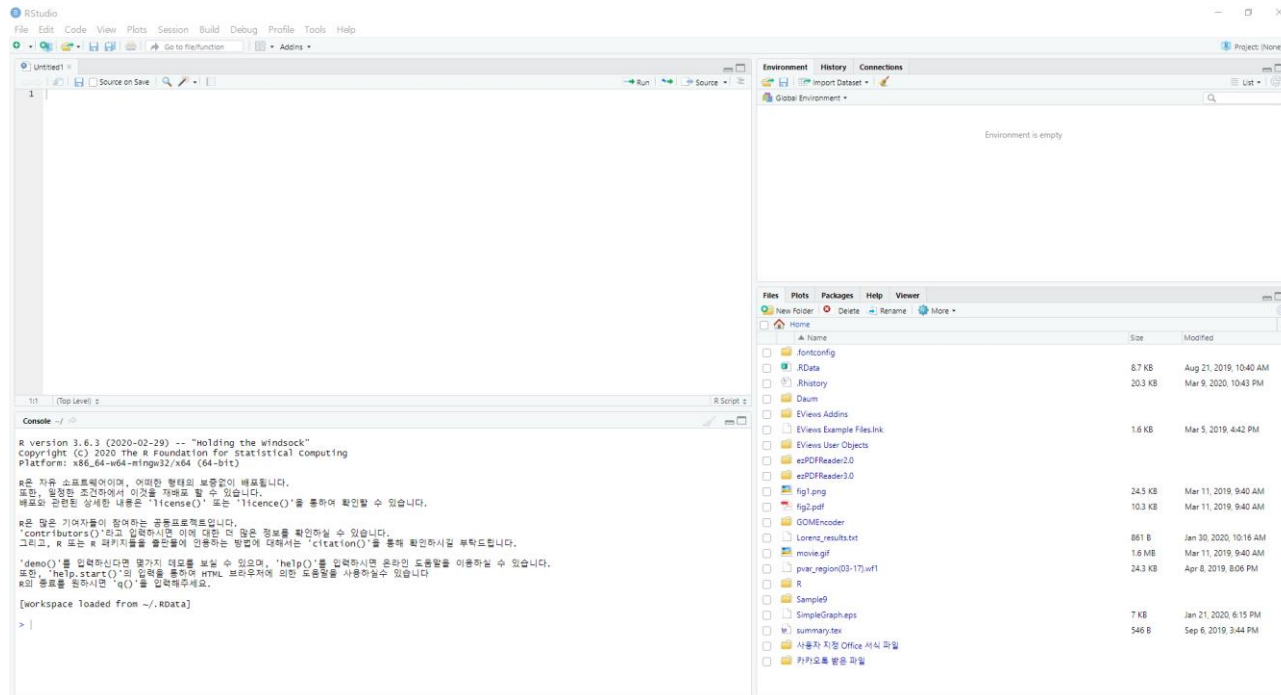
IV. RStudio 시작하기

V. 명령어 실행방법

- 바탕화면에 있는 RStudio 아이콘을 클릭하거나 바탕화면의 좌측 하단에 있는 시작  을 클릭하여 R 아래에 있는  를 클릭



- Sources 창, Console 창, Environment/History 창, Files, Packages/Plots, Help, Viewer 창이 나타남
- Source 창 : 프로그램 Source를 편집 / 프로그램 내의 R 명령어에 커서를 두고 Ctrl+Enter로 실행
- Console 창 : 명령어를 입력하여 실행하고 결과를 확인
 - 상하 화살표를 이용하여 이전 명령어를 편집 및 실행
- Environment/History 창 :
 - Environment 창 : 변수 또는 객체의 목록과 값 확인
 - History 창 : 명령어 History를 확인 및 검색하고 더블 클릭하여 Console 창으로 보냄
- Files, Packages/Plots, Help, Viewer 창
 - Files, Packages 창 : 파일과 폴더 및 패키지 목록
 - Plots, Help, Viewer 창 : 그래프, 도움말, HTML 등 명령어 실행 결과



1. 직접 명령문

- Console 창에서 명령어를 직접 입력하여 엔터를 쳐서 실행하거나 print() 함수를 사용하여 실행
- R을 마치 계산기처럼 사용
- 콘솔의 환영 메시지는 Edit>Clear Console(또는 Ctrl+L)을 선택하여 지운 후 (예제 1-1)과 같이 명령어를 입력하여 엔터를 치면 (예제 1-1)의 실행 결과를 보여줌

(예제 1-1) 직접 명령문 1
> 2^3
> 2*3
> 3/3
> 3+3
> 3-3
> q()

(예제 1-1)의 실행결과
> 2^3 # 2의 3승을 계산
[1] 8
> 2*3 # 2x3을 계산
[1] 6
> 3/3 # 3÷3을 계산
[1] 1
> 3+3 # 3+3을 계산
[1] 6
> 3-3 # 3-3을 계산
[1] 0
> q()

- Ctrl+L을 실행하여 console 창의 내용을 지운 후 (예제 1-2)와 같이 명령어를 입력하여 엔터를 치면 (예제 1-2)의 실행 결과를 보여줌

(예제 1-2) 직접 명령문 2

```
> print(2^.5, digits=5)
> print(2*3, digits=5)
> print(2/3, digits=5)
> print(3+3, digits=5)
> print(3-3, digits=5)
> q()
```

(예제 1-2)의 실행결과

```
> print(2^.5, digits=5) # 2의 제곱근을 계산하여 0이 아닌 숫자 5개 인쇄
[1] 1.4142
> print(2*3, digits=5) # 2의 3승을 계산하여 0이 아닌 숫자 5개 인쇄
[1] 6
> print(2/3, digits=5) # 2÷3을 계산하여 0이 아닌 숫자 5개 인쇄
[1] 0.66667
> print(3+3, digits=5) # 3+3을 계산하여 0이 아닌 숫자 5개 인쇄
[1] 6
> print(3-3, digits=5) # 3-3을 계산하여 0이 아닌 숫자 5개 인쇄
[1] 0
```

2. 할당 명령문

- 특정한 데이터 또는 연산 결과를 새로운 문자열에 할당하여 하나의 객체를 정의하는 명령문
- 할당 명령문의 형태는 할당 연산자인 <- (또는 ->)를 사용하는 형태와 할당 함수인 assign()을 사용
- 여기서 $x \leftarrow c(1,2,3,4,5)$ 는 1부터 5까지 5개의 수치형(numeric) 원소를 결합함수인 c()로 묶어 길이 5인 열 벡터를 생성한 후 식별문자 'x'에 할당하는 명령문
- c는 concatenate의 앞 글자인데 combine 또는 column이라고 기억해 두는 것도 편리함

$$X = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

- (예제 1-3)과 같이 명령어를 입력하여 엔터를 치면 (예제 1-3)의 실행 결과를 보여줌
- 할당 명령문에 의해 생성된 객체를 제거하려면 rm() 함수를 이용
- (예제 1-4)와 같이 명령어를 입력하여 엔터를 치면 (예제 1-4)의 실행 결과를 보여줌

(예제 1-3) 할당 명령문 1

```
> x<-c(1,2,3,4,5)
> y<-c(1:10)
> z<-x+y
> x
> y
> z
>rm(z)
>z
```

(예제 1-3)의 실행결과

```
> x<-c(1,2,3,4,5) # c는 여러 개의 값(1,2,3,4,5)을 x로 저장하는 명령문
> y<-c(1:10)      # 1부터 10까지 숫자를 y로 저장
> z<-x+y         # x와 y의 합 z를 계산
> x
[1] 1 2 3 4 5
> y
[1] 1 2 3 4 5 6 7 8 9 10
> z
[1] 2 4 6 8 10 7 9 11 13 15
> rm(z)          # z를 제거
> z
Error: object 'z' not found
```

(예제 1-4) 할당 명령문 2

```
> assign("x", c(1,2,3,4,5))
> assign("y", c(1:10))
> assign("z", x+y)
> x
> y
> z
> rm(z)
> z
```

(예제 1-4)의 실행결과

```
> assign("x", c(1,2,3,4,5)) # 여러 개의 값(1,2,3,4,5)을 x에 저장
> assign("y", c(1:10))     # 1부터 10까지 숫자를 y로 저장
> assign("z", x+y)        # x와 y의 합을 z로 저장
> x
[1] 1 2 3 4 5
> y
[1] 1 2 3 4 5 6 7 8 9 10
> z
[1] 2 4 6 8 10 7 9 11 13 15
> rm(z)
> z
Error: object 'z' not found
```

3. 코드 입력 및 실행

- Source 창에서 프로그램 Source를 작성, 편집, 저장, 실행, 불러오기 등을 할 수 있음
- 작성된 프로그램을 한 줄씩 실행하는 방법은 Run을 클릭(또는 Ctrl+Enter)
- 여러 줄 또는 모든 줄을 동시에 실행하는 방법은 여러 줄 또는 모든 줄을 선택하고 Run을 클릭 (Ctrl+Enter)
- 모든 줄을 동시에 실행하는 또 다른 방법은 Ctrl+Alt+R
- b3-ch1-1.R과 같이 명령어를 입력하여 Ctrl+Alt+R을 클릭하면 b3-ch1-1.R 실행 결과를 보여줌

```
b3-ch1-1.R
x<-c(1:10)
x
sort(x)
sort(x, decreasing=T)
mean(x)
median(x)
quantile(x)
diff(range(x))
var(x)
sd(x)
```

```
b3-ch1-1.R의 실행결과
> x<-c(1:10) # 1부터 10까지 숫자를 x로 저장
> x
 [1] 1 2 3 4 5 6 7 8 9 10
> sort(x) # x를 오름차순으로 정렬
 [1] 1 2 3 4 5 6 7 8 9 10
> sort(x, decreasing=T) # x를 내림차순으로 정렬
 [1] 10 9 8 7 6 5 4 3 2 1
> mean(x) # x의 산술평균을 계산
 [1] 5.5
> median(x) # x의 중위수를 계산
 [1] 5.5
> quantile(x) # x의 분위수를 계산
 0% 25% 50% 75% 100%
 1.00 3.25 5.50 7.75 10.00
> diff(range(x)) # x의 범위를 계산
 [1] 9
> var(x) # x의 분산을 계산
 [1] 9.166667
> sd(x) # x의 표준편차를 계산
 [1] 3.02765
```


4. 프로그램 저장하기 및 불러오기

- 작성된 프로그램 Source를 저장하기 위해서는 File/Save As를 선택한 후 다음 그림과 같이 본인이 원하는 디렉터리에 파일이름(예, b3-ch1-1)을 입력하면 되는데 프로그램의 확장자는 R로 지정이 됨
- 저장된 파일을 불러오기 위해서는 File/Open File을 선택한 후 불러 올 파일이 위치한 디렉터리에서 해당 파일을 불러오면 됨

