



# 데이터 분석을 위한 기본 개념 익히기

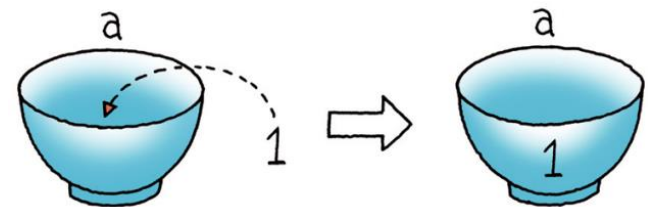
엄진영

- 변수(Variable)

- 다양한 값을 지니고 있는 하나의 속성
- 변수는 데이터 분석의 대상

변수			상수
소득	성별	학점	국적
1,000만 원	남자	3.8	대한민국
2,000만 원	남자	4.2	대한민국
3,000만 원	여자	2.6	대한민국
4,000만 원	여자	4.5	대한민국

```
a <- 1  
a  
## [1] 1  
b <- 2  
b  
## [1] 2  
c <- 3  
c  
## [1] 3  
d <- 3.5  
d  
## [1] 3.5
```



$a+b$

## [1] 3

$a+b+c$

## [1] 6

$4/b$

## [1] 2

$5*b$

## [1] 10

- 알아보기 쉽고 기억하기 쉽도록 의미를 담아 이름을 정함
- 변수명은 문자, 숫자, 대시(-), 언더바(\_)를 조합
  - 단 문자로 시작
- 변수명은 한글로 정해도 되지만 간혹 오류가 발생하는 경우가 있으니 영문으로 하길 권장
- 대소문자를 구분하므로 헛갈리지 않도록 모든 변수는 소문자로 만드는 습관을 들이는게 좋음



- c()

- ‘c’는 합치다를 의미하는 ‘combine’의 머릿글로 c() 함수는 변수에 여러개의 값을 넣는 기능을 함

```
var1 <- c(1, 2, 5, 7, 8)  # 숫자 다섯 개로 구성된 var1 생성
var1
## [1] 1 2 5 7 8
var2 <- c(1:5)           # 1~5까지 연속값으로 var2 생성
var2
## [1] 1 2 3 4 5
```

- seq()

```
var3 <- seq(1, 5)      # 1~5까지 연속값으로 var3 생성
```

```
var3
```

```
## [1] 1 2 3 4 5
```

```
var4 <- seq(1, 10, by = 2) # 1~10까지 2 간격 연속값으로 var4 생성
```

```
var4
```

```
## [1] 1 3 5 7 9
```

```
var5 <- seq(1, 10, by = 3) # 1~10까지 3 간격 연속값으로 var5 생성
```

```
var5
```

```
## [1] 1 4 7 10
```

```
var1
```

```
## [1] 1 2 5 7 8
```

```
var1+2
```

```
## [1] 3 4 7 9 10
```

```
var1
```

```
## [1] 1 2 5 7 8
```

```
var2
```

```
## [1] 1 2 3 4 5
```

```
var1+var2
```

```
## [1] 2 4 8 11 13
```



```
str1 <- "a"
```

```
str1
```

```
## [1] "a"
```

```
str2 <- "text"
```

```
str2
```

```
## [1] "text"
```

```
str3 <- "Hello World!"
```

```
str3
```

```
## [1] "Hello World!"
```

```
str4 <- c("a", "b", "c")
```

```
str4
```

```
## [1] "a" "b" "c"
```

```
str5 <- c("Hello!", "World", "is", "good!")
```

```
str5
```

```
## [1] "Hello!" "World"  "is"     "good!"
```

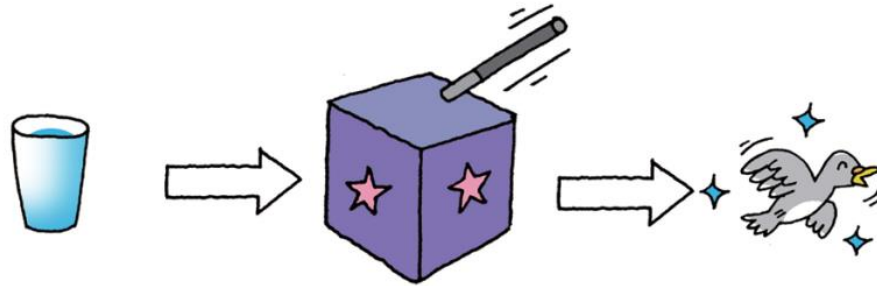
```
str1+2
```

```
## Error in str1 + 2: non-numeric argument to binary operator
```

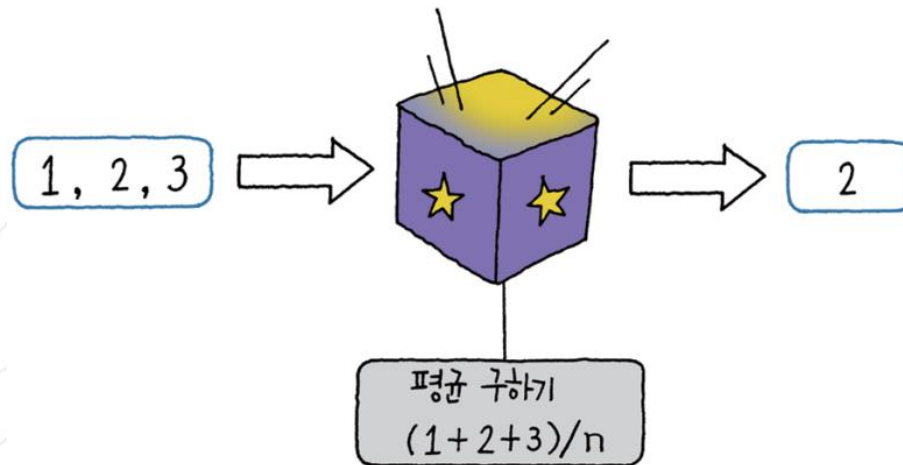


## • 함수

- 값을 넣으면 특정한 기능을 수행해 처음과 다른 값이 출력됨



마법 상자 같은 역할을 하는 함수



평균을 구하는 함수

```
# 변수 만들기  
x <- c(1, 2, 3)  
x  
## [1] 1 2 3  
# 함수 적용하기  
mean(x)  
## [1] 2  
max(x)  
## [1] 3  
min(x)  
## [1] 1
```

```
str5
## [1] "Hello!" "World" "is"   "good!"
paste(str5, collapse = ",") # 쉼표를 구분자로 str4의 단어들 하나로 합치기
## [1] "Hello!,World,is,good!"
```

- 함수의 옵션 설정하기 - 파라미터

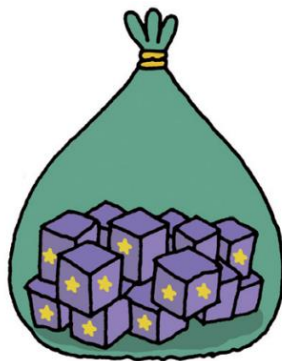
```
paste(str5, collapse = " ")
## [1] "Hello! World is good!"
```

- 함수의 결과물로 새 변수 만들기

```
x_mean <- mean(x)
x_mean
## [1] 2
str5_paste <- paste(str5, collapse = " ")
str5_paste
## [1] "Hello! World is good!"
```

- 패키지(packages)

- 함수가 여러 개 들어 있는 꾸러미
- 하나의 패키지 안에 다양한 함수가 들어있음
- 함수를 사용하려면 패키지 설치 먼저 해야함



패키지 설치하기



패키지 로드하기

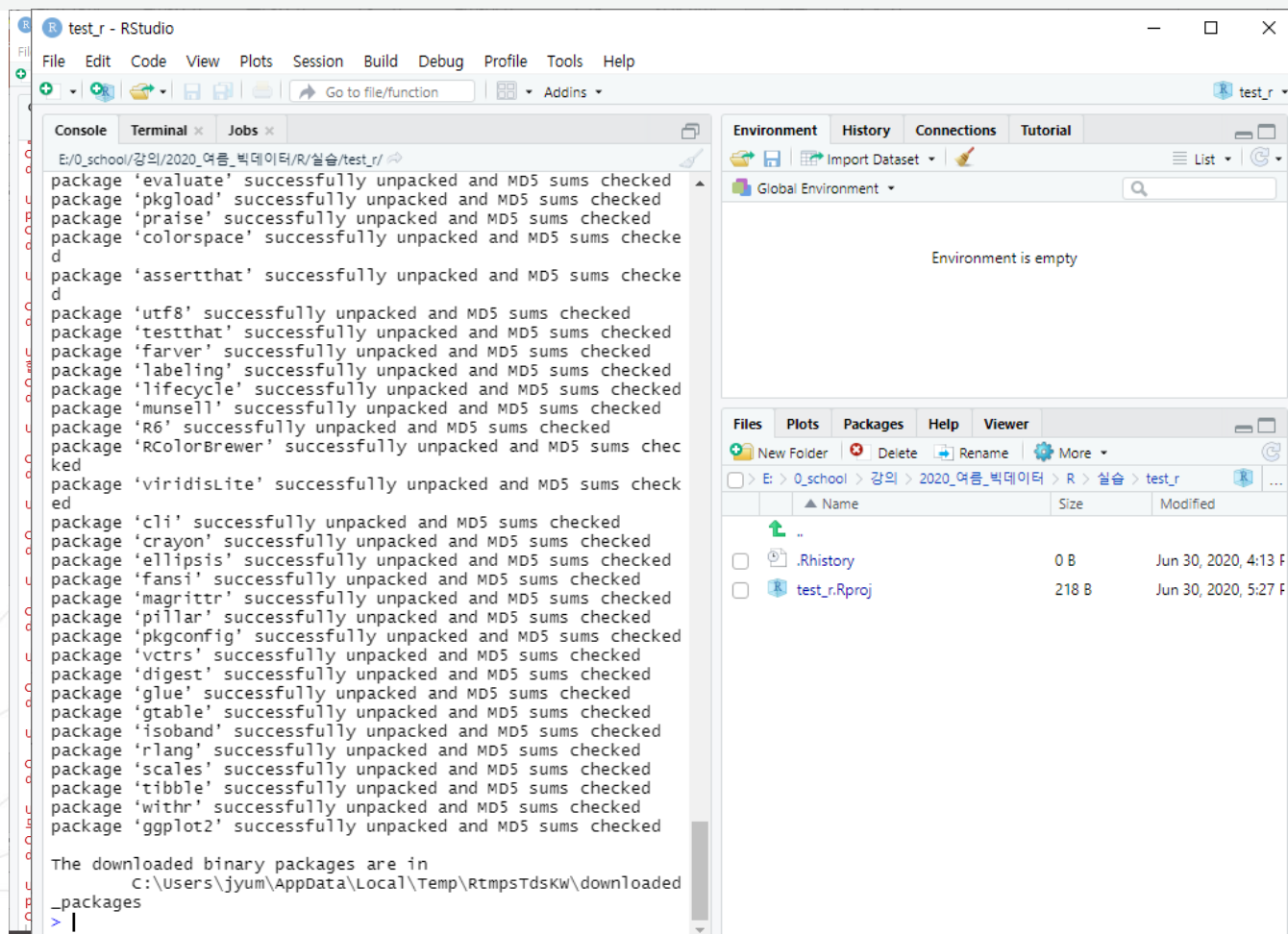


함수 사용하기

패키지는 한번만 설치하면 되지만 **패키지를 로드하는 작업은 R스튜디오를 새로 시작할때마다 반복해야 한다.** 패키지를 로드하지 않은 상태에서 함수를 실행하면 함수를 사용할 수 없다는 에러 메시지가 출력된다.

- 데이터를 그래프로 표현하는 작업을 할 때 가장 많이 사용하는 패키지인 ggplot2를 설치한다.

```
install.packages("ggplot2") # ggplot2 패키지 설치
```



- ggplot2 패키지 로드하기

```
library(ggplot2)      # ggplot2 패키지 로드
```

- 함수 사용하기

```
# 여러 문자로 구성된 변수 생성
```

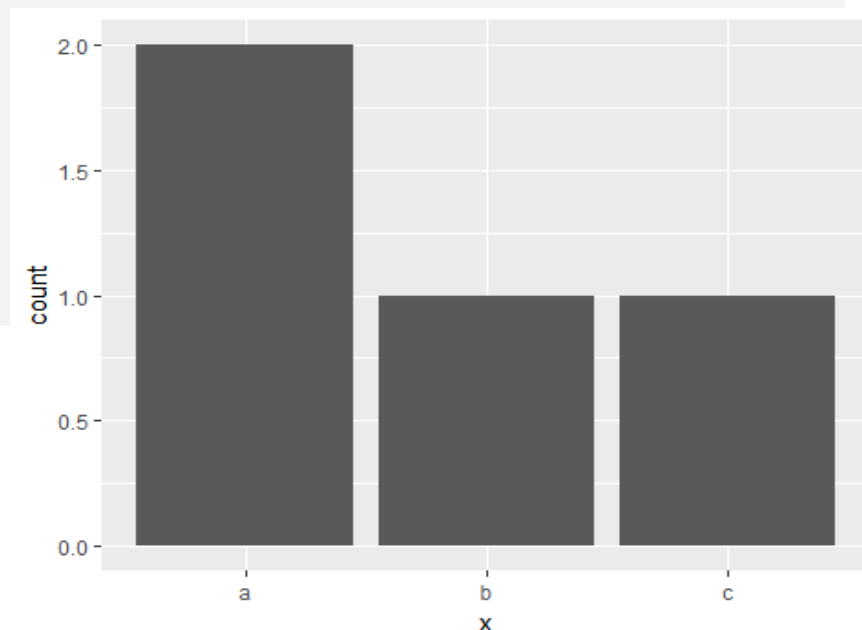
```
x <- c("a", "a", "b", "c")
```

```
x
```

```
## [1] "a" "a" "b" "c"
```

```
# 빈도 그래프 출력
```

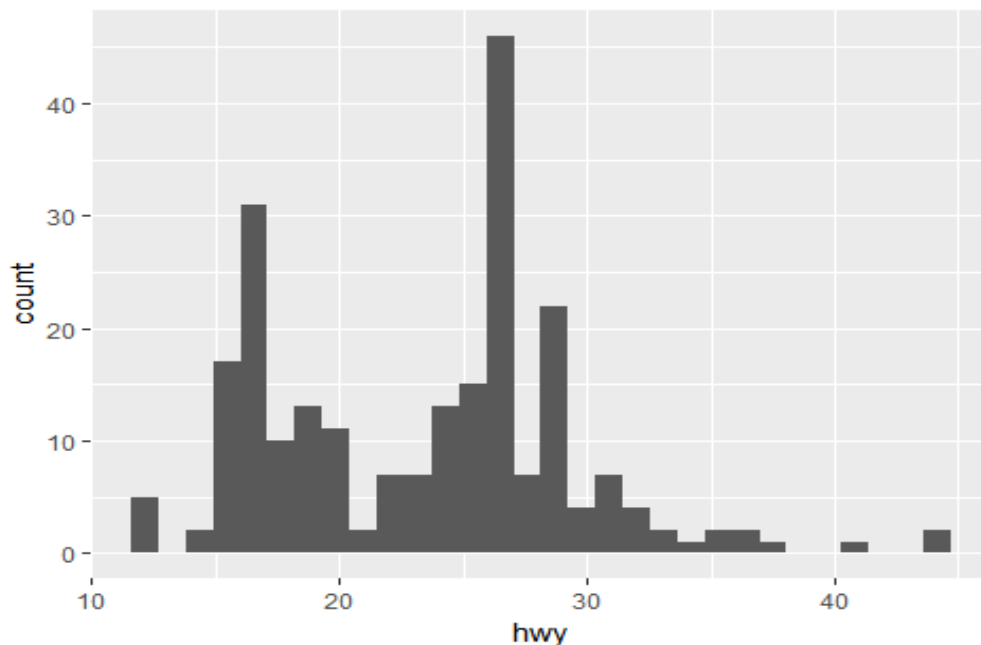
```
qplot(x)
```





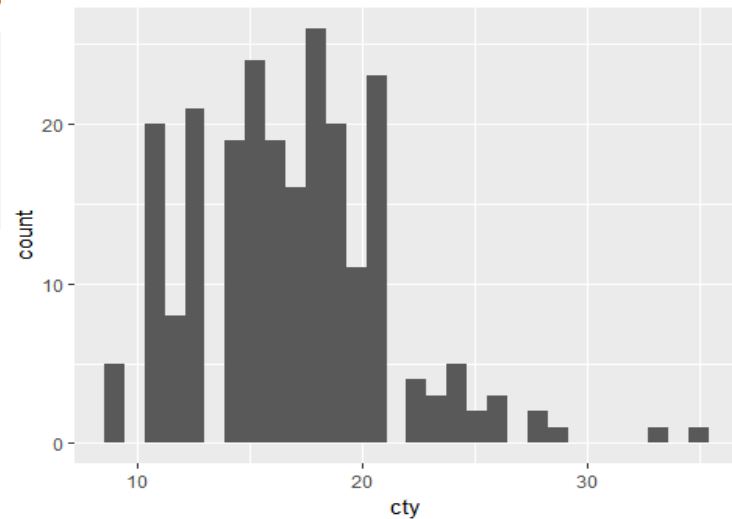
- Mpg(Mile per Gallon)데이터는 미국 환경보호국에서 공개한 자료로 1999~2008년 사이 미국에서 출시된 자동차 234종의 연비 관련 정보를 담고 있음
  - hwy는 자동차가 고속도로에서 1갤런에 몇 마일을 가는지 나타낸 변수

```
# data에 mpg, x축에 hwy 변수 지정하여 그래프 생성  
qplot(data = mpg, x = hwy)
```



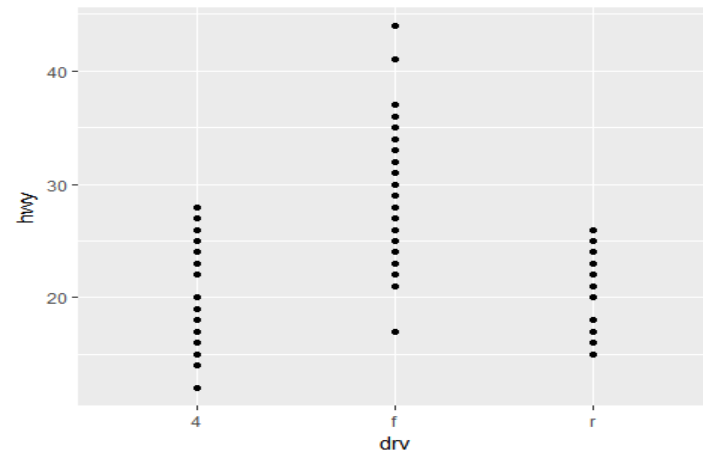
```
# x축 cty
```

```
qplot(data = mpg, x = cty)
```



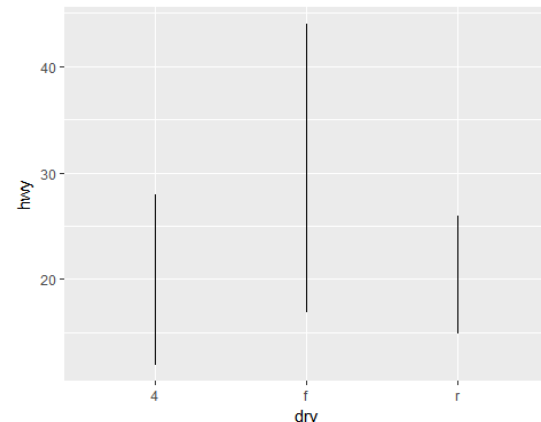
```
# x축 drv, y축 hwy
```

```
qplot(data = mpg, x = drv, y = hwy)
```



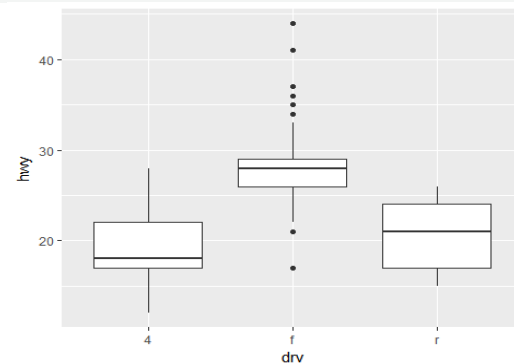
# x축 *drv*, y축 *hwy*, 선 그래프 형태

```
qplot(data = mpg, x = drv, y = hwy, geom = "line")
```



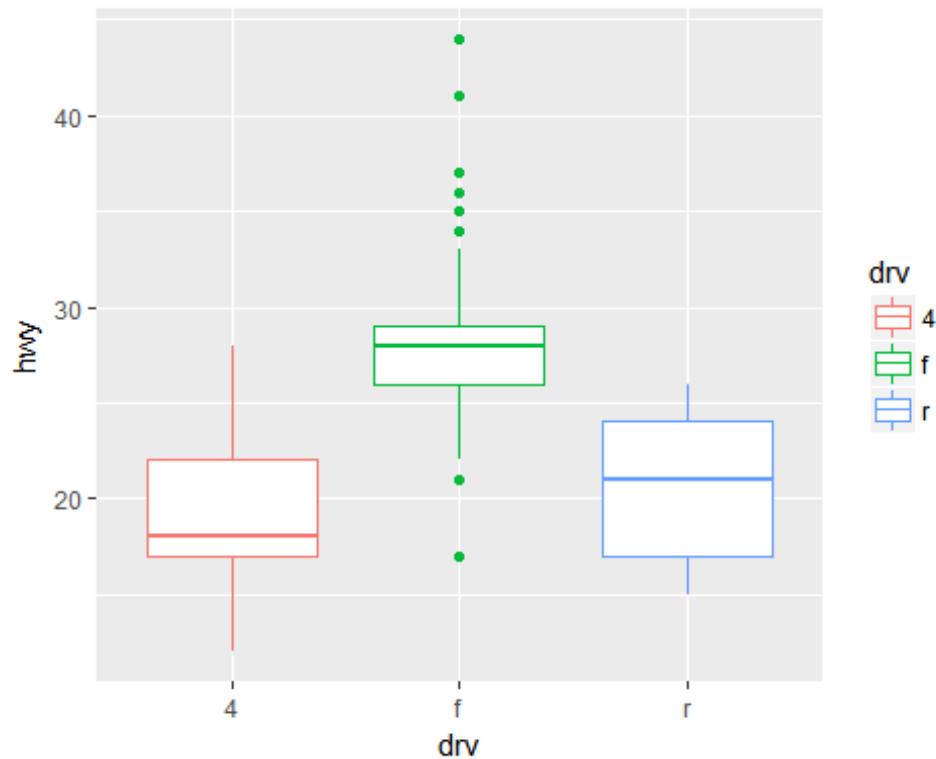
# x축 *drv*, y축 *hwy*, 상자 그림 형태

```
qplot(data = mpg, x = drv, y = hwy, geom = "boxplot")
```

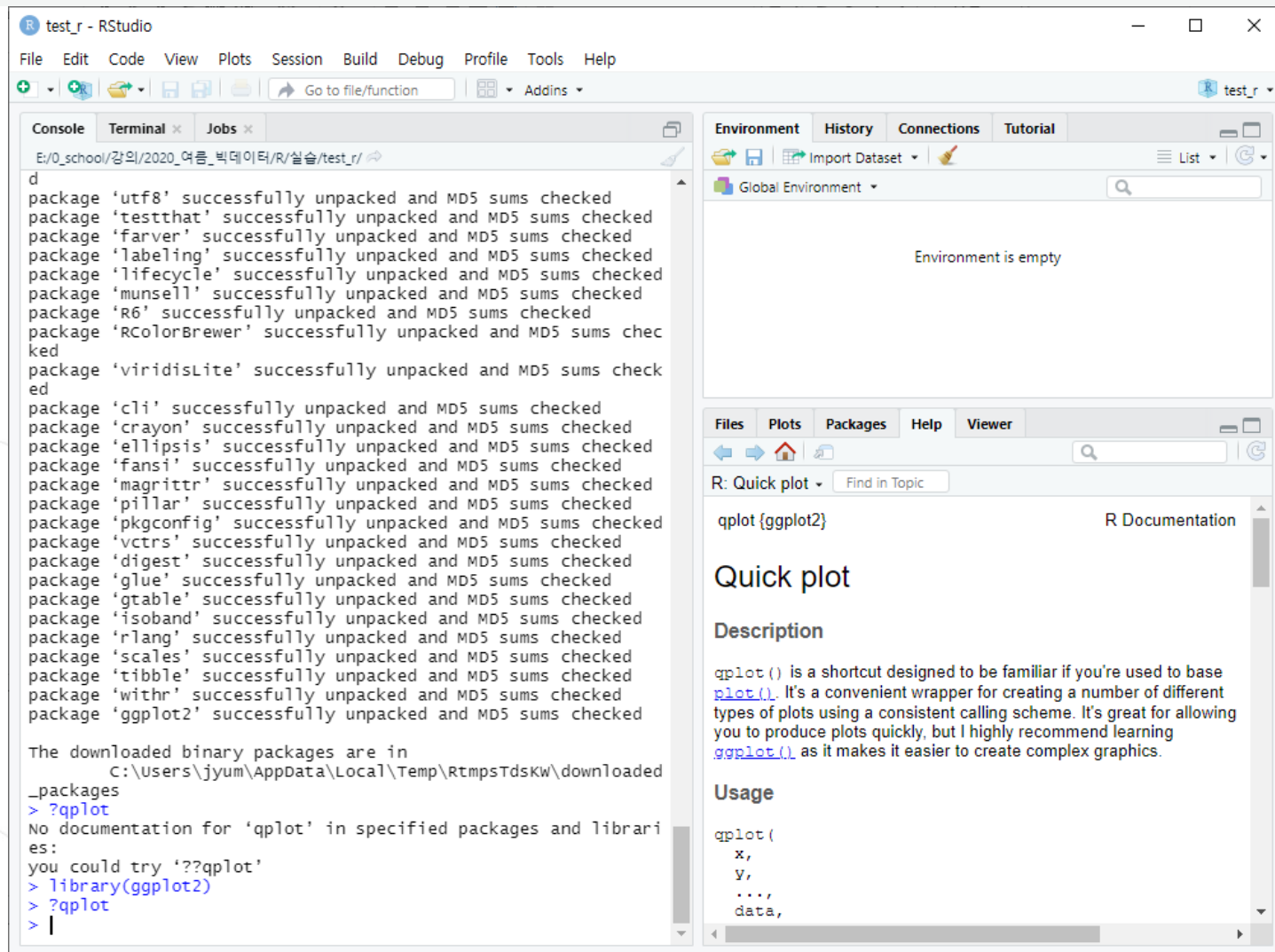


# x축 drv, y축 hwy, 상자 그림 형태, drv별 색 표현

```
qplot(data = mpg, x = drv, y = hwy, geom = "boxplot", colour = drv)
```



## ?qplot



The screenshot shows the RStudio interface. The console on the left displays the output of installing various packages, including 'utf8', 'testthat', 'farver', 'labeling', 'lifecycle', 'munsell', 'R6', 'RColorBrewer', 'viridisLite', 'cli', 'crayon', 'ellipsis', 'fansi', 'magrittr', 'pillar', 'pkgconfig', 'vctrs', 'digest', 'glue', 'gtable', 'isoband', 'rlang', 'scales', 'tibble', 'withr', and 'ggplot2'. The message "The downloaded binary packages are in C:\Users\jyum\AppData\Local\Temp\RtmpsTdsKw\downloaded\_packages" is shown, followed by the command `> ?qplot` and the response "No documentation for 'qplot' in specified packages and libraries: you could try '??qplot'", and finally `> library(ggplot2)` and `> ?qplot`.

The right pane shows the "R: Quick plot" help page from the R Documentation. The page title is "Quick plot". The description states: "qplot() is a shortcut designed to be familiar if you're used to base plot(). It's a convenient wrapper for creating a number of different types of plots using a consistent calling scheme. It's great for allowing you to produce plots quickly, but I highly recommend learning ggplot() as it makes it easier to create complex graphics." The usage section shows the function signature: `qplot(x, y, ..., data,`.

### Q1. 시험 점수 변수 만들고 출력하기

다섯 명의 학생이 시험을 봤습니다. 학생 다섯 명의 시험 점수를 담고 있는 변수를 만들어 출력해 보세요. 각 학생의 시험 점수는 다음과 같습니다.

80, 60, 70, 50, 90

### Q2. 전체 평균 구하기

앞 문제에서 만든 변수를 이용해서 이 학생들의 전체 평균 점수를 구해보세요.

### Q3. 전체 평균 변수 만들고 출력하기

전체 평균 점수를 담고 있는 새 변수를 만들어 출력해 보세요. 앞 문제를 풀 때 사용한 코드를 응용하면 됩니다.