


변하는 수, '변수' 이해하기

변수(Variable)

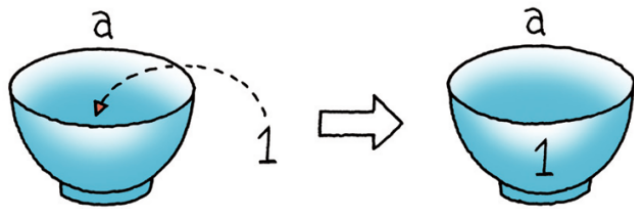
- 다양한 값을 지니고 있는 하나의 속성
- 변수는 데이터 분석의 대상



The diagram illustrates the classification of variables and constants. A box labeled '변수' (Variable) is connected by a red line to the first three columns of the table: '소득' (Income), '성별' (Gender), and '학점' (GPA). A box labeled '상수' (Constant) is connected by a red line to the fourth column: '국적' (Nationality).

변수	소득	성별	학점	상수	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

변수 만들기



```
a <- 1
a
## [1] 1

b <- 2
b
## [1] 2

c <- 3
c
## [1] 3

d <- 3.5
d
## [1] 3.5
```

변수로 연산하기

a+b

[1] 3

a+b+c

[1] 6

4/b

[1] 2

5*b

[1] 10

여러 값으로 구성된 변수 만들기

`c()`

```
var1 <- c(1, 2, 5, 7, 8)    # 숫자 다섯 개로 구성된 var1 생성  
var1
```

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

```
var2 <- c(1:5)              # 1~5 까지 연속값으로 var2 생성  
var2
```

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

seq()

```
var3 <- seq(1, 5)           # 1~5 까지 연속값으로 var3 생성
var3
## [1] 1 2 3 4 5

var4 <- seq(1, 10, by = 2)  # 1~10 까지 2 간격 연속값으로 var4 생성
var4
## [1] 1 3 5 7 9

var5 <- seq(1, 10, by = 3)  # 1~10 까지 3 간격 연속값으로 var5 생성
var5
## [1] 1 4 7 10
```

연속값 변수로 연산하기

```
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
```

문자로 된 변수 만들기

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

```
str1
```

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

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

```
str2
```

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

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

```
str3
```

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

연속 문자 변수 만들기

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

```
str4
```

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

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

```
str5
```

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


문자로 된 변수로는 연산할 수 없다

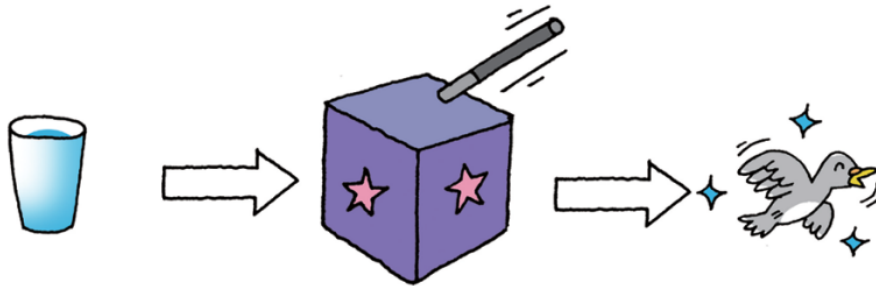
```
str1+2
```

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

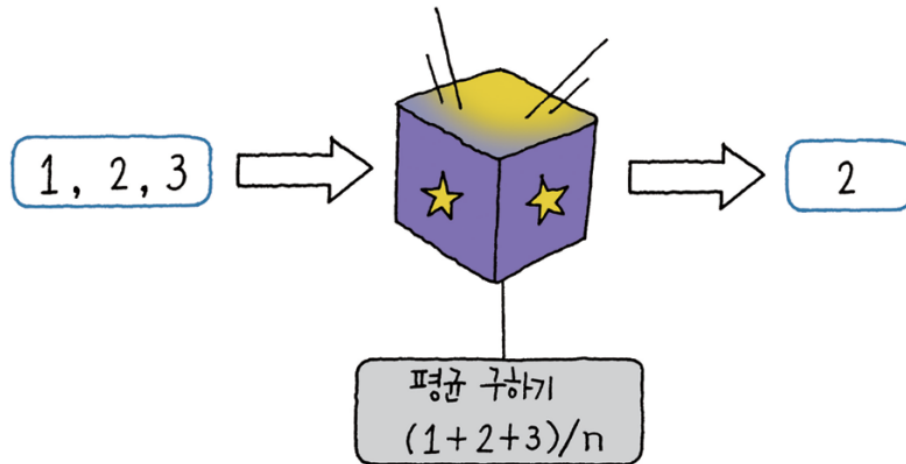
마술 상자 같은 '함수' 이해하기

함수

- 값을 넣으면 특정한 기능을 수행해 처음과 다른 값이 출력됨



마법 상자 같은 역할을 하는 함수



평균을 구하는 함수

숫자를 다루는 함수 사용하기

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
```

문자를 다루는 함수 사용하기

```
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!"
```

데이터는 어떻게 생겼나? - 데이터 프레임 이해하기

데이터 프레임

이름	영어 점수	수학 점수
김지훈	90	50
이유진	80	60
박동현	60	100
김민지	70	20

데이터 프레임



- '열'은 속성
- '행'은 한 사람의 정보

데이터가 크다 = 행이 많다 또는 열이 많다

데이터의 행이 늘어난다면?

번호	성별	연령
1	남자	26
2	여자	42
⋮	⋮	⋮
1,000,000	남자	27

데이터의 열이 늘어난다면?

번호	성별	연령	학점	연봉	...	출신지	전공
1	남자	26	3.8	2,700만	...	서울	경영
2	여자	42	4.2	4,000만	...	부산	심리
3	남자	27	2.6	3,200만	...	대전	사회

데이터 프레임 만들기 - 시험 성적 데이터를 만들어 보자!

데이터 프레임 한 번에 만들기

```
df_midterm <- data.frame(english = c(90, 80, 60, 70),  
                          math = c(50, 60, 100, 20),  
                          class = c(1, 1, 2, 2))
```

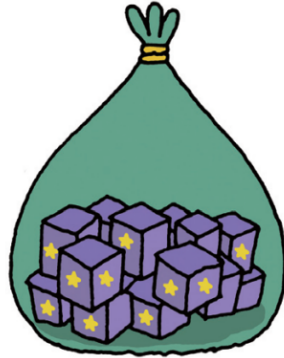
```
df_midterm
```

```
##   english math class  
## 1      90   50     1  
## 2      80   60     1  
## 3      60  100     2  
## 4      70   20     2
```

함수 꾸러미, '패키지' 이해하기

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

04-3. 외부 데이터 이용하기 - 축적된 시험 성적 데이터를 불러오자!

엑셀 파일 불러오기

```
# readxl 패키지 설치  
install.packages("readxl")  
  
# readxl 패키지 로드  
library(readxl)
```

```
df_exam <- read_excel("excel_exam.xlsx") # 엑셀 파일을 불러와서 df_exam 에 할당
df_exam # 출력
```

```
## # A tibble: 20 x 5
```

```
##       id class  math english science
##   <dbl> <dbl> <dbl>   <dbl>   <dbl>
## 1     1     1     50     98     50
## 2     2     1     60     97     60
## 3     3     1     45     86     78
## 4     4     1     30     98     58
## 5     5     2     25     80     65
## 6     6     2     50     89     98
## 7     7     2     80     90     45
## 8     8     2     90     78     25
## 9     9     3     20     98     15
## 10    10     3     50     98     45
## 11    11     3     65     65     65
## 12    12     3     45     85     32
## 13    13     4     46     98     65
## 14    14     4     48     87     12
## 15    15     4     75     56     78
## 16    16     4     58     98     65
## 17    17     5     65     68     98
## 18    18     5     80     78     90
## 19    19     5     89     68     87
## 20    20     5     78     83     58
```

```
mean(df_exam$english)
```

```
## [1] 84.9
```

```
mean(df_exam$science)
```

```
## [1] 59.45
```

직접 경로 지정

```
df_exam <- read_excel("d:/easy_r/excel_exam.xlsx")
```

[주의] Working directory에 불러올 파일이 있어야 함

엑셀 파일 첫 번째 행이 변수명이 아니라면?

```
df_exam_novar <- read_excel("excel_exam_novar.xlsx", col_names = F)
df_exam_novar
```

엑셀 파일에 시트가 여러 개 있다면?

```
df_exam_sheet <- read_excel("excel_exam_sheet.xlsx", sheet = 3)
df_exam_sheet
```

csv 파일 불러오기

- 범용 데이터 형식
- 값 사이를 쉼표(,)로 구분
- 용량 작음, 다양한 소프트웨어에서 사용

```
df_csv_exam <- read.csv("csv_exam.csv")  
df_csv_exam
```

```
##      id class math english science  
## 1     1     1   50      98       50  
## 2     2     1   60      97       60  
## 3     3     1   45      86       78  
## 4     4     1   30      98       58  
## 5     5     2   25      80       65  
## 6     6     2   50      89       98  
## 7     7     2   80      90       45  
## 8     8     2   90      78       25  
## 9     9     3   20      98       15  
## 10    10     3   50      98       45  
## 11    11     3   65      65       65  
## 12    12     3   45      85       32  
## 13    13     4   46      98       65  
## 14    14     4   48      87       12  
## 15    15     4   75      56       78  
## 16    16     4   58      98       65
```


##	17	17	5	65	68	98
##	18	18	5	80	78	90
##	19	19	5	89	68	87
##	20	20	5	78	83	58

문자가 들어 있는 파일을 불러올 때는 `stringsAsFactors = F`

```
df_csv_exam <- read.csv("csv_exam.csv", stringsAsFactors = F)
```

데이터 프레임을 CSV 파일로 저장하기

```
df_midterm <- data.frame(english = c(90, 80, 60, 70),  
                          math = c(50, 60, 100, 20),  
                          class = c(1, 1, 2, 2))
```

```
df_midterm
```

```
##   english math class  
## 1      90   50     1  
## 2      80   60     1  
## 3      60  100     2  
## 4      70   20     2
```

```
write.csv(df_midterm, file = "df_midterm.csv")
```

정리하기

1. 변수 만들기, 데이터 프레임 만들기

```
english <- c(90, 80, 60, 70) # 영어 점수 변수 생성
math <- c(50, 60, 100, 20)   # 수학 점수 변수 생성
data.frame(english, math)    # 데이터 프레임 생성
```

2. 외부 데이터 이용하기

엑셀 파일

```
library(readxl) # readxl 패키지 로드
df_exam <- read_excel("excel_exam.xlsx") # 엑셀 파일 불러오기
```

CSV 파일

```
df_csv_exam <- read.csv("csv_exam.csv") # CSV 파일 불러오기
write.csv(df_midterm, file = "df_midterm.csv") # CSV 파일로 저장하기
```

Rda 파일

```
load("df_midterm.rda") # Rda 파일 불러오기
save(df_midterm, file = "df_midterm.rda") # Rda 파일로 저장하기
```

05-1. 데이터 파악하기

함수	기능
head()	데이터 앞부분 출력
tail()	데이터 뒷부분 출력
View()	뷰어 창에서 데이터 확인
dim()	데이터 차원 출력
str()	데이터 속성 출력
summary()	요약통계량 출력

exam 데이터 파악하기

데이준 준비

```
exam <- read.csv("csv_exam.csv")
```

head() - 데이터 앞부분 확인하기

`head(exam)` *# 앞에서부터 6 행까지 출력*

##	id	class	math	english	science
## 1	1	1	50	98	50
## 2	2	1	60	97	60
## 3	3	1	45	86	78
## 4	4	1	30	98	58
## 5	5	2	25	80	65
## 6	6	2	50	89	98

`head(exam, 10)` *# 앞에서부터 10 행까지 출력*

##	id	class	math	english	science
## 1	1	1	50	98	50
## 2	2	1	60	97	60
## 3	3	1	45	86	78
## 4	4	1	30	98	58
## 5	5	2	25	80	65
## 6	6	2	50	89	98
## 7	7	2	80	90	45
## 8	8	2	90	78	25
## 9	9	3	20	98	15
## 10	10	3	50	98	45

tail() - 데이터 뒷부분 확인하기

```
tail(exam)      # 뒤에서부터 6 행까지 출력
```

```
##      id class math english science
## 15 15      4   75      56      78
## 16 16      4   58      98      65
## 17 17      5   65      68      98
## 18 18      5   80      78      90
## 19 19      5   89      68      87
## 20 20      5   78      83      58
```

```
tail(exam, 10)  # 뒤에서부터 10 행까지 출력
```

```
##      id class math english science
## 11 11      3   65      65      65
## 12 12      3   45      85      32
## 13 13      4   46      98      65
## 14 14      4   48      87      12
## 15 15      4   75      56      78
## 16 16      4   58      98      65
## 17 17      5   65      68      98
## 18 18      5   80      78      90
## 19 19      5   89      68      87
## 20 20      5   78      83      58
```


View() - 뷰어 창에서 데이터 확인하기

View(exam)

[유의] View()에서 맨 앞의 V는 대문자

dim() - 몇 행 몇 열로 구성되는지 알아보기

```
dim(exam)  # 행, 열 출력
```

```
## [1] 20  5
```

summary() - 요약통계량 산출하기

```
summary(exam)  # 요약통계량 출력
```

```
##           id           class           math           english
##  Min.      : 1.00    Min.      :1    Min.      :20.00    Min.      :56.0
##  1st Qu.: 5.75    1st Qu.:2    1st Qu.:45.75    1st Qu.:78.0
##  Median :10.50    Median :3    Median :54.00    Median :86.5
##  Mean   :10.50    Mean   :3    Mean   :57.45    Mean   :84.9
##  3rd Qu.:15.25    3rd Qu.:4    3rd Qu.:75.75    3rd Qu.:98.0
##  Max.    :20.00    Max.    :5    Max.    :90.00    Max.    :98.0
##      science
##  Min.      :12.00
##  1st Qu.:45.00
##  Median :62.50
##  Mean   :59.45
##  3rd Qu.:78.00
##  Max.    :98.00
```

ggplot2 – quick plot

ggplot2 패키지 설치하기, 로드하기

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

함수 사용하기

여러 문자로 구성된 변수 생성

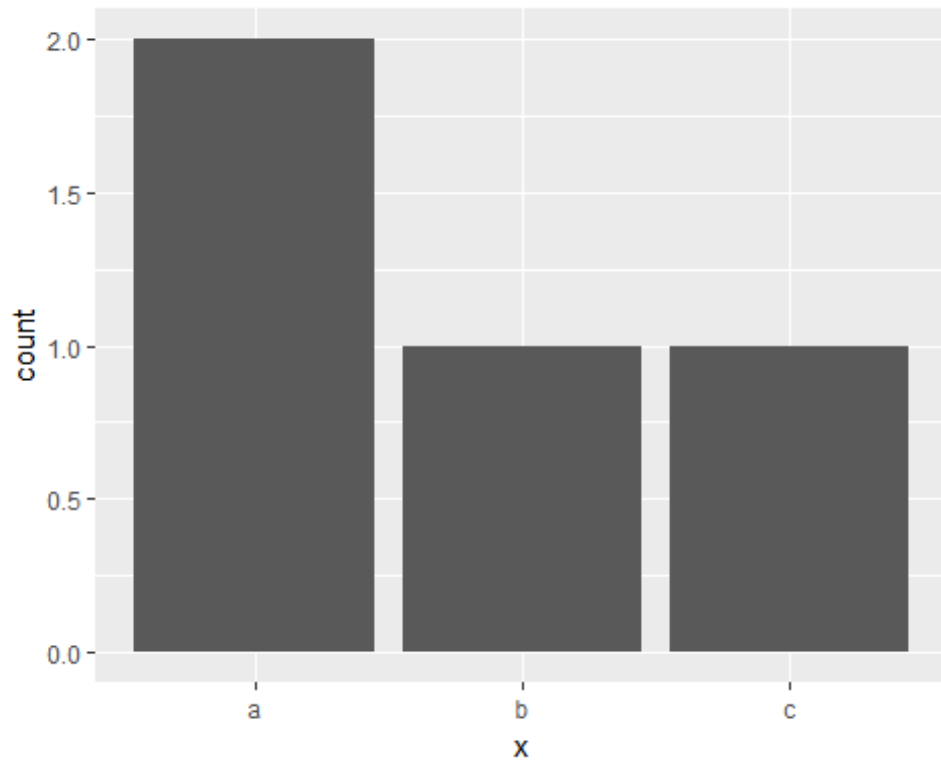
```
x <- c("a", "a", "b", "c")
```

```
x
```

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

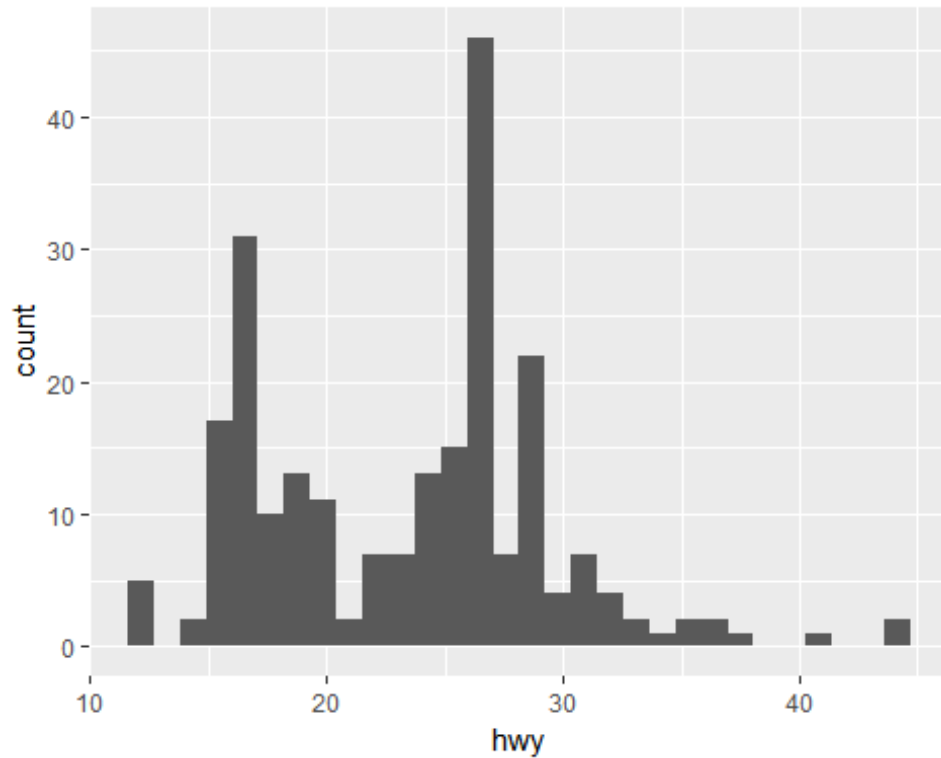
빈도 그래프 출력

```
qplot(x)
```



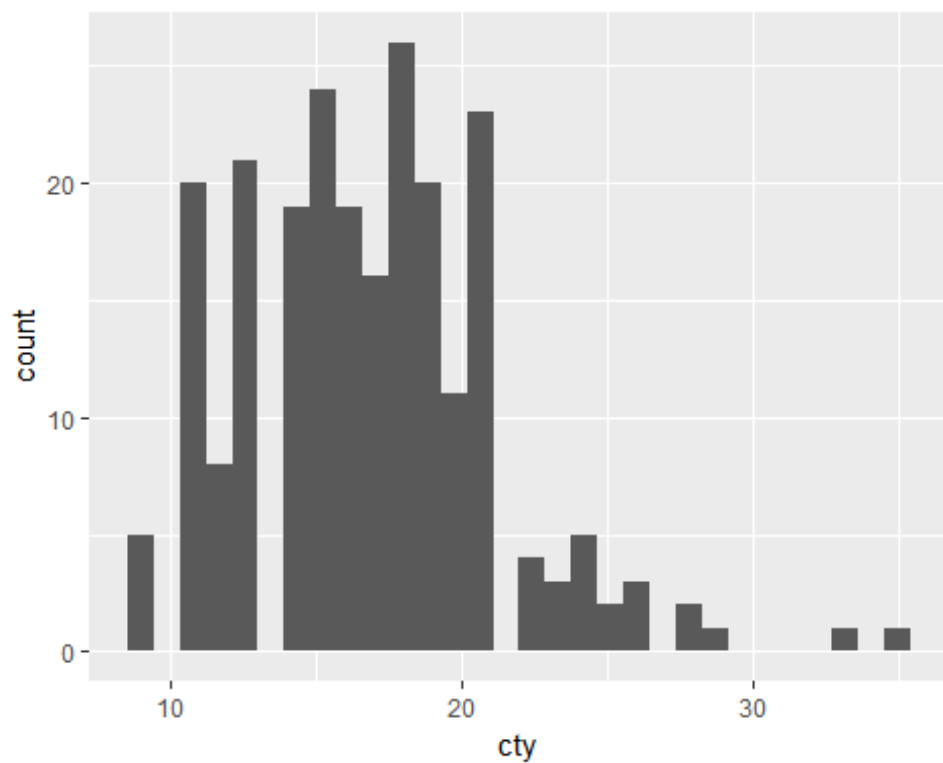
ggplot2의 mpg 데이터로 그래프 만들기

```
# data 에 mpg, x 축에 hwy 변수 지정하여 그래프 생성  
qplot(data = mpg, x = hwy)
```

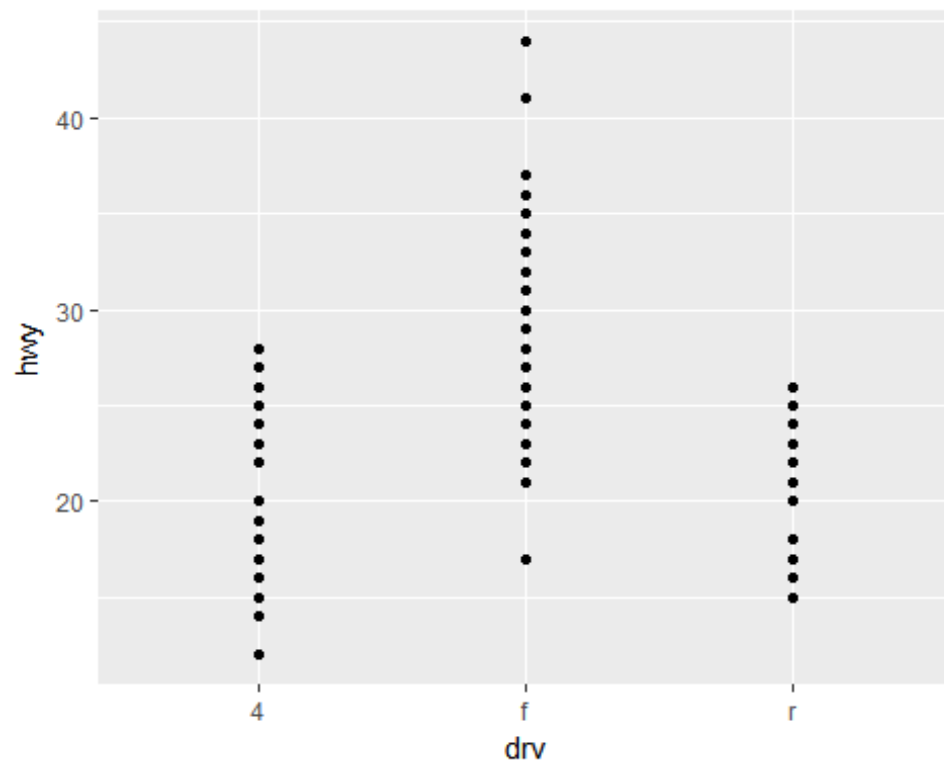


qplot() 파라미터 바꿔보기

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

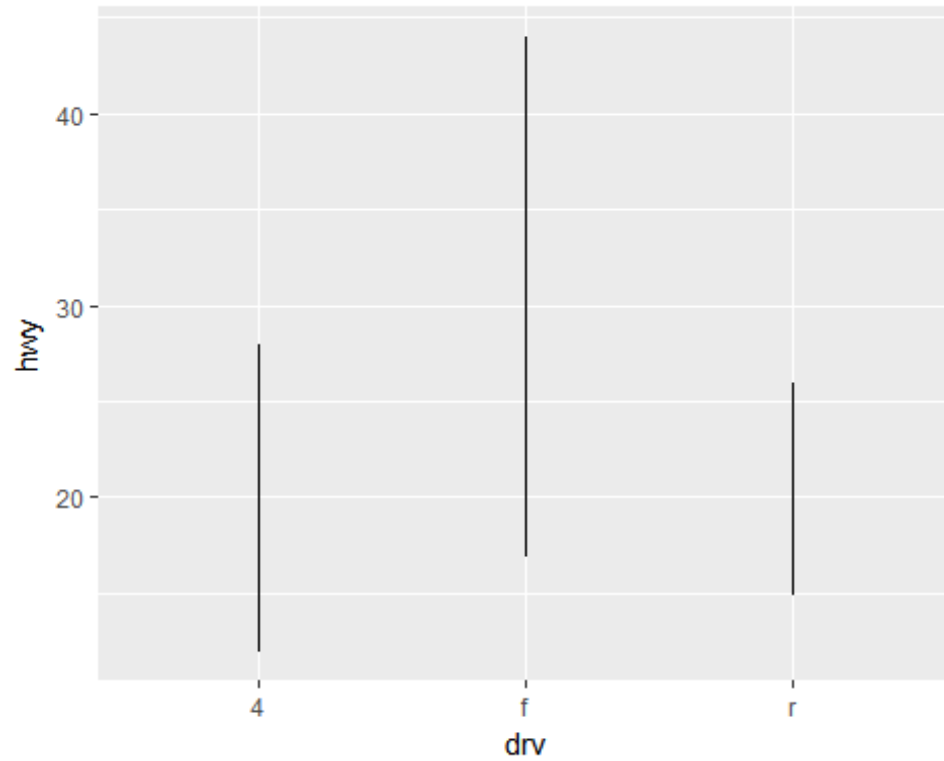



```
# x ≙ drv, y ≙ hwy  
qplot(data = mpg, x = drv, y = hwy)
```



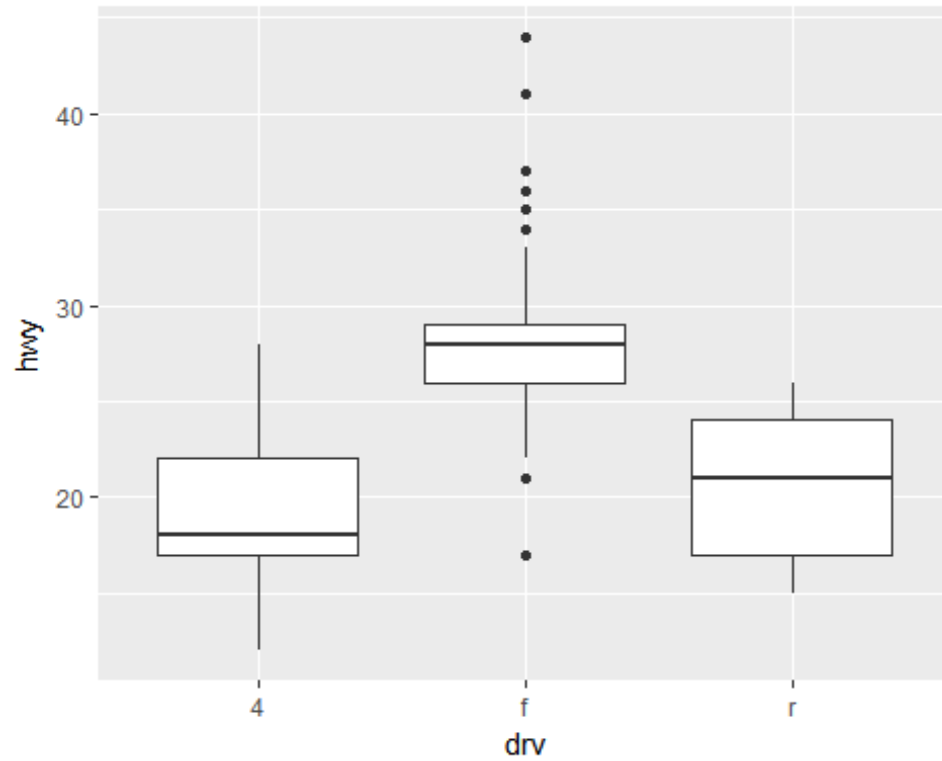
x 축 drv , y 축 hwy , 선 그래프 형태

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



x 축 *drv*, y 축 *hwy*, 상자 그림 형태

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



x 축 drv, y 축 hwy, 상자 그림 형태, drv 별 색 표현

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

