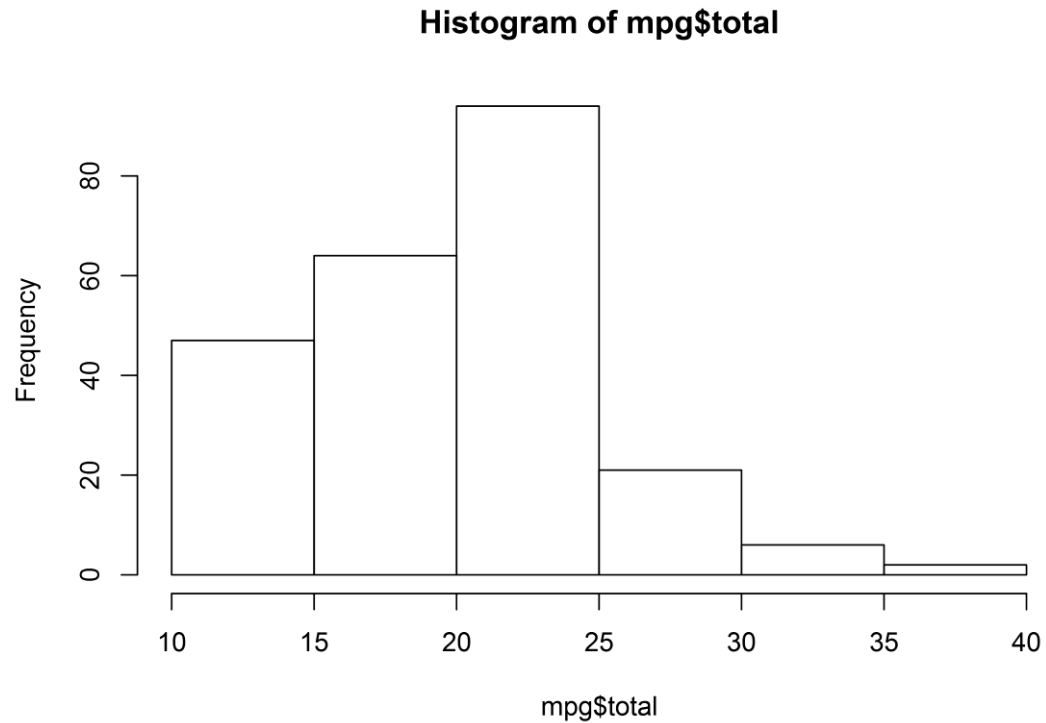


5. 데이터 분석 기초!

데이터 파악하기, 다루기 쉽게 수정하기



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
```

str() - 속성 파악하기

```
str(exam)  # 데이터 속성 확인
```

```
## 'data.frame':    20 obs. of  5 variables:
## $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
## $ class   : int  1 1 1 1 2 2 2 2 3 3 ...
## $ math    : int  50 60 45 30 25 50 80 90 20 50 ...
## $ english: int  98 97 86 98 80 89 90 78 98 98 ...
## $ science: int  50 60 78 58 65 98 45 25 15 45 ...
```


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
```

mpg 데이터 파악하기

ggplot2 의 mpg 데이터를 데이터 프레임 형태로 불러오기

```
mpg <- as.data.frame(ggplot2::mpg)
```

mpg 데이터 파악하기

`head(mpg)` *# Raw 데이터 앞부분 확인*

```
##      manufacturer model displ year  cyl      trans  drv  cty   hwy fl  class
## 1             audi   a4    1.8 1999   4    auto(l5)   f   18   29  p compact
## 2             audi   a4    1.8 1999   4 manual(m5)   f   21   29  p compact
## 3             audi   a4    2.0 2008   4 manual(m6)   f   20   31  p compact
## 4             audi   a4    2.0 2008   4    auto(av)   f   21   30  p compact
## 5             audi   a4    2.8 1999   6    auto(l5)   f   16   26  p compact
## 6             audi   a4    2.8 1999   6 manual(m5)   f   18   26  p compact
```

`tail(mpg)` *# Raw 데이터 뒷부분 확인*

```
##      manufacturer model displ year  cyl      trans  drv  cty   hwy fl  class
## 229    volkswagen  passat   1.8 1999   4    auto(l5)   f   18   29  p midsize
## 230    volkswagen  passat   2.0 2008   4    auto(s6)   f   19   28  p midsize
## 231    volkswagen  passat   2.0 2008   4 manual(m6)   f   21   29  p midsize
## 232    volkswagen  passat   2.8 1999   6    auto(l5)   f   16   26  p midsize
## 233    volkswagen  passat   2.8 1999   6 manual(m5)   f   18   26  p midsize
## 234    volkswagen  passat   3.6 2008   6    auto(s6)   f   17   26  p midsize
```

```
View(mpg)      # Raw 데이터 뷰어 창 확인
```

```
dim(mpg)       # 행, 열 출력
```

```
## [1] 234 11
```

```
str(mpg)       # 데이터 속성 확인
```

```
## 'data.frame':  234 obs. of  11 variables:
## $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
## $ model       : chr  "a4" "a4" "a4" "a4" ...
## $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
## $ year        : int  1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
## $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
## $ trans       : chr  "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
## $ drv         : chr  "f" "f" "f" "f" ...
## $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
## $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
## $ fl         : chr  "p" "p" "p" "p" ...
## $ class       : chr  "compact" "compact" "compact" "compact" ...
```

`summary(mpg)` # 요약통계량 출력

```
## manufacturer      model      displ      year
## Length:234        Length:234    Min.    :1.600    Min.    :1999
## Class :character   Class :character   1st Qu.:2.400    1st Qu.:1999
## Mode  :character   Mode  :character   Median :3.300    Median :2004
##                                     Mean   :3.472    Mean   :2004
##                                     3rd Qu.:4.600    3rd Qu.:2008
##                                     Max.    :7.000    Max.    :2008
##      cyl      trans      drv      cty
## Min.    :4.000    Length:234    Length:234    Min.    : 9.00
## 1st Qu.:4.000    Class :character   Class :character   1st Qu.:14.00
## Median :6.000    Mode  :character   Mode  :character   Median :17.00
## Mean   :5.889                                     Mean   :16.86
## 3rd Qu.:8.000                                     3rd Qu.:19.00
## Max.    :8.000                                     Max.    :35.00
##      hwy      fl      class
## Min.    :12.00    Length:234    Length:234
## 1st Qu.:18.00    Class :character   Class :character
## Median :24.00    Mode  :character   Mode  :character
## Mean   :23.44
## 3rd Qu.:27.00
## Max.    :44.00
```

2. 데이터 수정하기 - 변수명 바꾸기

dplyr 패키지 설치 & 로드

```
install.packages("dplyr") # dplyr 설치  
library(dplyr)           # dplyr 로드
```

데이터 프레임 생성

```
df_raw <- data.frame(var1 = c(1, 2, 1),  
                     var2 = c(2, 3, 2))
```

```
df_raw
```

```
##      var1 var2  
## 1      1    2  
## 2      2    3  
## 3      1    2
```

1. 데이터 프레임 복사본 만들기

```
df_new <- df_raw # 복사본 생성  
df_new           # 출력
```

```
##   var1 var2  
## 1    1    2  
## 2    2    3  
## 3    1    2
```


2. 변수명 바꾸기

```
df_new <- rename(df_new, v2 = var2) # var2 를 v2 로 수정
```

```
df_new
```

```
##   var1 v2
```

```
## 1    1  2
```

```
## 2    2  3
```

```
## 3    1  2
```

[유의] rename()에 '새 변수명 = 기존 변수명' 순서로 입력

수정 전후 비교

df_raw

```
##      var1 var2
## 1      1    2
## 2      2    3
## 3      1    2
```

df_new

```
##      var1 v2
## 1      1  2
## 2      2  3
## 3      1  2
```

혼자서 해보기

mpg 데이터의 변수명은 긴 단어를 짧게 줄인 축약어로 되어있습니다. cty 변수는 도시 연비, hwy 변수는 고속도로 연비를 의미합니다. 변수명을 이해하기 쉬운 단어로 바꾸려고 합니다. mpg 데이터를 이용해서 아래 문제를 해결해 보세요.

- Q1. ggplot2 패키지의 mpg 데이터를 사용할 수 있도록 불러온 뒤 복사본을 만드세요.
- Q2. 복사본 데이터를 이용해서 cty는 city로, hwy는 highway로 변수명을 수정하세요.
- Q3. 데이터 일부를 출력해서 변수명이 바뀌었는지 확인해 보세요. 아래와 같은 결과물이 출력되어야 합니다.

```
##      manufacturer model displ year cyl      trans drv  city highway fl   class
## 1             audi   a4    1.8 1999   4    auto(l5)  f    18        29  p compact
## 2             audi   a4    1.8 1999   4 manual(m5)  f    21        29  p compact
## 3             audi   a4    2.0 2008   4 manual(m6)  f    20        31  p compact
## 4             audi   a4    2.0 2008   4    auto(av)  f    21        30  p compact
## 5             audi   a4    2.8 1999   6    auto(l5)  f    16        26  p compact
## 6             audi   a4    2.8 1999   6 manual(m5)  f    18        26  p compact
```

정답

Q1. ggplot2 패키지의 mpg 데이터를 사용할 수 있도록 불러온 뒤 복사본을 만드세요.

```
mpg <- as.data.frame(ggplot2::mpg)      # mpg 데이터 불러오기
mpg_new <- mpg                          # 복사본 만들기
```

Q2. 복사본 데이터를 이용해서 cty는 city로, hwy는 highway로 변수명을 수정하세요.

```
mpg_new <- rename(mpg_new, city = cty)   # cty를 city로 수정
mpg_new <- rename(mpg_new, highway = hwy) # hwy를 highway로 수정
```

Q3. 데이터 일부를 출력해서 변수명이 바뀌었는지 확인해 보세요. 아래와 같은 결과물이 출력되어야 합니다.

```
head(mpg_new)                                # 데이터 일부 출력
```

##	manufacturer	model	displ	year	cyl	trans	drv	city	highway	fl	class
## 1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
## 2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
## 3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
## 4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
## 5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
## 6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact

05-3. 파생변수 만들기

파생변수
↓

이름	영어 점수	수학 점수		이름	영어 점수	수학 점수	평균
김지훈	90	50	→	김지훈	90	50	70
이유진	80	60		이유진	80	60	70
박동현	60	100		박동현	60	100	80
김민지	70	20		김민지	70	20	45

변수 조합해 파생변수 만들기

데이터 프레임 생성

```
df <- data.frame(var1 = c(4, 3, 8),  
                 var2 = c(2, 6, 1))
```

```
df
```

```
##   var1 var2  
## 1    4    2  
## 2    3    6  
## 3    8    1
```

파생변수 생성

```
df$var_sum <- df$var1 + df$var2  # var_sum 파생변수 생성
```

```
df
```

```
##   var1 var2 var_sum
```

```
## 1    4    2      6
```

```
## 2    3    6      9
```

```
## 3    8    1      9
```

파생변수 생성

```
df$var_mean <- (df$var1 + df$var2)/2 # var_mean 파생변수 생성
```

```
df
```

```
##   var1 var2 var_sum var_mean
## 1    4    2      6    3.0
## 2    3    6      9    4.5
## 3    8    1      9    4.5
```


mpg 통합 연비 변수 만들기

```
mpg$total <- (mpg$cty + mpg$hwy)/2 # 통합 연비 변수 생성
```

```
head(mpg)
```

```
##      manufacturer model displ year cyl      trans drv  cty   hwy fl    class
## 1             audi   a4    1.8 1999   4    auto(l5)   f   18    29 p compact
## 2             audi   a4    1.8 1999   4 manual(m5)   f   21    29 p compact
## 3             audi   a4    2.0 2008   4 manual(m6)   f   20    31 p compact
## 4             audi   a4    2.0 2008   4    auto(av)   f   21    30 p compact
## 5             audi   a4    2.8 1999   6    auto(l5)   f   16    26 p compact
## 6             audi   a4    2.8 1999   6 manual(m5)   f   18    26 p compact
```

```
##      total
```

```
## 1    23.5
```

```
## 2    25.0
```

```
## 3    25.5
```

```
## 4    25.5
```

```
## 5    21.0
```

```
## 6    22.0
```

```
mean(mpg$total)
```

```
## [1] 20.14957
```

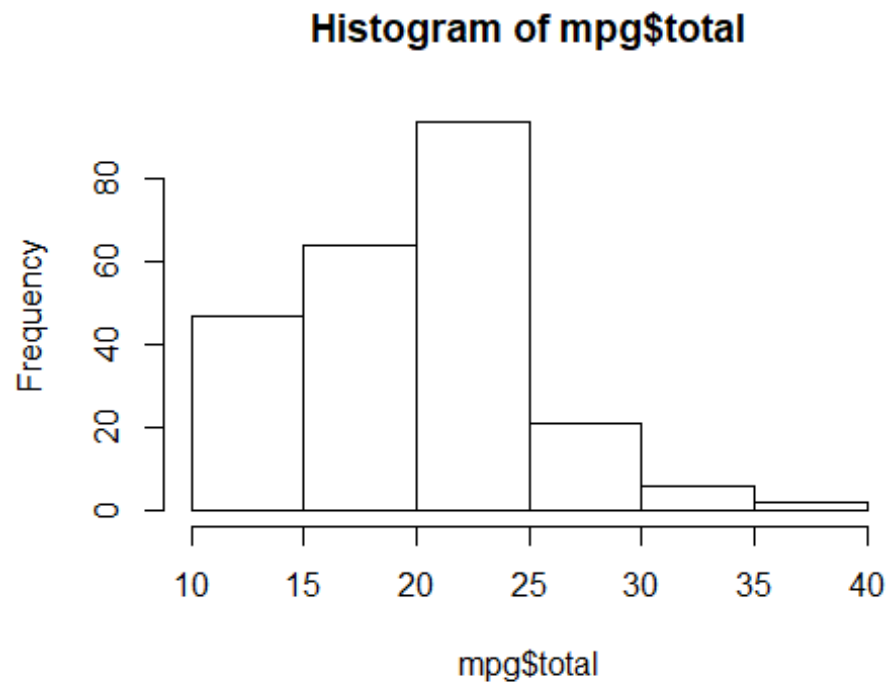
조건문을 활용해 파생변수 만들기

1.기준값 정하기

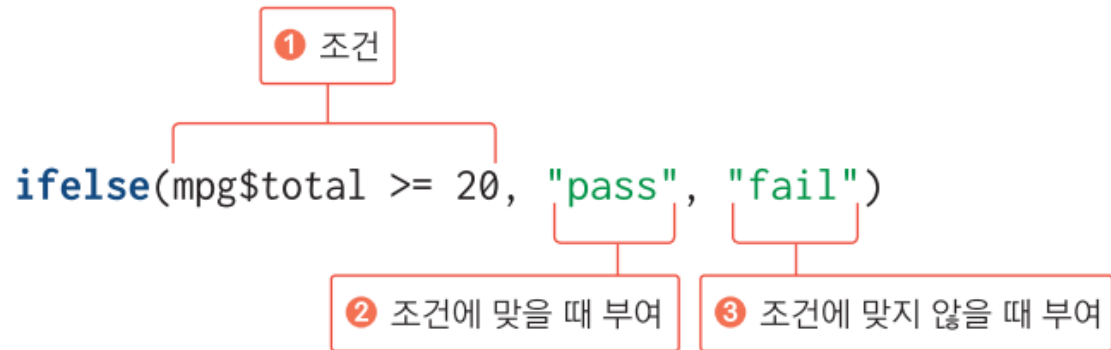
```
summary(mpg$total) # 요약 통계량 산출
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   
##    10.50  15.50   20.50   20.15  23.50   39.50
```

```
hist(mpg$total) # 히스토그램 생성
```



2. 조건문으로 합격 판정 변수 만들기



```
# 20 이상이면 pass, 그렇지 않으면 fail 부여  
mpg$test <- ifelse(mpg$total >= 20, "pass", "fail")
```

```
head(mpg, 20) # 데이터 확인
```

```
##      manufacturer      model displ year  cyl    trans  drv  cty  hwy
## 1          audi          a4    1.8 1999   4  auto(l5)   f   18   29
## 2          audi          a4    1.8 1999   4 manual(m5)   f   21   29
## 3          audi          a4    2.0 2008   4 manual(m6)   f   20   31
## 4          audi          a4    2.0 2008   4  auto(av)    f   21   30
## 5          audi          a4    2.8 1999   6  auto(l5)    f   16   26
## 6          audi          a4    2.8 1999   6 manual(m5)   f   18   26
## 7          audi          a4    3.1 2008   6  auto(av)    f   18   27
## 8          audi      a4 quattro  1.8 1999   4 manual(m5)   4   18   26
## 9          audi      a4 quattro  1.8 1999   4  auto(l5)    4   16   25
## 10         audi      a4 quattro  2.0 2008   4 manual(m6)   4   20   28
## 11         audi      a4 quattro  2.0 2008   4  auto(s6)    4   19   27
## 12         audi      a4 quattro  2.8 1999   6  auto(l5)    4   15   25
## 13         audi      a4 quattro  2.8 1999   6 manual(m5)   4   17   25
## 14         audi      a4 quattro  3.1 2008   6  auto(s6)    4   17   25
## 15         audi      a4 quattro  3.1 2008   6 manual(m6)   4   15   25
## 16         audi      a6 quattro  2.8 1999   6  auto(l5)    4   15   24
## 17         audi      a6 quattro  3.1 2008   6  auto(s6)    4   17   25
## 18         audi      a6 quattro  4.2 2008   8  auto(s6)    4   16   23
## 19  chevrolet c1500 suburban 2wd  5.3 2008   8  auto(l4)    r   14   20
## 20  chevrolet c1500 suburban 2wd  5.3 2008   8  auto(l4)    r   11   15
##      fl    class total test
## 1    p compact  23.5 pass
## 2    p compact  25.0 pass
## 3    p compact  25.5 pass
```

##	4	p	compact	25.5	pass
##	5	p	compact	21.0	pass
##	6	p	compact	22.0	pass
##	7	p	compact	22.5	pass
##	8	p	compact	22.0	pass
##	9	p	compact	20.5	pass
##	10	p	compact	24.0	pass
##	11	p	compact	23.0	pass
##	12	p	compact	20.0	pass
##	13	p	compact	21.0	pass
##	14	p	compact	21.0	pass
##	15	p	compact	20.0	pass
##	16	p	midsize	19.5	fail
##	17	p	midsize	21.0	pass
##	18	p	midsize	19.5	fail
##	19	r	suv	17.0	fail
##	20	e	suv	13.0	fail

3. 빈도표로 합격 판정 자동차 수 살펴보기

```
table(mpg$test)  # 연비 합격 빈도표 생성
```

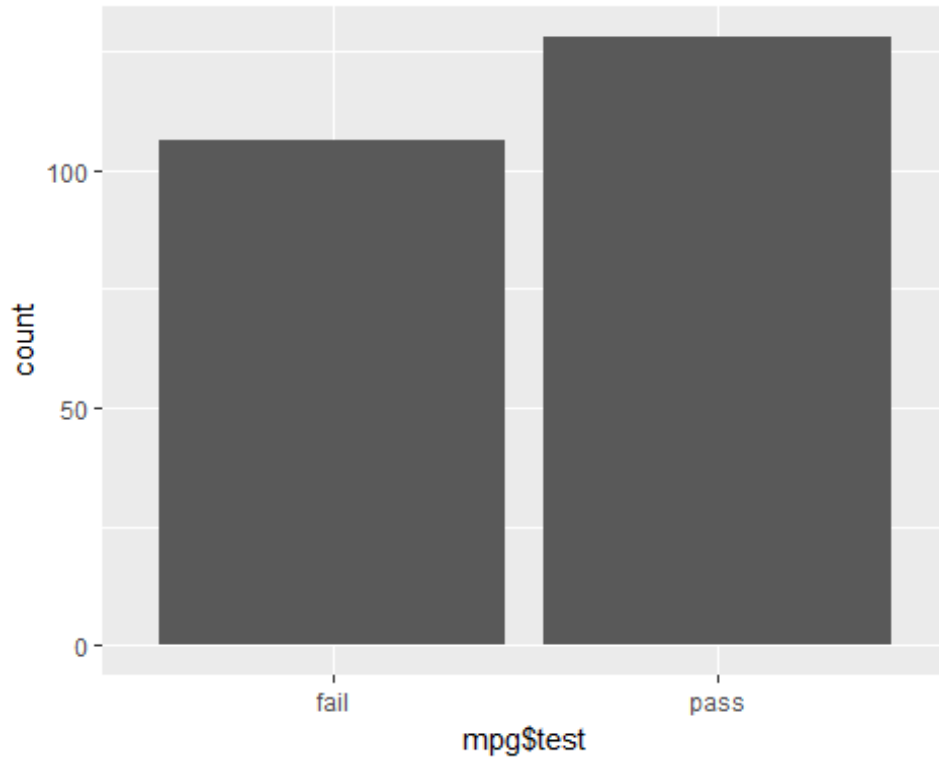
```
##
```

```
## fail pass
```

```
## 106 128
```

4. 막대 그래프 빈도 표현하기

```
library(ggplot2)  # ggplot2 로드  
qplot(mpg$test)   # 연비 합격 빈도 막대 그래프 생성
```



중첩 조건문 활용하기 - 연비 등급 변수 만들기

등급 total 기준

A 30 이상

B 20~29

C 20 미만

total 을 기준으로 A, B, C 등급 부여

```
mpg$grade <- ifelse(mpg$total >= 30, "A",  
                    ifelse(mpg$total >= 20, "B", "C"))
```

head(mpg, 20) # 데이터 확인

```
##      manufacturer      model displ  year  cyl      trans  drv  cty  hwy  
## 1      audi          a4    1.8 1999    4    auto(l5)   f   18   29  
## 2      audi          a4    1.8 1999    4 manual(m5)   f   21   29  
## 3      audi          a4    2.0 2008    4 manual(m6)   f   20   31  
## 4      audi          a4    2.0 2008    4    auto(av)   f   21   30  
## 5      audi          a4    2.8 1999    6    auto(l5)   f   16   26  
## 6      audi          a4    2.8 1999    6 manual(m5)   f   18   26  
## 7      audi          a4    3.1 2008    6    auto(av)   f   18   27  
## 8      audi      a4 quattro  1.8 1999    4 manual(m5)   4   18   26  
## 9      audi      a4 quattro  1.8 1999    4    auto(l5)   4   16   25  
## 10     audi      a4 quattro  2.0 2008    4 manual(m6)   4   20   28  
## 11     audi      a4 quattro  2.0 2008    4    auto(s6)   4   19   27  
## 12     audi      a4 quattro  2.8 1999    6    auto(l5)   4   15   25  
## 13     audi      a4 quattro  2.8 1999    6 manual(m5)   4   17   25  
## 14     audi      a4 quattro  3.1 2008    6    auto(s6)   4   17   25  
## 15     audi      a4 quattro  3.1 2008    6 manual(m6)   4   15   25  
## 16     audi      a6 quattro  2.8 1999    6    auto(l5)   4   15   24  
## 17     audi      a6 quattro  3.1 2008    6    auto(s6)   4   17   25  
## 18     audi      a6 quattro  4.2 2008    8    auto(s6)   4   16   23  
## 19  chevrolet c1500 suburban 2wd  5.3 2008    8    auto(l4)   r   14   20  
## 20  chevrolet c1500 suburban 2wd  5.3 2008    8    auto(l4)   r   11   15
```

##	fl	class	total	test	grade
## 1	p	compact	23.5	pass	B
## 2	p	compact	25.0	pass	B
## 3	p	compact	25.5	pass	B
## 4	p	compact	25.5	pass	B
## 5	p	compact	21.0	pass	B
## 6	p	compact	22.0	pass	B
## 7	p	compact	22.5	pass	B
## 8	p	compact	22.0	pass	B
## 9	p	compact	20.5	pass	B
## 10	p	compact	24.0	pass	B
## 11	p	compact	23.0	pass	B
## 12	p	compact	20.0	pass	B
## 13	p	compact	21.0	pass	B
## 14	p	compact	21.0	pass	B
## 15	p	compact	20.0	pass	B
## 16	p	midsize	19.5	fail	C
## 17	p	midsize	21.0	pass	B
## 18	p	midsize	19.5	fail	C
## 19	r	suv	17.0	fail	C
## 20	e	suv	13.0	fail	C

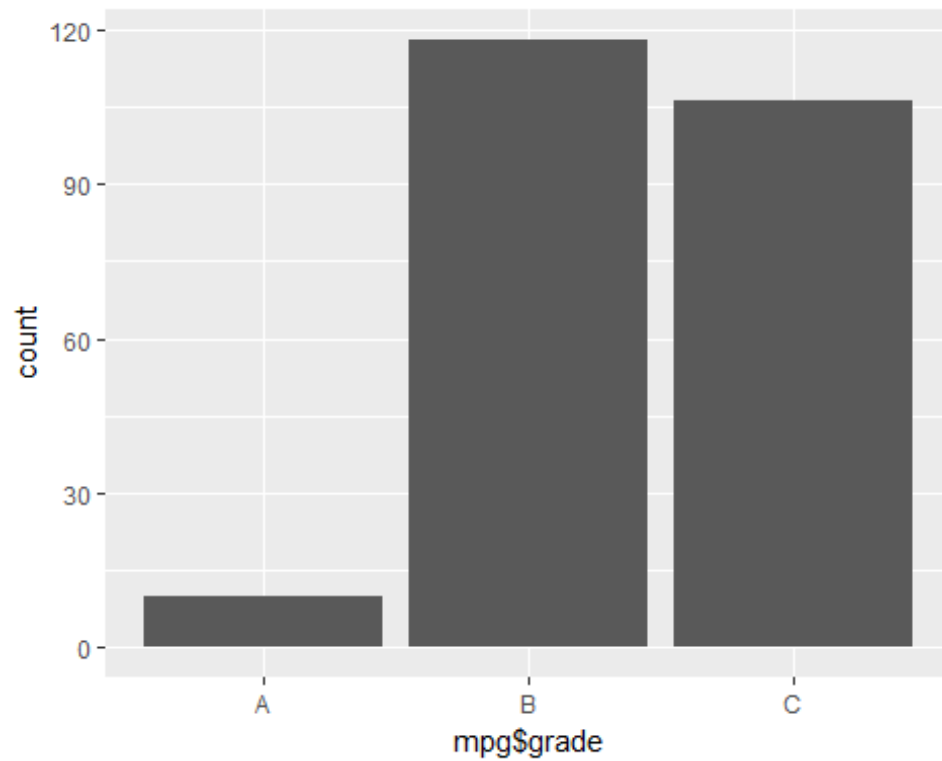
[유의] ifelse()가 두 번 반복되므로 열리는 괄호와 닫히는 괄호가 각각 두 개, 심표도 각각 두 개

빈도표, 막대 그래프로 연비 등급 살펴보기

```
table(mpg$grade) # 등급 빈도표 생성
```

```
##  
##   A    B    C  
##  10 118 106
```

```
qplot(mpg$grade) # 등급 빈도 막대 그래프 생성
```



원하는 만큼 범주 만들기

A, B, C, D 등급 부여

```
mpg$grade2 <- ifelse(mpg$total >= 30, "A",  
                     ifelse(mpg$total >= 25, "B",  
                             ifelse(mpg$total >= 20, "C", "D")))
```

정리하기

1. 데이터 준비, 패키지 준비

```
mpg <- as.data.frame(ggplot2::mpg)  # 데이터 불러오기
library(dplyr)                      # dplyr 로드
library(ggplot2)                    # ggplot2 로드
```

2. 데이터 파악

```
head(mpg)      # Raw 데이터 앞부분
tail(mpg)      # Raw 데이터 뒷부분
View(mpg)      # Raw 데이터 뷰어창에서 확인
dim(mpg)       # 차원
str(mpg)       # 속성
summary(mpg)   # 요약 통계량
```

정리하기

3. 변수명 수정

```
mpg <- rename(mpg, company = manufacturer)
```

4. 파생변수 생성

```
mpg$total <- (mpg$cty + mpg$hwy)/2 # 변수 조합
```

```
mpg$test <- ifelse(mpg$total >= 20, "pass", "fail") # 조건문 활용
```

5. 빈도 확인

```
table(mpg$test) # 빈도표 출력
```

```
qplot(mpg$test) # 막대 그래프 생성
```

분석 도전!

ggplot2 패키지에는 미국 동북중부 437개 지역의 인구통계 정보를 담은 `midwest`라는 데이터가 포함되어 있습니다. `midwest` 데이터를 사용해 데이터 분석 문제를 해결해보세요.

- 문제 1. `ggplot2` 의 `midwest` 데이터를 데이터 프레임 형태로 불러와서 데이터의 특성을 파악하세요.
- 문제 2. `poptotal`(전체 인구)을 `total` 로, `popasian`(아시아 인구)을 `asian` 으로 변수명을 수정하세요.
- 문제 3. `total`, `asian` 변수를 이용해 '전체 인구 대비 아시아 인구 백분율' 파생변수를 만들고, 히스토그램을 만들어 도시들이 어떻게 분포하는지 살펴보세요.
- 문제 4. 아시아 인구 백분율 전체 평균을 구하고, 평균을 초과하면 `"large"`, 그 외에는 `"small"`을 부여하는 파생변수를 만들어 보세요.
- 문제 5. `"large"`와 `"small"`에 해당하는 지역이 얼마나 되는지, 빈도표와 빈도 막대 그래프를 만들어 확인해 보세요.

분석 도전 정답

문제1. ggplot2의 `midwest` 데이터를 데이터 프레임 형태로 불러와서 데이터의 특성을 파악하세요.

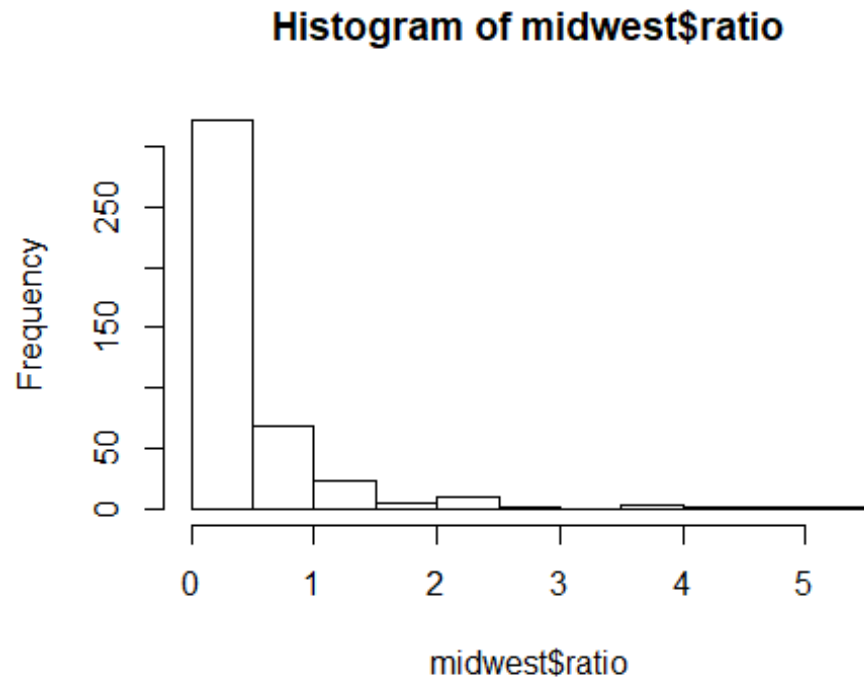
```
midwest <- as.data.frame(ggplot2::midwest)
head(midwest)
tail(midwest)
View(midwest)
dim(midwest)
str(midwest)
summary(midwest)
```


문제2. `poptotal`(전체 인구)을 `total`로, `popasian`(아시아 인구)을 `asian`으로 변수명을 수정하세요.

```
library(dplyr)
midwest <- rename(midwest, total = poptotal)
midwest <- rename(midwest, asian = popasian)
```

문제3. `total`, `asian` 변수를 이용해 '전체 인구 대비 아시아 인구 백분율' 파생변수를 만들고, 히스토그램을 만들어 도시들이 어떻게 분포하는지 살펴보세요.

```
midwest$ratio <- midwest$asian/midwest$total*100  
hist(midwest$ratio)
```



문제4. 아시아 인구 백분율 전체 평균을 구하고, 평균을 초과하면 "large", 그 외에는 "small"을 부여하는 파생변수를 만들어 보세요.

```
mean(midwest$ratio)
```

```
## [1] 0.4872462
```

```
midwest$group <- ifelse(midwest$ratio > 0.4872462, "large", "small")
```

문제5. "large"와 "small"에 해당하는 지역이 얼마나 되는지, 빈도표와 빈도 막대 그래프를 만들어 확인해 보세요.

```
table(midwest$group)
```

```
##
```

```
## large small
```

```
##    119    318
```

```
library(ggplot2)
```

```
qplot(midwest$group)
```

