Day31_2 R

상수 변수 생성

```
a <- 1
a
```

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

```
b <- 2
(a+b)/2
```

```
## [1] 1.5
```

벡터 만들기

```
v1 <- c(1,2,5,8,9)
v1
```

```
## [1] 1 2 5 8 9
```

- 연속된 숫자로 벡터를 만드는 방법

```
v2 <- c(1:5)
v2
```

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

```
v3 <- seq(1,5)
v3
```

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

```
v4 <- seq(1, 10, by=3)
v4
```

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

- 벡터 연산

```
v4+1
```

```
## [1]  2  5  8 11
```

- 문자열 벡터

```
s1 <- 'a'
s2 = 'text'
s3 = 'hi'
s4 = c(s1, s2, s3)
s4
```

```
## [1] "a"    "text" "hi"
```

```
s4+1
```

```
## Error in s4 + 1: 이항연산자에 수치가 아닌 인수입니다
```

- 상수형 벡터 기본 연산

```
mean(v1)
```

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

```
max(v1)
```

```
## [1] 9
```

```
min(v1)
```

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

- 문자형 벡터 연결

```
paste(s1, s2, s3)
```

```
## [1] "a text hi"
```

```
paste(s4)
```

```
## [1] "a"    "text" "hi"
```

```
paste(s4, collapse = ",")
```

```
## [1] "a,text,hi"
```

```
paste(s4, collapse = "*")
```

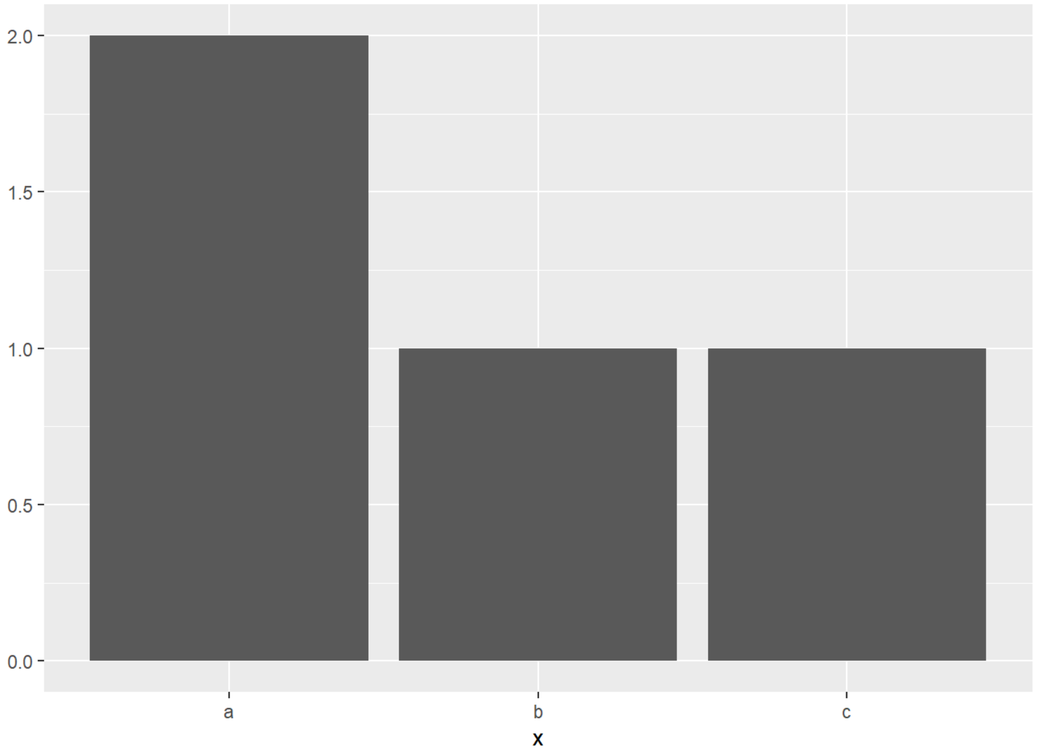
```
## [1] "a*text*hi"
```

시각화

```
# install.packages("ggplot2")
library(ggplot2)
```

- 빈도 그래프

```
x <- c("a", "a", "b", "c")
qplot(x) # 빈도 그래프
```



mpg : ggplot2에 있는 데이터 셋

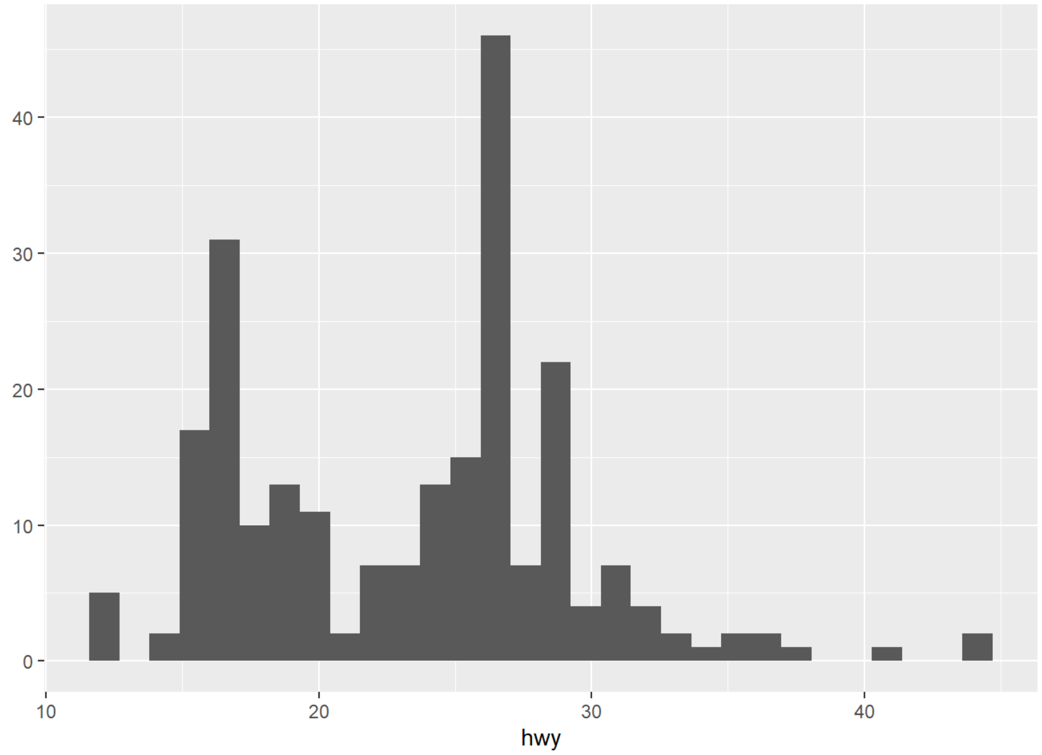
```
mpg # 기본적으로 설치되어 있는 데이터 셋
```

```
## # A tibble: 234 x 11
##    manufacturer model    displ  year   cyl trans   drv     cty   hwy fl    class
##    <chr>        <chr>    <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
##   1 audi         a4         1.8  1999     4 auto(l~ f        18    29 p     comp~
##   2 audi         a4         1.8  1999     4 manual~ f        21    29 p     comp~
##   3 audi         a4         2    2008     4 manual~ f        20    31 p     comp~
##   4 audi         a4         2    2008     4 auto(a~ f        21    30 p     comp~
##   5 audi         a4         2.8  1999     6 auto(l~ f        16    26 p     comp~
##   6 audi         a4         2.8  1999     6 manual~ f        18    26 p     comp~
##   7 audi         a4         3.1  2008     6 auto(a~ f        18    27 p     comp~
##   8 audi         a4 quat~   1.8  1999     4 manual~ 4        18    26 p     comp~
##   9 audi         a4 quat~   1.8  1999     4 auto(l~ 4        16    25 p     comp~
##  10 audi         a4 quat~   2    2008     4 manual~ 4        20    28 p     comp~
## # ... with 224 more rows
```

- 연비 빈도 그래프

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

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

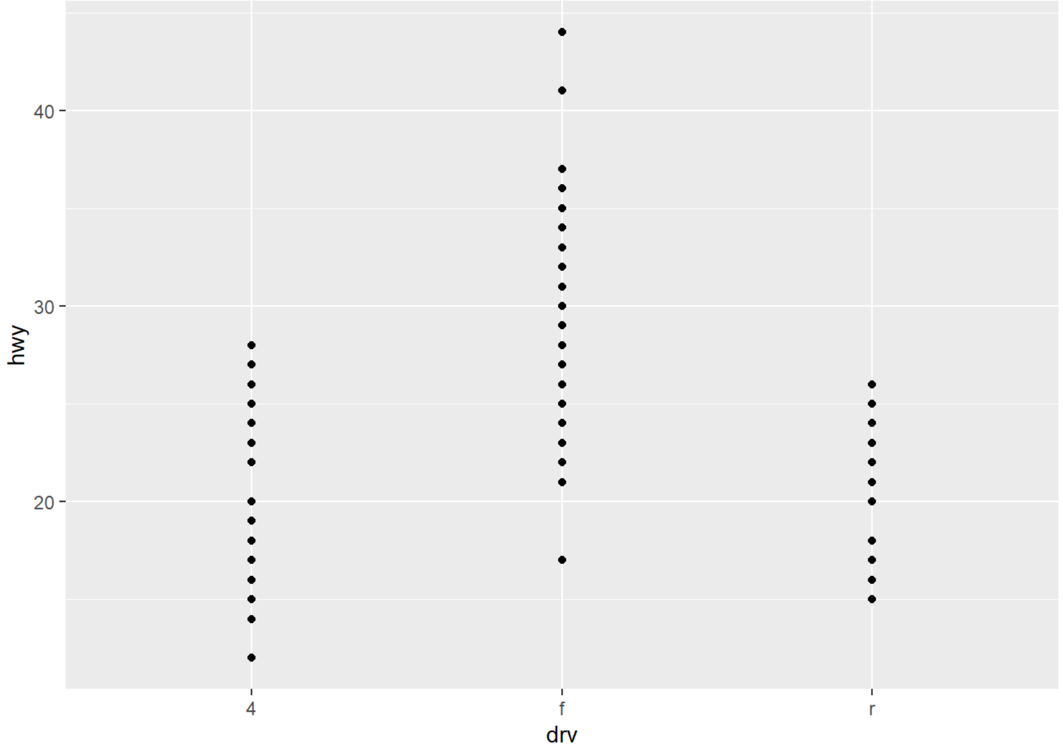


- 도움말

```
# help(mpg)
```

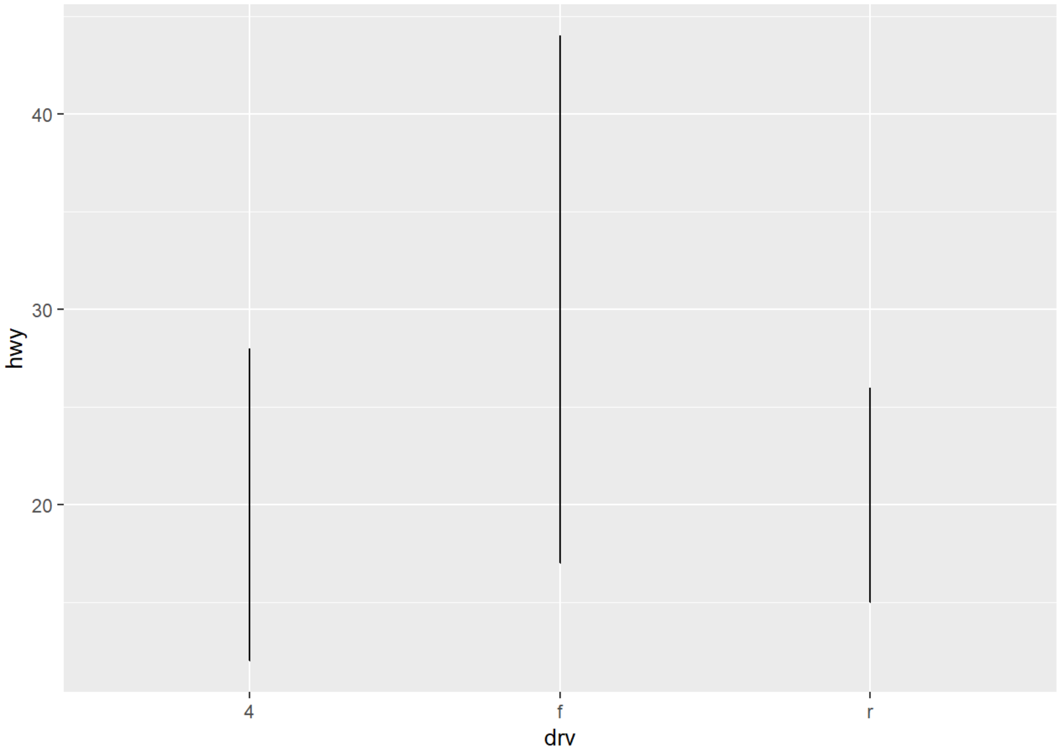
- 구동 방식에 따른 연비

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

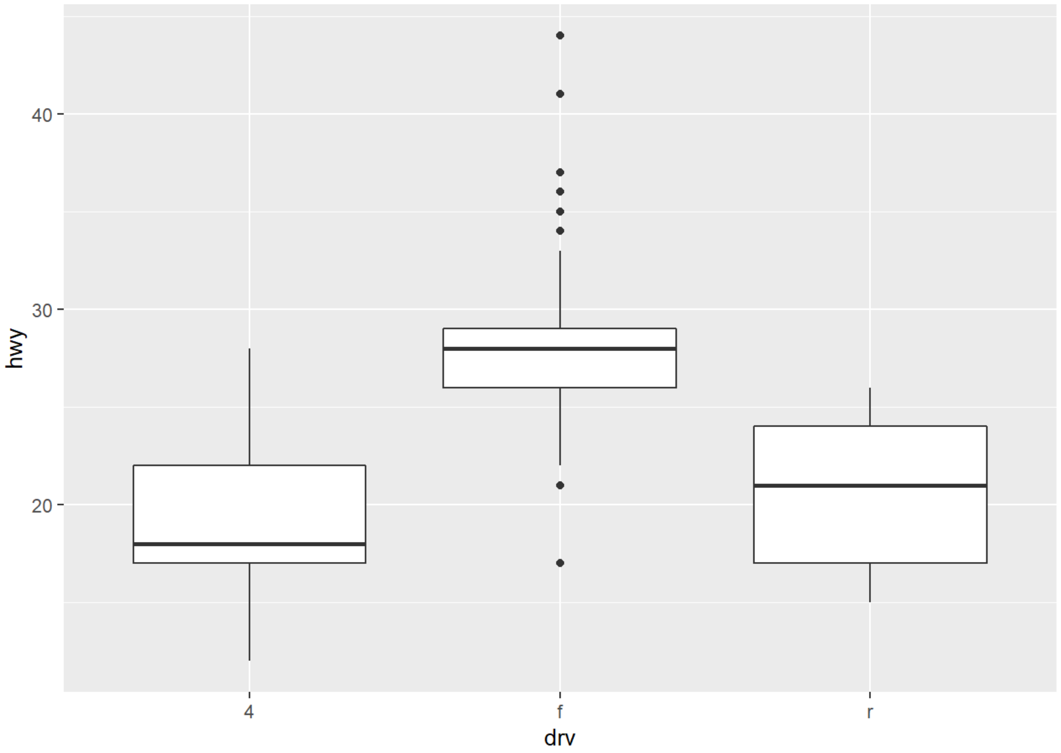


- 여러 방식으로 그래프 그리기

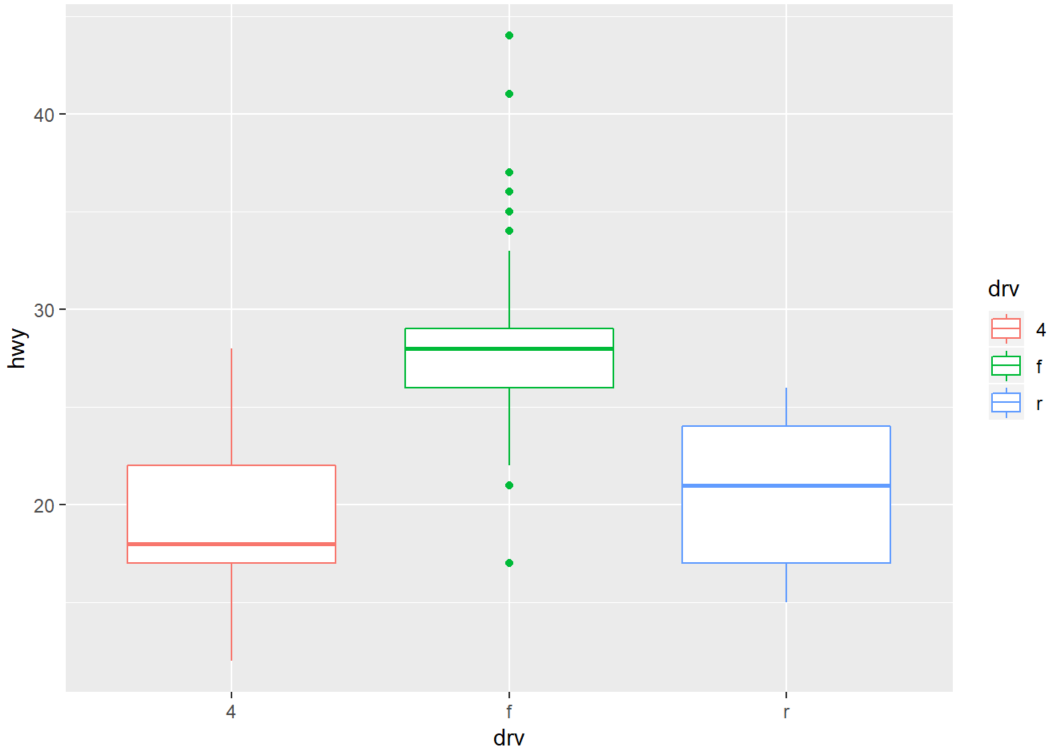
```
qplot(data=mpg, x=drv, y=hwy, geom = 'line')
```



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



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



data.frame

- 데이터 프레임 생성

```
eng = c(90, 80, 60, 70)
math = c(50, 10, 20, 90)

df_mid = data.frame(eng, math)
df_mid
```

```
##   eng math
## 1  90   50
## 2  80   10
## 3  60   20
## 4  70   90
```

- 데이터프레임 정보

```
str(df_mid)
```

```
## 'data.frame':    4 obs. of  2 variables:
##  $ eng : num  90 80 60 70
##  $ math: num  50 10 20 90
```

```
class = c(1,1,2,2)
df_mid = data.frame(eng, math, class)
df_mid
```

```
##   eng math class
## 1  90   50     1
## 2  80   10     1
## 3  60   20     2
## 4  70   90     2
```

- 데이터프레임 열 접근하기

```
df_mid$eng
```

```
## [1] 90 80 60 70
```

```
mean(df_mid$eng)
```

```
## [1] 75
```

- 변수 선언 없이 데이터 프레임 생성

```
df = data.frame(eng = c(90, 80, 60, 70),
math = c(50, 10, 20, 90),
class = c(1,1,2,2))
df
```

```
##   eng math class
## 1  90   50     1
## 2  80   10     1
## 3  60   20     2
## 4  70   90     2
```

r데이터분석_Data 폴더에 있는 파일 분석

```
# install.packages("readxl")
library(readxl)
```

- xlsx파일 불러오기

```
getwd()
```

```
## [1] "C:/Users/student/Desktop/R_script"
```

```
df = read_excel("r데이터분석_Data/Data/excel_exam.xlsx")
df
```

```
## # 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
```

```
df$english
```

```
##  [1] 98 97 86 98 80 89 90 78 98 98 65 85 98 87 56 98 68 78 68 83
```

- header가 없는 경우

```
novar_df = read_excel("r데이터분석_Data/Data/excel_exam_novar.xlsx", col_names = F)
```

```
## New names:
## * `` -> ...1
## * `` -> ...2
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
```

```
novar_df
```

```
## # A tibble: 8 x 5
##     ...1  ...2  ...3  ...4  ...5
##    <dbl> <dbl> <dbl> <dbl> <dbl>
## 1      1     1    50    98    50
## 2      2     1    60    97    60
## 3      3     2    25    80    65
## 4      4     2    50    89    98
## 5      5     3    20    98    15
## 6      6     3    50    98    45
## 7      7     4    46    98    65
## 8      8     4    48    87    12
```

- csv파일 불러오기

```
df = read.csv("r데이터분석_Data/Data/csv_exam.csv")
df
```

```
##    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
```

```
str(df)
```

```
## '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 ...
```

- 파일 저장하기

```
write.csv(df, file="mydf.csv")
```

- 다양한 방법으로 데이터 출력

```
exam = read.csv("r데이터분석_Data/Data/csv_exam.csv")
head(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
```

```
head(exam, 10)
```

```
##    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(exam, 10)
```

```
##    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(exam) # 창이 새로 생성되면서 data를 표 형태로 보여줌
```

```
## Error in view(exam): 함수 "view"를 찾을 수 없습니다
```

- 데이터 구조

```
dim(exam)
```

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

```
str(exam)
```

```
## '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 ...
```

- 데이터 기초통계

```
summary(exam)
```

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

데이터 프레임 컬럼 이름변경

```
df = data.frame(v1 = c(1,2,1), v2 = c(2,3,2))
df
```

```
##   v1 v2
## 1  1  2
## 2  2  3
## 3  1  2
```

- 컬럼이름 변경
  - rename함수가 ‘dplyr’ 패키지에 존재

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

```
## 
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
## 
##     filter, lag
```

```
## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union
```

- v1을 var1으로 변경

```
df = rename(df, var1=v1)
df
```

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

- 다른 방법

```
names(df) = c("v1", "var2")
df
```

```
##   v1 var2
## 1  1    2
## 2  2    3
## 3  1    2
```

- 열 생성

```
df$v_sum = df$v1 + df$var2
df
```

```
##   v1 var2 v_sum
## 1  1    2     3
## 2  2    3     5
## 3  1    2     3
```

mpg파일 분석

- 기본적인 정보 보기

```
str(mpg)
```

```
## Classes 'tbl_df', 'tbl' and '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" ...
```

```
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
```

```
view(mpg)
```

```
## Error in view(mpg): 함수 "view"를 찾을 수 없습니다
```

```
summary(mpg)
```

```
##  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                                        
```

- total 컬럼추가 = hwy, cty의 평균

    - cty : 도시에서의 연비
    - hwy : 고속도로에서의 연비

```
mpg$total = (mpg$cty + mpg$hwy)/2
mpg
```

```
## # A tibble: 234 x 12
##    manufacturer model displ  year   cyl trans drv     cty   hwy fl    class
##    <chr>        <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
##  1 audi         a4      1.8  1999     4 auto~ f        18    29 p     comp~
##  2 audi         a4      1.8  1999     4 manu~ f        21    29 p     comp~
##  3 audi         a4      2    2008     4 manu~ f        20    31 p     comp~
##  4 audi         a4      2    2008     4 auto~ f        21    30 p     comp~
##  5 audi         a4      2.8  1999     6 auto~ f        16    26 p     comp~
##  6 audi         a4      2.8  1999     6 manu~ f        18    26 p     comp~
##  7 audi         a4      3.1  2008     6 auto~ f        18    27 p     comp~
##  8 audi         a4 q~   1.8  1999     4 manu~ 4        18    26 p     comp~
##  9 audi         a4 q~   1.8  1999     4 auto~ 4        16    25 p     comp~
## 10 audi         a4 q~   2    2008     4 manu~ 4        20    28 p     comp~
## # ... with 224 more rows, and 1 more variable: total <dbl>
```

```
summary(mpg$total)
```

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

- 데이터 타입 변경

```
head(as.data.frame(mpg))
```

```
##   manufacturer model displ year cyl      trans drv cty hwy fl   class total
## 1         audi    a4   1.8 1999   4   auto(l5)   f  18  29  p compact  23.5
## 2         audi    a4   1.8 1999   4 manual(m5)   f  21  29  p compact  25.0
## 3         audi    a4   2.0 2008   4 manual(m6)   f  20  31  p compact  25.5
## 4         audi    a4   2.0 2008   4   auto(av)   f  21  30  p compact  25.5
## 5         audi    a4   2.8 1999   6   auto(l5)   f  16  26  p compact  21.0
## 6         audi    a4   2.8 1999   6 manual(m5)   f  18  26  p compact  22.0
```

```
str(as.data.frame(mpg))
```

```
## 'data.frame':    234 obs. of  12 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" ...
##  $ total       : num  23.5 25 25.5 25.5 21 22 22.5 22 20.5 24 ...
```

- 조건에 맞춰 열 생성하기

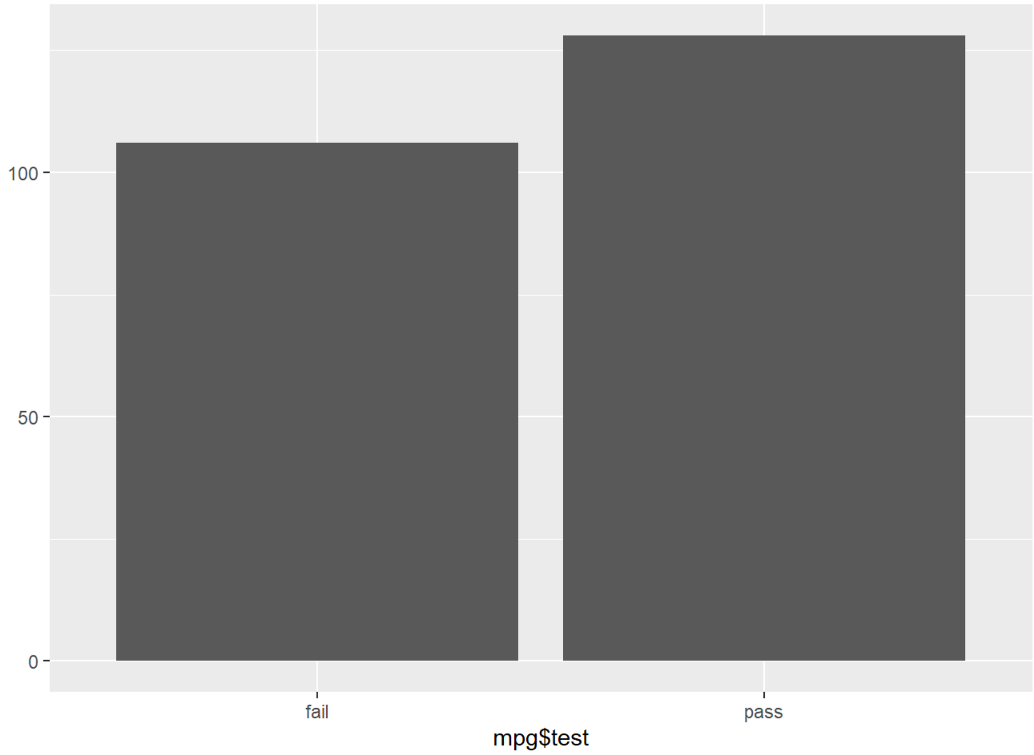
```
mpg = as.data.frame(mpg)

mpg$test = ifelse(mpg$total >= 20, "pass", "fail")
head(mpg, 20)
```

```
##    manufacturer              model displ year cyl      trans drv cty hwy fl
## 1          audi                 a4   1.8 1999   4   auto(l5)   f  18  29  p
## 2          audi                 a4   1.8 1999   4 manual(m5)   f  21  29  p
## 3          audi                 a4   2.0 2008   4 manual(m6)   f  20  31  p
## 4          audi                 a4   2.0 2008   4   auto(av)   f  21  30  p
## 5          audi                 a4   2.8 1999   6   auto(l5)   f  16  26  p
## 6          audi                 a4   2.8 1999   6 manual(m5)   f  18  26  p
## 7          audi                 a4   3.1 2008   6   auto(av)   f  18  27  p
## 8          audi         a4 quattro   1.8 1999   4 manual(m5)   4  18  26  p
## 9          audi         a4 quattro   1.8 1999   4   auto(l5)   4  16  25  p
## 10         audi         a4 quattro   2.0 2008   4 manual(m6)   4  20  28  p
## 11         audi         a4 quattro   2.0 2008   4   auto(s6)   4  19  27  p
## 12         audi         a4 quattro   2.8 1999   6   auto(l5)   4  15  25  p
## 13         audi         a4 quattro   2.8 1999   6 manual(m5)   4  17  25  p
## 14         audi         a4 quattro   3.1 2008   6   auto(s6)   4  17  25  p
## 15         audi         a4 quattro   3.1 2008   6 manual(m6)   4  15  25  p
## 16         audi         a6 quattro   2.8 1999   6   auto(l5)   4  15  24  p
## 17         audi         a6 quattro   3.1 2008   6   auto(s6)   4  17  25  p
## 18         audi         a6 quattro   4.2 2008   8   auto(s6)   4  16  23  p
## 19    chevrolet c1500 suburban 2wd   5.3 2008   8   auto(l4)   r  14  20  r
## 20    chevrolet c1500 suburban 2wd   5.3 2008   8   auto(l4)   r  11  15  e
##      class total test
## 1  compact  23.5 pass
## 2  compact  25.0 pass
## 3  compact  25.5 pass
## 4  compact  25.5 pass
## 5  compact  21.0 pass
## 6  compact  22.0 pass
## 7  compact  22.5 pass
## 8  compact  22.0 pass
## 9  compact  20.5 pass
## 10 compact  24.0 pass
## 11 compact  23.0 pass
## 12 compact  20.0 pass
## 13 compact  21.0 pass
## 14 compact  21.0 pass
## 15 compact  20.0 pass
## 16 midsize  19.5 fail
## 17 midsize  21.0 pass
## 18 midsize  19.5 fail
## 19     suv  17.0 fail
## 20     suv  13.0 fail
```

```
table(mpg$test)
```

```
## 
## fail pass 
##  106  128
```

```
qplot(mpg$test)
```



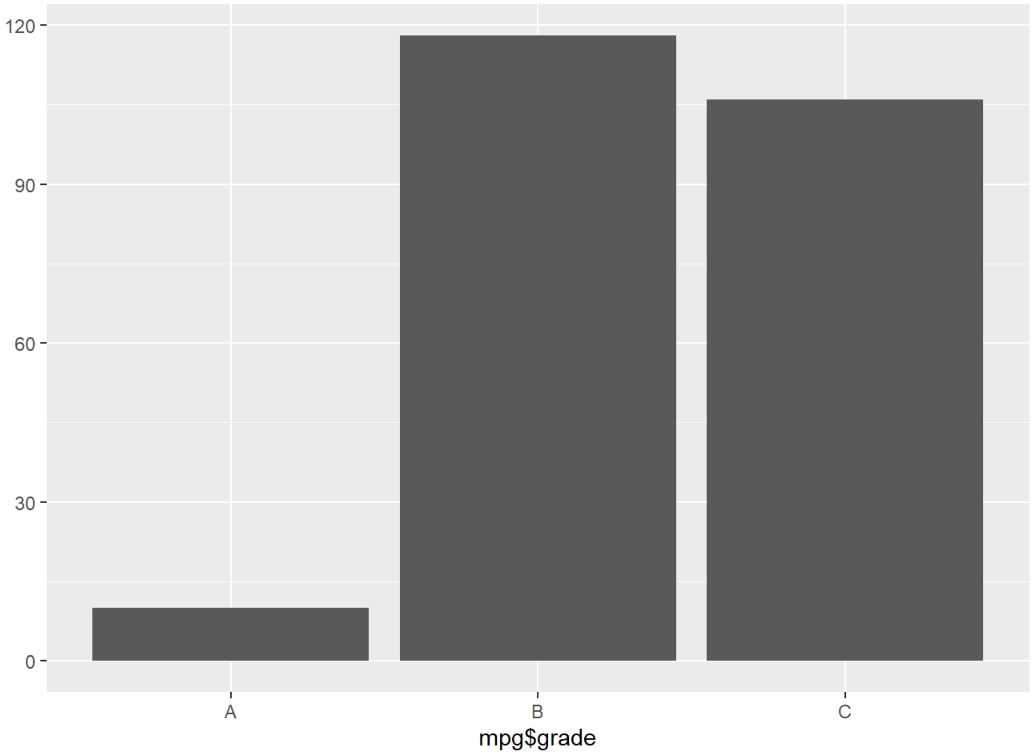
```
mpg$grade = ifelse(mpg$total >= 30, "A", ifelse(mpg$total >= 20, "B", "C"))
head(mpg)
```

```
##   manufacturer model displ year cyl      trans drv cty hwy fl   class total
## 1         audi    a4   1.8 1999   4   auto(l5)   f  18  29  p compact  23.5
## 2         audi    a4   1.8 1999   4 manual(m5)   f  21  29  p compact  25.0
## 3         audi    a4   2.0 2008   4 manual(m6)   f  20  31  p compact  25.5
## 4         audi    a4   2.0 2008   4   auto(av)   f  21  30  p compact  25.5
## 5         audi    a4   2.8 1999   6   auto(l5)   f  16  26  p compact  21.0
## 6         audi    a4   2.8 1999   6 manual(m5)   f  18  26  p compact  22.0
##   test grade
## 1 pass     B
## 2 pass     B
## 3 pass     B
## 4 pass     B
## 5 pass     B
## 6 pass     B
```

```
table(mpg$grade)
```

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

```
qplot(mpg$grade)
```



파이프라인(단축키 : ctrl + shift + ‘M’)

```
exam = read.csv("r데이터분석_Data/Data/csv_exam.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
```

- exam에서 class가 1인것만 추출

```
exam %>% filter(class == 1)
```

```
##   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
```

- exam에서 class==1인 것만 추출한 것에서 math가 50이상인 것만 추출

```
exam %>% filter(class == 1) %>% filter(math >= 50)
```

```
##   id class math english science
## 1  1     1   50      98      50
## 2  2     1   60      97      60
```

- exam에서 class가 1이아니고, math가 50 이상인 데이터 추출

```
exam %>% filter(class != 1) %>% filter(math >= 50)
```

```
##    id class math english science
## 1   6     2   50      89      98
## 2   7     2   80      90      45
## 3   8     2   90      78      25
## 4  10     3   50      98      45
## 5  11     3   65      65      65
## 6  15     4   75      56      78
## 7  16     4   58      98      65
## 8  17     5   65      68      98
## 9  18     5   80      78      90
## 10 19     5   89      68      87
## 11 20     5   78      83      58
```

- class 2이면서, 영어점수가 80점 이상인 데이터 추출

```
exam %>% filter(class == 2) %>%  filter(english >= 80)
```

```
##   id class math english science
## 1  5     2   25      80      65
## 2  6     2   50      89      98
## 3  7     2   80      90      45
```

```
exam %>% filter(class == 2 & english >= 80)
```

```
##   id class math english science
## 1  5     2   25      80      65
## 2  6     2   50      89      98
## 3  7     2   80      90      45
```

```
exam %>% filter(class == 2 | english >= 80)
```

```
##    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 12     3   45      85      32
## 12 13     4   46      98      65
## 13 14     4   48      87      12
## 14 16     4   58      98      65
## 15 20     5   78      83      58
```

- class 1 or 3, 5 추출

```
exam %>% filter(class != 2 & class != 4)
```

```
##    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   9     3   20      98      15
## 6  10     3   50      98      45
## 7  11     3   65      65      65
## 8  12     3   45      85      32
## 9  17     5   65      68      98
## 10 18     5   80      78      90
## 11 19     5   89      68      87
## 12 20     5   78      83      58
```

```
exam %>% filter(class == 1 | class == 3 | class == 5)
```

```
##    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   9     3   20      98      15
## 6  10     3   50      98      45
## 7  11     3   65      65      65
## 8  12     3   45      85      32
## 9  17     5   65      68      98
## 10 18     5   80      78      90
## 11 19     5   89      68      87
## 12 20     5   78      83      58
```

```
exam %>% filter(class %in% c(1,3,5))
```

```
##    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   9     3   20      98      15
## 6  10     3   50      98      45
## 7  11     3   65      65      65
## 8  12     3   45      85      32
## 9  17     5   65      68      98
## 10 18     5   80      78      90
## 11 19     5   89      68      87
## 12 20     5   78      83      58
```

- 특정열 추출하기

```
exam$math
```

```
##  [1] 50 60 45 30 25 50 80 90 20 50 65 45 46 48 75 58 65 80 89 78
```

```
exam %>% select(math) # dataframe 형식을 추출
```

```
##    math
## 1    50
## 2    60
## 3    45
## 4    30
## 5    25
## 6    50
## 7    80
## 8    90
## 9    20
## 10   50
## 11   65
## 12   45
## 13   46
## 14   48
## 15   75
## 16   58
## 17   65
## 18   80
## 19   89
## 20   78
```

```
exam %>%  select(-math, -class)
```

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

- class가 1인 행에 대해 english 추출

```
exam %>%
  filter(class == 1) %>%
  select(english)
```

```
##   english
## 1      98
## 2      97
## 3      86
## 4      98
```

- id와 math만 추출(6행 까지만)

```
exam %>%
  select(id, math) %>%
  head(6)
```

```
##   id math
## 1  1   50
## 2  2   60
## 3  3   45
## 4  4   30
## 5  5   25
## 6  6   50
```

- math를 기준으로 오름차순 정렬

```
exam %>% arrange(math)
```

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

- 내림차순 정렬

```
exam %>% arrange(desc(math))
```

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

- class를 기준으로 오름차순 정렬을 한 후, math로 오름차순 정렬

```
exam %>% arrange(class, math)
```

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

- class로 오름차순 정렬 후, math로 내림차순 정렬

```
exam %>% arrange(class, desc(math))
```

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

파생변수

```
exam %>%
  mutate(total = math + english + science) %>%
  head
```

```
##   id class math english science total
## 1  1     1   50      98      50   198
## 2  2     1   60      97      60   217
## 3  3     1   45      86      78   209
## 4  4     1   30      98      58   186
## 5  5     2   25      80      65   170
## 6  6     2   50      89      98   237
```

- science가 60점 이상 pass, 미만이면 fail
- test열 추가(ifelse)

```
exam %>%
  mutate(total = ifelse(science >= 60, "pass", "fail")) %>%
  head
```

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

- total = math + english + science
- 오름차순 정렬 하여 상위 10개 출력

```
exam %>%
  mutate(total = math + english + science) %>%
  arrange(total) %>%
  head(10)
```

```
##    id class math english science total
## 1   9     3   20      98      15   133
## 2  14     4   48      87      12   147
## 3  12     3   45      85      32   162
## 4   5     2   25      80      65   170
## 5   4     1   30      98      58   186
## 6   8     2   90      78      25   193
## 7  10     3   50      98      45   193
## 8  11     3   65      65      65   195
## 9   1     1   50      98      50   198
## 10  3     1   45      86      78   209
```