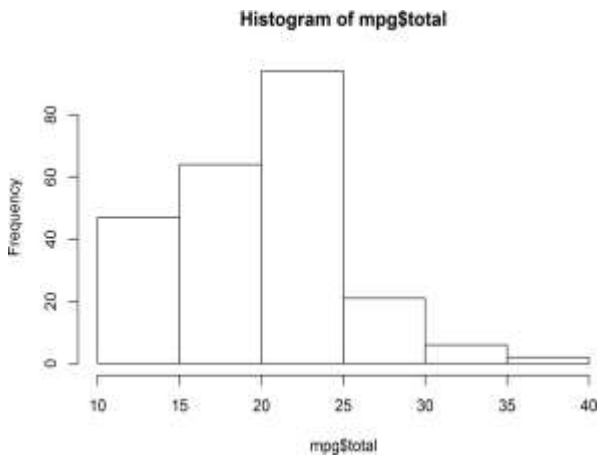
데이터 파악하기
데이터 추출하기
데이터 수정하기
데이터 병합하기

Chapter 5

데이터 파악하기

데이터프레임 함수

함수	기능
head()	데이터 앞부분 출력
tail()	데이터 뒷부분 출력
View()	뷰어 창에서 데이터 확인
str()	데이터 속성 출력
dim()	데이터 차원 출력
nrow()	데이터 행 출력
ncol()	데이터 열 출력
summary()	요약 통계량 출력
rownames()	행이름
colnames()	열이름



데이터프레임 함수

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
```

데이터프레임 함수

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
```

View() - 뷰어 창에서 데이터 확인하기

```
View(exam)
```

[유의] View()에서 맨 앞의 V는 대문자

데이터프레임 함수

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 ...
```

dim() - 몇 행 몇 열로 구성되는지 알아보기

```
dim(exam)  # 행, 열 출력
## [1] 20  5
```

데이터프레임 함수

nrow() – 몇 행으로 구성되는지 알아보기

```
nrow(exam) # 행 출력
## [1] 20
```

ncol() – 몇 열로 구성되는지 알아보기

```
ncol(exam) # 열 출력
## [1] 5
```

colnames() – 열 이름 확인

```
colnames(exam)  # 열 이름
## [1] id class math english science
```

Chapter 5

데이터 추출하기

데이터 추출

데이터 준비하기

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

```
   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
```

행 데이터 추출

행 번호로 행 추출하기

대괄호안 쉼표 기준, 왼쪽에 행 번호(인덱스) 입력

- 인덱스(Index) : 데이터의 위치 또는 순서를 의미하는 값
- 인덱싱(Indexing) : 인덱스를 이용해 데이터를 추출하는 작업

```
exam[1,]  # 1 행 추출

##   id class math english science
## 1  1     1   50      98      50

exam[2,]  # 2 행 추출

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

```
exam[1:3,] # 행, 변수 모두 인덱스

##   id class math english science
## 1  1     1   50      98      50
## 2  2     1   60      97      60
## 3  3     1   45      86      78
```

열 데이터 추출

열 번호로 변수 추출하기

대괄호안 쉼표 오른쪽에 조건을 입력

```
exam[,1]  # 첫 번째 열 추출

##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20

exam[,2]  # 두 번째 열 추출

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

exam[,3]  # 세 번째 열 추출

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

변수명으로 변수 추출하기

```
exam[, "class"]  # class 변수 추출

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

exam[, "math"]   # math 변수 추출

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

exam[,c("class", "math", "english")]  # class, math, english 변수 추출
```

행 열 데이터 추출

행, 변수 동시 추출하기

```
exam[1, 3] # 행, 변수 모두 인덱스

## [1] 50

exam[1:3, 2:3] # 1-3행, 2-3열

##
 class  math
1     1   50
2     1   60
3     1   45

exam[c(1,3) , c(3, 5)] # 1,3행, 3,5열

##
math   science
1   50      50
3   45      78
```

조건에 만족하는 행 추출

조건을 충족하는 행 추출하기

```
exam[exam$class == 1,]  # class가 1인 행 추출
```

```
##   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
```

```
exam[exam$math >= 80,]  # 수학점수가 80 점 이상인 행 추출
```

```
##    id class math english science
## 7   7     2   80      90      45
## 8   8     2   90      78      25
## 18 18     5   80      78      90
## 19 19     5   89      68      87
```

```
exam[exam$class == 1 & exam$math >= 50,] # 1반 이면서 수학점수가 50점 이상
```

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

문제

Q1. `data.frame()`과 `c()`를 조합해서 표의 내용을 데이터 프레임으로 만들어 출력해보세요.

제품	가격	판매량
사과	1800	24
딸기	1500	38
수박	3000	13

Q2. 앞에서 만든 데이터 프레임을 이용해서 과일 가격 합계 평균,
판매량 합계, 평균을 구해보세요.

Chapter 5

데이터 수정하기

행 데이터 수정하기

데이터 준비하기

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

데이터 행 수정하기

```
exam[1, ] <- 100
head(exam)

##
   id class math english science
1 100   100  100     100     100
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

exam[c(1,3,5), ] <- 100
##
 id class math english science
1 100   100  100     100     100
2   2     1   60      97      60
3 100   100  100     100     100
4   4     1   30      98      58
5 100   100  100     100     100
6   6     2   50      89      98
```

열 데이터 추가하기

데이터 열 추가하기

```
exam[, 6] <- "하이?"
head(exam)

##
   id class math english science    V6
1 100   100  100     100     100 하이?
2   2     1   60      97      60 하이?
3 100   100  100     100     100 하이?
4   4     1   30      98      58 하이?
5 100   100  100     100     100 하이?
6   6     2   50      89      98 하이?

exam[, ncol(exam) + 1 ] <- "hello" #마지막 컬럼을+1을 이용해서 추가하기
##
   id class math english science    V6    V7
1 100   100  100     100     100 하이? hello
2   2     1   60      97      60 하이? hello
3 100   100  100     100     100 하이? hello
4   4     1   30      98      58 하이? hello
5 100   100  100     100     100 하이? hello
6   6     2   50      89      98 하이? hello
```

열 데이터 추가하기

데이터 열 이름 추가하기

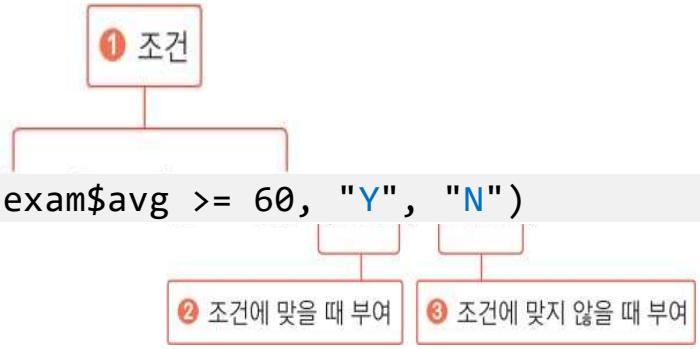
```
exam[, "xxx"] <- "asdasd" #컬럼명을 지정해서 추가하기
head(exam)

##
   id class math english science    V6    V7    xxx
1 100   100  100     100     100 하이? hello asdasd
2   2     1   60      97      60 하이? hello asdasd
3 100   100  100     100     100 하이? hello asdasd
4   4     1   30      98      58 하이? hello asdasd
5 100   100  100     100     100 하이? hello asdasd
6   6     2   50      89      98 하이? hello asdasd
```

컬렴명 수정

colnames() 를 이용해서....

조건에 따른 열 데이터 추가하기




```
ifelse(exam$avg >= 60, "Y", "N")
```

```
# 평균이 60이상이면 Y, 그렇지 않으면 N
exam$pass_fail <- ifelse(exam$avg >= 60, "Y", "N")
```

Chapter 5

데이터 병합과 빈도수

데이터 병합과 빈도수 확인

함수	기능
cbind()	컬럼을 합칩니다
rbind()	row를 합칩니다
table()	컬럼의 빈도수를 구합니다
hist()	빈도에 대해 간략한 histogram을 만듭니다

데이터 병합

```
# 데이터 준비
temp_mpg <- as.data.frame(mpg)
temp_mpg

# 행 열 슬라이싱
aaa <- temp_mpg[1:3, 1:5]
aaa

##
  manufacturer model displ year cyl
1         audi    a4   1.8 1999   4
2         audi    a4   1.8 1999   4
3         audi    a4   2.0 2008   4

# 행 열 슬라이싱
bbb <- temp_mpg[9:11, 1:5]
bbb

##
   manufacturer      model displ year cyl
9          audi a4 quattro   1.8 1999   4
10         audi a4 quattro   2.0 2008   4
11         audi a4 quattro   2.0 2008   4
```

데이터 병합

```
# 컬럼병합
cbind(aaa, bbb)

##
  manufacturer model displ year cyl manufacturer      model displ year cyl
1         audi    a4   1.8 1999   4         audi a4 quattro   1.8 1999   4
2         audi    a4   1.8 1999   4         audi a4 quattro   2.0 2008   4
3         audi    a4   2.0 2008   4         audi a4 quattro   2.0 2008   4

# 로우병합
rbind(aaa, bbb)

##
   manufacturer      model displ year cyl
1          audi         a4   1.8 1999   4
2          audi         a4   1.8 1999   4
3          audi         a4   2.0 2008   4
9          audi a4 quattro   1.8 1999   4
10         audi a4 quattro   2.0 2008   4
11         audi a4 quattro   2.0 2008   4
```

문제2

mpg 데이터 cty는 도시연비, hwy 변수는 고속도로 연비를 의미합니다. 변수명을 이해하기 쉬운 단어로 바꾸려고 합니다.
mpg 데이터를 이용해서 아래 문제를 해결해 보세요.

- Q1. ggplot2 패키지의 mpg데이터를 사용할 수 있도록 불러온 뒤 복사본을 만드세요.
- Q2. 복사본 데이터를 이용해서 cty는 city로, hwy는 highway로 변수명을 수정하세요.
- Q3. 복사본 데이터를 이용해서 cty + hwy / 2의 total컬럼을 만드세요.
- Q4. 컬럼 total의 평균을 출력하세요.
- Q5. total에 따른 파생변수 test를 생성합니다. (조건: total >= 20 이상 PASS 나머지 FAIL)
- Q6. total에 따른 파생변수 grade를 생성합니다. (24이상 A, 20이상 B, 나머지는 C)
- Q7. 데이터 일부를 출력해서 변수명이 바뀌었는지 확인해 보세요. 아래와 같은 결과물이 출력되어야 합니다.

	manufacturer	model	displ	year	cyl	trans	drv	city	highway	fl	class	total	test	grade
	<chr>	<chr>	<dbl>	<int>	<int>	<chr>	<chr>	<int>	<int>	<chr>	<chr>	<dbl>	<chr>	<chr>
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	comp~	23.5	pass	B
2	audi	a4	1.8	1999	4	manual(~	f	21	29	p	comp~	25	pass	A
3	audi	a4	2	2008	4	manual(~	f	20	31	p	comp~	25.5	pass	A
4	audi	a4	2	2008	4	auto(av)	f	21	30	p	comp~	25.5	pass	A
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	comp~	21	pass	B
6	audi	a4	2.8	1999	6	manual(~	f	18	26	p	comp~	22	pass	B

문제3

ggplot2 패키지에는 미국 동북중부 437개 지역의 인구통계 정보를 담은 midwest라는 데이터가 포함되어 있습니다. midwest 데이터를 사용해 데이터 분석 문제를해결해보세요.

- Q01. ggplot2 의 Midwest 데이터를 데이터 프레임 형태로 불러와서 데이터의 (구조, 끝부분, 뷰 창, 차원, 요약)을 파악하세요.
- Q02. poptotal(전체 인구)을 total 로, popasian(아시아 인구)을 asian 으로 변수명을 수정하세요.
- Q03. total, asian 변수를 이용해 '전체 인구 대비 아시아 인구 백분율' 파생변수를 만들고, 히스토그램을 만들어 도시들이 어떻게 분포하는지 살펴보세요.
- Q04. 아시아 인구 백분율 전체 평균을 구하고, 평균을 초과하면 "large", 그 외에는 "small"을 부여하는 파생변수(group)을 만들어 보세요

```
large small
  119   318
```

- Q05. group의 빈도수를 확인하세요.

소제목

Chapter 5

수고하셨습니다