R 데이터 분석 입문

Chapter 10

데이터 전처리

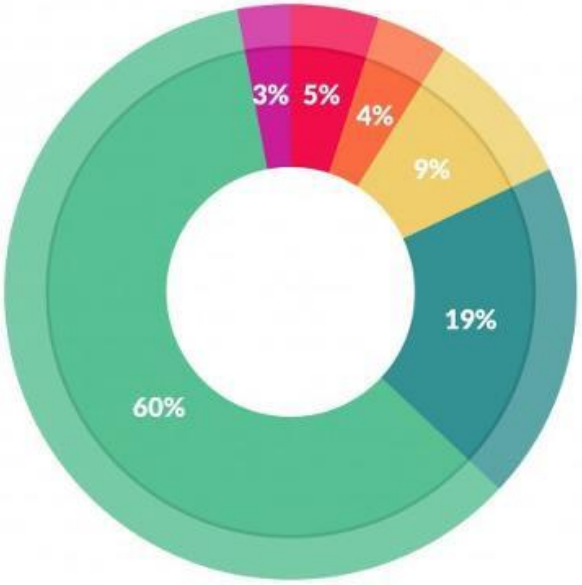
오 세 종

DANKOOK UNIVERSITY

Contents

개요

- 분석을 위한 데이터셋을 확보했다 하더라도 바로 분석을 할 수 없는 경우가 대부분
- 결측값, 이상치, 상이한 단위, 오입력, ...
- 데이터 분석에 적합하도록 데이터셋을 정제해야 하는데 이를 전처리 (data preprocessing) 라고 한다.
- 실제 데이터를 분석하는 시간보다 전처리에 더 많은 시간이 소요되는 경우가 많다.
- 효율적으로 데이터를 전처리 할 수 있는 능력이 중요함



What data scientists spend the most time doing

- *Building training sets: 3%*
- *Cleaning and organizing data: 60%*
- *Collecting data sets; 19%*
- *Mining data for patterns: 9%*
- *Refining algorithms: 4%*
- *Other: 5%*

https://towardsdatascience.com/data-preprocessing-for-non-techies-basic-terms-and-definitions-ea517038a4e5

1. 결측값(missing value)

- 결측값: 데이터셋에서 입력이 누락된 값
- NA 로 표시됨
- 결측값이 포함된 경우 산술 연산에 문제가 생길 수 있음

```
x <- c(1,2,3,NA,5,8)
sum(x)
sum(x, na.rm=T)    # 결측값을 제외하고 연산
```

```
> x <- c(1,2,3,NA,5,8)
> sum(x)
[1] NA
> sum(x, na.rm=T)
[1] 19
```

- 대부분 산술 연산 함수는 결측값 제외 옵션을 제공

1. 결측값(missing value)

- Vector 에 NA 가 몇 개나 포함되어 있는지 확인

```
z <- c(1,2,3,NA,5,NA,8)
is.na(z)
sum(is.na(z))
```

```
> z <- c(1,2,3,NA,5,NA,8)
> is.na(z)
[1] FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE
> sum(is.na(z))
[1] 2
```

- NA 를 0 으로 치환

```
z[is.na(z)] <- 0
z
```

```
> z[is.na(z)] <- 0
> z
[1] 1 2 3 0 5 0 8
```

1. 결측값 (missing value)

- Vector 에 포함된 NA 제거

```
x <- c(1,2,3,NA,5,8)
x
y <- as.vector(na.omit(x)) # vector 에서 NA 제거
y
```

```
> x <- c(1,2,3,NA,5,8)
> x
[1]  1  2  3 NA  5  8
> y <- as.vector(na.omit(x)) # vector 에서 NA 제거
> y
[1] 1 2 3 5 8
```

1. 결측값 (missing value)

- 2차원 배열의 NA

```
# NA 를 포함하는 test 데이터 생성
x <- iris
x[1,2]<- NA; x[1,3]<- NA
x[2,3]<- NA; x[3,4]<- NA
head(x)
```

```
> head(x)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1          NA           NA         0.2  setosa
2          4.9         3.0           NA         0.2  setosa
3          4.7         3.2          1.3          NA  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
>
```

1. 결측값 (missing value)

- 각 컬럼별로 결측값이 몇 개 있는지 확인

```
col_na <- function(y) {
  return(sum(is.na(y)))
}
na_count <-sapply(x, FUN=col_na)
na_count
```

```
> na_count
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
           0            1            2            1            0
```

1. 결측값 (missing value)

- NA 를 포함한 행(row)은 몇 개인지 파악

```
is.na(x)                          # 데이터셋 전체
rowSums(is.na(x))                 # 행별 NA 개수
sum(rowSums(is.na(x))>0)          # NA 가 포함된 행의개수
```

```
> is.na(x)
      Sepal.Length Sepal.Width Petal.Length Petal.Width Species
 [1,]        FALSE        TRUE         TRUE        FALSE   FALSE
 [2,]        FALSE       FALSE         TRUE        FALSE   FALSE
 [3,]        FALSE       FALSE        FALSE         TRUE   FALSE
 [4,]        FALSE       FALSE        FALSE        FALSE   FALSE
 [5,]        FALSE       FALSE        FALSE        FALSE   FALSE
```

```
> rowSums(is.na(x))
  [1] 2 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [48] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [95] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[142] 0 0 0 0 0 0 0 0 0 0
> sum(rowSums(is.na(x))>0)
[1] 3
```

1. 결측값 (missing value)

- NA 를 포함한 행(row)을 제외하고 새로운 데이터 생성

```
head(x)
x[!complete.cases(x),]
y <- x[complete.cases(x),]
head(y)
```

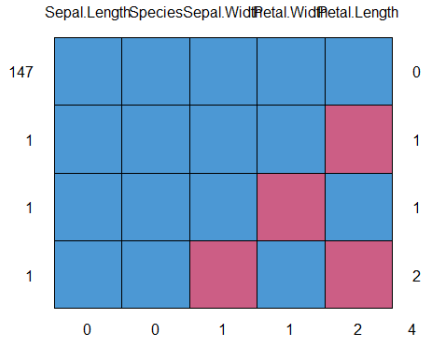
```
> head(x)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1          NA           NA         0.2  setosa
2          4.9         3.0           NA         0.2  setosa
3          4.7         3.2          1.3          NA  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> x[!complete.cases(x),]
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1          NA           NA         0.2  setosa
2          4.9         3.0           NA         0.2  setosa
3          4.7         3.2          1.3          NA  setosa
> y <- x[complete.cases(x),]
> head(y)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
7          4.6         3.4          1.4         0.3  setosa
8          5.0         3.4          1.5         0.2  setosa
9          4.4         2.9          1.4         0.2  setosa
```

1. 결측값 (missing value)

- 결측값을 적당한 값으로 추정하여 치환

```
library(mice)                          # 결측값 추정 지원
md.pattern(x)                          # 결측값 통계
result <- mice(x, m=5, maxit = 50,
   method = 'pmm', seed = 500)         # 결측값 예측
#get complete data ( 2nd out of 5)
impute_x <- complete(result,2)         # 예측값 반영
head(x)
head(impute_x)
head(iris)
```

```
> md.pattern(x)
    Sepal.Length Species Sepal.Width Petal.Width Petal.Length
147            1       1           1           1            1 0
  1            1       1           1           1            0 1
  1            1       1           1           0            1 1
  1            1       1           0           1            0 2
               0       0           1           1            2 4
```



0: 결측값
1: 정상

1. 결측값 (missing value)

```
> head(x)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1          NA           NA         0.2  setosa
2          4.9         3.0           NA         0.2  setosa
3          4.7         3.2          1.3          NA  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> head(impute_x)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.2          1.5         0.2  setosa
2          4.9         3.0          1.5         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
>
```

결측값 추정을 지원하는 다른 패키지들 :

Amelia
missForest
Hmisc
mi

[연습문제 1]

- 다음과 같이 결측값이 포함된 데이터셋 ds 를 생성한다

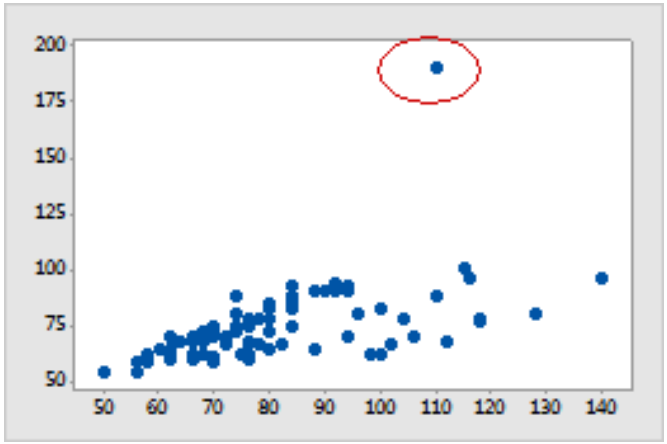
```
library(mice)
ds <- ampute(iris[,1:4], 0.2)$amp)
```

1. 각 컬럼별로 결측값이 몇 개인지 보이시오
2. 결측값이 포함된 행들의 데이터를 보이시오
3. 결측값이 포함된 행은 몇 개인지 보이시오
4. 결측값이 포함된 행들을 제외하고 새로운 데이터셋(ds.new)을 만들어 보이시오

2. 이상치(outlier)

- 정상 범위 밖에 있는 값
- 입력 실수일수도 있고 실제 값 일수도 있다.
- 전체 데이터 분포의 특성에 영향을 미친다.
- 품질관리에서는 불량을 찾을 때 제일 먼저 보는 것
- 은행 거래에서 사기거래를 탐지할 때 사용되기도 한다

- 이상치를 제외하고 분석을 할지, 포함해서 분석을 해야 할지 판단을 해야 한다.



http://support.minitab.com/en-us/minitab-express/1/help-and-how-to/basic-statistics/inference/supporting-topics/data-concepts/identifying-outliers/

https://blogdotrichanchordotcom.wordpress.com/2016/05/23/outlier-detection-an-overview-and-applications/

2. 이상치(outlier)

- (1) 논리적으로 있을 수 없는 값이 있는 지 찾아본다
  예) 성별에서 좋아하는 색깔을 1~5 로 표시하기로 했는데 7 이 있음
  예) 몸무게에 마이너스 값이 있음

- (2) 상식을 벗어난 값이 있는지 찾아본다.
  예) 나이가 100살 이상인 사람

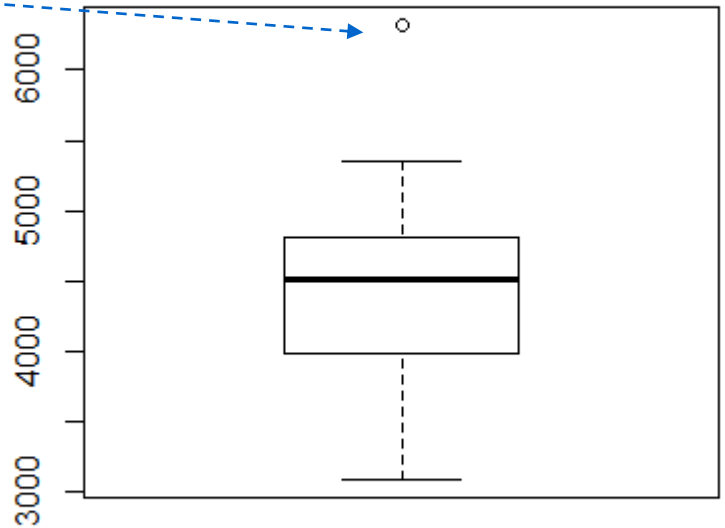
- (3) boxplot 을 통해 찾아본다.

2. 이상치(outlier)

- 이상치 탐색

```
st <- data.frame(state.x77)
boxplot(st$Income)
boxplot.stats(st$Income)$out
```

```
> boxplot.stats(st$Income)$out
[1] 6315
```



2. 이상치(outlier)

- 각 컬럼의 이상치를 NA 처리한 후 NA 를 포함한 행들을 제거

```
out.val <- boxplot.stats(st$Income)$out
st$Income[st$Income %in% out.val] = NA
st$Income
newdata <- st[complete.cases(st),]
```

```
> st$Income
 [1] 3624   NA 4530 3378 5114 4884 5348 4809 4815 4091 4963 4119 5107
[14] 4458 4628 4669 3712 3545 3694 5299 4755 4751 4675 3098 4254 4347
[27] 4508 5149 4281 5237 3601 4903 3875 5087 4561 3983 4660 4449 4558
[40] 3635 4167 3821 4188 4022 3907 4701 4864 3617 4468 4566
```

%in% 을 사용한 이유는 **out.val** 이 여러 값을 포함하는 경우도 있기 때문.

[연습문제 2]

- state.x77 데이터셋을 st 에 저장한 후 st 의 각 변수(컬럼) 들에 대해 이상치가 존재하는지 boxplot 그려 확인하시오

```
st <- data.frame(state.x77)
```

- 이상치가 존재하는 경우 이상치를 NA 로 저장하시오

- st 에서 NA 가 존재하는 행들을 제거하여 st2 에 저장하시오

3. 정렬(sort)

- 벡터의 정렬

```
v1 <- c(1,7,6,8,4,2,3)
order(v1)
v1 <- sort(v1)                          # 오름차순
v1
v2 <- sort(v1, decreasing=T)            # 내림차순
v2
```

```
> v1 <- c(1,7,6,8,4,2,3)
> order(v1)
[1] 1 6 7 5 3 2 4
> v1 <- sort(v1)                        # 오름차순
> v1
[1] 1 2 3 4 6 7 8
> v2 <- sort(v1, decreasing=T)          # 내림차순
> v2
[1] 8 7 6 4 3 2 1
```

3. 정렬(sort)

- 기준 변수(컬럼)값에 의한 2차원 배열의 정렬

```
head(iris)
order(iris$Sepal.Length)
iris[order(iris$Sepal.Length),]
```

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> order(iris$Sepal.Length)
  [1]  14   9  39  43  42   4   7  23  48   3  30  12  13  25  31  46
 [17]   2  10  35  38  58 107   5   8  26  27  36  41  44  50  61  94
 [33]   1  18  20  22  24  40  45  47  99  28  29  33  60  49   6  11
 [49]  17  21  32  85  34  37  54  81  82  90  91  65  67  70  89  95
 [65] 122  16  19  56  80  96  97 100 114  15  68  83  93 102 115 143
 [81]  62  71 150  63  79  84  86 120 139  64  72  74  92 128 135  69
 [97]  98 127 149  57  73  88 101 104 124 134 137 147  52  75 112 116
[113] 129 133 138  55 105 111 117 148  59  76  66  78  87 109 125 141
[129] 145 146  77 113 144  53 121 140 142  51 103 110 126 130 108 131
[145] 106 118 119 123 136 132
> iris[order(iris$Sepal.Length),]
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
14           4.3         3.0          1.1         0.1     setosa
9            4.4         2.9          1.4         0.2     setosa
39           4.4         3.0          1.3         0.2     setosa
43           4.4         3.2          1.3         0.2     setosa
42           4.5         2.3          1.3         0.3     setosa
4            4.6         3.1          1.5         0.2     setosa
7            4.6         3.4          1.4         0.3     setosa
```

3. 정렬(sort)

- 기준 변수(컬럼)값에 의한 2차원 배열의 정렬

```
iris[order(iris$Sepal.Length, decreasing=T),]
iris[order(iris$Species, iris$Sepal.Length),]
```

```
> iris[order(iris$Sepal.Length, decreasing=T),]
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
132          7.9         3.8          6.4         2.0  virginica
118          7.7         3.8          6.7         2.2  virginica
119          7.7         2.6          6.9         2.3  virginica
123          7.7         2.8          6.7         2.0  virginica
136          7.7         3.0          6.1         2.3  virginica
106          7.6         3.0          6.6         2.1  virginica
131          7.4         2.8          6.1         1.9  virginica
108          7.3         2.9          6.3         1.8  virginica
110          7.2         3.6          6.1         2.5  virginica
```

```
> iris[order(iris$Species, iris$Sepal.Length),]
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
14          4.3         3.0          1.1         0.1     setosa
9           4.4         2.9          1.4         0.2     setosa
39          4.4         3.0          1.3         0.2     setosa
43          4.4         3.2          1.3         0.2     setosa
42          4.5         2.3          1.3         0.3     setosa
4           4.6         3.1          1.5         0.2     setosa
7           4.6         3.4          1.4         0.3     setosa
23          4.6         3.6          1.0         0.2     setosa
48          4.6         3.2          1.4         0.2     setosa
```

3. 정렬(sort)

- order() 와 rank()

```
v3 <- c(1,7,2,5)
order(v3)
rank(v3)
```

```
> order(v3)
[1] 1 3 4 2
> rank(v3)
[1] 1 4 2 3
```

1 2 5 7

order(v3):
v3을 정렬한다고 했을때 현재 **v3** 에서 어떤 순서로 뽑아야 하는가

rank(v3):
v3에서 현재의 값의 순위(등수)

[연습문제 3]

- 1. state.x77 데이터셋을 Population 을 기준으로 오름차순 정렬을 하시오
- 2. state.x77 데이터셋을 Income 을 기준으로 내림차순 정렬을 하시오
- 3. Illiteracy(문맹률) 인 낮은 상위 10개주의 이름과 문맹률을 보이시오

```
> head(state.x77)
           Population Income Illiteracy Life Exp Murder HS Grad Frost
Alabama          3615   3624        2.1    69.05   15.1    41.3    20
Alaska            365   6315        1.5    69.31   11.3    66.7   152
Arizona          2212   4530        1.8    70.55    7.8    58.1    15
Arkansas         2110   3378        1.9    70.66   10.1    39.9    65
California      21198   5114        1.1    71.71   10.3    62.6    20
Colorado         2541   4884        0.7    72.06    6.8    63.9   166
             Area
Alabama     50708
Alaska     566432
Arizona    113417
Arkansas    51945
California 156361
Colorado   103766
```

주의. **state.x77** 은 **matrix** 이기 때문에 컬럼을 지칭할때
state.x77$Population 과 같이 쓸수 없고 **state.x77[,1]** 과 같이 써야한다.

4. 분리(split) & ,선택(subset)

- 분리(split) : 데이터셋을 주어진 기준에 따라 여러개로 분리한다

```
sp <- split(iris, iris$Species)
sp              # list
```

```
> sp <- split(iris, iris$Species)
> sp
$setosa
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4         0.2  setosa
2           4.9         3.0          1.4         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
4           4.6         3.1          1.5         0.2  setosa
5           5.0         3.6          1.4         0.2  setosa
6           5.4         3.9          1.7         0.4  setosa
7           4.6         3.4          1.4         0.3  setosa
49          5.3         3.7          1.5         0.2  setosa
50          5.0         3.3          1.4         0.2  setosa

$versicolor
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
51           7.0         3.2          4.7         1.4 versicolor
52           6.4         3.2          4.5         1.5 versicolor
53           6.9         3.1          4.9         1.5 versicolor
54           5.5         2.3          4.0         1.3 versicolor
55           6.5         2.8          4.6         1.5 versicolor
56           5.7         2.8          4.5         1.3 versicolor
57           6.3         3.3          4.7         1.6 versicolor
58           4.9         2.4          3.3         1.0 versicolor

$virginica
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
101          6.3         3.3          6.0         2.5 virginica
102          5.8         2.7          5.1         1.9 virginica
103          7.1         3.0          5.9         2.1 virginica
104          6.3         2.9          5.6         1.8 virginica
```

4. 분리(split) & ,선택(subset)

```
summary(sp)
class(sp)
sp$setosa
```

```
> summary(sp)
           Length Class      Mode
setosa     5      data.frame list
versicolor 5      data.frame list
virginica  5      data.frame list
>
```

```
> class(sp)
[1] "list"
> sp$setosa
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4         0.2  setosa
2           4.9         3.0          1.4         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
4           4.6         3.1          1.5         0.2  setosa
5           5.0         3.6          1.4         0.2  setosa
6           5.4         3.9          1.7         0.4  setosa
7           4.6         3.4          1.4         0.3  setosa
8           5.0         3.4          1.5         0.2  setosa
```

4. 분리(split) & ,선택(subset)

- 선택(subset) : 조건에 맞는 행(row)들을 추출

```
subset(iris, Species == "setosa")
subset(iris, Sepal.Length > 5.1)
subset(iris, Sepal.Length > 5.1 &
             Sepal.Width > 3.9)

subset(iris, Sepal.Length > 5.1,
       select=c(Petal.Length,Petal.Width))
```

```
> subset(iris, Sepal.Length > 5.1,
+        select=c(Petal.Length,Petal.Width))
    Petal.Length Petal.Width
6            1.7         0.4
11           1.5         0.2
15           1.2         0.2
16           1.5         0.4
17           1.3         0.4
19           1.7         0.3
21           1.7         0.2
28           1.5         0.2
29           1.4         0.2
32           1.5         0.4
```

4. 분리(split) & ,선택(subset)

- 조건에 맞는 값들의 인덱스 알아내기

```
x <- c(3,1,7,8,5,9,4)
which(x>5)
which(iris$Species=="setosa")
which.max(iris$Sepal.Length)
which.min(iris$Sepal.Width)
```

```
> x <- c(3,1,7,8,5,9,4)
> which(x>5)
[1] 3 4 6
> which(iris$Species=="setosa")
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
[23] 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
[45] 45 46 47 48 49 50
> which.max(iris$Sepal.Length)
[1] 132
> which.min(iris$Sepal.width)
[1] 61
```

5. 샘플링(sampling)

- 많은 데이터중 일부를 선택해야 할 때

```
x <- 1:100
# x 에서 10개의 수를 임의로 추출 (비복원 추출)
y <- sample(x, size=10, replace = FALSE)
y
```

```
> x <- 1:100
> # x 에서 10개의 수를 임의로 추출 (비복원 추출)
> y <- sample(x, size=10, replace = FALSE)
> y
 [1] 99 38 98 45 74 10 73 42 40 15
>
```

5. 샘플링(sampling)

```
# iris 에서 50개의 행(row)을 임의로 추출 (비복원 추출)
idx <- sample(nrow(iris), size=50,
              replace = FALSE)
iris.50 <- iris[idx,]
head(iris.50)
```

```
> head(iris.50)
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
17           5.4         3.9          1.3         0.4     setosa
140          6.9         3.1          5.4         2.1  virginica
62           5.9         3.0          4.2         1.5 versicolor
52           6.4         3.2          4.5         1.5 versicolor
6            5.4         3.9          1.7         0.4     setosa
27           5.0         3.4          1.6         0.4     setosa
>
```

replace = FALSE 가 default 이기 때문에 생략가능

5. 샘플링(sampling)

- 조합(combination)

```
combn(5,3)                    # 5개중 3개를 뽑는 조합

x = c("red","green","blue","black","white")
com <- combn(x,2)             # x 의 원소를 2개씩 뽑는 조합
com
for(i in 1:ncol(com)) {
  cat(com[,i], "\n")
}
```

```
> combn(5,3)
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    1    1    1    1    1    1    2    2    2     3
[2,]    2    2    2    3    3    4    3    3    4     4
[3,]    3    4    5    4    5    5    4    5    5     5
> x = c("red","green","blue","black","white")
> com <- combn(x,2)
> com
     [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]    [,8]
[1,] "red"   "red"   "red"   "red"   "green" "green" "green" "blue"
[2,] "green" "blue"  "black" "white" "blue"  "black" "white" "black"
     [,9]    [,10]
[1,] "blue"  "black"
[2,] "white" "white"
> for(i in 1:ncol(com)) {
+   cat(com[,i], "\n")
+ }
red green
red blue
red black
red white
```

[연습문제 4]

1. state.x77 데이터셋의 자료를 state.region 에 있는 지역정보에 따라 5개 데이터 그룹으로 분리하시오. 분리된 데이터의 내용을 보이시오

2. state.x77 에서 면적 (area)이 Alabama 보다 크고 California 보다 작은 주의 이름과 인구수(Population), 소득(Income), 면적(Area)을 보이시오

3. iris 에서 40개의 행을 임의 추출하여 iris.40 에 저장하고, 나머지 행들은 iris.110에 저장하시오. iris.40 과 iris.110 의 내용을 보이시오

4. iris 에서 3개의 품종(Species)을 2개씩 짝지어 보이시오

6. 데이터 요약(aggregate)

- 2차원 배열에 있는 데이터를 기준 변수(컬럼)에 따라 집계함

- iris 데이터셋에서 각 품종별로 꽃잎 꽃받침의 폭과 길이의 평균을 보이시오

```
agg <- aggregate(iris[,-5], by=list(iris$Species),
                 FUN=mean)
agg
```

```
> agg <- aggregate(iris[,-5], by=list(iris$Species), FUN=mean)
> agg
     Group.1 Sepal.Length Sepal.Width Petal.Length Petal.Width
1     setosa        5.006       3.428        1.462       0.246
2 versicolor        5.936       2.770        4.260       1.326
3  virginica        6.588       2.974        5.552       2.026
```

집계기준

6. 데이터 요약(aggregate)

- iris 데이터셋에서 각 품종별로 꽃잎 꽃받침의 폭과 길이의 표준편차를 보이시오

```
agg <- aggregate(iris[,-5], by=list(iris$Species),
                 FUN=sd)
agg
```

```
> agg
     Group.1 Sepal.Length Sepal.Width Petal.Length Petal.Width
1     setosa    0.3524897   0.3790644    0.1736640   0.1053856
2 versicolor    0.5161711   0.3137983    0.4699110   0.1977527
3  virginica    0.6358796   0.3224966    0.5518947   0.2746501
>
```

6. 데이터 요약(aggregate)

- mtcars 데이터셋에서 cyl,vs 을 기준으로 다른 컬럼들의 최대값을 보이시오

```
head(mtcars)
agg <- aggregate(mtcars, by=list(mtcars$cyl,
                    mtcars$vs),FUN=max)
agg
```

```
> agg
  Group.1 Group.2  mpg cyl  disp  hp drat    wt  qsec vs am gear carb
1       4       0 26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
2       6       0 21.0   6 160.0 175 3.90 2.875 17.02  0  1    5    6
3       8       0 19.2   8 472.0 335 4.22 5.424 18.00  0  1    5    8
4       4       1 33.9   4 146.7 113 4.93 3.190 22.90  1  1    5    2
5       6       1 21.4   6 258.0 123 3.92 3.460 20.22  1  0    4    4
>
```

mtcars$cyl mtcars$vs

6. 데이터 요약(aggregate)

```
head(mtcars)
agg <- aggregate(mtcars, by=list(cyl=mtcars$cyl,
                    vs=mtcars$vs),FUN=max)
agg
```

```
> agg
  cyl vs  mpg cyl  disp  hp drat    wt  qsec vs am gear carb
1   4  0 26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
2   6  0 21.0   6 160.0 175 3.90 2.875 17.02  0  1    5    6
3   8  0 19.2   8 472.0 335 4.22 5.424 18.00  0  1    5    8
4   4  1 33.9   4 146.7 113 4.93 3.190 22.90  1  1    5    2
5   6  1 21.4   6 258.0 123 3.92 3.460 20.22  1  0    4    4
```

6. 데이터 요약(aggregate)

- attach, detach

```
agg <- aggregate(mtcars, by=list(mtcars$cyl,
                    mtcars$vs),FUN=max)
agg
```

```
attach(mtcars)
agg <- aggregate(mtcars, by=list(cyl,
                    vs),FUN=max)
agg
detach(mtcars)
```

데이터셋의 이름이 길 때 사용하면 편리

실습x

7. 데이터 병합 (merge)

- 공통 컬럼을 매개로 하여 2개의 2차원 배열을 하나로 병합함

```
x <- data.frame(name=c("a","b","c"),
math=c(90,80,40))
y <- data.frame(name=c("a","b","d"),
korean=c(75,60,90))
x
y
merge(x,y, by=c("name"))
```

```
> x
  name math
1    a   90
2    b   80
3    c   40
> y
  name korean
1    a     75
2    b     60
3    d     90
> merge(x,y, by=c("name"))
  name math korean
1    a   90     75
2    b   80     60
```

공통컬럼의 이름이 같은 경우는 **by=c(..)** 생략 가능

7. 데이터 병합 (merge)

```
x
y
merge(x,y, all.x=T)
merge(x,y, all.y=T)
merge(x,y, all=T)
```

```
> x
  name math
1    a   90
2    b   80
3    c   40
> y
  name korean
1    a     75
2    b     60
3    d     90
```

```
> merge(x,y, all.x=T)
  name math korean
1    a   90     75
2    b   80     60
3    c   40     NA
> merge(x,y, all.y=T)
  name math korean
1    a   90     75
2    b   80     60
3    d   NA     90
> merge(x,y, all=T)
  name math korean
1    a   90     75
2    b   80     60
3    c   40     NA
4    d   NA     90
```

7. 데이터 병합 (merge)

- 공통 컬럼이름이 다른 경우

```
x <- data.frame(name=c("a","b","c"),
math=c(90,80,40))
y <- data.frame(sname=c("a","b","d"),
korean=c(75,60,90))
x
y
merge(x,y, by.x=c("name"), by.y=c("sname"))
```

```
> x
  name math
1    a   90
2    b   80
3    c   40
> y
  sname korean
1     a     75
2     b     60
3     d     90
> merge(x,y, by.x=c("name"), by.y=c("sname"))
  name math korean
1    a   90     75
2    b   80     60
```

[연습문제 5]

- 1. 제공된 파일중 subway.csv 와 subway_latlong 파일을 읽은후 병합하여 subway.tot 에 저장하시오.
  - 병합기준 컬럼은 STATION_CD 와 station
- 2. subway.tot 에서 역이름과 날짜를 기준으로 하루 평균 탑승인원 (onr_tot) 과 하차인원(off_tot) 을 집계하여 보이시오
- 3. subway.tot 에서 2011 년도에 대해서만 역이름 기준 총 탑승인원 (onr_tot) 과 하차인원(off_tot) 을 집계하여 보이시오
- 4. subway.tot 에서 2011 년도에 대해서 호선(LINE_NUM)별 총 탑승인원 (onr_tot) 과 하차인원(off_tot) 을 집계하여 보이시오

[tip]

- R 의 함수에서 매개변수를 표현하는데는 여러가지 방식이 있음

일반

```
agg <- aggregate(iris[,-5], by=list(iris$Species),
                 FUN=mean)
agg
```

formular 방식

```
agg <- aggregate(.~Species, data=iris,
                 FUN=mean)
```

```
> agg <- aggregate(.~Species, data=iris,
+                  FUN=mean)
>
> agg
     Species Sepal.Length Sepal.Width Petal.Length Petal.Width
1     setosa        5.006       3.428        1.462       0.246
2 versicolor        5.936       2.770        4.260       1.326
3  virginica        6.588       2.974        5.552       2.026
```

[tip]

Usage

```
aggregate(x, ...)

## Default S3 method:
aggregate(x, ...)

## S3 method for class 'data.frame'
aggregate(x, by, FUN, ..., simplify = TRUE, drop = TRUE)

## S3 method for class 'formula'
aggregate(formula, data, FUN, ...,
          subset, na.action = na.omit)

## S3 method for class 'ts'
aggregate(x, nfrequency = 1, FUN = sum, ndeltat = 1,
          ts.eps = getOption("ts.eps"), ...)
```

Arguments

x	an R object.
by	a list of grouping elements, each as long as the variables in the data frame x. The elements are coerced to factors before use.
FUN	a function to compute the summary statistics which can be applied to all data subsets

실습x