상자그림(box plot)

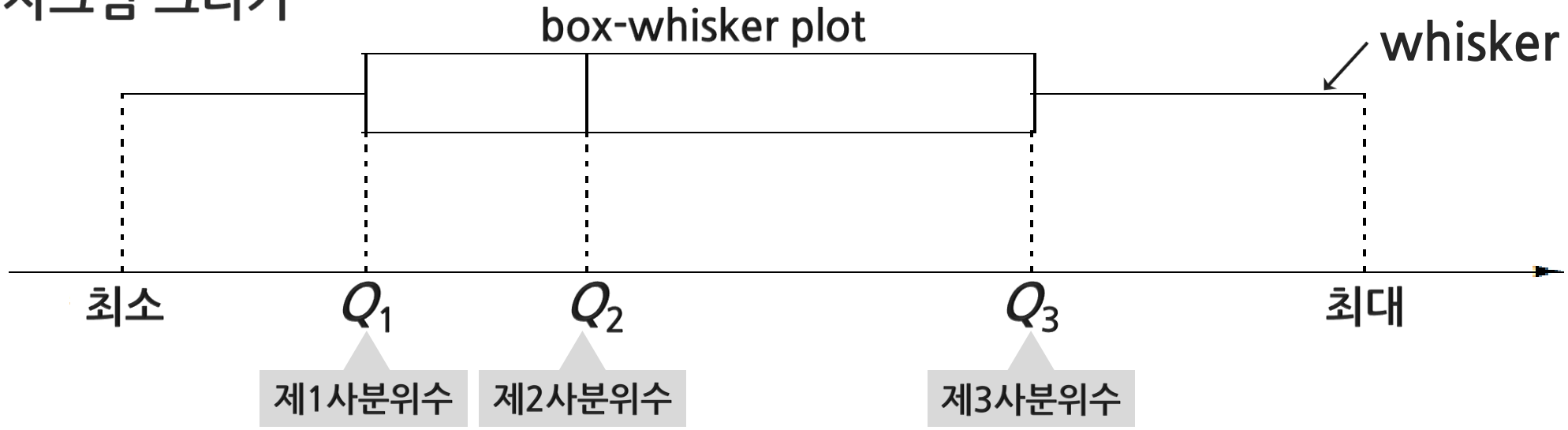
- 상자그림 작성법
- boxplot 함수

상자그림 작성법

상자그림

사분위에 해당하는 부분은 상자로 표현하고 그 밖의 범위를 선으로 연결하는 그림

상자그림 그리기



* 최솟값과 최댓값이 다른 값에 비해 너무 작거나 큰 경우 이 선이 너무 길어질 수 있으므로 대개 이 선의 길이가 상자의 넓이(= Q_3 - Q_1= 사분위수범위: IQR)의 1.5배까지만 선을 그리고 그보다 바깥에 위치하는 자료는 점으로 표시함

boxplot 함수

R-언어에서 상자그림을 그려주는 내장함수

```
boxplot(formula, data = NULL, ..., subset, na.action = NULL)
또는
boxplot(x, ..., range = 1.5, width = NULL, varwidth = FALSE, outline = TRUE,
        border = par("fg"), col = NULL, horizontal = FALSE, ...)
```

매개변수

- x : 상자그림을 그릴 자료를 포함한 벡터, 행렬, 데이터 프레임으로 자료가 두 개의 이상의 열을 가질 경우 각 열의 상자그림을 그림을 side-by-side로 그리는 것이 기본
- formula : y ~ x 형태로 설정하며 x의 각 값에 따른 y의 상자그림을 그림
- data : formula에서 사용한 변수를 포함하는 데이터 프레임을 설정
- subset : 설정한 데이터 프레임에서 일부만 가져 올 때 해당 조건을 설정
- range : 수염(whisker)의 길이를 최대 몇 배의 IQR까지 할 것인지 설정
- horizontal : 상자를 가로로 그릴지 설정(F가 기본값)
- border : 상자의 색깔
- col : 상자 내부의 색깔
- outline : 수염의 길이 보다 더 멀리 있는 자료를 표시할지 설정

boxplot 함수

예시

사용할 자료

```
> BMI <- read.table(url("http://jupiter.hallym.ac.kr/ftpdata/data/bmi.txt"),
      col.names=c("height", "weight", "year", "religion", "gender", "marriage"))
```

자료출처

- url 함수를 사용하여 인터넷 사이트에서 바로 읽어 올 수 있으며, 위의 인터넷주소를 브라우저 창에 붙여 넣으면 브라우저에서 볼 수 있음

내용 : 2000년, 177명에 대한 조사 결과

- 키, 몸무게, 출생년도
- 종교(Bu=불교, C1=개신교, C2=가톨릭, No=없음)
- 성별(F=여자, M=남자)
- 결혼여부(N=미혼, Y=기혼)

boxplot 함수

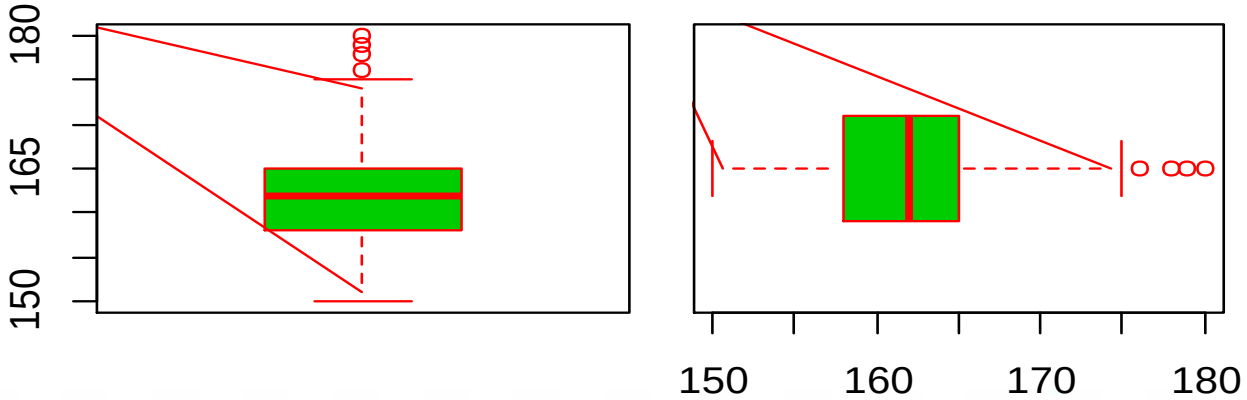
예시 1

자료

이 자료에서 키에 대한 상자그림을 세로(기본값) 및 가로로 그려보자. 이 때 상자내부의 색깔은 3번(Green)으로 설정하였다.

```
par(mfrow=c(1,2))
boxplot(BMI$height, border=2, col=3)
boxplot(BMI$height, border=2, col=3, horizontal=T)
```

함수 사용 결과



* 수염 위쪽(오른쪽)의 점은 지나치게 큰 값을 표현한 것으로 제3사분위수+1.5IQR 보다 큰 값들이 점으로 표시됨

boxplot 함수

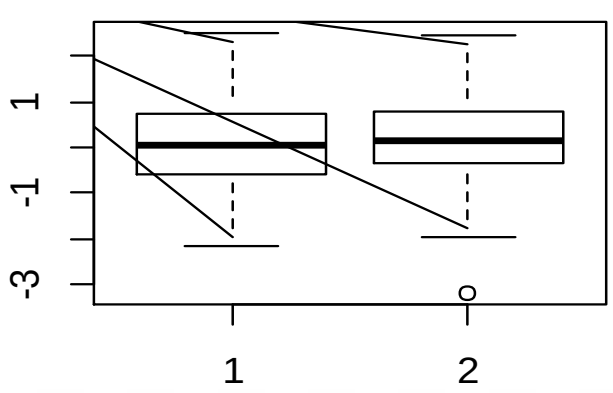
예시 2

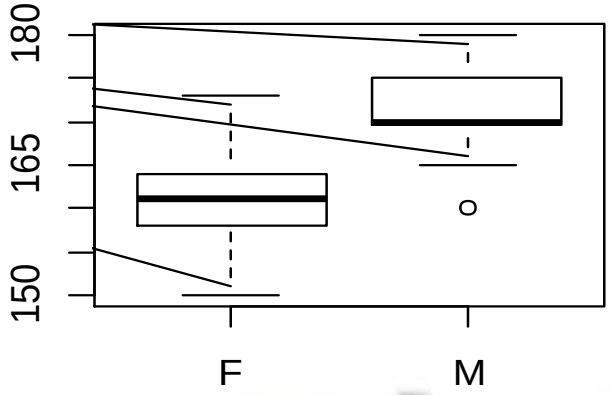
자료

(비교 상자그림) 상자 그림으로 자료의 분포나 5개의 분위수 값 등을 알 수 있어서 하나만으로도 충분히 의미가 있지만 그룹별 상자 그림을 좌우로 그려 그룹별 분포를 비교하는데 많이 사용된다. 다음은 100개의 표준정규분포에서 난수 2세트를 비교한 것과 BMI 자료에서 성별에 따른 상자그림을 그려본 것이다.

```
par(mfrow=c(1,2))
boxplot(rnorm(100), rnorm(100))
boxplot(height~ gender, data=BMI)
```

함수 사용 결과





boxplot 함수

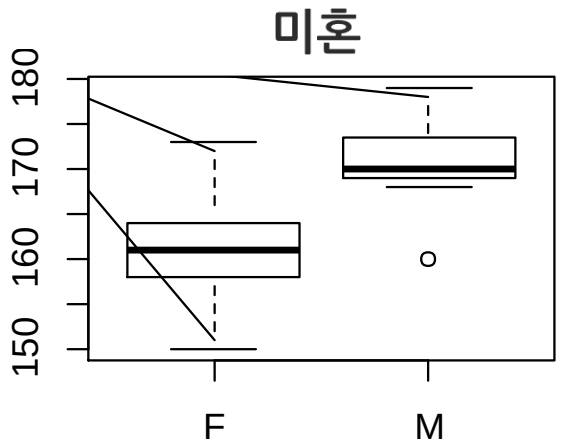
예시 3

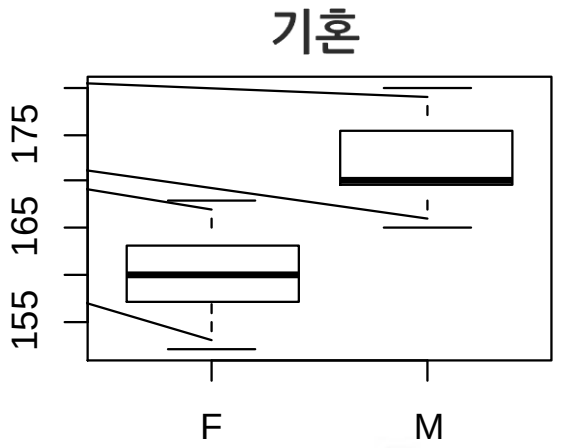
자료

(자료의 일부분만 그리기, 제목 주기) 상자 그림을 포함하여 많은 R-그래픽에서 옵션으로 자료의 일부분만 선택할 수 있는 subset을 설정할 수 있다. 다음은 BMI 자료에서 미혼자의 성별 키에 대한 상자그림과 기혼자의 성별 상자그림을 그려본 프로그램이다.

```
boxplot(height~ gender, data=BMI, subset= marriage=="N", main="미혼")
boxplot(height~ gender, data=BMI, subset= marriage=="Y", main="기혼")
```

함수 사용 결과





boxplot 함수

예시 4

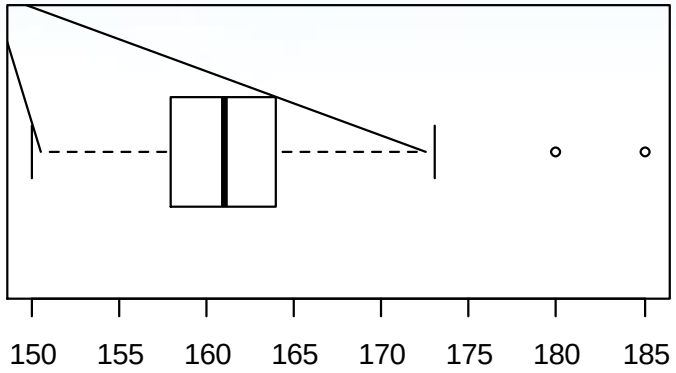
자료

(outliers 에 대한 설정) boxplot 함수는 Q_3+ 1.5IQR 보다 큰 값이나 Q_1 - 1.5IQR 보다 작은 값은 이상치(지나치게 크거나 작은 값)로 판단하고 수염(whisker)를 최대 또는 최솟값까지 연장하지 않고 점으로 표시함. 기본적으로 사용한 1.5배 대신 다른 값을 사용하거나(range 옵션), 아예 이상치를 점으로 표시하지 않고 최대 또는 최솟값까지 수염을 연장하는 경우에 대해서 알아보자.

```
x <- c(BMI$height[BMI$gender == 'F'], 180, 185)
(참고:
IQR=6, 1.5IQR=9, = 164+9=173
> quantile(x)
0% 25% 50% 75% 100%
150 158 161 164 185
)
par(mfrow=c(2,2))
boxplot(x, horizontal=T)
boxplot(x, range=3, horizontal=T)
boxplot(x, horizontal=T, outline=F)
```

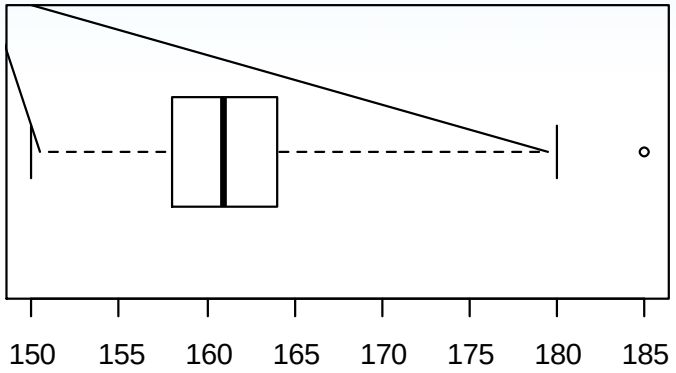
boxplot 함수

예시 4



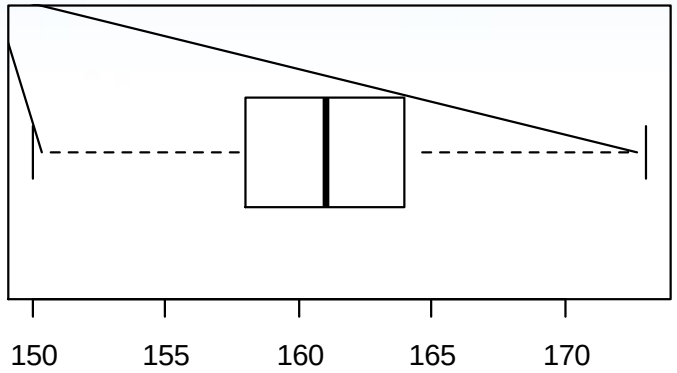
* 기본값을 사용함

➔ 180, 185는 이상치이므로 이는 점으로 표현됨



* range가 3

➔ 보다 큰 185만 이상치로 표시됨



* outline=F로 설정

➔ 수염은 최소 및 최댓값까지 연장됨

줄기잎 그림 (stem-and-leaf plot)

- 줄기잎 그림 작성법
- stem 함수
- aplpack 패키지 사용
- stem.leaf.backback 함수를 사용한 비교 줄기잎 그림

줄기잎 그림 작성법

줄기잎 그림

자료를 줄기와 줄기에 달린 잎으로 표현하는 그림

줄기잎 그림 그리기

9 | 15 ◀ 91점, 95점 각 1명

8 | 3336689 ◀ 83점 3명, 86점 2명, 88점 1명, 89점 1명

7 | 2346677 ◀ 72점 1명, 73점 1명, 74점 1명, 76점 2명, 77점 2명

6 | 799 ◀ 67점 1명, 69점 2명

5 |

4 |

3 | 1 ◀ 31점 1명

줄기잎 그림 작성법

줄기잎 그림의 장단점

장점

- 자료의 분포를 바로 알 수 있음
- 모든 자료가 보여지므로 사실상 자료 전체를 보여줌

단점

- 자료의 수가 많을 때 구현하기 곤란함

stem 함수

R-그래픽에서 줄기잎 그림은 stem 함수로 작성함

사용법

```
stem(x, scale = 1, width = 80,...)
```

매개변수

- x : 줄기잎 그림을 그릴 자료
- scale : 줄기잎 그림의 한 줄기의 크기를 조절함
- width : 한 줄기에 최대한 그릴 잎의 개수를 설정

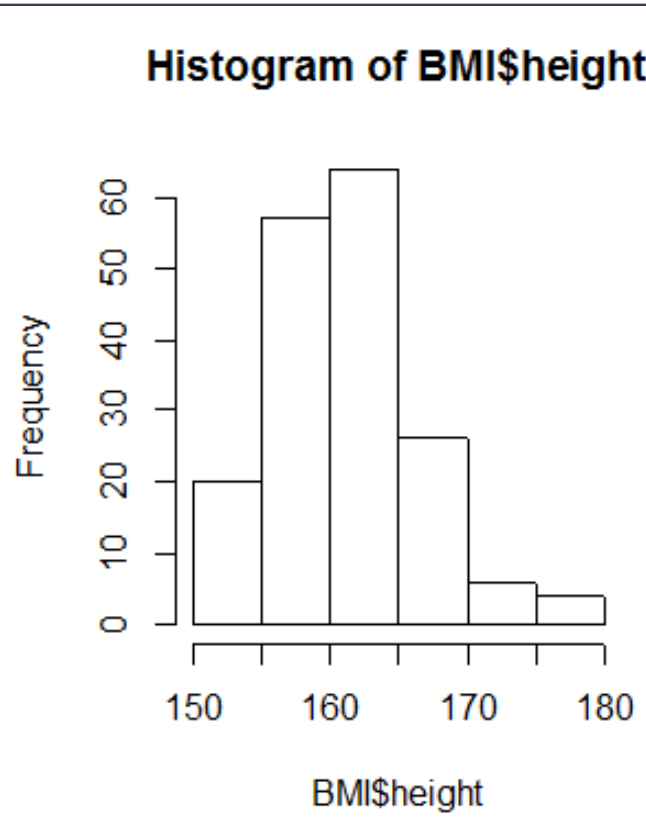
stem 함수

예시 1

자료

BMI 자료의 키에 대한 줄기잎 그림과 히스토그램의 비교

```
> stem(BMI$height)

  The decimal point is at the |

 150 | 00
 152 | 00000000
 154 | 0000000000
 156 | 000000000
 158 | 00000000000000000000
 160 | 00000000000000000000000000000000000000
 162 | 00000000000000000000000000000000
 164 | 00000000000000000000000000000000
 166 | 0000000000
 168 | 0000000000
 170 | 000000
 172 | 000
 174 | 000
 176 | 0
 178 | 00
 180 | 0
```



stem 함수

예시 2

자료

(각 줄기의 크기 설정) scale의 값을 설정하여 각 줄기의 크기를 바꾼다.
아래는 scale을 0.5로 바꾸어(스케일을 축소하여 한 줄기에 더 많은 잎이 달림) BMI 자료에서 키에 대한 줄기잎 그림을 그린 예이다.

```
> stem(BMI$height, scale=.5)

  The decimal point is 1 digit(s) to the right of the |

  15 | 00222333334
  15 | 55555555566666777788888888888888899
  16 | 00000000000000000000000000011111112222222222222222333333333333444
  16 | 55555555555555555666677777788888888899
  17 | 0000002334
  17 | 55689
  18 | 0
```

stem 함수

예시 3

자료

(R의 줄기잎 그림은 이상치를 적절하게 배제하지 못함) 상자그림과 달리 R의 stem 함수는 이상치를 줄기잎 그림에서 배제하지 못하므로 때로 당혹스런 결과가 나온다. 다음은 BMI 자료에서 키를 입력하다가 오타로 1,000이 입력된 경우를 가정하여 줄기잎 그림을 그린 보기이다.

```
> stem(c(BMI$height, 1000))

  The decimal point is 2 digit(s) to the right of the |

   1 | 555555555556666666666666666666666666666666666666666666666666666666+97
   2 |
   2 |
   3 |
   3 |
   . .
   9 |
   9 |
  10 | 0
```

stem 함수

예시 4

자료

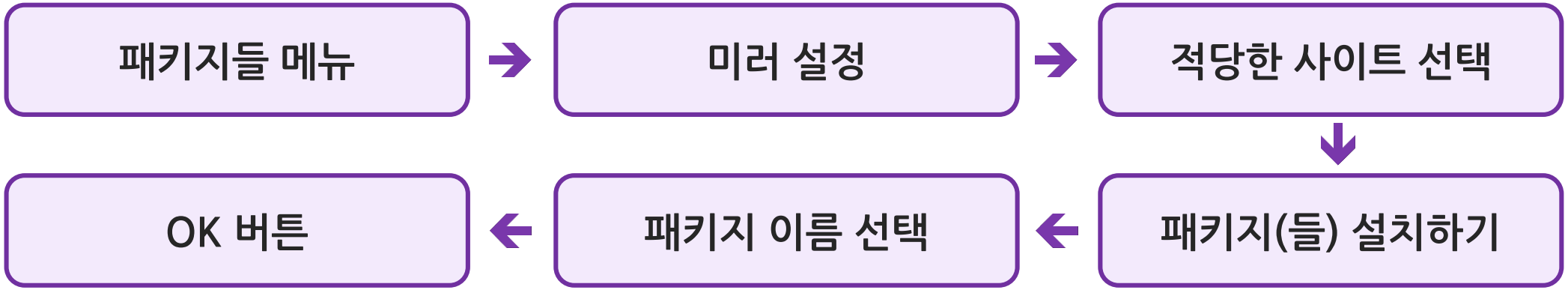
(한 줄기에 포현하지 못한 잎들) 위의 그림에서 줄기 1의 끝에 보면 +97이 있는데 이는 한 줄기에 너무 많은 잎이 있어서 한 줄에 다 그리지 못한 자료가 97개 더 있다는 뜻이다. 이를 줄을 바꾸어서라도 한 줄기에 모든 잎을 표현하게 강제하여 (width=200; 최대 200개까지의 잎들) 다시 그려보면 아래와 같다.

```
> stem(c(BMI$height, 1000), width=200)

  The decimal point is 2 digit(s) to the right of the |

   1 |
555555555566666666666666666666666666666666666666666666666666666666666666
666666666666666666666666666666666666666666666666666666666666666666666666
   2 |
   2 |
   3 |
   3 |
   : :
   9 |
   9 |
  10 | 0
```

aplpack 패키지 사용

줄기잎 기본 줄기잎 그림, 비교 줄기잎 그림이 가능함

패키지의 설치



설치된 패키지를 사용하려면 library 함수로 해당 패키지를 불러옴

패키지를 사용하면 stem.leaf 함수와 stem.leaf.backback 함수가 제공되며 stem.leaf는 일반 줄기잎 그림, stem.leaf.backback 은 비교 줄기잎 그림을 그림

aplpack 패키지 사용

줄기잎 기본 줄기잎 그림, 비교 줄기잎 그림이 가능함

stem.leaf 함수와 stem.leaf.backback 함수 사용법

```
stem.leaf(data, trim.outliers=TRUE ...)
stem.leaf.backback(x,y, trim.outliers=TRUE ...)
```

매개변수

- data : 줄기잎 그림을 그릴 자료가 포함된 벡터
- x, y : 비교 줄기잎 그림을 그릴 두 개의 벡터
- trim.outliers : 1.5배 IQR을 넘는 값은 LO, HI에 따로 배치함(기본값은 TRUE)

aplpack 패키지 사용

예제

자료

(stem.leaf가 기본값으로 처리는 하는 이상치) 1.5배 IQR을 넘는 자료는 어떻게 표시되는지 앞에서와 마찬가지로 BMI 자료의 키에 오타로 인한 1,000이 있다고 가정하고 stem.leaf 함수로 줄기잎 그림을 그려보자.

```
library(aplpack)
stem.leaf(c(BMI$height, 1000), trim.outliers=TRUE)
```

aplpack 패키지 사용

예제

함수 사용 결과

```
> stem.leaf(c(BMI$height, 1000), trim.outliers=TRUE)
1 | 2 : represents 12
 leaf unit : 1
               n : 178
    2      15* | 00
   10         t | 22233333
   20         f | 4555555555
   29         s | 666667777
   48       15. | 888888888888888899
   84      16* | 000000000000000000000000000001111111
  (29)       t | 22222222222222223333333333333
   65         f | 44444444445555555555555555555
   37         s | 6666777777
   27      16. | 8888888899
   17      17* | 000000
   11         t | 233
    8         f | 455
    5         s | 6
    4      17. | 89
    2      18* | 0
HI : 1000
```

* HI : 1000으로 이상치의 값을 따로 출력하였음
* 첫 번째 열의 숫자는 각 줄기에 있는 잎의 위로부터 또는 아래로부터의 누적개수임 (괄호 안에 있는 숫자 기준)
* 숫자 중 괄호가 있는 것은 중앙값이 포함된 줄기임

stem.leaf.backback 함수를 사용한 비교 줄기잎 그림

예제

자료

두 개의 벡터에 대한 줄기잎 그림을 한 번에 좌우로 그린(back to back)다.
다음은 BMI 자료에서 남자의 키와 여자의 키에 대한 줄기잎 그림을 back to back으로
비교 줄기잎 그림을 그린 보기이다.

```
library(aplpack)
stem.leaf.backback(BMI$height[BMI$gender=='F'], BMI$height[BMI$gender== 'M'])
```

stem.leaf.backback 함수를 사용한 비교 줄기잎 그림

예제

함수 사용 결과

```
> stem.leaf.backback(BMI$height[BMI$gender=='F'], BMI$height[BMI$gender== 'M'])
1 | 2 : represents 12, leaf unit : 1
                  BMI$height[BMI$gender == "F"]        BMI$height[BMI$gender == "M"]
 2                                        00 | 15* |
10                                  33333222 |  t |
20                                5555555554 |  f |
29                                 777766666 |  s |
48                      998888888888888888 | 15. |
(35)  11111110000000000000000000000000000 | 16* | 0                                1
75                  33333333333322222222222222222 |  t |
46                5555555555555555444444444444 |  f | 55                               3
20                                 7777776666 |  s |
10                                   98888888 | 16. | 89                               5
                                              | 17* | 000000                           (6)
 2                                        33 |  t | 2                                8
                                              |  f | 455                              7
                                              |  s | 6                                4
                                              | 17. | 89                               3
                                              | 18* | 0                                1
n:                                       158          19
```