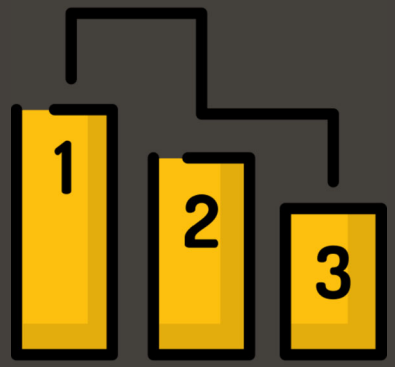
R 프로그래밍 기초다지기

6강 - 범주형 변수와 시각화

슬기로운통계생활

Issac Lee

Factor를 배워보자.



팩터 (Factor) 에 대하여

범주형 변수를 다루는 도구

- 범주형 변수 (Categorical variable) 는 변수가 가질 수 있는 값이 한정된 변수 했었다.
- 벡터에 정보가 더 들어가 있는 개체로 볼 수 있음

```
x <- c(1, 13, 5, 2, 1)
x_factor <- factor(x)
x_factor
```

```
## [1] 1  13 5  2  1
## Levels: 1 2 5 13
```

팩터를 구성하는 요소

벡터 (vector)와 레벨 (Level)

- factor를 factor로 만드는 핵심 요소
- 벡터에 레벨 정보를 입힌 것이 `factor`라고 생각할 수 있음
- 미리 정의된 레벨 정보를 벡터로 참조하는 구조

```
class(x_factor)
```

```
## [1] "factor"
```

```
unclass(x_factor)
```

```
## [1] 1 4 3 2 1
## attr(,"levels")
## [1] "1"  "2"  "5"  "13"
```

팩터 (factor) 선언 방법

`factor()` 함수를 사용

- 들어가는 내용과 레벨 정보를 입력
- 레벨 값과 정보 값은 다를 수 있음
  - 앞으로 들어올 레벨 값을 처음에 지정
  - 이미 정의된 레벨에 해당하는 값만 넣을 수 있음.

```
x
```

```
## [1]  1 13  5  2  1
```

```
x_factor2 <- factor(x,
    levels = c(1, 2, 5, 7, 13))
x_factor2
```

```
## [1] 1  13 5  2  1
## Levels: 1 2 5 7 13
```

레벨 조정하기

기존 레벨을 수정

```
x_factor
```

```
## [1] 1  13 5  2  1
## Levels: 1 2 5 13
```

```
levels(x_factor)
```

```
## [1] "1"  "2"  "5"  "13"
```

```
levels(x_factor) <-
  paste("school", 1:4)
x_factor
```

```
## [1] school 1 school 4 school
## Levels: school 1 school 2 sch
```

순서가 존재할 때 (Ordered factor)

순위 변수 (Ordinal variables)

- `나쁘다, 중간, 좋다.` 의 경우 순서가 존재함.
- 이런 경우 레벨에 코딩을 같이 넣어줄 수 있음.

```
con_vector <- c("bad", "good", "soso", "good")
x_factor3 <- factor(con_vector,
                    levels = c("bad", "soso", "good"),
                    ordered = TRUE)
x_factor3
```

```
## [1] bad  good soso good
## Levels: bad < soso < good
```

`tapply()` 함수와 팩터

팩터 레벨에 따른 함수 적용

- 문법: `tapply(대상 벡터, 나누는 기준, 적용함수)`

```
age <- sample(20:60, 6)
gender <- sample(c("남자", "여자
              6, replace = TRUE
age; gender
```

```
## [1] 49 45 26 27 48 31
```

```
## [1] "남자" "여자" "여자" "여자
```

```
tapply(age, gender, mean)
```

```
## 남자 여자
## 49.0 35.4
```

`tapply()` 함수 응용

펭귄 종류별 부리 길이 계산

- 2개 팩터도 가능
- 문제: `aggregate()` 함수와의 차이점은?

```
library(palmerpenguins)
with(penguins,
     tapply(bill_length_mm,
            species, mean,
            na.rm = TRUE))
```

```
##    Adelie Chinstrap    Gentoo
##  38.79139  48.83382  47.50488
```

```
with(penguins,
     tapply(bill_length_mm,
         list(species, island),
         mean, na.rm = TRUE))
```

```
##              Biscoe    Dream T
## Adelie    38.97500 38.50179
## Chinstrap       NA 48.83382
## Gentoo    47.50488       NA
```

`split()` 함수를 사용한 데이터 쪼개기

데이터 나누기

- 문법: `split(대상, 나누는 기준)`

```
x_factor
```

```
## [1] school 1 school 4 school
## Levels: school 1 school 2 sch
```

```
split(1:5, x_factor)
```

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

by() 함수

tapply() vs. by()

- tapply()의 첫 입력값은 항상 벡터
- by()는 행렬이나 데이터 프레임이 와도 됨

```
with(penguins,
     tapply(bill_length_mm,
            species, mean,
            na.rm = TRUE))
```

```
by(penguins,
  penguins$species,
  function(df){ with(df,
      var(bill_length_mm,
          bill_depth_mm,
          na.rm = TRUE)) }
)
```

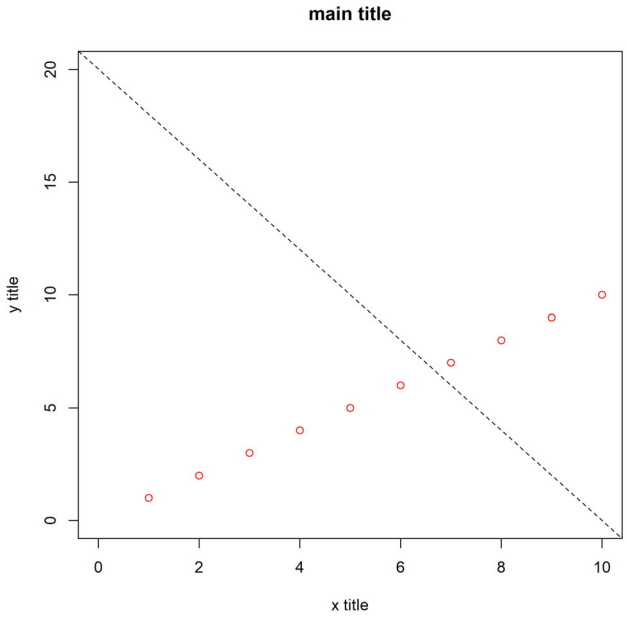
```
## penguins$species: Adelie
## [1] 1.268602
## ------------------------------
## penguins$species: Chinstrap
## [1] 2.477801
## ------------------------------
## penguins$species: Gentoo
```

시각화 맛보기

R plot 개념 이해

- 레이어 개념으로 이루어짐

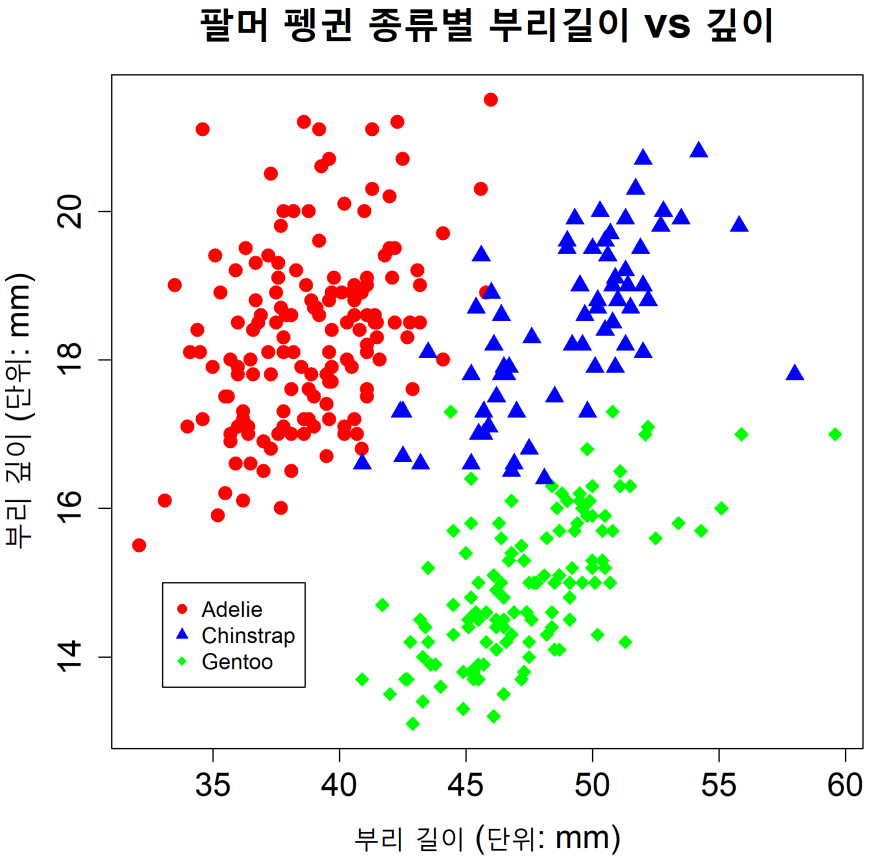
```
plot(0, 0,
     xlim = c(0, 10),
     ylim = c(0, 20),
     type = "n",
     xlab = "x title",
     ylab = "y title",
     main = "main title")
points(1:10, 1:10, col = "red")
abline(a = 20, b = -2,
       lty = "dashed")
```



펭귄데이터 시각화

```
with(penguins,
plot(bill_length_mm,
     bill_depth_mm,
     col = c("red", "blue", "green")[as.factor(species)],
     pch = c(16:18)[as.factor(species)],
     main = "팔머 펭귄 종류별 부리길이 vs 깊이",
     xlab = "부리 길이 (단위: mm)",
     ylab = "부리 깊이 (단위: mm)",
     cex = 1.5, # 점 크기
     cex.main = 2, # 제목
     cex.lab = 1.5, # 축 제목
     cex.axis = 1.5) # 축 숫자
)
legend(33, 15, legend = c("Adelie", "Chinstrap", "Gentoo"),
       col = c("red", "blue", "green"),
       pch = 16:18)
```

시각화 결과


팔머 펭귄 종류별 부리길이 vs 깊이
부리 깊이 (단위: mm)
14
16
18
20
35
40
45
50
55
60
부리 길이 (단위: mm)
Adelie
Chinstrap
Gentoo

종별 부리 길이 vs. 깊이

```
mean_points <-
  aggregate(cbind(bill_length_mm, bill_depth_mm) ~ species,
            data = penguins,
            mean)
names(mean_points) <- c("species", "x", "y")
mean_points
```

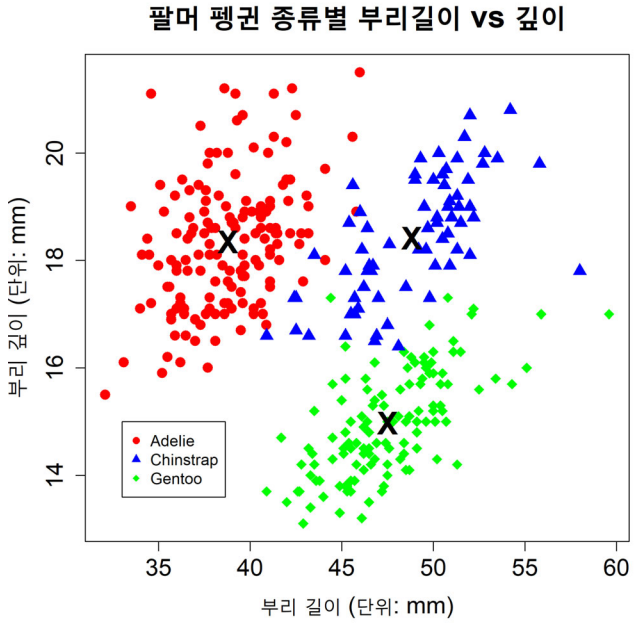
```
##     species        x        y
## 1    Adelie 38.79139 18.34636
## 2 Chinstrap 48.83382 18.42059
## 3    Gentoo 47.50488 14.98211
```

주석 (Annotation)

text() 함수

- 문법: `text(x, y, label)`
- 각 종별 평균 부리 길이 및 깊이 표시

```
text(mean_points[, c("x", "y")],
     label = "X",
     font = 2, # 굵게
     cex = 2)  # 크기
```



다음시간

사용자 함수(Function)와 루프 (Loops)

참고자료 및 사용교재

[1] The art of R programming

- R 공부하시는 분이면 꼭 한번 보셔야 하는 책입니다.
- 위 교재의 한글 번역본 빅데이터 분석 도구 R 프로그래밍도 있습니다. 도서 제목 클릭하셔서 구매 하시면 저의 사리사욕을 충당하는데 도움이 됩니다.