# 화면구현

## R 내장함수와 ggplot2 패키지를 활용한 데이터 시각화 및 비교 분석

2023.04.03

**B2 팀**

**서영석, 박용태, 이현호, 전국림**

# <목차>

# 1. 서론

## 1) 패키지별 데이터 시각화 비교 배경

데이터 시각화란 데이터 전처리 및 분석 후 사용자 입장에서 쉽게 이해 할 수 있도록 도표, 그래프, 그래픽을 이용하여 시각적으로 표현하고 전달하는 과정이다.

이렇듯 분석을 잘해봤자 사용자 입장에서 쉽게 이해가 되지 않거나 직관적이지 않다면 훌륭한 분석이라고 보기 어려울 것이다.

이 보고서에서는 R 의 내장함수와 시각화 패키지의 대표 중 하나인 ggplot2 를 사용하여 각각 어떠한 장/단점이 있고 사용법과 시각화 결과를 비교해 보고자 한다.

## 2. 본론

### 1-1) 막대그래프 (가로)

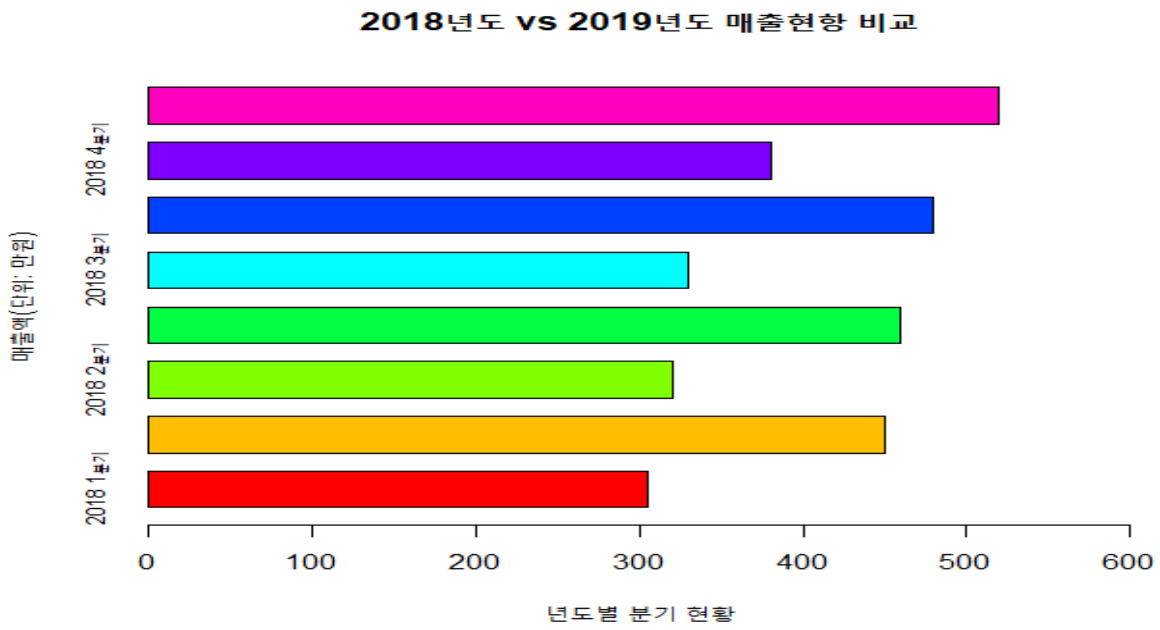


그림 1. barplot 막대그래프 (가로)

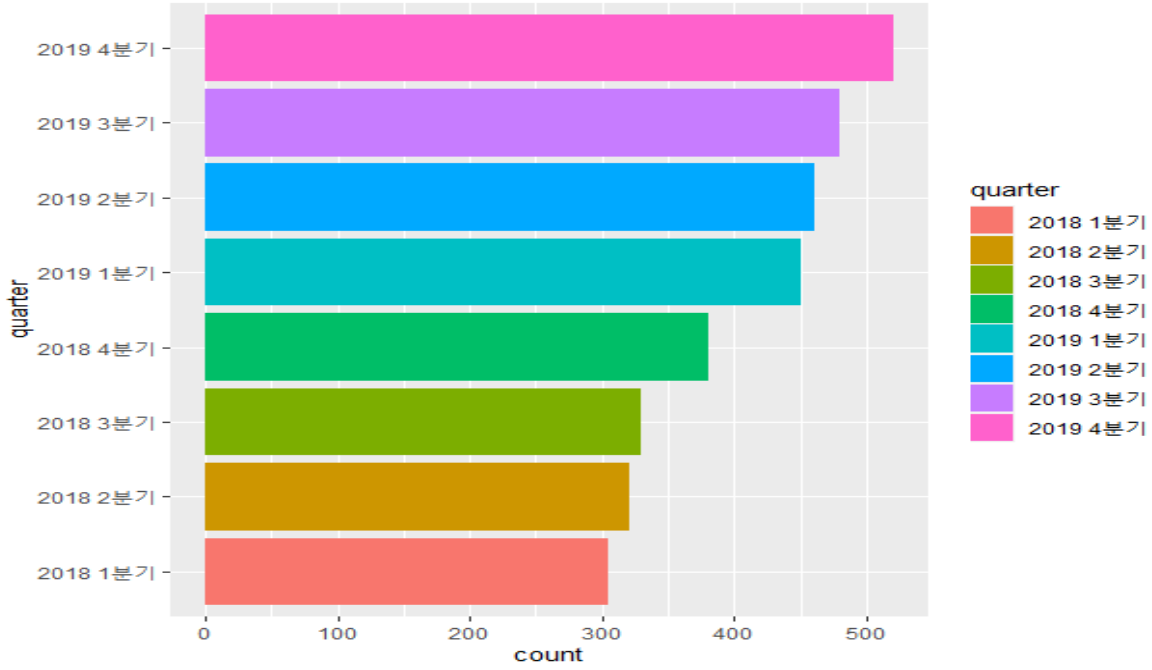


그림 2. ggplot2 막대그래프 (가로)

```
# barplot 데이터셋
chart_data <- c(305, 450, 320, 460, 330, 480, 380, 520)
names(chart_data) <- c("2018 1 분기", "2019 1 분기",
                       "2018 2 분기", "2019 2 분기",
                       "2018 3 분기", "2019 3 분기",
                       "2018 4 분기", "2019 4 분기")
chart_data

# barplot 가로막대그래프
barplot(chart_data, xlim = c(0, 600), horiz = T,
        ylab = "매출액(단위: 만원)",
        xlab = "년도별 분기 현황",
        col = rainbow(8), space = 0.5, cex.names = 0.8,
        main = "2018 년도 vs 2019 년도 매출현황 비교")
# horiz : 수평, 가로 막대 표현 여부 T = 가로, default : 세로
# space 속성 : 막대의 굵기와 간격 지정
# cex.names 속성 : 축 이름의 크기 지정
```

**표 1. barplot code**

```
# ggplot2 데이터셋
value <- c(305, 450, 320, 460, 330, 480, 380, 520)
quarter <- c("2018 1 분기", "2019 1 분기",
             "2018 2 분기", "2019 2 분기",
             "2018 3 분기", "2019 3 분기",
             "2018 4 분기", "2019 4 분기")
chart_data2 <- data.frame(quarter = quarter, value = value)

# ggplot2 가로막대그래프
ggplot(chart_data2) +
  aes(x = quarter, weight = value, fill = quarter) +
  geom_bar() +
  coord_flip()
```

**표 2. ggplot2 code**

## 1-2) 막대그래프 (세로)

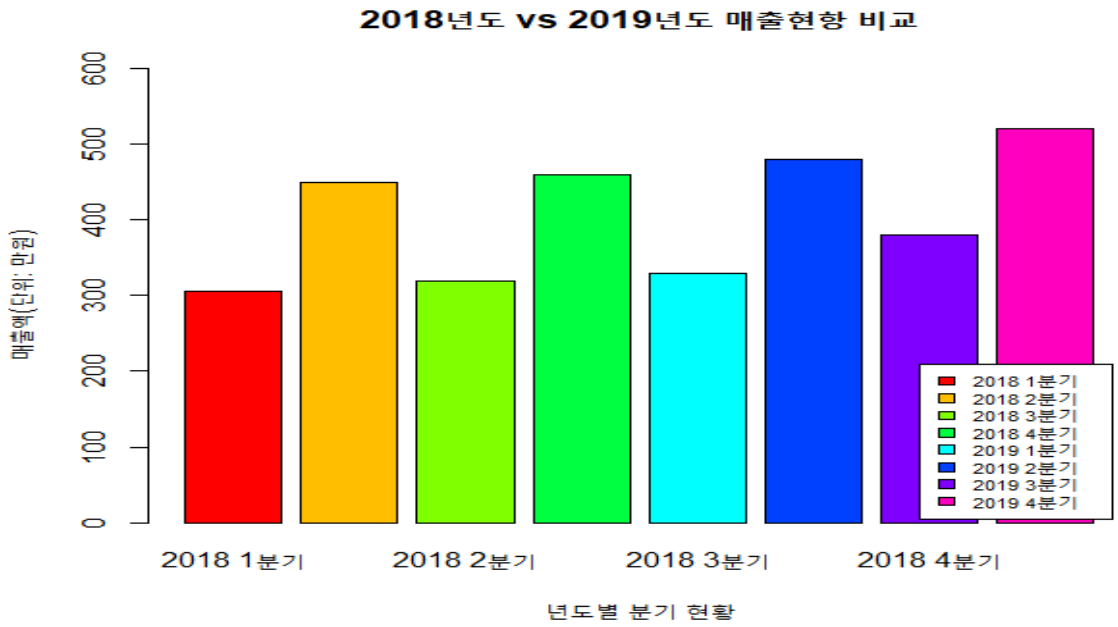


그림 1. barplot 막대그래프 (세로)

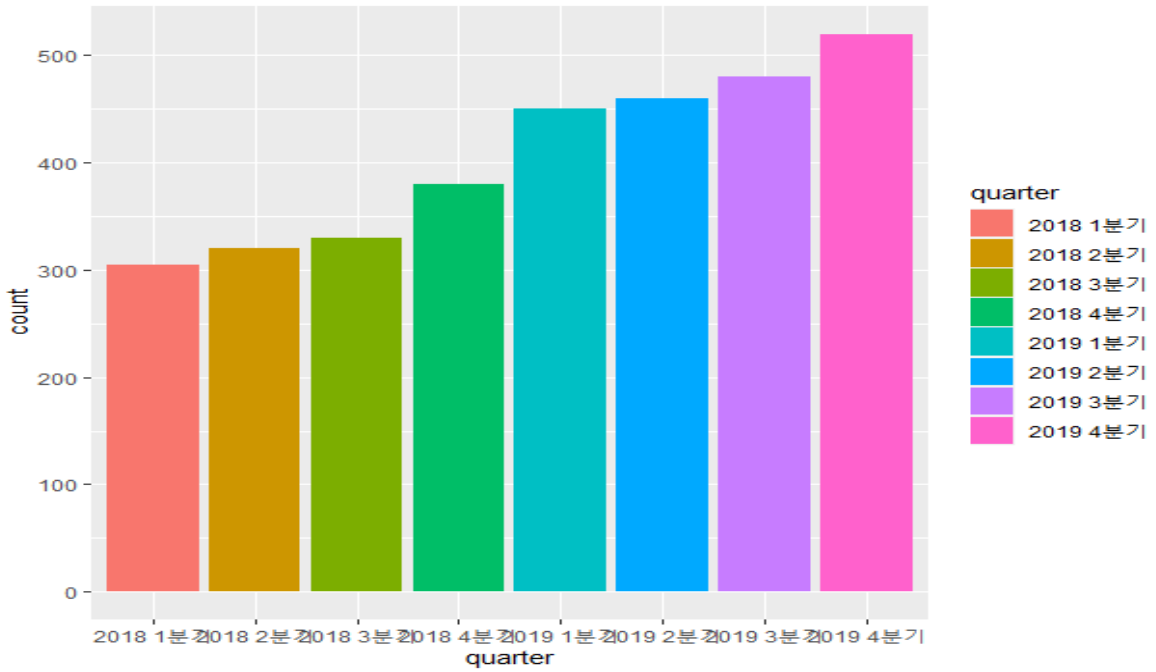


그림 2. ggplot2 막대그래프 (세로)

```
# barplot 데이터셋
chart_data <- c(305, 450, 320, 460, 330, 480, 380, 520)
names(chart_data) <- c("2018 1 분기", "2019 1 분기",
                       "2018 2 분기", "2019 2 분기",
                       "2018 3 분기", "2019 3 분기",
                       "2018 4 분기", "2019 4 분기")
chart_data

# barplot 세로막대그래프
barplot(chart_data, ylim = c(0, 600),
        ylab = "매출액(단위: 만원)",
        xlab = "년도별 분기 현황",
        col = rainbow(8),
        main = "2018 년도 vs 2019 년도 매출현황 비교")
legend(7.8, 210, c("2018 1 분기", "2018 2 분기", "2018 3 분기", "2018 4 분기",
                   "2019 1 분기", "2019 2 분기", "2019 3 분기", "2019 4 분기"),
       cex = 0.7, fill = rainbow(8))
# legend : 범례
```

**표 1. barplot code**

```
# ggplot2 데이터셋
value <- c(305, 450, 320, 460, 330, 480, 380, 520)
quarter <- c("2018 1 분기", "2019 1 분기",
             "2018 2 분기", "2019 2 분기",
             "2018 3 분기", "2019 3 분기",
             "2018 4 분기", "2019 4 분기")
chart_data2 <- data.frame(quarter = quarter, value = value)

# ggplot2 세로막대그래프
ggplot(chart_data2) +
  aes(x = quarter, weight = value, fill = quarter) +
  geom_bar()
```

**표 2. ggplot2 code**

## 2) 누적 막대그래프

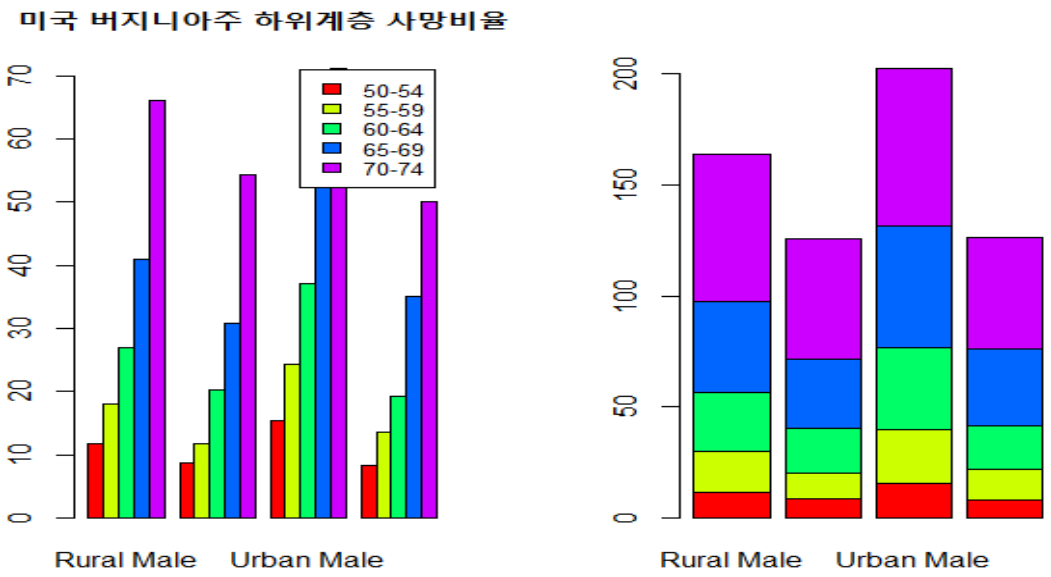


그림 1. barplot 개별 / 누적 막대그래프

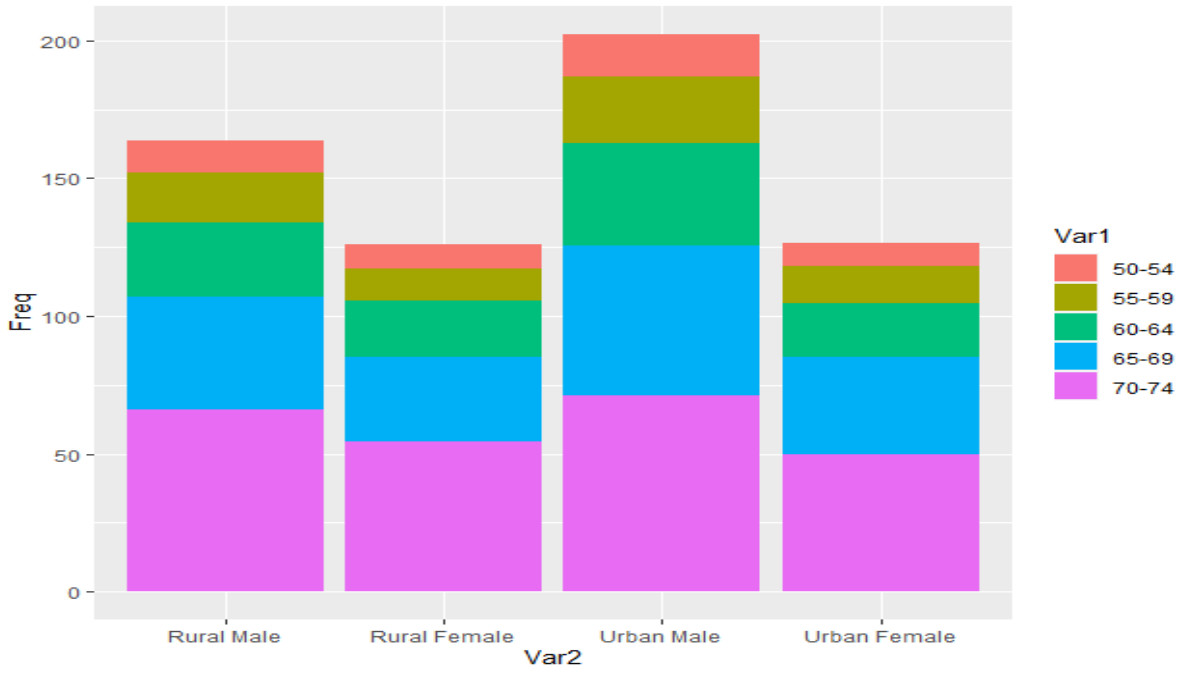


그림 2. ggplot2 누적 막대그래프

```
# 데이터 가져오기
data("VADeaths")
VADeaths

# barplot 개별 / 누적 막대그래프
par(mfrow = c(1,2))
barplot(VADeaths, beside = T, col = rainbow(5),
          main = "미국 버지니아주 하위계층 사망비율")
legend(15, 71, c("50-54", "55-59", "60-64", "65-69", "70-74"),
         cex = 0.8, fill = rainbow(5))

barplot(VADeaths, beside = F, col = rainbow(5))
title(main = "미국 버지니아주 하위계층 사망비율", font.main = 4)
legend(3.8, 200, c("50-54", "55-59", "60-64", "65-69", "70-74"),
         cex = 0.8, fill = rainbow(5))
```

**표 1. barplot code**

```
# 데이터 가져오기
data("VADeaths")
VADeaths

# ggplot 누적 막대그래프
VADeaths_df <- as.data.frame.table(VADeaths)
ggplot(VADeaths_df, aes(x=Var2, y=Freq, fill = Var1))+
  geom_bar(stat = 'identity')
```

**표 2. ggplot2 code**

### 3) 점 차트

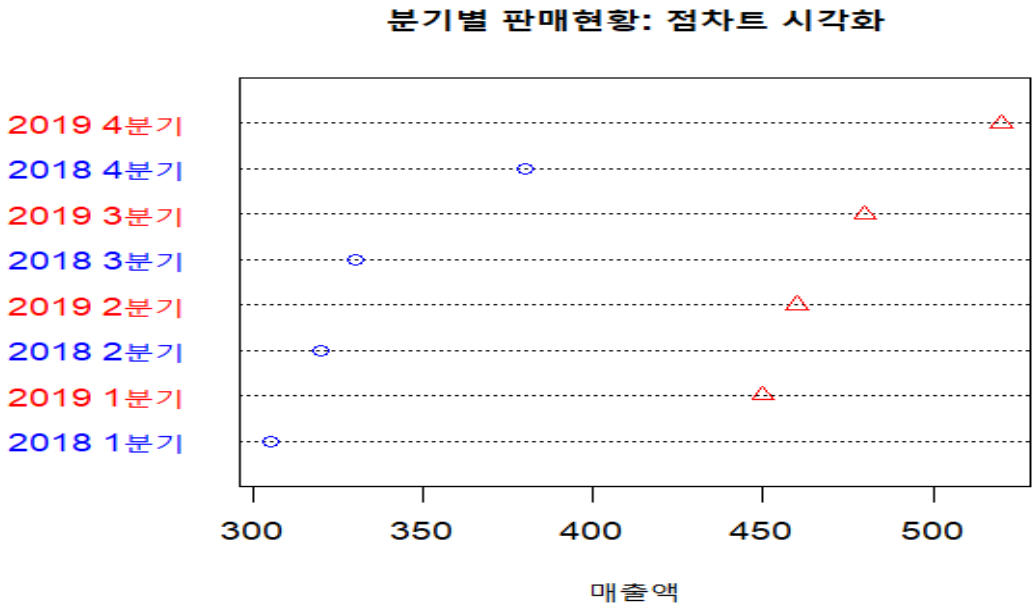


그림 1. dotchart 점 차트

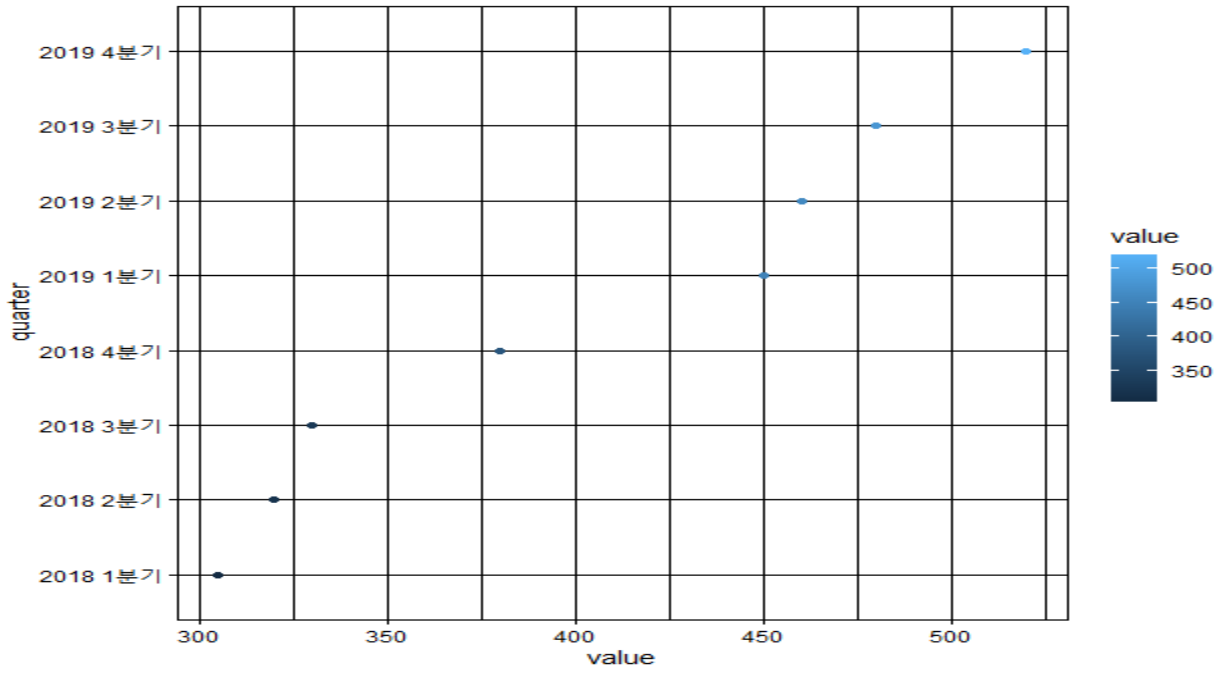


그림 2. ggplot2 점 차트

```r
# dotchart 데이터셋
chart_data <- c(305, 450, 320, 460, 330, 480, 380, 520)
names(chart_data) <- c("2018 1 분기", "2019 1 분기",
                       "2018 2 분기", "2019 2 분기",
                       "2018 3 분기", "2019 3 분기",
                       "2018 4 분기", "2019 4 분기")

# dotchart 점 차트
dotchart(chart_data, color = c("blue", "red"),
         lcolor = "black", pch = 1:2,
         labels = names(chart_data),
         xlab = "매출액",
         main = "분기별 판매현황: 점차트 시각화",
         cex = 1.2)
# col : 레이블과 점 색상 지정
# lcolor : 구분선 색상 지정
# pch : 점 모양
# labels : 점에 대한 레이블 표시
# cex : 확대
```

**표 1. dotchart code**

```r
# ggplot2 데이터셋
value <- c(305, 450, 320, 460, 330, 480, 380, 520)
quarter <- c("2018 1 분기", "2019 1 분기",
             "2018 2 분기", "2019 2 분기",
             "2018 3 분기", "2019 3 분기",
             "2018 4 분기", "2019 4 분기")
chart_data2 <- data.frame(quarter = quarter, value = value)

# ggplot2 점 차트
ggplot(chart_data2, aes(value, quarter, colour = value)) +
  geom_point()+
  theme_linedraw()
```

**표 2. ggplot2 code**

## 4) 원형 차트

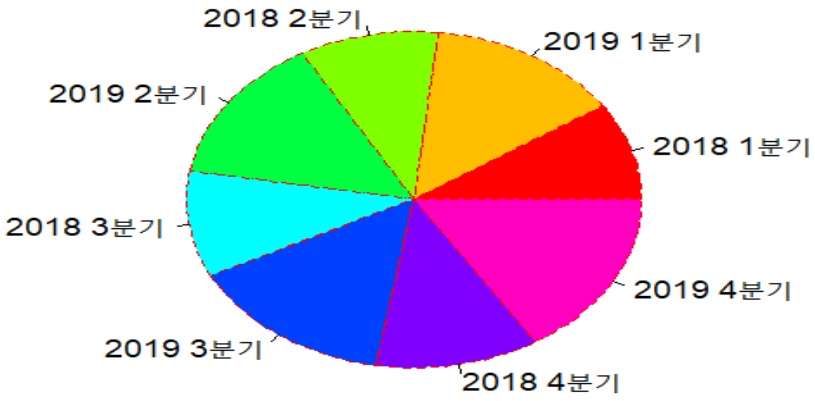


그림 1. pie 원형 차트

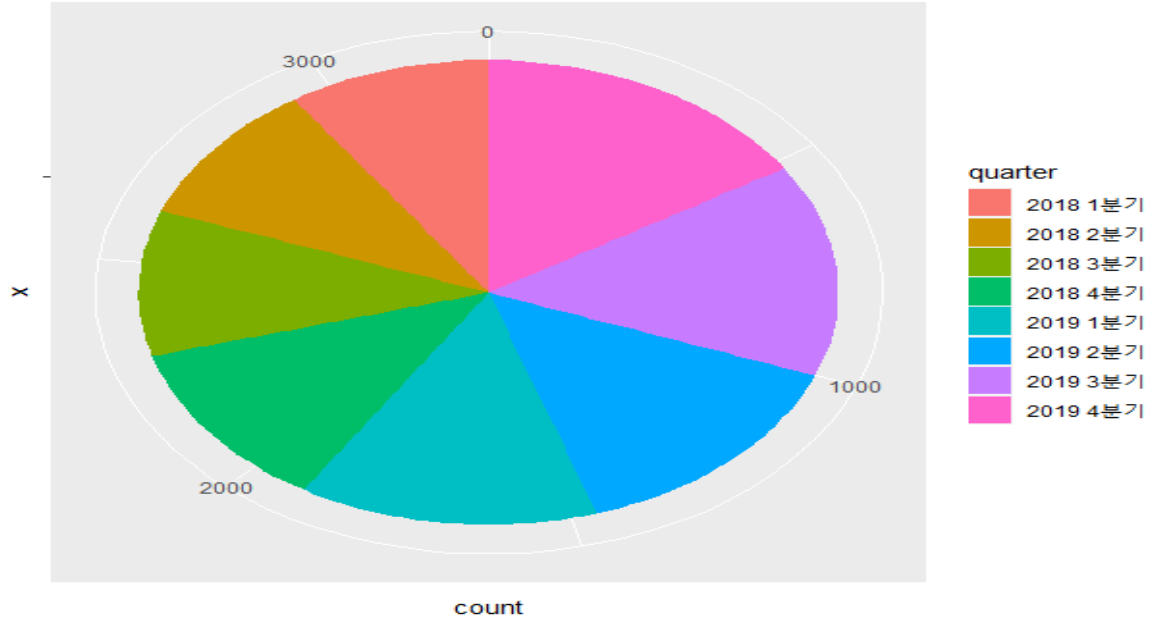


그림 2. ggplot2 원형 차트

```
# pie 데이터셋
chart_data <- c(305, 450, 320, 460, 330, 480, 380, 520)
names(chart_data) <- c("2018 1 분기", "2019 1 분기",
                       "2018 2 분기", "2019 2 분기",
                       "2018 3 분기", "2019 3 분기",
                       "2018 4 분기", "2019 4 분기")

# pie 원형 차트
pie(chart_data, labels = names(chart_data), col = rainbow(8), cex = 1.2,
    border = 2, lty = 2)
title("2018~2019 년도 분기별 매출현황")
# border : 테두리색
# lty : 점선 종류
```

**표 1. pie code**

```
# ggplot2 데이터셋
value <- c(305, 450, 320, 460, 330, 480, 380, 520)
quarter <- c("2018 1 분기", "2019 1 분기",
             "2018 2 분기", "2019 2 분기",
             "2018 3 분기", "2019 3 분기",
             "2018 4 분기", "2019 4 분기")
chart_data2 <- data.frame(quarter = quarter, value = value)

# ggplot2 원형 차트
circle <- ggplot(chart_data2) +
  aes(x='', weight = value, fill = quarter) +
  geom_bar()
circle + coord_polar('y')
```

**표 2. ggplot2 code**

## 5) 상자 그래프

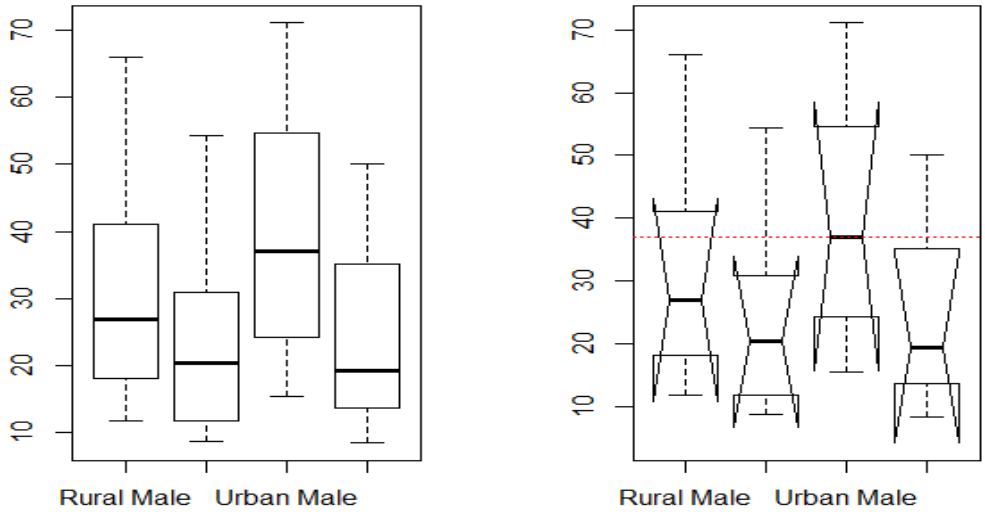


그림 1. boxplot 상자 그래프

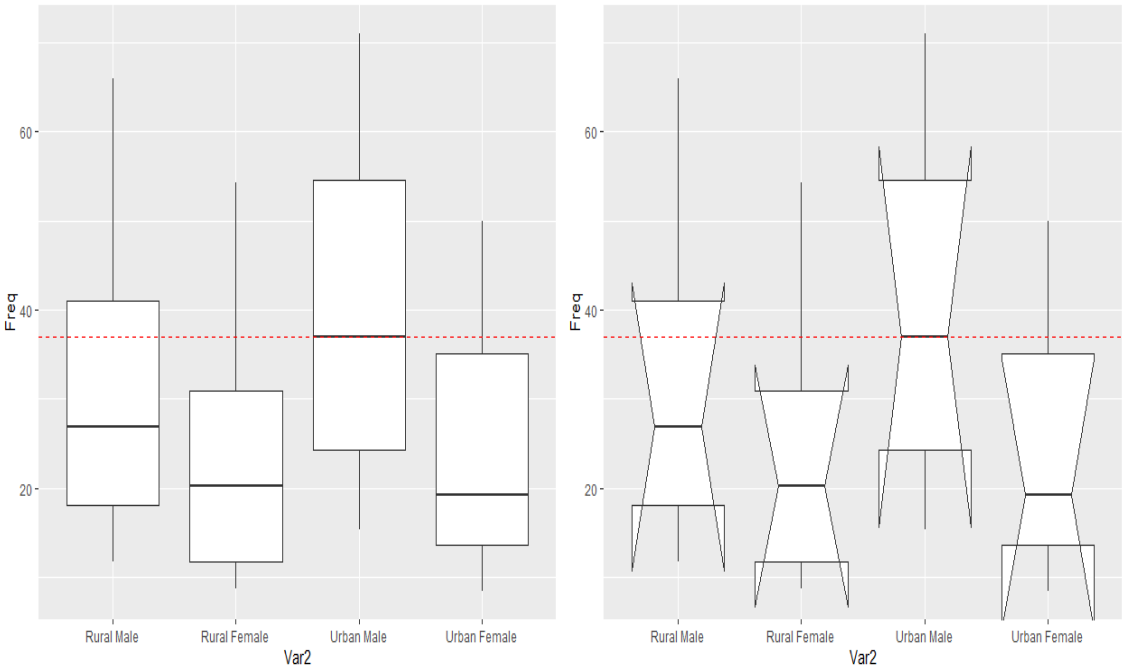


그림 2. ggplot2 상자 그래프

```r
# boxplot 데이터셋
data("VADeaths")
VADeaths

# boxplot 상자 그래프
boxplot(VADeaths, range = 0)

# notch = TRUE
boxplot(VADeaths, range = 0, notch = T)
abline(h = 37, lty = 3, col = "red")
# h : 선 그을 위치
# notch = T : 중위수 기준 허리선 추가
```

**표 1. boxplot code**

```r
# ggplot2 데이터셋
data("VADeaths")
VADeaths

# ggplot2 상자 그래프
VADeaths_df <- as.data.frame.table(VADeaths)
box <- ggplot(VADeaths_df, aes(x = Var2, y = Freq))
box <- box + geom_boxplot(notch = F) +
  geom_abline(intercept = 37, slope = 0, color = 2, linetype = 'dashed')
box

VADeaths_df <- as.data.frame.table(VADeaths)
box <- ggplot(VADeaths_df, aes(x = Var2, y = Freq))
box <- box + geom_boxplot(notch = T) +
  geom_abline(intercept = 37, slope = 0, color = 2, linetype = 'dashed')
box
```

**표 2. ggplot2 code**

### 6) 히스토그램

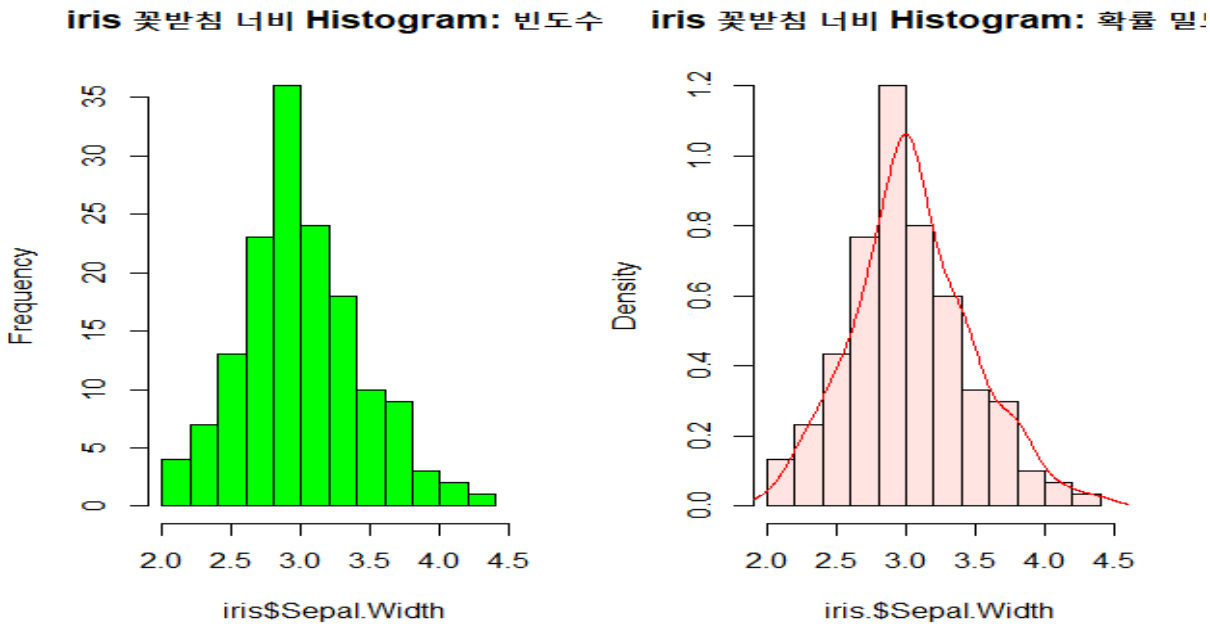


그림 1. hist 히스토그램

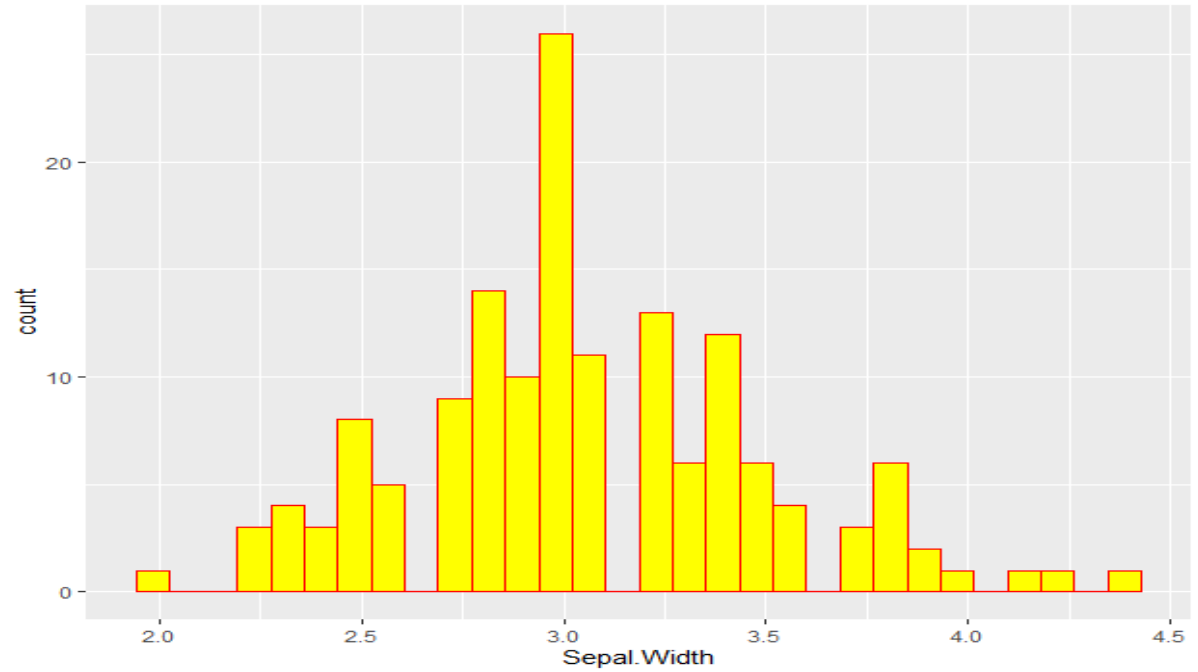


그림 2. ggplot2 히스토그램

```r
# hist 데이터셋
data(iris)

# 빈도수에 의해서 히스토그램 그리기
par(mfrow = c(1, 2))
hist(iris$Sepal.Width, xlab = "iris$Sepal.Width",
     col = "green",
     main = "iris 꽃받침 너비 Histogram: 빈도수", xlim = c(2.0, 4.5))

# 확률 밀도에 의해서 히스토그램 그리기
hist(iris$Sepal.Width, xlab = "iris.$Sepal.Width",
     col = "mistyrose", freq = F,
     main = "iris 꽃받침 너비 Histogram: 확률 밀도", xlim = c(2.0, 4.5))

# 밀도를 기준으로 line 추가하기
lines(density(iris$Sepal.Width), col = "red")
```

**표 1. hist code**

```r
# ggplot2 데이터셋
data(iris)

# ggplot2 히스토그램
hist2 <- ggplot(data = iris, aes(x = Sepal.Width))
hist2 + geom_histogram(fill = 7, color = 2)
# fill : 막대 색 채우기
# color : 막대 테두리 색
```

**표 2. ggplot2 code**

## 7) 산점도

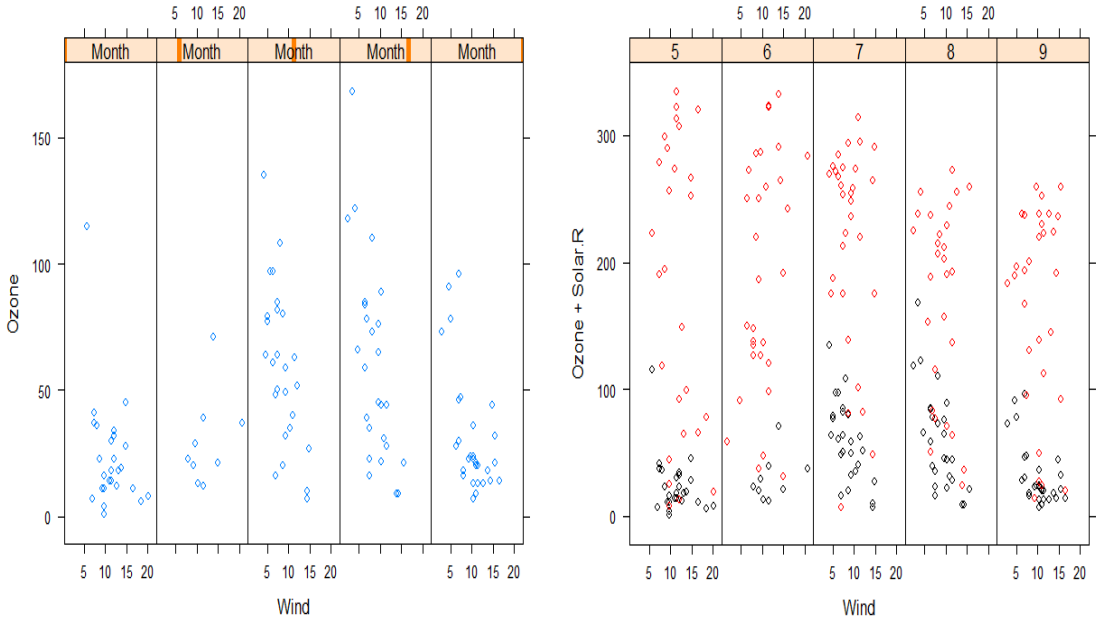


그림 1. xyplot 산점도 (y 축 변수 1 개 / 2 개)

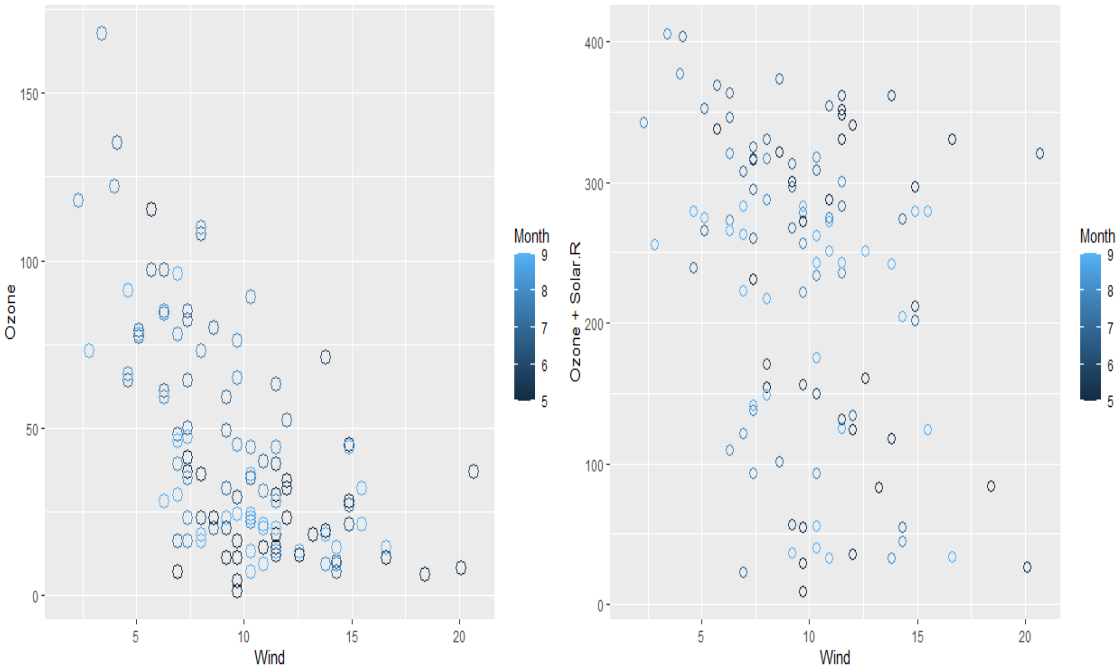


그림 2. ggplot2 산점도 (y 축 변수 1 개 / 2 개)

```r
# xyplot 데이터셋
library(datasets)
data("airquality")

# xyplot 산점도
xyplot(Ozone ~ Wind | Month, data = airquality, layout=c(5,1))
# y 축 ~ x 축 컬럼 | 조건, 사용할 데이터, 레이아웃

# 2 개 변수를 y 축에 표시
xyplot(Ozone + Solar.R ~ Wind | factor(Month),
       data = airquality,
       col = c("black", "red"),
       layout = c(5, 1))
```

**표 1. xyplot code**

```r
# ggplot2 데이터셋
data("airquality")

# ggplot2 산점도
air <- ggplot(data = airquality, aes(x = Wind, y = Ozone, color = Month))
air <- air+ geom_point(shape=1, size=4)
air
# shape : 점 모양
# size : 점 크기

# y 축 변수 2 개 합
air <- ggplot(data = airquality, aes(x = Wind, y = Ozone + Solar.R, color = Month))
air <- air+ geom_point(shape = 1, size = 3)
air
```

**표 2. ggplot2 code**

## 8) 중첩 자료 시각화

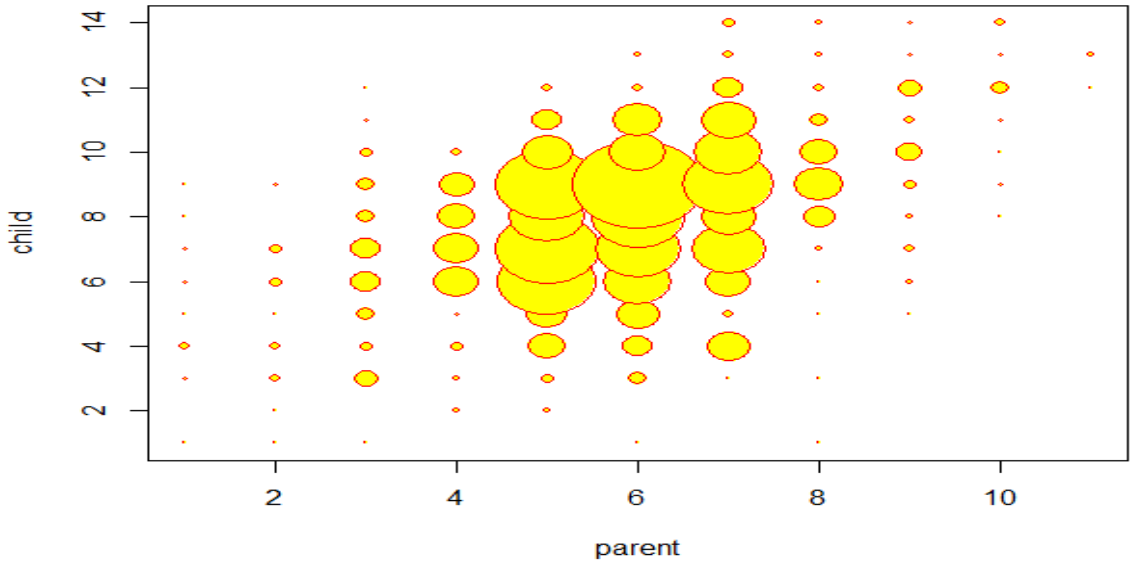


그림 1. plot 중첩 자료 시각화

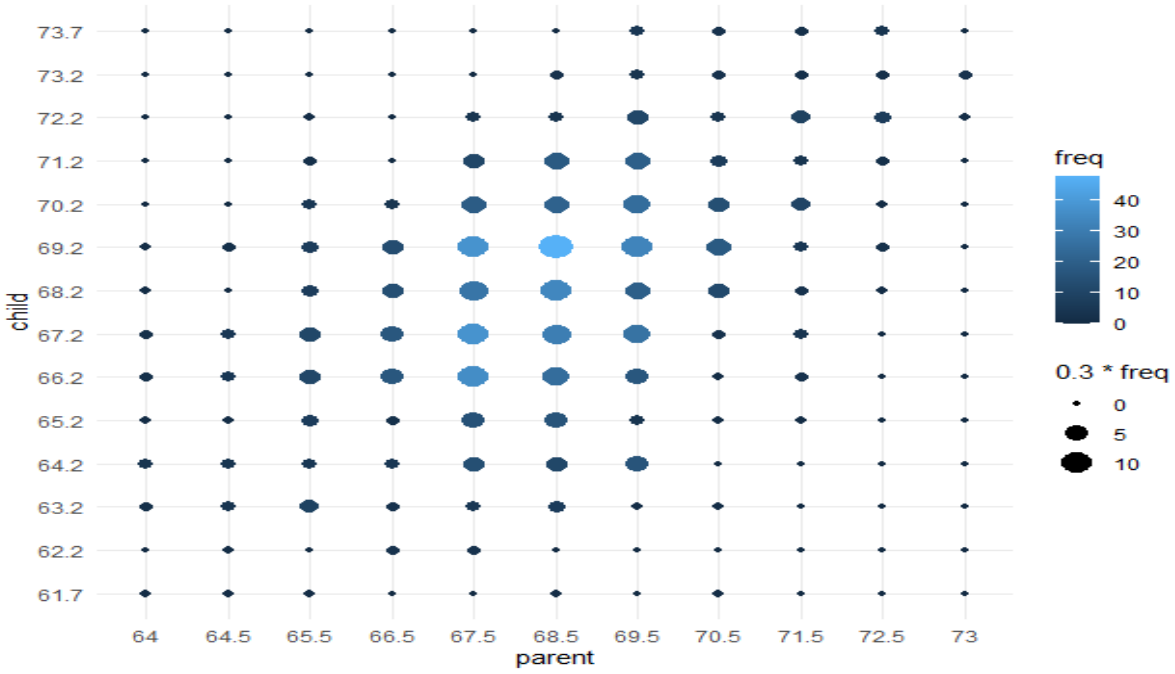


그림 2. ggplot2 중첩 자료 시각화

```
# plot 데이터셋
library(UsingR)
data(galton)
galtonData <- as.data.frame(table(galton$child, galton$parent))

# plot 중첩 자료 시각화
names(galtonData) = c("child", "parent", "freq")
parent <- as.numeric(galtonData$parent)
child <- as.numeric(galtonData$child)

plot(parent, child,
        pch = 21, col = "red", bg = "yellow",
        cex = 0.2 * galtonData$freq,
        xlab = "parent", ylab = "child")
# col : 도형 테두리 색
# bg : 도형 내부 색
```

**표 1. plot code**

```
# ggplot2 데이터셋
data(galton)
galtonData <- as.data.frame(table(galton$child, galton$parent))

# ggplot2 중첩 자료 시각화
names(galtonData) = c("child", "parent", "freq")
galton2 <- ggplot(data = galtonData)
galton2 + aes(x = parent, y = child, colour = freq) +
  geom_point(mapping = aes(size = 0.3 * freq)) +
  scale_color_gradient() + theme_minimal()
```

**표 2. ggplot2 code**

## 9) 변수 간의 비교 시각화

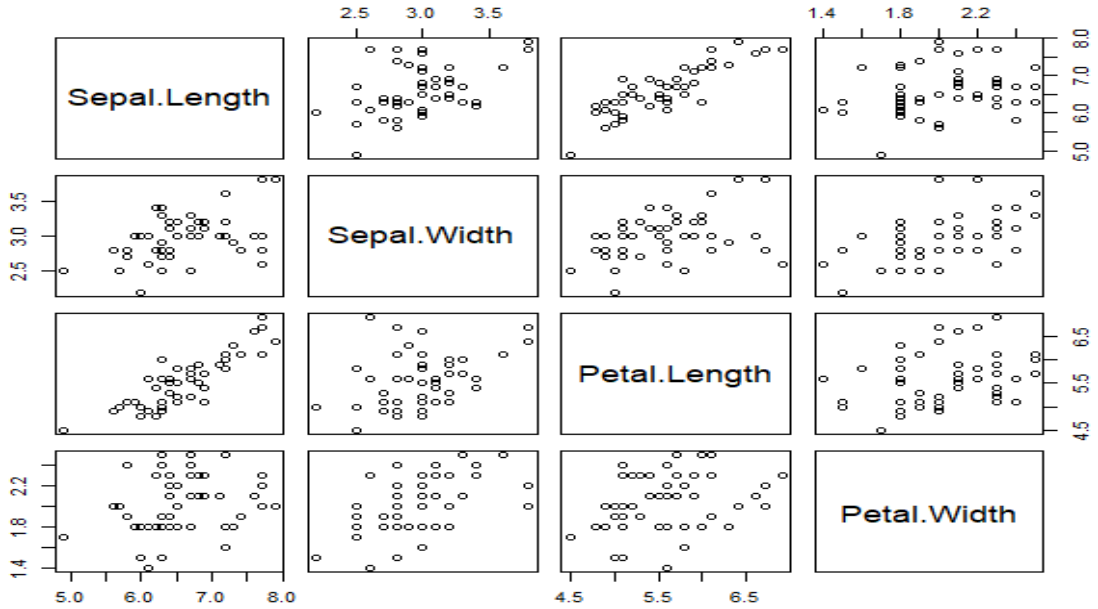


그림 1. pairs 변수 간의 비교 시각화

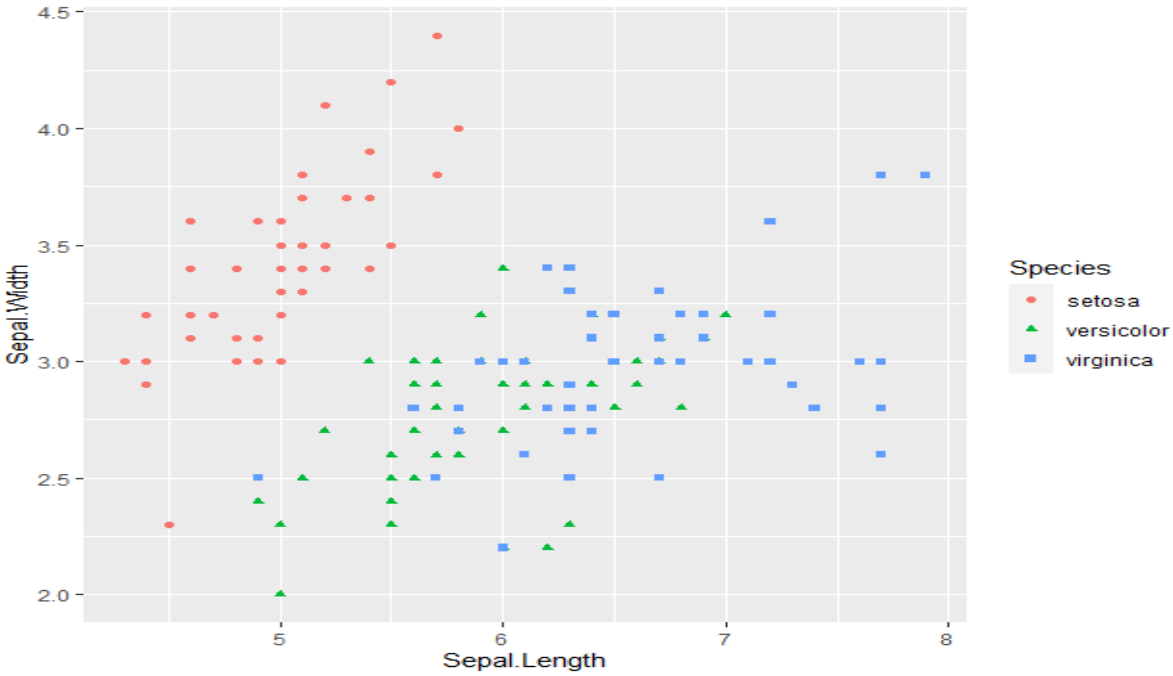


그림 2. ggplot2 변수 간의 비교 시각화

```
# pairs 데이터셋
data(iris)

# pairs 변수 간의 비교 시각화
pairs(iris[iris$Species == "virginica", 1:4])
pairs(iris[iris$Species == "setosa", 1:4])
pairs(iris[iris$Species == "versicolor", 1:4])
# pairs() : numeric 컬럼 대상 변수들 사이
# 비교 결과를 행렬구조의 분산된 그래프로 제공
```

**표 1. pairs code**

```
# ggplot2 데이터셋
library(ggplot2)
data(iris)

# ggplot2 변수 간의 비교 시각화
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width,
                           color = Species, shape = Species), data=iris)
```

**표 2. ggplot2 code**

## 10) 밀도 그래프

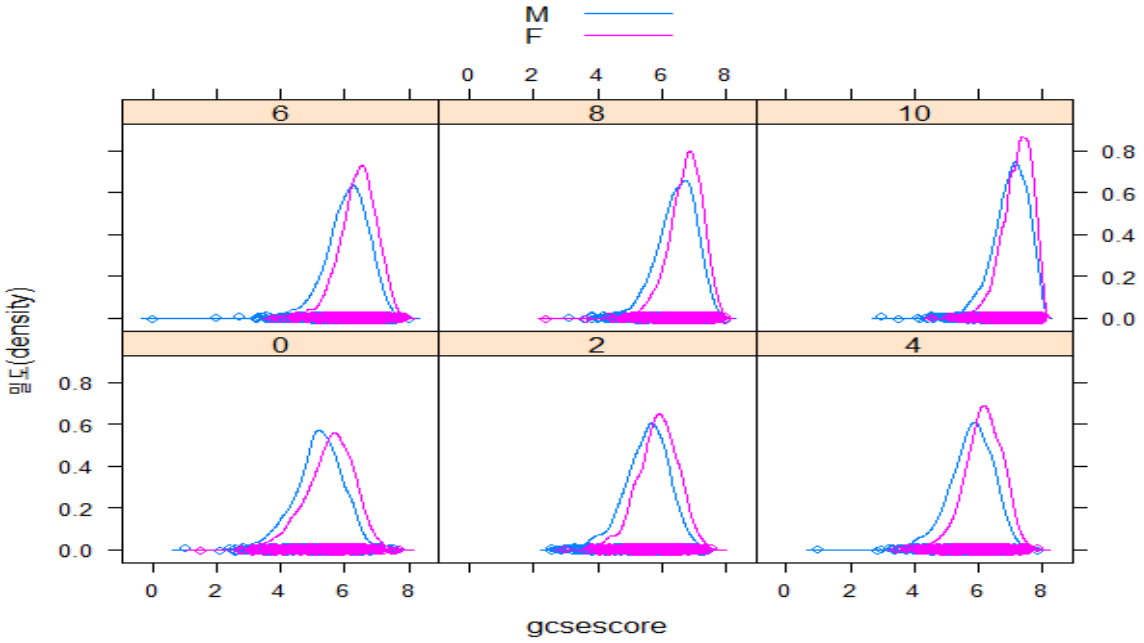


그림 1. density 밀도 그래프

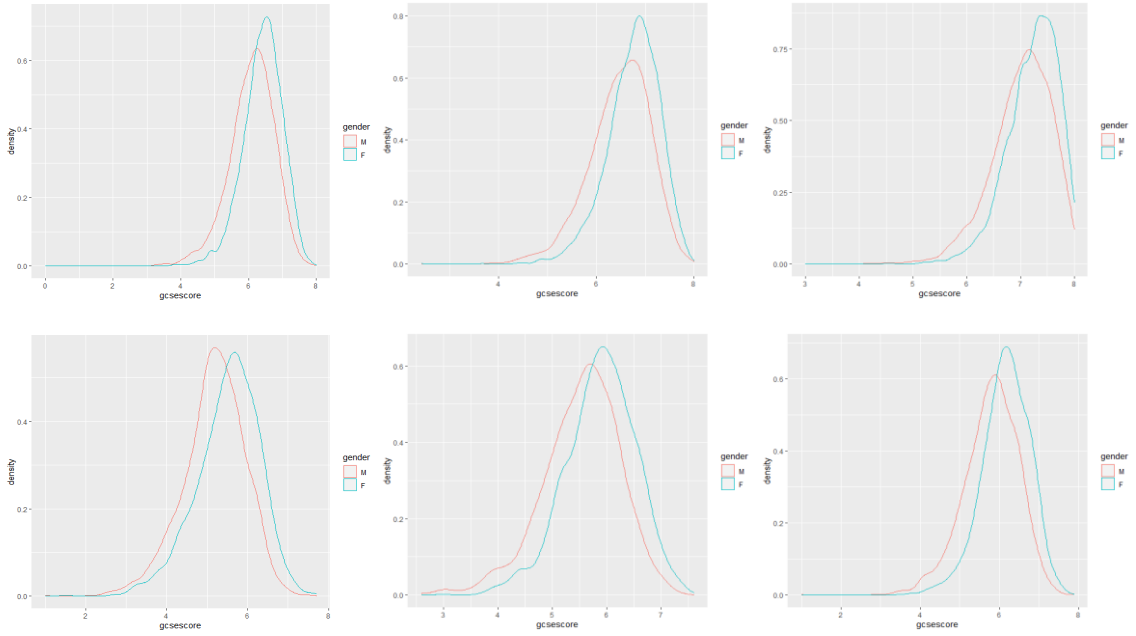


그림 2. ggplot2 밀도 그래프 (성별 / 점수 0~10 점)

```
# densityplot 데이터셋
library(lattice)
library(mlmRev)
data("Chem97")

# densityplot 밀도 그래프
densityplot(~gcsescore | factor(score), data = Chem97,
            groups = gender, plot.Points = T,
            auto.key = T)
```

**표 1. densityplot code**

```
# ggplot2 데이터셋
library(lattice)
library(mlmRev)
data("Chem97")

# score 기준 데이터 분리
chem97_0<-subset(Chem97, score == 0)
chem97_2<-subset(Chem97, score == 2)
chem97_4<-subset(Chem97, score == 4)
chem97_6<-subset(Chem97, score == 6)
chem97_8<-subset(Chem97, score == 8)
chem97_10<-subset(Chem97, score == 10)

# ggplot2 밀도 그래프(점수 / 성별)
ggplot(data = chem97_0) +   geom_density(mapping = aes(x = gcsescore, colour = gender))
ggplot(data = chem97_2) +   geom_density(mapping = aes(x = gcsescore, colour = gender))
ggplot(data = chem97_4) +   geom_density(mapping = aes(x = gcsescore, colour = gender))
ggplot(data = chem97_6) +   geom_density(mapping = aes(x = gcsescore, colour = gender))
ggplot(data = chem97_8) +   geom_density(mapping = aes(x = gcsescore, colour = gender))
ggplot(data = chem97_10) +   geom_density(mapping = aes(x = gcsescore, colour = gender))
```

**표 2. ggplot2 code**

# 3. 분석 결과 및 결론

## 1) 패키지별 상이점, 장/단점 비교

R 의 내장함수와 ggplot2 패키지로 시각화를 진행 후 어떤 식으로 시각화가 되는지 비교해보았다.

R 의 내장함수는 별도의 패키지를 사용하지 않아도 된다는 장점과 변수를 직접적으로 지정해주어 직관성이 뛰어 나다는 느낌을 받았다. 하지만 이러한 방식들로 볼 때 대량의 데이터에서보다는 소량의 데이터에서 효과적이고 편하게 사용 할 수 있을 것 같다.

ggplot2 패키지는 사용 할 수 있는 옵션이 너무 많아 처음 사용 시 일일이 찾아보고 이해해야 할게 많아 사용이 어려웠지만, 이를 모두 사용할 정도로 능숙해진다면 어느 패키지보다 세세하고 직관적으로 시각화가 가능할 것이다.

R 의 내장함수와 ggplot2 패키지 각각의 장단점이 있지만, 딱 어떤 패키지가 더 좋다고 정하지 않고 시각화 대상이 되는 데이터셋에 따라 적절한 패키지를 사용하여 사용자 입장에서 쉽게 이해할 수 있고 직관적으로 표현하는 것이 빅데이터 분석가의 임무이자 실력이 될 것이다.

## 참고 자료

1) VADeaths 데이터셋

2) Iris 데이터셋

3) Airquality 데이터셋

4) Galton 데이터셋