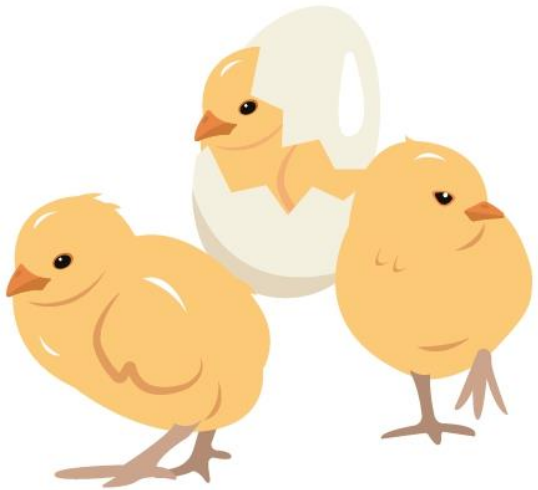
모두를 위한

R 데이터 분석 입문

2판



Chapter 08
데이터 시각화

목차

Section 01

데이터 시각화 기법

1. 데이터 시각화의 중요성

- **데이터 시각화(data visualization) :** 숫자 형태의 데이터를 그래프나 그림 등의 형태로 표현하는 과정
- 데이터 분석 과정에서 중요한 기술 중의 하나
- 데이터를 시각화 하면 데이터가 담고 있는 정보나 의미를 보다 쉽게 파악
- 경우에 따라서는 시각화 결과로부터 중요한 영감을 얻기도 함

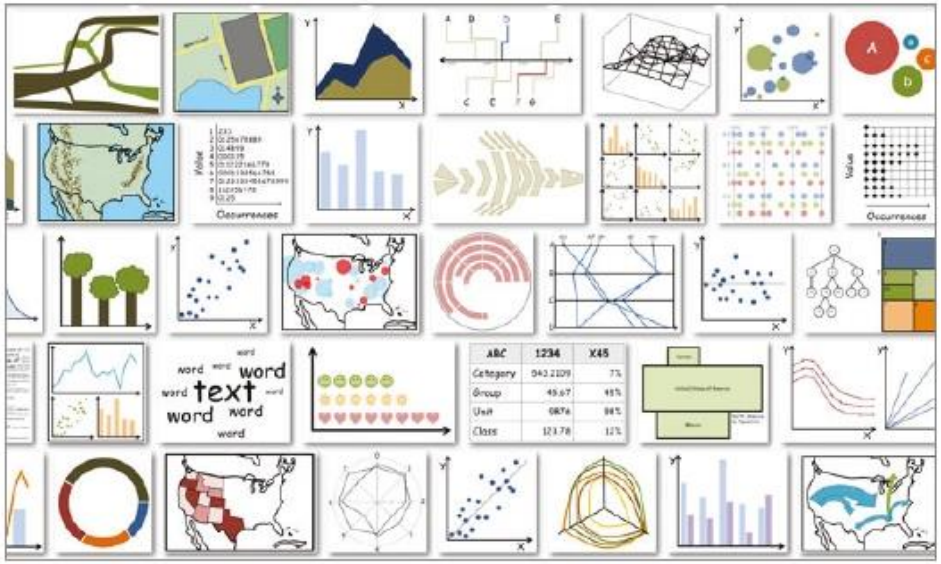


그림 8-1 다양한 데이터 시각화의 사례들

2. 트리맵

2.1 GNI2014 데이터셋으로 트리맵 작성하기

- 사각타일 형태로 구성되어 있으며, 각 타일의 크기와 색깔로 데이터 크기를 나타냄
- 각각의 타일은 계층 구조가 있기 때문에 데이터에 존재하는 계층 구조도 표현
- treemap 패키지 설치 필요
- 예제 데이터셋 : treemap 패키지 안에 포함된 GNI2014. 2014년도의 전 세계 국가별 인구, 국민총소득(GNI), 소속 대륙의 정보를 담고 있음

코드 8-1

```
library(treemap)                                  # treemap 패키지 불러오기
data(GNI2014)                                     # 데이터 불러오기
head(GNI2014)                                     # 데이터 내용보기
treemap(GNI2014,
            index=c("continent","iso3"),          # 계층구조 설정(대륙-국가)
            vSize="population",                   # 타일의 크기
            vColor="GNI",                         # 타일의 컬러
            type="value",                         # 타일 컬러링 방법
            title="World's GNI")                  # 트리맵 제목
```

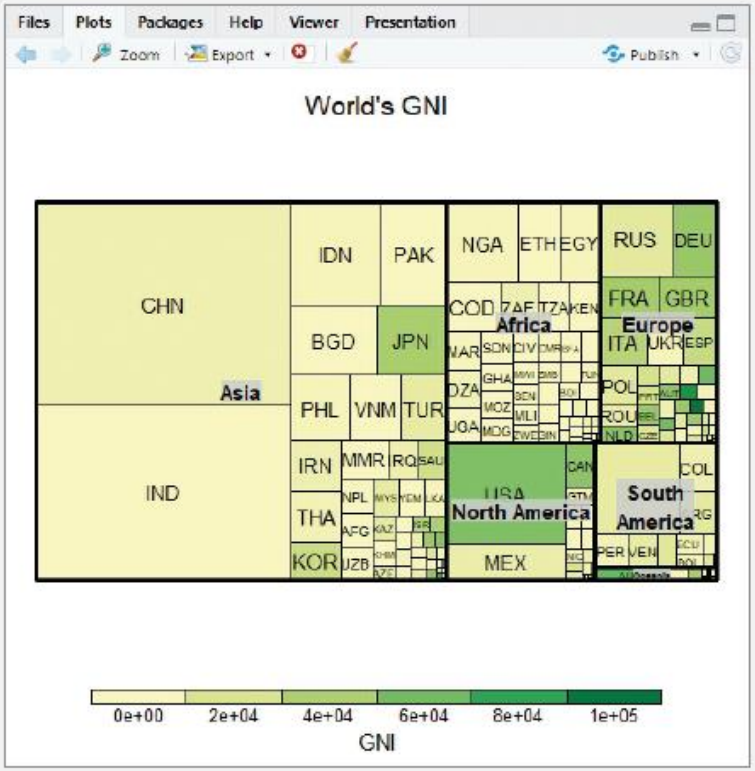
```
> library(treemap)                    # treemap 패키지 불러오기
> data(GNI2014)                       # 데이터 불러오기
> head(GNI2014)                       # 데이터 내용보기
  iso3         country     continent population    GNI
3  BMU         Bermuda North America      67837 106140
4  NOR          Norway        Europe    4676305 103630
5  QAT           Qatar          Asia     833285  92200
6  CHE     Switzerland        Europe    7604467  88120
7  MAC Macao SAR, China          Asia     559846  76270
8  LUX      Luxembourg        Europe     491775  75990
```

표 8-1 GNI2014 데이터셋에 포함된 각 열의 의미

열의 이름	의미
iso3	국가를 식별하는 표준 코드
country	국가명
continent	국가가 속한 대륙명
population	국가의 인구
GNI	국가의 국민총소득

1. 데이터 시각화 기법

```
> treemap(GNI2014,
+         index=c("continent","iso3"),   # 계층구조 설정(대륙-국가)
+         vSize="population",            # 타일의 크기
+         vColor="GNI",                  # 타일의 컬러
+         type="value",                  # 타일 컬러링 방법
+         title="World's GNI")           # 트리맵 제목
>
```



1. 데이터 시각화 기법

- `GNI2014`

  트리맵을 그릴 대상이 되는 데이터셋이다. 데이터프레임 형태여야 한다.

- `index=c("continent","iso3")`

  트리맵상에서 타일들이 대륙(continent) 안에 국가(iso3)의 형태로 배치되는 것을 지정한다.

- `vSize="population"`

  타일의 크기를 결정하는 열을 지정하는데, 여기서는 인구수(population)로 지정하였다.

- `vColor="GNI"`

  타일의 색을 결정하는 열을 지정하는데, 여기서는 소득(GNI)으로 지정하였다.

- `type="value"`

  타일의 컬러링 방법을 지정하는 것으로 "value"는 vColor에서 지정한 열에 저장된 값의 크기에 의해 색이 결정됨을 의미한다. "value" 외에도 "index", "comp", "dens" 등을 지정할 수 있다.

- `title="World's GNI"`

  트리맵의 제목을 지정한다.

2.2 state.x77 데이터셋으로 트리맵 작성하기

코드 8-2

```
library(treemap)                                  # treemap 패키지 불러오기
st <- data.frame(state.x77)     # 매트릭스를 데이터프레임으로 변환
st <- data.frame(st, stname=rownames(st)) # 주 이름 열 stname을 추가

treemap(st,
 index=c("stname"),                               # 타일에 주 이름 표기
 vSize="Area",                                    # 타일의 크기
 vColor="Income",                                 # 타일의 컬러
 type="value",                                    # 타일 컬러링 방법
 title="USA states area and income" )   # 트리맵의 제목
```

1. 데이터 시각화 기법


Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
USA states area and income
Alaska
Arizona
Nevada
Colorado
Wyoming
Oregon
Nebraska
South Dakota
North Dakota
Missouri
Idaho
Oklahoma
Illinois
Wisconsin
Florida
Texas
California
Utah
Washington
Arkansas
Alabama
North Carolina
New York
Georgia
Mississippi
Tennessee
Ohio
Virginia
Kansas
Michigan
Pennsylvania
Kentucky
Maine
South Carolina
Montana
New Mexico
Minnesota
Iowa
Louisiana
Indiana
West Virginia
Hawaii
Maryland
3000
3500
4000
4500
5000
5500
6000
6500
Income

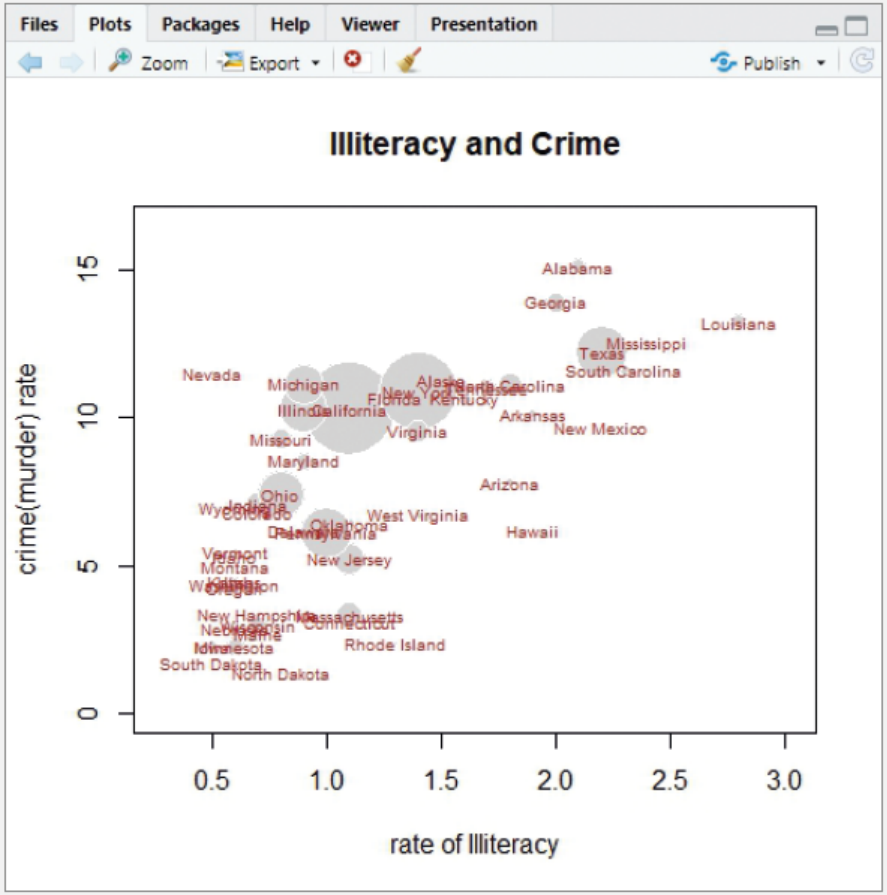
3. 버블차트

- 버블 차트(bubble chart): 앞에서 배운 산점도 위에 버블의 크기로 정보를 표시하는 시각화 방법
- 산점도가 2개의 변수에 의한 위치 정보를 표시한다면, 버블 차트는 3개의 변수 정보를 하나의 그래프에 표시

코드 8-3

```
st <- data.frame(state.x77)     # 매트릭스를 데이터프레임으로 변환
symbols(st$Illiteracy, st$Murder,          # 원의 x, y 좌표의 열
        circles=st$Population,             # 원의 반지름의 열
        inches=0.3,                        # 원의 크기 조절값
        fg="white",                        # 원의 테두리 색
        bg="lightgray",                    # 원의 바탕색
        lwd=1.5,                           # 원의 테두리선 두께
        xlab="rate of Illiteracy",
        ylab="crime(murder) rate",
        main="Illiteracy and Crime")
text(st$Illiteracy, st$Murder,             # 텍스트가 출력될 x, y 좌표
        rownames(st),                      # 출력할 텍스트
        cex=0.6,                           # 폰트 크기
        col="brown")                       # 폰트 컬러
```

1. 데이터 시각화 기법



- 전반적으로 문맹률이 높아질수록 범죄율이 증가하는 추세
- 인구수가 많은 주가 대체로 범죄율도 높은 것을 확인
- 범죄율이 가장 낮은 주는 North Dakota

1. 데이터 시각화 기법

- `st$Illiteracy, st$Murder`

  2차원 좌표의 x축과 y축으로 나타낼 열을 지정한다(여기서 x축은 문맹률, y축은 범죄율(살인율)). x축의 값과 y축의 값이 만나는 지점에 원이 그려진다.

- `circles=st$Population`

  원의 크기(반지름)를 결정할 열을 지정한다(여기서는 인구수).

- `inches=0.3`

  원의 크기를 조절하는 매개변수로, 매개변수값이 클수록 원이 크게 그려진다.

- `fg="white"`

  원의 테두리선 색을 지정한다.

- `bg="lightgray"`

  원의 바탕색을 지정한다.

- `lwd=1.5`

  원의 테두리선 두께를 지정한다.

- `xlab="rate of Illiteracy"`

  x축의 레이블을 지정한다.

- `ylab="crime(murder) rate"`

  y축의 레이블을 지정한다.

- `main="Illiteracy and Crime"`

  그래프의 제목을 지정한다.

- `st$Illiteracy, st$Murder`

  텍스트를 표시할 위치에 대한 x축과 y축 좌표값을 나타내는데, symbols() 함수에 있는 원의 x축과 y축 좌표값과 일치시킨다.

- `rownames(st)`

  표시할 텍스트를 지정한다. st의 행 이름은 미국 각 주의 이름이다.

- `cex=0.6`

  텍스트의 크기를 지정한다.

- `col="brown"`

  텍스트의 색을 지정한다.

4. 모자이크 플롯

- **모자이크 플롯(mosaic plot):** 다중변수 범주형 데이터에 대해 각 변수의 그룹별 비율을 면적으로 표시하여 정보를 전달

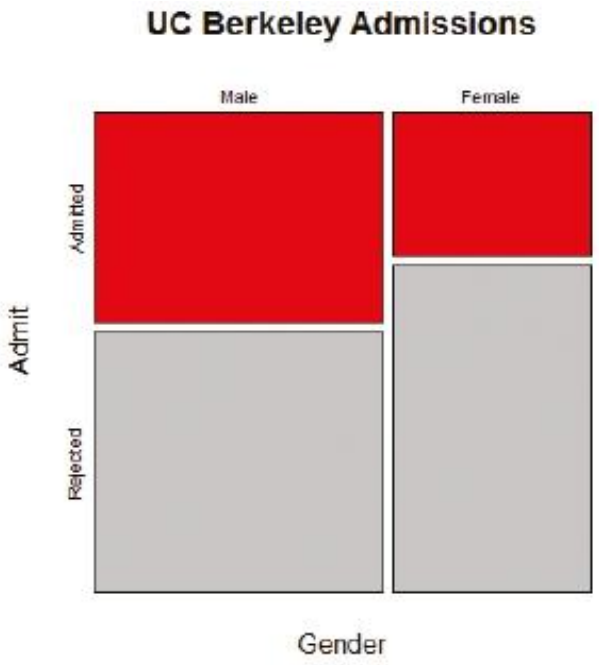
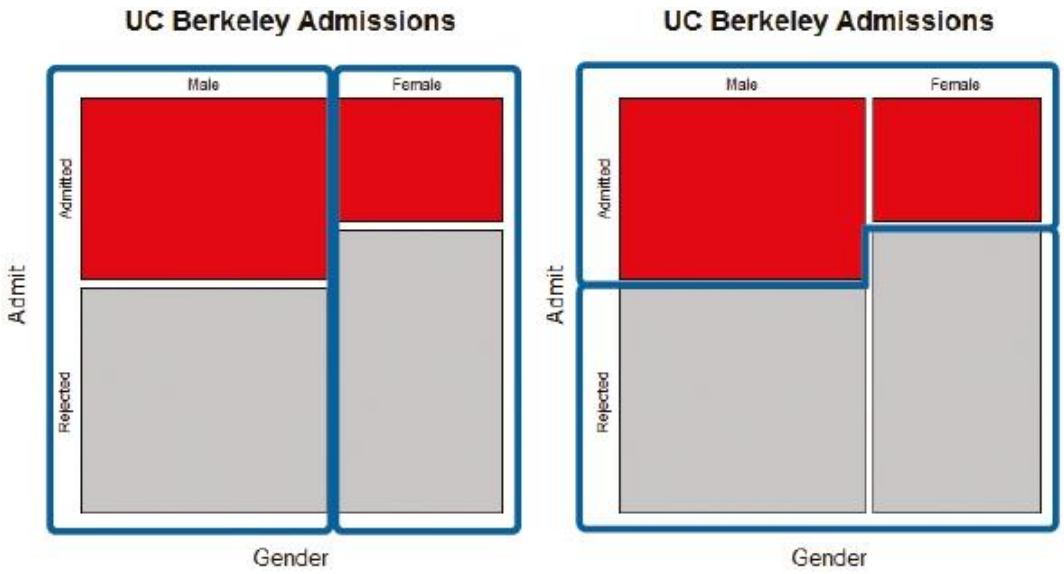


그림 8-2 모자이크 플롯의 예

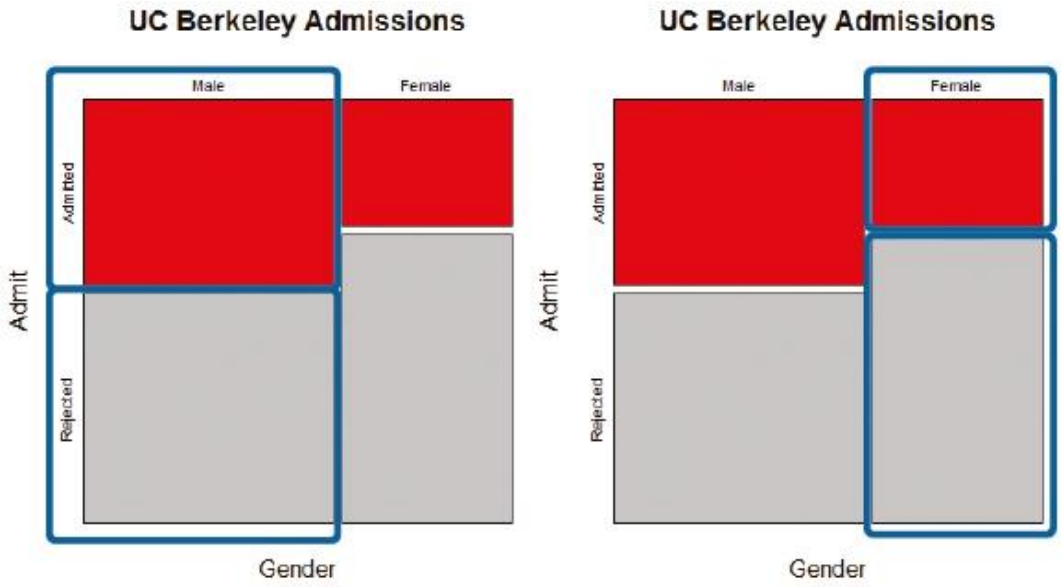
- 예제 데이터: UCBAdmissions
- 미국의 버클리대학교 대학원의 지원자와 합격자 통계를 성별, 학과별로 정리
- 아래는 지원자와 합격자 통계를 성별로 구분하여 모자이크 플롯으로 나타낸 것



(a) 남성 지원자와 여성 지원자의 비율 (b) 합격자와 불합격자의 비율

그림 8-3 모자이크 플롯의 해석 1

- 왼쪽의 전체 면적이 남성(male) 지원자의 수를 나타내고, 오른쪽의 전체 면적이 여성(female) 지원자의 수를 나타냄
- 남성 지원자의 수가 여성 지원자 수에 비해 1.5배 정도 많음
- 위쪽 빨간색 면적은 합격자의 수를, 아래쪽 회색 면적은 불합격자의 수를 나타냄
- 전체 지원자에서 합격자의 비율이 50%가 안 되는 것을 확인



(a) 남성 지원자의 합격자와 불합격자 비율 (b) 여성 지원자의 합격자와 불합격자 비율

그림 8-4 모자이크 플롯의 해석 2

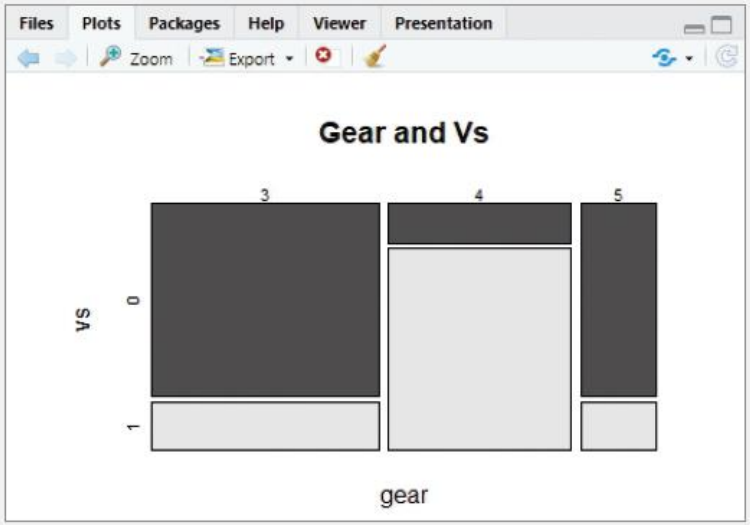
- 남성 지원자의 합격자 비율과 불합격자 비율
- 여성 지원자의 합격자 비율과 불합격자 비율
- 여성 지원자의 합격률이 남성 지원자의 합격률보다 눈에 띄게 낮음

코드 8-4

```
head(mtcars)
mosaicplot(~gear+vs, data = mtcars, color=TRUE,
           main ="Gear and Vs")
```

```
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
> mosaicplot(~gear+vs, data = mtcars, color=TRUE,
+           main ="Gear and Vs")
```

1. 데이터 시각화 기법



- `~gear+vs`

  모자이크 플롯을 작성할 대상 변수를 지정한다. ~ 다음의 변수가 x축 방향으로 표시되고, + 다음의 변수가 y축 방향으로 표시된다.

- `data = mtcars`

  모자이크 플롯을 작성할 대상 데이터셋을 지정한다.

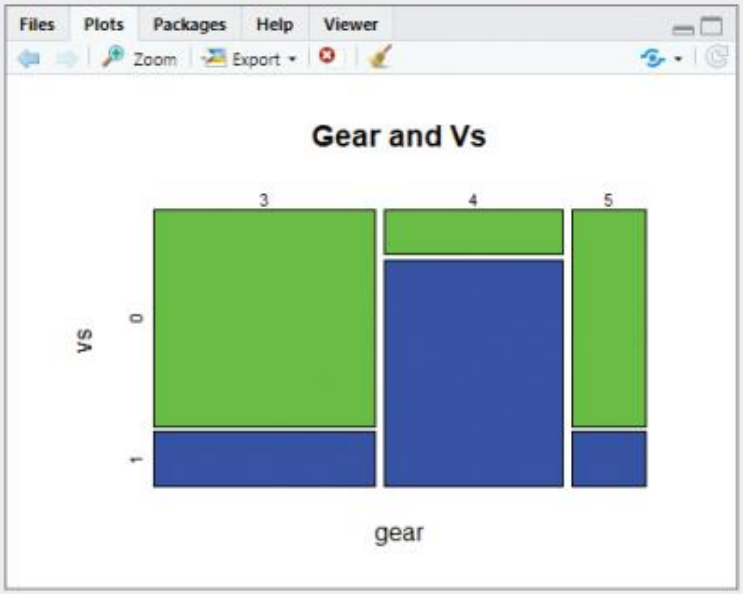
- `color=TRUE`

  y축 변수의 그룹별로 음영을 달리하여 표시한다.

- `main ="Gear and Vs"`

  모자이크 플롯의 제목을 지정한다.

```
> mosaicplot(~gear+vs, data = mtcars, color=c("green","blue"),
+            main ="Gear and Vs")
```



Section 02

ggplot 패키지

2. ggplot 패키지

- 지금까지는 그래프를 작성할 때 주로 R에서 제공하는 기본적인 함수들을 이용
- 보다 미적인 그래프를 작성하려면 ggplot 패키지를 주로 이용
- ggplot은 R의 강점 중의 하나가 ggplot이라고 할 만큼 데이터 시각화에서 널리 사용
- ggplot은 복잡하고 화려한 그래프를 작성할 수 있다는 장점이 있지만, 그만큼 배우기 어렵다는 것이 단점
- ggplot2 패키지의 설치 필요

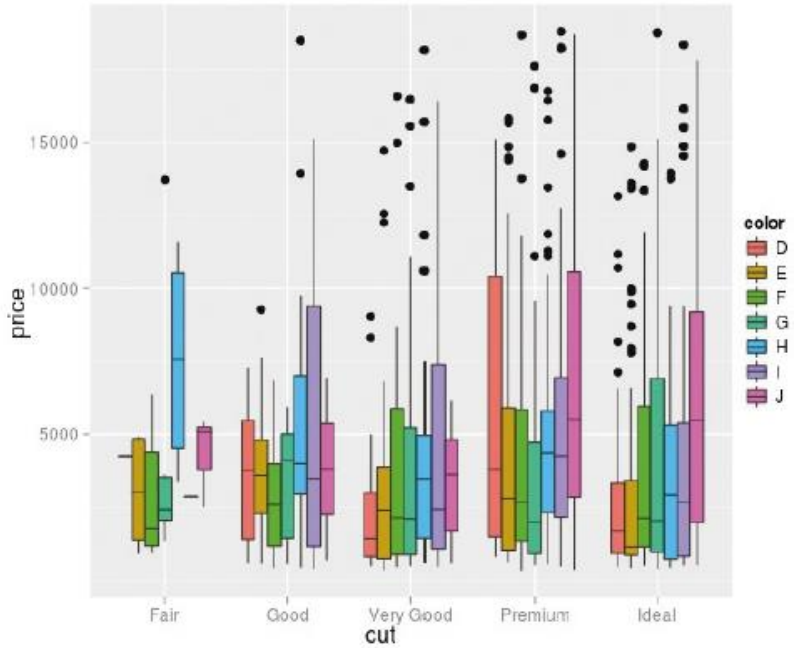


그림 8-5 ggplot의 사례 © https://jeongil.tistory.com/353

1. ggplot 명령문의 기본 구조

- 하나의 ggplot() 함수와 여러 개의 geom_xx() 함수들이 +로 연결되어 하나의 그래프를 완성
- ggplot() 함수의 매개변수로 그래프를 작성할 때 사용할 데이터셋 (data=xx)와 데이터셋 안에서 x축, y축으로 사용할 열 이름(aes(x=x1,y=x2))을 지정
- 이 데이터를 이용하여 어떤 형태의 그래프를 그릴지를 geom_xx()를 통해 지정

ex) geom_bar()

```
ggplot(data=xx, aes(x=x1,y=x2)) +
 geom_xx() +
 geom_yy() +
 ..
```

2. 막대그래프의 작성

2.1 기본적인 막대그래프 작성하기

코드 8-5

```
library(ggplot2)
month <- c(1,2,3,4,5,6)
rain <- c(55,50,45,50,60,70)
df <- data.frame(month,rain)              # 그래프를 작성할 대상 데이터
Df

ggplot(df, aes(x=month,y=rain)) +         # 그래프를 그릴 데이터 지정
 geom_bar(stat="identity",     # 막대의 높이는 y축에 해당하는 열의 값
 width=0.7,                               # 막대의 폭 지정
 fill="steelblue")                        # 막대의 색 지정
```

2. ggplot 패키지

```
> library(ggplot2)
>
> month <- c(1,2,3,4,5,6)
> rain  <- c(55,50,45,50,60,70)
> df <- data.frame(month,rain)
> df
  month rain
1     1   55
2     2   50
3     3   45
4     4   50
5     5   60
6     6   70

> ggplot(df, aes(x=month,y=rain)) +       # 그래프를 그릴 데이터 지정
+    geom_bar(stat="identity",            # 막대의 높이는 y축에 해당하는 열의 값
+             width=0.7,                  # 막대의 폭 지정
+             fill="steelblue")           # 막대의 색 지정
```

2. ggplot 패키지

- `df`

  그래프를 작성할 데이터가 저장되어 있는 데이터프레임을 지정한다. 매트릭스는 데이터프레임으로 변환하여 입력해야 한다.

- `aes(x=month,y=rain)`

  aes( )는 그래프를 그리기 위한 x축, y축의 열을 지정한다.

  · x=month: x축을 구성하는 열이 month임을 지정
  · y=rain: y축을 구성하는 열이 rain임을 지정

- `stat="identity"`

  막대의 높이는 ggplot( ) 함수에서 y축에 해당하는 열(여기서는 rain)에 의해서 결정되도록 지정한다.

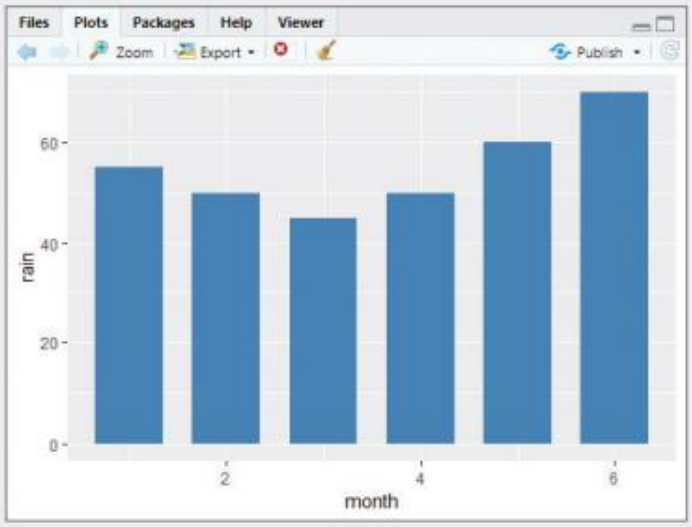
- `width=0.7`

  막대의 폭을 지정한다.

- `fill="steelblue"`

  막대의 내부 색을 지정한다.

2. ggplot 패키지

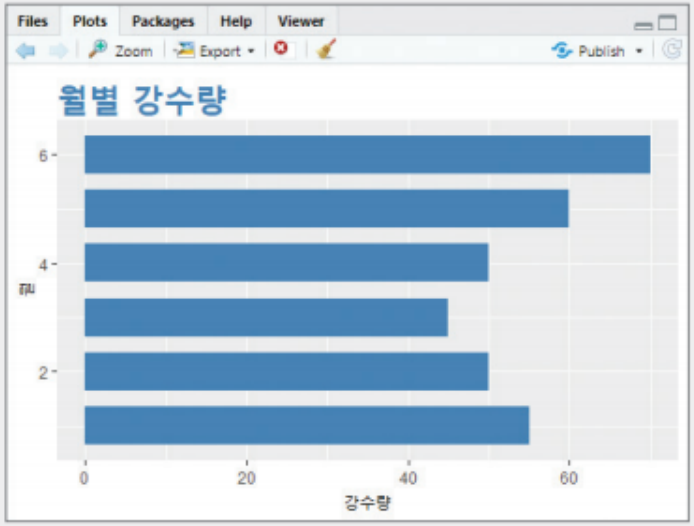

Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
60
40
20
0
rain
2
4
6
month

2.1 막대그래프 꾸미기

코드 8-6

```
ggplot(df, aes(x=month,y=rain)) +          # 그래프를 그릴 데이터 지정
 geom_bar(stat="identity",                 # 막대 높이는 y축에 해당하는 열의 값
 width=0.7,                                # 막대의 폭 지정
 fill="steelblue") +                       # 막대의 색 지정
 ggtitle("월별 강수량") +                   # 그래프의 제목 지정
 theme(plot.title = element_text(size=25, face="bold", colour="steelblue")) +
 labs(x="월",y="강수량") +                  # 그래프의 x, y축 레이블 지정
 coord_flip( )                             # 그래프를 가로 방향으로 출력
```

2. ggplot 패키지



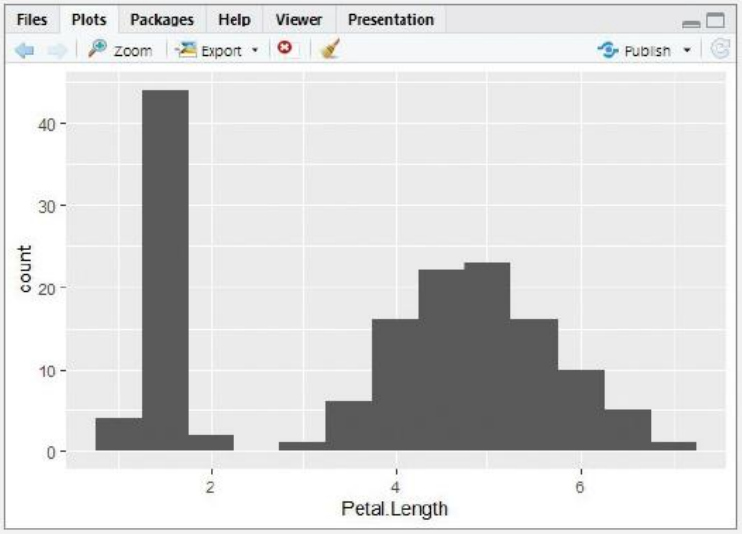
- `ggtitle("월별 강수량")`
  그래프의 제목을 지정하는 함수이다.
- `theme(plot.title = element_text(size = 25, face = "bold", colour="steelblue"))`
  지정된 그래프에 대한 제목의 폰트 크기, 색 등을 지정한다. 이 경우 폰트 크기는 25, 볼드 처리, 폰트 컬러는 강청색으로 지정했다.
- `labs(x="월",y="강수량")`
  그래프의 x축 레이블과 y축 레이블을 지정한다.
- `coord_flip()`
  막대를 가로로 표시하도록 한다.

3. 히스토그램의 작성

3.1 기본적인 히스토그램 작성하기

코드 8-7

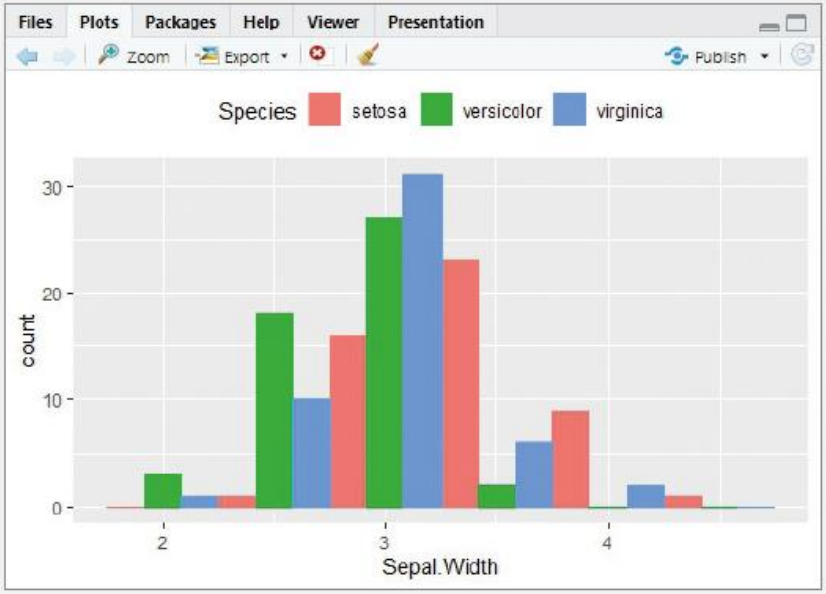
```
library(ggplot2)

ggplot(iris, aes(x=Petal.Length)) +          # 그래프를 그릴 데이터 지정
 geom_histogram(binwidth=0.5)                # 히스토그램 작성
```



3.2 그룹별 히스토그램 작성하기

코드 8-8

```
library(ggplot2)

ggplot(iris, aes(x=Sepal.Width, fill=Species, color=Species)) +
        geom_histogram(binwidth = 0.5, position="dodge") +
        theme(legend.position="top")
```



2. ggplot 패키지

- `x=Sepal.Width`

  히스토그램을 작성할 대상 열을 지정한다.

- `fill=Species`

  히스토그램의 막대 내부를 채울 색을 지정한다. 여기서는 Species(품종)를 지정했는데, Species(품종)는 팩터 타입이기 때문에 숫자 1, 2, 3으로 변환될 수 있다. 품종별로 막대의 색이 다르게 채워진다.

- `color=Species`

  히스토그램의 막대 윤곽선의 색을 지정한다.

- `binwidth = 0.5`

  데이터 구간을 0.5 간격으로 나누어 히스토그램을 작성한다.

- `position="dodge"`

  이 히스토그램은 3개 품종의 히스토그램이 하나의 그래프에 작성된다. 동일 구간에 대해 3개의 막대가 그려진다. position은 동일 구간의 막대들을 어떻게 그릴지를 지정하는데, "dodge"는 막대들을 겹치지 않고 병렬로 그리도록 지정하는 것이다.

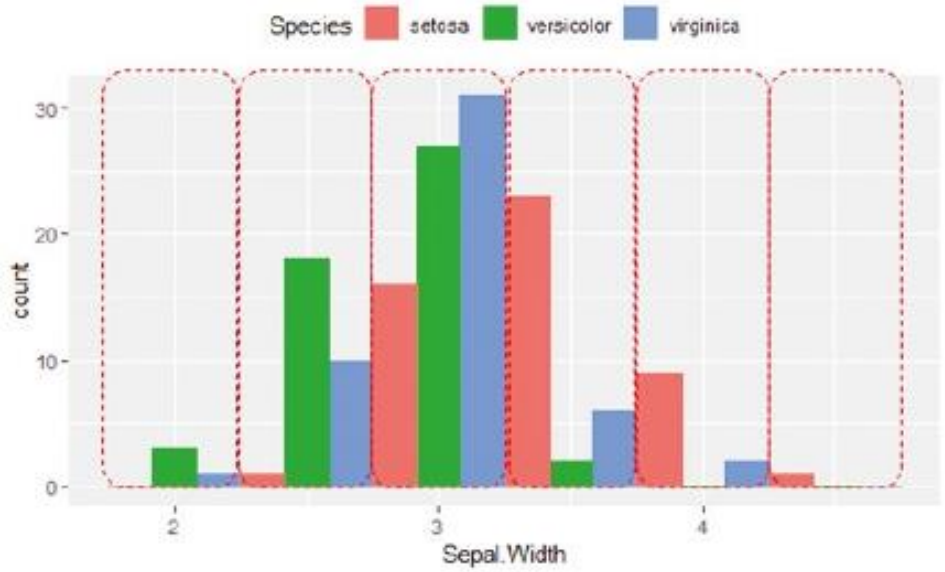


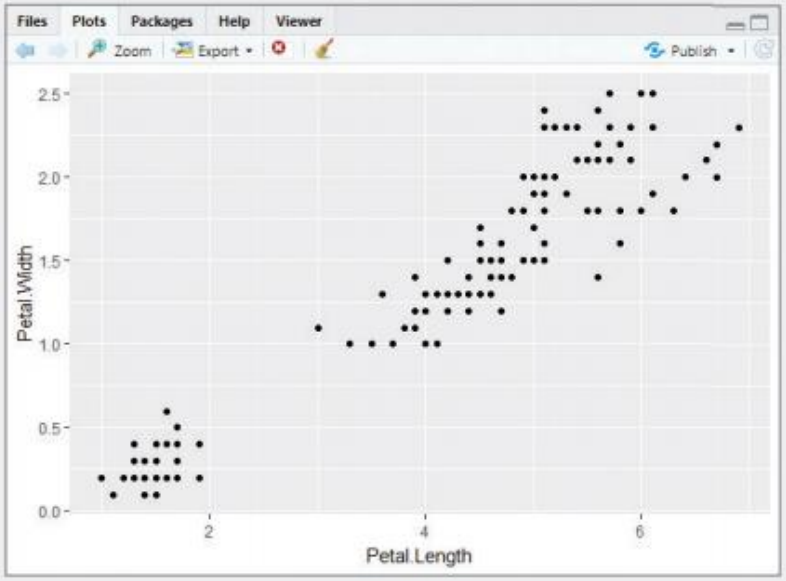
그림 8-6 꽃받침의 폭에 대한 품종별 히스토그램

4. 산점도의 작성

4.1 기본적인 산점도 작성하기

코드 8-9

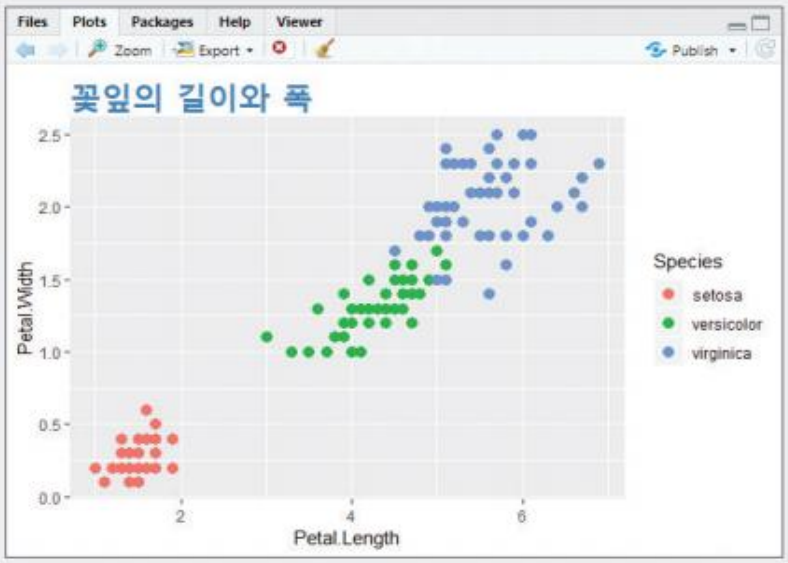
```
library(ggplot2)
ggplot(data=iris, aes(x=Petal.Length, y=Petal.Width)) +
 geom_point( )
```



4.2 그룹이 구분되는 산점도 작성하기

코드 8-10

```
library(ggplot2)
ggplot(data=iris, aes(x=Petal.Length, y=Petal.Width,
                      color=Species)) +
 geom_point(size=3) +
 ggtitle("꽃잎의 길이와 폭") +                    # 그래프의 제목 지정
 theme(plot.title = element_text(size=25, face="bold", colour="steelblue"))
```



5. 상자그림의 작성

5.1 기본적인 상자그림 작성하기

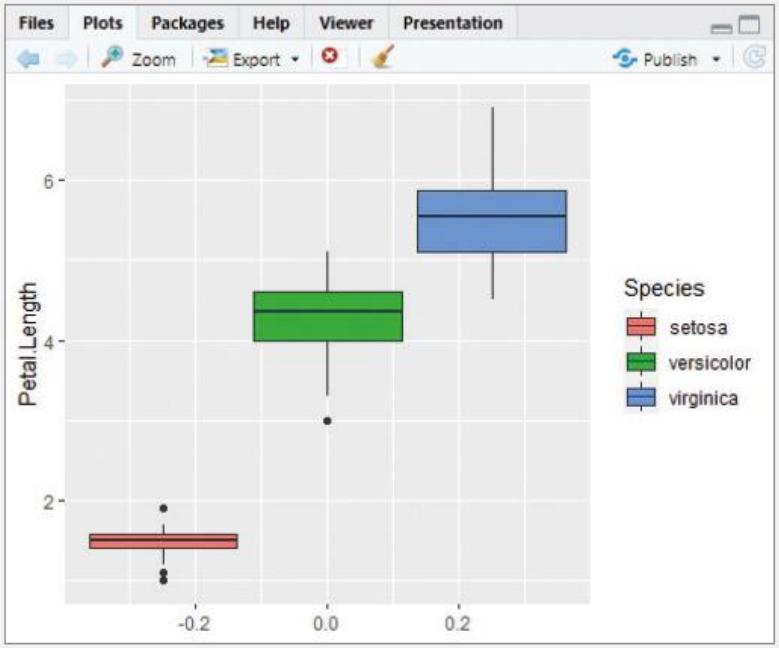
코드 8-11

```
library(ggplot2)

ggplot(data=iris, aes(y=Petal.Length)) +
 geom_boxplot(fill="yellow")
```

5.2 그룹별 상자그림 작성하기

코드 8-12

```
library(ggplot2)

ggplot(data=iris, aes(y=Petal.Length, fill=Species)) +
 geom_boxplot( )
```



6. 선그래프의 작성

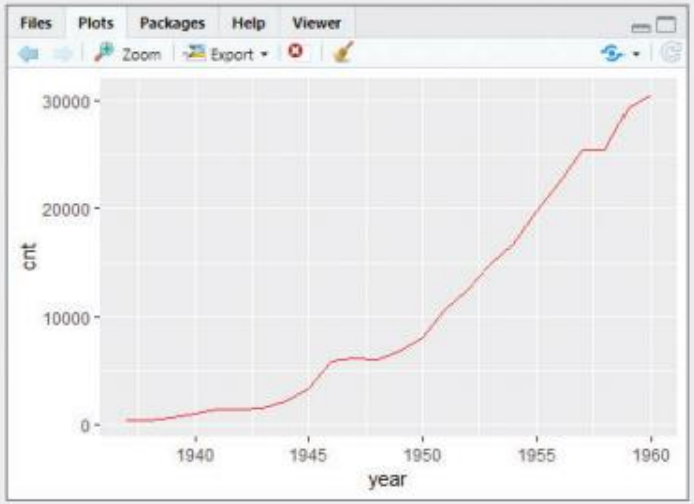
코드 8-13

```
library(ggplot2)

year <- 1937:1960
cnt <- as.vector(airmiles)
df <- data.frame(year,cnt)                          # 데이터 준비
head(df)

ggplot(data=df, aes(x=year,y=cnt)) +                # 선그래프 작성
 geom_line(col="red")
```

```
> year <- 1937:1960
> cnt <- as.vector(airmiles)
> df <- data.frame(year,cnt)                  # 데이터 준비
```

2. ggplot 패키지

```
> head(df)
  year   cnt
1 1937  412
2 1938  480
3 1939  683
4 1940 1052
5 1941 1385
6 1942 1418
> ggplot(data=df, aes(x=year,y=cnt)) +      # 선그래프 작성
+    geom_line(col="red")
>
```



Section 03

차원 축소

1. 차원 축소의 개념

- 산점도는 2차원 평면상에 두 변수의 값으로 좌표로 정하여 위치를 나타내는 방법으로 데이터의 분포를 관찰할 수 있는 시각화 도구
- 변수가 4개인 4차원 데이터에 대한 산점도는 어떻게 그릴 수 있을까?
  → 4차원을 2차원으로 축소하여 그림
- 차원 축소(dimension reduction)란 고차원 데이터를 2,3 차원 데이터로 축소하는 기법을 말하는데, 2,3 차원으로 축소된 데이터로 산점도를 작성하여 데이터 분포를 확인하면 고차원상의 데이터 분포를 추정 가능
- 어떻게 차원을 축소 하는가? → 3차원상의 물체에 빛을 비추면 2차원 평면에 물체의 그림자가 생기는 것과 비슷한 방법(3차원이 2차원으로 축소됨)
- 데이터의 차원을 축소하면 원래 가지고 있던 정보의 손실이 일어남

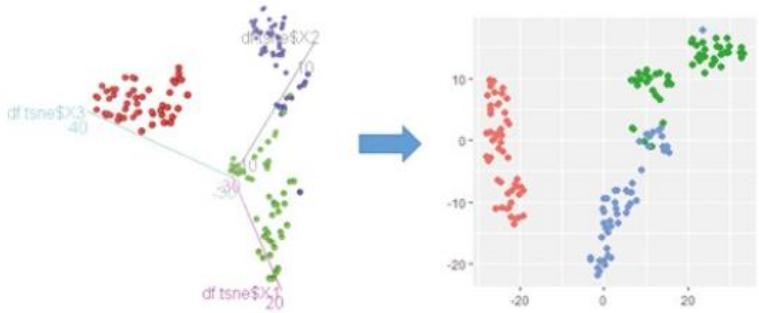


그림 8-7 3차원상의 데이터 분포를 2차원상의 분포로 변환하는 사례

2. R을 이용한 차원 축소

2.1 4차원 데이터를 2차원 산점도로 작성하기

코드 8-14

```
library(Rtsne)
library(ggplot2)
ds <- iris[,-5]                    # 품종 정보 제외

## 중복 데이터 제거
dup = which(duplicated(ds))
dup                                # 143번째 행 중복
ds <- ds[-dup,]
ds.y <- iris$Species[-dup]         # 중복을 제외한 품종 정보

## t-SNE 실행
tsne <- Rtsne(ds,dims=2, perplexity=10)

## 축소결과 시각화
df.tsne <- data.frame(tsne$Y)
head(df.tsne)
ggplot(df.tsne, aes(x=X1, y=X2, color=ds.y)) +
 geom_point(size=2)
```

3. 차원 축소

```
> library(Rtsne)
> library(ggplot2)
>
> ds <- iris[,-5]                          # 품종 정보 제외
> ## 중복 데이터 제거
> dup = which(duplicated(ds))
> dup                                      # 143번째 행 중복
[1] 143

> ds <- ds[-dup,]
> ds.y <- iris$Species[-dup]               # 중복을 제외한 품종 정보
> ## t-SNE 실행
> tsne <- Rtsne(ds,dims=2, perplexity=10)
```

- t-sne를 이용하려면 중복된 데이터가 존재하면 안됨
- 이것을 검사하는 명령문이 which(duplicated(ds))인데, 만일 중복이 있으면 중복된 행의 번호를 dup에 보관
- dup의 값을 보면 143번째 행이 중복되었다고 나오는데 실제로 143번째 행은 102번째 행과 동일

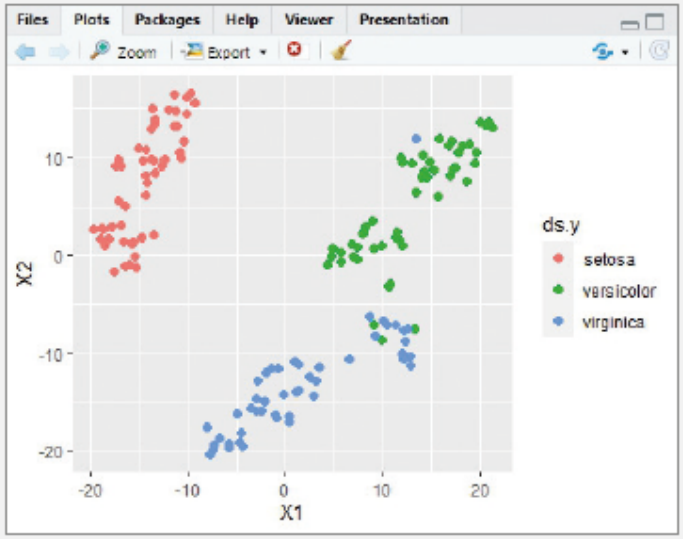
3. 차원 축소

- **`ds`**
  차원 축소 대상 데이터셋이다.
- **`dims=2`**
  ds를 몇 차원으로 축소할지 지정하는데, 2 또는 3이 일반적이다.
- **`perplexity=10`**
  차원 축소 과정에서 데이터를 샘플링하는데, 샘플의 개수를 몇 개로 할지 지정한다. (대상 데이터의 행의 수)/3 보다 작게 지정한다.

```
> ##  축소결과 시각화
> df.tsne <- data.frame(tsne$Y)
> head(df.tsne)
         X1        X2
1 -21.21195  6.847538
2 -14.96598  1.196396
3 -17.08658 -1.832247
4 -16.29159 -1.130622
5 -20.17702  7.507484
6 -26.10980 12.302749
```

3. 차원 축소

```
> ggplot(df.tsne, aes(x=X1, y=X2, color=ds.y)) +
+     geom_point(size=2)
>
```



2.2 4차원 데이터를 3차원 산점도로 작성하기

코드 8-15

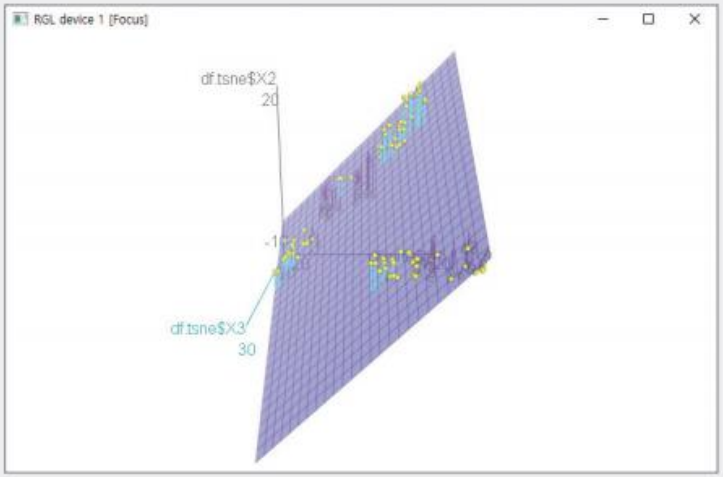
```
install.packages(c("rgl", "car"))
library("car")
library("rgl")
library("mgcv")

tsne <- Rtsne(ds,dims=3, perplexity=10)
df.tsne <- data.frame(tsne$Y)
head(df.tsne)

# 회귀면이 포함된 3차원 산점도
scatter3d(x=df.tsne$X1, y=df.tsne$X2, z=df.tsne$X3)

# 회귀면이 없는 3차원 산점도
points <- as.integer(ds.y)
color <- c('red','green','blue')
scatter3d(x=df.tsne$X1, y=df.tsne$X2, z=df.tsne$X3,
          point.col = color[points],      # 점의 색을 품종별로 다르게
          surface=FALSE)                  # 회귀면을 표시하지 않음
```

3. 차원 축소

```
> install.packages(c("rgl", "car"))
> library("car")
> library("rgl")
> library("mgcv")

tsne <- Rtsne(ds,dims=3, perplexity=10)
> df.tsne <- data.frame(tsne$Y)
> head(df.tsne)
         X1        X2       X3
1  7.028892 -3.443516 31.23626
2 12.551731  2.492051 23.93481
3 14.325457 -1.469314 23.83501
4 13.171411 -1.026139 22.36516
5  7.846652 -4.932413 31.66133
6  0.824220 -4.808023 35.76660
```

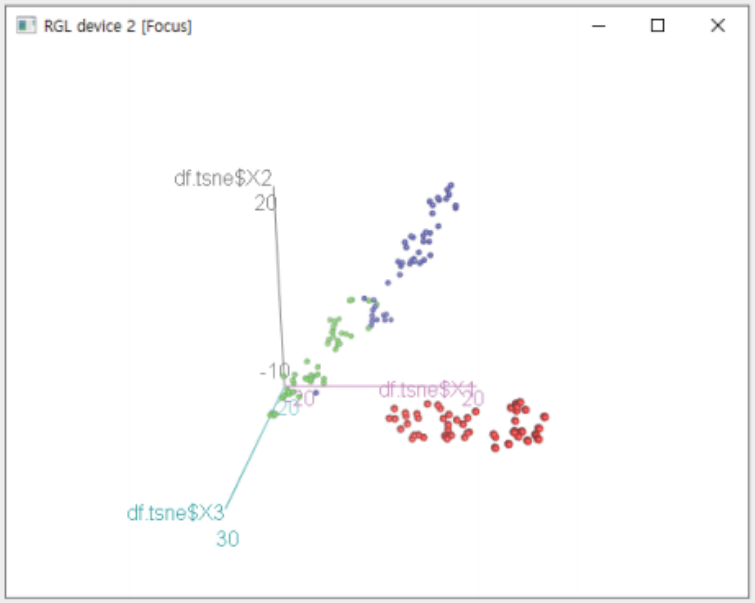
3. 차원 축소

```
> # 회귀면이 포함된 3차원 산점도
> scatter3d(x=df.tsne$X1, y=df.tsne$X2, z=df.tsne$X3)
>
```



3. 차원 축소

```
> # 회귀면이 없는 3차원 산점도
> points <- as.integer(ds.y)
> color <- c('red','green','blue')
> scatter3d(x=df.tsne$X1, y=df.tsne$X2, z=df.tsne$X3,
+           point.col = color[points],        # 점의 색을 품종별로 다르게
+           surface=FALSE)                    # 회귀면을 표시하지 않음
```



Thank you!

한빛아카데미 Hanbit Academy, Inc.