제8강
데이터 시각화

Section 01

데이터 시각화 기법

Section 02

ggplot 패키지

2. ggplot 패키지

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

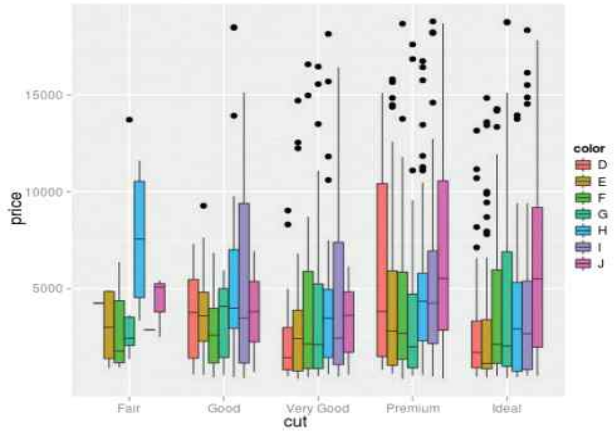


그림 8-5 ggplot의 사례

2. ggplot 패키지

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() -막대그래프

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

2. ggplot 패키지

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.

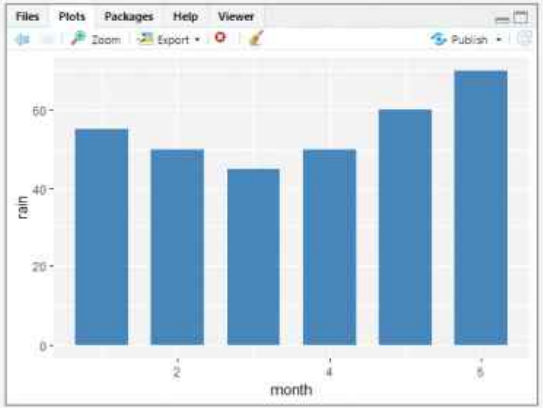
- `width=0.7`

  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
5
month

2. ggplot 패키지

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.

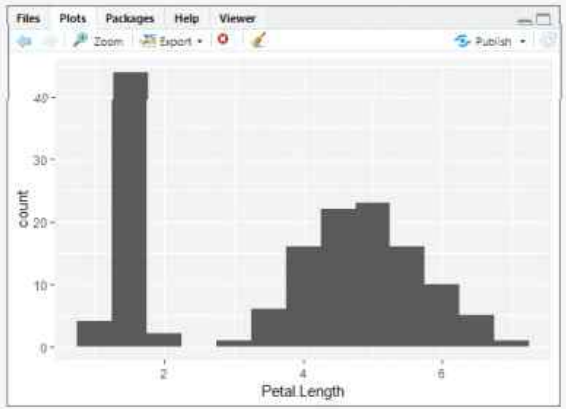
2. ggplot 패키지

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
```



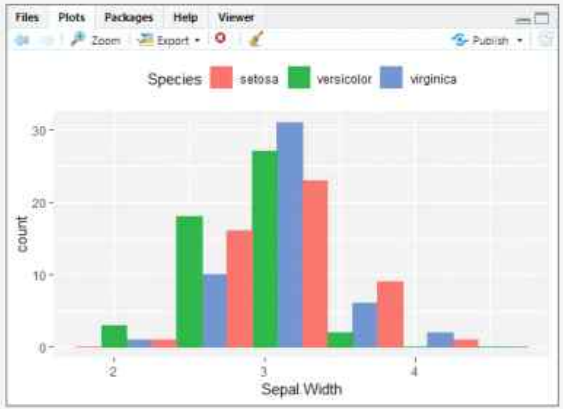
앞서, 히스토그램에 대해서 공부를 했다.
히스토그램은 연속형 숫자자료에 대해 일정 길이로 구간을 나눈 뒤 각 구간에 속하는 자료값이 몇 개 있는지 카운트하는 차트이다.
여기서, 매개변수로 온 binwidth는 막대의 데이터 구간의 간격을 0.5로 하라는 것이다.

2. ggplot 패키지

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")
```



색상은 알다시피 팩터이기에 1,2,3으로 변경이 가능하다.하여, 분홍색,녹색,파랑색 막대가 함께 표시되어진다.
아울러 dodge는 막대들이 겹치지 않고 위치를 잡도록 하는 옵션이며, theme(legend.position= "top")는 범례를 상위로 잡는 옵션이다.

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.

2. ggplot 패키지

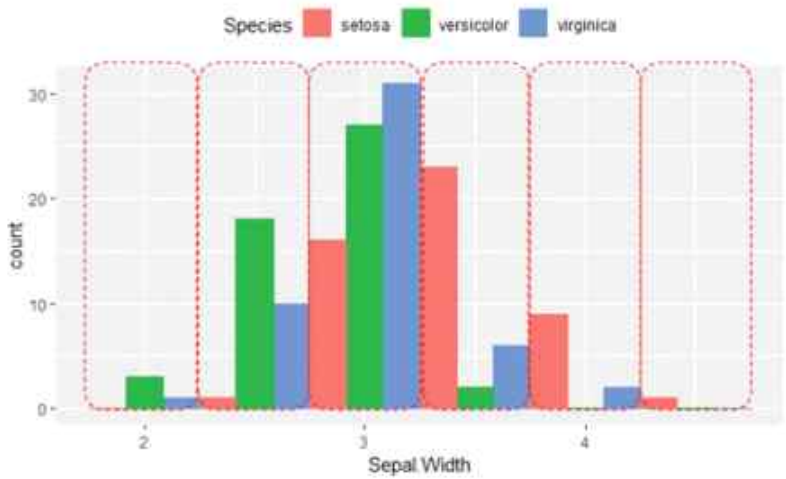


그림 8-6 꽃받침의 폭에 대한 품종별 히스토그램

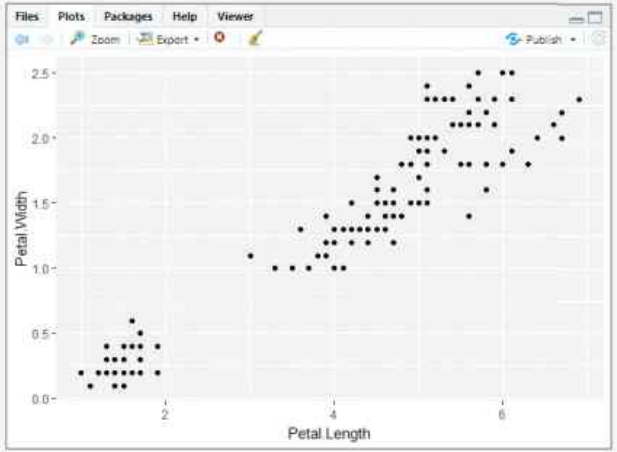
2. ggplot 패키지

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( )
```



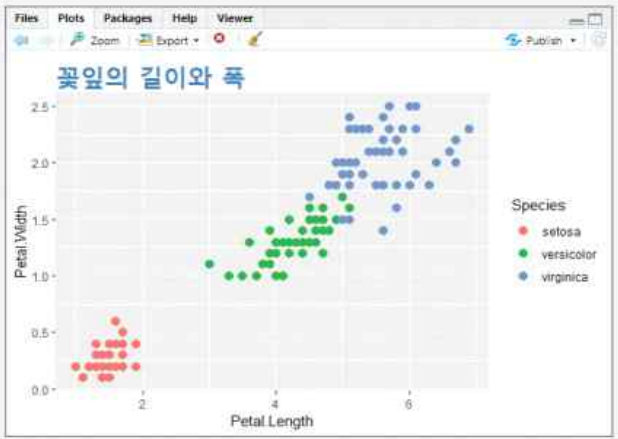
앞서, 배웠듯이 산점도는 2개 혹은 그 이상의 변수들의 분포도를 의미한다.
geom_point()는 산점도를 그려라는 것이다.

2. ggplot 패키지

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"))
```



산점도를 그룹별로 나타내는데 역시 color는 Species를 주어 3가지 색깔을 가지게 하였다. 아울러, geom_point()의 매개변수 size=3은 점의 크기를 3으로 하라는 매개변수이다.
맨 아래 코드는 제목에 대한 설정이다.
또한, 품종별로 점의 모양도 품종별로 다르게 하고자 한다면, shape=Species를 aes()안에 추가를 하면된다.

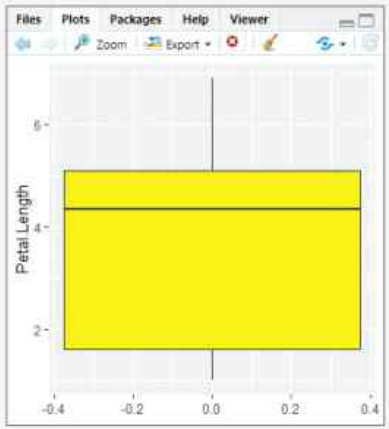
2. ggplot 패키지

5. 상자그림의 작성

5.1 기본적인 상자그림 작성하기

코드 8-11

```
library(ggplot2)

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



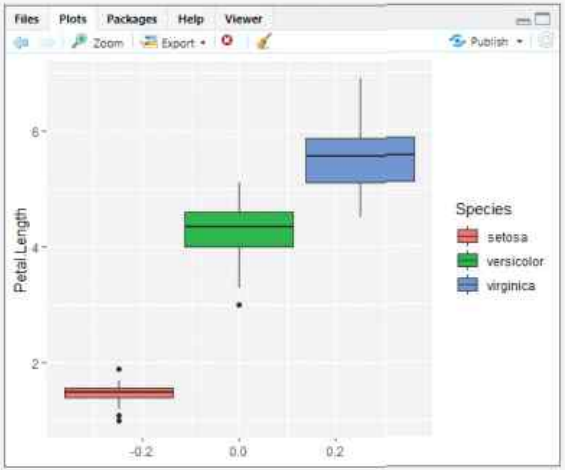
앞서, 배웠듯이 박스플랏은 사분위수를
나타내는데 사용된다.
geom_boxplot()는 박스플랏을 그려라는 것이다.

2. ggplot 패키지

5.2 그룹별 상자그림 작성하기

코드 8-12

```
library(ggplot2)

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



그룹별로 나타내고 있다.

2. ggplot 패키지

6. 선그래프의 작성

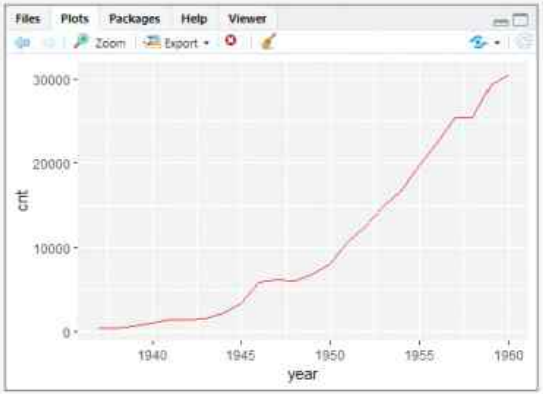
코드 8-13

```
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")
>
```



이와 같이 ggplot패키지는 자신이 원하는 차트를 이쁘게도 하고 여러가지 옵션을 지원하기에 그만큼 익히는데 시간이 걸린다 하지만, 그리고자 하는 차트가 있다면 인터넷 서칭을 해보면 자료가 방대하다.
하여, 최대한 비슷한 것을 가져와서 소스를 수정하여 사용하는 것이 가장 빠르다.

감사합니다.