

R 교재 ch05. 데이터 시각화

1. 시각화 도구 분류

데이터 분석의 도입부에서 전체적인 데이터 구조를 살펴보기 위해서 시각화 도구 사용

이산변수인 경우: 막대, 점, 원형 차트

연속변수: 상자 박스, 히스토그램, 산점도

[표 5.1] 칼럼 특성의 시각화 도구 분류

hist(히스토그램), plot(산점도), barplot(막대 차트), pie(원형 차트), abline(선 추가),
boxplot(상자 박스), scatterplot3d(3차원 산점도), pair(산점도 매트릭스)

2. 이산변수 시각화

이산변수(discrete quantitative data): 정수 단위로 나누어 측정할 수 있는 변수
막대차트, 점 차트, 원 차트 이용

2.1 막대 차트 시각화

barplot()함수를 이용하여 세로 막대 차트와 가로 막대 차트 그리기

(1) 세로 막대 차트

barplot()함수는 세로 막대 차트 제공

실습 (세로 막대 차트 그리기)

barplot()함수

where

ylim: y축 값의 범위

col: 각 막대를 나타낼 색상 지정

main: 차트의 제목

1단계: 차트 작성을 위한 자료 만들기

```
chart_data <- c(305, 450, 320, 460, 330, 480, 380, 520)
```

```
names(chart_data) <- c("2018 1분기", "2019 1분기",  
                      "2018 2분기", "2019 2분기",  
                      "2018 3분기", "2019 3분기",  
                      "2018 4분기", "2019 4분기")
```

```
str(chart_data)
```

```
chart_data
```

2단계: 세로 막대 차트 그리기

```
barplot(chart_data, ylim = c(0, 600),  
        col = rainbow(8),  
        main = "2018년도 vs 2019년도 매출현황 비교")
```

barplot()

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/barplot>

실습 (barplot()함수 도움말 보기)

help("barplot")

실습 (막대 차트의 가로축과 세로축에 레이블 추가하기)

```
barplot(chart_data, ylim = c(0, 600),  
        ylab = "매출액(단위: 만원)",  
        xlab = "년도별 분기 현황",  
        col = rainbow(8),  
        main = "2018년도 vs 2019년도 매출현황 비교")
```

막대 차트에 축 이름 추가: xlab속성, ylab속성

(2) 가로 막대 차트

barplot()함수에 horiz속성을 TRUE로 지정

실습 (가로 막대 그리기)

```
barplot(chart_data, xlim = c(0, 600), horiz = T,  
        ylab = "매출액(단위: 만원)",  
        xlab = "년도별 분기 현황",
```

```
col = rainbow(8),
main = "2018년도 vs 2019년도 매출현황 비교")
```

가로 막대 차트로 변형되지만 xlim속성 사용

실습 (막대 차트에서 막대 사이의 간격 조정하기)

```
barplot(chart_data, xlim = c(0, 600), horiz = T,
        ylab = "매출액(단위: 만원)",
        xlab = "년도별 분기 현황",
        col = rainbow(8), space = 1, cex.names = 0.8,
        main = "2018년도 vs 2019년도 매출현황 비교")
```

space속성: 막대의 굵기와 간격 지정

space속성값이 클수록 막대의 굵기는 작아지고, 막대와 막대 사이의 간격은 넓어진다.

cex.names속성: 축 이름의 크기 지정

실습 (막대 차트에서 막대의 색상 지정)

```
barplot(chart_data, xlim = c(0, 600), horiz = T,
        ylab = "매출액(단위: 만원)",
        xlab = "년도별 분기 현황",
        space = 1, cex.names = 0.8,
        main = "2018년도 vs 2019년도 매출현황 비교",
        col = rep(c(2, 4), 4))
```

col속성: 색상 설정

where

1	2	3	4	5	6	7
black	red	green	blue	skyblue	purple	yellow

"col=rep(c(2,4), 4)"에서 2번과 4번 색상 사용, 4번 반복

실습: 막대 차트에서 색상 이름을 사용하여 막대의 색상 지정하기

```
barplot(chart_data, xlim = c(0, 600), horiz = T,  
        ylab = "매출액(단위: 만원)",  
        xlab = "년도별 분기 현황",  
        space = 1, cex.names = 0.8,  
        main = "2018년도 vs 2019년도 매출현황 비교",  
        col = rep(c("red", "green"), 4))
```

색상값이 아닌 색상의 이름을 사용

(3) 누적 막대 차트

하나의 컬럼에 여러 개의 자료를 가지고 있는 경우 자료를 개별적인 막대로 표현 또는 누적형태로 표현

실습 (누적 막대 차트 그리기)

1단계: 메모리에 데이터 가져오기

```
data("VADeaths")  
VADeaths
```

2단계: VADeaths데이터 셋 구조 보기

```
str(VADeaths)  
class(VADeaths)  
mode(VADeaths)
```

VADeaths데이터 셋 설명

<https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/VADeaths>

barplot()함수

Where

beside = T/F: X축 값을 측면으로 배열, F인 경우 하나의 막대에 누적

font.main: 제목 글꼴 지정

legend(): 범례의 위치, 이름, 글자 크기, 색상 지정

title(): 아트 제목, 차트 글꼴 지정

* RStudio에서 차트를 그릴 때는 차트가 그려지는 Plots영역을 최대한 확대한 후 스크립트를 실행해야 범례가 깨지지 않고 표시

3단계 : 개별 차트와 누적 차트 그리기

```
par(mfrow = c(1, 2))
```

```
barplot(VADeaths, beside = T, col = rainbow(5),
```

```
      main = "미국 버지니아주 하위계층 사망비율")
```

```
legend(19, 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))
```

par()함수: RStudio의 차트가 나타나는 영역에서 두 개 이상의 차트를 동시에 볼 수 있게 함.

'beside= T' 속성: 하나의 막대에 누적

차트에 제목 넣기: main 속성을 이용 또는 title()함수 이용

font.main속성: 차트 제목의 글꼴 유형 지정

2.2 점 차트 시각화

실습 (점 차트(dotchart) 도움말 보기)

help(dotchart)

dotchart()함수

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/dotchart>

where

x: 데이터

label: 점에 대한 설명문

cex: 점의 확대

pch: 점 모양

color: 점의 색

lcolor: 선의 색

main: 차트 제목

xlab: x축의 이름

실습 (점 차트 사용하기)

```
par(mfrow = c(1, 1))
```

```
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)
```

dotchart()함수의 주요 속성:

col: 레이블과 점 색상 지정

lclor: 구분선(line) 색상 지정

pch(plotting character): 점 모양

labels: 점에 대한 레이블 표시

xlabs: x축 이름

cex(character expansion): 레이블과 점의 크기 확대

2.3 원형 차트 시각화

실습 (원형 차트(pie)도움말 보기)

help(pie)

pie()함수

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/pie>

where

x: 데이터

labels: 원형 차트에서 각 조각에 대한 설명문

col: 색상

border: 테두리 색

lty: 선 타입

main: 차트 제목을 지정

실습 (분기별 매출현황을 파이 차트로 시각화하기)

```
par(mfrow = c(1, 1))
```

```
pie(chart_data, labels = names(chart_data), col = rainbow(8), cex = 1.2)
```

```
title("2018~2019년도 분기별 매출현황")
```

'clockwise = TRUE'속성: 시계방향으로 데이터 표시. Default는 FALSE

3. 연속변수 시각화

연속변수(Continuous quantitative data): 시간, 길이 등과 같이 연속성을 가진 변수
상자 그래프, 히스토그램, 산점도

3.1 상자 그래프 시각화

상자 그래프: 요약정보를 시각화하는데 효과적
데이터의 분포 정도와 이상치 발견을 목적으로 하는 경우 사용

실습 (VADeaths 데이터 셋을 상자 그래프로 시각화하기)

1단계: "notch=FALSE"일 때

```
boxplot(VADeaths, range = 0)
```

'range=0' 속성에 의해 칼럼의 최소값과 최대값을 점선으로 연결

2단계: "notch=TRUE"일 때

```
boxplot(VADeaths, range = 0, notch = T)
```

```
abline(h = 37, lty = 3, col = "red")
```

추가된 'notch = T' 속성에 의해 중위수 기준으로 허리선이 추가

abline() 함수에 의해 지정된 y좌표(h 속성)에 빨간색(col 속성) 점선(lty 속성) 적용

실습 (VADeath 데이터 셋의 요약통계량 보기)

```
summary(VADeaths)
```

3.2 히스토그램 시각화

히스토그램(histogram): 측정값의 범위(구간)를 그래프의 x축으로 놓고, 범위에 속하는 측정값의 출현 빈도수를 y축으로 나타낸 그래프 형태

분포곡선: 히스토그램에 도수의 값을 선으로 연결하여 얻어지는 곡선

실습 (iris 데이터 셋 가져오기)

```
data(iris)
names(iris)
str(iris)
head(iris)
```

names() 함수: 컬럼명 보기

iris 데이터 셋 설명

<https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/iris>

실습 (iris 데이터 셋의 꽃받침 길이(Sepal.Length) 컬럼으로 히스토그램 시각화)

```
summary(iris$Sepal.Length)
hist(iris$Sepal.Length, xlab = "iris$Sepal.Length", col = "magenta",
     main = "iris 꽃 받침 길이 Histogram", xlim = c(4.3, 7.9))
```

hist() 함수

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/hist>

where

xlab: x축 이름

co: 차트 생성

main: 차트 제목

xlim: x축 범위

실습 (iris 데이터 셋의 꽃받침 너비(Sepal.Width)컬럼으로 히스토그램 시각화)

```
summary(iris$Sepal.Width)
hist(iris$Sepal.Width, xlab = "iris$Sepal.Width", col = "mistyrose",
     main = "iris 꽃받침 너비 Histogram", xlim = c(2.0, 4.5))
```

실습 (히스토그램에서 빈도와 밀도 표현하기)

1단계: 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))
```

2단계: 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))
```

3단계: 밀도를 기준으로 line 추가하기

```
lines(density(iris$Sepal.Width), col = "red")
```

오른쪽 그래프는 'freq=F'속성에 의해 계급에 대한 밀도(Density)를 y축으로 표현한 결과 density()와 lines()함수에 의해서 밀도 그래프에 분포곡선이 그려짐

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/density>

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/lines>

실습 (정규분포 추정 곡선 나타내기)

정규분포는 평균값을 중앙으로 좌우대칭인 종 모양(Bell-shape)을 이루고 있다.

1단계: 계급을 밀도로 표현한 히스토그램 시각화

```
par(mfrow = c(1, 1))  
hist(iris$Sepal.Width, xlab = "iris$Sepal.Width", col = "mistyrose",  
      freq = F, main = "iris 꽃받침 너비 Histogram", xlim = c(2.0, 4.5))
```

2단계: 히스토그램에 밀도를 기준으로 분포곡선 추가

```
lines(density(iris$Sepal.Width), col = "red")
```

3단계: 히스토그램에 정규분포 추정 곡선 추가

```
x <- seq(2.0, 4.5, 0.1)  
curve(dnorm(x, mean = mean(iris$Sepal.Width),  
           sd = sd(iris$Sepal.Width)),  
      col = "blue", add = T)
```

curve()함수와 dnorm()함수를 이용하여 정규분포의 곡선 추가

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/curve>

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/Normal>

3.3 산점도 시각화

산점도(scatter plot): 두 개 이상의 변수들 사이의 분포를 점으로 표시한 차트를 의미

plot()함수

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/plot>

실습 (산점도 그래프에 대각선과 텍스트 추가하기)

1단계: 기본 산점도 시각화

```
price <- runif(10, min = 1, max = 100)
plot(price, col = "red")
```

2단계: 대각선 추가

```
par(new = T)
line_chart = 1:100
plot(line_chart, type = "l", col = "red", axes = F, ann = F)
```

3단계: 텍스트 추가

```
text(70, 80, "대각선 추가", col = "blue")
```

text()함수

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/text>

실습 (type속성으로 산점도 그리기)

```
par(mfrow = c(2, 2))
plot(price, type = "l")
plot(price, type = "o")
plot(price, type = "h")
```

```
plot(price, type = "s")
```

plot()함수 내 type속성을 이용하여 점을 선으로 연결

실습 (pch속성으로 산점도 그리기)

plot()함수 내 pch(plotting character)속성을 이용하여 30가지의 다양한 형태의 연결점 표현 가능

col(color)속성으로 연결점과 선의 색상과 굵기 지정 가능

1단계: pch속성과 col, cex속성 사용

```
par(mfrow = c(2, 2))
plot(price, type = "o", pch = 5)
plot(price, type = "o", pch = 15)
plot(price, type = "o", pch = 20, col = "blue")
plot(price, type = "o", pch = 20, col = "orange", cex = 1.5)
plot(price, type = "o", pch = 20, col = "green", cex = 2.0, lwd = 3)
```

2단계: lwd속성 추가 사용

```
par(mfrow=c(1,1))
plot(price, type="o", pch=20,
      col = "green", cex=2.0, lwd=3)
```

pch속성으로 점의 모양 지정
col속성으로 색상을 지정
cex속성으로 점의 모양을 확대
lwd속성으로 선의 굵기를 지정

plot()함수의 시각화 도구 목록

```
methods("plot")
```

method()함수에 "plot"을 넣어 plot()함수에서 제공하는 시각화 기능 확인 가능
기능 확인해 볼 것!

```
> methods("plot")
```

[1] plot.acf*	plot.data.frame*	plot.decomposed.ts*
[4] plot.default	plot.dendrogram*	plot.density*
[7] plot.ecdf	plot.factor*	plot.formula*
[10] plot.function	plot.hclust*	plot.histogram*
[13] plot.HoltWinters*	plot.isoreg*	plot.lm*
[16] plot.medpolish*	plot.mlm*	plot.ppr*
[19] plot.prcomp*	plot.princomp*	plot.profile.nls*
[22] plot.raster*	plot.spec*	plot.stepfun
[25] plot.stl*	plot.table*	plot.ts
[28] plot.tskernel*	plot.TukeyHSD*	

```
> methods("plot")
```

plot.acf*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/acf>

<https://rdr.io/r/stats/plot.acf.html>

plot.data.frame*

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/plot.data.frame>

<https://www.stat.berkeley.edu/~s133/R-4a.html>

plot.decomposed.ts*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/decompose>

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/ts.plot>

<https://rpubs.com/davoodastarky/TSA1>

plot.default

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/plot.default>

plot.dendrogram*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/dendrogram>

<https://www.gastonsanchez.com/visually-enforced/how-to/2012/10/03/Dendrograms/>

plot.density*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/plot.density>

<https://www.statmethods.net/graphs/density.html>

plot.ecdf

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/ecdf>

<http://www.sthda.com/english/wiki/ggplot2-ecdf-plot-quick-start-guide-for-empirical-cumulative-density-function-r-software-and-data-visualization>

plot.factor*

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/plot.factor>

<https://rdr.io/r/graphics/plot.factor.html>

plot.formula*

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/plot.formula>

<https://cran.r-project.org/web/packages/ggformula/vignettes/ggformula.html>

plot.function

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/curve>

<https://www.journaldev.com/36083/plot-function-in-r>

plot.hclust*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/hclust>

<https://r-charts.com/part-whole/hclust/>

plot.histogram*

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/plot.histogram>

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/hist>

<https://www.datamentor.io/r-programming/histogram/>

plot.HoltWinters*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/plot.HoltWinters>

<https://www.r-bloggers.com/2012/07/holt-winters-forecast-using-ggplot2/>

plot.isoreg*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/plot.isoreg>

https://www.imsbio.co.jp/RGM/R_rdf?f=stats/man/plot.isoreg.Rd&d=R_rel

plot.lm*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/plot.lm>

<https://sejohnston.com/2012/08/09/a-quick-and-easy-function-to-plot-lm-results-in-r/>

plot.medpolish*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/medpolish>

<https://rdr.io/r/stats/medpolish.html>

plot.mlm*

<https://deforster.github.io/MLMplotting.html>

<https://maths-people.anu.edu.au/~johnm/r-book/3edn/scripts/mlm.html>

plot.ppr*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/plot.ppr>

<https://rdr.io/r/stats/plot.ppr.html>

plot.prcomp*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/prcomp>

<http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/118-principal-component-analysis-in-r-prcomp-vs-princomp/>

plot.princomp*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/princomp>

<https://www.gastonsanchez.com/visually-enforced/how-to/2012/06/17/PCA-in-R/>

plot.profile.nls*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/plot.profile.nls>

<https://stat.ethz.ch/R-manual/R-devel/library/stats/html/plot.profile.nls.html>

plot.raster*

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/plot.raster>

<https://www.rdocumentation.org/packages/raster/versions/3.4-13/topics/plot>

plot.spec*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/plot.spec>

<https://stat.ethz.ch/R-manual/R-devel/library/stats/html/plot.spec.html>

plot.stepfun

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/plot.stepfun>

<https://stat.ethz.ch/R-manual/R-devel/library/stats/html/plot.stepfun.html>

plot.stl*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/stlmethods>

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/stl>

<https://stat.ethz.ch/R-manual/R-devel/library/stats/html/stl.html>

plot.table*

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/plot.table>

<https://magoosh.com/data-science/how-to-make-an-r-plot-table/>

plot.ts

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/plot.ts>

<https://stat.ethz.ch/R-manual/R-devel/library/stats/html/plot.ts.html>

plot.tskernel*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/kernel>

<https://www.rdocumentation.org/packages/latticeExtra/versions/0.6-29/topics/panel.tskernel>

<https://rdr.io/cran/latticeExtra/man/panel.tskernel.html>

plot.TukeyHSD*

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/TukeyHSD>

<https://rpubs.com/brouwern/plotTukeyHSD2>

plot() 함수에서 시계열 객체 사용하여 추세선 그리기

data("WWWusage") # 시계열 데이터 가져오기

str(WWWusage) # 데이터셋 구조

plot(WWWusage) # plot.ts(WWWusage)와 같다.

3.4 중첩 자료 시각화

2차원 산점도 그래프는 x축과 y축의 교차점에 점(point)을 나타내는 원리로 그려진다. 동일한 좌표값을 갖는 여러 개의 자료가 존재한다면 점이 중첩되어 해당 좌표에는 하나의 점으로만 표시
중첩된 자료를 중첩된 자료의 수 만큼 점의 크기를 확대하여 시각화하는 방법

실습 (중복된 자료의 수 만큼 점의 크기 확대하기)

1단계: 두개의 벡터 객체 준비

```
x <- c(1, 2, 3, 4, 2, 4)
y <- rep( 2, 6)
x; y
```

2단계: 교차테이블 작성

```
table(x, y)
```

3단계: 산점도 시각화

```
plot(x, y)
```

4단계: 교차테이블로 데이터프레임 생성

```
xy.df <- as.data.frame(table(x, y))
xy.df
```

as.data.frame() 함수: 교차테이블 결과를 데이터프레임으로 변환

<https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/as.data.frame>

2차원 그래프에서 같은 좌표에 중복 수를 가중치로 적용 가능

plot()함수 내 주요 속성:

col: 점 색상
pch: 점 모양 지정
cex: 점 크기 확대

5단계: 좌표에 중복된 수 만큼 점을 확대

```
plot(x, y,  
      pch = "@", col = "blue", cex = 0.5 * xy.df$Freq,  
      xlab = "x 벡터의 원소", ylab = "y 벡터 원소")
```

실습 (galton 데이터 셋을 대상으로 중복된 자료 시각화하기)

1단계: galton 데이터 셋 가져오기

```
library(UsingR)  
data(galton)
```

데이터프레임 생성하여 중복 수 컬럼(Freq) 생성

2단계: 교차테이블을 작성하고, 데이터프레임으로 변환

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

3단계: 컬럼 단위 추출

```
names(galtonData) = c("child", "parent", "freq")  
head(galtonData)  
parent <- as.numeric(galtonData$parent)  
child <- as.numeric(galtonData$child)
```

4단계: 점의 크기 확대

```
par(mfrow = c(1, 1))
```

```
plot(parent, child,  
      pch = 21, col = "blue", bg = "green",  
      cex = 0.2 * galtonData$freq,  
      xlab = "parent", ylab = "child")
```

plot()함수 내 cex속성을 이용하여 중복 자료를 시각화

3.5 변수간의 비교 시각화

변수와 변수 사이의 관계를 시각화

실습 (iris 데이터 셋의 4개 변수를 상호 비교)

```
attributes(iris)
pairs(iris[iris$Species == "virginica", 1:4])
pairs(iris[iris$Species == "setosa", 1:4])
```

attributes() 함수: 컬럼명 확인

<https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/attributes>

pairs() 함수: matrix 또는 데이터프레임의 numeric컬럼을 대상으로 변수들 사이의 비교 결과를 행렬구조의 분산된 그래프로 제공

<https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/pairs>

실습 (3차원으로 산점도 시각화)

1단계: 3차원 산점도를 위한 scatterplot3d 패키지 설치 및 로딩

```
install.packages("scatterplot3d")
library(scatterplot3d)
```

scatterplot3d() 함수: 3차원 프레임 생성

형식: scatterplot3d(x, y, z, xlab, ylab, zlab, type)

<https://www.rdocumentation.org/packages/scatterplot3d/versions/0.3-41/topics/scatterplot3d>

2단계: 꽃의 종류별 분류

```
iris_setosa = iris[iris$Species == 'setosa', ]
iris_versicolor = iris[iris$Species == 'versicolor', ]
```

```
iris_virginica = iris[iris$Species == 'virginica', ]
```

3단계: 3차원 틀(Frame)생성하기

```
d3 <- scatterplot3d(iris$Petal.Length,  
                    iris$Sepal.Length,  
                    iris$Sepal.Width,  
                    type = 'n')
```

4단계: 3차원 산점도 시각화

```
d3$points3d(iris_setosa$Petal.Length,  
            iris_setosa$Sepal.Length,  
            iris_setosa$Sepal.Width,  
            bg = 'orange', pch = 21)
```

```
d3$points3d(iris_versicolor$Petal.Length,  
            iris_versicolor$Sepal.Length,  
            iris_versicolor$Sepal.Width,  
            bg = 'blue', pch = 23)
```

```
d3$points3d(iris_virginica$Petal.Length,  
            iris_virginica$Sepal.Length,  
            iris_virginica$Sepal.Width,  
            bg = 'green', pch = 25)
```

cf)

<https://www.rdocumentation.org/packages/rgl/versions/0.105.13/topics/points3d>

ch5 연습문제 풀기