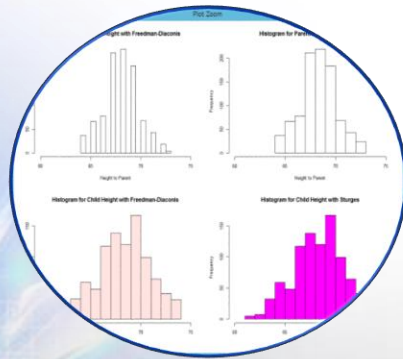


## Part-II. 탐색적 데이터 분석과 전처리



- 5. 데이터 시각화
- 6. 데이터 조작
- 7. EDA & 데이터 전처리
- 8. 고급 시각화 분석
- 9. 정형과 비정형 데이터 처리



# 5. 데이터 시각화

## chap05\_DataVisualization 수업내용

1. 이산변수 시각화
2. 연속변수 시각화
3. 동일데이터가 겹친 경우 시각화 표현
4. 변수 간의 산점도 matrix
5. 차트 결과 파일 저장
6. 칼럼의 속성에 따른 시각화 도구 분류



# 1. 이산변수 시각화

## 1. 이산변수(discrete quantitative data) 시각화

- 정수단위로 나누어 측정할 수 있는 변수

### ● barplot() 형식 - 막대차트 그리기 함수

```
help("barplot") # barplot() 함수 형식 보기
```

```
barplot(height, width = 1, space = NULL,  
        names.arg = NULL, legend.text = NULL, beside = FALSE,  
        horiz = FALSE, density = NULL, angle = 45,  
        col = NULL, border = par("fg"),  
        main = NULL, sub = NULL, xlab = NULL, ylab = NULL,  
        xlim = NULL, ylim = NULL, xpd = TRUE, log = "",  
        axes = TRUE, axisnames = TRUE,  
        cex.axis = par("cex.axis"), cex.names = par("cex.axis"),  
        inside = TRUE, plot = TRUE, axis.lty = 0, offset = 0,  
        add = FALSE, args.legend = NULL, ...)
```



# 1. 이산변수 시각화

- 시각화를 위한 데이터 셋 가져오기

# 막대차트 데이터 생성

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

```
names(chart_data) <- c("2014 1분기","2015 1분기","2014 2분기","2015  
2분기","2014 3분기","2015 3분기","2014 4분기","2015 4분기")
```

```
str(chart_data)
```

```
chart_data
```

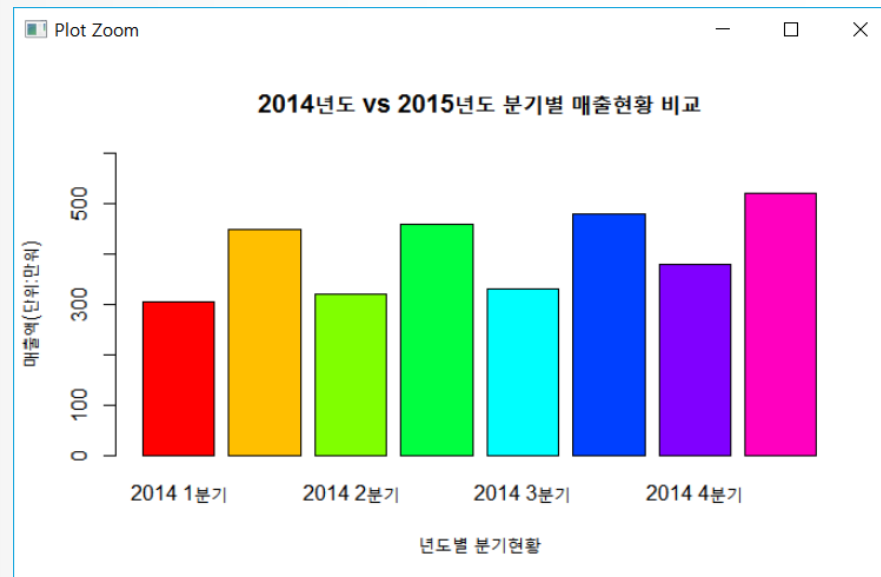


# 1. 이산변수 시각화

## ① 막대차트 시각화

# 세로 막대 차트

```
barplot(chart_data, ylim=c(0,600),  
        col=rainbow(8), main="2014년도 vs 2015년도 분기별 매출현황 비교")  
barplot(chart_data, ylim=c(0,600), ylab="매출액(단위:만원)", xlab="년도별  
분기현황",  
        col=rainbow(8), main="2014년도 vs 2015년도 분기별 매출현황 비교")
```

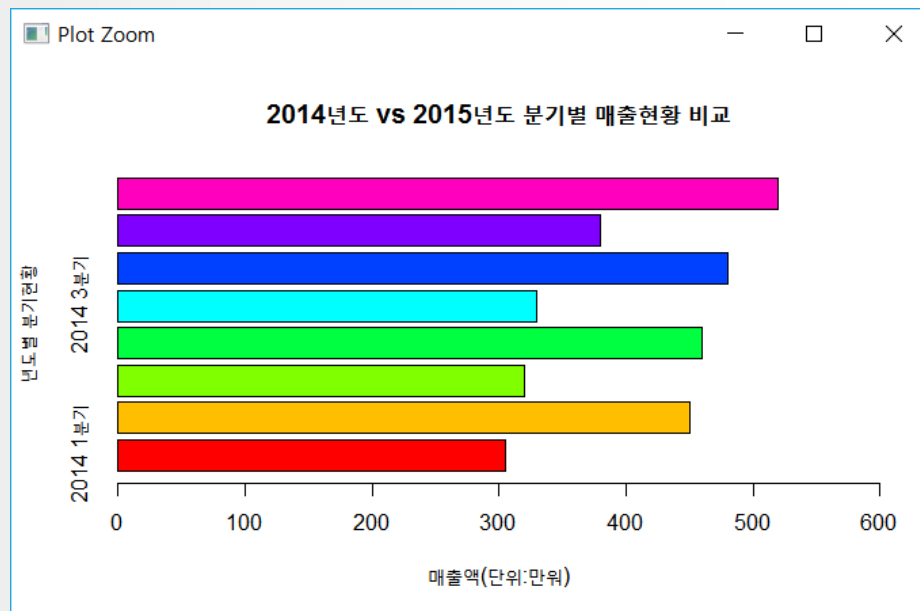




# 1. 이산변수 시각화

## ● 가로막대 차트 시각화

```
barplot(chart_data, xlim=c(0,600), horiz=TRUE,  
        xlab="매출액(단위:만원)", ylab="년도별 분기현황",  
        col=rainbow(8),  
        main = "2014년도 vs 2015년도 분기별 매출현황 비교")
```

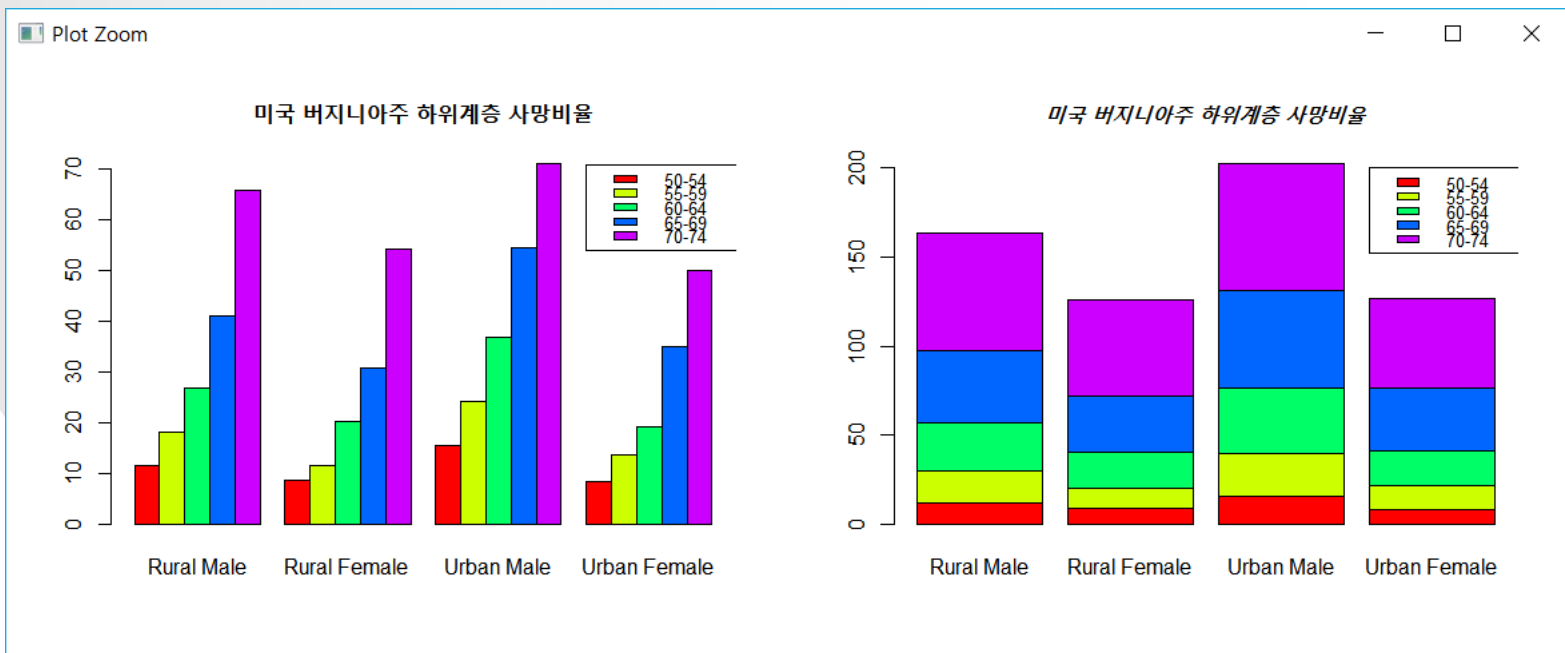




# 1. 이산변수 시각화

## ● 1행 2열 차트 그리기

```
par(mfrow=c(1,2)) # 1행 2열 그래프 보기  
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))
```





# 1. 이산변수 시각화

## ② 점 차트 시각화

- **dotchart()** 형식 – 점 차트 그리기 함수

**help(dotchart)**

```
dotchart(x, labels = NULL, groups = NULL, gdata = NULL,  
  cex = par("cex"), pt.cex = cex,  
  pch = 21, gpch = 21, bg = par("bg"),  
  color = par("fg"), gcolor = par("fg"), lcolor = "gray",  
  xlim = range(x[is.finite(x)]),  
  main = NULL, xlab = NULL, ylab = NULL, ...)
```



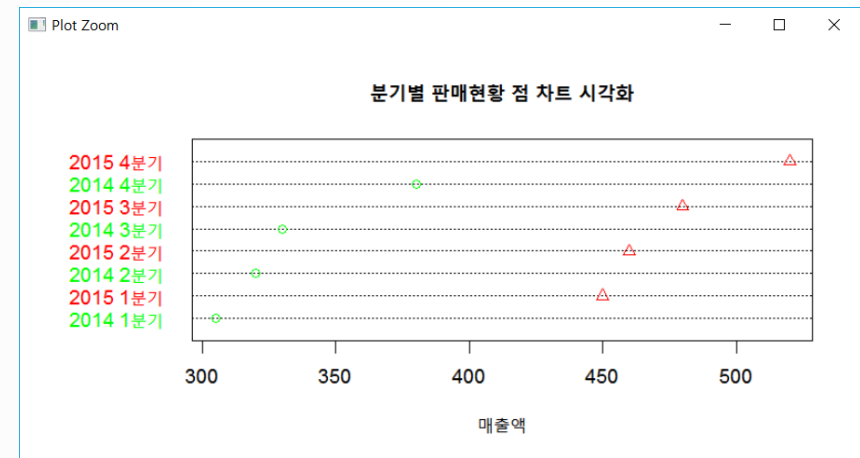


# 1. 이산변수 시각화

## ② 점 차트 시각화

```
dotchart(chart_data, color=c("green","red"), lcolor="black",  
         pch=1:2, labels=names(chart_data), xlab="매출액",  
         main="분기별 판매현황 점 차트 시각화", cex=1.2)
```

```
# col=9:10 -> BR(검정), AR(빨강)  
# col=9:10 -> BR(검정), AR(빨강)  
# lcolor="black" -> 구분선(line) 검정색  
# pch=1:2 -> 점 모양 : 원(1), 삼각형(2), +(3)  
# labels=names(Severity_Counts) : 점 레이블 표시  
# cex=1.2 -> 1.2배 확대(character expansion)
```





# 1. 이산변수 시각화

## ③ 파이 차트 시각화

- **pie()** 형식 – 파일 차트 그리기 함수

**help(pie)**

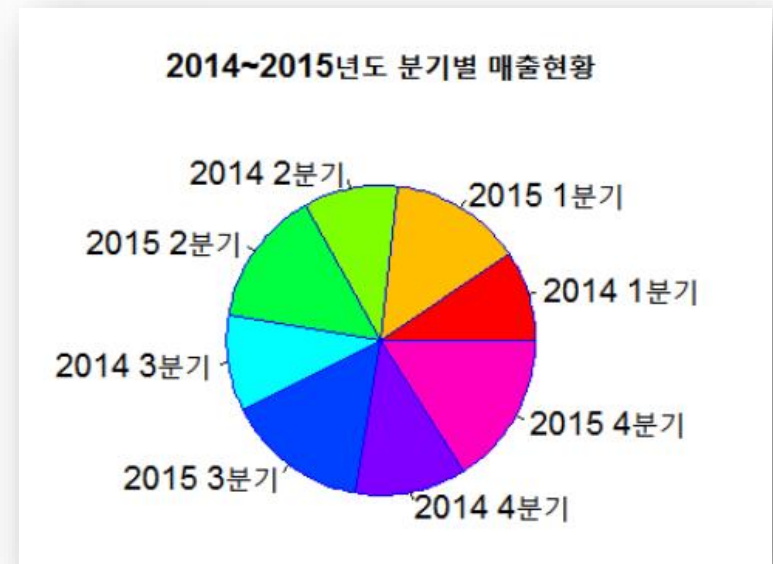
**pie(x, labels = names(x), edges = 200, radius = 0.8,  
clockwise = FALSE, init.angle = if(clockwise) 90 else 0,  
density = NULL, angle = 45, col = NULL, border = NULL,  
lty = NULL, main = NULL, ...)**



# 1. 이산변수 시각화

## ③ 파이 차트 시각화

```
pie(chart_data, labels = names(chart_data),  
border='blue', col=rainbow(8), cex=1.2)  
title("2014~2015년도 분기별 매출현황")
```





## 2. 연속변수 시각화

### 2. 연속변수(Continuous quantitative data)

- 시간, 길이 등과 같이 연속성을 가진 실수 단위 변수값

- 데이터 셋 가져오기

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

```
# range=0 : 최소값과 최대값을 점선으로 연결하는 역할
```

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

```
# notch=T : 중위수 비교시 사용되는 옵션 <- 허리선
```

```
abline(h=37, lty=3, col="red") # 기준선 추가(lty=3 : 선  
스타일-점선)
```



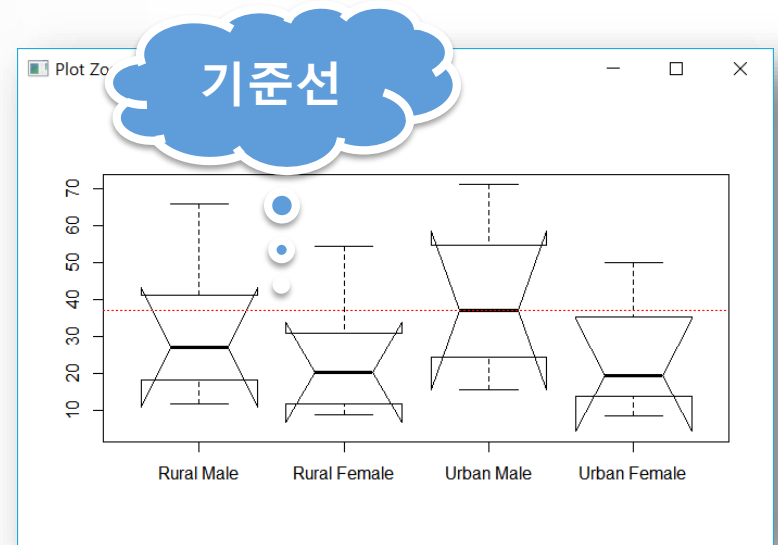
## 2. 연속변수 시각화

### ① 상자 그래프 그래프 시각화

- ✓ 상자 그래프는 요약정보를 시각화한다.
- ✓ 데이터의 퍼짐 정도와 이상치 발견이 목적

#### > summary(VADeaths)

Rural Male	Rural Female	Urban Male	Urban Female
Min. :11.70	Min. : 8.70	Min. :15.40	Min. : 8.40
1st Qu.:18.10	1st Qu.:11.70	1st Qu.:24.30	1st Qu.:13.60
Median :26.90	Median :20.30	Median :37.00	Median :19.30
Mean :32.74	Mean :25.18	Mean :40.48	Mean :25.28
3rd Qu.:41.00	3rd Qu.:30.90	3rd Qu.:54.60	3rd Qu.:35.10
Max. :66.00	Max. :54.30	Max. :71.10	Max. :50.00





## 2. 연속변수 시각화

### ② 히스토그램 시각화

# 데이터 셋 가져오기

```
data(iris) # iris 데이터 셋 가져오기
```

```
names(iris) # "child" "parent"
```

```
str(iris) # 928 2
```

```
head(iris)
```

```
# Sepal.Length Sepal.Width Petal.Length Petal.Width  
Species
```

```
summary(iris$Sepal.Length)
```

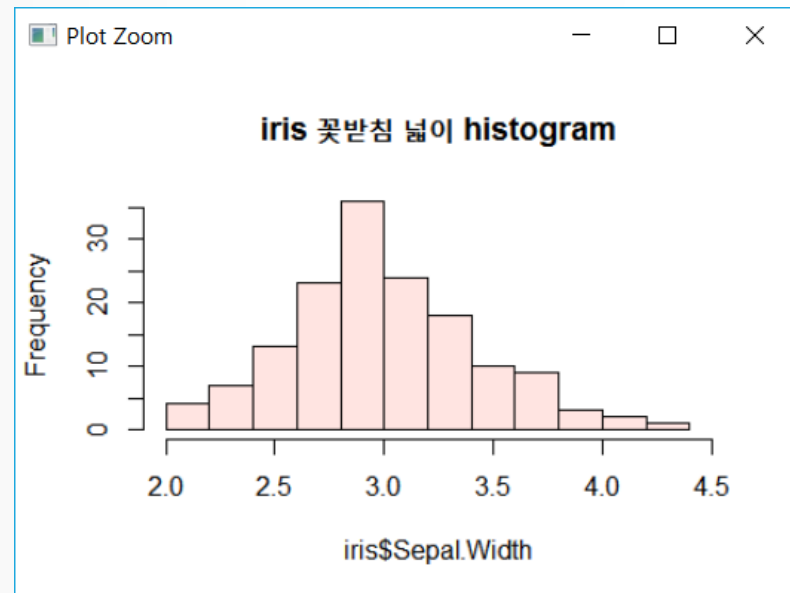
```
summary(iris$Sepal.Width)
```



## 2. 연속변수 시각화

- 히스토그램 시각화(parent)

```
hist(iris$Sepal.Width, xlab="iris$Sepal.Width",  
     col="mistyrose",  
     main="iris 꽃받침 넓이 histogram", xlim=c(2.0, 4.5))  
# col="mistyrose" : 색상(흐릿한 장미) 적용  
# breaks="FD" : Freedman-Diaconis, 구간 너비  
# xlab : x축 이름, main : 제목, xlim : x축 범위
```

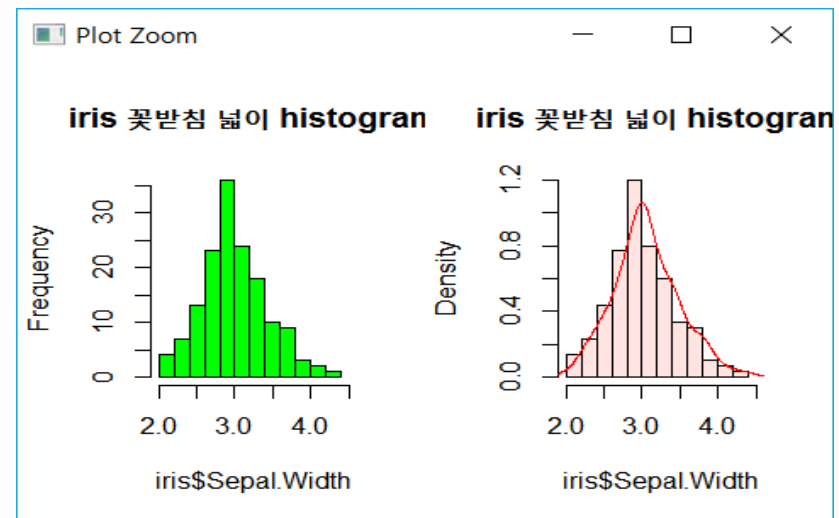




## 2. 연속변수 시각화

### ● 히스토그램 시각화(child)

```
par(mfrow=c(1,2))  
hist(iris$Sepal.Width, xlab="iris$Sepal.Width",  
     col="green",  
     main="iris 꽃받침 넓이 histogram", xlim=c(2.0, 4.5))  
# 확률 밀도로 히스토그램 그리기 - 연속형변수의 확률  
hist(iris$Sepal.Width, xlab="iris$Sepal.Width",  
     col="mistyrose", freq = F,  
     main="iris 꽃받침 넓이 histogram", xlim=c(2.0, 4.5))  
# 밀도를 기준으로 line을 그려준다.  
lines(density(iris$Sepal.Width), col="red")
```



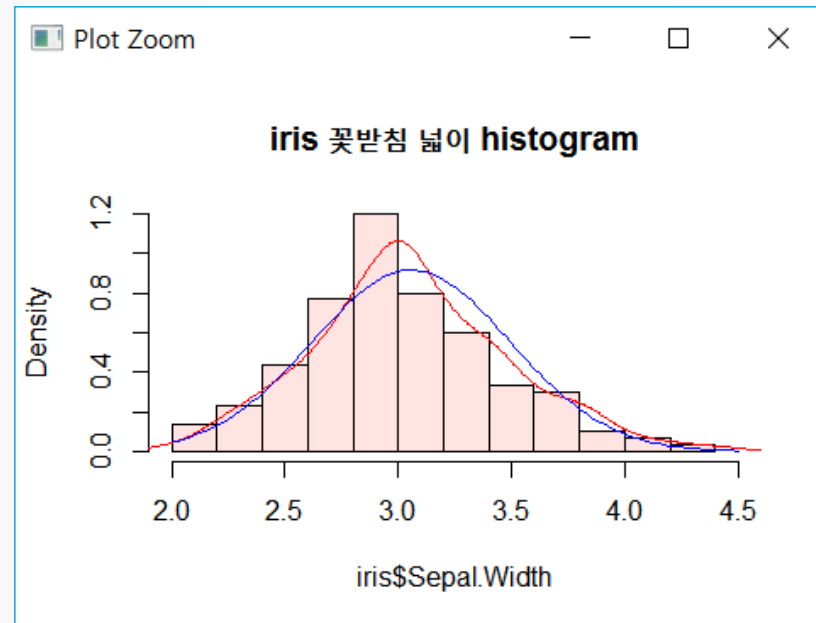




## 2. 연속변수 시각화

- 정규분포 곡선 추정

```
par(mfrow=c(1,1))  
hist(iris$Sepal.Width, xlab="iris$Sepal.Width",  
     col="mistyrose", freq = F, main="iris 꽃받침 넓이 histogram", xlim=c(2.0, 4.5))  
# 밀도를 기준으로 line을 그려준다.  
lines(density(iris$Sepal.Width), col="red")  
curve(dnorm(x, mean=mean(iris$Sepal.Width), sd=sd(iris$Sepal.Width)),  
      col="blue", add = T)
```





## 2. 연속변수 시각화

### ③ 산점도 시각화

`price <- runif(10, min=1, max=100)` # 1~100사이 10개 난수 발생

`price #price <- c(1:10)`

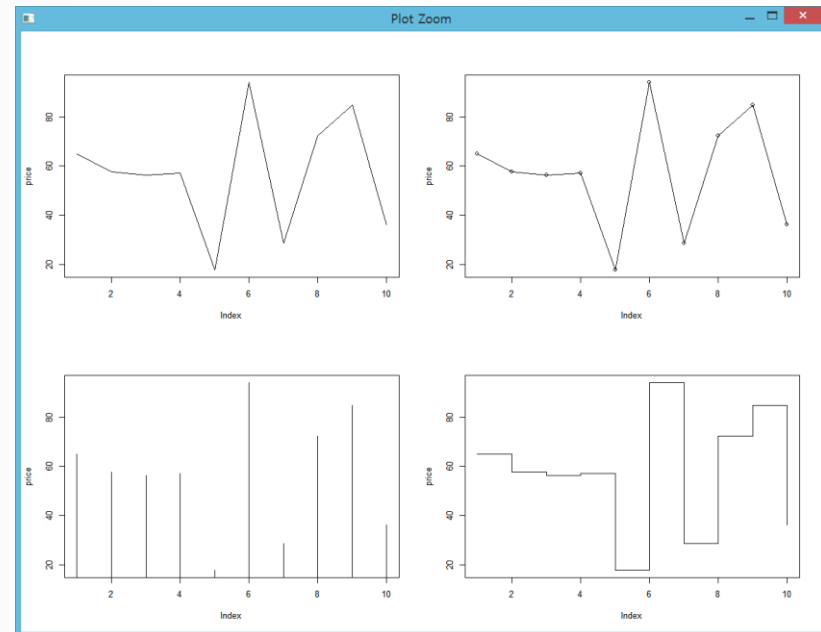
`par(mfrow=c(2,2))` # 2행 2열 차트 그리기

`plot(price, type="l")` # 유형 : 실선

`plot(price, type="o")` # 유형 : 원형과 실선(원형 통과)

`plot(price, type="h")` # 직선

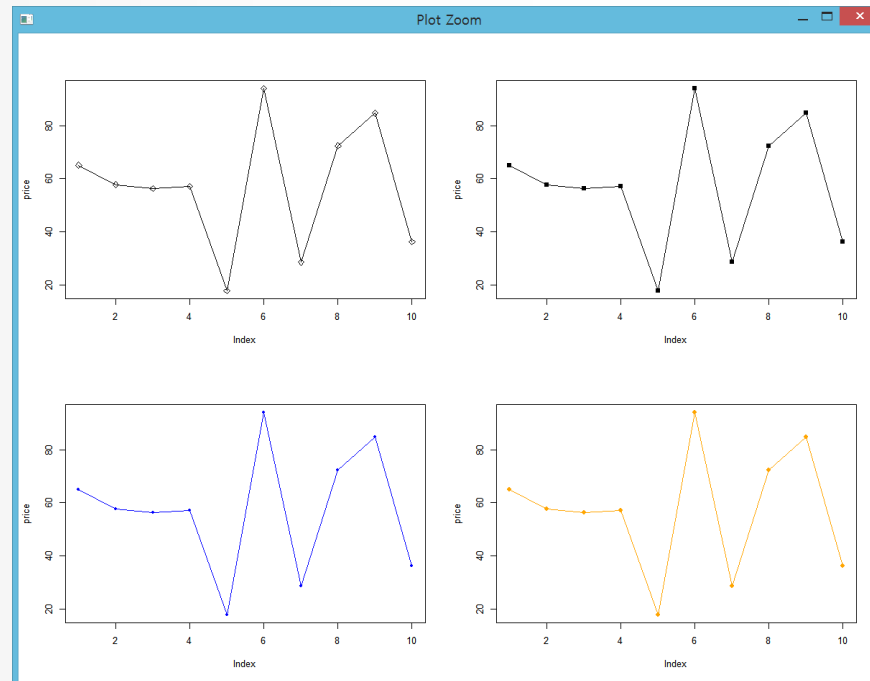
`plot(price, type="s")` # 꺾은선





## 2. 연속변수 시각화

# plot() 함수 속성 : pch : 연결점 문자타입-> plotting character-번호(1~30)  
plot(price, type="o", pch=5) # 빈 사각형  
plot(price, type="o", pch=15)# 채워진 마름모  
plot(price, type="o", pch=20, col="blue") #color 지정  
plot(price, type="o", pch=20, col="orange", cex=1.5) #character expansion(확대)  
plot(price, type="o", pch=20, col="green", cex=2.0, lwd=3) #lwd : line width





## 2. 연속변수 시각화

- 산점도와 회귀선 시각화

# galton 데이터 셋을 이용한 변수 간 상관관계

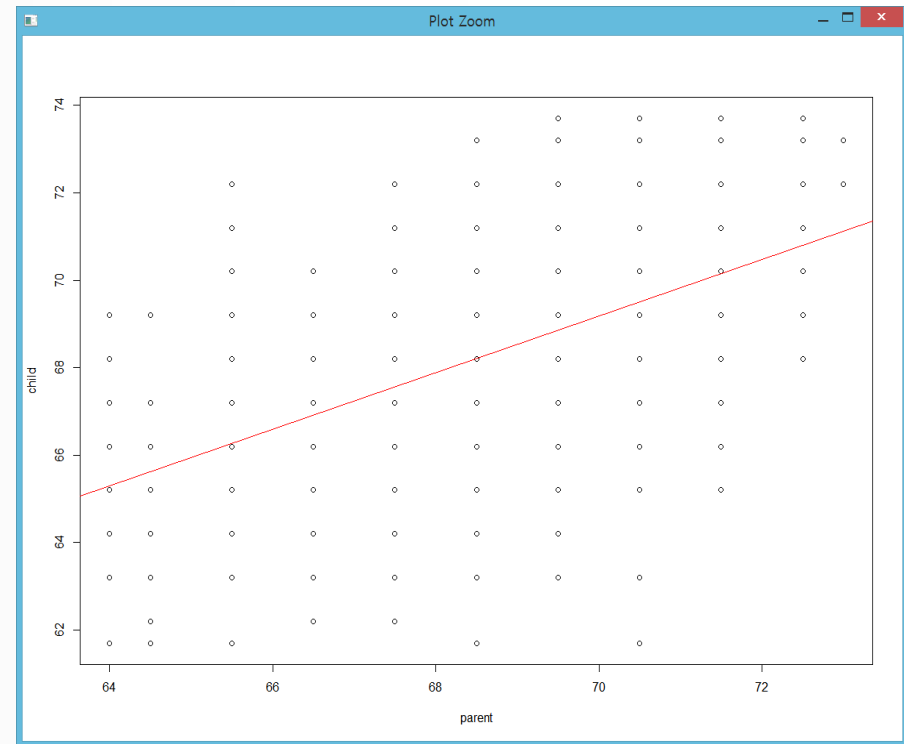
# parent와 child 변수 대상

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

```
plot(child~parent, data=galton)
```

```
out = lm(child~parent, data=galton)
```

```
abline(out, col="red")
```



부모와 자식 간 변수 비교



### 3. 중복 데이터 시각화

- 중복 데이터 시각화

- 시간, 길이 등과 같이 연속성을 가진 실수 단위 변수값

# 1) 데이터프레임으로 변환 : 컬럼 단위의 데이터 활용을 위해서

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

```
freqData # Var1 Var2 Freq(중복 수)
```

```
str(freqData) # 154 obs(928 관측치가 중복 제외한 154개 관측치 생성 )
```

```
names(freqData)=c("child","parent", "freq") # 컬럼에 이름 지정
```

# 2) 프레임 -> 벡터 -> 수치데이터변환, cex : 빈도수에 0.15 곱(가중치 적)

```
parent <- as.numeric(as.vector(freqData$parent))
```

```
child <- as.numeric(as.vector(freqData$child))
```

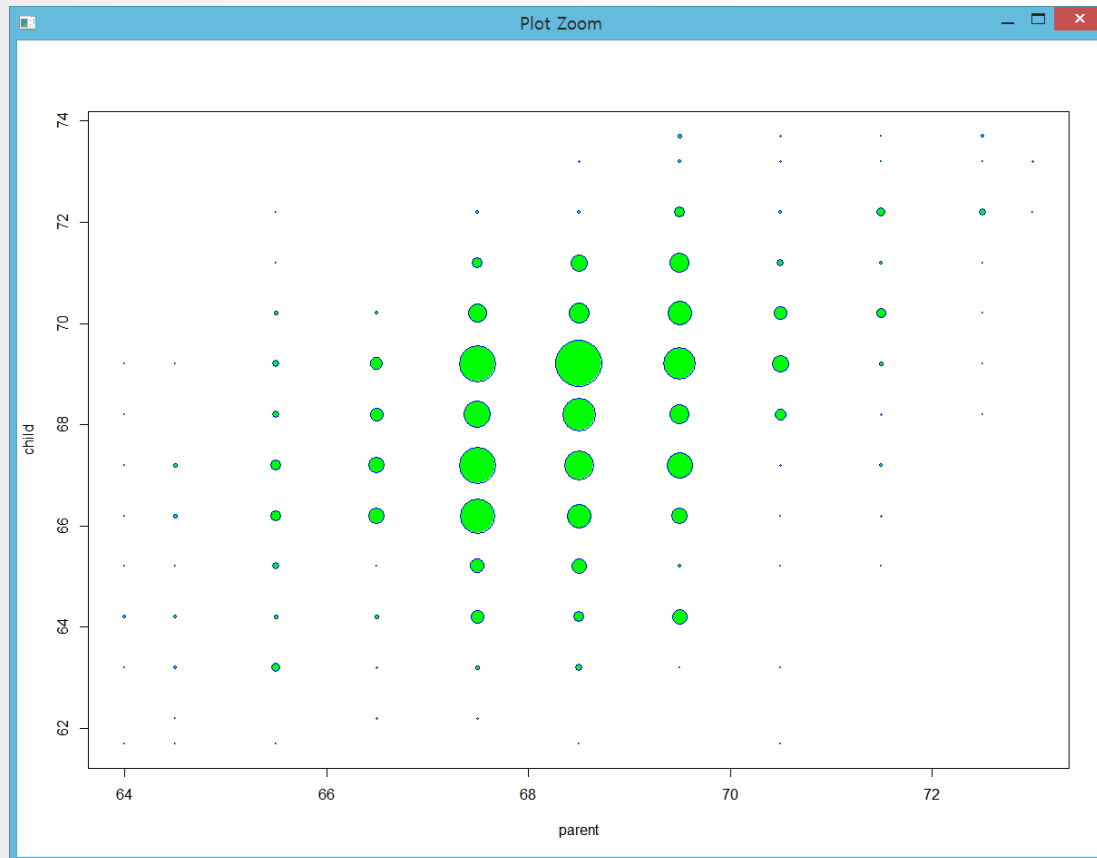
```
plot(child~parent, pch=21, col="blue", bg="green",
```

```
      cex=0.15*freqData$freq, xlab="parent", ylab="child")
```



### 3. 중복 데이터 시각화

- 빈도수를 적용한 가중치 적용

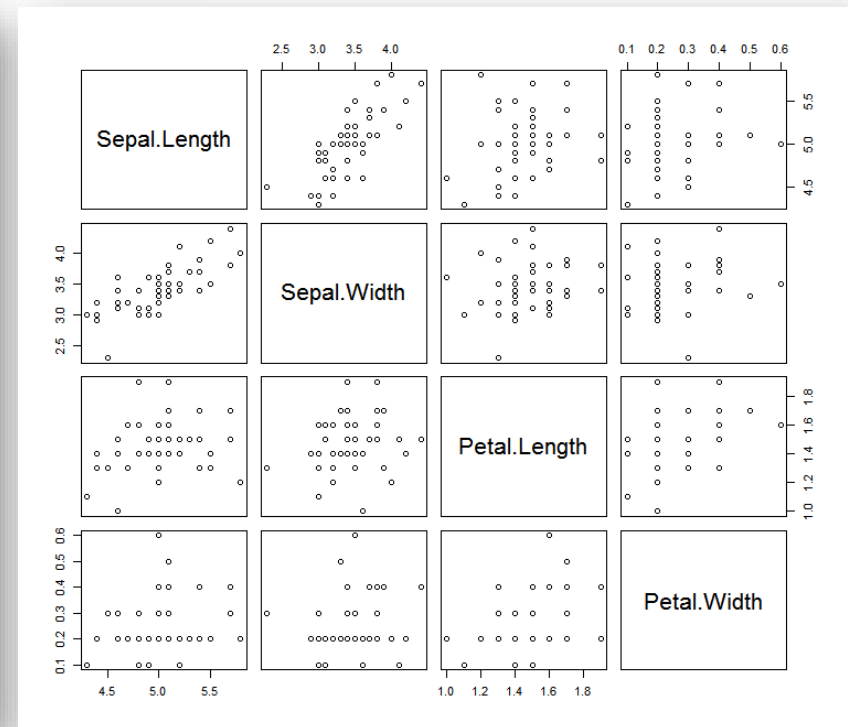
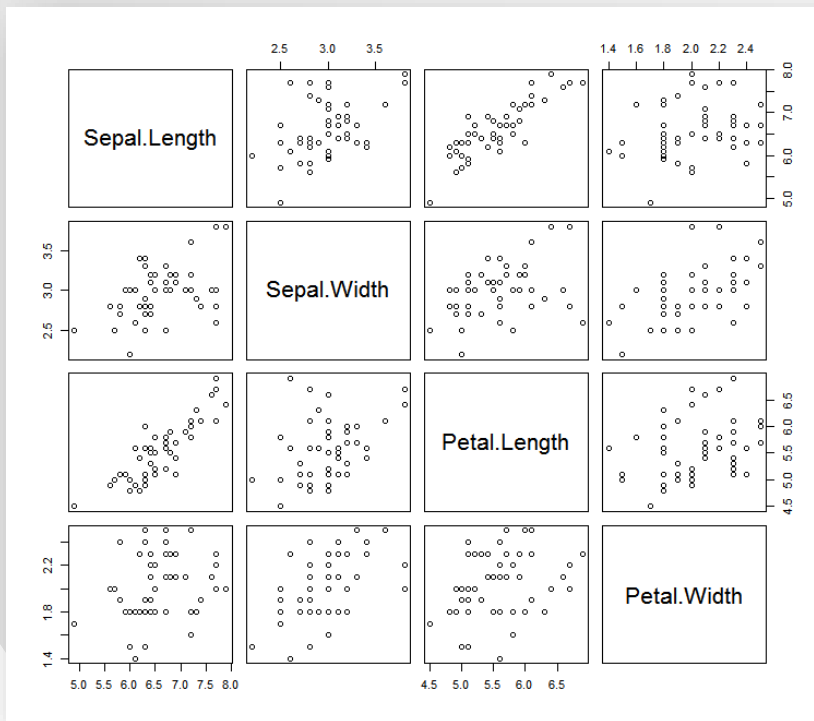


부모와 자식 간 변수 비교



## 4. 변수 간 산점도 matrix

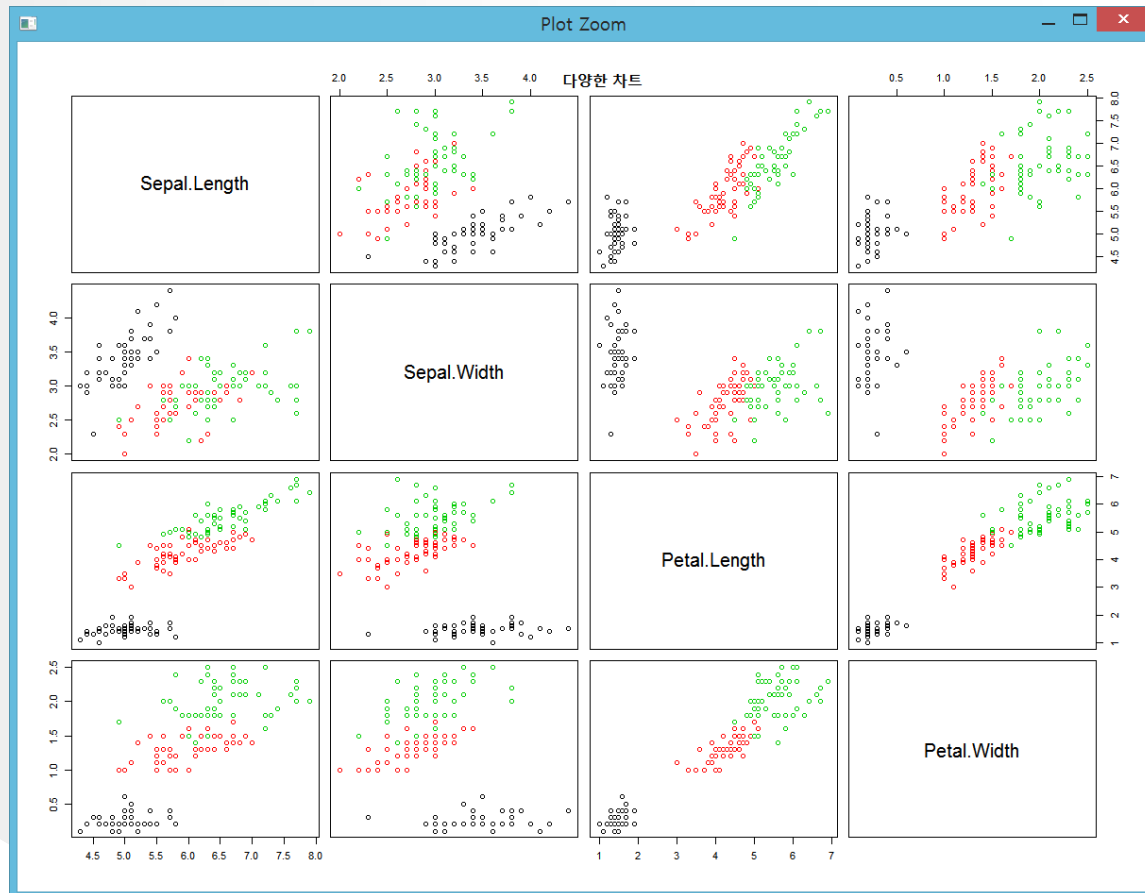
- 변수간 비교 시각화 결과





## 4. 변수 간 산점도 matrix

`plot(iris) # iris 데이터를 대상으로 제공되는 모든 차트 그려줌`  
`plot(iris[, -5], col=iris[,5]) # 5번컬럼 제거, 색지정으로 사용`  
`title(main="다양한 차트")`







## 5. 차트 결과 파일 저장

- 파일로 차트 저장하기

```
setwd("C:/Rwork/Part-II") # 폴더 지정
```

```
jpeg("iris.jpg", width=720, height=480) # 픽셀 지정 가능
```

```
plot(iris$Sepal.Length, iris$Petal.Length, col=iris$Species)
```

```
title(main="iris 데이터 테이블 산포도 차트")
```

```
dev.off() # 장치 종료
```

```
# "c:/Rwork/Part-II" <- 이미지 파일 확인
```



## 6. 변수와 시각화 도구

- 칼럼의 속성에 따른 시각화 도구 분류
  - 칼럼 수와 자료의 형태에 따라서 시각화 도구가 달라진다.

칼럼 특성			시각화 도구
칼럼 수	수치형	범주형	
1	1		hist, plot, barplot
1		1	pie, barplot
2	2		plot, abline, boxplot
3	3		scatterplot3d
n	n	n	pairs