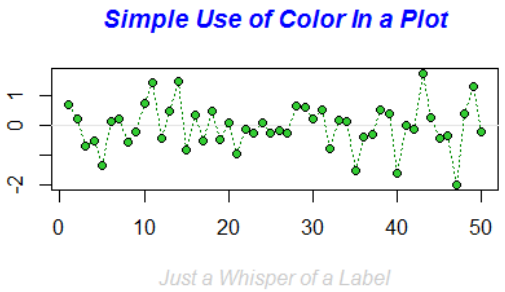
[6] data visualization

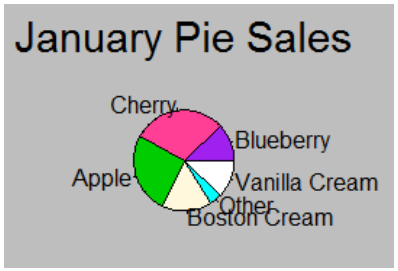
data visualization

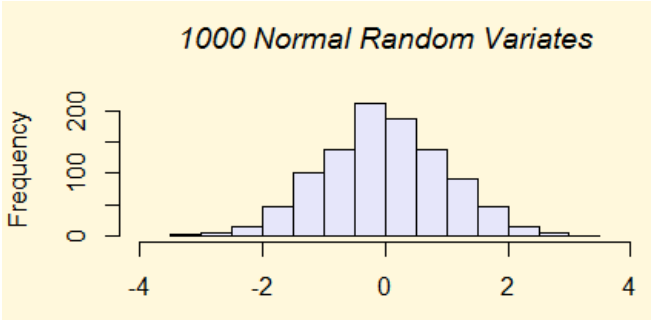
- Graphics package
- plot(), curve()

graphics

- graphics 패키지는 R의 가장 기본이 되는 시각화 기능을 지원
- 전체 함수의 목록은 library(help = “graphics”)를 통해 확인
- 그래프 함수 (고수준(high-level) 그래픽 함수)
  - plot() 산점도(scatter plot) / barplot() 막대그래프 / curve() 함수그래프 / pie() 원그래프
  - hist() 히스토그램 /boxplot()
- 그래프 속성 정의 함수(저수준(low-level) 그래픽 함수)
  - 그래프에 점, 선, 면, 문자, 좌표축, 범례 등의 다양한 그래프의 속성을 정의하는 함수
  - title() 제목 , legend() 범례, point() 점, abline() 직선, text() 문자 등

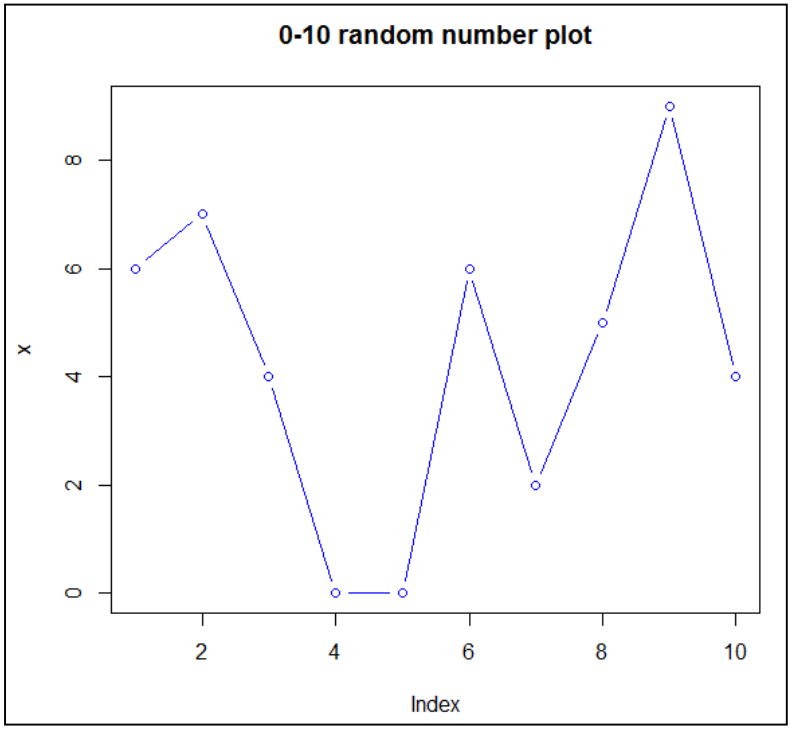




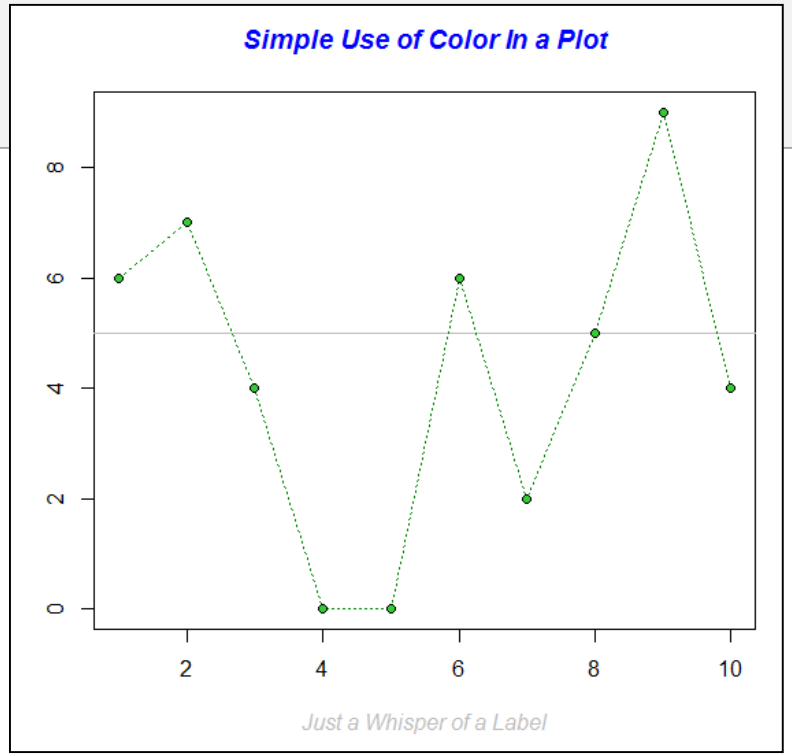


plot() 예

```
#데이터 준비
(x = runif(10))                 # 0-1사이의 10개 난수발생
(x = as.integer(x * 10))   # 0-10사이 정수 값으로 변경
# x를 파란색 점선타입으로 plot, 제목 설정
plot(x, main="0-10 random number plot", type="b",  col="blue")
```



```
#비어있는 plot에  선, 점을 그리고 제목을 설정
plot(x,  ann=FALSE, type = "n")  #x,y축 라벨 없이, no plotting.
abline(h = 5, col = "gray")  #gray 색상으로 y축 5에 선그리기
lines(x, col = "green4", lty = "dotted") # x값을 그린4 색상, 점선으로 그리기
points(x, bg = "limegreen", pch = 21) #x값을 limegreen색으로 채워진 원점으로 그리기
#주제목(main), x축 제목(xlab) 설정
title(main = "Simple Use of Color In a Plot",     xlab = "Just a Whisper of a Label",
      col.main = "blue", col.lab = "gray",   cex.main = 1.2, cex.lab = 1.0, font.main = 4,
font.lab = 3)
plot.new ()  #plot 지우기, frame()
```



plot()

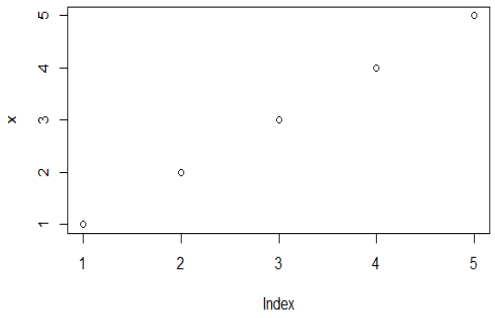
- plot()　　: 객체들을 도표상에 표시(plot)하는 함수
- 형식

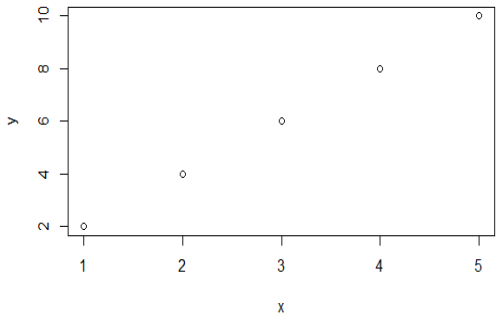
  plot(y축, 옵션)

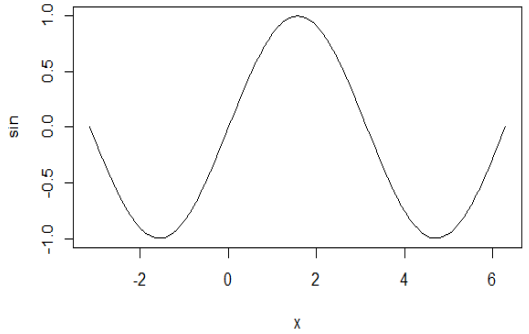
  plot(x축, y축, 옵션)

  plot(함수명, x축 하한선, x축 상한선)

```
x = 1:5
y = seq(2,10,2)
plot(x)
plot(x,y)
plot(sin,-pi, 2*pi)     #함수 그래프
curve(sin,-pi, 2*pi)    #함수 그래프
```

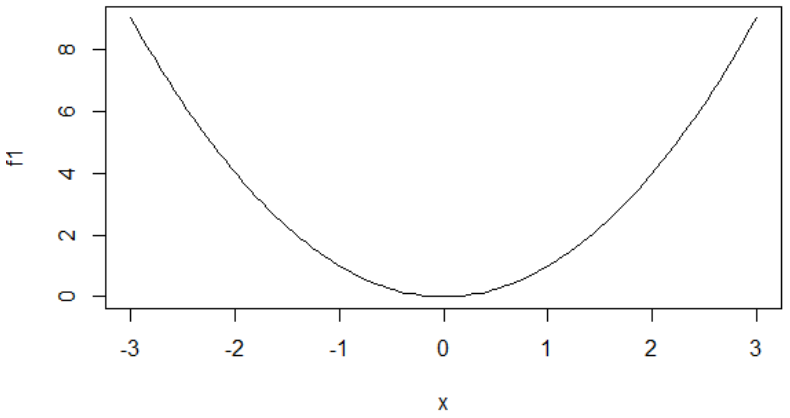


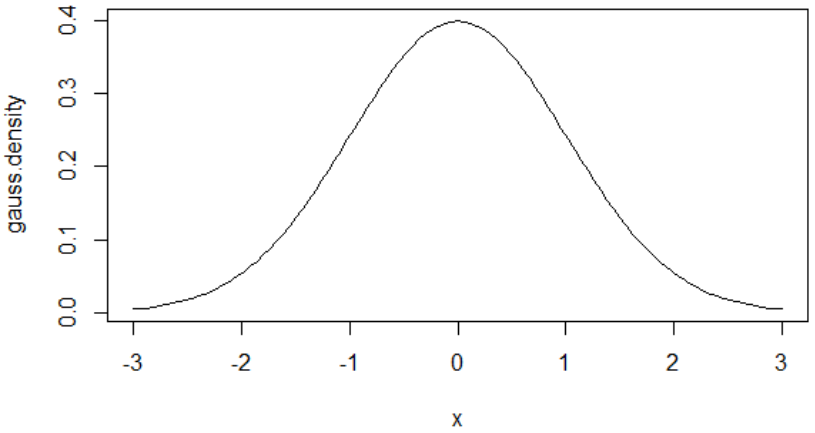




plot()

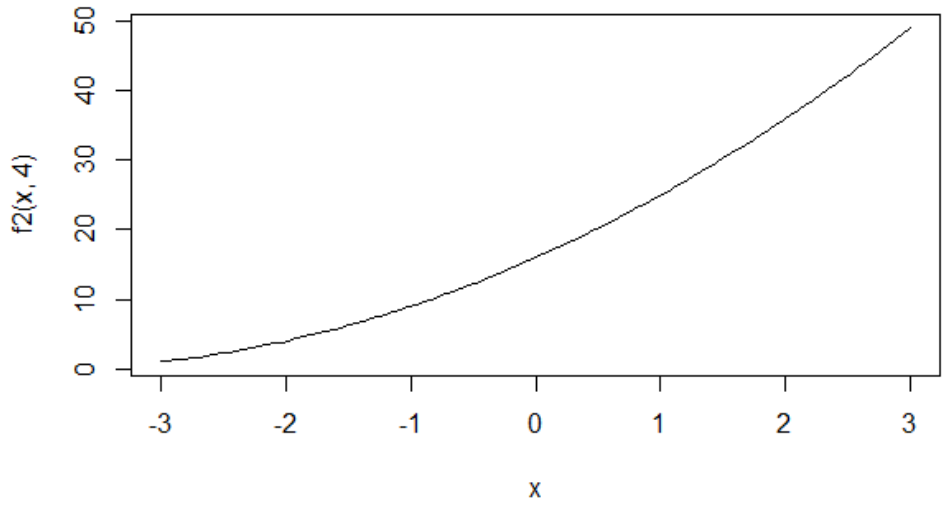
```
#사용자 정의 함수 그래프 예
f1 = function(x){
  x^2
}
plot(f1, -3,3)
curve(f1(x), -3, 3)

#표준 정규 분포 함수
gauss.density=function(x){
  1/ sqrt(s*pi) * exp(-x^2 /2)
}
plot(gauss.density,-3, 3)
```





curve()

```
#함수 인수 2개 이상일 경우  함수 그래프 예


f2 = function(x,y){
  2*x*y + x^2+y^2
}
curve(f2(x, 4), -3,3)
```



plot() 그래프 옵션

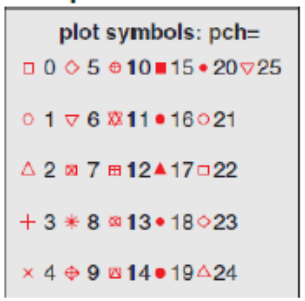
- 그래프 옵션

파라미터	Option 및 설명
type =	그래프의 형태를 지정 type="p" 점(point) 그래프 type="l" 선(line) 그래프 type="b" 점과 선으로 이어서 그림 type="o" 선이 점 위에 겹쳐진 형태 type="h" 수직선으로 그림 type="s" 계단(step)형 그래프
xlim = ylim =	x축과 y축의 상한과 하한. xlim = c(1,10) 또는 xlim = range(x)
xlab= ylab=	x축과 y축의 이름(label) 부여
main =	그래프의 위쪽에 놓이는 주 제목(main title).
sub =	그래프의 아래쪽에 놓이는 소 제목(subtitle).
bg=	그래프의 배경화면 색깔

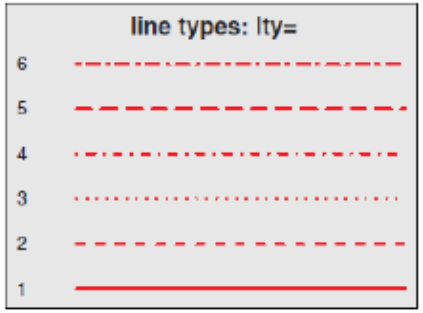
plot() 그래프 옵션

파라미터	Option 및 설명
pch =	표시되는 점의 모양
lty =	선의 종류 1: 실선(solid line) 2: 파선 (dashed) 3: 점선: 점선(dotted) 4: dot-dash
col=	색깔 지정: "red","green","blue" 및 색상을 나타내는 숫자
mar =	c(bottom, left, top, right) 의 순서로 가장자리 여분 값을 지정. 디폴트는 c(5,4,4,2) + 0.1
asp =	종횡의 비율 Apsect ratio (= y/x)

- pch



- lty

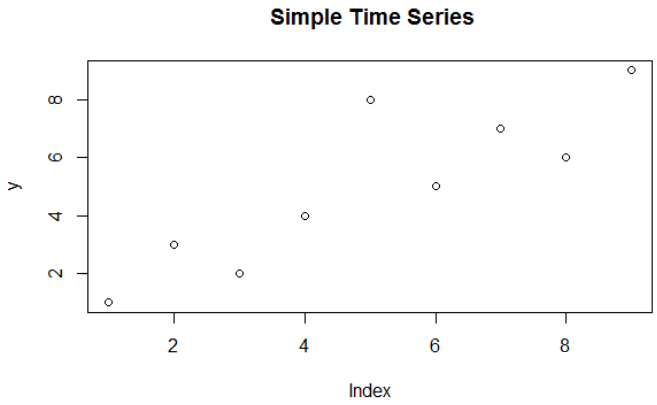


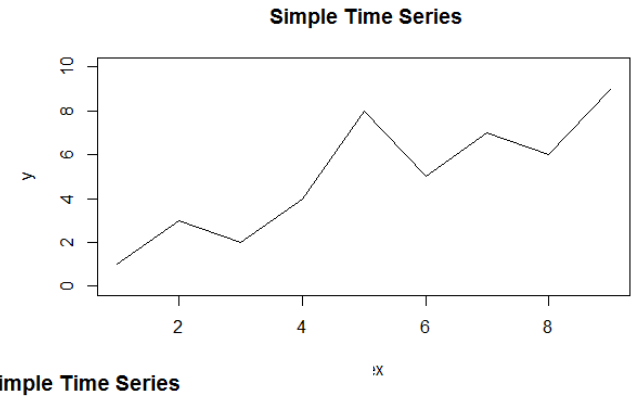
- Colors()로 색상 확인

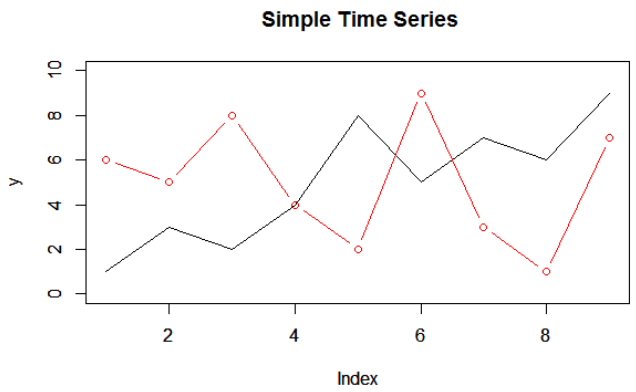
```
> colors()
  [1] "white"              "aliceblue"          "antiquewhite"       "antiquewhite
1"
  [5] "antiquewhite2"      "antiquewhite3"      "antiquewhite4"      "aquamarine"
  [9] "aquamarine1"        "aquamarine2"        "aquamarine3"        "aquamarine4"
 [13] "azure"              "azure1"             "azure2"             "azure3"
 [17] "azure4"             "beige"              "bisque"             "bisque1"
 [21] "bisque2"            "bisque3"            "bisque4"            "black"
 [25] "blanchedalmond"     "blue"               "blue1"              "blue2"
 [29] "blue3"              "blue4"              "blueviolet"         "brown"
 [33] "brown1"             "brown2"             "brown3"             "brown4"
 [37] "burlywood"          "burlywood1"         "burlywood2"         "burlywood3"
```

plot()

```
y= c(1,3,2,4,8,5,7,6,9)
plot(y, main=" Simple Time Series")

plot(y, main=" Simple Time Series", ylim=c(0,10), type="l") #제목, y축 범위, type 선 설정

par(new = T)    #그래프 겹쳐 그리기
z = c(6,5,8,4,2,9,3,1,7)
plot(z, ylim=c(0,10), type="b", ylab="", col="red") #type 선,점 선택, 색상 red
```

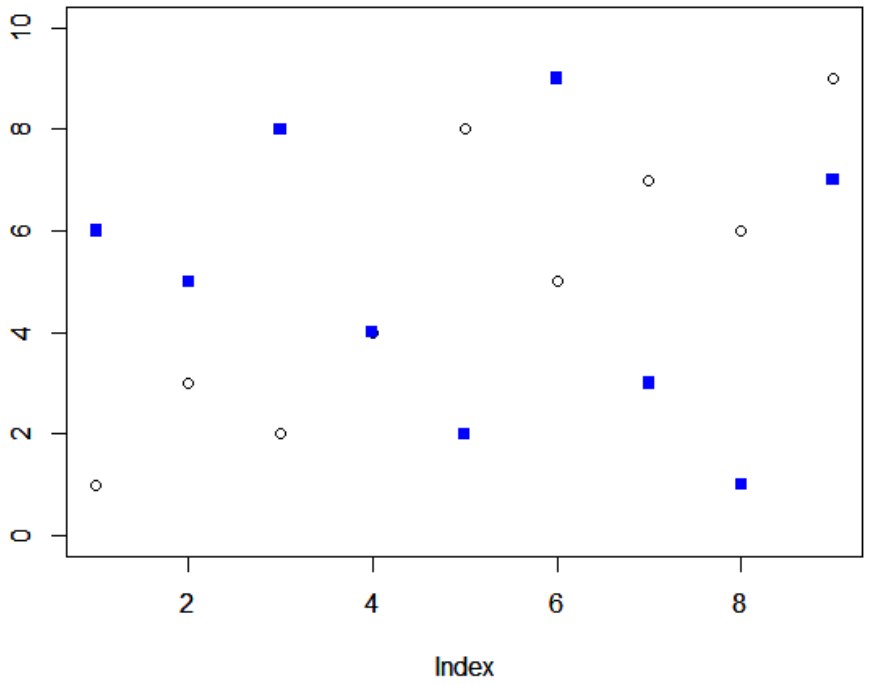






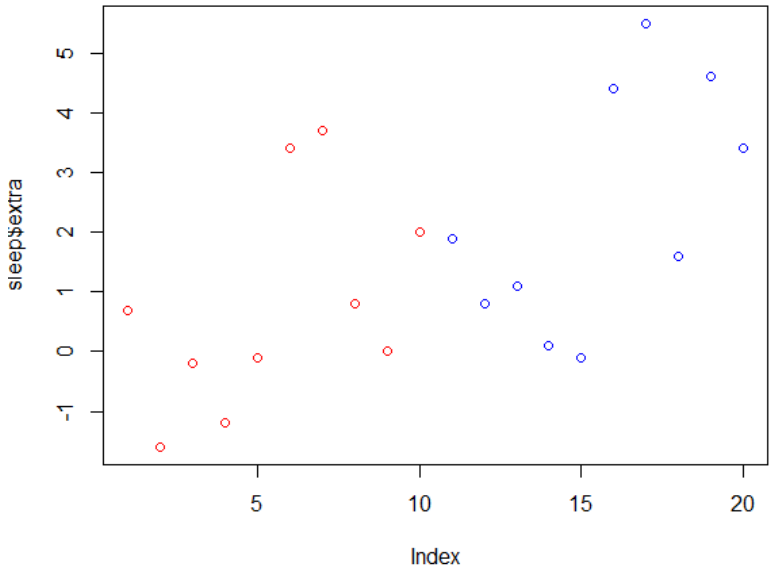
plot()

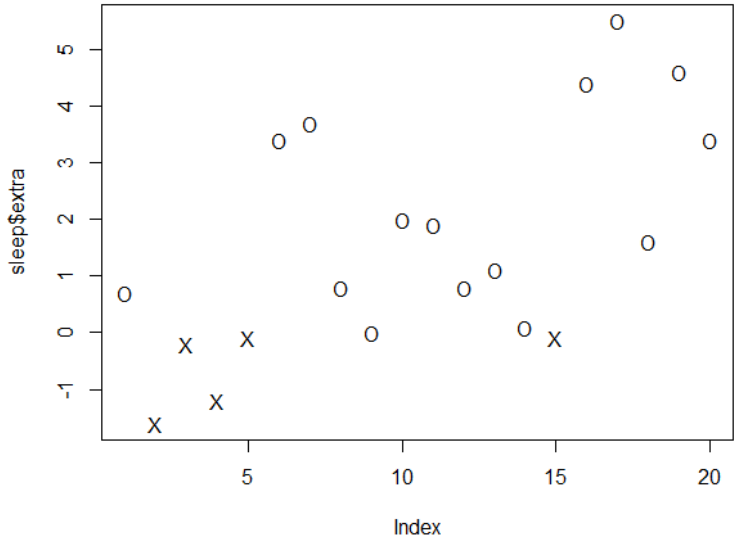
```
y= c(1,3,2,4,8,5,7,6,9)
z = c(6,5,8,4,2,9,3,1,7)

plot(y, ylim=c(0,10), ylab="", pch=1)  # 원모양의 점 플롯
par(new = T)
plot(z, ylim=c(0,10), ylab="", pch=15, col="blue") #파란색 사각점 플롯
```



plot()

```
#조건에 따른 모양 설정
data(sleep)
plot( sleep$extra, col=ifelse(sleep$group == 1,"red","blue"))
plot( sleep$extra, pch=ifelse(sleep$extra < 0,"X","O"))
```

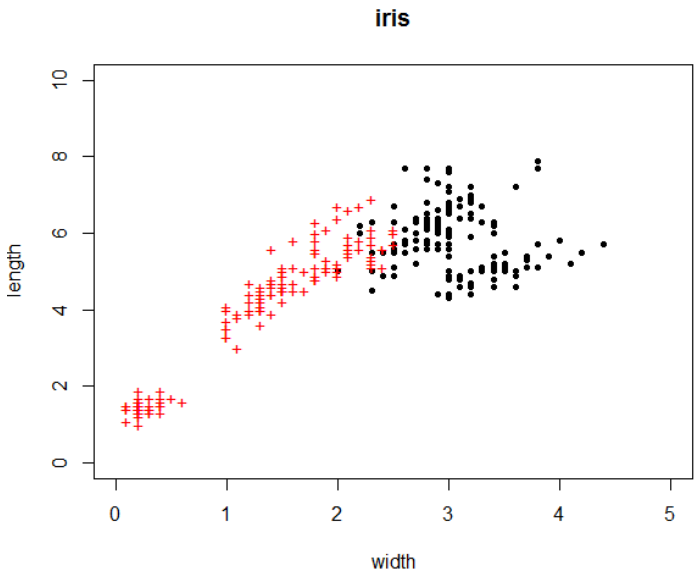




points

- points() : 이미 그려진 plot 에 추가로 점을 표시

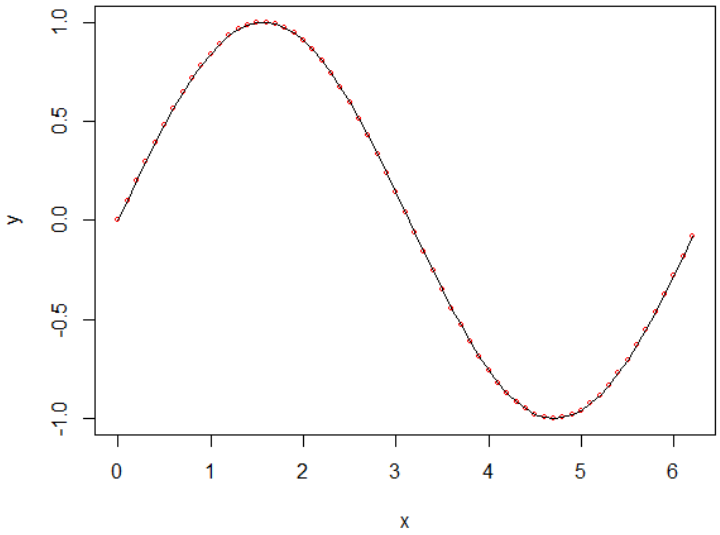
```
with (iris , {
   plot (NULL , xlim =c(0, 5) , ylim =c(0, 10) ,
         xlab =" width ", ylab =" length ", main =" iris ", type ="n")
   points ( Sepal.Width , Sepal.Length , pch =20)
   points ( Petal.Width , Petal.Length ,  pch="+", col="red")
   })
```



lines

- lines() : plot()으로 그래프를 그린 뒤 선을 추가

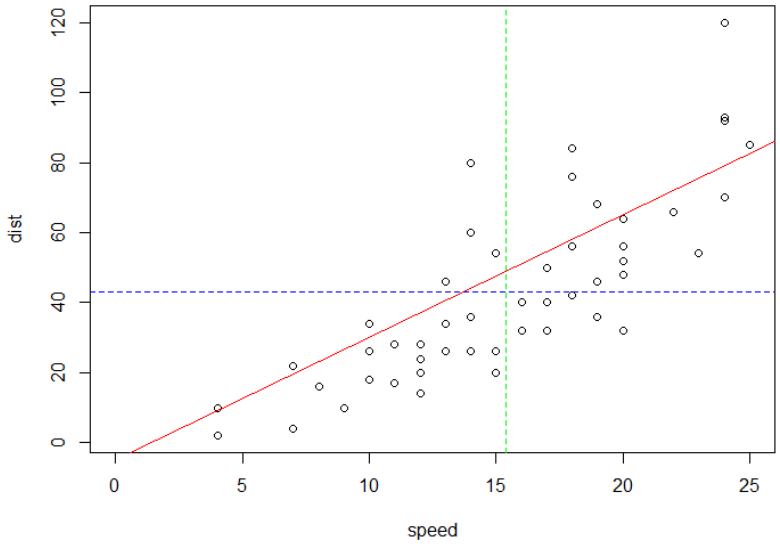
```
#[0, 2π]까지 sin 그래프
x = seq (0, 2*pi , 0.1)
y = sin(x)
plot (x, y, cex=.5 , col ="red ")  #점 크기를 작게(cex  디폴트 1)
lines (x, y)
```



abline

- abline() : 직선을 그리는 함수
  abline(h=y) : y좌표에 수직인 수평선
  abline(v=x) : x좌표에 수직인 수 직선
  abline(a,b) : y = a + bx 형태의 직선(a:절편, b : 기울기)

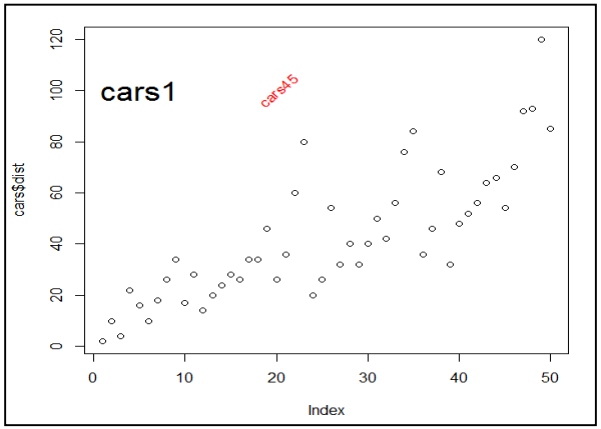
```
plot (cars , xlim =c(0, 25) )
abline (a=-5, b=3.5 , col ="red ")    #절편 -5, 기울기 b인 직선
abline (h= mean ( cars $ dist ), lty =2, col =" blue ") #dist 평균 수평선
abline (v= mean ( cars $ speed ), lty =2, col =" green ") #speed 평균 수직선
```

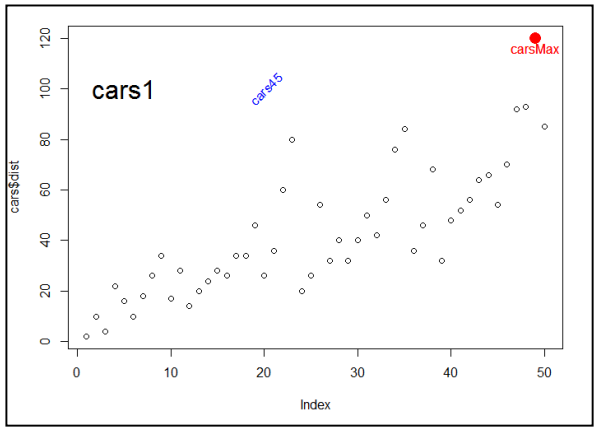


text

- text() : 그래프에 문자를 그리는 함수
- 형식 : text(x, y, labels, srt)
  - x,y좌표에 labels를 표시
  - srt : 각도

```
plot (cars$dist)
text(5, 100, "cars1", cex=2)  #5,100위치에 크기 2, 문자

#20,100위치에  45도 각도로 빨간색 문자
text(20, 100, "cars45", srt=45, col='red')

#최대값의 위치에 문자와  빨간색 큰점 그리기
max = which.max(cars$dist)
text(max, cars$dist[max]-4, "carsMax", col='red')
points(max, cars$dist[max], pch=16, col='red', cex=2)
```

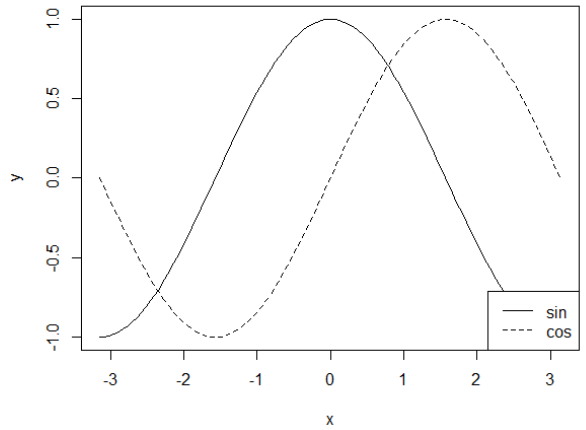




legend

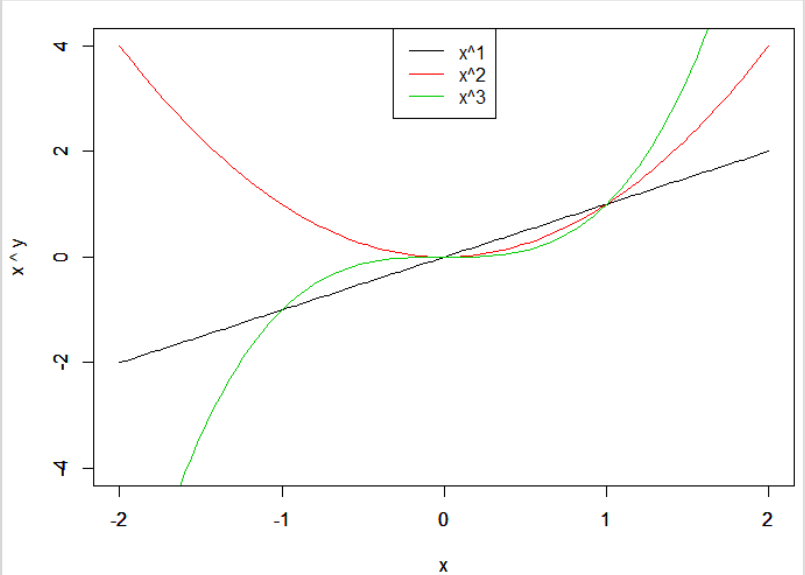
- legend() : 범례(기호 설명표)를 표시
- 형식: legend(x, y=NULL, lty, pch, col, legend)
  - 범례가 보여질 (x, y) 좌표를 지정하거나 미리 정의된 키워드(bottomright, bottom, bottomleft, left, topleft, top, topright, right, center) 중 하나로 범례의 위치를 지정
  - 선종류(lty), 점종류(pch), 선종류(col)등 지정
  - legend : 설명 (문자열 벡터) 지정

```
plot(sin, -pi, pi, xlab="x", ylab="y", lty=1)
plot(cos, -pi, pi, lty=2, add=T)
legend("bottomright", lty=1:2, c("sin", "cos"))
```



legend

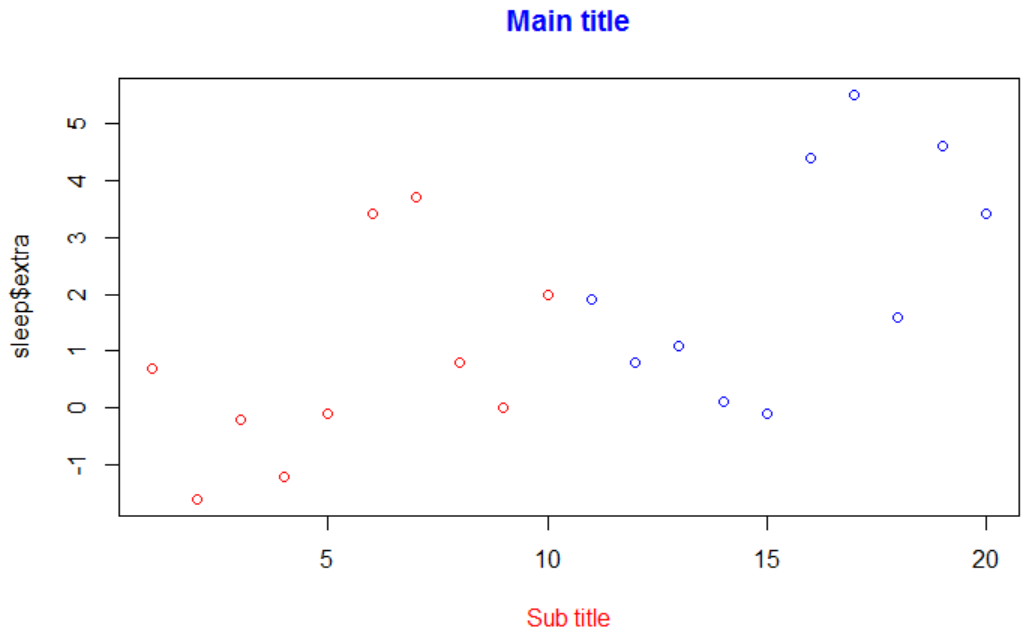
```
#x^1, x^2, x^3 그래프에 범례지정 예
square = function(x,y) x ^ y
curve(square(x, 1), xlim=c(-2,2), ylim=c(-4,4), ylab="x ^ y“, col=1)
par(new=T)      #겹쳐그리기
curve(square(x, 2), xlim=c(-2,2), ylim=c(-4,4), ylab="", col=2)
par(new=T)      #겹쳐그리기
curve(square(x, 3), xlim=c(-2,2), ylim=c(-4,4), ylab="", col=3)
legend("top", lty = 1, col=1:3, c("x^1", "x^2", "x^3"))  #범례
```



title

- title() : 제목과 부제목 그리는 함수

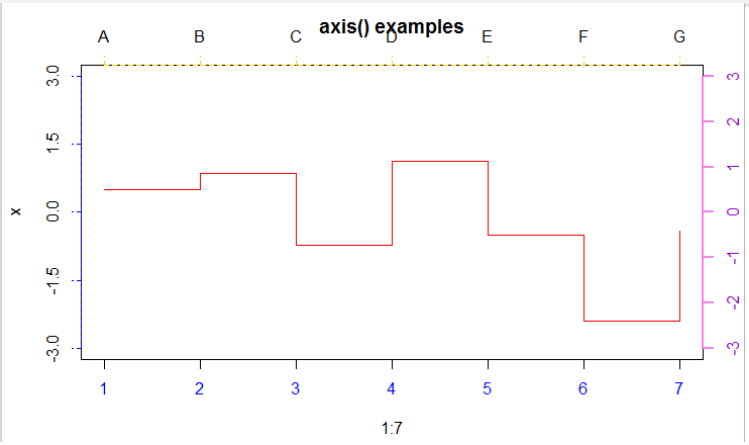
```
data(sleep)
plot( sleep$extra, col=ifelse(sleep$group == 1, "red","blue"), xlab="")
title(main = "Main title",       xlab = "Sub title",  col.main = "blue", col.lab = "red" )
```



axis

- axis() : 좌표축 그리기 함수

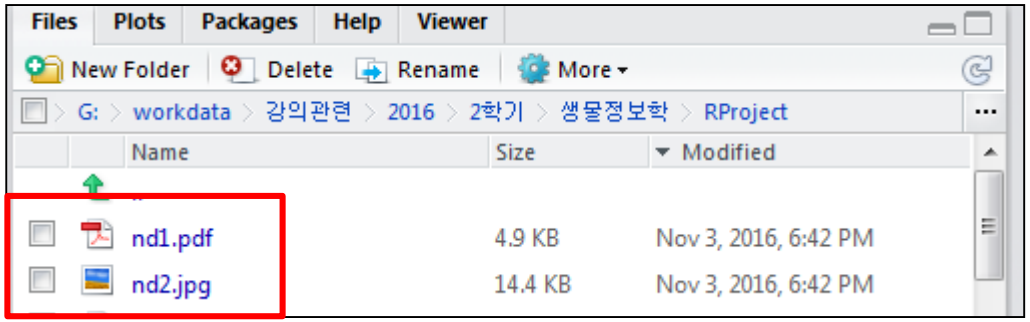
```
# Normal Distribution
#평균 0, 표준편차 1인 정규분포를 따르는  임의수 7개 추출 하여 그리기
x = rnorm(7)
#축을 없애고  그리기
plot(1:7, x , main = "axis() examples", type = "s", ylim=c(-3, 3), axes = F, col = "red")
box()  #테두리 그리기
#axis() 축 그리기, side : 1=below, 2=left, 3=above and 4=right
#아래축 라벨을 파란색으로 지정
axis(side=1, col.axis = "blue")
#왼쪽 축, -3~3까지 4개 눈금,파랑색 점선 굵기 0.5,
axis(side=2, yaxp=c(-3,3, 4), col = "blue", lty = 2, lwd = 0.5)
#위축, 1:7 범위를 A:G로 표기, 금색 점선 굵기 0.5,
axis(side=3, 1:7, LETTERS[1:7], col = "gold", lty = 2, lwd = 0.5)
#오른쪽축, 보라색 굵기2 선, 축라벨색 어두운 보라색
axis(side=4, col = "violet", col.axis = "dark violet", lwd = 2)
```



그래프 저장

- png(), jpeg(), pdf() : 파일로 저장하는 함수

```
##정규분포함수 그래프를 파일에 저장
#파일디바이스를 열어서 그리기
pdf(file="nd1.pdf") #pdf 디바이스 열기
curve(dnorm, -4, 4, main="Normal Distribution")  #그리기
dev.off()  #디바이스 닫기
#그래프 디바이스에 그려진 그래프를  파일디바이스로 복사
curve(dnorm, -4, 4, main="Normal Distribution") # 그리기
dev.copy(jpeg, "nd2.jpg")  #jpg 디바이스로 출력
dev.off()  #디바이스 닫기
```

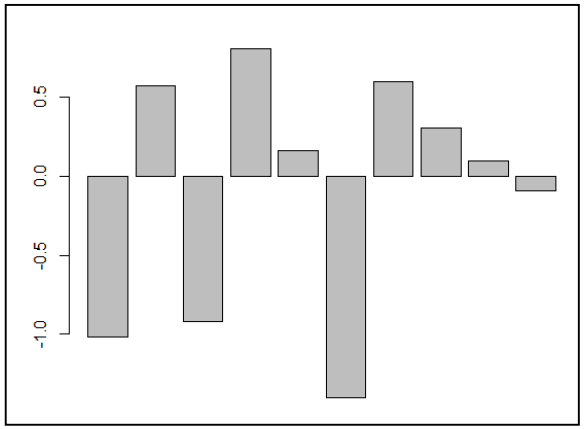


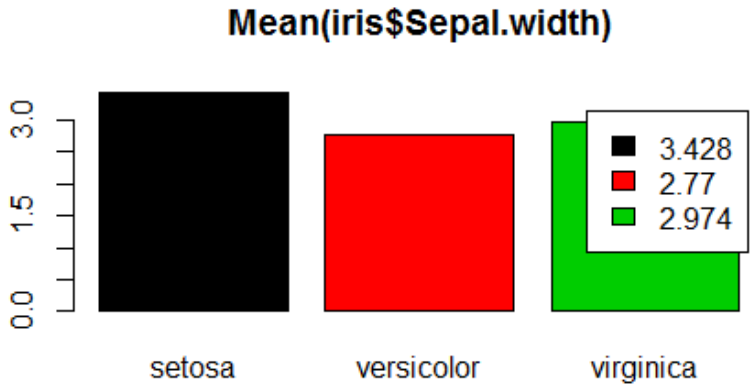
현재 작업디렉토리에 파일로 저장

barplot

- barplot() : 막대 그래프를 그리는 함수

```
##정규분포를 따르는 난수 10개에 대한 막대그래프 예
(b = rnorm(10))
barplot(b)


#iris 데이터셋의 Sepal.Width의 종별 평균에 대한  막대 그래프 예
m = tapply ( iris $ Sepal.Width , iris $ Species , mean )
barplot (m , main="Mean(iris$Sepal.width)", col=c(1,2,3), legend.text= c(m))
```

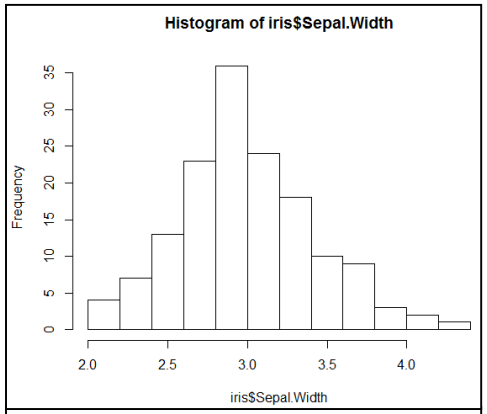


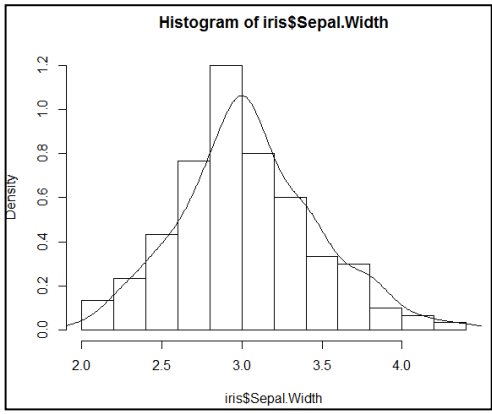


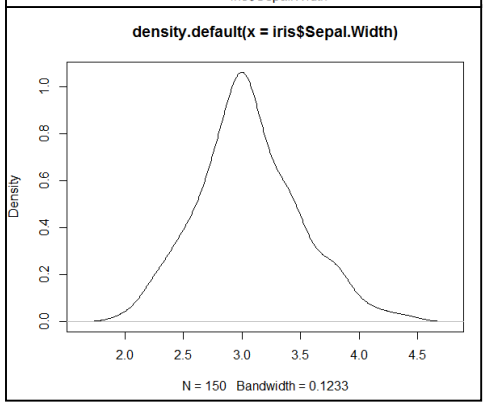
히스토그램

- Histogram : 도수 분포의 상태를 막대 그래프로 나타낸 것
- hist() : 자료의 분포를 확인할 수 있는 히스토그램을 그리는 함수
  - 각 구간별 빈도수를 그리거나 확률 밀도(density)를 그리기 위한 함수
  - freq = NULL : 구간별 빈도수로 그리기, FALSE이면 각 구간의 확률
  - 확률 밀도 그래프 : density() 함수를 사용하여 확인

```
hist ( iris $ Sepal.Width )

hist ( iris $ Sepal.Width , freq = FALSE )
lines ( density ( iris $ Sepal.Width ))

plot ( density ( iris $ Sepal.Width ))
```

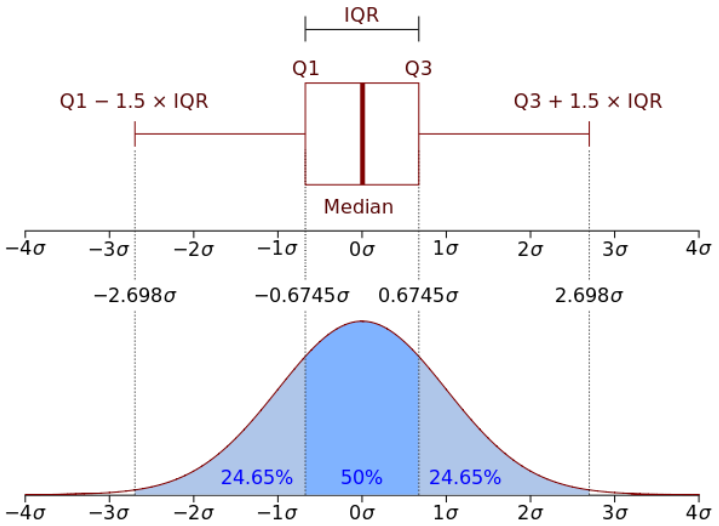






Box plot

- boxplot() : 데이터의 분포를 보여주는 상자그림
  - 자료로부터 얻어낸 통계량인 5가지 요약수치를 이용하여 그려짐
  - 5가지 요약 수치 : 최소값, 제 1사분위(Q 1) 제 2사분위(Q 2, median), 제 3사분위(Q 3), 최대값
  - 사분위범위(IQR, Inter Quartile Range): '제3사분위수 – 제1사분위수'로 계산
  - Q1−1.5 * IQR 보다 큰 데이터 중 가장 작은 값(lower whisker), Q3 +1.5 * IQR 보다 작은 데이터 중 가장 큰 값(upper whisker)을 각각 표시
  - outlier : lower whisker 보다 작은 데이터 또는 upper whisker 보다 큰 데이터로 그래프에 점으로 표시

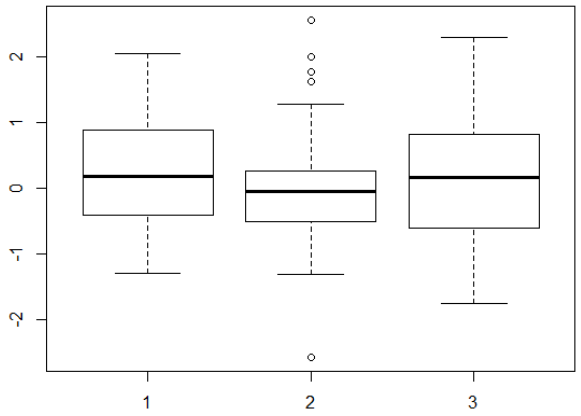


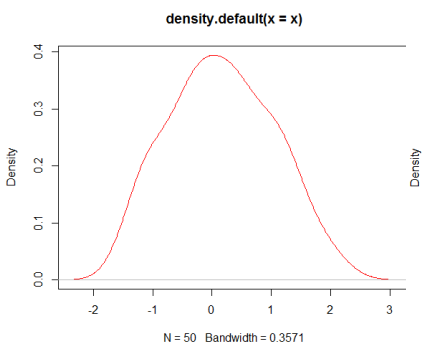
Percentiles and Quartiles

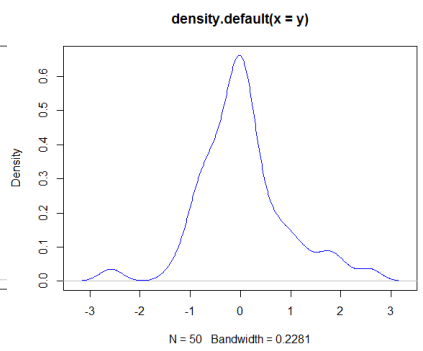
- Percentiles(백분위수)
  - N개의 데이터를 오름차순으로 나열하여 100등분한 각각의 절단점
  - 25^{th} percentile : 데이터의 25%에 해당되는 값
  - 75^{th} percentile : 데이터의 75%에 해당되는 값
- Quartiles(사분위수)
  - N개의 데이터를 오름차순으로 나열하여 4그룹으로 나눈값
  - 전체 100%를 4개의 균등한 부분으로 분할(25%, 50%, 75% 100%)
  - 제1 사분위수(하위 사분위수) Q1 : 누적도수 25%에 해당되는 값 , 25^{th} percentile
  - 제2 사분위수(중위 사분위수) Q2 : 누적도수 50%에 해당되는 값 , 중앙값
  - 제3 사분위수(또는 상위 사분위수) Q3 : 누적도수 25%에 해당되는 값 , 75^{th} percentile
  - 사분위수 범위 IQR = Q3-Q1.

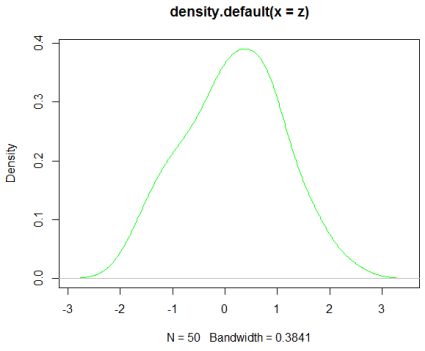
Box plot

```
x = rnorm(50)
y = rnorm(50)
z = rnorm(50)
>boxplot(x,y,z)
>summary(x)
Min. 1st Qu. Median Mean 3rd Qu. Max.
-2.5510 -0.9271 -0.2601 -0.2949 0.3124 1.9240
>summary(y)
Min. 1st Qu. Median Mean 3rd Qu. Max.
-2.209000 -0.387300 -0.022230 0.007806 0.404700 2.104000
>summary(z)
Min. 1st Qu. Median Mean 3rd Qu. Max.
-2.40400 -0.57190 0.03603 -0.05108 0.50510 2.28500
```





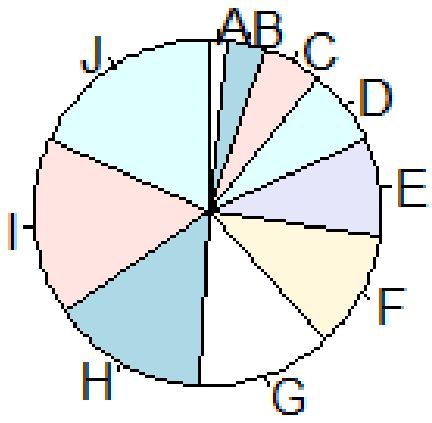




pie

- pie(): 데이터의 비율을 알아보기 위한 파이 그래프를 그리는 함수

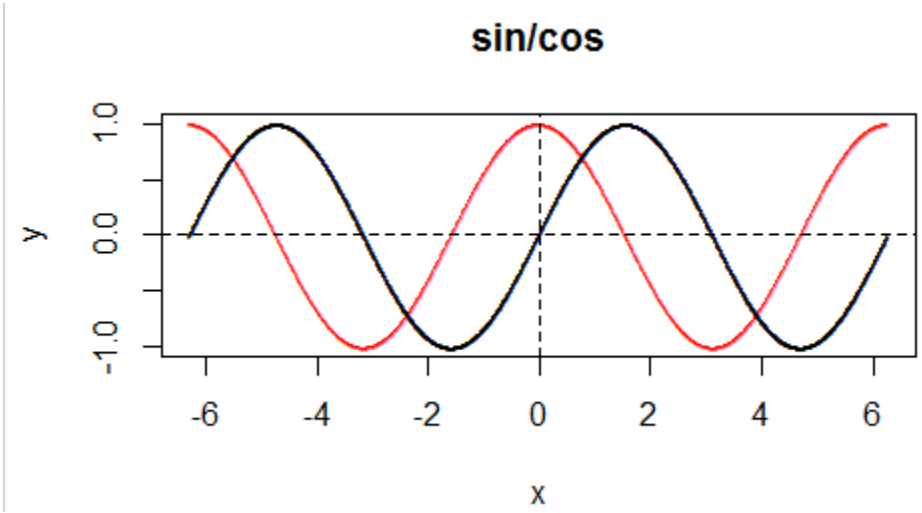
```
x = 1:10
pie(x, radius=1,  labels=LETTERS[x], clockwise=T, main="Pie graph(1:10)")
```



행렬데이터 그리기

- 행렬에 저장된 데이터 그리기
- matplot(), matlines(), matpoints() : plot(), lines(), points() 함수와 유사, 행렬(matrix) 형태로 주어진 데이터를 그래프에 그리기

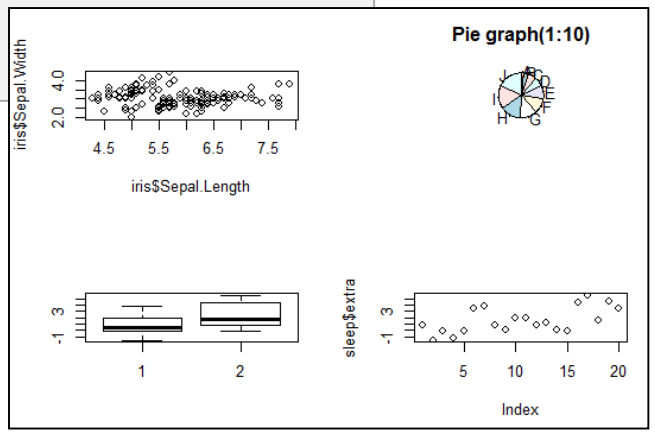
```
x =  seq (-2*pi , 2*pi , 0.01 )
y = matrix (c( cos (x), sin (x)), ncol =2)
matplot (x, y, col=c(" red ", " black "), cex =.2, main="sin/cos")
abline (h=0, v =0, lty=2)
```



그래프의 배열(mfrow)

- par()문에 mrow를 지정하여 한 창에 여러개의 그래프를 나열
- 형식: par(mfrow= c(nr, nc))
  - nr: 행의 수, nc: 열의 수
  - par() 문에 mfrow를 지정하면 이전에 저장된 par 설정이 반환되므로 par 설정으로 되돌리기 위해 저장하여 처리

```
oldp = par(mfrow=c(2,2)) #이전설정 저장
plot(iris$Sepal.Length , iris$Sepal.Width)
x = 1:10
pie(x, radius=1,  labels=LETTERS[x], clockwise=T, main="Pie
graph(1:10)")  #
data(sleep)
plot(sleep$group,sleep$extra) #factor기준으로 박스플롯
plot(sleep$extra) #factor기준으로 박스플롯
par(oldp)  #이전 설정 되돌리기
```



Identify

- Identify
  - 그래프상에서 특정 점을 클릭하면 클릭된 점과 가장 가까운 데이터를 그려준다

```
plot(iris$Sepal.Length , iris$Sepal.Width)
id=identify(iris$Sepal.Length , iris$Sepal.Width)
```