

제4강: R 그래픽 및 차트 기능

금융 통계 및 시계열 분석

TRADE INFORMATIX

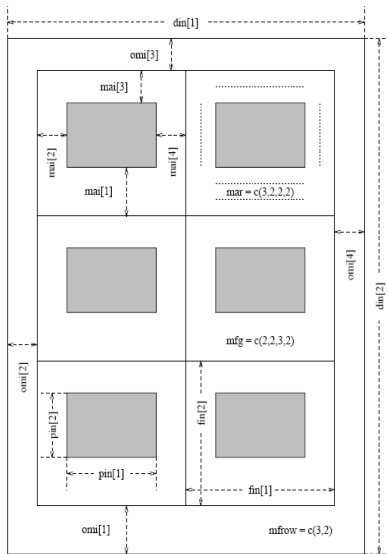
2014년 1월 17일

- 1 R 그래픽 관련 참고 자료
- 2 R 그래픽 구조
- 3 디바이스
- 4 그래픽 설정
- 5 차트 종류별 기능
 - Bar 차트
 - Pie 차트
 - Line 차트
 - Scatter Plot
 - Pair Plot
 - Box Plot
 - QQ Plot
- 6 R 그래픽 Annotation
- 7 R 그래픽 추가 설정
- 8 R 그래픽 레이아웃
- 9 R 그래픽 패키지
 - grid 패키지
 - lattice 패키지
 - ggplot2 패키지

- ❑ <https://www.stat.auckland.ac.nz/~paul/Talks/Rgraphics.pdf>
- ❑ <http://www.statmethods.net/advgraphs/>
- ❑ <http://rgraphics.limnology.wisc.edu/>
- ❑ <http://www.stat.tamu.edu/~jkim/Rfigurelayout.pdf>
- ❑ <http://www.stat.tamu.edu/~jkim/Rlinestyle.pdf>
- ❑ <http://www.stat.tamu.edu/~jkim/Rcolorstyle.pdf>

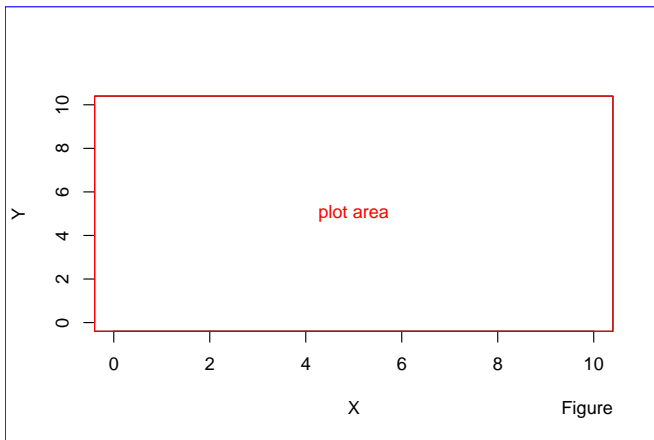
R 그래픽 구조

- device, figure, plot 의 3단계 구조
- figure는 그리드 형태로 여러개의 plot을 가질 수 있음



R 그래픽 구조 (계속)

```
> plot(0:10, 0:10, type="n", xlab="X", ylab="Y")  
> text(5, 5, "plot area", col="red")  
> box("plot", col="red")  
> mtext("Figure", SOUTH~-1, line=3, adj=1.0)  
> box("figure", col="blue")
```



□ 디바이스(Device)

- ▶ 그래픽 처리를 위한 드라이버
- ▶ pdf, postscript, xfig, bitmap, pictex, X11, svg, png, jpeg, bmp, tiff 등
- ▶ R은 유한개의 디바이스를 오픈하고 한번에 한개의 디바이스를 선택하여 사용

□ 관련 명령어

- ▶ `dev.cur()` : 현재 선택된 디바이스
- ▶ `dev.list()` : 현재 열려있는 디바이스
- ▶ `dev.set()` : 디바이스 선택
- ▶ `dev.off()` : 현재 선택된 디바이스를 닫음
- ▶ `dev.next()` : 현재 열려있는 디바이스 중 다음 디바이스
- ▶ `dev.prev()` : 현재 열려있는 디바이스 중 이전 디바이스

1. 저장하려는 그래픽 포맷에 대한 디바이스와 파일을 오픈

- ❑ `bmp(filename, width, height, units, pointsize, bg, antialias)`
- ❑ `jpeg(filename, width, height, units, pointsize, bg, antialias, quality)`
- ❑ `png(filename, width, height, units, pointsize, bg, antialias)`
- ❑ `tiff(filename, width, height, units, pointsize, bg, antialias, compression)`

2. 차트 작업

3. `dev.off()`로 선택된 디바이스 닫음

그래픽 파라미터 설정

- ❑ 그래픽 파라미터 설정 방법은 두가지가 있다.
 1. `plot()`, `lines()` 등의 플롯 명령의 인수로 넣는 방법
 2. `par()` 명령으로 외부에서 설정하는 방법
- ❑ `par()` 설정은 이후의 모든 그래픽에 적용되므로 기존 설정을 저장할 필요가 있는 경우 오브젝트에 저장
- ❑ 그래픽이 그려진 다음에는 `par()` 명령이 효과 없음

```
> old.par <- par()
> par(old.par)
```


par() 명령으로 설정 가능한 파라미터 1

이름	의미	이름	의미
din	device 크기 (inch)	omi	device 마진 (inch)
fin	figure 크기 (inch)	oma	device 마진 (line)
pin	picture 크기 (inch)	omd	device 마진 (device fraction)
mgp	축 마진 (line))	mai	figure 마진 (inch)
		mar	figure 마진 (line)
mfrow	figure 그리드 설정	mfg	figure 그리드내 위치 지정
col	플롯 색상	cex	글자 및 심볼 확대배율
col.axis	축 색상	cex.axis	축 확대배율
col.lab	라벨 색상	cex.lab	라벨 확대배율
col.main	제목 색상	cex.main	제목 확대배율
col.sub	부제목 색상	cex.sub	부제목 확대배율
fg	foreground 색상	crt	글자 회전 각도
bg	background 색상	cra	글자크기 (픽셀단위)

par() 명령으로 설정 가능한 파라미터 2

이름	의미
new	TRUE면 plot명령이 나와도 앞서 그림을 지우지 않음
pty	플롯 영역 스타일. 's'(정사각형) 'm'(최대 크기)
lty	line type. "solid", "dashed", "dotted", "dotdash", "longdash"
lwd	line width. 디폴트 1
lend	line end style. "round", "butt", "square"
ljoin	line join style. "round", "mitre", "bevel"
pch	심볼 정의 (0:18 정수)
font	font
font.axis	축 font
font.lab	라벨 font
font.main	제목 font
font.sub	부제목 font
las	축 라벨 스타일. 0(축에 평행), 1(수평), 2(축에 수직), 3(수직)
xaxt	x 축 유형. 's' 표준, 'n' 축없음
xlog	x 축 로그 스케일 사용 여부 (TRUE면 로그)

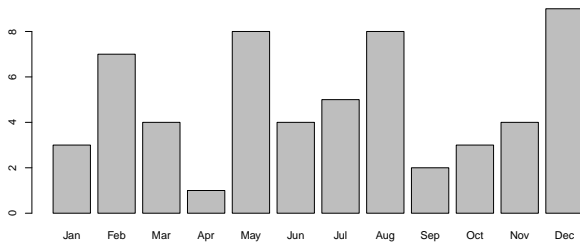
□ R 에서 기본적으로 지원하는 차트/플롯 종류

- ▶ Bar chart
- ▶ Pie chart
- ▶ Line chart
- ▶ Scatter plot
- ▶ Pairs plot
- ▶ Box plot

Bar 차트

□ `barplot(height, names)`

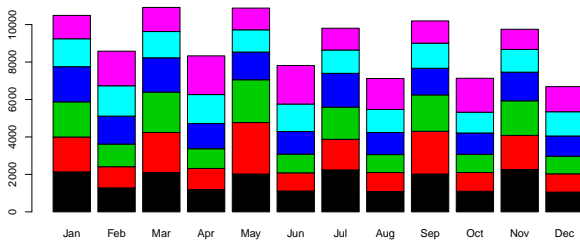
```
> category <- month.abb  
> value <- c(3,7,4,1,8,4,5,8,2,3,4,9)  
> barplot(value, names=category)
```



Stacked Bar 차트

□ data가 matrix이면 자동으로 stacked bar 차트

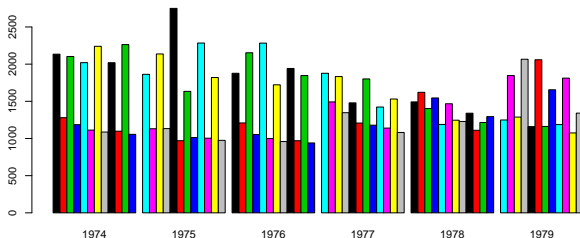
```
> data <- matrix(as.vector(mdeaths), ncol=12)  
> barplot(data, names=month.abb, col=palette("default"))
```



Grouped Bar 차트

□ plot 에서 beside=TRUE 설정

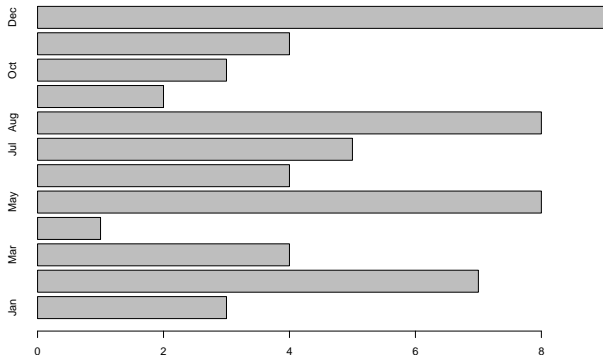
```
> data <- matrix(as.vector(mdeaths), nrow=12, byrow=TRUE)
> barplot(data, names=start(mdeaths)[1]:end(mdeaths)[1], beside=TRUE, col=palette("default"))
```



Horizontal Bar 차트

❑ horiz=TRUE

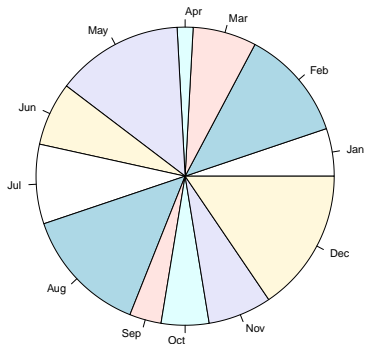
```
> category <- month.abb  
> value <- c(3,7,4,1,8,4,5,8,2,3,4,9)  
> barplot(value, names=category, horiz=TRUE)
```



Pie 차트

❏ `pie(data, labels, radius, clockwise, init.angle)`

```
> category <- month.abb  
> value <- c(3,7,4,1,8,4,5,8,2,3,4,9)  
> pie(value, labels=category)
```



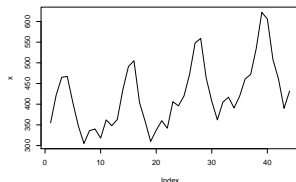
□ `plot(x, y, type)` 명령

□ `type` 인수는 라인 유형 결정

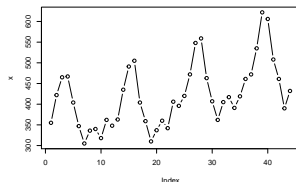
- ▶ `type='p'` : 포인트만 그림
- ▶ `type='l'` : 라인만 그림
- ▶ `type='b'` : 포인트와 라인 모두 (both) 그림 (겹치지 않음)
- ▶ `type='o'` : 포인트와 라인을 겹쳐서 그림 (overplotted)
- ▶ `type='c'` : both 유형에서 포인트 제외
- ▶ `type='n'` : 실제 라인/포인트는 그리지 않음
- ▶ `type='s'` : 계단형 스텝 (zero-order-hold)
- ▶ `type='S'` : 계단형 스텝 (backward zero-order-hold)
- ▶ `type='h'` : 히스토그램 스타일

Line 차트 유형 1

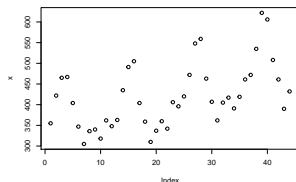
```
> x <- AirPassengers[-(1:100)]  
> plot(x, type='l', lty=1)
```



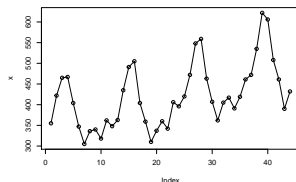
```
> x <- AirPassengers[-(1:100)]  
> plot(x, type='b', lty=1)
```



```
> x <- AirPassengers[-(1:100)]  
> plot(x, type='p', lty=1)
```

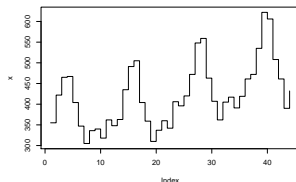


```
> x <- AirPassengers[-(1:100)]  
> plot(x, type='o', lty=1)
```

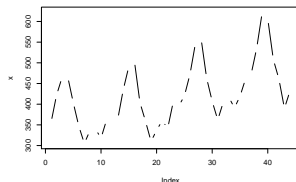


Line 차트 유형 2

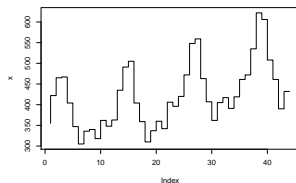
```
> x <- AirPassengers[-(1:100)]  
> plot(x, type='s', lty=1)
```



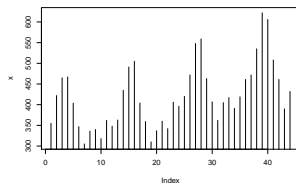
```
> x <- AirPassengers[-(1:100)]  
> plot(x, type='c', lty=1)
```



```
> x <- AirPassengers[-(1:100)]  
> plot(x, type='S', lty=1)
```



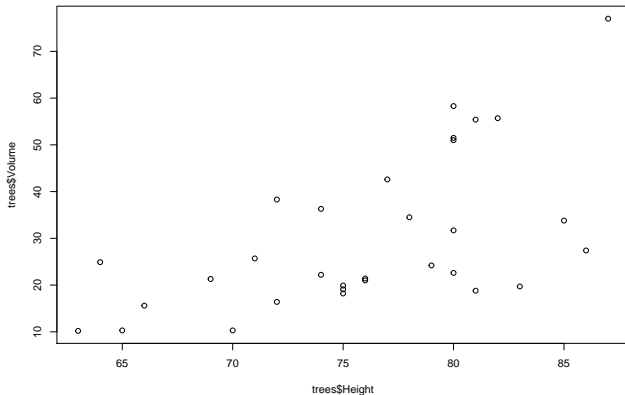
```
> x <- AirPassengers[-(1:100)]  
> plot(x, type='h', lty=1)
```



Scatter Plot

- ❑ 두 개의 시계열이 있는 경우 상관도를 보기 위한 차트
- ❑ `plot(x, y, type='p')`

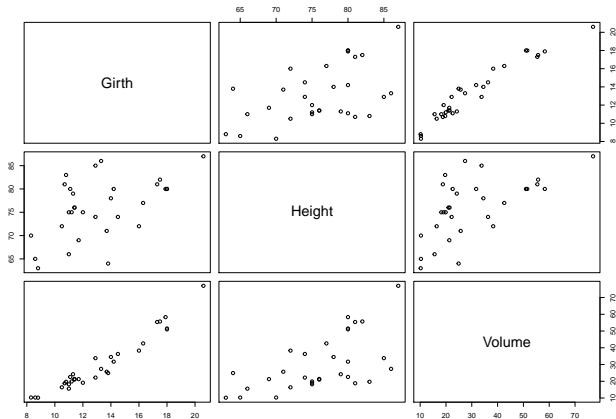
```
> plot(trees$Height, trees$Volume)
```



Pair Plot

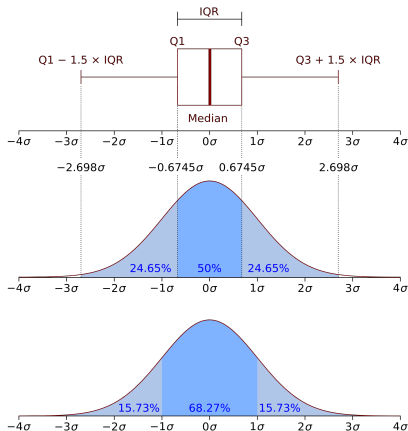
- ❑ 두 개 이상의 시계열이 있는 경우 상관도를 보기 위한 차트
- ❑ `pairs(data.frame)`

```
> plot(trees)
```



Box Plot

- Box plot 또는 Box-Whisker plot
 - ▶ center line : 중간값 (median)
 - ▶ box : 1st quartile, 3rd quartile
 - ▶ whisker : ± 1.5 IQR (Inter-quartile range, $IQR = Q3 - Q1$)
 - ▶ circle : whisker 밖의 아웃라이어 (outlier)

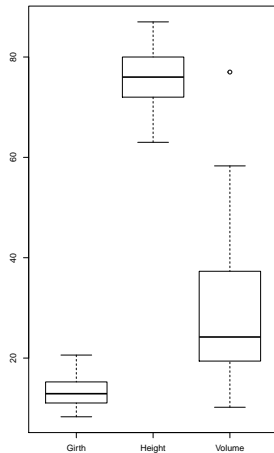


Box Plot 예

```
> trees
```

	Girth	Height	Volume
1	8.3	70	10.3
2	8.6	65	10.3
3	8.8	63	10.2
4	10.5	72	16.4
5	10.7	81	18.8
6	10.8	83	19.7
7	11.0	66	15.6
8	11.0	75	18.2
9	11.1	80	22.6
10	11.2	75	19.9
11	11.3	79	24.2
12	11.4	76	21.0
13	11.4	76	21.4
14	11.7	69	21.3
15	12.0	75	19.1
16	12.9	74	22.2
17	12.9	85	33.8
18	13.3	86	27.4
19	13.7	71	25.7
20	13.8	64	24.9
21	14.0	78	34.5
22	14.2	80	31.7
23	14.5	74	36.3
24	16.0	72	38.3
25	16.3	77	42.6
26	17.3	81	55.4
27	17.5	82	55.7
28	17.9	80	58.3
29	18.0	80	51.5
30	18.0	80	51.0
31	20.6	87	77.0

```
> boxplot(trees)
```



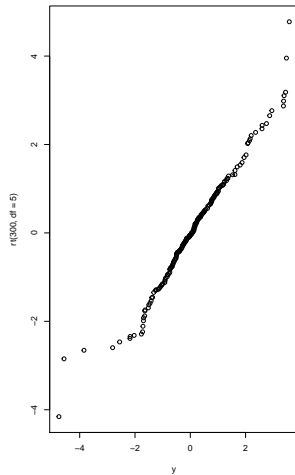
- ❑ 정규분포 혹은 특정 샘플집합과의 Quantile-Quantile plot
- ❑ 동일한 quantile 값을 가지는 값을 양쪽 집합에서 구해서 x,y 값으로 사용
- ❑ normality/skewness/kurtosis 등을 한 눈에 파악
 - ▶ `qqnorm(y)` : 정규 분포와 qq 비교
 - ▶ `qqplot(x, y)` : 일반 샘플집합과 qq 비교
 - ▶ `qqline(y)` : QQ plot 기준선 추가

QQ Plot 예

```
> y <- rt(200, df = 5)  
> qqnorm(y); qqline(y, col = 2)
```

or,

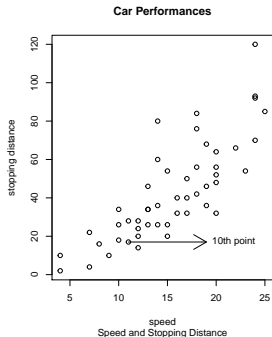
```
> qqplot(y, rt(300, df = 5))
```



- ❑ annotation 명령어는 추가적인 선/점/문자 등을 그리는 명령
- ❑ 여러 유형의 차트가 합쳐진 복합차트, 여러개의 축을 가지는 차트, 혹은 화살표나 설명을 가진 차트 등을 생성할 때 사용
- ❑ 내부적으로 `plot.new()`를 호출하지 않으므로 사용하기 전에 이미 `plot()`, `bar()` 등 `plot.new()`를 호출하는 명령이 실행되어 플롯 영역이 존재해야 한다.
 - ▶ `lines` : 라인 플롯
 - ▶ `points` : 점 플롯
 - ▶ `symbol` : 특수문자
 - ▶ `text` : 문자
 - ▶ `abline` : 직선
 - ▶ `arrows` : 화살표
 - ▶ `rect` : 사각형
 - ▶ `polygon` : 다각형
 - ▶ `qqline` : QQ plot 기준선

R 그래픽 Annotation 예

```
> plot(cars, main="Car Performances",  
+       sub="Speed and Stopping Distance",  
+       xlab="speed", ylab="stopping distance")  
> arrows(cars$speed[10], cars$dist[10],  
+        cars$speed[10] + 8, cars$dist[10])  
> text(cars$speed[10] + 8, cars$dist[10],  
+       labels="10th point", pos=4)
```



□ plot 명령 인수

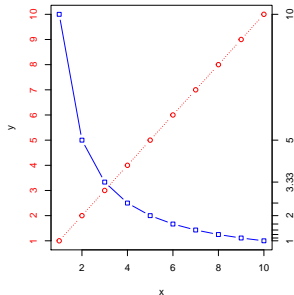
- ▶ `main` : 타이틀 캡션
- ▶ `sub` : 서브 타이틀 캡션
- ▶ `xlab` : x 축 라벨 캡션
- ▶ `ylab` : y 축 라벨 캡션

□ 외부 명령어

- ▶ `title` : 타이틀
- ▶ `legend` : 레전드
- ▶ `box` : 플롯 경계 상자
- ▶ `axes` : 축
- ▶ `mtext` : 마진 문자 (플롯 경계 바깥)

- ❑ 축 범위는 plot 명령의 인수로 설정
 - ▶ `xlim`: x 축 최대/최소값 범위 설정
 - ▶ `ylim`: y 축 최대/최소값 범위 설정
 - ▶ `axis(side, at, labels, tick)`
- ❑ `axis(side, at, labels, tick)`: 축 생성 및 틱 설정
 - ▶ `side`: 축 위치 (1=아래, 2=왼쪽, 3=위 and 4=오른쪽)
 - ▶ `at`: 틱 위치
 - ▶ `labels`: 틱 라벨

```
> x <- c(1:10)
> y <- x
> z <- 10/x
> plot(x, y, type="b", yaxt="n",
+       pch=21, col="red", lty=3)
> lines(x, z, type="b", pch=22, col="blue")
> axis(2, at=x, labels=x, col.axis="red")
> axis(4, at=z, labels=round(z,digits=2))
```

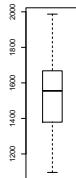
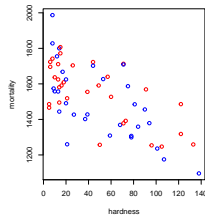
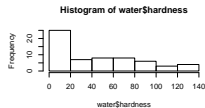


Multiple Plot

- ❑ 한 figure에 여러 개의 plot을 grid 형태로 넣는 multiple plot을 그리려면 `layout()` 명령어 사용
- ❑ `layout(mat, widths, heights)`
 - ▶ `mat` : 그리드 설정 명령어
 - $n \times m$ 행렬을 넣으면 $n \times m$ 그리드 설정
 - 행렬 값은 0부터 최대 nm 의 정수
 - 빈공간에는 0을 넣고 나머지는 1부터 순서대로 설정
 - 추후 `plot.new()`를 실행하거나 이를 내부적으로 실행하는 명령을 사용하면 1로 설정된 영역부터 차례대로 그려짐
 - ▶ `widths` : 폭 벡터. 그리드 셀의 상대적 크기 조정
 - ▶ `heights` : 높이 벡터. 그리드 셀의 상대적 크기 조정

Multiple Plot의 예

```
> library("HSAUR2")
> layout(matrix(c(2, 0, 1, 3), 2, 2, byrow=T),
+         c(2, 1), c(1, 2))
> plot(mortality ~ hardness, data=water,
+       col=c("red", "blue"))
> hist(water$hardness)
> boxplot(water$mortality)
```

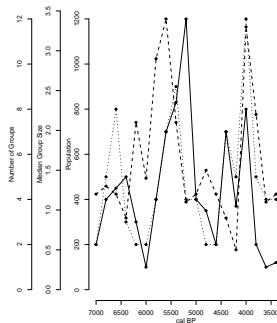


Multiple Axes

- ❑ 한 plot에 다른 스케일을 가지는 여러가지 자료를 그리려면 복수개의 축 (multiple axes)가 필요
- ❑ 다양한 plot 설명 명령어 및 annotation 명령어 필요
- ❑ 생성 방법
 1. `par(mar)` 명령으로 마진을 넓게 설정
 2. 축 스케일을 설정하기 위한 `range` 계산
 3. `axes=F` 인수로 축과 박스가 없는 플롯을 그림. 필요하면 `xlab=""` 등으로 라벨 생성을 막음 이 때 `xlim`, `ylim` 등의 인수로 축 스케일을 사용자가 고정
 4. 필요할 경우 추가 플롯 그림. 이 때 `lines()`과 같은 annotation 명령어는 그냥 사용 가능하지만 `plot()`과 같이 새로운 플롯 영역을 생성하는 명령어를 사용할 때는 `par(new=TRUE)` 명령으로 홀드시킨다. 마찬가지로 `xlim`, `ylim` 등의 인수로 축 스케일을 사용자가 고정
 5. `axes()` 명령으로 필요한 축을 생성
 6. `box()` 명령으로 필요한 경계박스 생성
 7. `mtext()` 명령으로 축 라벨을 생성

Multiple Axes의 예

```
> time <- seq(7000,3400,-200)
> pop <- c(200,400,450,500,300,100,400,700,830,1200,
+         400,350,200,700,370,800,200,100,120)
> grp <- c(2,5,8,3,2,2,4,7,9,4,4,2,2,7,5,12,5,4,4)
> med <- c(1.2,1.3,1.2,0.9,2.1,1.4,2.9,3.4,2.1,1.1,
+         1.2,1.5,1.2,0.9,0.5,3.3,2.2,1.1,1.2)
> par(mar=c(5, 12, 4, 4) + 0.1)
> plot(time, pop,
+      type="l", col="black",
+      xlim=c(7000,3400), ylim=c(0,max(pop)),
+      axes=F, xlab="", ylab="", main="")
> points(time, pop, pch=20, col="black")
> axis(2, ylim=c(0,max(pop)), col="black", lwd=2)
> mtext(2, text="Population", line=2)
> par(new=T)
> plot(time, med,
+      type="l", lty=2, lwd=2,
+      xlim=c(7000,3400), ylim=c(0,max(med)),
+      axes=F, xlab="", ylab="", main="")
> points(time, med, pch=20)
> axis(2, ylim=c(0,max(med)), lwd=2, line=3.5)
> mtext(2, text="Median Group Size", line=5.5)
> par(new=T)
> plot(time, grp,
+      type="l", lty=3, lwd=2,
+      xlim=c(7000,3400), ylim=c(0,max(grp)),
+      axes=F, xlab="", ylab="", main="")
> points(time, grp, pch=20)
> axis(2, ylim=c(0,max(grp)), lwd=2, line=7)
> mtext(2, text="Number of Groups", line=9)
> axis(1,pretty(range(time),10))
> mtext("cal BP",side=1,col="black",line=2)
```



- ❑ <http://cran.r-project.org/web/views/Graphics.html> 참조
- ❑ 일반 그래픽 패키지
 - ▶ grid 패키지 : base R의 그래픽 기능 강화
 - ▶ lattice 패키지 : S/plus 의 Trellis 패키지 R 구현
 - ▶ ggplot2 패키지 : Leland Wilkinson 문법 구현
- ❑ interactive 그래픽 (rstudio 서버 버전에서 사용 불가)
 - ▶ rggobi
 - ▶ iplots
- ❑ 특화된 그래픽 패키지
 - ▶ orderbook : limit order book
 - ▶ rgl : openGL 3d graphics
 - ▶ visreg : regression models
 - ▶ vcd, vcdExtra : categorical data
 - ▶ rCharts : javascript rendering

- ❑ R base의 low-level 그래픽 기능을 재구성
- ❑ viewport와 object 기반
- ❑ viewport
 - ▶ R base figure에 해당하는 공간
- ❑ object
 - ▶ text, line, circle, rectangle 등 그래픽 구성요소

lattice 패키지: 지원하는 플롯 유형

이름	의미
<code>histogram()</code>	히스토그램
<code>densityplot()</code>	커널 덴서티 플롯
<code>qqmath()</code>	theoretical qq plot
<code>qq()</code>	qq plot
<code>stripplot()</code>	strip chart
<code>bwplot()</code>	comparative box-whisker plot
<code>dotplot()</code>	cleveland dot plot
<code>barchart()</code>	bar plot
<code>xyplot()</code>	scatter plot
<code>splom()</code>	scatter plot matrix
<code>contourplot()</code>	contour plot of surface
<code>levelplot()</code>	false color level plot
<code>wireframe()</code>	3d perspective plot
<code>cloud()</code>	3d scatter plot
<code>parallel()</code>	parallel coordinate plot

lattice 패키지 : Factor-conditional grid plot

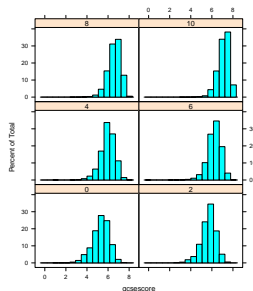
```
> data(Chem97, package="mlmRev")
> head(Chem97, n=10)
```

	lea	school	student	score	gender	age	gcscscore	gcsecnt
1	1	1	1	4	F	3	6.625	0.33931571
2	1	1	2	10	F	-3	7.625	1.33931571
3	1	1	3	10	F	-4	7.250	0.96431571
4	1	1	4	10	F	-2	7.500	1.21431571
5	1	1	5	8	F	-1	6.444	0.15831571
6	1	1	6	10	F	4	7.750	1.46431571
7	1	1	7	6	F	1	6.750	0.46431571
8	1	1	8	8	F	4	6.909	0.62331571
9	1	1	9	4	F	3	6.375	0.08931571
10	1	1	10	10	F	0	7.750	1.46431571

```
> xtabs(~score, data=Chem97)
```

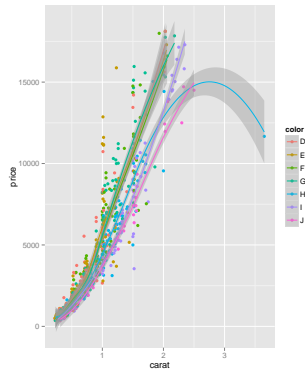
score	0	2	4	6	8	10
	3688	3627	4619	5739	6668	6681

```
> data(Chem97, package="mlmRev")
> library("lattice")
> histogram(~gcscscore|factor(score), data=Chem97)
```



ggplot2 패키지: 예 1

```
> library("ggplot2")
> set.seed(1410)
> dsmall <- diamonds[sample(nrow(diamonds), 1000), ]
> p <- ggplot(dsmall, aes(carat, price, color=color)) +
+   geom_point() + geom_smooth()
> print(p)
```



ggplot2 패키지: 예 2

