

사용자 정의 그래프 함수 만들기

KRUG 3차 Meet up

(2012-03-22)

유충현

Agenda

● R graphics 개요

- graphics devices
- graphics 함수
- plot region
- chart 생성 순서

● 사용자 정의 그래프

- simple chart

● 사용자 정의 그래프 함수

- lineChart
- scatter and boxplot
- pairbar plot

● 타 패키지 확장하기

- RGoogleMaps extension

● Q&A

R graphics 개요

● graphics device

- windows graphics devices
 - windows, X11, win.graph
 - win.metafile, win.print
- graphics file devices
 - bmp, jpeg, png, tiff
- other graphics file devices
 - pdf, postscript

● graphics device의 확장

- Cairo package : R + cairo graphics
- rgl package : R + openGL

● 유용한 graphics packages

- grid package
- lattice package, ggplot2 package

● low-level graphics 함수

- 캔버스에 그래프를 그리는 도구로서의 기능
 - points : 점을 그린다.
 - lines, segments : 선을 그린다.
 - rect, polygon : 면을 그린다.
- 글자를 그리는 도구로서의 기능
 - text
- 그래프를 꾸미는 도구로서의 기능
 - box, axis, abline, arrows, title, legend
- 이미 생성된 graphics에 덧그림

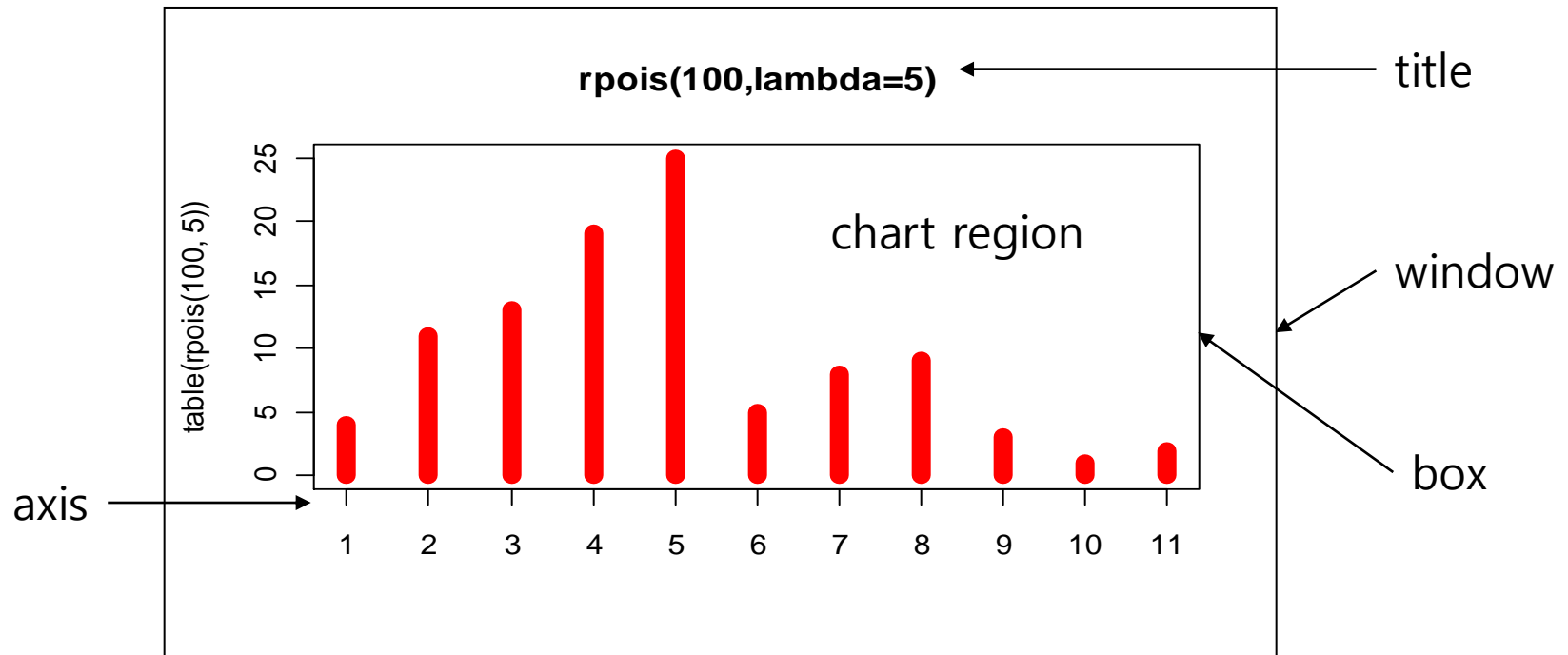
● high-level graphics 함수

- plot : generic 함수
- hist, barplot, boxplot, ...
- 함수 호출 시마다 새로운 graphics이 생성됨

plot region

R graphics 개요

● plot region의 구조



● plot region의 생성 함수

- window : plot.window 함수
- box : box 함수
- axis : Axis 함수
- title : title 함수
- chart region : low-level graphics 함수

chart 생성 순서

R graphics 개요

● 1st. 새로운 plot을 생성

- > plot.new()
- global graphics parameter(using **par**)를 가져온 후,
- **margins**을 포함한 빈 graphics window를 만든다.
- graphics window 모양



● 2nd. 캔버스의 높이 및 폭 정의 및 종횡비 지정

- `> plot.window(xlim, ylim, log = "x", asp=1)`
- graphics를 그리기 위한 **axis scale**
- `xlim, ylim, log` (축에 log 취함), `asp` (가로, 세로 종횡 비)

● 3rd. chart 그리기

- low-level graphics 함수를 이용해서 chart 그림
- 점, 선, 면을 그리는 함수 이용

● 4th. chart 주변 꾸미기

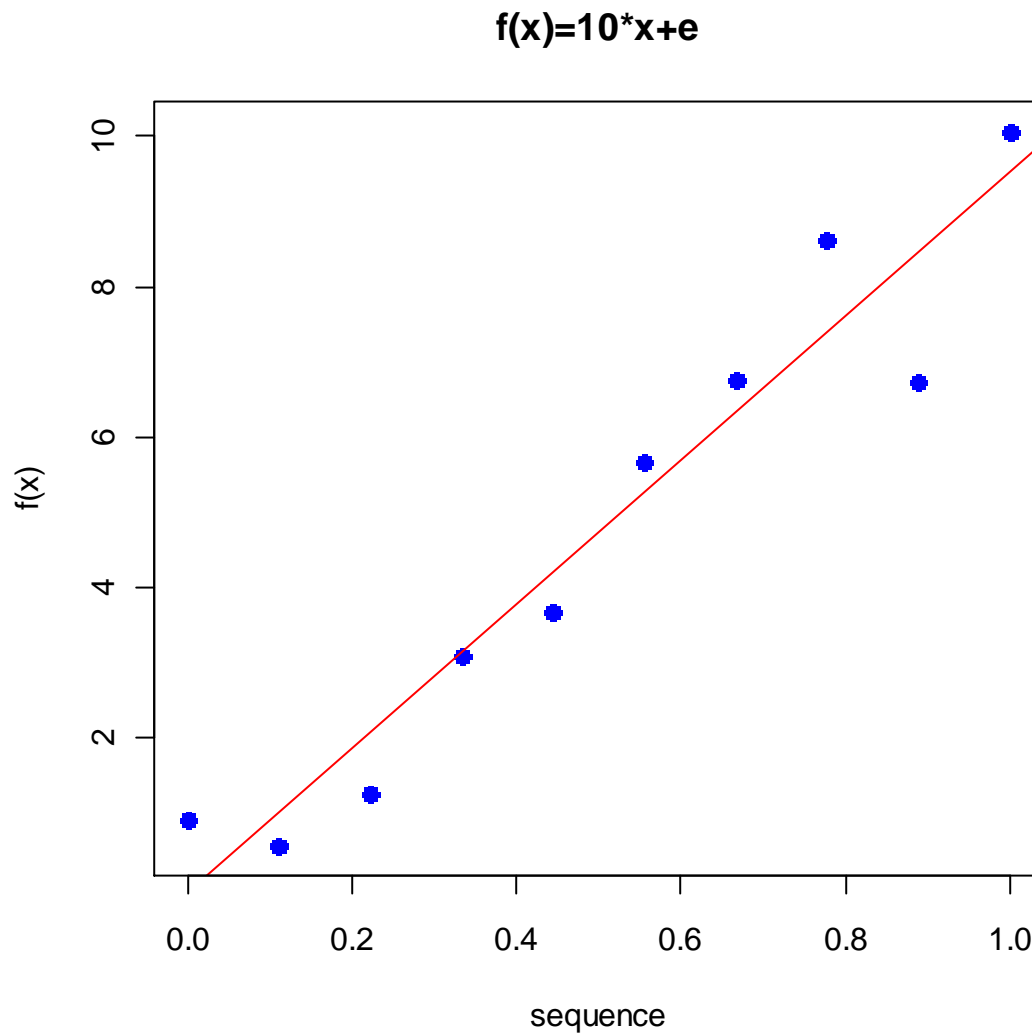
- `> axis(1, 1:4, LETTERS[1:4])`
 - x-축, y-축 등 각 축을 꾸밈
- `> title(main="The histogram of score")`
 - 제목, 부제목, x-축 이름, y-축 이름 등 chart의 주석을 꾸밈
- `> box()`
 - chart 영역을 사각형으로 테두리를 그림

사용자 정의 그래프

simple chart

사용자 정의 그래프

● regression chart



● regression chart script

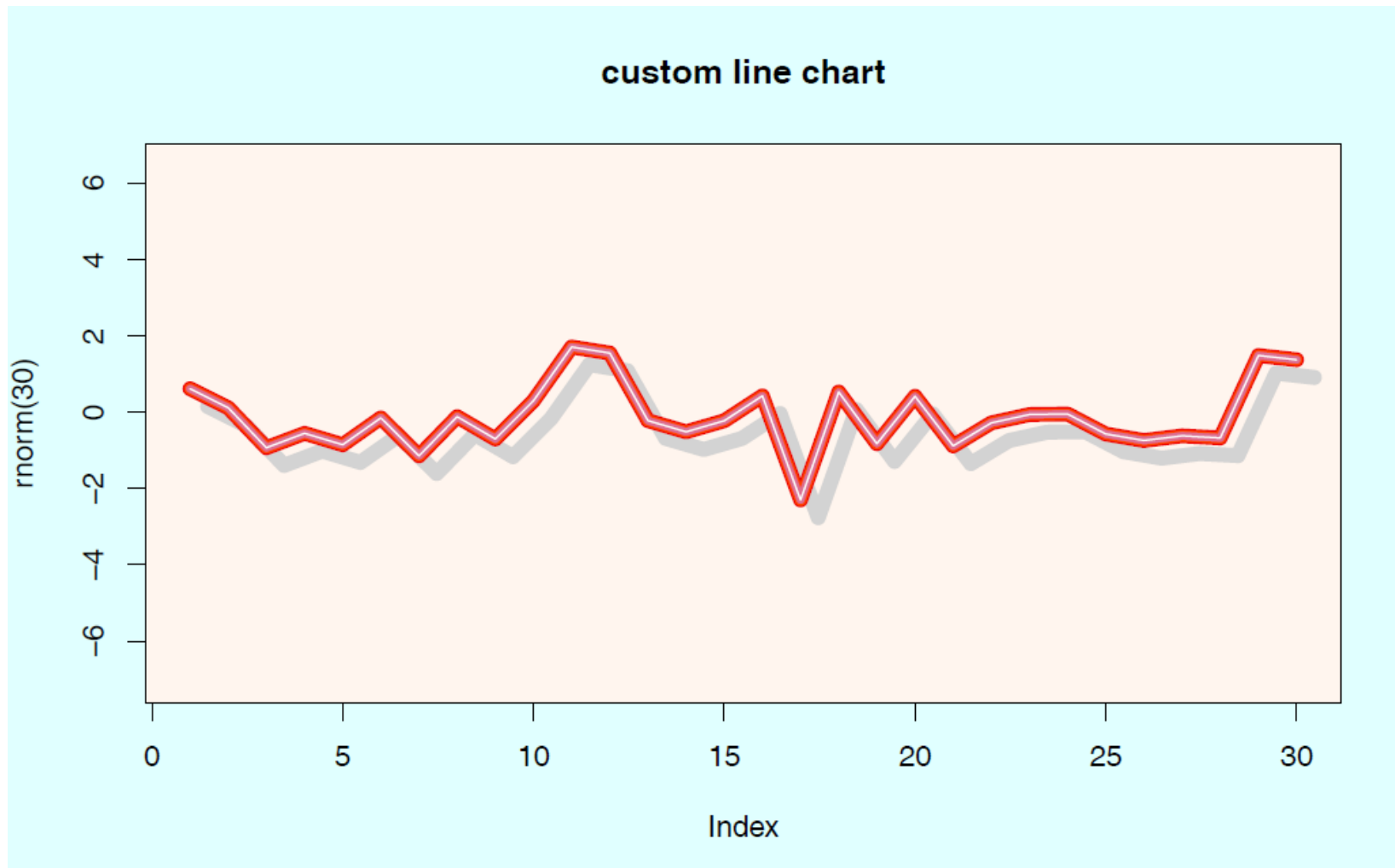
```
x <- seq(0, 1, length=10)
y <- 10 * x + rnorm(length(x))
plot.new()
plot.window(xlim = range(x), ylim = range(y))
points(x, y, col="blue", cex=1.2, pch=16)
abline(lm(y~x), col="red")
axis(1)
axis(2)
title(main = "f(x)=10*x+e")
title(ylab = "f(x)")
title(xlab = "sequence")
box()
```

사용자 정의 그래프 함수

lineChart

사용자 정의 그래프 함수

● custom line chart



lineChart

사용자 정의 그래프 함수

● custom line chart 함수

● 함수원형

```
lineChart <- function(x, y=NULL, shadow=F, axes=T, main = NULL,  
  sub = NULL, xlab = NULL, ylab = NULL, log = "", col=1,  
  xlim = NULL, ylim = NULL, bg = "gray", fg="gray95", ...)
```

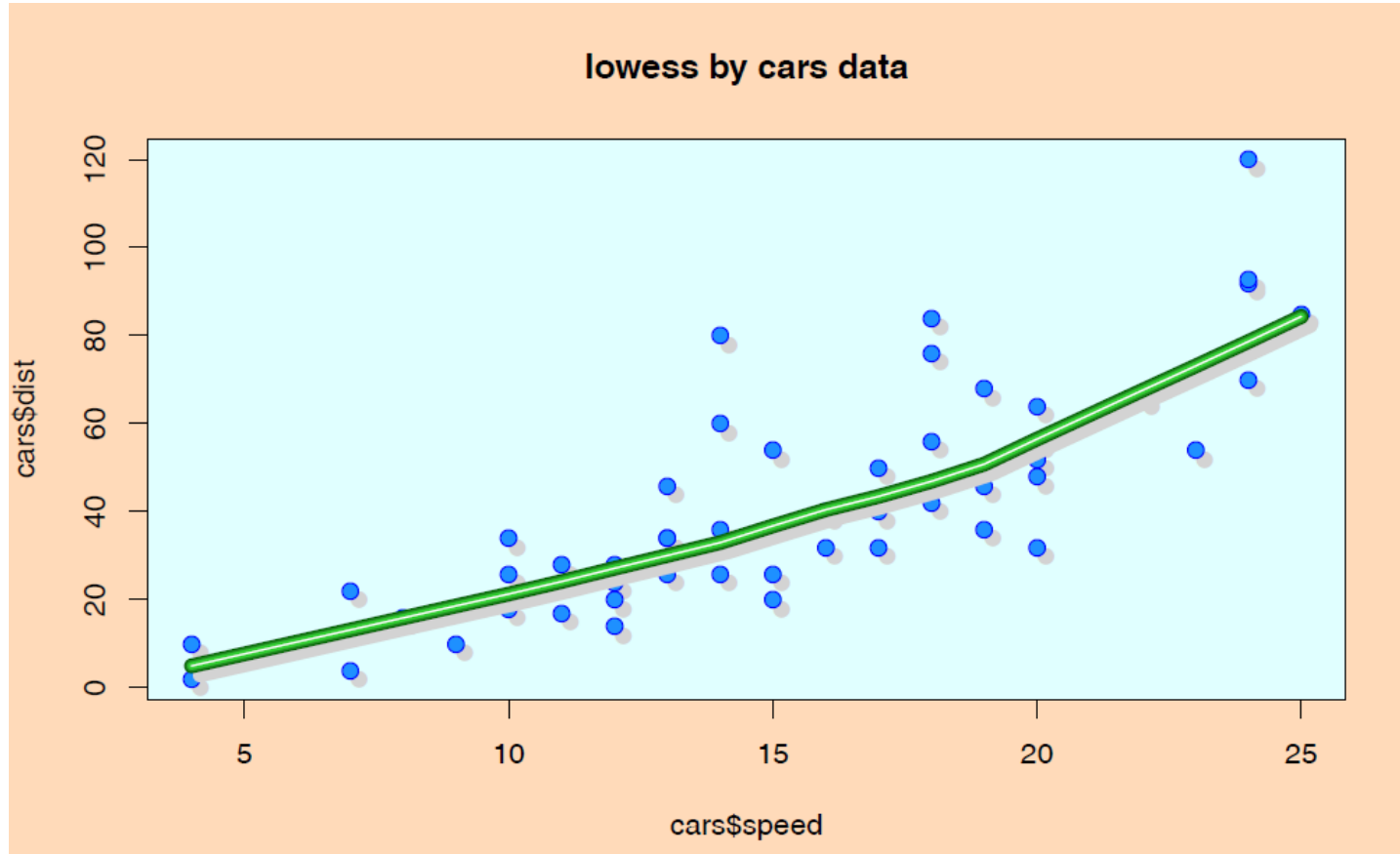
● 호출방법

```
> x <- 1:10  
> y <- rnorm(10)  
> z <- rnorm(10)  
> lineChart(x, z, col=3)  
> lineChart(y, shadow=T, main="User define function",  
+ xlab="sequence", ylab="random number")  
> lineChart(rnorm(30), shadow=T, main="custom line chart",  
+ fg="seashell", bg="lightcyan")
```

lineChart

사용자 정의 그래프 함수

● custom line chart

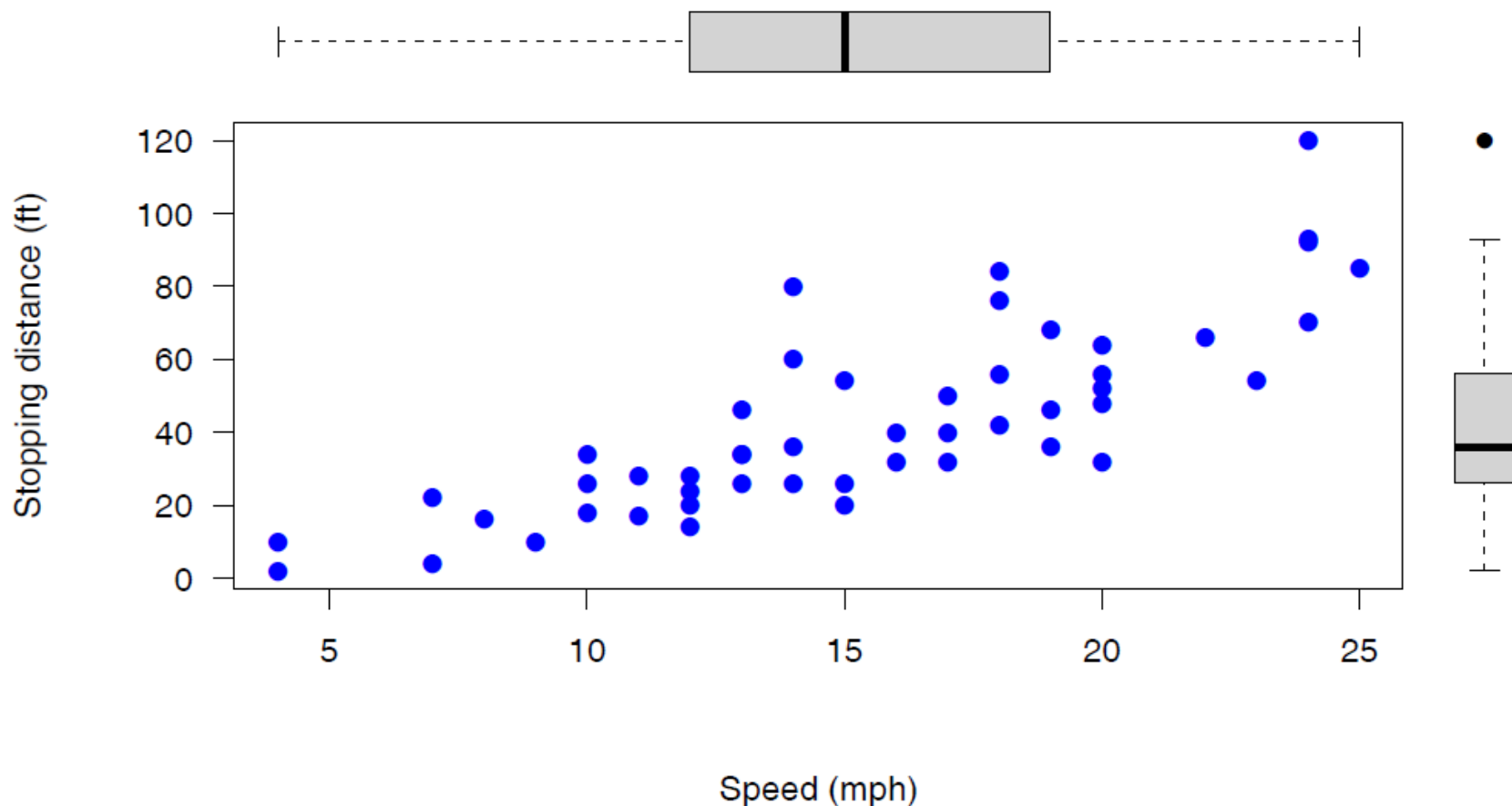


```
> lineChart(cars$speed, cars$dist, col=3, lowess=T, shadow=T,  
+ main="lowess by cars data", fg="lightcyan", bg="peachpuff")
```

scatter and boxplot

사용자 정의 그래프 함수

Speed and Stopping Distances of Cars



scatter and boxplot

사용자 정의 그래프 함수

● layout 함수를 이용한 plot region의 분할

- > par(mfrow=c(2, 2))

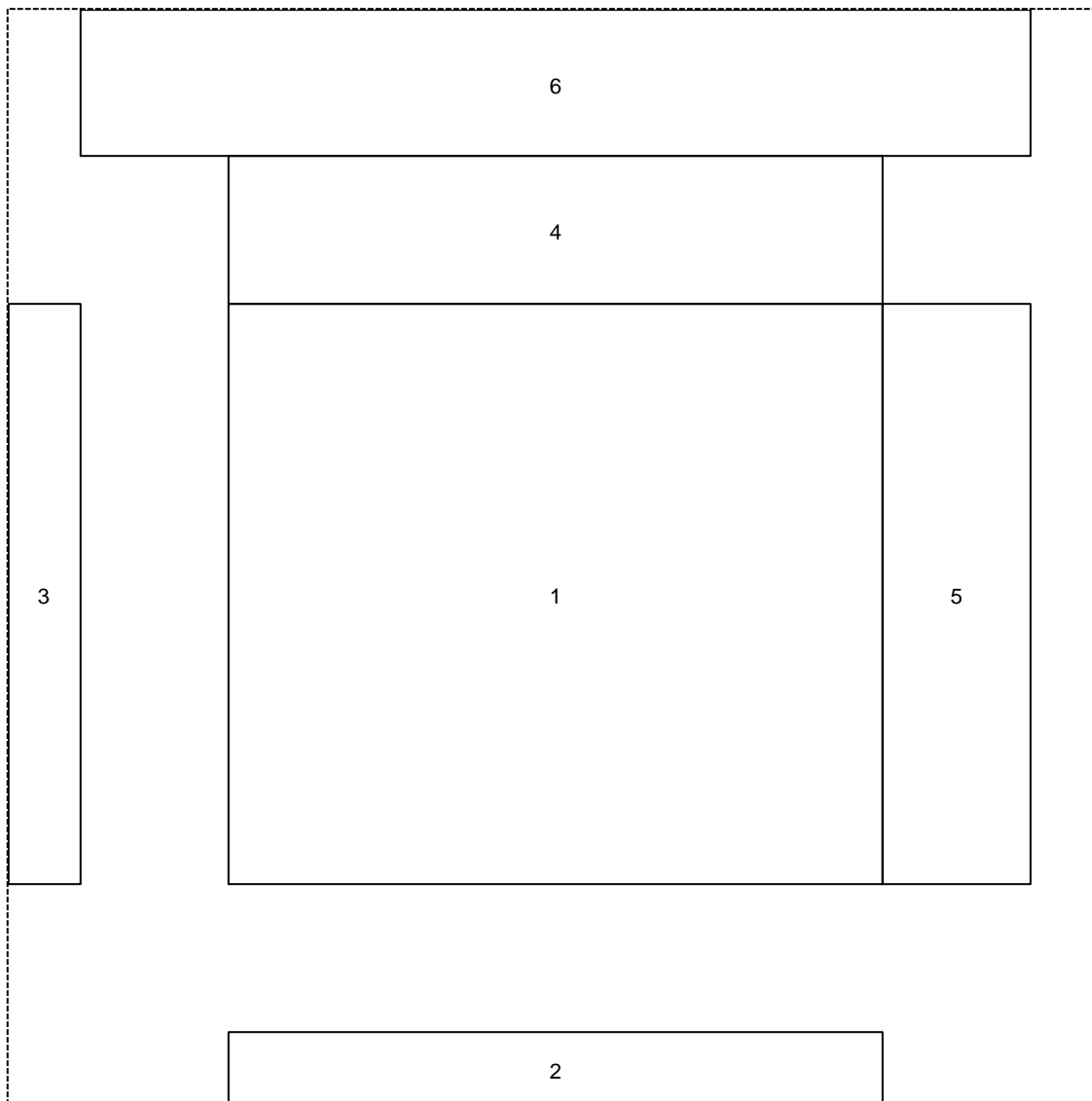
- par 함수를 이용한 plot region의 분할

- layout 함수를 이용한 plot region의 분할

```
> layout(rbind(c(0,6,6,6,0),  
+             c(0,0,4,0,0),  
+             c(3,0,1,5,0),  
+             c(0,0,0,0,0),  
+             c(0,0,2,0,0)),  
+       height = c(lcm(2), lcm(2), 1, lcm(2), lcm(1)),  
+       width  = c(lcm(1), lcm(2), 1, lcm(2), lcm(1)))  
> layout.show(6)  
> box("outer", lty = "dotted")
```

scatter and boxplot

사용자 정의 그래프 함수



scatter and boxplot

사용자 정의 그래프 함수

● boxscatter 함수

● 함수원형

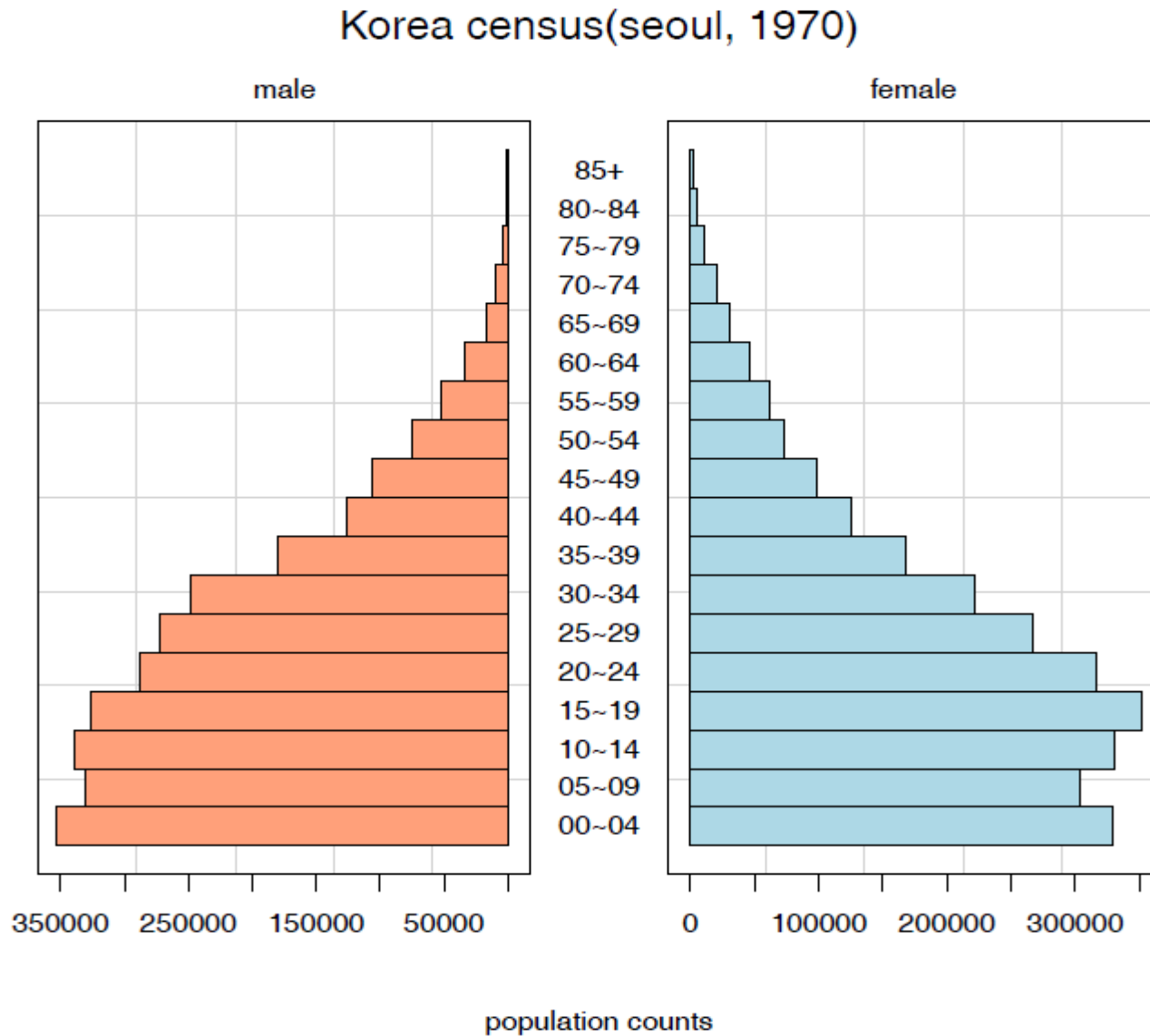
```
boxscatter <- function (x, y, main="Boxplot and Scatterplot",  
                        pch=16, bpch=16, col=1, bcol=0,  
                        xlab = NULL, ylab = NULL, ...)
```

● 호출방법

```
> boxscatter(cars$speed, cars$dist)  
> boxscatter(cars$speed, cars$dist, pch="*")  
> boxscatter(cars$speed, cars$dist, xlab="Speed (mph)",  
+ ylab="Stopping distance (ft)", main="Speed and Stopping  
+ Distances of Cars", col="blue", bcol="lightgray", cex=1.2)
```

pairbar plot

사용자 정의 그래프 함수



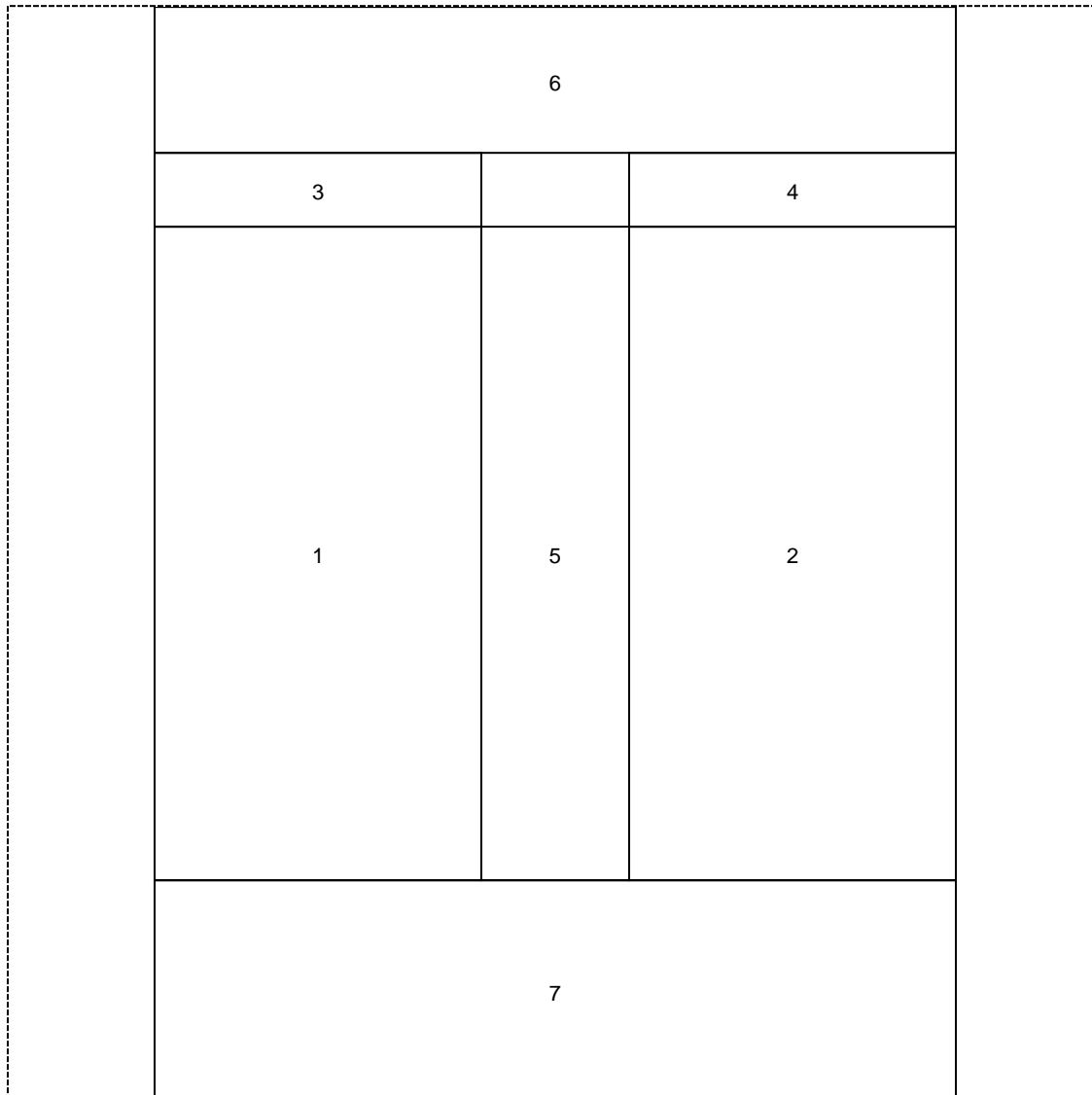
● layout 함수를 이용한 plot region의 분할

● > layout 함수를 이용한 plot region의 분할

```
> layout(matrix(c(0,6,6,6,0,  
+                0,3,0,4,0,  
+                0,1,5,2,0,  
+                0,7,7,7,0),  
+          nc = 5, byrow = TRUE),  
+        widths = c(lcm(2), 1, lcm(2), 1, lcm(2)),  
+        heights = c(lcm(2), lcm(1), 1, lcm(3)))  
> layout.show(7)  
> box("outer", lty = "dotted")
```

pairbar plot

사용자 정의 그래프 함수



pairbar plot

사용자 정의 그래프 함수

● plotPairBar 함수

● 함수원형

```
plotPairBar <- function(left, right, level, main = NULL, sub = NULL,  
  l.lab = NULL, r.lab = NULL,  
  l.col = "lightsalmon", r.col="lightblue")
```

● 호출방법

```
> male <- xtabs(frequency~gender+age.group, data=population.info,  
+ subset=gender=="male" & year=="1970" & region=="Seoul",  
+ drop.unused.levels=T)  
> female <- xtabs(frequency~gender+age.group, data=population.info,  
+ subset=gender=="female" & year=="1970" & region=="Seoul",  
+ drop.unused.levels=T)  
> plotPairBar(male, female, names(male[1,]))  
> plotPairBar(male, female, names(male[1,]),  
+ main="Korea census(seoul, 1970)", sub="population counts")  
> plotPairBar(male, female, names(male[1,]), main="Korea census(seoul, 1970)",  
+ sub="population counts", l.col="limegreen", r.col="gold")
```

타 패키지 확장하기

● RGoogleMaps package

- Google의 Google Static Maps API와 연동하는 R package
- Google Static Maps API를 호출하는 기본 기능만 있음
- 사용자가 다양한 주제도를 그리기 위해서는 함수 개발이 필요함

● 함수 구현 대상

● Scatter Plot

- 지도 위에 scatter plot을 그림
- frequency의 규모에 따라 points의 크기를 달리함
- 비율척도의 값에 따라 points의 색상을 달리함

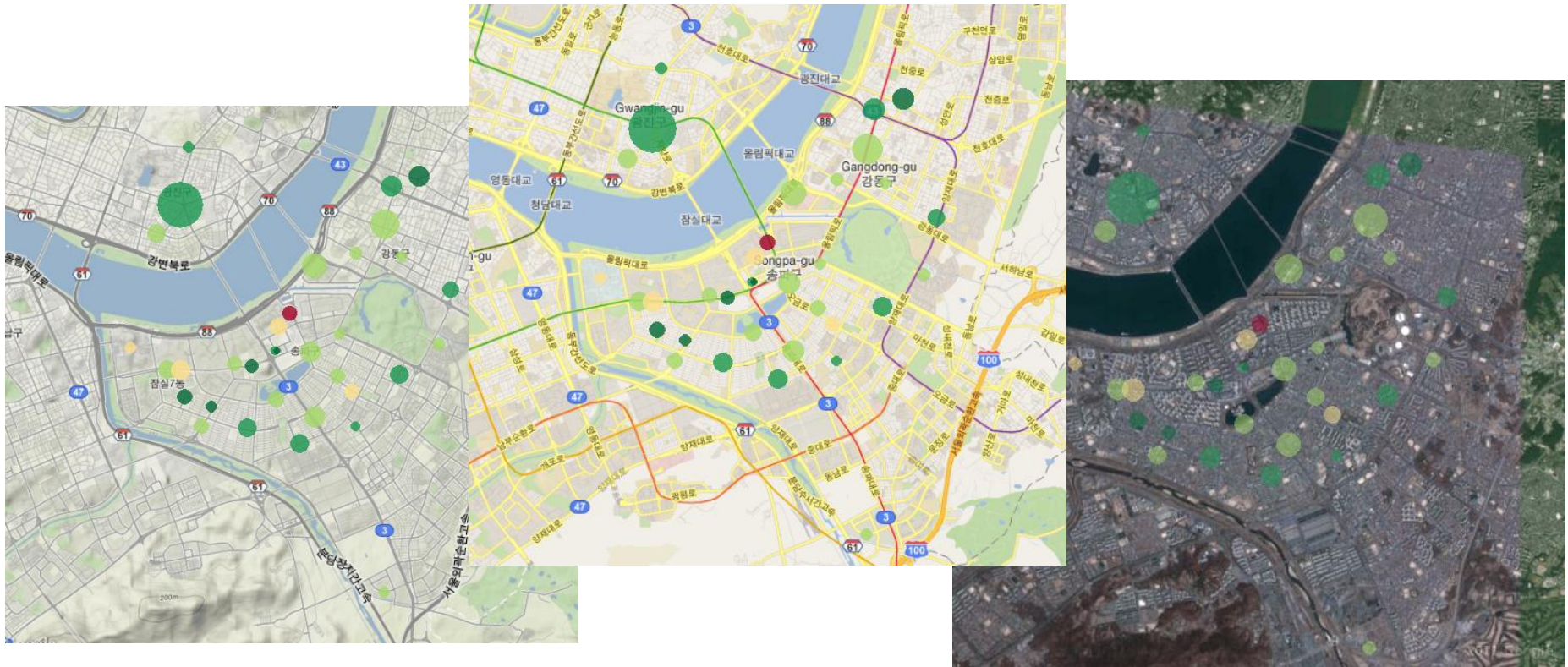
● Pie Chart

- 지도 위에 파이차트를 그림
- frequency의 규모에 따라 반지름의 크기를 달리함
- 범주별 frequency에 따라서 파이의 크기를 달리함

RGoogleMaps extension

타 패키지 확장하기

● scatter plot



● ScatterOnStaticMap 함수

● 함수원형

```
ScatterOnStaticMap <- function(x, lat, lon, size=scale(x[,1]),  
  xlim=range(lon), ylim=range(lat), col=NULL, pch=16, cex=NULL,  
  col.regions=NULL, level.cnt=10, maptype=c("roadmap", "mobile",  
  "satellite", "terrain", "hybrid", "mapmaker-roadmap", "mapmaker-  
  hybrid")[4], alpha=1, title=NULL)
```

● 호출방법

```
> head(x)  
  I_LATITUDE  I_LONGITUDE    CNT    RATIO  
1   37.51484    127.0738    309         3  
2   37.53197    127.0786    720         2  
3   37.51155    127.0806    741         2  
> ScatterOnStaticMap(x=x[,c("CNT","RATIO")], lat=x$I_LATITUDE,  
+ lon=x$I_LONGITUDE, alpha=0.8,  
+ title=list(labels="Test scatterplot", font=3, col=4))
```

● ScatterOnStaticMap 주요 코드

```
library(RColorBrewer)
```

```
library(RgoogleMaps)
```

```
alpha <- round(alpha * 255)
```

```
if(is.null(col.regions)) col.regions <-  
  colorRampPalette(rev(brewer.pal('RdYlGn', n=10)))
```

```
if (is.null(col)) col <- level.colors(x[,2], at=do.breaks(range(x[,2]), level.cnt),  
  col.regions=col.regions)
```

```
col <- apply(col2rgb(col), 2, function(x) rgb(x[1], x[2], x[3], alpha=alpha, max=255))
```

```
if (is.null(cex)) cex <- size + abs(min(size)) + 1
```

```
filename <- "tmp.png"
```

```
MyMap <- GetMap.bbox(lonR=xlim, latR=ylim, maptype=maptype, destfile=filename)
```

```
tmp <- PlotOnStaticMap(MyMap, lat=lat, lon=lon, cex=cex, pch=pch, col=col, add=F)
```

RGoogleMaps extension

타 패키지 확장하기

● pie chart



● PieOnStaticMap 함수

● 함수원형

```
PieOnStaticMap <- function(x, lat, lon,  
  maptype=c("roadmap", "mobile", "satellite", "terrain", "hybrid",  
    "mapmaker-roadmap", "mapmaker-hybrid")[4],  
  edges=200, clockwise=FALSE, init.angle=if (clockwise) 90 else 0,  
  col=NULL, alpha=1, title=NULL)
```

● 호출방법

```
x <- rbind(c(800, 400, 800), c(300, 200, 500))  
lat <- c(35.18254, 35.19294)  
lon <- c(129.2041, 129.2061)
```

```
PieOnStaticMap(x, lat, lon, alpha=0.8)
```

RGoogleMaps extension

타 패키지 확장하기

● ScatterOnStaticMap 주요 코드

```
radius <- margin.table(x, 1)/max(margin.table(x, 1)) * MyMap$zoom * 2

x <- prop.table(x, 1)
x <- apply(x, 1, function(t) c(0, cumsum(t)/sum(t)))
dx <- diff(x)
nx <- apply(dx, 2, length)

t2xy <- function(t, radius) {
  t2p <- ifelse(clockwise, -2, 2) * pi * t + init.angle * pi/180
  list(x = radius * cos(t2p), y = radius * sin(t2p))
}

for (i in 1L:NCOL(x)) {
  cols <- rep(col, length.out=nx[i])
  for (j in 1L:nx[i]) {
    n <- max(2, floor(edges * dx[j, i]))
    P <- t2xy(seq.int(x[j, i], x[j + 1, i], length.out=n), radius[i])
    polygon(c(P$x, 0)+locs$newX[i],c(P$y,0)+locs$newY[i],border="#00000000",col=cols[j])
  }
}
```

Q & A