

# 统计图形和 R

2017 年 12 月

邹海翔

深圳市规划国土发展研究中心





# 目录

统计图形和 R

R 绘图

R 图形混合嵌入

1 R 绘图

2 R 图形混合嵌入



# 目录

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

3

1 R 绘图

2 R 图形混合嵌入



# grid 绘图系统

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

4

- 前面介绍的 graphics 程序包是 R 的标准绘图系统,也称为 graphics 绘图系统,除此之外 R 中还有另一套截然不同的 grid 绘图系统
- 使用 grid 绘图系统前需要先用 library(grid) 加载 grid 程序包,该程序包由 [Paul Murrell](#) 开发维护
- grid 绘图系统的设计初衷是为了克服 graphics 系统中元素不能动态修改的弱点

16

# grid 绘图系统

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

4

- 前面介绍的 graphics 程序包是 R 的标准绘图系统,也称为 graphics 绘图系统,除此之外 R 中还有另一套截然不同的 grid 绘图系统
- 使用 grid 绘图系统前需要先用 library(grid) 加载 grid 程序包,该程序包由 [Paul Murrell](#) 开发维护
- grid 绘图系统的设计初衷是为了克服 graphics 系统中元素不能动态修改的弱点

```
1 # 基础统计图形库处理方式
2 plot(0:1, 0:1)
3 rect(0, 0, 1, 1, col = "red")
4 # 为了改变颜色,必须重画整幅图形
5 plot(0:1, 0:1)
6 # 虽然可以用新的矩形覆盖旧的,但旧矩形仍然存在
7 rect(0, 0, 1, 1, col = "blue")
8
```

```
1 # grid 绘图系统的处理方式
2 grid.rect(name = "rect0")
3 # 修改它的填充颜色为红色
4 grid.edit("rect0", gp = gpar(fill =
5   ↪ "red"))
6 # 修改为蓝色,不需要重新用 grid.rect() 画矩形
7 grid.edit("rect0", gp = gpar(fill =
8   ↪ "blue"))
```

# grid 绘图系统

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

4

- 前面介绍的 graphics 程序包是 R 的标准绘图系统,也称为 graphics 绘图系统,除此之外 R 中还有另一套截然不同的 grid 绘图系统
- 使用 grid 绘图系统前需要先用 library(grid) 加载 grid 程序包,该程序包由 [Paul Murrell](#) 开发维护
- grid 绘图系统的设计初衷是为了克服 graphics 系统中元素不能动态修改的弱点

## grid 系统和 graphics 系统的区别

- ☞ grid 用视口 (viewports) 将绘图设备分割为不同的区域,绘图对象 (grob) 可以在不同的视口中进行共享,比 graphics 中的处理方式更加灵活
- ☞ grid 绘图对象可以被修改或者从一个图形中移除,而不需要重新绘制所有的图形,但是在 graphics 中则必须重绘
- ☞ grid 绘图系统是一个绘图框架,其原生的 grid 程序包仅提供低级绘图函数用于绘制统计图形中的元素,不像 graphics 程序包还集成了高级绘图函数用于绘制常用的统计图形,因此直接用 grid 程序包绘制统计图形比较繁琐
- ☞ 两套系统的绘图函数和绘图参数完全不同,不能混用!

# grid 绘图系统

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

4

- 前面介绍的 graphics 程序包是 R 的标准绘图系统,也称为 **graphics 绘图系统**,除此之外 R 中还有另一套截然不同的 **grid 绘图系统**
- 使用 grid 绘图系统前需要先用 **library(grid)** 加载 grid 程序包,该程序包由 [Paul Murrell](#) 开发维护
- grid 绘图系统的设计初衷是为了克服 graphics 系统中元素不能动态修改的弱点

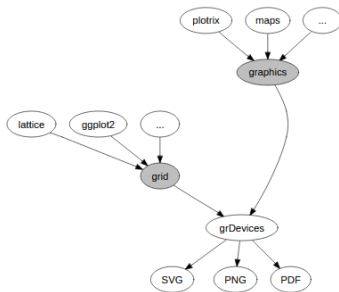


图: **lattice**和**ggplot2**是基于 grid 包开发的绘图程序包,这样就在 grid 绘图系统中使用高级绘图函数来简化统计图形的绘制过程

16

# lattice 程序包

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

5

- lattice 程序包是基于 grid 包的编写的一套统计图形库，2000 年发布了第一个版本，由 [Deepayan Sarkar](#) 等人开发维护，从 R 3.0 版本开始纳入 base 包，**不需要再单独安装**
- lattice 设计理念来自 S-PLUS 中的 Trellis 图形，是一种**多元数据可视化的方法**，其所有绘图要素被保存在一个 **trellis 对象** 中，并且在**嵌板 (panel)** 中绘制，另外，在每个 panel 上方还有一个条板 (strip) 区域用来描述分类信息



# lattice 程序包

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

5

- lattice 程序包是基于 grid 包的编写的一套统计图形库，2000 年发布了第一个版本，由 [Deepayan Sarkar](#) 等人开发维护，从 R 3.0 版本开始纳入 base 包，**不需要再单独安装**
- lattice 设计理念来自 S-PLUS 中的 Trellis 图形，是一种**多元数据可视化的方法**，其所有绘图要素被保存在一个 **trellis 对象** 中，并且在**嵌板 (panel)** 中绘制，另外，在每个 panel 上方还有一个条板 (strip) 区域用来描述分类信息

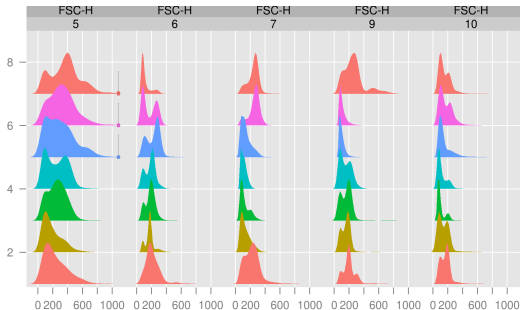


图: 不同的 GvHD 病患者在细胞检测中的 FSC-H 结果数据

16

# lattice 程序包

统计图形和 R

R 绘图

grid 绘图系统

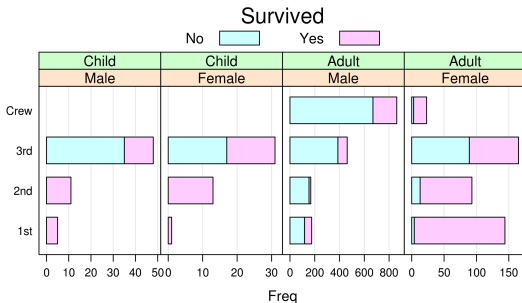
lattice 程序包

ggplot 程序包

R 图形混合嵌入

5

- lattice 程序包是基于 grid 包的编写的一套统计图形库，2000 年发布了第一个版本，由 [Deepayan Sarkar](#) 等人开发维护，从 R 3.0 版本开始纳入 base 包，**不需要再单独安装**
- lattice 设计理念来自 S-PLUS 中的 Trellis 图形，是一种**多元数据可视化的方法**，其所有绘图要素被保存在一个 **trellis 对象** 中，并且在**嵌板 (panel)** 中绘制，另外，在每个 panel 上方还有一个条板 (strip) 区域用来描述分类信息



图：泰坦尼克号生存率的交叉分类数据

16



# lattice 程序包

公式参数

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

6

- lattice 的绘图函数都会用公式作为第一位置参数
- 如果在不同 panel 中绘制多元数据，公式参数根据 “|” 符号后的变量对数据进行分类
- 如果是在同一 panel 中绘制多元数据，则使用 group 参数对数据进行分类

16

- lattice 的绘图函数都会用公式作为第一位置参数
- 如果在不同 panel 中绘制多元数据，公式参数根据 “|” 符号后的变量对数据进行分类
- 如果是在同一 panel 中绘制多元数据，则使用 group 参数对数据进行分类

公式参数	含义
$\sim y$	单变量数据
$\sim y z$	根据 $z$ 变量对单变量数据划分 panel
$y \sim x$	二元变量数据
$y \sim x z$	根据 $z$ 变量对二元变量数据划分 panel
$y \sim x a + b$	根据多条件变量划分 panel，等价于 $y \sim x a$ 和 $y \sim x b$
$y_1 + y_2 \sim x$	多元变量数据绘图，等价于 $y_1 \sim x$ 和 $y_2 \sim x$
$z \sim x * y$	绘制三维图形 $(x,y,z)$

表: lattice 中的公式参数



# lattice 程序包

公式参数

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

6

- lattice 的绘图函数都会用公式作为第一位置参数
- 如果在不同 panel 中绘制多元数据，公式参数根据 “|” 符号后的变量对数据进行分类
- 如果是在同一 panel 中绘制多元数据，则使用 group 参数对数据进行分类

16

# lattice 程序包

## 公式参数

统计图形和 R

```
1. > densityplot(mpg, data=mtcars)
```

R 绘图

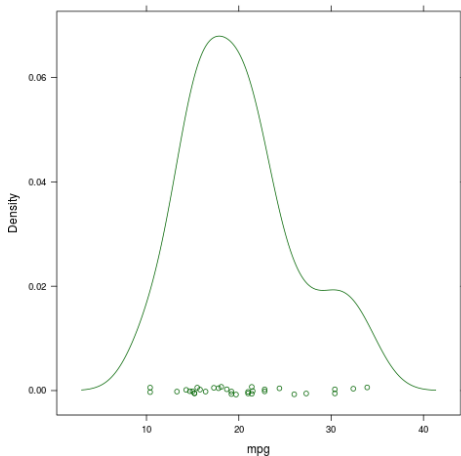
grid 绘图系统

lattice 程序包

ggplot 程序包

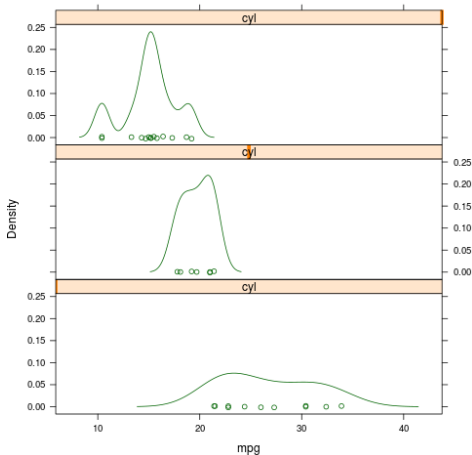
R 图形混合嵌入

7



图：在 panel 中绘制单变量数据

```
1. > densityplot(~mpg | cyl, data=mtcars, layout=c(1,3))
```



图：在不同 panel 中绘制单变量分类数据



# lattice 程序包

## 公式参数

统计图形和 R

R 绘图

grid 绘图系统

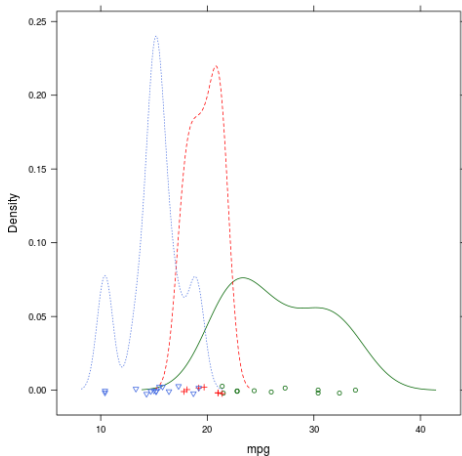
lattice 程序包

ggplot 程序包

R 图形混合嵌入

7

```
1. > densityplot(~mpg, data=mtcars, group=cyl)
```



图：在同一 panel 中绘制单变量分类数据



# lattice 程序包

## 公式参数

统计图形和 R

```
1 > EE <- equal.count(ethanol$E, number=9, overlap=1/4)
2 > xyplot(NOx ~ C | EE, data = ethanol)
```

R 绘图

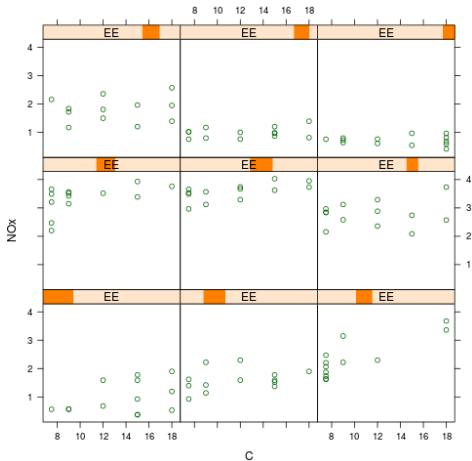
grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

7



图：在不同 panel 中绘制多变量分类数据

16

- lattice 包中提供了大量标准高级绘图函数用于直接绘制常用的统计图形

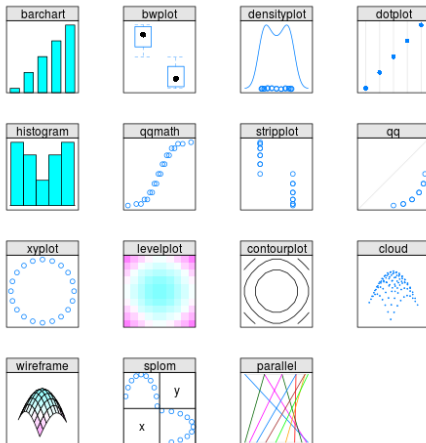


图: lattice 中的标准高级绘图函数

# lattice 程序包

## 标准高级绘图函数

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

9

lattice 函数	公式参数	描述	graphics 对应函数
barchart()	$y \sim x$	条形图	barplot()
bwplot()	$y \sim x$	箱线图	boxplot()
densityplot()	$\sim y$	核密度图	plot.density()
dotplot()	$\sim y$	Cleveland 点图	dotchart()
histogram()	$\sim x$	直方图	hist()
stripplot()	$\sim y$	带状图	stripchart()
xyplot()	$y \sim x$	散点图	plot()
contourplot()	$z \sim x * y$	等高线图	contour()
cloud()	$z \sim x * y$	三维散点图	无
levelplot()	$z \sim x * y$	颜色图	image()
wireframe()	$z \sim x * y$	三维透视图	persp()
qq()	$\sim x$	QQ 图	qqnorm()
splom()	$\sim data.frame$	散点图矩阵	pairs()
parallel()	$\sim data.frame$	平行坐标图	无

表: lattice 包与 graphics 包的对应函数



# lattice 程序包

panel 函数和 strip 函数

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

10

- lattice 中每个高级绘图函数都有默认的 panel 参数和 strip 参数，实质上对应的是两个匿名函数：**panel()** 和 **strip()**
- 这两个函数可以用来对 panel 区域和 strip 区域需要绘制图形以及显示的分类描述信息进行自定义扩展

16

# lattice 程序包

## panel 函数和 strip 函数

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

11

```

1 # 在高级绘图函数中自定义 panel 和 strip 的示例
2 types.plain <- c("p", "l", "o", "r", "g", "s", "S", "h", "a", "smooth")
3 types.horiz <- c("s", "S", "h", "a", "smooth")
4 horiz <- rep(c(FALSE, TRUE), c(length(types.plain), length(types.horiz)))
5 types <- c(types.plain, types.horiz)
6 x <- sample(seq(-10, 10, length.out = 15), 30, TRUE)
7 y <- x + 0.25 * (x + 1)^2 + rnorm(length(x), sd = 5)
8
9 xyplot(y ~ x | gl(1, length(types)),
10        xlab = "type",
11        ylab = list(c("horizontal=TRUE", "horizontal=FALSE"), y = c(1/6, 4/6)),
12        ↪ as.table = TRUE, layout = c(5, 3), between = list(y = c(0, 1)),
13        # 自定义 strip 函数,... 参数表示直接继承 xyplot 中的其他参数
14        strip = function(...) {
15          # 调用标准 panel 函数 panel.fill 填充每个 strip 的颜色
16          panel.fill(trellis.par.get("strip.background")$col[1])
17          type <- types[panel.number()]
18          # 调用底层 grid 绘图函数
19          grid::grid.text(label = sprintf("%s", type), x = 0.5, y = 0.5)
20          grid::grid.rect()
21        },
22        scales = list(alternating = c(0, 2), tck = c(0, 0.7), draw = FALSE),
23        par.settings = list(layout.widths = list(strip.left = c(1, 0, 0, 0, 0))),
24        # 自定义 panel 函数,... 参数表示直接继承 xyplot 中的其他参数
25        panel = function(...) {
26          type <- types[panel.number()]
27          horizontal <- horiz[panel.number()]
28          # 调用标准 panel 函数 panel.xyplot 按照预设参数每个 panel 中绘制图形
29          panel.xyplot(...,
30                       type = type,
31                       horizontal = horizontal)
32        })[rep(1, length(types))]
```

16

# lattice 程序包

## panel 函数和 strip 函数

统计图形和 R

R 绘图

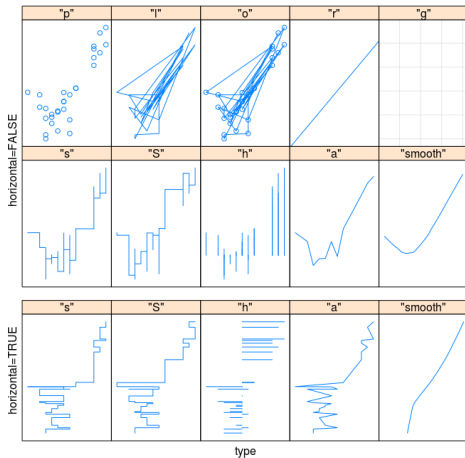
grid 绘图系统

**lattice 程序包**

ggplot 程序包

R 图形混合嵌入

11



图：在高级绘图函数 `xyplot` 中自定义 `panel` 和 `strip` 的示例

16

# lattice 程序包

## panel 函数和 strip 函数

统计图形和 R

R 绘图

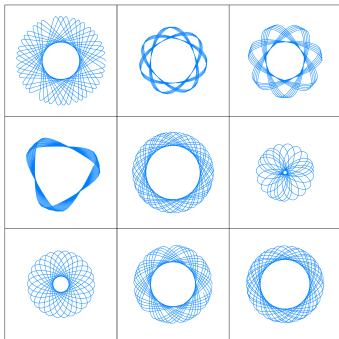
grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

12



```

1 # 自定义一个 panel 函数绘制内旋轮线
2 # 注意: 这个函数的所有参数都不是必选参数, 而且没有...
  ↳ 参数, 这意味外部数据无法传入该函数参与绘图
3 > panel.hypotrochoid <- function(r, d,
  ↳ cycles = 10, density = 30)
4 {
5   if (missing(r)) r <- runif(1, 0.25, 0.75)
6   if (missing(d)) d <- runif(1, 0.25 * r, r)
7   t <- 2*pi*seq(0,cycles,by = 1/density)
8   x <- (1-r)*cos(t)+d*cos((1-r)*t/r)
9   y <- (1-r)*sin(t)-d*sin((1-r)*t/r)
10  panel.lines(x, y)
11 }
12 # 自定义 prepanel 函数来绘制 panel 的外框
13 > prepanel.hypocycloid <- function(x, y) {
14   list(xlim=c(-1, 1),ylim = c(-1, 1))
15 }
16
17 # 将 xyplot 函数传递给一个 trellis 对象 p, 这里传
  ↳ 入的 x 参数其实并没有参与绘图
18 > p <- xyplot(x=c(-1, 1) ~ c(-1, 1), aspect
  ↳ = 1, cycles = 15, scales = list(draw
  ↳ = FALSE), xlab = "", ylab = "",
  ↳ panel = panel.hypotrochoid)
19 # 对象 p 循环绘图
20 > p[rep(1, 9)]
  
```

图: 通过外部自定义 panel 函数来绘制图形

16

# lattice 程序包

图形参数和主题

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

13

- 在 lattice 中, 所有的 trellis 对象都包含一个主题 (theme), theme 中拥有完整的图形要素设置
- 当前 theme 的参数可以直接通过 `trellis.par.get()` 函数获取, 通过 `trellis.par.set()` 函数直接修改; 对 theme 参数的设置作用于当前绘图设备中的所有高级绘图函数

```

1 # 罗列 trellis 对象中的所有图形参数
2 > names(trellis.par.get())
3 [1] "grid.pars"      "fontsize"      "background"
4 [4] "panel.background" "clip"          "add.line"
5 [7] "add.text"       "plot.polygon"  "box.dot"
6 [10] "box.rectangle"  "box.umbrella"  "dot.line"
7 [13] "dot.symbol"     "plot.line"     "plot.symbol"
8 [16] "reference.line" "strip.background" "strip.shingle"
9 [19] "strip.border"   "superpose.line" "superpose.symbol"
10 [22] "superpose.polygon" "regions"       "shade.colors"
11 [25] "axis.line"      "axis.text"     "axis.components"
12 [28] "layout.heights" "layout.widths" "box.3d"
13 [31] "par.xlab.text"  "par.ylab.text" "par.zlab.text"
14 [34] "par.main.text"  "par.sub.text"

```

16



- 在 lattice 中, 所有的 trellis 对象都包含一个主题 (theme), theme 中拥有完整的图形要素设置
- 当前 theme 的参数可以直接通过 `trellis.par.get()` 函数获取, 通过 `trellis.par.set()` 函数直接修改; 对 theme 参数的设置作用于当前绘图设备中的所有高级绘图函数

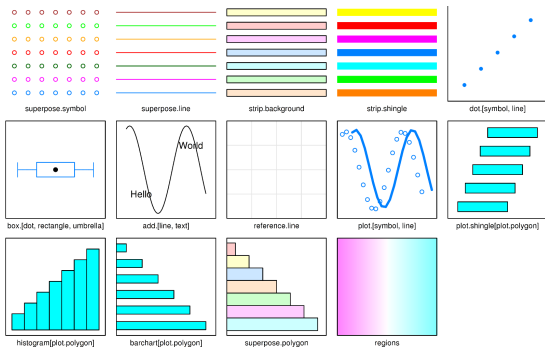
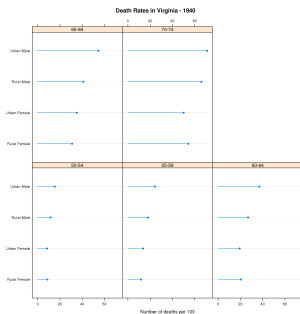


图: trellis 对象中所有的图形参数

- 在 lattice 中, 所有的 trellis 对象都包含一个主题 (theme), theme 中拥有完整的图形要素设置
- 当前 theme 的参数可以直接通过 `trellis.par.get()` 函数获取, 通过 `trellis.par.set()` 函数直接修改; 对 theme 参数的设置作用于当前绘图设备中的所有高级绘图函数



```
1 # 绘制 dotplot 传递给 trellis 对象 vad.plot
2 > vad.plot <-
3   dotplot(reorder(Var2, Freq)~Freq | Var1,
4           data = as.data.frame.table(VADeaths),
5           origin = 0, type = c("p", "h"),
6           main = "Death Rates in Virginia -
7               ↪ 1940",
8           xlab = "Number of deaths per 100")
9 > vad.plot
```

图: 通过直接修改 trellis 对象的图形参数实现修改图形

# lattice 程序包

图形参数和主题

统计图形和 R

R 绘图

grid 绘图系统

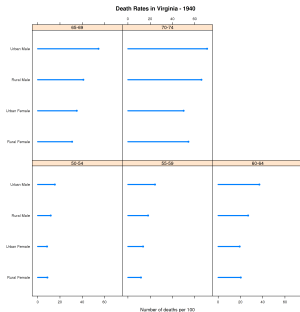
lattice 程序包

ggplot 程序包

R 图形混合嵌入

13

- 在 lattice 中, 所有的 trellis 对象都包含一个主题 (theme), theme 中拥有完整的图形要素设置
- 当前 theme 的参数可以直接通过 `trellis.par.get()` 函数获取, 通过 `trellis.par.set()` 函数直接修改; 对 theme 参数的设置作用于当前绘图设备中的所有高级绘图函数



```

1 # 在上图基础上修改绘图参数
2 # 获得当前主题的 dot.line 设置
3 > dot.line.settings <- trellis.par.get("dot.line")
4 # 将 dot.line 的颜色设置为透明不可见
5 > dot.line.settings$col <- "transparent"
6 # 应用新的参数设置
7 > trellis.par.set("dot.line", dot.line.settings)
8 # 获得当前主题的 plot.line 设置
9 > plot.line.settings <- trellis.par.get("plot.line")
10 # 将 plot.line 的线宽设置为 3, 默认是 1
11 > plot.line.settings$lwd <- 3
12 # 应用新的参数设置
13 > trellis.par.set("plot.line", plot.line.settings)
14 > vad.plot
  
```

图: 通过直接修改 trellis 对象的图形参数实现修改图形

16



# lattice 程序包

图形参数和主题

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

14

- 除了设置 `theme` 参数之外，还可以通过 `par.settings` 参数仅对当前图形进行图形参数调整，这比较类似 `par()` 函数的作用
- 另外，`lattice` 中提供了 `update.trellis` 函数来更新 `trellis` 对象的参数，配合 `par.settings` 参数可以在不重绘图形的情况下实现图形的修改

16

# lattice 程序包

图形参数和主题

统计图形和 R

R 绘图

grid 绘图系统

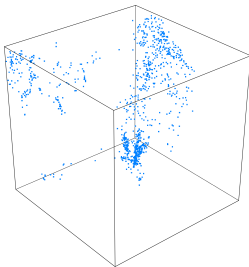
lattice 程序包

ggplot 程序包

R 图形混合嵌入

14

- 除了设置 theme 参数之外，还可以通过 **par.settings** 参数仅对当前图形进行图形参数调整，这比较类似 par() 函数的作用
- 另外，lattice 中提供了 **update.trellis** 函数来更新 trellis 对象的参数，配合 par.settings 参数可以在不重绘图形的情况下实现图形的修改



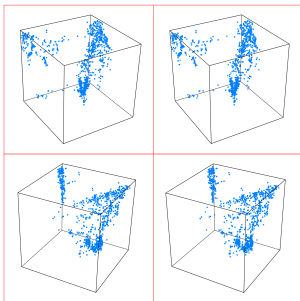
```

1 # 首先定义一个三维散点图，并将其传入 trellis 对象 p
2 > p <-
3   cloud(depth ~ long + lat, quakes, zlim = c(690,
4     ↳ 30), pch = ".", cex = 4, zoom = 1, xlab
5     ↳ = NULL, ylab = NULL, zlab =
6     ↳ NULL, scales = list(draw = FALSE),
7     # 用 par.settings 参数设置坐标线为透明
8     par.settings = list(axis.line = list(col =
9       ↳ "transparent")))
10 > p
  
```

图: 通过 update 函数和 par.settings 对象修改图形

16

- 除了设置 theme 参数之外，还可以通过 **par.settings** 参数仅对当前图形进行图形参数调整，这比较类似 par() 函数的作用
- 另外，lattice 中提供了 **update.trellis** 函数来更新 trellis 对象的参数，配合 par.settings 参数可以在不重绘图形的情况下实现图形的修改



```

1 > npanel <- 2
2 # 设置三维散点图的旋转视角
3 > rotz <- seq(-30, 30, length = npanel)
4 > roty <- c(3, 0)
5 # 用 update.trellis 函数更改原有图形
6 # 注意: 由于 update.trellis 函数是一个继承自 update 函
   ↳ 数的 S3 型对象，而传入的 p 是 trellis 对象，因此这
   ↳ 里直接可以写 update
7 > update(p[rep(1, 2 * npanel)],
8         layout = c(2, npanel),
9         panel = function(..., screen) {
10           crow <- current.row()
11           ccol <- current.column()
12           panel.cloud(..., screen = list(z =
13             ↳ rotz[crow], x = -60, y =
14             ↳ roty[ccol]))},
15 # 用 par.settings 参数设置坐标线颜色为红
16 par.settings=list(axis.line=list(col="red"))
  
```

图: 通过 trellis 对象的图形参数实现修改图形



# ggplot 程序包

统计图形和 R

R 绘图

grid 绘图系统

lattice 程序包

ggplot 程序包

R 图形混合嵌入

15

- lattice 包具有精细的绘图参数，对于定制统计图形具有极大地灵活性，但是过于复杂的参数往往将新手拒之门外；相比之下，ggplot2 包以易用性为原则进行设计，而同时又兼具了一定的灵活性

•

16



# 目录

统计图形和 R

R 绘图

R 图形混合嵌入

16

1 R 绘图

2 R 图形混合嵌入

16



汇报完毕  
谢谢!

