

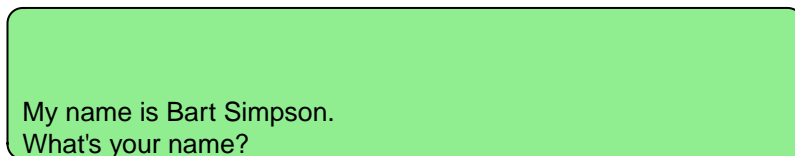
High level lattice functions differ from conventional R graphics functions because they do not draw anything. Instead, they create an object of class “trellis” which may be printed and plotted to produce actual plots.

Lattice plots are highly customizable and easy to be modified. The `par()` command used in the conventional R graphics has no effect in lattice plots. In what follows, I shall take you on a journey to explore fascinating functionalities that lattice plots have to offer.

```
grid.roundrect(width=.65,height=.2,gp=gpar(fill="yellow"))
grid.text("My name is Bart Simpson.")
```



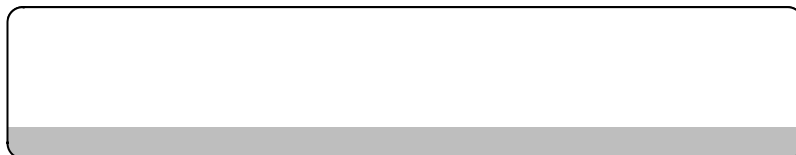
```
pushViewport(viewport(width=.65,height=.2))
grid.roundrect(gp=gpar(fill="light green"))
grid.text("My name is Bart Simpson.",x=unit(2, "mm"),y=unit(1.5, "lines"),just="left")
grid.text("What's your name?",x=unit(2, "mm"),y=unit(0.5, "lines"),just="left")
popViewport()
```



```
labels <- c("My name is Bart Simpson", "What's your name?")
vp <- viewport(width=max(stringWidth(labels))+unit(4, "mm"),height=
  unit(length(labels),"lines"))
pushViewport(vp)
grid.roundrect(gp=gpar(fill="cyan"))
grid.text(labels,x=unit(2, "mm"),y=unit(2:1 - 0.5, "lines"),just="left")
popViewport()
```

My name is Bart Simpson  
What's your name?

```
pushViewport(viewport(width=.65,height=.2))  
grid.roundrect(gp=gpar(fill="grey"))  
grid.clip(y=unit(1, "lines"),just="bottom")  
grid.roundrect(gp=gpar(fill="white"))  
popViewport()
```



```

labels <- c("My name is Bart Simpson.", "What's your name?", "My name is Homer Simpson.",
           "Boring!")
vp <-viewport(width=max(stringWidth(labels))+unit(4, "mm"),height=
             unit(length(labels),"lines"))
pushViewport(vp)
grid.roundrect()
grid.clip(y=unit(1, "lines"),just="bottom")
grid.roundrect(gp=gpar(fill="grey"))
grid.clip(y=unit(2, "lines"),just="bottom")
grid.roundrect(gp=gpar(fill="white"))
grid.clip(y=unit(3, "lines"),just="bottom")
grid.roundrect(gp=gpar(fill="grey"))
grid.clip()
grid.text(labels,x=unit(rep(2, 4), "mm"),y=unit(4:1 - .5, "lines"),just="left")
popViewport()

```

|                           |
|---------------------------|
| My name is Bart Simpson.  |
| What's your name?         |
| My name is Homer Simpson. |
| Boring!                   |

```

tableBox <- function(labels, x=.5, y=.5) {
  nlabel <- length(labels)
  tablevp <- viewport(x=x, y=y,width=max(stringWidth(labels)) + unit(4, "mm"),
                    height=unit(nlabel, "lines"))
  pushViewport(tablevp)
  grid.roundrect()
  if (nlabel > 1) {
    for (i in 1:(nlabel - 1)) {
      fill <- c("white", "grey")[i %% 2 + 1]
      grid.clip(y=unit(i, "lines"), just="bottom")
      grid.roundrect(gp=gpar(fill=fill))
    }
  }
}

```

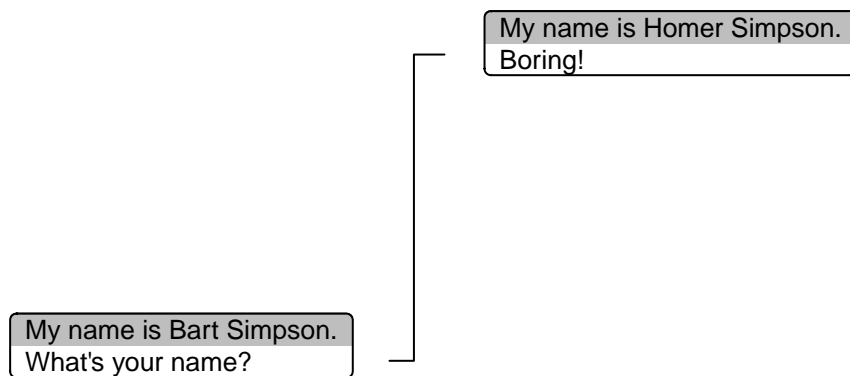
```

    }
  }
  grid.clip()
  grid.text(labels,x=unit(2, "mm"), y=unit(nlabel:1 - .5, "lines"),just="left")
  popViewport()
}

tableBox(c("My name is Bart Simpson.", "What's your name?"),x=0.3,y=0.3)
tableBox(c("My name is Homer Simpson.", "Boring!"),x=0.7,y=0.7)

x1 <- c(0.47, 0.49, 0.49, 0.515)
y1 <- c(0.28, 0.28, 0.685, 0.685)
grid.xspline(x1, y1)

```

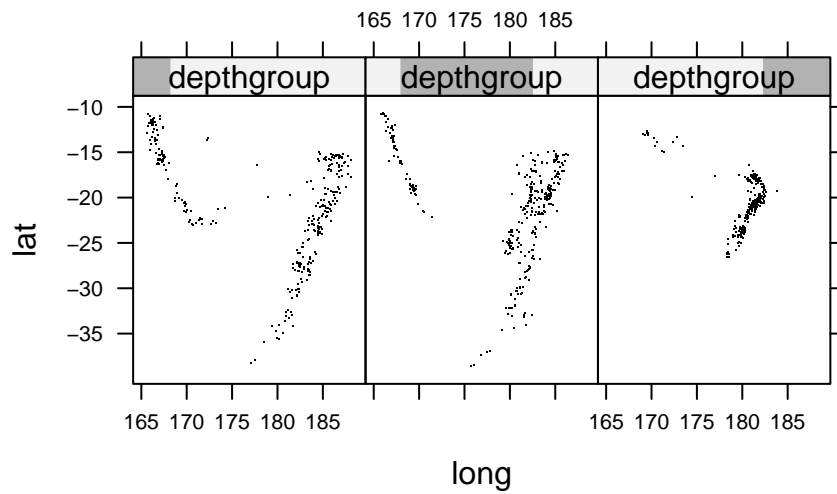


```

depthgroup <- equal.count(quakes$depth, number=3, overlap=0)

trellis.par.set(theme = canonical.theme("postscript", col=FALSE))
trellis.par.set(list(dot.symbol=list(pch=1), axis.text=list(cex=0.7)))
temp <- xyplot(lat ~ long | depthgroup,
              data=quakes, pch=".",
              layout=c(3, 1))
print(temp, panel.width=list(1.21, "inches"), panel.height=list(1.5, "inches"))

```



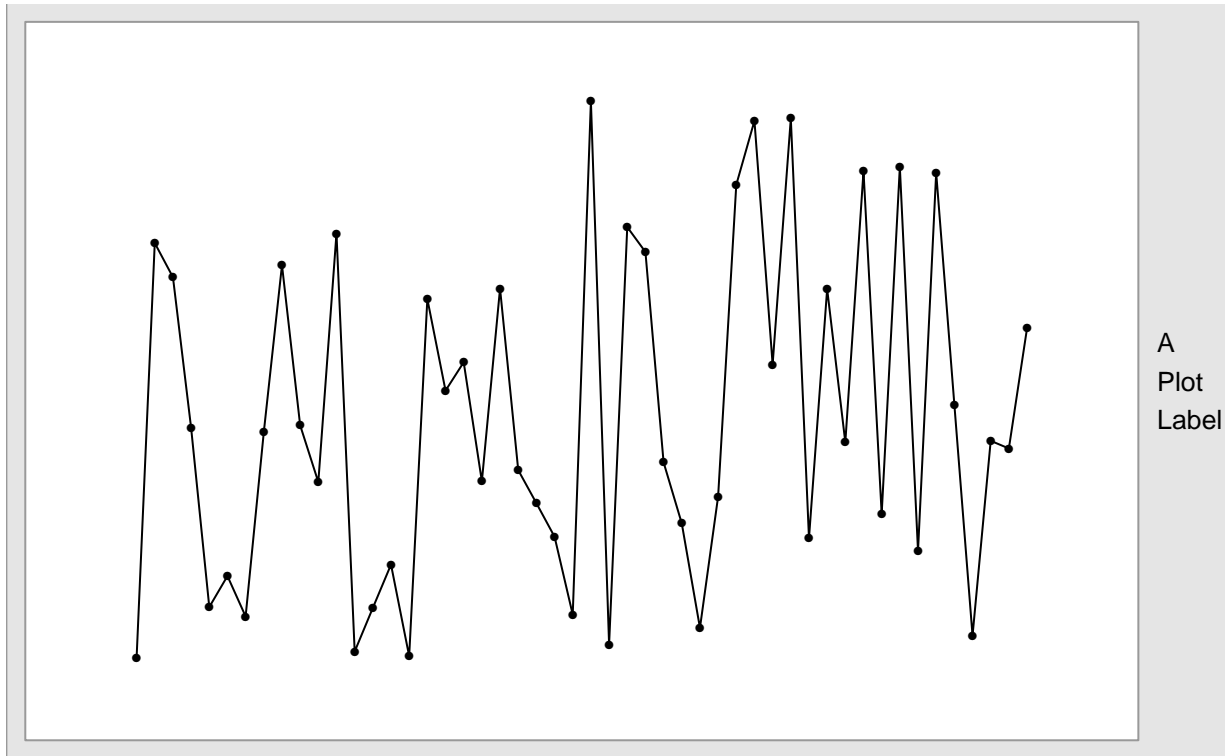
```

label <- textGrob("A\nPlot\nLabel ", x=0, just="left")
x <- seq(0.1, 0.9, length=50)
y <- runif(50, 0.1, 0.9)
gplot <- gTree(children=gList(rectGrob(gp=gpar(col="grey60", fill="white")),
                             linesGrob(x, y), pointsGrob(x, y, pch=16, size=unit(1.5, "mm"))),
vp=viewport(width=unit(1, "npc") - unit(5, "mm"), height=unit(1, "npc") - unit(5, "mm")))

layout <- grid.layout(1, 2, widths=unit(c(1, 1), c("null", "grobwidth")), list(NULL, label)))

grid.rect(gp=gpar(col="grey60", fill="grey90"))
pushViewport(viewport(layout=layout))
pushViewport(viewport(layout.pos.col=2))
grid.draw(label)
popViewport()
pushViewport(viewport(layout.pos.col=1))
grid.draw(gplot)
popViewport(2)

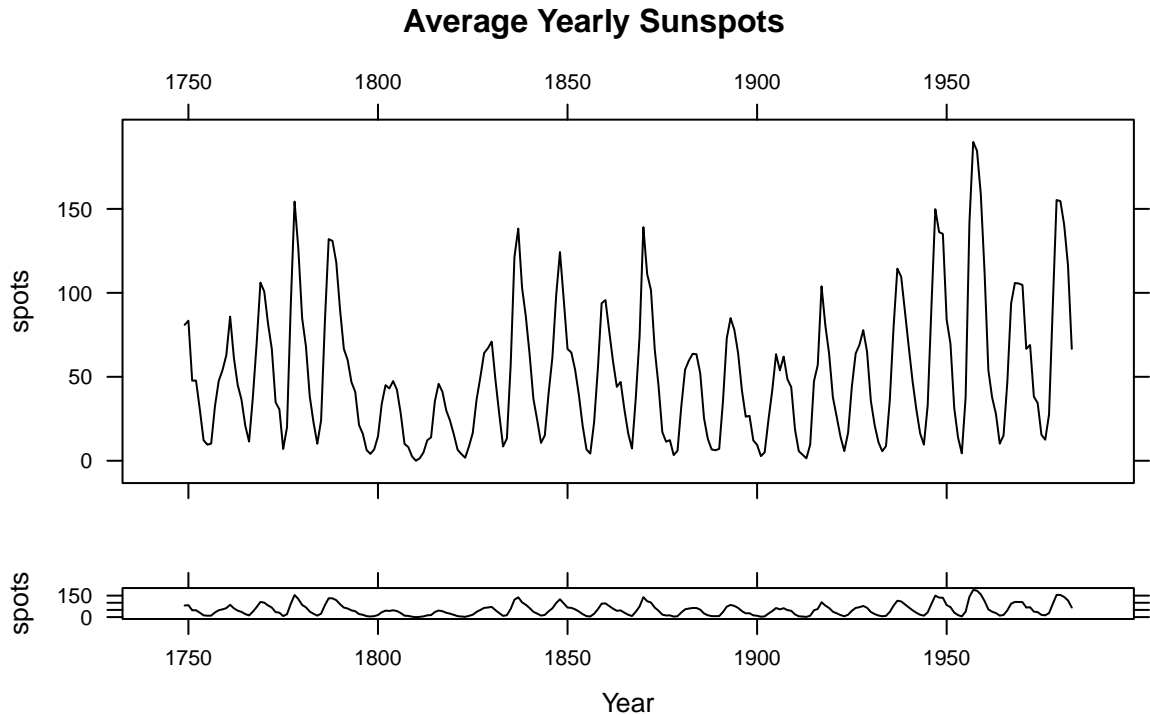
```



```

trellis.par.set(theme = canonical.theme("postscript", col=FALSE))
trellis.par.set(list(fontsize=list(text=10)))
spots <- by(sunspots, gl(235, 12, lab=1749:1983), mean)
plot1 <- xyplot(spots ~ 1749:1983, xlab="", type="l", main="Average Yearly Sunspots",
               scales=list(x=list(alternating=2)))
plot2 <- xyplot(spots ~ 1749:1983, xlab="Year", type="l")
print(plot1, position=c(0, 0.2, 1, 1), more=TRUE)
print(plot2, position=c(0, 0, 1, 0.33))

```



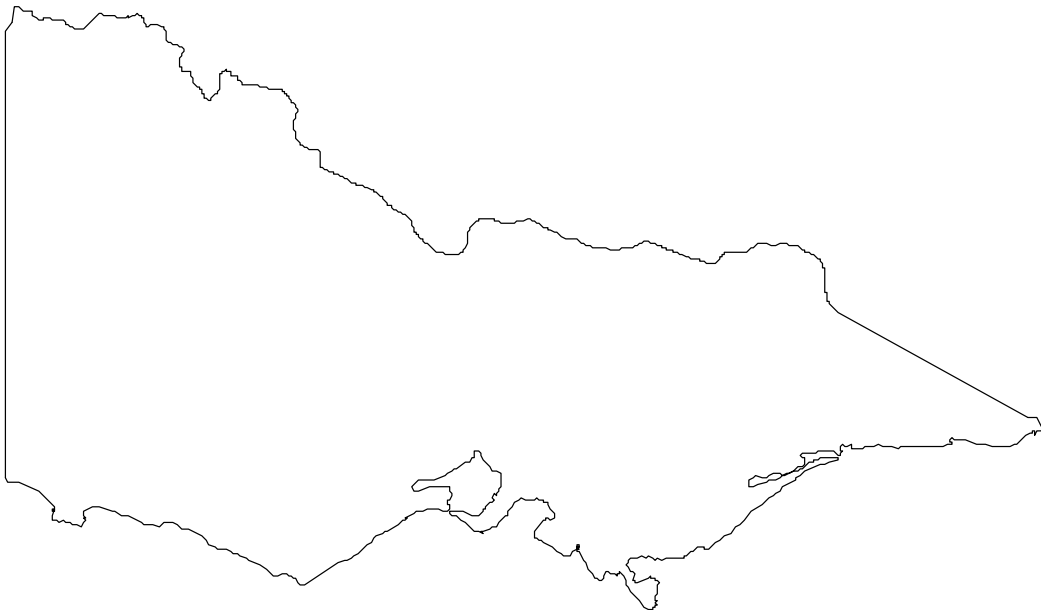
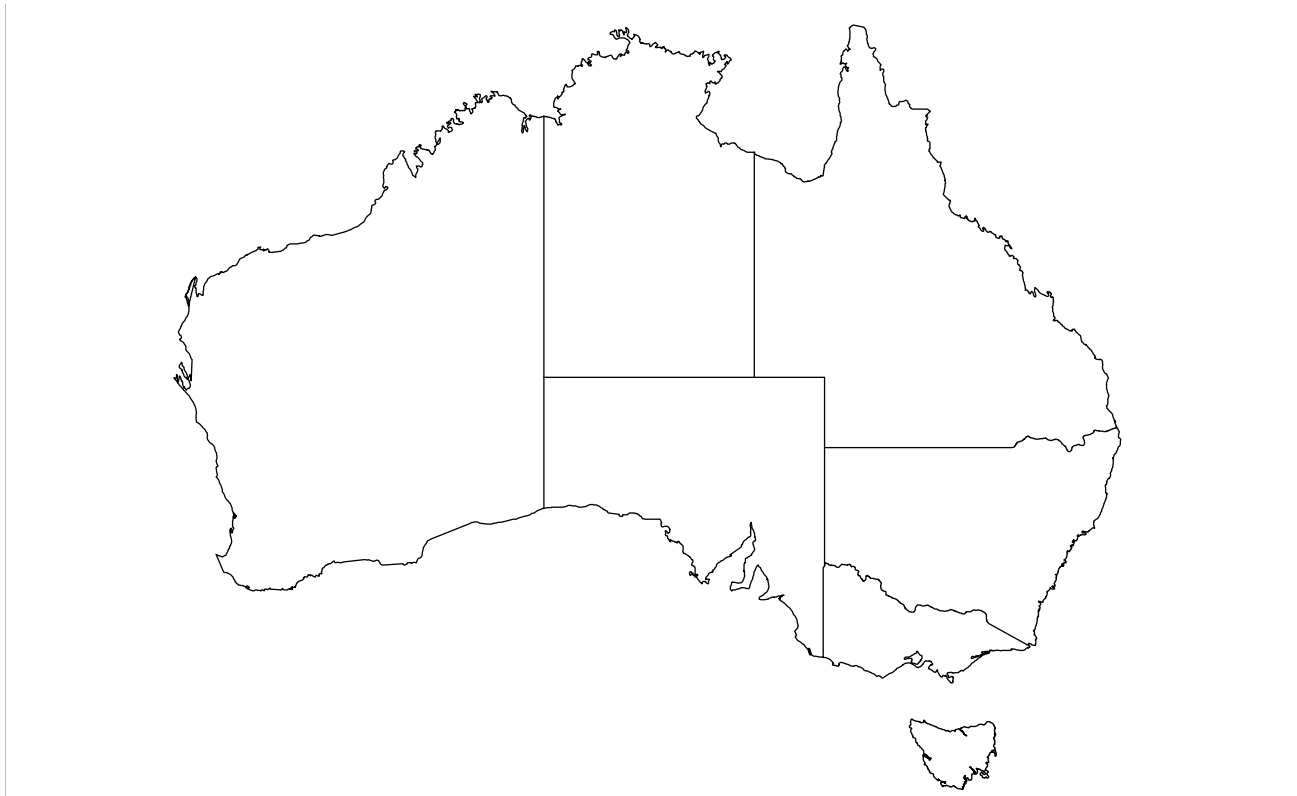
```

grid.ozFun <- function(ozRegion) {
  pushViewport(
    viewport(name="ozlay",
             layout=grid.layout(1,1,
                                widths=diff(ozRegion$rangex),
                                heights=diff(ozRegion$rangey),
                                respect=TRUE)))
  pushViewport(viewport(name="ozvp",
                        layout.pos.row=1,
                        layout.pos.col=1,
                        xscale=ozRegion$rangex,
                        yscale=ozRegion$rangey,
                        clip=TRUE))

  index <- 1
  for(i in ozRegion$lines) {
    grid.lines(i$x, i$y, default.units="native",
              name=paste("ozlines", index, sep=""))
    index <- index + 1
  }
  upViewport(2)
}

grid.rect(gp=gpar(col="grey"))
grid.ozFun(ozRegion())

```



#### References:

- [1] Murrell, Paul. (2006) R Graphics, Chapman & Hall/CRC.
- [2] Sarkar, Deepayan. (2009) lattice: lattice graphics, R package version 0.17-25, <http://CRAN.R-project.org/package=lattice>.