

# Seeing Through grid Graphics

Paul Murrell, paul@stat.auckland.ac.nz

The University of Auckland

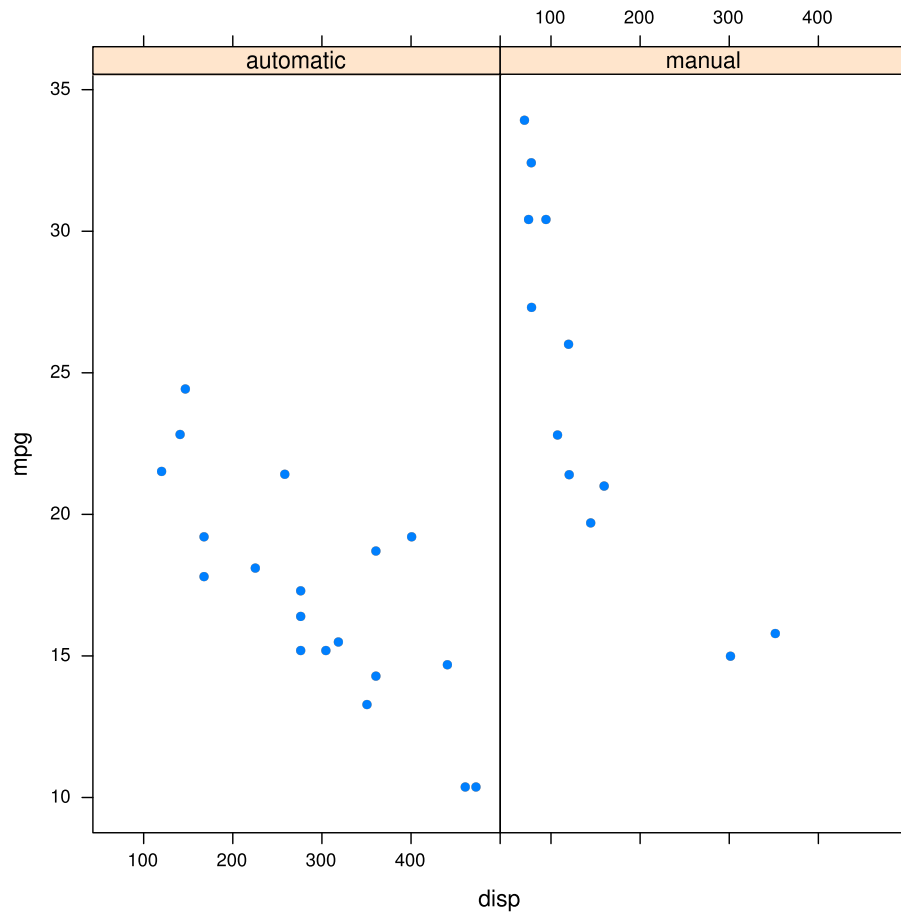
## Seeing Through grid Graphics

*The Big Idea:* **Don't be a Dead End**

## R Graphics

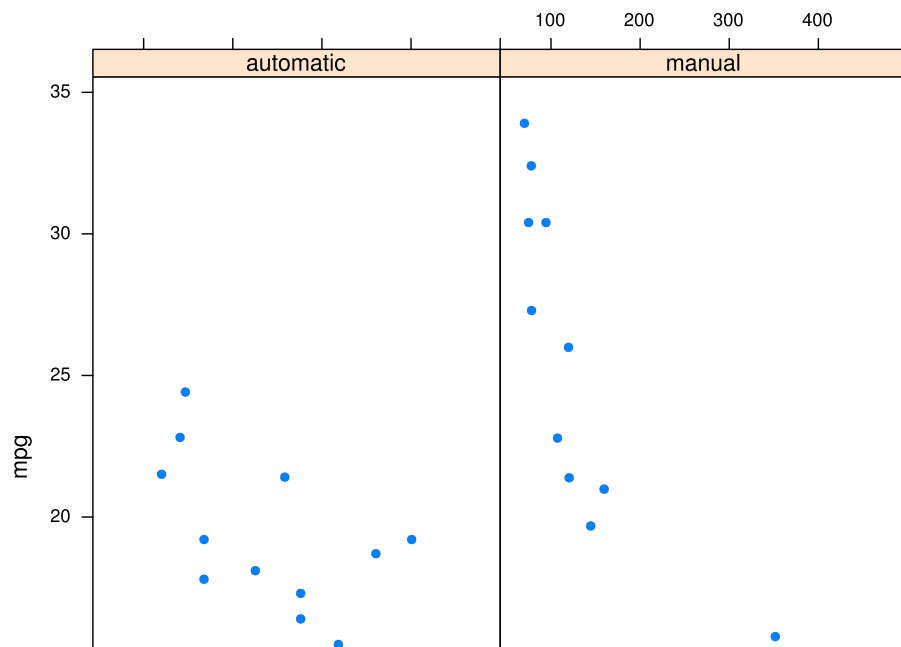
# An Example Plot

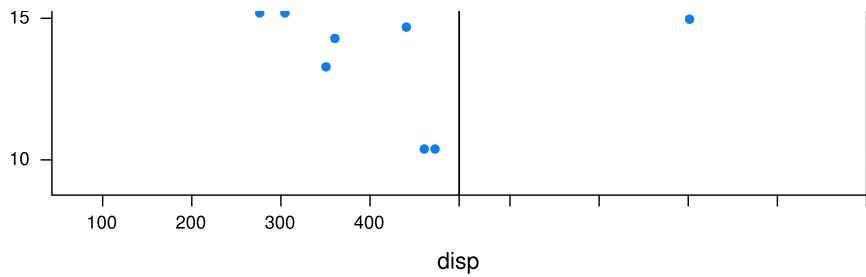
```
library(lattice)  
plot <- xyplot(mpg ~ disp | trans, data=mtcars)
```



# An Example Plot

What can I do with this plot **AFTER** it has been drawn ?



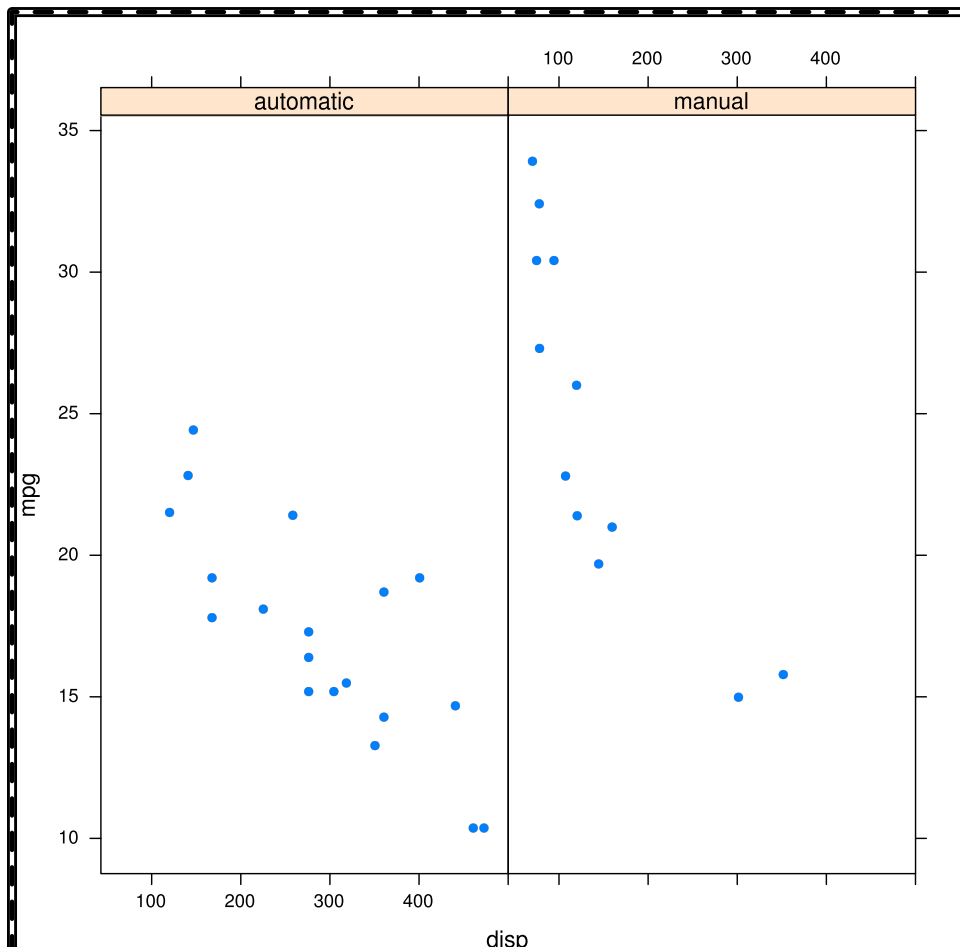


## Not Interested

```
xyplot(x,
  data,
  allow.multiple = is.null(groups) || outer,
  outer = !is.null(groups),
  auto.key = FALSE,
  aspect = "fill",
  panel = lattice.getOption("panel.xyplot"),
  prepanel = NULL,
  scales = list(),
  strip = TRUE,
  groups = NULL,
  xlab,
  xlim,
  ylab,
  ylim,
  drop.unused.levels = lattice.getOption("drop.unused.levels"),
  ...)
```

## Allow More Drawing

```
print(plot)
grid.rect(width=rep(.99, 3), height=rep(.99, 3),
  gp=gpar(lty=c("solid", "solid", "dashed"),
  lwd=c(7, 3, 3), col=c("black", "white"), fill=NA))
```

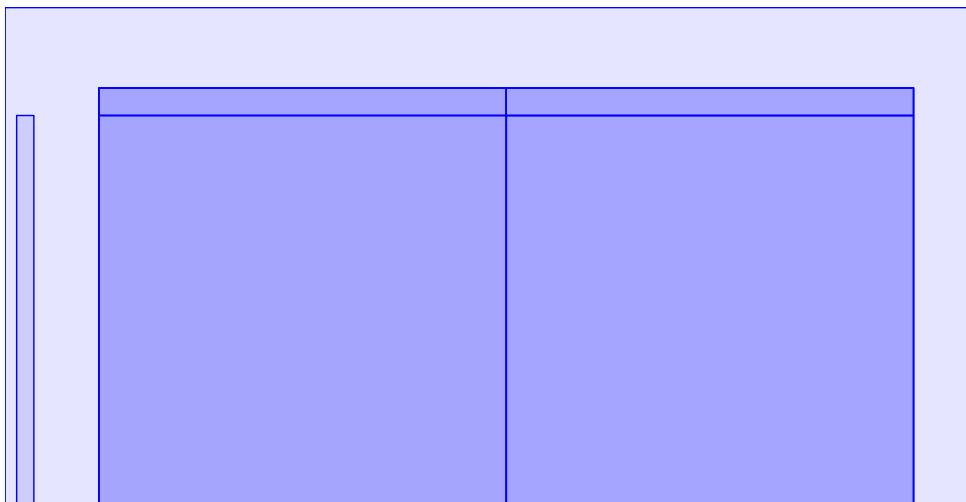


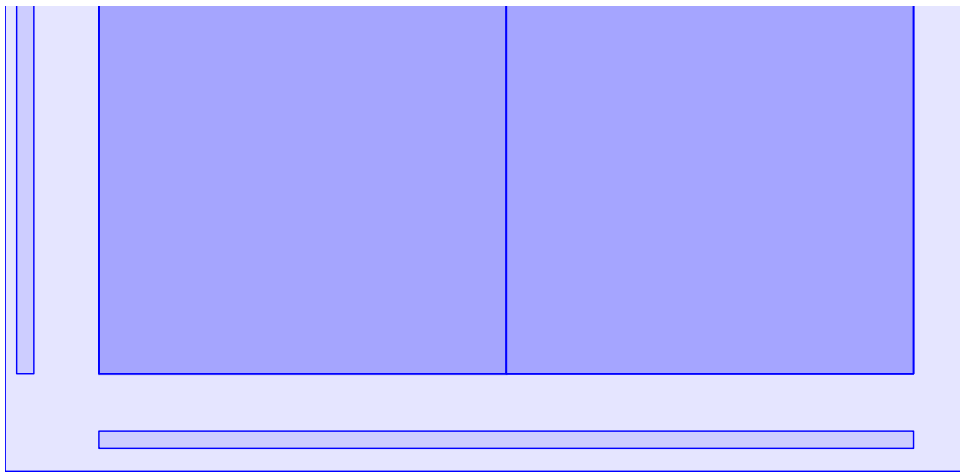


## Allow More Drawing

## Remember the Drawing Context

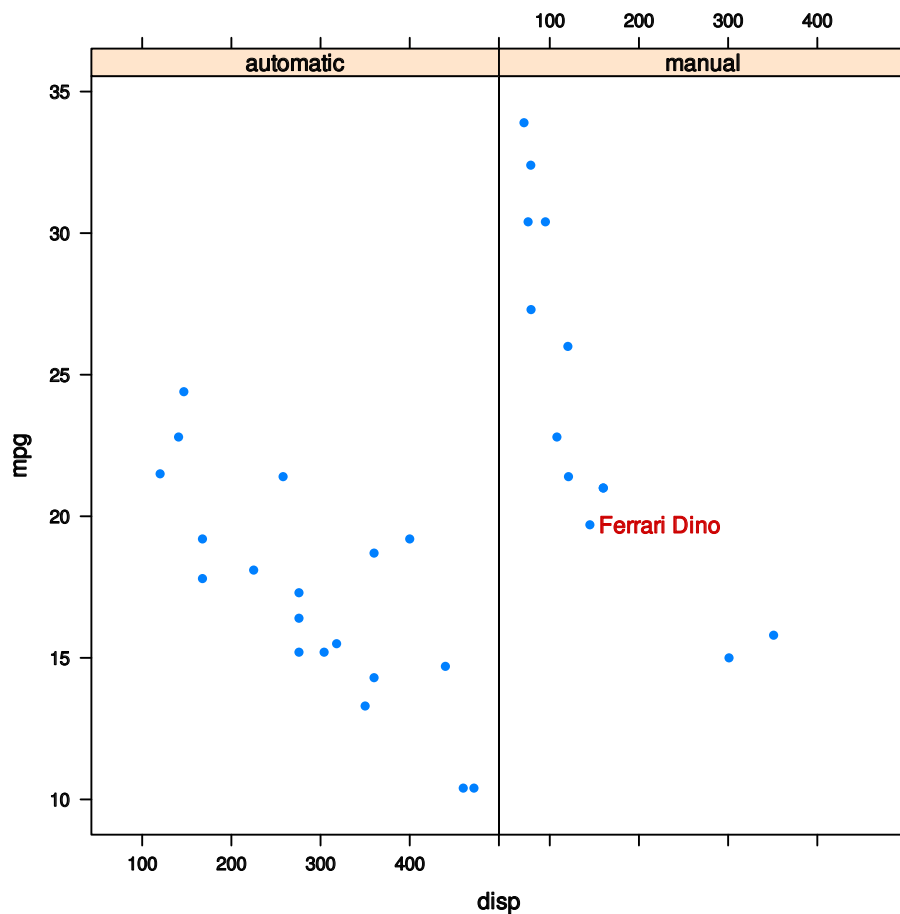
```
print(plot)  
showViewport()
```





## Allow Access to the Drawing Context

```
print(plot)
downViewport("plot_01.panel.1.2.vp")
grid.text("Ferrari Dino", x=155, y=19.7, just="left", default="native",
gp=gpar(col="red3", fontface="bold"))
```



## Allow Access to the Drawing Context

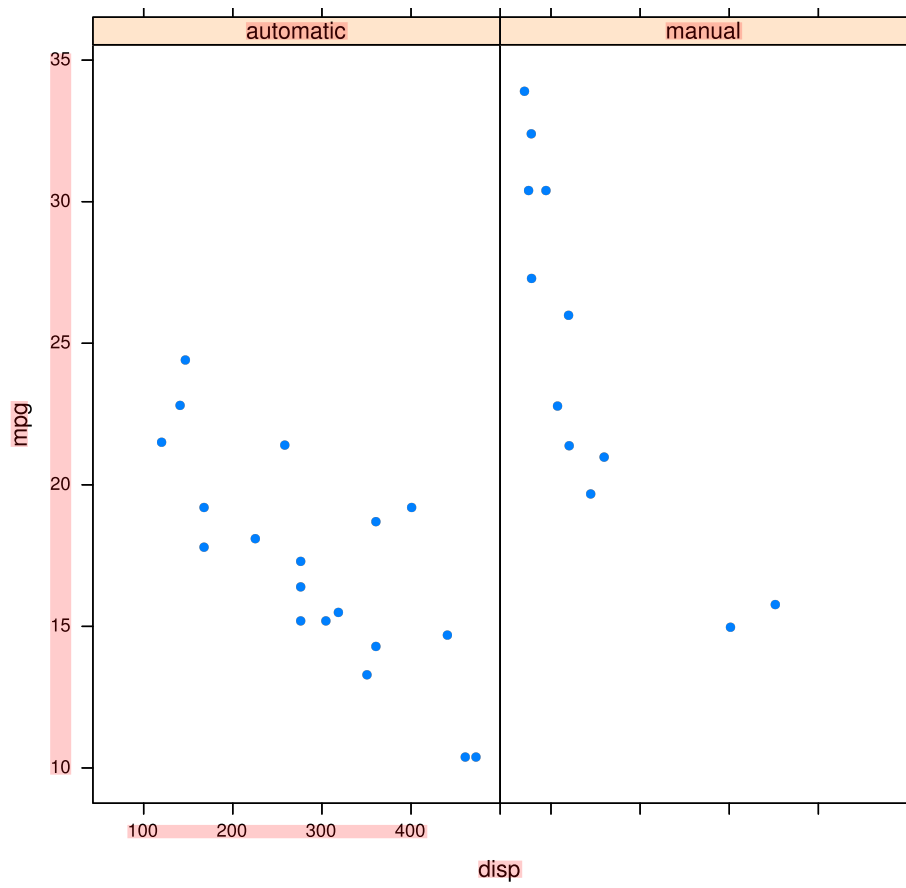
+

→

## Remember What was Drawn

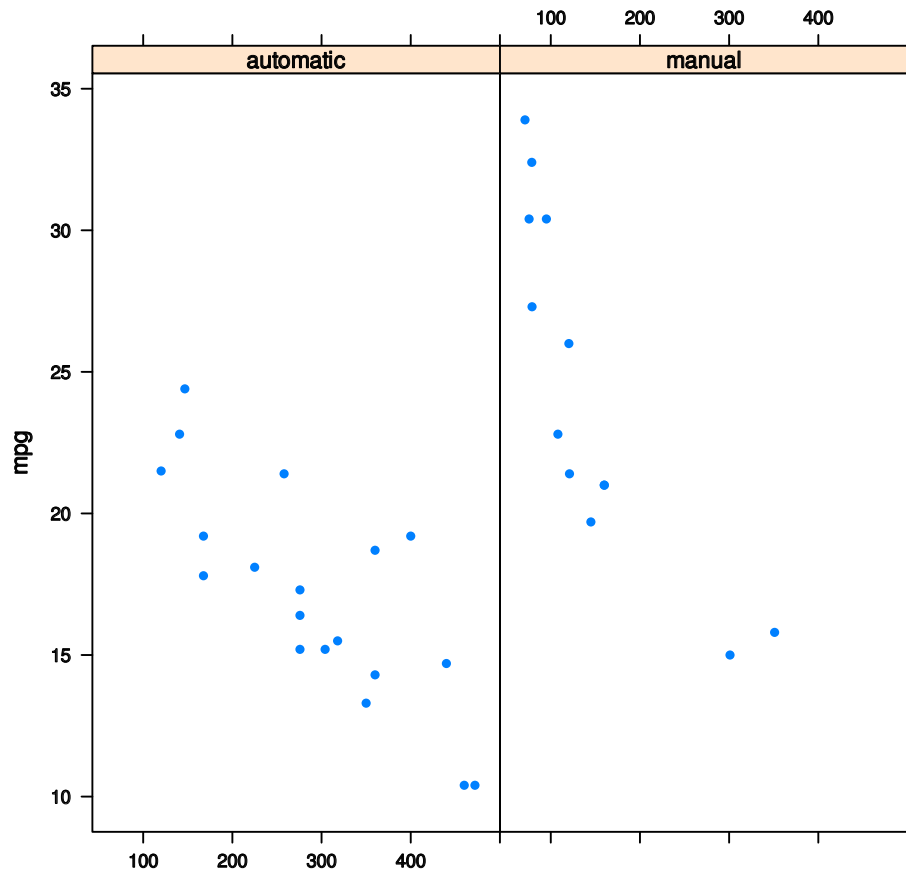
```
print(plot)  
showGroB()
```

100 200 300 400



## Allow Access to What was Drawn

```
print(plot)
grid.edit("plot_01.xlab", label="Engine Displacement")
```



## Allow Access to What was Drawn



## Allow Labels and Structure

```
grid.roundrect(gp=gpar(col=NA, fill=rgb(1,0,0,.5)), name="box")  
grid.text("Ferrari Dino", name="label")
```

Ferrari Dino



```
grid.ls()
```

```
box  
label
```

## Allow Labels and Structure

```
gTree(children=gList(roundrectGrob(gp=gpar(col=NA, fill=rgb(1,0,0,.5)),  
                        name="box"),  
                textGrob("Ferrari Dino", name="label")),  
       name="boxedLabel")
```

Ferrari Dino

```
grid.ls()
```

```
boxedLabel  
box  
label
```

## The Example Plot

## Provide Labels and Structure

```
print(plot)  
grid.ls(viewports=TRUE)
```

```
viewport[ROOT]  
rect[plot_01.background]  
viewport[plot_01.toplevel.vp]
```

```
viewport[plot_01.xlab.vp]
text[plot_01.xlab]
viewport[plot_01.ylab.vp]
text[plot_01.ylab]
viewport[plot_01.figure.vp]
```

## Export Labels and Structure

## Export Labels and Structure

R code ...

```
library(gridSVG)
print(plot)
grid.export()
```

SVG code ...

```
<svg>
  <rect id="plot_01.background.1.1"/>
  <g id="plot_01.toplevel.vp.1">
    <g id="plot_01.toplevel.vp:plot_01.xlab.vp.1">
      <text id="plot_01.xlab.1">
    ...
```

## Export Labels and Structure

## Export Labels and Structure

## Export Labels and Structure

R code ...

---

```
pushViewport(viewport(name="viewport"))
grid.polygon("ID:1")
grid.polygon("ID:2")
...
```

SVG code ...

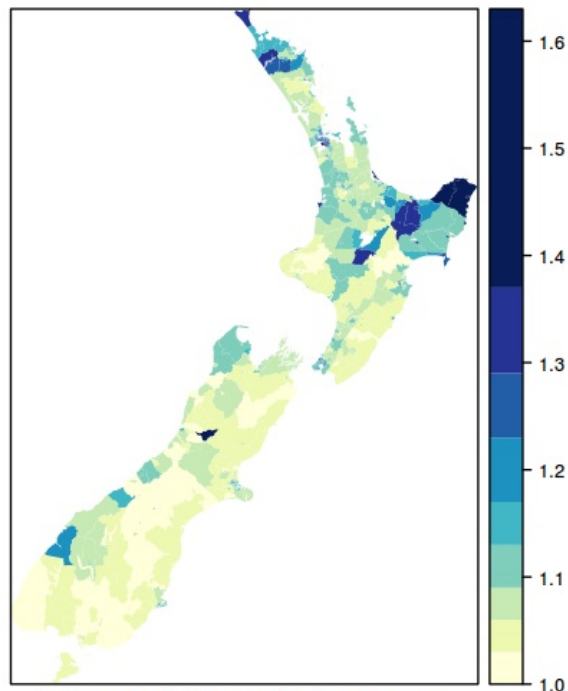
```
<svg>
  <g id="viewport">
    <polygon id="ID:1" .../>
    <polygon id="ID:2" .../>
  </g>
</svg>
```

Javascript code ...

```
svgRoot = root.getElementById("viewport");
...
```

## Don't be a Dead End

The Royal Society of New Zealand: Languages in Aotearoa New Zealand (**original**)



## Don't be a Dead End

The Royal Society of New Zealand: Languages in Aotearoa New Zealand (**recycled**)

# Don't be a Dead End

## Summary

- Don't be a Dead End
  - *grid*
    - Allow more drawing
    - Record the drawing context
    - Allow access to the drawing context
    - Record what was drawn
    - Allow access to what was drawn
    - Allow labels and structure
  - *lattice*
    - Provide a naming scheme
    - Provide structure
  - *gridSVG*
    - Permit labels and structure

## Acknowledgements

- lattice
- RSNZ and Paul Behrens
- SVGPan.js
- Bickler and the other dude for brokenness example
- Steve Mosher and Google map
- Edna Lu and David Scott for metaplot