
S Graphics

Paul Murrell

`paul@stat.auckland.ac.nz`

The University of Auckland

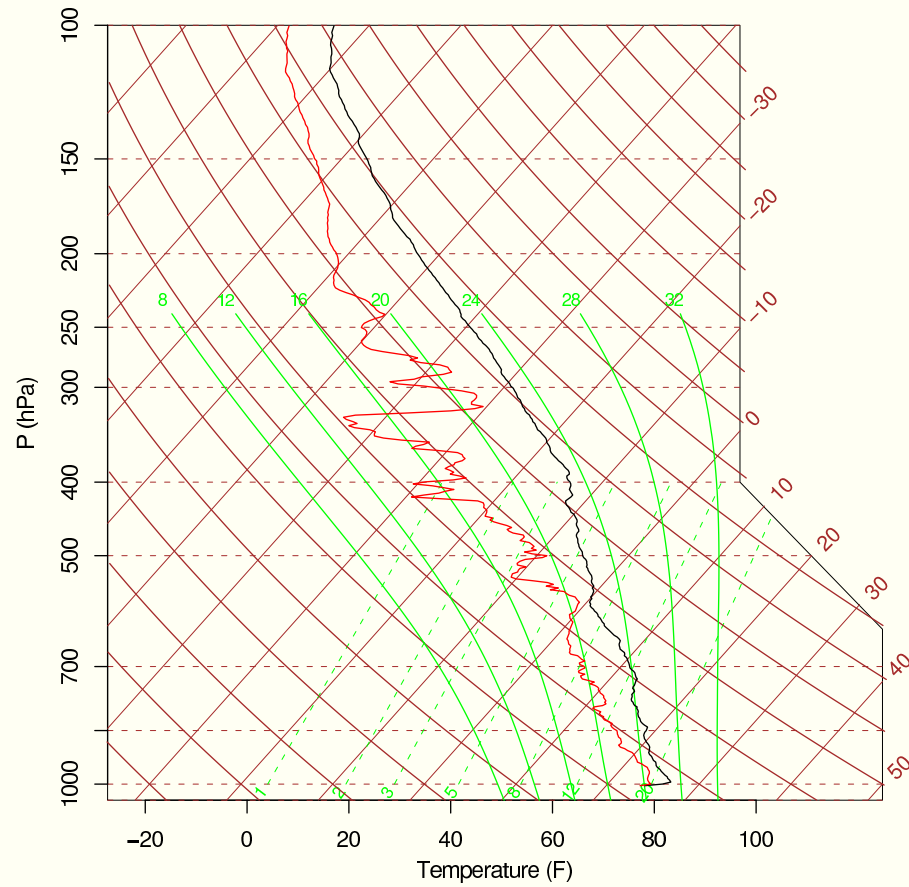
Overview

- Things that are covered
 - Command-line Graphics
 - Customising Traditional **S** Graphics
 - Customising Traditional Trellis Graphics
- Things that are not covered
 - GUI Graphics
 - Editable Graphics
 - Traditional High-level Graphics functions

Customising Graphics

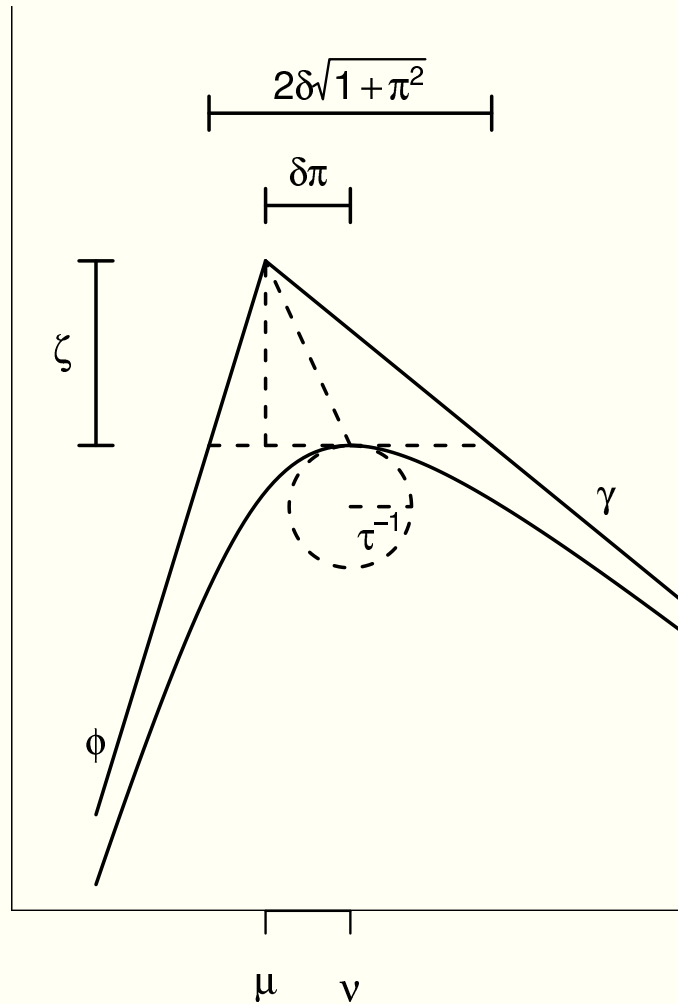
- Why customize?
 - Annotation
 - Diagrams
 - New Plots
- **S** Graphics Fundamentals
 - Graphics Regions and Coordinate Systems
 - Directing Graphics Output
 - Producing Graphics Output
- Adding to Existing Plots
- Plots from First Principles

Annotation



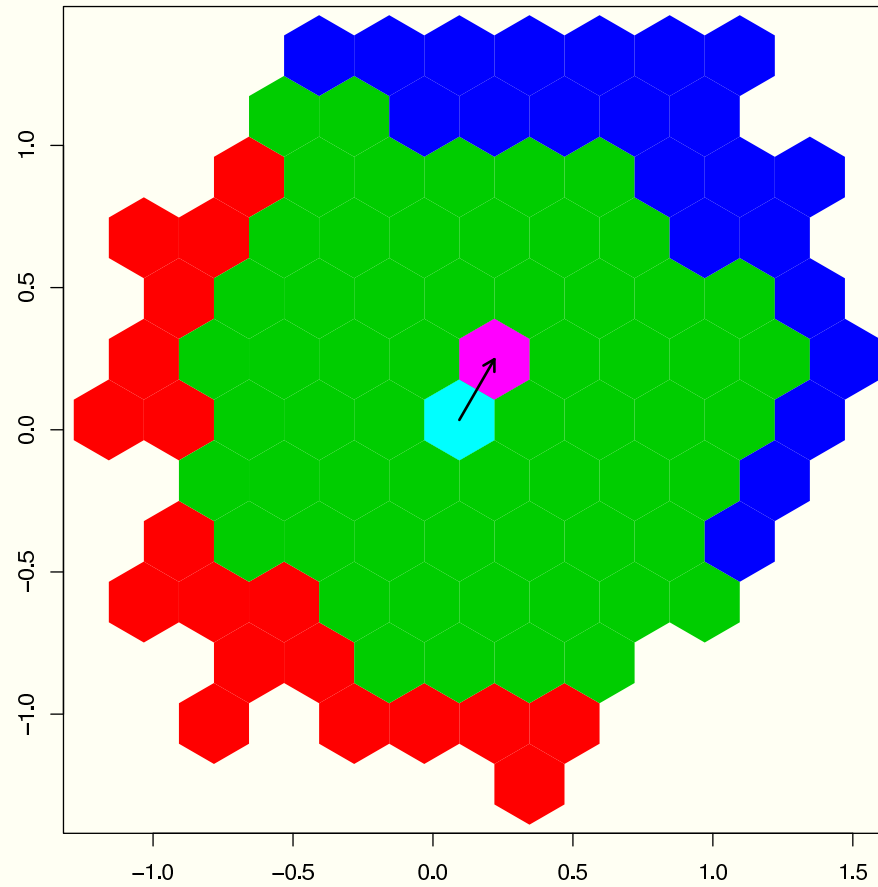
Tim Hoar, Eric Gilleland, Doug Nychka

Diagrams



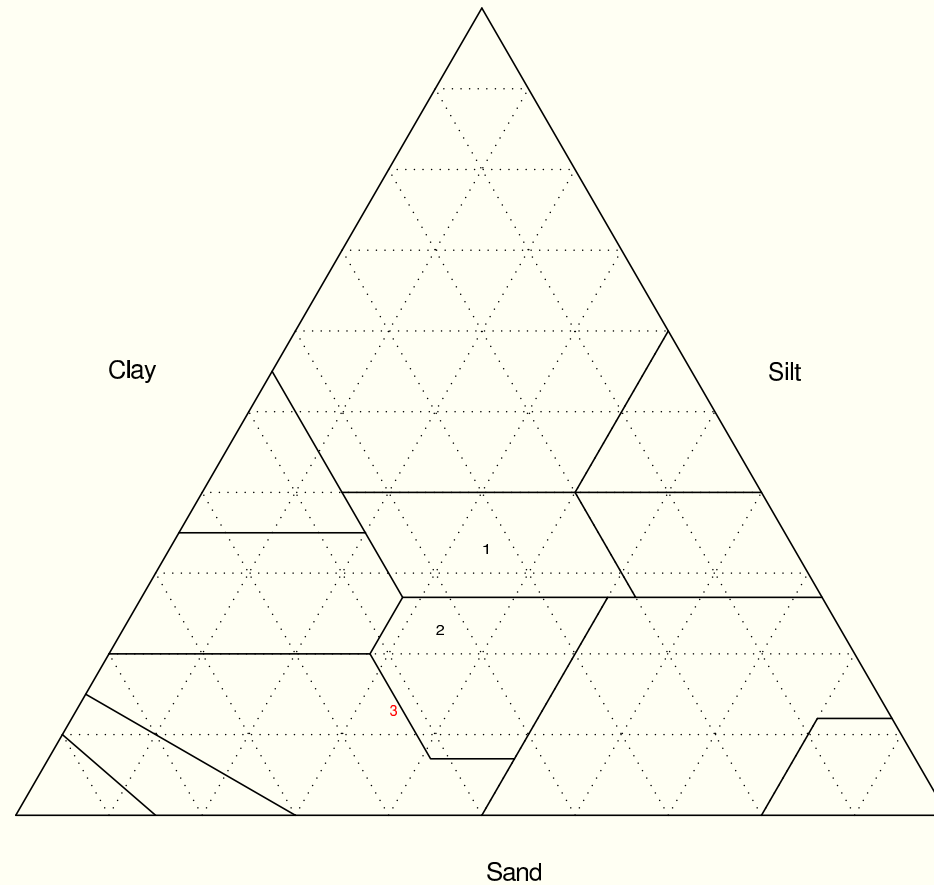
David Scott, The University of Auckland

New Plots - 1



The Bioconductor Project

New Plots - 2

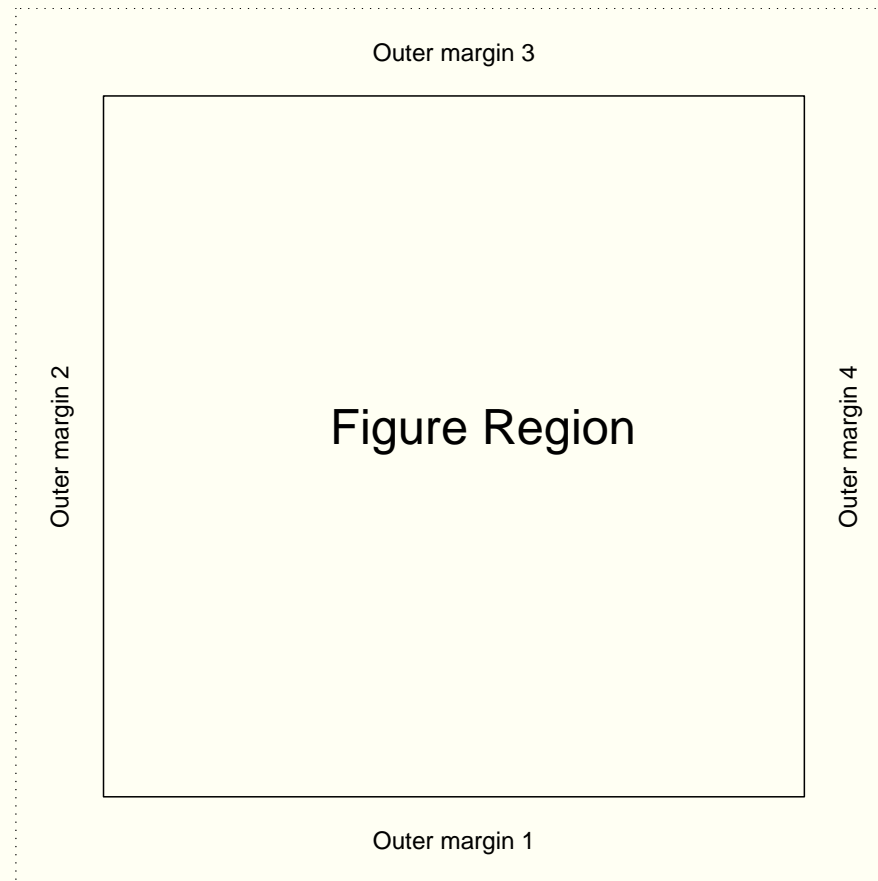


Corey A. Moffet, Texas Tech University

S Graphics Fundamentals

- Graphics Regions and Coordinate Systems
 - Outer Margins
 - Figure Regions
 - Figure Margins
 - Plot Regions
- Directing Graphics Output
 - Which graphics functions to use
- Producing Graphics Output
 - Graphical parameters

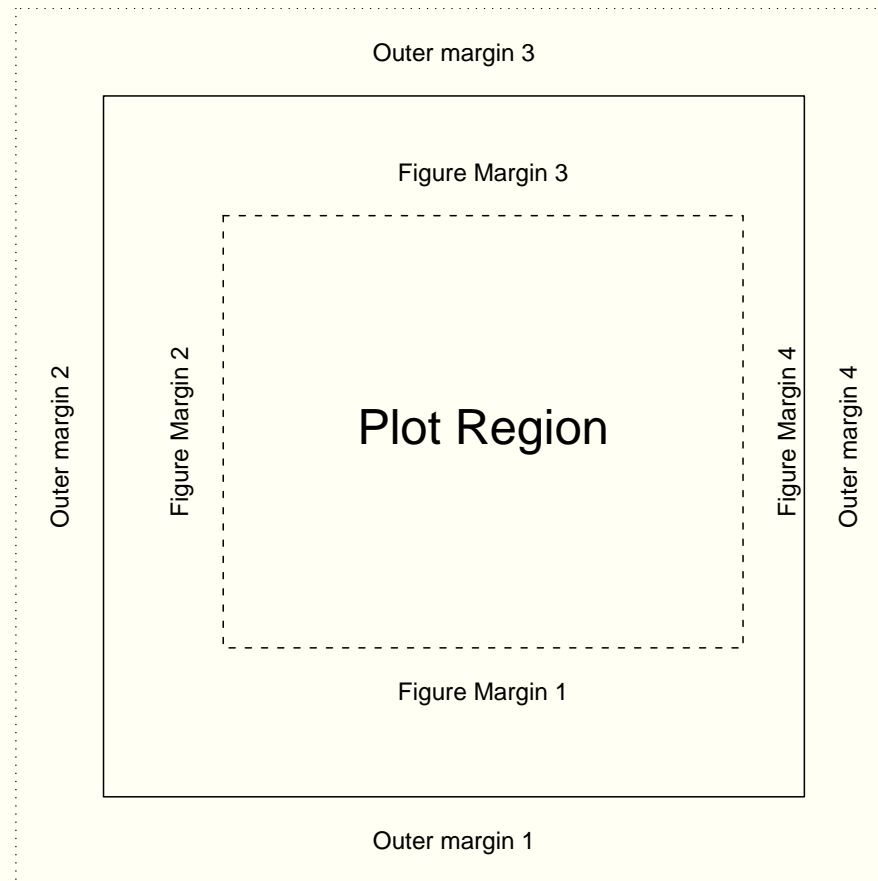
Outer Margins and Figure Region



```
par(oma=c(0, 0, 0, 0), omi=)
```

```
par(mfrow=c(1, 1), mfcoll=c(1, 1), fig=, fin=)
```

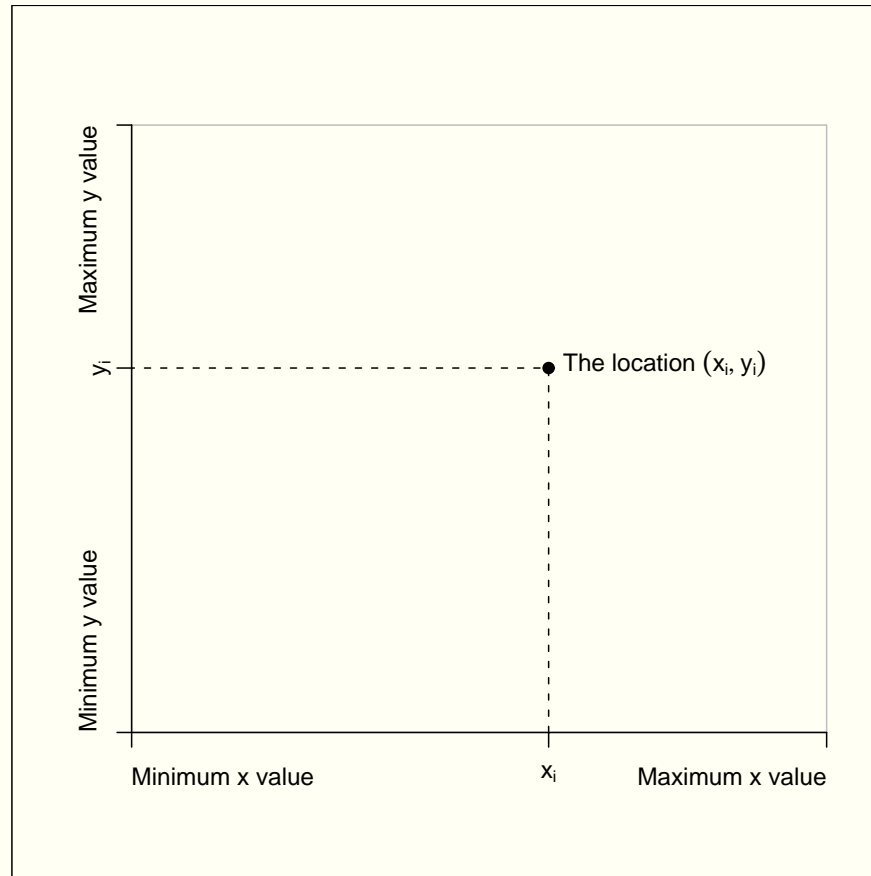
Figure Margins and Plot Region



```
par(mar=c(5.1, 4.1, 4.1, 2.1), mai=)
```

```
par(pty="m", pin=, plt=)
```

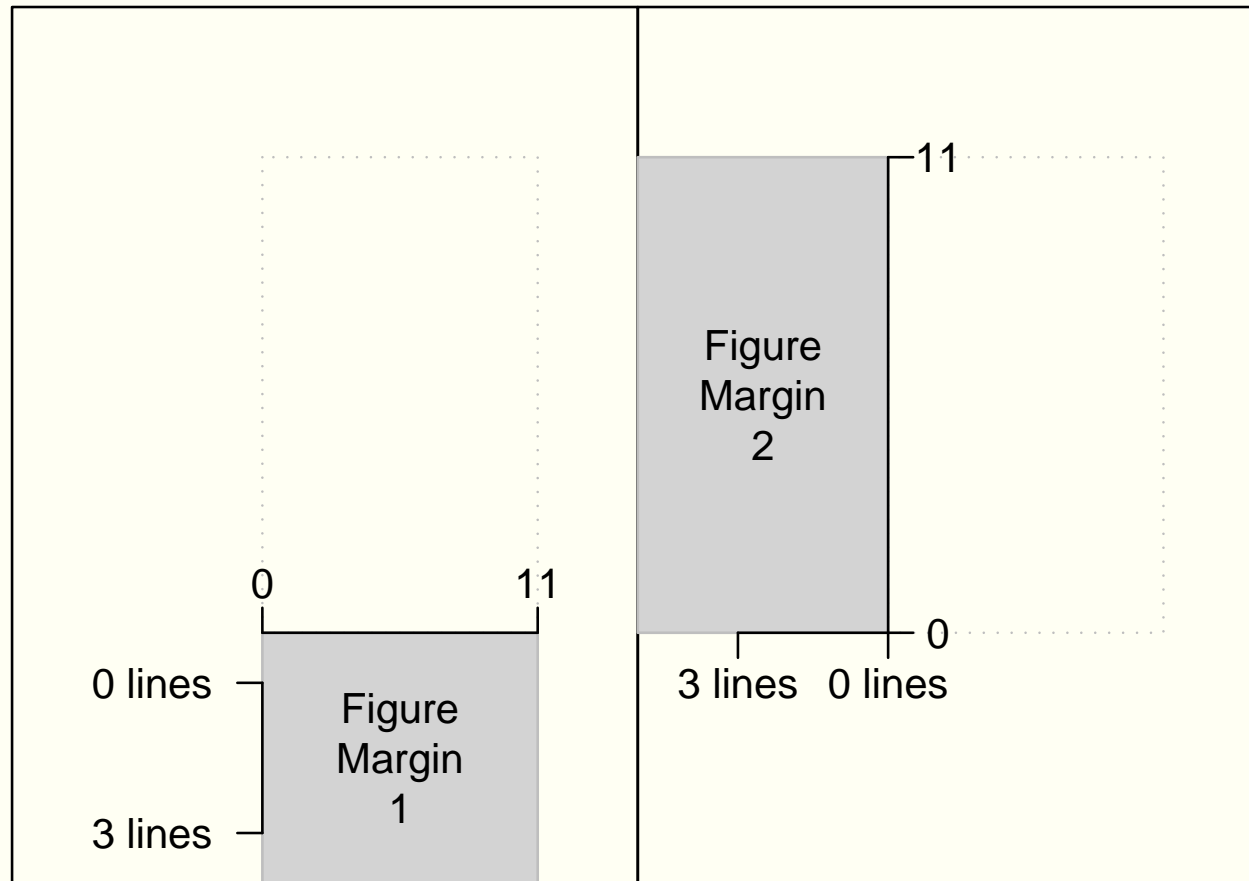
User Coordinates



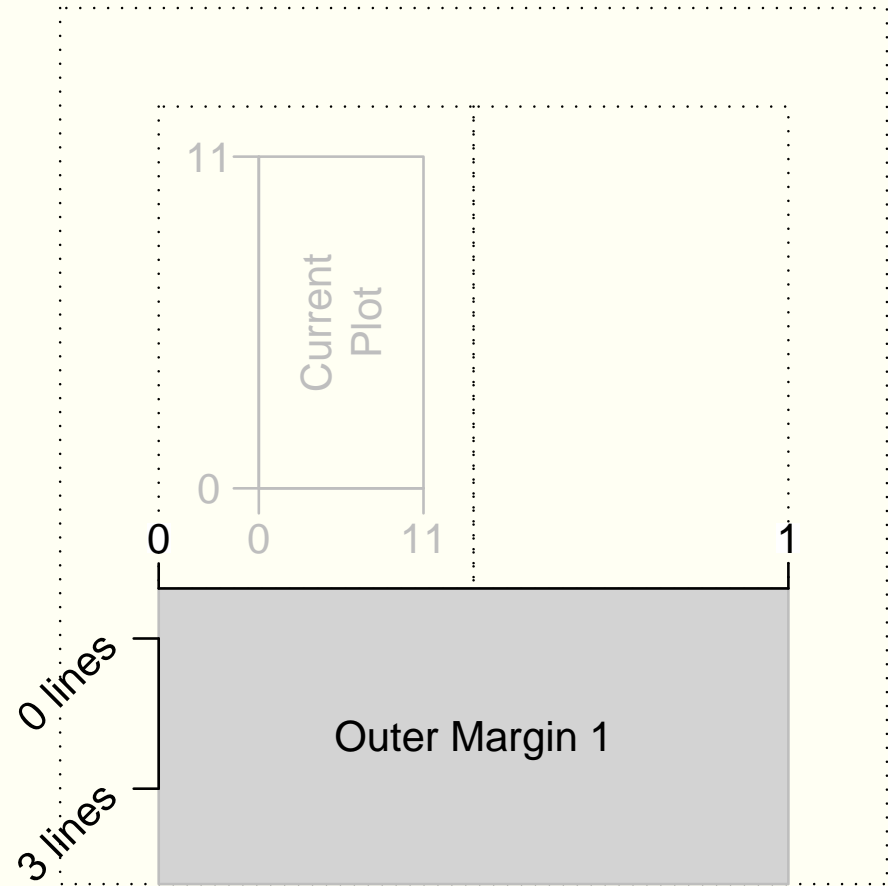
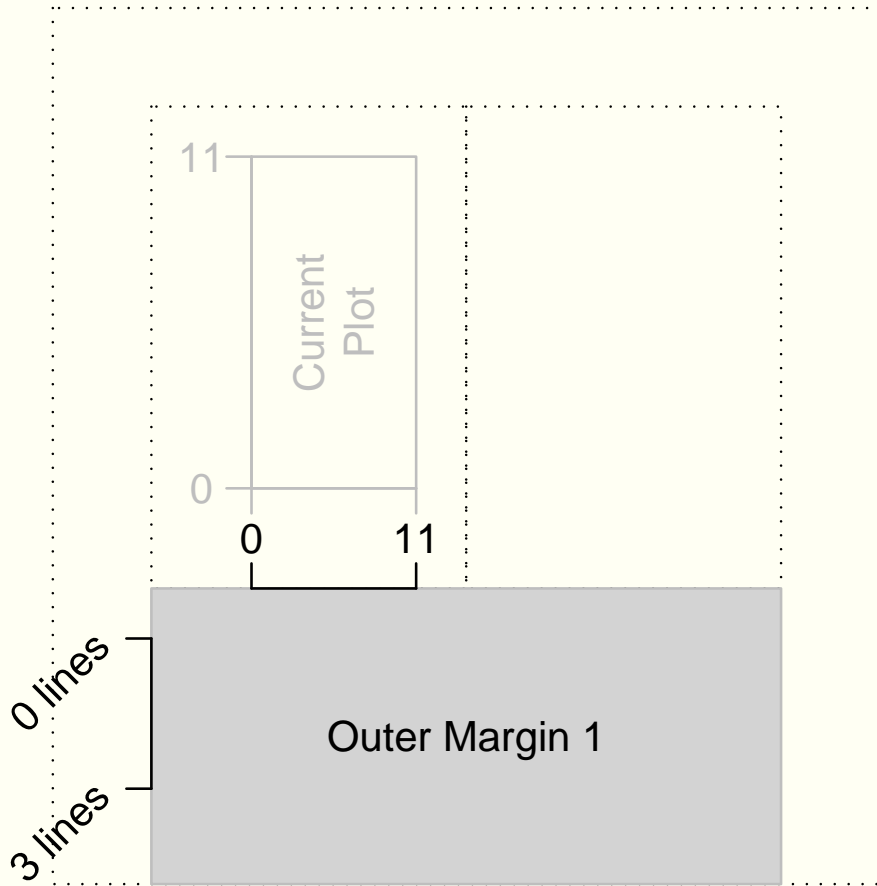
```
<plot.function>( ..., xlim=, ylim=)
```

```
par(xaxs="r", yaxs="r")
```

Figure Margin Coordinates



Outer Margin Coordinates



Directing Graphics Output

Plot Region	Figure Margins	Outer Margins
<code>text()</code>	<code>mtext()</code>	<code>mtext()</code>
<code>points()</code>	<code>axis()</code>	
<code>lines()</code>		
<code>arrows()</code>		
<code>polygon()</code>		
<code>segments()</code>		
<code>box()</code>		
<code>abline()</code>		

Graphical Parameters

- Permanent settings

`par (<param>=)`

- Temporary settings

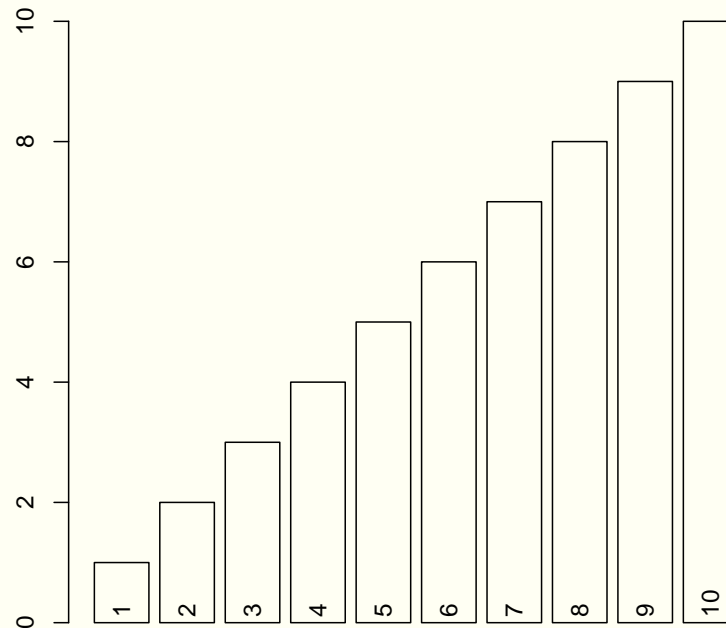
`<plot.function> (. . . , <param>=)`

<code>col</code>	colour of lines, text, ...
<code>lwd</code>	line width
<code>lty</code>	line type
<code>font</code>	font face (plain, bold, italic)
<code>pch</code>	type of plotting symbol
<code>srt</code>	string rotation

Adding to Existing Plots

● User Coordinates are not always obvious

```
> midpts <- barplot(1:5, density = -1)  
> text(midpts, 0.1, 1:5, srt = 90, adj = 0)
```



Adding to Existing Plots

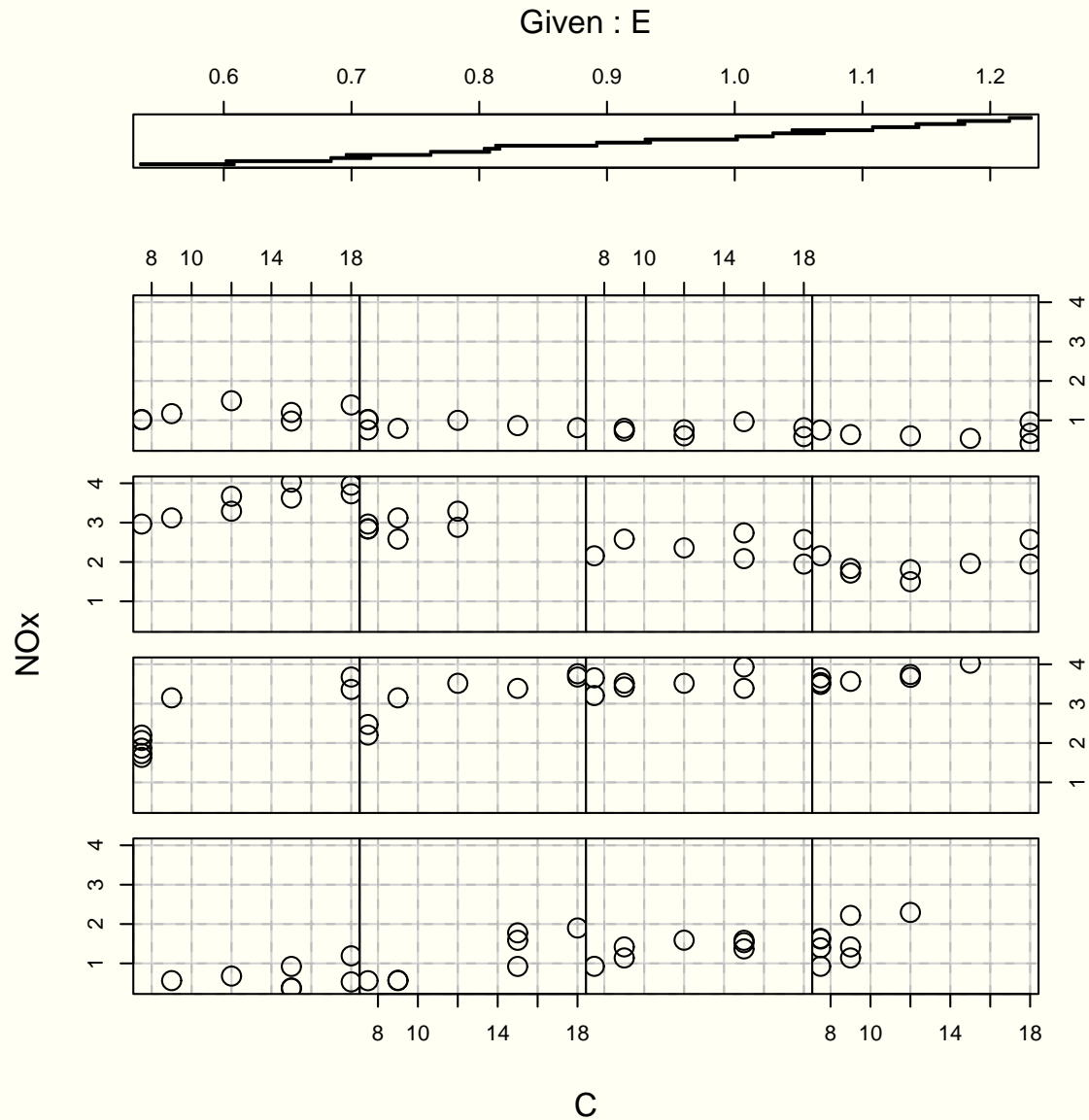
- Querying the current region and coordinate settings

```
> plot(0:1, 1:2)
> par("usr")
[1] -0.04  1.04  0.96  2.04
```

- Some complex plots revert region and coordinate settings (panel functions)

```
> E.intervals <- co.intervals(ethanol$E, 16, 0.25)
> coplot(NOx ~ C | E, given.values = E.intervals, data = ethanol,
+        panel = function(x, y, col, pch) {
+            points(x, y);
+            axis(1, tck=1, lty=2, labels=F);
+            axis(2, tck=1, lty=2, labels=F) })
```

Adding to Existing Plots



Plots from First Principles

- Create regions and coordinate systems

```
> plot(0, type="n", xlim=c(0, 10), ylim=c(0,10),  
+      axes=F, xlab="", ylab="")
```

- Draw data symbols in plot region

```
> points(1:10)
```

- Draw axes and labels in the figure margins

```
> box()  
> axis(1)  
> axis(2)  
> mtext("1:10", side=2, line=3)
```

Plots from First Principles

