R grid Graphics

Paul Murrell

The University of Auckland
New Zealand

Overview

- A user's view of statistical graphics
- A developer's view of statistical graphics
- Making the transition via `grid`

Overview

- A user's view of statistical graphics
- A developer's view of statistical graphics
- Making the transition via `grid`

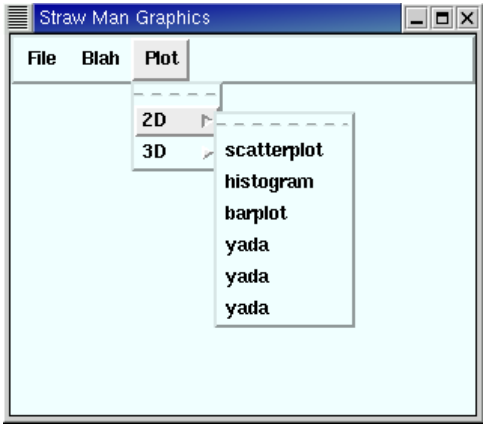
Overview

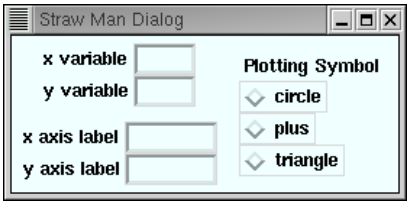
- A user's view of statistical graphics
- A developer's view of statistical graphics
- Making the transition via `grid`

Overview

- A user's view of statistical graphics
- A developer's view of statistical graphics
- Making the transition via `grid`

A User's View


Straw Man Graphics
File
Blah
Plot
2D
3D
scatterplot
histogram
barplot
yada
yada
yada


Straw Man Dialog
x variable
y variable
x axis label
y axis label
Plotting Symbol
circle
plus
triangle

A User's View

- Advantages
  - High-level conceptual view
- Disadvantages
  - Eventually discover something impossible
  - Unable to see what is possible

A User's View

- Advantages
  - High-level conceptual view
- Disadvantages
  - Eventually discover something impossible
  - Unable to see what is possible

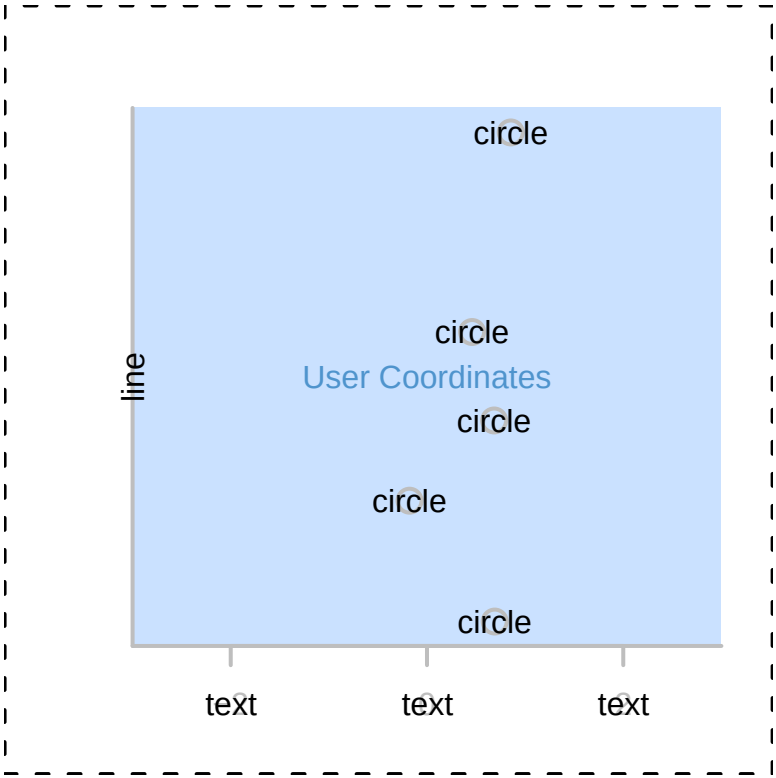
A User's View

- Advantages
  - High-level conceptual view
- Disadvantages
  - Eventually discover something impossible
  - Unable to see what is possible

A User's View

- Advantages
  - High-level conceptual view
- Disadvantages
  - Eventually discover something impossible
  - Unable to see what is possible

A Developer's View



A Developer's View

- Disadvantages
  - Look funny, dress funny, have no life, have no friends
  - Technical, low-level view
- Advantages
  - Almost anything is possible
  - More things appear possible

A Developer's View

- Disadvantages
  - Look funny, dress funny, have no life, have no friends
  - Technical, low-level view
- Advantages
  - Almost anything is possible
  - More things appear possible

A Developer's View

- Disadvantages
  - Look funny, dress funny, have no life, have no friends
  - Technical, low-level view
- Advantages
  - Almost anything is possible
  - More things appear possible

A Developer's View

- Disadvantages
  - Look funny, dress funny, have no life, have no friends
  - Technical, low-level view
- Advantages
  - Almost anything is possible
  - More things appear possible

A Developer's View

- Disadvantages
  - Look funny, dress funny, have no life, have no friends
  - Technical, low-level view
- Advantages
  - Almost anything is possible
  - More things appear possible

The Transition from User to Developer

The S Language (S-Plus and R) is designed to blur the distinction and ease the transition between users and developers of statistical analysis and graphics software.

- High-level plotting functions (e.g., `plot()`)
- Low-level plotting functions (e.g., `lines()`, `text()`)

The Transition from User to Developer

The S Language (S-Plus and R) is designed to blur the distinction and ease the transition between users and developers of statistical analysis and graphics software.

- High-level plotting functions (e.g., `plot()`)
- Low-level plotting functions (e.g., `lines()`, `text()`)

The Transition from User to Developer

The S Language (S-Plus and R) is designed to blur the distinction and ease the transition between users and developers of statistical analysis and graphics software.

- High-level plotting functions (e.g., `plot()`)
- Low-level plotting functions (e.g., `lines()`, `text()`)

The Transition from User to Developer

In the S tradition, the `grid` add-on package for R is an attempt to provide users with a developer's view of statistical graphics ...

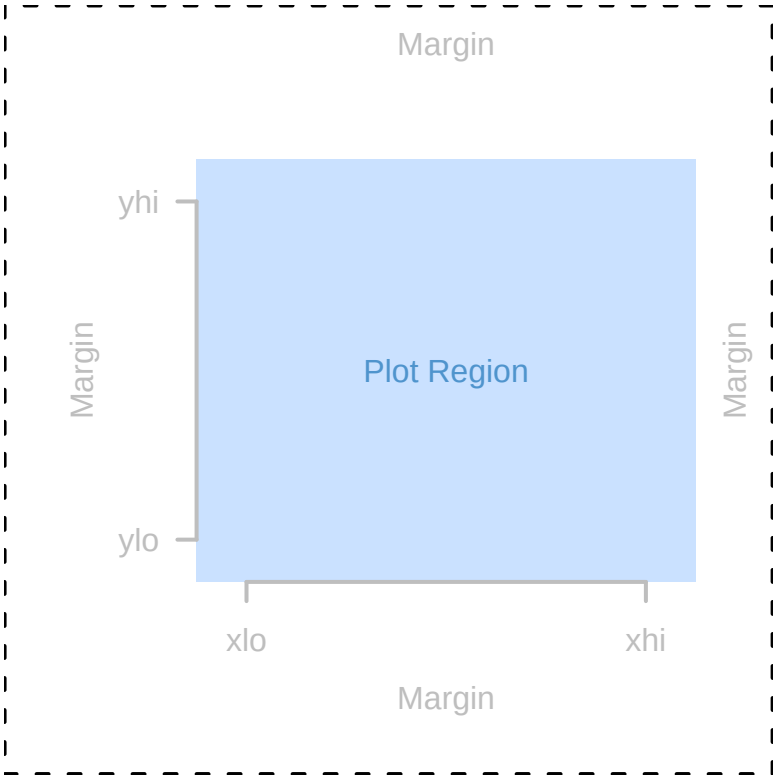
... without losing all your friends.

The Transition from User to Developer

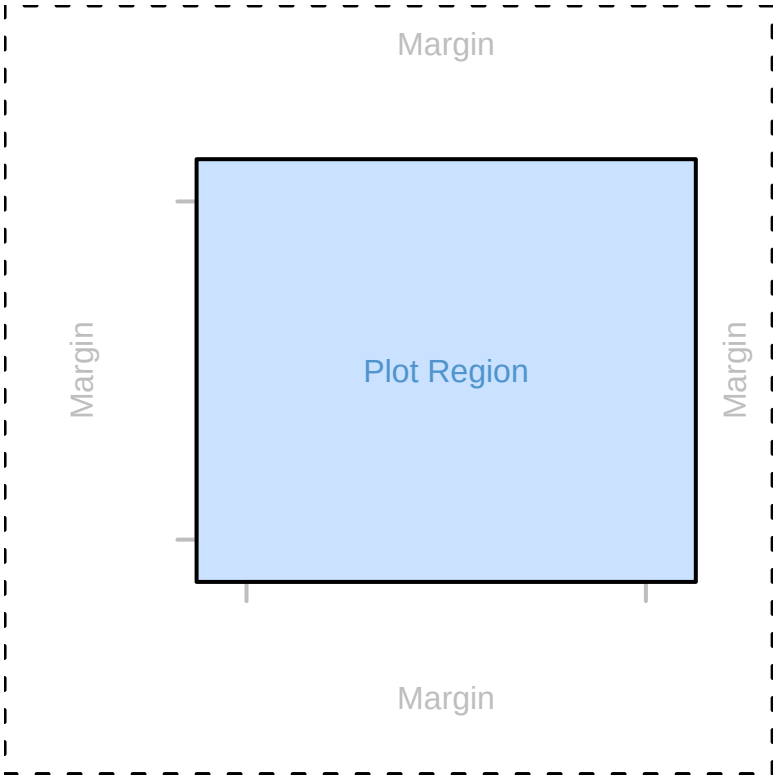
In the S tradition, the `grid` add-on package for R is an attempt to provide users with a developer's view of statistical graphics ...

... without losing all your friends.

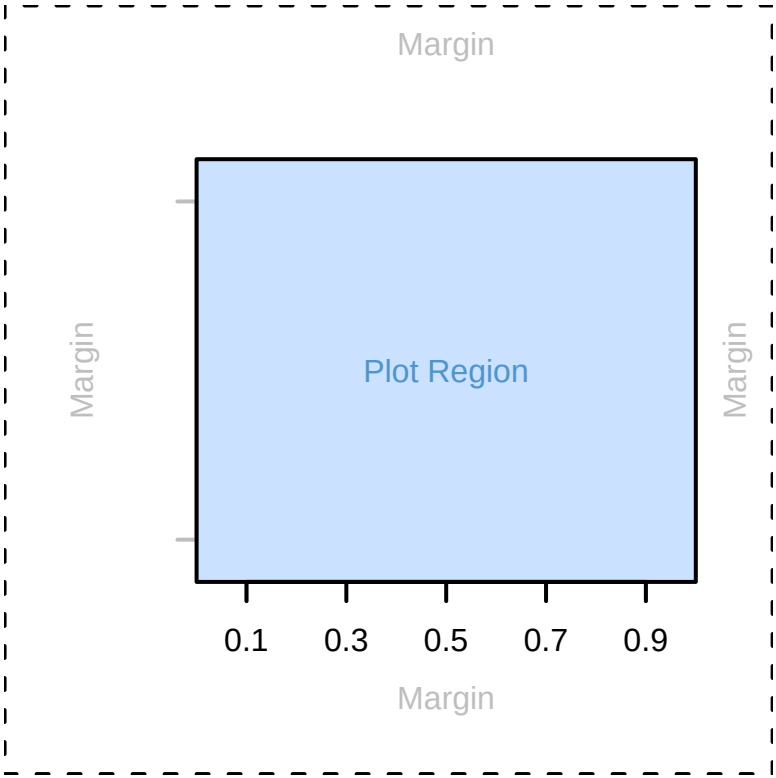
A Developer Constructs a Scatterplot



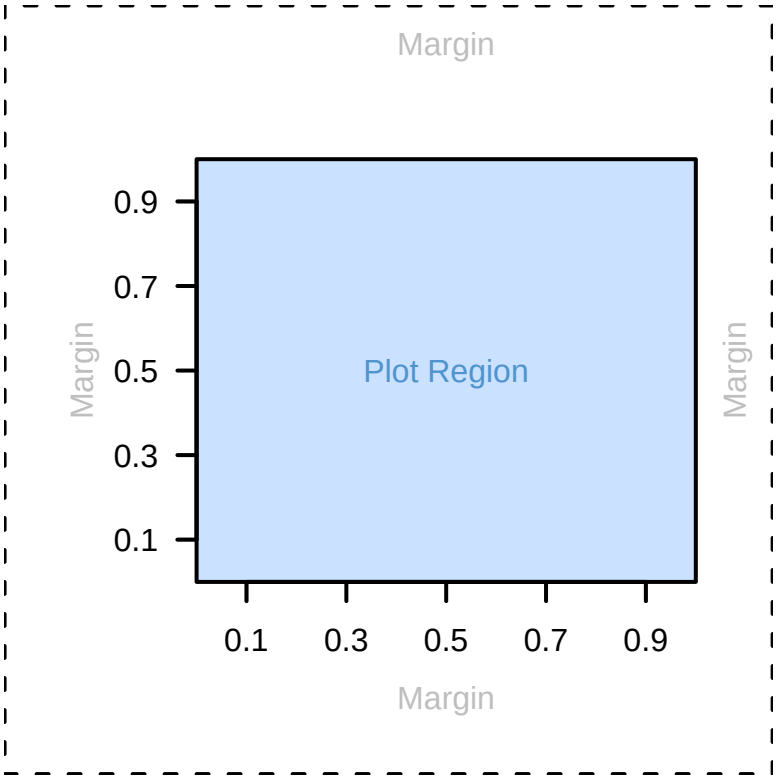
A Developer Constructs a Scatterplot



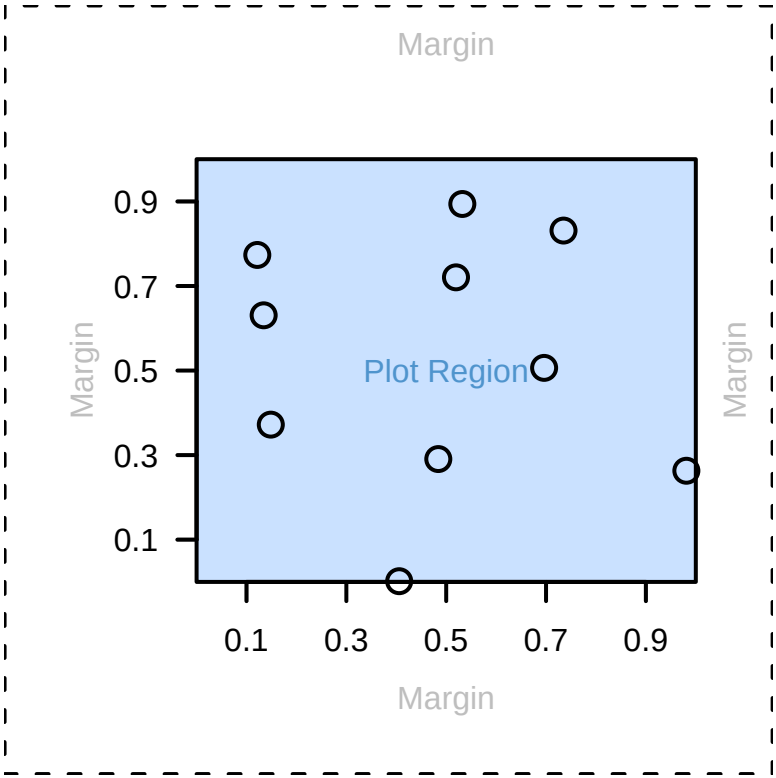
A Developer Constructs a Scatterplot



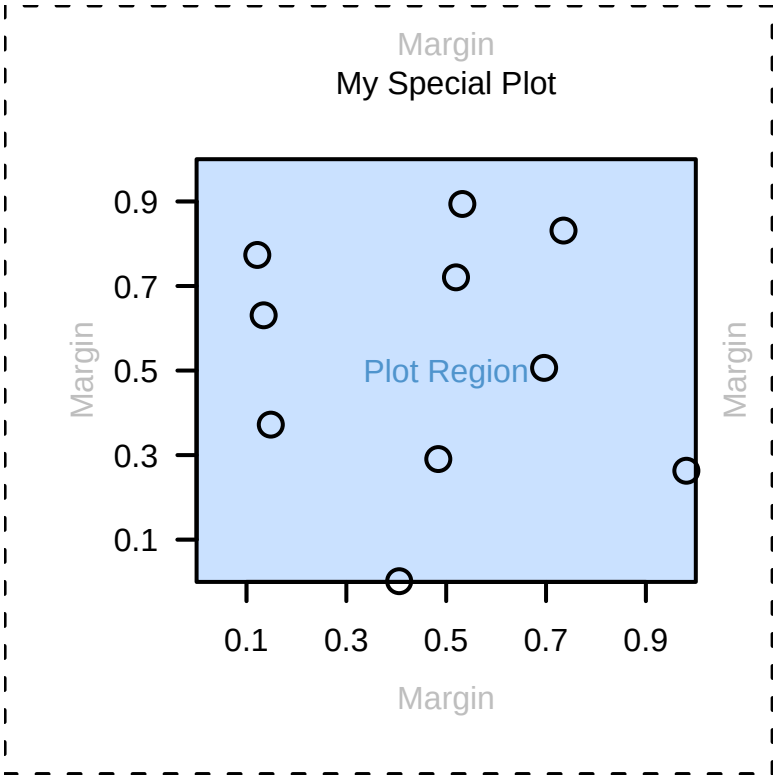
A Developer Constructs a Scatterplot



A Developer Constructs a Scatterplot



A Developer Constructs a Scatterplot



Constructing a Scatterplot in grid

What is R?

- A language and environment for statistical computing and graphics

A sample session:

```
shell$ R
R> 1 + 1
[1] 2
R> if (1 + 1 == 2) TRUE else FALSE
[1] TRUE
R> plot(1:10, 1:10)
```

Constructing a Scatterplot in grid

What is R?

- A language and environment for statistical computing and graphics

A sample session:

```
shell$ R
R> 1 + 1
[1] 2
R> if (1 + 1 == 2) TRUE else FALSE
[1] TRUE
R> plot(1:10, 1:10)
```

Constructing a Scatterplot in grid

What is R?

- A language and environment for statistical computing and graphics

A sample session:

```
shell$ R
R> 1 + 1
[1] 2
R> if (1 + 1 == 2) TRUE else FALSE
[1] TRUE
R> plot(1:10, 1:10)
```

Constructing a Scatterplot in grid

What is R?

- A language and environment for statistical computing and graphics

A sample session:

```
shell$ R
R> 1 + 1
[1] 2
R> if (1 + 1 == 2) TRUE else FALSE
[1] TRUE
R> plot(1:10, 1:10)
```

Constructing a Scatterplot in grid

What is R?

- A language and environment for statistical computing and graphics

A sample session:

```
shell$ R
R> 1 + 1
[1] 2
R> if (1 + 1 == 2) TRUE else FALSE
[1] TRUE
R> plot(1:10, 1:10)
```

Constructing a Scatterplot in grid

What is R?

- A language and environment for statistical computing and graphics

A sample session:

```
shell$ R
R> 1 + 1
[1] 2
R> if (1 + 1 == 2) TRUE else FALSE
[1] TRUE
R> plot(1:10, 1:10)
```

Constructing a Scatterplot in grid

What is R?

- A language and environment for statistical computing and graphics

A sample session:

```
shell$ R
R> 1 + 1
[1] 2
R> if (1 + 1 == 2) TRUE else FALSE
[1] TRUE
R> plot(1:10, 1:10)
```

Constructing a Scatterplot in grid

What is R?

- A language and environment for statistical computing and graphics

A sample session:

```
shell$ R
R> 1 + 1
[1] 2
R> if (1 + 1 == 2) TRUE else FALSE
[1] TRUE
R> plot(1:10, 1:10)
```

Constructing a Scatterplot in grid

What is grid?

- An add-on package for R

A sample session:

```
shell$ R
R> library(grid)
R> grid.lines()
```

Constructing a Scatterplot in grid

What is grid?

- An add-on package for R

A sample session:

```
shell$ R
R> library(grid)
R> grid.lines()
```

Constructing a Scatterplot in grid

What is grid?

- An add-on package for R

A sample session:

```
shell$ R
R> library(grid)
R> grid.lines()
```

Constructing a Scatterplot in grid

What is grid?

- An add-on package for R

A sample session:

```
shell$ R
R> library(grid)
R> grid.lines()
```

Constructing a Scatterplot in grid

What is grid?

- An add-on package for R

A sample session:

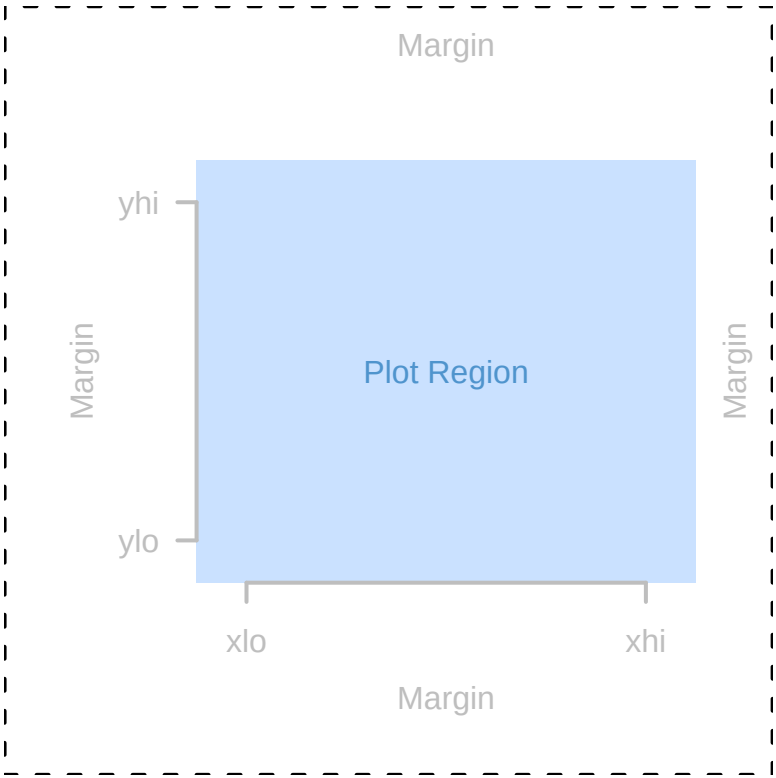
```
shell$ R
R> library(grid)
R> grid.lines()
```

`grid` Viewports

- `grid` viewports define a rectangular region and associate several coordinate systems with the region.

```
R> push.viewport(plotViewport(c(5, 5, 4, 2)))
```

grid Viewports

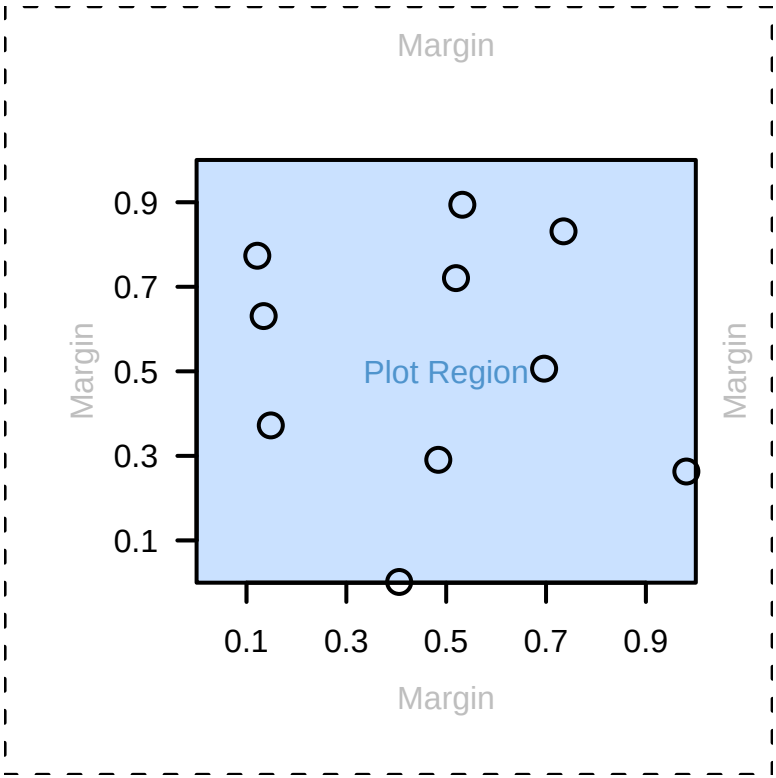


grid Graphical Objects

- grid provides various objects for adding to an image; these are always drawn relative to the current viewport.

```
R> grid.rect()
R> grid.xaxis(at=seq(.1, .9, length=5))
R> grid.yaxis(at=seq(.1, .9, length=5))
R> grid.points(x, y)
```

`grid` Graphical Objects

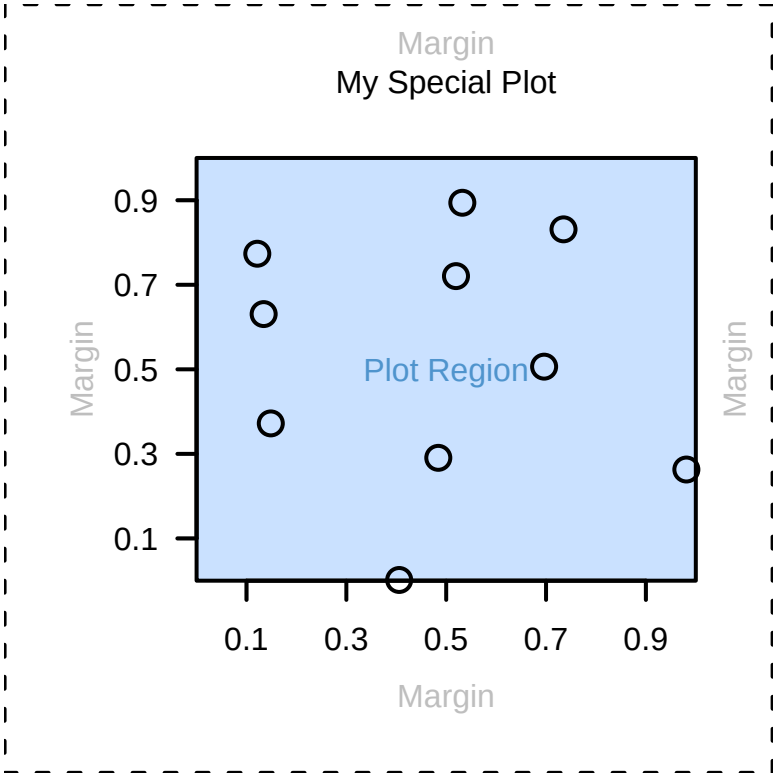


grid Units

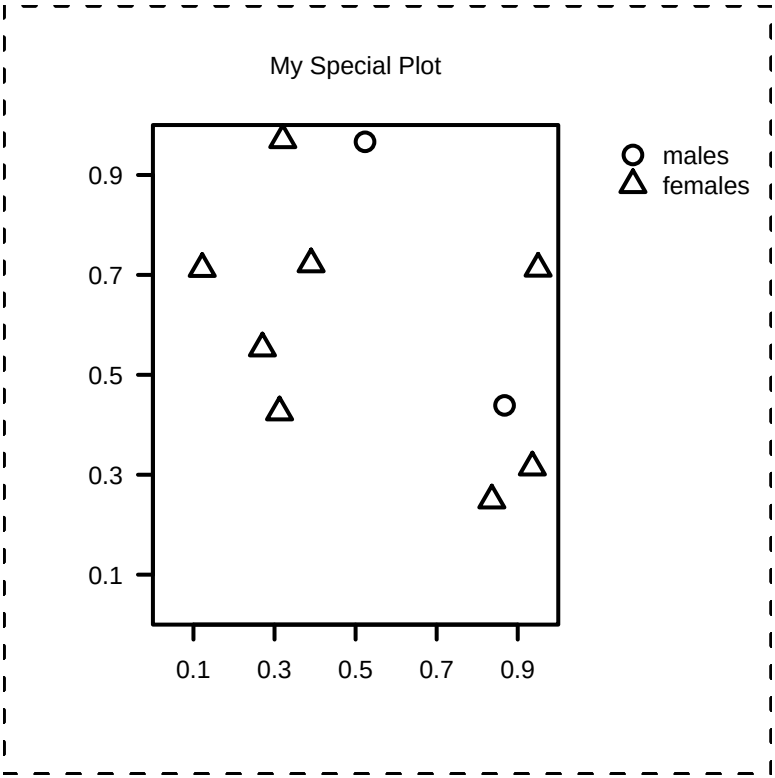
- grid provides several coordinate systems within every viewport; unit objects associate a value with a particular coordinate system.

```
R> grid.text("My Special Plot",
             y=unit(1, "npc") + unit(2, "lines"))
```

grid Units


Margin
My Special Plot
0.9
0.7
0.5
0.3
0.1
Margin
Plot Region
Margin
0.1
0.3
0.5
0.7
0.9
Margin

A Developer Constructs a Legend

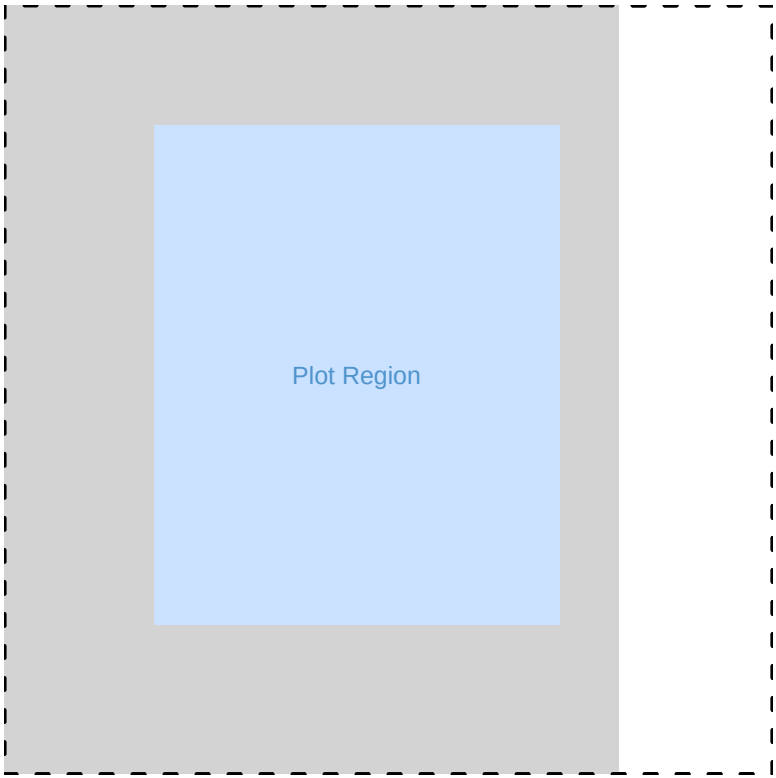


A Developer Constructs a Legend

- `grid` viewports can be nested within each other.

```
R> push.viewport(viewport(x=0, width=0.8,
                          just="left"))
R> push.viewport(plotViewport(c(5, 5, 4, 2)))
```

A Developer Constructs a Legend

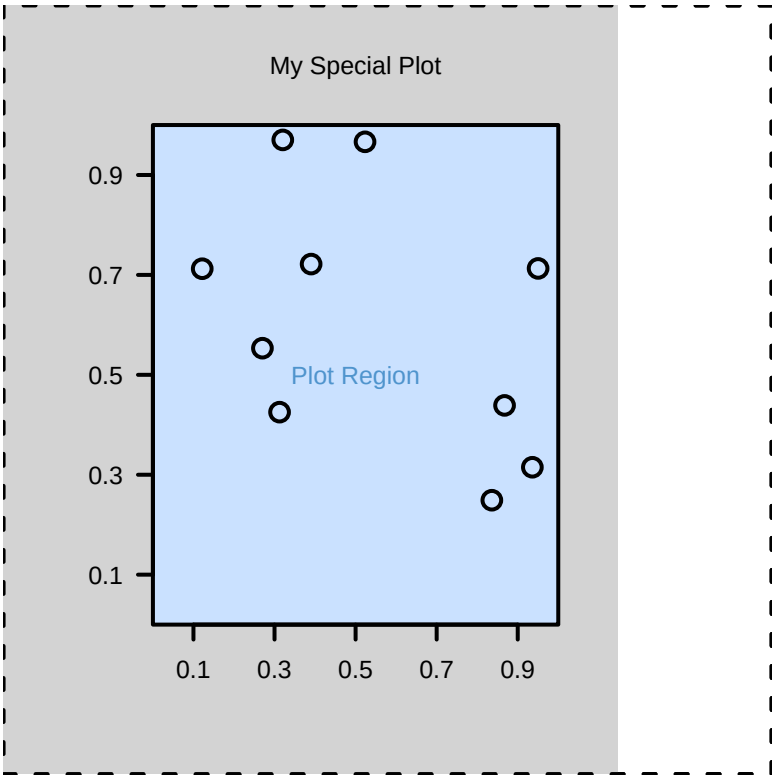


A Developer Constructs a Legend

- Drawing occurs within the current viewport.

```
R> grid.rect()
R> grid.xaxis(at=seq(.1, .9, length=5))
R> grid.yaxis(at=seq(.1, .9, length=5))
R> grid.points(x, y, pch=pch)
R> grid.text("My Special Plot",
             y=unit(1, "npc") + unit(2, "lines"))
```

A Developer Constructs a Legend

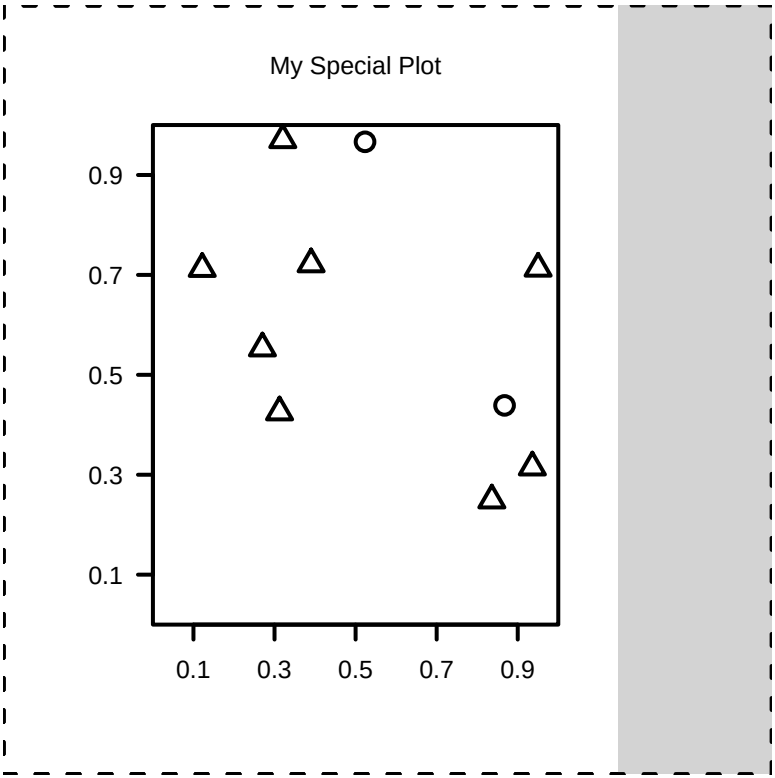


A Developer Constructs a Legend

- For the legend we just set up a different viewport ...

```
R> pop.viewport()
R> push.viewport(viewport(x=1, width=0.2,
                          just="right"))
```

A Developer Constructs a Legend

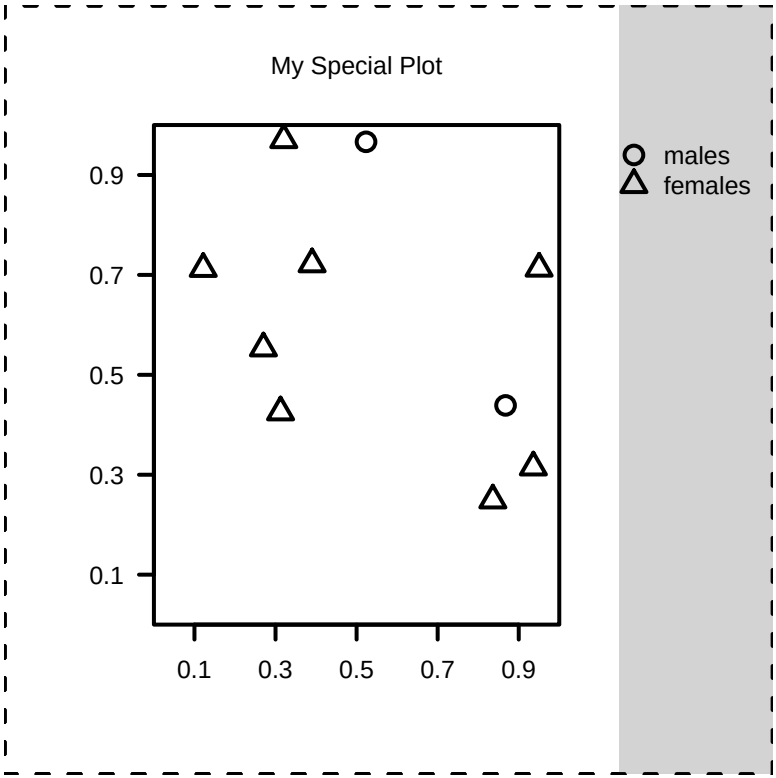


A Developer Constructs a Legend

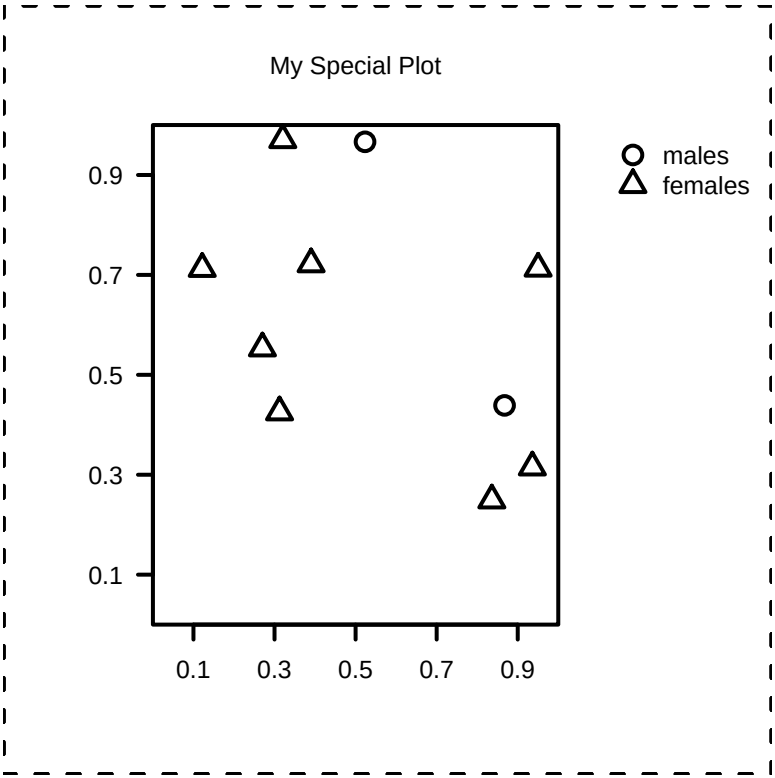
- ... and draw some data symbols and text.

```
R> grid.points(pch=1:2, x=unit(c(0.5, 0.5), "lines"),
                 y=unit(1, "npc") - unit(1:2, "lines"))
R> grid.text(c("males", "females"), just="left",
               x=unit(c(1.5, 1.5), "lines"),
               y=unit(1, "npc") - unit(1:2, "lines"))
```

A Developer Constructs a Legend



A Developer Constructs a Legend



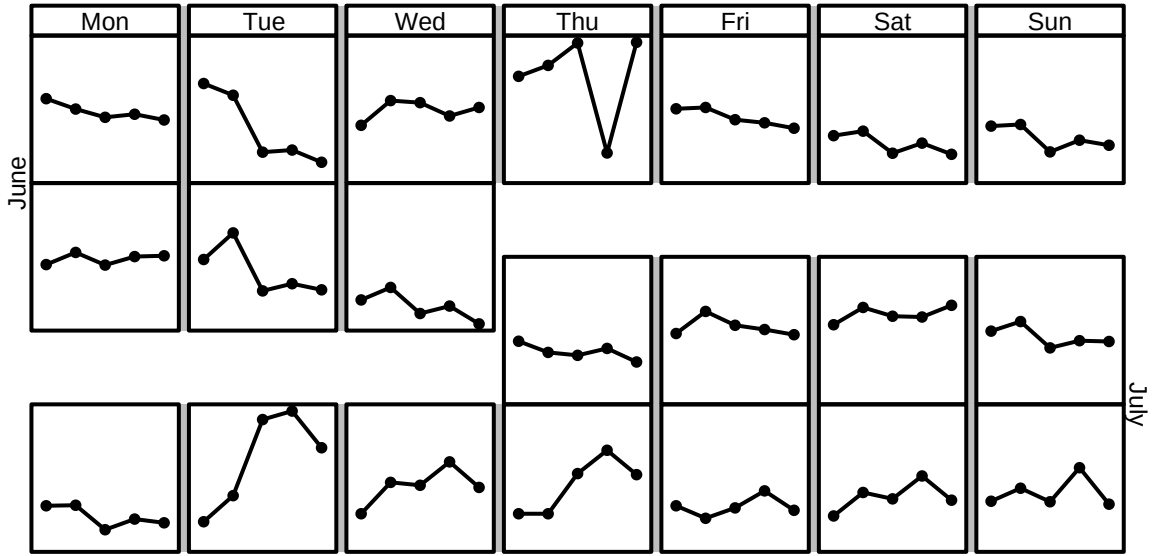
A Developer Gets Creative

The ozone2 Data Set

The response is 8-hour average (surface) ozone (from 9AM-4PM) measured in parts per billion (PPB) for 153 sites in the midwestern US over the period June 3, 1987 through August 31, 1987, 89 days.

Nychka, D., Cox, L., Piegorsch, W. (1998)
Case Studies in Environmental Statistics Lecture Notes in Statistics, Springer Verlag, New York

A Developer Gets Creative



A Developer Gets Creative

- First create a margin for labels

```
R> margin <- unit(1, "lines")
R> push.viewport(viewport(x=margin,
                          y=margin,
                          width=unit(1, "npc") - 2*margin,
                          height=unit(1, "npc") - 2*margin,
                          just=c("left", "bottom")))
```

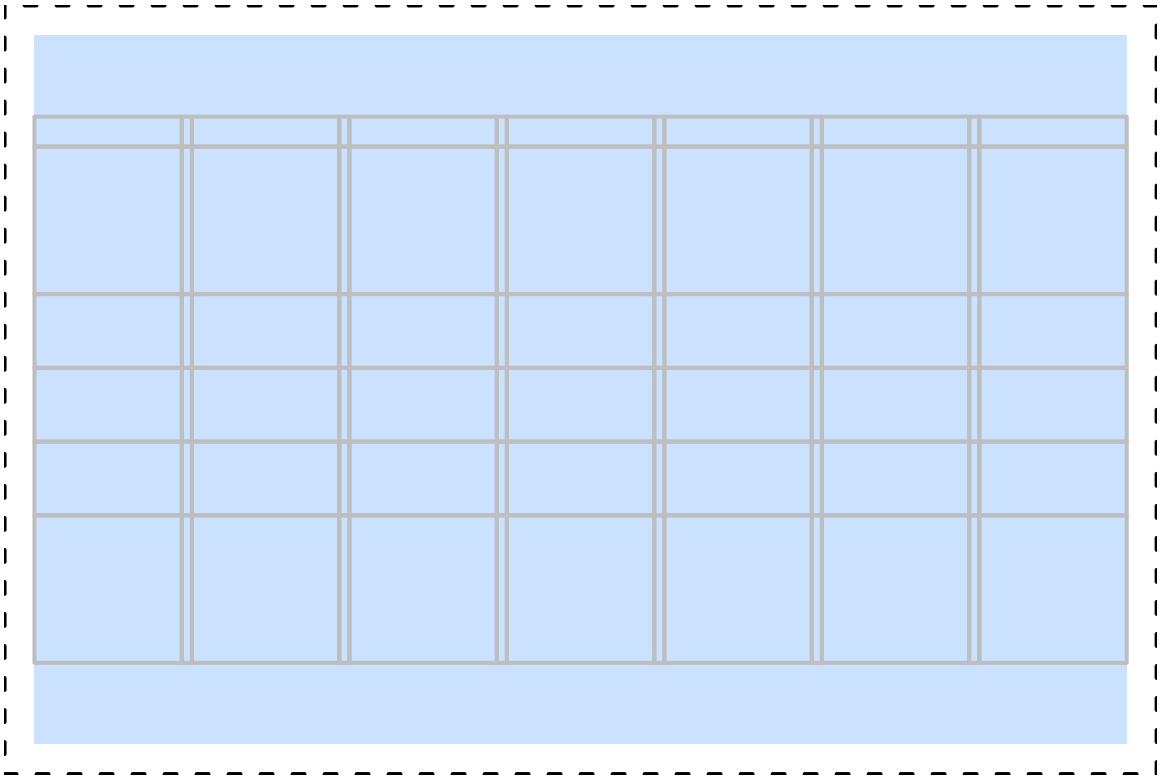
A Developer Gets Creative

A Developer Gets Creative

- Now carve up the viewport into many different regions. Grid layouts are very useful for this sort of thing.

```
R> widths <- unit(c(1,2,1,2,1,2,1,2,1,2,1,2,1),
                  rep(c("null", "mm"), len=7)
R> heights <- unit(c(1, 1, 0.5, 0.5, 0.5, 1),
                   c("lines", rep("null", 5)))
R> push.viewport(
     viewport(
       layout=grid.layout(13, 6,
                          widths=widths,
                          heights=heights,
                          respect=TRUE)))
```

A Developer Gets Creative

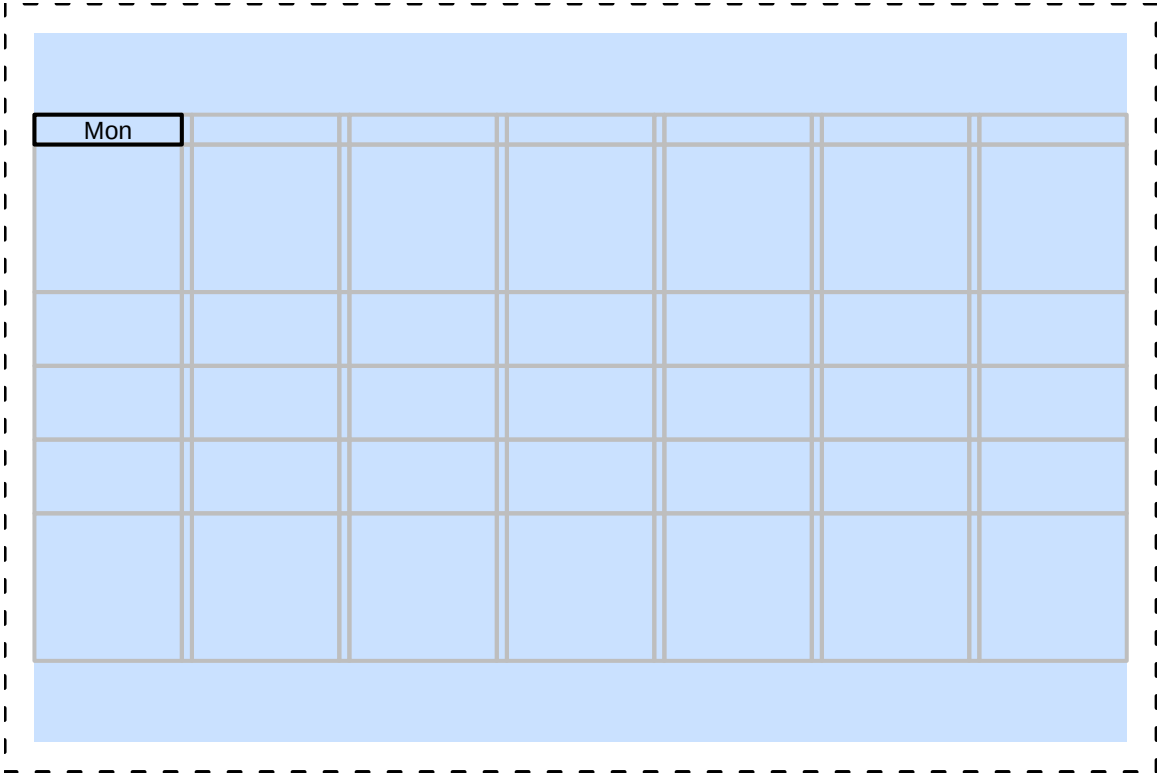


A Developer Gets Creative

- Draw in a particular region by pushing a viewport that occupies that region.

```
R> push.viewport(viewport(layout.pos.row=1,
                          layout.pos.col=1))
R> grid.rect()
R> grid.text("Mon")
R> pop.viewport()
```

A Developer Gets Creative

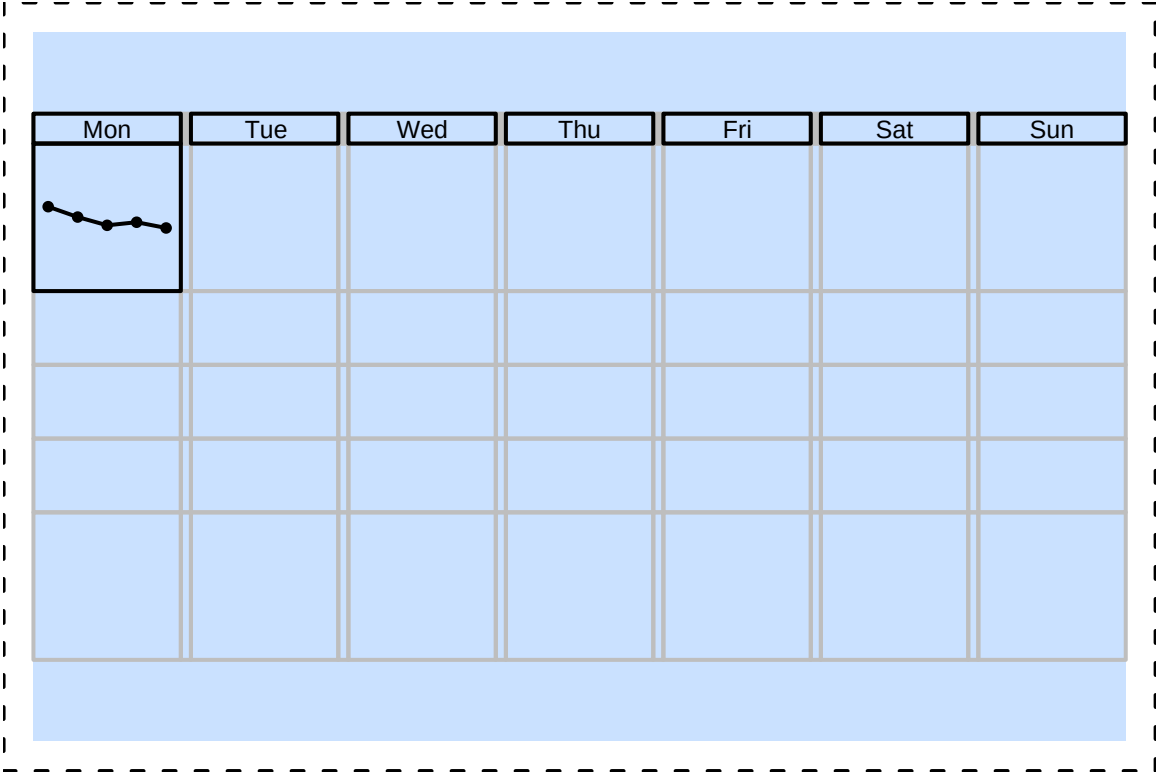


A Developer Gets Creative

- Draw in a particular region by pushing a viewport that occupies that region.

```
R> push.viewport(viewport(layout.pos.row=2,
                          layout.pos.col=1))
R> push.viewport(datavp)
R> grid.rect()
R> grid.lines(1:5, ozdata[1,], default="native")
R> grid.points(1:5, ozdata[1,],
               pch=16, size=unit(2, "mm"))
R> pop.viewport(2)
```

A Developer Gets Creative

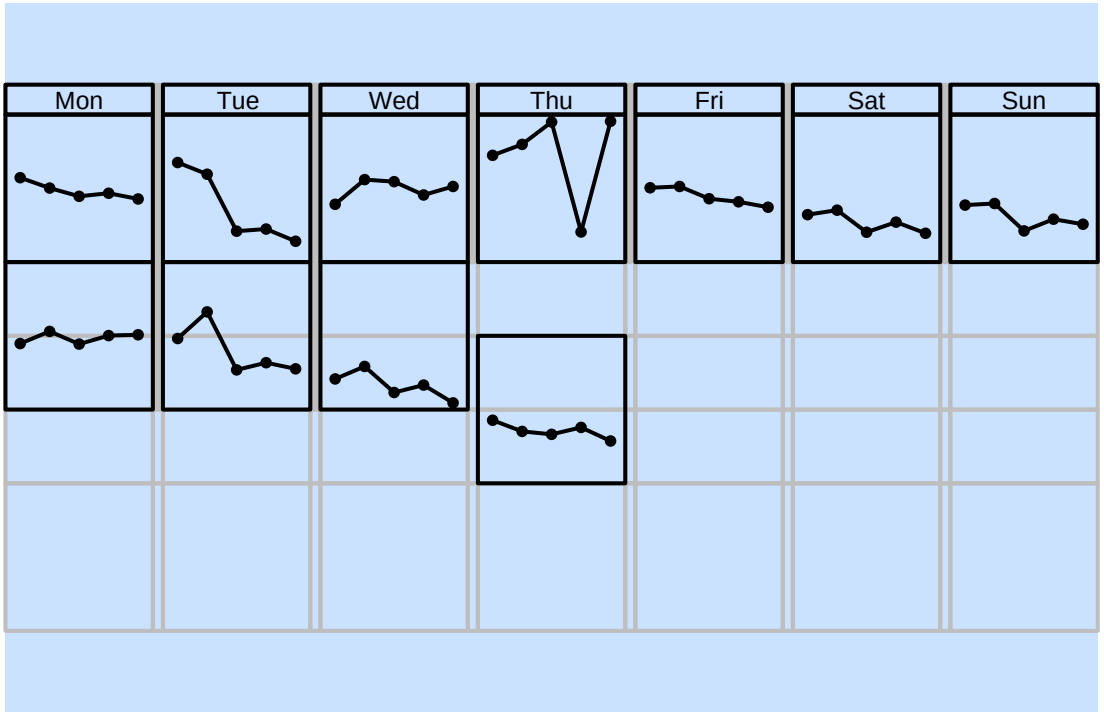


A Developer Gets Creative

- It is possible for a viewport to occupy a combination of several regions.

```
R> push.viewport(viewport(layout.pos.row=4:5,
                          layout.pos.col=7))
R> push.viewport(datavp)
R> grid.rect()
R> grid.lines(1:5, ozdata[11,], default="native")
R> grid.points(1:5, ozdata[11,],
               pch=16, size=unit(2, "mm"))
R> pop.viewport(2)
```

A Developer Gets Creative

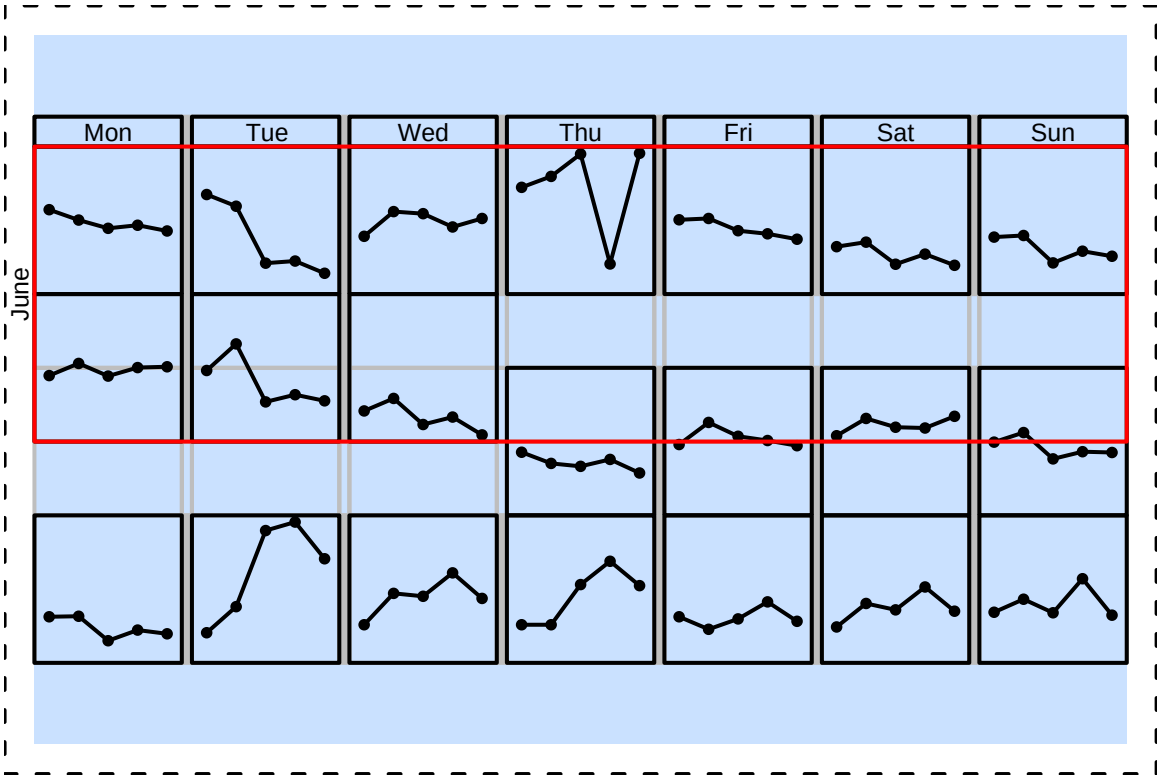


A Developer Gets Creative

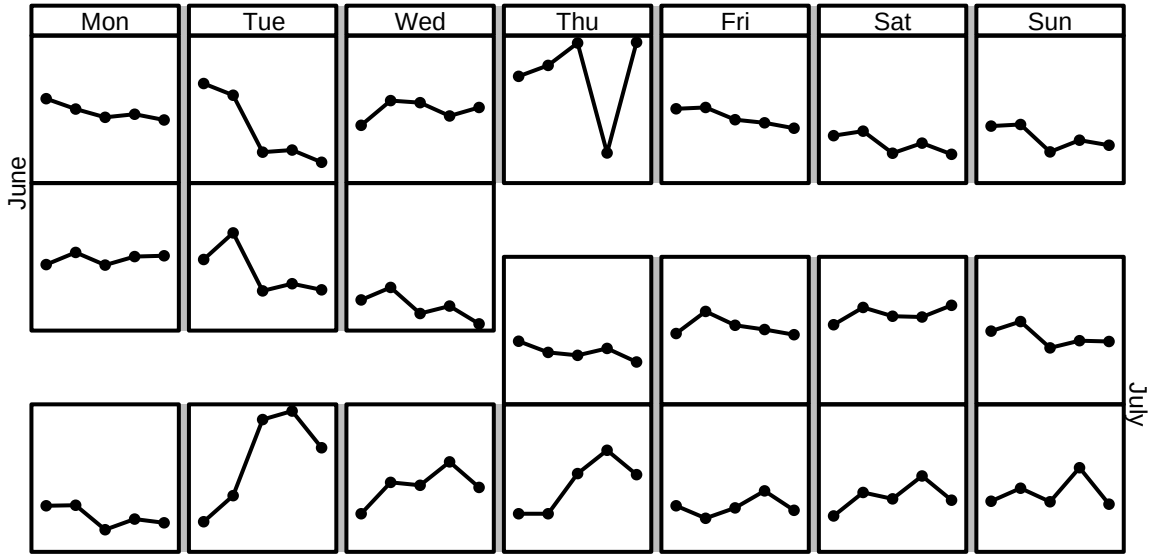
- It is possible for a viewport to occupy a combination of several regions.

```
R> push.viewport(viewport(layout.pos.row=2:4))
R> grid.text("June", x=unit(-0.5, "lines"),
             rot=90)
R> pop.viewport()
```

A Developer Gets Creative



A Developer Gets Creative



Other Stuff and Future Directions

- Rotated viewports
- Frames and packing
- Editing grid objects
- `grid` locator

- Integration of `grid` and standard ("base") graphics
- Constant improvements in convenience

Other Stuff and Future Directions

- Rotated viewports
- Frames and packing
- Editing grid objects
- `grid` locator

- Integration of `grid` and standard ("base") graphics
- Constant improvements in convenience

Other Stuff and Future Directions

- Rotated viewports
- Frames and packing
- Editing grid objects
- `grid` locator

- Integration of `grid` and standard ("base") graphics
- Constant improvements in convenience

Other Stuff and Future Directions

- Rotated viewports
- Frames and packing
- Editing grid objects
- `grid` locator

- Integration of `grid` and standard ("base") graphics
- Constant improvements in convenience

Other Stuff and Future Directions

- Rotated viewports
- Frames and packing
- Editing grid objects
- `grid` locator

- Integration of `grid` and standard ("base") graphics
- Constant improvements in convenience

Other Stuff and Future Directions

- Rotated viewports
- Frames and packing
- Editing grid objects
- `grid` locator

- Integration of `grid` and standard ("base") graphics
- Constant improvements in convenience

Other Stuff and Future Directions

- Rotated viewports
- Frames and packing
- Editing grid objects
- `grid` locator

- Integration of `grid` and standard ("base") graphics
- Constant improvements in convenience

Summary

- `grid` is an attempt at providing a framework in which graphical pieces are easy to access, combine, and manipulate.
- I do not want or expect all users to create all of their graphs from small pieces, but ...

  ... I do want all users to be able to see the pieces that their graphs were created from and be able to add or modify the pieces as easily and coherently as possible and ...

  ... it would be nice if some users created some of their graphs from small pieces some of the time; I think we would see more interesting and more illuminating graphs as a result.

Summary

- `grid` is an attempt at providing a framework in which graphical pieces are easy to access, combine, and manipulate.
- I do not want or expect all users to create all of their graphs from small pieces, but ...

  ... I do want all users to be able to see the pieces that their graphs were created from and be able to add or modify the pieces as easily and coherently as possible and ...

  ... it would be nice if some users created some of their graphs from small pieces some of the time; I think we would see more interesting and more illuminating graphs as a result.

Summary

- `grid` is an attempt at providing a framework in which graphical pieces are easy to access, combine, and manipulate.
- I do not want or expect all users to create all of their graphs from small pieces, but ...

  ... I do want all users to be able to see the pieces that their graphs were created from and be able to add or modify the pieces as easily and coherently as possible and ...

  ... it would be nice if some users created some of their graphs from small pieces some of the time; I think we would see more interesting and more illuminating graphs as a result.

Summary

- `grid` is an attempt at providing a framework in which graphical pieces are easy to access, combine, and manipulate.
- I do not want or expect all users to create all of their graphs from small pieces, but ...

  ... I do want all users to be able to see the pieces that their graphs were created from and be able to add or modify the pieces as easily and coherently as possible and ...

  ... it would be nice if some users created some of their graphs from small pieces some of the time; I think we would see more interesting and more illuminating graphs as a result.

Summary

- `grid` is an attempt at providing a framework in which graphical pieces are easy to access, combine, and manipulate.
- I do not want or expect all users to create all of their graphs from small pieces, but ...

  ... I do want all users to be able to see the pieces that their graphs were created from and be able to add or modify the pieces as easily and coherently as possible and ...

  ... it would be nice if some users created some of their graphs from small pieces some of the time; I think we would see more interesting and more illuminating graphs as a result.