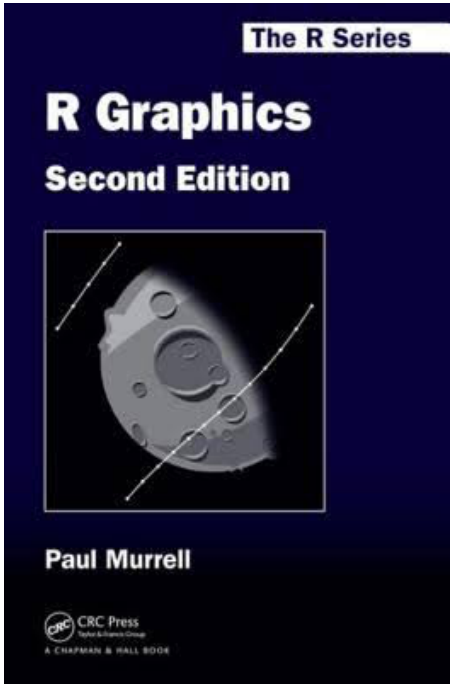
Base Graphics

Prof. Joyce Robbins

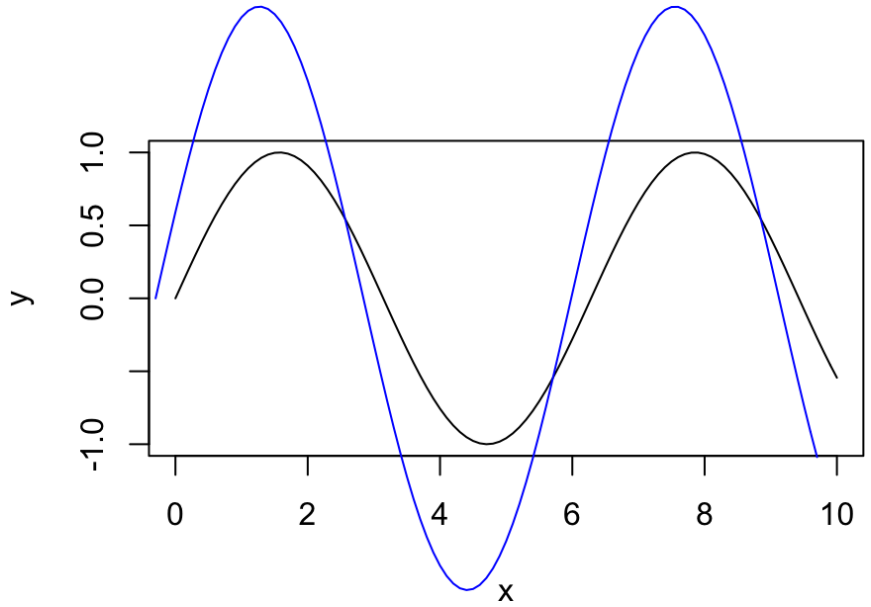
Resources

Paul Murrell, *R Graphics*



Why are we doing this?

```r
par(xpd = TRUE)
x <- seq(0, 10, .1)
y <- sin(x)
plot(x, y, type = "l")
lines(x - .3, y*2, col = "blue")
```



par()

w/o parameters, shows all the current settings

```
par()
```

```
## $xlog
## [1] FALSE
##
## $ylog
## [1] FALSE
##
## $adj
## [1] 0.5
##
## $ann
## [1] TRUE
##
## $ask
## [1] FALSE
##
## $bg
## [1] "white"
##
## $bty
## [1] "o"
##
## $cex
## [1] 1
##
## $cex.axis
## [1] 1
##
## $cex.lab
```

```
## [1] 1
##
## $cex.main
## [1] 1.2
##
## $cex.sub
## [1] 1
##
## $cin
## [1] 0.15 0.20
##
## $col
## [1] "black"
##
## $col.axis
## [1] "black"
##
## $col.lab
## [1] "black"
##
## $col.main
## [1] "black"
##
## $col.sub
## [1] "black"
##
## $cra
## [1] 10.8 14.4
##
## $crt
## [1] 0
##
## $csi
## [1] 0.2
##
## $cxy
## [1] 0.0399 0.1205
##
## $din
## [1] 5.0 3.5
```

```
##
## $err
## [1] 0
##
## $family
## [1] ""
##
## $fg
## [1] "black"
##
## $fig
## [1] 0 1 0 1
##
## $fin
## [1] 5.0 3.5
##
## $font
## [1] 1
##
## $font.axis
## [1] 1
##
## $font.lab
## [1] 1
##
## $font.main
## [1] 2
##
## $font.sub
## [1] 1
##
## $lab
## [1] 5 5 7
##
## $las
## [1] 0
##
## $lend
## [1] "round"
##
```

```
## $lheight
## [1] 1
##
## $ljoin
## [1] "round"
##
## $lmitre
## [1] 10
##
## $lty
## [1] "solid"
##
## $lwd
## [1] 1
##
## $mai
## [1] 1.02 0.82 0.82 0.42
##
## $mar
## [1] 5.1 4.1 4.1 2.1
##
## $mex
## [1] 1
##
## $mfcol
## [1] 1 1
##
## $mfg
## [1] 1 1 1 1
##
## $mfrow
## [1] 1 1
##
## $mgp
## [1] 3 1 0
##
## $mkh
## [1] 0.001
##
## $new
```

```
## [1] FALSE
##
## $oma
## [1] 0 0 0 0
##
## $omd
## [1] 0 1 0 1
##
## $omi
## [1] 0 0 0 0
##
## $page
## [1] TRUE
##
## $pch
## [1] 1
##
## $pin
## [1] 3.76 1.66
##
## $plt
## [1] 0.164 0.916 0.291 0.766
##
## $ps
## [1] 12
##
## $pty
## [1] "m"
##
## $smo
## [1] 1
##
## $srt
## [1] 0
##
## $tck
## [1] NA
##
## $tcl
## [1] -0.5
```

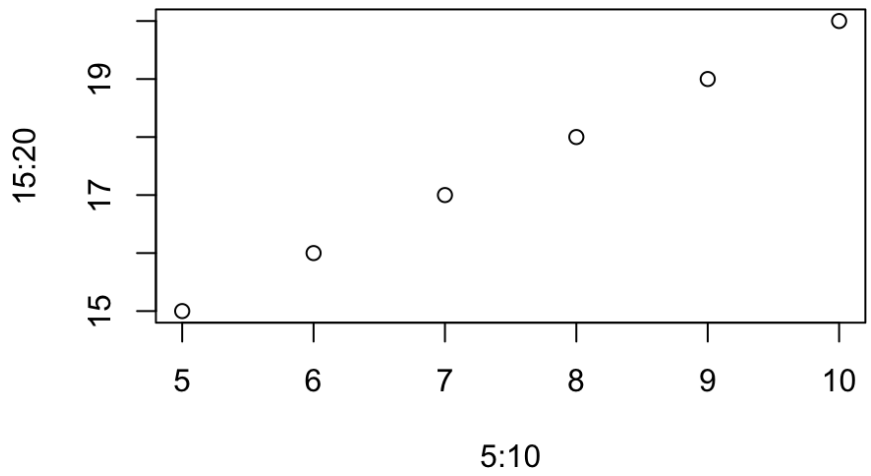
```
##
## $usr
## [1] 0 1 0 1
##
## $xaxp
## [1] 0 1 5
##
## $xaxs
## [1] "r"
##
## $xaxt
## [1] "s"
##
## $xpd
## [1] FALSE
##
## $yaxp
## [1] 0 1 5
##
## $yaxs
## [1] "r"
##
## $yaxt
## [1] "s"
##
## $ylbias
## [1] 0.2
```

`par()`

or just one

- `par("usr")` c(*x1*, *x2*, *y1*, *y2*)

```
plot(5:10, 15:20)
```

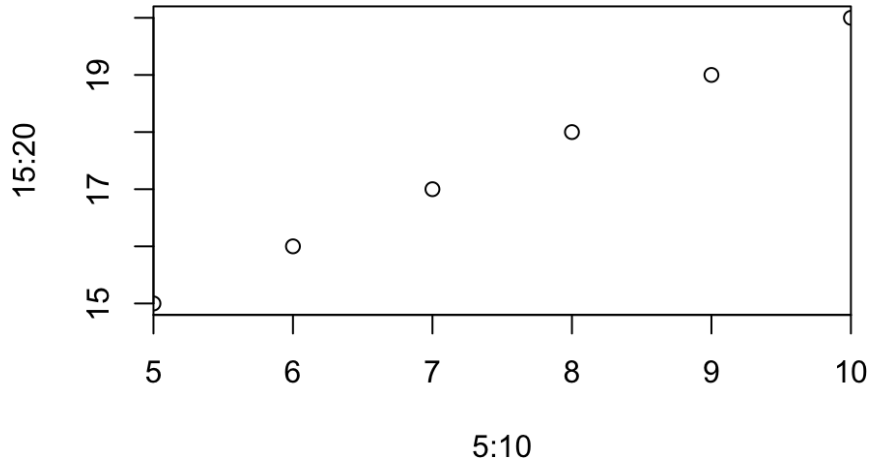



```
par()$usr
```

```
## [1]  4.8 10.2 14.8 20.2
```

par()

```
plot(5:10, 15:20, xaxs = "i")
```




```
par()$usr
```

```
## [1]  5.0 10.0 14.8 20.2
```

par()

- useful to keep old parameters
- `op <- par(mar = c(723, 1234, 123, 1))`
- `> op$mar`
  `[1] 5.1 4.1 4.1 2.1`
- Restore original paratmers with
  `> par(op)`
- `> par("mar")`
  `[1] 5.1 4.1 4.1 2.1`
- If you forget, there's always "Session" "Restart R" :)

Using `par()` with Rmarkdown

- changes to `par()` do not carry over from one chunk to the next
- unless you use
  `knitr::opts_knit$set(global.par = TRUE)`

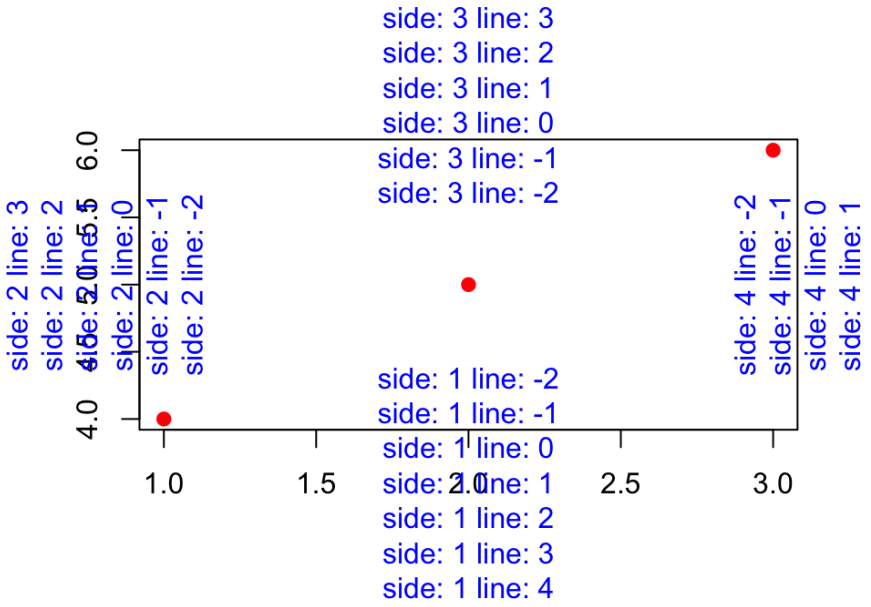
Figure margin `mar()`

- can only be set in **par()**
- mar = c(*bottom, left, top, right*)
- default: c(5.1, 4.1, 4.1, 2.1) **lines from plot region**

Figure margin `mar()` (and `mtext()`)

- can only be set in **par()**
- mar = c(*bottom, left, top, right*)
- default: c(5.1, 4.1, 4.1, 2.1) **lines from plot region**

```
plot(1:3, 4:6, ann = FALSE, pch = 19, col = "red")
for (i in 1:4) {
  for (j in -2:10) {
    mtext(paste("side:", i, "line:", j), side = i,
          line = j, col = "blue")
  }
}
```



Outer margin `oma()`

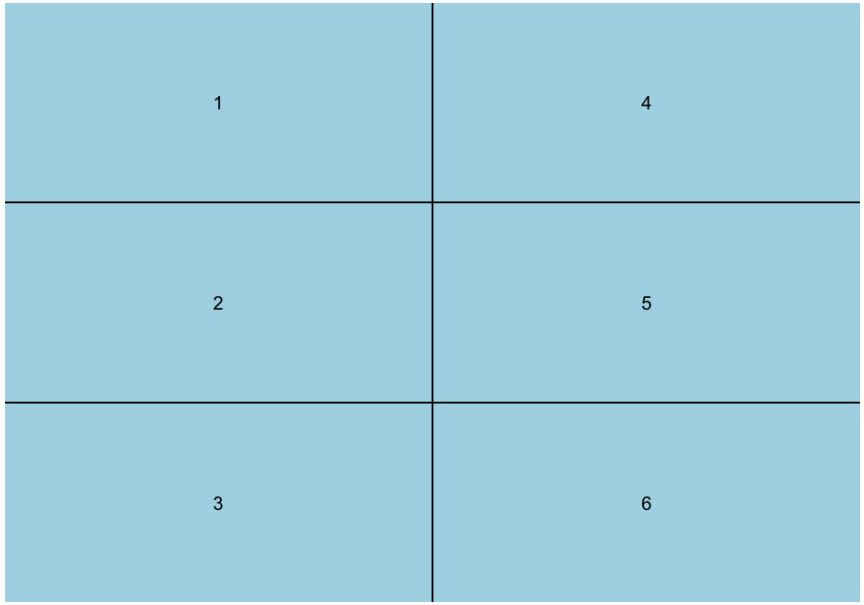
- can only be set in `par()`
- oma = c(*bottom, left, top, right*)
- default: c(0, 0, 0, 0) **lines from figure margin**
- useful with multiple plots

`mfrow()`, `mfcol()`, `layout()`

- `par(mfrow = c(3, 2))`
- `par(mfcol = c(3, 2))`
- `layout()` for plots of unequal sizes

layout.show()

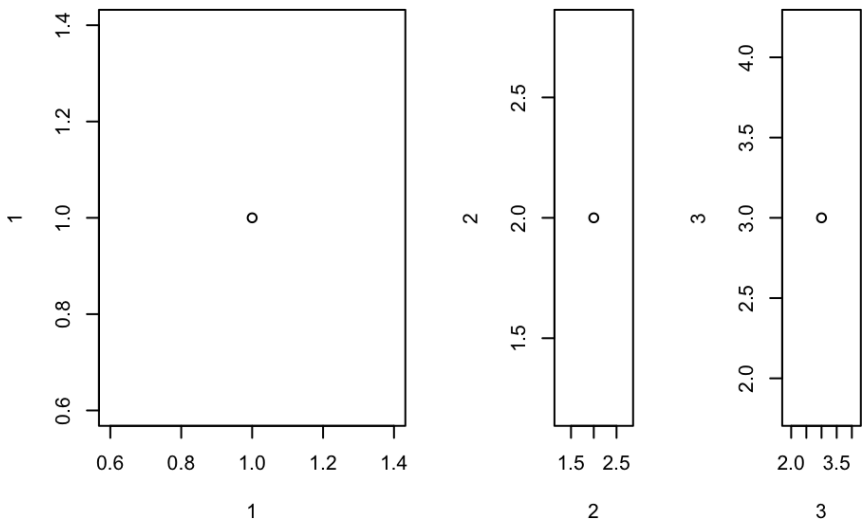
```
op <- par(mfcol = c(3, 2), bg = "lightblue")
layout.show(6)
```




```
par(op)
```

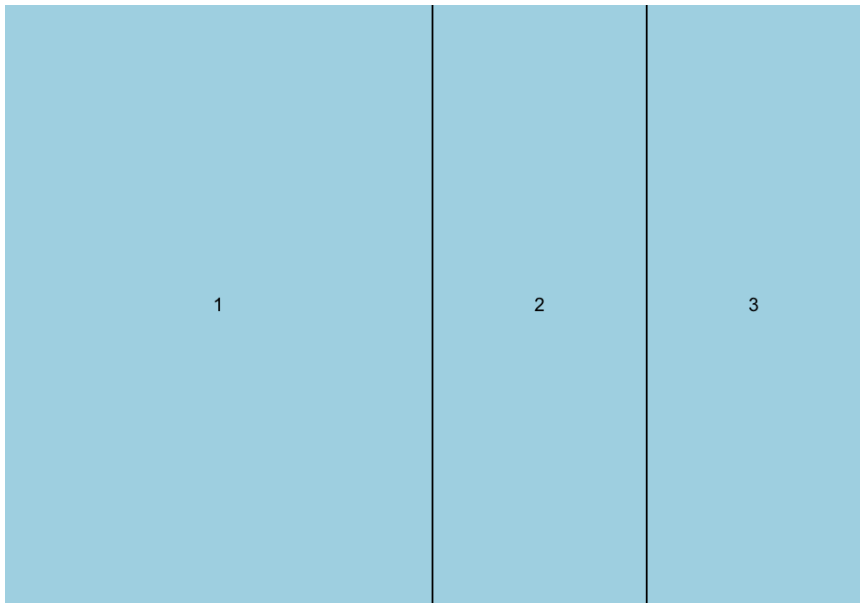
layout()

```
layout(rbind(c(1, 2, 3)), widths = c(2, 1, 1))
plot(1,1)
plot(2,2)
plot(3,3)
```



layout()

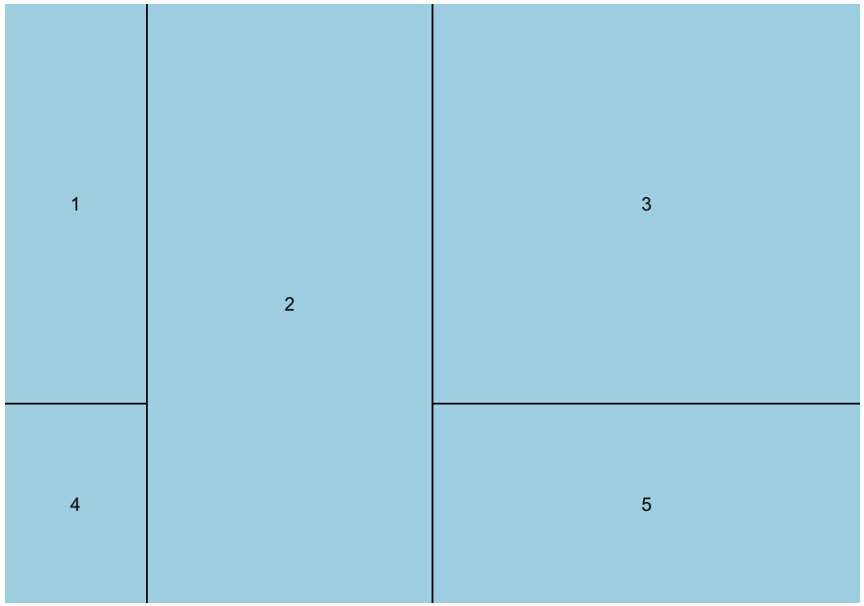
```
op <- par(bg = "lightblue")
layout(rbind(c(1, 2, 3)), widths = c(2, 1, 1))
layout.show(3)
```




```
par(op)
```

`layout()` – double row or column

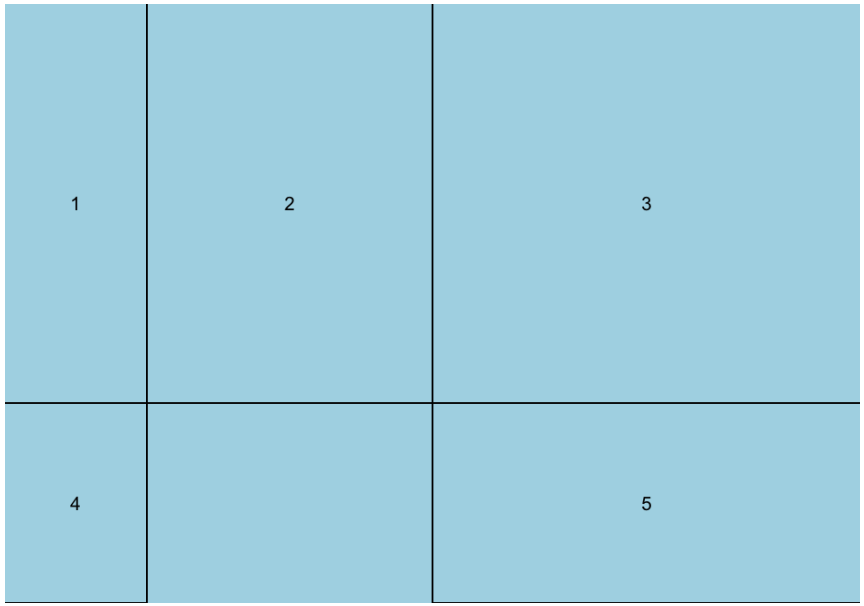
```
op <- par(bg = "lightblue")
layout(rbind(c(1, 2, 3),
             c(4, 2, 5)),
       widths = c(1, 2, 3),
       heights = c(2, 1))
layout.show(5)
```




```
par(op)
```

`layout()` – skip a plot

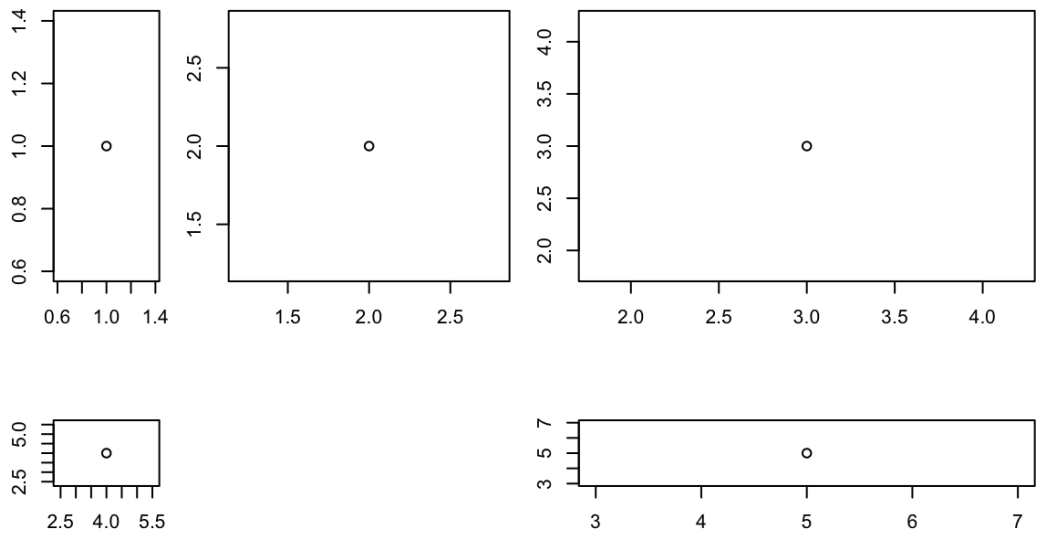
```
op <- par(bg = "lightblue")
layout(rbind(c(1, 2, 3),
             c(4, 0, 5)),
       widths = c(1, 2, 3),
       heights = c(2, 1))
layout.show(5)
```




```
par(op)
```

`layout()` – skip a plot

```
op <- par(mar = c(3, 2, 3, 1))
layout(rbind(c(1, 2, 3),
             c(4, 0, 5)),
       widths = c(1, 2, 3),
       heights = c(2, 1))
for (i in 1:5) plot(i,i)
```



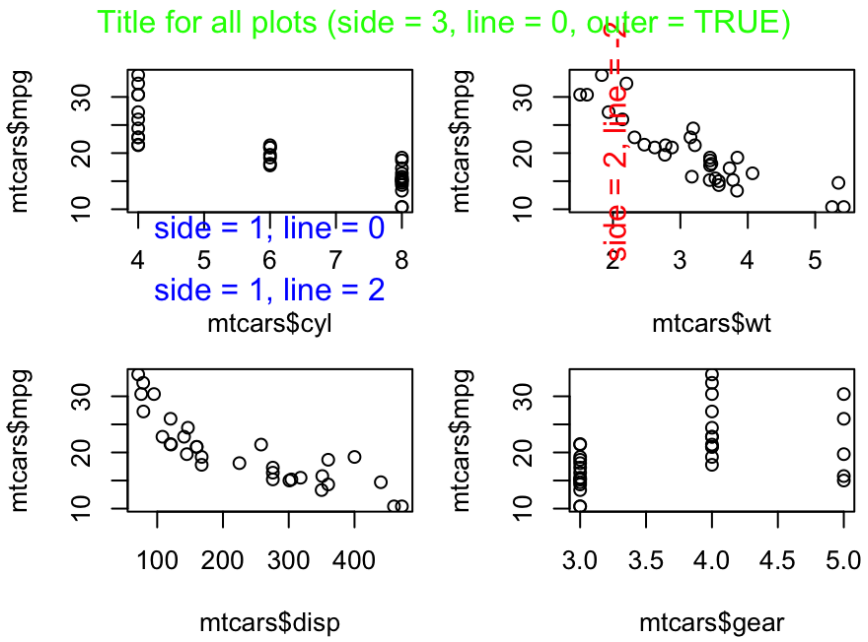

```
par(op)
```

Margin text `mtext()`

- first argument is *text* > - side is specified by 1, 2, 3, or 4 (**bottom, left, top, right**)
- use **outer = TRUE** for outer margin

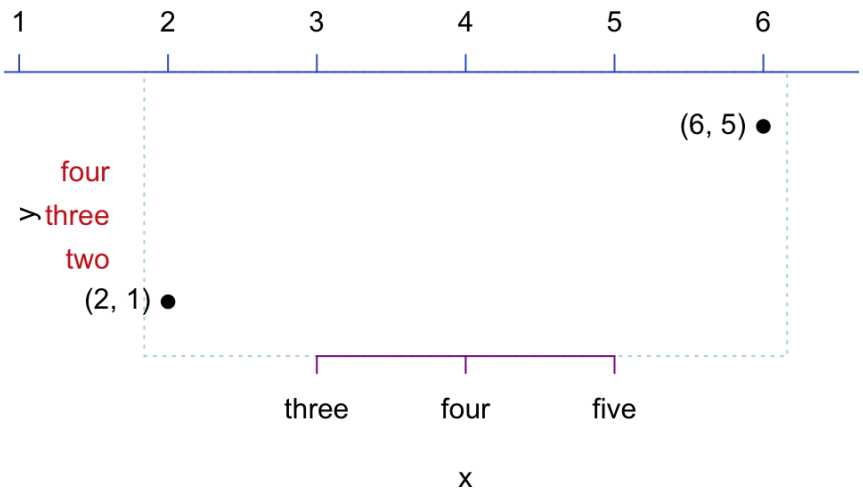
Example: par, mar, oma, mfrow, mtext

```
par(mar = c(4, 4, 1, 1),
    oma = c(1, 1, 1, 1),
    mfrow = c(2, 2))
plot(mtcars$cyl, mtcars$mpg)
mtext("side = 1, line = 0", side = 1, line = 0, col = "blue")
mtext("side = 1, line = 2", side = 1, line = 2, col = "blue")
plot(mtcars$wt, mtcars$mpg)
mtext("side = 2, line = -2", side = 2, line = -2, col = "red")
plot(mtcars$disp, mtcars$mpg)
plot(mtcars$gear, mtcars$mpg)
mtext("Title for all plots (side = 3, line = 0, outer = TRUE)",
      side = 3, line = 0, outer = TRUE, col = "green")
```



axis()

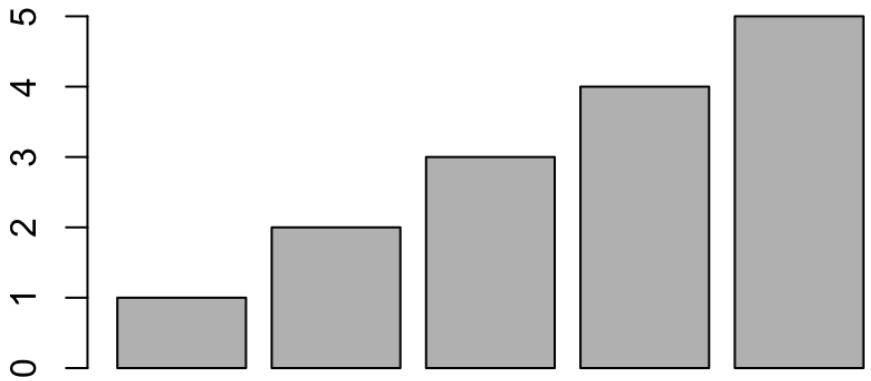
```
par(xpd = TRUE)
x <- c(2, 6)
y <- c(1, 5)
plot(x, y, pch = 19, axes = FALSE, ylim = c(0, 6))
box(lty = 'dotted', col = 'lightblue')
text(x, y, labels = c("(2, 1)", "(6, 5)"), pos = 2)
axis(side = 1, at = 3:5,
     labels = c("three", "four", "five"),
     col = "darkmagenta")
axis(side = 2, at = 2:4,
     labels = c("two", "three", "four"),
     tick = FALSE, col.axis = "firebrick3", las = 1)
axis(side = 3, at = 0:7, col = "royalblue3")
```



colors: http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf

Getting summary stats from plots

```
x <- 1:5
info <- barplot(x)
```

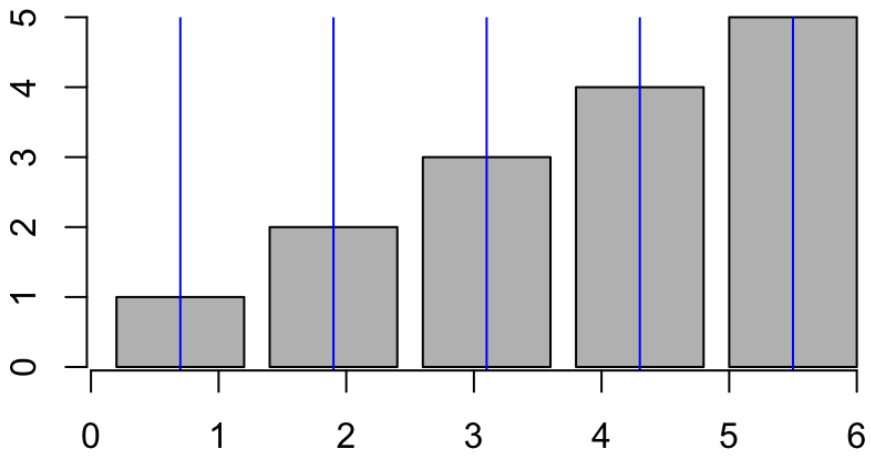



```
info
```

```
##      [,1]
## [1,]  0.7
## [2,]  1.9
## [3,]  3.1
## [4,]  4.3
## [5,]  5.5
```

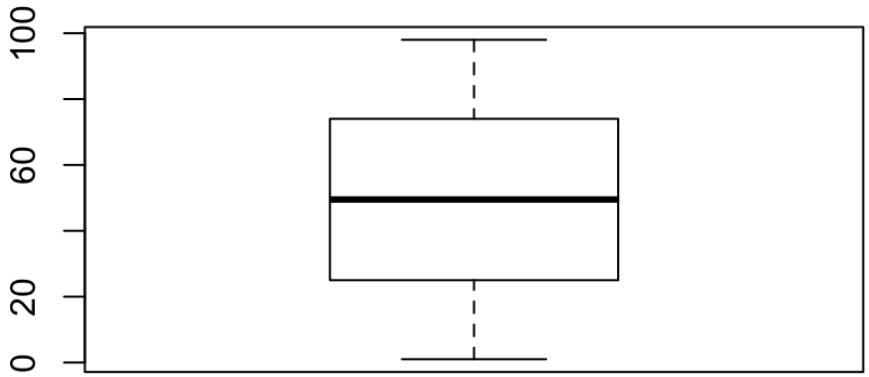
Getting summary stats from plots

```r
x <- 1:5
info <- barplot(x)
abline(v = info[,1], col = "blue")
axis(1)
```



Getting summary stats from plots

```r
df <- data.frame(mydata = 1:98)
info <- boxplot(df)
```




```r
info
```

```
## $stats
##      [,1]
## [1,]  1.0
## [2,] 25.0
## [3,] 49.5
## [4,] 74.0
## [5,] 98.0
## attr(,"class")
##    mydata
## "integer"
##
## $n
## [1] 98
```

```
##
## $conf
##      [,1]
## [1,] 41.7
## [2,] 57.3
##
## $out
## numeric(0)
##
## $group
## numeric(0)
##
## $names
## [1] "mydata"
```

Getting summary stats from plots

Value

List with the following components:

`stats` – a matrix, each column contains the extreme of the lower whisker, the lower hinge, the median, the upper hinge and the extreme of the upper whisker for one group/plot. If all the inputs have the same class attribute, so will this component.

`n` – a vector with the number of observations in each group.

`conf` – a matrix where each column contains the lower and upper extremes of the notch.

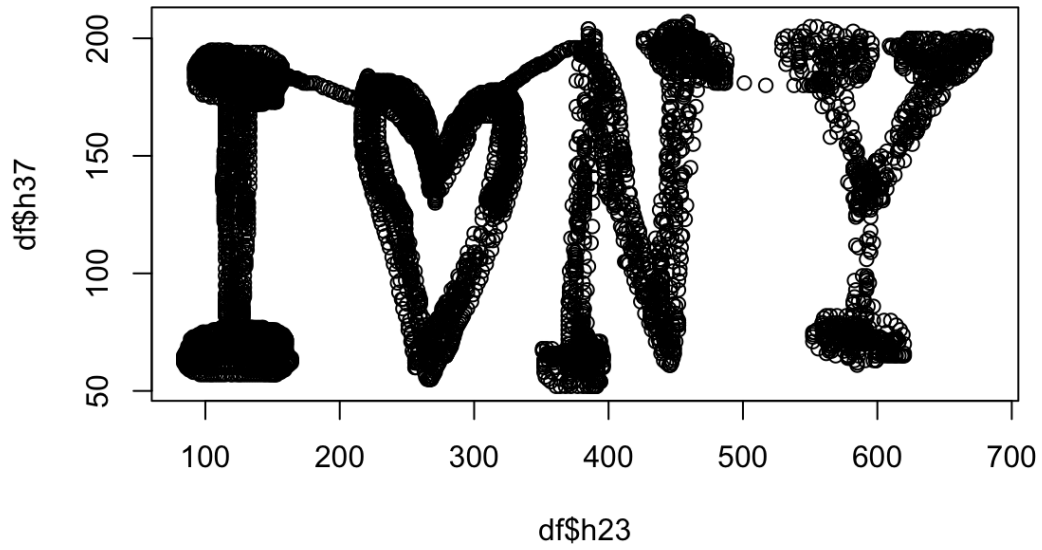
`out` – the values of any data points which lie beyond the extremes of the whiskers.

`group` – a vector of the same length as out whose elements indicate to which group the outlier belongs.

`names` – a vector of names for the groups.

locator()

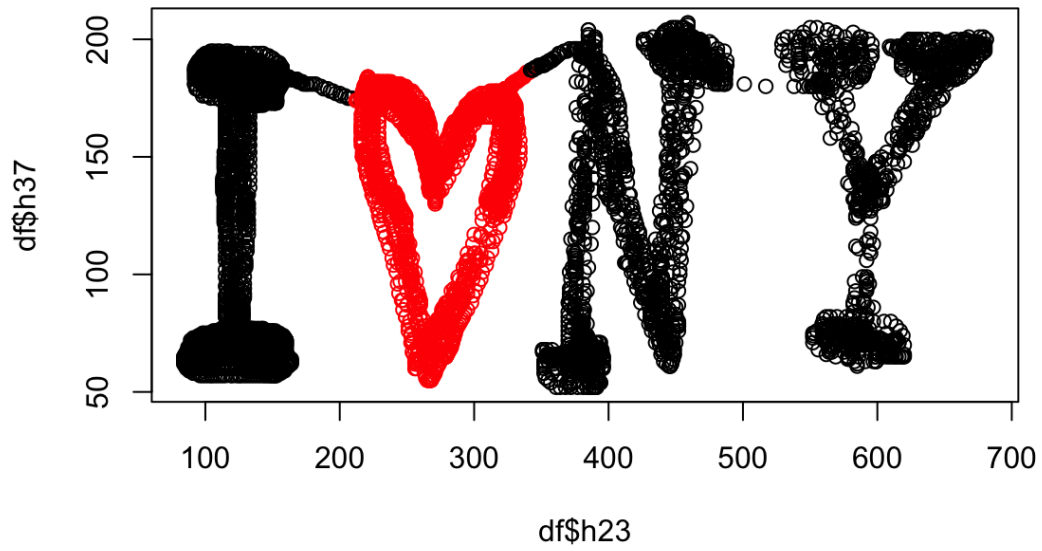
```
df <- read.csv("data1.csv")
plot(df$h23, df$h37)
```



Color

- create extra columns to indicate color

```
df$color <- ifelse(df$h23 > 209 & df$h23 < 342,
                   "red", "black")
plot(df$h23, df$h37, col = df$color)
```



Rmarkdown

- `.html` vs. `nb.html` files
- Chunk output inline
- `opts_chunk$set()`

Continuous Variables (Chapter 3)

Prof. Joyce Robbins

Continuous Variables

We’re looking for features such as:

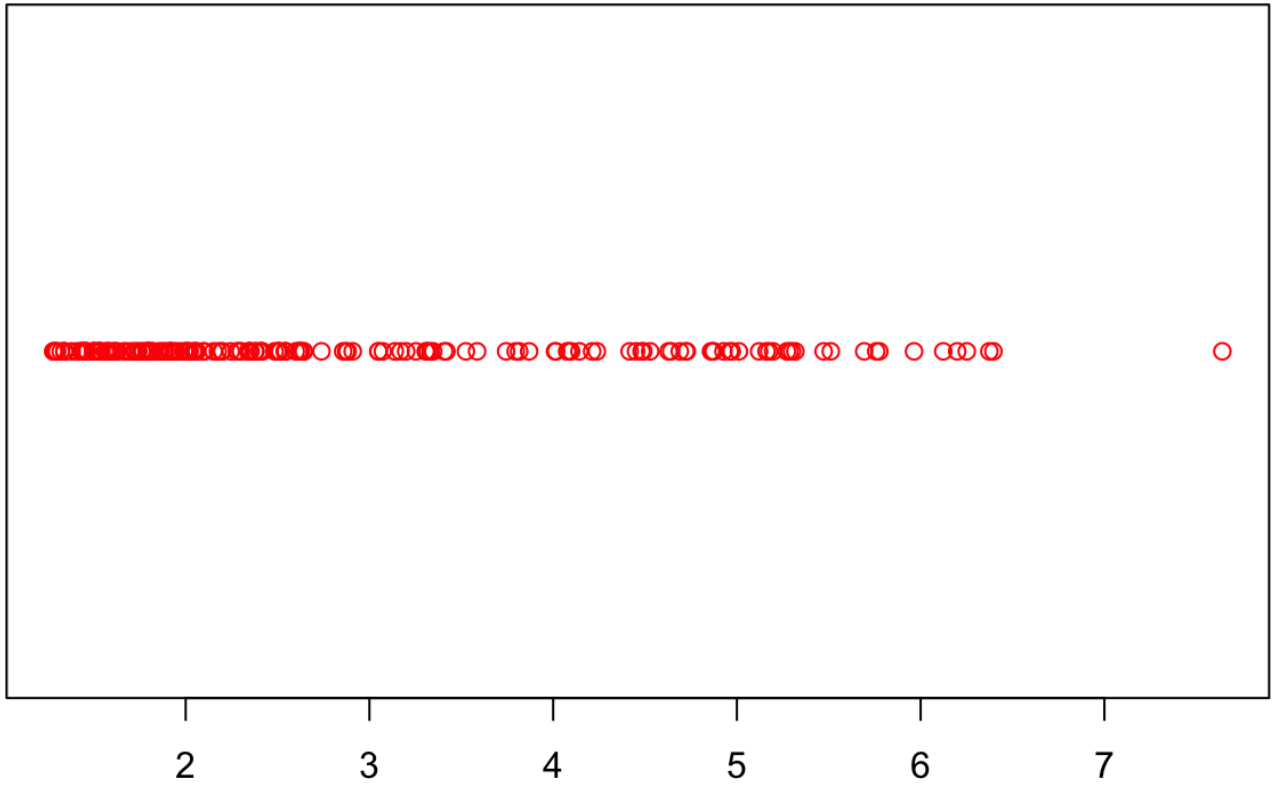
- Asymmetry
- Outliers
- Multimodality
- Gaps
- Heaping
- Rounding
- Impossibilities / Errors

Basic Options

- Stripcharts / rug plot
- Stem and leaf plot
- Dotplots
- Histogram
- Boxplot

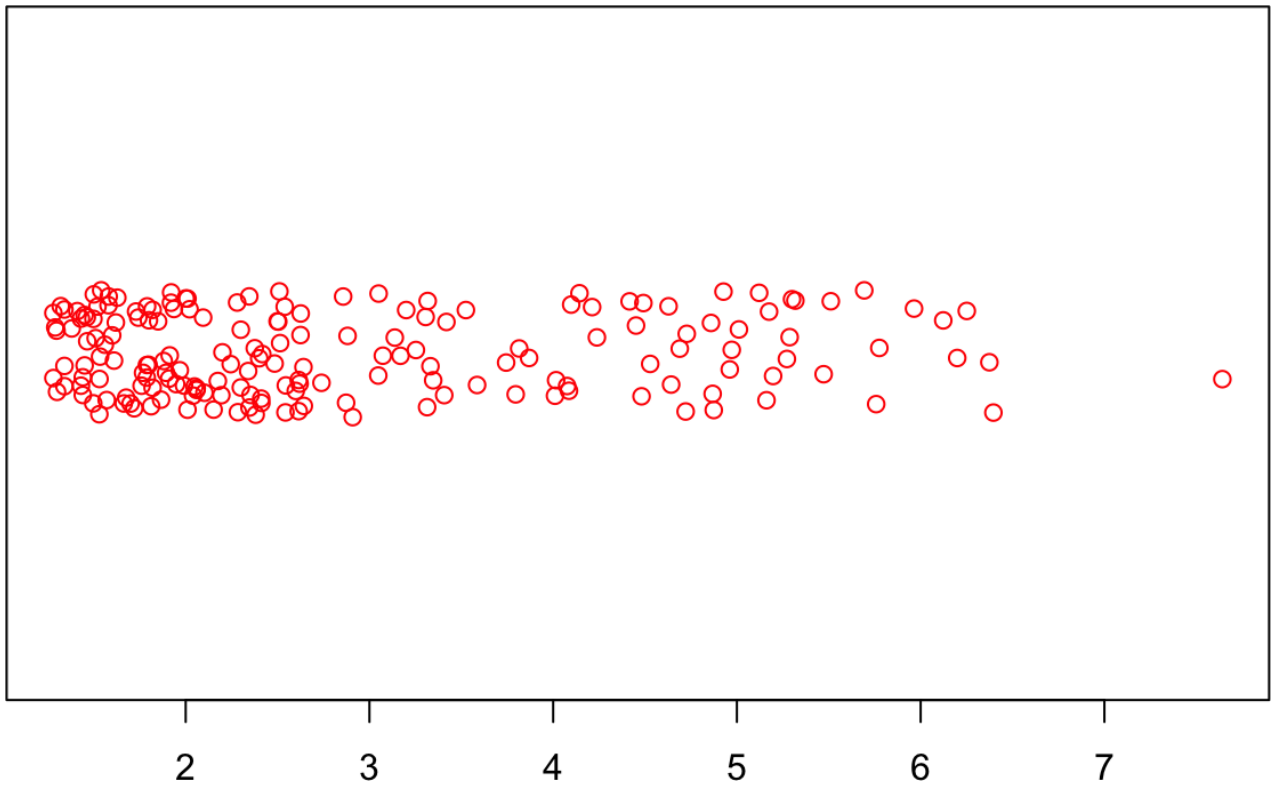
Strip charts

```
world <- read.csv("countries2012.csv")
stripchart(world$TFR, col = "red", pch = 21)
```



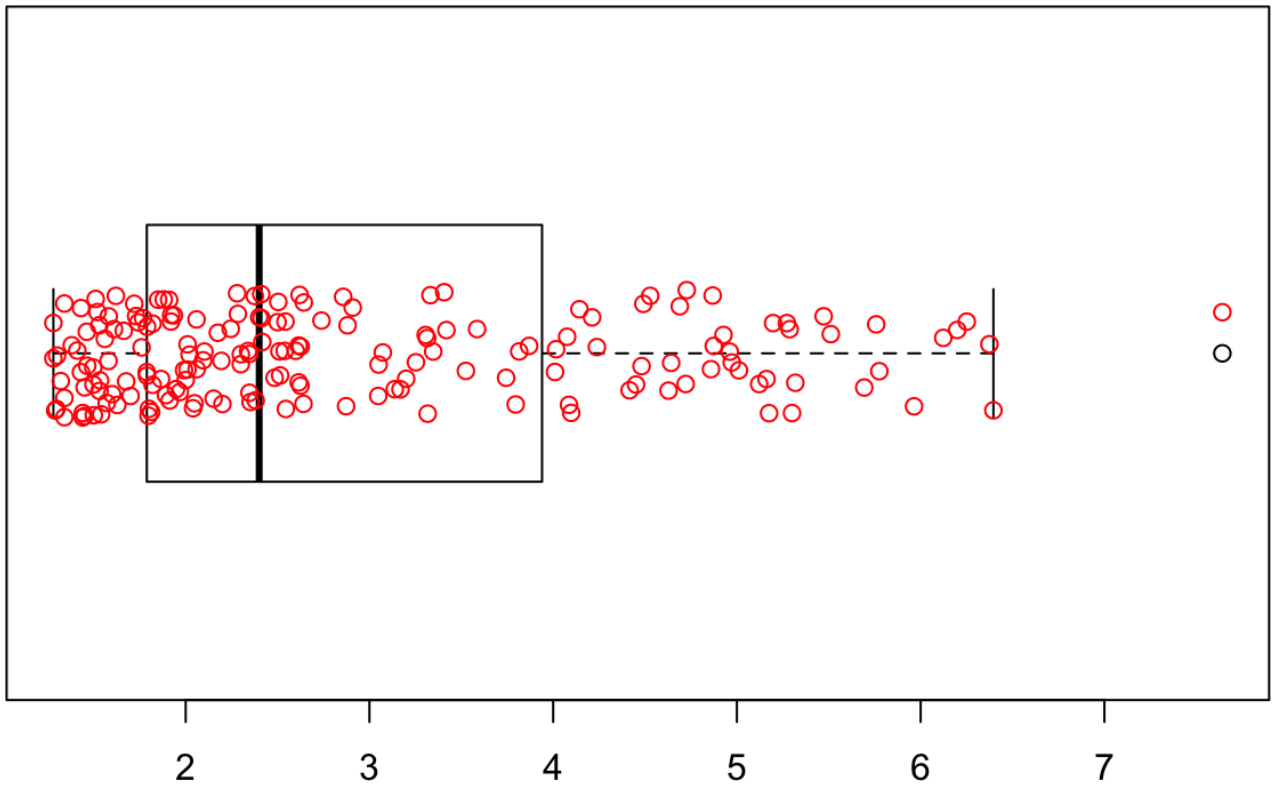
Strip charts

```
stripchart(world$TFR, col = "red", pch = 21,
           method = "jitter")
```



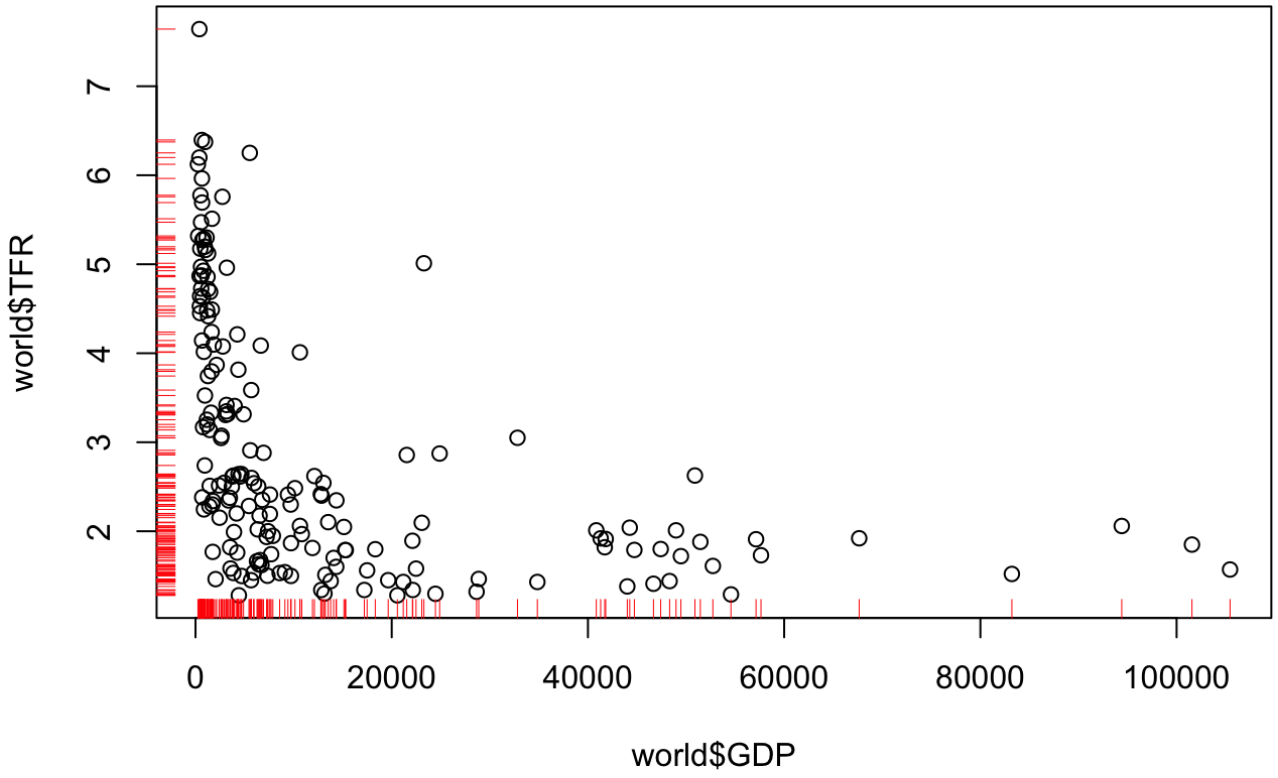
Strip charts w/ boxplot

```
boxplot(world$TFR, horizontal = TRUE)
stripchart(world$TFR, col = "red", pch = 21, add = TRUE, method = "jitter")
```



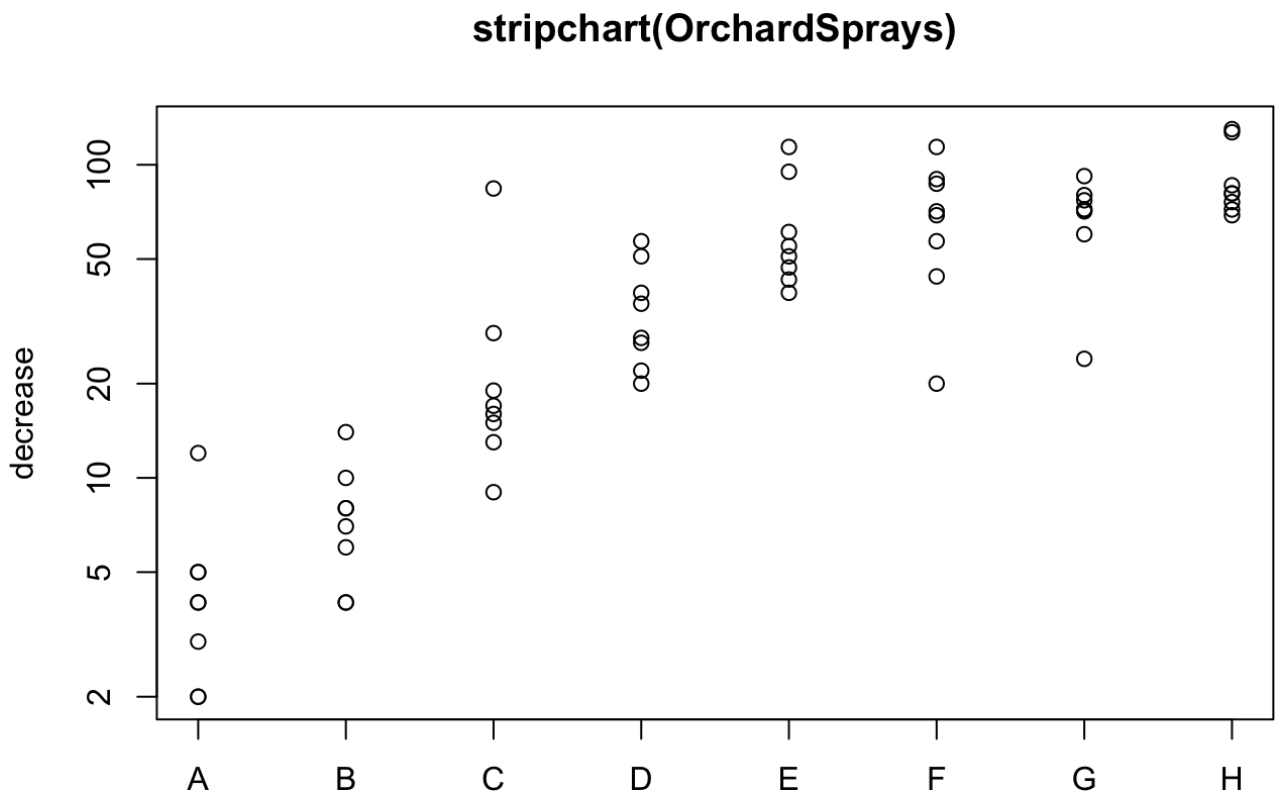
Rug plot

```
plot(world$GDP, world$TFR)
rug(world$GDP, col = "red")
rug(world$TFR, col = "red", side = 2)
```



Strip charts

```
stripchart(decrease ~ treatment,
    main = "stripchart(OrchardSprays)",
    vertical = TRUE, log = "y",
    data = OrchardSprays, pch = 21)
```



Stem and leaf plot

```r
prices <- c(379, 425, 450, 450, 499, 529, 535, 535, 545, 599, 665, 675, 699, 699, 725, 725, 745, 799)
stem(prices)
```

```
##
##   The decimal point is 2 digit(s) to the right of the |
##
##   3 | 8
##   4 | 355
##   5 | 03445
##   6 | 078
##   7 | 00335
##   8 | 0
```

Dot plot

```
library(ggplot2)
ggplot(world, aes(TFR, y = factor("TFR"))) +
    geom_dotplot()
```