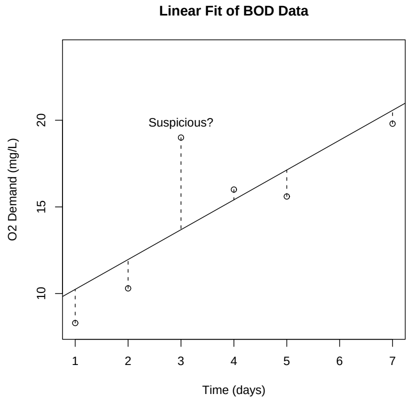
R Graphics

- Base Graphics
  - high-level, plot-generating functions (e.g., `plot`, `boxplot`)
  - lower-level, annotating functions (e.g., `lines`, `points`, `text`)

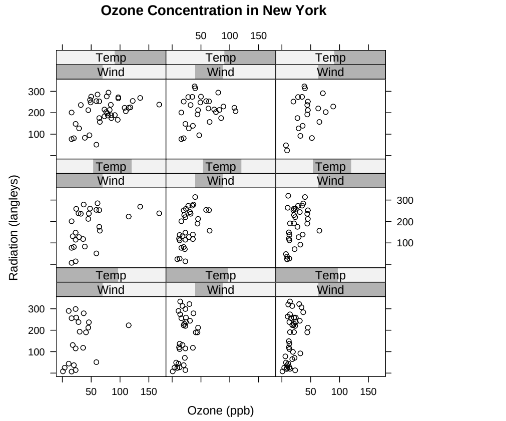
```
> plot(Time, demand,
       main="Linear Fit of BOD Data",  ylim=c(8,24),
       xlab="Time (days)", ylab="O2 Demand (mg/L)")
> abline(coef(l))
> segments(Time, demand, Time, fitted(l), lty="dashed")
> text(Time[3], demand[3]+par("cxy")[2], "Suspicious?")
NULL
>
```



R Graphics

- Lattice Graphics*
  - Created "all at once"
  - Well-suited for multivariate data

```
> library(lattice)
> data(environmental)
> environmental$Wind <- equal.count(environmental$wind, 3)
> environmental$Temp <- equal.count(environmental$temp, 3)
> xyplot(radiation ~ ozone | Wind * Temp, data=environmental,
         main="Ozone Concentration in New York",
         xlab="Ozone (ppb)", ylab="Radiation (langleys)")
>
```

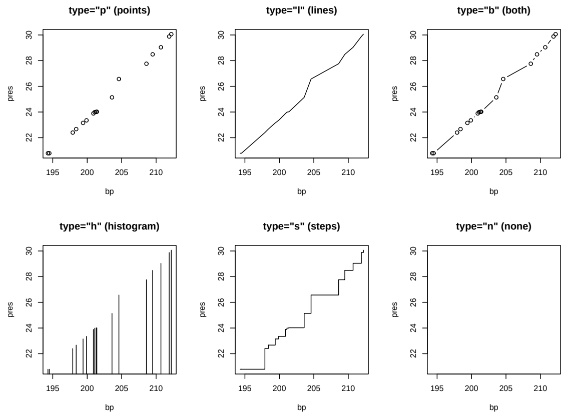


*Also called "Trellis Graphics" in the S-PLUS world.

`plot`: Bivariate Scatterplots

Form: `> plot(`*numeric*`, `*numeric*`, type="`*?*`")`

Various types:



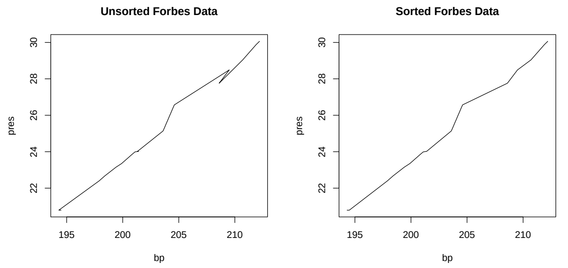
all generated with

```
> plot(bp, pres, type="x")  # bp and pres from the forbes data
```

Sorting Obs. for Line Graphs

Actually, the Forbes data looks different:

```
> library(MASS)
> attach(forbes)
> plot(bp, pres, type="l", main="Unsorted Forbes Data")
> detach()
> forbesS <- forbes[sort.list(forbes$bp),]
> attach(forbesS)
> plot(bp, pres, type="l", main="Sorted Forbes Data")
>
```

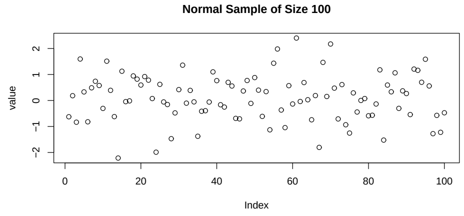


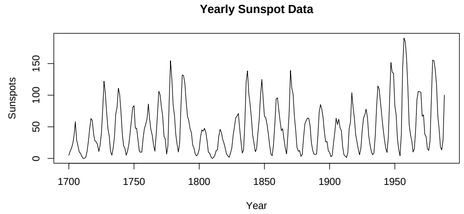
```
> forbes$bp
 [1] 194.5 194.3 197.9 198.4 199.4 199.9 200.9 201.1 201.4 [ . . . ]
> sort(forbes$bp)
 [1] 194.3 194.5 197.9 198.4 199.4 199.9 200.9 201.1 201.3 [ . . . ]
> sort.list(forbes$bp)
 [1]  2  1  3  4  5  6  7  8 10  9 11 12 14 13 15 16 17
>
```

`plot`: Univariate Scatterplots

Form: > `plot(`*numeric vector or time series*`)`

```
> plot(rnorm(100), main="Normal Sample of Size 100",
       ylab="value")
> sunspot.year
Time Series:
Start = 1700
End = 1988
Frequency = 1
  [1]   5.0  11.0  16.0  23.0  36.0  58.0  29.0  [ . . . ]
> plot(sunspot.year, main="Yearly Sunspot Data",
       ylab="Sunspots", xlab="Year")
>
```





`plot`: Barplots

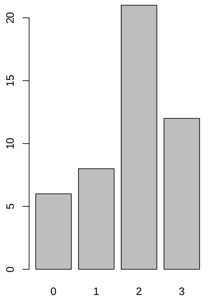
Form: > `plot(`*factor[*, *factor]*`)`

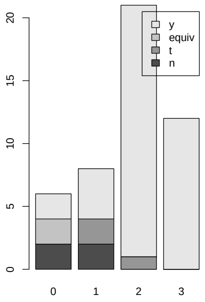
- barplot of a single factor

```
> attach(quest)
> table(r)
r
 0  1  2  3
 6  8 21 12
> plot(r)
>
```

- split barplot of two factors

```
> table(prereq.course)
prereq.course
    n     t equiv     y
    4     3     2    38
> plot(r, prereq.course)
>
```

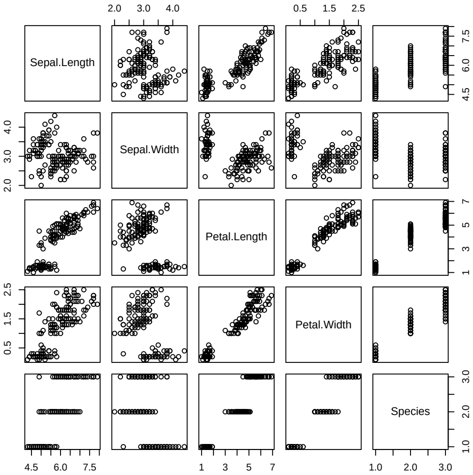




Other Forms of `plot`

- Boxplots by Group (see slide 32)
  Form: > `plot(`*factor*, *numeric*`)`
- Diagnostic Plots (see slides 31 and 35)
  Form: > `plot(`*fitted model*`)`
- Data Frames
  Form: > `plot(`*data frame*`)`

```
> plot(iris)
```

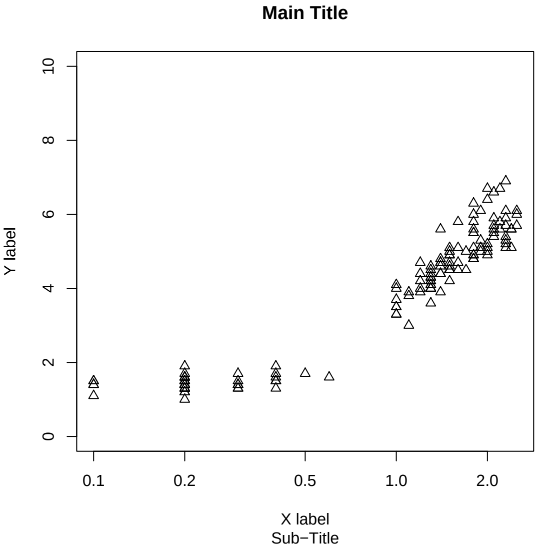


`plot` Parameters

- `type`: plot type
- `main`, `sub`: main and sub-titles
- `xlab`, `ylab`: axis labels
- `xlim=c(-5,5)`, `ylim=c(0,20)`: axis ranges
- `log="x"`, `="y"`, or `="xy"`: use logarithmic axes
- `col="red"`: colour
- `lty="dashed"`, `lwd=2`: line type and width for line plots
- `pch=1` or `pch="A"`: symbols or characters for point plots

Example of Some Parameters

```
> plot(Petal.Width, Petal.Length,
       main="Main Title", sub="Sub-Title",
       xlab="X label", ylab="Y label",
       ylim=c(0,100), log="x", pch=2)
>
```



Getting Help on `plot`

```
> help(plot)
plot              package:graphics              R Documentation

Generic X-Y Plotting

Description:

     Generic function for plotting of R objects.  For more
     details about the graphical parameter arguments, see 'par'.

[ . . . ]
Details:

     For simple scatter plots, 'plot.default' will be
     used. However, there are 'plot' methods for many R objects,
     including 'function's, 'data.frame's, 'density' objects,
     etc.  Use 'methods(plot)' and the documentation for these.

[ . . . ]
See Also:

     'plot.default', 'plot.formula' and other methods; 'points',
     'lines', 'par'.
>
```

Generic Functions and Methods

Some functions are *generic*: they use a different version when used on different objects:

```
> lm.object <- lm(y ~ x, data=mydata)
> aov.object <- aov(y ~ x, data=mydata)
> summary(lm.object)   # actually runs summary.lm(lm.object)
> summary(aov.object)  # actually runs summary.aov(aov.object)
> plot(lm.object)      # actually runs plot.lm(lm.object)
> plot(x, y)           # actually runs plot.default(x, y)
>
```

Generic functions will have one or more methods, a default method and object-specific methods:

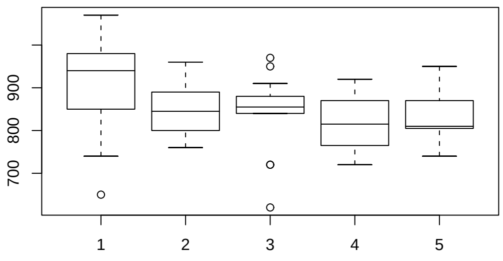
```
> methods(plot)
[ . . . ]
 [7] plot.data.frame*    plot.decomposed.ts* plot.default
[10] plot.dendrogram*    plot.density        plot.ecdf
[13] plot.factor*        plot.formula*       plot.hclust*
[16] plot.histogram*     plot.isoreg*        plot.lm
[ . . . ]
> methods(summary)
[ . . . ]
 [4] summary.aov         summary.aovlist     summary.connection
 [7] summary.data.frame  summary.default     summary.ecdf*
[10] summary.factor      summary.glm         summary.infl
[13] summary.lm          summary.loess*      summary.manova
[ . . . ]
> methods(search)
no methods were found
>
```

Getting help:

```
> help(plot)          # get generic, and useless, help
> help(plot.lm)       # get useful help
> help(plot.default)  # ditto
>
```

More High-Level Plots: `boxplot`

```
> morley
    Expt Run Speed
1      1   1   850
2      1   2   740
3      1   3   900
[ . . . ]
98     5  18   800
99     5  19   810
100    5  20   870
> split(morley$Speed, morley$Expt)
$"1"
 [1]  850  740  900 1070  930  850  950  980  980  880 1000  [ . . . ]
$"2"
 [1] 960 940 960 940 880 800 850 880 900 840 830 790 810 880 [ . . . ]
[ . . . ]
> boxplot(split(morley$Speed, morley$Expt))
>
```



`boxplot` Parameters

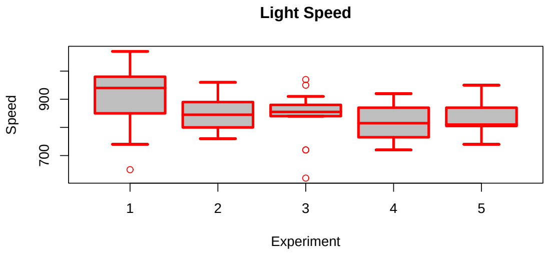
- `plot` parameters: `main`, `sub`, `xlab`, `ylab`, `ylim`, `log`, `lty`, `lwd`, `pch`
- `boxplot`-specific parameters
  - `varwidth=T`: box widths prop. to $\sqrt{n}$
  - `notch=T`: add "confidence notches"
  - `outline=F`: don't draw outliers
  - `names`: box labels
  - `boxwex=0.8`: scale factor for box widths
  - `border="pink"`: color(s) for box outlines and outlier symbols
  - `col="purple"`: color(s) for box bodies
  - `horizontal=T`: draw horizontal boxes
- `add=T`: add this plot to current plot

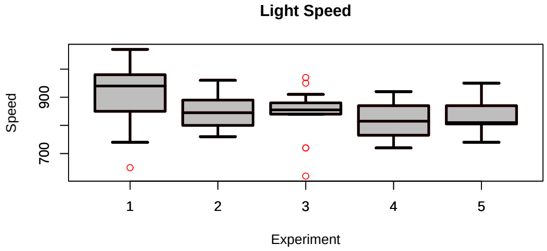
The following looks "best" on a color display:

```
> boxplot(split(morley$Speed, morley$Expt),
          main="Ugliest Boxplot on Earth",
          xlab="Experiment", ylab="Speed",
          col=1:5, border=6:2, lty=1:5, lwd=1:5,
          pch=c("A","B","C","D","E"), ylim=c(600,1100))
>
```

A Cool Trick: Red Outliers

```
> boxplot(split(morley$Speed, morley$Expt),
          main="Light Speed", xlab="Experiment", ylab="Speed",
          border="red", lwd=3, lty=1, col="grey")
> boxplot(split(morley$Speed, morley$Expt),
          border="black", lwd=3, lty=1, outline=F, add=T)
>
```



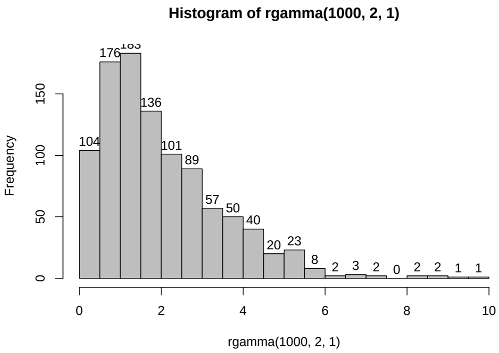


More High-Level Plots: `hist`

Parameters:

- `plot` parameters and `add=T`
- `breaks`: method to select breakpoints (e.g., `"Sturges"`), number of cells, or vector of breakpoints
- `probability=T`: display relative frequencies (probabilities) instead of raw frequencies (counts)
- `labels`: show freqs on top of bars

```
> hist(rgamma(1000,2,1), labels=T, col="grey", breaks=15)
>
```


Histogram of rgamma(1000, 2, 1)
Frequency
rgamma(1000, 2, 1)