Lecture 7

An introduction to R Graphics

MCB 416A/516A
Statistical Bioinformatics and Genomic Analysis

Prof. Lingling An
Univ of Arizona

Outline

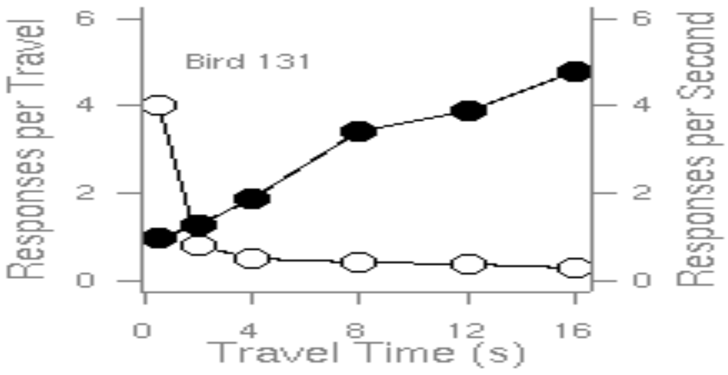
- Overview - gallery of R Graph examples
- Basic R graphics functions
- Practice

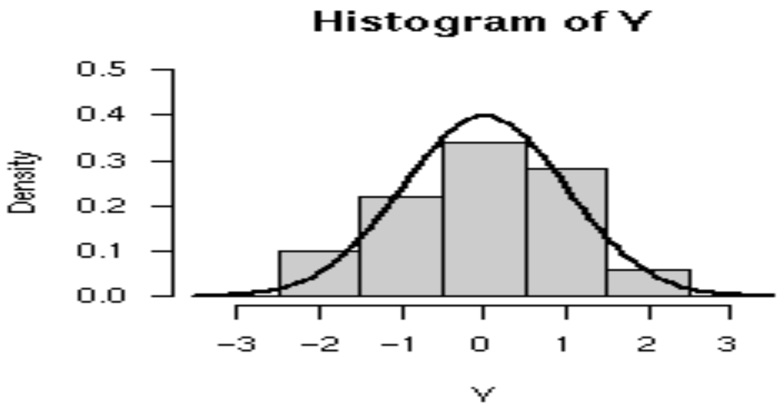
Overview

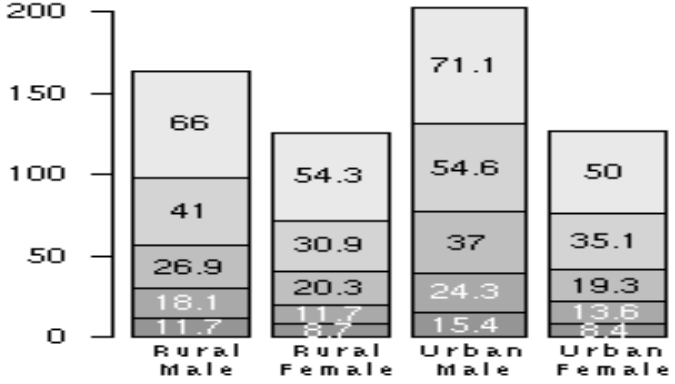
- One of the best things about R is the ability to create publication-quality graphs.
- Graph function syntax follows general functional rules of the R language: the fuzzy boundaries between dataset management, statistical analysis, and graphics ultimately is an advantage to the language, but can be a little overwhelming at first.
- Corollary: Understanding graphics will facilitate understanding R programming in general---and it gives pretty instant feedback.

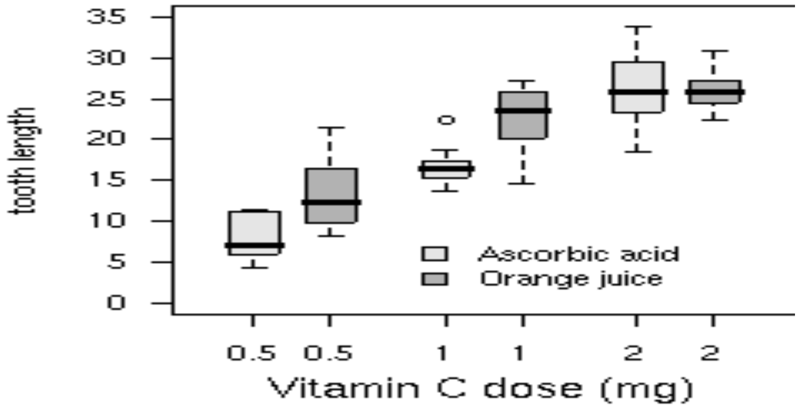
Overview-2

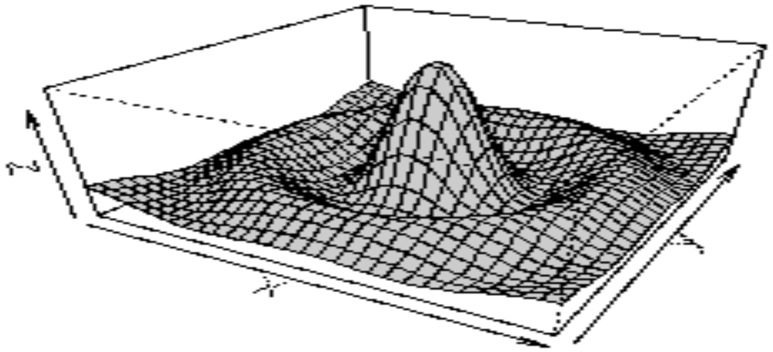
- There are numerous functions available for producing diverse output, with many commonalities in syntax between them.
- There also are many ways to accomplish a given task.
- How a plotting function deals with data often depends on the data type (e.g. matrix, factor, vector) given to it.

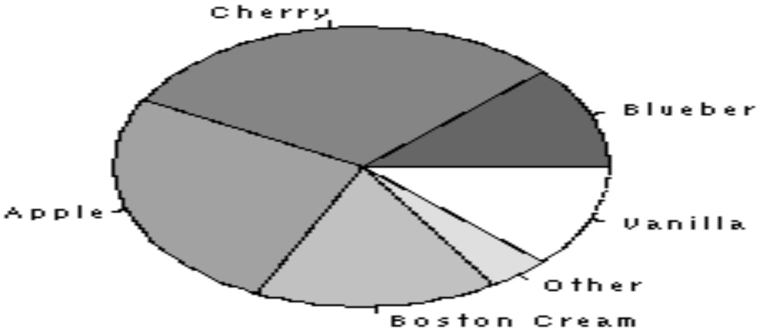

Bird 131
Responses per Travel
Responses per Second
Travel Time (s)
0
4
8
12
16
0
2
4
6

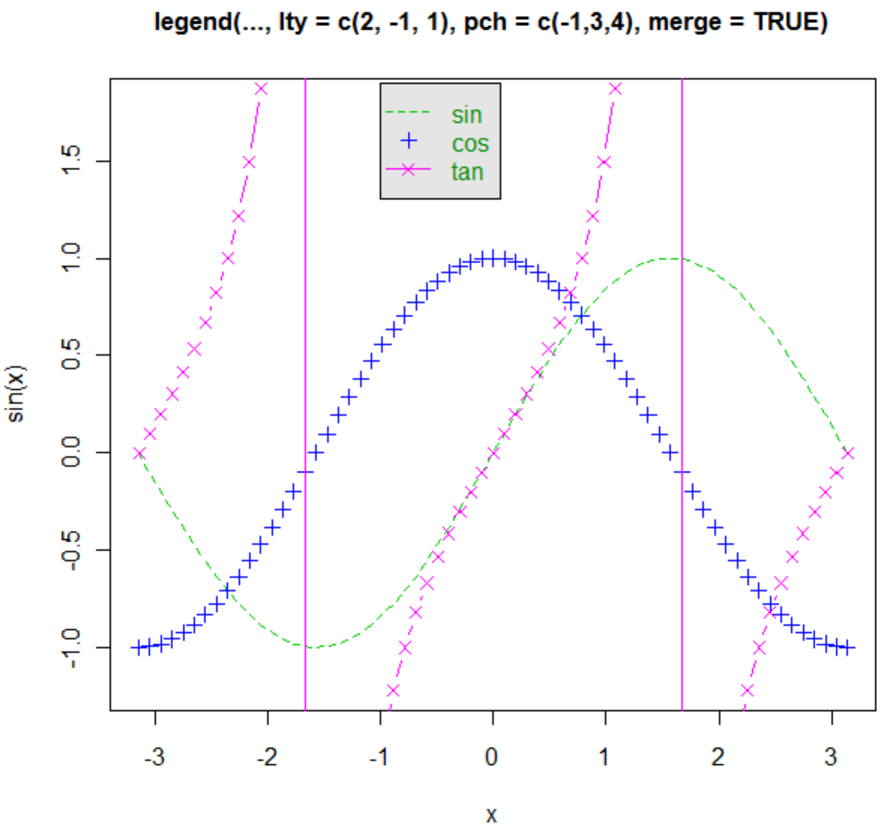

Histogram of Y
Density
Y
0.0
0.1
0.2
0.3
0.4
0.5
−3
−2
−1
0
1
2
3


200
150
100
50
0
66
41
26.9
18.1
11.7
54.3
30.9
20.3
11.7
8.7
71.1
54.6
37
24.3
15.4
50
35.1
19.3
13.6
8.4
Rural Male
Rural Female
Urban Male
Urban Female

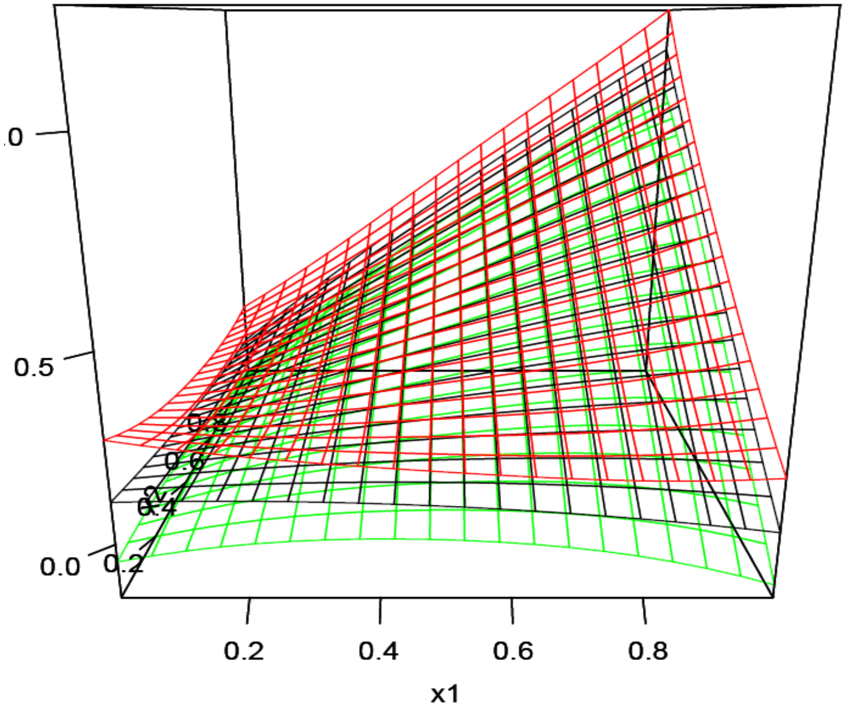

tooth length
0
5
10
15
20
25
30
35
Ascorbic acid
Orange juice
0.5
0.5
1
1
2
2
Vitamin C dose (mg)


Z
X
Y


Cherry
Blueber
Vanilla
Other
Boston Cream
Apple

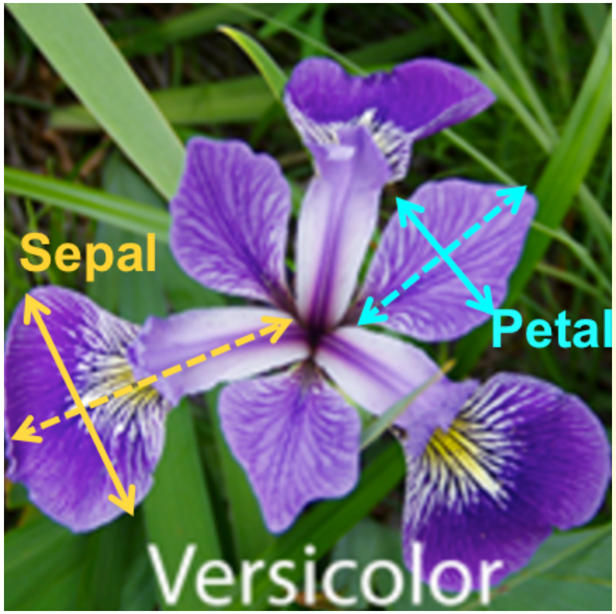


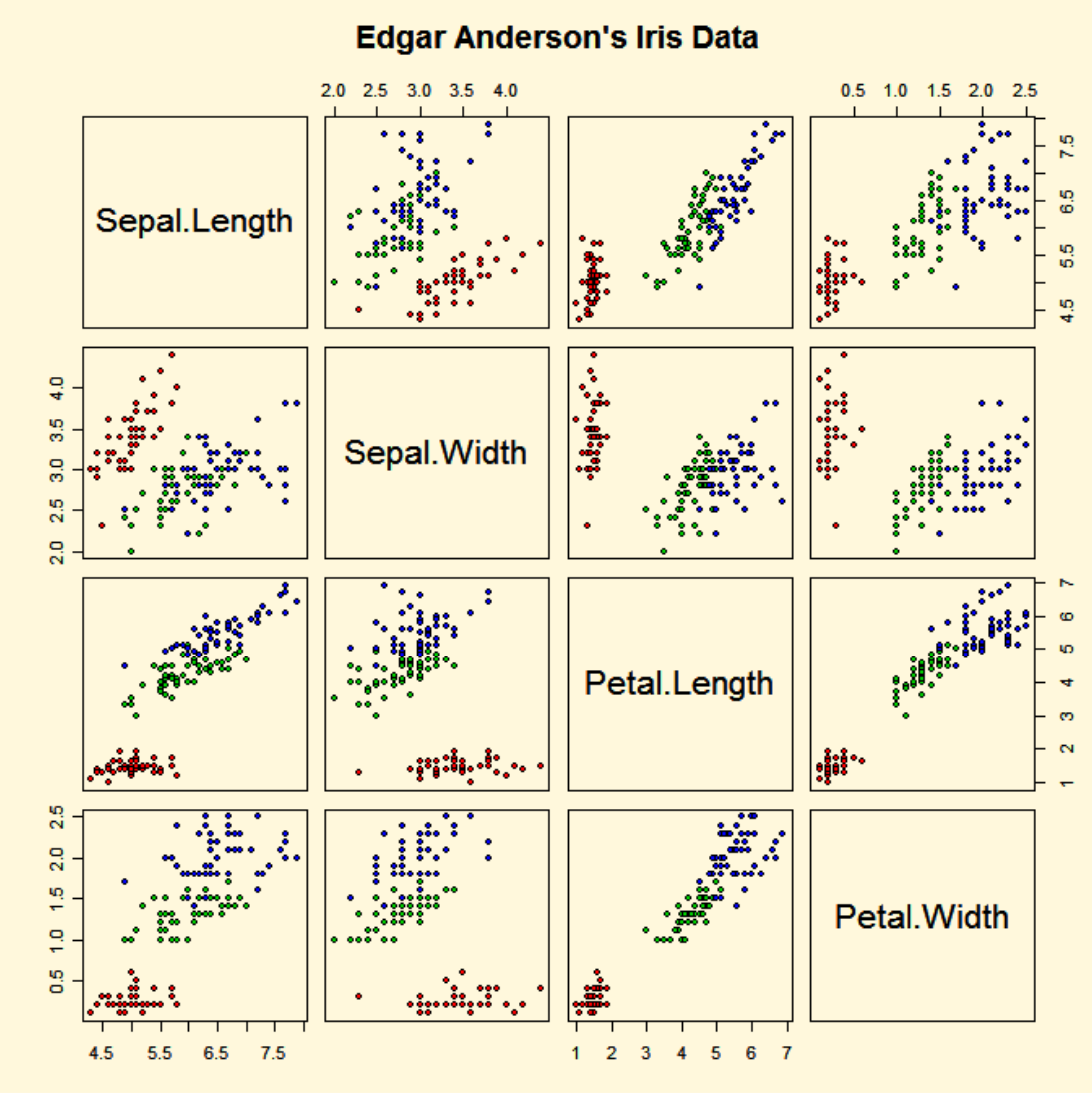
$$y \approx x_0^2 + x_1 x_2 + U(-0.3, 0.3)$$

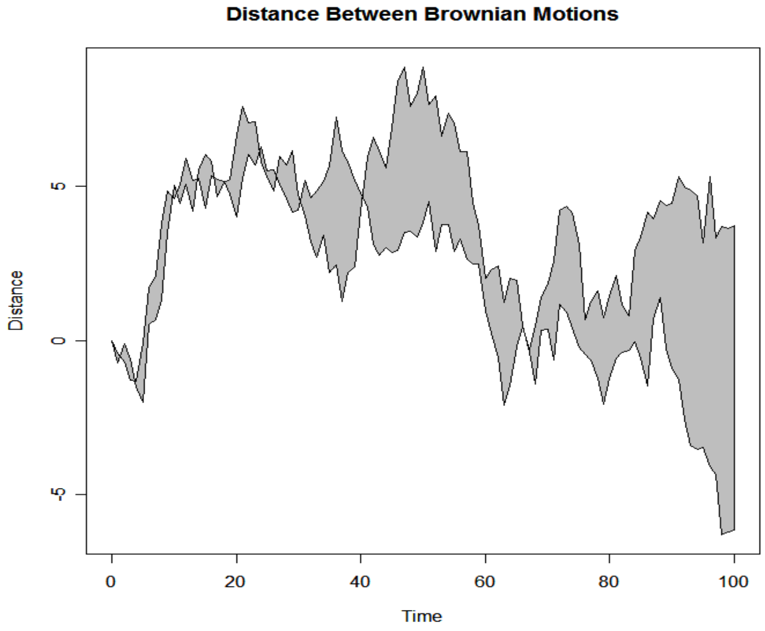


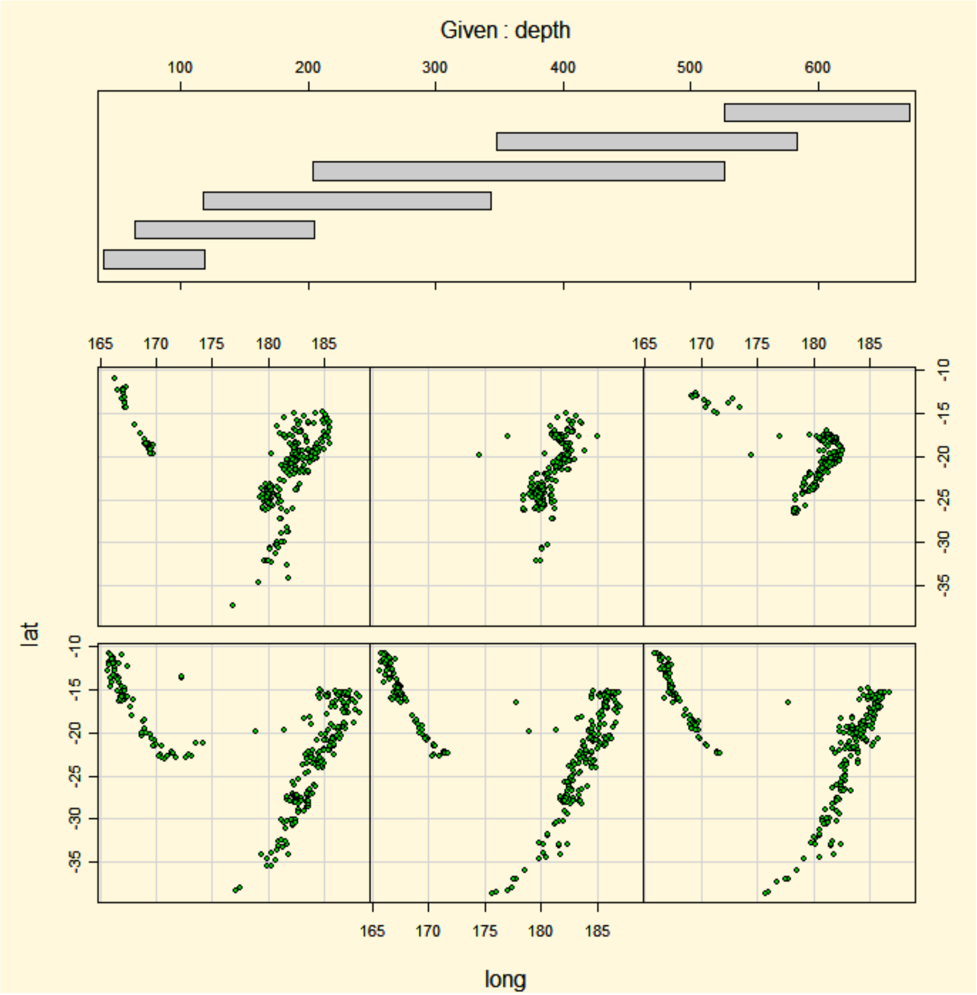
x0=0.25, red/green are +/−2 se

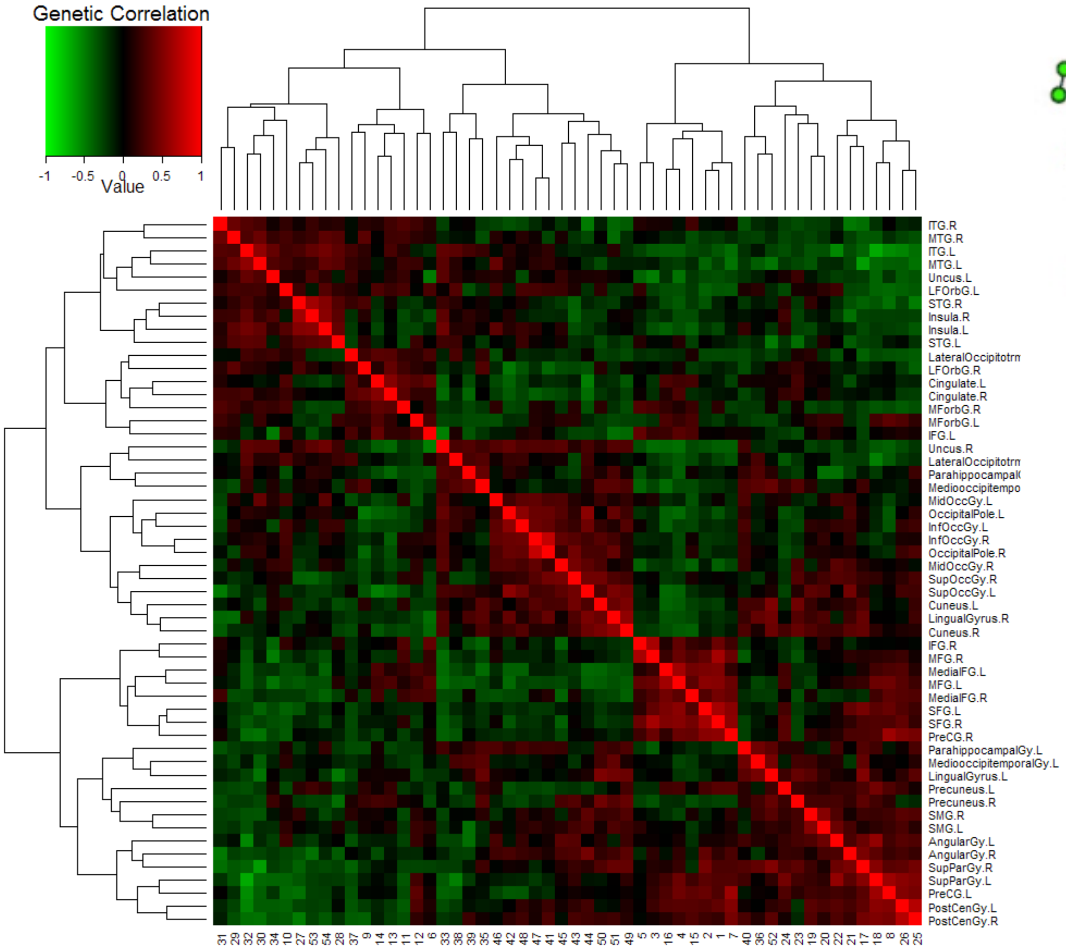
"multi-bivariate" visualization…

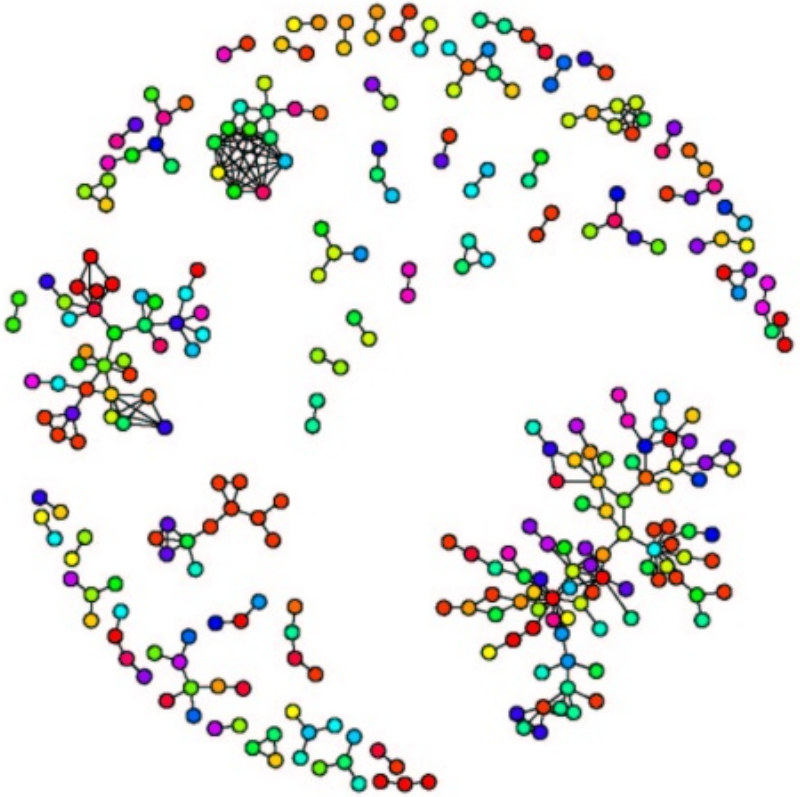


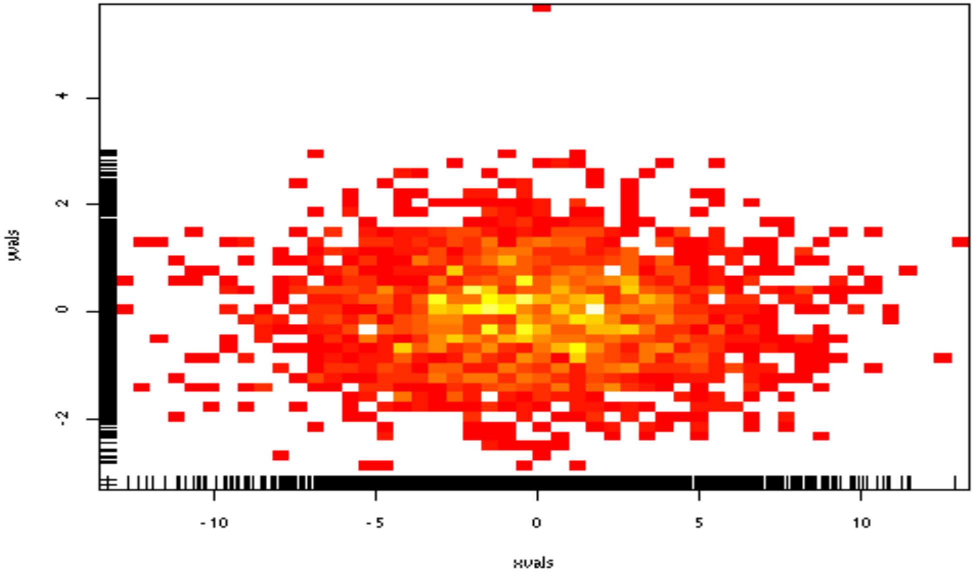


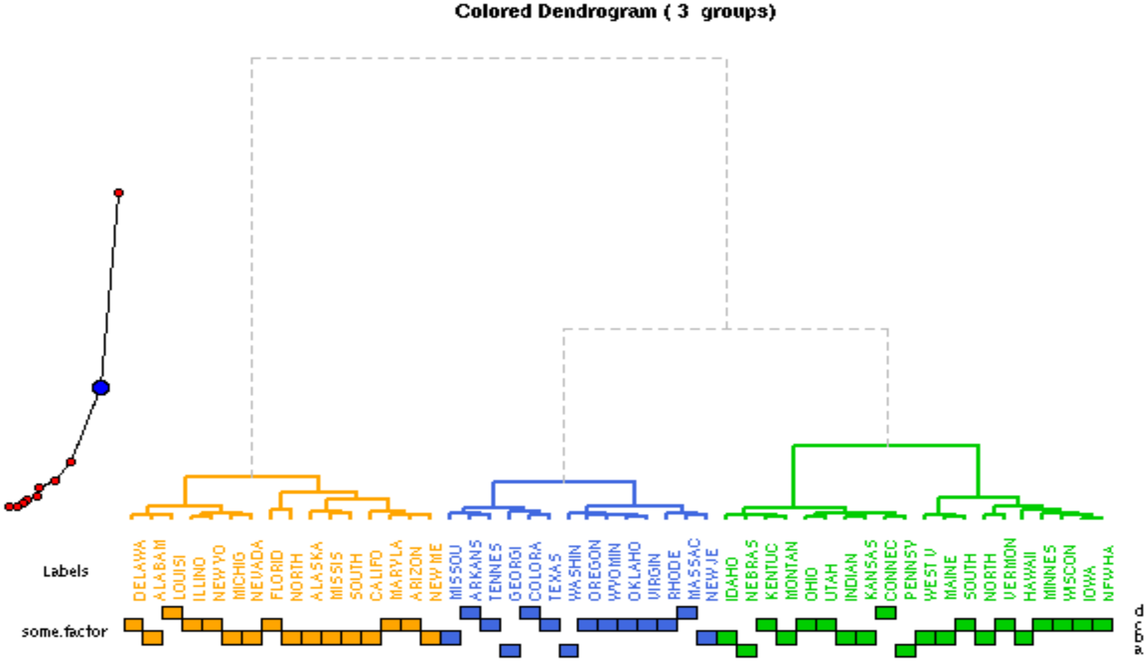

Distance Between Brownian Motions
Distance
-5
0
5
0
20
40
60
80
100
Time


Given : depth
100
200
300
400
500
600
165
170
175
180
185
lat
-10
-15
-20
-25
-30
-35
long

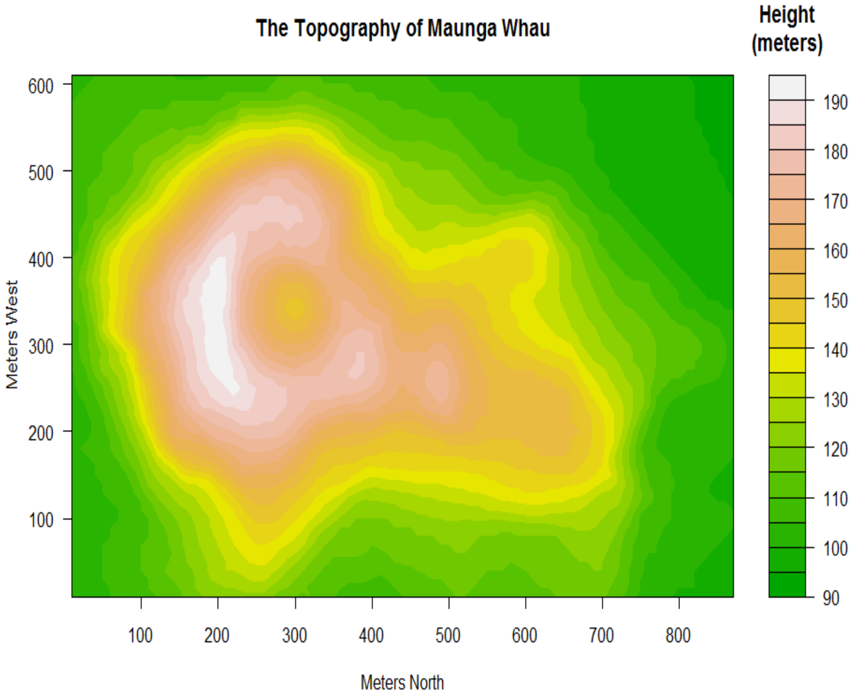

Genetic Correlation
-1
-0.5
0
0.5
1
Value
ITG.R
MTG.R
ITG.L
MTG.L
Uncus.L
LFOrbG.L
STG.R
Insula.R
Insula.L
STG.L
LateralOccipitotm
LFOrbG.R
Cingulate.L
Cingulate.R
MForbG.R
MForbG.L
IFG.L
Uncus.R
LateralOccipitotm
Parahippocampal(
Medioccipitempo
MidOccGy.L
OccipitalPole.L
InfOccGy.L
InfOccGy.R
OccipitalPole.R
MidOccGy.R
SupOccGy.R
SupOccGy.L
Cuneus.L
LingualGyrus.R
Cuneus.R
IFG.R
MFG.R
MedialFG.L
MFG.L
MedialFG.R
SFG.L
SFG.R
PreCG.R
ParahippocampalGy.L
MediooccipitemporalGy.L
LingualGyrus.L
Precuneus.L
Precuneus.R
SMG.R
SMG.L
AngularGy.L
AngularGy.R
SupParGy.R
SupParGy.L
PreCG.L
PostCenGy.L
PostCenGy.R
31 29 32 30 34 10 27 53 54 28 37 9 14 13 11 12 6 33 38 39 35 46 42 48 47 41 45 43 44 50 51 49 5 3 16 4 15 2 1 7 40 36 52 24 23 19 20 22 21 17 18 8 26 25



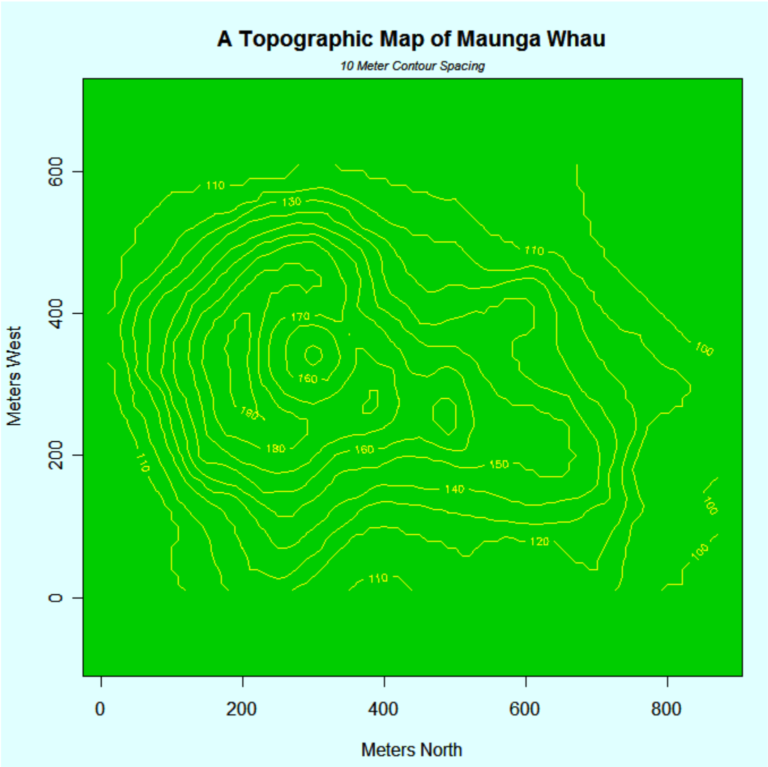

yvals
4
2
0
-2
-10
-5
0
5
10
xvals

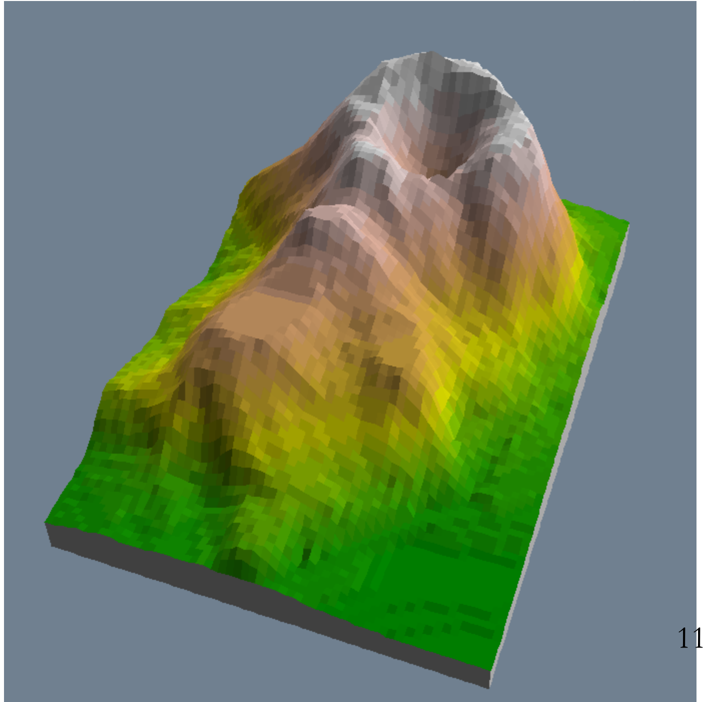

Colored Dendrogram (3 groups)
Labels
DELAWA
ALABAM
LOUISI
ILLINO
NEWYO
MICHIG
NEVADA
FLORID
NORTH
ALASKA
MISSIS
SOUTH
CALIFO
MARYLA
ARIZON
NEW ME
MISSOU
ARKANS
TENNES
GEORGI
COLORA
TEXAS
WASHIN
OREGON
WYOMIN
OKLAHO
VIRGIN
RHODE
MASSAC
NEW JE
IDAHO
NEBRAS
KENTUC
MONTAN
OHIO
UTAH
INDIAN
KANSAS
CONNEC
PENNSY
WEST V
MAINE
SOUTH
NORTH
VERMON
HAWAII
MINNES
WISCON
IOWA
NEWHA
some.factor
d
c
b
a

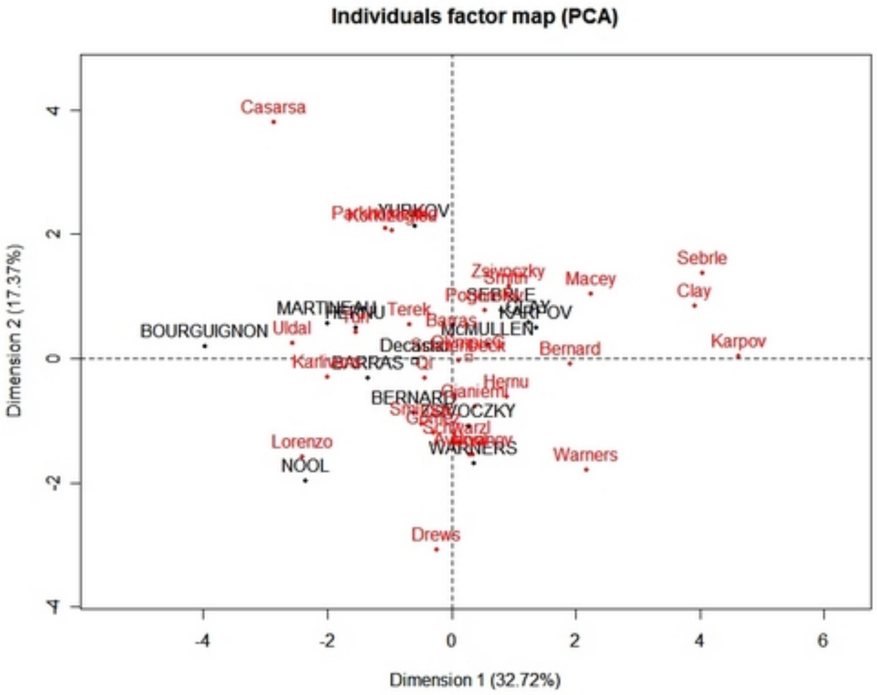
Ooooh….Aaaah…

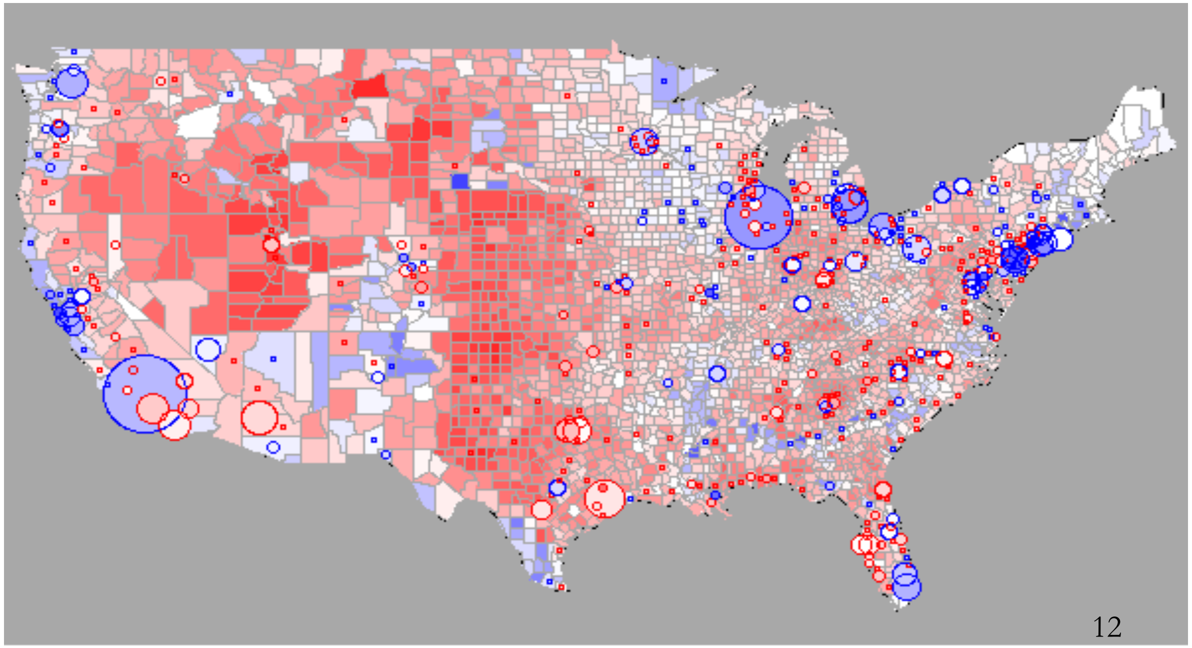


filled.contour(.) from R version 2.9.2 (2009-08-24)






Individuals factor map (PCA)
Casarsa
YURKOV
Parkhomenko
Korkizoglou
Sebrle
Clay
Zsivoczky
Smith
Macey
Pogorelov
SEBRLE
CLAY
KARPOV
MARTINEAU
HERNU
Turi
Terek
Barras
McMULLEN
BOURGUIGNON
Uldal
Decastar
Karpov
Bernard
Karlivans
BARRAS
Qi
Hernu
Ojaniemi
BERNARD
Smirnov
ZSIVOCZKY
Gomez
Schwarzl
Averyanov
WARNERS
Warners
Lorenzo
NOOL
Drews
Dimension 1 (32.72%)
Dimension 2 (17.37%)
-4
-2
0
2
4
6
-4
-2
0
2
4



Examples of high level plotting commands:

- **plot()**
  - hist()
  - boxplot()
  - barplot()
  - dotchart()
  - pie()
  - qqplot()
  - pairs()

- “3D” functions
  - heatmap()
  - image()
  - persp()
  - contour()
  - filled.contour()
  - heatmap.2()

Low-level plotting commands

- points()

- lines()
- abline()
- arrows()
- segments()

- rug()

- …

- text()
- mtext()
- legend()

- polygon()
- grid.rect()

- …

Par

- par can be used to set or query graphical parameters.
- Just like other R objects, par elements are similarly modifiable, with slightly different syntax
  - e.g. par("bg"="lightcyan")
  - This would change the background color of all subsequent plots to light cyan
- When par elements are modified directly (as above, this changes all subsequent plotting behavior
- Some par elements can be modified from within high and low level plotting functions. In this case,

Par parameter examples often modifiable from within plotting functions

- bg – plot background color
- lty – line type (e.g. dot, dash, solid)
- lwd – line width
- pch – dot type
- col – color
- cex – text size inside plot
- mex – text size in margins

- mfcol/mfrow – multiple plot option
  - 2 element vector (#rows,#cols)
  - e.g. par(mfrow=c(1,2))

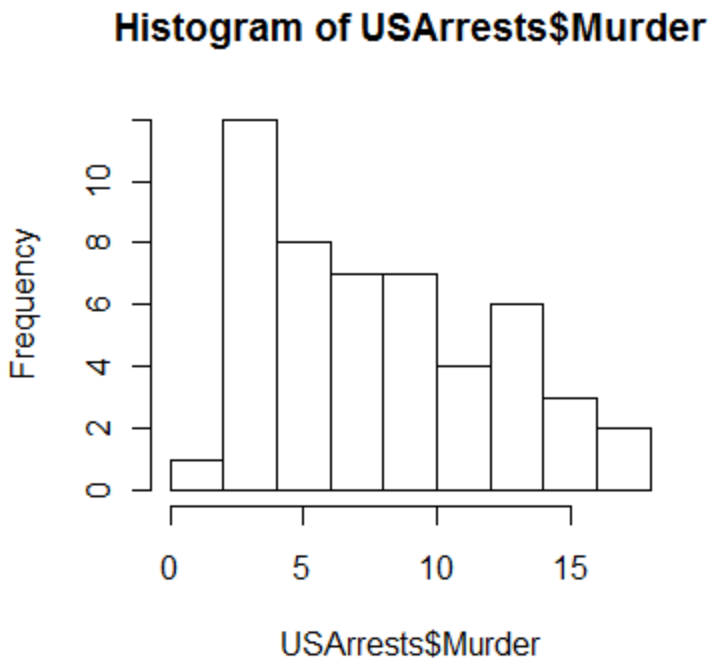
… many, many more (you tend to learn them as you need them)

Devices

- Specify Destination of Graphics Output
- Could be windows in R
- Could be files
  - Not Scalable
    - JPG
    - BMP
    - PNG
  - Scalable:
    - Postscript
    - pdf
  - Others
    - Win.metafile
- Define resolution/size

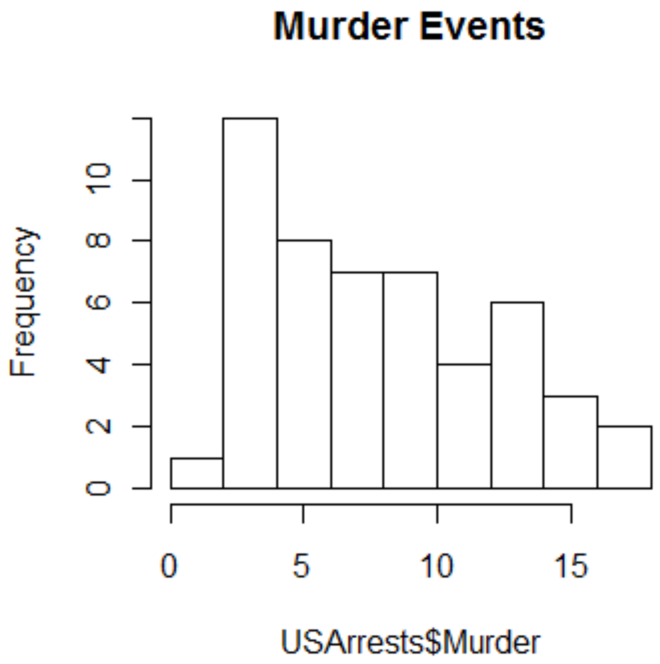
Basic Graphics

- data set: USArrests
- Histogram
  -
    ```
    >hist(USArrests$Murder)
    ```



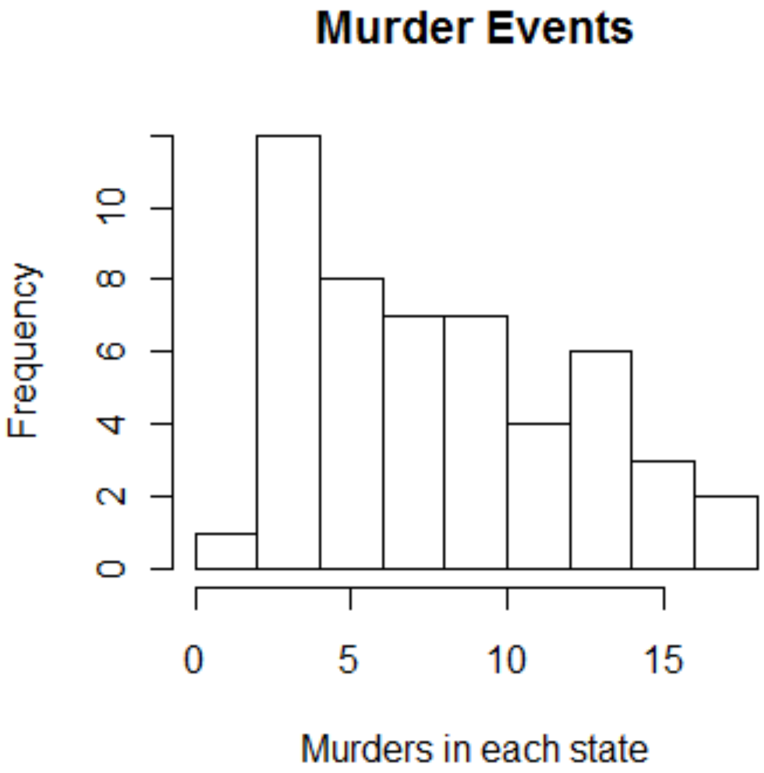
Basic Graphics

- Add a title…
  - The “main” statement will give the plot an overall heading.
  - `>hist(USArrests $Murder, main="Murder Events")`



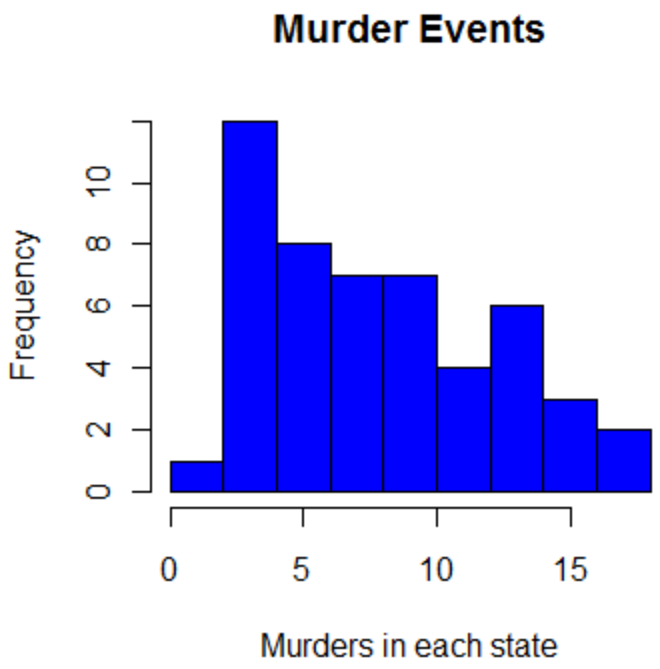
Basic Graphics

- Adding axis labels…
- Use “xlab” and “ylab” to label the X and Y axes, respectively.
- `>hist(USArrests $Murder, main="Murder Events", xlab="Murders in each state")`

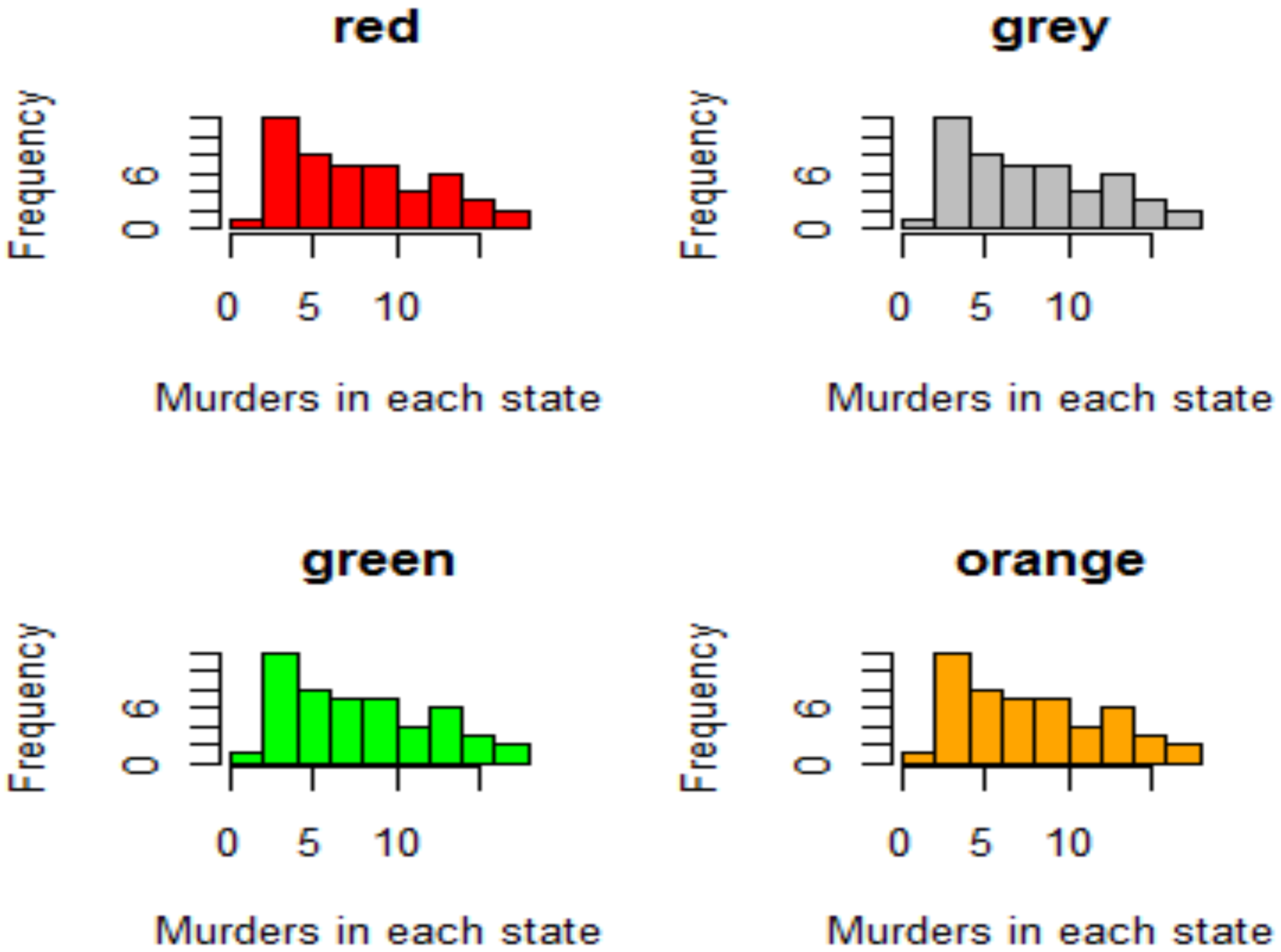


Basic Graphics

- Changing colors…
- Use the col statement.
  - ?colors will give you help on the colors.
  - Common colors may simply put in using the name.
  -
    ```
    >hist(USArrests
    $Murder,
    main="Murder
    Events",
    xlab="Murders in
    each state",
    col="blue")
    ```

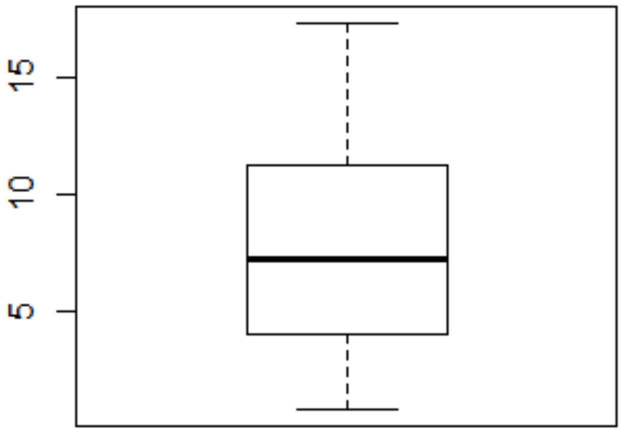


More colors


red
Frequency
0 6
0 5 10
Murders in each state
grey
Frequency
0 6
0 5 10
Murders in each state
green
Frequency
0 6
0 5 10
Murders in each state
orange
Frequency
0 6
0 5 10
Murders in each state

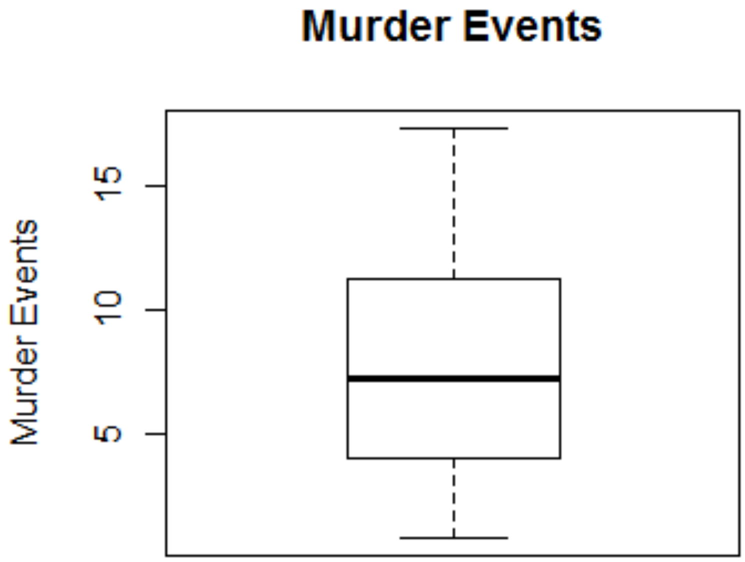
Basic Plots

- Box Plots
- `>boxplot(USArrests$Murder)`



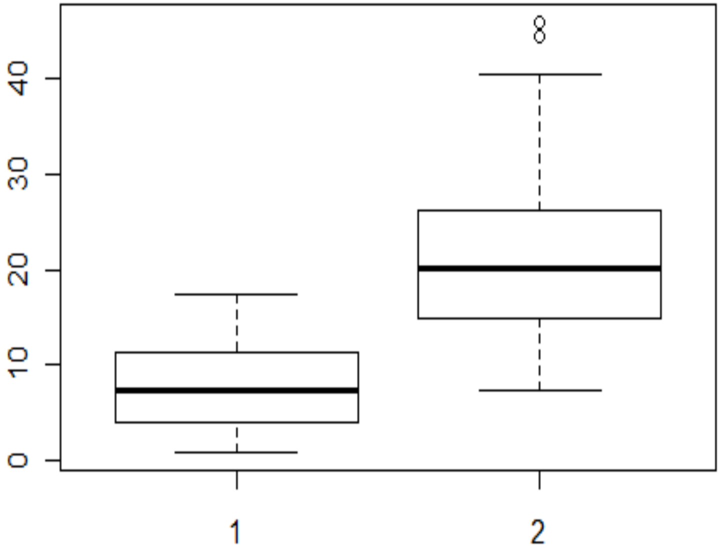
Boxplots

- Change it!
-
```
>boxplot(USArrests$Murder, main="Murder Events", ylab="Murder Events")
```

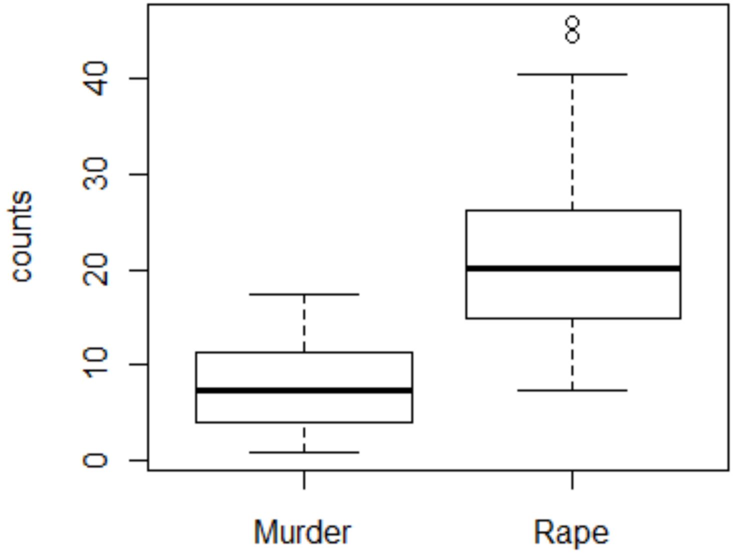


Box-Plots - Groupings

- What if we want several box plots side by side to be able to compare them.
- `>boxplot(USArrests$Murder, USArrests$Rape)`



Boxplots - Groupings



```
> boxplot(USArrests$Murder, USArrests$Rape,
ylab="counts", names=c("Murder", "Rape"))
```

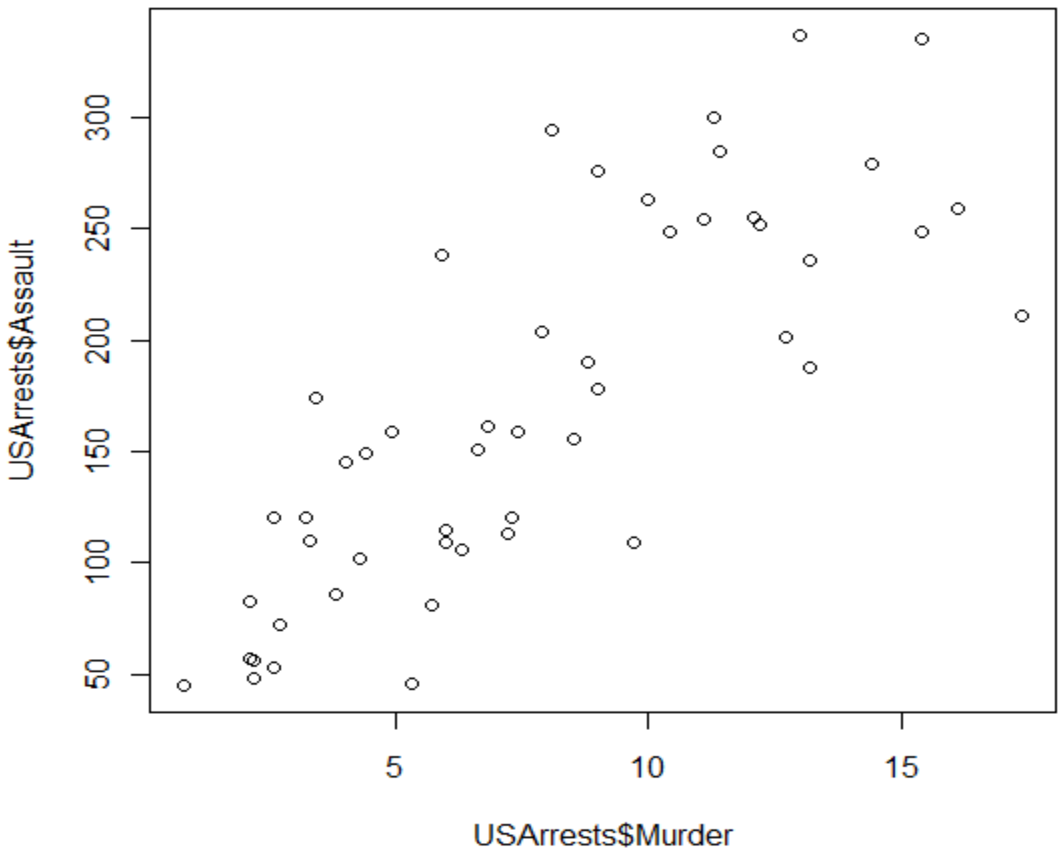
Scatter Plots

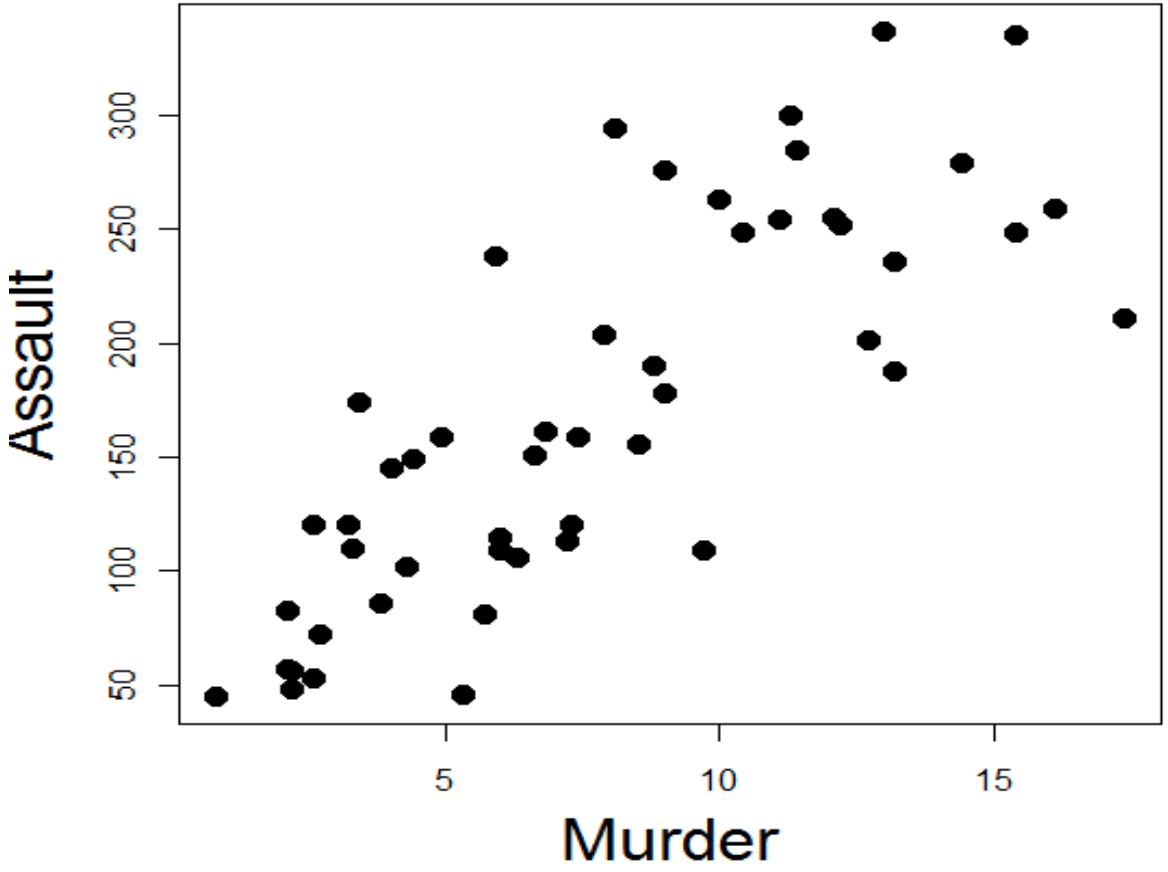
- Suppose we have two variables and we wish to see the relationship between them.
- A scatter plot works very well.
- R code:

  -
    ```
    plot(x,y)
    ```
- Example

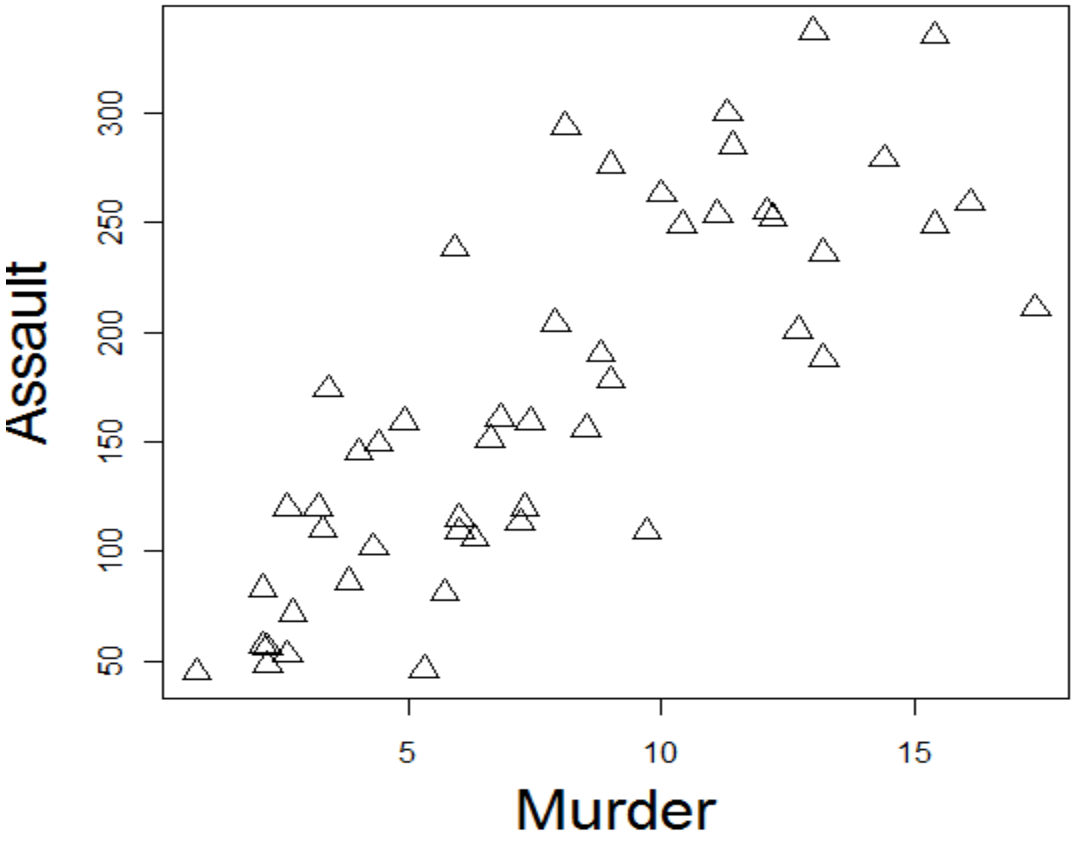
  -
    ```
    >plot(USArrests$Murder, USArrests$Assault)
    ```

Scatterplots





```
> plot(USArrests$Murder, USArrests$Assault, xlab="Murder",
ylab="Assault", cex=1.5, pch=19, cex.lab=2)
```



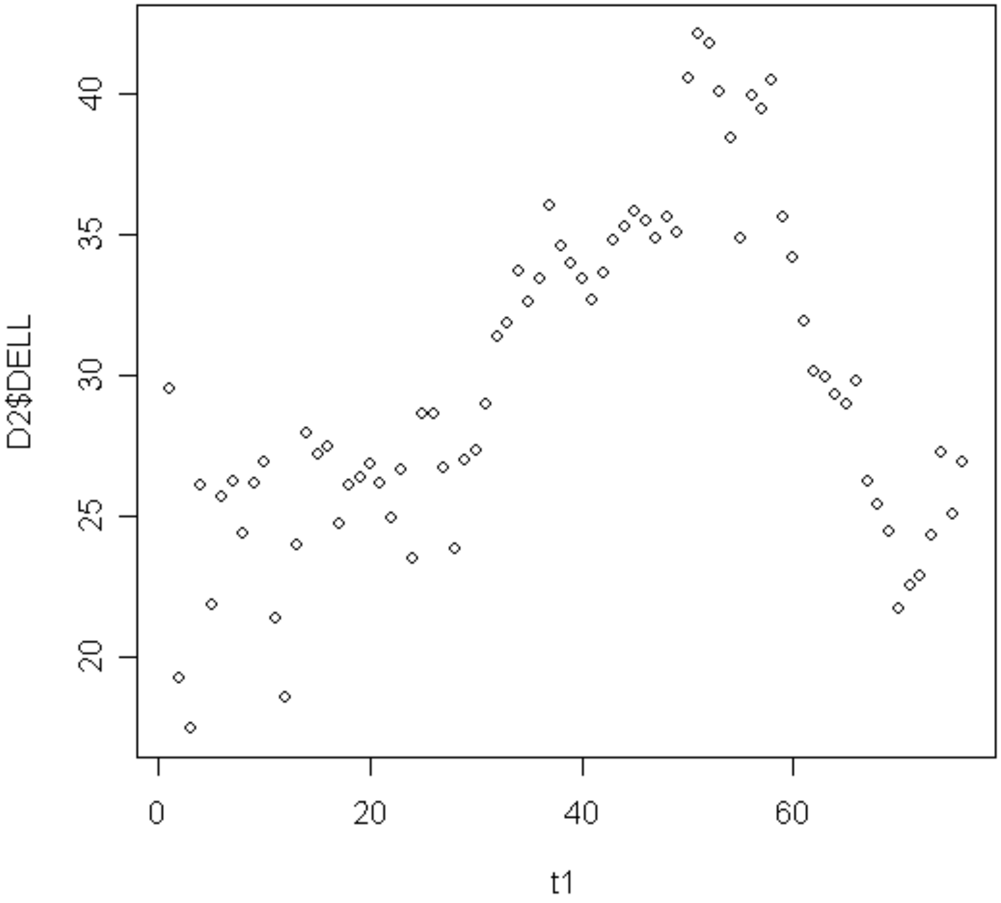
```
> plot(USArrests$Murder, USArrests$Assault, xlab="Murder",
ylab="Assault", cex=1.5, pch=2, cex.lab=2)
```

Line Plots

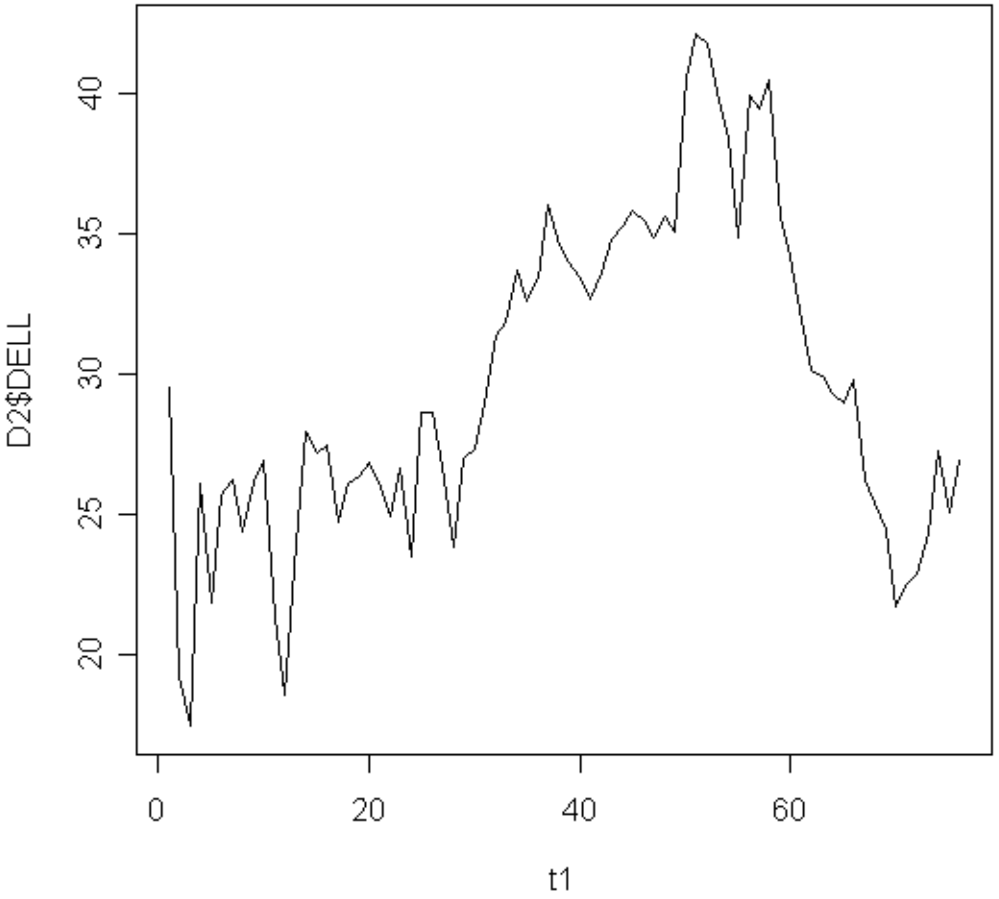
- Often data comes through time.
- Consider Dell stock

```
D2 <- read.csv("H:\\Dell.csv",header=TRUE)
t1 <- 1:nrow(D2)
plot(t1,D2$DELL)
```

Line Plots

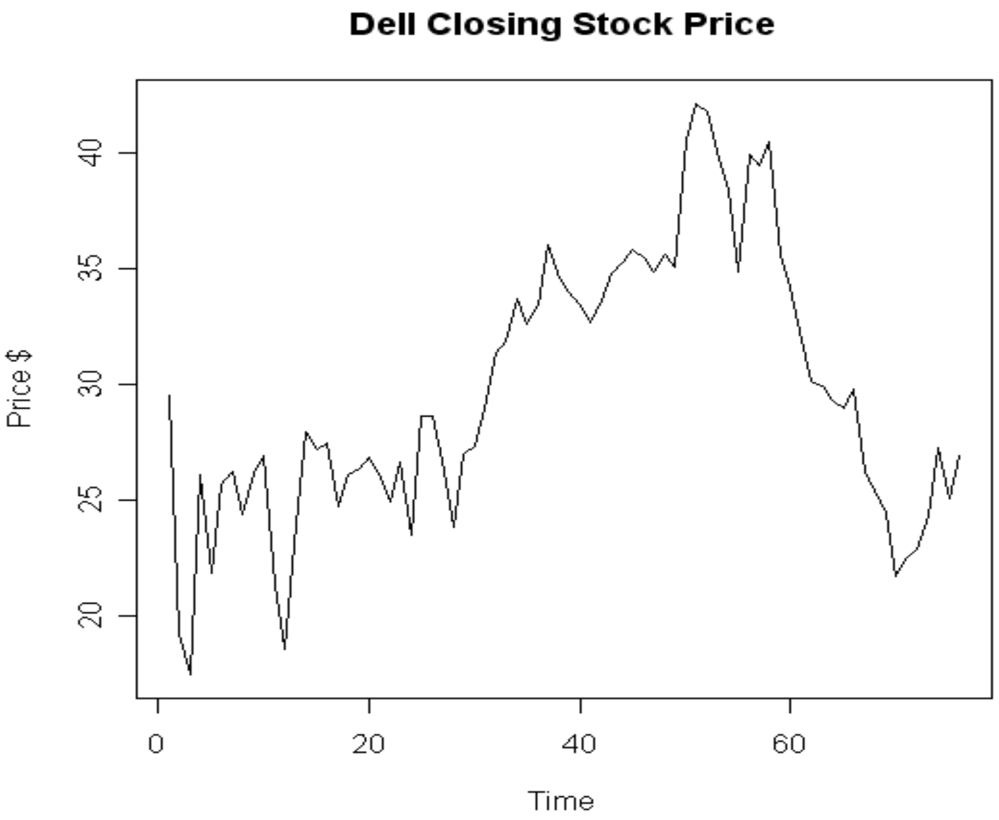


Line Plots



```
plot(t1,D2$DELL,type="l")
```

Line Plots



```
>plot(t1,D2$DELL,type="l",main='Dell Closing Stock Price',
xlab='Time',ylab='Price $'))
```

Basic graphics functions in R

Axis	Generic function to add an Axis to a Plot
abline	**Add Straight Lines to a Plot**
arrows	Add Arrows to a Plot
assocplot	Association Plots
axTicks	Compute Axis Tickmark Locations
axis	Add an Axis to a Plot
axis.POSIXct	Date and Date-time Plotting Functions
barplot	**Bar Plots**
box	Draw a Box around a Plot
boxplot	**Box Plots**
bxp	Draw Box Plots from Summaries
cdplot	Conditional Density Plots
contour	Display Contours
coplot	Conditioning Plots
curve	Draw Function Plots
dotchart	**Cleveland Dot Plots**
filled.contour	Level (Contour) Plots
fourfoldplot	Fourfold Plots
frame	Create / Start a New Plot Frame
graphics-package	The R Graphics Package
grid	Add Grid to a Plot
hist	**Histograms**
hist.POSIXt	Histogram of a Date or Date-Time Object

Basic graphics functions in R

identify	Identify Points in a Scatter Plot
image	**Display a Color Image**
layout	Specifying Complex Plot Arrangements
legend	**Add Legends to Plots**
lines	**Add Connected Line Segments to a Plot**
locator	Graphical Input
matplot	Plot Columns of Matrices
mosaicplot	Mosaic Plots
mtext	Write Text into the Margins of a Plot
pairs	**Scatterplot Matrices**
panel.smooth	Simple Panel Plot
par	**Set or Query Graphical Parameters**
persp	**Perspective Plots**
pie	**Pie Charts**

Basic graphics functions in R

points	**Add Points to a Plot**
polygon	Polygon Drawing
rect	Draw One or More Rectangles
rug	Add a Rug to a Plot
screen	Creating and Controlling Multiple Screens on a Single Device
segments	Add Line Segments to a Plot
spineplot	Spine Plots and Spinograms
stars	Star (Spider/Radar) Plots and Segment Diagrams
stem	Stem-and-Leaf Plots
stripchart	1-D Scatter Plots
strwidth	Plotting Dimensions of Character Strings and Math Expressions
sunflowerplot	Produce a Sunflower Scatter Plot
symbols	Draw Symbols (Circles, Squares, Stars, Thermometers, Boxplots) on a Plot
text	**Add Text to a Plot**
title	**Plot Annotation**
xinch	Graphical Units

Practice example

Edgar Anderson's Iris Data

This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are *Iris setosa*, *versicolor*, and *virginica*.

```
> iris
```

Practice example -2

```
>head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> tail(iris)
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
```

Practice example -3

```
>class(iris)                          -- check the class attribute
>names(iris)                          -- variable names in the data
>pairs(iris, col=iris$Species)        -- draw pair-wise plot

>getwd()                              -- find the current working directory
>postscript(“iris.pairwiseplot.ps     -- define the output file name and device
   ”)
>pairs(iris, col=iris$Species)        -- plot
>dev.off()                            -- close the device

>postscript(“c:\mcb                   -- specify the path of the output file
   \iris.pairwiseplot.ps”)
>pairs(iris, col=iris$Species)
>dev.off()
```

Practice example -4

```
>hist(iris$Sepal.Length)
>with(iris, hist(Sepal.Length))

> with(iris, hist(Sepal.Length, probability=TRUE))
> with(iris, lines(density(Sepal.Length)))
>

> boxplot(iris$Sepal.Length)
> boxplot(iris$Sepal.Length, xlab="Sepal.Length")
> boxplot(iris)
> boxplot(iris[,-5])
>
```

Practice example -5

```
> library(lattice)
> trellis.device(theme = "col.whitebg")
>xyplot(Sepal.Length ~ Sepal.Width, groups = Species, data =
   iris)
> xyplot(Sepal.Length ~ Sepal.Width, groups = Species, data =
   iris, auto.key=list(points=T))
> xyplot(Sepal.Length ~ Sepal.Width, groups = Species, data =
   iris, auto.key=list(points=T,columns=3))


 > xyplot(Sepal.Length ~ Sepal.Width | Species, type = c("p",
   "smooth"), data = iris, pch = 17, col = 2, main = "Iris",
   span=2, layout=c(3,1))
>
```

More example: Volcano

```
> require(grDevices) # for colours
> filled.contour(volcano, color = terrain.colors, asp = 1)# simple

> x <- 10*1:nrow(volcano)
> y <- 10*1:ncol(volcano)
> filled.contour(x, y, volcano, color = terrain.colors, plot.title =
title(main = "The Topography of Maunga Whau", xlab = "Meters North",
ylab = "Meters West"), plot.axes = { axis(1, seq(100, 800, by = 100))
     axis(2, seq(100, 600, by = 100)) }, key.title = title(main="Height
   \n(meters)"),           key.axes = axis(4, seq(90, 190, by = 10)))
> mtext(paste("filled.contour(.) from", R.version.string), side = 1,
line = 4, adj=1, cex = .66)
```

# Annotating a filled contour plot

```
> a <- expand.grid(1:20, 1:20)
> b <- matrix(a[,1] + a[,2], 20)
> filled.contour(x = 1:20, y = 1:20, z = b, plot.axes={ axis(1);
axis(2); points(10,10) })
```

More example: Persian Rug Art

```
> x <- seq(-4*pi, 4*pi, len = 27)
> y<-x
> r <- sqrt(outer(x^2, y^2, "+"))
> filled.contour(cos(r^2)*exp(-r/(2*pi)), axes =
FALSE)
 ## rather, the key *should* be labeled:
> filled.contour(cos(r^2)*exp(-r/(2*pi)),
frame.plot = FALSE, plot.axes = {})
```