DATA ANALYTICS

Unit 1:Data Visualization and R Graphics

Mamatha.H.R

Department of Computer Science and Engineering

DATA ANALYTICS

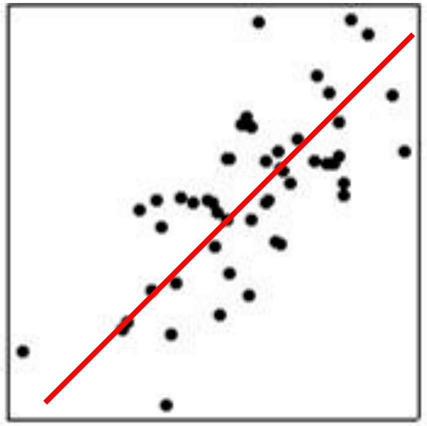
Unit 1:Data Visualization and R Graphics

Mamatha H R

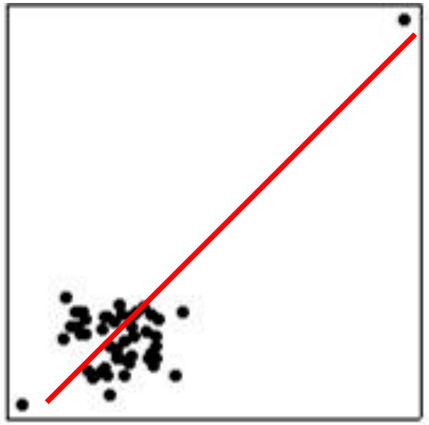
Department of Computer Science and Engineering

Graphics: Why plot your data?

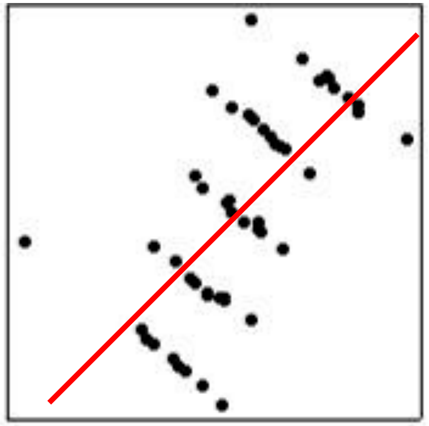
- Three data sets with exactly the same bivariate summary statistics:
  - Same correlations, linear regression lines, etc
  - Indistinguishable from standard printed output



Standard data



r=0 but + 2 outliers



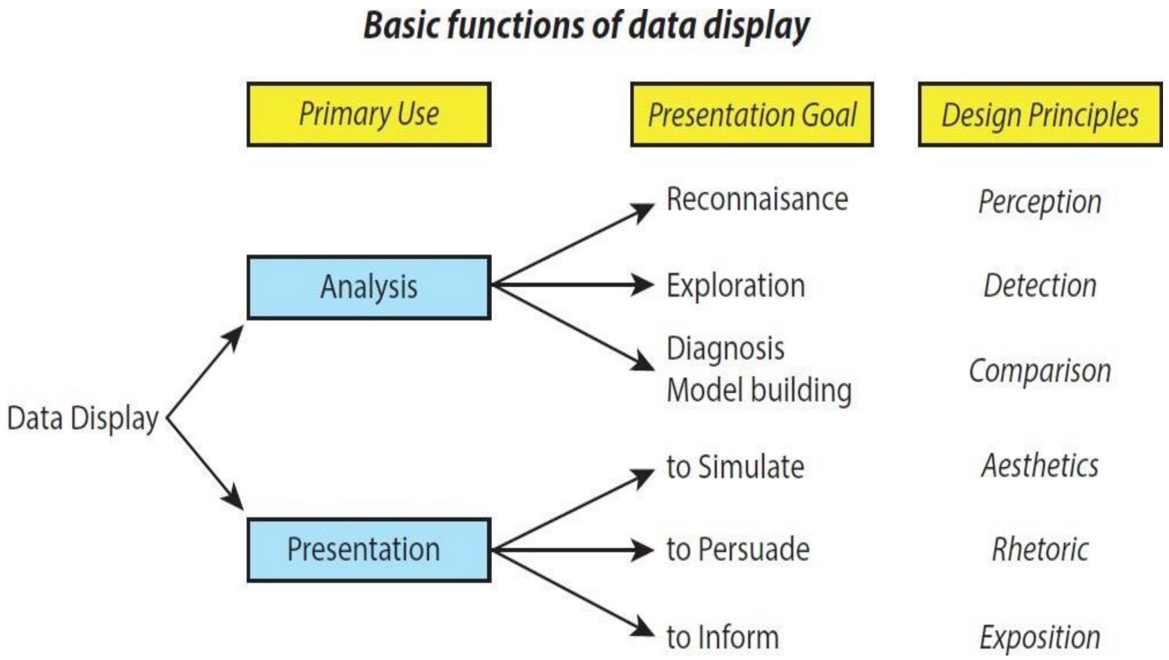
Lurking variable?

Roles of graphics in data analysis

- Graphs (& tables) are forms of communication:
  - What is the audience?
  - What is the message?

Analysis graphs: design to see patterns, trends, aid the process of data description, interpretation

Presentation graphs: design to attract attention, make a point, illustrate a conclusion



Basic functions of data display

Primary Use | *Presentation Goal* | *Design Principles*

Data Display → Analysis → Reconnaisance, Exploration, Diagnosis Model building

Data Display → Presentation → to Simulate, to Persuade, to Inform

Perception
Detection
Comparison
Aesthetics
Rhetoric
Exposition

The 80-20 rule: Data analysis

- Often ~80% of data analysis time is spent on data preparation and data cleaning
  1. data entry, importing data set to R, assigning factor labels,
  2. data screening: checking for errors, outliers, …
  3. Fitting models & diagnostics: whoops! Something wrong, go back to step 1
- Whatever you can do to reduce this, gives more time for:
  - Thoughtful analysis,
  - Comparing models,
  - Insightful graphics,
  - Telling the story of your results and conclusions

The 80-20 rule: Graphics

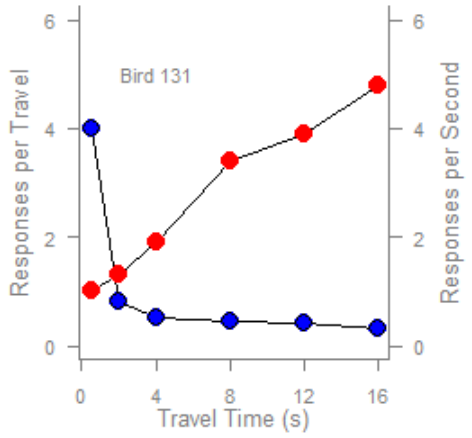
- **Analysis graphs**: Happily, 20% of effort can give 80% of a desired result
  - Default settings for plots often give something reasonable
  - 90-10 rule: Plot annotations (regression lines, smoothed curves, data ellipses, …) add additional information to help understand patterns, trends and unusual features, with only 10% more effort

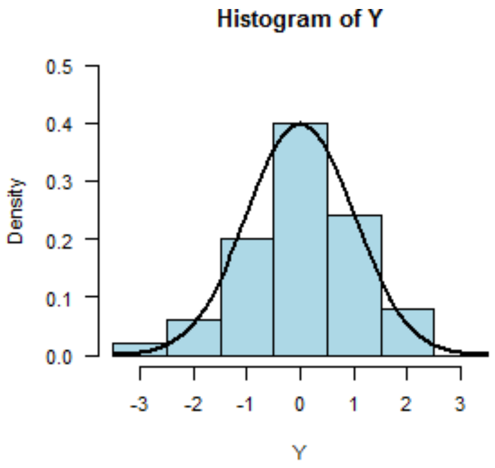
The 80-20 rule: Graphics

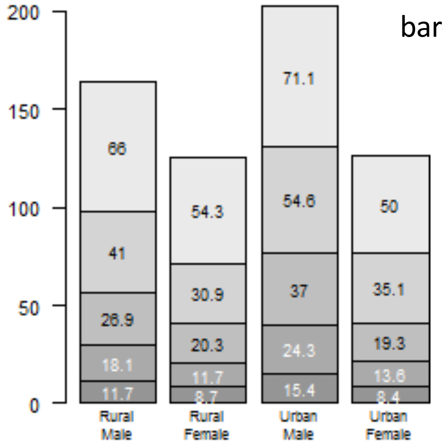
- **Presentation graphs**: Sadly, 80% of total effort may be required to give the remaining 20% of your final graph
  - Graph title, axis and value labels: should be directly readable
  - Grouping attributes: visually distinct, allowing for BW vs color
    - color, shape, size of point symbols;
    - color, line style, line width of lines
  - Legends: Connect the data in the graph to interpretation
  - Aspect ratio: need to consider the H x V size and shape

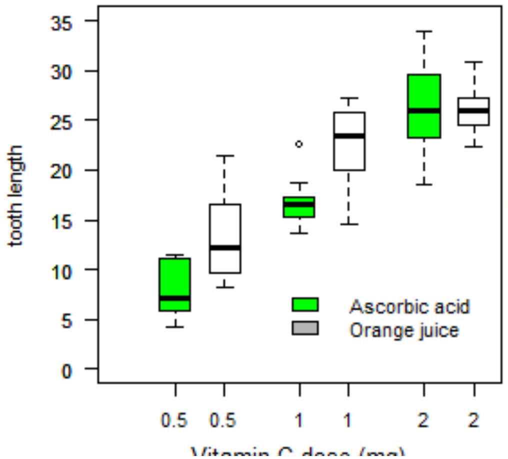
What can I do with R graphics?

A wide variety of standard plots (customized)

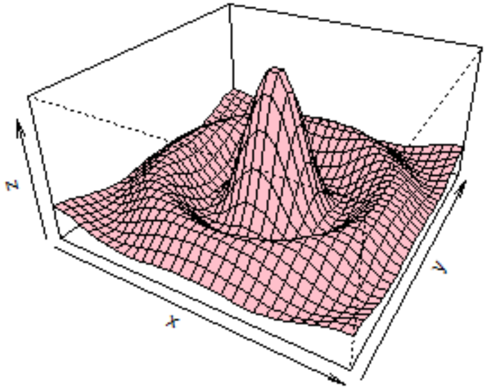




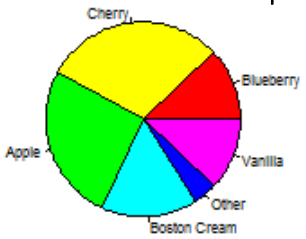

barchart()



3D plot: persp()



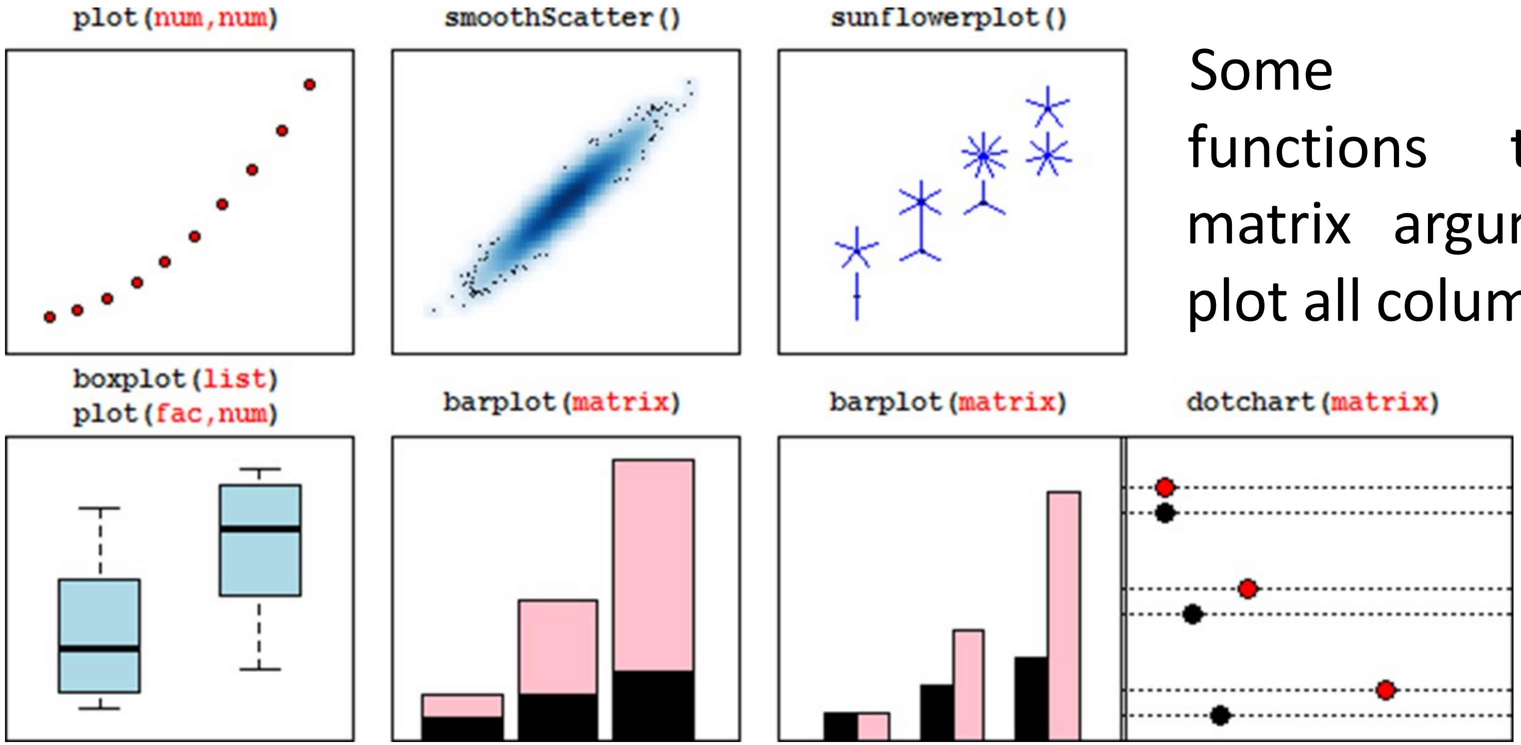
pie()



Bivariate plots

R base graphics provide a wide variety of different plot types for bivariate data

The function **plot(x, y)** is generic. It produces different kinds of plots depending on whether x and y are numeric or factors.

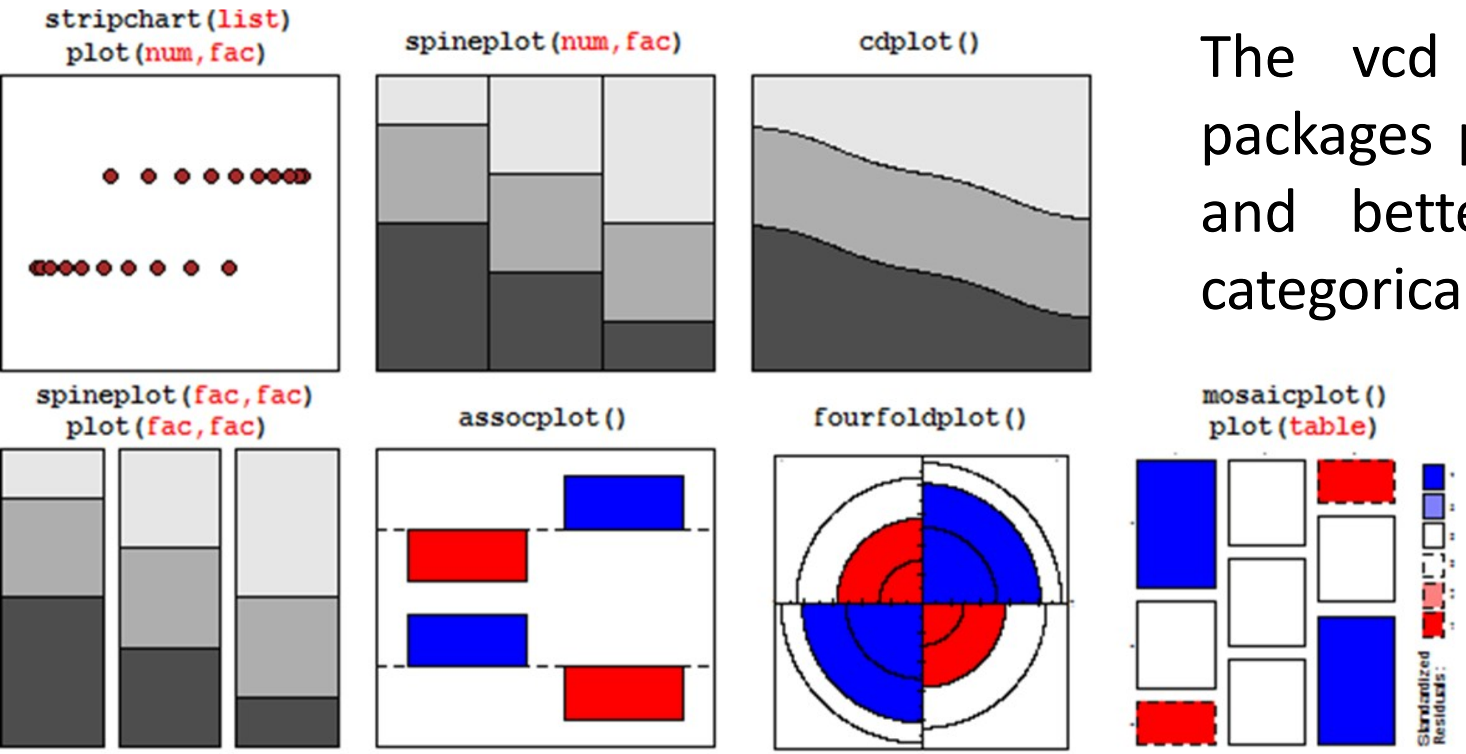


Some plotting functions take a matrix argument & plot all columns

Bivariate plots

A number of specialized plot types are also available in base R graphics

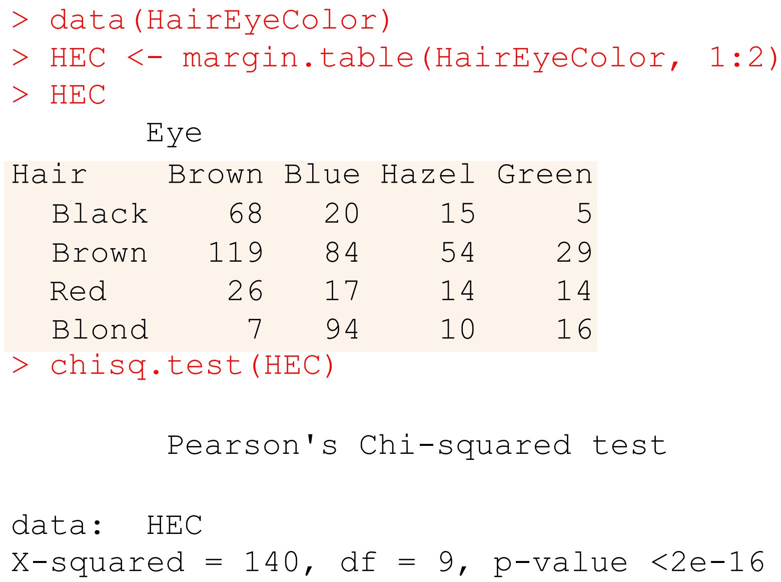
Plot methods for factors and tables are designed to show the association between categorical variables



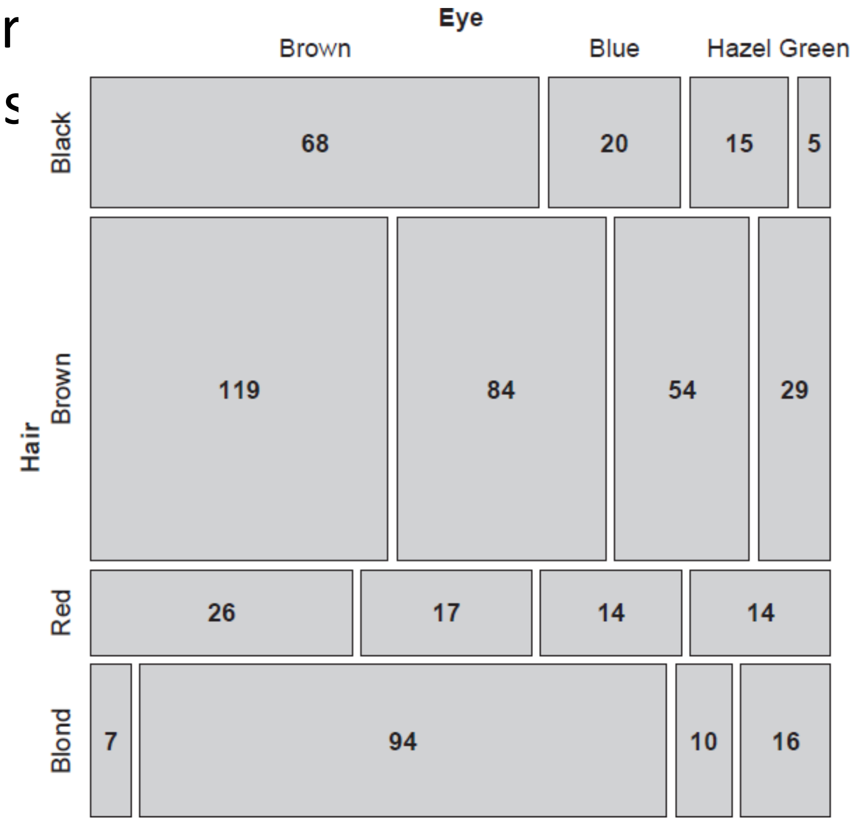
The vcd & vcdExtra packages provide more and better plots for categorical data

Mosaic plots

Similar to a grouped bar char
Shows a frequency table with tile s
area ~ frequency



```
> data(HairEyeColor)
> HEC <- margin.table(HairEyeColor, 1:2)
> HEC
       Eye
Hair    Brown Blue Hazel Green
  Black    68   20    15     5
  Brown   119   84    54    29
  Red      26   17    14    14
  Blond     7   94    10    16
> chisq.test(HEC)

        Pearson's Chi-squared test

data:  HEC
X-squared = 140, df = 9, p-value <2e-16
```



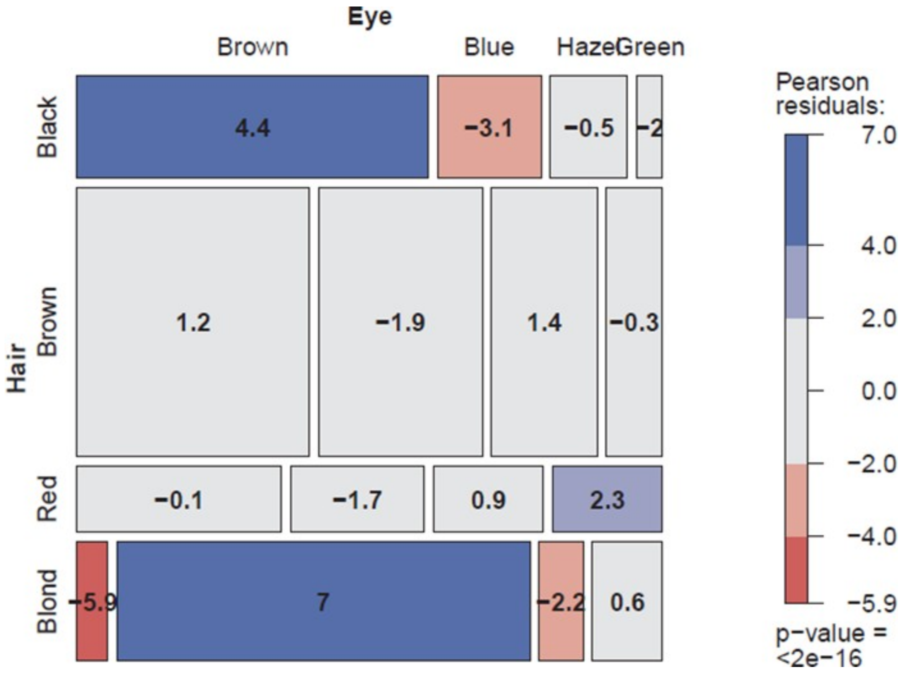
How to understand the association between hair color and eye color?

Mosaic plots

Shade each tile in relation to the contribution to the Pearson $?^2$ statistic

$$\chi^2 = \sum r_{ij}^2 = \sum \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

```
> round(residuals(chisq.test(HEC)),2)
      Eye
Hair        Brown   Blue  Hazel   Green
   Black     4.40  -3.07  -0.48   -1.95
   Brown     1.23  -1.95   1.35   -0.35
  Red       -0.07  -1.73   0.85    2.28
   Blond    -5.85   7.05  -2.23    0.61
```

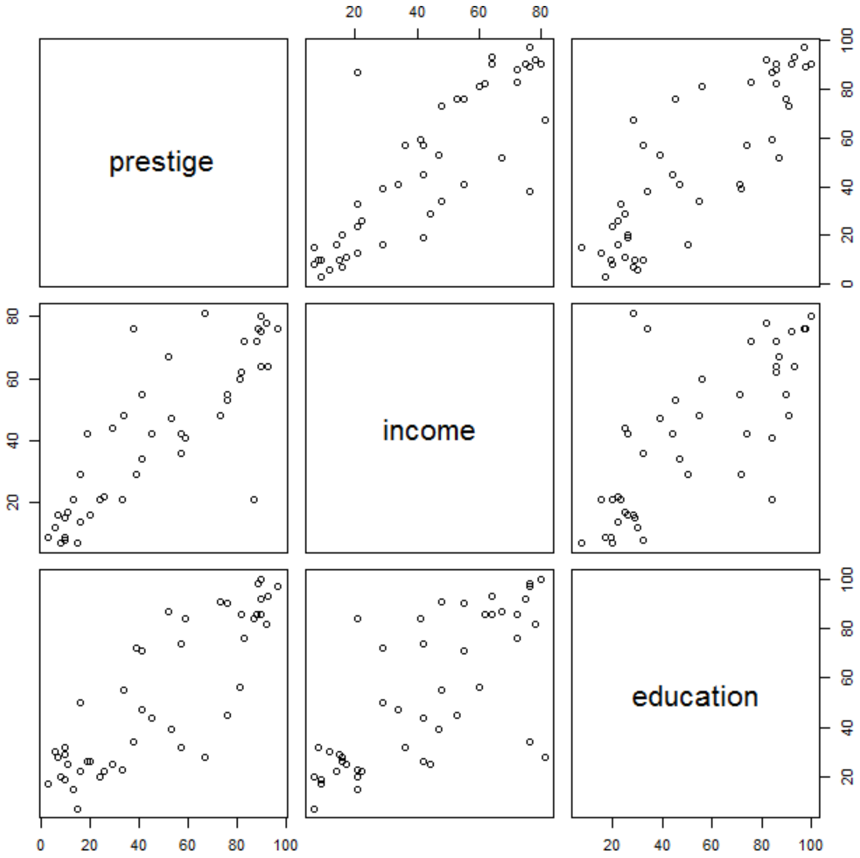


Multivariate plots

The simplest case of multivariate plots is a **scatterplot matrix** – all pairs of bivariate plots

In R, the generic functions **plot()** and **pairs()** have specific methods for data frames

```
data(Duncan, package="car")
plot(~ prestige + income +
      education,
      data=Duncan)
pairs(~ prestige + income +
      education,
      data=Duncan)
```

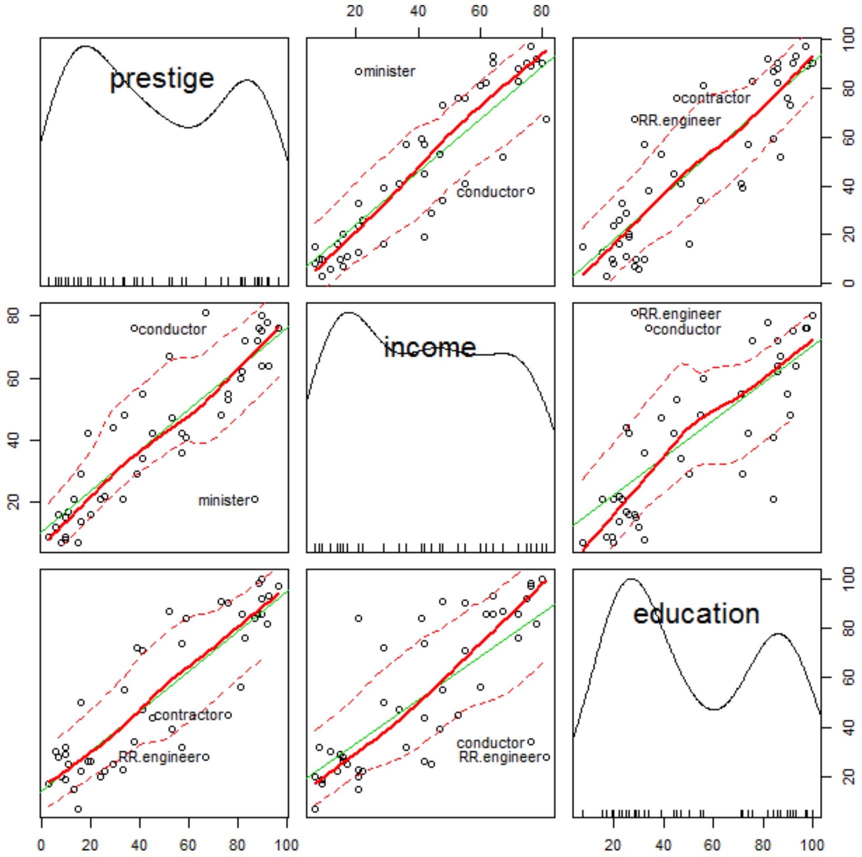


Multivariate plots

These basic plots can be enhanced in many ways to be more informative.

The function scatterplotMatrix() in the car package provides

- univariate plots for each variable
- linear regression lines and loess smoothed curves for each pair
- automatic labeling of noteworthy observations (id.n=)



```
library(car)
scatterplotMatrix(~prestige + income + education,
                                            data=Duncan,
id.n=2)
```

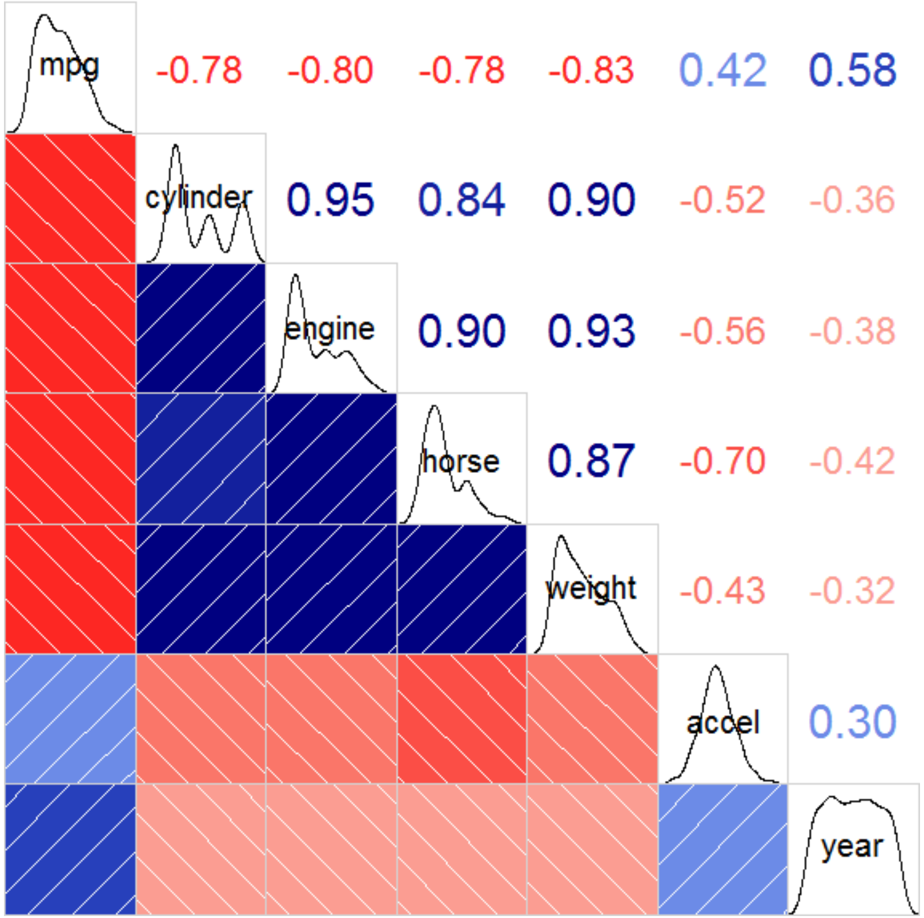
Multivariate plots: corrgrams

For larger data sets, visual summaries are often more useful than direct plots of the raw data

A corrgram (“correlation diagram”) allows the data to be rendered in a variety of ways, specified by panel functions.

Here the main goal is to see how mpg is related to the other variables

```
corrgram(cars[,3:9], upper.panel=panel.cor, diag.panel=panel.density)
```

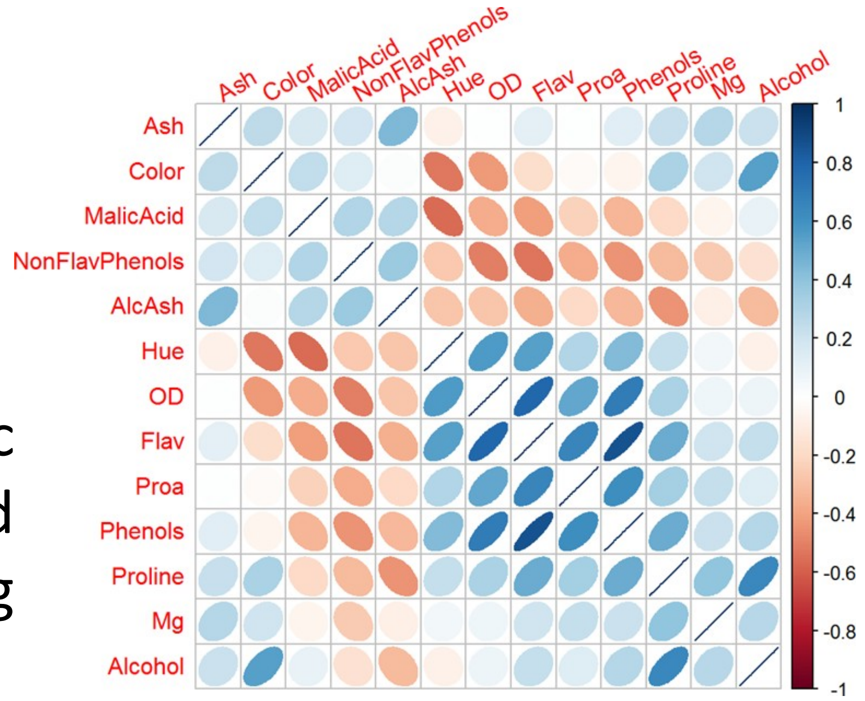


Multivariate plots: corrgrams

For even larger data sets, more abstract visual summaries are necessary to see the patterns of relationships.

This example uses schematic ellipses to show the strength and direction of correlations among variables on a large collection of Italian wines.

Here the main goal is to see how the variables are related to each other.

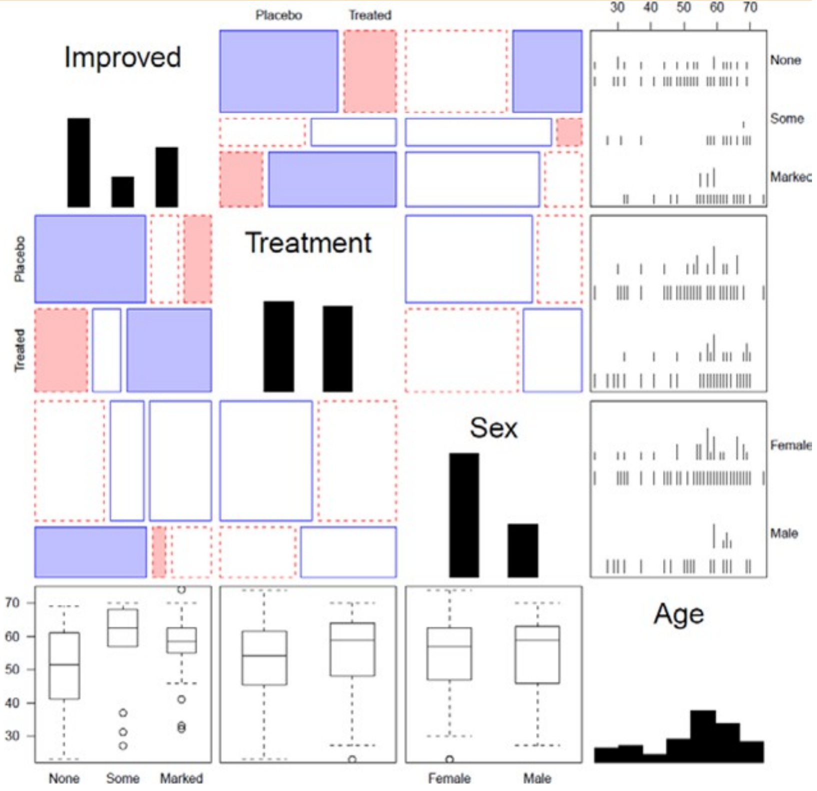


```
library(corrplot)
corrplot(cor(wine), tl.srt=30, method="ellipse", order="AOE")
```

Generalized pairs plots

Generalized pairs plots from the gpairs package handle both categorical (**C**) and quantitative (**Q**) variables in sensible ways

x	y	plot
Q	Q	scatterplot
C	Q	boxplot
Q	C	barcode
C	C	mosaic



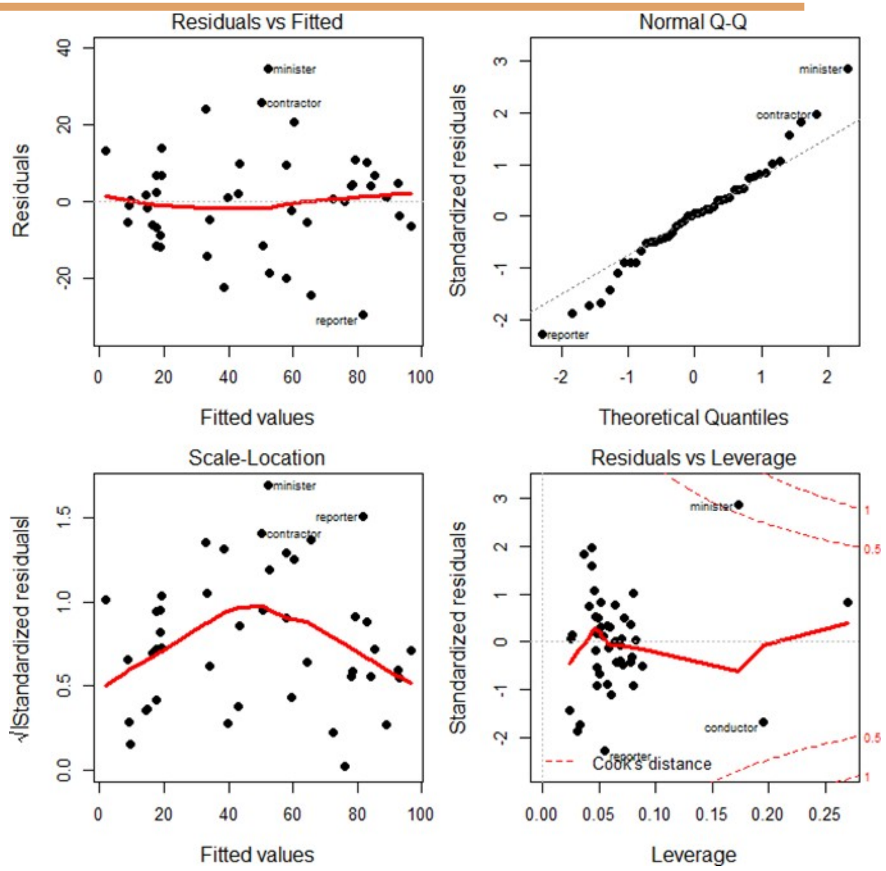
```
library(gpairs)
data(Arthritis)
gpairs(Arthritis[, c(5, 2:5)], …)
```

Models: diagnostic plots

Linear statistical models (ANOVA, regression), $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, require some assumptions: $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \boldsymbol{\sigma}^2)$

For a fitted model object, the plot() method gives some useful diagnostic plots:

- residuals vs. fitted: any pattern?
- Normal QQ: are residuals normal?
- scale-location: constant variance?
- residual-leverage: outliers?



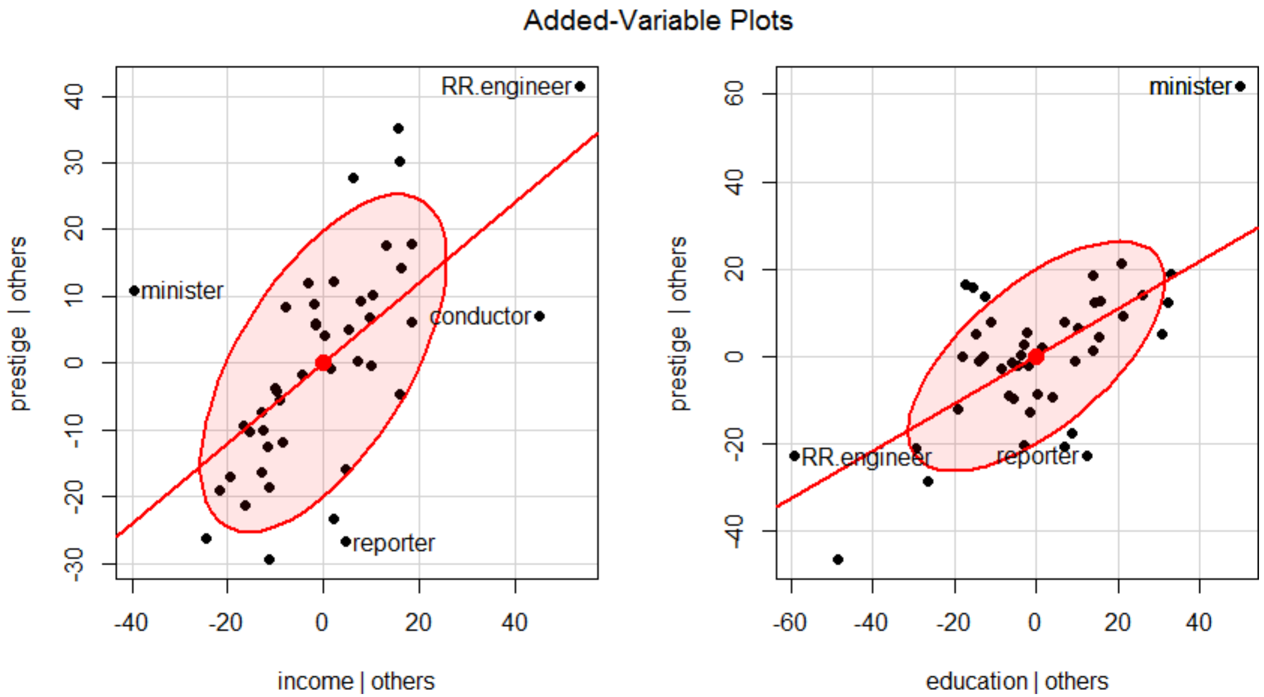
```
duncan.mod <- lm(prestige ~ income +
education, data=Duncan)
plot(duncan.mod)
```

Models: Added variable plots

The car package has many more functions for plotting linear model objects

Among these, added variable plots show the partial relations of y to each x, holding all other predictors constant.

```
library(car)
avPlots(duncan.mod, id.n=2,ellipse=TRUE, …)
```



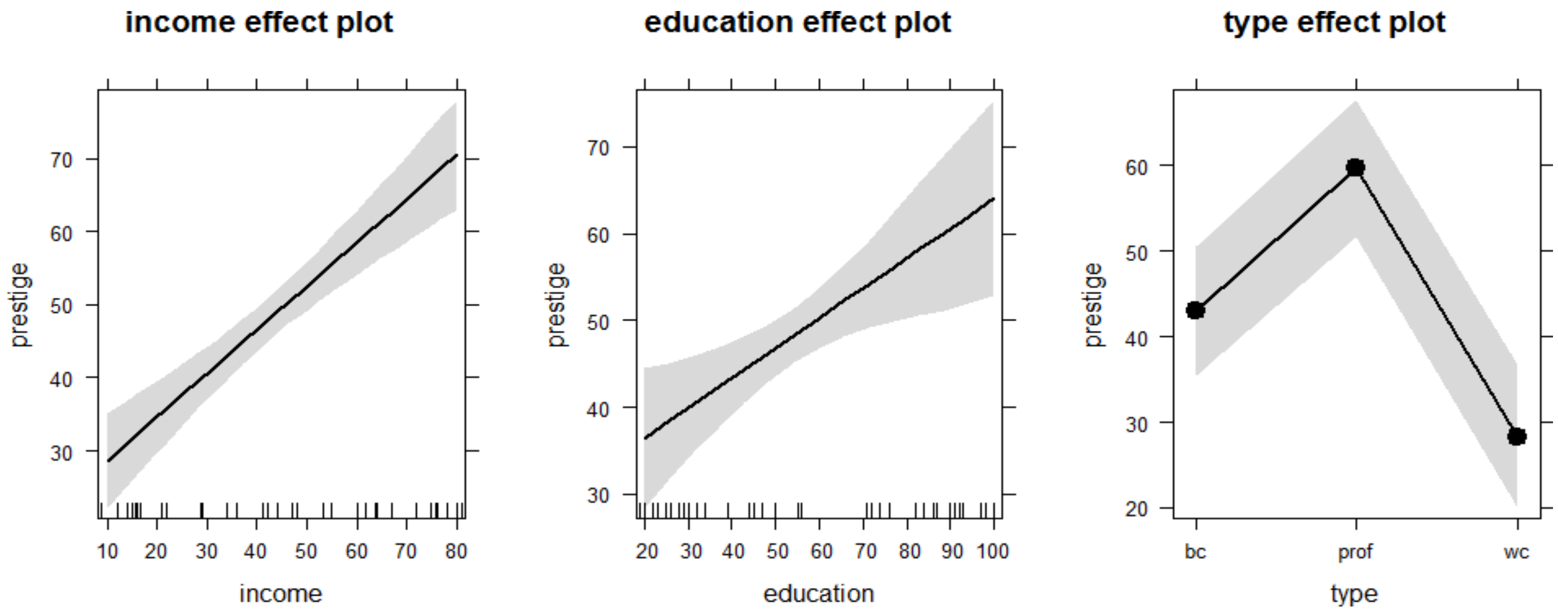
Each plot shows: partial slope, β_j influential obs.

Models: Effect plots

Fitted models are more easily interpreted by plotting the predicted values.

Effect plots do this nicely, making plots for each high-order term, controlling for others

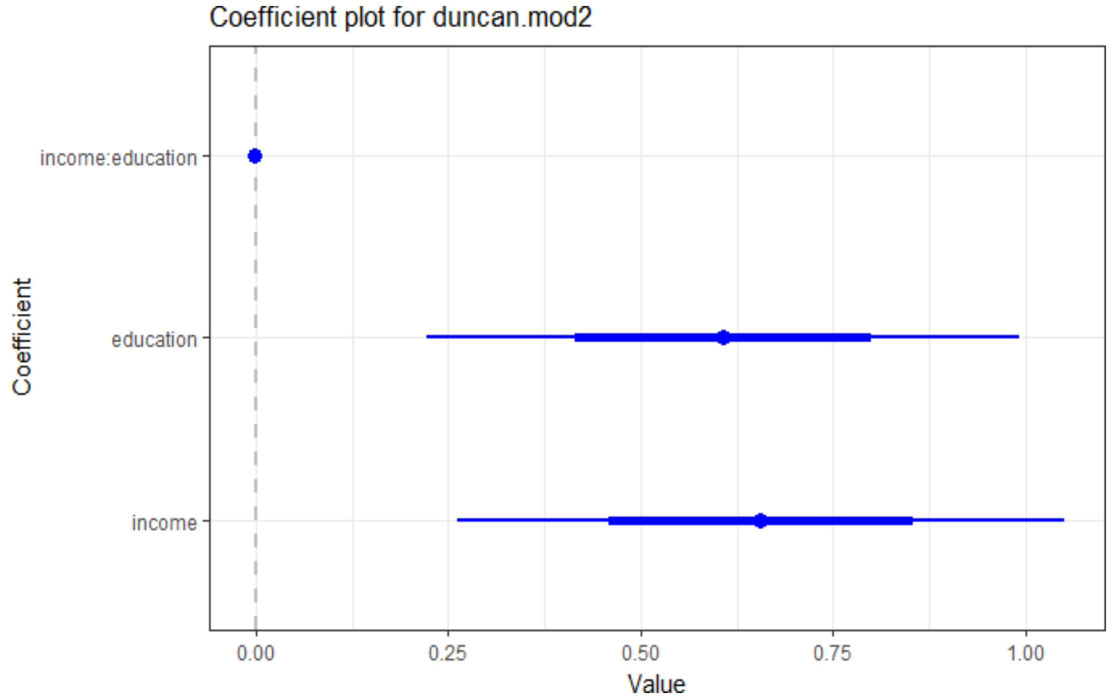
```
library(effects)
duncan.eff1 <- allEffects(duncan.mod1)
plot(duncan.eff1)
```



Models: Coefficient plots

Sometimes you need to report or display the coefficients from a fitted model. A plot of coefficients with CIs is sometimes more effective than a table.

```
library(coefplot)
duncan.mod2 <- lm(prestige ~ income * education, data=Duncan)
coefplot(duncan.mod2, intercept=FALSE, lwdInner=2, lwdOuter=1,
     title="Coefficient plot for duncan.mod2")
```



3D graphics

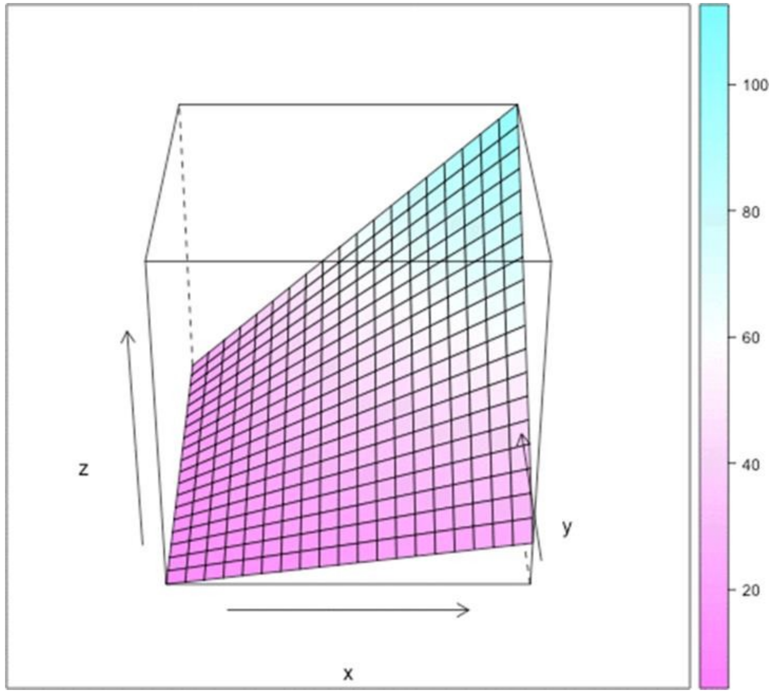
R has a wide variety of features and packages that support 3D graphics

This example illustrates the concept of an interaction between predictors in a linear regression model

It uses:
lattice::wireframe(z ~ x + y, ...)

The basic plot is “printed” 36 times rotated 10^o about the z axis to produce 36 PNG images.

The ImageMagick utility is used to convert these to an animated GIF graphic



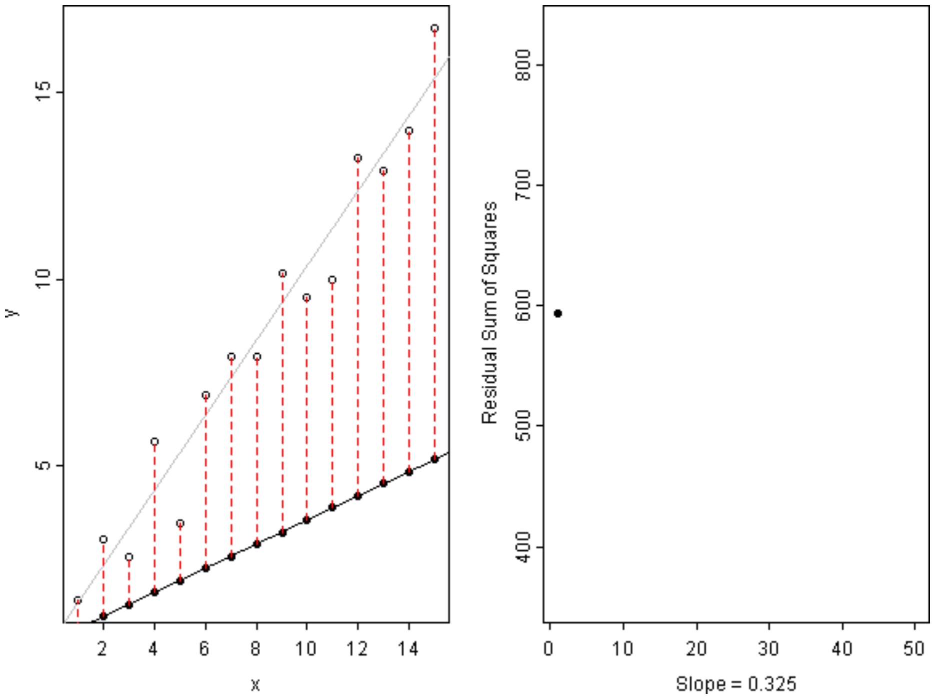
Statistical animations

Statistical concepts can often be illustrated in a dynamic plot of some process.

This example illustrates the idea of least squares fitting of a regression line.

As the slope of the line is varied, the right panel shows the residual sum of squares.

This plot was done using the animate package

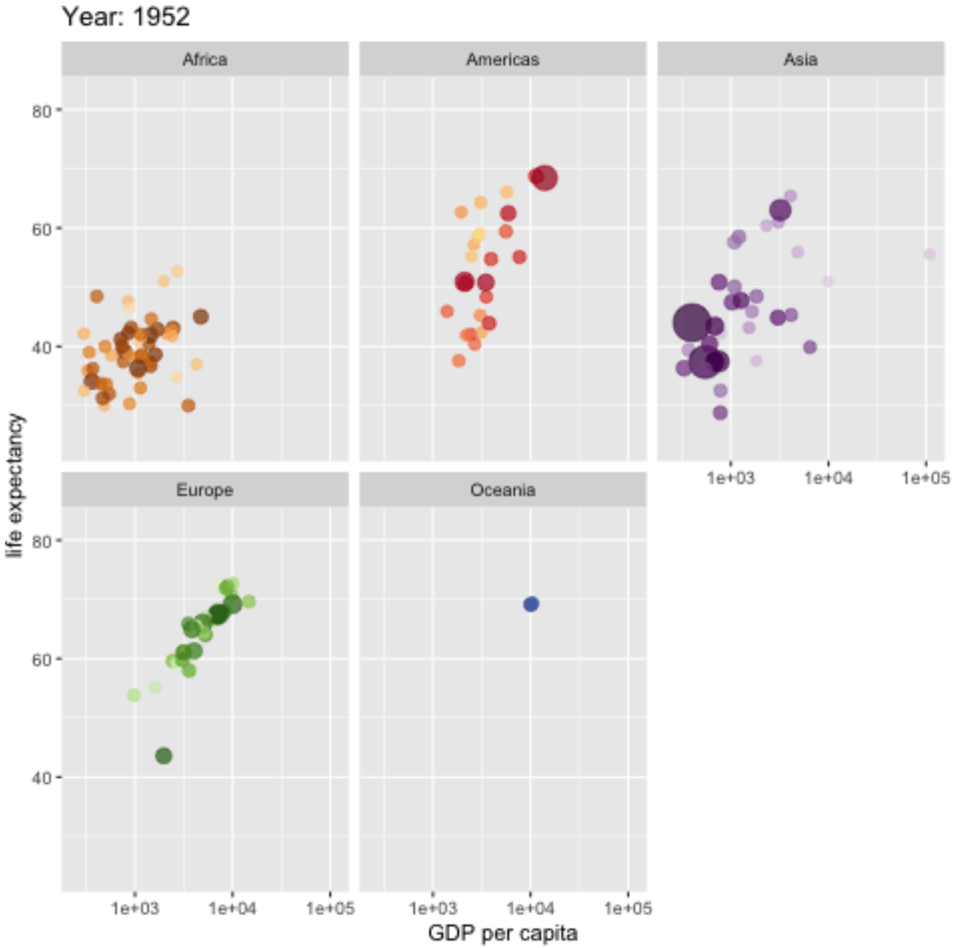


Data animations

Time-series data are often plotted against time on an X axis.

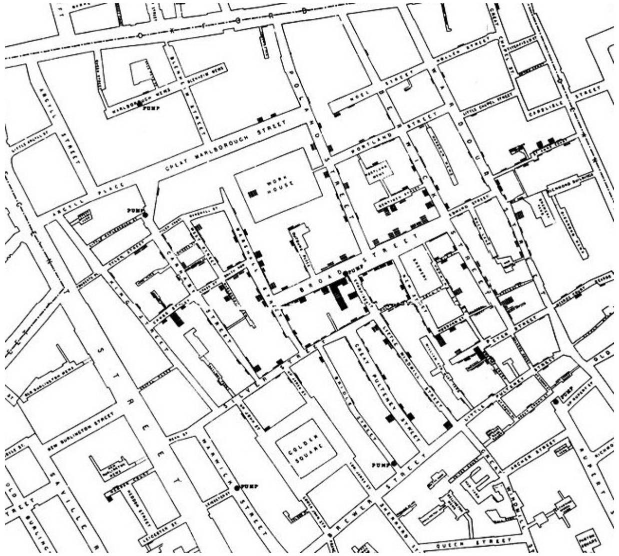
Complex relations over time can often be made simpler by animating change – liberating the X axis to show something else

This example from the tweenr package (using gganimate)



Maps and spatial visualizations

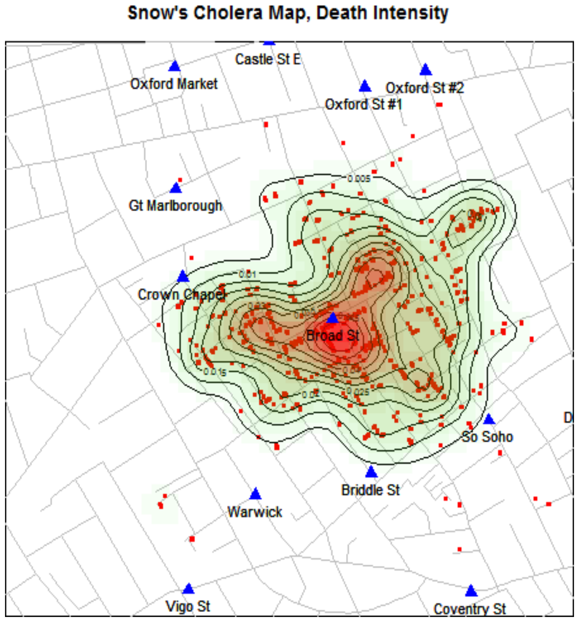
Spatial visualization in R, combines map data sets, statistical models for spatial data, and a growing number of R packages for map-based display



Dr. John Snow's map of cholera in London, 1854

Enhanced in R in the HistData package to make Snow's point

Portion of Snow's map:



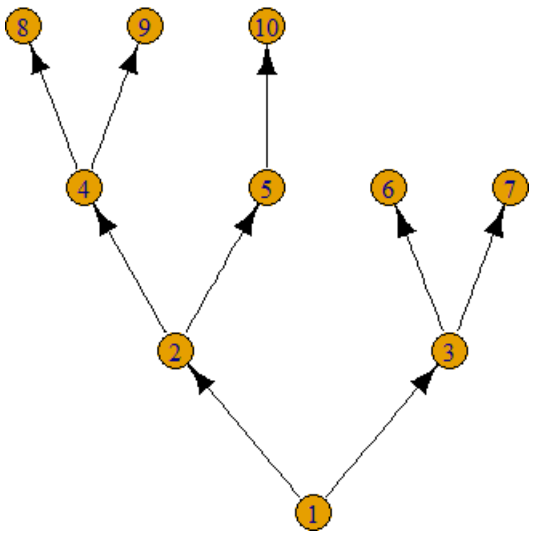
```
library(HistData)
SnowMap(density=T
RUE,
   main="Snow's Cholera Map, Death
   Intensity")
```

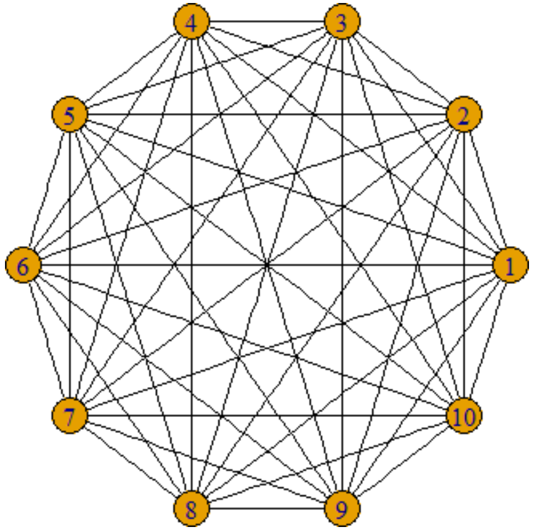
Diagrams: Trees & Graphs

A number of R packages are specialized to draw particular types of diagrams. igraph is designed for network diagrams of nodes and edges

```
library(igraph)
tree <- graph.tree(10)
tree <- set.edge.attribute(tree, "color",
value="black")  plot(treeIgraph,
   layout=layout.reingold.tilfor
   d(tree, root=1,
   flip.y=FALSE))
```

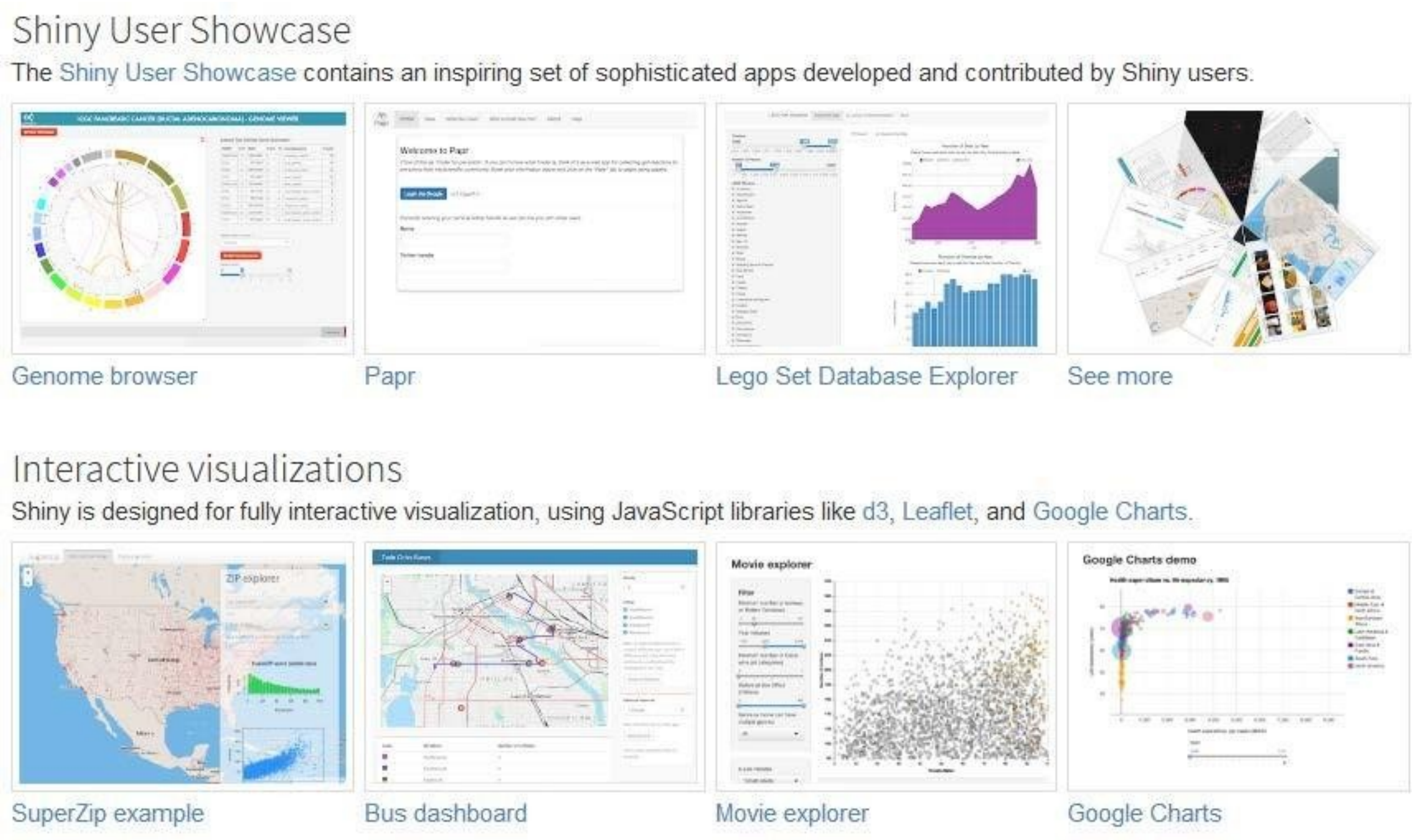
```
full <- graph.full(10)
fullIgraph <- set.edge.attribute(full,
"color",  value="black")
plot(full, layout=layout.circle)
```





shiny: Interactive R applications

shiny, from R Studio, makes it easier to develop interactive applications



Shiny User Showcase

The Shiny User Showcase contains an inspiring set of sophisticated apps developed and contributed by Shiny users.

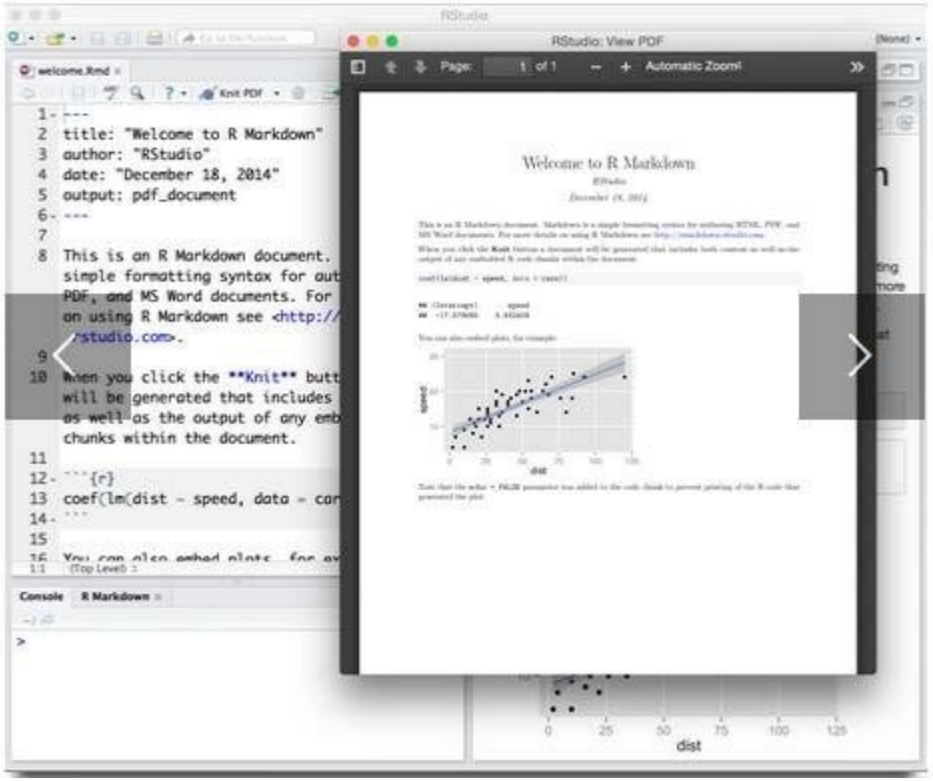
Genome browser Papr Lego Set Database Explorer See more

Interactive visualizations

Shiny is designed for fully interactive visualization, using JavaScript libraries like d3, Leaflet, and Google Charts.

SuperZip example Bus dashboard Movie explorer Google Charts

Reproducible analysis & reporting

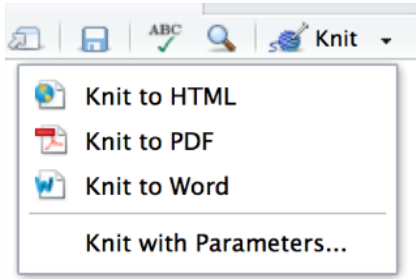


~~~
---
title: "Welcome to R Markdown"
author: "RStudio"
date: "December 18, 2014"
output: pdf_document
---

This is an R Markdown document.
simple formatting syntax for aut
PDF, and MS Word documents. For
on using R Markdown see <http://
rstudio.com>.

When you click the **Knit** butt
will be generated that includes
as well as the output of any emb
chunks within the document.

```{r}
coef(lm(dist ~ speed, data = car
```

You can also embed plots, for ex
~~~

R Studio, together with the knitr and rmarkdown packages provide an easy way to combine writing, analysis, and R output into complete documents

.Rmd files are just text files, using rmarkdown markup and knitr to run R on “code chunks”

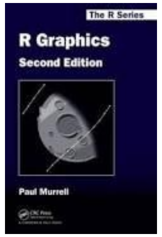
A given document can be rendered in different output formats:



Exercise

- Explore on how to use R productively in analysis & reporting
- Find one or more examples of data graphs from your research area
  - What are the graphic elements: points, lines, areas, regions, text, labels, ???
  - How could they be “described” to software such as R?
  - How could they be improved?

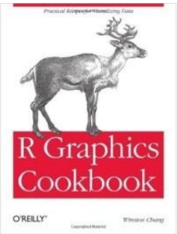
References



Paul Murrell, *R Graphics*, 2nd Ed.

Covers everything: traditional (base) graphics, lattice, ggplot2, grid graphics, maps, networkdiagrams, …

R code for all figures: https://www.stat.auckland.ac.nz/~paul/RG2e/



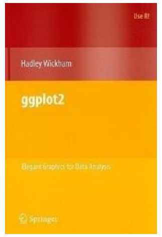
Winston Chang, *R Graphics Cookbook: Practical Recipes for Visualizing Data*

Cookbook format, covering common graphing tasks; the main focus is on ggplot2

R code from book: http://www.cookbook-r.com/Graphs/

Deepayn Sarkar, *Lattice: Multivariate Visualization with R*

R code for all figures: http://lmdvr.r-forge.r-project.org/



Hadley Wickham, *ggplot2: Elegant graphics for data analysis*, 2nd Ed.

ggplot2 Quick Reference: http://sape.inf.usi.ch/quick-reference/ggplot2/

PES UNIVERSITY ONLINE

THANK YOU

Dr.Mamatha H R

Professor,Department of Computer Science

mamathahr@pes.edu

+91 80 2672 1983 Extn 834