R and Statistics (I)

Ashish Mahabal
AyBi199, Caltech, 19 May 2011
Statistics is extensively used

- 15,000 astronomical studies per year
- 5% have “statistics” in their abstract
- 20% treat variable objects or multivariate datasets
However, it is not well understood

5 out of 4 people have trouble with Statistics.
(People also assume) it is misused

There are lies, damned lies and statistics
-Benjamin Disraeli
Limited number of methods still dominate

Traditional methods: preWWII

- Fourier transform (Fourier 1807)
- Least sq. and chisq (Legendre 1805, Pearson 1901)
- Kolmogorov-Smirnov test (Kolomogrov 1933)
- Principal Component Analysis (Hotelling 1936)

Xkcd/26
Advanced methods available in most systems

- Matlab
- Mathematica
- IDL
- Octave
- NumPy
- PDL
Examples of available functions

- Descriptive statistics (e.g. boxplot)
- Two- and k-sample tests (e.g. Wilcoxon rank-sum test)
- Density estimation (e.g. Kernel smoothing)
- Correlation and regression (e.g. PCA)
- Censored data (e.g. Survival)
- Multivariate classification (e.g. H clustering)
- External functions (e.g. K-density)
4th paradigm, D2K, ...

S and R

- **S**: John Chambers (Bell Labs)
- **S-plus**: 1988: Douglas Martin (UWash)
- **R**: 1993: Ross Ihaka, Robert Gentleman
  - Current version 2.13 (13 April 2011)
  - Lexical scoping (ala Scheme)
  - Procedural/functions
  - Object Oriented
  - Command line
R follows S

- Linear and nonlinear modeling
- Statistical tests
- Time series analysis
- Classification
- Clustering
- ...

http://www.r-project.org/
(25 standard/recommended packages)
Comprehensive R Archive Network

- http://cran.r-project.org/
- http://www.bioconductor.org/
- Over 4300 (3/11) user contributed packages
- Strength: people contributed
- Weakness: organic growth – uniformity lost (e.g. plots)

**AMORE**
A MORE flexible neural network package

**ARES**
Allelic richness estimation, with extrapolation beyond the sample size

**AcceptanceSampling**
Creation and evaluation of Acceptance Sampling Plans

**AdMit**
Adaptive Mixture of Student-t distributions

**AdaptFit**
Adaptive Semiparametic Regression

**AlgDesign**
AlgDesign

**Amelia**
Amelia II: A Program for Missing Data

**AnalyzeFMRI**
Functions for analysis of fMRI datasets stored in the ANALYZE or NIFTI format

**Animal**
Analyze time-coded animal behavior data
More extensively used

- 43% data-miners use R (Rexer's Annual Data Miner Survey in 2010; Boston; 735 in 60 countries)
- [http://rgl.neoscientists.org/about.shtml](http://rgl.neoscientists.org/about.shtml) (3D visualization with interface to R)
- RapidMiner [http://rapid-i.com/content/view/181/190/](http://rapid-i.com/content/view/181/190/)
Not so a few years ago ...

- Columns are autoselected (and can be deselected)
- Parameter choices for functions are conveniently placed
- Can be used from your own webpages on tables residing elsewhere
- Java/perl
- ASCII/fits

Multivariate classification

- Kmeans partitioning (m)
- H clustering (m)

Apply cuts?  YES  NO

Clusters: 2
Metric: euclidean
Max. iterations: 10
Method: average
Height to cut at: 0
Clusters: 2
Toy Demos

- Rediscovering HR diagram
- Rediscovering FP of Globular Clusters
- Looking for outliers in color-color space
A Generic Machine-Assisted Discovery Problem: Data Mapping and a Search for Outliers
An Example: Discoveries of High-Redshift Quasars and Type-2 Quasars
An Example: Discoveries of High-Redshift Quasars and Type-2 Quasars
Simple Clustering Analysis: Gaussian Mixture Modeling

Assumptions:
- There are \( k \) Gaussian components. The \( i \)’th component is called \( w_i \)
- Component \( w_i \) has an associated mean vector \( m_i \), and a covariance matrix \( S_i \)

The challenge:
- Find \( k \) and all \( m_i \) and \( S_i \)

The ugly reality: Things are seldom Gaussian
Clustering Analysis:
How many different kinds of things do we have here, and who belongs to what group, with what probability?

An example:
The bivariate luminosity and surface brightness distribution of galaxies.

(Also: subclustering, merged images, etc. etc.)

Driver & Cross (2000)
K-means

1. Ask user how many clusters they’d like. *(e.g. k=5)*
2. Randomly (?) guess k cluster Center locations
3. Associate each data point with its nearest center
4. Compute the new mean centers
5. Iterate until some convergence criterion is reached
K-means terminates.
Minimal spanning trees

- Choose a point A and connect it to its nearest neighbor, B
- Now choose a point which is closest to either A, or to B
- Continue till all points are covered
- Cut-off at a required length
Hierarchical clustering

Dendrogram (Average group linkage)
Exploring outliers

• Palomar-QUEST synoptic sky survey
• 9 mix-and-match colors from 8 filters
• Aim: finding outliers in color-color space for spectroscopic follow-up
• 1000 random objects
Boxplot

- Reveals relationships between colors (mean, median, overlap, outliers)
Clustering

• K-means provides various cluster centers along with withinss and a list of possible outliers
K-density

- Probability - density association for outliers
Visual confirmation
(found from 1000 random objects)
Running R

1. Create a separate sub-directory, say `work`, to handle a particular problem.
   
   ```
   $ mkdir work
   $ cd work
   ```

2. Start the R program with the command
   
   ```
   $ R
   ```

3. At this point R commands may be issued (see

4. To quit the R program the command is
   
   ```
   > q()
   ```

You can optionally save data.
Getting help

- help(solve)
- ?search
- help(“[[“)
- help.start()  # this is for html help
- ??matrix
- example(pairs)
• source is like `<
• sink is like `>
• .Rhistory, .Rdata
• ls() / objects(), rm() deal with objects
• ls()
• rm(list = ls())
Dealing with tables/objects (R frame)

- X = read.table("foo",header=TRUE)
- objects()
- objetos(X)
- names(X)
- X
- Name1
- X$Name1
- attach(X)
- Name1

<table>
<thead>
<tr>
<th>Num</th>
<th>Name1</th>
<th>Name2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1</td>
<td>3.3</td>
</tr>
<tr>
<td>2</td>
<td>4.4</td>
<td>5.4</td>
</tr>
</tbody>
</table>
Assignments

• \( x \leftarrow c(10.4, 5.6, 3.1, 6.4, 21.7) \)
• assign("x", c(10.4, 5.6, 3.1, 6.4, 21.7))
• \( c(10.4, 5.6, 3.1, 6.4, 21.7) \to x \) (!)
• \( 1/x \) # 0.09,0.17,0.32,0.15,0.04
• \( y \leftarrow c(x,0,x) \) # 10.4,.. ,21.7,0,10.4,..,21.7
• \( v \leftarrow \text{rep}(x) + y + 1 \) # x repeated 2.2 times, 1 11
• \( \text{var} = \text{sum}((x-\text{mean}(x))^2)/(\text{length}(x)-1) \)
Simple/standard functions

- $+$, $-$, $\times$, $/$, $^\wedge$
- log, exp, sin, cos, tan, sqrt
- range, min, max, length, sum, var, prod, sort
- sort.list, order, pmax, pmin
- sqrt(-17+0i) \# overloaded operators
Sequences

- 1:30  # 1,2,3,
- n <- 10; 1:n-1; 1:(n-1)  # : has precedence
- 2*1:15  #2,4,..,30
- `seq()` with named params: to, from, by, length
  - `s4 <- seq(length=51, from=-5, by=.2)`
  - Along can be used only by itself to create same sized vector as another 1:length(vector)
- `s5 <- rep(x, times=5)  # x1 x2 .. Xn x1 x2 ...`
- `s6 <- rep(x, each=5)  # x1 x1 x1 x1 x1 x2 x2 x2 ..`
Logical vectors

• n <- x >13  # conditional
  – TRUE, FALSE, NA
  – length(n) == length(x)
  – c1 & c2  # intersection
  – c1 | c2  # union
  – !c1  # negation
  – FALSE = 0 and TRUE = 1 when coerced
  – Missing values, NA, NaNs is.na(x)
Indexing

• `x[1:10]`  # get first 10
• `x[-(1:5)]`  # leave out first 5
• `x[is.na(x)] <- 0`  # replace missing values by 0
• `y[y < 0] <- -y[y < 0]`  OR  `y <- abs(y)`

• `labs <- paste(c("X","Y"), 1:10, sep="\"")`  
  `== c("X1", "Y2", "X3", "Y4", "X5", "Y6", "X7", "Y8", "X9", "Y10")`
Classes of objects and unclass() 

- Matrices 
- Factors (categorical data) 
- Data matrices 
- Functions 
- (also numeric, character, logical, raw) 
- Vectors are atomic (all elements of one mode)  
  - Coercion easy 
- Lists are non-atomic (are recursive too)
Arrays and matrices

• `dim(z) <- c(3,5,100)`  # 3-d array with size
• `c(a[2,1,1], a[2,2,1], a[2,3,1], a[2,4,1],
    a[2,1,2], a[2,2,2], a[2,3,2], a[2,4,2])`
• `a[,]`  # the entire array
• `x <- array(1:20, dim=c(4,5))`  # 4 by 5 array
• `i <- array(c(1:3,3:1), dim=c(3,2))`  # 3x2 array
• `x[i] <- 0`  # set elements 9, 6, 3 of x to 0
Outer products, functions

- `ab <- a %o% b`
- `ab <- outer(a, b, "*")`

Generalized
- `f <- function(x, y) cos(y)/(1 + x^2)`
- `z <- outer(x, y, f)`

- `A * B`  # element by element matrix product
- `A %*% B`  # matrix multiplication
- "%!%" <- function(X, y) { ... }
Lists (and data.frames=restricted)

- Lst <- list(name="Fred", wife="Mary", no.children=3, child.ages=c(4,7,9))
- Always numbered
  - Lst[[4]] # returns [1] 4 7 9
  - Lst[[4]][2] # returns 7
  - Lst[4][2] # returns Null
Reading from files

• HousePrice <- read.table("houses.data", header=TRUE)

• read.table(file, header = FALSE, sep = " ", quote = "\"",
dec = ".", row.names, col.names, as.is = !stringsAsFactors,
na.strings = "NA", colClasses = NA, nrows = -1, skip = 0, check.names =
TRUE, fill = !blank.lines.skip, strip.white = FALSE, blank.lines.skip = TRUE,
comment.char = "#", allowEscapes = FALSE, flush = FALSE,
stringsAsFactors = default.stringsAsFactors(), fileEncoding = "",
encoding = "unknown")

read.csv, read.csv2, read.delim, read.delim2
Accessing built-in datasets

- `data()`
- `data(AirPassengers)`
- `?AirPassengers`
- `new <- edit(AirPassengers)`
- `x <- array(c(AirPassengers[1:144]), dim=c(12,12))`
- `pairs(x)`

"I hope that wasn't our pilot."
Conditionals

- if (expr_1) expr_2 else expr_3
- for (name in expr_1) expr_2
- while (condition) expr
• Scope
• Arguments
• Customizing
• Factors
• Contrasts
Plotting

- plot(x, y)  # scatterplot
- plot(xy)    # scatterplot from 2-col matrix
- plot(x)     # timeseries or real/img
- plot(f)     # barplots for factors
- plot(f, y)  # boxplots for factors
- pairs(X)
- hist(), dotchart(), image(), contour(), ...
- points(), text(), math, multiple, interactive
colors and classification

- d2 = read.table("dataset2.Rframe", header=TRUE)
- objects()
- names(d2)
- umg
- d2$umg
- attach(d2)
- umg
- plot(umg, gmr)
- pairs(d2)
• demo(graphics) to see some graphics capabilities

• library(ggplot2)
• help(diamonds)
• dsmall <- diamonds[sample(nrow(diamonds), 100), ]
• qplot(carat, price, data=dsmall, shape=cut)
• dsmall
• qplot(carat, price, data=diamonds, alpha=l(1/500))
• qplot(carat, price, data=diamonds, geom=c("smooth","point"))
• qplot(carat, data=diamonds, geom="density", colour = color)