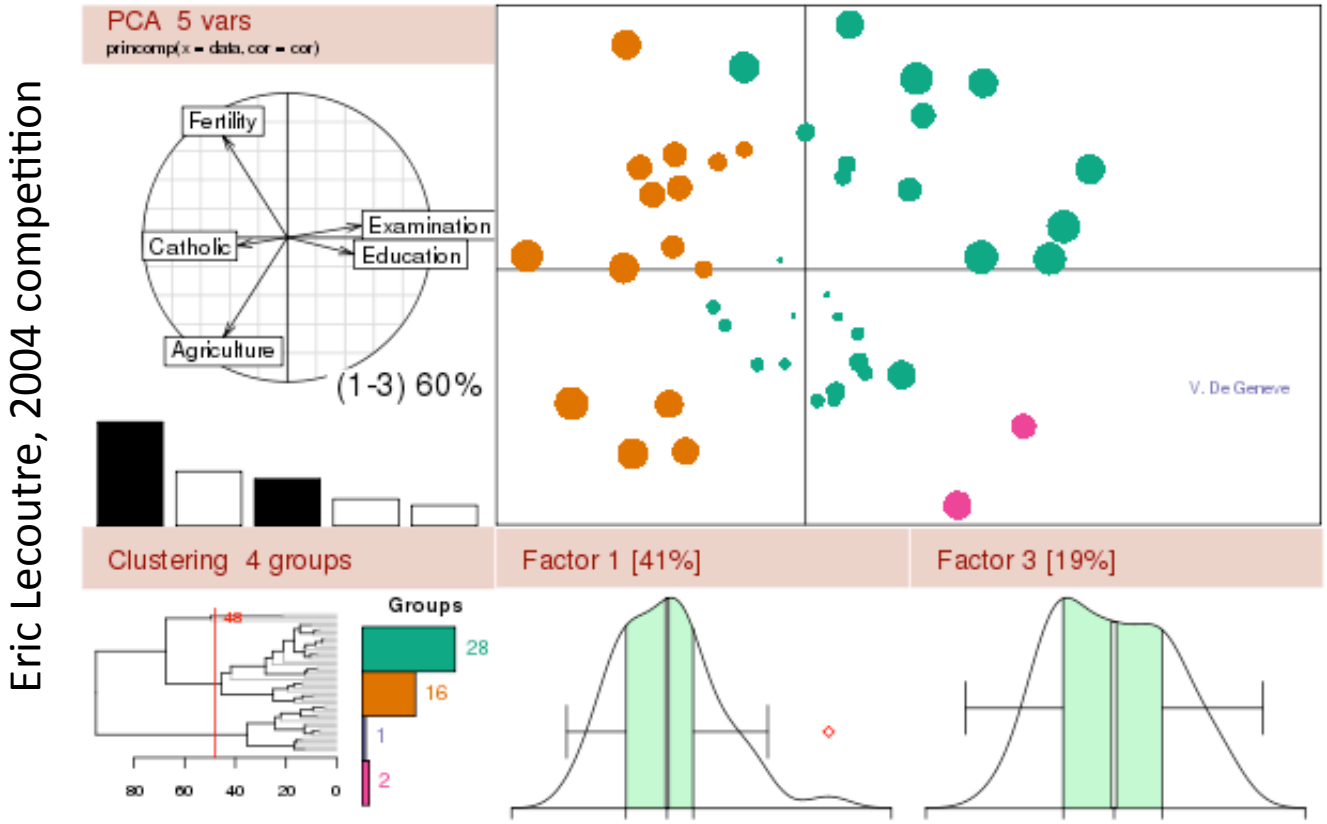
R and Statistics (I)



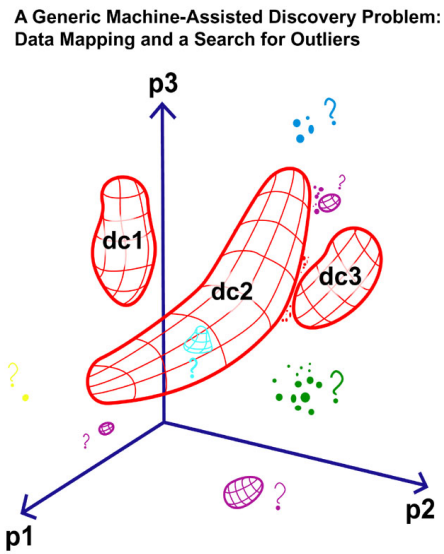
Eric Lecoutre, 2004 competition

Ashish Mahabal

AyBi199, Caltech, 19 May 2011

Statistics is extensively used

- 15000 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.

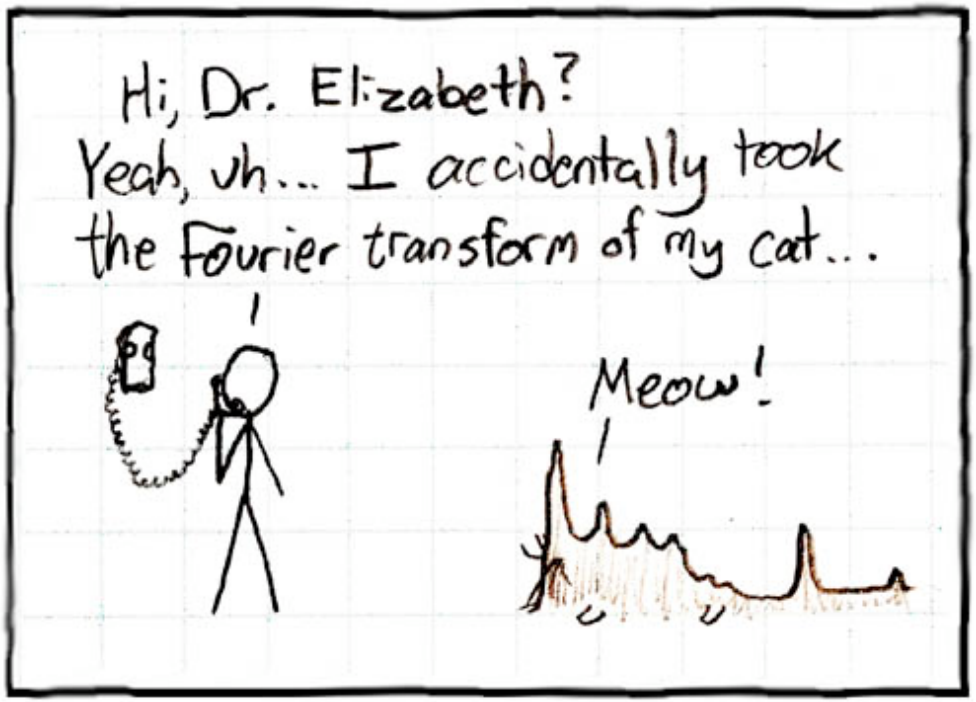
(People also assume) it is misused

There are lies, damned lies and statistics
-Benjamin Disraeli

Limited number of methods still dominate

Traditional methods: preWWII

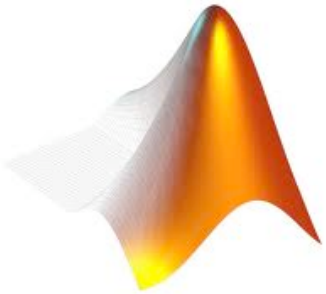
- Fourier transform (Fourier 1807)
- Least sq. and chisq (Legendre 1805, Pearson 1901)
- Kolmogorov-Smirnov test (Kolomogrov 1933)
- Principal Component Analysis (Hotelling 1936)



Xkcd/26

Advanced methods available in most systems

- Matlab
- Mathematica
- IDL
- Octave
- NumPy
- PDL

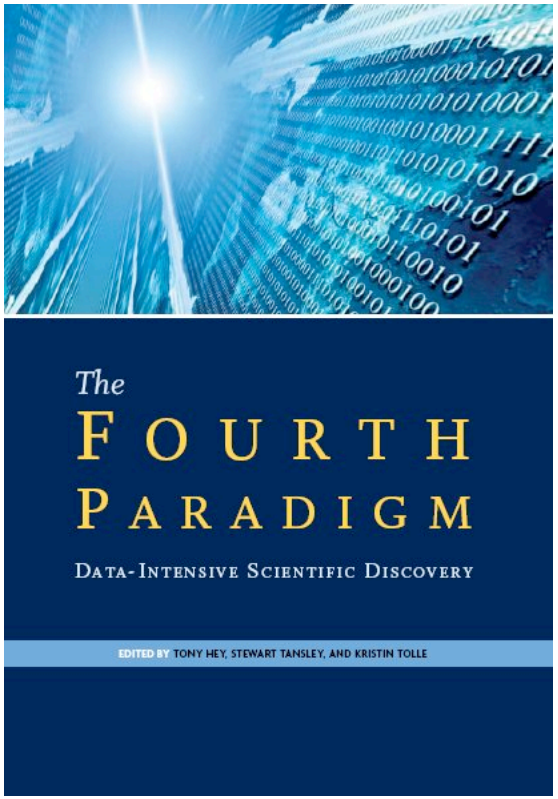


Examples of available functions

- 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, …

- http://research.microsoft.com/en-us/collaboration/fourthparadigm/



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

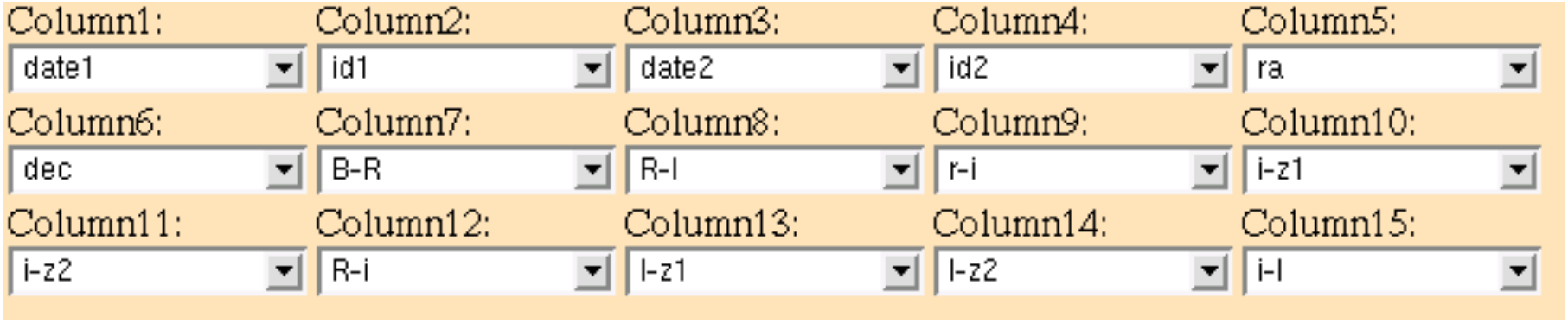
  (3D visualization with interface to R)
- RapidMiner http://rapid-i.com/content/view/181/190/)
- Weka (http://www.cs.waikato.ac.nz/ml/weka/)

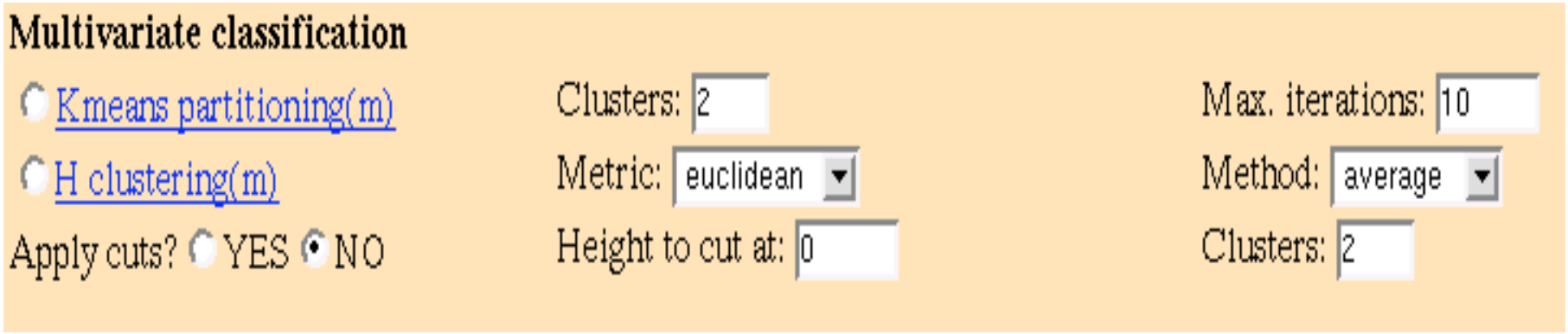
  R interface

Not so a few years ago …

VOStat

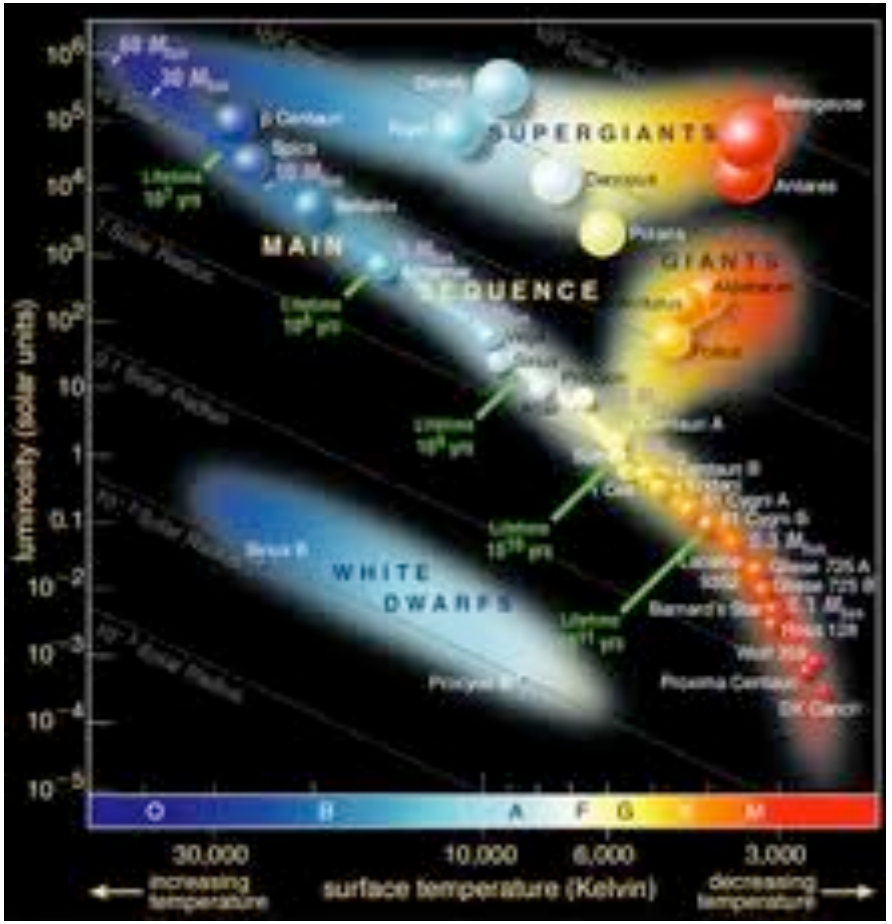
- Columns are autoselected (and can be deselected)
- Parameter choices for functions are conveniently placed
- Can be used from your own webpages on tables residing elsewhere
- Java/perl
- ASCII/fits



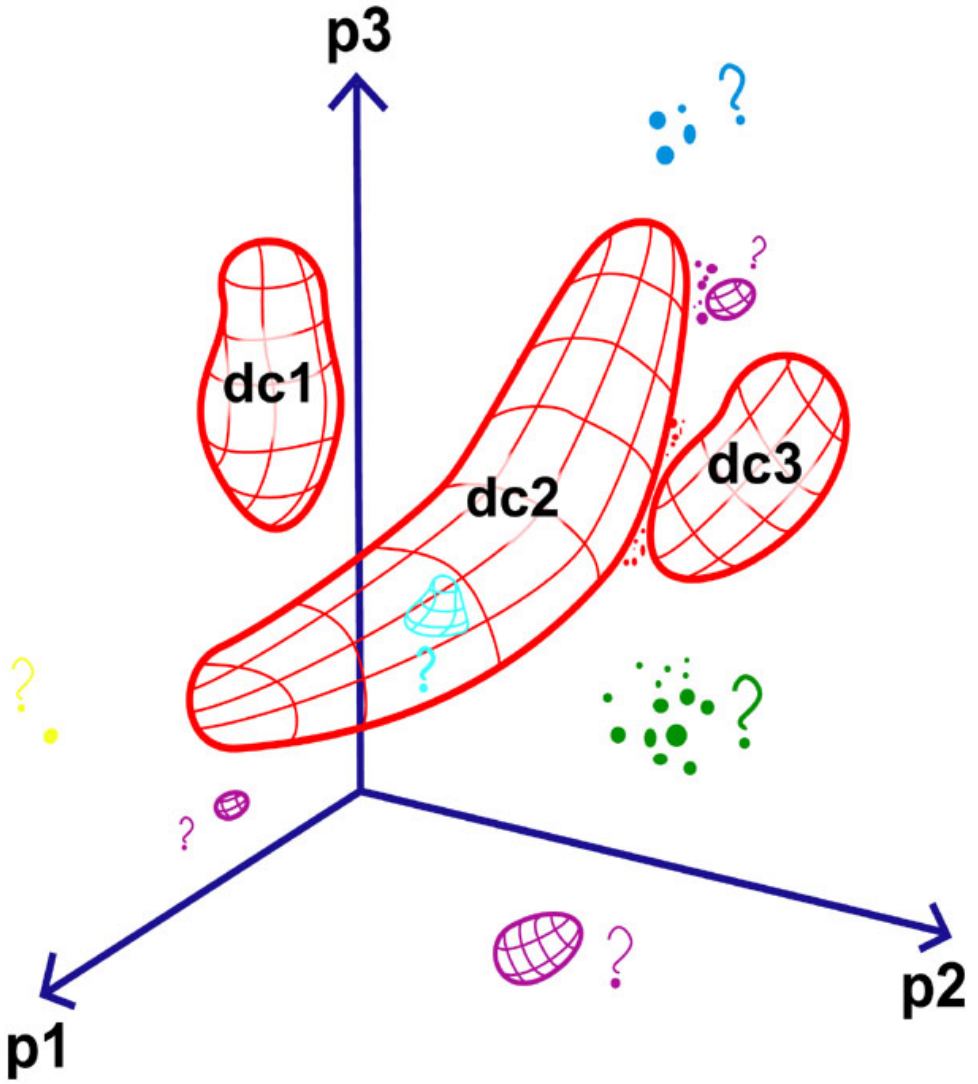


Toy Demos

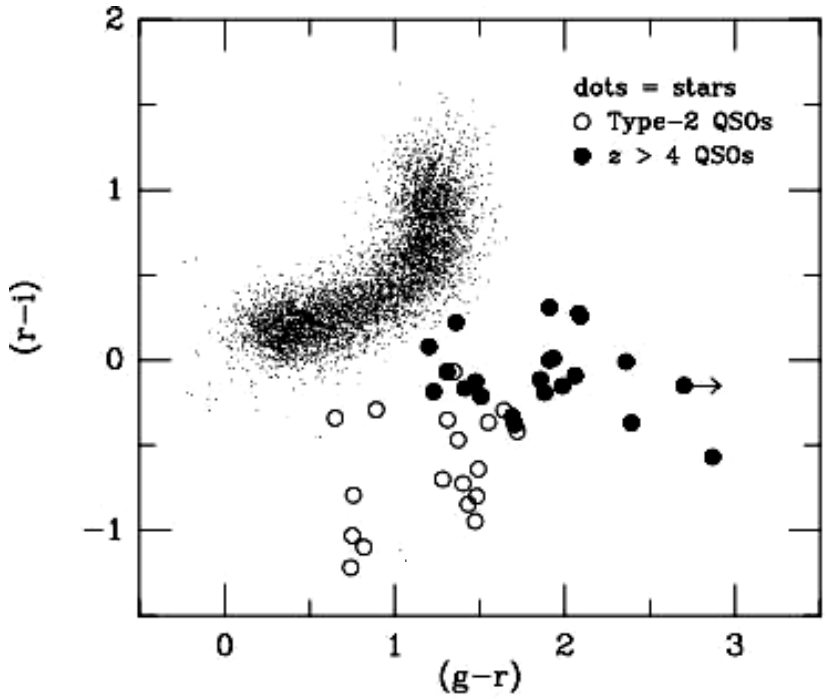
- Rediscovering HR diagram
- Rediscovering FP of Globular Clusters
- Looking for outliers in color-color space



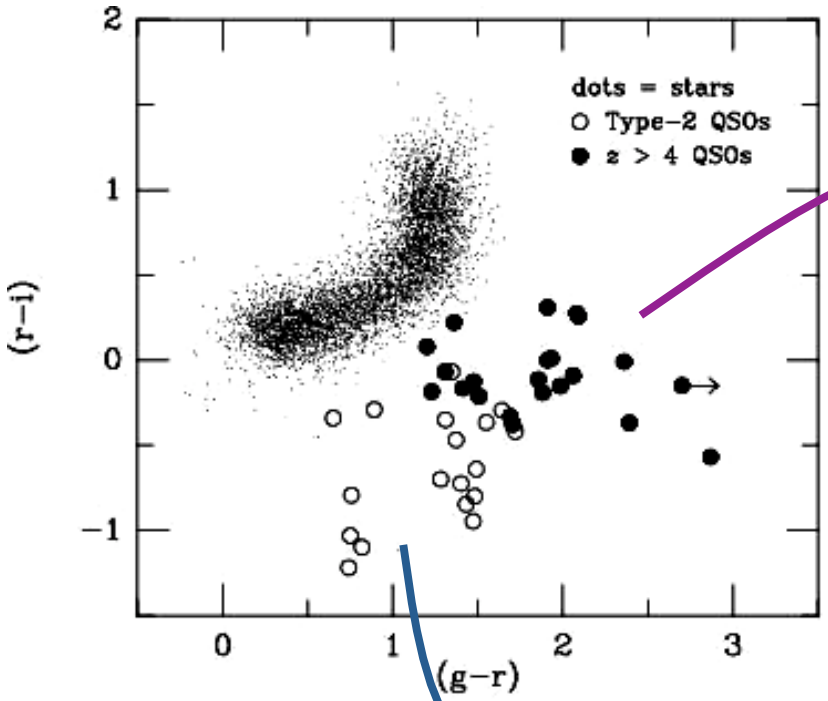
A Generic Machine-Assisted Discovery Problem:
Data Mapping and a Search for Outliers



An Example: Discoveries of High-Redshift Quasars and Type-2 Quasars

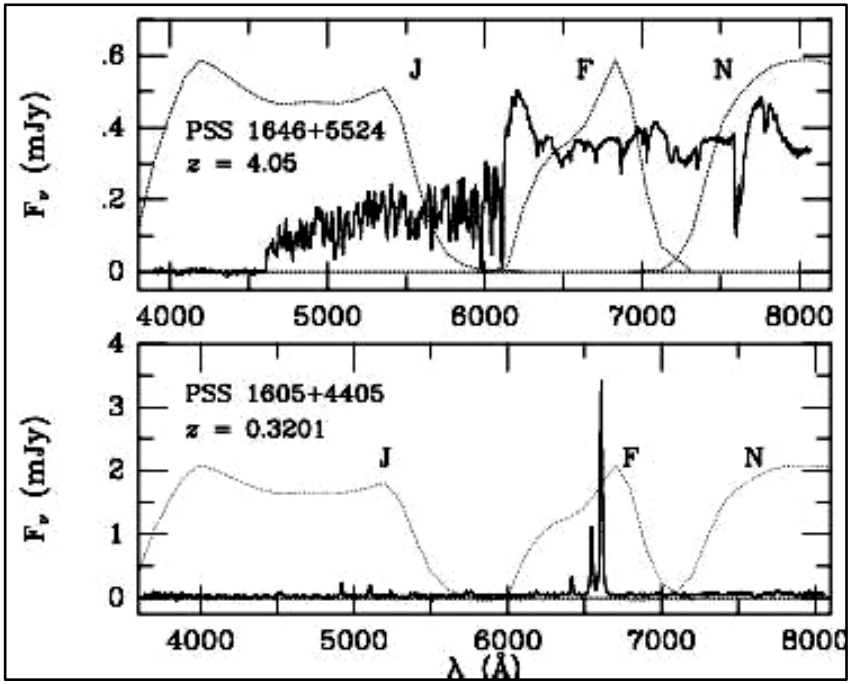


An Example: Discoveries of High-Redshift Quasars and Type-2 Quasars



High-z QSO

Type-2 QSO



Simple Clustering Analysis: Gaussian Mixture Modeling

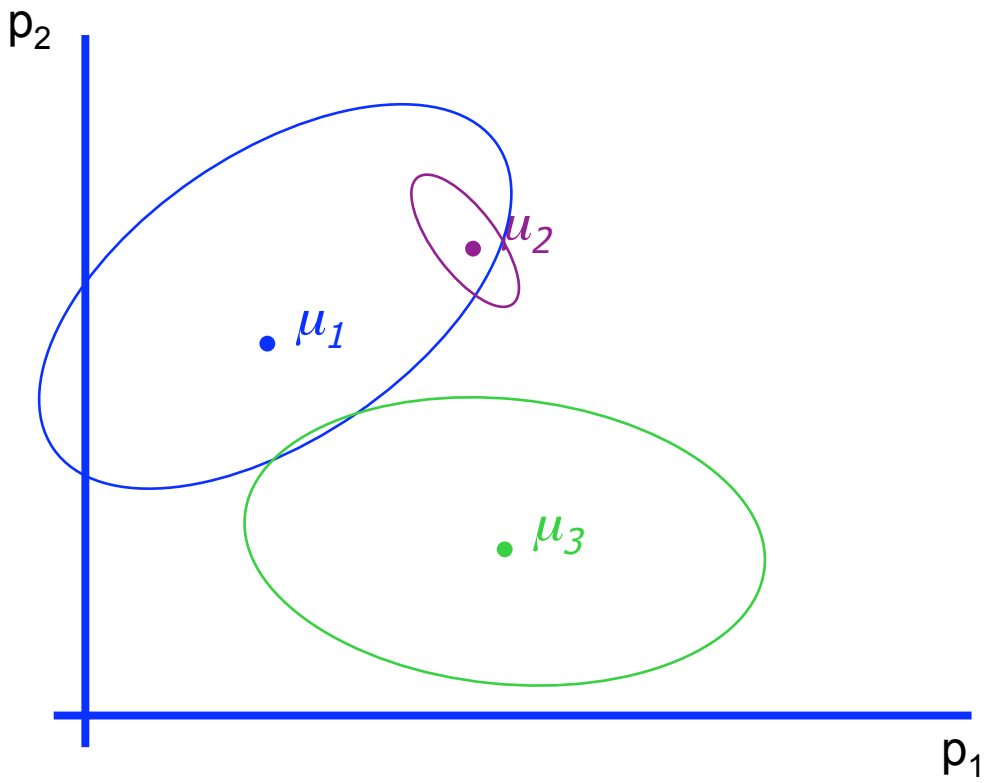
Assumptions:

- 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

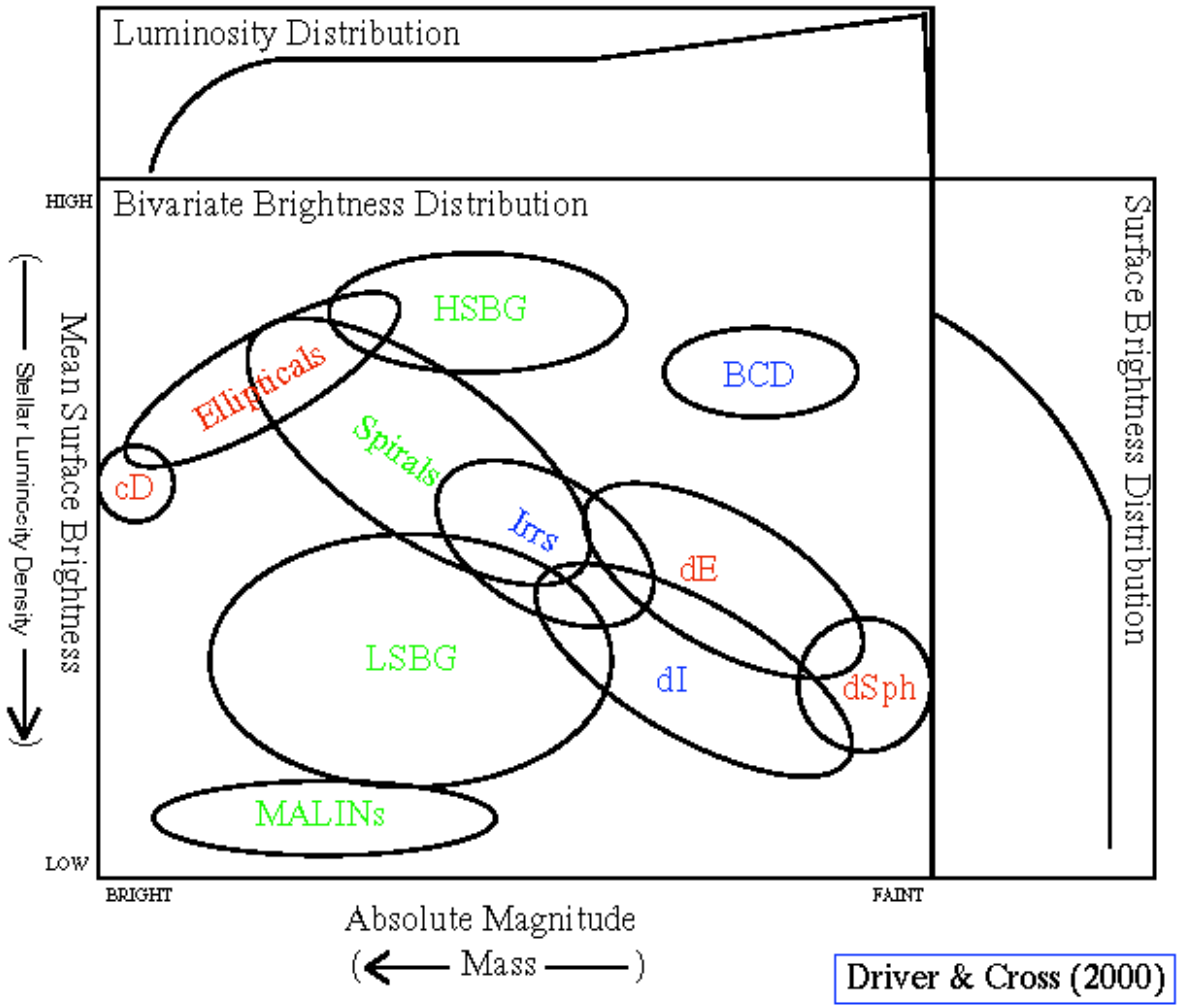


Clustering Analysis:

How many different kinds of things do we have here, and who belongs to what group, with what probability?

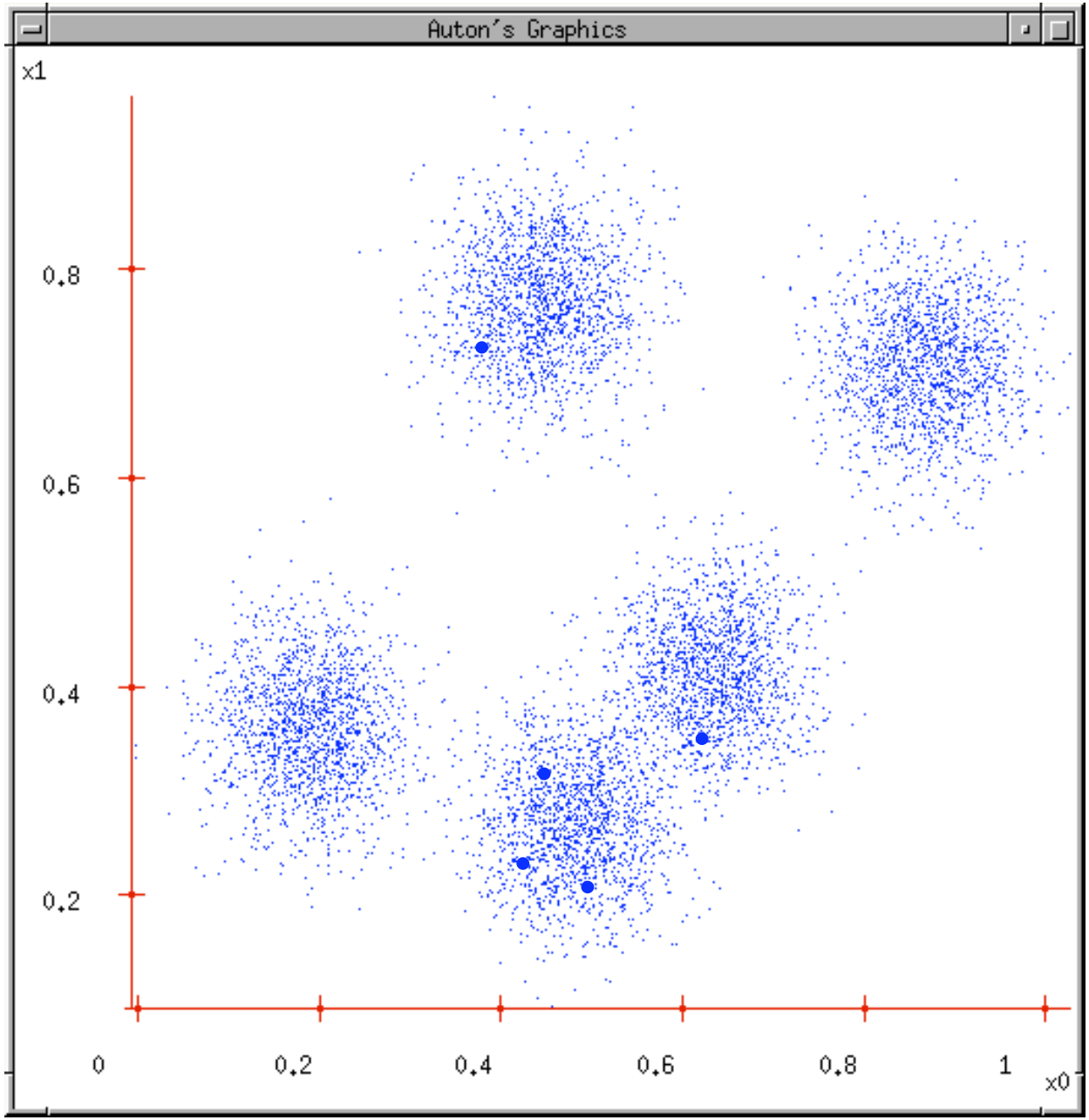
An example:
The bivariate luminosity and surface brightness distribution of galaxies.

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

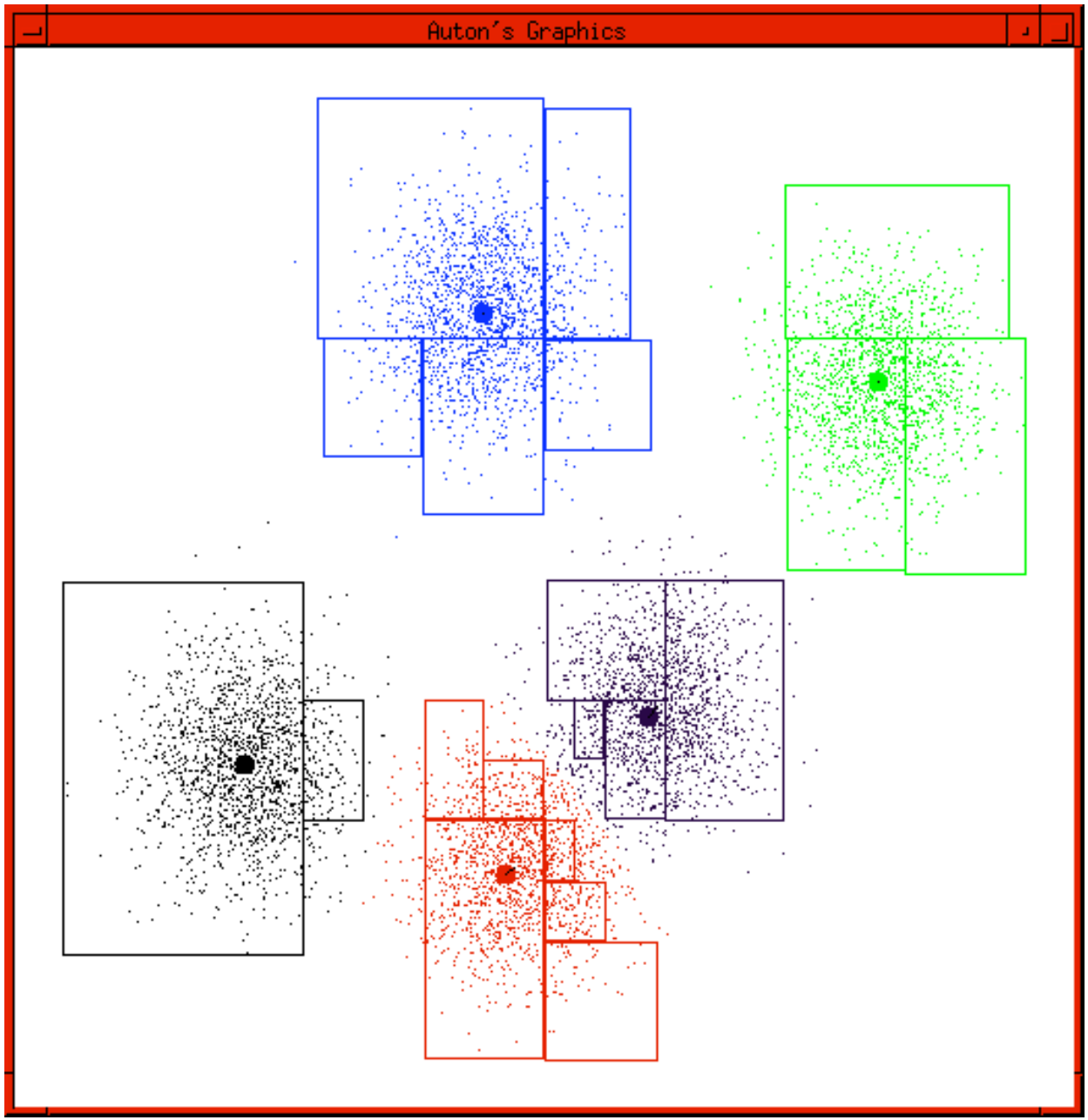


K-means

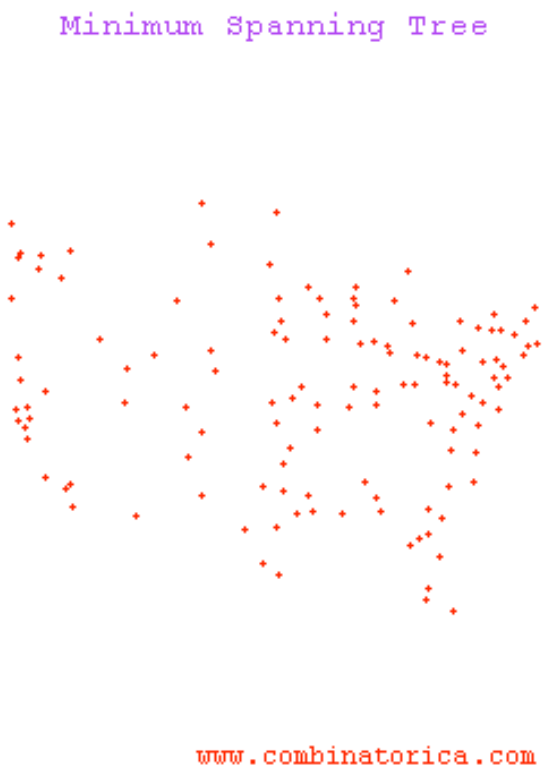
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly (?) guess k cluster Center locations
3. Associate each data point with its nearest center
4. Compute the new mean centers
5. Iterate until some convergence criterion is reached



K-means terminates

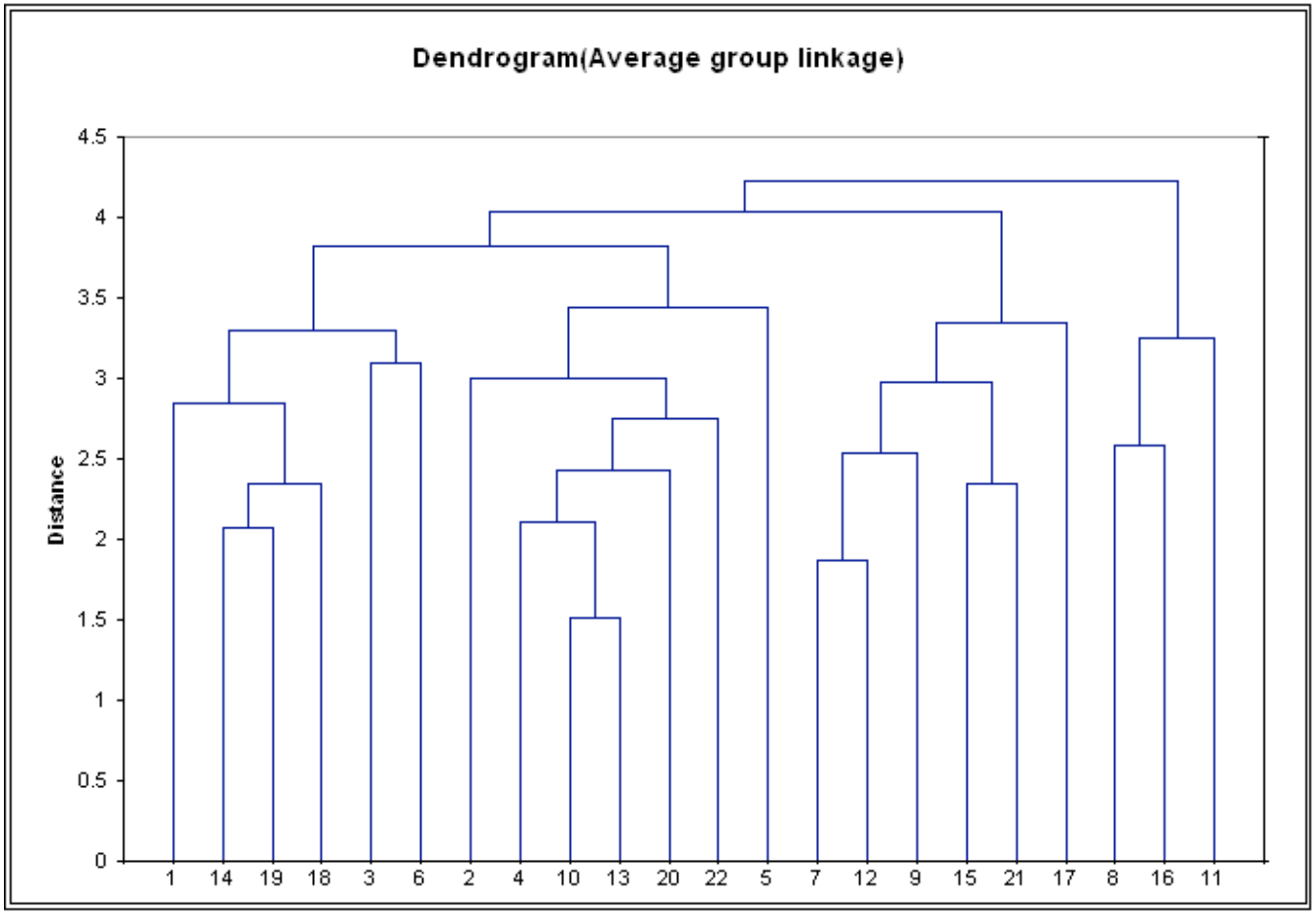


Minimal spanning trees



- 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



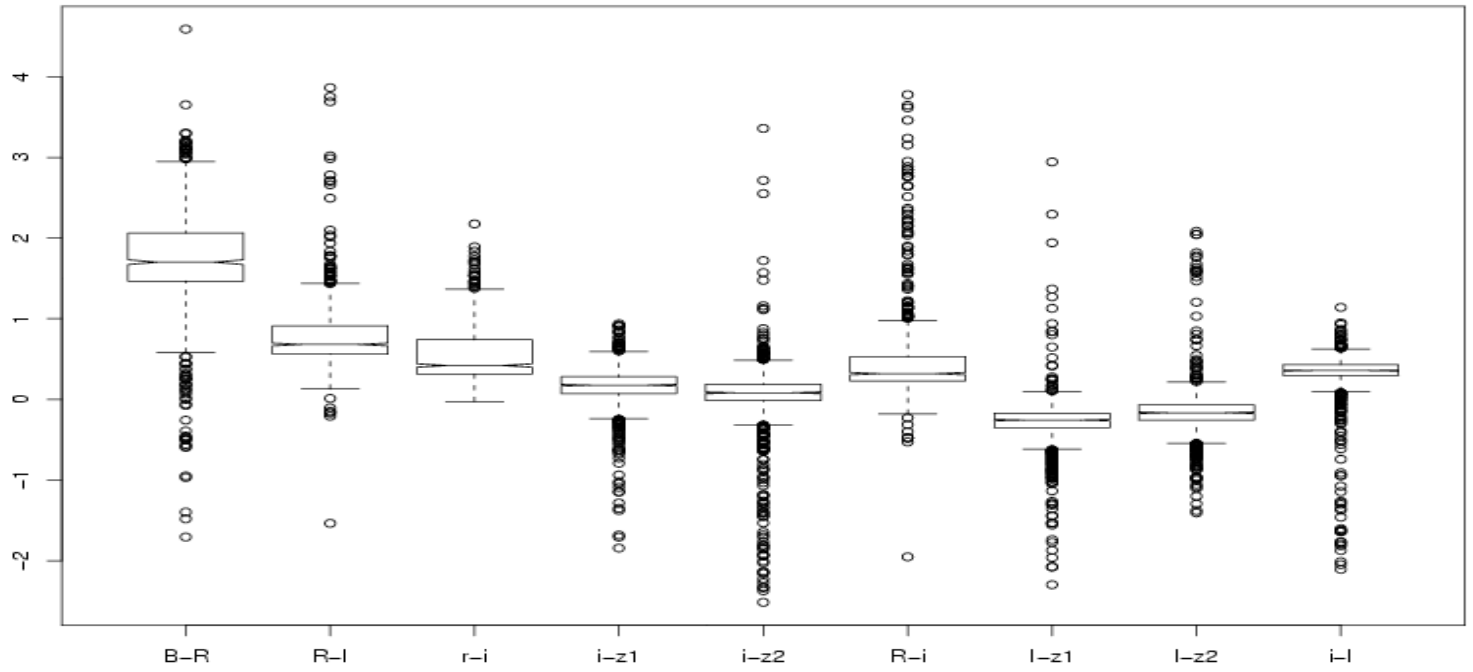
Exploring outliers

- 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

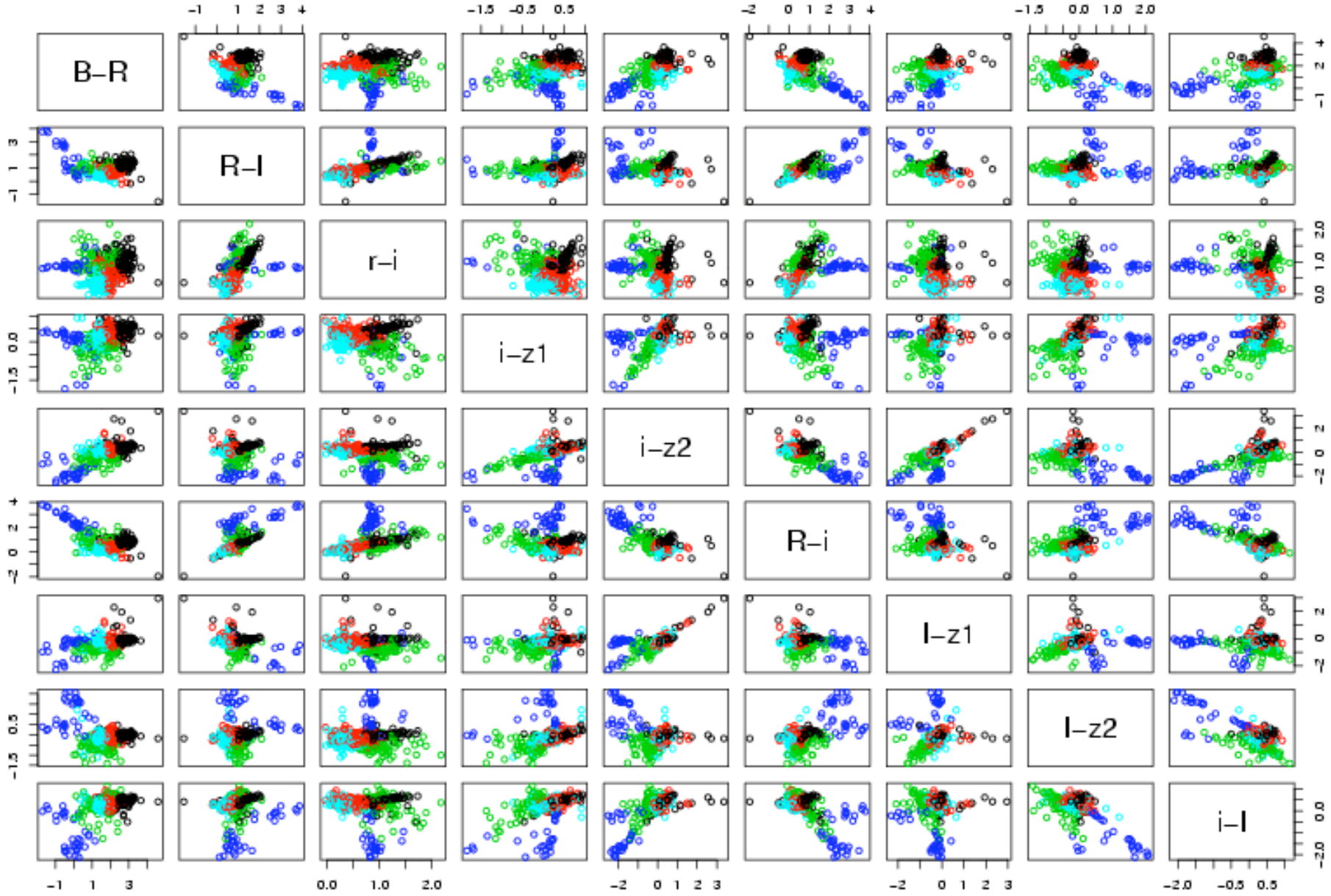
(mean, median, overlap, outliers)



Clustering

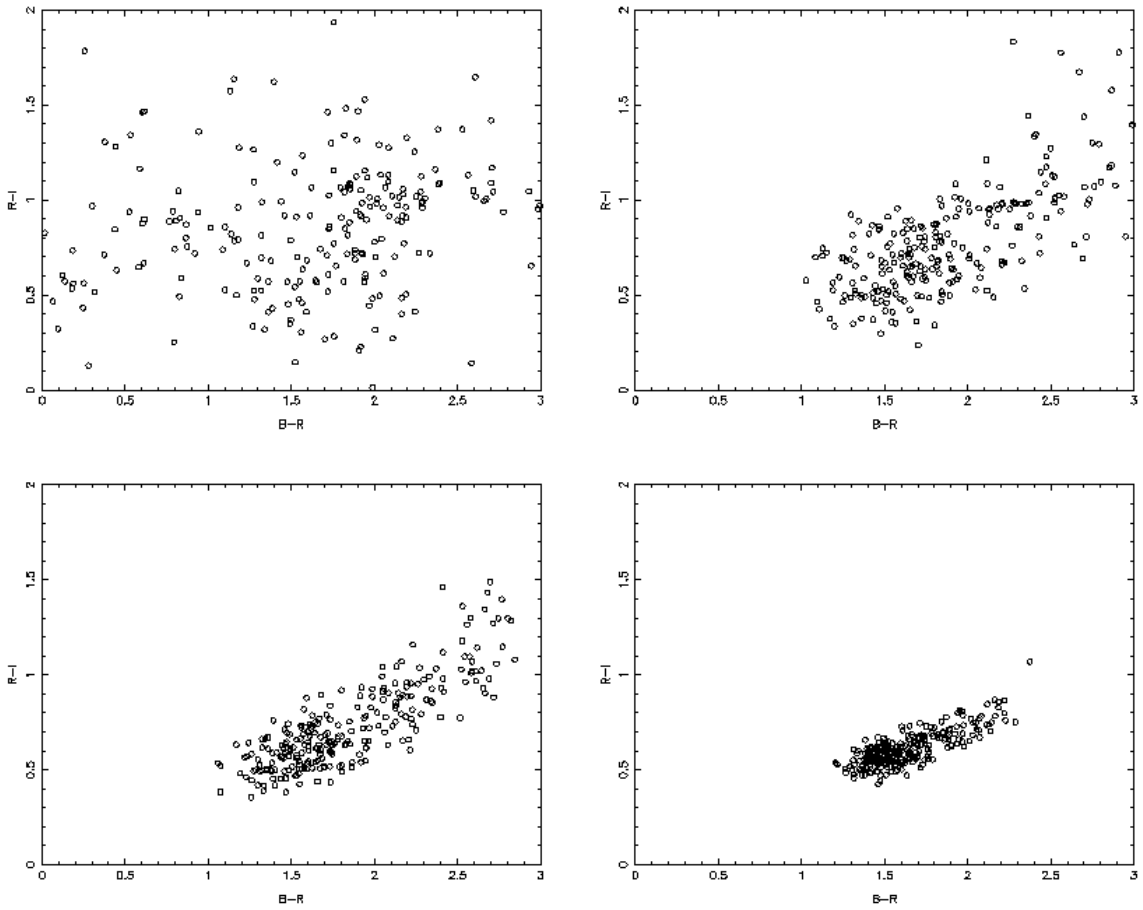
- K-means provides various cluster centers along with withinss and a list of possible outliers

```
> print(cl$size)
[1]  75 122 480 289  33
> print(cl$centers)
     X2.1553   X0.8833    X0.563     X0.2347     X0.1909   X0.5069    X.0.1855
1  1.4074840 0.9671000 1.0442640 -0.29106000 -0.51805867 0.7100160 -0.77514267
2  2.6638713 1.1741434 0.9580811  0.40006393  0.39761967 0.7380221 -0.03850164
3  1.5062098 0.5789365 0.3302465  0.12387542  0.05890729 0.2437696 -0.27625958
4  2.0228934 0.7910471 0.5455412  0.24857820  0.15998754 0.4003003 -0.23075917
5 -0.1626848 1.5104515 0.8905485 -0.08033333 -1.67197273 2.5554939 -0.62693030
     X.0.1417    X0.3764
1 -0.54814400  0.2570840
2 -0.03605738  0.4361213
3 -0.21129146  0.3351669
4 -0.14216851  0.3907467
5  0.96470909 -1.0450424
> print(cl$withinss)
[1] 118.08204 126.88979  95.40461  87.19068 142.91592
> print(cl$cluster)
  [1] 1 4 3 2 3 3 3 4 3 3 3 3 3 3 3 4 4 3 3 4 3 3 3 3 3 4 5 3 3 3 3 3 3 3 3 3 5
```


B–R
R–I
r–i
i–z1
i–z2
R–i
I–z1
I–z2
i–I

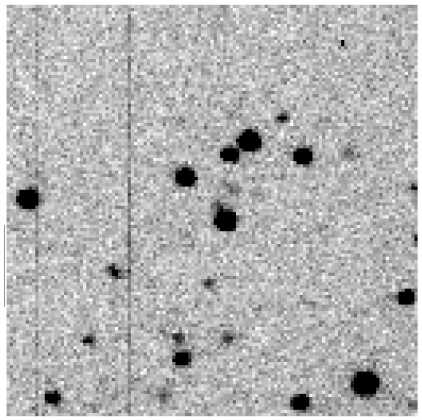
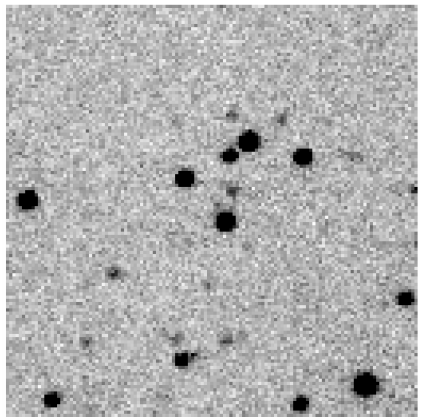
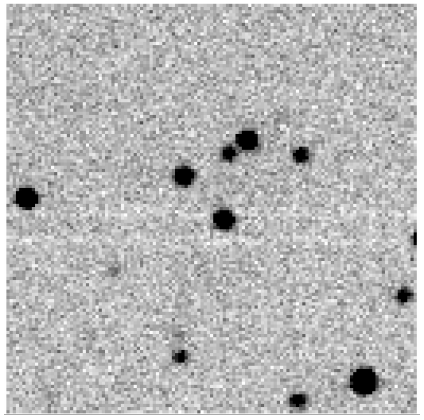
K-density

- Probability - density association for outliers



Visual confirmation

(found from 1000 random objects)

Running R

1. Create a separate sub-directory, say `work`, to h
   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

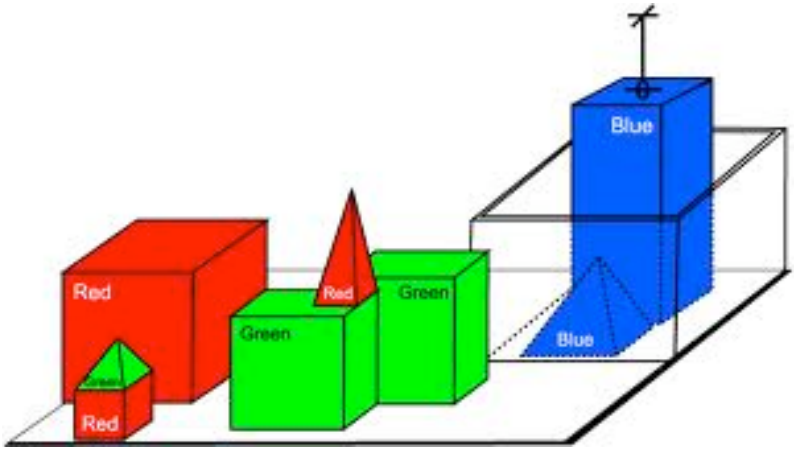
- help(solve)
- ?search
- help("[[")
- help.start() # this is for html help
- ??matrix
- example(pairs)

- 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()
- objets(X)
- names(X)
- X
- Name1
- X$Name1
- attach(X)
- Name1

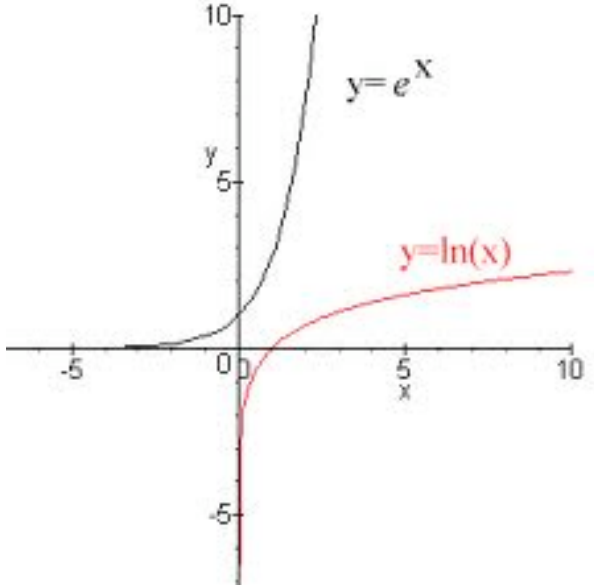
Num	Name1	Name2
1	1.1	3.3
2	4.4	5.4



Assignments

- x <- 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) -> x (!)
- 1/x # 0.09,0.17,0.32,0.15,0.04
- y <- c(x,0,x) # 10.4,.. ,21.7,0,10.4,..,21.7
- v <- rep(x) + y + 1 # x repeated 2.2 times, 1 11
- var = sum((x-mean(x))^2)/(length(x)-1)

Simple/standard functions



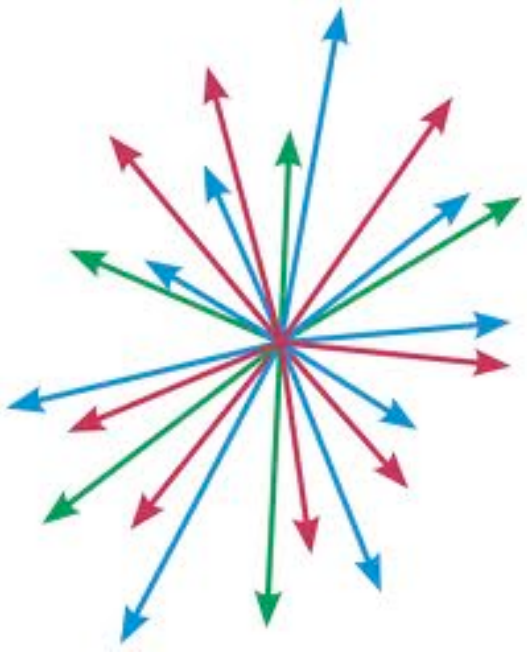
- + - * / ^
- log, exp, sin, cos, tan, sqrt
- range, min, max, length, sum, var, prod, sort
- sort.list, order, pmax, pmin
- sqrt(-17+0i) # overloaded opeartors

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 ..

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.nan(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")

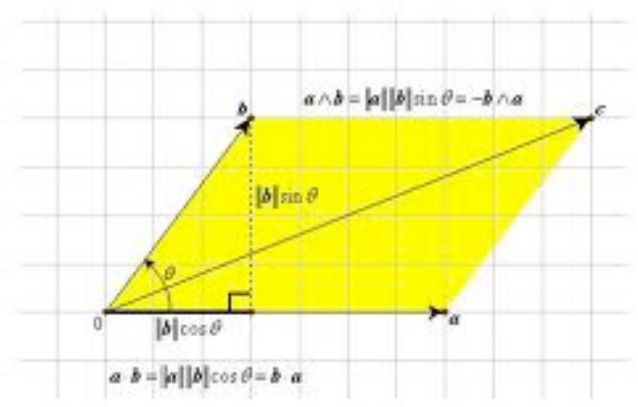
Classes of objects and unclass()

- 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
- Lst[3] # returns $no.children [1] 3

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)

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

Demo

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=I(1/500))
- qplot(carat, price, data=diamonds, geom=c("smooth","point"))
- qplot(carat, data=diamonds, geom="density", colour = color)