The R Environment

A high-level overview

Deepayan Sarkar

Indian Statistical Institute, Delhi

6 October 2010

An article in the New York Times

The New York Times

Business Computing

Data Analysts Captivated by R's Power

Stuart Isett for The New York Times

R first appeared in 1996, when the statistics professors Robert Gentleman, left, and Ross Ihaka released the code as a free software package.

By ASHLEE VANCE
Published: January 6, 2009

To some people R is just the 18th letter of the alphabet. To others, it's the rating on racy movies, a measure of an attic's insulation or what pirates in movies say.

TWITTER
SIGN IN TO E-MAIL
PRINT
SINGLE PAGE

More Articles in Technology »

Subscribe to Technology RSS Feeds

Technology News
Internet | Start-Ups
Business Computing | Companies
Bits Blog
Personal Tech
Pogue's Posts

MOST POPULAR - TECHNOLOGY

E-MAILED | BLOGGED | VIEWED

1. 2 Brothers Await Broad Use of Medical E-Records
2. Pogue's Posts: Line2: Turn an iPod Touch into an iPhone
3. The Social Network That Gets Down to Business
4. Verizon Wireless to Pay Millions in Refunds
5. State of the Art: A Simple Swipe on a Phone, and You're Paid
6. Toshiba to Offer 3-D TV That Does Not Require Special Glasses

From the article

R is [...] a popular programming language used by a growing number of data analysts inside corporations and academia.

Companies as diverse as Google, Pfizer, Merck, Bank of America, the InterContinental Hotels Group and Shell use it.

What exactly is R?

- R is a language and environment for statistical computing and graphics.
- It is a Free Software project which is similar to the S language and environment which was developed at Bell Laboratories by John Chambers and colleagues.
- R can be considered as a different implementation of S.

What exactly is R?

- R is a language and environment for statistical computing and graphics.
- It is a Free Software project which is similar to the S language and environment which was developed at Bell Laboratories by John Chambers and colleagues.
- R can be considered as a different implementation of S.

The origins of S

- Developed at Bell Labs (statistics research department)
- Primary goals
  - Interactivity: Exploratory Data Analysis vs batch mode
  - Flexibility: Novel vs routine methodology
  - Practical: For actual use, not (just) academic research

The evolution of S

1970s Initial implementation (Fortran, mostly internal use)

1980s UNIX version, wider distribution in academia
"New S" (major redesign)

1990s "Statistical modeling language". Licensing (S-PLUS).
Addition of formal object-oriented programming

ACM Software System Award

1983 UNIX

1986 TeX

1989 PostScript

1995 World-Wide Web

1995 NCSA Mosaic

1998 To John Chambers

"For The S system, which has forever altered how people analyze, visualize, and manipulate data."

1999 Apache

2002 Java

ACM Software System Award

1983 UNIX

1986 TeX

1989 PostScript

1995 World-Wide Web

1995 NCSA Mosaic

1998 To John Chambers

"For The S system, which has forever altered how people analyze, visualize, and manipulate data."

1999 Apache

2002 Java

R

≈1993 Started as teaching tool by Robert Gentleman & Ross Ihaka at the Univesity of Auckland

... because S didn't run on the Apple computers they had

1995 Convinced by Martin Mächler to release as Free Software

2000 Version 1.0 released

R

≈1993 Started as teaching tool by Robert Gentleman & Ross Ihaka at the Univesity of Auckland

... because S didn't run on the Apple computers they had

1995 Convinced by Martin Mächler to release as Free Software

2000 Version 1.0 released

R

$\approx$1993 Started as teaching tool by Robert Gentleman & Ross Ihaka at the Univesity of Auckland

... because S didn't run on the Apple computers they had

1995 Convinced by Martin Mächler to release as Free Software

2000 Version 1.0 released

R

- Not really that different from S, but the Free Software/Open Source development model has made it a larger success

Why the success?

- Rapid prototyping
- Interfaces to external software
- Easy dissemination of research (through packages)
- Reproducible research

Why the success?

- Rapid prototyping
- Interfaces to external software
- Easy dissemination of research (through packages)
- Reproducible research

Why the success?

- Rapid prototyping
- Interfaces to external software
- Easy dissemination of research (through packages)
- Reproducible research

Rapid prototyping

S is a programming language and environment for all kinds of computing involving data. It has a simple goal:

To turn ideas into software, quickly and faithfully

—John Chambers
Programming with Data

S is a programming language

```
> fibonacci = function(n) {
      if (n < 2)
          x = seq(length = n) - 1
      else {
          x = c(0, 1)
          while (length(x) < n) {
              x = c(x, sum(tail(x, 2)))
          }
      }
      x
  }
> fib10 = fibonacci(10)
> fib10

 [1]  0  1  1  2  3  5  8 13 21 34
```

Easy to call C for efficiency

File fib.c:

```
#include <Rdefines.h>

SEXP do_fibonacci(SEXP nr)
{
    int i, n = INTEGER_VALUE(nr);
    SEXP ans = PROTECT(NEW_INTEGER(n));
    int *x = INTEGER_POINTER(ans);
    x[0] = 0; x[1] = 1;
    for (i = 2; i < n; i++) x[i] = x[i-1] + x[i-2];
    UNPROTECT(1);
    return ans;
}
```

Easy to call C for efficiency

```
$ R CMD SHLIB fib.c
gcc -std=gnu99 -shared -L/usr/local/lib64 -o fib.so fib.o -
make[1]: Leaving directory `/home/deepayan/tmp/ROverview'
```

```
> dyn.load("fib.so")
> cfib10 = .Call("do_fibonacci", as.integer(10))
> cfib10
```

```
 [1]  0  1  1  2  3  5  8 13 21 34
```

Vectorized computation

The Fibonacci series has a closed-form expression as well.

$$F(n) = \frac{\phi^n - (1-\phi)^n}{\sqrt{5}}, \quad \text{where} \quad \phi = \frac{1+\sqrt{5}}{2}$$

```
> phi <- (1 + sqrt(5)) / 2
> n <- 0:9
> n

 [1] 0 1 2 3 4 5 6 7 8 9

> (phi^n - (1 - phi)^n) / sqrt(5)

 [1]  0  1  1  2  3  5  8 13 21 34
```

S is a programming language

- ...designed for interactive use
- ...with a focus on data analysis
  - Basic data structures are vectors
  - Large collection of statistical functions
  - Advanced statistical graphics capabilities

S is a programming language

- ...designed for interactive use
- ...with a focus on data analysis
  - Basic data structures are vectors
  - Large collection of statistical functions
  - Advanced statistical graphics capabilities

The barley data

	yield	variety	year	site
1	27.00	Manchuria	1931	University Farm
2	48.87	Manchuria	1931	Waseca
3	27.43	Manchuria	1931	Morris
4	39.93	Manchuria	1931	Crookston
5	32.97	Manchuria	1931	Grand Rapids
6	28.97	Manchuria	1931	Duluth
7	43.07	Glabron	1931	University Farm
8	55.20	Glabron	1931	Waseca
9	28.77	Glabron	1931	Morris
10	38.13	Glabron	1931	Crookston
11	29.13	Glabron	1931	Grand Rapids
12	29.67	Glabron	1931	Duluth
13	35.13	Svansota	1931	University Farm
14	47.33	Svansota	1931	Waseca
15	25.77	Svansota	1931	Morris

The “SAS approach”

`data/barley_models.txt`

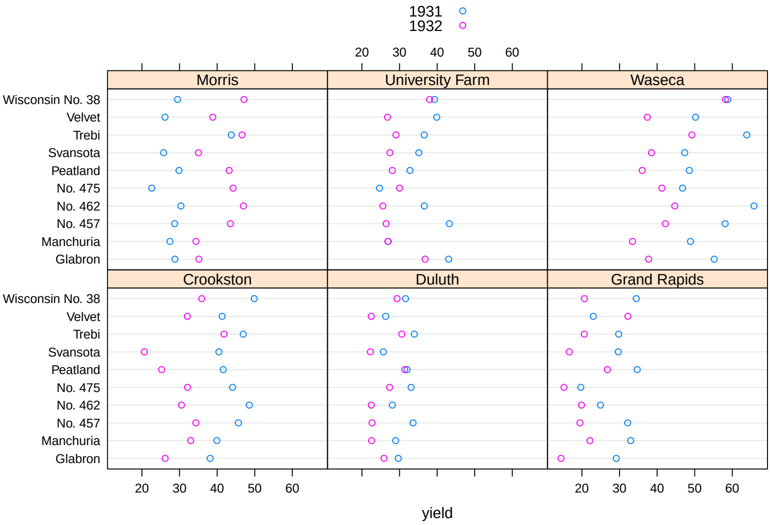
The barley data

```
> barleyYield = read.csv("data/barley.csv", header = TRUE)
> barleyYield
      yield           variety year             site
1  27.00000         Manchuria 1931 University Farm
2  48.86667         Manchuria 1931           Waseca
3  27.43334         Manchuria 1931           Morris
4  39.93333         Manchuria 1931        Crookston
5  32.96667         Manchuria 1931     Grand Rapids
6  28.96667         Manchuria 1931           Duluth
7  43.06666           Glabron 1931 University Farm
8  55.20000           Glabron 1931           Waseca
9  28.76667           Glabron 1931           Morris
10 38.13333           Glabron 1931        Crookston
11 29.13333           Glabron 1931     Grand Rapids
12 29.66667           Glabron 1931           Duluth
13 35.13333          Svansota 1931 University Farm
14 47.33333          Svansota 1931           Waseca
15 25.76667          Svansota 1931           Morris
```

The barley data

```
> str(barleyYield)

'data.frame': 120 obs. of  4 variables:
 $ yield   : num  27 48.9 27.4 39.9 33 ...
 $ variety: Factor w/ 10 levels "Glabron","Manchuria",..: 2 2 2
 $ year    : Factor w/ 2 levels "1931","1932": 1 1 1 1 1 1 1 1 1
 $ site    : Factor w/ 6 levels "Crookston","Duluth",..: 5 6 4 1
```

```
> dotplot(variety ~ yield | site, groups = year,
          data = barleyYield, auto.key = TRUE)
```



Fitting models

All two-factor interactions:

```
> fm1 = lm(yield ~ (variety + site + year)^2,
           data = barleyYield)
```

Main effects only:

```
> fm2 = lm(yield ~ variety + site + year,
           data = barleyYield)
```

Hypothesis testing

```
> anova(fm2, fm1)

Analysis of Variance Table

Model 1: yield ~ variety + site + year
Model 2: yield ~ (variety + site + year)^2
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    104 4176.2
2     45  658.5 59    3517.8 4.0747 1.523e-06
```

Sequential ANOVA

```
> anova(fm1)

Analysis of Variance Table

Response: yield
             Df Sum Sq Mean Sq F value    Pr(>F)
variety       9 1052.6  116.95  7.9927 6.052e-07
site          5 6633.9 1326.77 90.6736 < 2.2e-16
year          1  847.3  847.30 57.9058 1.283e-09
variety:site 45 1205.8   26.79  1.8312   0.02259
variety:year  9  209.8   23.31  1.5929   0.14646
site:year     5 2102.2  420.44 28.7337 5.821e-13
Residuals    45  658.5   14.63
```

Further inspection

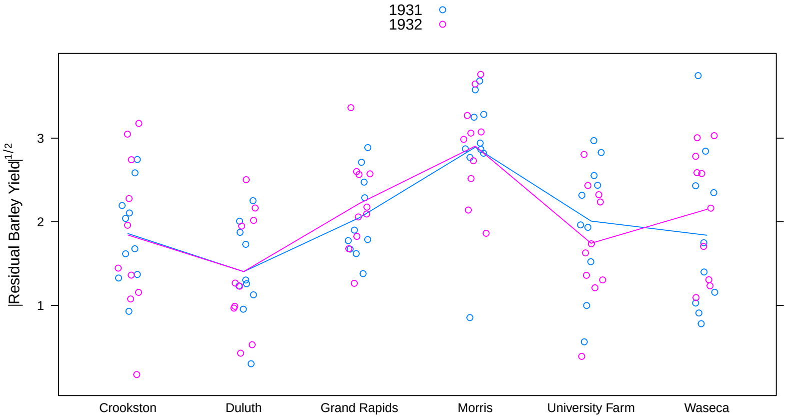
```
> coef(fm2)
            (Intercept)        varietyManchuria
             38.9983317              -1.8777758
         varietyNo. 457          varietyNo. 462
              2.5055583               2.0361150
         varietyNo. 475         varietyPeatland
             -1.5805550               0.8388900
        varietySvansota            varietyTrebi
             -2.9638883               6.0583275
          varietyVelvet varietyWisconsin No. 38
             -0.2805567               6.0527800
             siteDuluth       siteGrand Rapids
             -9.4233315             -12.4883315
             siteMorris     siteUniversity Farm
             -2.0199980              -4.7533310
             siteWaseca                year1932
             10.6883330              -5.3144453
```

Further inspection

```
> round(residuals(fm2), digits = 3)
      1       2       3       4       5       6       7       8 
 -5.367   1.058  -7.667   2.813   8.334   1.269   8.822   5.513 
      9      10      11      12      13      14      15      16 
 -8.212  -0.865   2.623   0.092   3.852   0.611  -8.248   4.432 
     17      18      19      20      21      22      23      24 
  6.121  -0.911   5.936   0.827 -10.564   2.616  -3.196  -2.994 
     25      26      27      28      29      30      31      32 
 -3.737   8.088   0.730   1.877  -2.802  -1.700   6.516   5.908 
     33      34      35      36      37      38      39      40 
-10.784   4.163   3.151   1.519   0.319  14.044  -8.648   7.532 
     41      42      43      44      45      46      47      48 
 -3.613  -3.511  -2.317  -1.959  -7.951   1.763   7.351   1.586 
     49      50      51      52      53      54      55      56 
 -7.998  -1.339 -12.798   6.682  -5.229   5.072  -0.998   3.061 
     57      58      59      60      61      62      63      64 
-13.564   4.816   1.904  -4.028  -0.153  -9.028   4.581   1.161 
     65      66      67      68      69      70      71      72 
```

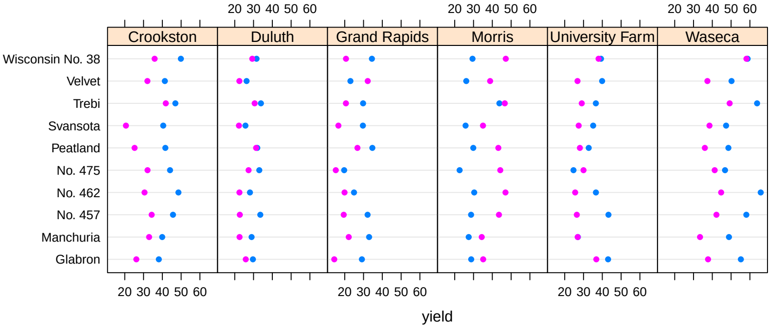
Residual plot

```
> stripplot(sqrt(abs(residuals( fm2 ))) ~ site, groups = year,
        data = barleyYield, jitter.data = TRUE,
        auto.key = TRUE, type = c("p", "a"),
        ylab = expression(abs("Residual Barley Yield")^{1/2}))
```



A closer look

```
> dotplot(variety ~ yield | site,
          groups = year, data = barleyYield,
          pch = 16, layout = c(6, 1))
```



A revised model

```
> morris = barleyYield$site == "Morris"
> barleyYield$year[ morris ] =
      ifelse(barleyYield$year[morris] == "1931", "1932", "1931")
> fm1 = lm(yield ~ (variety + site + year)^2,
           data = barleyYield)
> fm2 = lm(yield ~ variety + site + year,
           data = barleyYield)
> anova(fm2, fm1)
```

```
Analysis of Variance Table

Model 1: yield ~ variety + site + year
Model 2: yield ~ (variety + site + year)^2
  Res.Df     RSS Df Sum of Sq      F  Pr(>F)
1    104 2378.34
2     45  713.74 59    1664.6 1.7788 0.02296
```

Summary

The “S approach” is to work with objects.

- Model fits produce objects, usually stored as variables
- Queried *interactively* for further analysis

```
> anova(fm)
> summary(fm)
> residuals(fm)
```

A mixed effect model

$$y_i = \mu + \alpha_i + \beta_j + b_k + \varepsilon_{ijk}$$

where

y_i = yield of barley

μ = overall mean

α_i = additive effect of i-th variety

β_j = additive effect of j-th year

$b_k \sim \mathcal{N}(0, \tau^2)$ = effect of k-th site

$\varepsilon_{ijk} \sim \mathcal{N}(0, \sigma^2)$ = error

All b_k, ε_{ijk} independent

A mixed effect model

- Parameters μ, α_i, β_j and τ^2, σ^2
- Difficult to find MLEs for such models in general
- Functionality provided by two "add-on" packages
  - nlme Stable, widely used ($\approx$ 2000)
  - lme4 Experimental, active development

A mixed effect model

- Parameters μ, α_i, β_j and τ^2, σ^2
- Difficult to find MLEs for such models in general
- Functionality provided by two "add-on" packages
  - nlme Stable, widely used ($\approx$ 2000)
  - lme4 Experimental, active development

A mixed effect model

```
> library(package = "nlme")
> fm3 = lm(yield ~ variety + year, data = barleyYield)
> fm4 = lme(yield ~ variety + year, data = barleyYield,
           random = ~ 1 | site, method = "ML")
> anova(fm4, fm3)
```

```
    Model df      AIC      BIC    logLik   Test  L.Ratio p-value
fm4     1 13 754.9219 791.1593 -364.4610
fm3     2 12 882.8063 916.2562 -429.4032 1 vs 2 129.8844  <.0001
```

A twist

We are testing

$$H_0 : \tau^2 = 0$$

- Falls on boundary of parameter space
- Assumptions for asymptotic χ^2 distribution in LRT violated

Some exploration

```
> yhat = fitted(fm4, level = 0)
> sigma.hat = sqrt(sum(residuals(fm4)^2) / length(yhat))
> barleyYield$ynew =
      yhat + rnorm(length(yhat),
                   mean = 0, sd = sigma.hat)
> fm5 = lm(ynew ~ variety + year, data = barleyYield)
> fm6 = lme(ynew ~ variety + year, data = barleyYield,
            random = ~ 1 | site, method = "ML")
> anova(fm6, fm5)
```

```
    Model df      AIC      BIC    logLik   Test      L.Ratio
fm6     1 13 713.6566 749.8940 -343.8283
fm5     2 12 711.6566 745.1065 -343.8283 1 vs 2 1.234696e-07
    p-value
fm6
fm5  0.9997
```

```
> a <- anova(fm6, fm5)
> str(a)
```

```
Classes 'anova.lme' and 'data.frame': 2 obs. of  9 variables:
 $ call    : Factor w/ 2 levels "lme.formula(fixed = ynew ~ varie
 $ Model   : int  1 2
 $ df      : num  13 12
 $ AIC     : num  714 712
 $ BIC     : num  750 745
 $ logLik  : num  -344 -344
 $ Test    : Factor w/ 2 levels "","1 vs 2": 1 2
 $ L.Ratio: num  NA 1.23e-07
 $ p-value: num  NA 1
 - attr(*, "rt")= int 2
 - attr(*, "verbose")= logi FALSE
```

```
> a$logLik
```

```
[1] -343.8283 -343.8283
```

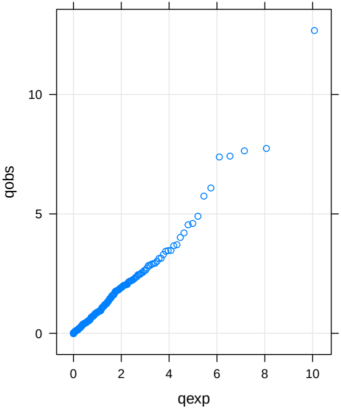
```
> -2 * diff(a$logLik)
```

```
[1] -1.234696e-07
```

```
> LRTstat.sim = function()
  {
      barleyYield$ynew =
          yhat + rnorm(length(yhat), mean = 0, sd = sigma.hat)
      fm5 = lm(ynew ~ variety + year, data = barleyYield)
      fm6 = lme(ynew ~ variety + year, data = barleyYield,
                random = ~ 1 | site, method = "ML")
      -2 * diff(anova(fm6, fm5)$logLik)
  }
> LRTstat.sim()

[1] -8.58613e-08
```

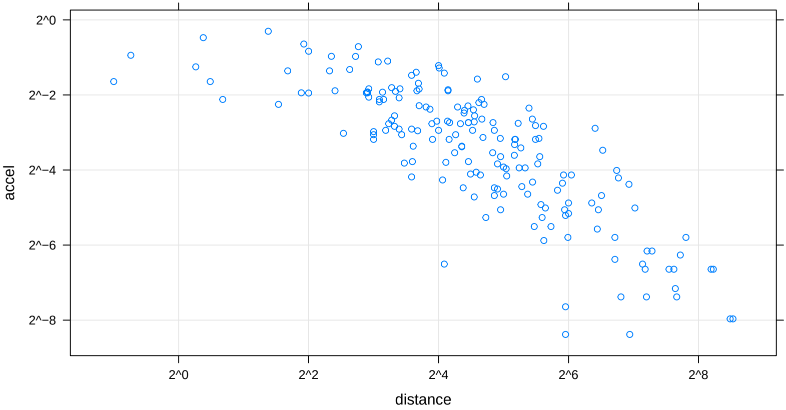
```
> replicate(10, LRTstat.sim())

 [1] -6.844800e-08  1.161523e-01 -1.267522e-07  1.850964e-01
 [5] -8.329323e-08 -1.193092e-07 -9.316761e-08 -1.204604e-07
 [9] -7.669269e-08  3.967130e-02

> sim1000 <- replicate(1000, LRTstat.sim())
> table(zapsmall(sim1000) == 0)

FALSE  TRUE
  345   655
```

```
> nzero <- round(1000 * 2 / 3)
> qobs <- sort(sim1000)
> qexp <- c(rep(0, nzero),
            qchisq(ppoints(1000 - nzero), df = 1))
> xyplot(qobs ~ qexp, grid = TRUE, aspect = "iso")
```

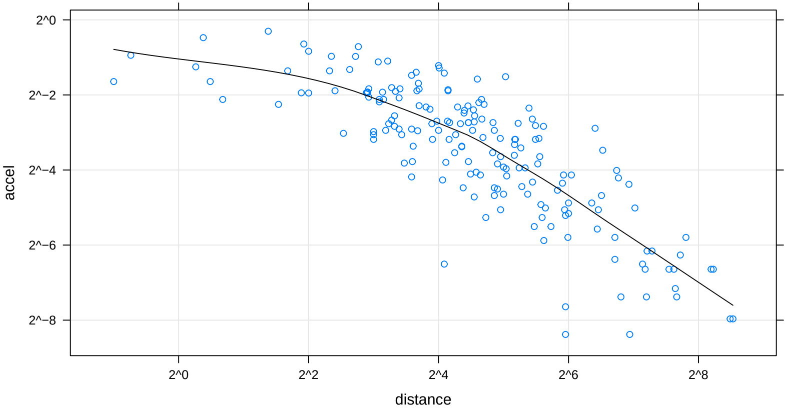


Even graphics is programmable

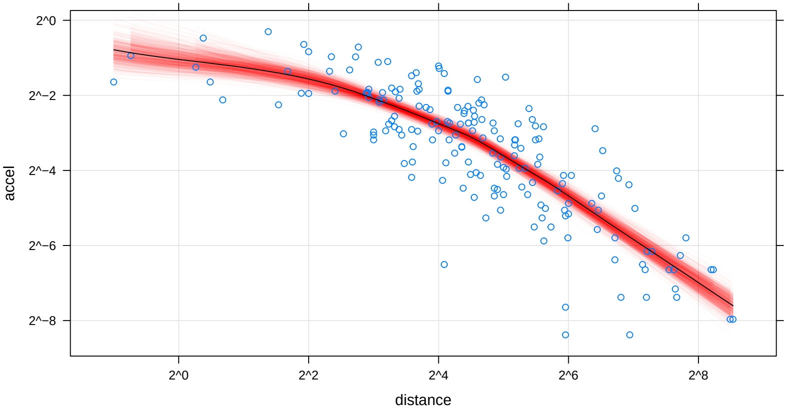
```
> data(Earthquake, package = "nlme")
> xyplot(accel ~ distance, Earthquake,
         scales = list(log = 2), grid = TRUE)
```



```
> xyplot(accel ~ distance, Earthquake, scales = list(log = 2),
         panel = function(x, y, ...) {
             panel.grid(h = -1, v = -1)
             panel.points(x, y, ...)
             panel.loess(x, y, col = "black")
         })
```



```
> xyplot(accel ~ distance, Earthquake, scales = list(log = 2),
         panel = function(x, y, ...) {
             panel.grid(h = -1, v = -1)
             n <- length(x)
             for (i in 1:1000) {
                 bs.id <- sample(1:n, replace = TRUE) ## SRSWR
                 panel.loess(x[bs.id], y[bs.id],
                             col = "red", alpha = 0.02)
             }
             panel.points(x, y, ...)
             panel.loess(x, y, col = "black")
         })
```


2^0
2^−2
2^−4
2^−6
2^−8
accel
2^0
2^2
2^4
2^6
2^8
distance

S indexing

This works using vectorized indexing in S:

```
> a = c(21, 29, 31)
> a[ c(1, 2, 2, 1, 1) ]
[1] 21 29 29 21 21
```

Another example...

A random walk on the lattice

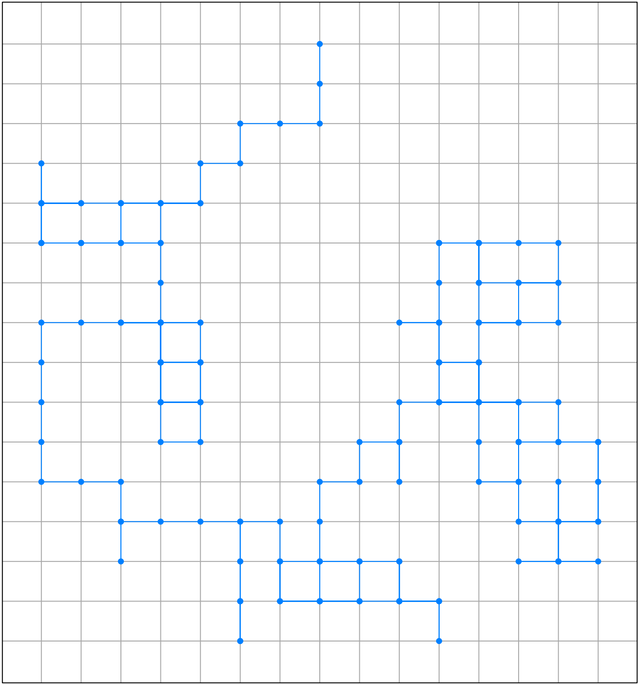
```
> d <- data.frame(x = c(-1, 0, 1, 0),
                  y = c( 0,-1, 0, 1))
> d
   x  y
1 -1  0
2  0 -1
3  1  0
4  0  1
```

```
> eps <- d[sample(1:4, 15, replace = TRUE), ]
> eps
```

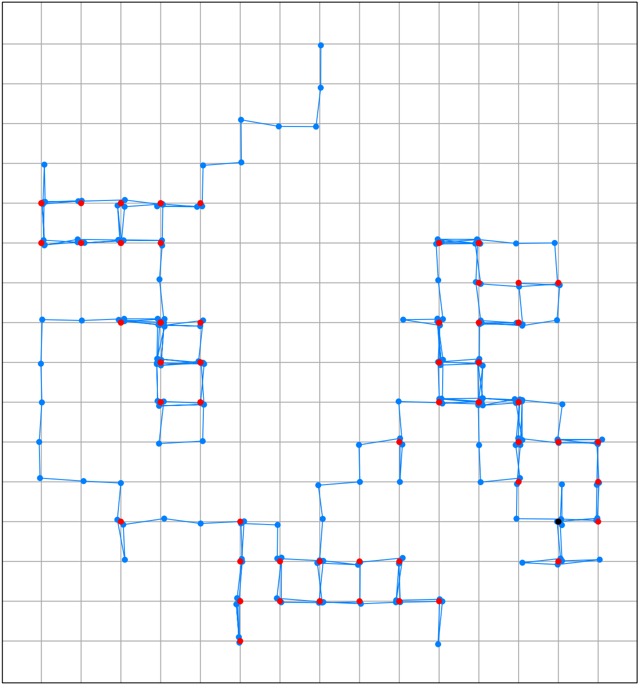
```
     x  y
1   -1  0
3    1  0
4    0  1
2    0 -1
4.1  0  1
2.1  0 -1
2.2  0 -1
1.1 -1  0
3.1  1  0
4.2  0  1
2.3  0 -1
1.2 -1  0
1.3 -1  0
2.4  0 -1
4.3  0  1
```

```
> rw <- lapply(rbind(0, eps), cumsum)
> rw

$x
 [1]  0 -1  0  0  0  0  0  0 -1  0  0  0 -1 -2 -2 -2

$y
 [1]  0  0  0  1  0  1  0 -1 -1 -1  0 -1 -1 -1 -2 -1
```

```
> eps <- d[sample(1:4, 200, replace = TRUE), ]
> rw <- lapply(rbind(0, eps), cumsum)
> xyplot(y ~ x, data = rw, type = "o", pch = 16,
         scales = list(draw = FALSE),
         xlab = "", ylab = "", aspect = "iso",
         abline = list(col = "darkgrey",
                       v = unique(rw$x),
                       h = unique(rw$y)))
```



```
> xyplot(y ~ x, data = rw,
         scales = list(draw = FALSE),
         xlab = "", ylab = "", aspect = "iso",
         panel = function(x, y) {
             panel.abline(col = "darkgrey",
                          v = unique(x),
                          h = unique(y))
             n <- length(x)
             panel.points(x + runif(n, -0.1, 0.1),
                          y + runif(n, -0.1, 0.1),
                          type = "o", pch = 16)
             dup <- duplicated(data.frame(x, y))
             panel.points(x[dup], y[dup], pch = 16, col = "red")
             panel.points(0, 0, pch = 16, col = "black")
         })
```



Powerful built-in tools

$+$

Programming language

$\Downarrow$

Flexibility

Interfacing external software

- Not all cool software developed by R community
- Core open source philisophy: code re-use

"don't rediscover the wheel!"

- R facilitates interfacing with external software
- Three examples:
  - C++ Sparse matrix library (Tim Davis, U of Florida)
  - Graphviz (AT&T research)
  - Qt (formerly Trolltech, now Nokia)

Interfacing external software

- Not all cool software developed by R community
- Core open source philisophy: code re-use

"don't rediscover the wheel!"

- R facilitates interfacing with external software
- Three examples:
  - C++ Sparse matrix library (Tim Davis, U of Florida)
  - Graphviz (AT&T research)
  - Qt (formerly Trolltech, now Nokia)

Sparse matrix computations

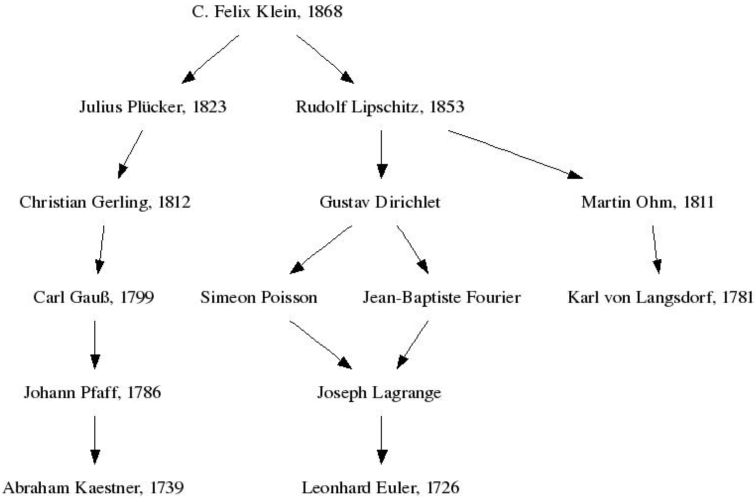
- Occurs naturally in statistical modeling

$$\boldsymbol{X}^T \boldsymbol{X} \boldsymbol{\beta} = \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{y}$$

- Interface to UMFPACK
  `http://www.cise.ufl.edu/research/sparse/umfpack/`
  in R package Matrix, used by lme4

Graphviz

- Open source software for graph layout / visualization



Graphviz

- Open source software for graph layout / visualization
- Uses own interface and graph specification language
- Also provides a C library (`Cgraph`)
- Used by the R package Rgraphviz for layout calculations
- Rendering done using R graphics

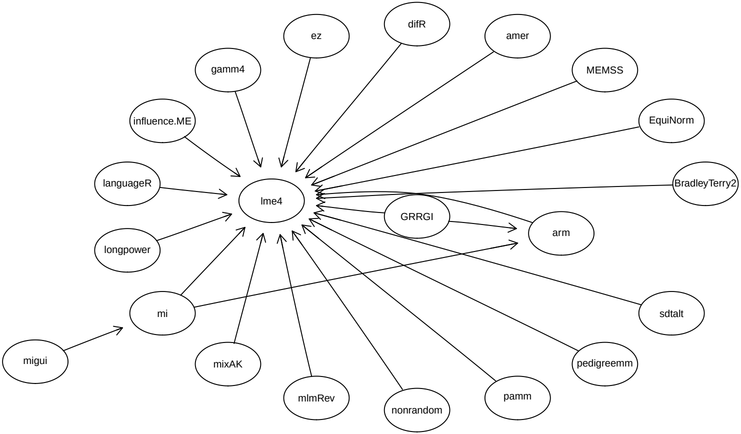
R package dependency graph

```
> library(graph)
> library(Rgraphviz)
> library(pkgDepTools)
> g = makeDepGraph("http://cran.r-project.org",
                   keep.builtin = TRUE)

> g

A graphNEL graph with directed edges
Number of Nodes = 2589
Number of Edges = 4728
```

Reverse dependencies

```
> revDepGraph <- function(g, pkg) {
      olen <- 0
      pkgKeep <- pkg
      elist <- g@edgeL
      elist <- elist[ !(sapply(elist, is.null)) ]
      while (length(pkgKeep) > olen) {
          olen <- length(pkgKeep)
          w <- which(g@nodes %in% pkgKeep)
          revdep <- sapply(elist, function(x) any(w %in% x$edges
          pkgKeep <- union(pkgKeep, names(revdep)[revdep])
      }
      subGraph(pkgKeep, g)
  }
> gsub <- revDepGraph(g, "lme4")
> gsub

A graphNEL graph with directed edges
Number of Nodes = 21
Number of Edges = 22
```

```
> library(Rgraphviz)
> graph.par(nodes = list(shape = "ellipse"))
> gl = layoutGraph(gsub, layoutType = "twopi")
> renderGraph(gl)
```



Qt

- Powerful cross-platform GUI programming library (C++)
- Used to create KDE, Skype, Opera (browser)
- "Language bindings" make features accessible from R
- User can program in R, not C++
- `qtdemo.R`

Qt

- Powerful cross-platform GUI programming library (C++)
- Used to create KDE, Skype, Opera (browser)
- "Language bindings" make features accessible from R
- User can program in R, not C++
- `qtdemo.R`

Dissemination of research

- Rapid prototyping $\Rightarrow$ quick implementation of research ideas
- Well-structured packaging system allows dissemination
- CRAN: Comprehensive R Archive Network: > 2500 packages
- Other specialized collections (Bioconductor, Omegahat)

Reproducible research

- Reproducibility: a core principle of the scientific method
- Difficult in biology, physics, etc. but conceptually trivial for computational experiments
- ...But publications often leave out details (possibly nontrivial)

Reproducible research

- Reproducibility: a core principle of the scientific method
- Difficult in biology, physics, etc. but conceptually trivial for computational experiments
- ...But publications often leave out details (possibly nontrivial)

Sweave

- Inspired by literate documents (Knuth)
- Enables mixing of R code and LaTeX
- "Source file" reproduces both analysis and report
- Reproducible research + convenience
- `rnw/graph.Rnw`

Sweave

- Inspired by literate documents (Knuth)
- Enables mixing of R code and LaTeX
- "Source file" reproduces both analysis and report
- Reproducible research + convenience
- `rnw/graph.Rnw`

Summary

- R is a feature-rich interactive language + environment ideally suited to data analysis as well as other kinds of numerical computations
- Some learning required before it can be used effectively
- Typical mind-blocks for newcomers:
  - R is *not* C!
  - Vectorization (easy to get past with a little experience)
  - Functional approach to programming
  - "Computing on the language", e.g., replicate(10, LRTstat.sim())

Summary

- R is a feature-rich interactive language + environment ideally suited to data analysis as well as other kinds of numerical computations
- Some learning required before it can be used effectively
- Typical mind-blocks for newcomers:
  - R is *not* C!
  - Vectorization (easy to get past with a little experience)
  - Functional approach to programming
  - "Computing on the language", e.g., replicate(10, LRTstat.sim())

Summary

- R is a feature-rich interactive language + environment ideally suited to data analysis as well as other kinds of numerical computations
- Some learning required before it can be used effectively
- Typical mind-blocks for newcomers:
  - R is *not* C!
  - Vectorization (easy to get past with a little experience)
  - Functional approach to programming
  - "Computing on the language", e.g., `replicate(10, LRTstat.sim())`

Questions?