# R in Actuarial Science
# a brief overview

**Arthur Charpentier**

charpentier.arthur@uqam.ca

http ://freakonometrics.hypotheses.org/

UQÀM Université du Québec à Montréal

Quantact

January 2013, Universiteit van Amsterdam

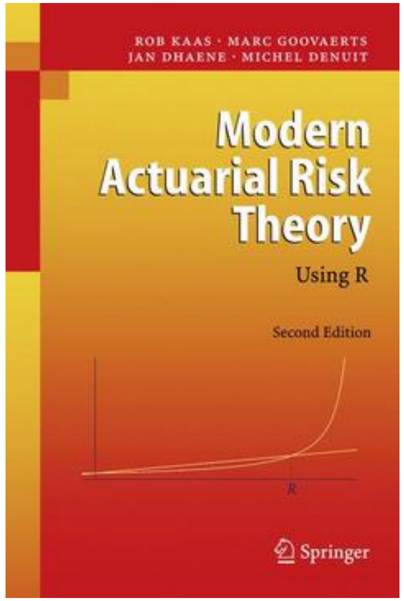
## Agenda

- **Introduction to R**
- **Why R in actuarial science ?**
  - Actuarial science ?
  - A vector-based language
  - A large number of packages and libraries for predictive models
  - Working with (large) databases in R
  - A language to plot graphs
- **Reproducibility issues**
- **Comparing R with other statistical softwares**
  - R in the insurance industry and amongst statistical researchers
  - R versus MsExcel Matlab, SAS, SPSS, etc
  - The R community
- **Conclusion ( ?)**

## R

*"R (and S) is the 'lingua franca' of data analysis and statistical computing, used in academia, climate research, computer science, bioinformatics, pharmaceutical industry, customer analytics, data mining, finance and by some insurers. Apart from being stable, fast, always up-to-date and very versatile, the chief advantage of R is that it is available to everyone free of charge. It has extensive and powerful graphics abilities, and is developing rapidly, being the statistical tool of choice in many academic environments."*

**Appendix A**
**The 'R' in Modern ART**



# A brief history of R

R is based on the S statistical programming language developed by Joe Chambers at Bell labs in the 80's


R is an open-source implementation of the S language, developed by Robert Gentlemn and Ross Ihaka

## actuarial science ?

- students in actuarial programs
- researchers in actuarial science
- actuaries in insurance companies (or consulting firms, or financial institutions, etc)

# Using a vector-based language for life contingencies

A life table is a vector

```
> TD[39:52,]
   Age    Lx
39  38 95237
40  39 94997
41  40 94746
42  41 94476
43  42 94182
44  43 93868
45  44 93515
46  45 93133
47  46 92727
48  47 92295
49  48 91833
50  49 91332
51  50 90778
52  51 90171
```

```
> TV[39:52,]
   Age    Lx
   38 97753
   39 97648
   40 97534
   41 97413
   42 97282
   43 97138
   44 96981
   45 96810
   46 96622
   47 96424
   48 96218
   49 95995
   50 95752
   51 95488
```

*TABLE I.*

| Ages par années. | Survivans selon M. Halley. | N'ayant pas eu la pet. vérole. | Ayant eu la pet. vérol. | Prenant la pet. vérole pendant ch. année. | Morts de la pet. vérole pendant chaq. ann. | Somme des morts de la pet. vérole. | Morts par d'autres maladies pend. chaq. année. |
|---|---|---|---|---|---|---|---|
| 0 | 1300 | 1300 | 0 | | | | |
| 1 | 1000 | 896 | 104 | 137 | 17,1 | 17,1 | 283 |
| 2 | 855 | 685 | 170 | 99 | 12,4 | 29,5 | 133 |
| 3 | 798 | 571 | 227 | 78 | 9,7 | 39,2 | 47 |
| 4 | 760 | 485 | 275 | 66 | 8,3 | 47,5 | 30 |
| 5 | 732 | 416 | 316 | 56 | 7,0 | 54,5 | 21 |
| 6 | 710 | 359 | 351 | 48 | 6,0 | 60,5 | 16 |
| 7 | 692 | 311 | 381 | 42 | 5,2 | 65,7 | 12,8 |
| 8 | 680 | 272 | 408 | 36 | 4,5 | 70,2 | 7,5 |
| 9 | 670 | 237 | 433 | 32 | 4,0 | 74,2 | 6 |
| 10 | 661 | 208 | 453 | 28 | 3,5 | 77,7 | 5,5 |
| 11 | 653 | 182 | 471 | 24,4 | 3,0 | 80,7 | 5 |
| 12 | 646 | 160 | 486 | 21,4 | 2,7 | 83,4 | 4,3 |
| 13 | 640 | 140 | 500 | 18,7 | 2,3 | 85,7 | 3,7 |
| 14 | 634 | 123 | 511 | 16,6 | 2,1 | 87,8 | 3,9 |
| 15 | 628 | 108 | 520 | 14,4 | 1,8 | 89,6 | 4,2 |
| 16 | 622 | 94 | 528 | 12,6 | 1,6 | 91,2 | 4,4 |
| 17 | 616 | 83 | 533 | 11,0 | 1,4 | 92,6 | 4,6 |
| 18 | 610 | 72 | 538 | 9,7 | 1,2 | 93,8 | 4,8 |
| 19 | 604 | 63 | 541 | 8,4 | 1,0 | 94,8 | 5 |
| 20 | 598 | 56 | 542 | 7,4 | 0,9 | 95,7 | 5,1 |
| 21 | 592 | 48,5 | 543 | 6,5 | 0,8 | 96,5 | 5,2 |
| 22 | 586 | 42,5 | 543 | 5,6 | 0,7 | 97,2 | 5,3 |
| 23 | 579 | 37 | 542 | 5,0 | 0,6 | 97,8 | 6,4 |
| 24 | 572 | 32,4 | 540 | 4,4 | 0,5 | 98,3 | 6,5 |

## Using a vector-based language for life contingencies

If age $x \in \mathbb{N}_*$, define $\boldsymbol{P} = [{}_kp_x]$, and `p[k,x]` corresponds to ${}_kp_x$.

The (curtate) expectation of life defined as

$$e_x = \mathbb{E}(K_x) = \sum_{k=1}^{\infty} k \cdot {}_{k|1}q_x = \sum_{k=1}^{\infty} {}_kp_x$$

and we can compute $\boldsymbol{e} = [e_x]$ using

```
> life.exp = function(x){sum(p[1:nrow(p),x])}
> e = Vectorize(life.exp)(1:m)
```

The expected present value (or actuarial value) of a temporary life annuity-due is

$$\ddot{a}_{x:\overline{n}|} = \sum_{k=0}^{n-1} \nu^k \cdot {}_kp_x = \frac{1 - A_{x:\overline{n}|}}{1-\nu}$$

## Using a vector-based language for life contingencies

and we can define $\boldsymbol{A} = [\ddot{a}_{x:\overline{n}|}]$ as

```
> for(j in 1:(m-1)){ adot[,j]<-cumsum(1/(1+i)^(0:(m-1))*c(1,p[1:(m-1),j])) }
```

Define similarly the expected present value of a term insurance

$$A^{1}_{x:\overline{n}|} = \sum_{k=0}^{n-1} \nu^{k+1} \cdot {}_{k|}q_x$$

and the associated matrix $\boldsymbol{A} = [A^{1}_{x:\overline{n}|}]$ as

```
> for(j in 1:(m-1)){ A[,j]<-cumsum(1/(1+i)^(1:m)*d[,j]) }
```

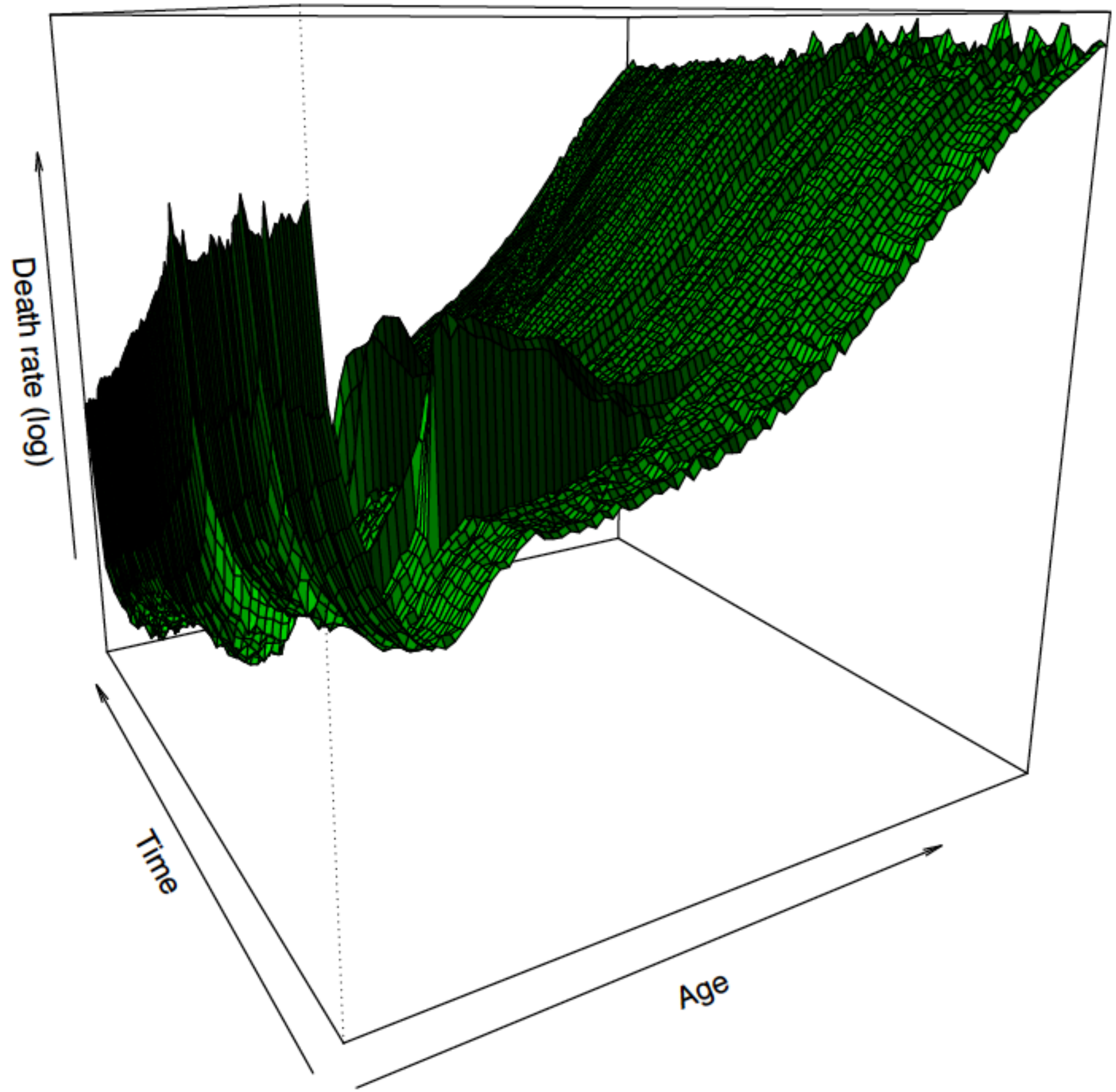
**Remark** : See also Giorgio Alfredo Spedicatos `lifecontingencies` package, and functions `pxt`, `Axn`, `Exn`, etc.

## Using a matrix-based language for prospective life models

Life table $\boldsymbol{L} = [L_x]$ is no longer a matrix (function of age $x$) but a matrix $\boldsymbol{L} = [L_{x,t}]$ function of the date $t$.

```
> t(DTF)[1:10,1:10]
   1899  1900  1901  1902  1903  1904  1905  1906  1907  1908
0 64039 61635 56421 53321 52573 54947 50720 53734 47255 46997
1 12119 11293 10293 10616 10251 10514  9340 10262 10104  9517
2  6983  6091  5853  5734  5673  5494  5028  5232  4477  4094
3  4329  3953  3748  3654  3382  3283  3294  3262  2912  2721
4  3220  3063  2936  2710  2500  2360  2381  2505  2213  2078
5  2284  2149  2172  2020  1932  1770  1788  1782  1789  1751
6  1834  1836  1761  1651  1664  1433  1448  1517  1428  1328
7  1475  1534  1493  1420  1353  1228  1259  1250  1204  1108
8  1353  1358  1255  1229  1251  1169  1132  1134  1083   961
9  1175  1225  1154  1008  1089   981  1027  1025   957   885
```

Similarly, define the force of mortality matrix $\boldsymbol{\mu} = [\mu_{x,t}]$


Death rate (log)
Time
Age

## Using a matrix-based language for prospective life models

Assume - as in Lee & Carter (1992) model - that

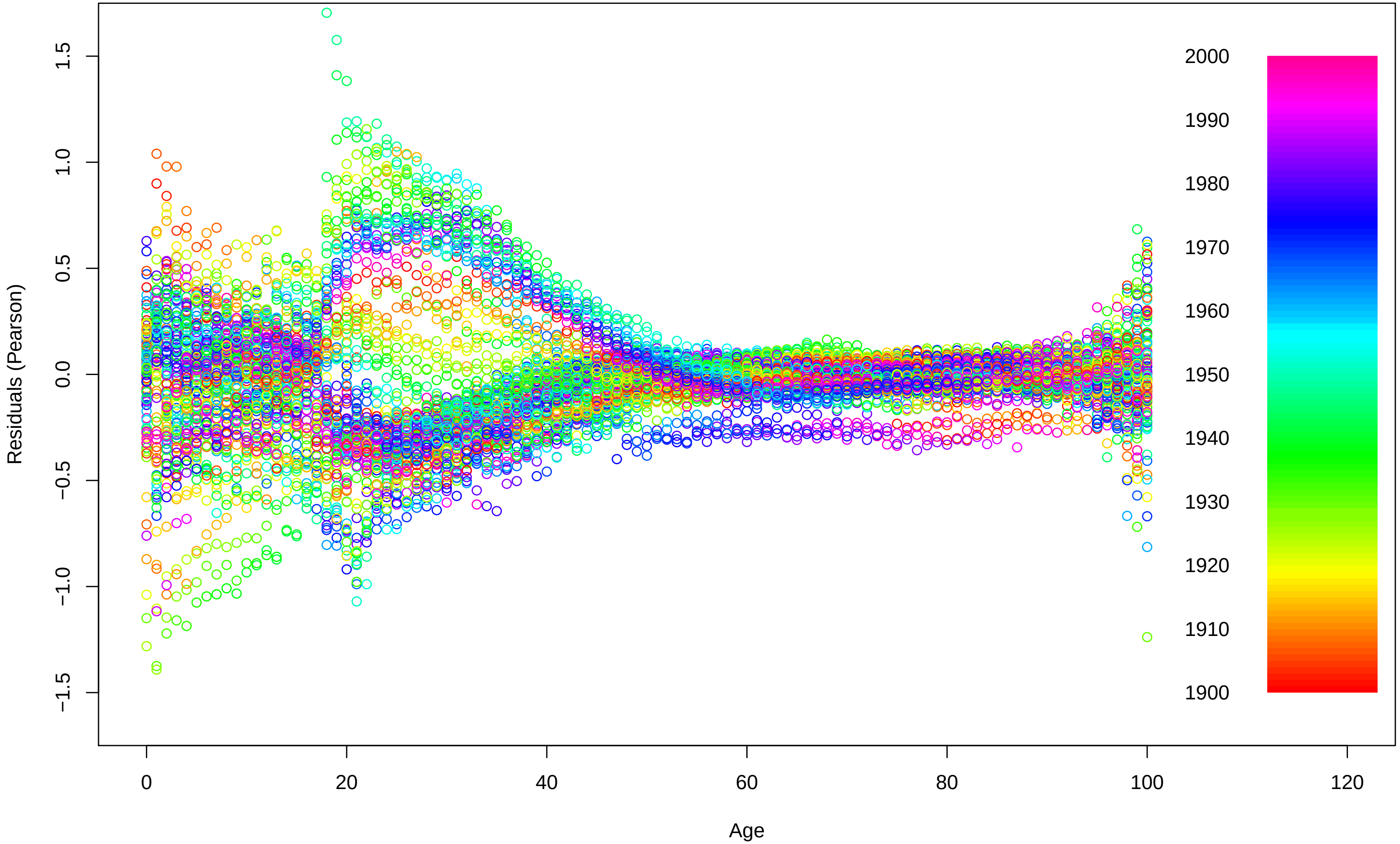
$$\log \mu_{x,t} = \alpha_x + \beta_x \cdot \kappa_t + \varepsilon_{x,t},$$

with some i.i.d. noise $\varepsilon_{x,t}$.

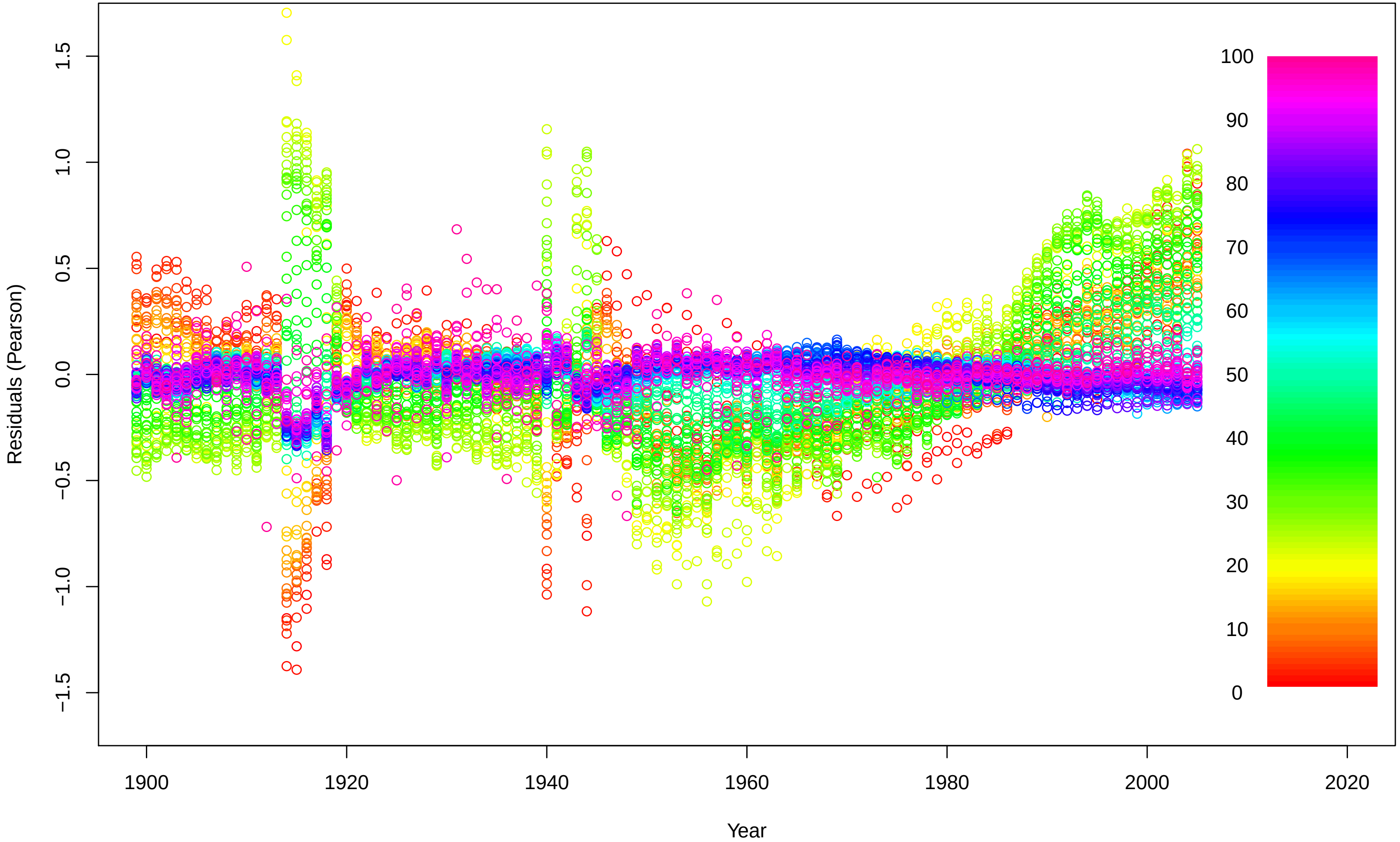
Package `demography` can be used to fit a Lee-Carter model,

```
> library(demography)
> MUH =matrix(DEATH$Male/EXPOSURE$Male,nL,nC)
> POPH=matrix(EXPOSURE$Male,nL,nC)
> BASEH <- demogdata(data=MUH, pop=POPH, ages=AGE, years=YEAR, type="mortality",
+ label="France", name="Hommes", lambda=1)
> RES=residuals(LCH,"pearson")
```

# Residuals in Lee & Carter model



# Residuals in Lee & Carter model



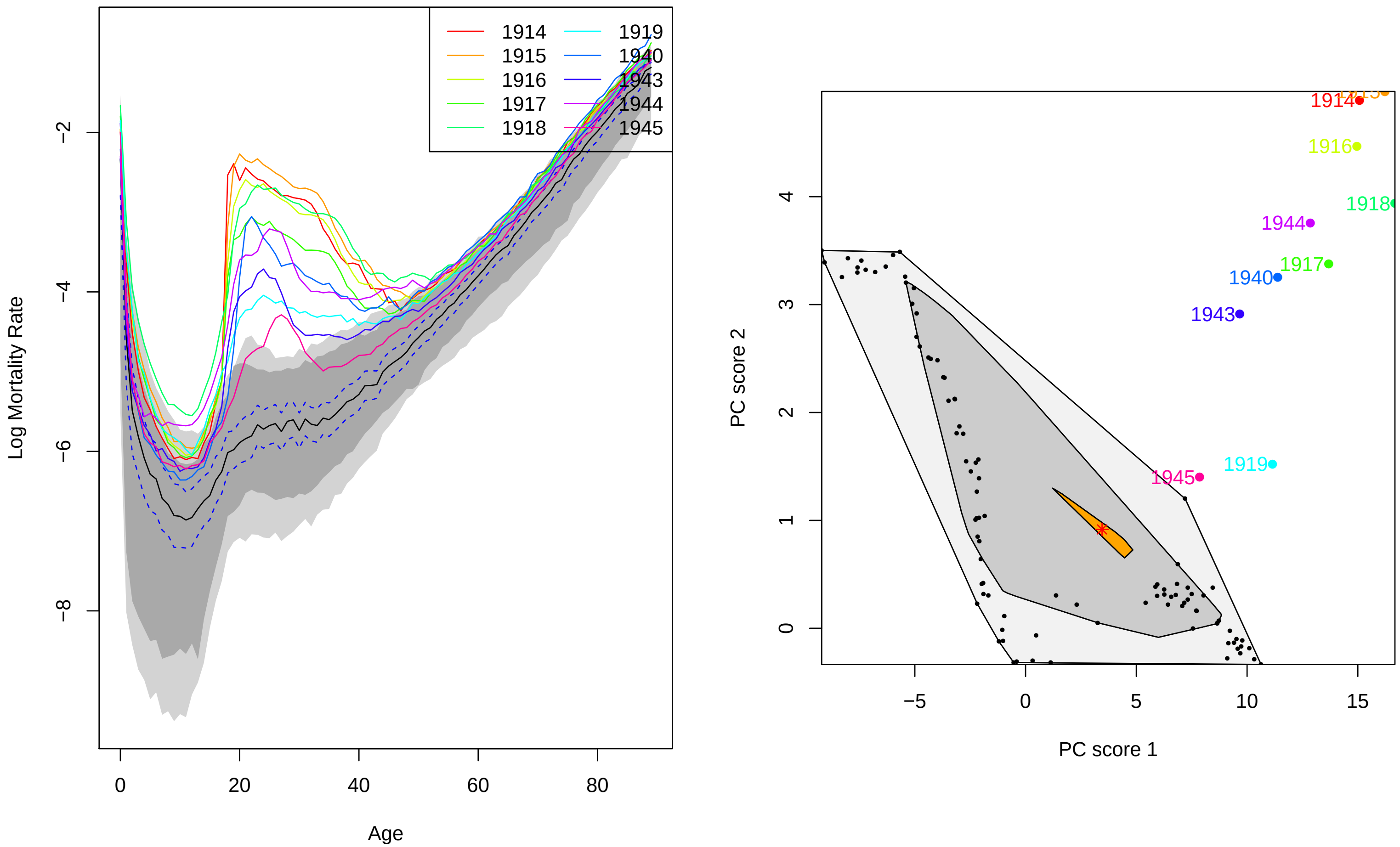
## Using a matrix-based language for prospective life models

One can consider more advanced functions to study mortality, e.g. bagplots, since $\mu_{x,t}$ is a functional time series,

```
> library(rainbow)
> MUH=fts(x = AGE[1:90], y = log(MUH), xname = "Age",yname = "Log Mortality Rate")
> fboxplot(data = MUHF, plot.type = "functional", type = "bag")
> fboxplot(data = MUHF, plot.type = "bivariate", type = "bag")
```

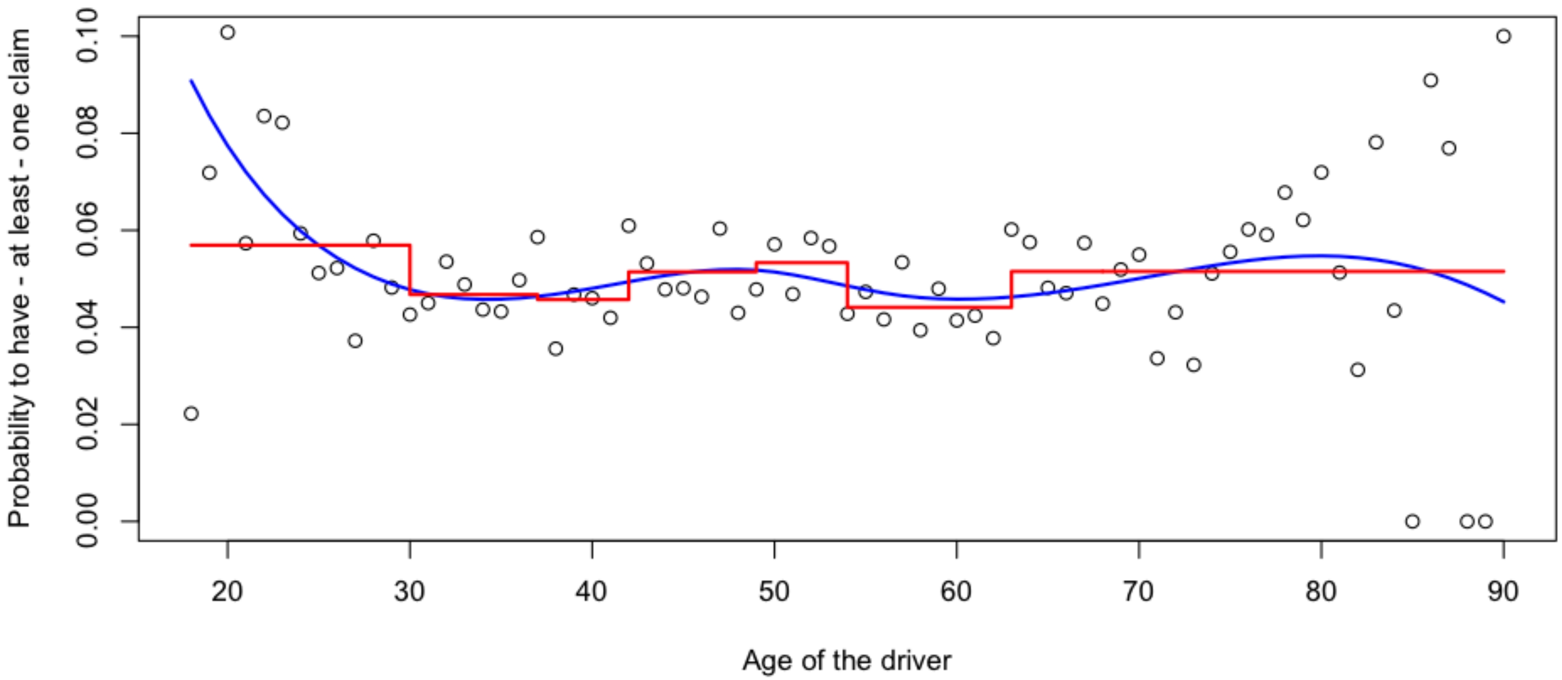
**Source** : http ://robjhyndman.com/

# Using a matrix-based language for prospective life models



# Predictive models in actuarial science

```
> TREE = tree((nbr>0)~ageconducteur,data=sinistres,split="gini",mincut = 1)
> age = data.frame(ageconducteur=18:90)
> y1 = predict(TREE,age)
> reg = glm((nbr>0)~bs(ageconducteur),data=sinistres,family="binomial")
> y = predict(reg,age,type="response")
```



# Working with databases

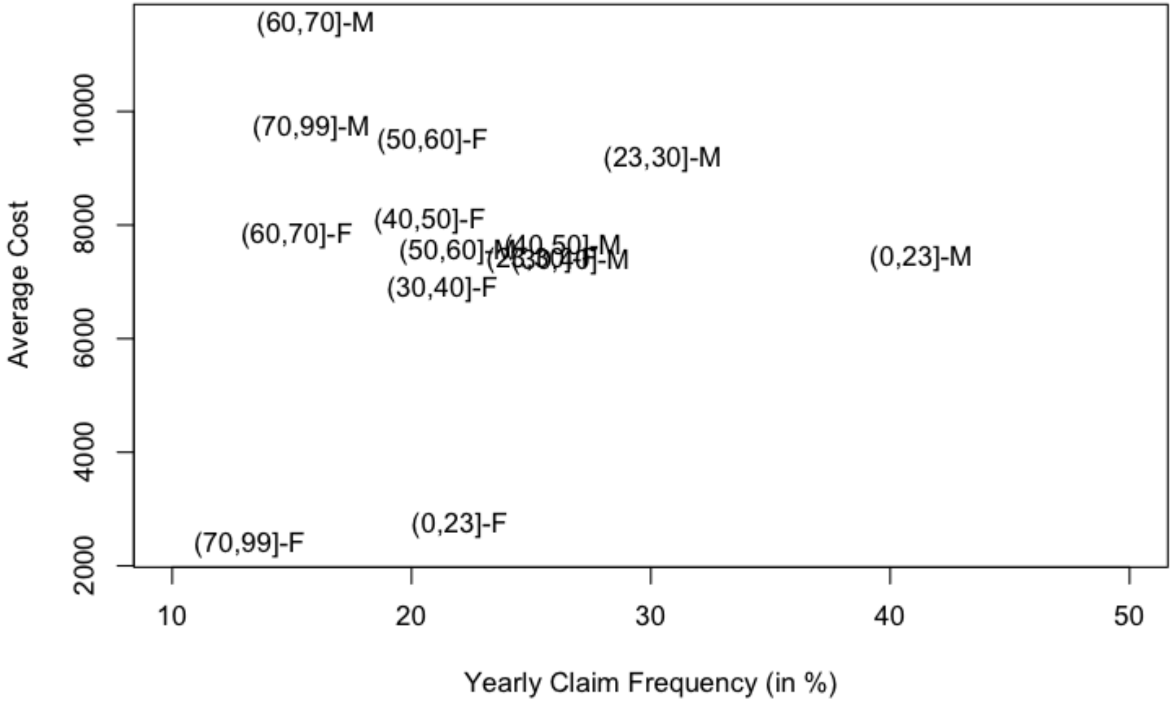
```
> baseCOUT = read.table("http://freakonometrics.free.fr/baseCOUT.csv",
+  sep=";",header=TRUE,encoding="latin1")
>  tail(baseCOUT,4)
     numeropol  debut_pol    fin_pol freq_paiement langue  type_prof alimentation type_ter
6512     87291 2002-10-16 2003-01-22       mensuel      A Professeur   Vegetarien
6513     87301 2002-10-01 2003-09-30       mensuel      A Technicien   Vegetarien
6514     87417 2002-10-24 2003-10-21       mensuel      F Technicien   Vegetalien      Semi
6515     88128 2003-01-17 2004-01-16       mensuel      F     Avocat   Vegetarien      Semi
             utilisation presence_alarme marque_voiture sexe exposition age duree_permis a
6512 Travail-occasionnel             oui           FORD    M  0.2684932  47           29
6513              Loisir             oui          HONDA    M  0.9972603  44           24
6514 Travail-occasionnel             non     VOLKSWAGEN    F  0.9917808  23            3
6515              Loisir             non           FIAT    F  0.9972603  23            4
```

# Working with databases

```
> str(baseCOUT)
'data.frame':       6515 obs. of  18 variables:
 $ numeropol       : int  6 27 27 76 76 87 105 139 145 145 ...
 $ debut_pol       : Factor w/ 2223 levels "1995-02-06","1995-03-01",..: 2 415 1030 1018
 $ fin_pol         : Factor w/ 2252 levels "1995-09-22","1995-10-04",..: 15 281 1097 1087
 $ freq_paiement   : Factor w/ 2 levels "annuel","mensuel": 1 2 2 2 2 2 2 1 2 2 ...
 $ langue          : Factor w/ 2 levels "A","F": 1 2 2 2 2 2 2 2 2 2 ...
 $ type_prof       : Factor w/ 10 levels "Actuaire","Autre",..: 10 10 10 10 10 6 10 6 10
 $ alimentation    : Factor w/ 3 levels "Carnivore","Vegetalien",..: 1 1 1 1 1 3 1 3 1 1
 $ type_territoire: Factor w/ 3 levels "Rural","Semi-urbain",..: 3 2 2 3 3 2 3 2 2 2 ...
 $ utilisation     : Factor w/ 3 levels "Loisir","Travail-occasionnel",..: 2 2 2 2 2 2 2
 $ presence_alarme: Factor w/ 2 levels "non","oui": 2 2 1 1 1 1 1 2 2 2 ...
 $ marque_voiture : Factor w/ 30 levels "ALFA ROMEO","AUDI",..: 19 11 11 9 9 29 29 29 28
 $ sexe            : Factor w/ 2 levels "F","M": 2 2 2 1 1 2 1 2 2 2 ...
 $ exposition      : num  0.995 0.244 1 1 0.997 ...
 $ age             : int  42 51 53 42 44 47 37 43 32 32 ...
 $ duree_permis    : int  21 22 24 21 23 18 16 24 12 12 ...
 $ age_vehicule    : int  19 24 16 15 15 14 20 23 16 16 ...
 $ coutsin         : num  280 814 137 609 18687 ...
```

# Working with databases

```
> cost = aggregate(coutsin~ AgeSex,mean, data=baseCOUT)
> frequency = merge(aggregate(nbsin~ AgeSex,sum, data=baseFREQ),
 + aggregate(exposition~ AgeSex,sum, data=baseFREQ))
> frequency$freq = frequency$nbsin/frequency$exposition
> base.freq.cost = merge(frequency, cost)
```



# Working with MSExcel folders

On a Windows platform, it is possible to use the `ODBConnectExcel` function of the `library(RODBC)`. The

first step is to connect the file, using

```
> sheet = "c:\\Documents and Settings\\user\\excelsheet.xls"
> connection = odbcConnectExcel(sheet)
> spreadsheet = sqlTables(connection)
```

Here, `spreadsheet$TABLE_NAME` will return sheet names. Then, we can make a SQL request

```
> query = paste("SELECT * FROM",spreadsheet$TABLE_NAME[1],sep=" ")
> result = sqlQuery(connection,query)
```

**Remark** : An alternative, available to all platform, is to use the `read.xls` function of the `library(gdata)`.

# Working with large databases

It is possible to read zipped files (even online ones)

```
> import.zip = function(file){
+ temp = tempfile()
+ download.file(file,temp);
+ read.table(unz(temp, "baseFREQ.csv"),sep=";",header=TRUE,encoding="latin1")}
> system.time(import.zip("http://freakonometrics.free.fr/baseFREQ.csv.zip"))
trying URL 'http://freakonometrics.free.fr/baseFREQ.csv.zip'
Content type 'application/zip' length 692655 bytes (676 Kb)
opened URL
==================================================
downloaded 676 Kb

   user  system elapsed
      0.762       0.029       4.578
> system.time(read.table("http://freakonometrics.free.fr/baseFREQ.csv",
+ sep=";",header=TRUE,encoding="latin1"))
   user  system elapsed
      0.591       0.072       9.277
```

# Working with large databases

It is possible to import only some parts of a large database, e.g. specific colums ...

```
> mycols = rep("NULL", 18)
> mycols[c(1,4,5,12,13,14,18)] <- NA
> baseCOUTsubC = read.table("http://freakonometrics.free.fr/baseCOUT.csv",
+  colClasses = mycols,sep=";",header=TRUE,encoding="latin1")
> head(baseCOUTsubC,4)
  numeropol freq_paiement langue sexe exposition age    coutsin
1         6        annuel      A    M  0.9945205  42   279.5839
2        27       mensuel      F    M  0.2438356  51   814.1677
3        27       mensuel      F    M  1.0000000  53   136.8634
4        76       mensuel      F    F  1.0000000  42   608.7267
```

## Working with large databases

... or specific raws in the dataset

```
> baseCOUTsubCR = read.table("http://freakonometrics.free.fr/baseCOUT.csv",
+  colClasses = mycols,sep=";",header=TRUE,encoding="latin1",nrows=100)
> tail(baseCOUTsubCR,4)
    numeropol freq_paiement langue sexe exposition age   coutsin
97       1193       mensuel      F    F  0.9972603  55  265.0621
98       1204       mensuel      F    F  0.9972603  38 9547.7267
99       1231       mensuel      F    M  1.0000000  40  442.7267
100      1245        annuel      F    F  0.6767123  48  179.1925
```

**Remark** : With `library(colbycol)` read big text files column by column.

# Working with huge databases

**Problem** : Poisson regression, with 150 million observations, 70 degrees of freedom

- Proc GENMOD in SAS (16-core Sun Server) takes around 5 hours
- installing a Hadoop cluster takes around 15 hours
- (standard) R on a 250Gb server, still running after 3 days,
- Use of `RevoScaleR` package in R, 5.7 minutes (same output as SAS)

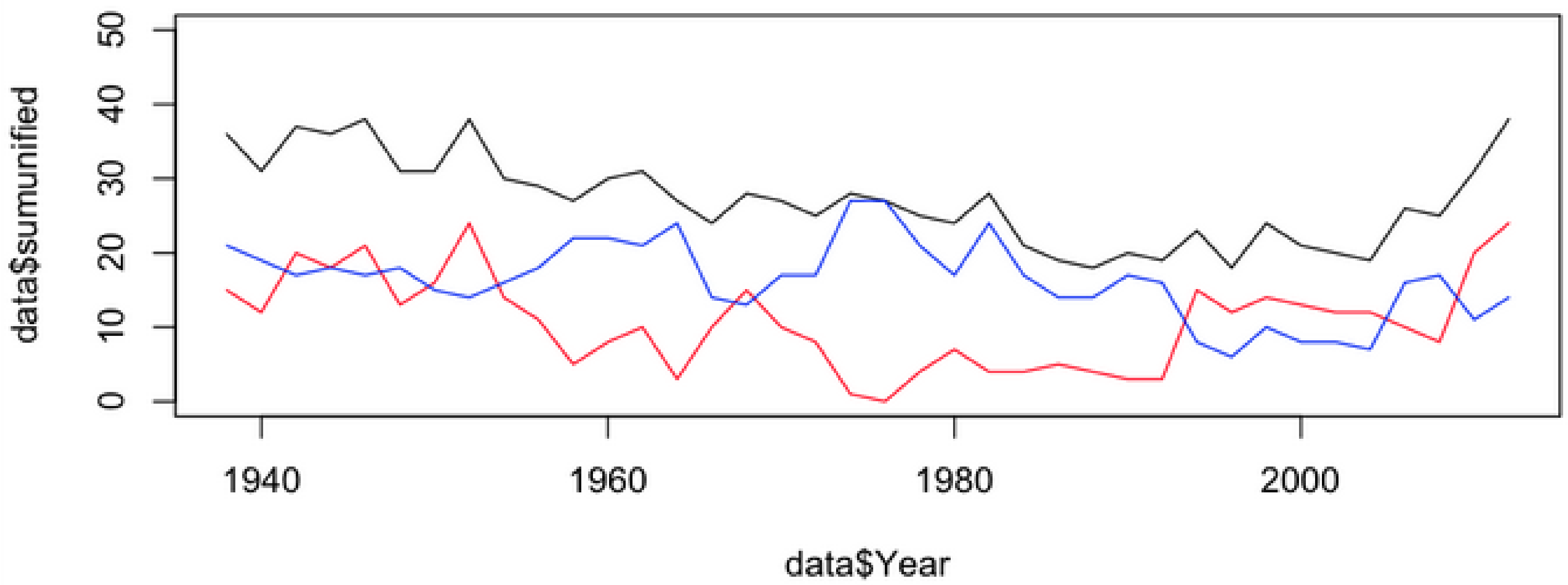
**Source** : http ://www.inside-r.org/blogs/2012/10/25/allstate-compares-sas-hadoop-and-r-big-data-insurance-models

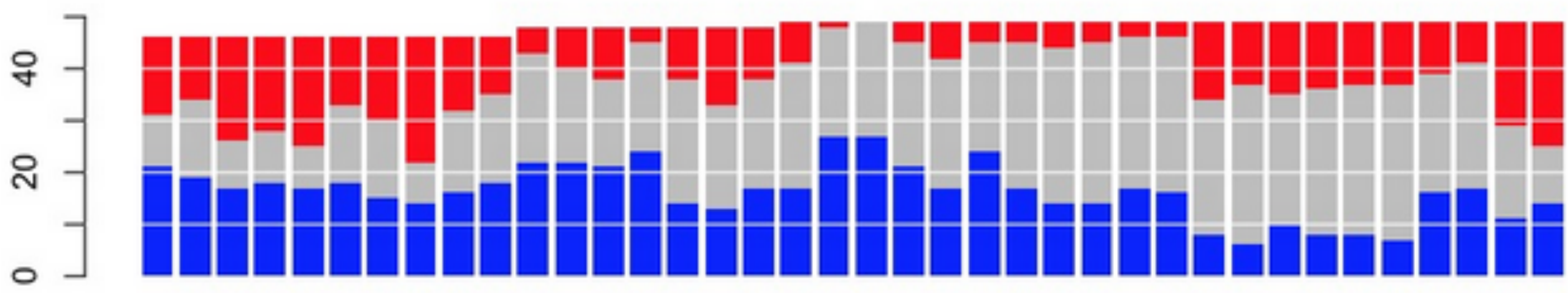
## Graphs, R and The New York Times

'*If you can picture it in your head, chances are good that you can make it work in R. R makes it easy to read data, generate lines and points, and place them where you want them. Its very flexible and super quick. When youve only got two or three hours until deadline, R can be brilliant.*" Amanda Cox, a graphics editor at the New York Times. "*R is particularly valuable in deadline situations when data is scant and time is precious.*".

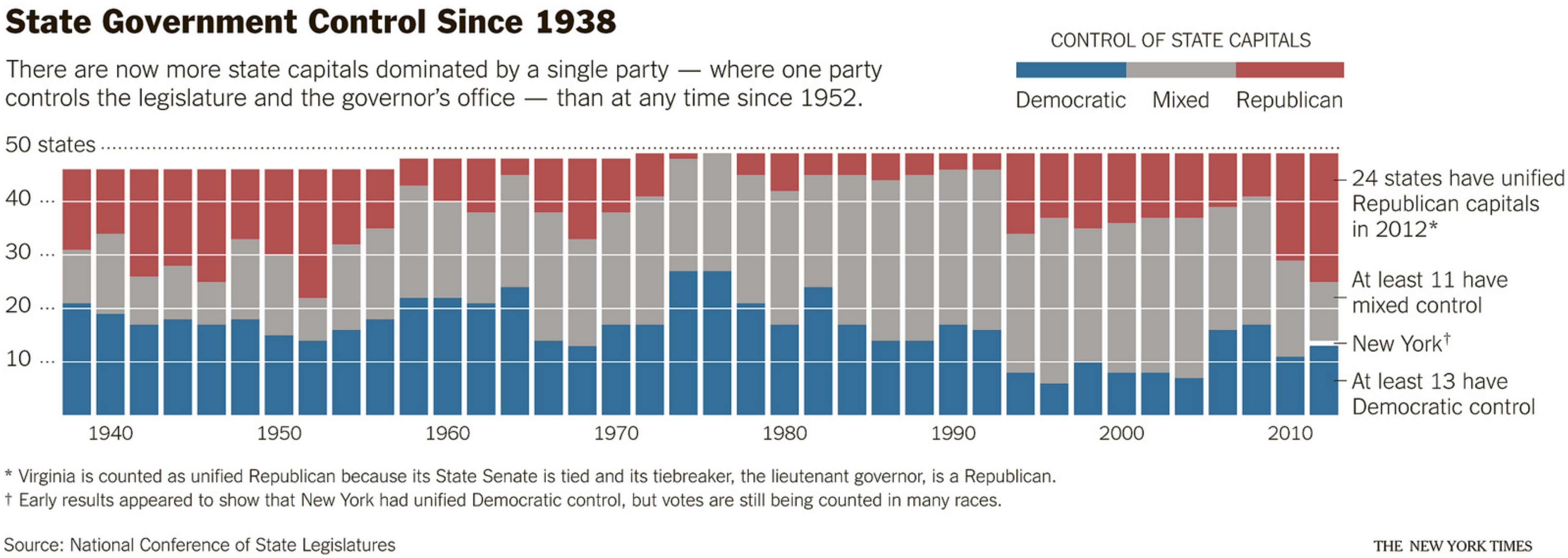
**Source** : http ://chartsnthings.tumblr.com/post/36978271916/r-tutorial-simple-charts

# Graphs, R and The New York Times

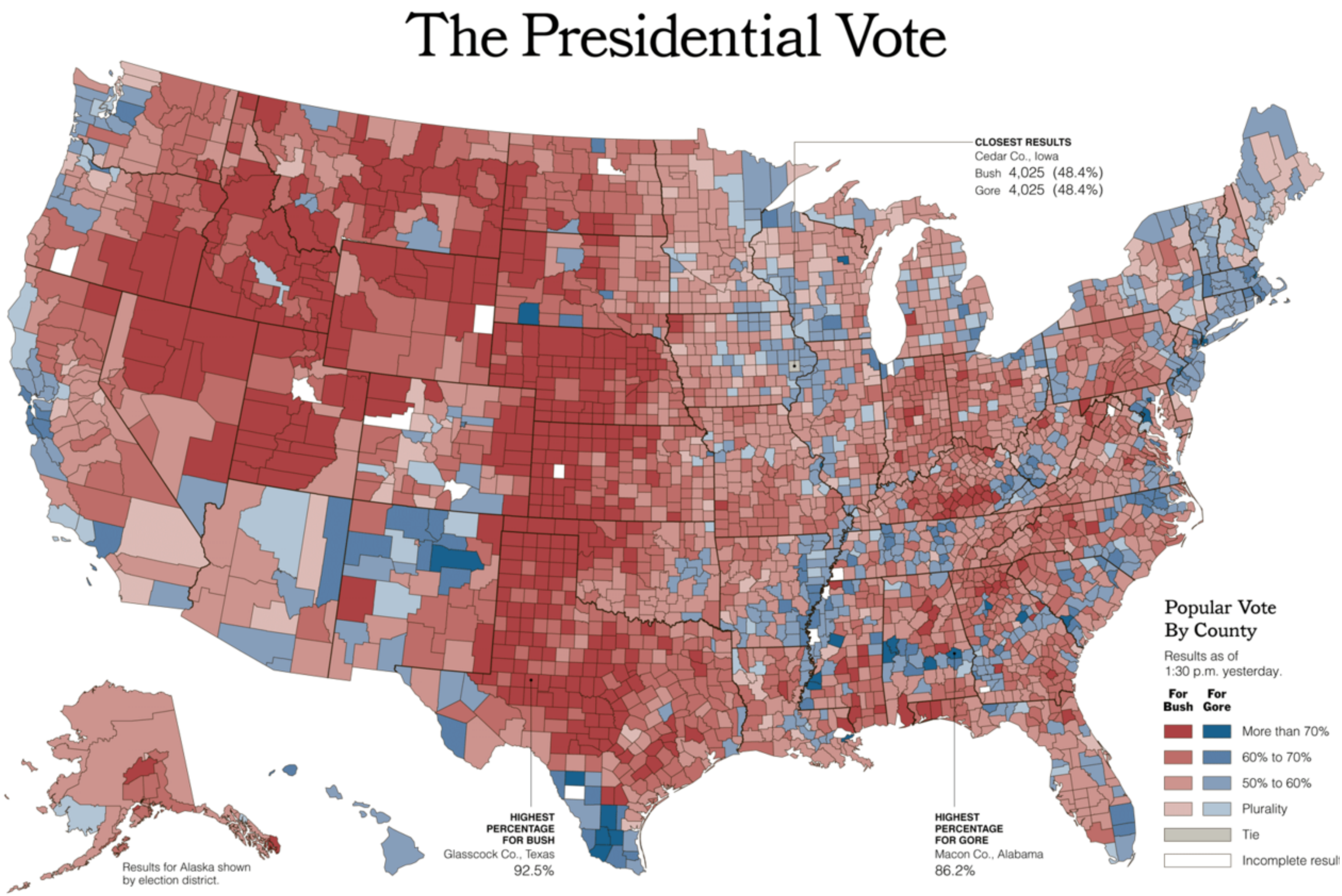


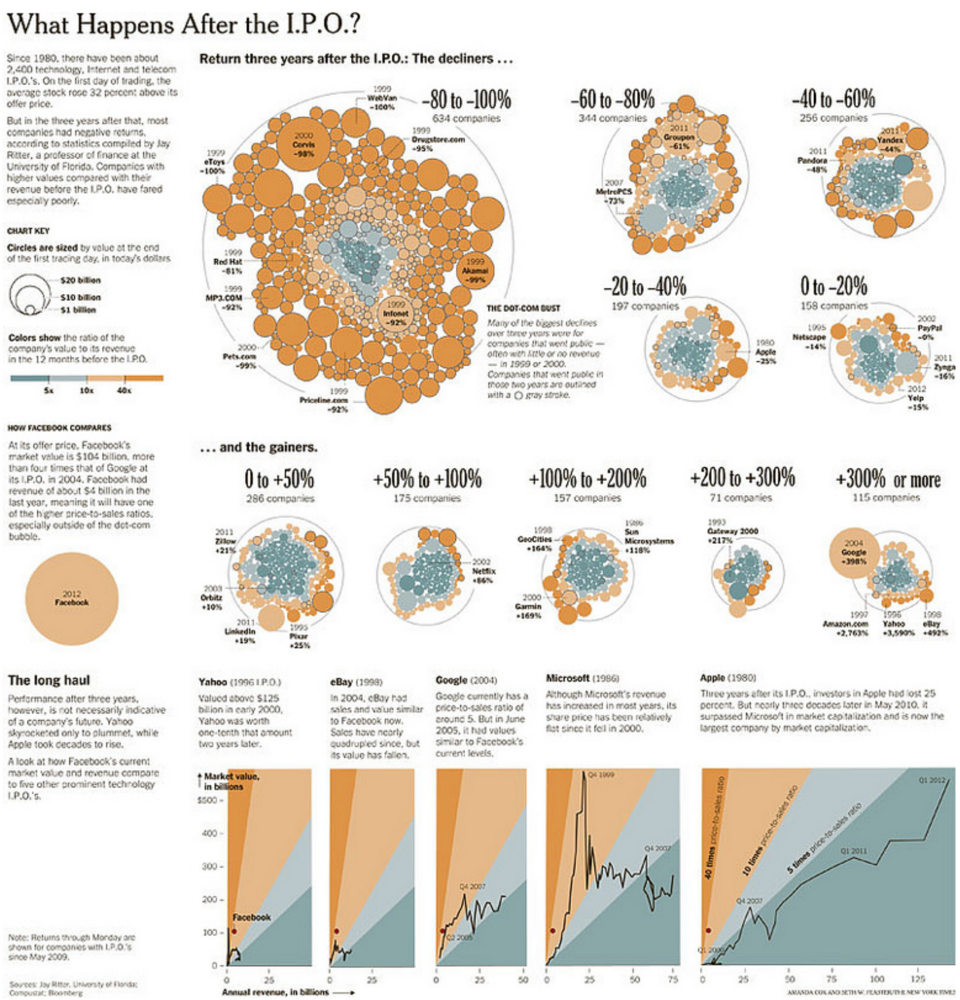
# Graphs, R and The New York Times

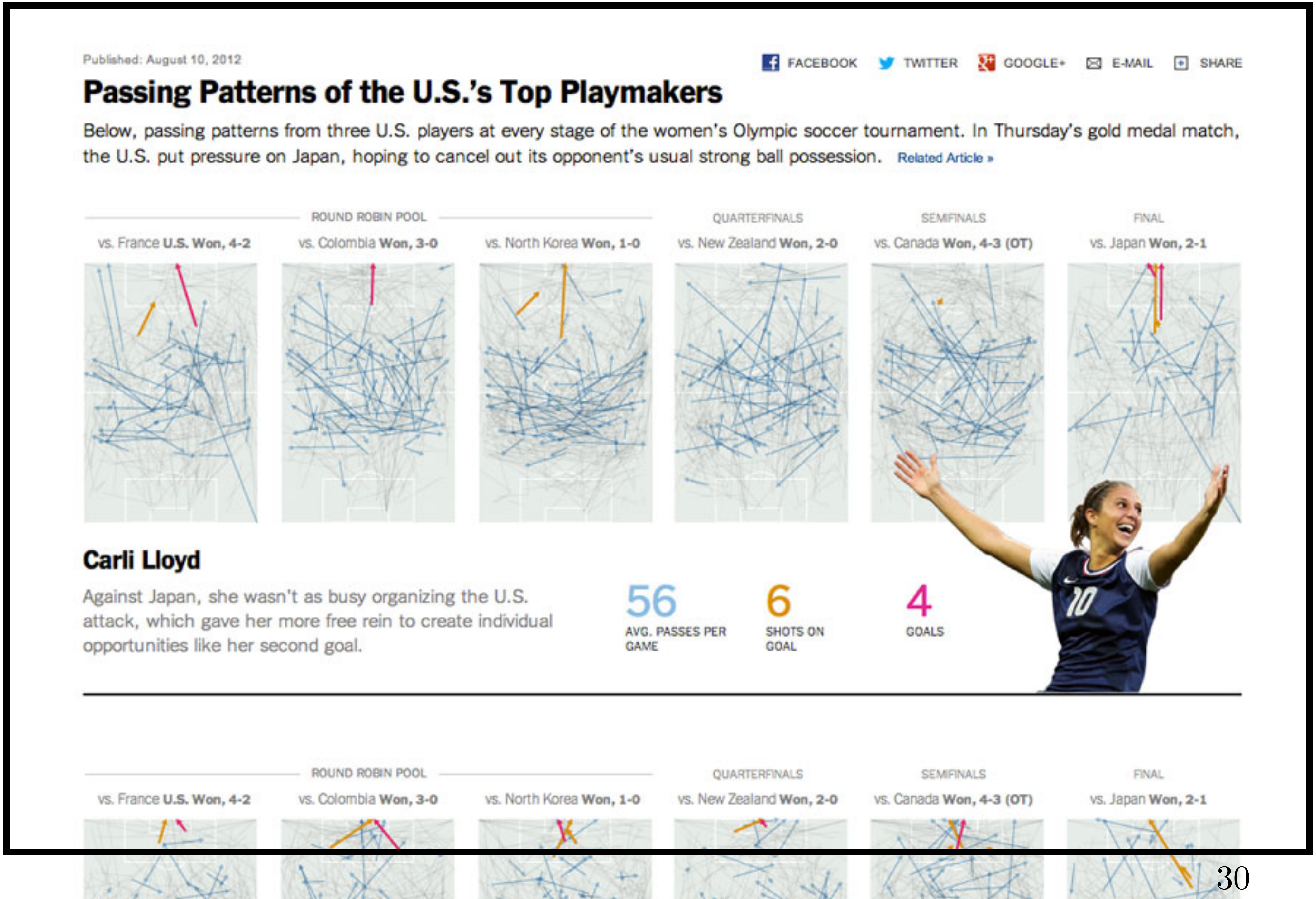


# Graphs, R and The New York Times



# Graphs, R and The New York Times




Published: August 10, 2012
FACEBOOK TWITTER GOOGLE+ E-MAIL SHARE
Passing Patterns of the U.S.'s Top Playmakers
Below, passing patterns from three U.S. players at every stage of the women's Olympic soccer tournament. In Thursday's gold medal match, the U.S. put pressure on Japan, hoping to cancel out its opponent's usual strong ball possession. Related Article »
ROUND ROBIN POOL
QUARTERFINALS
SEMIFINALS
FINAL
vs. France U.S. Won, 4-2
vs. Colombia Won, 3-0
vs. North Korea Won, 1-0
vs. New Zealand Won, 2-0
vs. Canada Won, 4-3 (OT)
vs. Japan Won, 2-1
Carli Lloyd
Against Japan, she wasn't as busy organizing the U.S. attack, which gave her more free rein to create individual opportunities like her second goal.
56
AVG. PASSES PER GAME
6
SHOTS ON GOAL
4
GOALS
ROUND ROBIN POOL
QUARTERFINALS
SEMIFINALS
FINAL
vs. France U.S. Won, 4-2
vs. Colombia Won, 3-0
vs. North Korea Won, 1-0
vs. New Zealand Won, 2-0
vs. Canada Won, 4-3 (OT)
vs. Japan Won, 2-1

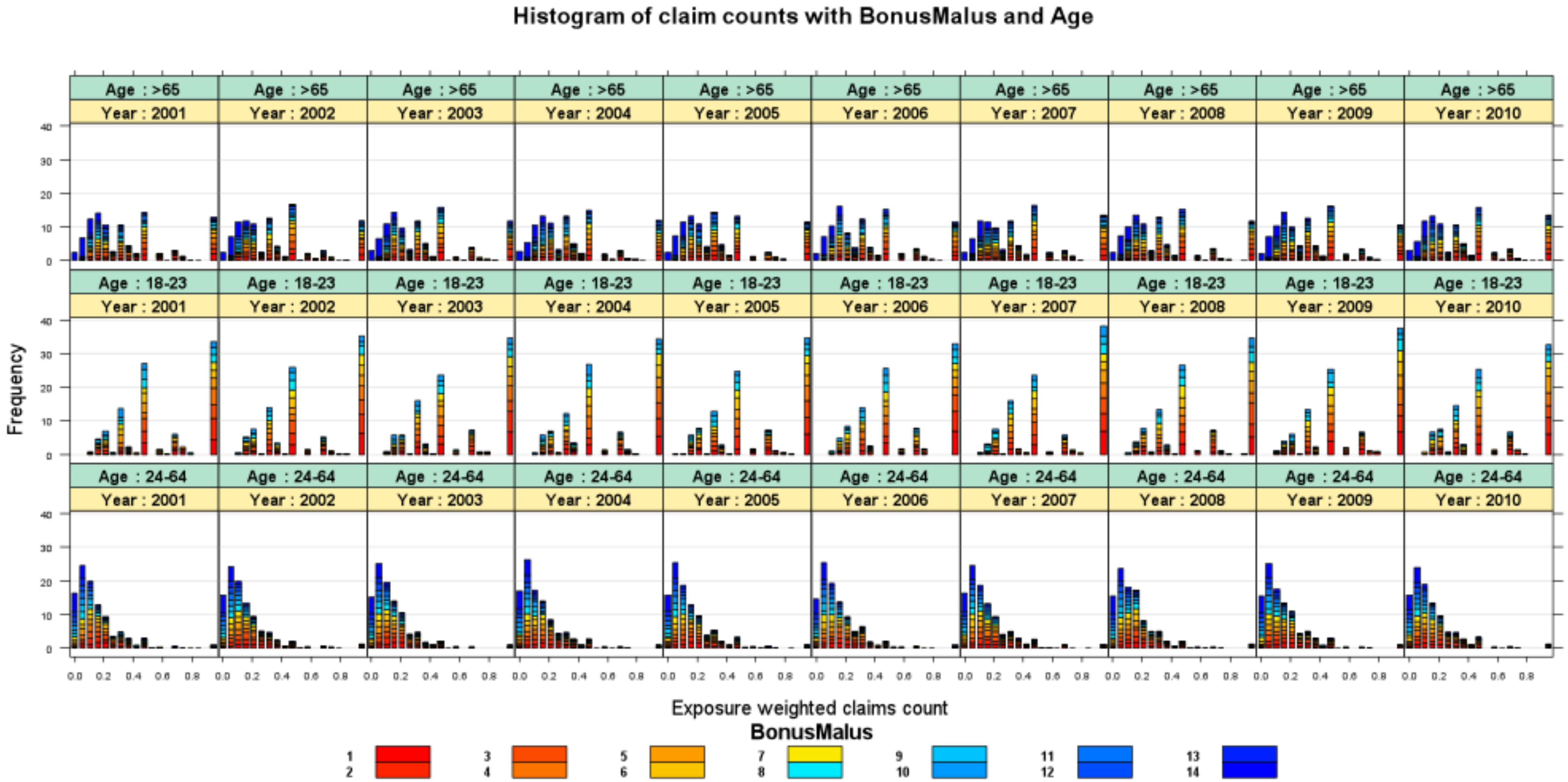
## Graphs in actuarial communication

"*Its not just about producing graphics for publication. Its about playing around and making a bunch of graphics that help you explore your data. This kind of graphical analysis is a really useful way to help you understand what youre dealing with, because if you cant see it, you cant really understand it. But when you start graphing it out, you can really see what youve got.*" Peter Aldhous, San Francisco bureau chief of New Scientist magazine.

"*The commercial insurance underwriting process was rigorous but also quite subjective and based on intuition. R enables us to communicate our analytic results in appealing and innovative ways to non-technical audiences through rapid development lifecycles. R helps us show our clients how they can improve their processes and effectiveness by enabling our consultants to conduct analyses efficiently*". John Lucker, team of advanced analytics professionals at Deloitte Consulting Principal.

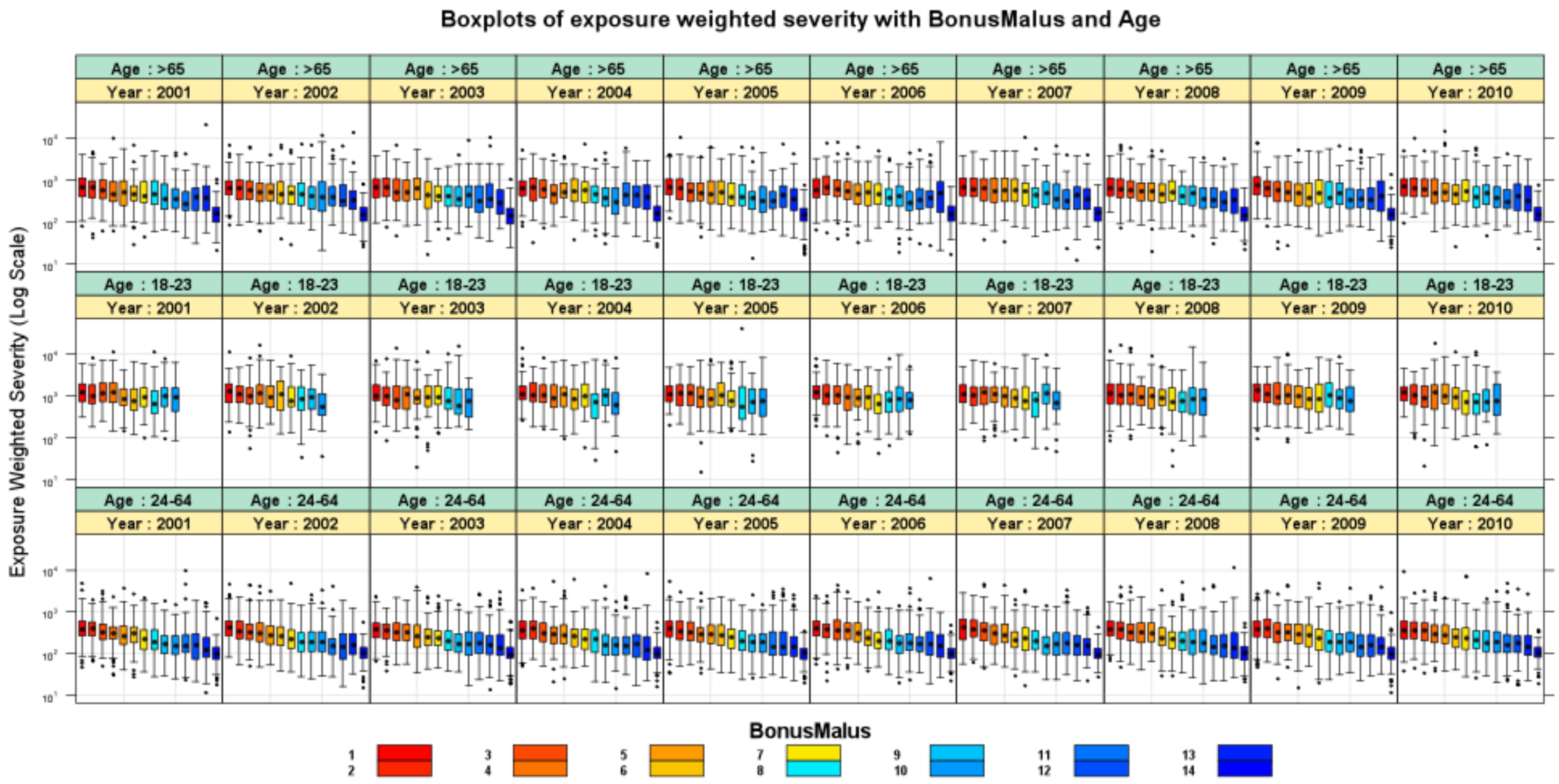
see also Gelman (2011).

# Graphs in actuarial communication




**Source** : http ://www.londonr.org/Presentations/RInActuarialAnalysis.pptx, data from Kaas et al. (2001)

# Graphs in actuarial communication




**Source** : http ://www.londonr.org/Presentations/RInActuarialAnalysis.pptx, data from Kaas et al. (2001)

## Reproducibility issues

"*Commonly research involving scientific computations are reproducible in principle, but not in practice. The published documents are merely the advertisement of scholarship whereas the computer programs, input data, parameter values, etc. embody the scholarship itself. Consequently authors are usually unable to reproduce their own work after a few months or years.*"

Schwab et al. (2000)

"*The goal of reproducible research is to tie specific instructions to data analysis and experimental data so that scholarship can be recreated, better understood and verified.* "

**Source** : http ://cran.open-source-solution.org/web/views/ReproducibleResearch.html

# Reproducibility issues

## Repeatability of published microarray gene expression analyses

John P A Ioannidis[1–3], David B Allison[4], Catherine A Ball[5], Issa Coulibaly[4], Xiangqin Cui[4], Aedín C Culhane[6,7], Mario Falchi[8,9], Cesare Furlanello[10], Laurence Game[11], Giuseppe Jurman[10], Jon Mangion[11], Tapan Mehta[4], Michael Nitzberg[5], Grier P Page[4,12], Enrico Petretto[11,13] & Vera van Noort[14]

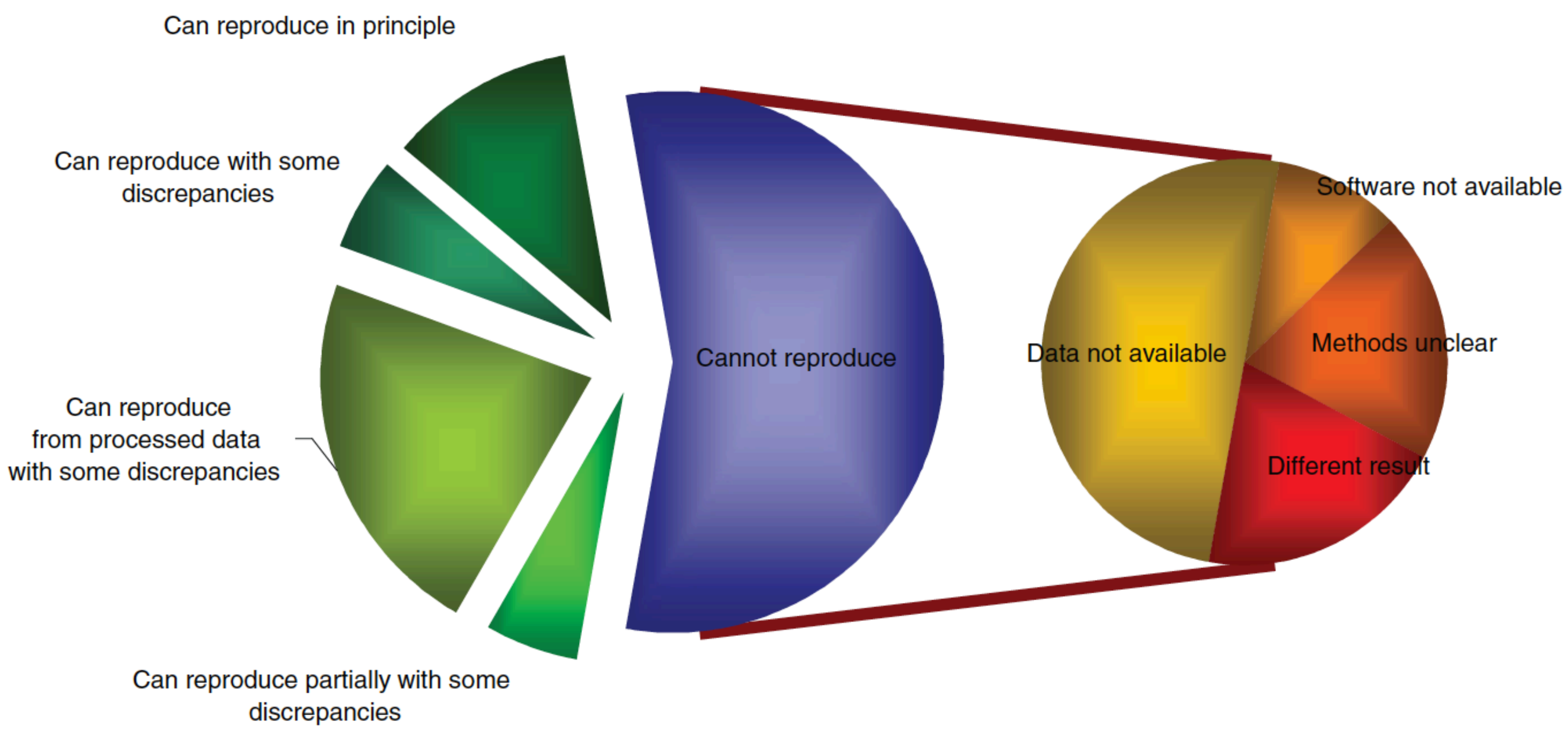



**Figure 1** Summary of the efforts to replicate the published analyses.

# R versus other (statistical) softwares

"*The power of the language R lies with its functions for statistical modelling, data analysis and graphics ; its ability to read and write data from various data sources ; as well as the opportunity to embed R in excel or other languages like VBA. In the way SAS is good for data manipulations, R is superior for modelling and graphical output*"

**Source** : http ://www.actuaries.org.uk/system/files/documents/pdf/actuarial-toolkit.pdf

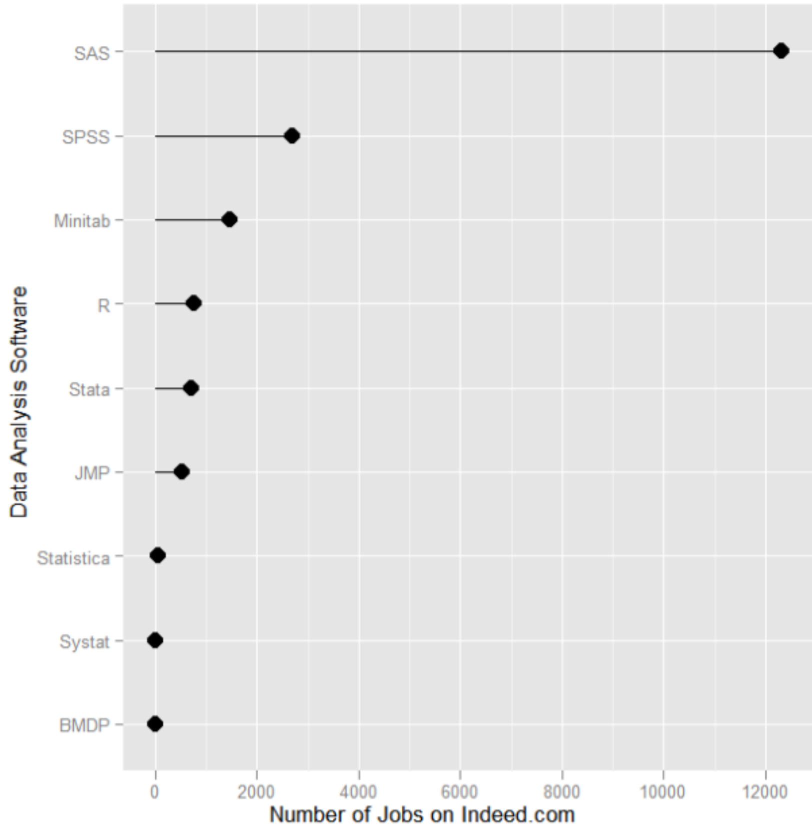
## R versus other (statistical) softwares

| | |
|---|---|
| SAS | PC : $ 6,000 per seat - server : $28,000 per processor |
| Matlab | $ 2,150 (*commercial*) |
| Excel | |
| SPSS | $ 4,975 |
| EViews | $ 1,075 (*commercial*) |
| RATS | $ 500 |
| Gauss | - |
| Stata | $ 1,195 (*commercial*) |
| S-Plus | $ 2,399 per year |

**Source** : http ://en.wikipedia.org/wiki/Comparison_of_statistical_packages

# R in the non-academic world

What software skills are employers seeking ?




**Source :** http ://r4stats.com/articles/popularity/

## R in the insurance industry

From 2011, Asia Capital Reinsurance Group (ACR) uses R to Solve Big Data Challenges

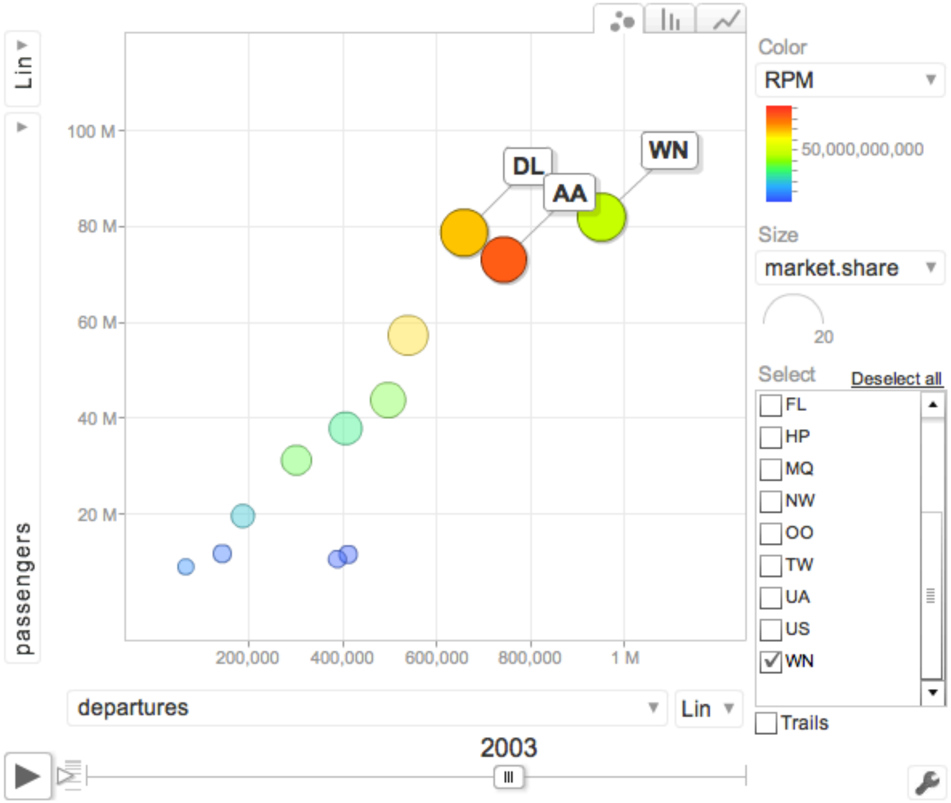
**Source** : http ://www.reuters.com/article/2011/07/21/idUS133061+21-Jul-2011+BW20110721

From 2011, Lloyd's uses motion charts created with R to provide analysis to investors.

**Source** : http ://blog.revolutionanalytics.com/2011/07/r-visualizes-lloyds.html

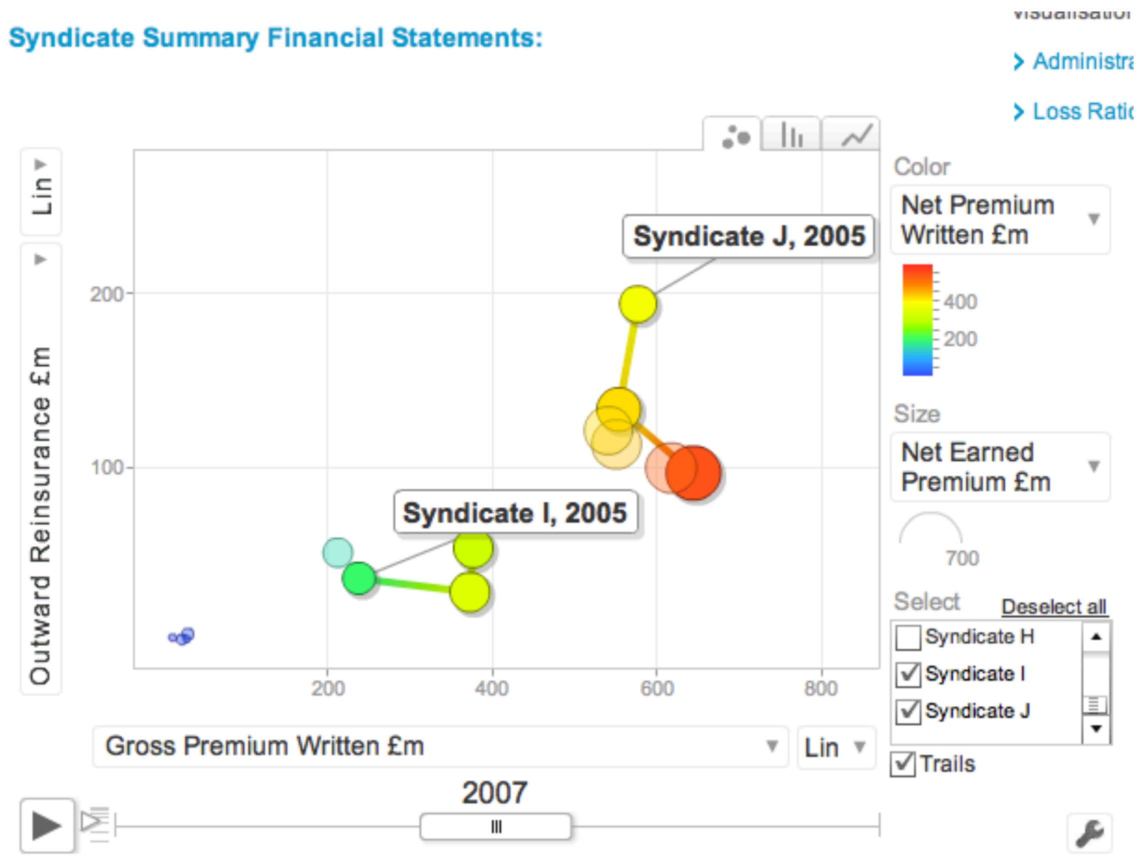

**Source** : http ://www.revolutionanalytics.com/what-is-open-source-r/companies-using-r.php

# R in the insurance industry



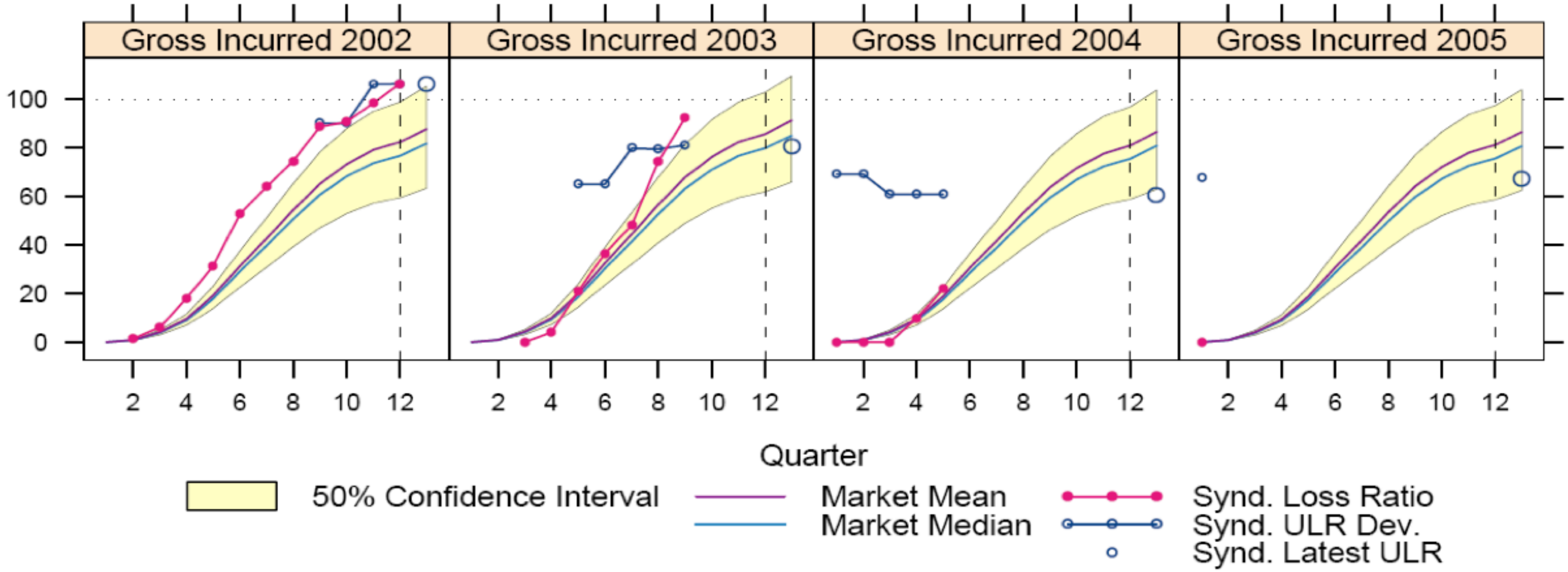

**Source** : http ://jeffreybreen.wordpress.com/2011/07/14/r-one-liners-googlevis/

# R in the insurance industry




**Source** : http ://jeffreybreen.wordpress.com/2011/07/14/r-one-liners-googlevis/

# R in the insurance industry




**Source** : http ://lamages.blogspot.ca/2011/09/r-and-insurance.html, i.e. Markus Gesmann's blog

# Popularity of R versus other languages

as at January 2013,

**Transparent Language Popularity**

| | | |
|---|---|---|
| 1. | C | 17.780% |
| 2. | Java | 15.031% |
| 8. | Python | 4.409% |
| 12. | R | 1.183% |
| 22. | Matlab | 0.627% |
| 27. | SAS | 0.530% |

**Source** : http ://lang-index.sourceforge.net/

**TIOBE Programming Community Index**

| | | |
|---|---|---|
| 1. | C | 17.855% |
| 2. | Java | 17.417% |
| 7. | Visual Basic | 4.749% |
| 8. | Python | 4.749% |
| 17. | Matlab | 0.641% |
| 23. | SAS | 0.571% |
| 26. | R | 0.444% |

**Source** : http ://www.tiobe.com/index.php/

# Popularity of R versus other languages

as at January 2013, tags

stackoverflow

| | |
|---|---|
| C++ | 399,323 |
| Java | 348,418 |
| Python | 154,647 |
| R | 21,818 |
| Matlab | 14,580 |
| SAS | 899 |

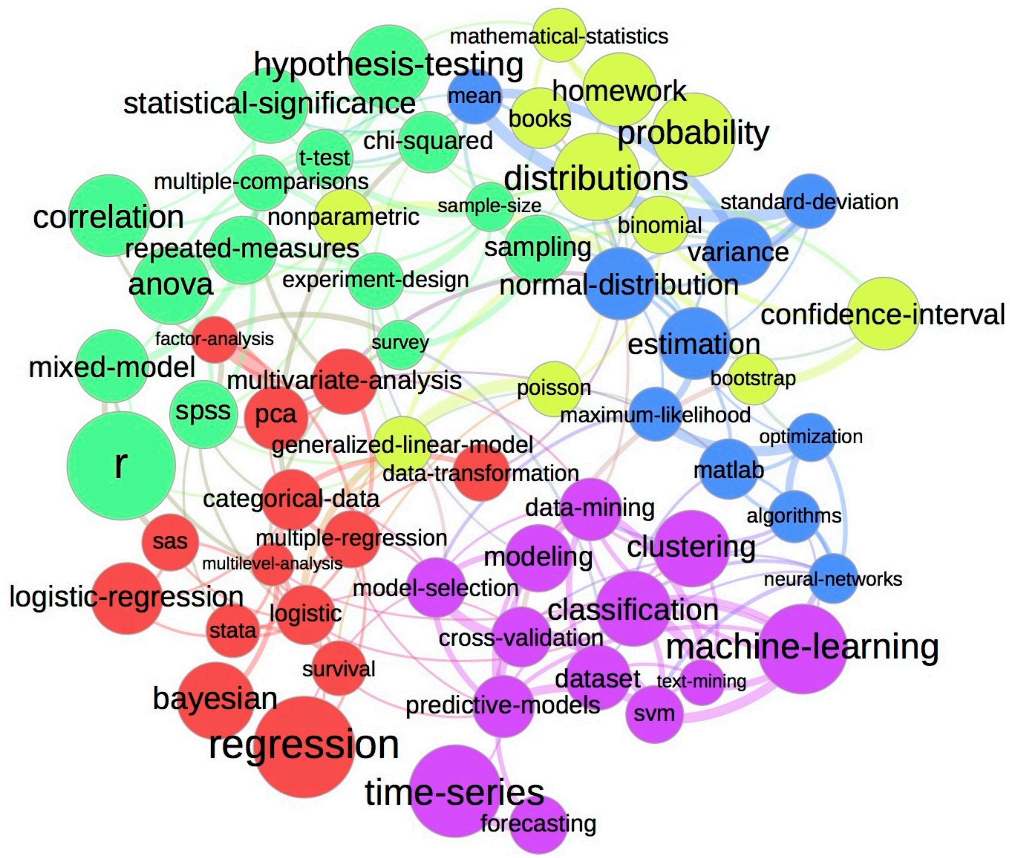
**Source** : http ://stackoverflow.com/tags ?tab=popular

Cross Validated

| | |
|---|---|
| R | 3,008 |
| Matlab | 210 |
| SAS | 187 |
| Stata | 153 |
| Java | 26 |

**Source** : http ://www.tiobe.com/index.php/

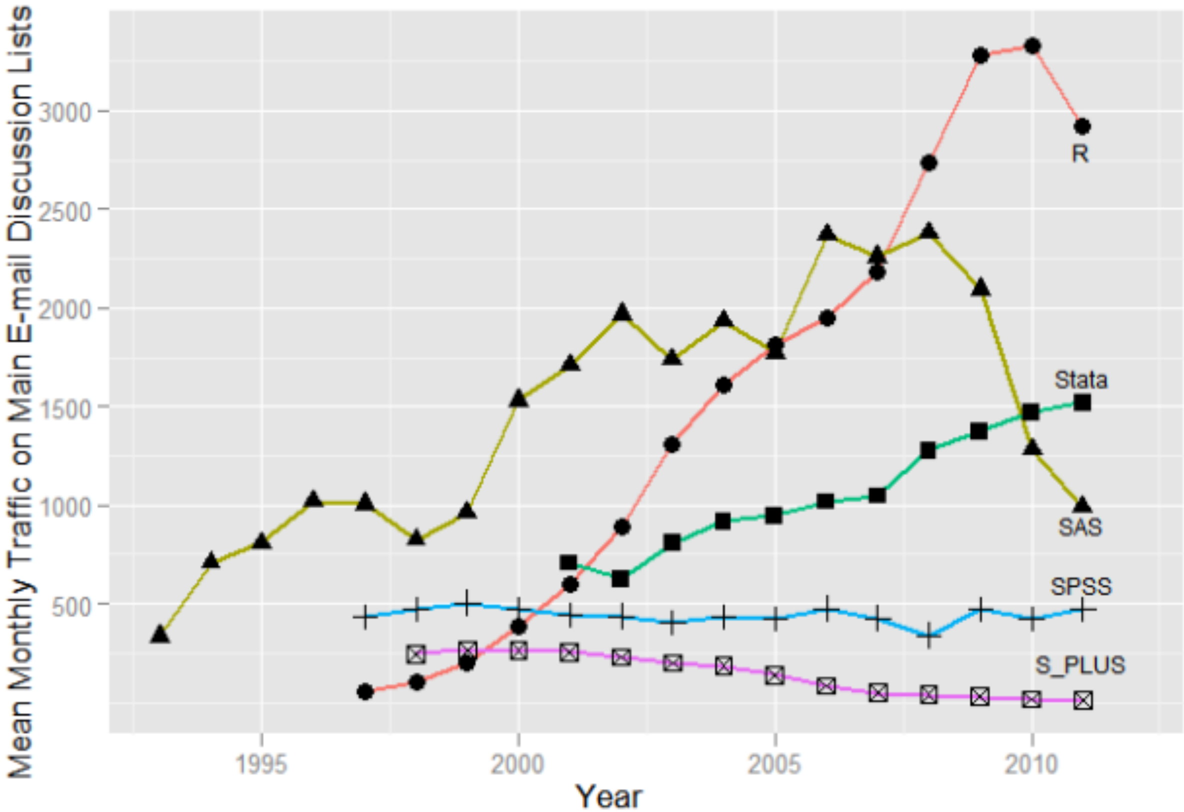
# R versus other statistical languages




**Source** : http ://meta.stats.stackexchange.com/questions/1467/tag-map-for-crossvalidated

# R versus other statistical languages

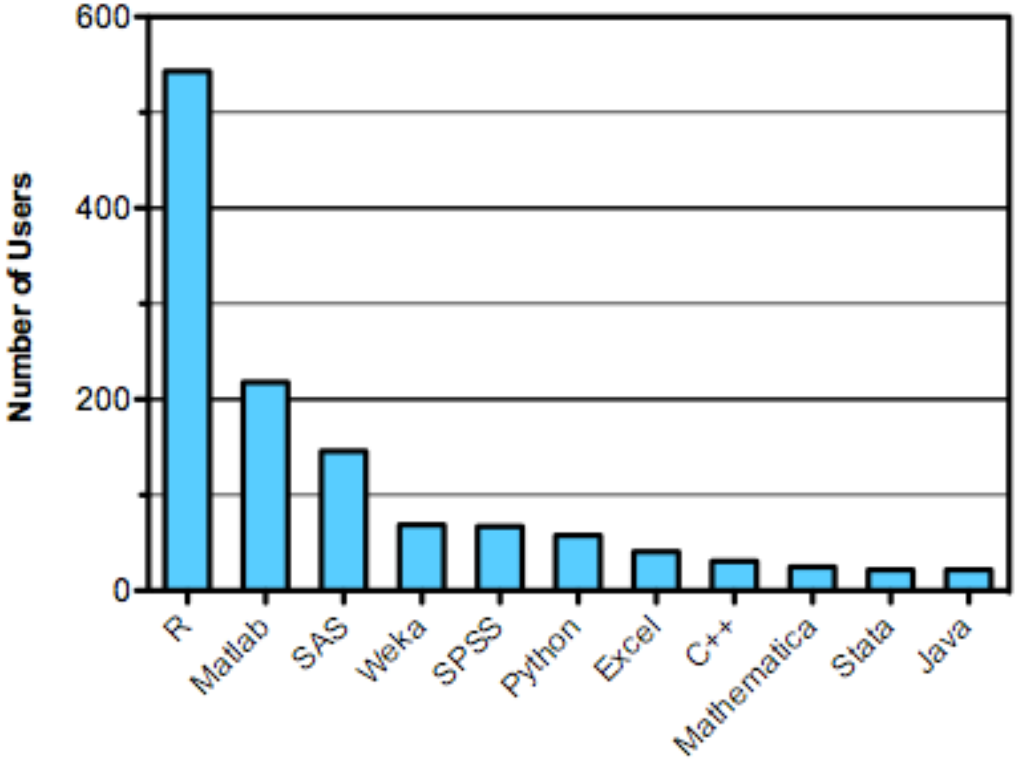
Plot of listserv discussion traffic by year (through December 31, 2011)




**Source** : http ://r4stats.com/articles/popularity/

# R versus other statistical languages

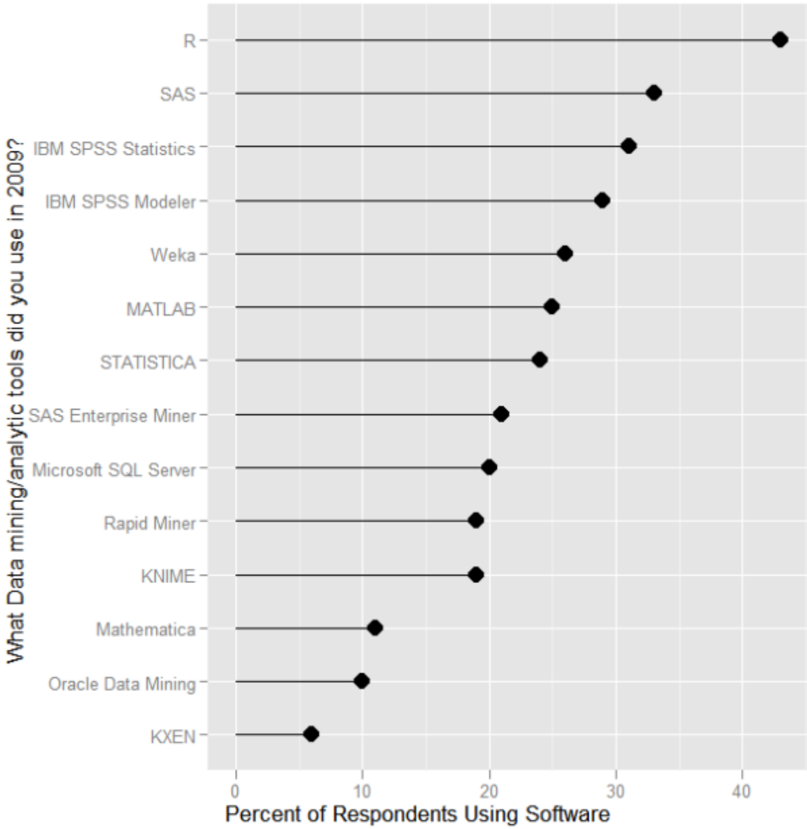
Software used by competitors on Kaggle




**Source** : http ://r4stats.com/articles/popularity/ and http ://www.kaggle.com/wiki/Software

# R versus other statistical languages

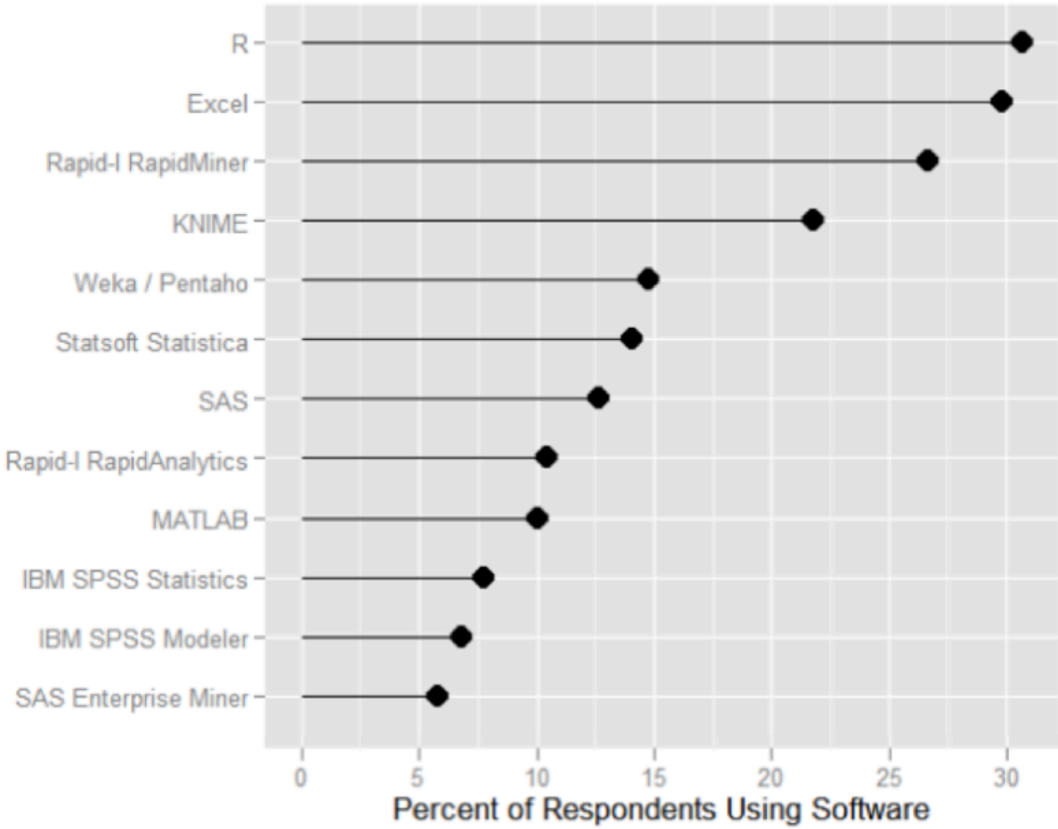
Data mining/analytic tools reported in use on Rexer Analytics survey, 2009.




**Source** : http ://r4stats.com/articles/popularity/

# R versus other statistical languages

"What programming languages you used for data analysis in the past 12 months ?"




**Source** : http ://r4stats.com/articles/popularity/

# R versus other statistical languages

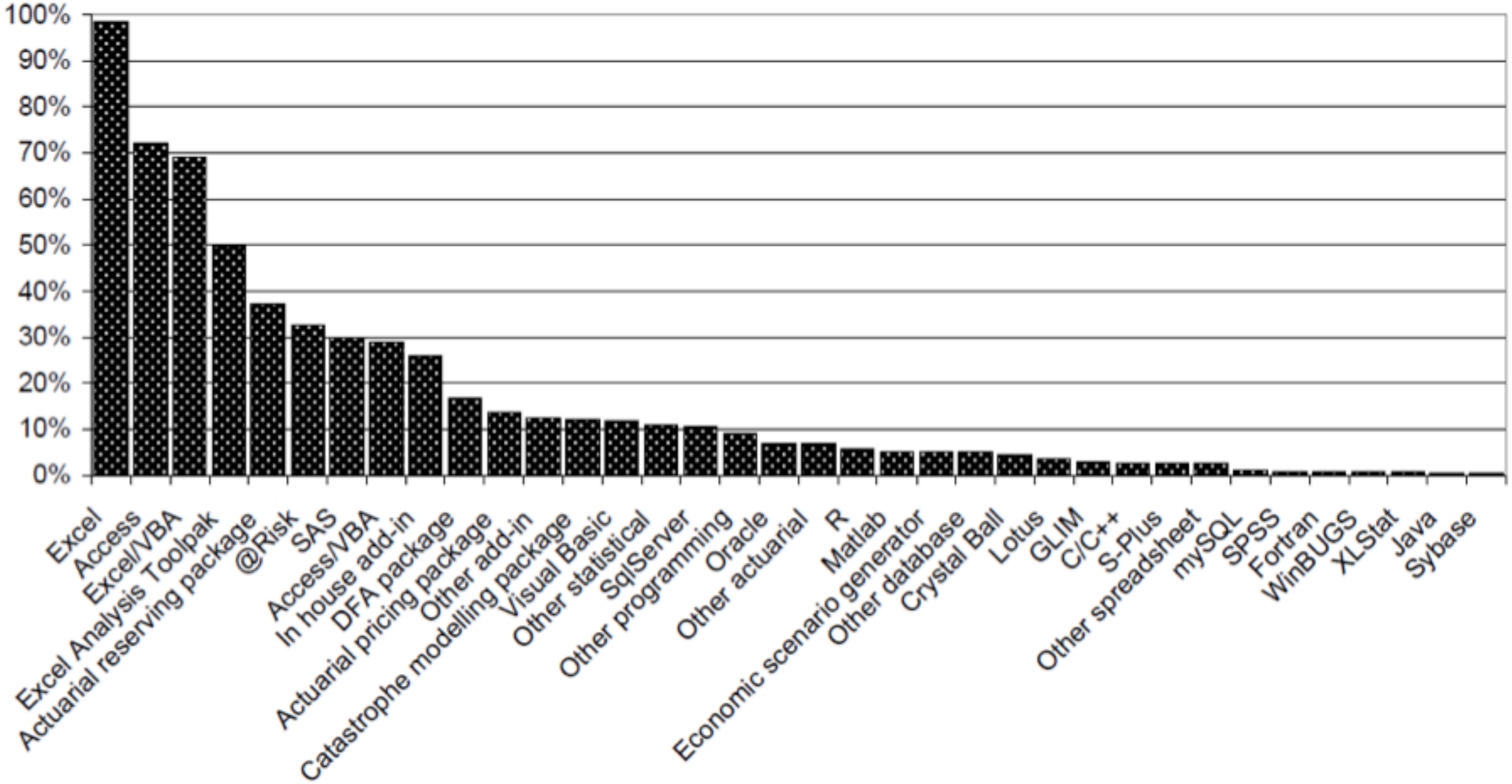
"What programming languages you used for data analysis ?"

| **What programming languages you used for data mining / data analysis in the past 12 months?** [570 voters] | |
|---|---|
| R (257) | 45% |
| SQL (184) | 32% |
| Python (140) | 25% |
| Java (139) | 24% |
| SAS (121) | 21% |
| MATLAB (83) | 15% |
| C/C++ (73) | 13% |
| Unix shell/awk/gawk/sed (59) | 10% |
| Perl (45) | 7.9% |
| Hadoop/Pig/Hive (35) | 6.1% |
| Lisp (4) | 0.7% |
| Other (70) | 12.0% |
| None (7) | 1.2% |

**Source** : http ://r4stats.com/articles/popularity/

# R versus other ‘statistical’ softwares, for actuaries

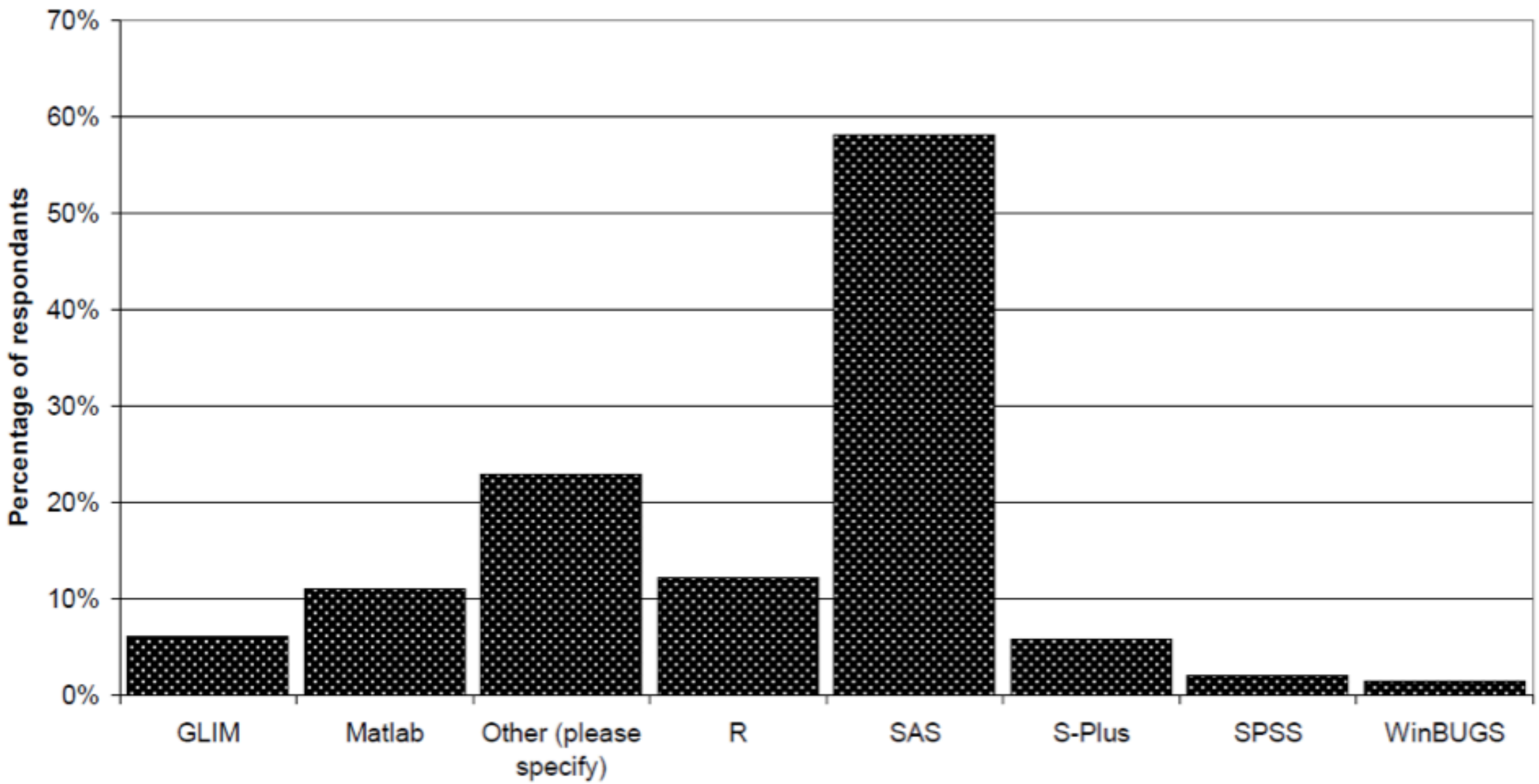
Softwares used by UK actuaries, and CAS actuaries




**Source :** : http ://www.palisade.com/downloads/pdf/Pryor.pdf

# R versus other statistical softwares, for actuaries

Statistical softwares used by UK actuaries, and CAS actuaries




**Source :** : http ://www.palisade.com/downloads/pdf/Pryor.pdf

## The R community, forums, blogs, books

"*I cant think of any programming language that has such an incredible community of users. If you have a question, you can get it answered quickly by leaders in the field. That means very little downtime.*" Mike King, Quantitative Analyst, Bank of America.

"*The most powerful reason for using R is the community*" Glenn Meyers, in the Actuarial Review.

"*The great beauty of R is that you can modify it to do all sorts of things. And you have a lot of prepackaged stuff thats already available, so youre standing on the shoulders of giants*", Hal Varian, chief economist at Google.

**Source** : : http ://www.nytimes.com/2009/01/07/technology/business-computing/07program.html

R news and tutorials contributed by 425 R bloggers (as at Jan. 2013)

**Source** : : http ://www.r-bloggers.com/

## R versus other softwares used in actuarial science

SAS is a commercial software developed by the SAS Institute ;
- it includes well-validated statistical algorithms,
- licensing is expensive
- new statistical methods might be incorporated only after a significant lag
- it includes data management tools, and is undertaken using row by row (observation-level) operations

(see Kleinman & Horton (2010) for more details)

Matlab better programming environment (e.g. better documentation, better debuggers, better object browser), can be without doing any programming. It is a commercial software, there are more integrated add-ons and more support (but one has to pay for it). R is stronger for statistic.

To define a vector, the common syntax is `v=[0,1,2]`, then we use `v(2)`.

Consider the smoothing function in Matlab,

```
[f,df,gcv,sse,penmat,y2cmat] = smooth_basis(argvals, y, fdparobj)
```

(see chapter 2 in Ramsay, Hooker & Graves (2009) for more details)

R is a free, open-source software, developed by R development core team, and people from the R community.

– programming environment for data analysis
– statisticians often release R functions to implement their work concurrently with publication
– R is a vector-based language, where columns (variables) are manipulated

To define a vector, the common syntax is `v=c(0,1,2)`, then we use `v[2]` Consider the smoothing function in Matlab,

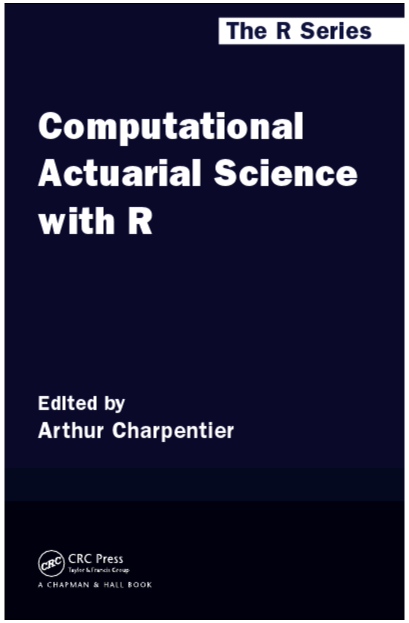
```
smoothlist = smooth.basis(argvals, y, fdparobj)
```

i.e. the output is a single object (a list, the counterpart of struct objects in Matlab)

## Take-home message

*"The best thing about R is that it was developed by statisticians.*
*The worst thing about R is that it was developed by statisticians."*

Bo Cowgill, Google




To go further...

forthcoming book on Computational Actuarial Science