WOW, DEEP LEARNING IS SOOO...
FUCKING USELESS
memegenerator.net

Figure:

Statistics with Julia

“.....Can he do it on a cold, wet Wednesday night in Stoke?”

Statistics with Julia

1st Year	Exploratory Data Analysis, Summary Statistics, Probability, Graphical Methods
2nd Year	Hypothesis Testing, Confidence Intervals, Probability Distributions, Linear Models
3rd Year	ANOVA and Experimental Design, Residuals, Chi Squared, Stepwise Regression
4th Year	PCA, Clustering, Logistic Regression

Statistics with Julia

What is like to teach statistics vs **What it should be like**

The Future according to Kevin

- Remove Pen and Paper Calculations
  (Keep a few good ones)
- Sort out Bad Instructional Design
  (time is short...dont waste time with stupid crap)
- Why the flip is this still an exam question? Is this still the 30s?
- (The t-test is actually fairly robust to non-normality).
- e.g. Sum of Squares Identities in Experimental Design..**BY HAND...F.R.O!!!**

Statistics with Julia

Writing Stats Exams!!

- Copy and Paste questions some past papers
- change a few numbers here and there
- Transforms weights of dogs into heights of cats

$$X \sim 1000, 25^2$$

- Why fix that equation? too much like work?

Exam papers take time.... Hey, you got better things to do!!!

Statistics with Julia

Hypothesis Testing it a bit like a trial

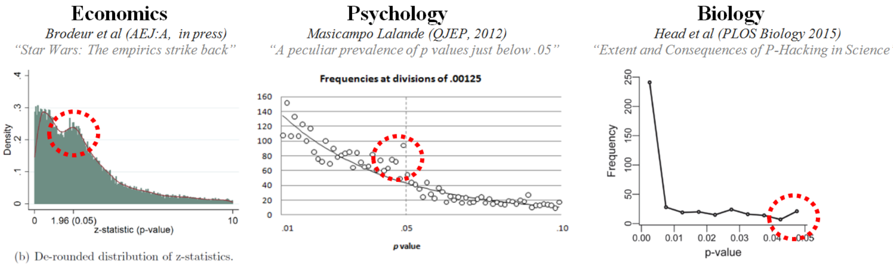
Ho : Innocent

H1 : Guilty

Got enough evidence to convict? Reasonable doubt

Statistics with Julia

- Put in more p-values...but learning to critique the analysss proplerly
- Tell them about P-hacking
- and anyway...What exactly is a confidence interval (for mean?)



Economics

Brodeur et al (AEJ:A, in press)

"Star Wars: The empirics strike back"

Psychology

Masicampo Lalande (QJEP, 2012)

"A peculiar prevalence of p values just below .05"

Biology

Head et al (PLOS Biology 2015)

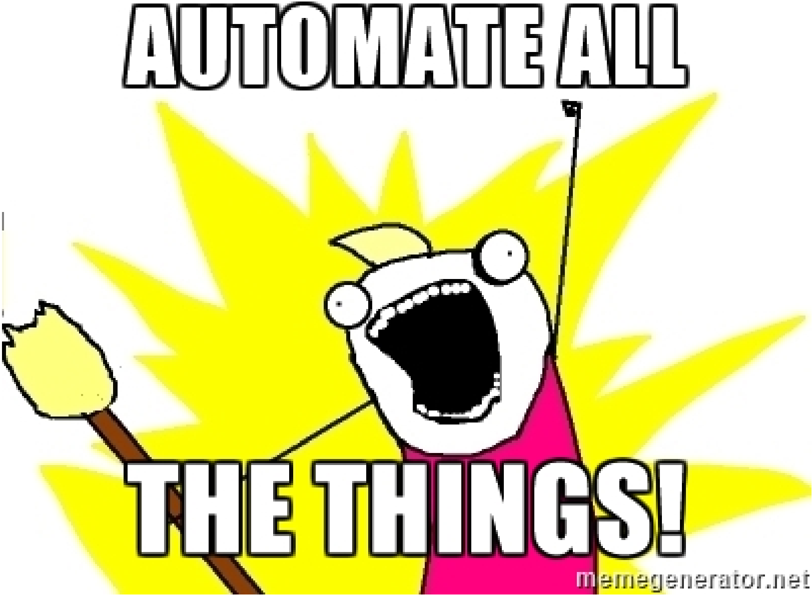
"Extent and Consequences of P-Hacking in Science

(b) De-rounded distribution of z-statistics.

Figure:

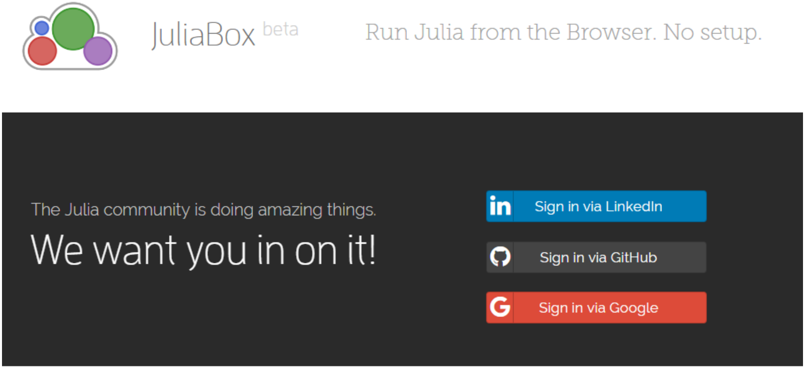
Statistics with Julia

- Never omit "Type I" Error and "Type II" Error
- HT is not about what is true or false it is above what you can prove (back up with a sufficient amount of evidence)
- You'd be surprised about how many people dont know that.


AUTOMATE ALL
THE THINGS!
memegenerator.net

Open source, interactive data science and scientific computing

Figure:



JuliaBox beta

Run Julia from the Browser. No setup.

The Julia community is doing amazing things.

We want you in on it!

Figure:

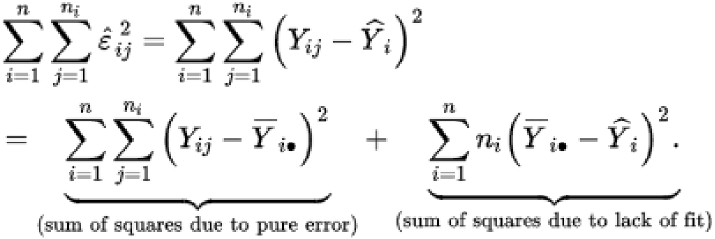
Statistics with Julia

Surely students could handle some code?

- `sample()`
- `mean()`
- `t.test()`

They dont have to like it, but they would prefer having to do some basic computing as opposed to.....

Statistics with Julia



$$\sum_{i=1}^{n}\sum_{j=1}^{n_i}\hat{\varepsilon}_{ij}^{\,2} = \sum_{i=1}^{n}\sum_{j=1}^{n_i}\left(Y_{ij}-\widehat{Y}_i\right)^2$$

$$= \underbrace{\sum_{i=1}^{n}\sum_{j=1}^{n_i}\left(Y_{ij}-\overline{Y}_{i\bullet}\right)^2}_{\text{(sum of squares due to pure error)}} + \underbrace{\sum_{i=1}^{n} n_i\left(\overline{Y}_{i\bullet}-\widehat{Y}_i\right)^2.}_{\text{(sum of squares due to lack of fit)}}$$

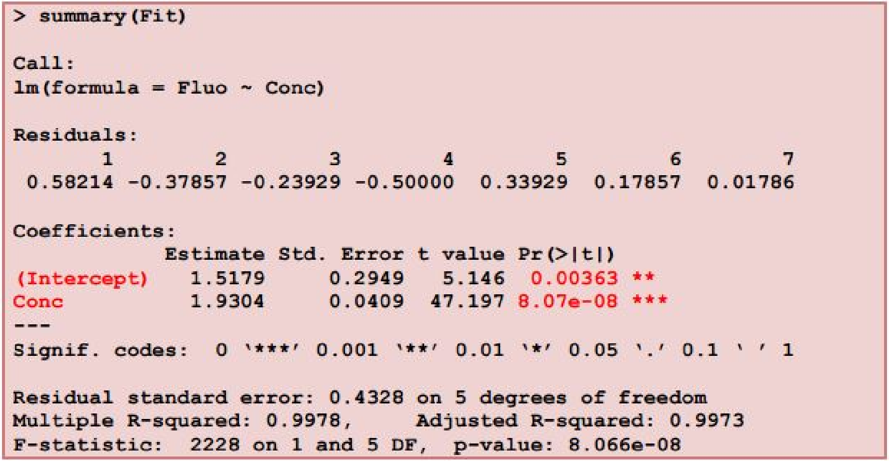
2 Hours of my life wasted!!!
(Although some are worth keeping)

Statistics with Julia

Non-parametric statistics

- ranked data
- Likert scale
- **carrying out a heart transplant with a shovel**
- hard to prove anything to satisfactory degree

Statistics with Julia



```
> summary(Fit)

Call:
lm(formula = Fluo ~ Conc)

Residuals:
       1        2        3        4        5        6        7
 0.58214 -0.37857 -0.23929 -0.50000  0.33929  0.17857  0.01786

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.5179     0.2949   5.146  0.00363 **
Conc          1.9304     0.0409  47.197 8.07e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4328 on 5 degrees of freedom
Multiple R-squared: 0.9978,     Adjusted R-squared: 0.9973
F-statistic:  2228 on 1 and 5 DF,  p-value: 8.066e-08
```

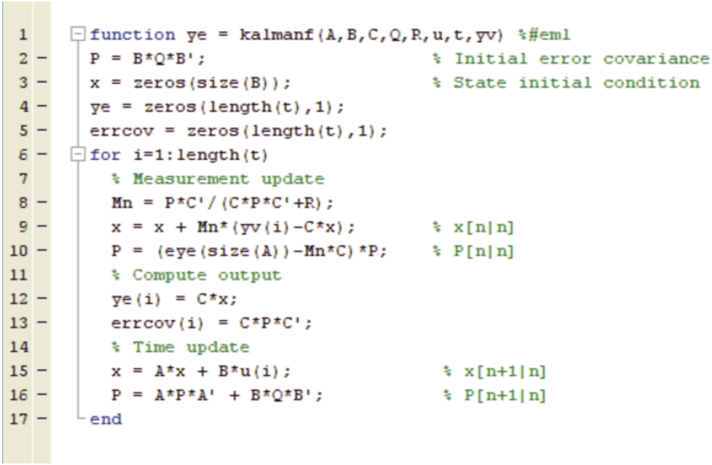
Figure:

Statistics with Julia

This is `R`, but same argument applies to Julia.

- `AIC()`
- `summary()`
- `cor.test()`
- `plot(Fit)`

```
julia> # Pkg.clone("git://github.com/JuliaQuant/MarketData.

julia> using MarketData, Gadfly, HypothesisTests

julia> dist = percentchange(cl).values;

julia> funkydist = dist[100:300];

julia> SignTest(funkydist)
Sign test

median = 0.0
x = 111
n = 201

Two-sided p-value:
p = 0.15816534520094128
```



```
function ye = kalmanf(A,B,C,Q,R,u,t,yv) %#eml
P = B*Q*B';                     % Initial error covariance
x = zeros(size(B));             % State initial condition
ye = zeros(length(t),1);
errcov = zeros(length(t),1);
for i=1:length(t)
  % Measurement update
  Mn = P*C'/(C*P*C'+R);
  x = x + Mn*(yv(i)-C*x);       % x[n|n]
  P = (eye(size(A))-Mn*C)*P;    % P[n|n]
  % Compute output
  ye(i) = C*x;
  errcov(i) = C*P*C';
  % Time update
  x = A*x + B*u(i);              % x[n+1|n]
  P = A*P*A' + B*Q*B';           % P[n+1|n]
end
```

Figure:

Statistics with Julia

- StatsBase
- DataFrames
- RDatasets

- Stuff that gets me
- I still think in "R", not MATLAB
- Why is this not working (in Julia)?

```
myData[myData < 400]
```

If I thought in MATLAB, Julia is fairly easy to pick up

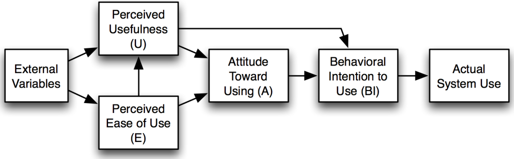
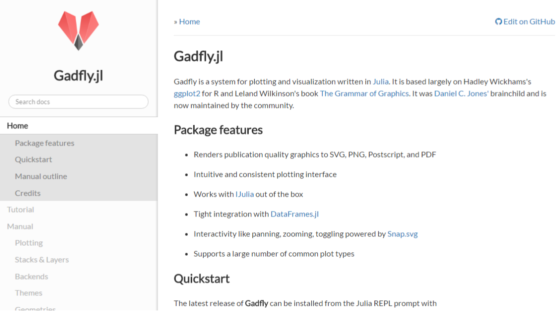

Perceived Usefulness (U)
External Variables
Attitude Toward Using (A)
Behavioral Intention to Use (BI)
Actual System Use
Perceived Ease of Use (E)

Figure:



Gadfly.jl

Gadfly is a system for plotting and visualization written in Julia. It is based largely on Hadley Wickhams's ggplot2 for R and Leland Wilkinson's book The Grammar of Graphics. It was Daniel C. Jones' brainchild and is now maintained by the community.

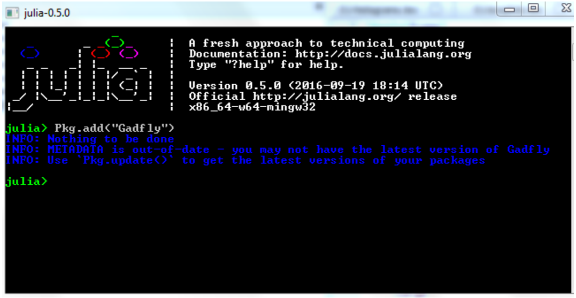
Package features

- Renders publication quality graphics to SVG, PNG, Postscript, and PDF
- Intuitive and consistent plotting interface
- Works with IJulia out of the box
- Tight integration with DataFrames.jl
- Interactivity like panning, zooming, toggling powered by Snap.svg
- Supports a large number of common plot types

Quickstart

The latest release of **Gadfly** can be installed from the Julia REPL prompt with

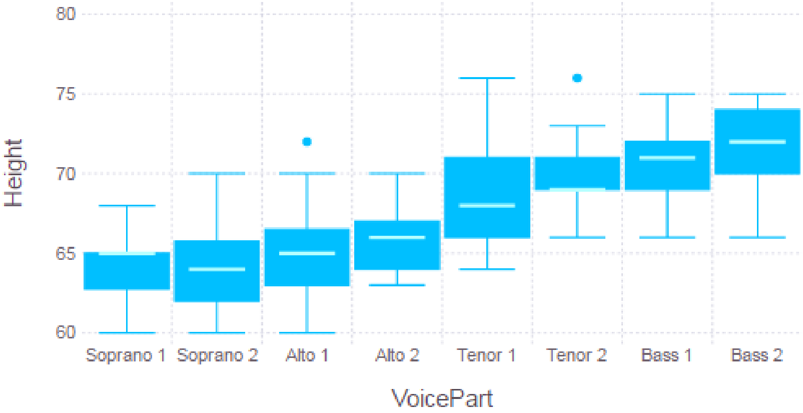
Figure:



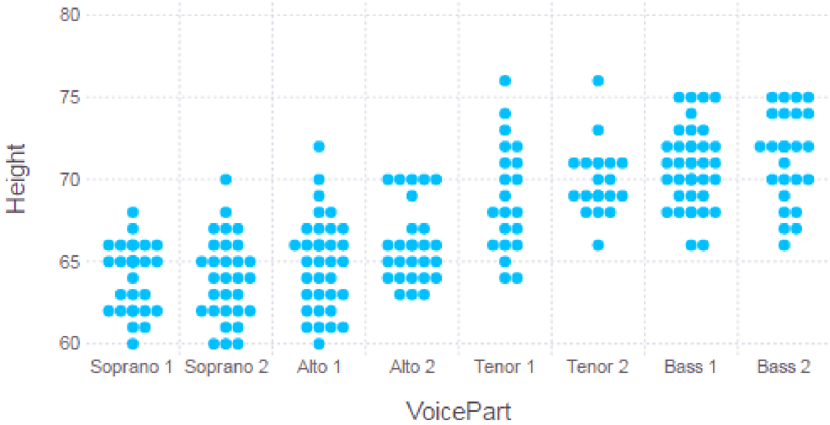
```
               _
   _       _ _(_)_     |  A fresh approach to technical computing
  (_)     | (_) (_)    |  Documentation: http://docs.julialang.org
   _ _   _| |_  __ _   |  Type "?help" for help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 0.5.0 (2016-09-19 18:14 UTC)
 _/ |\__'_|_|_|\__'_|  |  Official http://julialang.org/ release
|__/                   |  x86_64-w64-mingw32

julia> Pkg.add("Gadfly")
INFO: Nothing to be done
INFO: METADATA is out-of-date - you may not have the latest version of Gadfly
INFO: Use `Pkg.update()` to get the latest versions of your packages

julia>
```

Figure:

Statistics with Julia

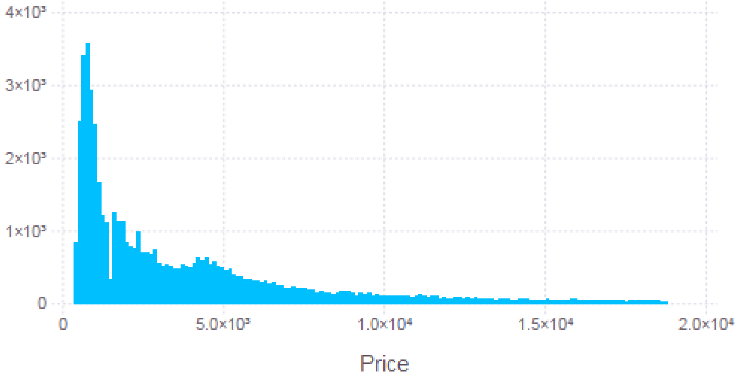


Statistics with Julia



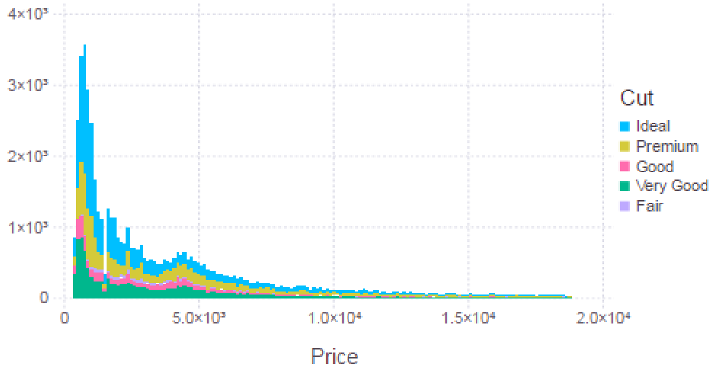
Statistics with Julia

```
plot(dataset("ggplot2", "diamonds"),
    x="Price", Geom.histogram)
```



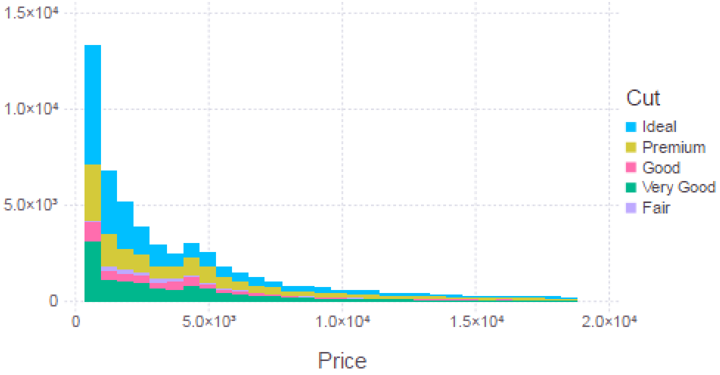
Statistics with Julia

```
plot(dataset("ggplot2", "diamonds"),
      x="Price", color="Cut", Geom.histogram)
```



Statistics with Julia

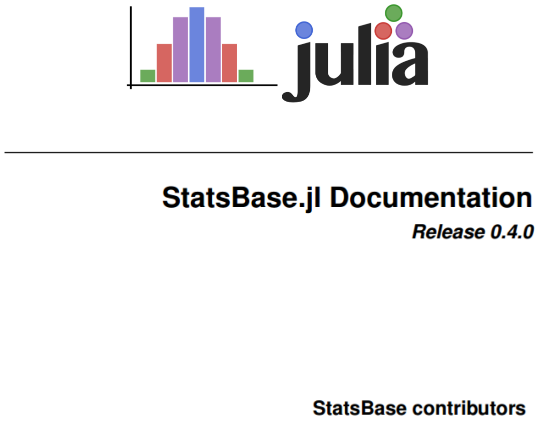
```
plot(dataset("ggplot2", "diamonds"), x="Price",
     color="Cut", Geom.histogram(bincount=30))
```



JuliaStats

Statistics and **Machine Learning** made easy in Julia.

- Easy to use tools for statistics and machine learning.
- Extensible and reusable models and algorithms
- Efficient and scalable implementation
- Community driven, and open source



StatsBase.jl Documentation

Release 0.4.0

StatsBase contributors

Statistics with Julia

StatsBase.jl

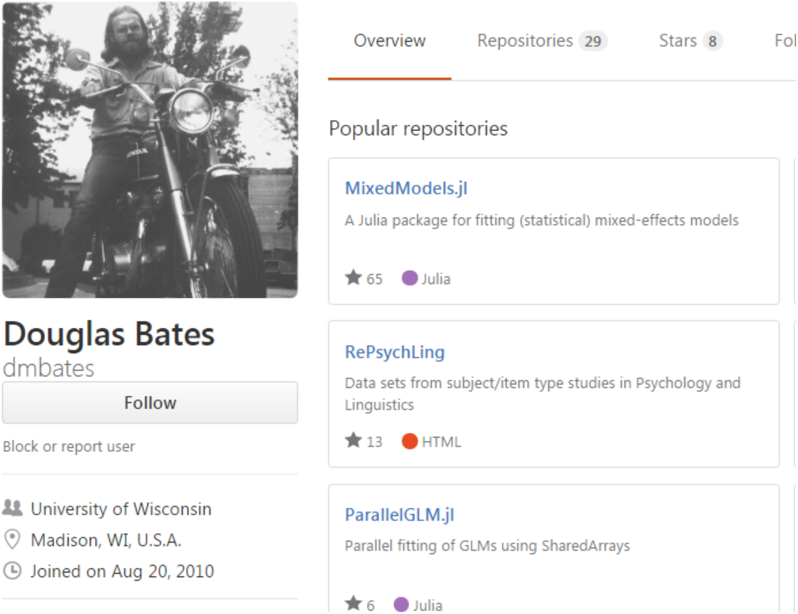
- StatsBase.jl is a Julia package that provides basic support for statistics.
- Particularly, it implements a variety of statistics-related functions, such as scalar statistics, high-order moment computation, counting, ranking, covariances, sampling, and empirical density estimation.

Statistics with Julia

Measure of Centrality : Who am I?

$$\frac{Q_1 + 2Q_2 + Q_3}{4}$$

Statistics with Julia



Douglas Bates
dmbates

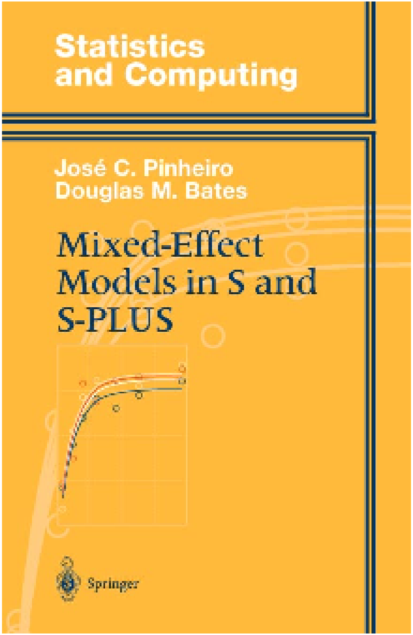
University of Wisconsin
Madison, WI, U.S.A.
Joined on Aug 20, 2010

Popular repositories

MixedModels.jl
A Julia package for fitting (statistical) mixed-effects models
★ 65 ● Julia

RePsychLing
Data sets from subject/item type studies in Psychology and Linguistics
★ 13 ● HTML

ParallelGLM.jl
Parallel fitting of GLMs using SharedArrays
★ 6 ● Julia



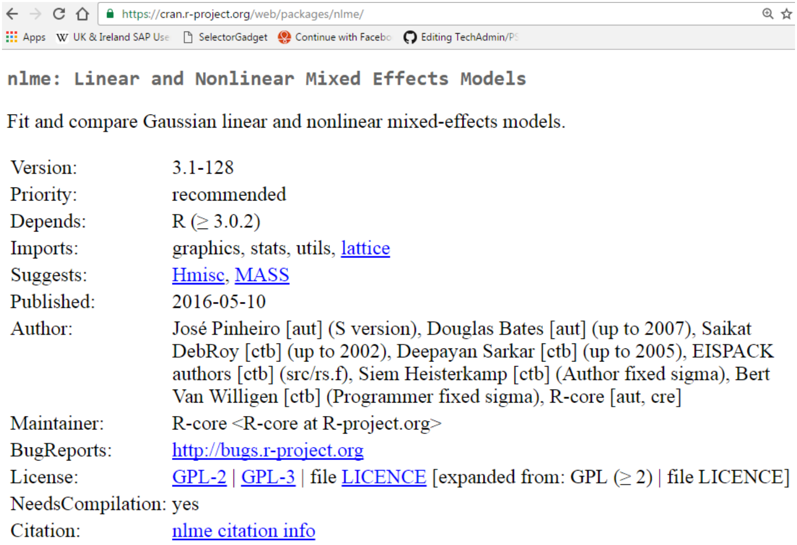
Statistics and Computing

José C. Pinheiro
Douglas M. Bates

Mixed-Effect Models in S and S-PLUS

Springer

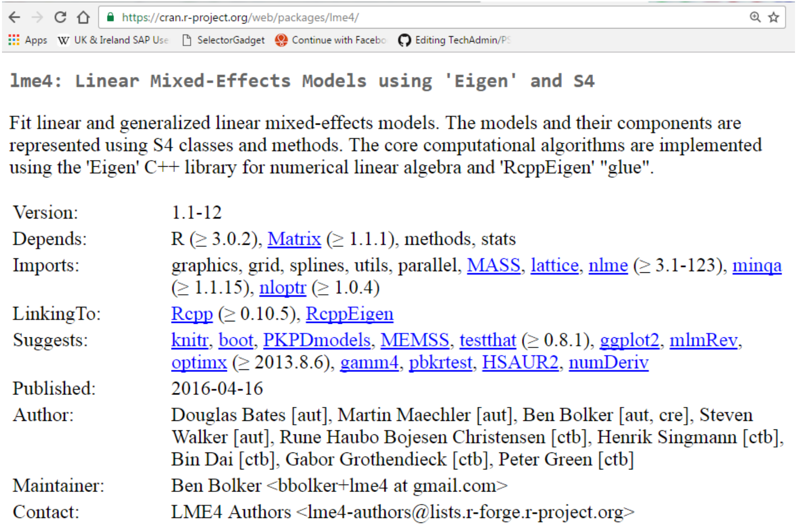
Figure:



nlme: Linear and Nonlinear Mixed Effects Models

Fit and compare Gaussian linear and nonlinear mixed-effects models.

Version:	3.1-128
Priority:	recommended
Depends:	R (≥ 3.0.2)
Imports:	graphics, stats, utils, lattice
Suggests:	Hmisc, MASS
Published:	2016-05-10
Author:	José Pinheiro [aut] (S version), Douglas Bates [aut] (up to 2007), Saikat DebRoy [ctb] (up to 2002), Deepayan Sarkar [ctb] (up to 2005), EISPACK authors [ctb] (src/rs.f), Siem Heisterkamp [ctb] (Author fixed sigma), Bert Van Willigen [ctb] (Programmer fixed sigma), R-core [aut, cre]
Maintainer:	R-core <R-core at R-project.org>
BugReports:	http://bugs.r-project.org
License:	GPL-2 | GPL-3 | file LICENCE [expanded from: GPL (≥ 2) | file LICENCE]
NeedsCompilation:	yes
Citation:	nlme citation info

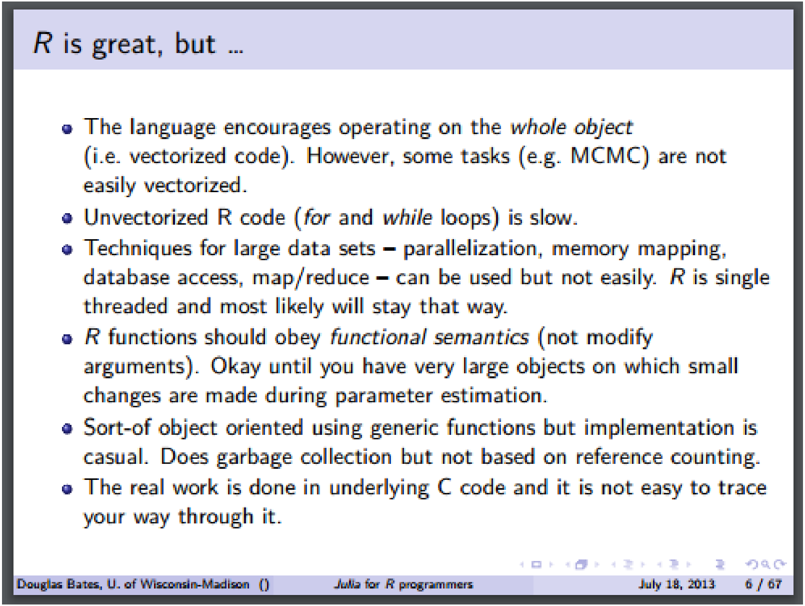


lme4: Linear Mixed-Effects Models using 'Eigen' and S4

Fit linear and generalized linear mixed-effects models. The models and their components are represented using S4 classes and methods. The core computational algorithms are implemented using the 'Eigen' C++ library for numerical linear algebra and 'RcppEigen' "glue".

Version:	1.1-12
Depends:	R (≥ 3.0.2), Matrix (≥ 1.1.1), methods, stats
Imports:	graphics, grid, splines, utils, parallel, MASS, lattice, nlme (≥ 3.1-123), minqa (≥ 1.1.15), nloptr (≥ 1.0.4)
LinkingTo:	Rcpp (≥ 0.10.5), RcppEigen
Suggests:	knitr, boot, PKPDmodels, MEMSS, testthat (≥ 0.8.1), ggplot2, mlmRev, optimx (≥ 2013.8.6), gamm4, pbkrtest, HSAUR2, numDeriv
Published:	2016-04-16
Author:	Douglas Bates [aut], Martin Maechler [aut], Ben Bolker [aut, cre], Steven Walker [aut], Rune Haubo Bojesen Christensen [ctb], Henrik Singmann [ctb], Bin Dai [ctb], Gabor Grothendieck [ctb], Peter Green [ctb]
Maintainer:	Ben Bolker <bbolker+lme4 at gmail.com>
Contact:	LME4 Authors <lme4-authors@lists.r-forge.r-project.org>

Statistics with Julia



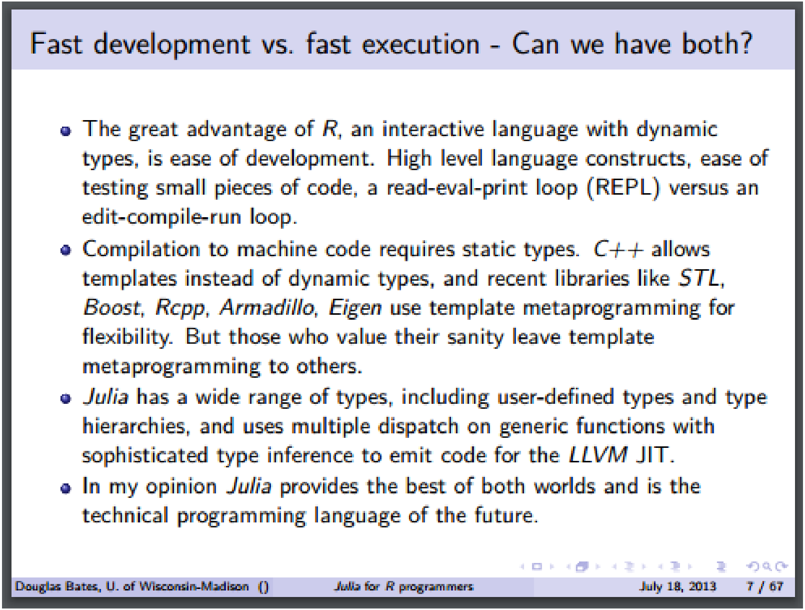
R is great, but ...

- The language encourages operating on the *whole object* (i.e. vectorized code). However, some tasks (e.g. MCMC) are not easily vectorized.
- Unvectorized R code (*for* and *while* loops) is slow.
- Techniques for large data sets – parallelization, memory mapping, database access, map/reduce – can be used but not easily. *R* is single threaded and most likely will stay that way.
- *R* functions should obey *functional semantics* (not modify arguments). Okay until you have very large objects on which small changes are made during parameter estimation.
- Sort-of object oriented using generic functions but implementation is casual. Does garbage collection but not based on reference counting.
- The real work is done in underlying C code and it is not easy to trace your way through it.

Douglas Bates, U. of Wisconsin-Madison () *Julia for R programmers* July 18, 2013 6 / 67

Figure:

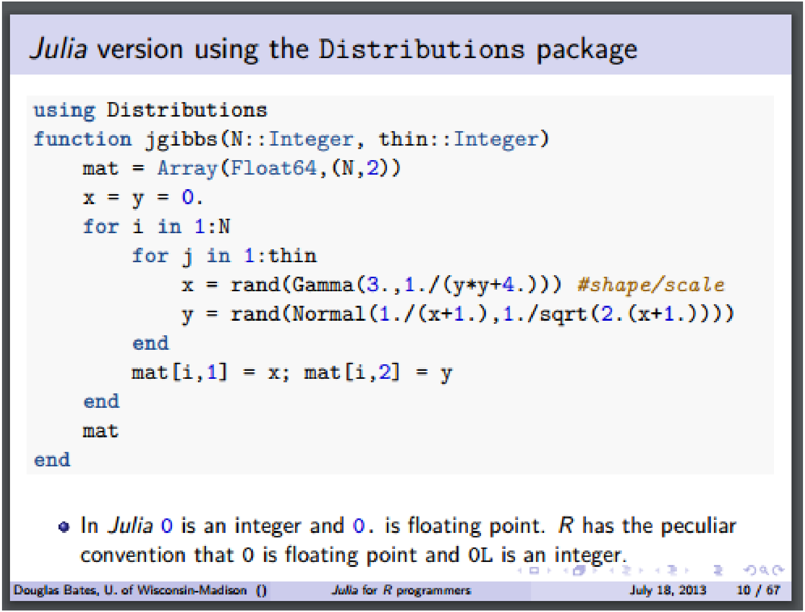
Statistics with Julia



Fast development vs. fast execution - Can we have both?

- The great advantage of *R*, an interactive language with dynamic types, is ease of development. High level language constructs, ease of testing small pieces of code, a read-eval-print loop (REPL) versus an edit-compile-run loop.
- Compilation to machine code requires static types. *C++* allows templates instead of dynamic types, and recent libraries like *STL*, *Boost*, *Rcpp*, *Armadillo*, *Eigen* use template metaprogramming for flexibility. But those who value their sanity leave template metaprogramming to others.
- *Julia* has a wide range of types, including user-defined types and type hierarchies, and uses multiple dispatch on generic functions with sophisticated type inference to emit code for the *LLVM* JIT.
- In my opinion *Julia* provides the best of both worlds and is the technical programming language of the future.

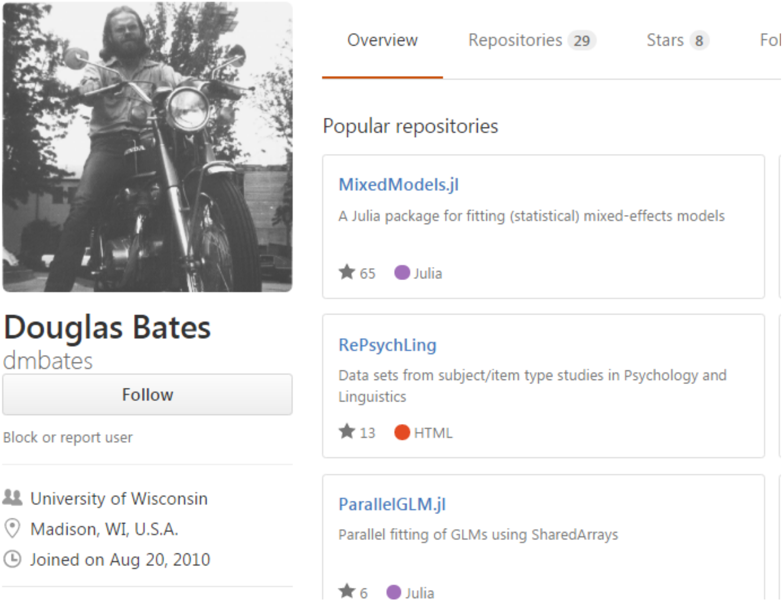
Statistics with Julia



Julia version using the `Distributions` package

```
using Distributions
function jgibbs(N::Integer, thin::Integer)
    mat = Array(Float64,(N,2))
    x = y = 0.
    for i in 1:N
        for j in 1:thin
            x = rand(Gamma(3.,1./(y*y+4.))) #shape/scale
            y = rand(Normal(1./(x+1.),1./sqrt(2.(x+1.))))
        end
        mat[i,1] = x; mat[i,2] = y
    end
    mat
end
```

- In *Julia* `0` is an integer and `0.` is floating point. *R* has the peculiar convention that `0` is floating point and `0L` is an integer.



Douglas Bates

dmbates

University of Wisconsin

Madison, WI, U.S.A.

Joined on Aug 20, 2010

Popular repositories

MixedModels.jl

A Julia package for fitting (statistical) mixed-effects models

★ 65 ● Julia

RePsychLing

Data sets from subject/item type studies in Psychology and Linguistics

★ 13 ● HTML

ParallelGLM.jl

Parallel fitting of GLMs using SharedArrays

★ 6 ● Julia

Statistics with Julia

`MixedModels`

- development started in 2012 by Bates
- little or no documentation outside of examples
- implemented exclusively in Julia (about 1600 lines of code)
- fits LMMs. Development of GLMM capabilities is planned.
- single formula specification similar to lme4.

Statistics with Julia

Douglas Bates on Mixed Models

The most important aspect of Julia is ”one language”. You develop in the same language in which you optimize.

The type system in Julia allows me to incorporate the different kinds of penalized least squares solvers in what to me is a clean way, thereby taking advantage of structural simplifications in simple, but common, cases.

It is possible to do this in R/C++/Rcpp/EIgen but it would be a massive headache and perhaps beyond my abilities to do it well.

Statistics with Julia

Douglas Bates on Mixed Models

The numerical methods implemented in lme4 are, in my opinion, superior to those in nlme, mainly through the use of the relative covariance factor and the profiled log-likelihood.

These may seem like details but to me they are very important. The motiviation for incorporating sparse matrix classes in the Matrix package and accessing the CHOLMOD code was to provide a general method for fitting such models.

Using C++, Rcpp and RcppEigen was motivated by trying to provide generality and speed. The end result is confusing (my fault entirely) and fragile.

Statistics with Julia

OnlineStats.jl

- `OnlineStats.jl` provides online algorithms for statistical models.
- Online algorithms are well suited for streaming data or when data is too large to hold in memory.
- Observations are processed one at a time and all algorithms use O(1) memory.

https://github.com/joshday/OnlineStats.jl

Statistics with Julia

```
using OnlineStats
o = Mean()

All OnlineStats can be updated

y = randn(100)

for yi in y
fit!(o, y)
end

# or more simply:
fit!(o, y)
OnlineStats share a common interface

value(o)  # associated value of an OnlineStat
nobs(o)   # number of observations used
```

What Can OnlineStats Do? While many estimates can be calculated analytically with an online algorithm, several type rely on stochastic approximation.

Summary Statistics

Mean: Mean, Means

Variance: Variance, Variances

Quantiles: QuantileMM, QuantileSGD

Covariance Matrix: CovMatrix

Maximum and Minimum: Extrema

Skewness and Kurtosis: Moments

Sum/Differences: Sum, Sums, Diff, Diffs

Density Estimation

```
distributionfit(D, data)
For D in [Beta, Categorical, Cauchy, Gamma, LogNormal, Nor
Gaussian Mixtures: NormalMix
Predictive Modeling
```

Linear Regression: LinReg, StatLearn

Logistic Regression: StatLearn

Poisson Regression: StatLearn

Support Vector Machines: StatLearn

Quantile Regression: StatLearn, QuantRegMM

Huber Loss Regression: StatLearn

L1 Loss Regression: StatLearn

Other

K-Means clustering: KMeans

Bootstrapping: BernoulliBootstrap, PoissonBootstrap

Approximate count of distinct elements: HyperLogLog