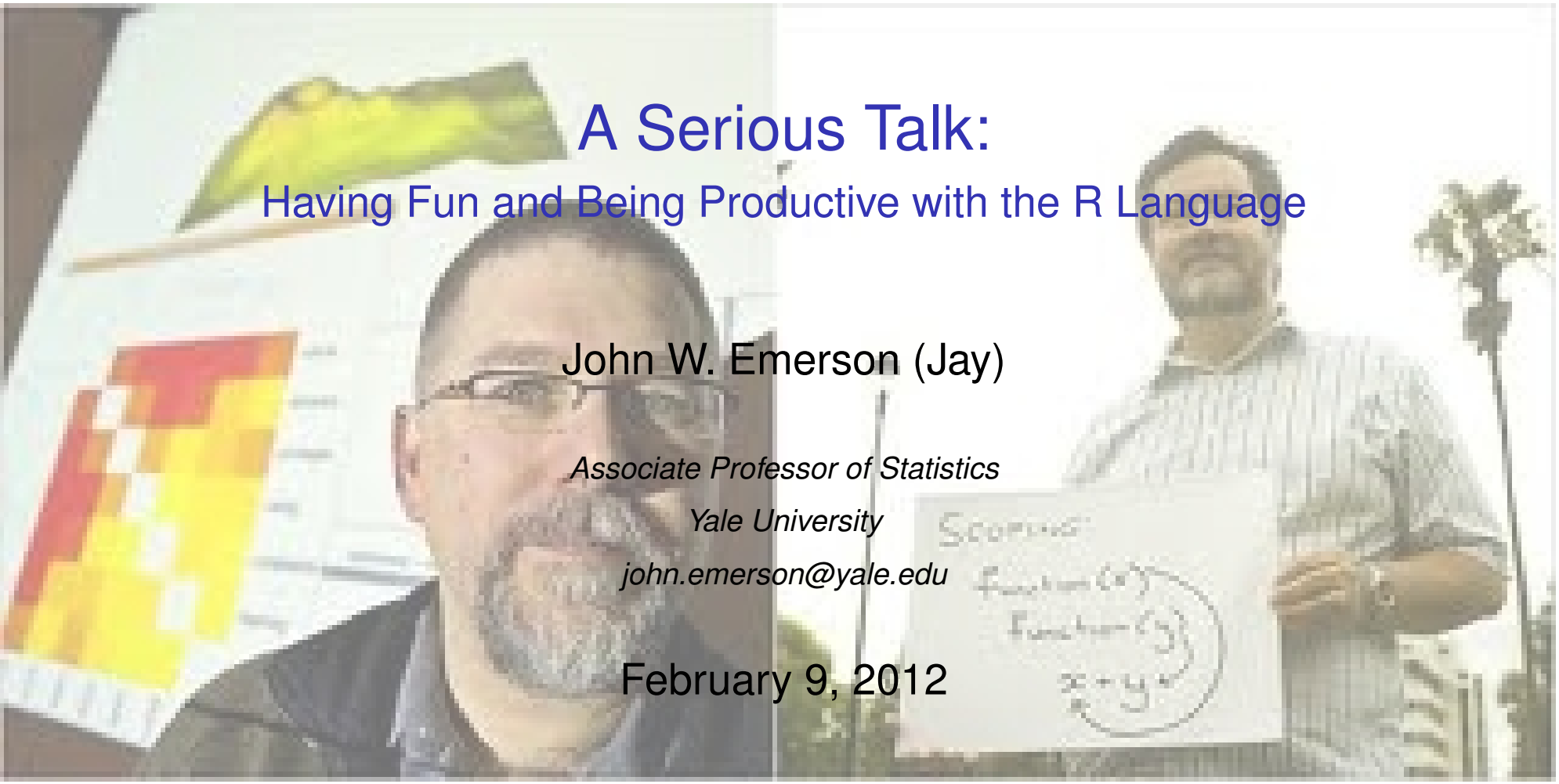
Source: `http://kuwisdelu.blogspot.com/`



A Serious Talk:

Having Fun and Being Productive with the R Language

John W. Emerson (Jay)

Associate Professor of Statistics

Yale University

john.emerson@yale.edu

February 9, 2012

John Chambers: The S Language

What is R (concise version)?

- An open-source statistical programming environment based on John Chambers' S language
- A system for interactive data analysis
- A high-level programming language

More about R (courtesy of Venables, Smith, and the R Core Development Team)

Among other things it has

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either directly at the computer or on hardcopy, and
- a well developed, simple and effective programming language (called 'S') which includes conditionals, loops, user defined recursive functions and input and output facilities. (Indeed most of the system supplied functions are themselves written in the S language.)

More about R (courtesy of Venables, Smith, and the R Core Development Team

The term “environment” is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R is very much a vehicle for newly developing methods of interactive data analysis. It has developed rapidly, and has been extended by a large collection of packages. However, most programs written in R are essentially ephemeral, written for a single piece of data analysis.

Why R?

R is the *lingua franca* of statistics

- The syntax is simple and well-suited for data exploration and analysis.
- It has excellent graphical capabilities.
- It is extensible, with over 2500 packages available on CRAN alone.
- It is open source and freely available for Windows/MacOS/Linux platforms.
- A fantastic language/environment for R&D even if production tools are eventually coded in C/C++, Java, ...

Other Statistical Software

- Excel (not really)
- SAS (expensive, but...)
- SPSS, Stata, JMP, Minitab, and many others...
- MATLAB (not statistical software *per se*, but highly regarded – by me and others – as a programming language with many statistical features)

Flavors (or flavours) of R

- Start here: `http://www.r-project.org/`
- Formal R distributions offered for Debian and Ubuntu (at least...)
- Binaries available for Windows and Mac users (building from source takes a bit more work on these platforms than on Linux)
- REvolution Analytics R, `http://www.revolutionanalytics.com/`
- R Studio, `http://rstudio.org/`

But for serious folks

```
jay@bayesman:~/R-2.13.2$ ./configure
< snip >
jay@bayesman:~/R-2.13.2$ make
< snip >
jay@bayesman:~/R-2.13.2$ make install
< snip >
jay@bayesman:~/R-2.13.2$ R
< see next slide >
```

Lots of hand-waving here, but details probably not needed for the current audience.

```
R version 2.13.2 (2011-09-30)
Copyright (C) 2011 The R Foundation for Statistical Computi
ISBN 3-900051-07-0
Platform: x86_64-unknown-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

An Important Question

An Important Question

What the heck is this wooly academic going to talk about today?

An Important Question

What the heck is this wooly academic going to talk about today?

- I want to show you why I find R so exciting, through examples at many different levels of sophistication.
- I ask you to look beyond the specifics of each example. Don't lose sight of the forest through the trees.
- I hope you'll believe my claim that yes, you can hit the ground running with R, and that R empowers you rather than limits you. R gives you the flexibility to think on your feet and innovate, potentially going boldly where no analyst as gone before.

So please, ask questions!

Example 1: Examining web server logs

See `Scrape_access_log.R`.

Detour!

`http://www.r-project.org` and CRAN

Example 2: As long as we're talking about web stuff and R packages...

CGI Programming with R via Simon Urbanek's Rserve and FastRWeb packages

- Anyone: see `http://jayemerson.blogspot.com`, Oct 6, 2011 post, for links and basic installation instructions. Sorry, I'm not an active blogger or tweeter.
- Jay: sudo /var/FastRWeb/code/start
- Jay: cat /var/FastRWeb/web.R/LUG.R
- Jay: `http://localhost/cgi-bin/R/LUG`
- Jay: `http://localhost/cgi-bin/R/LUG?n=1000`
- Jay: `http://localhost/cgi-bin/R/cpmain`
- Anyone: `http://epi.yale.edu/dataexplorer/countryprofiles`

Example 3: College basketball point spread and game results

See `http://www.goldsheet.com`, and look at `cbb_2008_09.txt`.

We could do this interactively (from `Scrape_cbb.R`) or with the following slides.

Example 3: College basketball point spread and game results

```
> x <- scan("cbb_2008_09.txt", what = "", sep = "\n")
> x[74:86]
 [1] "ALABAMA"
 [2] "(SUR:  18-14   PSR:  15-13)"
 [3] "11/16  Mercer                          69-72  H"
 [4] "11/19  Florida A&M                     89-48  H"
 [5] "11/24  Oregon              L    -4'    69-92  N"
 [6] "11/25  Chaminade           W    -20    78-56  V"
 [7] "11/26  Saint Joseph's      W    +1'    58-48  N"
 [8] "12/2   Alabama A&M                     64-56  H"
 [9] "12/7   La.-Lafayette       W    -15'   61-44  H"
[10] "12/13  Texas A&M           L    -3     78-86  H1"
[11] "12/17  Tennessee St.       L    -17    75-66  H"
[12] "12/22  Chattanooga         W    -12    82-63  H"
[13] "12/28  Yale                L    -17    66-63  H"
```

From data scraping to data analysis

```
> home <- substring(x, 43, 43)
> table(home)
home
        :     E     H     n     N     V
  402     1     1  3227     1   709  2574
> x <- x[home == "H"]
```

From data scraping to data analysis

```
> temp <- substring(x, 29, 35)
> temp2 <- gsub("'", ".5", temp)
> temp3 <- gsub("P", "0", temp2)
> cbind(temp, temp2, temp3)[3112:3117, ]
     temp       temp2       temp3
[1,] "  -19' " "  -19.5 " "  -19.5 "
[2,] "  -10' " "  -10.5 " "  -10.5 "
[3,] "       " "        "  "        "
[4,] "  P    " "  P     "  "  0     "
[5,] "  -9'   " "  -9.5   " "  -9.5   "
[6,] "  -7    " "  -7    "  "  -7    "
```

From data scraping to data analysis

```
> temp <- substring(x, 36, 42)
> temp <- strsplit(temp, "-")
> table(sapply(temp, length))
   1    2
   1 3226
> which(sapply(temp, length) == 1)
[1] 3005
> x[3005]
[1] "12/23  Lehigh                              PPD.    H"
> for (i in which(sapply(temp, length) == 1)) {
+     temp[[i]] <- c(NA, NA)
+ }
> table(sapply(temp, length))
   2
3227
```

From data scraping to data analysis

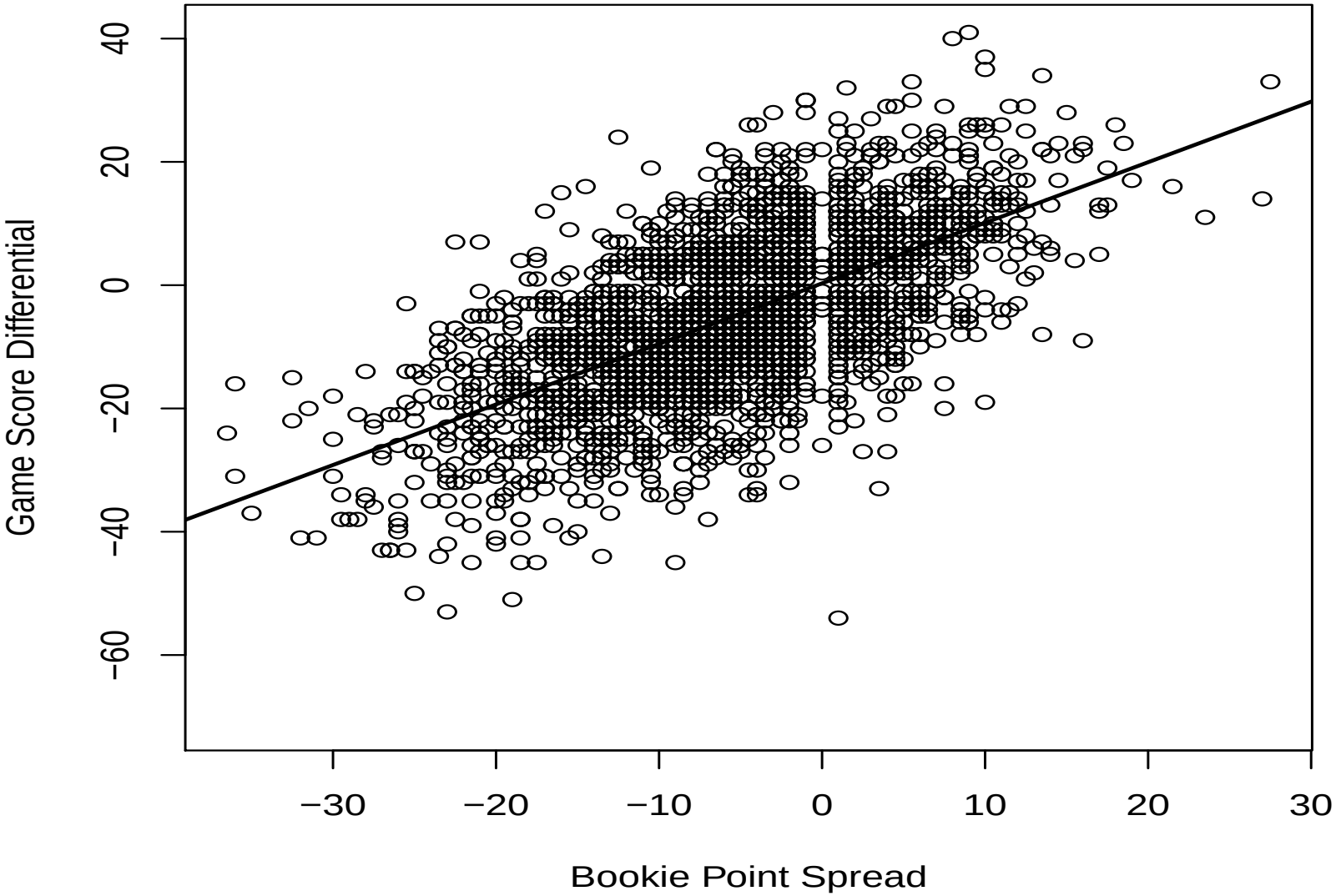
```
> scores <- matrix(unlist(temp), ncol=2, byrow=TRUE)
> z <- data.frame(pointspread = as.numeric(temp3),
+                 hscore = as.numeric(scores[,1]),
+                 vscore = as.numeric(scores[,2]))
> z$gamediff <- z$vscore - z$hscore


> z[3115:3117, ]
     pointspread hscore vscore gamediff
3115         0.0     69     74        5
3116        -9.5     65     61       -4
3117        -7.0     74     45      -29
> summary(z$gamediff)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-71.000 -17.000  -8.000  -7.923   2.000  41.000
   NA's
  1.000
```

A linear model

```
> lm.bb <- lm(gamediff ~ pointspread, data=z)
> summary(lm.bb)
...
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.29826    0.24646    1.21    0.226
pointspread  0.98231    0.02412   40.73   <2e-16 ***
...
Residual standard error: 10.49 on 2580 degrees of freedom
  (645 observations deleted due to missingness)
Multiple R-squared: 0.3914, Adjusted R-squared: 0.3911
F-statistic:  1659 on 1 and 2580 DF,  p-value: < 2.2e-16
```

A plot: college hoops



Bayesian change point analysis, the package management system, the C/C++ interface, massive matrices and shared memory, and a portable framework for parallel computing

Bayesian change point analysis, the package management system, the C/C++ interface, massive matrices and shared memory, and a portable framework for parallel computing

Really?

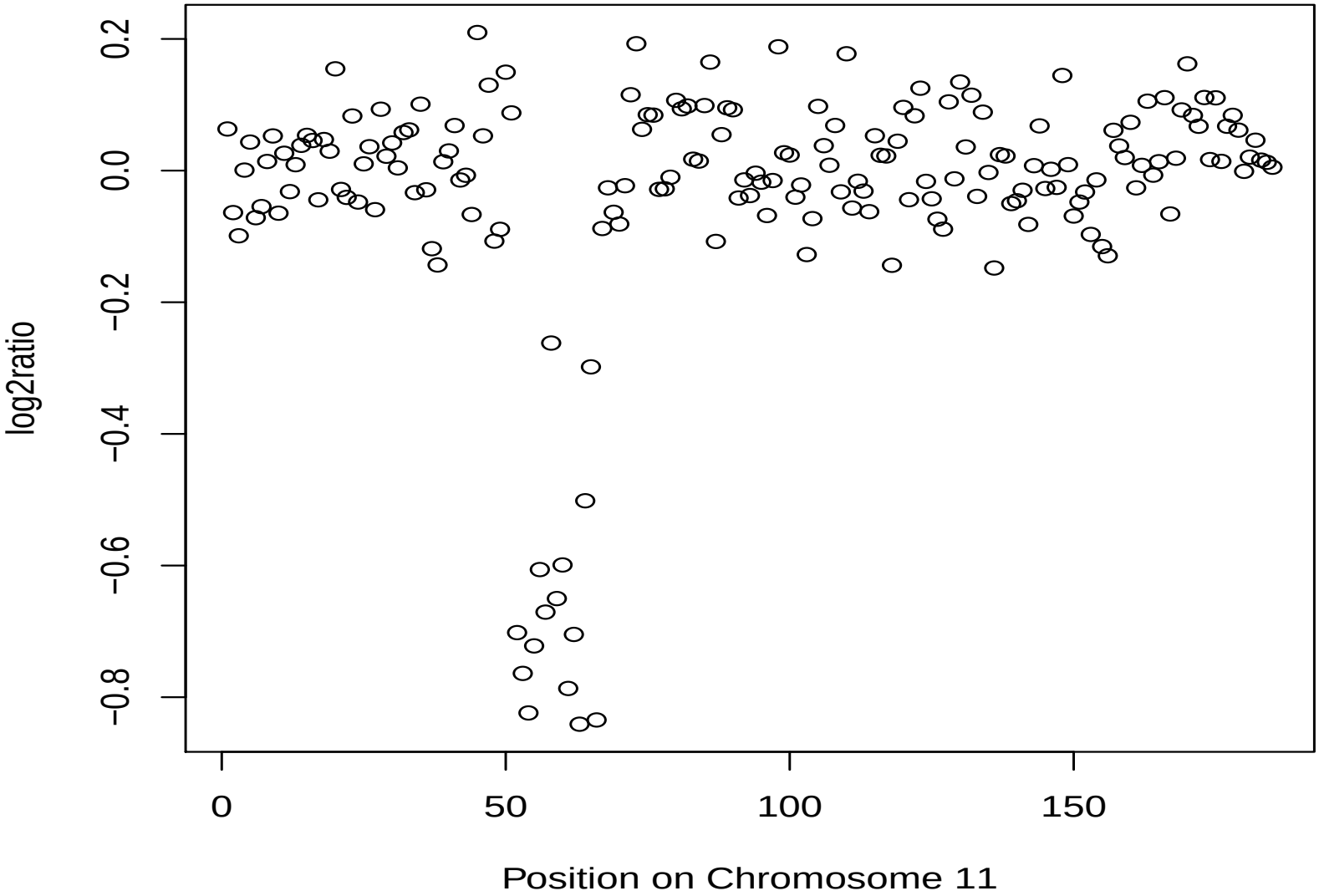
Bayesian change point analysis, the package management system, the C/C++ interface, massive matrices and shared memory, and a portable framework for parallel computing

Well... let me pause for questions.

I understand completely if some of you need to (or want to) run off at this point (don't be shy, I think the pub sounds pretty good at the moment).

If the group would be more interested in a completely different example of using R for interactive data analysis, we could look for nationalistic bias in judging of Olympic diving?

Example: Coriell cell lines (raw data)

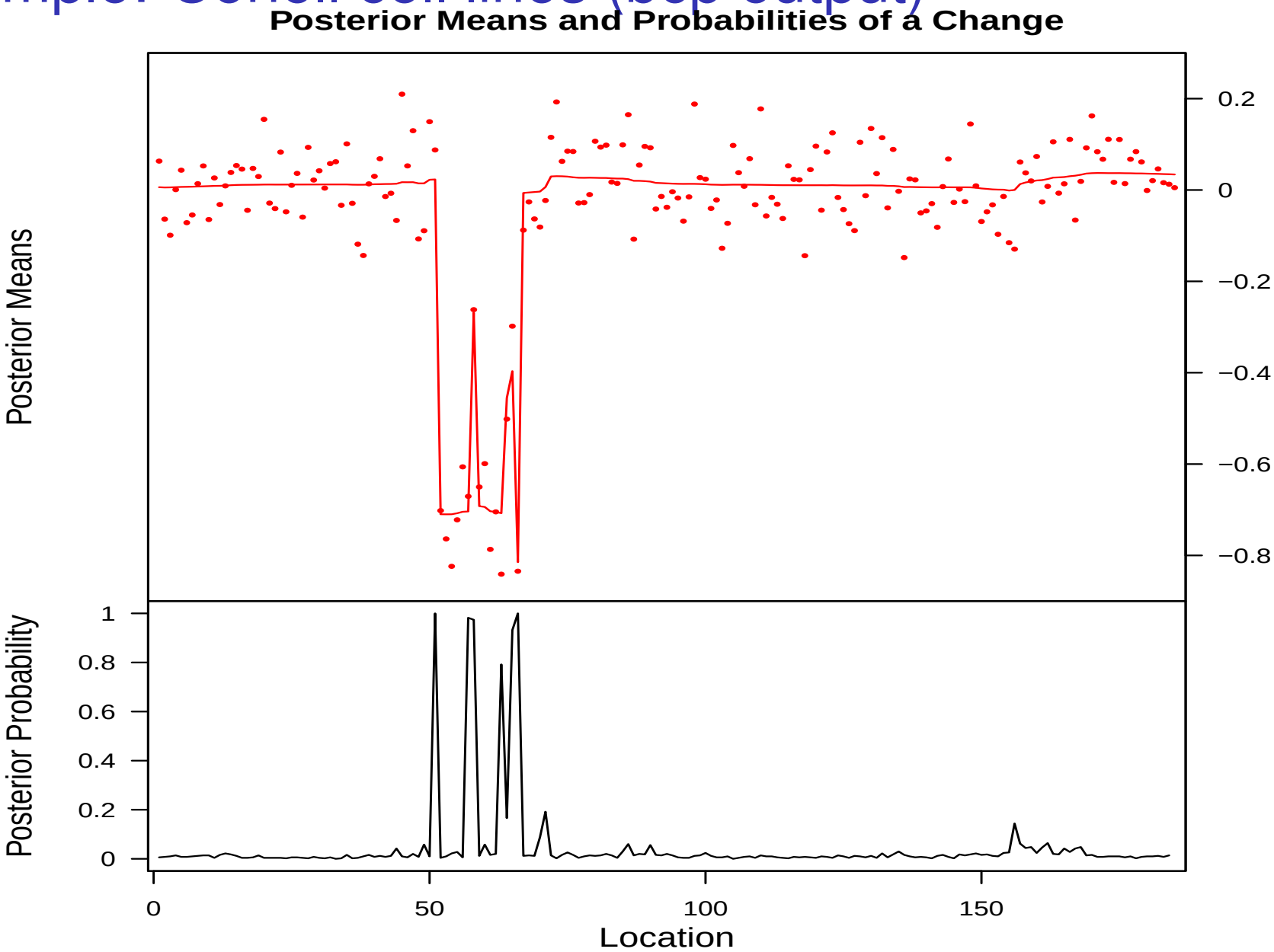


Example: Coriell cell lines (bcp analysis)

See `http://cran.r-project.org/web/packages/bcp/`.

```
> library(bcp)
> data(coriell)
>
> chrom11 <- as.vector(na.omit(
+                coriell$Coriell.05296[
+                  coriell$Chromosome == 11]))
>
> bcp.11 <- bcp(chrom11)
> plot(bcp.11)
```

Example: Coriell cell lines (bcp output)



The Bayesian change point model

ρ unknown partition into continuous blocks, with the transition between blocks being the "change points."

p probability of a change point at position i, independently for all i.

X_i observations assumed independent $N(\mu_i, \sigma^2)$, where in this notation the μ_i are equal for all i within a block.

μ_{jk} mean of block from position $j+1$ to k, with prior $N(\mu_0, \sigma_0^2/(k-j))$. Note: larger deviations from μ_0 are expected for shorter blocks, but weak signals can be detected when sufficient data are available.

w defined as $\sigma^2/(\sigma^2+\sigma_0^2)$ for convenience.

(See the supplementary materials for more information if you are interested.)

Notes on Bayesian change point analysis

- Barry and Hartigan (1993): “A Bayesian analysis for change point problems” in *JASA*, 88, 309-319.
- Erdman and Emerson (2008): “A fast Bayesian change point analysis for the segmentation of microarray data” in *Bioinformatics*, 24 (19), 2143-2148.
- Erdman and Emerson (2007): “bcp: an R package for performing a Bayesian analysis of change point problems” in *JSS*, 23 (3).
- An exact implementation of the Bayes procedure is possible but the calculations would be $O(n^3)$.
- Package **bcp** provides a fast $O(n)$ MCMC implementation:
  - inefficient MCMC would be $O(n^2)$
  - solves some nasty numerical problems with large data
  - supports parallel MCMC
  - newly extended for multivariate series with a common change point structure
  - now uses package Rcpp for a more elegant interface to the C code

Building an R build environment

If you want to build R packages, you'll need the full R build environment (not just the pre-compiled R binaries that most non-Linux R users get from CRAN). See

`http://www.stat.yale.edu/~jay/RPC/RPackages.pdf`

A simple package: function babywhatis()

```
> ls()
 [1] "bcp.11"   "chrom11" "coriell" "home"     "i"
 [6] "lm.bb"    "scores"  "temp"    "temp2"    "temp3"
[11] "x"        "z"
> rm(list = ls())
> ls()
character(0)
> babywhatis <- function(x) {
+     if (!is.data.frame(x)) {
+         x <- data.frame(x)
+         warning("Object coerced to a data frame.\n")
+     }
+     return(unlist(lapply(x, class)))
+ }
> ls()
[1] "babywhatis"
```

A simple package: the package skeleton

```
> package.skeleton("MyToolkit")
Creating directories ...
Creating DESCRIPTION ...
Creating Read-and-delete-me ...
Saving functions and data ...
Making help files ...
Done.
Further steps are described in
  './MyToolkit/Read-and-delete-me'.
```

Let's go investigate together; we'll explore the package structure, make minor modifications, and will check/build/install it. More information is available in `http://www.stat.yale.edu/~jay/RPC/RPackages.pdf`; Jay, use the local RPC folder.

C/C++ interface: A simple example (matrix column minima)

This material wouldn't display well in slides. Again see

`http://www.stat.yale.edu/~jay/RPC/RPackages.pdf`

and, specifically, materials in

`http://www.stat.yale.edu/~jay/RPC/MyToolkitWithC/`

foreach

The user may register any one of several “parallel backends” like **doMC** or **doSNOW**, or none at all. The code will either run sequentially or will make use of the parallel backend, if specified, without code modification.

```
> library(foreach)
> library(doMC)
> registerDoMC(2)
>
> a <- 10
> ans <- foreach(i = 1:5, .combine = c) %dopar%
+     {
+         a + i^2
+     }
>
> ans
 [1]   11   14   19   26   35
```

Parallel MCMC in package bcp

- An older version of **bcp** used **NetWorkSpaces** for parallel MCMC; this was very difficult to install and use, and the code was not portable to other parallel environments.
- The new **bcp** uses Steve Weston's **foreach** package, and the user may choose from a variety of parallel backends.
- I strongly recommend `foreach()` for parallel programming, to both users and package developers.
- Again, see supplementary materials for more information.

Time permitting...

Jay, use `moreexamples.txt`. Or else

```
> x <- read.csv(paste("http://www.stat.yale.edu/~jay/",
+     "diving/Diving2000.csv", sep = ""), as.is = TRUE)
```

Thanks!

- Dirk Eddelbuettel, Bryan Lewis, Steve Weston, and Martin Schultz, for their feedback and advice over the last four years
- Bell Laboratories (Rick Becker, John Chambers and Allan Wilks), for development of the S language
- Ross Ihaka and Robert Gentleman, for their work and unselfish vision for R
- The R Core team
- John Hartigan, for years of teaching and mentoring
- John Emerson (my father, Middlebury College), for getting me started in statistics and pushing me to learn to write code
- Many of my students, for their willingness argue with me (Chandra, Mike, Taylor, and Susan, in particular)

Material

- `http://www.stat.yale.edu/~jay/LUG/`
- `http://www.stat.yale.edu/~jay/RPC/`
- `http://jayemerson.blogspot.com/` (links provided there to Simon Urbanek's cool stuff for CGI programming with R)