

# The Joy of Sweave:

A Beginner's Guide to Reproducible Research with Sweave

Mario Pineda-Krch

Centre for Mathematical Biology, University of Alberta

<http://pineda-krch.com>

January 17, 2011



The Joy of Sweave: A Beginner's Guide to Reproducible Research with Sweave by [Mario Pineda-Krch](#) is licensed under a [Creative Commons](#)

[Attribution-NonCommercial-ShareAlike 3.0 Unported License](#).

# Reproducible Research

*"An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures."*

— Jon Claerbout

- Reproducibility is one of the cornerstones of the modern scientific method
- Traditionally papers publishing experimental and mathematical results contain sufficient information to reproduce the results, e.g. empirical methods or mathematical proofs
- Reproducible Research in computational sciences is about reproducible computational results, e.g. simulation and analysis results (not reproducing experimental results), using the same methods (algorithms, seed) as in the paper
- The majority of computational research is not easily reproducible because
  - Algorithms are typically not described in published papers

- Journals do not require computer code to be deposited in a repository (see e.g. Dryad (<http://datadryad.org/>), KNB (<http://knb.ecoinformatics.org/>), and GenBank (<http://www.ncbi.nlm.nih.gov/genbank/>))
- The code/documentation mismatch fallacy
- Sweave provides a solution to this problem

## Online resources for RR

- Wave lab at Stanford (<http://www-stat.stanford.edu/~wavelab/>): Real world RR using MatLab
- Reproducible Research (<http://reproducibleresearch.net/>): web site is intended to collect information and useful links about reproducible research

*Selected Papers*

\* \* \*

THE JOURNAL  
*of*  
IRREPRODUCIBLE  
RESULTS

THIRD EDITION

*A selection of superb and  
irreproducible research from the illustrious  
and irreproducible archives of the Society  
for Basic Irreproducible Research*

# Coombs et al. 2007. Nature Medicine 13:1276-1277

## Microarrays: retracing steps

### To the editor:

Recently, Potti *et al.*<sup>1</sup> published an article in *Nature Medicine* reporting an approach predicting whether a tumor will respond to chemotherapy. Using publicly available data, they derived signatures from microarray profiles of the NCI-60 human cancer cell lines with known *in vitro* sensitivity or resistance to a particular drug. They used these profiles to predict *in vivo* chemotherapeutic response to seven different drugs. In order to help investigators at our institution use similar approaches, we tried to reproduce their results. We used the same published data and additional information generously supplied by the authors regarding methods, lists of cell lines called sensitive or resistant, and the software used to perform their analysis.

We report here our inability to reproduce their findings. Details of our methods and results are described in the supplementary information (Supplementary Reports 0–9) and are summarized here.

1. We cannot reproduce their selection of cell lines. The most sensitive and resistant lines should be used to focus on drug effects. However, the  $GI_{50}$  (the concentration needed to reduce the growth of treated cells to half that of untreated cells) concentrations for their sensitive and resistant lines overlap (Supplementary Report 3). Our analyses used both their cell lines and ones we selected independently.

2. The lists of genes initially reported in the supplementary information on the *Nature Medicine* website<sup>1</sup> are wrong because of an 'off-

complexity of many bioinformatics analyses. This complexity requires extensive double-checking and documentation to ensure both data validity and analysis reproducibility. We believe that this situation may be improved by an approach that allows a complete, auditable trail of data handling and statistical analysis. We use Sweave<sup>4,5</sup>, a package that allows analysts to combine source code (in R)<sup>6</sup> and documentation (in LaTeX)<sup>7</sup> in the same file. Our Sweave files are available at (<http://bioinformatics.mdanderson.org/Supplements/ReproRsch-Chemo/>). Running them reproduces our results and generates figures, tables and a complete PDF manuscript.

The idea of using the NCI-60 cell lines to predict patient response to chemotherapy is exciting. Our analysis, however, suggests that it did not work here.

Kevin R Coombes, Jing Wang & Keith A Baggerly

Department of Bioinformatics and Computational Biology, University of Texas M.D. Anderson Cancer Center, Houston, Texas 77030, USA.  
e-mail:kcoombes@mdanderson.org

Note: Supplementary information is available on the *Nature Medicine* website.

# Baggerly et al. 2005. J. Natl. Cancer Inst. 97:307-309

## BRIEF COMMUNICATION

### Signal in Noise: Evaluating Reported Reproducibility of Serum Proteomic Tests for Ovarian Cancer

*Keith A. Baggerly, Jeffrey  
S. Morris, Sarah R. Edmonson,  
Kevin R. Coombes*

Proteomic profiling of serum initially appeared to be dramatically effective for diagnosis of early-stage ovarian cancer, but these results have proven difficult to reproduce. A recent publication reported good classification in one dataset using results from training on a much earlier dataset, but the authors have since reported that they did not perform the analysis as described. We examined the reproducibility of the proteomic patterns across datasets in more detail. Our analysis reveals that the pattern that enabled successful classification is biologically implausible and that the method, properly applied, does not classify the data accurately. We show that the method used in previously published studies does not establish reproducibility and performs no better than chance for classifying the second dataset, in part because the second dataset is easy to classify correctly. We conclude that the reproducibility of the proteomic profiling approach has yet to be established. [J Natl Cancer Inst 2005;97:307-9]

## Waxman & Gavrilets. 2005. Journal of Evolutionary Biology 18:1139-1154

of empirical data and the corresponding prediction of Adaptive Dynamics.

Lastly, we note that quite a lot of the work in the literature on Adaptive Dynamics relies on numerical simulation of the behaviour of populations. We observe

that there is often a somewhat sketchy description of the computational procedures adopted. To enable later workers to reproduce computational work, we would like to see clearer descriptions of this important aspect of the research, which should be viewed as being as important as an experimental protocol.



## Don't do stats or simulations? No problem!

- Mathematical biologists doing numerics are not out of the woodwork
- Numerics also needs to be reproducible

## What is $\LaTeX$ ?

*"Is  $\LaTeX$  hard to use? It's easy to use if you're one of the 2% of the population who thinks logically and can read an instructional manual. The other 98% of the population would find it very hard or impossible to use."* — Leslie Lamport

- A document preparation system for high-quality typesetting.
- First developed in 1985 by Leslie Lamport and based on Donald E. Knuth's TeX typesetting language.
- Designed for the production of technical and scientific documentation.
- Based on the idea that it is better to leave document design to document designers, and to let authors get on with writing documents.
- Automatic generation of bibliographies and indexes.
- Pronounced "Lah-tech" or "Lay-tech".

## Online resources for $\LaTeX$

- The Comprehensive  $\TeX$  Archive Network aka CTAN: the authoritative collection of materials related to the  $\TeX$  typesetting system. (<http://www.ctan.org/>)
- The Not So Short Introduction to  $\LaTeX 2_{\epsilon}$ : The unofficial manual. (<http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>)
- $\LaTeX$  at Wikibooks: wiki guide to the  $\LaTeX$  markup language. (<http://en.wikibooks.org/wiki/LaTeX>)
- The comprehensive  $\LaTeX$  symbol list: lists 2826 symbols and the corresponding  $\LaTeX$  commands and packages necessary to produce them. (<http://statweb.calpoly.edu/jdoi/web/reference/symbols-a4.pdf>)

## A first L<sup>A</sup>T<sub>E</sub>X example

```

\documentclass [12pt]{article}
\newcommand{\dsfrac}[2]{\frac{\displaystyle #1}{\displaystyle #2}}
\author{Mario Pineda-Krch}
\title{Just another tritrophic model}
\begin{document}
\maketitle
\section{The Hastings \& Powell model}
The non-dimensional version of the Hastings \& Powell (1991) model
is given by,
\begin{equation}
\begin{array}{lcl}
\dsfrac{dx}{dt} & = & x(1-x) - \dsfrac{a_1 x}{1+b_1 x}y \\
\dsfrac{dy}{dt} & = & \dsfrac{a_1 x}{1+b_1 x}y - \dsfrac{a_2 y}{1+b_2 y}z - d_1 y \\
\dsfrac{dz}{dt} & = & \dsfrac{a_2 y}{1+b_2 y}z - d_2 z
\end{array}
\end{equation}
\end{document}

```

# Just another tritrophic model

Mario Pineda-Krch

January 15, 2011

## 1 The Hastings & Powell model

The non-dimensional version of the Hastings & Powell (1991) model is given by,

$$\begin{aligned}\frac{dx}{dt} &= x(1-x) - \frac{a_1x}{1+b_1x}y \\ \frac{dy}{dt} &= \frac{a_1x}{1+b_1x}y - \frac{a_2y}{1+b_2y}z - d_1y \\ \frac{dz}{dt} &= \frac{a_2y}{1+b_2y}z - d_2z\end{aligned}\tag{1}$$

# What is R?

*"R is a language and environment for ~~statistical computing~~ computational research and graphics."* — Freely adapted from <http://www.r-project.org/about.html>

- Highly extensible via user-developed packages; 2751 of packages (only at CRAN) and counting.
- Much of R is written in R.
- Command line interface and scriptable.
- Easily integrates with low-level languages (e.g. C/C++ and Fortran).
- Source code freely available allowing for algorithm transparency and modification.
- Free as in freedom and priceless.
- Compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

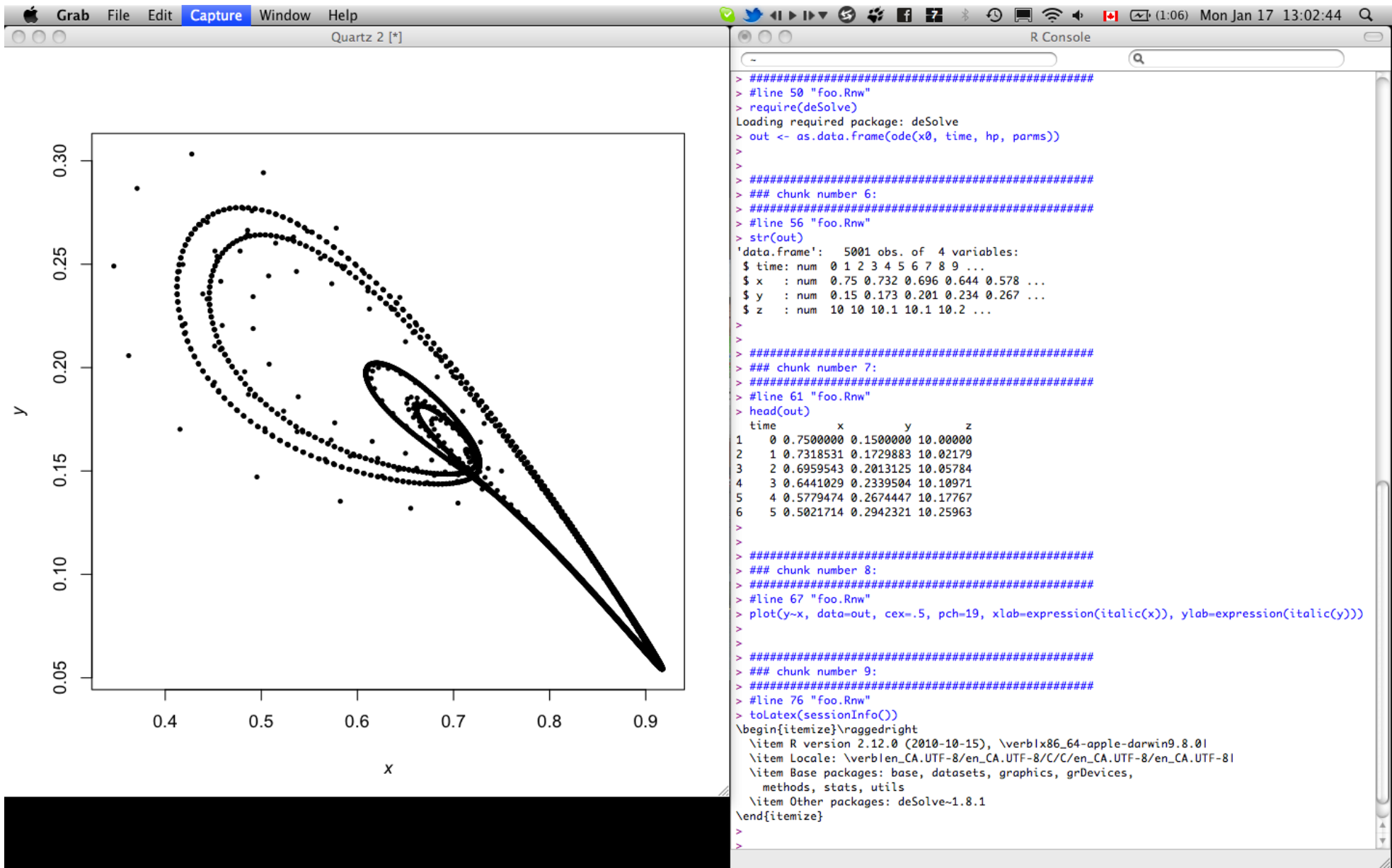
## Online resources for R

- The Comprehensive R Archive Network aka CRAN (<http://cran.r-project.org/>): a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R
- R-blogger (<http://www.r-bloggers.com/>): R news contributed by (147, and counting) R bloggers
- R-Forge (<https://r-forge.r-project.org/>): central platform for the development of R packages, R-related software and further projects.
- Stack Overflow (<http://stackoverflow.com/questions/tagged/r>): collaboratively built and maintained programming Q&A site.
- Journal of Statistical Software (<http://www.jstatsoft.org/>): publishes articles, book reviews, code snippets, and software. Lots of R stuff. Open source (i.e. free access). See in particular the Special Volume on Ecology and Ecological Modelling in R (<http://www.jstatsoft.org/v22>).

- The R Journal (<http://journal.r-project.org/>): peer reviewed journal focusing on introduction and review of packages, R programming tips and tricks, etc.
- GillespieSSA (<http://pineda-krch.com/gillespiessa/>): a package providing an interface to several stochastic simulation algorithms for generating simulated trajectories of finite population continuous-time model.



# A first (non-statistical) R example



## R + L<sup>A</sup>T<sub>E</sub>X = Sweave

*"Sweave provides a flexible framework for mixing text and S code for automatic document generation. A single source file contains both documentation text and S code, which are then woven into a final document containing the documentation text together with the S code and/or the output of the code (text, graphs)"* — Freidrich Leisch

- A function in R
- Allows for integration of code (R) with prose (L<sup>A</sup>T<sub>E</sub>X)
- A simple text file consisting of a sequence of code and documentation segment, aka *chunks*
- Extremely simple syntax— once you know R and L<sup>A</sup>T<sub>E</sub>X learning Sweave is trivial
- Enables the creation of dynamic documents
- R code is executed and the results (output, graphics) incorporated when the document is generated

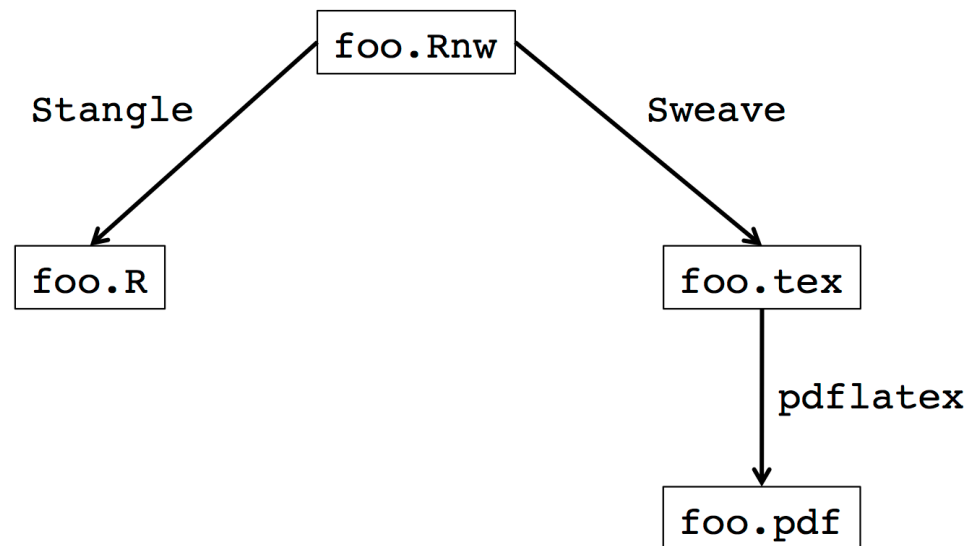
- Leverages  $\text{\LaTeX}$  typesetting capabilities and R's computational strengths
- Easy to regenerate the re-run the code and regenerate the documentation if the inout changes
- Can make computational research more transparent and reproducible — to others and to oneself

## Online resources for Sweave

- The Sweave Homepage (<http://www.stat.uni-muenchen.de/~leisch/Sweave/>): The official home of Sweave by its father, Freidrich Leisch
- Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis (<http://www.stat.uni-muenchen.de/~leisch/Sweave/Sweave-compstat2002.pdf>)
- Sweave Demo (<http://www.stat.umn.edu/~charlie/Sweave/>): short introduction to Literate Programming, Reproducible Research and Sweave

## (The no sweat) Sweave in $\approx 20$ seconds

- Create a file `foo.Rnw`
- Enter your prose ( $\text{\LaTeX}$ ) the good old fashioned  $\text{\LaTeX}$  way
- Enclose R code chunks between `<<>>=` (on a line of its own) and `@` (on a line of its own)
- To produce the documentation weave the source like so:
  - R CMD Sweave `foo.Rnw` from the shell
  - `Sweave('foo.Rnw')` from within R.



## A first Sweave example

```

\documentclass[12pt]{article}
\newcommand{\dsfrac}[2]{\frac{\displaystyle #1}{\displaystyle #2}}
\author{Mario Pineda-Krch}
\title{Just another tritrophic model}
\begin{document}
\maketitle
\section{The Hastings \& Powell model}
The non-dimensional version of the Hastings \& Powell (1991) model
is given by,
\begin{equation}
\begin{array}{lcl}
\dsfrac{dx}{dt} & = & x(1-x) - \dsfrac{a_1 x}{1+b_1 x}y \quad \quad \\
\dsfrac{dy}{dt} & = & \dsfrac{a_1 x}{1+b_1 x}y - \dsfrac{a_2 y}{1+b_2 y}z - d_1 y \quad \quad \\
\dsfrac{dz}{dt} & = & \dsfrac{a_2 y}{1+b_2 y}z - d_2 z
\end{array}
\end{equation}

Define the nondimensional system in R,
<<>>=
hp = function(time, population, parms){

```

```

x <- population[1]
y <- population[2]
z <- population[3]
with(as.list(parms),{
  dx = x*(1-x) - (a1*x)/(1+b1*x)*y
  dy = (a1*x)/(1+b1*x)*y - (a2*y)/(1+b2*y)*z - d1*y
  dz = (a2*y)/(1+b2*y)*z - d2*z
  out = c(dx, dy, dz)
  list(out)
})
}
@

```

I eyeballed the initial conditions from Figure 2 in HP91.

```
<<>>=
```

```
x0 <- c(x=0.75, y=0.15, z=10)
```

```
@
```

Declare the time vector,

```
<<>>=
```

```
time <- seq(0, 5000)
```

```
@
```

Here I am using the same parameters as in Figure 2 in HP91 (which is identical to Figure 2 in KH94),

```
<<>>=
```

```
parms <- c(a1=5.0, b1=2.5, a2=0.1, b2=2.0, d1=0.4, d2=0.01)
```

```
@
```

Solve the system numerically,

```
<<>>=
```

```
require(deSolve)
```

```
out <- as.data.frame(ode(x0, time, hp, parms))
```

```
@
```

Look at the structure of the result object,

```
<<>>=
```

```
str(out)
```

```
@
```

and the beginning of the time series,

```
<<>>=
```

```
head(out)
```

```
@
```



```
\end{document}
```

# Weaving

Sweave('foo.Rnw') produces the documentation

## Just another tritrophic model

Mario Pineda-Krch

January 17, 2011

### 1 The Hastings & Powell model

The non-dimensional version of the Hastings & Powell (1991) model is given by,

$$\begin{aligned}\frac{dx}{dt} &= x(1-x) - \frac{a_1x}{1+b_1x}y \\ \frac{dy}{dt} &= \frac{a_1x}{1+b_1x}y - \frac{a_2y}{1+b_2y}z - d_1y \\ \frac{dz}{dt} &= \frac{a_2y}{1+b_2y}z - d_2z\end{aligned}\tag{1}$$

Define the nondimensional system in R,

```
> hp = function(time, population, parms) {
+   x <- population[1]
+   y <- population[2]
+   z <- population[3]
+   with(as.list(parms), {
+     dx = x * (1 - x) - (a1 * x)/(1 + b1 * x) * y
+     dy = (a1 * x)/(1 + b1 * x) * y - (a2 * y)/(1 + b2 * y) *
+       z - d1 * y
+     dz = (a2 * y)/(1 + b2 * y) * z - d2 * z
+     out = c(dx, dy, dz)
+     list(out)
+   })
+ }
```

I eyeballed the initial conditions from Figure 2 in HP91.

```
> x0 <- c(x = 0.75, y = 0.15, z = 10)
```

Declare the time vector,

```
> time <- seq(0, 5000)
```

Here I am using the same parameters as in Figure 2 in HP91 (which is identical to Figure 2 in KH94),

```
> parms <- c(a1 = 5, b1 = 2.5, a2 = 0.1, b2 = 2, d1 = 0.4, d2 = 0.01)
```

Solve the system numerically,

```
> require(deSolve)
```

```
> out <- as.data.frame(ode(x0, time, hp, parms))
```

Look at the structure of the result object,

```
> str(out)
```

```
'data.frame':      5001 obs. of  4 variables:
 $ time: num  0 1 2 3 4 5 6 7 8 9 ...
 $ x   : num  0.75 0.732 0.696 0.644 0.578 ...
 $ y   : num  0.15 0.173 0.201 0.234 0.267 ...
 $ z   : num  10 10 10.1 10.1 10.2 ...
```

and the beginning of the time series,

```
> head(out)
```

```
  time      x      y      z
1  0 0.7500000 0.1500000 10.00000
2  1 0.7318531 0.1729883 10.02179
3  2 0.6959543 0.2013125 10.05784
4  3 0.6441029 0.2339504 10.10971
5  4 0.5779474 0.2674447 10.17767
6  5 0.5021714 0.2942321 10.25963
```

# Tangling

Stangle('foo.Rnw') produces the R code

```
#####
### chunk number 1:
#####
#line 19 "listing3.Rnw"
hp = function(time, population, parms){
  x <- population[1]
  y <- population[2]
  z <- population[3]
  with(as.list(parms),{
    dx = x*(1-x) - (a1*x)/(1+b1*x)*y
    dy = (a1*x)/(1+b1*x)*y - (a2*y)/(1+b2*y)*z - d1*y
    dz = (a2*y)/(1+b2*y)*z - d2*z
    out = c(dx, dy, dz)
    list(out)
  })
}

#####
```

```

### chunk number 2:
#####
#line 35 "listing3.Rnw"
x0 <- c(x=0.75, y=0.15, z=10)

#####

### chunk number 3:
#####
#line 40 "listing3.Rnw"
time <- seq(0, 5000)

#####

### chunk number 4:
#####
#line 45 "listing3.Rnw"
parms <- c(a1=5.0, b1=2.5, a2=0.1, b2=2.0, d1=0.4, d2=0.01)

#####

### chunk number 5:

```

```
#####
#line 50 "listing3.Rnw"
require(deSolve)
out <- as.data.frame(ode(x0, time, hp, parms))

#####
### chunk number 6:
#####
#line 56 "listing3.Rnw"
str(out)

#####
### chunk number 7:
#####
#line 61 "listing3.Rnw"
head(out)
```

## Adding figures

```
\begin{figure}[!h]
\begin{center}
<<fig=TRUE, width=7, height=7>>=
plot(y~x, data=out, cex=.5, pch=19, xlab=expression(italic(x)), ylab
=expression(italic(y)))
@
\end{center}
\caption{Phase plane plot of the resource ( $x$ ) and the predator  $y$ }
.
}
\end{figure}
```

## Just another tritrophic model

Mario Pineda-Krch

January 17, 2011

### 1 The Hastings & Powell model

The non-dimensional version of the Hastings & Powell (1991) model is given by,

$$\begin{aligned}\frac{dx}{dt} &= x(1-x) - \frac{a_1x}{1+b_1x}y \\ \frac{dy}{dt} &= \frac{a_1x}{1+b_1x}y - \frac{a_2y}{1+b_2y}z - d_1y \\ \frac{dz}{dt} &= \frac{a_2y}{1+b_2y}z - d_2z\end{aligned}\tag{1}$$

Define the nondimensional system in R,

```
> hp = function(time, population, parms) {
+   x <- population[1]
+   y <- population[2]
+   z <- population[3]
+   with(as.list(parms), {
+     dx = x * (1 - x) - (a1 * x)/(1 + b1 * x) * y
+     dy = (a1 * x)/(1 + b1 * x) * y - (a2 * y)/(1 + b2 * y) *
+       z - d1 * y
+     dz = (a2 * y)/(1 + b2 * y) * z - d2 * z
+     out = c(dx, dy, dz)
+     list(out)
+   })
+ }
```

I eyeballed the initial conditions from Figure 2 in HP91.



```
> x0 <- c(x = 0.75, y = 0.15, z = 10)
```

Declare the time vector,

```
> time <- seq(0, 5000)
```

Here I am using the same parameters as in Figure 2 in HP91 (which is identical to Figure 2 in KH94),

```
> parms <- c(a1 = 5, b1 = 2.5, a2 = 0.1, b2 = 2, d1 = 0.4, d2 = 0.01)
```

Solve the system numerically,

```
> require(deSolve)
```

```
> out <- as.data.frame(ode(x0, time, hp, parms))
```

Look at the structure of the result object,

```
> str(out)
```

```
'data.frame':      5001 obs. of  4 variables:
 $ time: num  0 1 2 3 4 5 6 7 8 9 ...
 $ x   : num  0.75 0.732 0.696 0.644 0.578 ...
 $ y   : num  0.15 0.173 0.201 0.234 0.267 ...
 $ z   : num  10 10 10.1 10.1 10.2 ...
```

and the beginning of the time series,

```
> head(out)
```

```
   time      x      y      z
1  0 0.7500000 0.1500000 10.00000
2  1 0.7318531 0.1729883 10.02179
3  2 0.6959543 0.2013125 10.05784
4  3 0.6441029 0.2339504 10.10971
5  4 0.5779474 0.2674447 10.17767
6  5 0.5021714 0.2942321 10.25963
```

```
> plot(y ~ x, data = out, cex = 0.5, pch = 19, xlab = expression(italic(x)),  
+      ylab = expression(italic(y)))
```

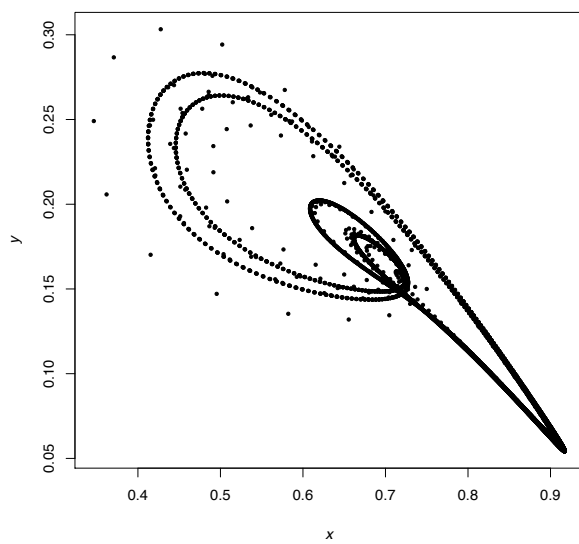
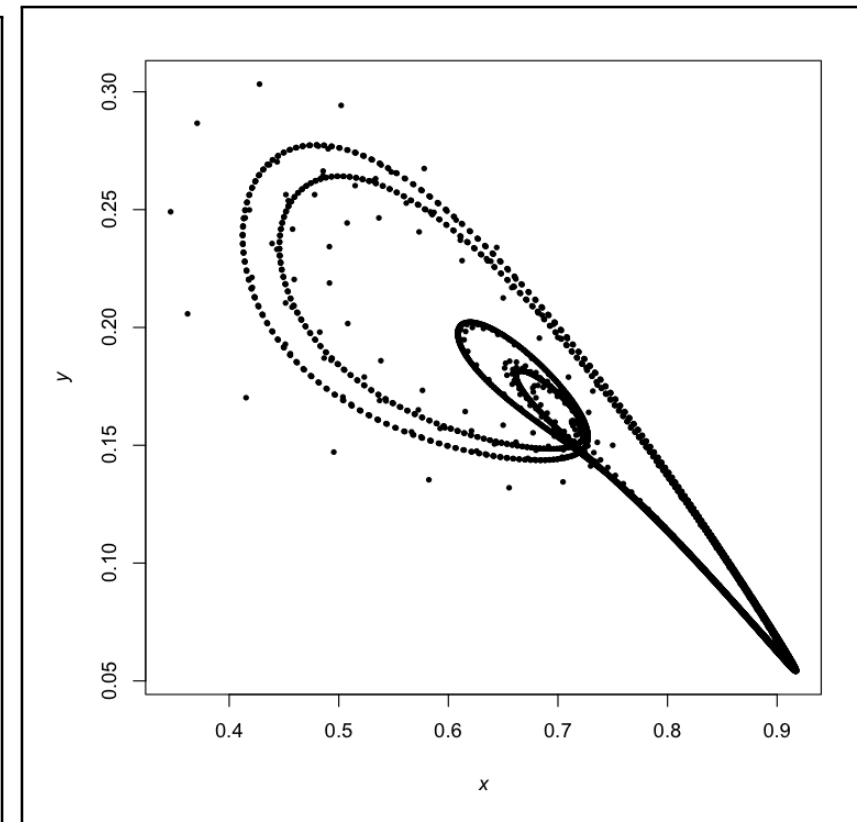
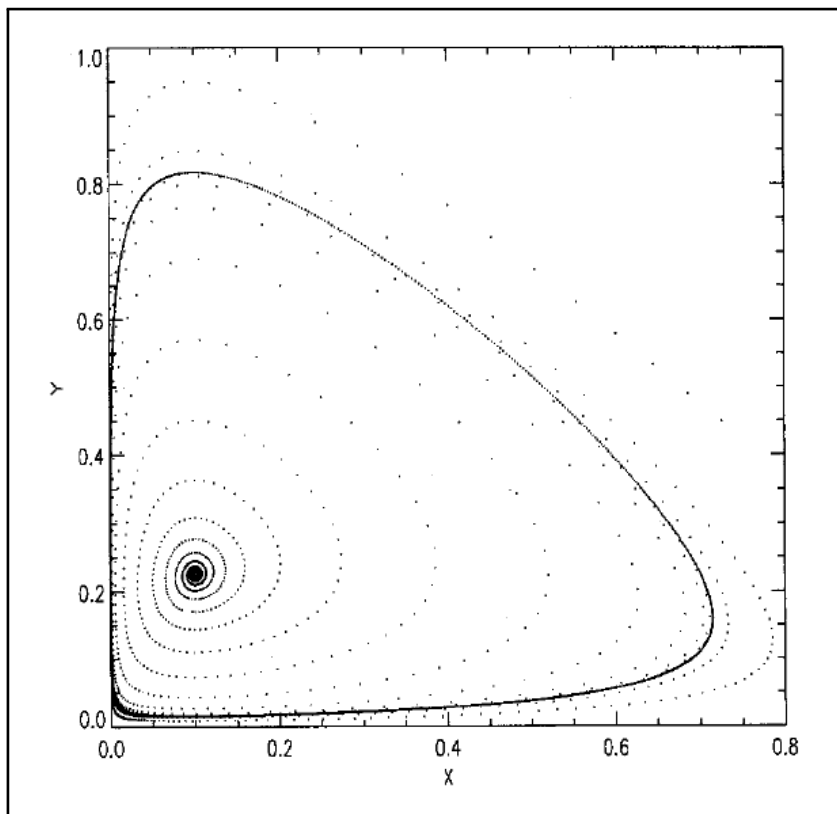


Figure 1: Phase plane plot of the resource ( $x$ ) and the predator  $y$ .

# Congratulations you have now reproduced the result in Hastings & Powell (1991)



Eh? What's going on Alan?  
 Try setting  $z = 0$  (left as an exercise for the reader)

## Inline evaluations

```
<<>>=
```

```
a <- a
```

```
@
```

We can print parameter values like this, `$a=\Sexpr{a}$`.

We can print parameter values like this,  $a = 1$ .

We can also evaluate expressions inline, e.g. `\Sexpr{sqrt(12)}` and `\Sexpr{sd(rnorm(100))}`.

We can also evaluate expressions inline, e.g. 3.46410161513775 and 1.04714290778536.

## Dealing with big data sets

- R memory hungry beast (typically limited to data sets much smaller than the available RAM)
- The OS can only access 4GB of memory on a 32 bit system, R will give you an error message at around 2GB
- Packages for dealing with big data: `ff` and `bigmemory`
- More info at CRAN Task View: High-Performance and Parallel Computing with R (<http://cran.r-project.org/web/views/HighPerformanceComputing.html>)
- Saving large data sets as `.Rdata` objects will often result in smaller files (compared to ASCII files)

## Dealing with time consuming computations

- A few simple approaches:
  - Get a faster computer!
  - Use C/C++ for the time consuming parts
  - Cache (intermediate) data

### Caching data,

```
file.name <- "foo.data"
if (!file.exists(file.name)) {

  # Perform time consuming computations and record results in
  object 'out'

  save(out, file=file.name)
} else load(file.name)
```

Caveat: do not forget to remove `foo.data` if you intend to recreate it (e.g. if the time consuming algorithm has changed).

## Dealing with complex projects

*"You must understand, young ~~Hobbit~~ R apprentice, it takes a long time to say anything in ~~Old-Entish~~ Sweave. And we never say anything unless it is worth taking a long time to say."*

— Freely adapted from Treebeard, LOTR

- Avoid monolithic Sweave files — split into smaller logical components
- Use a Makefile for build a project that consists of multiple Sweave files (<http://www.stat.auckland.ac.nz/~stat782/downloads/make-tutorial.pdf>)

# Makefile

An embarrassingly trivial Makefile,

```
foo.pdf : foo.tex
    pdflatex foo.tex ; pdflatex foo.tex

foo.tex : foo.Rnw
    R CMD Sweave foo.Rnw

foo.R : foo.Rnw
    R CMD Stangle foo.Rnw

clean :
    rm -rf *.eps *.pdf *.tex *.log *.aux
```

Why is pdflatex invoked twice? (left as an exercise for the reader)



## make examples

To generate documentation run,

```
make
```

To weave (generate  $\text{\LaTeX}$  source) run,

```
make foo.tex
```

To tangle (generate R code) run,

```
make foo.R
```

To remove junk run,

```
make clean
```

## Session information

```
<<results=tex>>=  
toLatex(sessionInfo())  
@
```

### 2 Session information

```
> toLatex(sessionInfo())
```

- R version 2.12.0 (2010-10-15), x86\_64-apple-darwin9.8.0
- Locale: C
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: deSolve 1.8.1

# Acknowledgements

- Alan Hastings for pointing out the virtues of  $z = 0$
- The Lewis Research Group for feedback and discussion
- Thank you for your time!

These slides, `foo.Rnw` and the associated `Makefile` are available at <http://pineda-krch.com/2011/01/17/the-joy-of-sweave/>



The Joy of Sweave: A Beginner's Guide to Reproducible Research with Sweave by [Mario Pineda-Krch](#) is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](#).