# Extending R

Mohammad H. Ferdosi, Cedric Gondro

# Who I am?

1. PhD – University of New England (2016)
2. Research Fellow – Animal Genetics and Breeding Unit (2015 – present)
3. Started programming 24 years ago. I have a very good experience in  C, C++, C# and R.
4. Author of hsphase (An R package submitted to CRAN – first version 2013)
5. I have been using R since 2008
6. R
    - Rcpp
    - Sweave
    - Functional programming
    - Object Oriented Programming (OOP)
    - Shiny

# Why should we extend R?

1. Large data – large problems
2. Improving read and write operations in R
3. Byte-code compiler
4. Managing memory
5. Parallel computation
6. External interfaces in R
    1. Linking R to C++
7. Using R inside other applications
8. Reporting in R
9. Building your own R package

- “S is great, but serious data analysis will always have to be done in Fortran.”

John Chamber Bell Labs management

- “Do not believe magic! understand what is going on”

Bjarne Stroustrup – Author of C++

# Integrated Development Environment (IDE)

- Rstudio
  - https://www.rstudio.com/
- StatET (Eclipse plugin)
  - https://projects.eclipse.org/projects/science.statet
  - http://www.walware.de/goto/statet
  - https://datascienceplus.com/eclipse-an-alternative-to-rstudio-part-1
- RKWard
  - https://rkward.kde.org/
- R Tools for Visual Studio
- R Language Support (jetbrain)

# Format R code

https://cran.r-project.org/web/packages/formatR/index.html

[https://yihui.name/formatR](https://yihui.name/formatR)

# Large data (Big data) – large problems

- Definition!
  - Wikipedia:
    - “Big data is an all-encompassing term for any collection of data sets so large or complex that it becomes difficult to process them using traditional data processing applications.”
  - Next-gen sequencing platforms
  - Depends on the software and hardware?!

# Improving read and write operations in R

- Common methods
  - *read.table*
  - *read.csv*
  - Check *“foreign”* package (DBF, SAS, SPSS and …)
- Efficient methods
  - *fread, fwrite (data.table)*
  - *Readr package*
  - *scan*
  - *readLines*
  - *load* and *save* - .Rdata or .rda
  - *readRDS* and *saveRDS* - .RDS
  - Or *read.big.matrix* (in *“bigmemory”* package)

# Byte-code compiler (enabled by default - R 3.4.0)

- Easy to use
- No code change

```
library(compiler)
enableJIT(3)
```

LLVM for R: Compiling toolkit for R

https://github.com/duncantl/Rllvm

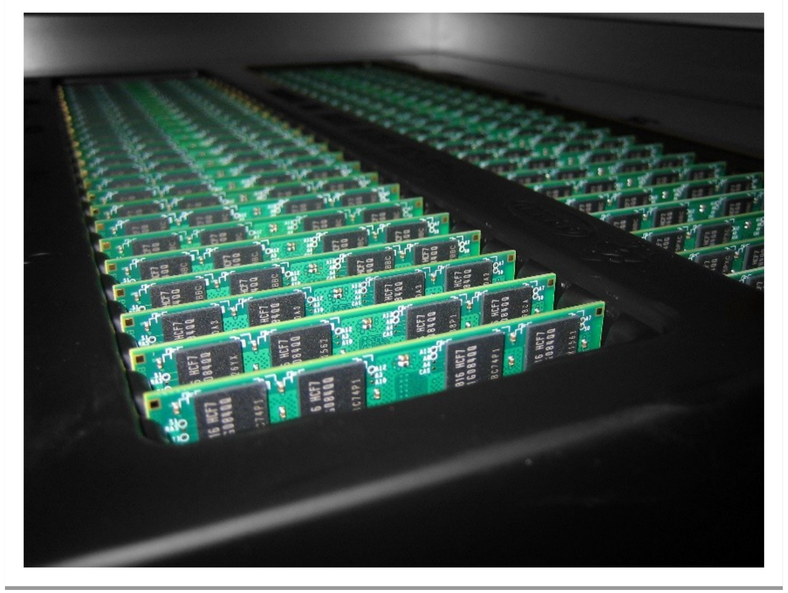
http://projecteuclid.org/download/pdfview_1/euclid.ss/1408368570

$$\bar{a} = \frac{v - v_0}{\Delta t}$$

- Should we use Julia?!

# Managing memory

- 32 or 64 bit
  - $2^{32}$ vs $2^{64}$ → 4GB vs 16 EB
- R version 3
  - Significant improvement
- Check memory usage:
  - *ll* function in the “*gdata*” package or
  - *memory.size* → windows only



# External interfaces in R

- R function for this purpose:
  - *System*
  - *System2*
    - *Example in Linux*
      - *system(“ls”)*
    - *Example in windows (for cmd internal commands)*
      - *system(“cmd /c dir”)*

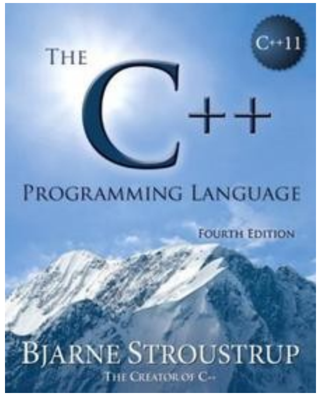
- Mixing perl, python, BASH or any program with R

# Linking R to C++

- How can we extend R?
  - Fortran, C/C++ (Rcpp – Rcpp11), C#, Java (rJava), Perl (RSPerl), Python (rPython), Latex (Sweav), Qt, GTK and ….

Java™

PROGRAMMING REPUBLIC OF PERL



python™

# Rcpp

- You should know R and C++ plus Rcpp!
- Write the program in pure C++ and then link it to R
  - Easier to debug and finding the problem
- Difference between Rcpp and standard R (Dirk Eddelbuettel, Romain Francois)

# Rcpp Example

- #include <Rcpp.h>
  - It tells the C++ compiler to include Rcpp headers
- // [[Rcpp::export]]
  - "//" means comment in C++ but not hear!
  - This function must convert to Rcpp function
- sourceCpp
  - Compile the program and link it to R
  - Piece of cake, compare to developing an R package

# Parallel computation

- CPU speed and number of cores
- Packages
  - “*snowfall*” http://www.gnu.org/software/parallel/logo.png
  - “*foreach*”

CRAN Task View: High-Performance and Parallel Computing with R (http://cran.r-project.org/).

GNU**parallel**

# Parallel computation (C++ interface)

- OpenMP (Open Multi-Processing )
  - FORTRAN and C++
- Pthreads (Portable Operating System Interface - threads)
- C++11 (Thread support library)
- RcppParallel (Threading Building Blocks)
- Threading Building Blocks (TBB)

# Using R inside other applications

- Hide R complexity from the end user
  - affylmGUI
  - AffyPipe
  - R commander
  - rattle

# Reporting in R

- Result → Report (pdf, html or … )
- Latex + R = Sweave → Rstudio, StatET, ESS and …
- R Markdown
    - ```{r}
    - head(data)

    -
      ```
- R code in Sweave:

<<>>=

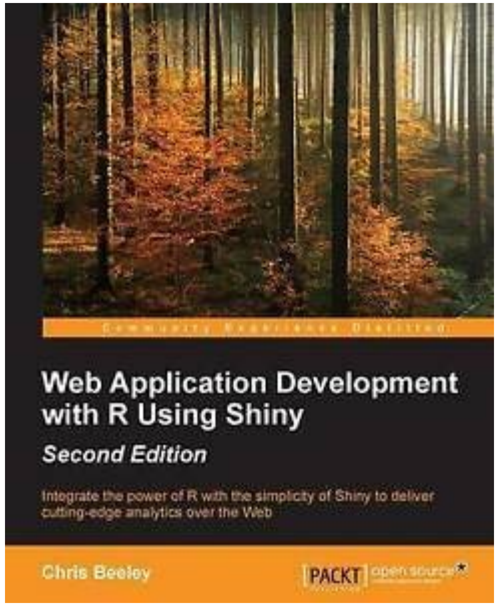
R code

@
      ```

# Reporting in R

- “*R2HTML*” package
  - *HTML*
  - *HTMLInsertGraph*

- Shiny
  - http://shiny.rstudio.com/



Community Experience Distilled

Web Application Development with R Using Shiny

*Second Edition*

Integrate the power of R with the simplicity of Shiny to deliver cutting-edge analytics over the Web

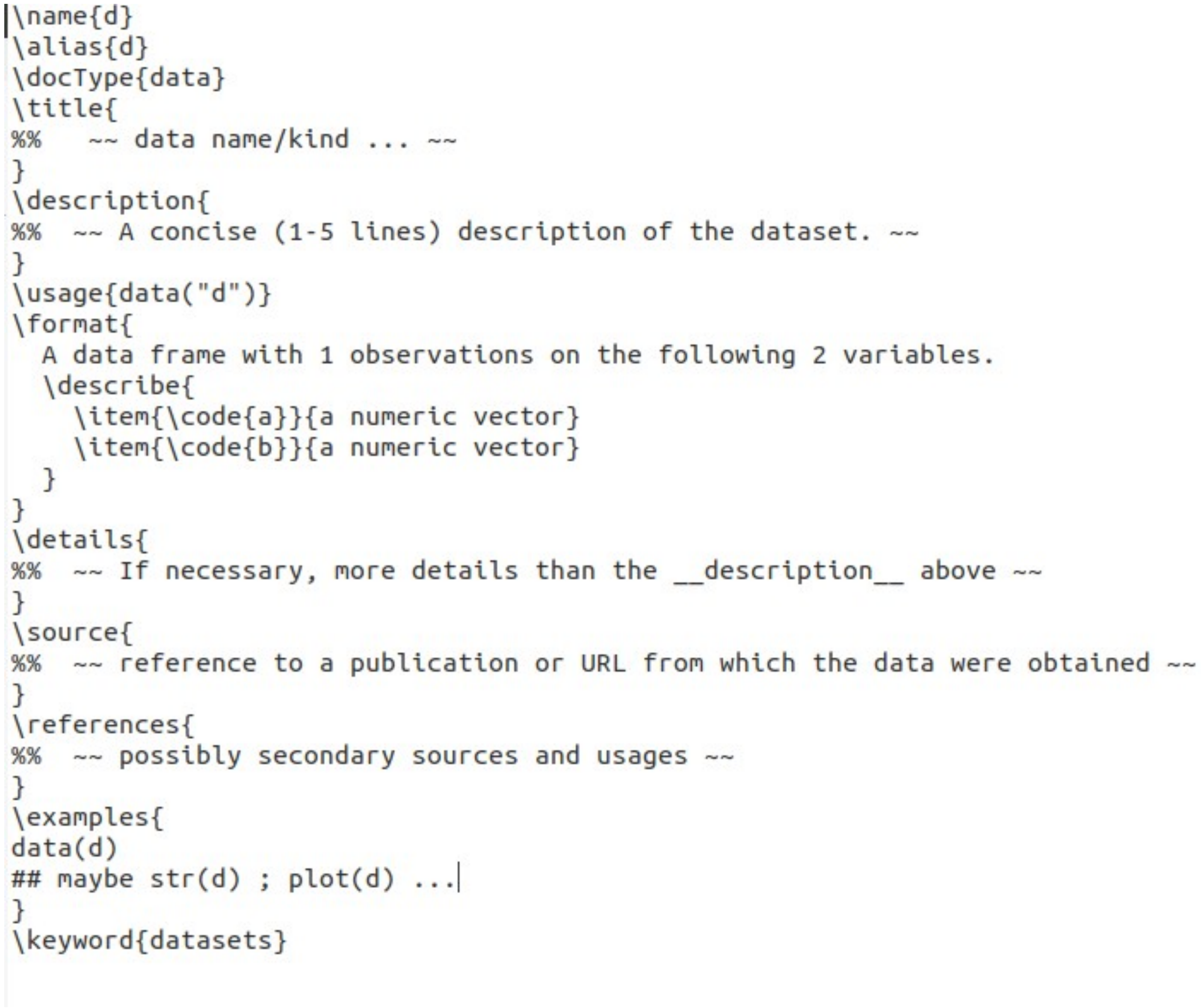
Chris Beeley

PACKT open source

# Building your own R package

- Be sure to read “Writing R Extensions” page in R project website (if you want to submit it to CRAN)
- https://cran.r-project.org/doc/manuals/r-release/R-exts.html
- Rtools
- package.skeleton
  - Simply run example(package.skeleton)
  - Check the package.skeleton help
  - Load the package
-

# R help



```
\name{d}
\alias{d}
\docType{data}
\title{
%%   ~~ data name/kind ... ~~
}
\description{
%%  ~~ A concise (1-5 lines) description of the dataset. ~~
}
\usage{data("d")}
\format{
  A data frame with 1 observations on the following 2 variables.
  \describe{
    \item{\code{a}}{a numeric vector}
    \item{\code{b}}{a numeric vector}
  }
}
\details{
%%  ~~ If necessary, more details than the __description__ above ~~
}
\source{
%%  ~~ reference to a publication or URL from which the data were obtained ~~
}
\references{
%%  ~~ possibly secondary sources and usages ~~
}
\examples{
data(d)
## maybe str(d) ; plot(d) ...
}
\keyword{datasets}
```

# Roxygen

- #' Add together two numbers
- #'
- #' @param x A number
- #' @param y A number
- #' @return The sum of \code{x} and \code{y}
- #' @examples
- #' add(1, 1)
- #' add(10, 1)
- add <- function(x, y) {
-   x + y
- }

# Files and Folders

- Description
- NAMESPACE
- Read-and-delete-me!
- data
- man (roxygen2)
- R
- Scr
- demo
- ...

# Building and installing R package

- Write the help for the functions
- R CMD check mypkg
- R CMD build  mypkg
- R CMD INSTALL  mypkg