

SER502 Project 1 (Spring 2015)

R Programming Language

Team 12

- Aditya Narasimhamurthy
- Neeraj Bahl
- Richa Mittal

History of R

- R is a language and environment for statistical computing and graphics and is highly extensible.
- R was initially written in 1991 by Robert Gentleman and Ross Ihaka—also known as "*R & R*" of the Statistics Department of the University of Auckland.
- It is a GNU project inspired by the S environment. The S language is used more for research in statistical methodology, and R provides an Open Source route to participation in that activity.
- One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed.
- R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form.

Features

R is an integrated suite of software facilities for data-manipulation, calculation and graphical display.

It includes:

- 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 on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

Features

- Like S, R is designed around a true computer language.
- Easy for users to follow the algorithmic choices made.
- For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.
- R can be extended (easily) via *packages*. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.
- R has its own LaTeX-like documentation format.

Compilers

- JIT Compilers
- JIT compilers represent a hybrid approach, with translation occurring continuously, as with interpreters, but with caching of translated code to minimize performance degradation.
- It also offers other advantages over statically compiled code at development time, such as handling of late-bound data types and the ability to enforce security guarantees.
- There are two R packages that offers Just-in-time compilation to R users:
 - the **{jit} package**, and
 - the **{compiler} package**

Compilers

- The **{jit} package**, created by Stephen Milborrow, provides just-in-time compilation of R loops and arithmetic expressions in loops, enabling such code to run much faster.
- The **{compiler} package**, created by Luke Tierney, offers a byte-code compiler for R. The advantage here is that the {compiler} package does not offer a jit compiler to a machine code, but it does offer it in order to turn it into byte code.
- The byte compiler was first introduced with R 2.13, and starting with R 2.14, all of the standard functions and packages in R were pre-compiled into byte-code. The benefit in speed depends on the specific function but code's performance can improve up by a factor of 2x times or more.

Garbage collection

- The memory allocated for R objects is not freed by the user; instead, the memory is from time to time *garbage collected*. That is, some or all of the allocated memory not being used is freed or marked as re-usable.
- If you create an R object in your C code, you must tell R that you are using the object by using the PROTECT macro on a pointer to the object. This tells R that the object is in use so it is not destroyed during garbage collection.
- There is a corresponding macro UNPROTECT that takes as argument an int giving the number of objects to unprotect when they are no longer needed. The protection mechanism is stack-based, so UNPROTECT(n) "unprotects" the last n objects which were protected.

Object Oriented System

R has 3 Object Oriented Systems:

- **S3** implements a style of OO programming called generic-function OO. This is different from most programming languages, which implement message-passing OO. Typically, this object has a special appearance in the method call, usually appearing before the name of the method/message.
- **S4** works similarly to S3, but is more formal. There are two major differences to S3.
 - S4 has formal class definitions, which describe the representation and inheritance for each class.
 - It has special helper functions for defining generics and methods.
- **Reference classes**, called RC for short, implements message-passing OO, so methods belong to classes, not functions. \$ is used to separate objects and methods.
 - RC objects are also mutable: they are modified in place. This makes them harder to reason about, but allows them to solve problems that are difficult to solve with S3 or S4.

Installation

- Install R

- Windows: <http://cran.r-project.org/bin/windows/base/>
- Mac: <http://cran.r-project.org/bin/macosx/>
- Version: 3.1.2

- RStudio Download:

- <http://www.rstudio.com/products/rstudio/download/>
- Version: 0.98.1102

R Basics

- Free open source, allows user contribution to packages
- Built in statistical functions
 - Such as `dnorm(x)`, `pnorm(q)`, `median(x)`, `sd(x)`
- Interpreted not compiled
- Assignment

> `n <- 10` or `10 -> n`

- Functions

```
Test_Function <- function()
```

```
{
```

```
...
```

```
}
```

- Help

> `?ls` or `help("ls")`

Sample Program# 1

```
# A function to return square of the input
Square <- function(x) {
  return(x^2)
}

# Function to read user input
readinteger <- function()
{
  n <- readline(prompt="Enter an integer: ")
  return(as.integer(n))
}

# Pass the user input to square function and print the value
print(Square(readinteger()))
```

Output:

Enter an integer: 5

[1] 25

Features

- Read/Write to:
 - text, csv, xml, binary files
- Database Connectivity:
 - MySQL, Oracle, SQLite, TeraData
- Integration with other environments:
 - C, JAVA, Perl, Python
- Distributed Computing:
 - EC2
- Object Oriented
 - Encapsulation
 - Polymorphism
 - Inheritance

Sample Program 2

```
par(mfrow=c(2,2))          # 2 rows, 2 columns.
# Now the next 4 pictures will be placed on these 4 regions.
# 1 --
plot(density(runif(100)), lwd=2)
text(x=0, y=0.2, "100 uniforms")    # Showing you how to place text at will
abline(h=0, v=0)
x=seq(0.01,1,0.01)
# 2 --
plot(x, sin(x), type="l")
lines(x, cos(x), type="l", col="red")
# 3 --
plot(x, exp(x), type="l", col="green")
lines(x, log(1/x), type="l", col="orange")
# 4--
plot(x, exp(-x), lwd=1, col="red")
```

Sample Program# 3

```
setClass(  
  "Person",  
  representation(name="character", age="numeric"),  
  prototype(name=NA_character_, age=NA_real_)  
)  
setClass("Engineer",  
  representation(language="character"),  
  contains="Person")  
setGeneric(  
  "profile",  
  function(object) {  
    standardGeneric("profile")  
  }  
)
```

```
setMethod(  
  "profile",  
  signature("Engineer"),  
  function(object) {  
    paste("Works on",  
      paste(object@language, collapse=" "))  
  }  
)  
User1 <- new("Engineer",  
  name="User1",  
  age=74,  
  language=c("JAVA"))  
  
print (talent(donald))
```

Schematic View

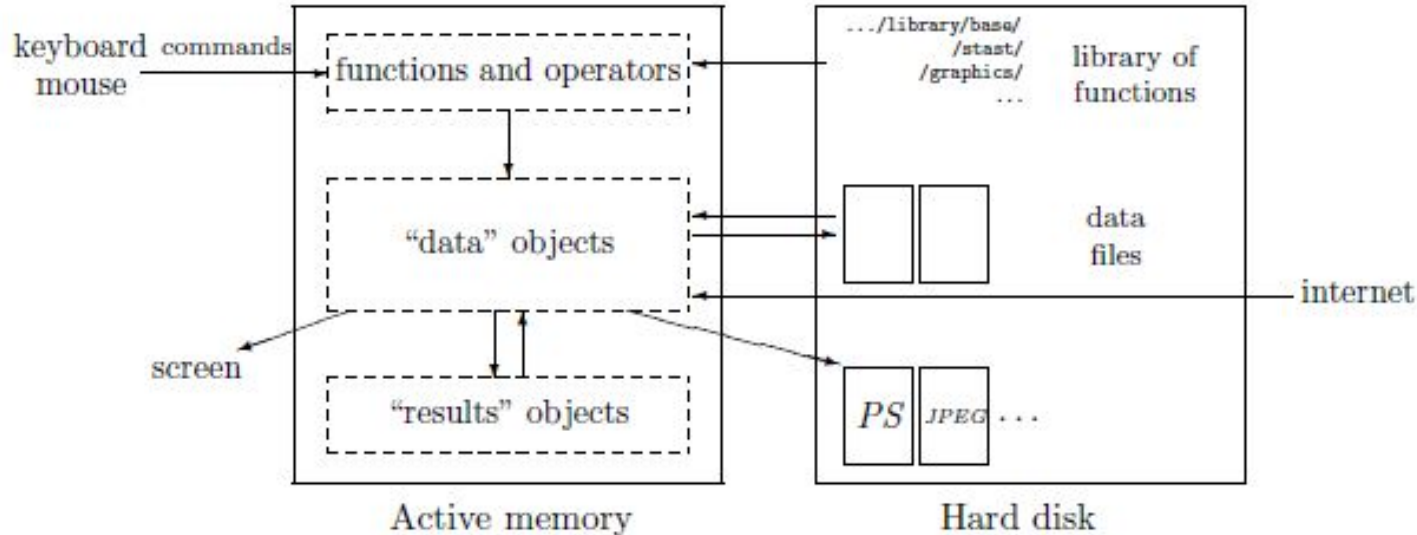


Figure 1: A schematic view of how R works.

Applications of R

- **Statistical** Computation and Graphics
- Flexible to apply to any number and domain of applications as **packages (R extensions)** can be developed, installed and used as per specific requirements.
- Domains/Applications for R are based on the packages that Users **contribute** to the public or you develop for your requirement.
- As of 2/15/2015, 08:10 MST, there are **6317 packages** on the R CRAN website (Yes, it's impressively active community, hence the timestamp! Inspiring!)

Applications of R cont...

(Categories)

- Statistical Analysis
 - Scientific
 - Genetics
 - Survival
 - Social
 - Spatio-Temporal Data
 - Psychometric
 - Phylogenetics
 - Pharmacokinetic
 - Time Series
- Graphics
- High Performance Computing
- Clinical Trials
- Machine Learning
- Natural Language Processing
- Optimization and Mathematical Programming
 - Differential Equations
 - Distributions
 - Numerical Mathematics
- Bayesian Inference

Applications cont...

Package - Set of Functions and Datasets

Package Installation:

```
install.packages("package_name")
```

```
library("package_name")
```

Some of the notable Organisations (Non Commercial) that use R

- Genomics - Bioconductor Project - 934 *packages!*
- Omega Project for Statistical Computing
- FDA (Food and Drug Administration)
- Benetech
- Human Rights Data Analysis Group
- Consumer Financial Protection Bureau
- National Weather Service

Some of the notable Organisations (Commercial) that use R

- Google
- Facebook
- Ford Motor Company
- Microsoft
- John Deere
- Lloyds
- Mozilla
- New York Times
- Nordstrom
- Uber
- Thomas Cook
- Twitter
- Trulia
- TechCrunch
- OK Cupid

Context

- R is influenced by S and Scheme
 - Similar in appearance to S
 - Underlying implementation and semantics are derived from *Scheme*

Comparison (What makes R unique)

	SAS	R	SPSS
License	Proprietary	Free and Open Source (GNU General Public License)	Proprietary
Developers	SAS Institute	R Development Core Team (Public can contribute)	IBM Corporation
Ease of Learning	Easy	Difficult (However, the Open Source Community is creating high quality tutorials)	Easy
Customization	Limited	Ability to create custom packages increases the scope of applications	Limited

Comparison cont... (Popularity)

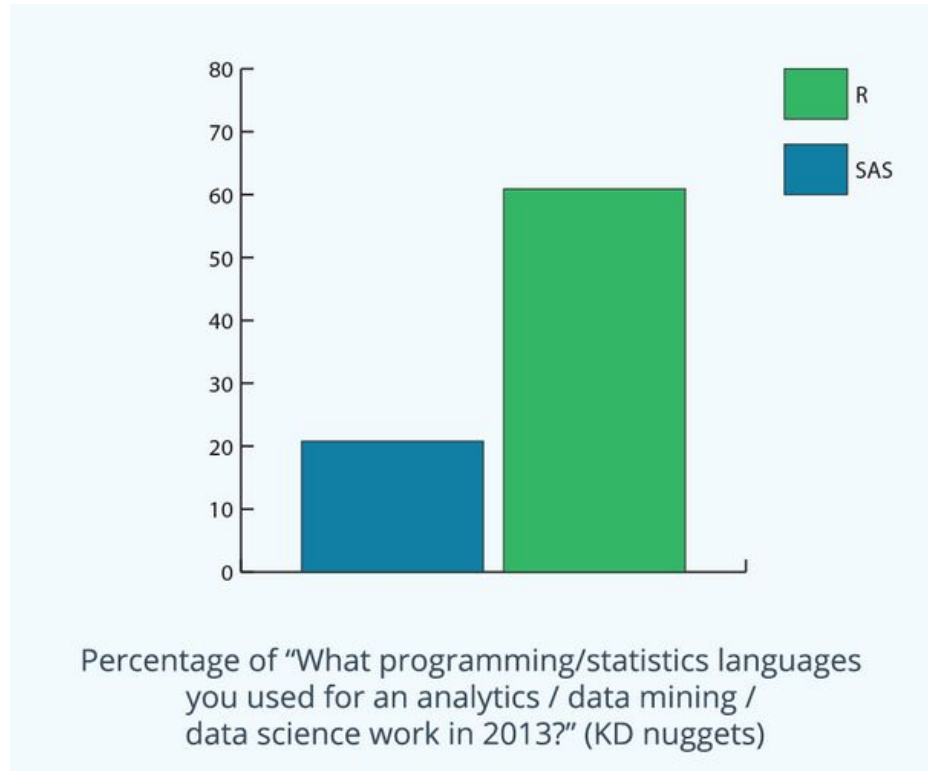


50% of Kaggle winners use R



The number of R related posts on Stack Overflow is more than 7-fold the number of posts on SAS

Comparison cont... (Popularity)



Thank You



References

- 1) <http://www.r-project.org/>
- 2) <http://cran.r-project.org/>
- 3) <https://www.coursera.org/course/rprog>
- 4) <http://www.revolutionanalytics.com/companies-using-r>
- 5) <http://datacamp.wpengine.com/wp-content/uploads/2014/05/infograph.png>
- 6) <http://www.mayin.org/ajayshah/KB/R/>
- 7) <http://www.statmethods.net/management/functions.html>
- 8) http://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf
- 9) <http://www.dartmouth.edu/~jacko/v2/2014/01/an-interview-with-the-man-who-can-only-speak-in-questions/> (Image of Questions on last slide)
- 10) <http://en.wikipedia.org>