Starting That First R Project: Managing the Learning Curve

Munawar Cheema
Munawar.Cheema@mbs.ac.uk

2013-02-07 Thursday

Outline

1. Introduction
2. Users versus Programmers
3. Considerations for Data
4. Managing Dependencies
5. Exploratory Programming
6. Literate Programming
7. Conclusion

Target Audience

Reasons for diving into R

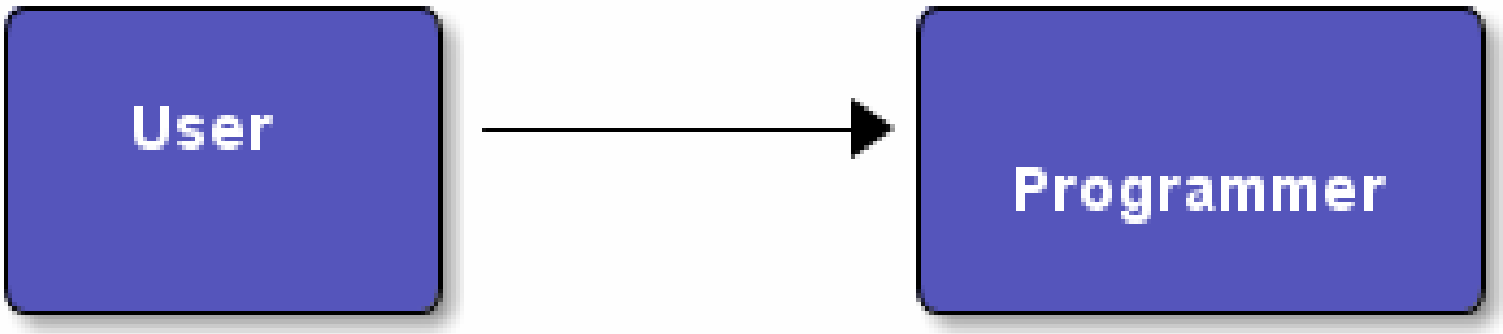
- Someone who may have been a casual user of R for simple charting and regression analysis looking to delve deeper
- Looking to start a new data intensive project and considering R
- A hobbyist looking to do something with data in a more intensive fashion
- Proof of concept at work to use R for a situation it is ideally suited for rather than a general purpose programming language

My Background—Sink or Swim

Market Microstructure

- Advisers are Prof. Mike Bowe and Prof. Stuart Hyde of Manchester Business School
- Agricultural futures markets
- Confronted with a data set of almost 10 Million records across several hundred files, I turned to R
- Have been fascinated by this wonderful programming language ever since
- Summarize, aspects of R programming that affect early *decisions*

Section 2: Users versus Programmers



Programmer or Statistician?

- John Chambers talks about a continuum from a pure user to a programmer of a statistical system.
- Learning R allows us to choose a point on this *continuum*

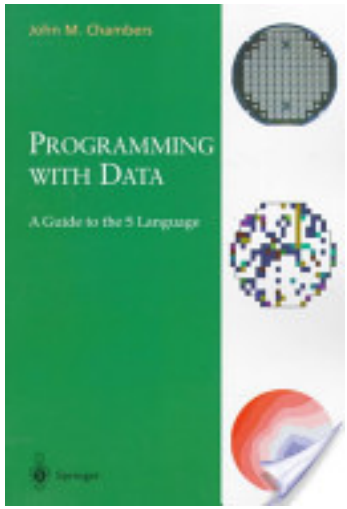


Figure : "The Green Book"

A deep look at R from a Programmer's Viewpoint

- Mission as users and creators of statistical software, is to enable the best and most thorough exploration of the data possible.
- This is tempered by the Prime Directive that computations be understandable and trustworthy.

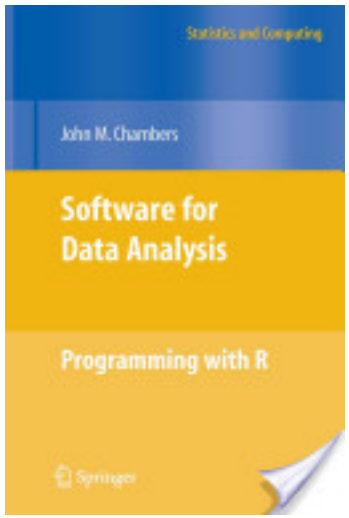


Figure : "Chambers' Masterpiece on R"

Section 3: Considerations for Data

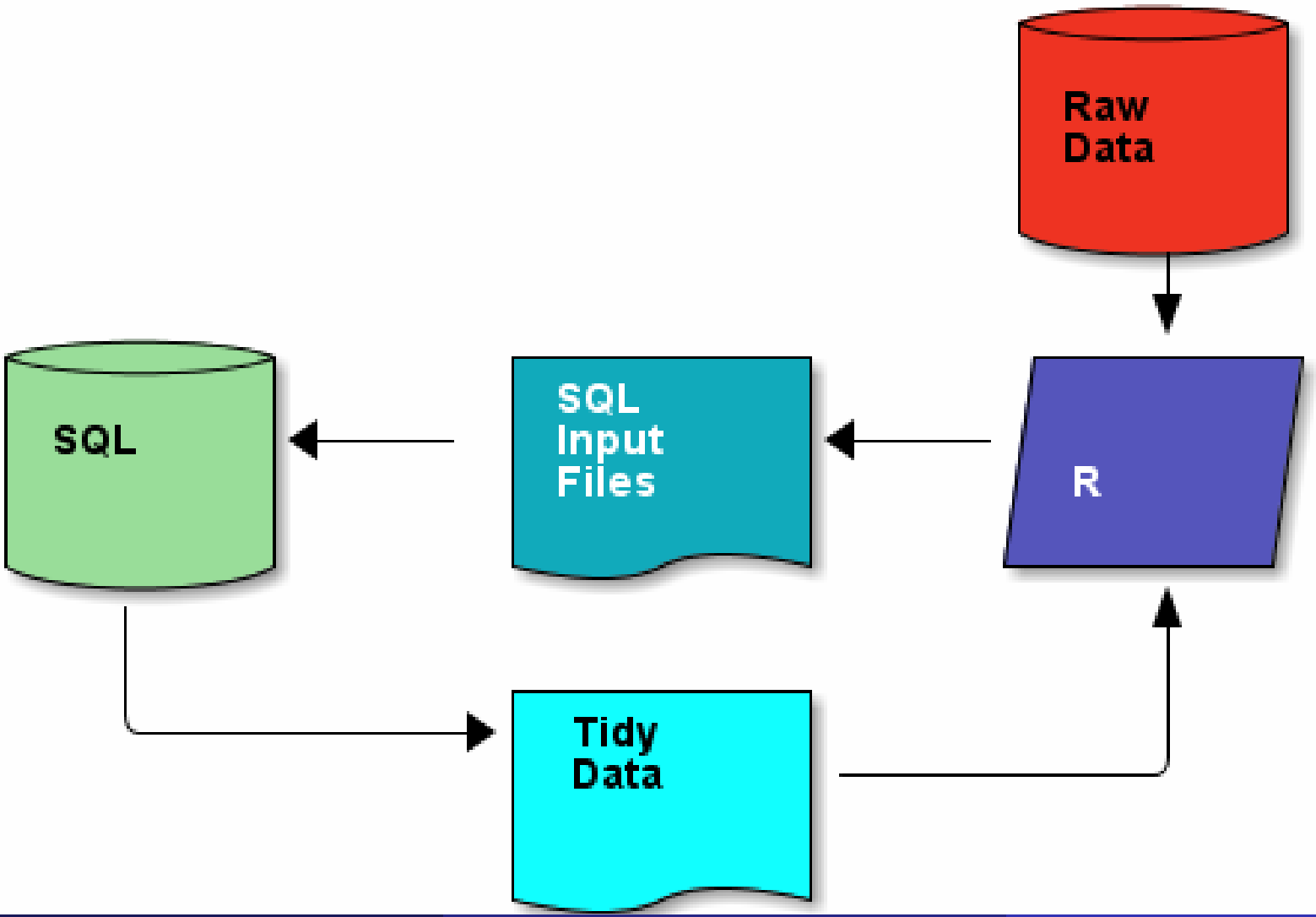
In many cases cleaning data is the most time consuming task for a statistician!

"It is often said that 80% of the effort of analysis is in data cleaning. Despite the amount of time it takes, there has been little research on how to clean data well. This paper attempts to tackle a small, but important, subset of data cleaning: data tidying.

Tidy data is easy to manipulate, model and visualise, and has a specific structure: variables are stored in columns, observations in rows, and one type of experimental unit per file. This structure makes it easy to tidy messy data, because only a small set of tools are needed to deal with a large number of messy data sets."

—Hadley Wickham, Tidy Data, 2011

Tidy Data—Solution for my data set



Data Driven Programming

```
LOAD DATA INFILE
'C:/home/mcheema/ag-project/2009-Project-1/sql-create-cbot/cbot-tick-data/corn-futures/C_FCBOT_O_tick_
Q1-1995.csv'
INTO TABLE CBOT_Tick FIELDS TERMINATED BY "," LINES TERMINATED BY "\n"
IGNORE 3 LINES
(@var1, @var2,Symbol,MonthCode,YearCode,TradeType,TradePrice,@var3)
SET TradeDate = str_to_date(@var1,'%d-%b-%Y'),
TradeTime = str_to_date(@var2,'%H:%i:%s'),
BarNo=timetobarno(str_to_date(@var2,'%H:%i:%s'));
```

Figure : "R for Data Driven Programming"

```
paste0(c("C","W","O","S"),
       "_FCBOT_tick_Q",
       as.numeric(sapply(1:4,
                         function (q) rep(q,4))),
       "_",
       as.numeric(sapply(1995:1998,
                         function (yyyy) rep(yyyy,16))))
```

Data Driven Programming (Cont.)

```
+ + + + +  [1] "C_FCBOT_tick_Q1_1995" "W_FCBOT_tick_Q1_1995" "O_FCBOT_tick_Q1_1995"
[4] "S_FCBOT_tick_Q1_1995" "C_FCBOT_tick_Q2_1995" "W_FCBOT_tick_Q2_1995"
[7] "O_FCBOT_tick_Q2_1995" "S_FCBOT_tick_Q2_1995" "C_FCBOT_tick_Q3_1995"
10] "W_FCBOT_tick_Q3_1995" "O_FCBOT_tick_Q3_1995" "S_FCBOT_tick_Q3_1995"
13] "C_FCBOT_tick_Q4_1995" "W_FCBOT_tick_Q4_1995" "O_FCBOT_tick_Q4_1995"
16] "S_FCBOT_tick_Q4_1995" "C_FCBOT_tick_Q1_1996" "W_FCBOT_tick_Q1_1996"
19] "O_FCBOT_tick_Q1_1996" "S_FCBOT_tick_Q1_1996" "C_FCBOT_tick_Q2_1996"
22] "W_FCBOT_tick_Q2_1996" "O_FCBOT_tick_Q2_1996" "S_FCBOT_tick_Q2_1996"
25] "C_FCBOT_tick_Q3_1996" "W_FCBOT_tick_Q3_1996" "O_FCBOT_tick_Q3_1996"
28] "S_FCBOT_tick_Q3_1996" "C_FCBOT_tick_Q4_1996" "W_FCBOT_tick_Q4_1996"
31] "O_FCBOT_tick_Q4_1996" "S_FCBOT_tick_Q4_1996" "C_FCBOT_tick_Q1_1997"
34] "W_FCBOT_tick_Q1_1997" "O_FCBOT_tick_Q1_1997" "S_FCBOT_tick_Q1_1997"
37] "C_FCBOT_tick_Q2_1997" "W_FCBOT_tick_Q2_1997" "O_FCBOT_tick_Q2_1997"
40] "S_FCBOT_tick_Q2_1997" "C_FCBOT_tick_Q3_1997" "W_FCBOT_tick_Q3_1997"
43] "O_FCBOT_tick_Q3_1997" "S_FCBOT_tick_Q3_1997" "C_FCBOT_tick_Q4_1997"
46] "W_FCBOT_tick_Q4_1997" "O_FCBOT_tick_Q4_1997" "S_FCBOT_tick_Q4_1997"
49] "C_FCBOT_tick_Q1_1998" "W_FCBOT_tick_Q1_1998" "O_FCBOT_tick_Q1_1998"
52] "S_FCBOT_tick_Q1_1998" "C_FCBOT_tick_Q2_1998" "W_FCBOT_tick_Q2_1998"
55] "O_FCBOT_tick_Q2_1998" "S_FCBOT_tick_Q2_1998" "C_FCBOT_tick_Q3_1998"
58] "W_FCBOT_tick_Q3_1998" "O_FCBOT_tick_Q3_1998" "S_FCBOT_tick_Q3_1998"
61] "C_FCBOT_tick_Q4_1998" "W_FCBOT_tick_Q4_1998" "O_FCBOT_tick_Q4_1998"
64] "S_FCBOT_tick_Q4_1998"
```

Figure : "R for Data Driven Programming—Output"

Section 4: Managing Dependencies

"... software is only sustainable if it can stay alive in the minds of its developers"

—Christophe Rhodes, Lisp hacker

Managing Dependencies: A Word About Free Software

R is *Free Software*

- There is value to having a community owned compiler toolchain, browser, operating system, etc
- Do not conflate "Free Software" with the need for software developers to earn a living
- Understand the motivations for many open source projects and watch out for "Tom Sawyerism"
  - Like minded individuals working together on a shared goal
  - A business or developer whose core compentency lies elsewhere will often open source code needed for its infra-structure
  - Part of an academic investigation

Managing Dependencies: Prefer Base and Recommended R Packages

Base R Packages

- base
- compiler
- datasets
- graphics
- grDevices
- grid
- methods
- splines
- stats4

Recommended Packages

- KernSmooth
- MASS
- Matrix
- boot
- class
- cluster
- codetools
- lattice
- mgcv
- nlme
- nnet

Managing Dependencies, Example I:

Choice of data manipulation and representation tools:

- MySQL an external database
- RMySQL an external package
- data.frame, readLines() and other base R tools
- data.table, bigmemory, RHadoop and other external packages

Managing Dependencies Examples II:

Sometimes there is little downside to adding a dependency:

- XML Many alternative possibilities if the dependency breaks and the convenience of the package is tremendous
- RJSON Similar to the XML package albeit to a lesser extent
- vars Vector Autoregressive modelling (VAR) Further Dependencies:
  - MASS
  - strucchange
  - lmtest
  - urca
  - sandwich

Example III: Mind The Dependencies

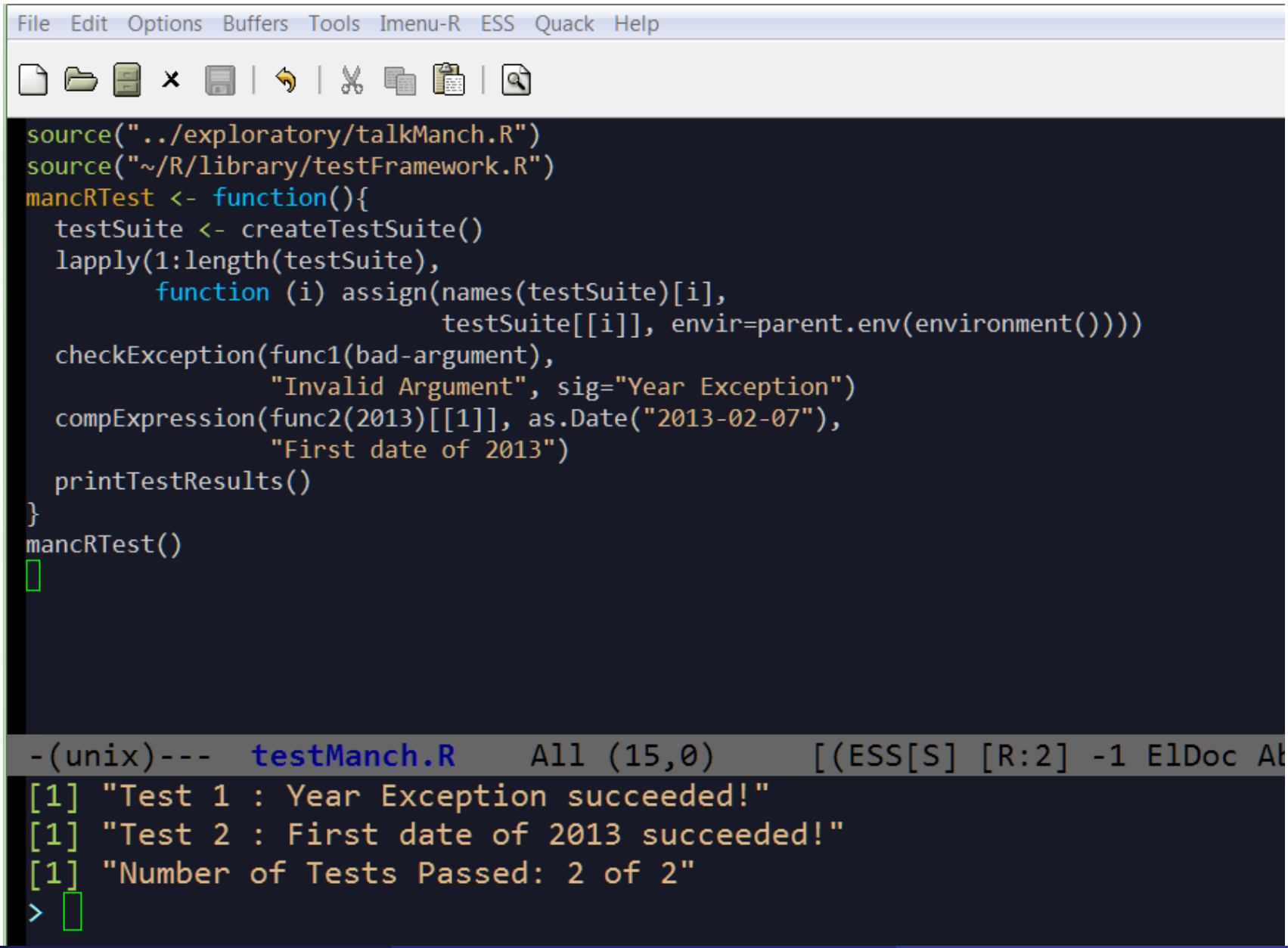
```
/usr/include/stdlib.h:81:7: note: expected 'void *' but argument is of type 'const char *'
tikzDevice.c: At top level:
tikzDevice.c:2002:17: warning: 'contains_multibyte_chars' defined but not used
c:/PROGRA~1/R/R-215~1.2/etc/x64/Makeconf:172: recipe for target `tikzDevice.o' failed
make: *** [tikzDevice.o] Error 1
ERROR: compilation failed for package 'tikzDevice'
* removing 'x:/home/mcheema/R/library/tikzDevice'

The downloaded source packages are in
        'C:\Users\mcheema\AppData\Local\Temp\RtmpaKjMtg\downloaded_packages'
Warning message:
In install.packages("tikzDevice", repos = "http://R-Forge.R-project.org",  :
  installation of package 'tikzDevice' had non-zero exit status
>
```

Figure : tickzDevice Package Long Standing Support Ends

Use R for Everything Possible

"No-one became a great artist by studying easels and paint brushes"
—Peter Norvig, Director of Research, Google

Test Driven Programming: Workflow and the REPL



Section 6: Literate Programming:

What types of reports and output do you need?

- Reproducible Data
- Formatted code in Latex or other word processor
- Graphs and Plots with mathematical notation
- Sweave which is part of base R
- Knittr an added dependency but very popular
- Emacs and Org-Mode good for multi-language

Literate Programming: Start Simple

Choose a set of tools and stick with them

- Google Style Guide contains useful recommendations
  - Line Length 80 characters
  - Consistent variable naming scheme
  - Prefer S3 over S4-controversial but I concur
  - Proper Indentation
- Latex Listings package or similar for your editor
  - Simply cut and paste
- Base R and Lattice graphics with the plotmath library
  - Side benefit deeper understanding of R

Conclusion

Decisions for *that* first project

- Choose an appropriate destination on the user -> programmer continuum
- Choose the type of reports, plots, and tables you want to produce
- Determine a set of tools and packages that meet these choices and minimize dependencies
- Avoid, noise from the blogosphere and the lure of a multitude of packages and focus on mastery of the chosen set of tools
- Constantly, improve workflow and use *R functions* wherever possible
- Prosper :)