Reproducible Research and R Workflow

Melboure R Users Group (melbURN)

Jeromy Anglim

Psychological Sciences, University of Melbourne

1st December 2010

`jeromyanglim.blogspot.com`

Outline

1 Introduction

2 Workflow

3 Tools
- Eclipse and StatET
- Version Control
- make and makefiles
- Sweave and LaTeX

4 Sweave Examples
- 1. Console Report
- 2. Multiple Reports
- 3. Database Driven Document
- 4. Non-console Report

Quote from John Chanmbers

The Mission:
Enable the best and most thorough exploration of data possible.
...
The Prime Directive:
The computations and the software for data analysis should be trustworthy.

Source: John M. Chambers, Chapter 1, *Software For Data Analysis: Programming with R*

What is the End Product?

- Report
  - Console displayed versus no console displayed
  - Batch versus once off
- Data:
  - Cleaned
  - Processed
  - Documented
- Data anlysis software:
  - R Package
  - A model

Focus of this talk

- A workflow for writing reproducible data driven reports

The Initial Challenge for the R Learner

How should you

- divide a project into files and folders?
- incorporate R analyses into a report?
- convert default R output into publication quality tables, figures, and text?
- build the final product?
- sequence the analyses?
- divide code into functions?

i.e., How do you efficiently achieve the Mission and fulfill the Prime Directive?

David Smith's Tips on R Workflow

- *Transparency*: Logical organisation of units
- *Maintanability*: Standardisation, clear comments
- *Modularity*: DRY Principle, Discrete units
- *Portability*: Relative paths, minimise dependencies, dependencies are clear
- *Reproducibility*: Easy to reproduce results
- *Efficiency*: Easy to maintain and modify

Source: `http://blog.revolutionanalytics.com/2010/10/a-workflow-for-r.html`

Josh Reisch LCFD Model

1. load.R
2. clean.R
3. func.R
4. do.R

Source: `http://stackoverflow.com/questions/1429907/workflow-for-statistical-analysis-and-report-writing/1434424`

John Myles White and ProjectTemplate

Best practice ideas

- Efficient creation of new projects
- Standardised folder and file structure (i.e., `data`, `diagnostics`, `doc`, `graphs`, `lib`, `logs`, `profiling`, `reports`, `tests`)
- Automatic data loading
- README and TODO files
- Encourages unit testing
- Standardised location of `library()` statements
- and more . . .

ProjectTemplate

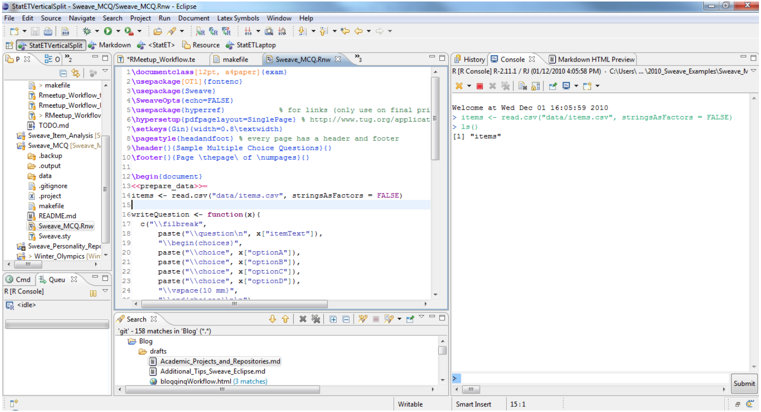
```
install.packages(’ProjectTemplate’)
library(’ProjectTemplate’)
?ProjectTemplate

create.project(’my-project’)

setwd(’my-project’)

load.project()
```

See also `http://www.johnmyleswhite.com/notebook/2010/08/26/projecttemplate/`

R Programming Environments

- Rgui
- Emacs + ESS
- Eclipse + StatET
- Any text editor + command line
- and many more . . .

Eclipse and StatET: Screenshot



```
\documentclass[12pt, a4paper]{exam}
\usepackage[OT1]{fontenc}
\usepackage{Sweave}
\SweaveOpts{echo=FALSE}
\usepackage{hyperref}            % for links (only use on final pri
\hypersetup{pdfpagelayout=SinglePage} % http://www.tug.org/applicat
\setkeys{Gin}{width=0.8\textwidth}
\pagestyle{headandfoot} % every page has a header and footer
\header{}{Sample Multiple Choice Questions}{}
\footer{}{Page \thepage\ of \numpages}{}

\begin{document}
<<prepare_data>>=
items <- read.csv("data/items.csv", stringsAsFactors = FALSE)

writeQuestion <- function(x){
  c("\\filbreak",
      paste("\\question\n", x["itemText"]),
      "\\begin{choices}",
      paste("\\choice", x["optionA"]),
      paste("\\choice", x["optionB"]),
      paste("\\choice", x["optionC"]),
      paste("\\choice", x["optionD"]),
      "\\vspace{10 mm}",
```

```
Welcome at Wed Dec 01 16:05:59 2010
> items <- read.csv("data/items.csv", stringsAsFactors = FALSE)
> ls()
[1] "items"
```

Eclipse and StatET: Benefits

- Good support for version control
- Easy to hook in external tools like sh, cmd, and make
- File search
- Allows for multiple integrated consoles
- Configurable multi-element display (particularly good on big monitors)

- Understands R (indentation, colour coding, code folding, outline view)
- Great shortcut keys for sending R code to console and getting help
- Understands Sweave and LaTeX
- Project explorer for projects, folders, files
- R object explorer and content assist
- Command history and Queue

Eclipse and StatET: Resources

- StatET Website:

  `http://www.walware.de/goto/statet`

- Longhow Lam's Guide:

  `http://www.splusbook.com/RIntro/RCourseMaterial.html`

- My Guide:

  `http://jeromyanglim.blogspot.com/2010/02/getting-started-with-sweave-r-latex.html`

Version Control: Practical Benefits

- Rewind a project or a file to a previous state (encourages experimentation)
- Provides a record of changes
- Facilitates collaboration
- Facilitates backup
- Shows changes between files
- Facilitates code sharing and reproducibility

Version Control: Conceptual Benefits

- the distinction between source and derived files
- the nature of dependencies:
  - dependencies between elements of code
  - dependencies between files within a project
  - and dependencies with files and programs external to the repository
- the nature of a repository and how repositories should be divided
- the nature of committing and documenting changes and project milestones

Git: A Version Control System

- Popular
- Github
- Experts (e.g., Handley Wickham, Linus Torvalds)

EGit: A Git plugin for Eclipse

- Simple graphical interface integrated with Eclpise
- Good for getting started with version control

Tutorial on Getting Started:

`http://jeromyanglim.blogspot.com/2010/11/getting-started-with-git-egit-eclipse.html`

make and makefiles

- One-click build
- Efficient build
- Reliable build
- Separate source from derived files
- Clean derived files
- Run alternative builds
- Encourages clear thinking about dependencies

Tutorial on getting started:

`http://jeromyanglim.blogspot.com/2010/11/makefiles-for-sweave-r-and-latex-using.html`

Example makefile

```
output = .output
rnwfile = Sweave_MCQ
backup = .backup

all:
    R CMD Sweave $(rnwfile).Rnw
    -mkdir $(output)
    -cp *.sty $(output)
    -mv *.tex *.pdf *.eps $(output)
    cd $(output); texify --run-viewer --pdf $(rnwfile).tex

clean:
    -rm $(output)/*

backup:
    -mkdir $(backup)
    cp $(output)/$(rnwfile).pdf $(backup)/$(rnwfile).pdf
```

Sweave

- Weave S (i.e., R) code chunks with LaTeX in a single self-describing document.

Key Benfits

- Reproducibility
- Efficiency
- Reliability
- Education & Communication

- Manual: `http://www.stat.uni-muenchen.de/~leisch/Sweave/`
- My guide to getting started: `http://jeromyanglim.blogspot.com/2010/02/getting-started-with-sweave-r-latex.html`

Overview of Examples

Different Types of Sweave Documents

- Console Report
- Multiple Reports
- Database Driven Document
- Non-console Report

For each example links are provided to complete copies of source code with explanation.

1. Console Report: Item Analysis

- http://jeromyanglim.blogspot.com/2010/11/sweave-tutorial-3-console-input-and.html

2. Multiple Reports: Personality Feedback

- `http://jeromyanglim.blogspot.com/2010/11/sweave-tutorial-2-individual.html`

3. Database Driven Document: Multiple Choice Questions

- `http://jeromyanglim.blogspot.com/2010/11/sweave-tutorial-using-sweave-r-and-make.html`

4. Non-console Report: Winter Olympic Medals

- `https://github.com/jeromyanglim/Sweave_Winter_Olympics`