Research Methods in Political Science I

3. Reproducible Research

Yuki Yanai

School of Law and Graduate School of Law

October 21, 2015

KOBE UNIVERSITY

Today's Menu

1. Reproducible Research
   - Reproducibility
   - Make Your Research Reproducible
2. Conducting Reproducible Research with R and RStudio
   - Document, Document, Document!
   - High Readability Coding
   - Introduction to Literate Programming

What Is Reproducible Research?

Reproducible research

- Research that can be reproduced by the other researchers.
- Publish the results with data and computer codes used in the research

- Research transparency
- Deeper understanding of research

Replication and Reproduction

1. The same research findings can be obtained by others using the same data and methods: **reproduction**
2. The same research findings can be obtained by applying the same methods to other (similar) data: **replication**

- Scientific research should satisfy both 1 and 2
- Without 2, research is falsified (normal research)
- Without 1, research is not considered scientific (pseudo science or no science)
- Some controversies

What Should We Do?

- Document your research procedure in detail
- Prepare not only the results (research paper) but also the documents recording the research procedure
- Publish your data sets
- Publish your computer codes to get the results
  - You must write codes that are highly readable
  - You must have write a lot so that others can understand what you did

Merits of Reproducible Research

- Sophisticate work flow
- Facilitate collaboration
- Get more citations (especially if you publish you data)
- Enhance the accumulation of knowledge
- Service to the research community ($\rightarrow$ higher reputation)

Reproducible research benefits us all!

How to Record Your Research

- How you collected the data
- How you created your data set and variables in it
- How you analyzed the data
- How you interpreted the results and why
- Time stamp (date) for each

Managing Your Projects with RStudio

- You should manage all files for each of your research projects (e.g., each research paper) in a single place (folder)
- RStudio helps you manage your projects
- Create a new project in RStudio: [File] $\rightarrow$ [New Project]
  - If you already have a folder for the project, choose [Existing Directory]
  - If you'd like to start a project from scratch, choose [New Directory]
- Give the project an unambiguous name: file extension is ".Rproj"
- When you use (open) the project in RStudio, the project folder is automatically selected for your working directory: you don't have to run `setwd()`

How to Write R Codes in RStudio

- Choose the project: [File] $\rightarrow$ [Open Project]
- Create a new R script: [File] $\rightarrow$ [New File] $\rightarrow$ [R Script]
- Save the script with a name: file extension is ".R"
- A script is just a text file: you can open it with any text editor.
- Add comments starting with "#"
- Write a short explanation of the script at the beginning
- Run a code on the current line by hitting "ctrl（or cmd）$+$ enter"

What Should Be Written in an R Script

- File name
- Purpose of the script
- Input files (e.g., raw data) and output files (e.g., modified data set)
- Date of creation and creator's name
- Dates of modifications and the name of modifier
- Comments to the codes

Example Script

```
##############################################
## example.R
## wd: ~/classes/rm1/
## Purpose: Explain how to write R codes
## Datasets used:
##    data/fake-data-01.csv
##    data/fake-data-02.dta
## Created: 2014-10-14 Yuki Yanai
## Last Modified: 2015-10-11 YY
##############################################

## clear all the objects in the work space
rm(list = ls())

## load ggplot2 package to create beautiful figures
library('ggplot2')
```

What You Should Consider

- Readability: appropriate spacing, line breaks, indentation (blocking)
- Consistency of variable naming: e.g, linear_model or linearModel
- Can you understand the codes if you read it next week, next month, next year, in five years, . . . ?
- Can other people understand your codes?
- Too few comments? (Never too many comments)
  Rough standard: 30–70% of the script should be comments
- Same for other languages (e.g, do files for Stata)

Pros and Cons of R Scripts

Pros

- You can run the entire script by R

```
## run the script
source("example.R")
```

- Easy to make a script

Cons

- Not designed for codes with wordy explanations
- Can't read the results and explanation together
- Except the code highlighting by color, it's just a text file

Maximize Readability of Your Codes

- Correctness is a necessary condition for good codes but not a sufficient condition
- Good codes have high readability, other things equal
  - Easy to maintain, revise, and recycle the codes
  - Facilitate collaborative work
  - Higher transparency

Readability（1）: Comments

Write a lot of comments!

- Write what you'd like to know when you read others' codes
- E.g., comments for an R function to calculate the mean

```
get_mean <- function(x) {  ## calculate the mean
  ## Argument: x = a numerical vector
  ## Return: mean_x = the arithmetic mean of x

  n <- length(x)  ## length of the vector x
  sum_x <- sum(x) ## add all the values in x
  mean_x <- sum_x / n

  return(mean_x)  ## return the mean
}
```

Readability (2): Code Block by Indentation

Indent code block (2 or 4 spaces)!

- A bad example

```
for(i in 1:n){
for(j in 1:k){
x[i, j] <- mean(rnorm(10))
}}
```

- A good example

```
for(i in 1:n){  ## loop for the rows of x
  for(j in 1:k){  ## loop for the columns of x
    x[i, j] <- mean(rnorm(10))
  }
}
```

Readability（3）: Appropriate Spaces and Line Breaks

Use spaces and line breaks so that the codes look more beautiful

- Bad:

```
a<-(1+2)*4+5-8
plot(x,y,xlim=c(1,10),ylim=c(-5,5),xlab="x-la
bel",ylab="y-label",main="Title␣of␣fig")
```

- Good

```
a <- (1 + 2) * 4 + 5 - 8
plot(x, y, xlim = c(1, 10), ylim = c(-5, 5),
     xlab = "x-label", ylab = "y-label",
     main = "Title␣of␣fig")
```

What Is Literate Programming

Literate programming

Write computer programs with the explanation and interpretation of the codes in natural languages (e.g., English or Japanese)

- Donald Knuth (TEX's developer) proposed the concept
- Write a single file, and you'll get your data analyses and write-up done at once

Literate Programming with RStudio

1. Open a Project
2. Choose [File] $\rightarrow$ [New File] $\rightarrow$ [R Markdown]
3. Save the file with a name: file extension is ".Rmd"
4. Write header info
5. Write the explanations of your codes and the interpretation of the results in normal sentences: use Markdown
6. Write R codes in code chunks
7. Click [Knit HTML] at the top of the top left pane to create an HTML file

Introduction to R Markdown

Write the header information: Header starts and ends with three hyphens.

```
---
title: "Introduction to Literate Programming with RStudi
author: "Yuki Yanai"
date: "October 21, 2015"
output:
    html_document
    theme: united
    highlight: tango
    toc: true
---
```

R Markdown: How to Write Sentences

Write sentences as you do with an text editor or word processing software

- headings are signified by "#": The fewer "#", the higher the heading level
- Words between "*" or "_" will be italic.
- Words between "**" or "__" will be bold
- Words between "***"" or "___" will be bold italic
- List (bullet points):
  - Unordered list can be created by "-" (hyphen)
  - Ordered list can be created by numbers "1.", "2.", ...
  - Indent by tab to create nested lists
- URL link: [something to display](URL)
- Image: ![Words to show in place of image](image file path)

R Markdown: Code Chunks

- Beginning of the chunk: ```` ```{r chunk-name, chunk-options} ````
- End: ```` ``` ````
- Write R codes in between the two sets of three back quotes
- Give each chunk code a unique name
- Specify chunk options if necessary

R Markdown: Write Codes in Sentences

- To show the code itself, write it between back quotes: E.g., `` `sessionInfo()` ``
- To show the result of the codes in a sentence:
  - (input) the variable of x is `` `r var(x)` ``
  - (output) the variance of x is 30.8

Next Week

- How to present your results
- How to make tables and figures with R
- How to use ggplot2
- etc.