Lab 1: Logistics and R review

CSSS / POLS 510 — Maximum Likelihood Estimation

Calvin Garner

9/28/2018

Let's talk about me

Logistics

1. **Lab Sessions**: Fri, 3:30-5:20pm* in Savery 117
   - Covers application of material from lecture using examples; clarification and extention of lecture material; Q & A for homeworks and lectures
   - Materials will be available on the **course website**
   - * Probably won't take the whole time
2. **Office Hours**: Weds, 9am-11am in Gowen 30
   - Available for trouble shooting and specific questions about homework and lecture materials
   - Time is subject to change and, if it does, I will e-mail the list
3. **Homeworks**: 5-6 due every 2 weeks or so
   - Must be typed up
   - Ideally, done using R or R Studio with write up in LaTeX
   - Using R Studio with R Markdown is an easy way to do this
   - We will use two of Chris's packages extensively: simcf and tile

Logistics - Goals

1. **Be-Able-Tos**: When this course is over, you should be able to do the following (and much more):
   - Identify the proper distribution and model for your data (logistic, ordered, multinomial, count)
   - Run the model using both the glm function and "by hand" using optim, extract parameters of interest, and interpret these in probabilities
   - Compute predicted probabilities and use simulation to find the confidence intervals of $\hat{\pi}$ across counterfactuals values of $\boldsymbol{x}$
   - Use cross-validation to assess the predictive accuracy of several models and also compare these models across a variety of in-sample goodness of fit tests
   - Use one of several algorithms to impute missing data

Logistics - R

1. **The stuff in R**: For the homework assignments and project you will need to feel comfortable
   - importing (and exporting) data sets
   - tidying and transforming data
   - analyzing data (conceptual part of the course)
   - generating plots of your data and results
   - writing basic functions and loops for repeated procedures

- Fortunately, for those of you new to R, there are many resources to get you up to speed
  - Zuur et al. (2009) on syllabus, Chapter 1-5
  - R for Data Science (Wickham and Groleman 2017)
  - Not on the syllabus, but worth thinking about: tidyverse and related courses on datacamp
  - RStudio cheat sheets - help on tidyverse and other R programming

Logistics - R

2. I have to read lots of your code. Please be considerate when writing code and submitting assignments.
   - Do not print unnecessary code and output. Learn how to use `results = "hide"` and `echo = TRUE` in R Markdown.
   - Name well
     - functions vs. all other objects
     - readability is about consistency (dot.naming, camelCaseNaming, pothole_naming)
     - short, clear, consistent – help future you (and present me)
   - Specify arguments fully, e.g.

```
rbinom(n = 1000, size = 30, prob = 0.49) # GOOD!

rbinom(1000, 30, 0.49) # LESS GOOD!
```

   - See the Google R styleguide for an example

Logistics - Social Sciences & Computing

1. There are best practices for computing in the social sciences. You should aim for transparency and replicability in your work in general, and clarity and consistency in your code.
   - Best Practices (Wilson et al. 2014)
   - Good Enough (Wilson et al. 2017)

R Refresher

1. A programming language that is particularly good for the type of data manipulation, statistical analysis, and visualization that many quantitative social scientists do.
   - Free and open source == lots of packages that build on the base functionality
   - The first thing I remember learning about R is that it is "object oriented" – what does that mean? What are the objects of data in R? Specifically, what are the data types and structures?

R Refresher

2. Data Types
   - character, numeric (integer or double), logical, complex
   - data can also be missing
3. Data Structures
   - `(atomic) vector` (1d), `matrix` (2d), `array` (nd)
   - `list` (1d), `data.frame` (2d)

- (for much more see here or here)
- Why in the world does this matter?

R Refresher

4. R as calculator
   - Standard mathematical operators (e.g. `+ - * / ^` etc.)
   - Functions (e.g., `mean()`) take arguments (inputs)
   - Logical operators (e.g. `==, >, <, >=, <=, !=`) return `TRUE` `FALSE` or `NA`

```
1 + 7
```

```
## [1] 8
```

```
mean(1 + 7)
```

```
## [1] 8
```

```
mean(1 + 7) >= 4
```

```
## [1] TRUE
```

```
NA >= 4
```

```
## [1] NA
```

5. Create objects with assignment operator <-
   - Don't use = for assignment (even thought it works)

Vector practice

Create the following vectors

1. `vector_1` : The numbers one through five and then the number six five times
2. `vector_2` : 10 randomly drawn numbers from a normal distribution with a mean 10 and a s.d. of 1
3. `vector_3` : Results of 10 single binomial trials with a probability of 0.4
4. `vector_4` : Sample 100 observations from a 5-trial binomial distribution with a probability of success of 0.4
5. `vector_5` : The numbers one through three and the word apple

Vector practice

```r
#Clear memory
rm(list=ls())

vector_1 <- c(1, 2, 3, 4, 5, 6, 6, 6, 6, 6)
vector_1 <- c(1:5, 6, 6, 6, 6, 6)
vector_1 <- c(seq(from = 1, to = 5, by = 1), rep(6, 5))
vector_2 <- rnorm(n = 10, mean = 10, sd = 1)
vector_3 <- rbinom(n = 10, size = 1, prob = 0.4)
vector_4 <- rbinom(n = 100, size = 5, prob = 0.4)
vector_5 <- c(1:3, "apple")
```

- Why no `c()` for vectors 2-4?

Vector practice

6. What type of data is vector_2?
7. Round up vector_2 to two decimal place
8. What happened in vector_5?

Vector practice

```
is.character(vector_2)
```

```
## [1] FALSE
```

```
mode(vector_2)
```

```
## [1] "numeric"
```

```
round(vector_2, 2)
```

```
##  [1]  9.11 10.28  8.75 10.47 10.26  9.33  9.94 11.34 11.02 10.17
```

```
class(vector_5)
```

```
## [1] "character"
```

```
vector_5
```

```
## [1] "1"     "2"     "3"     "apple"
```

Matrices

7. matrix_1: Create 5 by 5 matrix containing all NAs
8. Assign matrix_1 the row names (a,b,c,d,e) and the column names (1,2,3,4,5)
9. Replace the NAs in the first columne of matrix_1 with Inf

Matrices

```r
matrix_1 <- matrix(NA, nrow=5, ncol=5)

rownames(matrix_1) <- c("a", "b", "c", "d", "e")

colnames(matrix_1) <- c(1, 2, 3, 4, 5)

matrix_1[, 1] <- Inf
```

```r
matrix_1<-matrix(NA,nrow=5,ncol=5)
rownames(matrix_1)<-c("a","b","c","d","e")
colnames(matrix_1)<-c(1,2,3,4,5)
matrix_1[,1]<-Inf
```

- (Which is easier to read?)

Lists

10. Create a list that contains vector_1, vector_2, and matrix_1
11. Locate vector_2 from the list

Lists

```
list_1 <- list(vector_1, vector_2, vector_3, matrix_1)

list_1[[2]]
```

```
##  [1]  9.107726 10.279678  8.746628 10.465361 10.263042  9.333213  9.942154
##  [8] 11.343016 11.018425 10.169122
```

```
# OR

names(list_1) <- c("vector_1", "vector_2", "vector_3", "matrix_1")

list_1$vector_2
```

```
##  [1]  9.107726 10.279678  8.746628 10.465361 10.263042  9.333213  9.942154
##  [8] 11.343016 11.018425 10.169122
```

Data Frames Practice 1

Data frames are a special type of list in which each row has same length. It is also a matrix like object, yet its elements - unlike elements in a matrix - doesn't have to be of same type. Most of the data we use are in data frames. Time for some practice:

1. Load Lab1data.csv in R
2. What is the data structure? What does that tell us about type?
3. Check the names and summary statistics of the data. Fix any names that are less than good.
4. Remove observations with missing values
5. Plot GDP per capita (on the x-axis) and polity2 (on the y-axis)
6. Create a new variable called "democracy". Assign 0 to countries with negative value or zero polity2 score, and assign 1 to countries with positive score.
7. Use a loop to do the same recoding

Data Frames Practice 2

Let's analyze the data that we gathered about experience with R and LaTeX:

1. Read in the data "first_day_survey.csv"
2. Inspect the data. What format are they in? What values do the data take, and how do those values correspond with the survey?
3. Generate some summary statistics.
4. How are these two variables related to each other?

Data Frames Practice 2

5. Are there any problems with the way the data are coded? (Think about lecture yesterday.)

Data Frames Practice 2

6. Recode the data such that
   - 0 = "What's that?"
   - 1 = "I've heard of it"
   - 2 = "I can use it or apply it"
   - 3 = "I understand it well"
7. Why is this coding method better?
8. Generate some plots of the data: histograms are good here, scatterplots even better.