Introduction to R Programming

Brian Vegetabile

2016 Statistics Bootcamp
Department of Statistics
University of California, Irvine

September 13th, 2016

What learn to program in R?

- Programming can help supplement statistics and probability in a major way.
  - Specifically experiments and simulations can be created in R which allow us to validate our theories
- Pick up any modern statistics paper and there will be a section titled 'Simulations' which supports the results of the paper
- Learning to program in R is especially important given that it is historically a programming language designed for statistical applications.

From the most recent issue of the Journal of the American Statistical Association (JASA)

a dependence structure that is time-varying (which is the more likely scenario compared to the simplistic assumption of stationarity), then it would be appropriate to fit a nonstationary model. In Section S.2.1 (supplementary materials), we provide another illustrative example to show that our method can easily detect negatively correlated neurons.

2.2.6. Simulation Studies

To further illustrate the advantage of our method, we conducted two simulation studies to compare our dynamic model with the method in Kass, Kelly, and Loh (2011), which is the state-of-the-art approach for studying cross-neuronal interactions. For the first simulation study, we compared the two methods in terms of their ability to correctly detect the correlation between two neurons. To this end, we simulated data according to the approach we discussed above. Namely, the two neurons are independent for the first 80% of time ($\zeta_t = 1$ for $t \leq 0.8$) and dependent for the last 20% of time ($\zeta_t \geq 1$ for $t > 0.8$). We conducted simulations for five scenarios where we set the extra term ζ_t for the last 20% of time to 1.0, 1.2, 1.4, 1.6, and 1.8, respectively. Corresponding to each pair (which we call "Pair1") under each of the scenarios, we also simulates an independent pair (which we call "Pair2"), where $\zeta_t = 1$ over the entire time. We then apply

parameters), and then test the models on the remaining one-third of the data by using the firing status of one neuron to estimate the firing probability for the second neuron. We derive the estimated firing probability, $\hat{p}_{2k}$ for the kth observation in the test set for the second neuron using parameters estimated using the training data. Models are evaluated based on their log predictive probability (LPP) using the estimated firing probability, $\hat{p}_{2k}$, as follows:

$$\text{LPP} = \sum_{k \in \text{test set}} y_{2k} \log(\hat{p}_{2k}) + (1 - y_{2k}) \log(1 - \hat{p}_{2k}).$$

The model with a larger LPP value is considered to have greater predictive power. As before, we repeat each scenario 100 times. Figure 6 shows the 95% intervals of LPP for each scenario using our proposed model (shown as red bars) and the method of Kass, Kelly, and Loh (2011) (shown as green bars). Again, the results show that both models display an increasing predictive power as the strength of correlation between two neurons increases. Moreover, consistent with the previous results, our proposed

Example - Birthday Problem I

- From probability theory this is a historic problem called "The Birthday Problem"
- The problem concerns itself with a set of n randomly chosen individuals and asks whats is the probability that any pair of individuals in the room share a birthday
  - Note that this is different than asking does anyone have **my** birthday.

Example - Birthday Problem II

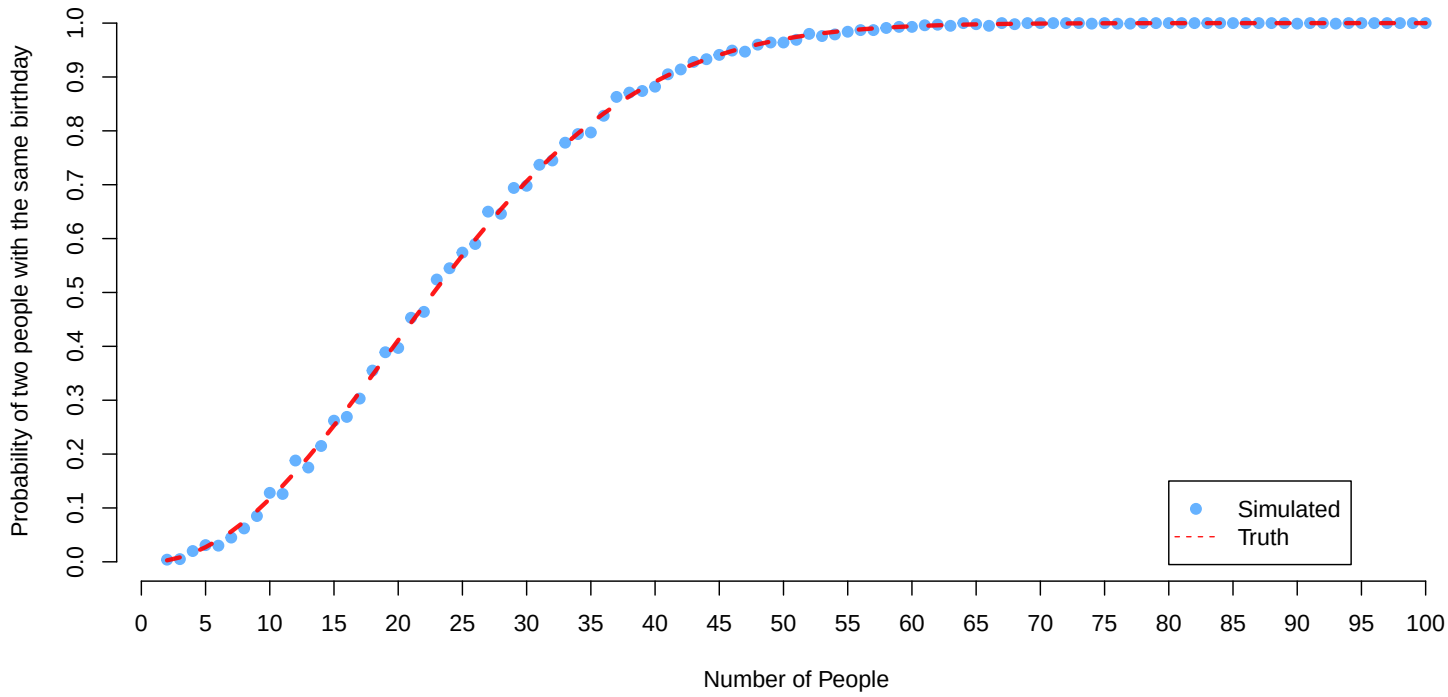
- We can solve this using probability theory and show that

$$P(n) = \frac{365!}{365^n (365 - n)!}$$

- This probability exercise is less fun than coding it up ourselves
- Additionally we can gain some insight from simulating the birthday problem

Example - Birthday Problem III

- Let's compare the simulation results with the theoretical results



- We see good agreement between theory and simulation

What is R I

From the CRAN page's 'Intro to R'

- R is a language and environment for statistical computing and graphics.
- R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering,) and graphical techniques, and is highly extensible.
- One of Rs strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed.

What is R II

- 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.
- Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented.

Why R

- Compare that with the python introductory documentation
  - Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming.
  - Pythons elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.
- Clearly python is intended for more than statistical applications, while R was built with statistics in mind!

R Studio

- The easiest place to start working with R is to work in R Studio
- Contains everything you need to get started quickly
  - Text Editor to write Code
  - Console to Run Code
  - Window for Plotting
  - Frames for showing the environment, files, etc.

Getting Started with R

- While R could be used in the command line as a calculator, the primary benefit is being able to keep track of, and manipulate variables
- We will introduce the basic data structures in R that will be most used throughout the year

References