Parallel Computing in R and Simulations on the Cluster

Computing Club

April 30, 2019

Lamar Hunt

Background

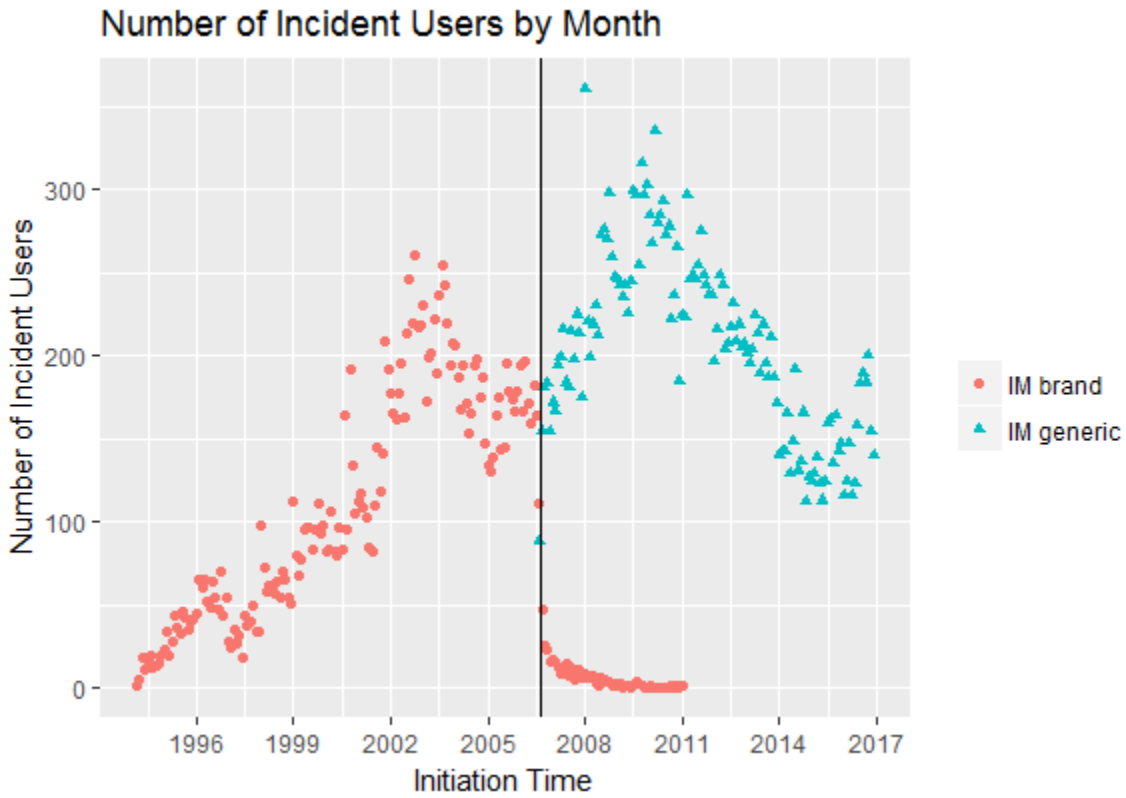
- Brand and Generic drugs are generally considered “equivalent”, however, the FDA does not require generic drug producers to perform a clinical trial of efficacy and safety
- All that is required is a check of “bioequivalence”, i.e., that
  1. the generic has the same amount of active ingredient
  2. the generic drug is absorbed the same amount
- However, sometimes bioequivalence is not assessed properly, there is fraud, or bioequivalence is not enough to establish equivalent safety and efficacy (e.g., aspirin with a bit of arsenic is bioequivalent to aspirin, but not as safe)

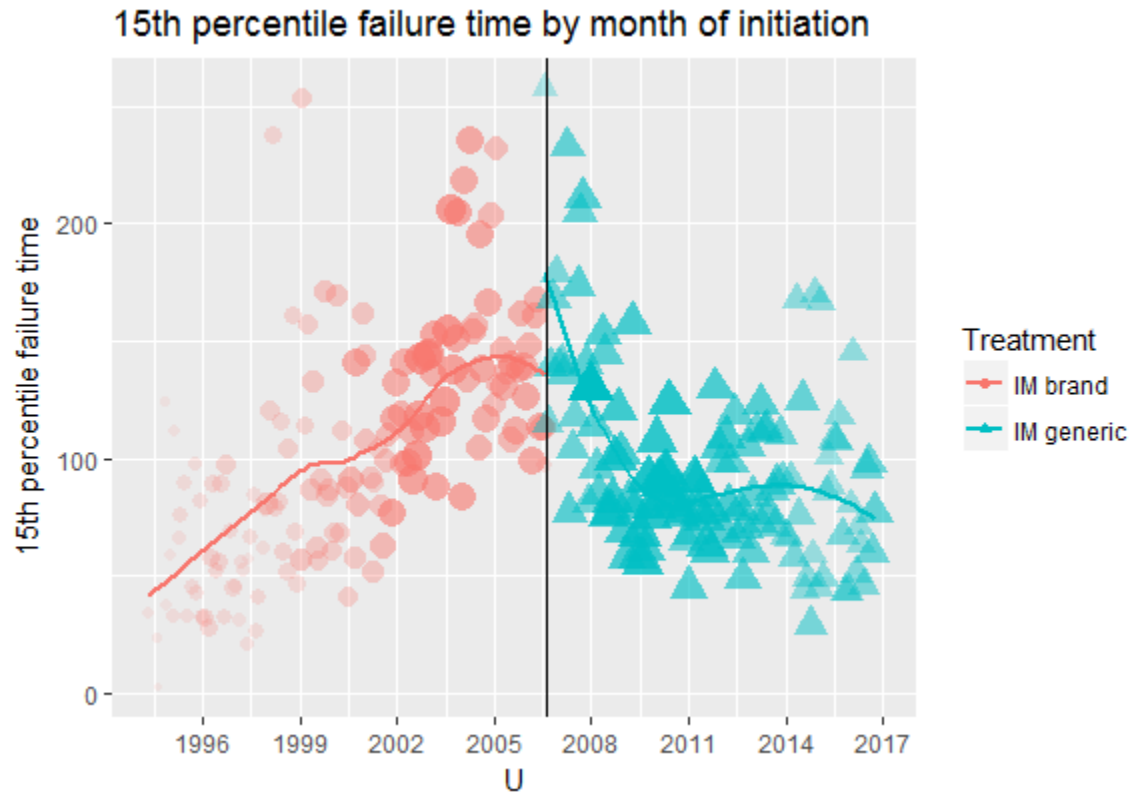
Goal of research

- The FDA therefore needs a method to assess **therapeutic** equivalence (i.e., safety and efficacy) of brand and generic drugs that are on the the market, especially when concerns arise (and they have).
- Ideally, this would be done with readily available observational data. In our case, we will use insurance claims data.
- The goal is to develop a method that can assess therapeutic equivalence using insurance claims data
- We apply the method to venlafaxine (an anti-depressant), and use time to failure (switching to another anti-depressant within 9 months) as the clinical outcome of interest.

Methodological Challenges

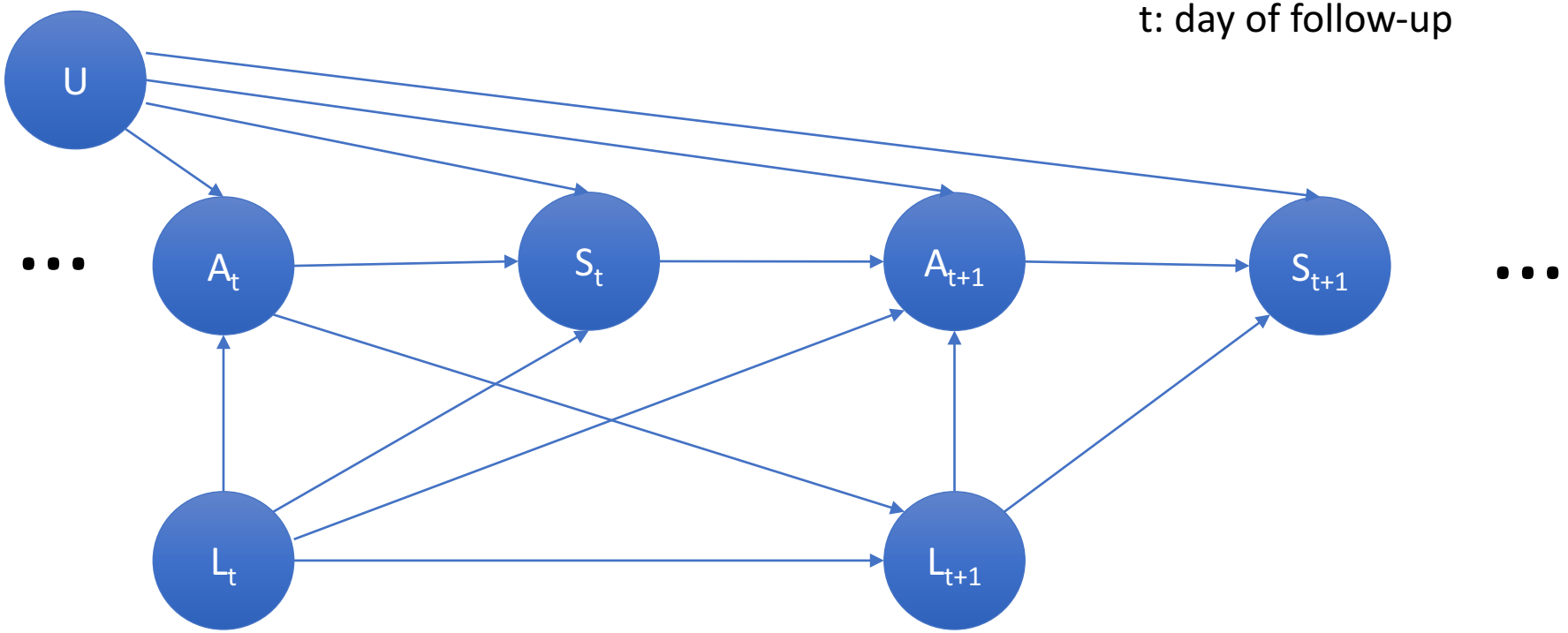
- If we performed an ideal randomized trial, we could avoid any bias due to confounding
- However, in observational data, the people taking brand and generic are different in many key ways.
- We identified 3 key ways in which brand and generic users differed
  1. Generic users can only initiate treatment after the generic is available. Thus, brand and generic users are often being treated in different decades.
  2. Patients don't adhere to treatment. That is, they are not filling their Rx's when they should. This could lead to "time-varying confounding"
  3. Generic Rx fills generally have lower co-pays (out of pocket costs), which could impact the results


Number of Incident Users by Month
Number of Incident Users
300
200
100
0
1996
1999
2002
2005
2008
2011
2014
2017
Initiation Time
IM brand
IM generic


15th percentile failure time by month of initiation
15th percentile failure time
200
100
0
1996
1999
2002
2005
2008
2011
2014
2017
U
Treatment
IM brand
IM generic

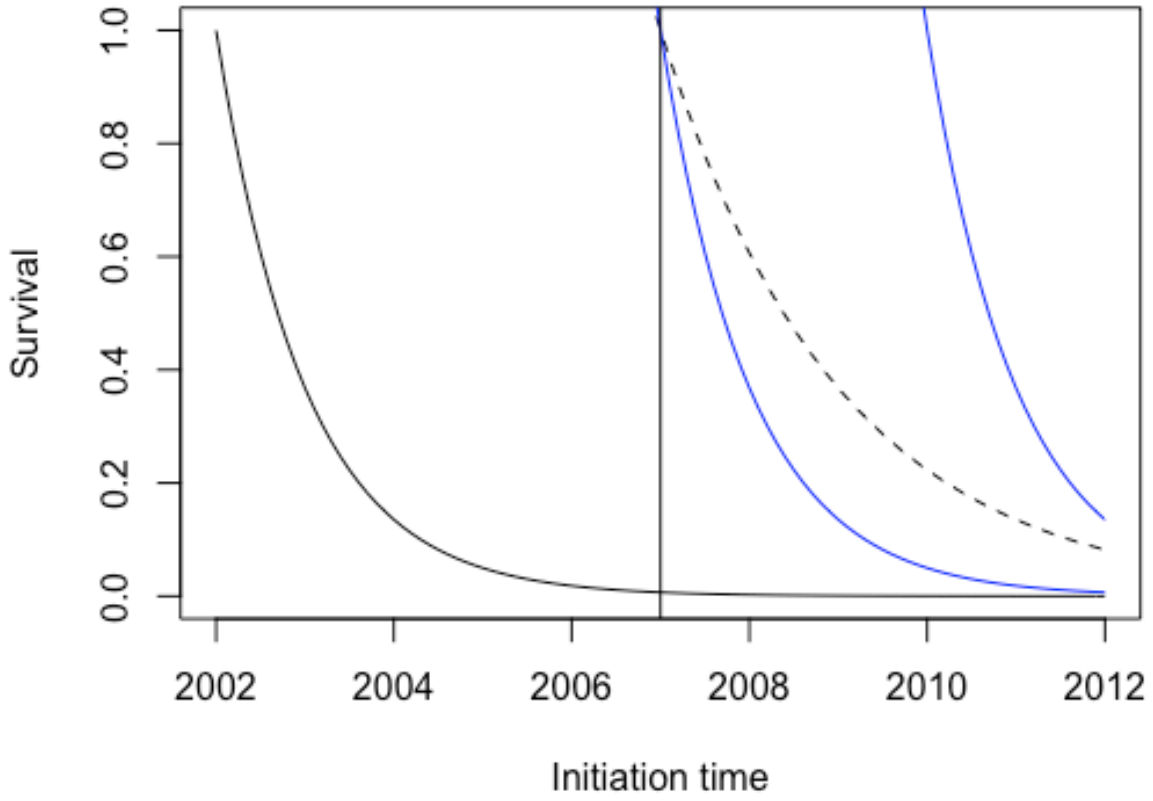
Causal Diagram (DAG)

A: exposure (cost and drug form)
S: survival
L: Rx Burden
U: initiation time
t: day of follow-up



Solutions

- To handle different initiation times, we applied a method known as “Regression Discontinuity”



Solutions

- To handle the ”time-varying confounding” we applied a technique known as G-computation
- This involves splitting the follow-up time into days, and introducing a daily variable that indicates whether the patient has had failure by that day
- We can then model the probability of failure on each day, conditional on all past covariate values. This will control for the confounding
- We also need to model the time-varying covariates themselves (i.e., the Rx burden and out of pocket costs)

Computational Challenges

- Broadly, the challenges we faced fell into 3 categories
  1. Too much data
  2. Very intense modelling procedures (numerical integration wrapped inside a bootstrap)
  3. Not enough computing power
- Anyone doing research with insurance claims databases will face these challenges!

Too much data

- The sample size was about 42K, but each patient has an observation on each of the 270 days of follow-up. That's about 11 million records.
- We also had about 50 variables in the analysis file. The data file was about 5gb.

Computationally Intense Modeling

- Performing the G-computation in our setting involved fitting 8 models as well as perform numerical integration twice
- However, we had to use the bootstrap to estimate confidence intervals, so we nested the entire analysis procedure into the bootstrap.
- In addition, performing regression discontinuity requires doing sensitivity analyses to check assumptions. So we had to run the entire analysis multiple times under different settings

Limited Computational Resources

- We were forced to perform all analyses on the servers provided by the data vendor that we got our data from (Optum Labs)
- However, we were only provided with the amount of RAM and number of cores that were available on a single computer
- Therefore, when we performed the analysis in the most straightforward way, we estimated the time to be about 70 days per analysis (not including the sensitivity analyses).
- Finally, Optum Labs servers shut down periodically, which interrupts any analysis currently running.

Limited Computational Resources

- Not only the analysis took a long time.
- Both data cleaning and exploration were painfully slow, and we frequently ran out of RAM when simply loading the data and running a few models, or creating new variables on the entire dataset.
- The solution to this was to:
  1. be very careful about removing unneeded objects from R as soon as they were not needed
  2. use the “data.table” package for fast manipulation of data.frames (instead of dplyr!), and apply vectorized functions that use fast C code in the background (e.g., “colmeans()”, “apply()”). This package also has the functions “fread” and “fwrite” (fast read and fast write) for reading and writing .csv files.
  3. Constantly make multiple calls to “gc()” (garbage collection) after anything is removed from R
  4. Use sped up model fitting packages like “speedglm” for GLMs, and “rms” for categorical data models like ordinal logistic regression.
  5. Compare speed of various options using the function “system.time()”

Speeding things up

- In addition to applying simple tricks to decrease memory use, we had to drastically reduce the time it took to analyze the data
  1. Randomly partition the dataset into K partitions, and write each partition out to the hard drive. Remove everything from the workspace, or even just restart R.
  2. Run the analysis on each partition in **parallel**, making sure to only load each partition of the data onto the computing node that needs it. Sacrifice what you can to make it run fast (e.g., we computed the bootstrap on each partition with only 100 bootstrapped samples—not ideal)
  3. Write out the results from each analysis to a file on the hard drive
  4. Combine the results afterwards

Speeding things up

- Combining the results from each partition relies on the fact that the partitions are independent of each other.
- If we want to estimate E[X], and $\mathbf{X_1}$ and $\mathbf{X_2}$ are two independent samples, then $\bar{X} = (\bar{X}_1 + \bar{X}_2)/2$ is a consistent estimator of E[X]
- On each partition we got an estimate of the variance using the bootstrap. This allowed us to estimate the variance of $\bar{X}$ using the fact that: VAR($\bar{X}$) = VAR($\bar{X}_1$)/4 + VAR($\bar{X}_2$)/4
- We then created confidence intervals using this estimate of the variance, relying on the normal approximation (central limit theorem), since the sample size was so large. Note that any estimates forced to be between 0 and 1 (e.g. a probability) should be transformed to the log scale first

Some Resources

- https://cran.r-project.org/web/packages/speedglm/speedglm.pdf
  - Vignette for “speedglm” package
- https://cran.r-project.org/web/packages/data.table/vignettes/datatable-intro.html
  - Introduction to data.table package
- https://nceas.github.io/oss-lessons/parallel-computing-in-r/parallel-computing-in-r.html
  - Introduction to parallel computing in R

Simulation Studies

- Every new method you develop will require performing a simulation study.
- This involves generating thousands of random datasets from a known true distribution, and then applying your method to each dataset.
- It gives you a sense of the bias and variance of your estimator under settings that you control

Simulation Studies

- In my case, I had to generate thousands of datasets that were each very large (due to the repeated measures on each day of follow-up)
- The best option is to use the JHPCE Cluster (https://jhpce.jhu.edu/)
- You can get training and access by contacting Jiong Yang