**The Road Most Traveled:**

**A Simulation of Commute Times from Philadelphia to Oaks**

Matthew Huber

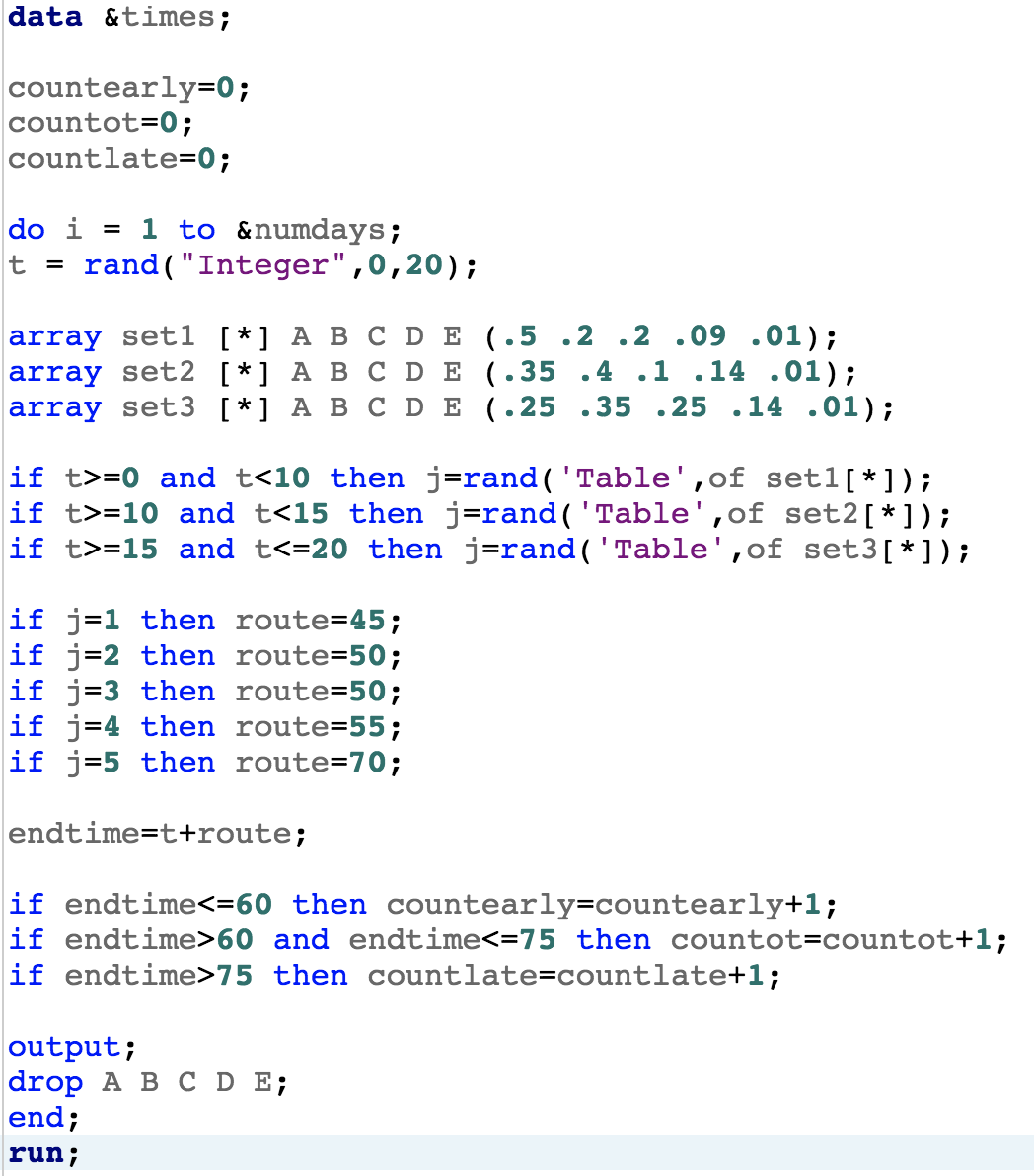
When presented with the task of creating a statistical problem to solve in SAS using simulation techniques, it did not take me long to formulate an idea that was simple, practical, and applicable to my everyday life. I now live in Philadelphia, near the art museum and the Eastern State Penitentiary. However, I work for SEI, an investment company located in Oaks, PA, which is located about 20 miles northwest of the city. Now, normally 20 miles is not that long of a commute, yet every morning it takes me almost an hour to get from apartment to office, sometimes even more. Each morning I find myself starting up my car and the Waze app to see which path I will take to get to work the fastest in an attempt to both not be late and to get to work early and get paid more money since I am hourly and not salaried.

Given this context, I knew immediately that I wanted to simulate my commute time to work, and figure out the probability of being early, on time, or late. Using prior experiences and a good memory, I set about laying out the framework for my code. I knew that I wanted to contain most, if not all of my code, in one macro that I can invoke a number of times. Each time I would invoke the macro, I would change the number of days that are being sampled so that I could determine the probabilities mentioned above for standard time periods, such as a week, a month, and a year.

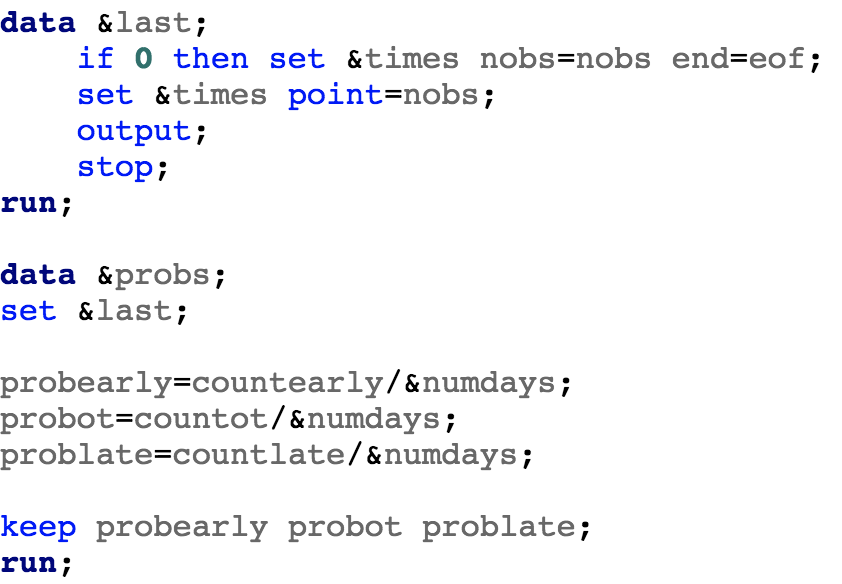
I entitled my first macro commute, for obvious reasons, and defined four parameters. These parameters were numdays, times, last, and probs. The numdays parameter is used to set the number of days that I want the macro to run for. Times is the parameter used for the name of the data set that I create to generate and store all of my other variables. As seen below, the data set that I create begins with creating count variables that will be later used to calculate probabilities for being early (countearly), on time (countot), and late (countlate). I then invoke a do loop, which comprises the rest of this data set. The do loop iterates a total number of numdays times, creating a data set with one observation for each day that I am running the macro for. The first thing the do loop does is generate an integer between 0 and 20. 0 stands for 7:00 AM, the earliest that I leave for work in the morning, and 20 stands for 7:20, the latest that I leave my apartment. The distribution is fairly uniform I’ve found, so that’s why I have the time being generated randomly without any weighted probabilities.

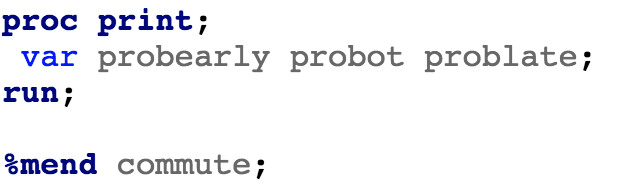


The next part of the do loop creates 3 arrays, one for each specific time frame that I will explain further in a bit. In each array, there are 5 probabilities. These probabilities correlate to one of 5 different routes that Waze takes me each morning. Routes A, B, C, D, and E. Route A correlates to Kelly Drive and has a base commute time of 45 minutes from apartment to office. Route B takes me on Belmont Avenue to City Avenue and onto 76, and has a base commute time of 50 minutes. Route C takes me on Belmont Avenue to Manayunk and onto 76 at that exit, also with a base commute time of 50 minutes. Route D takes me all back roads from Philly to Conshohocken and onto 76 at the intersection with 476. This base commute time is 55 minutes. Lastly, Route E is if Waze is either particularly glitchy or if the universe is punishing me for some reason, because it will put me straight on 76 from the get go and I will sit in traffic for over an hour (usually between 70-80 minutes but I chose to be conservative and go with a base time of 70 minutes).

 The reason I have 3 different arrays is that depending on when I leave in the morning, the probability that Waze will take me on one of the 5 routes mentioned above changes. If I leave between 7 and 7:10, the chances of routes A through E can be seen in the array set1. Set2 contains the probabilities if I leave between 7:10 and 7:15. If I leave after 7:15, set3 contains the updated probabilities for A through E. The if statements assign a value of 1 through 5 to the variable j depending on which array I am invoking. So, for example, if t=12 (leave my apartment at 7:12), then j will generate a value from set2. There is a 35% chance that value will be 1, a 40% chance that value will be 2, a 10% chance that value will be 3, a 14% chance the value will be 4, and a 1% chance that the value will be 5. The piece of code following the if statements assigns the base times as the variable route, with 1-5 corresponding to routes A-E respectively.

Lastly, I define the variable endtime to be the base time t plus the base route time route. I then have three if statements categorizing the end time as either early, which will be me arriving at work before 8, on time with me arriving at work between 8 and 8:15, and late with me arriving at work after 8:15. I then output all the variables for that obnservation, drop the values of A-E, end the loop, and run the data set.

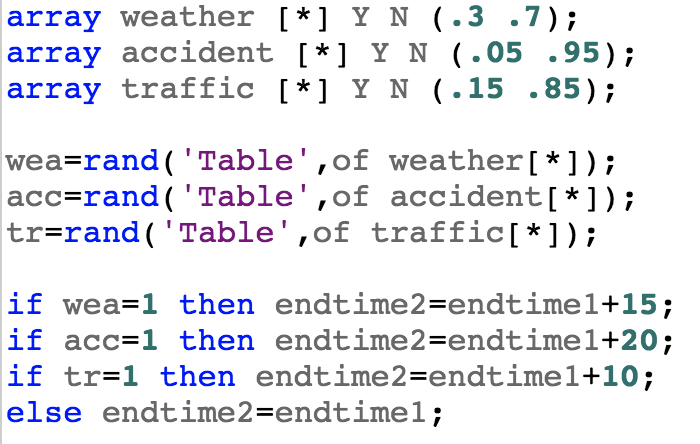
The next part of my code contains two data sets and the last two of the parameters defined in my macro. The first data set, which uses the “last” parameter to name the data set, takes the times data set generated above and pulls the last observation. I do this in order to extract the final counts for each of the count variables. I then define the second data set using the probs parameter and set the last data set. I then define three more variables, probearly, probot, and problate as the count variables divided by numdays. Then I have a keep statement so that only probearly, probot, and problate are left in the data set.



Finally, to end the first macro, I print out the three probability variables to examine what I ended up with and determine how often I will be early, on time, and late.

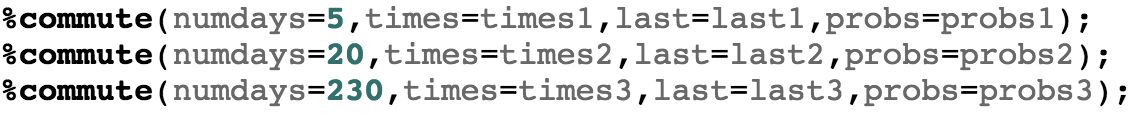
I didn’t want to stop there though. I knew that my commute is never that simple, and that there are always three major factors that will affect my commute times. Weather, accidents, and traffic. Well, heavier traffic than is the norm on 76 every morning. So, I went about defining a second macro. The basic structure, parameters, data sets, and variables are all the same, however I added a few things.

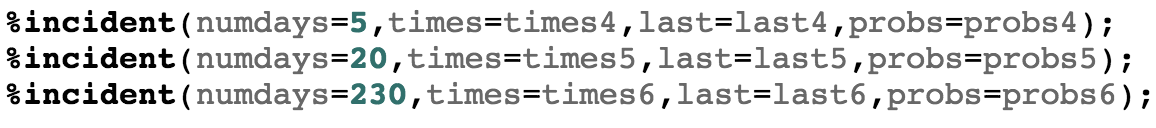




The biggest change occurred within the do loop, where I added the below code. In it, I define three more arrays. Each array contains the rough estimates of probabilities for bad weather, an accident, and heavier traffic. I then create three binary variables, wea for weather, acc for accident, and tr for traffic. I do this so that these incidents can pile on top of each other and make the code a bit more complex. I then calculate a second end time that factors in the additional times added per incident (15 more minutes for bad weather, 20 for an accident, and 10 for heavier traffic). The rest of the macro is consistent with the commute macro above.

To end the simulation, I wanted to invoke 6 macros. 3 commute macros determining the probabilities for 5 days (a work week), 20 days (a work month), and 230 days (a work year factoring in weekends and my vacation time). 3 incident macros for the same time periods.





Since most of these numbers are generated at random, the macros that invoke a smaller number of days will naturally fluctuate. However, once the number of days increases, I found that the probabilities seem to converge to similar numbers each time I run the full code. The good news is that no matter what the circumstances, I will rarely be late to work.

