System Simulation, Fall 2018

Simulation Challenge Problem #1: Pacific Playland Guest Arrival Model

Pacific Playland is a fictional amusement park. We will, for purposes of discussion and illustration, occasionally use computer simulation and modeling techniques to explore "what if" questions about park's operations.

For our first challenge, we'll construct a pseudo-random number generator that can generate random guest arrival times that model random arrival times collected at the park's gate on a typical day. Along with this document, you will find a file called `ARRIVE_DATA` that contains one hundred thousand guest arrival times coded as indicated in that comment block at the beginning of that file. I have also included a file that contains the same data, except that the times are given in units of "fractional hours" (xx.yy) where yy is the "fraction past the hour.

Your group should produce a pseudo-random number generator that can, on demand, produce a random arrival time in the range of 10:00 and 23:59 and 59 seconds on the same day. The times generated should, if plotted as a histogram, mimic the probability mass distribution underlying the original source data. You may use a pre-existing random number generation library that provides uniformly distributed random numbers. You may also use the methods of inverse transform sampling or rejection sampling to create your own code that creates random arrival times.

On your way to generating your code, you may wish to attempt the following conceptual exercises:

1. Figure one shows a relative frequency histogram of the arrival times in `ARRIVE_DATA`. The y-axis shows number of people and the x-axis shows arrival time. The width of each bar represents the "range of time" each bar represents. Figure one has 84 bars with each representing a period of 10 minutes during the day. So, the height of the first bar is all the people arriving between 10:00 and 10:10 AM.
    a. Generate the same histogram using tools and methods available to you. If you have no other preference, GNUPLOT is a freely available

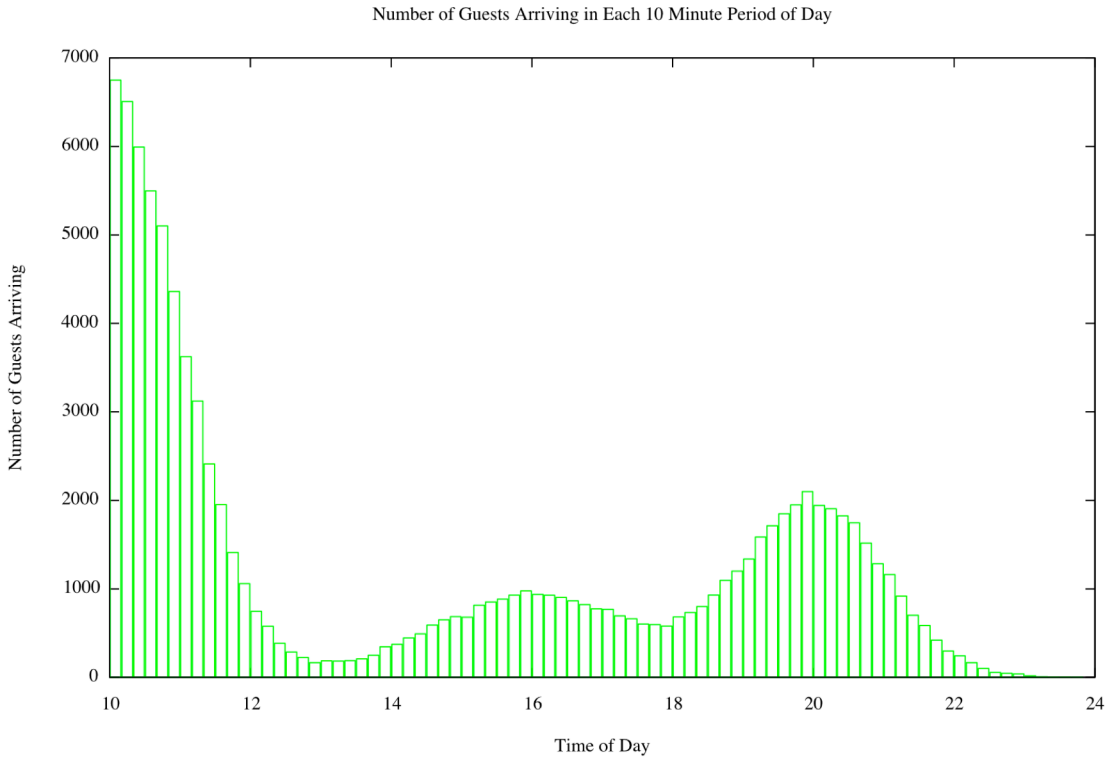

Figure One: Relative Frequency of Guest Arrival Times at Pacific Playland Amusement Park on a Typical Day

program that can be used to generate many sorts of graphs. The GNUPLOT script used to generate figure one is included along with this writeup. Also included is a Python Jupyter notebook that does the same thing. You can use the freely available Anaconda Python package to load and modify the notebook.

b. What happens to the look of the histogram as one increases or decreases the number of bars (which also decreases or increases the width of the bars). For this data, does it make sense to have more than 50400 bars? Why or why not?

2. How would one represent the relative frequency histogram in a general purpose programming language? How would one, from that representation, create a representation of the Probability Mass Function (PMF)? How would one, from the PMF, create a representation of the cumulative probability distribution function (CMF). Write code that generates PMF and CMF functions based on the data I provided. Demonstrate this code for the instructor or TA before moving on. Make sure everyone in your group understands it.

3. Using the method of inverse transform and the CMF write code that generates random numbers from the distribution underlying the data in `ARRIVE_DATA`. Note that the "width" of the bars in your CDF/Inverse CDF may affect the precision of the data you can generate. Think carefully on this issue and design your code appropriately. Demonstrate this code for your instructor.
4. Using the method of rejection and the PMF, write code that generates random numbers from the distribution underlying the data in `ARRIVE_DATA`. Note that the "width" of the bars in your PMF may affect the precision of the data you can generate. Think carefully on this issue and design your code appropriately. Demonstrate this code for your instructor.
5. Use the code you generated in #3 and #4 to create two data sets of random arrival times. You should generate one set for each method and each set should have at least 200K random values in it. Produce appropriate histograms of both data sets. Use these histograms, and any other tests you might deem appropriate, to verify that you are generating arrival times from the correct distribution. Show the histograms to your instructor or TA.