EM384: Analytical Methods for Engineering Management

Lesson 35: Monte Carlo Simulation Applications in Python

26 April 2023

Table of contents

1. Lesson Objectives

2. Monte Carlo Simulation in Python

3. Practical Exercises

4. Conclusion

Lesson Objectives

Lesson Objectives

- Design and implement simple Monte Carlo simulations in python.
- Visualize the results of monte carlo simulation in python.
- Create and interpret an empirical cumulative distribution function (ECDF) from the results of the model.

Monte Carlo Simulation in Python

Monte Carlo Simulation

Given a system…

And ideas of how to model the system…

And some data that describes the behavior of the system…

- Determine how you plan to measure the performance of the system (system performance measures) → will be your model output(s).
- Develop a base case model of the system that estimates system performance measures. You can start with simple diagram, essential mathematical relationships, and logic.
- Identify components of the model susceptible to random variation. Collect data that represents this random variation. If data is not available, estimate the random variation.
- Fit the observed data to known random variable distributions. These random variables will be used to create overall system behavior variance.
- Develop pseudo-code that iterates through multiple iterations of your model. Each iteration will incorporate a random outcome of the variables identified in the previous step.
- Implement your pseudo code in Python.

EM384 Standards for Monte Carlo Simulation in Python

Before starting: Sketch out your model using pseudo-code! (this is a general structure of your code using words and abbreviations). Then, in Python:

1. Import the required libraries at the top.
2. Define your fixed parameters by assigning values to variables. Pick names that make sense.
3. Define an empty list (or array) to in which save the results of your Monte Carlo Simulation.
4. Create a iteration structure (e.g. FOR loop) to iterate through your model. Inside your iteration structure:
   - 4.1 Define values that are random variables.
   - 4.2 Use calculations and logic to obtain the result(s) for the current iteration.
   - 4.3 ADD the result of your current(s) iteration to your results list.
5. Create a histogram of your results list data.
6. Create the ECDF function of your results list data
7. Use the ECDF to answer questions related to probability of events.

Practical Exercises

PE1 Car Wash

You recently purchased a car wash for your gas station. The wash component takes triangular(min=23,mode=30,max=35) and the dry component takes N(mean = 15 seconds, std dev = 5 seconds).

- Let x_1 = wash time, Triangular(min=23,mode=30,max=35)
- Let x_2 = dry time, Normal(mean = 15 seconds, std dev = 5 seconds)
- Let $x_1 + x_2 = y$ = complete car wash cycle time

What is the first and 99th percentiles for expected car wash times?

PE2 Struggling Deli

Your deli experiences a high volume of orders between 11am and 1pm for the lunch rush. Your restaurant specializes in salads, wraps, and burgers. About 30% of the meals are burgers, 50% are wraps, and 20% are salads. Your cook prepares meals with the following time distributions:

- Burgers take Normal(mean = 3, std dev = 1) minutes. Burger profit = $1.50
- Wraps take Normal(mean = 2, std dev = 0.5) minutes. Wrap profit = $0.75
- Salads take Normal(mean = 1.5, std dev = 0.25) minutes. Salad profit = $0.50

Questions:

1. You have one cook that manages the orders in a first come, first serve sequence. How much profit do you expect to make during a typical lunch period?
2. What is the probability of making less than $50?
3. What is the probability of making more than $55?

Conclusion

Next Class

Homework:

- Review Block 4 readings and previous Python tutorial videos.

Next Lesson:

- Design and implement simple Monte Carlo simulations in python.
- Visualize the results of monte carlo simulation in python.
- Create and interpret an empirical cumulative distribution function (ECDF) from the results of the model.