Simulating a Queue

CS 547

Procedure

To simulate a single-server queue, keep track of five lists of values.

- the randomly generated time between customer arrivals
- the actual arrival time of each customer
- the time each customer enters service
- the randomly generated service time of each customer
- the time each customer completes service

As the simulation progresses, we'll continue to generate new customers until we reach a pre-set limit, N. Suppose customers 1 through $k-1$ have been generated, and we now want to generate customer k.

First, use the inverse CDF method to generate an interarrival time for customer k, then set

```
arrival_time[k] = arrival_time[k-1] + interarrival_time
```

Customer k can enter service in one of two ways.

- if the queue is empty when k arrives, it immediately enters service
- otherwise, it waits until customer $k-1$ departs, then enters service

Therefore,

```
enter_service_time[k] = max(completion_time[k-1], arrival_time[k])
```

Finally, generate a random service time for customer k and calculate its completion time.

```
completion_time[k] = enter_service_time[k] + service_time
```

The residence time for customer k is the time between when it arrives and when it finally completes.

```
residence_time[k] = completion_time[k] - arrival_time[k]
```

Average the residence times for each customer to estimate the overall average residence time, $\overline{R}$.

To begin the simulation, you can simply assume the first customer arrives to an empty queue at time 0 and immediately begins service.

Example Output

```
 interarrival  arrival  enter_service  service  completion

    0.4006      0.4006     0.4006       2.6595    3.0601
    2.0285      2.4291     3.0601       0.1946    3.2547
    0.6796      3.1086     3.2547       1.6883    4.9430
    0.9295      4.0381     4.9430       0.0936    5.0366
    1.6489      5.6870     5.6870       0.3492    6.0362
    0.7058      6.3928     6.3928       1.1255    7.5183
```

Simulation Error

I ran the simulation algorithm five times using an M/M/1 queue with $\lambda = 1$ and $\mu = \frac{4}{3}$ and calculated the following estimates of $\overline{R}$:

```
R_bar =

    3.2471    2.9386    2.6589    2.8736    3.2909
```

Using the M/M/1 residence time equation, we can calculate that the true value of $\overline{R}$ should be 3. All of the simulations produced estimates with some amount of error from the true residence time.

By its very nature, a simulation is attempting to estimate a system parameter using only a finite amount of data. Therefore, there will always be some amount of error between the results produced by a simulation and the true value of the parameter being measured. There are two basic ways of dealing with this problem:

- run the simulation for a very long time, to attempt to minimize the error by collecting more data
- run the simulation multiple times and use statistical techniques to estimate the true parameter

The use of statistical estimates is an important part of simulation studies. Because there is always some unknowable amount of error in the results, the estimate produced by a single run of a simulator is generally not reliable.

Behavior of Simulation Errors

When the parameter of interest is calculated using an average – which it is in this case, since we're estimating $\overline{R}$ – the error between the simulation results and the true value tends to follow a normal distribution.[1]

The figure shows a histogram of the results produced by running 10000 simulations. Notice the clear bell-shaped curve, which is centered at 3, the true residence time.

Because the simulation errors tend to be normally distributed, it's possible to use the statistics of the normal distribution (or another closely related distribution) to derive estimates of the true value of $\overline{R}$, even using only a small number of simulated trials.

[1] This is a consequence of the famous *Central Limit Theorem*.

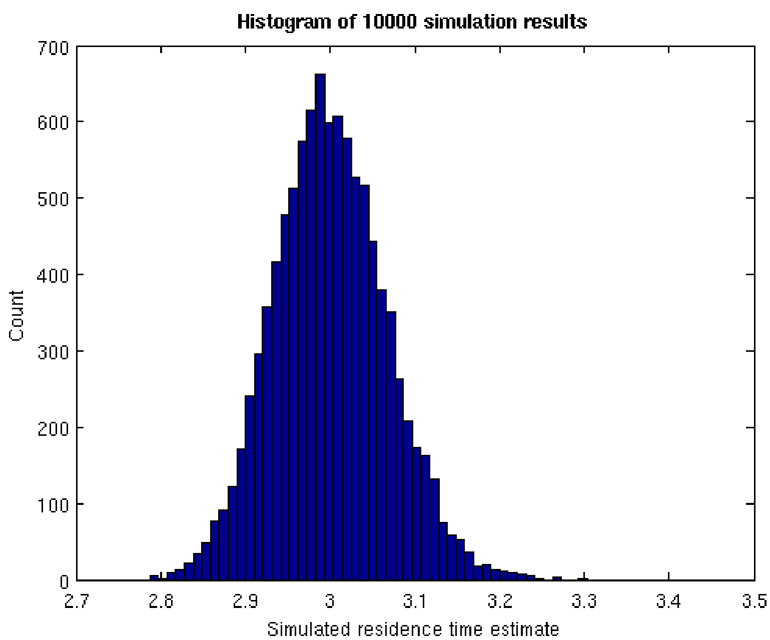

Figure 1: Errors in the residence time estimate tend to be normally distributed