Overview: You will incorporate input modeling techniques to use arrival data collected by University of Richmond students from the *Eight Fifteen* (campus coffee shop), along with service data previously collected from *Tyler's Grill* (short-order take-out), in a multiple-server queueing model, and then use output analysis, including batch means, to recommend a number of servers for the *Eight Fifteen*.

1. On Lyceum, I will provide `msq.py`, which is an extended `ssq.py` (using `simulus` in Python) that implements a multiple-server queueing model, consistent with Section 5.2.2 of the provided MSQ reading (see Lyceum).

   In a PDF write-up, provide reasonable summary discussion of the model / source code to demonstrate your understanding of the MSQ model, including the different event types, event-type algorithms, etc. and how those are deployed using Simulus.

---

2. Use the method of moments (MOM) to separately fit a *gamma* (k,Θ) and a *lognormal* (μ,σ^2) distribution to the *Tyler's Grill* service data available in the `simEd` library in R.

   (a) In class, we worked through the details of MOM for *gamma*. In your write-up, provide the mathematical details and estimates for k and Θ determined using `tylersGrill$serviceTimes`.

   (b) To use MOM for determining parameter estimates for the *lognormal* distribution, the easiest approach is to first take the log of the service data and then fit the *normal* distribution using MOM. (This is equivalent to fitting the *lognormal* distribution.) In your write-up, provide the mathematical details, and estimates for μ and σ determined using `tylersGrill$serviceTimes`.

   (c) Provide an appropriately labeled graphic that includes a plot of the ecdf of the service data (recall use of `plot.stepfun` — see examples from class), and cdf curves of the fitted *gamma* and *lognormal* distributions superimposed. Use solid-blue and line width 2 for *gamma*; use dotted-red and line width 2 for *lognormal*.

   (d) Use the Kolmogorov-Smirnov goodness of fit test to evaluate the fits of the fitted *gamma* and *lognormal* distributions. Using this test, which is determined to be the better fit? Discuss in your writeup, including the D statistic for each.

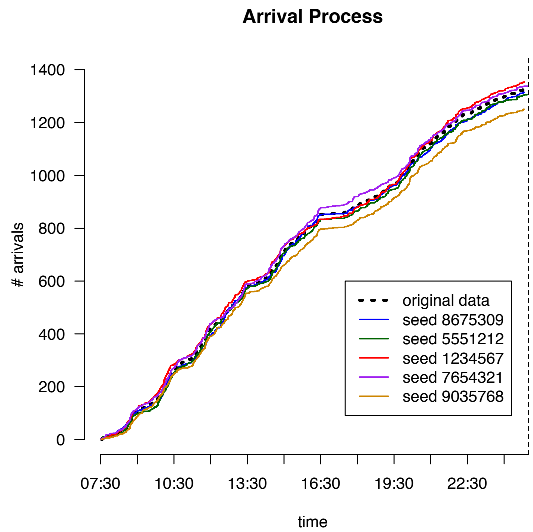
   (e) Update your `msq.py` service method such that each call returns one service time drawn from the best fitting distribution above.

---

3. Extend Algorithm 9.3.3 from the provided reading (and discussed in the provided lecture video) so that the algorithm will work in an event-driven fashion. That is, Algorithm 9.3.3 as given produces an entire list of arrival times. You must implement a method that, on each call, will remember the last arrival and produce the next interarrival using the cumulative event rate function produced by the provided *Eight Fifteen* arrival data. (You are not permitted to generate all arrival times in advance and then fetch from a list of times.)

   (a) Implement this algorithm as its own function in a separate Python program named `arrivals.py`. Your function should return a single interarrival time per function call.

   (b) Provide convincing graphical evidence that your arrival-process function, in isolation, produces output consistent with the collected *Eight Fifteen* arrival data. An example figure is provided below, showing five different single-day arrival process sequences superimposed on the original data.



(c) In your writeup, provide the algorithm in pseudocode, and provide and discuss the resulting figure.

(d) Modify your `msq.py` implementation to include this new implementation to generate interarrivals.

---

4. Experiment with different numbers of servers in the context of the arrival and service processes above. Provide meaningful statistics, with confidence intervals. In particular, use the method of batch means to produce 95% confidence intervals for the average sojourn time of customers in the system. (Since you do not know the true distribution of sojourn times here, to convince yourself of correctness you should compute several such intervals and look for overlap among the intervals. Note that, in this context, "several" need not be 100, as required in the batch means lab — three to five should suffice.)

---

5. Provide summary discussion that could be used to convince the *Eight Fifteen* management that a particular number of servers, fixed throughout the day, could be used to provide "reasonable" service based on the given arrival and service data. Remember that any recommendation will involve tradeoffs — additional servers do not come free of charge.

Submitting: Include your source code and write-up (including PDF and necessary supporting figures). Your write-up must include:

- discussion of the next-event `msq.py` implementation (event types, event-type algorithms, etc.);
- appropriate derivations, graphics, and discussion used in fitting the service data;
- appropriate discussion and graphics from the non-parametric approach for generating arrival data;
- numerical and/or graphical evidence of your experimentation of different numbers of servers;
- based on your simulation experimentation and analysis, convincing discussion of the number of servers to recommend to management with justification.

Upload your source code and PDF write-up to Lyceum.