

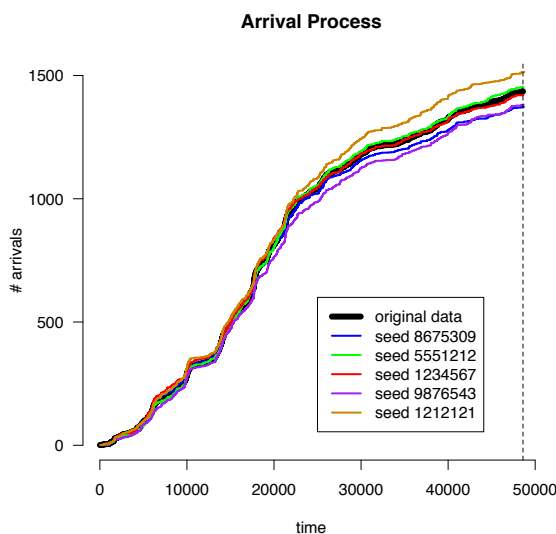
Overview: You will incorporate input modeling techniques to use the Tyler’s Grill data in a multiple-server queueing model, and then use output analysis, including batch means, to recommend a number of servers for Tyler’s Grill.

This work must be done on your own, not in a group.

1. Read Section 5.2.2 of the textbook. Then inspect the source code for the existing `msq` function from the `simEd` library, which implements a multiple-server queue model.

Benchmark the `msq` implementation. Do so by providing evidence that output from `msq` using one server matches output from the `ssq` implementation.

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 service data.
 - (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`), 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? Report the D statistic for each.
 - (e) Modify the service method in your `msq` code to use service times drawn from the best fitting distribution above.
3. Extend Algorithm 9.3.3 from the textbook to work in a next-event fashion. That is, Algorithm 9.3.3 as given produces a 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 Tyler’s Grill arrival data.
 - (a) Include this code as a function that can be passed as the interarrival function to `msq`.
 - (b) Provide convincing graphical evidence that your arrival-process function, in isolation, produces output consistent with the collected Tyler’s Grill arrival data. An example is provided below, showing five different single-day arrival process sequences superimposed on the original data.



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.)
5. Provide discussion that could be used to convince Tyler’s Grill management that a particular number of servers, fixed throughout the day, could be used to provide “reasonable” service based on the collected 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. Your write-up must include:

- discussion of the next-event msq implementation (event types, algorithms for those event types, etc.), and evidence benchmarking your msq against ssq;
- appropriate derivations, graphics, and discussion used in fitting the service data;
- appropriate discussion and graphics associated with the non-parametric approach for generating arrival data;
- appropriate numerical and/or graphical evidence of your experimentation of different numbers of servers;
- convincing discussion of the recommended number of servers to use based on your simulation experimentation.

Package your work into a tarball named similar to `cmssc326_final_bo4pz.tgz` and drop into the `final` folder within your shared Box folder for this course.