Steady-State Simulation

Chuljin Park& Seong-Hee Kim &
Barry L. Nelson

Hanyang University/Georgia Tech
/ Northwestern University

Output Analysis for Steady-State Simulation

- Suppose we are modeling a system for which steady-state analysis makes sense.
- Recall that the goal is to estimate *long-run performance* (as $T_E \rightarrow$ *infinity*), after the impact of the initial conditions have vanished.

Illustration: *M/M/1* Queue

- For this queue steady-state results are known:

$$w_Q = \frac{\lambda}{\mu(\mu - \lambda)}$$

$$L_Q = \frac{\lambda^2}{\mu(\mu - \lambda)}$$

Estimation

- If we had to estimate these via simulation, then we would observe two types of data within a replication
  - Y_i, the delay in queue of the *ith* customer (Arena calls this tally data)
  - $Y(t)$, the number in queue at time t (Arena calls this discrete-change or time-persistent data)
- These data would not be identically distributed because system congestion would be low in the beginning of the run.

Equivalent Definition

- If we could run an infinitely long simulation, then with probability 1…

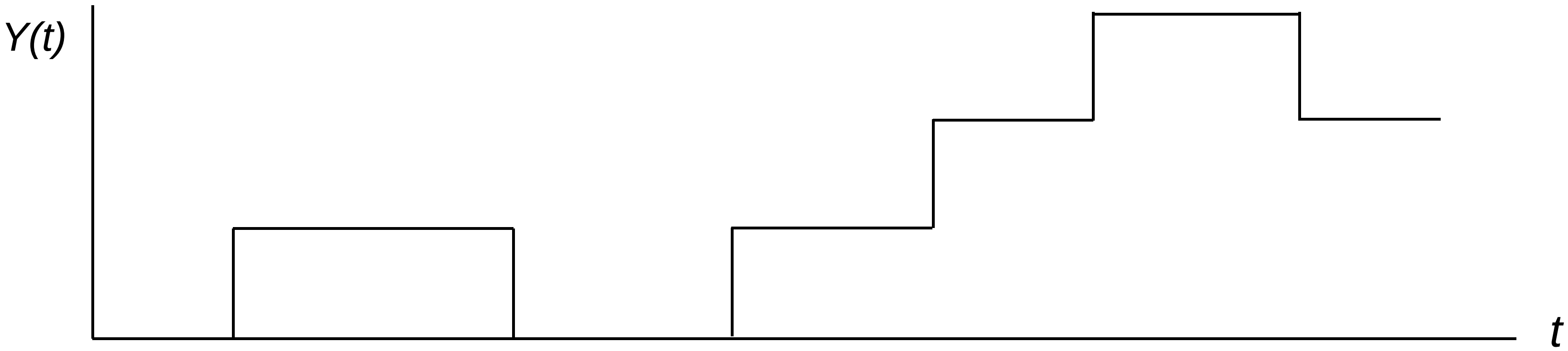
$$w_Q = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} Y_i \quad (\text{"Tally" data})$$

$$L_Q = \lim_{T_E\to\infty} \frac{1}{T_E}\int_0^{T_E} Y(t)\,dt \quad (\text{"Time Persistent" data})$$

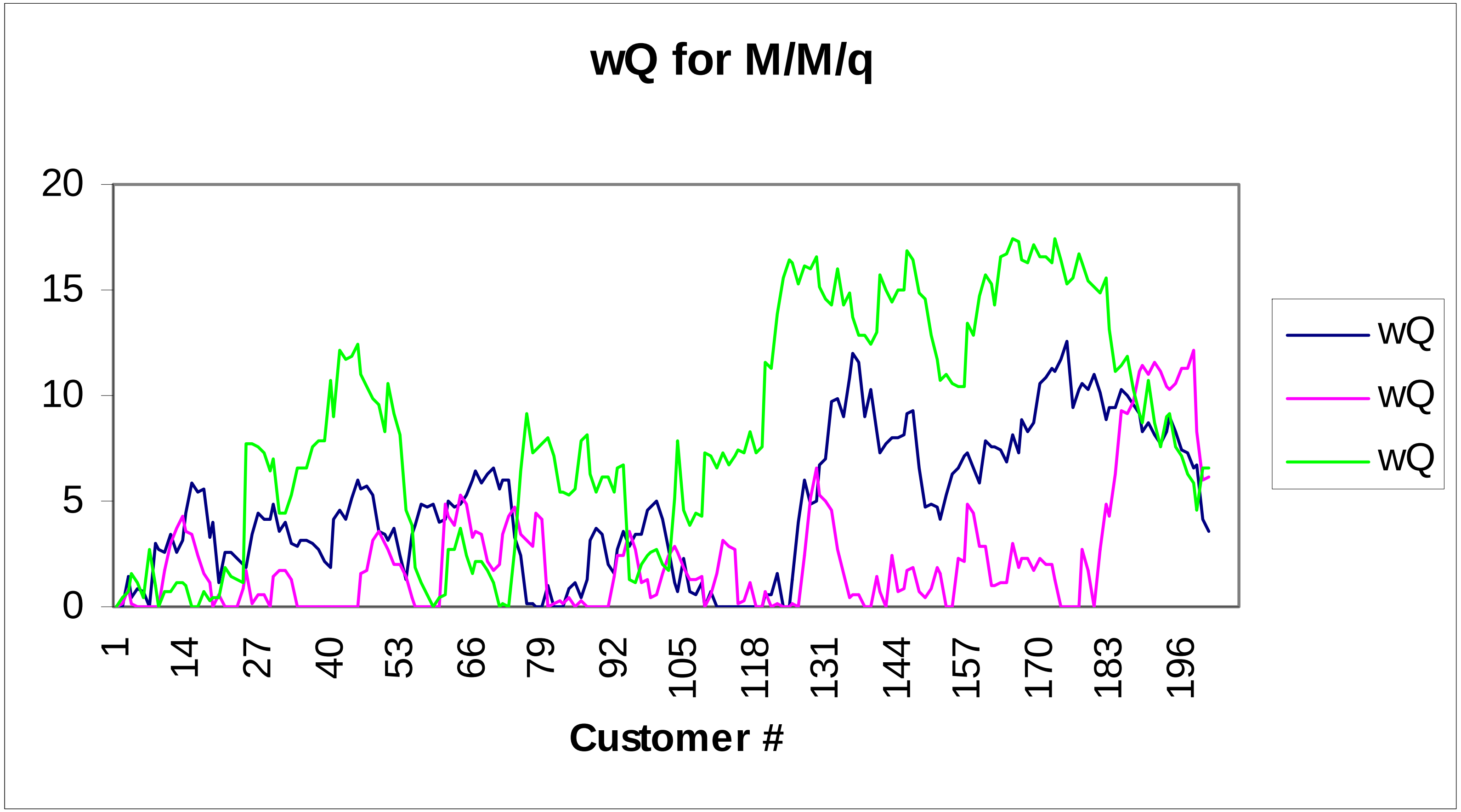
- The problem occurs because we must stop short of infinity.

Time-Persistent Averages

- For variables such as # in queue and # busy servers, the value and time spent at each value matter.
- The average is the area under the curve divided by the time interval.



M/M/1 with $\lambda=1$, $\mu=1.1$



Convergence

- Clearly there is an upward trend at the beginning.
- If there is a “steady state,” then the *true* mean delay in queue will stabilize, although the output process itself will always be variable.
- We want to estimate the long-run mean (or probability or quantile).

Impact of Bias

- If we ignore the “warm-up period” then our estimates will be biased (low, in this case).
- We cannot replicate away the bias; making many replications of a biased estimator gives us a highly precise estimate of the wrong value!

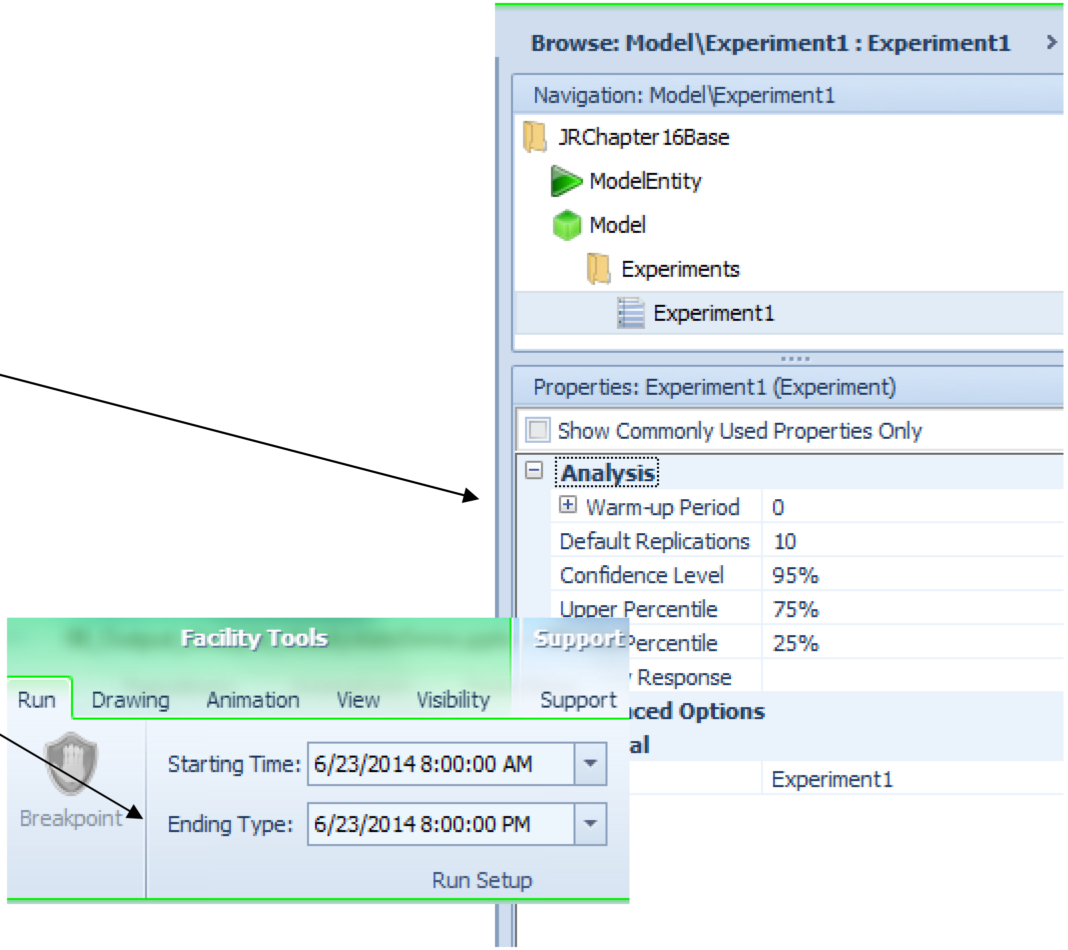
Replication-Deletion Approach

- The idea is to delete the data collected during the “warm-up period.”
- All data from time $[0, T_0]$ is discarded; our estimates are based on data collected during time period $[T_0, T_0 + T_E]$.
- We then do standard analysis for terminating systems using the truncated data.

Deleting Data in Simio

Amount of simulated time to delete

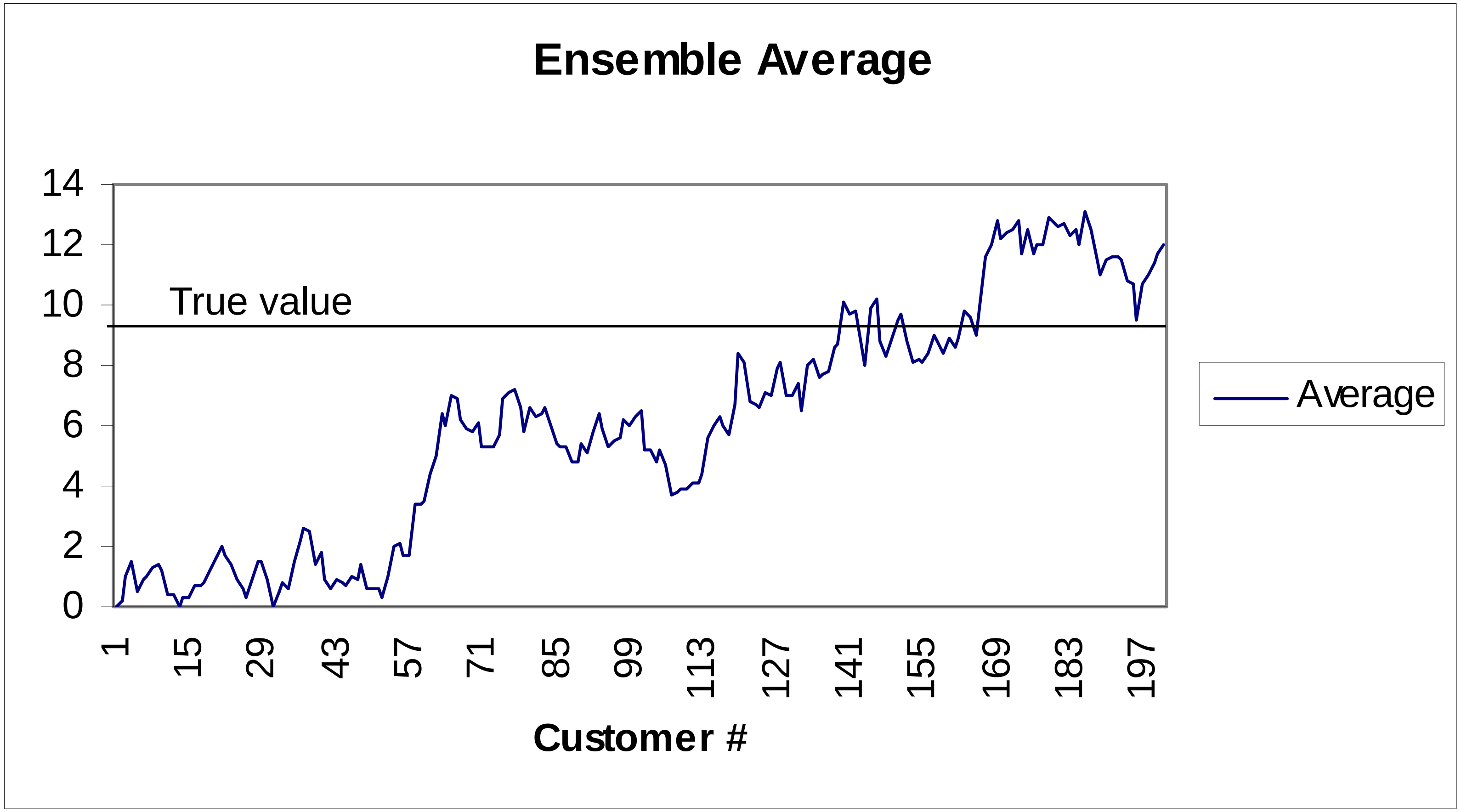
Total length of the replication, including the deleted time



Determining the Deletion Amount

- Any single replication can be misleading.
- Three approaches:
  1. Plot a number of replications (ok)
  2. Average across a number of replications (better)
  3. Average across and smooth (batch or moving average) a number of replications (best)

Smoothing *M/M/1* Output



Warm-Up in Simio

- Do not use “average” statistics by time *t* to determine the warm-up period.
- The number of servers busy at time *t*, the number of customers waiting at time *t*, or work-in-process at time *t* are fine.
- If waiting time for each customer available, it should be fine, too. However, average waiting time by the current time should not be used.

Design & Analysis (mean)

- If we use the replication-deletion approach, then means are handled just as in terminating simulation:
  - Use the sample average as the point estimator.
  - Use the standard confidence interval.
  - Plan the number of replications needed to make the c.i. short enough.

Mean

- Q: What is the expected delay of each job in the job shop?
- Y = average delay of all jobs from each replication
- Note that Y is a within-replication average we used before.
- Therefore, the analysis method for the mean is same as before.

More D&A: Prob/Quantile

- What is the probability that job wait time is greater than 1 hour?
- What is the 0.75 quantile of job wait times?
- For probabilities, the basic output *cannot* be the replication average, because then the probability depends on the length of the replication.

Key

- Our basic observations cannot be within-replication averages (Y) since our performance measures are in terms of individual job wait times not averages.
- Individual wait times are dependent….
- Bad news is that interval estimations (C.I.) are not valid any more for dependent data.
- But good news is that point estimators are still unbiased and consistent even for dependent data.

Key (continued)

- We get a point estimate from each replication for probability, and quantile of individual wait time.
- Then we have I.I.D. independent R point estimates and use them to get a point estimate and interval estimate for each performance measure.

Probability

- Q: What is the probability that a job waits more than one hour in queue in the job shop that runs 3 shifts/day?
- From each replication, get the total number of jobs processed (T) and the number of jobs whose delays are larger than one hour (N). Then Y = N/T.

$$\overline{Y} = \frac{\sum_{i=1}^{R} Y_i}{R}, \left[\overline{Y} \pm t_{1-\alpha/2, R-1} \frac{S}{\sqrt{R}} \right] \; where \;\; S^2 = \frac{\sum_{i=1}^{R} \left(Y_i - \overline{Y} \right)^2}{R-1}$$

Quantile

- Q: How much delay would 75% jobs experience in the job shop?
- From each replication, record the total number of jobs served (T). Then sort delays from the smallest to the largest. Then Y = T * 0.75 th smallest delay.

$$\overline{Y} = \frac{\sum_{i=1}^{R} Y_i}{R}, \left[\overline{\mathrm{Y}} \pm t_{1-\alpha/2, R-1} \frac{S}{\sqrt{R}} \right] \ where \ \ S^2 = \frac{\sum_{i=1}^{R} \left(Y_i - \overline{Y} \right)^2}{R-1}$$

Single-Rep Designs

- Since we are trying to estimate a limit, maybe we should make just 1 long rep.
  - Minimizes the bias of the estimates
  - Minimizes the amount of data we have to discard (do it only once)
- The only difficulty is that data within a replication are typically *dependent*.

The Effect of Dependence

- Dependence affects our variance estimators, and thus our confidence intervals.
- Positive dependence tends to make the confidence interval too short, convincing us we have a precise estimate when we don’t.

Details

Let $\bar{Y}$ be the sample mean of n observations.

$$\sigma^2(\bar{Y}) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathrm{cov}(Y_i, Y_j) \neq \frac{\sigma^2}{n} \text{ unless data are i.i.d.}$$

Our usual estimator for the variance of the mean,

$\frac{S^2}{n}$, estimates $\frac{\sigma^2}{n}$.

Batching

- Even when data are dependent, the dependence diminishes as the observations get farther apart in time.
- Thus, estimators computed from large enough "batches" should be nearly independent.

$$\underbrace{Y_1,\ldots,Y_d}_{\text{deleted}},\underbrace{Y_{d+1},\ldots,Y_{d+m}}_{\bar{Y}_1},\underbrace{Y_{d+m+1},\ldots,Y_{d+2m}}_{\bar{Y}_2},\ldots,\underbrace{Y_{d+(k-1)m+1},\ldots,Y_{d+km}}_{\bar{Y}_k}$$

Batching Notes

- Any statistic can be computed within a batch, including probabilities and quantiles.
- For continuous-time (time-persistent) data, batching is by time rather than by count.
- The key question is, how large do the batches need to be?

Simio Automated Batching

- When you make a “single” long replication, Simio attempts to form 95% CIs using the method of batch means for some statistics.
- Hueristic rule for batch size: tests the *lag-1 autocorrelation* < 0.1 between the batch means.
- When unsuccessful it will report NaN in the half width column.