More on Input Distributions

Importance of Using the “Correct” Distribution

Replacing a distribution with its mean

• mean interarrival time = 1 minute
• mean service time = 0.99 minute

Case 1: Exponential interarrival and service times
(M/M/1 queue, assume actual system)
Long-run average number in queue ≈ 98

Case 2: Constant interarrival and service times
Average number in queue = 0

Conclusion: One must also capture the variability in the input processes.

Using the Wrong Distribution

• Single-server queueing system with exponential interarrival times
• Weibull, Exponential, Normal, and Lognormal distributions were fit to 200 observed service times (see histogram).

Histogram of Service-Time Data.

Best Fit

Density / histogram overplot for the service-time data.
Simulation Results Based on 100,000 Delays

<table>
<thead>
<tr>
<th>Service-time distribution</th>
<th>Average delay</th>
<th>Percentage error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weibull</td>
<td>4.36</td>
<td>–</td>
</tr>
<tr>
<td>exponential</td>
<td>6.71</td>
<td>53.9</td>
</tr>
<tr>
<td>normal</td>
<td>6.04</td>
<td>38.5</td>
</tr>
<tr>
<td>lognormal</td>
<td>7.19</td>
<td>64.9</td>
</tr>
</tbody>
</table>

*Best fit

Lab Session Summary

Lab Session Summary – Model Building

- Create Modules, Connect Modules to generate process flow
- Define attributes, TNOW
- Define Input arrival rates, service rates
- Define resource schedule rules
  - Shifts, breaks, failure frequency, expression builder.
- Define Project parameters
- Resource utilizations, queue length, time in queue
- Define run parameters
  - Run length, time units, replications
- Define animation parameters
- Run simulation
- Generate Reports

Output Analysis

Lab Session Summary

What We'll Do ...

- Time frame of simulations
- Strategy for data collection and analysis
- Confidence intervals
- Comparing two alternatives
- Comparing many alternatives via the Arena Process Analyzer (PAN)
- Searching for an optimal alternative with OptQuesty

Types of Statistics Reported

- Many output statistics are one of three types:
  - **Tally** – avg., max, min of a discrete list of numbers
    - Used for discrete-time output processes like waiting times in queue, total times in system
  - **Time-persistent** – time-average, max, min of a plot of something where the x-axis is continuous time
    - Used for continuous-time output processes like queue lengths, WIP, server-busy functions (for utilizations)
  - **Counter** – accumulated sums of something, usually just counts of how many times something happened
    - Often used to count entities passing through a point in the model
Introduction

- Random input leads to random output (RIRO)
- Run a simulation (once) — what does it mean?
  - Was this run "typical" or not?
  - Variability from run to run (of the same model)?
- Need statistical analysis of output data
  - From a single model configuration
  - Compare two or more different configurations
  - Search for an optimal configuration
- Statistical analysis of output is often ignored
  - This is a big mistake — no idea of precision of results

Time Frame of Simulations

- Terminating: Specific starting, stopping conditions
  - Run length will be well-defined (and finite; known starting and stopping conditions)
- Steady-state: Long-run (technically forever)
  - Theoretically, initial conditions don't matter (but practically they usually do)
  - Not clear how to terminate a simulation run (theoretically infinite)
  - Interested in system response over long period of time
  - This is really a question of intent of the study
  - Has major impact on how output analysis is done
  - Sometimes it's not clear which is appropriate

Strategy for Data Collection and Analysis

- For terminating case, make IID replications
  - Run > Setup > Replication Parameters: Number of Replications field
  - Check both boxes for Initialize Between Replications
- Separate results for each replication — Category by Replication report
- Model 5-2, Daily Profit, Daily Late Wait Jobs; 10 replications

Strategy for Data Collection and Analysis (cont'd.)

- The confidence level of simulation output drawn from a set of simulation runs depends on the size of data set.
  - The larger the number of runs, the higher is the associated confidence.
  - However, more simulation runs also require more effort and resources for large systems.
  - Thus, the main goal must be in finding the smallest number of simulation runs that will provide the desirable confidence.

How Many Replications?

- Category Overview report will have some statistical-analysis results of the output across the replications
- Turn off animation altogether for max speed
  - Run > Run Control > Batch Run (No Animation)

Confidence Intervals for Terminating Systems

- Replication summary outputs as the basic data:
  - Sample Mean
  - Sample Standard Deviation
  - 95% Confidence Interval Express
  - Minimum Value
  - Maximum Value
- Possibly most useful part — 95% confidence interval on expected values
- This information (except standard deviation) is in Category Overview report
- One way to reduce the half-width of CI is to increase \( n \), the sample size (in this case the number of replications)
Half Width and Number of Replications

- Prefer smaller confidence intervals — precision
- Notation: \( n \) = no. replications
  \( \bar{X} \) = sample mean
  \( s \) = sample standard deviation
  \( t_{a/2, n-1} \) = critical value from \( t \) tables
- Confidence interval: \( \bar{X} \pm t_{a/2, n-1} \frac{s}{\sqrt{n}} \)
- Half-width = \( t_{a/2, n-1} \frac{s}{\sqrt{n}} \)
- Can't control \( t \) or \( s \)
- Must increase \( n \) — how much?

Half Width and Number of Replications (cont'd.)

- Set half-width = \( h_0 \), solve for \( n \): \( \frac{t_{a/2, n-1} s}{h_0} = \frac{z}{\sqrt{n}} \)
- Not really solved for \( n \) (\( t \), \( s \) depend on \( n \))
- Approximation:
  - Replace \( t \) by \( z \), corresponding normal critical value
  - Pretend that current \( s \) will hold for larger samples
  - Get \( n \approx \frac{z^2 s^2}{h_0^2} \)
  - Easier but different approximation:
    \( n \approx \frac{z^2 s^2}{h^2} \) (\( n \) grows quadratically as \( h \) decreases)

Interpretation of Confidence Intervals

- Interval with random (data-dependent) endpoints that's supposed to have stated probability of containing, or covering, the expected value
  - “Target” expected value is a fixed, unknown number
  - Expected value = average of infinite number of replications
- Not an interval that contains, say, 95% of the data
  - That's a prediction interval ... useful too, but different
- Usual formulas assume normally-distributed data
  - Never true in simulation
  - Might be approximately true if output is an average, rather than an extreme
  - Central limit theorem

Finding the Best System
(Comparing Alternative Solutions)

Introduction

- Simulations are typically performed to compare two or more alternate solutions
  - Consider comparisons among a small number of systems (say 2 to 30)
  - The method that is appropriate depends on the type of comparison desired and properties of the output data
  - Consider situations when there is no known functional relationship among alternative systems
Comparing Two Systems

Example: We evaluate two different “re-start” strategies that an airline can evoke following a major traffic disruption, e.g., a snowstorm in the Northeast. We want to minimize a certain cost function associated with the re-start.

- $X_i =$ cost from $i^{th}$ run of strategy 1. Assume:
  - $X_1, X_2, \ldots, X_n$ i.i.d. normal with unknown mean $\mu_X$ and unknown variance.
- $Y_i =$ cost from $i^{th}$ run of strategy 2. Assume:
  - $Y_1, Y_2, \ldots, Y_m$ i.i.d. normal with unknown mean $\mu_Y$ and unknown variance.

Goal: Obtain confidence interval for $\mu_X - \mu_Y$.

Two Systems (cont'd)

- How can we justify these assumptions?
  - Independent data by controlling random numbers between replications.
  - Identically distributed by performing replications under identical conditions.
  - Normally distributed data (since we add up many sub-costs to get overall costs for both strategies).

Paired $t$-Test

- $H_0 = \mu_X - \mu_Y = 0$
- Take $n$ observations from both strategies
- Set $D_i = X_i - Y_i$ for $i = 1, 2, \ldots, n$

$100(1-\alpha)%$ confidence interval:

$$\mu_X - \mu_Y \pm t_{1-\alpha/2, n-1} \sqrt{S_D^2/n + S_Y^2/m}$$

Paired $t$-Test Cont...

- Find the CI
- IF CI includes 0, $H_0$ is true. (Improved system is not better!)
  - $n_X \neq n_Y$
  - $n_X \neq n_Y$

Strong evidence that $\mu_X < \mu_Y$ System Y is better than X

Weak evidence that one system is better than the other

Strong evidence that $\mu_X > \mu_Y$ System X is better than Y

Modified $t$-Test (Welch Approach)

- Does not pair observations
- $n_X \neq n_Y$

- $n_X \neq n_Y$

$N_X = \frac{1}{n} \sum X_i$ and $N_Y = \frac{1}{m} \sum Y_i$ (Sample mean of $X_i$ and $Y_i$)

$S_X^2 = \frac{1}{n-1} \sum (X_i - N_X)^2$ (Sample variance of $X_i$)

$S_Y^2 = \frac{1}{m-1} \sum (Y_i - N_Y)^2$ (Sample variance of $Y_i$)

Modified $t$-Test (Welch Approach) Cont...

- $100(1-\alpha)%$ confidence interval:

$$\mu_X - \mu_Y \pm t_{1-\alpha/2, \nu} \sqrt{\frac{S_X^2}{n} + \frac{S_Y^2}{m}}$$

Where

$$\nu = \frac{(S_X^2/n + S_Y^2/m)^2}{\frac{S_X^4}{n(n-1)} + \frac{S_Y^4}{m(m-1)}}$$
## Comparing Two Alternatives

- Usually compare alternative system scenarios, configurations, layouts, sensitivity analysis
- Model 6-3
  - Model 6-1, but add file Daily Profit.dat to Statistic module, Output column, Daily Profit row
  - Saves this output statistic to this file for each replication
- Two versions
  - Base case – all inputs as originally defined
  - More-bookings case – Change Max Load from 24 to 28 hours
    (allow more bookings per day ... increase utilization, profit? Maybe)

## Reasonable but not-quite-right idea

- Make confidence intervals on expected outputs from each alternative, see if they overlap
- Base case: $492.63 \pm 13.81$, or $[478.82, 506.44]$  
  - More-bookings case: $564.53 \pm 22.59$, or $[541.94, 587.12]$  
- But this doesn’t allow for a precise, efficient statistical conclusion

## Compare Means via the Output Analyzer

- Output Analyzer is a separate application that operates on .dat files produced by Arena
  - Launch separately from Windows, not from Arena
- To save output values (Expressions) of entries in Statistic data module (Type = Output) – enter filename.dat in Output File column
  - Just did for Daily Profit, not Daily Late Wait Jobs
  - Will overwrite this file name next time … either change the name here or out in Windows before the next run
- .dat files are binary … can only be read by Output Analyzer

## Results:

- Confidence interval on difference misses 0, so conclude that there is a (statistically) significant difference

## Evaluating Many Alternatives (K > 2)

- Hypothesis Tests. Of course, it is well known that we will eventually reject $H_0: \mu_1 = \cdots = \mu_k$ if the sample size is large enough.
- Simultaneous Confidence Intervals
  - Bonferroni
  - Tukey-Kramer
- Ranking and Selection Procedures
- Multiple Comparison Procedures
R&S selects the best system, or a subset of systems that includes the best.
- Guarantee probability of correct selection.

MCPs treat the comparison problem as an inference problem.
- Account for simultaneous errors.

Both are relevant in simulation:
- Normally distributed data by batching.
- Independence by controlling random numbers.
- Multiple-stage sampling by retaining seeds.

Form simultaneous CI for \( \mu_i - \mu_j \) for all \( i \neq j \)

Systems are simulated independently

\( Y_{ij} \) i.i.d. outputs \( Y_{i1}, \ldots, Y_{in} \)

\[ \bar{Y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij} \] (Sample mean \( Y_i \))

\[ S^2 = \frac{1}{n-1} \sum_{i=1}^{k} \left( \frac{1}{n_i} - 1 \right) \left( \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij} - \bar{Y}_i \right)^2 \] (Sample variance of \( Y_i \))

Tukey's simultaneous confidence interval's are:

\[ \bar{Y}_i - \bar{Y}_j \pm t \sqrt{\frac{S^2}{n_i} + \frac{S^2}{n_j}} \]

where \( t = t_{\nu, \alpha} \) for any values of the \( n_i \)

\[ \nu = \frac{\nu^2}{\nu + k \sum_{i=1}^{k} (n_i - 1)} \]

Coverage \( \geq 1 - \alpha \) for any values of the \( n_i \)

Example

Compare 4 alternate designs for the architecture of a new computer system. Response time is the performance measure of interest. Smaller response time is preferred.

- For each architecture we take \( n = 6 \) replications
- Following is the summary data:

<table>
<thead>
<tr>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>S^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>72</td>
<td>85</td>
<td>76</td>
<td>62</td>
<td>100.9</td>
</tr>
</tbody>
</table>

Determine, with confidence 0.95, bounds on the difference between the expected response time of each architecture.

\[ t = 3.96 \text{ from the } t \text{- table} \]

For instance, a CI for \( \mu_1 - \mu_4 \) is 85 - 62 \( \pm 16 \) or, 23 \( \pm 16 \) milliseconds.

Since this CI does not contain 0, and since shorter response time is Better, we can screen out architecture 2 for further consideration!
**PAN Scenarios**

A scenario in PAN is a combination of:
- A program (.p) file
- Set of input controls that you choose
  - Chosen from Variables and Resource capacities – think ahead
  - You fill in specific numerical values
- Set of output responses that you choose
  - Chosen from automatic Arena outputs or your own Variables
  - Values initially empty … to be filled in after run(s)
- To create a new scenario in PAN, double-click where indicated, get Scenario Properties dialog
  - Specify Name, Tool Tip Text, .p file, controls, responses
  - Values of controls initially as in the model, but you can change them in PAN – this is the real utility of PAN
- Duplicate (right-click, Duplicate) scenarios, then edit for a new one
- Think of a scenario as a row

**PAN Projects and Runs**

A project in PAN is a collection of scenarios
- Program files can be the same .p file, or .p files from different model .doe files
- Controls, responses can be the same or differ across scenarios in a project – usually will be mostly the same
- Think of a project as a collection of scenario rows – a table
- Can save as a PAN (.pan extension) file

Select scenarios in project to run (maybe all)
PAN runs selected models with specified controls
PAN fills in output-response values in table
- Equivalent to setting-up, running them all “by hand” but much easier, faster, less error-prone

**Model 6-4 for PAN Experiments**

Same as Model 6-3 except remove Output File entry in Statistic module
- PAN will keep track of outputs itself, so this is faster

Controls – set up a formal 2^3 factorial experiment
- 2^3 = 8 Scenarios
- Also do Base Case

<table>
<thead>
<tr>
<th>Control (factor)</th>
<th>Low Level</th>
<th>High Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Load</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>Max Wait</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Wait Allowance</td>
<td>0.5</td>
<td>2.0</td>
</tr>
</tbody>
</table>

**Running Model 6-4 with PAN**

- Scenarios
  - Select all to run (click on left of row, Ctrl-Click or Shift-Click for more)
  - To execute, Run > Go or F5

**Statistical Comparisons with PAN**

- Model 6-4 alternatives were made with 100 replications each
  - Better than one replication, but what about statistical validity of comparisons, selection of “the best”?
- Select Total Cost column, Insert > Chart (or right-click on column, then Insert Chart)
  - Chart Type: Box and Whisker
  - Next, Total Cost; Next defaults
  - Next, Identify Best Scenarios
    - Bigger is Better, Error Tolerance = 0 (not the default)
    - Show Best Scenarios; Finish

**Statistical Comparisons with PAN (cont’d)**

- Vertical boxes: 95% confidence intervals
- Red scenarios statistically significantly better than blues
- More precisely, red scenarios are 99% sure to contain the best one
- Narrow down red set — more replications, or Error Tolerance > 0
- More details in book
A Follow-Up PAN Experiment

- From 2^2 factorial experiment, it's clear that Max Load matters the most, and bigger appears better
  - It's factor 1, varying between “-” and “+” in each scenario as ordered there, creating clear down/up/down/up pattern
  - Could also see this by computing main effects estimates
    - Consult an experimental-design text
- Eliminate other two factors (fix them at their base-case levels) and study Max Load alone
  - Let it be 20, 22, 24, ..., 40
  - Set up a second PAN experiment to do this, treated chart as before

A Follow-Up PAN Experiment (cont'd.)

- Here, profit-maximizing Max Load is about 30
- But Daily Late Wait Jobs keeps increasing (worsening) as Max Load increases
  - At profit-maximizing Max Load = 30, it's 0.908 job/day, which seems bad since we only take 5 wait jobs/day
  - Would like to require that it be at most 0.75 job/day ... still want to maximize Daily Profit
- Allow other two factors back into the picture ...

Searching for an Optimal Alternative with OptQuest

- The scenarios we've considered with PAN are just a few of many possibilities
- Seek input controls maximizing Daily Profit while keeping Daily Late Wait Jobs ≤ 0.75
- Formulate as an optimization problem:
  - Maximize Daily Profit
  - Subject to 20 ≤ Max Load ≤ 40
  - 1 ≤ Max Wait ≤ 7
  - 0.5 ≤ Wait Allowance ≤ 2.0
  - Daily Late Wait Jobs < 0.75
- Reasonable starting place – best acceptable scenario so far (the base case, actually)
  - Where to go from here? Explore all of feasible three-dimensional space exhaustively? No.

OptQuest

- OptQuest searches intelligently for an optimum
  - Like PAN, OptQuest
    - Runs as a separate application ... can be launched from Arena
      - “Takes over” the running of your model
    - Allows you to specify the input controls and the output (just one) response objective
  - Unlike PAN, OptQuest
    - Allows you to specify constraints on the input controls
    - Allows you to specify requirements on outputs
    - Decides itself what input-control-value combinations to try
    - Uses internal heuristic algorithms to decide how to change the input controls to move toward an optimum configuration
- You specify stopping criterion for the search

Using OptQuest

- Tools > OptQuest for Arena
  - New session (File > New or Ctrl+N or )
  - Make sure the desired model window is active
  - Select controls – Variables, Resource levels
    - Max Load, Lower Bound = 20, Upper Bound = 40, Conts.
    - Max Wait, Lower Bound = 1, Upper Bound = 7, Discrete (Input Step Size 1)
    - Wait Allowance, Lower Bound = 0.5, Upper Bound = 2, Conts.
  - Constraints—none here other than earlier Bounds
  - Objective and Requirement
    - Daily Profit: Response – Select Maximize Objective
    - Daily Late Wait Jobs: Response – Select Requirement, enter 0.75 for Upper Bound

Using OptQuest (cont'd.)

- Options window – computational limits, procedures
  - Time tab – run for 20 minutes
  - Precision tab – vary number of replications from 10 to 100
  - Preferences tab – various settings (accept defaults)
- Can revisit Controls, Constraints, Objective and Requirements, or Options windows via
- Run via wizard (first time through a new project), or Run > Start or
- View > Status and Solutions and
  - View > Performance Graph to watch progress
  - Can't absolutely guarantee a true optimum
  - Usually finds far better configuration than possible by hand