Students are encouraged to browse through chapter 4 of the textbook. A good chunk of it will hopefully review. For this particular LGA, **students should read**:

1. §4.1.3 — specifically the section and example in ***Serial Correlation***.
2. Read about **THE SQUARE ROOT RULE**
   i. §4.2.2 — specifically the section and example in ***Accuracy of Point Estimates***.
   ii. §4.3.2 — specifically the section and examples in ***Point Estimations***
3. §4.4 — specifically the section ***Covariance and Correlation***, definition 4.4.2 where the terms (and perhaps equations) should be familiar to you from a previous statistics course. Don't get bogged down in all the math on your way to theorem 4.4.2 but be sure to notice that the author describes ***mean squared orthogonal distance* (MSOD)** regression, not the conventional "least squares" regression. Both techniques estimate a best fit line to pairwise data (u_i, v_i). The vocabulary of definition 4.4.2 should be familiar to you from a previous statistics course, but notice the *Welford* styled iterative computation equations given in theorem 4.4.3. Finally, read the first portion of §4.4.2 ***Serial Correlation*** you may ignore the `acs` program description as it is a little beyond the scope of our course.

The following numbered questions should be split across your group and the solutions discussed during the next lecture period. Students should review the learning goals for the day, determine which are applicable to their questions and provide answers or commentary to their group members. When using the Internet to formulate answers (some questions may require this), keep track of **where** you find your information on the web. You may be asked for, and are expected to have (in Email-able form), URLs supporting your investigations.

1. The **square root rule** is presented in §4.2.2 and §4.3.2 of the reading. This question will expand on the type of experiment presented in example 4.2.8 (but we **won't** be using the dice game craps).

   First, retrieve your group's (or previous group's) solution to `lga-monte-carlo-probs.pdf` book question 2.3.5, which determined the probability that a random chord length of a circle is greater than the circle's radius. Modify the code (you may need to turn it into a function) so that it accepts the the number of replications (N) as an argument and returns the fraction of random chords out of N that had a chord length greater than the radius. Note that there are $N+1$ possible return values for this code: $\{\frac{0}{N}, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N}{N}\}$. **This will be called a single experiment**: calculating N random chord lengths and returning the fraction that had a length greater than the radius.[1]

   (a) Like example 4.2.8, you will run the experiment $n = 1000$ times. Each experiment returns an estimate of the probability in question. We collect 1000 of these together as a distribution and expect it to contain the true probability (namely the mean of the distribution). Calculate the standard deviation s_N of these $n = 1000$ samples and arrange your code to report N and s_N. (Now would be a good time to borrow the Welford API from your group's `lga-sim-statistics.pdf` solution!)

   Now, confirm the square root rule: vary N from 8 to 1600 by steps of 16 and collect the data points (N, s_N). Consider a scatter plot to ease visualization, how much larger should N be to decrease the standard deviation by half? Does your plot have the same shape as $f(x) = 1/\sqrt{x}$.

   (b) I've heard this question from students each semester:

   > Does the square root rule still apply if we leave N alone, but increase n?

   So let's get this answered for your group. Copy your code and make the change: keep $N = 25$, but let n (the number of experiments) range from 10 to 2000 (you decide on the increment). How much larger should n be to decrease the standard deviation by half? Investigate with some other values of N (perhaps 4, 16, 64, 256). Do you still see the square root rule in the results (again, scatter plots make the visualization easier).

[1] And it doesn't matter what value you use for the radius.

2. §4.3 of the textbook discusses continuous data histogram construction from simulation data (definition 4.3.1). Write a small utility program that accepts two command line arguments and produces two output files. Call it `binner`, it will be invoked like this:

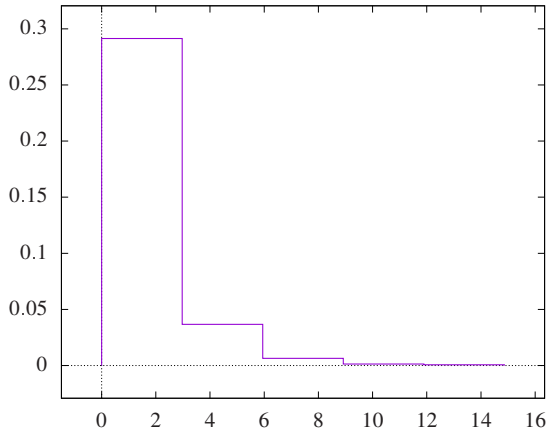
   ```
   $ ./binner 5 datafile.dat
   ```

   Where 5 is the number of bins to calculate for the histogram and `datafile.dat` is (not surprisingly) the file containing the textual data. `binner` produces two files, the names of which are determined by simply appending `.mid` and `.sq` onto the provided data file name.

   `datafile.dat.mid` will contain two numbers per line that are the bin **midpoint**, m_j, and **frequency** $\hat{f}(m_j)$ (height, definition 4.3.1). Run on the data file `datafile.dat` with 5 bins would create `datafile.dat.mid` containing

   ```
   1.48812   0.291375
   4.46024   0.0366742
   7.43236   0.00639274
   10.4045   0.00134584
   13.3766   0.00067292
   ```

   `datafile.dat.sq` will contain two numbers per line as well, but they will be the line segment endpoints of a properly drawn histogram. You can then tell any plotting program to simply graph a "connect the dots" plot and violà, you have a clean looking histogram. For clarity sake, consider the histogram drawn from `datafile.dat.mid`:

   The beginning line segments of the histogram are (**approximately**):

   $$\begin{aligned} (0,0) &\rightarrow (0,0.29) \\ (0,0.29) &\rightarrow (3,0.29) \\ (3,0.29) &\rightarrow (3,0.04) \\ (3,0.04) &\rightarrow (6,0.04) \\ (6,0.04) &\rightarrow (6,0.006) \end{aligned}$$

   which completes the first two bins of the histogram.
   The actual content of `datafile.dat.sq` would look like:

   ```
   0.00205842   0
   0.00205842   0.291375
   2.97418      0.291375
   2.97418      0.0366742
   ⋮            ⋮
   14.8627      0
   ```

   Demonstrate the results of your efforts with plotted histograms of `triangle.dat` for k (number of bins) 15, 30, and 65.

   Your `binner` utility should be buildable and runnable on alamode machines, and your learning group will use it in future LGAs — so make it good!

3. (a) From the reading of §4.4, you will recall that the equations for **conventional** least squares regression are

$$V = mU + b \qquad m = r\frac{s_v}{s_u} \qquad b = \bar{v} - m\bar{u}$$

where U and V are not necessarily from the paired (bivariate) data set of $\{(u_i, v_i)\}_{i=1}^{n}$.

The equations for MSOD regression (the preferred technique for all simulation results) are in theorem 4.4.2 of the text.

Write an API or object interface for these two techniques. The regression techniques can be contained in one interface or two different (but hopefully similar) APIs. **Hint!** begin with the your group's Welford $\bar{x}$ and s API from `lga-sim-statistics.pdf` !

(b) Now write a small application that reads a file of input data (textual, two values per line representing a (u, v) pair) and calculates the slope-intercept form coefficents for the two regression techniques.

(c) Run your new application on the data contained in `msod-vs-lr.dat` and `lr-vs-msod.dat` . For each data set (in separate graphs), plot the resulting two lines (labeled with a key of course) as well as a scatterplot of the paired data.

When you rally with your group again, you will compare the results of these two techniques in the context of computer simulation analysis.

4. §4.3 of the textbook discusses continuous data histograms and the estimation of $\bar{x}$ and s from them (definition 4.3.2, the definition of p_j that follows it, definition 4.3.3, the discretized equations that follow).[2]

Ideally, when presenting data in histogram form we keep in mind the need to provide quality visualization of data along with minimizing the quantization error in the result. Question 2 provides a command line utility for generating histogram midpoints and frequencies from a data file. **Your task** is to estimate a histogram's mean and standard deviation[3] given the **midpoints** and **frequencies** (heights) of its bins.

Write a command line utility named `histostats` that accepts as a command line argument a file with $(m_j, \hat{f}(m_j)$ histogram bin data (one pair per line, these are precisely the values described for question 2's `.mid` output file). For instance, a histogram defined by the $(m_j, \hat{f}(m_j)$ values of (`datafile.dat.mid`).

$$(1.48812, 0.291375), \quad (4.46024, 0.0366742), \quad (7.43236, 0.00639274),$$

$$(10.4045, 0.00134584), \quad (13.3766, 0.00067292)$$

would have $\bar{x} = 1.98446$ and $s = 1.65065$.

[2] Sometimes the raw data is not available to a simulation writer, and they must do all they can to best approximate parameters for probabilistic models. : (

[3] Technically, we are estimating the histogram's **original data set's** mean and standard deviation. We "short hand" this to saying these measures belong to the histogram.

5. (a) Use the SSQ solution for lga-uniform-arrivals.pdf question #2 and make the following change: incorporate your group's Welford $\bar{x}$ and s API from lga-sim-statistics.pdf so that you incrementally calculate $\bar{w}$ on each pass through the loop. The SSQ should run for 1000 jobs, with interarrival times from $Exponential(1)$ and service times from $Uniform(0.5, 1.5)$.

   The Experiment
   Run the simulation 101 times. **With the results of the first run**, calculate a 95% confidence interval for $\bar{w}$, the average time for a job to traverse the SSQ (also called the sojourn time). Statistical confidence intervals may have been a good bit in the past for you, so here is the equation:

$$\bar{w} \pm 1.960 \frac{s}{\sqrt{n-1}}$$

   Where s is the standard deviation from your Welford API and n should be 1000.

   With the results of the remaining 100 **independent runs** (use a different seed for each run), count how many of the 100 simulation $\bar{w}$'s fall within the 95% confidence interval, and how many fall outside of the CI. Don't be surprised if your results **don't agree** with your expectations.

   (b) Add a command line argument or copy and modify your solution to part a, so that the w_i sojourn times used are for the **last 1000** jobs processed by the SSQ, not all the jobs. Re-run **the experiment** using a total of 3000 jobs.

   Does the quality (predictive ability, accuracy) of your confidence interval improve?

Be prepared to share your observations and conclusions with your group in the next lecture.