A Distance-Education Exercise in Practical Stochastic Data Analysis

John P. Mullen, Ph.D.
Industrial Engineering Department
New Mexico State University
jomullen@nmsu.edu

Abstract

When students study topics such as queueing systems or stochastic processes, they may do well with theoretical aspects, but then have difficulty applying that theory to a real-world situation. This paper describes an online two-phase modeling exercise in which student teams carry out a number of practical steps, including data extraction, tests of homogeneity, dealing with censored data, performing goodness of fit and estimating confidence intervals for parameters, based on sets of simulated data from a stochastic process. This exercise is carried out online by teams that are geographically diverse, which gives the students additional experience in collaborating via a computer-moderated system, as well as the technical aspects of making a presentation online.

The exercise’s intent is to give distance-education students experience with issues that typically arise in practical projects, prior to embarking on a real-world applied project. As a result, they are better able to focus on the project itself without being distracted by these common aspects of analysis. Data is simulated using discrete-event simulation in MatLab, but analysis can be carried out in either MatLab or MS Excel. The MatLab code is available from the author.

Introduction

Practical projects, in which students apply the theory and methods from a course to a real-world situation, are commonly part of engineering courses. In face-to-face courses, there is usually a high level of informal interaction among students and between instructors and students during the project phase, concerning key issues of application, etc. However, in online courses, this degree of interaction can fall. Additionally, students have the complication of dealing with computer-moderated collaboration and presenting work in an online environment. The following describes an exercise developed with the intent of giving distance-education students a similar level of exposure and experience with these issues that they would get in a conventional face-to-face course. One key objective is that later they are better able to focus on the project itself.

Background

Distance education offers many advantages, but one key aspect affecting success is the degree of interaction among students in the class and between students and the instructor [1-3]. In addition, fostering student-to-student feedback is important because it tends to be more immediate feedback than instructor-student feedback [4-8]. Also, by fostering student-to-student interaction, some of the workload is shifted away from the instructor [8]. However, to be effective, this student-to-student interaction should involve a task requiring critical thinking [8, 9].
Important Note
A key aspect of this exercise is the nature of the students. This author has used variations of this method in six different graduate-level engineering distance courses. Most of the students in these courses are masters-level students who are working professionals. Students who have taken other engineering distances courses are more familiar with Canvas. As a result, much of the necessary knowledge is already known by some members of the class, but not distributed among all class members. This approach may not work with less-experienced and knowledgeable students without additional instruction and support. The specifics in this paper are for a queueing analysis course conducted via a Canvas Learning Management System (LMS). However, the procedures are described in a way that allows adaptation to other online environments.

Paper Organization
The first section below describes the exercise in general terms applicable to many online courses. The second section describes the details of the stochastic exercise, of interest to one teaching a stochastic modeling or analysis course. Following the descriptions is a brief discussion of technical issues in the online environment, followed by an assessment of the effectiveness of the exercise, conclusions, and recommendations.

Online Exercise Description
This exercise varies from course to course, but usually contains the following six elements.
1. Preliminary data collection
2. Team formation
3. Phase I of the analysis.
4. Instructor and class feedback on Phase I
5. Phase II of the exercise
6. Instructor feedback on Phase II
In some courses, there is only one phase because the technical aspects of data analysis are less involved than in courses involving stochastic analysis. These steps are detailed below.

Preliminary Data Collection
At the start of class, students fill out a questionnaire stating whether they have access to and their levels of familiarity with key software, with the Canvas LMS, and relevant topics, such as matrix math and statistical analysis. In addition, they report their approximate location, e.g. El Paso, TX, and fill out a matrix indicating which two-hour time blocks they can be available and which blocks they prefer to be available for synchronous activities. This last item is necessary because students are distributed all over the USA and sometimes in foreign countries and because different students have different work schedules.

Team Formation
The instructor forms teams of approximately five students each following these guidelines
- Each team is geographically diverse.
- At least one team member has access to each necessary software tool
- At least one team member has a moderate to high ability to use necessary software
- At least one team member has a moderate to high level of knowledge in necessary areas, such as data analysis and Canvas.
All members have at last a few hours each week in common on their availability tables for synchronous meetings. The instructor then sets up Canvas groups for each team. This feature gives each team a private work area in which they can use announcements, group email, a private discussion board, whiteboards, announcements, and chat rooms to collaborate. Details will vary by LMS, but a key objective is to provide a reasonable computer-moderated environment for each team.

**Phase 1 of the Analysis**

Student teams extract information from data sets, analyze it, and then present a report, including a one-page executive summary. The problem is at a difficulty level appropriate for a team of five and the one-page constraint on the summary forces the team to consider efficient ways of presenting information. Teams have two weeks to complete this phase.

**Student tasks**

Fig. 1 shows the instructions to the students. After setting up their teams, the first problem the teams face is extracting necessary information from the data sets. This seems to be the hardest part of the exercise. Students, then compute point and interval estimates of key parameters, assuming exponential distributions, for each set independently and for the pooled data. The final step of Phase I is a one-page executive summary and report.

All of the methods students need to employ were presented in class prior to the start of the exercise. In this exercise, they try to apply these principles to the given situation. The instructor provides self-checked exercises on key procedures and maintains a Q&A page on Canvas in which he posts student questions and their answers.

**Instructor and Class Feedback**

There are three sources of feedback

1. Each student in the class provides anonymous feedback to each team.
2. Each student confidentially rates the performance of every member of his or her team.
3. The instructor provides detailed feedback, as well as a score.

This part of the exercise takes about a week.

Anonymous feedback

The instructor first uploads each report to Canvas to see if they work properly. If a report does not work, the instructor determines the issue and reports that to the team so they can fix it. The instructor publishes reports that work via a wiki page so that all class members can view them.

Students then provide feedback by completing a graded anonymous survey for each team’s report. The survey is shown in Fig. 2. To get full credit, each student must complete the survey for each executive summary, including that from their own team. Students’ responses are not associated with their names. The contents of these summaries have no impact on the team’s grade, but provide additional feedback from the class.

Once the surveys are completed, each team receives a summary of the class feedback for its report. The Canvas LMS automatically generates these summaries. An example summary is shown in Fig. 3. For questions 1 and 2, the report provides a graphical tabulation of the...
responses. For questions 3 and 4, it provides a full list of all open-ended responses. In Fig. 3, some of the responses to questions 3 and 4 are not visible, but the actual reports are interactive HTML pages that allow students to scroll the response area and to see all of the responses.

Figure 1: Student Instructions for Phase I.

1. **Appearance:** What is your impression of the appearance of the report?
   - The report had a very nice, professional appearance.
   - The report was mostly very nice in appearance, but one or two things looked a bit rough.
   - The report was functional, but did not have a smooth finish.
   - The appearance of the report distracted from its content.

2. **Clarity:** Which of the following statements best describes how clearly the report conveyed information.
   - The report was very clearly written. I was able to understand each item as I read it.
   - The report was clearly written, but there were one or two places that I had a little difficulty with.
   - The report was understandable, but I had difficulty with several parts.
   - I had a lot of difficulty understanding this report.

3. **Liked Best:** What is the one feature of the report which you liked the best? Why?

4. **Liked Worst:** What is the one thing about this report that you liked the least? How would you fix it?

Figure 2: Survey for Class Feedback
Confidential ratings
Each student rates every member of his or her team by means of a Canvas quiz similar to that in Fig. 4. Students get points for completing the survey, but also get a chance to let the instructor know what happened in the process of doing the exercise. The average rating partially determines individual student’s scores. For example, if the team’s score is 80 points and a team member’s average rating is 90%, that individual’s score will be $80 \times 0.90 = 72$ points.

Although the instructor knows who gave whom what rating, this is treated as confidential information. Each member only knows his or her own average rating. If a student feels that the average rating is unfair, he or she can ask for a re-evaluation. The instructor will then ask the team to review who did what and submit a revised rating. In over 30 years of using this method, this writer has only been asked to do a re-rating twice.

Instructor Feedback
The instructor creates a PDF feedback file using Adobe Acrobat Pro, consisting of the one-page summary. He then makes comments directly on the summary using Acrobat’s commenting features and a digitizer pad. He also appends to that feedback file any part of the rest of the report requiring comment, as well as any supplemental material. For example, if the team’s answers are incorrect, the instructor provides the correct answers so the team can find and correct their errors. An example report with identification removed is shown in Fig. 5.
1. Select your team from the list below.
   *Each team is listed, together with the names of all members, as one option*

2. If all members of your team did a fair share of the work for this exercise, just enter “All 100%” below.
   
   If some did not do a fair share, list each member below, starting with yourself, along with your rating for that member. If a person did a fair share, rate that person at 100%. If a team member did less than a fair share, tell me what percent of a fair share that person did.

   For each team member, I will average the ratings from all members and use that to adjust their grade on the exercise. For example, if the team score is 80 points and a member’s average rating is 90%, that person’s grade will be $80 \times 0.90 = 72$ points.

   If you rate a person less than 90%, you need to tell me why that person deserved less than 90%. If you rate a person less than 90%, but give no reason, I will adjust that rating to 90%.

Figure 4: Peer Rating Form

The instructor uses the rubric in Fig. 6, to grade the reports. Then, if a student’s average rating is less than 100%, the instructor adjusts that student’s individual grade. The procedure in Canvas is:

1. When the assignment is created, it is designated as a group assignment, but the “Assign Grades to Each Student Individually” box is left unchecked. In this mode, grading and feedback will be by group.
2. Before grading, the assignment is muted. This way, students will only see the final grade.
3. Using the rubric in Fig. 6, the instructor enters the report grade and sends the feedback file to each team while in Speed Grader.
4. If any students are rated below 100%,
   a. The instructor returns to the assignment definition page and checks the box “Assign Grades to Each Student Individually”. The grades and feedback so far will remain intact, but the instructor can now adjust individual grades.
   b. The instructor returns to the Speed Grader and adjusts grades as necessary by overriding the original score and notifying each student with an adjusted grade of the reason. Also, any feedback at this time will go to the individual student.
5. After entering all grades, the instructor unmutes the assignment.

Of course, the details of this procedure will vary by LMS. In a previous LMS, the author did all the grade computations in a spreadsheet, then provided general feedback to each team, with individual grades, adjusted by peer evaluations, for each student.

**Phase II of the exercise**

Phase II is initiated after the teams have digested the feedback from Phase I. Because they use the same data in Phase II as in Phase I, this phase focuses more on analysis. Also, by the time the teams start Phase II, more analysis has been covered in the course. The instructions for Phase II are shown in Fig. 7.
Executive summary report Phase-1

What data? What if someone was on hold at the start, then entered service?

? Why?

Team name:
Members:
Assumptions:
- If the arrival time is not known the data is considered to be left censored.
- If the service time is not fully observed then the data is considered to be right censored.
- The queue discipline in the system is considered to be first come first serve.
- The last customer in the queue was always considered to be the first to renge.
- At the end of the observation period the customers still in service were considered to have received complete service at the specified time point.

Description:
The arrival rate ($\lambda_{MLE}$) was estimated as the ratio of number of observed arrivals ($r$) to the observed time ($T$) which is 60 minutes ($\lambda_{MLE} = \frac{r}{T}$). Similarly the service rate ($\mu_{MLE}$) was estimated as the ratio of total number of services that were completed ($1/r$) to the sum of observed service times ($1/T$). For estimating the confidence interval type-1 censoring was considered as the length of the observation period (60 mins) was decided in advance ($1/\lambda = \frac{x^2}{2} = \frac{x^2}{x}$). For the combined data sheet the $r$ and $T$ values are the summation of the individual data sheets values respectively ($t_{com} = t_{1} + t_{2} + t_{3}, T_{com} = T_{1} + T_{2} + T_{3}$). The point estimators and the confidence intervals were estimated similarly as the individual data.

The estimated values are as shown in the below table:

<table>
<thead>
<tr>
<th>Data Set</th>
<th>$\lambda$</th>
<th>$\mu$</th>
<th>90% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Set 1</td>
<td>15</td>
<td>4.517</td>
<td>9.64 ≤ $\lambda$ ≤ 22.49</td>
</tr>
<tr>
<td>Data Set 2</td>
<td>7</td>
<td>6.45</td>
<td>3.63 ≤ $\lambda$ ≤ 12.49</td>
</tr>
<tr>
<td>Data Set 3</td>
<td>10</td>
<td>6.24</td>
<td>5.8 ≤ $\lambda$ ≤ 16.33</td>
</tr>
<tr>
<td>Combined</td>
<td>10.67</td>
<td>5.59</td>
<td>7.9 ≤ $\lambda$ ≤ 14.14</td>
</tr>
</tbody>
</table>

Table 1

All your rate and interval estimates are correct. However, you should state intervals something like $9.64 \leq \lambda \leq 22.49$ and indicate in your table that rates are in arrivals/completions per hour.

Findings and observations:

In each of the cases, the arrival rate was greater than the service rate, hence there was a queue. In addition the arrival rates were to some extent skewed towards the lower limit of the confidence interval and the service rates were observed to be approximately at the center of the confidence interval limits.

For the last, $5.6 - 4 = 1.6$, but $7.7 - 5.6 = 2.1$

The number of reneges was negatively correlated with the differences between service rate and the arrival rate. Furthermore, the increase in the arrival rate caused the number of customers entering the queue to increase as well.

Good, but you had a few misses. Also, didn’t you discover anything in the process of doing this exercise?

You had XLSX files, which are not accessible by all students.

Figure 5: Example of Feedback on Phase I
Although the analysis in this phase is more complicated, the overall exercise is less difficult than the first phase because the teams are better organized and are more comfortable with the technical details of collaborating and presenting reports in Canvas. Consequently, this phase only takes one week.

**Instructor feedback on Phase II**

This part is much the same as for Phase I. Because of limitations of time, students are not required to read and review the summaries, though many do and provide informal feedback via the discussion board. Students do rate their team members and average ratings still affect individual grades for Phase II. Also, the same rubric is used in Phase II as in Phase I

**Specific Details of the Exercise’s Content**

The following specific details would be of interest to one teaching a course in queueing system modeling and analysis, whether face-to-face, or online. With slight modification, they would also be of interest to one conducting other courses in stochastic modeling or analysis.

**Model Scenario**

The setting is a small travel agency, which has $c$ clerks and $K$ telephone lines. Customers who call do one of the following; 1) connect directly to a clerk, 2) connect, but go on hold if all clerks are busy, or 3) not connect because all lines are busy. The arrival process is stochastic, but with
unchanging parameters and service times are identically distributed for all servers. Customers on hold wait for some random amount of time before hanging up (reneging), unless they connect to a server first.

Because the instructor had used this basic scenario to introduce a number of concepts in class, students are familiar with it. However, for this exercise, students are not told the nature of the arrival or service processes. Instead, they receive three sets of data from a simulated transaction logger with starts some time after the agency opens up and stops logging one hour later. While the data sets for each team are from the same version of the model, different teams get data generated from this model with different arrival and service processes.

<table>
<thead>
<tr>
<th>In the first phase, you started an analysis of three sets of data. In this phase, you will finish that analysis. This phase consists of:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Making any necessary corrections to Part I</td>
</tr>
<tr>
<td>2. Testing the hypothesis that the data sets can be combined for arrivals and separately for service times.</td>
</tr>
<tr>
<td>3. Testing the two distributions for likely candidates (Erlang-k or exponential).</td>
</tr>
<tr>
<td>4. Finding point and 90% confidence interval estimates for ( \lambda ) and ( \mu ), consistent with your results in the steps above.</td>
</tr>
<tr>
<td>5. If your ( k ) estimate is barely a maximum-likelihood point, doing a sensitivity analysis for values of ( k ) with a similar likelihood.</td>
</tr>
<tr>
<td>6. Adding the new results to your earlier report, and</td>
</tr>
<tr>
<td>7. Commenting on anything new that strikes you, any interesting thing you learned, and how confident you are of these results.</td>
</tr>
</tbody>
</table>

**Figure 7: Instructions for Phase II**

**Data Generation**

The data files are generated using a simulation of a \((M/E_k/c/K)\) queueing system. The discrete-event simulation is based on one in [10], with numerous enhancements, and is implemented in MatLab. The user can specify run length, the time logging starts, the mean interarrival time, the mean service time, the Erlang shape parameter \((k)\), the number of servers \((c)\), and the number of telephone lines \((K)\). This model also allows for the possibility that customers on hold may renege, so the mean time until reneging can be specified. The distribution of time before hanging up while on hold is exponential. Finally, the user selects a random number seed for repeatability.

The program simulates three independent runs and produces an ASCII file containing the data for each run. It also produces a summary report showing all inputs, as well as the correct results for a Phase I analysis. The program is available from the author.

The data sets represent the output of a data logging system which records changes of state in a simulated agency. Each data set represents a single run and consists of a number of rows having the following structure:

\[
<\text{time}> \ <\text{N}(t^+)> \ <\text{code}> \ <\text{server}>
\]

Time is recorded to hundredths of a minute. \(\text{N}(t^+)\) is the number in the system immediately after the event that occurred at \(<\text{time}>\). The \(<\text{code}>\) tells you what happened. If a server is involved, that is noted in the last column. Event codes are shown in Table 1. Fig. 8 depicts a data file.
Table 1: Data Logging Event Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>System status – this is the first entry at time zero.</td>
</tr>
<tr>
<td>1</td>
<td>An arrival has just occurred.</td>
</tr>
<tr>
<td>2</td>
<td>A call has just been completed</td>
</tr>
<tr>
<td>3</td>
<td>A caller on hold just hung up.</td>
</tr>
<tr>
<td>4</td>
<td>A caller on hold has just been connected to a server</td>
</tr>
<tr>
<td>5</td>
<td>End of session (EOS) – the last entry of the observation period.</td>
</tr>
</tbody>
</table>

- If the status code = 0, the last column is the system capacity (number of telephone lines).
- For a customer entering, the “server” is -1 if the customer goes into the queue. For reneges, the “server” is zero. Otherwise, the last column is the server number.
- When a customer completes a call and someone is in the queue, you will see an entry for the completion (code 2) followed by a code 4 entry for the same time.

Student Analysis in Phase I

Data Extraction
The first major task the teams have is extracting the information from the data. One method students use is to import the data into MS Excel and proceed from there. Others choose to use MatLab. An instructor-provided MatLab script file (Fig. 9) displays the data and plots N(t). This file also gives students wishing to use MatLab examples of how to open and read ASCII files. However, even students using MS Excel use this script file to see N(t), which helps them visualize the information in the data and provides a means of checking their methods.

Phase I analysis
In this phase, students assume that the interarrival times and service times are exponentially distributed. The maximum-likelihood estimator for a rate parameter, e.g. λ, in this situation is

\[ \lambda_{MLE} = \frac{r}{T} \]  \hspace{1cm} (1)

where \( r \) is the number of observed events and \( T \) is the total time during which events could have occurred [11]. Students learned of Eq. (1) earlier, saw several examples and used it in exercises, but for this analysis, they needed to work out the following details.

- For the arrival rate (\( \lambda \)), \( r \) is the number of arrivals and \( T \) is the length of the observation period less the times during which all lines were busy.
- For the service rate (\( \mu \)), \( r \) is the number of completed service times and \( T \) is the sum of all customers’ service times, including that of those still in service at the end of the observation period.
- Although not required in the exercise for this course, for the time until balking rate (\( \theta \)), \( r \) would be the number of customers who hung up while on hold and \( T \) would be the total of all hold times, including that for customers who hung up, that for customers who entered service from hold and that for those on hold at the end of the observation period.
Finally, when combining data from all three sets, for each parameter, $r_c = r_1 + r_2 + r_3$ and $T_c = T_1 + T_2 + T_3$.

Common problems at this point include mixing up rates and times, mixing up hours and minutes, and leaving out units.

| 0.000000e+00 | 2.000000e+00 | 0.000000e+00 | 1.000000e+00 |
| 1.690000e+00 | 1.000000e+00 | 2.000000e+00 | 2.000000e+00 |
| 4.510000e+00 | 2.000000e+00 | 1.000000e+00 | 2.000000e+00 |
| 5.810000e+00 | 1.000000e+00 | 2.000000e+00 | 1.000000e+00 |
| 7.660000e+00 | 2.000000e+00 | 1.000000e+00 | 1.000000e+00 |
| 9.970000e+00 | 1.000000e+00 | 2.000000e+00 | 2.000000e+00 |
| 1.257000e+00 | 2.000000e+00 | 1.000000e+00 | 2.000000e+00 |
| 1.630000e+00 | 3.000000e+00 | 1.000000e+00 | -1.000000e+00 |
| 1.775000e+00 | 4.000000e+00 | 1.000000e+00 | -1.000000e+00 |
| 1.868000e+00 | 3.000000e+00 | 2.000000e+00 | 1.000000e+00 |
| 1.868000e+00 | 3.000000e+00 | 4.000000e+00 | 1.000000e+00 |
| 1.941000e+00 | 2.000000e+00 | 3.000000e+00 | 0.000000e+00 |
| 2.750000e+00 | 1.000000e+00 | 2.000000e+00 | 2.000000e+00 |
| 3.076000e+00 | 0.000000e+00 | 2.000000e+00 | 1.000000e+00 |
| 3.247000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |
| 3.565000e+00 | 0.000000e+00 | 2.000000e+00 | 1.000000e+00 |
| 4.046000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |
| 5.339000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |
| 5.940000e+00 | 2.000000e+00 | 1.000000e+00 | 2.000000e+00 |
| 5.977000e+00 | 3.000000e+00 | 1.000000e+00 | -1.000000e+00 |
| 6.000000e+00 | 3.000000e+00 | 5.000000e+00 | 0.000000e+00 |

Figure 8: Example Data File

Another task in Phase I is finding confidence intervals. Students previously learned of the following $(1-\alpha)\times100\%$ approximate confidence interval for an exponential rate parameter, e.g. $\lambda$, in the event of Type I censoring[11]

$$\frac{\chi^2_{\alpha}r + 1}{2T} < \lambda < \frac{\chi^2_{1-\alpha}r + 1}{2T}$$

Generally, once the teams work out the $r$ and $T$ values for Eq. (1), dealing with Eq. (2) is not an issue.

**Phase II tasks**

By the time of the second phase, students have been introduced to the Erlang distribution model, methods to estimate $k$, the so-called Goodness of Fit Tests$^3$, the Kruskal-Wallis test of homoscedasticity [12] and adjustments needed in the presence of censoring.

**Homoscedasticity**

The first step in Phase II is to determine whether information from the three data sets can be combined. The Kruskal-Wallis test is used because it does not presume a particular distribution. Because the three data sets for each team are created by the same model, the tests generally conclude that the data may be combined. Once the K-W test indicates the data can be combined, the rest of the analysis is performed on the combined data only.
Characterization
In the queueing analysis course, students only check for the exponential and Erlang distributions because the queuing models presented in the course do not employ other distributions [13]. If the distribution is not one of these two, a general queueing model would be used. In other stochastic analysis courses, other models might be included. For the Erlang distribution, the MLE of the rate is the same as for the exponential. To find $k$, the general likelihood model for censored data is used.

$$L[k;\hat{\mu}_{MLE};(t_i)] = \prod_{i=1}^{r} f(t_i; k, \hat{\mu}_{MLE}) \prod_{i=r+1}^{n} \left[ 1 - F(C_i; k, \hat{\mu}_{MLE}) \right]$$ (3)

where the observations are ordered so the first $r$ are observed and the remaining $n-r$ are right-censored at time $C_i$. In practice, the natural log is taken on both sides of this equation. For the Erlang, the common tactic is to use its relationship to the gamma distribution to find $F(t)$, and then solve for the maximum likelihood numerically. Another approach is to use MatLab’s `gamfit()` function, but it usually produces non-integer estimates of $k$, so at some point teams end up employing the numerical likelihood method.

Checking data fit to the model
Students use the Kaplan-Meyer (Product-Limit) method to determine the empirical distribution function, $F_o(t)$ [12]. Then, they perform a Goodness of Fit test on the observed values. In this exercise, the number of observed values is generally in the 40 to 50 range, so students use the appropriate one of Lilliefors adaptations [14, 15] of the Kolmogorov-Smirnov Goodness of Fit Test [16]. Because of the moderate sample size, they perform this test at the $\alpha = 0.10$ level.

Online Technical Issues
The author has implemented this sort of exercise in a number of forms using different LMS systems and support software. Below is a list of current issues and steps to counter them.
Document Stability: Until recently, students used a variety of word processing programs, but recently they use multiple versions of MS® Word. At times, a document that is edited in different versions of MS® Word sometimes became unstable. In addition, students with some older versions of MS® Word could not open DOCX files.

Analysis Tools: In many courses, MS® Excel is adequate to do all of the analysis. Additionally, students seem to prefer being able to see the data in table form. However, in some classes, students are required to obtain MatLab because of its superior ability to produce contour plots and more advanced statistical analysis. As a result, students may use either or both, using the MatLab – MS® Excel interface.

Publishing reports: The one way in which different LMSs seem to differ the most is in the way student reports are published. In some LMSs, reports can be prepared in the student team’s private area, then published in a way that all students can see them. Canvas has that capability, as long as the document does not have any links to local files. However, if there are such links, publishing does not work.

Collaboration tools: At the minimum, students should have a private folder for files or a capability to attach files to posts in a private threaded discussion area. There should also be a way to hold synchronous meetings.

To deal with stability and compatibility issues, there had been a course requirement to use only DOC and XLS files for collaboration and presentations. However, NMSU has recently made MS® Office 365 available to all students and, in recent courses, students have all agreed to use MS® Office 365. As a result, these issues no longer seem to be a problem at NMSU.

Standardization of analysis tools is important for at least three reasons. Firstly, all students can access all examples provided by the instructor. Second, collaboration is difficult if different members have different tools. Finally, having a standard set of tools means all students can access the files in the reports. On campus, this is not a problem because students can use the departmental computer lab, but in distance courses, some students may have to purchase software. At this time, the only software the author uses that students have to buy is MatLab. For all other analysis, the author employs MS® Excel or free software, such as LINGO®.

At this time, the following publishing method is used as a workaround. Students collaborate via Canvas to write their report. Then one team member puts all files associated with the report into a single folder on his or her personal computer. That person then creates a main index file in MS® Word to index and link to all other files in the report. The student then saves it as a filtered web page and zips all files up into a single repository. After team members verify that the zip works, they submit it as their report. Depending on the LMS, this procedure may be unnecessary. However, it would probably work with any LMS that does not provide a publishing function for multi-document reports.

At this time, Canvas seems to offer adequate collaboration tools with a minimum of action required by the instructor. However, in the past, some tools required extra instructor action. At times, the author has encouraged students to use Skype® and Cmap Tools® to alleviate some LMS deficiencies. Recently, NMSU has incorporated Adobe Connect® into Canvas, which can be used by teams for conferencing and presentations, but sessions have to be set up by the instructor.
Exercise Impact

This author has used variations of this exercise in numerous courses over the last eight years. Consistently, the following items have been noted:

- Compared to distance students’ work prior to initializing this kind of exercise, student projects tend to be higher in quality along all dimensions.
- Students often indicate in course-end reviews that while the exercise is difficult, it is very useful, not only in the course, but also in their professional life.
- Most of the students with poor ratings in Phase I improve greatly in Phase II and usually end up being rated at 100% in that phase.
- Although students are free to select their own teams for the semester project, about half of the students choose to have a geographically-diverse team.
- There is evidence of collaborating in other activities once the teams have formed.

The last point sometimes makes it necessary to remind students of which activities are to be individual efforts and which activities can be group efforts.

Conclusions and Recommendations

While distance education offers many advantages, especially in graduate school, there is a tendency to have less interaction. This can negatively affect student performance, but it is possible to improve interaction and, consequently, student performance, with instructor-guided exercises. Additionally, those exercises can be a means of familiarizing students with key procedures in a course and the details of using the LMS without adding greatly to the instructor’s workload.

The exercise above is from a course on queueing system modeling and analysis. Its content was driven by the kinds of mistakes students tended to make in course-end projects. Of course, these elements of the exercise may not be appropriate for other courses.

Acknowledgement

The author would point out that a few years ago such projects were much more difficult to accomplish in Canvas and wants to thank Canvas support for responding to faculty input on this and other matters.

References


Endnotes

2 Instructure. 6330 South 3000 East, Suite 700, Salt Lake City, UT 84121. info@instructure.com. 1-800-203-6755.
3 Actually, such tests only determine badness of fit because $H_0$ is that $F = F_0$ and there is no way to calculate the Type II error. That is, $\alpha$ only tell us the probability of falsely rejecting $F_0$, not the probably of falsely accepting $F_0$.
4 For small to medium-sized samples, $\alpha$ values of 0.2 or 0.1 are used in order to keep the Type II error down and reduce the risk of falsely accepting $F_0$ as an appropriate distribution.
5 LINDO Systems, Inc. URL: www.lindo.com
7 Ihmc Cmap Tools. URL: cmap.ihmc.us. Free for educational use.
8 Adobe Connect. URL: www.adobe.com/products/adobeconnect.com. This one is not free.

John P. Mullen
Dr. Mullen is currently an associate professor in the Department of Industrial Engineering at New Mexico State University. His primary area of interest is operations research, with an emphasis on wireless ad-hoc networks and other stochastic systems.