

Statistical Data Mining

Applications in Student Learning

A Data Visualization Approach

By

Andrew (Andras) Moldovan

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE HONOURS

In

The Irving K. Barber School of Arts and Sciences
(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

April 2013

© Andrew Moldovan 2013

Abstract

With the advent of online education, an abundance of information has become available for study and analysis. A previous system had been implemented for 1st Year Physics classes at UBC Okanagan allowing professors to assign quizzes and assignments for students to complete online. Student answers were given in the form of natural language text and evaluated by a parser which would return whether or not the student was correct. While this system had been in use for over two years, it had never been properly analyzed. This Honours Thesis undertook to accomplish two things: first, to create a tool which professors could use to understand how their students are faring in real time by providing them a collection of charts, alerts, and real time statistical information (the Dashboard). Second, to analyze the aggregate data derived from the students to better understand how students learn. To accomplish this goal, computer code was written to parse and interpret each and every incorrect answer to understand where the student went wrong in answering the question. The Dashboard has been fully implemented and is in use by professors of 1st year physics classes, who have already used the system to change their assignments and quizzes as suggested by the alerts. On the aggregate side, it was found that most student attempts are incorrect due to the value of the answer (95%), as opposed to incorrect units (39%) for example. However, 10% of incorrect answers stem from missing negatives and 14% are within 5% of the correct answer, which shows that many students are actually close to solving the problem correctly. The numbers do not sum to 100% because each incorrect answer maybe incorrect due to multiple reasons. Overall a great deal of work has been accomplished in creating the online dashboard and studying the data in detail, and this work can be used by professors for years to come.

Table of Contents

Abstract	1
Table of Contents	2
Table of Figures	5
List of Tables	6
Background	7
Objective of this Thesis	7
Data Collection	7
Terms and Definitions	8
<i>Program</i>	8
<i>Test/Assignment</i>	8
<i>Question (Q)</i>	8
<i>Question Part (QP)</i>	8
<i>Missing Negative</i>	8
<i>Wrong Number Value</i>	8
<i>Within 5%</i>	9
<i>Wrong Units</i>	9
<i>No Answer</i>	9
Methodology	9
<i>Parsing Student Answers</i>	9
<i>Initial Results</i>	10
<i>Equations</i>	10
Implementation	11
<i>Professor Dashboard</i>	11
Report Generation	11
Student Information Report	12
Guesser	12
Button Masher	12
Statistical Ranking of Students.....	13
Courses Report	13
Assignment Report	13
Daily Report	14
Live Data	14
Dashboard Walkthrough.....	15
Main Page.....	15
Course Page.....	16

Course Page Continued	17
Assignment Page	18
Students Page	19
Individual Student (Below Average)	20
Individual Student (Good, but “Guesser”)	20
Individual Student (Good)	21
Results and In Depth Analysis.....	21
<i>Course Differences.....</i>	22
t-tests	22
Question Part Breakdown.....	24
Histogram of Compartments	26
Assignment Breakdown	27
Number of Students.....	28
Overall Time Charts.....	29
Time Chart Breakdown	30
Average Number of Attempts vs. Correct Rate for a Question Part.....	32
Average Number of Attempts for a User vs. Overall Grade of User	34
Average Number of Days started a Test before Due Date vs. Overall Grade	36
Scatter Plot of Missing Negatives	38
Scatter Plot of Wrong Number/Value.....	39
Scatter Plot of Within 5%.....	40
Scatter Plot of Wrong Units	42
Most Common Answers.....	43
Conclusions	44
Detailed Analysis of a Problem Question Part.....	45
<i>Popular Answers:.....</i>	45
Future Work	47
<i>Equations.....</i>	47
<i>Automatic Hints.....</i>	47
<i>Alerting Students about Guessing in Real Time.....</i>	48
Conclusion	48
Acknowledgments	50
Appendix A. Technical Summary	51
<i>Open Source JavaScript Libraries.....</i>	51
Google Charts	51
HighCharts.....	51
Bootstrap	51
<i>Dashboard Report Generation</i>	51
getAssignmentReport.jsp	51
allQPwrongreasonchart.jsp.....	52
getQPstuff.jsp.....	52
getQuestionPartAnswersEACHCOURSE ()	52
getQuestionPartAnswersIndividualReturnNumber ()	53

getStudentInfo.jsp	53
getDailyReport.jsp	53
getCoursesReport.jsp.....	53
realtimedata.js	54
livedatainfo.jsp.....	54
getlatestdata.jsp.....	54
index.php.....	54
script.js.....	55
Running the Report Generation.....	55
<i>Student Answer Parsing</i>	56
getAnswersInd.jsp.....	57
<i>equations.jsp</i>	58

Table of Figures

Figure 1. A screenshot of the main page of the dashboard.....	15
Figure 2. A screenshot of the top half of the “course” page of the dashboard.	16
Figure 3. Bottom half of the course page.	17
Figure 4. Screenshot of the “assignment” page on the dashboard.....	18
Figure 5. Screenshot of the “student’s page on the dashboard.	19
Figure 6. A “Below Average” student view.	20
Figure 7. Screenshot of a “Good” student, but with many alerts.....	20
Figure 8. Screenshot of a “Good” student.	21
Figure 9. Chart showing the question part breakdown for course A.	25
Figure 10. Chart showing the question part breakdown for course B.....	25
Figure 11. Showing the histogram of what compartment student answers fall into for course A.	26
Figure 12. Showing the histogram of what compartment student answers fall into for course B.	26
Figure 13. Showing the assignment breakdown for course A.	27
Figure 14. Showing the assignment breakdown for course B.	27
Figure 15. Number of students who have attempted each assignment for course A.....	28
Figure 16. Number of students who have attempted each assignment for course B.....	28
Figure 17. Showing the overall time chart for course A.	29
Figure 18. Showing the overall time chart for course B.	29
Figure 19. Chart to show the number of attempts per hour each hour before a due date. Assignment 5 course A.	30
Figure 20. Time breakdown for quizzes in course A.	31
Figure 21. Time breakdown for assignments in course A.....	31
Figure 22. Time breakdown for quizzes in course B.	32
Figure 23. Time breakdown for assignments in course.	32
Figure 24. Average number of attempts vs. correct rate for all question parts in course A.	33
Figure 25. Average number of attempts vs. correct rate for all question parts in course B.	33
Figure 26. Average number of attempts vs. overall grade of students in course A.	34
Figure 27. Average number of attempts vs. overall grade of students in course A, no outliers.	35
Figure 28 Residual graph of average number of attempts vs. overall grade of students.....	35
Figure 29. Average number of days started a test before due date vs. overall grade of student in course A.	36
Figure 30. Residual graph of average number of days started a test before due date vs. overall grade of students. Course A.	37
Figure 31. Line of best fit for average number of days started a test before due date vs. overall grade of student in course A.	37
Figure 32. Scatter plot of “Missing Negative” for all Question Parts in Course A.	38
Figure 33. Scatter plot of “Missing Negative” for all Question Parts in Course B.	39
Figure 34. Scatter plot of “Wrong Number” for all Question Parts for course A.	39
Figure 35. Scatter plot of “Wrong Number” for all Question Parts for course B.....	40
Figure 36. Scatter plot of “Within 5%” for all Question Parts for course A.	41
Figure 37. Scatter plot of “Within 5%” for all Question Parts for course B.	41
Figure 38. Scatter plot of “Wrong Units” for all Question Parts for course A.	42
Figure 39. Scatter plot of “Wrong Units” for all Question Parts for course B.	42

Figure 40. A diagram of the flow of data from the report generation to the dashboard. In red are server side pages. Orange are client side pages built personally for this dashboard, and in blue are open sourced libraries. Green is the dashboard. 56

List of Tables

Table 1. Table of all raw values for each compartment for course A and course B.	22
Table 2. A table of t values for the different compartments for course A and course B.....	23
Table 3. A table showing the percentage difference between course A and course B for the different compartments.....	24
Table 4 For all assignments in course A, showing the numbers before due date that majority of attempts were made.	30
Table 5 Table of the most popular answers overall, correct and incorrect for both course A and B.....	43
Table 6. A table of the most common answers from question part 6303 from Assignment 6, blocking by the correct answer.	45

Background

Three years ago, another Honours student created the Auto-Ed website (<http://autoedu.ok.ubc.ca/login.php>) for first year physics students. This website was built to allow students to complete their physics assignments online. TAs and the professors upload the assignments and the students are then required to go online to complete them. Over the past three years, Physics 112 and 122 have been taught in such a way that all of their assignments are posted online to this website. While systems such as this have been created before (Coursera <https://www.coursera.org/> and others), Auto-Ed is distinct as it allows for questions to be asked whose answers are not just multiple choice. Students are required to give answers in natural language by typing in the answer themselves (4.0 V, -3.94 N, etc...). Their answer is then parsed and evaluated and the student is informed whether or not they are correct, by either a checkmark or an “x” appearing beside their answer. Another interesting feature of Auto-Ed is that each student is shown a slightly different question than his or her peers. Each question comes with numerical values, and these values are randomly generated between a set of possible values. This way student A will have to complete the question using the exact same reasoning and formulae, but will get a different answer than student B. Students are allowed to attempt the question as many times as they need to before either they get it correct, or give up.

Objective of this Thesis

With a data set that was already quite large after 3 years, it was found that two key items were missing from this system. First, professors were not given feedback about how their students were performing, or how the assignments and questions were faring. Therefore a professor dashboard had to be created which would give this information to professors for their use. Second, no in-depth analysis of the data had ever been attempted to understand how students are interacting with the system and the quality of the questions asked. Therefore the objectives were to create a professor use dashboard and to conduct an in-depth analysis.

Data Collection

Before this Thesis, the system was only collecting a student’s correct answer. This meant that while a student’s grade could be computed, no real analysis could happen, since the student’s incorrect answers were not known. Therefore beginning in September of 2012 (first month of work on this thesis), all data was being collected, including a student’s incorrect answers. By the end of the school year (April

2013), over 450,000 data points had been collected. The data that was collected was the student's id, the id of the assignment, and test, the timestamp of the attempt, the correct answer, the student's answer, and the number of marks the student received.

Terms and Definitions

This section clarifies specific terms used in the thesis.

Program

All of the data analysis and work was done by writing computer code. The code was written predominately in Java Server Pages (JSP), JavaScript, and SQL. All of the code is found online at cs-suse-4.ok.ubc.ca and the data is hosted on a Microsoft SQL Server database at cssql.ok.ubc.ca. The word program shall be used in this document to represent the entire collection of code and data that was written to achieve the goals of this Honour's Thesis, unless otherwise noted.

Test/Assignment

TAs and professors are allowed to upload a new test or assignment. The two terms shall be used interchangeably henceforth. An assignment is made up of a list of questions. An assignment is given a specific template id, which is unique for each assignment.

Question (Q)

A question is also given a template id and is made up of text and each Question contains Question Parts.

Question Part (QP)

A question part belongs to a certain question, and represents a part or sub-question of a question. In terms of a normal paper assignment a question gives the required information, but does not actually require an answer, and a question part asks a very specific question based on the information given in the question. A question part requires an answer and so students are generally responding to question parts. Therefore, this thesis will mostly talk about question parts.

Missing Negative

A student answer could be incorrect because they were missing a negative. For example if the correct answer was -9.8 m/s^2 then an answer that was incorrect due to a missing negative would be of the form 9.8 m/s^2 .

Wrong Number Value

A student answer could be incorrect because the actual value of the number they were inputting was different than the actual value of the correct answer. For example if the correct answer

was 2.0 N, and the student wrote 1.0 N, then they would be incorrect because of a Wrong Number Value.

Within 5%

A student answer could be incorrect because of a Wrong Number Value, but the answer could be within 5% which would show that they were fairly close to the correct answer.

Wrong Units

A student answer could be incorrect because the student was incorrect about the units. This was a fairly common error as many students were making simple syntax errors, such as not capitalizing units, or making mistakes such as (m/s[2] instead of m/s²).

No Answer

Occasionally students would submit an answer of nothing. This was a fairly common occurrence especially at the beginning of the courses, as students were figuring out the system.

Methodology

This section describes how the work was accomplished throughout the past year, and how the results were obtained. In particular it will focus on the main challenges encountered. For a full description of the software written, the reader is directed to the Appendix where the Technical Summary resides.

Parsing Student Answers

One of the largest challenges in analyzing the data was parsing the student answers. Answers came in a wide variety of inputs (anything from 1.0 N to -.432 m/s²) and the program had to be able to handle virtually any type of input. Each answer was broken up into separate pieces based on the differences between numbers and characters. Therefore an answer such as 1.0 N would be broken up into [1][.][0][][N] and a number such as -.432 m/s² would be broken up into [-][.][432][][m/s²]. An identical procedure was performed on the correct answer. Once both the correct answer and student answer were broken up, the separate pieces were rebuilt into the “number value” and the “units”. Then the “number value” and the “units” of the student answer were compared to the correct answer. Answers were compartmentalized into either correct or incorrect. Incorrect answers were further compartmentalized into Missing Negative, Wrong Number Value, Within 5%, Wrong Units, and No Answer.

Initial Results

Once the student answers were fully compartmentalized, the program was amended to include a feature that would take a particular question part, find all of its answers and analyze them by using the above method. This data was then aggregated into a chart to show the statistics for that particular question part. The charts were created using the Google Charts API. Once this was created, work was done to create a website where a user could go and find all of the assignments, all of their questions and all of their question parts. For each question part, charts were shown about how that question part was being done by students.

In addition to the charts for each question part, data was aggregated for each assignment and for each course section that a user might be interested in. Charts and statistics were also created for each student in the system. These initial charts and statistics were used in some part to create the in-depth analysis that is described below.

Equations

In addition to compartmentalizing incorrect answers into Wrong Number, Missing Negative, Wrong Units, Within 5% and No Answer, it was attempted to determine what equation was being used to generate the number that the student gave. Therefore, it can be considered that this work was attempting to further compartmentalize each Wrong Number answer into further sub-divisions.

Due to all questions having a random number generator creating the values, each student will be answering a slightly different question. Therefore the correct answer for each student will be different even though they are answering the same question. It was determined that it would be beneficial to be able to determine how a student arrived at their incorrect answer. For example, was it just a simple careless mistake (maybe they forget to add an extra term, or just did the multiplication wrong,) or something more complicated (such as a lack of knowledge about the theory behind the problem). However, this is not an easy problem to solve. What was proposed was to take the correct equation, which gives the correct answer for that random value of the question part, and perturb it in some systematic way: remove terms, add extra terms, remove multiplication and change to addition or many other combinations. For example taking an equation of the form (an actual equation from the system):

$$v1 = \sqrt{(2 * m1 * 9.8 * \frac{h}{m1 + m2} - 9.8 * h}$$

It would be possible to perturb the equation by removing variable terms (m_1 , m_2 , h), adding additional terms (other variables given in the question, but not actually used to calculate the correct answer, might be used by students mistakenly), inverting the negative and positive signs, changing multiplication, addition, subtraction and division operators and even completely removing the square root operator. For each of these new versions of the equation, the correct answer would be calculated from the new equation (using a special parser and parse tree) and tested against the student answer. If any of these new answers match the student answer then we can assume that the equation that was used to derive that answer may have been the equation the student used. It is of course not necessarily true that is the equation they used, but it might have been. Then the goal was to find any patterns in the equations that were being used for the same question part. If a particular equation was used by a majority of the students, then it might represent an error in the question part's answer, or it might represent a topic that was not properly understood by students.

Unfortunately, as work progressed on this part of the project it was discovered that it was more difficult than originally thought. The reason for this difficulty was due to actually parsing the newly created equations to test whether or not they equaled the student's answer. In addition, due to time constraints, it was left to the end, and unfortunately work was never resumed on this section.

Implementation

Professor Dashboard

The dashboard was created to be used by the professors of 112/122 Physics to convey to them statistical information about how their students and assignments are faring in almost real time.

Report Generation

Due to the size of the data-set (over 450,000 data points), it is virtually impossible to convey useful information to a user in real time. The amount of time it takes to analyze the data points would be too much, and it would take over 10 seconds to load just one chart, which of course would make the whole process un-usable. Therefore it was decided that instead of showing live data, report data would be shown. The reports would be created at midnight every night and would show data for the previous day. This way when a professor logs into the system in the morning, they would know what had happened yesterday.

There are a total of 4 reports that are run daily: student information report, courses report, assignment report and daily report.

Student Information Report

This report runs by finding all of the students for each of the active courses and for each student it finds all of the attempts they have made. For each attempt it analyzes and compartmentalizes them into being either correct or incorrect and then further breaks down the incorrect responses into the categories discussed above. This report also finds the overall grade of the users. Below are two interesting patterns of student behaviour and a statistical way of describing students in aggregate form.

Guesser

The reader should recall that the questions being asked of users have randomized values. What this means is that while the question is the exact same, the values in the question (the height of the building, the speed of the cat, etc...) are randomized for each student. Therefore each student answers a different question. However, it also means that a student can figure out the range of possible answers. Some students used this to simply guess their way to the question. It was observed that students would enter in a series of answers such as this:

$0V, 1V, 2V, \dots, 48V, 49V, 50V$

Getting the correct answer in this way is clearly not what was intended by the creators of the system or by their physics professors. Therefore a method was devised to use some simple pattern matching to find a pattern such as this and to warn the professors that a student was doing this. No mention of this has been made to the students, but it is thought that in the future, this system could be implemented in real time for student submissions, and it would alert them that what they are doing is wrong and should cease immediately.

Button Masher

A button masher is different from a guesser in that they simply enter the exact same answer over and over again. For example if the correct answer was $48.5 V$ they would enter $34.1 m/s$ or something else that is incorrect. However, they apparently feel that their answer is correct even if the first time they entered it, they were told it was incorrect. Therefore they just keep pushing the submit button over and over again until they realize that in fact they are wrong. Then they try a different answer, let's say $76.4 V$ and again get it wrong, but feel that they are correct and keep pushing the submit button over and over again. This was fairly odd behaviour until conversations were held with some students who expressed frustration over how sometimes their answer would be correct in value,

but incorrect in units only because they did not know how to enter in the units properly. For example the correct answer might be $12.3 E5$ which written in standard scientific notation would be 12.3×10^5 . However, it is generally understood that whenever typing in equations into a computer, the capital letter E is used to denote a power of ten. Though it seems that this was not expressed well enough to the students (or they were just not paying attention) and so many students will enter their answer in the form 12.3×10^5 and even though their answer is technically correct, the computer will return an incorrect response, which leads to frustration by the students who decide to “button mash”. This shows how important it is to inform students about how to properly enter in their answers.

Statistical Ranking of Students

Based on the above data (average number of attempts, wrong number attempts, wrong units, missing negatives, no answer, grade, and their likelihood of being a “guesser” or “button masher”), a statistical ranking is created for each student. The ranking is based on the performance of all of the other students and places students inside one standard deviation from the norm as “normal”, below a standard deviation as “below average” and above a standard deviation as “good”.

Courses Report

The courses report finds all of the average information overall for the all of the assignments of this course. It finds the average total attempts, the average wrong number attempts, etc... for all of the question parts for each assignment. It then sums up all of the data for each question part for each assignment. Therefore for each assignment there will be one statistic for total number of attempts, total wrong number attempts, total wrong unit attempts, etc... Then this report generation calculates the standard deviation for all of the assignments in total for this course and sees if any of the assignments are above the first standard deviation. If they are then an alert is created for that assignment. This report also finds and presents the number of users that attempted each assignment.

Assignment Report

The assignment report finds all of the detailed information about each assignment for this course. For each assignment, it finds all of the attempts made for it, and analyzes and compartmentalizes each of those attempts. Then a chart is created from that data to show how the students have been doing on this assignment. Data is aggregated for each question part of each assignment, and a breakdown is shown of all of the question parts, which will show any major outliers that need to be addressed by the professors. Alerts are created on this basis. In addition to this, a time chart is created based on the time that each attempt was made. This is calculated by summing the total

number of attempts made per hour before the due date. The chart shows when the most activity happened for this assignment.

Daily Report

The daily report is quite simple. It finds all of the attempts made in the past day, analyzes them, and creates alerts based on what question parts were doing badly relative to the rest.

Live Data

An interesting feature that was built early in the project was a live data chart. Using the HighCharts JavaScript library, this chart queries the database every second for any new attempts being made. If a new attempt was made since the last second, it retrieves it, analyzes and compartmentalizes it, and plugs it into the chart. The chart updates itself every second, and the data points (during a high frequency time) will be moving off the left of the chart every second. It is quite interesting to watch students keep making their attempts and being able to see each one. A professor is able to click on each individual attempt (data point) and be taken to a page that has a wealth of information. For example, it shows all of the student's attempts that they have made for this assignment. It also shows a chart breakdown of the student's overall answers throughout the course, and shows a chart breakdown of the assignment as well, to compare against.

Dashboard Walkthrough

This section provides a walkthrough of what the dashboard looks like and what a professor would see when using the system.

Main Page

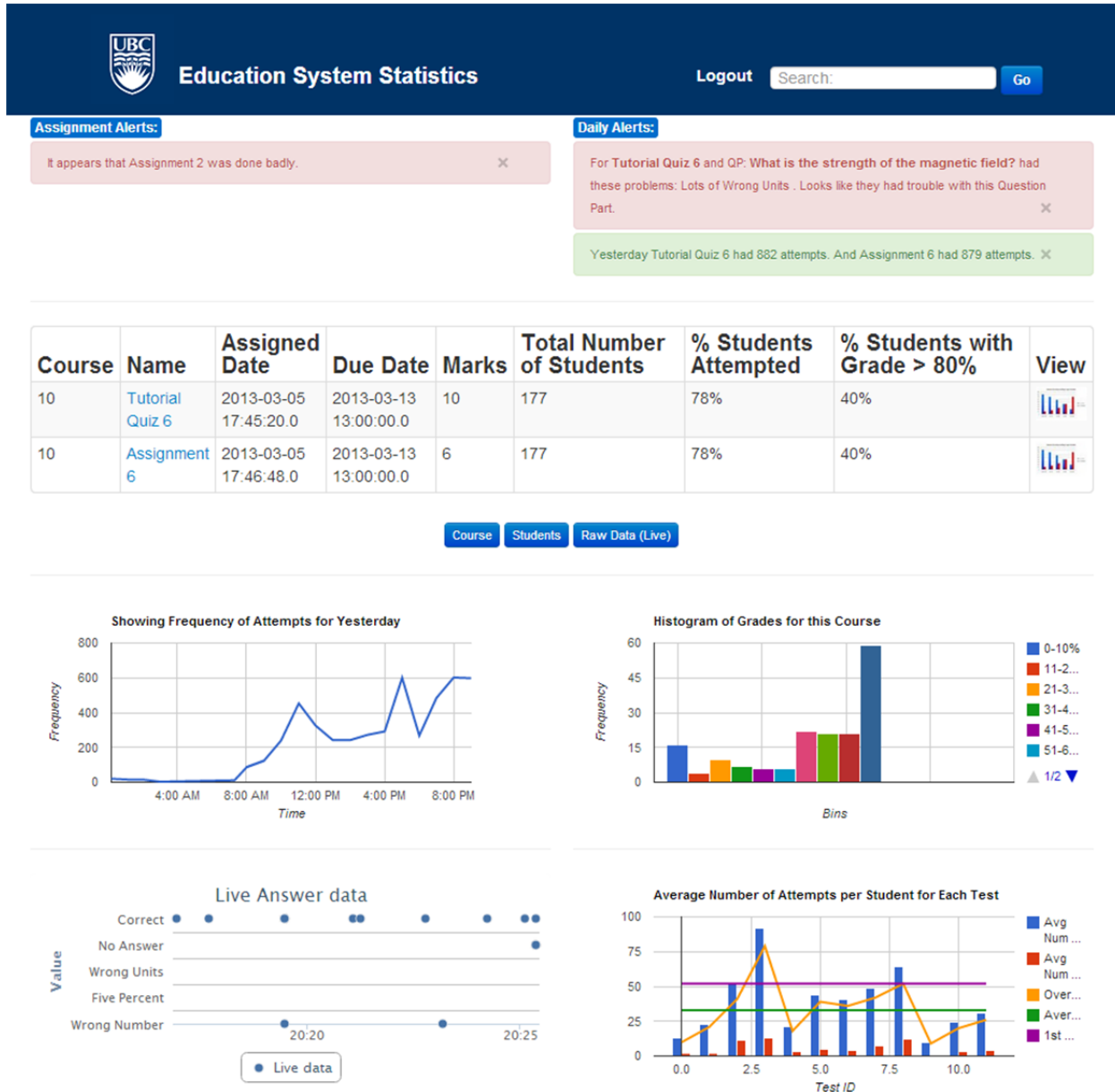


Figure 1. A screenshot of the main page of the dashboard.

The alerts are displayed at the top of the page just below the header. Inside the header is a search bar that allows a professor to search for students by name or student number. Typing in a character will show a dropdown of all students who fit that character either in their name or their student number. The table at the top shows all of the currently active assignments, the number of

students who have attempted the assignment (a grade over 0%) and students who have already gotten a score over 80%. A professor is also able to view the assignment in more detail. Below that is a time chart showing the frequency of attempts for the previous day. Next to that is a histogram of all the students grades overall in the course. In the bottom left is the live data chart, and to its right is a breakdown of how the assignments have been done overall relative to each other. If a user clicks on the course button just below the table, they will be taken to the following page.

Course Page



Figure 2. A screenshot of the top half of the “course” page of the dashboard.

The course page shown above is approximately half of what a professor would see. See Figure 3 for the rest of the page. At the top of this page, the professor is shown a time chart for all of the

attempts made for their course section. The red lines represent the due dates. This way a professor is able to perceive when students are making the majority of their attempts. Next is a breakdown of all of the assignments in this course section. The bars represent a summation of all of the question parts for each assignment. Outliers are represented by being outside of the first standard deviation line. Lastly the reader will see a chart showing the total number of students who attempted each assignment.

Course Page Continued

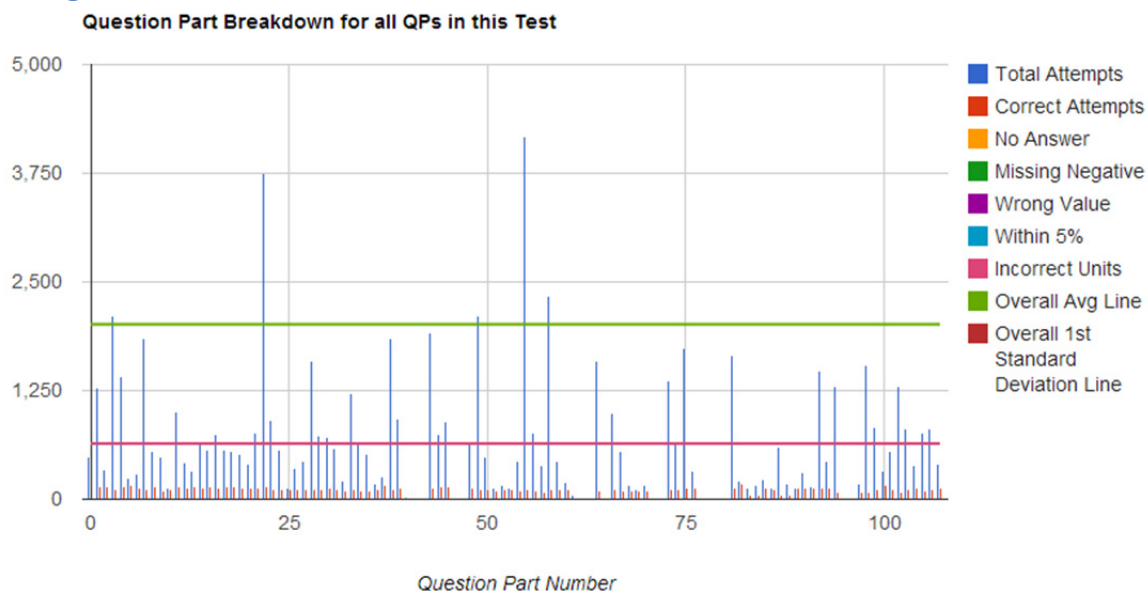


Figure 3. Bottom half of the course page.

The rest of the course page is made up of this chart which shows an overall breakdown of how each question part in the course is being done. Again badly done question parts are found by finding data points that lie outside of the standard deviation lines. This is because if there are more attempts being made for a particular question part then by definition (knowing that there is always approximately the same number of students) there are more incorrect attempts being made. This chart also shows a breakdown of all of the different statistics for a question part (the different compartments that answers can be thrown into). Professors are encouraged to use statistics such as this to find question parts that have serious problems and attempt to fix them. While it might be claimed that some questions are just more difficult than others (very true), it is an unfortunate consequence of having an online system that students in many cases simply enter their correct answers incorrectly into the system as discussed above in the button masher section. Therefore some explanation about how the computer expects the answers would greatly reduce frustration and annoyance on the part of the students, but still leave the question at the same level of difficulty.

Assignment Page

Back

Pick a course: Choose Pick an assignment: Choose

This report was generated at: 2013-03-16 21:21:20.067 making it accurate up to that point in time. This is for

Tutorial Quiz 6

There have been a total of 142 students who have attempted this assignment.

Question Part 1 with name Calculate the resistance of the copper wire loop (You may use ohms in place of Ω). seems to be very badly done. Maybe take a look at it?

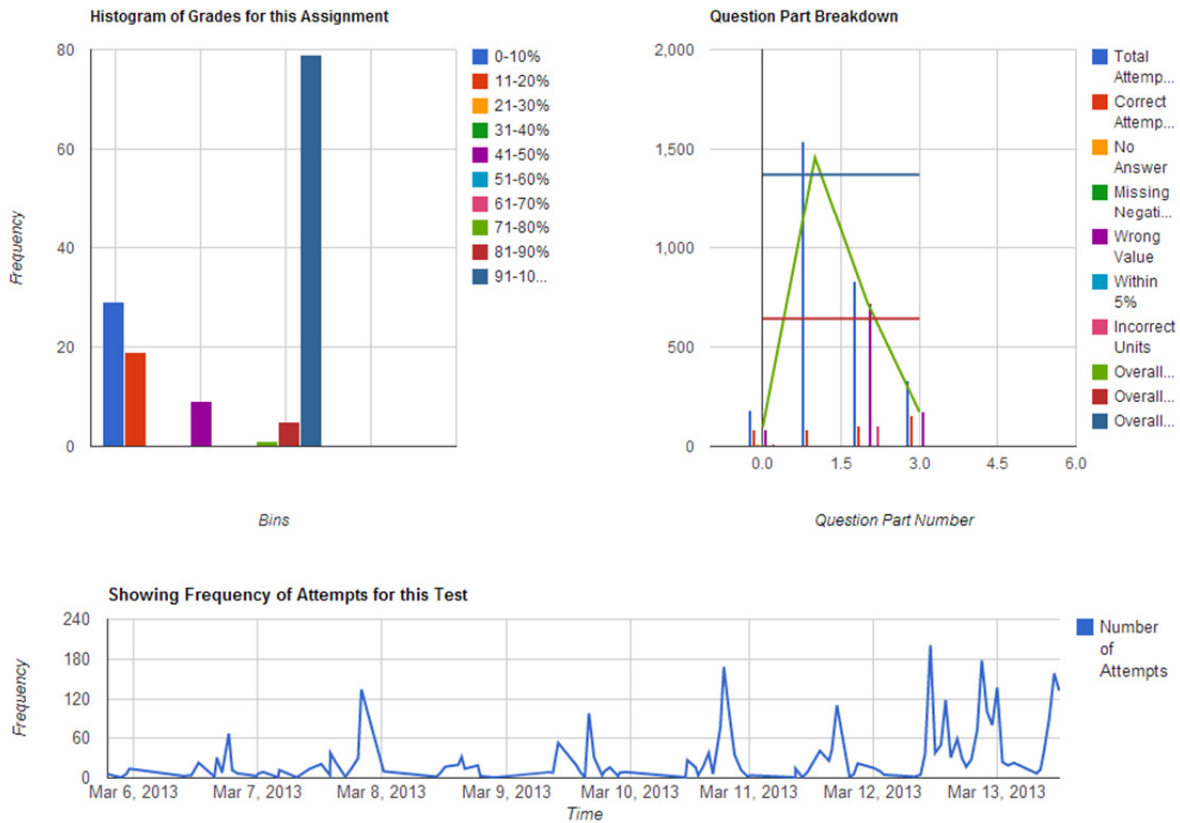
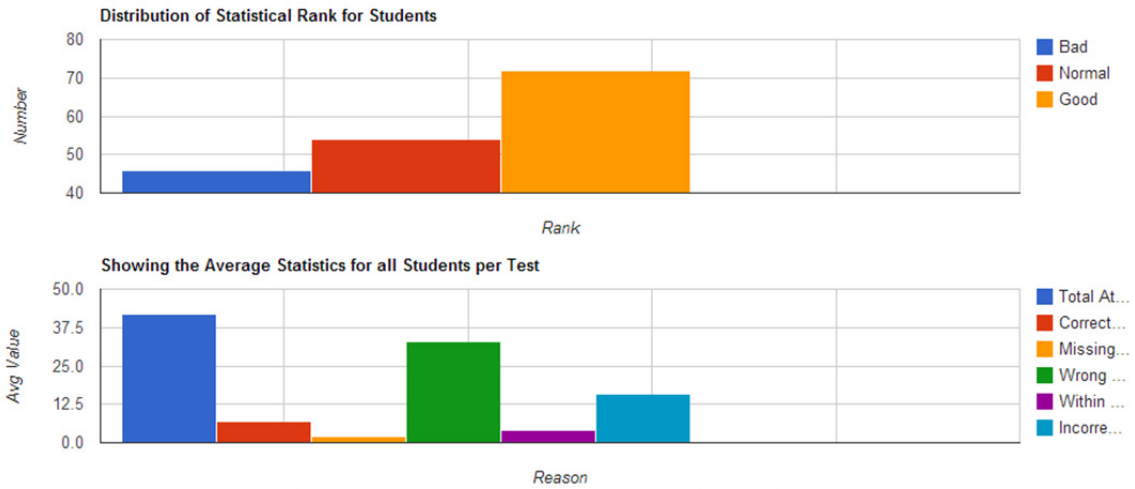


Figure 4. Screenshot of the “assignment” page on the dashboard.

The assignment page is accessed by clicking on the chart icon on the front page for current assignments. From there a professor is able to pick and view any other assignment in the system. For any assignment, a number of alerts can be shown about any trouble question parts. Also a chart breaking down the question parts is shown beside a histogram of students’ grades for just this assignment. Lastly a time chart showing the frequency of attempts is shown.

Students Page

[Back](#)



Student Name	Student ID	Overall Average Grade (all Tests) %	View	Statistical Ranking	Status
[Redacted]	[Redacted]	99.31	View	Good	Caught Up
[Redacted]	[Redacted]	5.56	View	Below Average	Behind by 10 Tests
[Redacted]	[Redacted]	88.14	View	Good	Caught Up
[Redacted]	[Redacted]	94.38	View	Good	Caught Up
[Redacted]	[Redacted]	82.87	View	Good	Caught Up
[Redacted]	[Redacted]	62.02	View	Normal	Behind by 2 Tests
[Redacted]	[Redacted]	93.87	View	Good	Caught Up
[Redacted]	[Redacted]	93.39	View	Good	Caught Up
[Redacted]	[Redacted]	81.81	View	Normal	Caught Up
[Redacted]	[Redacted]	96.67	View	Good	Caught Up
[Redacted]	[Redacted]	26.89	View	Below Average	Behind by 2 Tests
[Redacted]	[Redacted]	15.61	View	Below Average	Behind by 6 Tests

Figure 5. Screenshot of the students page on the dashboard.

The student page is accessed by clicking on the Students button on the main page. It shows a number of different results about how students are faring overall in this course section. At the top the reader can see a chart showing the distribution of student rankings based on the statistical ranking discussed above. Next is a chart showing the breakdown of all students for all assignments, showing the average total number of attempts, wrong number attempts, etc.... Then there is the table which shows all students enrolled in the course (clearly redacted for the purposes of this Thesis write up) along with their overall course grade and their status based on how far back they are in terms of assignments.

Individual Student (Below Average)

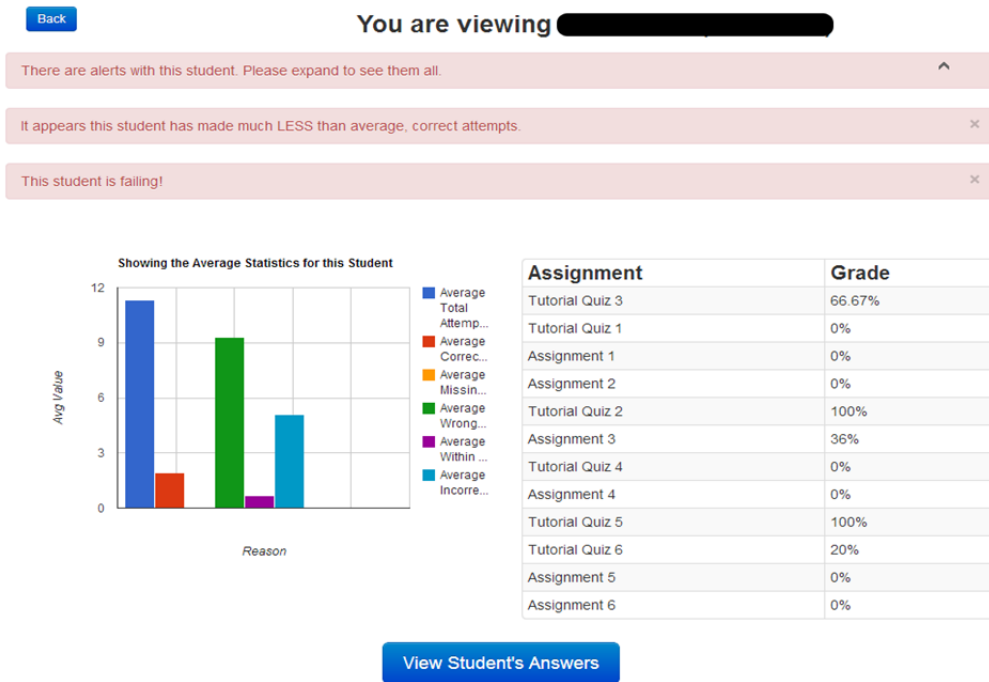


Figure 6. A “Below Average” student view.

The above picture shows what a professor would see if they click to view a student that is below average. There are alerts for this student because they are failing and are not making enough correct attempts. Also there is a breakdown of what this student has done overall in terms of total attempts, wrong number attempts, etc... and a list of all of their grades for all of the assignments. In addition a professor is able to view all of the student’s answers in a table format by clicking the bottom button. This can be used when a student claims that an answer they made was correct, but it was not recorded correctly.

Individual Student (Good, but “Guesser”)

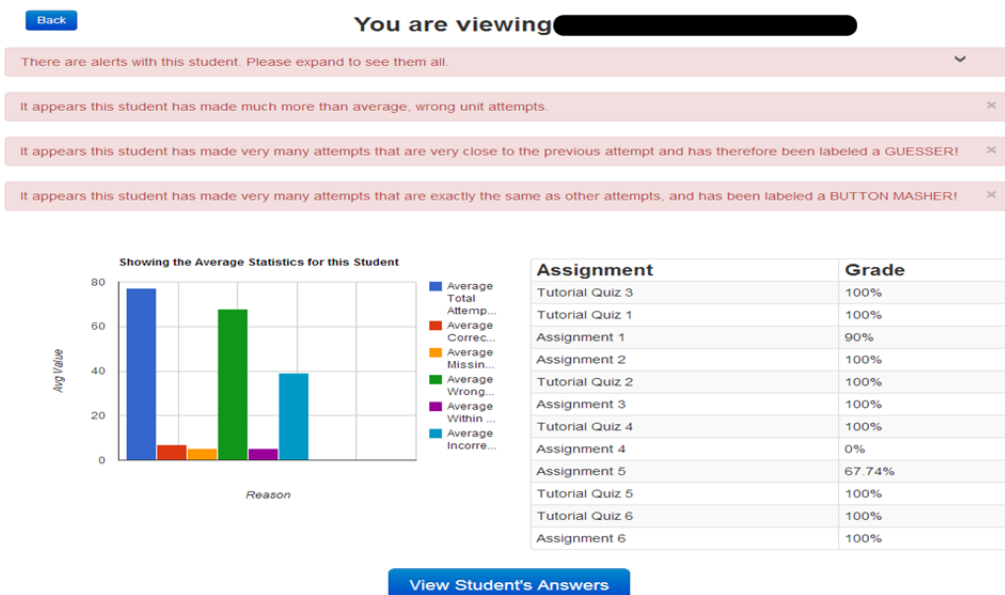


Figure 7. Screenshot of a “Good” student, but with many alerts.

The picture above shows a student who has been statistically ranked as good (clearly with good overall grades), but they are also a guesser and a button masher, therefore their grades overall might be somewhat suspect, and their midterm and final grades might not be as high as their assignment grades.

Individual Student (Good)

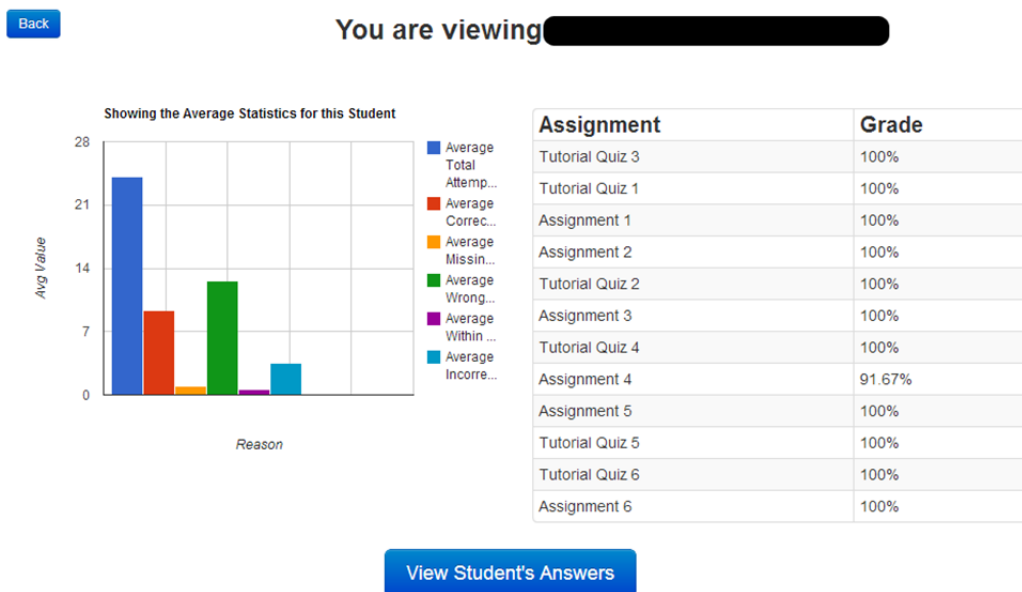


Figure 8. Screenshot of a “Good” student.

Figure 8 shows the results of a student who is labeled “Good” and does not have any alerts. This is a perfect student.

Results and In Depth Analysis

With the full implementation and development of the dashboard it was found that a vast amount of data existed that just could not be analyzed properly by a professor who only has a few minutes a day to spend on the website. Therefore an in-depth analysis was undertaken that would provide detailed information. The two course sections from the fall semester of 2012 were used. Both sections were teaching the same Physics 112 course, however, they were taught by two different professors. In addition, the sections both used the exact same assignments, questions and question parts. This allows for a comparison to be made between the two courses.

The in-depth analysis compared the two course sections (course A and course B) against each other to see if there are any real differences. It also allows a professor to see which assignments and question parts are not being done well overall with the goal of changing them for the next year of

teaching. It should be noted that all charts and figures presented below were generated by web pages that were created for this specific purpose.

Course Differences

To determine if there are any statistically significant differences between the two course sections, first a comparison was made based on the different compartments that incorrect answers can fall into. Statistical sampling was used on each of the different compartments. Below is an example calculation for the compartment Missing Negatives.

t-tests

The goal is to find if course A has more missing negatives on average than course B. For each compartment, it is necessary to compute the following proportion:

$$\frac{\left(\frac{\text{missingNegatives}}{\text{numStudents}}\right)}{\left(\frac{\text{totalAttempts}}{\text{numStudents}}\right)} = P_a$$

The raw numbers are below:

Table 1. Table of all raw values for each compartment for course A and course B.

Category	Course A	Course B
Number of Students	182	207
Total Number of Attempts	52900	57212
Number of Missing Negative Attempts	4243	4563
Number of Wrong Number Attempts	44226	46206
Number of Wrong Unit Attempts	11002	13321
Number of Attempts Within 5%	8495	10694
Number of No Answer Attempts	1543	1963

Next for the Missing Negative compartment, P_1 and P_2 are both calculated:

$$P_1 = \text{course A}$$

$$P_1 = \frac{\frac{4243}{182}}{\frac{52900}{182}}$$

$$P_1 = \frac{23.3132}{290.6593}$$

$$P_1 = 0.0802$$

$$P_2 = \text{course B}$$

$$P_2 = \frac{\frac{4563}{207}}{\frac{57212}{207}}$$

$$P_2 = \frac{22.0434}{276.386}$$

$$P_2 = 0.0798$$

Next find the difference of the variance using:

$$v(P_1) - v(P_2) = P_1(1 - P_1) + P_2(1 - P_2) + 2(P_1P_2)$$

$$v(P_1) - v(P_2) = 0.159999$$

Next the square root of this will be the standard error:

$$\text{standard error}(se) = \sqrt{0.159999}$$

$$se = 0.39999$$

Next do the t test:

$$t = \frac{\beta_1 - \beta_2}{se}$$

$$t = \frac{\frac{4243}{182} - \frac{4563}{207}}{0.39999}$$

$$t = \frac{23.313 - 22.0434}{0.39999}$$

$$t = 3.175$$

Looking up the table of t-values we see that for a very large number of degrees of freedom, if a t value is greater than 1.96 for an $\alpha = 0.05$ it is statistically significant at a 95% confidence interval.

Table 2. A table of t values for the different compartments for course A and course B.

Compartment	t-value	t-value threshold	Outcome
Missing Negative	3.175	1.96	Course A bigger
Wrong Number	15.43421	1.96	Course A bigger
Five Percent	5.88134	1.96	Course B bigger
Wrong Units	8.46655	1.96	Course B bigger
No Answer	3.98998	1.96	Course B bigger

This means that there is a statistically significant difference between the two courses for all of the different attributes. Doing the same calculations as above but sampling from the total sample of 110112 points and taking the total number of attempts per average for each course we see that again there is a statistically significant difference.

$$t = 14.2838 > 1.96 (\alpha = 0.05) \text{ (course A is bigger)}$$

It is also interesting to note the percentage difference between the courses for each of the compartments.

Table 3. A table showing the percentage difference between course A and course B for the different compartments.

Compartment	Course A	Course B	Estimated Difference
Total Attempts	+	-	5%
Missing Negative	+	-	4%
Wrong Number/Value	+	-	9%
Within 5%	-	+	10%
Wrong Units	-	+	5%

Interestingly it seems that course A has more attempts overall on average and more missing negatives and more wrong number attempts, but has less within five percent, wrong unit and no answer attempts. It is difficult to make conclusions about this, but it might be hypothesized that the professor from course B gave more focus to getting the right value, while the professor from course A gave more directions about how to do the online assignments and so students knew how to enter in the units better.

Question Part Breakdown

In continuing with the comparison between the two course sections, the charts below (Figure 9 and Figure 10) show the overall distribution of all question parts for both course sections. The data points represent the total number of attempts, or the total number of correct attempts, etc... that were made on this question part. The higher the number for an individual question part, the more poorly that question part was done. This is because more attempts by definition mean more incorrect attempts as there can only be a certain number of correct attempts. First shown is the chart for course A and it is apparent that while there are a few outliers, the largest outlier is number 61 which is from assignment 6

and is this question: “What is the magnitude of the gravitational force acting on the planet due to the sun? Use scientific notation with three significant figures.”

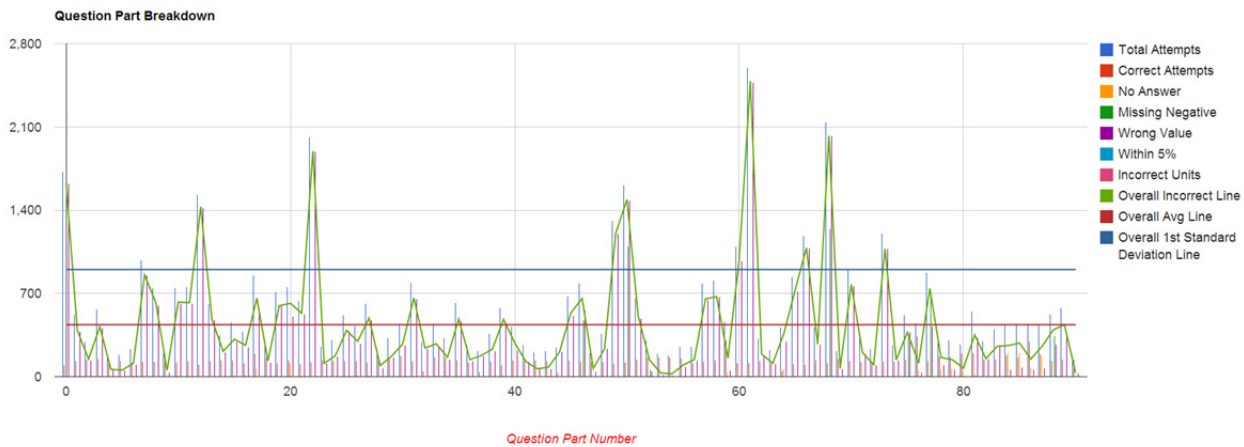


Figure 9. Chart showing the question part breakdown for course A.

Next the chart for course B is shown, and it is apparent that the outliers are the same as in the previous chart, especially the largest outlier of number 61 from assignment 6. The reader shall see below that all of the subsequent charts, figures and data show that that question from assignment 6 was by far the worst done overall and should be changed.

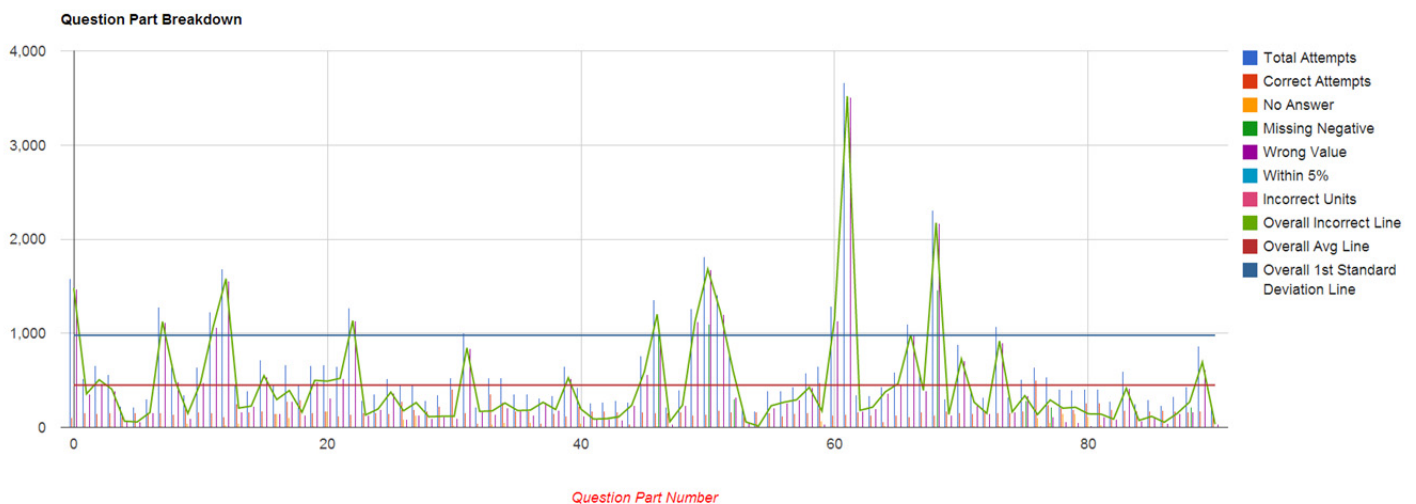


Figure 10. Chart showing the question part breakdown for course B.

What is apparent from Figure 9 and Figure 10 is that while the two course sections were taught by different professors, students from both sections had trouble with the same question parts. This means that irrespective of the professor, students will encounter difficulty with the same concepts.

Histogram of Compartments

In addition to a simple breakdown of all of the question parts for a course section, it is possible to view a histogram of what percentage of incorrect answers fall into a specific compartment. The two charts have been presented below in the same order as before showing course A first, and course B second.

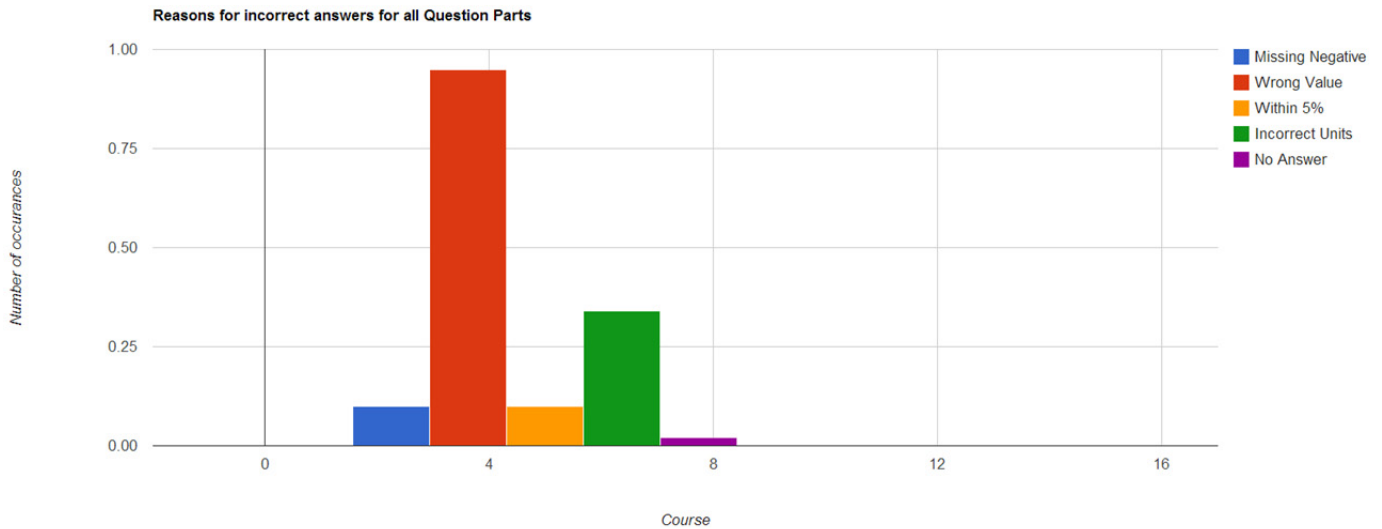


Figure 11. Showing the histogram of what compartment student answers fall into for course A.

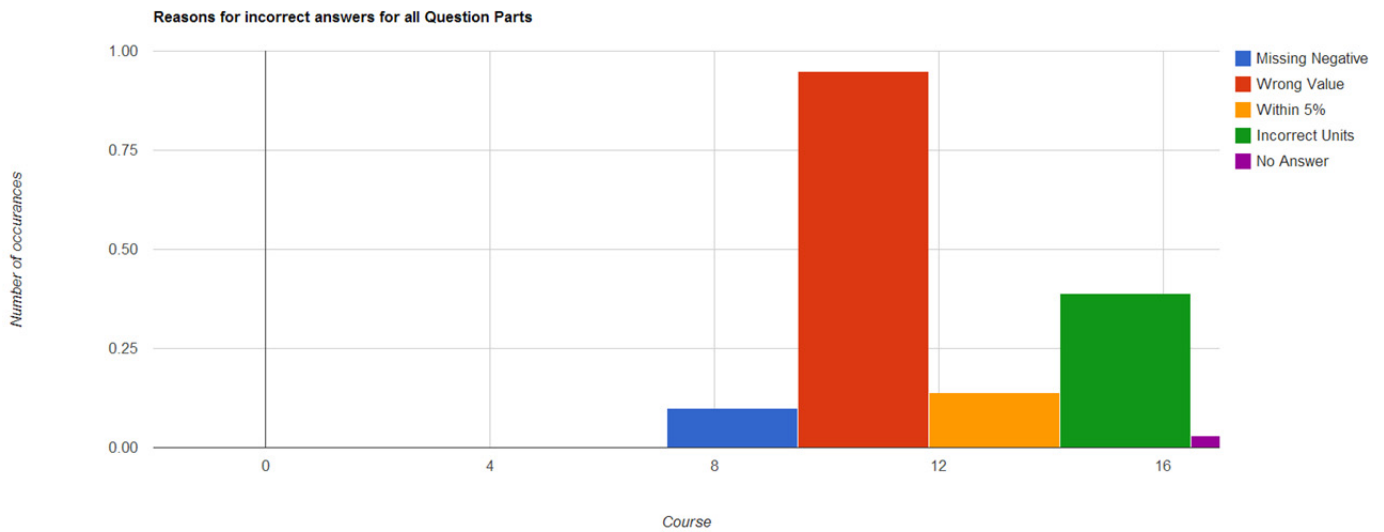


Figure 12. Showing the histogram of what compartment student answers fall into for course B.

The two courses do not have much difference. For both courses the ratio of Wrong Number answers compared to all answers is hovering around the 0.92 line, which means that 92% of all answers were incorrect because of having the wrong number. The other compartments also match for the two course sections, which again goes to show that the difference of professors is very slight.

Assignment Breakdown

Next is shown the breakdown of the two course sections as based on the assignments. For these charts all of the question parts of each assignment have been aggregated to create a total value for each assignment. These sum values are used to compare the assignments against each other. An assignment is considered to have been done poorly if it has more average number of attempts relative to the other assignments.

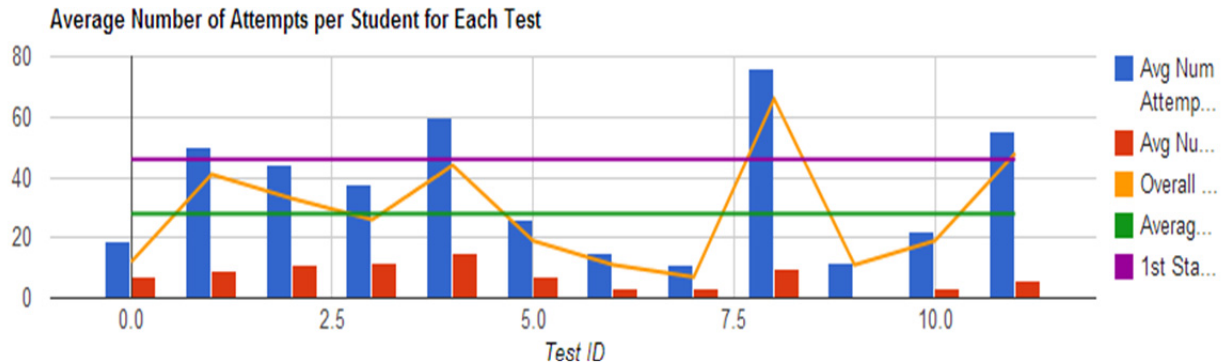


Figure 13. Showing the assignment breakdown for course A.

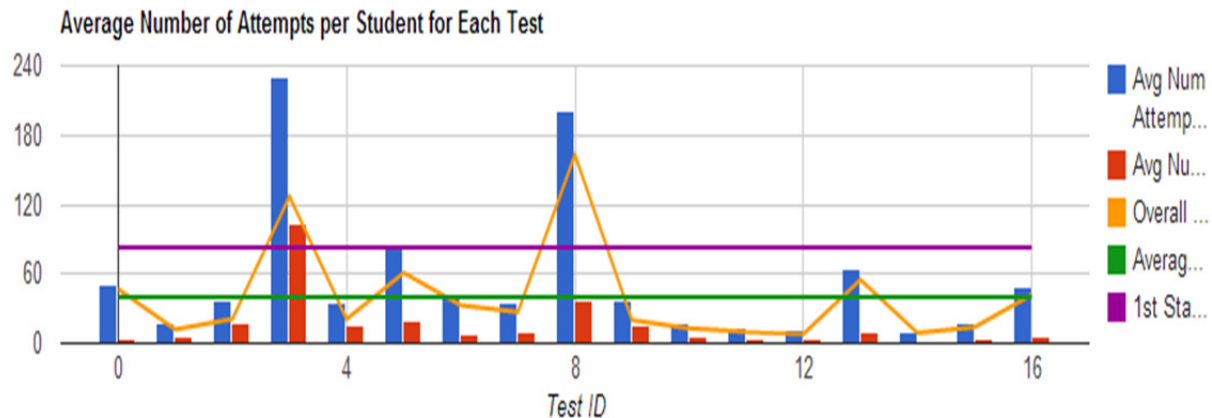


Figure 14. Showing the assignment breakdown for course B.

Figure 13 and Figure 14 are on a different x axis scale because there were more students in course B than in course A. They also at first seem to show a large difference. However, deeper consideration is required to properly understand what has happened. When a TA or professor uploads an assignment, the database keeps track of it; however, when an edit needs to be made (correct a typo, or change a question) a brand new assignment has to be created. There is no edit assignment feature. Therefore a new assignment is uploaded, however, the old assignment is still in the database, and in many cases has already been released to the students who have started to answer its questions. But when the new assignment comes out, it supersedes the old one, so that students will have to redo their

attempts, however it will still have recorded the old attempts on the old assignment, which shows up in the above two charts. It seems that course B has had this problem more than course A, and is therefore reflected in the above chart. Course B had 16 assignments altogether, while course A only has 11. Once the faulty assignments are thrown out, it is seen that for course A, assignment 5 was done the worst overall and for course B it is seen that assignment 2 and pre-tutorial quiz 4 were done the worst. A difference between the two courses that is somewhat surprising. This difference is one of the only differences between the two course sections, and further analysis needs to be done to understand this difference.

Number of Students

Below are presented two charts that compare the number of students who have attempted each of the assignments.

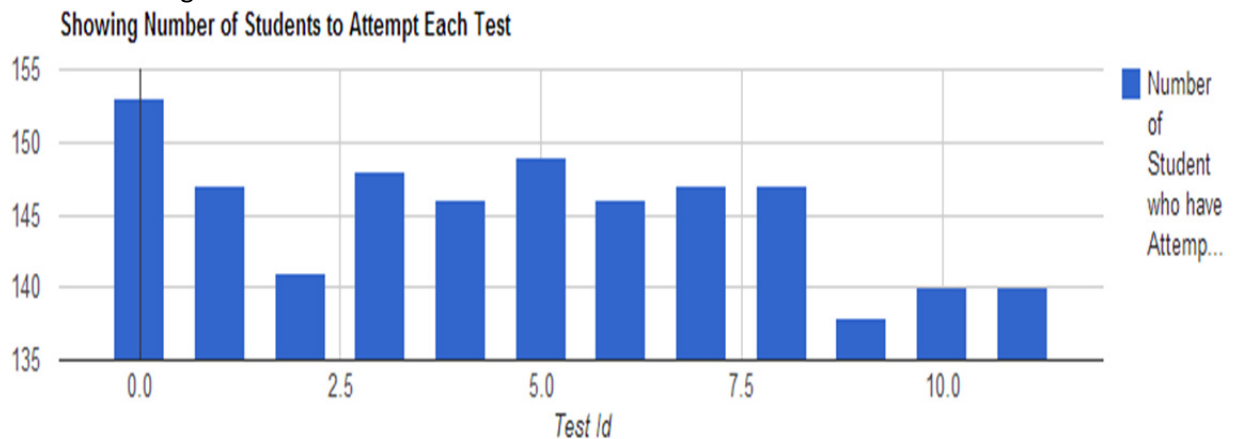


Figure 15. Number of students who have attempted each assignment for course A.

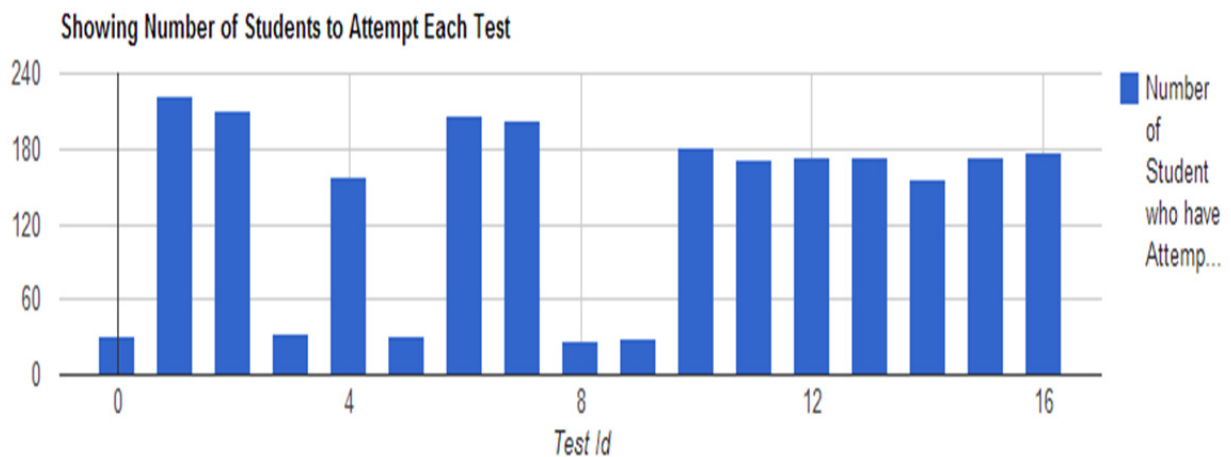


Figure 16. Number of students who have attempted each assignment for course B.

Comparing the two charts, again it is seen that at first glance there seems to be a large difference. But once again we are presented with the fact that course B had faulty assignments which were generally attempted by very few students and so can be ignored. Once those assignments are discarded then approximately the same number of students are attempting the assignments as the year goes on (give or take about 20 or so students). Course A had fewer students and therefore its variance is smaller than in course B where there were more students at the beginning of the course.

Overall Time Charts

In keeping with the theme of comparing the two course sections, the frequency of attempts per hour was graphed as shown in Figure 17 and Figure 18.

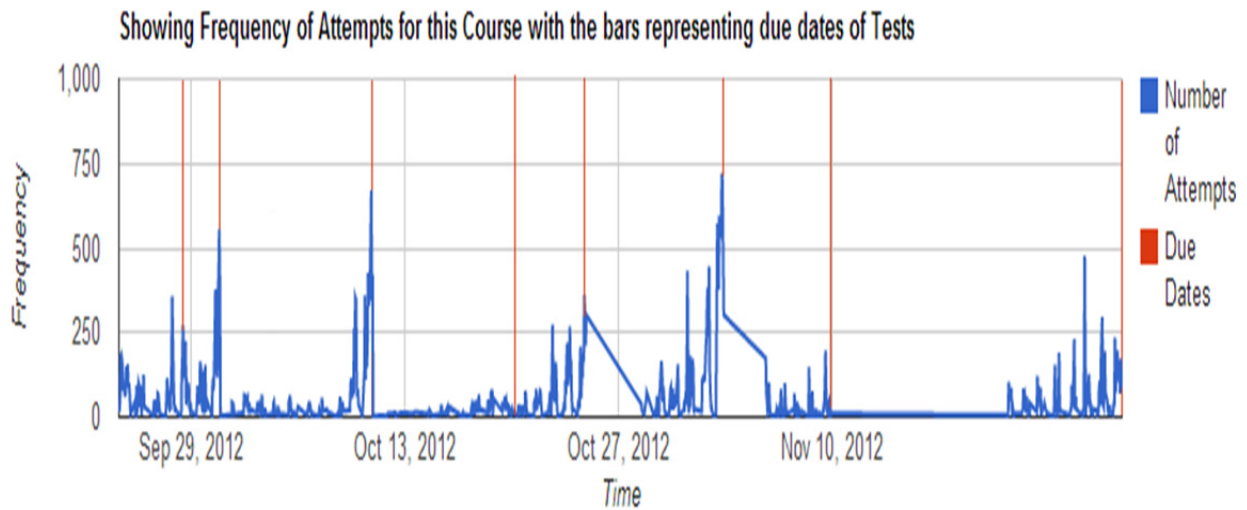


Figure 17. Showing the overall time chart for course A.

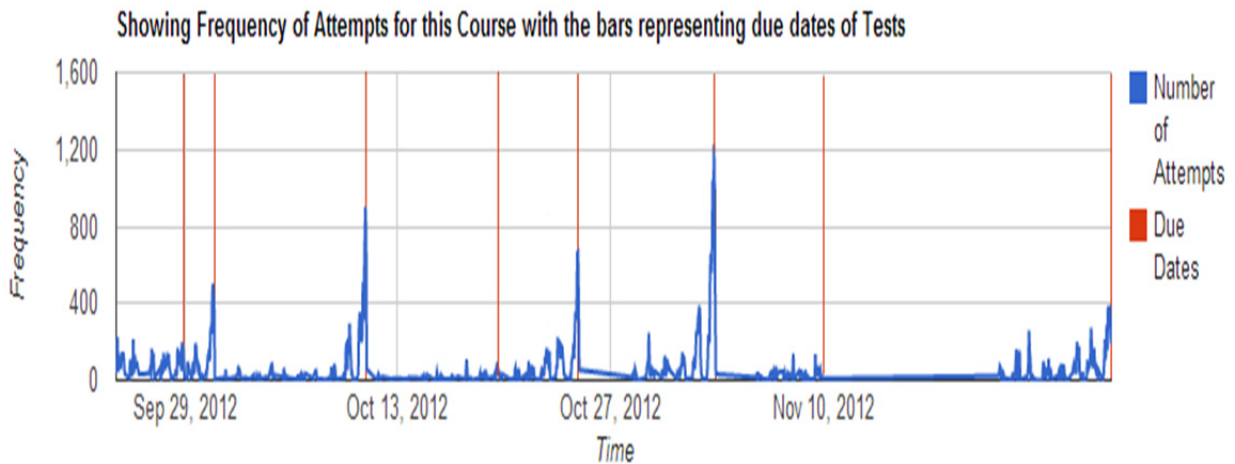


Figure 18. Showing the overall time chart for course B.

Both charts show the same characteristics. Just before a due date, there is a very large spike in the frequency of attempts, and then after a due date has passed, virtually all attempts stop. It is interesting to note that this applies to both courses, which means that the times students are making attempts are independent of the professor.

Time Chart Breakdown

While seeing the overall time charts for both course sections is interesting, it is an overall depiction, and it can be broken down further. The goal was to find the number of hours before a due date that the majority of attempts were made. What was done is to start at the due date line and work backwards adding up all of the attempts until 66% of attempts have been reached. 66% was chosen because it acts as the first standard deviation. Whatever point the 66% line is reached, that is how many hours prior to a due date the majority of attempts are made. For example we see assignment 5 for course A in Figure 19, and observe that the majority of attempts occurred just 27 hours prior to the due date.

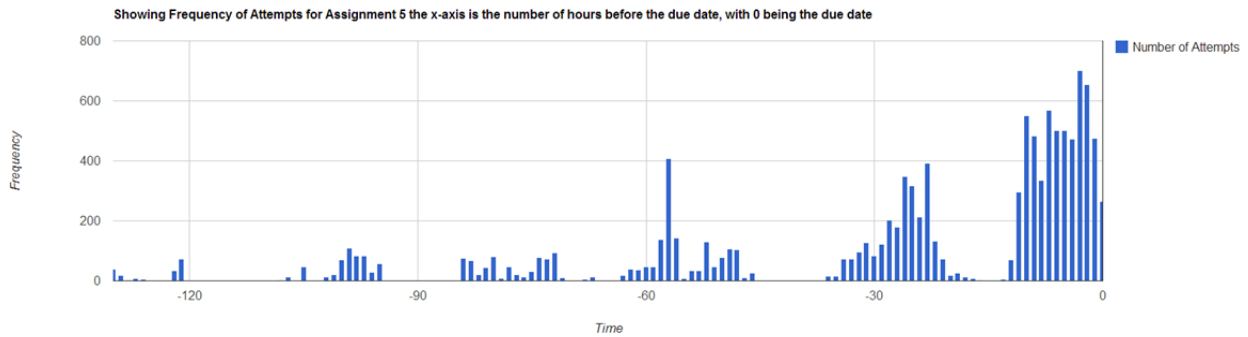


Figure 19. Chart to show the number of attempts per hour each hour before a due date. Assignment 5 course A.

Table 4 shows a list of the amount of hours prior to a due date that the majority of attempts were made. For example for assignment 5 in course A that number would be 27.

Table 4 For all assignments in course A, showing the numbers before due date that majority of attempts were made.

Course	Assignment	Hours before due date that majority of attempts were made.
A	Assignment 1	48 hours
A	Assignment 2	26 hours
A	Assignment 4	35 hours
A	Assignment 5	27 hours
A	Assignment 6	72 hours

A	Average for Assignments for course A	42 hours
A	Tutorial Quiz 2	95 hours
A	Tutorial Quiz 3	268 hours
A	Tutorial Quiz 4	58 hours
A	Tutorial Quiz 5	72 hours
A	Tutorial Quiz 6	77 hours
A	Tutorial Quiz 7	104 hours
A	Average for Quizzes	112 hours

What was seen in the above table is very interesting because there is a very large discrepancy between the timing of the quizzes and the assignments, within the same course section. One possible reason for this is that the quizzes were much smaller and took a lot less time to complete. Students who started the quizzes early were generally able to finish them in one go. While for assignments, they generally were not able to finish them in one sitting. The early students for quizzes did not make any more attempts closer to the due date, but they were making them for the assignments; which meant that those early students had a much larger effect on the timing than for the assignments.

Since there was such a large discrepancy between the quizzes and the assignments, the two have been broken up into two different aggregate charts.

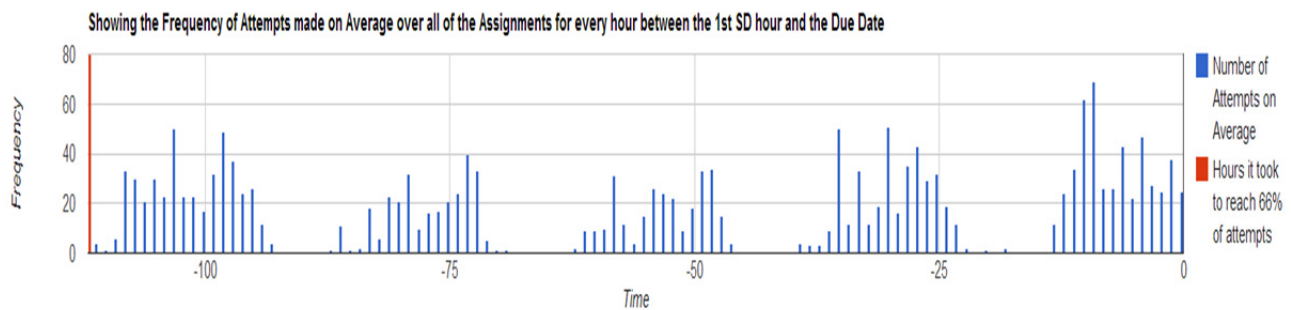


Figure 20. Time breakdown for quizzes in course A.

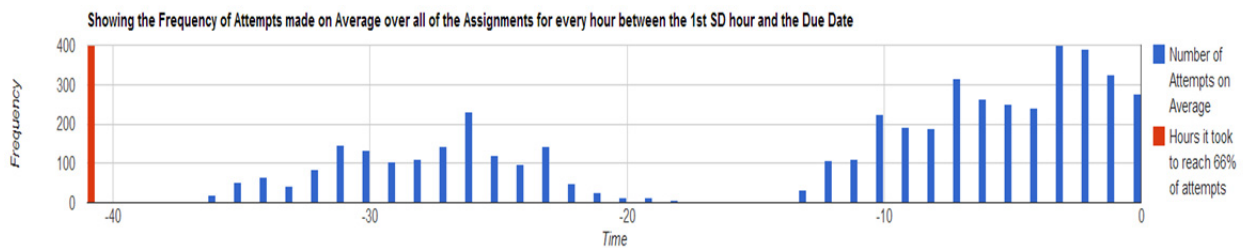


Figure 21. Time breakdown for assignments in course A.

It is also interesting to note the very apparent downtimes at night in both charts.

When looking at course B we see basically the same patterns so only the aggregate charts are presented here.

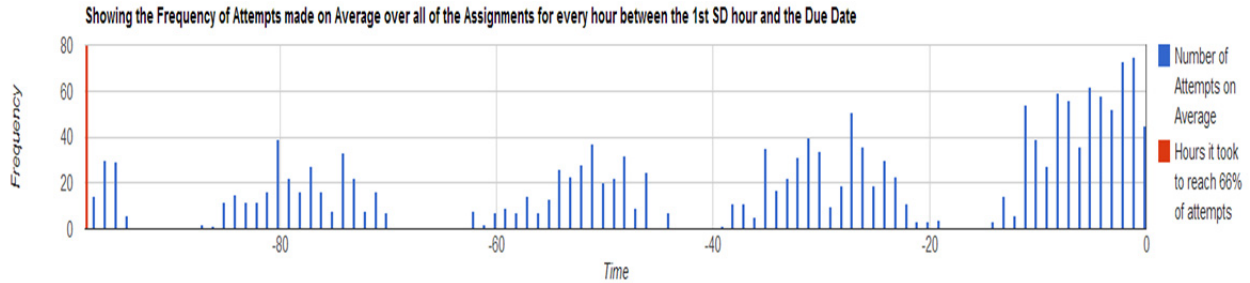


Figure 22. Time breakdown for quizzes in course B.

For assignments:

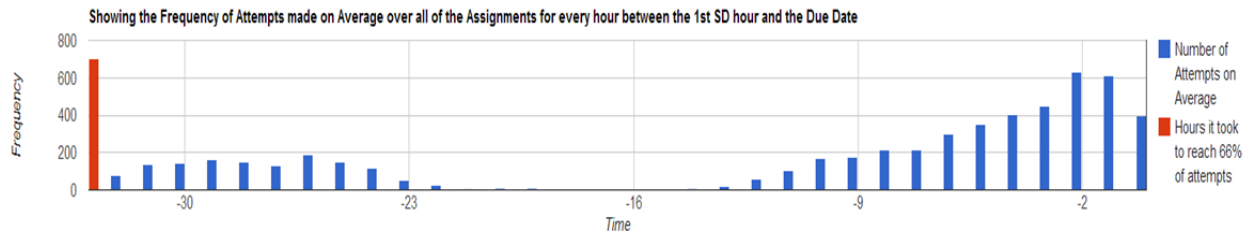


Figure 23. Time breakdown for assignments in course.

As we can see the two courses are very similar to each other. What is curious though is how for both courses the average amount of time to get 66% of attempts for assignments is 42 hours, while it is 112 hours for the quizzes.

Average Number of Attempts vs. Correct Rate for a Question Part

The next few experiments that were run examined the overall statistics of the course sections. This first one tries to compare the average number of attempts that students have made on a single question part against the overall correct rate of that question part. The correct rate is a simple calculation which computes the total number of correct attempts over the total number of attempts.

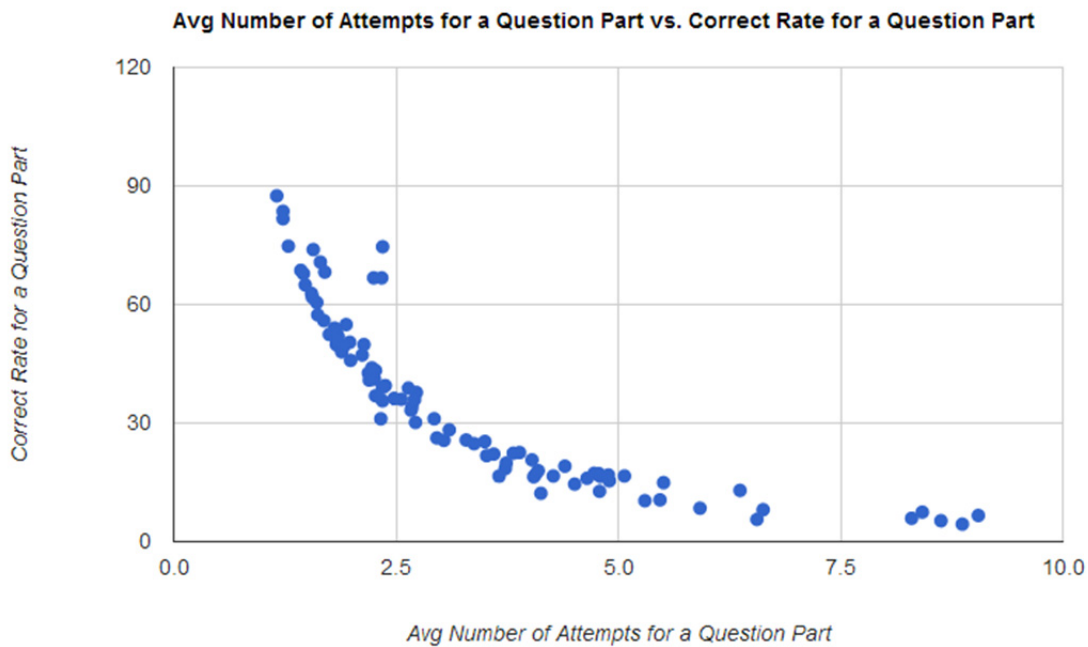


Figure 24. Average number of attempts vs. correct rate for all question parts in course A.

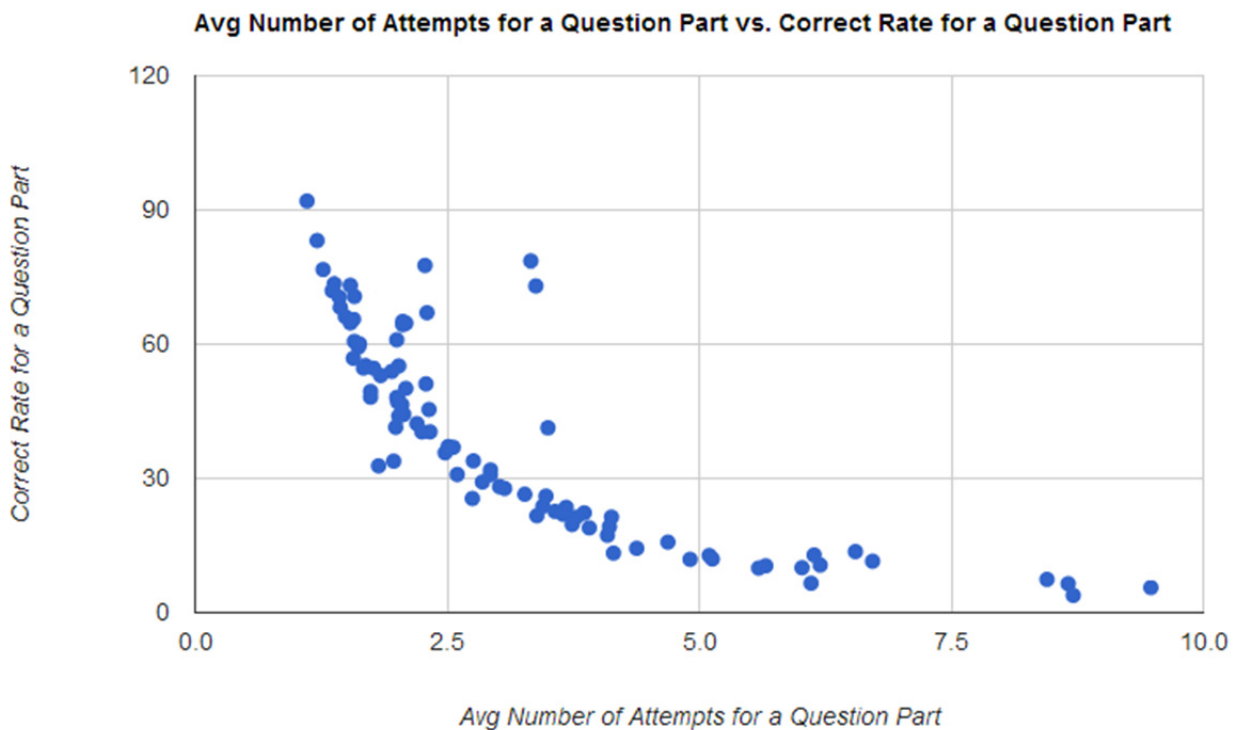


Figure 25. Average number of attempts vs. correct rate for all question parts in course B.

Both of the above charts clearly show a $\frac{1}{x}$ relationship which of course makes sense. As there are more attempts on a question part, its correct rate is going to have to decrease since there are always going to be a maximum number of correct attempts (total number of students), but there can be theoretically speaking at least, infinite number of total attempts.

Average Number of Attempts for a User vs. Overall Grade of User

As a brief interlude from the comparison of the two course sections, two analyses are presented that only apply to course A. This is because the overall grades of the students' were obtained for course A, but not for course B. This first chart shows the average number of attempts that a student made overall throughout the semester vs. their overall course grade. The goal of this test was to see if students who make more attempts get a lower overall grade (making more attempts, means they are getting more incorrect) or not.

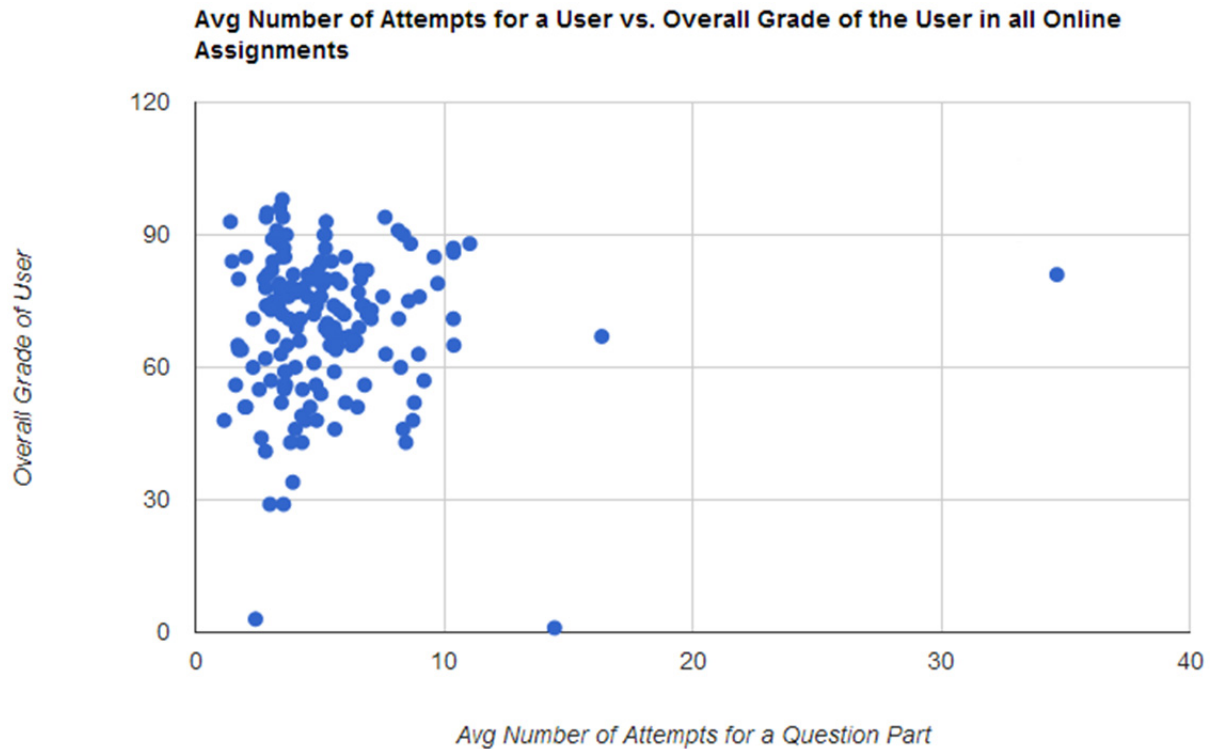


Figure 26. Average number of attempts vs. overall grade of students in course A.

The above chart shows no clear distribution, however, there are some very obvious outliers here. It is actually interesting to see that one student made an average of 35 attempts per question part (clearly somebody who is mostly guessing their answers) actually had an overall grade of 88.

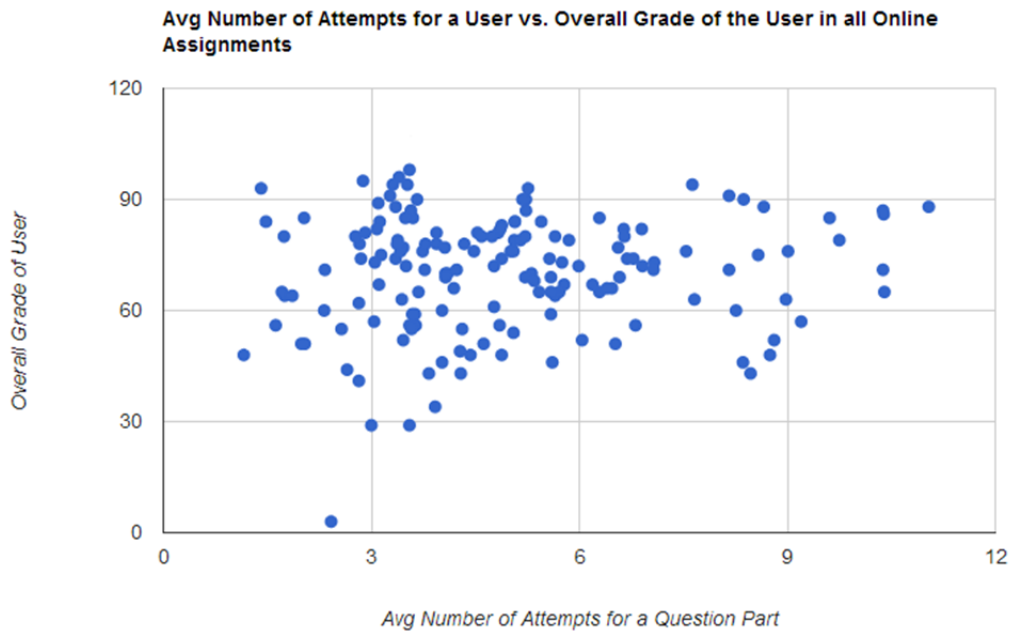


Figure 27. Average number of attempts vs. overall grade of students in course A, no outliers.

Looking at the new chart (Figure 27), we still see no actual pattern; however, a statistical regression was attempted on it. After fitting a linear model to the data, the statistical properties were analyzed.

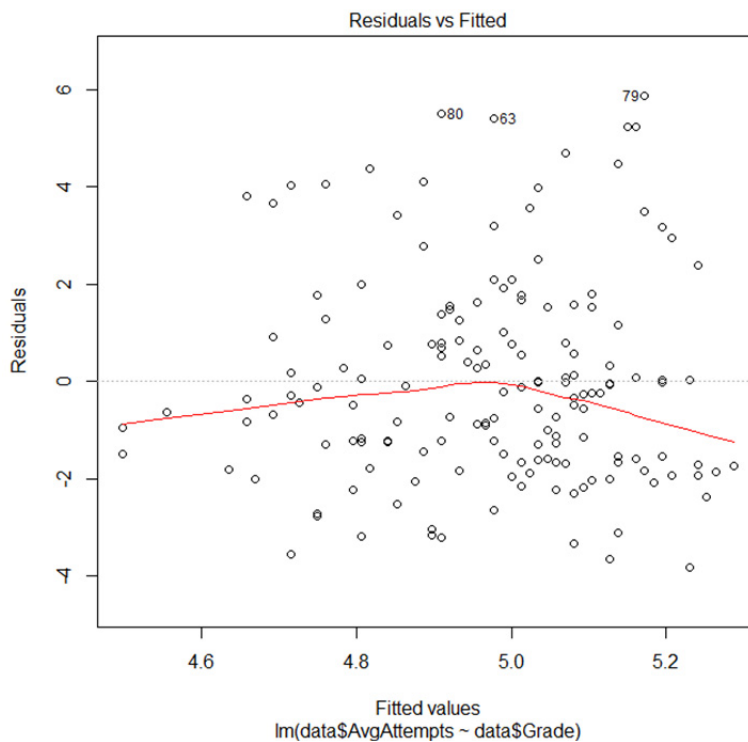


Figure 28 Residual graph of average number of attempts vs. overall grade of students.

The residual plot should show all of the data points scattered around the zero line. However, clearly this is not happening, as is shown by the red line that is drawn. This shows that the linear model that was fit to the data is not explaining the data itself. In other words, a line of best fit is fitted to the

data, and the residuals are the error from the line of best fit and the actual data point. In a good fit, the residuals would be randomly scattered which shows a good fit for the line, but unfortunately this is not happening. In addition to the residuals, it is a good idea to look at the R^2 value and the p value of the variable. It is observed that the R^2 value is only 0.006 which means that only a tiny percentage of the error is being explained by this line of best fit, which is another bad sign. Lastly it is observed the p-value of the Grade variable is 0.332 which is terrible as it needs to be below 0.05 to show anything. Therefore it is easy to state that the grade of a student does not explain the average number of attempts they make.

Average Number of Days started a Test before Due Date vs. Overall Grade

It was tested to see if there is a trend between the overall grade of a student, and the average number of days that they started a test before the due date. So if the due date was on October 19th, and they started a test on October 17th, then they would have started the test 2 days early. This value is averaged over all assignments and quizzes and plotted against the overall grade of the student.

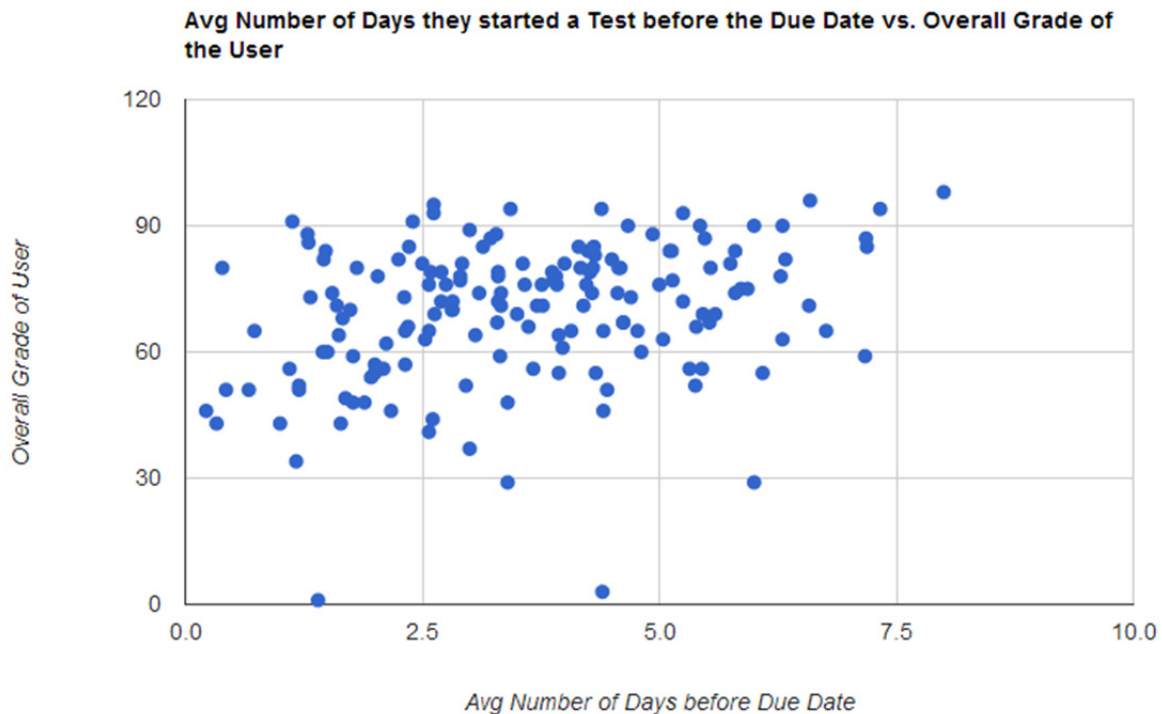


Figure 29. Average number of days started a test before due date vs. overall grade of student in course A.

The data-set looks as if there might be an upward trend, however it is not certain. Therefore a linear model is created by trying to explain the grade of the student using the number of days they attempted a test before the due date on average and it gave a residual plot as seen in Figure 30. The line of best fit can be seen in Figure 31.

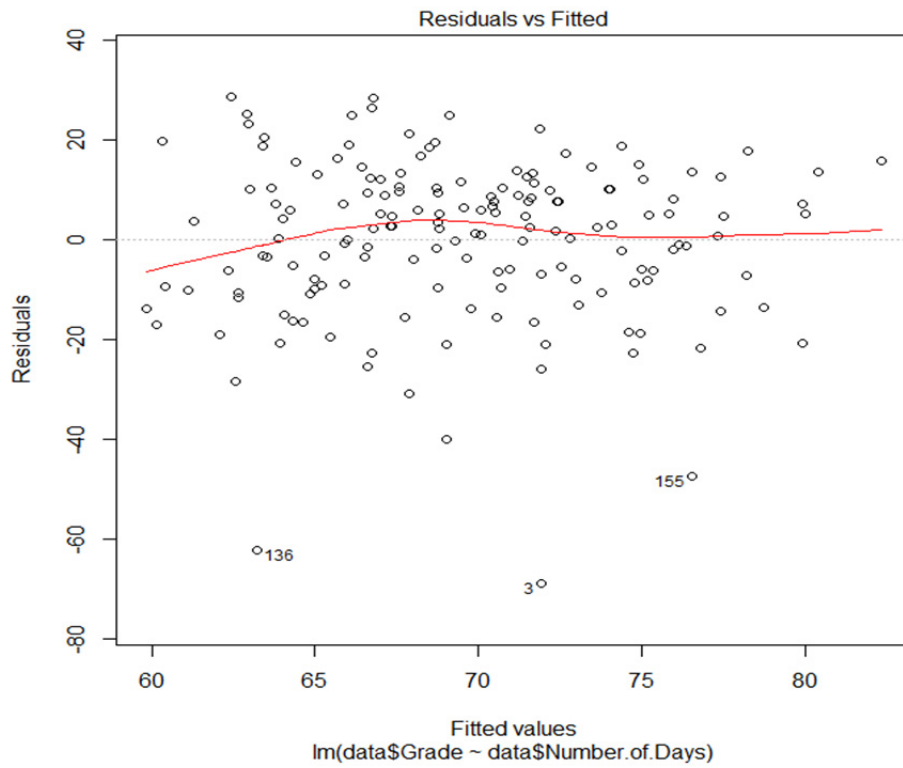


Figure 30. Residual graph of average number of days started a test before due date vs. overall grade of students. Course A.

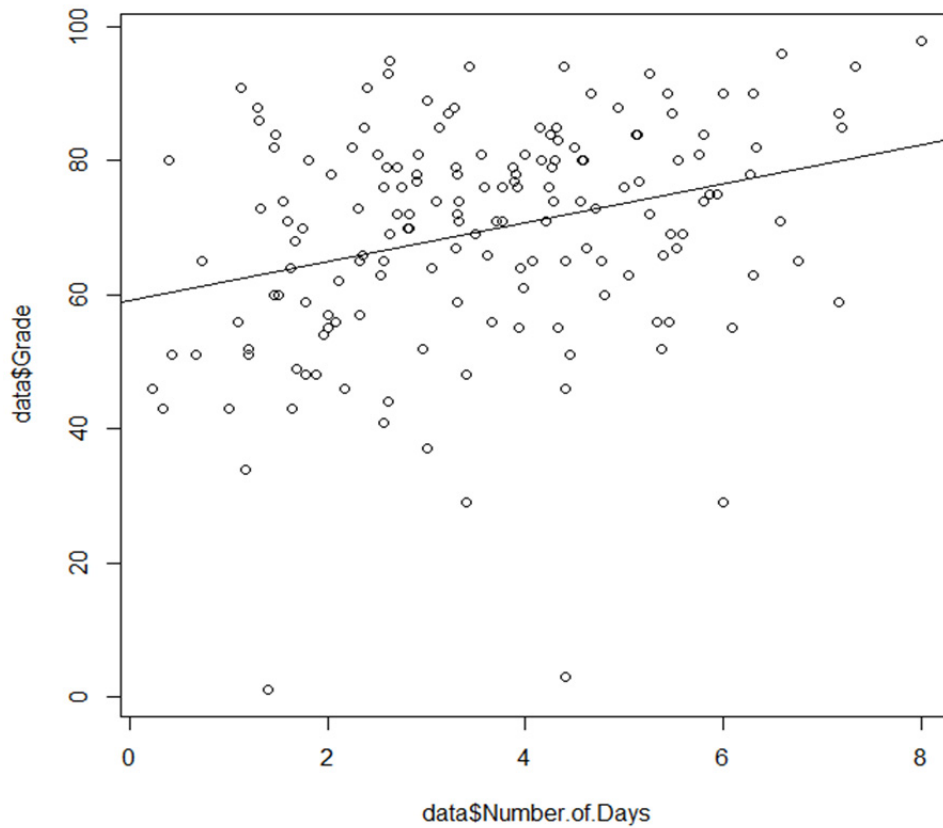


Figure 31. Line of best fit for average number of days started a test before due date vs. overall grade of student in course A.

The residual plot above is certainly much nicer looking (almost perfectly scatters around the zero line), than the previous experiment. The R^2 value here is 0.09, which again is not very high, but it is certainly higher than in the previous experiment. However, it is interesting that the p-value of Number of Days as explaining their overall grade is 0.000111, which is very significant at over 99%. What this says is that there is most likely some sort of relationship between how early a student starts an assignment, and their overall final grade. It is impossible however, to call this causation, since Overall Grade is such a complex variable that it would be foolish to attempt to explain it with something as simple as how early a student starts an assignment. This finishes the brief interlude which concerned itself only course A.

Scatter Plot of Missing Negatives

Returning to the comparison between course sections, it is observed that another way of identifying problem question parts is to create a scatter chart of all of the different question parts on the x-axis. On the y-axis is put a ratio of the number of attempts that were incorrect for this particular reason, over the total number of attempts. For example if 15 percent of all attempts for a particular question were incorrect due to a Missing Negative, then that data point would have a y component of 0.15. In this fashion it is very easy to notice outliers and to notice the reason for an outlier. This also helps professors identify areas where they need to focus more to make sure their students fully understand the concepts.

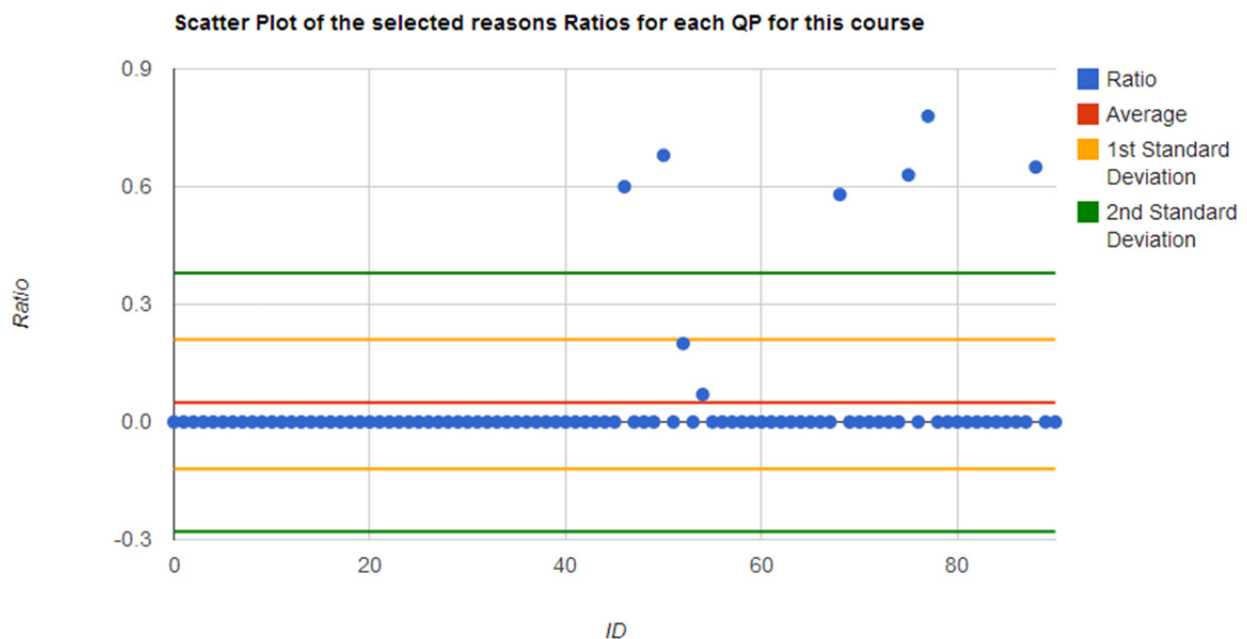


Figure 32. Scatter plot of "Missing Negative" for all Question Parts in Course A.

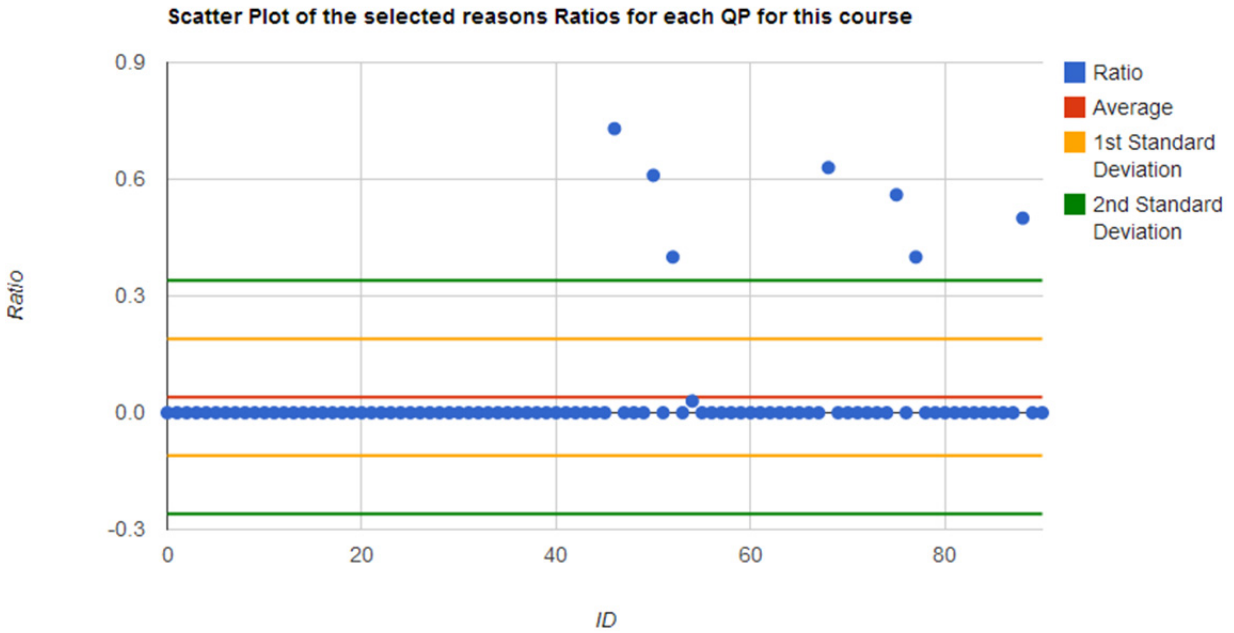


Figure 33. Scatter plot of “Missing Negative” for all Question Parts in Course B.

The above two charts show a fairly striking pattern, or lack thereof. Clearly most question part correct answers do not have a negative in them; therefore the ratio of missing negatives for most question parts is 0. However, for the question parts where there is a negative in the correct answer, the ratio jumps to a fairly high level. Between the two courses the reader can see that while the actual ratios are different, they are high (above 0) for the same question parts, which again makes sense. It is interesting to see that for both course sections, question part 55 has a Missing Negative ratio that is just slightly above zero. This probably means that the correct answer for that question part was very obviously negative.

Scatter Plot of Wrong Number/Value

Next the same scatter plot from above is viewed except this time the reason for incorrect used is the ratio of Wrong Number/Value answers.

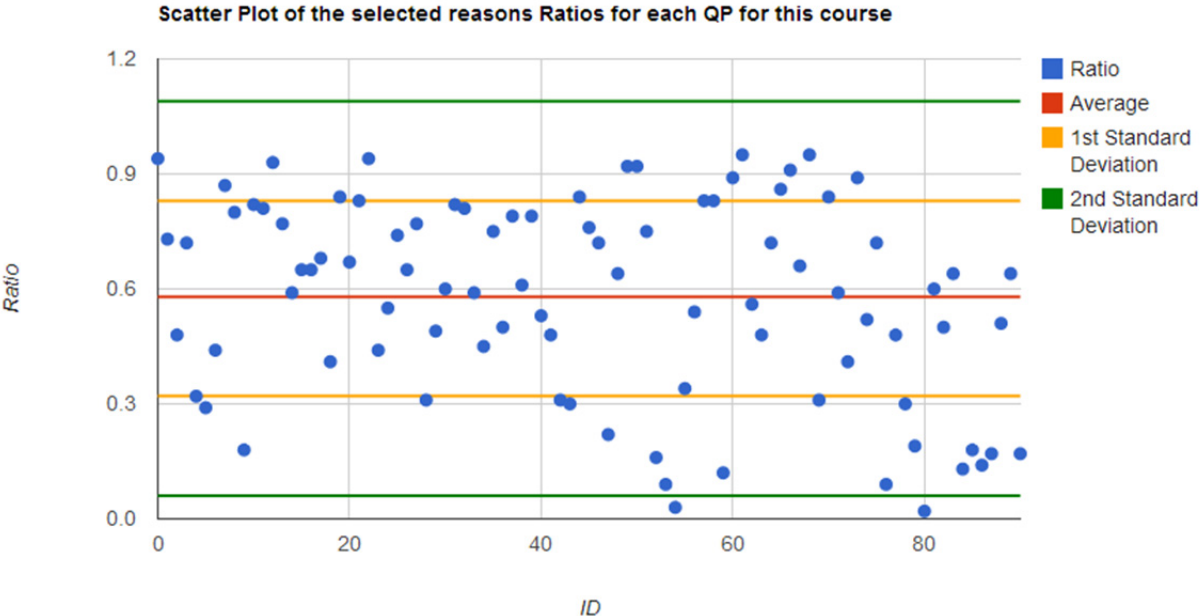


Figure 34. Scatter plot of “Wrong Number” for all Question Parts for course A.

As can be seen there is a large variance, with many question parts having a ratio of 0.92 which shows that they are terribly done as compared to the rest. This means that 92% of attempts were wrong due to wrong value. If the reader will recall from the Question Part Breakdown section, it was identified that a question part from assignment 6 (number 61 here), was the largest outlier, showing that it had been done terribly relative to the other question parts. Once the assignment ID's were compared it was found that that same assignment 6 question part was one of the question parts in this scatter plot with a ratio of 0.92. It did not have any negatives, so it did not show up in the Missing Negatives scatter plots.

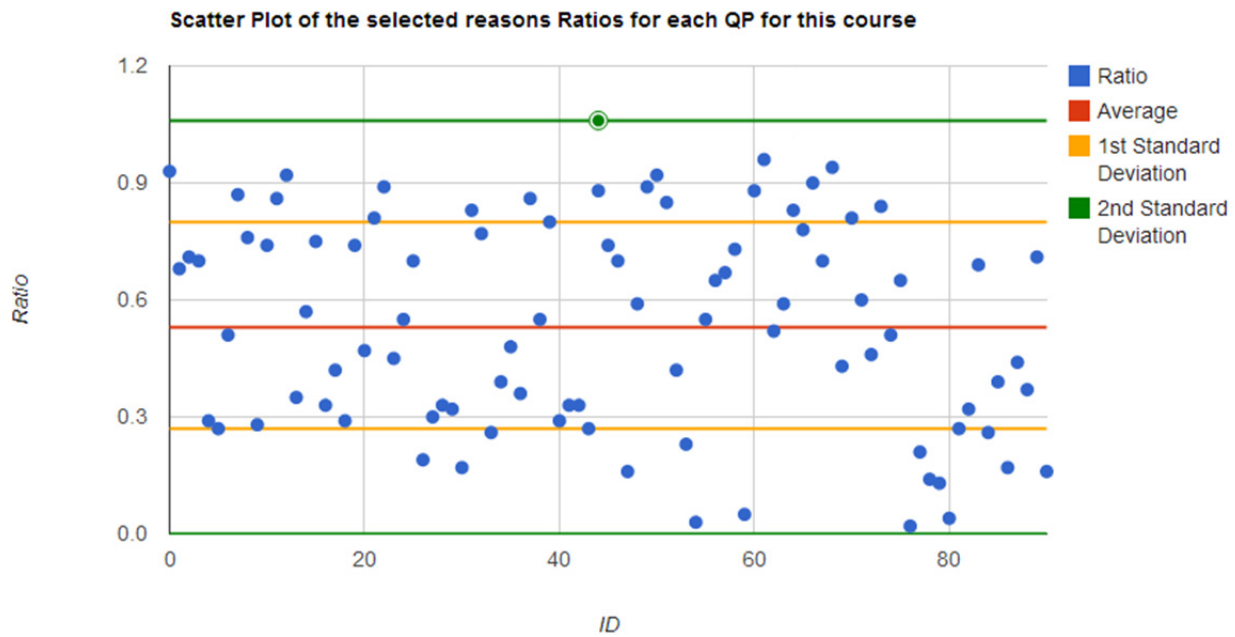


Figure 35. Scatter plot of “Wrong Number” for all Question Parts for course B.

Looking at the same scatter plot as above but for course B, it is obvious that the two are fairly similar. The ratios are slightly different, but only slightly. It is possible to see that number 61 is the largest outlier here, which again goes to prove that problem questions in one course section are problem questions in both course sections.

Scatter Plot of Within 5%

Up next comes the scatter plot showing the reason for being incorrect as a Wrong Number/Value but being within five percent of the correct value.

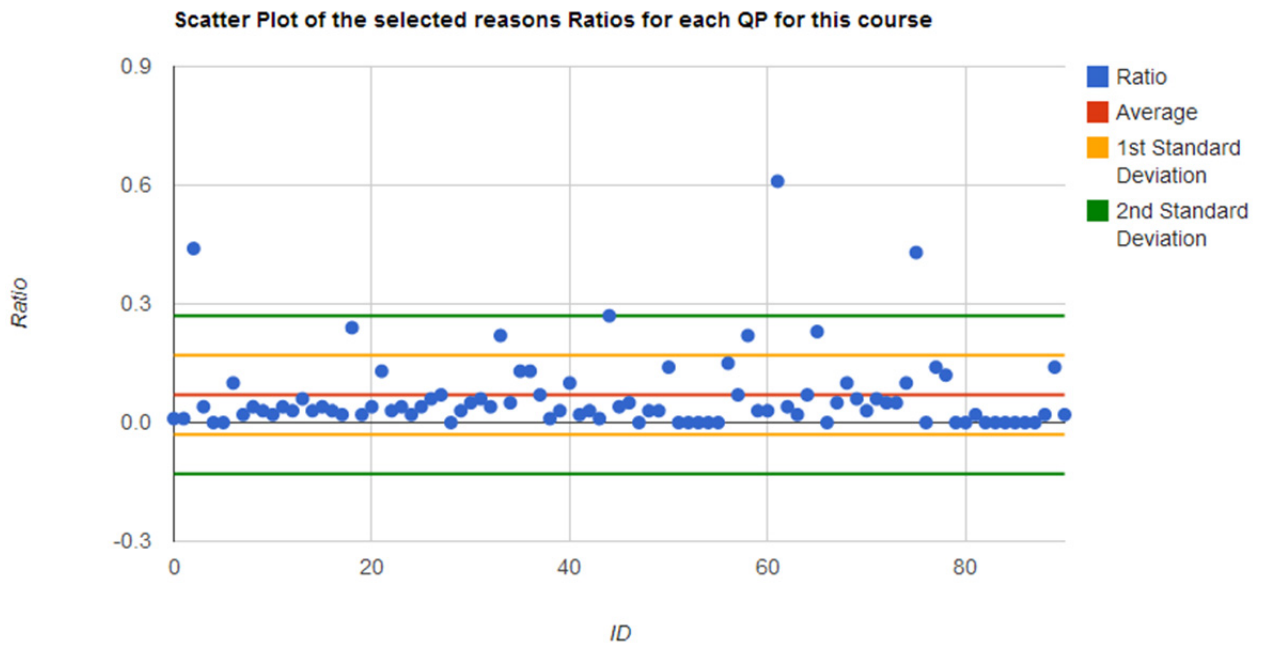


Figure 36. Scatter plot of “Within 5%” for all Question Parts for course A.

Most of the question parts from above fall underneath the standard deviation lines and so are not very large outliers meaning that relative to the rest of the question parts, they all have about the same number of incorrect attempts that fall within the within five percent category. Interestingly, number 61 is the largest outlier again, which means that while most students were incorrect with their answers for that question, they were very close.

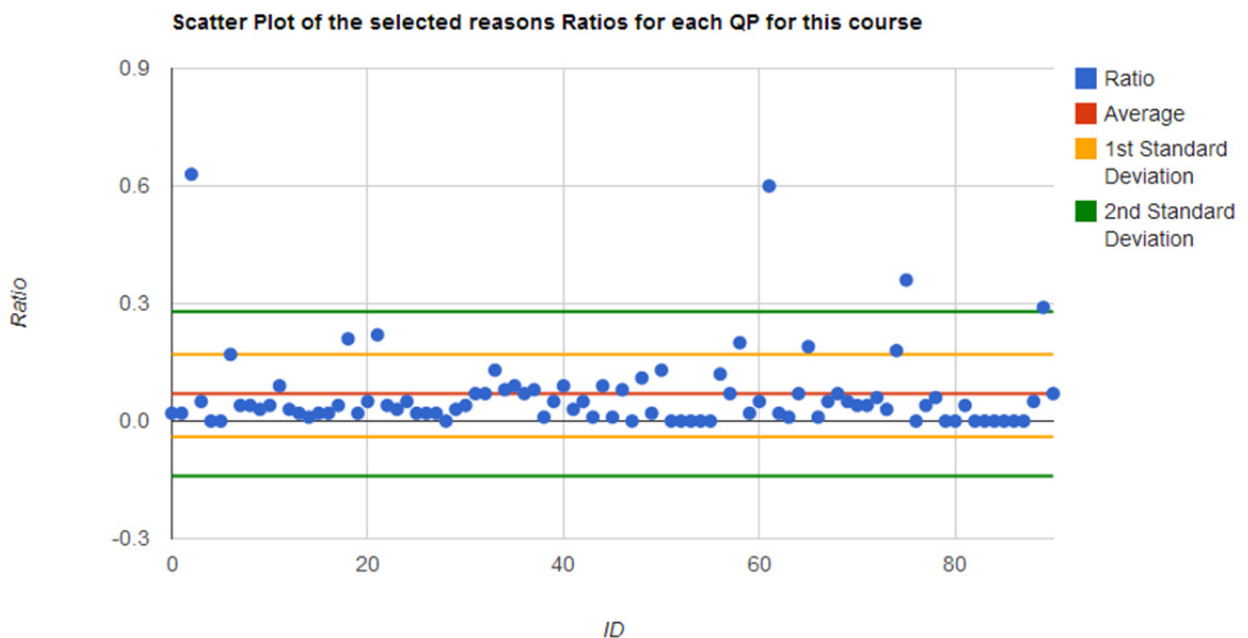


Figure 37. Scatter plot of “Within 5%” for all Question Parts for course B.

The same patterns can be seen for course B as can be seen for course A. However, in this case number 61 is not the worse overall; it seems that number 2 is which is interesting.

Scatter Plot of Wrong Units

Below are the same scatter plots from above but this time for Wrong Units.

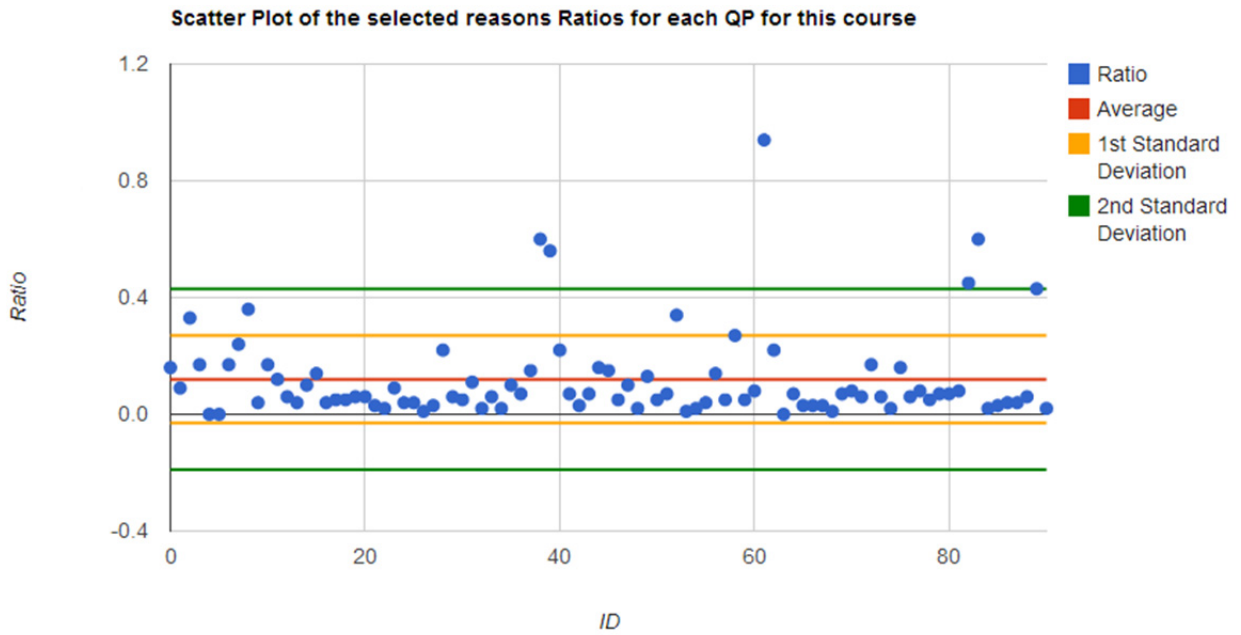


Figure 38. Scatter plot of “Wrong Units” for all Question Parts for course A.

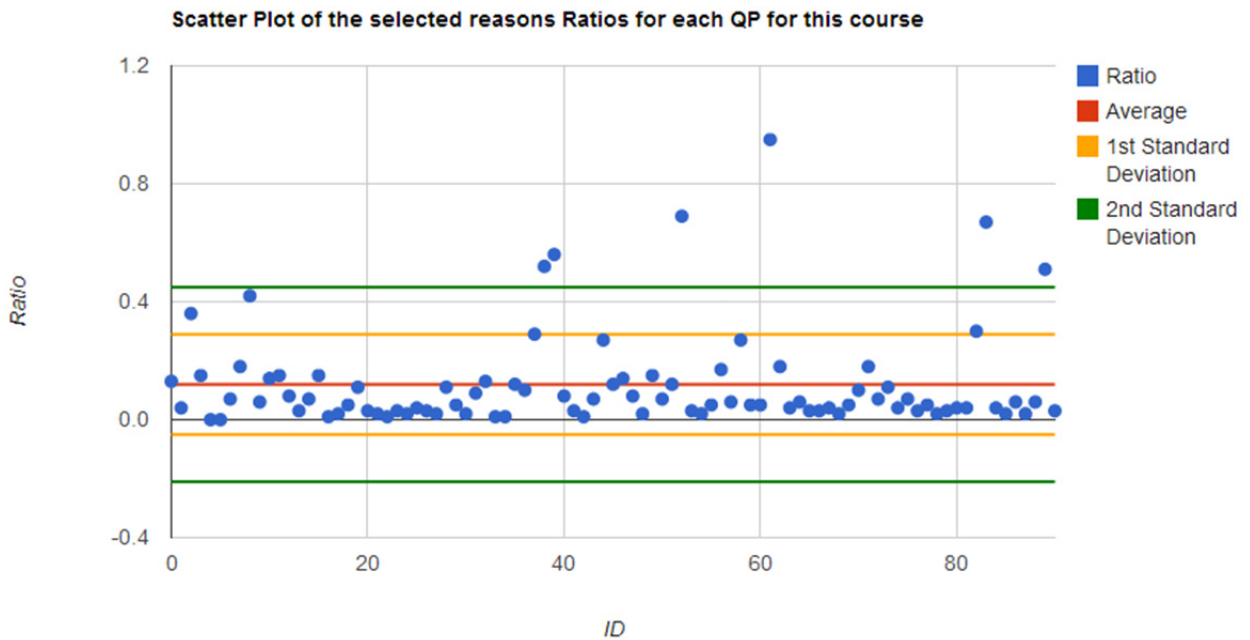


Figure 39. Scatter plot of “Wrong Units” for all Question Parts for course B.

The plots for both course A and course B are fairly similar, with a few points standing outside of the second standard deviation line. And of course our old friend number 61 shows up again as the question part that was done the most poorly, with a ratio of 0.92 for both course sections. This of course is very terrible.

What does seem fairly obvious from looking at the answers just overall and the charts above is that getting the units wrong is a major reason why many answers are incorrect. Also from talking to students who take the course, it is obvious that many students get incredibly frustrated when they do not know how to enter the units (or the power term for scientific notation) in correctly. Therefore one major change that needs to happen with the system is to give better directions on how to enter the values properly.

Most Common Answers

Table 5 below contains the most common answers overall in the system. These include correct and incorrect answers for both course sections.

Table 5 Table of the most popular answers overall, correct and incorrect for both course A and B.

Student Answer	Frequency
No Answer	3506
0m/s	1179
0 m/s	736
9.8m/s ²	486
0N	457
$v_{S} = v_{L}$	359
0m	356
9.8 m/s ²	314
A	292
The force is toward the sun	291
0	281
0 N	281
-10km/h	226
5.4m/s	225
0J	224

9.9m/s	215
357m	213
20km/h	201
0 m	185
0km/h	184
-40km/h	179
-10 km/h	170
147J	170
4.9m/s	168
9.8m/s	166
0 J	161
20 km/h	155
6.16N	149
-40 km/h	141
0 km/h	134

It is very interesting to note that of the top 30 most common answers, 10 were some variation of zero. Either 0m/s, or 0m, or just 0, ten of the most common answers were zero. This is fascinating as it shows the types of questions that the professors give to students. Apparently professors like a question that has a zero answer.

Conclusions

The above charts and plots and tables showed a few points about the course sections and their question parts. First it is fairly obvious that the two course sections are very similar to each other. While one course had more wrong number attempts, another had more that were within five percent. While one course had more missing negatives, another had more wrong units. Both course sections' students messed up on the exact same questions, almost in an identical manner with their ratios being very similar. And the students' from both sections were attempting their assignments at the same time as well. Neither sections' students were starting earlier nor later, there were no question parts that were done very badly for one question part, while being done well for another question part, and about the same proportion of students who started the section finished it for both sections. This is fairly remarkable given that the two sections were taught by different professors. It is also interesting to note

that as predicted, question parts that were attempted more times had a lower correct rate than those question parts which were attempted less frequently. Lastly it is fascinating to note that students who attempt an assignment early will get a better mark overall, as compared with their peers who start an assignment later. Unfortunately there was no such pattern for the average number of attempts affecting the students' grades, which had been predicted at first. The reasoning of course was that if a student makes very many attempts, they must not know the course material very well and so will do worse overall on the midterms and final exam, however, this did not pan out. Students who made very few attempts were getting high and low grades and students make lots of attempts were also getting high and low grades.

In addition to the overall analysis, it was found that one question part in particular stood out as being very badly done. Therefore a further analysis of it shall be done in the subsequent section.

Detailed Analysis of a Problem Question Part

From the previous section it was seen that question part 61 from assignment 6 with the text of: "What is the magnitude of the gravitational force acting on the planet due to the sun? (Use scientific notation with three significant figures)" and had the question template part id of 6303 was done terribly. Therefore further analysis shall be done on the question part.

Popular Answers:

First we look at the 30 most popular answers overall blocking by the correct answer so that we can ignore the randomization done.

Table 6. A table of the most common answers from question part 6303 from Assignment 6, blocking by the correct answer.

Correct Answer	Student Answer	Frequency
3.2773580246913573E21 N	3.28 N	96
2.123728E21 N	2.12x10 ²¹ N	53
8.848866666666667E21 N	8.85E27N	50
3.2773580246913573E21 N	3.28x10 ²⁷ N	44
3.539546666666667E22 N	3.54x10 ²⁸	35
1.6591624999999996E22 N	-1.659x10 ²⁸	30
5.417673469387754E21 N	5.41x10 ²¹ N	29
1.0618639999999997E22 N	1.06E28N	27

7.37405555555554E21 N	7.37 x 10^(21) N	27
2.123728E21 N	2.125x10^37	26
7.37405555555554E21 N	7.37E33N	26
2.949622222222216E22 N	2.95x10^28	25
8.494912E21 N	2.308x10^36	25
7.37405555555554E21 N	2.21E33N	24
1.32733E22 N	1.33x10^28 N	22
2.949622222222216E22 N	2.95x10^28N	22
3.185592000000003E21 N	3.19 x 10^27	20
3.185592000000003E21 N	3.19 x 10^27 N	20
8.84886666666667E21 N	8.85 x 10^27 N	20
2.123728E21 N	2.12 *10^21 N	18
2.123728E21 N	2.12 x 10^21 N	18
8.84886666666667E21 N	8.85x10^27N	18
1.061863999999997E22 N	2.65E36N	17
2.123728E21 N	2.12 x10^21 N	17
2.949622222222216E22 N	2.95E28N	17
7.37405555555554E21 N	2.212E33N	17
8.84886666666667E21 N	4.99 x 10^27 N	17
1.061863999999997E22 N	2.66E^22N	15
2.123728E21 N	2.12 x 10^27 N	15
2.949622222222216E22 N	3 x 10^22N	15

As we can see most of the answers at the top are all wrong. Either because of units (missing them) or because they had the right value but not the right power (e.g. the top answer is 3.28 but it should be 3.28 E21). As we can see from majority of the answers above, the student answers have been correct in their actual value, but they have gone wrong either in the format of the scientific notation, forgetting units, or simply forgetting to add a power term to the answer. What this is saying is that students were unaware of how to properly enter in scientific notation numbers (can be fixed easily with a quick description in the question), or are simply not entering in units and power terms due to being lazy or simply not remembering.

Further from the scatter plot section, this question part had a ratio of 0.92 for both Wrong Value and Wrong Units. Therefore, students were getting the value wrong and the units wrong 92% of the time, which is too high. The question was most likely very difficult, and that is not something that need necessarily be changed. Unfortunately there are still limitations with online education systems such as these and so students were unfairly getting answers incorrect by simply not knowing how to enter them properly. At the risk of repeating this point too much, it is very important that more explanation is given about how to properly enter in an answer, especially when it comes to scientific notation.

Future Work

The following are items for future work.

Equations

It would be very exciting to be able to implement the equation section that was discussed under the Methodology section. To be able to make an educated guess as to what equation a student used to get their incorrect answer would be a very exciting and novel thing. At this point the only error encountered was the difficulty in integrating Dr. Lawrence's parse tree code into the project code so that the parsing found in that code could be utilized instead of having to re-invent the wheel. Once the parsing was properly implemented, a number of for loop iterations would have to be run for every possible variable that may or may not be in the equation, along with switching all of the operators and adding in extra terms from the question. For each simply check to see if the students answer is being returned and if so then a possible equation has been found.

Automatic Hints

A very interesting idea proposed early in the project was to create a system that would automatically generate hints for questions that were done poorly. It would utilize the most common incorrect answers, the equation section from above, and once the program knew what was going wrong in the question it would be able to create a hint that would try and help students answer the question better.

Implementing this would be slightly more difficult than the equation section. The reason for a larger challenge is because this entire Thesis has focused on the professor side of the website. No work has been done on the student side. However, another project has been running in parallel that has been updating the student side and it unfortunately (due to legacy reasons) has a very complicated way of

accessing student answers and data. However, if this can be overcome then it would be a fairly easy task to implement on the user side as the page that generates the questions would simply have to do one extra database check. Of course implementing the program that would create the hints also would not be trivial.

Alerting Students about Guessing in Real Time

As described in the Student Information Section, students who “Guess” the correct answer (answering consecutive numbers in the range that the random number generator works), are labeled as “Guessers” and an alert is created for the professor to see. However, what would be very interesting is to create a system that would automatically recognize the fact that a student keeps guessing consecutive numbers and alert the student about it as they are still answering questions. This would serve the purpose of dissuading students from doing this and would force them to actually attempt the question.

Any work attempting to implement this would face the same obstruction by the fact that accessing student answers and data is quite complicated on the user side. However, once this has been overcome, it would be an almost trivial problem of keeping track of their last few answers on the client side and just comparing against that list whenever a new answer comes in. If at any point they are deemed to be guessing then an alert would be shown. If they keep it up then, they could be repeatedly warned with the final act being a suspension of their account.

Conclusion

For over eight months, this data has been collected, analyzed and studied. The author has been excited and fascinated throughout this period about how large and varied a data-set this was, and how there are almost endless possibilities in investigating it. Throughout this investigation two main objectives were completed: a professor dashboard and an in-depth analysis of the data.

The dashboard is already in use, and can be used to alter question parts and assignments by the professors of the course to reflect the alerts that were prepared by the automatic data mining that occurs every day. The dashboard also informs professors as to how their students are doing in a very detailed and exact manner, from statistically ranking each student relative to the other students, to giving an overall grade and a breakdown of all of the grades of the student for all assignments, from a full breakdown of the average number of Wrong Number/Value attempts, Incorrect Unit attempts, etc... to allowing a professor to see every single answer that a student has ever submitted on the system. All

of this data is presented in a clear and user friendly way and does not require any major learning curve to understand. In addition, the dashboard provides detailed information about each assignment given out by professors. It tells them about any pressing information in the form of alerts, and provides detailed information in the form of a breakdown of the reasons the students are getting question parts incorrect. It also features a histogram of all student grades for this assignment, and a time chart of when students are making attempts. Next the dashboard provides in-depth information about a course overall and alerts the professor to any assignments that were done poorly in relation to the rest. It shows a time chart of all student attempts superimposed on the due dates, to check for patterns and shows a breakdown of all question parts in total. Lastly the dashboard also provides a full histogram of all students' grades for the entire course and allows a professor to be entertained by the live answer chart which updates itself every second. On the dashboard there also exists a link to a page that gives all of the detailed information used for the in-depth analysis in real time.

Secondly this Thesis attempted an in-depth analysis of the data-set that was acquired over the semester. This in-depth analysis provided some fascinating insights into the differences between the two course sections, or lack thereof. While there are differences between the two sections, they generally offset each other. While there are more wrong number attempts for one course, there are more wrong unit attempts for another. In both sections, the same question parts posed a problem to the students and the students made attempts around the same time for all of the assignments. In addition it was found that students who attempted an assignment earlier were likelier to get a higher overall mark, while having more or less average attempts overall did not affect their overall grade. Both sections' students made the same proportion of errors from the same results (92% of incorrect answers were due to Wrong Numbers, etc...) and for the two sections it was found that the majority (66%) of attempts were made only 37 hours before the due date of an assignment.

One of the most important findings from the in-depth analysis was the result that many students actually have the correct answers when they are marked incorrect, and have simply messed up the units in some way. They might have entered in the units incorrectly (especially for scientific notation), or their syntax was just wrong. Due to the constraints of the computer interpreting their answers, there must be a specified format for all answers and anything differing from that will invariably be marked incorrect. Therefore, professors should give more hints as to how to properly enter in the correct answer to prevent a large amount of frustration from their students.

In essence the work done on this thesis was just a drop in the bucket with what could be accomplished with this data and with the number of students who are using it. The possibilities are

almost limitless. However, with just the work accomplished for this thesis, a very real and professional teaching tool has been created that can help professors and students like never before.

Acknowledgments

I would like to thank my supervisor Dr. Ramon Lawrence, for his guidance, support and many ideas. Without him of course none of this work could have been accomplished. In addition I would like to thank everyone in the Database Laboratory that helped in some way with project ideas, especially: Eric Wein, and Giuseppe Burtini.

Appendix A. Technical Summary

The system was built over the full year, and it therefore has a fair amount of code (JSP, JavaScript) that is not being used for the final product version. The technical summary will describe only the relevant pieces of code and the pages that are in use for the current implementation.

Open Source JavaScript Libraries

Google Charts

The Google Charts library is an open source JavaScript library that can be used to draw charts on a webpage. It is very easy to use with a simple import and the functions required to do the actual drawing are also quite simple. More information can be found at <https://developers.google.com/chart/>.

HighCharts

The HighCharts library is another open source JavaScript library for creating and drawing charts. Predominately Google Charts was used for chart rendering; however HighCharts has functionality for live charts. Live meaning that every x number of seconds it will redraw the chart with the new data on it instead of having it be static. More information can be found at <http://www.highcharts.com/>.

Bootstrap

Bootstrap is a combination CSS3 and JavaScript library. It allows for easy website design. The JavaScript allows a programmer to use the alert systems and progress bars, etc... Examples and the library can be found at <http://twitter.github.io/bootstrap/>.

Dashboard Report Generation

For the dashboard to work in a timely manner, report generation must be done the night before which creates the data that will be queried whenever a professor goes onto the dashboard. The report generation is run every night at midnight and is made up of four different pages that all run JSP code. They all combine the necessary data and store it into special database tables. The data is stored in a specifically separated fashion which can be easily parsed by the client side. Therefore most of the data is dumped to the database in a comma separated fashion that is recognized by the JavaScript.

`getAssignmentReport.jsp`

This page finds all of the assignments for the current courses. For each of these assignments, it first gets the average course information. The average course information is provided by the `allQPwrongreasonchart.jsp` page which is accessed by a GET request for each course and

getting all Assignments for that course. It then parses the data that is passed back to get out the relevant data about the averages and the standard deviations.

Once this page has all of the average information for each of the courses it is running, it finds all of the assignments for each of those courses and finds all of the answers that were filed for this assignment and categorizes them by using the `getQPWrongReason` page. It then finds the total number of students who have attempted the assignment. For each assignment, it also finds the overall mark for all students for this assignment. Lastly it finds all of the times that students have made attempts on the assignment as well.

Once all of the data has been acquired (mostly stored in ArrayLists), it is all stored in the `assignmentReport` database table.

`allQPwrongreasonchart.jsp`

This page takes in GET parameters from the user and creates charts based on all of the different question parts for the course and assignment given by the user. What this means is that it finds the total number of Wrong Number/Value, Wrong Unit, Missing Negative, No Answer, and Within 5%, for all question parts of all assignments for the given course. It then averages each of those, and finds the standard deviations and creates a chart out of this data. The chart is created using Google Charts as usual.

`getQPstuff.jsp`

This page is called by many of the report generation pages. It has one function `getQPWrongReason` which takes in a parameter of a course, a test and a connection. For this particular test in the specified course, it goes and finds the number of wrong number attempts, the number of wrong unit attempts and so on. It does this by first finding all of the question parts for this assignment and then calling up the function `getQuestionPartAnswersEACHCOURSE` from the `getAnswers.jsp` page. This returns the total number of wrong number attempts and wrong unit attempts, etc. for this one question part.

Next this page finds the average number of attempts per user for each question part, and then returns the total numbers from wrong value attempts, wrong unit attempts etc. and also returns all of the standard deviations and averages.

`getQuestionPartAnswersEACHCOURSE()`

This exists in the `getAnswers.jsp` page and takes in a parameter of a question part. This function will find all of the attempts made on this question part and it will return the total number of

correct attempts, attempts, wrong number attempts, wrong unit attempts, missing negative attempts, no answer attempts and within 5% attempts.

`getQuestionPartAnswersIndividualReturnNumber()`

Takes in an individual answer and the correct answer and will return whether or not the answer was incorrect because of wrong value/number, wrong units, within 5%, missing negatives or no answer.

`getStudentInfo.jsp`

This report will find data about each of the students. For each student in specified course, it will find all of their answers for all of the question parts for all of the assignments that they have done. For each answer it will run the answer through the function `getQuestionPartAnswersIndividualReturnNumber()` which returns whether or not this answer is incorrect because of what reason. Then it is also going to pattern match for button mashing and guesser just by keeping a list of former answers and checking if this new is the same or very close to previous answers.

Next this page will find the overall mark of the student for each of the assignments in the course that they belong to. Lastly it is going to create a number of string variables which will all be in some way separated and put all of the data into those strings. The strings are then inserted into the special database table `userReport` which holds all of the data about all of the students.

`getDailyReport.jsp`

This page is the simplest of all of the report generators and it simply queries the database for all of the answers that were attempted in the last 24 hours and runs each answer through the `getQuestionPartAnswersIndividualReturnNumber()` function. It aggregates all of this, counts the number of users, who those users were, and who made what attempt and then dumps all of this data to the database in the same vein as the previous ones.

`getCoursesReport.jsp`

Initially this page will calculate all of the average and standard deviation values for a course in general. It does this by calling the `allQPwrongreasonchart.jsp` page to do the work. For each assignment in the course it finds the total number of students who have attempted it. Then the data received from `allQPwrongreasonchart.jsp` is stored into ArrayLists about each assignment (this data is the total number of incorrect attempts, correct attempts, missing negative attempts, etc...). This data is then standardized and aggregated to create averages and standard deviations for the overall

assignment data. In addition each attempt's timestamp is stored in an ArrayList for further analysis later. Lastly all of this data is dumped into the `courseReport` table in the database.

[realtimedata.js](#)

This page is called from `index.php` and simply acts as a method to call and draw the live data chart. The live data chart is built in HighCharts as it allowed for the simplest implementation of a live chart. It queries the database using AJAX calls to `getlatestdata.jsp` every second and takes in the returned data and parses it. If some data has been returned it means that this is new data and should be displayed on the chart. Therefore it goes ahead and adds the data to the series arrays of the chart and calls up the proverbial "repaint" function which re draws the chart with the new data point added in. For each data point drawn onto the chart it is possible to click on it, which will display a URL link. The link will take the user to the `livedatainfo.jsp` page.

[livedatainfo.jsp](#)

This page takes in the test id of the attempt that was clicked on in the live data chart and it finds out who the student was that made that attempt, for what question part of what assignment it was and the statistics of the individual student and the question part. It also shows all of the attempts made by this student for this particular test and whether or not each attempt is correct/incorrect and why.

[getlatestdata.jsp](#)

This page acts as the server side method for the live data chart. It gets a request from `realtimedata.js` in the form of an AJAX call with 2 possible parameters: whether or not this is a first call, and the date of the last displayed data point. If "first" parameter is true that means that the page is just being initialized and it needs to acquire some data points to display. It queries the database for the last 15 data points and for each it checks whether or not they are incorrect and if so it checks why the answer is incorrect. It puts it all of this into a string which it then returns to the JavaScript page. If the "first" parameter is false that means that the chart is already initialized and it is simply checking to see whether or not there have been any attempts made since the last data point displayed (the date of which is the second parameter). If there is a new data point it goes through the same process and before and the result is returned to the JavaScript page.

[index.php](#)

The index page of the dashboard holds the entire HTML markup for it. While there are many different "pages" on the dashboard (main page, course page, student page, etc...), they act more as views since they are not on different pages. Each "page" is simply a different div in the HTML and the

divs are toggled using JQuery in the `script.js` page. The reason this page is `.php` instead of `.jsp` is because of the session checking for authentication. The rest of the Auto Ed site is built in `php` and they hold session variables to check for properly authenticated users. It made sense to use the same language to make this process easier. In addition the `index.php` page has a number of JavaScript functions at the bottom. All of these simply act as drawing methods to draw the many Google Charts that exist on the dashboard. These functions are all called from the `script.js` page.

`script.js`

The `script.js` page acts as the main JavaScript page for the dashboard. All JavaScript functions excluding the ones in `index.php` and the open source libraries can be found here. This is a massive page of more than 1500 lines of code, but it can be broken down into a few main categories. It loads in all of the data from all of the report tables (student data, course data, daily data and assignment data), parses the comma separated lines, interprets them and then displays them on the page mostly in the form of Google Charts and some occasional tables. It also creates the alerts for each assignment, and for what happened in the past day.

Running the Report Generation

To successfully run the report generation, the four reporting pages must be run every night. These pages are `getAssignmentReport.jsp`, `getStudentInfo.jsp`, `getDailyReport.jsp`, `getCoursesReport.jsp`.

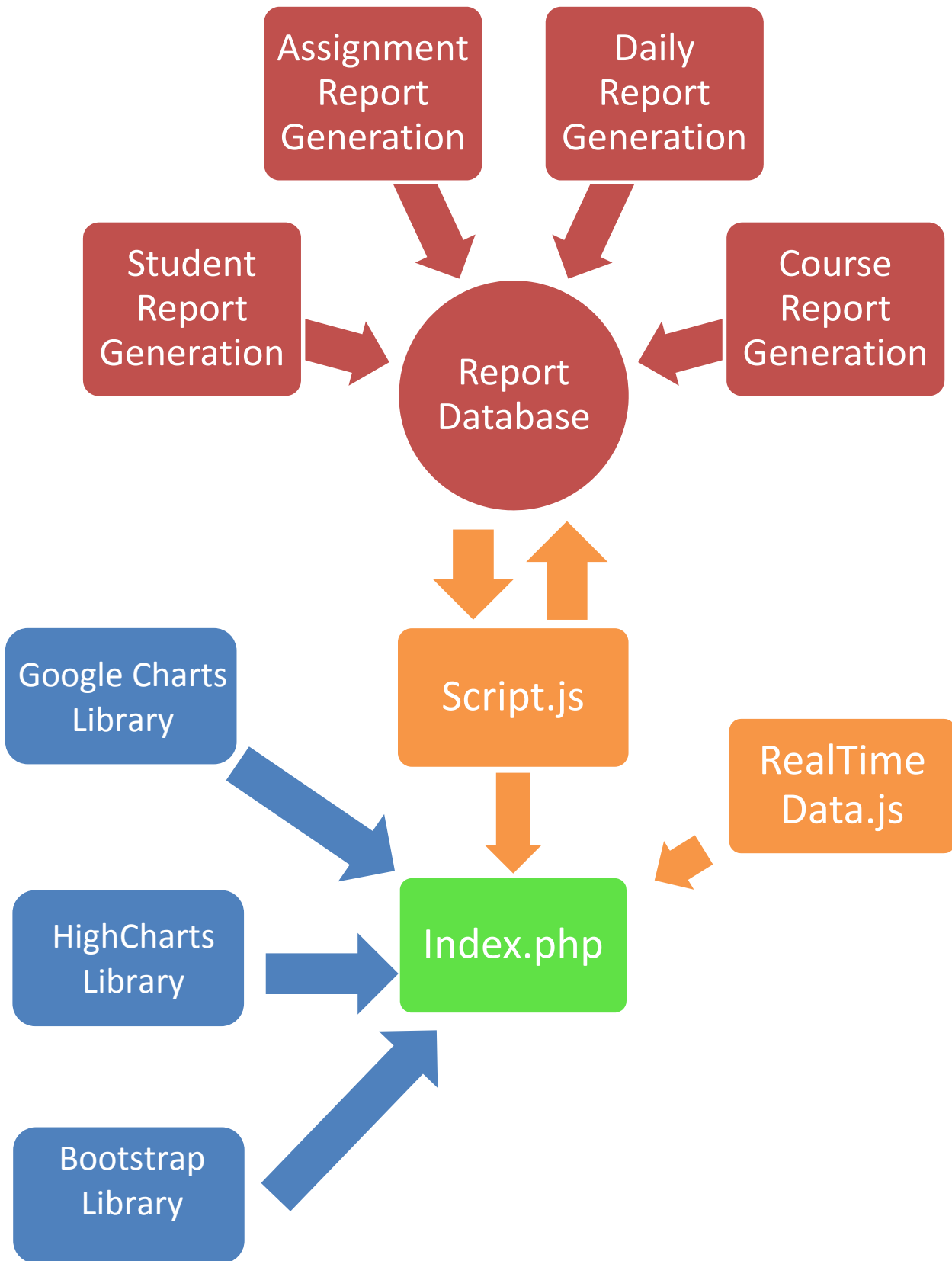


Figure 40. A diagram of the flow of data from the report generation to the dashboard. In red are server side pages. Orange are client side pages built personally for this dashboard, and in blue are open sourced libraries. Green is the dashboard.

Student Answer Parsing

Each student answer that was input into the system had to be parsed and analyzed and placed into compartments. As described above, those compartments were Missing Negative, Wrong Value, Wrong Unit, No Answer and Within 5%. Because student answers are in the form of natural language input (anything that a student wanted to input could be written into answer text box) it was a challenge to properly analyze and compartmentalize them. Therefore a number of different pages were created to do this analysis, some of which are now deprecated. Each of the remaining pages are different in the nature of what answers they analyze, but the actual analysis is the exact same for each individual answer. Focus here will be placed on one page that analyzes individual answers without any preamble about querying for different answers based on students or question parts.

`getAnswersInd.jsp`

This page houses two methods, both of which serve the same purpose but return slightly different values. `getQuestionPartAnswersIndividual()` is a method that returns a concatenated string of 0's and 1's, depending on whether or not this particular answer is incorrect because of Wrong Value, Wrong Units, Within 5%, No Answer, or Missing Negative. Each of those compartments will be returned either as a 1 or a 0. The method calling this method will parse the returned string. To achieve the analyzing, each student answer was broken up using string regular expressions. An answer, 14.5m/s, would be broke up into [14], [.), [5], [m/s] which means that it splits up numbers from characters. Then the method will check to see if there is a negative in the correct answer (which has also been broken up in the same above fashion) and whether or not there is a corresponding negative in the student answer. If no then the Missing Negative count will be set to 1. Else the program moves on. Next a check is done to determine whether or not there is a decimal in the correct answer and in the student answer. Then the student answer's and the correct answer's actual values are put together (if there is a decimal) into one variable. Then the student answer and the correct answer are compared to see if they are the same or not. If not then the Wrong Value counter is changed to 1. If they are not the same, it is determined whether or not the student incorrect answer is within 5%, and if so the within 5% counter is changed to 1. Lastly the string that remains after the actual values are put together into one unit variable. The student answer's unit variable is checked against the correct answer's unit variable. If they differ then the Wrong Units counter is changed to 1. The method then returns a string of those counters. The method `getQuestionPartAnswersIndividualReturnNumber()` does the same process above but

also returns the student answer's actual value in the process for further processing by whatever method that called it.

`equations.jsp`

This page is one of the largest jsp pages in the entire system, and work on it took the longest out of any other areas. Unfortunately due to time constraints, work had to be suspended on this area and has been left for future work. In essence what this page tries to accomplish is to further categorize incorrect student answers that were incorrect due to Wrong Value/Number. In fact what the goal was to make a guess as to what equation the student used to get that incorrect answer. Then once the so called incorrect equation is known, it is possible for a professor to know where a student made their mistake. Also it would be possible to aggregate all of the incorrect equations used to try and find some patterns. Maybe all students were using the same incorrect equation, in which case potentially the question itself is faulty.

One concept that is very important but might be missed in the database is that there are question and question part templates, and instances of those templates. Professors create the templates, and students see the instances. The templates store the random variables that will be evaluated for each student. This makes it so that each student will see a slightly different version of the same question. However, it significantly complicates the process. An instance of a template is seen by the student with the random variables all evaluated to a single value.

The goal of this page is to take the correct equation (that generates the correct answer for all instances) which uses the random variables found in the templates, and to basically manipulate the equation by removing/adding terms, changing operators and negative signs, each time evaluating the new equation formed and seeing if the new equation evaluates to the answer given by the student. To be able to manipulate the correct equation, it is necessary to have each of the variables in the correct equation assigned to an actual value to allow for evaluation of the new equations.

To accomplish this goal, a large amount of string parsing had to be done. The plan was to first find all of the random variables and their ranges in the question and question part templates and then find the numbers that a student saw in the question and question part instances, and then match those up. Therefore giving the program all of the variables in the correct equation and more depending on whether or not extra variables were given to the student. Each question has variables in it that are either randomized, and therefore put inside `<eqn>` tags, or they are just there in plaintext not changing for any students which makes them just numbers. For example a question such as this: "With a mass of `<eqn $f=(1, 4, 1)>` the object slides down a ramp of degree 50" has two numbers in it. The randomized

value of f that is unknown at this point but will go from 1 to 4 with increments of 1, and the set value of 50. Therefore at first the program needs to find all of the variables (in this case the f) and the “other” numbers that are set. For each of the random variables it also needs to store the range. Once all of the variables and numbers were found it was realized that for each Question, there also exists at least one Question Part. Of course the question parts are the things that students answer, therefore that also needs to be properly parsed for the variables and other numbers. Now all of the random variables have been found and stored along with their range. Then the program needs to take the actual question and question part instance that a student saw with the randomized variables actually evaluated to a certain value. So what the program has is for each question part that a student answers, the student will also get relevant information from the question. However, to find what those variables are, the question part template and the question template also need to be analyzed. The program at this point has analyzed and stored all of the variables from the templates, so now it needed to go and find all of the numbers in the question and question part instances that the student actually saw. The difference of course is that while the templates have variables encased in `<eqn>` tags (making them relatively easy to find), the instances just have a constant stream of text that also happens to include numbers. To parse out those numbers was not trivial as they are generally also surrounded by other characters.

However, the work was accomplished and each number from the question instance and the question part instance was found and stored. Then each found number had to be matched to its respective variable by the ranges of the variables. At this point all variables in the templates had finally been matched to the numbers found in the instances. Next came the interesting part of actually manipulating the correct equation. First it was checked whether or not a student answer was correct. If it was, then no further action was taken. If it was incorrect then it was checked what compartment that fell into. If it was a missing negative, then the simplest thing was done whereby the student’s answer was multiplied by minus 1, and checked against the correct answer. If it matched then that means that the student was only missing a negative. However, if it didn’t match then further action needed to be taken.

Unfortunately at this point work had to be discontinued due to technical issues and time constraints. The technical issue was that to evaluate a newly created equation required a large parse tree. This work had already been completed by Dr. Lawrence but it never actually returned the equation, it would only return whether an answer passed to it was correct or not. Therefore it was required to change the code written by Dr. Lawrence which was found to be difficult with the server situation. Therefore work was discontinued and never returned to.

Lastly the code in `equations.jsp` is called by each question in the system and for each question, each question part is evaluated, and for each question part template each student answer for an instance of that template is evaluated. Generally there are a few thousand of these student attempts per question part, therefore running the code on a single question takes a few seconds at least. The code has been tested for all questions in the system, and except for one question instance, the code is able to correctly find all of the variables, match them up to their respective values in the instances and even check whether or not a multiplication by minus one will solve the issue. The one stand out is a question that takes in scientific notation, and support for that type of answer has not been instituted as of yet.