Project Number: JP-0501

Statistical Teaching Aids

An Interactive Qualifying Project Report
submitted to the Faculty
of the
Worcester Polytechnic Institute
in partial fulfillment of the requirements for the
Degree of Bachelor of Science

by
William A. Pfeil

05/04/2006

Professor Joseph D. Petruccelli, Advisor

---

1. Teaching
2. Computers
3. Statistics

Abstract

This Interactive Qualifying Project (IQP) involved creating technological aids to effectively teach elementary statistics principles to students in a self-paced learning environment. The aids created consist of three exercises, which are designed to educate students about the properties of two different regression methods. Each exercise consists of an introductory background reading section, followed by an interactive Java[1] applet with steps to guide the user through various activities, and concludes with some thought-provoking questions to test the students' understanding. The interactive environment incorporates principles derived from the latest research in student learning and computer instruction. The exercises were tested by an undergraduate Worcester Polytechnic Institute (WPI) statistics class and revised in response to test results. The final product has been placed online along with exercises developed by previous project groups that are currently used in elementary statistics courses at WPI.

[1] http://java.sun.com/

Table of Contents

Executive Summary

Most undergraduate students entering a mathematics-related field take at least one introductory statistics course. Many students do not sufficiently learn the material, for many reasons. Some do not see the relevance of the material; others simply have little interest in the subject. Of those that do well in the course, some do not attain a deep understanding of the concepts, but rely on memorization to succeed.

The project's goal was to establish online exercises to convey statistics concepts using Java applets. These exercises will aid statistics students in learning about regression methods. The exercises are designed to encourage *deep learning* – a learning style which leads to a deep understanding of the ideas being taught – by being self-paced, intuitive, fun, and interactive. The project's deliverables were three interactive multi-step instructional Java applets (exercises) about regression methods, with accompanying web pages for background information and online learning evaluation methods.

Every student learns in a different way, but there are general learning theories that may be applied to teach concepts effectively to the majority of the target audience of undergraduate statistics students. Research into methods for encouraging deep learning led to use of the following techniques in the exercises:

- Learning by doing
- Using problem-based learning
- Giving assignments requiring more than just memorization
- Encouraging student reflection
- Allowing for independent learning
- Rewarding understanding

Storyboards were a useful tool in planning the exercises,. Steps for each exercise were laid out before implementing any exercise. During the implementation phase, modularity was a key design principle. The design of the applets was split into three framework components: the visual layout, Cartesian plotting functionality, and data querying methods. The exercises were tested for technical correctness and for learning

effectiveness. Technical correctness was evaluated through a unit-testing methodology, through peer computer scientist evaluation, and through an actual field test with statistics students. Learning effectiveness was evaluated through statistics students' responses to a survey and through analysis of their responses to exercise questions.

The three exercises that resulted from this project cover *Least Squares Regression*, *Least Absolute Deviations Regression,* and *a comparison between the two regression methods, respectively*. Overall, it was noted that the students found the exercises acceptable, and learned the concepts well. As a result of testing and observation several revisions were made to the exercises. These involved small changes in the visual aesthetics, and clarifications of explanations of terms and exercise steps.

In conclusion, we found that the learning theories researched and applied were effective in producing interactive teaching aids that improve upon conventional statistical teaching methods. Planning and testing played key roles in delivery of the project on time without any technical setbacks. The exercises have been released for use in the public domain.[2] An unanticipated result of the research conducted for this project was the author's creation of an entry on least absolute deviations regression in the online encyclopedia *Wikipedia*.[3]

[2] http://www.math.wpi.edu/Course_Materials/SAS/lablets/7.3/73_choices.html
[3] http://en.wikipedia.org/wiki/Least_absolute_deviations

1. Introduction

Instructors have long noted the difficulties many students experience in trying to learn the material in introductory statistics courses. Many students have little interest in the subject and/or do not see the relevance of the material. Others do not think highly of the conventional "lecturer & whiteboard" teaching medium. Often, even students who succeed in the course will not retain what they have learned, since they have only memorized the material.

Research suggests that self-paced, interactive teaching materials can help improve student learning. *[Hartley, 1998, Burgess and Strong, 2003]* As this research explains, interactive learning encourages what is called *deep processing*, which helps students obtain a deeper understanding of underlying concepts.

Within recent years, technological innovations have spurred many new teaching methods. The gradual acceptance of the Internet, specifically, has opened up new possibilities for interactive learning. Java applets have harnessed this interactive power by providing a simple Application Programming Interface (API) to create graphical interfaces, easily embedded into Hypertext Markup Language (HTML) web documents.

Worcester Polytechnic Institute (WPI) hosts a set of statistical teaching aids online, developed through Interactive Qualifying Projects such as this one. These teaching aids consist of Java applets embedded in a laboratory evironment containing explanations, directions and questions. The purpose of the applets is to demonstrate statistical concepts through guided interactive student exploration. At the beginning of this project there were 14 labs and their associated applets in use. These had been created and tested over a six-year period. In this project, we added to this learning environment by creating and testing three additional applets and their associated supporting materials.

The goal of this project was to create exercises to convey statistics concepts using Java applets. The exercises consist of a section with a short introduction to a given topic followed by an interactive applet with steps to guide the user through activities, concluding with some thought-provoking questions to test the students' understanding. The interactive online environment makes learning statistics more intuitive and simple

than if presented statically in a textbook or lecture. The final outcome of the project was tested by a WPI statistics class and revised as necessary to be effective and error-free.

2. Background

The goal of this project was to create exercises using Java applets to convey statistics concepts, extending work completed in previous WPI IQPs *[Bellmore et al., 2005, Clein and Holmes, 2003, Gottreu and Slater, 2003, Kawato, 2003, and especially Lieser and Whitford, 2001]*. Through careful planning and research, this IQP developed tools to encourage students to learn and retain concepts by experimentation and hands-on activity. A classic proverb portrays a key advantage to interactive learning methods: *"Tell me and I will forget, show me and I may not remember, involve me and I will understand."*

2.1. Theories of Learning

Besides technological requirements, an educational software designer must consider many factors to produce a successful design. The most important consideration is how students learn. Students who *memorize* material as a series of facts retain information only short-term, whereas *understanding* leads to long-term retention of concepts. For example, instead of stating a plain fact such as "Cheetahs are the fastest of land animals," a more effective approach might give a context and reason such as in the statement "Cheetahs depend on their speed for survival, and have adapted to become the fastest land animal in order to catch their prey." This example illustrates one strategy of applying a fact to a situation to encourage *deep learning* behavior. There is much prior research on learning patterns that we considered in order to develop educational tools to more effectively help students learn concepts.

2.1.1. Existing Literature

Understanding the way people learn is critical when designing learning aids. Research by cognitive scientists and neuroscientists helps in understanding learning processes.

James Hartley, research professor at the University of Keele, explains research on two different types of student learning strategies, *deep learning* and *surface learning [Hartley, 1998]*. As the names suggest, using a deep learning strategy involves searching

for key points, looking at the intentions of an instructor or author, and aiming for a general deep understanding, while using a surface learning strategy involves, for example, rushed reading and mass remembrance of all facts. When reading, *deep processors*, or those who use a deep learning strategy, will look at the meaning behind words, and surface processors will remember words only. Research indicates that the learning strategy has a profound effect on how well material is learned and retained. In an experiment, *[Marton and Saljo, 1976]* groups of students were given a passage to read. These students were classified into deep and surface processors, based upon questioning as to how they had read the passage. The results showed that all of the deep processors remembered the main points of the passage, and none of the surface processors summarized the main points adequately. In the 1990s, research began to find which factors may influence learners to be surface processors, and which might encourage learners to be deep processors. The book *Improving the Quality of Student Learning, [Gibbs, 1992]* lists the following factors thought to promote surface and deep processing in students.

Surface Processing Factors

- Heavy workloads
- High class contact hours
- Lack of opportunity to explore subjects in depth
- Lack of choice over subjects and methods of study
- An anxiety-provoking assessment system

Deep Processing Factors

- Project work
- Learning by doing
- Using problem-based learning
- Giving assignments requiring more than just memorization
- Giving group assignments
- Encouraging student reflection
- Allowing for independent learning

- Providing tasks that are not busywork, but real problems
- Rewarding understanding and penalizing reproduction
- Involving students in the choice of assessment methods

Visual aids are helpful in increasing retention of information. Visualizations are more commonly retained, in general, than are written materials.[4] Studies show retention of information, as well as quicker acquisition of knowledge, is increased when visual aids are used during presentations.[5] Clark and Mayer *[Clark, 2003]* state that presentations with text and visuals encourage students to actively participate in learning by making connections between the words and pictures. Listed below are ten reasons to use visual aids in presentations, *[Pike, 1994]* all of which benefit the learner.

Reasons to Use Visual Aids in Presentations

- To attract and maintain attention
- To reinforce main ideas
- To illustrate and support the spoken word
- To minimize misunderstanding
- To increase retention
- To add realism
- To save time and money by efficiently helping to improve communication
- To aid in organizing thoughts
- To ensure that key concepts are covered
- To aid instructors in delivering high quality presentations

These reasons apply to any learning environment, whether online or offline.

Verbal or written encouragement plays a key role in learning for those who are *extrinsically motivated*. Extrinsically motivated students complete an exercise to attain course credit or similar extrinsic reward. For these students, repeated praise may encourage deep processing by helping them recognize their progress in some way or

[4] Roediger, H. "Memory: Explicit and Implicit"
[5] http://www.soe.umd.umich.edu/maaipt/research/Arnold-Larkin.pdf.pdf

another[6] and helping them reflect on what they have done. Too much praise can actually decrease self-motivation, as praise or rewards for undeserving effort can become insincere.[7] It is safe to assume that most students will use the technological learning aids created in this project only to satisfy an academic requirement. Therefore the aids target extrinsically motivated students, leading them though exercises, and do not require a great amount of self-motivation. The exercises in this project give praise after every few short accomplishments. "Give yourself a pat on the back" is a canonical example of such encouragement.

Exciting tasks result in a higher "hedonic tone," or pleasurable state of learning. Hebb and Apter present classic models relating relaxation, anxiety, boredom, and excitement to motivation. *[Hebb, 1955, Apter, 1985]* The models state the obvious: minimizing boredom and anxiety, or conversely, maximizing excitement and relaxation, result in an optimal attitude for learning.

2.1.2. Relevance of Learning Theories to the Project

The exercises for this project encourage deep processing and discourage shallow processing in a few ways, as outlined in the list by Gibbs *[Gibbs, 1992]*. First and foremost, students can learn by doing instead of by listening. The students also have the opportunity to experiment endlessly if they desire. This gives them the chance to explore the subject in depth. They are not forced to the next step; they may interact with the applet for as long as they wish or go back to any previous step. During the exercise, informal questions are presented for the students to think about. These are meant to help the student learn as he/she proceeds through the exercise, as opposed to allowing them to quickly run through the required steps. Finally, after completing each exercise, a set of questions is provided to assess student understanding. The exercise questions are thought-provoking and require understanding, not just memorization. The questions are based largely on the informal "*To think about*" questions that are presented to the students during each exercise. Surveys were also given to students, and responses were used to

[6] http://www.nwrel.org/request/oct00/textonly.html
[7] Brooks, S.R., Freiburger, S.M., & Grotheer, D.R. (1998). *Improving elementary student engagement in the learning process through integrated thematic instruction.*

improve or fix any shortcomings in the applets and to valuate the effectiveness of the learning aids.

Visualizations accompanied by text are important aids in understanding and long-term retention of concepts. The interactive applet serves this visual requirement well. The exercises are designed to minimize anxiety and boredom in order to put students in the optimal state of mind for learning. The self-paced strategy and selection of an appropriate level of difficulty help reduce student anxiety, and the use of real-world applications in the examples and lively animation and colors in the displays help prevent boredom.

Written encouragement helps students feel that they are learning. Even simple phrases like "Whew!" following an exercise help validate the student's work and generate an enjoyable feeling about a job accomplished, promoting relaxation and deep processing.

2.2. Technological Learning Aids

Teaching aids have existed for countless years; computers are just the latest addition to the many aids. Computers are now commonplace in learning environments. James Hartley *[Hartley, 1998]* states that supporters of new technology argue that computer-aided learning (CAL) has the following advantages over classroom learning.

- Learning is individualized
- Learning is self-paced
- There can be instant feedback upon responding
- The programs have been written by experts, and tried and tested before being published
- The whole procedure can be cost-effective

While CAL is relatively new compared to traditional teaching methods, there is still much prior art. Many have quickly taken advantage of the new versatile medium.

2.2.1. Prior Art

This project is an addition to work completed by groups of WPI students over the past five years. Their accomplishments are summarized in Table 1. The results of their work may be found online.[8] From evaluation of their work and student feedback, this project builds upon their experience.

IQP Group	Accomplishments
Bellmore, Jarrod Thomas Blaquiere, David Lee Lewis, Adam LaFond Rahman, Evgeny	Lab 1.1, Lab 3.1, Lab 3.2, Lab 4.5, Lab 4.6, Lab 5.3 *This group made applets which introduced the following concepts: sample variation, simple and stratified random samples, paired comparison design, probability, population, the Central Limit Theorem, estimation, prediction, and tolerance.*
Kawato, Takeshi	Lab 4.1, Lab 4.2, Lab 4.3 *This student made applets to teach the Bernoulli distribution model, the binomial distribution model, and the Central Limit Theorem.*
Gottreu, Brian Phillip Slater, Jeremy Aaron	Lab 2.2, Lab 2.3 *This group created applets to teach about power transformations and stationary processes.*
Clein, Robert Haiman Holmes, Samuel Benjamin	Lab 4.5, Lab 4.6, Lab 5.3 (CGI) *This group implemented labs 4.5, 4.6, and 5.3 originally, using the Common Gateway Interface (CGI). These labs were later re-programmed in Java by Bellmore, Blaquiere, Lewis, and Rahman.*
Lieser, Eric Dale Whitford, Paul Charles	Lab 2.1, Lab 7.1, Lab 7.2 *This group created applets to teach about the resistance of the mean, median, quartiles, and standard deviation summary statistics, the method of least squares, and power transformations.*

Table 1: WPI Statistical Teaching Aid IQPs' Accomplishments

[8] http://www.math.wpi.edu/Course_Materials/SAS/lablets/statlab.html

The first task of this project was to improve and extend an existing least squares regression lab exercise created by two WPI students in a previous IQP. *[Lieser and Whitford, 2001]* The *least squares* regression exercise (Lab 7.1 on the WPI web-based Statistics Labs site[9]) was developed as a tool to allow students to investigate the concept of least squares regression. A screenshot of this applet is shown in Figure 1. This lab allows a student to dynamically adjust a line to try to approximate a least squares fit for a given set of points. The slope and intercept of the line, the sum of squared errors (SSE), and a residual plot are dynamically updated during experimentation, and provide feedback as the student tries to obtain the least squares line.

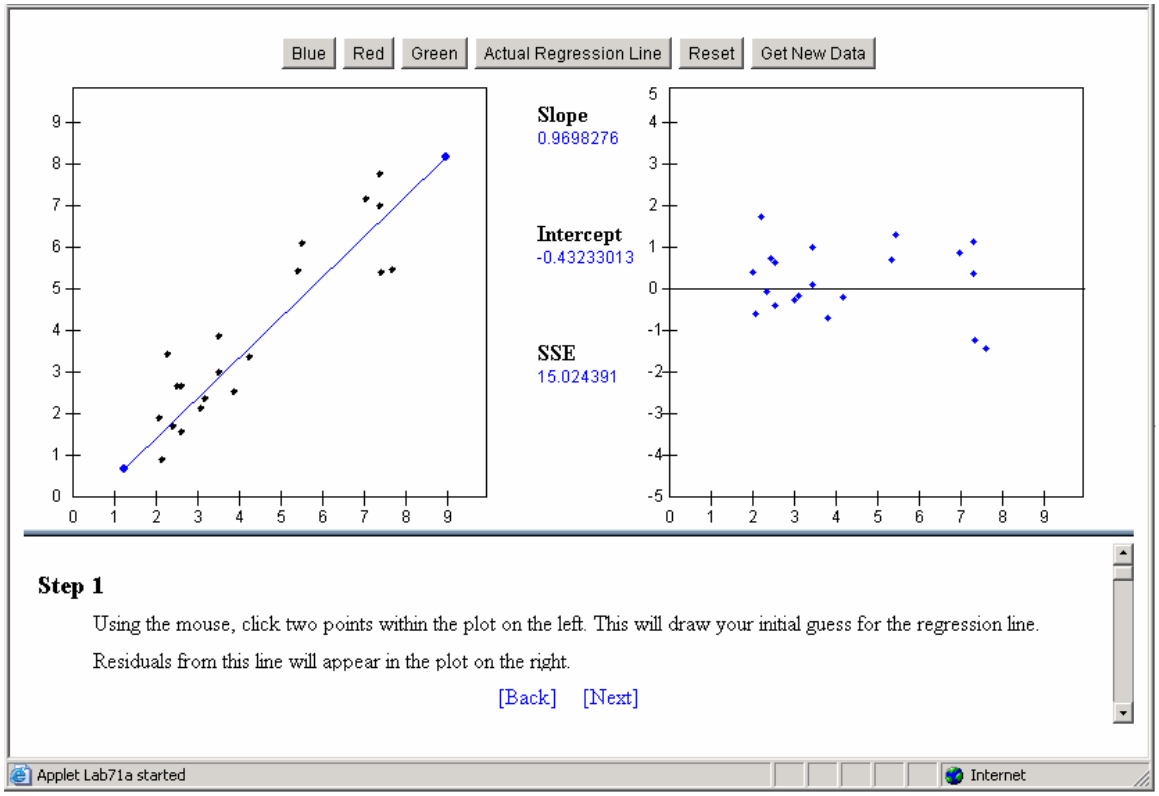


Figure 1: Previous IQP's applet for Lab 7.1 – the method of least squares

The lab has been be improved by an added graphical representation of the least squares fitting procedure. Specifically, the individual squared errors are displayed on the plot as growing/shrinking squares, along with one square that changes in size according

[9] http://www.math.wpi.edu/Course_Materials/SAS/lablets/statlab.html

to the sum of squared errors. An example of these features is shown by an applet for least squares on the Key Curriculum Press's JavaSketchpad site.[10] The lab has been extended by:

- Making a parallel set of plots to illustrate fitting by *least absolute deviations*, a more statistically robust fitting method
- Creating plots to compare the two fitting methods (*least squares* and *least absolute deviations)*
- Giving the students the ability to move data points in some cases, as opposed to only the regression line
- Supplying sequential steps and questions to help students realize the differences between the fitting methods

Many other educational applets exist on the web, including statistical teaching applets. Duke University hosts a collection of statistics applets.[11] The VESTAC project[12] (Visualization of and Experimentation with STAtistical Concepts) provides applets to teach many key statistics concepts. California State University hosts a collection[13] created by Charles Stanton. Virginia Tech's department of statistics hosts "Statistical Java,"[14] a site with many applets. The Rice Virtual Lab in Statistics[15] has a similar collection, and many more exist. Over 600 more sites are linked on the Interactive Statistical Calculation Pages[16] that have online statistical functions (not necessarily Java-based).

The reason technological learning aids are so popular is clear: they are self-paced, involve user interaction, are very accessible and intuitive, can have visual and auditory learning aids, give instant responses, allow for online discussion of material, and have relatively cheap production costs when compared to other methods of teaching. Although some of these advantages are also available in a classroom, computer-aided learning has the general advantage of allowing students the freedom to experiment with ideas that

[10] http://www.keypress.com/sketchpad/javasketchpad/gallery/pages/least_squares.php
[11] http://www.isds.duke.edu/sites/java.html
[12] http://www.kuleuven.ac.be/ucs/java
[13] http://www.math.csusb.edu/faculty/stanton/m262/
[14] http://kitchen.stat.vt.edu/~sundar/java/applets/
[15] http://www.ruf.rice.edu/~lane/rvls.html
[16] http://members.aol.com/johnp71/javastat.html

cannot or might not be brought up by a student in a classroom because of embarrassment, time constraints, or the fact that questions might not arise until students attempt to do exercises *for themselves*. The National Academic Press article "Technology to Support Learning"[17] explains many more advantages.

2.2.2. Technology Requirements

As mentioned, Java applets are used as the medium for delivering interactive lessons. The J2SE 1.4.2 specification was used for reference.[18] At the time of writing, this is one major revision prior to the latest J2SE version. This specification was used for compatibility reasons. The abstract high-level Java class "Graphics" was used to render images, text, and HTML panes. There is much prior source code available to applet creators, which helped keep programming from becoming the design bottleneck. Rather, the author could focus largely on the educational content of the exercises. Creating applets is an art that has been mastered by many, and various resources are available on both applets in general and the Graphics class. The Java specification itself is relatively easy to follow; coupled with examples and lessons from sites such as JavaWorld[19] and JavaReference.com[20], all programming resource requirements were satisfied.

On the user side, since Java programs have multi-platform capability built in, all that is required for a statistics student is a computer with Internet access and Java installed.

2.3. Least Squares Regression

Least squares regression is one method used to fit a line to a set of (x,y) data. It is by far the most popular regression method used, partly due to its computational simplicity. The term "least squares regression" is sometimes used synonymously with "linear regression" or just simply "regression." Conceptually, the least squares regression line is the line that minimizes the sum of the squares of the vertical deviations from the line to

[17] http://www.nap.edu/html/howpeople1/ch9.html
[18] http://java.sun.com/j2se/1.4.2/docs/api/
[19] http://www.javaworld.com/
[20] http://www.javareference.com/

each of the data points. Computationally, there are numerous methods of finding the least squares line.

For comparison, another regression technique is least absolute deviations regression. Conceptually, a least absolute deviations regression line is the line that minimizes the sum of the absolute values of the vertical deviations from the line to each of the data points. Computationally, a least absolute deviations line is more difficult to find than a least squares line, though there are numerous methods of finding the least absolute deviations line.

Although not as popular as least squares, least absolute deviations has some advantages as well. Table 2 outlines some of the most important differences between the two methods.

Least Squares	**Least Absolute Deviations**
Not very robust	*Robust*
Always one solution	*Possibly multiple solutions*
"Stable" solution	*Possibly "unstable" solution*

Table 2: Differences between least squares and least absolute deviations regression

The motivation for including least absolute deviations as an alternative to least squares in this project was its superior robustness. In fact, at the start of this project, we were unaware of the issues with instability and multiple solutions. The latter properties were discovered through research and experimentation, and were incorporated into the exercises we developed. The three properties require some explanation.

Least absolute deviations regression is *robust* in that outliers do not have a large affect on the regression line. In contrast, a least squares line will be affected by all data points. This property is apparent when dragging data points vertically in Exercise 7.3c. As an outlier is created by dragging a point up or down, the least absolute deviations line sometimes won't move *at all*. However, the least squares line always adjusts itself when any data point is moved vertically. The SSE is also affected more drastically by outliers than is the Sum of Absolute Errors (SAE) used with least absolute deviations because the SAE changes according to the *square* of the residuals.

For a given data set, least squares always produces only *one solution* (the regression line is unique). In some cases, least absolute deviations produces multiple solutions. The least absolute deviations properties are explored in exercises 7.3b and 7.3c and will be explained in further detail in section 2.4.

When we say that the least squares solution is *stable*, we mean that, as any data point is moved horizontally, the least squares regression line moves continuously. The least absolute deviations line, however, may jump a large amount for a small horizontal adjustment of a data point. The reason is that (1) the fitting algorithm used only finds one solution, and (2) as one of the data points is moved horizontally from a value at which there is a single LAD solution, across a value which allows multiple solutions to another value having a single solution, the displayed line will jump from one unique solution to the next across the multiple-solutions region. This behavior becomes apparent when dragging data points horizontally in Exercise 7.3c.

2.3.1. Solving Methods

There are many methods of calculating the least squares regression line. For most uses, the simplest method of finding the least squares regression line will suffice. The simplest method is an analytical solving method used in this project. However, in some cases, where solution accuracy is very important, one must consider *numerical stability* issues when choosing an algorithm. Two algorithms may produce algebraically equivalent solutions, but in running the algorithms using finite amounts of precision in the data types, actual solutions may differ. An algorithm that is numerically stable will be more likely to produce a solution that is closer to the (correct) algebraic solution. Since finite-precision data types are used for all computations, there is always a small amount of error in the data representation. The key to achieving numerically stable algorithms is to find *robust algorithms*; that is, using *infinite-precision data types* is obviously not a feasible way to solve numerical instability problems. A simple example where numerical instability may arise is during divisions. If the divisor of an operation is a variable, and there is a possibility that the variable could be zero, this is clearly a problem. However, if the divisor has a possibility of being a very small number with respect to the numerator and/or may be comparable in size to the levels of precision used in the data type,

numerical instability issues arise. That is, the error in the divisor may drastically affect the quotient. Numerical instability is a large topic, and this is only an example. Kincaid *[2003]* provides more in-depth information about numerical methods.

Besides analytical least squares solving methods, there are also algorithms to calculate the least squares regression line using numerical (or iterative) methods. These methods are generally more difficult to understand and implement. However, these methods were not necessary for this project. To learn more about these methods, see *[Bjorck, 1996]*.

2.3.2. Chosen Solving Method

As stated earlier, an analytical method was used to solve for the least squares regression line. The method of least squares has the nice property that its solution may be computed analytically; not all methods have this desirable property. For instance, the least absolute deviations regression line *cannot* be computed analytically.

The goal of least squares is to find the line of best fit (where "best" is determined by criteria mentioned previously) defined by the slope, *m*, and the y-intercept, *b*, in the following equation of a line in 2D space: $y = m \times x + b$. The full derivation of the regression line formulas for *m* and *b* may be found in *[Wolfram]*. Since the solution is derived analytically, it requires few computations, and no iterations or repeated guesses:

$$b = \frac{SS_{xy}}{SS_{xx}}$$

and

$$m = \bar{y} - b \times \bar{x},$$

where

$$SS_{xx} = \sum_{i=1}^{N} (x_i - \bar{x})^2 \quad \textbf{and} \quad SS_{xy} = \sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})$$

and

$\bar{x}$ **= the mean of all x values,** $\bar{y}$ **= the mean of all y values,** N **= the # of points**

This method is not as numerically stable as some methods, nor is it the fastest method. However, it serves the purpose for the applets in this project for three reasons: *(1)* the algorithm was simple to implement, *(2)* it was numerically stable enough to produce sufficiently accurate results for our purposes and *(3)* it was fast enough to update the least squares regression line in real-time in the case of at least 5 data points. To elaborate on numerical stability, the solution is only numerically unstable when $SS_{xx} = 0$, which is the case when all of the data points have the same *x* value. In this case the slope and intercept are rightfully undefined. If SS_{xx} is very close to 0, but not equal to 0, numerical stability could theoretically be a problem. However floating-point precision is precise enough to produce a valid answer for all cases allowed in this exercise. The slope and intercept only blow up if all points have the same *x* value. The third requirement, an algorithm that achieves real-time updates, was necessary in order to allow the student to drag points while updating the displayed regression line immediately whenever any point was moved to a new location. To avoid any potential real-time screen refresh problems, the number of data points was restricted to a maximum of 5 for all exercises

2.4. Least Absolute Deviations Regression

As shown in Table 2, we have observed three interesting properties of least absolute deviations regression: robustness, instability, and non-uniqueness of solutions. These properties are in direct contrast with the least squares regression method. One interesting fact that may help to explain a couple of the least absolute deviations properties is that if the least absolute deviations line is unique (i.e., there are not multiple solutions), the solution *must* pass through at least two of the data points. Since in most cases there is only one solution, the least absolute deviations line usually "latches" onto two data points, and the line can exhibit non-smooth changes as data points are smoothly changed in the horizontal direction. For example, the line may appear to "jump" from latching onto points 1 and 2 to latching onto points 2 and 3 as point 1 is moved horizontally. As explained in section 2.3, we call this the "unstable" property of least absolute deviations regression.

The fact that a solution latches onto two points also explains the "robust" property in the following way. If there exists an outlier, and a least absolute deviations line must latch onto two data points, there is no way that the outlier will be one of those two points because that will not minimize the sum of absolute deviations if the point is extreme. Moving the outlier to an even more extreme position will have no effect on the LAD line, since the only choice for a new "latch" point would be the outlier, which can only increase the SAE.

The "latching" is sure to take place only if the least absolute deviations solution is unique. From our observations, when there are multiple solutions, some of the solutions will pass exactly through one point or through no points at all. It is not completely clear to us what determines the cases in which there are multiple solutions. However, from what we have observed, it appears that the region that encompasses the multiple least absolute deviations lines is one continuous region and is bounded by at least two lines that have the same sum of absolute errors. Based on this observation, we developed an algorithm to find and show multiple solutions if they exist.

We can show that certain sets of points symmetric about a horizontal line will produce a region bounded by two least absolute deviations lines with the same sum of absolute errors. One such set of points is shown in Figure 2. This example is demonstrated to students in Exercise 7.3b.

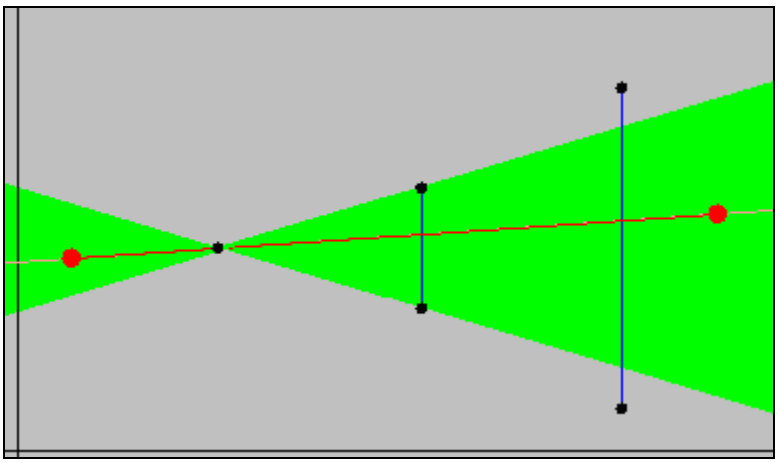


Figure 2: A set of data points with reflection symmetry and multiple least absolute deviations solutions. The "solution area" is shaded in green. The red line shown is a

user-adjustable line used in Exercise 7.3b. The vertical blue lines represent the absolute errors from the red line to each data point.

To understand why there are multiple solutions in this case, consider the red line in the green region. Its sum of absolute errors is some value *S*. If one were to tilt the line upward slightly, while still keeping it within the green region, the sum of errors will still be *S*. It will not change because the distance from each point to the line grows on one side of the line, while the distance to each point on the opposite side of the line diminishes by the same amount. Thus the sum of absolute errors remains the same. This argument also shows that there are infinitely many least absolute deviations lines.

2.4.1. Solving Methods

Though the idea of least absolute deviations regression is just as straightforward as that of least squares regression, the least absolute deviations line is not as simple to compute. Unlike least squares regression, least absolute deviations regression does not have an analytical solving method. Therefore, an iterative approach is required. Literature on least absolute deviations was difficult to find and actual algorithms were even harder to come by, besides some complex examples written in Fortran. Nevertheless, many methods were eventually discovered thanks to various resources (see section 5.2. Acknowledgements).

Table 3 shows a list of least absolute deviations solving methods considered for this project, and comments. The algorithms are then explained in detail.

Algorithm	Comments
Simplex-based methods	The preferred method, but has a complex implementation, requiring linear programming knowledge.
Iteratively Re-weighted Least Squares	Has an instability problem that is not easy to overcome.
Incrementally adjust m and b until convergence	Does not always find the global minimum.
Basic iterative approach discussed in [Li and Arce, 2003]	Does not always find the global minimum.
Wesolowsky's direct descent method [Wesolowsky, 1981]	The chosen method – fast and simple.

Li's new proposed algorithm [Li and Arce, 2003]	Another viable method, though not as simple as Wesolowsky's.
Check all combinations of point-to-point lines	Simplest method, although is slow for a large number of data points. Used for finding multiple solutions.

Table 3: Least absolute deviations solving algorithm summary

Most of these methods start with the least squares solution for the slope *m* and the intercept *b* and make successive guesses to reach the solution. Each method has its advantages and disadvantages. To evaluate correctness of the implementation of these methods, the slope, intercept, and sum of absolute errors produced by the algorithm was compared against the same values produced by a Least Absolute Deviations IMSL Fortran routine, which uses a simplex-based solving method.

Simplex-based methods are the "preferred" way to solve the least absolute deviations problem. A simplex method is a method for solving a problem in linear programming. The most popular algorithm is the *Barrodale-Roberts* modified simplex algorithm. This algorithm would have been the top choice for this project if it were not for the steep learning curve (understanding the algorithm requires background in the field of linear programming), and the fact that the simpler method we implemented performed equally well for our applications

Another possible solving method considered was *Iteratively Re-Weighted Least Squares* (*IRLS*). This method did not always produce the correct solution due to numerical instability issues. The complete algorithm is given in Appendix section A.1. The important point in the algorithm to note is that in equation (1), the weight $w_i = \frac{1}{|Y_i - B_{0(n-1)} - B_{1(n-1)} X_i|}$ will be undefined when the denominator is zero (i.e. when the sum of absolute errors is zero) and will be very large when the sum of absolute errors is small. One potential solution to this problem is, for the point(s) that cause the sum of absolute errors to be less than some threshold value (around 0.0001), the sum of absolute errors should be fixed to that threshold value for the current iteration. In other words:

$$w_i = \frac{1}{\max(0.0001, |\text{sum of absolute errors}|)}$$

However, this adjustment did not help the IRLS method produce the line of least absolute deviations in all cases. Thus, this method was discarded.

It was then decided to try an original solving method that was intuitively simple. The algorithm would start with the least squares solution for slope *m* and intercept *b*. Then, it would essentially perturb *m* and *b* in either direction until a minimum was found. The algorithm is outlined in Appendix section A.2. When this algorithm did find the global minimum, the line was exactly correct. That is, the application that was used to compare solutions produced a line with the same slope and intercept. However, the problem with this algorithm is it did not *always* find the *global* minimum, or the line of *least* absolute deviations, for all cases.

Another basic iterative approach to finding the least absolute deviations line is discussed in *[Li and Arce, 2003]*. Though it is stated that this method does not always find the global minimum, we were interested to see if the algorithm's failure was rare enough to be disregarded. The algorithm is outlined in Appendix section A.3, extracted from *[Li and Arce, 2003]*. After implementing this algorithm, it was found that it did not find the global minimum in numerous cases. Thus, this algorithm was also discarded.

Wesolowsky's direct descent method *[Wesolowsky, 1981]* is another least absolute deviations solving method. It is an efficient algorithm and is also simple to implement. Wesolowsky's algorithm was the method chosen for this project. The algorithm is summarized in *[Li and Arce, 2003]* and is reproduced in Appendix section A.4.

A new proposed algorithm discussed in *[Li and Arce, 2003]* by Li is very similar to Wesolowsky's algorithm. Instead of updating intercept values using a weighted median, it updates slope values. This algorithm is also slightly faster than Wesolowsky's, according to Li. The algorithm gains efficiency by transforming coordinates to make better guesses. The algorithm is given in Appendix section A.5.

The final and most intuitive method devised to find the line of least absolute deviations arrived from the discovery of the fact that at least one least absolute deviations line solution must fall on at least two points. Using this information, one can find the equation of the line determined by any two points and calculate the sum of absolute errors; the line with the smallest sum of errors is a line of least absolute deviations.

Although this algorithm is very slow for a large number of data points, it works very well for the small number of data points (5) used throughout the exercises. Although this algorithm was not used for finding the least absolute deviations line, this algorithm was used to identify multiple solutions, as explained in section 2.4.2. Chosen Solving Method.

2.4.2. Chosen Solving Method

As stated earlier, Wesolowsky's direct descent method *[Wesolowsky, 1981]* is an efficient algorithm that also has a simple implementation. The algorithm always produced correct solutions quickly. For these reasons (which are essentially the same criteria as for the least squares solving method), Wesolowsky's algorithm was the chosen method for this project.

Because it always finds a single solution, Wesolowsky's algorithm could not be used for finding multiple least absolute deviations solutions when they existed. However, as stated earlier, one can find the equation of each line falling between any two points and calculate the sum of absolute errors for each line. If two or more lines have the same sum of absolute errors, then those lines define the bounds of the region of multiple solutions. This is how the green shaded region of multiple solutions was calculated in Figure 2. Since this algorithm can be slow depending on the number of data points used, the check for multiple solutions can be easily enabled or disabled from within the source code.

3. Methodology

Following a proper methodology was crucial in successful completion of this project. In creating the exercises, the logical steps of planning, implementation, and testing were followed.

3.1. Planning

From research on learning theories, the following were kept in mind during the planning stage: promoting deep learning, using visual aids, giving written encouragement, and minimizing anxiety and boringness. The first step taken in planning was to clearly define the educational objectives for each exercise, and indicate how they would be met. After this step, the general web site layout was planned. Finally, prior to beginning any programming, storyboards were laid out to portray how the steps of each exercise should proceed. The exercises consist of a progression of pages and activities. Storyboards allow the designer to foresee caveats, anomalies, or simply awkward layouts before they arise during implementation. Storyboards were made prior to coding any exercise in this project.

The educational objectives for each exercise are outlined in the nested lists in Figures 3 - 5. The first level in each list outlines the major objectives of each exercise. The second level explains how each objective is accomplished. Each exercise ends with a set of questions designed to be included in the exercise to test student understanding.

***Exercise 7.3a*: Method of Least Squares**

- **Convey the basic concept of the method of least squares: a way to find the best fitting line to a set of points by minimizing the sum of the squares of the vertical deviations of the points from the line.**
  How:
  - **Give a textual description of the idea. Explain that the squares on the plot are a visualization of the squared residual of each point from the line, and the large square is a visualization of the sum of those errors (the SSE).**
  - **Allow for free play with the graph (allow them to move the line freely to where they think it is the "best fit" by dragging its endpoints). They will be asked to observe the graphical squares and sum of squares grow/shrink.**
  - **"To think about: How is moving the regression line changing the SSE?"**
- **Develop an intuition (deep learning) about the least squares regression line and how data points affect it, through structured activity.**
  How:
  - **Show them a point that is the (mean of x, mean of y). Let them know that the regression line passes through this point.**
  - **Let them guess a line that they believe to be the regression line.**
  - **Show them the actual line.**

Questions to answer afterwards:

- **What is the SSE? How does it relate to the squares drawn from each point?**
- **How did you use the SSE to guess a regression line?**
- **Why was the point (mean of x, mean of y) helpful to know when trying to find the regression line?**

Figure 3: Educational objectives for Exercise 7.3a.

***Exercise 7.3b*: Method of Least Absolute Deviations**

- **Convey the basic concept of the method of least absolute deviations: a way to find the best fitting line to a set of points by minimizing the sum of the vertical deviations of the points from the line.**
  - **How:**
    - **Give a textual description of the idea. Explain that the lines on the plot drawn from each point are a visualization of the residual of each point from the line. The length of each line is the absolute deviation for each corresponding point. The large bar is a visualization of the sum of those absolute deviations.**
    - **Allow for free play with the graph (allow them to move the line freely to where they think it is the "best fit" by dragging its endpoints). They will be asked to observe how the graphical deviation lines and sum of deviations bar grow/shrink.**
    - **"To think about: How is moving the line changing the sum of absolute deviations?"**
- **Develop an intuition (deep learning) about the least absolute deviations regression line and how data points affect it, through structured activity.**
  - **How:**
    - **Let them guess a line that they believe to be the regression line.**
    - **Show them the actual line.**
- **Introduce the fact that some sets of points may have more than one valid least absolute deviations solution.**
  - **How:**
    - **Give them a data set known to have multiple solutions**
    - **Show them one solution, and then show the region of multiple solutions.**
    - **Allow them to move the line around within the region of multiple solutions, so that they may see that it does not change the SAE.**

Questions to answer afterwards:

- **How did you use the sum of absolute deviations to guess a regression line?**
- **From what you've experienced, you know there are some data sets that have more than one least absolute deviations solution. Do you think that for these cases there are infinitely many solutions? Or not? Or depends?**
- **You've seen that it's possible to have more than one valid least absolute deviations line. Do you think that the "non-uniqueness" property is a good thing or a bad thing? (Hint: see the *Introduction* for this exercise)**

Figure 4: Educational objectives for Exercise 7.3b.

***Exercise 7.3c*: Least Squares Fitting Method versus Least Absolute Deviations Fitting Method**

- **Define Least Squares Fitting Method: a way to find the best fitting line to a set of points by minimizing the sum of the squares of the vertical deviations of the points from the line.**
- **Define Least Absolute Deviations Fitting Method: a way to find the best fitting line to a set of points by minimizing the sum of the vertical deviations of the points from the line.**
- **Compare the two methods. Let them try to figure out their basic differences.**
  How:
  - **Show side-by-side plots of the same data, and explain that they are the same data points. One display will have a least squares regression line fitted to the data, and the other will have a least absolute deviations regression line fitted to the data.**
  - **Allow for free play with the *data points* (both graphs' data points will be updated).**
- **Develop an intuition (deep learning) about the differences, through structured activity.**
  How:
  - **Instruct them to drag points horizontally, and ask how it affects each regression line's slope differently (show how much each line jumps around). This shows the "instability" property of LAD.**
  - **Instruct them to drag points vertically (to create outliers), and ask how it affects each regression line's slope differently (show how resistant each line is to outliers). This shows the "robustness" property of LS.**
  - **Ask them to drag a point to the left of the mean of X approximately 200 units vertically upward. Record the slope before and after the move.**
  - **"To think about: Which method of fitting a line (SSE or sum of absolute deviations) seems more resistant to outliers? That is, which regression line changes less as points are moved?"**

Questions to answer afterwards:

- **Which method, least squares or least absolute vertical deviations, is more robust? That is, which method is more resistant to changes in the data values?**
- **Which line, the least squares line or least absolute deviations line, seems more stable? That is, which line moves more smoothly as points are moved?**
- **[Give them a quiz to see if they can tell the difference between least absolute deviations (LAD) fitting and least squares (LS) fitting.]**

Figure 5: Educational objectives for Exercise 7.3c.

Before creating individual storyboards for each exercise, the web site interface was planned. To keep consistent with the layout of previous statistical exercises developed for WPI IQPs, the new layout was not changed greatly from the old designs. Figure 6 shows the plans for the site layout.

Web Site Layout Planning

We decided to continue with the same layout used in the labs developed by previous project groups, for conformity.

As before, clicking the "Applet" link opens up a new window for the applet, which contains all steps for the exercise. The general layout shown here remains the same, with only textual content changes in the Introduction, Glossary, Printing & Saving instructions, and Questions. The textual content appears in the right frame below as links are clicked. Additionally, a way to gather online responses to exercise questions and survey responses has been built into the Questions page.

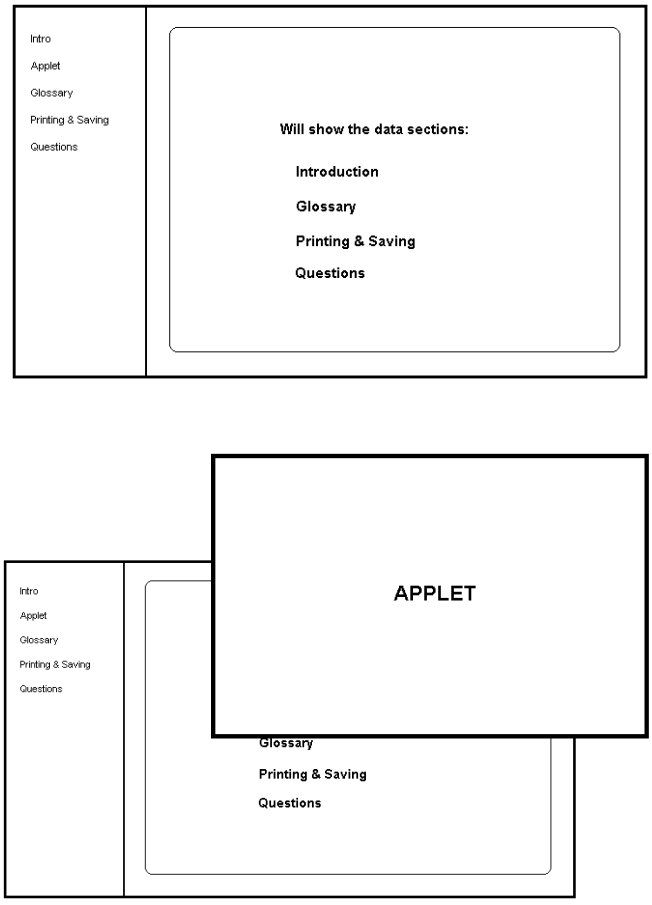
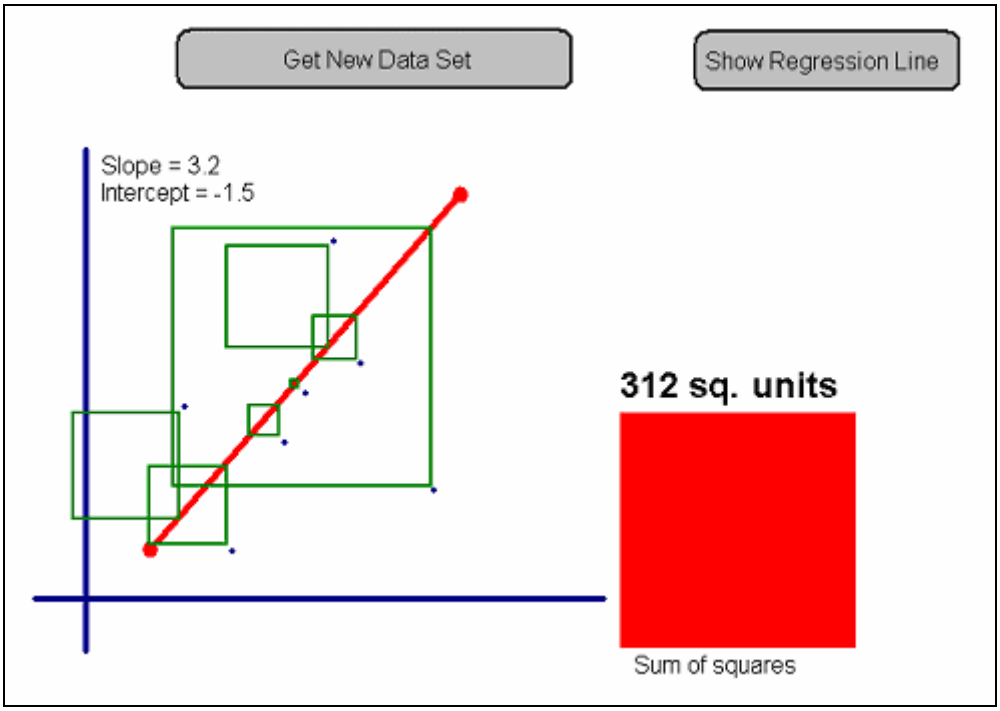


Figure 6: Planned web site layout.

The storyboards outlined how each particular visual layout and the overall progression through the exercise aids students in learning the material. The individual

storyboards for each exercise are shown in Figures 7 - 9. Each storyboard consisted of a visual that showed the basic components of the applet. Each storyboard also listed the steps that the student would follow.

Exercise 7.3a Storyboard



Step 1
In this exercise, we will investigate a popular method of fitting a line to a set of (x,y) data known as the *method of least squares*.

Step 2
The method of least squares is a way to find the best-fitting line by minimizing the sum of the squares of vertical deviations from the data points to the line. These deviations are called *errors* or *residuals*.

Step 3
In the plot shown above to the left, four of the five data points are represented by small black points, and the fifth by a small light blue point (the significance of the light blue point is explained later). The squares of the errors are represented by the other colored squares. The area of each colored square is the square of the error corresponding to one data point.

Step 4
The large square on the right represents the sum of the areas of all the squares, which is called the SSE or sum of squared errors.

Step 5
The red line shown above is *not* the line of best fit (or "regression line"). Move the line around by dragging its endpoints (red dots) to get a feel for how moving it affects the SSE. The SSE will update as the line is

moved.

Step 6
To think about: How is moving the line changing the SSE?

Step 7
Okay, now that you've had some time to play, let's try something new. Now you will try and guess the regression line, given a new set of points. Click the "Get New Data Set" button. You should now have a new set of points, and a default line drawn in red.

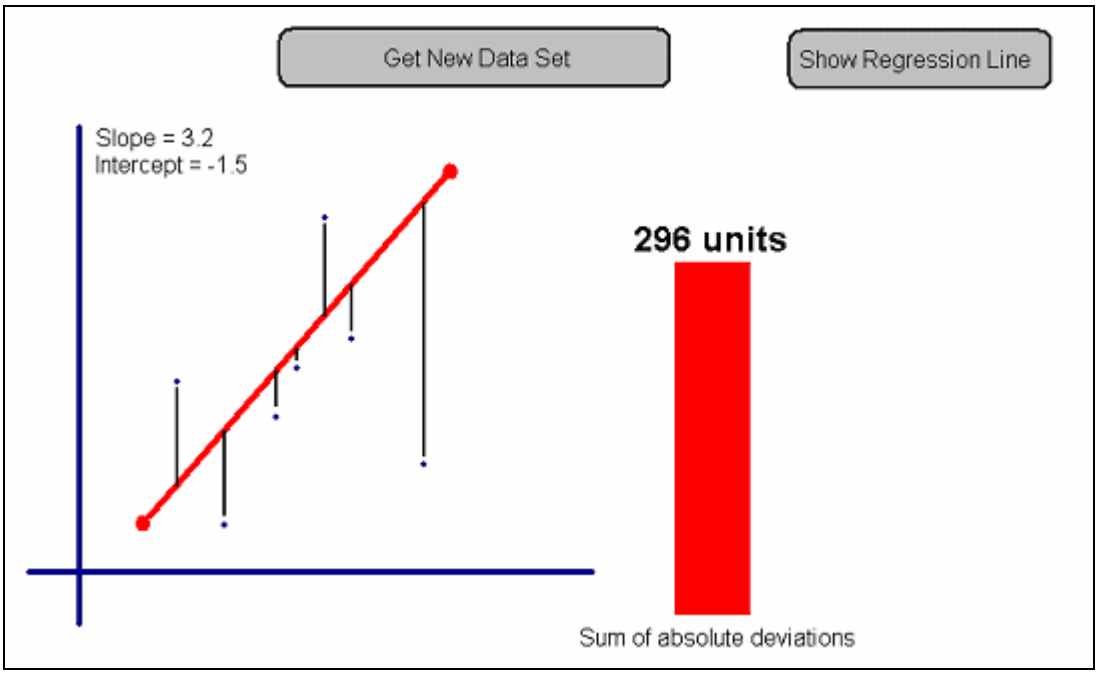
Step 8
Note that the data points are random, but the light blue data point has been chosen to fall on the means of the x and y values. This is because the least-squares regression line always passes through the coordinate (mean of x, mean of y), so this point can help in placing your line to minimize the SSE.

Step 9
Drag the line around as before, and try to find the regression line. Watch the sum of squared errors grow and shrink to help you choose this line of best fit.

Step 10
When you believe you have the regression line, click the "Show Regression Line" button to have the regression line appear in blue. Compare it to your line. A blue square representing the smallest possible SSE will also appear on the right. Record how close your SSE was to the smallest possible SSE. Give yourself a pat on the back if your SSE was within 10% of the best SSE, or try again by clicking "Get New Data Set". Save both plots when finished, using the buttons at the top.

Figure 7: Storyboard for Exercise 7.3a.

Exercise 7.3b Storyboard



Step 1
In this exercise, we investigate an alternative to the method of *least squares* for fitting a line to a set of (x,y) data: the method of *least absolute deviations*. Instead of minimizing the sum of squared errors, as least squares does, the method of least absolute deviations minimizes the sum of the absolute values of the errors.

Step 2
The method of least absolute deviations is a way to find the best-fitting line by minimizing the sum of the absolute values of the deviations from the points to the line. These deviations are called *errors* or *residuals*. In the plot shown above to the left, the five data points are represented by small black points, and the absolute values of the errors are represented by the lengths of the line segments drawn from the points to the line.

Step 3
The bar on the right represents the SAE: sum of all the absolute values of the errors. The method of least absolute deviations seeks the line that minimizes the SAE.

Step 4
The line shown above is *not* the line that minimizes the SAE, which we will call the "regression line" for the remainder of this applet. Move the line around by dragging its endpoints to get a feel for how moving it affects the SAE. The SAE bar will update as the line is moved.

Step 5
To think about: How is moving the line changing the SAE?

Step 6
Now that you've had a chance to play around, we'll try something new. Now you will try and guess the regression line, given a new set of points. Click the "Get New Data Set" button. You should now have a

new set of points, and a default line drawn in red.

Step 7
Drag the line around as before, and try to find the regression line. Watch the SAE grow and shrink to help you choose the best line.

Step 8
When you believe you have the regression line, click the "Show Regression Line" button to have the regression line appear in blue. Compare it to your line. A blue bar representing the best SAE will also appear on the right. Record how close your SAE is to the smallest possible SAE. If your SAE within 10% of the best SAE, give yourself a pat on the back. Otherwise, try again with a new data set by clicking "Get New Data Set". Save both plots when finished, using the buttons at the top.

Step 9
Finally, we'll play with one last idea. Some data sets can have more than one valid least absolute deviations line. Click the "Get New Data Set" button. This time you will be given a data set that is known to have more than one valid least absolute deviations line. (Can you recognize why it does?) Click the "Show Actual Regression Line" button to show one solution in blue. Click "Next Step" to see a green region representing the set of all valid lines.

Step 10
Drag your line so that is falls completely within the green region. Try a few different slopes for the line, and make sure that the SAE doesn't change. You may notice that this is hard to do, since the green region gets very skinny at one point. Click "Next Step" to have one of the regression line points locked to a useful pivot point.

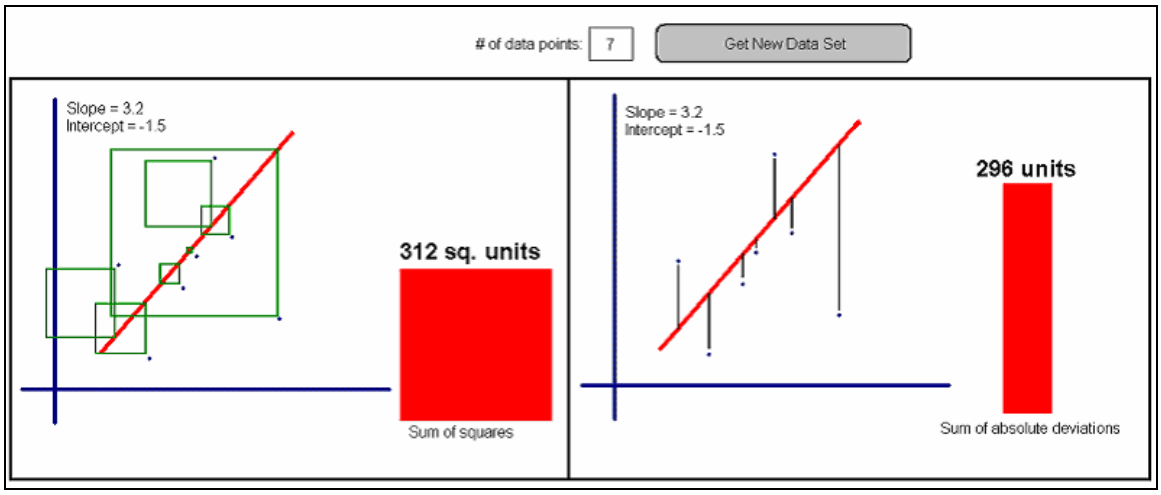
Step 11
Now one of the red regression line points is locked at a pivot point within the green region. Now it should be easier to try a few different slopes for the line, and make sure that the SAE doesn't change.

Step 12
Click the "Get New Data Set" button one last time. Doing this brings up a new set of data points. This time, we want to see if there exists a set of data points that are *not* symmetric about some axis, but still has many valid least absolute deviations lines. Click the "Show Actual Regression Line" button to show one solution in blue. Click "Next Step" to see a green region representing the set of all valid lines.

Step 13
Drag your line so that is falls completely within the green region. Try a few different slopes for the line, and make sure that the SAE doesn't change.

Figure 8: Storyboard for Exercise 7.3b.

Exercise 7.3c Storyboard



Step 1
In this exercise, we compare the *least squares method* with the *least absolute deviations method* for finding regression lines. You have the ability to see the two methods simulated side-by-side.

Step 2
The method of least squares is a way to find the best-fitting line by minimizing the sum of the squares of vertical deviations from the points to the line. These vertical deviations are also known as errors, and the quantity minimized is called the sum of squared errors or SSE. The method of least absolute deviations is similar, but instead minimizes the sum of absolute values of the errors, known as the sum of absolute errors or SAE.

Step 3
In the plots above, the data points are exactly the same. The left plot has a least squares regression line shown, and the right plot has a least absolute deviations regression line shown. The regression lines are fixed according to the data, but you may move the data points in either plot. Go ahead and move some points around. Note that the data points still mirror each other. What changes do you see?

Step 4
To think about: Do you observe that the slopes of the two regression lines are different at some times? Why might they be different? When might they be the same?

Step 5
To think about: Which line seems more "unstable": the least squares line or the least absolute deviations line?

Step 6
Now, click the "Get New Data Set" button to get a new set of points. You will now create an *outlier*, or a data point that does not match the general trend of the rest. Drag the second data point from the left (on either plot) vertically upward about 200.0 units (For the next few steps, points are restricted to vertical movement). Use the "cursor position" to help you. Save each plot using the buttons, before and after the adjustment.

Step 7
To think about: Which error measurement (SSE or SAE) seems more resistant to outliers? That is, which

regression line changes less as points are moved?

Step 8
To think about: In what kind of real-life case might you want to use a method that is more resistant to outliers? Or, in what case might it be *critical* to pay attention to any and all outliers?

Figure 9: Storyboard for Exercise 7.3c.

The final applets do not look exactly like the visuals above, but the final look has a close resemblance to the storyboard drawings.

3.2. Implementation

Coding of each applet took place after completion of each corresponding storyboard. The reason for alternating between applets and storyboards, as opposed to creating all storyboards and subsequently, all applets, is that the number of applets to be produced was undetermined at the start of the project, and depended on how quickly progress was made.

As a first implementation step, a framework was developed in Java. The framework consists of three common components that were necessary for all three exercises: the visual layout, Cartesian plotting functionality, and a simple way to externally query the plot for statistics on all data.

The first framework component produced was the visual layout, as it was simplest to implement and because it dictated much of how the rest of the program had to be designed. The visual layout was created and adjusted to be aesthetically pleasing. Four Java *Panel*s were created to separate each section of the applet: one Panel for the buttons at the top of the exercise, one Panel for the plot(s), one Panel for loading and showing instructions in the form of HTML pages, and one Panel for the "Next Step" and "Prev Step" buttons at the bottom. Each exercise has these basic components. Figure 10 exhibits these four Panels in Exercise 7.3a.

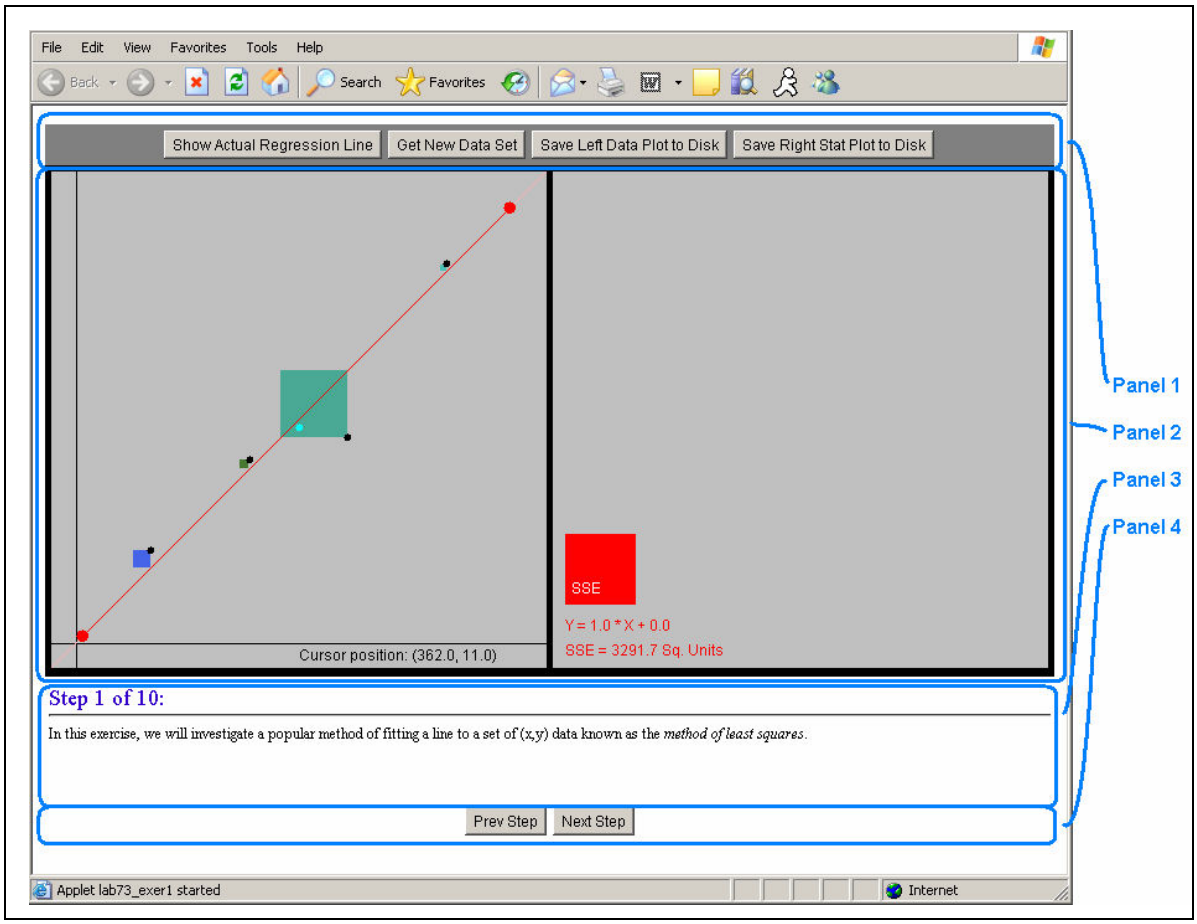


Figure 10: The four basic Panels used in each exercise (Exercise 7.3a shown).

An HTML page display was used for the instruction panel so that the instruction steps could be easily changed without touching the Java source code.

A framework for Cartesian plotting functionality was designed to present a simple interface for the programmer to plot points, squares, and lines. In addition, provisions were made to toggle many options, such as:

- Dragging points or the line in the plot
- Showing a regression line versus allowing a user-adjustable line
- Enabling least squares or least absolute deviations regression lines
- Adding points where the mouse is clicked
- Filling in the squares solid with color or only outline them
- Showing non-unique solutions (in the case of least absolute deviations)

Also, the programmer can easily set:

- The numerical extents of the plot
- The extent of the random distribution of the generated points
- The number of points in the plot

The third framework component implements an interface to allow the programmer to easily query the plot data to retrieve a useful set of statistics, to be used for display on the plot. The interface allows the user to fetch all essential data, such as:

- The type of line being used (whether movable, least squares regression line, or least absolute deviations regression line)
- The movable line's slope and intercept, its corresponding sum of absolute errors, and its corresponding sum of squared errors
- The actual regression line's slope and intercept, its corresponding sum of absolute errors, and its corresponding sum of squared errors

Finally, after implementing the common framework for the applets, miscellaneous necessary functions were implemented, such as methods involving output to a file. Each plot may be saved to a *.PNG* file using the corresponding buttons at the top of each exercise.

3.3. Testing Procedure

The applets were initially tested for technical correctness through peer computer scientist testing and though a unit testing methodology, and revised in response. The applets were then tested for learning effectiveness by having students in a WPI undergraduate statistics class complete each exercise and provide feedback about their experiences. The feedback from the WPI statistics class also allowed for technical criticism, which is another way that the exercises were tested for technical correctness. These testing practices are explained in detail in sections 4.2. and 4.3.

4. Results

This project's deliverables were three interactive Java applets with accompanying web pages for background information and learning evaluation methods.

4.1. Exercise Descriptions

The exercises developed fulfill the goals of the introduction and the educational objectives stated in Figures 3 - 5. The applets created provide a complete lesson demonstrating some of the most important properties of the least absolute deviations fitting method and the more popular least squares fitting method. The applets provide unambiguous visualization tools for both methods. In addition, least absolute deviations and least squares are clearly and concisely contrasted via Exercise 7.3c.

4.1.1. Exercise 7.3a – Least Squares Regression

Exercise 7.3a teaches about the least squares regression method. A plot of data points is shown, along with a least squares regression line. The square of the distance from each point to the line is represented visually by a geometric square. The exercise steps lead the student through various lessons and convey different concepts. The exercise begins with a description of the least squares fitting method, and definition of some terms, such as "errors" and "residual." The elements of the onscreen display are also explained. The students are told that the squares represent the square of each residual, and the large square on the right represents the sum of all the squares of the residuals. The students are then instructed to play with a movable line and observe what happens. This "free play" portion of the lab allows the students to gain some intuition on their own about how moving the line affects the sum of squared errors. During the exercise, the students are asked thought-provoking questions to make sure that they understand the lesson.

After the students finish with the "free play" steps, they are instructed to try to guess the regression line by placing the movable line where they think the regression line is. They are given a hint to help them do so: one point is colored blue and lies on the coordinate $(\bar{x}, \bar{y})$, which is the mean of x and y values. It is a known fact that this point

lies on the least squares regression line, and they are told this fact to help them place a line. Once they guess a line, the students may check their guess by having the applet display the least squares regression line. If their guess was close to the actual regression line, they are congratulated. They are instructed to guess again if their guess was not close to the actual line.

4.1.2. Exercise 7.3b – Least Absolute Deviations Regression

Exercise 7.3b teaches the students about the least absolute deviations (LAD) regression method. A plot of data points is shown, along with a least absolute deviations regression line. The absolute deviation from each point to the line is represented visually by a vertical line. The exercise is very similar in structure to Exercise 7.3a. The exercise instructs the students to play with a movable line and observe its effects on the sum of absolute errors. In addition, the exercise takes the students through the steps of trying to guess an LAD regression line. These steps are similar to those of the least squares regression exercise.

Besides teaching about the basics of the least absolute deviations fitting method, an interesting property of the fitting method is introduced to the students: the fact that LAD regression lines may not be unique. Two different predetermined cases that generate multiple solutions are shown to the students in the last few steps of the exercise. The students are shown the region of multiple solutions, and are instructed to confirm that any line falling completely within that region represents a valid solution. This is accomplished by telling the students to drag a movable line around in the region of multiple solutions and verifying that the sum of absolute errors does not change.

4.1.3. Exercise 7.3c – Least Squares versus Least Absolute Deviations

Exercise 7.3c contrasts the two fitting methods discussed in exercises 7.3a and 7.3b: the method of least squares and the method of least absolute deviations. This exercise shows two side-by-side plots that have the same set of data points. Instead of allowing the students to adjust a movable line as in the first two exercises, the student is allowed to move the data points around in this exercise. The students may drag data

points on one plot, and the opposite plot will immediately have its points updated to match it. Each plot also has a regression line fitted to the data points, which is updated as the data points are moved. The difference between the plots is that the left plot has a least squares regression line fitted to the data, while the right plot has a least absolute deviations line fitted to the data. By moving the data points around, the student can observe the differences between each regression line in real time.

The first few steps instruct the student to have some "free play" time with the applet, as in the first two exercises. In this exercise, they are instructed to move the points instead of a line (since the lines shown are regression lines, and are not user-adjustable). Afterwards, the exercise teaches the student about the fact that least absolute deviations is a more robust fitting method than least squares (or, is more resistant to the effects of outliers). It teaches this by instructing the students to create an outlier by dragging a point vertically upward, and having the students observe how each regression line changes.

4.2. Technical Evaluation

Testing an application is of no less importance than planning and coding an application. Andreas Schaefer, system architect for J2EE,[21] believes that unit testing (the extreme programming practice of continuously testing all aspects of code) "acts as the devil's advocate making sure the design works when used in a client." He states, "The unit test is the client."[22]

Automated unit tests are popular in Java programming and are an effective way to automatically test functionality after small revisions. Where applicable, such tests were conducted using JUnit,[23] a popular unit testing suite. Of course, unit tests are only as effective as the programmer who builds them. Even with unit testing, it was important to have users try to "break" the program by testing extreme inputs and conditions. Software engineering peers at WPI served this purpose well, along with elementary statistics students. Fortunately, it turned out that no technical problems arose during peer testing. The only revisions that were necessary pertained to the effectiveness of the applets in conveying the mathematical concepts and in keeping the students' interest.

[21] http://java.sun.com/j2ee/index.jsp
[22] http://weblogs.java.net/blog/schaefa/archive/2004/10/unit_test_are_a.html
[23] http://www.junit.org

4.3. Learning Effectiveness Evaluation

Though the technical evaluation is important to verify that the applets work correctly, it is equally important to verify that the goal of teaching the concepts effectively was satisfied. A total of 30 students from an introductory WPI Statistics class completed all three exercises. The statistics course, MA2612 Applied Statistics II, is the WPI course the applets are geared towards. During their 1-hour laboratory period, the students were instructed to access and complete the three exercises online.[24] The students worked alone on desktops running Windows XP in WPI's Kaven Hall, room 207. Most students completed all exercises and responses in about 40 minutes; only one student took the full hour. The fact that no student finished in, say, 10 minutes is a positive indication that the students were unable to quickly jump through the exercises, without stopping to understand the concepts. The author of the applets and the MA2612 teaching assistant, Danny Jin, were present during the laboratory to answer any questions and resolve any problems with the exercises. Responses to exercise questions were gathered online, as well as responses to a survey per exercise. The exercise questions and responses, as well as survey questions and responses, are listed in Appendix B. The survey responses lend insight into how well the exercises kept the students' interest, and the exercise questions are intended to show how well the students learned the main concepts.

4.3.1. Survey Results and Revisions

The following remarks are general conclusions drawn from the responses to the survey questions. A complete listing of questions and a summary of the responses is given in Appendix B. Some changes were made to the exercises and applets based on student responses. The changes made were not necessarily based upon the frequency of responses, but were based on perceived effectiveness and feasibility of implementation. The majority of the survey responses showed no indication of unhappiness with the

[24] http://www.math.wpi.edu/Course_Materials/SAS/lablets/7.3/73_choices.html

exercises. Therefore, we responded to suggestions that could only clearly improve the lesson.

Two students noted that it was unclear that the movable line from exercises 7.3a and 7.3b could be moved from both endpoints. As a result, the exercise instructions were modified to clarify this point. Two other students stated that they didn't understand how what they were doing related to what they were supposed to be learning. To remedy this, instructions for all three exercises were made more explicit in general, without adding wordiness.

A total of three students gave comments about the aesthetics of the exercises. Two students complained about the font for the steps being too small, and two students suggested improving the color scheme. These comments are important to consider, as effective visualizations and fun exercises are key factors that encourage deep processing. We agree with the comments concerning the small fonts, and this has been remedied. The color scheme was chosen to have well-contrasting colors that are not too bright and distracting. A couple of colors were modified slightly to be more aesthetically pleasing, but the light gray background color for the plots was retained.

Two students stated that they could have used more background information up front. However, one student also stated that the length of each exercise was just right; any longer and he would have lost interest. To respond to these issues, a link was added to the introduction section of each exercise, leading to a site with more background information on the relevant topic(s). A student can choose to follow the link if he/she wants more background prior to completing the exercises.

4.3.2. Exercise Question Results and Revisions

By far, the questions that the students had the most trouble with were the ones involving "stability" and "robustness" from exercise 7.3c. When asked which line was more robust and which line was more unstable, the students seemed confused. This confusion indicated that the terms were not explained well, which also turned out to be an indication that we, the authors, did not understand the properties well enough to explain them sufficiently. We had previously thought that least squares had a "continuous

solution," and least absolute deviations had a "discontinuous solution." This is actually untrue: both methods have continuous solutions. The "jumping" that the least absolute deviations line exhibits during horizontal data point movement is due to the regression line choosing one side or another of a region of multiple solutions. This reasoning was added to a step in Exercise 7.3c in an attempt to clear up confusion, as well as for correctness' sake. To make the concepts "stability" and "robustness" clearer in the students' minds, the terms "robust" and "unstable" have been bolded within the exercise, and again in the questions section. By doing this, the students can make a connection between the terms and the actions they took during the exercise. Also, the instruction steps now state more explicitly what property is currently being demonstrated.

This was the only major ambiguity in the lessons that needed clarification; however, all responses can be studied in Appendix B. As with the survey results, the *quantity* of similar types of exercise question responses did not wholly guide the changes made; the changes were based on feasibility of implementation. Improvements were made only if they were considered beneficial.

5. Conclusions and Recommendations

The exercises developed in this project aid statistics students in learning about regression methods. The project's deliverables were three interactive Java applets with accompanying web pages for background information and online learning evaluation methods. We gained much insight into student learning factors, as well.

Many learning theories were considered in designing the exercises. Presenting "exciting" lessons and verbal or written encouragement were found to be beneficial to student learning. Visual aids were also found to be advantageous in teaching material effectively. The most general factor to consider when teaching concepts was *deep learning* versus *surface learning*. It was important to encourage deep learning throughout the exercise through various methods. Encouraging deep learning helps students understand concepts, as opposed to memorization of facts. From student responses, it was clear that some students had gained a deep understanding of the regression methods. For instance, the quiz from the exercise 7.3c question set asked students to identify which of two plots was using which regression method, and to justify their response. To answer this question, students had to stop and think of what would be a good way to test which line was which. Responses to the "justification" part of the question show that many students set up a quick experiment to test either the "stability" or "robustness" property. Since the students knew the properties of each line and knew how to test for those properties, we can conclude that those students took more away from the exercises than just memorized facts.

Planning and testing played a key role in the success of this project. Storyboarding was helpful in identifying weak points in the visual layout and/or steps of the exercises, or in finding necessary additions or removals of features. Testing the applets for technical correctness was helpful; due to testing, absolutely no technical problems arose during the use of the exercises by the WPI statistics class. Testing the exercises for learning effectiveness was useful for finding weak points. Survey responses were helpful for hearing what students found to be confusing, and exercise question responses were helpful in showing exactly what students did not understand (even if they thought they did). In one case, it was found that student misunderstanding implicated a

lack of proper explanation of a concept, which in turn indicated a lack of understanding of the concept on our part.

The three interactive exercises consist of: an exercise teaching about the Least Squares Regression method, an exercise teaching about the Least Absolute Deviations regression method, and an exercise comparing the two methods. In general, the statistics students found the exercises to be easy to follow and understand. After testing, small revisions were made where needed, and only when the revisions would be a definite improvement.

This project uncovered a large number of least absolute deviations solving methods discovered or created by the author. This report enumerates and provides pseudo code for most methods. The summary of least absolute deviations and algorithms for solving it has been added to the phenomenally popular informational wiki, Wikipedia.org,[25] for the public good, as "least absolute deviations" did not have an entry at this site.

The least absolute deviations applet has the ability to do a couple of novel things that, to the best of the author's knowledge, no current publicly available visualization tools offer:

- Show clearly multiple solutions when they exist.
- Show the least absolute deviations line equation.

If these resources are available, they are certainly hard to discover. A couple applets were found to demonstrate least absolute deviations versus least squares solutions, but they did not show the equations of the lines nor multiple solutions when they existed. An implication of this is that a lot of scholars may be "in the dark," so to speak, about the properties of least absolute deviations. This project hopes to bring some recognition to the least absolute deviations method.

In conclusion, we found that the learning theories researched and applied were effective in producing interactive teaching aids that improve upon conventional statistical teaching methods. The project also recognizes least absolute deviations, a useful but little-known regression method, as an alternative to the popular least squares regression method. The exercises were a technical and educational success (based on survey and

[25] http://en.wikipedia.org/wiki/Least_absolute_deviations

exercise question results) and were well accepted among the WPI statistics students who evaluated them.

5.1. Future Work

There are many statistical topics that may benefit from online interactive learning aids. The following are a few suggestions.

- *Expand the lessons to teach other regression methods*
  Least squares and least absolute deviations are certainly not the only regression methods. Other methods exist, and could be contrasted with the two presented in this project. A solid framework already exists for plotting regression, and may be easily expanded upon.

- *Formal definitions of, and lessons teaching leverage points and outliers*
  Though "outliers" were mentioned in Exercise 7.3c when comparing the robustness of least squares and least absolute deviations, outliers were not formally defined. "Leverage points" are conceptually similar to outliers, and would be an appropriate concept to contrast with outliers in an independent exercise.

- *Though unrelated: multivariate data analysis / transformations*
  Development of an exercise was started that would teach about transformations on multivariate data,[26] but no exercises were completed. Completion of this lesson would be beneficial to undergraduate statistics students.

5.2. Acknowledgements

Thanks are in order to the following resources for their help in this project.

- WPI Professor J. Petruccelli, for helping to find least absolute deviations solving and for continually editing this report.

- WPI Professor R. Clements, for helping identify alternative least absolute deviations solving methods.

[26] http://www.math.wpi.edu/Course_Materials/SAS/lablets/7.3/73_choices.html

- The *Usenet* community (*sci.stat.math* and *sci.math* groups) and the *lp_solve* linear programming solver user community, for their ideas for least absolute deviations solving methods, and for remedies for IRLS numerical instability problems.

- All references from section 6, but especially *[Li and Arce, 2003]*.

6. Bibliography

Apter, M.J. (1989) *Reversal Theory: motivation, emotion and personality* London: Routledge

Atherton, J. S. (2005) *Learning and Teaching: L and T template.* Available online: http://www.learningandteaching.info/learning/motivanx.htm. Accessed: 25 September 2005.

Bellmore, Jarrod Thomas. Blaquiere, David Lee. Lewis, Adam LaFond. Rahman, Evgeny. (2005) *Statistical Teaching Aids.* Worcester Polytechnic Institute Interactive Qualifying Project.

Bjorck, Ake. (1996) *Numerical Methods for Least Squares Problems.*

Brewster, Cori. Fager, Jennifer. (2000) *Increasing Student Engagement and Motivation: From Time-on-Task to Homework*. Northwest Regional Education Laboratory. Available online: http://www.nwrel.org/request/oct00/textonly.html

Brooks, S.R., Freiburger, S.M., & Grotheer, D.R. (1998). *Improving elementary student engagement in the learning process through integrated thematic instruction.* Unpublished master's thesis, Saint Xavier University, Chicago, IL. (ERIC Document Reproduction Service No. ED 421 274)

Burgess, Dr. Lesta A., Strong, Dr. Shawn D. (2003) *Trends in Online Education: Case Study at Southwest Missouri State University*. Journal of Industrial Technology. *Volume 19, Number 3 - May 2003 to July 2003.* Available online: http://www.nait.org/jit/Articles/burgess041403.pdf

Clark, C., Mayer, R. (2003) *Proven Guidelines for Consumers and Designers of Multimedia Learning*. e-learning and the Science of Instruction. San Francisco, CA. Jossey-Bass/Pfeiffer.

Clein, Robert Haiman. Holmes, Samuel Benjamin. (2003) *Online Statistics Labs.* Worcester Polytechnic Institute Interactive Qualifying Project.

Dorai-Raj, S., Anderson-Cook, C., Robinson, T. (2000-2002) *Statistical Java: An Interactive Environment for Teaching Statistics*. Virginia Tech Department of Statistics. Available online: http://kitchen.stat.vt.edu/~sundar/java/applets/

Finzer, Bill. *The Geometer's Sketchpad: Least Squares.* Available online: http://www.keypress.com/sketchpad/javasketchpad/gallery/pages/least_squares.php

Gibbs, G. (1992) *Improving the Quality of Student Learning*, Bristol: Technical and Educational Services.

Gottreu, Brian Phillip. Slater, Jeremy Aaron. (2003) *Statistical Teaching Aids.* Worcester Polytechnic Institute Interactive Qualifying Project.

Hartley, James. (1998) *Learning and Studying: A Research Perspective*. New York, NY. Routledge Publishing.

Hebb, D.O. (1955) *Drives and the C.N.S. (Conceptual Nervous System). Psychological Review* 62:243-54

Institute of Statistics and Decision Sciences. *ISDS – Statistics Sites*. Available online: http://www.isds.duke.edu/sites/java.html

Java Technology. Available online: http://java.sun.com/

JavaReference.com. Available online: http://www.javareference.com/

JavaWorld. Available online: http://www.javaworld.com/

JUnit.org. *JUnit, Testing Resources for Extreme Programming*. Available online: http://www.junit.org

Kawato, Takeshi. (2003) *Statistical Teaching Aids with Java Applets.* Worcester Polytechnic Institute Interactive Qualifying Project.

Kincaid, David Cheney, Ward. (2003) *Numerical Mathematics and Computing*.

Li, Yinbo. Arce, Gonzalo. (2003) *A Maximum Likelihood Approach to Least Absolute Deviation Regression.* Available online: http://www.hindawi.com/GetArticle.aspx?doi=10.1155/S1110865704401139

Lieser, Eric Dale. Whitford, Paul Charles. (2001) *Statistical Teaching Aids.* Worcester Polytechnic Institute Interactive Qualifying Project.

Lindsay, Peter H. Normal, Donald A. (1977) *Human Information Processing*. New York, NY. Academic Proess, Inc.

Lucas, A. F. (1990) *Using Psychological Models to Understand Student Motivation.* In M. D. Svinicki, *The Changing Face of College Teaching.* New Directions for Teaching and Learning, no. 42. San Francisco: Jossey-Bass.

Marton, F., Saljo, R. (1976) *On Qualitative Differences in Student Learning: 1. Outcome and Process'*. British Journal of Educational Psychology, 46, 1:4-11.

Michiels, Stefan. Raeymaekers, Bert. (2000) *Java Applets for Visualization of Statistical Concepts*. Katholieke Universiteit Leuven - University Center for Statistics. Available online: http://www.kuleuven.ac.be/ucs/java

National Academic Press. (1999) *How People Learn: Brain, Mind, Experience, and School.* Available online: http://www.nap.edu/html/howpeople1/

Pezzullo, John C. *Interactive Statistical Calculation Pages*. Available online: http://members.aol.com/johnp71/javastat.html

Pike, R. W. (1994) *Visual aids: how to keep their attention when you absolutely have to talk. Creative Techniques Training Handbook.* (pp. 41-75).

Rice University. *Rice Virtual Lab in Statistics*. Available online: http://www.ruf.rice.edu/~lane/rvls.html

Roediger, H. *Memory: Explicit and Implicit.* Paper presented at the Symposium, Recent Advances in Research on Human Memory, National Academy of Sciences. Washington, DC.

Schaefer, Andreas. *Unit Test are at least as Important as the Coding Itself.* Personal web log. Available online: http://weblogs.java.net/blog/schaefa/archive/2004/10/unit_test_are_a.html

Stanton, Charles. *Java Demos for Probability and Statistics.* Available online: http://www.math.csusb.edu/faculty/stanton/m262/

Sun Microsystems. *Java 2 Platform, Enterprise Edition (J2EE)*. Available online: http://java.sun.com/j2ee/index.jsp

Sun Microsystems. *Java 2 Platform Std. Ed. v1.4.2 API*. Available online: http://java.sun.com/j2se/1.4.2/docs/api/

Wesolowsky, G. O. (1981) *A new descent algorithm for the least absolute value regression problem*. Communications in Statistics, Simulation and Computation, vol B10, no. 5, pp. 479-491.

Wolfram Research: Mathworld. *Least Squares Fitting.* Available online: http://mathworld.wolfram.com/LeastSquaresFitting.html.

Worcester Polytechnic Institute. *WPI Web-based Statistics Labs*. Available online: http://www.math.wpi.edu/Course_Materials/SAS/lablets/statlab.html.

Appendix A: Least Absolute Deviations Solving Methods

A.1. Iteratively Re-weighted Least Squares

The IRLS algorithm finds the *b* and *m* that minimizes:

(1) $\sum_{i=0}^{N-1} w_i (Y_i - b_{(n)} - m_{(n)} X_i)^2$ **, where** $w_i = \frac{1}{|Y_i - b_{(n-1)} - m_{(n-1)} X_i|}$.

$b_{(n-1)}$,$m_{(n-1)}$ represents parameters found from the previous iteration, and $b_{(n)}$,$m_{(n)}$ are the parameters to solve for.

Take derivatives of (1) with respect to *b* and *m*.

(2) $\frac{d}{db} \sum_{i=0}^{N-1} w_i (Y_i - b_{(n)} - m_{(n)} X_i)^2 = -2 * \sum_{i=0}^{N-1} w_i (Y_i - b_{(n)} - m_{(n)} X_i)$

(3) $\frac{d}{dm} \sum_{i=0}^{N-1} w_i (Y_i - b_{(n)} - m_{(n)} X_i)^2 = -2 * \sum_{i=0}^{N-1} w_i X_i (Y_i - b_{(n)} - m_{(n)} X_i)$

Set (2) and (3) equal to zero to find the minimum of each equation.

(4) $-2 * \sum_{i=0}^{N-1} w_i (Y_i - b_{(n)} - m_{(n)} X_i) = 0$

(5) $-2 * \sum_{i=0}^{N-1} w_i X_i (Y_i - b_{(n)} - m_{(n)} X_i) = 0$

(4) and (5) can be simplified.

(6) $\sum_{i=0}^{N-1} w_i Y_i = b_{(n)} \sum_{i=0}^{N-1} w_i + m_{(n)} \sum_{i=0}^{N-1} w_i X_i$

(7) $\sum_{i=0}^{N-1} w_i X_i Y_i = b_{(n)} \sum_{i=0}^{N-1} w_i X_i + m_{(n)} \sum_{i=0}^{N-1} w_i (X_i)^2$

In (6) and (7), the only unknowns are *b* and *m*. Simply solve using your preferred method.

(8) $b_{(n)} = \frac{\sum_{i=0}^{N-1} w_i (X_i)^2 \sum_{i=0}^{N-1} w_i Y_i - \sum_{i=0}^{N-1} w_i X_i \sum_{i=0}^{N-1} w_i X_i Y_i}{\sum_{i=0}^{N-1} w_i \sum_{i=0}^{N-1} w_i (X_i)^2 - \left(\sum_{i=0}^{N-1} w_i X_i \right)^2}$

$$\textbf{(9)}\ m_{(n)} = \frac{-\sum_{i=0}^{N-1} w_i X_i \sum_{i=0}^{N-1} w_i Y_i + \sum_{i=0}^{N-1} w_i \sum_{i=0}^{N-1} w_i X_i Y_i}{\sum_{i=0}^{N-1} w_i \sum_{i=0}^{N-1} w_i (X_i)^2 - \left(\sum_{i=0}^{N-1} w_i X_i \right)^2}$$

After a sufficient[27] number of iterations *n*, equations (8) and (9) define the least absolute deviations regression line.

A.2. Incrementally adjust m and b until convergence

(1) Choose an initial *m* and *b* from the least squares solution.

(LEAVE *b* ALONE, PERTURB *m*)

(2) Set 'step_size' = 1.0
(3) Decrease *m* by 'step_size' and calculate sum of absolute errors.
(4) Multiply 'step_size' by 2.
(5) Repeat steps 3-4 until the sum of absolute errors starts to increase.
(6) Repeat steps 3-5, only "Increase…" in step 3.
(7) Divide 'step_size' by 2.
(8) Decrease M by 'step_size' and calculate sum of absolute errors.
(9) Repeat steps 7-8 until the sum of absolute errors starts to increase.
(10) Divide 'step_size' by 2.
(11) Increase *m* by 'step_size' and calculate sum of absolute errors.
(12) Repeat steps 10-11 until the sum of absolute errors starts to increase.
(13) Repeat steps 7-12 until *m* barely changes from its last estimate.

(LEAVE *m* ALONE, PERTURB *b*)

(14) Do the same as steps 3-13 for *b*, with *m* fixed at its new value.
(15) Save the values of *m* & *b*. Repeat steps 3-14 until *m* & *b* both converge, or until 100 iterations pass (to avoid an infinite loop).

Steps 2 through 6 are included solely to locate the relative location of the line quickly. These steps increase the "step size" in order to quickly locate the point at which the line produces a larger sum of absolute errors than the previous line. Then the step size is divided by two, and the slope is moved in the opposite direction. This is repeated until the slope is "good enough," at which point the intercept is perturbed in the same fashion. After the intercept is "good enough," we return to the slope to readjust it for optimality. We continue this procedure until both the slope and intercept converge.

A.3. Basic iterative approach discussed in [Li and Arce, 2003]

(1) Set k = 0. Find an initial value m_0 for *m*, such as the least squares solution.

(2) Set k = k+1 and obtain a new estimate of *b* for a fixed m_{k-1} using:

[27] Define "sufficient" for the specific application.

$$b_k = MED\left(Y_i - m_{k-1} X_i \Big|_{i=1}^{N} \right)$$

(3) Obtain a new estimate of *m* for a fixed b_k using:

$$m_k = MED\left(|X_i| \lozenge \frac{Y_i - b_k}{X_i} \Big|_{i=1}^{N} \right)$$

(4) Once m_k and b_k do not deviate from a_{k-1} and b_{k-1} within a tolerance range, end the iteration. Otherwise, go back to step 2.

$\lozenge$ is the replication operator. For example, $A \lozenge B$ produces *B* replicated *A* times. *MED* is the weighted median of a set, found for a set of positive real weights via the following procedure.

$$Y = MED\left(W_i \lozenge X_i \Big|_{i=1}^{N} \right)$$

(1) Calculate the threshold $W_0 = \frac{1}{2} \sum_{i=1}^{N} W_i$.

(2) Sort all the samples into $X_{(1)}, \ldots, X_{(N)}$ with the corresponding concomitant weights $W_{[1]}, \ldots, W_{[N]}$.

(3) Sum the concomitant weights beginning with $W_{[1]}$ and continuing up in order.

(1) The weighted median output is the sample $X_{(j)}$ whose weight causes the inequality $\sum_{i=1}^{j} W_i \geq W_0$ to hold first.

A.4. Wesolowsky's direct descent method [Wesolowsky, 1981]

(1) Set *k* = 0. Choose initial values for m_0, b_0, such as the least squares solution. Choose *j* such that $|Y_j - m_0 X_j - b_0|$ is a minimum.

(2) Set *k* = *k+1*. Use the weighted median structure to get the update for *b*:

$$b_k = MED\left(\left| 1 - \frac{X_i}{X_j} \right| \lozenge \frac{Y_i - \frac{Y_j X_i}{X_j}}{1 - \frac{X_i}{X_j}} \Big|_{i=1}^{N} \right)$$

(3) (a) If $b_k - b_{k-1} = 0$: if $k \geq 3$, go to step 4; else, set *j* = *i* and go to step 2.

(b) If $b_k - b_{k-1} \neq 0$: set *j* = *i* and go to step 2.

(4) Let $b^* = b_k$, $m^* = \frac{Y_j}{X_j} - \frac{b^*}{X_j}$, where b^*, m^* are the final solution parameters.

A.5. Li's proposed new algorithm [Li and Arce, 2003]

(1) Set $k = 0$. Initialize b to be b_0 using the least squares solution:

$$b_0 = \frac{\sum_{i=1}^{N}(X_i - \overline{X})(\overline{Y}X_i - \overline{X}Y_i)}{\sum_{i=1}^{N}(X_i - \overline{X})^2}$$

Calculate m_0 by weighted median:

$$m_0 = MED\left(|X_i| \Diamond \frac{Y_i - b_0}{X_i} \Big|_{i=1}^{N} \right)$$

Keep the index j that satisfies $m_0 = \frac{Y_j - b_0}{X_j}$. In the parameter space, (m_0, b_0) is on the edge line with slope ($-X_j$) and intercept Y_j.

(2) Set $k = k+1$. In the sample space, right shift the coordinates by X_j so that the newly formed y'-axis goes through the original (X_j, Y_j). The transformations in the sample space are

$$X_i' = X_i - X_j \ , \ Y_i' = Y_i$$

and the transformations in the parameter space are

$$m_{k-1}' = m_{k-1} \ , \ b_k' = b_{k-1}' = b_{k-1} + m_{k-1} X_j \ .$$

The shifted sample space (X', Y') corresponds to a new parameter space (m', b'), where $(-X_j', Y_j')$ represents a horizontal line.

(3) Perform a weighted median to get a new estimate of a':

$$m_k' = MED\left(|X_i'| \Diamond \frac{Y_i' - b_k'}{X_i'} \Big|_{i=1}^{N} \right).$$

Keep the new index t that gives $a_k' = \frac{Y_t' - b_k'}{X_t'}$.

(4) Transform back to the original coordinates:

$$a_k = a_k' \ , \ b_k = b_k' - a_k' X_j$$

(5) Set $j = t$. If a_k is identical to a_{k-1} within a tolerance, end the program. Otherwise, go back to step 2.

Appendix B: Survey/Exercise Questions and Responses

Response Summary

7.3a Questions (30 responses total)	7.3a Survey (18 responses total)
7.3b Questions (30 responses total)	7.3b Survey (17 responses total)
7.3c Questions (29 responses total)	7.3c Survey (14 responses total)

Survey Questions

Exercise 7.3a

1. Were the exercise instructions clear and easy to follow? If not, please specify why not.

- Yes (17/18)
- Yes, very straightforward

2. Was the interface (i.e. way of dragging points, moving through steps) easy to use for this exercise?

- Yes (16/18)
- It was difficult to understand at first because there was no instruction given to move the red line by dragging the red dots.
- No, I didn't know you could move the bottom portion until I made it show the Regression Line the first time. Add a rotation for the line instead of a drag.

3. Were there any problems/errors you found while doing the exercise?

- No (16/18)
- Text was really small
- It was difficult to understand at first because there was no instruction given to move the red line by dragging the red dots

4. What was the hardest concept to understand, if any?

- None (14/18)
- SSE, but not really
- SSE
- The mathematical language used
- The overall concept since not too much background was given on the topic and has not been covered in class yet. The general idea is very apparent but the specifics of it are a little vague based on the descriptions of what is going on.

5. Did you find the applet to be helpful in learning about least squares?

- Yes (17/18)
- Slightly, I would have rather seen some mathematical background instead of graphical before using this applet.

6. What did you like about using applets in this exercise?

- It was interactive.
- Very easy to use
- Interactivity

- It's easy
- More hands on than just reading about it in a book
- It gives you a visual real time model to work with. Everything is not static and you can actually see the math and what happens when you change things.
- It provides an interactive interface
- Interactive.
- It was fun
- Hands on, like visual learning
- Easy
- That you could see things visually and made it easy to determine a better line of fit, with a more detailed background on the subject the graphs would have made much more sense.
- Simple and straight forward.
- Relating the visuals to the numbers
- I think it's easier to understand because it's visual and we're not just reading formulas out of a book
- It made the process interactive.
- Visual
- It was easy to visual what I was doing

7. What did you dislike about using applets in this exercise?

- Nothing (13/18)
- The data that was used was "just data" there wasn't any relation to a real world application/example.
- Can only rely on what is stated.
- Fonts too small.
- Not enough background information
- Confusing at first

8. Do you have any general suggestions on how to improve this exercise?

- No (15/18)
- Use a story, or example to demonstrate the data.
- Have the student see this in lecture before looking at it in lab in order to gain some more details about the topic and then see it visually which I think would give a better understanding.
- Add a rotation ability instead of drag and drop. The length of the line is nearly meaningless.

9. Which browser are you currently using?

Internet Explorer
FireFox
Netscape Navigator
Other

- Internet Explorer (18/18)

Exercise 7.3b

1. Were the exercise instructions clear and easy to follow? If not, please specify why not.

- Yes (16/17)
- Not really, I don’t really understand what I was supposed to do.

2. Was the interface (i.e. way of dragging points, moving through steps) easy to use for this exercise?

- Yes (17/17)

3. Were there any problems/errors you found while doing the exercise?

- No (17/17)

4. What was the hardest concept to understand, if any?

- None (11/17)
- SAE could have been explained better.
- That there could be many solutions and why you would want that to happen.
- Why this type of analysis is useful.
- How to calculate the SAE because there is so much writing in the introduction that I just skipped over it
- The area
- Understanding multiple solution regions took some thinking.
- Calculating the slope and y-intercept of the line at the end without a calculator.

5. Did you find the applet to be helpful in learning about least absolute deviations?

- Yes (16/17)
- Could have gone a little more in depth

6. What did you like about using applets in this exercise?

- Instant feedback
- The visuals
- Again, very hands on, I like to be able to play around and see what happens visually.
- Simple and straight forward.
- It's really easy.
- It was fun.
- That the visual changed right when a changed the line
- Provides visualization that describes the method.
- It's not reading out of a textbook, it actually let's you see how each movement affects the whole
- More hands on than a lecture
- It was able to show visual representation of the material.
- Easy to use
- Visual
- It’s interactive
- Interactivity
- They worked well and were concise

7. What did you dislike about using applets in this exercise?

- Nothing (12/17)
- Colors are uninspiring.
- It was in Java.
- Can't ask questions, not enough help
- That there wasn't enough description about what was going on.
- I didn’t really know what I was looking at

8. Do you have any general suggestions on how to improve this exercise?

- None (16/17)
- Just maybe finding some way to make the intro more interactive
- Comment: “I liked the time frame that it took, any longer and I would have lost interest.”

9. Which browser are you currently using?

Internet Explorer
FireFox
Netscape Navigator

Other

- Internet Explorer (17/17)

Exercise 7.3c

1. Were the exercise instructions clear and easy to follow? If not, please specify why not.

- Yes (14/14)

2. Was the interface (i.e. way of dragging points, moving through steps) easy to use for this exercise?

- Yes (14/14)

3. Were there any problems/errors you found while doing the exercise?

- No (14/14)

4. What was the hardest concept to understand, if any?

- Nothing (12/14)
- Probably keeping the names of the two methods straight
- The mathematical language

5. Did you find the applet to be helpful in comparing least squares and least absolute deviations?

- Yes (14/14)

6. What did you like about using applets in this exercise?

- It's not reading out of a textbook, it was hands on.
- That is showed everything vertically.
- Simple and straight forward.
- The visualization of the relationship between the two methods
- Visual
- Very hands on
- The ability to see the concept in use and try different line positions
- It compares the two with identical data sets, which can be easier to understand.
- Easy and simple.
- Interactivity and seeing the differences visually between the two methods.
- The fact that I could see both methods simultaneously
- That it explained things thoroughly
- Real-time comparison; much easier than a written description.

7. What did you dislike about using applets in this exercise?

- Nothing (12/14)
- That it didn't describe in enough detail what was going on.
- Bland colors.

8. Do you have any general suggestions on how to improve this exercise?

- No (12/14)
- Stylize the design.. not rainbow colors but something more appealing and modern.
- Taking one survey at the end of the entire exercise rather than at the end of each section.

9. Which browser are you currently using?

Internet Explorer
FireFox
Netscape Navigator
Other

- Internet Explorer (14/14)

Exercise Questions

Exercise 7.3a

1. What is the SSE? How does it relate to the squares drawn from each point?

Percent incorrect: 6.67%
Incorrect responses:
- SSE is the Sum of the squared errors. It is the sum of the squares created from the horizontal and vertical distance between a data point and the line of regression.
- It is the sum of squared errors. The vertical or horizontal distance to the line, whichever is shorter, is squared and this gives an area, the SSE is the sum of the areas for every error.

2. How did you use the SSE to guess a regression line?

<judgmental>

3. Why was the point $(\bar{x}, \bar{y})$ *helpful to know when trying to find the regression line?*

Percent incorrect: 0%

Exercise 7.3b

1. How did you use the sum of absolute deviations to guess a regression line?

<judgmental>

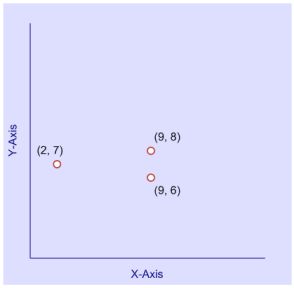
2. From what you've experienced, you know there are some data sets that have more than one least absolute deviations solution. Do you think that for these cases there are infinitely many solutions? Or not? Or depends?

Percent incorrect: 30%
Incorrect responses:
- It depends on the set. Not all of these sets could have infinitely many solutions because there were regions that the line needed to be in.
- I think it could mean either. The data set could indeed imply an infinite number of solutions. Or perhaps the solution is not a regression line but an area function.
- It depends on the structure of the query, or the type of data your working with.
- It depends. But the solution could be characterized by a family of lines.
- So far I had one where there is no way I could get a low SAE , so I would say it depends on the data sets that are presented because data itself could vary drastically at times.
- It can depend. Some do seem to have an myriad of solutions but some do only when certain points are locked.
- I believe that it depends upon the data set.
- Some sets of data points have more than one valid least absolute deviations line. However, a least squares regression line is always a unique solution for any given data set.
- No I don't think there are infinitely many solutions.

- Theoretically, if there were an infinite number of points all set equidistant from each other there could be infinite solutions. Most likely there will always be a finite number of solutions.

3. You've seen that it's possible to have more than one valid least absolute deviations line. Do you think that the "non-uniqueness" property is a good thing or a bad thing? (Hint: see the Introduction for this exercise)

<judgmental>



4. The plot shown above has three data points. The least absolute deviations method produces multiple valid lines for this set of data points. Can you identify where one least absolute deviations line might be? (Hint: One line goes through two of the data points.)

*4a. Find the equation of the line (solve for m and b in "y = m*x + b").*

Percent incorrect: 46.67%
Incorrect responses:
- y = (1/7)*x+1
- y = 1 * x + 7
- y = 1/7 * x + 6
- y = (1/7) * x + (2 - 2/7)
- y = 1/7 * x + 6.5
- y = 1/7 * x + 13/7
- y = 1/7 * x + 12/7
- y = (1/7) * x + 1
- y = (1/7)x + 5/7
- y = 1/7 * x + 6
- y = -1/7m * x + 2.5
- y = 1 * x + 5
- y =1/7 * x + 1.8
- y = 4/9 * x + 4

4b. Calculate its Sum of Absolute Errors (SAE).

Percent incorrect: 16.67%
Incorrect responses:
- SAE = 1
- SAE = 3
- SAE = <empty>
- SAE = 11
- SAE = 4

4c. It turns out that the horizontal line "y = 7" is another least absolute deviations line. Calculate this line's SAE, and make sure it's the same as from part 4b.

<no answer required>

4d. Can you guess the area where the set of multiple solutions lies? Describe it. (Hint: Think of the green region that appeared during the exercise. Describe that region, in this case.)

Percent incorrect: 16.67%
Incorrect responses:
- With the green region touching the point of the mean of x and y values, the green region would be nearly horizontal at about y = 7.
- Area=0
- Where the multiple solutions lie the area is the same.
- <blank>
- Below the (2,7) data point and slightly to the right.

Exercise 7.3c

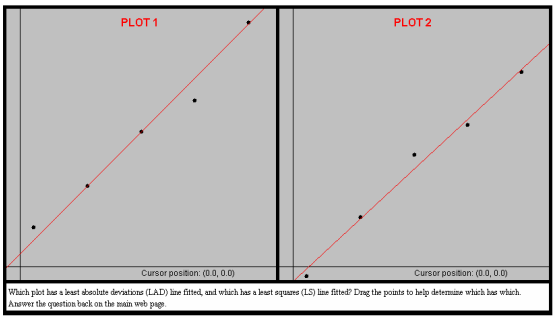
1. Which method, least squares or least absolute deviations, is more robust? That is, which method is more resistant to changes in the data values?

Percent incorrect: 13.80%
Incorrect responses:
- Least squares
- The Least Squares regression line
- The LS since when changing data points in the LAD the slightest change will alter the linear regression line. With LS you need to drastically alter one of the end points.
- The least square is more robust, and the least absolute deviations is more resistant to changes in the data values.

2. Which line, the least squares line or least absolute deviations line, seems more stable? That is, which line moves more smoothly as points are moved?

Percent incorrect: 55.17%
Incorrect responses:
- Absolute deviations line
- The least absolute deviations
- Least absolute deviations line
- The least absolute deviations line seems more stable because it had much smaller changes in its slope as data points moved.
- SAE seemed more stable.
- The least absolute deviations is more stable as in the resistance to change.
- The Least absolute deviations regressions line.
- The least absolute deviations line seems more stable. It did not move as easily when points were being moved.
- The least absolute deviations regression line
- Least absolute deviations
- The LAD since changing any of the data points causes the linear regression line to change instantly. You can see this when using the quiz applet below very clearly.
- The least deviations line
- Least absolute deviation
- The method of Least Absolute Deviations
- The Least Absolute Deviations line seemed more stable
- A least absolute deviations line is more stable and the line moves more smoothly as the points are moved.

3. Take this quiz to see if you can tell the difference between least absolute deviations (LAD) fitting and least squares (LS) fitting.



( ) PLOT 1: LAD line, PLOT 2: LS line
( ) PLOT 1: LS line, PLOT 2: LAD line

Percent incorrect: 13.79%
Incorrect responses:
- answer 2
- answer 2
- answer 2
- answer 2

Please explain your choice for question 3.

When 3. was wrong, here were the answers to 4.:
Incorrect responses:
- In LAD you need to offset the values, whereas with LS you need to get as close to all points as possible.
- According to question 1 and 2.
- Because plot2 has more outlined points than plots one that will allow the line move more smoothly as points are moved.
- When you move points on the left you notice the regression line moves up and down since the LAD deals with deviations vertically that is clearly which one the LAD is.

Appendix C: Programmer's Notes

Each applet has four java files: *dot.java*, *lab73_exer*.java*, *PlotCanvas.java*, and *StatCanvas.java*. *dot.java* contains an extremely simple class implementing 2D coordinate objects. *lab73_exer*.java* is the main applet entry point. This is where the visual layout of the applet is defined. *PlotCanvas.java* contains a class *PlotCanvas* that extends the class *Canvas*. This class presents an easy-to-use interface for Cartesian plotting and regression. Finally, *StatCanvas.java* contains a class *StatCanvas* that also extends the class *Canvas*. This class is used to query *PlotCanvas* for data and to display any visualization of SSE or SAE in exercises 7.3a and 7.3b.

Some code remains commented in the source files (most notably in the *lab73_exer*.java* files). All *System.out.println( )* statements are commented for the final release, as this output clogs the Java console. Some Java *Button* declarations and initializations are commented out as well; these buttons were used to toggle features that were helpful when debugging the applets. They may be uncommented and used if desired.

It was necessary to generate *signed .jar files* and *certificate files* for each applet, to allow the applet to have extra permissions that allows saving of a screenshot to the hard disk, for example. Four batch files were created to automatically generate *signed .jar files* and *certificate files* for each applet, given a *.jar file* for each original applet. (Although there are only three exercises, question three of exercise 7.3c introduces a fourth applet.) The batch files are named:

sign1.bat
sign2.bat
sign3.bat
sign3_quiz.bat

Make sure the JDK[28] is installed before running the batch files. Edit the files in a text editor to make sure the path to the *.jar files* and to JDK's *bin* directory are appropriately chosen.

The web pages accompanying the applets are fairly self-explanatory if one understands basic HTML. Only a small amount of JavaScript is used in launching a window of fixed size to display an applet. One should not need to edit this framework drastically to create new pages.

[28] http://java.sun.com/javase/index.jsp