DESIGNING GAMES FOR UNDERSTANDING IN A DATA ANALYSIS ENVIRONMENT

Tim Erickson
Lick-Wilmerding High School
Epistemological Engineering
755 Ocean Avenue
San Francisco, CA 94112
eepsmedia@gmail.com

This paper describes design issues surrounding "Data Games," small-scale, short web-based games that generate data for analysis in mathematics or statistics classes. The games are embedded in a data analysis environment. We discuss design for the games themselves as well as some curriculum and assessment issues.

Ordinarily, when a student plays a game on a computer, a great deal of data is generated, but that data is never used. Data Games is a project funded by the US National Science Foundation for which we have been developing web-based games embedded in a data-analysis environment. As students play, the system collects the data, and students can analyze it. We have designed the games so that data analysis is rewarding, that is, it's the best way to improve your performance.

Data Games do not need to be about statistics. Rather, we believe that they are an engaging environment in which "regular" math is useful. That is, the skills and habits of mind we associate with data science offer another approach to mathematical understanding—and another perspective is always good. A data-rich approach like this can reinforce understanding; help students see how principles apply in many contexts; and for some students, it may unlock math topics that have eluded them in the past.

But to accomplish this, you need the right games. This paper will address questions about designing these games for learning, and will reflect on my use of Data Games materials in a statistics class in a high school in San Francisco, California. I will then attempt to draw some conclusions and suggestions for further development and investigation.

THE BIG ISSUES OF DATA GAME DESIGN

Here is the vision: Students want to do well in games, so if data analysis helps them do better, they will want to learn how to analyze the game data. We hope this experience in data analysis transfers to data analysis outside the game context. This leads to some important requirements:

- Data analysis has to be *useful* in the game, so that analysis really improves your performance.
- Furthermore, data analysis has to be *easy*. If it's hard, it won't be worth the effort just to do well in the game.
- The games themselves have to be *quick*. We're in a classroom, after all; whatever we do—the play and the data analysis—has to happen in minutes, not hours of play.
- The games have to integrate easily into classroom life.

Note how these requirements imply that the games are small-scale by nature. We can imagine many larger, richer settings where students can be data scientists, ranging from real-life projects to analyses of complex games to MMPORPGs where data analysis plays a central role. But that's not what we're about. It should be easy for many teachers to use Data Games without a lot of preparation, and when they do, almost all students will learn and succeed.

Making data analysis useful is the hardest of these requirements; it's where the "art" is in the game design. But the games do not stand alone. The games are embedded in a web-based data-analysis environment (currently called "DG") so that the data collection is automatic and so that it is easy to make graphs and do calculations. Also, the activities themselves—the student instructions, for example—are designed and presented separate from the games. That is, the games do not teach data analysis: they are an environment in which data analysis is useful.

This will become clearer with an example. But first, a few words about whether the games are for teaching mathematics in general, or statistics in particular.

STATS OR MATH?

The Data Games project itself, at the high-school level, is really about incorporating data science into the *mathematics* curriculum. That is, it's related to statistics through data, but it addresses issues that do not appear in (i.e., are beneath) traditional statistics curricula.

Yet they are appropriate in my particular course. Regular Statistics at Lick-Wilmerding High School in San Francisco, California is taken by students in the last two years of their high school careers. Students in this class have generally not had great success in secondary mathematics. Since it has no prerequisites, any student who does not score high enough to move on is likely to take this course.

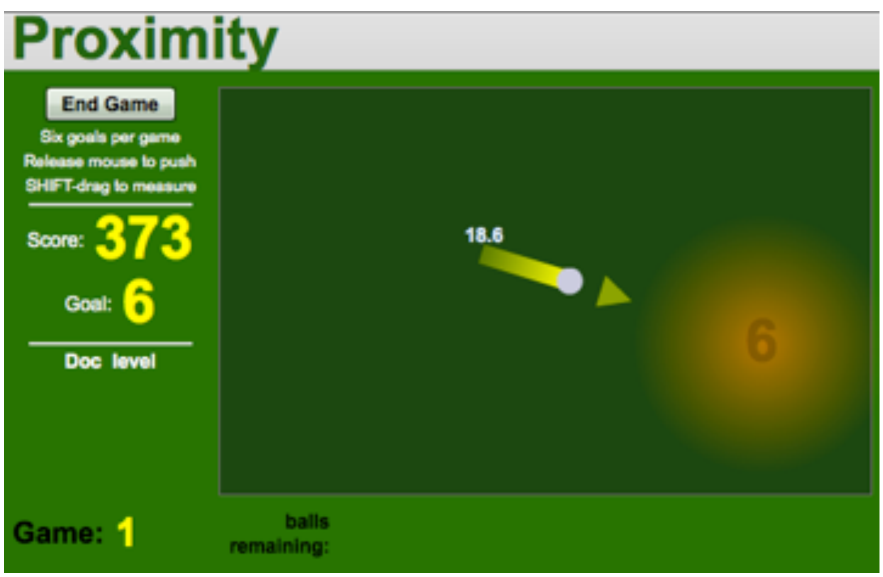
As a consequence, we need to spend substantial time in the course reinforcing important mathematical concepts and skills, for example, fractions, formulas, direct proportion, linear relationships, and reading graphs (Ben-Zvi and Garfield 2004). Although these are not traditional topics in a statistics course, they are the foundation for many basic statistical ideas as well as essential understandings for secondary mathematics. On a more fundamental level, these students need concrete experience finding meaning in graphs and understanding how multiple representations of data correspond to one another.

So in this paper, we will begin by focusing on topics like these, partly because they are so useful for these students, but also because they show how we can teach in a data-rich manner to students in other classes.

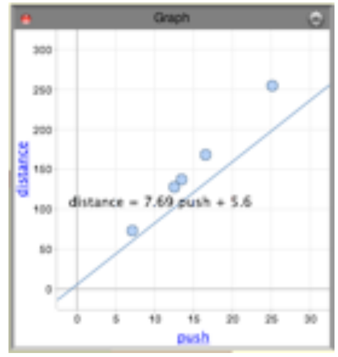
These games are also useful for teaching more clearly *statistical* topics as well, for example, coping with variability. We will return to that prospect late in the paper.

EXAMPLE GAME: PROXIMITY

In *Proximity*, students try to propel a white ball to the middle of an orange target. The closer you get to the center of the target, the higher your score. A "bullseye" is worth 100 points. You get 6 balls per game, and the target moves after each shot. There is a tool to measure distances on the screen.



Students learn to "drive" the interface quickly, and seem to intuit the object of the game. If you ask, "how did you know how hard to push the ball?" they know that to go farther you have to push harder. But their understanding is not quantitative. So we encourage them to make a graph. With the click of a button, they make a scatter plot like the one at right.



They can easily put a movable line on the graph (and even lock its intercept to go through the origin if they have that insight) and find its slope. They measure the distance before they shoot, and divide by the slope, to figure out how hard to push the ball.

Just playing "by feel" you can get scores over 250 without much trouble. Using data analysis, students can beat 450 reliably.

So: data analysis improves your performance. It's easy because the data are automatically collected in the DG environment. It's a quick game, short learning curve, six balls and you're done. You can get to a computer lab, play many games, analyze, debrief, and be back in an hour.

The astute reader might object that what I have described does not require very much mathematical understanding. True. Let's address that issue as we discuss the game design.

PROXIMITY DESIGN DISCUSSION

Let's look at some game-design issues as they relate to *Proximity* and to some other games.

But first, we must report a potentially important observation worthy of further study, or at least attention in design: In general, students are ingenious at avoiding math. They're not being contrary or lazy; it's just that when they're playing a game, even riveted by one, most fail to use the tools we have built. For example, they will write down—on paper—the data they need when the screen shows every move in a data table. They won't use a graph to help them determine what to calculate even when the line—and its equation—are right there on the screen.

Here are two conjectures for why: First, it's math class and they're used to writing down numbers and looking for patterns. Second, there is an "activation energy" to using a tool like a graph; if they have a familiar technique (looking for patterns in a table of numbers) they'll prefer to use it even if the new one (a graph) would save them time, confusion, and agony.

How do you design a game so that students actually use math, and its tools, on purpose? Let's look at one approach; we'll see how field-testing positively impacted the specifics of *Proximity*'s design.

Making the Data Essential

It is hard to design a game where data analysis is *useful*. If students can succeed in the game just playing it "by feel," it has failed as a data game.

In the first level of *Proximity* ("Doc"), students learn from the graph (or from a table) that the distance you travel is always close to 10 times the "push" you give the ball. When students succeed, we know they can measure distance and divide by 10—but with experience (or class chatter) everyone can do well without really using data to determine a strategy. That is, "Doc" is a good introduction, but we don't really know if students can come up with a winning procedure from scratch.

For that, we need a second level (called "Sneezy"). In the Sneezy level, this slope—the coefficient in the distance-push relationship—changes every game, that is, every set of six balls. You can't simply divide by 10; you have to look at each game anew and deal with its data.

Making this work is not trivial, however. It took repeated classroom field-test sessions to see what students actually do. Simply making the slope vary was not enough; at first, students reverted to playing by feel, so we implemented a system of requirements—you had to get a particular score on a level to move up to the next. That way, an activity could have as its goal to pass through Sneezy and get to the *next* level ("Dopey").

This was too onerous; too frequently, the ball would get caught near an edge, and it was too hard to hit the target. Students tried to use the data but became discouraged. Lowering the required score, however, invited playing by feel again. So we changed the game so that the interesting and challenging edge shots (alas) no longer appeared—and raised the score requirement.

Still, some students would get lucky, and by chance the slope they had to figure out was near what it had been before; or they just guessed well. So getting a high score was mechanical: measure the distance, divide by some number. They didn't need to look critically at their data. So we changed the game again: now you had to beat a score (425) *twice in a row* to advance, and the software guaranteed that the slopes would be substantially different from 10 and from one another.

(Why 425? You need the first ball to determine the slope; if that gives you no points, you need an average of 85 on each remaining ball to make the goal. That seems the right level of difficulty.)

This whole dance may seem obvious in retrospect, but it points out the vital importance of field test.

Skill and Luck

Now consider the balance between skill and luck. Of course, we lean towards demanding (mathematical) skill, but luck plays two important roles. The first is when the mathematical lesson is about probability. One of the games, *Wheel*, is a roulette-like game designed to teach students about the law of large numbers and expected value. So some students will be lucky, and that's part of the point. But luck also creates interest; if you don't know exactly what's going to happen, you're more likely to pay attention. Some more advanced students can be frustrated that despite understanding everything perfectly, they cannot get a perfect score, but they are the exception—and not the population of our greatest concern.

Luck comes with variability. If everything is completely determined, it's uninteresting. Furthermore, we want students to learn to cope with variability. But variability comes in many forms in a data game. In *Proximity*, for example, there are several sources:

- At higher levels, the relationship between push and distance has a little randomness in it.
- Not all "push" values are possible. As you prepare a shot, the mouse pointer's position is quantized to the nearest pixel; the internal relationship between those pixels and the push means that the value jumps in discrete steps. You can always get close, but you seldom get the precise value you want.
- There is variability in how you fit a model to the data, so calculations based on models will give varying results.
- There is variability in how well players aim at the target; if you're off by a few degrees, you'll miss the center of the target and cannot get the highest possible score.

This last source is particularly interesting. It has at least two drawbacks, though: first, it rewards game-playing experience, something we are loath to do (likewise, none of our games have a "twitch factor" that rewards fast reactions). Second, and more subtly, if students got a lower score simply because of poor aim, we would have to lower the threshold for success. Once we do that, it's easier for someone who does not really understand the mathematical model to succeed.

So in the case of *Proximity*, based on field test experience, we changed the game so that if you point *close* to the direction of the target, it automatically points the right direction. Now a student who makes a good model and understands it can get a vey high score—subject to the pixel-quantization problem we mentioned earlier.

This is an example of a long-standing tension in designing technology for learning: when do you make the student do things themselves, and when do you step in and do it for them? In this case, aiming perfectly is not an important learning outcome, so we take care of it—in a way that many users will not even notice.

GENERAL DESIGN CONSIDERATIONS

The lessons of design in *Proximity* apply to other games. Let's look at more.

Topics

It's not clear which topics are really amenable to treatment through a data game, but as we have gotten more experience, we are finding more approaches.

There is also another tension here: which comes first, the topic or the game? We'd like to think that the math topics drive the need for games, but realistically, it works both ways: sometimes, a great game idea is irresistible, and works to pull the associated topic into the light.

Simplify, Simplify

When we start to design a game, our tendency (and yours, we bet) will be to make it too complicated. Overcomplication takes two forms:

- The context may be too complex: To make a plausible context, we might invent a situation that takes too long to understand—so long that many students will be behind, or lost—or bored. *Why* are we trying to get the white ball in the gold blob? Is the white ball an antidote to a spreading poison? Whom are we trying to save? It's too much: a good context is wonderful, but sometimes it's OK simply to try to get a high score.
- The math may be too complex: To get at the meat of some realistic and interesting curricular topic, we might have to use more math than students can handle *in a short amount of time*. The original design for *Proximity* was more like miniature golf, where you had to bounce the ball to get around corners. Fun, but adding the math of bouncing (and the tools to measure angles) made things too complicated. So for this game, we stuck with direct proportion.

Our experience in classes and early field tests is clear: it is almost always best to remove complication and detail. The simpler the game, the better. To be sure, it's good to have something attractive in the game—in *Proximity*, the orange blob animates as you get close—but you don't need much.

Speed and Suspense

A non-math issue, but important: a computer game can, in principle, give your results almost instantly. But should it? No.

First of all, there is a game-aesthetic principle of creating suspense. This does not have to be very sophisticated to be effective; in *Proximity*, the ball moves towards the target, slowing as it goes. Will it make it to the target? Will it overshoot? We could have moved the ball immediately, but it would not be nearly as fun or engaging. Moving and taking time creates interest.

There is another reason to slow down: students easily fall into a pattern of pressing keys—filling in numbers and pressing return—as rapidly as possible. The (un)reasoning seems to be: if you can get a high score by playing five games mindlessly in the same time it would take you to play one game thoughtfully, you should go for it. We therefore deliberately slow the games down, so there is a greater benefit to thoughtfulness.

This is also one of the motivators for *automation*, addressed in the next section.

THREE PHASES OF A DG ACTIVITY

A lesson based on a game must be more than simply playing the game. We have found that lessons that seem to work best have three phases: an introduction, a time for play and basic data analysis, and then some sort of consolidation phase.

For the introduction, it is usually sufficient to tell the students to play. If the game is well-designed, they will learn the game mechanics automatically, uncovering the few subtleties with a little practice and exploration. We have also created short instructional videos, one to introduce each game. This introductory phase also gives students a chance to get some of the initial distraction of play out of their system.

This phase can take place in the classroom, or can be assigned as homework. In Proximity, this might involve getting a score of 300 or better, or unlocking the second level; in order to do that, they have to have learned how to shoot the ball at the target and possibly to measure distances.

In the second phase, we challenge students to use data analysis to improve their performance. Put another way, they transition from simply playing the game to using math as they play. Students do not generally do this on their own; this requires some direction from the teacher or the video, and some help using graphs and other analysis tools. In *Proximity*, for example, some students need to be prodded to look at the graph and figure out an appropriate push based on the equation of the line. Often a question like, "how did you decide to use that amount of push?" will help students move in the right direction.

In the classroom, this generally takes the form of students working in pairs to accomplish some goal such as unlocking a higher level. We often write hard-copy or online worksheets to accompany the play, asking questions that probe both the basics of the game mechanics ("What's the maximum score you can get in this game?") and more sophisticated issues ("Describe how you use the graph to help you know how hard to push the ball?").

After students have done this, they need to consolidate and solidify their learning. This is the third phase, which can take several forms. Although you can help students consolidate their learning by asking them to reflect, or write about what they learned, we like the option of a more performance-based approach, and ask, what could students *do* to make their learning more explicit and apparent?

One intriguing answer made possible by technology is to have students automate the process of playing the game. That is, teach a “bot” to play and win. This has two plusses:

- It takes the students out of simply playing and forces them to be explicit about their strategies. Furthermore, this often involves encoding their ideas in symbolic mathematics.
- It relieves the tedium of an artificially-slow game. We speed up automated play, which seems to be a reward to students.

There are design challenges here: how do you get students to teach a bot to play without programming? We’re working on that; one answer is to restrict the types of strategies students can employ. In low levels of *Proximity*, for example, where the results are deterministic—there is no variability—we could stipulate that the first push is a small fixed value, which you can use to completely determine the slope. Then on balls 2 through 6, you use the that slope, and the distance to the current goal, to determine the push value for each ball. This requires students to enter a single formula rather than writing a more general program. (In one field test, a pair of ingenious year-7 students decided to start every game with a “push” of 1.)

We have implemented player strategies most successfully in the game *Markov*—a version of rock-paper-scissors—where there are only three choices available for every move, so students simply push buttons to specify their choices for each of the 9 possible game situations.

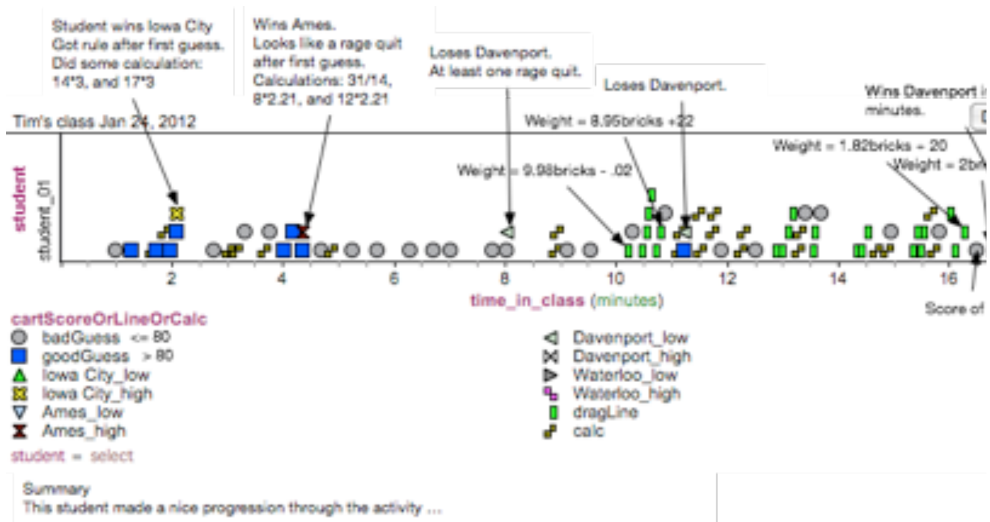
ASSESSMENT AND DATA GAMES: THE LOGS

If students play some games, how do you know they learn anything? Ordinary assessment practices work fine; for example, you can give students tasks ranging from proximal to distal (Ruiz-Primo et al., 2002) to see what they understand about the math topics in the game. But in this short paper I want to focus on *game logs*.

Because the games are played on the Web, we can, in principle, record everything the students do. Taking advantage of this opportunity, we have the games and the DG environment “log” certain student actions. We see their every move, their games and scores; when they unlock new levels; when they make graphs. We can see the equations of the lines they put on graphs and the calculations they make with the in-game calculator.

Let’s look at an example of a prototype graph showing a student’s work on the game *Cart Weight*, with annotations. In this game, you guess the weights of five carts that have different numbers of bricks on them. In the first level, “Iowa City,” the cart is weightless and each brick weighs 3 units. Most students see the pattern immediately. In the second level, “Ames,” the cart weighs 8 and each brick weighs 4, so students have to deal with the intercept.

In the “Davenport” level, however, the slope and intercept change every game. This is a challenge, and requires dealing with your data efficiently. Here are extracts from one student’s log:



The student masters "Iowa City" in two minutes. It takes two minutes more to master "Ames." But then there are many poor guesses as the student puzzles out Davenport (minutes 5–10, making a few calculations). Finally, the student makes a graph, and we can see that they put a line on the graph and adjust it repeatedly in minute 11. Just after that, they make their first good guess in Davenport—but not enough to get a high score for the whole game.

Between 12 and 14 minutes they make many calculations and adjustments, but make four bad guesses. Realizing they have a low score, they "rage quit" and start a new game before that game ends. At minutes 15 and 16, they make the first two guesses for the next game—poor guesses, but thats OK: they needed two points to determine the line. At 16:30, they have made a line with the equation **Weight = 1.82 * bricks + 20**, and get a good score of 76 on the next guess. They then adjust the line to **Weight = 2 * bricks + 17**, and ace the last two carts, giving them a high score for the level.

This period of time—a scant 20 minutes—constitutes the first phase and most of the second of a plausible activity about linear relationships. *Cart Weight* has no "automation" mode yet, so the third, consolidation phase was a paper handout where, among other things, students wrote about the meaning of the numbers in the equation for the line (the slope is the weight of a single brick; the intercept is the weight of the cart) and wrote instructions for how to beat the Davenport level.

These logs give us an intriguing window into a student's thinking. We can see early success, then struggle, then bringing tools to bear, and finally success on a more challenging task. We hope to connect logs like this one with other artifacts and observations, and with logs from later sessions. This can even help us assess the elusive "habits of mind" we would like nascent data scientists to develop. For example, do students turn to graphs more quickly in subsequent sessions?

DATA GAMES FOR STATISTICS

In this paper, we have been focusing on using Data Games for what is essentially *mathematics* instruction as opposed to *statistics* instruction. But you can use Data Games for stats as well. After all, the games are just sources of data; all we have to do is come up with game situations in which understanding statistical ideas improves your performance in the game.

We alluded above to *Wheel*, a wheel-of-fortune sort of game, where observing for long enough will reveal the biases in the wheel, and an understanding of expected value will show you the way to a moneymaking strategy.

Even more of these have been created by the Data Games group in Amherst, MA, USA, especially for middle-grades students, and address issues such as understanding variability (e.g., Shaughnessy et al, 2004). These include *Ship Odyssey*, a treasure-hunt game where you get imperfect information about the location of a treasure. How much imperfect information do you need before you're confident that your "hook"—which has a finite extent—will snag the treasure? In such games, players balance limited resources (in this case, samples against hooks) in order to maximize their “take”; it’s all about understanding how sample size affects how well you know the underlying population mean.

Another game, *Rock Roll*, explores issues behind experimental design. One can imagine endless variations. We can make a game where stratified sampling, for example, gives better results than a SRS. The art will come in deciding exactly what the point of such a game is, and tuning the game so that an understanding of sampling really does make a difference.

Sources of Variability

At this point, I’d like to spend some time expanding, from a design point of view, on the topic of luck and variability that we discussed about three pages ago. Variability, from whatever source, can make a game more interesting. How does a game’s role in *statistics* education affect how we create variability?

Wheel is straightforward: it’s a gambling game, and there’s randomness. The variability is central and authentic.

In higher levels of *Proximity*, as we alluded to above, we add in a little noise to the distance a ball travels. Students have to cope with the notion that the line may not go through all the points as it did in lower levels (and as it generally does in math textbooks). It also means that the strategy of using the first point to compute the slope is insufficient; a data-aware strategy would be to recompute the slope each time to be the *average* distance divided by the *average* push.

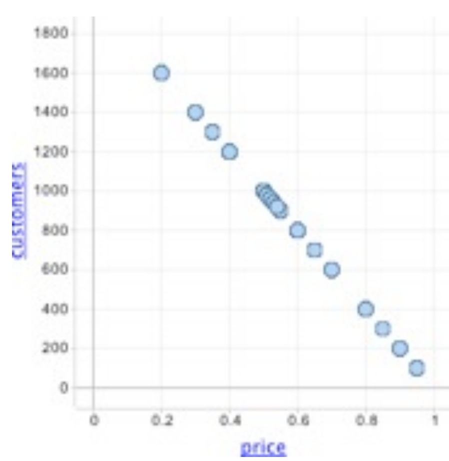
In *Ship Odyssey*, there’s an underlying story: you send specially-trained rats down to find the treasure. After they find the treasure, they swim back to the surface. Alas, the turbulent waters force them off course in a random fashion, creating a distribution of rat-surfacing locations; you use those to decide where to drop your grappling hook. That is, the “rat” feature in the story introduces noise. The advantage is that it makes the story is fun and engaging; the disadvantage is that the rat narrative may get in the way of understanding the data—or at the least, it takes time to explain.

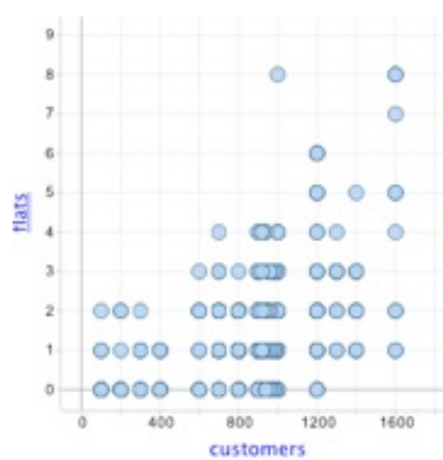
Another game with statistics content is *Floyd’s of Fargo*. This is an insurance game; you’re insuring cars against flat tires (like the rat narrative, this takes some time to explain). Your job is to set the premium. A new tire costs \$100. The lower the premium, the more people will buy your insurance—but the more you’ll have to pay out in claims. Your goal is to make as much money as possible.

There is an optimum price, and you can find it empirically.

So where’s the variation? In the first version of Floyd’s, this was the situation:

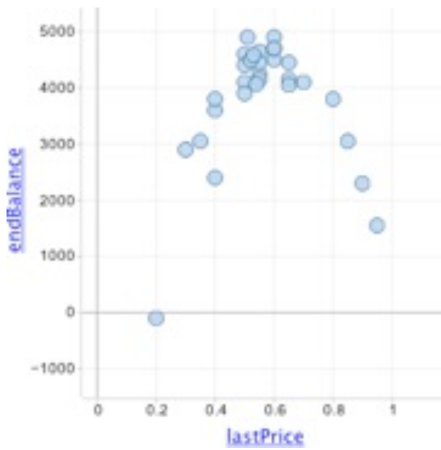
- The relationship between premium price and the number of customers was linear: the number N of customers was $N = 2000(1 - P)$, where P is the premium price in dollars. That is, if you give it away, you’ll have 2000 customers, but if you charged a dollar or more, no one would buy. (Left-hand graph, below.)
- The number of flats your customers had—which determined your costs—was calculated probabilistically: the chance that any one car would get a flat was 0.002.

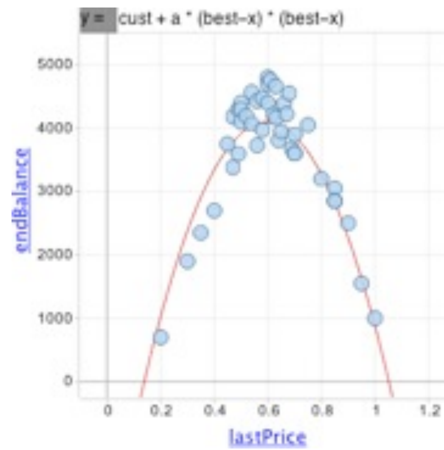




Thus while the number of customers was deterministic, the payout had variability that arose naturally. Students could "watch" cars to try to determine that unknown probability. If they understood expected value and did some algebra, they could find the theoretical quadratic relating profit to premium price and find its vertex.

But if not—the usual case—the graphs they got would have interesting variability, and they could still fit a curve to the data. Using sliders for the parameters (in the graph below right, they're **cust**, **a**, and **best**, and the formula is in vertex form) they can do this visually without resorting to automatic fitting or to transformations (Erickson, 2008).





Students can therefore play this game on many levels. It can be a modeling game, where you find the appropriate graph, fit a parabola, and realize which parameter gives you the largest profit. Or it can be a rich optimization problem. But you can also study the nature of the underlying probability model and see how the number of flats varies with the same number of customers. In addition, the game gives you a glimpse into other statistical issues, for example, heteroscedasticity.

We are now planning an easier level where the number of flats is (unrealistically) deterministic, and a harder level where the number of customers is governed by some stochastic model. Our model for the flats also assumes independence; breaking that assumption (suppose the flats come in clumps) could make for even more complicated analysis—but for now, we keep it simple. Students have enough to cope with just understanding what that graph with the parabola means.

The lesson, though, is that statistical issues appear in these games in different ways, through the deliberate introduction of noise or through the choice of phenomena where sampling and combined events create variability naturally. All of these work, and all of them present students with data that vary, very much like data they will encounter in real life.

I suggest that one could develop Data Games purposefully to address specific topics in statistics education, for example, the seven areas of variability compiled and elucidated by Garfield and Ben-Zvi (2005); and we could use the learning analytics tools described above—instead of or in addition to test items—to assess student understanding of these in a "performance" context.

CONCLUSION AND DIRECTION

Playing a Data Game (like most instruction) does not create understanding. But we have designed Data Games and the associated activities to be a good opportunity for developing and solidifying understanding of mathematical ideas. Likewise, success at a Data Game is not ironclad evidence of understanding or of a well-engrained habit of mind, but it can be a useful tool in a teacher's assessment arsenal.

The logs are a particularly exciting windfall from this project. We have never before been able to see so many students' progress in such detail. Could we get access to them in real time, and use them to make instructional decisions on the fly? That's one direction for future work.

Then there's the question of the breadth of topics these small games might encompass. We encourage your idea and participation as we come to understand what's possible and move forward to design new games and activities.

Avid readers interested in the project should visit http://www.kcptech.com/datagames/.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under: KCP Technologies Award ID: 0918735 and UMass Amherst Award ID: 0918653. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

The author gratefully acknowledges the helpful comments of anonymous reviewers.

REFERENCES

D. Ben-Zvi, & J. Garfield, (Eds.). 2004. *The challenge of developing statistical literacy, reasoning, and thinking*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

T. Erickson. 2008. "A Pretty Good Fit." *The Mathematics Teacher*. 102(4), 256–262.

J. Garfield and D. Ben-Zvi. 2005. "A Framework for Teaching and Assessing Reasoning about Variability." *Statistics Education Research Journal*, 4(1), 92-99, http://www.stat.auckland.ac.nz/serj

M. A. Ruiz-Primo, R. J. Shavelson, L. Hamilton, and S. Klein. 2002. "On the Evaluation of Systemic Science Education Reform: Searching for Instructional Sensitivity." *Journal of Research in Science Teaching* 39(5), 369–393.

J. M. Shaughnessy, M. Ciancetta, and D. Canada. 2004. "Types of Student Reasoning on Sampling Tasks." *28th Conference of the International Group for the Psychologgy of Mathematics Education*. Vol. 4, 177–184.