HIERARCHICAL DATA VISUALIZATION AS A TOOL FOR DEVELOPING STUDENT UNDERSTANDING OF VARIATION OF DATA GENERATED IN SIMULATIONS

William Finzer
Concord Consortium, USA
wfinzer@concord.org

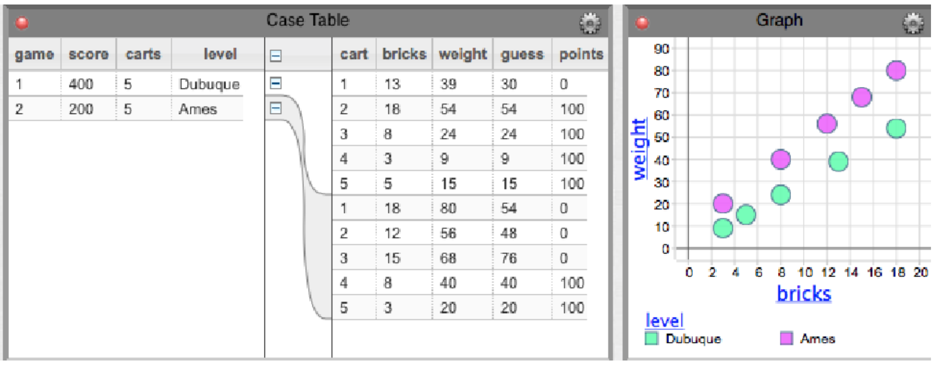
Data Games is a data exploration environment in which data generated by playing simple games is used by students to build models with which they can improve their game-playing strategies. The data is structured hierarchically, with games at the upper level containing game events at the lower level. Visualization of these two levels, typically with multiple graphs linked by dynamic selection, reveals patterns and variation among the games and illuminates the role that chance plays in the games. We explore the parallels of Data Games to repeated sampling simulations and in a NetLogo forest fire simulation. We look at uses of hierarchical data visualization techniques in each of these three situations.

INTRODUCTION

Repetition may yield unvarying sameness, or a trend, or randomness, or a co-varying relationship. Repetition underlies the search for truth; a single observation or measurement begs for company. Repetition takes place at multiple levels, from the finest grain measurements of an instrument's voltage, to large scale replication of a drug trial. This is a paper about imposing structure on repetition in the service of understanding variation.

Lehrer (2009) describes students engaged in repeated measurement of physical objects, such as the circumference of a person's head, and the process by which they made sense of variability: "In summary, although variability is at first glance an antonym of structure, a view from the mathematics of chance suggests instead that variability reflects a structure, however initially obscure." Konold and Harradine (2014) in their discussion of students' measurements of objects they "manufactured" from Play-Doh show how students make use of the visualization capabilities of TinkerPlots (Konold, 2012) to tease apart the structure of these measurements. Heer's (2012) taxonomy of ways to interact dynamically with data visualizations shows what is currently possible. There is a strong interaction between learners' efforts to make sense of data and the affordances of the technology they are using to manipulate that data.

THE GAME OF CART WEIGHT



Case Table

game	score	carts	level
1	400	5	Dubuque
2	200	5	Ames

cart	bricks	weight	guess	points
1	13	39	30	0
2	18	54	54	100
3	8	24	24	100
4	3	9	9	100
5	5	15	15	100
1	18	80	54	0
2	12	56	48	0
3	15	68	76	0
4	8	40	40	100
5	3	20	20	100

Figure 1. The data from two games of Cart Weight. The graph shows that the relationship between bricks and weight changes from one level to the next.

Consider Cart Weight (Finzer, 2014), a simple game in which each move consists of trying to guess the weight of a cart loaded with bricks. As you play, a record of your moves flows as data into a surrounding data analysis environment where you can explore the data in tables and graphs that you construct. Figure 1 shows a table a graph as they might look after two games of Cart Weight.

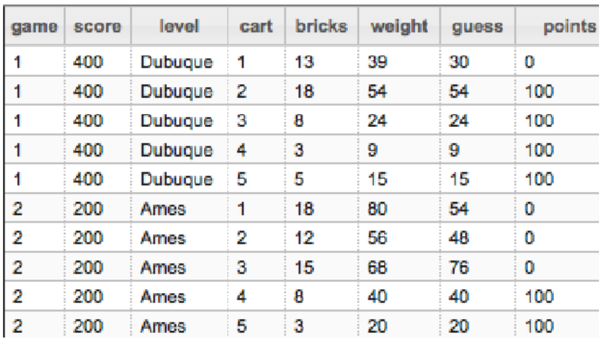
In K. Makar, B. de Sousa, & R. Gould (Eds.), *Sustainability in statistics education. Proceedings of the Ninth International Conference on Teaching Statistics (ICOTS9, July, 2014), Flagstaff, Arizona, USA.* Voorburg, The Netherlands: International Statistical Institute. iase-web.org [© 2014 ISI/IASE]

The table in Figure 1 lists the games in its left panel. The turns appear in the right panel, grouped with their respective game by curved lines. The games and moves constitute a hierarchy with games at the top level and moves at the next lower level. We say that each game *has* five moves, and each move *belongs to* one game.

This paper is about visualization of the structure of hierarchical data and how such visualization can aide students in understanding variation at each level of the hierarchy.

COMPARING HIERARCHICAL AND FLAT

Most raw data that students encounter through grade 14 or so is presented in the rows and columns of tables. Each row contains a single *case*. Each column contains *values* belonging to one *attribute* of the cases. In the header of each column is an *attribute name*. Such a data representation is called "flat" because it fits easily in two dimensions, one for rows or cases, and a second for columns or attributes. Typically, there will be only two or three attributes, the ones deemed important for the problem students will investigate, and the entire table will fit on a textbook page.

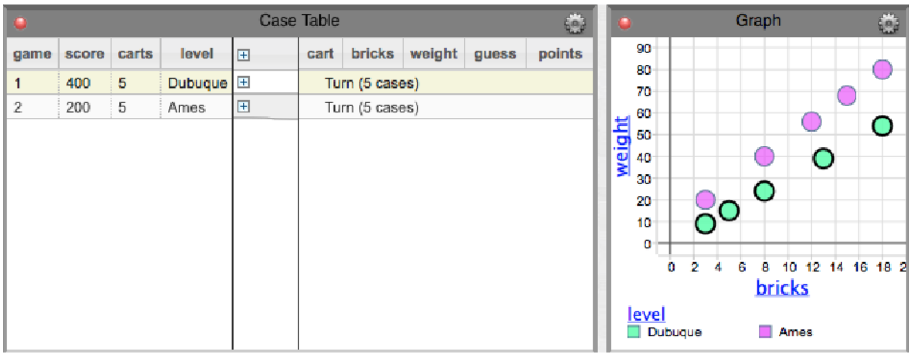


game	score	level	cart	bricks	weight	guess	points
1	400	Dubuque	1	13	39	30	0
1	400	Dubuque	2	18	54	54	100
1	400	Dubuque	3	8	24	24	100
1	400	Dubuque	4	3	9	9	100
1	400	Dubuque	5	5	15	15	100
2	200	Ames	1	18	80	54	0
2	200	Ames	2	12	56	48	0
2	200	Ames	3	15	68	76	0
2	200	Ames	4	8	40	40	100
2	200	Ames	5	3	20	20	100

Figure 2. A restructuring of the Cart Weight data from Figure 1
so that it fits in a flat, row-by-column table.

We can flatten the game data shown above with the result shown in Figure 2. There are some clear disadvantages compared to the hierarchical structure: (1) The repetition of values in the first three columns is distracting. (2) Each row, instead of being either a game *or* a turn is now a combination of row *and* turn.

A simple visualization of the hierarchical structure has several affordances that help the user make sense of the data. Figure 3 illustrates these.



game	score	carts	level	cart	bricks	weight	guess	points
1	400	5	Dubuque	Turn (5 cases)				
2	200	5	Ames	Turn (5 cases)				

Figure 3. The turns are collapsed and the first game is selected in the table
causing points from that game to be selected in the graph.

- Selecting a row that represents a game causes all turn level data to be selected also.
- The turn level data can be selectively collapsed to allow focus on just the games or on the moves from a particular game.
- The number of points in a graph of game level attributes will equal the number of games rather than the number of turns as it would when made from flat data.

VARIATION

The values of an attribute vary from one case to the next. This variation may be systematic—the value of *game* increases by one for each new game. Or it may be *determined* by the game or simulation or user—the Cart Weight game chooses a new number of *bricks* for each turn. Or the variation may be partially or completely *random*—the value of *weight* may have a linear relationship to the value of *bricks* plus a certain amount of noise.

We can be interested in variation at the game level. What is the distribution of scores for all the games? Or, assuming we compute the linear relationship between bricks and weight for each game, how does the slope or intercept vary within a game level or from one level to the next?

MODELING THE SIMULATION OF REPEATED SAMPLING AS A HIERARCHICAL PROCESS

In place of playing a game, substitute repeated sampling to build up a sampling distribution of some quantity measured for each sample. Figure 4 lays out a mapping from one data game to simulation.

Data Game		Repeated Sampling Simulation
Game	—>	One sample
Move	—>	One value in a sample
Score	—>	Quantity measured for each sample

Figure 4. A map of elements of a data game onto elements of a repeated sampling simulation.

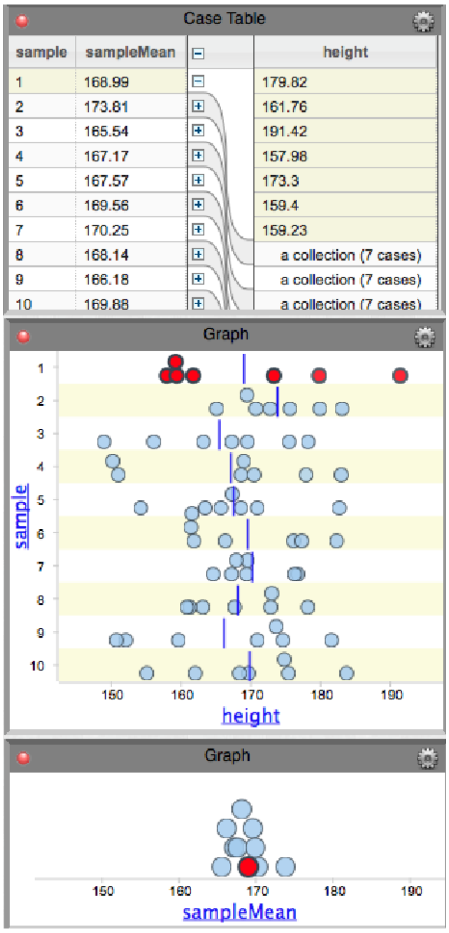
Figure 5 shows an example of results from a repeated sampling simulation.

The hierarchical structure of the data is clear in the table where each row in the left panel represents one sample expanding to individual values belonging to the sample in the right panel. The graph below the table displays dot plots for each of the samples, again reflecting the hierarchical structure of the data in that the number of each dot plot and its vertical line for the sample mean belong to the top level (parent) level of the hierarchy, and the individual points belong to the bottom (child) level.

The bottom graph plots just the sample means computed for each sample; it is the sampling distribution the simulation is designed to create.

Understanding the role of the data elements in this representation is helped by being able to click on rows in the table and points in the graphs. Figure 5 shows that clicking on the first sample row selects that row, all the rows corresponding to its sample values, the points in the middle graph corresponding to those values, and the point in the sampling distribution corresponding to the mean of the sample values. This dynamic linking of selection can be a powerful aide to puzzling out the role of a sampling distribution in statistical inference.

The hierarchical structure of the data gathered in the simulation reflects the sources of variation: At the lower level the variation in values (the shape of its distribution) comes from the population while at the upper level the variation manifest in the shape of the sampling distribution is intrinsic to the sampling process itself.



Case Table

sample	sampleMean	⊟	height
1	168.99	⊟	179.82
2	173.81	⊞	161.76
3	165.54	⊞	191.42
4	167.17	⊞	157.98
5	167.57	⊞	173.3
6	169.56	⊞	159.4
7	170.25	⊞	159.23
8	168.14	⊞	a collection (7 cases)
9	166.18	⊞	a collection (7 cases)
10	169.88	⊞	a collection (7 cases)

Figure 5. A hierarchical representation of the data resulting from a repeated sampling simulation.

THE NETLOGO FOREST FIRE SIMULATION

In the forest fire simulation (Tisue, 2004) shown in Figure 6, the forest is initially a square of pixels each with a certain probability of being green for a tree or black for ground. When the simulation starts, each of the pixels along the left edge turns red, indicating that it is a burning tree. For each tick of the simulation, green pixels with an edge touching a red pixel become red, and red pixels become brown to indicate a burned tree. In Figure 6, the simulation started with 0.59 probability (density) of trees and has so far burned 20.9% of the forest.

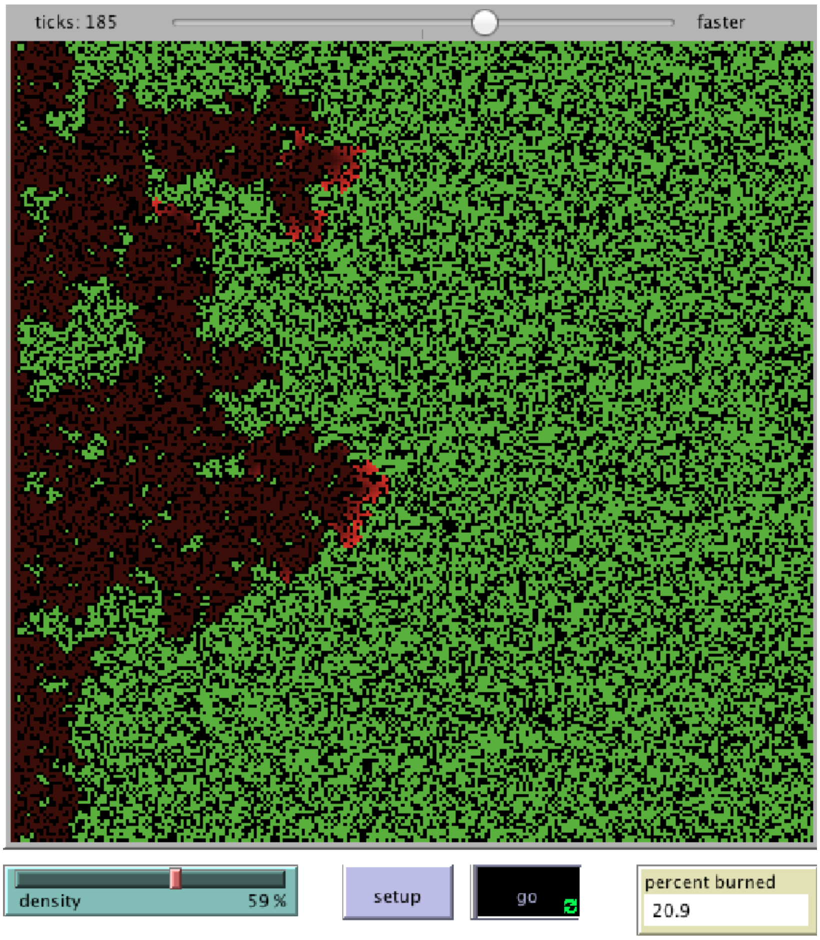


Figure 6. The NetLogo forest fire simulation partway through a burn that began at the left edge

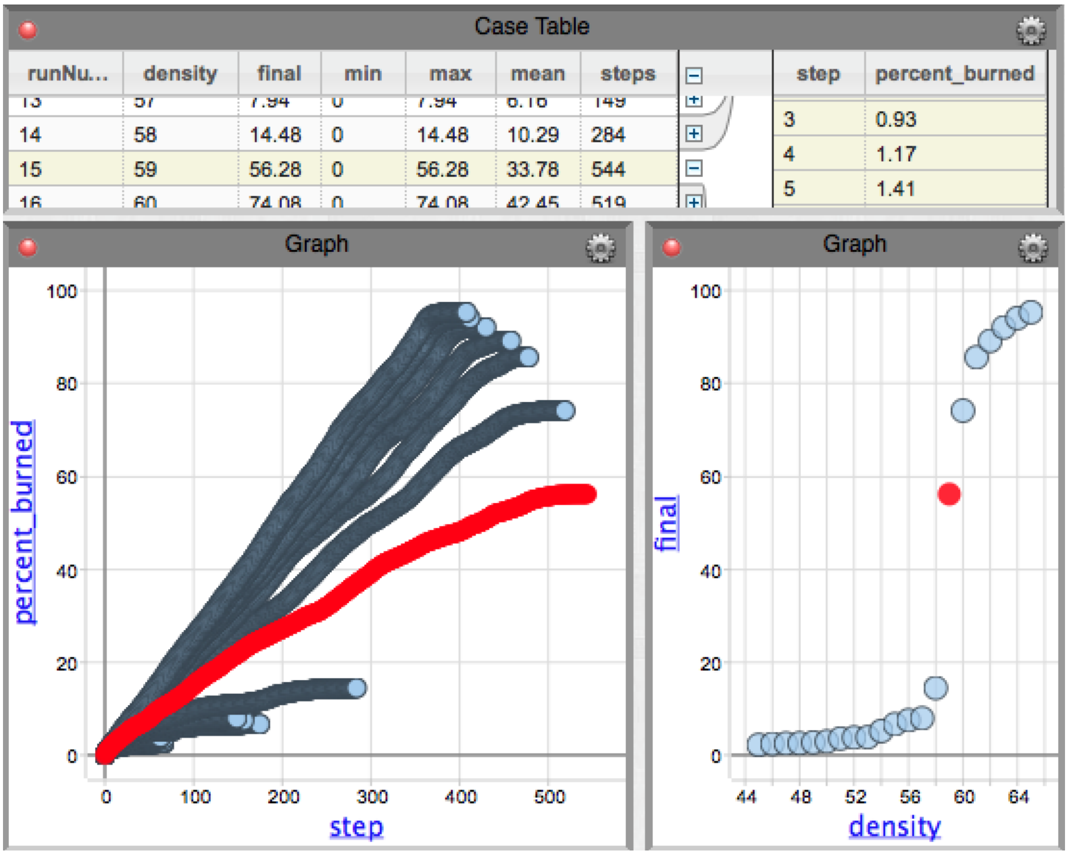
This simulation has one parameter, the density of trees, and it makes sense to explore the behavior of the simulation as this parameter is changed. Figure 7 shows the data obtained by varying the density one percentage point at a time from 45% to 65%. Each of the curves that appear in the *percent_burned* versus *step* graph corresponds to run of the simulation. The *final* versus *density* graph at the parent level shows that there is a narrow transition zone in which fires go from burning less than 20% of the forest to burning more than 80%.

The hierarchical structure of the data allows exploration of the relationship between the *density* parameter and the shape of the curve in the *percent_burned* graph. For example, the fire at a *density* of 59% burns at a medium speed for about 500 steps until it slows to a halt during the last 50 steps. The variation of shape with *density* is not easily characterized in spite of its striking visual characteristics.

DISCUSSION

In each of the three situations described something is repeated—playing a game, taking a sample from a population, and running a simulation. Each repetition generates data. For each repetition one or more summary characteristics of the generated data are of interest: the score for the game, the mean of the sampled values, the final burned percentage. These characteristics constitute a dataset in their own right, one that derives from the repetitions. We have chosen to structure these data as a hierarchy so that each summary case is explicitly linked to the repetitions.

Software that encourages hierarchical structuring of data naturally lends itself to dynamic linking between levels, and this dynamic linking holds promise as a tool for increasing student understanding of the relationship between repetitions of a process and measurements made on those repetitions.



Case Table

runNu...	density	final	min	max	mean	steps
13	57	7.94	0	7.94	6.16	149
14	58	14.48	0	14.48	10.29	284
15	59	56.28	0	56.28	33.78	544
16	60	74.08	0	74.08	42.45	519

step	percent_burned
3	0.93
4	1.17
5	1.41

Figure 7. Running the forest fire simulation for a range of densities shows that there is a narrow transition zone around a density of 59%.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under: KCP Technologies Award ID: 0918735, UMass Amherst Award ID: 0918653, and Concord Consortium Award ID IIS-1147621. 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.

REFERENCES

Finzer W. (2013) *Data Games* [Web Application] <ccssgames.com>.

Finzer, W. (2014). Games, data, and habits of mind. In T. Wassong, D. Frischemeier, P. R. Fischer, R. Hochmuth, & P. Bender (Eds.), *Mit werkzeugen mathematik und stochastik lernen - using tools for learning mathematics and statistics.* Wiesbaden: Springer Spektrum.

Heer, J., & Shneiderman, B. (2012). Interactive dynamics for visual analysis. Queue, 10(2), 30.

Konold, C., & Miller, C. (2012). *TinkerPlots* [Computer Software]. Emeryville, CA: Key Curriculum.

Konold, C., & Harradine, A. (2014). Contexts for highlighting signal and noise. In T. Wassong, D. Frischemeier, P. R. Fischer, R. Hochmuth, & P. Bender (Eds.), *Mit werkzeugen mathematik und stochastik lernen - using tools for learning mathematics and statistics.* Wiesbaden: Springer Spektrum.

Lehrer, R., & Kim, M. -J. (2009). Structuring variability by negotiating its measure. *Mathematics Education Research Journal*, *21*(2), 116-133.

Tisue, S., & Wilensky, U. (2004). NetLogo: Design and implementation of a multi-agent modeling environment. In *Proceedings of agent.*