GROUPING AND THE POWER OF HIERARCHICAL DATA

Tim Erickson
Epistemological Engineering
eepsmedia@gmail.com

This paper explores what can happen if we organize data hierarchically instead of in a flat rectangular structure. We will see that, given a data analysis platform in which this is easy and natural, a hierarchical structure lets us perform many "data moves" (Erickson et al., 2019) dynamically and transparently. This is not limited to datasets that exhibit what we might call "natural hierarchy." With many datasets, we can use hierarchy to group and regroup, create summaries, filter data, and even recode categorical values. We conjecture that students who work with hierarchical data will better understand underlying data moves, such as grouping, and be better equipped as advanced students and as informed citizens.

We wrestle our data into tidy, rectangular formats. We insist on one case per row, one attribute per column. But what if that were no longer necessary? What if technology were more flexible? In this paper, we look in depth at an alternative to flat tables: *hierarchical* data. Hierarchical tables are not obscure or esoteric: Konold, Finzer, and Kreetong (2014, 2017) asked students to produce an organized record of the information in fictitious "snapshots" of cars on two roads. They found that a majority of students spontaneously created hierarchical tables to cope with the data.

What does hierarchical data organization entail, and what are its consequences for students and teachers? The Common Online Data Analysis Platform (CODAP) (The Concord Consortium, 2014) supports hierarchical tables, so it is a good tool to use in our exploration.

NATURAL HIERARCHY: AN EXAMPLE FROM THE US CENSUS

United States Census data has two types of records: *households* and *persons*. They are published in separate flat, rectangular files in a familiar structure: one household (or person) per row, and one variable per column.

A household record lists attributes such as whether the dwelling is owned or mortgaged, whether the location is urban or rural, and whether residents have internet access. Person records have data such as age, race, marital status, and income. We also know which household is associated with each person.

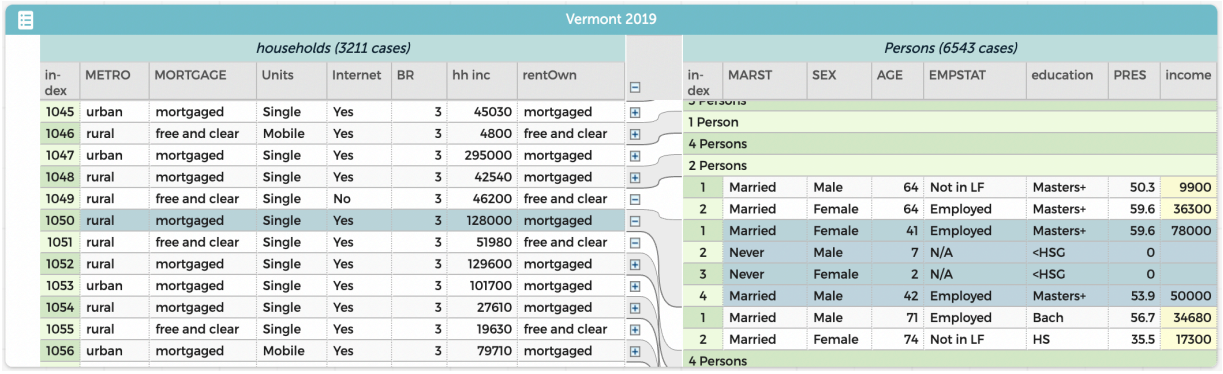
I call this a "natural hierarchy" because cases in the households table "contain" the cases in the persons table: the people that actually live inside the dwellings. You could use the same data structure with students in classrooms … or cars in different parking structures … or potatoes in different sacks. The containers have attributes of their own (room number, number of parking spaces, what the sack is made of). The cases inside these containers do not actually have those attributes—whether a house has internet is not an attribute of a person living there—but the `internet` attribute "applies" to the people within—and we can use it in further investigation.

This structure is essentially hierarchical. In this census data, households are at a higher, superordinate level and persons are at a lower, subordinate level of the hierarchy.

Figure 1 shows how a CODAP table displays this arrangement. The highlighted (shaded) cells show a single household of four people. On the left, you see data about the household: it is rural; they have internet; there are three bedrooms; the house is mortgaged. On the right, we see data about the four inhabitants, apparently a mother, a father, and two children. Both parents are in their 40s, have masters' degrees or more, and are employed, although the mother earns more, and her job has a higher "prestige score" (`PRES`). The two children, aged 2 and 7, not surprisingly, have never been married, have no income, and have not yet completed high school.

Considered separately, the two tables contain different units of analysis: households and individual people. On the left, you could calculate what fraction of households are urban. On the right, you could see what fraction of people are married. But looking at them together, you could answer new questions, such as what fraction of *people* live in urban areas or what fraction of *households* contain children.

In S. A. Peters, L. Zapata-Cardona, F. Bonafini, & A. Fan (Eds.), *Bridging the Gap: Empowering & Educating Today's Learners in Statistics. Proceedings of the 11th International Conference on Teaching Statistics (ICOTS11 2022), Rosario, Argentina.* International Association for Statistical Education. iase-web.org



Vermont 2019

households (3211 cases)

index	METRO	MORTGAGE	Units	Internet	BR	hh inc	rentOwn
1045	urban	mortgaged	Single	Yes	3	45030	mortgaged
1046	rural	free and clear	Mobile	Yes	3	4800	free and clear
1047	urban	mortgaged	Single	Yes	3	295000	mortgaged
1048	rural	mortgaged	Single	Yes	3	42540	mortgaged
1049	rural	free and clear	Single	No	3	46200	free and clear
1050	rural	mortgaged	Single	Yes	3	128000	mortgaged
1051	rural	free and clear	Single	Yes	3	51980	free and clear
1052	rural	mortgaged	Single	Yes	3	129600	mortgaged
1053	urban	mortgaged	Single	Yes	3	101700	mortgaged
1054	rural	mortgaged	Single	Yes	3	27610	mortgaged
1055	rural	free and clear	Single	Yes	3	19630	free and clear
1056	urban	mortgaged	Mobile	Yes	3	79710	mortgaged

Persons (6543 cases)

index	MARST	SEX	AGE	EMPSTAT	education	PRES	income
1 Person							
4 Persons							
2 Persons							
1	Married	Male	64	Not in LF	Masters+	50.3	9900
2	Married	Female	64	Employed	Masters+	59.6	36300
1	Married	Female	41	Employed	Masters+	59.6	78000
2	Never	Male	7	N/A	<HSG	0	
3	Never	Female	2	N/A	<HSG	0	
4	Married	Male	42	Employed	Masters+	53.9	50000
1	Married	Male	71	Employed	Bach	56.7	34680
2	Married	Female	74	Not in LF	HS	35.5	17300
4 Persons							

Figure 1. Hierarchical table showing household and person records from Vermont in 2019. The highlighted cells are a single household containing four persons, presumably a family. Data is from the American Community Survey. In the text values, LF stands for “Labor Force" and HSG stands for “high-school graduate.”

Summary Values

Notice also that the household income ($128,000 for our selected family) is the sum of the incomes of the inhabitants, calculated from the person data. That is, there are two kinds of data in a household record: data that are directly about the household (e.g., urban or rural) and data that are derived from the person records it contains. If we want to calculate some different quantity, we can do so. For example, we might write a formula that counts the number of children under 18 or calculates the average age.

Any of these calculated values is the result of some aggregate calculation, performed over the members of the household. Each household has its own value that depends on the people inside it. This is a simple idea, but has some important consequences:

- These summary values naturally belong at the higher level in the hierarchy. That is, a column for number-of-children should be in the households table, not the persons table.
- Formulas for these attributes will naturally draw on functions that aggregate data such as median (e.g., `median(income)`) or count (e.g., `count(Age < 18)`).
- You can analyze high-level data such as these aggregate values using the same tools you use to analyze the low-level data. For example, you can graph the distribution of household incomes just as you can graph individual incomes.

Pedagogical Comments

The idea that an aggregate value belongs higher up, that it applies to a group, is powerful and conceptually deep (Haldar et al., 2018). It is not surprising that students can sometimes be confused. In CODAP, students must also understand that when CODAP calculates a value using an aggregating formula, that formula “sees” only the subordinate cases contained by the case where the formula is being calculated. That is, the formula `count(Age < 18)` looks only at the ages of the people in “this” household when it counts those under 18.

Hierarchy also addresses a curious pedagogical problem with spreadsheets. If you have a table of people, with their incomes in a column, you can calculate the median income using a formula. But where do you put that calculation? One traditional place is at the bottom of the table—but if you do that, the table no longer contains only cases of individuals; it also has one row of summary information. Hierarchy gives you a reliable, unambiguous place to put that calculation—and avoid calculating a new aggregate value that mistakenly includes the previous summary.

Finally, the possibility of making new summary attributes opens up a world of investigations that draw on multiple levels of the hierarchy. Do houses with more bedrooms have more people living in them? What is the relationship between job prestige and whether a house is owned or rented? Such multi-level questions are important—and we rarely ask them. What does it take for a student to understand and address such a question successfully?

BEYOND NATURAL HIERARCHY: USING HIERARCHY FOR GROUPING

Organizing data hierarchically makes perfect sense in contexts like the Census. But what about in contexts where there is no "containment?" Suppose we have only the "persons" dataset and no "households"—is hierarchy of any use?

Suppose we want to compare income by education level. Any analysis platform has a way to group the data by education, perform parallel analyses, and compare the results. In CODAP, you would graph **`income`**, then drag the education attribute to the other axis. Under the hood, CODAP *implicitly* groups the data to make a split dot plot, where you can display the median for each group (Figure 2).

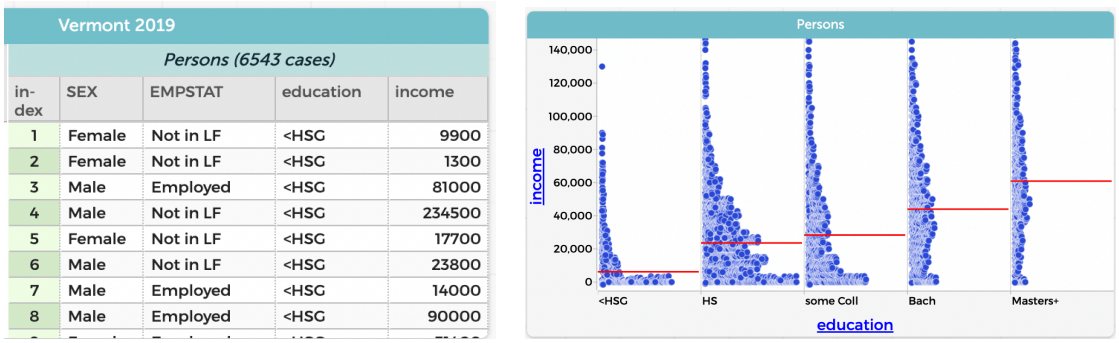


Figure 2. Incomes of 6,543 persons, split by education level. As we would suspect, the median incomes (horizontal lines in red) are higher for higher education levels.

But there is another, more explicit way to group the data. Dragging the education attribute to the left in the CODAP table *promotes* **`education`** in the hierarchy, making a new top level. Now each *category* of education becomes a single case on the left side of the table. Each one represents a group. On the right, the cases of individual people are sorted into those groups, visually connected to the groups on the left (Figure 3).

On the left-hand, group side, we make a new column **`medInc`** and calculate **`median(income)`**. The software calculates the median *over the people in that group*, that is, for everyone with that education level, and displays the medians in the table. We can then use the **`medInc`** column to graph those numbers separately, without all the individuals. By creating a hierarchy, we have created a new unit of analysis: an education level.

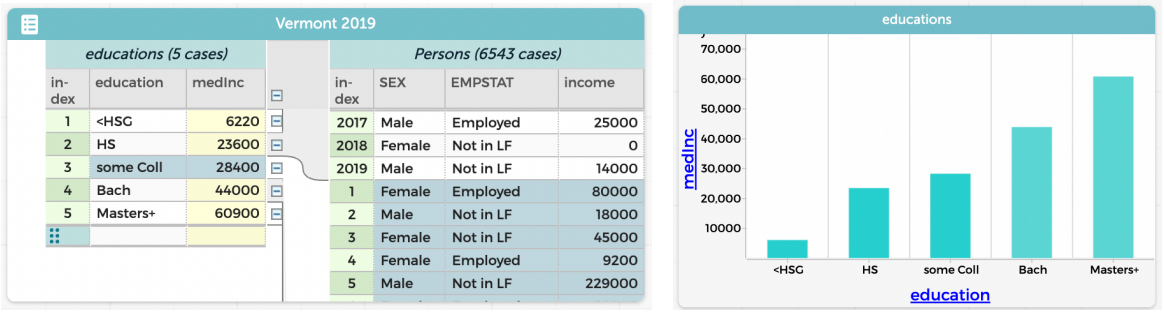


Figure 3. The same data displayed in Figure 2 but split hierarchically by education level. Cases with "some college" are selected. The graph shows just the medians instead of all 6,543 cases.

To generalize: we can group data *explicitly* by altering the hierarchy in the dataset. We create groups as top-level cases identified by the values of a grouping variable; the lower-level data cases are contained within the groups.

An important point here is that we seldom ask students to think about how their data are organized, and seldom do we ask them to do the organizing. A hierarchical scheme is particularly useful if students are comparing groups, and helps students wrestle with the meaning and genesis of the aggregate values they might use to make those comparisons. Although CODAP's hierarchical tables can

be visually confusing at first, that hierarchy is an interesting alternative to flat files, and has some potential pedagogical benefits.

- In this scheme, grouping is not a silent, implicit action. Students explicitly create groups in order to accomplish something such as comparing incomes across education levels.
- Because grouping requires purposeful dragging, students are more aware that grouping has taken place, and more aware of the connection between the raw microdata on the right and the processed aggregate values on the left.
- A grouped, hierarchical table makes it easy to select individual groups and filter a dataset to highlight or exclude that data, all by clicking rather than writing a formula. That is, the layout makes it easy to make some important data moves such as filtering.
- Students can use hierarchical tables to clean data efficiently, for example, to blank out values coded as missing, or to recode categorical attributes into fewer categories.
- A hierarchical organization lets students see individual values and their aggregate summaries side by side. By being able to see the variation in the original values, they will have a better sense of how meaningful aggregate differences are.
- Grouping is dynamic and mutable: if you un-group by dragging `education` back to the right, and then drag `sex` to the left, the table will show median incomes by sex. That is, you can re-use the formula in a new, parallel context.
- Being able to change the way a dataset is grouped—by dynamically changing its hierarchical structure—helps students see what grouping really is and why it is useful. This includes something as prosaic (and profound) as noticing that if you have a lot of duplicate values in a column, that column is ripe for grouping. That is, duplicate values signal a potential grouping variable.
- Changing the structure of a database helps students see that a dataset can appear very differently depending how you look at it. Purposefully working with multiple representations gives the student a better understanding of data and data organization.
- High-school students who used CODAP in a semester-long introductory data science course appeared, anecdotally, to understand grouping-by-hierarchy well and to use it effectively to accomplish their data-analysis goals. That is, they spontaneously grouped previously-unseen data, then used that grouping to recode categorical data, calculate and compare summary values, and make visualizations of those summaries. For example, using a dataset with thousands of individual murder cases, they drew conclusions about weapons used (recoding weapon into a binary: firearm or not) and how that was associated with the gender of the perpetrator.

We now see how hierarchy is closely tied to grouping, one of the core data moves (Erickson et al., 2019). We also notice how hierarchy facilitates and benefits from other data moves such as filtering and summarizing. Making it easier for students to do these data moves—altering a dataset's contents, structure, or values—is vital to introducing students to data science. Data moves help students access, organize, and explore richer and more complex data sets, and using hierarchy might be an important tool in that new toolbox.

Note how a professional package such as R, using the tidyverse library (Wickham & Grolemund, 2017), treats grouping. In that system, the user explicitly creates groups using the `group_by()` method, where the argument is the grouping variable. The result is typically a data frame containing the summary, that is, a separate, tidy, flat file of the aggregate, higher-level data.

This is efficient and elegant, but I suspect that many beginners are confused by the meaning of the grouping; how, precisely, the new table got the numbers that appear within it; and what the relationship is between the numbers in the two tables. I conjecture that a student who has experienced and implemented grouping using hierarchical tables in a flexible, dynamic environment such as CODAP will have an easier time when confronted with the additional abstraction and syntax of a programming environment.

REFERENCES

Erickson, T., Wilkerson, M., Finzer, W., & Reichsman, F. (2019). Data moves. *Technology Innovations in Statistics Education, 12*(1). https://doi.org/10.5070/T5121038001

Haldar, L. C., Wong, N., Heller, J. I., & Konold, C. (2018). Students making sense of multi-level data. *Technology Innovations in Statistics Education, 11*(1). https://doi.org/10.5070/T5111031358

Konold, C., Finzer, W., & Kreetong, K. (2014). *Students' methods of recording and organizing data* [Paper presentation]. American Educational Research Association Annual Meeting, New Orleans, LA.

Konold, C., Finzer, W., & Kreetong, K. (2017). Modeling as a core component of modeling data. *Statistics Education Research Journal, 16*(2), 191–212. https://doi.org/10.52041/serj.v16i2.190

The Concord Consortium. (2014). *Common Online Data Analysis Platform* (Version 2.0) [Computer software]. https://codap.concord.org/app/

Wickham, H., & Grolemund, G. (2017, April 3-7). *R for data science*. O'Reilly Media. https://r4ds.had.co.nz/