Schema-First Pipelines for Clinical Signals: Tidy Structuring as the Engine of Health Data Analytics

Lovish Gour

lovishsharma313@gmail.com

Union Commonwealth University, Barbourville, Kentucky, USA

Abstract. This paper shows how we can re-frame the analytical step of pre-processing as a first-class analytic step in health datasets. Furthermore, it sets out a tidy schema, where each variable is a column, each observation is a row, each observational unit is a table, etc. A suite of compact tools is prescribed, which includes melting, splitting, casting, normalization, etc. These tools can transform heterogeneous clinical, epidemiologic, and sensor records into interoperable, model-ready tables. Through a worked mortality case that engineers hourly cause-of-death profiles, joins code books, and constructs proportions, it demonstrates how schema-first tidying boosts downstream visualization, modeling, and reproducible workflows. In addition, it provides concrete implementation guidance through composable tools. Finally, it illustrates how standardizing structures improves validity, auditability, and scalability across health data analytics pipelines.

Keywords: Data cleaning, data tidying, clinical data, health analytics, reproducible research, data pipelines.

1 Introduction

In clinical trials, monitoring for births, deaths, and disease is an important public health surveillance tool. Nonetheless, analysis requires considerable effort to clear and prepare data. Dasu and Johnson discovered that 80 percent of data analysis is just cleaning the data. Data greatly originates from electronic health records (EHRs), medical devices, registries, and patient-reported outcome health domains. Datasets generated from these sources commonly have a misaligned structure, contain missing values, and have varied formatting that is difficult to analyze at a later time.

Despite the extensive time invested, there is still a lack of systematic methods for cleaning health data. Most of the literature conducted includes the algorithmic aspects of the analysis and considers pre-processing as an ad-hoc manual process. This research helps to fill in the gap with health data pipelines with an explicit schema-first approach that designs the dataset organization around tidy data principles.

An example definition for Tidy data is “Tidy data is a standard way to organize data values within a data set. In tidy data, each variable is a column, each observation is a row, and each type of observational unit is a table”. Data in this format will always be useful for analysis.

The advantages of tidy data are multifold. It facilitates the use of analysis tools because many statistical and visualization tools assume tidy data. Data transformations can be made visible

IICTDS 2025, December 19-20, Chandigarh, India
Copyright © 2026 EAI
DOI 10.4108/eai.19-12-2025.2362622

and reproducible. It reduces the chances of making mistakes because people misunderstand the structure of the data set. Audit-ability and analytical validity are essential benefits in health settings.

The work is based on the foundations of relational databases, data reshaping tools, and efforts to interactively clean data. However, it builds on these concepts to tackle some of the challenges that are specific to health data and puts forward a strategy for implementing tidy pipelines. A case study illustrates how organized tidiness applies to the investigation into the temporal patterning of death.

The remainder of this paper is organized as follows. Health Data and the Principles of Tidy Data The essential tidying operations for health datasets are contained in Section . Section describes a case study of mortality. Section discusses tidy tools for health analytics. Section reflects on limitations and future work. Section concludes.

2 Tidy Data Principles for Health Analytics

Organizing data in a standard way so that the structure of the data and the semantics are in line is called tidy data. A dataset is tidy when: (1) each variable is a column, (2) each observation is a row, and (3) each type of observational unit is a table. The principles we apply,inspired by Codd's third normal form ensures that data is semantically meaningful and computationally tractable [1].

In health analytics identifiers, clinical measurements, timestamps, or diagnostic codes are potential variables. Measurements can be individual lab results, vital signs, or medication assignments [2]. When these are kept tidy, analytic workflows become easier. If each row is just one measurement event, filtering, grouping, and visualizing blood pressure readings for trend analysis will be straightforward.

Looking at the blood pressure data in an untidy format in Table 1. Here, multiple variables systolic blood pressure and diastolic blood pressure are spread across columns in a way that confuses their measurement type and time [3]. This set-up makes it difficult to analyze. For example, calculating the average systolic pressure requires extracting numbers from several columns.

Table 1. Untidy blood pressure measurements.

PatientID	Visit1_SBP	Visit1_DBP	Visit2_SBP
P001	120	80	118
P002	130	85	128

Table 2 shows data in tidy format. To this point, there is one recorded blood pressure at each visit, with systolic and diastolic values in separate rows. It is easier to analyze because filtering by measurement type or calculating an average is a normal operation.

Table 2. Tidy blood pressure measurements.

PatientID	Visit	Type	Value
P001	**1**	**SBP**	**120**
P001	**1**	**DBP**	**80**
P001	**2**	**SBP**	**118**
P002	**1**	**SBP**	**130**
P002	**1**	**DBP**	**85**
P002	**2**	**SBP**	**128**

Health data often involves multiple levels of observation. A longitudinal study has patient information (one row per patient), repeated clinical measurements (multiple rows per patient), and environmental data (one row per day). According to tidy data principles, the various varieties of data ought to occupy separate tables [4]. Removing unnecessary repetition helps to ensure that no conflicts arise in the text [5].

It helps in better documentation and metadata. When variables have a consistent order, it makes it easier to navigate code-book and data dictionary. Furthermore, these factors are mandatory for conducting health research in collaboration [6]. Finally, the organization of variables makes data interpretation easy. Moreover, a machine can easily verify that clean datasets are indeed valid.

3 Methods for Tidying Health Data

Health datasets in the real world are seldom tidy. Lots of users have problems with the content of their data, including column headers that are values, not variable names. A column can store multiple variables. Rows and columns can store variables at the same time. One observational unit is split across multiple tables, and multiple observational units are in a table. This part explains step-by-step methods to get the answer.

3.1 Melting: Transforming Columns into Rows

When the headers of the columns are values of a variable, it can melt (or stack) the columns into a row. This transformation is commonly used when column names encode time points (e.g., Week1, Week2, Week3), resulting in a long-format dataset where time becomes an explicit variable. This is common in longitudinal health data.

The melt function takes in a data frame to investigate the changed identifier variables and measured variables, and collapses the data. The outcome is a dataset in a long-format table with additional variable and value columns. Table 3 and Table 4 shows melting on a simple dataset with its values.

Table 3. Example of melting.

PatientID	Week1	Week2	Week3
P01	5.2	5.5	5.8
P02	6.1	6.0	6.2

Table 4. Example of melting with values.

PatientID	Week	Value
P01	Week1	5.2
P01	Week2	5.5
P01	Week3	5.8
P02	Week1	6.1
P02	Week2	6.0
P02	Week3	6.2

3.2 Splitting Columns

When we melt the variable column, it usually contains multiple information points. The column Week1_SBP might carry information on both time and measurement type. Separating this column into distinct variables (for instance, Week and Type) cleans up the data even more. We can either use string splitting or regular expressions to do the same.

3.3 Casting: Reshaping Data by Variables

Casting refers to the action of converting several rows into columns. When we have categorical variables stored in rows but need to pivot them to columns, this is useful. For instance, if a dataset has a row for systolic and diastolic blood pressure, casting can reshape it into one column each.

3.4 Normalization: Separating Observational Units

If a table has different observational units, it must be broken down into more than one table. A clinical dataset might be made up of both patient demographic and laboratory data. This should be stored in different tables connected with a patient identifier. This avoids duplication and ensures consistency. Figure 1 shows the melting transform wide data into long format.

3.5 Combining Multiple Tables

On the contrary, when data for a single observational unit is separated into more than one table (e.g., yearly files), they should be folded into one. This often involves reading each file, adding a source identifier, and binding them together.

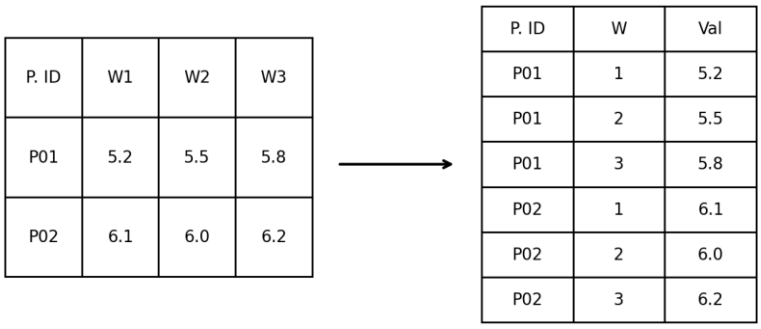


Fig. 1. Melting transforms wide data into long format.

4 Case Study: Mortality Data Analysis

For demonstration of tidy data application and practicality, we utilize mortality data in Mexico. The dataset holds a wide assortment of information about 539,000 deaths that occurred in 2008. We are looking for the reasons behind most of the deaths that occur indirectly.

4.1 Data Loading and Initial Inspection

The raw numbers are listed with a new line for each death instance, containing columns for date and the cause of death. The table 5 presents a sample of death data. Inspections of death records show that the hour of death is sometimes not written and instead the cause of death is written out as a medical code.

Table 5. Sample of mortality dataset.

Year	Month	Day	Hour	COD
2008	1	1	1	B20
2008	1	2	4	I67
2008	1	3	8	I50
2008	1	4	12	I50
2008	1	5	16	K70

4.2 Data Tidying Steps

We applied the following steps to tidy the data:

1. **Filter**: Removed records with missing hour of death. 2. **Aggregate**: Counted deaths by hour and cause. 3. **Join**: Merged with a codebook to translate COD codes into disease names. 4. **Transform**: Computed proportional death rates by cause and hour. 5. **Normalize**: Separated cause-specific profiles for further analysis.

These steps produced a tidy dataset where each row represents the death count for a given cause and hour, with additional columns for disease name and the proportional rate.

4.3 Analysis and Visualization

The total number of deaths per hour across the world was computed. Next was comparing the overall death rate with its percentage. Mean square error was next used to calculate every single cause's difference from the overall pattern. Points with big differences and very high numbers were marked as unusual.

Figure 2 shows the relationship between sample size and deviation. A robust regression line highlights causes with residuals above 1.5, indicating unusual temporal patterns.

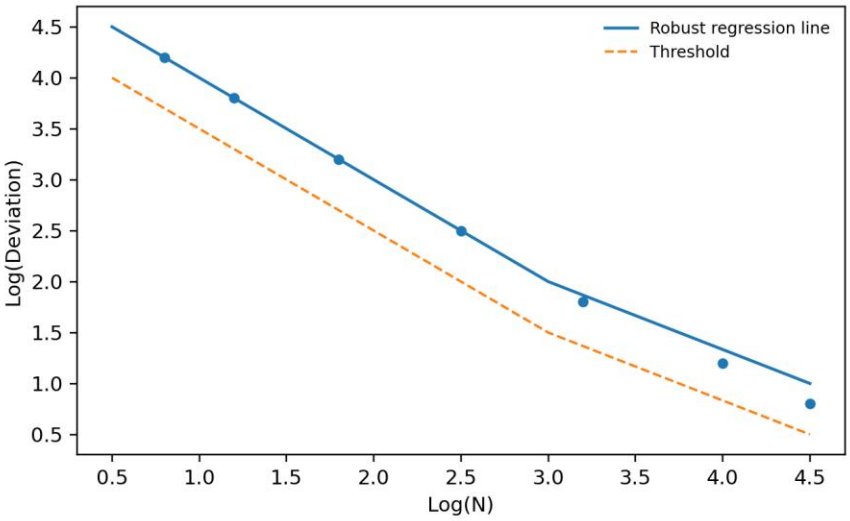


Fig. 2. Log–log plot of deviation vs. sample size.

4.4 Results

The analysis revealed distinct temporal patterns in cause-specific mortality. For example, deaths due to external causes such as violent crimes, drug overdoses, and alcohol-related conditions showed pronounced hourly variations, potentially reflecting social and environmental influences. Meanwhile, countries with higher mortality rates also exhibited correlations with factors such as weaker immune defenses and higher air pollution levels[7]. The tidy data approach facilitated efficient and transparent analysis of these patterns.

5 Tools for Health Analytics

Tidy data combined with output and input tools is a very powerful thing to have. This segment talks about tools and statistics related to health.

5.1 Data Manipulation

Data manipulation features include filtering, modification, combining, and arranging. Tidy data will simplify operations because each element is one variable. There are various packages, such as dplyr, which provide a grammar of data manipulation that enables expressive and composable data transformation workflows [8]. These functions have been made for neat data and can be mixed for bigger projects.

There could be different methods to follow the procedure in order to save time and perfection.

```
bp_summary              <-              bp_tidy              %>%
  filter(Type             ==              "SBP")           %>%
  group_by(PatientID,                   Group)             %>%
  summarise(MeanSBP = mean(Value))
```

This code is readable, reusable, and efficient. Similar functionality is available in Python via pandas [2].

5.2 Data Visualization

Data organization makes it possible to use presentation tools like **ggplot2**. Variables are mapped directly to visual aesthetics such as position, color, and shape within plotting frameworks.

For example, to plot hourly death rates by cause, one might write:

```
ggplot(hod_tidy, aes(x = Hour, y = Proportion, color = Cause)) +
  geom_line()
```

The process makes it possible to have multiple additives of different colors in paragraph. Tidy data eliminates the need to transform data before it.

5.3 Statistical Modeling

Most statistical models assume tidy data. In a data frame for linear regression in R, every row reveals an observation, and every column presents a variable. Using tidy data aside from Spreadsheets is helpful in limiting complexity with the model formula.

Local effects as well as the overall trend of the entire population benefit from this data format. Data organized clear fix to store in a way that helps identify relationships.

5.4 Reproducible Reporting

Tidy data pipelines work well when paired with reporting tools. Since the method is scripted, it can be seen as fair and can be audited all the way to the final conclusion. Maintaining such records is important so the analyses is traceable.

6 Discussion

The schema first approach reactions provide users big benefits in analytics. Data structures reduce ad hoc operations, therefore, enhancing collaboration for users. However, several challenges and limitations remain.

First, adopting tidy data requires upfront investment. Big researchers have to learn some new principles and tools, so some of their old data will have to be reclassified [9]. Large amounts of information, which is updated frequently, justify the financial investment.

Second, not all data naturally fit the tidy model. Special formats might be developed for data with multiple dimensions to organize it efficiently. Additionally, in some situations, neat organization of tables can assist when looking for specific information, because of functions like tidyr and pandas.

The cleanliness of data is not enough to remove errors from the data. The most underestimated limit occurs due to inaccurate recording. Principles of tidy data should be paired with broader cleaning frameworks.

Data analysis is far from the process of just crunching the numbers. How do researchers interact with tidy data [10]? What tools and interfaces best support tidy workflows [11]? These remains an open research area. Research studies and active involvement by users could explore those questions.

Adopting tidy data principles provides a strong foundation for scalable and reproducible health analytics. As health informatics evolves, both data resources and regulatory frameworks will benefit from structured, well-organized approaches to data management.

7 Conclusion

This paper explains a schema-first approach for health data that emphasizes structured and tidy data organization, using a clean approach. Using observations as rows and variables as columns will support and simplify the process of research. Methods used to find patterns within data helped with the process's speed.

The methods described here accelerated the discovery of temporal and causal patterns in mortality data, demonstrating the practical value of tidy structuring.

References

[1] H. Wickham, "The split-apply-combine strategy for data analysis," Journal of Statistical Software, vol. 40, no. 1, pp. 1–29, 2011.

[2] W. McKinney, "Data structures for statistical computing in python," in Proceedings of the 9th python in science conference, 2010, pp. 51–56.

[3] K. Müller and H. Wickham, "A framework for making predictive models useful in practice," Journal of Statistical Software, vol. 92, no. 1, pp. 1–29, 2020.

[4] T. Dasu and T. Johnson, Exploratory data mining and data cleaning. John Wiley & Sons, 2003.

[5] D. Sarkar, "Lattice: Multivariate data visualization with r," Journal of Statistical Software, vol. 25, no. 1, pp. 1–3, 2008.

[6] L. Wilkinson, The grammar of graphics. Springer Science & Business Media, 2005.

[7] H. Wickham, "Reshaping data with the reshape package," Journal of Statistical Software, vol. 21, no. 12, pp. 1–20, 2007.

[8] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer, "Wrangler: Interactive visual specification of data transformation scripts," in Proceedings of the SIGCHI conference on human factors in computing systems, 2011, pp. 3363–3372.

[9] H. Wickham, "ggplot2: Elegant graphics for data analysis," Journal of Statistical Software, vol. 35, no. 1, pp. 1–3, 2009.

[10] H. Wickham, "Tidy data," Journal of Statistical Software, vol. 59, no. 10, pp. 1–23, 2014.

[11] E. F. Codd, The relational model for database management: Version 2. Addison-Wesley Longman Publishing Co., Inc., 1990.