Stat 159 Presentation: Organizing Data in Spreadsheets

Authors: Morgan Smart, Dakota Lim, Gary Nguyen and Lauren Hanlon

Storing data in spreadsheets has been a common practice among data collectors. Microsoft Excel is the most popular spreadsheet software that a lot of people now store their data in. But what is troublesome is with that practice of storing data using spreadsheets is that it makes preparation for data analysis a lengthy and tedious process. As data analysts obtain raw data from spreadsheet sources, they have to spend a significant amount of time rearranging the layout of data they obtained. Karl Browman, a Professor in the Department of Biostatistics & Medical Informatics at the University of Wisconsin –Madison, argues that a more efficient practice is to have data collectors follow general guidelines on storing data in spreadsheets. This way, data analysts can easily handle any data handed to them, and thus would have more time analyzing their data.

As a result, Karl Browman created the 12 guidelines on the topic of organizing data in spreadsheets, with tutorials available on his Github account. The guidelines, in the order placed on his personal website are: consistency, format of dates, cell filling, non-emptiness for cell, rectangular form for data, data Dictionary, data-only excel files, no coloring or highlighting as data, good and consistent naming for things, backing up data regularly in multiple locations, using data validation to avoid errors , and saving data in plain text files. We will elaborate on each guideline throughout this report.

Consistent coding practices are a key component of any properly organized data set. Everything from variable names, table names, notes, comments, row or column identifiers, to file names and dating conventions needs to be kept in a logical, informative, and consistent format. For example, an overlooked aspect of naming conventions is what to name missing data, often times programming languages will fill in missing data with “NA” or “NaN” values which can cause errors when iterating through said data. A more informative approach is to replace all missing values with a single large integer which does not appear elsewhere in the data, now when performing actions on said data we can easily identify the missing values without causing errors. After choosing a naming convention or style, it’s important to be consistent throughout the project and follow the conventions across all files. Discrepancies between files and data can cause confusion and reduce interpretability.

Consistency with dates is another important factor in data organization. Although there are several options for formatting dates, when it comes to writing code, it is best to store dates as text in the following format: YYYY-MM-DD. Using this convention, we can be sure that all data will be consistent and easily legible. Once again, the key here is to be consistent. If someone wanted to, they could code all dates as integers (as operating systems do) counting from some arbitrary date in the past. All naming conventions are equally valid so long as we use them consistently, the YYYY-MM-DD method is recommended by the article simply because of its legibility and wide spread acceptance as a conventional method of recording dates.

When formatting data tables, it is a common practice to eliminate any blank spaces for the ease of computer processing. For example, if a data table has a date column, some data authors may choose to write the date once and only re-enter it if the date changed from the previous entry. This will cause issues if we refactor the data or sample it because all the blank data will get lost. Similarly, some data tables record the same data over time intervals for the same subjects and store the data horizontally using column names as time interval markers. In this situation, it is best to replicate the rows of the data and include a time marker column because a vertical table is much easier to interpret and can be iterated over with computer programs much more effectively.

In order to maintain the ease of interpretability of a data table, it is a common practice to only include one piece of information per column. For example, some data may be initially written as pairwise data, like a "model-year" column in a data table recording car sales. If we wanted to conduct extensive analysis on this data it would be best to create separate "model" and "year" columns since they are two unique pieces of data. Separating pairwise data will also prevent the data type from being messed up. This is why we also do not include comments in the data table. Including comments in a data table leads to string data types in columns that should not be strings. It's also best to never merge cells. Merging cells can be tricky and lead to large amounts of white space. Although it may make the data look more appealing on an excel table, it ultimately leads to issues when attempting to analyze the data.

When creating tables of your data, it is imperative to practice proper structured formatting. Broman stresses the importance of keeping our data in the shape of a rectangle, with rows corresponding to data entries, and columns corresponding to our variables. If we find that our data isn't fitting nicely into a single rectangle, it may be necessary to create multiple tables, to later be joined using a distinct ID, or other indicator. It's best to avoid blank rows between tables (within a table), because while it might make sense to do this within an Excel document, when we try to run analysis on the data, we will find ourselves cleaning the data and run the risk of confusing "tables" within a larger table. Formatting our tables in nice rectangles is crucial to reproducibility, otherwise the analyst will need to spend time studying the layout instead of simply reading row names and being able to understand the data at hand. Another important practice when creating tables is to format our tables in a way where we minimize the number of repeated values. If we find ourselves working with multiple repeated values, we may want to consider separating the tables and then later combining them.

Creating a data dictionary is crucial to reproducibility. A data dictionary is part of the metadata. Examples of entries in our metadata include variable names, descriptions of these variables, information about the variables (such as the measurement unit), and potentially some summary statistics about the variables (such as min/max/number of entries). It is important to note that the data dictionary should retain the same format as described previously, in a rectangle. A ReadMe file would also be included in any reproducible project, and the data dictionary should be included in this ReadMe.

While this might be contested by others who work with data, Broman writes that raw data files should not contain calculations. It might be easy to include a simple calculation (such as profit, derived from revenue and costs) in the raw files, but if you feel that this calculation must be saved, it's important to create a duplicate of the raw file and complete the calculation. This is important because if the work is to be reproduced with new data, we will more than likely get the raw data in the same format as before, and if our analysis is dealing with this "profit" column, then we will surely run into errors when importing a new raw data file. Broman also notes that changing the raw data files might lead to accidental typos or deleting data.

We feel as though this might be obvious, but it is worth noting that we should not use font color or highlighting as data. This might arise when using Excel as our primary mode of data analysis as it might be easier to visualize, but if we're using colors or highlighting data then this will not translate if we import our data into an analysis tool (such as R or Python), and these comments will be lost. If we feel the need to distinguish our data, or put emphasis on outliers, we simply add an additional column such as "outlier" and indicate whether or not it's true with a TRUE/FALSE value. This is much more conducive to reproducibility.

Choosing good names is another important practice. A good name is one that doesn't include spaces or special characters and is short but meaningful. Spaces should be avoided because if spaces are used in names, when we want to call that name, we'll have to put quotation marks around it. We're also careful to not include extraneous spaces when creating names because this will require us to put quotations around the name when calling it. Special characters shouldn't be used in names because in certain cases these special characters may have other meanings. For example, if we create an R variable called "laundry$", R would think we are trying to call a blank column of a data frame named "laundry." It's also important to keep names short so they're easy to call but not too short that their meanings are lost. A name called "weights" is a better name than "w." because "w." could stand for a host of things. We also shouldn't include "final" in any of our names. Inevitably, we will end up having more than one final version leading to "final1", "final1_rev1", and so on.

Making backups could make or break our work. If we don't backup our work, a burning building destroy could destroy an entire project. We prevent this from happening by backing up our work on systems such as git or dat. Additionally, we keep all versions of our data files in case we make an error and want to return to a prior version. When we're compiling a spreadsheet of data and have reached our final dataset, we write-protect the file so no changes can be made to the data on accident.

We can further prevent our dataset from being accidently corrupted by using data validation techniques. It is common to make mistakes when manually entering in data. To check what we've entered in Excel, we use Excel's data validation feature. We simply select the column we want to verify, in the menu bar click "Data -> Validation," and choose the appropriate validation criteria (such as a limit on text length). We can also select the datatype of a column to prevent the data from being construed. For example, we may have a column in Excel

that is a list of IDs and we want these IDs to be read in by R as characters (rather than as numeric). We make these IDs readable as characters by changing the column's datatype to "Text."

To increase the reproducibility of our work, we save our data as a comma or tab delimited plain text file. These kinds of files never require and kind of software in order to be processed and thus can be accessed by anyone with a computer. If our datafile contains any special formatting that would be compromised by saving it as a plain text file, we try and find a way to make our data file less complex so it can be saved as a plain text file without loss of information. If this isn't possible, then we don't save it as a plain text file--in this case, saving it as a plain text file would be preventing our work from being reproduced.

To conclude, there are a few last things we're careful of. One is watching for automatic data changes because the data type isn't specified. For example, we may have a number that is too big so it shows up in scientific form--and when we enter the data into R--it's now read as a character rather than an integer because it contains letters. Another note is if a cell's value is zero, we don't leave it blank. Leaving a cell blank will result in it's value becoming "NULL" when the data is taken in. Zeros are data too so we make sure to fill them in. A final handy trick is we utilize "Freeze Panes" in Excel to freeze the column and row names so they're always visible as one scrolls down or across the dataset.

Following the Karl Browman's recommendations listed in this paper is a sure way to create a dataset that is easy to interpret and easy to share with others without loss of accuracy. Although there are many methods for organizing data in spreadsheets, following Browman's recommendations, one can decrease errors in datasets and increase the reproducibility of all analysis done using the dataset. As more and more data is being collected, good practice in storing data using common spreadsheets software is strongly recommended as analysis of data requires pre-processing of raw data.