datasharing memo

Bo

9/24/2019

How to share data with a statistician

This is a guide for anyone who needs to share data with a statistician or data scientist. The target audiences I have in mind are:

- Collaborators who need statisticians or data scientists to analyze data for them
- Students or postdocs in various disciplines looking for consulting advice
- Junior statistics students whose job it is to collate/clean/wrangle data sets

The goals of this guide are to provide some instruction on the best way to share data to avoid the most common pitfalls and sources of delay in the transition from data collection to data analysis. The Leek group works with a large number of collaborators and the number one source of variation in the speed to results is the status of the data when they arrive at the Leek group. Based on my conversations with other statisticians this is true nearly universally.

My strong feeling is that statisticians should be able to handle the data in whatever state they arrive. It is important to see the raw data, understand the steps in the processing pipeline, and be able to incorporate hidden sources of variability in one's data analysis. On the other hand, for many data types, the processing steps are well documented and standardized. So the work of converting the data from raw form to directly analyzable form can be performed before calling on a statistician. This can dramatically speed the turnaround time, since the statistician doesn't have to work through all the pre-processing steps first.

What you should deliver to the statistician

To facilitate the most efficient and timely analysis this is the information you should pass to a statistician:

1. The raw data.
2. A tidy data set
3. A code book describing each variable and its values in the tidy data set.

4. An explicit and exact recipe you used to go from 1 -> 2,3

Let's look at each part of the data package you will transfer.

The raw data

It is critical that you include the rawest form of the data that you have access to. This ensures that data provenance can be maintained throughout the workflow. Here are some examples of the raw form of data:

- The strange binary file your measurement machine spits out
- The unformatted Excel file with 10 worksheets the company you contracted with sent you
- The complicated JSON data you got from scraping the Twitter API
- The hand-entered numbers you collected looking through a microscope

You know the raw data are in the right format if you:

1. Ran no software on the data
2. Did not modify any of the data values
3. You did not remove any data from the data set
4. You did not summarize the data in any way

If you made any modifications of the raw data it is not the raw form of the data. Reporting modified data as raw data is a very common way to slow down the analysis process, since the analyst will often have to do a forensic study of your data to figure out why the raw data looks weird. (Also imagine what would happen if new data arrived?)

The tidy data set

The general principles of tidy data are laid out by Hadley Wickham in this paper and this video. While both the paper and the video describe tidy data using R, the principles are more generally applicable:

1. Each variable you measure should be in one column
2. Each different observation of that variable should be in a different row
3. There should be one table for each "kind" of variable
4. If you have multiple tables, they should include a column in the table that allows them to be joined or merged

While these are the hard and fast rules, there are a number of other things that will make your data set much easier to handle. First is to include a row at the top of each data table/spreadsheet that contains full row names. So if you measured age at diagnosis for patients, you would head that column with the name `AgeAtDiagnosis` instead of something like `ADx` or another abbreviation that may be hard for another person to understand.

Here is an example of how this would work from genomics. Suppose that for 20 people you have collected gene expression measurements with RNA-sequencing. You have also collected demographic and clinical information about the patients including their age, treatment, and diagnosis. You would have one table/spreadsheet that contains the clinical/demographic information. It would have four columns (patient id, age, treatment, diagnosis) and 21 rows (a row with variable names, then one row for every patient). You would also have one spreadsheet for the summarized genomic data. Usually this type of data is summarized at the level of the number of counts per exon. Suppose you have 100,000 exons, then you would have a table/spreadsheet that had 21 rows (a row for gene names, and one row for each patient) and 100,001 columns (one row for patient ids and one row for each data type).

If you are sharing your data with the collaborator in Excel, the tidy data should be in one Excel file per table. They should not have multiple worksheets, no macros should be applied to the data, and no columns/cells should be highlighted. Alternatively share the data in a CSV or TAB-delimited text file. (Beware however that reading CSV files into Excel can sometimes lead to non-reproducible handling of date and time variables.)

The code book

For almost any data set, the measurements you calculate will need to be described in more detail than you can or should sneak into the spreadsheet. The code book contains this information. At minimum it should contain:

1. Information about the variables (including units!) in the data set not contained in the tidy data

2. Information about the summary choices you made
3. Information about the experimental study design you used

In our genomics example, the analyst would want to know what the unit of measurement for each clinical/demographic variable is (age in years, treatment by name/dose, level of diagnosis and how heterogeneous). They would also want to know how you picked the exons you used for summarizing the genomic data (UCSC/Ensembl, etc.). They would also want to know any other information about how you did the data collection/study design. For example, are these the first 20 patients that walked into the clinic? Are they 20 highly selected patients by some characteristic like age? Are they randomized to treatments?

A common format for this document is a Word file. There should be a section called "Study design" that has a thorough description of how you collected the data. There is a section called "Code book" that describes each variable and its units.

How to code variables

When you put variables into a spreadsheet there are several main categories you will run into depending on their data type:

1. Continuous
2. Ordinal
3. Categorical
4. Missing
5. Censored

Continuous variables are anything measured on a quantitative scale that could be any fractional number. An example would be something like weight measured in kg. Ordinal data are data that have a fixed, small (< 100) number of levels but are ordered. This could be for example survey responses where the choices are: poor, fair, good. Categorical data are data where there are multiple categories, but they aren't ordered. One example would be sex: male or female. This coding is attractive because it is self-documenting. Missing data are data that are unobserved and you don't know the mechanism. You should code missing values as `NA`. Censored data are data where you know the missingness mechanism on some level. Common examples are a measurement being below a detection limit or a patient being lost to follow-up. They should also be coded as `NA` when you don't have the data. But you should also add a new column to your tidy data called, "VariableNameCensored" which should have values of `TRUE` if censored and `FALSE` if not. In the code book you should explain why those values are missing. It is absolutely critical to report to the analyst if there is a reason you know about that some of the data are missing. You should also not impute/make up/ throw away missing observations.

In general, try to avoid coding categorical or ordinal variables as numbers. When you enter the value for sex in the tidy data, it should be "male" or "female". The ordinal values in the data set should be "poor", "fair", and "good" not 1, 2 ,3. This will avoid potential mixups about which direction effects go and will help identify coding errors.

Always encode every piece of information about your observations using text. For example, if you are storing data in Excel and use a form of colored text or cell background formatting to indicate information about an observation ("red variable entries were observed in experiment 1.") then this information will not be exported (and will be lost!) when the data is exported as raw text. Every piece of data should be encoded as actual text that can be exported.

The instruction list/script

You may have heard this before, but reproducibility is a big deal in computational science. That means, when you submit your paper, the reviewers and the rest of the world should be able to exactly replicate the

analyses from raw data all the way to final results. If you are trying to be efficient, you will likely perform some summarization/data analysis steps before the data can be considered tidy.

The ideal thing for you to do when performing summarization is to create a computer script (in `R`, `Python`, or something else) that takes the raw data as input and produces the tidy data you are sharing as output. You can try running your script a couple of times and see if the code produces the same output.

In many cases, the person who collected the data has incentive to make it tidy for a statistician to speed the process of collaboration. They may not know how to code in a scripting language. In that case, what you should provide the statistician is something called pseudocode. It should look something like:

1. Step 1 - take the raw file, run version 3.1.2 of summarize software with parameters a=1, b=2, c=3
2. Step 2 - run the software separately for each sample
3. Step 3 - take column three of outputfile.out for each sample and that is the corresponding row in the output data set

You should also include information about which system (Mac/Windows/Linux) you used the software on and whether you tried it more than once to confirm it gave the same results. Ideally, you will run this by a fellow student/labmate to confirm that they can obtain the same output file you did.

What you should expect from the analyst

When you turn over a properly tidied data set it dramatically decreases the workload on the statistician. So hopefully they will get back to you much sooner. But most careful statisticians will check your recipe, ask questions about steps you performed, and try to confirm that they can obtain the same tidy data that you did with, at minimum, spot checks.

You should then expect from the statistician:

1. An analysis script that performs each of the analyses (not just instructions)
2. The exact computer code they used to run the analysis
3. All output files/figures they generated.

This is the information you will use in the supplement to establish reproducibility and precision of your results. Each of the steps in the analysis should be clearly explained and you should ask questions when you don't understand what the analyst did. It is the responsibility of both the statistician and the scientist to understand the statistical analysis. You may not be able to perform the exact analyses without the statistician's code, but you should be able to explain why the statistician performed each step to a labmate/your principal investigator.

Contributors

- Jeff Leek - Wrote the initial version.
- L. Collado-Torres - Fixed typos, added links.
- Nick Reich - Added tips on storing data as text.
- Nick Horton - Minor wording suggestions.