How to share data for collaboration

Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health and C

Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health and C

June 2, 2017

Abstract

Within the statistics community, a number of guiding principles for sharing data have emerged; however, these principles are not always made clear to collaborators generating the data. To bridge this divide, we have established a set of guidelines for sharing data. In these, we highlight the need to provide raw data to the statistician, the importance of consistent formatting, and the necessity of including all essential experimental information and pre-processing steps carried out to the statistician. With these guidelines we hope to avoid errors and delays in data analysis.

Keywords: tidy data, guidelines, statistician, data sharing, analysis

*We want to thank Jenny Bryan and the referees, Nicholas Horton and Stephen Turner, for their helpful contributions and edits. We would additionally like to thank Foram Ashar, Pat Carlson, Claude Chaunier, Leonardo Collado-Torres, Dan Fowler, David Jankoski, Sean Kross, Gene Miller, Leslie Myint, and Nick Reich for their suggestions and edits.

1 Introduction

A set of general principles for sharing data have emerged within the statistics community (Browman & Woo n.d., Wickham (2014),Wilson et al. (2016),White et al. (2013)). But these principles are not always clear to researchers, scientists, or collaborators generating the data. This has led to a disconnect between those generating data and those analyzing it about the best way for data to be shared. To bridge this divide, we have developed general guidelines for anyone generating data who anticipates their data will be shared with a statistician, data scientist, or analyst at some point during their project. 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 (Leek & Peng 2015). This guide focuses on data sharing between collaborators. It does not directly address best practices for how to make the data behind a published paper available in public repositories; however, following many of the guidelines we present will be critical during that process as well.

When it comes to collaborations between data collectors and statisticians, it is a reasonable expectation that the statistician should be able to handle and analyze the data in whatever state they arrive. For this to be possible, the statistician must be provided the raw data, information on any steps taken to preprocess the data, and enough information about the experimental conditions to allow the statistician to identify and incorporate hidden sources of variability into his or her analysis (Baggerly 2010). On the data generator's end, it can be expected that he or she will receive results from a statistician in a reasonable amount of time. From our experience in the Leek group (*Leek Group* n.d.) (where we work with a large number of collaborators to analyze data) and from conversations with other statisticians, the number one source of delay in the speed of returning results to collaborators is the condition of the data when they arrive.

To help meet the expectations of both the data generator and the statistician, all of the necessary information must be provided and provided in a consistent and well-organized manner to the statistician. Consistent data sharing reduces the likelihood of errors during analysis and also decreases analysis turnaround time.

We provide these guidelines on data sharing and explain the reasoning behind them from

the analyst's perspective. We envision these will be useful to the following individuals:

- Collaborators who need statisticians or data scientists to analyze data
- Students or postdocs in various disciplines looking for consulting advice
- Junior statistics students whose job it is to collate, clean, and wrangle data sets
- Statisticians or data scientists seeking a concise guide to share with collaborators to clarify best practices for data sharing

2 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 (Wickham 2014).
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.

For clarity, we will further define each part of the data package transferred.

2.1 The raw data

It is critical that you include the rawest form of the data to which you have access. 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 (*JSON: The Fat-Free Alternative to XML* n.d.) data you got from scraping the Twitter API (*Twitter* n.d.)
- 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. Did not remove any data from the data set
4. Did not summarize the data in any way

If you made any modifications to the raw data, it is not the raw form of the data. For example, statisticians are often supplied summary statistics (such as averages) rather than the underlying raw data used to calculate these summary statistics. While the intent of the data collector is to be helpful, the reality is that this slows the analyst down. Statisticians can easily calculate any appropriate summary statistics from the raw data. Being provided the raw data is essential for accurate analysis.

It can often help to consider what would happen if new data for a study were to arrive to the statistician. If this new data requires no modification before being combined with the first set of raw data provided, it is likely raw 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.

2.2 The tidy data set

The general principles of tidy data have been laid out by Hadley Wickham previously (Wickham 2014). While the paper describes tidy data using R (*R: The R Project for Statistical Computing* n.d.), 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 data.
4. If you have multiple tables, they should include a column in the table (with the same column label!) that allows them to be joined or merged (see **Figure 1A-B**).

While these are the most critical decisions, there are a number of additional things that will make your data set much easier to handle (Browman & Woo n.d.) (summarized in **Box1**). Briefly, it is best to include a row at the top of each data table or spreadsheet that contains informative column names. And, each cell should include only one value or

A Hormone_Data_raw.csv

PatientID	Cortisol	IGF1	…	Hormone50
ID1	17.4	327	…	33.5
ID2	18.1	412	…	44.2
⋮	…	…	…	…
ID40	20.2	264	…	28.6

B Demographic_Data.csv

PatientID	Age	Sex	BMI	CollectionDate	Diagnosis
ID1	45	female	18	2016-09-25	control
ID2	12	female	17	2016-09-25	diabetes
⋮	…	…	…	…	…
ID40	40	male	22	2016-09-29	control

D Pseudocode.docx:

1. Values for hormone levels were received from company Y and input into Horome_Data_raw.csv. No processing has been done on these values.

2. Values in Demographic_Data.csv were obtained upon visit to clinicX. Data were extracted from electronic medical record and input into Excel by Jane Doe.

C CodeBook.docx:

Study Design:

Experimental Question: This study looks to determine whether or not there are differences in hormone levels in individuals with diabetes relative to healthy controls.

Sample Details: 20 individuals with diabetes and 20 unrelated age- and sex-matched controls were included for study. Individuals were recruited to the study using flyers posted throughout Johns Hopkins Hospital and online recruitment through www.website.com. Informed consent was obtained from all study participants. Blood was drawn by a single phlebotomist in clinic X and all samples processed on the same day they were collected by company Y.

Code Book/Data dictionary:

Variable	Description	Units	CodingNotes	OtherNotes
Age	Age At Blood Draw	years	numerical	Taken from electronic medical record
Sex	Self-reported	'male', 'female'	2-level factor	Confirmed using electronic medical record
BMI	weight/height	kg/m^2	numerical	Measured day of blood draw
Collection Date	Date of Blood Draw	date	YYYY-MM-DD	Collection of blood by phlebotomist
Diagnosis	Individual diagnosis	'diabetes', 'control'	2-level factor	`diabetes` = Type 2 Diabetes. Confirmed by medical record.
Cortisol	Stress Hormone	µg/dL	numerical	Required fasting and to be measured in the AM (8-10am)
IGF1	Insulin-Like Growth Factor 1	ng/dL	numerical	Did not require fasting, but taken at the same time as other measures
⋮	⋮	⋮	⋮	⋮
Hormone50	Hormone Name	ng/dL	numerical	Hormone Details

Figure 1: Data Organization **A**. Raw (and tidy) data. The raw measurements taken during the experiment are included here for the 40 patients (rows) and 20 hormones (columns) measured. Indicating that these are raw values conveys that no manipulation or computation has been done on the included values. **B**. Tidy data. These data convey the important clinical information to your statistician for each sample (rows) across a number of variables (columns). With a single value included in each cell and consistently and informatively named column headers and values, these tidy data can be easily understood and used by your statistician. Importantly, `PatientID` is coded in the exact same way between Hormone_Data_raw.csv and Demographic_Data.csv enabling easy merging by the statistician. **C**. Code Book. Here, all pertinent and detailed information about both the experiment and the data are conveyed to the statistician. This is the place for any extra detail that does not fit into the data generator's tidy data tables or spreadsheets. **D**. Pseudocode. This includes information about any processing steps taken to get the data into the form in which the statistician is receiving it.

unit of information. Sentences should generally be avoided here; any lengthy explanations should instead be included in the "Code book."

To provide an experimental example, suppose you want to know if individuals with diabetes have altered hormonal profiles. To answer this question, you carry out an experiment in which blood is drawn from 20 individuals with diabetes as well as 20 healthy controls. This blood was used to measure blood levels for 50 different hormones. These measurements would comprise your first 'kind' of data (see #3 above). You have also collected demographic information from the patients including their age, sex, BMI, and diagnosis (your second "kind" of data). For this example, your first table or spreadsheet would include the measurements for the 20 different hormones. This would have 41 rows (a row for the name of the measured hormones at top and then one row for each of the 40 individuals in your study) and 51 columns (one for `PatientID` and then one for each measurement) (**Figure 1A**). You would have another table or spreadsheet that contains the demographic information. It would have six columns (`PatientID`, `Age`, `Sex`, `BMI`, `CollectionDate`, `Diagnosis`) and 41 rows (a row with variable names, then one row for each patient) (**Figure 1B**).

With regards to the formatting of these data, 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 cells should be highlighted. Alternatively, the data could be shared in either a CSV or TAB-delimited text file. Caution should always be taken when reading CSV files into Excel as it can sometimes lead to non-reproducible handling of date variables, time variables, and variables that Excel incorrectly assumes are date or time variables (Zeeberg et al. 2004). For example, Excel incorrectly assumes the gene `SEPT9` is the date `Sept-9` due to default date format conversions. Floating-point format conversions cause similar problems. To avoid these issues, use ISO 8601 (Newman & Klyne n.d.) guidelines when coding date and time variables. (See **Box2**). Whenever data is being formatted, the person tidying the data must be extremely careful that no unintended alterations have been made. Spot checking data after tidying, which includes, but is not limited to, ensuring the correct number of columns and rows are present and that column labels are accurate and consistent across spreadsheets, is crucial.

When..	Be sure to...	So Do this...	Avoid this...	Why?
Naming variables (aka assigning column headers)	Use meaningful variable names	`AgeAtDiagnosis`	`ADx`	`ADx` is an unclear and uninformative abbreviation
Naming variables	Avoid spacing in column headers	`AgeAtDiagnosis`	`Age At Diagnosis`	Spacing in variable names makes the analyst's life more difficult
Naming variables	Use consistent capitalization	`AgeAtDiagnosis`	Using both `AgeAtDiagnosis` and `ageatdiagnosis`	Using consistent column names across tables/spreadsheets simplifies any merging the statistician may have to do.
Naming variables	Avoid using separators, but if it's necessary, use an underscore (`_`)	`IGF1` (or `IGF_1`)	`IGF.1`, `IGF-1`, `IGF/1`, `IGF,1`	Separators (commas, periods, hyphens, slashes, spaces etc.) often have different meanings in coding languages than they do in text. Avoiding them avoids error.
Coding variables	Avoid unnecessary spaces	'male'	'male '	That extra space after 'male ' makes it different from 'male' without a space.
Coding variables	Be consistent!	'male'	'Male',`male', and 'M',	In the eyes of the statistician, 'Male',`male', and 'M' could be incorrectly perceived as three different values.
Coding variables	Be careful of spelling errors	'male'	'maale'	That extra 'a' makes these two different categories.
Coding date and time	Use ISO 8601 coding	'YYYY-MM-DD'	'MM/DD/YY` and `Month Day, Year`	Consistency simplifies the analyst's life, and YYYY-MM-DD will not be misconstrued if opened in Excel.
Coding missing data	Not leave any cells blank and use a consistent value	'NA'	'0', '-9', red-highlighted blank cells, '.', ' ', '-', …	Each cell should be filled with a consistent value. Pick a way to denote missingness (ideally 'NA') and stick with it. Avoid using numbers or punctuation to denote missing data.
Entering data	Stick to text and numbers	Convey all information with direct text/numerical entry	Using cell highlighting or font color to convey information	Your analyst may not use the same platform for analysis as you used for data entry, so avoiding font color and cell highlighting will minimize issues.
Generating an Excel file	Save the data in an appropriate format	Use one worksheet per table and save as CSV or text files	Multiple worksheets	Statisticians require this format to import your data onto other platforms.
Entering Data	Avoid entering unnecessary lines of text at the start	Start your first row with variable names	Adding lines of text	This violates the rules of tidy data and makes processing more difficult. Include this information in the "Code book" instead.
Opening files in Excel	Know and avoid its pitfalls	Consistently include one value per cell and be careful of date and time data.	Using macros, splitting cells, and merging cells	These formats are not amenable to data analysis on other platforms.

Figure 2: Box1

2.3 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 (also referred to as a 'data dictionary') contains this information. At a 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 blood example, the statistician would want to know what the unit of measurement for each demographic variable is (age in years, treatment dose, level of diagnosis and how heterogeneous). They would also want to know any other information about how you did the data collection and study design. For example, are these the first 20 patients with diabetes that walked into the clinic? Are they 20 highly selected patients by some characteristic like age? Are they randomized to treatments? This is the place for any detail about either the experimental design or the data itself that may be informative to the statistician.

A common format for this document is a Word file. There should be a section called "Study design" that has a thorough description of the question being asked by the study as well as how you collected the data. An additional section called "Code book" should be provided to describe each variable and its units. This information is frequently conveyed most simply in tabular form. In this case, the columns of the table would contain columns including `VariableName`, `Description`, `Units`, `CodingNotes`, and `OtherNotes`. Further columns that provide additional information to the statistician should be included. (**Figure 1C**)

2.3.1 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. Missing data are data that are unobserved and you don't know the mechanism. Missing values should be coded as `NA`. If, however, missingness is coded in an alternative manner, this should be explicitly noted in the code book. 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, or throw away missing observations.

Explanations for the reasoning behind variable coding guidelines can be found in **Box2**. Generally, 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 coding is attractive because it is self-documenting; any ambiguity or need for interpretation by the analyst is removed. This will ultimately avoid potential mix-ups 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 ob-

served 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.

2.4 The instruction list

You may have heard this before, but reproducibility is a big deal in computational science (Peng 2011). To accomplish this goal, best practices have been discussed in detail previously (Wilson et al. 2016). However, for simplicity here, this means that 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 or data analysis steps before the data can be considered tidy and passed off to your statistician.

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. Ideally, this script would be run a couple of times to ensure the code produces the same output.

Alternatively, in many cases, the person who collected the data may not know how to code in a scripting language. However, he or she still has incentive to make the data tidy for a statistician to speed the process of collaboration. In this case, the statistician should be provided something called pseudocode, which is simply a simple explanation, often broken down into steps, to explain what has been done to the data (**Figure 1D**). 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, the specific version of any software used, and whether you tried it

more than once to confirm it gave the same results. Ideally, you will run this by a fellow student or labmate to confirm that they can obtain the same output file you did.

3 What you should expect from the analyst

When you turn over a properly tidied data set it dramatically decreases the workload on the statistician and minimizes the likelihood of errors during analysis. By taking the time to tidy the data, the data generator, who knows the details of the data generated better than anyone else, can expect to get the analysis back sooner and can be more confident in its accuracy. 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 and 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 or your principal investigator.

4 Discussion

These guidelines aim to provide guidelines for effective and efficient data sharing between those generating data and those analyzing it. We highlight the need for data generators to (1) provide data in a tidy and consistently coded format, (2) include all the necessary experimental information regarding data generation, and (3) to explain any steps taken to

pre-process the data. If followed, these guidelines will both speed up analysis turnaround time and minimize the likelihood of errors during analysis.

5 Funding

This work was supported by NIH R01 GM105705.

References

Baggerly, K. (2010), 'Disclose all data in publications', *Nature* **467**(7314), 401.

Browman, K. & Woo, K. (n.d.), 'Data organization in spreadsheets'.
URL: *https://github.com/dsscollection/data-org-spreadsheets*

JSON: The Fat-Free Alternative to XML (n.d.).
URL: *http://www.json.org/xml.html*

Leek, J. T. & Peng, R. D. (2015), 'Opinion: Reproducible research can still be wrong: Adopting a prevention approach', *Proceedings of the National Academy of Sciences* **112**(6), 1645–1646.

Leek Group (n.d.).
URL: *http://jtleek.com/*

Newman, C. & Klyne, G. (n.d.), 'Date and Time on the Internet: Timestamps'.
URL: *https://tools.ietf.org/html/rfc3339.html*

Peng, R. D. (2011), 'Reproducible Research in Computational Science', *Science* **334**(6060), 1226–1227.
URL: *http://science.sciencemag.org/content/334/6060/1226*

R: The R Project for Statistical Computing (n.d.).
URL: *https://www.r-project.org/*

Twitter (n.d.).

URL: *https://twitter.com/*

White, E. P., Baldridge, E., Brym, Z. T., Locey, K. J., McGlinn, D. J. & Supp, S. R. (2013), 'Nine simple ways to make it easier to (re)use your data', *Ideas in Ecology and Evolution* **6**(2).

URL: *http://ojs.library.queensu.ca/index.php/IEE/article/view/4608*

Wickham, H. (2014), 'Tidy data', *Journal of Statistical Software* **59**(10).

URL: *https://www.jstatsoft.org/article/view/v059i10*

Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L. & Teal, T. K. (2016), 'Good Enough Practices in Scientific Computing', *arXiv:1609.00037 [cs]* . arXiv: 1609.00037.

URL: *http://arxiv.org/abs/1609.00037*

Zeeberg, B. R., Riss, J., Kane, D. W., Bussey, K. J., Uchio, E., Linehan, W. M., Barrett, J. C. & Weinstein, J. N. (2004), 'Mistaken Identifiers: Gene name errors can be introduced inadvertently when using Excel in bioinformatics', *BMC Bioinformatics* **5**, 80.

URL: *http://dx.doi.org/10.1186/1471-2105-5-80*