The American Statistician

ISSN: 0003-1305 (Print) 1537-2731 (Online) Journal homepage: https://www.tandfonline.com/loi/utas20

Data Organization in Spreadsheets

Karl W. Broman & Kara H. Woo

To cite this article: Karl W. Broman & Kara H. Woo (2018) Data Organization in Spreadsheets, The American Statistician, 72:1, 2-10, DOI: 10.1080/00031305.2017.1375989

To link to this article: https://doi.org/10.1080/00031305.2017.1375989

THE AMERICAN STATISTICIAN
2018, VOL. 72, NO. 1, 2–10
https://doi.org/10.1080/00031305.2017.1375989

Taylor & Francis
Taylor & Francis Group

Data Organization in Spreadsheets

Karl W. Broman[a] and Kara H. Woo[b]

[a]Department of Biostatistics & Medical Informatics, University of Wisconsin-Madison, Madison, WI; [b]Information School, University of Washington, Seattle, WA

ABSTRACT

Spreadsheets are widely used software tools for data entry, storage, analysis, and visualization. Focusing on the data entry and storage aspects, this article offers practical recommendations for organizing spreadsheet data to reduce errors and ease later analyses. The basic principles are: be consistent, write dates like YYYY-MM-DD, do not leave any cells empty, put just one thing in a cell, organize the data as a single rectangle (with subjects as rows and variables as columns, and with a single header row), create a data dictionary, do not include calculations in the raw data files, do not use font color or highlighting as data, choose good names for things, make backups, use data validation to avoid data entry errors, and save the data in plain text files.

ARTICLE HISTORY
Received June 2017
Revised August 2017

KEYWORDS
Data management; Data organization; Microsoft Excel; Spreadsheets

1. Introduction

Spreadsheets, for all of their mundane rectangularness, have been the subject of angst and controversy for decades. Some writers have admonished that "real programmers don't use spreadsheets" and that we must "stop that subversive spreadsheet" (Casimir 1992; Chadwick 2003). Others have advised researchers on how to use spreadsheets to improve their productivity (Wagner and Keisler 2006). Amid this debate, spreadsheets have continued to play a significant role in researchers' workflows, and it is clear that they are a valuable tool that researchers are unlikely to abandon completely.

The dangers of spreadsheets are real, however—so much so that the European Spreadsheet Risks Interest Group keeps a public archive of spreadsheet "horror stories" (*http://www.eusprig.org/horror-stories.htm*). Many researchers have examined error rates in spreadsheets, and Panko (2008) reported that in 13 audits of real-world spreadsheets, an average of 88% contained errors. Popular spreadsheet programs also make certain types of errors easy to commit and difficult to rectify. Microsoft Excel converts some gene names to dates and stores dates differently between operating systems, which can cause problems in downstream analyses (Zeeberg et al. 2004; Woo 2014). Researchers who use spreadsheets should be aware of these common errors and design spreadsheets that are tidy, consistent, and as resistant to mistakes as possible.

Spreadsheets are often used as a multipurpose tool for data entry, storage, analysis, and visualization. Most spreadsheet programs allow users to perform all of these tasks, however we believe that spreadsheets are best suited to data entry and storage, and that analysis and visualization should happen separately. Analyzing and visualizing data in a separate program, or at least in a separate copy of the data file, reduces the risk of contaminating or destroying the raw data in the spreadsheet.

Murrell (2013) contrasted data that are formatted for humans to view by eye with data that are formatted for a computer. He provided an extended example of computer code to extract data from a set of files with complex arrangements. It is important that data analysts be able to work with such complex data files. But if the initial arrangement of the data files is planned with the computer in mind, the later analysis process is simplified.

In this article, we offer practical recommendations for organizing spreadsheet data in a way that both humans and computer programs can read. By following this advice, researchers will create spreadsheets that are less error-prone, easier for computers to process, and easier to share with collaborators and the public. Spreadsheets that adhere to our recommendations will work well with the tidy tools and reproducible methods described elsewhere in this collection and will form the basis of a robust and reproducible analytic workflow.

For an existing dataset whose arrangement could be improved, we recommend against applying tedious and potentially error-prone hand-editing to revise the arrangement. Rather, we hope that the reader might apply these principles when designing the layout for future datasets.

2. Be Consistent

The first rule of data organization is *be consistent*. Whatever you do, do it consistently. Entering and organizing your data in a consistent way from the start will prevent you and your collaborators from having to spend time harmonizing the data later.

Use consistent codes for categorical variables. For a categorical variable like the sex of a mouse in a genetics study, use a single common value for males (e.g., "`male`"), and a single common value for females (e.g., "`female`"). Do not sometimes write "`M`,"

CONTACT Karl W. Broman *kbroman@biostat.wisc.edu* Department of Biostatistics & Medical Informatics, University of Wisconsin-Madison, Madison, WI 53706.
Color versions of one or more of the figures in the article can be found online at *www.tandfonline.com/r/TAS*.

sometimes "male," and sometimes "Male." Pick one and stick to it.

Use a consistent fixed code for any missing values. We prefer to have every cell filled in, so that one can distinguish between truly missing values and unintentionally missing values. R users prefer "NA." You could also use a hyphen. But stick with a single value throughout. Definitely do not use a numeric value like -999 or 999; it is easy to miss that it is intended to be missing. Also, do not insert a note in place of the data, explaining why it is missing. Rather, make a separate column with such notes.

Use consistent variable names. If in one file (e.g., the first batch of subjects), you have a variable called "Glucose_10wk," then call it exactly that in other files (e.g., for other batches of subjects). If it is variably called "Glucose_10wk," "gluc_10weeks," and "10 week glucose," then downstream the data analyst will have to work out that these are all really the same thing.

Use consistent subject identifiers. If sometimes it is "153" and sometimes "mouse153" and sometimes "mouse-153F" and sometimes "Mouse153," there is going to be extra work to figure out who is who.

Use a consistent data layout in multiple files. If your data are in multiple files and you use different layouts in different files, it will be extra work for the analyst to combine the files into one dataset for analysis. With a consistent structure, it will be easy to automate this process.

Use consistent file names. Have some system for naming files. If one file is called "Serum_batch1_2015-01-30.csv," then do not call the file for the next batch "batch2_serum_52915.csv" but rather use "Serum_batch2_2015-05-29.csv." Keeping a consistent file naming scheme will help ensure that your files remain well organized, and it will make it easier to batch process the files if you need to.

Use a consistent format for all dates, preferably with the standard format YYYY-MM-DD, for example, 2015-08-01. If sometimes you write 8/1/2015 and sometimes 8-1-15, it will be more difficult to use the dates in analyses or data visualizations.

Use consistent phrases in your notes. If you have a separate column of notes (e.g., "dead" or "lo off curve"), be consistent in what you write. Do not sometimes write "dead" and sometimes "Dead," or sometimes "lo off curve" and sometimes "off curve lo."

Be careful about extra spaces within cells. A blank cell is different than a cell that contains a single space. And "male" is different from " male " (i.e., with spaces at the beginning and end).

3. Choose Good Names for Things

It is important to pick good names for things. This can be hard, and so it is worth putting some time and thought into it.

As a general rule, do not use spaces, either in variable names or file names. They make programming harder: the analyst will need to surround everything in double quotes, like "glucose 6 weeks", rather than just writing glucose_6_weeks. Where you might use spaces, use underscores or perhaps hyphens. But do not use a mixture of underscores and hyphens; pick one and be consistent.

Table 1. Examples of good and bad variable names.

good name	good alternative	avoid
Max_temp_C	MaxTemp	Maximum Temp (°C)
Precipitation_mm	Precipitation	precmm
Mean_year_growth	MeanYearGrowth	Mean growth/year
sex	sex	M/F
weight	weight	w.
cell_type	CellType	Cell type
Observation_01	first_observation	1st Obs.

Be careful about extraneous spaces at the beginning or end of a variable name. "glucose" is different from "glucose " (with an extra space at the end).

Avoid special characters, except for underscores and hyphens. Other symbols ($, @, %, #, &, *, (,), !, /, etc.) often have special meaning in programming languages, and so they can be harder to handle. They are also a bit harder to type.

The main principle in choosing names, whether for variables or for file names, is *short, but meaningful.* So not *too* short. The Data Carpentry lesson on using spreadsheets (see *http://www.datacarpentry.org/spreadsheet-ecology-lesson/02-common-mistakes*) has a nice table with good and bad example variable names, reproduced in Table 1. We agree with all of this, though we would maybe cut down on some of the capitalization. So maybe max_temp, precipitation, and mean_year_growth.

Finally, never include "final" in a file name. You will invariably end up with "final_ver2." (We cannot say that without referring to the widely cited PHD comic, *http://bit.ly/phdcom_final*.)

4. Write Dates as YYYY-MM-DD

When entering dates, we strongly recommend using the global "ISO 8601" standard, YYYY-MM-DD, such as 2013-02-27. (See the related xkcd comic, *https://xkcd.com/1179*.)

Microsoft Excel's treatment of dates can cause problems in data (see *https://storify.com/kara_woo/excel-date-system-fiasco*). It stores them internally as a number, with different conventions on Windows and Macs. So, you may need to manually check the integrity of your data when they come out of Excel.

Excel also has a tendency to turn other things into dates. For example, some gene symbols (e.g., "Oct-4") may be interpreted as dates and reformatted. Ziemann, Eren, and El-Osta (2016) studied gene lists contained within the supplementary files from 18 journals for the years 2005–2015, and found that ~20% of the lists had errors in the gene names, related to the conversion of gene symbols to dates or floating-point numbers.

We often prefer to use a plain text format for columns in an Excel worksheet that are going to contain dates, so that it does not do anything to them. To do this:

- Select the column
- In the menu bar, select Format → Cells
- Choose "Text" on the left

However, if you do this on columns that *already* contain dates, Excel will convert them to a text value of their underlying numeric representation.

	A	B	C
1	Date	Assay date	Weight
2		12/9/05	54.9
3		12/9/05	45.3
4	12/6/2005	e	47
5		e	45.7
6		e	52.9
7		1/11/2006	46.1
8		1/11/2006	38.6

Figure 1. A spreadsheet with inconsistent date formats. This spreadsheet does not adhere to our recommendations for consistency of date format.

Another way to force Excel to treat dates as text is to begin the date with an apostrophe, like this: `'2014-06-14` (see *http://bit.ly/twitter_apos*). Excel will treat the cells as text, but the apostrophe will not appear when you view the spreadsheet or export it to other formats. This is a handy trick, but it requires impeccable diligence and consistency. Alternatively, you could create three separate columns with year, month, and day. Those will be ordinary numbers, and so Excel will not mess them up. Finally, you could represent dates as an 8-digit integer of the form `YYYYMMDD`, for example, `20140614` for `2014-06-14` (see Briney 2017).

Figure 1 displays a portion of a spreadsheet that we got from a collaborator. We do not quite remember what those e's were for, but in any case having different date formats within a column makes it more difficult to use the dates in later analyses or data visualizations. Use care about dates, and be consistent.

5. No Empty Cells

Fill in all cells. Use some common code for missing data. Not everyone agrees with us on this point (e.g., White et al. (2013) stated a preference for leaving cells blank), but we would prefer to have "`NA`" or even a hyphen in the cells with missing data, to make it clear that the data are known to be missing rather than unintentionally left blank.

Figure 2 contains two examples of spreadsheets with some empty cells. In Figure 2(a), cells were left blank when a single value was meant to be repeated multiple times. Please do not do this! It is additional work for the analyst to determine the implicit values for these cells. Moreover, if the rows are sorted at some point there may be no way to recover the dates that belong in the empty cells.

The spreadsheet in Figure 2(b) has a complex layout with information for different treatments. It is perhaps clear that columns B-E all concern the "1 min" treatment, and columns F-I all concern "5 min," and that columns B, C, F, and G all concern "normal," while columns D, E, H, and I concern "mutant." But while it may be easy to see by eye, it can be hard to deal with this in later analyses.

You could fill in some of those cells, to make it more clear. Alternatively, make a "tidy" version of the data (Wickham 2014), with each row being one replicate and with the response values all in one column, as in Figure 3. We will discuss this further in Section 7.

6. Put Just One Thing in a Cell

The cells in your spreadsheet should each contain one piece of data. Do not put more than one thing in a cell.

For example, you might have a column with "`plate position`" as "`plate-well`," such as "`13-A01`." It would be better to separate this into "`plate`" and "`well`" columns (containing "`13`" and "`A01`"), or even "`plate`," "`well_row`," and "`well_column`" (containing "`13`," "`A`," and "`1`"). Or you might be tempted to include units, such as "`45 g`." It is better to write `45` and put the units in the column name, such as `body_weight_g`. It is even better to leave the column as `body_weight` and put the units in a separate data dictionary (see Section 8). Another common situation is to include a note within a cell, with the data, like "`0 (below threshold)`." Instead, write "`0`" and include a separate column with such notes.

Finally, do not merge cells. It might look pretty, but you end up breaking the rule of *no empty cells*.

7. Make it a Rectangle

The best layout for your data within a spreadsheet is as a single big rectangle with rows corresponding to subjects and columns corresponding to variables. The first row should contain variable names, and *please do not use more than one row for the variable names.* An example of a rectangular layout is shown in Figure 4.

Some datasets will not fit nicely into a single rectangle, but they will usually fit into a set of rectangles, in which case you can make a set of Excel files, each with a rectangle of data. It is best to keep each rectangle in its own file; tables scattered around a worksheet are difficult to work with, and they make it hard to export the data to CSV files. You might also consider having a single Excel file with multiple worksheets. We prefer to have multiple files with one sheet each so we can more easily save the data as CSV files, but if you do use multiple worksheets in a file be sure to use a consistent structure.

Some data do not even fit into a set of rectangles, but then maybe spreadsheets are not the best format for them, as spreadsheets are inherently rectangular.

The data files that we receive are usually *not* in rectangular form. More often, there seem to be bits of data sprinkled about. Several examples are shown in Figure 5. In the spreadsheets in Figure 5(a) and 5(b), the data analyst will need to study the layout, work out what everything means, and then spend some time rearranging things. If, from the start, the data were organized as a rectangle, it would save the analyst a great deal of time. The example in Figure 5(c) was based on a dataset that had a separate worksheet for each subject, each in that complicated format. If all of the worksheets have exactly the same layout, then it is not too hard to pull out the relevant information and combine it into a rectangle. (One might write a script in R, Python, or Ruby.) But it is preferable to not have means and SDs and fold change calculations cluttering up the raw data values, and it seems that even for data entry, it would be easier to have all of the measurements on one worksheet. Sometimes it is hard to see how to

A

	A	B	C
1	id	date	glucose
2	101	2015–06–14	149.3
3	102		95.3
4	103	2015–06–18	97.5
5	104		117.0
6	105		108.0
7	106	2015–06–20	149.0
8	107		169.4

B

	A	B	C	D	E	F	G	H	I
1		1 min				5 min			
2	strain	normal		mutant		normal		mutant	
3	A	147	139	166	179	334	354	451	474
4	B	246	240	178	172	514	611	412	447

Figure 2. Examples of spreadsheets that violate the "no empty cells" recommendation. (a) A spreadsheet where only the first of several repeated values was included. (b) A spreadsheet with a complicated layout and some implicit column headers. For a tidy version of this data, see Figure 3.

	A	B	C	D	E
1	strain	genotype	min	replicate	response
2	A	normal	1	1	147
3	A	normal	1	2	139
4	B	normal	1	1	246
5	B	normal	1	2	240
6	A	mutant	1	1	166
7	A	mutant	1	2	179
8	B	mutant	1	1	178
9	B	mutant	1	2	172
10	A	normal	5	1	334
11	A	normal	5	2	354
12	B	normal	5	1	514
13	B	normal	5	2	611
14	A	mutant	5	1	451
15	A	mutant	5	2	474
16	B	mutant	5	1	412
17	B	mutant	5	2	447

Figure 3. A tidy version of the data in Figure 2(b).

reorganize things as a rectangle, as in the example in Figure 5(d). It is sort of a rectangle; we could fill in the empty cells in the first two columns by repeating the individual, date, and weight values. But it seems wrong to repeat the weights, since they are not repeated measurements.

It is perhaps better to make two separate tables, one with the weights, and one with these other measurements (which are for an in vivo assay, the glucose tolerance test: give a mouse some glucose and measure serum glucose and insulin levels

	A	B	C	D	E
1	id	sex	glucose	insulin	triglyc
2	101	Male	134.1	0.60	273.4
3	102	Female	120.0	1.18	243.6
4	103	Male	124.8	1.23	297.6
5	104	Male	83.1	1.16	142.4
6	105	Male	105.2	0.73	215.7

Figure 4. An example spreadsheet with a rectangular layout. This layout will aid future analyses.

at different times afterward). An example of this is shown in Figure 6. Note that we have also changed the handling of the "`lo off curve`" and "`off curve lo`" notes that were within the insulin column, by inserting "`NA`" and adding a "note" column (and being consistent in the text used in the note). We also added a column name for the first column with subject identifiers.

The layouts in Figure 6(a) and 6(b) are examples of "tidy" data (Wickham 2014): each row is an experimental unit, which is usually just a subject but in the case of Figure 6(b) is a single assay measurement on a subject. Reorganizing the data into a "tidy" format can simplify later analysis. But the rectangular aspect is the most important part.

Another issue we often see is the use of two rows of header names, as in Figure 7. In this sort of situation, we often see merged cells: merging the "`week 4`" cell with the two cells following, so that the text is centered above the three columns with "`date`," "`weight`," and "`glucose`." We would prefer to have the week information within the variable name. So, for example, there could be a single header row containing `Mouse ID`, `SEX`, `date_4`, `weight_4`, `glucose_4`, `date_6`, `weight_6`,

A

	A	B	C	D	E	F
1						
2		101	102	103	104	105
3	sex	Male	Female	Male	Male	Male
4						
5		101	102	103	104	105
6	glucose	134.1	120.0	124.8	83.1	105.2
7						
8		101	102	103	104	105
9	insulin	0.60	1.18	1.23	1.16	0.73

B

	A	B	C	D	E	F	G
1	1MIN						
2			Normal			Mutant	
3	B6	146.6	138.6	155.6	166	179.3	186.9
4	BTBR	245.7	240	243.1	177.8	171.6	188.1
5							
6	5MIN						
7			Normal			Mutant	
8	B6	333.6	353.6	408.8	450.6	474.4	423.8
9	BTBR	514.4	610.6	597.9	412.1	447.4	446.5

C

	A	B	C	D	E	F	G
1							
2	Date	11/3/14					
3	Days on diet	126					
4	Mouse #	43					
5	sex	f					
6	experiment		values			mean	SD
7	control		0.186	0.191	1.081	0.49	0.52
8	treatment A		7.414	1.468	2.254	3.71	3.23
9	treatment B		9.811	9.259	11.296	10.12	1.05
10							
11	fold change		values			mean	SD
12	treatment A		15.26	3.02	4.64	7.64	6.65
13	treatment B		20.19	19.05	23.24	20.83	2.17

D

	A	B	C	D	E	F
1		GTT date	GTT weight	time	glucose mg/dl	insulin ng/ml
2	321	2/9/15	24.5	0	99.2	lo off curve
3				5	349.3	0.205
4				15	286.1	0.129
5				30	312	0.175
6				60	99.9	0.122
7				120	217.9	lo off curve
8	322	2/9/15	18.9	0	185.8	0.251
9				5	297.4	2.228
10				15	439	2.078
11				30	362.3	0.775
12				60	232.7	0.5
13				120	260.7	0.523
14	323	2/9/15	24.7	0	198.5	0.151
15				5	530.6	off curve lo

Figure 5. Examples of spreadsheets with nonrectangular layouts. These layouts are likely to cause problems in analysis.

etc. Alternatively, make it a "tidy" dataset with each row being a subject on a particular day, as shown in Figure 8.

Have sympathy for your analyst (which could be yourself): organize your data as a rectangle (or, if necessary, as a set of rectangles).

8. Create a Data Dictionary

It is helpful to have a separate file that explains what all of the variables are. It is helpful if this is laid out in rectangular form, so that the data analyst can make use of it in analyses.

Such a "data dictionary" might contain:

- The exact variable name as in the data file
- A version of the variable name that might be used in data visualizations
- A longer explanation of what the variable means
- The measurement units
- Expected minimum and maximum values

This is part of the *metadata* that you will want to prepare: information *about* the data. You will also want a `ReadMe` file that includes an overview of the project and data.

An example data dictionary is displayed in Figure 9. Note that this is a rectangular dataset, like any other. The first column contains the variable names. The second column is a more

A

	A	B	C
1	id	GTT date	GTT weight
2	321	2/9/15	24.5
3	322	2/9/15	18.9
4	323	2/9/15	24.7

B

	A	B	C	D	E
1	id	GTT time	glucose mg/dl	insulin ng/ml	note
2	321	0	99.2	NA	insulin below curve
3	321	5	349.3	0.205	
4	321	15	286.1	0.129	
5	321	30	312	0.175	
6	321	60	99.9	0.122	
7	321	120	217.9	NA	insulin below curve
8	322	0	185.8	0.251	
9	322	5	297.4	2.228	
10	322	15	439	2.078	
11	322	30	362.3	0.775	
12	322	60	232.7	0.5	
13	322	120	260.7	0.523	
14	323	0	198.5	0.151	
15	323	5	530.6	NA	insulin below curve

Figure 6. Reorganization of Figure 5(d) as a pair of rectangles.

	A	B	C	D	E	F	G	H	I	J	K
1			week 4			week 6			week 8		
2	Mouse ID	SEX	date	weight	glucose	date	weight	glucose	date	weight	glucose
3	3005	M	3/30/2007	19.3	635	4/11/2007	31	460.7	4/27/2007	39.6	530.2
4	3017	M	10/6/2006	25.9	202.4	10/19/2006	45.1	384.7	11/3/2006	57.2	458.7
5	3434	F	11/22/2006	26.6	238.9	12/6/2006	45.9	378	12/22/2006	56.2	409.8
6	3449	M	1/5/2007	27.5	121	1/19/2007	42.9	191.3	2/2/2007	56.7	182.5
7	3499	F	1/5/2007	19.8	220.2	1/19/2007	36.6	556.9	2/2/2007	43.6	446

Figure 7. A spreadsheet with two header rows. It is better to have a single header row. See Figure 8 for a tidy data layout that eliminates the need for multiple header rows and repeated column headers.

readable version, as might be used in data visualizations. The third column groups the variables into different categories, which might also be used in data visualizations. The last column is a description.

Lots of other information could be included. For example, information about the allowed values for the variables would be helpful in identifying data entry errors.

9. No Calculations in the Raw Data Files

Often, the Excel files that our collaborators send us include all kinds of calculations and graphs. We feel strongly that your primary data file should contain *just the data* and nothing else: no calculations, no graphs.

If you are doing calculations in your data file, that likely means you are regularly opening it and typing into it. Doing so incurs some risk that you will accidentally type junk into your data.

(Has this happened to you? You open an Excel file and start typing and nothing happens, and then you select a cell and you can start typing. Where did all of that initial text go? Well, sometimes it got entered into some random cell, to be discovered later during data analysis.)

Your primary data file should be a pristine store of data. Write-protect it, back it up, and do not touch it.

If you want to do some analyses in Excel, make a copy of the file and do your calculations and graphs in the copy.

10. Do Not Use Font Color or Highlighting as Data

You might be tempted to highlight particular cells with suspicious data, or rows that should be ignored. Or the font or font

	A	B	C	D	E	F
1	mouse_id	sex	week	date	glucose	weight
2	3005	M	4	3/30/2007	19.3	635
3	3005	M	6	4/11/2007	31	460.7
4	3005	M	8	4/27/2007	39.6	530.2
5	3017	M	4	10/6/2006	25.9	202.4
6	3017	M	6	10/19/2006	45.1	384.7
7	3017	M	8	11/3/2006	57.2	458.7
8	3434	F	4	11/22/2006	26.6	238.9
9	3434	F	6	12/6/2006	45.9	378
10	3434	F	8	12/22/2006	56.2	409.8
11	3449	M	4	1/5/2007	27.5	121
12	3449	M	6	1/19/2007	42.9	191.3
13	3449	M	8	2/2/2007	56.7	182.5
14	3499	F	4	1/5/2007	19.8	220.2
15	3499	F	6	1/19/2007	36.6	556.9
16	3499	F	8	2/2/2007	43.6	446

Figure 8. A tidy version of the data in Figure 7.

	A	B	C	D
1	name	plot_name	group	description
2	mouse	Mouse	demographic	Animal identifier
3	sex	Sex	demographic	Male (M) or Female (F)
4	sac_date	Date of sac	demographic	Date mouse was sacrificed
5	partial_inflation	Partial inflation	clinical	Indicates if mouse showed partial pancreatic inflation
6	coat_color	Coat color	demographic	Coat color, by visual inspection
7	crumblers	Crumblers	clinical	Indicates if mouse stored food in their bedding
8	diet_days	Days on diet	clinical	Number of days on high–fat diet

Figure 9. An example data dictionary.

A

	A	B	C
1	id	date	glucose
2	101	2015–06–14	149.3
3	102	2015–06–14	95.3
4	103	2015–06–18	97.5
5	104	2015–06–18	1.1
6	105	2015–06–18	108.0
7	106	2015–06–20	149.0
8	107	2015–06–20	169.4

B

	A	B	C	D
1	id	date	glucose	outlier
2	101	2015–06–14	149.3	FALSE
3	102	2015–06–14	95.3	FALSE
4	103	2015–06–18	97.5	FALSE
5	104	2015–06–18	1.1	TRUE
6	105	2015–06–18	108.0	FALSE
7	106	2015–06–20	149.0	FALSE
8	107	2015–06–20	169.4	FALSE

Figure 10. Highlighting in spreadsheets. (a) A potential outlier indicated by highlighting the cell. (b) The preferred method for indicating outliers, via an additional column.

color might have some meaning. Instead, add another column with an indicator variable (e.g., `"trusted"` with values `TRUE` or `FALSE`).

For example, in Figure 10(a), a suspicious entry is highlighted. It would be better to include an additional column that indicates the outliers (as in Figure 10(b)). The highlighting is nice visually, but it is hard to extract that information for use in the later analysis. Analysis programs can much more readily handle data that are stored in a column than data encoded in cell highlighting, font, etc. (and in fact this markup will be lost completely in many programs).

Another possible use of highlighting would be to indicate males and females in a mouse study by highlighting the corresponding rows in different colors. But rather than use highlighting to indicate sex, it is better to include a `sex` column, with values `Male` or `Female`.

11. Make Backups

Make regular backups of your data. In multiple locations. And consider using a formal version control system, like git, though it is not ideal for data files. If you want to get a bit fancy, maybe look at dat (*https://datproject.org/*).

A

	A	B	C	D	E
1	id	sex	glucose	insulin	triglyc
2	101	Male	134.1	0.60	273.4
3	102	Female	120.0	1.18	243.6
4	103	Male	124.8	1.23	297.6
5	104	Male	83.1	1.16	142.4
6	105	Male	105.2	0.73	215.7

B

```
id,sex,glucose,insulin,triglyc
101,Male,134.1,0.60,273.4
102,Female,120.0,1.18,243.6
103,Male,124.8,1.23,297.6
104,Male,83.1,1.16,142.4
105,Male,105.2,0.73,215.7
```

Figure 11. (a) An example spreadsheet. (b) The same data as a plain text file in CSV format.

Keep all versions of the data files, so that if something gets corrupted (e.g., you accidentally type over some of the data and do not notice it until much later), you will be able to go back and fix it. Before you start inserting more data, make a copy of the file with a new version number: `file_v1.xlsx`, `file_v2.xlsx`, ...

When you are not actively entering data, and particularly when you are done entering data, *write-protect the file*. That way, you will not accidentally change things.

- On a Mac, right-click on the file in Finder and select "Get Info." In the menu that opens, there is a section at the bottom on "Sharing & Permissions." Click on "Privilege" for yourself and select "Read only."
- In Windows, right-click on the file in Windows Explorer and select "Properties." In the "General" tab, there is a section at the bottom with "Attributes." Select the box for "Read-only" and click the "OK" button.

Back up your data!

12. Use Data Validation to Avoid Errors

Regarding the task of data entry, it is important to ensure that the process is as error-free and repetitive-stress-injury-free as possible. One useful tool for avoiding data entry errors is the "data validation" feature in Excel (see *http://bit.ly/excel_dataval*), to control the type of data or the values that users can enter into a cell.

- Select a column
- In the menu bar, choose Data → Validation
- Choose appropriate validation criteria. For example,
  - A whole number in some range
  - A decimal number in some range
  - A list of possible values
  - Text, but with a limit on length

At the same time, you could select particular data types for the column, such as `text`, to avoid having dates (or transcription factor names!) get mangled by Excel. We mentioned this before in the discussion of dates, but it is worth repeating:

- Select the column
- In the menu bar, select Format → Cells
- Choose "Text" on the left

This may seem cumbersome, but if it helps you to avoid data entry mistakes, it would be worth it.

13. Save the Data in Plain Text Files

Keep a copy of your data files in a plain text format, with comma or tab delimiters. We generally use comma-delimited (CSV) files. The spreadsheet in Figure 11(a) would be saved as a plain text file with commas separating the fields, as in Figure 11(b).

The CSV format is not pretty to look at, but you can open the file in Excel or another spreadsheet program and view it in the standard way. More importantly, this sort of nonproprietary file format does not and never will require any sort of special software. And CSV files are easier to handle in code.

If any of the cells in your data include commas, Excel will put double-quotes around the contents of each cell when it is saved in CSV format. That requires slightly more finesse to deal with, but it is generally not a concern.

To save an Excel file as a comma-delimited file:

- From the menu bar, File → Save As
- Next to "Format:," click the drop-down menu and select "Comma Separated Values (CSV)"
- Click "Save"
- Excel will say something like, "This workbook contains features that will not work...". Ignore that and click "Continue."
- Quit Excel. It will ask you, "Do you want to save the changes you made?" Click "Don't Save," because you just saved them. (Excel really does not want you to use a format other than its own.)

Note that there is also an option to save as "Tab Delimited Text." Many people prefer that, especially those who work in countries where commas are used a decimal separators.

Also note that, if your Excel file *did* contain critical features that will not work when saved as a plain text file, such as highlighted cells, that is a problem; those features *will* be lost. For your primary data file, keep things simple.

Summary

Spreadsheet programs (such as Microsoft Excel, Google Sheets, and LibreOffice Calc) are valuable tools for entering, organizing, and storing data. They can also be used for calculations, analysis, and visualizations, but we have focused on the data organization aspects here, and we encourage users interested in doing calculations or making data visualizations within spreadsheets to keep their primary data files pristine and data-only, and to do their calculations and visualizations in separate files.

We have offered a number of suggestions for how best to organize data within a spreadsheet. Our primary concerns are to protect the integrity of the data, and to ease later analysis.

Focus primarily on adopting these principles for future projects. While your current data files may not meet these standards, it is best not to use copy-and-paste to rearrange the files. By doing so, there is a good chance of introducing errors. Data rearrangement is best accomplished via code (such as with an R, Python, or Ruby script) so you never lose the record of what you did to the data.

Acknowledgment

The authors thank Lance Waller, Lincoln Mullen, and Jenny Bryan for their comments to improve the article.

References

Briney, K. (2017), "Two Strategies for Working with Dates in Excel," available at *http://dataabinitio.com/?p=798*. [4]

Casimir, R. J. (1992), "Real Programmers Don't Use Spreadsheets," *SIGPLAN Not.*, 27, 10–16. [2]

Chadwick, D. (2003), "Stop That Subversive Spreadsheet!" in *Integrity and Internal Control in Information Systems V*, ed. M. Gertz, IFIP—the International Federation for Information Processing 124. New York: Springer, pp. 205–211. Available at *http://link.springer.com/chapter/10.1007/978-0-387-35693-8_13*. [2]

Murrell, P. (2013), "Data Intended for Human Consumption, Not Machine Consumption," in *Bad Data Handbook*, ed. Q. E. MacCallum, Sebastopol, CA: O'Reilly Media, pp. 31–51. [2]

Panko, R. (2008), "What We Know About Spreadsheet Errors," available at *http://panko.shidler.hawaii.edu/SSR/Mypapers/whatknow.htm* [2]

Wagner, J. M., and Keisler, J. (2006), "Enhance Your Own Research Productivity Using Spreadsheets," in eds. M. P. Johnson, B. Norman, and N. Secomandi, *Models, Methods, and Applications for Innovative Decision Making*, INFORMS Tutorials in Operations Research, Catonsville, MD: INFORMS, pp. 148–162. *https://doi.org/10.1287/educ.1063.0028*. [2]

White, E. P., Baldridge, E., Brym, Z. T., Locey, K. J., McGlinn, D. J., and Supp, S. R. (2013), "Nine Simple Ways to Make It Easier to (Re)use Your Data," *Ideas in Ecology and Evolution*, 6, 1–10. [4]

Wickham, H. (2014), "Tidy Data," *Journal of Statistical Software*, 59, 1–23. [4,5]

Woo, K. H. (2014), "Abandon All Hope, Ye Who Enter Dates in Excel," *Data Pub*, available at *https://datapub.cdlib.org/2014/04/09/abandon-all-hope-ye-who-enter-dates-in-excel/*. [2]

Zeeberg, B. R., Riss, J., Kane, D. W., Bussey, K. J., Uchio, E., Linehan, W. M., Barrett, J. C., and Weinstein, J. N. (2004), "Mistaken Identifiers: Gene Name Errors Can Be Introduced Inadvertently When Using Excel in Bioinformatics," *BMC Bioinformatics*, 5, 80. [2]

Ziemann, M., Eren, Y., and El-Osta, A. (2016), "Gene Name Errors Are Widespread in the Scientific Literature," *Genome Biology*, 17, 177. [3]