Techniques For Collecting Data

John Bear
Statistics Lab
University of Arizona
August 15, 2012

Motivation

If we run a statistical procedure on a set of data that contains errors, we are guaranteed to get a result that does not measure what we want. Moreover, there are always mistakes in any large set of data. Over the last 15 years, many studies have been done on the quality of data in spreadsheets. Typically (see Raymond Panko, 2008) they find that 90% or more of the spreadsheets they check have errors. One study by Lawrence and Lee, 2004, (n= 30 spreadsheets) found that 100% of the sheets they examined contained errors. These studies are mainly on business data, but the results are commensurate with other studies on error rates in software (Raymond and Panko, 2008). The point here is that errors happen, but we can put procedures in place to catch them when they do.

In this paper we assume people are using Excel spreadsheets to collect data, and then using R to analyze the data. We describe ways of using Excel's data validation and formula capabilities to reduce the number of errors in collected data.

To err is human. People can type 11 when they mean 1. They might type Feb 31 when they mean March 31, or use the year 2060 when they mean 1960. A 240 pound man can have his weight entered as 40 by mistake, without the initial 2, or his height can be entered in feet when it should be in centimeters. The question is not whether mistakes like this will occur. They will. The question is not even how often. They happen too often. The question is, how can we catch them. In this paper, we describe ways to find many of these types of errors systematically.

The other time saving consideration we describe is how to avoid doing some of the other things that lead to complications in R. For instance, spaces can cause problems. So can NA. So can column names with special characters like # and *$*.

Excel Spreadsheet Into R

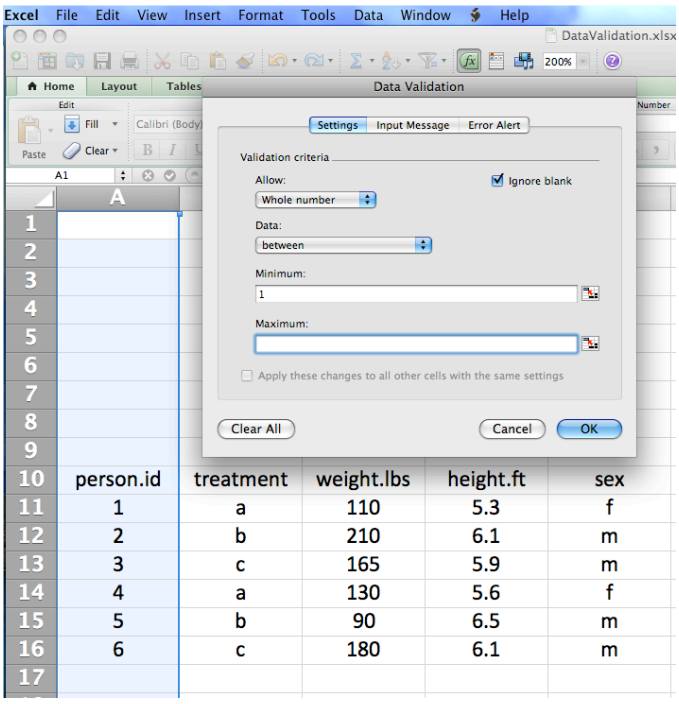
A common scenario is that data (numbers for measurements, like weights and heights, and words for categories, like *male* and *female*, or *before* and *after*) are collected in a spreadsheet, and then that spreadsheet is read into R (the statistical programming language) where the bulk of the statistical analysis is done. In this case we can use two of Excel's capabilities to dramatically reduce errors. One is the data validation functionality that lets you specify what sorts of values are

appropriate for a given column (or group of cells), and then lets you specify how you want to be notified if an inappropriate value is entered into a cell. The other is Excel's formula capability, which we can use to display counts of the cells that do not have values in the appropriate range. We can also use Excel's formulas to display simple summary statistics above each column as a way of helping the data recorder check for anomalies.

Checking as Data is Entered

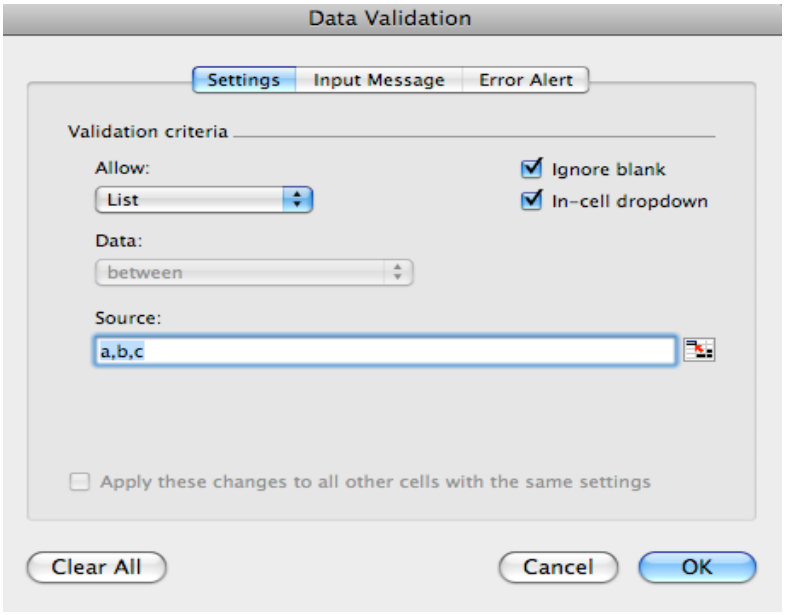
Here we show how to get Excel to check your entries as you put them in, and let you know if something seems questionable. We start with a sample spreadsheet of made-up information about made-up people. The columns are: person.id, treatment, height, weight, and sex. To get the dialogue box in Figure 1, at the top menu bar for Excel we clicked on Data, and then in the drop-down menu we clicked on Validation. In this picture we are specifying that the values in the person.id column should be whole numbers, and the smallest should be 1. It turns out that even though I don't really want to specify a maximum value for id's Excel seems to force me to put in a maximum, so I put in 100.

Figure 1: Specifying the possible values for person.id



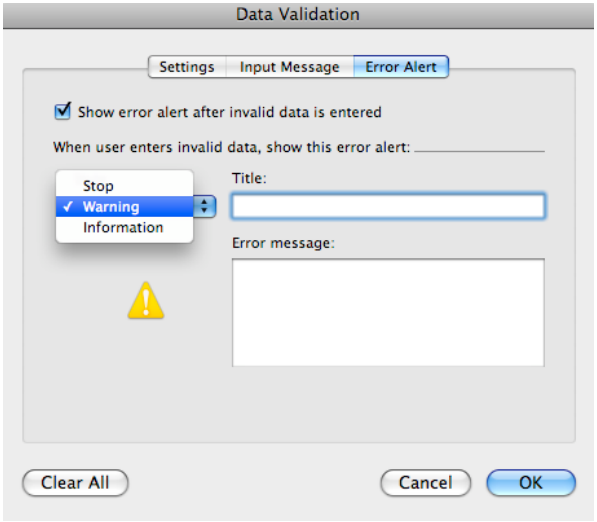
For the treatment column (see Figure 2), I want to specify that the only possible values are *a*, *b*, and *c*. That is done by selecting *list* in the Data:Validation Settings dialogue, and then typing in the alternatives, separated by commas.

Figure 2: Constraining the values for treatment



For the other columns, *weight.lbs*, *height.ft*, and *sex*, we can also specify appropriate ranges. We cannot say for sure what the minimum weight or height for a person in the study will be, but we can give a range of likely values. So in this case I specified that weights can go from 100 lb. to 300 lb., and heights can go from 5 feet to 7 feet. These ranges need to be flexible. Somebody could weigh 90 pounds, for instance. In order to allow that, we keep the range as is, but we click on the Error Alert tab at the top of the data validation dialogue, to change the setting from Error to Warning, as shown in Figure 3. This makes it so that when you enter data into a cell where the

Figure 3: Changing to Warning From Error



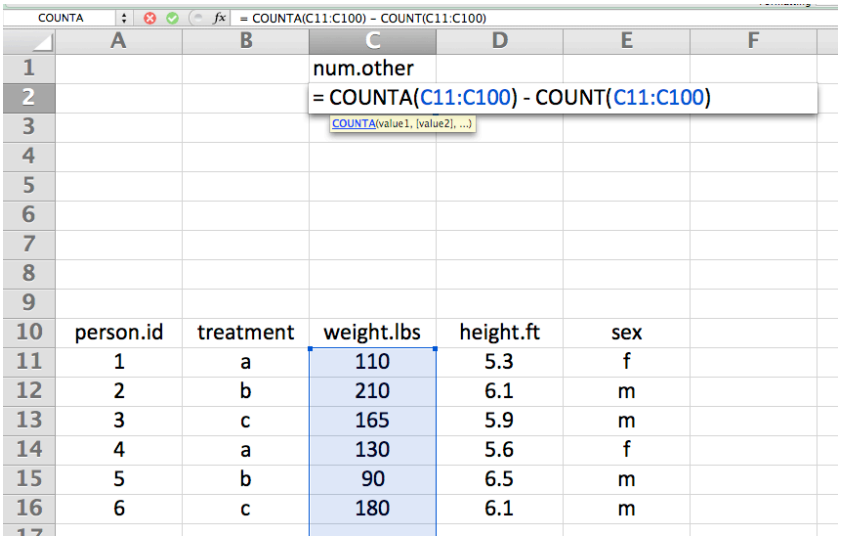
value is outside the prescribed range, Excel will pop up a warning window to let you know. It is essentially saying, *`Are you sure you want to do this?'* If you really do want to, it will let you. Having a warning instead of an error is a trade-off. The error makes it harder to put errors into the spreadsheet, but it also makes it harder to put the formulas in that we will show in the next section. That's why I prefer the warning over the error.

Diagnostic Formulas

In a column of things that are supposed to be numbers, like the column called weight.lbs in Figure 1, we can insert a formula above the column header to check and make sure that everything is in fact a number. A way to do this is to count the number of cells in that column which are not empty, and then count the number of cells containing numbers. When you subtract one from the other, (allowing for the column header, which is not a number) you should get zero. If you don't, then there are some cells that are not empty which contain something other than numbers. That means you know you have errors you need to find.

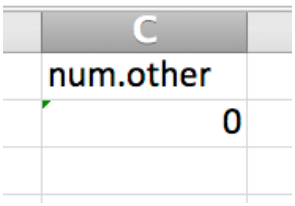
In our example spreadsheet, the headers for the table are in row 10, and then the data are below that, starting in row 11. We use the space above to put in these formulas for checking. The formula I have used here starts counting things at row 11, and arbitrarily figures I'll never have more than 100 elements in the column, so it stops counting at row 100. This may be inelegant, but it beats leaving in errors which you could have found and fixed. Here is the formula I use for checking column C, the weight.lbs column: = COUNTA(C11:C100) - COUNT(C11:C100) . Its use is shown in Figure 4. Above the formula is just a text label to remind people that the cell below contains a count of the things that are of the wrong type and almost certainly need to be fixed.

Figure 4: Formula above weight.lbs column checks for non-numbers.



Once you have put in the formula and hit enter, if there are no errors of this type, a zero is displayed as in Figure 5. In addition there is a mark in the upper left corner

Figure 5: 0 shows column has no obvious `type' errors.



of the cell, indicating that the number displayed has been calculated by a formula, and was not just typed in.

With a column of numbers it is also helpful to be able to see the minimum value, the maximum value, and the mean. These formulas are illustrated in Figures 6, 7, and 8.

Figure 6: Min Formula for weight.lbs

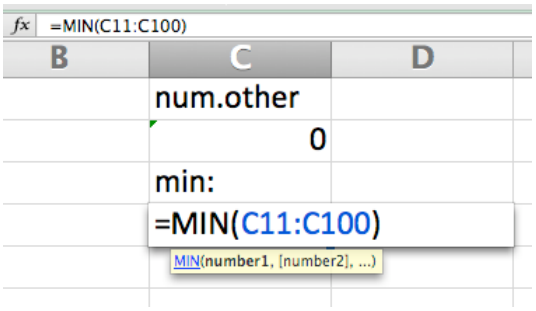


Figure 7: Average formula for weight.lbs

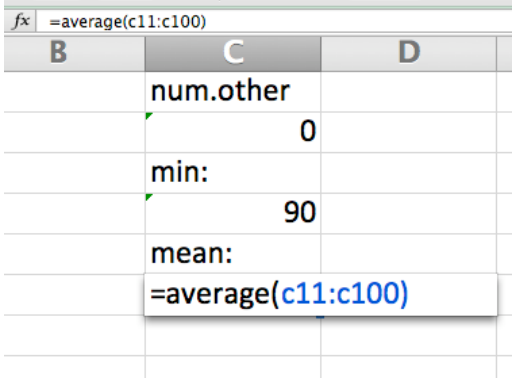
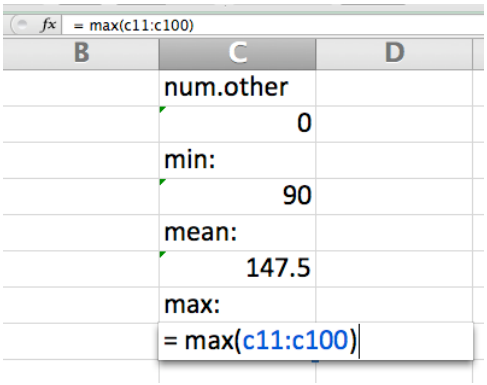


Figure 8: Max formula for weight.lbs



In Figure 9, below, we see how the weight.lbs column appears once we have installed all three formulas. At a glance we can tell that the minimum weight in the column is 90 pounds, the mean is 147.5, and the maximum is 210. We know that 90 pounds is a possible weight for a human adult, so that isn't an error for certain, but if we look at the whole row, we see that the 90 pound weight is for a male who is 6 and a half feet tall. That is extremely unlikely to be correct. Let's assume that when we go back and check other records, we find that this person was really 290 pounds, but the initial 2 was inadvertently left off. Although this particular example was made up, it is an example of a type of mistake that we have encountered in real research data. With these formulas, the people who are doing the data collection have a good chance of catching the errors. Without them what can happen is that they send the spreadsheet to the statistician and then after a while they get back some questions. When we catch it early like this we save time, and avoid some frustration.

Figure 9: Minimum of 90 in weight.lbs column should raise a red flag.

F20

	A	B	C	D	E
1			num.other		
2			0		
3			min:		
4			90		
5			mean:		
6			147.5		
7			max:		
8			210		
9					
10	person.id	treatment	weight.lbs	height.ft	sex
11	1	a	110	5.3	f
12	2	b	210	6.1	m
13	3	c	165	5.9	m
14	4	a	130	5.6	f
15	5	b	90	6.5	m
16	6	c	180	6.1	m

Diagnostic Formulas for Checking Categorical Data

For columns with categorical data, we want to do similar kinds of checks. We want to know how many cells in the column have a value that is not one of the ones we think are allowable. That might seem so simple as to be not even worth checking, but remember, this spreadsheet will be read into R, so slight inconsistencies like extraneous spaces in a cell, or a capital letter when everything should be lowercase will need to be found and fixed before analysis.

The formula we want to use for the treatment column counts all the cells below B11 that have anything at all in them, and then subtracts the number of a's, b's, and c's. If the result is 0, at least we didn't find any errors. Of course we can't know there are no errors, only that we didn't find any of this particular type. The formula we use is here:

```
= COUNTA(B11:B100) - (SUM(EXACT(B11:B100,"a")+0) +
  SUM(EXACT(B11:B100,"b")+0) +
  SUM(EXACT(B11:B100,"c")+0))
```

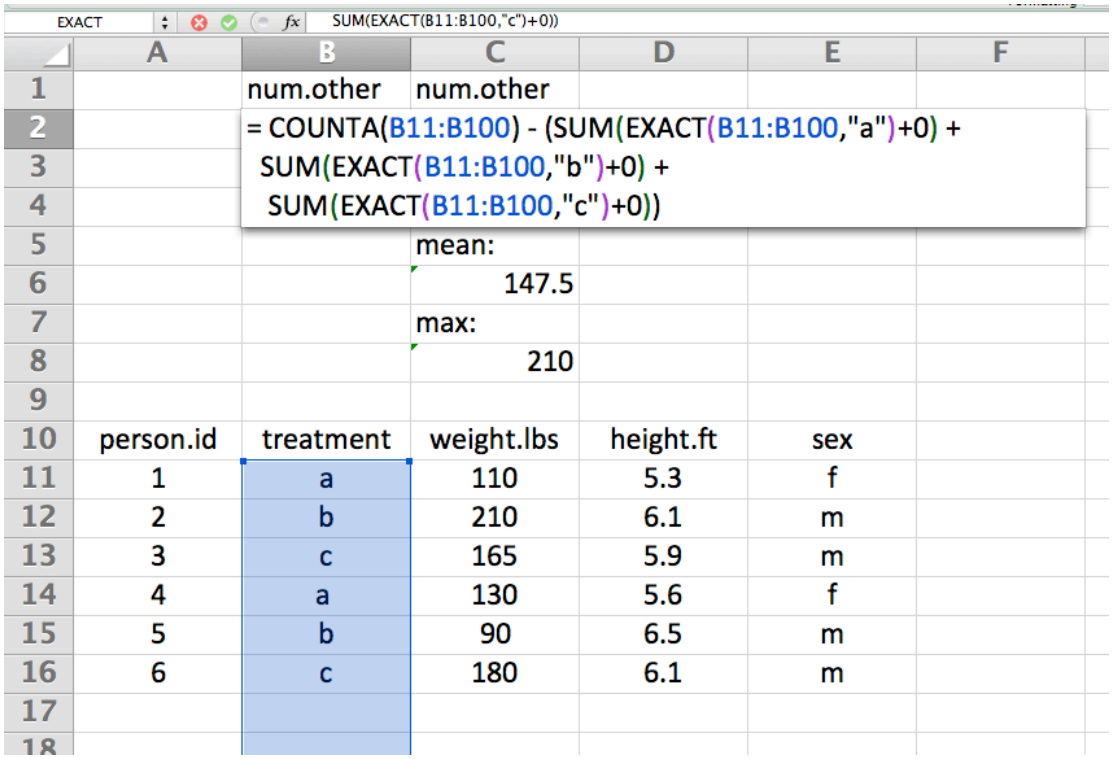
The function *counta* counts the cells that have anything in them at all. The expression SUM(EXACT(B11:B100,"b")+0) counts the number of times "b" occurs in the column. Of course there are similar expressions for "a" and "c". Using *exact* in this way allows us to find out whether anybody accidentally typed a space in a cell, as well as other possibilities.

This formula with exact is a special kind of Excel formula called an *array formula*. For an array formula, after typing it into the cell you cannot just hit enter. You must type control-shift-enter. If you don't remember this, and just type enter, the cell will display this: #VALUE! .

Remember: Array formulas require control-shift-enter

Figure 10 shows the formula for checking that the only entries in the column are "a", "b", or "c".

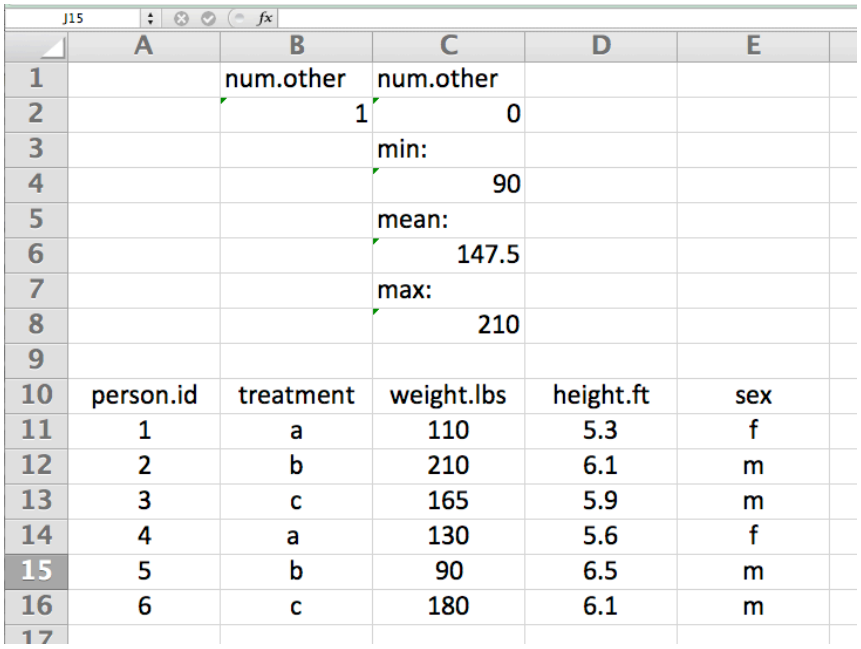
Figure 10: Formula for Checking Treatment Categorical Values



	A	B	C	D	E	F
1		num.other	num.other			
2						
3						
4						
5			mean:			
6			147.5			
7			max:			
8			210			
9						
10	person.id	treatment	weight.lbs	height.ft	sex	
11	1	a	110	5.3	f	
12	2	b	210	6.1	m	
13	3	c	165	5.9	m	
14	4	a	130	5.6	f	
15	5	b	90	6.5	m	
16	6	c	180	6.1	m	
17						
18						

Once we type control-shift-enter, the spreadsheet looks like Figure 11. Remember, we want the contents of cell B2 to be zero, but in this case it is 1. That shows us something is not quite right. Looking down the B column, nothing leaps out as

Figure 11: Formula in B2 catches a nonobvious error.



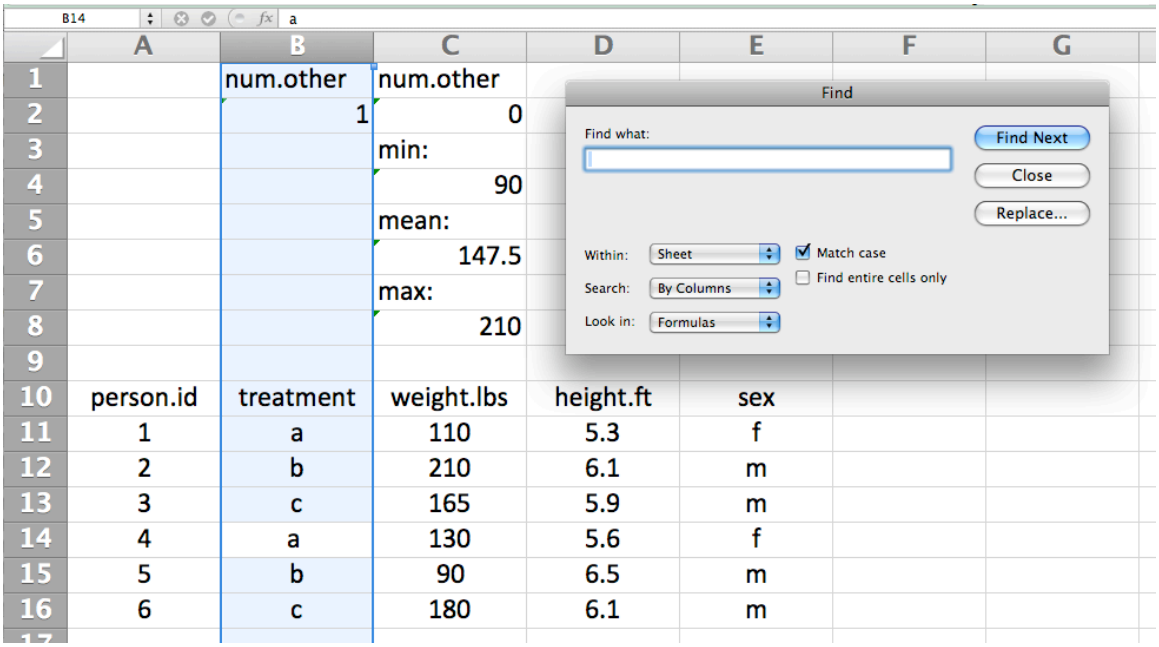
	A	B	C	D	E
1		num.other	num.other		
2		1	0		
3			min:		
4			90		
5			mean:		
6			147.5		
7			max:		
8			210		
9					
10	person.id	treatment	weight.lbs	height.ft	sex
11	1	a	110	5.3	f
12	2	b	210	6.1	m
13	3	c	165	5.9	m
14	4	a	130	5.6	f
15	5	b	90	6.5	m
16	6	c	180	6.1	m
17					

obviously wrong. If we think there might be an extraneous space, we can search for it explicitly. That is shown in Figure 12. In fact the cell B14 does have a space in it, and that shows up when we search. If we correct it, the diagnostic formula in B2 will show a zero instead of a 1 (not pictured).

We put formulas into the cells above *treatment* which display the number of a's, b's, and c's. These are just the parts of the formula shown in Figure 10, and described above. For illustration, Figure 13 shows the results when there is still an extraneous space in cell B14. The actual use of these cells for diagnosis and checking goes something like this. First we notice that something is amiss, because the number in B2 is not zero. Next we notice that there are 2 c's and 2 b's, but only 1 a, even though it looks like there should be 2 a's. That narrows down the problem quite a bit. At that point we can look at each cell, or search for spaces, or try something else, depending on what seems appropriate.

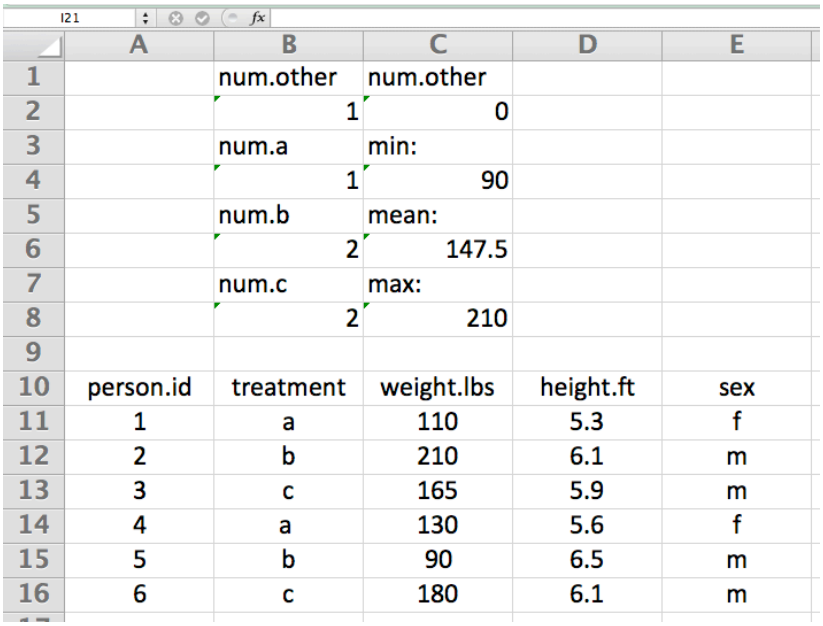
The diagnostics in the B column can also bring to light another kind of error. In cases where more than one person is collecting the data, people might not be

Figure 12: Searching for extraneous space in treatment column – find B14.



	A	B	C	D	E	F	G
1		num.other	num.other				
2		1	0				
3			min:				
4			90				
5			mean:				
6			147.5				
7			max:				
8			210				
9							
10	person.id	treatment	weight.lbs	height.ft	sex		
11	1	a	110	5.3	f		
12	2	b	210	6.1	m		
13	3	c	165	5.9	m		
14	4	a	130	5.6	f		
15	5	b	90	6.5	m		
16	6	c	180	6.1	m		

Figure 13: Categorical diagnostics for treatment column.



	A	B	C	D	E
1		num.other	num.other		
2		1	0		
3		num.a	min:		
4		1	90		
5		num.b	mean:		
6		2	147.5		
7		num.c	max:		
8		2	210		
9					
10	person.id	treatment	weight.lbs	height.ft	sex
11	1	a	110	5.3	f
12	2	b	210	6.1	m
13	3	c	165	5.9	m
14	4	a	130	5.6	f
15	5	b	90	6.5	m
16	6	c	180	6.1	m

consistent about how they enter missing data. For instance if somebody enters a dash (-) for a missing value, or types NA, that will show up in the count in cell B2. Similarly, if some of the data entry people type in A, B, and C for the treatments, and others use lowercase a, b, and c, these disparities will show up in cell B2.

Dates

Always record 4 digits for the year. Consider requiring the date to be either after today or before today, depending on what it represents. For instance, we have encountered birthdates of people who have been seen and measured, where the birthday was in the future. What should have been recorded as 19xx was interpreted or recorded as 20xx.

Feet

As a unit of length, feet are lousy. Fractions of a foot are usually measured in inches, and inches are, alas, not tenths of a foot. There is a tendency to put the measurement into a single column, with both feet and inches. Please don't do that. Each column should only correspond to one type of unit, feet or inches, never both. That means probably you should use meters or centimeters instead, but if you must use feet and inches, please create two adjacent columns, one for feet, then a second one for inches. If you don't do this at the start, it will still need to be done before any serious analysis can happen. Possibly you won't have to do it, but someone will.

In the examples above I used feet and tenths of a foot because I wanted a height (6 and a half feet) that I thought most Americans would realize was too tall to weigh 90 pounds. It was a made up example, and probably not the best.

Anticipating R's Foibles

If the data, once they have been collected, are to be read into R and analyzed, there are a few additional things that should be done. One is to pick column names that R can deal with well. That means they should start with a letter or underscore, and contain **only letters, numbers, underscore, and period**. In the examples above, sometimes I used a period to connect two parts of the column name. For instance, I have columns of people's weights and heights, and I want the units of measurement to be part of the column name. I used a dot to connect weight with the units (lbs), producing the name weight.lbs, and similarly for height.ft.

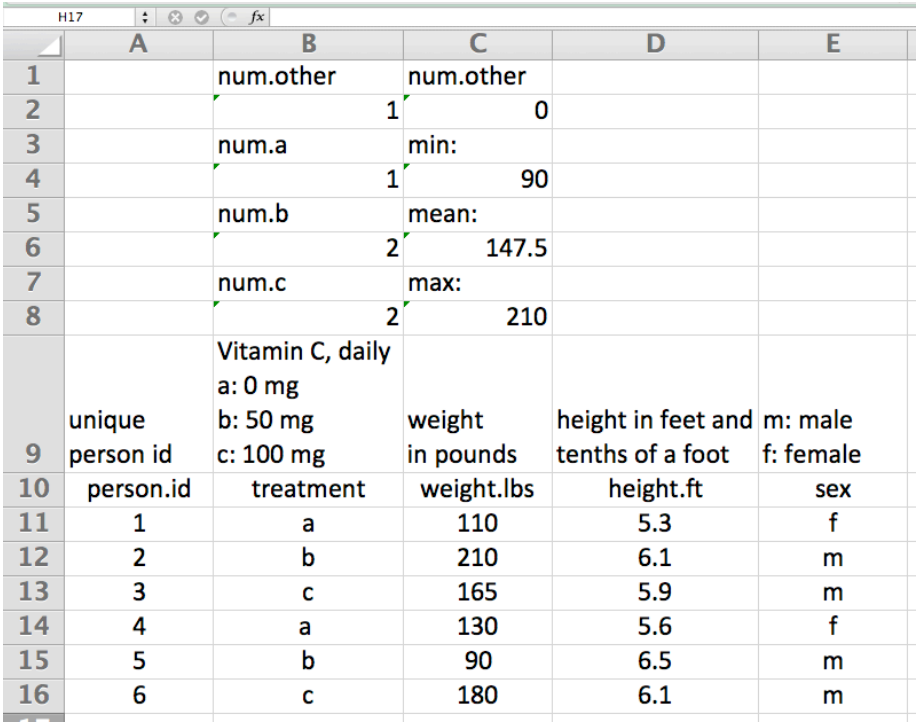
Please don't ever use spaces in names of columns or names of categories. When R reads in a table of data, it converts the spaces to dots (periods). The fewer discrepancies there are between the spreadsheet representation and the R representation, the easier it is to catch errors.

In general, since R makes a distinction between upper- and lowercase, and treats, for instance, *Mg* as being a completely different symbol from *mg*, I recommend only using lowercase letters in column names, and names of categories (e.g., treatment types, male/female, etc.). Lowercase letters are quicker and easier to type than capitals, and if you always use only lowercase, you don't run into problems of inconsistency.

Column Legend

It can be helpful to have a little bit of documentation in the cell directly above each column. It is especially helpful for the person doing the analysis if he did not collect the data and is not familiar with the experiment. A sample is shown in Figure 14. Useful bits of information include the units of measurement for numbers, and a brief description of what the categorical descriptors mean.

Figure 14: Sample documentation in cell immediately above each column



	A	B	C	D	E
1		num.other	num.other		
2		1	0		
3		num.a	min:		
4		1	90		
5		num.b	mean:		
6		2	147.5		
7		num.c	max:		
8		2	210		
9	unique person id	Vitamin C, daily a: 0 mg b: 50 mg c: 100 mg	weight in pounds	height in feet and tenths of a foot	m: male f: female
10	person.id	treatment	weight.lbs	height.ft	sex
11	1	a	110	5.3	f
12	2	b	210	6.1	m
13	3	c	165	5.9	m
14	4	a	130	5.6	f
15	5	b	90	6.5	m
16	6	c	180	6.1	m

Reading the Data into R

There is a function called `read.xls()` in the `gdata` package. It does a fine job of reading in a spreadsheet. There are two key things to know about it. One, you must have Perl installed on your machine, and you must tell the function where it (Perl) is. Two, you can specify a pattern telling where the table of data actually starts. This allows you to leave all the diagnostic information in the file and skip over it right to the beginning of the data table. The incantation below specifies the name of the file, the pattern, which should match the first cell of the table, *person.id* in this case, and then the location of Perl on my computer.

```
myexample <-  read.xls("DataValidation.xlsx", pattern = "person.id",
                       perl = "/usr/local/ActivePerl-5.14/bin/perl")
```

When I do this in R, I get the results below, which are just what I would want.

```
> myexample <-
      read.xls("~/jbear/Statlab/DataValidation/DataValidation.xlsx",
      pattern = "person.id"
      perl = "/usr/local/ActivePerl-5.14/bin/perl")
> myexample

     person.id treatment weight.lbs height.ft sex
```

```
1          1          a          110        5.3   f
2          2          b          210        6.1   m
3          3          c          165        5.9   m
4          4          a          130        5.6   f
5          5          b           90        6.5   m
6          6          c          180        6.1   m
```

Conclusion

The suggestions in this paper are meant to more tightly integrate the analysis phase with the data gathering phase, and reduce the length of time between when errors are found and when they are corrected. When errors are found as the data are being collected, they can be corrected on the spot, but when an entire spreadsheet containing errors is sent to some statistician somewhere else, who may not be familiar with the experimental setup, the detection and correction can take much longer.