PharmaSUG 2019 - Paper DV-184

Figure it out! Using significant figures from reported lab data to format TLF output

Elizabeth Thomas and Lauren Williams, Everest Clinical Research, Inc.

ABSTRACT

Laboratory data comes from a variety of sources with differing levels of precision and often needs to be manipulated and/or transformed before the results are reported in a table, listing, or figure. While reporting precision is well-established for many standard lab parameters, pre-specification of reporting precision for a novel lab value can be challenging. One option is to let the data, as reported, determine the precision. This paper gives a short refresher on the rules of significant figures (sig figs). You will learn how to calculate sig figs given a character-formatted result, how to keep track of sig figs for transformed data (sums, averages, ratios, and transformation by a constant), and how to format a result given an unrounded value and sig fig.

INTRODUCTION

When we do an analysis, we can often overlook the underlying precision with which data was captured. Data that comes from multiple sources, whether due to different machines, different recording practices, or any other reason, can have its precision quantified, to ensure that reported precision does not overstate results.

In this paper, you will learn how to extract precision from a number formatted as a character, how to work with that data so that precision is preserved through significant figures and decimal places, and see how that may influence reporting decisions.

WHAT IS A SIGNIFICANT FIGURE?

Significant figures are one way to capture the level of precision of a measurement. Put simply, it counts the number of meaningful digits in a value. Five basic rules describe how to count significant figures given a character-formatted value:

1. All non-zero digits are significant.
2. Zeros between non-zero digits are significant.
3. Leading zeros are never significant (e.g. 0.0026).
4. After the decimal point, all digits are significant: 2.00
5. In a number without a decimal point, trailing zeroes may or may not be significant. For the purposes of this paper, we will assume that all trailing zeros in values reported as whole numbers are significant – this simplifying assumption allows us to calculate sig figs from the character representation of any value.

CALCULATING SIGNIFICANT FIGURES FROM A CHARACTER STRING

With the simplifying assumption made for rule 5, we can calculate the number of sig figs from a character value by looking at the length of the string once we remove the decimal place and strip all leading zeros:

```
%macro getsig(varc=,sigf=);
   &sigf.=lengthn(strip(tranwrd(strip(compress(&varc.,"","kd"))||"*",
   "0"," ")))-1;
%mend getsig;
```

We remove non-digit characters with the `compress(&varc.,"","kd")` function, then strip leading and trailing blanks and add a placeholder character '*' representing the end of the string. Next we replace all zeroes in the string with spaces, and strip again to remove the leading spaces (`left` would also work).

The position of the end of string character is now indexed to the number of significant figures + 1, so we use `lengthn` and subtract one to get the number of significant figures.

COMPUTATION RULES FOR SIGNIFICANT FIGURES

1. When adding or subtracting numbers, the result should be rounded to the last decimal place that appears in both numbers:
$$\begin{array}{r} 0.00050 \\ +\ 2.003 \\ \hline 2.00350 = 2.004 \end{array}$$
2. When multiplying or dividing by a constant, the result should be rounded such that the final result has the same number of sigfigs as the original number.
3. When multiplying or dividing two numbers with some uncertainty, the result should be rounded so that the product/quotient has the same number of sigfigs as the number with the lesser number of significant figures.
4. When taking a log of a number (base 10), first write the number in scientific notation:
$$\log(0.000273) = \log(2.73 \times 10^{-4})$$
Note that the first part of a number in scientific notation (2.73), the mantissa, will determine the fractional part of the log, and that the second part (10^{-4}), the characteristics, will determine the integer part. The characteristics part has no uncertainty, so the result will have the same number of decimal places as the number of significant figures in the mantissa:
$$\begin{aligned} &= 0.436 - 4 \\ &= -4.436 \end{aligned}$$
If a natural log is needed, note that $\ln(x) = \ln(10) \cdot \log(x)$, and use the rules for multiplication by a constant.
With the assumption we've made in #5 above, the significant figures of the mantissa are equal to the number of significant figures in the number, so this simplifies to: the number of decimal places of precision of log(x) is equal to the number of significant figures of x.
5. A number taken to an exact power (e.g. squared or square root) will have the same number of significant figures as the original number.

These 6 rules will cover the most-used transformations as well as summary statistics for most lab values. Note that in general, figuring out the precision of the final result depends on either significant figures or decimal places. Both can be determined from the character variable, but neither can be reliably extracted from a numeric format, since trailing zeroes are important.

CAPTURING THE PRECISION INFORMATION OF A VARIABLE

We propose translating the character version of a variable into a triplet of numeric variables, which will allow significance-preserving computation:

- VARn = numeric version of the variable
- SIGF = number of significant figures
- SIGD = number of significant decimal places.

We can expand the macro first presented above to obtain these three values from a character variable:

```
%macro getsig(varc=,varn=,sigf=,sigd=);
   length &varn. &sigf. &sigd. tempindex 8;
   &varn.=input(&varc.,best.);
   &sigf.=lengthn(strip(tranwrd(strip(compress(&varc.,"","kd"))||"*",
   "0"," ")))-1;
   tempindex=index(&varc.,'.');
```

```
   if tempindex>0 then &sigd.=lengthn(strip(&varc.))-tempindex;
   else &sigd.=0;
   drop tempindex;
%mend getsig;
```

TRANSLATING BETWEEN REPRESENTATIONS OF PRECISION

So far, we've seen that there are three ways to show the precision of a variable:

1. Character representation, including significant trailing zeroes
2. Numeric representation and significant figures
3. Numeric representation and significant decimal places

In order to use all the operations, we need to be able to translate between these representations, not just from a character variable to representations 2 or 3.

Three helper macros are defined in this paper; while this is not a complete set of transformations, they can be used in combination to translate in any direction.

Significant decimal places from numeric representation and significant figures

To get significant decimal places given a number and the significant figures, first we need to figure out where you would start counting sigfigs, relative to the decimal place. To do this we take the floor of log base 10 of the value (so anything between 0.1 and 1 will have templog=-1). We want for the number of significant figures to equal the number of significant decimals when templog=-1, and for significant decimals to increase relative to significant figures as templog gets more negative. Mathematically, that translates to $Significant\ dp = signficant\ figures - (floor(\log_{10}(value)) + 1)$. The maximum with zero protects against negative significant decimal places, in case a previous computation led to a whole number value with non-significant trailing zeroes.

```
%macro getsigd(varn=,sigf=,sigd=);
   length &sigd. templog 8;
   templog=floor(log10(&varn.));
   &sigd.=max(0,&sigf.-(templog+1));
   drop templog;
%mend getsigd;
```

Significant figures from numeric representation and significant decimal places

Similarly, we can obtain the significant figures from the number and significant decimal places, assuming all trailing zeroes are significant.

```
%macro getsigf(varn=,sigf=,sigd=);
   length &sigf. templog 8;
   templog=floor(log10(&varn.));
   &sigf.=&sigd.+templog+1;
   drop templog;
%mend getsigf;
```

Character representation from numeric representation and significant decimal places

Finally, formatting the character variable is quite straightforward if the number of significant decimal places is known:

```
%macro getsigc(varc=,varn=,sigd=);
   length &varc. $ 20 tempfmt $8;
   tempfmt="20."||strip(put(&sigd.,best.));
   &varc.=putn(&varn.,tempfmt);
   drop tempfmt;
%mend getsigc;
```

EXAMPLES

The two examples provided in this paper cover common scenarios: you need to take a ratio between two parameters, or you need to collapse records using an average. The first is primarily multiplicative in nature, and relies heavily on significant figures. The second combines addition with multiplication by a known constant, so it represents a slightly more complex algorithm.

All examples are constructed from simulated data. No actual data was used in this paper.

CALCULATION OF A LIVER FUNCTION RATIO FOR LOCAL LABS

Liver function, as measured by Alanine Transaminase (ALT), Aspartate Transaminase (AST), or Bilurubin, is often compared to the upper limit of normal (ULN) in order to standardize and check for severity of abnormalities.

ULNs can vary between laboratories, which can depend on locality (Neuschwander-Tetri, B., 2008). Local labs may also differ in precision reported both for values and for normal limits.

Simulating 5 sites, with 5 patients at each site, with ULNs that differ both in value and in reported precision, we show the results of applying the macros to get precision variables for both the reported original value (lborres: aval, sigf_aval, sigd_aval), and the upper limit of normal (lbornrhi: uln, sigf_uln, sigd_uln) in Output 1:

Obs	usubjid	paramcd	lborres	lbornrhi	aval	sigf_ aval	sigd_ aval	uln	sigf_uln	sigd_uln
1	001-001	ALT	18.4	54.8	18.40	3	1	54.80	3	1
2	001-002	ALT	58.9	54.8	58.90	3	1	54.80	3	1
3	001-003	ALT	118.1	54.8	118.10	4	1	54.80	3	1
4	001-004	ALT	105.0	54.8	105.00	4	1	54.80	3	1
5	001-005	ALT	43.1	54.8	43.10	3	1	54.80	3	1
6	002-001	ALT	21.80	59.00	21.80	4	2	59.00	4	2
7	002-002	ALT	69.42	59.00	69.42	4	2	59.00	4	2
8	002-003	ALT	280.16	59.00	280.16	5	2	59.00	4	2
9	002-004	ALT	60.17	59.00	60.17	4	2	59.00	4	2
10	002-005	ALT	9.78	59.00	9.78	3	2	59.00	4	2
11	003-001	ALT	16.62	38.26	16.62	4	2	38.26	4	2
12	003-002	ALT	81.97	38.26	81.97	4	2	38.26	4	2
13	003-003	ALT	130.56	38.26	130.56	5	2	38.26	4	2
14	003-004	ALT	35.19	38.26	35.19	4	2	38.26	4	2
15	003-005	ALT	24.90	38.26	24.90	4	2	38.26	4	2
16	004-001	ALT	92.67	61.16	92.67	4	2	61.16	4	2
17	004-002	ALT	45.55	61.16	45.55	4	2	61.16	4	2
18	004-003	ALT	252.21	61.16	252.21	5	2	61.16	4	2
19	004-004	ALT	50.33	61.16	50.33	4	2	61.16	4	2
20	004-005	ALT	25.36	61.16	25.36	4	2	61.16	4	2
21	005-001	ALT	33.03	65.71	33.03	4	2	65.71	4	2
22	005-002	ALT	63.84	65.71	63.84	4	2	65.71	4	2
23	005-003	ALT	299.33	65.71	299.33	5	2	65.71	4	2
24	005-004	ALT	58.02	65.71	58.02	4	2	65.71	4	2
25	005-005	ALT	63.14	65.71	63.14	4	2	65.71	4	2

Output 1. Output from a PRINT Statement

From there we calculate the ratio simply:

```
ratio_uln=aval/uln;
sigf ratio=min(sigf aval,sigf uln);
%getsigd(varn=ratio_uln,sigf=sigf_ratio,sigd=sigd_ratio);
%getsigc(varc=ratio_uln_c1,varn=ratio_uln,sigd=sigd_ratio);
```

We can also construct an alternate character presentation of the ratio (ratio_uln_c2) by taking the minimum sigd_ratio, and using that to format every value. Both styles are shown in Output 2.

Obs	usubjid	paramcd	lborres	lbornrhi	aval	sigf_ aval	sigd_ aval	uln	sigf_uln	sigd_uln	ratio_ uln	sigf_ ratio	sigd_ ratio	ratio_ uln_c1	ratio_ uln_c2
1	001-001	ALT	18.4	54.8	18.40	3	1	54.80	3	1	0.33577	3	3	0.336	0.34
2	001-002	ALT	58.9	54.8	58.90	3	1	54.80	3	1	1.07482	3	2	1.07	1.07
3	001-003	ALT	118.1	54.8	118.10	4	1	54.80	3	1	2.15511	3	2	2.16	2.16
4	001-004	ALT	105.0	54.8	105.00	4	1	54.80	3	1	1.91606	3	2	1.92	1.92
5	001-005	ALT	43.1	54.8	43.10	3	1	54.80	3	1	0.78650	3	3	0.786	0.79
6	002-001	ALT	21.80	59.00	21.80	4	2	59.00	4	2	0.36949	4	4	0.3695	0.37
7	002-002	ALT	69.42	59.00	69.42	4	2	59.00	4	2	1.17661	4	3	1.177	1.18
8	002-003	ALT	280.16	59.00	280.16	5	2	59.00	4	2	4.74847	4	3	4.748	4.75
9	002-004	ALT	60.17	59.00	60.17	4	2	59.00	4	2	1.01983	4	3	1.020	1.02
10	002-005	ALT	9.78	59.00	9.78	3	2	59.00	4	2	0.16576	3	3	0.166	0.17
11	003-001	ALT	16.62	38.26	16.62	4	2	38.26	4	2	0.43440	4	4	0.4344	0.43
12	003-002	ALT	81.97	38.26	81.97	4	2	38.26	4	2	2.14245	4	3	2.142	2.14
13	003-003	ALT	130.56	38.26	130.56	5	2	38.26	4	2	3.41244	4	3	3.412	3.41
14	003-004	ALT	35.19	38.26	35.19	4	2	38.26	4	2	0.91976	4	4	0.9198	0.92
15	003-005	ALT	24.90	38.26	24.90	4	2	38.26	4	2	0.65081	4	4	0.6508	0.65
16	004-001	ALT	92.67	61.16	92.67	4	2	61.16	4	2	1.51521	4	3	1.515	1.52
17	004-002	ALT	45.55	61.16	45.55	4	2	61.16	4	2	0.74477	4	4	0.7448	0.74
18	004-003	ALT	252.21	61.16	252.21	5	2	61.16	4	2	4.12377	4	3	4.124	4.12
19	004-004	ALT	50.33	61.16	50.33	4	2	61.16	4	2	0.82292	4	4	0.8229	0.82
20	004-005	ALT	25.36	61.16	25.36	4	2	61.16	4	2	0.41465	4	4	0.4147	0.41
21	005-001	ALT	33.03	65.71	33.03	4	2	65.71	4	2	0.50266	4	4	0.5027	0.50
22	005-002	ALT	63.84	65.71	63.84	4	2	65.71	4	2	0.97154	4	4	0.9715	0.97
23	005-003	ALT	299.33	65.71	299.33	5	2	65.71	4	2	4.55532	4	3	4.555	4.56
24	005-004	ALT	58.02	65.71	58.02	4	2	65.71	4	2	0.88297	4	4	0.8830	0.88
25	005-005	ALT	63.14	65.71	63.14	4	2	65.71	4	2	0.96089	4	4	0.9609	0.96

Output 2. Output from a PRINT Statement

While ratio_uln_c1 gives as much precision on each variable as possible, ratio_uln_c2 maintains a minimum level of precision and may be more suitable for standardizing output.

VISIT AVERAGES FOR VARIABLY REPEATED TESTS

Some lab tests or vital signs may be repeated multiple times, but the number of replicates may depend on the volume of blood drawn or the individual site's testing schedule, and the individual measurement precision may vary. Often the analysis requires that these multiple records are summarized together at a single visit, taking the average of all measurements of a given parameter from that subject on that visit.

We simulated three sites with possibly-different measurement precision, three subjects at each site, with two visits each and randomly selected multiple replications of Systolic Blood Pressure on each subject at each visit (shown in Output 3 after the initial processing of character value into the numeric triplet).

Obs	usubjid	paramcd	vsorres	visitnum	aval	sigf_ aval	sigd_ aval
1	001-001	SYSBP	146.1	1	146.1	4	1
2	001-001	SYSBP	133.0	1	133.0	4	1
3	001-001	SYSBP	173.6	1	173.6	4	1
4	001-001	SYSBP	114.6	1	114.6	4	1
5	001-001	SYSBP	106.6	2	106.6	4	1
6	001-001	SYSBP	139.2	2	139.2	4	1
7	001-001	SYSBP	132.1	2	132.1	4	1
8	001-001	SYSBP	147.2	2	147.2	4	1
9	001-001	SYSBP	106.2	2	106.2	4	1
10	002-001	SYSBP	140	1	140.0	3	0
11	002-001	SYSBP	107	1	107.0	3	0
12	002-001	SYSBP	160	2	160.0	3	0
13	003-001	SYSBP	80	1	80.0	2	0
14	003-001	SYSBP	74	1	74.0	2	0
15	003-001	SYSBP	136	1	136.0	3	0
16	003-001	SYSBP	136	1	136.0	3	0
17	003-001	SYSBP	159	1	159.0	3	0
18	003-001	SYSBP	152	2	152.0	3	0
19	003-001	SYSBP	112	2	112.0	3	0

Output 3. Output from a PRINT Statement

Following the initial processing, we use proc means to get the sum of values (aval_s), mean of values (aval_m), and number of significant decimal places in the sum (sigd_s, calculated as the minimum sigd_aval). From here, we can find the significant figures of the sum (sigf_s), which is equal to the significant figures of the mean – because the number of records for each subject/visit combination is known exactly, and division by a constant preserves significant figures according to Rule 2. As in the earlier example, we create one version of the summary records that gives individual record-level precision (avalc_m1), and another that takes the minimum number of decimal places and formats every record to that level (avalc_m2), seen in Output 4 below.

```
data sbp_m;
  set sbp_m;
  %getsigf(varn=aval_s,sigf=sigf_sum,sigd=sigd_s);
  length sigf_m 8;
  sigf_m=sigf_sum;
  %getsigd(varn=aval_m,sigf=sigf_m,sigd=sigd_m);
  %getsigc(varc=avalc_m1,varn=aval_m,sigd=sigd_m);
run;
```

Obs	usubjid	paramcd	visitnum	nrec	aval_m	sigd_m	sigf_m	aval_s	sigd_s	avalc_m1	avalc_m2
1	001-001	SYSBP	1	4	141.825	1	4	567.3	1	141.8	142
2	001-001	SYSBP	2	5	126.260	1	4	631.3	1	126.3	126
3	002-001	SYSBP	1	2	123.500	0	3	247.0	0	124	124
4	002-001	SYSBP	2	1	160.000	0	3	160.0	0	160	160
5	003-001	SYSBP	1	5	117.000	0	3	585.0	0	117	117
6	003-001	SYSBP	2	2	132.000	0	3	264.0	0	132	132

Output 4. Output from a PRINT Statement

CONCLUSION

Working with laboratory data or other data that is measured with some varying precision can be tricky. Learning how to represent numeric precision allows you to pass that precision to be through in computations.

Precision alone is not the only factor that you should weigh when considering how to structure your outputs. The precision as calculated is a recommended maximum amount of precision that should be represented in final outputs. Consider also readability, comparability, and industry norms when you decide how much precision should be reported in tables and listings.

REFERENCES

Neuschwander-Tetri, B. et al. 2008. “Influence of Local Reference Populations on Upper Limits of Normal for Serum Alanine Aminotransferase Levels.” Arch Intern Med., 168(6): 663-666

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Liz Thomas
Everest Clinical Research
liz.thomas@ecrscorp.com