PhUSE EU Connect 2018

Paper AS01

Overlooked Tiny Numeric Representation Errors in SAS may Lead to Substantially Misleading p-Values

Ying Zhou, Lali Sandalic, Holger Bohnemeier, David Haryanto, Jens Klement,
PRA Health Sciences, Germany

ABSTRACT

0.1 + 0.2 ≠ 0.3 and related numeric representation errors in SAS are well known among experienced SAS users. However, these errors are usually regarded as tiny and inconsequential. For SAS beginners, little attention may be given to numeric representation errors since neither error nor warning messages appear.

In our paper, concrete examples are provided to illustrate that overlooked numeric representation errors may lead to misleading p-values by default which eventually may reverse the statistical conclusion. The examples include rank-based statistical tests (such as Wilcoxon Rank-Sum Test) and the statistical tests in comparison to a reference value (such as t-Test and Wilcoxon Signed-Rank Test). Statistical backgrounds and numeric format conversions such as Format HEX16. are mentioned to provide an insight into the issue.

In conclusion, the overlooked tiny numeric representation errors may affect the statistical result considerably. Applicable solutions, their advantages and limitations are also discussed.

INTRODUCTION

Nowadays, the use of computational systems for handling numbers is commonplace. However, the results are usually taken as granted, and users tend to neglect system-inherent hardware and software limitations from time to time.

Numeric representation errors are a common problem in computational systems. SAS, or any other computational system, is unable to store infinite numbers, while the real number system used commonly in our daily work and life is infinite. The infinite real number system has to be represented by a finite set of numbers, which leads to numeric representation errors (SAS Technical Paper, 2014; Tambascia, 2011). For example, in the Windows Calculator (e.g., Windows 7), the square root of 4 minus 2 (√4 - 2) does not equal to 0 but leaves a remnant within the range of 10^{-38} in scientific mode.

Any numeric representation approach has advantages and limitations, without a perfect solution to avoid numeric representation errors. The floating-point representation has been used in the SAS system to store numeric values, because it provides a best compromise between precision and magnitude (SAS Technical Paper, 2014). But it may be a source of numeric representation errors. For experienced users, it is not a surprise that 0.2 + 0.1 does not equal to 0.3 in SAS (Lehrkamp, 2016), or that 4.8 - 4.6 does not equal to 0.2 (Tambascia, 2011). In SAS, 0.5 - 0.25 - 0.125 - 0.125 equals to 0 whereas 0.3 - 0.2 - 0.1 does not, because for the former one all numbers are integer powers of 2 and can be represented with a finite binary representation ($0.5 = 1 * 2^{-1}$, $0.25 = 1 * 2^{-2}$, $0.125 = 1 * 2^{-3}$) whereas it is not the case for the latter one.

Very often SAS users may overlook numeric representation errors because they regard them as tiny and inconsequential when calculating various statistics (Stutzman, 2014), or because they do not realize these errors since neither “error” nor “warning” messages appear in the log.

Until now, the influence of numeric representation errors in SAS applications in available literature has been limited to comparing the variables and sub-setting the datasets, concluding that these tiny errors have little impact on the results in most cases (Stutzman, 2014; SAS Technical Paper, 2014). The potential risks of erroneous statistical results due to numeric representation errors have caught little attention among SAS users.

In this publication, we will show that numeric representation errors may also produce misleading p-values that eventually may reverse the statistical conclusion. To the best of our knowledge, this is the first publication to present how overlooked numeric representation errors in SAS may affect the statistical results.

The following two examples will illustrate how overlooked tiny numeric representation errors can lead to substantially misleading p-values in Wilcoxon Rank-Sum Test (Example 1) and in Wilcoxon Signed-Rank Test and t-Test (Example 2). The SAS syntaxes can be found in the appendix. All these example analyses were conducted using SAS Version 9.4 on the platform Windows Server 2012 Data Center (64-bit operating system). The size of numeric representation errors may vary depending on the computer system and software.

THE PROBLEMS

EXAMPLE 1 (RANK-BASED NON-PARAMETRIC STATISTICAL TESTS: WILCOXON RANK-SUM TEST)

The first example illustrates a misleading p-value due to numeric representation errors in the Wilcoxon Rank-Sum Test, a rank-based non-parametric statistical test. Other rank-based non-parametric statistical tests include Wilcoxon Signed-Rank Test, Signed Test, Rank-Based ANOVA, as well as advanced methods of inferential statistics such as robust regression methods by PROC ROBUSTREG in SAS (Liu, 2015). If the assumption of normally distributed data cannot be met, the rank-based non-parametric statistical tests could be more appropriate and robust. In these statistical tests, the ranks play a decisive role on the statistical results instead of means and standard deviations. The ranks reflect the order of the absolute values, but the extent of the differences among values may not be taken into account. The first rank is assigned to the smallest value, and the second rank to the second smallest value, no matter if the differences among these values are as large as 10^7 or as tiny as 10^{-7}.

In the first example, we compare Product A and Product B regarding the Change from Baseline (BL) of a parameter “Para” in a study. First, the Changes from BL (timepoint1 – timepoint0) are calculated, then the units are converted from grams into milligrams, and finally the Wilcoxon Rank-Sum Test is applied. Since there are ten subjects per group, a normal approximation to the Wilcoxon Rank-Sum Test can be used (Walker, 2010). All p-values presented in this example will be calculated using the normal approximation.

Table 1 Different p-values of Wilcoxon Rank-Sum Test for Product Comparison Caused by Not-Handled versus Handled Numeric Representation Errors

Subject Number	Product	Para[g] (t0)	Para[g] (t1)	Default Change from BL (no additional round function): Para[mg] (t1-t0)*1000 in Format 19.15	Default Change from BL: Rank	Expected Change from BL (with additional round function): Para[mg] (t1-t0)*1000 in Format 19.15	Expected Change from BL: Rank
1	A	0.1226	0.12	-2.600000000000000	5	-2.600000000000000	5
2	A	0.1226	0.1194	-3.199999999999990	**4**	-3.200000000000000	**3.5**
3	A	0.1224	0.1207	-1.699999999999990	9	-1.700000000000000	9
4	A	0.1241	0.1233	-0.799999999999990	**15.5**	-0.800000000000000	**15**
5	A	0.1255	0.1201	-5.400000000000000	1	-5.400000000000000	1
6	A	0.133	0.1312	-1.799999999999990	7.5	-1.800000000000000	7.5
7	A	0.1219	0.1203	-1.599999999999990	**11**	-1.600000000000000	**10.5**
8	A	0.121	0.1211	0.100000000000000	**19**	0.100000000000000	**18.5**
9	A	0.1227	0.1203	-2.400000000000000	6	-2.400000000000000	6
10	A	0.1238	0.1204	-3.400000000000000	2	-3.400000000000000	2
11	B	0.1226	0.121	-1.600000000000000	**10**	-1.600000000000000	**10.5**
12	B	0.1194	0.1195	0.099999999999980	**18**	0.100000000000000	**18.5**
13	B	0.1203	0.1188	-1.500000000000000	12	-1.500000000000000	12
14	B	0.1326	0.1337	1.100000000000000	20	1.100000000000000	20
15	B	0.1207	0.1189	-1.799999999999990	7.5	-1.800000000000000	7.5
16	B	0.1227	0.1195	-3.200000000000000	**3**	-3.200000000000000	**3.5**
17	B	0.1222	0.121	-1.200000000000000	13	-1.200000000000000	13
18	B	0.1203	0.1195	-0.800000000000000	**14**	-0.800000000000000	**15**
19	B	0.1231	0.1224	-0.700000000000000	17	-0.700000000000000	17
20	B	0.1211	0.1203	-0.799999999999990	**15.5**	-0.800000000000000	**15**
					Ranks Sum for A = **80** P=**0.0638**		Ranks Sum for A =**78** P=**0.0445**

Numbers in bold red show different ranks between both calculations

In Table 1, the two columns under “Default Change from BL” are the unit-converted Changes from BL and their ranks obtained from SAS directly, without applying the round function additionally before the NPAR1WAY procedure. In comparison, the columns under “Expected Change from BL” represent the expected results, which are generated by an additional rounding step of the unit-converted Change from BL before the Wilcoxon Rank-Sum Test. Moreover, if the unit-converted Change from BL is calculated manually, the same results are achieved as demonstrated in the column “Expected Change from BL”.

By comparing the default versus expected Change from BL and their corresponding ranks in Table 1, some ranks under the column “Default Change from BL” are observed to be different from the correct ranks due to numeric representation errors. For the second subject, for example, the default unit-converted Change from BL in SAS is -3.199999999999990 and erroneously rank 4, while the correct calculated result of Change from BL for that observation is -3.200000000000000 and rank 3.5, respectively. This results in PROC NPAR1WAY considering the values of Subjects 2 and 16 being different while they are in fact tied values (SAS Support, Usage Note 23117).

Numeric representation errors result in a change of the ranks, thus affecting the statistical results. If the default results of “Change from BL” and ranks from SAS are used directly, then a misleading two-sided p-value of 0.0638 is produced. If the round function is applied in addition before the statistical test, the two-sided p-value equals to 0.0445 which indicates the correct statistically significant difference between Product A and Product B using a common significance level $\alpha = 0.05$.

Please note that in order to illustrate the difference between neglecting and managing numeric representation errors, we are showing unit-converted Change from BL with 15 decimal places in SAS. If the default numeric format has been used, or if the common format for decimal places has been applied explicitly, such as 1, 2 or 3 decimal places, SAS shows the apparently correct unit-converted Change from BL. However, SAS applies erroneous unit-converted Change from BL to the ranking with numeric representation errors by default, and thus the ranks are wrong and the p-value is substantially misleading. In these cases, neither error messages appear in the log nor unexpected intermediate results (namely the undesirable unit-converted Change from BL) appears in the SAS datasets. Everything seems to be fine. But unfortunately the overlooked tiny numeric representation errors lead to a substantially misleading p-value without any error/warning messages or hints.

In order to provide an insight into the issue in Example 1, Table 2 displays the calculated results of the parameter “Para” in HEX16. format, which converts the floating-point numbers from the decimal numeral system to their hexadecimal formats, and thus reflects the actually stored internal numbers on the computational platform.

Table 2 Insight into the Numeric Representation Errors by HEX16. Format in the First Example (Observations 1 to 3 and 16)

SN	Para[g] (t0)	Para[g] (t1)	Change from BL without Unit-conversion: Para[g] (t1 - t0)			Unit-converted Change from BL: Para[mg] (t1-t0)*1000		
	DEC	DEC	DEC	HEX (without rounding)	HEX (with rounding)	DEC	HEX (without rounding)	HEX (with rounding)
1	0.1226	0.12	-0.0026	BF654C985F06F6**A0**	BF654C985F06F6**94**	-2.6	C004CCCCCCCCCC**D8**	C004CCCCCCCCCC**CD**
2	0.1226	0.1194	-0.0032	BF6A36E2EB1C43**20**	BF6A36E2EB1C43**2D**	-3.2	C009999999999999**8D**...	C0099999999999**9A**
3	0.1224	0.1207	-0.0017	BF5BDA5119CE07**40**	BF5BDA5119CE07**5F**	-1.7	BFFB3333333333**14**	BFFB3333333333**33**
…	…	…	…	…	…	…	…	…
16	0.1227	0.1195	-0.0032	BF6A36E2EB1C43**40**	BF6A36E2EB1C43**2D**	-3.2	C00999999999999**AC**	C0099999999999**9A**

Numbers in bold red show differences in HEX16. format between both calculations
Abbreviations: SN: Subject Number; BL: Baseline; DEC: Decimal representation in default format; HEX: Hexadecimal representation (HEX16.)

Taking the Subjects 2 and 16 as an example, if no round function has been additionally applied, the Changes from BL without Unit-conversion are the same for both the subjects in decimal default format (both are -0.0032). They remain the same even with using format 19.15 (both are -0.003200000000000). However, the HEX16. format reveals that these two subjects have already different values for the Change from BL without Unit-conversion (Subject 2: BF6A36E2EB1C43**20** and Subject 16: BF6A36E2EB1C43**40**). The further unit conversion through multiplying by 1000 has even led to an increase of the numeric representation error. Thus, the difference between Subject 2 (-3.199999999999990) and Subject 16 (-3.200000000000000) can be observed in format 19.15 for the Unit-converted Change from BL. If the round function is used in addition, the HEX16. format shows that both, Subject 2 and Subject 16, have the same Change from BL without Unit-conversion (BF6A36E2EB1C432D) and the same Unit-converted Change from BL (C00999999999999A).

This example shows how the unit conversion through multiplying by 1000 has increased the numeric representation errors. It has to be supposed that the size of the numeric representation errors could accumulate and increase considerably if some intermediate calculations such as multiplications are done many times (Imelda, 2008; Henrici, 1964).

EXAMPLE 2 (STATISTICAL TESTS IN COMPARISON TO A REFERENCE VALUE: WILCOXON SIGNED-RANK TEST AND T-TEST)

The second example illustrates the misleading p-values due to numeric representation errors in the statistical tests in comparison to a reference value. The statistical tests in comparison to a reference value refer to the parametric or non-parametric statistical tests which are used to infer whether a population statistic (such as mean or median) differs from a reference value. The common one-sample t-Test, and the tests for pairwise comparison such as paired t-Test and Wilcoxon Signed-Rank Test (the paired difference versus a reference value such as zero) also belong to such statistical tests. In these tests, the individual values (or the individual differences) are compared to the reference value. The numeric representation errors may lead to incorrect statistical results if each or most individual values in the comparison have a misleading result only due to numeric representation errors.

For a better understanding of the dummy data in the second example, we provide a concrete context. A new product of antiperspirant (Product B) is supposed to be compared to a control antiperspirant or placebo product (Product A) in terms of efficacy to control sweat. We have ten healthy volunteers and the two products are randomly applied to the left and right underarm of each subject according to a randomization plan. Before the product application, two pads are weighed and documented as Pad Tare Weight (m0) for each subject. After applying the products, the two pads are put under the armpit, one for the left side and the other for the right side. Then subjects stay in a sauna room for 15 minutes. After the sauna time, the two pads are weighed again, and documented as Pad Gross Weight (m1).

The Sweat Weight for each product is a parameter for the efficacy to control sweat, which will be calculated by subtracting Pad Tare Weight (m0) from Pad Gross Weight (m1). Since each subject has applied both Product A and Product B, the pairwise statistical analysis can be conducted. The null hypothesis is that there is no difference between Product A and Product B for the Sweat Weight, and the alternative hypothesis is that there is such a difference. To test this hypothesis, the mean paired difference for the Sweat Weight between Product B side and Product A side will be compared to the reference value zero.

Table 3 Misleading p-Values of Wilcoxon Signed-Rank Test and t-Test for Product Comparison Caused by Not-Handled Numeric Representation Errors

Subject Number	Product	Pad Tare Weight[mg] (m0)	Pad Gross Weight[mg] (m1)	Sweat Weight: Pad Net Weight [mg] (m1-m0) in Format 19.15	Paired Difference for Sweat Weight [mg] (Product B – Product A) in Format 19.15
1	A	119.4	130.2	10.799999999999900	0.000000000000014
	B	119.2	130.0	10.799999999999900	
2	A	123.3	137.1	13.800000000000000	0.000000000000014
	B	119.5	133.3	13.800000000000000	
3	A	121.0	132.6	11.599999999999900	0.000000000000014
	B	120.3	131.9	11.600000000000000	
4	A	121.2	132.2	10.999999999999900	0.000000000000014
	B	124.4	135.4	11.000000000000000	
5	A	123.3	136.1	12.800000000000000	0.000000000000014
	B	119.5	132.3	12.800000000000000	
6	A	122.0	134.6	12.599999999999900	0.000000000000014
	B	121.3	133.9	12.600000000000000	
7	A	118.2	131.4	13.200000000000000	0.000000000000014
	B	118.6	131.8	13.200000000000000	
8	A	127.9	138.7	10.799999999999900	0.000000000000028
	B	126.5	137.3	10.800000000000000	
9	A	135.8	147.0	11.199999999999900	0.000000000000028
	B	167.2	178.4	11.200000000000000	
10	A	126.3	140.2	13.899999999999900	0.000000000000014
	B	119.6	133.5	13.900000000000000	

Wilcoxon Signed-Rank Test

$S = (R^{(+)} – R^{(-)})/2= (55-0)/2=27.5$

P = 0.0020

t-Test

$t= \frac{\overline{y} - \mu_0}{s/\sqrt{n}} = \frac{1.71 * 10^{-14} - 0}{5.99 * 10^{-15}/\sqrt{10}} =9.0$

P < 0.0001

As shown in Table 3, the dummy data have been generated intentionally for the equal Sweat Weight (Pad Net Weight) between Product A and Product B per subject. Thus, there should be no difference between Product A and Product B, not to mention a statistically significant difference. However, only due to numeric representation errors, the paired difference between products regarding the Sweat Weight seems to be larger than zero as shown in the last column.

For the purpose of our illustration, we conduct the Wilcoxon Singed-Rank Test and paired t-Test for these dummy data in this context without validating the assumptions of statistical tests, in order to show how the incorrect p-value might be produced. As mentioned in the next section, the validation of the assumptions for the statistical test and the size of treatment effects should be checked, so that such incorrect p-values can be recognized and avoided.

As to the Wilcoxon Signed-Rank Test in this example, all the differences between Product B and Product A are positive because of numeric representation errors. Thus, the sum of the ranks associated with the positive paired differences $R^{(+)}$ is (1 + 10) * 10 / 2 = 55, while the sum of the ranks associated with the negative paired differences $R^{(-)}$ is 0. The signed-rank statistic is 27.5 and the corresponding p-value is 0.0020. Therefore, the Wilcoxon Signed-Rank Test shows a highly significant difference between Product A and Product B regarding the Sweat Weight (Pad Net Weight), although there are no differences other than those due to numeric representation errors that are generated in SAS at its default settings.

As to the paired t-Test in this example, all positive paired differences in the last column also contribute to the high statistical significance ($P < 0.0001$). Although no difference exists between products, the numeric representation errors lead to a mean paired difference of $1.71 * 10^{-14}$, and a standard deviation of $5.99 * 10^{-15}$. In this t-Test, the paired difference is compared to the reference value 0. The result is that the t-Test statistic equals to 9.0 and the p-value is smaller than 0.0001.

THE SOLUTIONS AND DISCUSSION

The above two examples illustrate that substantially misleading p-values might be generated due to overlooked tiny numeric representation errors. How can such misleading statistical results be avoided? We have several suggestions.

First, it is highly recommended to take the tiny numeric representation errors into account if rank-based non-parametric tests or the statistical tests versus a reference value are performed. It has to be noted that numeric representation errors are a technical limitation of SAS (and other similar computational software) and that neither warning nor error messages appear in the SAS log.

Second, we suggest to perform additional rounding directly after calculating intermediate results such as Change from Baseline or unit conversion (namely, to perform the First sort of rounding in the Table 4). If the appropriate length of decimal places or significant digits is available, the concrete round function can be applied. For example, in the first example, we expect that the intermediate results (Unit-converted Change from BL) are accurate to the tenths decimal place, then we can apply the round function ROUND (*argument*, 0.1) directly after calculating the intermediate results (*argument* is a numeric constant, variable, or expression to be rounded). If the appropriate length of decimal places or significant digits is unclear, or if the same rounding procedure is performed for all of the intermediate results in a dataset without complicated rules, the general round function such as ROUND (*argument*, 1E-12) can be applied, with a rounding factor which is generally smaller than any value that may be encountered (Stutzman, 2014). The rounding approach is also consistent with the suggestions for conducting non-parametric statistical tests by PROC NPAR1WAY in the SAS documentation (SAS Support, Usage Note 23117): “If your response variable is calculated in a DATA step or procedure in SAS, rounding the response variable before running the PROC NPAR1WAY is especially necessary”.

In the above two examples, if round function ROUND (*argument*, 0.1) or ROUND (*argument*, 1E-12) is added directly after calculating the Unit-converted Change from BL, numeric representation errors can be prevented and the correct p-values can be achieved.

Moreover, if rounding has been performed directly after calculating intermediate results, other mistakes caused by numeric representation errors can also be prevented. For example, sub-setting datasets by employing IF, SELECT and WHERE statements or merging datasets using a non-integer by-variable may lead to erroneous results if additional rounding is not applied.

One of the major challenges to apply this approach in practice is the popular and common idea to keep full machine precision during intermediate calculations (Cole, 2015). Sometimes the sponsor requests to perform all computations prior to rounding. They prefer that intermediate calculations are not modified and that rounding is only performed in the final output of listing and tables. However, this and previous publications regarding numeric representation errors (SAS Technical Paper, 2014; Stutzman, 2014; Imelda, 2008) show that using the ROUND function in addition helps to avoid numeric representation errors and ensures correct results by means of SAS.

Regarding the objectives and timing of rounding, we want to distinguish two sorts of rounding for handling data in SAS. As summarized in Table 4, the First (Numeric Representation Errors, NRE) sort of rounding is to be performed directly after intermediate calculations, with the objective to ensure correctly calculated intermediate results without numeric representation errors. The Second (OUTPUT) sort of rounding is to be performed at the final presentation in listing and tables, with the objective to display the results in a comprehensible, readable and reasonable fashion. In Statistical Analysis Plan (SAP) there usually exists a section about how to present the data for the outputs, where the rounding rules for the Second (OUTPUT) sort of rounding may be defined. For the First (NRE) sort of rounding, as mentioned above, both general and concrete round functions can be used. General round functions can make the First (NRE) sort of rounding simpler and more practical, without assigning complicated rounding rules for each intermediate result. If general round functions are applied, then it may be necessary to conduct another concrete rounding step for the Second (OUTPUT) sort of rounding. In comparison, if the concrete rounding has been performed at the time of First (NRE) sort of rounding, then the two objectives of rounding have been achieved at the same time, and another rounding step for the final output of listings is probably not needed anymore.

An additional consideration for the rounding for the output listings and tables (OUTPUT sort of rounding) is the appropriate choice of the rounding unit. Too few decimal places may lead to unprecise values and incorrect ranks, while exaggerated precision may also be an issue. It is important to report the analytical results in a clear, unbiased manner, taking accuracy and precision into account (FDA, 2013). The data collected in a raw format should have reasonable decimal places, which usually depends on the precision of the instruments used for the measurements. To express arithmetic results with a proper number of digits, the principle is that numerical results are reported with a precision near that of the least precise numerical measurement used to generate the number (FDA, 2013). For more details and more complicated statistical measures, please refer to published recommendations (or reporting rules) (FDA, 2013; Cole, 2015). In most instances a reasonable combination of rules of decimal places and significant digits leads to a satisfactory result that is both precise and comprehensible (Cole, 2015).

Table 4 Two Sorts of Rounding for Handling Data in SAS

Sorts of Rounding	Objective of Rounding	Timing of Rounding	How to Round
First (NRE[1])	Avoid NRE, so as to ensure correct statistical results, correct sub-setting and merging of datasets, as well as correct comparison of variables	Directly after calculating intermediate results	General rounding, or Concrete rounding[2]
Second (OUTPUT)	Present the results in a comprehensible, readable and reasonable way	Final presentation in the listings and tables	Concrete rounding[2]

[1]Abbreviation: NRE: Numeric Representation Errors

[2]General rounding: rounding with a rounding factor which is generally smaller than any value that may be encountered, such as ROUND (argument, 1E-12); Concrete rounding: rounding with a clear rounding factor based on available appropriate lengths of decimal places or significant digits, such as ROUND (argument, 0.1).

Third, the validation of the assumptions for the statistical tests, as well as the size of treatment effects should be checked in addition to the p-value. The confidence interval around the effect size may also provide helpful warnings of numeric representation errors. As illustrated in the above examples, in order to prevent such misleading statistical results, blindly programming or just running macros for statistical tests should be avoided. The assumptions for the statistical tests should be considered, and descriptive analyses (such as mean, median, standard deviation, maximum and minimum values) should be performed before conducting inferential analyses. Furthermore, attention should be paid to the size of treatment effects, instead of only looking at the p-value (FDA, 1998). Example 2 above shows very small absolute differences where the clinical meaningfulness appears to be doubtful. Nevertheless, the very small p-values suggest a statistically highly significant difference. In these cases, careful thought should be given to values that could achieve statistical significance versus values that are clinically meaningful. Moreover, while p-values can only provide a dichotomous decision, confidence intervals can provide a range of likely values for the true effect size within a given confidence. In Example 2, as to the paired t-test, the effect size equals to $1.71 * 10^{-14}$, which is the mean of the paired differences of Product B minus Product A regarding the Sweat Weight (namely the mean of the last column in Table 3) and has a 95% confidence interval (CI) of $1.28 * 10^{-14}$ to $2.13 * 10^{-14}$. The tiny effect size as well as the tiny upper and lower CI limits can provide statisticians or programmers with two helpful warnings: First, is the statistically significant effect size clinically meaningful? Second, could the statistical significance be due to inadvertent numeric representation errors? These warnings may prompt statisticians or programmers to carry out further checks of their results.

Fourth, using integers can help represent numeric values exactly (SAS Technical Paper, 2014). Integer arithmetic can be performed in the expressions where the numeric representation error is not a problem anymore, as long as no values exceed the maximum size of integer that can be represented with confidence on the respective computational platform.

For example, in the Example 1, INT(t0 * 10000) and INT(t1 * 10000) can be applied directly after reading the source data. The scale factor 10000 helps to shift the fraction to the integer portion, and the INT function helps to remove any representation errors that might have been introduced by the fraction. By converting the non-integer values into integers at the beginning, numeric representation errors and misleading p-values in these two examples can be prevented.

One main advantage of this approach is that it tackles the numeric representation error at the beginning, so that the subsequent calculations of addition, subtraction and multiplication of non-integer values will not be affected by numeric representation errors. But the subsequent division, extraction of square root, logarithm or other complicated calculations may also encounter non-integer values and numeric representation errors. Another limitation is that if the results are reported, or descriptive or inferential analyses are conducted, it should always be kept in mind that the values were previously converted into integers by multiplication with a scale factor, resulting in additional division being necessary for back calculation. Furthermore, this approach may be only applicable for simple studies. For the majority of clinical research, it may be complicated and error-prone to consider and handle scale factors, decimal places as well as significant digits for each output and analysis results.

CONCLUSION

Although numeric representation errors seem to be very small, and neither warning nor error messages appear in SAS, overlooked tiny numeric representation errors may influence the statistical tests considerably and could reverse the direction of statistical conclusion. Numeric representation errors should be taken into account, especially before rank-based statistical tests or statistical tests in comparison to a reference value are performed. It is suggested to use the round function adequately in order to prevent misleading results. Attention should also be given to the size of treatment effects, assumptions of statistical tests and confidence intervals in addition to p-values. Using integers is another approach to avoid these problems. In summary, to avoid an incorrect statistical result, potential numeric representation errors should not be overlooked.

REFERENCES

Cole, T. J. (2015). “Too many digits: the presentation of numerical data”. Archives of disease in childhood. archdischild-2014.

FDA. (1998). Guidance for Industry, E9 Statistical Principles for Clinical Trials (ICH E9). Retrieved from

https://www.fda.gov/downloads/drugs/guidancecomplianceregulatoryinformation/guidances/ucm073137.pdf (accessed on SEP 20, 2018)

FDA. (2013). ORA Laboratory Manual, Volume III, Section 4-Basic Statistics and Data Presentation. Retrieved from https://www.fda.gov/downloads/ScienceResearch/FieldScience/LaboratoryManual/UCM092179.pdf (accessed on SEP 20, 2018)

Henrici, P. (1964). “Elements of numerical analysis”, New York: John Wiley & Sons. Inc.

Imelda, C.G. (2008). “Rounding in SAS®: preventing numeric representation problems”. SESUG

Lehrkamp, M. (2016). “Nicht jede Zahl ist das, was sie zu sein scheint“. KSFE (in German)

Liu, R., McKean, J.W. (2015). “Robust rank-based and nonparametric methods”. Switzerland: Springer

SAS Technical Paper. (2014). "Dealing with numeric representation error in SAS® applications". Cary, NC, USA: SAS Institute Inc.

SAS Support, Usage Note 23117: “Why does NPAR1WAY not find ties in my data set?”. Retrieved from http://support.sas.com/kb/23/117.html (accessed on SEP 20, 2018)

Stutzman, P. (2014). “What do you mean 0.3 doesn't equal 0.3? Numeric Representation and Precision in SAS and Why it Matters”. PharmaSUG

Tambascia. N. (2011). “Introduction to numeric precision and representation issues. Why 4.8 minus 4.6 is not always equal to 0.2”. PhUSE

Walker, G.A.,Shostak, J. (2010). “Common statistical methods for clinical research with SAS® examples”. Cary, NC, USA: SAS Institute Inc.

All output reported in this paper was generated by SAS Version 9.4 on the platform Windows Server 2012 Data Center.

ACKNOWLEDGMENTS

We would like to express our sincere thanks to Stefanie Krause, Frans Sollie, Waldorf Pasadaroha and Friedrich Vollmer for reviewing this paper. We would also like to thank Stefanie Krause and Susanne Schaefer for their support while preparing the paper and presentation. In addition, I am very grateful to my previous colleagues Friedrich Vollmer and Daniela Eichstedt for their encouragement and help of writing this paper.

CONTACT INFORMATION

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

Ying Zhou

PRA Health Sciences, Berlin, Germany

Puris Building, Potsdamer platz 10, 4th Floor, 10785, Berlin

Email: zhouying@prahs.com / smile.zhou@hotmail.com

David Haryanto

PRA Health Sciences, Berlin, Germany

Puris Building, Potsdamer platz 10, 4th Floor, 10785, Berlin

Email: haryantodavid@prahs.com

Jens Klement

PRA Health Sciences, Mannheim, Germany

Gottlieb-Daimler-Str. 10, 68165, Mannheim

Email: klementjens@prahs.com

APPENDIX

SAS SYNTAX FOR THE EXAMPLES

```
/*********************************************************************************
 * Example 1 Wilcoxon Rank-Sum Test, in Table 1 and Table 2
*********************************************************************************/
DATA example1;
        INFILE DATALINES DLM='09'x; *This is only necessary for SAS Enterprise Guide;
        INPUT Subjid Product $ para_t0 para_t1;
```

```
        DATALINES;
        1       A       0.1226  0.12
        2       A       0.1226  0.1194
        3       A       0.1224  0.1207
        4       A       0.1241  0.1233
        5       A       0.1255  0.1201
        6       A       0.133   0.1312
        7       A       0.1219  0.1203
        8       A       0.121   0.1211
        9       A       0.1227  0.1203
        10      A       0.1238  0.1204
        11      B       0.1226  0.121
        12      B       0.1194  0.1195
        13      B       0.1203  0.1188
        14      B       0.1326  0.1337
        15      B       0.1207  0.1189
        16      B       0.1227  0.1195
        17      B       0.1222  0.121
        18      B       0.1203  0.1195
        19      B       0.1231  0.1224
        20      B       0.1211  0.1203
;
RUN;

DATA example1;
        SET example1;
        para_diff_default = para_t1 - para_t0; *Change from BL, not-handled by rounding;
        para_diff_rounded   = ROUND ((para_t1 - para_t0), 0.0001); *Change from BL, handled by rounding;
        para_unitcved_default = (para_t1 - para_t0)*1000; *Unit-converted Change from BL, not-handled by
rounding;
        para_unitcved_rounded   = ROUND ((para_t1 - para_t0)*1000, 0.1); *Unit-converted Change from BL,
handled by concrete rounding;
        para_unitcved_rounded_general   = ROUND ((para_t1 - para_t0)*1000, 1E-12); *Unit-converted Change
from BL, handled by general rounding;
RUN;

PROC PRINT DATA=example1;
RUN;
PROC PRINT DATA=example1;
        FORMAT para_: 19.15;
RUN;
PROC PRINT DATA=example1;
        WHERE subjid in (1 2 3 16);
        FORMAT para_: HEX16.;
RUN;

*Rank;
PROC RANK DATA=example1 OUT=rank1;
        VAR para_unitcved_default;
        RANKS para_unitcved_default_rank;
RUN;
PROC RANK DATA=rank1 OUT=rank2;
        VAR para_unitcved_rounded;
        RANKS para_unitcved_rounded_rank;
RUN;
PROC PRINT DATA=rank2;
RUN;

*Wilcoxon Rank-Sum Test;
PROC NPAR1WAY DATA=example1 WILCOXON;
        CLASS product;
        VAR para_unitcved_default;
RUN;
PROC NPAR1WAY DATA=example1 WILCOXON;
```

```
          CLASS product;
          VAR para_unitcved_rounded;
RUN;
PROC NPAR1WAY DATA=example1 WILCOXON;
          CLASS product;
          VAR para_unitcved_rounded_general;
RUN;

PROC NPAR1WAY DATA=example1 WILCOXON;
          CLASS product;
          VAR para_diff_default;
RUN;
PROC NPAR1WAY DATA=example1 WILCOXON;
          CLASS product;
          VAR para_diff_rounded;
RUN;

*Compare not-handled (default) versus handled (rounded) numeric representation errors;
PROC COMPARE BASE=example1;
          VAR para_diff_default;
          WITH para_diff_rounded;
RUN; *There exists difference of approximately 10^-17 to 10^-19;
PROC COMPARE BASE=example1;
          VAR para_unitcved_default;
          WITH para_unitcved_rounded;
RUN; *There exists difference of approximately 10^-14 to 10^-16;
PROC COMPARE BASE=example1;
          VAR para_unitcved_rounded;
          WITH para_unitcved_rounded_general;
RUN; *All values compared are exactly equal;


/*********************************************************************
*Example 2, Wilcoxon Sign-Rank Test, and t-Test, in Table 3
*********************************************************************/
DATA example2;
          INFILE DATALINES DLM='09'x; *This is only necessary for SAS Enterprise Guide;
          INPUT A_wgt_m0 A_wgt_m1 B_wgt_m0 B_wgt_m1;
          DATALINES;
          119.4     130.2     119.2     130.0
          123.3     137.1     119.5     133.3
          121.0     132.6     120.3     131.9
          121.2     132.2     124.4     135.4
          123.3     136.1     119.5     132.3
          122.0     134.6     121.3     133.9
          118.2     131.4     118.6     131.8
          127.9     138.7     126.5     137.3
          135.8     147.0     167.2     178.4
          126.3     140.2     119.6     133.5
;
RUN;

DATA example2;
          SET example2;
          A_netwgt_default = A_wgt_m1 - A_wgt_m0;*Sweat Weight (Pad Net Weight) for Product A, not-handled by rounding;
          B_netwgt_default = B_wgt_m1 - B_wgt_m0;*Sweat Weight (Pad Net Weight) for Product B, not-handled by rounding;
          A_netwgt_rounded   = ROUND ((A_wgt_m1 - A_wgt_m0), 0.1);*Sweat Weight (Pad Net Weight) for Product A, handled by concrete rounding;
          B_netwgt_rounded   = ROUND ((B_wgt_m1 - B_wgt_m0), 0.1);*Sweat Weight (Pad Net Weight) for Product B, handled by concrete rounding;
          A_netwgt_rounded_general   = ROUND ((A_wgt_m1 - A_wgt_m0), 1E-12);*Sweat Weight (Pad Net Weight) for Product A, handled by general rounding;
```

```
	B_netwgt_rounded_general   = ROUND ((B_wgt_m1 - B_wgt_m0), 1E-12);*Sweat Weight (Pad Net Weight) 
for Product B, handled by general rounding;

	B_minus_A_default= B_netwgt_default - A_netwgt_default; *paired difference for Sweat Weight between 
Product B and Product A, not-handled by rounding;
RUN;

PROC PRINT DATA=example2;
	FORMAT A_netwgt_: B_netwgt_: B_minus_A_default 19.15;
RUN;

PROC COMPARE BASE=example2;
	VAR A_netwgt_rounded;
	WITH B_netwgt_rounded;
RUN; *All values compared are exactly equal;
PROC COMPARE BASE=example2;
	VAR A_netwgt_rounded_general;
	WITH B_netwgt_rounded_general;
RUN; *All values compared are exactly equal;
PROC COMPARE BASE=example2;
	VAR A_netwgt_default ;
	WITH B_netwgt_default ;
RUN; *There exists difference of approximately 10^-14;

*Wilcoxon Sign-Rank Test;
PROC UNIVARIATE DATA=example2 ALPHA=0.05;
	VAR B_minus_A_default;
RUN;

*t-Test;
PROC TTEST DATA=example2 H0=0 ALPHA=0.05 SIDES=2;
	VAR B_minus_A_default;
RUN;

/**********************************************************************************************
*Example.  0.5 – 0.25 – 0.125 – 0.125 equals to 0 whereas 0.3 – 0.2 – 0.1 doesn’t
**********************************************************************************************/
DATA example3;
	RETAIN result1 equal1 result2 equal2;
	LENGTH equal1 $50. equal2 $50. ;
	result1=0.5-0.25-0.125-0.125;
	IF result1= 0 THEN equal1="0.5-0.25-0.125-0.125 equals to 0";
		ELSE equal1="0.5-0.25-0.125-0.125 doesn't equal to 0";
	result2=0.3-0.2-0.1;
	IF result2= 0 THEN equal2="0.3-0.2-0.1 equals to 0";
		ELSE equal2="0.3-0.2-0.1 doesn't equal to 0";
RUN;

PROC PRINT DATA=example3;
	FORMAT result1 result2 25.22;
RUN;
```