PhUSE US Connect 2018

Paper DH13

Programmers are from Mars and Statisticians are from Venus

Deepak Ananthan, Zifo RnD Solutions, Chennai, India
Shweta Padmanaban, Zifo RnD Solutions, Chennai, India
Sangeetha Sathaiah, Zifo RnD Solutions, Chennai, India

ABSTRACT

How good would it be if programmers start talking statistician's language? If they start reading a table they generated as a story instead of just viewing it as mere numbers? If they could run the story in front of their eyes and identify oddities which could have been caused during study conduct or a simple programming error? True that it is not in a programmer's natural ability to do this. This paper discusses about bridging the gap thereby statisticians can spend their time on more meaningful analyses.

INTRODUCTION

In any clinical trial, once the last subject's last visit is completed, the next step is the most crucial and awaited part of the game – knowing the results! The clock starts to tick for Data Managers to lock the database, Statistical Programmers to run the outputs on final data, Statisticians to finalize and interpret the results. Any slight delay in one of these activities can have a ripple effect and cause major delays in releasing the study results.

Lot of times, oddities in data and programming errors can easily be identified by interpreting the summaries. Tables tell the story of what happened in the study and listings tell the story of what happened to a subject. Programmers often don't listen to these stories. When these are found during Statistical Review which is almost at the end of the game, it can be frustrating. Statisticians end up spending more time on identifying errors than the actual analysis and interpretations.

This paper gives a few examples of data issues and programming errors that could have been easily identified by the Statistical Programmers if they had worn the glasses of a Statistician.

WHAT IS ODD ABOUT YOUR ODDS RATIO?

Programmers consider the terms "p-value", "confidence interval", "odds ratio", etc. to be out of their league. Though defining and interpreting these requires statistical knowledge, there are simple rules to ensure that the results make sense.

Table 1. Odds Ratio for Primary Endpoints – Per Protocol Population

	Odds Ratio (Treatment vs. Placebo)	95% Confidence Interval	Unadjusted P-value
Endpoint 1	0.221	(0.004, 0.323)	0.6122
Endpoint 2	0.038	(0.153, 2.215)	0.2218
Endpoint 3	0.732	(0.061, 3.371)	0.7465

The above table may seem inconceivable at first, but if you observe closely it will tell you an obviously incorrect story. Odds ratio is simply the ratio between odds that an outcome can occur with exposure to treatment and placebo. If odds are same in both groups, odds ratios will be 1.

In the above table, for 'Endpoint 1', the confidence interval does not include '1' which means that the odds of treatment and placebo groups in the population are almost always different. Then the p-value (probability that there is no difference between treatment and placebo groups) should have been very low (<0.05, statistically significant) but that is not the case here. Similarly, the result for 'Endpoint 2' is very strange where the confidence interval does not include the estimate (Odds ratio). Confidence Interval is built around the estimates, so this is an obvious error. It is not possible that a sample has an estimate of x, which is outside the range of what we estimated for the population.

Both these issues could have been due to usage of incorrect formula for CI and p-values. By applying these simple rules to interpret, we can detect the error while programming and hence save time later.

NOT ALL RELATIONSHIPS ARE COMPLICATED

While every chapter is important, it is the continuity of the chapters that makes a story meaningful. When different summaries tell conflicting stories, it is important to understand why. Often, different programmers work on different sets of tables, due to which the relationship between outputs is overlooked.

Several tables within a study go hand-in-hand. Let's take an example in which we have adverse events leading to death. In this case, the Investigator marks the AE outcome as 'Death', and this is reflected in the overall summary table for the treatment group as shown below:

Table 2. Overview of All Adverse Events – Safety Population

	Treatment (N=10)	Placebo (N=10)	Total (N=20)
All adverse events	5 (50.0 %)	1 (10.0 %)	6 (30.0 %)
Serious adverse events	0	0	0
Treatment Emergent adverse events	2 (20.0 %)	1 (10.0 %)	3 (15.0 %)
Adverse Events leading to Death	2 (20.0 %)	0	2 (10.0 %)

If 2 subjects had adverse events leading to death, then in the Subject Disposition summary, at least two subjects should have reason for withdrawal as 'Death' in treatment group. But that is not reflected in the table below:

Table 3. Summary of Subject Disposition – Intent to Treat Population

Reason for Withdrawal	Treatment (N=10)	Placebo (N=10)	Total (N=20)
Lost to Follow-up	1 (10.0 %)	0	1 (5.0 %)
Withdrawal of Consent	0	0	0
Protocol violation	2 (20.0 %)	1 (10.0 %)	3 (15.0 %)
Death	1 (10.0 %)	0	1 (5.0 %)
Physician decision	0	0	0

Tracing this back to analysis datasets, there was a discrepancy between death flag in ADSL and ADAE datasets. Further investigation revealed that the investigator marked the AE outcome as 'Death' in the AE form, but chose a different reason for discontinuation in the Early Termination form. Since the flag in ADSL was pulled from the Early Termination form as opposed to the flag in AE which came from AE form, it resulted in discrepancy.

WHEN THE OPERATION IS UNSUCCESSFUL

There are numerous functions that can be used to derive the same result in SAS. There is hardly any restriction in choosing an operator and each programmer has their own style. Programming should be based on the purpose and not convenience. Let's take an example which is quite common in clinical trials. The study has 2 study periods where the subjects will move from period 1 to period 2. One subject discontinues after the first period. We're interested in the distribution of treatment duration of the subjects in both periods.

Table 4. Snapshot of Exposure data

Subject	Treatment Duration (days)		
	Period 1	Period 2	Overall
001-001	5	3	8
001-002	5	3	8
002-001	5	.	(5 or Missing?)
002-002	5	3	8
003-001	5	2	7
003-003	5	3	8

It seems obvious that the overall duration for '002-001' is 5 days and not missing. Overall duration is a simple addition of duration in period 1 and period 2. The programmer can either use the function sum(Period_1, Period_2) or the traditional operator Period_1 + Period_2. However similar it may seem, both performs different operations. The former returns the value of 5 (ignoring the missing value in period 2) and the latter returns missing (taking the missing value into consideration).

Table 5. Summary of Treatment Exposure – Safety Population

	Period 1 (N=6)	Period 2 (N=5)	Overall (N=6)	
			Incorrect (+)	Correct (sum)
Treatment Duration (days)				
n/nmiss	6/0	5/1	5/1	6/0
Mean	5.0	2.8	7.8	7.3
Standard Deviation	0.00	0.45	0.45	1.21
Median	5.0	3.0	8.0	8.0
Min, Max	5, 5	2, 3	7, 8	5, 8

As anyone can see, the correct and incorrect columns above give totally different interpretations. The incorrect interpretation is that only 5 subjects had taken treatment, and the duration ranged between 7-8 days for all 5 subjects. But the correct interpretation is that, all 6 subjects had taken treatment and the duration ranged between 5-8 days in the study. It is evident that one subject discontinued after period 1. This doesn't require any statistical knowledge; good understanding of the study could solve this right at the programming stage. Some casual choices the programmer makes can lead to unforeseen impact!

STATIC OR DYNAMIC?

Program re-runs to generate outputs for latest data is pretty common when there are interim safety reviews. Since the turnaround time for such re-runs is very limited, we only have time to run the programs and perform an overall review of the outputs. We may overlook the challenges posed by a live database that has not yet been locked.

Working with MedDRA coded terms, there are high chances of having adverse events that are not yet coded for the most recent data. We summarize them in the tables as 'Not Coded' as shown below.

Table 6. Summary of Adverse Events – Safety Population

System Organ Class Preferred Term	Treatment (N=21)	Placebo (N=10)	Total (N=31)
General Disorders	15 (71.4%)	5 (50.0%)	20 (64.5%)
Fever	12 (57.1%)	3 (30.0%)	15 (48.4%)
Cold	7 (33.3%)	3 (30.0%)	10 (32.3%)
Lungs	2 (9.5%)	2 (20.0%)	4 (21.9%)
Lung Carcinoma	0	1 (10.0%)	1 (3.2%)
Not Coded	2 (9.5%)	1 (10.0%)	3 (9.7%)
Not Coded	2 (9.5%)	1 (10.0%)	3 (9.7%)

Note: Subjects are counted once per System Organ Class and once per Preferred Term.

What we accept in a dynamic database may not necessarily be acceptable for a completed study. In this case, it is essential that all terms are coded at the end of the study, when the database is locked. In the above case, programmers verify the numbers and ensure they pass validation. But when we take a step back, it is not actually hard to realize that all events should have been coded before final analysis. Looking out for such signals and raising an alert flag where needed is highly critical. Such signals can also be programmed proactively so that SAS throws a warning or error in log until it's resolved.

WHEN THE INVESTIGATOR HAS BUTTERFINGERS

Data entry errors are quite common and summaries easily throw spotlight on them. With good understanding of study, we can gauge the possible ranges within which certain results can lie. For example, in an adult clinical trial, the age should range from 18 to 65 years or as defined in the inclusion/exclusion criteria. This is not always the case though. In the below demographic summary, the minimum value for age in treatment group is 0 years.

Table 7. Summary of Subject Demographics – Enrolled Population

	Treatment (N=21)	Placebo (N=10)	Total (N=31)
Age (years)			
n	21	10	31
Mean	38.5	34.8	35.3
Standard Deviation	10.56	6.77	9.94
Median	35.5	36.7	38.6
Min, Max	0, 65.6	19.5, 59.9	0, 65.6

How can a person of age 0 enter the trial? It is indisputable that the recorded age is incorrect. In the demographic form in EDC, the Investigator mistakenly entered same dates for informed consent date and birth date. As a result, the EDC calculated the age as 0 years.

Figure 1. Snapshot of Subject Demographics Page

There can be cases where this could be due to programming errors too. A quick scan of the descriptive statistics can instantly reveal such issues.

OVER DOSE: PILLS OR DOSAGE?

Data listings are commonly seen as data dump and not interpreted. But the truth is, listings back up the results shown in summaries and provide more insights.

In this study, all subjects take 2 pills with different dosages (50 mg or 75 mg or 100 mg) depending on their assignment. If a subject is assigned 150 mg, they will be dispensed two tablets: one with 100 mg, the other with 50 mg. Subject '002-003' was assigned 150 mg but took 100+75 mg pills. The first question to ask is, whether the tablet was dispensed as per assigned dose?

Table 8. Study Medication Compliance – Safety Population

		# Tablets Dispensed		
Subject	**Assigned Dose**	**50 mg**	**75 mg**	**100 mg**
001-001	150 mg BID	20		20
002-001	175 mg BID		25	25
002-002	200 mg BID			40
002-003	150 mg BID		25	25
003-001	150 mg BID	20		20

This could lead to drug overdose because the subject was dispensed incorrect dosage (75 mg instead of 50 mg). Or this could have been a data entry error. Or the programmer could have mapped the value to the incorrect column. Listings are not just appendices and interpreting listings could possibly reveal a lot of data issues which are critical in a study. Issues that are not being identified in summaries could be easily captured by understanding the listings.

OUTLIERS OR LIARS?

We often confuse data issues with outliers. Outlier is an abnormal value recorded and is valid. If the heart rate of a subject is entered as 1000 beats/minute, that cannot be considered as an outlier because such a result is not possible.

Vital signs, Physical Examination and ECG are some of the measurements that have standard normal ranges and units across the globe. In clinical space, applying this knowledge while reviewing the tables can be very useful. PR Interval in msec usually ranges from 120-200 msec. But the maximum value recorded was > 1000 msec as shown below, which is impossible.

Table 9. Summary of ECG Parameters – Safety Population

Parameter Statistic	Treatment (N=21)	Placebo (N=10)	Total (N=31)
PR Interval (msec)			
n/nmiss	20/1	10/0	30/1
Mean	234.73	183.34	187.28
Standard Deviation	35.921	11.429	30.984
Median	189.50	190.40	189.50
Min, Max	115.3, 1005.0	127.4, 205.3	115.3, 1005.0

To validate any summary, programmers verify the numbers with raw data and the formats with the Analysis Plan (SAP). By taking one more step in checking if the summaries make sense or not, we can identify any oddities in data.

The above summary was due to a data entry error. The investigator entered '1005' instead of '100.5' which hugely influenced the results. Capturing this at the programming stage would expedite the process of raising queries and resolving them, rather than noticing this at Statistical Review stage, when all outputs are considered close to final.

KNOWING YOUR UNKNOWN

In clinical data management, everyone understands the pain of processing the external data when there is enormous amount of data points making manual quality check almost impossible. Take an example of plasma concentration data summary in a Single Ascending Dose (SAD) phase.

Table 10. Summary of Plasma Concentration (nM) – Safety Population

	50 mg (N=8)	100 mg (N=8)	150 mg (N=8)	200 mg (N=8)	250 mg (N=8)
Day 1, 2 hrs post dose					
n/nmiss	8/0	6/2	8/0	0/8	
Mean	3.29	2.92	0.00	NA	
Standard Deviation	2.02	2.49	0.00	NA	
Median	4.14	3.97	0.00	NA	
Min, Max	0.0, 5.1	0.1, 5.9	0.0, 0.0	NA, NA	

When we look at 150 mg and 200 mg cohorts, one had all results as 0 and another one had all results as missing. Does this mean all subjects in 150 mg cohort had '0 nM' plasma concentration and all subjects in 200 mg had 'missing' results? Plasma concentration is usually not detectable for subjects in placebo group, but we know that is not the case here.

Further digging back into the external data, some missing data were entered as 0 and some were left blank. While the labs may not find it important to distinguish unknowns and zeros, this has a huge impact in the world of statistics. The worst nightmare a Statistician could have is when missing is assumed to be zero or the vice versa because their interpretations are contrasting.

What about the 250 mg cohort in the table? Either this cohort was not expected to have plasma concentration measured at this time point, or the column was incorrectly displayed in the program. Only understanding of the study schedule can validate which one is right.

CONCLUSION

Data issues and programming errors can be identified at various stages: edit checks, offline checks, data cleaning, double programming etc. Large amount of study data that is collected, queried, cleaned and standardized conceive a meaningful story in the form of TLF outputs. Statistical programmers get to read this story before anyone else. If they can understand and challenge it at an early stage, the quality of the outputs that goes for final Statistical Review can be greatly improved. Thus, Statisticians can spend more time on analyzing and exploring underlying trends in the data.

CONTACT INFORMATION

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

1. Author Name: Deepak Ananthan
   Company: Zifo RnD Solutions
   Address: 21A, Anna Salai, Littlemount, Saidapet, Chennai, Tamil Nadu 600015
   City / Postcode: Chennai / 600015
   Work Phone: +91 44 43114002
   Fax:
   Email: Deepak.A@zifornd.com
   Web: www.zifornd.com

2. Author Name: Shweta Padmanaban
   Company: Zifo RnD Solutions
   Address: 21A, Anna Salai, Littlemount, Saidapet, Chennai, Tamil Nadu 600015
   City / Postcode: Chennai / 600015
   Work Phone: +91 44 43114002
   Fax:
   Email: Shweta.P@zifornd.com
   Web: www.zifornd.com

3. Author Name: Sangeetha Sathaiah
   Company: Zifo RnD Solutions
   Address: 21A, Anna Salai, Littlemount, Saidapet, Chennai, Tamil Nadu 600015
   City / Postcode: Chennai / 600015
   Work Phone: +91 44 43114002
   Fax:
   Email: Sangeetha.S@zifornd.com
   Web: www.zifornd.com