Paper CC01

Data cleaning and spotting outliers with UNIVARIATE

Michael Auld, Eisai Ltd, London UK

ABSTRACT

Timely and strategic cleaning of data is crucial for the success of the analysis of a clinical trial. I will demonstrate 2-step code to identify outlier observations using PROC UNIVARIATE and a short data step. This may be useful to anyone attempting to clean systematic data conversion errors in large data sets like Laboratory Test Results.

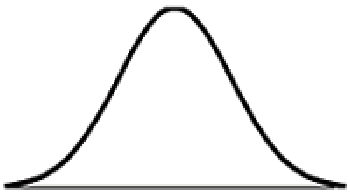
INTRODUCTION

Data quality is essential for the trustworthiness of the final analysis that determines if a drug is efficacious or safe, and thus worthy of regulatory approval. Computers have long helped play a part in bringing the products to market faster, and with the introduction of EDC or Electronic Data Capture, to replace the paper-based Case Report Form or CRF, expectations are that these timelines should fall further.

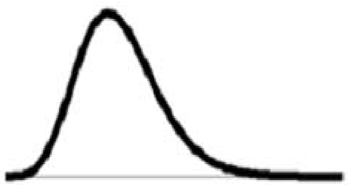
However, there still remains a risk that data may not be quite what it was expected to be. This may be attributable to unforeseen bias emerging at sites, and so it is useful to establish this as early as possible in the trial to determine if the expected statistical model was wrong. It may also be indicative of errors in the data entry into the database.

STATS TEST DUMMIES

To illustrate this, take a look at these two distribution curves.



Normal Distribution



Non-symmetrical F-shaped curve

The first one shows a normally distributed population, common in most high school math(s) book(s). The second shows an F-shaped curve, similar to the first but biased to the right fringes. I won't explain here about Distribution curves or Anova, but I only want to focus on the interesting bits at the fringes, known as the outliers and why they could be important to a clinical trial.

When outliers become extreme observations at either the left or the right it could alter the assumptions made by the statistician at study set-up about the behaviour of the recruited population - which could jeopardise the proof of the trial and ultimately expensive failure.

PROC UNIVARIATE TO THE RESCUE

The SAS® procedure UNIVARIATE is a very sophisticated tool that has a lot of statistical weaponry that it has accumulated over the years, most of which I personally don't understand or use (I am not a statistician!). My main use in the past as a SAS programmer was to get the statistics required for Table outputs not found in PROC MEANS or SUMMARY.

Invoking ODS TRACE ON in your program and issue a PROC UNIVARIATE you can see the datasets that can be made use of. For a large amount of data or data analysed by a large group of parameters, a large amount of pages may be written. Using ODS SELECT can help to cut-down this output into something more manageable to read.

```
ODS TRACE ON;
PROC UNIVARIATE DATA=sds.lb;
  CLASS lbtest;
  ID usubjid;
  VAR lbstresn;
RUN;
ODS TRACE OFF;
```

The data sets that are available (the names of which are written to the LOG window) correspond with the default output produced by the procedure in that order.

```
Output Added:
-------------
Name:       Moments
Label:      Moments
Template:   base.univariate.Moments
Path:       Univariate.aval.Moments
-------------

Output Added:
-------------
Name:       BasicMeasures
Label:      Basic Measures of Location and Variability
Template:   base.univariate.Measures
Path:       Univariate.aval.BasicMeasures
-------------

Output Added:
-------------
Name:       TestsForLocation
Label:      Tests For Location
Template:   base.univariate.Location
Path:       Univariate.aval.TestsForLocation
-------------

Output Added:
-------------
Name:       Quantiles
Label:      Quantiles
Template:   base.univariate.Quantiles
Path:       Univariate.aval.Quantiles
-------------

Output Added:
-------------
Name:       ExtremeObs
Label:      Extreme Observations
Template:   base.univariate.ExtObs
Path:       Univariate.aval.ExtremeObs
-------------

Output Added:
-------------
Name:       MissingValues
Label:      Missing Values
Template:   base.univariate.Missings
Path:       Univariate.aval.MissingValues
-------------
```

EXTREME VALUES

The extreme observations are the ones of interest and deserve our attention as being more than just the normal outliers at the end of the bell-curve. These are the ones that skew the distribution into the F-shape shown earlier. UNIVARIATE by default lists the top 5 and bottom 5 observations (identified by the VAR statement) ranked in term of their value. The numbers to be displayed can be controlled by the NEXTROBS option of UNIVARIATE

```
ODS SELECT ExtremeObs;
PROC UNIVARIATE DATA=sds.lb NEXTROBS=10;
  CLASS lbtest;
  ID usubjid;
  VAR lbstresn;
RUN;
```

```
                         The UNIVARIATE Procedure
                           Variable:  lbstresn
                         LBTEST = ABS. NEUTRO.COUNT

                          Extreme Observations

--------------------Lowest-------------------      ------------------Highest-----------------

  Value   subjid                          Obs        Value   subjid                          Obs

0.00000   0074-0018                    459425         2730   0067-0017                    412339
0.00000   0053-0008                    311137         2920   0067-0017                    412471
0.00000   0053-0008                    311125         2920   0067-0017                    412472
0.00036   0100-0012                    607593         3200   0067-0017                    412498
0.00215   0033-0013                    188278         3200   0067-0017                    412499
0.01000   0048-0019                    279017         3500   0067-0017                    412525
0.01000   0048-0019                    279016         3500   0067-0017                    412526
0.01500   0084-0008                    511085         3680   0067-0017                    412432
0.01900   0064-0013                    397497         3680   0067-0017                    412433
0.02000   0048-0019                    279007         5330   0059-0005                    352791
                                      The SAS System

                         The UNIVARIATE Procedure
                           Variable:  lbstresn
                            LBTEST = ALBUMIN

                          Extreme Observations

-------------------Lowest------------------      ------------------Highest------------------

Value   subjid                          Obs        Value   subjid                          Obs

0.029   0027-0008                    147516         70.5   0017-0019                     91342
0.031   0015-0007                     71825         70.9   0017-0019                     91369
0.034   0015-0007                     71852         71.1   0017-0019                     91528
0.034   0015-0007                     71720         71.8   0017-0019                     91648
0.036   0101-0014                    613366         72.0   0052-0018                    302167
0.036   0091-0017                    554794         73.0   0022-0013                    121578
0.037   0091-0017                    554369         75.0   0053-0005                    309194
0.037   0059-0010                    354782         77.8   0052-0018                    303814
0.037   0059-0010                    354743         97.0   0052-0006                    297772
0.037   0027-0008                    147252        470.0   0076-0005                    467674
```

The number of extreme observations may vary from parameter to parameter, but as a quick, dirty way to identify dirty data this method is still quite effective. The option NEXTRVALS does a similar thing by showing the extreme values. What they both lack is the context of the extreme values compared to the rest of the data in the curve. This is why the Quantiles analysis is the most useful.

The easiest thing to do would be just to hand a full list of the percentiles at 5, 95, 1 or 99, 10 or 90 for each parameter, but that would be too much data, and would merely hide the true negatives amongst some false negatives. A manual read of the outlier values in context by scrolling through page after page of output and intervening when a value “jumps out” as an outlier is closer to what we want to achieve – and indeed this rather tedious method is how I have performed this task in the past.

```
                              The SAS System

                          The UNIVARIATE Procedure
                            Variable:  lbstresn
                        LBTEST = ABS. NEUTRO.COUNT

                         Quantiles (Definition 5)

                          Quantile       Estimate

                          100% Max       5330.000
                          99%              10.380
                          95%               6.424
                          90%               5.350
                          75% Q3            3.960
                          50% Median        2.910
                          25% Q1            2.120
                          10%               1.512
                          5%                1.110
                          1%                0.450
                          0% Min            0.000
                              The SAS System

                          The UNIVARIATE Procedure
                            Variable:  lbstresn
                              LBTEST = ALBUMIN

                         Quantiles (Definition 5)

                          Quantile      Estimate

                          100% Max       470.000
                          99%             61.700
                          95%             53.000
                          90%             49.000
                          75% Q3          46.000
                          50% Median      43.000
                          25% Q1          39.550
                          10%             36.300
                          5%              34.000
                          1%              26.000
                          0% Min           0.029
```

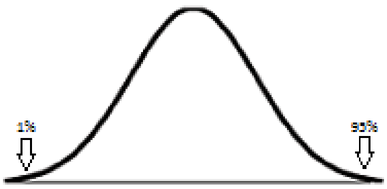
Note the extreme gap between 100th and 99th percentiles

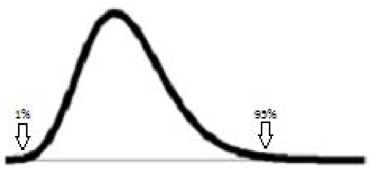
Large gap also observed between 0th and 1st percentiles

Of course, this is both time consuming and prone to error. Another drawback is the format of the output - when trying to relay this information back to Data Management, patient IDs and other contextual information to help identify the potentially erroneous observations are required. I began to analyse the human process and try to formulate the logic that SAS could use to perform the same task.

METHOD AND REASONING

To establish what distinguishes the percentiles as extreme – the bit that makes them "stand out" required some experimentation, but the idea was the context or the value relative to the rest of the data.





When the data looked right, it was when the distance between the 95th percentile and the maximum value was greater than the distance between all the rest of the quantiles. My method was to try and predict what the 0th and the 100th percentiles *should* have been, had the data been as expected. To get a reliable average centile, I used PROC UNIVARIATE and took the 95th and the 5th. As I was no longer taking the default printed output, I no longer required the ods select, and have issued both an OUTPUT statement and NOPRINT option. The ID statement is no longer required, as the quantiles cover a range of patient ID rather than specific ones.

```
PROC UNIVARIATE DATA=sds.lb NOPRINT;
  CLASS lbcat lbtest;
  VAR lbstresn;
  OUTPUT OUT=mydata PCTLPTS=5 95 MIN=min MAX=max PCTLPRE=p;
RUN;
```

The output data set produced contains an observation for each parameter, and 4 other columns, P0, P5, P95 and P100. The same result can be achieved by using

```
  output out=mydata min=min p5=p5 p95=p95 max=max;
```

but that takes longer to write and curiously longer for SAS to process!

	LAB Test or Examination Name	the largest value, lbstresn	the smallest value, lbstresn	the 5.0000 percentile, lbstresn	the 95.0000 percentile, lbstresn	pn	p0	p100
1	ABS. BAND COUNT	0	0	0	0	0	0	0
2	ABS. BASO.COUNT	20	0	0	0.1	0.0011111111	0	0.1055555556
3	ABS. EOS. COUNT	200	0	0	0.33	0.0036666667	0	0.3483333333
4	ABS. LYMPH. COUNT	2390	0.06	0.7	2.52	0.0202222222	0.5988888889	2.6211111111
5	ABS. MONO. COUNT	800	0	0.14	0.981	0.0093444444	0.0932777778	1.0277222222
6	ABS. NEUTRO.COUNT	5330	0	1.11	6.424	0.0590444444	0.8147777778	6.7192222222
7	ALBUMIN	470	0.029	34	53	0.2111111111	32.944444444	54.055555556
8	ALKALINE PHOSPHATASE	4490	1.99	57	427	4.1111111111	36.444444444	447.55555556
9	ALT (SGPT)	679	0.28	10.5	80	0.7722222222	6.6388888889	83.861111111
10	APTT	40.6	20	20	40.6	0.2288888889	20	40.6
11	AST (SGOT)	683	0.27	15	80	0.7222222222	11.388888889	83.611111111
12	BANDS	819	0	0	10	0.1111111111	0	10.555555556
13	BASOPHILS	13.4	0	0	1.64	0.0182222222	0	1.7311111111
14	BILIRUBIN	0	0	0	0	0	0	0
15	BUN	6.4	6.4	6.4	6.4	0	6.4	6.4
16	CALCIUM	214	0.3175	2.07	2.66	0.0065555556	2.0372222222	2.6927777778
17	CHLORIDE	250	10.5	96	110	0.1555555556	95.222222222	110.77777778
18	CREATININE	8309.6	12.25	46	101	0.6111111111	42.944444444	104.05555556
19	EOSINOPHILS	74	0	0	6	0.0666666667	0	6.3333333333
20	GGT	884	14	16	794	8.6444444444	14	837.22222222
21	GLUCOSE	38.14	2.1	4.22	8.51	0.0476666667	3.9816666667	8.7483333333
22	HEMATOCRIT	426	0.198	0.362	42.6	0.4693111111	0.198	44.946555556
23	HEMOGLOBIN	12300	3.8	100	144	0.4888888889	97.555555556	146.44444444
24	LDH	47000	0.56	146	845	7.7666666667	107.16666667	883.83333333
25	LYMPHOCYTES	2368	0	14	50	0.4	12	52
26	MAGNESIUM	121	0.09	0.6724	1.05	0.0041955556	0.6514222222	1.0709777778
27	MONOCYTES	444	0	2	15	0.1444444444	1.2777777778	15.722222222
28	NEUTROPHILS	8775	0	36	76.8	0.4533333333	33.733333333	79.066666667
29	PHOSPHORUS	132	0.16	0.87	1.568	0.0077555556	0.8312222222	1.6067777778
30	PLATELET COUNT	161000	0.139	139	403	2.9333333333	124.33333333	417.66666667
31	POTASSIUM	18	1.2	3.57	5.01	0.016	3.49	5.09
32	PROTEIN	500	0	0	0.3	0.0033333333	0	0.3166666667
33	PROTHROMBIN TIME	51.5	11	11	51.5	0.45	11	51.5
34	PTT	60	25.1	25.1	60	0.3877777778	25.1	60

The next step was to calculate the nth percentile in a data step by subtracting p5 from p95, then dividing by 90. The projected or expected min or max value (based upon our calculated average gap) can then be derived. Note that if the projection exceeds the observed min or max values then the projection is reset to these observed values – which is a healthy indication for that data to be clean.

```
DATA nthdegree;
  SET mydata(WHERE=(NOT MISSING(max)));
  pn = (p95 - p5)/90;
  p0 = MAX(p5 - (5*pn), min);
  p100 = MIN(p95 + (5*pn), max);
RUN;
```

The final step is to then use this data set to select the extreme observations that fall outside the projected min and max and the observed min and max. The SAS key words are capitalised to distinguish the variable names min and max from the SAS functions MIN and MAX.

The SQL join here is a useful technique employing a merge of data sets on values falling in a particular range, and unless I'm mistaken something that can only be done in SQL and not in a data step. (MERGE BY works on exact matches of the key variables mentioned in the BY statement).

```
PROC SQL NOPRINT;
  CREATE TABLE lab_outliers as
  SELECT lb.*
        ,extreme.min
        ,extreme.p0
        ,extreme.p5
        ,extreme.p95
        ,extreme.p100
        ,extreme.max
  FROM nthdegree AS extreme LEFT JOIN sds.lb
  ON lb.lbcat EQ extreme.lbcat
  AND lb.lbtest EQ extreme.lbtest
  AND ((extreme.min <= lb.lbstresn < extreme.p0)
  OR (extreme.p100 < lb.lbstresn <= extreme.max))
  ORDER BY usubjid, lbcat, lbtest, visitnum
  ;
QUIT;
```

It is worth emphasizing that this method may still just reflect the actual observed extremes in the collected data, and indicate an observed skew in the population, but at other times data error (either at entry or at conversion).

The extreme data can then be sent back to Data Management for querying – and as it is in data set form this could be in the form of a transport file or an excel spreadsheet ready for turnaround.

FURTHER READING

CODY, Ron, Cody's Data Cleaning Techniques Using SAS, SAS Press Series 2008

Base SAS Procedures Guide, SAS Publishing

CONTACT INFORMATION

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

michael_auld@eisai.net