Automatic Estimation of Dixon’s Test for Extreme Values Using a SAS® Macro Driven Program  
Don Li and Edmond Edwards, Biovail Contract Research, Toronto, ON, Canada

ABSTRACT
Dixon’s test[1,2] for extreme values is a convenient method to statistically determine outliers from continuous data. Based on Dixon’s simple algorithms, this paper describes a way of automatically estimating outlying values conditionally from the parameters of bioequivalence study data using a SAS macro driven program. The key points of programming strategy are listed and the model result is shown in the paper. This program might be an example that how to use SAS to handle the user defined conditional calculations and logical comparisons.

INTRODUCTION
There are several methods available for statistically testing the outlying values in a continuous variable. Among those, Dixon’s test for extreme values is a very simple, straightforward, easy computing and explaining one. Although the mathematical process is as simple as one plus two, the way of calculating the value of r according to the subject number and of matching r with the special distributed criteria takes couple manually processing steps. The purpose of this article is to provide the solution of having this job done by a simple clicking through a SAS macro.

METHODS
ALGORITHM
Dixon’s test looks the ratio between the difference of the minimum/maximum value and its neighbor value and the difference of applied maximum and minimum values. This ratio is calculated as r and supposed r is following a certain distribution. However, the applied minimum/maximum values and related neighbor values are conditionally dependent.

Suppose there are k subjects in the data. If looking for extreme value at lowest end:

\[ r_L = \frac{(X_n - X_1)}{(X_m - X_1)} \]

If looking for extreme value at highest end:

\[ R_H = \frac{(X_k - X_m)}{(X_k - X_n)} \]

Then the calculated value of \( r_L \) and \( R_H \) are compared with the criteria provided by the method for determining if the lowest/highest value is an outlier[1].

Obviously the complicity here is not calculation but the logic processing and multiple comparison and judging from comparison. This is, however, the advantage the programming languages like SAS possess.

PROGRAMMING STRATEGY
The program is especially established for the two or three way cross-over bioequivalence studies. In this kind of data, the parameters under study are the ratios from \( \text{AUC}(0-t) \), \( \text{AUC}(0-\infty) \) and \( \text{Cmax} \). The outlier test is to find the subjects with ratio that is statistically out of the reasonable range of distribution. The program is designed to handle the different comparison of drug formulation (such as A vs. B, A vs. C or B vs. C etc.). The key points of doing Dixon’s test with a SAS program are:

- Set up direct paths to get connected with the source SAS data set (with libname definition) and output file (with printto procedure).

\[ \text{libname sasdata } "&\text{indata}"; \]
\[ \text{proc printto print}="&\text{output}"; \]

The macro variables will be defined in macro calling.

- Make the program flexible for tackling different comparison combination (by adding macro parameters in calling; e.g. for testing drug: t=A, and for reference drug: r=B).
• Store the distribution criteria in data with sequence so it can be handled easily with ARRAY.

```
data rtable;
array rr{28} r1-r28;
array nn{28} n1-n28;
do i=1 to 28;
   input nn{i} rr{i} @@;
end; drop i; cards;
3 0.941 4 0.765 5 0.642 6 0.560 7 0.507
8 0.554 9 0.512 10 0.477 11 0.576 12
0.546 13 0.521 14 0.546 15 0.525 16
0.507 17 0.490 18 0.475 19 0.462 20
0.450 21 0.440 22 0.430 23 0.421 24
0.413 25 0.406 26 0.399 27 0.393 28
0.387 29 0.381 30 0.376;
run;
```

• Remove the missing ratios before the Dixon’s test, especially for AUC_{0-inf}, so that no invalid missing value is taken in to calculation and generates false r value. Determine subject number and save it in a macro variable ‘subn’.

• Conditionally calculate the r-values according to the number of subjects using a data ARRAY.

```
DATA _null_; 
set aryauct;
if &_subn1<8 then do;
   rlow=(x2-x1)/(x&_subn1-x1);
   rhig=(x&_subn1-x%eval(&_subn1-
1))/(x&_subn1-x2);
end;
else if &_subn1<11 then do;
   rlow=(x2-x1)/(x%eval(&_subn1-1)-x1);
   rhig=(x&_subn1-x%eval(&_subn1-
1))/(x&_subn1-x2);
end;
else if &_subn1<14 then do;
   rlow=(x3-x1)/(x%eval(&_subn1-2)-x1);
   rhig=(x&_subn1-x%eval(&_subn1-
2))/(x&_subn1-x3);
end;
else do;
   rlow=(x3-x1)/(x%eval(&_subn1-2)-x1);
   rhig=(x&_subn1-x%eval(&_subn1-
2))/(x&_subn1-x3);
end;
call symput('rlow',rlow);
call symput('rhig',rhig);
run;
```

• Store the calculation results in to macro variables for later uses (using data driven macro statement call symput, as shown above).

• Compare the calculated r values by n with the critical r value by same n (obtained through ARRAYS);

```
data _null_; 
set rtable; rv=0.3500;
array rr{28} r1-r28;
array nn{28} n1-n28;
do i=1 to 28;
   if &_subn1=nn{i} then rv=rr{i};
end;
call symput('rv',put(rv,8.4));
run;
```

and generate variables telling whether the extreme values are outliers accordingly. Save the result of comparison in an output data.

• Summarize the results of output data in a table report using customized format (see the example).

The macro requests just the routes for in-data and out-result and the compared formulations (usually represented as A, B and C).

Macro calling:

```
%dixon(lib=path, outfile=path, t=A, r=B)
```

For example:

```
%dixon(lib=h:\studies\2343\data,
    outfile=h:\studies\2343\result\outlier.txt,
    t=A, r=C)
```

RESULTS

DETAILED REPORT

The results is reported separately for AUC\((0-t)\), AUC\((0-\infty)\) and Cmax. The table format looks like:

<table>
<thead>
<tr>
<th>RANK</th>
<th>SUBJECT</th>
<th>AUC RATIO</th>
<th>OUTLIER?</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>29.3570</td>
<td>NO</td>
<td>0.1922</td>
</tr>
<tr>
<td>2</td>
<td>13</td>
<td>36.1282</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>53.7543</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>04</td>
<td>56.6195</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>06</td>
<td>62.1957</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>25</td>
<td>63.4342</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>151.1212</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>09</td>
<td>154.5113</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>21</td>
<td>156.3150</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>22</td>
<td>201.5591</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>11</td>
<td>224.7652</td>
<td>NO</td>
<td>0.4003</td>
</tr>
</tbody>
</table>

Critical r value: 0.4130 (n=24)
SUMMARY TABLE

The summary table with conclusion of the findings of the three parameters under study is generated like:

SUMMARY TABLE OF OUTLIER TEST: Dixon's Method
STUDY # 9999. COMPOUND NAME: XXXX
A vs. B
SUBJECT(*)  AUC(0-t)  AUC(0-inf)  Cmax  CONCLUSION
--------------------------------------------
03          -        -         -        NO
11          -        -         -        NO
12          -        -         -        NO

(*) Subjects at the extremes and whose values were tested as outliers

DISCUSSION AND CONCLUSION

From statistical point of view, outliers refer to the relatively small or large values in a data set, and their values are statistically different from, and not belong to, the main body of data. The existing of outliers may dramatically change the statistical conclusion, for instance the conclusion of bioequivalence, on the data, especially by some marginal fail situations.

The method of testing outliers from Dixon and correspond SAS program are discussed in this article. The conditional calculations and comparisons in the process of Dixon's test for extreme values are compiled in a SAS macro driven program. This program integrates the processes in to a one-sentence macro calling. It can handle the flexible combination of different formulation comparisons and save the result in either detailed and summarized report files. In the same way, other similar calculations from certain processes can be programmed (for instance T-procedure of outlier test). This simple strategy might just be an example of how the several steps job can be organized to one through SAS macro programming.

Statistically determining outlying values is just a beginning of answering the question if the subject with outlying values should be excluded from the analysis. Although frequently the clients require outlier test, especially for the marginal failing case, substantial reasons of causing the skews must be provided. The reason could be either from clinical process, such as bad compliance of drug administration, or from Lab events, such as contaminated sample. The regulatory agencies like FDA are quite strict on the case of discarding data without thorough explanation and documentation.

REFERENCES


CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Don Li
Biovail Contract Research
460 Comstock Road
Toronto, ON, M1L 4S4
Canada
Work Phone: (416) 752-3636 Ext. 305
Fax: (416) 752 7610
Email: dli@biovail.com
Web: www.geocities.com/dli56/