The SAS® Calculations of Areas Under the Curve (AUC) for Multiple Metabolic Readings

Keh-Dong Shiang, Department of Biostatistics & Department of Diabetes, City of Hope National Medical Center, Duarte, CA

ABSTRACT

In biomedical studies, the computation of Area Under the Curve (AUC) is a convenient way to combine multiple readings, such as some metabolic values (glucose and insulin) or blood (serum or plasma) concentration within a specific time interval, into a single index in drug toxicology and pharmacokinetics. Thus, computation of a representative index AUC value will be an essential and interesting task in our metabolic studies and SAS programming applications.

The SAS program I present here is a flexible and compact code for parameters adaptable input and comprehensive AUC computations, which is suitable to be applied to the case of a discrete curve over a specific time interval. In the program, SAS macro, SQL and GPLOT procedures are used to calculate and present sample data. This code can be utilized to all three kinds of AUC computations, the Total AUC (no baseline), Net Incremental AUC (with baseline) and Positive Incremental AUC (with baseline). The “negative” measured values after subtracting the baseline value are taken into account and also within the scope of all possible readings. Moreover, the sample data can be measured at various time intervals (\( \Delta t \)), and the intervals do not need to be a constant during the measurements.

INTRODUCTION

As you know, the computation of AUC is just a convenient way to combine multiple readings, such as some metabolic values\(^1\) (glucose, insulin and c-peptide) within a specific time interval, into a single index. Thus, computation of a representative index AUC value will be an essential and interesting task in our metabolic studies.

Let’s start from the basic theory of calculus - integration. If the curve function \( F(X) \) is smooth (neither sharp corners nor sharp turning points) and continuous, a curve-fitting and a numerical integration method might need to be applied to compute the area under a curve. This means the definite integral of function \( F(X) \) is just the area under a curve, which can be represented as:

\[
AUC = \int_{a}^{b} F(X)\,dX
\]

where \([a, b]\) is the finite X domain. In Yeh’s SUGI paper\(^2\), three SAS macros were developed to perform the plasma concentration-time-curve AUC calculation using ‘trapezoidal rule’.

If a metabolic reading curve can not be described as a smooth and continuous function \( F(X) \), the “trapezoidal rule” will be the ‘golden rule’ to compute the area of each individual small trapezoid (see Figure 1 and SAS\(^3\) graphic code below). Then, the AUC computing method will become relatively simple: fundamental plane geometry - the total sum of those individual areas equals the total area under the curve (i.e., total AUC). The SAS macro\(^4\) I have developed is suitable for this type of “discrete” curve over a specified time interval, but can not be applied to the smooth continuous case as shown in the above equation. In our discrete time curve model, there are a
total of three methods in general use for computing AUCs termed 'Total', 'Net Incremental' and 'Positive Incremental' areas. These three methods are all covered in the macro and can be easily applied to calculate any one of the three AUCs by giving the first parameter, baseline value, in macro as 0, 1 and 2, respectively.

DATA Datafile;
LENGTH Xtime Yvalue 8;
INFILE DATALINES;
INPUT Xtime Yvalue;
FORMAT Xtime 5.1 Yvalue 6.2;
DATALINES;
-15.0 4.0
-10.0 4.5
-5.0 3.5
0.0 4.0
15.0 5.0
30.0 6.5
45.0 8.0
60.0 6.5
90.0 5.4
120.0 4.5
;
RUN;

SYMBOL1 V=NONE C=GREEN I=NEEDLE;
SYMBOL2 V=DOT C=RED I=JOIN;
PROC GPLOT DATA = Datafile;
PLOT Yvalue*Xtime Yvalue*Xtime / OVERLAY FRAME VREF=4.0 CVREF=BLUE
LVREF=34 HMINOR=0 VMINOR=0 VAXIS=0 TO 10 BY 1 HAXIS=-20 TO 140 BY 20;
TITLE 'Sample Data Plot: Metabolic Values vs. Time';
LABEL Yvalue='Y Values' Xtime='Relative Time (min)';
RUN;

Figure 1. Sample plot of some metabolic readings, Yvalues, versus time points, Xtimes.
To avoid the terminological ambiguity in this area summation, it is important to clarify the definitions of those three AUCs mentioned earlier. The ‘total’ area is the area under the reading curve down to a blood metabolic level of “zero”. The ‘net incremental’ area is the area under the curve above the baseline value. The baseline value can be the fasting level, minimum value among specified time points, or the averaged value of the negative time points. Please note that the areas below the baseline, if exist, are counted and calculated in this scenario, i.e., deducting from the total area. The ‘positive incremental’ area is similar except that areas beneath the baseline are ignored; therefore, the positive incremental area cannot be negative.

Before presenting our SAS macro for computing the AUCs, several published works and computational methods regarding OGTT and IVGTT are introduced in the following:

During the Oral Glucose Tolerance Test (OGTT), the areas under the glucose and insulin curves were computed by using the following formula\(^5,6\):

\[
\text{AUC} = 0.25 \times (\text{fasting value}) + 0.5 \times (\text{half-hour value}) + 0.75 \times (1\text{-hour value}) + 0.5 \times (2\text{-hour value})
\]

In fact, this formula can be easily proved and is just the simplified form of the summation result for a special case (i.e., observations measured and recorded at 4 time points only) by using ‘trapezoidal rule’:

\[
(V_0 + V_1) \times \frac{0.5}{2} + (V_1 + V_2) \times \frac{0.5}{2} + (V_2 + V_3) \times \frac{1.0}{2}
\]

\[
= 0.25 \times V_0 + 0.25 \times V_1 + 0.25 \times V_1 + 0.25 \times V_2 + 0.5 \times V_2 + 0.5 \times V_3
\]

\[
= 0.25 \times V_0 + 0.5 \times V_1 + 0.75 \times V_2 + 0.5 \times V_3
\]
where $V_0$, $V_1$, $V_2$ and $V_3$ are the fasting value, half-hour value, 1-hour value and 2-hour value, respectively.

During a Basal Insulin Intravenous Glucose Tolerance Test (BI-IVGTT), Cobelli et al. defined an index value, Glucose Effectiveness (GE), as

$$GE = \frac{D}{AUC}$$

where $D$ is the glucose bolus dose and AUC is the area under the curve of the blood glucose concentration above basal. Actually, this AUC should be the 'incremental' area we indicated earlier.

**SAS AUC CALCULATIONS USING TRAPEZOIDAL RULE**

In OGTT study, the glucose data (Table 3) in Psyrogiannis et al.'s paper are cited here and compared to our initial SAS AUC computation using 'trapezoidal rule'.

**SAS Initial Code Using Trapezoidal Rule**

```sas
DATA Datafile;
LENGTH Xtime Yvalue 8;
INFILE DATALINES;
INPUT Xtime Yvalue;
FORMAT Xtime 5.1 Yvalue 6.2;
DATALINES;
0.0  4.53
0.5  8.40
1.0  8.40
2.0  5.40
;
RUN;

DATA Computed;
SET Datafile;
DROP LagTime LagValue;
LagTime = LAG(Xtime);
LagValue = LAG(Yvalue);
IF Xtime = 0 THEN DO;
  LagTime = 0;
  LagValue = 0;
END;
Trapezoid = (Xtime-LagTime)*(Yvalue+LagValue)/2;
SumTrapezoid + Trapezoid;
RUN;
```

**SAS Output Result**

<table>
<thead>
<tr>
<th>Obs</th>
<th>Xtime</th>
<th>Yvalue</th>
<th>Trapezoid</th>
<th>Sum Trapezoid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>4.53</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>8.40</td>
<td>3.2325</td>
<td>3.2325</td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>8.40</td>
<td>4.2000</td>
<td>7.4325</td>
</tr>
<tr>
<td>4</td>
<td>2.0</td>
<td>5.40</td>
<td>6.9000</td>
<td>14.3325</td>
</tr>
</tbody>
</table>
SAS CODE USING THE FORMULA PRESENTED IN REFERENCES 5 AND 6

DATA Computed;
SET Datafile;
IF Xtime = 0.0 THEN V = 0.25 * Yvalue;
IF Xtime = 0.5 THEN V = 0.5 * Yvalue;
IF Xtime = 1.0 THEN V = 0.75 * Yvalue;
IF Xtime = 2.0 THEN V = 0.5 * Yvalue;
Sum + V;
RUN;

SAS Output Result

<table>
<thead>
<tr>
<th>Obs</th>
<th>Xtime</th>
<th>Yvalue</th>
<th>V</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>4.53</td>
<td>1.1325</td>
<td>1.1325</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>8.40</td>
<td>4.2000</td>
<td>5.3325</td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>8.40</td>
<td>6.3000</td>
<td>11.6325</td>
</tr>
<tr>
<td>4</td>
<td>2.0</td>
<td>5.40</td>
<td>2.7000</td>
<td>14.3325</td>
</tr>
</tbody>
</table>

The above two SAS computations yield the same glucose AUC value, 14.3325 mmol/l, which is also consistent with Psyrogiannis’s result⁶. In addition, the statements with LAG function and using trapezoidal rule shown above will be applied later in the main AUC macro.

Another related publication⁸ discussing the trapezoidal rule is also referenced here. In reference number 8, Tai ‘invented’ an area equation like this:

\[
Area = \frac{1}{2} \sum_{i=1}^{n} X_{i-1} \times (Y_{i-1} + Y_i) = \frac{1}{2} [X_0 \times (Y_0 + Y_1) + X_1 \times (Y_1 + Y_2) + ... + X_{n-1} \times (Y_{n-1} + Y_n)]
\]

The only author, MM Tai, in this paper called this equation “Tai’s formula”. Tai divided the total area under a curve into individual small segments such as squares, rectangles and triangles. These small areas can be precisely determined by existing geometric formulas. In this mathematical model, the areas of the individual segment are then added to obtain the total area under the curve. As a matter of fact, you can easily tell that “Tai’s formula” presented above is the summing application of the ‘trapezoidal rule’, since a trapezoid is simply the combination of a triangle and a rectangle.

The main purpose of the SAS program I have completed so far is to calculate the areas under the curves by using trapezoidal rule. It consists of one SAS Data step (see optional code block, this data file can be an external file), two SAS PROC SQL procedures (also optional, selecting one to evaluate baseline value), and one SAS AUC macro. The AUC macro performs the main AUC computations. The output SAS data set will list each individual summed value for all small trapezoids.

The key macro AUC includes 3 mandatory parameters, baseline, dataset and output, which are explained below:

%MACRO AUC(baseline, dataset, output);

  baseline
    Specify this parameter to define the AUC type.
    0 means 'Total AUC'
    1 means 'Net Increment AUC'
    2 means 'Positive Increment AUC'
- **dataset**
  Specify the name of the SAS input data set.

- **output**
  Specify this parameter as the name of SAS output data set.

* This is the Data step for observed values input. These data can also be read from external file, such as SQL table;

```sas
DATA DataFile;
LENGTH Xtime Yvalue 8;
INFILE DATALINES;
INPUT Xtime Yvalue;
FORMAT Xtime 5.1 Yvalue 6.2;
DATALINES;
  0.0   10.1
  30.0   13.7
  60.0   15.0
  90.0   14.0
 120.0   12.8
 150.0   11.1
 180.0   8.90
;
RUN;
```

` The following two SQL procedures are optional code blocks, choosing one of them to compute the baseline value from some early observations of the data file and store it as a macro variable, BaseY;

` Baseline value is taken as the mean of all observed values prior to the time zero.;

```sas
PROC SQL;
SELECT MEAN(Yvalue) FORMAT=6.2 LABEL='Avg Yvalue' INTO : BaseY FROM Datafile WHERE Xtime LT 0;
QUIT;
```

` Baseline value is taken as the mean of the first 3 observed values.;

```sas
PROC SQL INOBS=3;
SELECT MEAN(Yvalue) FORMAT=6.2 LABEL='Avg Yvalue' INTO : BaseY FROM Datafile;
QUIT;
```

` This is the macro that calculates the 3 AUCs.;

```sas
%MACRO AUC(baseline, dataset, output);
DATA &output;
SET &dataset (WHERE=(Xtime GE 0));
RETAIN Basevalue;
IF &baseline = 0 THEN Basevalue = 0.0;
    &BaseY shown in the following statement is the macro variable defined in any one of the above SQL procedures;
    IF (&baseline = 1 OR &baseline = 2) AND _N_ = 1 THEN Basevalue = &BaseY;
Yvalue = Yvalue - Basevalue;
DROP LagTime LagValue;
LagTime = LAG(Xtime);
LagValue = LAG(Yvalue);
IF Xtime = 0 THEN DO;
    LagTime = 0;
    LagValue = 0;
END;
```
IF &baseline = 2 AND Yvalue > 0 AND LagValue <= 0.0 THEN DO;
* Connecting line with positive slope, only the area of right triangle (above baseline) is counted.;
   DROP Ratio;
   Ratio = Yvalue / (ABS(LagValue)+Yvalue);
   Trapezoid = Ratio*(Xtime-LagTime)*(Yvalue+0.00)/2;
END;
ELSE IF &baseline = 2 AND Yvalue < 0 AND LagValue >= 0.0 THEN DO;
* Connecting line with negative slope, only the area of left triangle (above baseline) is counted.;
   DROP Ratio;
   Ratio = LagValue / (LagValue+ABS(Yvalue));
   Trapezoid = Ratio*(Xtime-LagTime)*(0.00+LagValue)/2;
END;
ELSE IF &baseline = 2 AND Yvalue < 0 AND LagValue < 0 THEN Trapezoid = 0.0;
* Negative trapezoidal area is not counted.;
ELSE Trapezoid = (Xtime-LagTime)*(Yvalue+LagValue)/2;
* The rest of all positive trapezoidal areas are counted.;
SumTrapezoid + Trapezoid;
FORMAT Trapezoid SumTrapezoid 8.3;
RUN;
%MEND AUC;

To further verify and debug this SAS code, I have cited Wolever et al’s sample data and result9. In Table 1 of this reference, blood glucose (mmol/L) data and AUCs on the GI in a diabetic subject being treated with insulin are as follows:

<table>
<thead>
<tr>
<th>Diabetic Subject</th>
<th>0 min</th>
<th>30 min</th>
<th>60 min</th>
<th>90 min</th>
<th>120 min</th>
<th>150 min</th>
<th>180 min</th>
<th>AUC Inc</th>
<th>AUC Net</th>
<th>AUC Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bread</td>
<td>10.1</td>
<td>13.7</td>
<td>15.0</td>
<td>14.0</td>
<td>12.8</td>
<td>11.1</td>
<td>8.9</td>
<td>475</td>
<td>465</td>
<td>2283</td>
</tr>
</tbody>
</table>

While calling the macro, the first baseline parameter, 0, 1 or 2, were applied separately:

%SAS Output Result for Total AUC

<table>
<thead>
<tr>
<th>Obs</th>
<th>Xtime</th>
<th>Yvalue</th>
<th>Basevalue</th>
<th>Trapezoid</th>
<th>SumTrapezoid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>10.1</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>30.0</td>
<td>13.7</td>
<td>0</td>
<td>137.000</td>
<td>137.000</td>
</tr>
<tr>
<td>3</td>
<td>60.0</td>
<td>15.0</td>
<td>0</td>
<td>345.000</td>
<td>345.000</td>
</tr>
<tr>
<td>4</td>
<td>90.0</td>
<td>14.0</td>
<td>0</td>
<td>435.000</td>
<td>435.000</td>
</tr>
<tr>
<td>5</td>
<td>120.0</td>
<td>12.8</td>
<td>0</td>
<td>402.000</td>
<td>402.000</td>
</tr>
<tr>
<td>6</td>
<td>150.0</td>
<td>11.1</td>
<td>0</td>
<td>558.500</td>
<td>558.500</td>
</tr>
<tr>
<td>7</td>
<td>180.0</td>
<td>8.9</td>
<td>0</td>
<td>300.000</td>
<td>2283.000</td>
</tr>
</tbody>
</table>

%SAS Output Result for Net Incremental AUC

<table>
<thead>
<tr>
<th>Obs</th>
<th>Xtime</th>
<th>Yvalue</th>
<th>Basevalue</th>
<th>Trapezoid</th>
<th>SumTrapezoid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.00</td>
<td>10.1</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>30.0</td>
<td>3.60</td>
<td>10.1</td>
<td>54.000</td>
<td>54.000</td>
</tr>
<tr>
<td>3</td>
<td>60.0</td>
<td>4.90</td>
<td>10.1</td>
<td>127.500</td>
<td>127.500</td>
</tr>
<tr>
<td>4</td>
<td>90.0</td>
<td>3.90</td>
<td>10.1</td>
<td>132.000</td>
<td>132.000</td>
</tr>
</tbody>
</table>

%MEND AUC;
In the same table and reference, blood glucose (mmol/L) data and AUCs on the GI in a nondiabetic subject are displayed below.

<table>
<thead>
<tr>
<th>Nondiabetic Subject</th>
<th>0 min</th>
<th>15 min</th>
<th>30 min</th>
<th>45 min</th>
<th>60 min</th>
<th>90 min</th>
<th>120 min</th>
<th>AUC Inc</th>
<th>AUC Net</th>
<th>AUC Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bread</td>
<td>4.3</td>
<td>4.8</td>
<td>6.2</td>
<td>5.5</td>
<td>5.0</td>
<td>3.9</td>
<td>4.1</td>
<td>66</td>
<td>55</td>
<td>571</td>
</tr>
</tbody>
</table>

The same SAS AUC macro is also applied, and the macro calling statements are written as:

```sas
%AUC(0, Datafile, Computed);
```

**SAS Output Result for Total AUC**

<table>
<thead>
<tr>
<th>Obs</th>
<th>Xtime</th>
<th>Yvalue</th>
<th>Basevalue</th>
<th>Trapezoid</th>
<th>Trapezoid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>4.30</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>15.0</td>
<td>4.80</td>
<td>0</td>
<td>68.250</td>
<td>68.250</td>
</tr>
<tr>
<td>3</td>
<td>30.0</td>
<td>6.20</td>
<td>0</td>
<td>82.500</td>
<td>150.750</td>
</tr>
<tr>
<td>4</td>
<td>45.0</td>
<td>5.50</td>
<td>0</td>
<td>87.750</td>
<td>238.500</td>
</tr>
<tr>
<td>5</td>
<td>60.0</td>
<td>5.00</td>
<td>0</td>
<td>78.750</td>
<td>317.250</td>
</tr>
<tr>
<td>6</td>
<td>90.0</td>
<td>3.90</td>
<td>0</td>
<td>133.500</td>
<td>450.750</td>
</tr>
<tr>
<td>7</td>
<td>120.0</td>
<td>4.10</td>
<td>0</td>
<td>120.000</td>
<td>570.750</td>
</tr>
</tbody>
</table>

```sas
%AUC(1, Datafile, Computed);
```

**SAS Output Result for Net Incremental AUC**

<table>
<thead>
<tr>
<th>Obs</th>
<th>Xtime</th>
<th>Yvalue</th>
<th>Basevalue</th>
<th>Trapezoid</th>
<th>Trapezoid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.00</td>
<td>4.3</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>15.0</td>
<td>0.50</td>
<td>4.3</td>
<td>3.750</td>
<td>3.750</td>
</tr>
<tr>
<td>3</td>
<td>30.0</td>
<td>1.90</td>
<td>4.3</td>
<td>18.000</td>
<td>21.750</td>
</tr>
<tr>
<td>4</td>
<td>45.0</td>
<td>1.20</td>
<td>4.3</td>
<td>23.250</td>
<td>45.000</td>
</tr>
<tr>
<td>5</td>
<td>60.0</td>
<td>0.70</td>
<td>4.3</td>
<td>14.250</td>
<td>59.250</td>
</tr>
<tr>
<td>6</td>
<td>90.0</td>
<td>-0.40</td>
<td>4.3</td>
<td>4.500</td>
<td>63.750</td>
</tr>
<tr>
<td>7</td>
<td>120.0</td>
<td>-0.20</td>
<td>4.3</td>
<td>-9.000</td>
<td>54.750</td>
</tr>
</tbody>
</table>

```sas
%AUC(2, Datafile, Computed);
```

**SAS Output Result for Positive Incremental AUC**

<table>
<thead>
<tr>
<th>Obs</th>
<th>Xtime</th>
<th>Yvalue</th>
<th>Basevalue</th>
<th>Trapezoid</th>
<th>Trapezoid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.00</td>
<td>4.3</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>15.0</td>
<td>0.50</td>
<td>4.3</td>
<td>3.750</td>
<td>3.750</td>
</tr>
</tbody>
</table>
To double examine the AUC SAS code, the sample data and computed result in reference 8 is also used for verification. In the table 1 of Tai's study, blood glucose data and total AUC are summarized as follows:

<table>
<thead>
<tr>
<th>Time (min)</th>
<th>0</th>
<th>30</th>
<th>60</th>
<th>90</th>
<th>120</th>
<th>Total AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glucose (mg/dL)</td>
<td>95</td>
<td>147</td>
<td>124</td>
<td>111</td>
<td>101</td>
<td>14400</td>
</tr>
</tbody>
</table>

These values are read and written as a part of SAS code:

```sas
DATALINES;
0.0 95.0
30.0 147.0
60.0 124.0
90.0 111.0
120.0 101.0;

%AUC(0, Datafile, Computed);

SAS Output Result for Total AUC

<table>
<thead>
<tr>
<th>Obs</th>
<th>Xtime</th>
<th>Yvalue</th>
<th>Basevalue</th>
<th>Trapezoid</th>
<th>Trapezoid</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>95.00</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>30.0</td>
<td>147.00</td>
<td>0</td>
<td>3630.000</td>
<td>3630.000</td>
<td>3630.000</td>
</tr>
<tr>
<td>3</td>
<td>60.0</td>
<td>124.00</td>
<td>0</td>
<td>4065.000</td>
<td>7695.000</td>
<td>7695.000</td>
</tr>
<tr>
<td>4</td>
<td>90.0</td>
<td>111.00</td>
<td>0</td>
<td>3525.000</td>
<td>11220.00</td>
<td>11220.00</td>
</tr>
<tr>
<td>5</td>
<td>120.0</td>
<td>101.00</td>
<td>0</td>
<td>3180.000</td>
<td>14400.00</td>
<td>14400.00</td>
</tr>
</tbody>
</table>
```

Again, our computed result matches with Tai's data.

**USING SAS WEIGHTED VARIABLE AND MEANS PROCEDURE**

By using a SAS built-in procedure, “PROC MEANS”, the overall weighted (weight variable: TimeDiff) averaged Y values can also be simply obtained. By using the same data set, the SAS statements and result are listed below:

**SAS Code and Result**

```sas
DATA DataFile;
LENGTH Xtime Yvalue 8;
INFILE DATALINES;
INPUT Xtime Yvalue;
FORMAT Xtime 5.1 Yvalue 6.2;
DATALINES;
0.0 95.0
30.0 147.0
60.0 124.0
90.0 111.0
120.0 101.0;
RUN;
```
DATA Computed;
SET Datafile;
DROP LagTime LagValue;
LagTime = LAG(Xtime);
LagValue = LAG(Yvalue);
IF Xtime = 0 THEN DO;
    LagTime = 0;
    LagValue = 0;
END;
TimeDiff = Xtime-LagTime;
ValueAvg = (Yvalue+LagValue)/2;
* Please note that 'trapezoidal rule' is not used here.;
RUN;

PROC MEANS DATA=Computed MEAN MAXDEC=2;
VAR ValueAvg;
WEIGHT TimeDiff;
RUN;

The MEANS Procedure
Analysis Variable: ValueAvg

         Mean

    120.00

The overall weighted mean of the blood glucose value is 120.00 mg/dL, and the total specific time interval 120 minutes is already given. Therefore, the total AUC value can be simply calculated as

AUC = Time Interval × Overall Weighted Mean = 120 × 120.00 = 14400.00

This total AUC value is consistent with the earlier macro computed result, 14400.00.

Furthermore, a discussion paragraph and a sample data table copied from an Internet published web page (Food and Agriculture Organization of the United Nations) are also cited below (TABLE 1).

INCREMENTAL AREA UNDER THE CURVE

A number of different methods have been used to calculate the area under the curve. For most glycemic index data, the area under the curve has been calculated as the incremental area under the blood glucose response curve (IAUC), ignoring the area beneath the fasting concentration. This can be calculated geometrically by applying the trapezoid rule. When a blood glucose value falls below the baseline, only the area above the fasting level is included. Sample data are shown in Table 1.

<table>
<thead>
<tr>
<th>TABLE 1</th>
<th>Sample blood glucose responses to the ingestion of 50g carbohydrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minutes</td>
<td>0</td>
</tr>
<tr>
<td>Standard #1</td>
<td>4.3</td>
</tr>
</tbody>
</table>
The same SAS AUC macro with the input data set below yields the result of SumTrapezoid=113.950 (see output result below), which matches with the above IAUC value, 114, in reference 10.

SAS Datalines

DATALINES;
0.0 4.3
15.0 6.3
30.0 7.9
45.0 5.3
60.0 4.1
90.0 4.6
120.0 4.9
;

In this case, this incremental AUC actually means “Positive Incremental AUC”, so input value ‘2’ for the first baseline parameter in SAS macro should be used.

%AUC(2, Datafile, Computed);

SAS Output Result

<table>
<thead>
<tr>
<th>Obs</th>
<th>Xtime</th>
<th>Yvalue</th>
<th>Basevalue</th>
<th>Trapezoid</th>
<th>Sum Trapezoid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.00</td>
<td>4.3</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>15.0</td>
<td>2.00</td>
<td>4.3</td>
<td>15.000</td>
<td>15.000</td>
</tr>
<tr>
<td>3</td>
<td>30.0</td>
<td>3.60</td>
<td>4.3</td>
<td>42.000</td>
<td>57.000</td>
</tr>
<tr>
<td>4</td>
<td>45.0</td>
<td>1.00</td>
<td>4.3</td>
<td>34.500</td>
<td>91.500</td>
</tr>
<tr>
<td>5</td>
<td>60.0</td>
<td>0.20</td>
<td>4.3</td>
<td>6.250</td>
<td>97.750</td>
</tr>
<tr>
<td>6</td>
<td>90.0</td>
<td>0.30</td>
<td>4.3</td>
<td>2.700</td>
<td>100.450</td>
</tr>
<tr>
<td>7</td>
<td>120.0</td>
<td>0.60</td>
<td>4.3</td>
<td>13.500</td>
<td>113.950</td>
</tr>
</tbody>
</table>

DISCUSSION

In the study of Allison et al\textsuperscript{11}, they pointed out a problem while computing AUCs that the “truncation” of negative area (i.e., ignoring the area below the baseline) is equivalent to throwing away much of the variance in any readings below the baseline. They concluded that interpretation of glucose and insulin data are subject to limitations of analysis depending upon which method is used to calculate the AUC.

In reference 12, Potteiger et al\textsuperscript{12} found that following nine months of exercise the net incremental and positive incremental methods for computing insulin AUC would result in an interpretation of no change in AUC, while employing the total AUC method would suggest an improvement for insulin action following nine months of aerobic exercise. They concluded that use of the total AUC method yields a different result from the other methods, the net incremental and positive incremental methods, and may be the preferred method of choice because it is not dependent upon an ever-changing “baseline” level for glucose and insulin.

In the paper of Wolf et al\textsuperscript{13}, they indicated that the calculation of relative glycemic response differs from glycemic index in the estimation of incremental AUC. In the calculation of net incremental AUC for relative glycemic response, the negative area was subtracted from the total area, whereas the calculation of positive incremental AUC for glycemic index ignored this area below baseline. In their case, large discrepancies were seen between the relative glycemic response and glycemic index because serum glucose concentration fell below the basal fasting concentration near 120 minutes. Discrepancies such as this have generated much debate about which method is most appropriate\textsuperscript{14,15,11}. However, Wolever et al\textsuperscript{9} suggested that the
postprandial glycemic response should not be measured after 2 hours for computing incremental AUC in normal subjects, thus avoiding the discrepancies between values.

I was thinking that the best estimate of metabolic AUC, avoiding the possible biased factors, might be obtained by computing two areas, the total area during a treatment condition and the total area during a control condition, and then subtracting the latter from the former. However, the negative values (beneath the baseline) will probably not occur and should not be the major concerns in most metabolic studies.

**SUMMARY**

The SAS program I presented here is a flexible, compact and simple code for parameter input and comprehensive AUC computations. It can be applied to all three kinds of AUC computations, the Total AUC (no baseline or baseline=0), Net Incremental AUC (with baseline) and Positive Incremental AUC (with baseline). The "negative" Y values after subtracting the baseline value are taken into account and also within the scope of all possible metabolic readings. Moreover, the sample data can be measured at different time intervals (\(\Delta t\)), and the intervals do not need to be a constant during the measurements.

**REFERENCES**


Incremental AUC, Net Incremental AUC and Total AUC - Table 1. Effect of different methods of calculating the area under the curve from the same blood glucose data on the GI in a diabetic subject being treated with insulin and in a nondiabetic subject.

Chapter 4 - The role of the glycemic index in food choice.
Incremental AUC calculation - Table 1. Sample blood glucose responses to the ingestion of 50g carbohydrate


ACKNOWLEDGMENTS

I would like to take this opportunity to acknowledge all of the assistance and support given to me by Drs. Fouad Kandeel, Jeff Longmate, David Ikle and Joyce Niland at City of Hope National Medical Center. Again, special thanks go to Dr. Longmate and Dr. Kandeel for stimulating my interest and many fruitful discussions in this study project. I also express my appreciation to my colleague, Lorraine Lesiecki, for correcting the errors during the writing of this paper.

TRADEMARKS

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION

Contact the author at:

Keh-Dong Shiang, Ph.D.
Department of Biostatistics & Department of Diabetes