**Measuring Time in Range for Multiple A1c Values**

This note shows how to calculate percent time spent in range for A1c lab tests. Suppose that the lab values for the patient are stored in the file C. At each lab reading, we also have the age of the patient in years. Date of lab values is protected data and cannot be displayed but age at lab value is not and can be used for our purposes. The lab values for patient 299 include the following readings:

| **Portion of C** |
| --- |
| **id** | **A1c** | **Age** |
| 299 | 6.3 | 24.29 |
| 299 | 6.6 | 25.09 |
| 299 | 6.3 | 25.95 |
| 299 | 10 | 27.69 |
| 299 | 9.6 | 27.96 |
| 299 | 7.4 | 28.15 |
| 299 | 8 | 28.65 |
| 299 | 7.5 | 29.47 |
| 299 | 6.8 | 30 |
| 299 | 6.8 | 30.77 |
| 299 | 7 | 31.63 |
| 299 | 8.4 | 32.15 |
| 299 | 7.6 | 32.47 |
| 299 | 8.7 | 33.04 |
| 299 | 8.1 | 33.31 |
| 299 | 7.3 | 33.55 |
| 299 | 6.6 | 34.21 |
| 299 | 9.1 | 34.83 |
| 299 | 7.6 | 35.15 |

First, we look at the pairs of consecutive A1c readings.  The following code shows how we join the table C with itself, now called C\_1, to get to consecutive pairs of lab values:

SELECT C.id, C.age AS Age1, Min(C\_1.Age) AS Age2, Avg(C.AvgOfA1C) AS A1, First(C\_1.AvgOfA1C) AS A2 INTO A1cPairs

FROM C AS C\_1 INNER JOIN C ON C\_1.id = C.id

WHERE (((C\_1.age)>[c].[age]))

GROUP BY C.id, C.age

ORDER BY C.id, C.age, Min(C\_1.Age);

 The result is displayed into the table A1cPairs. The first two pairs of readings are in range.  The third pair of reading is in range for a portion of the time, and we need to calculate what portion of time it is in range.

| **Portion of A1cPairs** |
| --- |
| **id** | **Age1** | **Age2** | **A1** | **A2** |
| 299 | 24.29 | 25.09 | 6.3 | 6.6 |
| 299 | 25.09 | 25.95 | 6.6 | 6.3 |
| 299 | 25.95 | 27.69 | 6.3 | 10 |
| 299 | 27.69 | 27.96 | 10 | 9.6 |
| 299 | 27.96 | 28.15 | 9.6 | 7.4 |
| 299 | 28.15 | 28.65 | 7.4 | 8 |
| 299 | 28.65 | 29.47 | 8 | 7.5 |
| 299 | 29.47 | 30 | 7.5 | 6.8 |
| 299 | 30 | 30.77 | 6.8 | 6.8 |
| 299 | 30.77 | 31.63 | 6.8 | 7 |
| 299 | 31.63 | 32.15 | 7 | 8.4 |
| 299 | 32.15 | 32.47 | 8.4 | 7.6 |
| 299 | 32.47 | 33.04 | 7.6 | 8.7 |
| 299 | 33.04 | 33.31 | 8.7 | 8.1 |
| 299 | 33.31 | 33.55 | 8.1 | 7.3 |
| 299 | 33.55 | 34.21 | 7.3 | 6.6 |
| 299 | 34.21 | 34.83 | 6.6 | 9.1 |
| 299 | 34.83 | 35.15 | 9.1 | 7.6 |

A1c is assumed to be in control for under 80-year-old patients if it is in the range 5 to 6.9. For over 80-year-old, it is in control if it is in range 5 to 8.5. In this step, we calculate at what age the values of 5 and 6.9/8.5 occur for each pair of consecutive A1c reading.  This is done by fitting a line to the two points defined by age and A1c lab values.  Then, this line is extrapolated to see the age at 5 or at 6.9/8.5. The code for doing so is provided below:

SELECT A1cPairs.id

, A1cPairs.Age1

, A1cPairs.Age2

, A1cPairs.A1

, A1cPairs.A2

, IIf([A1]=[a2],IIf([Age1]<[Age2],[Age1],[Age2]),[Age1]+(([Age2]-[Age1])/([A2]-[A1]))\*(5-[A1])) AS AgeAt5, IIf([a1]=[a2],IIf([age1]<[Age2],[Age2],[Age1]),[Age1]+(([Age2]-[Age1])/([A2]-[A1]))\*(IIf([Age1]>79.99,8.5,6.9)-[A1])) AS AgeAtUCL

, IIf([Age1]<[Age2],[Age1],[Age2]) AS LowerAge

, IIf([age1]<[Age2],[Age2],[Age1]) AS HigherAge

, IIf([a1]<[A2],[A2],[A1]) AS HigherA, IIf([a1]<[A2],[A1],[A2]) AS LowerA

FROM A1cPairs

ORDER BY A1cPairs.id, A1cPairs.Age1, A1cPairs.Age2;

The lower age is the minimum of the age at the two readings. Higherage is the maximum of age at the two readings. The result of this analysis is stored in the Table 4AgeAtUpperLower.

| **4AgeAtUpperLower** |
| --- |
| **id** | **Age1** | **Age2** | **A1** | **A2** | **AgeAt5** | **AgeAtUCL** |
| 299 | 24.29 | 25.09 | 6.3 | 6.6 | 20.82 | 25.89 |
| 299 | 25.09 | 25.95 | 6.6 | 6.3 | 29.69 | 24.22 |
| 299 | 25.95 | 27.69 | 6.3 | 10 | 25.34 | 26.23 |
| 299 | 27.69 | 27.96 | 10 | 9.6 | 31.05 | 29.77 |
| 299 | 27.96 | 28.15 | 9.6 | 7.4 | 28.35 | 28.19 |
| 299 | 28.15 | 28.65 | 7.4 | 8 | 26.15 | 27.73 |
| 299 | 28.65 | 29.47 | 8 | 7.5 | 33.56 | 30.45 |
| 299 | 29.47 | 30 | 7.5 | 6.8 | 31.35 | 29.92 |
| 299 | 30 | 30.77 | 6.8 | 6.8 | -1.57+15 | >99 |
| 299 | 30.77 | 31.63 | 6.8 | 7 | 23.05 | 31.20 |
| 299 | 31.63 | 32.15 | 7 | 8.4 | 30.89 | 31.59 |
| 299 | 32.15 | 32.47 | 8.4 | 7.6 | 33.53 | 32.76 |
| 299 | 32.47 | 33.04 | 7.6 | 8.7 | 31.12 | 32.11 |
| 299 | 33.04 | 33.31 | 8.7 | 8.1 | 34.68 | 33.84 |
| 299 | 33.31 | 33.55 | 8.1 | 7.3 | 34.24 | 33.67 |
| 299 | 33.55 | 34.21 | 7.3 | 6.6 | 35.73 | 33.93 |
| 299 | 34.21 | 34.83 | 6.6 | 9.1 | 33.82 | 34.29 |
| 299 | 34.83 | 35.15 | 9.1 | 7.6 | 35.72 | 35.31 |

Next, we look at the number of years the two readings are in range.  If the minimum reading is above 6.9/8.5 then 0 days in range, if the maximum reading is below 5, then 0 years in range, otherwise if either the minimum or the maximum reading is within the range, then years in range is calculated from differences of age at upper and lower limits.  The upper limit is set based on minimum of (a) age of  maximum of the two readings and (b) age of 6.9/8.5.  The lower limit is set based on maximum of (a) age of minimum of the two readings and (b) age at 5.  Here is the code for doing so:

SELECT [5AgeAtUpperLower].id

, IIf([higherA]>IIf([age1]>79.99,8.5,6.9)

,[AgeAtUCL],[HigherAge]) AS [Upper]

, IIf([LowerA]<5,[AgeAt5],[LowerAge]) AS [Lower]

,IIf([LowerA]>=IIf([age1]>79.99,8.5,6.9),0,IIf([HigherA]<5,0,IIf([higherA]>IIf([age1]>79.99,8.5,6.9),[AgeAtUCL],[HigherAge])-IIf([LowerA]<5,[AgeAt5],[LowerAge]))) AS InRange

, [5AgeAtUpperLower].A1

, [5AgeAtUpperLower].A2

, [5AgeAtUpperLower].LowerA

, [5AgeAtUpperLower].HigherA

, [5AgeAtUpperLower].LowerAge

, [5AgeAtUpperLower].HigherAge

, [5AgeAtUpperLower].AgeAt5

, [5AgeAtUpperLower].AgeAtUCL

FROM 5AgeAtUpperLower;

The results are stored in 6InRangePairs Table and shown below. The third record is of interest as the lower and upper A1c values range from 6.3 to 10.  Only 0.28 years is in range and the remainder of the difference in age at these tests is out of range.

| **Portion of 6InRangePairs** |
| --- |
| **id** | **InRange** | **A1** | **A2** | **LowerA** | **AgeAt5** | **AgeAtUCL** | **LowerAge** | **HigherAge** |
| 299 | 0.80 | 6.3 | 6.6 | 6.3 | 20.82 | 25.89 | 24.29 | 25.09 |
| 299 | 0.86 | 6.6 | 6.3 | 6.3 | 29.69 | 24.22 | 25.09 | 25.95 |
| 299 | 0.28 | 6.3 | 10 | 6.3 | 25.34 | 26.23 | 25.95 | 27.69 |
| 299 | 0 | 10 | 9.6 | 9.6 | 31.05 | 29.77 | 27.69 | 27.96 |
| 299 | 0 | 9.6 | 7.4 | 7.4 | 28.35 | 28.19 | 27.96 | 28.15 |
| 299 | 0 | 7.4 | 8 | 7.4 | 26.15 | 27.73 | 28.15 | 28.65 |
| 299 | 0 | 8 | 7.5 | 7.5 | 33.56 | 30.45 | 28.65 | 29.47 |
| 299 | 0.45 | 7.5 | 6.8 | 6.8 | 31.35 | 29.92 | 29.47 | 30 |
| 299 | 0.77 | 6.8 | 6.8 | 6.8 | -1.57+15 | 99.2 | 30 | 30.77 |
| 299 | 0.42 | 6.8 | 7 | 6.8 | 23.05 | 31.20 | 30.77 | 31.63 |
| 299 | 0 | 7 | 8.4 | 7 | 30.89 | 31.59 | 31.63 | 32.15 |
| 299 | 0 | 8.4 | 7.6 | 7.6 | 33.53 | 32.76 | 32.15 | 32.47 |
| 299 | 0 | 7.6 | 8.7 | 7.6 | 31.12 | 32.11 | 32.47 | 33.04 |
| 299 | 0 | 8.7 | 8.1 | 8.1 | 34.68 | 33.84 | 33.04 | 33.31 |
| 299 | 0 | 8.1 | 7.3 | 7.3 | 34.24 | 33.67 | 33.31 | 33.55 |
| 299 | 0.37 | 7.3 | 6.6 | 6.6 | 35.73 | 33.93 | 33.54 | 34.21 |
| 299 | 7.36E-02 | 6.6 | 9.1 | 6.6 | 33.82 | 34.29 | 34.21 | 34.83 |
| 299 | 0 | 9.1 | 7.6 | 7.6 | 35.72 | 35.31 | 34.83 | 35.15 |

In the final step, we sum the percent of time the patient is in range across each pair of readings. The number of years followed is calculated from the difference of maximum age to minimum age, provided from the file 3MinDate. The code for calculating days in range is as follows:

SELECT [6InRangePairs].id

, Sum([Inrange]\*365) AS DaysInRange

, Avg(([MaxOfAge]-[MinAge])\*365) AS DaysFollowed

, Sum([Inrange]/([maxofage]-[MinAge])) AS PercentInRange

INTO DaysInRange

FROM 3MinDate INNER JOIN 6InRangePairs ON [3MinDate].id = [6InRangePairs].id

GROUP BY [6InRangePairs].id;

For patient 299, we calculated the total time in range to be 1480 days, days out of range to be 3967, which indicates that 37% of days the patient’s A1c values were in range.

| **Portion of Days In Range** |
| --- |
| **cc** | **Days In Range** | **Days Followed** | **Percent In Range** |
| 299 | 1480.22 | 3967.00 | 0.37 |