Paper CT-12

Hidden Biases Using SAS Dates

Steve James, CDC, Atlanta, GA

ABSTRACT

Long-time SAS users are well familiar with the fact that SAS stores dates as the number of days from Jan 1, 1960. However what they may not realize is that there are some hidden biases that can be introduced as a result of using SAS dates. It's not a problem with SAS dates themselves, but how they might be used that's an issue. For instance, how you identify the time interval between two dates expressed in months or years can introduce a bias. This paper discusses several types of biases and other errors that can be introduced when using SAS dates.

INTRODUCTION

The CDC is developing a system to analyze immunization data of vaccinations provided to children age 0-18 years. Whether a shot is valid is a key characteristic the CDC is concerned about and the age the child was when the vaccine was administered is critical in identifying that. In order for the shot to be valid, the child must be at or above the minimum age for that particular shot. We have the date-of-birth for the child, as well as the date the vaccine was administered so we can calculate the age at vaccination. We also need to know the specific guideline for the particular vaccine. The guidelines specifying the minimum age that a child can receive a given vaccine are set by The Advisory Committee on Immunization Practices (ACIP). For example, the minimum age for the Influenza (Flu) vaccine is 6 months. The question is how to ensure that we apply the guidelines for dose validation properly.

INITIAL THOUGHTS ABOUT DATES

While thinking about the requirements of the system we came up with some interesting questions. One was "How many days are there in a year?" Before you scoff consider that if a child is born on Jan 1 2011 we don't consider that child to be 1 year old until Jan 1, 2012. The thought was that the 365th day is the day before the child's birthday so technically she doesn't turn 1 until the 366th day. However if you subtract SAS dates of 1/1/2011 from 1/1/2012 you get 365 so that turned out not to be a problem. There were similar concerns if the unit of measure was months. However these concerns were largely ignored with the general statement that "SAS knows how to handle dates" and discussion went to the next topic. That was an example of our first bias: we don't have to worry about dates because "SAS knows how to handle dates." Yes, SAS does know how to handle dates but do we?

CANADIAN HOCKEY AND BIRTHDAY BIAS

Malcolm Gladwell wrote in *Outliers* about how the elite junior Canadian hockey teams have a disproportionate number of players who were born in the first half of the year. Most of them were born in the months from January to June and significantly fewer were born from July to December, particularly the last quarter.

The reason that hockey players in these leagues these teams tend to be born in the first half of the year is based on the way youth hockey leagues pick their players. Most youth hockey leagues require a child to be a certain age by January 1st. However having a cutoff date like that provides children born just after that date a distinct advantage when it comes to performance relative to their peers.

Imagine that you're born on January 2nd and you're 9 years old. You have to be 10 years-old on January 1 in order to play in the league but you'll only be 9 years and 364 days and you won't be able to join. However the next year when you're able to join the league, no one will be older than you on the team. You'll be more developed physically and emotionally than all of your peers, and at that age the difference is significant. In fact, for the kid who was born on Jan 1 who thought he was so lucky to be able to play, you'll be a full year older than him/her. You'll turn 11 on Jan 2 in essence before the season even starts.

Children born just after the cutoff date are older than their peers at a time when the difference in age is quite significant. As a result, on average, they tend to outperform their peers and are perceived as better hockey players. They get invited to all-star teams. They get extra playing time and more specialized coaching which improves their performance even more and the gap between them and others increases. When and if that initial advantage is lost is unclear, but the fact that it exists at all is illuminating.

BACK TO VACCINATION DATA

In the design phase of the system the business analyst (who was unfamiliar with SAS) wrote down the necessary minimum ages as a certain number of days. One case that stood out was when she converted the age of 6 months to 180 days. Why was that number chosen? Was it the right number? Researching the precise answer to that allowed the discovery of a second bias with SAS dates. It turns out that the number of days in 6 months varies based on which 6-month period you're talking about. Since the number of days varies each month one collection of months might well have a different number of days from another, and in fact they do. Jan 1, 2012 to July 1, 2012 has 182 days while March 1, 2012 to Sept 1, 2012 has 184. November 1, 2012 to May 1, 2012 has only 181 days and June 1, 2012 to December 1, 2012 has 183 days. That gave us a variation of up to 3 days just based on the particular dates you were born. This would mean that for vaccinations that forced a child to be 6 months old we'd be say a child born in January would have a valid vaccine after 181 days when in fact it should be 182. We'd bias the results simply based on the calendar date the child were born. And it's safe to say that the particular date a child was born should not affect the likelihood a vaccination is valid.

Using the awareness of hidden biases from Canadian hockey teams it became clear that by converting the number of years or months into the number of days would lead to incorrect results. Some children would be more likely while other children would be less likely to have a valid vaccination for no other reason than the calendar date when they had been born. The scary thing is that this is something that we wouldn't have been likely to test. We would have tested a minimum age of 6 months certainly, but not tested enough cases with different birth dates to highlight the bias.

This bias would occur with any interval that was defined by either months or years, since those units of time vary. However the guidelines that we were using defined the time intervals using those imprecise measures. What could we do?

SOLUTION

The problem wasn't that SAS didn't handle the dates properly or that there was an inherent bias in SAS dates, but the ACIP guidelines where written in a way that made using SAS dates a bit harder. The guidelines specified intervals in months and years and as we've seen those intervals vary in length based on where they fall in the calendar. We couldn't change the guidelines, we had to change the way we applied them.

For example, the ACIP Guidelines specify a minimum age of 6 months for the Influenza (Flu) Vaccine. You could use a brute force method by calculating the age of the patient at vaccination in years, months, weeks and days and have code like the following:

```
MinAgeUnit='months' ;
MinAgeValue = 6 ;

If MinAgeUnit = 'years' then do ;
    If AgeinYears < MinAgeValue then valid = 'no' ;
    Else valid = 'yes' ;
    End ;

Else If MinAgeUnit = 'months' then do ;
    If AgeinMonths < MinAgeValue then valid = 'no' ;
    Else valid = 'yes' ;
    End ;

Else If MinAgeUnit = 'weeks' then do ;
    If AgeinWeeks < MinAgeValue then valid = 'no' ;
    Else valid = 'yes' ;
    End ;

Else If MinAgeUnit = 'days' then do ;
    If AgeinDays < MinAgeValue then valid = 'no' ;
    Else valid = 'yes' ;
    End ;
```

While this approach works fine and is clear exactly what the code is trying to do, we chose instead to calculate the date when the minimum age is reached and compare that date with the date of vaccination

```
MinAgeUnit='months' ;
MinAgeValue = 6 ;
MinAgeDate = INTNX(MinAgeUnit, Dob,MinAgeValue,'same') ;
If VaxDate < MinAgeDate then valid='no' ;
    Else valid='yes' ;
```

Not only is this code more concise but it could be more easily modified to accommodate the 4-day grace period that ACIP guidelines requires (meaning if a child is 4 days away from the cutoff we count them as having reached the minimum age). It would have been far more difficult using that with a brute force approach.

THE "LEAP DAY" PROBLEM

Another issue that came up later was the question we called the leap day problem. When would a child born on a leap day (Feb 29th) turn a year old? Was it Feb 28th or March 1st?

It turns out to be a more common problem than just leap days. For example, when is a child who was born on January 31st one month old? There is no February 31st. Would it be February 28th or March 1st? For our application we determined that it was a better business decision to choose the start of the next month rather than the end of the month for determining vaccine validity. However using the INTNXC function as above returns the date at the end of the month, which is not what we wanted. We needed to modify the calculation of the minimum age date (MinAgeDate) as follows:

The following code can be used to accommodate the "leap day" problem.

```
MinAgeDate = INTNX(MinAgeUnit, Dob,MinAgeValue,'same') ;
If MinAgeUnit in ('months','year') and day(MinAgeDate) < day(DOB)
then MinAgeDate = MinAgeDate + 1 ;
```

ERRORS WHEN CALCULATING AGE IN YEARS

Because of the way SAS stores dates, you have to be careful about how you calculate a person's age in years. Some formula work in certain conditions and not in other conditions. Depending on how you define age you could introduce a bias. A common way to calculate age is the following:

```
Age = INT((VaxDate – DOB) / 365.25)) ;
```

This works most of the time but is off by one when Date and DOB have the same month and day (e.g. 1/1/11 and 1/1/12). It works better if you use 365 rather than 365.25 because for most dates the span is 365 days and you divide it by 365.25 you get 0.9993 which is truncated to 0. However using 365 will give results that are off by one on consecutive dates involving a leap year (e.g. 01/01/2012 – 12/31/2012).

The long-accepted and most accurate formula for calculating age in years as of a certain date is the following:

```
Age = INT((INTCK('month', DOB, VaxDate) - (DAY(DOB)>DAY(VaxDate)))/12) ;
```

This calculates the number of months between two dates using the INTCK function. You need to add the logical condition (Day(DOB)>Day(VaxDate)) which subtracts 1 from the total when the day part of the DOB is earlier in the month than the day part of the target date. That corrects the condition when, for example, the target date is 1/15/12 but the DOB is 1/31/11. The INTCK function would return 12 in that condition though the correct answer for calculating age is 11. Once you have the correct number of months you can divide by 12 and take the integer value and end up with the correct number of years.

Starting with SAS 9.3 YRDIF and INTCK functions have new options that allow you to accurately calculate age in years. Specifics about these functions are available in SAS documentation, but examples of their use are:

```
Age = YRDIF(DOB,VaxDate,'AGE') ;
Age = INTCK('YEAR',DOB,VaxDate','C') ;
```

The only difference between these two functions and the previous calculation is the way that it handles DOB on a leap day. The standard INTCK calculation says that a child born on Feb 29th turns 1 on Feb 28th while the SAS 9.3 functions don't indicate the child is 1 year old until March 1st. There appears to be no standard convention for when Leap Day babies should celebrate their birthdays.

CONCLUSION

It's easy to do, but it's presumptuous to think that just because SAS can store dates precisely that you won't have problems with dates. SAS date architecture alone does not solve all of your problems. You can unwittingly introduce subtle biases (aka errors) in how you use SAS dates when dealing with months and years. These problems can easily be remedied, but require some awareness and forethought.

REFERENCES

- Gladwell, Malcolm, 2008, Outliers, New York, New York, Little Brown and Company
- Gilsen, Bruce, Improve Your Dating: The INTNX Function Alignment Value SAMEDAY, 2006, Available at http://www2.sas.com/proceedings/sugi31/027-31.pdf
- SAS Institute, Sample 24808: Calculating Age with Only One Line of Code, Available at http://support.sas.com/kb/24/808.html
- SAS Institute, YRDIF Function, Available at http://support.sas.com/documentation/cdl/en/lefunctionsref/63354/HTML/default/viewer.htm#p1pmmr2dtec32an1vbsqmm3abil5.htm,
- SAS Institute, INTCK Function, Available at http://support.sas.com/documentation/cdl/en/syntaxidx/64656/HTML/default/index.htm#/documentation/cdl/en/lefunctionsref/63354/HTML/default/p1md4mx2crzfaqn14va8kt7qvfhr.htm
- Zdeb, Michael, How to calculate age for the certain date?, 1/12/2012, SAS-L Post

CONTACT INFORMATION

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

Name: Steve James
Enterprise: Centers for Disease Control and Prevention
Address: 1600 Clifton Road NE, MS-A27
City, State ZIP: Atlanta, GA 30333
Work Phone: 404-639-6041
E-mail: spj1@cdc.gov