

Calculating Time Intervals Using RETAIN

Denise Gaskins, Regenstrief Institute, Indianapolis, IN

ABSTRACT

When I first began learning how to use the SAS® system there was mention that one could carry data from one record to the next using RETAIN but no one was sure how to do this. I was told that it was hard to get it to work. Being a novice I took their word for it. Eventually, through collaboration, an adaptable algorithm using RETAIN was put together. This paper demonstrates how to use the RETAIN statement to compute an interval of time. It also references an undocumented way to use SAS time informats.

INTRODUCTION

The sections of program within this text were written to calculate the amount of time between return visits to the Emergency Room (ER) per patient (RTURN_HR) and specifically to identify the number of times patients returned within 72 hours after they were released.

The RETAIN statement causes a variable to retain its value from one iteration of the DATA step to the next. RETAIN is useful when calculating these time intervals between visits because each unique visit is in different records within the same data set.

The procedures utilized within this example can be adapted to similar situations.

CREATING THE FIRST DATA SET

The first DATA step reads the raw data into a temporary data set to be used later. The data also need to be sorted by the fields that will be compared when making calculations in subsequent DATA steps.

The SAS System stores time data in seconds. There are 86400 seconds within a 24-hour day. When time data are reported as AM or PM, any regular SAS time informat will convert the data into relevant seconds based on the AM/PM value. Our data source formats time data as HH:MM with an AM or PM suffix (ex. 11:23 PM). The TIME8. informat can accurately read time data formatted this way.

Reading the raw data into a temporary data set:

```
DATA ER;
  INFILE 'D:\ER_REMIND\ER_VISIT.DAT' MISOVER;
  LENGTH HOSPNO $9 CLINIC $15;
  FORMAT APT_DATE DATEOUT MMDDYY8.;
  FORMAT TIME TIMEOUT TIMEAMP8.;
  INPUT
    @1 APT_DATE DATE9.
  / @1 TIME TIME8.
  / @1 HOSPNO $
  / @1 CLINIC & $
  / @1 DATEOUT DATE9.
  / @1 TIMEOUT TIME8.
  ;
  DIS_DIFF=DATEOUT-APT_DATE;
  LABEL
    APT_DATE='DATE OF THE VISIT'
    TIME='TIME THE PATIENT ARRIVED'
```

```
HOSPNO='UNIQUE PATIENT IDENTIFIER'
CLINIC='CLINIC PATIENT VISITS'
DATEOUT='DATE OF DISCHARGE FROM THE VISIT'
TIMEOUT='TIME OF DISCHARGE FROM THE VISIT'
DIS_DIFF='INDICATES NUMB DAYS PT AT CLINIC'
;
RUN;

PROC SORT DATA=ER;
  BY HOSPNO APT_DATE DATEOUT TIMEOUT;
RUN;
```

The field DIS_DIFF was created to help clean the data. If a patient remains in the ER more than two consecutive dates, according to the data in one record, then that record is not included in subsequent data sets.

CREATING THE TIME INTERVAL VARIABLE

The fields referenced by the RETAIN statement are created on the very first iteration of the DATA step within the ELSE DO section of the code shown below. It can be seen in the example that new fields are created and their values are assigned the values of the unique patient identifier and the date and time of the discharge from the visit, respectively. These created fields' values are reassigned with each DATA step iteration.

Example:

```
DATA INTRVL;SET ER;
WHERE DATEOUT NE. AND DIS_DIFF < 2;
RETAIN OLDHOSP DISPDATE DISPTIME;
IF HOSPNO=OLDHOSP AND (APT_DATE NE DISDATE)
THEN DO;
  DAY_HR=(((APT_DATE-DISDATE) - 1) * 24);
  PARTIALS=((86400-DISPTIME) + TIME)/3600;
RTURN_HR=DAY_HR+PARTIALS;
  DISPDATE=DATEOUT;
  DISPTIME=TIMEOUT;
END;
ELSE IF HOSPNO=OLDHOSP AND
(APT_DATE=DISPDATE) THEN DO;
  RTURN_HR=((TIME - DISPTIME)/3600)
  DISPDATE=DATEOUT;
  DISPTIME=TIMEOUT;
END;
ELSE DO;
  OLDHOSP=HOSPNO;
  DISPDATE=DATEOUT;
  DISPTIME=TIMEOUT;
END;
RUN;
```

There are three possible scenarios for our data: 1) the patient returns on a different day, 2) the patient returns on the same day, and 3) the patient does not return within our time window of interest.

If the patient returns on a different day:

```
IF HOSPNO=OLDHOSP AND (APT_DATE NE DISDATE)
THEN DO;
  DAY_HR=(((APT_DATE-DISDATE) - 1) * 24);
  PARTIALS=((86400-DISPTIME) + TIME)/3600;
```

```

RTURN_HR=DAY_HR+PARTIALS;
  DISPDATE=DATEOUT;
  DISPTIME=TIMEOUT;
END;

```

Remember, these data are sorted by hospno, apt_date, dateout, and timeout which are the fields utilized to compare information between records. When doing our calculations we also need to remember that the SAS system stores time data in seconds.

The number of days between visits is converted into hours (DAY_HR). We subtract 1 from the difference of the two dates because we only want the 24-hour days to be included into the calculation of this field's value. We will calculate the partial days time values with the next statement.

The amount of time until the end of the day from the time of discharge of the first visit is added to the amount of time from the beginning of the day till the time of the return visit for the patient within the current record's data (PARTIALS). There are 3600 seconds in one hour. By dividing the time difference by 3600 the field's value reflects the time difference in hours.

The field RTURN_HR contains the sum of the values of these two fields and is the total time, in hours, between the two visits.

The fields referenced in the RETAIN statement are then reassigned appropriate values. The OLDHOSP field does not need to be reassigned a value because it equals the current record's value already.

If the patient returns on the same day:

```

ELSE IF HOSPNO=OLDHOSP AND
(APT_DATE=DISPDATE) THEN DO;
  RTURN_HR=((TIME - DISPTIME)/3600)
  DISPDATE=DATEOUT;
  DISPTIME=TIMEOUT;
END;

```

RTURN_HR contains the amount of time between the time of discharge from the first visit till the time the patient returned for the second visit the same day, in hours. DISPDATE and DISPTIME are reassigned appropriate values to carry into the next record. The same situation for OLDHOSP exists for this section as was mentioned before. The OLDHOSP field does not need to be reassigned a value because it equals the current record's value already.

If the patient does not return in the time window of interest:

```

ELSE DO;
  OLDHOSP=HOSPNO;
  DISPDATE=DATEOUT;
  DISPTIME=TIMEOUT;
END;

```

The fields used in the RETAIN statement are reassigned appropriate values to carry into the next record and RTURN_HR for this record will be *null*.

CONCLUSION

We now have the field RTURN_HR that has the time interval data of interest and can use it for further data manipulations and computations as needed. For our purposes, I created another field based on the value of RTURN_HR and generated frequencies on the created field to show how many times patients returned to the ER within 24-hour increments (ex. 24 hrs, 48 hrs, 72 hrs, ect).

Within the body of this paper you have been shown a way to calculate time spans between consecutive events using the RETAIN statement. RETAIN can be used for whatever reason you would need to carry the same value of a field from one record into the next.

ACKNOWLEDGMENTS

The author would like to thank Bryan Aguirre, Joe Kesterson, and Faye Smith for their helpful comments on earlier drafts of this paper.

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.

CONTACT INFORMATION

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

Denise Gaskins
 Regenstrief Institute
 1050 Wishard Boulevard
 Indianapolis, IN 46202
 Work Phone: (317)630-7690
 Fax: (317)630-6611
 Email: gaskins_d@regenstrief.iupui.edu