

Find Daylight Saving Time and Forget It

Chao-Ying Hsieh, Southern Company Services, Inc., Atlanta, GA

ABSTRACT

Finding daylight saving time is a common task for manipulating time series data. The date of daylight saving time changes every year. If SAS® programmers depend on manually entering the value of daylight saving time in their program; the maintenance of the program becomes tedious.

Using SAS function can easily find daylight saving time. This paper discusses several ways to capture and utilize daylight saving time.

INTRODUCTION

Analysts at Southern Company analyze a variety of electricity data to understand their customers' electricity consumption. Often times, analysts will need to identify the Daylight Saving Time for a specific calculation. Daylight saving time (DST) is in effect from the second Sunday of March at 2 a.m. (local time) to the first Sunday of November at 2 a.m. (local time). It does not fall on a fixed calendar date.

The easiest way to code is for the user to input the date range for DST manually. This makes the programs extremely difficult to maintain. If the inexperienced user forgets to input the correct date range, the calculation results will be incorrect. The best solution is to use SAS program to identify date range of the DST without entering the dates manually.

DYNAMICALLY FINDING DAYLIGHT SAVING TIME

Function NWKDOM returns the date for the nth occurrence of a weekday for the specified month and year. Currently, daylight saving time in the United States begins at 2:00 a.m. on the second Sunday of March and ends at 2:00 a.m. on the first Sunday of November.

The following syntax shows how to use the function NWKDOM to find the daylight saving time is in the year 2017:

```
Year=2017;           ← Set variable year as 2017
dst_beg= nwkdom(2, 1, 3, year);  ← DST begin date
dst_end= nwkdom(1, 1,11, year);  ← DST end date
```

STORE DAYLIGHT SAVING TIME IN DATA

The above code only works for a specific year. To make the program capable of identifying any given year, we can create a data set with a wide range of years:

```
data a;
format dst_beg dst_end mmddyy10.;
do year=2000 to 2100;
    dst_beg= nwkdom(2, 1, 3, year);
    dst_end= nwkdom(1, 1,11, year);
output;
end;
run;
```

Table 1 shows the first five records of data A. This table can serve as a look-up table to inner join with the time series data.

DST_BEG	DST_END	YEAR
03/12/2000	11/05/2000	2000
03/11/2001	11/04/2001	2001
03/10/2002	11/03/2002	2002
03/09/2003	11/02/2003	2003
03/14/2004	11/07/2004	2004

Table 1. The First Five Records of Data A

STORE DAYLIGHT SAVING TIME IN MACRO

Although programmers can use the data merge or SQL join with the time series data by date, there are other ways to identify DST without using the step of data join. One of them is to store the begin and end date of DST into macro variables. The following coding creates two macro variables, MAR_DST and NOV_DST:

```
proc sql noprint;
  select dst_beg format=best16. into: mar_dst
  separated by ','
  from a;
  select dst_end format=best16. into: nov_dst
  separated by ','
  from a;
quit;
```

I can use these two macro variables in the data step or a SQL query when I need to manipulate data for DST.

Following is an example that I use to calculate the mean of KW on the end date of DST:

```
proc sql;
  create table novavg as
  select meter_number, date, hour, mean(kw) as kw
  from meter_reading
  where date in (&nov_dst)           ← use macro variable NOV_DST
  group by meter_number, date, hour
  order by meter_number, date, hour;
quit;
```

STORE DAYLIGHT SAVING TIME IN FORMAT

Besides using the macro variables to capture begin and end dates of DST, using SAS format is also a good option to bypass the data join.

The following code creates a data set called FMT. Variable START is the start date of DST in March. Variable END is the end date of DST in November. The value of variable LABEL is '*'. The last record is the value that does not fall in the range of START and END. By using PROC FORMAT CNTLIN option, format DST is created.

```
data fmt(drop=year);
  attrib hlo length=$1
  start end format=mmddy10.;
  fmtname="dst";
  type="N";
do year=2000 to 2100;
  start= nwkdom(2, 1, 3, year);
  end= nwkdom(1, 1, 11, year);
```

```

        label='*';
        output;
    end;

    start=.;end=.;
    hlo="O";
    label='';
    output;
run;
proc format cntlin=fmt;run;

```

Table 2 shows the last five observations of data FMT. The last record has an 'O' for the variable HLO. This is for the date value not in the range of 'START' and 'END.'

HLO	START	END	FMTNAME	TYPE	LABEL
	03/10/2097	11/03/2097	Dst	N	*
	03/09/2098	11/02/2098	Dst	N	*
	03/08/2099	11/01/2099	dst	N	*
	03/14/2100	11/07/2100	dst	N	*
O			dst	N	

Table 2. The Last Five Records of Data FMT

If I want to have the time series data when DST is in effect, I can use a PUT statement to identify which records are within the date range of DST:

```

data dst_reading;
set meter_reading;
where put(date,dst.)='*';
run;

```

CONCLUSION

Finding the Daylight Saving Time used to be a chore. Thanks to many functions that SAS provides, programmers can now combine different techniques to make their programs more robust. Even though it will take more time to plan a well written code instead of entering the values manually, the end result is that a new program is created that can be easily maintained.

REFERENCES

Rick Wicklin, "Spring forward, fall back: Using SAS to compute the onset of daylight saving time" The DO Loop, Statistical programming in SAS with an emphasis on SAS/IML programs. Available at

<http://blogs.sas.com/content/iml/2012/03/08/using-sas-to-compute-the-onset-of-daylight-saving-time.html>

SAS® 9.4 Language Reference: Concepts, Fourth Edition

<https://support.sas.com/documentation/cdl/en/ircon/68089/HTML/default/viewer.htm#titlepage.htm>

ACKNOWLEDGMENTS

A special thanks to Robert Tsai and Celia Wang for their thoughtful review of this manuscript.

CONTACT INFORMATION

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

Chao-Ying Hsieh
Southern Company Services, Inc

chsieh@southernco.com

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 trademarks of their respective companies.