ABSTRACT
As programmers, we are frequently asked to report by periods that do not necessarily correspond to weeks on the calendar – for example, a client may want to see sales by week since the day a product was released. This paper presents simple, easy-to-use, self-contained SAS code that utilizes date functions, macro variables, a DO/UNTIL loop, and the FORMAT procedure to create formatted values for any weekly periods needed, based on dates specified by the user. It also presents a second example of code that can be used with different start dates within the same data set.

INTRODUCTION
No matter the industry, SAS programmers generally do a lot of programming with dates. SAS has an extremely robust set of date formats and functions that can be used in many cases, but at times we need to be a bit clever to set up groupings that don’t fit a standard week (or other period of time). The two SAS programs presented in this paper can be used to bucket your dates as needed, in order to quantify into groupings based on a specific starting date or dates.

CODE FOR A SINGLE STARTING POINT
This code can be used if there is a single starting point for all observations in your data set. For example, the number of visits to a website after it went live. The only requirement to run the code is to let SAS know the start and end dates of the period of interest by populating them into macro variables:

```sas
%LET start='01May2012'd;
%LET end = date();
```

Now that the start and end dates of the period of interest have been set, we need to create a data set that will be used by PROC FORMAT. The following DATA step figures out the start and end dates of each 7-day period between the start and end dates set above. Note that the variables FMTNAME, START, END, and LABEL will all be used by PROC FORMAT to set the parameters of your format.

```sas
DATA SET_WEEKS;
RETAIN fmtname 'weeks';
start=&start;
end=start+6;
label="Week: "||PUT(start,mmddyy10.)||" - "||PUT(end,mmddyy10.);
OUTPUT;
DO UNTIL (end > &end);
start=start+7;
end=end+7;
label="Week: "||PUT(start,mmddyy10.)||" - "||PUT(end,mmddyy10.);
OUTPUT;
END;
KEEP fmtname start end label;
FORMAT start end mmddyy10.;
RUN;
```
We can now use PROC FORMAT to create our date format:

```
PROC FORMAT CNTLIN=SET_WEEKS;
RUN;
```

We now have a format that we can use to group our dates by the periods we need. The following code shows an example of how the format can be used:

```
DATA test;
DO date=&start TO &end;
  OUTPUT;
END;
KEEP date;
RUN;

PROC FREQ data=test;
  TABLES date;
  FORMAT date weeks.;
RUN;
```

**The FREQ Procedure**

<table>
<thead>
<tr>
<th>Date</th>
<th>Frequency</th>
<th>Percent</th>
<th>Cumulative Frequency</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week: 05/01/2012 - 05/07/2012</td>
<td>7</td>
<td>9.59</td>
<td>7</td>
<td>9.59</td>
</tr>
<tr>
<td>Week: 05/08/2012 - 05/14/2012</td>
<td>7</td>
<td>9.59</td>
<td>14</td>
<td>19.18</td>
</tr>
<tr>
<td>Week: 05/15/2012 - 05/21/2012</td>
<td>7</td>
<td>9.59</td>
<td>21</td>
<td>28.77</td>
</tr>
<tr>
<td>Week: 05/22/2012 - 05/28/2012</td>
<td>7</td>
<td>9.59</td>
<td>28</td>
<td>38.36</td>
</tr>
<tr>
<td>Week: 05/29/2012 - 06/04/2012</td>
<td>7</td>
<td>9.59</td>
<td>35</td>
<td>47.95</td>
</tr>
<tr>
<td>Week: 06/05/2012 - 06/11/2012</td>
<td>7</td>
<td>9.59</td>
<td>42</td>
<td>57.53</td>
</tr>
<tr>
<td>Week: 06/12/2012 - 06/18/2012</td>
<td>7</td>
<td>9.59</td>
<td>49</td>
<td>67.12</td>
</tr>
<tr>
<td>Week: 06/19/2012 - 06/25/2012</td>
<td>7</td>
<td>9.59</td>
<td>56</td>
<td>76.71</td>
</tr>
<tr>
<td>Week: 06/26/2012 - 07/02/2012</td>
<td>7</td>
<td>9.59</td>
<td>63</td>
<td>86.30</td>
</tr>
<tr>
<td>Week: 07/03/2012 - 07/09/2012</td>
<td>7</td>
<td>9.59</td>
<td>70</td>
<td>95.89</td>
</tr>
<tr>
<td>Week: 07/10/2012 - 07/16/2012</td>
<td>3</td>
<td>4.11</td>
<td>73</td>
<td>100.00</td>
</tr>
</tbody>
</table>

**CODE FOR MULTIPLE STARTING POINTS**

What if we have different starting dates for different key variables? In the following code, we want to identify sales by week once after a member enrolls in a loyalty program. Since each member could have a different enrollment date, we need to be able to calculate the sales by week for each individual based on when he/she enrolled.
The first step is to identify the period of interest, give it a name, and indicate how many periods we want to track:

```%LET DaysInPeriod = 7;
%LET PeriodName = Week;
%LET TotalPeriods = 26;```

The next step now uses those inputs to create a data set that will be used by PROC FORMAT. This time, instead of using a date range for the format parameters, we will be setting a range for each period based on the number of days from the starting period:

```DATA DriveFormat;
LENGTH label $50;
RETAIN fmtname 'Period';
i = 1;
start = 0;
end = &DaysInPeriod - 1;
label = compbl("&PeriodName "||put(i,best.));
OUTPUT;
DO i = 2 to &TotalPeriods;
  start = end + 1;
  end = start + &DaysInPeriod - 1;
  label = compbl("&PeriodName "||put(i,best.));
  OUTPUT;
END;
RUN;```

We can use the FREQUENCY procedure to make sure the code above set the periods of interest correctly:

```PROC FREQ DATA=DriveFormat;
  TABLES start * end * label / LIST MISSING;
RUN;```

We can use the FREQUENCY procedure to make sure the code above set the periods of interest correctly:

```PROC FREQ DATA=DriveFormat;
  TABLES start * end * label / LIST MISSING;
RUN;```
We can now use PROC FORMAT to create our date format:

```sas
PROC FORMAT CNTLIN=DriveFormat;
RUN;
```

Here is an example of using the format. The data set SampleData contains 3 fields: ID, Enrollment Date, and Sales Date. The first thing we need to do is to calculate the number of days between each member's enrollment date and each sale. We will delete any records for sales that took place prior to enrollment, as we are only interested in behavior post-enrollment.

```sas
DATA CalculateDays;
SET SampleData;
days = SaleDate - EnrolLDate;
if days < 0 then delete;
if days > (&DaysInPeriod * &TotalPeriods) - 1 then delete;
period = PUT(days, period.);
RUN;
```

<table>
<thead>
<tr>
<th>id</th>
<th>EnrollDate</th>
<th>SaleDate</th>
<th>days</th>
<th>period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>02/10/2011</td>
<td>02/10/2011</td>
<td>0</td>
<td>Week 1</td>
</tr>
<tr>
<td>2</td>
<td>02/10/2011</td>
<td>02/28/2011</td>
<td>18</td>
<td>Week 3</td>
</tr>
<tr>
<td>3</td>
<td>02/10/2011</td>
<td>03/06/2011</td>
<td>24</td>
<td>Week 4</td>
</tr>
<tr>
<td>4</td>
<td>02/10/2011</td>
<td>03/18/2011</td>
<td>36</td>
<td>Week 6</td>
</tr>
<tr>
<td>5</td>
<td>02/10/2011</td>
<td>04/08/2011</td>
<td>57</td>
<td>Week 9</td>
</tr>
<tr>
<td>6</td>
<td>02/10/2011</td>
<td>04/18/2011</td>
<td>67</td>
<td>Week 10</td>
</tr>
<tr>
<td>7</td>
<td>02/10/2011</td>
<td>04/27/2011</td>
<td>76</td>
<td>Week 11</td>
</tr>
<tr>
<td>8</td>
<td>02/10/2011</td>
<td>05/07/2011</td>
<td>86</td>
<td>Week 13</td>
</tr>
<tr>
<td>9</td>
<td>02/10/2011</td>
<td>05/09/2011</td>
<td>88</td>
<td>Week 13</td>
</tr>
<tr>
<td>10</td>
<td>02/10/2011</td>
<td>05/26/2011</td>
<td>105</td>
<td>Week 16</td>
</tr>
<tr>
<td>11</td>
<td>02/10/2011</td>
<td>06/02/2011</td>
<td>112</td>
<td>Week 17</td>
</tr>
<tr>
<td>12</td>
<td>02/10/2011</td>
<td>06/11/2011</td>
<td>121</td>
<td>Week 18</td>
</tr>
</tbody>
</table>

**Figure 6.**
Screenshot of data set CalculateDays

Now we use PROC FREQ to get the number of sales for each member in each period:

```sas
PROC FREQ DATA=CalculateDays noprint;
   TABLES id * period / LIST MISSING OUT=freqs;
RUN;
```

<table>
<thead>
<tr>
<th>id</th>
<th>period</th>
<th>Frequency Count</th>
<th>Percent of Total Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Week 1</td>
<td>1</td>
<td>0.518134715</td>
</tr>
<tr>
<td>2</td>
<td>Week 10</td>
<td>1</td>
<td>0.518134715</td>
</tr>
<tr>
<td>3</td>
<td>Week 11</td>
<td>1</td>
<td>0.518134715</td>
</tr>
<tr>
<td>4</td>
<td>Week 13</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Week 16</td>
<td>1</td>
<td>0.518134715</td>
</tr>
<tr>
<td>6</td>
<td>Week 17</td>
<td>1</td>
<td>0.518134715</td>
</tr>
<tr>
<td>7</td>
<td>Week 18</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Week 20</td>
<td>1</td>
<td>0.518134715</td>
</tr>
<tr>
<td>9</td>
<td>Week 21</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 7.**
Screenshot of data set FREQS
In this example, we want each weekly period available as a variable, so we will use the TRANSPOSE procedure to create our weekly variables.

```
PROC TRANSPOSE DATA=freqs OUT=Transposed(DROP=_name_ _label_);
  BY id;
  VAR count;
  ID period;
RUN;
```

As shown in the example above, the weekly periods do not necessarily get created in the order we might want to have them stored in our data set. The following section of code shows how to re-order your variables and to set any missing weekly variables to zero:

```
DATA Ordered;
  LENGTH id &PeriodName._1 - &PeriodName._&TotalPeriods 8;
  SET transposed;
  ARRAY xvar{*} &PeriodName._1 - &PeriodName._&TotalPeriods;
  DO i = 1 TO dim(xvar);
    IF xvar{i} = . THEN xvar{i} = 0;
  END;
  DROP i;
RUN;
```

Now we can do anything we want to with our new variables:

```
PROC MEANS DATA=ordered SUM;
  VAR &PeriodName._1 - &PeriodName._&TotalPeriods;
RUN;
```
CAVEATS

The first code will group dates into formatted values that are each 7 days long, with the exception of the last formatted value. If the time period specified is not evenly divisible by 7, the final formatted value will contain all fewer than 7 days. This is important to remember to either clarify results or adjust time periods as necessary.

When using the second bit of code, be aware of any members in your dataset who may have a “master date” that is not far enough in the past to allow for each period’s activity. For example, if you are looking for 8 weeks of sales post-enrollment, and one of your members enrolled 4 weeks ago, that person will show no sales for weeks 5-8, when in actuality weeks 4-8 have simply not happened yet for that individual. You may wish to account for this when setting up your dataset or add logic to this code to remove everyone that does not yet have the full period available.

CONCLUSION

In conclusion, some basic SAS statements, functions, formats, and procedures can allow you to bucket information into the date groupings you need. These pieces of code can be easily modified to adjust to many different unique situations.

REFERENCE – FIRST CODE IN ITS ENTIRITY FOR EASY COPY/PASTE

Here is the entire first program for anyone who wants do copy, paste, and use!

```sas
********************************************************************************;
** start date will be used as day 1 of week 1 - all weeks will be 7-day weeks **;
** that start on the same day of the week as this date **;
********************************************************************************;
%LET start='01May2012'd;
%LET end = date();

******************************************************************************;
** create a dataset that will drive proc format - note, this will stop once it **;
** reaches a date greater than the end date specified **;
******************************************************************************;
DATA SET_WEEKS;
RETAIN fmtname 'weeks';
start=&start;
```
end=start+6;
label="Week: "||PUT(start,mmddyy10.)||" - "||PUT(end,mmddyy10.);
OUTPUT;
DO UNTIL (end > &end);
start=start+7;
end=end+7;
label="Week: "||PUT(start,mmddyy10.)||" - "||PUT(end,mmddyy10.);
OUTPUT;
END;
KEEP fmtname start end label;
FORMAT start end mmddyy10.;
RUN;

**********************************************************************************;
* use proc format to create the format to the work folder - the format's name **;
** is weeks.                                                                    **;
**********************************************************************************;
PROC FORMAT CNTLIN=SET_WEEKS;
RUN;

**********************************************************************************;
** this section just a sample showing how the format can now be used **;
*******************************************************************************;
DATA test;
  DO date=&start TO &end;
    OUTPUT;
  END;
  KEEP date;
RUN;

PROC FREQ data=test;
  TABLES date;
  FORMAT date weeks.;
RUN;

**********************************************************************************;
** end of program                                                               **;
**********************************************************************************;

REFERENCE – SECOND CODE IN ITS ENTIRITY FOR EASY COPY/PASTE
Here is the entire first program for anyone who wants do copy, paste, modify, and use!

**********************************************************************************;
** set macro variables                                                          **;
**********************************************************************************;
%LET DaysInPeriod = 7;
%LET PeriodName = Week;
%LET TotalPeriods = 26;

**********************************************************************************;
** this step creates the dataset that will drive PROC format                    **;
**********************************************************************************;
DATA DriveFormat;
LENGH label $50;
RETAIN fmtname 'Period';
i = 1;
start = 0;
end = &DaysInPeriod - 1;
label = compbl("&PeriodName "||put(i,best.));
OUTPUT;
DO i = 2 to &TotalPeriods;
    start = end + 1;
    end = start + &DaysInPeriod - 1;
    label = compbl("&PeriodName ", put(i,best.));
    OUTPUT;
END;
RUN;

******************************************************************************
** frequency to check our periods and make sure they calculated properly    **
******************************************************************************
PROC FREQ DATA=DriveFormat;
    TABLES start * end * label / LIST MISSING;
RUN;

******************************************************************************
** create the format                                                        **
******************************************************************************
PROC FORMAT CNTLIN=DriveFormat;
RUN;

******************************************************************************
** the dataset SAMPLEDATA contains 3 fields - ID, enrollment date, sale date **
** - calculate the days between enrollment date and each sale date, and       **
** remove any sales that took place before enrollment                        **
******************************************************************************
DATA CalculateDays;
SET SampleData;
    days = SaleDate - EnrollDate;
    if days < 0 then delete;
    if days > (&DaysInPeriod * &TotalPeriods) - 1 then delete;
    period = PUT(days, period.);
RUN;

******************************************************************************
** count the number of sales for each member, grouping by our formatted value **
******************************************************************************
PROC FREQ DATA=CalculateDays noprint;
    TABLES id * period / LIST MISSING OUT=freqs;
RUN;

******************************************************************************
** now we can use PROC transpose to have a separate variable for each period **
******************************************************************************
PROC TRANSPOSE DATA=freqs OUT=Transposed(DROP=_name_ _label_);
    BY id;
    VAR count;
    ID period;
RUN;

******************************************************************************
** this step shows how to use our macro variables to order the new variables **
******************************************************************************
DATA Ordered;
    LENGTH id &PeriodName._1 - &PeriodName._&TotalPeriods 8;
    SET transposed;
    ARRAY xvar{*} &PeriodName._1 - &PeriodName._&TotalPeriods;
    DO i = 1 TO dim(xvar);
        IF xvar[i] = . THEN xvar[i] = 0;
    END;
DROPOUT;
*** now we can use all of our weekly variables however we want ***
*** now we can use all of our weekly variables however we want ***
PROC MEANS DATA=ordered SUM;
  VAR &PeriodName._1 - &PeriodName._&TotalPeriods;
RUN;

*** end of program ***
*** end of program ***

CONTACT INFORMATION

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

Name: Susan Bakken
Enterprise: Aimia
Address: 1405 Xenium Lane
City, State ZIP: Plymouth, MN 55441
Work Phone: 763.445.3619
E-mail: susan.bakken@aimia.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.