

DATE Intervals - The INTCK Function

Debbie Hoppe, John Alden Life Insurance Company, Sacramento, CA

Introduction

SAS provides us the INTCK function to calculate the number of time intervals that occur between two dates. INTCK can be confusing to some programmers because it is not always intuitive. The purpose of this paper is to illustrate by example how INTCK calculates time intervals.

INTCK calculates the number of interval boundaries between two dates - not necessarily the elapsed time.

Day Intervals

The two date intervals that can be used to calculate the number of days between two dates are DAY and WEEKDAY. In order to calculate the number of days during this conference, you might code:

```
CONFTIME=INTCK('DAY', '10SEP95'D, '12SEP95'D);
```

The answer you would get is 2. SAS considers that a day starts at 12:00 a.m. and what is really being calculated here is that there are two day "starts" between these two days.

If you were to code:

```
CONFTIME=INTCK('WEEKDAY', '10SEP95'D, '12SEP95'D);
```

you would also get an answer of 2. Although September 10 is a Sunday, what is really being counted here is the number of weekday "starts" occurring between the two days with Monday through Friday being considered as weekdays.

There is a certain amount of flexibility that you can give the WEEKDAY interval. If you wanted to just consider Saturdays to be weekends, but Sunday to be a weekday, you could code:

```
CONFTIME=INTCK('WEEKDAY7W', '09SEP95'D, '12SEP95'D);
```

which would indicate that only Saturday (day 7) is considered to be a weekend day. The answer you would receive would be 3 - since Sunday, September 10 would be considered to be a weekday. Conversely, you could code WEEKDAY to only consider Sunday as a weekend day by replacing the "7" in the expression with a "1".

Week Intervals

The same general logic that applies to counting days applies to counting weeks. SAS considers that weeks start on Sunday, and when you calculate week intervals, the result is the number of week "starts" that occur during the period. For example, this code:

```
CONFTIME=INTCK('WEEK', '09SEP95'D, '12SEP95'D);
```

will return a value of 1. This is because a week starts on Sunday - and there is one Sunday between these two dates. However, if you code:

```
CONFTIME=INTCK('WEEK', '10SEP95'D, '12SEP95'D);
```

the result will be 0 because there are no week "starts" between these dates.

Once again, SAS gives us some flexibility in dealing with week periods and how they are defined. We can tell SAS that we want to calculate multiple week intervals as well as defining which day of the week we want a week to start on. By coding

```
CONFTIME=INTCK('WEEK1.2', '10SEP95'D, '12SEP95'D);
```

we are telling SAS that we want to calculate the number of one week periods (the 1 before the period) starting on Monday (day 2 of the week) between the two dates. In this case, the result returned will be 1.

Less than 1 month Intervals

SAS provides us with a couple of intervals for calculating portions of months. These are TENDAY and SEMIMONTH.

```
CONFTIME=INTCK('TENDAY', '10SEP95'D, '12SEP95'D);
```

would return a result of 1. Although there are only two days between the starting and ending date, one of the days that is passed is the 11th of September, and that is the start of a "tenday".

If we replace 'TENDAY' with 'SEMIMONTH' in the above expression, the result will be 0. This is because there are no semi-month "starts" between the 10th and 12th of September - semi-months start on the 1st and 16th of each month.

Month Intervals

There is a pattern emerging here, isn't there? INTCK behaves very predictably - as long as we understand exactly what it is counting. So let's move on and take a look at month intervals and see how SAS defaults as well as how we can shift intervals.

```
NEXTCONF=INTCK('MONTH', '10SEP95'D, '17OCT96'D);
```

returns a value of 13 - because there are 13 month "starts" - or first of the months that occur between these two dates.

```
NEXTCONF=INTCK('MONTH2', '10SEP95'D, '17OCT96'D);
```

returns a value of 6. The MONTH2 expression tells SAS to count the number of 2 month intervals between the two dates. The two month interval "starts" would be January 1, March 1, May 1, July 1, September 1, and November 1. However

```
NEXTCONF=INTCK('MONTH2.2', '10SEP95'D, '17OCT96'D);
```

returns a value of 7. This is because what we are telling SAS to do is to count the number of two month intervals where the intervals start on month 2 (the 2 after the period) - or February. Therefore, the two month interval "starts" are February 1, April 1, June 1, August 1, October 1, and December 1. In this case, there are 7 of these dates included in the date range specified.

Semiyear and Year Intervals

Finally, SAS gives us the QTR, SEMIYEAR and YEAR intervals to calculate years and portions of years.

```
NEXTQTR=INTCK('QTR', '01OCT95'D, '31DEC95'D);
```

returns a value of 0 while

```
NEXTQTR=INTCK('QTR','01OCT95'D,'01JAN96'D);
```

and

```
NEXTQTR=INTCK('QTR','30SEP95'D,'31DEC95'D);
```

both return a value of 1. This is because quarter "starts" occur on January 1, April 1, July 1, and October 1. What would happen in each of these examples if the SEMIYEAR interval were substituted for the QTR interval? The first example would return 0, while the 2nd would return 1 and the 3rd a 0. Semiyear "starts" are on January 1 and July 1.

By default, SAS calculates the number of years by calculating the number of year "starts" in the period. Since a year starts on January 1,

```
YRCOUNT=INTCK('YEAR','30DEC95'D,'02JAN96'D);
```

returns a value of 1.

The Conclusion and Cheat Sheet

All of the examples that have been shown in this paper serve to prove that the INTCK function does not calculate elapsed time, instead it calculates the number of interval "starts" between two dates using fixed points in time. Generally speaking, an interval "start" is the beginning of the interval. But here's a quick cheat sheet that gives the official default "start" for each of the date intervals:

DAY and WEEKDAY - the start of the day

WEEK - Sunday

TENDAY - the 1st, 11th, and 21st of the month

SEMIMONTH - the 1st and 16th of the month

MONTH - the 1st of the month

QTR - the 1st of January, April, July, and October

SEMIYEAR - the 1st of January and 1st of July

YEAR - the 1st of January

You can gain flexibility by using interval shifts to count multiple intervals as well as changing the "start". Interval shift examples:

YEAR.2 - number of years where years start on February 1

YEAR2.2 - number of 2 year period where the 2 year periods start on February 1.

MONTH3.2 - number of 3 month periods where the 3 month periods start on February 1.

WEEK3.2 - number of 3 week periods where the weeks start on Monday (day 2).

The SAS System also includes a set of interval values for dealing with both date-time and time variables. These function very similarly to the date intervals.

References

SAS Institute Inc. 1991. SAS Technical Report P-222 Changes and Enhancements to Base SAS Software Release 6.07. Cary, NC: SAS Institute, Inc.

SAS Institute Inc. 1990. SAS Language Reference, Version 6, First Edition, Cary, NC: SAS Institute, Inc.