LISTEN DATA
MAKE YOUR DATA TELL A STORY

HOME   SAS   R   PYTHON   DATA SCIENCE   CREDIT RISK   EXCEL   SQL   CALCULATORS   MORE

SEARCH...   GO

SAS : INTCK FUNCTION WITH EXAMPLES

Deepanshu Bhalla   22 Comments   SAS Function

The **INTCK** function is used to calculate the difference between dates, times or datetime values in SAS.

Syntax of INTCK Function

The syntax of INTCK is as follows -

```
INTCK(date-or-time-interval, start-date-or-time, end
```

1. **date-or-time-interval :** Date or time period needs to be defined in the first parameter. For eg. MONTH, YEAR, QTR, WEEK, HOUR, MINUTE etc. Specify period in single quotes.

2. **start-date-or-time :** Starting date or time to calculate the number of periods.

3. **end-date-or-time :** End date or time to calculate the number of periods.

4. **method : Optional Parameter.** Method to calculate the difference such as 'CONTINUOUS' or 'DISCRETE'. By default, it is DISCRETE.

Simple Example of INTCK Function

Calculate the number of years between two dates. In this case, two dates are 01JAN2015 and 01JAN2017.

```
data temp;
date1 = '01JAN2015'd;
date2 = '01JAN2017'd;
```

Follow us on
Facebook

```
no_of_years   = intck ('YEAR', date1, date2);
format date1 date2 date9.;
proc print data = temp;
run;
```

The **'YEAR'** keyword tells SAS to calculate the number of intervals between dates in terms of year. Since 01JAN2015 is a starting date, it is specified in the INTCK function before 01JAN2017. The FORMAT statement is used to display datevalues in date format when we print our results.

The output is shown below -

Obs	date1	date2	no_of_years
1	01JAN2015	01JAN2017	2

Other alias of year - 'YEARS' and 'YR'-

```
no_of_years   = intck ('YEARS', date1, date2)
no_of_years   = intck ('YR', date1, date2)
```

SAS INTCK Examples

Like calculation of years, we can use other intervals such as semiyear, quarter, month, week, day. The examples of these intervals are displayed below -

```
data temp;
date1 = '01JAN2015'd;
date2 = '01JAN2017'd;
no_of_years    = intck ('YEAR', date1, date2);
no_of_semiyears  = intck ('SEMIYEAR', date1, date2);
no_of_quarters   = intck ('QUARTER', date1, date2);
no_of_months   = intck ('MONTH', date1, date2);
no_of_weeks    = intck ('WEEK', date1, date2);
no_of_days    = intck ('DAY', date1, date2);
format date1 date2 date9.;
proc print data = temp noobs;
run;
```

date1	date2	no_of_years	no_of_semiyears	no_of_quarters	no_of_months	no_of_weeks	no_of_days
01JAN2015	01JAN2017	2	4	8	24	105	731

Custom Intervals using INTCK Function

Suppose you are asked to calculate the number of 4 months interval between two dates.

```
data temp;
date1 = '01JAN2015'd;
date2 = '01JAN2017'd;
no_of_4months  = intck ('MONTH4', date1, date2);
run;
```

The **MONTH4** interval implies interval is of 4 months. It is equal to the number of months divided by 4. Don't confuse it with QUARTERS. QUARTERS is equal to interval of 3 months. Remember 4 Quarters in an year.

Result : no_of_4months = 6

Similarly, we can use the custom intervals in YEAR, QUARTER and other periods. For example, '**YEAR2**' tells SAS the interval is of 2 years. It would return 1 for the above mentioned dates.

How to Set Starting Point for Calculation

Let's compare `YEAR` and `YEAR.3` by calculating the number of years between two dates using the INTCK function.

```
data temp;
date1 = '31JAN2015'd;
date2 = '31DEC2016'd;
diff  = intck ('YEAR', date1, date2);
diff2  = intck ('YEAR.3', date1, date2);
format date1 date2 date9.;
proc print;
run;
```

Function	Result
intck ('YEAR', '31JAN2015'd, '31DEC2016'd)	1
intck ('YEAR.3', '31JAN2015'd, '31DEC2016'd)	2

How does it work :

1. **intck ('YEAR', date1, date2)** - It checks the number of times first of January appears as first of january is set as a starting point by default. The variable diff returns 1 as it includes only 01JAN 2016.

2. **intck ('YEAR.3', date1, date2)** - It checks the number of times first of March appears as **YEAR.3 refers to the period starting from 1st of March to end of February next year**. The variable diff2 returns 2 as it includes 01 March 2015 and 01March 2016.

Is it a month difference?

INTCK function says there is a month difference between 25OCT2016 and 03NOV2016. But there is no month difference between 01OCT2016 and 31OCT2016. How?

```
data temp;
month1= intck('month', '25OCT2016'd, '03NOV2016'd);
month2= intck('month', '01OCT2016'd, '31OCT2016'd);
proc print;
run;
```

Function	Result
intck ('month', '25OCT2016'd, '03NOV2016'd)	1
intck ('month', '01OCT2016'd, '31OCT2016'd)	0

INTCK checks whether the first day of the month lies with in the range. In the first case, 01 November falls between October 25 and November 03 so it returns 1. In the second case, it returns 0 as 01 November does not fall between 01OCT2016 and 31OCT2016.

How to correct it?

Add one more parameter at end of INTCK function. In the parameter, specify **'C'** which refers to **continuous method** for calculation.

```
data temp;
month1= intck('month', '25OCT2016'd, '03NOV2016'd, 'C');
proc print;
run;
```

The above function returns 0.

The **CONTINUOUS method** calculates continuous time from the start-of-period date specified in the second parameter of INTCK function.

Calculating Weekdays with INTCK Function

Suppose you are asked to calculate the number of weekdays -

```
data eg;
weekdays = intck('WEEKDAY', '11DEC2016'd ,'18DEC2016'd);
proc print;
run;
```

It returns 5. In this case, **saturday and sunday are considered weekends** and excluding from the calculation.

How to Define 6 days working

If you need to calculate number of working days between 2 dates considering **6 weekdays** -

```
data eg;
weekdays = intck('WEEKDAY1W', '11DEC2016'd ,'18DEC2016'd);
proc print;
run;
```

WEEKDAY1W implies sunday as weekend (1=Sunday, 2= MONDAY... 7=Saturday)

How to Set Custom Weekends

```
data eg;
weekdays = intck('WEEKDAY24W', '11DEC2016'd ,'16DEC2016'd);
proc print;
run;
```

WEEKDAY24W means MONDAY and WEDNESDAY are weekends. The above function returns 3.

Calculate between Datetime values

Suppose you need to calculate hours, minutes and seconds between two datetime values.

```
data temp2;
hours=intck('hour','01jan2016:10:50:00'dt,'01jan2016:11:55:00'dt);
minutes=intck('minute','01jan2016:10:50:00'dt,'01jan2016:11:55:00'd
seconds=intck('second','01jan2016:10:50:00'dt,'01jan2016:11:55:00'd
proc print noobs;
run;
```

hours	minutes	seconds
1	65	3900

Result - 1 hour, 65 minutes and 3900 seconds

Time Difference using INTCK Function

To calculate the time difference between two times using the INTCK function, you need to specify the time interval you want to measure. Here's the syntax of the INTCK function:

```
data temp3;
hours=intck('hour','12:00:00't, '23:05:00't);
minutes=intck('minute','12:00:00't,'23:05:00't);
seconds=intck('second','12:00:00't,'23:05:00't);
proc print noobs;
run;
```

Result : 11 hours 665 minutes 39900 seconds

🔥 Related Posts

- How to Round Numbers in SAS

- SAS: LOG Functions

- SAS: Numeric Functions (with Examples)

- SAS: TRANWRD and TRANSLATE Functions (with Examples)

- How to Use COUNTW Function in SAS (with Examples)

SAS Tutorials : Top 100 SAS Tutorials

📢 Spread the Word!

   Share   Share   Tweet

About Author:

Deepanshu founded ListenData with a simple objective - Make analytics easy to understand and follow. He has over 10 years of experience in data science. During his tenure, he worked with global clients in various domains like Banking, Insurance, Private Equity, Telecom and HR.

While I love having friends who agree, I only learn from those who don't

   Let's Get Connected   Email   in

LinkedIn

👑 Post Comment

22 Responses to "SAS : INTCK Function with Examples"

Anonymous January 30, 2017 at 10:25 PM

Thank u so much Deepanshu Bhalla. Awesome Explanation.
I have a doubt,
In the Set Custom Weekends part the differnce b/w the dates 11DEC2016 and
16DEC2016 is '6 days'. So on passing the function 'WEEKDAY24W' it will
consider Monday and Wednesday as weekends, so the number of weekdays
must be
(6-2) = 4 right? But y is it returning 3? Can u please explain?

Replies

Deepanshu Bhalla January 31, 2017 at 12:32 PM

It is because the difference between two days are 5 (not including
16DEC). See the code below -
data eg;
days = intck('DAY', '11DEC2016'd ,'16DEC2016'd);
proc print;
run;

Anonymous February 1, 2017 at 3:15 AM

Thank You got it. :)

Unknown April 30, 2018 at 5:44 AM

Calculate the months,weeks and days separately since the
coupon expiry date (from current date to 31-mar-2014)
if there is given coupon expiry date starting from
20feb2014,then how to calculate such type of problem??explain
plz

Anonymous February 19, 2017 at 7:33 AM

Hi thank you for your help. I just have one question - When I am trying to find
out the no of weeks between 01/11/2014 and 01/03/2015, it is showing 51
when the answer should be 52. Can you please suggest why this is so?

Replies

sri February 21, 2017 at 4:27 AM

Week start from Sunday. As you are taking date which is starting from
Saturday, it is calculating Saturday as 1 week.
try to use continuous method for calculation.

sri February 21, 2017 at 4:01 AM

Great explaination. Thank you

REPLY   DELETE

Unknown May 20, 2017 at 1:14 AM

Hi Deepanshu, when i m using "C" in intck fn sas hitting error saying "intck function call has too many arguments". please explain.

REPLY   DELETE

Replies

Anonymous June 6, 2017 at 3:57 AM

"c" method is not supported in older version.

DELETE

REPLY

Pradeep verma June 29, 2017 at 3:49 AM

App_Yr_Mon = intnx('month',disbursed_dt,0,'E').....What does E stand for ?

REPLY   DELETE

Replies

Unknown January 13, 2018 at 1:09 AM

E stands for 'ending' usually returns last day of the month, etc. Don't confuse intck with intnx function. It's a different function at all.

DELETE

REPLY

Unknown November 12, 2017 at 6:45 PM

Hi
data x;
input name$ date1 date2 date10.;
x1=intck('Month',date1,date2);
datalines;
neha '01DEC2018'd '01DEC2018'd
Gupta '01DEC2018'd '11DEC2018'd
Goyal '01DEC2018'd '10DEC2018'd
;
Run;

Could you please guide me what is the wrong with this code?

REPLY   DELETE

Replies

Anonymous November 18, 2017 at 11:15 AM

Hello

try this :

```
data x;
input name$ date1 date10. date2 date10.;
x1=intck('Month',date1,date2);
format date1 date2 date10.;
datalines;
neha 01DEC2018 01DEC2018
Gupta 01DEC2018 11DEC2018
Goyal 01DEC2018 10DEC2018
;
Run;
```

DELETE

REPLY

Anonymous April 20, 2018 at 5:56 PM

Can you please show any example which have multiple dates in dataset
Like i have a dataset which have 10 participant. Date of birth for all employees
are different and i want to know how old they are on 12/30/2017 date.

REPLY   DELETE

Unknown April 30, 2018 at 5:41 AM

Calculate the months,weeks and days separately since the coupon expiry date
(from current date to 31-mar-2014)
if there is given coupon expiry date starting from 20feb2014,then how to
calculate such type of problem??explain plz

REPLY   DELETE

Shivangi February 5, 2019 at 2:43 AM

what is the use of noobs?

REPLY   DELETE

Replies

Keshav Gupta March 17, 2019 at 5:38 AM

It will not show observations no. as first variable.

DELETE

Unknown June 22, 2020 at 11:19 AM

it supresses the number of obesevatin

DELETE

REPLY

Principal SAS Programmer December 23, 2021 at 8:52 PM

I love this site as quick correct responses to questions asked !!

REPLY   DELETE

Unknown March 18, 2022 at 12:08 AM

how week is calculated between two dates if i have 19DEC2019 and 24DEC2019 then sas says its 1 week but 06JAN2020 and 10JAN2020 sas says week is 0 how? please tell me logic

REPLY   DELETE

learner March 31, 2022 at 8:28 AM

can anyone write the code to find the differences between those dates.

data arjun;
input startday:$10. endday:$10. ;
datalines;
11jan2012 30jan2012
12jan2012 30jan2012
11mar2012 30mar2012
12apr2012 30may2012
31mar2022 15may2022
31mar2022 25may2022
;
run;

REPLY   DELETE

mayasa May 10, 2022 at 8:23 AM

Thank you so much Deepanshu Bhalla. Great explanation. I just have a question: I want to use intck function for data deduplication. I need to remove obs for the same person with different testing dates <12 months and keep the obs with > 12 months difference.

REPLY   DELETE

ADD COMMENT

← PREV   NEXT →