How can I utilize the MDY function in SAS with examples?

Authored by

stats writer

June 23, 2024

The MDY function in SAS is used to convert a given date value into a SAS date. It takes in three parameters - month, day, and year - and returns a SAS date value. This function can be utilized in various ways to manipulate and analyze date data in SAS. For example, it can be used to calculate the number of days between two given dates, or to extract the month or year from a date. Additionally, it can be used in data transformations and data cleaning processes, such as converting a string date into a proper SAS date format. Overall, the MDY function in SAS provides a powerful tool for accurately handling and manipulating date data in SAS programs.

Use the MDY Function in SAS (With Examples)

You can use the MDY function in SAS to return a date value from month, day and year values.

This function uses the following syntax:

MDY(month, day, year)

where:

month: Integer value for month from 1 to 12day: Integer value for day of the month from 1 to 31year: A two-digit or four-digit integer that represents the year

The following example shows how to use the MDY function in practice.

Example: How to Use the MDY Function in SAS

Suppose we have the following dataset in SAS that

contains information about sales made on various dates at some retail store:

```
/*create dataset*/
data my_data;
input month day year sales;
datalines;
4 15 2022 94
6 17 2022 88
7 25 2022 90
8 14 2022 105
10 13 2022 119
12 15 2022 100
1 4 2023 87
3 15 2023 90
5 29 2023 130
;
run;

/*view dataset*/
proc printdata=my_data;
```

Obs	month	day	year	sales
1	4	15	2022	94
2	6	17	2022	88
3	7	25	2022	90
4	8	14	2022	105
5	10	13	2022	119
6	12	15	2022	100
7	1	4	2023	87
8	3	15	2023	90
9	5	29	2023	130

The following code shows how to use the MDY function to create dates using the numeric values in the month, day and year columns:

```
/*create new dataset*/
data new_data;
set my_data;
date_numeric = mdy(month, day, year);
date_worddate = put(mdy(month, day, year), worddate.);
date_date9 = put(mdy(month, day, year), date9.);
date_mmddyy10 = put(mdy(month, day, year), mmddyy10.);
run;

/*view dataset*/
```

proc printdata=new_data;

Obs	month	day	year	sales	date_numeric	date_worddate	date_date9	date_mmddyy10
1	4	15	2022	94	22750	April 15, 2022	15APR2022	04/15/2022
2	6	17	2022	88	22813	June 17, 2022	17JUN2022	06/17/2022
3	7	25	2022	90	22851	July 25, 2022	25JUL2022	07/25/2022
4	8	14	2022	105	22871	August 14, 2022	14AUG2022	08/14/2022
5	10	13	2022	119	22931	October 13, 2022	13OCT2022	10/13/2022
6	12	15	2022	100	22994	December 15, 2022	15DEC2022	12/15/2022
7	1	4	2023	87	23014	January 4, 2023	04JAN2023	01/04/2023
8	3	15	2023	90	23084	March 15, 2023	15MAR2023	03/15/2023
9	5	29	2023	130	23159	May 29, 2023	29MAY2023	05/29/2023

Notice that we used the MDY function to create four new columns that all contain dates in various formats.

Note #1: You can find a complete list of potential date formats in SAS .

Note #2: You can find the complete documentation for the SAS MDY function .