How can I read dates with both 2-digit and 4-digit year?

Authored by

stats writer

July 1, 2024

Reading dates with both 2-digit and 4-digit year can be achieved by following a few simple steps. First, it is important to understand that dates can be written in different formats, such as month/day/year or day/month/year. To accurately read a date, it is important to know the format in which it is written. Once the format is determined, the next step is to identify the year component. In dates with 2-digit year, it is important to note that the first two digits represent the year in the 20th century, while the last two digits represent the year in the 21st century. For example, 03/05/21 would represent March 5th, 2021. In dates with 4-digit year, the first two digits represent the century and the last two digits represent the year. For example, 03/05/2021 would represent March 5th, 2021. By understanding these conventions and paying attention to the format, one can accurately read dates with both 2-digit and 4-digit year.

How can I read dates with both 2-digit and 4-digit year? | Stata FAQ

Let's read in some dates in string format.

```
clear
input str10 date_str
"4/16/2005"
"3/29/11"
"8/15/2009"
"5/14/07"
"5/3/08"
"10/10/2010"
end

list, clean

date_str
```

1. 4/16/2005
2. 3/29/11
3. 8/15/2009
4. 5/14/07
5. 5/3/08
6. 10/10/2010

As you can see there is a mixture of 2-digit and 4-digit years. Data entry issues like this
are more common than you might suspect. So, we will use the date function along
with the %td format.

```
generate date=date(date_str,"MDY")
(3 missing values generated)

format %td date

list, clean

     date_str       date
1. 4/16/2005  16apr2005
2. 3/29/11 .
3. 8/15/2009  15aug2009
4. 5/14/07 .
```

5. 5/3/08 .
6. 10/10/2010 10oct2010

All three of the dates with 4-digit years came out fine. We could use some string
function or regular expression to insert the value 20 in front of the 2-digit years.
However, a simper way is to use the date function with the mask "MD20Y".

replace date=date(date_str,"MD20Y")
(3 real changes made)

list, clean

date_str date
1. 4/16/2005 16apr2005
2. 3/29/11 29mar2011
3. 8/15/2009 15aug2009
4. 5/14/07 14may2007
5. 5/3/08 03may2008
6. 10/10/2010 10oct2010

That worked just fine, but what if you had dates that go back to the previous century?

Check out these dates.

```
clear
input str10 date_str
"4/16/2005"
"3/29/11"
"8/15/2009"
"5/14/07"
"5/3/08"
"10/10/2010"
"7/13/99"
"2/19/1997"
end

list, clean

date_str
1. 4/16/2005
2. 3/29/11
3. 8/15/2009
4. 5/14/07
5. 5/3/08
6. 10/10/2010
7. 7/13/99
```

8. 2/19/1997

We will start out as before.

generate date=date(date_str,"MDY")
(4 missing values generated)

format %td date

list, clean

date_str date
1. 4/16/2005 16apr2005
2. 3/29/11 .
3. 8/15/2009 15aug2009
4. 5/14/07 .
5. 5/3/08 .
6. 10/10/2010 10oct2010
7. 7/13/99 .
8. 2/19/1997 19feb1997

Next, instead of one replace, we will use two. We will use the mask
"MD20Y" if the last two values of date_str are less than "50". When

the last two values are greater than "50" we will use the mask "MD19Y".

```
replace date=date(date_str,"MD20Y") if substr(date_str,-2,2)<"50"
(3 real changes made)

replace date=date(date_str,"MD19Y") if substr(date_str,-2,2)>"50"
(1 real change made)

list, clean

     date_str       date
  1. 4/16/2005   16apr2005
  2. 3/29/11     29mar2011
  3. 8/15/2009   15aug2009
  4. 5/14/07     14may2007
  5. 5/3/08      03may2008
  6. 10/10/2010  10oct2010
  7. 7/13/99     13jul1999
  8. 2/19/1997   19feb1997
```