Dates and Times in Stata

Working with Strings and Formatting

CONVERT STRING TO STATA DATETIME

generate datevar = **date**(stringdatevar, "DMY")
creates a Stata date variable from string date variable, recorded as the number of days since 1 January 1960
generate double datetimevar = **clock**(stringdatetimevar, "DMYhms")
creates a Stata datetime variable from string datetime variable, recorded as the number of milliseconds since 1 January 1960 at 00:00:00.000 (midnight)
D=date, M=month, Y=year, 19Y or 20Y indicates millennium of 2-digit years.
# used for non-date text that is between day/month/year/hour/min/sec.

CONVERT BETWEEN DATES AND TIMES

generate ddate = **dofc**(mydatetime)
converts a datetime variable to a date variable
1 Jan 2022 10:59:000 ⟶ 1 Jan 2022
generate double cdate = **cofd**(mydate)
converts a date variable to a time variable
1 Jan 2022 ⟶ 1 Jan 2022 00:00:000

FORMAT VARIABLE TO DISPLAY AS DATETIME

format ddate **%td**
date variable will now appear as text
format cdate **%tc**
datetime variable will now appear as text
If you don't format your Stata date/time variables they will appear as numbers. Formatting is for your visual benefit only.

DISPLAY THE DATETIME VARIABLE AS NUMERIC

display td(1 jan 2022)
display the number that represents a given date
display %15.0f tc(1 jan 2022 12:00:00.000)
display the number that represents a given datetime

Extracting Components of a Datetime Variable

EXTRACT DAYS

generate dayOfMonth = **day**(mydate)
calculate which day of the month the date is (1-31)
generate dayOfYear = **doy**(mydate)
calculate which day of the year the date is (1-366)
generate daysInMonth = **daysinmonth**(mydate)
calculate the total number of days in the month (28-31)
generate dayOfWeek = **dow**(mydate)
calculate the day of the week (1-7)

EXTRACT HOURS, MINUTES, SECONDS

generate hour = **hh**(mydatetime)
calculate the hour of the day the datetime is in (0-23)
generate minute = **mm**(mydatetime)
calculate the minute of the hour the datetime is in (0-59)
generate seconds = **ss**(mydatetime)
calculate the second of the minute the datetime is in (0.000-59.999)

EXTRACT WEEKS, MONTHS, QUARTERS, HALF-YEARS, YEARS

generate weekInYear = **week**(mydate)
calculate which week of the year the date is in (1-52)
generate monthOfYear = **month**(mydate)
calculate which month of the year the date is in (1-12)
generate quarterOfYear = **quarter**(mydate)
calculate which quarter of the year the date is in (1-4)
generate halfOfYear = **halfyear**(mydate)
calculate which half of the year the date is in (1-2)
generate year = **year**(mydate)
calculate which year the date is in (likely 1800-2100)

NOTE: The extracted variable is a separate variable from the original variable. If you remove your original datetime variable you will lose the specificity of the original variable.

Set Up

SET UP DATASET (or get the do-file at https://sdas.au/date)

```
clear
set obs 10
set seed 62470177
generate mydate = runiformint(td(1 jan 2022), td(30 jun 2022))
generate mydate2 = runiformint(td(1 jul 2022), td(31 dec 2022))
format mydate* %td
generate double mydatetime = runiformint(tc(1 jan 2022 00:00), tc(30 jun 2022 23:59))
generate double mydatetime2 = runiformint(tc(1 jul 2022 00:00), tc(31 dec 2022 23:59))
format mydatetime* %tc
```

Variables containing weeks / months / quarters / half-years are stored as the number of weeks / months/ quarters / half-years since 1 January 1960. These must be formatted according to your variable.

4 =

%td == 5 January 1960
%tw == 29 January 1960
%tm == 1 May 1960
%tq == 1 January 1961
%th == 1 January 1962

Assemble a Date/Time

ASSEMBLE A DATE VARIABLE FROM DAY, MONTH, YEAR VARIABLES

generate date3 = **mdy**(monthOfYear, dayOfMonth, year)
format date3 %td
calculate a stata date variable from separated month (1-12), day (1-31), and year(probably 1800-2100)

ASSEMBLE A DATETIME VARIABLE FROM DATE OR DAY, MONTH, YEAR, HOUR, MINUTE, SECOND VARIABLES

generate double datetime3 = mdyhms(monthOfYear, dayOfMonth, year, hour, minute, seconds)
format datetime3 **%tc**
calculate a stata datetime variable from separated month, day, year, hour (0-23), minute (0-59), and seconds (00.000-59.999)
generate double datetime4 = **dhms**(date3, hour, minute, seconds)
format datetime4 **%tc**
calculate a stata datetime variable from separated stata date, hour, minute, and seconds

ASSEMBLE A WEEK, MONTH, QUARTER, HALF-YEAR DATE VARIABLE FROM WEEKS, MONTHS, QUARTERS, HALF-YEARS AND YEARS VARIABLES

generate date4 = **yw**(year, weekInYear)
format date4 %tw
calculate a stata date variable from week (1-52) and year
generate date5 = **ym**(year, monthOfYear)
format date5 %tm
calculate a stata date variable from month and year
generate date6 = **yq**(year, quarterOfYear)
format date6 %tq
calculate a stata date variable from quarter (1-4) and year
generate date7 = **yh**(year, halfOfYear)
format date7 %th
calculate a stata date variable from halfyear (1-2) and year

Perform Calculations

CALCULATE AGE

generate age = **age**(mydate, today())
calculate the age (in years)
generate decimal_age = **age_frac**(mydate, today())
calculate the age (in years) including fraction of current year

CALCULATE LEAP YEAR

generate leapyear = **isleapyear**(year(mydate))
calculate if the year is a leapyear, 1 if it is a leap year, 0 if it is not

CALCULATE DIFFERENCE BETWEEN TWO DATES OR DATETIMES

generate difference = **datediff**(mydate, mydate2, "day")
calculate the difference (in days) between two dates, can use "month" or "year" in place of "day"
generate double datetime_difference = **clockdiff**(mydatetime, mydatetime2, "hour")
calculate the difference (in hours) between two datetimes, can use "day", "month", "year", "minute", or "second" in place of "hour"

CALCULATE BIRTHDAY (OR ANY ANNIVERSARY DATE)

generate bday = **birthday**(mydate, year(today()))
format bday **%td**
calculate the exact birthday of a given year
generate next_bday = **nextbirthday**(mydate, mydate2)
format next_bday **%td**
calculate the date of the next birthday after a given date
generate last_bday = **previousbirthday**(mydate2, mydate)
format last_bday **%td**
calculate the date of the last birthday before a given date

USE DATE RANGES FOR SUBSETTING

summarize decimal_age if mydate < td(30 jun 2022)
summarise the decimal age if mydate is before 30 June 2022
list mydate if mydate >= td(1 jan 2022) & mydate <= td(28 feb 2022)
lists dates if they are within the months of January or February
count if mydatetime > tc(1 April 2022 00:00)
count the number of observations with a datetime after 1 April at midnight

sdas TechTips
Statistics | Data | Analytics | Solutions
support@surveydesign.com.au
www.surveydesign.com.au
02 6247 0177