Short supplementary notes to handling time series in SAS

Useful formats

year4. Gives the year of a date variable as four digits, for instance 2019.

monyy7. Gives the month of a date variable as three letters and the year with four digits, for instance MAY2019.

ddmmyy10. Gives a date in a Danish way as for example 07/08/2019 for August 7..

dandfwkx31. Gives a date written in a formal Danish way as for instance including the weekday as for example søndag, den 27. oktober 2019.

datetime20.3 Gives date and time as for instance 27OCT19:09:08:02.916. Notes the three decimals of the second

downame. Gives only the weekday written in letters as for example Sunday.

See help to date time formats by Google

Useful functions

date() gives today's date in a date variable. Note that no arguments should be stated within the parentheses.

datetime() gives the date and exact time NOW in a date-time variable. Note that no arguments should be stated within the parentheses.

'12oct2020'd gives date in form of a date variable.

'21oct2021:11:59:37'dt gives date and time in form of a date time variable.

day(d) extracts the day as a number 1 - 31 for the date variable named d.

week(d) extracts the week number for the date variable named d.

month(d) extracts the month number for the date variable named d.

qtr(d) extracts the quarter number for the date variable named d.

year(d) extracts the year for the date variable named d.

weekday(d) gives the day number in the actual week for the date variable date - note that it is the American way as Sunday is the first day in the week so Saturday is weekday 7.

datepart(dt) extracts the date part of at date-time variable named dt

timepart(dt) extracts the time part of a date-time variable named dt

mdy(month,day,year) constructs a date variable for the date specified by the arguments. The arguments could be either variables or specific numbers

hms(h,m,s) constructs a time variable given by numeric variables h, m and s where h means hours, m minutes and s seconds (note that the number s could include decimals like s=23.457 for a time close to twenty three and a half seconds.

dhms(d,h,m,s) constructs a date-time variable for the the date variable d, and the time given by numeric variables h, m and s where h means hours, m minutes and s seconds (note that the number s could include decimals like s=23.457 for a time close to twenty three and a half seconds.

Seek help to date time functions, Google!

Note on PROC TIMESERIES

Important

Note that PROC TIMESERIES cannot aggregate day intervals for date-time variables. If the variable dt is a a date time variable and it should be used as an ID variable for aggregation over a whole day, the code

id dt interval=day ...;

is not working. Instead one should aggregate for interval=hour24, i.e..

id dt interval=hour24. ...;

However it is more intuitive to first extract the date part by the function datepart, see above, and then aggregate using this new constructed date variable. This is also convenient if the the days are to be aggregated further to e.g. months.

accumulate

In PROC TIMESERIES aggregation could be performed in many ways

accumulate=total gives the sum of all observed numbers in the interval

accumulate=average gives the average of all observations in the interval

accumulate=n gives the number of observations the interval

accumulate=min gives the minimum of the observations in the interval

accumulate=max gives the maximum of the observations in the interval

accumulate=stddev gives the standard deviation of the observations in the interval

setmissing

In PROC TIMESERIES missing observations could be set in various ways

setmissing=missing simply gives the value . (a punctuation mark) that is a missing value of a numerical variable.

setmissing=previous gives the value of the latest calculated aggregate value

setmissing=next gives the value of the next calculated aggregate value

setmissing= a number, e.g. setmissing=0 gives the value zero.

See Google for help to PROC TIMESERIES