TIME SERIES MAGIC:
SMOOTHING, INTERPOLATING, EXPANDING AND COLLAPSING
TIME SERIES DATA WITH PROC EXPAND
Andrew H. Karp
Sierra Information Services, Inc.
San Francisco, California    USA

Introduction

PROC EXPAND performs important operations on time series contained in SAS® System data sets. Among the key features of this procedure, found in the Econometrics and Time Series (ETS) module of SAS System software, are:

- conversion of observations from one sampling frequency to another
- interpolation of missing values in a time series
- changing the attributes of a time series observation
- transformation of observations in the series

PROC EXPAND creates an output SAS data set. It does not create printed output.

Benefits of Using PROC EXPAND

PROC EXPAND's capabilities make it a superior alternative to using other SAS System procedures (e.g., PROC MEANS or PROC SUMMARY) of aggregation of data collected in time. As will be discussed below, EXPAND's ability to estimate values of missing data in the series is not available in other procedures.

Extensive data step programming is also obviated by use of the TRANSFORMIN and TRANSFORMOUT options in PROC EXPAND. These options provide an array of tools that can be applied to observations in a SAS data set. In many instances the transformation operators available in PROC EXPAND cannot be implemented within the Data Step. Release 6.12 of SAS/ETS® software contains extensive additions to the EXPAND's data transformation capabilities. (See below.)

Key Terms and Concepts

Familiarity with key concepts underlying how the SAS System processes date, time and datetime variables is central to effective use of this procedure. These concepts are discussed in detail in Chapter 21 of the SAS/ETS® Users Guide, Version 6, Second Edition (“Date Intervals, Formats and Functions”) and this author's paper Working With SAS® Date and Time Functions elsewhere in these Proceedings. Among the key concepts underlying use of PROC EXPAND are:

- **Time Series**: a series (i.e., a vector) of data collected at equally spaced points in time
- **Period/Frequency**: the point in time (or period) at which the observations were collected (e.g., hourly, daily). In some instances the term **sampling frequency**, is used, and is defined as the frequency with which samples from some process were taken.
- **Attribute**: the point in the period where the observation was made (e.g., beginning, middle, end, total).
- **Interpolation**: changing from a lower frequency series to a higher frequency series.
- **Aggregation**: changing from a higher frequency series to a lower frequency series.

Figure 1 below demonstrates the concepts of interpolation and aggregation. If the analyst wants to convert daily observations in to monthly observations, she would **aggregate** the data. In this instance the **sampling frequency** with which the observations were taken is ‘daily.’ Conversely, if she had a series of quarterly values from which monthly values were to be estimated, she would **interpolate** values with a monthly sampling frequency from the quarterly series.
Input Data Set

Two important conventions need to be observed in the SAS data set upon which PROC EXPAND will operate. As described below, at least one variable in the input data set must be a SAS date, time or datetime variable. The name of this variable is usually placed in the ID statement. Unexpected results (explained below) will result if this convention is not observed.

Second, the ‘analysis variable(s)’ (that is, the variable(s) to be placed in the CONVERT statement must ‘form time series,’ which means their values are sorted in ascending order (from lowest to highest value) of the date, time or datetime variable which will be used in the ID statement. A value of the date time or datetime variable must be present for each interval in the series. For example, if the variables to be analyzed represent 24 months of observations, then the series must contain a SAS date value for each of the 24 months, even if one or more observations for the analysis variable(s) are missing. As will be discussed below, PROC EXPAND can be used to estimate values of missing data in the analysis variables.

General Form of PROC EXPAND

The general form of PROC EXPAND is:

PROC EXPAND DATA=dsn OUT=newdsn
FROM= timeint TO = timeint ;
ID datevar;
CONVERT varname = newvar/
OBSERVED = (inattrib, outattrib);
METHOD = method;

FROM and TO statements
These statements identify the sampling frequency with which the data were obtained (FROM) and the sampling frequency to which they are to be converted (TO).

ID Statement
The ID statement, while not strictly required, identifies a SAS date, time or date time variable which serves as the reference for the variables upon which PROC EXPAND is to operate. The observations in your data set must be sorted by the values of the variable in the ID statement or PROC EXPAND will not execute. Values of the variables placed in the ID statement are used by the SAS system to:

- identify observations in the output data set created by PROC EXPAND
- determine the time span between observations, as well as detect if gaps, caused by omitted observations, exist in the input data set
- account for ‘calendar effects,’ such as the number of days in a month and leap years, when performing the desired aggregations and/or interpolations.

If an ID statement is not used the SAS System will assume that the input series starts with the SAS date (or datetime, depending on the sampling frequency placed in the FROM statement) value of 0 (zero), or January 1, 1960. This can lead to undesirable results and use of the ID statement is, therefore, strongly recommended.

Interpolation of Missing Values

Series W10 (Quarterly U.S. Beer Production from the First Quarter of 1975 to the Fourth Quarter of 1982) in Wei’s Time Series Analysis: Univariate and Multivariate Methods was converted from raw data to a SAS data set, WORK.BEER. Each observation gives the quarterly total number of barrels of beer.
produced in the United States. The analysis variable is therefore PROD and the date time variable is QTR.

Suppose the input data set in this example, WORK.BEER, contained one or missing values for beer production, and all the analyst wanted to do was estimate values for the missing data. The following code fragment above will result in PROC EXPAND creating a data set with interpolated values for the missing data.

```
PROC EXPAND DATA = WORK.BEER OUT = NEWBEER FROM = QTR;
CONVERT PROD/OBSERVED=TOTAL;
ID QTR;
RUN:
```

In actual practice, only one invocation of the procedure would be required to accomplish both tasks, as PROC EXPAND would first estimate the values of missing data and then perform the aggregation of interest.

**Interpolation of Observations from a Lower to a Higher Sampling Frequency**

Using the quarterly beer production data again, assume that another (not shown here) data set is available with monthly values of beer production starting in January 1983. An analyst wants to take the quarterly data in the WORK.BEER data set, estimate monthly values from the quarterly data, and then append the estimated monthly data to the more recent monthly series before preparing a forecast.

The process by which the quarterly observations will be used to estimate monthly values is called interpolation, and is carried out by PROC EXPAND in much the same was as is aggregation of observations from a higher to a lower sampling frequency, which as discussed in the previous section.

```
PROC EXPAND DATA = WORK.BEER
FROM = QTR TO = MONTH
OUT = BYMONTH;
ID QTR;
CONVERT PROD = MTHPROD / OBSERVED = TOTAL;
RUN;
```

**Aggregation of Observations from a Higher to a Lower Sampling Frequency**

PROC EXPAND can be used to quickly aggregate these quarterly observations to yearly observations:

```
PROC EXPAND DATA = WORK.BEER
FROM = QTR TO = YEAR
OUT = YEARLY;
ID QTR;
CONVERT PROD = YEARPROD / OBSERVED = TOTAL;
RUN;
```

The result is a temporary SAS data set, YEARLY, containing the aggregated quarterly observations of beer production. The new data set contains two variables, YEAR and YEARPROD, representing the SAS date variable corresponding to the aggregated observations and the total yearly production, respectively.

**Combining Interpolation and Aggregation**

The examples presented thus far in the paper show separate invocations of PROC EXPAND for aggregation of data from quarterly to yearly observations and to estimate values of missing data in the time series.

**Transformation of Variables by PROC EXPAND**

The TRANSFORMIN and TRANSFORMOUT options in PROC EXPAND, first implemented in Release 6.08, provide a powerful array of tools which operate on the values of variables placed in the CONVERT statement. In most situations the analyst will using these feature of PROC EXPAND to be a superior alternative to performing variable transformations in the data step.

The data set “Case 9: Air-carrier freight” in Pankratz' *Forecasting with Univariate Box-Jenkins Models* was read in to a SAS data set (WORK.AIRCARGO) and will be used to demonstrate both the TRANSFORMIN and TRANSFORMOUT options.
The TRANSFORMIN Option
This optional statement performs transformations on input values before any aggregation or interpolation occurs. Using the WORK.AIRCARGO data set, which gives the number of tons of air cargo flown in the United States by month, the TRANSFORMIN option will rapidly convert the original values (tons) into pounds by use of the following code fragment:

```sas
PROC EXPAND DATA = WORK.AIRCARGO OUT = NEWCARGO FROM = MONTH;
ID DATE;
CONVERT TONS = POUNDS / OBSERVED = TOTAL TRANSFORMIN = (* 2000);
RUN:
```

In this example, the values of the variable TONS will be multiplied by 2000 and output to a new data set, NEWCARGO. The variable POUNDS in data set NEWCARGO is the result of applying the operations specified in the TRANSFORMIN statement. Since no other operations (e.g., interpolation or aggregation) are specified, the sole effect of this code fragment would be to transform from tons to pounds the monthly amount of air cargo flown in the United States.

The TRANSFORMOUT Option
This optional statement transforms data on values after PROC EXPAND performs aggregation, interpolation, estimation of values of missing data on the values of variables placed in the CONVERT statement but before they are output to a SAS data set. The TRANSFORMIN and TRANSFORMOUT options can be used in a single invocation of PROC EXPAND, as demonstrated by the following code fragment:

```sas
PROC EXPAND DATA = AIRCARGO OUT = NEW FROM = MONTH TO = QTR;
ID DATE;
CONVERT TONS = LOGPOUNDS / OBSERVED = TOTAL TRANSFORMIN = (* 2000)
TRANSFORMOUT = (LOG);
RUN:
```

In this example, tons are converted to pounds before the time series is aggregated from month to quarter. The natural logarithm is then taken on the output values before they are placed in the SAS data set NEW.

A complete list of operations that can be performed in the TRANSFORMIN and TRANSFORMOUT options is given the PROC EXPAND documentation.

Note: SAS and SAS/ETS are trademarks of SAS Institute, Inc.

Forty-two new operators for the TRANSFORMIN and TRANSFORMOUT options have been added in Release 6.12 of SAS/ETS® Software. Among the new operators are:

- Classical decomposition operations
- Centered moving:
  - Average
  - Minimum
  - Range
- Cumulative moving:
  - Sum
  - Median
  - Range
- Backward:
  - Moving Average
  - Sum
  - Median

Enhancements Control of Missing Values in Release 6.12
Prior to Release 6.12 of SAS/ETS Software PROC EXPAND only replaced missing values in a series with values interpolated via implementation of a cubic (default) or other spline function. Starting with Release 6.12, the SETMISS number operator allows the user to specify a number to be used to replace missing values in the series. The TRIM, TRIMLEFT and TRIMRIGHT operators will set values at the beginning or end of the series to missing. Finally, the MISSONLY MEAN operator will replace missing values with the arithmetic mean of the series.

Conclusions:
PROC EXPAND is an extremely powerful procedure for performing common operations on data collected in the time domain. Its abilities to aggregate and interpolate data are not duplicated in other parts of the SAS System. Addition of the TRANSFORMIN and TRANSFORMOUT options adds an additional array of tools to the procedure, many of which obviate the need for extensive data step programming. The range of transformation operators in PROC EXPAND was increased in Release 6.12, and now include moving averages and sums, centered moving averages, and backward moving averages, all common requirements when working with time series data. In
addition, Release 6.12 contains new options to control replacement of missing values in the series.

Learning the features of PROC EXPAND is, therefore, time well spent for those users of SAS System software who frequently work with time series data.

References:

Pankratz, Alan, *Forecasting with Univariate Box-Jenkins Models*, (Wiley: 1983)


The author can be contacted at:
Sierra Information Services, Inc.
1489 Webster Street Suite 1308
San Francisco, CA 94115 USA
415/441-0702
SFBAY0001 @ AOL.COM

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.