IBM SPSS Forecasting 26

Note

Before using this information and the product it supports, read the information in "Notices" on page 7.

Product Information

This edition applies to version 26, release 0, modification 0 of IBM® SPSS Statistics and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Forecasting

The following forecasting features are included in SPSS® Statistics Professional Edition or the Forecasting option.

Introduction to Time Series

A **time series** is a set of observations obtained by measuring a single variable regularly over a period of time. In a series of inventory data, for example, the observations might represent daily inventory levels for several months. A series showing the market share of a product might consist of weekly market share taken over a few years. A series of total sales figures might consist of one observation per month for many years. What each of these examples has in common is that some variable was observed at regular, known intervals over a certain length of time. Thus, the form of the data for a typical time series is a single sequence or list of observations representing measurements taken at regular intervals.

Table 1. Daily inventory time series

Time	Week	Day	Inventory level
t_1	1	Monday	160
t_2	1	Tuesday	135
t_3	1	Wednesday	129
t_4	1	Thursday	122
t_5	1	Friday	108
t_6	2	Monday	150
		...	
t_{60}	12	Friday	120

One of the most important reasons for doing time series analysis is to try to forecast future values of the series. A model of the series that explained the past values may also predict whether and how much the next few values will increase or decrease. The ability to make such predictions successfully is obviously important to any business or scientific field.

Time Series data

Column-based data

Each time series field contains the data for a single time series. This structure is the traditional structure of time series data, as used by the Seasonal Decomposition procedure, and the Spectral Analysis procedure. For example, to define a time series in the Data Editor, expand the **Variable List** and enter a variable name for any blank row. Each observation in a time series corresponds to a case (a row in the Data Editor).

If you open a spreadsheet that contains time series data, each series should be arranged in a column in the spreadsheet. If you already have a spreadsheet with time series arranged in rows, you can open it anyway and use **Data** > **Transpose...** to flip the rows into columns.

Multidimensional data

For multidimensional data, each time series field contains the data for multiple time series. Separate time series, within a particular field, are then identified by a set of values of categorical fields referred to as *dimension* fields.

For example, sales data for different regions and brands might be stored in a single *sales* field, so that the dimensions in this case are *region* and *brand*. Each combination of *region* and *brand* identifies a particular time series for *sales*. For example, in the following table, the records that have 'north' for *region* and 'brandX' for *brand* define a single time series.

Table 2. Multidimensional data

date	region	brand	sales
01/01/2014	north	brandX	82350
01/01/2014	north	brandY	86380
01/01/2014	south	brandX	91375
01/01/2014	south	brandY	70320
01/02/2014	north	brandX	83275
01/02/2014	north	brandY	85260
01/02/2014	south	brandX	94760
01/02/2014	south	brandY	69870

Data transformations

A number of data transformation procedures that are provided in the Core system are useful in time series analysis. These transformations apply only to column-based data, where each time series field contains the data for a single time series.

- The Define Dates procedure (on the Data menu) generates date variables that are used to establish periodicity and to distinguish between historical, validation, and forecasting periods. Forecasting is designed to work with the variables created by the Define Dates procedure.
- The Create Time Series procedure (on the Transform menu) creates new time series variables as functions of existing time series variables. It includes functions that use neighboring observations for smoothing, averaging, and differencing.
- The Replace Missing Values procedure (on the Transform menu) replaces system- and user-missing values with estimates based on one of several methods. Missing data at the beginning or end of a series pose no particular problem; they simply shorten the useful length of the series. Gaps in the middle of a series (*embedded* missing data) can be a much more serious problem.

See the *Core System User's Guide* for detailed information about data transformations for time series.

Estimation and validation periods

It is often useful to divide your time series into an *estimation*, or *historical*, period and a *validation* period. You develop a model on the basis of the observations in the estimation (historical) period and then test it to see how well it works in the validation period. By forcing the model to make predictions for points you already know (the points in the validation period), you get an idea of how well the model does at forecasting.

The cases in the validation period are typically referred to as holdout cases because they are held-back from the model-building process. Once you're satisfied that the model does an adequate job of forecasting, you can redefine the estimation period to include the holdout cases, and then build your final model.

Seasonal Decomposition

The Seasonal Decomposition procedure decomposes a series into a seasonal component, a combined trend and cycle component, and an "error" component. The procedure is an implementation of the Census Method I, otherwise known as the ratio-to-moving-average method.

Example

A scientist is interested in analyzing monthly measurements of the ozone level at a particular weather station. The goal is to determine if there is any trend in the data. In order to uncover any real trend, the scientist first needs to account for the variation in readings due to seasonal effects. The Seasonal Decomposition procedure can be used to remove any systematic seasonal variations. The trend analysis is then performed on a seasonally adjusted series.

Statistics

The set of seasonal factors

Seasonal Decomposition data considerations

Data The variables should be numeric.

Assumptions

The variables should not contain any embedded missing data. At least one periodic date component must be defined.

Estimating seasonal factors

1. From the menus choose:

   Analyze > **Forecasting** > **Seasonal Decomposition**
2. Select one or more variables from the **Available variables** list and move them into the **Selected variables** list. Note that the list includes only numeric variables.

Model Type

The Seasonal Decomposition procedure offers two different approaches for modeling the seasonal factors: multiplicative or additive.

- *Multiplicative*. The seasonal component is a factor by which the seasonally adjusted series is multiplied to yield the original series. In effect, seasonal components that are proportional to the overall level of the series. Observations without seasonal variation have a seasonal component of 1.
- *Additive*. The seasonal adjustments are added to the seasonally adjusted series to obtain the observed values. This adjustment attempts to remove the seasonal effect from a series in order to look at other characteristics of interest that may be "masked" by the seasonal component. In effect, seasonal components that do not depend on the overall level of the series. Observations without seasonal variation have a seasonal component of 0.

Moving Average Weight

The Moving Average Weight options allow you to specify how to treat the series when computing moving averages. These options are available only if the periodicity of the series is even. If the periodicity is odd, all points are weighted equally.

- *All points equal*. Moving averages are calculated with a span equal to the periodicity and with all points weighted equally. This method is always used if the periodicity is odd.
- *Endpoints weighted by .5*. Moving averages for series with even periodicity are calculated with a span equal to the periodicity plus 1 and with the endpoints of the span weighted by 0.5.

Optionally, you can:

- Click **Save** to specify how new variables should be saved.

Seasonal Decomposition: Save

Create Variables

Allows you to choose how to treat new variables.

- *Add to file*. The new series created by Seasonal Decomposition are saved as regular variables in your active dataset. Variable names are formed from a three-letter prefix, an underscore, and a number.

- *Replace existing*. The new series created by Seasonal Decomposition are saved as temporary variables in your active dataset. At the same time, any existing temporary variables created by the Forecasting procedures are dropped. Variable names are formed from a three-letter prefix, a pound sign (#), and a number.
- *Do not create*. The new series are not added to the active dataset.

New variable names

The Seasonal Decomposition procedure creates four new variables (series), with the following three-letter prefixes, for each series specified:

SAF *Seasonal adjustment factors.* These values indicate the effect of each period on the level of the series.

SAS *Seasonally adjusted series.* These are the values obtained after removing the seasonal variation of a series.

STC *Smoothed trend-cycle components.* These values show the trend and cyclical behavior present in the series.

ERR *Residual or "error" values.* The values that remain after the seasonal, trend, and cycle components have been removed from the series.

SEASON command additional features

The command syntax language also allows you to:

- Specify any periodicity within the `SEASON` command rather than select one of the alternatives offered by the Define Dates procedure.

See the *Command Syntax Reference* for complete syntax information.

Spectral Plots

The Spectral Plots procedure is used to identify periodic behavior in time series. Instead of analyzing the variation from one time point to the next, it analyzes the variation of the series as a whole into periodic components of different frequencies. Smooth series have stronger periodic components at low frequencies; random variation ("white noise") spreads the component strength over all frequencies.

Series that include missing data cannot be analyzed with this procedure.

Example
The rate at which new houses are constructed is an important barometer of the state of the economy. Data for housing starts typically exhibit a strong seasonal component. But are there longer cycles present in the data that analysts need to be aware of when evaluating current figures?

Statistics
Sine and cosine transforms, periodogram value, and spectral density estimate for each frequency or period component. When bivariate analysis is selected: real and imaginary parts of cross-periodogram, cospectral density, quadrature spectrum, gain, squared coherency, and phase spectrum for each frequency or period component.

Plots For univariate and bivariate analyses: periodogram and spectral density. For bivariate analyses: squared coherency, quadrature spectrum, cross amplitude, cospectral density, phase spectrum, and gain.

Spectral Plots data considerations

Data The variables should be numeric.

Assumptions

The variables should not contain any embedded missing data. The time series to be analyzed should be stationary and any non-zero mean should be subtracted out from the series.

- *Stationary*. A condition that must be met by the time series to which you fit an ARIMA model. Pure MA series will be stationary; however, AR and ARMA series might not be. A stationary series has a constant mean and a constant variance over time.

Obtaining a spectral analysis

1. From the menus choose:

   Analyze > **Forecasting** > **Spectral Analysis**

2. Select one or more variables from the **Available variables** list and move them to the **Selected variables** list. Note that the list includes only numeric variables.
3. Select one of the Spectral Window options to choose how to smooth the periodogram in order to obtain a spectral density estimate. Available smoothing options are Tukey-Hamming, Tukey, Parzen, Bartlett, Daniell (Unit), and None.

- *Tukey-Hamming*. The weights are Wk = .54Dp(2 pi fk) + .23Dp (2 pi fk + pi/p) + .23Dp (2 pi fk - pi/p), for k = 0, ..., p, where p is the integer part of half the span and Dp is the Dirichlet kernel of order p.
- *Tukey*. The weights are Wk = 0.5Dp(2 pi fk) + 0.25Dp (2 pi fk + pi/p) + 0.25Dp(2 pi fk - pi/p), for k = 0, ..., p, where p is the integer part of half the span and Dp is the Dirichlet kernel of order p.
- *Parzen*. The weights are Wk = 1/p(2 + cos(2 pi fk)) (F[p/2] (2 pi fk))**2, for k= 0, ... p, where p is the integer part of half the span and F[p/2] is the Fejer kernel of order p/2.
- *Bartlett*. The shape of a spectral window for which the weights of the upper half of the window are computed as Wk = Fp (2*pi*fk), for k = 0, ... p, where p is the integer part of half the span and Fp is the Fejer kernel of order p. The lower half is symmetric with the upper half.
- *Daniell (Unit)*. The shape of a spectral window for which the weights are all equal to 1.
- *None*. No smoothing. If this option is chosen, the spectral density estimate is the same as the periodogram.

Span. The range of consecutive values across which the smoothing is carried out. Generally, an odd integer is used. Larger spans smooth the spectral density plot more than smaller spans.

Center variables. Adjusts the series to have a mean of 0 before calculating the spectrum and to remove the large term that may be associated with the series mean.

Bivariate analysis—first variable with each. If you have selected two or more variables, you can select this option to request bivariate spectral analyses.

- The first variable in the Variable(s) list is treated as the independent variable, and all remaining variables are treated as dependent variables.
- Each series after the first is analyzed with the first series independently of other series named. Univariate analyses of each series are also performed.

Plot Periodogram and spectral density are available for both univariate and bivariate analyses. All other choices are available only for bivariate analyses.

- *Periodogram*. Unsmoothed plot of spectral amplitude (plotted on a logarithmic scale) against either frequency or period. Low-frequency variation characterizes a smooth series. Variation spread evenly across all frequencies indicates "white noise."
- *Squared coherency*. The product of the gains of the two series.
- *Quadrature spectrum*. The imaginary part of the cross-periodogram, which is a measure of the correlation of the out-of-phase frequency components of two time series. The components are out of phase by pi/2 radians.
- *Cross amplitude*. The square root of the sum of the squared cospectral density and the squared quadrature spectrum.

- *Spectral density*. A periodogram that has been smoothed to remove irregular variation.
- *Cospectral density*. The real part of the cross-periodogram, which is a measure of the correlation of the in-phase frequency components of two time series.
- *Phase spectrum*. A measure of the extent to which each frequency component of one series leads or lags the other.
- *Gain*. The quotient of dividing the cross amplitude by the spectral density for one of the series. Each of the two series has its own gain value.

By frequency. All plots are produced by frequency, ranging from frequency 0 (the constant or mean term) to frequency 0.5 (the term for a cycle of two observations).

By period. All plots are produced by period, ranging from 2 (the term for a cycle of two observations) to a period equal to the number of observations (the constant or mean term). Period is displayed on a logarithmic scale.

SPECTRA command additional features

The command syntax language also allows you to:

- Save computed spectral analysis variables to the active dataset for later use.
- Specify custom weights for the spectral window.
- Produce plots by both frequency and period.
- Print a complete listing of each value shown in the plot.

See the *Command Syntax Reference* for complete syntax information.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Index

E

H

S

V

Printed in USA