

SAS® Visual Forecasting on SAS® Viya™

Quickly and automatically generate large numbers of trustworthy forecasts so your organization can operate more efficiently and effectively



Key Benefits

- **Streamline and automate your forecasting process.** SAS Visual Forecasting automatically produces large-scale time series analysis and hierarchical forecasting. Less manual intervention reduces the chance of personal bias in the forecasting process. It also means forecast analysts don't have to spend their time building and monitoring forecasting models for every time series. They can focus on high-value forecasts or other strategic tasks.
- **Manage organizational planning challenges.** This solution takes advantage of SAS® Viya™, a new distributed, in-memory engine, to deliver results for millions of forecasts at break-through speeds. This solution generates forecasts on an enterprise scale - quickly, automatically and as accurately as can reasonably be expected so you can better plan for the future.
- **Empower users with language options.** Python, Java, R and Lua programmers can develop models in the open source language of their choice, access tested and trusted SAS time series and forecasting capabilities, and execute their models within the new SAS Viya engine for faster results.

Overview

SAS Visual Forecasting provides automatic variable, event and model selection. It then automatically generates your forecasts. With SAS Visual Forecasting, you can:

- Adopt a forecasting ecosystem that provides a resilient, distributed, optimized generic time series analysis scripting environment for cloud computing.
- Pick up the data once, and run everything you need, taking advantage of fast distributed processing.
- Use a scripting language that optimizes and compiles locally on each node on which it is running.

```

7 /* Accumulate data into monthly intervals, create time series data set at the lowest level of hierarchy: */
8 /* Region, Product Line, and Product Name */
9
10 proc tsmodel data = mycas.pricedata /* Single pass through the data for lowest level */
11      outsum = mycas.outsum_low
12      outobj = (
13          outfor = mycas.outfor_low
14          outstat = mycas.outstat_low
15          outstat = mycas.outstat_low
16      )
17      ;
18 id date interval = month;
19 var sale /acc = sum ;
20 var price/acc = average;
21 by regionName productLine productName;
22
23 /* Load ATSM package */
24 require atsm;
25
26 /* User defined program statements */
27
28 submit;
29 declare object dataframe(tsdf);
30 declare object diagnose(diagnose);
31 declare object diagspec(diagspec);
32 declare object forecast(foreng);
33 declare object outfor(outfor);
34 declare object outstat(outstat);
35 declare object outstat(outstat);
36 declare object outstat(outstat);
37
38 rc = dataframe.Initialize();
39 rc = dataframe.AddY(sale);
40 rc = dataframe.AddX(price);
41
42 /* Diagnose time series at the lowest level of the hierarchy: Region, Product Line, and Product Name */
43

```

Figure 1: The SAS Studio programming interface in SAS Visual Forecasting.

Capabilities

Large-scale automatic forecasting

SAS Visual Forecasting automatically analyzes large numbers of time series so forecasters don't have to do this for each series. It also generates large quantities of statistically based forecasts in a distributed, in-memory engine without the need for human intervention unless desired. Together, these capabilities provide flexible, scalable and powerful automated forecasting software.

Scripting language enables distributed processing of time series analysis

Distributed systems break up large files and process each piece separately, and this is problematic for time series analysis where the ordering of data is crucial. Time series analysis algorithms typically require that the time series data be stored contiguously in memory and in sorted order. SAS Visual Forecasting shuffles the data so that each time series (or BY group) is copied into the memory of a single computing node. Each time series is executed on one or more threads of a node, and each node executes the compiled script for each of its assigned time series. This makes large-scale time series analysis and forecasting possible on an enormous scale. And the scripting language is optimized for the machine it is running on, so users don't have to rewrite code for different machines.

Time series analysis

The TSMODEL procedure executes user-defined programs to convert time-stamped transactional data into a time series format. And it uses flexible hierarchies to organize the data so you can segment and manipulate data for better results.

- Supports autocorrelation and cross-correlation analysis.
- Supports count series analysis.

- Supports seasonal decomposition and adjustment analysis.
- Provides diagnostic testing for seasonality, stationarity and intermittency.

Time series modeling

The TSMODEL procedure generates forecast models automatically.

- Supports ARIMA (dynamic regression).
- Supports transfer functions definitions.
- Supports exponential smoothing models (ESM).

Automatic time series analysis and forecasting

The automatic time series analysis and forecasting package contains several objects, each designed to perform a particular task in the time series analysis process.

- Supports automatic time series model generation.
- Provides automatic input variable and event selection.
- Supports automatic model selection.
- Provides automatic forecast generation.

Hierarchical reconciliation

Each series in the hierarchy is modeled and forecast individually. Forecasts are then reconciled at multiple levels of the hierarchy in a top-down fashion. Users can adjust a forecast at the top level and apportion it to lower levels so the hierarchy maintains consistency, and individual forecasts (by products, locations, etc.) roll up to the top number. Without reconciliation, lower-level forecasts won't add up to the top-level forecast.

Open, cloud-enabled, in-memory platform

SAS Visual Forecasting takes advantage of the SAS Viya engine for even quicker insights. SAS Viya brings new enhancements to the SAS Platform, including high availability, faster in-memory processing and native cloud support. It also provides an open analytics coding environment. Whether it's SAS, Python, Java, R or Lua, analytical professionals can access the power of SAS using their language of choice. And with public REST APIs, you can add SAS Analytics to other applications. All analytical assets are managed within a common environment to provide a single, governed model inventory across applications.

Learn More

Find out more at sas.com/visualforecasting.

To contact your local SAS office, please visit: sas.com/offices

