Taking Back Control with the Interactive Modeling Node in SAS® Visual Forecasting

Danny Modlin
Senior Analytical Training Consultant

#SASinnovate

Contents

Abstract

Visual Forecasting, within SAS® Model Studio, provides users with the functionality to perform automatic, large-scale forecasting. Automatic model generation is part of this process. From ESM to ARIMAX, to UCM models, SAS® Visual Forecasting will take the lead and generate new model specifications that accommodate the signal components in your data. On occasion, there will be time series that will need extra focus due to unique circumstances. Will we be able to take back model generation control from SAS® Visual Forecasting if these unique times surface? Yes! Enter Visual Forecasting's super node: the Interactive Modeling Node. In this talk, we will explore the capabilities of the Interactive Modeling Node and how we can explore the series, adjust current models, and create models from scratch all from within one node.

Getting Started with Your Project

Data Dictionary

You will work with a transactional data set named **lookingglass_forecast**. The data consists of information from a telecommunications group whose goal is to predict sales.

The data set is already accumulated into a time series with **Txn_Month** as the time variable. Potential predictor variables of the continuous response **sale** are **price**, **discount**, and **cost**. The data set also contains the attribute variables that will be used in hierarchical forecasting, **productline** and **productname**.

Variables in LOOKINGGLASS_FORECAST

Variable	Type	Length	Format	Informat	Label
Txn_Month	Num	8	MMDDYY10.	DATE15.	Transaction Date (Month)
productline	Char	8			Name of product line
productname	Char	12			Product name
sale	Num	8			Unit Sale
price	Num	8			Unit Price
discount	Num	8			Price Discount
cost	Num	8			Unit Cost

Creating a Forecasting Project and Loading the Data

In this section, you create a new forecasting project in Model Studio, **baseline sales forecasts**, and load data into the project. The **baseline sales forecasts** project is used throughout the workshop.

1. Navigate to the upper left corner in SAS Viya Landing Page, click **Applications** > **Build Models**. This takes you to Model Studio. Click on the **New Project** button.

   Model Studio is an integrated visual environment that provides a suite of analytic tools to facilitate end-to-end data mining, text, and forecast analysis. The tools in Model Studio are designed to take advantage of SAS Viya programming and cloud processing environments to deliver and distribute the results of the analysis, such as champion models, score code, and results. It does all this fast.

   Note: If this is your first session, there will be no existing projects unless projects were set up for you. (If projects already exist, the **New Project** button is available in the upper right corner.)

2. Name your project **baseline sales forecasts**.

   Note: Naming your project something relevant and adding a reasonably detailed description of the project is considered a forecasting best practice.

3. For **Type**, select **Forecasting**.

   There are three types: Data Mining and Machine Learning, Forecasting, and Text Analytics. This workshop will only deal with Forecasting.

Once you select a type and you create your project, you can't change the Forecasting type. So, you would have to reopen a new project. Make sure you've got the right type before you save the project at the bottom of this window.

4. For **Template**, use the drop-down menu or the Browse button and choose **Auto-forecasting**.
5. For **Data Source**, click **Browse** to select the modeling data source.
6. Make sure the Show area says **Available** to view in-memory tables that are available for model building. Scroll and find the **LOOKINGGLASS_FORECAST** in-memory table. Alternatively, you can click on the search bar and type lookingglass and click the magnifying glass icon or hit Enter.
7. Select **LOOKINGGLASS_FORECAST**, click **OK**.
8. For **Location**, use the dropdown arrow to select **My Folder**, and then click **Save** to create the new project. The project now appears.
9. Ensure that the project's **Data** tab is selected to assign variable roles.

   A Note on Variable Assignment:

   - Individual variables can be selected for role assignments by either clicking the variable name or by selecting the corresponding check box.

- Individual variables are **deselected** after their role is assigned by either **clearing their check box** or **selecting another variable's name**.
- More than one variable can be selected at the same time using the check boxes.
- Because selecting a variable using the check box does not deselect others, it is easy for new users to **inadvertently re-assign** variable roles. Taking a few minutes to get comfortable with the variable selection functionality is considered the best practice for using the software.

10. The **Txn_Month** variable is assigned to the role of Time for the project.

    Note: Other time intervals are available by selecting the down arrow next to **Month**. The time interval combined with the Multiplier and Shift options indicates that the desired interval of the time series data is one month and that the 12-month annual cycle starts in January. These options can be changed to modify the time index if it is appropriate for your data.

11. **Sale** is the target for the analysis. Click **sale** in the middle variables list panel. In the right property panel, select **Dependent**.

    Note: Missing interpretation options enable the user to interpret, or impute, values for embedded missing values in the series. By default, embedded missing values have no value assigned to them.

12. Deselect **sale**. Assign **productline** and **productname,** in that order, to the **BY Variable** role.
13. Set Reconciliation level to **productline**.
14. Additional variables will be assigned to roles. **Price**, **discount**, and **cost** can be useful as explanatory variables in subsequent analyses. Select these three variables, where the order does not matter, and change their roles to **Independent**.

    Note: For each of these variables, accumulation is accomplished by averaging observed values in each month.

15. Change **Usage in system-generated models** to **Try to Use**.

If I select Force to use, then each of these three variables will be used in every one of the models for every one of the series. I don't want to do that. Let's see some other options-- Try to use and Use if significant.

Try to Use will test each of the variables in each of the series. So, for each model, for each of the 918 series, in the dataset each of the variables will be tested to see if they're statistically significant in the model, and whether they benefit the model with respect to a fit statistic, such as Akaike's information criterion. So, there are two criteria. If a variable passes both of those tests, they'll be used in the model. If the variable doesn't, it won't be used in the model.

Try to Use is slightly different from Use if significant. Use if significant only tests to see if the variable is statistically significant and doesn't check to see if it improves the model's fit statistic.

So I'm going to use Try to use. Now my data is ready to start my pipelines.

Autoforecasting Node

Performing Basic Forecasting with a Pipeline

In this section, you perform basic forecasting with a pipeline.

1. Navigate to the **Pipelines** tab.

   The Auto-Forecasting template is the default pipeline template for Visual Forecasting. It consists of the essential steps in a forecasting analysis:

   - accumulates the data into time series
   - automatically identifies, estimates, and selects forecast models for the time series
   - assesses forecasting results
   - publishes results for use outside the pipeline

Note: If the modeling data are hierarchically arranged, the identification, estimation, and selection steps in the default forecasting pipeline are done on series in the base level of the hierarchy.

2. Select **Run Pipeline** in the upper right corner of the workspace.

   Then we see the circles, the empty circles start up. And by the time we see all circles filled with green ink and a white checkmark, then the pipeline will be completed. Now that it's done, let's look at the Auto Forecasting node.

Auto-forecasting Node Results

1. Right-click on the **Auto-forecasting** node and select **Results**.

   Because the Auto-forecasting node is designed to be run with minimal input from the analyst, relatively few options are surfaced for this node. The Auto-forecasting node automatically identifies, estimates, and generates forecasts for the 918 series in the base or product name level of the modeling hierarchy. Most of the forecast models selected for these series are in the ARIMAX family.

   For each series, three families of time series models are considered by default: ARIMAX (ARIMA with exogenous variables), ESM (exponential smoothing models), and IDM (intermittent demand models). The champion model for each series is chosen based on root mean square error. Other selection statistics are available in the Model selection criterion option.

   The MAPE Distribution histogram is in the upper left-hand corner. The distribution of Mean Absolute Percent Error (MAPE) for forecasts in the product name level of the hierarchy can be used to compare the accuracy of different forecast models. Each of the bars represents the proportion of the series that have a specific range of MAPE values. In general, smaller values of MAPE imply greater accuracy.

   The Model Family histogram is in the upper right. Among each of these 918 series, we can see what percentage of each model family was selected as the best across all the time series.

The Model Type chart, located in the lower left, summarizes systematic variation found in the identification process. This chart shows information about the selected models across all the time series. What percentage of the models has a trend component, a seasonal component, or an predictor variable used? They can be overlapping series; therefore, we can see that the percentage is summed to over 100%.

The Execution Summary, located in the lower right, provides information about results that are potentially problematic, anomalous, or both. We can see there were 918 series. There weren't any series that failed to forecast. There were only six series with forecasts equal to zero, meaning in the forecast range, the forecasts were all zero. Then there is a lot of summary information about the number of series that had flat forecasts. Flat forecasts mean that in the forecast range, the forecast values were a constant.

2. Click the **Output Data** tab above the MAPE Distribution plot.

   Several output tables are created. You can view them by clicking on them.

   You'll notice that there is a unique product line, product name combination for each line. So that's a unique series identified by its product line and product name. And notice that we have multiple lines of data in this data set, each for a different month. So, if I scroll down, I can see that the Time ID. And once we get to 2017, we see the actual values is missing. What that means is that this is the forecast horizon. The forecast horizon, of course, doesn't have any actual values, but it can have predicted values and so on. So that's information you might want to obtain from that forecast table.

3. Click the **OUTMODELINFO** data source to open it.

   For each series, the selected model is named, and attributes of the model are displayed.

   Once again, we see information for every product line, product name combination. In other words, every series. And in this data set, we see the name of the model, or the label for the model, is the type of model that was chosen as the champion model for that series. So, for Line01, Product01, that series, it was an ARIMA model with regression parameters. It is under the ARIMA family. There were no dependent variables here. And we can get information about whether there are seasonal components, whether there are trend components, whether there are inputs presence, and so on.

   So that's information for the Champion model. If I want to see what the competitors were, we can click on the OUTSELECT Data Source. And now you'll see there are three lines for each one of the series. So Line01, Product01 is three different lines. And you can see which models were under consideration by looking at the Model column. And then the next column for Selected Status, you can see that the selected row, the selected model for this series, as we'd seen before, was the ARIMA model with regression parameters. And you can see why that happened by scrolling farther to the right, each one of these has fit statistics and accuracy statistics calculated. So, if we looked at the Mean Absolute Percent Error column, for those first three rows, you could see why that model won the competition for that series.

4. Close the **Results** window.

5. Right-click and open the results of the **Model Comparison** node.

   The Champion Model is the Auto-forecasting model, which is the only one included in the pipeline. WMAE and WMAPE are weighted sums of the MAE and MAPE values across all series. WMAPE and WMAE represent average performance of all the models in a modeling node.

Note: For the WMAPE and WMAE, the final computation is based on weighted measurements from each time series, where more weight is given to time series with a higher average of the dependent variable.

6. Close the **Results** window.

Interactive Modeling Node

Creating Custom Models with the Interactive Modeling Node

1. Add the Interactive Modeling node to the pipeline.

    a. Select **Nodes** (to the upper left of the pipeline diagram).

    b. Expand **Postprocessing**.

    c. Left-click and then drag and drop the **Interactive Modeling** node on the Auto-Forecasting node.

    d. Right-click the **Interactive Modeling** Node and run it.

2. Open the Interactive Modeling node and explore the Forecast tab functionality.

    a. Right-click the **Interactive Modeling** node and open it.

       Note: There are three tabs: Forecast, Modeling, and Series Analysis. It should open showing the Forecast tab results, but if it doesn't, click the Forecasting tab.

    b. Expand **Default** attributes.

    c. Select **Product01**. Note that you may need to click the individual time series in the Series pane to have the image appear.

       Note: The functionality on the Forecast tab is the equivalent of the Forecast tab in the Forecast Viewer. Recall that for **Product01**, there is evidence of promotion or Price effects, but no discernable seasonality or trend.

3. Explore the Modeling tab functionality.

    a. Click the Modeling tab. We've seen three of the models shown in the Forecast Viewer, but the champion seems to have changed into a **PREDECESSOR** model, and its type is Inherited.

       Note: The IM node is running another Auto-Forecasting node with default settings 'under the hood'. In this case, the models shown (the Model Selection list for Product01) are identical to the three we've seen in the Forecast Viewer plus the Champion model from the predecessor Modeling node. Because there are two equivalent Auto-Forecasting runs with default settings in the Pipeline, the **PREDECESSOR** Champion model and the **DIAG1_REGARIMA** model are the same. Note the in-sample MAPE values. This can change when the Interactive Modeling node is attached to an Auto-Forecasting node whose default setting has been changed or when it's attached to a different type of Modeling node.

b. Select the **DIAG1_REGARIMA1** model.

c. Click the **View diagnostic plot/table** button.

d. Select **Model Fit** > **Parameter Estimates**.

e. Select the **DIAG1_ARIMAX1** model. The Parameter Estimates table changes to show results for the selected model. It has two estimated parameters.

f. Select the **View diagnostic plot/table** button > **Model Fit** > **Statistics of Fit**. This table shows fit statistics associated with the selected model. Scroll over in the table to view the available fit measures.

   Note: In addition to MAPE, there are over 50 reported statistics of fit.

g. Select the **DIAG1_REGARIMA1** model.

h. Select the **View diagnostic plot/table** button > **Basic error analysis** > **Prediction Errors**.

   Note: There are a couple of large residuals that lie outside the Two Standard Error band. These might warrant further investigation.

i. Select the **View diagnostic plot/table** button > **Error autocorrelation analysis** > **White noise probability test (log scale)**.

   Note: Students who are comfortable with ARIMA model identification will find familiar plots like the ACF, PACF, and IACF for the residuals of the model among the diagnostics listed. We'll find some of these same diagnostics on the Series Analysis tab, but there the generated ACF, PACF, and IACF are based on the time series. The white noise test indicates that the model's residuals are not white noise. Some spikes exceed the 0.05 threshold line. This tells us that it might be useful to add terms to the model.

4. Create a custom model specification based on a generated specification.

   a. Click the **Model Details** button in the top right of the Model list. The Model Details have similar information to the Parameter Estimates table plus information about differencing and the transfer function specification for inputs.

   b. Close the pop-up window.

   c. Click the **Copy** icon.

      Note: Generated models cannot be directly edited. First, a copy of the model is made and then the copy can be edited.

   d. Name the model **myREGARIMA1_1**.

e. Select **Independent Variables**.

f. Select **Edit** (pencil) next to the pre-selected input variable.

g. Change Simple Numerator Factors from 0 to **1**.

Note: We've changed the way that the input variable enters the model with a Numerator order 1 term. This specifies that when the input variable jumps in a given month, Sales are impacted in that month and in the month following.

h. Select **Save** > **Save**.

Note: The Custom model myREGARIMA1 has the best MAPE of the models listed, but it is not declared the Champion model for Product01. It can be made the Champion using the following steps:

I. Make sure **myREGARIMA1_1** is selected. Right-click and choose **Set as champion**.

II. Note the Pending Changes alert in the upper right corner.

III. Select **Save**. You can select which series to save changes to or not. Click **Save** again to apply the changes.

IV. Click the Forecast tab.

Note: The forecasts shown are generated by the new champion, **myREGARIMA1_1** custom specification.

5. Look at Series Analysis diagnostics.

a. Click the **Series Analysis** tab.

Note: A Series Analysis is pre-loaded for the dependent variable, Sale, for the selected series, Line01:Product01. There are three tabs on the left of the main window: Filters, Model Inputs, and Analyses. It's opened in Series Analysis by default.

b. Click the **Add analysis** (plus sign) button for the Sale variable.

Note: In addition to standard diagnostics like the ACF, PACF, and White Noise test, available analyses include decompositions and seasonal adjustments.

c. Scroll through the pop-up list to see the available diagnostics for the series.

Note: Additional series can be analyzed by dragging and dropping them from the Model Inputs column into the Series Analysis diagram.

6. Create a new custom model.

    a. Click the **Modeling** tab.

    b. Click the **Create Model** icon.

    c. Close the Interactive Modeling Node.

    Note: Selecting one of the listed families of models provides a point-and-click interface for creating a new custom model and adding it to the model selection list. This is a way to create a custom model that is not based on an existing Generated model.

Background Information

SAS Landing and the Applications Menu

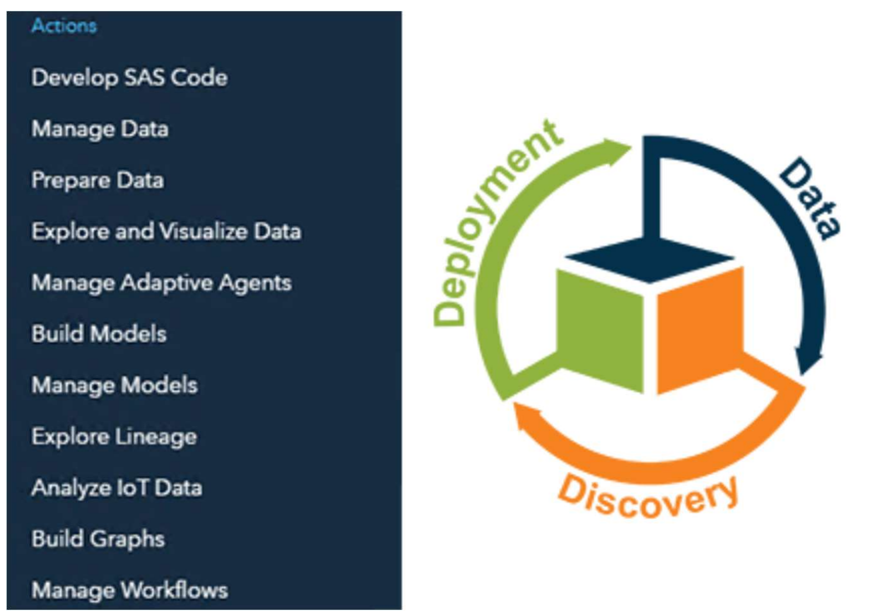
SAS Landing

SAS Landing is a common interface for SAS Viya applications, and it enables you to easily view, organize, and share your content from one place. From SAS Landing, you access the features of SAS Visual Forecasting through Model Studio.

Applications Menu

The options on the Applications menu are actions that fall within the three phases of the analytics life cycle (Data, Discovery, and Deployment). For example, to access Model Studio from the Applications menu, select **Build Models**. Building models relates to the Discovery phase.

Model Studio, included in SAS Viya, is an integrated visual environment that provides a suite of analytic data mining tools to facilitate end-to-end data mining analysis. The data mining tools supported in Model Studio are designed to take advantage of the SAS Viya programming and cloud processing environments to deliver and distribute analytic model data mining champion models, score code, and results.



Here are other examples:

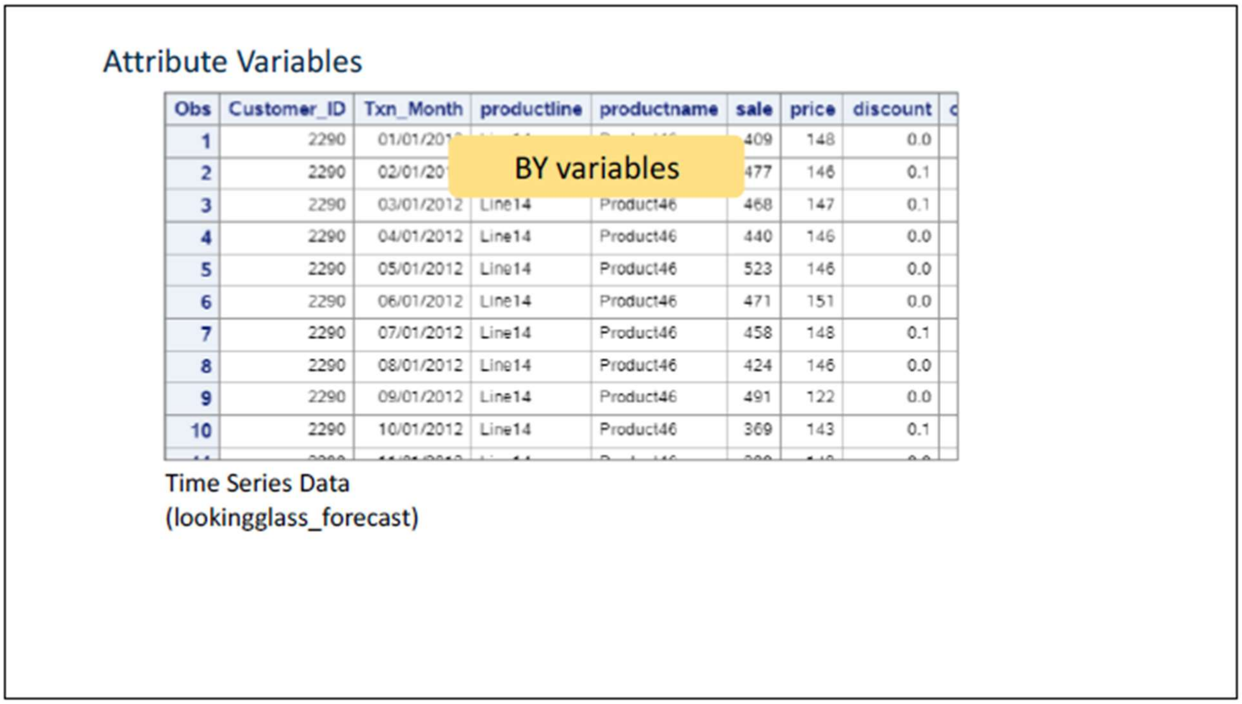
• To access SAS Model Manager, select **Manage Models.**

• To access SAS Visual Analytics, select **Explore and Visualize Data**. From SAS Visual Analytics, you can access the SAS Visual Statistics add-on functionality, which enables you to use pipelines. In this course, you do not use SAS Visual Analytics and SAS Visual Statistics.

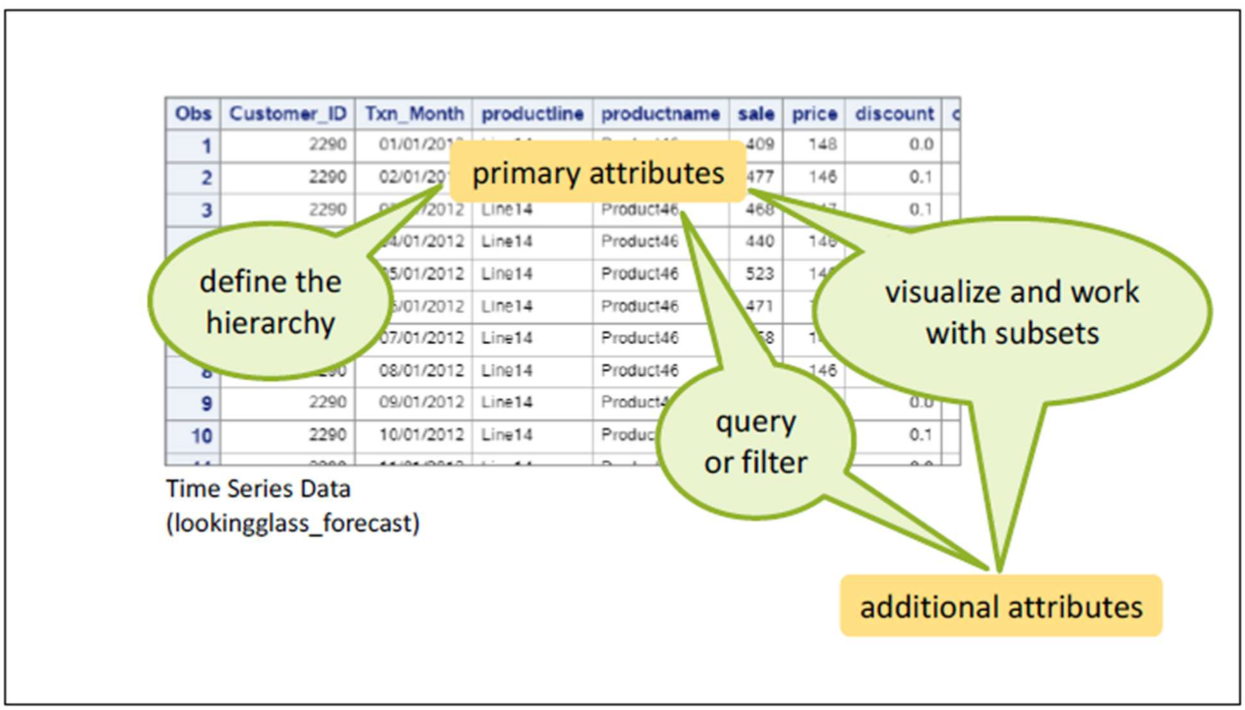
From other applications, you can use the Applications menu to return to SAS Drive.

Note: Remember that access to specific applications is determined by the permissions that are associated with your account.

Attribute Variables



When we created the baseline sales forecast project, we defined two variables in the time series data as BY variables: **productline** and then **productname**.

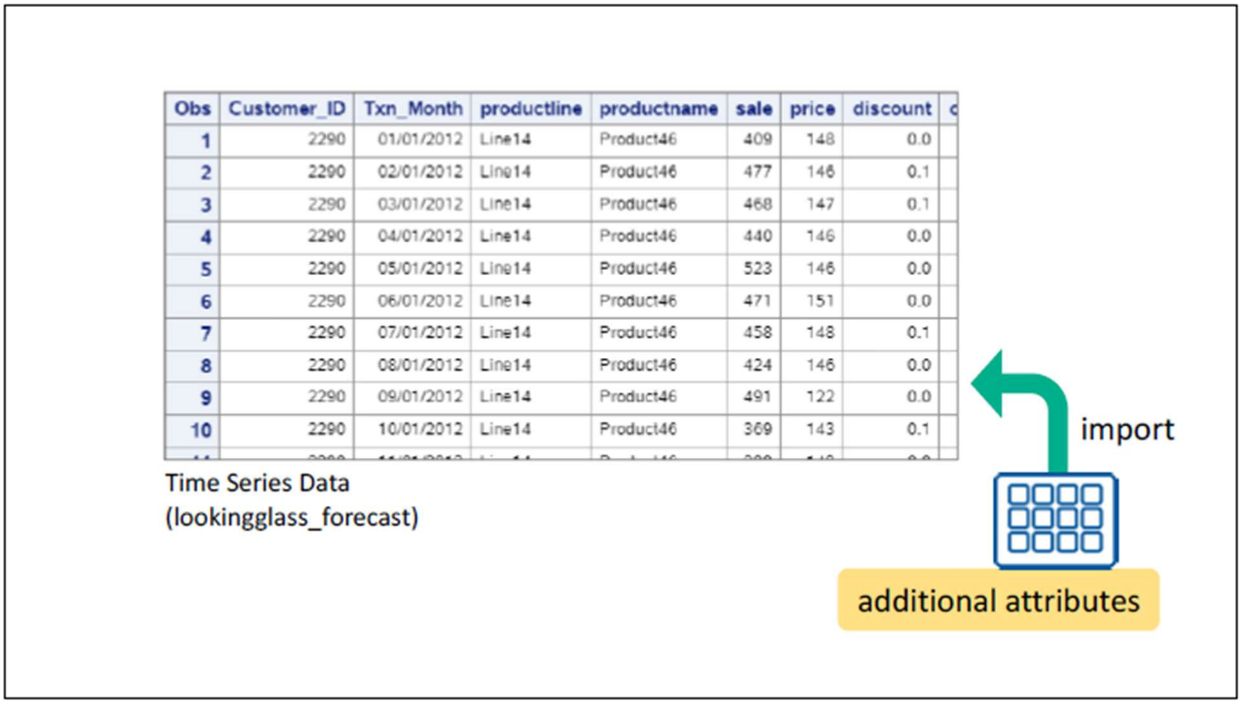


BY variables are the primary attributes of a project.

SAS Visual Forecasting uses the primary attributes to define the hierarchical structure of the time series.

A forecasting project can also have additional attributes.

All attributes are characteristics that you can use to query or filter specific subsets of your time series. You can visualize and work with subsets of project data based on specific values of the attributes. Attributes also affect how you can apply overrides or perform other post-modeling tasks.



Obs	Customer_ID	Txn_Month	productline	productname	sale	price	discount
1	2290	01/01/2012	Line14	Product46	409	148	0.0
2	2290	02/01/2012	Line14	Product46	477	146	0.1
3	2290	03/01/2012	Line14	Product46	468	147	0.1
4	2290	04/01/2012	Line14	Product46	440	146	0.0
5	2290	05/01/2012	Line14	Product46	523	146	0.0
6	2290	06/01/2012	Line14	Product46	471	151	0.0
7	2290	07/01/2012	Line14	Product46	458	148	0.1
8	2290	08/01/2012	Line14	Product46	424	146	0.0
9	2290	09/01/2012	Line14	Product46	491	122	0.0
10	2290	10/01/2012	Line14	Product46	369	143	0.1

It is a common practice to import additional attributes to your forecasting project from another data source.

Obs	Customer_ID	Txn_Month	productline	productname	sale	price	discount
1	2290	01/01/2012	Line14	Product46	409	148	0.0
2	2290	02/01/2012	Line14	Product46	477	146	0.1
3	2290	03/01/2012	Line				
4	2290	04/01/2012	Line				
5	2290	05/01/2012	Line				
6	2290	06/01/2012	Line				
7	2290	07/01/2012	Line				
8	2290	08/01/2012	Line				
9	2290	09/01/2012	Line				
10	2290	10/01/2012	Line				

Obs	productline	productname	Cust_Region	margin_cat
1	Line01	Product01	South	LOW
2	Line01	Product02	South	MED
3	Line01	Product03	Great Lakes	LOW
4	Line02	Product04	Greater Texas	LOW
5	Line02	Product05	Greater Texas	LOW
6	Line02	Product06	South	LOW
7	Line02	Product07	Pacific	LOW
8	Line03	Product08	Pacific	LOW
9	Line03	Product09	Greater Texas	HIG
10	Line03	Product10	Southwest	LOW

For the baseline sales forecast project, we have an additional data set, named **lg_attributes**, that contains additional attributes, **Customer Region** and **Margin Category**.

Imported attributes data must contain a row for each unique combination of the primary attributes, or BY variables, that are defined in the time series data.

Loading an Attributes Table to Subset the Time Series

A unique and useful feature in SAS Visual Forecasting is the ability to visualize the modeling data and operate on generated forecasts outside the hierarchy defined by the project's BY variables. The hierarchical arrangement of the modeling data for this project is defined by product characteristics. However, it is routinely useful to be able to explore and operate on forecasts across facets of the data such as customer demographics or geographic regions.

In the last demonstration, we created a project and added data. The only attributes defined were the BY variables. Now we'd like to add other attributes to subset the time series analyses.

In this demonstration, you incorporate the **LG_ATTRIBUTES** table into the **baseline sales forecasts** project and then use the variables in the table to expand the ways that the modeling data can be visualized.

1. From the Data tab, change the data source type from **Time Series** to **Attributes** by navigating to the data sources panel, selecting the **New data source menu** and then selecting **Attributes**.

   The attributes data set is not yet here in memory. So once again, I need to import it.

   Note: A default attributes table is created when the BY variables are assigned in the project. The BY variables that define the modeling hierarchy are primary attributes for the project.

2. Make sure the Show area says **Available** to view in-memory tables. Scroll and find the **lg_attributes** in-memory table. Alternatively, you can click on the search bar and type lg.
3. Select **lg_attributes** data set. Then click **ADD**.

   The in-memory table, **LG_ATTRIBUTES**, is now the attributes table for the project. The first two attributes are the by variable that I selected earlier, productline, and productname. This table contains two new attributes: a geographic indicator, **Cust_Region**, and a margin flag, **margin_cat**. The margin flag categorizes the profitability of product names as *LOW*, *MED*, or *HIG* (high).
4. Switch to the **Pipelines** tab by selecting it.

   This first pipeline includes a Data node, Auto-forecasting, Model Comparison, and Output.
5. Right-click and run the **Data** node.
6. After the **Data** node runs (you will see a green circle with a check mark inside), right-click the green checkmark and select **Time series viewer**.

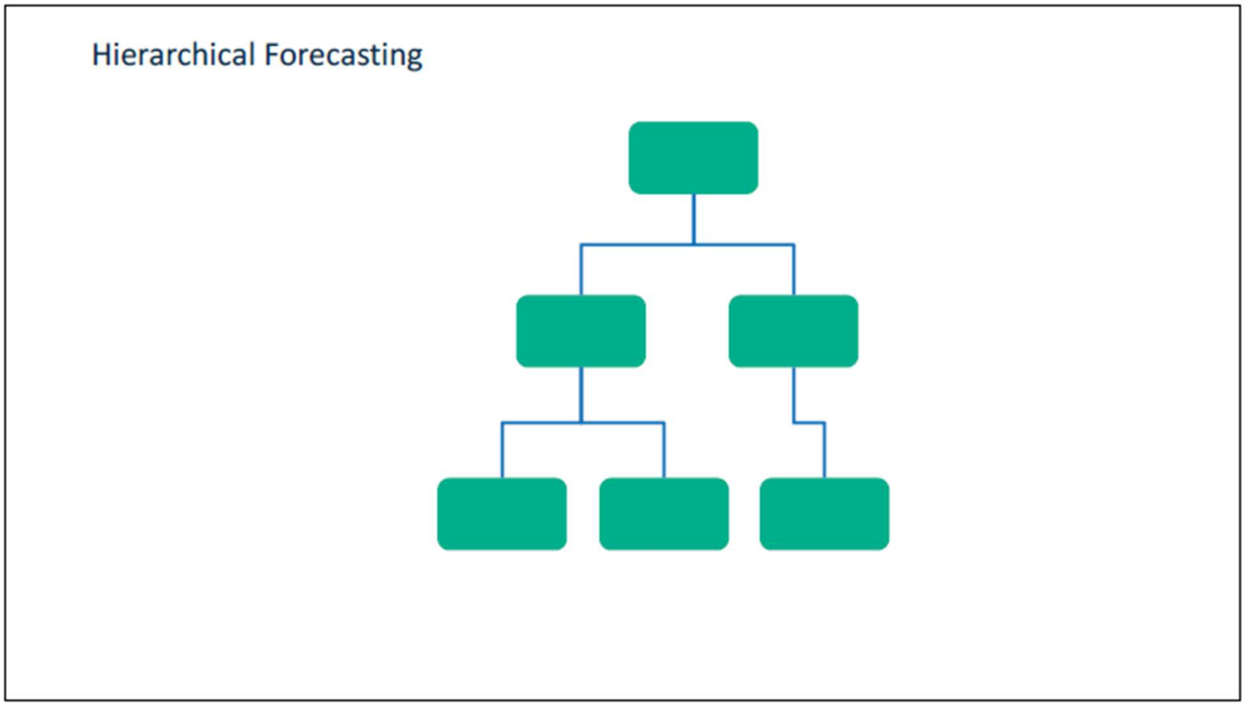
   The envelope plot shows the aggregated data at the top level of the hierarchy (918 of 918 series). The colored bands illustrate one and two standard deviations around the aggregated series. The available attribute variables are listed in the left filters panel.
7. The available attribute variables are listed on the left side of the window: **product line**, **Product Name**, **Cust_Region**, and **margin_cat**. You can explore time series in the middle level of the hierarchy by expanding the product line attribute. By default, the product line attribute should already be expanded. Visualize demand for the product line series, Line07. Under the **productline** attribute, select **Line07**.

   The plot changes on the fly to show an aggregation of the four product names contained in Line07: Product 21, Product 22, Product 23, and Product 24. Notice that the Envelope Plot changes because it is now relevant for only the four product lines in Line07.
8. Expand the **Cust_Region** attribute.

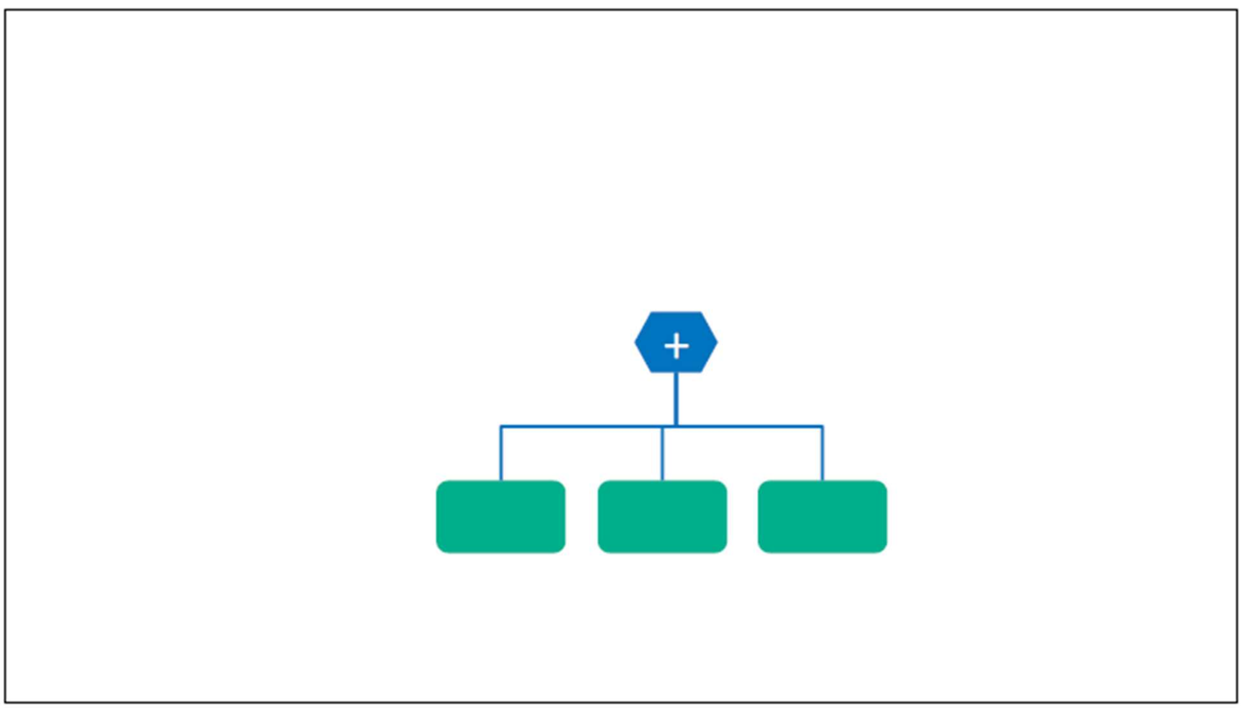
   There are two customer regions in Line 07. Those are **Pacific** and **Greater Texas**, three in **Pacific** and one in **Greater Texas**.
9. Selecting **Greater Texas** plots the one product name that flows through both Line07 and the Greater Texas region, product line 24.

   Note: You might need to click the series in the right pane for the image to appear.
10. You can select **Reset All** to remove the filters that you created based on attributes and return to 918 series displayed.
11. Close the Time Series Viewer.

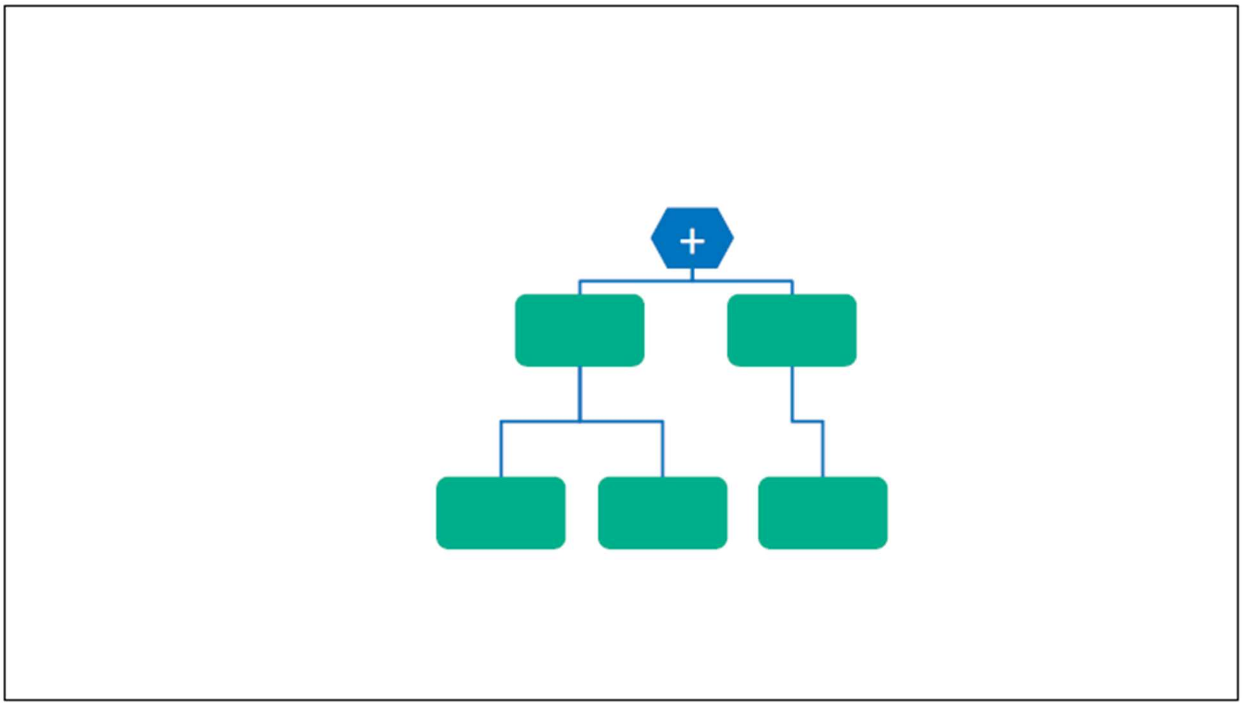
Hierarchical Forecasting



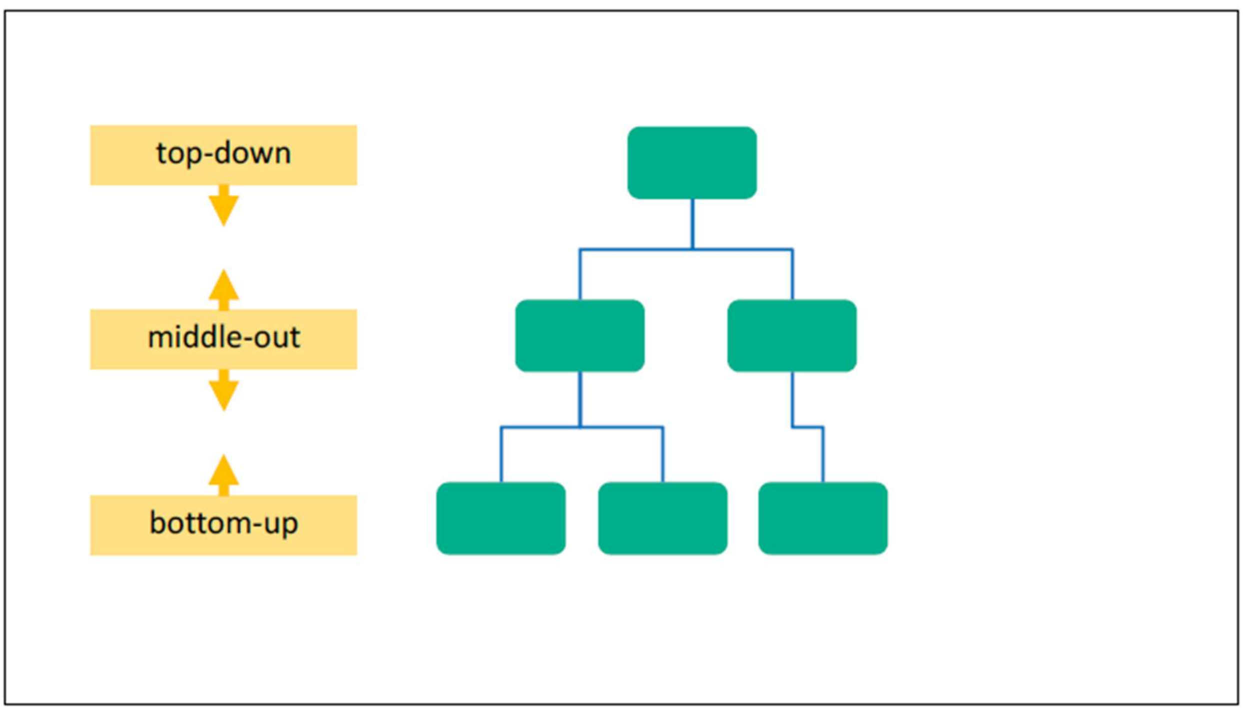
Modeling data with a hierarchical structure adds complexity and tasks to the forecasting process.



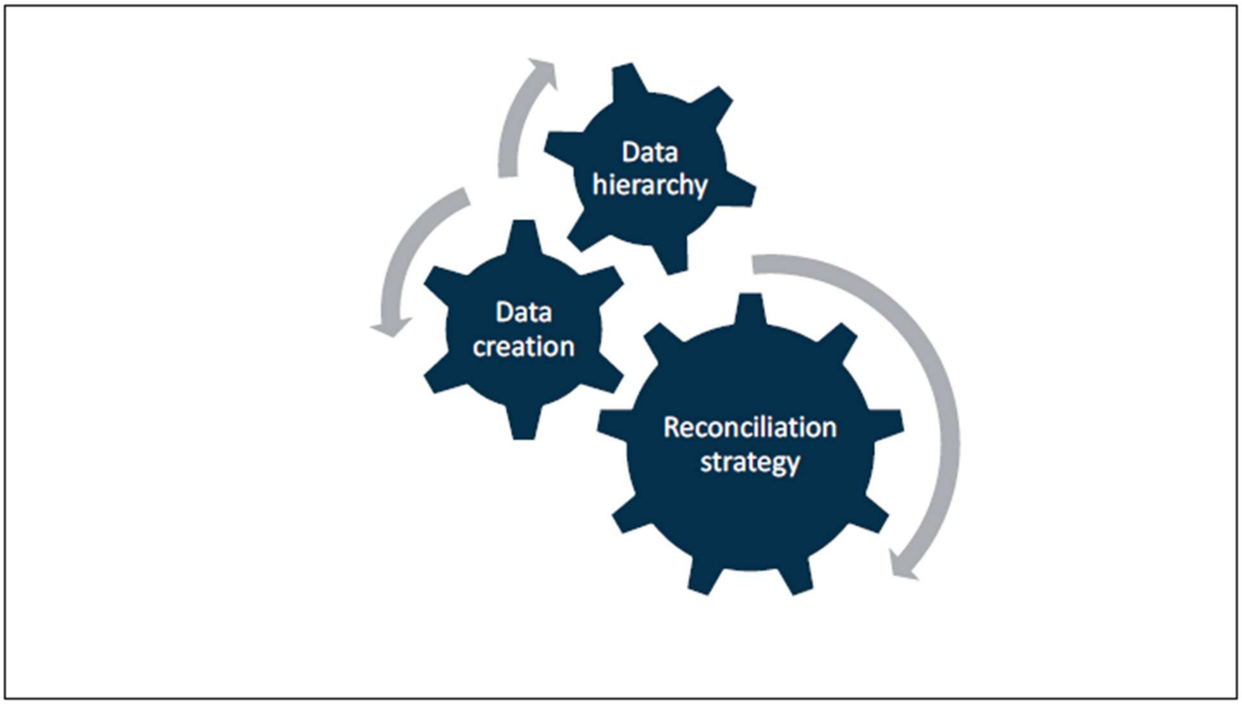
On the data preparation end, each transactional series in the base level of the hierarchy needs to be accumulated to an equally spaced interval.



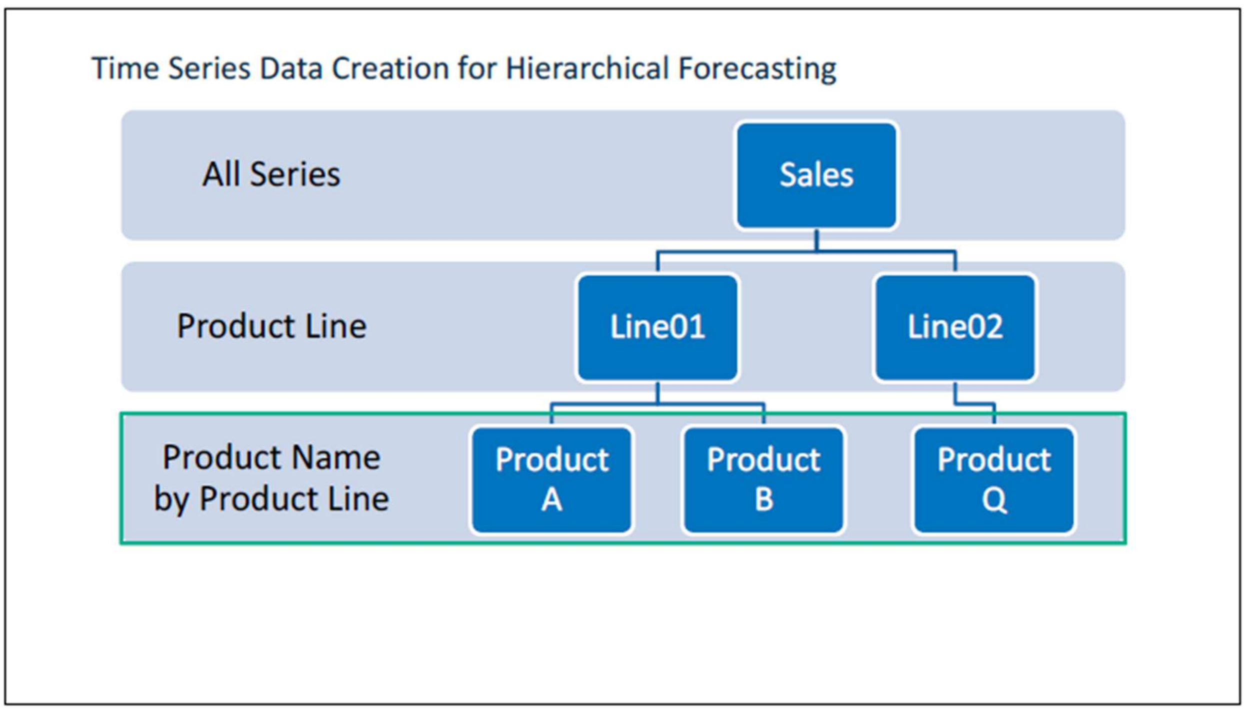
After this, time series data in the upper levels of the hierarchy are constructed from the base level data through the process of aggregation.



Usually, forecasters analyzing hierarchical data want their statistical forecasts to reconcile. The process of reconciliation is based on the choice of reconciliation type: bottom-up, middle-out, or top-down. Given a reconciliation type, forecast reconciliation is performed using the method of forecast proportions by default. We will explore reconciliation options available in the software.



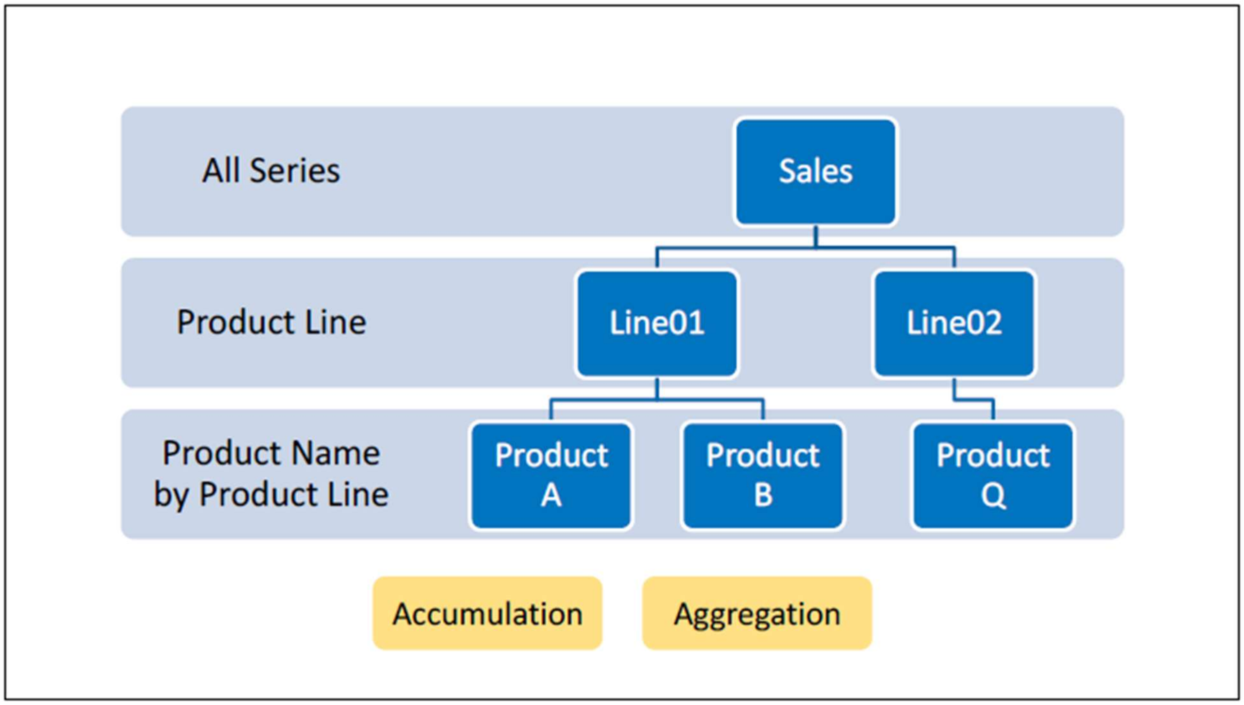
Finally, choices related to which products to include in the data hierarchy, data creation, and the implemented reconciliation strategy interact. These interactions have an impact on a project's forecasting performance.



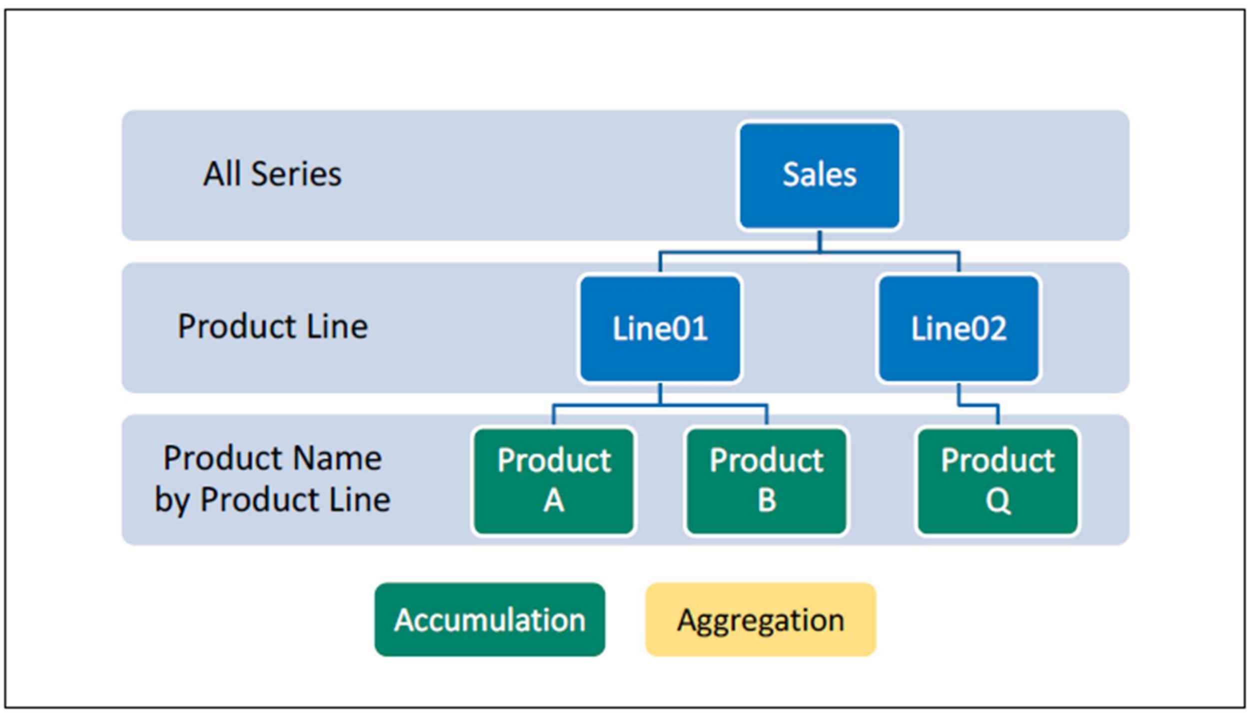
Recall that when you created the baseline sales forecast project, you defined two variables in the time series data as BY variables, **productline** and **productname**. At that time, we mentioned that SAS Visual Forecasting uses these primary attributes to define the hierarchical structure of the time series.

You included additional attribute variables, but they didn't define the hierarchy.

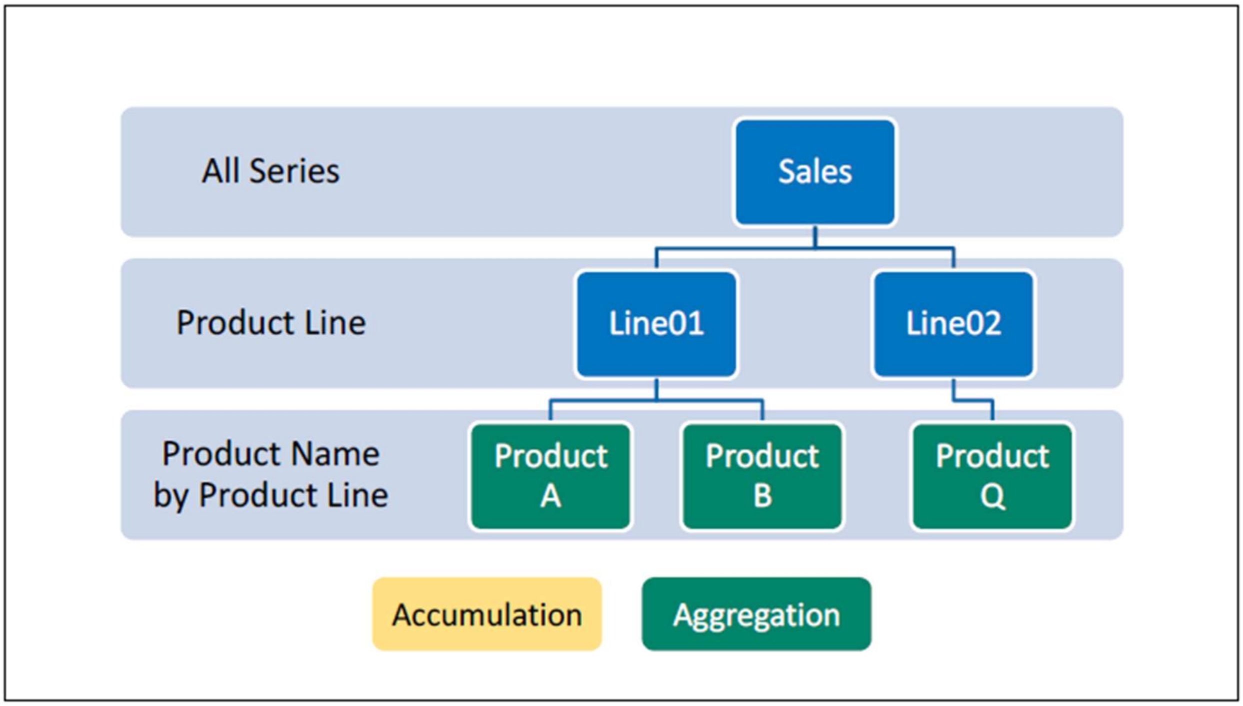
So far, you have seen results only at the base level of the hierarchy, where the series is broken down by **productline** and **productname** within **productline** . Let's discuss true hierarchical forecasting.



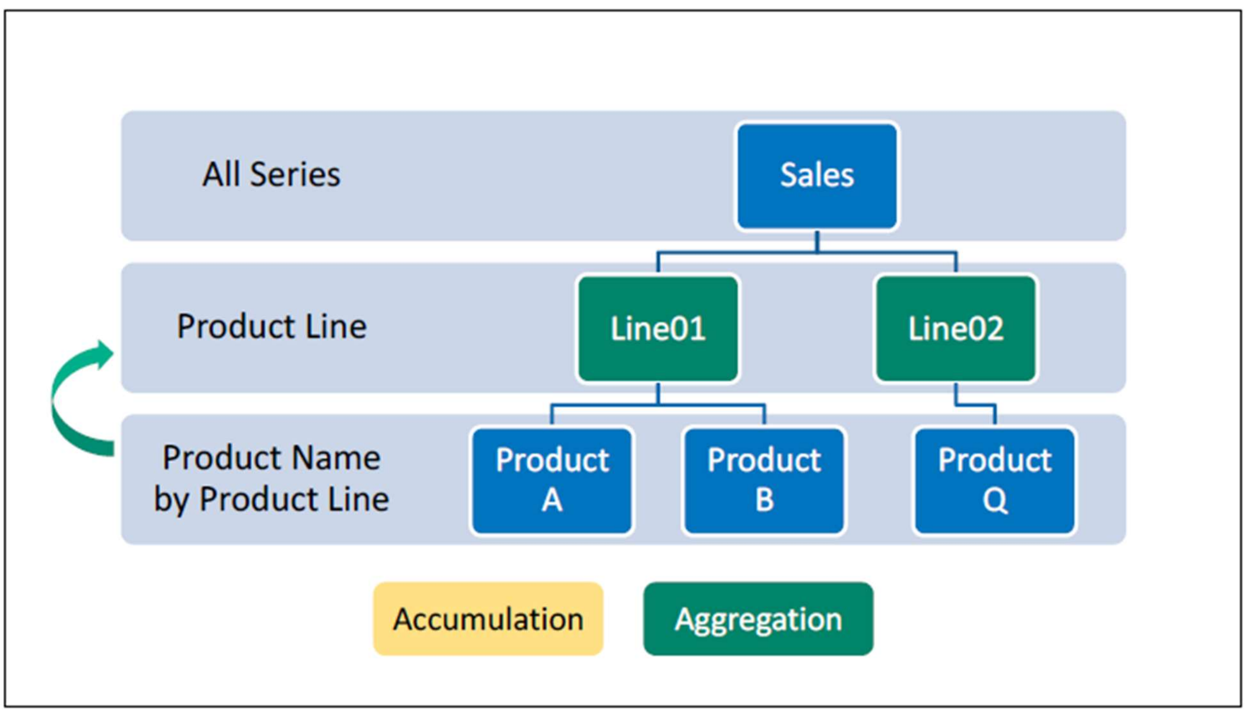
There are two options for creating time series data from transactional data in Model Studio: accumulation and aggregation.



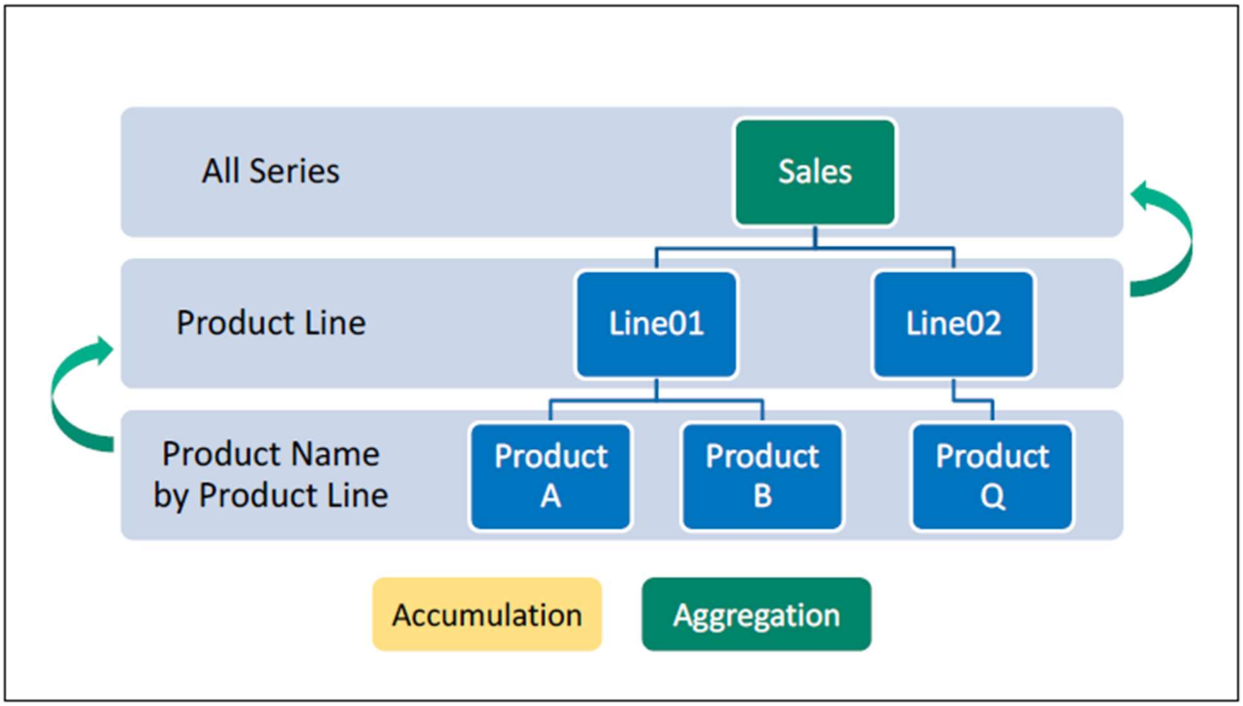
You already learned about accumulation. Data accumulation produces the time series at the bottom level of the hierarchy from transactional series in the data.



Data aggregation constructs the data hierarchy by aggregating the time series in the bottom level of the hierarchy.

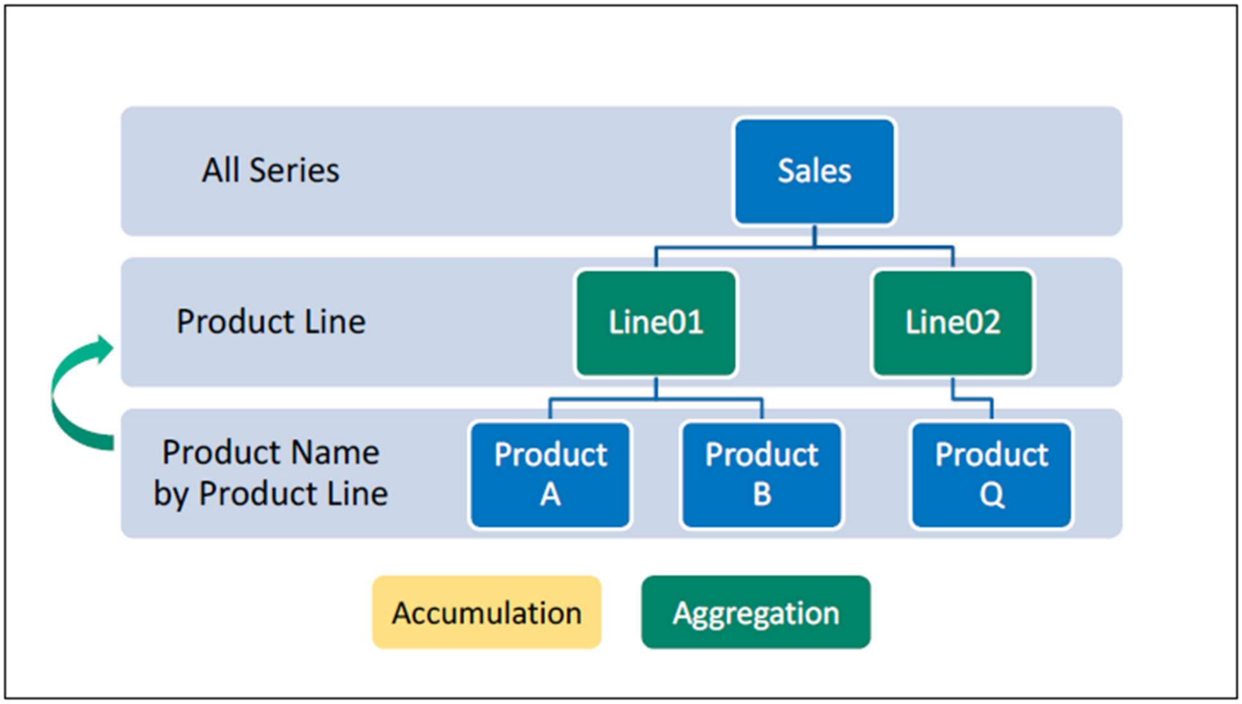


After the time series are created at the bottom level, you can create the middle level of the hierarchy by summarizing the series at the base level.

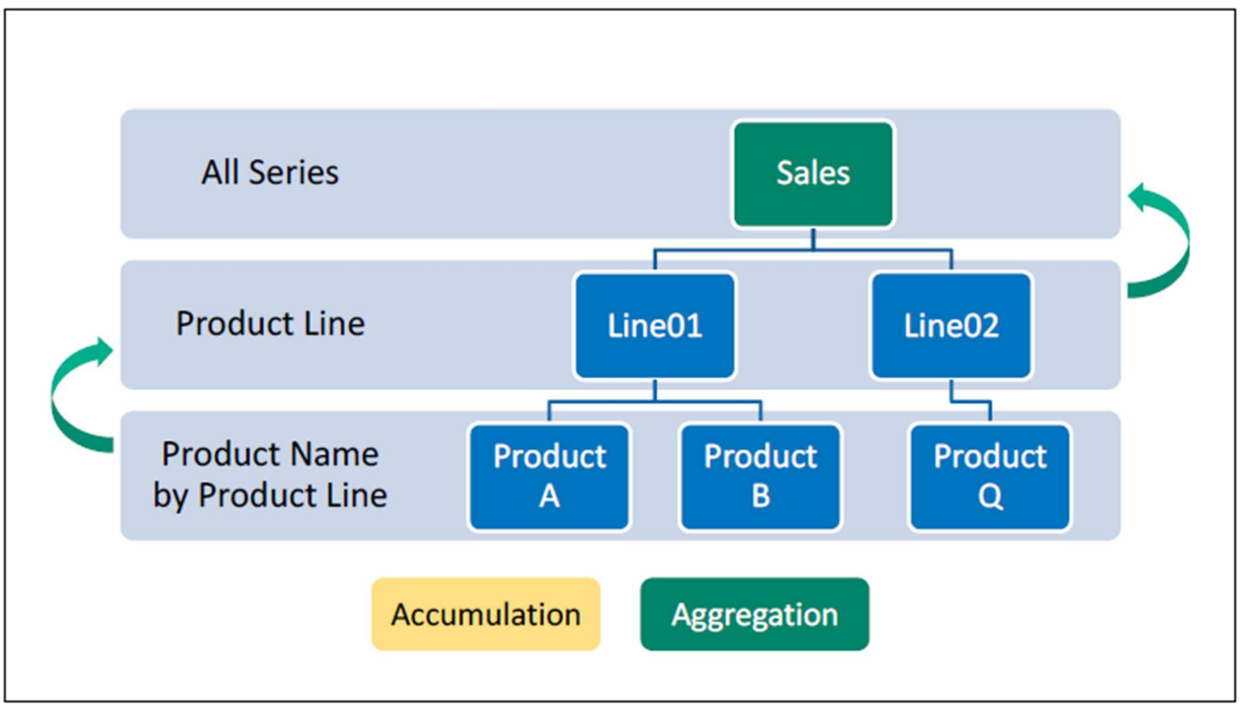


After the middle level series are created, you can create the top-level series by summarizing the series at the middle level.

Aggregation produces time series in the middle and top levels of the hierarchy according to BY groups in the data. A statistical forecast is then produced for every series in every level of the data hierarchy.

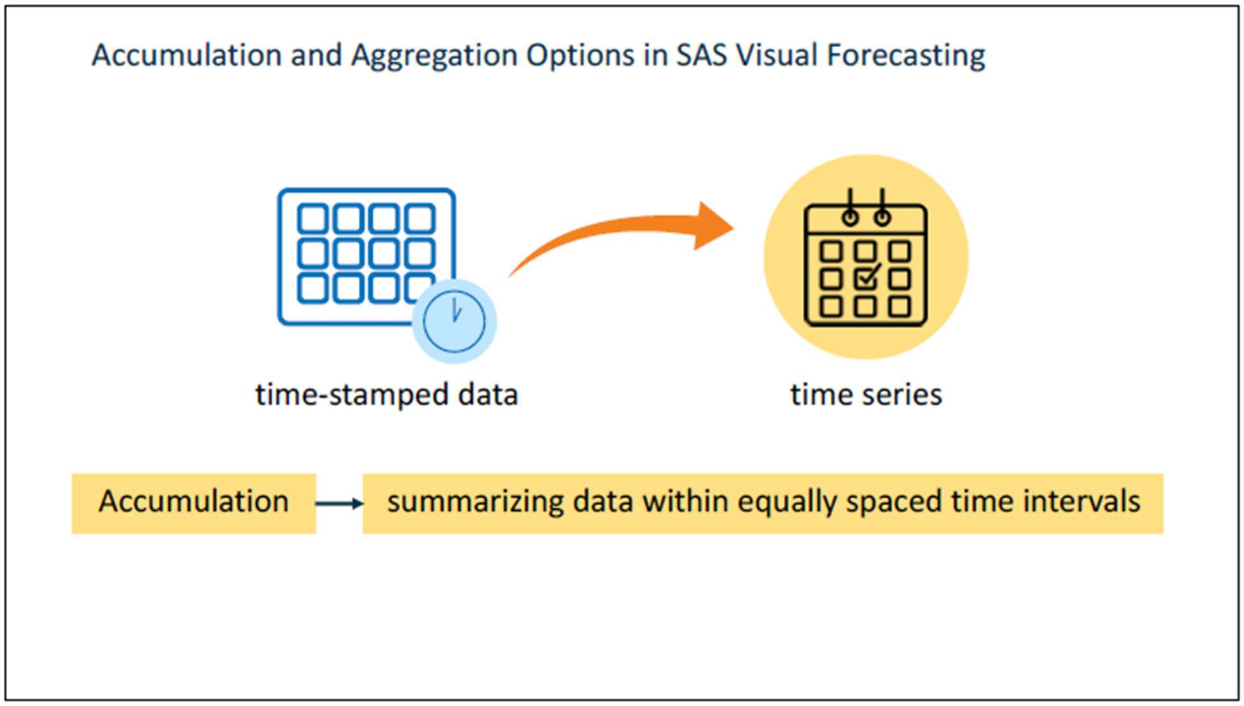


In our example, aggregation of the base level results produces product line series in the middle level of the hierarchy …

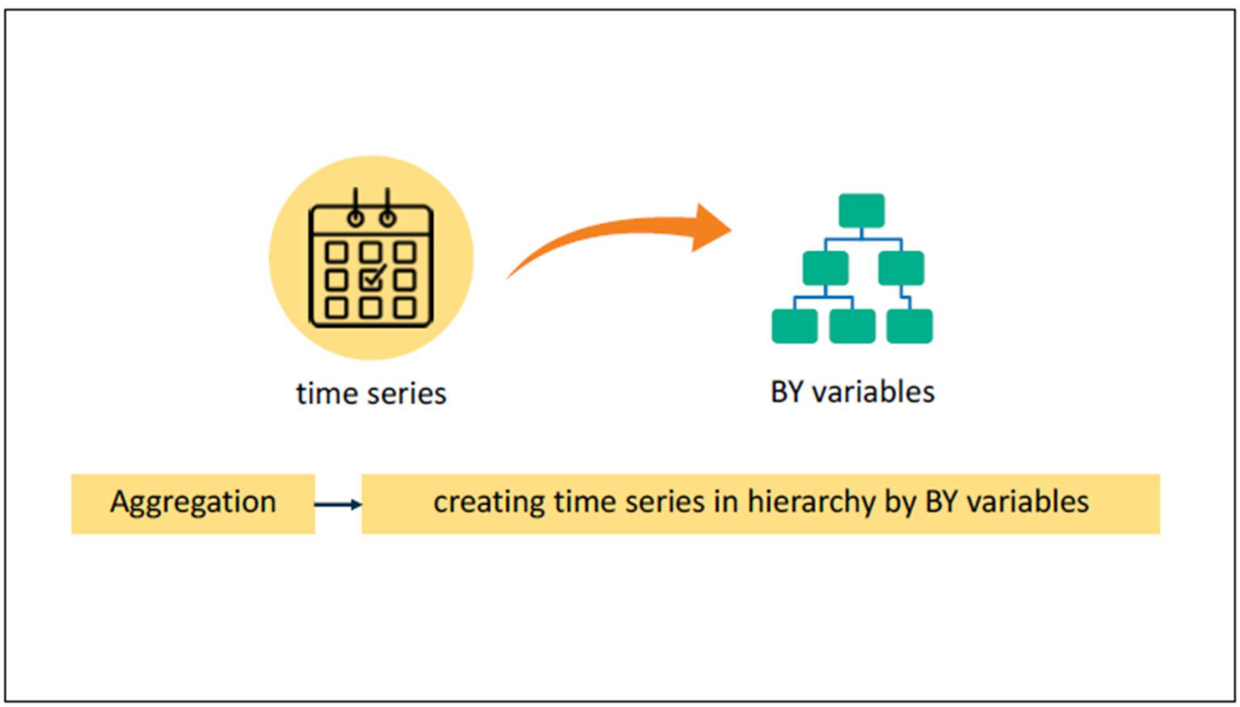


… and total sales in the top level of the hierarchy.

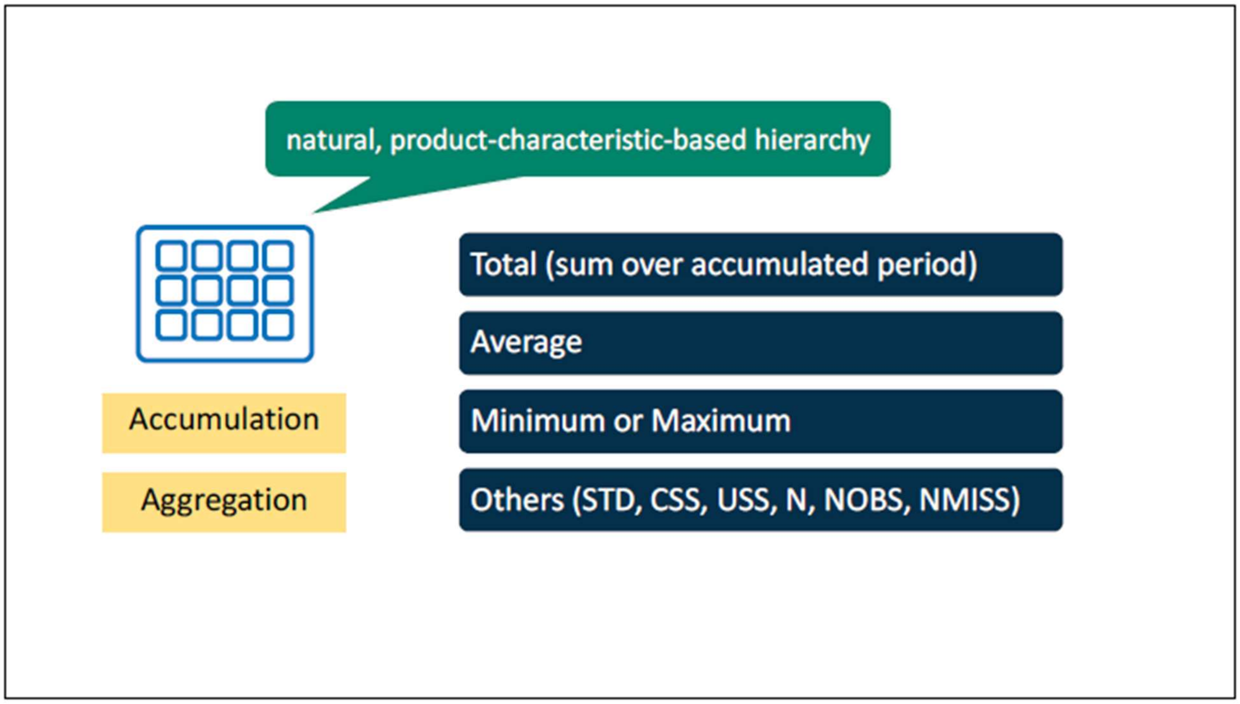
The sales for each month across all the different BY groups are summed.



Accumulation is the process of taking timestamped data and rolling them up into time series by summarizing data within equally spaced time intervals.



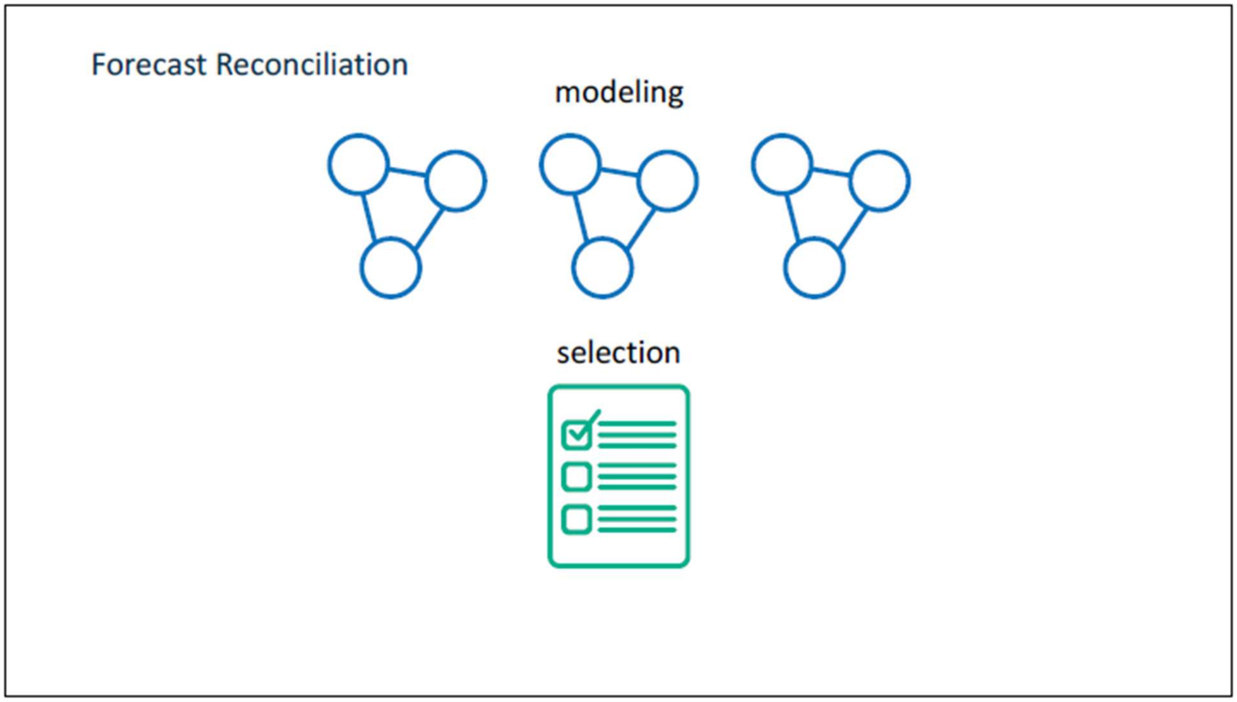
Aggregation is the process of summarizing the time series themselves in a hierarchy defined by BY variables. Accumulation and aggregation are processes that create the series, so they must occur before any modeling is performed.



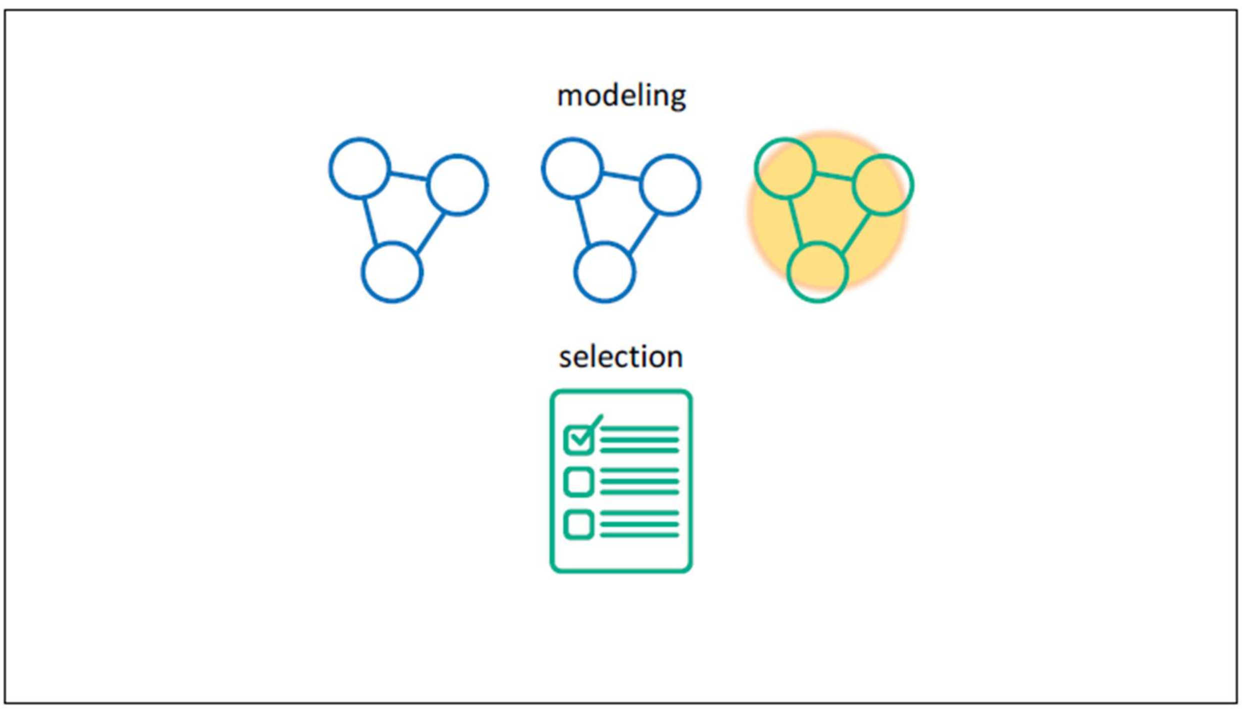
Although accumulation and aggregation options are identical, there are situations in which you will want to choose different options to operate on different levels of the hierarchy. The data used in this lesson's demonstration have a natural, product-characteristic-based hierarchy.

Total or sum is appropriate for the target in the case of demand. If your target is price or cost, there are times when average would be a better representation.

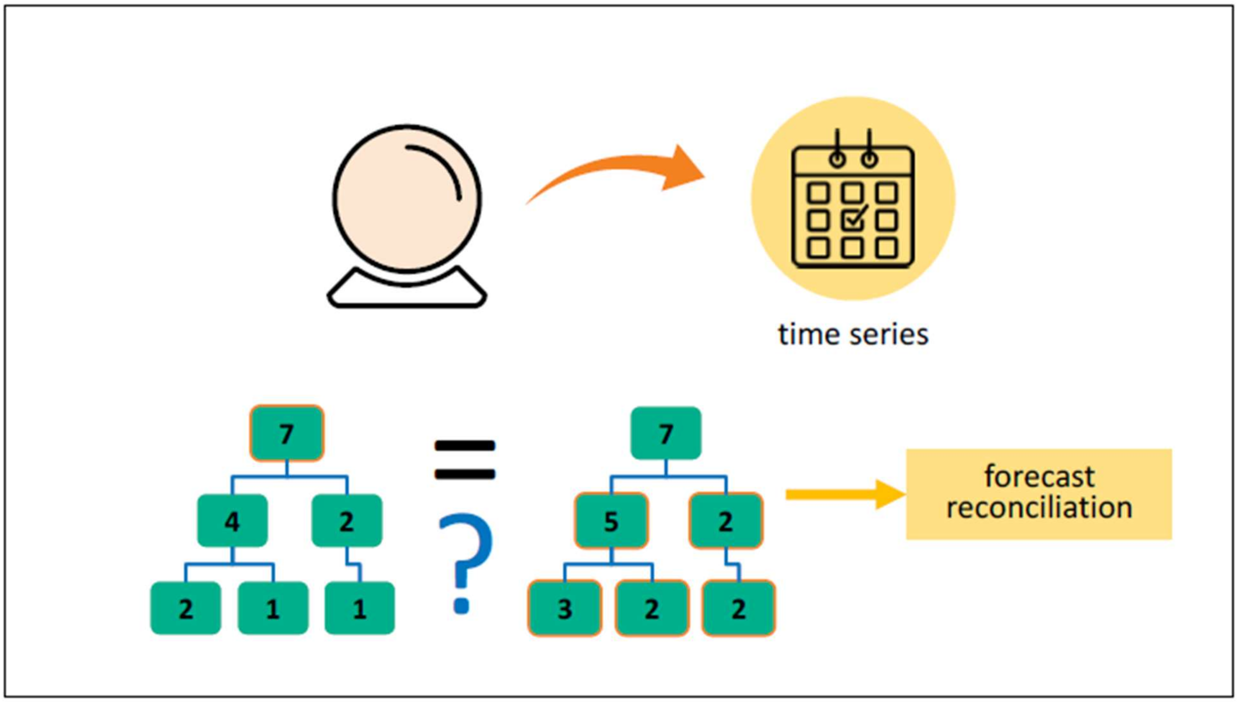
In instances where you have disproportionate costs, you might want to summarize according to minimum or maximum. This method builds in a safety measure by forcing you to over-forecast or under-forecast on average.



After the series are created with accumulation or aggregation, modeling occurs, and many of the models that you've learned about are tested.



The champion models are selected for each series using honest assessment on a holdout sample, if possible, or accuracy on the fit sample if data splitting isn't feasible.

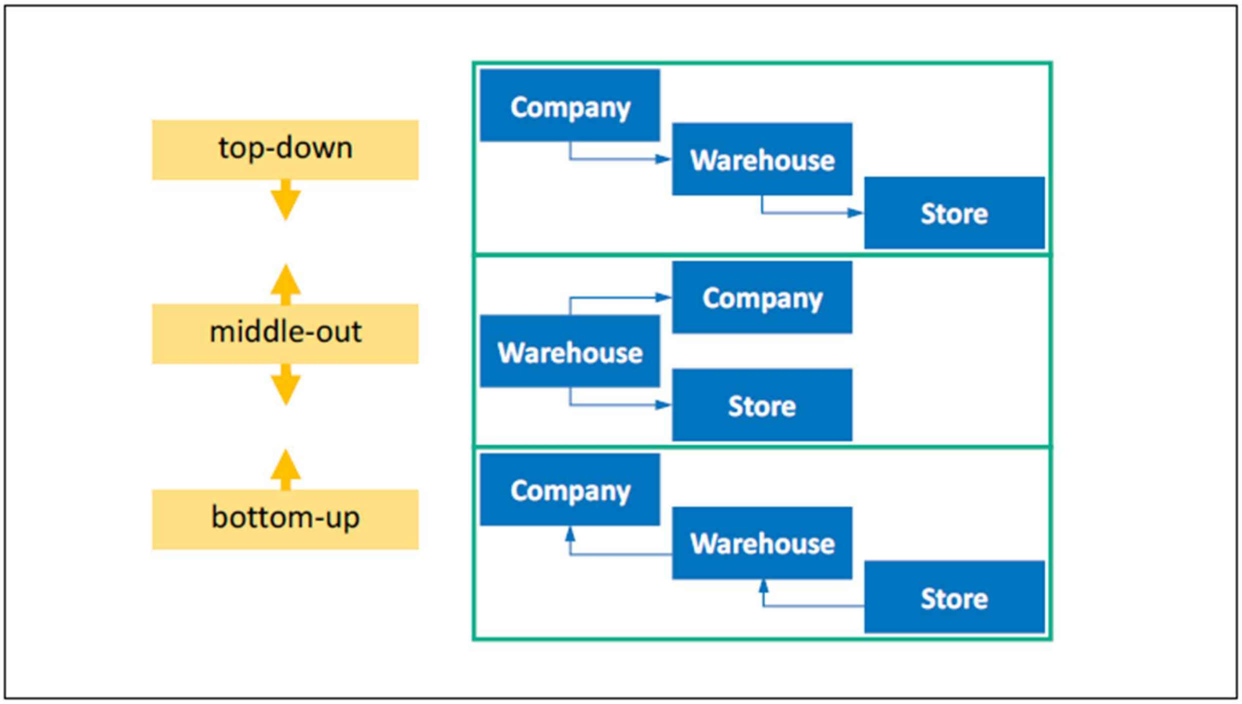


At this point, forecasts for the future are generated for all series.

However, there is no guarantee that the aggregation of forecasts from the lower to the upper levels of the hierarchy will match the actual forecasts from those upper levels.

Inconsistencies must be resolved before you can deploy the models.

The process of resolving the inconsistencies between levels of a hierarchy is called *forecast reconciliation*. In contrast to data creation, forecast reconciliation occurs at the other end of the forecasting process.

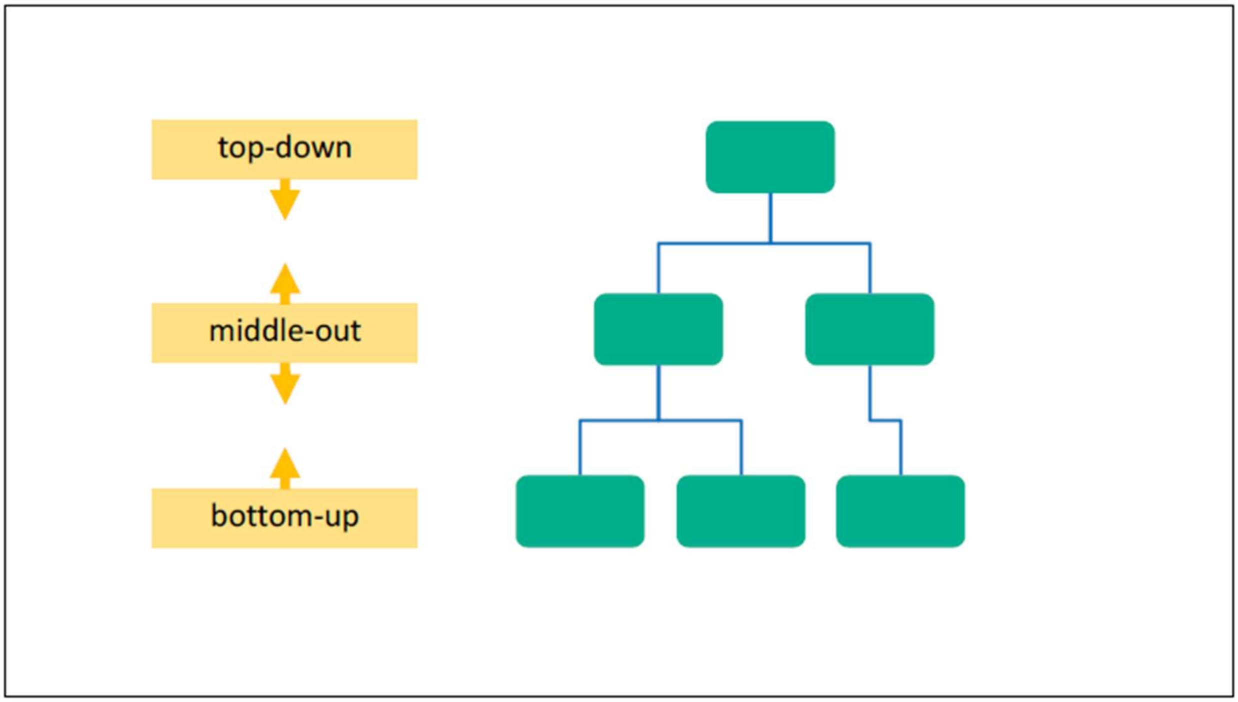


The three main reconciliation approaches are top-down, middle-out, and bottom-up. These approaches are straightforward.

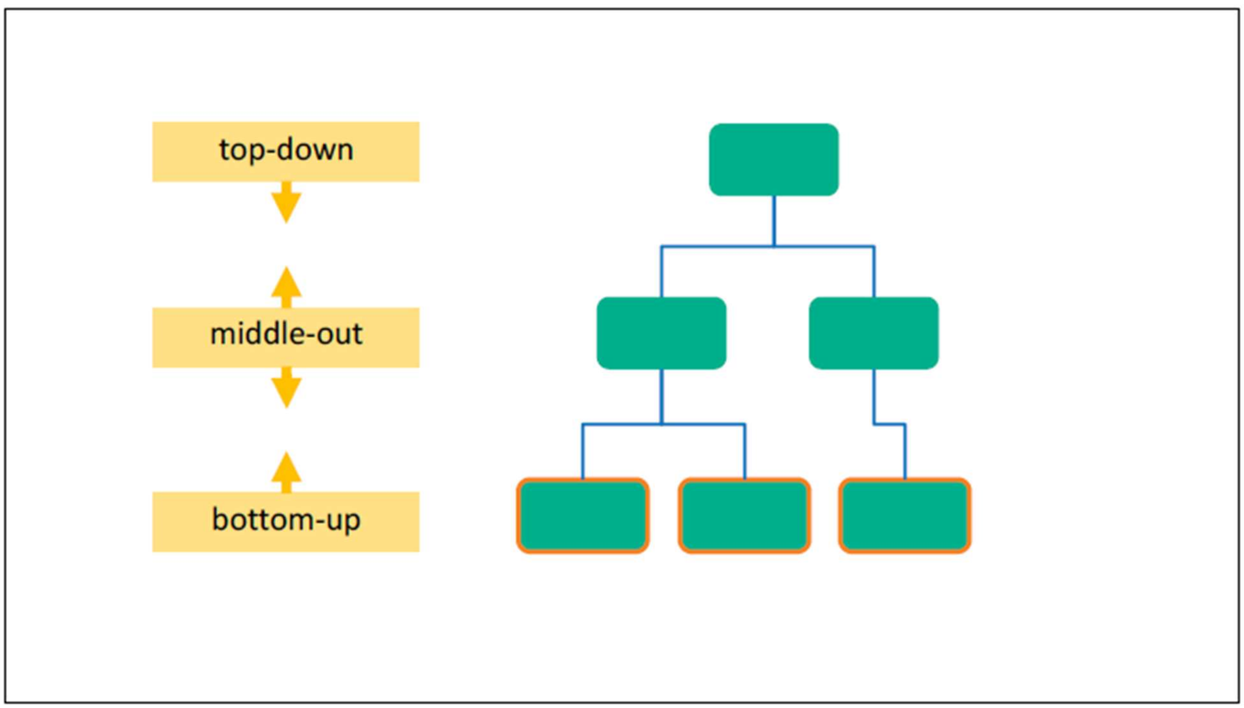
The top-down approach treats the statistical forecast at the top level as unchangeable. We adjust the middle-level forecasts to accommodate the top-level forecast and then modify the bottom-level forecast to be consistent with the middle-level forecast.

When you use the middle-out approach, you select the middle-level forecast to be immutable. You resolve any inconsistency with the other levels of the hierarchy by adjusting those levels. There might be more than one middle level of the hierarchy. In this case, you specify which of the middle levels is held constant for reconciliation.

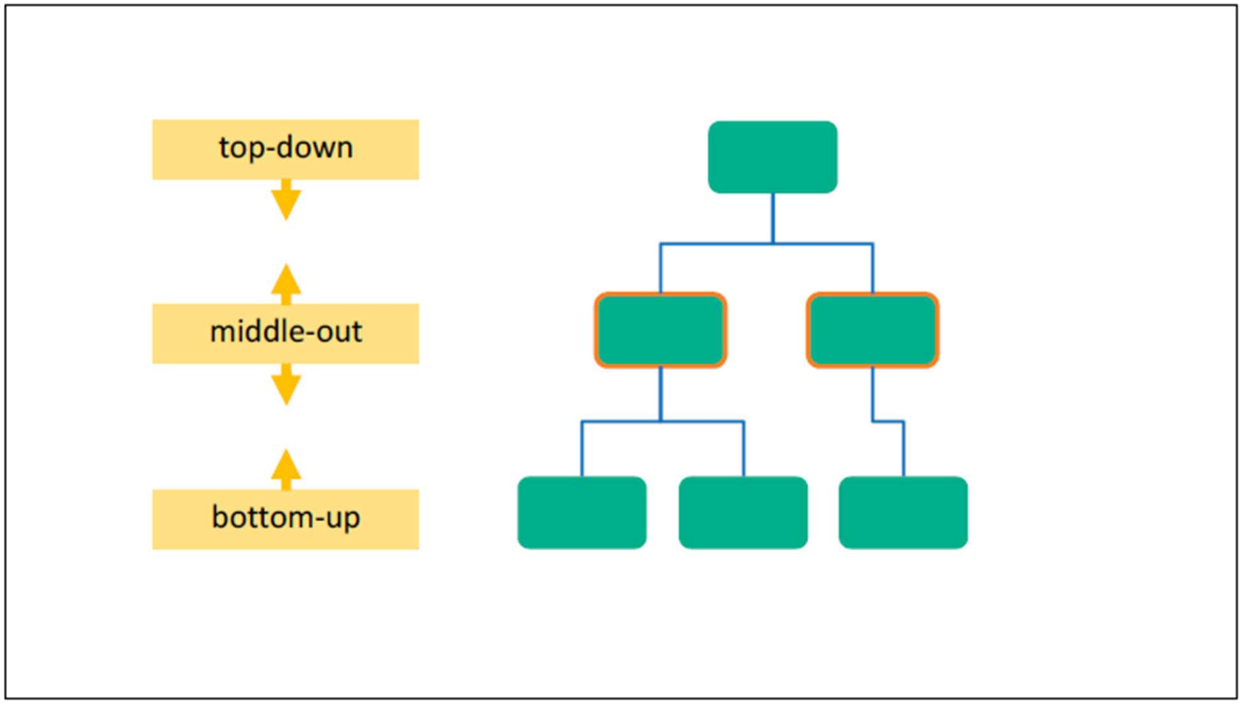
Finally, the bottom-up approach protects the bottom level of the hierarchy. You adjust the forecasts at the middle level first to achieve consistency, and then you adjust the highest-level forecasts to achieve consistency with the middle-level forecasts.



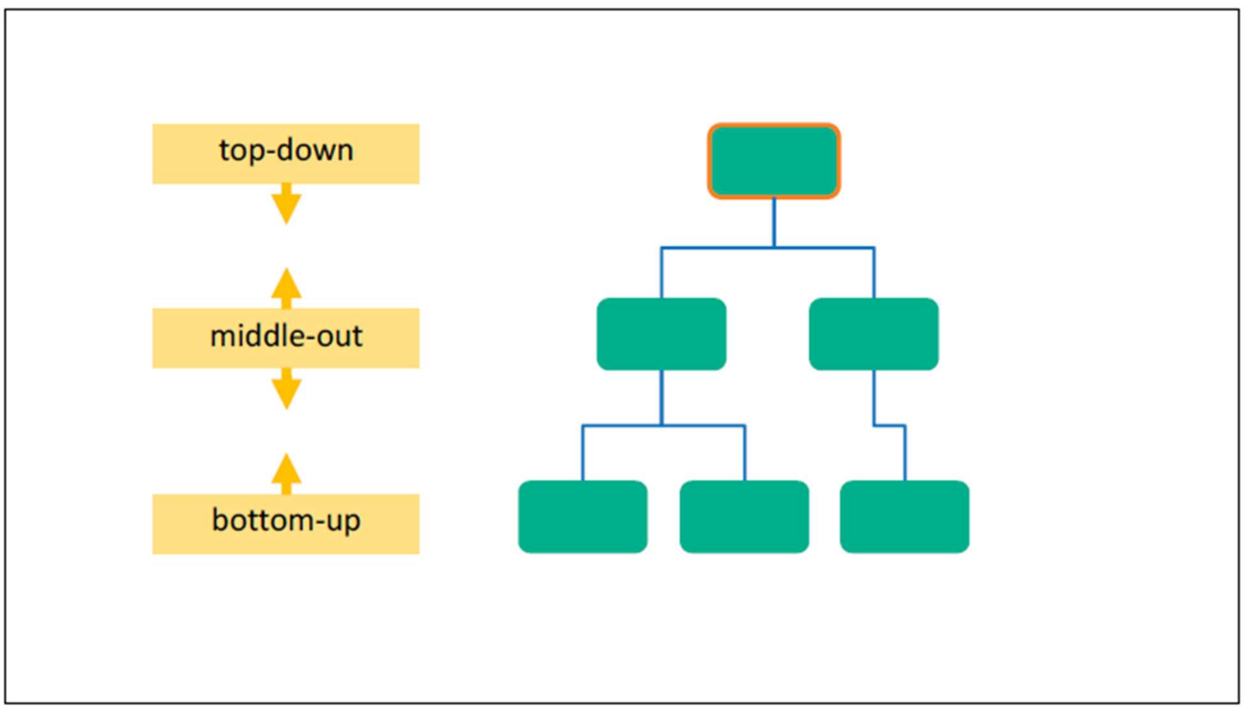
The data creation methods that we have discussed can play a role in choosing a reconciliation strategy.



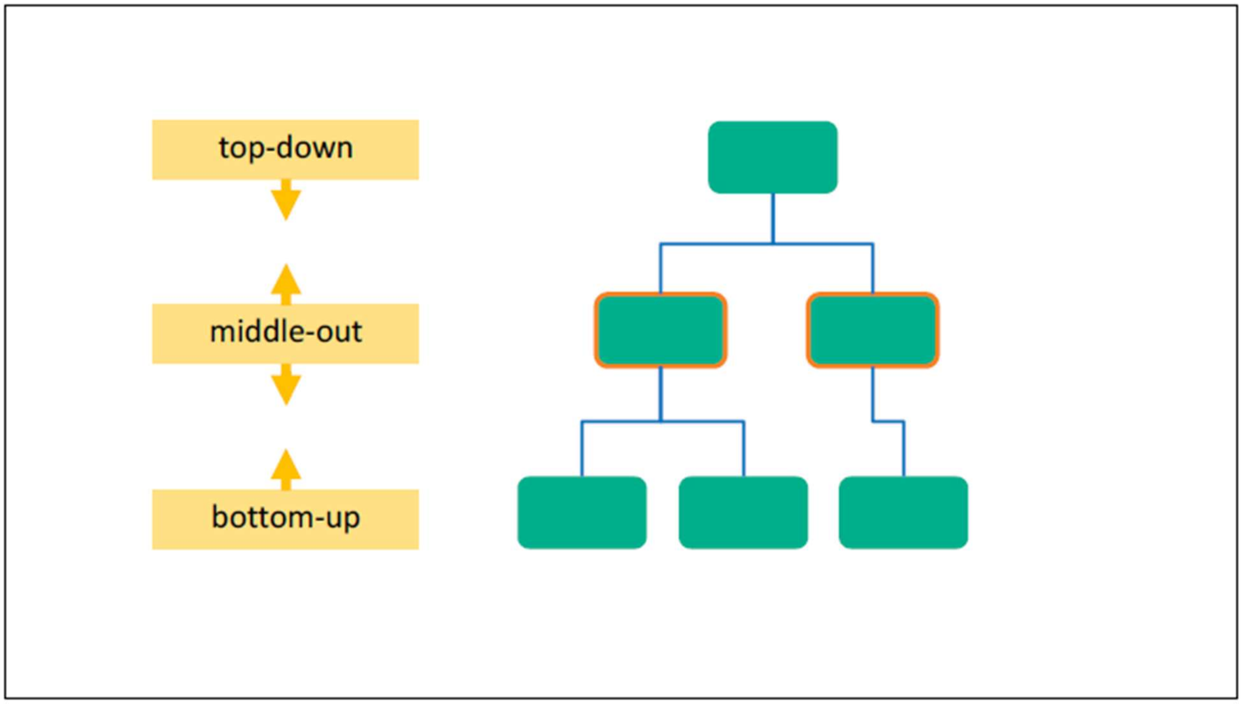
Data accumulated to the bottom level of the hierarchy are usually noisy and might be sparse. For example, consider store- or SKU-level data.



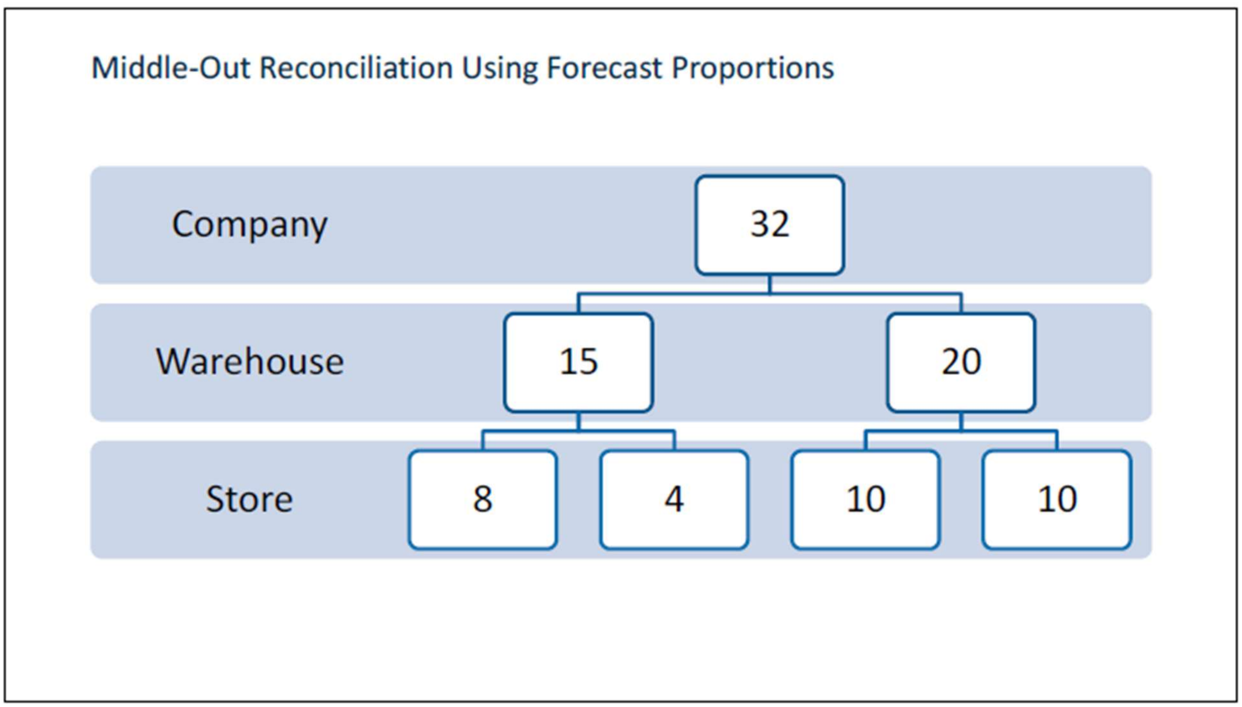
Data aggregation methods are smoothing processes, and sparseness is usually less of a problem at higher levels of the hierarchy. Patterns such as seasonality and trend are usually more easily discernible in data aggregated to the distribution center level, for example.



However, as data continues to be aggregated up the hierarchy, they can become overly smoothed. Interesting and predictive signals can be washed out.

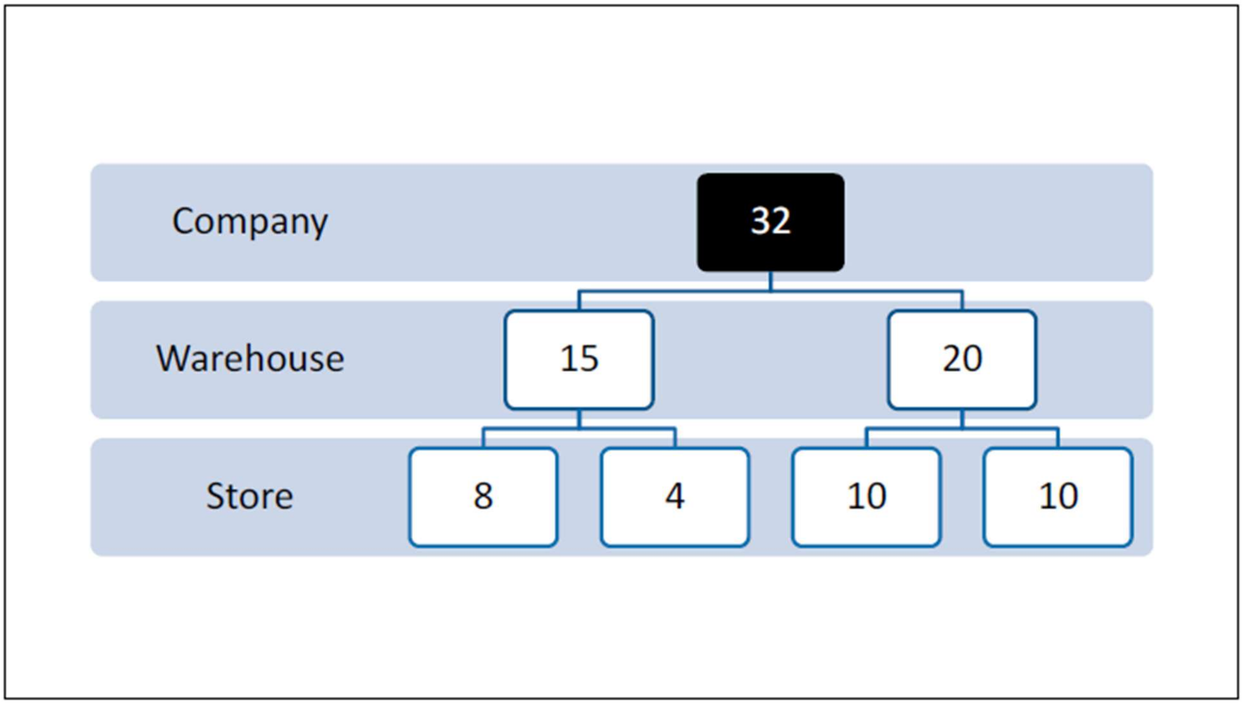


This leads to a rule of thumb for choosing a strategy for reconciliation. The best level to reconcile is usually somewhere in the middle of the hierarchy. This level is where the models get the best look at the available signal in the data.

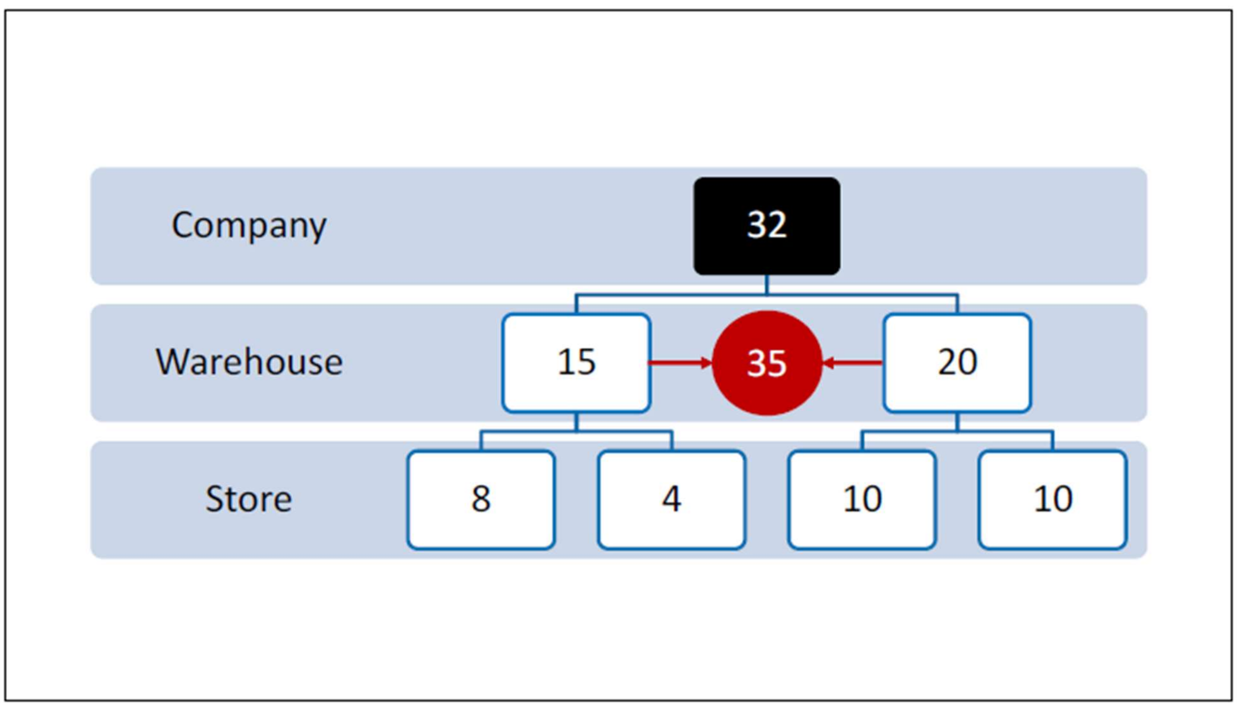


Let's look at a simple illustration of a middle-out reconciliation strategy using Forecast Proportions, the default methodology for distributing reconciliation effects.

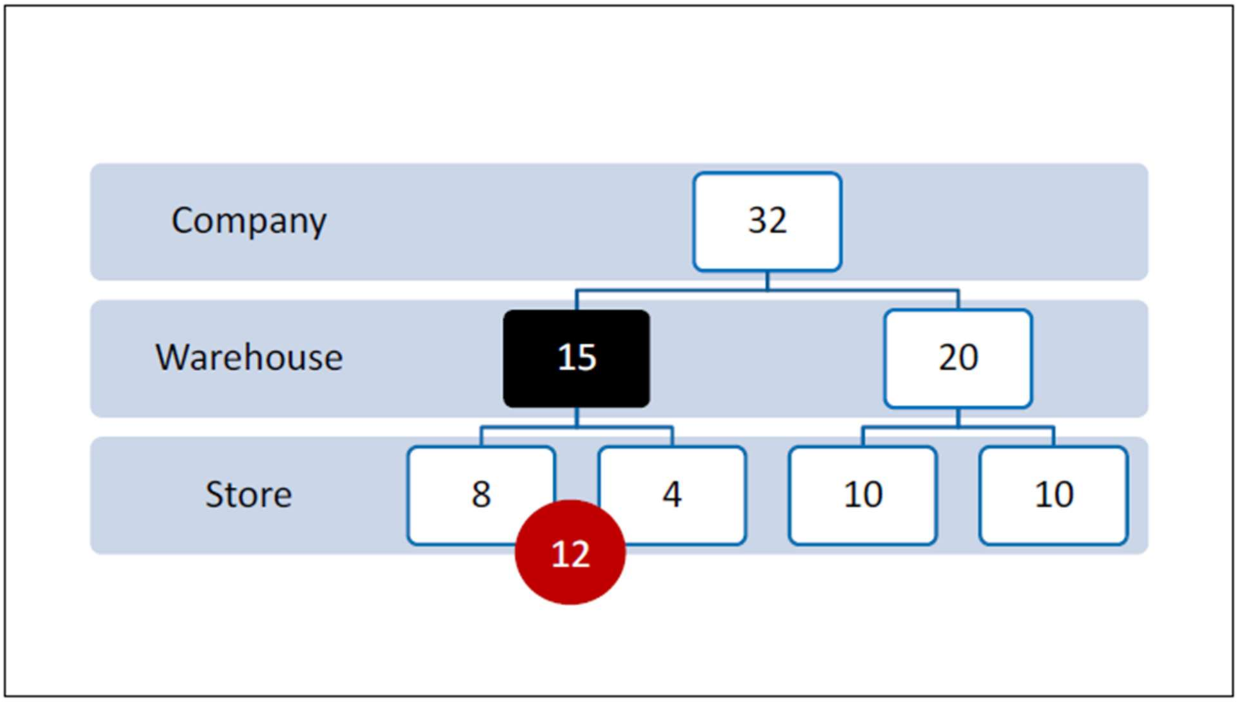
Statistical forecasts for January 2019 are shown in boxes on Company, Warehouse, and Store rows.



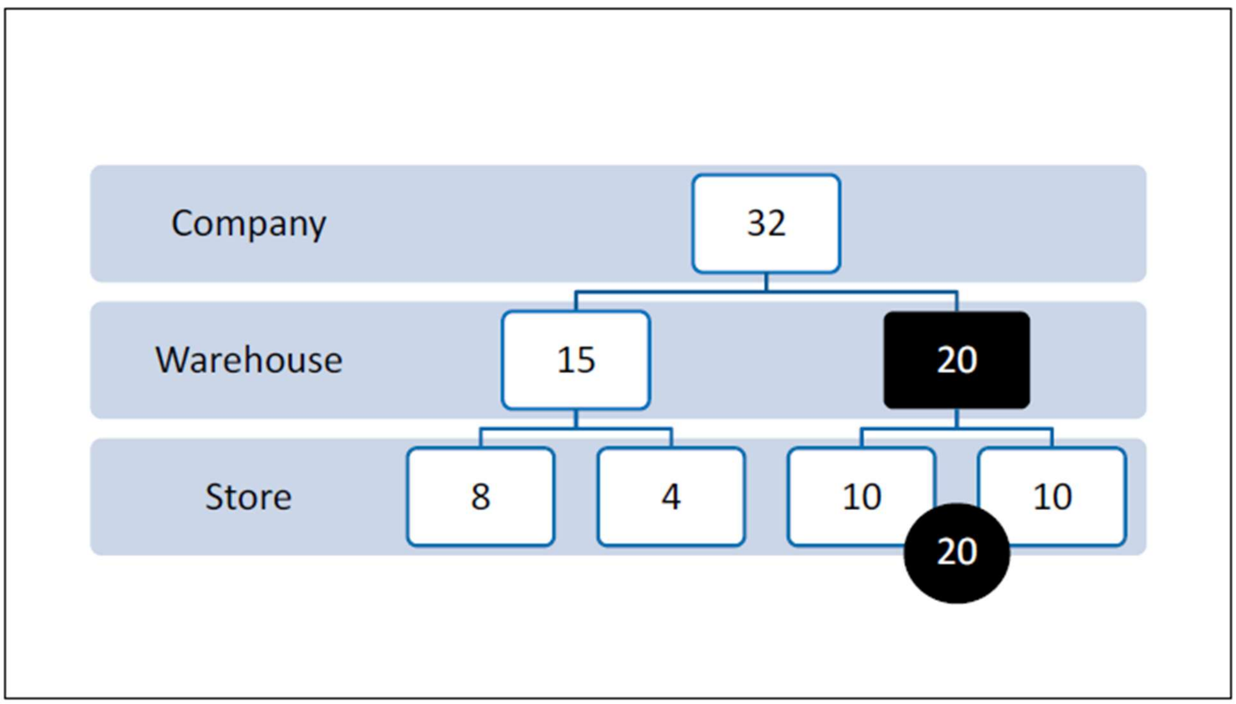
The company forecast is 32 units.



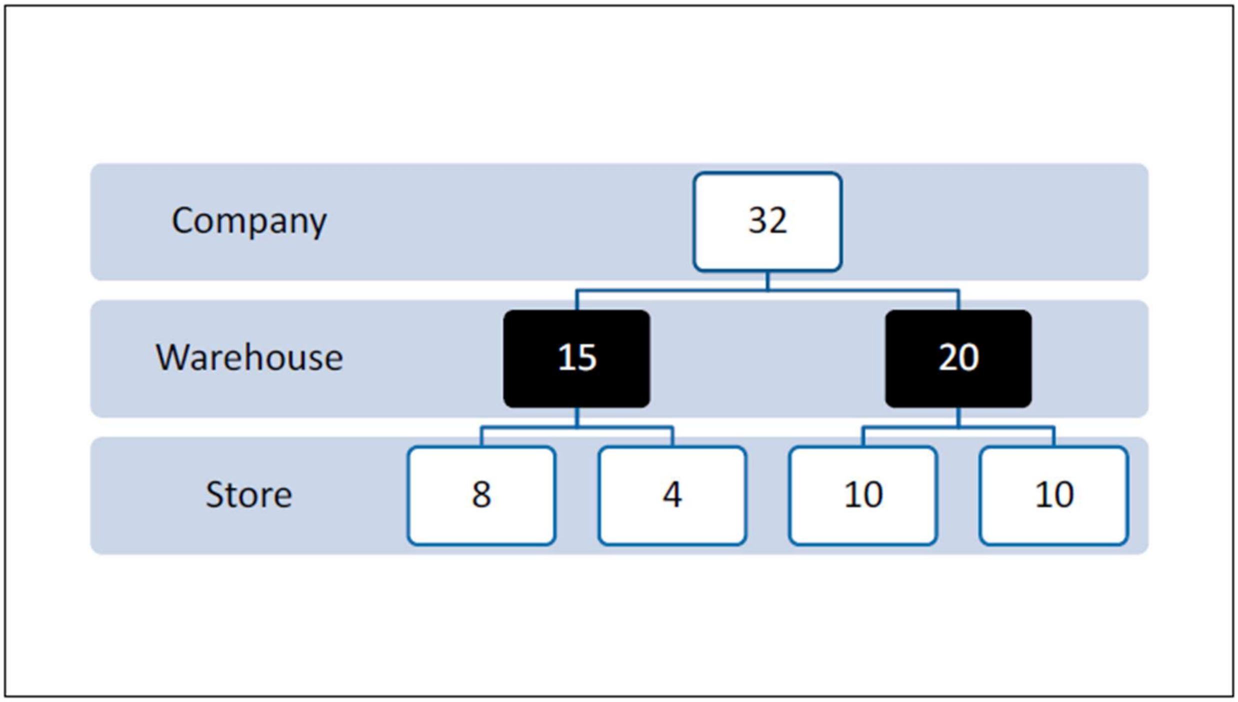
However, the two warehouses owned by the company have forecasts of 15 units and 20 units. The sum of the warehouse forecasts is 35 units, which is inconsistent with the company forecast of 32 units. This is possible because the company series was modeled independently from the warehouse series models.



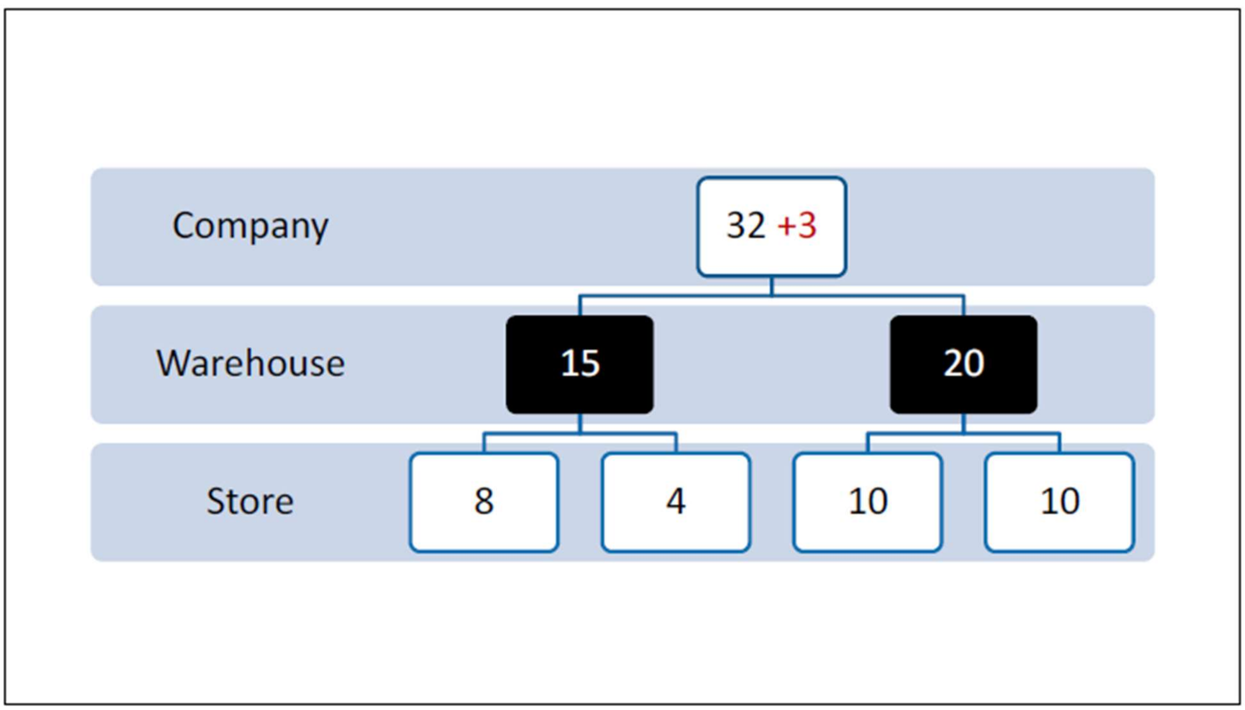
Similarly, the sum of the two stores' forecasts for the first warehouse is 12 units, whereas the forecast for the warehouse itself is 15 units.



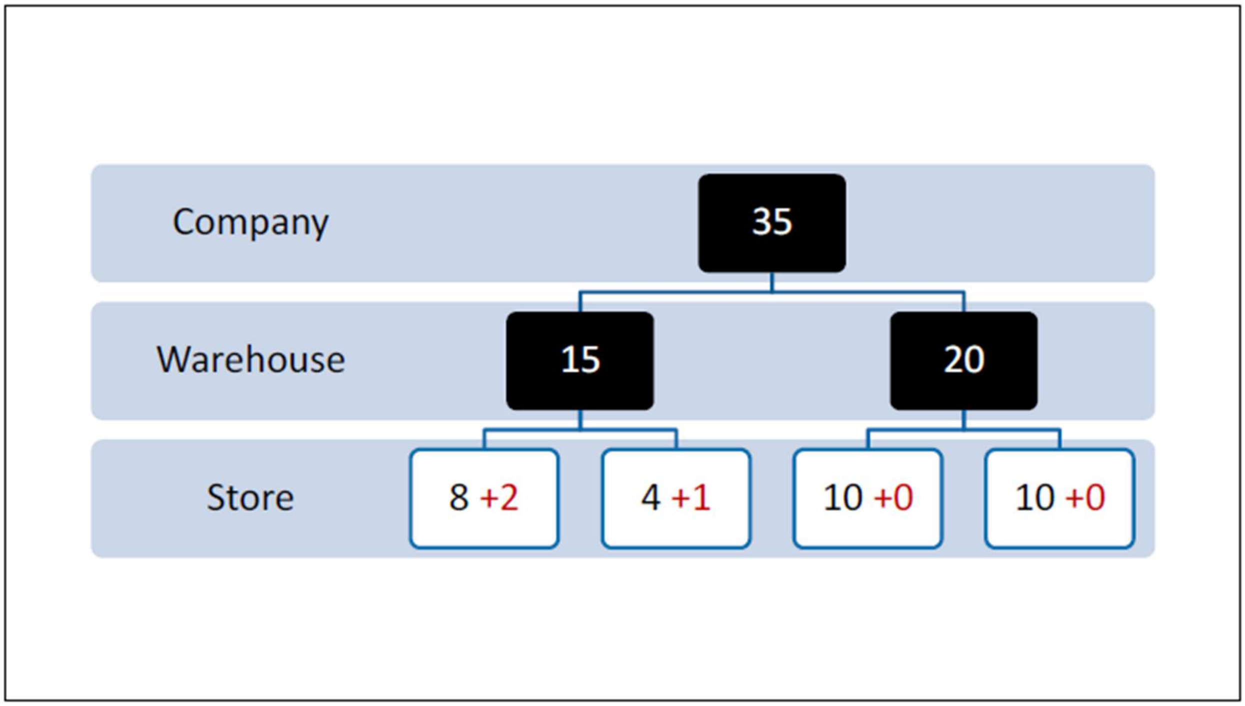
The sum of the store forecasts for the second warehouse is already consistent with the second warehouse forecast of 20 units.



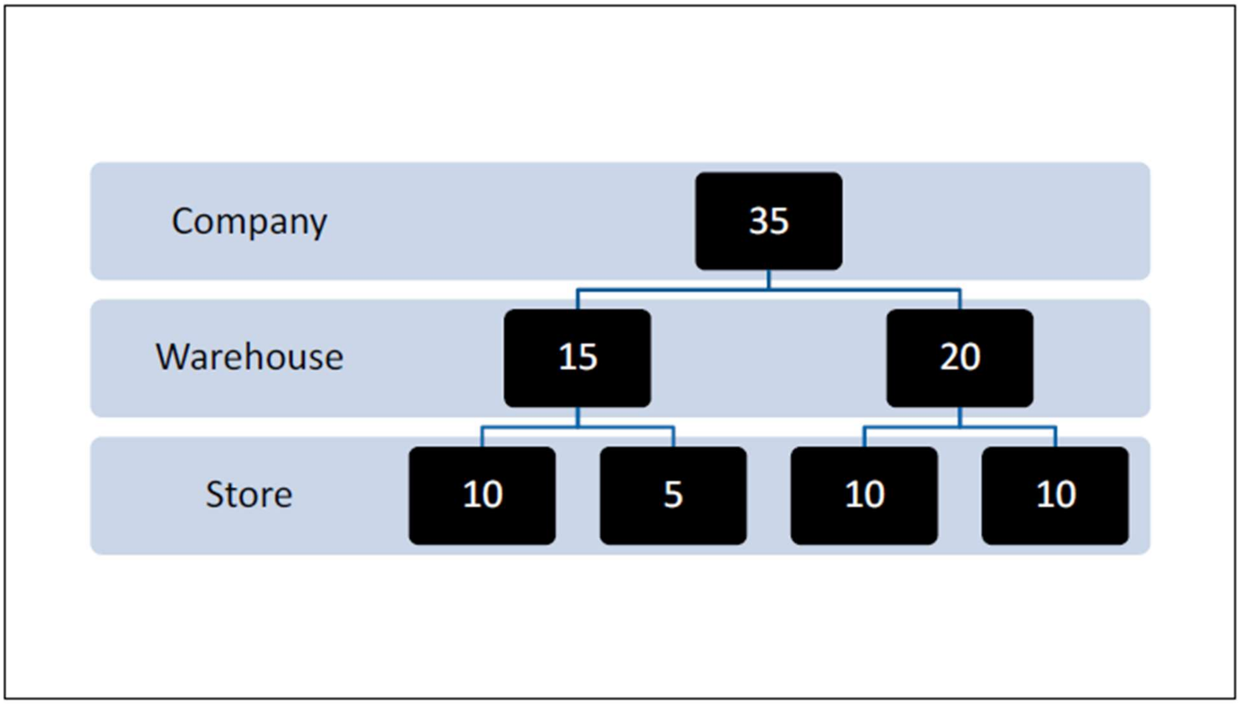
The company decides that the warehouse forecasts are the basis for reconciliation. Because warehouse is at the middle level of the hierarchy, we perform middle-out reconciliation. The warehouse forecast values, 15 and 20, are set as the standards for reconciliation.



The discrepancy between the top-level company forecast, 32, and the sum of the middle-level warehouse forecasts, 35, is three units. The company-level forecast is increased by three to make the top and middle levels consistent. To reconcile up a level of a hierarchy, find the difference between the higher level and the sum of the lower level of the hierarchy, and then add that value to the higher level.



To reconcile the values down a level of a hierarchy, you must decide how to apportion the reconciliation between series at the lower level of the hierarchy. In this case, the difference between the first warehouse forecast and its two stores' forecasts is three units. The company has decided to use proportional apportioning of the three units. The first store accounts for 8/12, or 2/3, of the sum of the store-level forecasts. We apportion two-thirds of the three reconciliation units to the first store. The second store is apportioned the remaining unit.



The reconciled forecasts are now consistent up and down the hierarchy.

Suggested SAS Courses to Learn More Information

[SAS Forecasting and Econometrics Learning Subscription](https://learn.sas.com/totara/program/view.php?id=82)
(https://learn.sas.com/totara/program/view.php?id=82)

[Forecasting Using Model Studio in SAS Viya](https://learn.sas.com/course/view.php?id=562)
(https://learn.sas.com/course/view.php?id=562)