Forecasting using ARIMA

One of the most common applications of predictive analytics is to forecast time-based data. This analytic uses the Auto-Regressive Integrated with Moving Average (ARIMA) algorithm to project a sequence of values ahead into the future, based on the assumption that data points taken over time may have an internal structure that can be measured.

While ARIMA tends to be esoteric and complex, this analytic uses the "auto.arima" function from R's "forecast" package to search through a variety of possible models in order to find the best one. Not only does this script generate the expected forecast values, it also provides outputs based on confidence bands, nominally set at 80% and 95% confidence levels.

For more detailed information on ARIMA, please consult the documents [3] and [4] referenced at the end.

How to Deploy to MicroStrategy:

Prerequisite: Please follow the instructions in the R Integration Pack User Guide [1] for configuring your MicroStrategy environment with R and that the R Script functions have been installed in your MicroStrategy project(s).

1) Download the ARIMA.R file from the R Script Shelf [2] and place it in the RScripts folder where the R Integration Pack is installed (usually C:\Program Files (x86)\R Integration Pack\RScripts).
2) From the R console, run the ARIMA.R script to verify the script runs correctly. For details, see the "**Running from the R Console**" section below.
3) Cut-and-paste the metric expression below in any metric editor. This metric returns the forecast of seasonal values.
4) Use the new metric in reports, dashboards and documents.

Metric Expressions:

1) **Forecast**: Returns the forecasted value as a number

```
RScript<_RScriptFile="ARIMA.R", _InputNames="Target", SortBy=(Month), StringParam8="",
StringParam9="", NumericParam1=12, NumericParam2=12, NumericParam3=80,
NumericParam4=95>(Target)
```

2) **ForecastLo1**: Returns the forecasted lower value of the first confidence band as a number

```
RScript<_RScriptFile="ARIMA.R", _InputNames="target", SortBy=(Month),
[_OutputVar]="ForecastLo1", StringParam9="ARIMA", NumericParam1=12, NumericParam2=12,
NumericParam3=80, NumericParam4=95>(target)
```

3) **ForecastHi1**: Returns the forecasted upper value of the first confidence band as a number

```
RScript<_RScriptFile="ARIMA.R", _InputNames="target", SortBy=(Month),
[_OutputVar]="ForecastHi1", StringParam9="ARIMA", NumericParam1=12, NumericParam2=12,
NumericParam3=80, NumericParam4=95>(target)
```

4) **ForecastLo2**: Returns the forecasted lower value of the second confidence band as a number

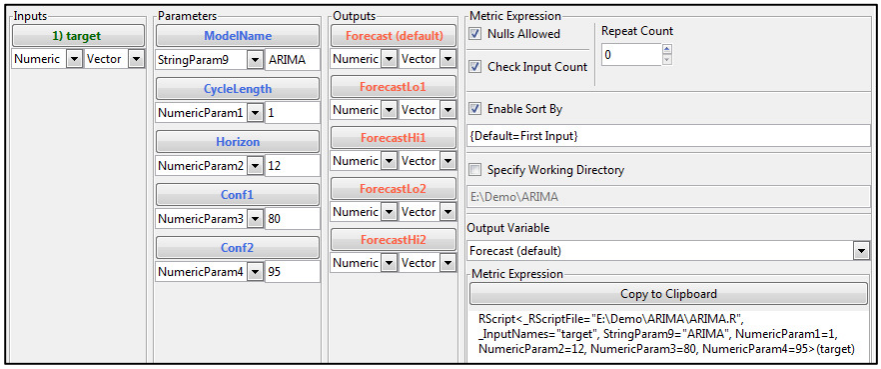
```
RScript<_RScriptFile="ARIMA.R", _InputNames="target", SortBy=(Month),
[_OutputVar]="ForecastLo2", StringParam9="ARIMA", NumericParam1=12, NumericParam2=12,
NumericParam3=80, NumericParam4=95>(target)
```

Metric Expressions (continued):

5) **ForecastHi2: Returns the forecasted upper value of the second confidence band as a number**

```
RScript<_RScriptFile="ARIMA.R", _InputNames="target", SortBy=(Month),
[_OutputVar]="ForecastHi2", StringParam9="ARIMA", NumericParam1=12, NumericParam2=12,
NumericParam3=80, NumericParam4=95>(target)
```

Analytic Signature:



Input:

Target: Value to forecast. All non-null values are used by the ARIMA algorithm, rows with null values for the Target are excluded from the machine learning algorithm (these are typically future values to be forecast).

Parameters:

ModelName: Uses StringParam9 with a default of "ARIMA". This parameter specifies the file name (and optionally the path) to use for the files saved by the script. Please note the R Script automatically appends the appropriate file extensions to this file name.

CycleLength: Uses NumericParam1 with a default of 1 (no seasonal cycle). Number of periods in the seasonal cycle. For example, use 7 for daily data with day of week seasonality, use 12 for monthly data with month of year seasonality.

Horizon: Uses NumericParam2 with a default of 12. Number of time periods ahead in the forecast horizon.

Conf1: Uses NumericParam3 with a default of 80. Confidence value (in percent) of the first prediction interval.

Conf2: Uses NumericParam4 with a default of 95. Confidence value (in percent) of the second prediction interval.

Outputs:

Forecast: Forecasted value

ForecastLo1: Lower value of the first confidence band.

ForecastHi1: Higher value of the first confidence band.

ForecastLo2: Lower value of the second confidence band.

ForecastHi2: Higher value of the second confidence band.

Additional Results Generated by the R Script:

Two files are stored in the RScripts folder, or the working directory if one is specified by the _WorkingDir function parameter:

.Rdata File: This file persists the state of several objects from the R environment for later inspection and analysis, including target, model (the ARIMA model object), arimaFcst (the forecast object), and Forecast (the expected value).

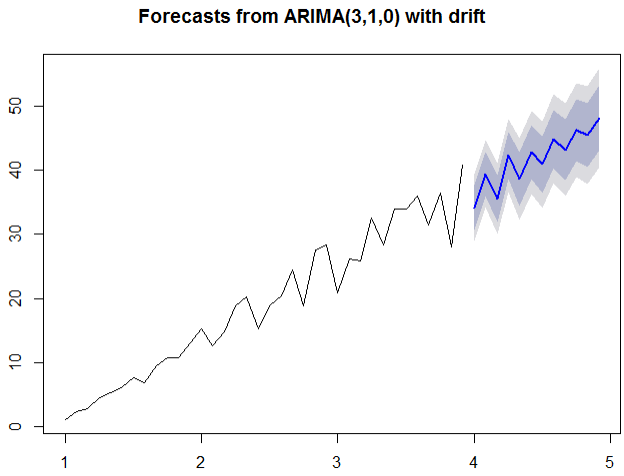
.jpg File: This is the image showing the forecast and confidence bands.

Running from the R Console:

In addition to processing data from MicroStrategy during execution of a report or dashboard, the R script is also configured to run from the R console. Running the script for the R Console verifies that the script is functioning as expected, a good practice when initially deploying this analytic to a new system (for more details, see "Configuring dual execution modes" in [1]).

When run from the R Console, if the script is executing properly, a "Success!" message will appear in the console and the image shown here will be plotted.

If a "Success!" message does not appear, then please note the error in order to take appropriate action. For common pitfalls, please consult the **Troubleshooting** section below.



Troubleshooting:

This section covers certain situations you might encounter but it's not intended as a comprehensive list of possible errors.

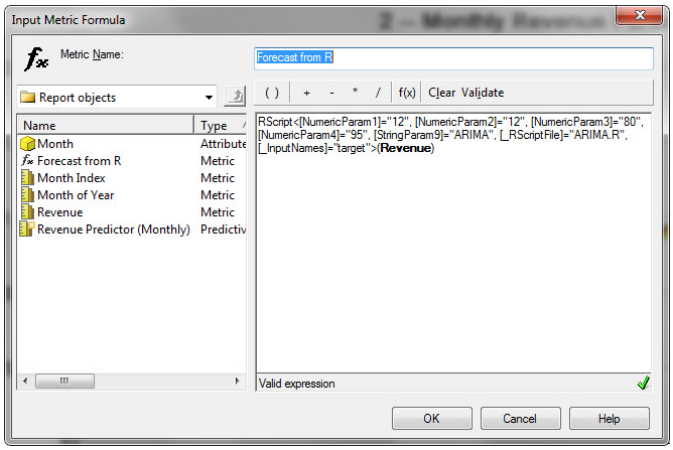
1) If an error occurs, the report may fail with an error message, or nulls returned as the output. In these cases, please refer to the RScriptErrors.log file generated for further guidance and the DSSErrors.log. Please consult the User Guide [1] and the R documentation for additional guidance.

2) The script will attempt to install one required R package:
   forecast: This R package contains the ARIMA algorithm used to build the forecast model.
   If the package is not successfully installed, you can install using the R console using the command:
   `install.pacakges("forecast", repos="http://cran.rstudio.com/")`

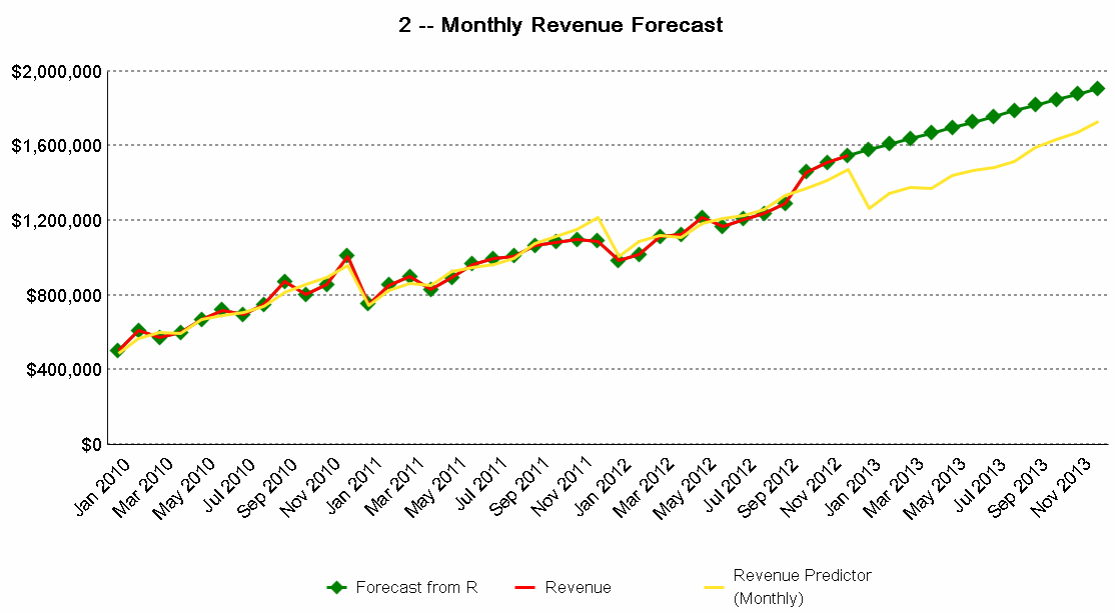
Example:

Using the "2 – Monthly Revenue Forecast" report from the MicroStrategy Tutorial (see Tutorial\Public Objects\Reports\MicroStrategy Platform Capabilities\MicroStrategy Data Mining Services\Linear Regression\Monthly), insert a new metric called Forecast from R (or another name you choose), paste in the Seasonal Forecasting metric expression and assign the inputs as follows:

- Target = Revenue
- NumericParam1 = 12 (this is the number of time periods in the seasonal cycle)



Click ok and you should see that the new results linearly extend the most recent trend of the data:



References:

1) MicroStrategy R Integration Pack User Guide:
https://rintegrationpack.codeplex.com/documentation
2) R Script Shelf:
http://rintegrationpack.codeplex.com/wikipage?title=R%20Script%20%22Shelf%22&referringTitle=Home
3) http://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average
4) http://people.duke.edu/~rnau/411arim.htm

