SAP BI BLOG
all things business intelligence

decisionfirst technologies

Building a One-Way ANOVA R Extension in SAP Predictive Analytics

One of the first statistical tools many new analysts use is the Analysis of Variance, a collection of statistical methods used to decompose and understand the causes of variation within a set of data. One-way ANOVA is perhaps the most basic of these methods and a staple of most statistical software. While SAP has included many predictive algorithms in their SAP Predictive Analytics Expert application, it is missing many of the common descriptive algorithms used by data scientists to better understand their data. Luckily, it is relatively easy to build in custom R extensions to accommodate any descriptive statistical needs.

What is One-Way ANOVA?

One-Way Analysis of Variance is used to compare means within 3 or more samples to evaluate whether or not all groups have the same mean (in effect, there is no difference in the monitored statistic between the groups). Of course there is natural variation in data, so the actual means of the groups may vary slightly, but the age old question persists: **is the difference statistically significant???**

One-Way ANOVA is an omnibus test, which means that if the null hypothesis (all means are the same) is rejected, it offers no additional information on which of the group(s) are different from each other, simply that at least one of them is different enough to reject the hypothesis that they are all the same.

For additional background on One-Way ANOVA, see the relevant Wikipedia article.

Building the Custom R Extension

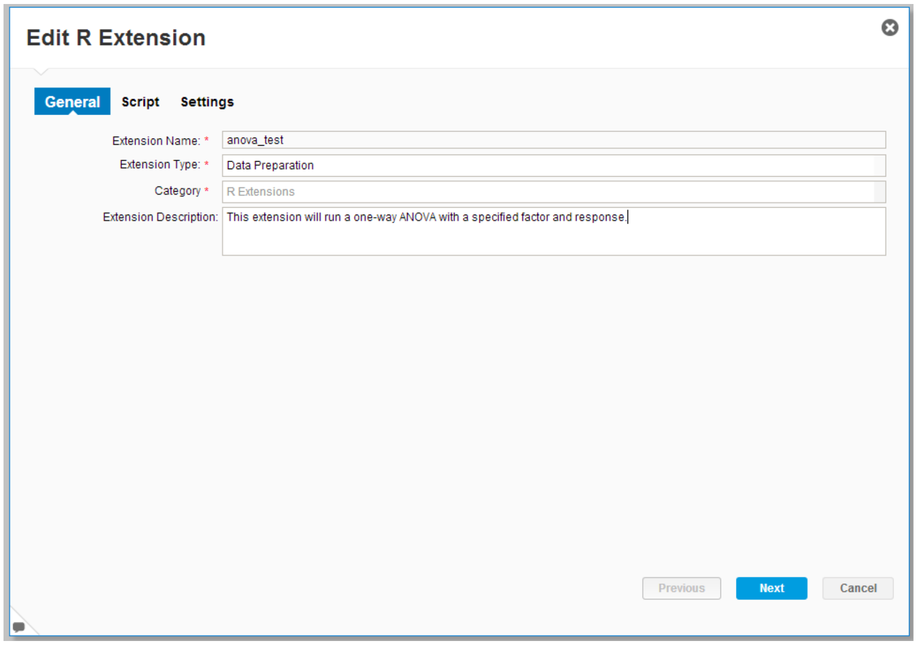
Unlike many of the predictive modeling components, we will be configuring ANOVA as a "Data Preparation" algorithm because we are not building a model, and do not need the typical set of predictive inputs and outputs. After naming our extension ("anova_test" in this example), we proceed to the script editor, where we have a basic R script for a function that runs the ANOVA model and returns a dataframe with the commonly-known ANOVA table.

R Script for One-Way ANOVA extension:

```
anova_test<-function(InputDataFrame, FactorColumn, DependentColumn){
  finalString<-paste(paste(DependentColumn, "~" ),FactorColumn);

  lr_model<-lm(finalString, InputDataFrame);
   par(mfrow=c(2,2))
   plot(lr_model)
   anova_fit<-anova(lr_model)
    print(anova_fit)
  anovadf<-data.frame(anova_fit)
  anovadf<-data.frame(Factor=rownames(anovadf), anova_fit)

return (list(out=anovadf))
}
```



Edit R Extension

General Script Settings

Extension Name: * anova_test

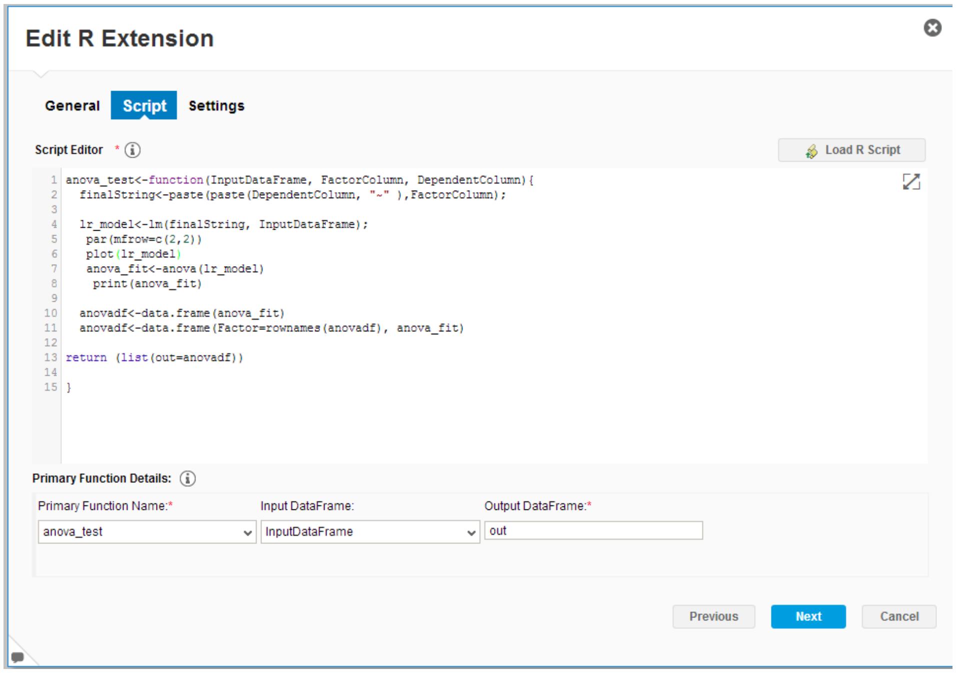
Extension Type: * Data Preparation

Category * R Extensions

Extension Description: This extension will run a one-way ANOVA with a specified factor and response.

Previous Next Cancel

R Script in the R extension configuration window:



Edit R Extension

General Script Settings

Script Editor *

Load R Script

```
anova_test<-function(InputDataFrame, FactorColumn, DependentColumn){
  finalString<-paste(paste(DependentColumn, "~" ),FactorColumn);

  lr_model<-lm(finalString, InputDataFrame);
   par(mfrow=c(2,2))
   plot(lr_model)
   anova_fit<-anova(lr_model)
    print(anova_fit)

  anovadf<-data.frame(anova_fit)
  anovadf<-data.frame(Factor=rownames(anovadf), anova_fit)

return (list(out=anovadf))

}
```

Primary Function Details:

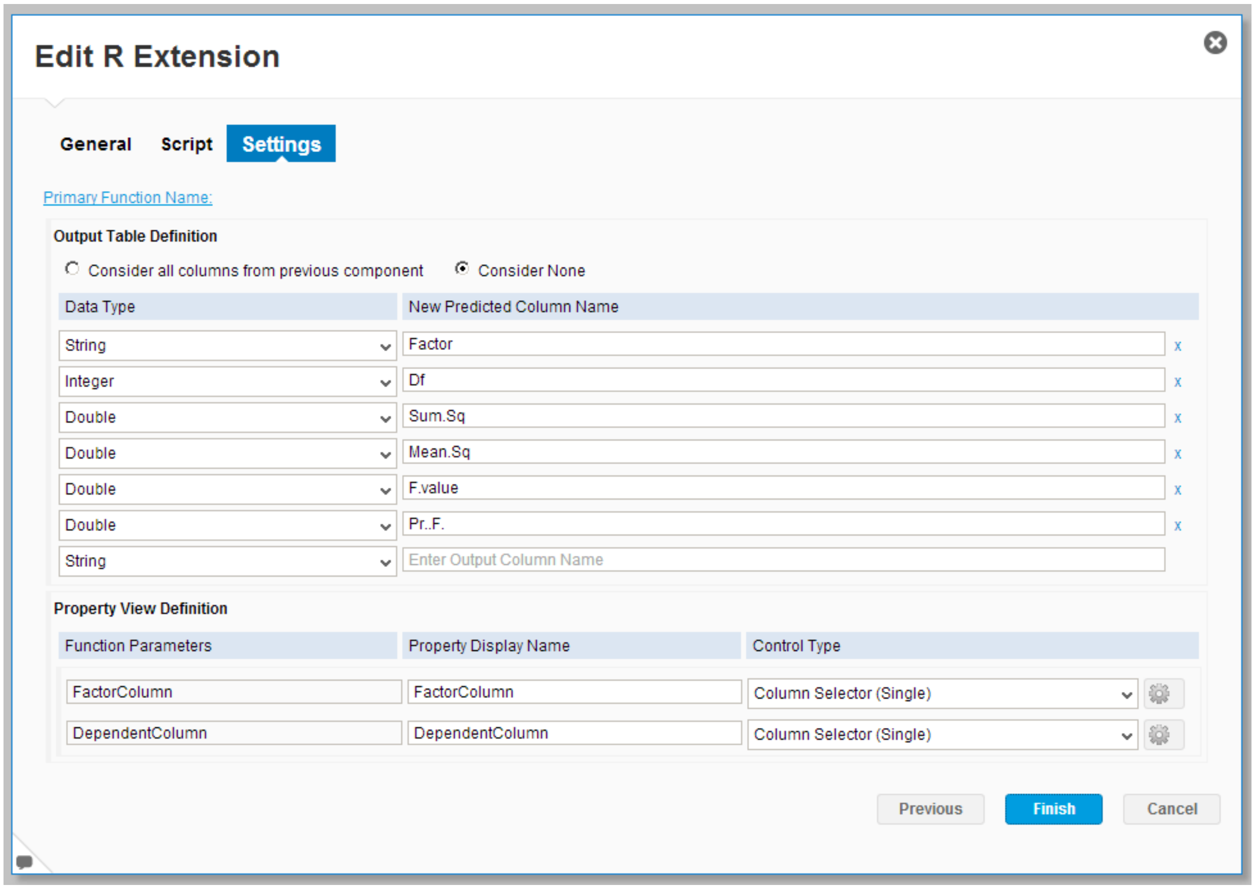
Primary Function Name:* anova_test

Input DataFrame: InputDataFrame

Output DataFrame:* out

Previous Next Cancel

Since the only function we have defined is anova_test, we select this as our Primary Function Name. The user must input 2 pieces of information: which column in our input dataset is the grouping variable and which is the response variable of interest.



Primary Function Name:

Output Table Definition

○ Consider all columns from previous component ● Consider None

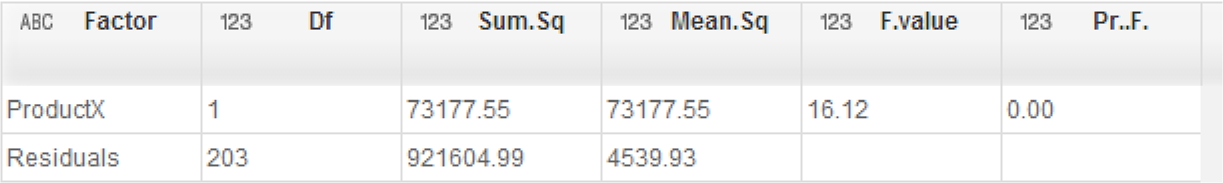
Data Type	New Predicted Column Name
String	Factor
Integer	Df
Double	Sum.Sq
Double	Mean.Sq
Double	F.value
Double	Pr..F.
String	Enter Output Column Name

Property View Definition

Function Parameters	Property Display Name	Control Type
FactorColumn	FactorColumn	Column Selector (Single)
DependentColumn	DependentColumn	Column Selector (Single)

The output is more complex; we actually must create output columns for each of the columns included in the standard ANOVA output table.

This is a preview of what the output will look like:

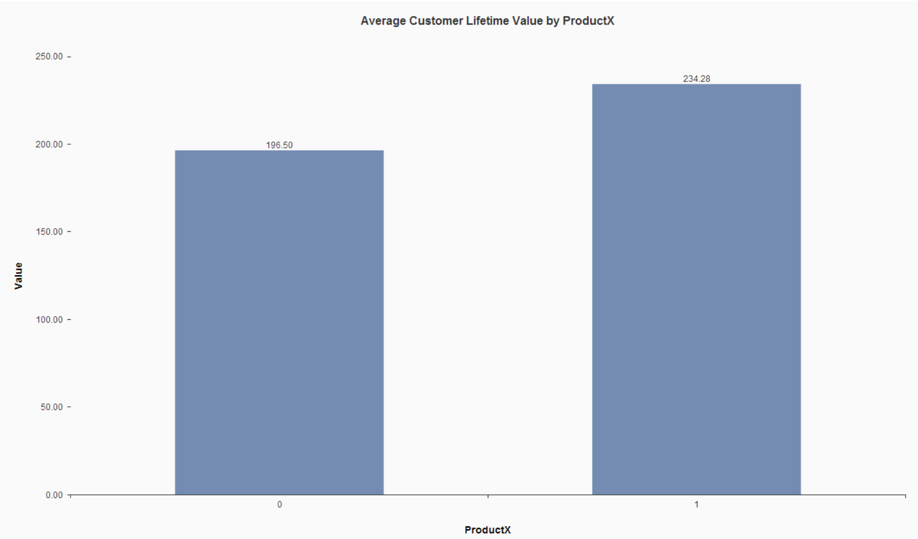


ABC Factor	123 Df	123 Sum.Sq	123 Mean.Sq	123 F.value	123 Pr..F.
ProductX	1	73177.55	73177.55	16.12	0.00
Residuals	203	921604.99	4539.93		

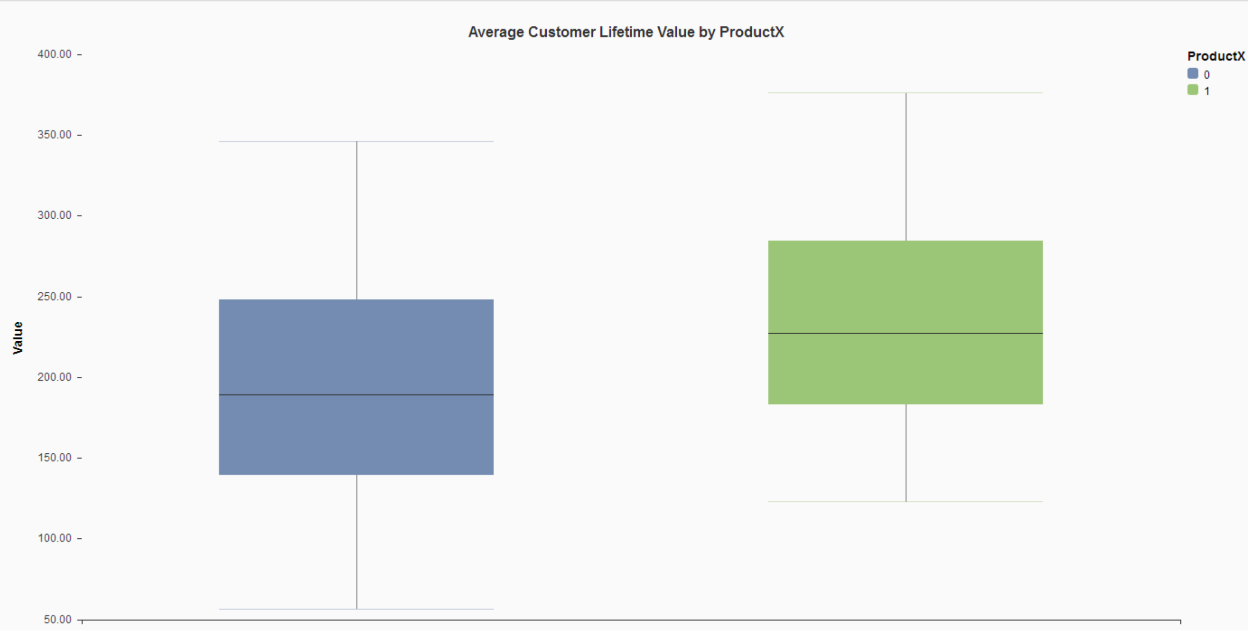
Example Use Case

To illustrate how the One-Way ANOVA component works, I'll use the Customer Lifetime Value dataset I've used several times before. This dataset shows the customer lifetime value for a set of customers, along with 3 descriptive characteristics: age, income, and whether or not they've purchase a specific product (Product X).

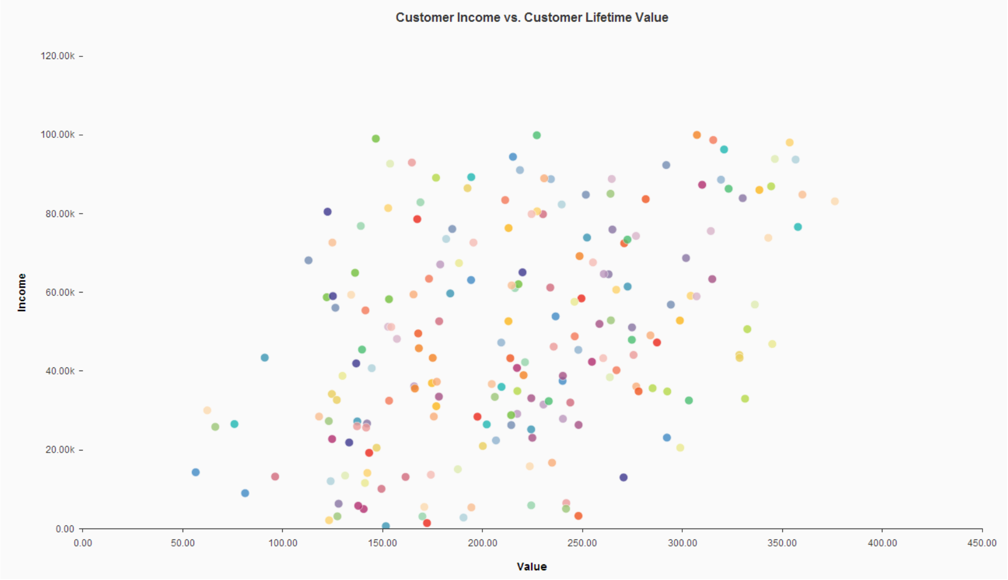
It is of course of interest to our organization to understand which factors correlate with high-value customers, specifically whether they purchased our flagship product, Product X. An initial analysis of average customer lifetime value for customers that purchased Product X vs. those that didn't shows that product X customers have an average value of $234.38 vs. $196.50 for those that did not purchase Product X.

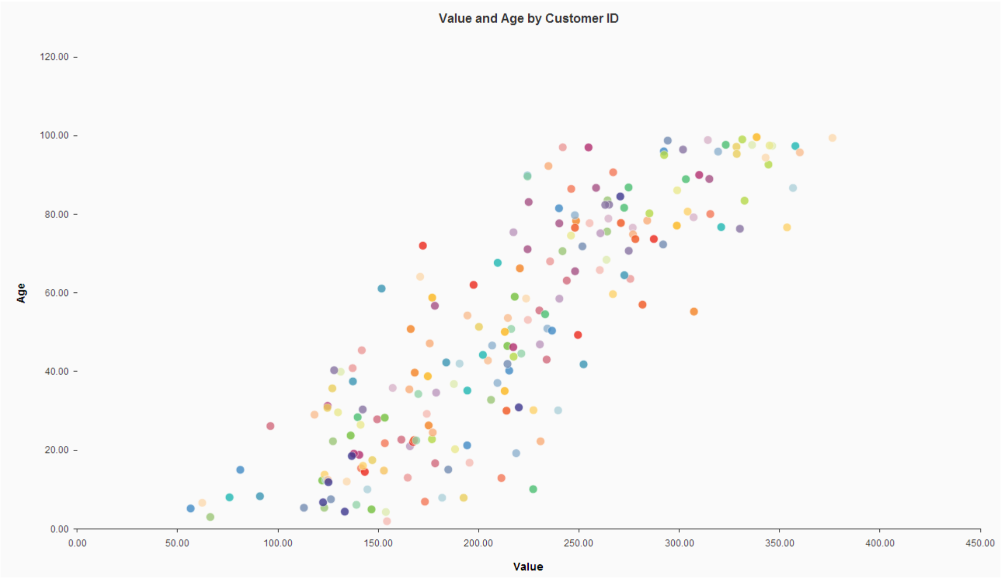


However, there is a lot of variation in customer value, so we can look at the distributions of lifetime value in a box plot comparing customers that purchased Product X to those that didn't.

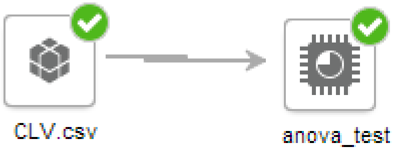


However, we have also seen strong correlations between customer income and customer lifetime value and age and customer lifetime value (shown in the scatter plots below), so we'd like to better understand whether Product X is truly statistically significant in understanding customer lifetime value.

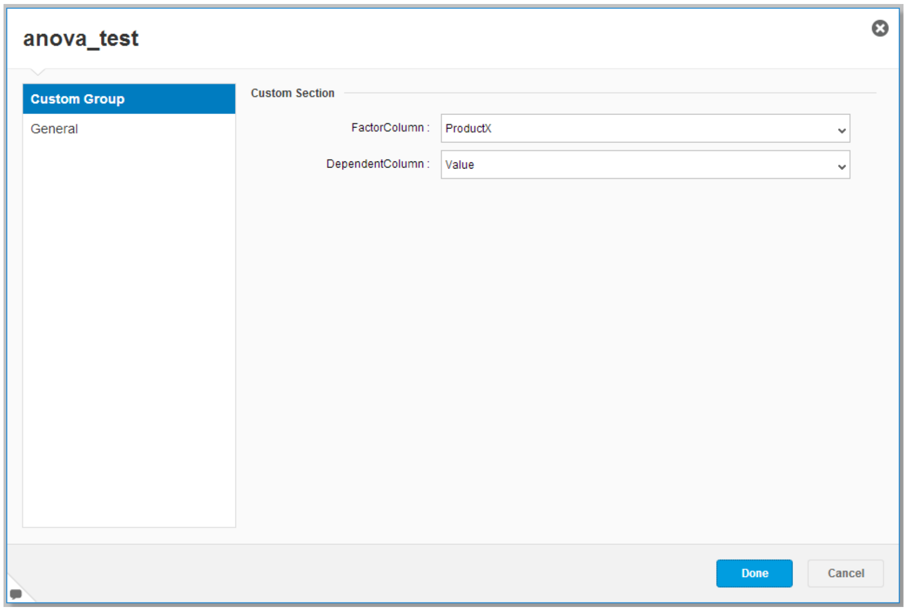




Generally, one-way ANOVA is only used when there are 3 or more groups to be evaluated, so this is not an ideal use case, but it will illustrate the point. We add the anova_test component to the predictive workflow in Expert Analytics' Predict pane:



And configure it to use ProductX purchase (1/0) as the factor and Value (customer lifetime value) as the dependent column as shown in the screen below.

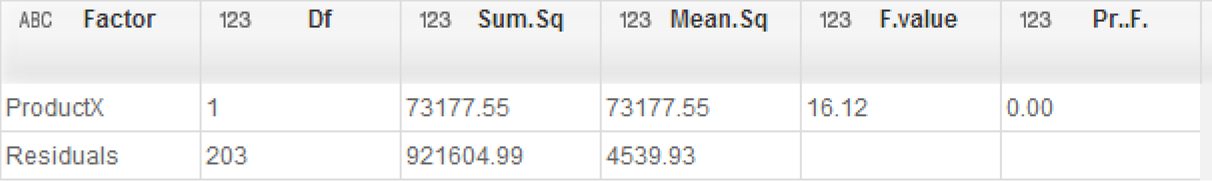


anova_test

FactorColumn : ProductX

DependentColumn : Value

After running, we can review the results of the analysis of variance, showing that a large portion of the error is accounted for by the product X factor; the P value (probability of the null hypothesis being true) is significantly below a reasonable threshold (for example, p=0.05), indicating that we can reject the null hypothesis (that Product X purchasers have the same lifetime value as non-Product X purchasers) with a high degree of confidence.



ABC Factor	123 Df	123 Sum.Sq	123 Mean.Sq	123 F.value	123 Pr..F.
ProductX	1	73177.55	73177.55	16.12	0.00
Residuals	203	921604.99	4539.93		

Hillary Bliss
Decision First Technologies
Hillary.bliss@protiviti.com
twitter @HillaryBlissDFT

Hillary Bliss is a Senior Manager at Protiviti, and specializes in data warehouse design, ETL development, statistical analysis, and predictive modeling. She works with clients and vendors to integrate business analysis and predictive modeling solutions into the organizational data warehouse and business intelligence environments based on their specific operational and strategic business needs. She has a master’s degree in statistics and an MBA from Georgia Tech.