Salford Predictive Modeler®

Introducing MARS®

This guide provides a brief introduction to Multivariate Adaptive Regression Splines

Minitab ▷

Minitab ▶®

Introduction to MARS®

Welcome to MARS®, the world's first truly successful automated regression modeling tool. Multivariate Adaptive Regression Splines (MARS) was developed in the early 1990s by world-renowned Stanford physicist and statistician Jerome Friedman, but has become widely known in the data mining and business intelligence worlds only recently through our seminars and the enthusiastic endorsement of leading data mining specialists. We expect that you will find MARS an essential component in your data analysis tool kit.

MARS is an innovative and flexible modeling tool that automates the building of accurate predictive models for continuous and binary dependent variables. It excels at finding optimal variable transformations and potential interaction within any regression-based modeling solution and easily handles the complex data structure that often hides in high-dimensional data. In doing so, this new approach to regression modeling effectively uncovers important data patterns and relationships that are difficult, if not impossible, for other methods to reveal.

Although regression is one of the most widely used tools in statistical analysis, it is rarely used in data mining. Nevertheless, a well-crafted regression model can be ideal for predictive modeling and data mining because of the following important characteristics:

♦ A regression model predicts the outcome variable by forming a weighted sum of the predictor variables; thus, the predicted outcome changes in a smooth and regular fashion as the inputs change.

♦ This is in contrast to a decision tree where a small change in a predictor could either move a prediction to a different node in the tree or result in no change in the prediction at all.

♦ When scoring a database, regression models typically produce unique scores for each record. Decision trees assign the same score to all records arriving at a specific node; thus, the smaller the decision tree, the fewer the number of unique scores assigned.

♦ In a regression, it is often possible to "read" the effect of a predictor variable on the outcome by examining its slope coefficient; experienced modelers can often read the effects of interactions directly from the model itself.

Developing a good regression model is usually an extremely time intensive activity requiring considerable modeling expertise, even for small databases. For the large databases common in data mining projects, model-building challenges have deterred data miners from using this otherwise very effective tool. However, with the advent of MARS, regression models can now be routinely and automatically developed for the most complex data structures.

Given a target variable and a set of candidate predictor variables, MARS *Automates* all aspects of model development and model deployment, including:

♦ Variable selection: separating relevant from irrelevant predictor variables.

♦ Transforming predictor variables exhibiting a nonlinear relationship with the target.

♦ Determining interactions between predictor variables.

♦ Handling missing values with new nested variable techniques.

♦ Conducting extensive self-tests to protect against overfitting.

MARS enables you to rapidly search through all possible models and to quickly identify the "optimal" solution. Because the software can be exploited via intelligent default settings, for the first time analysts at all technical levels can easily access MARS' innovations.

Minitab ▷®

MARS essentially builds flexible models by fitting piecewise linear regressions; that is, the nonlinearity of a model is approximated through the use of separate regression slopes in distinct intervals of the predictor variable space. An example of a piecewise linear regression is shown below.
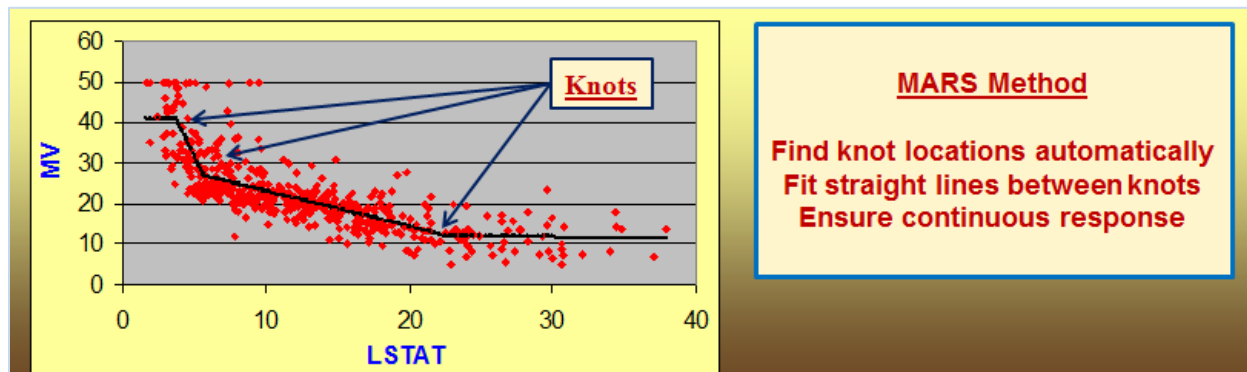
Notice how the slope of the regression line is allowed to change from one interval to the other as the two "knot" points are crossed.

The variables to use and the end points of the intervals for each variable are found via a fast but very intensive search procedure. In addition to searching variables one by one, MARS also searches for interactions between variables, allowing any degree of interaction to be considered.

The optimal MARS model is selected in a two-stage process. In the first stage, MARS constructs an overly large model by adding "Basis functions" - the formal mechanism by which variable intervals are defined. Basis functions represent either single variable transformations or multivariable interaction terms. As Basis functions are added, the model becomes more flexible and more complex, and the process continues until a user-specified maximum number of Basis functions have been reached. In the second stage, Basis functions are deleted in order of least contribution to the model until an optimal model is found. By allowing for any arbitrary shape for the function as well as for interactions, and by using this two-stage model selection method, MARS is capable of reliably tracking the very complex data structures that often hide in high-dimensional data.

The MARS output contains an easy-to-deploy regression model that can be simply applied to new data from within MARS or exported as SAS®-, Classic, C, PMML, Java or Topology compatible code. To facilitate interpretation of the model, the output includes interpretive summary reports as well as exportable two- and three-dimensional curve and surface plots. It is also reasonably easy to convert the MARS basis function output into an Excel VBA function (you could use the C language output as a starting point) for straightforward deployment inside an Excel spreadsheet environment. This approach could prove useful for any detailed preliminary user testing of your model before a more industrial strength or customized tool is deployed for corporate wide usage.

Minitab ▶

Getting Started

Most predictive modeling techniques give more satisfactory results if the input data are properly cleaned. MARS is no exception to this general rule. Data cleaning involves correcting data entry errors, resolving inconsistencies, and, in particular, capping, removing, or replacing outliers. Univariate outliers (*i.e.,* outliers relative to a single variable) can be detected via straightforward descriptive statistics produced optionally in every MARS run. Multivariate outliers are of less concern and can be detected in CART® runs (as outliers are typically isolated in small terminal nodes). Once the data are clean and have passed elementary tests of soundness, MARS modeling can begin. Note: CART is more resistant to data problems than MARS. If you have serious concerns regarding your data quality and reliability, be sure to conduct some CART analyses as well.

MARS General Settings

Before proceeding with the rest of the MARS modeling, consider configuring the general engine settings applicable to all future runs in the **Options** window available through the **Edit>Options…** menu. In this guide we focus on the MARS specific tab (shown below).

✓   Generic tabs are available in the **Options** dialog section (as shown below).

✓   You can also set the random number seed that will be used by MARS during any cross-validation runs (see below).
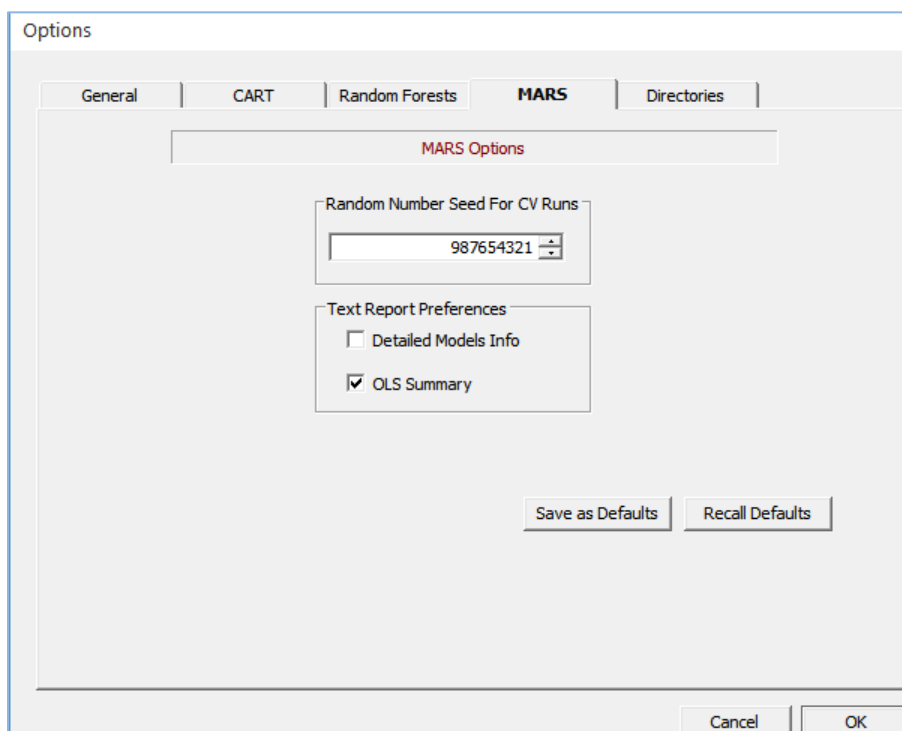
♦   You can request detailed models info to be printed in the **Classic Output** window by selecting the **Detailed Models Info** box. By default, only the optimal model details are printed.

♦   You can also request additional OLS Summary results obtained by running OLS regression on the final set of Basis functions entering into the MARS optimal model. This provides independent confirmation to all MARS coefficients calculated using internal fast updating techniques as well as provides conventional statistical results of running OLS such as t-statistics, p-values, etc.

Modeling Dataset

We will illustrate all the basic steps involved in setting up and building a MARS model by working with the Boston housing dataset available in the Sample Data folder of the SPM installation. The dataset is referenced in a number of different sources and has become manor benchmark in testing various regression algorithms. Below we include a brief description of the dataset for your reference.

↳ Harrison, D. and D. Rubinfeld. Hedonic Housing Prices & Demand For Clean Air. *Journal of Environmental Economics and Management,* v5, 81-102 , 1978

   ◆ 506 census tracts in City of Boston for the year 1970

   ◆ Goal: study relationship between quality of life variables and property values

   ◆ MV   median value of owner-occupied homes in tract ('000s)

   ◆ CRIM   per capita crime rates

   ◆ NOX   concentration of nitrogen oxides (pphm)

   ◆ AGE   percent built before 1940

   ◆ DIS   weighted distance to centers of employment

   ◆ RM   average number of rooms per house

   ◆ LSTAT   percent neighborhood 'lower SES'

   ◆ RAD   accessibility to radial highways

   ◆ CHAS   borders Charles River (0/1)

   ◆ INDUS   percent non-retail business

   ◆ TAX   tax rate

   ◆ PT   pupil teacher ratio

Minitab ▷®

To open an input data file:

♦   Select **File>Open>Data File…** menu item (or click on the **Open File** icon in the toolbar). To change or set default input and output directories, use **Edit>Options...** menu item and select the **Directories** tab in the **Options** dialog window.

♦   In the **Open Data File** window, select the dataset's file format in the **Files of type** selection box and then browse for the boston.csv file.
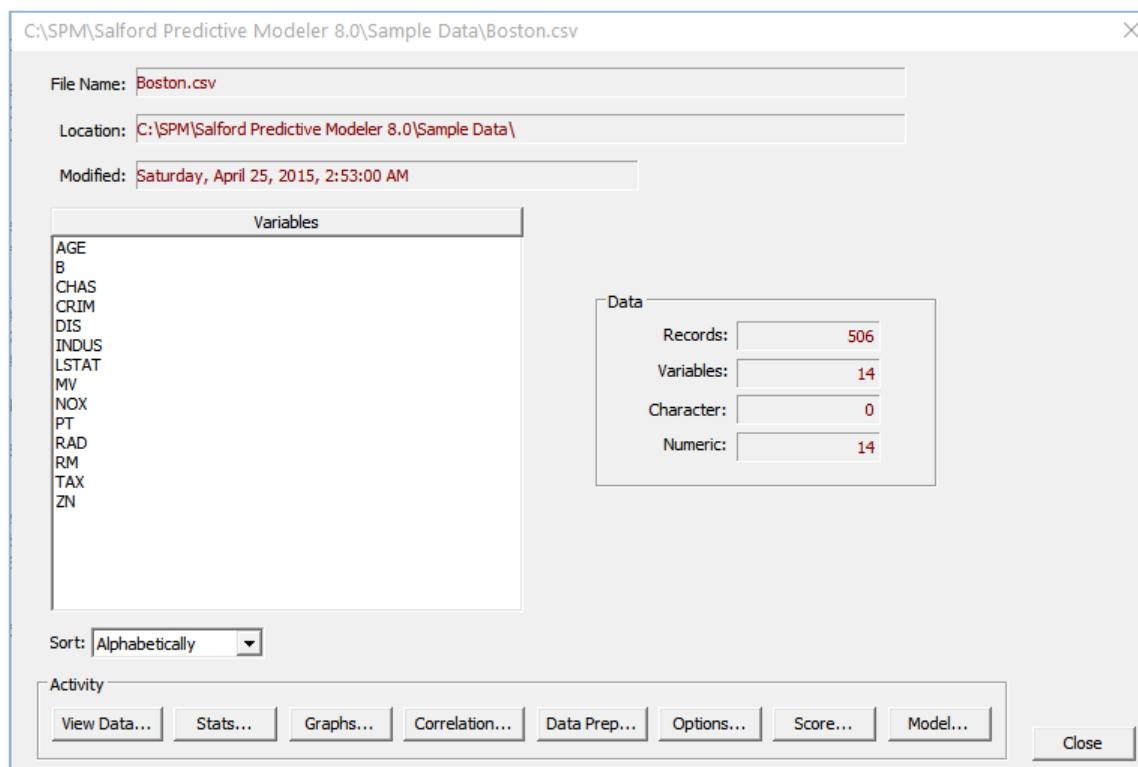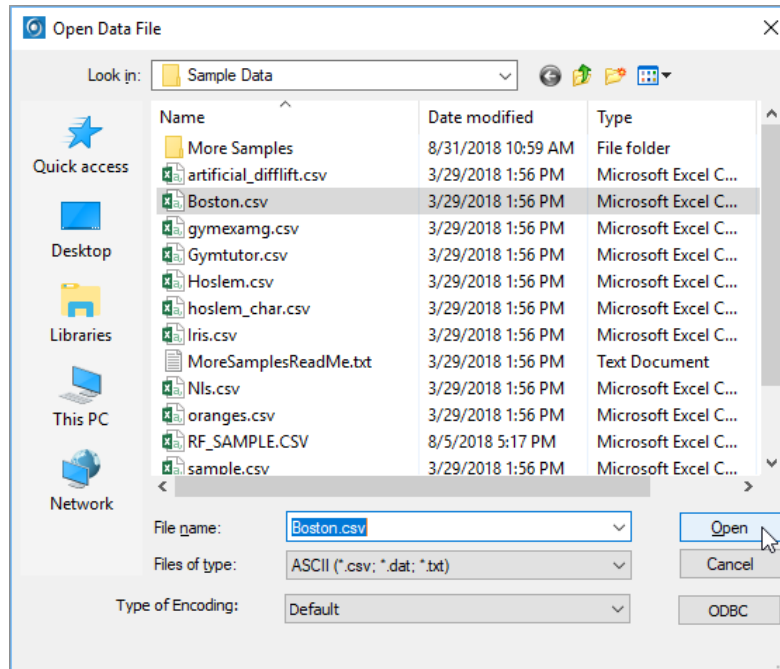
After you open a file, the **Model Setup** dialog automatically opens (you will need to click on the **Model…** button in the selected window as shown above).

Setting up MARS Model

The **Model Setup** window contains a number of generic tabs common to all available data mining engines. These tabs are described in Model Setup section. In what follows, we will focus on MARS-specific controls.

The Model tab

In the **Model** tab:

♦   Select **MARS** in the **Analysis Engine** selection box.

♦   Select **File Order** in the **Sort:** selection box.

♦   Select **Regression** radio button in the **Target Type** group.

♦   Select MV as the target variable.

♦   Select all remaining variables as (continuous) predictors.

♦   Note: to check all other variables as predictors quickly, click the "Predictor" label to highlight all variables and then click the "Select Predictors" checkbox to check all of the boxes simultaneously

Note the following additional controls available to you (not used in this example):

♦ **Categorical** column – when checked, the corresponding variable will be treated as categorical. MARS handles categorical variables by constructing dummy variables for all possible groupings of levels and then exhaustively searching for the best dummy to enter into the model constructed so far. The grouping approach is more powerful compared to traditional approaches which construct dummies only for distinct categorical levels.

✓ Variables with character values are always checked as categorical and cannot be unchecked as categorical

♦ **Weight** column – allows you to check a weight variable, if any.

♦ **Non Transform** column – allows you to disable MARS transformations of the corresponding variable, thus forcing it to enter only linearly.

✓ Checking all predictors as non-transforming will reduce MARS to a forward step-wise linear regression engine with flexible handling of categorical predictors. This is often useful in establishing a base-line linear model.

♦ **Non Interacting** column – forces the corresponding variable to enter only as a main effect even when interactions are allowed.

✓ By default, MARS builds generalized additive models (no interactions), this can be changed in the **MARS** tab (shown above).

Note on the Non Transform Variables

Forbidding transformations of selected variables is equivalent to forbidding knots. If a variable enters at all, it will have a pseudo-knot at the minimum value of the variable in the training data. There is no guarantee, of course, that the variable will be kept in the final model after the backwards deletion stage.

Reasons to forbid transformations of certain variables include:

♦ *A priori* judgment on the part of the modeler, and

♦ Variable is a score or predicted value from another model and needs to stay linear for interpretability.

If transformations are forbidden on all variables, MARS will produce a variation of a stepwise regression. This model can be used as a baseline from which to measure the benefits of transformations.

Note on the Binary Target Variables

In its current implementation, MARS is a regression procedure, meaning that it always treats the target variable as continuous. In spite of this, MARS can be used effectively to estimate binary response models, models in which the target is typically coded as 0 or 1. MARS prediction for such problems will normally be fractions between 0 and 1 and thus can be viewed as probabilities. However, nothing prevents the MARS model from predicting negative values or values larger than 1. Therefore, after reviewing binary response models the MARS predictors should be viewed as scores rather than literal probabilities.

In view of the above, MARS includes features to assist binary response modeling. To invoke them, choose **Classification/Logistic Binary** mode in the **Target Type** group.

In the logistic binary mode MARS will generate reports on the assumption that the target is either 0 or 1 or a probability between 0 and 1. Using a default threshold of 0.5, MARS will consider all target values (actual or predicted) less than 0.5 to correspond to non-response and values greater than 0.5 to correspond to response. After classifying all predicted and actual values of either 0 or 1, a prediction success (confusion matrix) table is produced cross-classifying actual vs. predicted class assignments.

Minitab ▶®

It will often be necessary to adjust the threshold dividing 0 from 1 to obtain sensible results. Use the slider found in the post-processing dialogs to change the threshold as needed. A table showing all four cells of the prediction success matrix (true positive, false positive, true negative, false negative) for each threshold from 0 .01 to 0.99 can help you decide quickly on the best threshold setting.

Remember that the binary option is a reporting feature; it does not change the way in which MARS generates its models because the underlying algorithm is not specifically adapted to the problem. Theoretically, it would be possible to get more accurate models by running a logistic regression version of MARS; however, no such version currently exists due to a number of theoretical and computational challenges such an approach would entail. *However, a true ultra-fast logistic regression MARS is currently in development at Salford Systems. Please revisit our website periodically for news regarding its release.*

The Testing tab

Aside from the common methods to partition data into train and test samples or cross-validation, there is a specific **MARS legacy mode** (selected by default). In this mode, MARS builds a sequence of models using all available data (in other words, there is no test partition, all input data is used for learning) after which the optimal model is determined based on the GCV (generalized cross-validation) criterion (a variant of the mean squared error which is penalized by model size and user-defined degrees of freedom parameter discussed later).

Given the small size of the dataset (506 observations), we will use the legacy approach in what follows.

The MARS Tab

This tab includes the key controls over the MARS engine. In what follows, we describe the meaning of each individual control in approximate order of their practical significance.

Max Basis Functions

⌨ BOPTIONS BASIS = <n>

This setting specifies how many forward steps MARS takes to generate its maximal model. Usually each step adds two Basis functions, so a limit of 40 Basis functions (which is what we will use in our example) would be reached with 20 forward steps. The default maximum Basis functions is 15 and can be modified with the up/down arrows or by entering a new value.

In the first stage of model development, MARS constructs an overly large model by progressively adding new Basis functions or splines (new main effects or interaction terms). How far MARS carries this first stage is user-specified via selection of the maximum number of Basis functions. The default setting is a maximum of 15 Basis functions, a rather low limit. You will almost always want to set this limit much higher.

A rule of thumb is that this maximum should be at least two to four times the size of the "truth"; thus, if previous experience suggests that a robust model has approximately 35 predictors, the maximum

number of Basis functions should be set to at least 70 and more likely 100. The larger this maximum is set, the longer a MARS run will take. MARS will attempt to create as many Basis functions as are allowed

Minitab ▶®

even if there is no sensible way to create that many. You should specify the maximum number judiciously, as MARS will take your limit literally.

From the numeric stability point of view, the practical limit on the maximum number of Basis functions is reached around 250. While it is possible to push it even higher on modern powerful workstations, accumulation of the round off errors may reduce the overall model performance. Therefore, applying various variable reduction techniques on contemporary datasets of very large sizes (in terms of predictors) becomes more and more relevant. Over the years we found that using a more robust (with respect to noisy data) TreeNet® data mining engine, or in some instances even the CART engine to identify an important subset of the original predictors may benefit the overall performance of the follow up MARS model as well as reduce the computational burden on the algorithm. This way, MARS will focus on the right subset of predictors from the very beginning instead of wasting valuable time and resources (and also a limited number of available Basis function "slots") on potentially noisy and irrelevant predictors that just happened to look better at some point during the model building process.

The limit should be reassessed when you increase the maximum number of allowable interactions. A main effects model can only search one variable at a time so the number of possible Basis functions is limited by the number of distinct data values; however, a two-way interaction model has many more possible functions. To ensure that both interactions and main effects are properly searched, the maximum number might need to be increased.

The number of Basis functions needed in an optimal model will depend on how fast the function is changing slope and how much the function slope changes over its entire range. The faster the change, the more knots required to track it.

Maximum Interactions

⌨ BOPTIONS INTERACTIONS=<n>

This control limits the highest degree of interaction MARS can consider. The default setting is 1, which disallows interactions; a setting of 2 would allow 2-way interactions, 3 would permit triple products of Basis functions, and so on. The interactions setting can be increased or decreased with the up/down arrows or by entering a new value.

✓ Missing values are handled automatically by creating cross products between missing and non-missing indicator functions and legitimate basis functions. Even though these formally look like interactions, they are handled by a different internal mechanism (nested basis functions) and therefore always appear in the presence of missing values regardless of the maximum interactions setting.

Ridge

⌨ BOPTIONS MARSRIDGE = <v>

Maintains MARS numerical stability. The default value is 1.e-7.

Speed Factor

⌨ BOPTIONS SPEED = [1|2|3|4|5]

When interactions are allowed, the computational burden can increase substantially due to introduction of all possible cross-products of Basis functions. The speed factor controls the trade-off between speed and accuracy in MARS' search for the best model.

The speed of 1 will force MARS to consider all possible pair of Basis functions during the interactions search. This is the slowest brute force approach which also tends to be the most accurate.

With the higher speed values, MARS will bypass the exhaustive search a number of times by considering only the best candidates (the top-ranked candidates) from previous steps as possible interaction pairs. The exact number of candidates that MARS considers is progressively reduced with each higher-speed setting. Every so often, MARS will search all candidate Basis functions and update its list of top contenders, giving every Basis function a chance to be selected. This substantially reduces the computational burden of the algorithm and may result to significant time savings during model building.

What should you do if you get different models using different speed settings? The final choice of the model is up to you. Because the speed setting simply influences how thorough MARS is in its search for the best Basis functions, it is possible that a less-thorough search would yield a better model.

✓ The speed setting affects only models with interactions. The setting makes no difference for additive MARS models (the default).

Min Obs. Between Knots

⌨ BOPTIONS MINSPAN = <n>

This control sets the number of observations required between knots, or window/bandwidth size. The default value of 0 forces MARS to manage this setting automatically using internal heuristics based on sample size and model complexity. However, you may change this control to any positive integer. The value of 1 will force MARS to consider a knot at every observed data value; thus the MARS regression may change slope or direction anywhere and as often as the data dictate. You can make MARS less locally adaptive by increasing the number of data points required between knots. Setting this parameter to a value like 20, 50, or 100 can be very useful in data mining applications. Use the up/down arrow keys to change the setting or to directly enter a new value.

By default, MARS allows a knot (*i.e.,* a new hockey-stick Basis function) to be generated at every observed data value; this default allows the MARS regression to change slope or direction anywhere and as often as the data dictate. To the extent that these knots are redundant, they will be deleted in the backwards stage.

Consider, for example, a simple model with one predictor variable LSTAT. MARS might generate a function like that shown below.

Minitab ▷®
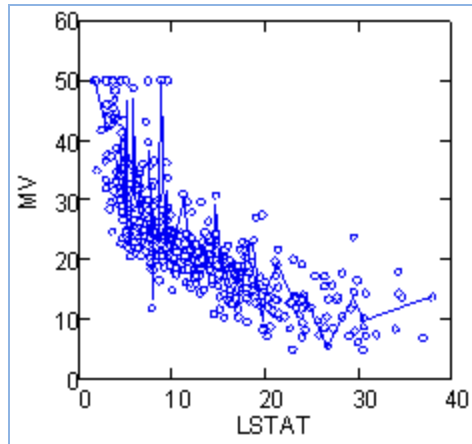
The function traces out a very jagged relationship between LSTAT and MV with many sign changes. If MARS selects such a model, it will have strong support in the data. Nevertheless, this degree of flexibility may be undesirable in many applications and a smoother albeit less locally-accurate model might be preferred.

An effective way to restrain knot placement, *i.e.,* to make MARS less locally adaptive, is to specify a moderately large minimum span or, equivalently, a minimum number of observations between knots. If, for example, the minimum number is set to 100, there must be at least 100 observations (as opposed to data values) between knots.

For data mining applications, settings as high as several hundred or more may be appropriate to restrain the adaptiveness of MARS. Even if "true" wobbles exist in the data, a high setting may also be useful as a simplifying constraint.

✓ Using an unreasonably large value will eventually force MARS to override it with its own internally computed value so that computations can safely proceed forward.

Penalty on Added Variables

⌨ `BOPTIONS PENALTY = <x>`

Since MARS produces a regression model, undesirable high collinearity between predictors is possible. To reduce the potential for collinearity you can set a "penalty on added variables, a fractional penalty for increasing the number of variables (not Basis functions) in the model.

The penalty affects the forward-stepping process, producing models containing fewer distinct variables. It causes MARS to favor reusing variables already in the model over adding new variables. As the penalty is increased, MARS tends to create new knots in existing variables or generate interaction terms involving existing variables.

Here is how the penalty copes with the multicollinearity problem. Suppose, for example, that X1, X2, and X3 are all highly correlated. If X1 is entered into the model first and there is a penalty on added variables, MARS will lean towards using X1 exclusively instead of some combination of X1, X2, and X3. If the correlation between the variables is quite high, there will be little lost in fit as a result. The penalty can also be used to encourage parsimonious models containing few variables, though they might contain many Basis functions.

Minitab ▶®

✓ The count of variables in the model is entirely distinct from the count of Basis functions. A variable could be represented by just one Basis function or by dozens. The added variable penalty induces MARS to make do with fewer variables but the final model could have quite a few Basis functions.

The default setting is "no penalty." Other choices, which can be set by clicking on the appropriate radio button, are moderate (0.05), heavy (0.10), and a user-specified penalty, which cannot exceed 1.0. The best value depends on the specific situation. You will probably have to experiment to find the best setting for your problem.

Degrees of Freedom for Knot Optimization

⌨ KNOT FIXED = <x>

This setting allows you to control the number of degrees of freedom charged for knot optimization. The penalty is used to reflect the fact that MARS conducts an extraordinarily intensive search to find the knots, creating a risk of overfitting. The penalty itself is used internally when MARS calculates and reports the GCV measure as well as the number of effective parameters associates with each model.

In the legacy mode (as per the **Testing** tab) MARS selects the optimal model based on the smallest GCV measure while the entire dataset is used in training. Thus, penalizing each knot helps MARS select an honest rather than over fit the model in the absence of the test sample.

The default degrees of freedom penalty setting is fixed at three. This setting can be changed to any non-negative number by entering a new value. Setting the penalty higher will tend to favor a smaller optimal model. Thus, you can influence the size and complexity of your model by setting this parameter higher or lower. If MARS is estimating null models or models containing only a handful of Basis functions, you can encourage larger models by fixing the degrees of freedom parameter at a low number such as 1, 2 or 3. Conversely, if MARS is including too many parameters or knots for your liking, you can force a smaller number by setting this parameter quite high.

✓ Using values as high as 10, 30 or even 200 may be necessary to estimate a model of suitable size.

When the test sample is available or genuine cross-validation is used (the non-legacy options in the **Testing** tab), the degrees of freedom setting is of lesser importance as it only changes the reported GCV and the number of effective parameters based on the learn sample, while the optimal model is selected based on the test sample (or cross-validated) performance.

✓ This means that changing the degrees of freedom penalty in the non-legacy mode does not change the optimal model picked by MARS.

Regardless of the official optimal model picked by MARS (determined by the DF penalty in the legacy mode or based on the Test/CV sample performance), you can always select models of different sizes manually by looking at the backwards deletion sequence (described in the next section).

Minitab▶®

The Interactions Tab

When the **Maximum Interactions** control is set to 2 or higher in the **MARS** tab, MARS will consider all possible interactions between selected predictor variables (unless they are explicitly declared as **Non Interacting** in the **Model** tab). The **Interactions** tab allows you to limit or restrict pair-wise interactions between any variable and any other variable in the model. This tab (accessible only after predictor variables are selected in the **Model** tab and interactions are turned on in the **Options and Limits** tab) displays the selected predictor variables in a matrix, with each predictor appearing as both a row and a column header.

✓ If you checked the **Non Interacting** box in the **Model** tab for a particular predictor variable, that particular variable will not appear in the **Interactions** dialog.

To disallow a pair-wise interaction, click on the cell that corresponds to the variable pair you would like to restrict. The selected pair will be unchecked (symmetrically in two boxes in the interactions matrix).

You can also highlight multiple rows (or columns) in the matrix, and then jointly check or uncheck multiple boxes by pressing the **Check** or **Uncheck** buttons.

Saving the Model

Finally, press the **Save Grove…** button to save the MARS model about to be built into a binary grove file. The grove file can be used later to view, score, and translate the model.

✓ At this point only the intent to save a named grove file is communicated to the engine. The actual file will be created only at the end of the model building process initiated by pressing the **Start** button.

This step is optional, when the grove file is not specified MARS will always create a temporary grove at the end of any modeling run. You can then save the temporary grove into a permanent named grove using the controls available in the **MARS Results** window (described later).

💣 The temporary grove is automatically deleted once the application is closed.
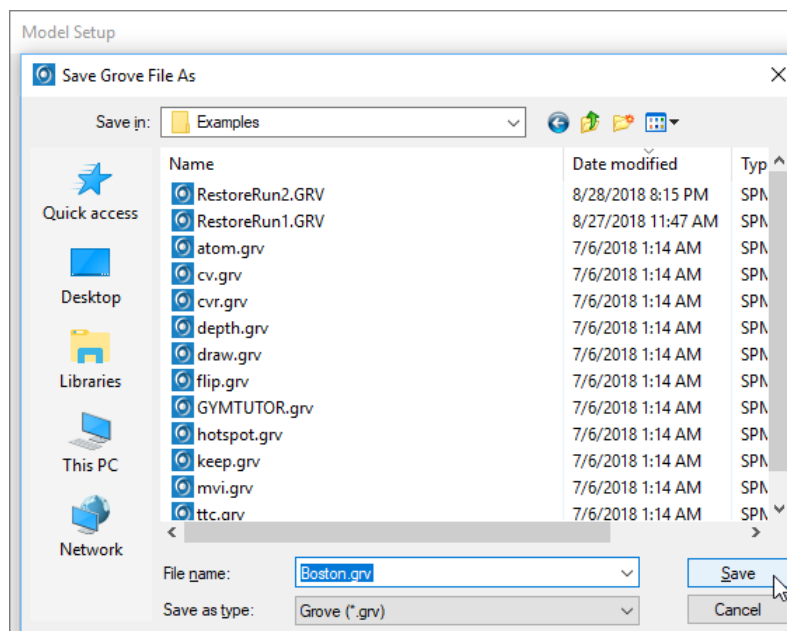
Starting the Model

Simply press the **Start** button to initiate the model building process.

Minitab ▶

Viewing the MARS Model

When the MARS engine finishes, **MARS Results** window appears on the screen.
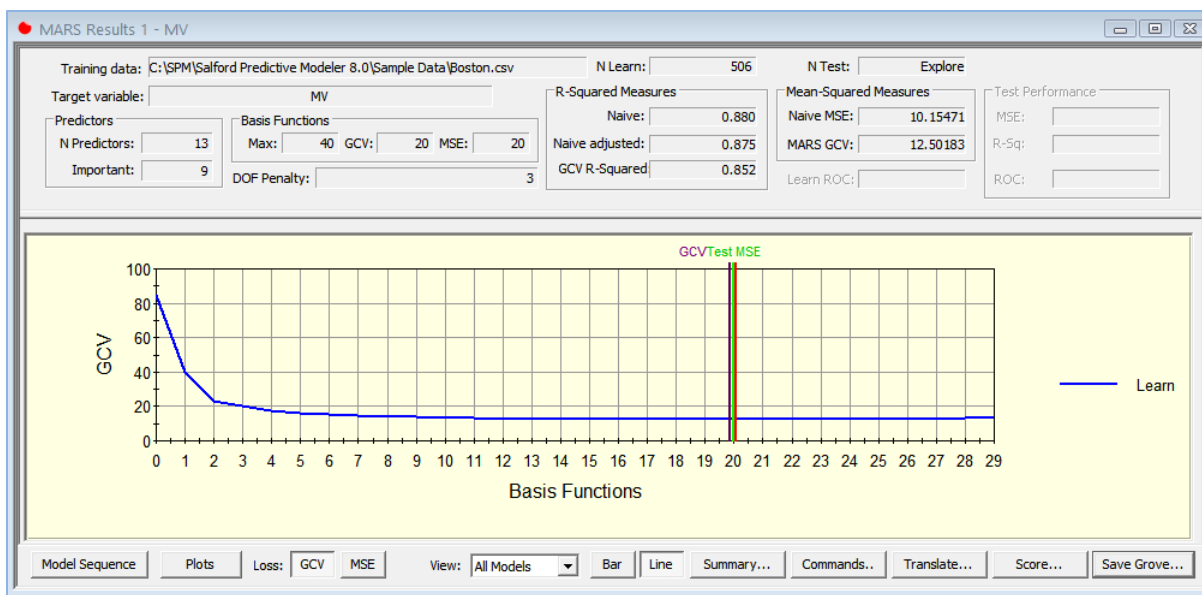
The top of the window includes basic stats about the model, like the name of the target variable, the number of original predictors, number of predictors participating in optimal model (**Important**), the number of learn records (506), the number of test records (Explore - because in the legacy mode the entire dataset is used for learning), requested number of Basis functions (40), optimal model size in basis functions (20) as well as a number of standard performance measures such as Naive and Adjusted R-square, Mean Squared Error (MSE), and penalized version of it known as GCV R-Squared.

✓ Note the default degrees of freedom penalty (DOF) set at 3.

The lower part of the results window contains the model performance in terms of GCV indexed by the model size in Basis functions. The optimal model is defined as the minimum GCV model and has 20 Basis functions (highlighted by the vertical line).

Note the large flat area on the GCV curve around the optimal model. This offers you a number of possible alternative models which might be nearly as good as the optimal model. You may use your judgment and outside knowledge to decide which model size suits you best.

Switch between the GCV and MSE performance curves by pressing the **GCV** or **MSE** buttons at the bottom. Note that the MSE curve is strictly decreasing thus always favoring the largest model in terms of Basis functions.

Just like anywhere else, we display the Naive MSE, which simply divides the sum of squared errors by the sample size without the degrees of freedom adjustment. This simplifies comparisons across multiple engines and avoids the largely controversial topic of calculating the actual number of degrees of freedom consumed in any exhaustive search algorithm.

✓ The GCV measure was designed to take into account the actual degrees of freedom consumed, which is why it is used by MARS to find the optimal model size in the legacy mode (when no test partition available).

Model Summary

Press the **Summary…** button in the **MARS Results** window to open detailed summaries for the currently selected optimal model with 20 Basis functions. The generic tabs shown on the **MARS Results: Summary** window are described in Summary window section. Below we focus on the MARS specific summary tabs.

Basis Functions Tab

This tab shows the complete analytical expression of the selected model using SAS-compatible language. The top section includes all Basis functions that entered into the model. The middle section shows the model equation as a linear combination of Basis functions. The last section shows a simplified equation listing only the variable names.

In essence, this is the final model code which can be studied and easily extracted.

Final Model Tab

The **Final Model** tab displays the same information as the **Basis Functions** tab, except in a table form. Right click your mouse anywhere within the table and you can copy the entire contents into the clipboard for further use. Most likely you will find that the **Basis Functions** tab provides a more convenient way to study MARS model compared to the table form.

MARS Results 1 - MV: Summary for 20 Basis Functions

	Resid. Trim Cnt	Resid. Outliers CUM %	Resid. Outliers Counts	Resid. Box Plot	ANOVA Decomp.
Summary	Dataset	Gains	Variable Importance	Resid. Trim %
Final Model	Basis Functions	Plots		

Final Model After Backward Stepwise Elimination

Basis Function	Coefficients	Variable	Sign	Knot
0	26.59272			
1	-0.57175	LSTAT	+	6.07000
2	2.00549	LSTAT	-	6.07000
3	7.63026	RM	+	6.43100
5	-9.30229	DIS	+	1.42540
6	46.04399	DIS	-	1.42540
7	-0.11963			11.16040
8	0.72349			11.16040
9	-1.43028	PT	+	12.60000
10	-20.90391	NOX	+	0.48800
13	0.03157	TAX	-	300.00000
14	1.12042	RAD	+	1.00000
17	-0.08820	B	+	386.75000
21	-0.84987	RAD	+	3.00000
23	0.33680	LSTAT	+	23.34000
25	93.84077	RM	+	7.39300
27	0.94940	PT	+	16.60000
29	8.17157	DIS	+	1.74550
31	-82.71466	RM	+	7.52000
33	-19.23891	RM	+	7.18500
37	0.87970	DIS	+	5.21460

Minitab ▶

ANOVA Decomposition Tab

This tab provides important insights into structural significance of individual MARS components. Compared to tree modeling engines, MARS models are somewhat unique because they offer classical model structure as a linear combination equation. Therefore, any MARS model can be decomposed into a collection of structural components spawned by individual variables or groups of variables.
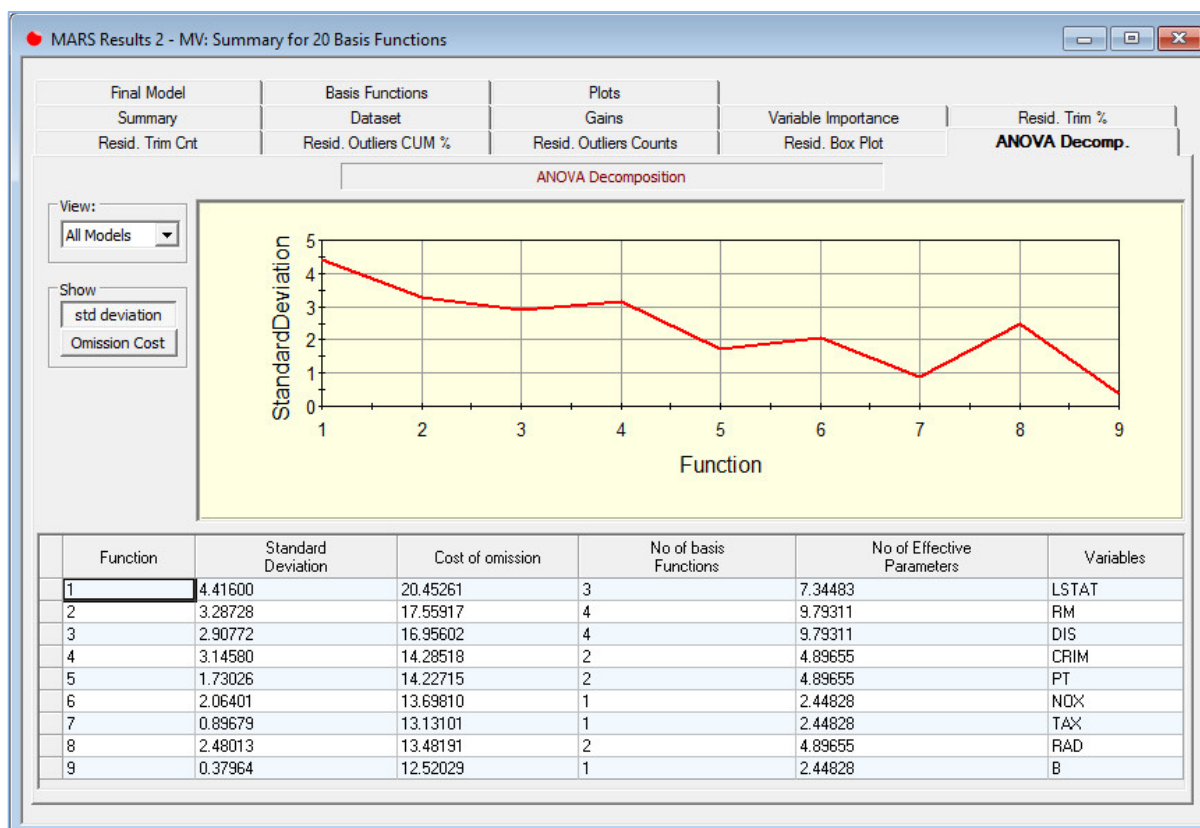
In the absence of interactions (the default), MARS model is decomposed into a set of univariate components defined by the originating variable. For example, looking into the **Basis Function** tab you may observe that only three Basis functions (BF1, BF2, and BF23) were created using the variable LSTAT. This group of Basis functions is reported in the first row of the ANOVA table and analyzed as an independent structural component. Note that it has the standard deviation of 4.416, which is also reported on the graph above.

Here is the complete list of entries available in the ANOVA table:

♦ **Function** – group number, all groups are sorted according to the cost of omission (see below).

♦ **Standard Deviation** – standard deviation of the group treated as a whole.

♦ **Cost of Omission** – the increase in the GCV when the group is removed from the model and the remaining basis functions are refit to the original target using the OLS regression. Thus, the cost of omission shows the unrecoverable loss of model performance due to variable exclusion from the existing model.

♦ **No of Basis Functions** – number of Basis functions in the group.

♦ **No of Effective Parameters** – internally computed MARS measure combining the degrees of freedom penalty (set by the user) and the type and number of Basis functions in the group to

approximate the model complexity in terms of parameters taking into account the exhaustive search paradigm.

✓ This measure is used by MARS in order to compute GCV and reported for completeness.

♦ **Variables** – the name of the variable originating the group.

✓ When interactions are allowed, some groups are defined based on multiple interacting variables listed in this column.

Press the **std deviation** or **Omission Cost** buttons to switch between two different graphs in the upper part of the display.

Variable Importance Tab

This tab lists variables according to their importance scores, which are calculated on the 100% scale such that the most important variable always gets the 100% score.
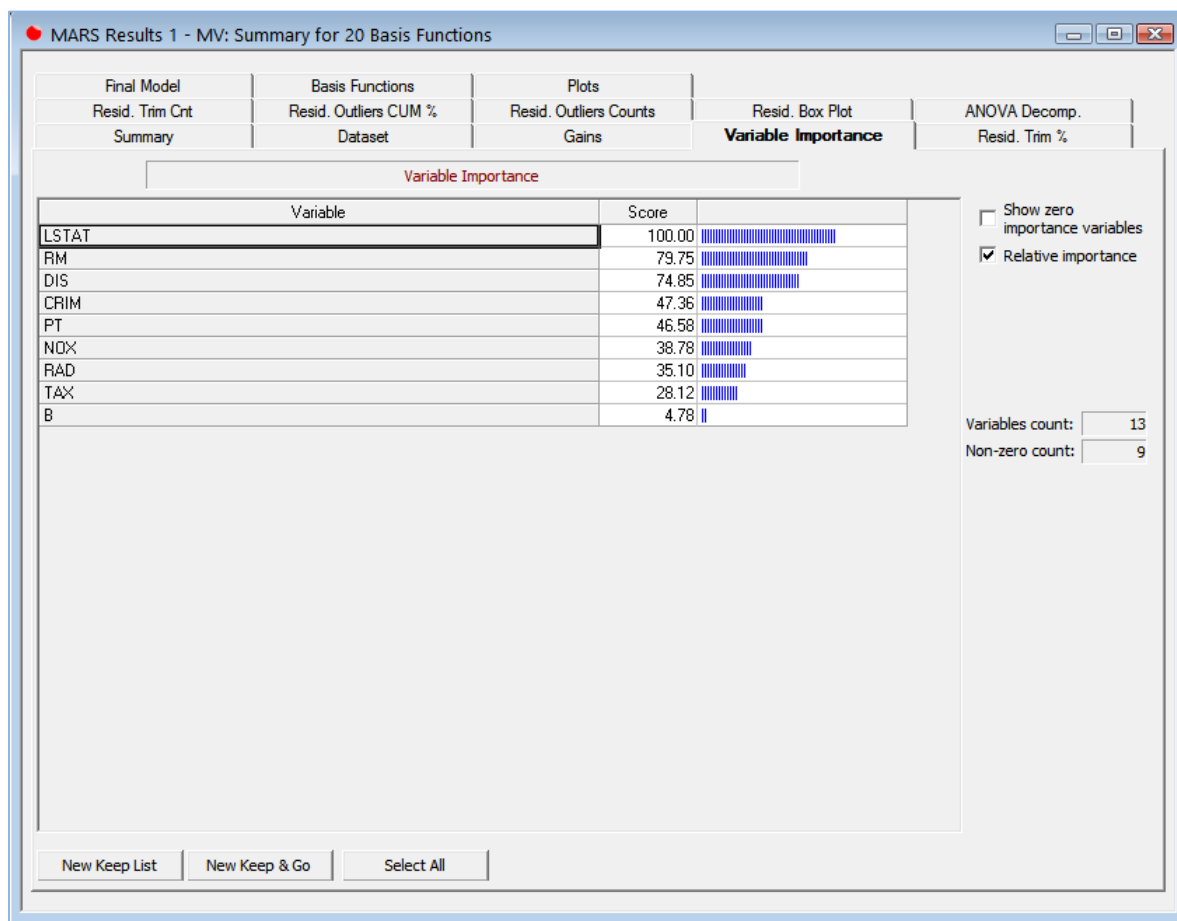
Each importance score is based on an internally computed raw variable importance score, which is turn represents the increase in the model GCV once all basis functions containing the specific variable (either by itself or as part of an interacting group) have been dropped and the remaining Basis functions refit to the original target using OLS regression.

When no interactions are allowed, the raw variable importance is equivalent to the cost of omission already presented in the **ANOVA Decomposition** tab. When interactions are present, the raw variable

importance score cannot be easily derived from the ANOVA decomposition table because multiple ANOVA groups (main effects and interactions) will be dropped simultaneously before the model is refit.

Thus, compared with tree-based engines MARS variable importance approach is quite different because it explicitly removes the variable-specific part of the model with a follow-up model refit to access the actual damage. There is no simple way to remove a part of a tree spawned by any specific variable without inflicting irreparable damage to the underlying tree logic.

Plots Tab

The Plots tab graphically depicts univariate and bivariate ANOVA groups comprising the MARS model.

✓ This tab is only available for additive and 2-way interactions MARS models. Higher order interactions models are difficult to interpret and the graphs are disabled.

Click on the **Show All** button to open all available plots in a single window.
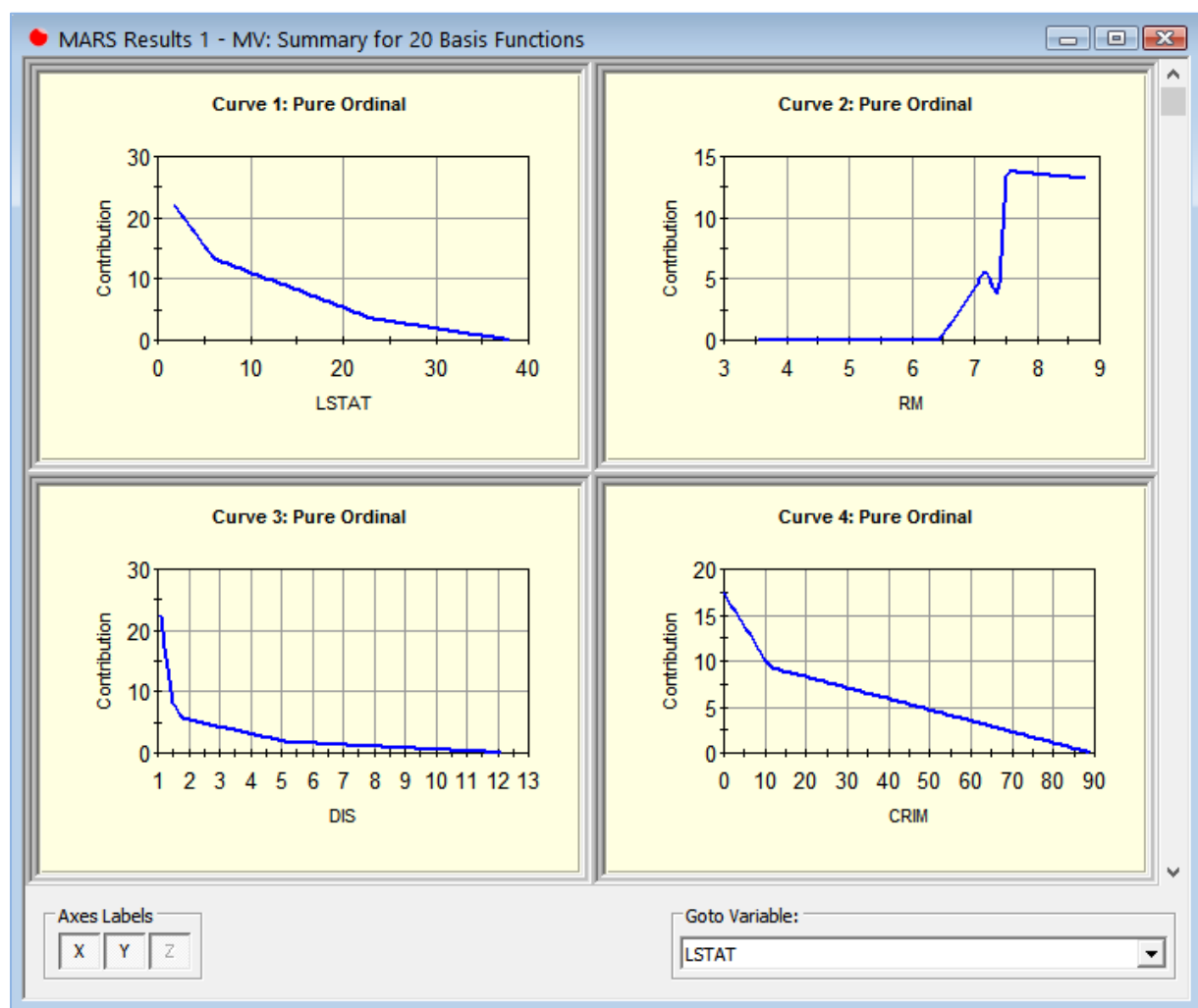
Observe how individual Basis functions are combined to form piece-wise linear transformations of predictors.

Double click on an individual graph to open it in a separate window.

Minitab >

Right click a graph to export it into an image file, copy it to the clipboard, or add it to the report.

✓   Note that there are no bivariate plots when no interactions are present.

Picking an Alternative Model

All the summaries we have looked at so far were for the optimal GCV model having 20 Basis functions. You may have noticed that the RM dependency plot has an extra "kink" between 7 and 8. In cases like this, you may wish to override MARS suggested optimal model with a smaller or in some cases larger model. Fortunately, MARS allows you to easily navigate and see models of different sizes indexed by the number of Basis functions. Just click anywhere within the graph area in the **MARS Results** window, the new model will be marked using the red vertical line. You can now press the **Summary…** button again to have all displays updated for the new model in focus.

Alternatively, press the **Model Sequence** button to open **MARS Results : MARS Selector** window containing additional summary information on each available model in a nice table form.
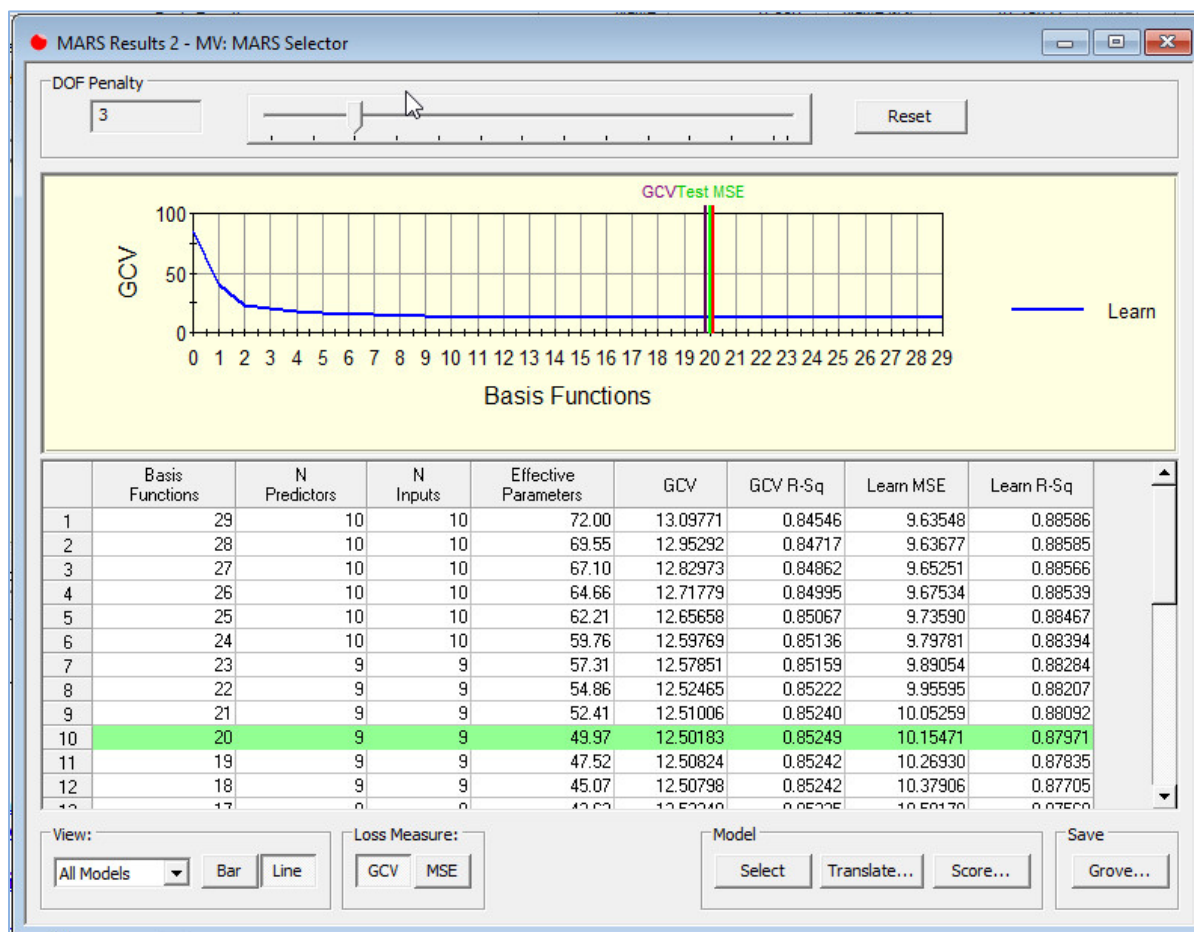
Double-click on the line for the 15-Basis function model to request detailed summaries. Switch to the **Plots** tab and press the **Show All** button to see dependence plots for the new selected model (which has fewer Basis functions than the original model).
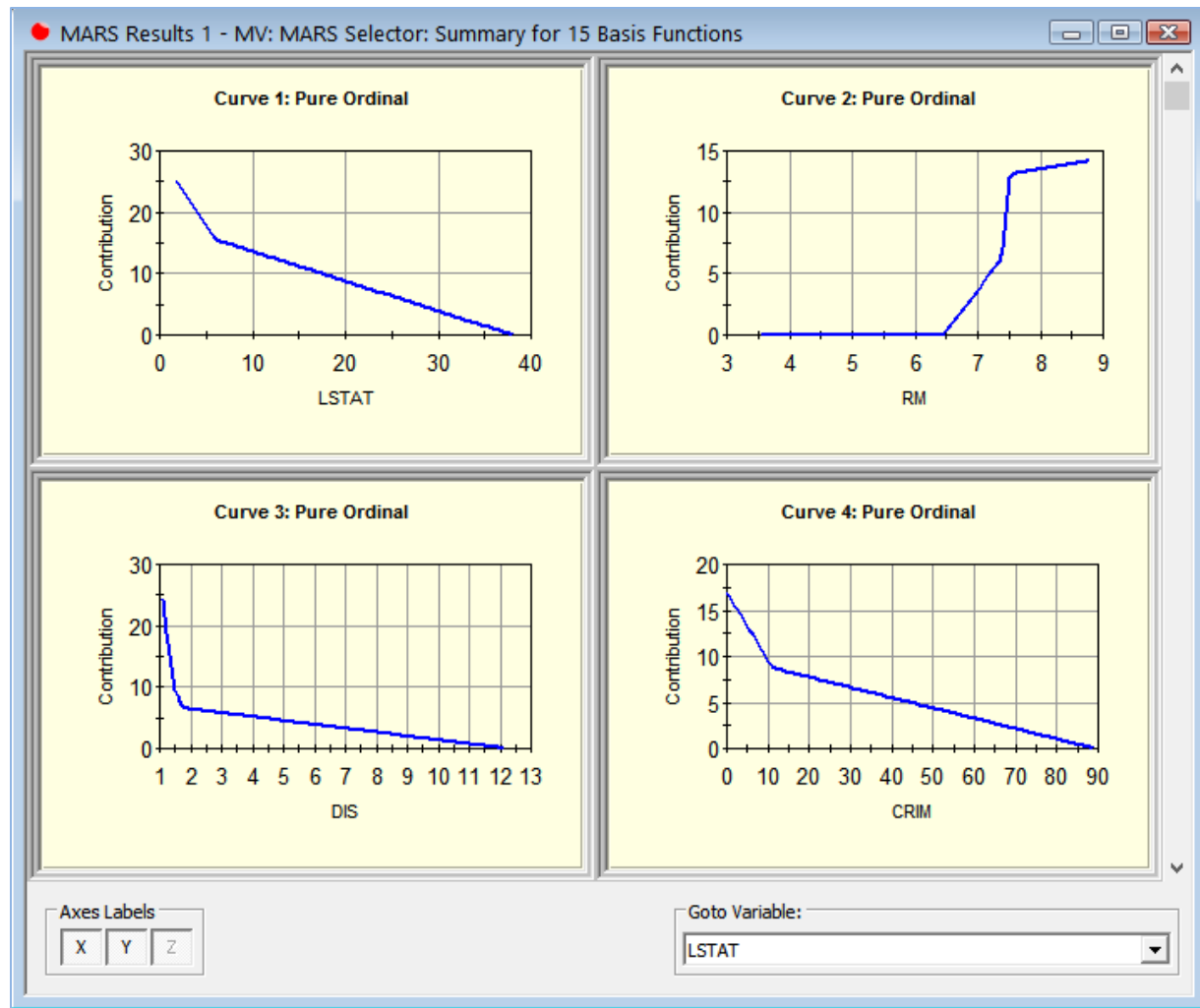
Observe the simpler shape of the RM plot more in agreement with our expectations. Also observe that some of the remaining plots have been simplified also because overall 5 Basis functions were removed compared to the optimal model.

You can easily navigate back and forth among the models of different sizes. Each model can be viewed, scored, and translated. Use your own judgment combined with outside knowledge of the problem you are studying to make final decisions on the optimal model size. MARS may give you an educated guess, but the final choice is yours.

MARS Interactions

MARS allows both global control over the maximum degree of any interaction and local control over any specific pair-wise interaction. The global control in the **Options and Limits** tab can be used, for example, to allow two-way or three-way interactions. (The default setting is no interaction).

If you simply permit MARS to introduce interactions into a model, MARS will consider all interactions between Basis functions constructed from any variables. Depending on your type of problem, you may wish to forbid some specific interactions. MARS allows you to either exclude a variable from any interaction or to specify in detail precisely which interactions are allowed and which are forbidden.

The **Interactions** tab provides a matrix with all candidate predictor variables appearing in both row and column headers. Any cell in this matrix can be set to disallow a *local* interaction; for example, an interaction between INDUS and RAD may be disallowed in any context (two-way, three-way, etc.).

Specific variables can be easily excluded from *all* interactions by checking the variable row in the **Non-Interacting** column on the **Model** tab. Thus, for example, INDUS could be prohibited from interacting with any other variable but interactions involving other variables in the model allowed.

Developing a robust MARS model is a sequential process not unlike developing a parametric regression model. The difference is that MARS does much of the work for you. We recommend you start by not allowing any interactions; this default will develop a "main effects" model that will be the most easily understood. For such a model, MARS will search for an optimal transform for each variable via its decomposition into Basis functions, and the final model will be a sum of coefficients multiplying single Basis functions.

Examine the main-effects model—what story does it tell? Is it plausible? How does the model fit the data? As there is no guarantee that a main-effects model will be adequate for predictive accuracy or a faithful representation of the true data generation process, it is necessary to then experiment with higher-order interactions. Allow two- and three-way interactions to see if the model fit can be substantially improved.

In some modeling situations, you may discover that introducing higher and higher order interactions improves model performance but renders the interpretation extremely cumbersome (plots are not available for 3-way interactions and higher). Often it is a result of multiplicative nature of the dependency. For example, a number of models in Econometrics are introduced as products of positive factors or their functions. In this case, a simple log-transform of the target variable will change the multiplicative nature of the dependency into additive, thus allowing MARS to easily handle it using low order interactions and perhaps even no interactions at all!

✓   While it may be extremely beneficial to consider transformation of the target variable as in the example above, trying to transform individual predictors is of a lesser value due to MARS' natural ability to automatically find the transformations using the Basis functions machinery.

Finally, the TreeNet engine has very powerful interaction enforcing and reporting machinery which can further clarify which interactions are actually needed. This way, TreeNet can be used to identify both the important predictors and the minimum required interactions structure, while follow up MARS run will then reduce the complexity of a TreeNet model by casting the relationship as a simple linear combination of Basis functions.

Minitab ▶®

Interactions Example

We proceed by allowing two-way interactions in the Boston Housing dataset.

♦   Set up model parameters as described in the beginning of this chapter.

♦   Set the **Maximum Interactions** control in the **MARS tab** to 2.

Press the **Start** button to initiate modeling process.

Note that the R-squared is now in the low nineties indicating a very close fit to the learn sample.

Now press the **Summary…** button to view modeling details.

First note that the **Basis Functions** tab now includes cross-products of Basis functions based on different variables thus naturally introducing two-way interactions.

MARS Results 2 - MV: Summary for 30 Basis Functions

Resid. Trim Cnt	Resid. Outliers CUM %	Resid. Outliers Counts	Resid. Box Plot	ANOVA Decomp.
Summary	Dataset	Gains	Variable Importance	Resid. Trim %
Final Model	**Basis Functions**	Plots		

Basis Functions

```
BF1  = max( 0, LSTAT - 6.07);
BF2  = max( 0, 6.07 - LSTAT);
BF3  = max( 0, RM - 6.431);
BF4  = max( 0, 6.431 - RM);
BF5  = max( 0, NOX - 0.647) * BF3;
BF6  = max( 0, 0.647 - NOX) * BF3;
BF9  = max( 0, DIS - 1.4254);
BF10 = max( 0, 1.4254 - DIS);
BF12 = max( 0, 19.6091 - CRIM);
BF14 = max( 0, 19.1 - PT) * BF12;
BF15 = max( 0, CRIM - 0.00631976) * BF10;
BF16 = max( 0, B - 0.320007) * BF10;
BF17 = max( 0, LSTAT - 29.29) * BF4;
BF19 = max( 0, NOX - 0.693) * BF1;
BF20 = max( 0, 0.693 - NOX) * BF1;
BF21 = max( 0, AGE - 18.8) * BF3;
BF22 = max( 0, 18.8 - AGE) * BF3;
BF23 = max( 0, NOX - 0.77) * BF4;
BF24 = max( 0, 0.77 - NOX) * BF4;
BF25 = max( 0, TAX - 233);
BF26 = max( 0, 233 - TAX);
BF27 = max( 0, RAD - 3);
BF28 = max( 0, 3 - RAD);
BF29 = max( 0, AGE - 98.8);
BF30 = max( 0, 98.8 - AGE);
BF31 = max( 0, B - 232.6) * BF30;
BF32 = max( 0, 232.6 - B) * BF30;
BF33 = max( 0, PT - 18.6) * BF26;
BF35 = max( 0, INDUS - 10.81) * BF2;
BF38 = max( 0, 5.631 - RM) * BF29;
BF39 = max( 0, NOX - 0.385) * BF10;
BF40 = max( 0, B - 0.320007) * BF2;


Y = 17.6478 - 0.637791 * BF1 + 22.8506 * BF2 + 17.4943 * BF3
            + 1.72486 * BF4 - 181.449 * BF5 - 16.2294 * BF6
            - 0.692922 * BF9 + 655.688 * BF10 + 0.264521 * BF12
            + 0.030466 * BF14 - 7.22908 * BF15 - 1.0026 * BF16
            + 0.475823 * BF17 + 2.14556 * BF19 + 2.66248 * BF20
            - 0.0973675 * BF21 - 0.429587 * BF22 - 55.5152 * BF23
            - 18.0597 * BF24 - 0.00575335 * BF25 + 0.172521 * BF26
            + 0.200144 * BF27 - 0.968007 * BF28 + 0.000335597 * BF31
            - 0.000732027 * BF32 - 0.223715 * BF33 + 0.280773 * BF35
            + 3.80426 * BF38 - 530.687 * BF39 - 0.0563924 * BF40;


MODEL MV = BF1 BF2 BF3 BF4 BF5 BF6 BF9 BF10 BF12 BF14
           BF15 BF16 BF17 BF19 BF20 BF21 BF22 BF23 BF24 BF25
           BF26 BF27 BF28 BF31 BF32 BF33 BF35 BF38 BF39 BF40;
```

SAS | Basic

Now switch to the **ANOVA Decomposition** tab and observe that the model now has groups spawned y two variables (interactions) in addition to the univariate groups (main effects).

Minitab ▷®

Function	Standard Deviation	Cost of omission	No of basis Functions	No of Effective Parameters	Variables
1	20.04770	10.72901	2	5.30000	LSTAT
2	7.03136	8.93589	2	5.30000	RM
3	20.40346	9.52355	2	5.30000	DIS
4	1.36501	7.54969	1	2.65000	CRIM
5	1.87114	8.78386	2	5.30000	TAX
6	1.86026	7.69461	2	5.30000	RAD
7	2.04511	9.43407	4	10.60000	NOX, RM
8	0.97216	7.78254	1	2.65000	CRIM, PT
9	2.99113	8.60476	1	2.65000	CRIM, DIS
10	11.57350	7.55905	1	2.65000	DIS, B
11	0.61561	7.30945	1	2.65000	RM, LSTAT
12	2.15422	8.28351	2	5.30000	NOX, LSTAT
13	2.41468	7.63717	3	7.95000	RM, AGE
14	1.67393	8.20076	2	5.30000	AGE, B
15	0.70715	7.47304	1	2.65000	TAX, PT
16	0.89624	7.55007	1	2.65000	INDUS, LSTAT
17	4.58956	7.55613	1	2.65000	NOX, DIS
18	17.16660	7.23638	1	2.65000	B, LSTAT

Note the **Cost of Omission** column showing the increase on model GCV when the corresponding group is dropped out from the model description and the resulting model is refit to the original target.

Now switch to the Variable Importance tab for an alternative look at the model decomposition:

Variable	Score	
RM	100.00	‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖
LSTAT	73.92	‖‖‖‖‖‖‖‖‖‖‖‖‖‖
DIS	73.29	‖‖‖‖‖‖‖‖‖‖‖‖‖‖
NOX	56.08	‖‖‖‖‖‖‖‖‖‖
CRIM	50.74	‖‖‖‖‖‖‖‖‖
AGE	40.93	‖‖‖‖‖‖‖
TAX	38.09	‖‖‖‖‖‖
B	36.55	‖‖‖‖‖‖
PT	31.14	‖‖‖‖‖
RAD	22.04	‖‖‖
INDUS	18.76	‖‖‖

The relative score is based on the raw scores obtained by dropping all Basis function terms that include the referencing variable and then refitting the rest of the model to the original target. Note that all interacting terms containing the referencing variable are also dropped because of the practical impossibility to separate them into individual stand-alone components.

Finally, switch to the **Plots** tab and then press the **Show All** button to see all plots in one window.
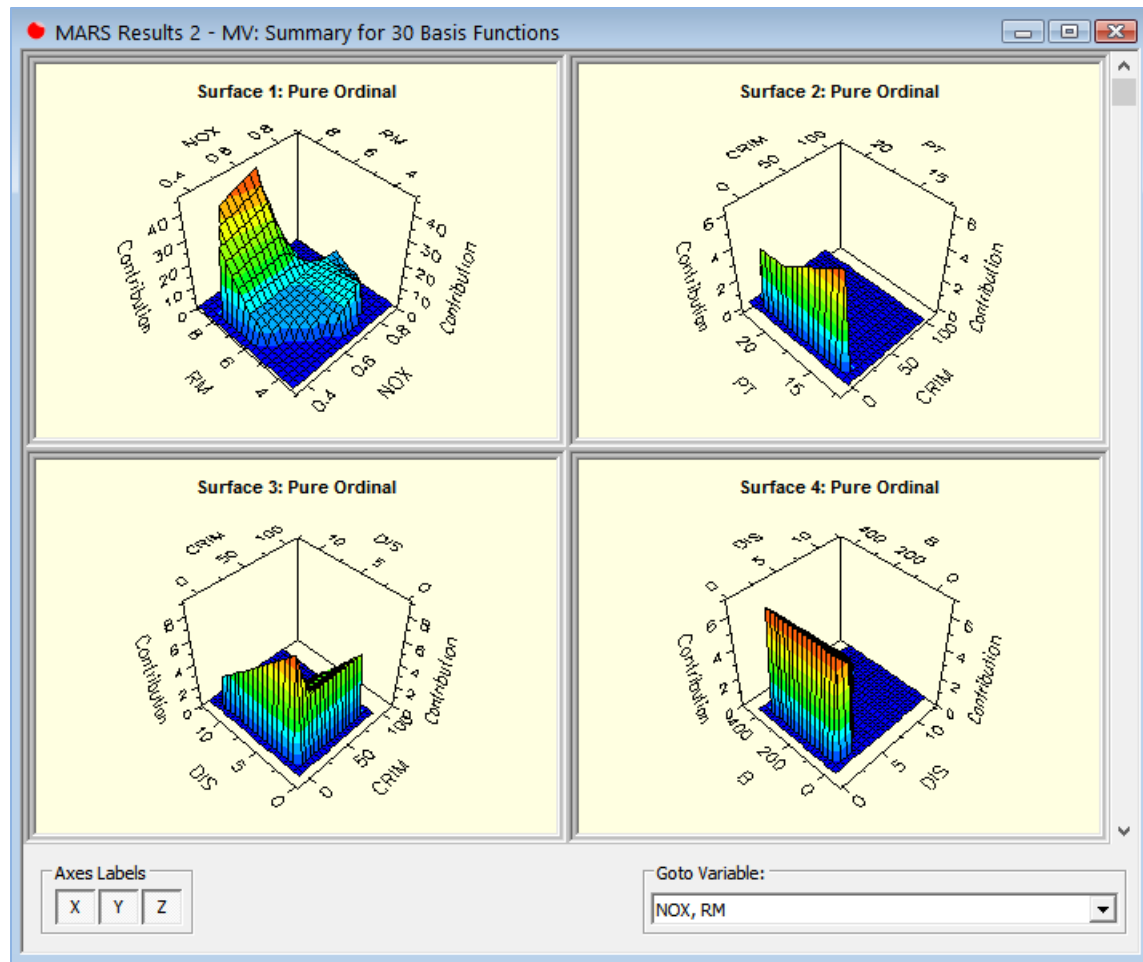
Note the 3-D plots corresponding to the interacting pairs of variables.

Click on the RM x NOX plot to view it in a separate window:
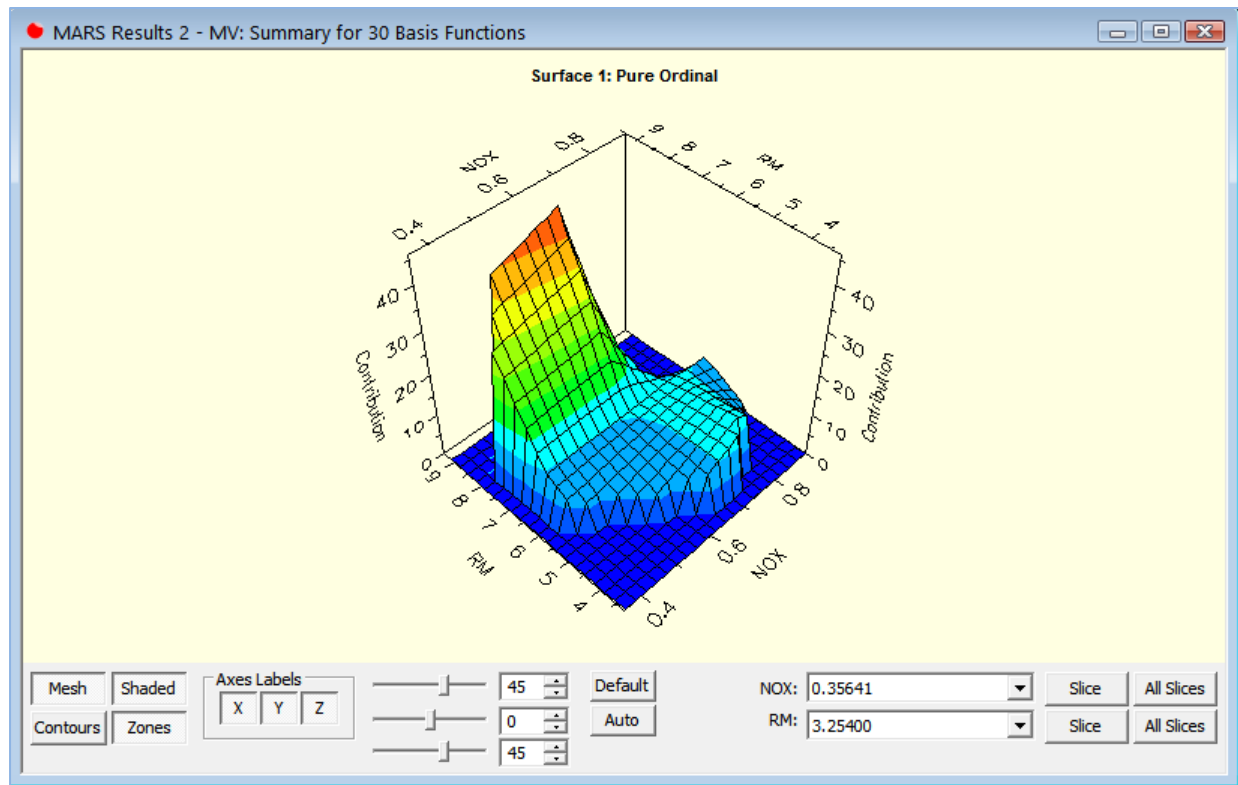
The **Contribution** values are calculated by assembling all terms (Basis functions) based on RM-only, NOX-only, or cross-product of RM and NOX.

✓   Only the specific RM x NOX cross products are included, all remaining cross-products like NOX x LSTAT etc. are not included in generating this graph (they will be included on their own graph).

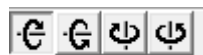The main plot area is based on the convex hull of the given variable pair in the learn sample, areas of the plot marked in dark blue lie outside of the convex hull and have no observed values in the learn sample.

Use the rotation controls   to change the viewing angle of the plot.

Use the   controls to turn on or off individual axis labels.
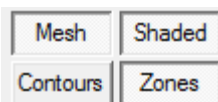
Use the   controls to change the rendering method and level markers on the graph.

Finally, press the **Slice** or **All Slices** buttons to see plot slices at different values. This may provide further help in determining where individual interactions actually occur and to what extent.

Right-mouse-click on any graph if you need to save or export it into a graphics format.

Minitab ⊳®

Modeling Binary Response in MARS

The original intent behind MARS development was is correcting some of the key deficiencies of regression trees in CART. Later it was discovered that MARS can also be quite useful in modeling binary targets coded as 0 and 1. MARS-based binary classification models are unique in the following way:

♦ Direct control over the order and structure of interactions, including main effect only models (cannot be accomplished with regression trees).

♦ Piece-wise linear continuous response (as opposed to piece-wise constant discontinuous response in regression trees).

♦ Model shown in a simple equation form utilizing Basis functions – very similar in spirit to the classical linear regression.

The only major deficiency of MARS models is that they are built using least squares loss function directly interpreting the target variable as a continuous variable with only two observed values 0 and 1. Thus, nothing guarantees that the MARS model predictions will be constrained to the [0, 1] interval. That is, they cannot be interpreted directly as probabilities and should be treated as generalized probability scores. However, this is usually not a real issue because most modern applications typically focus on the rank order of predictions or simple classification. It is also possible to refit MARS model in a true probability space by running a follow-up step-wise logistic regression using MARS-constructed basis functions as predictors.

With the above important considerations taken into account, MARS may be a very powerful approach to model binary response.

Example of Binary Response Modeling

We illustrate this approach using the spambase.csv dataset (see the **Model Setup** window described earlier):

♦ Open the spambase.csv dataset.

♦ Select TARGET as the target variable.

♦ Select all remaining variables as predictors.

♦ Select **Classification/Logistic Binary** as the **Target Type**.

♦ Set fraction of cases selected at random for testing at 0.5.

♦ Set **Maximum Interactions** to 1 (main effects only).

♦ Set **Max Basis Functions** to 50.

♦ Press the **Start** button to initiate modeling process.

Minitab▸®
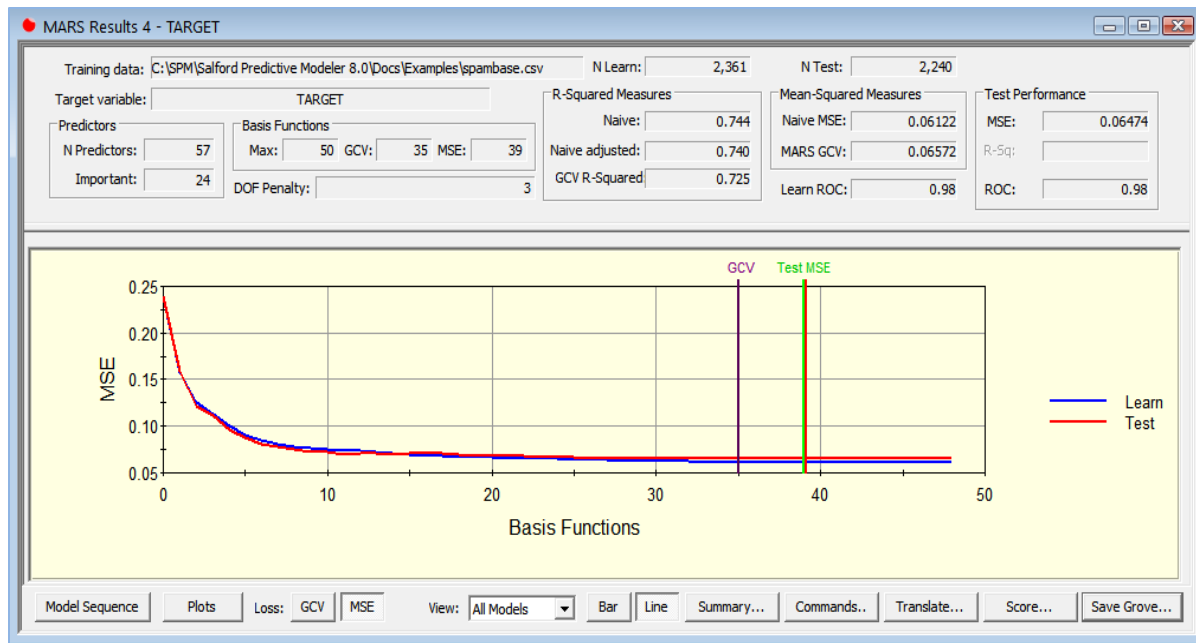
Note the two MSE performance curves reflecting performance on learn and test partitions (up until now we used the legacy MARS mode which had no test sample). Both curves are in excellent agreement with each other indicating absence of overfitting.

Press the **Summary…** button to see the **MARS Results** window. Observe new tabs like **Gains/ROC** and **Confusion Matrix** directly related to the binary nature of the target. Switch to the **Confusion Matrix** tab and press the **Balance** button.

Note the excellent classification performance (94%-95% accuracy in each of the two target classes) of the MARS model based on the independent 50% test sample.

MARS model produces continuous score as a result of combining a number of Basis functions. The Threshold value is then used to make the final class assignment. In this example, all records with the score above 0.415 are assigned to class 1 (spam) while all of the remaining records are assigned to class 0 (no spam).

You can manually change the value of the threshold to achieve different trade-off between sensitivity and specificity.

Alternatively, press the **Show Table…** button to see detailed tabulation of sensitivity versus specificity for different values of the classification threshold.

MARS Results 4 - TARGET: Summary for 39 Basis Functions: Threshold table

Threshold	Class 0			Class 1			% Missclass
	N Correct	N Misclass	% Missclass	N Correct	N Misclass	% Missclass	
-1.08168	0	1,360	100.00000	880	0	0.00000	60.71429
-1.05495	1	1,359	99.92647	880	0	0.00000	60.66964
-1.02822	1	1,359	99.92647	880	0	0.00000	60.66964
-1.00150	1	1,359	99.92647	880	0	0.00000	60.66964
-0.97477	1	1,359	99.92647	880	0	0.00000	60.66964
-0.94804	1	1,359	99.92647	880	0	0.00000	60.66964
-0.92132	1	1,359	99.92647	880	0	0.00000	60.66964
-0.89459	1	1,359	99.92647	880	0	0.00000	60.66964
-0.86786	1	1,359	99.92647	880	0	0.00000	60.66964
-0.84114	1	1,359	99.92647	880	0	0.00000	60.66964
-0.81441	1	1,359	99.92647	880	0	0.00000	60.66964
-0.78768	1	1,359	99.92647	880	0	0.00000	60.66964
-0.76096	1	1,359	99.92647	880	0	0.00000	60.66964
-0.73423	1	1,359	99.92647	880	0	0.00000	60.66964
-0.70750	1	1,359	99.92647	880	0	0.00000	60.66964
-0.68078	1	1,359	99.92647	880	0	0.00000	60.66964
-0.65405	1	1,359	99.92647	880	0	0.00000	60.66964
-0.62732	1	1,359	99.92647	880	0	0.00000	60.66964
-0.60060	1	1,359	99.92647	880	0	0.00000	60.66964
-0.57387	1	1,359	99.92647	880	0	0.00000	60.66964
-0.54714	1	1,359	99.92647	880	0	0.00000	60.66964
-0.52042	1	1,359	99.92647	880	0	0.00000	60.66964
-0.49369	1	1,359	99.92647	880	0	0.00000	60.66964
-0.46697	1	1,359	99.92647	880	0	0.00000	60.66964
-0.44024	2	1,358	99.85294	880	0	0.00000	60.62500
-0.41351	2	1,358	99.85294	880	0	0.00000	60.62500
-0.38679	4	1,356	99.70588	880	0	0.00000	60.53571
-0.36006	6	1,354	99.55882	880	0	0.00000	60.44643
-0.33333	9	1,351	99.33824	880	0	0.00000	60.31250
-0.30661	14	1,346	98.97059	880	0	0.00000	60.08929
-0.27988	22	1,338	98.38235	880	0	0.00000	59.73214
-0.25315	31	1,329	97.72059	880	0	0.00000	59.33036
-0.22643	39	1,321	97.13235	880	0	0.00000	58.97321
-0.19970	49	1,311	96.39706	880	0	0.00000	58.52679
-0.17297	69	1,291	94.92647	880	0	0.00000	57.63393
-0.14625	99	1,261	92.72059	880	0	0.00000	56.29464

Learn | Test | Holdout

Note the negative and greater than 1.0 values of the threshold in the table indicating that the predicted probability scores went outside of the [0, 1] interval.

Scoring and Translating MARS models

Press the **Score** button in the **MARS Results window** to open a generic **Score Data** dialog, or press the **Translate…** button to open a generic **Model Translation** dialog. Both dialogs are described in their respective sections.

Using these dialogs, you can score or translate any MARS model from the model sequence indexed by the number of Basis functions. Click the **Select…** button in the **Score Data** or **Model Translation** windows to see the complete list of MARS models available.

Minitab▸®

✓ By default, MARS will score the optimal model based on the minimum GCV (legacy mode) or minimum Test MSE (learn-test or cross-validation mode).

Shaping and Refining MARS Models

MARS models can be shaped and refined using the following techniques, each of which can influence the final model:

♦ Changing the number of Basis functions generated in the forward stage.

♦ Forcing variables into the model.

♦ Forbidding transformation of selected variables.

♦ Placing a penalty on the number of distinct variables (in addition to the number of Basis functions).

♦ Specifying a minimum distance (or minimum span) between knots.

♦ Allowing select interactions only.

♦ Modifying MARS search intensity.

♦ Manually selecting a model other than the optimal model from the selector.

Most of these controls have already been discussed earlier. It what follows, we provide additional insights on some of the less obvious topics.

Forcing Variables into the Model

At present, there no simple way to force variables into a MARS model. An indirect way, however, can be used to force a variable into a MARS model *linearly*: regress the target variable on the variable you wish to include and then use the residuals of the regression equation as the new target variable. That is, first run a linear regression in the form shown below:

$$y = constant + b\,X + e,$$

then derive and save the residuals, *e*. Finally, use *e* as the target variable in your MARS model and specify all other variables, including X, as candidate predictors. Note that X needs to be included as a legal second stage predictor to capture potential non-linearity.