

# Using Insightful Miner to Apply S-PLUS to Large Data Sets

---

**Edition 1**

*David M Smith*  
*S-PLUS Product Marketing Manager*

Insightful Miner Technical Note

# Table of Contents

<b>Abstract</b>	<b>3</b>
<b>Introduction</b>	<b>3</b>
<b>Computing on large data sets: the pipeline architecture</b>	<b>4</b>
Data partitioning	4
Data caching	5
Meta-data and data integrity	5
<b>The S-PLUS Script node</b>	<b>5</b>
<b>Running S-PLUS scripts in single-block mode</b>	<b>7</b>
Example: Fitting an S-PLUS GAM model within Insightful Miner	8
<b>Running S-PLUS scripts in multi-block mode</b>	<b>12</b>
Example: Scoring a GAM model on a large data set	13
<b>Summary</b>	<b>14</b>
Applications of the S-PLUS script node	14
<b>For more information</b>	<b>15</b>
<b>Appendix: Common Questions from S-PLUS users about Insightful Miner</b>	<b>16</b>
1. Isn't Insightful Miner just another interface to S-PLUS?	16
2. Do I need S-PLUS to use Insightful Miner?	16
3. If I have the S-PLUS Library for Insightful Miner installed, what extra functionality do I get?	16
4. Will the S-PLUS Script node in Insightful Miner allow me to run my existing S-PLUS scripts on bigger data sets than I can with S-PLUS?	16
5. What is the benefit of incorporating an S-PLUS script into an Insightful Miner workflow?	17
6. Will my S-PLUS scripts need to be modified for use with Insightful Miner?	17
7. Can any S-PLUS script be configured to run in streaming mode on large data sets?	17
8. Does the S-PLUS Script Node provide a specialized editor for S programming?	18
9. Do I have access to all of S-PLUS from the S-PLUS Script Node?	18
10. Can I use my own S-PLUS functions and libraries within the S-PLUS Script Node?	18
11. Can I access the S-PLUS command line from Insightful Miner?	18
12. When I create S-PLUS objects using the S-PLUS Script Node, where are they stored?	19
13. Can I access languages other than S-PLUS from Insightful Miner?	19
14. Can I customize the interface for S-PLUS Script Nodes?	19
15. What if I have other questions not addressed here?	20
<b>About Insightful Corporation</b>	<b>20</b>

## Abstract

Insightful Miner 3 includes many new features to help you to make use of existing functions and scripts created with S-PLUS and apply them to data analysis work flows within Insightful Miner. In this white paper, we will discuss some of the available features and provide an example of using a custom predictive model created in S-PLUS to create predictions on a very large database by using Insightful Miner.

The material in this white paper is aimed towards S-PLUS users interested in learning about how to apply existing S-PLUS code to larger data sets. We assume the reader knows how to read data files in S-PLUS and how to program S-PLUS functions. Basic familiarity with Insightful Miner, as found at [www.insightful.com/products/iminer](http://www.insightful.com/products/iminer), is also assumed.

A collection of common questions from S-PLUS users about Insightful Miner is presented with answers at the end of this Technical Note.

## Introduction

*For existing users of S-PLUS, Insightful Miner brings big-data capabilities to the realm of exploratory analysis and predictive modeling.*

Insightful Corporation's flagship statistical data analysis environment S-PLUS is the acknowledged leader in graphical display, exploratory analysis and advanced modeling of data. S-PLUS is an extraordinarily powerful and flexible system which relies on *in-memory* processing: when working with data objects, the entire data set is read into physical memory (RAM) at once. This allows S-PLUS routines to have rapid and dynamic access to data, which in turn enables the widest possible range of statistical algorithms to be included in the program.

The tradeoff for this flexibility is the amount of data that can be handled by S-PLUS. Since the entire data set must be accommodated in RAM, this provides a hard limit on the size of data that can be processed. Statistical algorithms often require multiple working copies of the data when processing the data, which further limits the size of data that can be handled for certain applications.

By contrast, Insightful Miner is designed around an out-of-memory model: instead of reading the entire data set into memory at once, the data is processed incrementally, with only a small amount in memory at any one time. This allows Insightful Miner to process data sets of essentially unlimited size, but limits the algorithms available to those that can be implemented using out-of-memory techniques.

Although Insightful Miner and S-PLUS are separate applications, they are designed to work in concert together. Insightful Miner and S-PLUS complement each other to provide a single application that is both highly scalable and extremely flexible. In this white paper, existing S-PLUS users will learn:

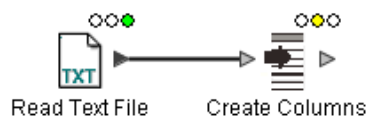
- How to use Insightful Miner to manipulate and select from large data sets, in preparation for in-memory modeling with S-PLUS
- How predictive models created using S-PLUS can be applied to score very large data using out-of-memory techniques provided by Insightful Miner

## Computing on large data sets: the pipeline architecture

S-PLUS performs all computations **in-memory**. In other words, when performing computations on a data set (such as calculating summary statistics or training a predictive model, to name two examples), the entire data set must be available – in memory – to the computational algorithm. Additional memory may also be required to store temporary data and the final result. This places an upper limit on the size of data that can be analyzed: the amount of physical RAM available to the computer. Adding virtual memory (or “swap space”) to the computer increases the size of data set that can be analyzed, but with a significant penalty to computational speed. Even when virtual memory is used, architectural limitations of the operating system impose further constraints: for example, 32-bit Windows systems are unable to address more than a couple of gigabytes of data.

In order to be able to perform computations on truly large data sets – gigabyte-class data sets – in-memory algorithms are not sufficient. Instead, **out-of-memory** algorithms that do not require the entire data set to be in memory are needed.

Insightful Miner implements a unique **pipeline architecture** that provides the framework for out-of-memory algorithms. For example, consider an Insightful Miner “Create Columns” node of one input and one output, with input data sourced from a “Read Text File” node:



In this hypothetical example, the purpose of the Create Columns node is to add a new column to the source data file, whose value is the square root of the first column. In doing so, the Create Columns node never sees the entire source data file at once. Instead, the pipeline architecture divides the source data row-wise into small chunks. The process within the Create Columns node is something like this:

1. Read 1000 lines of data from the input port into memory
2. Calculate the square root of the first column
3. Write out 1000 lines of data to the output port (including the new column)
4. Discard the current data block, reclaiming the used memory
5. If more data remains on the input port, return to Step 1

This process continues until the data on the input port is exhausted. Since only 1000 lines of data are in memory during any one iteration, this amount of memory required is fixed regardless of the size of the source data file.

The purpose of the pipeline architecture is to handle the various details of passing blocks of data between nodes: data partitioning, caching, meta-data and data integrity.

### Data partitioning

The pipeline architecture divides the incoming data stream into blocks, and delivers each block in turn to the input data port. The size of each data block is configurable for each node, and defaults to 10000 rows of data. The specified block size is an upper limit – the amount of data delivered to the node may be less than 10000 rows if less data is available (at the end of the file, for example), or if the amount of memory required by the block exceeds a limit set by the user: the “Max megabytes per block” setting of the Advanced Worksheet Properties. In the latter case, the number of rows will be reduced to adhere to the per-block limitation. The

default setting of the “Max megabytes per block” option is 10 megabytes, and comes into play only when the number of columns in the data stream is very large.

Nodes may have more than one input, and in this case the pipeline architecture also handles the synchronization of data from the various inputs.

For most applications, the actual block size used does not materially affect the computational performance of Insightful Miner. The computational time is dominated by data input and output (reads and writes on disk), and the overhead associated with managing individual blocks in the pipeline is negligible.

## Data caching

After processing each incoming data block, Insightful Miner by default stores the results of the output port in a *cache file* on disk. This cache file stores a compressed binary representation of the data sent to the output port. To save disk space, you can optionally prevent storage of the cache file on a per-node or a per-worksheet basis. However, there are several benefits to saving the cache file. First and foremost, the existence of the cache file allows you to view the results of intermediate nodes in the workflow (with the Table Viewer, for example), which is of great benefit when in the exploratory phase of developing a new data analysis application. Secondly, cache files improve speed of development when interactively modifying a workflow, since the results of nodes whose inputs have not changed do not need to be recomputed each time the workflow is run. This is particularly beneficial when using a complex SQL query on static databases, since the data extraction step does not need to be run each time you modify the network. Finally, the cache files represent an archival record of the results at each stage of the data analysis process, which may be useful for auditing purposes or for debugging multi-step computations.

## Meta-data and data integrity

The pipeline architecture is also responsible for calculating and maintaining the meta-data: column names, types, roles, and summary statistics. This information is passed from node to node to ensure it is available and up-to-date at every stage.

As part of this process, the pipeline architecture uses the meta-data to validate the data and ensure data integrity. For example, this includes ensuring that the incoming data stream matches the declared types of the data columns, and reporting any discrepancies that occur.

## The S-PLUS Script node

The S-PLUS script node in Insightful Miner allows the user to apply S-PLUS functions to data flowing through the Insightful Miner workflow. This allows you to apply the entire range of S-PLUS functions to any data in the stream. You can insert an S-PLUS script function anywhere in the worksheet, for example as shown:



Figure 1: An S-PLUS Script node used to convert separate time and data columns into a single column containing both time and date information.

When you create an S-PLUS script node, you can choose how many inputs and outputs it should have (including zero in both cases), but for the purposes of illustration we'll assume it has one input and one output. Data at the input port is converted to an S-PLUS data frame, and the goal of the S-PLUS script is to create an output data frame that is passed to the output port.

*The S-PLUS Script node allows you to apply S-PLUS code to data in an Insightful Miner workflow. Insightful Miner automatically creates the S-PLUS list object `IM` for use in the script, and gives you access to the incoming data as a data frame object.*

By editing the properties of the S-PLUS script node, you can enter a script that transforms the incoming data as required. The lines of code in the script are used as the body of a function used to process the data. This function is defined automatically within Insightful Miner as an S-PLUS function of one argument:

```
function(IM) { ... }
```

The script you paste into the Properties tab of the S-PLUS Script node is inserted as the body of this function. The object `IM` is a list object automatically created by Insightful Miner and passed to the function, and contains (amongst other useful information) the data linked to the input port converted to an S-PLUS data frame as a list element named `IM$in1`.

The return value of the function must be a data frame, and this is passed to the output port of the node and the data then flows through the rest of the workflow as normal data in the pipeline.

In the example below, ordinary S-PLUS commands are used to convert an incoming data stream that includes two columns: **tdate** (date of transaction, for example 01MAY1997) and **ttim** (time of transaction in seconds since midnight). For analysis purposes, later in the stream a single column containing the date *and* time is needed. (Insightful Miner supports columns of type "date" similar to `timeDate` objects in S-PLUS.) The `timeDate` function is used to create a new column **Positions** which is appended to the incoming data frame `IM$in1` and then returned as the output.

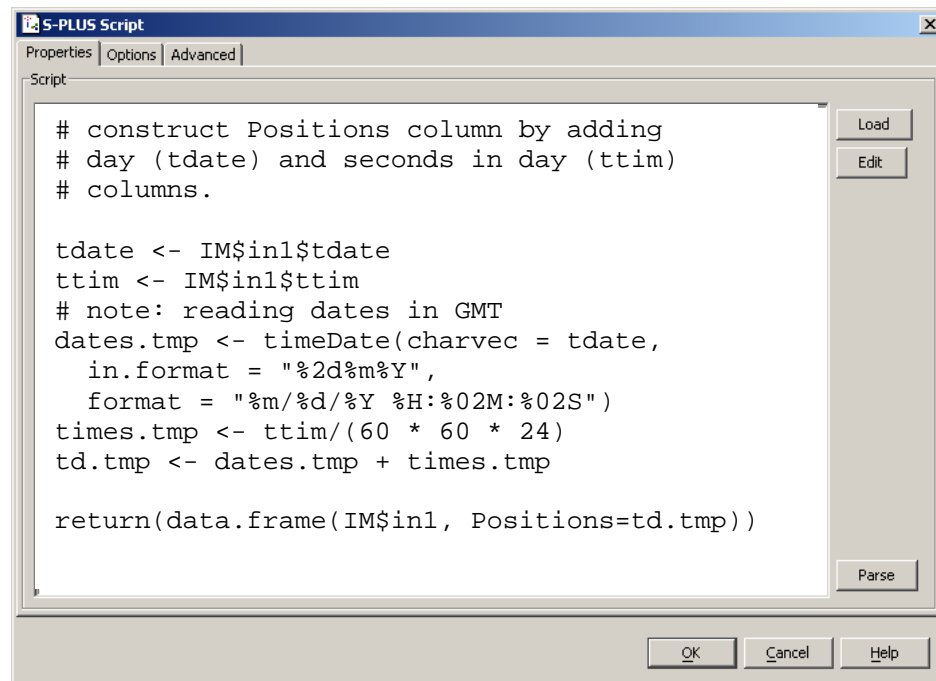


Figure 2: An S-PLUS script node converts incoming data to create a new column in the output

## Converting S-PLUS scripts for use with Insightful Miner

When converting S-PLUS scripts for use with Insightful Miner, it's important to remember that the S-PLUS script node can run in one of two different modes: **single-block mode** and **multi-block mode**. Once the decision of which mode you require is made, converting existing S-PLUS scripts is a simple process.

The basic difference between the two modes concerns the way that data is passed to the S-PLUS script. In single-block mode, the entire data set is collected into a single object and the S-PLUS script is run only once. In multi-block mode (also called streaming mode), the incoming data stream is divided into a number of small chunks, and the script is once for each chunk of data. The different modes are discussed in more detail below.

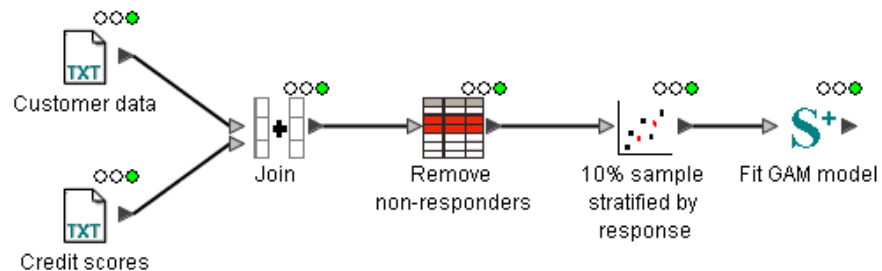
## Running S-PLUS scripts in single-block mode

*Because memory is reclaimed when an S-PLUS script node completes, converting a single S-PLUS script to a series of Insightful Miner S-PLUS Script nodes will often consume less RAM overall than running the script in S-PLUS.*

When an S-PLUS script is run in single-block mode, all of the data is collected at the input port before it is passed to the S-PLUS script. The script is then run exactly once. This most closely mimics the behavior of S-PLUS itself, so running S-PLUS scripts using single-block mode usually requires little modification to run in Insightful Miner. The downside is that because all of the data is collected in memory before the script is run, you usually won't be able to process larger data sets with the S-PLUS script node in Insightful Miner than you were able to in S-PLUS alone. Once the node has completed processing, all of the memory used by the script is immediately reclaimed, so breaking a long S-PLUS script into a series of nodes can often reduce the memory requirements compared to running a single script in S-PLUS.

Embedding an S-PLUS script in an Insightful Miner workflow also has the benefit that a large data file (much larger than S-PLUS can handle) can be easily preprocessed – using sampling, aggregation, or other methods to reduce the data set size – before the data is passed to the S-PLUS script. This data preparation step with large data sets is difficult to do with S-PLUS alone.

*Running an S-PLUS script in “single block mode” within Insightful Miner requires minimal changes to your script. But you have to ensure that enough memory is available on the PC to accommodate the entire amount of data that the script will use.*



*Figure 3: An Insightful Miner pipeline is used to merge, prepare and sample a large database before passing the data to and S-PLUS Script Node for analysis.*

To convert a script from S-PLUS to Insightful Miner and run it in single-block mode, first create an S-PLUS Script Node. Open the properties dialog for the node, and select the **Options** tab. Ensure that in the “Row Handling” section, the radio button for “Single Block” is selected (this is the default in Insightful Miner 3.0). With this selected, the S-PLUS script will only be run once, with all of the incoming data collected into a single data frame before the script is run. This means that enough RAM must be available on the PC to accommodate all of the data at once; if this is not the case, an out of memory error will occur. You can prevent this occurring by also selecting the “Max Rows” sub-option, in which case the rows of the

incoming data will be randomly sampled (without replacement) down to the specified number of rows. This is a useful option if sampling is an acceptable strategy, and you want to ensure the node will run regardless of the size of the input data.

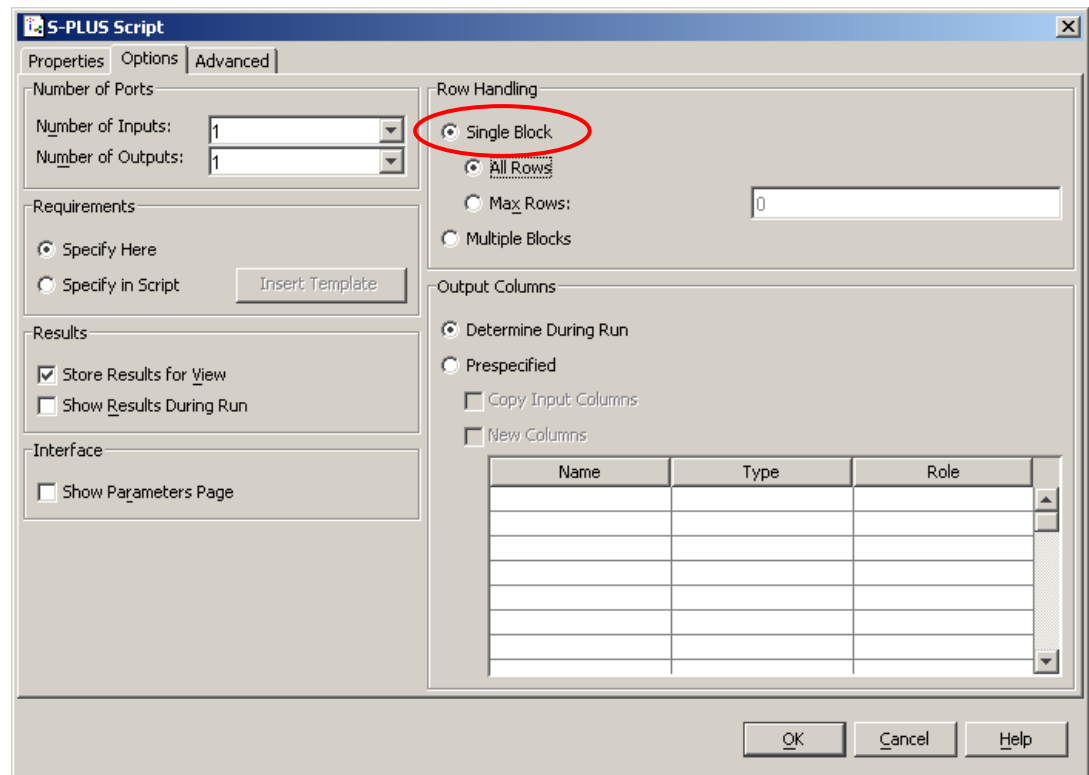


Figure 4: The default Row Handling in an S-PLUS script node is Single Block.

With the Options set as described, you can now enter your script into the Properties tab of the dialog. For simple scripts, you can type directly into the S-PLUS Script Node in Insightful Miner, but for scripts that are more complex it's usually easier to develop and debug the script in S-PLUS, taking advantage of the features included in the S-PLUS development environment such as the script window and easy access to help files. Once the script is developed, you can then cut-and-paste it into Insightful Miner, or load the script file directly into the node using the Load button on the Properties page.

### Example: Fitting an S-PLUS GAM model within Insightful Miner

*This example shows how an S-PLUS script node in single-block mode can be used to fit a predictive model and store the model object for scoring on a large data set.*

An example helps to illustrate this process. Let's suppose we'd like to use a Generalized Additive Model (GAM) to predict whether or not a given loan is likely to result in default, based on known information about the loan. (A more complete treatment of this example is given in Chapter 3 of the *Insightful Miner 3 Getting Started Guide*.) A likely scenario is that we've already modeled the data in S-PLUS, but would like to deploy this model in Insightful Miner. In S-PLUS, the `gam` function would be used to fit a generalized additive model, using a call like this:

```
mortdef.gam<-gam(Status ~ Delinquency*log(PercPastDue+1)+
  log(MonthsPastDue+1)+CurrentLTV+
  bs(CreditScore,knots=c(850,1050),degree=1)+
  I((PaymentDiff > -50)*(PaymentDiff + 50)),
  data=mortdef.data, family=binomial)
```

Here we assume the data is already available in a data frame called `mortdef.data`, and contains the necessary columns as referenced in the model formula. You can create the object `mortdef.data` yourself by importing the file `examples\MortgageDefaultExample\mortdef.train.txt` found in the Insightful Miner installation tree as a comma-separated file in S-PLUS. The Import Data dialog used for this is shown below.

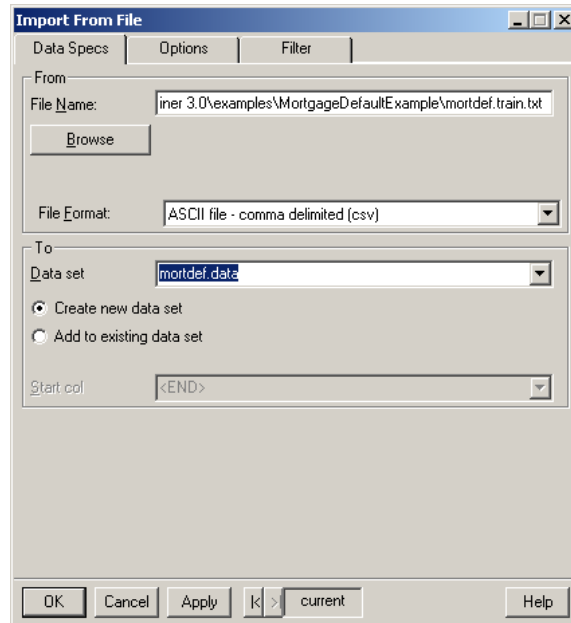


Figure 5: Reading the `mortdef.train.txt` example data set into S-PLUS

With the data imported into S-PLUS, we are ready to convert our script for use in Insightful Miner. First, note that this script *must* be run in single-block mode, since the entire data set must be passed to the `gam` function in a single call. We can simulate this in S-PLUS by creating a function of one argument, called `IM`, and making our script the body of the function:

```
IMsingleblock <- function(IM) {
  mortdef.gam <- gam(Status ~ Delinquency*log(PercPastDue+1)+
    log(MonthsPastDue+1)+CurrentLTV+
    bs(CreditScore,knots=c(850,1050),degree=1)+
    I((PaymentDiff > -50)*(PaymentDiff + 50)),
    data=IM$in1, family=binomial)
}
```

We did make one change to the script: the `data=` argument to `gam` was changed to `IM$in1`, since `IM$in1` is the data frame containing the input data. We can then test our function from the S-PLUS command line by calling it directly, creating the input `IM` object as it would be within Insightful Miner:

```
IMsingleblock(list(in1=mortdef.data))
```

So far, our function fits the model but doesn't do much else. Our purpose in creating a node in Insightful Miner was to make it possible to fit the model to new data, and save the results

for viewing in the worksheet. We can extend the function by adding code to print the model and to produce diagnostic charts:

```
IMsingleblock <- function(IM) {
  mortdef.gam <- gam(Status ~ Delinquency*log(PercPastDue+1)+
    log(MonthsPastDue+1)+CurrentLTV+
    bs(CreditScore,knots=c(850,1050),degree=1)+
    I((PaymentDiff > -50)*(PaymentDiff + 50)),
    data=IM$in1, family=binomial)
  print(summary(mortdef.gam))
  plot(mortdef.gam) # this call will fail
}
```

Unfortunately, this code does not work as written (we get an error from the call to `plot`: object "IM" not found). This is one area where converting the code from a simple script to a function call has an impact: according to the S object scoping rules, within a function body the object `IM` is no longer in scope at the call to `plot`. We use the standard S-PLUS trick of assigning the data object to frame 1 before the call, which makes the function run as intended.

```
IMsingleblock <- function(IM) {
  assign("data",IM$in1,frame=1)
  mortdef.gam <- gam(Status ~ Delinquency*log(PercPastDue+1)+
    log(MonthsPastDue+1)+CurrentLTV+
    bs(CreditScore,knots=c(850,1050),degree=1)+
    I((PaymentDiff > -50)*(PaymentDiff + 50)),
    data=data, family=binomial)
  print(summary(mortdef.gam))
  plot(mortdef.gam)
}
```

This function does not produce any output. Within an Insightful Miner S-PLUS Script Node, any data frame may be passed to the output port of the node. We can extend our function further by returning the original data frame, plus the predicted probability of default from the GAM model:

```
IMsingleblock <- function(IM) {
  assign("data",IM$in1,frame=1)
  mortdef.gam <- gam(Status ~ Delinquency*log(PercPastDue+1)+
    log(MonthsPastDue+1)+CurrentLTV+
    bs(CreditScore,knots=c(850,1050),degree=1)+
    I((PaymentDiff > -50)*(PaymentDiff + 50)),
    data=data, family=binomial)
  print(summary(mortdef.gam))
  plot(mortdef.gam)
  pred.p <- predict(mortdef.gam, type="response")
  return(cbind(data,probability=pred.p))
}
```

We can now paste the body of this function into the S-PLUS Script node and connect the input to a read text file node that reads the data file `mortdef.train.txt`.

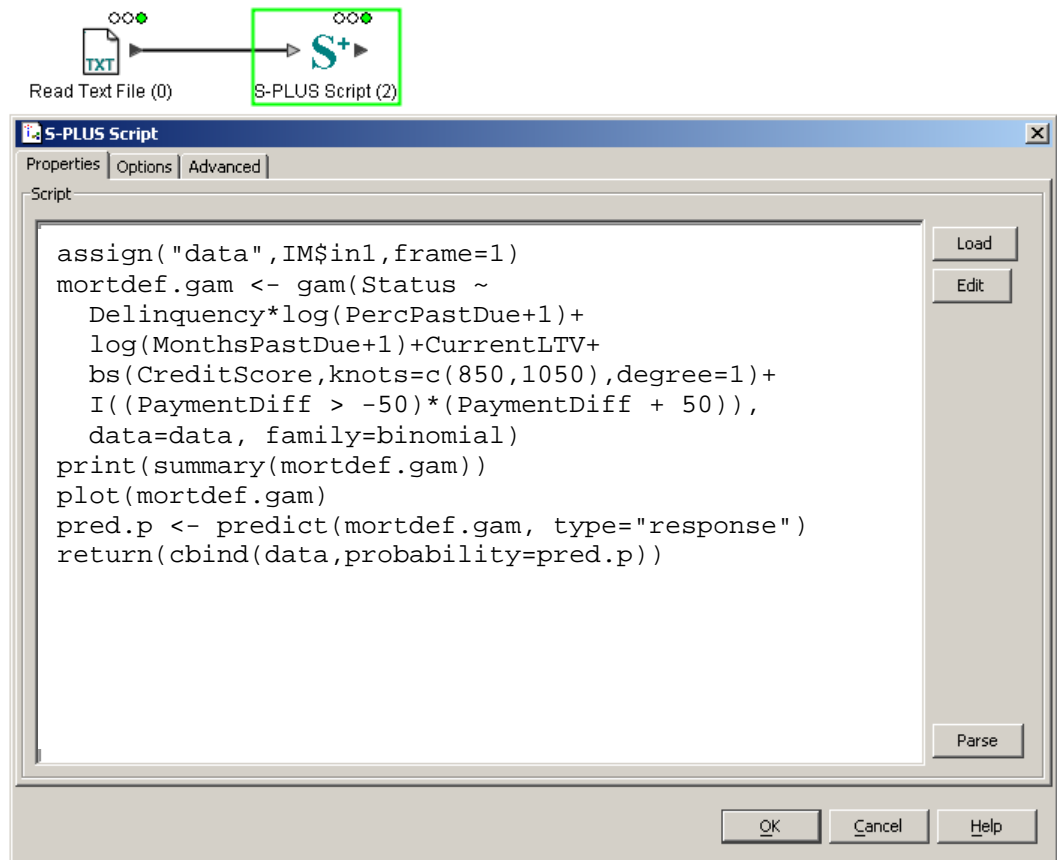


Figure 6: Our example S-PLUS code in the Script node.

In this example, we are simply appending the predicted probability as a new column to the input data to create the output. Other possibilities include outputting the predicted class (default or non-default), identifying loans with probability of default above a certain threshold, and so on. It is also possible to annotate the output in such a way that it becomes a suitable input for an Insightful Miner model assessment node. This is useful for comparing several S-PLUS models and native Insightful Miner models in a single chart.

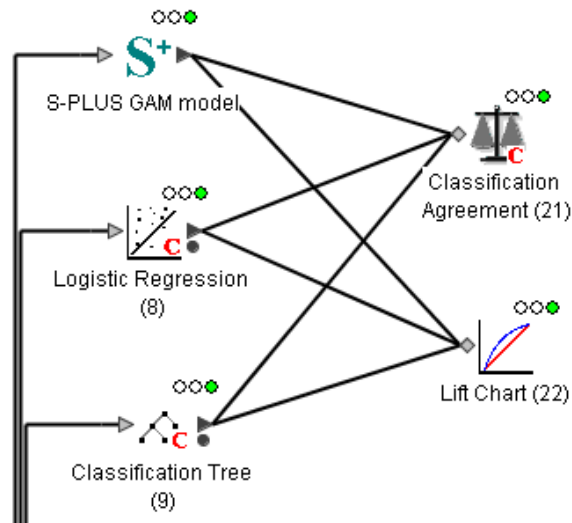


Figure 7: The performance of an S-PLUS GAM model is compared to a Logistic Regression model and a Classification Tree model using Insightful Miner's model assessment nodes

If you plan to use the model for scoring (calculating predictions on a new data set) elsewhere in the worksheet, you'll need to store the model object for use in another S-PLUS script node. There are several ways to save model objects (and indeed, any S-PLUS object) for use in another script node. You can use the `assign` function to save the object in a specified chapter, or you can use the `data.dump` function to save them in a file for later retrieval. In this example we save both the training data object and the model object, since both are needed for use with `predict.gam`:

```
# Save the model object and data to a file for use in a
# predict script node
assign("mortdef.gam", mortdef.gam, where=1)
on.exit(remove("mortdef.gam", where=1))
data.dump(c("mortdef.data", "mortdef.gam"), "mortdefGam.sdd")
```

We'll make use of these saved objects when creating predictions from the model, as shown below.

## Running S-PLUS scripts in multi-block mode

For some data processing applications, it is feasible to divide the incoming data into smaller pieces, and process each piece separately. In situations where this is possible, running an S-PLUS script node in multi-block mode allows you to apply your script to data sets of essentially unlimited size.

When running in multi-block mode, the incoming data is *not* presented to the script as a single data frame. Instead, Insightful Miner divides the incoming data into a series of smaller **blocks**. By default, each block is set at 10,000 rows – this setting can be changed in the Advanced tab of the Script node. The first block of data is passed to the script as the data frame `IM$in1`, and the script is run (again, as the body of an anonymous function). The output of the function is written to a cache file, and the first block of data is discarded (and the memory reclaimed). Then, the next 10,000 rows of data is passed to the script (again, as the data frame `IM$in1`) and the script is run a second time, with the output data appended to the cache file. This process continues until no data remains to be processed.

Because the script is run multiple times (as many times as there are blocks of data), multi-block mode is best suited for applications where each block of data can be processed independently. Simple row-based transformations or column creations are trivial to implement in streaming mode. For example a thresholded log transform like

```
IM$in1$logexposure<-ifelse(IM$in1$expo>5,log(IM$in1$expo),0)
```

can easily be applied to very large data sets in streaming mode. Combining multiple columns into a new transformed column is also possible as described in the time/date manipulation example given earlier.

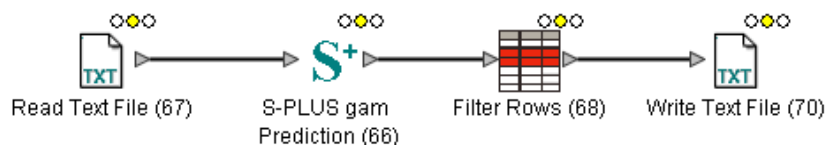
More complex transformations are also possible in streaming mode using advanced functionality of the S-PLUS script node. By making certain settings in the `IM` object it is possible to have more direct control over how data is passed from the pipeline engine to the S-PLUS function. For example, it is possible for each data block to overlap, allowing for time-series style operations (for example, differences or updating models). Multiple passes through the data are also possible. The details of such use are beyond the scope of this article, and the reader is referred to the Insightful Miner User's Guide for further information about this functionality.

### Example: Scoring a GAM model on a large data set

*This example shows a saved model object can be used with an S-PLUS script in streaming mode to calculate scores on a very large data set.*

One especially useful application of the S-PLUS script node in multi-block mode is to calculate predictions (scores) from a previously-trained predictive model on a very large data set. Since the prediction from each row of data can be calculated independently from each other row, processing the data block by block is trivial in this case.

The pipeline for a scoring application is usually very simple. All that is required is to read in the new data, calculate the predictions using S-PLUS, and process the outputs.



*Figure 8: A pipeline for scoring an S-PLUS GAM model on a large data set. The Filter node is used to select out only those loans with a low predicted probability of default.*

The code to generate the predictions is equally simple. Once the saved model object has been retrieved, it is simply a matter of using the S-PLUS `predict` function on the new data set. (We assume here that the scoring data set has the same column names as the training data. If not, some data manipulation before the scoring step will be required.) The code below continues the previous loan prediction example to apply the trained GAM model to a database of 10 million new loans:

```

# Read the training data and model object "mortdef.gam"
data.restore("mortdefGam.sdd")
on.exit(remove(c("mortdef.data", "mortdef.gam"), where=1))

# Return probabilities and classifications on new data.
pred.prob <- predict(mortdef.gam,type="response",newdata=IM$in1)
pred.class <- as.factor(
  ifelse(pred.prob > 0.5, "Default", "NoDefault"))
out <- IM$in1
out$PREDICT.prob <- 1-pred.prob
out$PREDICT.class <- pred.class
return(list(out1=out))

```

Since each block of data is processed independently, no special coding is required to calculate such predictions on very large data sets. With each block, all the memory used by the S-PLUS computation is reclaimed, so there is no memory “leakage” as the computation progresses. This technique can therefore be efficiently applied to data sets of essentially unlimited size, and the time of computation is directly proportional to the total number of rows of data that are scored.

## Summary

Insightful Miner complements S-PLUS by allowing the S-PLUS programmer to apply existing S-PLUS scripts and functions to very large data sets. Insightful Miner handles all the details of manipulating and selecting data, while the S-PLUS script node provides all of the power of the S language and its associated modules and libraries. The S-PLUS script node allows for customization of Insightful Miner, allowing it to readily adapt as requirements change.

## Applications of the S-PLUS script node

Because the S-PLUS script node gives you access to the full range of S-PLUS functionality, including functions provided by additional S-PLUS modules such as S+FinMetrics or S+ArrayAnalyzer, the potential of the S-PLUS script node is essentially unlimited. Here are just a few applications for how it can be used, in both single-block mode and in multi-block mode.

*In single-block mode, applied to in-memory, sampled, or aggregated data:*

- Add a custom chart to a workflow
- Fit a predictive model
- Create a custom report

*In multi-block mode, on large streaming data sets:*

- Perform row-wise or sequential update transformations of columns
- Create new columns based on row-wise combinations of other columns
- Implement custom sampling or filtering techniques
- Generate random data for simulations
- Generate predictions from a previously-trained predictive model
- Create custom data output formats

## For more information

Examples of S-PLUS and Insightful Miner in action for business applications can be found in our series of on-line webcasts. You can view past webcasts at Insightful's website: [www.insightful.com/news\\_events/events.asp](http://www.insightful.com/news_events/events.asp). For more information about S-PLUS and Insightful Miner, please contact an Insightful office near you, or email [info@insightful.com](mailto:info@insightful.com)

### Global Headquarters

1700 Westlake Avenue North Suite 500  
Seattle, WA 98109  
Tel: 206.283.8802 • 800.569.0123  
Fax: 206.283.6310  
email: [info@insightful.com](mailto:info@insightful.com)

### Insightful UK

5th Floor  
Network House  
Basing View  
Basingstoke, Hampshire  
RG21 4HG  
Tel: +44 (0) 1256 339800  
Fax: +44 (0) 1256 339839  
[info.uk@insightful.com](mailto:info.uk@insightful.com)

### Insightful Switzerland

Christoph Merian Ring 11  
Ch - 4153 Reinach  
Switzerland  
Tel: +41 61 717 9340  
Fax: +41 61 717 9341

### Insightful France

7, rue Auber  
31000 Toulouse  
France  
Tel: +33 0 5 62 27 70 60  
Fax: +33 0 5 62 27 70 61

## Appendix: Common Questions from S-PLUS users about Insightful Miner

### 1. Isn't Insightful Miner just another interface to S-PLUS?

No. Insightful Miner is an entirely separate application to S-PLUS, and includes its own algorithms – developed by Insightful specifically for Insightful Miner – for reading, manipulating and modeling data. Unlike S-PLUS, which works with complete data objects loaded into memory, Insightful Miner uses its unique pipeline architecture to “stream” data through the algorithms. Because the data is never all loaded into memory at once, Insightful Miner can use these algorithms to process much larger data sets than S-PLUS is able to alone.

### 2. Do I need S-PLUS to use Insightful Miner?

No. Insightful Miner runs as a stand-alone application, and does not require S-PLUS to be available. But if you have both S-PLUS and Insightful Miner installed, you can install the “S-PLUS Library for Insightful Miner” (included on the S-PLUS CD-ROM) to add additional functionality to Insightful Miner.

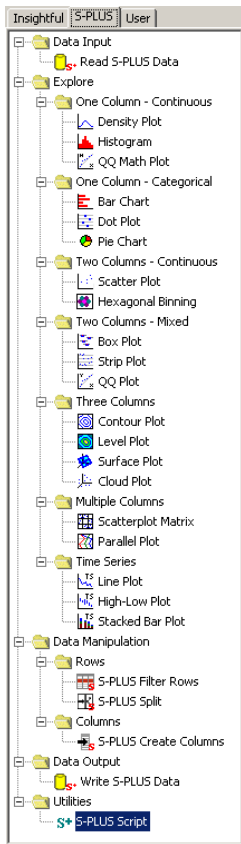
### 3. If I have the S-PLUS Library for Insightful Miner installed, what extra functionality do I get?

The S-PLUS Library for Insightful Miner (installed separately) adds 26 new nodes to Insightful Miner which make use of S-PLUS functionality. These include: the S-PLUS Script Node (which allows you to run S-PLUS scripts within Insightful Miner); nodes to Create Columns, Filter Rows, and Split a data set using S-PLUS expressions; nodes to read and write S-PLUS data sets (in `data.dump` format); and nodes for creating 1-D, 2-D, 3-D and Time Series charts (modeled on the Graph menu dialogs from the S-PLUS for Unix Java GUI).

### 4. Will the S-PLUS Script node in Insightful Miner allow me to run my existing S-PLUS scripts on bigger data sets than I can with S-PLUS?

Often, but not always. The S-PLUS Script node runs scripts using the same S-PLUS language engine included with S-PLUS itself, so all computations are done in-memory just as they are with S-PLUS. This means that enough RAM must be available to accommodate all of the data that will be used with the script.

This is in contrast to the “native” Insightful Miner nodes, which use Insightful Miner’s pipeline architecture to “stream” the data through the node, instead of loading it all into memory at once. This means data sets much larger than the available RAM can be processed with the node. The S-PLUS Script node can also be configured to run in streaming mode, which works well when it makes sense to split the data up into smaller pieces and run the script on each piece in turn. One common application of this technique is to create predictions from an already-trained model for each row of a very large data set. Since the prediction for each row can be calculated independently, very large data sets – tens of gigabytes of data is not uncommon – can be processed with an S-PLUS script. Using an S-PLUS Script node in this way is not unlike using the capital-F For loop in S-PLUS, except that Insightful Miner

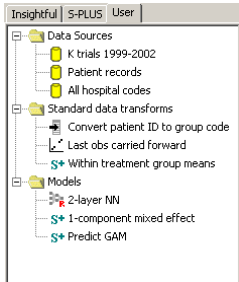


The S-PLUS Library for Insightful Miner adds 26 new nodes for S-PLUS based scripts, data access, manipulation and graphics

handles all the details of splitting the source data file and collecting the individual results.

## 5. What is the benefit of incorporating an S-PLUS script into an Insightful Miner workflow?

There are several benefits:



*Insightful Miner allows you to create libraries of custom nodes – including S-PLUS Script nodes – and share them with other users.*

- Even if your S-PLUS script must process all of the data at once, by incorporating it into an Insightful Miner you can more easily work with large data sets to prepare the data for use by the S-PLUS script. Insightful Miner is excellent at sampling and manipulating large data sets, and so provides an ideal environment for delivering the data needed to S-PLUS, and for collecting the results. By using Insightful Miner for the data access and manipulation steps, and S-PLUS for modeling and graphics on sampled or aggregated data, you can work more efficiently with large data sets than you can with S-PLUS alone.
- In some situations, you can configure your S-PLUS script to process small sections of a large data set sequentially. Insightful Miner handles all the details of passing small sections of the data to the script in turn. Since not all of the data is in memory at once, you can process extremely large data sets using this technique.
- Creating an S-PLUS Script Node in Insightful Miner is just like adding a custom node, which you can re-use in other Insightful Miner worksheets. You can also create your own library of S-PLUS Script Nodes, and share that library with other users. This is a great way of making your S-PLUS code available to other Insightful Miner users (who aren't S-PLUS experts) in an easy-to-use environment – it's a lot simpler to hook up a custom node in a workflow than it is to start S-PLUS and run a script.

## 6. Will my S-PLUS scripts need to be modified for use with Insightful Miner?

Not always, but some minor modifications are sometimes needed. Mainly, this is to ensure the script works as the body of a function (rather than as a command-line script), as this is how the script is called within Insightful Miner. The S scoping rules are a little different in this case, and you may need to ensure that objects in the body of the script are in scope and can be accessed from within subfunctions. In addition, you will need to configure the script to work as part of an Insightful Miner node – this means that the input is one or more data frames, and the output is one or more data frames. No data type other than `data.frame` is supported as an input or an output of an Insightful Miner node. However, the S-PLUS script is of course free to use objects of any type (perhaps sourced from files or other S-PLUS libraries) in its calculations.

## 7. Can any S-PLUS script be configured to run in streaming mode on large data sets?

Not all scripts make sense when run in the streaming “multi-block” mode. Some S-PLUS functions need all of the data available in a single object to perform their calculations. An example is the modeling functions like `gam` or `nlme` which require a single function call to create the model. There is no simple way of (say) dividing the data into two, and then combining the results of two calls to `gam` to create a single model. Another example is graphics: you would usually only want to make a single call to `plot` or `pairs` to explore the data, rather than having separate plots for each “chunk” of data processed by the script.

If it *does* make sense divide the data into smaller sections, and then run the script separately on each section, then it is usually easy to convert the script for use in multi-block mode. Data transformations which take advantage of row-by-row vector notation lend themselves naturally to this: for example, `df$log.age <- log(df$age)` can easily be run in streaming mode. A common application is the use of the `predict` function to calculate predictions from a previously fitted model on a large data set. Examples like these can easily be run in multi-block mode and applied to very large data sets.

## **8. Does the S-PLUS Script Node provide a specialized editor for S programming?**

A basic text editor is supplied within the S-PLUS Script Node to support simple modifications. For editing large amounts of code, we recommend using S-PLUS itself (this also makes debugging easier) and cut-and-pasting the code into Insightful Miner. The S-PLUS Script Node in Insightful Miner also provides convenient “Load” and “Edit” buttons that allow you to import a script you’ve edited elsewhere, or to edit the script with the editor of your choice. The editor invoked when you click the “Edit” can be configured with the “Tools | Options” dialog.

## **9. Do I have access to all of S-PLUS from the S-PLUS Script Node?**

Yes, the complete S-PLUS language engine is available from within the S-PLUS Script Node. This includes all of the functions and objects available from the S-PLUS command line.

S-PLUS for Windows GUI functionality is not available, since Insightful Miner does not include the S-PLUS for Windows Graphical User Interface. This includes all functions that begin with `gui` (such as `guiCreate` and `guiModify`) and the `graphsheet` device. The `java.graph` device is available for graphics, and displays within the Insightful Miner interface to allow you to present S-PLUS graphics as an output of your S-PLUS Script node.

## **10. Can I use my own S-PLUS functions and libraries within the S-PLUS Script Node?**

Yes. The `library` function works as usual to add custom S-PLUS libraries to the S-PLUS search path, and you can use the `attach` function directly in the usual fashion. The `module` function also works to access S-PLUS modules such as S+FinMetrics, S+ArrayAnalyzer and S+SpatialStats.

## **11. Can I access the S-PLUS command line from Insightful Miner?**

Yes. You can enable the S-PLUS command line with the “View Command Line” option in the View menu. (This option is only available if the S-PLUS Library for Insightful Miner is installed.) It allows you to type an S-PLUS statement, which is evaluated and the results displayed in Insightful Miner’s output pane. It’s not really designed for extensive programming – use S-PLUS instead for “real” work – but it’s handy for quick evaluations and to check the status of the S-PLUS engine when debugging S-PLUS script nodes.



This is a very simple interface, but it's easy to program while still being flexible. If you need a more sophisticated interface, such as a dedicated dialog with its own text fields, drop-down selections, radio buttons and such, Insightful Professional Services can create one for you. This would be delivered to you as a custom library for Insightful Miner that includes a new node to your specifications. Please contact Insightful Sales if you would like more information about this option. It is not presently possible to create custom node interfaces in Insightful Miner without the assistance of Insightful Professional Services.

## 15. What if I have other questions not addressed here?

If you already own Insightful Miner, live technical support is available as part of your Maintenance and Support Agreement. You can contact the technical support team by email, fax or telephone. Information on how to get support for Insightful products, and on-line information such as FAQs, is available from the Insightful website at [www.insightful.com/support](http://www.insightful.com/support).

If you haven't yet purchased Insightful Miner, your Sales representative will be happy to help you with any questions you may have.

## About Insightful Corporation

Insightful Corporation (NASDAQ: IFUL) provides enterprises with scalable data analysis solutions that drive better decisions faster by revealing patterns, trends and relationships. The company is a leading supplier of software and services for statistical data analysis, data mining and knowledge access enabling clients to gain intelligence from numeric and text data. The company's products include S-PLUS®, Insightful Miner, StatServer®, S-PLUS Analytic Server®, S+FinMetrics™, S+Wavelets®, and S+NuOPT™. Headquartered in Seattle, Insightful has offices in New York City, North Carolina, France, Switzerland, and the United Kingdom with distributors around the world. For more information, visit <http://www.insightful.com/>, email [info@insightful.com](mailto:info@insightful.com) or call 1-800-569-0123.

IMTN1103 Copyright © 2003 Insightful Corporation. All rights reserved. Printed in the United States of America. This data sheet is for informational purposes only. INSIGHTFUL MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. S-PLUS, StatServer, S-PLUS Analytic Server, InFact and S+Wavelets are registered trademarks and S+ArrayAnalyzer, S+FinMetrics, S+NuOPT and S+SeqTrial are trademarks of Insightful Corporation. All product names mentioned herein may be trademarks or registered trademarks of their respective companies