IBM SPSS Modeler 15 R Modeling Nodes

Note
Before using this information and the product it supports, read the information in "Notices" on page 9.

Product Information

This edition applies to version 15, release 0, modification 0 of IBM SPSS Modeler and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

IBM SPSS Modeler R Nodes

IBM SPSS Modeler R Nodes - Overview

To complement IBM® SPSS® Modeler and its data mining abilities, the IBM SPSS Modeler R nodes enable expert R users to input their own R script to carry out model building and model scoring.

If you have a compatible copy of R installed, you can connect to it from IBM SPSS Modeler and carry out model building and model scoring using custom R algorithms that can be deployed in IBM SPSS Modeler. You must also have a copy of IBM SPSS Modeler - Essentials for R installed. IBM SPSS Modeler - Essentials for R provides you with tools you need to start developing custom R applications for use with IBM SPSS Modeler. See the document *IBM SPSS Modeler - Essentials for R: Installation Instructions* for information about installation instructions and version compatibility.

Note: We recommend that you instantiate your data in a Type node before using the IBM SPSS Modeler R nodes.

The IBM SPSS Modeler R plug-in contains the following nodes:

The R Building node enables you to enter custom R script to perform model building and model scoring deployed in IBM SPSS Modeler. Executing an R Building node generates an R model nugget. See the topic "R Building Node" for more information.

The R model nugget resembles a standard IBM SPSS Modeler model nugget (also known as a model applier node), and defines a container for a generated model to be used when the model is added to the IBM SPSS Modeler canvas from the **Models** tab of the manager pane. The R model nugget can be edited to view the various forms of model output. See the topic "R Model Nugget" on page 4 for more information.

Note: These nodes are not currently supported in the scoring service of IBM SPSS Collaboration and Deployment Services.

R Building Node

With the R Building node, you can carry out model building and model scoring using R scripting within IBM SPSS Modeler. This makes it possible to carry out model building and scoring using algorithms that are written in R, and enables the user to develop modeling methods that are tailored to a particular problem. Execution of an R Building node generates an R model nugget.

To use this node, you must have installed IBM SPSS Modeler 15 Fix Pack 2, IBM SPSS Modeler updates and extensions for IBM SPSS Analytic Server, and IBM SPSS Modeler - Essentials for R, and have a compatible version of R installed on your computer. See the Release Notes for installation instructions and compatibility information.

R Building Node - Syntax Tab

R model building syntax. You can enter, or paste, custom R scripting syntax for model building into this field.

R model scoring syntax. You can enter, or paste, custom R scripting syntax for model scoring into this field. When the node is executed, the R script in this field is copied over to the R model nugget that is generated. The script itself is only executed when the R model nugget is executed.

Note: For more information about the syntax that is supported for use in these fields, see the topic "Syntax" on page 3.

Run. To create an R model nugget, click **Run**. The R model nugget is added to the Models palette, and optionally to the stream canvas.

R Building Node - Model Options Tab

Model Name. When **Auto** is selected, the model name is automatically set to the string "R Syntax". Select **Custom** to specify a custom model name in the adjoining text field. If you have executed the node once, and you do not specify a different model name before you execute the node again, the model from the previous execution will be overwritten.

Read Data Options. With these options, you can specify how missing values and variables with date or datetime formats are handled.

- **Convert missing values to the R 'not available' value (NA).** When selected, any missing values are converted to the R `NA` value. The value `NA` is used by R to identify missing values. Some R functions that you use might have an argument that can be used to control how the function behaves when the data contain `NA`. For example, the function might allow you to choose to automatically exclude records that contain `NA`. If this option is not selected, any missing values are passed to R unchanged, and might cause errors when your R script is executed.
- **Convert date/time fields to R classes with special control for time zones.** When selected, variables with date or datetime formats are converted to R date/time objects. You must select one of the following options:
  - **R POSIXct.** Variables with date or datetime formats are converted to R `POSIXct` objects.
  - **R POSIXlt (list).** Variables with date or datetime formats are converted to R `POSIXlt` objects.

  Note: The POSIX formats are advanced options. Use these options only if your R script specifies that datetime fields are treated in ways that require these formats. The POSIX formats do not apply to variables with time formats.
- **Output Options.** With these options, you can specify how output from R is displayed.
  - **Display R graphs as HTML.** When selected, R graphs are displayed in HTML format on the **Graph Output** tab of the R model nugget. The **Graph Output** tab displays only those plots that are generated from executing the R script in the **R model building syntax** field of the **Syntax** tab. See the topic "R Model Nugget - Graph Output Tab" on page 4 for more information.
  - **Display R text output.** When selected, any text output that is produced by executing the R script in the **R model building syntax** field is displayed on the **Text Output** tab of the R model nugget. See the topic "R Model Nugget - Text Output Tab" on page 5 for more information. If you want the text output to be saved to a file, include a call to the R `sink` function in your script. Any output that is produced after the call to the `sink` function is not displayed on the **Text Output** tab. Any R error messages or warnings that result from executing your R model building script are always displayed on the **Console Output** tab of the R Building node.

R Building Node - Console Output Tab

The **Console Output** tab contains any output that is received from the R console when the R script in the **R model building syntax** field on the **Syntax** tab is executed. This output might include R error messages or warnings that are produced when the R script is executed, and text output from the R console. The output can be used, primarily, to debug the R script. The **Console Output** tab also contains the R script from the **R model building syntax** field. Every time the model building script is executed, the content of the **Console Output** tab is overwritten with the output received from the R console. The console output cannot be edited.

If **Display R text output** is selected on the **Model Options** tab, the text output from the R console can instead be viewed on the **Text Output** tab of the R model nugget. Any R error messages or warnings that

are produced when the R script is executed will still be displayed on the **Console Output** tab. See the topic "R Model Nugget - Text Output Tab" on page 5 for more information.

Syntax

Within the **R model building syntax** and **R model scoring syntax** fields on the **Syntax** tab of the R Building node dialog box, only statements and functions that are recognized by R are allowed.

In your R model scoring script, to use a function from a library that must be loaded by including a call to the R `library` function, you must load the library in your R model scoring script, even if the library has already been loaded in your R model building script.

During the model scoring process, to display the value of an R object that is defined in your R script, you must include a call to the R `print` function in the **R model scoring syntax** field. For example, to display the value of an R object that is called `data`, include the following line in your R script:

```
print(data)
```

The value of the R object `data` is displayed on the **Console Output** tab of the R model nugget.

You cannot include a call to the R `setwd` function in your R script because this function is used by IBM SPSS Modeler to control the file path of the R scripts output file.

Stream parameters that are defined for use in CLEM expressions and scripting are not recognized if used in R scripts.

In the **R model building syntax** field, you must assign the model object that is generated when your model building script is executed to the R object `modelerModel`. IBM SPSS Modeler retains this model object in the R model nugget to pass back to R when scoring data. The model object `modelerModel` can be referenced in the model scoring script. For more information, see the section "Example" on page 5. If you assign more than one model object to `modelerModel` in your model building script, only the last model object is retained for scoring data.

Additionally, there are some R objects that are automatically populated when an R Building node and an R model nugget are used in a stream:

- **modelerData.** This is an R data frame that is automatically populated with the data that flows into the R Building node and R model nugget.
- **modelerDataModel.** This is an R data frame that is automatically populated with the data model that flows into the R Building node and R model nugget. The data model describes the type and structure of the data (that is, the metadata) that flows into the nodes.

Any other R objects that are defined in the R script in the **R model building syntax** field will not be recognized if they are used in the R model scoring script. If you want to make reference to these R objects in your model scoring script, you must redefine them in the R script in the **R model scoring syntax** field.

The R script that is entered into the **R model building syntax** and **R model scoring syntax** fields is used to manipulate the R objects `modelerData` and `modelerDataModel`. For example, you might want to add to the data model, `modelerDataModel`, using your model scoring R script. The data model `modelerDataModel` must be modified to match any changes that were made to the data `modelerData`. When the R Building node is successfully executed, a model is generated and an R model nugget is created. The R object `modelerData` is automatically used as the output data of the R model nugget. The R object `modelerDataModel` is automatically used as the output data model of the R model nugget.

R Model Nugget

The R model nugget is generated and placed on the Models palette after executing the R Building node, which contains the R script that defines the model building and model scoring. By default, the R model nugget contains the R script that is used for model scoring, options for reading the data, and any output from the R console. Optionally, the R model nugget can also contain various other forms of model output, such as graphs and text output. After the R model nugget is generated and added to the stream canvas, an output node can be connected to it. The output node is then used in the usual way within IBM SPSS Modeler streams for obtaining information about the data and models, and for exporting data in various formats.

To use this node, you must have installed IBM SPSS Modeler 15 Fix Pack 2, IBM SPSS Modeler updates and extensions for IBM SPSS Analytic Server, and IBM SPSS Modeler - Essentials for R, and have a compatible version of R installed on your computer. See the Release Notes for installation instructions and compatibility information.

R Model Nugget - Syntax Tab

The **Syntax** tab is always present in the R model nugget.

R model scoring syntax. The R script that is used for model scoring is displayed in this field. By default this field is enabled but not editable. To edit the R model scoring script, click **Edit**.

Edit. Click **Edit** to make the **R model scoring syntax** field editable. You can then edit your R model scoring script by typing in the **R model scoring syntax** field. For example, you might want to edit your R model scoring script if you identify an error in your model scoring script after you have executed the R model nugget. Any changes that you make to the R model scoring script in the R model nugget will be lost if you regenerate the model by executing the R Building node.

R Model Nugget - Model Options Tab

The **Model Options** tab is always present in the R model nugget.

Read Data Options. With these options, you can specify how missing values and variables with date or datetime formats are handled.

- **Convert missing values to the R 'not available' value (NA).** When selected, any missing values are converted to the R `NA` value.
- **Convert date/time fields to R classes with special control for time zones.** When selected, variables with date or datetime formats are converted to R date/time objects. You must select one of the following options:
  - **R POSIXct.** Variables with date or datetime formats are converted to R `POSIXct` objects.
  - **R POSIXlt (list).** Variables with date or datetime formats are converted to R `POSIXlt` objects.

  Note: The POSIX formats are advanced options. Use these options only if your R script specifies that datetime fields are treated in ways that require these formats.

R Model Nugget - Graph Output Tab

The **Graph Output** tab is present in the R model nugget if requested by selecting the **Display R graphs as HTML** check box on the **Model Options** tab of the R Building node dialog box. Graphs that result from executing the model building R script can be displayed on this tab. For example, if your R script contains a call to the R `plot` function, the resulting graph is displayed on this tab. If you execute the model building script again, without having first specified a different name for the model, the content of the **Graph Output** tab from the previous execution will be overwritten.

R Model Nugget - Text Output Tab

The **Text Output** tab is present in the R model nugget if requested by selecting the **Display R text output** check box on the **Model Options** tab of the R Building node dialog box. This tab can display only text output. Any text output that is produced by executing your R model building script is displayed on this tab. If you execute the model building script again, without having first specified a different name for the model, the content of the **Text Output** tab from the previous execution will be overwritten. The text output cannot be edited.

If you include a call to the R `sink` function in your script, any output that is produced after this function is saved to the specified file and is not displayed on the **Text Output** tab.

Note: R error messages or warnings that result from executing your R model building script are always displayed on the **Console Output** tab of the R Building node.

R Model Nugget - Console Output Tab

The **Console Output** tab is always present in the R model nugget. It contains any output that is received from the R console when the R script in the **R model scoring syntax** field on the **Syntax** tab of the R model nugget is executed. This output includes any R error messages or warnings that are produced when the R script is executed, and any text output from the R console. The output can be used, primarily, to debug the R script. Every time the model scoring script is executed, the content of the **Console Output** tab is overwritten with the output received from the R console. The console output cannot be edited.

Example

In this example, a linear model is fitted to the example data set DRUG1n, using the variable Age as the model input field and the variable Na as the model target field. The linear model is then used to score the same data set.

1. Add a Variable File node, from the Sources palette, to the stream canvas.
2. Double-click the Variable File node to open the node dialog box.
3. Click the ellipsis button (...) to the right of the **File** field to select the DRUG1n data set. The file that contains the DRUG1n data set can be found in the **Demos** folder.
4. Click **OK** to close the Variable File node.
5. Add an R Building node, from the Modeling palette, to the stream canvas and connect it to the Variable File node.
6. Double-click the R Building node to open the node dialog box.
7. In the **R model building syntax** field on the **Syntax** tab, enter the following R script:

   ```
   modelerModel<-lm(Na~Age,data=modelerData)
   plot(x=modelerData$Na,y=modelerData$Age,xlab="Na",ylab="Age")
   cor(modelerData$Na,modelerData$Age)
   ```

   The R object `modelerData` is automatically populated with the DRUG1n data set.

   When the node is executed, the R object `modelerModel` contains the results of the linear model analysis.
8. On the **Model Options** tab, select **Display R graphs as HTML**. When the node is executed, a plot of the target field Na against the input field Age is displayed on the **Graph Output** tab of the R model nugget.
9. On the **Model Options** tab, select **Display R text output**. When the node is executed, the correlation between the target field Na and the input field Age is written to the **Text Output** tab of the R model nugget.
10. In the **R model scoring syntax** field on the **Syntax** tab, enter the following R script:

```
result<-predict(modelerModel,newdata=modelerData)
modelerData<-cbind(modelerData,result)
var1<-c(fieldName="NaPrediction",fieldLabel="",fieldStorage="real",fieldMeasure="",fieldFormat="",
fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)
```

When the R model nugget is executed, the following R objects are created:

- The R object `result` contains the predicted values of the target field, Na, obtained from the model `modelerModel`.
- The R object `modelerData` is a data frame that contains the original data with an extra field that contains the predicted values of the target field.
- The R object `var1` sets up a new field for the data model that describes the type and structure of the predicted values of the target field.
- The R object `modelerDataModel` contains the data model for the original data with an extra field for the predicted values of the target field.

11. Click **Run** to execute the R Building node. An R model nugget is added to the Models palette.
12. Add the R model nugget to the stream canvas.
13. Add a Table node, from the Output palette, to the stream canvas.
14. To see the predicted values of the target field, connect the Table node to the R model nugget, double-click the Table node, and click **Run**.
15. The table contains the predicted values in the field named *NaPrediction*; this field was created by the model scoring R script.

Scripting Properties

This section lists the scripting properties specific to the R Building node and R model nugget.

R Building Node

The scripting type of the R Building node is `buildr`.

Example

```
set :buildr.score_syntax = """
result <- predict(modelerModel, newdata= modelerData)
modelerData <- cbind(modelerData, result)
var1<-c(fieldName="test", fieldLabel="", fieldStorage="real", fieldMeasure="", fieldFormat="",
fieldRole="")
modelerDataModel<-data.frame(modelerDataModel, var1) """
```

In addition to common node properties, the following are also available.

Table 1. buildr properties.

buildr Properties	**Values**	**Property description**
`build_syntax`	*string*	R scripting syntax for model building.
`score_syntax`	*string*	R scripting syntax for model scoring.
`convert_datetime`	*flag*	Option to convert variables with date or datetime formats to R date/time formats.
`convert_datetime_class`	`POSIXct` `POSIXlt`	Options to specify to what format variables with date or datetime formats are converted.

Table 1. buildr properties (continued).

buildr Properties	Values	Property description
convert_missing	*flag*	Option to convert missing values to R NA value.
output_html	*flag*	Option to display graphs on a tab in the R model nugget.
output_text	*flag*	Option to write R console text output to a tab in the R model nugget.

R Model Nugget

The scripting type of the R model nugget is applyr.

In addition to common node properties, the following are also available.

Table 2. applyr properties

applyr Properties	Values	Property Description
convert_datetime	*flag*	Option to convert variables with date or datetime formats to R date/time formats.
convert_datetime_class	POSIXct POSIXlt	Options to specify to what format variables with date or datetime formats are converted.
convert_missing	*flag*	Option to convert missing values to R NA value.

Notices

This information was developed for products and services offered worldwide.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group
ATTN: Licensing
200 W. Madison St.
Chicago, IL; 60606
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other product and service names might be trademarks of IBM or other companies.

Index

Printed in USA