



Spotfire Statistica®

R Integration: Features and Options

Version 14.2.0 | March 2024

Contents

Contents	2
Introduction	4
Overview and Summary	6
Basic Architecture and Features of R Support in Statistica	8
COM Interface to R Environment	9
R Integration Support Macros (R.svb and R.r)	10
R Scripts as Native Statistica Macros	11
Retrieving Results	12
Statistica Capabilities	13
R Language Extensions: Passing Data to R and Retrieving Results	14
ActiveDataSet	14
Spreadsheet(filePathOrName)	15
RouteOutput(x, name, description)	17
Uses(package, [lib], [repos], [quiet], [attachImports])	19
R Workspace Node	21
Executing R Scripts from within Statistica Visual Basic	22
R Scripts are Statistica Macros	22
Passing Parameters to R Scripts: Collection Object	23
Managing Script Results	27
More Examples	28
R Support in Statistica Server	29
Statistica Server is a Powerful R Server	30
Off-Loading R Scripts from Statistica to Statistica Server	30

Integrating R with Statistica Server	32
Deploying R Nodes as Part of End-to-End Validated Workflow	32
Creating R-based Analysis Configurations	32
Calling R Scripts from SVB Analysis Configurations	34
Final Comments, and Some Caveats	35
Error Handling	35
Strengths and Limitations	35
Spotfire Documentation and Support Services	37
Legal and Third-Party Notices	39

Introduction

R is a programming language and environment for statistical computing.

Most of the R environment and its source code are currently available under the GNU GPL license (<http://www.r-project.org/about.html>). Note that none of the components of the R environment constitute “unrestricted freeware”; instead they are available only under the terms of specific licenses that the users who intend to download those applications need to accept prior to downloading and with which they need to comply. Also, those licenses can change over time and, therefore, users are cautioned to thoroughly familiarize themselves with the terms every time they download any components of the R environment.

COMadaptR, licensed under LGPL (>=2.1) with parts under GPL2, facilitates communication to the R world. It is available to download from <https://tran.tibco.com/statistica/>. This package is based on the GPL2/LGPL2 version of an earlier application, the statconnDCOM library.

Statistica can interface with R via COMadaptR facilitating bidirectional data transfer and presentation of resulting outputs.

This interface makes it possible for all Statistica products to provide comprehensive support for interaction with the R platform, providing the ability to:

- run R scripts within the Statistica environment, sending results to Statistica Reports, Workbooks, and Graphs
- process Statistica data sets in R and import tabular results from R into Statistica Spreadsheets
- creating R nodes that can be managed with a metadata store and within templates; versioned, approved, audit logged
- utilize R in Statistica Server



Note: It is the user's responsibility to ensure compliance with terms of all applicable licenses for R and all components of the R environment. Always carefully review all the license agreements before accepting them as they can change over time.

Automatic installation of the COMadaptR support library is included in Statistica 12.0 SP3 and above when Internet access is available. When Statistica starts and detects that an R installation is present, the application will ask the user for permission to download and install from <https://tran.tibco.com/statistica/>.

Read the KB article, [Manually enabling Statistica-R Integration for Statistica](#), for manual installation instructions for the COMadaptR support library.

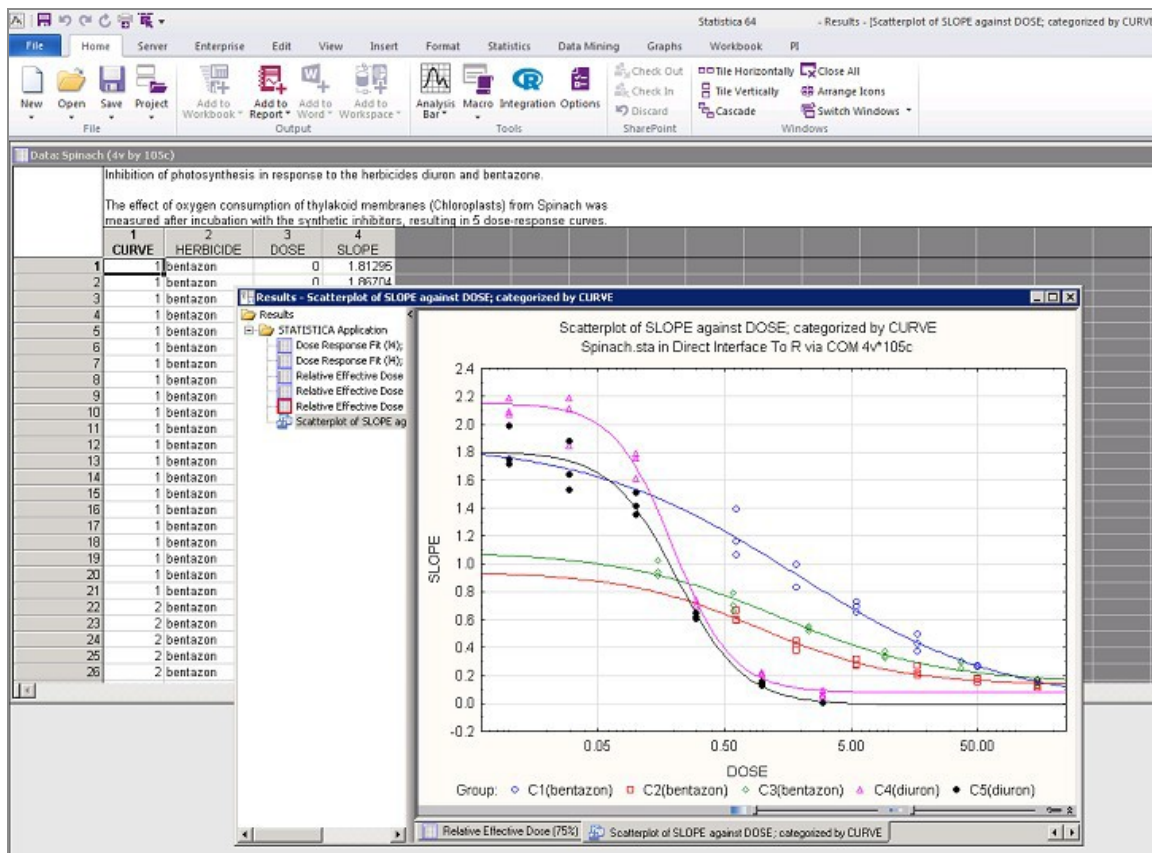
You can verify that the COMadaptR library is installed correctly and that all its dependencies are satisfied by running one of the R examples that accompany Statistica. After all the required third-party components are installed, you should be able to open and execute R scripts within the Statistica environment (on the **File** tab, click **Open Examples** and browse to the **R** folder).

R is highly extensible. Users can submit libraries (packages) implementing a set of functions, usually for a specific area of their expertise or research. The R community maintains several centralized repositories that make hundreds of such packages readily available to all users over the Internet. Many of these packages cater specifically to highly specialized audiences with particular data analysis needs.

Overview and Summary

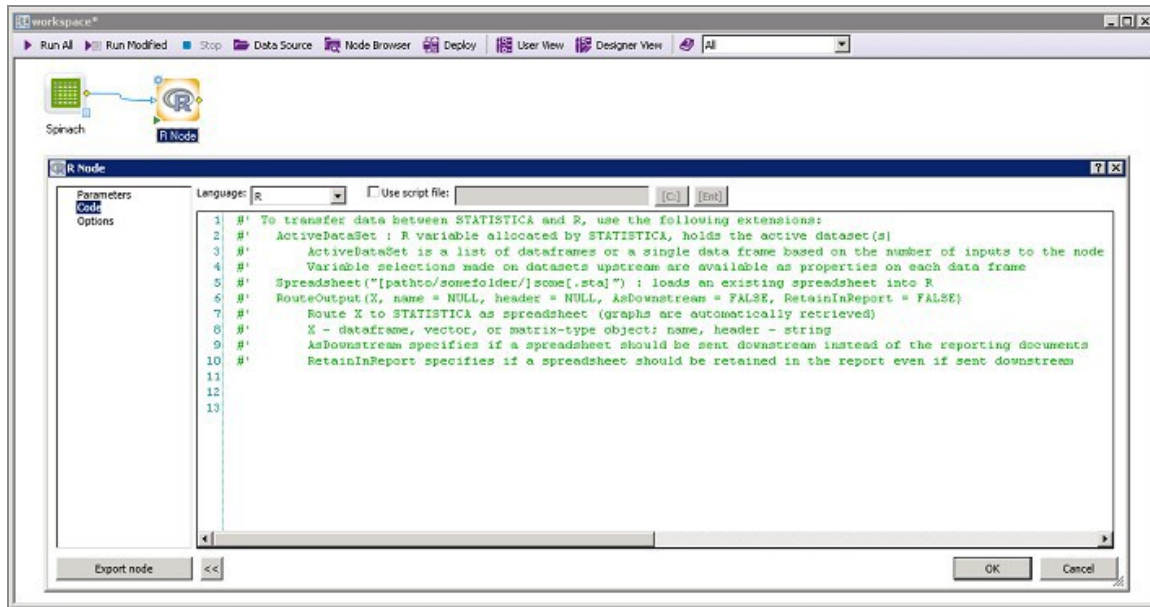
The goal of this document is to provide a detailed description of features that make the diversity and power of R fully available. These features enable users to combine the capabilities of both Statistica and R platforms.

- Native R scripts can be run directly within Statistica or Statistica Server
- R output can be retrieved as native Statistica Spreadsheets and Graphs, and managed with flexible Statistica Workbook containers



Thus, businesses can now use the specialized routines and capabilities of R with Statistica to add new R-based nodes to Workspaces. These Workspaces can be versioned, approved and shared with roles based security.

Integrating R into Statistica also can make specialized R functionality available as reusable analysis templates to users not familiar with the R language.



Basic Architecture and Features of R Support in Statistica

R support in Statistica was designed to create an integrated Statistica-R platform that enables users to run R programs or scripts directly inside Statistica so that they can take full advantage of the specialized capabilities available in R.

The R Integration environment in Statistica was specifically designed for the following enhancements:

- Enable users to run R scripts as is, and retrieve the results into Statistica reports
 - all R console output is copied into the report; R commands are highlighted
 - plots generated by the script are automatically embedded in the report as scalable images
 - these plots are also replicated as Statistica Graphs (scalable “metafile” images are placed inside these graphs), thus enabling annotation using powerful graphical facilities in Statistica; the graphs can then be printed or exported into a variety of image formats
 - the reports can be edited, printed, and saved as PDF files
- Provide R language extensions functions for R scripts run from the Statistica environment that:
 - transfer data from Statistica Spreadsheets into R data frames
 - extract tabular data from R variables into Statistica Spreadsheets; these “results” spreadsheets (as well as all graphs produced by the script) are returned according to Output Manager settings (“routed”), e.g., placed into Statistica Workbooks
- Execute R scripts as native Macros from within Statistica Visual Basic (SVB) programs
 - scripts can be parameterized with a Collection of objects (numbers, strings, arrays, additional R code or overridden R functions, spreadsheets) that are mapped to R variables accessible to the script; this approach provides fine-grained control over scripts’ behavior in repeated runs or when used as the

backend for custom Statistica modules.

- by default, all script output is routed by Statistica Output Manager; scripts may also be executed using a method that instead returns its output as a Document Collection, giving developers an easy way to extract specific analysis results that could be used for further processing, e.g., as input data for further analyses in Statistica or in R, or for graphing.
- R Node is integrated into Statistica which makes specialized R functionality available as reusable analysis templates to users who are not familiar with the R language.

Taken together, these enhancements not only enable users to run R scripts directly in the Statistica desktop environment, but also provide a way to embed specialized R functionality into custom interactive analysis modules, Workspace nodes, analysis configurations, or to offload such scripts to Statistica Server for server-side processing.

Developing and debugging R scripts

Statistica does not supply a complete R development and debugging environment. The console application and tools supplied with standard R installation perform those functions very well, and are already familiar to R users and developers.

COM Interface to R Environment

The R environment must be installed on the same computer with Statistica desktop or on the Statistica Server. The latest version of the R environment can be obtained from the CRAN website (<http://cran.r-project.org>).

To access the R environment, Statistica uses the COMadaptR library distributed under GNU Lesser Public License; this library has two components: one of them acts as an R COM Server; the other is used for callbacks from R to Statistica.

After R is installed, Statistica automatically detects and downloads the version of COMadaptR based on the R version installed on the machine. For example, for the version R 3.x, COMadaptR version 1 is detected and downloaded.



Tip: It is best practice to use R versions 3.6.1 and 4.2.2.

The next time you run Statistica, a dialog asks if R Integration should be enabled. Click Yes, and Statistica automatically installs the COMadaptR library and register one of the components with the Windows registry. Note that these steps might require administrative privileges – depending on your operating system version and settings, the system might ask you to confirm these actions, possibly requesting administrator credentials. In addition, access to the Internet is required since COMadaptR is downloaded from <https://tran.tibco.com/statistica/>.

Read the KB article, [Manually enabling Statistica-R Integration for Statistica](#), for manual installation instructions for the COMadaptR support library.



Note: The COMadaptR library is independent from Statistica and remains on your computer until it is uninstalled manually.

The COMadaptR library provides a simple yet powerful COM (Component Object Model) interface to the R environment. This interface can be used directly by SVB programs in Statistica – an example of such a use case is included in the Statistica application, located in Examples\R\Dose Response folder (open Direct Interface To R via COM.stw and run the embedded SVB macro). But usage of such an interface directly by end users is very ineffective, sometimes unproductive, and usually inflexible. It may also significantly degrade the overall performance of interactions with R if performed incorrectly.

Therefore, the following architectural extensions have been added to the Statistica platform to provide a seamless and effective R Integration experience for end users. The example mentioned previously also demonstrates the significant reduction in the efforts required to implement the same analysis using the new built-in features in Statistica, simply open and run DoseResponse.r.

R Integration Support Macros (R.svb and R.r)

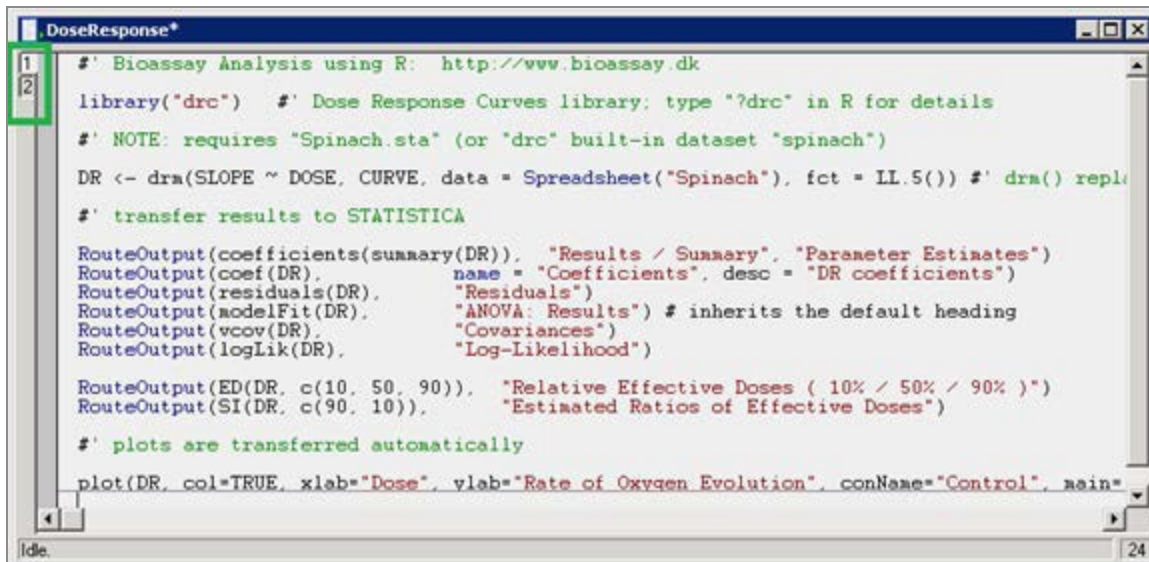
Statistica installation includes a Statistica Visual Basic macro called R.svb and an R script called R.r. These files contain the support code required to manage interactions through COM in Statistica and the R side respectively. When an R script is executed in Statistica, it is parsed by the support macro, which then transfers data and script parameters, submits script content to the R environment, manages error conditions, and also handles script outputs, ensuring that they are properly transferred back to Statistica. Support script R.r implements Statistica-specific R language extensions functions; one of its primary responsibilities is translating Statistica Spreadsheets to R data frames and back.

Although the support code is write-protected by default, it is accessible for inspection and can be modified or enhanced to support new functionality required for specific use cases, although users should do so at their own risk. The R.svb macro supports standalone execution to simplify debugging and testing of modifications.

R Scripts as Native Statistica Macros

Statistica recognizes .R (and .S) file extensions as R scripts. Such files can be opened by selecting the **File** tab and clicking **Open**. It also registers at the operating system level, .R/.S files as Statistica Macros, and therefore these files can be opened in Statistica from a file browser by double-clicking on them.

R scripts are displayed in slightly modified Statistica Visual Basic Macro windows. Such windows actually contain two scripts: the R script itself and the R Integration Support Macro (R.svb), accessible through two tabs in the upper-left corner (circled on the next image).



```

1  #' Bioassay Analysis using R: http://www.bioassay.dk
2  library("drc")  #' Dose Response Curves library; type "?drc" in R for details
  #' NOTE: requires "Spinach.sta" (or "drc" built-in dataset "spinach")
  DR <- dra(SLOPE ~ DOSE, CURVE, data = Spreadsheet("Spinach"), fct = LL.5()) #' dra() repl
  #' transfer results to STATISTICA
  RouteOutput(coefficients(summary(DR)), "Results / Summary", "Parameter Estimates")
  RouteOutput(coef(DR), name = "Coefficients", desc = "DR coefficients")
  RouteOutput(residuals(DR), "Residuals")
  RouteOutput(modelFit(DR), "ANOVA: Results") # inherits the default heading
  RouteOutput(vcov(DR), "Covariances")
  RouteOutput(logLik(DR), "Log-Likelihood")
  RouteOutput(ED(DR, c(10, 50, 90)), "Relative Effective Doses ( 10% / 50% / 90% )")
  RouteOutput(SI(DR, c(90, 10)), "Estimated Ratios of Effective Doses")
  #' plots are transferred automatically
  plot(DR, col=TRUE, xlab="Dose", ylab="Rate of Oxygen Evolution", conName="Control", main=

```

There is limited R code highlighting available (strings, language extensions, VB-style comments).

Creating a new R script

You can also create a new R script within Statistica. Select the **File menu > New menu > Macro tab**, and select the R (requires R Statistical Environment) option button. This option will only be available if R Integration support is installed on your machine.

This will open an empty R script window. You can now type or paste in an R program. R Integration support also includes an optional text file called `R.inc`, placed into the default installation directory along with `R.svb`. The contents of the file are copied to the beginning of each new R script created in this manner.

Running R Scripts within Statistica

Click the toolbar button or select the **Run > Run Macro** menu command or click F5. This action executes the `R.svb` macro for the currently active R script.

Although breakpoints are not supported for the R script itself, it's possible to set breakpoints and debug the `R.svb` macro on the second tab while running the R script.

i Note: In order to take advantage of R Integration features described in this document, R scripts should be executed from within Statistica. Although it is possible to develop and debug complex R programs within this environment, it was not specifically designed for these purposes; the R environment itself might be better suited for such activities.

Retrieving Results

Console Session

The minimal output produced during execution of an R script is a Statistica Workbook that represents an R console session, including highlighted commands and any output generated by the R environment. Such a report will be produced even if the script is empty. The contents of this report can be edited and manipulated in the same way that you would edit any other Statistica report.

Graphical Output

All plots created during an R session are automatically transferred into the Statistica environment as Windows Metafiles (vector graphics format), which means they can be resized without loss of quality.

These plots are placed into the R console session report, creating a natural flat report of the R session with embedded plots tied to the graphics commands.

Moreover, the plots are also replicated as Statistica Graphs that become a part of the R script output. The metafile images are embedded into graph objects as locked “background” this enables users to annotate R plots in Statistica using a familiar point-and-click interface with a set of text and drawing objects (such as lines and arrows, rectangles and ellipses, polygons and pattern/color fill areas, etc.). And since these annotations are anchored to relative positions in the plot area, they will remain correctly “attached” to the plot if the graph is resized.

Therefore, these graphs can be flexibly designed and further enhanced using Statistica graphics tools, saved in other formats (e.g., JPG or GIF), or printed (e.g., to PDF files).

The individual R plot components (the structural elements of the plot) are not accessible for manipulation in Statistica Graphs, and hence, the rich capabilities of Statistica for creating and then further editing graphs (scaling, point markers, fit lines, etc.) are not available. However, integration between R and Statistica provides opportunities to extract data from R and then render important graphs inside the Statistica environment (by writing Statistica Visual Basic macros that will execute R scripts, extract results, and then post-process those results as necessary; this will be illustrated later).

Statistica Capabilities

Since R script output consists entirely of native Statistica objects,

- these objects are properly “routed” by Statistica according to Output Manager settings, which means that, depending on user selection, they could be placed into standalone windows or in a workbook
- each and all of them can be further managed using the extensive capabilities of the Statistica platform; e.g., they can be annotated; stored in a (compressed) workbook; exported as Microsoft Office documents; printed; saved as a PDF file; converted to one of many popular formats; archived as version-controlled, “auditable” items in Statistica Enterprise; shared among other users in a web-based, client-server Statistica Server environment, and so on.

R Language Extensions: Passing Data to R and Retrieving Results

The R Integration Support Macro (R.r) implements several extensions to R language – keywords and functions that can be used inside R scripts executed within the Statistica environment. These extensions enable scripts to pass data to R and retrieve results from the R environment.

Important: The R language is case sensitive; therefore R language extensions for Statistica are also case sensitive - they are only recognized by the Statistica environment when typed exactly as shown.

Tabular data represented in Statistica in the form of spreadsheets are mapped into the equivalent R structures – data frames. The mapping preserves as much information as possible for both formats: for example, text label variables in a spreadsheet become factor objects in a data frame, Missing Data values are mapped to NA indicators; data types and variable and case names are transferred both ways, etc.

ActiveDataSet

The ActiveDataSet keyword was adopted from the Statistica Visual Basic language and it performs the same function in R scripts: it references the active Statistica data spreadsheet.

In the desktop Statistica environment, active data set usually means the top-most visible spreadsheet, which can act as a data source. It can also be a spreadsheet in a workbook selected as Active Input. This notion is redefined and extended for server-based environments (Statistica Server, Enterprise), but the keyword is still valid and refers to the corresponding server-side mapping of the active data source. If no active data set is defined or available, the R script that uses it fails. The same is true for SVB macros.

If the R Integration Support Macro encounters the ActiveDataSet keyword in the R script, it transfers the actual Statistica active data set into the R environment and assigns it to a variable of the same name. Therefore, this keyword represents a data frame variable and can be handled as such in the script.

Example:

```
ActiveDataSet[1:5] # display a subset of active dataset(variables 1 through5)

str(ActiveDataSet) # view the structure of the data frame

plot(ActiveDataSet$MEASURE01) # plot"MEASURE01"variable values
```

Spreadsheet(FilePathOrName)

Spreadsheet(path.or.object, [applySelectionConditions], [recodeTextLabelAsFactors], [recodeMissingDataToNA], [getCaseNames], [getTextValues], [attachObject], [username], [password], [connection string], [stationname])

path.or.object – character vector (literal or variable) or a valid COM object pointing to a Statistica spreadsheet, e.g. “c:/data/sample.sta”, “enterprise://SomeFolder1/SomeFile.sta” or “adstudy”.

applySelectionConditions – TRUE (default) or FALSE flag indicating whether selection conditions defined on spreadsheet should be employed.

recodeTextLabelCodesAsFactors – TRUE (default) or FALSE flag indicating whether text labels defined for variables should be treated as factors or text.

recodeMissingDataToNA – TRUE (default) or FALSE flag indicating whether the missing data should be treated as not applicable.

getCaseNames – TRUE (default) or FALSE flag indicating whether row names should be fetched.

getTextValues – TRUE (default) or FALSE flag indicating whether text variables are returned.

attachObject – TRUE or FALSE (default) flag indicating whether spreadsheet should be attached to data frame as an “Object” attribute.

"username, password, connectionstring, stationname" – character strings (default value: “”) to be used to fetch spreadsheets from the enterprise system when user does not have an integrated login.

Use the **Spreadsheet()** extension to load a specific Statistica data file into R and transfer the data in that file to an R data frame.

Similar to the `ActiveDataSet` keyword, the return value of the **`Spreadsheet()`** function should be treated as a data frame variable with the contents closely matching that of the corresponding Statistica spreadsheet.

One useful feature supported by this function is the use of default search paths for spreadsheet files that are specified only as simple file names. This means that if the function parameter consists only of a file name, e.g., `Spreadsheet("some.sta")`, R Integration Support code looks for this file in several locations: first, it checks the folder where the R script itself is located (if it was saved to disk), and then it checks the `Examples\Datasets` folder for the current Statistica installation. Support code also appends the default `.sta` file extension if one is not present. Therefore, the following options are available:

- R scripts can reference the accompanying data sets (placed in the same folder) simply by name.
- Spreadsheets that are included in every Statistica installation as demonstration/example data sets can be referenced by name in much the same way as built-in R data sets.

Example:

```
Spreadsheet("c:\\myfiles\\mydata.sta") # display the dataset in R console

Spreadsheet("\\server\\share\\data.sta") # same, but read from a network
share

Spreadsheet("thisdemo.sta") # file in the same folder as this script

Spreadsheet("enterprise://folder/thisdemo.sta") # file in Statistica
Enterprise

# plot a histogram for MEASURE05 variable from Statistica example
dataset Adstudy: hist(Spreadsheet("Adstudy")$MEASURE05)

...

advert = Spreadsheet("Adstudy")$ADVERT # retrieve a data frame variable
(factor)

is.factor(advert)# => [1] TRUE

levels(advert) # => [1] "PEPSI" "COKE"
```

RouteOutput(x, name, description)

x – a data frame, matrix, array, number, or a string

name – character vector containing name (in Statistica) for the object being routed

description – character vector containing a description for the object being routed

SpreadsheetName, SpreadsheetHeader – literal string, e.g. “A Frequency Table”

This extension transfers various types of data from the R environment into Statistica Spreadsheets.

Although the function was introduced to retrieve tabular data (such as data frames, matrices, or arrays) into spreadsheets, single-value data such as numbers or strings can be passed as well and will be placed in single-cell spreadsheets. x can be an R variable or a literal value.

The name of this extension, RouteOutput(), hints at the similarity of its behavior to the equivalent Statistica Visual Basic function: the “results” spreadsheets recreated by the function in the Statistica environment become the standard output of the R script / analysis and follow Output Manager settings (in Statistica, select the Tools tab, click Options, and select the Analyses/Graphs: Output Manager tab), i.e., they are “routed” either to individual windows or to a workbook (or multiple workbooks for each analysis, with optional output reports, e.g., as a Microsoft Word document); the most popular setting is a single results workbook.

Optional parameters name and description specify the name and header of the resulting spreadsheet. It is recommended to provide a value for the spreadsheet name for visual distinction in the tree view of the results workbook.

Note that R plots transferred into Statistica as native graphs do not require explicit output routing – all plots generated during a script run are automatically transferred and routed according to Output Manager settings.

Important: Many functions in R, specifically the ones that perform statistical modeling, represent their results as structured objects, sometimes of significant complexity. These objects cannot be reduced to a single table, and therefore cannot be handled by the RouteOutput() extension (they could be automatically traversed in search of tabular components, but since the object structures are specific to a particular method, such an approach would generate a significant amount of “junk” output). However, since the results (the actual data of interest) are either stored in such objects as tabular components or produced by applying an object’s method to some input data, this limitation does not

pose any problems – particular results can be easily extracted from such a statistical model object and routed back to Statistica.

Example:

```
# build a statistical model (Dose Response Curves)

model <- multdrc(SLOPE ~ DOSE, CURVE, data = PestSci)

...

str(model) # display the complex structure of this model/results in R console

...

# transfer results to Statistica

RouteOutput(coefficients(summary(model)), "Summary", "Parameter Estimates")

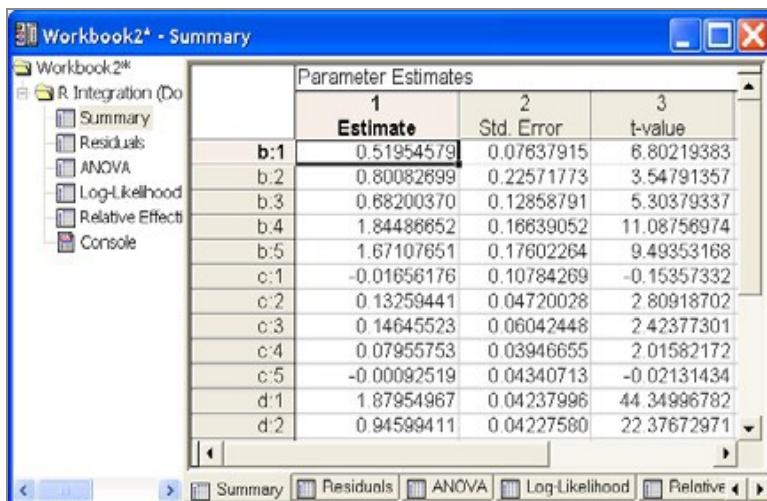
RouteOutput(residuals(model), name = "Residuals", header = "Some Header")

RouteOutput(anova(model), "ANOVA") # inherits the default heading from data frame

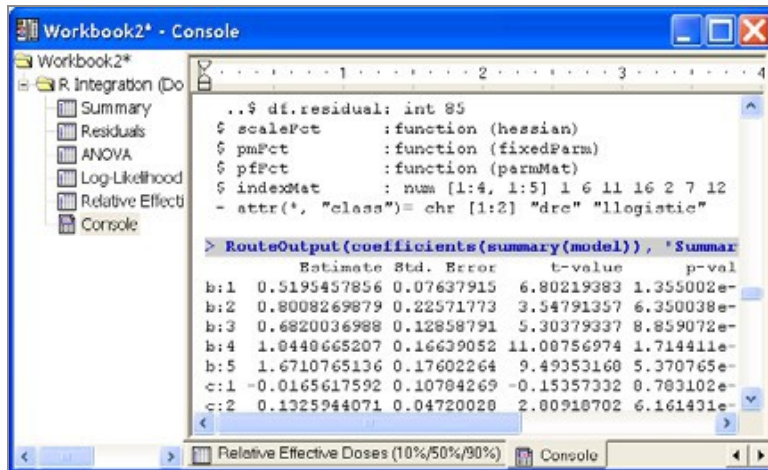
RouteOutput(logLik(model), "Log-Likelihood")

RouteOutput(ED(model, c(10, 50, 90)), "Relative Effective Doses (10%/50%/90%)")
```

The results for this example may look like this:



	1 Estimate	2 Std. Error	3 t-value
b:1	0.51954579	0.07637915	6.80219383
b:2	0.80082699	0.22571773	3.54791357
b:3	0.68200370	0.12858791	5.30379337
b:4	1.84486652	0.16639052	11.08756974
b:5	1.67107651	0.17602264	9.49353168
c:1	-0.01656176	0.10784269	-0.15357332
c:2	0.13259441	0.04720028	2.80918702
c:3	0.14645523	0.06042448	2.42377301
c:4	0.07955753	0.03946655	2.01582172
c:5	-0.00092519	0.04340713	-0.02131434
d:1	1.87954967	0.04237996	44.34996782
d:2	0.94599411	0.04227580	22.37672971



Uses(package, [lib], [repos], [quiet], [attachImports])

package – character vector containing name of package, e.g. “drc”

lib – character vector containing path to library directories to install package if needed

repos – list of repositories to use to download package if needed

quiet – TRUE or FALSE (default) flag indicating whether interactive dialogs should be suppressed during package installation

attachImports – TRUE OR FALSE (default) flag indicating whether or not attach imports from the package and its dependencies to the current R session

The Uses() function can be used to ensure that the respective package (library) named package AND its dependencies (packages that are required in order for this package to run) are installed (on the computer/server where the R script is running) and loaded into the R environment automatically. If some of these libraries are not present, they will be installed and then loaded.

R defines several levels for specification of package dependencies. One of them is Imports – it lists packages with namespaces that have to be loaded for the package to run, but the namespaces do not need to be attached to the current/global environment, meaning that the package accesses such imported packages using namespace-qualified names, e.g., tseriesChaos::mutual(x) instead of mutual(x). If you need to use some of these namespace-enclosed functions in your own R code, you can either explicitly use namespace qualification [tseriesChaos::mutual(x)], attach the library to current environment [Uses

(`"tseriesChaos"`) or `library(tseriesChaos)`], or set the optional `Uses()` parameter `attachImports` to `TRUE` (this will attach all the imports of the package).

Note that this extension is not necessary for interaction between Statistica and R, and it's possible to implement it within the R language itself; rather, it simplifies the process of conditional library installation and loading by encapsulating it in a single call.

Example:

```
Uses("drc") # make sure that the respective package is installed and loaded  
...  
DR <- multdrc(SLOPE ~ DOSE, CURVE, data = PestSci) # call the package methods
```

This program fits dose response curves to the respective variables of the built-in `PestSci` data set by calling `multdrc` function defined in `"drc"` package. `Uses("drc")` ensures that the function is available by installing and loading the package, if necessary.

R Workspace Node

The R workspace node method is used to save R node in the Enterprise Manager workspace and run it on the Statistica server.

For more information, see the [General Scripting Nodes and General Code nodes](#), [Statistica Workspace Toolbar](#), and [Help Page in Enterprise Manager](#).

Executing R Scripts from within Statistica Visual Basic

A typical use case for leveraging specialized R functionality within Statistica is to call R scripts from inside a Statistica Visual Basic (SVB) macro. This way, you can build new modules using the SVB User Interface library and methods implemented in R scripts. Likewise, such functionality is required in order to create Statistica Enterprise SVB analysis configurations, or Statistica Workspace nodes (for Statistica or Statistica Server) that leverage R.

However, to provide any non-trivial functionality within an R script, you need to parameterize that script with user-selected parameters, variables lists, input spreadsheets, and so on. Statistica provides a simple and powerful way to pass such parameters to R scripts.

R Scripts are Statistica Macros

In most cases, Statistica treats R scripts in the same way as native SVB macros. This applies to the Statistica Object Model as well: Macro objects in SVB programs can now represent R scripts. Therefore, R scripts can be created, opened, edited, saved, and executed from within SVB scripts.

This in turn means that R functionality is available in Statistica Enterprise analysis configurations and Statistica Workspace nodes since they are SVB-based.

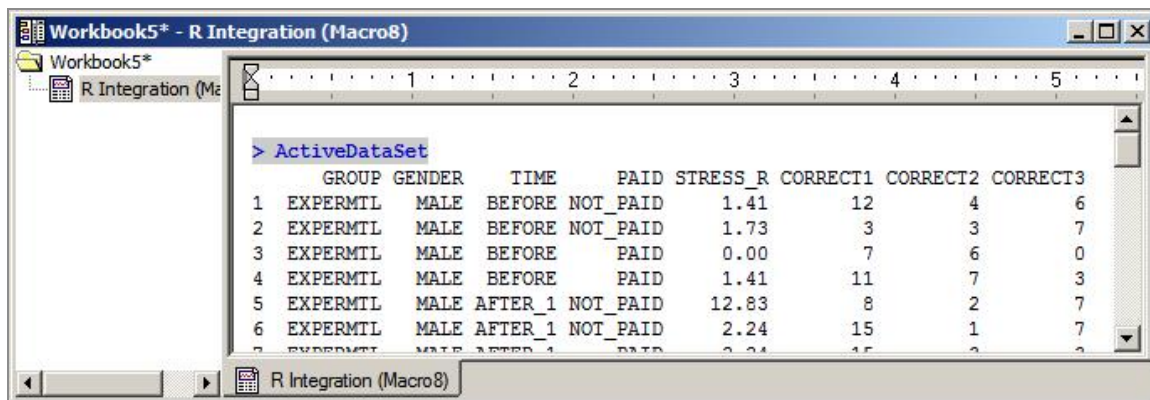
Existing R script files can be opened with `Macros.Open("path\to\some.r")` or created on-the-fly with `Macros.New()` and `Macro.Code`. Note that in the latter case, Statistica needs help in distinguishing R scripts from SVB macros – this can be achieved either by specifying the name for a new macro with the `.R` extension (even if you are not going to save it on disk), or by explicitly setting `Macro.Scripting` to 5 (R Macro Type). Run the scripts by calling `Macro.Execute`.

Important: The `Macro.Scripting` type for R scripts is 5 (later will be mapped to a symbolic constant).

Example:

```
Sub Main
Dim R As New Macro
R.Code = "ActiveDataSet" ' simple R script created on-the-fly
R.Scripting = 5 ' R Macro Type = 5
R.Execute End Sub
```

This Statistica Visual Basic macro runs a simple R script containing only a single command `ActiveDataSet` which, as described in the previous section, is an R language extension for Statistica that will transfer (and in this case display) the currently active Statistica data file in R. For example, if you run this macro after opening the example data file `Exp.sta`, a listing of that file will be displayed in a report window that represents the R console session:



	GROUP	GENDER	TIME	PAID	STRESS_R	CORRECT1	CORRECT2	CORRECT3
1	EXPERMTL	MALE	BEFORE	NOT_PAID	1.41	12	4	6
2	EXPERMTL	MALE	BEFORE	NOT_PAID	1.73	3	3	7
3	EXPERMTL	MALE	BEFORE	PAID	0.00	7	6	0
4	EXPERMTL	MALE	BEFORE	PAID	1.41	11	7	3
5	EXPERMTL	MALE	AFTER_1	NOT_PAID	12.83	8	2	7
6	EXPERMTL	MALE	AFTER_1	NOT_PAID	2.24	15	1	7

Passing Parameters to R Scripts: Collection Object

Collection is a COM library introduced with Statistica Version 8 MR3 that implements generic Collection objects.

Such objects enable users to store keyword-tagged Variant values. Variants are COM structures that can store one of many supported data types, including arrays and references to other COM objects, example, spreadsheets.

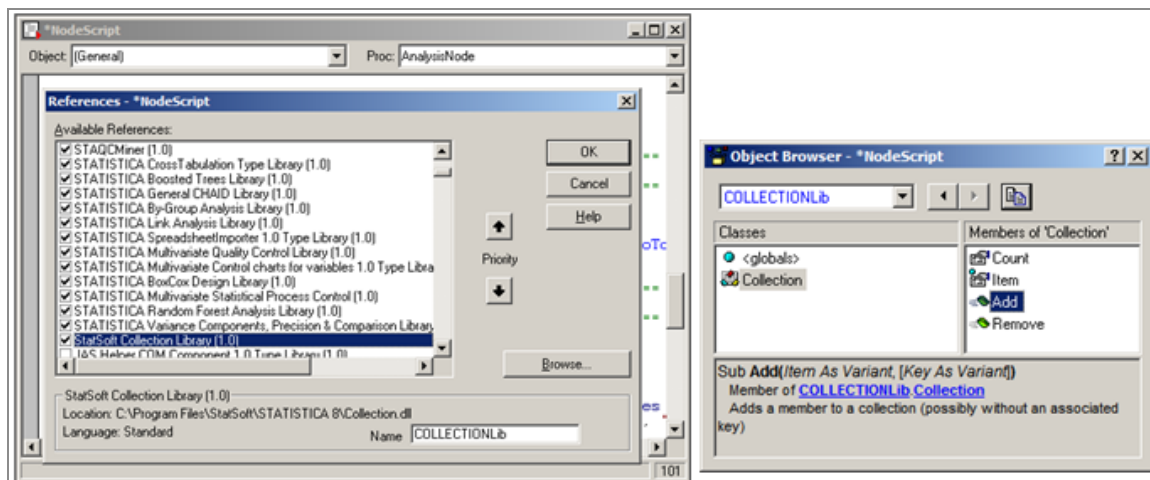
What this means for R Integration is that you can now store {parameter name = parameter value} pairs in such a Collection, then pass this parameter set to Statistica prior to R script execution, thus achieving script parameterization. The R Integration Support macro

transfers all the parameters from the Collection to the R environment as R variables named using the respective parameter keyword tags, making it possible for the script that is executed in this environment to reference all these variables (script parameters) directly by name.

R Integration supports the following types of named parameters: strings, numbers, arrays, and spreadsheets. Support for Statistica Spreadsheets is particularly useful: for example, such a parameter can override the `ActiveDataSet` keyword, allowing a single script to serve as an R engine for an interactive module, Enterprise analysis configuration, and Workspace node implementations.

Also note that string parameters without a keyword tag are treated as R code that should be executed prior to execution of the script itself. This is analogous to SVB (hidden code) and can be used to, for example, define a common set of new functions or global variables or constants.

In order to use Collection objects, SVB scripts must include a reference to the Spotfire Statistica® Collection Library (add it using the Tools > References menu while editing the macro):



The Collection object has several generic properties and methods that are needed to manipulate the contents of a collection: `Count`, `Add(Item, [Key])`, `Remove(KeyOrIndex)`, and `Item(KeyOrIndex)` that returns an `Item` object with `Key` and `Value` properties. However, due to the use of so-called default object properties (`Item` is the default property of a `Collection` and it returns its `Value` property by default) interaction with a `Collection` object is reduced to intuitively clear assignment operations:

```
Dim param As New Collection
param("number") = 57
param("string") = "A string sample..."
```

After you have assembled the parameter collection, execute the parameterized R script by calling `Macro.ExecuteWithArgument(Parameters as Collection)`.

Example:

```
Dim s1 As New Spreadsheet, s2 As New Spreadsheet ' ... populate s1, s2 with data

var1 = Array("CASE 1", "CASE 2")
var2 = Array(1, 2, 3, 4, 5)

' * don't use spaces in parameter names
' * some names are "locked" and can't be used [e.g. 'text', 'str', 'sample']

Dim param As New Collection
param("number") = 57
param("string") = "A STRING sample..."
param("string_array") = var1 ' add items with an assignment operator
param.Add(var2, "number_array") ' OR using explicit Add() method
param("SomeSpreadsheet") = s1
param("ActiveDataSet") = s2 'override the value of 'ActiveDataSet' keyword

' string parameters without associated keys will be treated as R code and
' will be executed before the script - an analog of SVB 'hidden code' ' * define a function that
will be available to the R script

param.Add("func <- function(x) { cat('Called func(x) with x =', x) }")

' * another way to define a global constant or variable
param.Add("Statistica.Version = '" & Version & "'")
```

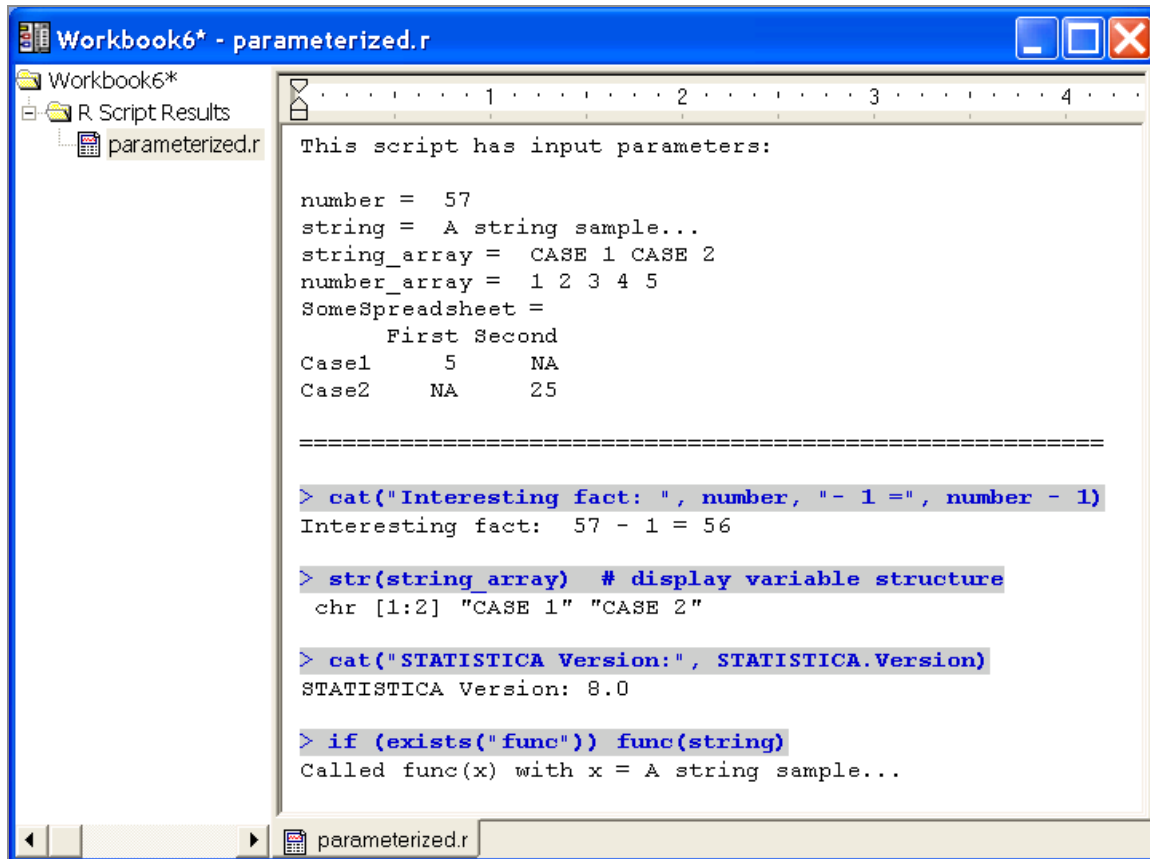
' now run the R script with this collection of parameters

' (parameters become R variables – the script can reference them by name)

Dim m As Macro

Set m = Macros.Open(MacroDir & "\parameterized.r")

m.ExecuteWithArgument(param)



The R workspace node method can be used to save R node in a workspace in Enterprise Manager and run it on Statistica Server.

For more information, see the [Help](#) page.

Managing Script Results

Both `Macro.Execute()` and `Macro.ExecuteWithArgument()` methods handle R script output (spreadsheets, graphs, reports) exactly the same way as the output from SVB macros – it is routed by Statistica according to Output Manager settings, e.g., placed into a results workbook.

This approach might be sufficient for many use cases, but is not flexible enough in situations where the output from an R script is treated as an intermediate result for subsequent processing in Statistica.

For example, an SVB script implementing a custom interactive module might implement a multiple stage analysis workflow, calling into R, analyzing the results, requesting additional inputs from the user, and calling into R repeatedly with previous results as inputs. Or an SVB macro could use tabular results returned from R to produce graphs that are more expressive and flexible than what R is able to generate.

Since the items processed by the Statistica Output Manager are intended to be a final representation of analysis results, it is not trivial to access them individually for further processing – one would need to know the type of representation selected by the user (single or multiple workbooks, individual windows) and programmatically iterate through the respective set of output items (windows, workbook contents represented as a tree), searching the item of interest by name.

Macro.ExecuteNoRouteOutput([Parameters as Collection]) As StaDocCollection

This method has been added to accommodate such special use cases: R script output is not “routed” at all, so the script runs silently; all the output from the script is collected into a `StaDocCollection` object (standard Statistica container for documents such as spreadsheets, graphs, reports, or workbooks) that is returned as the result of script execution. It also accepts an optional `Collection` for parameterization.

Now SVB macro developers can easily access individual components of R script output, e.g., to extract individual cell data from spreadsheets or to create a complex graph based on multiple columns from several spreadsheets.

Example:

Dim m As Macro

Set m = Macros.Open(MacroDir & "\some.r")

```
Dim Routput As StaDocCollection
```

```
Set Routput = m.ExecuteNoRouteOutput() ' silent, no output displayed
```

```
' do something with "Routput": extract and operate on individual documents ' trivial case:  
results are routed (displayed), as if from Macro.Execute()
```

```
RouteOutput(Routput, "R Script Results").Visible = True
```

More Examples

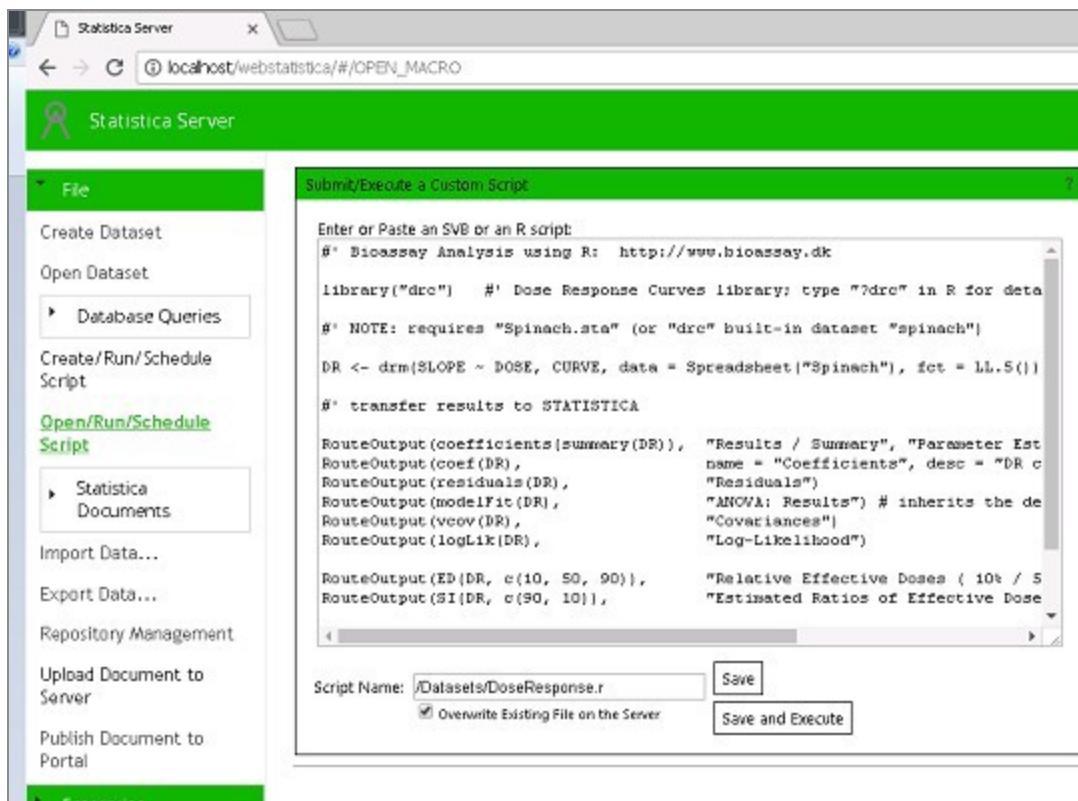
At this point it is demonstrated that all the functional components required to build custom applications within the Statistica platform can take advantage of the specialized functionality available in R.

All installations of Statistica now have a set of examples that provide a more detailed demonstration of the described features; you will find these examples in the [Statistica]\Examples\R folder. These examples may also be used as templates for development.

R Support in Statistica Server

Statistica and Statistica Server are based on identical Statistica libraries, and support identical functionality. This is true for R support in Statistica as well: you can execute R in Statistica Server in much the same way as from desktop (thick-client) Statistica.

For example, sign into Statistica Server, and select **File > Upload Document to Server** and then run. Or select **File > Create/Run/Schedule script** and copy text from standalone.r which is available in the R folder. This folder is located in the Statistica executable directory, for example, c:\Program Files\Statistica\Statistica 13\Examples\R.



Important: Make sure to save a new script with the .R extension.

Note that this script does not require a data file. If you are using `ActiveDataSet` or `Spreadsheet()` extensions, keep in mind that the data sources have to be located on the server. `Spreadsheet()` should reference data sets by their Statistica Server URL paths; `ActiveDataSet` will be mapped to the spreadsheet that you open in Statistica Server, but this can be overridden by a parameter of the same name passed to the R script from SVB.

Also, in order to execute any scripts in Statistica Server, the user must have the proper permissions on the server.

Statistica Server is a Powerful R Server

Statistica Server was designed as a powerful and flexible server/web-based analytical platform, relying on the Statistica Visual Basic engine for diversity of its functionality, as well as extensibility.

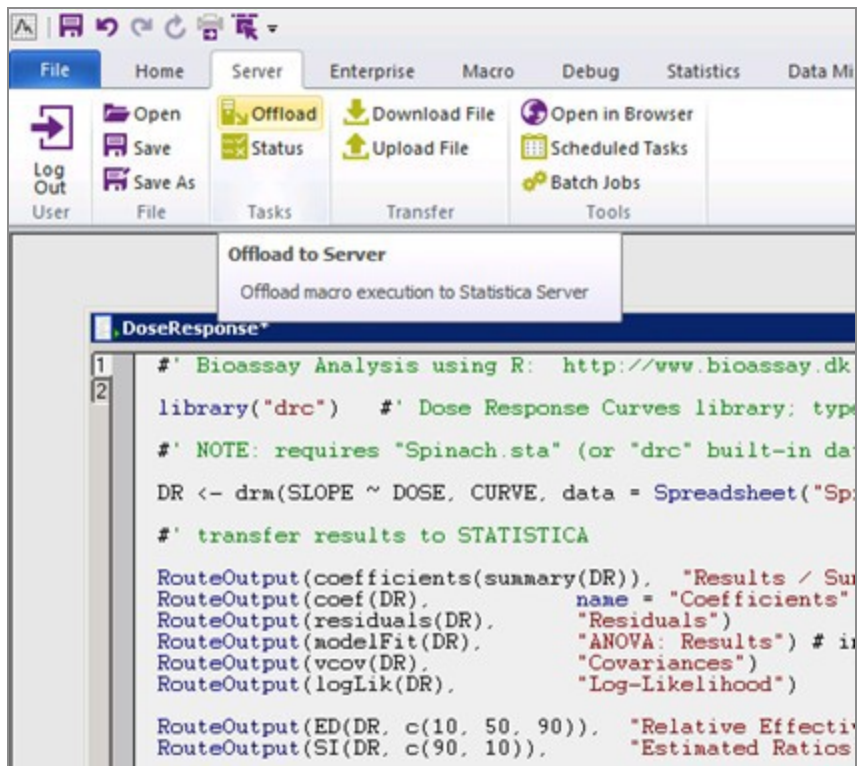
R scripts are handled by Statistica Server in much the same way as standard SVB macros. The third-party components that the Statistica platform relies upon to provide its R runtime environment (such as R library and R COM Server library) are also well suited for handling multiple simultaneous R sessions.

Thus, Statistica Server represents an ideal platform for a powerful and flexible multi-processor R server that can handle a large number of users, providing security, scheduling, load balancing, etc.

Off-Loading R Scripts from Statistica to Statistica Server

Like SVB scripts or Workspaces, R scripts can be offloaded from a desktop environment for processing on Statistica Server. This is particularly useful when an R script is expected to require significant computing resources; users can submit such scripts to the server (example, overnight) and retrieve the results at a later time.

If your installation is set up to work with Statistica Server (Tools → Options: Server/Web), you can offload any R task by selecting Server → Offload Task:



Options to transfer the data file to the server side is described in the Statistica documentation.

After clicking OK, the respective R script will execute on the Statistica Server or will be scheduled to be executed, depending on the server load.

The progress of the analysis can be monitored using the Server → Task Status... dialog.

The results can be retrieved from the server using the Results button (or double-click on the task); the representation of the results of an offloaded task is equivalent to the same task running locally.

Integrating R with Statistica Server

R Integration in Statistica would be incomplete without support for Statistica Server.

Now, R programs can be used in or as analysis templates. This means R analyses can be distributed to end users who are not familiar with VB or R programming. Furthermore, because the Enterprise platform provides numerous options and features to specifically enable the application of template analyses for validated applications (example, for manufacturers who need to comply with the requirements of FDA 21 CFR Part 11), such as version control and audit trails, these features make R available for deployment in mission-critical manufacturing applications.

Deploying R Nodes as Part of End-to-End Validated Workflow

To deploy R nodes as a part of an end-to-end validated workflow, click the **Deploy Workspace to Enterprise** button to add the entire workflow as an Analysis Configuration in the Enterprise Manager.

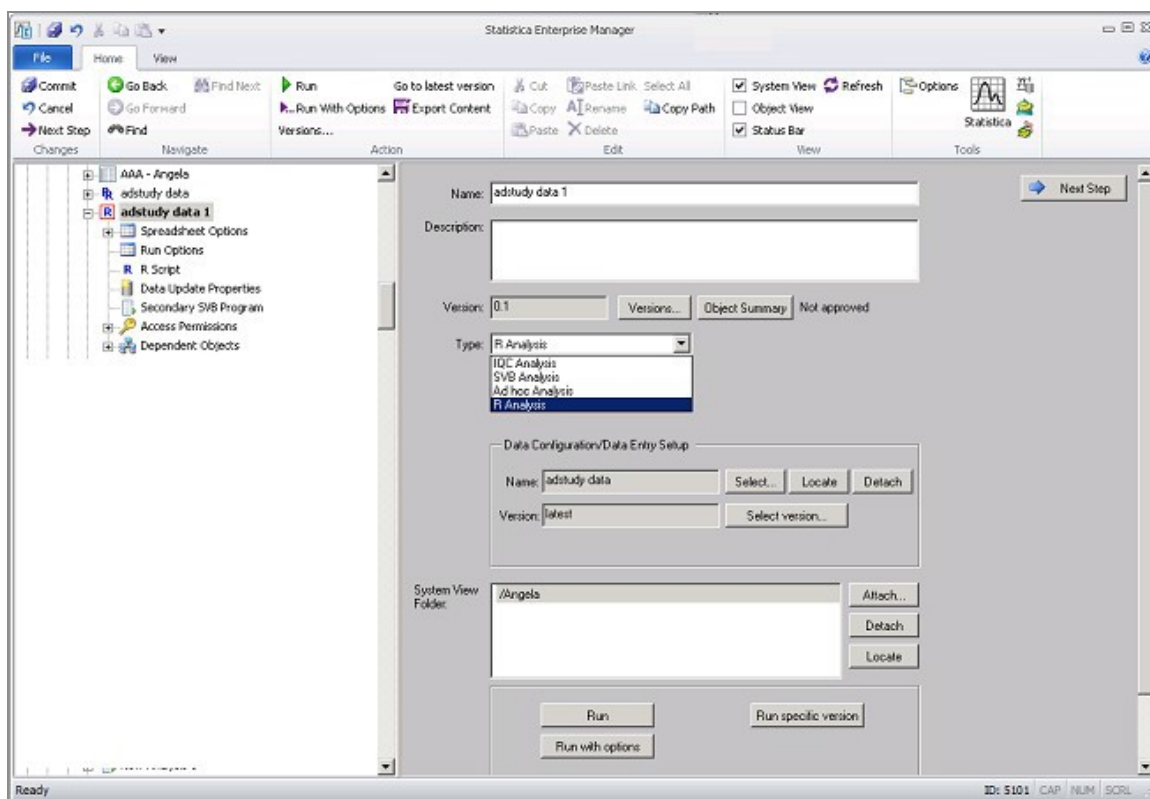
For more information, see the [Help](#) page.

For more information, see the [R Node help page](#), [General Scripting Nodes](#) and [General Code nodes](#), [Statistica Workspace Toolbar](#), and the [Help Page in Enterprise Manager](#).

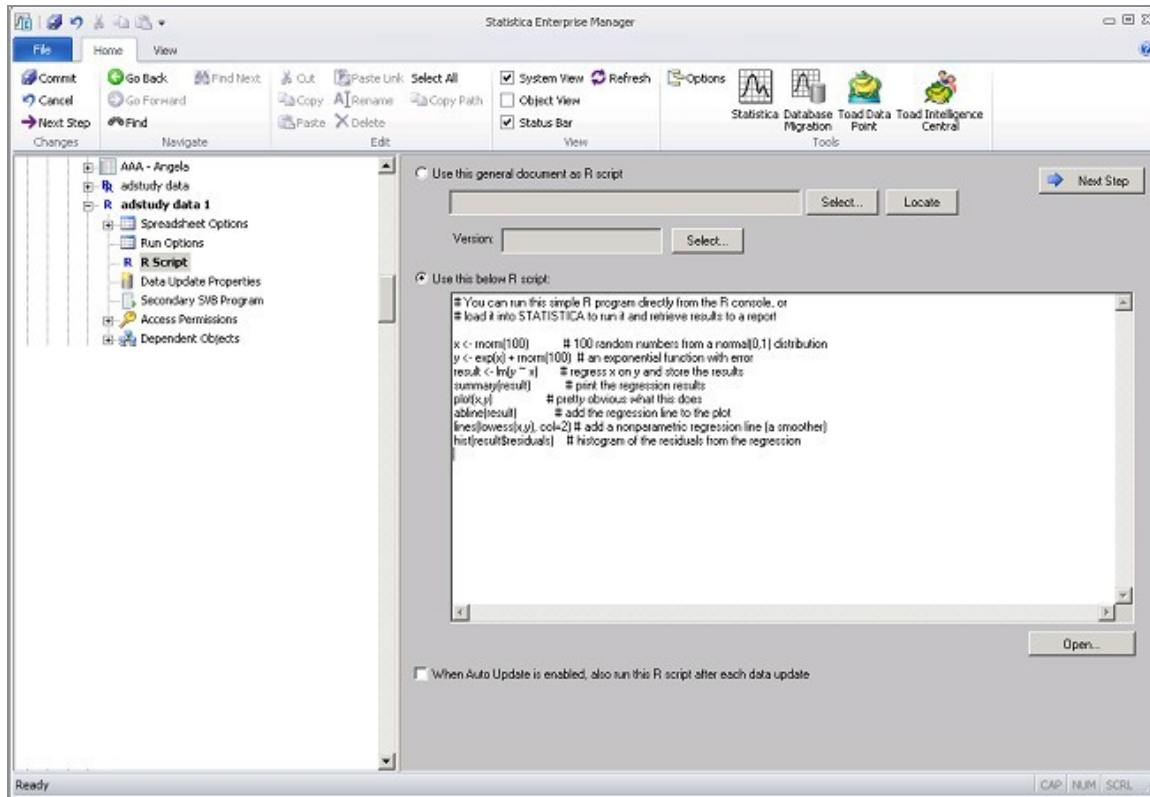
Creating R-based Analysis Configurations

The Statistica system implements standard control charting and SPC analyses out-of-the-box; non-trivial analysis templates are usually created using Statistica Visual Basic (e.g., by using Statistica Macro-recording capabilities). In addition, Statistica Enterprise and Statistica Server now recognize R scripts, and can store and process them in the same way as SVB scripts.

When creating a new Analysis Configuration in Statistica Enterprise, select the R Analysis option:



Then type, paste in, or load the respective R script.



Depending on the configuration of the Statistica Enterprise system, this reusable analysis template is now available to the respective users of both Statistica and Statistica Server environments, and the results of this template can be combined into standard reports.

Moreover, as with all items saved in Statistica Enterprise, administrators can restrict viewing or editing of these R scripts or workflows that use them. All scripts and analyses are versioned with complete audit trails to support FDA 21 CFR Part 11 requirements.

Calling R Scripts from SVB Analysis Configurations

Similarly, R functionality can also be utilized from inside SVB analysis templates, which can retrieve the results from R for further processing or display. In this case, R scripts that are loaded for the respective templated analyses in Statistica Enterprise can be placed into a secure folder repository from which they can be loaded or they can be embedded into the SVB code (assign R script text to the SVB macro's Code property). Both approaches allow for the R Code to be version-controlled and for it to only be available to those with the appropriate access permissions.

Final Comments, and Some Caveats

The features provided in Statistica for integration with R are quite flexible and make the thousands of highly specialized R functions and features available to all Statistica solutions. However, users who are planning to utilize these features are advised to consider the following possible issues, in particular, certain system limitations of the R platform.

Error Handling

Most of the error conditions generated within the R environment (example, syntax and runtime errors caused by an R script) or by the integration support libraries (example, broken R installation or missing components) are intercepted and handled by Statistica. Developers can use error handling facilities available in either environment, for example On Error handlers in SVB macros calling R scripts.

However, occasionally R programs can crash or hang. In case the program hangs, program control does not return to Statistica. Therefore, careful validation of the respective R scripts is crucial for enterprise-level deployment of R analysis templates.

Strengths and Limitations

A word of caution regarding the quality of R algorithms: R comes without warranty or guarantees. In practice, many of the algorithms available in R are the result of diligent work over many years by one or a few individuals who are experts in the respective methodology or domain. However, this does not mean that the software was created following rigorous software development lifecycle methodology, or stringent standard operating procedures for software requirements gathering, design, implementation, and testing. Therefore, in order to build a mission-critical or validated application around a component that depends on R, it is absolutely critical that you carefully validate all results for the use cases to which the software is to be applied.

A word of caution regarding scalability, large data sets, etc: Another caveat regarding R that needs to be considered before building solutions around R concerns its basic architecture. Unlike Statistica, data in R must be (in practically all cases) resident in the

computer's memory. This restriction, in combination with hardware-level and operating system-level memory limitations, may or may not pose an obstacle for any one individual user, but will need to be considered carefully when building R-based server applications accessible to multiple users.

Spotfire Documentation and Support Services

For information about this product, you can read the documentation, contact Spotfire Support, and join Spotfire Community.

How to Access Spotfire Documentation

Documentation for Spotfire products is available on the [Product Documentation website](#), mainly in HTML and PDF formats.

The [Product Documentation website](#) is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on [Spotfire Statistica® Product Documentation](#) page and [Spotfire Statistica®](#).

How to Contact Support for Spotfire Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our [product Support website](#).
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the [product Support website](#). If you do not have a username, you can request one by clicking **Register** on the website.

How to Join Spotfire Community

Spotfire Community is the official channel for Spotfire customers, partners, and employee subject matter experts to share and access their collective experience. Spotfire Community offers access to Q&A forums, product wikis, and best practices. It also offers access to

extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from Spotfire products. In addition, users can submit and vote on feature requests from within the [Spotfire Ideas Portal](#). For a free registration, go to [Spotfire Community](#).

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

Statistica, Spotfire, Process Tree Viewer, Process Data Explorer, Predictive Claims Flow, Live Score, Electronic Statistics Textbook, and Data Health Check, are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

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

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file

for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.tibco.com/patents>.

Copyright © 1995-2024. Cloud Software Group, Inc. All Rights Reserved.