

**TIBCO Statistica™**

# **Statistica - R Integration: Features and Options**

*Software Release 13.4  
May 2018*

**Two-Second Advantage®**



## Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that 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 TIBCO Software Inc.

TIBCO, Better Decisioning, Data Health Check, Data Science, Decisioning Platform, Electronic Statistics Textbook, Information Bus, Live Score, Making the World Productive, Messaging Appliance, Predictive Claims Flow, Process Data Explorer, Process Tree Viewer, Rendezvous, Statistica, Statsoft, Statsoft Iberica, The Power of Now, TIB, TIBCO Rendezvous, and Two-Second Advantage are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and other countries.

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.

THIS 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 THIS SOFTWARE VERSION 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. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

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 "READ ME" FILES.

Copyright © 2017-2018 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

## Table of Contents

Introduction .....	5
Overview and Summary .....	7
Basic Architecture and Features of R Support in Statistica.....	9
COM Interface to R Environment.....	10
R Integration Support Macros (R.svb / R.r) .....	11
R Scripts as Native Statistica Macros .....	11
Retrieving Results.....	13
Statistica Capabilities .....	13
R Language Extensions: Passing Data to R and Retrieving Results .....	15
ActiveDataSet.....	15
Spreadsheet(FilePathOrName) .....	16
RouteOutput(x, name, description) .....	17
Uses(package, [lib], [repos], [quiet], [attachImports]).....	19
Executing R Scripts from within Statistica Visual Basic .....	20
R Scripts are Statistica Macros .....	20
Passing Parameters to R Scripts: Collection Object .....	21
Managing Script Results .....	24
More Examples.....	25
R Support in Statistica Server .....	26
Statistica Server is a Powerful R Server.....	27
Off-Loading R Scripts from Statistica to Statistica Server .....	27
Integrating R with Statistica Server.....	29
Creating R-based Analysis Configurations.....	29
Calling R Scripts from SVB Analysis Configurations.....	30
Final Comments, and Some Caveats .....	32
Error Handling .....	32

Strengths and Limitations .....	32
---------------------------------	----

## Introduction

R is a programming language and environment for statistical computing ([www.r-project.org](http://www.r-project.org)). 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 ( $\geq 2.1$ ) 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,

<https://support.tibco.com/s/article/Manually-enabling-Statistica-R-Integration-for-Statistica-v13-3-0>, for manual installation instructions for the COMadaptR support library.

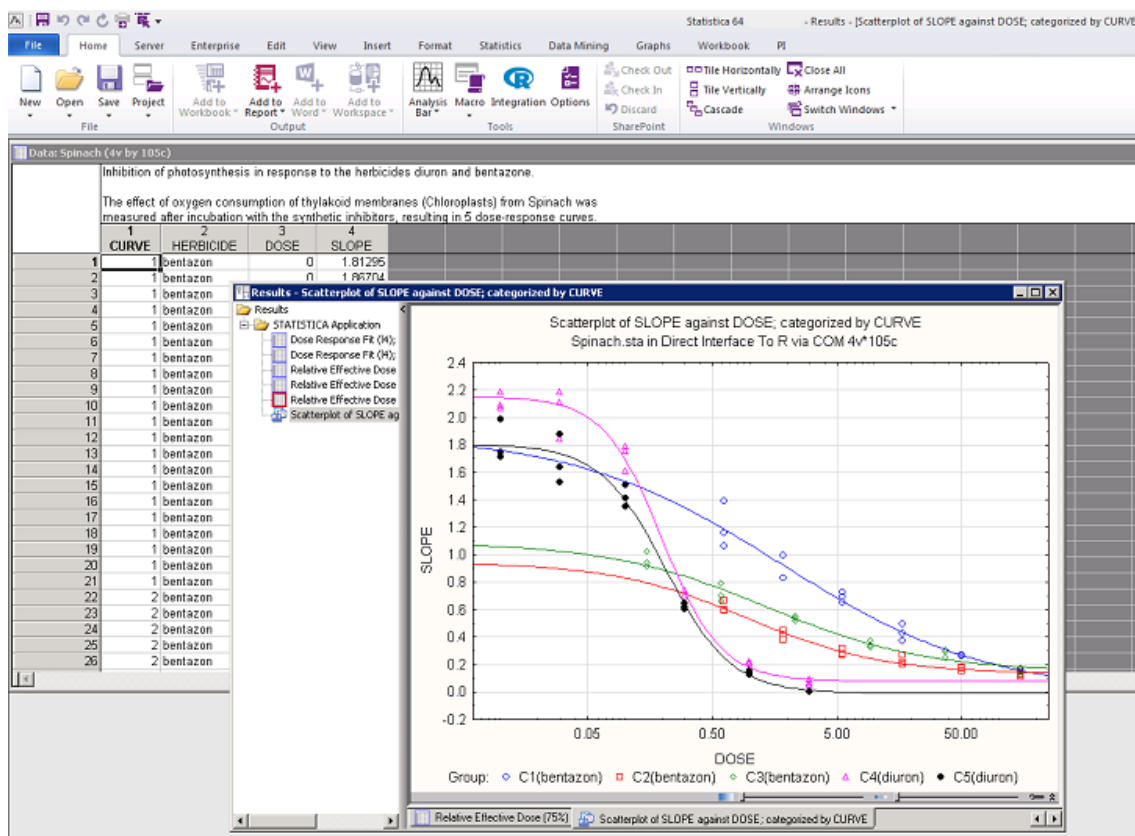
You can verify that the COMadaptR library is installed correctly (and that all its dependencies are satisfied) by running one of the R examples that accompany Statistica. Once all the required third-party components are installed, you should be able to open and execute R scripts within the Statistica 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/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

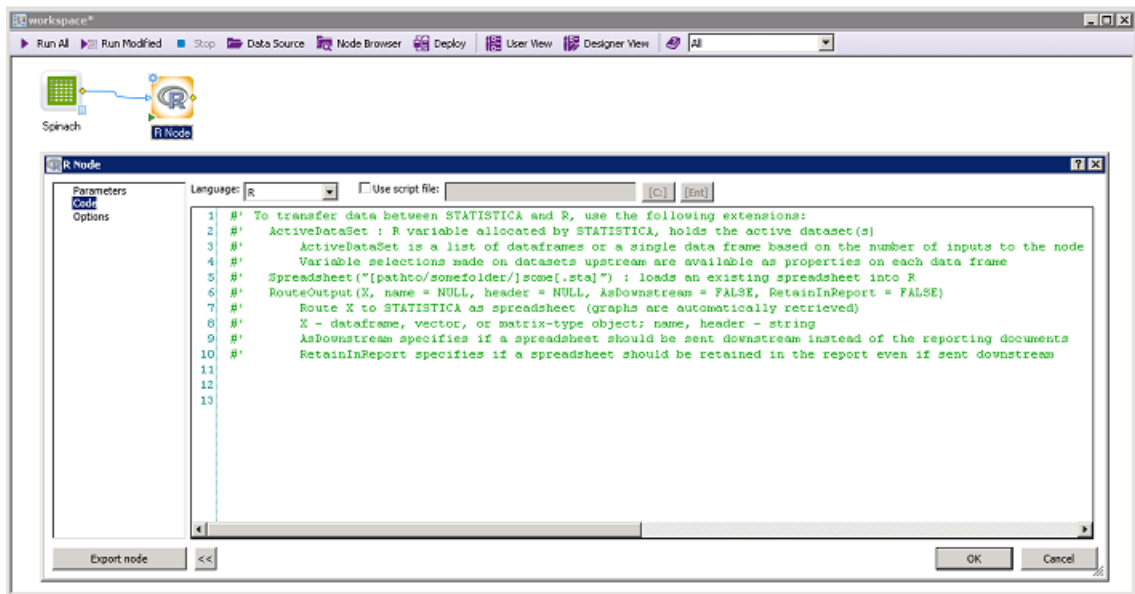
The goal of this document is to provide a detailed description of features that make the diversity and power of R fully available. These features enable users to combine the capabilities of both Statistica and R platforms.

- 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 (“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 to:

- 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.

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 a R COM Server; the other is used for callbacks from R to Statistica.

Once R is installed, Statistica can automatically detect it. The next time you run Statistica, a dialog box asks if R Integration should be enabled. Click Yes, and Statistica will automatically install 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 will be required since COMadaptR is downloaded from <https://tran.tibco.com/statistica/>.

Read the KB article, <https://support.tibco.com/s/article/Manually-enabling-Statistica-R-Integration-for-Statistica-v13-3-0>, for manual installation instructions for the COMadaptR support library.

Note: The COMadaptR library is independent from Statistica and will remain 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 above 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 / 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 = IL.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 press F5). This action will execute 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.

**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 the Statistica Document Management System; shared among other users in a web-based, client-server Statistica Server environment, etc.

## 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 will only be 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/available, the R script that uses it will fail (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 through 5)
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

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

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 will look for this file in several locations: first, it will check the folder where the R script itself is located (if it was saved to disk), and then it will check the Examples\Datasets folder for the current Statistica installation. Support code will also append 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.

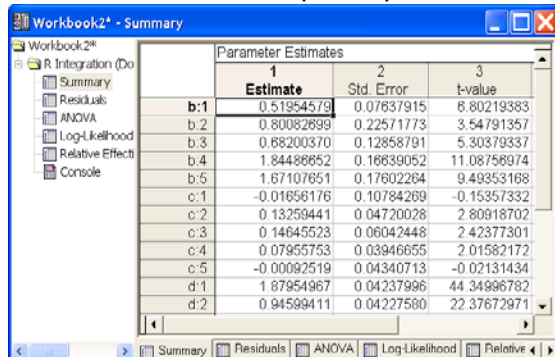
Note that R plots transferred into Statistica as native graphs do not require explicit output routing – all plots generated during a script run are automatically transferred and routed according to Output Manager settings.

**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:

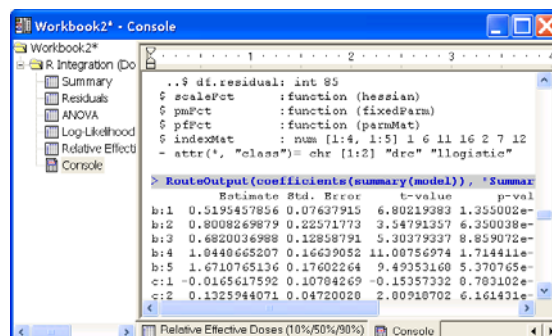
```
# build a statistical model (Dose Response Curves)
model <- multdrc(SLOPE ~ DOSE, CURVE, data = PestSci)
...
str(model) # display the complex structure of this model/results in R console
...
# transfer results to Statistica
RouteOutput(coefficients(summary(model)), "Summary", "Parameter Estimates")
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:



Parameter Estimates

	Estimate	Std. Error	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



```
..$ df.residual: int 85
$ scalePct : function (hessian)
$ pmPct : function (fixedParam)
$ pPct : function (paramMat)
$ indexStat : num [1:4, 1:5] 1 6 11 16 2 7 12
- attr(*, "class")= chr [1:2] "drc" "llogistic"

> RouteOutput(coefficients(summary(model)), "Summary", "Parameter Estimates")
Estimate Std. Error t-value p-val
b:1 0.5195457856 0.07637915 6.80219383 1.355002e-
b:2 0.8008269879 0.22571773 3.54791357 6.350038e-
b:3 0.6820036988 0.12858791 5.30379337 8.859072e-
b:4 1.8448665207 0.16639052 11.08756974 1.714411e-
b:5 1.6710765136 0.17602264 9.49353168 5.370765e-
c:1 -0.0165617592 0.10784269 -0.15357332 0.703102e-
c:2 0.1325944071 0.04720028 2.80918702 6.161431e-
```

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

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

lib – character vector containing path 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.

## 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.

But, in order to provide any non-trivial functionality within an R script in such use cases, you need to be able to “parameterize” that script with user-selected parameters, variables lists, input spreadsheets, etc. Statistica provides a simple and powerful way to pass such parameters to R scripts.

### R Scripts are Statistica Macros

In (almost) all 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
7	EXPERMTL	MALE	AFTER_1	NOT_PAID	2.24	15	2	2

## Passing Parameters to R Scripts: Collection Object

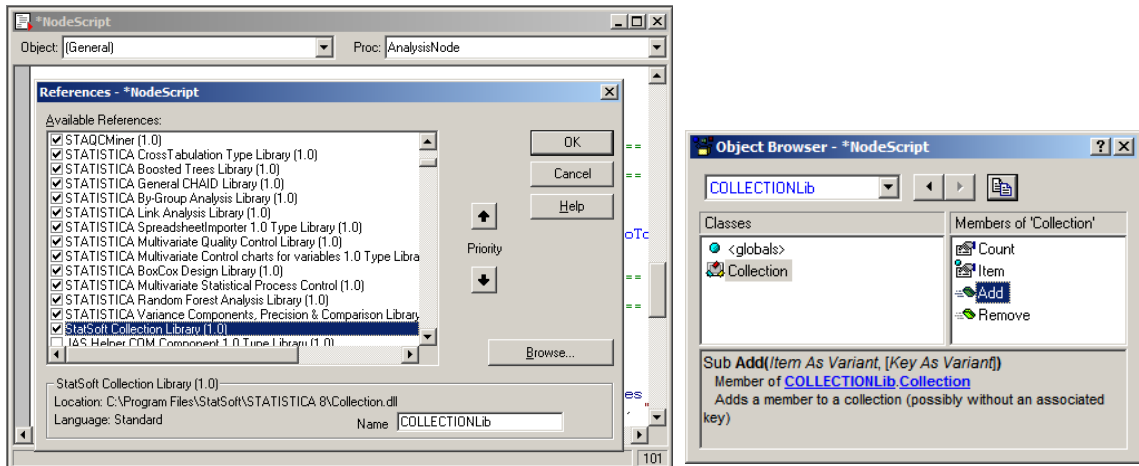
Collection is a COM library introduced with Statistica Version 8 MR3 that implements generic Collection objects. Such objects enable users to store keyword-tagged Variant values. Variants are COM structures that can store one of many supported data types, including arrays and references to other COM objects, e.g., 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 will transfer 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/constants.

In order to use Collection objects, SVB scripts must include a reference to the TIBCO 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:

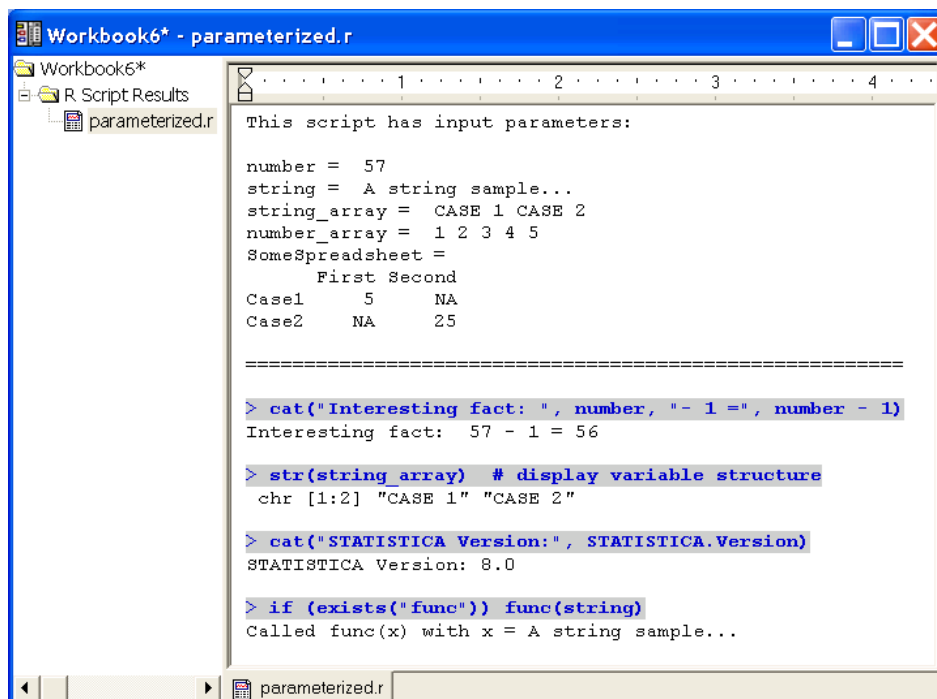
```
Dim param As New Collection
param("number") = 57
param("string") = "A string sample..."
```

Once 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)
```



## 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

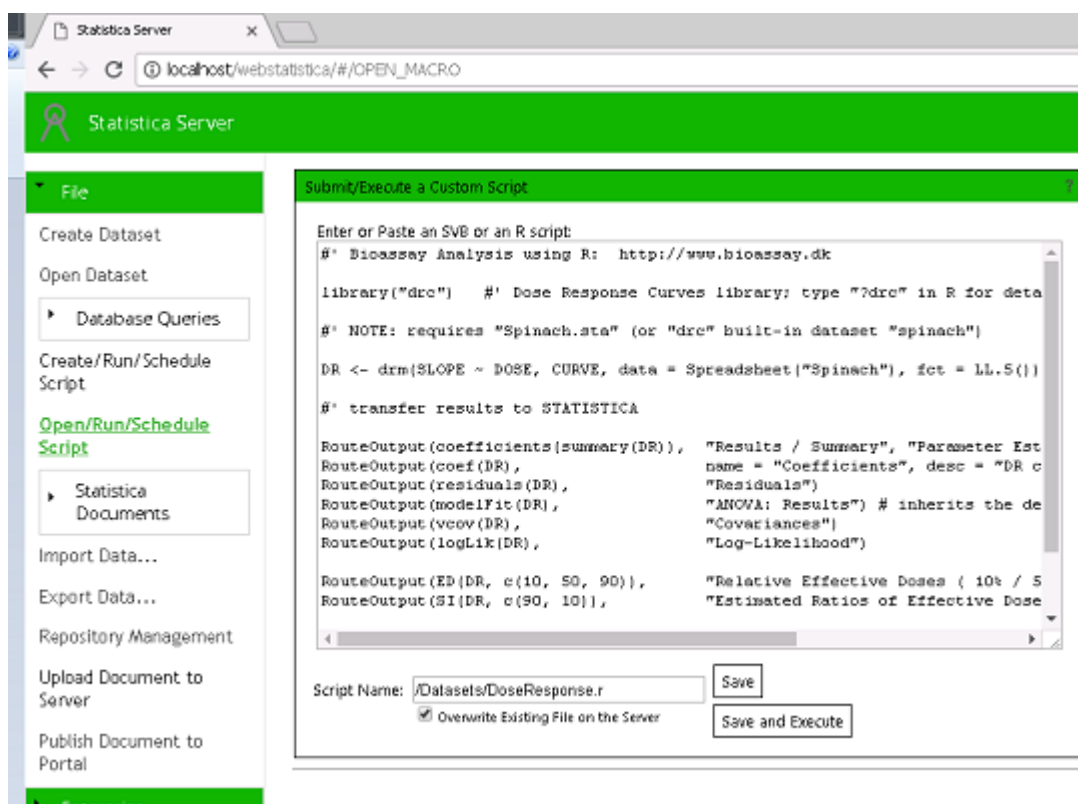
At this point it is demonstrated that all the functional components required to build custom applications within the Statistica platform can take advantage of the specialized functionality available in R.

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 into 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.

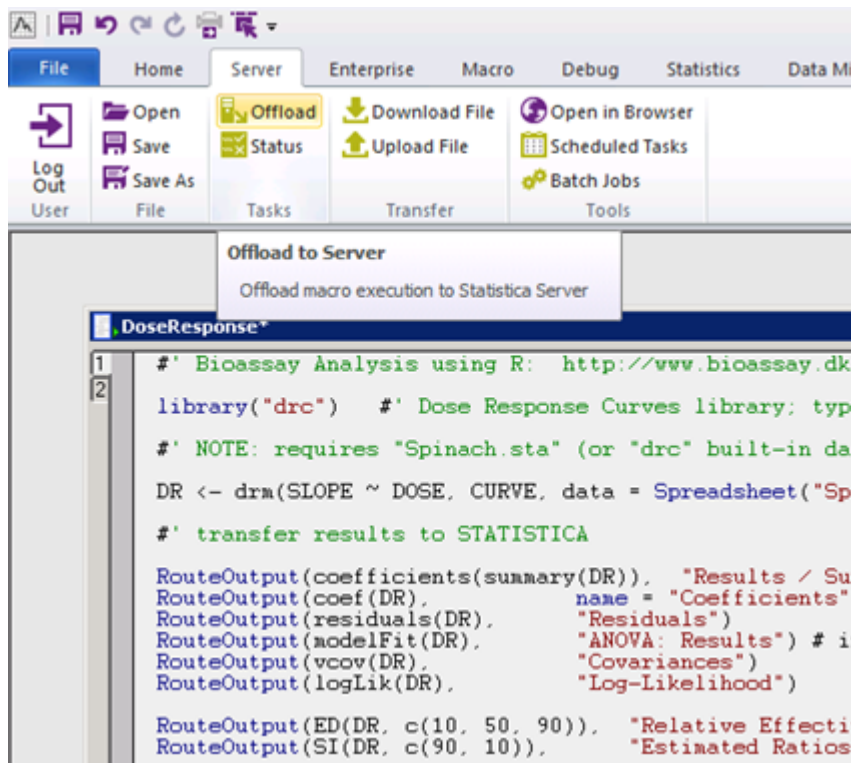
R scripts are handled by Statistica Server in much the same way as standard SVB macros. The third-party components that the Statistica platform relies upon to provide its R runtime environment (such as R library and R COM Server library) are also well suited for handling multiple simultaneous R sessions.

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 (e.g., 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 via the Server → Task Status... dialog box.

The results can be retrieved from the server via 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.

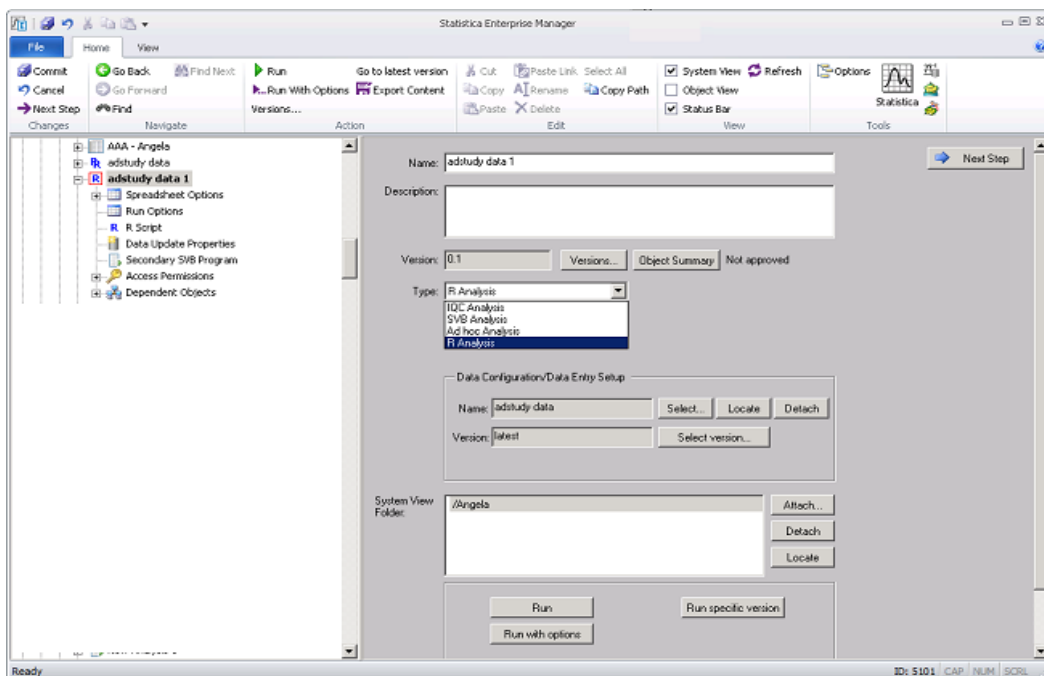
## Integrating R with Statistica Server

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 templated analyses for validated applications (e.g., 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.

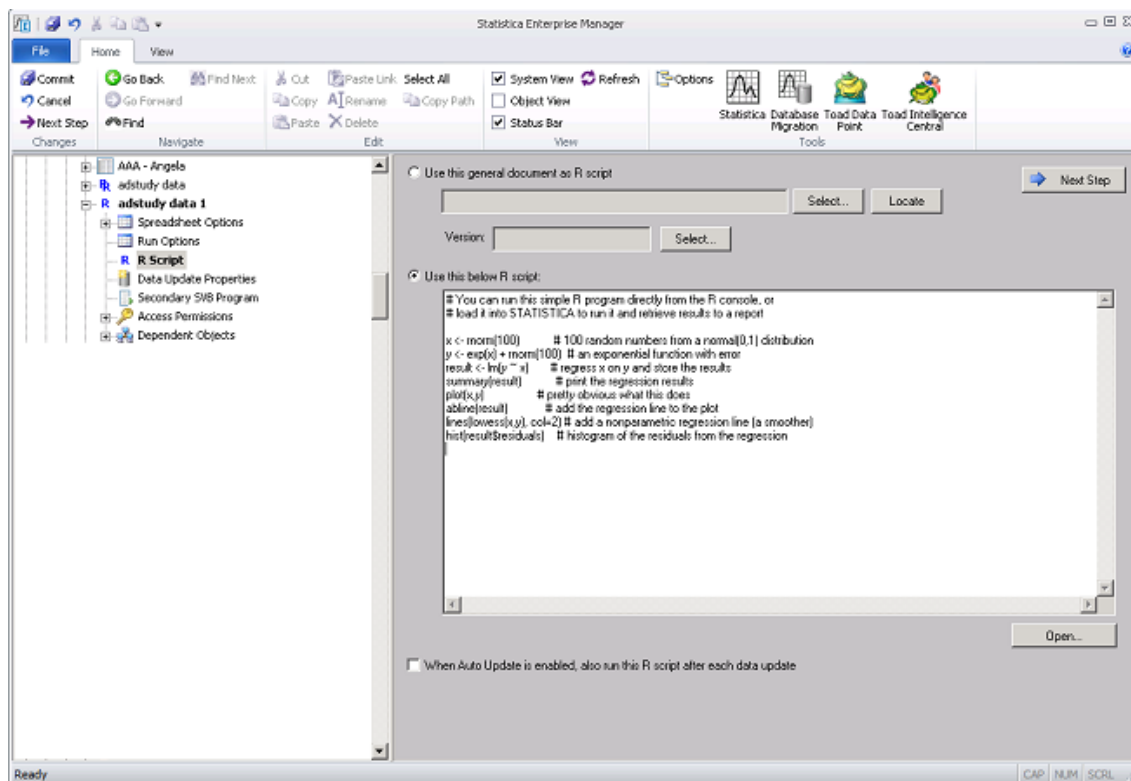
## Creating R-based Analysis Configurations

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, if the respective Statistica Enterprise installation is integrated with the Statistica Document Management System, these R scripts can be “locked down,” i.e., 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 either be placed into a secure folder repository from which they can be loaded (and where they can be managed via

Statistica Document Management System integrated with Statistica Enterprise), or they can be embedded into the SVB code (assign R script text to the SVB macro's Code property).

## 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 (e.g., syntax and runtime errors caused by an R script) or by the integration support libraries (e.g., 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 the latter case, 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 (most) 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.