Smarten
Advanced Data Discovery
Powered by ElegantJ BI

# Working with R Integration

Business Intelligence & Advanced Data Discovery

Document Version: 2.0
Product Version: 5.0 and above

| Document Information | |
|---|---|
| Document ID | Smarten-Working-with-R-Integration |
| Document Version | 2.0 |
| Product Version | 5.0 and above |
| Date | 1-October-2018 |
| Recipient | NA |
| Author | EMTPL |



**Statement of Confidentiality, Disclaimer and Copyright**

This document contains information that is proprietary and confidential to EMTPL, which shall not be disclosed, transmitted, or duplicated, used in whole or in part for any purpose other than its intended purpose. Any use or disclosure in whole or in part of this information without the express written permission of EMTPL is prohibited.

Any other company and product names mentioned are used for identification purpose only, may be trademarks of their respective owners and are duly acknowledged.

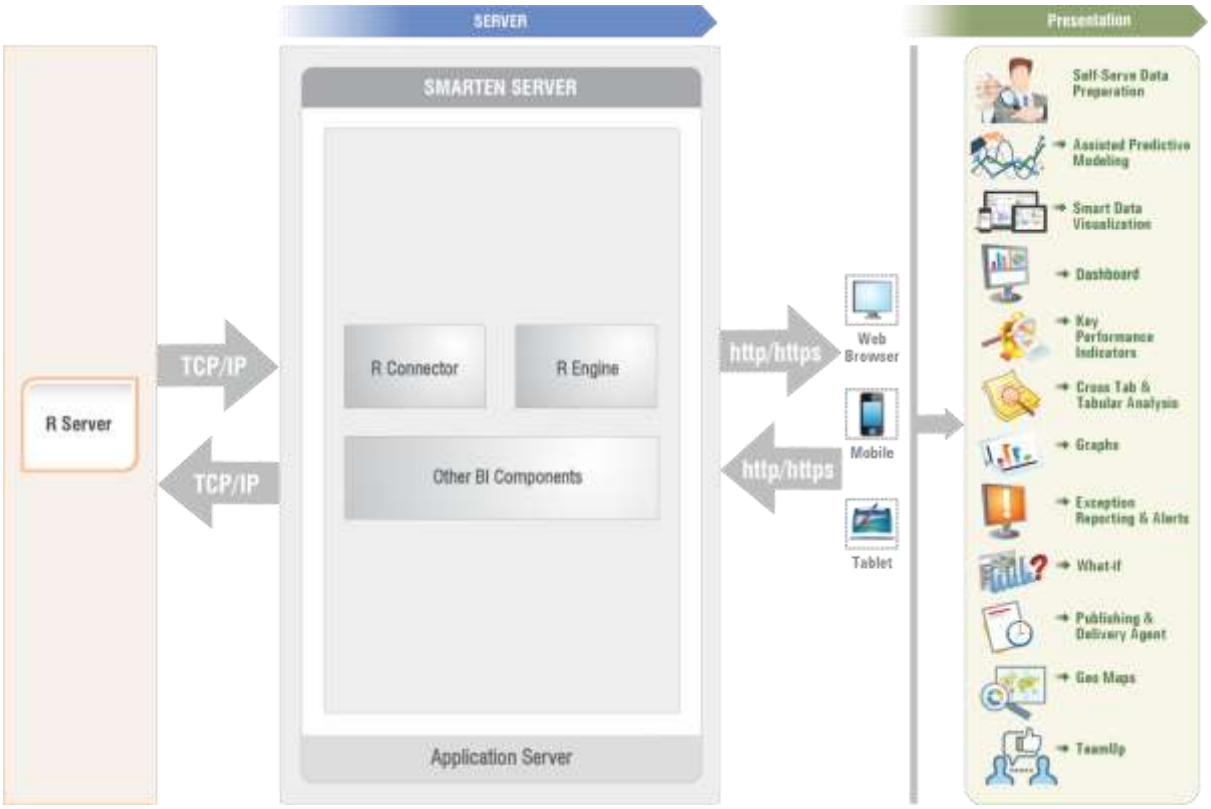
**Disclaimer**

This document is intended to support administrators, technology managers or developers using and implementing Smarten. The business needs of each organization will vary and this document is expected to provide guidelines and not rules for making any decisions related to Smarten. The overall performance of Smarten depends on many factors, including but not limited to hardware configuration and network throughput.

# Contents

# 1 Introduction

Smarten integrates seamlessly with R, the popular open source programming language and software environment for statistical computing. This feature allows users to integrate any R script with Smarten by configuring input and output variables of both R script and the Smarten suite. Users have access to a full suite of self-serve visualization and analytical tools to present and share results from R script in a report, on a dashboard, with interactive graphs, in ad hoc reporting, cross-tab and tabular reports, and key performance indicators (KPIs). Data scientists and business analysts do not need extensive training or knowledge to use the Smarten advanced data discovery suite or to leverage the R script integration.




**SMARTEN—R INTEGRATION SYSTEM ARCHITECTURE**

R is an open source programming language and software environment supported by the R Foundation for Statistical Computing. It is widely used among statisticians and data miners for developing statistical and data analysis.

# 2 R Integration

In Smarten, the process of R integration starts with creating an R script profile by configuring input and output variables for an R script. To integrate an R script with the Smarten suite, input variables of an R script are mapped with Smarten data that can either come from a cube or dataset, entered manually by the user, or a combination of both. Similarly, the output variables of an R script are mapped with Smarten cube and dataset. Users have the option to either upload an R script file or paste an R script while creating the profile.

Smarten allows an R script profile to be used as one of the data sources to create cubes or datasets, which are created by associating an R script profile as a data source and associating input and output variables of an R script with the Smarten cube or dataset. Smarten supports both real-time and cache architecture for cubes and datasets created from R script output. Cache cubes and datasets store indexed, preaggregated data along with metadata in the cubes and datasets, whereas real-time cubes and datasets store only metadata information and do not store any data in the cubes and datasets.

R cubes and datasets created from R script output can be accessed by users from front-end objects, such as dashboards, crosstab, tabular, graphs, GeoMap, SmartenView, SmartenInsight and KPI, and the resultant data can be analyzed as per requirement.

R Integration can be achieved within Smarten by the following steps:

- R Server Configuration
- Create R Script Profile
- Create Smarten Cube or dataset with R Script Profile as data source
- Access R cubes or datasets from Smarten front-end tools

## 2.1 R Server Configuration

Configuring the R server is a prerequisite for R integration. The user needs to provide the host name and port number of the R server on which the R script has to be executed. Smarten allows the user to test the connection to the R server before saving it.

For more details on how to install the R server, please refer to https://www.r-statistics.com/tag/r-community/

## 2.2 Create an R Script Profile

An R script profile contains important information related to an R script, which is then used as a data source to create an R cube or dataset within Smarten. The user can enter required parameters, such as names and types of the input and output variables of the R Script. Data for input variables can be either from an existing Smarten cube or dataset or entered manually by the user or a combination of both. Output variables can be defined depending on the R script that is being configured.

Smarten provides a way to create an R script profile either by uploading an R script from its saved location or by pasting the R script itself.

> Note:
> The user should have the R script ready before creating the profile.

Shown below are the steps involved in creating an R script profile:

**Step 1:** Specify R server connection details
**Step 2:** Select an existing R script from its location, or paste the R script.
**Step 3:** Define input variables and their types for the R script. Three types of input are available.

| Input type | Description |
|---|---|
| Text | This type of input allows users to enter data manually to be considered as input for the R script variable. |
| Single column | This type of input allows users to map only one column of the Smarten cube or dataset with input variable of the R script. Data belonging to the selected column will be considered as input for the R script variable. |
| Multi column | This type of input allows users to map more than one column of a Smarten cube or dataset with input variable of the R script. Data belonging to all the selected columns will be considered as input for the R script variable. |

**Step 4:** Define Query parameters, if required, for the R script.

These types of variables allow a user to provide query parameters to the R script from Smarten front-end during run-time. The R script should be written to handle such Query parameters in run-time and return the results back to Smarten.

**Step 5:** Define Output variables of the R script along with their type. Two types of output are available

| Output type | Description |
|---|---|
| Table | Output of the R script is saved in tabular form in its output variable, which allows adding the output as columns or rows in the Smarten cube or dataset. For example, based on the demographic information and actual response received from recipients, the output of a predicted response in the form of "Yes" or "No" can be appended as a column. Similarly, the forecast sales values for the next 3 months can be appended as rows to the original data table having sales data for the past 3 months. |

## 2.3 Create R Cube or dataset with the R Script Profile as a Data Source

Once an R script profile has been created, the next step is to create an R cube or dataset by associating the R script profile as its data source. Smarten provides easy-to-use steps to create an R cube or dataset.

Shown below are the steps for creating an R cube profile:

**Step 1:** Select the R script profile as a data source. In case of R cubes, select type of cube – Cache or Real-time.

**Step 2:** Map input variables of the R script with Smarten.
If input variables are of Single column or Multiple columns input type to the R script:

- Identify and select an existing Smarten cube or dataset as the input cube or dataset for the R script.
- If the input variable configured through the R script is a "Single column" type, the user can map only one column of the Smarten input cube or dataset with the R script input variable. Data belonging to the mapped column will be considered as input data for the R script.
- If the input variable configured through the R script is a "Multi column" type, the user can map one or more columns of the Smarten input cube or dataset with the R script input variable. Data belonging to all mapped columns will be considered as input data for the R script.
- If the input variable configured through the R script is a "Text" type, the user can enter data manually. The data entered manually by the user will be considered as input data for the R script.

> Note:
> All the input variables entered by the user are passed as text data type variables to the R script, and any further processing required, including data type conversion or transformation, should be done within the R script.

For example, to identify the relationship between a customer's age and spending capacity, the user needs to enter values of age as "10, 20, 30, 40" in an input variable, namely "Age" and spending capacity as "300, 700, 900, 1300" in another input variable, namely "Purchase." The R script would then transform these comma separated text values into individual numbers and sequence each value of "Age" with its corresponding value of "Purchase" to create an input table with columns of "Age" and "Purchase" as follows:

| Age | Purchase |
|---|---|
| 10 | 300 |
| 20 | 700 |
| 30 | 900 |
| 40 | 1300 |

Please refer to example 2.5.1in this document for more details.

Similarly, when multiple values are required in a single variable, the user needs to enter all the values separated by a comma (or other separator character). These comma separated values are then passed to the R script, which, in turn, creates a single column table with each value of the variable as a row. For example, to find the average of sales value for a week, let's say the user enters the values of sales as "5050," "2000," "1300," "3500," 4932," "7921," "3612" in an input variable, namely "Sales Value." The R script would then create an input table with "Sales Value" as a column and its values as rows as follows:

| Sales Value |
|---|
| 5050 |
| 2000 |
| 1300 |
| 3500 |
| 4932 |
| 7921 |
| 3612 |

**Step 3:** Enter the details to configure Query parameters of the R script.

Enter the data to configure Query parameters. These are the default mandatory values with which the R script will be queried and a R cube or dataset will be created. Please refer to example 2.5.4 in this document for more details.

> Note:
> Query parameters can be configured while generating R cubes or datasets.

> Initially, a R cube or dataset is generated with default values of Query parameters entered by the user while creating the R cube or dataset. So, a real-time object generated from a real-time R cube or dataset will display the data fetched by that R cube or dataset with default Query parameters applied. Smarten allows a user to regenerate the real-time object with different output by entering different values of Query parameters from the object itself. The user can change the values of Query parameters in run-time through the "R script parameters" button in the toolbar of the real-time object. As a result, the R script will be re-executed with new values of Query parameters, and the real-time R cube or dataset will be rebuilt for the new output, which will, in turn, be instantly reflected in the real-time object.

**Step 4:** Map the output variable of the R script with Smarten.

In case the R script returns more than one output, only one output can be mapped with the Smarten cube or dataset at a time. Other cubes or datasets should be generated to handle different outputs from the same R script.

For example, let us say that an R script using the regression statistics method provides two outputs as follows:

a) it predicts the loan amount on the basis of demographics and other attributes of the applicant.

b) it indicates how accurate the model is in terms of predicting the eligible loan amount.

Please refer to example 2.5.3 in this document for more details.

Three types of output are available:

| **Output type** | **Description** |
|---|---|
| Output data as an individual table | Output of the R script is saved as a table in the Smarten cube or dataset.<br>For example, an eCommerce company wants to identify the relationship between a customer's age and spending capacity. So, the output here would be the coefficient correlation between the data in a table with Age and Purchase as columns. Please refer to example 2.5.1 in this document for more details. |
| Append output data as a column | Output of the R script is added as a column of the Smarten cube or dataset.<br>For example, a financial services company runs an email marketing campaign and wants to classify the prospects into likely respondents and unlikely respondents. So, based on the demographic information and actual response received, the output here would be the predicted response in the form of "Yes" or "No," which can be appended as a column named "Likely to Respond?"; please refer to example 2.5.2 in the document for more details. |

| | |
|---|---|
| Append output data as a row | Output of the R script is added as a row of the Smarten cube or dataset.<br>For example, an eCommerce company with its past sales data may want to forecast its sales in the future. In this case, let's say the past sales data for 3 years is available in columns "Date" and "Sales value." So, the forecast sales values for the next 10 days can be appended as rows to the original data table. Please refer to example 2.5.5 in this document for more details. |

## 2.4 Access R Cubes or datasets from Smarten Front-end Tools

An R cube/dataset is just like any other cube or dataset created from a database or a CSV file that can be accessed by users from front-end objects, such as dashboards, crosstab, tabular, graphs, GeoMap, and KPI.

R Integration supports both cache and real-time cube and dataset architecture. Objects created by accessing cache cubes and datasets display information that is already stored within the cubes and datasets until such cubes and datasets are rebuilt to fetch the latest data, whereas the real-time R cubes and datasets extract the data as and when required and retrieve the latest data by executing the respective R script every time an object is accessed through them.

> Note:
> Analytic functions available to users depend on the type of cube or dataset used in a particular front-end object.
> Please refer to the
> "Smarten-Feature Matrix Based on Data Sources" document
> for a list of functions available while using Cache and Real-time cubes or datasets.

## 2.5 Access R Cubes or datasets from Smarten Front-end Tools

Let us understand the process of R Integration for three different scenarios, using sample R scripts.

### 2.5.1 Show all data value

This property allows users to display or hide all the data values in the visualization.

**Scenario:**
An eCommerce company wants to identify the relationship between a customer's age and spending capacity. If it turns out that the purchase amount is directly proportional to the customer's age, then the marketing activities can be designed for each age group in a different manner, for example, targeting a higher age group more to drive revenue.

**Sample R Script:**

```
x <- as.list(strsplit(xVals,","))
x1 <- as.numeric(unlist(x))
y <- as.list(strsplit(yVals,","))
y1 <- as.numeric(unlist(y))
#data_1 <- data.frame(xVals)
```

```
data_1 <- data.frame(x1,y1)

if(length(data_1) == 2)
{
 Value <- cor(data_1[[1]],data_1[[2]], method="pearson")
 Title <- c("Correlation Value")
 corr <- data.frame(Title,Value)
} else {
#Correlation between multiple variables
  value <- data.frame(cor(data_1, method = "pearson"))
  n <- names(value)
  corr <- rbind.data.frame(value)
  corr <- cbind.data.frame(n,corr)
  col <- "Title"
  for(i in 1:length(n))
  {
    c <- n[i]
    col <- c(col,c)
  }
  colnames(corr) <- col
}
```

● *Input variable* ● *Output variable*

**What it does:**

This sample R script is supposed to calculate the correlation coefficient between data entered manually by the user. The data for input variables of this R script is entered manually by the user in the form of text. Since the output of this R script is numeric, it first converts both the texts entered by the user into numeric and then calculates the correlation coefficient between them. This R script has two input variables, namely “xVals” and “yVals” and one output variable, namely “corr” that saves the value of correlation coefficient under the title “Correlation Value.”

So the input data for this R script would be a customer’s age and purchase amount by the customer, and the output would be correlation coefficient indicating the strength of the relationship between age and purchase amount.

**Method:**

Correlation is a statistical measure that indicates the extent to which two variables fluctuate together. This R script uses the Karl Pearson method to calculate the correlation coefficient between the data.

**Input and Output variables:**

The following table lists input and output variables configured for the sample R script and Smarten. Each input variable is mapped with data entered manually by the user to get the desired output in the form of a table.

| R script | | | | R cube or dataset | | | |
|---|---|---|---|---|---|---|---|
| Input | | Output | | Input | | Output | |
| Name | Type | Name | Type | Name | Value | Name | Type |
| xVals | Text | corr | Table | Age | 100,20,40,50,60,70 | Correlation Coefficient | Output data as individual table |
| yVals | Text | | | Purchase | 300,700,900,1000,1300, 1600 | | |

- **Create an R script profile:**

While creating an R script profile, the user needs to identify and define the input and output variables within the R script, which are then mapped with Smarten.

Shown below are the input variables being defined for this sample R script.


**SCRIPT PROFILE—MANUAL INPUT VARIABLES FOR AN R SCRIPT**

In this example, two input variables, namely “xVals” and “yVals” have been defined with input type as “Text”.

There are no Query parameters required for this R script.

Shown below is the output variable being defined for this sample R script.


**SCRIPT PROFILE—OUTPUT VARIABLE FOR AN R SCRIPT**

The output variable for this R script is “corr”, which contains the resultant data in the form of a table.

- **Create an R cube with the R script profile as a data source:**

While creating the R cube from the R script profile, the user needs to enter values for both variables.


**R CUBE—MANUAL DATA FOR INPUT VARIABLES FOR AN R SCRIPT**

In this example, the values of “Age” and “Purchase” have been entered so that the correlation coefficient can be calculated between these two data sets.

So, the data entered for “Age” and “Purchase” will be considered as input for variable “xVals” and “yVals” as follows:

| Age | Purchase |
|---|---|
| 10 | 300 |
| 20 | 700 |
| 40 | 900 |
| 50 | 1000 |
| 60 | 1300 |
| 70 | 1600 |

**INPUT DATA FOR AN R SCRIPT**



- **Access R cubes from Smarten front-end tools:**

The R Cube created can now be accessed by users from front-end objects, such as dashboards, crosstab, tabular, graphs, GeoMap, and KPI, and the resultant data can be analyzed as per requirement. In this example, the output stored in the “corr” variable can be accessed through various Smarten objects. A sample tabular report is shown below.

| Title | Value |
|---|---|
| Correlation Value | 0.97 |

**OUTPUT DATA**

## 2.5.2 Single column or Multi column as Input type and Output Data Appended as a Column in R Cube or Dataset

**Scenario:**
A financial services company runs an email marketing campaign and wants to classify the prospects into likely respondents or unlikely respondents and target them efficiently to drive conversions and increase revenue. For instance, if it turns out that the middle-aged, married, and high income segment is more likely to respond to emails, then this segment can be targeted more to increase the response and conversion rate.

**Sample R Script:**

```
library(lattice)
library(ggplot2)
library(caret)
library(pROC)
library(Hmisc)
library(survival)
library(rms)
library(MASS)
library(arm)
library(pscl)
library(fmsb)

mydata <- data.frame(yVals,xVals)
for (i in 1:ncol(mydata))
{
 if(class(mydata[i]) == "factor")
 {
  mydata[i] <- factor(mydata[i])
 } else
 {
  mydata[i] <- mydata[i]
 }
}
x <- data.frame(xVals)
yNames <- names(yVals)
xNames <- c()
for (i in 1:ncol(x))
{
 if(class(x[[i]]) == "factor")
 {
  f <- paste("factor(", names(x[i]), ")")
  if(is.null(xNames) == TRUE)
  {
   xNames <- paste(xNames, f)
  } else
```

```
   { xNames <- paste(xNames, f, sep = "+")}
  } else


{
   if(is.null(xNames) == TRUE)
   {
    xNames <- paste(xNames, names(x[i]))
   } else
 {
     xNames <- paste(xNames, names(x[i]), sep = "+")
   }
  }
}
formula <- as.formula(paste(yNames, paste(xNames), sep = " ~ "))
logistic_full_model <- glm(formula, data = mydata, family = "binomial")
mydata <- mydata[,-1]
prediction_1 <- predict(logistic_full_model,mydata,type = "response")
PredictedResponseFlag <- round(prediction_1,0)
```

● *Input variable* ● *Output variable*

**What it does:**

This R script takes the input data from the Dimension and Measure columns of an Smarten cube. Based on these inputs, it then predicts the response that may or may not be received by the respondents. This R script has two input variables, namely “xVals” and “yVals” and one output variable, namely “PredictedResponseFlag” that saves the result under the title “Likely to Respond?”

So the input data for xVals would be demographic information, such as Age, Job, Marital Status, Education, Previous Default Status, House Owner Status, Existing Loan, Day of Month, and Days since last email. The input data for yVals would be the actual Response received. Based on these inputs, the output variable would contain the predicted response in the form of “Yes” or “No.”

**Method:**

Classification is a process by which data is split into groups on the basis of preassigned categories or classes available. This R script uses the logistic regression classification method to generate the model.

**Input and Output variables:**

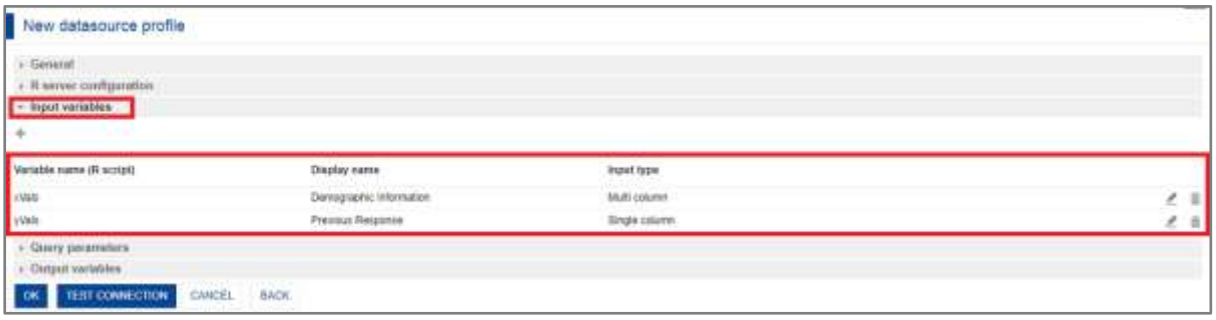
The following table lists the input and output variables configured for the sample R script and Smarten. The R script input variables are mapped with their respective columns of an Smarten cube for input data to get the desired output in the form of a table.

| R script | | | | R cube | | |
|---|---|---|---|---|---|---|
| Input | | Output | | Input | Output | |
| Name | Type | Name | Type | Cube columns | Name | Type |
| xVals | Multi column | PredictedResponseFlag | Table | Age, Job, Marital Status, Education, Previous Default Status, House Owner Status, Existing Loan, Day of Month and Days since last email. | Likely to Respond? | Append output data as column |
| yVals | Single column | | | Response | | |

- **Create an R script profile:**

While creating an R script profile, the user needs to identify and define the input and output variables within the R script, which are then mapped with Smarten.

Shown below are the input variables being defined for this R script.




**R SCRIPT PROFILE—CUBE DATA/MANUAL INPUT VARIABLES FOR AN R SCRIPT**

In this example, two input variables, namely “xVals” and “yVals” have been defined with input type as “Multi column input” and “Single column input” respectively.

There are no Query parameters required for this R script.

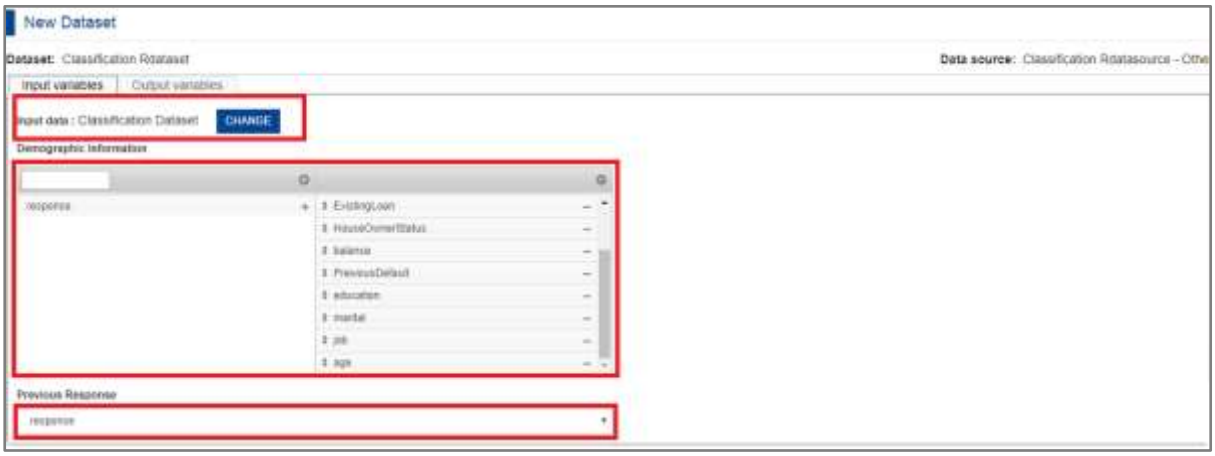
Shown below is the output variable being defined for this R script.


**R SCRIPT PROFILE—OUTPUT VARIABLE FOR AN R SCRIPT**

The output variable for this R script is “PredictedResponseFlag”, which contains the resultant data in the form of a table.

- **Create cube with the R script profile as a data source:**

While creating the R cube from the R script profile, the user needs to map input variables of the R script with Smarten cube columns.




**R CUBE—MAPPING OF CUBE COLUMNS WITH CUBE DATA/MANUAL INPUT VARIABLES FOR AN R SCRIPT**

In this example, the "Classification Dataset" is the input cube for the R script. Since the first input variable "xVals" is of the "Multi column" type, nine columns of input cube have been mapped with this input variable. These columns are "Age," "Job," "Marital Status," "Education," "Previous Default Status," "House Owner Status," "Existing Loan," "Day of Month," and "Days since last email." Since the second input variable "yVals" is a "Single column" type, only one Dimension column of the input cube has been mapped with this input variable. The column is "Response."

So, in this example, data belonging to the selected cube columns will be considered as input for variable "xVals," and data belonging to cube column of "Response" will be considered as input for variable "yVals" as follows:

| ID | Age | Job | Marital Status | Education | Balance | House Owner Status | Existing Loan Status | Day Of Month | Days Since Last Email | Response |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 58 | Management | Married | Graduation | 2143 | Yes | No | 5 | 261 | No |
| 2 | 44 | Technician | Single | Secondary | 29 | Yes | No | 5 | 151 | No |
| 3 | 33 | Entrepreneur | Married | Secondary | 2 | Yes | Yes | 5 | 76 | No |
| 4 | 47 | Blue-Collar | Married | Post Graduation | 1506 | Yes | No | 5 | 92 | No |
| 5 | 33 | Others | Single | Post Graduation | 1 | No | No | 5 | 198 | No |
| 6 | 35 | Management | Married | Graduation | 231 | Yes | No | 5 | 139 | No |
| 7 | 28 | Management | Single | Graduation | 447 | Yes | Yes | 5 | 217 | No |
| 8 | 42 | Entrepreneur | Divorced | Graduation | 2 | Yes | No | 5 | 380 | No |
| 9 | 58 | Retired | Married | Primary | 121 | Yes | No | 5 | 50 | No |
| 10 | 43 | Technician | Single | Secondary | 593 | Yes | No | 5 | 55 | No |
| 11 | 41 | Admin. | Divorced | Secondary | 270 | Yes | No | 5 | 222 | No |
| 12 | 29 | Admin. | Single | Secondary | 390 | Yes | No | 5 | 137 | No |
| 13 | 53 | Technician | Married | Secondary | 6 | Yes | No | 5 | 517 | No |
| 14 | 58 | Technician | Married | Post Graduation | 71 | Yes | No | 5 | 71 | No |
| 15 | 57 | Services | Married | Secondary | 162 | Yes | No | 5 | 174 | No |

**INPUT DATA FOR AN R SCRIPT**

Similarly, the output variable of the R script "PredictedResponseFlag" is mapped with Dimension and Measure columns of the Smarten cube.


**R CUBE—MAPPING OF CUBE COLUMNS WITH OUTPUT VARIABLE FOR AN R SCRIPT**

- **Access R cubes from Smarten front-end tools:**

The R cube created can now be accessed by users from front-end objects, such as dashboards, crosstab, tabular, graphs, GeoMap, and KPI, and the resultant data can be analyzed as per requirement. In this example, the output stored in the "PredictedResponseFlag" variable can be accessed through various Smarten objects. A sample tabular report is as follows:

| ID | Age | Job | Marital Status | Education | Balance | House Owner Status | Existing Loan Status | Day Of Month | Days Since Last Email | Response | Likely To Respond ? |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 58 | Management | Married | Graduation | 2143 | Yes | No | 5 | 261 | No | No |
| 2 | 44 | Technician | Single | Secondary | 29 | Yes | No | 5 | 151 | No | No |
| 3 | 33 | Entrepreneur | Married | Secondary | 2 | Yes | Yes | 5 | 76 | No | No |
| 4 | 47 | Blue-Collar | Married | Post Graduation | 1506 | Yes | No | 5 | 92 | No | No |
| 5 | 33 | Others | Single | Post Graduation | 1 | No | No | 5 | 198 | No | No |
| 6 | 35 | Management | Married | Graduation | 231 | Yes | No | 5 | 139 | No | No |
| 7 | 28 | Management | Single | Graduation | 447 | Yes | Yes | 5 | 217 | No | No |
| 8 | 42 | Entrepreneur | Divorced | Graduation | 2 | Yes | No | 5 | 380 | No | No |
| 9 | 58 | Retired | Married | Primary | 121 | Yes | No | 5 | 50 | No | No |
| 10 | 43 | Technician | Single | Secondary | 593 | Yes | No | 5 | 55 | No | No |
| 11 | 41 | Admin. | Divorced | Secondary | 270 | Yes | No | 5 | 222 | No | No |
| 12 | 29 | Admin. | Single | Secondary | 390 | Yes | No | 5 | 137 | No | No |
| 13 | 53 | Technician | Married | Secondary | 6 | Yes | No | 5 | 517 | No | No |
| 14 | 58 | Technician | Married | Post Graduation | 71 | Yes | No | 5 | 71 | No | No |
| 15 | 57 | Services | Married | Secondary | 162 | Yes | No | 5 | 174 | No | No |

**OUTPUT DATA**

Since the output in the R cube has been configured to be **appended as a column**, the resultant prediction is displayed in a separate column of "Likely to Respond?"

### 2.5.3 More Than One Output

**Scenario:**

A company catering to financial services may want to decide how much of a loan should be given to a new loan applicant based on his/her demographics and other attributes.

**Sample R Script:**

```
memory.limit(size = 15088)


data <- data.frame(yVals,xVals)


x <- data.frame(xVals)


yNames <- names(yVals)

xNames <- c()


for (i in 1:ncol(x))

{

 if(class(x[[i]]) == "factor")

 {
```

```
  f <- paste("factor(", names(x[i]), ")")
  if(is.null(xNames) == TRUE)
  {
   xNames <- paste(xNames, f)
  } else
  { xNames <- paste(xNames, f, sep = "+")}
 } else
 {
  if(is.null(xNames) == TRUE)
  {
   xNames <- paste(xNames, names(x[i]))
  } else
  {

   xNames <- paste(xNames, names(x[i]), sep = "+")
  }
 }
}
formula <- as.formula(paste(yNames, paste(xNames), sep= " ~ "))
model <- lm(formula, data=data)
r <- summary(model)
a <- r$fstatistic
p <- pf(a[1],a[2],a[3],lower.tail = F)
p_value <- p[1]
PredictedLoanAmount <- predict(model)

PredictedLoanAmount <- cbind.data.frame(data,PredictedLoanAmount)

#Regression statistics table
```

```
regression_stats_table = data.frame(matrix(vector(),6, 2 ,
                                dimnames=list(c(), c("Statistics", "Value"))),stringsAsFactors=F)

regression_stats_table[1,1]<-'Multiple R'
regression_stats_table[2,1]<-'Multiple R Square'
regression_stats_table[3,1]<-'Adjusted R Square'
regression_stats_table[4,1]<-'Anova-F statistics'
regression_stats_table[5,1]<-'Anova-Pvalue'
regression_stats_table[6,1]<-'Standard Error'

regression_stats_table[1,2]<-sqrt(r$r.squared)
regression_stats_table[2,2]<-r$r.squared
regression_stats_table[3,2]<-r$adj.r.squared
regression_stats_table[4,2]<-r$fstatistic[1]
regression_stats_table[5,2]<-p_value
regression_stats_table[6,2]<-r$sigma
```

● *Input variable* ● *Output variable*

**What it does:**

This R script takes the input data from the Dimension and Measure columns of an Smarten cube. Based on these inputs, it then provides two outputs: a) it predicts the loan amount, and b) it indicates how accurate the model is in terms of predicting the eligible loan amount. This R script has two input variables, namely “xVals” and “yVals” and two output variables, namely “PredictedLoanAmount” that saves the result under the title “Predicted Loan Amount” and “regression_stats_table” that saves the result under the title “Regression Statistics table.”

So, the input data for “xVals” would be demographics and other attributes of applicants, and the input data for “yVals” would be the actual loan amount given. Based on these inputs, the output variable “PredictedLoanAmount” would contain the predicted loan amount, and “regression_stats_table” would indicate how accurate the model is in terms of predicting the eligible loan amount.

**Method:**

Regression is a statistical technique that explores the relationship between two or more variables.

**Input and Output variables:**

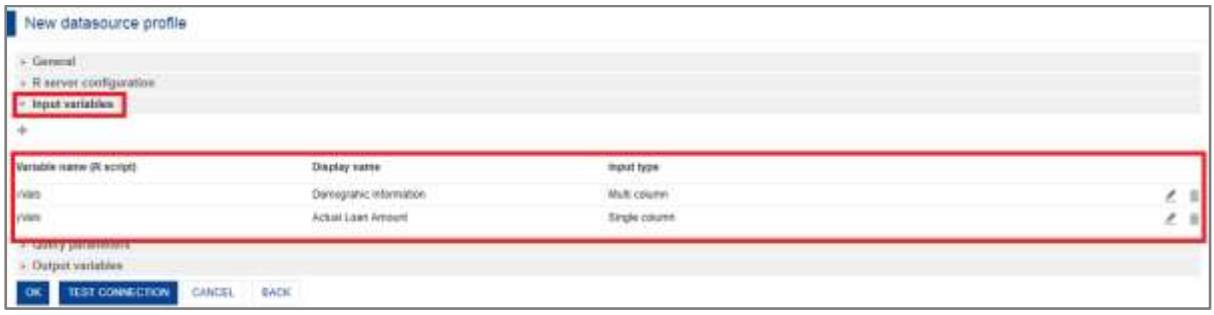
The following table lists the input and output variables configured for the sample R script and Smarten. The R script input variables are mapped with their respective columns of the Smarten cube for input data to get desired outputs in the form of a table.

| R script | | | | R cube | | |
|---|---|---|---|---|---|---|
| Input | | Output | | Input | Output | |
| **Name** | **Type** | **Name** | **Type** | **Cube columns** | **Name** | **Type** |
| xVals | Multi column | PredictedLoanAmount | Table | Grade<br>Employment<br>Tenure<br>House Ownership Status<br>Annual Income<br>Verification Status<br>Debt to Income Ratio | Predicted Loan Amount | Output data as individual table |
| | | regression_stats_table | Table | | Regression Statistics table | Output data as individual table |
| yVals | Single column | | | Loan Amount | | |

- **Create an R script profile:**

While creating an R script profile, the user needs to identify and define the input and output variables within the R script, which are then mapped with Smarten.

Shown below are the input variables being defined for this R script.




**R SCRIPT PROFILE—CUBE DATA/MANUAL INPUT VARIABLES FOR AN R SCRIPT**

In this example, two input variables, namely “xVals” and “yVals” have been defined with input type as “Multi column input” and “Single column input” respectively.

There are no Query parameters required for this R script.

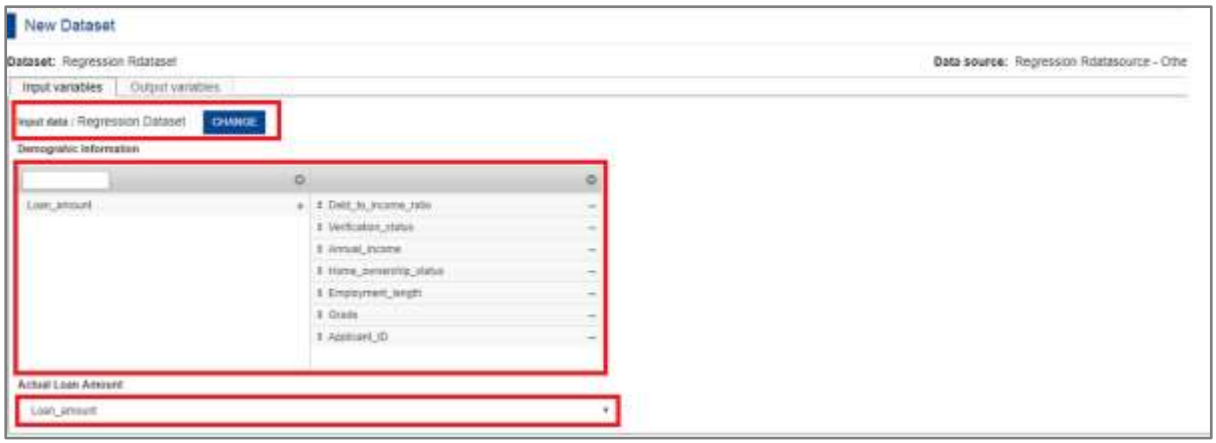
Shown below are the output variables being defined for this R script.


**R SCRIPT PROFILE—OUTPUT VARIABLES FOR AN R SCRIPT**

The two output variables for this R script are “PredictedLoanAmout” and “regression_stats_table”, and both contain the resultant data in the form of a table.

- **Create cube with the R script profile as a data source:**

While creating the R cube from the R script profile, the user needs to map input variables of the R script with Smarten cube columns.




**R CUBE—MAPPING OF CUBE COLUMNS WITH CUBE DATA/MANUAL INPUT VARIABLES FOR AN R SCRIPT**

In this example, “Regression Dataset” is the input cube for the R script. Since the first input variable “xVals” is of a “Multi column” type, seven columns of input cube have been mapped with this input variable. These columns are “Grade,” “Employment Tenure,” “House Ownership Status,” “Annual Income,” “Verification Status,” and “Debt to Income Ratio.” Since the second input variable “yVals” is a “Single column” type, only one Measure column of the input cube has been mapped with this input variable. The column is “Loan_amount.”

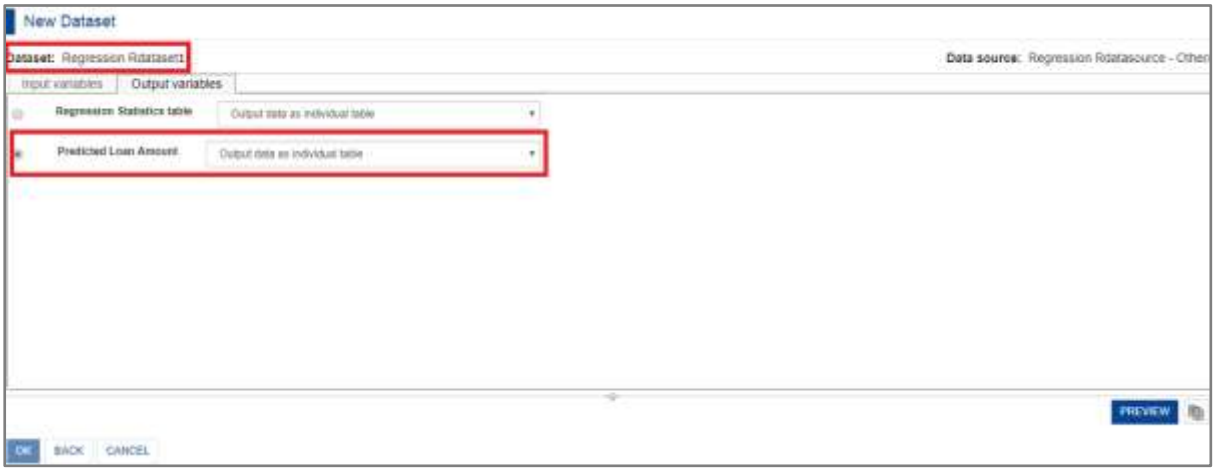
So, in this example, data belonging to the selected cube columns will be considered as input for variable “xVals,” and the data belonging to cube column of “Loan_Amount” will be considered as input for variable “yVals” as follows:

| Applicant ID | Grade | Employment Tenure | Home Ownership Status | Annual Income | Verification Status | Debt To Income Ratio | Loan Amount |
|---|---|---|---|---|---|---|---|
| 1 | B | <1 Year | RENT | 5000 | Not Verified | 26 | 1450 |
| 2 | B | <1 Year | RENT | 5000 | Not Verified | 27 | 4000 |
| 3 | B | <1 Year | RENT | 5001 | Not Verified | 26 | 1200 |
| 4 | D | <1 Year | RENT | 5001 | Verified | 30 | 4000 |
| 5 | C | <1 Year | RENT | 5002 | Not Verified | 26 | 3600 |
| 6 | B | <1 Year | RENT | 5003 | Not Verified | 25 | 2000 |
| 7 | D | <1 Year | RENT | 5003 | Not Verified | 27 | 1000 |
| 8 | C | <1 Year | RENT | 5003 | Verified | 30 | 3000 |
| 9 | B | <1 Year | RENT | 5004 | Not Verified | 29 | 4000 |
| 10 | B | <1 Year | RENT | 5004 | Not Verified | 29 | 4000 |
| 11 | B | <1 Year | RENT | 5004 | Not Verified | 30 | 3700 |
| 12 | C | <1 Year | RENT | 5005 | Verified | 26 | 2000 |
| 13 | A | <1 Year | RENT | 5005 | Verified | 30 | 2475 |
| 14 | A | <1 Year | RENT | 5006 | Not Verified | 25 | 4800 |
| 15 | A | <1 Year | RENT | 5007 | Not Verified | 25 | 3000 |
| 16 | B | <1 Year | RENT | 5008 | Not Verified | 27 | 4000 |
| 17 | A | <1 Year | RENT | 5009 | Not Verified | 27 | 1800 |
| 18 | B | <1 Year | RENT | 5009 | Verified | 28 | 2400 |
| 19 | A | <1 Year | RENT | 5009 | Not Verified | 29 | 2500 |
| 20 | C | <1 Year | RENT | 5011 | Not Verified | 28 | 3000 |

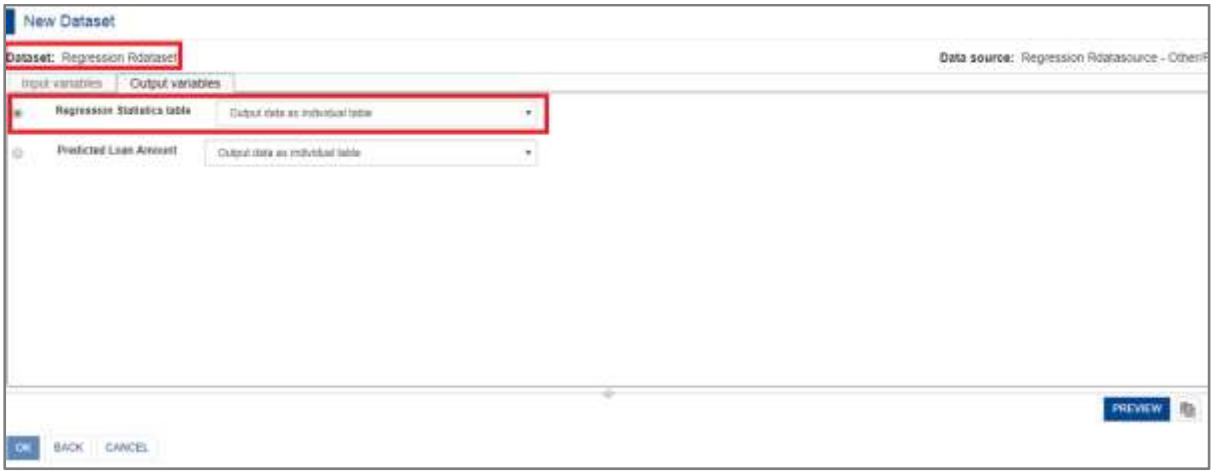
**INPUT DATA FOR AN R SCRIPT**

Similarly, the output variables of R script “PredictedLoanAmount” and “regression_stats_table” also need to be mapped with Dimension and Measure columns of the Smarten cube, but since these are two different outputs, the user needs to create two different R cubes and map each output with its respective R cube. The process of creating both R cubes will be the same, just that the output variable will change for each R cube.

Please refer to the following images for both outputs.




**R CUBE—MAPPING OF CUBE COLUMNS WITH OUTPUT VARIABLE FOR AN R SCRIPT**




**R CUBE—MAPPING OF CUBE COLUMNS WITH OUTPUT VARIABLE FOR AN R SCRIPT**

- **Access R cubes from Smarten front-end tools:**

The R cube created can now be accessed by users from front-end objects, such as dashboards, crosstab, tabular, graphs, GeoMap, and KPI, and the resultant data can be analyzed as per requirement.

In this example, output stored in “Predicted Loan Amount” and “Regression Statistics table” can be accessed through various Smarten objects. Sample tabular reports for each of them are shown below.

| Applicant ID | Grade | Employment Tenure | Home Ownership Status | Annual Income | Verification Status | Debt To Income Ratio | Loan Amount | Predicted Loan Amount |
|---|---|---|---|---|---|---|---|---|
| 1 | B | <1 Year | RENT | 5000 | Not Verified | 26 | 1450 | 2921 |
| 2 | B | <1 Year | RENT | 5000 | Not Verified | 27 | 4000 | 2920 |
| 3 | B | <1 Year | RENT | 5001 | Not Verified | 26 | 1200 | 2921 |
| 4 | D | <1 Year | RENT | 5001 | Verified | 30 | 4000 | 3300 |
| 5 | C | <1 Year | RENT | 5002 | Not Verified | 26 | 3600 | 2930 |
| 6 | B | <1 Year | RENT | 5003 | Not Verified | 25 | 2000 | 2923 |
| 7 | D | <1 Year | RENT | 5003 | Not Verified | 27 | 1000 | 3052 |
| 8 | C | <1 Year | RENT | 5003 | Verified | 30 | 3000 | 3176 |
| 9 | B | <1 Year | RENT | 5004 | Not Verified | 29 | 4000 | 2916 |
| 10 | B | <1 Year | RENT | 5004 | Not Verified | 29 | 4000 | 2916 |
| 11 | B | <1 Year | RENT | 5004 | Not Verified | 30 | 3700 | 2914 |
| 12 | C | <1 Year | RENT | 5005 | Verified | 26 | 2000 | 3183 |
| 13 | A | <1 Year | RENT | 5005 | Verified | 30 | 2475 | 3187 |
| 14 | A | <1 Year | RENT | 5006 | Not Verified | 25 | 4800 | 2943 |
| 15 | A | <1 Year | RENT | 5007 | Not Verified | 25 | 3000 | 2943 |
| 16 | B | <1 Year | RENT | 5008 | Not Verified | 27 | 4000 | 2920 |
| 17 | A | <1 Year | RENT | 5009 | Not Verified | 27 | 1800 | 2940 |
| 18 | B | <1 Year | RENT | 5009 | Verified | 28 | 2400 | 3171 |
| 19 | A | <1 Year | RENT | 5009 | Not Verified | 29 | 2500 | 2936 |
| 20 | C | <1 Year | RENT | 5011 | Not Verified | 28 | 3000 | 2927 |

**OUTPUT DATA—PREDICTED LOAN AMOUNT**

| Statistics | Value |
|---|---|
| Adjusted R Square | 0.92 |
| Anova-F Statistics | 27071.02 |
| Anova-Pvalue | 0.00 |
| Multiple R | 0.96 |
| Multiple R Square | 0.92 |
| Standard Error | 2269.80 |

**OUTPUT DATA—REGRESSION STATISTICS TABLE**

### 2.5.4 Run-Time Query Parameters, and Output as an Individual Table

**Scenario:**

A company catering to financial services may want to decide what loan amount should be given to a new loan applicant based on his/her demographics and other attributes.

**Sample R Script:**

```
library(stringr)
annual_income <- as.numeric(as.character(annual_income))
dti <- as.numeric(as.character(dti))
apply_data <- data.frame(grade,emp_length,home_status,annual_income,verification_status,dti)
memory.limit(size = 15088)
data <- data.frame(yVals,xVals)
x <- data.frame(xVals)
yNames <- names(yVals)
xNames <- c()
for (i in 1:ncol(x))
{
 if(class(x[[i]]) == "factor")
 {
  f <- paste("factor(", names(x[i]), ")")
  if(is.null(xNames) == TRUE)
  {
   xNames <- paste(xNames, f)
  } else
  { xNames <- paste(xNames, f, sep = "+")}
 } else
 {
  if(is.null(xNames) == TRUE)
  {
xNames <- paste(xNames, names(x[i]))
  } else
  {
   xNames <- paste(xNames, names(x[i]), sep = "+")
  }
 }
}
formula <- as.formula(paste(yNames, paste(xNames), sep = " ~ "))
model <- lm(formula, data=data)
r <- summary(model)
coefficients<- r$coefficients
coeff_tab <- as.data.frame(coefficients[,c(1:4)])
data.frame(row.names(coeff_tab))
ents_table <- cbind.data.frame(v,coeff_tab)
coefficients_table)[1] <-paste("Variable")
coefficients_table)[2] <-paste("Coefficients")
coefficients_table)[3] <-paste("Standard Error")
coefficients_table)[4] <-paste("t value")
coefficients_table)[5] <-paste("p value")
eff <- data.frame(str_split_fixed(coefficients_table$Variable, "\\)", 2))
```

```
split_coeff <- cbind.data.frame(split_coeff,coefficients_table$Coefficients)
if(grade %in% split_coeff$X2 == TRUE )
{
 gradeVal <- split_coeff[grepl(grade,split_coeff$X2), 3]
} else
{
 gradeVal <- 0
}
if(emp_length %in% split_coeff$X2 == TRUE)
{
 empVal <- split_coeff[grepl(emp_length,split_coeff$X2), 3]
} else
{
 empVal <- 0
}
if(home_status %in% split_coeff$X2 == TRUE)
{
 homeVal <- split_coeff[grepl(home_status,split_coeff$X2), 3]
} else
{
 homeVal <- 0
}
if(verification_status %in% split_coeff$X2 == TRUE)
{
 verificationVal <- split_coeff[grepl(verification_status,split_coeff$X2), 3]
} else
{
 verificationVal <- 0
}
ai <- "Annual.income"
if( ai %in% split_coeff$X1 == TRUE)
{
 incomeVal <- split_coeff[grepl(ai,split_coeff$X1), 3]
} else
{
 incomeVal <- 0
```

```
}
di <- "Debt.to.income.ratio"
if(di %in% split_coeff$X1 == TRUE)
{
  dtiVal <- split_coeff[grepl(di,split_coeff$X1), 3]
} else
{
  dtiVal <- 0
}
loan_amount <- split_coeff[1,3] + gradeVal + empVal + homeVal + verificationVal + (annual_income *
          incomeVal) + (dti * dtiVal)
PredictedLoanAmount  <- cbind.data.frame(apply_data,loan_amount)
```

● *Input variable* ● *Query parameters* ● *Output variable*

**What it does:**

This R script takes the input data from the Dimension and Measure columns of an Smarten cube. Based on these inputs, it then predicts the loan amount specifically for the data entered by the user in Query parameters. This R script has two input variables, namely “xVals” and “yVals,” six Query parameters, namely “grade,” “emp_length,” “home_status,” “annual_income,” “verification_status,” “dti,” and one output variable, namely “PredictedLoanAmount” that saves the result under the title “Predicted Loan Amount.”

So the input data for “xVals” would be demographics and other attributes of applicants, and the input data for “yVals” would be the actual loan amount given. Based on these inputs, the output variable “PredictedLoanAmount” would contain the predicted loan amount for a particular set of demographic data entered by the user in Query parameters.

**Method:**

Regression is a statistical technique that explores thr relationship between two or more variables.

**Input and Output variables:**

The following table lists the input and output variables configured for the sample R script and Smarten. The R script input variables are mapped with their respective columns of the Smarten cube for input data to get desired outputs in the form of a table.

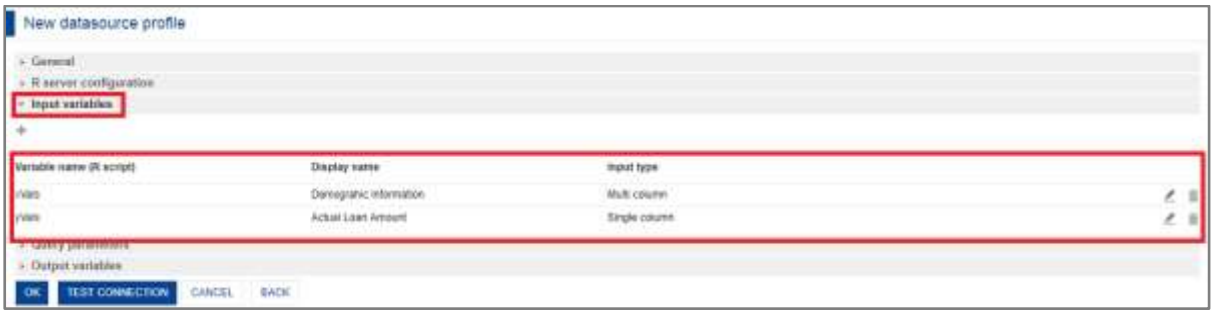
| R script | | | | | | R cube | | | |
|---|---|---|---|---|---|---|---|---|---|
| Input | | Query Parameters | | Output | | Input | Query | Output | |
| Name | Type | Name | Type | Name | Type | Cube columns | Values | Name | Type |
| xVals | Multi column | grade<br>emp_length<br>home_status<br>annual_income<br>verification_status<br>dti | Text<br>Text<br>Text<br>Text<br>Text<br>Text | Predicted LoanAmount | Table | Grade<br>Employment Tenure<br>House Ownership Status | D<br>4+ Years<br>OWN<br>16435 | Predicted Loan Amount | Output data as individual table |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Annual Income Verification Status<br>Debt to Income Ratio | Verified<br>4 | | |
| yVals | Single column | | | | | Loan Amount | | | |

- **Create an R script profile:**

While creating an R script profile, the user needs to identify and define the input and output variables within the R script, which are then mapped with Smarten. In case there are any Query parameters, they should also be defined.

Shown below are the input variables being defined for this R script.




**R SCRIPT PROFILE—CUBE DATA/MANUAL INPUT VARIABLES FOR AN R SCRIPT**

In this example, two input variables, namely “xVals” and “yVals” have been defined with input type as “Multi column input” and “Single column input” respectively.

Shown below are the Query parameters being defined for this R script.


**R SCRIPT PROFILE—QUERY PARAMETERS FOR AN R SCRIPT**

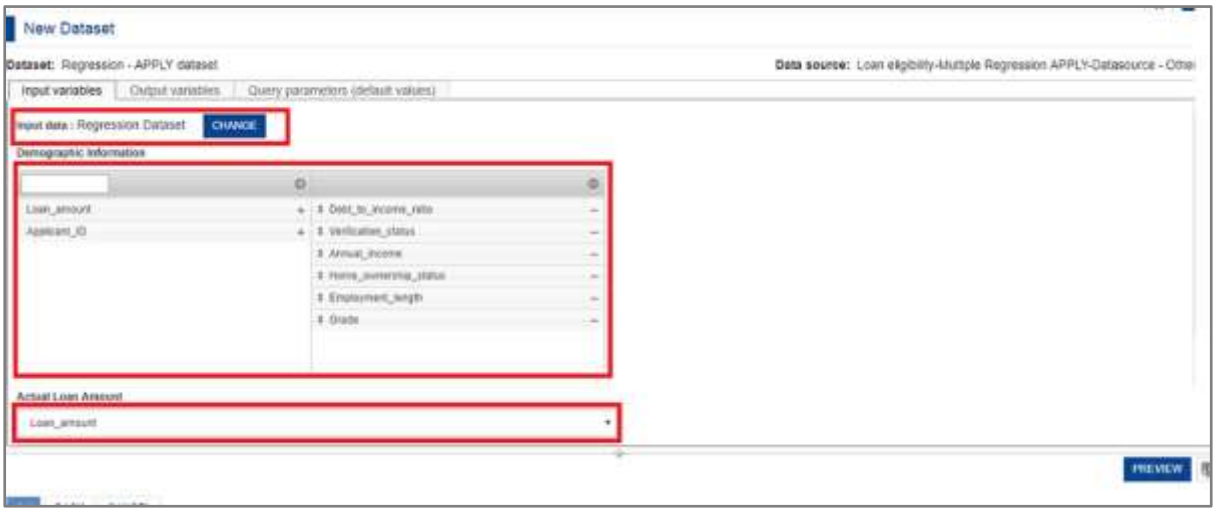
Shown below is the output variable being defined for this R script.


**R SCRIPT PROFILE—OUTPUT VARIABLE FOR AN R SCRIPT**

The output variable for this R script is “PredictedLoanAmount”, which contains the resultant data in the form of a table.

- **Create real-time R cube with the R script profile as a data source:**

While creating the real-time R cube from the R script profile, the user needs to map input variables of the R script with Smarten cube columns.




**R CUBE—MAPPING OF CUBE COLUMNS WITH INPUT VARIABLES FOR AN R SCRIPT**

In this example, “Regression Dataset” is the input cube for the R script. Since the first input variable “xVals” is a “Multi column” type, seven columns of input cube have been mapped with this input variable. These columns are “Grade,” “Annual Income,” “Verification Status,” “Employment Tenure,” “House Ownership Status,” and “Debt to Income Ratio.” Since the second input variable “yVals” is a “Single column” type, only one Measure column of the input cube has been mapped with this input variable. The column is “Actual_Loan_amount.”

So, in this example, data belonging to the selected cube columns will be considered as input for variable “xVals,” and the data belonging to cube column of “Loan_Amount” will be considered as input for variable “yVals” as follows:

| Applicant ID | Grade | Employment Tenure | Home Ownership Status | Annual Income | Verification Status | Debt To Income Ratio | Loan Amount |
|---|---|---|---|---|---|---|---|
| 1 | B | <1 Year | RENT | 5000 | Not Verified | 26 | 1450 |
| 2 | B | <1 Year | RENT | 5000 | Not Verified | 27 | 4000 |
| 3 | B | <1 Year | RENT | 5001 | Not Verified | 26 | 1200 |
| 4 | D | <1 Year | RENT | 5001 | Verified | 30 | 4000 |
| 5 | C | <1 Year | RENT | 5002 | Not Verified | 26 | 3600 |
| 6 | B | <1 Year | RENT | 5003 | Not Verified | 25 | 2000 |
| 7 | D | <1 Year | RENT | 5003 | Not Verified | 27 | 1000 |
| 8 | C | <1 Year | RENT | 5003 | Verified | 30 | 3000 |
| 9 | B | <1 Year | RENT | 5004 | Not Verified | 29 | 4000 |
| 10 | B | <1 Year | RENT | 5004 | Not Verified | 29 | 4000 |
| 11 | B | <1 Year | RENT | 5004 | Not Verified | 30 | 3700 |
| 12 | C | <1 Year | RENT | 5005 | Verified | 26 | 2000 |
| 13 | A | <1 Year | RENT | 5005 | Verified | 30 | 2475 |
| 14 | A | <1 Year | RENT | 5006 | Not Verified | 25 | 4800 |
| 15 | A | <1 Year | RENT | 5007 | Not Verified | 25 | 3000 |
| 16 | B | <1 Year | RENT | 5008 | Not Verified | 27 | 4000 |
| 17 | A | <1 Year | RENT | 5009 | Not Verified | 27 | 1800 |
| 18 | B | <1 Year | RENT | 5009 | Verified | 28 | 2400 |
| 19 | A | <1 Year | RENT | 5009 | Not Verified | 29 | 2500 |
| 20 | C | <1 Year | RENT | 5011 | Not Verified | 28 | 3000 |

**INPUT DATA FOR AN R SCRIPT**

Next, the values of Query parameters need to be entered. These are the default values with which the R script will be queried, and a real-time R cube will be created for the first time.


**QUERY PARAMETER VALUES FOR AN R SCRIPT**

Similarly, the output variable of R script "PredictedLoanAmount" also needs to be mapped with Dimension and Measure columns of the Smarten cube.


**R CUBE—MAPPING OF CUBE COLUMNS WITH OUTPUT VARIABLE FOR AN R SCRIPT**

- **Access R cubes from Smarten front-end tools:**

The R cube created can now be accessed by users from front-end objects, such as dashboards, crosstab, tabular, graphs, GeoMap, and KPI, and the resultant data can be analyzed as per requirement.

In this example, output stored in "Predicted Loan Amount" can be accessed through various Smarten real-time objects. A sample tabular report is as follows:

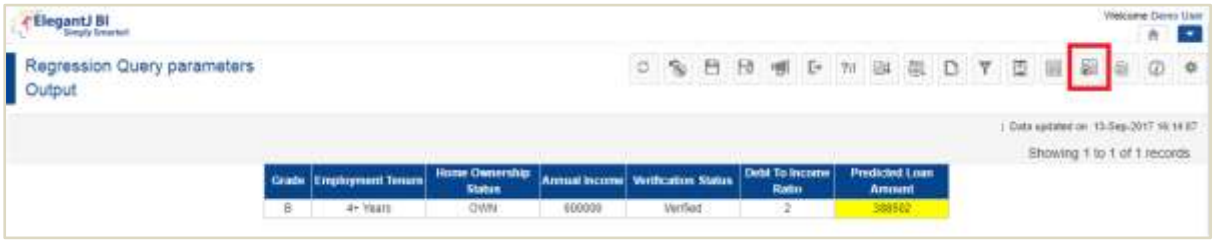
| Grade | Employment Tenure | Home Ownership Status | Annual Income | Verification Status | Debt To Income Ratio | Predicted Loan Amount |
|---|---|---|---|---|---|---|
| B | 4+ Years | OWN | 600000 | Verified | 2 | 388502 |

**OUTPUT DATA—PREDICTED LOAN AMOUNT FOR DEFAULT QUERY PARAMETER VALUES**

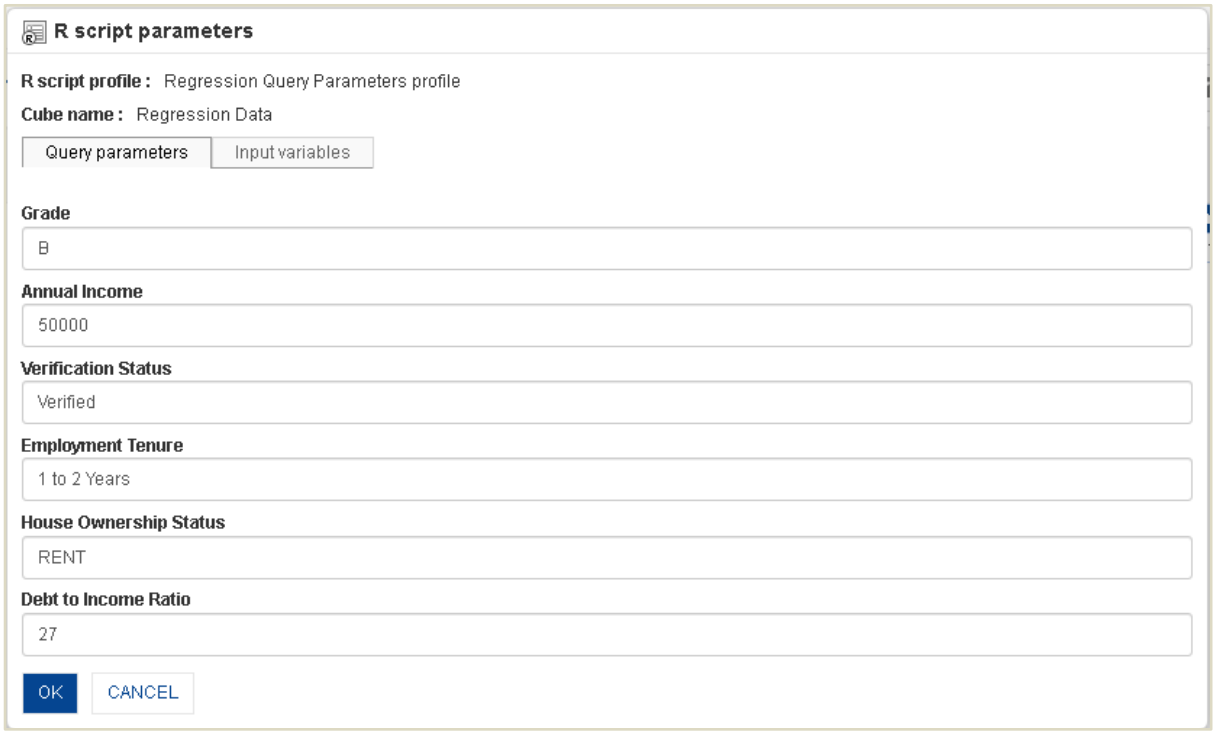
The real-time object, such as the tabular report shown above, displays the output stored in the real-time R cube created for a particular set of data entered in Query parameters.

But if a user wants to regenerate the real-time object for different values of Query parameters, Smarten allows the user to enter those values in run-time from the object itself. The user can change

the values of the Query parameters through the “R script parameters” button in the toolbar of the real-time object. As a result, the R script will be re-executed with new values of Query parameters, and the real-time R cube will be rebuilt for the new output, which, in turn, will be instantly reflected in the real-time object.




**R SCRIPT PARAMETERS ICON**




**NEW QUERY PARAMETERS**

| Grade | Employment Tenure | Home Ownership Sta | Annual Income | Verification Status | Debt To Income Ratio | Predicted Loan Amou |
|---|---|---|---|---|---|---|
| B | 1 To 2 Years | RENT | 50000.00 | Verified | 27.00 | 25500.19 |

**OUTPUT DATA—PREDICTED LOAN AMOUNT FOR NEW QUERY PARAMETER VALUES**

## 2.5.5 Output Data Appended as Rows in an R Cube or Dataset

**Scenario:**

A manufacturing company may want to forecast its sales to understand future demand. It has daily sales data for the past three years and wants to predict sales data for the next 10 days.

**Sample R Script:**

```
data <- data.frame(xVals,yVals)
cols <- c("Date","Sales")
colnames(data) <- cols

forecast_period <- as.numeric(forecast_period)
date_format <- "%Y-%m-%d"
data$Date <- format(as.Date(as.character(data$Date), format = date_format),"%Y-%m-%d")
data <- data[order(data$Date),]

myvector <- data$Sales
start_date <- data$Date[1]
end_date <- data$Date[nrow(data)]
ts <- ts(myvector,start = as.numeric(noquote(unlist(strsplit(start_date, "-")))),end =
as.numeric(noquote(unlist(strsplit(end_date, "-")))),frequency = 365)
library(forecast)
fitHW <- HoltWinters(ts)
pred_test <- predict(fitHW,n.ahead=forecast_period)
predictions_test <- as.data.frame(pred_test)
 dates <- data.frame()
for (i in 1:forecast_period)
{
 dates[i,1] <- as.character(as.Date(end_date) + i)
}
predictions_test <- cbind.data.frame(dates, predictions_test)
colnames(predictions_test) <- cols
predictions_test$Date <- format(as.Date(as.character(predictions_test$Date), format = "%Y-%m-
%d"),date_format)
predictions_test$Date <- as.character(predictions_test$Date)
```

● *Input variable* ● *Output variable*

**What it does:**

This R script takes the input data from the Dimension and Measure columns of an Smarten cube and also manual input from the user. Based on these inputs, it then forecasts the values of sales for a certain future time period. This R script has three input variables, namely "xVals," "yVals," and "forecast_period." The input data for xVals and yVals come from the Dimension and Measure columns of the Smarten cube, whereas input data for "forecast_period" is entered manually by the user. It has one output variable, namely "predictions_test" that saves the result under the title "Forecast Sales." So, the input data for "xVals" would be the date of sales, and the input data for "yVals" would be the actual sales value for that date. The input data for "forecast_period" would be the period for which the forecast is required. Based on these inputs, the output variable "predictions_test" would contain the predicted sales for a time period entered by the user.

**Method:**

Holt-Winters is one of the methods or algorithms used to forecast data points in a series, provided the series is "seasonal," i.e., repetitive over some period.

**Input and Output variables:**

The following table lists the input and output variables configured for the sample R script and Smarten. The R script input variables are mapped with their respective columns of the Smarten cube for input data to get desired outputs in the form of a table.

| **R script** | | | | **R cube** | | | |
|---|---|---|---|---|---|---|---|
| **Input** | | **Output** | | **Input** | | **Output** | |
| **Name** | **Type** | **Name** | **Type** | **Cube columns** | **Manual** | **Name** | **Type** |
| xVals | Single column | predictions_test | Table | Date | No. of days for forecast | Forecast Sales | Append Output data as row |
| yVals | Single column | | | Actual Sales Amount | | | |
| forecast_period | Text | | | | | | |

- **Create an R script profile:**

While creating an R script profile, the user needs to identify and define the input and output variables within the R script, which are then mapped with Smarten. In case there are any Query parameters, they should also be defined.

Shown below are the input variables being defined for this R script.


**R SCRIPT PROFILE—CUBE DATA/MANUAL INPUT VARIABLES FOR AN R SCRIPT**

In this example, input variables, namely “xVals” and “yVals” have been defined with input type as “Single column input,” whereas the “forecast_period” has been defined with input type as “Text.”

There are no Query parameters required for this R script.

Shown below are the output variables being defined for this R script.


**R SCRIPT PROFILE—OUTPUT VARIABLE FOR AN R SCRIPT**

The output variable for this R script is “predictions_test”, which contains the resultant data in the form of a table.

- **Create cube with the R script profile as a data source:**

While creating the R cube from the R script profile, the user needs to map input variables of the R script with Smarten cube columns.


**R CUBE—MAPPING OF CUBE COLUMNS WITH INPUT VARIABLES FOR AN R SCRIPT**

In this example, “Sales Forecasting Holt Winters Dataset” is the input cube for the R script. Since the first input variable “xVals” is a “Single column” type, only one Dimension column, namely “Date” of input cube has been mapped with this input variable. Since the second input variable “yVals” is also a “Single column” type, only one Measure column, namely “Sales” of input cube has been mapped with this input variable. The third input variable “forecast_period” is a “Text” type, and therefore the value for this variable has to be entered manually by the user. For this example, this is the number of days for which the sales value has to be predicted.

So, in this example, data belonging to the selected cube columns will be considered as input for variable “xVals” and “yVals” as follows:

| Date | Sales |
|---|---|
| 01-Jan-2014 | 80.27 |
| 02-Jan-2014 | 101.75 |
| 03-Jan-2014 | 312.71 |
| 04-Jan-2014 | 49.28 |
| 05-Jan-2014 | 69.20 |
| 06-Jan-2014 | 94.67 |
| 07-Jan-2014 | 288.50 |
| 08-Jan-2014 | 58.45 |
| 09-Jan-2014 | 77.79 |
| 10-Jan-2014 | 91.33 |
| 11-Jan-2014 | 299.18 |
| 12-Jan-2014 | 117.50 |
| 13-Jan-2014 | 104.45 |
| 14-Jan-2014 | 82.28 |
| 15-Jan-2014 | 116.22 |
| 16-Jan-2014 | 312.83 |
| 17-Jan-2014 | 57.64 |
| 18-Jan-2014 | 53.62 |
| 19-Jan-2014 | 87.57 |
| 20-Jan-2014 | 91.22 |
| 21-Jan-2014 | 110.03 |

**INPUT DATA FOR AN R SCRIPT**

Similarly, the output variable of the R script "predictions_test" is mapped with the Dimension and Measure columns of the Smarten cube.


**R CUBE—MAPPING OF CUBE COLUMNS WITH OUTPUT VARIABLE FOR AN R SCRIPT**

- **Access R cubes from Smarten front-end tools:**

The R cube created can now be accessed by users from front-end objects, such as dashboards, crosstab, tabular, graphs, GeoMap, and KPI, and the resultant data can be analyzed as per requirement. In this example, the output stored in the "predictions_test" variable can be accessed through various Smarten objects. A sample tabular report is as follows:

| Date | Sales |
|---|---|
| 2016-12-21 | 48.71 |
| 2016-12-22 | 56.58 |
| 2016-12-23 | 95.96 |
| 2016-12-24 | 89.22 |
| 2016-12-25 | 91.82 |
| 2016-12-26 | 152.83 |
| 2016-12-27 | 104.17 |
| 2016-12-28 | 108.22 |
| 2016-12-29 | 114.96 |
| 2016-12-30 | 61.81 |
| 2016-12-31 | 78.56 |
| 2017-01-01 | 165.38 |
| 2017-01-02 | 303.22 |
| 2017-01-03 | 299.88 |
| 2017-01-04 | 371.43 |
| 2017-01-05 | 107.70 |
| 2017-01-06 | 128.98 |
| 2017-01-07 | 183.52 |
| 2017-01-08 | 95.49 |
| 2017-01-09 | 59.38 |
| 2017-01-10 | 164.02 |

**OUTPUT DATA**

Since the output in the R cube has been configured to be **appended as rows**, the resultant prediction will be added and displayed in new rows of the existing table.

# 3 Product and Support Information

Find more information about ElegantJ BI-Smarten and its features at www.smarten.com
Support: support@smarten.com
Sales: sales@smarten.com
Feedback & Suggestions: support@smarten.com
Support & Knowledgebase Portal: support.smarten.com