IBM SPSS Statistics and open source: A powerful combination

**Highlights**

- Why integrate SPSS Statistics and open source?
- Integrating open source with IBM SPSS Statistics
- Extension bundles: Using R or Python programs created by others
- Writing your own R or Python programs

The purpose of this paper is to demonstrate the features and capabilities provided by the integration of IBM® SPSS® Statistics and open source programming languages like R and Python. Open source users get access to superior data management, ease of use and presentation-quality output that is available from IBM SPSS Statistics. SPSS Statistics users get access to a rich, ever-expanding collection of statistical analysis and graphing libraries to help them gain deeper insights from their data. Using IBM SPSS Statistics and open source together enables the most of both worlds.

**Why integrate SPSS Statistics and open source?**

IBM SPSS Statistics is one of the world’s leading statistical software solutions. It provides predictive models and advanced analytics to help solve business and research problems. For many businesses, research institutions and statisticians, it is the de facto standard for statistical analysis. Organizations use SPSS Statistics to:

- Understand data
- Analyze trends
- Forecast and plan
- Validate assumptions
- Drive accurate conclusions

SPSS Statistics has been continually developed and tested since 1968. Over that period, many forms of statistical analysis have been embedded in the software. In addition, the algorithms that execute the equations have been tested by developers and users in academia, in laboratories and in virtually every type of business. As a result, users can be confident that the software has been thoroughly tested and its results found to be reliable.
The SPSS Statistics environment makes it easier for you to quickly access, manage and analyze datasets, including survey data, corporate databases, data downloaded from the web and much more. Advanced statistical procedures and visualization can provide a robust, user friendly and integrated platform for understanding your data and solving complex business and research problems.

IBM SPSS Statistics can help you address all facets of the analytical process, from data preparation and management to analysis and reporting. It provides tailored functionality and customizable interfaces for different skill levels and functional responsibilities. It also enables users to create high-resolution graphs and presentation-ready reports to easily communicate results.

For example, consider Robert, who is interested in analyzing data on miles-per-gallon (mpg) for different types of cars. Figure 1 shows a segment of what the data looks like in the SPSS Statistics Data Editor.

For this analysis, Robert might first run the Descriptives procedure to get an idea of the distribution of the data for mpg. He does this from the Descriptives dialog box (Figure 2).

The output (in this case, tabular output) is shown in the SPSS Statistics Viewer (Figure 3).
Supported open-source programming languages

Given the popularity of Python and R programming languages in the field of data mining, statistics and data science, IBM is continuing to invest in integrating SPSS Statistics and these open-source programming languages to empower hybrid statistics teams of different skill levels and backgrounds with best-in-class integrated statistical package and experiences. R is an open source programming language and software environment for statistical computing and graphics (www.r-project.org). The R language has been very popular with statisticians and data miners for developing statistical software and is widely used for advanced data analysis. R provides a wide variety of advanced statistical and graphical techniques and is highly extensible. R is available as free software under the terms of the Free Software Foundation GNU General Public License. It runs on Windows and MacOS, a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux). R can be easily extended with packages. Note that on Windows only, the BASIC language is also supported for scripting.

On the other hand, Python is a general-purpose and high-level programming language (www.python.org). You can use Python for developing desktop GUI applications, websites and web applications. Also, Python, as a high level programming language, allows you to focus on core functionality of the application by taking care of common programming tasks. The simple syntax rules of the programming language makes it easier for you to keep the code base readable and the application maintainable.

Among the topics that R and Python users commonly discuss are scalability, basic data and output management issues such as connecting to databases, improving output quality and sharing algorithms with others not familiar with programming. On the other hand, SPSS Statistics users, however, might want to use some of the R functions that are not available within the product without having to learn how to program.

Therefore, integrating R or Python with SPSS Statistics makes sense. The combined strength of both helps address the needs of the two user groups. SPSS Statistics is a convenient platform from which R or Python users can handle large data sets and get high-quality graphs and other forms of output. Some of the other benefits are the ease of use of SPSS Statistics and the ability to distribute integrated R or Python packages to a wide range of users who are not familiar with those programming languages. This integration also provides SPSS Statistics users with easy access to a very large number of open source statistical functions.
Integrating open source with IBM SPSS Statistics

SPSS Statistics comes with Python and R integration already set up. This includes corresponding versions of the R and Python language runtimes.

Each user on a particular machine receives their own virtual environment containing the R and Python language compilers, runtimes and tooling. These virtual environments are installed upon first use of the program, and can be customized to add or change specific libraries and features of the language.

Users can also install separate versions of either the R or Python runtimes externally and redirect the program to use those environments instead of the default virtual environment that is set up upon first use. This enables users to leverage their own pre-configured language installations, or to support newer or older minor versions of the language compilers and runtimes.

The SPSS Statistics-Integration Plug-in for R is automatically installed and extends the SPSS Statistics command syntax language with the full capabilities of the R programming language. The plug-in also provides access to an R integrated development environment, which makes it easy for users to develop, test and debug R programs for use with SPSS Statistics. It is available for Windows, Mac OS and SPSS Statistics Server. After installing Statistics, you can choose to use R programs that have already been written or you can write your own.

The SPSS Statistics-Integration Plug-In for Python works similarly, and provides access to standard Python runtimes and libraries.
Extension bundles: Using R or Python programs created by others

SPSS Statistics and R or Python integration enables you to take advantage of the R or Python programs that others have written and packaged as extension bundles. The pre-coded algorithms remove the need for intense R or Python programming, especially if you are pressed for time or expertise in R or Python programming is scarce. These R or Python programs are deployed as extension bundles.

After an extension bundle is installed, its dialog box is accessible from the SPSS Statistics menus, and the extension command can be run as if it were any built-in command. The R or Python program functions as if it were a native dialog box and a syntax command.

To understand an extension bundle better, consider again the example where the user is analyzing data on mpg. Suppose Robert is now interested in analyzing mpg as a function of engine size, horsepower and curb weight of the vehicle. However, he wants to go beyond standard linear regression and do the analysis with quantile regression.

Quantile regression is provided as an extension bundle in the integration plug-in for R, and Robert can use it to understand the distribution of mpg as a function of the predictors. For the more technically minded, quantile regression estimates one or more conditional quantiles (0 <= q < 1) for a linear model. In contrast, ordinary regression estimates only the conditional mean.

The Quantile Regression dialog box looks just like any native SPSS Statistics dialog box (Figure 4).

Robert can simply specify the dependent and independent variables and the list of quantiles and then click OK to run the analysis. SPSS Statistics calls R and does the analysis using the R quantreg package, but Robert sees none of that. The results of the analysis from R are then presented as tabular and chart output in the SPSS Statistics Viewer. As in any typical regression analysis, results include tabular output of

![Figure 4: Quantile Regression dialog box.](image)

![Figure 5: A table of regression coefficients.](image)
the regression coefficients. In this case, a separate table of regression coefficients is created for each specified quantile (Figure 5), where the table for the 0.05 quantile is displayed. The results shown in the table come from running the analysis in R, retrieving the results from R and displaying them in the SPSS Statistics Viewer.

The quantile regression procedure also produces chart output that shows the coefficients for each of the predictors as a function of the specified quantiles (Figure 6). The chart is actually generated by R but is automatically rendered in the SPSS Statistics Viewer. Again, all of that complexity is hidden from Robert, who just sees the chart output.

Where to find extension bundles

A collection of extension bundles that implement a variety of R or Python statistical algorithms are automatically installed with the product.

Many more extension bundles that implement R or Python statistical algorithms are available to search for and download. They are hosted on the IBM Predictive Analytics GitHub repository at: ibmpredictiveanalytics.github.io.

The Extension Hub contains over 100 custom extensions. You can download them from within SPSS Statistics. This “Extension Hub” feature is available from Extensions > Extension Bundles. Already installed bundles can be updated in the same way.
Writing your own R or Python programs

Using extension bundles is just one way of using R or Python in SPSS Statistics. You can write your own R or Python program and integrate it in SPSS Statistics at various levels by:

- Creating a custom dialog that generates the syntax for an R or Python extension command or explicit R or Python code
- Creating an extension command
- Running R or Python code directly from within SPSS Statistics

These functions enable you to:

- Read case data from the active dataset into R or Python
- Get information about data in the active dataset
- Get output results from syntax commands
- Write results (back to a new dataset, to pivot table and to graphics) from R or Python to be displayed in SPSS Statistics

You can also drive the SPSS Statistics processor from Python programs using the Python IDE of your own choice. See here for more information.

Creating a custom dialog that generates the syntax for an R or Python extension command or explicit R or Python code

With the Custom Dialog Builder, you can create a user interface that generates command syntax for an extension command implemented in R or Python. You can then view the output (Figures 7–9) from running the dialog in the SPSS Statistics Viewer. An R or Python program can also be directly embedded in a custom dialog.

Figure 7: List of available R extension commands in SPSS Statistics.

Figure 8: A syntax template for explicit R code.

Figure 9: A syntax template for explicit R code.
Creating an extension command

An extension command is a custom SPSS Statistics command that is implemented in R or Python. You can integrate an R or Python program into SPSS Statistics by creating an extension command that implements the R or Python program (Figure 10). Integrating an R or Python algorithm into SPSS Statistics is particularly useful when a user needs an advanced statistical function but lacks the expertise or time to create such a program. In such a scenario, a methodology group, which creates R or Python algorithms for much needed statistical functions, could write the code and distribute it as an extension command. The user can then use it just as if it were a built-in SPSS Statistics command.

To create an extension command:

- Write the program as you would an R or Python function
- Define the SPSS Statistics syntax for the extension command in an xml file that specifies the command name, the subcommands and the keywords (Figure 11)
- Declare the syntax in an “R Run Function” or “Python Run Function” and call the function (Figure 12)
- Use the built-in extension utilities such as extension.py and its R language equivalent to help speed development

The user input is automatically validated and mapped to variables and passed to the implementing function.

The R or Python code calls APIs in the integration plug-in for R or integration plug-in for Python respectively that interacts with SPSS Statistics. Text in the R or Python code that is intended for output, such as pivot table labels and error messages, can be enabled for translation.

Interaction between R or Python and SPSS Statistics enables your programmers to write code unique to your organization and execute it with common SPSS command syntax.
Running R or Python code directly from within SPSS Statistics

If you are familiar with R, you can run your own R code in SPSS Statistics. To run the R code, you enclose the code in a BEGIN PROGRAM R – END PROGRAM block of SPSS Statistics command syntax. Similarly, to run Python code, you enclose the code in a BEGIN PROGRAM PYTHON3 - END PROGRAM block of SPSS Statistics command syntax.

When the syntax is submitted, the code inside the block is executed in R or Python respectively. The code is typically a combination of ordinary R or Python code and calls to SPSS-specific R functions (provided with the Integration Plug-in for R) or Python functions (provided with the integration plug-in for Python) respectively that enable R or Python to interact with SPSS Statistics.

Figure 13 shows an example of an R program to run a regression. The functions that facilitate the process of using R programming features with SPSS command syntax include:

- spssdata.GetDataFromSPSS (Gets data from the active dataset)
- spssdictionary.GetDictionaryFromSPSS (Gets variable dictionary information from the active dataset)
- spsspivottable.Display (Renders tabular output from R as a pivot table that can be displayed in the IBM SPSS Statistics Viewer or can be written to an external file with the SPSS Statistics Output Management System. Pivot tables produced with this function are just like pivot tables produced by native SPSS code)

Extension bundles make it easy to distribute R or Python programs to users who don’t have open source skills.

Figure 13: An R program for running a regression.

All artifacts can be easily packaged to create an extension bundle for easy distribution and installation. The extension bundle contains:

- A custom dialog package (.cfe) file that specifies the custom dialog box
- An XML file that specifies the syntax of the extension command
- The Implementation code file(s) written in R or Python
- A MANIFEST directory containing a MANIFEST.MF file that describes the extension contents

Other users can simply install the extension bundle from the SPSS Statistics menus. They can then use the dialog and extension command in the same manner as a native dialog or command. To enable collaboration and knowledge sharing, contributed R or Python-based extensions can be shared between users of SPSS Statistics.
Conclusion

Both SPSS Statistics and open source (R or Python) can independently boast strengths that have been tested over time and are strongly accepted in the statistical community. Moreover, these strengths complement each other to create an even more powerful set of statistical functions and features that benefit the statistical community as a whole.

R or Python users can access superior data management capabilities, which enables them to handle much larger data sets. Also, the R and Python output generation and management API from SPSS Statistics provides R and Python users with a richer set of graphical and pivot table output options, which can lead to a better user experience. Finally, SPSS Statistics acts as an ideal deployment vehicle to distribute R or Python packages to a wide range of users.

SPSS Statistics users gain access to many more statistical functions, which enables them to carry out complicated analysis without the hassles of learning a complex programming language such as R or Python. The advantages of using open source (R or Python) and SPSS Statistics together are many and worth considering.

Try SPSS Statistics now
To start your no-cost trial of SPSS Statistics, visit: ibm.com/account/reg/us-en/signup?formid=urx-19774

For more information
To learn more about using SPSS Statistics with open source (R or Python) programming languages and to access a rich set of resources, visit the following websites:
ibm.com/docs/en/spss-statistics
ibm.com/community/spss-statistics
ibmpredictiveanalytics.github.io