Data Example Applications

These example applications for S-PLUS are designed to demonstrate the use of functions that import or manipulate data in the USGS library. A general knowledge of S-PLUS is assumed. For users needing introductions to S-PLUS, several are available in bookstores and online. Examples include Burns, Patrick, 2008, S Poetry, last accessed March 19, 2009, at *http://www.burns-stat.com/pages/spoetry.html*, and Lam, Longhow, 2001, An introduction to S-PLUS for Windows: Amsterdam, The Netherlands, CANdiensten, 230 p., last accessed March 26, 2010, at *http://www.splusbook.com*. A detailed introduction to the S language is provided by Kraus, A., and Olson, M., 2002, The basics of S-PLUS: New York, Springer, 420 p. Another example of an introduction that concentrates on statistical applications is Venables, W.N., and Ripley, B.D., 2002, Modern applied statistics with S: New York, Springer, 495 p.

A few type face conventions are used to assist the user in interpreting the guidelines. S-PLUS object names and function arguments are in *italics* and function names are followed by parentheses *()*, window names are highlighted in **bold**, and object class names are in a sans-serif font. User commands, called calls, are in `plain text` and on separate lines. Any printed output from a call, whether in a **Commands Window** or **Report Window**, is shown in `plain text` in a box. Plots are shown as is, copied from the default settings for an S-PLUS **Graph Sheet**; however, individual user preferences can change the way the plots are displayed. Any S-PLUS dialog menu directions are shown in **bold** separated by a vertical bar (|). Column names and other dialog box entries are underlined.

An important way to manage data is to use the **Object Explorer Window**. The **Object Explorer Window** can be opened by clicking on the small icon that is a little yellow folder with two blue dots above it  .

Application 1. *Import an RDB file*
Application 2. *Import an Excel file*
Application 3. *Convert data by result to data by sample*

Import an RDB file

Relational database (RDB) files are a convenient way to transfer rectangular data (data in columns and rows) between databases because the data are stored in ASCII format. Each RDB file has three sections. The top, or header, section is optional and contains descriptive information about the data. Each line is preceded by an octothorp (#). The next section contains two lines: the first line contains the column names and the second line contains the column definitions. This information and all data are separated by tabs. The last section contains the data.

This example application uses several different RDB files to demonstrate the differences between the several formats. All of the files are located in the USGS library folder, which is typically C:\Program files\TIBCO\splus81\local\library\USGS.

Step 1. Working with NWIS data

The USGS National Water Information System (NWIS) consists of several components and subsystems including the Web accessible interface NWISWeb. The various NWIS subsystems use multiple, different formats and conventions for output. As a consequence, different techniques are needed to handle the same data within S-PLUS depending on how it was obtained from NWIS. The best format for importing to S-PLUS from NWIS is the tab-delimited RDB file output. Most other formats are acceptable, but require more work to import the file.

Daily values data, typically streamflow, continuous groundwater levels, or continuous water-quality data, usually consists of a single value for each day of the (water) year. Sometimes values are missing. Output from the Automated Data Processing System (ADAPS) using the nwts2rdb command produces an RDB file with one line for each day with a missing value represented by a blank value field. This converts easily to an S-PLUS cts (calendar time-series) object, which requires time on a regular interval. NWISWeb on the other hand does not produce a line for a day with a missing value. This converts easily to an S-PLUS its (irregular time-series) object, which can record time at various intervals.

Groundwater-level RDB output from NWISWeb does not always use the appropriate column type specifiers. The conversion from a data.frame to an its object therefore requires different coding than with the output from the Ground-Water Site Inventory (GWSI) subsystem.

The four RDB files listed below provide examples of these different techniques:

- The RDB file MarshRiverDVA.rdb was retrieved on November 25, 2002 from ADAPS using nwts2rdb. It contains daily values for October through March of water year 1999 at USGS station 05067500 (MARSH RIVER NEAR SHELLY, MN). Notice that there are no streamflow data for the period 1998-11-04 through 1999-03-16, but there is a line in the RDB file for each of these days.
- The RDB file MarshRiverDVW.rdb was retrieved on November 25, 2002 from NWISWeb and is for the same station and period as in MarshRiverDVA.rdb. In this example, the streamflow data for the period 1998-11-04 through 1999-03-16 are again missing, but there is not a line for each of the missing days.
- The RDB file TogoGWSI.rdb was retrieved on November 25, 2002 from GWSI. It contains groundwater levels for August 1999 through September 2001 at USGS station 474921093144001, local number 062N23W26CDC, near Togo in Itasca County, Minnesota. Notice that the column type specifications are "8D 4S 7N," corresponding to Date, String, and Numeric, respectively. This is consistent with the data in those columns.
- The RDB file TogoNWISWeb.rdb was retrieved on November 25, 2002 from NWISWeb and is for the same station and period as in TogoGWSI.rdb. Notice that the column type specifications are all "s," corresponding to String although one of the columns contains dates and others contain numeric data.

There are two methods for importing data into S-PLUS: the **Commands Window**, described in Step 2a, or the **Menu Dialog Windows**, described in Step 2b. Using the **Menu Dialog Window** avoids typing errors and allows one to rapidly import many files from a single folder.

Step 2a. Importing data using the Commands Window

Use the *importRDB()* function to create a *data.frame* object for each RDB file by typing the following calls in the **Commands Window**:

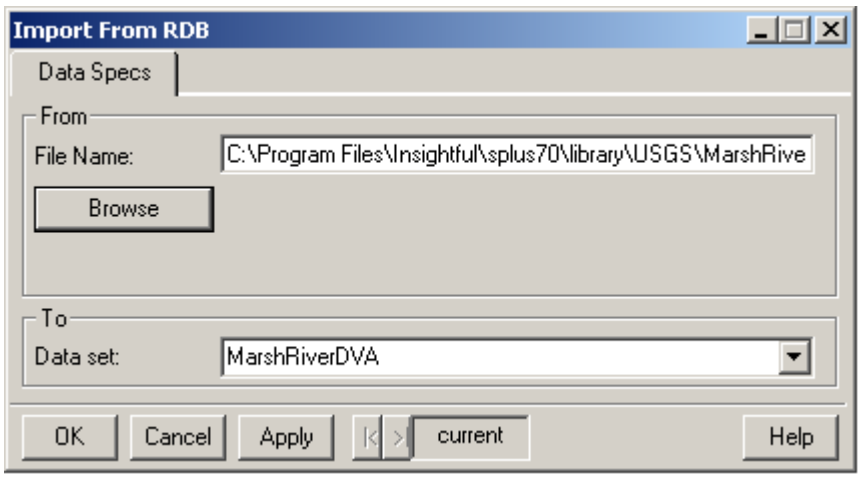
```
MarshRiverDVA.df <- importRDB(paste(getenv("S_HOME"),
      "\\library\\USGS\\MarshRiverDVA.rdb", sep=""))
```

```
MarshRiverDVW.df <- importRDB(paste(getenv("S_HOME"),
      "\\library\\USGS\\MarshRiverDVW.rdb", sep=""))
TogoGWSI.df      <- importRDB(paste(getenv("S_HOME"),
      "\\library\\USGS\\TogoGWSI.rdb", sep=""))
TogoNWISWeb.df   <- importRDB(paste(getenv("S_HOME"),
      "\\library\\USGS\\TogoNWISWeb.rdb", sep=""))
```

These calls produce the no report output, but the data.frame objects can be opened in a **Data Window** by clicking on name or icon in the right pane in the **Object Explorer**.

Step 2b. Importing data using the Menu Dialogs

Click on **File | Import Data | From RDB…** to open the **Import From RDB** window.



Click the **Browse** button and navigate to the USGS library folder in the **Select RDB file to import** window. Select the MarshRiverDVA.rdb file and then click **OK**. This inserts the complete pathname in the **File Name:** box and MarshRiverDVA in the **Data set:** box. Click **Apply** to import the file and keep the **Import from RDB** window open. Repeat if needed for the MarshRiverDVW.rdb, TogoGWSI.rdb, and TogoNWISWeb.rdb files, clicking **OK** on the last one. Using the **Import from RDB** window never displays the imported data in a **Data Window**.

Import an Excel file

Excel is a common format for exchanging data between systems. Excel files are desirable because of the formatting options, however they can present some challenges for importing data into S-PLUS. One of those challenges is importing character qualification codes for water-quality data.

A column containing character, also called string, data can be imported into S-PLUS as either type factor or type character. Each of these types has advantages and disadvantages. Character data that is set as type factor typically facilitates plotting and data analysis. For example, boxplots can be plotted by STAID if the column STAID is of type factor but will not be labeled correctly if it is of type character. In general, prior to importing any data from Excel or other formats in the **Import From File** window, verify or change the default **General Settings** to import text as type factor. This setting is on the **Data** page of the **Options | General Settings...** menu item. Set **Default Text Col** to factor. An important exception to importing character data as factor is when the data contain columns of data qualification codes such as remarks codes like "<," "E," or no code ("") for water-quality data. In this case, it is usually better to import the column as type character so that "" is not converted to the missing value (NA) when it represents no code. It is possible to convert the column to type factor after the data are imported.

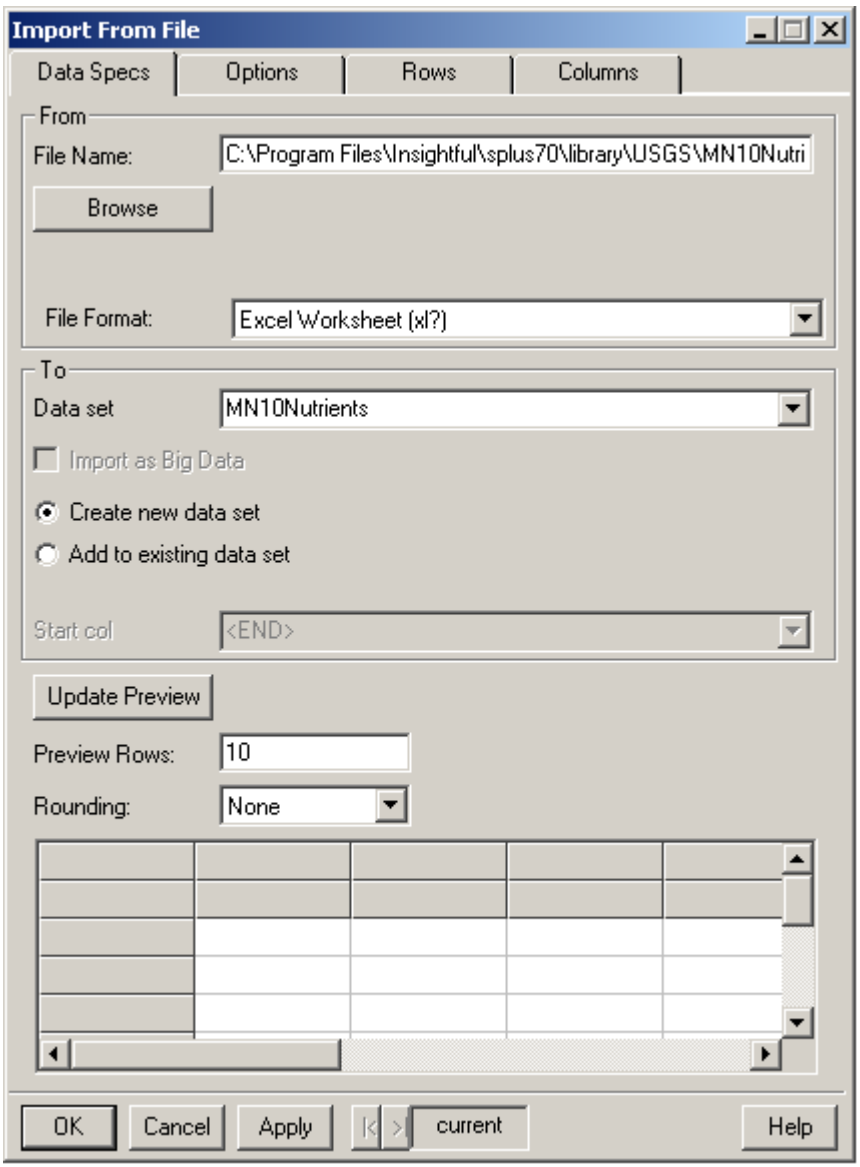
This example application uses the MN10Nutrients.xls Excel file to demonstrate some issues related to importing water-quality data from Excel files. The file is located in the USGS library folder, which is typically C:\Program files\TIBCO\splus81\local\library\USGS. The first line in the Excel file contains expanded descriptions of the column. The second line contains column names that are appropriate for S-PLUS.

MN10Nutrients.xls [Compatibility Mode]

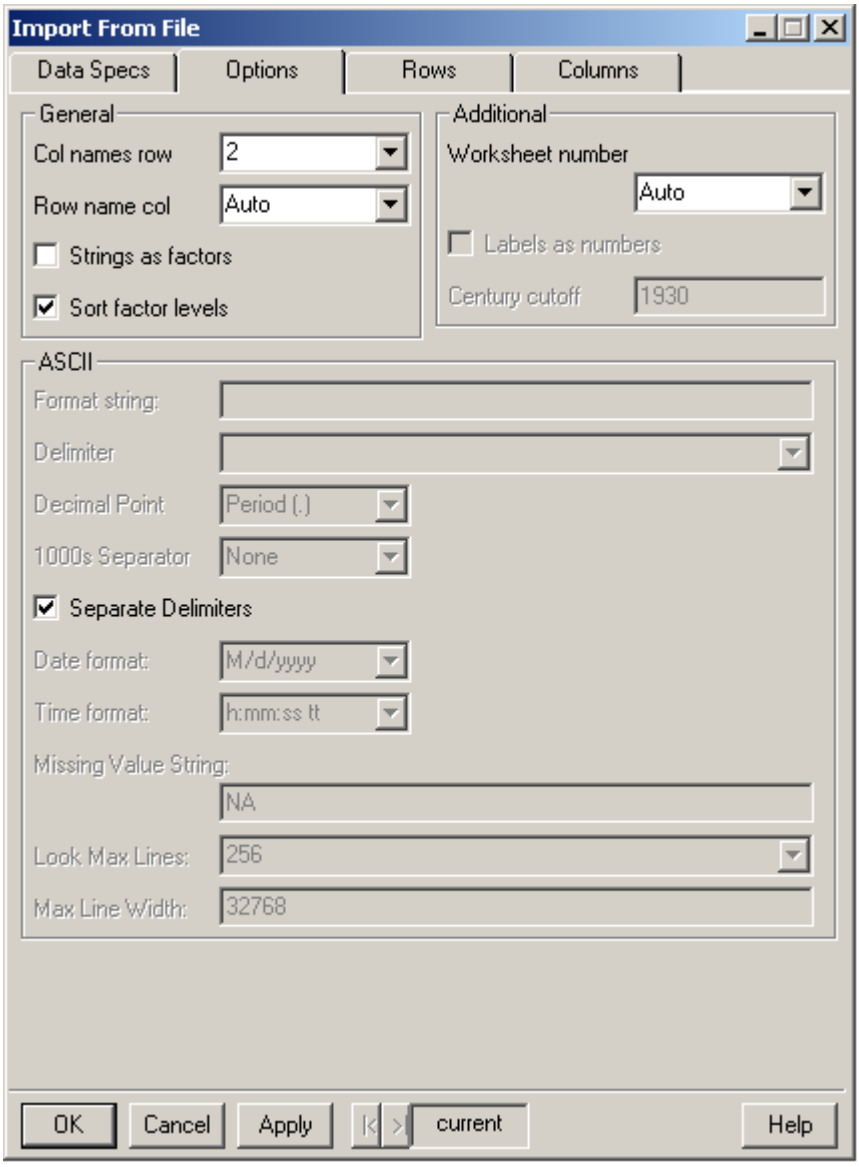
	A	B	C	D	E	F
1	STAID	SNAME	DATES	TIMES	R00608 Remark code for 00608	00608 Ammonia, wf mg/l as N
2	STAID	SNAME	DATES	TIMES	Ammonia.r	Ammonia
3	5278560	SOUTH FORK CROW RIVER ABV OTTER LK NR HUT	20070228	1000		0.83957
4	5278560	SOUTH FORK CROW RIVER ABV OTTER LK NR HUT	20070829	730	<	0.02
5	5278580	SOUTH FORK CROW RIVER BELOW HUTCHINSON,	20070228	1330		0.15366

Step 1. Importing data using the Menu Dialogs

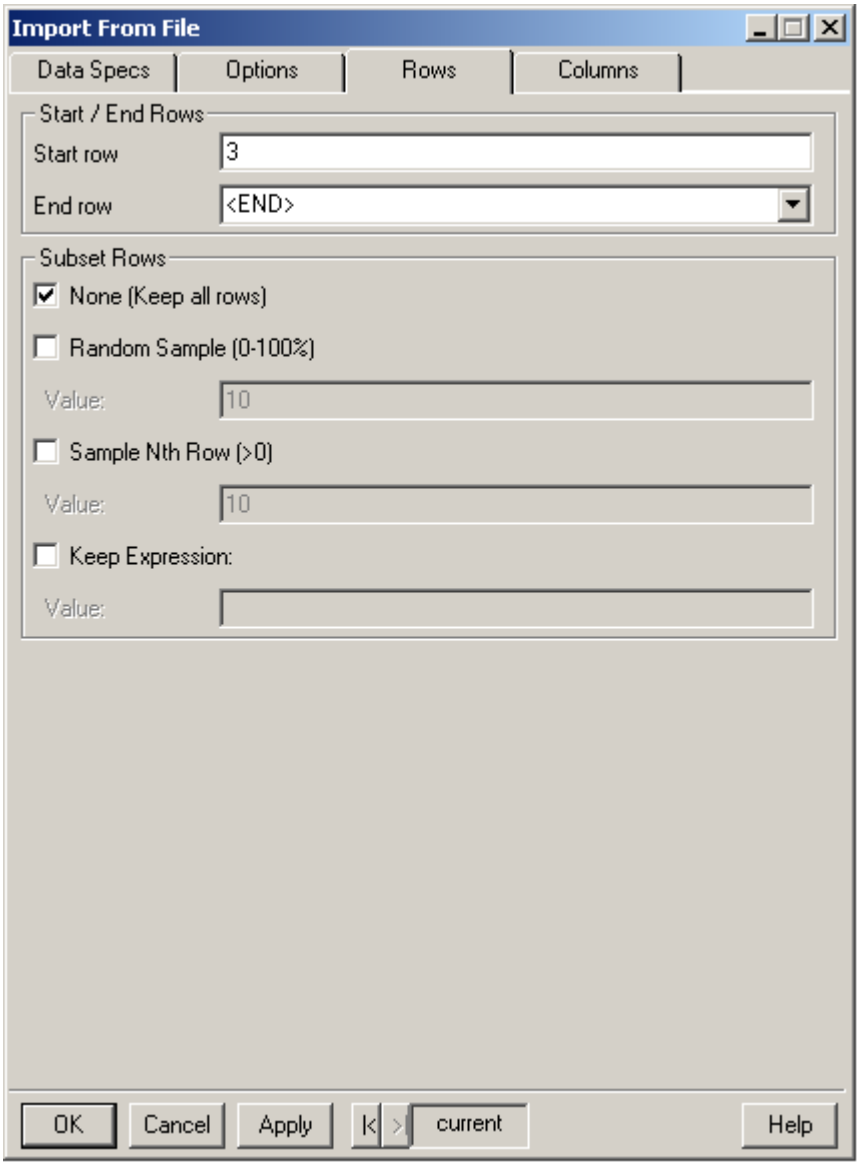
Click on **File | Import Data | From File…** to open the **Import From File** window.



Click the **Browse** button and navigate to the USGS library folder in the **Select file to import** window. Select the MN10Nutrients.xls file and then click **OK**. That will insert the complete pathname in the **File Name:** box, Excel Worksheet (xl?) in the **File Format:** box, and MN10Nutrients in the **Data set** box. Click the **Options** tab to display the following window.



The column names that were set up for S-PLUS are in row 2, so enter 2 into the **Col names row** box. The remark codes should be imported in character format and not as factors, so **Strings as factors** should be unchecked. If your Excel workbook has multiple worksheets, then set **Worksheet number** to the worksheet number (do not use the worksheet name). Click on the **Rows** tab to display the following window.



A value of 3 should be entered in the **Start row** box because the data start in row 3. The default, <END> in the **End row** box, reads to the last row that has any information, like formatting, not just the last row of data. That can be changed to read a specific block of the data. The same options are available under the **Columns** tab, but are not needed for this example. Click **OK** to import the file and close the **Import from File** window.

Convert data by result to data by sample

Sometimes data are stored by result; that is, in a format where some data are repeated in groups, and a column identifies unique values. Consider the example data shown below. The columns STAID and DATES form a group repeated for each value of PARAMETER. The column PARAMETER identifies the unique values in VALUE and RMK. In this case, RMK is a remark code associated with VALUE. For example, on October 13, 1992, at station 12345678, the value for Q was 1300, the value for Kjeldahl was <0.1, and the value for Total.P was 0.20.

qw.stacked.df

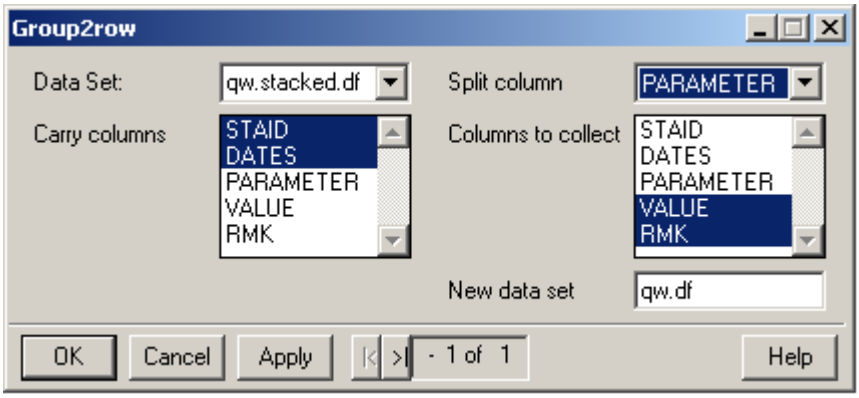
		1	2	3	4	5
		STAID	DATES	PARAMETER	VALUE	RMK
1		12345678	10/13/92 0:00:00 AM	Q	1300.00	
2		12345678	10/13/92 0:00:00 AM	Kjeldahl	0.10	<
3		12345678	10/13/92 0:00:00 AM	Total.P	0.20	
4		12345678	11/12/92 0:00:00 AM	Q	1200.00	
5		12345678	11/12/92 0:00:00 AM	Kjeldahl	0.20	
6		12345678	11/12/92 0:00:00 AM	Total.P	0.20	
7		12345670	10/12/92 0:00:00 AM	Q	125.00	
8		12345670	10/12/92 0:00:00 AM	Kjeldahl	0.20	
9		12345670	10/12/92 0:00:00 AM	Total.P	0.30	

Step 1. Restructure the data

This example uses the *qw.stacked.df* dataset available in the USGS library. It can be copied to the user's chapter by clicking and dragging the dataset in the left pane of the **Object Explorer Window**, or by typing the following call in the **Commands Window**.

```
qw.stacked.df <- qw.stacked.df
```

Click on **USGS | Group2row…** to open the **Group2row** dialog window. Select qw,stacked.df in the **Data Set:** box, which populates the options for the column selection boxes. The **Carry columns** can be any repeating group data. For this example, STAID and DATES are the repeating groups and should be selected as the **Carry columns**. The **Split column** identifies a single column that will be used to identify the unique values in the **Columns to collect**. For this example, PARAMETER should be selected. The **Columns to collect** are the unique values associated with the value in the **Split column.** For this example, VALUE and RMK are the unique values and should be selected. The **New data set** can be any valid S-PLUS object name. For this example, qw.df was entered in the box.



Click OK, which closes the **Group2row** window, creates a new dataset called *qw.df* and displays it in a **Data Window**. The new dataset contains eight columns: STAID, DATES, Q.VALUE, Q.RMK, Kjeldahl.VALUE, Kjeldahl.RMK, Total.P.VALUE, and Total.P.RMK and three rows corresponding to the unique combinations of STAID and DATES.