Expected behavior of importWizard

Jianhua Zhang

May 5, 2024

It takes three steps to import a file using importWizard. There are buttons that allow users to travel back and forth between the three steps (states). When importWizard is invoked, an environment object is created to keep data for each step. Data stored for each step include a list for the formal arguments that will be used to import the file at the end using the function `read.table` and an object of class colInfo with three slots for the name, data type, and retention of data columns of the file to be imported. Each step receives inputs from users until values for those arguments that are needed to import a file are filled. Some of the arguments are given default values that can be modified by users.

Step one: file browsing

When importWizard is invoked, a widget that appears will have the following elements:

- An entry box for the name of the file to be imported. A file with a full path name is expected if users chooses to type in a file name. An alternative is to use the `Browse` but to retrieve a file name.
- A `Browse` button that allows users to browse directories for the name of a file to import. The file name will be placed in the entry box that can be read in using the `Get` button.
- A `Get` button to get the file specified by the file name in the entry box.
- Two radio buttons defining the type of the file to be imported. One button for delimited file and one for fixed width file. Functions for importing fixed width files have not been implemented yet.
- A list box with number for users to choose from for the line number from which import begins.

- A list box to show the content of the file to be imported as lines read in one at a time.

- Four buttons (`Cancel`, `< Back`, `Next >`, `Finish`) to allow users to travel back and forth.

If importWizard is invoked with a file name, it opens that file and then displays the file name in the entry box. One of the two radio buttons will be checked if the system is able to detect if the file is delimited or has fixed width (A file is still considered as delimited if it only has a new line (
`n`) as the delimiter). The content of the file will be displayed as separate lines in the list box. Otherwise, the elements will be empty until users select a file using the `Browse` button, at which time the elements will be populated with relevant data.

When a file is displayed, the system will check to see if the file has a header, a delimiter separating data columns, and the data type for each data column (if any). The results will be used to initiate the values for the argument list. The initial values for the argument list for step one are set using values returned by a function named `guess.sep`. A new line (
`t`) is the default value for the delimiter if the file does not have any other delimiter.

Users are able to choose to import a file starting from a specified row by choosing a number from the list box for `Start import at row`. The default value is one (import from the first row)

After choosing a file and the row number to start importing, clicking the `Next >` button moves to step two. Clicking the `Cancel` button at any time cancels the importing process. The `Finish` button will remain inactivated until step three.

Step two: file examination

The widget for step two has the following widget elements:

- A group of radio buttons for the exact type of delimiter used in the file to be imported. A delimiter that is not listed by the radio buttons can be entered by selecting `Other` radio button and then enter the type of delimiter in the activated entry box beside `Other`.

- A list box for users to select the the type of quote used in the file. Both `"` and `'` are the default value.

- A canvas showing the file in columns if the file is delimited. No function has be implemented to deal with fixed width file yet at this moment.

The initial values for the argument list for step two are the same as the values when step one is exited. Any change by users will be written to the argument list for this step.

Clicking the `< Back` button moves back to step one with the values of the argument list set to that of the last state of step one. After setting the delimiter and quote, Clicking textNext > button moves to step three with the values of the argument list for step three set to the last state of step tow.

Step three: file manipulation

The widget for step three has the following widget elements:

- A `More Args...` button that allows users to select some of the additional arguments to import a file using read.table for advanced users. The values of those arguments are defaulted to some values that are commonly used to import a file. Change the values only when needed.
- A list box with the file to be imported shown in columns if. On top of each column there is a radio button to indicate whether a column should be dropped when the file is imported, an entry box for the name of the column when imported as a data frame, and another entry box for the data type of the column. Clicking the radio button will prevent the column below the button from being imported.

Clicking the `< Back` button moves back to step two. When the `Finish` button is clicked, a message box will pop up to allow users to decide whether to save the imported file as an R object in the global environment (`.GlobalEnv`) by a specified name.

importWizard returns (invisibly) a list with the first element being the arguments used to import the file and the second being the content of the file imported based on the arguments.

Usage

importWizard is driven by the interface. Users only need to type "importWizard()" or "importWizard(filename)" to begin the importing process.

1 Session Information

The version number of R and packages loaded for generating the vignette were:

- R version 4.4.0 Patched (2024-04-24 r86482), `aarch64-apple-darwin20`
- Locale: `C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8`
- Time zone: `America/New_York`

- TZcode source: `internal`
- Running under: `macOS Ventura 13.6.6`
- Matrix products: default
- BLAS: `/Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib`
- LAPACK: `/Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib` ; LAPACK version3.12.0
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Loaded via a namespace (and not attached): compiler 4.4.0, tools 4.4.0