

Preliminary Reference Guide/Manual for the R plugin for Gnumeric

Duncan Temple Lang

June 28, 2002

This is a very early version of a guide to using the R plugin for gnumeric. It is being written as we develop the plugin, so it may read a little awkwardly and disjointed. Please mail comments and suggestions to `omega-devel` mailing list

1 Getting Started

We will assume that you have the plugin installed and available to Gnumeric. If not, please see the installation documentation.

The first thing to do is ensure that the plugin is active. Do this You can customize which file is `source()`'d by R at startup time either via the contents of your `~/.Rprofile` file or by explicitly setting the environment variable `R_PROFILE`. Additionally, you can also set the environment variable `GNUMERIC_R_PROFILE` to specify the fully qualified name of a file which will be read after the usual R initialization.

If the function (or functions it calls) need access to the internals of gnumeric to retrieve or set cell values, format existing cells, create new sheets, etc. it should include a parameter named `.sheet`. Currently this must be the last argument in any call and is not specified by name, but by position.

One can also get a reference to the containing workbook from a sheet reference via the function `getGnumericWorkbook()`. This allows one to create new sheets, compute unique names for new sheets, get references to other sheets by index or name (using the `[]` operator).

2

The R-Gnumeric interface has two aspects to it. The first involves calling R functions from Gnumeric. This is the typical way people will use the interface. The second allows R to access elements within Gnumeric such as the cells in a sheet, the sheet itself and workbooks in which sheets are contained. The first place we start is with calling R functions from Gnumeric. Of course, somebody will have to have made these available to the Gnumeric session before we can call them. But we deal with that later.

3

4 Making R Functions Available to Gnumeric

One “exports” R functions to Gnumeric by calling the function `gnumeric.registerFunction()`. This takes the function object that is to be accessible to Gnumeric and a name by which it is to be known within Gnumeric. For a function that takes no arguments, we need only specify an additional argument that “explains” the function. For example, if we wanted to export the S function `date()`, we might do it with the following R command

```
gnumeric.registerFunction(date, help="get the current date")
```

This should be done before the Gnumeric session is started. To do this, you will need to have the appropriate commands in the R profile that is run when the plugin is initialized.

We need also to specify information about the number and types of arguments this function expects. We do this by giving a string

For example, we might export a *round()* via the command

5 Accessing Cells in a Sheet

One can get the values from different cells.

Hiding rows and columns.

6 Cell Formatting

There are different classes of numbers, such as currency, date, time, percentage, fraction, scientific, text, special, etc. With these, there are different strings.

6.1 Alignment

6.2 Font

Font name/family, style, and size. Color.

Underline, strikethrough sub- and super-script.

6.3 Color & Pattern

6.4 Border

Which lines are drawn and the line style.

6.5 Protection