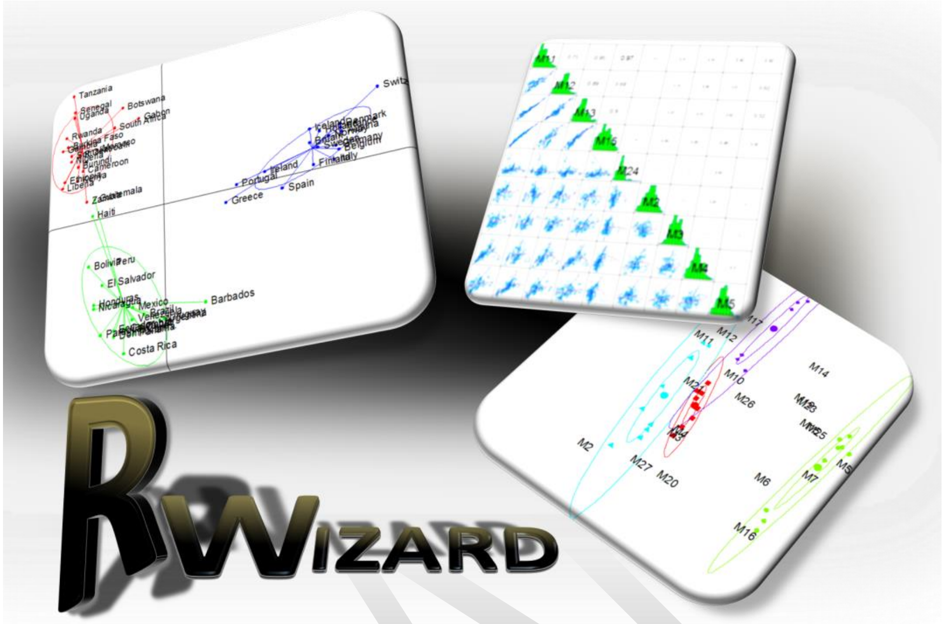


RWizard v.1.0 User's Manual

(Draft)

INDICE

1 Prerequisites

In order to be able to install Rwizard, you will need to have the following software already installed.

<u>Hardware & Operating System</u>

- Microsoft Windows Xp, Vista, 7/8 (32Bits / 64Bits recommended)
- At least 4GB RAM
- Graphic card with a minimum of 256Mb VRAM

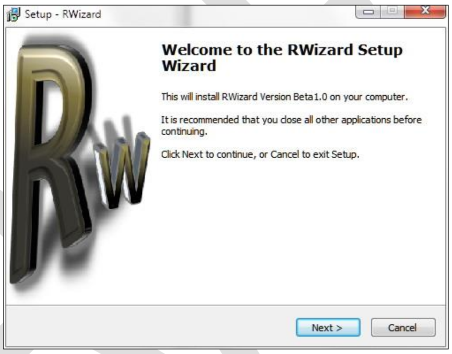
<u>Software</u>

- R Environment for statistical computation (recommended 3.0.1 or higher)
- .NET Framework 4.0
- Microsoft Visual C++ 2010 redistributable package

2 Installation and execution

2.1 Installing Rwizard, step by step

Once the installer for Rwizard has been downloaded, you must execute it and follow the instructions.



Setup - RWizard

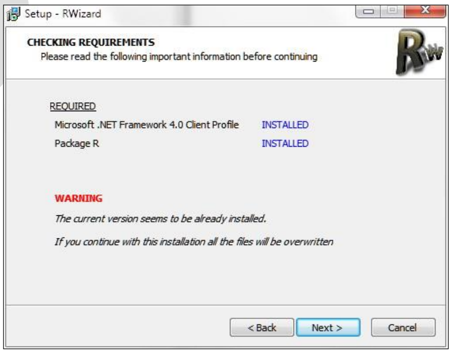
Welcome to the RWizard Setup Wizard

This will install RWizard Version Beta1.0 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

The installer will check whether the ".NET Framework 4.0" and the "R Environment for Statistical Computation" are installed. In case of any of them missing, it will cancel the installation.



Setup - RWizard

CHECKING REQUIREMENTS

Please read the following important information before continuing

<u>REQUIRED</u>

Microsoft .NET Framework 4.0 Client Profile	INSTALLED
Package R	INSTALLED

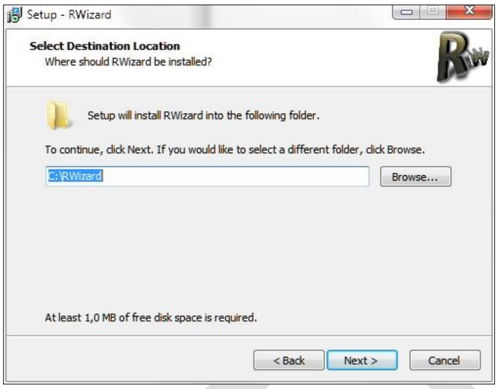
WARNING

The current version seems to be already installed.

If you continue with this installation all the files will be overwritten

The installer will also check if RWizard is already installed, in a previous or current version. In this case, you will not be able to change the target installation folder, since the same one it was installed to must be used.

The next step will consist of selecting the target folder (provided no other installations of RWizard were found).



Setup - RWizard

Select Destination Location

Where should RWizard be installed?

Setup will install RWizard into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

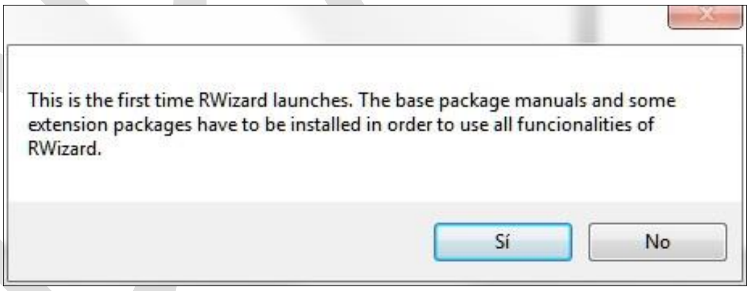
C:\RWizard

At least 1,0 MB of free disk space is required.

Finally, the installer will copy all the needed files and proceed with all settings, and then RWizard will be ready for its first execution.

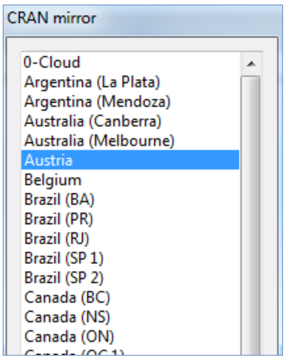
2.2 First time execution

The first time that RWizard is executed, it will need to configure and install various R packages, for which the supervision of the user is needed. Specifically, the software will ask for permission about downloading and installing R packages which are central for RWizard. Otherwise, its functionality will be severely limited.



This is the first time RWizard launches. The base package manuals and some extension packages have to be installed in order to use all funcionalities of RWizard.

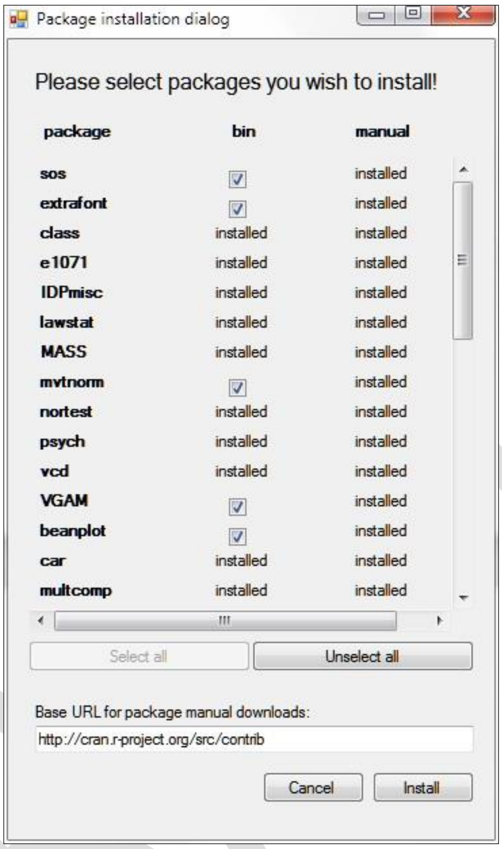
IMPORTANT. *You have to select an appropriate mirror considering the country from which you are downloading the packages. AUSTRIA is, for example, a general good choice.*



CRAN mirror

0-Cloud
Argentina (La Plata)
Argentina (Mendoza)
Australia (Canberra)
Australia (Melbourne)
Austria
Belgium
Brazil (BA)
Brazil (PR)
Brazil (RJ)
Brazil (SP 1)
Brazil (SP 2)
Canada (BC)
Canada (NS)
Canada (ON)

A package listing will appear. In this list, all the needed packages for RWizard are shown. Some of them can be already installed, while the others are pre-selected for installation. Either the binary will be needed, either its manual. The latter is needed for one of the RWizard's most advanced capabilities: the function editor. The recommended action is to click "Install" with all packages selected.



Please select packages you wish to install!

package	bin	manual
sos	☑	installed
extrafont	☑	installed
class	installed	installed
e1071	installed	installed
IDPmisc	installed	installed
lawstat	installed	installed
MASS	installed	installed
mvtnorm	☑	installed
nortest	installed	installed
psych	installed	installed
vcd	installed	installed
VGAM	☑	installed
beanplot	☑	installed
car	installed	installed
multcomp	installed	installed

Base URL for package manual downloads: http://cran.r-project.org/src/contrib

IMPORTANT. *The installation of all these packages requires a lot of time, which can be reduced depending on the net connection quality and the power of the computer.*

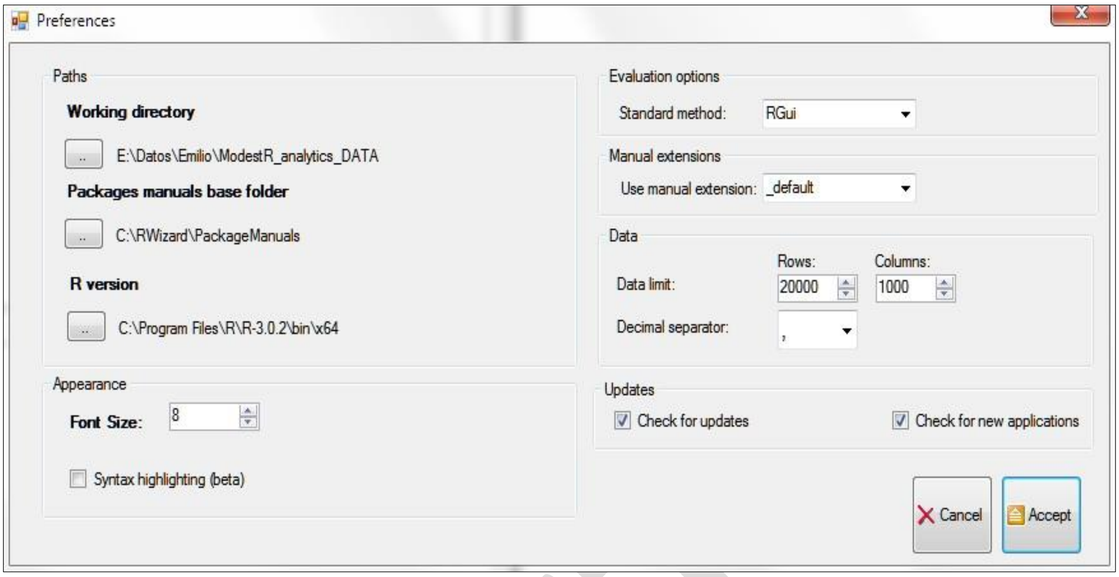
Please wait ...

Installing package sos

2.3 Preferences

Once the installation of packages is finished, you will need to correctly configure the settings of RWizard, selecting the option *preferences* in the main menu (Figure 1).

- *Paths*.- Settings for the paths giving access to the packages and the working directory.
  - *Working directory*.- Default folder in which the created scripts will be saved, and also from where output files of scripts will be loaded and save.
  - *Packages manuals base folder*.- Default folder for the base packages installed by RWizard.



Preferences

Paths

Working directory

.. E:\Datos\Emilio\ModestR_analytics_DATA

Packages manuals base folder

.. C:\RWizard\PackageManuals

R version

.. C:\Program Files\R\R-3.0.2\bin\x64

Appearance

Font Size: 8

□ Syntax highlighting (beta)

Evaluation options

Standard method: RGui

Manual extensions

Use manual extension: _default

Data

Data limit: Rows: 20000 Columns: 1000

Decimal separator: ,

Updates

☑ Check for updates ☑ Check for new applications

Figure 1 Preferences

  - *R version*.- Path to the folder in which the R Statistical Environment to be used by RWizard is installed.
- *Appearance*.- Allows to change the font size for the editor of scripts.
  - *Syntax highlighting* (Beta).- The editor highlights reserved words in the script.

- *Evaluation options*.- Sets how to compile and execute scripts in RWizard.
  - *Eager*.- Sends the script to R without any parsing. Take into account that in R, the *Replies* windows will show the result of the last line of the script, though not the console errors.
  - *Parse*.- Sends the script line by line to R. Again, note that in R, the *Replies* windows will show the result of each line of the script, though not the console errors.
  - *RGui*.- Sends each line to R as a single script, allowing to check in the *replies* window of R a result for each line, as well as the console errors.

- *Manual extensions*.- Allows to select which of the manual extensions to use: the default one, or the manual provided by RWizard, which is more advanced.
  - "default" contains the descriptions of arguments of the base manual, without modifications.
  - "RWizard" contains the improved descriptions and functionalities for some arguments, besides the base descriptions for unmodified arguments.

- *Data*.- Sets data loading and visualization in the "Data" work area.

  IMPORTANT.- *All RWizard applications work with CSV files only.*

  - *Data limit*.- Limit of rows and columns that are shown in the data panel of the "Data" work panel, or in the data panel of the function editor (minimum 100 rows and columns). The data panels will always use the given number of rows and columns for any file, even if the true rows or columns number is higher than the selected one. The file is always entirely loaded, anyway.
  - *Decimal separator*.- Numerical separator used for the CSV data file loading.

- *Updates*.- Allows to check the available updates for applications each time RWizard is launched.
    - ➢ Check for updates.- reports whether there are updates for RWizard and the installed applications.
    - ➢ Check for new applications.- reports whether there are new applications available for RWizard.

2.4 Packages

RWizard requires various R packages installed in order to be able to work correctly. It is important, once the preferences have been correctly set, to check out that the needed packages "*base*" and "*extension*" have been installed. This can be done by means of the *Packages* entry of the main menu, which will open the package management window (Figure 2). From here you will be able to install new packages, through CRAN or local files, as well as other related functionalities: installation of manuals for new or already installed packages, detection of missing packages needed for the current script, deactivation of loaded packages for its updating, and finally verification of the correct installation of needed packages for the right working of RWizard.

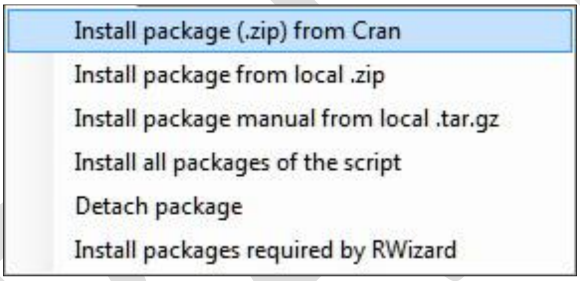

Install package (.zip) from Cran
Install package from local .zip
Install package manual from local .tar.gz
Install all packages of the script
Detach package
Install packages required by RWizard

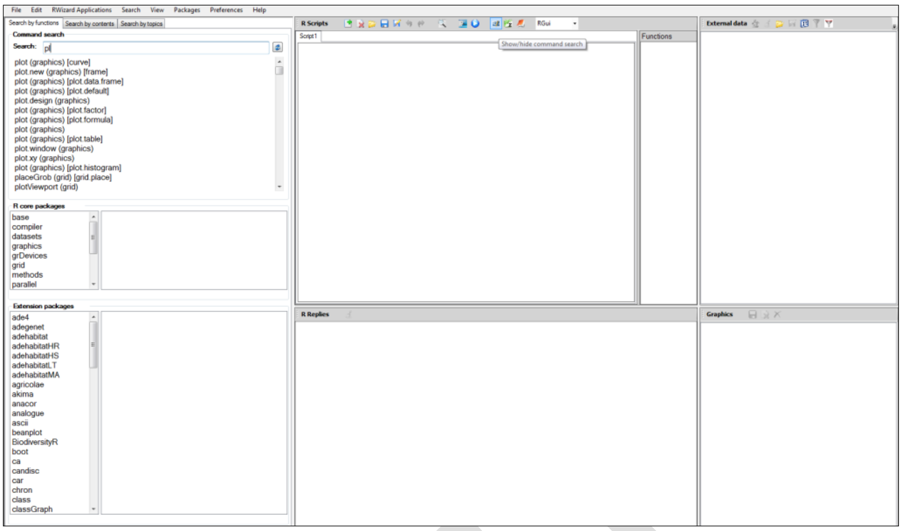
Figure 2. Packages menu

a) *Install package (.zip) from CRAN*.- Makes it possible to install a package from the CRAN of R. The first step is to select the country for the download of desired package, though it is possible for this step to be automatically skipped in case the same selection has been already carried out in the very same session. Next, a listing of the available R packages will be shown, in which you can select one for installation.

***Note**. - This process does only install the package (.zip) in R, making it possible to use it from RWizard, though does not install its corresponding manual. This means that the information in the manual will not appear in the "Comand Search" option, "Search by functions" tab, "Extension packages" section or from the Function Editor. You will have to follow the installation instructions detailed in the sections c) or d) below.*

b) *Install package from local .zip*.- Allows to install an R package from a local file, making it available from RWizard as well. Note that however, as explained in the previous point, the manual will not be installed.

c) *Install package manual from local .tar.gz.*- This option will install an R package manual from a local file, making its information available from the Function Editor, and from the “Comand Search” option, “Search by functions” tab, “Extension packages” section (as shown in Figure 3). This manual will give you a more detailed view of functions and its arguments, as well as related examples and explanations. You will have to previously download the *.tar.gz* file from CRAN (*http://cran.r-project.org/web/packages/available_packages_by_name.html*).



Search: pl

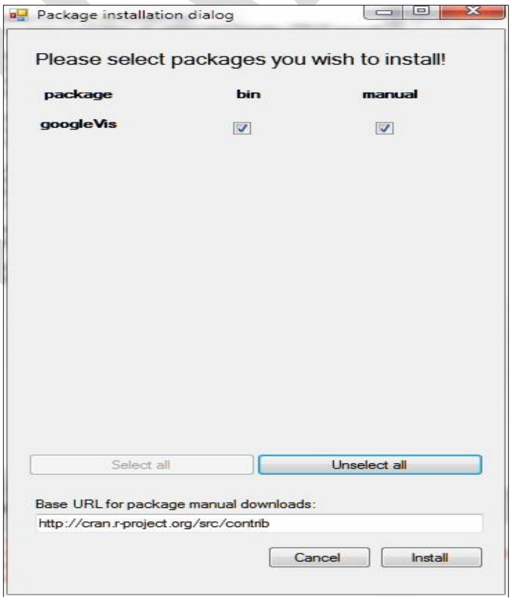
```
plot (graphics) [curve]
plot.new (graphics) [frame]
plot (graphics) [plot.data.frame]
plot (graphics) [plot.default]
plot.design (graphics)
plot (graphics) [plot.factor]
plot (graphics) [plot.formula]
plot (graphics)
plot (graphics) [plot.table]
plot.window (graphics)
plot.xy (graphics)
plot (graphics) [plot.histogram]
placeGrob (grid) [grid.place]
plotViewport (grid)
```

```
base
compiler
datasets
graphics
grDevices
grid
methods
parallel
```

```
ade4
adegenet
adehabitat
adehabitatHR
adehabitatHS
adehabitatLT
adehabitatMA
agricolae
akima
anacor
analogue
ascii
beanplot
BiodiversityR
boot
ca
candisc
car
chron
class
classGraph
```

Figure 3 Work area for function searching in manuals of installed packages, installed from .tar.gz files

d) *Install all packages of the script*.- Will install those missing packages in R for the current script (Figure 4). The needed packages will be identified from the script itself, looking for “*library*(package_name)” or "*require(package_name)*" sentences. Also note that the needed manuals are also detected, as they appear grayed in the Functions work area.



Please select packages you wish to install!

package	bin	manual
googleVis	☑	☑

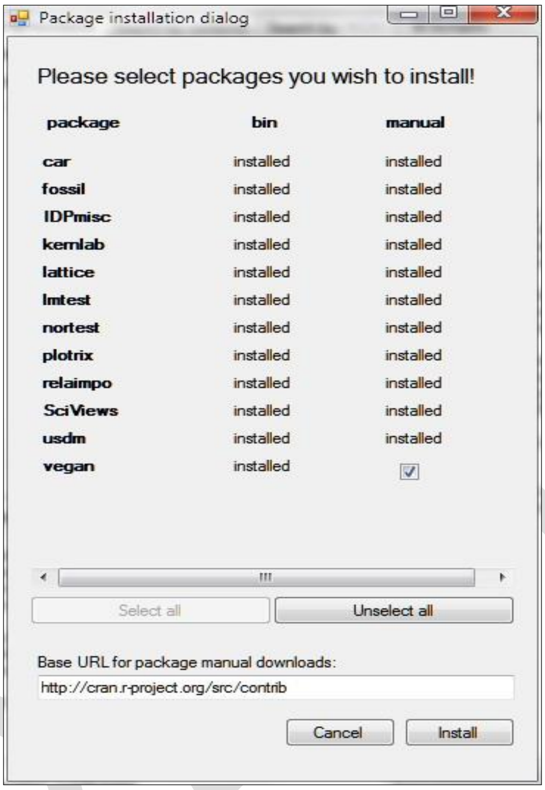
Base URL for package manual downloads:
http://cran.r-project.org/src/contrib

Figure 4 Install all packages of the script

e) *Detach package*.- It allows to disable a given package so it is possible to install its update.

f) *Install packages required by RWizard*.- Carries out a verification of the required packages and manuals for the correct working of Rwizard. In case any of them is missing, the package managent window will be launched (Figure 5). Note that in some rare cases, packages with names containing dots (such as “rpart.plot”) are not recognized as installed when they are.

Note*.- You are recommended to run this check after the first installation of the application.*



Package installation dialog

Please select packages you wish to install!

package	bin	manual
car	installed	installed
fossil	installed	installed
IDPmisc	installed	installed
kernlab	installed	installed
lattice	installed	installed
lmtest	installed	installed
nortest	installed	installed
plotrix	installed	installed
relaimpo	installed	installed
SciViews	installed	installed
usdm	installed	installed
vegan	installed	☑

Base URL for package manual downloads: http://cran.r-project.org/src/contrib

Figure 5 Install package required by RWizard

3 RWizard environment

RWizard is a tool easing the access to R, and therefore opens for users a broad spectrum of statistic techniques, with an ever growing offer.

We have developed a GUI that helps using the statistical and graphic programming language of R, so access to functions will be friendly and easy, while retaining all of their potential.

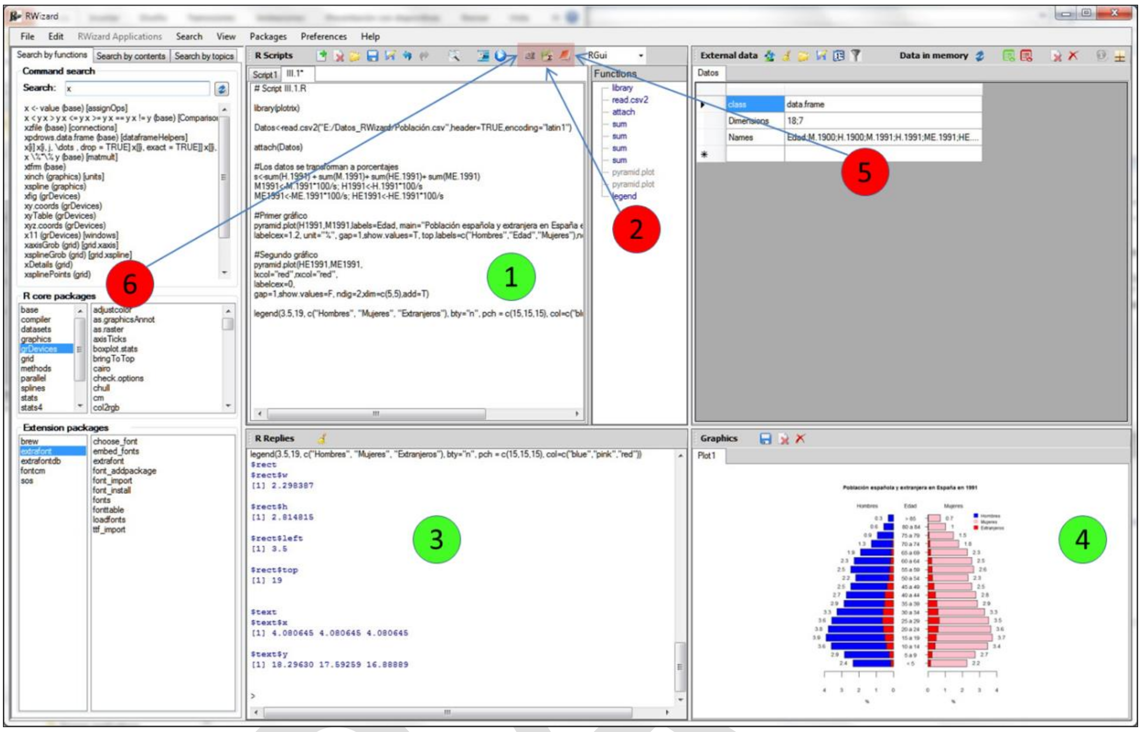
You will be able to create your own R scripts or import other previously created scripts. You will be able to access functions called within the script, detect required packages or install new packages (directly downloading them from the web).

RWizard comes with some applications developed sepcifically for this environment. These developments will offer designed scripts, with their user manuals ready, for multiple fields in the industry (please check the RWizard applications section).

3.1 Work areas in the environment

Work areas can be divided in two categories (Figure 6)

- Fixed work areas.- They are always visible (1-3-4)
- Hidable work areas.- They can be shown or hide (2-5-6).



```
# Script III.1.R

library(plotrix)

Datos<-read.csv2("E:/Datos_RWizard/Población.csv",header=TRUE,encoding="latin1")

attach(Datos)

#Los datos se transforman a porcentajes
s<-sum(H.1991) + sum(M.1991)+ sum(HE.1991)+ sum(ME.1991)
M1991<-M.1991*100/s; H1991<-H.1991*100/s
ME1991<-ME.1991*100/s; HE1991<-HE.1991*100/s

#Primer gráfico
pyramid.plot(H1991,M1991,labels=Edad, main="Población española y extranjera en España e
labelcex=1.2, unit="%", gap=1,show.values=T, top.labels=c("Hombres","Edad","Mujeres"),n

#Segundo gráfico
pyramid.plot(HE1991,ME1991,
lxcol="red",rxcol="red",
labelcex=0,
gap=1,show.values=F, ndig=2,xlim=c(5,5),add=T)

legend(3.5,19, c("Hombres", "Mujeres", "Extranjeros"), bty="n", pch = c(15,15,15), col=c("bl
```

```
legend(3.5,19, c("Hombres", "Mujeres", "Extranjeros"), bty="n", pch = c(15,15,15), col=c("blue","pink","red"))
$rect
$rect$w
[1] 2.298387

$rect$h
[1] 2.814815

$rect$left
[1] 3.5

$rect$top
[1] 19


$text
$text$x
[1] 4.080645 4.080645 4.080645

$text$y
[1] 18.29630 17.59259 16.88889


>
```

▸	class	data.frame
	Dimensions	18;7
	Names	Edad;M.1900;H.1900;M.1991;H.1991;ME.1991;HE...
*		

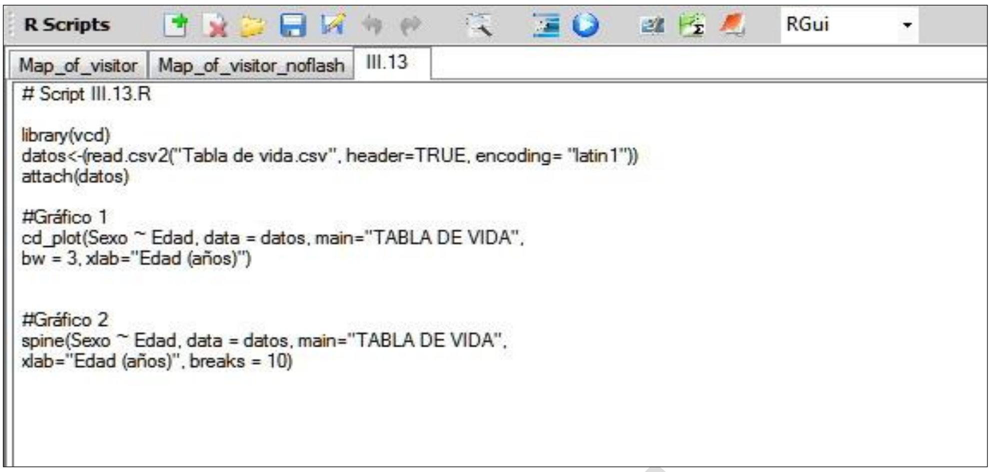
Figure 6 Work areas

1) Script creation and management.
2) List of functions used in the current script.
3) Results of the commands executed by scripts.
4) Graphs generated by scripts.
5) Information related to the generated data type and external data loaded.
    a. *External Data*.- Management of the data directly loaded in the script.
    b. *Data in Memory*.- Shows information of the data generated in a script when it is executed, of type *dataframe*, *matrix* o *list*.
6) Browse information related to functions, or search functions in R.

3.2 "*Scripts*" work area

This work area is used for executing and editing R scripts (Figure 7). After an execution, the results will be shown in the output work areas, such as "*R Replies*", "*Graphics*", "*Data*" o "*Functions*", as described in the following sections.

R code can be directly written in the script editor, loaded from disk, or add a call to an existing function comfortably: go to the "*command search*" work area, (section 3.7), "*search by functions*" tab, and write the name of the function to use (you will need the manual of the corresponding package installed, as explained in section 2.4Packages). When the function is double-clicked, the function editor will open, allowing you to enter the needed values for the arguments. By clicking on "Accept", the function call will be added to the script.



```
# Script III.13.R

library(vcd)
datos<-(read.csv2("Tabla de vida.csv", header=TRUE, encoding= "latin1"))
attach(datos)

#Gráfico 1
cd_plot(Sexo ~ Edad, data = datos, main="TABLA DE VIDA",
bw = 3, xlab="Edad (años)")


#Gráfico 2
spine(Sexo ~ Edad, data = datos, main="TABLA DE VIDA",
xlab="Edad (años)", breaks = 10)
```

Figure 7 "Script" work area

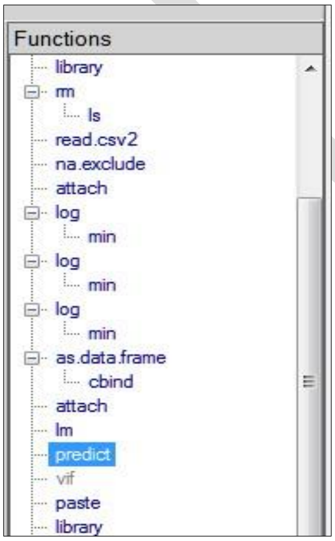
You can work with more than one script at the same time, each one in its corresponding tab, and use the data generated by one of the scripts in another one.

In order to execute a script, you can choose among a range of options: clicking on ▶ “*Execute script*” with the cursor over a line of the script, will execute that line; clicking on the button ☰ “*Select all*”, and then on button ▶, the entire script will be executed.

As code is added to the current script, it is possible to go to the work area “functions” (show it if hidden by clicking Σ), and a listing (in tree form) with all used functions will be displayed and dynamically updated. By double-clicking it the corresponding function editor will open (4.2Function Editor), allowing any change in them.

3.3 “*Functions*” work area

Nombre	Descripción	Imagen	Localización
“Functions”	Hides/shows the work area	Σ	Main tool bar



```
Functions
  library
  rm
    ls
  read.csv2
  na.exclude
  attach
  log
    min
  log
    min
  log
    min
  as.data.frame
    cbind
  attach
  lm
  predict
  vif
  paste
  library
```

Figure 8 "Functions" work area

The “*Functions*” work area is visible by default (Figure 8). When hidden, click on the Σ (Show/Hide Function tree) from the tool bar in order to make it visible, or from the main menu in *View >> Function tree*.

In order to view a listing of the functions that are used by a script, just move the cursor over the “*Functions*” work area, and it will be automatically updated for the current code (this action must be repeated each time the code is changed).

From this work area, by double-clicking a given function, its function editor will open (see section 4.2Function Editor), allowing to modify its arguments.

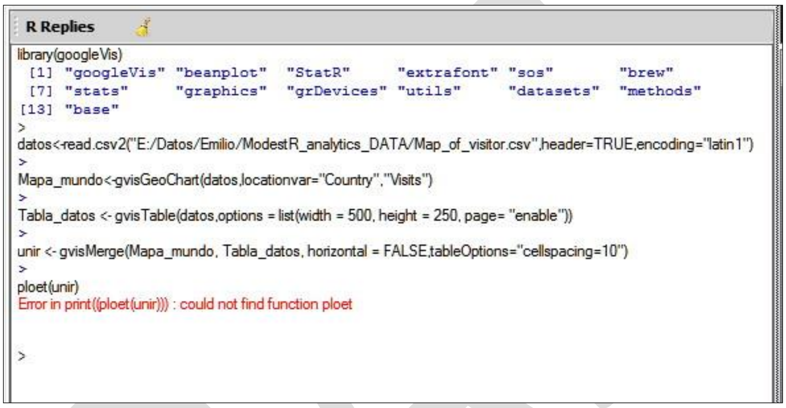
Functions in this list will be shown in blue or gray colour, meaning respectively whether they have their manual already installed or not (in which case the function editor will not be available). The function editor also gives access to help texts for the selected function.

In order to install the corresponding manual for those functions that do not have one, select from the main menu “*Packages*”-“*Install all packages of the script“*(check section 2.4Packages, option d).

NOTE.- *There are certain functions which appear in gray, meaning it will not be possible to access their manual. This is due to non-standard manuals which RWizard is unable to interpret. Its manual could even be shown as already installed.*

3.4 “*R Replies*” work area

The “R Replies” work area (Figure 9) shows the result of execution for each line of the current script, when executed.



```
library(googleVis)
  [1] "googleVis" "beanplot"  "StatR"     "extrafont" "sos"       "brew"
  [7] "stats"     "graphics"  "grDevices" "utils"     "datasets"  "methods"
 [13] "base"
>
datos<-read.csv2("E:/Datos/Emilio/ModestR_analytics_DATA/Map_of_visitor.csv",header=TRUE,encoding="latin1")
>
Mapa_mundo<-gvisGeoChart(datos,locationvar="Country","Visits")
>
Tabla_datos <- gvisTable(datos,options = list(width = 500, height = 250, page= "enable"))
>
unir <- gvisMerge(Mapa_mundo, Tabla_datos, horizontal = FALSE,tableOptions="cellspacing=10")
>
ploet(unir)
Error in print((ploet(unir))) : could not find function ploet


>
```

Figure 9 "R Replies" work area

The clean button 🧹 allows to erase the contents of the “*R Replies*” work area entirely. Errors in the output will be highlighted in red colour.

3.5 “*Graphics*” work area

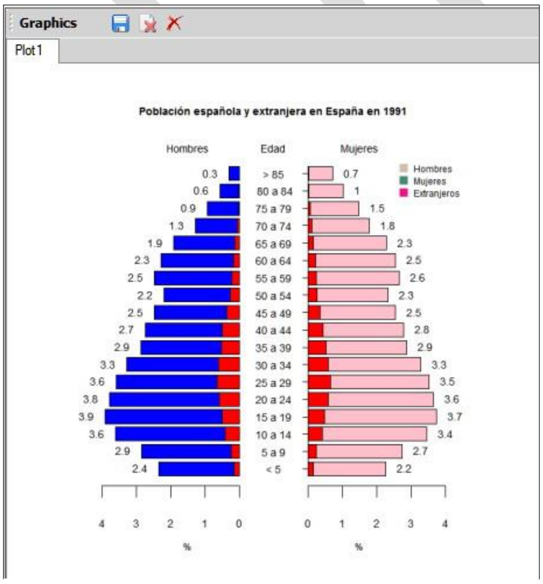


Figure 10 "Graphics" work area

This work area (Figure 10) shows those graphs created by scripts after their execution (these graphs are already shown in an extern R Graphics window).

For each of these graphs, a tab named “Plot” will be created, and using the toolbar of this work area you can:

- 💾 (*Save plot*) save the current graph individually.
- ☒ (*Close current*) close the current graph individually.
- ✗ (*Close all*) close all graphs.

The way to faithfully copy the graph in to a Word document or similar, you must go to the external windows for the graph, (figura 11), and by right-click it, select the "copy as metafile" option, and then go to the target application and select "paste".

If the graph uses colors with some transparency, then they will not be shown with the "copy as metafile" option, and you will have to use the "copy as bitmap" option instead.

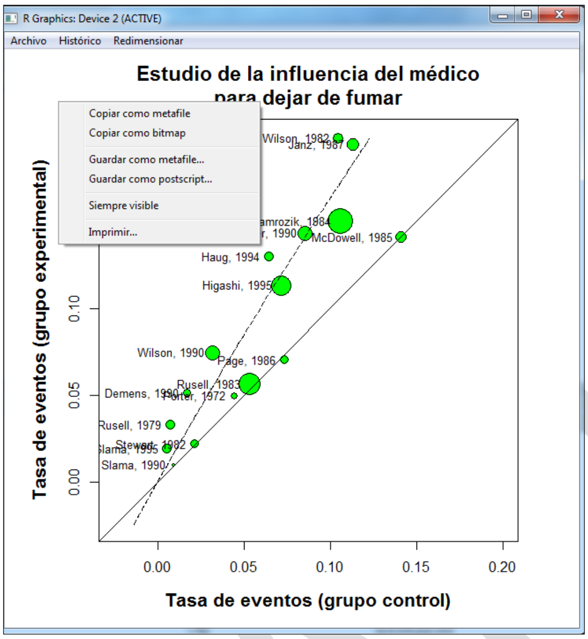
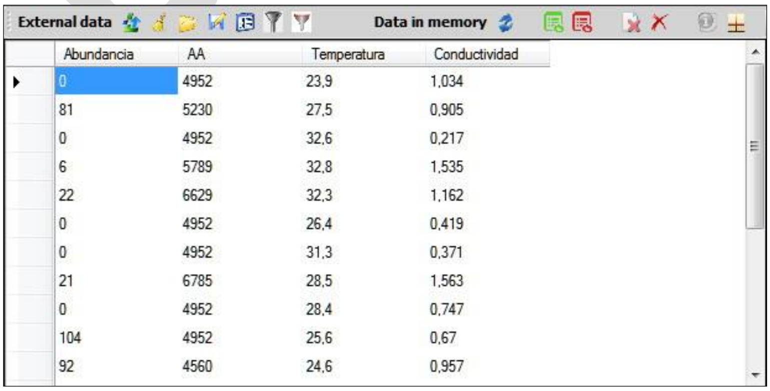


Figure 11 External windows R Graphics

3.6 *"Data"* work area

Nombre	Descripción	Imagen	Localización
"Data"	Hides/shows the work area	📝	Main tool bar

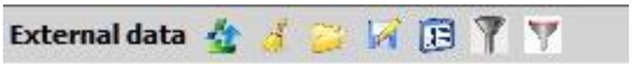
The *"Data"* work area is visible by default (Figure 12**¡Error! No se encuentra el origen de la referencia.**). When hidden, click on the notepad 📝 button (Show/Hide Data) from the tool bar to make it visible, or select the *"View"*-*"Data window"* main menu.



Abundancia	AA	Temperatura	Conductividad
0	4952	23,9	1,034
81	5230	27,5	0,905
0	4952	32,6	0,217
6	5789	32,8	1,535
22	6629	32,3	1,162
0	4952	26,4	0,419
0	4952	31,3	0,371
21	6785	28,5	1,563
0	4952	28,4	0,747
104	4952	25,6	0,67
92	4560	24,6	0,957

Figure 12 "Data" work area

This work area is divided in two sections:

- 

In this section you can load external data and add them to the corresponding code in the script editor.

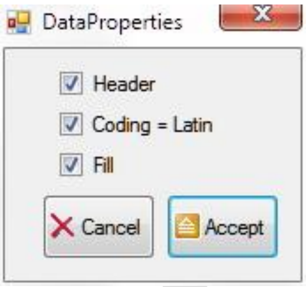
IMPORTANT.- That external data already added to the script cannot be managed from within this work area, since it simply recognizes the data of the last file loaded from within it.

Once the file has been loaded, it is also possible to directly modify the data in the table cells.

NOTE.- These modifications will be effective when the script is executed, provided the file is saved under the same name as it appears in the source code. Otherwise, the modifications will not be taken into account.

“External Data” menu options:

➢ (*Data properties*)


DataProperties

☑ Header
☑ Coding = Latin
☑ Fill

Allows to predefine the parameters for the function “read.csv2” o “read.csv” (depending on the working language) before it is added to the script. For example:

```
data<-read.csv2(      "F:/ANCOVA.csv",
                      header = TRUE,
                      Fill = FALSE,
                      encoding = "latin1" )
```

➢ (*Load data*)

Load CSV data. The numerical separator used for this data is set on the “*Preferences*” window (check 2.3).

Each time a data file is loaded, it will be available for modification until another file is opened.

➢ (*Save as)*

Makes it possible to save the loaded data using another file name.

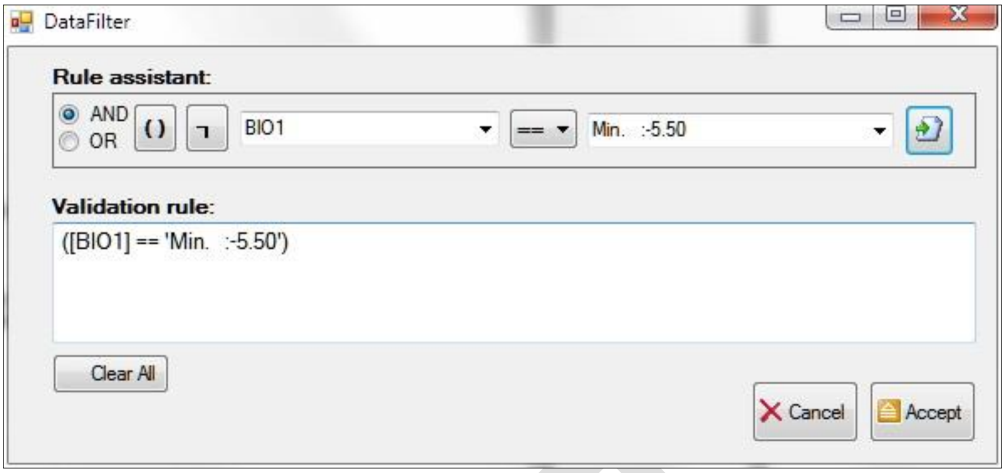
NOTE.- The source code of the script which uses the data with its original file name is not modified.

➢ (*Clean data grid*)

Erases data shown in this panel.

➢ (*Filter*)

Opens the “DataFilter” window (Figure 13), making it possible to select which data to be shown by building conditions. Once “Accept” is clicked, the function “subset” will be added to the current script using the conditions as parameters. Also the result of the filter is reflected on the work area, showing only the data that fulfills the conditions.



([BIO1] == 'Min. :-5.50')

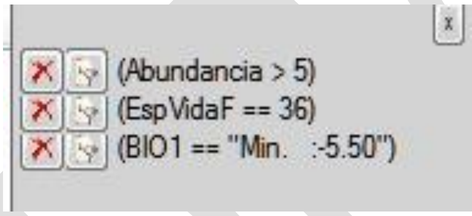
Figure 13 Data Filter window

For example:

```
data<-subset(data,(BIO1 == "Min. :-5.50"))
```

***NOTE**.- If you want to select more than one category for a given column, you must select “OR” to combine conditions. Otherwise, for categories in different columns, you must choose “AND”.*

- (*Delete/Edit filter*)



(Abundancia > 5)
(EspVidaF == 36)
(BIO1 == "Min. :-5.50")

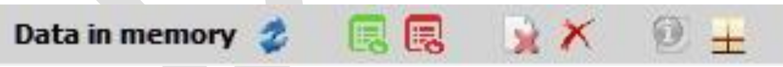
By means of this button you can manage all filters in the current script.

Eliminate the filter.

Modify the filter.

- Bring to front

Shows in the “External Data” section the last loaded file.

- 

This tool will only show those variables which are of the *matrix, list* or *dataframe* types generated after the execution of the script. They will be available until they explicitly erased.

***NOTE**.- These variables can be used in scripts while are not erased. In case of two variables with the same name, only the last one is available.*

- (*Refresh variables*)

Shows variables loaded in memory, provided they are of the *matrix, list* or *dataframe* types. For this button to be enabled, at least one script must be executed. This button will be disabled when all variables in memory are erased.

- (*Hide data*)

  Hides the variable in current tab.

- (*Hide all data*)

  Hides of variables in memory.

- (*Remove data*)

  Erases from memory the variable in current tab.

- (*Remove all data*)

  Erases all variables stored in memory.

- (*Basic information*)

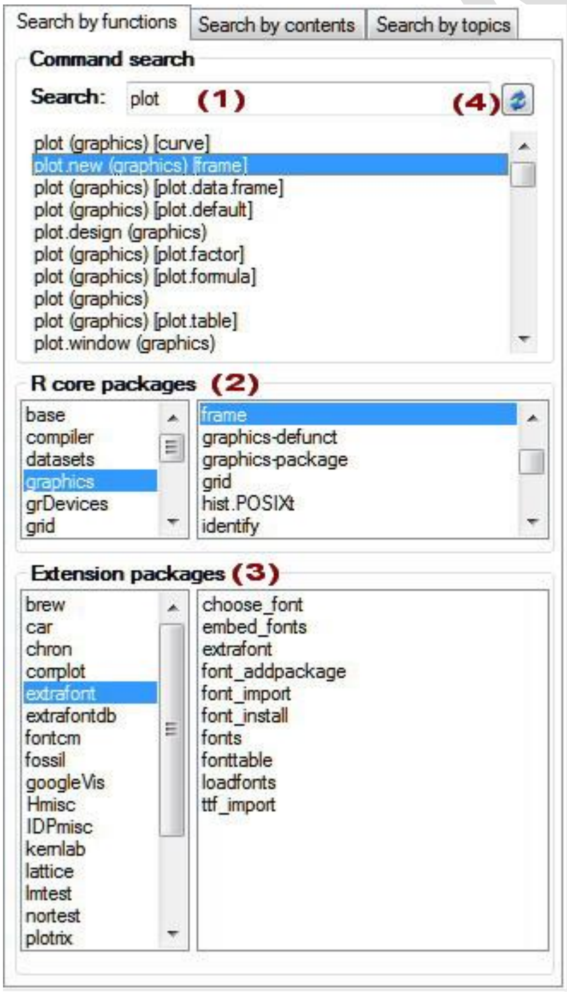
  Shows basic information, in excel format, about the variable in current tab.

- (*Summary*)

  Shows more detailed information about the variable in current tab, in text format.

3.7 *“Command Search”* work area

Nombre	Descripción	Imagen	Localización
“Command Search”	Hides/Shows work area		Main tool bar



Search: plot

Figure 14 "Search by functions" window

The *“Command Search”* work area allows to search information related to the R functions and packages, and easily browse those functions, even before the installation of the manuals for each package has been done (check 2.4Packages).

In this work area the following tabs are available:

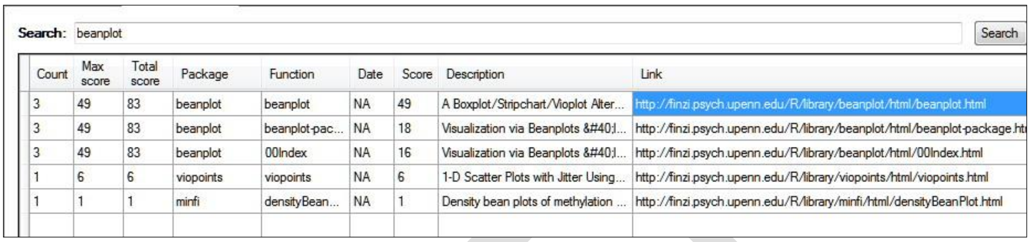
a) *Search by functions*: You can look up a function (Figure 14) by name by typing it in the *“Search”* (1) text box in order to browse its manual, provided it has been installed in the RWizard environment (listed in “R core packages” (2) or in “Extensión Packages” (3)).

***Note**.- R packages can be installed but missing the corresponding manual needed by Rwizard, avoiding the possibility to locate the target function. You have to install the manual to enable this (check section 2.4Packages)*

Once the target function is found, and after double-clicking it, the function editor will open, allowing to set the values for its arguments, and add the resulting function call to the current script (after clicking “Accept”).

In case the manual is not found, you can try clicking “*Refresh packages*” (4), which will update the package list. This is useful when the installation of the manual fails.

b) *Search by contents*: performs a contents-based search (Figure 15). For example, searching the name of a function will report all occurrences of that name, being the function itself, the references to that function in the description of other functions... in the example the ocurrences of *“beanplot”* are shown. Clicking the *“Link”* column a web browser will open the listed URL.



Search: beanplot

Count	Max score	Total score	Package	Function	Date	Score	Description	Link
3	49	83	beanplot	beanplot	NA	49	A Boxplot/Stripchart/Vioplot Alter...	http://finzi.psych.upenn.edu/R/library/beanplot/html/beanplot.html
3	49	83	beanplot	beanplot-pac...	NA	18	Visualization via Beanplots &#40;l...	http://finzi.psych.upenn.edu/R/library/beanplot/html/beanplot-package.ht
3	49	83	beanplot	00Index	NA	16	Visualization via Beanplots &#40;l...	http://finzi.psych.upenn.edu/R/library/beanplot/html/00Index.html
1	6	6	viopoints	viopoints	NA	6	1-D Scatter Plots with Jitter Using...	http://finzi.psych.upenn.edu/R/library/viopoints/html/viopoints.html
1	1	1	minfi	densityBean...	NA	1	Density bean plots of methylation ...	http://finzi.psych.upenn.edu/R/library/minfi/html/densityBeanPlot.html

Figure 15 "Search by contents” window

c) *Search by topics*: performs a search in forums specifically configured by default in RWizard. You can add more forums of your interest, as well as erase existing ones.

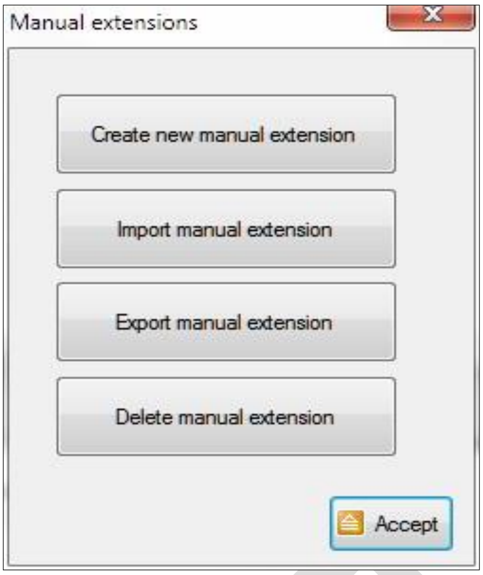
4 Advanced functionality

4.1 Extension manual

This will allow you to comfortably predefine the default argument types for functions, swifting the management of data for those arguments.

RWizard is released with a default “Manual Extension”, which is called *RWizard*. This manual extension is already activated in the configuration options which can be reached from the *“Preferences”* option in the main menu. Argument types of common use have been programmed in this extension, such as color arguments (offering a color palette instead of leaving to the user the burden of writing a color code), font arguments (offering a font dialog), file data selection, etc.

Any change done to a default argument type will always saved to the current manual extensions, which is set in the “*Preferences*” option, unless the “***_default***” extension is set. This is the most basic manual extension, which does not have definitions and does not allow modifications.



Manual extensions

Create new manual extension

Import manual extension

Export manual extension

Delete manual extension

Accept

Figure 16 Manual Extensions menu

In order to create a new manual extension, you must go the main menu and select the option "File"-"*Manual Extensions*" (Figure 16).

- *Create new manual extensión*.- Creates a new manual extension from scratch, or from an existing one.
- *Import manual extension*.- Import a manual extension created by other users.
- *Export manual extensión*.- Export any of the created manuals in order for other users to be able to import them.
- *Delete manual extensión*.- It permanently erases a created manual extension.

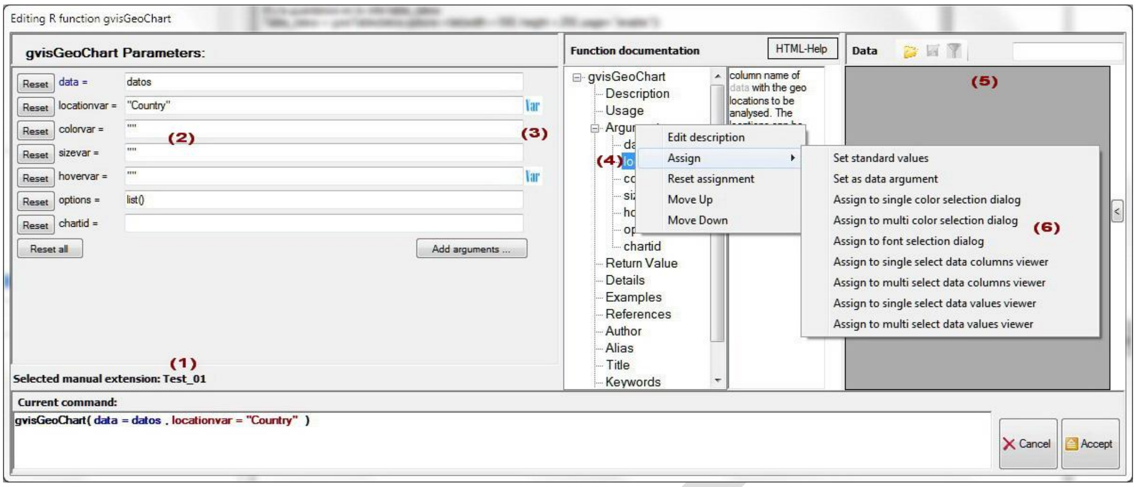
Once a new Manual extension has been created, you must set it as the current manual extension in the "*Preferences*" option of the main menu (check section 2.3).

In order to add or modify a new argument type, you must open the function editor for a function containing an argument which fits that argument type. This can either be done from a script that uses that function (double-clicking it in the function list will open the function editor), or either through the "*command search*".

Once in the function editor (Figure 17), either using the RWizard's own manual extension, either a customized one (1), you can check that some arguments (2) will have assigned icons (3) corresponding with their types.

IMPORTANT.- *Many of the default values will only contain data in case the corresponding data file has been previously loaded. In oder to load a data file, you must use the area at the right hand of the function editor (5). If the data does not appear, you will have to click on "open", choose the data file, and then click "Accept". Finally, you will have to close and reopen the function editor to see the updated data. When data is loaded in the editor using this procedure, a call to the corresponding data read is automatically added to the script.*

In order to access to the default values for a parameter which has a parameter type assigned, just click the corresponding icon (3) and a dialog will open, allowing you to choose the correct value. Remember that, for those values depending on data previously loaded, that data must already appear loaded in the data area (6).



Editing R function gvisGeoChart

gvisGeoChart Parameters:

data = datos
locationvar = "Country"
colorvar = ""
sizevar = ""
hovervar = ""
options = list()
chartid =

Selected manual extension: Test_01

Current command:

```
gvisGeoChart( data = datos , locationvar = "Country"  )
```

column name of data with the geo locations to be analysed. The

Figure 17 Predefine the default argument types for functions

From the central area, "*Function documentation*" we can see a summary of the function call being edited, and access each one of the parameters used. Specifically, you can expand the "*Arguments*" section, in oder to show all the arguments of the function. Various actions can be performed (6) by right-clicking the argument (4). The result of these actions (detailed below) will be stored in the current manual extension (unless the **_default** one is set, which does not allow any modification).

The available options (***remember that in order to have them reflected in the function editor, you will have to click "Accept" and re-open it***) are:

- *Edit description*.- you can modify the description for the selected argument through this option.

- *Assign*.- allows to assign an argument type to the selected argument, which can be on of the following:
  - *Set standard values*.- It allows you to create a list of values that later can be individually or collectively selected. Note that these values must be separated by a semicolon (';'), for example as in "Men; Women". (Associated icon ⬅).
  - *Set as data argument*.- Marks the argument as receptacle for the data loaded. It does not have an associated icon, but when the editor is re-opened the name of the argument will be highlighted in blue.
  - *Assign to single color selection dialog*.- The argument is a color, which will be selected from a color palette for individual selection (Associated icon ◉).
  - *Assign to multi color selection dialog*.- The argument is a set of colors, which will be selected from a palette of multiple choice colors. (Associated icon ◉).
  - *Assign to Font selection dialog*.- The argument is a font. When selected, a font selection dialog will allow to choose one from the presintalled fonts in the system (Associated icon A).

IMPORTANT.- *For the arguments to show the following assigned argument type selectors, you must click on "Accept" with the data for headers and columns loaded in the right part of the editor, and then re-open it.*

- *Assign to single select data columns viewer*.- Selects a variable from the loaded data (Associated icon Var).
- *Assign to multi select data columns viewer*.- Allows multiple selection of variables from the loaded data file. (Associated icon Var).
- *Assign to single select data values viewer*.- Selects a unique value from a variable of the loaded data. (Associated icon Var)
- *Assign to multi select data values viewer*.- Allows to select multiple values from a variable of the loaded data (Associated icon Var).

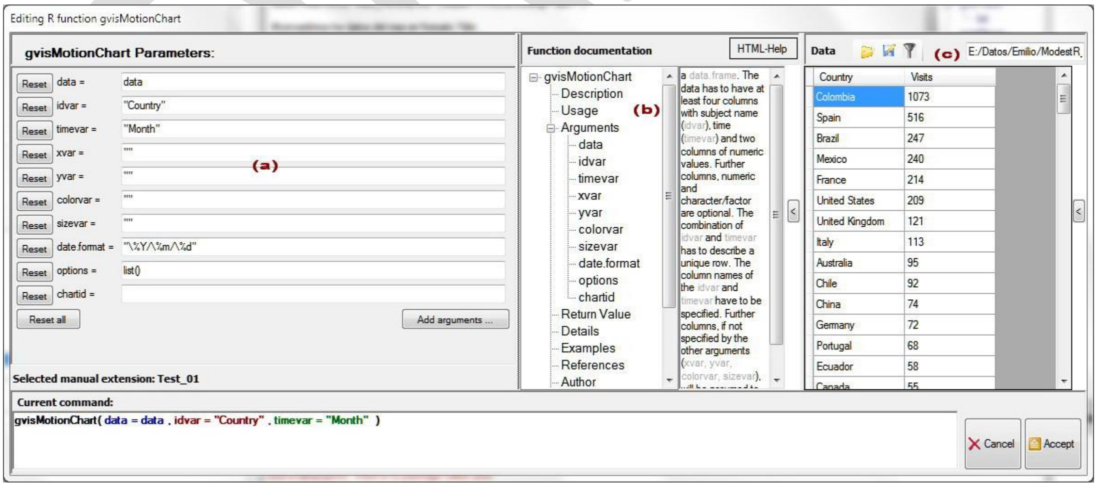
- *Reset assignment*.- Removes the assign of values to the argument (remember to click on "Accept" for the changes to be carried out).
- *Move Up*.- Move the argument upwards in the tree.
- *Move down*.- Move the argument downwards in the tree.

Each manual extension can contain predefined assignments different for each argument. The objective is to give you the freedom to select your own preselections or add new ones in the "RWizard" manual extension.

There is also an icon representing a function f(x), which is assigned to various parameters and is related to the applications distributed with RWizard. This mark will not be generated by any argument type, since is not available for the common user.

4.2 Function Editor

Another important functionality of RWizard is its ability to directly access functions in a script and modify the value of its arguments (Figure 18). You can reach this editor from the "*Functions*" work area, or from the "*Command Search*" one, though in the first case a function call will be modified, and in the latter a new one will be created.



data = data
idvar = "Country"
timevar = "Month"
xvar = ""
yvar = ""
colorvar = ""
sizevar = ""
date.format = "\%Y/\%m/\%d"
options = list()
chartid =

a data.frame. The data has to have at least four columns with subject name (idvar), time (timevar) and two columns of numeric values. Further columns, numeric and character/factor are optional. The combination of idvar and timevar has to describe a unique row. The column names of the idvar and timevar have to be specified. Further columns, if not specified by the other arguments (xvar, yvar, colorvar, sizevar),

Country	Visits
Colombia	1073
Spain	516
Brazil	247
Mexico	240
France	214
United States	209
United Kingdom	121
Italy	113
Australia	95
Chile	92
China	74
Germany	72
Portugal	68
Ecuador	58
Canada	55

Selected manual extension: Test_01

Current command:

```
gvisMotionChart( data = data , idvar = "Country" , timevar = "Month"  )
```

Figure 18 Function Editor

In this editor there are three separated areas:

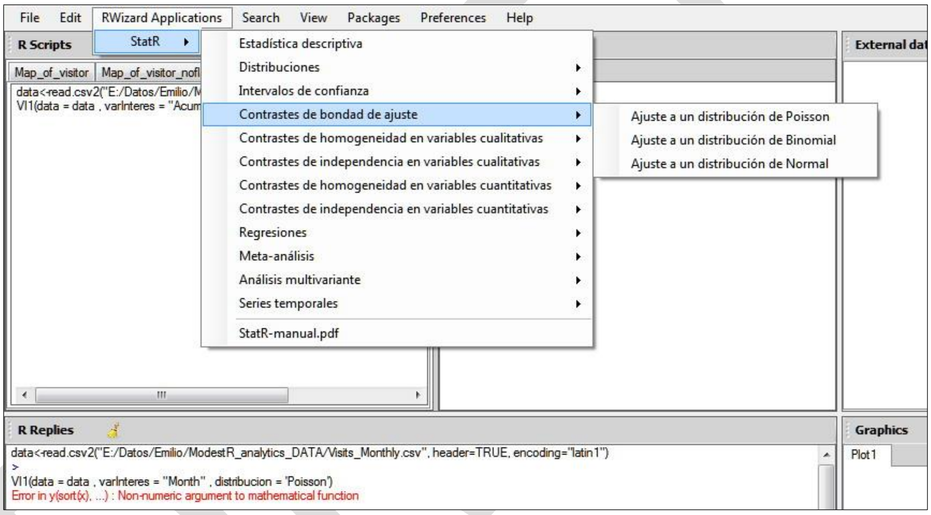
a) *Arguments of the function to edit*. For each argument you can modify its value, directly writing in the designated text box, or by means of the icons that can be generated by the manual extensions (check section 4.1Extension manual).

b) *Tree view for the function***.** It specifies descriptions, examples and characteristics. In case of arguments, you will be able to access to the default values which will appear in the a) area (check section 4.1Extension manual). The manual will only appear if installed in RWizard. Functions with installed manuals appear highlighted in blue (in other case, they will appear in gray). Check section 2.4Packages in order to know how to install a manual.

c) *Table of loaded data***.** Shows the loaded data file that will be used for those arguments assigned to one of the default argument types of the manual extension (check section 4.1Extension manual)

4.3 RWizard Applications

This main menu option will be enabled as soon as any of the applications developed for the RWizard environment is installed. They can be downloaded from the web page www.ipez.es/RWizard/Download.html. Once one of this applications is installed, a new menu will appear under *Rwizard applications* (Figure 19), showing its functionalities.


File
Edit
RWizard Applications
Search
View
Packages
Preferences
Help
R Scripts
StatR
External dat
Map_of_visitor
Map_of_visitor_nofl
Estadística descriptiva
Distribuciones
Intervalos de confianza
Contrastes de bondad de ajuste
Contrastes de homogeneidad en variables cualitativas
Contrastes de independencia en variables cualitativas
Contrastes de homogeneidad en variables cuantitativas
Contrastes de independencia en variables cuantitativas
Regresiones
Meta-análisis
Análisis multivariante
Series temporales
StatR-manual.pdf
Ajuste a un distribución de Poisson
Ajuste a un distribución de Binomial
Ajuste a un distribución de Normal
R Replies
Graphics
Plot1

```
data<-read.csv2("E:/Datos/Emilio/M
VI1(data = data , varInteres = "Acum
```

```
data<-read.csv2("E:/Datos/Emilio/ModestR_analytics_DATA/Visits_Monthly.csv", header=TRUE, encoding="latin1")
>
VI1(data = data , varInteres = "Month" , distribucion = 'Poisson')
Error in y(sort(x), ...) : Non-numeric argument to mathematical function
```

Figure 19 StatR Application menu