GALPon - Grupo de Amigos de Linux de Pontevedra - Web

Sábados Libres na Altamar

ENTORNOS GRÁFICOS Y CREACIÓN DE PAQUETES

26/04/2014

Miguel Ángel Rodríguez Muiños

mail@leugimsan.es

@mianromu

TAGS: #SLAltamar #GALPon #rstats

Programa:

- Introducción
  - R como lenguaje de programación
  - Los repositorios «oficiales»
    - CRAN
    - Bioconductor

- Entornos de usuario para R
  - RCommander
  - Rstudio
  - Otros…

- Desarrollo de entornos gráficos para nuestros programas
  - El paquete gWidgets2 (tcltk)

- Distribución y publicación de nuestras aplicaciones: Creación de paquetes
  - Creación de paquetes
    - La función *package.skeleton()*
    - Rtools
    - Rstudio
  - La Forja de R (R-Forge)
  - Publicación en CRAN

- Referencias bibliográficas

INTRODUCCIÓN

Qué es R?

Programa estadístico? -> Lenguaje de programación

La estructura de R -> Base + packages

Repositorios Oficiales

- CRAN -> http://cran.r-project.org/
  - El mirror de la OSL del CIXUG -> http://ftp.cixug.es/CRAN/
- Biconductor -> http://bioconductor.org/

La Comunidad R-Hispano -> http://r-es.org/

- Asóciate! -> http://r-es.org/Hazte+socio
- La lista R-help-es -> https://stat.ethz.ch/mailman/listinfo/r-help-es
- Las Jornadas de R -> http://r-es.org/VI+Jornadas

La complejidad de comenzar con R (soluciones)

- EpiLinux -> http://www.sergas.es/MostrarContidos_N3_T01.aspx?IdPaxina=50178
- BioStatFLOSS -> http://www.sergas.es/MostrarContidos_N3_T01.aspx?IdPaxina=62658

ENTORNOS DE USUARIO

"La necesidad de los entornos gráficos de usuario (GUI)"

Rcommander -> http://www.rcommander.com/

Curso de Estadística básica con RCommander

Anuncio: http://osl.cixug.es/publicado-o-curso-analise-estatistica-con-rcommander/

Acceso al curso: http://cursos.cixug.es/

Rstudio -> https://www.rstudio.com/

Deducer -> http://www.deducer.org/

RkWard -> http://es.wikipedia.org/wiki/RKWard

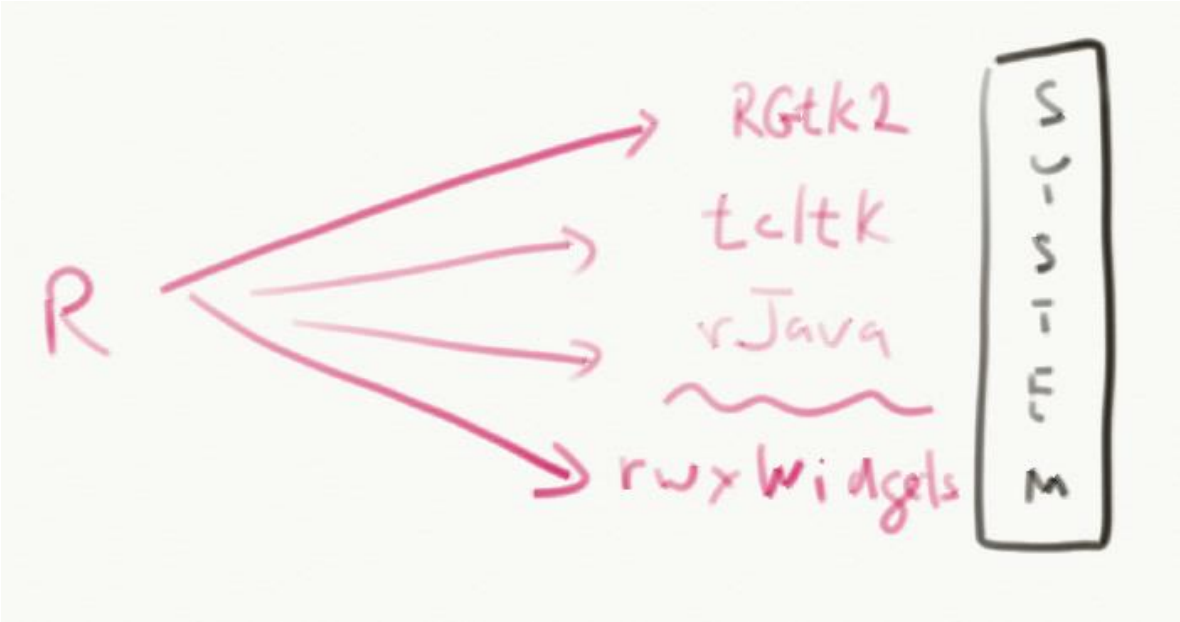
Red-R -> http://www.red-r.org/ **¿? ¿? ¿?**

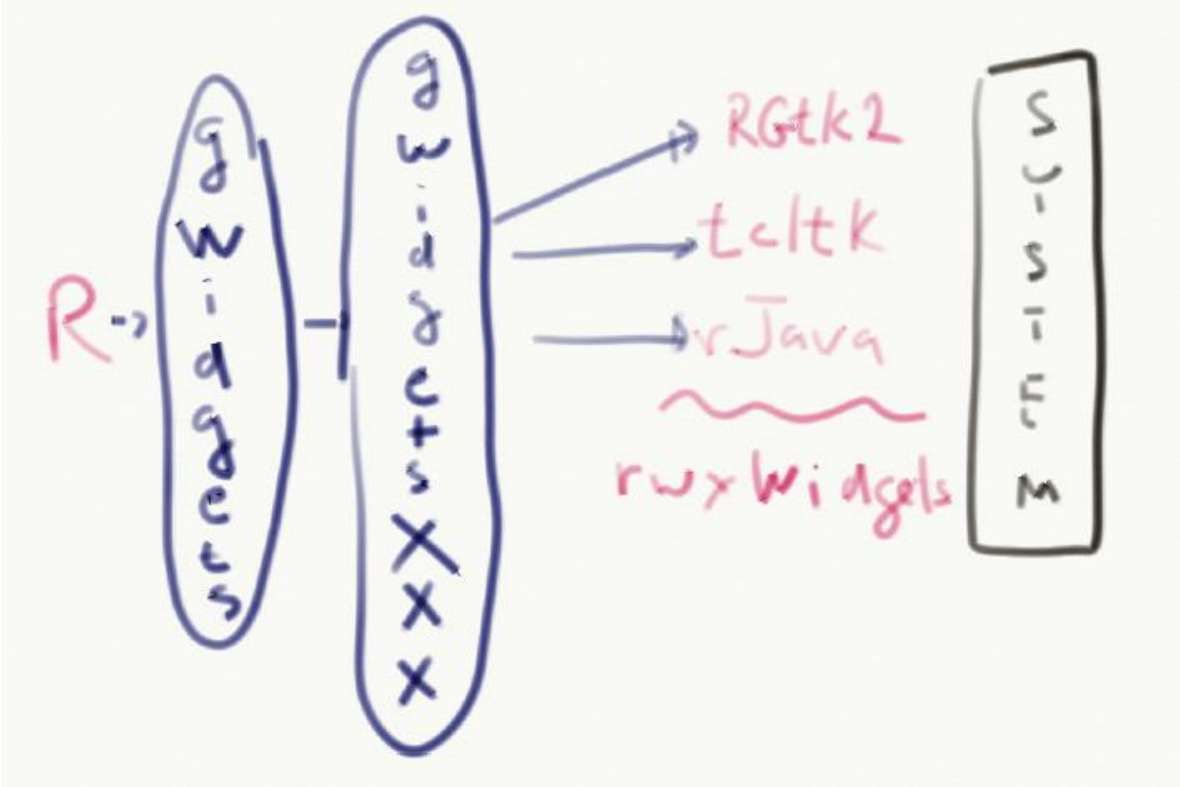
DESARROLLO DE ENTORNOS GRÁFICOS

El paquete gWidgets

http://cran.r-project.org/web/packages/gWidgets/vignettes/gWidgets.pdf

R has several packages (`RGtk2`, `tcltk`, `rJava`, `RwxWidgets`, ...) that allow the R user to interface with GUI toolkits.
The `gWidgets` package provides a *toolkit-independent* means to interface with these toolkits using an *simplified* programming interface.





Cómo empezar?

```
require(gWidgets)
options(guiToolkit="tcltk")
require(gWidgetstcltk)
```

sample dialogs

```
gmessage("Hello world", title="gmessage")
gmessage("Error, Error", title="gmessage",
  icon="error")
ginput("Enter your lucky number",text="7",
  title="ginput", icon="question")
 gconfirm("Ames is awesome", title="gconfirm")
 source(gfile())
 setwd(gfile(type="selectdir"))
```

gbutton

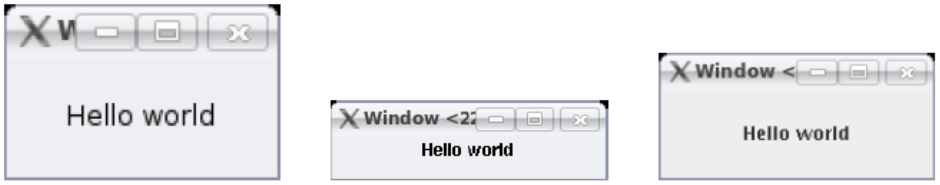
```
gbutton("Hello world", cont=TRUE)
```

button with icon

```
gbutton("ok", cont=TRUE)
```

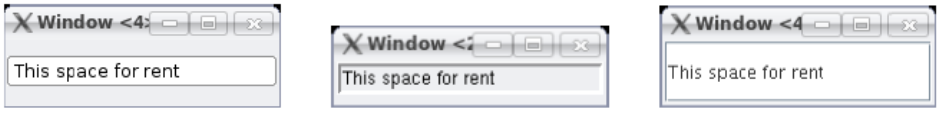
glabel

```
glabel("Hello world", cont=TRUE)
```



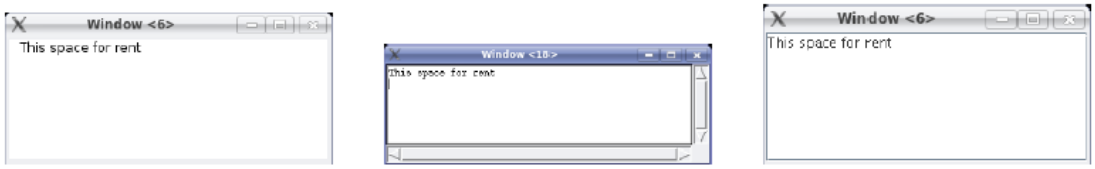
gedit

```
gedit("This space for rent", cont=TRUE)
```



gtext

```
gtext("This space for rent",  cont=TRUE)
```

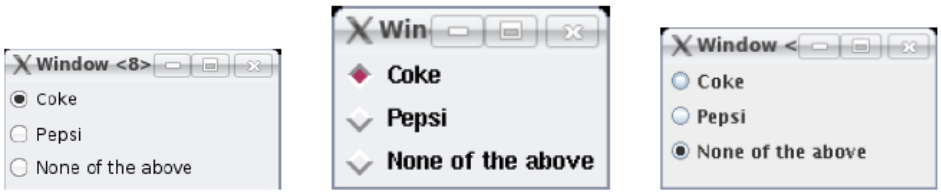


gcheckbox

```
gcheckbox("Do you like coke", cont=TRUE)
```

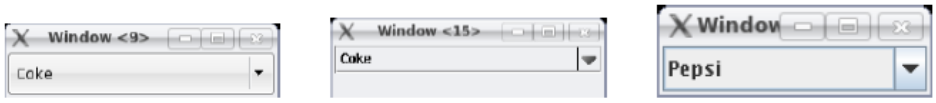
gradio

```
items = c("Coke","Pepsi","None of the above")
gradio(items, cont=TRUE)
```



gdroplist

```
gdroplist(items, cont=TRUE)
```



gdroplist: aka a combobox

```
gdroplist(items, editable=TRUE, cont=TRUE)
```

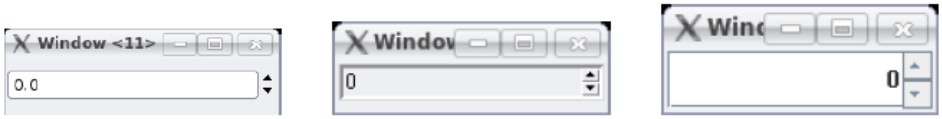
gslider

```
gslider(from=0, to = 100, by = 1, cont=TRUE)
```

[In `tcltk` only integer selections are possible. No ability (currently) to adjust displayed axis values.]

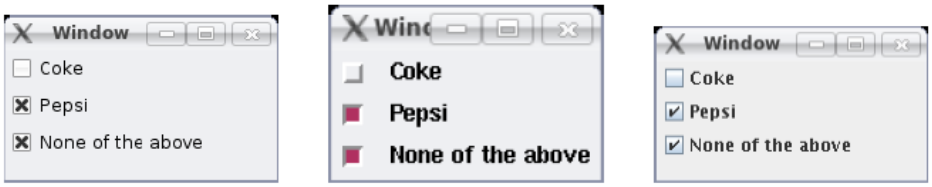
gspinbutton

```
gspinbutton(from=0, to = 1, by = 0.1, cont=TRUE)
```



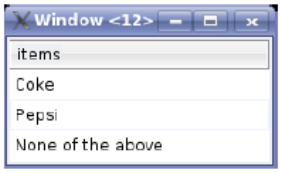
gcheckboxgroup

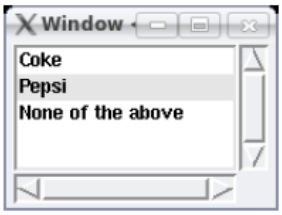
```
gcheckboxgroup(items, cont=TRUE)
```

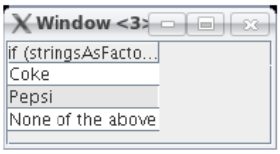


gtable

```
gtable(items, multiple=TRUE, cont=TRUE)
```

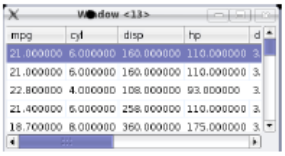


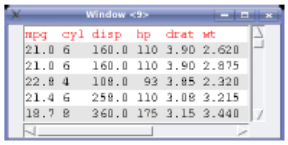


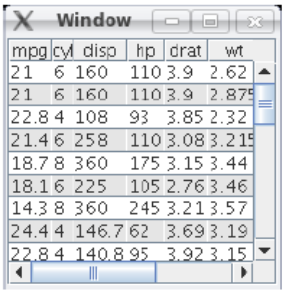


gtable

```
gtable(mtcars,chosencol=6, cont=TRUE)
```



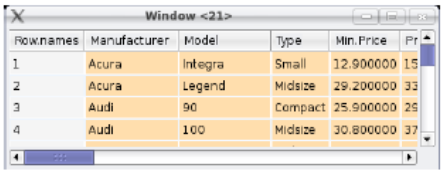


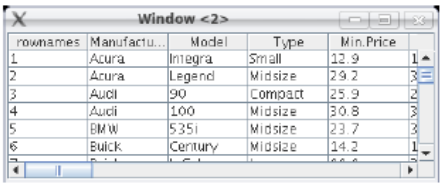


Editar un dataframe

gdf

```
require(MASS)
gdf(Cars93, cont=TRUE)
```

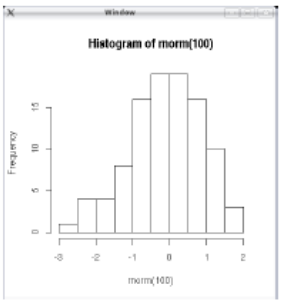


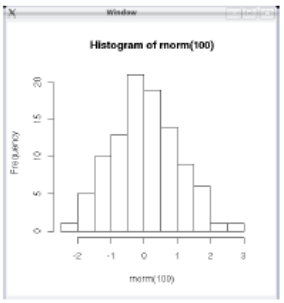


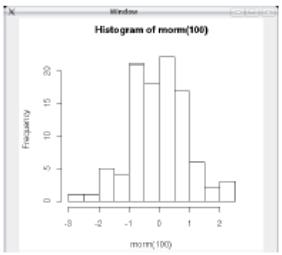
[Not available in tcltk (not in base set of libraries); double click to edit cell contents]

gimage

```
png("/tmp/rnorm.png")
hist(rnorm(100))
dev.off()  ## tcltk has limited number of formats
system("convert /tmp/rnorm.png /tmp/rnorm.gif")
gimage("/tmp/rnorm.gif", cont=TRUE)
```

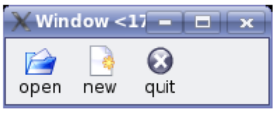


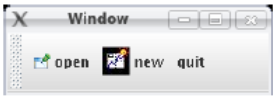




gtoolbar

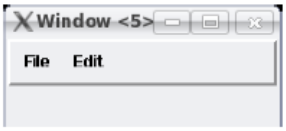
```
f = function(h,...) print("Hello world")
  tblst=list(
    open = list(handler=f, icon="open"),
    new  = list(handler=f, icon="new"),
    quit = list(handler=f, icon="close")
  )
gtoolbar(tblst, cont=TRUE)
```

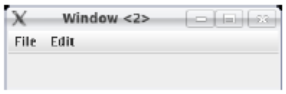




gmenu

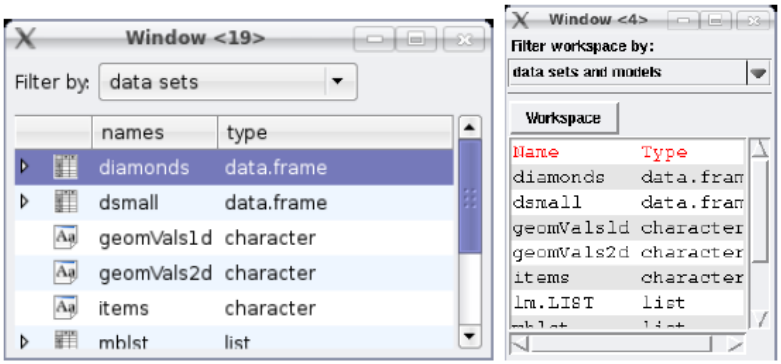
```
  mblst = list(
    File=list(
      open = list(handler=f,icon="open"),
      quit = list(handler=f,icon="close")
    ),
    Edit = list(
      cut  = list(handler=f),
      copy = list(handler=f)
    )
  )
  gmenu(mblst, cont=TRUE)
```





gvarbrowser

```
gvarbrowser(cont=TRUE)
```



[In RGtk2 a gtree widget is used. This isn't implemented in the other toolkits]

Cómo recogemos los valores seleccionados?

```
b = gradio(c("coke","pepsi","neither"), cont=TRUE)
> svalue(b)                # retrieve value
[1] "coke"
> svalue(b) <- "pepsi"     # set by name
> svalue(b)
[1] "pepsi"
```

`svalue`	get selected value
`svalue<-`	set selected value
`enabled<-`	turn widget gray, disable input
`size<-`	set widget size
`font<-`	set widget font

An interactive GUI has **handlers** which respond to **events** initiated by a user, eg. *a mouse click*, *pressing the enter key*, *pressing a key*, *a drag and drop* etc. In `gWidgets` each widget has a default handler. (eg. for buttons, a click; for gedit the enter key)
A function can be specified at the time of construction using the `handler` argument. The `action` argument is passed to the handler.

```
## print message
gbutton("press me", cont=TRUE, handler =
  function(h,...) print("hello world")
)
```

Contenedores

More complicated applications require some idea of a container to pack widgets into. In `gWidgets` there is a distinction between a top-level window (produced with `gwindow`, or using `cont=TRUE`) and other containers.
The easiest to use container is the `ggroup` container. This container packs in its child widgets (and containers) from left to right (the default) or from top to bottom (when `horizontal=FALSE`.)

Button box (take 1)

```
win = gwindow("Button box")
g = ggroup(cont = win)
gbutton("cancel", cont=g)
gbutton("ok", cont=g)
```

Use addSpring to push things to the right (or bottom)

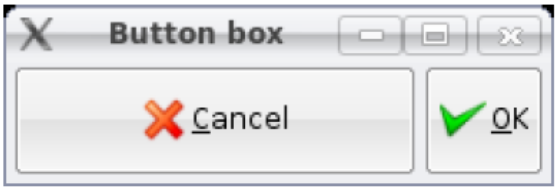
Button box (take 2)

```
win = gwindow("Button box")
g = ggroup(cont = win)
gbutton("cancel", cont=g)
addSpring(g)
gbutton("ok", cont=g)
```

The expand=TRUE argument will instruct the widget to fill as much space as it can.

Button box (take 3)

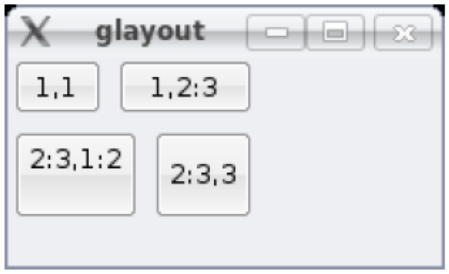
```
win = gwindow("Button box")
g = ggroup(cont = win)
gbutton("cancel", cont=g, expand=TRUE)
gbutton("ok", cont=g)
```



Glayout: un contenedor tipo "grid"

glayout

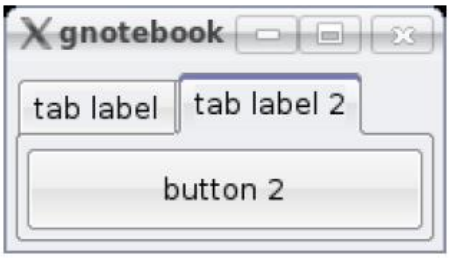
```
win = gwindow("glayout")
tbl = glayout(cont=win)
tbl[1,1] <- gbutton("1,1", cont=tbl)
tbl[1,2:3] <- gbutton("1,2:3", cont=tbl)
tbl[2:3,1:2] <- gbutton("2:3,1:2\n", cont=tbl)
tbl[2:3,3] <- gbutton("2:3,3", cont=tbl)
visible(tbl) <- TRUE  ## RGtk2 only
```



Gnotebook: varias pestañas

gnotebook

```
win = gwindow("gnotebook")
nb = gnotebook(cont=win)
gbutton("button",   cont=nb, label = "tab label")
gbutton("button 2", cont=nb, label = "tab label 2")
```



Gframe: contenedor tipo ventana

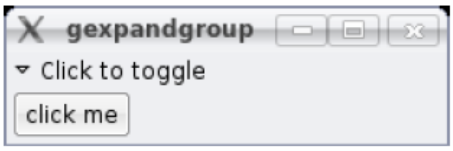
gframe

```
win = gwindow("gframe")
gp = ggroup(cont=win)
g = gframe("Text label", cont=gp)
gbutton("click me", cont=g)
```

Gexpandgroup: como gfraem pero el contenido puede ser ocultado

gexpandgroup

```
win = gwindow("gexpandgroup")
g = gexpandgroup("Click to toggle", cont=win)
gbutton("click me", cont=g)
visible(g) <- TRUE  ## open up
```



Panel ajustable entre dos grupos

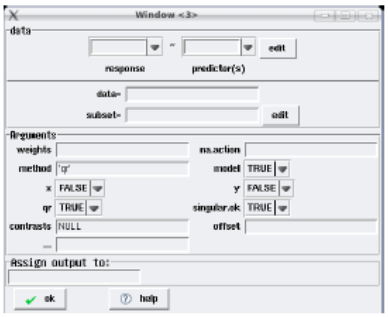
gpanedgroup

```
win = gwindow("gpanedgroup")
pg = gpanedgroup(cont=win)
gbutton("button 1", cont=pg)
gbutton("button 2", cont=pg) ## add twice
```

“Truco” final

ggenericwidget

```
ggenericwidget("lm", cont=TRUE)
```

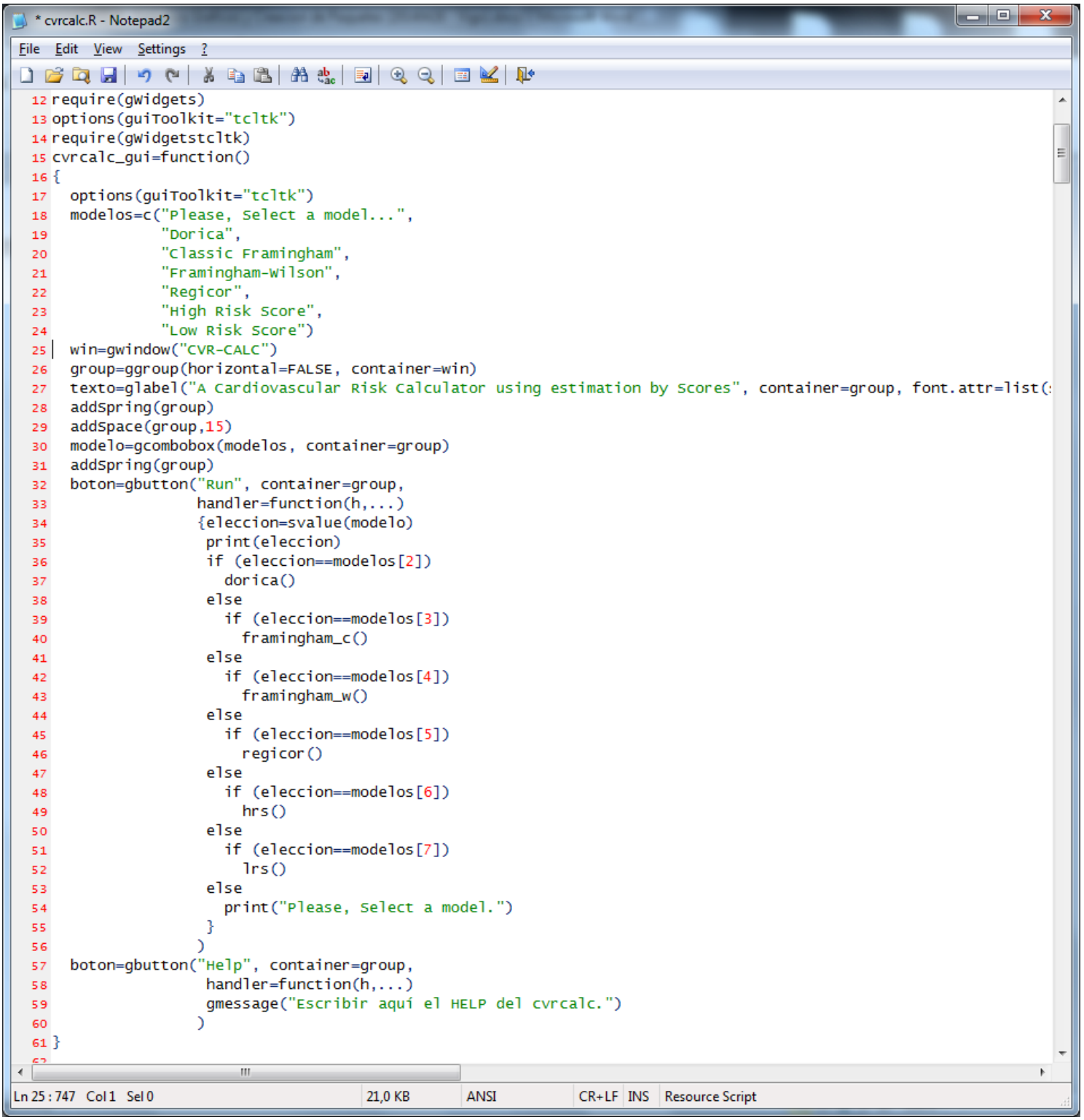


Una mejora de gWidgets: gWidgets2

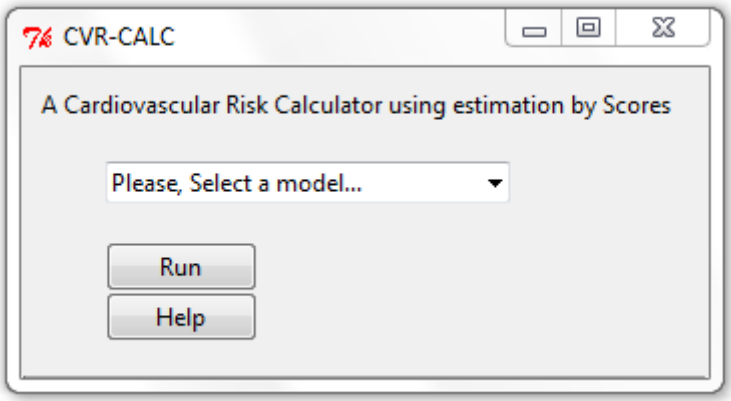
https://github.com/jverzani/gWidgets2

Ejemplo:

CVRCALC.R: Una calculadora de Riesgo Cardiovascular por Scores




```
require(gWidgets)
options(guiToolkit="tcltk")
require(gWidgetstcltk)
cvrcalc_gui=function()
{
  options(guiToolkit="tcltk")
  modelos=c("Please, Select a model...",
            "Dorica",
            "Classic Framingham",
            "Framingham-Wilson",
            "Regicor",
            "High Risk Score",
            "Low Risk Score")
  win=gwindow("CVR-CALC")
  group=ggroup(horizontal=FALSE, container=win)
  texto=glabel("A Cardiovascular Risk Calculator using estimation by Scores", container=group, font.attr=list(
  addSpring(group)
  addSpace(group,15)
  modelo=gcombobox(modelos, container=group)
  addSpring(group)
  boton=gbutton("Run", container=group,
                handler=function(h,...)
                {eleccion=svalue(modelo)
                 print(eleccion)
                 if (eleccion==modelos[2])
                   dorica()
                 else
                   if (eleccion==modelos[3])
                     framingham_c()
                 else
                   if (eleccion==modelos[4])
                     framingham_w()
                 else
                   if (eleccion==modelos[5])
                     regicor()
                 else
                   if (eleccion==modelos[6])
                     hrs()
                 else
                   if (eleccion==modelos[7])
                     lrs()
                 else
                   print("Please, Select a model.")
                 }
                )
  boton=gbutton("Help", container=group,
                 handler=function(h,...)
                 gmessage("Escribir aquí el HELP del cvrcalc.")
                )
}
```



Código fuente: http://goo.gl/Xy66FH

CREACIÓN DE PAQUETES

La Forja de R (R-Forge)

http://r-forge.r-project.org/

R-Forge (http://r-forge.r-project.org/) ofrece una plataforma central para el desarrollo de paquetes de R, además de software y proyectos relacionados con R. Los paquetes alojados en R-Forge se ponen a disposición de los usuarios en su código fuente, así como en formato binario precompilado para diversos sistemas operativos.

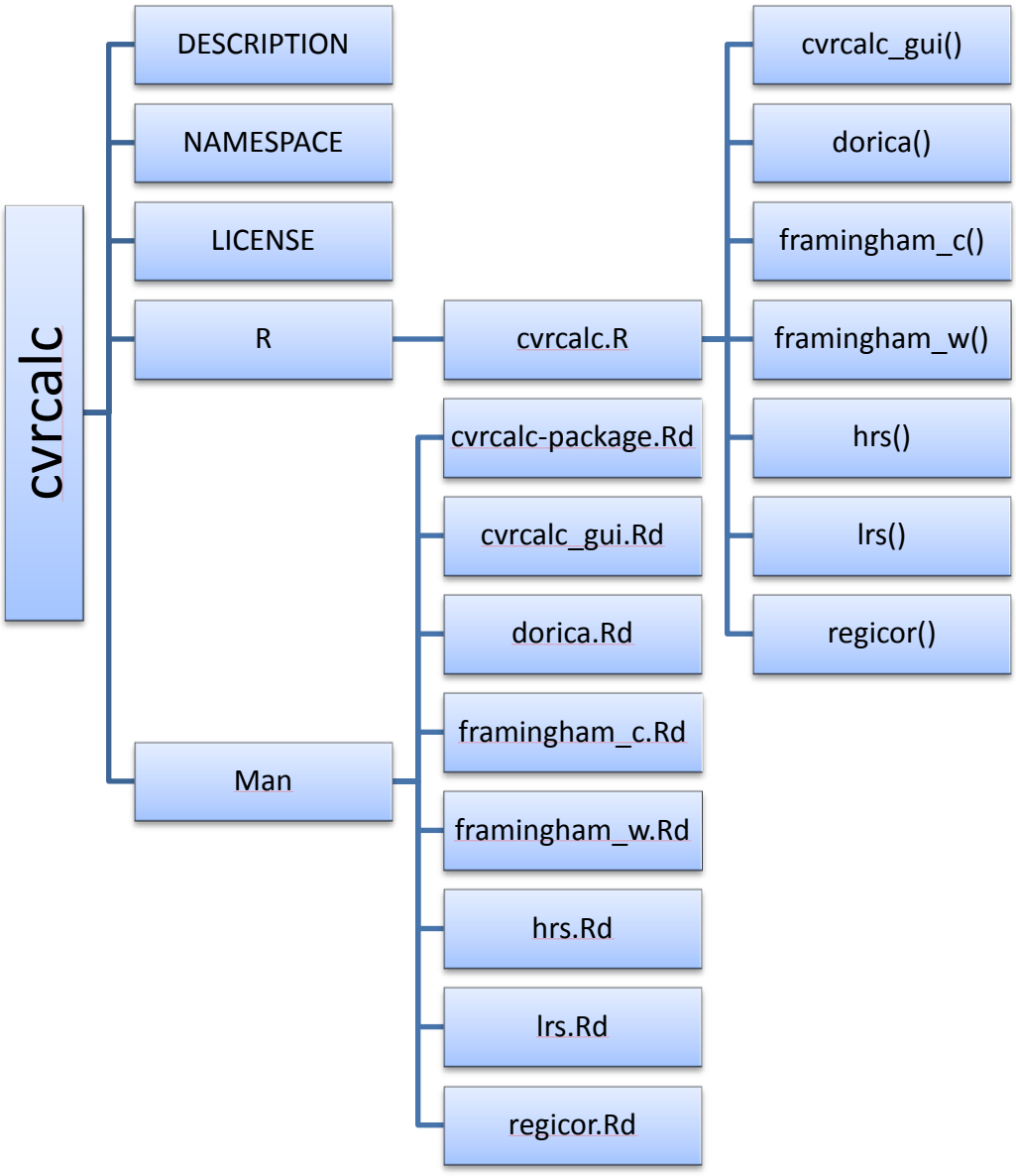
Los desarrolladores de R-Forge organizan su trabajo en Proyectos. Al llevar a cabo proyectos de software, el código fuente cambia con el tiempo: se crean nuevos archivos, se modifican o eliminan otros, se reescribe código, … Por lo general, varios autores trabajan en varias ramas del programa y realizar un seguimiento de cada cambio puede convertirse en una tarea complicada. Una solución general a este problema es usar un sistema de control de versiones (SVN). Un SVN realiza un seguimiento de la historia completa de la estructura de archivos del proyecto. En cualquier punto de la etapa de desarrollo es posible volver a cualquier etapa anterior en la historia para inspeccionar y restaurar archivos antiguos. Como cada etapa se asigna automáticamente una única versión que aumenta con el tiempo, este sistema recibe el nombre de Control de Versiones. En R-Forge se crea automáticamente un repositorio SVN (de control de versiones) para cada proyecto. Los miembros del proyecto solo tienen que instalar un cliente SVN de su elección para acceder a su repositorio. Además de la copia de seguridad inherente de cada versión dentro del repositorio se genera, diariamente, una copia de seguridad del repositorio entero.

CREACIÓN DE PAQUETES

El tutorial de Francesc Carmona ->
http://www.ub.edu/stat/docencia/Cursos-R/Radvanced/materials/Crear_paquetes_R.pdf

EL PAQUETE CVRCALC

http://cvrcalc.r-forge.r-project.org/



DESCRIPTION

```
Package: cvrcalc
Type: Package
Title: Cardiovascular Risk Calculator
Version: 1.0
Date: 2013-02-13
Author: Maria Teresa Seoane Pillado and Miguel Angel Rodriguez Muinos
Maintainer: M. A. Rodriguez Muinos <mail@leugimsan.es>
Description: A cardiovascular risk calculator by scores
Depends: R (>= 2.10.0), XLConnect, gWidgets, gWidgetstcltk
License: GPL-2

```

NAMESPACE

```
exportPattern("^[[:alpha:]]+")

import(XLConnect)
import(gWidgets)
import(gWidgetstcltk)

```

LICENSE

```
This software is distributed under the terms of the GNU General Public
License as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

A copy of version 2 of the GNU General Public License is in file GPL-2
in the sources of this package, and is also available at
http://www.r-project.org/Licenses/

```

cvrcalc.R

```
1 ##########################################################################
2 ## CVRCALC: A Cardiovascular Risk's Calculator by Scores
3 ## Developers: M Teresa Seoane Pillado & Miguel Angel Rodriguez Muinos
4 ## Contact: mail [at] leugimsan.es
5 ## From: A Coruna, Spain
6 ## Version: 1.0
7 ## creation Date: 2013/02/13
8 ## Last Version Date: 2013/08/08
9 ##########################################################################
10
11 # require(XLConnect)
12 # require(gWidgets)
13 # options(guiToolkit="tcltk")
14 # require(gWidgetstcltk)
15
16 cvrcalc_gui=function()
17 {
18 options(guiToolkit="tcltk")
19 modelos=c("Please, Select a model...",
20 "Dorica",
21 "Classic Framingham",
22 "Framingham-Wilson",
23 "Regicor",
24 "High Risk Score",
25 "Low Risk Score")
26
27 win=gwindow("CVR-CALC")
28 group=ggroup(horizontal=FALSE, container=win)
29 texto=glabel("A Cardiovascular Risk Calculator using estimation by
Scores", container=group, font.attr=list(style="bold"))
30 addSpring(group)
31 addSpace(group,15)
32 modelo=gcombobox(modelos, container=group)
33 addSpring(group)
34 boton=gbutton("Run", container=group,
35 handler=function(h,...)
36 {eleccion=svalue(modelo)
37 print(eleccion)
38 if (eleccion==modelos[2])
39 dorica()
40 else
41 if (eleccion==modelos[3])
42 framingham_c()
43 else
44 if (eleccion==modelos[4])
45 framingham_w()
46 else
47 if (eleccion==modelos[5])
48 regicor()
49 else
50 if (eleccion==modelos[6])
51 hrs()
52 else
53 if (eleccion==modelos[7])
54 lrs()
55 else
56 print("Please, Select a model.")
57 }
```

```
58 )
59 boton=gbutton("Help", container=group,
60 handler=function(h,...)
61 gmessage("Escribir aquí el HELP del cvrcalc.")

62 )
63 }
64
65
[.. aquí van las funciones…]
744
745 ### END ###
```

BIBLIOGRAFÍA

- R Core Team (2012). Writing R extensions
- Rossi, P.(2006). Making R Packages Under Windows: A Tutorial.
- Leisch, F. (2009). Creating R Packages: A Tutorial.
- Falcon, S. and Gentleman, R (2006). Lab: Writing packages in R.
- Verzani, J (2007). gWidgets: API for building interactive GUIs (useR!2007).
- Carmona, F (2013). Creación de paquetes de r en Windows (y Linux)


"That's all Folks!"

* * * * *