Package 'stacomirtools'

January 4, 2012

Type Package

Title stacomi ODBC connexion class and ggplot2 graphical interface for beginners

Version 0.2.36

Date 2012-01-04

Author Cedric Briand

Maintainer Cedric Briand<cedric.briand00@gmail.com>

Description (state=development) gWidget graphical interface to the
ggplot2 package see http://had.co.nz/ggplot2/, also S4 class wrappers for odbc connexion.

License GPL (>= 2)

Collate
'ConnexionODBC.r' 'RequeteODBC.r' 'RequeteODBCwhere.r''RequeteODBCwheredate.r' 'ggplot2usr.R' 'ggplot2u

LazyLoad yes

Depends proto, methods, ggplot2, gWidgets, gWidgetsRGtk2,RODBC

R topics documented:

`stacomirtools-package` *ggplot2 user interface and RODBC connector class*

Description

this package is intented to be used by beginners who will be able to access to the different layers of the ggplot2 package is yet far from providing the full capability of the ggplot2, but will help in building the first graphs and print their formula : this package is still in developpement and full of bugs, so far intended only for showing the aim of the project has been inially developped within a french project, which builds a database for migratory fishes control stations, along with several graphical tools help the users to 'view' their data

Details

Package:	stacomirtools
Type:	Package
Version:	0.2
Date:	2012-01-03
License:	GPL (>= 2)
LazyLoad:	yes

TODO

include the scales

Author(s)

Cedric Briand <cedric.briand00@gmail.com>

References

http://had.co.nz/ggplot2/

See Also

http://had.co.nz/ggplot2/

Examples

```
mtcars=mtcars # the dataframe used are listed in the base environment .GlobalEnv
ggplot2usr(data="mtcars")
```

---

build_proto *builds proto widgets*

Description

The function build_proto is called to build the proto widgets

Note

Unlike S3 or S4 class, the structure of proto objects is lost when the package is built embedded them in a function which will be called to create them.

Author(s)

cedric

References

http://wiener.math.csi.cuny.edu/pmg/gWidgets/index.html/ especially ProtoExample

---

chnames *This function replaces the variable names in a data.frame*

Description

This function replaces the variable names in a data.frame

Usage

```
chnames(objet, old_variable_name, new_variable_name)
```

Arguments

objet	a data frame
old_variable_name	a character vector with old variables names
new_variable_name	a character vector with new variables names

Value

objet

Author(s)

Cedric Briand <cedric.briand00@gmail.com>

`confirmDialog` *Confirmation dialog gtkwindow when selecting a different dataframe...*

Description

Confirmation dialog gtkwindow when selecting a different dataframe This is directly taken from gWidget vignette and slightly modified

Usage

```
confirmDialog(message, handlerok, handlercancel)
```

Arguments

`handlerok`	the handler triggered when ok is clicked
`handlercancel`	the handler triggered when cancel is clicked

Author(s)

John Verzani

`connect-methods` *Methods for Function connect*

Description

see individual .r files for help and examples

Methods

`signature(objet = "ConnexionODBC")` connect an odbc database,and eventually leaves it open for further queries, the connexion may send message in the native language if stacomiR package is in use

`signature(objet = "RequeteODBC")` connect an odbc database,performs an sql request

`signature(objet = "RequeteODBCwhere")` connect an odbc database,performs an sql request with where clause

`signature(objet = "RequeteODBCwheredate")` connect an odbc database,performs an sql request with where clause for an interval

Examples

```
##
#objet<-new("RequeteODBCwhere")
#connect(objet)
```

ConnexionODBC-class *Class "ConnexionODBC"*

Description

Mother class for connection, opens the connection but does not shut it

Objects from the Class

Objects can be created by calls of the form `new("ConnexionODBC", ...)`.

`baseODBC`: Object of class `"vector"` The database

`silent`: Object of class `"logical"` The mode

`etat`: Object of class `"character"` The state

`connexion`: Object of class `"ANY"` The connection

Slots

`baseODBC`: Object of class `"vector"` The database

`silent`: Object of class `"logical"` The mode

`etat`: Object of class `"character"` The state

`connexion`: Object of class `"ANY"` The connection

Methods

connect `signature(objet = "ConnexionODBC")`: Connection to the database

Note

Opens the connection but does not close it. This function is intended to be used with stacomiR package, where the error message are collected from the database It has also been programmed to work without the stacomiR package, as it will test for the existence of envir_stacomi environment.

Author(s)

cedric.briand00@gmail.com

Examples

```
showClass("ConnexionODBC")
## Not run:
 # this is the mother class, you don't have to use it, please use requeteODBC and daughter class instead
 objet<-new("ConnexionODBC")
 objet@baseODBC<-c("myodbcconnexion","myusername","mypassword")
 objet@silent<-FALSE
 objet<-connect(objet)
 odbcClose(objet@connexion)

## End(Not run)
```

`ex` *ex fonction to write to excel, not used within the program but can still be used*

Description

ex fonction to write to excel, not used within the program but can still be used

Usage

```
ex(d = NULL)
```

Arguments

`d`

Author(s)

Cedric Briand <cedric.briand00@gmail.com>

`extract_aes_param` *Build a list of aes and param by comparing values in the droplist and default and only using those different from default...*

Description

Build a list of aes and param by comparing values in the droplist and default and only using those different from default

Usage

```
extract_aes_param(list_aes)
```

Arguments

`list_aes`	:the list of defautl geom and params written in .RglobalEnv when selecting stat or geom in aes_frame

Value

list of list param and aes containing the values to be used in the layers

Author(s)

cedric

ggplot2usr *Main launch function...*

Description

Main launch function

Usage

```
ggplot2usr(data, envir=.GlobalEnv)
```

Author(s)

Cedric Briand <cedric.briand00@gmail.com>

Examples

```
 data("mtcars") # the dataframe used are listed in the base environment .GlobalEnv
 data("diamonds") # so that at least two dataframes appear in the list
 ggplot2usr(data="mtcars")
# at this stage you should see a frame with the following content
# BUTTON FRAME
# button geom<>stat => switches between geom and stat
# button Undo => you've done it wrong and want to start over with a simple geom_point
# checkbox Build layer => If checked this layer will be added to the plot
# button Graph => this is the button you will click to see the result (in a plot frame) and the formula use
# button data => updates the list of data and reloads it
# GEOM/STAT FRAME
# here you select the kind of layer you wish to use. Defaut aes and parameters will be loaded in the aes fr
# AES FRAME
# you have to choose the format of the required aesthetic for the layer
# for other aesthetics, you can choose either a variable from the dataframe (ex color<-cyl) or a parameter
# the choice of an aesthetic will override the choice of a parameter
# if the parameter differs from default, it will appear in the printed formula
# DATA
# lists all datasets in the user environment, if the column differ between two dataframes, a warning /
# confirmation will be issued.
# LAYOUT
# choice of the layout see \url{http://had.co.nz/ggplot2/}
# POSITION
# position \url{http://had.co.nz/ggplot2/}
```

ggploti_build *constructs the graph and prints the results...*

Description

constructs the graph and prints the results

Author(s)

cedric

haes *this handler first analyses the action (which is the name of the BoxinLayout)...*

Description

this handler first analyses the action (which is the name of the BoxinLayout) then calls the load_aes function

Usage

```
haes(h, ...)
```

Arguments

h	a handler

Author(s)

cedric

hChangedata *This function will check if data has the same column than previously*

Description

This function will check if data has the same column than previously, if not It will rebuild the graphical interface with the new data

Usage

```
hChangedata(h, ...)
```

Arguments

h	a handler for gdroplist data

Author(s)

cedric

`hplot` *handler function...*

Description

handler function

Usage

```
hplot(h, ...)
```

Arguments

`h`	a handler

Author(s)

cedric

`hretablir` *This function restarts with the default geom, when button Undo or 'Retablir' is pushed...*

Description

This function restarts with the default geom, when button Undo or 'Retablir' is pushed

Usage

```
hretablir(h, ...)
```

Arguments

`h`	handler

Author(s)

cedric

`hsw` *handler allowing the switch between stat and geom when button geom<>stat is triggered...*

Description

handler allowing the switch between stat and geom when button geom<>stat is triggered

Usage

```
hsw(h, ...)
```

Author(s)

cedric

`hUpdatedata` *This handler updates the list of data avalaible for drawing the graph...*

Description

This handler updates the list of data avalaible for drawing the graph If you want for instance to do some calculations in the dataframe and then reload it It uses the add and delete methods applicable to gframe to change the content of the combobox by re-building it

Usage

```
hUpdatedata(h, ...)
```

Arguments

`h`	handler for gdroplist data

Author(s)

cedric

`ind` *fonction pour renvoyer les index dans b des valeurs du vecteur a b peut apparaitre plusieurs fois dans a*

Description

fonction pour renvoyer les index dans b des valeurs du vecteur a b peut apparaitre plusieurs fois dans a

Usage

```
ind(a, b)
```

Arguments

a

b

Value

index of b in a

Note

attention le vecteur de resultat est dans le desordre

Author(s)

Cedric Briand <cedric.briand00@gmail.com>

`indrepeated` *fonction qui retourne l'index des valeurs repetees d'un vecteur*

Description

fonction qui retourne l'index des valeurs repetees d'un vecteur

Usage

```
indrepeated(a)
```

Arguments

a

Value

the index of repeated values within a vector

Author(s)

Cedric Briand <cedric.briand00@gmail.com>

induk *fonction qui renvoit l'index des valeurs apparaissant une seule fois*

Description

fonction qui renvoit l'index des valeurs apparaissant une seule fois

Usage

```
induk(a)
```

Arguments

a

Value

the index unique values within a vector

Author(s)

Cedric Briand <cedric.briand00@gmail.com>

is.even *is.even function modified from package sma (which did not verified that the entry was indeed an integer)*

Description

is.even function modified from package sma (which did not verified that the entry was indeed an integer)

Usage

```
is.even(x)
```

Arguments

x

Value

a logical

Author(s)

Cedric Briand <cedric.briand00@gmail.com>

`is.odd` *id.odd function modified from package sma (which did not verified that the entry was indeed an integer)*

Description

id.odd function modified from package sma (which did not verified that the entry was indeed an integer)

Usage

```
is.odd(x)
```

Arguments

`x`

Value

a logical

Author(s)

Cedric Briand <cedric.briand00@gmail.com>

`killfactor` *very usefull function used to "kill" these bloody factors that appears, noticeably after loading with odbc*

Description

very usefull function used to "kill" these bloody factors that appears, noticeably after loading with odbc

Usage

```
killfactor(df)
```

Arguments

`df`	a data.frame

Value

df

Author(s)

Cedric Briand <cedric.briand00@gmail.com>

layer_to_call *Function used to build layer calls and write print their formula...*

Description

Function used to build layer calls and write print their formula

Usage

```
layer_to_call(layer_name, layer_type, position, param, aes)
```

Arguments

layer_name	name of the layer
layer_type	currently only "geom" or "stat" for layer type object
position	
param	
aes	

Details

transfers arguments " " into NULL and the others in variable of class call a character vector is also returned to write the proper formula

Value

list list("char"=layer expression to be printed, "call" = a call to the layer object

Author(s)

cedric

load_aes *this functions loads the graphical interface with default elements from layer...*

Description

this functions loads the graphical interface with default elements from layer

Usage

```
load_aes(layer_choice)
```

Arguments

layer_choice	

Author(s)

cedric

ls.class — *Function used to list all elements belonging to a class within an environment...*

Description

Function used to list all elements belonging to a class within an environment

Usage

```
ls.class(name=.GlobalEnv, all.names=FALSE, pattern, class=data.frame)
```

Arguments

name	the name of the environment
all.names	see ls() for use
pattern	an optional regular expression see ls() for use
class	

Value

vector containing elements listed as belonging to a class

Author(s)

cedric

RequeteODBC-class — *Class "RequeteODBC"*

Description

ODBC Query. This class enables to retrieve data from the database. This class is inherited by RequeteODBCwhere and RequeteODBCwheredate

Objects from the Class

Objects can be created by calls of the form `new("RequeteODBC", sql=character(), query=data.frame())`.

`sql`: Object of class `"character"` The query

`query`: Object of class `"data.frame"` The result of the query

Slots

`sql`: Object of class `"character"` The "SELECT ..." part of the query

`query`: Object of class `"data.frame"` The result of the query

`baseODBC`: Object of class `"vector"` The name, user and password of the database

`silent`: Object of class `"logical"` True if the query must be executed silently, FALSE

`etat`: Object of class `"character"` The state of the query (Connecting, successful,...)

`connexion`: Object of class `"ANY"` The connexion

Extends

Class "ConnexionODBC", directly.

Methods

connect `signature(objet = "RequeteODBC")`: Connexion to the database

Note

Inherits from ConnexionODBC

Author(s)

cedric.briand00@gmail.com

See Also

ConnexionODBC RequeteODBCwhere RequeteODBCwheredate

Examples

```
showClass("RequeteODBC")
## Not run:
 objet=new("RequeteODBC")
 objet@open=TRUE # this will leave the connexion open, by default it closes after the query is sent
 #the following will work only if you have configured and odbc link
 objet@baseODBC=c("myodbcconnexion","myusername","mypassword")
 objet@sql= "select * from mytable limit 100"
 objet<-connect(objet)
 odbcClose(objet@connexion)
 envir_stacomi=new.env()
 # While testing I like to see the output of sometimes complex queries generated by the program
 assign("showmerequest",1,envir_stacomi) # can be anything just tests the existence of "showmerequest" in e
 objet=new("RequeteODBC")
 objet@baseODBC=c("myodbcconnexion","myusername","mypassword")
 objet@sql= "select * from mytable limit 100"
 objet<-connect(objet)
# the connexion is already closed, the query is printed

## End(Not run)
```

RequeteODBCwhere-class

Class "RequeteODBCwhere"

Description

SQL Query with WHERE and ORDER BY clauses.

Objects from the Class

Objects can be created by calls of the form `new("RequeteODBCwhere", where=character(),and=vector(),order_b`

`where`: Object of class `"character"` ~ The "WHERE" part of the query

`and`: Object of class `"vector"` ~ The "AND" part of the query

`order_by`: Object of class `"character"` ~ The "ORDER BY" part of the query

Slots

`where`: Object of class `"character"` ~ The "WHERE" part of the query

`and`: Object of class `"vector"` ~ The "AND" part of the query

`order_by`: Object of class `"character"` ~ The "ORDER BY" part of the query

`sql`: Object of class `"character"` ~ The "SELECT *..." part

`query`: Object of class `"data.frame"` ~ The result of the query

`baseODBC`: Object of class `"vector"` ~ The name, user and password of the database

`silent`: Object of class `"logical"` ~ TRUE if the query must be executed silently, FALSE else

`etat`: Object of class `"character"` ~ The state of the query (Connecting, successful,...)

`connexion`: Object of class `"ANY"` ~ The database connexion

Extends

Class `"RequeteODBC"`, directly. Class `"ConnexionODBC"`, by class "RequeteODBC", distance 2.

Methods

connect `signature(objet = "RequeteODBCwhere")`: Connect to the database

Note

Inherits from RequeteODBC the syntax is where="WHERE ..." and =vector("AND...","AND...") order_by="ORDER BY.."

Author(s)

cedric.briand00@gmail.com

See Also

`ConnexionODBC` `RequeteODBC` `RequeteODBCwheredate`

Examples

```
showClass("RequeteODBCwhere")
```

RequeteODBCwheredate-class
Class "RequeteODBCwheredate"

Description

Query with WHERE condition and overlaping dates clause.

Objects from the Class

Objects can be created by calls of the form `new("RequeteODBCwheredate", datedebut="POSIXlt",datefin="POSIX`

`datedebut`: Object of class `"POSIXlt"` ~ The starting date

`datefin`: Object of class `"POSIXlt"` ~ The ending date

`colonnedebut`: Object of class `"character"` ~ The name begin column

`colonnefin`: Object of class `"character"` ~ The name end column

Slots

`datedebut`: Object of class `"POSIXlt"` ~ The starting date

`datefin`: Object of class `"POSIXlt"` ~ The ending date

`colonnedebut`: Object of class `"character"` ~ The name of the begin column

`colonnefin`: Object of class `"character"` ~ The name of the end column

`where`: Object of class `"character"` ~ The WHERE clause

`and`: Object of class `"vector"` ~ The AND clause

`order_by`: Object of class `"character"` ~ The ORDER BY clause

`sql`: Object of class `"character"` ~ The SELECT clause

`query`: Object of class `"data.frame"` ~ The result of the query

`baseODBC`: Object of class `"vector"` ~ The database

`silent`: Object of class `"logical"` ~ The mode

`etat`: Object of class `"character"` ~ The state

`connexion`: Object of class `"ANY"` ~ The connexion

Extends

Class "RequeteODBCwhere", directly. Class "RequeteODBC", by class "RequeteODBCwhere", distance 2. Class "ConnexionODBC", by class "RequeteODBCwhere", distance 3.

Methods

connect `signature(objet = "RequeteODBCwheredate")`: Connexion to the database

Note

Inherits from RequeteODBCwhere and uses its connect method with a new SetAs

Author(s)

cedric.briand00@gmail.com

See Also

ConnexionODBC RequeteODBC RequeteODBCwhere

Examples

```
showClass("RequeteODBCwheredate")
```

---

tab2df *Function to transform a ftable into dataframe but just keeping the counts works with ftable of dim 2*

Description

Function to transform a ftable into dataframe but just keeping the counts works with ftable of dim 2

Usage

```
tab2df(tab)
```

Arguments

tab

Author(s)

Cedric Briand <cedric.briand00@gmail.com>

Index