

GUI in R

$\tau\rho$

Graphics User Interface in R

- Package :
 - Tcl/tk
 - Gtk+
 - Java

Using R package TCL/TK

- Element Techniques
 - Simple toplevel window with a button
 - Message boxes
 - File open/save dialogs
 - Pull-down menus and popup menus

Example 1

- Create Simple toplevel window with an Cancel button !
- Create a messages box with statement
`Welcome in the new world!`

Simple toplevel window with an Cancel button

```
require(tcltk)
tt <- tkoplevel()
C.but <- tkbutton(tt, text = "Cancel",
command = function() tkdestroy(tt))
tkgrid(C.but)
tkfocus(tt)
```

Creating Messages Box

- ```
GG <- tkmessageBox(title = "Greetings",
message = "Welcome in the new world!",
icon = "info", type = "ok")
```

# Example 2

- Write some statements to open file and save file

# Open File

```
fileName <- tclvalue(tkgetOpenFile())
if (!nchar(fileName)) {
tkmessageBox(message = "No file was selected!")
} else {
tkmessageBox(message = paste("The file selected
was", fileName))
}
```



# Save File

```
fileName2 <- tclvalue(tkgetSaveFile())
if (!nchar(fileName2)) {
tkmessageBox(message = "No file was selected!")
} else {
tkmessageBox(message = paste("The file selected
was", fileName2))
}
```

# Example 3

- Create a Pull Down Menus with menu title “File” that contains commands for opening, saving file and quit window

```
require(tcltk)
tt <- tktoplevel()
topMenu <- tkmenu(tt)
tkconfigure(tt, menu = topMenu)
fileMenu <- tkmenu(topMenu, tearoff = FALSE)
tkadd(fileMenu, "command", label = "Open File", command = function() {fileName <-
tclvalue(tkgetOpenFile())
if (!Inchar(fileName)) {
tkmessageBox(message = "No file was selected!")
} else {
tkmessageBox(message = paste("The file selected was", fileName))
}}
)
tkadd(fileMenu, "command", label = "Save File", command = function() {fileName2 <-
tclvalue(tkgetSaveFile())
if (!Inchar(fileName2)) {
tkmessageBox(message = "No file was selected!")
} else {
tkmessageBox(message = paste("The file selected was", fileName2))
}}
)

tkadd(fileMenu, "command", label = "Quit", command = function() tkdestroy(tt))
tkadd(topMenu, "cascade", label = "File", menu = fileMenu)
tkfocus(tt)
```

# Example 4

- Create a window for writing some comments with pop up menu “Copy”

```
require(tcltk)
tt <- tkoplevel()
txt <- tktext(tt)
tkgrid(txt)

copyText <- function()
.Tcl(paste("event", "generate", .Tcl.args(.Tk.ID(txt), "<<Copy>>"))
editPopupMenu <- tkmenu(txt, tearoff = FALSE)
tkadd(editPopupMenu, "command", label = "Copy <Ctrl-C>", command =
copyText)

RightClick <- function(x, y) { # x and y are the mouse coordinates
rootx <- as.integer(tkwininfo("rootx", txt)) # tkwininfo() return several
infos
rooty <- as.integer(tkwininfo("rooty", txt))
xTxt <- as.integer(x) + rootx
yTxt <- as.integer(y) + rooty

.Tcl(paste("tk_popup", .Tcl.args(editPopupMenu, xTxt, yTxt)))
}
tkbind(txt, "<Button-3>", RightClick) # Tcl recognizes three mouse
buttons

tkfocus(tt)
```