La construction de greffons RCommander

Milan Bouchet–Valat

Laboratoire de sociologie quantitative (LSQ-CREST)
Observatoire sociologique du changement (OSC-Sciences Po & CNRS)
Institut national d'études démographiques (INED)

nalimilan@club.fr

Rencontres R, Montpellier, 25 juin 2014

Programme

- Sous le capot de Rcmdr : Tcl/Tk en R
  - Principes
  - Comment trouver de la documentation
  - Un peu de pratique
- Les greffons Rcmdr :
  - Comment créer un paquet R
  - Les particularités de Rcmdr
  - Un peu plus de pratique
- Application à vos projets

Tcl/Tk

Tcl/Tk

- Boîte à outils (*toolkit*) créée en 1990/91 à Berkeley
- Heure de gloire dans les années 1990
- Avantages :
  - Multi-plateformes, relativement bien intégrée
  - Légère
  - Relativement complète, extensible
  - Assez bien conçue
- Limites :
  - Développement actuellement au ralenti
  - Certains widgets complexes manquants
  - Pas de possibilité de construire visuellement les interfaces
- Dans l'ensemble : très bien pour de petites interfaces, à éviter pour de grosses applications (préférer GTK ou Qt)

Tcl/Tk et R

- Paquet tcltk : publié en 2001 et toujours amélioré
- Intégré à la distribution standard
  - Sous Mac, il faut installer X11, et le thème n'est pas natif
  - Sous Linux aussi le thème par défaut est atroce
- Paquet tcltk2 (2006) : widgets additionnels
  - Thèmes pour Linux (passent aussi sous Mac)
  - Widget pour modifier des tables, etc.

Tcl/Tk et R : où trouver des infos

- Une seule page de documentation pour tous les widgets
- Peter Dalgaard, « A Primer on the R-Tcl/Tk Package » , *R News*, vol. 1(3), 2001.
- Peter Dalgaard, « Changes to the R-Tcl/Tk package », *R News,* vol. 2/3, 2002.
- Pas mal d'exemples le Web :
  - http://www.sciviews.org/_rgui/tcltk/
- Il faut utiliser la documentation pour d'autres langages et deviner comment l'adapter :
  - Manuel de référence, bien détaillé https://www.tcl.tk/man/tcl8.6/TkCmd/contents.htm
  - Doc de Tk moins austère, tutoriel : http://www.tkdocs.com
  - Doc pour Python (Tkinter) : https://docs.python.org/2/library/tkinter.html
- La meilleure solution : faire des recherches sur le Web, parfois adapter du code existant

Tcl/Tk et R : principe

- tcltk permet de lancer n'importe quelle commande Tcl : fonctions tcl() et .Tcl()
- Nombreuses fonctions pour la plupart des opérations courantes : nom commençant par tk
- Exemple de base : fenêtre avec un label

```
library(tcltk)
top <- tktoplevel()
lab <- tklabel(top, text="Un label")
tkgrid(lab)
```

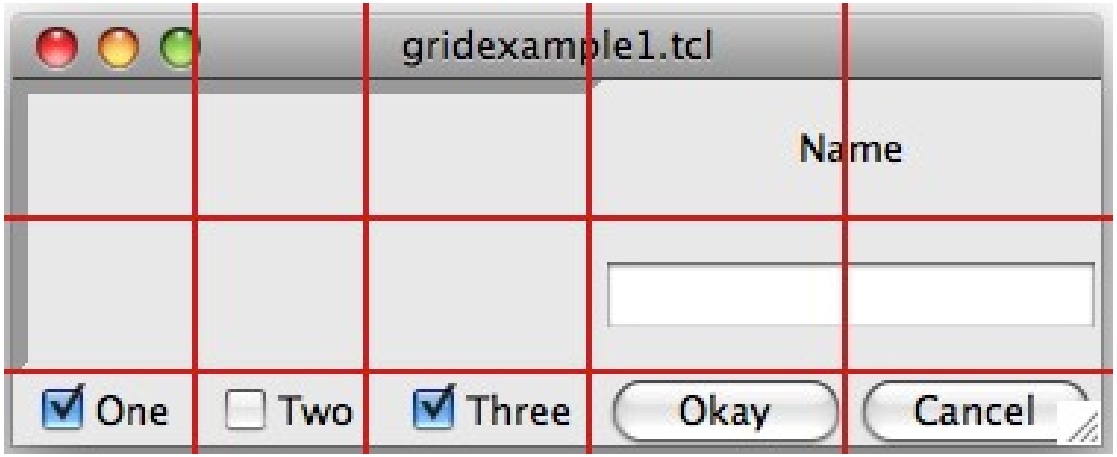
L’agencement des widgets

L'agencement des widgets

- Deux méthodes concurrentes et incompatibles :
  - pack(), ancienne commande, plus limitée : à éviter
  - grid(), plus riche : à préférer
  - Ne pas mélanger les deux !
- Pour le principe :
  http://www.tkdocs.com/tutorial/grid.html
  http://effbot.org/tkinterbook/grid.htm
- Pour la syntaxe en R :
  http://www.sciviews.org/_rgui/tcltk/Layout.html

L'agencement des widgets avec grid()

- Principe : attacher les widgets à un canevas
- Permet d'aligner les widgets horizontalement et verticalement
  - Chaque appel de grid() descend d'une ligne
  - Si on passe plusieurs widgets, chacun va dans une colonne, ex : `tkgrid(one, two, three, okay, cancel)`
  - Possibilité de préciser la ligne/colonne et l'étendue en lignes/colonnes, ex : `tkgrid(one, column=0, row=2)`



L'agencement des widgets avec grid()

Création des widgets :

```
top <- tktoplevel()
nameLabel <- ttklabel(top, text="Name")
nameEntry <- ttkentry(top)
checkButton1 <- ttkcheckbutton(top, text="One")
checkButton2 <- ttkcheckbutton(top, text="Two")
checkButton3 <- ttkcheckbutton(top, text="Three")
okButton <- ttkbutton(top, text="OK")
cancelButton <- ttkbutton(top, text="Cancel")
```

L'agencement des widgets avec grid()

Positionnement des widgets sur la grille :

```
tkgrid(nameLabel, row=0, column=3,
       columnspan=2)
tkgrid(nameEntry, row=1, column=3,
       columnspan=2)
tkgrid(checkButton1,
       checkButton2,
       checkButton3,
       okButton,
       cancelButton)
```

L'agencement des widgets avec grid()

Agencement plus soigné :

```
tkgrid(nameLabel, row=0, column=3,
       columnspan=2,
       padx=6, pady=6)
tkgrid(nameEntry, row=1, column=3,
       columnspan=2,
       padx=6, pady=6, sticky="ew")
tkgrid(checkButton1,
       checkButton2,
       checkButton3,
       okButton,
       cancelButton,
       padx=6, pady=6)
```

Quelques points utiles avec Tk

Tk et Ttk

- Ttk (*Themed Tk*) : widgets mieux intégrés à chaque plateforme
- Pour les widgets de base, on dispose d'une fonction tk* et d'une fonction ttk*
- Certains widgets n'existent que en version Ttk

```
ttkbutton       ttkimage          ttkpanedwindow
ttksizegrip     ttkcheckbutton    ttklabel
ttkprogressbar  ttktreeview       ttkcombobox
ttklabelframe   ttkradiobutton    ttkentry
ttkmenubutton   ttkscrollbar      ttkframe
ttknotebook     ttkseparator
```

Échanger des données avec Tk

- tclVar() : créer un objet R correspondant à une variable que l'on peut passer aux widgets Tk
- tclvalue() : récupérer/modifier la valeur contenue dans une variable Tk (ou retournée par une fonction Tk)

```
> x <- tclVar("une valeur")
> x
$env
<environment: 0x4d38e30>
attr(,"class")
[1] "tclVar"
> tclvalue(x)
[1] "une valeur"
```

  - Attention, les nombres deviennent des variables caractère : utiliser as.numeric(tclvalue(x))
  - Pour vérifier qu'une valeur entrée par l'utilisateur est bien un nombre valide, on teste is.na(as.numeric(tclvalue(x)))

Ouvrir et enregistrer des fichiers

- Ouvrir un fichier :

```
filename <- tclvalue(tkgetOpenFile())

if (filename == "")
    tkmessageBox(message="Aucun fichier sélectionné.")
else
    tkmessageBox(message=paste("Fichier sélectionné :",
                               filename))
```

- Sélectionner un dossier :

```
tclvalue(tkchooseDirectory())
```

- Enregistrer un fichier :

```
tclvalue(tkgetSaveFile())
```

- Autres options :

```
tclvalue(tkgetSaveFile(initialfile="fichier.txt",
  filetypes="{{Fichiers texte} {.txt}} {{Tous les fichiers} *}"))
```

Écrire des greffons Rcmdr

Les greffons Rcmdr

- John Fox,
  « The R Commander: A basic-statistics graphical user interface to R » ,
  Journal of Statistical Software, 14(9), 2003.
- John Fox, « Extending the R Commander by "plug-in" packages », *R News*, 7(3), 2003.
- Plus de 20 greffons
- Chaque greffon doit être un paquet R
  - Convention : nommer ces paquets RcmdrPlugin.*
  - Charger le paquet lance Rcmdr avec le greffon
  - Possibilité de charger les greffons depuis l'interface graphique
  - Les menus installés par les greffons sont visibles uniquement lorsque le greffon est chargé

La structure d'un paquet R

- http://cran.r-project.org/doc/manuals/R-exts.html
- DESCRIPTION : informations sur le paquet et ses dépendances
- NAMESPACE : fonctions à importer/exporter
- man/ : documentation
- tests/ : tests (optionnel, difficile avec les interfaces graphiques)
- inst/ : fichiers divers à installer
  - inst/etc/menus.txt : menus Rcmdr
  - inst/po/ : traductions compilées
- po/ : traductions non compilées
- R/ : code source R

Les fonctions et widgets Rcmdr

Les fonctions Rcmdr essentielles

- Malheureusement peu documentées
  - On peut les trouver avec ?Rcmdr.Utilities ou dans R/utilities.R
  - S'inspirer d'une boîte de dialogue qui ressemble à ce que vous voulez faire
- doItAndPrint() : lancer une commande et l'ajouter au script
- initializeDialog(), dialogSuffix(), OKCancelHelp(), closeDialog() : préparer une boîte de dialogue et la refermer
- putDialog()/getDialog() pour enregistrer les paramètres entrés par l'utilisateur
- setBusyCursor()/setIdleCursor() : indiquer que le programme est occupé

Les fonctions Rcmdr essentielles

- Séquence typique pour une boîte de dialogue :
  - initializeDialog(), getDialog() et création des différents widgets
  - Définition de la fonction onOK()
    - Validation des valeurs entrées par l'utilisateur
    - putDialog() avec les valeurs entrées
    - Puis closeDialog() et doItAndPrint() pour lancer le code
  - OKCancelHelp(), agencement des widgets avec grid() puis dialogSuffix()

Les widgets Rcmdr essentiels

- radioButtons() : choisir parmi plusieurs options incompatibles

```
buttons <- radioButtons(top, name="x",
                buttons=c("one", "two", "three"),
                labels=c("Un", "Deux", "Trois"),
                title="Un titre",
                initialValue="1")
tkgrid(getFrame(buttons))
tclvalue(xvariable)
```

- checkBoxes() : choisir plusieurs options compatibles

```
checkBoxes("boxes",
           boxes=c("one", "two", "three"),
           initialValues=c(1, 0, 0),
           labels=c("Un", "Deux", "Trois"),
           title="Un titre")
tkgrid(boxesFrame)
tclvalue(oneVariable)
```

Les widgets Rcmdr essentiels

- variableListBox() : choisir un ou plusieurs éléments dans une liste

```
vBox <- variableListBox(top,
      title="Choisir une variable")
tkgrid(vBox)
getSelection(vBox)
```

  - Arguments optionnels :
    - variableList : liste des éléments à proposer
    - initialSelection : numéro de l'élément à préselectionner
    - selectmode : "single" ou "multiple"

Deux widgets Tk utiles

- tkscale() : choisir un nombre dans un intervalle prédéfini

```
x <- tclVar(1)
tkscale(top, from=1, to=1500,
                    showvalue=TRUE, variable=x,
                    resolution=1,
orient="horizontal")
tclvalue(x)
```

- tkspin() : entrer n'importe quel nombre

```
x <- tclVar(1)
tkwidget(top, type="spinbox", from=0, to=100,
          inc=0.1, textvariable=x)
tclvalue(x)
```

D’autres fonctions Rcmdr

- Variables() : obtenir la liste des variables
  - numericP(), factorsP(), twoLevelFactorsP(), modelsP(), lmP()
  - fonctions list* pour obtenir la liste des variables, modèles…
- groupsBox() : choisir un facteur suivant lequel grouper une opération