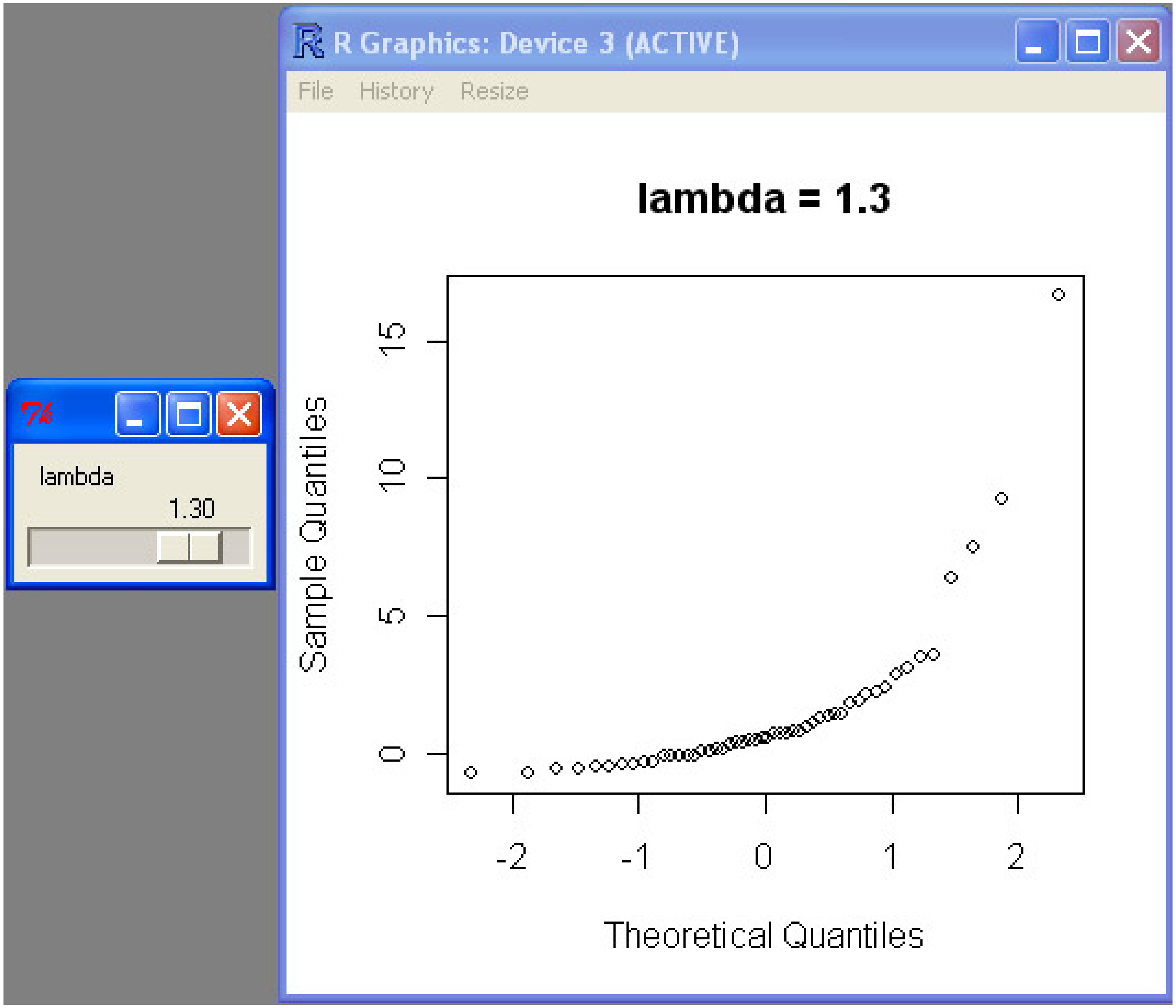
Statistical Cartoons

rpanel: simple interactive controls for R functions using the tcltk package

Ewan Crawford, Adrian Bowman

Department of Statistics, The University of Glasgow, UK

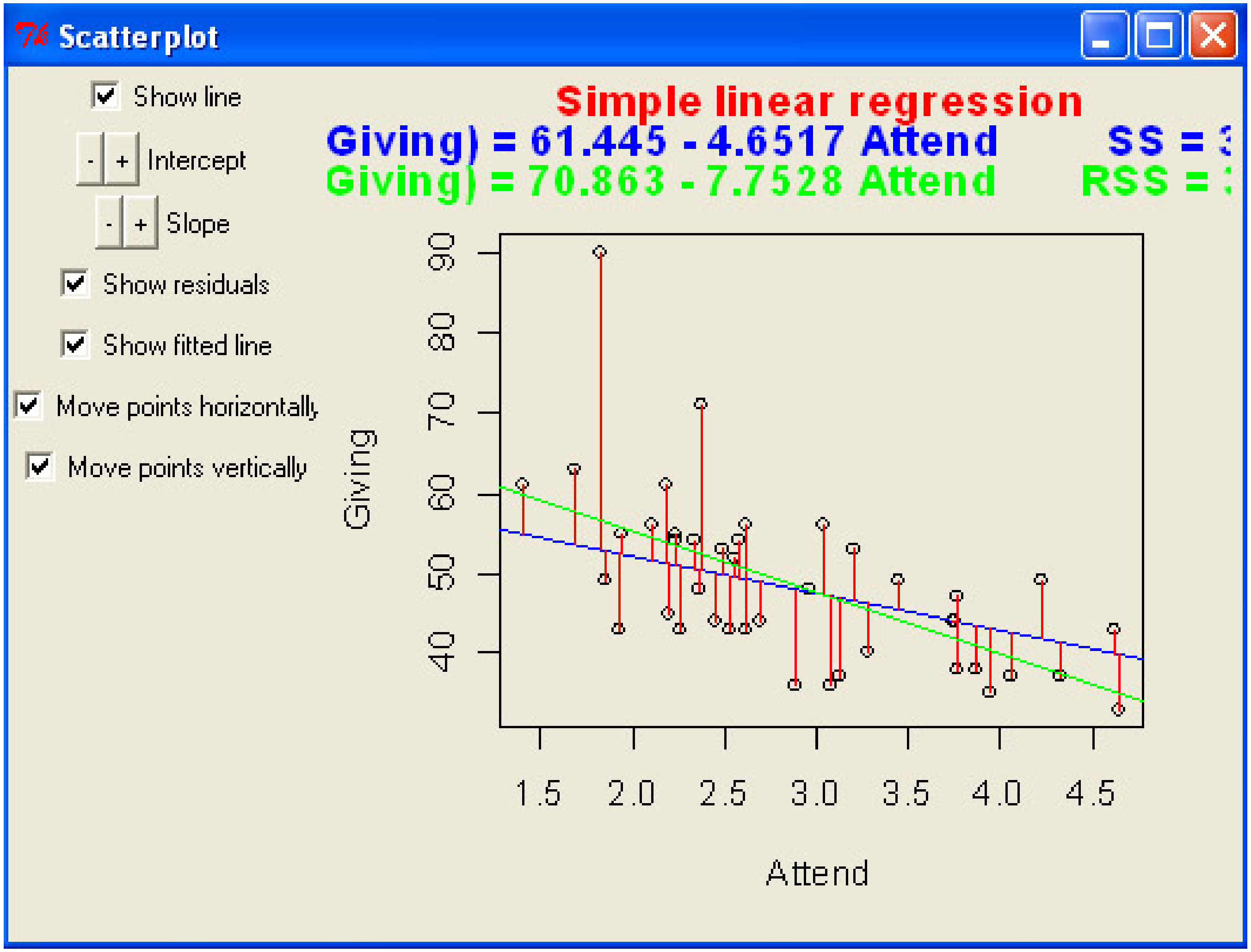
Elementary example



Code for example

```
library(rpanel)
x11(width=4,height=4)
qq.draw <- function(panel) {
   z <- bc.fn(panel$y, panel$lambda)
   qqnorm(z, main = paste("lambda =",
     round(panel$lambda, 2)))
   panel
   }
panel <- rp.control(y = exp(rnorm(50)), lambda = 1)
rp.slider(panel, lambda, -2, 2, qq.draw,
  showvalue = TRUE)
```

Second elementary example



☑ Show line

- + Intercept

- + Slope

☑ Show residuals

☑ Show fitted line

☑ Move points horizontally

☑ Move points vertically

Giving) = 61.445 - 4.6517 Attend SS = 3

Giving) = 70.863 - 7.7528 Attend RSS = 3

Second elementary example

Relationship with wider gui packages

```
Relationship with wider ("widgety") gui packages
- Rgtk2
- gWidgets
- playwith
- rwxwidgets
- JGR
- rtcltk


...
```

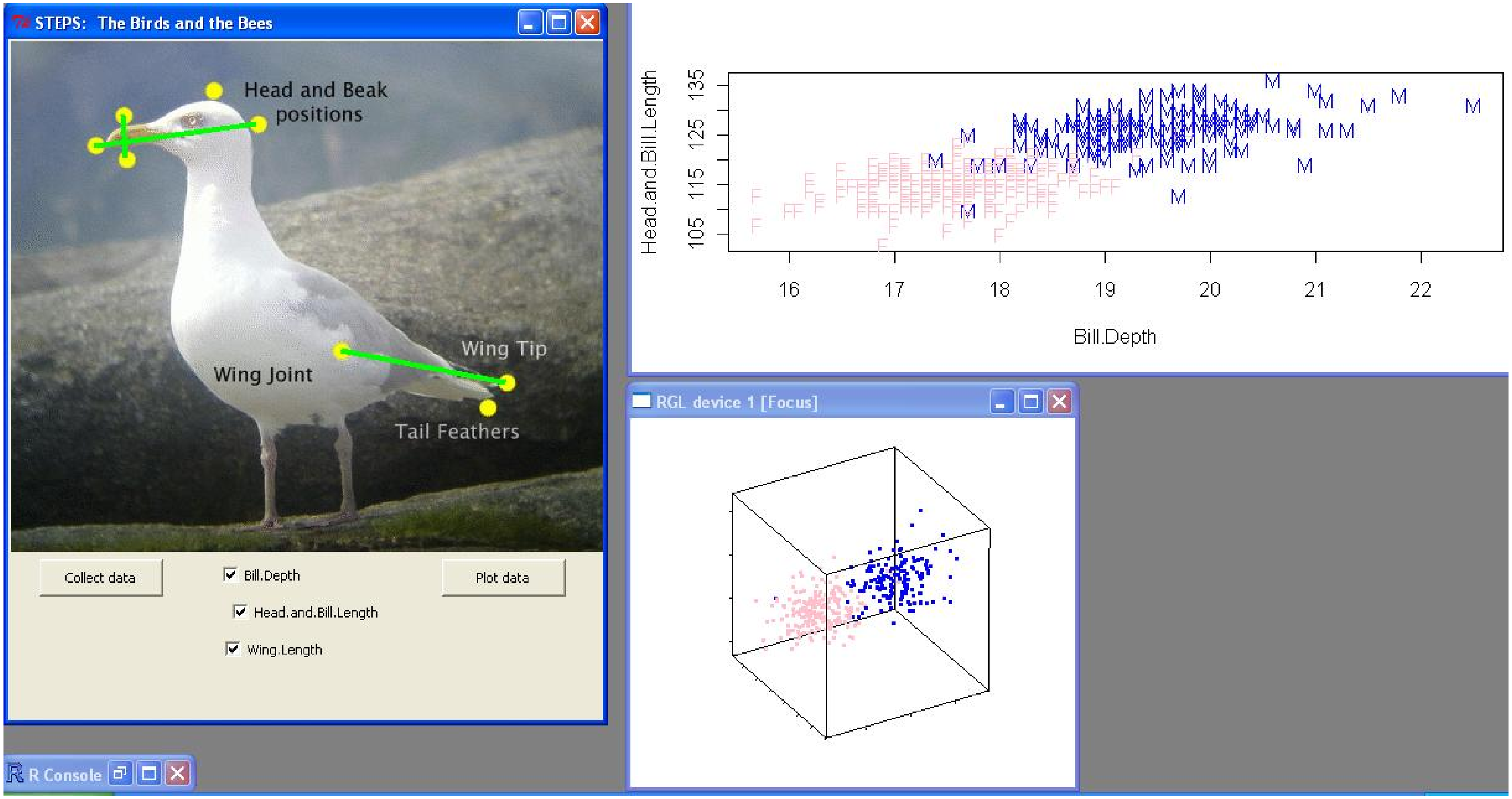
Indication of tools available

What we offer in terms of widgets

- panel/window
- button (with repeat)
- slider/scrollbar
- radiogroup
- textentry (or multiples)
- checkbox (or multiples)
- listbox
- doublebutton
- image
- line
- messagebox
- tkrplot

...

Gulls

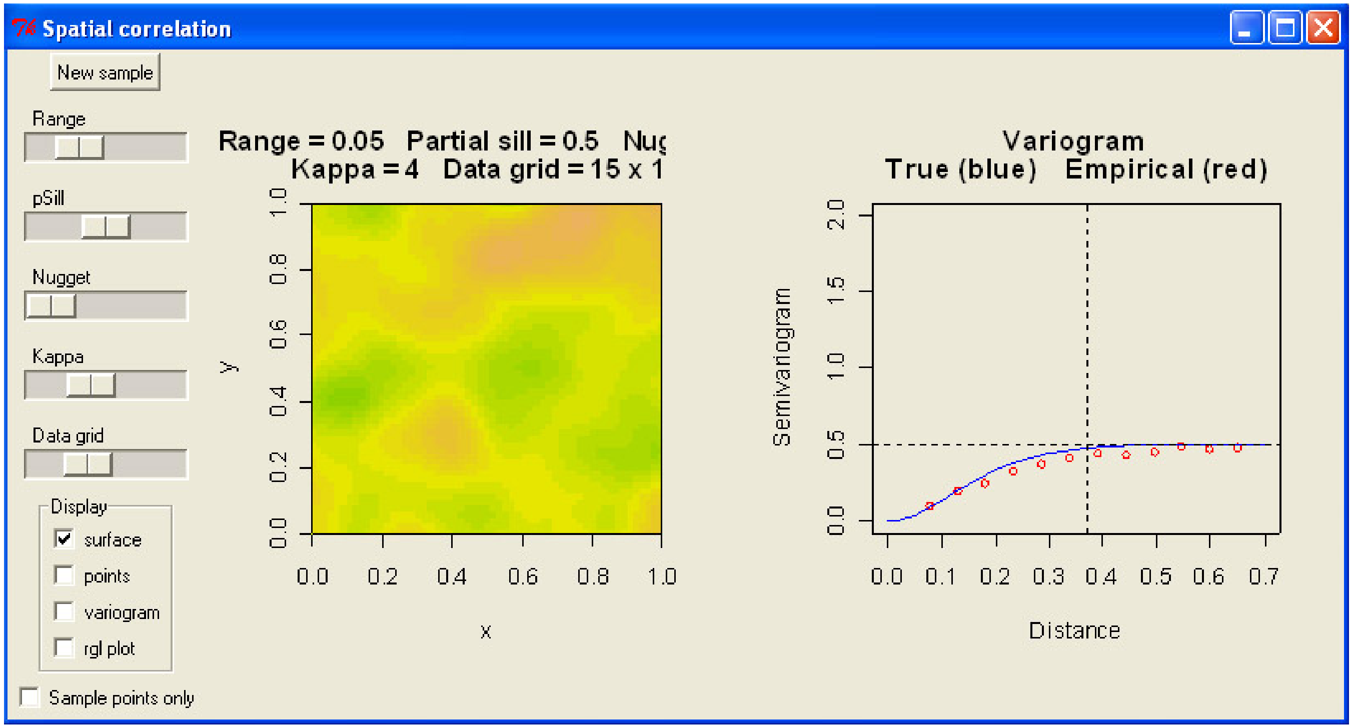


☑ Bill.Depth

☑ Head.and.Bill.Length

☑ Wing.Length

Spatial examples: Geosim



Display

☑ surface

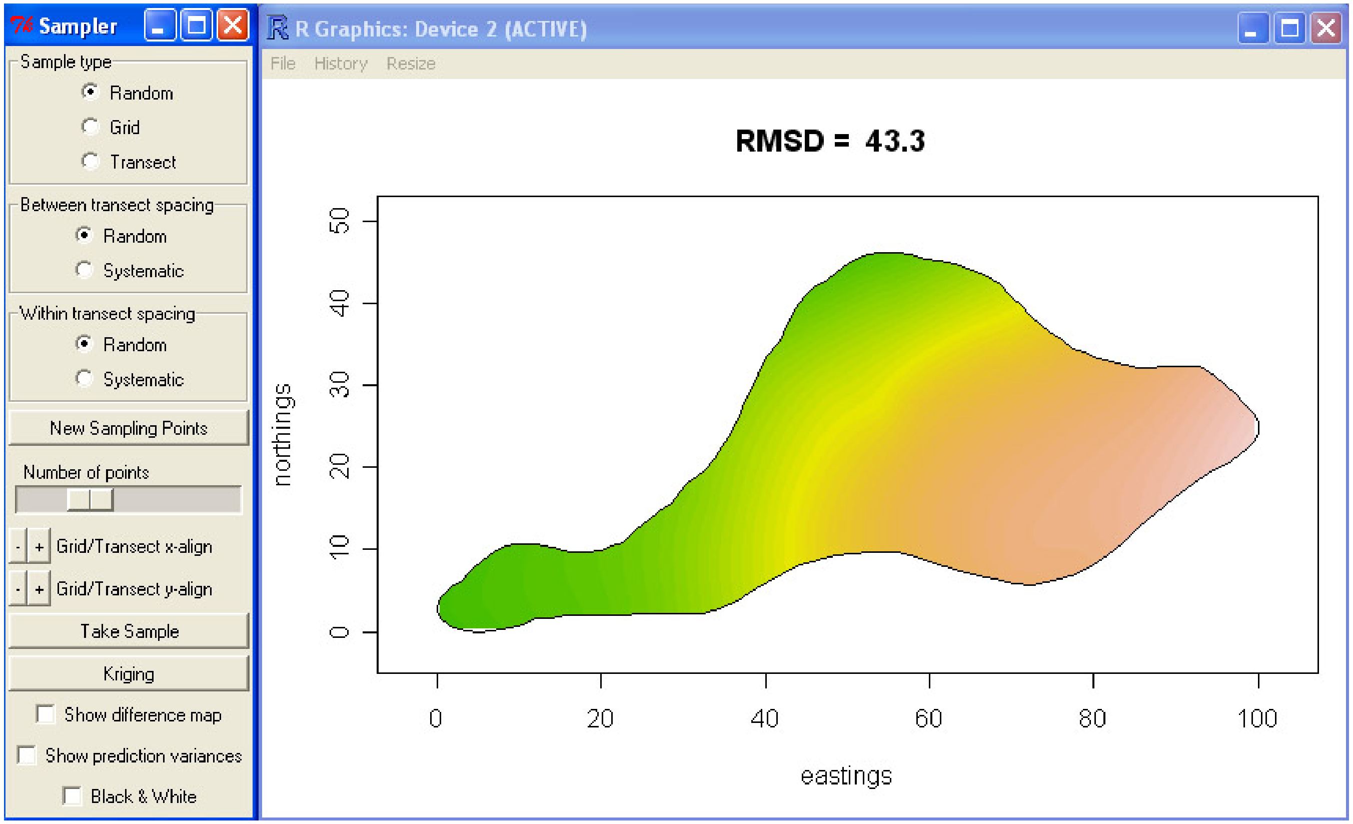
☐ points

☐ variogram

☐ rgl plot

☐ Sample points only

Spatial examples: Mururoa



Sample type
● Random
○ Grid
○ Transect

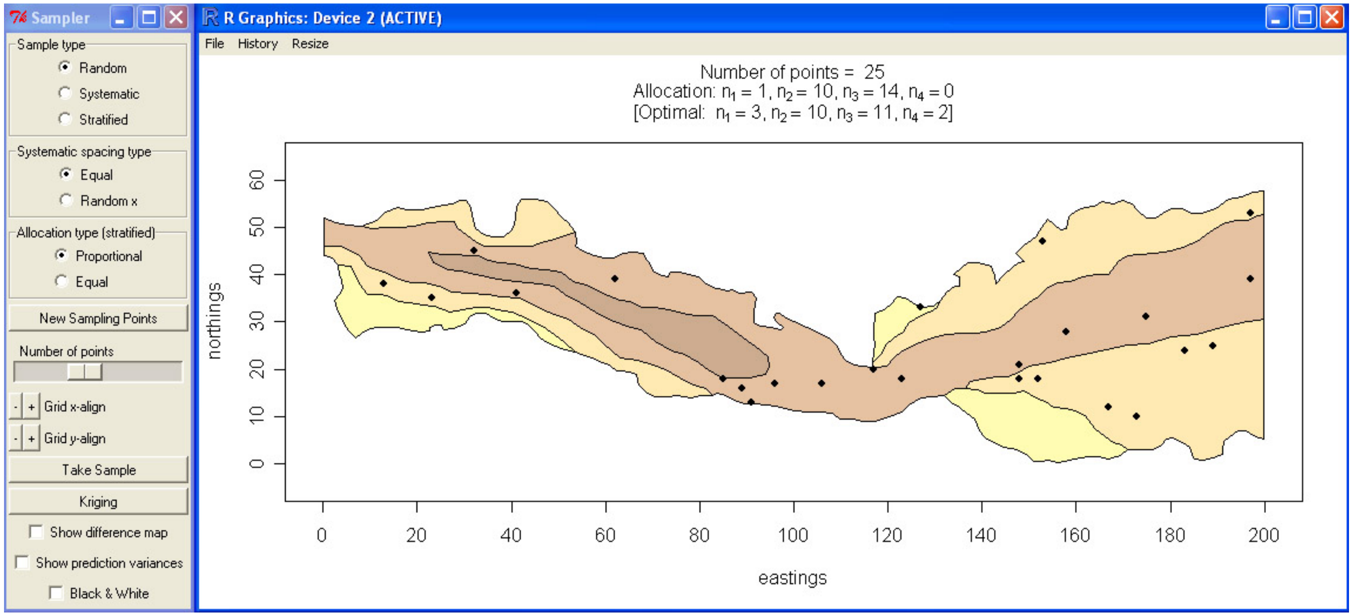
Between transect spacing
● Random
○ Systematic

Within transect spacing
● Random
○ Systematic

□ Show difference map
□ Show prediction variances
□ Black & White

RMSD = 43.3

Spatial examples: Rosyth



Sample type
- ● Random
- ○ Systematic
- ○ Stratified

Systematic spacing type
- ● Equal
- ○ Random x

Allocation type (stratified)
- ● Proportional
- ○ Equal

- □ Show difference map
- □ Show prediction variances
- □ Black & White

Number of points = 25

Allocation: $n_1 = 1, n_2 = 10, n_3 = 14, n_4 = 0$

[Optimal: $n_1 = 3, n_2 = 10, n_3 = 11, n_4 = 2$]

Current developments

```
Current developments
- multi-line text box
- combo box
- pos/grid
- tabbed notebook
- fonts
- slider group

...

- essentially all BWidgets and TkTable

- More Cartoons
```

Web site

www.stats.gla.ac.uk/~adrian/rpanel

ewan@stats.gla.ac.uk

adrian@stats.gla.ac.uk

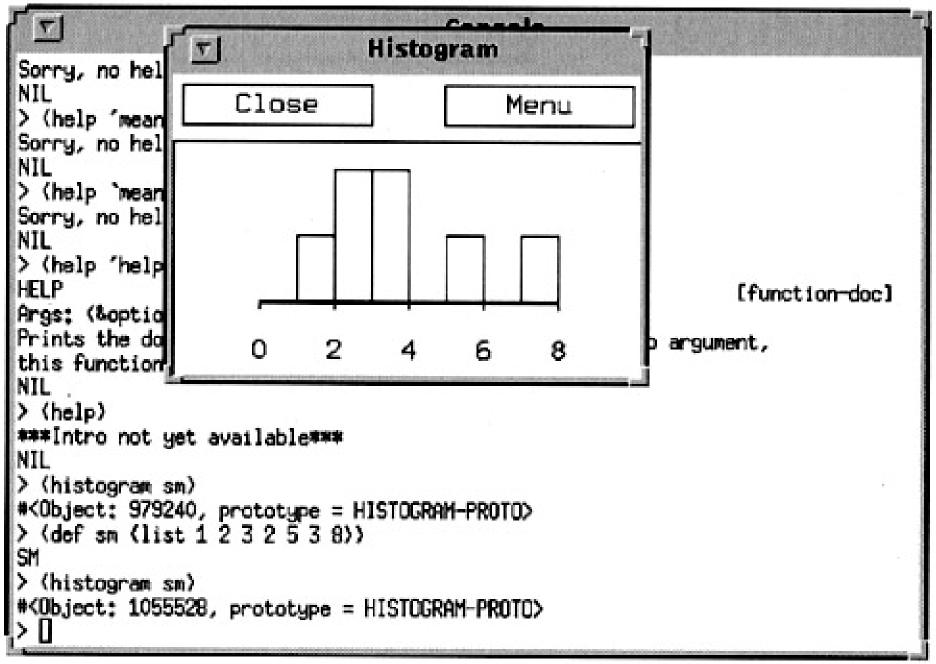
-

What

```
- what i’ll talk about, widgets and their uses
- cartoons explanation
```

Why

- applications
- rationale (teach dept etc etc)
- xlispstat etc



```
Sorry, no hel
NIL
> (help 'mean
Sorry, no hel
NIL
> (help `mean
Sorry, no hel
NIL
> (help 'help
HELP                                                        [function-doc]
Args: (&optio
Prints the do                                       o argument,
this function
NIL
> (help)
***Intro not yet available***
NIL
> (histogram sm)
#<Object: 979240, prototype = HISTOGRAM-PROTO>
> (def sm (list 1 2 3 2 5 3 8))
SM
> (histogram sm)
#<Object: 1055528, prototype = HISTOGRAM-PROTO>
> ▯
```

soa2

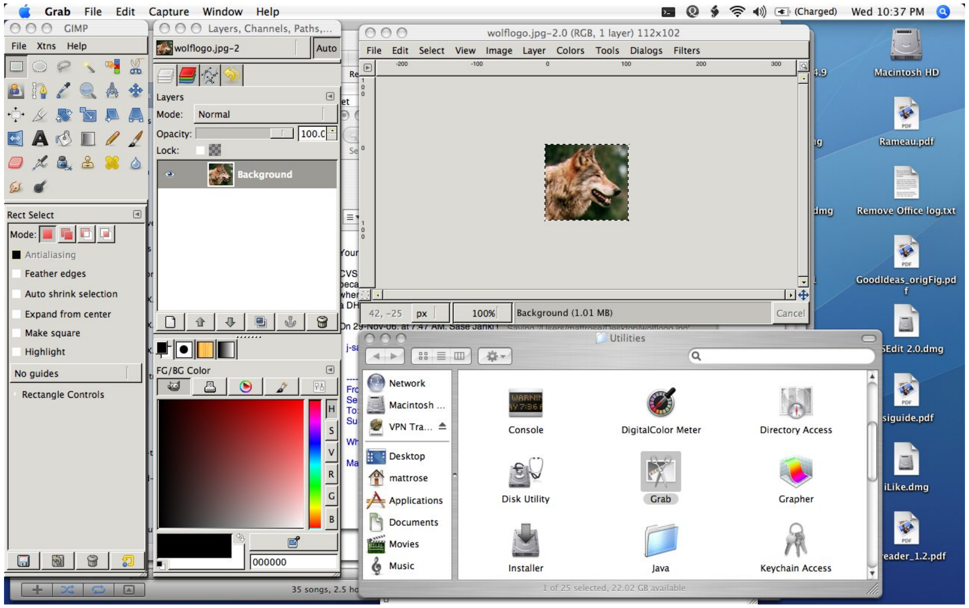
Big heading: RGtk2

Sub heading: GTK+

Project page:
http://www.gtk.org

An example of what can be done with GTK+, The GIMP

Illustrated by image



Mode: Normal

Opacity: 100.0

wolflogo.jpg-2.0 (RGB, 1 layer) 112x102

Background (1.01 MB)

1 of 25 selected, 22.02 GB available

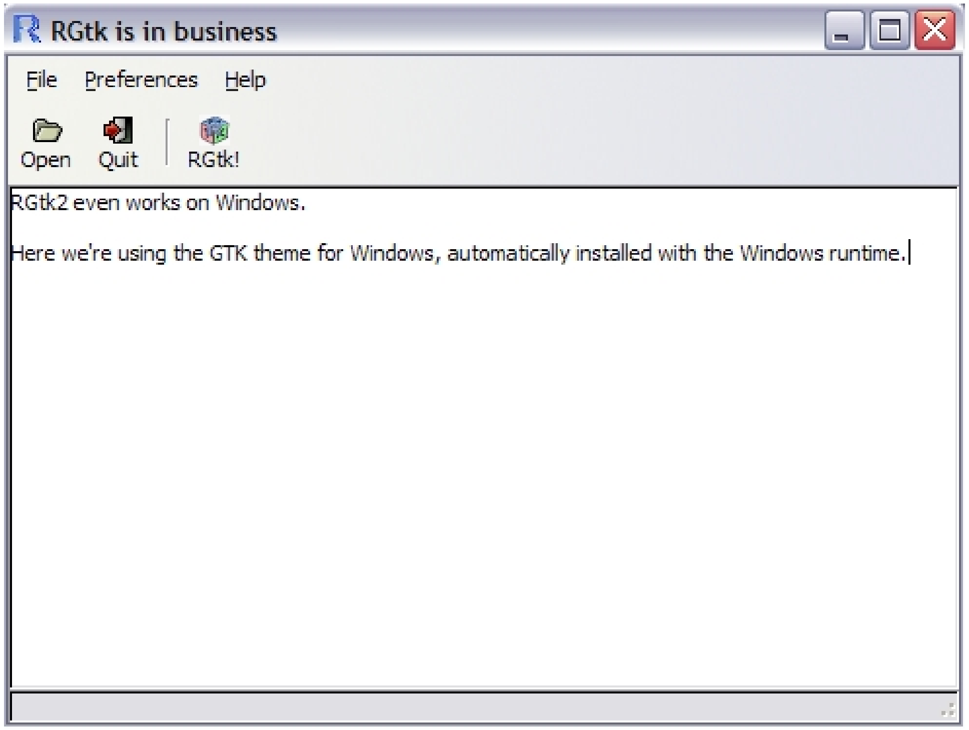
soa2

Sub heading: RGtk2

Project page:
http://www.ggobi.org/rgtk2/

Example of what can be done with Rgtk2, demo

Illustrated by image



RGtk2 even works on Windows.

Here we're using the GTK theme for Windows, automatically installed with the Windows runtime.

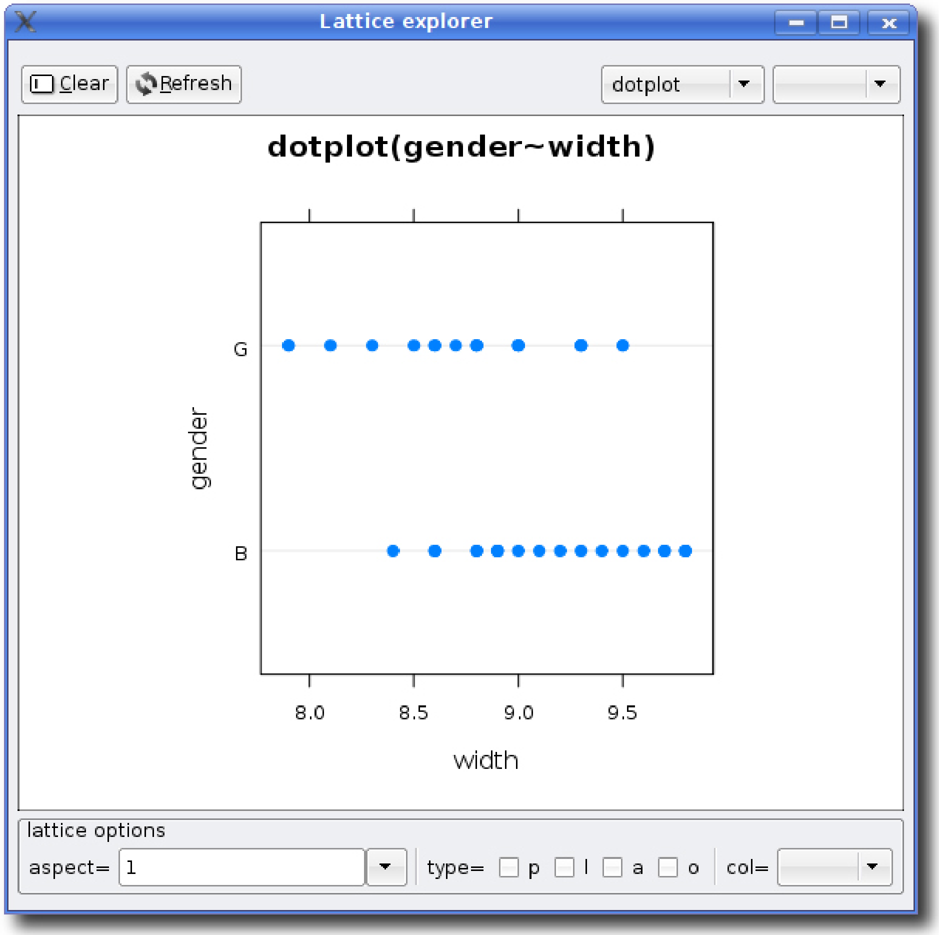
soa2

** Third slide: (May not all fit on the one slide, see r
about omitting the code below)

Big heading: Gwidgets

Project page:
http://wiener.math.csi.cuny.edu/pmg/gWidgets

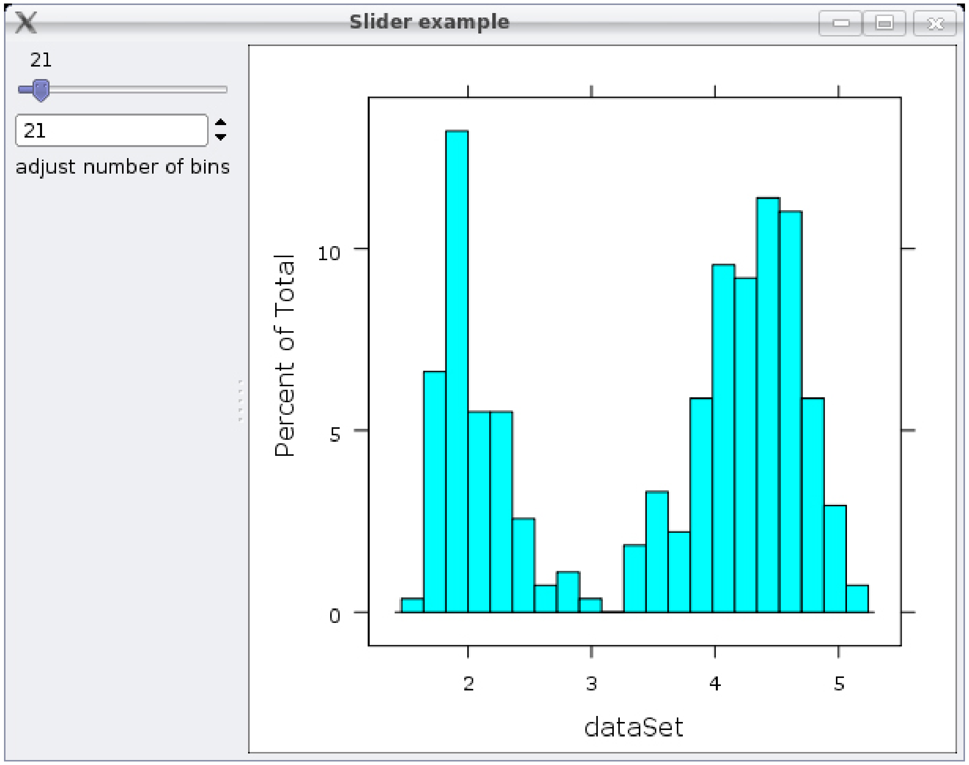
Illustrated by images



aspect= 1

type= □ p □ l □ a □ o

col=



21

21

adjust number of bins

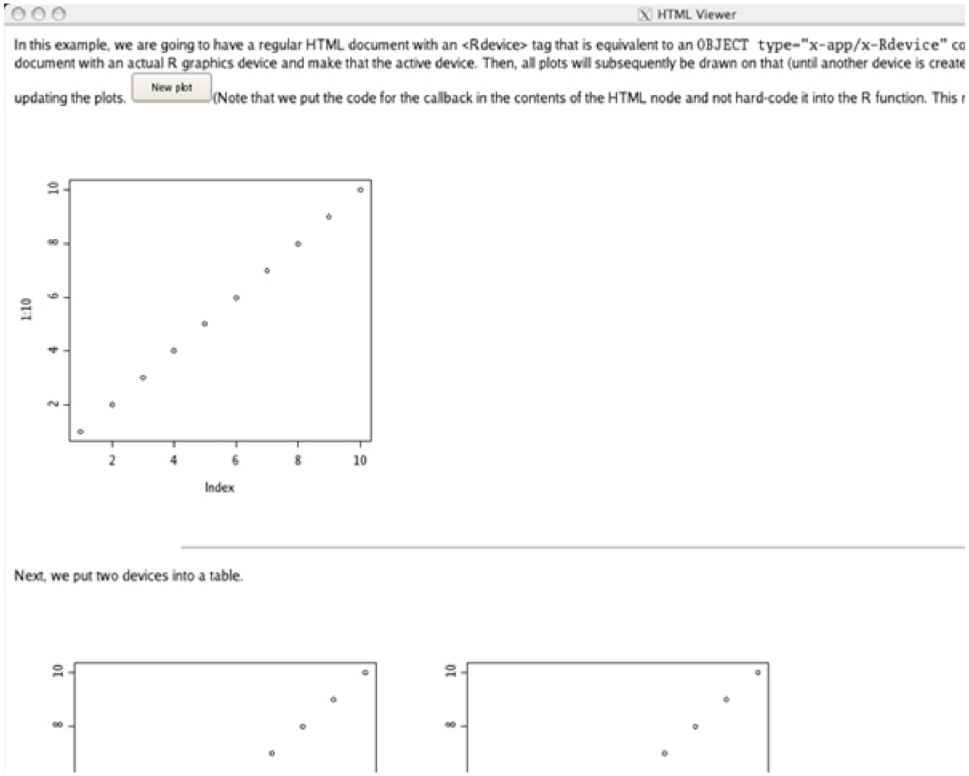
Gwidgets code example (for the slider image)
Adrian - if you see fit, do not include this code. It ma
including out of fairness as gwidgets is the only easy a
using rpanel

```
library(gWidgets)
```

```
options("guiToolkit"="RGtk2")
require(lattice)
dataSet = faithful$eruptions
w = gwindow("Slider example")
g = gpanedgroup(cont=w)
g1 = ggroup(horizontal = FALSE, cont = g) # first is lef
gg = ggraphics(cont = g)                     # second is

sl <- gslider(1, length(dataSet), by = 1, cont=g1, handl
function(h,...) {     print(histogram(dataSet, nint = sv
    svalue(sb) <- svalue(h$obj)
  })

sb <- gspinbutton(1, length(dataSet), by = 1, cont=g1, h
function(h,...) {
    print(histogram(dataSet, nint = svalue(h$obj)))
    svalue(sl) <- svalue(h$obj)
  })

l = glabel("adjust number of bins", cont=g1)
histogram(dataSet)                      # start it off
```

soa2

** Slide Four
Big heading: rwxwidgets
Project page:
http://www.omegahat.org/RwxWidgets/
Illustrate rwxwidgets image:



In this example, we are going to have a regular HTML document with an <Rdevice> tag that is equivalent to an OBJECT type="x-app/x-Rdevice" co
document with an actual R graphics device and make that the active device. Then, all plots will subsequently be drawn on that (until another device is create
updating the plots. New plot (Note that we put the code for the callback in the contents of the HTML node and not hard-code it into the R function. This r

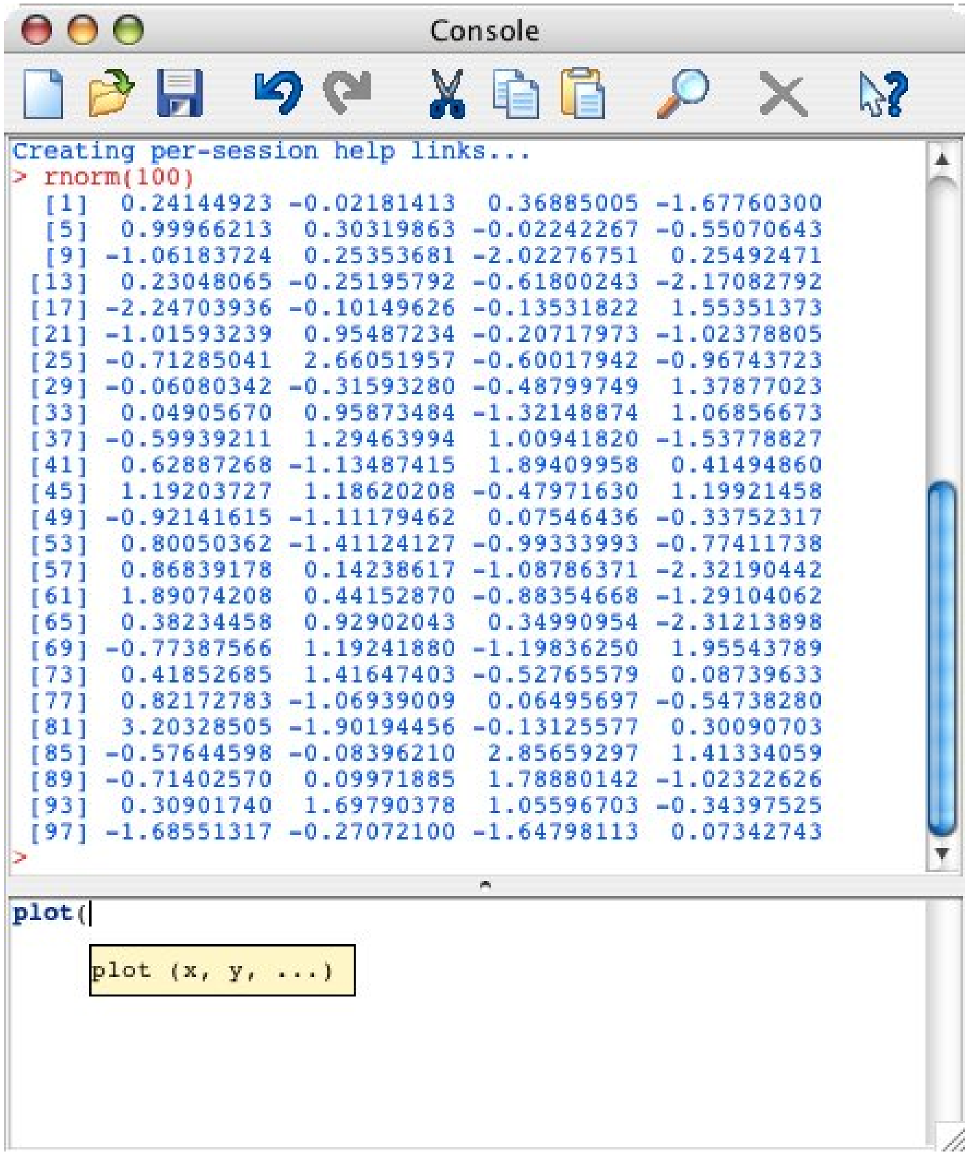
Next, we put two devices into a table.

** Slide Five

Big heading: JGR

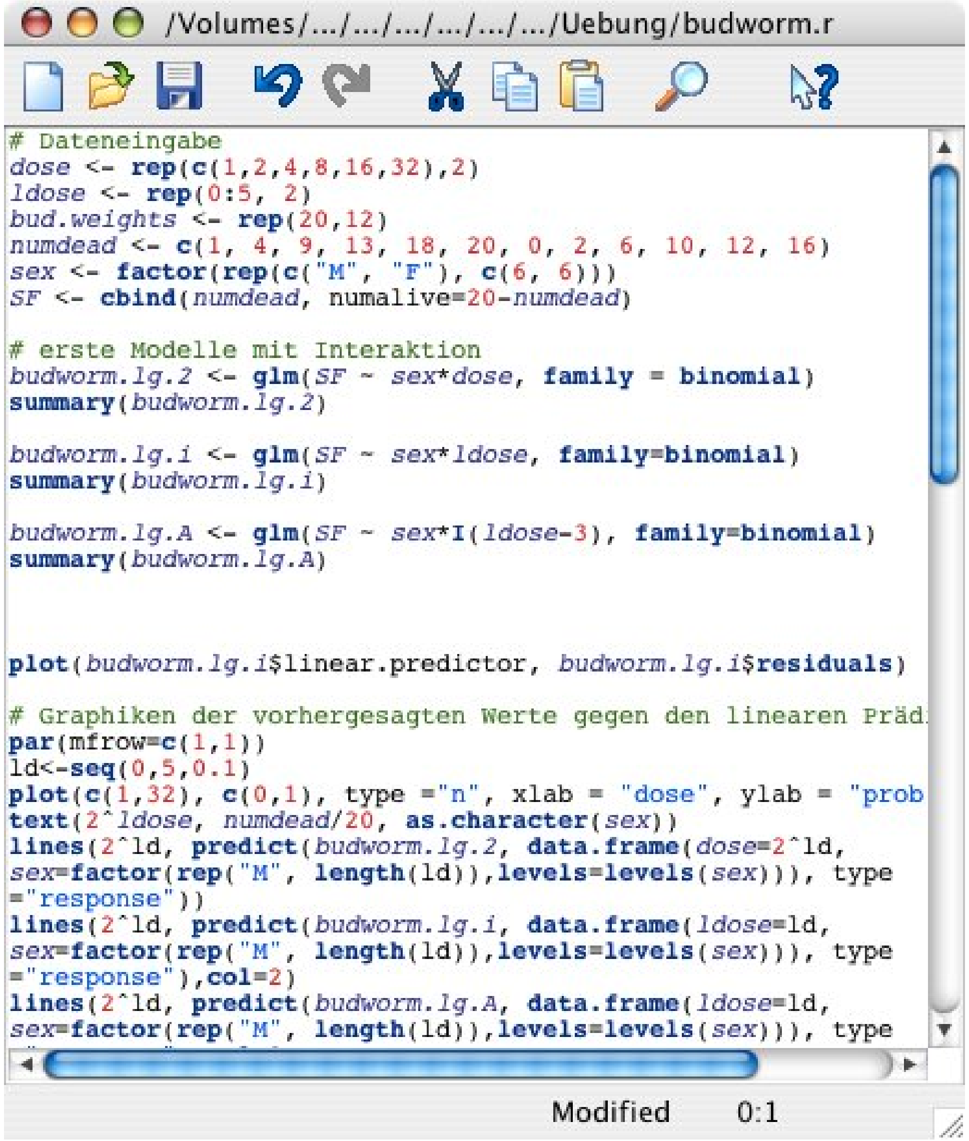
Project page:
http://rosuda.org/JGR/

Illustrations:



```
Creating per-session help links...
> rnorm(100)
  [1]  0.24144923 -0.02181413  0.36885005 -1.67760300
  [5]  0.99966213  0.30319863 -0.02242267 -0.55070643
  [9] -1.06183724  0.25353681 -2.02276751  0.25492471
 [13]  0.23048065 -0.25195792 -0.61800243 -2.17082792
 [17] -2.24703936 -0.10149626 -0.13531822  1.55351373
 [21] -1.01593239  0.95487234 -0.20717973 -1.02378805
 [25] -0.71285041  2.66051957 -0.60017942 -0.96743723
 [29] -0.06080342 -0.31593280 -0.48799749  1.37877023
 [33]  0.04905670  0.95873484 -1.32148874  1.06856673
 [37] -0.59939211  1.29463994  1.00941820 -1.53778827
 [41]  0.62887268 -1.13487415  1.89409958  0.41494860
 [45]  1.19203727  1.18620208 -0.47971630  1.19921458
 [49] -0.92141615 -1.11179462  0.07546436 -0.33752317
 [53]  0.80050362 -1.41124127 -0.99333993 -0.77411738
 [57]  0.86839178  0.14238617 -1.08786371 -2.32190442
 [61]  1.89074208  0.44152870 -0.88354668 -1.29104062
 [65]  0.38234458  0.92902043  0.34990954 -2.31213898
 [69] -0.77387566  1.19241880 -1.19836250  1.95543789
 [73]  0.41852685  1.41647403 -0.52765579  0.08739633
 [77]  0.82172783 -1.06939009  0.06495697 -0.54738280
 [81]  3.20328505 -1.90194456 -0.13125577  0.30090703
 [85] -0.57644598 -0.08396210  2.85659297  1.41334059
 [89] -0.71402570  0.09971885  1.78880142 -1.02322626
 [93]  0.30901740  1.69790378  1.05596703 -0.34397525
 [97] -1.68551317 -0.27072100 -1.64798113  0.07342743
>
```

```
plot(
```



```
# Dateneingabe
dose <- rep(c(1,2,4,8,16,32),2)
ldose <- rep(0:5, 2)
bud.weights <- rep(20,12)
numdead <- c(1, 4, 9, 13, 18, 20, 0, 2, 6, 10, 12, 16)
sex <- factor(rep(c("M", "F"), c(6, 6)))
SF <- cbind(numdead, numalive=20-numdead)

# erste Modelle mit Interaktion
budworm.lg.2 <- glm(SF ~ sex*dose, family = binomial)
summary(budworm.lg.2)

budworm.lg.i <- glm(SF ~ sex*ldose, family=binomial)
summary(budworm.lg.i)

budworm.lg.A <- glm(SF ~ sex*I(ldose-3), family=binomial)
summary(budworm.lg.A)



plot(budworm.lg.i$linear.predictor, budworm.lg.i$residuals)

# Graphiken der vorhergesagten Werte gegen den linearen Präd
par(mfrow=c(1,1))
ld<-seq(0,5,0.1)
plot(c(1,32), c(0,1), type ="n", xlab = "dose", ylab = "prob
text(2^ldose, numdead/20, as.character(sex))
lines(2^ld, predict(budworm.lg.2, data.frame(dose=2^ld,
sex=factor(rep("M", length(ld)),levels=levels(sex))), type
="response"))
lines(2^ld, predict(budworm.lg.i, data.frame(ldose=ld,
sex=factor(rep("M", length(ld)),levels=levels(sex))), type
="response"),col=2)
lines(2^ld, predict(budworm.lg.A, data.frame(ldose=ld,
sex=factor(rep("M", length(ld)),levels=levels(sex))), type
```

** Slide Five

** Slide Six

rtcltk

Pages:
http://bioinf.wehi.edu.au/~wettenhall/RTclTkExamples/

Illustative image: rtcltk_editbox2.jpg

soa

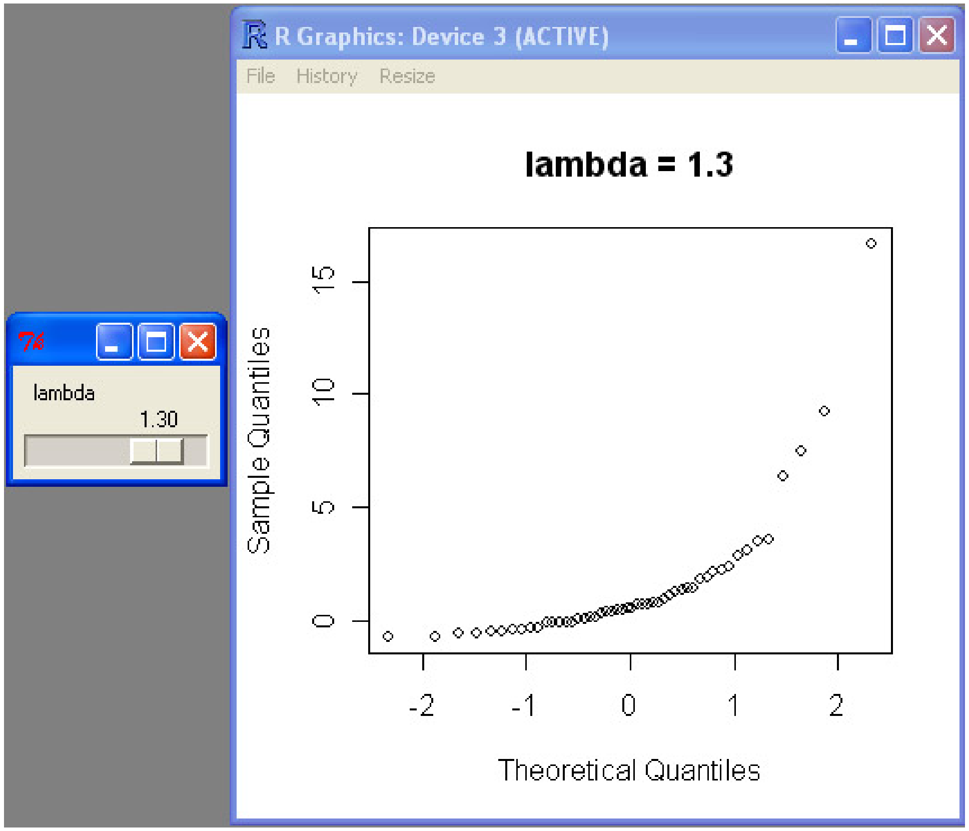
```
- ---as per lanc --- - get images -
+ new -> Playwith
```

method

- who did what
  - adrian b, simon urbanek, gavin, richard b
- tcl/tk and internals
- list of controls/widgets/facilities (but not exhaust

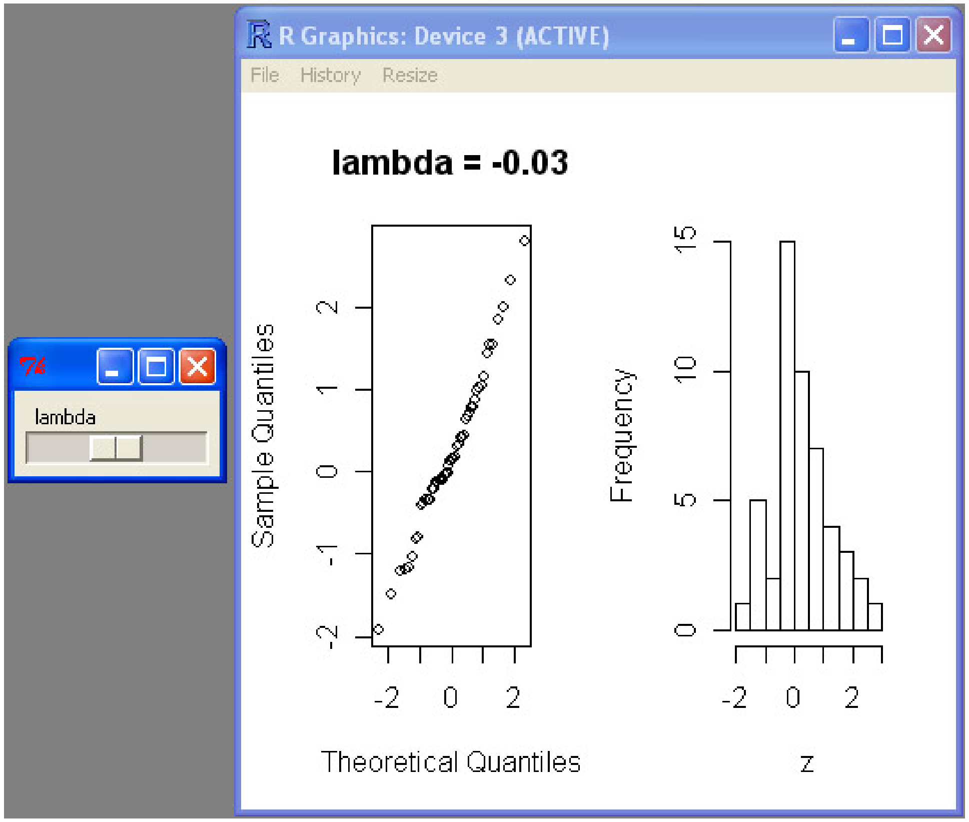
examples

- preview of examples - click to run - check what runs



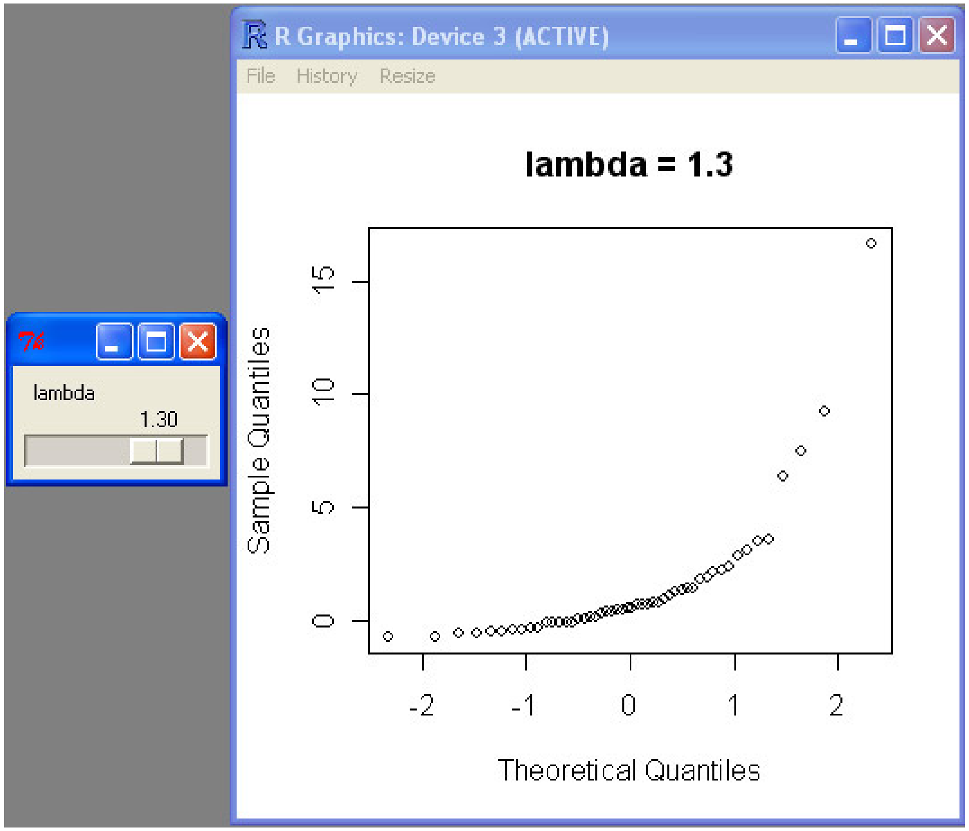
examples

- preview of examples - click to run - check what runs



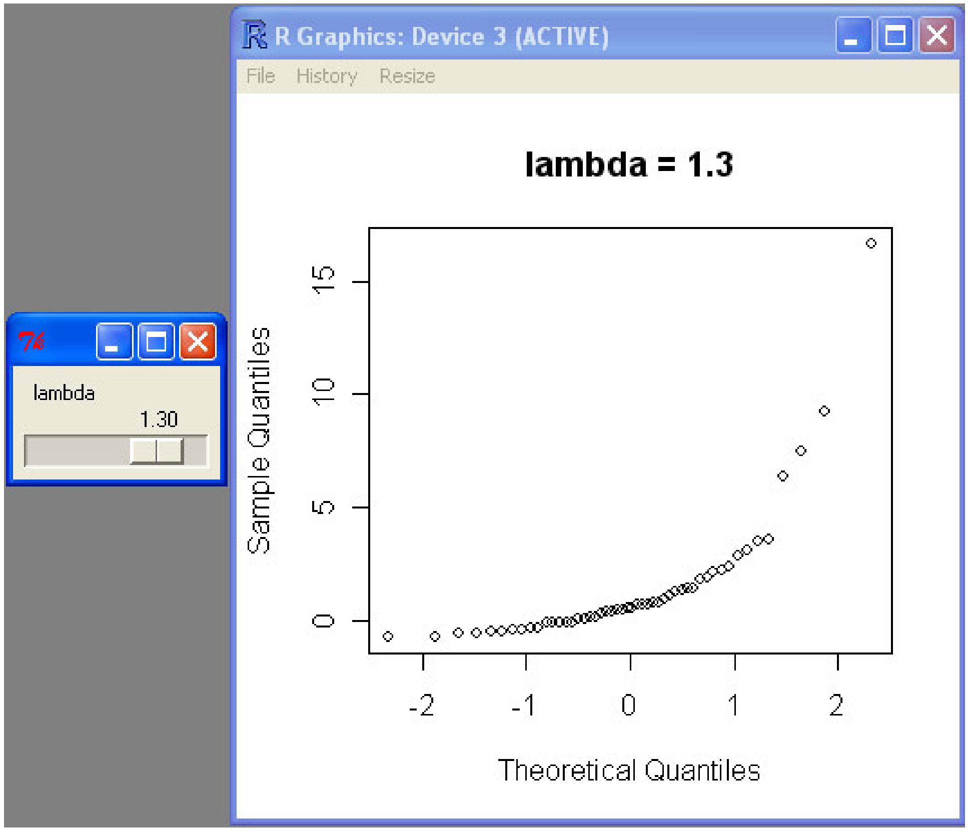
examples

- `tables(panel)`



examples

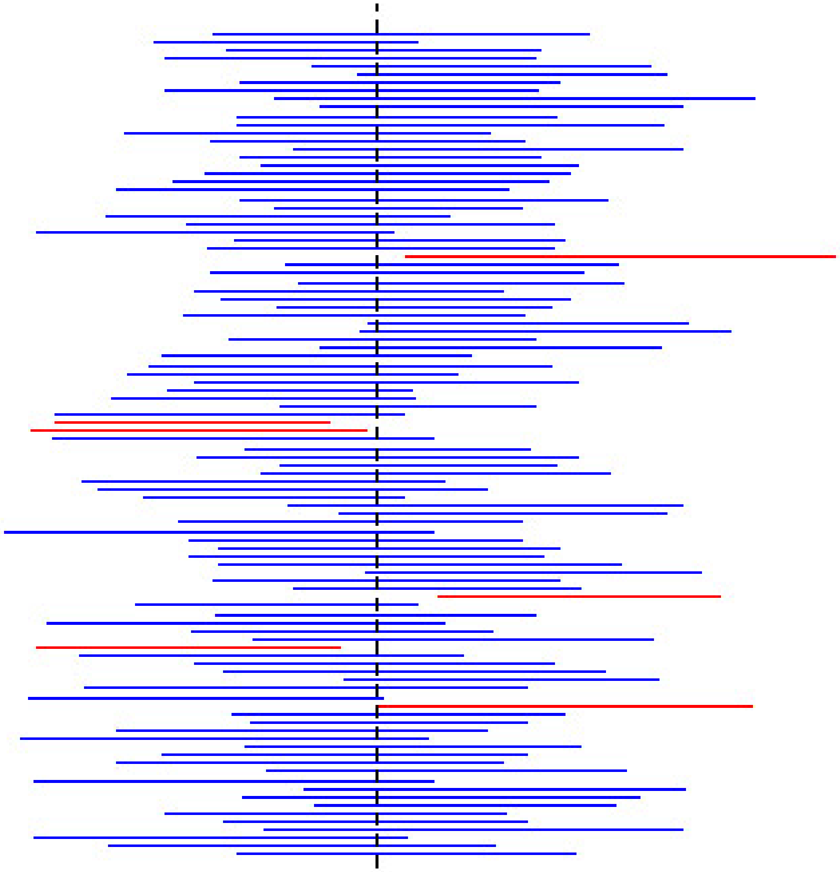
- normal fitting



lambda 1.30

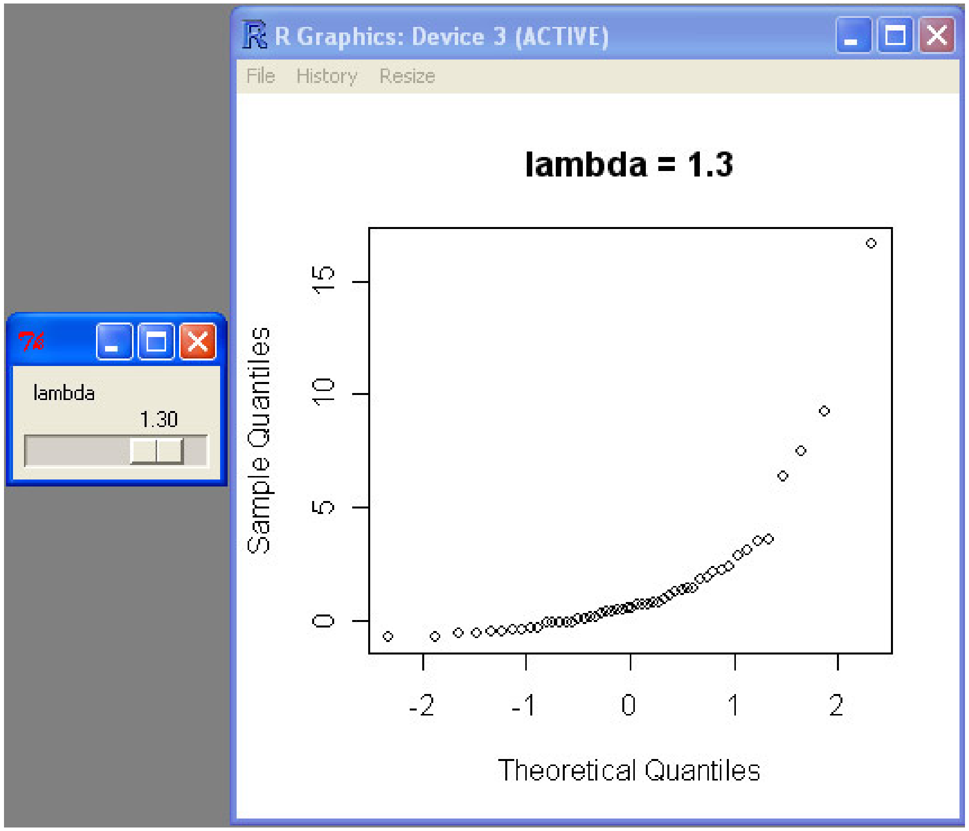
examples

`- preview of examples - click to run - check what runs`



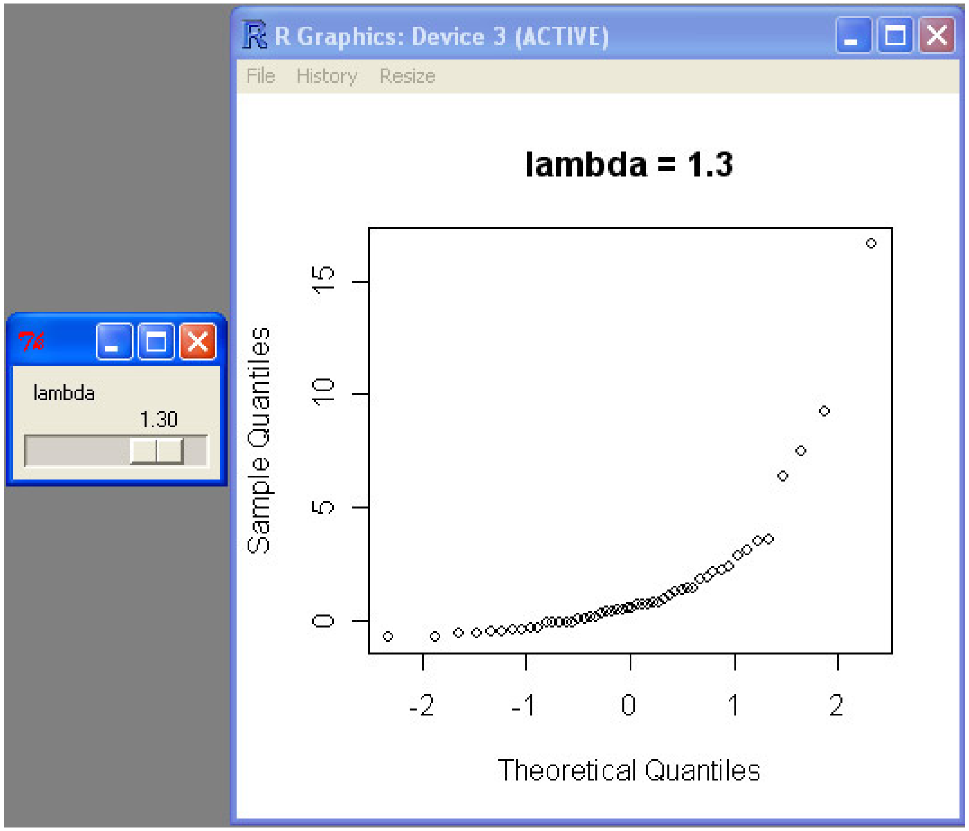
examples

- regression 1d



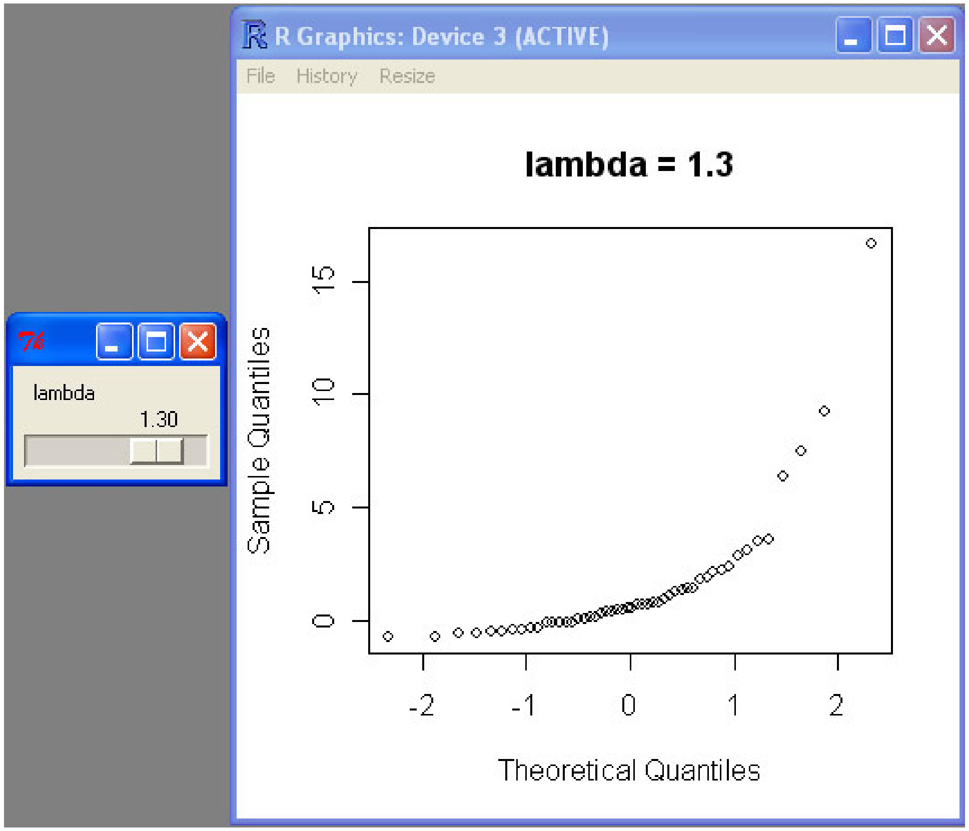
examples

- cosine regression



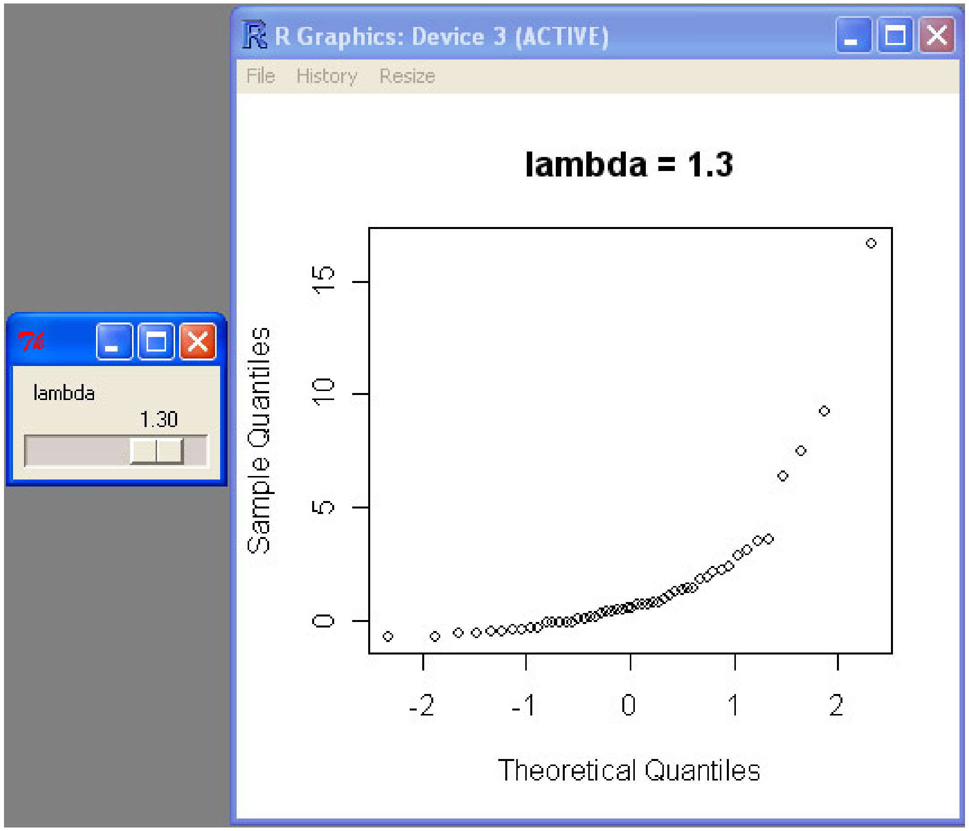
examples

- density est 1d



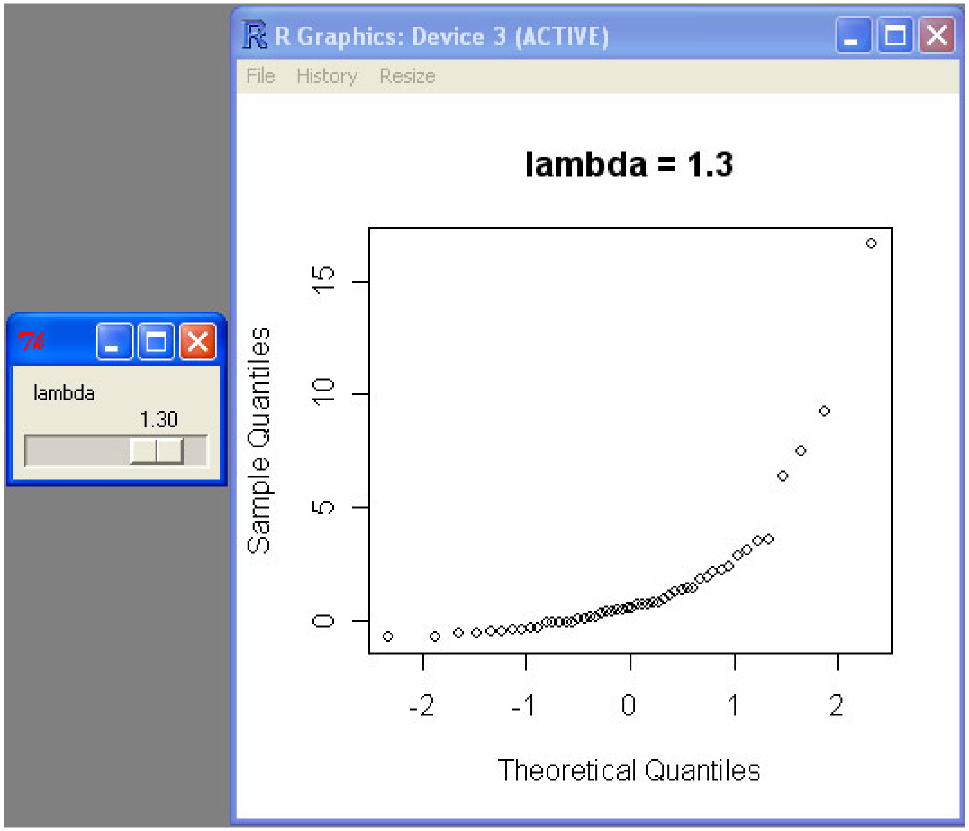
examples

- gulls



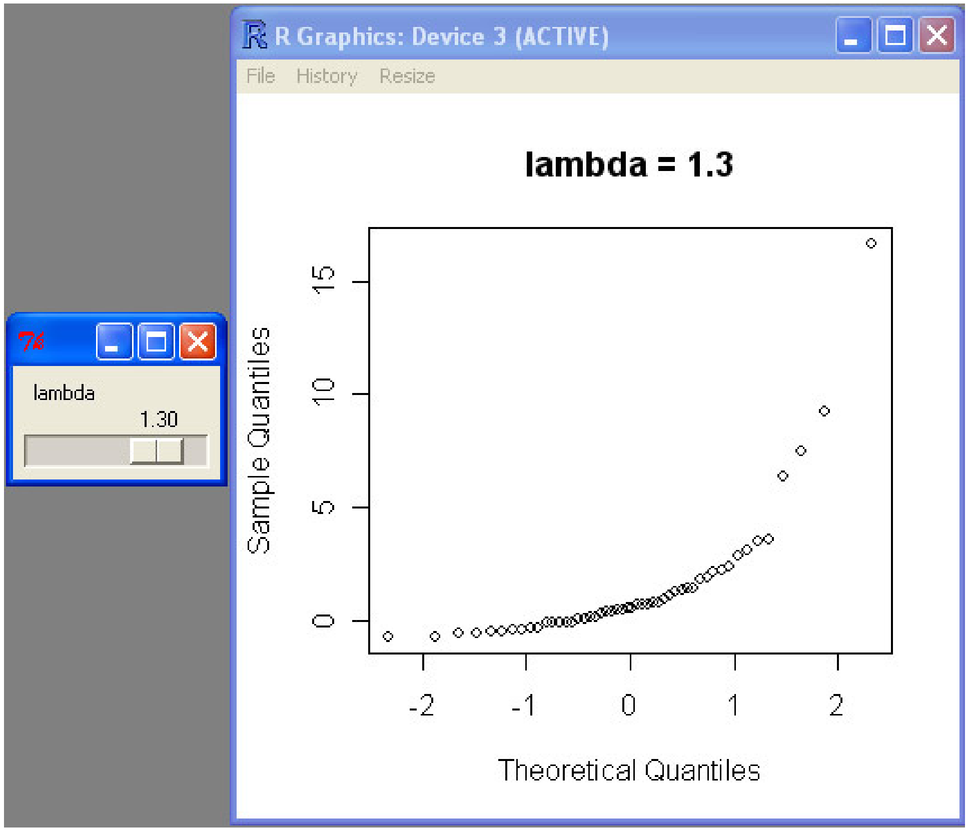
examples

- `clyde map`



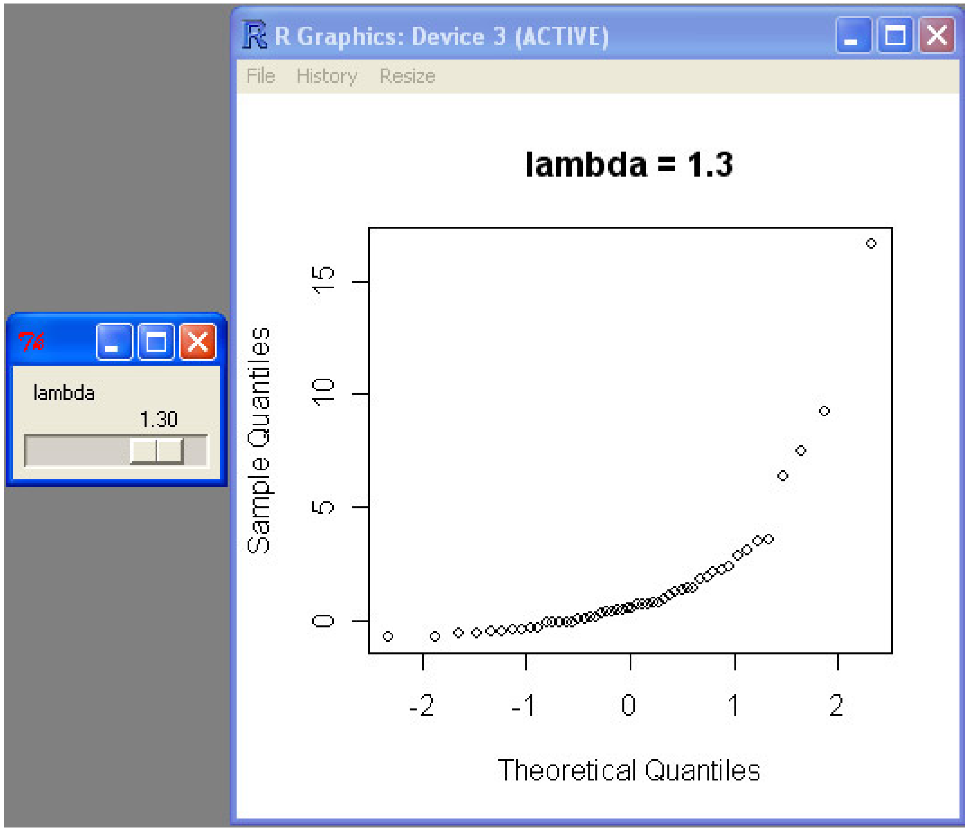
examples

- `spatial sim`



examples

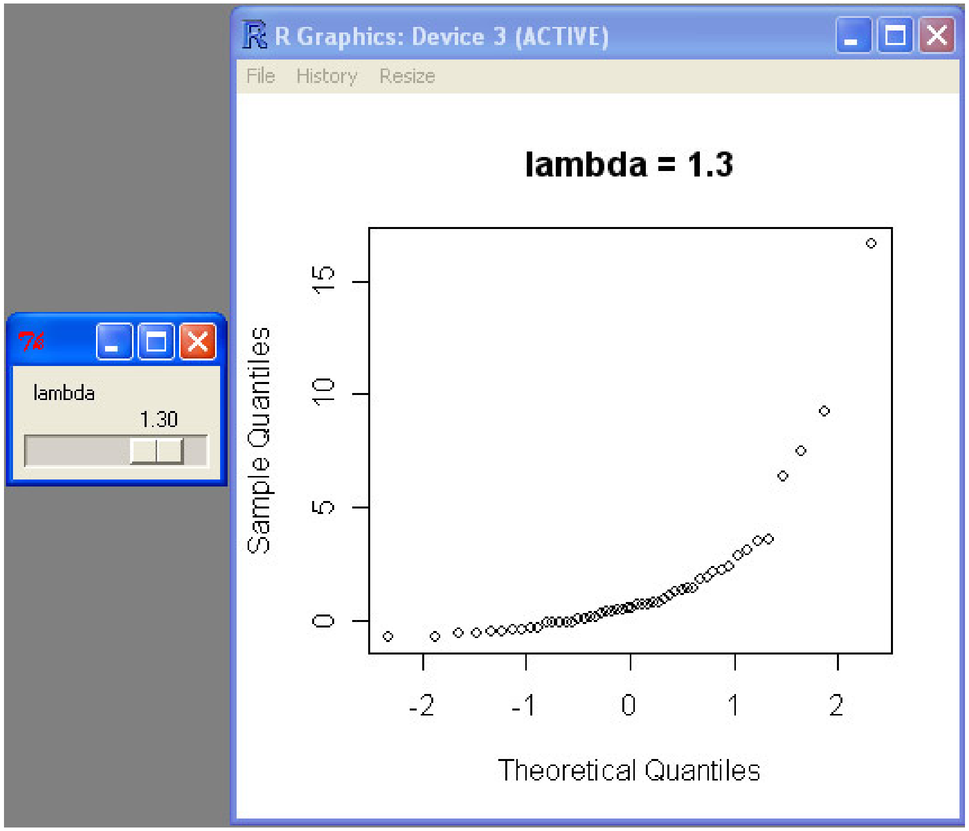
- mururoa



lambda 1.30

examples

- rosyth



code

```
*** As per Adrian's build up? ***

panel <- rp.control("Clyde data", do = DO, days = Days)
panel <- rp.slider(panel, day.adj, 0, 364,
action = days.plot)
panel <- rp.checkbox(panel, model.showing, model.fn,
title = "Show model")
Behind the scenes . . .
panel <- action(panel)
```

reaction

- discuss with AB
- applications

discussion

- my comments, pros, cons
- future plans
  - pos, grid, notebook, combo ...

Web site

www.stats.gla.ac.uk/~adrian/rpanel