Interactive Graphics and Data Visualization using `getGraphicsEvent()`

Nels Grevstad

Metropolitan State College of Denver

Interactive Graphics

- Graphics device (`windows()`, `X11`) accepts input from mouse or keyboard.
- Some possible uses:
  - Quickly (e.g. with a mouse click) create a subset of a data frame and plot it.
  - Simultaneously change plots in two (or more) graphics windows.
  - Rapidly (e.g. with a key press) scroll through many *conditioning plots* (or *facets*).

Terminology

- ***Event***: A mouse click or key press.
- ***Event handler***: A user defined function that serves as a response to an *event*.

getGraphicsEvent() Usage

First, define the *event handlers*:

```
my.mouseclickhandler <- function(buttons, x, y) {
    my commands here
}
my.mousemovehandler <- function(buttons, x, y) {
    my commands here
}
my.mousereleasehandler <- function(buttons, x, y) {
    my commands here
}
my.keyhandler <- function(key) {
    my commands here
}
```

getGraphicsEvent() Usage (Cont'd)

Next, set the *event handlers*:

```
setGraphicsEventHandlers(
   onMouseDown = my.mouseclickhandler,
   onMouseMove = my.mousemovehandler,
   onMouseUp = my.mousereleasehandler,
   onKeybd = my.keyhandler)
```

This:

1. Creates an environment (the *event environment*).
2. Copies the *event handlers* `my.mouseclickhandler()`, `my.keyhandler()`, and `my.mousemovehandler()` into objects called `onMouseDown()`, `onMouseMove()`, `onMouseUp()`, and `onKeybd()` in the *event environment*.

`getGraphicsEvent()` Usage (Cont’d)

Finally, call `getGraphicsEvent()` (with no arguments):

`getGraphicsEvent()`

This:

1. Waits for an *event*.
2. Searches the *event environment* for the appropriate *event handler*.
3. Executes the *event handler*, passing it the arguments:
   - `buttons` (0 for left click, 1 for center, 2 for right).
   - `x`, `y` (mouse location in "ndc" coordinates).
   - or `key` ("a", "b", "c", etc. depending on which key was pressed).

A Simple Example

```
my.mouseclickhandler <- function(buttons, x, y) {
  if (buttons == 2) {
    clicked.x <- grconvertX(x, from = "ndc",
        to = "user")
    clicked.y <- grconvertY(y, from = "ndc",
        to = "user")
    clicked.pt <- c(clicked.x, clicked.y)
    distances <-
        dist(rbind(clicked.pt, my.df))[1:my.n]
    nearest.pt.index <- which.min(distances)
    nearest.pt <- my.df[nearest.pt.index, ]
    dev.set(2)
    points(nearest.pt, pch=19)
    dev.set(dev.next())
    points(nearest.pt, pch=19)  # Returns NULL
  }
}
```

A Simple Example (Cont’d)

```
my.keyhandler <- function(key) {
    if (key == "q") return(invisible(1))
}

setGraphicsEventHandlers(
  onMouseDown = my.mouseclickhandler,
  onKeybd = my.keyhandler,
  which = dev.list()[1])

getGraphicsEvent()
```

Mosquito Data

- Weekly presence/absence observations for each of 9 mosquito species at 84 sites in Morgan County, CO over 14 weeks from May to August, 2010.
- Focus here is on two species, *Culex tarsalis* and *Aedes (Oc.) dorsalis*.

```
> head(morgan)
  Site.ID      Lat      Long wk n.species culex.t.present aedes.d.present
1  FM0001 40.23778 -103.7797 31         0               0               0
2  FM0001 40.23778 -103.7797 33         1               0               0
3  FM0001 40.23778 -103.7797 28         1               1               0
4  FM0001 40.23778 -103.7797 26         3               1               0
5  FM0001 40.23778 -103.7797 30         0               0               0
```

Some Useful Functions

```
grconvertX(); grconvertY()  # Convert between coord-
                            # inate systems

dev.new()                   # Open and activate a
                            # new graphics device

dev.set(); dev.next();      # Navigate multiple
dev.prev(); dev.cur();      # graphics devices
dev.list()

<<-                         # Assignment in the Glo-
                            # bal Environment (easier
                            # than using assign())
```

Some Useful Functions (Cont'd)

```
dist(); which.min()          # Compute pairwise dist-
                             # ances between points;
                             # find the index of the
                             # minimum.

layout(); split.screen()     # Put multiple plots in
                             # one graphics window
                             # (like setting
                             # par("mfrow"))

graphics.off()               # Close all graphics
                             # devices
```

Other Interactive Graphics Packages

- rgl
- RTclTk
- rggobi
- Some others ...

Thanks to ...

- Denver R Users Group organizers
- Bob Hancock and collaborators, MSCD Dept of Biology, for mosquito data
- Duncan Murdock, R Core Team, for creating `getGraphicsEvent()`