cranvas: Interactive Statistical Graphics in R via Qt

Yihui Xie

Department of Statistics, Iowa State University

September 23, 2011

Introduction to cranvas
• the name (CRAN + canvas ⇒ cranvas)

• demo
  – old problems (with new look) in cranvas
  – NRC rankings of statistics departments
Looking back

- standalone software packages: GGobi [Swayne et al., 2003], MANET [Unwin et al., 1996] and Mondrian [Theus, 2002], etc
• R itself has poor support for interactivity (ancient `locate()`? `getGraphicsEvent()`? no kidding! there is a **fun** package [Xie et al., 2011], and also **sudoku**)

• R packages: **iplots** [Urbanek and Wichtrey, 2011], **rggobi**, **playwith**, **rgl** (as I showed just now)...
Overview of cranvas

• next generation of GGobi (and more)
  – interactive statistical graphics in R
  – focus on speed, flexibility and elegance

• infrastructures (details later):
  – painting (drawing): **qtbase** & **qtpaint** (connect R with Qt)
  – data structure: **plumbr** & **objectSignals** (mutaframes, reference classes, signals)
– aesthetics: **scales** (and other **ggplot2**’s offspring)

• all available at [http://github.com/ggobi](http://github.com/ggobi) (to be released on Bioconductor & CRAN)

– refer to the wiki for installation under Linux & Mac
– Windows version not available at the moment

• usage

  – create a data object: `mf <- qdata(df)`
  – make a series of plots based on it: `qhist(x1, data = mf); qscatter(x2, x3, data = mf)`

• cranvas is still under active development
– milestone 1.0: get basic plots working (almost done)
– milestone 2.0 scheduled in November (after R 2.14.0): new types of plots, speed, documentation
Data structure: augmented data

> library(cranvas)
> data(flea, package = "tourr")
> head(flea)  # first 6 rows of flea

<table>
<thead>
<tr>
<th></th>
<th>tars1</th>
<th>tars2</th>
<th>head</th>
<th>aede1</th>
<th>aede2</th>
<th>aede3</th>
<th>species</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>191</td>
<td>131</td>
<td>53</td>
<td>150</td>
<td>15</td>
<td>104</td>
<td>Concinna</td>
</tr>
<tr>
<td>2</td>
<td>185</td>
<td>134</td>
<td>50</td>
<td>147</td>
<td>13</td>
<td>105</td>
<td>Concinna</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
<td>137</td>
<td>52</td>
<td>144</td>
<td>14</td>
<td>102</td>
<td>Concinna</td>
</tr>
<tr>
<td>4</td>
<td>173</td>
<td>127</td>
<td>50</td>
<td>144</td>
<td>16</td>
<td>97</td>
<td>Concinna</td>
</tr>
</tbody>
</table>
> str(flea)

data.frame: 74 obs. of 7 variables:
  $ tars1 : int 191 185 200 173 171 160 188 186 174 163 ...
  $ tars2 : int 131 134 137 127 118 118 134 129 131 115 ...
  $ head : int 53 50 52 50 49 47 54 51 52 47 ...
  $ aede1 : int 150 147 144 144 153 140 151 143 144 142 ...
  $ aede2 : int 15 13 14 16 13 15 14 14 14 15 ...
  $ aede3 : int 104 105 102 97 106 99 98 110 116 95 ...
  $ species: Factor w/ 3 levels "Concinna ","Heikert. ",...
```r
> qflea <- qdata(flea, color = species)
> head(qflea)  # what is the difference?
```

<table>
<thead>
<tr>
<th></th>
<th>tars1</th>
<th>tars2</th>
<th>head</th>
<th>aede1</th>
<th>aede2</th>
<th>aede3</th>
<th>species</th>
<th>.brushed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>191</td>
<td>131</td>
<td>53</td>
<td>150</td>
<td>15</td>
<td>104</td>
<td>Concinna</td>
<td>FALSE</td>
</tr>
<tr>
<td>2</td>
<td>185</td>
<td>134</td>
<td>50</td>
<td>147</td>
<td>13</td>
<td>105</td>
<td>Concinna</td>
<td>FALSE</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
<td>137</td>
<td>52</td>
<td>144</td>
<td>14</td>
<td>102</td>
<td>Concinna</td>
<td>FALSE</td>
</tr>
<tr>
<td>4</td>
<td>173</td>
<td>127</td>
<td>50</td>
<td>144</td>
<td>16</td>
<td>97</td>
<td>Concinna</td>
<td>FALSE</td>
</tr>
<tr>
<td>5</td>
<td>171</td>
<td>118</td>
<td>49</td>
<td>153</td>
<td>13</td>
<td>106</td>
<td>Concinna</td>
<td>FALSE</td>
</tr>
<tr>
<td>6</td>
<td>160</td>
<td>118</td>
<td>47</td>
<td>140</td>
<td>15</td>
<td>99</td>
<td>Concinna</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>.visible</th>
<th>.color</th>
<th>.border</th>
<th>.size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TRUE</td>
<td>#F8766D</td>
<td>#F8766D</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>TRUE</td>
<td>#F8766D</td>
<td>#F8766D</td>
<td>4</td>
</tr>
</tbody>
</table>
• the augmented data contains information about each observation (row)

  - `.brushed`: logical vector – whether a row is brushed or not
  - `.visible`: similar – visible or not?
  - `.color`, `.border`, `.size` – aesthetics of graphical elements
Data structure: mutaframe

• what we created by `qdata()` was a mutaframe (using `plumbr`)

• a mutaframe is like a data frame, but is mutable (technically, it is an environment)

```r
> str(qflea, max.level = 1)

Classes 'mutaframe', 'environment' <environment: 0x589cb20>
  - attr(*, "col.names")= chr [1:12] "tars1" "tars2" "head...
  - attr(*, "row.names")= chr [1:74] "1" "2" "3" "4" ...
```
- attr(*, "changed")=Formal class 'Signal' [package "objSignals"]
- attr(*, "Brush")=Formal class 'BRUSH' [package "cranvas"]
- attr(*, "Link")=Classes 'mutalist', 'environment' <environment>
  ..- attr(*, "envNames")= chr "linkid"
  ..- attr(*, "userNames")= chr "linkid"
- attr(*, "Scales")=Formal class 'Scales_meta' [package "cranvas"]
- attr(*, "Generator")= chr "d3b8be46da5faa45758f3609f5d9c98"

- modification to a mutaframe is always “global”, which enables us to pass and modify a mutaframe anywhere in functions

  - application to tour (projections always changing, and the plot needs to know this)
> a <- 10  # ordinary R object
> b <- a
> b <- 5
> a  # certainly a is not modified

[1] 10

> ## but mutaframes are different
> x <- qflea  # assign qflea to x
> qflea$tars1[1]  # first element of tars1

[1] 191

> x$tars1[1]  # same element in x
> x$tars1[1] <- 300  # modify it
> x$tars1[1]

[1] 300

> qflea$tars1[1]  # original copy also modified!

[1] 300
Data structure: listeners

• how does interaction work?
  – interaction does not modify the plot directly
  – instead, it modifies data and triggers listeners on data, which do the job of updating the plot
  – it makes developers’ life a lot easier

• what is a listener?
  – it is a function attached to a mutaframe
– (roughly speaking) whenever the mutaframe is modified, the function is executed

• which part of the mutaframe is modified?

– we should not abuse the listeners, hence comes the concept of the “data pipeline”
  – .brushed changed? update the brush layer!
  – .visible changed? update the main plot layer!
  – original data changed? go back to the beginning and re-calculate everything, then update all the layers!

• each time we create a plot, a listener is attached to the mutaframe
— so the `.brushed` column can update several plots at the same time
Data structure: a summary

• a data object is sitting behind the scene
• interactions change the data
• listeners on the data update the plot(s)
Interaction: brushing or selection

• mouse interactions
  – left click and move the brush; right click to resize the brush; middle button to toggle between brush and selection
  – brush mode and identify mode (use ? to toggle)

• selection mode
  – AND, OR, XOR, NOT, COMPLEMENT (use the initial letters)

• related keyboard interactions
– Delete makes elements invisible; F5: all visible
Interaction: linking

- one-to-one linking
  - brush one observation in this plot, highlight the same observation in other plots

- categorical linking
  - ..., all observations in the same category are brushed too

- kNN linking
- ..., k nearest neighbors are brushed
Interaction: misc

• + and – changes the alpha transparency

• PageDown and PageUp navigates through brush history

• ...
Examples: plots in cranvas now, and what’s new

• histogram (and spine plot): change binwidth with ↑ and ↓; shift bins with ← and →

• density plot: similar to histogram

• bar plot: application to missing value plot

• scatter plot: change size of points with ↑ and ↓
• boxplot: also show small boxplots for brushed observations inside original boxplots (not actually new)

• mosaic plot: dynamically change variables in the plot and layout

• parallel coordinates plot: rearrange order of variables (not new); show data ranges

• map: also cartograms

• time series plot: many, many new features (as a result of GSoC project)
Examples: flea and states demographics

• flea beetles
  – tour
  – categorical linking by species

• US states demographics
  – selection mode
  – kNN linking
Limitations

- suffers from Qt flaws and bugs
  - can be slow (mainly in scatter plots) due to changes in BSP tree algorithm and cache failure; I was told it WAS extremely fast
  - mysterious clipping (aggressive: points clipped into halves; imprecise: no clipping even when points reach beyond boundary)

- Qt is big (in size) and installation of `qtbase` / `qtpaint` under Windows may be difficult
Future plans

• automatic and *interactive* legend

• conditioning (faceting) like Trellis

• more types of plots (e.g. hexagons)

• a lot to learn from `ggplot2` and `iplots` (ix?)
Acknowledgements

- Heike Hofmann, Di Cook, Xiaoyue Cheng, Tengfei Yin (current developers)

- Barret Schloerke, Marie Vendettuoli (former developers)

- Hadley Wickham, Michael Lawrence (all the hard work on infrastructure)

- Deborah Swayne
Looking at the acknowledgement list

<table>
<thead>
<tr>
<th>cranvas developers</th>
<th>iplots developer</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Cat icons" /></td>
<td><img src="image2" alt="Cat icon" /></td>
</tr>
<tr>
<td><img src="image3" alt="Cat icons" /></td>
<td><img src="image4" alt="Cat icon" /></td>
</tr>
<tr>
<td><img src="image5" alt="Cat icons" /></td>
<td><img src="image6" alt="Cat icon" /></td>
</tr>
</tbody>
</table>
Questions & Comments?

- Testing and bug reports ([https://github.com/ggobi/cranvas/issues](https://github.com/ggobi/cranvas/issues)) are welcome

- Thanks!

References

[Swayne et al., 2003] Swayne, D. F., Temple Lang, D., Buja, A.,


