Data visualization, focusing on {ggplot} and mixed models

Goals/contexts of data visualization

Exploration

- want nonparametric/robust approaches: impose as few assumptions as possible
- boxplots instead of mean/standard deviation (generally base locations on medians rather than means)
- loess/GAM instead of linear/polynomial regression
- need speed: quick and dirty
- canned routines for standard tasks, flexibility for non-standard tasks
- manipulation in the context of visualization: need to summarize on the fly

Diagnostics

- attempt to determine fitting problems graphically: looking for absence of patterns in residuals
- e.g. scale-location plot, Q-Q plot; \texttt{plot.lm}
- plot methods: generic (e.g. residuals vs fitted) vs specific (e.g. residuals vs predictors)
- plotting predictions (intuitive) vs plotting residuals (amplifies/zooms in on discrepancies)
- plotting unmodeled characteristics (e.g. spatial, temporal autocorrelation): much easier to draw a picture than fit a model
• code contrasts for visual simplicity (e.g. deviations from linearity: Q-Q plots, signed square-root profiles)

Compare (negative) log-likelihood profiles ... 

... with signed square-root LL profiles ...

Presentation

• how closely should one match analyses with graphs? “Let the data speak for themselves” vs “Tell a story”
• display data (e.g. boxplots, standard deviations) or inferences from data (confidence intervals)
• superimposing model fits (geom_smooth)
• avoid excessive cleverness/data density
• coefficient plots vs parameter tables (Gelman, Pasarica, and Dodhia 2002)
• tradeoff between visual design (tweaking) and reproducibility: learning to futz with label positioning etc. may pay off in the long run (a few tools exist for automatic placement)
• order factors in a sensible order (i.e not alphabetical or numerical unless (1) the labels have some intrinsic meaning or (2) you expect that readers will be interested in looking up particular levels in the plot). This is sometimes
called the “what’s so special about Alabama?” problem, although the Canadian version would substitute “Alberta”.

Basic criteria for data presentation

Visual perception of quantitative information: Cleveland hierarchy (W. S. Cleveland and McGill 1984)

### Cleveland’s (1984) Graphical Feature Interpretation Hierarchy

<table>
<thead>
<tr>
<th>Best</th>
<th>Position along a common scale</th>
<th></th>
<th>Position along nonaligned scales</th>
<th></th>
<th>Length</th>
<th>Angle/ slope</th>
<th>Area</th>
<th>Volume</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Worst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Based on graphic (Figure 2) in Presentation Graphics (white paper) by Leland Wilkinson, SPSS, Inc and Northwestern Univ.

Data presentation scales with data size

- **small** show all points, possibly dodged/jittered, with some summary statistics: dotplot, beeswarm. Simple trends (linear/GLM)
- **medium** boxplots, loess, histograms, GAM (or linear regression)
- **large** modern nonparametrics: violin plots, hexbin plots, kernel densities: computational burden, and display overlapping problems, relevant
- combinations or overlays where appropriate (beanplot)

Rules of thumb

- (Continuous) response on the \(y\)-axis, most salient (continuous) predictor on the \(x\)-axis
- Put most salient comparisons within the same subplot (distinguished by color/shape), and nearby within the subplot when grouping bars/points
• Facet rows > facet columns
• Use transparency to include important but potentially distracting detail
• Do category levels need to be identified or just distinguished?
• Order categorical variables meaningfully
• Display population variation (standard deviations, boxplots) vs. estimate variation (standard errors, mean ± 2 SE, boxplot notches)
• Try to match graphics to statistical analysis, but not at all costs
• Choose colors carefully (\texttt{RColorBrewer}/\texttt{ColorBrewer}, \texttt{IWantHue}: respect dichromats and B&W printouts

Challenges

high-dimensional data (esp continuous)
Possible solutions:

• use color, shape for discrete predictor variables (up to ~10 categories); text plots
• small multiples, conditioning plots (shingles/facets); i.e. discretize continuous plots
• contour plots (worse)
• perspective plots (worst?)

large data sets
• problems with computation, file size, presentation
• file size: raster (PNG) instead of vector (PS/PDF), \texttt{pch="."}
• overplotting (alpha), kernel density estimation, hexagonal binning
• summarize (quantiles, kernel densities, etc.)

discrete data
• lots of point overlap; jittering OK for exploratory analysis but ugly. Need to summarize/bin appropriately (\texttt{stat_sum}); beeswarm plots

Spatial data
• the best parts of the Cleveland hierarchy ($x$ and $y$ axes) are already taken, usually have to resort to color/size/pie charts. Representing uncertainty is a big challenge, usually must be done separately (transparency/saturation?)
compositional data

- would like to display “sum to 1.0” constraint but also allow accurate comparison of magnitudes: stacked bars vs grouped bars (or dotplots)?
- harder if also need to represent uncertainty (would like to show correlations among components)
- *ternary diagrams*: nice but don’t generalize past 3 elements

Multilevel data

- often hard (or messy) to represent all levels of variation

next generation tools

- dynamic/exploratory graphics: *ggobi*, Mondrian, [latticist](http://code.google.com/p/latticist), JMP, Shiny, Gapminder, googleVis, rCharts
- GUI frameworks: JMP, R Commander, Deducer, Rattle, web interface to *ggplot2*
- presentation technologies: JCGS editorial with supplementary materials

Graphics culture

- the gods: Cleveland (hierarchy), Tufte (2001,”) (“chartjunk”, minimizing non-data-ink, sparklines)
- dionysians (infovis) vs. apollonians (statistical graphics) (Gelman and Unwin 2013)
- graphics nazis: dynamite plots, pie charts (esp. 3D), dual-axis figures …

Data visualization in R

Base graphics

- simple ‘canvas’ approach
- straightforward, easy to customize
- most plot methods written in base graphics

Lattice

- newer
- documented in a book (Sarkar 2008)
- based on grid graphics package
• faceting, conditioning plots
• much more automatic, better graphical defaults
• implements banking, other aspect ratio control
• more ‘magic’, harder to customize
• some plot methods (nlme package)
• latticeExtra, directlabels packages may be handy

**ggplot**

• newest
• based on Wilkinson’s “Grammar of Graphics”
• documented in a book (see below) and on a web site, as well as an active mailing list
• explicit mapping from variables to “aesthetics”: x, y, colour, size, shape
• implements faceting (not quite as flexibly as lattice: no aspect ratio control)
• some data summaries etc. built in
• easier to overlay multiple data sets, data summaries, model predictions etc.
• no 3D plots
• rendering can be slow
• gridExtra, ggExtra, directlabels package may be handy

**ggplot intro**

mappings + geoms

**Data**

Specified explicitly as part of a ggplot call:

```r
library(mlmRev)
head(Oxboys)
```

```
## Subject age height Occasion
## 1 1 -1.0000 140.5 1
## 2 1 -0.7479 143.4 2
## 3 1 -0.4630 144.8 3
## 4 1 -0.1643 147.1 4
## 5 1 -0.0027 147.7 5
## 6 1 0.2466 150.2 6
```

```r
library(ggplot2)
ggplot(Oxboys)
```
## Error: No layers in plot

But that isn’t quite enough: we need to specify a mapping between variables (columns in the data set) and aesthetics (elements of the graphical display: x-location, y-location, colour, size, shape ...)

```r
ggplot(Oxboys, aes(x = age, y = height))
```

## Error: No layers in plot

but (as you can see) that’s still not quite enough. We need to specify some geometric objects (called geoms) such as points, lines, etc., that will embody these aesthetics. The weirdest thing about `ggplot` syntax is that these geoms get added to the existing `ggplot` object that specifies the data and aesthetics; unless you explicitly specify other aesthetics, they are inherited from the initial `ggplot` call.

```r
ggplot(Oxboys, aes(x = age, y = height)) + geom_point()
```
• many more geoms (lines, bars, etc.)
• summarizers: smooth lines and summaries (geom_smooth, stat_sum)
• control of scales (e.g. log transforms, colors, etc.)
• faceting (grid and wrap)

See Karthik’s ggplot intro or my intro for disease ecologists, among many others.

References


