Mondrian

or

this is not a Toolkit

Martin Theus

martin@theusRus.de
Differences at Several Levels

• **Motivation**
  The motivation for writing Mondrian was (in the end) to build a tool that can be used by anyone who needs graphical methods for EDA for (almost) arbitrary datasets, regardless of his/her computing skills.

• **Concept**
  Above motivation calls for a more or less closed and complete application with no configuration efforts and little learning efforts.

• **Technical**
  The software design for a closed application does not necessarily need “orthogonal” components that can be combined to build new visualizations.
All Kinds of Data

• **Structure Data vs. Unstructured Data**
  Classical datasets in statistics are simple rectangular data matrices with rows corresponding to observations (cases) and columns are the different variables (attributes) measured per observation.

• **Data on different Scales**
  Above all, the scale of a variable is important for its potential role in an analysis. Scales are:
  – nominal (alpha-numeric or numeric)
  – ordinal (alpha-numeric or numeric)
  – continuous

• **Dark Ages of Statistical Data Visualization**
  For a long time, data visualization in statistics did only handle numerical data (as classical statistics does) with all the problems.
Data Handling in Mondrian

- Mondrian assumes that data sits either in datafiles or JDBC accessible databases and follows the strict rectangular layout. (datafiles may optionally point to a polygon description file)
- The dataset class handles all data requests (selection, color, ...)
- Internally all columns of the data table are stored as variables
- The table class manages all (multivariate) categorical data
Graph Drawing Objects

- Of central importance in Mondrian are the mechanisms for selecting and highlighting data on case level.
- The standard plot-canvas supports all that is needed for selections.
- If coordinate systems are used, standard zooming can be used.
- Each plot must implement the necessary methods to maintain the correct representation of a selection, color etc.
Plot Primitives

• **Points**
  Points, like in scatterplots, are **NO** objects and have a 1:1 correspondence to some columns of a single row in the data matrix.

• **Polylines**
  Analogous to points, polylines are the multivariate incarnation of a point, i.e., they correspond 1:1 to a row in the data matrix.

• **Rectangles**
  Rectangles are objects that correspond to either a single row of a table or a group of rows of a table and gather many cases.

• **Polygones**
  Polygones, as in maps, are a generalization of rectangles and link to a group of cases in the dataset, less strict as a table.
Decomposing a Graphic

• In an object-oriented programming environment/language, an effective definition of the graphical objects is key.

• Typical Objects
  – plot primitives
    ■ points
    ■ lines
    ■ boxes
  – axes
  – plot specifics

• Example: Histogram
  – primitives: boxes
  – axes
    ■ x: range
    ■ y: count or probability
  – plot specifics
    ■ origin and width control
Example of Plot Layout

- 4 layers can be defined to group the different plot components
  - Interaction layer
  - Selection layer
  - Object layer
  - Background layer

- The layers are defined according to their update frequencies from least frequent update to most frequent update, i.e.
  interaction ➤ selection/highlighting ➤ objects ➤ axes (background)

- Very important to speed up drawing times!
Interactions: Events

- Apart from JAVA’s standard events, Mondrian implements two application specific events:
  - Selection Event
  - DataChanged Event

- Event distribution (e.g. plot 2 changes the selection state)

```
Plot 1
Plot 2
Plot 3
Plot n
Parent: manages plot list
Event listener / distributor
```
Interactions: Conventions

• There is a tight and consistent mapping of interactions
  – Selections
    ■ click and drag ➢ create a selection rectangle / brush
    ■ click on selection rectangle handle ➢ resize this selection
    ■ popup-trigger on selection rectangles ➢ alter this selection
  – Queries
    ■ <alt>-mouse over coordinate system ➢ orientation query
    ■ <control>-mouse over objects ➢ query
    ■ <shift>-<control>-mouse over objects ➢ extended query
  – Alterations
    ■ meta-click and drag ➢ zoom in/out (middle click on Windows)
    ■ popup-trigger on background ➢ get/change plot options
    ■ alt-click and drag ➢ reorder objects
    ■ page-up/-down ➢ cycle through views
    ■ arrows up/down and left/right ➢ increase/decrease plot parameters
Animation free Zone

• In InfoVis, animation is almost a must; in statistics, animation will significantly reduce your credibility.

• Animations usually show a transition from one state to another
  – different layouts (mainly for graphs)
  – different scales (zoom operations; maps etc.)
  – different plot parameters (e.g., smoothing parameters)

• Animations help to preserve the context, which might be lost if the change happen too abruptly.

• Transitions should be avoided if the intermediate states are not meaningful.

• The only obligatory animation in statistical graphics can be found in 3-d rotating plots
What does it take to build a new Plot?

• Data handling: ✓

• Define new plot object
  – Derive new class from MPanel
  – [Aggregate data, and/or calculate statistics]
  – Define the paint() method using
    ▪ coordinate system
    ▪ plot primitives
  – Define selection methods
  – [Define custom interactions]

• Housekeeping
  – Add plot to the plot menu
  – Define variable constraints for the plot

• All coding has to be done in JAVA
Size Matters!

• Unlike classical statistical graphics tools, Mondrian takes care of large datasets, i.e., dataset with > 1,000,000 observations

• There are some standard techniques to cope with massive data
  – alpha-blending to cure overplotting
  – different forms of zooming (names may vary)
    ▪ standard
    ▪ logical (change representation of objects)
    ▪ censored zooming (only focus on the fringes)
    ▪ quantum zooming (only zoom in on the highlights)
    ▪ ...
  – automatic sorting options
  – automatic permutations

• Above all, make sure the plot is still working with large amounts of data; regarding rendering speed AND interpretability.
Summary

- The main difference between Mondrian and (other) InfoVis toolkits is probably the difference between building a visualization tool and implementing domain specific concepts and strategies.

- Structured data (as in graphs) directly constitutes the features within a dataset. If we assume to have randomness following a specific distribution, we might observe the features in the data only indirectly.

- Having “only” multivariate data of just a few structural different types of distributions, there is no need to create new graphical representations “by the minute”.

- Nevertheless, to create prototypes of a new statistical graph, it probably needs more flexibility than a “standard” toolkit offers.