Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projekt 1 Ausarbeitung

Truong Vinh Phan

Support for Interactive Visual Analytics in Various Software Packages

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Contents

1 Introduction 1

1.1 Motivation . . . 1

1.2 Used Hard- & Software . . . 1

2 Exploratory Data Visualization with Processing 1

2.1 Introduction and a Brief History . . . 1

2.2 Design and Features . . . 1

2.3 Example Project: Interactive Visualization of Google Trends Data for Searches of Various Opera Composers. . . . 2

2.3.1 Objectives & Requirements . . . 2

2.3.2 Solution . . . 2

2.4 Conclusion . . . 3

3 Interactive Visual Analytics with Mondrian 3

3.1 Introduction . . . 3

3.2 Support for Exploratory Data Analysis through Interactive Visualization . . . 4

3.2.1 Advanced Data Selection . . . 4

3.2.2 Use of Conventions . . . 4

3.2.3 Support for High Dimensional and Categorical Data . . . 4

3.3 Enhancements in Standard Plots . . . 7

3.3.1 Barcharts . . . 7

3.3.2 Histograms . . . 8

3.3.3 Scatterplots . . . 9

3.4 Conclusion . . . 10

References 11

1 Introduction

1.1 Motivation

Interactive techniques, such as linked data views and data brushing, combined with visualization methodologies like the Tree-maps, are at the core of any exploratory data analysis. In this report, two prominent visualization packages, Processing and Mondrian, are introduced and examined, to look at the available features, as well as how well they handle large data sets, which is vital for doing big data analysis through exploratory visualization. An example visualization project will also be implemented using the Processing development platform.

1.2 Used Hard- & Software

Hardware in use is standard desktop computer with Intel Core® i7 processor and 3 Gb memory. Used software packages include the Processing IDE version 2.2.1 (19/05/2014) [1] and Mondrian version 1.2 (01/11/2011) [4]

2 Exploratory Data Visualization with Processing

2.1 Introduction and a Brief History

Processing is an open-source programming language, development environment and online community. It was started as a project by Ben Fry and Casey Reas in 2001 at MIT Media Lab within John Maeda's Aesthetics and Computation research group and inspired by earlier languages like BASIC and Logo. Many ideas in Processing go back to Muriel Cooper's Visual Language Workshop and grew directly out of Maeda's Design by Numbers project in 1999. A JavaScript version of Processing (Processing.js) has been developed by John Resig (jQuery), as well as being in development are other versions of Processing that use Python, Ruby, Scala, etc.

2.2 Design and Features

Processing was built with focus on the electronic arts, new media art and visual design communities, with the purpose of teaching the fundamentals of computer programming in a visual context and serve as the foundation for eletronic sketchbooks. Processing was based on the Java language, but has simplified syntax and graphics programming model.

The latest stable release is Processing 2.2.1 in May, 2014. The version 2.0 release focuses on faster graphics, new infrastructure for working with data, and enhanced video playback and capture. The new Modes feature allows other programming platforms like JavaScript and Android, to be used from within the IDE with ease. The P2D and P3D renderers are now built with modern OpenGL and can utilize custom GLSL shaders. Processing has a large collection of libraries, contributed by the community, to further extend the functionality of the language, such as to facilitate computer vision, data visualization, music composition, etc.

2.3 Example Project: Interactive Visualization of Google Trends Data for Searches of Various Opera Composers.

2.3.1 Objectives & Requirements

The main objective of this example project is to create a visualization that would allow the users to interact with data and explore trends in Google searches for various opera composers over the last several years.

Dataset is taken from Google Trends and stored in a file “composers.csv” that has been cleaned to have just the relative popularity of 5 searches from January 2005 to December 2012. Keywords used are: “puccini opera”, “verdi opera”, “mozart opera”, “wagner opera” and “bizet opera”.

- Visualization should be an interactive bar chart with one bar for each composer and a scale of 0-100
- Visualization should have a slider that allows the user to choose the time period to be displayed by the bars. It should also have date labels for the ends of the slider’s range and current date shown on the slider button.
- Visualization should include images of the five composers and be exported as a standalone desktop application for Windows and Mac.

2.3.2 Solution

As we move the slider across to explore the data interactively, the barchart is changed to reflect the relative popularity of each composer’s name as a search term during that week, starting at the first week of 01/2005. We can conclude from the visualization, for example, that the composer Verdi and Mozart are generally very popular. Mozart’s popularity had a breakout for a couple of years, and reached its peak some time in December 2009. From this information, we can trace back and consult, for example, news archives for the year 2009 to find out the reason behind this phenomenon.

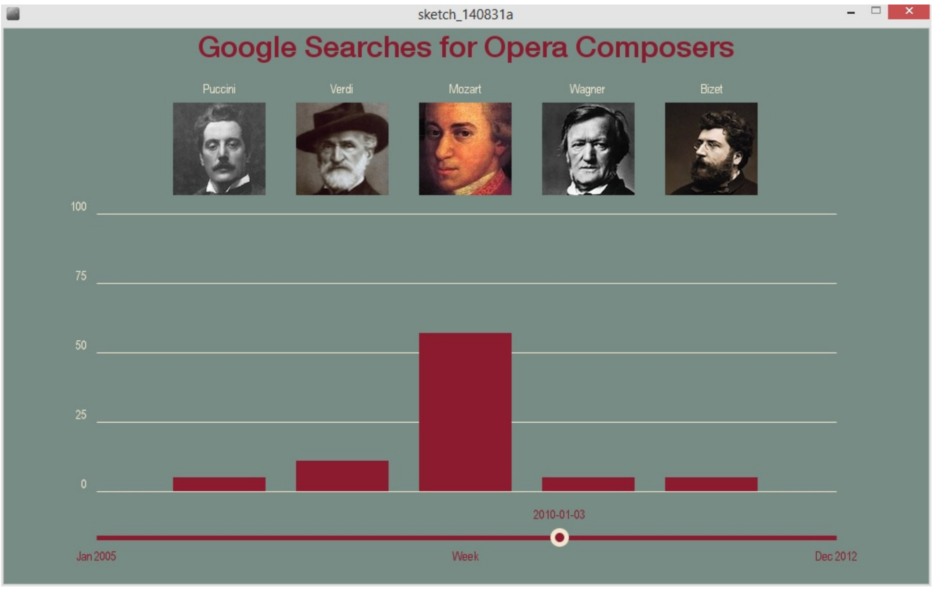


Figure 1: Visualization of Google searches for opera composers with Processing.

2.4 Conclusion

Processing is a fairly powerful platform for interactive visualization. It differs itself from other software packages in that it is a development platform, meaning the user has total control and freedom over the design and implementation of the visualization, which in turn enable many possibilities for the end user. Processing has a large collection of community-contributed libraries, which greatly extend the functionality and flexibility of the platform, even when working with huge data sets. The possible downside is the somewhat steep learning curve and complex visualizations mean complex coding. Processing has another advantage that it can export the visualization to various formats, either a desktop application or an app running on Android mobile devices, or even a JavaScript application to embed in websites.

3 Interactive Visual Analytics with Mondrian

3.1 Introduction

Mondrian is a desktop-based, data visualization application, whose development started in 1997 at AT&T Shannon Labs. Current development is taken place at the University of Ausburg. Germany with its main contributor being Martin Theus. It focuses on exploratory visualizations and seamless integration of categorical data. In addition to standard plots like histograms, barcharts, scatterplots, etc., Mondrian also provides advanced plots for high-dimensional, categorical (e.g. mosaic plots) or continuous data (parallel coordinates) and offers a wide range of interactions such as linked plots and advanced selection techniques for data exploration. Mondrian is purely Java-based.

3.2 Support for Exploratory Data Analysis through Interactive Visualization

3.2.1 Advanced Data Selection

One of the most notable features that Mondrian offers to support interactive visualization is its advanced selection technique. Selections is at the core of any interactive visualization because it is used to identify subgroups and patterns. The problem with standard ways of selecting data, as implemented in older visualization packages like [2] is that the old data will be lost, replaced by the new selection and thus refining selection or selecting over different plots is not possible. Mondrian's approach to data selection combines advanced techniques such as multiple combinations of selections using boolean functions (e.g. *and*, *or*, *xor*, *not*), selection sequencing and storing - as proposed in [3]. Mondrian keeps a list of any selection associated with a dataset, with each entry contains information like: screen coordinates and data coordinates of the selection, selection step, corresponding plot window and selection mode.

Selections are indicated by rectangles with eight handles that allow flexible resizing and various slicing techniques, as shown in Fig.2. Selections are stored in terms of data coordinates, which allows for correct coordinate translation in operations like zooming. Mondrian also translates selections into SQL code sequentially, ignoring boolean operators' precedence, modelled after the way the user thinks. An example: *S1 || S2 && S3* would be translated into *(S1 || S2) && S3.*

3.2.2 Use of Conventions

Mondrian's user interface and interactions are designed with regard to conventions, which allow for a flat learning curve and better usability. Interactions are categorized into various groups, including *Selections* (create selection rectangle, brushing, resize/slice selection, change selection mode), *Queries* (pop-up trigger on object via right-click), *Alterations* (zoom-out/in, change plot settings, reorder objects). Some interactions will vary between plot types.

3.2.3 Support for High Dimensional and Categorical Data

Besides linking with highlighting, Mondrian also offers advanced plots to support for visualizing high dimensional data. These include:

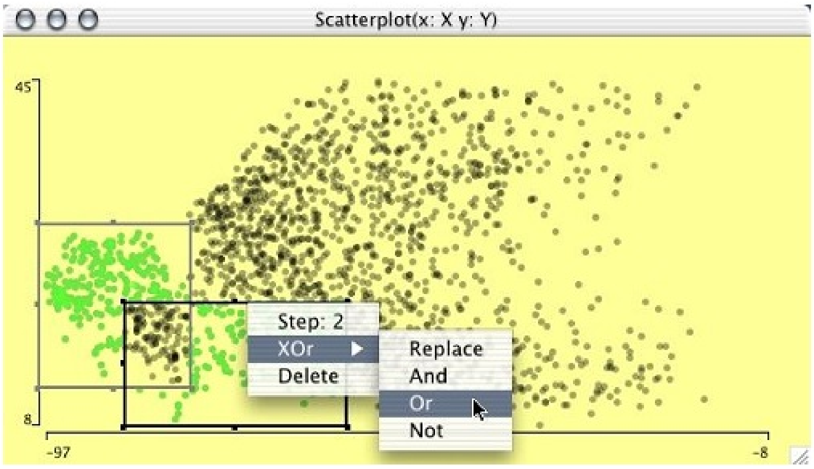
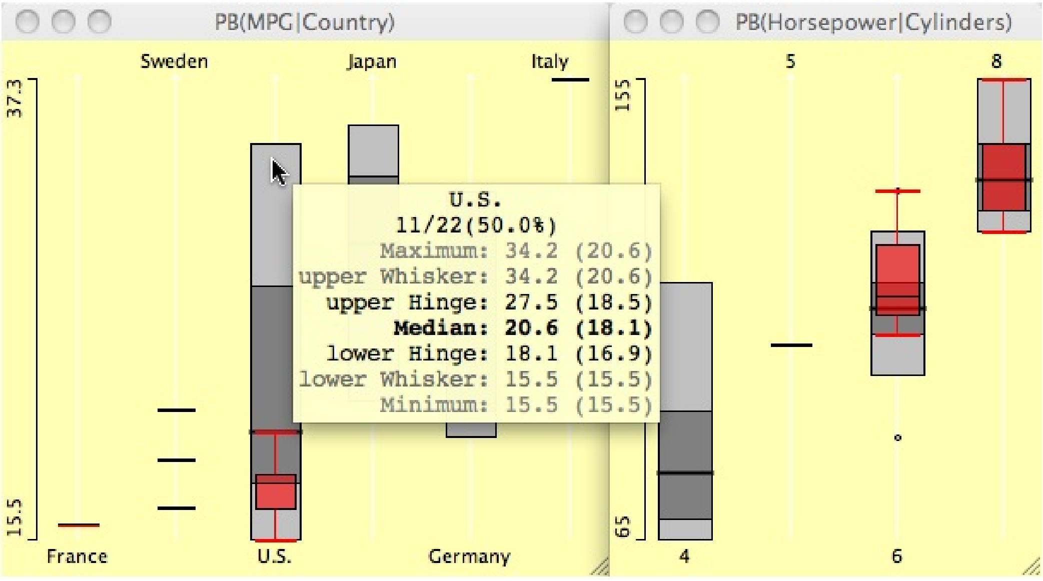


Figure 2: Selection rectangles.

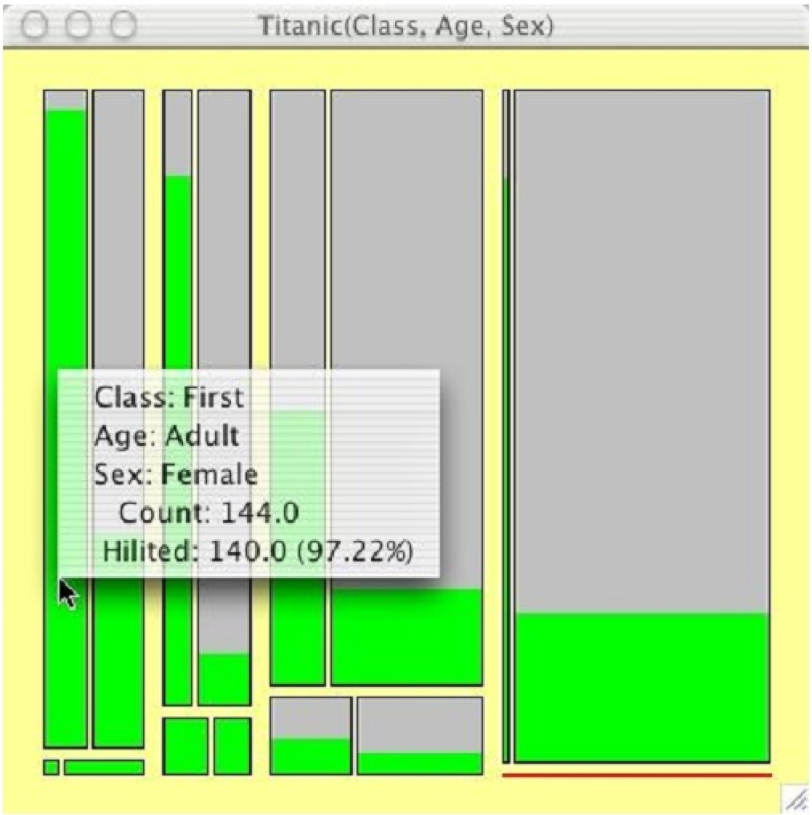
- *Parallel coordinates/Boxplots*: powerful plots for multi-variate data but only useful with interactions. Some advanced interactive versions of these plots are proposed in [] and []. Mondrian's implementation supports interactions like coordinates rearranging and axis zooming, as well as a special feature to plot categorical variables: a stacked barchart with left-to-right highlighting for each categorical variable.
- *Mosaic plots*: mostly offered as static version in other visualization packages (S-Plus, R) due to being relatively new. Mondrian offers to some degree interactions like variable reordering and a special feature, which allows for interactive graphical modeling of loglinear models based on mosaic plots, as shown in Fig.4
- *Weighted plots*: non-negative variable can be assigned as weight to make complex plots more flexible and interpretable.

Mondrian implements interactive barcharts and mosaic plots to support for analyzing categorical data, which include linked highlighting and interactive reordering of variables and categories.



```
           U.S.
        11/22(50.0%)
      Maximum: 34.2 (20.6)
upper Whisker: 34.2 (20.6)
  upper Hinge: 27.5 (18.5)
       Median: 20.6 (18.1)
  lower Hinge: 18.1 (16.9)
lower Whisker: 15.5 (15.5)
      Minimum: 15.5 (15.5)
```

Figure 3: Boxplots in Mondrian for a car data set.



Class: First
Age: Adult
Sex: Female
Count: 144.0
Hilited: 140.0 (97.22%)

Figure 4: Mosaic plot in Mondrian.

3.3 Enhancements in Standard Plots

Mondrian also implements minor tweaks in some of the standard plots, so that the process of exploratory data analysis can be made easier.

3.3.1 Barcharts

- Horizontal layout instead of vertical to allow for full-length viewing of category names
- Implements spine-plot view. Implements scrollable barcharts for large amount of categories.
- Provive four options to order categories
  - Lexicographic: default order, useful for looking up categories.
  - Manual sorting by user, which is activated by dragging the bars to new positions.
  - Absolute Size of Highlighting: sorts categories according to the absolute number of selected cases in a category.
  - Relative Size of Highlighting: sorts according to the relative amount of highlighting in the categories. Useful in spine-plot view to show the ordering of the selected proportions.
- Changes are propagated automatically to all other linked plots. Fig.5 shows two linked barcharts.

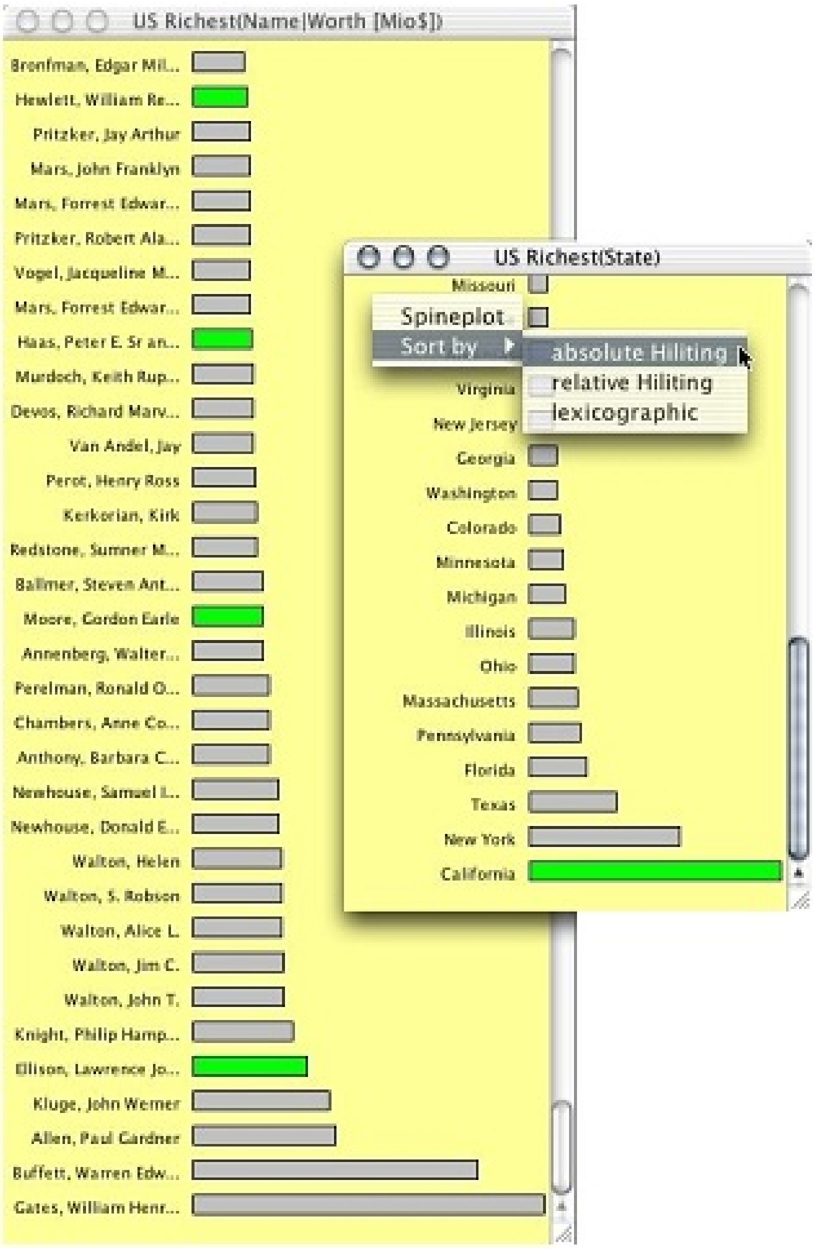


Figure 5: Two linked barcharts in Mondrian.

3.3.2 Histograms

- Ability to change the starting point and the width of the bins with plot scales being frozen during reparametrization to minimize visual distortion.
- Implements spinogram view (all same height bins, plotted next to each other), as illus-

trated in Fig.7

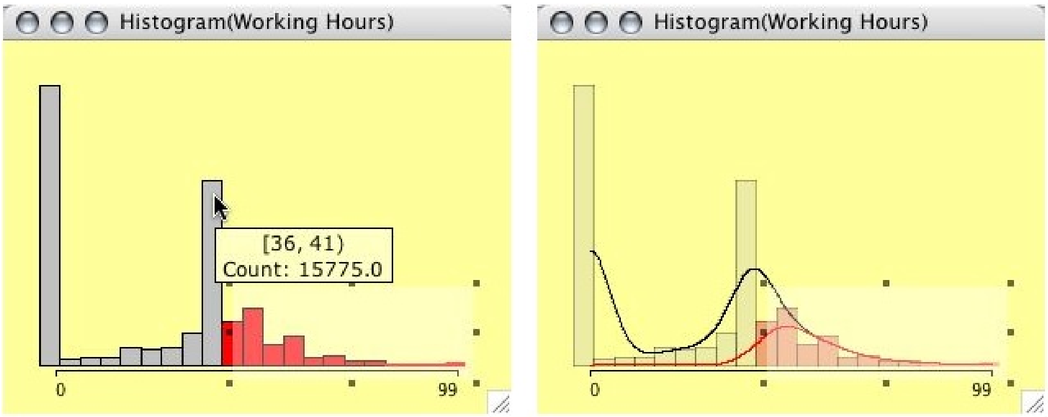


Figure 6: Histogram in Mondrian, visualizing the working hours of approx. 64000 household heads.

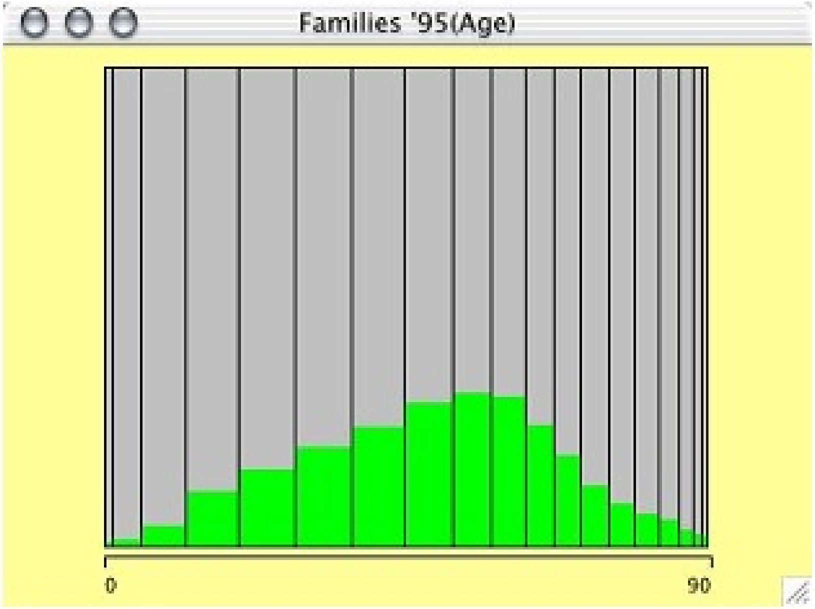
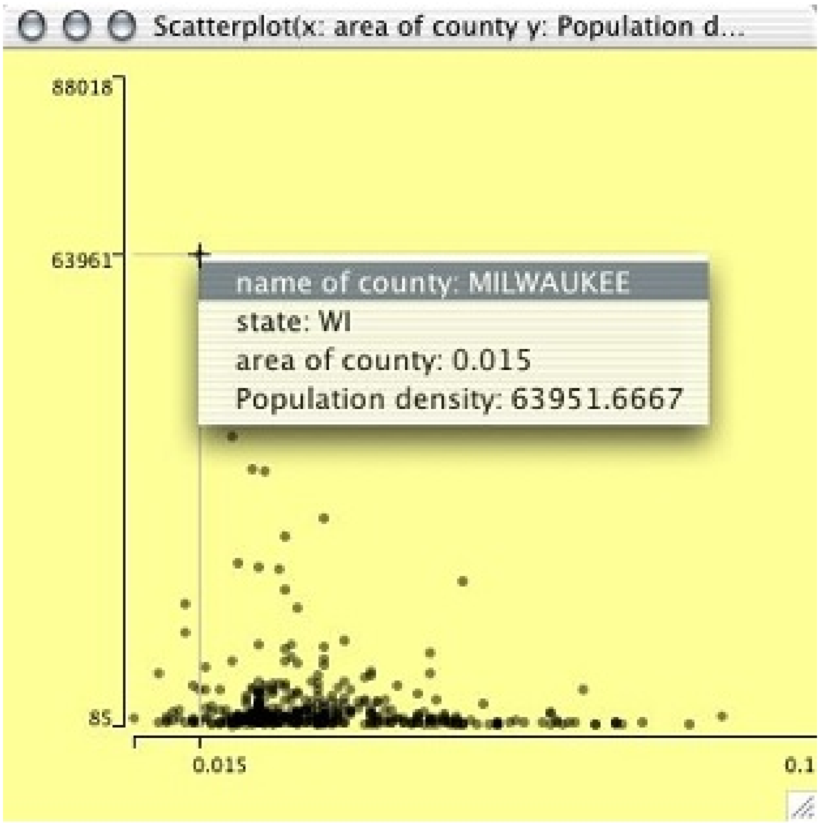


Figure 7: Spinogram view of the histrogram of the age distribution.

3.3.3 Scatterplots

- Maximum and minimum are constantly shown as basic orientation.

- Pop-up in plot window with detailed data for the closest point. Multiple variables can be chosen to be shown in detail in the pop-up. Variables with the same distance will be shown as a list in the pop-up.



name of county: MILWAUKEE
state: WI
area of county: 0.015
Population density: 63951.6667

Figure 8: Interrogation of data point in Mondrian's scatterplot.

3.4 Conclusion

Mondrian is 100% pure Java, which has the advantage of being platform independent. It supports a wide variety of standard plots, as well as advanced plots for high dimensional data. Its focus on exploratory data analysis means most plots support interactions. Mondrian can also establish direct connection to databases via the JDBC interface, which allows for analysis on huge data sets (i.e. big data). Current versions of the software also implement alpha-channel transparency to reduce clutters in plots with vast amount of data.

References

[1] FRY, Ben ; REAS, Casey: Processing Homepage. . – URL http://processing.org/

[2] SWAYNE, D. ; TEMPLE, D. ; BUJA, A. ; COOK, D.: Ggobi: Xgobi redesigned and extended. In: *Proceedings of the 33th Symposium on the Interface: Computing Science and Statistics* (2001)

[3] THEUS, M. ; HOFMANN, H. ; A., W.: Selection sequences - Interactive analysis of massive data sets. In: *Proceedings of the 29th Symposium on the Interface: Computing Science and Statistics* (1998)

[4] THEUS, Martin: Mondrian Homepage. . – URL http://stats.math.uni-augsburg.de/Mondrian/