DOI: 10.1111/j.1467-8659.2009.01696.x

Eurographics/ IEEE-VGTC Symposium on Visualization 2010
G. Melançon, T. Munzner, and D. Weiskopf
(Guest Editors)

Volume 29 (2010), Number 3

Supporting Exploratory Analysis with the Select & Slice Table

Yedendra B. Shrinivasan and Jarke. J. van Wijk, Eindhoven University of Technology, The Netherlands

Abstract

In interactive visualization, selection techniques such as dynamic queries and brushing are used to specify and extract items of interest. In other words, users define areas of interest in data space that often have a clear semantic meaning. We call such areas Semantic Zones, and argue that support for their manipulation and reasoning with them is highly useful during exploratory analysis. An important use case is the use of these zones across different subsets of the data, for instance to study the population of semantic zones over time. To support this, we present the Select & Slice Table. Semantic zones are arranged along one axis of the table, and data subsets are arranged along the other axis of the table. Each cell contains a set of items of interest from a data subset that matches the selection specifications of a zone. Items in cells can be visualized in various ways, as a count, as an aggregation of a measure, or as a separate visualization, such that the table gives an overview of the relationship between zones and data subsets. Furthermore, users can reuse zones, combine zones, and compare and trace items of interest across different semantic zones and data subsets. We present two case studies to illustrate the support offered by the Select & Slice table during exploratory analysis of multivariate data.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical User Interfaces (GUI)

1. Introduction

In interactive visualization, users select data items to drill down or highlight items in the visualizations. For selecting these data items, they use interaction techniques such as dynamic queries [AS94] and brushing [BC87, Che03], to specify conditions over functions of data attributes. During exploratory analysis, these selection techniques help users to progressively converge on interesting data items. Also, they can edit a selection specification, and thereby perform a divergent analysis.

Current visualization systems offer limited support to explicitly capture and reuse selections during an analysis. Often, brushing leads to selection of items, and when users change the visual mapping they can keep track of these selected items [HAW08, Gap10, Tab09, War94]. When they specify a new selection, the previous selection is lost. Hence, data selection is often transient in these visualization systems. It is difficult for users to manually keep track of these selection specifications during a long analysis process. Also, they cannot effectively reuse selection specifications, and compare the results of these specifications. Therefore, for effective reasoning based on data selection, we argue that support for capturing and manipulating selection specifications is highly useful during an exploratory analysis.

Areas of interest in data specified by data selection usually have a clear semantic meaning, unless users select items by accident. We enable users to capture such areas of interest in data as *Semantic Zones* or simply *Zones*. A zone holds either a selection specification, or a set of items extracted using the selection specification. It has a label provided by a user. Figure 1 shows four zones — *rich nations*; *developing nations*; *poor nations*; and *India, Brazil, and Kenya*. Also, in current visualization systems, users cannot quickly slice and dice the selected items over different subsets of the data to study the distribution of these items. Examples of such tasks are 'how many nations in different continents belong to each zone?' and 'how did the nations move to different zones over time?'

A popular approach to slice and dice a multi-dimensional dataset is a pivot table. The pivot table provides an aggregate summary of a data attribute by cross-tabulating the dimensions of a dataset. A visualization spreadsheet is another approach that helps users to compare visualizations representing different data sets side-by-side in a spread-

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

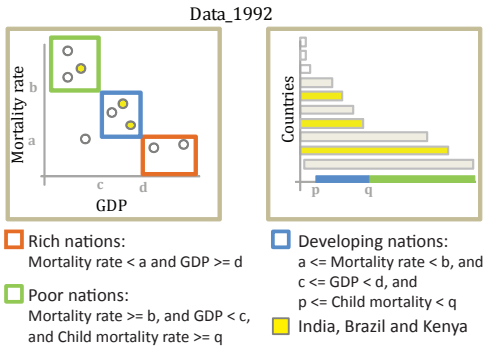

Figure 1: *Four semantic zones shown in two visualizations. A zone has either a data selection specification or a set of items extracted using a data selection specification. It has a label provided by a user.*

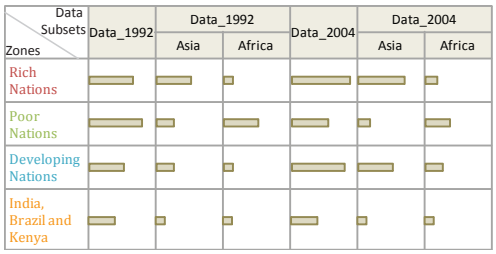

Figure 2: *A Select & Slice table showing the distribution of items of the four zones across different subsets of two datasets (Data_1992 and Data_2004). The length of a bar in a cell represents the number of nations.*

sheet [CRBK98]. It provides extensive cell manipulation operations similar to a spreadsheet. However, none of these interfaces can be directly used to manage and manipulate zones during an exploratory data analysis, for instance to see the contents of zones for different subsets of the data.

In this paper, we present a table interface that enables users to capture and manipulate zones during an exploratory analysis (see Figure 2). The table interface is used in addition to a data view that contains interactive visualization tools. Firstly, users can externalize zones from the data view and archive these in the header along one axis of the table. The labels of zones are displayed in the header of the axis. Secondly, users can retrieve items from different data subsets that match the selection specifications of zones. The data subsets are arranged along the other axis of the table; the labels of the data subsets are displayed in the header of the axis. A cell contains a set of items from a data subset that matches the specification of a zone. Thus, items of datasets are sliced based on the specifications of zones in the table. Hence we call this interface the *Select & Slice* table. Items in cells can be visualized in various ways, as a count, as an aggregation of a measure, or as a separate visualization, such that the table gives an overview of the relationship between zones and data subsets.

Next, users can edit specifications of zones using a zone editor attached to the Select & Slice table. During an analysis, they can reuse a zone specification by dragging its label from the table onto the data view. Then, users are enabled to drill down to a particular data set from the Select & Slice table in the data view. Next, they can logically combine the sets of items in the cells, and highlight the resulting items in the data view using simple mouse operations. Also they can study the distribution of items in the table using a set comparison operation and a keyword search. Thus, we adopt Shneiderman's information visualization mantra - overview first, zoom and filter, and details on demand - for manipulating zones during an exploratory analysis. Finally, we present two case studies that were conducted to understand the support offered by the Select & Slice table during exploratory analysis of multivariate data.

2. Related Work

First, we discuss existing techniques to capture and archive selections in visualization systems. Next, we present visualization techniques that are closely related to the Select & Slice table.

2.1. Selection Management

Several visualization systems help users to capture areas of interests in data specified through selection techniques. Visualization systems such as Aruvi [SvW08], Cross-filtered views [Wea08], Gapminder [Gap10] and Flare [HAW08] capture brushing as a declarative query, and reuse it when the view is transformed. QlikView [Qli10] tracks the users' selection process and helps them to define alerts based on the data attribute criteria. Doleisch et al. [DGH03] present a framework for capturing features using brushing, and archive these features using a feature definition language. They use a tree view to archive and edit the features. These archived features are used to steer 3D visualization of computational simulation data. Similarly, streamline predicates [SS06] are used for capturing flow structures while visualizing flow simulation data. In interactive analysis of simulation data [KMG*06], function graphs of attributes are used to specify areas of interest. These systems mainly focus on archiving and editing those regions of interests during exploratory analysis. They do not support reuse of selection specifications on subsets of data, and the comparison of the results of these specifications.

In Tableau [Tab09], users can create and analyze subsets of data using computed sets. A computed set is used as a derived dimension in the analysis. However, the computed sets cannot be simultaneously sliced across different subsets of data. Visualization systems such as XMDV [MW95]

and Mondrian [The02] support brush editing. Users can change the logical composition of brushes during an exploration process. XMDV can simultaneously display multiple N-dimensional brushes to compare brush results. Using a similar approach, Elmqvist [EDF08] supports multiple brushes in a scatterplot. Chen [Che03] uses a data-flow model to define multidimensional brushes. The number of brushes that can be simultaneously displayed in visualizations (XMDV and Mondrian), and tracked during animation (Aruvi, Gapminder and Flare) is limited. Moreover, in all these systems, users also cannot simultaneously reuse these brushes on different subsets of data, and compare the results of these brushes side by side.

2.2. Visualization techniques

Visualization techniques help users to interactively explore multi-dimensional data. Examples of such techniques are interactive axis reconfiguration, tables, re-orderable matrix, multi-dimensional scaling, dimensional stacking and glyphs. The Select & Slice table uses a tabular approach to slice and dice items of datasets using zones. This approach is closely related to spreadsheets and pivot tables.

A spreadsheet displays a grid of cells. A spreadsheet cell contains a value, or a formula that defines the content of the cell by combining values of other cells in the spreadsheet. When the content of a cell is changed, the entire sheet is automatically re-calculated. In a visualization spreadsheet [CRBK98], cells contain visualization operators that transform data into views. When the content of a cell is changed, all views in the spreadsheet are automatically updated. In contrast to a spreadsheet, users cannot directly edit contents of cells in the Select & Slice table. They can only edit the specifications of zones and subsets of datasets to change the contents of cells; and cells provide an overview of the relationship between zones and data sets. As a result, spreadsheets offer much flexibility and focus on management and reuse of data flows; whereas the Select & Slice table aims at offering ease of use for the management and reuse of selection specifications.

A pivot table, found in spreadsheet programs such as Microsoft Excel and OpenOffice.org Calc, helps to slice and dice multi-dimensional data. A pivot table provides an aggregate summary of a data attribute by cross-tabulating the dimensions of a dataset. The pivot table has hierarchical clusters of data attributes along its row and column headers. Polaris [STH08] adopts a tabular layout similar to a pivot table; its cells have visualizations automatically chosen based on the composition algebra and the graphic design criteria. In contrast to the pivot table and Polaris, the Select & Slice table headers have zones along one axis of the table and subsets of data along the other axis of the table. Also, a cell contains items retrieved from a data subset that match the specification of a zone. It provides visual summaries of the items in various ways, as a count, an aggregation of a measure, or as a separate visualization. The pivot table shows grand summaries of the data field at the end of the rows and columns. Items in the cells of the Select & Slice table are not mutually exclusive, as the zones can define overlapping areas of interest in data. Hence, the table cannot show grand summaries at the end of rows and columns.

In summary, a spreadsheet offers much flexibility, but does not directly support handling of user defined semantic zones and subsets of the data; a pivot table is too rigid in the sense that along both dimensions of the table the data have to be partitioned. We argue that the solution that we provide, that is, a combination of user defined zones and dataset slicing, is often very useful for analysis and visualization purposes. In the following section, we describe the implementation of the Select & Slice table to support capturing and manipulating zones during an exploratory analysis.

3. Approach

To support reasoning based on data selection in information visualization, we enable users to

- construct the Select & Slice table during an exploration process by
  - capturing the selection specifications or selected items from the data view as zones with user-defined labels;
  - retrieving items from different subsets of data using the specifications of zones; and
  - visualizing the retrieved items in various ways, as a count, as an aggregation over a measure, or as a separate visualization.
- study the distribution of the items in the table;
- support drilling down to a particular subset of data from the table in the data view.

The Select & Slice table is implemented in the Aruvi visualization system [SvW08]. Aruvi contains three integrated views — a data view, a navigation view and a knowledge view — to support the analytical reasoning process in information visualization. The data view offers interactive information visualization tools. Currently, users can explore data using multiple scatterplots and barcharts attached to a dynamic query interface in the data view. Multiple data sets from different databases can be accessed simultaneously during an analysis. Users provide a unique identifier to a dataset while loading it into Aruvi. The navigation view provides an overview of the exploration process by presenting all automatically captured visualization states to the user. The knowledge view enables users to capture interesting aspects of their exploration process by bookmarking visualizations, and recording and reordering findings using diagramming techniques. They can organize findings to build a case. The Select & Slice Table is implemented as a part of the knowledge view. The table can also be used to build a case by manipulating zones, and by studying the distribution of items retrieved from datasets for zones.

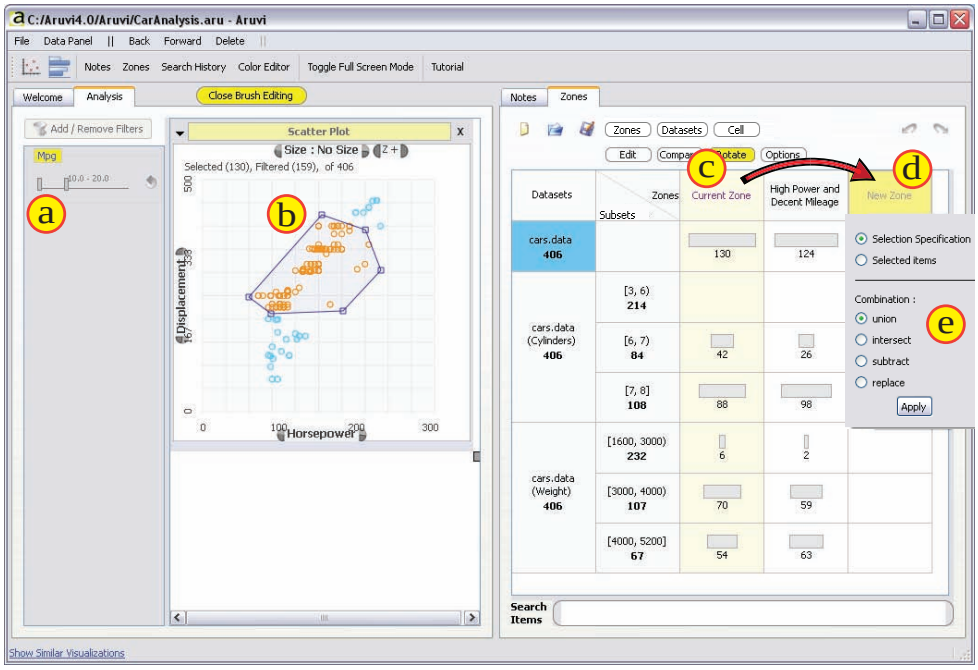

Figure 3: *The Select & Slice Table is shown as a part of the knowledge view in the Aruvi visualization system. A filter (a) and a brush (b) are combined to define the current zone (c). Users can define a new zone by dragging the current zone, an existing zone or a cell on to the 'new zone' place holder (d). (e) The new zone composition menu.*

We make use of a classic cars dataset from the 1983 ASA Statistical computing and graphics data expo (http://stat-computing.org/dataexpo/1983.html) to illustrate the features of the Select & Slice table. The dataset contains 406 cars with 9 attributes such as model name, mpg, number of cylinders and acceleration.

3.1. Constructing the Select & Slice Table

Encoding Selection

We encode selections specified by users in the data view using a SQL-like query language as in earlier systems (e.g., [OSAH98, HAW08, LRB*97, DKR97]), and graphics operations such as object in polygon test. A selection specification consists of conditions over functions of data attributes. In Aruvi, users can specify a selection using dynamic query widgets and brushing. First, items are optionally filtered using dynamic query widgets. These dynamic queries are directly expressed using SQL clauses. Then, a brush can be used to select items in the visualizations. A brush is specified by picking items, by dragging a rectangle, or by drawing a lasso over items in the visualizations. Picking selects an item using its object id (primary key). A rectangle brush is expressed using SQL BETWEEN or IN operators. For a lasso selection, first its bounding box is expressed as a rectangle brush; then, the selected items are identified using an object in polygon test. The type of visualization determines how these SQL and graphics operators are applied on attributes to select items in the visualization. For instance, in a scatterplot, a rectangle brush is expressed as the intersection of range queries on x- and y- axes attributes; in a barchart, it is expressed as the intersection of a range query on the measure axis, and a set of items selected in the category axis. Finally, the current selection in the data view is defined by intersecting dynamic queries (Figure 3a) and brushing (Figure 3b).

Creating a new Select & Slice Table

When a new Select & Slice table is created, it is populated with a current zone and the current dataset as shown in Figure 4a. The current zone holds the current selection specification from the data view throughout an exploration process. The current dataset is highlighted in blue in the Select & Slice table. It also shows the number of items selected from the current dataset based on the current selection in the data view, using a bar representation.

Defining Zones

A new zone is defined by dragging the current zone header onto the 'New Zone' placeholder (Figure 3d). Next, users can choose to store either the selection specification or the selected items (Figure 3e); and provide a label for the new zone. Figure 4b highlights a newly defined zone in green.

Figure 4: *(a) A new Select & Slice Table. (b) The table with a new zone (highlighted in green), and three new subsets of the data based on attribute slicing (highlighted in red).*

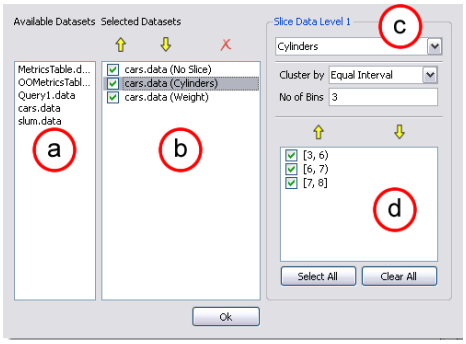

Figure 5: *Data subset definition interface. (a) A list of available datasets using unique dataset identifiers provided by the user. (b) A list of subsets of datasets. The labels for the subsets are automatically generated by combining the dataset identifier, and the attribute name that is used for subsetting. Users can rearrange, show, hide or remove a selected dataset from the list. (c) A subset definition panel. (d) The subsets of a selected dataset. An ordinal interval is represented using a standard interval notation; where [3, 6) means* $3 <= x < 6$.

Defining Datasets

Users can obtain more insight in the distribution of items in zones by defining subsets of datasets. For instance, in Figure 4b the original dataset is split up according to *Origin* countries. Users can subset a dataset based on one of its attributes. A nominal attribute can be used to subset data using either its unique domain values or groups of these. An ordinal attribute can be used to subset data using a clustering method such as equal intervals, quartile, percentile, standard deviation, unique values and custom intervals. A temporal attribute can be used to subset data by monthly, quarterly, yearly or custom intervals. Figure 5 shows the data subset definition interface in the Aruvi visualization system. Also, they can change the current dataset in the data view by selecting a dataset in the table.

Defining Cell Contents

Each cell contains a set of items. Users can request to show a summary (the number of elements, the average value of an attribute, etc.) or a visualization of all items. A bar or a bubble is used to visualize a summary of the items in a cell. First, the number of items, or a measure such as a total or an average value of an attribute is used to determine the length of a bar or the radius of a bubble in a cell. The lengths of bars and the radii of bubbles are normalized across cells of the table to simply comparison across rows and columns. Bars can be aligned either to the center, or to the left of cells; bubbles are placed at the center of cells. A label showing the number of items is placed below a bar or a bubble. Users can choose either a linear or a logarithmic scale for mapping the number of items onto a bar and a bubble.

To show all items, the active visualization from the data view can be shown in each cell and the items in the cell are plotted in that visualization. In this way, a visualization matrix is created to provide an overview of items in the cells of the table. Currently, in Aruvi a scatterplot can be shown in each cell.

Manipulating Zones

Dataset header elements hold selection specifications for data subsets. Cells contain the selection specifications of both zones and data subsets to retrieve items. Hence, each element of the headers and also each cell represents a zone. So, we enable users to define new zones also by dragging a header element or a cell onto the 'New Zone' placeholder. Existing zones can be combined using one of the operations union, intersect, subtract and replace (Figure 3e). They can also reuse a zone definition (filters and brushes) in the data view by dragging a zone onto the current zone in the table.

Most operations on zones can be done using simple manipulations. For cases where detailed inspection and editing is needed, users can also manipulate zones using a zone editor. The zone editor has two components: a combination editor (Figure 6a) and a list of selection specifications (Figure 6b). The combination editor allows users to logically compose selection specifications using a parse tree representation. A parse tree completion assistant helps users to construct a valid combination of selection specifications. Below the combination editor, a list of selection specifications is shown. Users can directly edit the selection specifications created using dynamic query widgets in the data view. For those selection specifications created using brushing, they

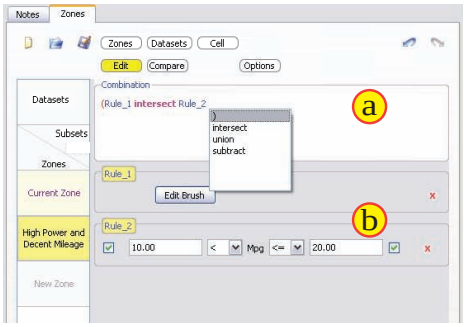

Figure 6: *Semantic zone editor. (a) Zone composition editor with a parse tree completion assistant. (b) A list of selection specifications (filters and brushes) that defines a semantic zone.*

can directly edit the corresponding brushes by restoring the original visualization state via the 'Edit Brush' button. An undo and redo mechanism is provided to the users for zone manipulation.

3.2. Studying Items Distribution

A set comparison operation and a keyword search are provided to study the distribution of items in the Select & Slice table. The zones and data subsets can be rearranged and rotated to support side-by-side comparison.

Set Comparison

A user can compare items of a cell with items of the other cells in the table. When the user double-clicks a cell, the Select & Slice table enters comparison mode. The selected cell used for comparison is filled with light red (Figure 7a). To identify the number of similar items in a cell with respect to the selected cell, the items of the cell are intersected with the items of the selected cell. The number of similar items in each cell with respect to the selected cell is shown as a ratio in blue below the bars or bubbles. Also the similarity ratio is visualized through a blue filling in the bars or bubbles. Using this comparison view, users can trace items across different zones and data subsets. For example, Figure 7 shows the distribution of Japanese cars (Figure 7a) across different zones and different subsets of the car dataset. One-fourth of the 'cars having good acceleration' (51 out of 220 cars) are Japanese in the dataset; and these Japanese cars (50 out of 51 cars) have between 3 and 6 cylinders (see Figure 7b). All these 51 Japanese cars weigh less than 3000 pounds (Figure 7c). This items distribution study shows Japanese car industry did not focus on producing powerful and heavy cars, but manufactured lightweight cars with good acceleration. For other aggregations apart from count, a blue filling and a gray filling in a cell represent a summary for the similar items and all items respectively.

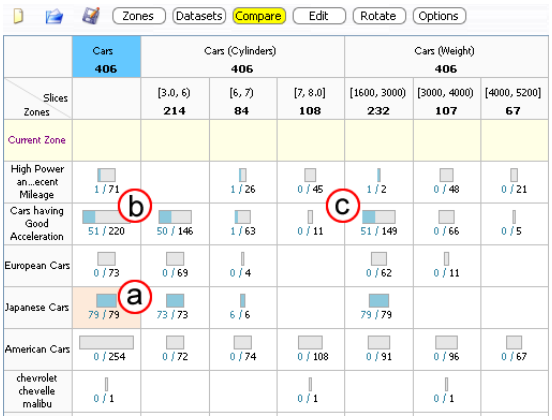

Figure 7: *Set comparison view shows the distribution of Japanese cars (a). (b) One-fourth of the 'cars having good acceleration' are Japanese in the dataset; these Japanese cars (50 out of 51 cars) have between 3 and 6 cylinders. (c) All these 51 Japanese cars weigh less than 3000 pounds.*

Keyword Search

Figure 8: *(a) Keyword search interface. (b) An item suggestion list. (c) Search results are visualized using colored dots in cells. A dot is colored based on its corresponding keyword's color in the search interface.*

Users can search for individual items in the table using a keyword search interface (Figure 8a), with an item suggestion list (Figure 8b). The keywords are separated by a '+' character and assigned a color. The search results are visualized using colored dots in cells. A dot is colored based on the corresponding keyword color in the search interface. Figure 8 shows a user searching for three cars: 'mazda glc 4', 'Chevrolet malibu' and 'Chevrolet chevelle malibu'. The search results are shown using colored dots in cells (Figure 8c). Using this keyword search, the user could infer that

the three cars have good acceleration; and also identify their origin country and cylinder specification.

3.3. Drill Down Analysis

During an exploratory analysis, users can compose a complex brush by selecting items in the table, and drill down to investigate these items in the data view. The brush is defined by logically combining the selected cells. Cells can be added, intersected or subtracted using click, shift+click, and ctrl+click; and these cells are marked green, red and blue respectively. When a user selects a cell, the selection status of that cell is toggled and the selection status of other cells is kept constant, similar to multi-selection mode in a list box widget. The order of the selection sequence is shown in the highlighted cells. The selection is cleared by pressing the escape key. In Figure 9, the selected cells in the Select & Slice table show 'American cars with 8 cylinders that are not heavy' (green ∩ blue \ red). These cars are highlighted in the scatterplot (Figure 9a). Detailed information about items selected by the brush is shown in the table's context menu (Figure 9b). They can also archive these items along with detailed information as a comma-separated file to study them in other software systems or for reporting purposes.

4. Case Studies

We present analysis processes of two data analysts to illustrate the support offered by the Select & Slice table for an exploratory analysis. The analysts are experts from different domains such as software quality analysis and urban planning. They often use information visualization tools for their day-to-day data analysis tasks.

First, analysts carried out their domain specific data analysis tasks using the Aruvi visualization system. Following that we conducted an informal interview to understand the usefulness of the Select & Slice table. We present our observations of their analysis processes, and discuss their feedback on the Select & Slice table.

4.1. Software Quality Analysis

The first analyst is a software quality consultant at the Laboratory for Quality Software, TU/e, The Netherlands. He derives software metrics, package structure and call-graphs for software systems from source-code and visualizes them to check their design quality. There are ten package design principles for developing an ideal package structure for a software system [Mar09]. Software quality analysts often use two metrics to study the quality of a package design: the stability metric (I), which measures the stability of dependencies, and the abstractness metric (A) [Roubtsov, personal communication]. There are three zones based on the relationship between A and I (Figure 10a). A *zone of pain*, where A and I are close to 0, contains packages that are rigid and

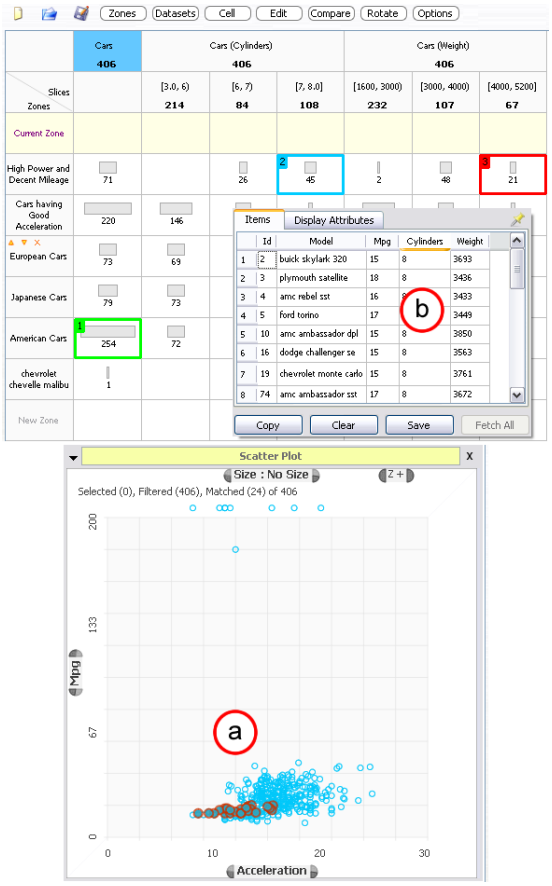

Figure 9: *Support for drill-down analysis. The selected cells 1, 2 and 3 highlighted in green, blue and red respectively compose a brush – 'American cars with 8 cylinders that are not heavy'. These cars are highlighted in the scatterplot (a). (b) Detailed information about those cars.*

cannot be changed or extended. A *zone of uselessness*, where A and I are close to 1, contains packages that are abstract and have no dependencies. The *acceptable packages* are close to the diagonal line connecting (A=0, I=1) and (A=1, I=0).

The analyst used Aruvi to compare two versions of JBoss, an enterprise application server. Initially, he loaded two datasets — *JBoss4* and a recent version of *JBoss* (JBoss 4.3) into Aruvi. He started exploring the JBoss4 dataset using a scatterplot. He plotted A along the x-axis and I along the y-axis. Using this view, he defined three zones — *Zone of pain*, *Zone of uselessness* and *Acceptable packages* in the Select & Slice table. Using these definition, he carried out two different analyses.

In the first analysis, he constructed a Select & Slice by slicing the two JBoss datasets with the three zones. He compared the recent version of JBoss against the previous version using the table. For this, he selected the acceptable

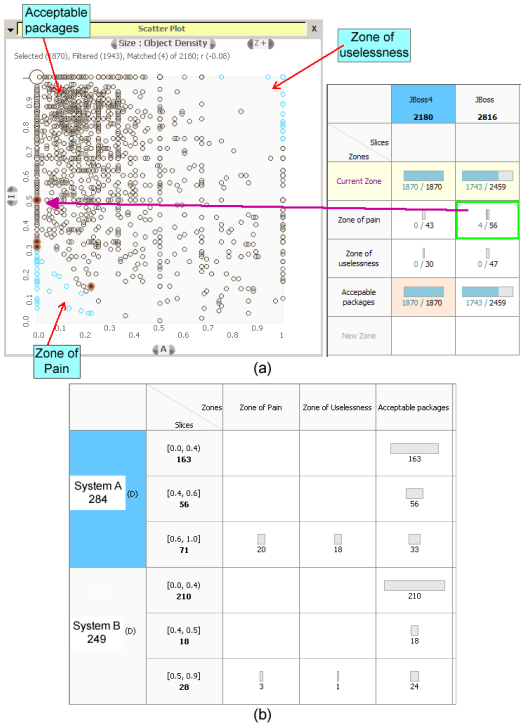

Figure 10: *Software quality analysis. (a) Comparison of two different versions of the JBoss, an enterprise application server. (b) Comparison of two different financial management software systems.*

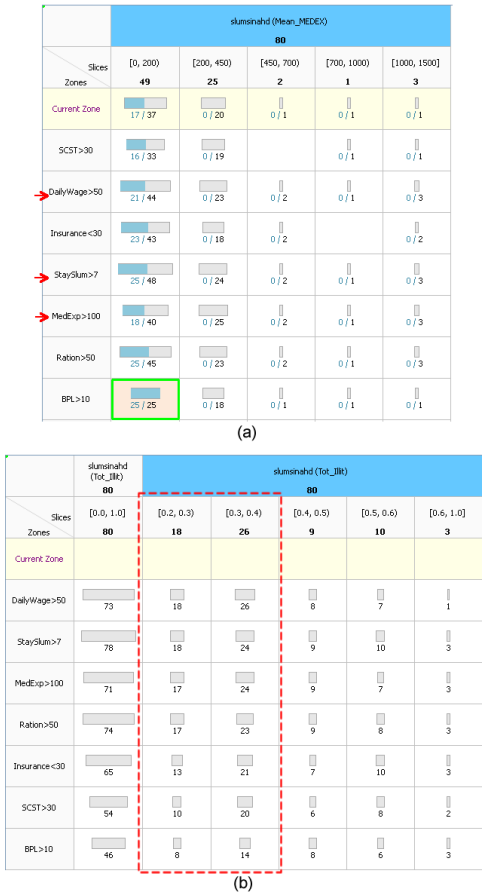

Figure 11: *Socio-economic data investigation. (a) Medical Expenses trend analysis and (b) The relationship between illiteracy and the seven zones helped the analyst to identify the intrinsic vulnerable slums.*

packages of JBoss4, and switched to comparison mode. The Select & Slice table, in Figure 10a shows this comparison. Based on this comparison, he studied the evolution of packages across two versions. He found that four of the acceptable packages from the previous version have moved to the zone of pain in the recent version. He highlighted the four packages in the scatterplot (Figure 10a) that visualizes the JBoss4 dataset. From this, he identified that one of the four packages was strongly affected (see purple arrow in Figure 10a); while the other three were already in the borderline. He hypothesized that the package might be strongly affected due to the changes made to incorporate some new features.

In the second analysis, he studied two finance management software systems from two different vendors — System A and System B (their names are sanitized), using the same approach as in the previous analysis. He reused the zones definition from the previous analysis. He compared the two systems by comparing the number of packages in the Zone of Pain and Zone of Uselessness. Based on these numbers, he found that system B has a good package design compared to system A.

The analyst usually follows a mathematical approach based on the normalized distance (D_n) to the diagonal line for identifying the acceptable packages. According to this approach [SRvdB09], packages that have $D_n < (\mu_{D_n} + 2\sigma_{D_n})$ are acceptable packages. However, this approach cannot explicitly identify if a package belongs to a zone of pain or zone of uselessness. To verify this approach, he constructed a new Select & Slice table by slicing the three zones with six data subsets (3 subsets for both systems). He divided the two datasets based on their D_n attribute (D) into three bins, using the standard deviation clustering method. This table is shown in Figure 10b. He found that some of the packages are found acceptable in the Select & Slice table, even when $D_n > (\mu_{D_n} + 2\sigma_{D_n})$. Also, he could locate and visualize these packages in the scatterplot, to support this claim. Thus, in addition to validating the results using the mathematical approach, he could also explicitly identify the packages and understand their distribution using the Select & Slice table and the data view.

Afterwards, we asked the analyst to explain the key as-

pects of the Select & Slice table that made a difference in his analysis process. He said that "defining zones using lasso selection in the scatterplot to analyze data based on design principles was a quite handy and natural way of doing analysis. I could also verify the zones approach with our mathematical approach."

4.2. Social Data Analysis

The second analyst is an urban planner working at the Centre for Environmental Planning & Technology University, Ahmedabad, India. He investigates socio-economic data for slums in Ahmedabad. His analysis had two main goals: to understand the factors affecting the medical expenses of people in slums, and to understand the reason behind such trends.

For this analysis, he loaded the socio-economic data of Ahmedabad (*slumsinahd*) into Aruvi. He explored the data using a scatterplot. During the exploration process, he identified 7 factors based on the demographics and socio-economic indicators to locate slums having poor living conditions. He used dynamic query widgets to specify these factors, and externalized these into 7 separate zones. They are the percentage of economical backward class people in a slum (*SCST* > *30%*), the number of people having temporary jobs (*daily wage* > *50*), the number of uninsured people (*insurance* < *30*), the number of people who have stayed in slums over 7 years (*stayslum* > *7*), the monthly medical expenses (*medexp* > *100* Indian Rupee - INR), the number of people who have access to the public distribution system (*Ration* > *50*) and the number of people below the poverty line (*BPL* > *10*).

To understand the trends in the medical expenses, the analyst divided the dataset using mean monthly medical expenses into 5 custom interval bins. The Select & Slice table in Figure 11a shows the overview of relationship between the 7 factors and the mean monthly medical expenses. He found that around 60% of slums (49 out of 80) have mean monthly expenses below 200 INR. Most of these slums fell under the poor socio-economic conditions described in zones such as daily wage (44 out of 49 slums) and stay in slums over 7 years (48 out of 49 slums). Then he compared the slums below the poverty line (BPL > 10, highlighted in Figure 11a) against the other factors in the first column. He found that for most of the slums below the poverty line (18 out of 25 slums) the monthly medical expenses constituted more than 50% of their monthly earnings (BPL in Ahmedabad is at 436 INR per month. The monthly medical expenditure is about 200 INR).

By analyzing the table in Figure 11a, he hypothesized that the high level of temporary jobs (like daily waged labor, unskilled labor) are because of the illiteracy prevailing in the slums. Subsequently, they are not able to improve their economic background as they do not have access to better education and training. Therefore, they are stuck in poor living conditions. However, the poor living conditions lead to high medical expenses. To prove these hypotheses, he projected these zones on the dataset divided using total illiteracy rate into 5 custom interval bins above 20% of total illiteracy rate (20% of the population is elderly and kids). The new Select & Slice table is shown in Figure 11b. He found that most of the slums have between 20 and 40% total illiteracy rate. Also all these slums have a high number of people with temporary jobs and high monthly medical expenses. Based on this view, he could affirm his hypotheses. He also concluded that these slums are the intrinsic vulnerable slums which are vulnerable to even small fluctuations in the socioeconomic conditions.

During the informal interview session, the analyst explained the key differences made by the Select & Slice table in his analysis process. Usually, he uses Microsoft Excel for analyzing the data. He would study the effects of the factors one at a time; however, he could not analyze them simultaneously. Also, he noted that a pivot table cannot be used for this purpose, where items are partitioned over the row and column attributes. He said "possibly I could have done this in Microsoft Excel. However, I could have never done the analysis so quickly and without breaking my head. Slicing Zones by different subsets of data helped me to put all my conditions parallel and compare them simultaneously." However, he felt that if these slum locations are plotted geographically, he could correlate the attribute values with other spatial accessibility functions, in order to make a better conclusion.

5. Conclusion

In this paper, we presented the Select & Slice table that helps to cross-tabulate semantic zones and data subsets. Semantic zones are areas of interest in data space specified through conditions over data attributes or as functions of data attributes that have a clear semantic meaning. Using the Select & Slice table, users can define and manipulate zones; and understand the relationship between zones and data subsets, visually and interactively. In addition, they can drill-down to a particular data subset, and investigate items of the table in the data view using drag & drop and other simple mouse operations. They can also get an overview of the distribution of items in the table using a set comparison operation, and a keyword search. Finally, we presented two case studies that illustrated the support offered by the Select & Slice table for exploratory data analysis.

Acknowledgements

This work is done under the Expression of Interest project. The project is supported by the VIEW programme of the Netherlands Organisation for Scientific Research (NWO) under research grant no. 643.100.502. We would like to thank the reviewers for their insightful comments, and the two analysts who participated in the case study.

References

[AS94] AHLBERG C., SHNEIDERMAN B.: Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *Proc. ACM CHI '94* (1994), pp. 313–317.

[BC87] BECKER R. A., CLEVELAND W. S.: Brushing scatterplots. *Technometrics 29*, 2 (1987), 127–142.

[Che03] CHEN H.: Compound brushing. In *Proc. IEEE InfoVis '03* (Oct 2003), pp. 181–188.

[CRBK98] CHI E. H., RIEDL J., BARRY P., KONSTAN J.: Principles for information visualization spreadsheets. *IEEE Computer Graphics and Applications 18*, 4 (1998), 30–38.

[DGH03] DOLEISCH H., GASSER M., HAUSER H.: Interactive feature specification for focus+context visualization of complex simulation data. In *VISSYM '03: Proceedings of the symposium on Data visualisation 2003* (2003), pp. 239–248.

[DKR97] DERTHICK M., KOLOJEJCHICK J., ROTH S. F.: An interactive visual query environment for exploring data. In *Proc. ACM UIST '97* (New York, NY, USA, 1997), ACM, pp. 189–198.

[EDF08] ELMQVIST N., DRAGICEVIC P., FEKETE J.-D.: Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1141–1148.

[Gap10] Google gapminder. http://www.gapminder.org/, 2010.

[HAW08] HEER J., AGRAWALA M., WILLETT W.: Generalized selection via interactive query relaxation. In *Proc. ACM CHI '08* (2008), pp. 959–968.

[KMG*06] KONYHA Z., MATKOVIC K., GRACANIN D., JELOVIC M., HAUSER H.: Interactive visual analysis of families of function graphs. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1373–1385.

[LRB*97] LIVNY M., RAMAKRISHNAN R., BEYER K., CHEN G., DONJERKOVIC D., LAWANDE S., MYLLYMAKI J., WENGER K.: Devise: integrated querying and visual exploration of large datasets. In *Proc. ACM SIGMOD '97,* (May 1997), pp. 301–312.

[Mar09] MARTIN R. C.: Design principles and design patterns. http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf, Accessed on Sept. 14 2009.

[MW95] MARTIN A. R., WARD M. O.: High dimensional brushing for interactive exploration of multivariate data. In *Proc. IEEE Visualization '95* (Nov 1995), pp. 271–278.

[OSAH98] OLSTON C., STONEBRAKER M., AIKEN A., HELLERSTEIN J. M.: Viqing: Visual interactive querying. In *Proc. IEEE Visual Languages '98* (Washington, DC, USA, 1998), IEEE Computer Society, p. 162.

[Qli10] Qlikview. http://www.qlikview.com/, 2010.

[SRvdB09] SEREBRENIK A., ROUBTSOV S., VAN DEN BRAND M.: D_n-based architecture assessment of java open source software systems. In *Proc. International Conference on Program Comprehension '09* (2009).

[SS06] SALZBRUNN T., SCHEUERMANN G.: Streamline predicates. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1601–1612.

[STH08] STOLTE C., TANG D., HANRAHAN P.: Polaris: a system for query, analysis, and visualization of multidimensional databases. vol. 51, pp. 75–84.

[SvW08] SHRINIVASAN Y. B., VAN WIJK J. J.: Supporting the analytical reasoning process in information visualization. In *Proc. ACM CHI '08* (2008), pp. 1237–1246.

[Tab09] Tableau software. http://www.tableausoftware.com/, 2009.

[The02] THEUS M.: Interactive data visualization using mondrian. *Journal of Statistical Software 7*, 11 (11 2002), 1–9.

[War94] WARD M. O.: Xmdvtool: integrating multiple methods for visualizing multivariate data. In *VIS '94: Proceedings of the conference on Visualization '94* (1994), pp. 326–333.

[Wea08] WEAVER C.: Multidimensional visual analysis using cross-filtered views. In *IEEE Symposium on Visual Analytics Science and Technology* (Oct. 2008), pp. 163–170.