United States Economy Visualization

Viktor Sandberg

Abstract— This projects was made in the course TNM048 Information Visualization with the goal of creating an application that uses a large data set with data about the real estate in United States. The data was used to create a visualization tool that gives the user an easy and intuitive way of learning about the economy in the country. The main purpose of the application is to analyze the average yearly income of the people in the country, and try to draw a conclusion why people in different areas do have a lower income.

Index Terms—Multivariate data, Economy Visualization, Data mining

1 Introduction

Economy is not something that is easily interpreted when looking at plain numbers in tables, but is easier when reading graphs or other visualization tools. Studies have shown that people who interact with data will more likely learn and remember, especially when it comes to interaction with visualizations [4]. This was the reason for creating an application were the focus was on having interaction with the data. The app's purpose is to make it easy for the user who are interested in the economy of United States to get an overview of how the economy is in the country. The application will also compare the economy differences between people who are having a High School degree and those who do not. The user can interact with the tool by navigating through each state of the country, which displays different kinds of information about that selected state.

This report will discuss how this application was made, explain what data is in the data set, the implementation of a data mining technique and how this project can be improved.

2 Background and Related Work

A lot of economy visualizations have been made prior to the one created in this project. Usually economics are displayed in static graphs or tables with a lot of numbers. One project that was made using the same data set as in this project, was looking at *Bad & Good Debts*. The project was about analyzing how many people had *bad loans* such as second mortgage or home equity loans, while comparing to people with *good loans* which was defined as all other loans and the reason for it. The user is met with lots of graphs and not a good explanation of how to use the tool [3].

What this economy visualization lacks is the functionality to interact with the data in a simple and intuitive way. Even if the product of this project displays a lot of numbers as well, the way the interaction works let the user dig in and learn about economics by interacting with each state [4].

3 Data

The data set that has been used in this project is called *Insightful & Vast USA Statistics* which contains 39000 different rows with 80 columns. The data set contains information about income, school degree, rent, debts, demographics and much more. The data has been recorded over a span of 5 years between 2012-2016 by an American company called *U.S. Census* [3]. All the data points consists of information about people who lives in a specific geographic location, that is known as a *place* in the data set. These places are areas in cities, which then lies in different states in the country. The dividing of the data makes it easy to visualize each data point on a map.

The data points different datums have been calculated by taking either the average, the standard deviation or the median out of all the people who have been recorded for the data in that area.

- *Viktor Sandberg student at Linköping University, Sweden, e-mail: viksa378@student.liu.se.*

4 Method

The application that was made is a visualization tool with focus on an having a interactive map. The map makes it possible for the user to navigate through the country by pressing different states. When doing this, the user is given information about that specific state.

4.1 Interaction and UI

To make the application, information from the data set was read. The data contained geographical location for each data point, to show them, a vector graphic map of the country was drawn with each points plotted onto it. The map was made in two layers: one that shows all the states and one layer that has all the counties. If the user clicks on a state, a zoom transition is made on that position. The fill color of the state disappears and the county borders and the data points become visible. This way, the map doesn't become too cluttered with points and the user can focus on one state at a time.

Other than the interaction with the map, the user is also given information when a state has been clicked. During the zoom transition, a menu appear on the right side with information such as population, dept percentage and yearly income in the state. It was also made possible that hovering over data points makes a tooltip appear with the information about that point.

To make it easier to compare the values of each state and all the data points different methods have been used. Each state was given a different color from a gradient, where a saturated blue color correspond to a higher yearly income and vice versa. By doing this, the user gets a quick overview of the country's economy and understand intuitively the meaning of the color without reading the label [5].

4.2 Graphs

Other than having the information text in the menus, different graphs were drawn to make the menu less cluttered with text and to give the user a quicker comparison between each state. There were 3 plots in total that was used: A bar chart, a pie chart and a scatter plot.

The purpose of the bar chart was to give a quick comparison of the selected state towards the others by coloring the selected state's bar in a yellow color. The height of each bar corresponded to the average yearly income of that state, and each state's bar was sorted in a descending order to make the state's average income clear comparing to the others.

The second plot that was created was a pie chart to display how many women and men lived in the selected state while comparing how many percent of each sex had a high school degree. A pie chart was useful in this situation because it was a way to display 4 variables at the same time.

The scatter plot was the last graph the was added with the purpose of giving the user a consistent way to compare all the data points in the selected state towards each other. The Y-axis in the plot corresponded to the percentage of the population that was having a high school degree, while the X-axis corresponded to the data-points average yearly income. This way it was easy to spot outliers and other abnormalities

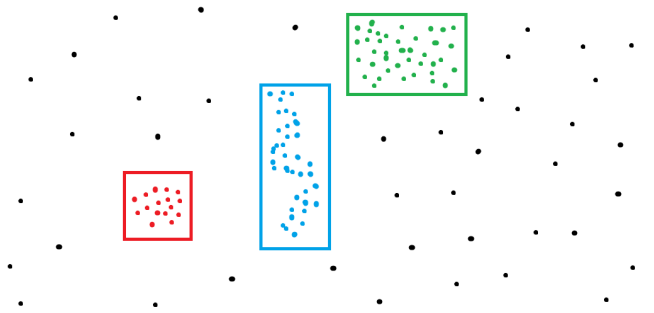

Fig. 1. Figure showing a set of example data points where the colored points correspond to a cluster

in the values. To make the outliers even more apparent, a data mining technique was used which will be explained in chapter 4.3.

4.3 Data mining

The algorithm used for data mining is called DBscan, *Density-based spatial clustering of applications with noise*, which is an algorithm that finds arbitrary shapes or clusters in the provided data. The reason why this data mining algorithm was used was manly because of it's quick run time of O($nlog(n)$). Because the algorithm was needed to run every time a new state was clicked, it needed to run fast. The algorithm works by grouping together points that are lying a chosen distance away from each other and marking them as a cluster. Then, leaving the other points where the closest neighbours are too far away as outliers (noise). Figure 1 shows the result how a cluster could have been achieved where an arbitrary choise of parameters have been used. The dots are the data points, and the colored dots within the rectangles are the formed clusters, the black dots are the outliers [2].

The algorithm works in the following way:

- Consider a data set with n-dimensions, in this example only 2 dimensions are used.
- From every point in the data set, a circle with a radius of *epsilon* is drawn. All the data points within this circle are then being counted.
- If the number of points within the circle is equal or exceeds *MinPts*, the centre point and the other points within radius *epsilon* are marked as being part of a cluster.
- The cluster is then expanded by recursively comparing the other points within the radius by doing the same operation as above, this is done on all points except on the centre point.
- When the recursion is done and if the cluster contains less points than *MinPts*, the cluster is being ignored and the comparison continues on the next data point.

4.3.1 Data mining in the application

The data mining technique was used in the application to find clusters in the scatter plot. The algorithm found the clusters in the data and then each cluster was dyed in a unique color. This did not only help distinguish the similarities of the data, but also helped showing all the data that did not follow any pattern. Since the data that was being analyzed had a large number on the X-axis and percentage [0,1] on the Y-axis, finding a value for *epsilon* and *MinPts* was not trivial. Therefore to get a value that was suitable for the data in all states, a lot of trail needed to be done to find a value.

5 Implementation

The visualization tool made in this project was created to be a web application with the focus of being used on laptops or desktops. The coding languages used were HTML, CSS and JavaScript. The library for handling all the visualization in the project was *D3.js*, and was chosen because of prior knowledge as well as the large amount of documentation that can be found online. The map was created using the a GeoJSON map of United States as well as a JavaScript library *TopoJSON* which eliminates redundancy by stitching together the GeoJSON paths, into one large map using arcs [1].

To have the application run better on average computing laptops, the data points that were plotted onto the map was scrubbed from 39000 points down to 10000. This made the application's transition act as they were intended and without stutter.

6 Results

The final product is an economy visualization application which has the starting window that can be seen in Figure 2. The figure shows the intractable map when nothing has been pressed, and a label in the top centre explaining that a saturated color of a state correspond to a high yearly ($) income, while a close to white color means a low yearly income. The interface was made not be cluttered for the user, but rather minimalistic with not too many graphical elements.

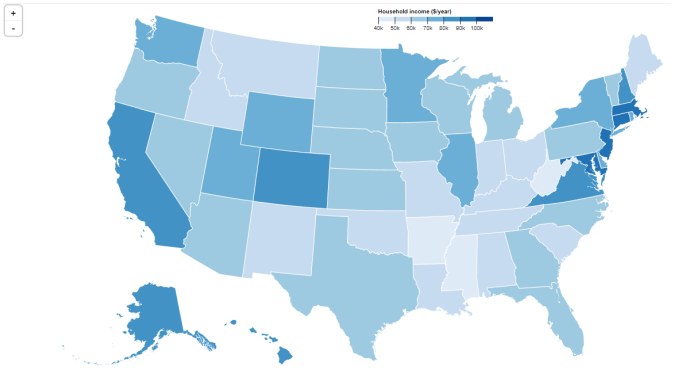

Fig. 2. The start page of the application: The interactive TopoJson map.

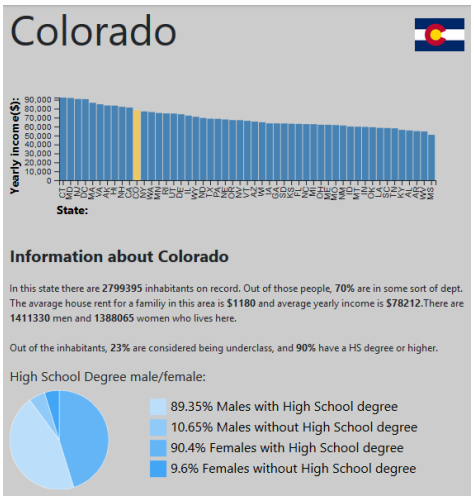

Fig. 3. Figure showing the state information menu when a state has been clicked.

The interaction with the application is centered around the map. The user can both hover and click on states to get more information. If a state has been clicked the menu shown in Figure 3 appears on the right side of the window. This menu contains information such as how many inhabitants there are in the state, how many percent that are in

some sort of a debt, the average rent and income. The user can also see the how wealthy the state is compared to the others and see how many percent are male or female. Also, when a state has been clicked, the geographical location of all major data points in that state are being visible on the map, as seen in Figure 4. The data points can also be hovered over to show additional information such as the area's name, yearly income and population. This is shown on a gray tooltip to the left in the window, which can also be seen in Figure 4.

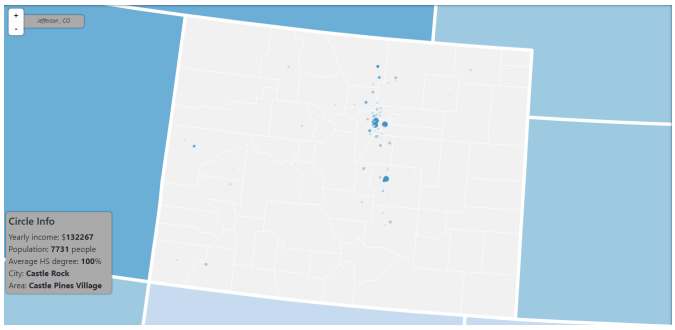

Fig. 4. Zoomed in map of a state, showing a tooltip of a hovered data item.

Figure 5 show the scatter plot where the data mining algorithm has colored all the clusters with different colors. The outlier can be seen as black dots on the graph. The user can hover over these dots to see more information about them on the same tooltip as in Figure 4. When a dot is hovered, it's geographical location is also shown by drawing a red circle around it's location on the map.

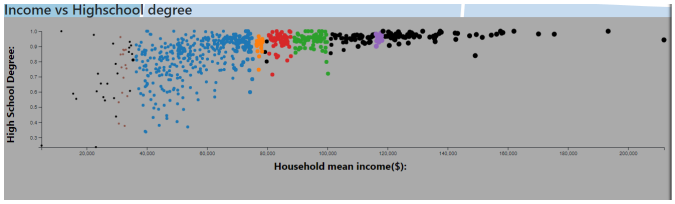

Fig. 5. Scatter plot with Y-axis as (%) High School Degree and X-axis as average yearly ($) income. Showing clusters in colors and noise/outliers as black dots

The loading time of the application is 2.6 seconds which is, considering that there 38000 data points are being drawn onto the map, a good result. However, when navigating the map with that amount of data points, the stutter becomes a problem. Therefore the amount of data points were reduced to only show data points with population over 4000 people.

One limitation of the application is that it is not trivial for the user that one can click onto the map to get more information about each state. In the maps original state, there are only a map of the country with no indication of intractability.

Another limitation is all the calculations that needs to be done every time a state is clicked. Even though the data points on the map have been drawn permanently, the graphs are being recalculated every click which slows down the application considerably.

7 Conclusions and Future Work

The conclusion that can be drawn from the results in this project is that even if the application have some limitations, the product is still something that have intuitive elements that makes for a good visualizing tool for economy data. The chosen color scheme of the plots help the user to get a quick understanding at the same time as it is aesthetically pleasing. The information that can be read about the states, and all the areas, can help the user to get a deeper understanding about the economic state in those areas.

Even if there are things that turned out good in the application, there are things that could have turned out better. To begin with, the colors in the plots and graphs turned out to be decent, but the overall gray color that was used in the menus, makes the UI look dull and boring. These colors could have been changed to something more uplifting and vibrant. Another thing that could have been done better are two implementation choices that was done regarding the scatter plot. Every time a new state is being loaded, all the data is being reloaded. This means that the data mining algorithm DBscan is executed many more times than necessary. This is one reason why the application takes a long time to transition between states, and it could have been implemented diffidently by storing the result of the DBscan so it only needs to be run when the application is loaded. Another thing that could have done differently is that even though the DBscan algorithm identifies clusters well and dye them in vibrant colors, the graph can get quite cluttered because of many points. This could have been solved by only display the centre point of each cluster, and then if hovering over that centre points the rest of the cluster is shown.

For future work of this application, the main focus would be to improve the run time. A code refactoring would therefore be a priority. It would also be interesting if the data set would contain information of how much the values would change over time. By having this data, it would be possible to implement a time line and see if there are any major changes over the states. Another thing that would be interesting would be to analyze all the unused data in the data set. More functionality could be added to the application and by looking and interacting with well thought out graphs over the unused data, the user could probably learn even more about the economy in United Sates.

References

[1] M. Bostock. Topojson. `https://github.com/topojson/topojson`, 2016-10-08.

[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[3] A. Geiger. Insightful & Vast USA Statistics. `https://www.kaggle.com/goldenoakresearch/us-acs-mortgage-equity-loans-rent-statistics`, 2018.

[4] R. Hijn-Neira and J. . Velzquez-Iturbide. How to improve assessment of learning and performance through interactive visualization. In *2008 Eighth IEEE International Conference on Advanced Learning Technologies*, pages 472–476, July 2008.

[5] K. Misue and H. Kitajima. Design tool of color schemes on the cielab space. In *2016 20th International Conference Information Visualisation (IV)*, pages 33–38, July 2016.