The College ScoreCard Data Visualization

~ Final Project ~

ECS 163 Winter Qtr 2016
Janak Jobanputra
Michaela Gobron

Data Description

The dataset we chose for our final project is a grouping of information on colleges throughout the United States over a span of 20 years. It is termed the "College ScoreCard" and has separate CSV files for the years 1996 to 2013, with each broken down into the following:

- ID that represents each school, and the location of the college
- The corresponding college name, the main and branch campuses
- Degree types offered
- Programs offered by the degree type
- Out of each program offered at the college, the percentage of degrees awarded
- In terms of admissions, the acceptance rate based on the number of applicants
- The SAT and ACT scores acquired by those admitted to the school
- Information about the student body (number of undergraduates, that number split into race/ethnicity, part-time status, students over 25 years of age, etcetera)
- The cost of attendance, tuition, fees, and the average net price broken up by income level
- Percentages of aid given to different students, the cumulative median debt accumulated for the different subgroups of students, and the typical monthly loan payments for the graduates coming out of the school
- The repayment rate of students who received loans from above
- Completion rates for students enrolled in the college, for first time and full time students
- And averages and median earnings of graduates (based on the different student subgroups) and for the former students that now earn over $25,000 annually

This follows the requirement of having four or more attributes; in fact, this dataset supersedes this expectation, allowing for us to look into other visualizations.

Initial Visualization Proposition

This data set is relevant because one can simply look at an individual CSV file and see how the colleges are performing in relation to the others for any given year. This could help potential college students who are in the midst of looking where they would like to spend the next four (or more) years of their lives. We were hoping to create some assistance to those that were unsure where they want to go and are only looking for schools that perform in specific areas, whether that be in academics, post college earnings, or just had a good completion rate. If students knew what kind of schools they wanted to go to, then this visualization could help them pinpoint where in the United States those schools are located. In addition to the location, the states would also display other statistics about the colleges within their borders.

We initially proposed to visualize the data on a map of the United States, with a heat map to show where the different locations on the map best fit the different criteria the user had chosen from in a drop down or scroll bar on the side. Upon clicking each state, we wanted to bring up a screen based only on viewing the data pertaining to only that state. We even wanted those individual maps to be color coded based on the criteria specified by the user. Additionally, on the aggregate US map, we wanted to try and implement a "chord-like" diagram to show the different migrations people took post graduation. Not to mention, we wanted to implement some form of animation to show how the data changed from year to year over the twenty year span.

That way, potential students can see if schools are improving or deteriorating in the different aspects available as time elapses.

Visualization Changes during Implementation

In the process of developing the implementation, we realized that the dataset, even taken one year at a time, is extremely huge, and will take a long time to load into the program (not to mention being difficult to submit). For this reason, we decided to put all the data in global variables during the setup. This way, the performance of the visualization would be buttery smooth, after giving it a couple seconds to load. We then extended this ideology to other important variables as well, since keeping them global allowed for much easier access, instead of having to deal with addresses and/or value passing through function declarations.

Another change we decided to make during implementation is to keep the draw function as minimal as possible. During previous projects, both of us had trouble finding code when we were trying to modify the visualizations. For this reason, we wanted to use the draw function as a 'controller' function, which, based on some simple logic, calls other functions. These other functions would actually house all of the implementation and would be aptly named to essentially provide 'live' commentary of what the function does.

From previous experience, we knew that a heavy part of our project would be designing interactions based on if the mouse was over a certain part of the visualization. This used to involve putting lengthy checkpoints scattered all across the code. However, this time around we designed a couple utility functions to deal with exactly this problem. The two that we used mainly were mouseOver(x,y,w,h) and checkFileExists(filePath).

One of the main changes from our original plans for this visualization is the main view that we implemented. We initially planned to use the map as a starting point for the visualization. However, after analyzing the capabilities of the Processing application and our timeline to finish the visualization, we realized that interfacing with a map would be rather difficult seeing as a simple function (such as mousing over a state) would require us to calculate the shape of the state in standard geometric shapes, pixel by pixel. Also, trying to scale the latitude and longitude given in the dataset to our US map size became quite difficult when plotting where the college was located on the map. Actual latitude and longitude lines are round (because the Earth is round) so a lot of the college positions became warped and not in the right place. This became painstakingly time consuming to debug, and not the most rewarding. Although there are API's available for this sort of function, we decided to just alter the initial view from a map to a grid of alphabetized shapes. This is much easier to do since we would not have to deal with states such as Alaska or Hawaii that are off the map, or worry about tiny states like Rhode Island or Connecticut.

Going into this project, we knew that 2 dimensional charts would be extremely helpful in demonstrating the data. However, in the past, we had experience with manually creating our own 2D graphs in processing. After seeing the number of dimension that could be displayed per state, we knew that creating the graphs manually would not be a feasible option. Therefore, we used a library of functions and classes we found online named 'gicentre'. This library is included in the zip file, in the main folder. To include the library in processing so that our code runs without a

hitch, it is important to include this library. Please open our main.pde file, find the Sketch option in the menu bar, and click “Add file”. After that, a dialog box will open up with the file system interface. Please locate the library file labelled “gicentreUtils.jar” and double click it. Figure 0 (haha coding joke) is a screenshot of the location of the add library functionality within processing. If this does not work, and there is still an error when running, choose the Sketch > Import Library > Add Library option, search for gicentre, and add that library.

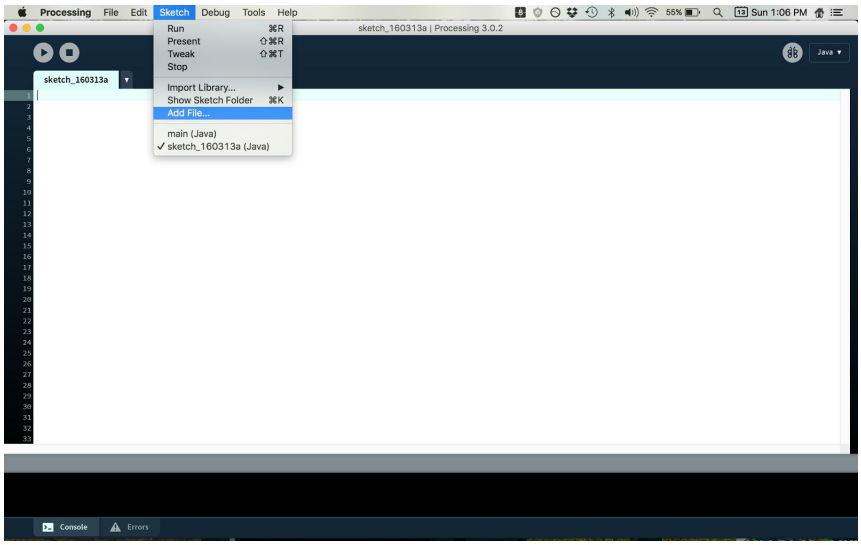

Figure 0

Visualization Description

The view opened when the Processing application is run is the only window that will be available to the user. The reason we chose to fit everything on one page instead of linking through the different pages is because it is much easier for the user to acclimate themselves to the options of one window, instead of expecting them to understand multiple interfaces at once.

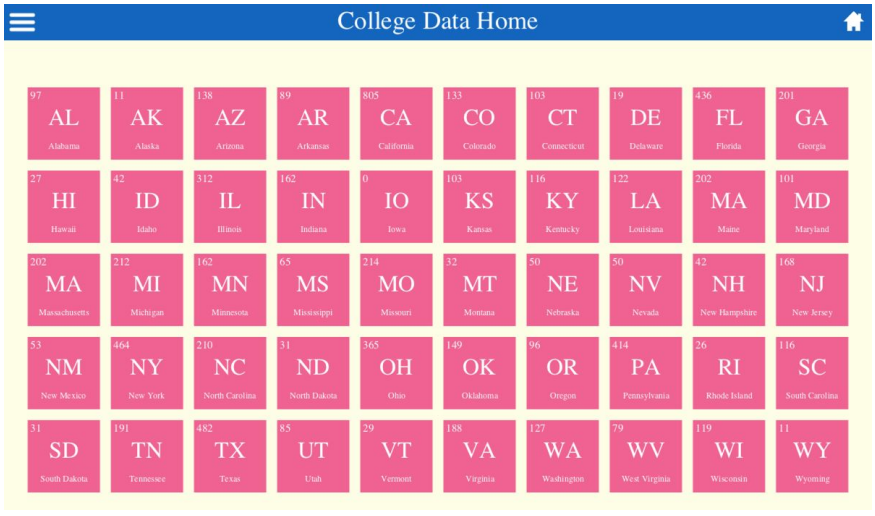

Figure 1

When the visualization opens, the first view is a grid of pink squares with a toolbar at the top. The toolbar will be discussed later. The pink squares represent each state, and they are labelled with the abbreviation in the middle of each square in big font. Under it is the full name of each state, and at the top left corner is the number of colleges reported in each state. Hovering over a state’s square will make it turn green, and clicking on the square will lead you to the next view, further information about the colleges in the state. A screenshot of this view is shown in figure 1.

Once inside this view, all previous views get erased, and the currently selected state's college information begins to display. On the left is a large map depicting the state and where its capital and large cities are located. To the right of this is a bar graph that shows information about the colleges in the state. Because some states had huge amounts of colleges reported, and others very few, we only showed 65 colleges maximum per state. (Given more time, we would implement how to scroll through the colleges. This would allow you to see the next 65 colleges if there were more than 65.) The bar graph has transposed axes compared to a normal bar graph, because it is much easier to read the college names this way, and to easily see which college has the highest value reported. If a college had no value reported, it is shown on the graph as a 0. The buttons to the right of this chart are labelled with the different metrics that we collected from the data. When a button is clicked, the chart is updated to show the values of each college for that metric. For example, if annual cost is clicked, then the graph changes to show the annual cost of each college, as shown in Figure 2. Additionally, between the graph and the buttons is a list of numbers. These numbers correspond to the colleges (as shown on the labels of the bar graph) in that state. When the user hovers over a number, a horizontal line appears to show the number that is being hovered over. When clicked, the number will lead to more information about that particular college.

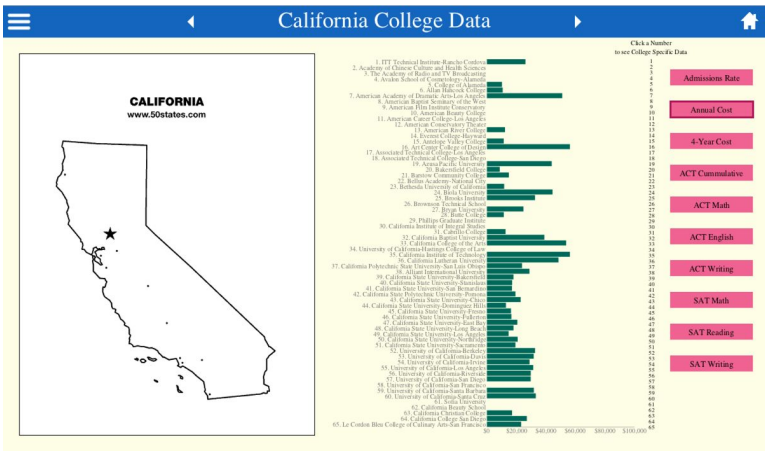

Figure 2

Upon clicking one of the numbers to the right of the bar chart on the state view, the user is directed to a view directed towards displaying the different statistics pertaining to that college specifically. At the top of the view (shown in Figure 3), the user can see which college they are viewing and what number they clicked on the previous view. On the left side, there are two pie charts to display the breakdown of the different races and what the parents' educational background is out of the entire student population. These metrics were chosen because they were some of the only ones that summed to 100% when added together; therefore, they could be compared with each other. To the right of each pie chart, there is a legend and a breakdown for the different slices to a more quantitative view. To the right of the view, there is a column of

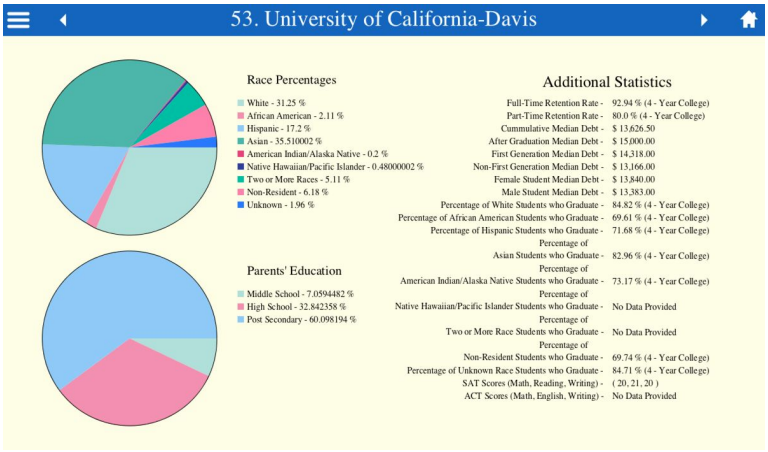

Figure 3

statistics. It is here where the user can view numerical data, which are updated as the user sifts through the college data. The retention rates and percentages are based on four-year and less-than-four-year colleges (which are shown in parentheses based on the length of the institution). There are also lines for the different percentages of the different races that graduated from their own population on campus. These numbers, when summed together, does not equal 100% so it could not be used visually in a pie chart. Lastly, there are SAT and ACT median scores the college admitted. After viewing all these statistics, the user can determine if they fit the requirements and demographics to gain a better sense of whether or not that college is a good fit for them. They can even see how much they should plan on spending for tuition, as well as if their tests scores line up to what the majority of students admitted received.

The toolbar is the one part of the program that stays consistent throughout all of the views. This is a conscious decision and allows for a sense of continuity through the program. The left-most icon (nicknamed the 'hamburger' icon) on the blue toolbar at the top is a symbol for menu. This symbol is often used by applications that use material design concepts to design the interface. When clicked, this toolbar will provide a menu of navigation for the entire interface, as shown in Figure 4. The different lines of text in the menu show the different views that we have gone through. Clicking on each one of these will bring you back to that view. To exit the menu, press the 'hamburger' icon again. At the top right of the toolbar is a home icon, that will bring you back to the home view with the grid of all the states, no matter where you are in the visualization.

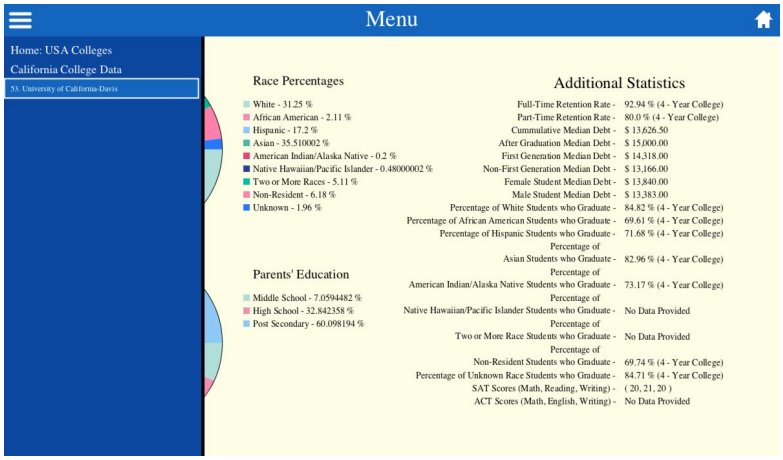

Figure 4

Task Delegation

We divided up the work extremely fairly. Janak is very creative and has an eye for aesthetics, while Michaela is better at the implementation. However, we both worked on the appearance and underlying code, feeding off each other for ideas and changes to make to improve the performance.

Janak is more comfortable with the data manipulation, parsing the files in search for the data points we are most interested in. He is also the most interested in how those data points will be displayed on the screen. Pulling from his graphic design experience, he is able to make the user interface extremely inviting and familiar. He also emphasizes that the user should be able to navigate themselves with extremely minimal instructions on screen. Most of the idea formation stems from Janak's creativity, since he has a knack for seeing trends in the data and thinking of a practical way to visualize them.

Michaela is more comfortable with the implementation, grabbing the extracted data and figuring out a way to format it and display it in the visualization. Once Janak parses the data, Michaela can take it and run with it to help create the graphs and charts and other visualizations Janak brainstormed from the beginning. Michaela's strengths lie in developing algorithms to filter and preprocess the data before it is visualized. Then, after the chart is close to working, Janak will move and scale it to fit in with his vision.

Lastly, troubleshooting will be tackled by both partners. We both are more than capable of completing all aspects of the project, so if something is not working, we are able to switch in and out of our predescribed roles to finish the project. This is why we are a good match, because we are both well-rounded and can help eachother out in the hopes of finishing the project.

Sites

Data - https://collegescorecard.ed.gov/data/

State Pics - http://www.50states.com/maps/

Libraries

Bar Chart - http://www.gicentre.net/utils/chart

Pie Chart - https://processing.org/examples/piechart.html