Most Popular Baby Names by Sex and Mother's Ethnic Group, 2011, New York City

California State University, Long Beach.
1250 Bellflower Blvd, Long Beach, CA 90840

Arun Kumar Shanmuga Velayutham, Drew Stiles, Sam Som

Abstract. In the effective presentation of observable phenomena a visualization must identify and address a distinct purpose in its application. While external parameters such as audience and utility will generally denote a distinct structure to a visualization, such parameters may not always be prevalent in the formation stages of the design. Without stringent necessitation an effective visualization must provide an environment for efficient exploration of the underlying dataset while allowing the user to discover and answer questions relevant to themselves. While it is impossible to address the infinite permutation of all questions one may have regarding the dataset, a generalized goal in association with supporting perspectives to meet this goal and its variations allows unforeseen, but related, questions to be answered from a limited but flexible set of tools provided by the visualization. This paper will seek to exhibit this flexibility and efficiency in addressing the popularity of New York baby names in 2011, while providing mechanisms for related questions and trends to be identified and understood.

Keywords: Human-Computer Interaction, Animation, Graphics, Baby Names, New York, 2011.

Introduction

The visualization presented here is an application of the dataset relating to the baby names of those born in New York during the year 2011. The visualization aims to address questions relating to the base question of which name was most popular during this period. At its root the question of strict popularity is both simple and broad allowing the designers to narrow the scope of use cases for the visualization to those within the bounds of this root question. Though it is simple, the question can assume numerous forms that relate tangentially to this base question allowing for more complex questions to be discovered within the context of the original query. In providing a generalized framework from which to explore and analyze the given phenomena, many questions relating to popularity can be answered and verified to a level of granularity required by an arbitrary end user.

That being said, two primary use cases are addressed here with regard to this dataset. The first being exploration of given samples. The second being the raw quantitative analysis of these samples. With these two supporting perspectives the

designers of this visualization seek to answer one base question, though other related questions may see solutions from the methods used in addressing this primary question. In doing so it made sense to consider only those names most popular for the given year. With this set of most popular names, other trends related to popularity are visible by giving the user an environment in which they may identify and explore other questions relating to the base question. Most importantly this paper intends to highlight techniques used to allow for the greatest flexibility and widest application possible, without sacrificing the quality and effectiveness in its practical use.

Data Preprocessing

It is often important to preprocess the data for various reasons like statistical analysis, discarding the partial record (record with missing values), removing duplicates etc. We have worked on a dataset "Most Popular Baby Names by Sex and Mother's Ethnic Group, New York City" provided by "Data.gov" website. This dataset contained data variables such as 'year of birth', 'gender', 'ethnicity', 'name', 'count' and 'rank'.

There is a total of 5889 records in this data containing duplicates. It would have been difficult to manually remove the duplicate records given the size. Hence, we have used JAXB a Java API to read data from the XML format of this data. Duplicate checks and missing data checks have been performed on this data and stored back into XML format. After removing the duplicates it has been found that only 1963 unique records falling under 4 ethnic groups are present in this data with 1206 distinct names, ranked between 1 to 97 and 172 distinct count values.

Further, the data contains 307 "Asian and Pacific Islander" ethnic records, 384 "Black and Hispanic" ethnic records, 618 "Hispanic" ethnic records and 654 "White Non-Hispanic" ethnic records. By gender, total female record in 1004 and the total male record is 959.

Visualization Design & Implementation

A. Exploration View

The first view covered here, and shown in the visualization, is one whose sole purpose is allowing the user to better understand, and familiarize themselves with the data. The exploratory view consciously lacks any numeric quantities, delegating such use cases to the subsequently mentioned analysis view. Designed to be both aesthetically pleasing and consciously engaging, the strengths of this view lie in its

effort to create a virtual system, governed by logical parameters, but free to manipulation within these rules.

I. Goal

At its core, this view is interested in aiding, and not directing, the user's thought processes. Due to the diverse and unpredictable nature of a given individual's purpose when interacting with a visualization, an approach that seeks to aid unabated exploration seemed best in this context. Therefore a primary goal for this view was to limit restrictions as much as possible, while presenting some coherence that would allow the user to incorporate the new information into existing knowledge regarding the dataset. For instance, by providing a distinct set of filters to group data points into related structures in which all components share a common attribute, the resulting visualization can build upon the user's preexisting knowledge of the dataset as it was prior to the transformation. In this manner, the user can freely apply these transformations and observe their results as they relate to the initial context, giving a sort of recursive granularity for refining the information available at each step. For example, by grouping each data point by respective ethnicity, the resulting structures reveal relative measures for the samples within that structure, whereby an object's size denotes its popularity, the largest object in these subgroups would then denote the group's most popular name. One step further would be to apply a second transformation, analogous to a new stack frame atop this single level of granularity, giving a new presentation layer which answers a new sub-question. For example, giving color to the current ethnic arrangement now provides an answer to which names for which ethnicity for which gender are the most popular. As can be observed numerous related questions can be addressed in a similar manner giving a seemingly large number of permutations.

Given the diverse set of questions likely to be addressed by the exploration view under analysis, the environment and related tools provided by this visualization must allow for this degree of flexibility to be accomplished by the visualization in a manner that is fluid to the user. More importantly, since such a broad scope of sub-questions may arise from the base question here, it is important that the visualization serves to provoke thought in the most unobtrusive way possible so as to allow for successful and uninterrupted exploration of the data set. Such thought is given in the next section which addresses thought patterns likely to arise when viewing the data from this view, and how these thoughts may be complemented by the visualization.

II. User Interface Considerations

In order to provide a successful environment for exploration, careful thought was put into how exactly one may construct an environment meeting this stated goal. Precisely, the design choices made here sought to provide a concrete set of software parameters from which a malleable experience could be crafted. This idea is based upon those laws which govern our interactions with our physical environment, and how the absoluteness of these law gives us a sense of that which can and cannot be done. Once we have an understanding of these laws, logic may be used to shape our

existence into a way we see fit, assuming it falls within the nature of these laws. That is, by providing a logical set of rules for the visualization, primarily forces, spatial boundaries, and object properties, the end user would ultimately come to understand these rules and could manipulate the environment to achieve some end that would serve the user's exploration. That is this sort of system would be most intuitive and approachable for any user. Quoting psychologist Jean Piaget,

> "The universe is built up into an aggregate of permanent objects connected by causal relations that are independent of the subject and are placed in objective space and time."[1]

Therefore any virtual system built in the same manner, would ideally reflect a minimal learning curve, enabling a viewer to engage with and modify the information with ease. Allowing a user to achieve this state is integral to enabling rapid and unabated exploration that while confined to certain programmatic laws, is also open to customization in ways that allow the dataset to reshape itself into forms that more readily address whichever question the user is currently seeking to answer. Therefore any such visualization wishing to provide these sorts of mechanisms must be careful not to provide visual cues which may ultimately discourage such exploration, due to a perceived concreteness by the viewer. For example, the initial state of the system under observation here was crafted in a manner that provided little more than the base level of information gleaned from viewing a sorted table view of the same data. Seen below the standard load view is a base state which does little to suggest possible trends or metrics of interest, instead this base state was chosen as such to force the viewer to begin thinking about the forms they may wish to observe. This approach is advantageous in that it directs the user's state of mind to begin thinking about how they would like to interact with the data. For instance, had color and grouping assignments been predetermined, the viewer may neglect to construct an environment independent of these groupings, instead operating within the predefined, but not concrete, bounds of the visualization's initial state. This is important in the context of exploratory visualizations where it is near impossible to presume all states which may be of interest to an arbitrary viewer.

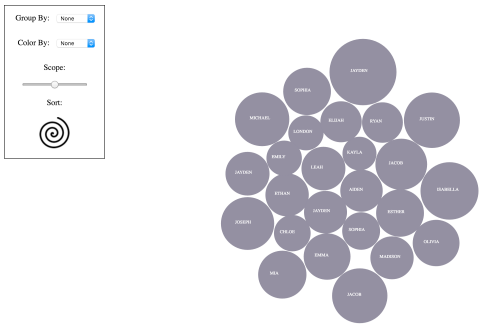


Figure 0. Default View

III. Design

With prior knowledge of both the goals and target behaviors in place, this section will seek to address concretely how the design decisions for the exploration view meet these criteria. Precisely the decision for the primary component, the "bubble chart", will be outlined along with its advantages and disadvantages. Other concerns, such as the apparent forces and accompanying interactions used in conjunction with each bubble object will provide a framework for considering the merit and overall success of these design choices as they relate to the visualization's stated goal.

a. The Base Unit

The exploration view is composed of numerous unique bubbles each of which relates to a single name in the dataset. To begin thinking about this choice of presentation it is necessary to ask why an individual bubble may serve as an accurate analogy to an abstract name pulled from the underlying dataset. As a single entity the bubble provides a container whose physical attributes may be manipulated in accordance with related metrics for each data sample. More importantly, each bubble, provides an equidistant container that allows labeling to be both direct, placed within the container itself, and readable, using the padding of the container's bounds, associating each name with a definite object. Considering the dataset here deals primarily with first names, labeling and its effectiveness are critical in the overall success of the visualization.

While effective labeling was the highest priority issue considered when selecting the base units of the visualization. Other properties of the circular bubble made it appealing in the context of this design. Since the stated goal for this visualization was to reflect popularity among names, a single scalar metric could be isolated to reflect the primary property of interest. Within the dataset itself, this metric was given by a count attribute assigned to each data sample. The magnitude of this name denoted popularity directly therefore meaning that this scalar metric could be mapped to properties within the visualization in a straightforward way requiring little indirection to represent the underlying data. This was another advantage to the circular nature of the bubble chart, where size, in this case radius, of the object could directly reflect the count attribute of a given name. Since both are scalar properties there correspondence was almost one-to-one meaning that little calculation would be required to associate the two corresponding metrics. Taking this notion even further, it can be suggested that the bubble radius size provides an accurate abstraction over raw numeric measures of popularity. For example, larger bubbles in the visualization take more precedence in the mind of the viewer, and therefore have more importance in the environment as they very much do when considered with respect to the visualization's stated goal.

The last attribute mapped to each bubble is a dynamic color assignment denoting some user selected transformation. In doing so, very primitive property assignments could dynamically reflect attributes relating to each name without

requiring the overhead of a viewport context switch. This is due to the fact that while static attributes of each circle, name and size, are consistent across the dynamic color mapping, reassignments of color only shift and actively inspected attribute while maintaining a sort of object permanence[1] in the viewer's mind. This allows position and color to shift at the user's command without requiring the underlying data sample being represented be reconsidered by the viewer. In contrast, imagine if every time a color or position was reassigned all prior graphical objects were redrawn discontinuously requiring the user to rescan the entire visible set for the name of interest. By choosing to maintain continuity between these transformations, this sort of permanence defines an interaction environment that allows for transformations to occur without disrupting the working model present in the user's mind at the time of viewing.

b. Bubble Groups

As a whole, the overall layout in the visualization provides an extra dimension which can be used to map additional attributes from the dataset. This layout, in coordination with the forces dictating the layout were both selected in a manner that would ultimately contribute to a comprehensive and cohesive notion of environment for the visualization. Primarily the individual position and associated groupings were used to map related attributes, such as gender or ethnicity to all names which shared those attribute values in the data, and position onscreen. Again, as previously mentioned the notion of object permanence is important in that as individual units of the visualization are transformed between their relative groupings, the sense of precisely which names those units belong to is unwavering. This allows groupings to be dynamic functions of the user's actions, giving rise to an interactive feedback loop that ultimately aides in the exploration of the underlying dataset.

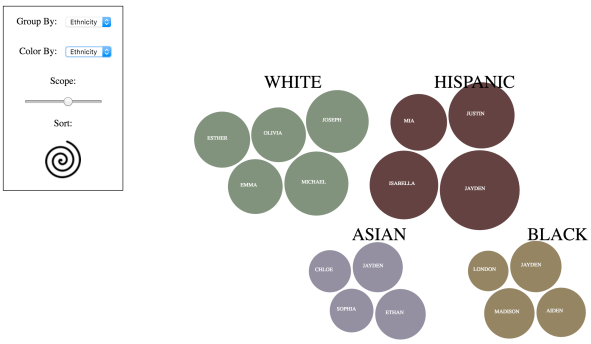


Figure 1. Group By Ethnicity

[1] The understanding that objects continue to exist even when they cannot be observed.

c. Environment

This freedom, and independence of each sample is rooted in the presentation of bubbles which may be transformed across regions without much confusion or a loss of accuracy in regard to the initial dataset. However, in order to impose some sense of relation and interactivity simulated forces are used to hold objects in a relative grouping in which the position is a reflection of some value. These forces are implemented in a manner that attempts to provide notions of environmental constants, which specify how a user may interact with the visualization. The primary force on display is a gravitational attraction to a region, where that region is currently housing all bubbles with some shared attribute. As both layout policy and force, the use of gravitational attraction links well with the object permanence for each bubble. That is, the transformations are subject to very real environmental parameters which influence how the graphical objects undergo change in accordance with user commands. This interaction gives both an aesthetic appeal to an exploring user but also provides a mechanism for enforcing change without requiring a context switch from the current frame of reference. This idea is critical in the environmental construction mentioned in the goals section of this design critique. Multiple forces and transformations can be applied to a data set all of which result in new perspectives of the same data, once the user has come to understand this interaction paradigm, combinations of these forces may be applied, allowing the user to construct new perspectives which operate within the concrete bounds of these forces.

d. Design Summary

Though containing three distinct levels of granularity for considering the dataset, a continuous frame of reference is applied to enable more rapid understanding of both the phenomena of interest, and the parameters of the visualization. Constructed in a bottom-up fashion the design paradigm on display in this exploratory view provides and efficient and accurate representation of name popularity through the use of labeled circles which appear both functionality and aesthetically pleasing to the end viewer. Working one level up on the layout of these base graphical units, the visualization offers supplemental degrees of information presentation that do not detract from those benefits of the base unit circles, while also conveying notions of dynamism and object permanence that give the user tools with which to manipulate and better understand the information they are viewing. Lastly a set of environmental parameters, primarily physical forces, unifies the circles and their perspective layouts in a way that allows the user to work within these parameters to enter a feedback loop for better understanding both the data, and the visualization's limitations, which in turn would ideally allow the user to logically formulate new questions which may be answered by the environment provided.

IV. Implementation

Implementing the aforementioned design for the exploration perspective was done exclusively in JavaScript. Though three primary libraries were used, D3, jQuery, and Underscore, D3 was the primary library for used in the generation and configuration of the display. jQuery was used for its event handling and element node access capabilities when D3 fell short of the desired access. The Underscore library was used in a sole instance where a data structure needed to be sorted and processed for unique values. Because of its prevalence in the code base, any mention to an external library for this section will refer to D3, as the roles played by the other libraries were minimal and outlined here in this paragraph.

a. Bubbles

All samples pulled from the underlying data, are mapped to a set of SVG g elements representing the root component of the bubbles in the visualization. Though transparent, most layout calculations are performed upon this root element, which in turn transforms all children accordingly. This root SVG g element is constructed dynamically as part of the enter selection from the original JavaScript object array containing all samples for the dataset. A small amount of preprocessing is done dynamically at initialization to calculate the respective radius for the SVG circle element representing the bubble graphic in the display. This bubble is appended to an SVG g element which is appended to the root g element in order to provide an encapsulation layer for the circle and text elements which compose the bubble and its label. The necessity for an "inner" g element comes from the fact that all D3 native transformations are applied to the g element resulting from the data join. Whereas in order to maintain compatibility with these transformations, all further transformations, primarily drag and drop, work with the inner g encapsulation element, which abstracts the D3 layout policy to an untouched layer, and allows layered transformations to work relative to this layer. This in turn provides a flexibility since group layout transformations are applied to the set of g elements created by the data join allowing further manipulation to work relative to this position. This allows a return to the group layout to be trivial since one must simply set the position inner g elements to zero in which case they return to their position within the root g element. This abstraction allows much of D3's internals to dictate complex layout assignments and frees the programmers to work relative to these layouts, defaulting back to thier positions when required.

The foundation for the explore view is built using dynamic groupings, which can be specified variably by the user. The implementation for these base groupings is delegated to the D3 library which provides the framework for such animations, requesting only the respective functions which will define portions of the desired animation. In order to achieve the desired grouping the sequence of commands directing this animation begins with a D3 force layout, whose behavior is given as a argument function to the force's tick method. This in turn sets the argument function to be called with each iteration of the force's lifespan for a given animation. At the

high-level the tick function is responsible for both assigning respective groupings for each individual element within the data join result set. Once those elements who share attributes are grouped, a second collision function is used to resolve the positions for each element in the data join group over the course of the animation. Since these tick and collision functions were written and provided by the D3, further details into their implementations are left here at the level of abstraction they were regarded with when programming this visualization. This level of abstraction required only that the attribute for grouping be provided along with a collision coefficient that altered the viscosity of the apparent, invisible, fluid in which the bubbles appear to be suspended. Ultimately the slow and gentle behavior exhibited in the final visualization was chosen using a smaller collision coefficient to these functions in order to give a sort of fluidity to the display that serves to aid the user in constructing a mental schema regarding the environment's nature.

b. Compare Panel

Perhaps the most rigorous portion of programming came in the form of the custom animation programmed as the compare panel environment. In order to distinguish this region of the display from the other, while providing a realistic analog to a containing body, forces and collision detection were programmed from scratch, since no other preexisting code could meet these needs. For brevity, the core of this environment is built into the following sequence. A user drags a circle from it's parent g element into the compare region. Upon releasing the mouse the circle is placed under the influence of a gravity force accelerating the object along the negative y-axis. Upon reaching either the bottom, one of the walls, or another circle, a resultant vector is calculated giving the object new velocity, while remaining subject to downward pull of gravity. An inelastic collision coefficient allows velocity to decrease by approximately one-third of its magnitude prior to impact therefore ensuring that the object will come to rest at some point. When rest is achieved the object is fixed to its position and the animation sequence ends. All animation drawing is handled by procedure outlined in the next section.

c. Animation Drawing

In order draw custom animations, a primitive frame by frame assignment was leveraged using the `setTimeout` function provided by the JavaScript language. This was achieved by initializing a frame variable to the value one prior to entering a while loop which continued as long as the magnitude of the object's velocity was greater than one. In turn the current state of the animation was encapsulated into a set of properties assigned within the SVG circle element undergoing acceleration. The vector forces acting on the circle would then update the velocity of the circle, and a new position was drawn to the screen for the corresponding frame. In order to ensure unique draw times the current frame number was multiplied by 60 milliseconds and incremented by one on each iteration of the bounding while loop. This way each update to the circle was drawn to the screen at some predetermined time governed by the length of the `setTimeout` wait. The number 60 milliseconds was chosen to give

a screen refresh rate of 60 frames per second, the fastest for most modern screen technologies.

```
function animateCollision(frame, dx, dy, circle, text, env) {

    circle.absX += parseFloat(dx);
    circle.absY += parseFloat(dy);

    setTimeout(function() {
        var current = getCurrentTranslation(circle);
        var x = parseFloat(current.x) + dx;
        var y = parseFloat(current.y) + dy;
        circle.setAttribute("transform", "translate(" + x + "," + y + ")");
        text.setAttribute("transform", "translate(" + x + "," + y + ")");
    }, frame * FRAMES_PER_SECOND);
}
```

Figure 2. Draw Time Assignment Function

B. Analysis View

I. Goal

Question to answer: What are the most popular names in each ethnicity group or a combination of ethnicity groups? Bar chart help the audience to visualize the most popular name in certain ethnicity group or a combination of groups. And the pie chart show the percentage of each ethnicity within the most popular names sample set. Both charts also allow users to compare names and see name-ethnicity relation among the most popular names.

II. Design

a. Popup vs always visible tooltips:

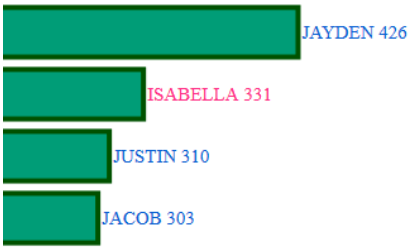


Figure 3. Always-visible tooltips

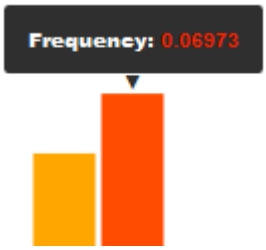


Figure 4. Popup tooltips

The first thing we have to decide is whether to use a popup or always-visible tooltips for our chart. The advantage of popup is that it help eliminate cluster. On the other hand, always-visible tooltips requires less interaction with the user. This come in handy when the user want to compare multiple bars; with always-visible tooltips users would be able to see all bars' tooltips immediately and able to compare them instantly. To accomplish this same task with popup, the user need to take a few extra steps. First they have to hover over a bar, then remember the information within the popup, then hover over another bar, and then compare the previously remembered information with this new information. If they want to compare more than 2 bars they have to repeat this process multiple time.

Decision: Our decision is to use always-visible tooltips with the bar chart and popup tooltips with the pie chart. The reason is that with always-visible tooltips the users can easily compare multiple names and see how much a certain name is more popular than other name. To avoid cluster of information, we limit the maximum number of bars in the chart to 40. Also, we use popup for pie chart because there is a maximum of 4 slices that can appear within the pie; this allows the users to easily compare information between the slice.

b. Standing vs sideway bars:

Figure 5. Bar chart

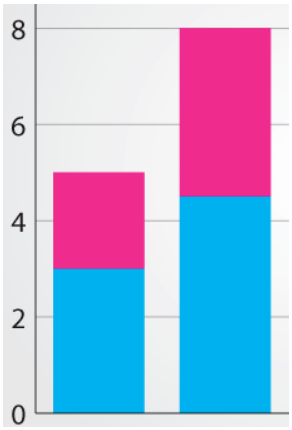


Figure 6. Column Chart

In this report, we will identify standing bar chart and sideway bar chart as column and bar chart, respectively. These 2 charts are identical in every way except for their orientations.

Decision: Our decision is to use the (sideway) bar chart. There are two reasons to support this decision. First, it is impossible to fit the , previously decided on, always-visible tooltips in the column chart without breaking the tooltips into two lines (one for name and one for count) or tilting it. Breaking tooltips into two lines does not solve the problem because some names are too long. Moreover, tilting the tooltips solve the fit problem, but now it is hard for the users to read the tooltips. Secondly,

bar chart is read from top to bottom. This top-to-bottom orientation imply ranking and hierarchy which this chart will be used for.

c. Colorization

Choosing colors for our visualization is important. There are 2 sets of colors that we need for our visualization, one for ethnicity and one for gender. For ethnicity we want to avoid color that can be associated with people's skin color. We also want color that have high contrast so that it is easy for users to read. What's more, we want the color set to be colorblind friendly if possible. For gender color, we have to decide between traditional color (blue for boy, and pink for girl) and an unbiased color. The advantage of traditional color is that people are familiar with it and this allow them to consume the information better. The pro of unbiased color is that it is neutral and we avoid offending people who are very sensitive about colors and its cultural connotation.

Decision: We get a set of colorblind-friendly color from colorbrew2.org to use for ethnicity and we will be using traditional color (pink and blue) for baby gender.

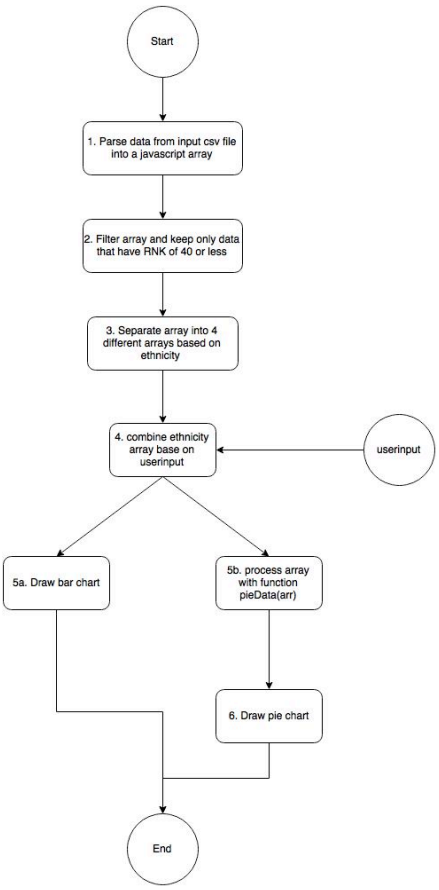
Figure 7. Color set for ethnicity

d. Interactive vs. Static

Now we have to decide whether we want our visualization to be interactive or static. The advantage of static visualization is that it does not distract user from the core visualization. The users can get straight to reading and digesting the information without having to interact or wait for any animation. However, interactive visualization is more engaging. Also with interactivity, we can condense more information into our visualization. For instance, we can let users view all the information or filter out information that they are not interested in.

Decision: we decide to make our visualization interactive. There will be two interactivity: ethnicity filtering and sample size selection.

- Ethnicity filtering: allow users to select one to four ethnicity to be display.
- Sample size selection: there are 4 sample size to choose from: top 10, 20, 30 and 40. We decide to stop on 40 because it got too cluster when go higher.



III. Implementation

1. CSV file that had been cleaned is passed to the system. Using 3D we parse the file into an array.
2. We then filter the array for any data that has attribute “RNK” (rank) 40 or less. The reason we chose 40 as our bound is because we only allow user to see the top 40 most popular names. By setting the bound at 40, we guarantee not to miss any popular names.
3. We now separate our array into 4 separate arrays based on ethnicity. (Note: this step is done along as step 2 to save machine cycle; we divide the steps here for readability)
4. With user input on what ethnic group or combination of ethnic groups that they want to explore, we combine those selected ethnicity arrays.
5. a. Our system can now use bars(arr) function to draw bar chart

   b. For pie chart, we need to process the array to count the number of each ethnicity within the selected data
6. Our system can now use pie(arr) function to draw pie chart.

Related Work

In general much of the inspiration behind the design of this visualization was derived from the DataHub visualization created as a part of the Harvard CS 171 course on information visualization[2]. Primarily the notion of providing a base level exploration view, with associated analytical views to probe deeper into the data were modeled after the work done in this visualization. The choice of circles, as covered previously was modeled after a D3 tutorial on the layout[3]. Combining these two influences gave the designers here the idea of providing a strict qualitative view for exploring the data in light and enjoyable manner that would address the largest portion of potential users. However in contrast to these initial influences, the work done here adds dimensions of interactivity not present in those influences. Primarily, the use of data structure space and attribute space as means of interaction in association with the graphical environment give the dynamic realism sought in creating the exploratory notions sought by this view. The data structure space here is denoted by the respective bubble grouping that may be configured in the display, the attribute space corresponds to the colored selections and bubble sizes used to convey metrics from the underlying data. In the spirit of exploration these interaction concepts are largely an aesthetic means designed to convey generic meaning while those use case which required more rigid analysis would be delegated to the analysis view in an attempt to provide broader application without sacrificing the visualizations approachability.

[2] http://teamdatahub.github.io
[3] http://bl.ocks.org/mbostock/4063269

References

1. Piaget, J.: The Construction of Reality in the Child. Psychology Press. Print. Cambridge, Abingdon, Oxon (1999)

Appendix

1. Software used:

A. D3 JavaScript library: We use the D3 library extensively for our visualization. We use it to draw svg components on screen. It also provide file processing functionality.
Version: 3.5.16

B. jQuery JavaScript library: We use this library as an add-on to our JavaScript. It helped to speed up our development process.
Version: 2.2.3

C. Firefox and Chrome browser: We use this for displaying and debugging the visualizations.
Version: 45.0.2(Firefox), 49.0.2623(Chrome)

D. Notepad ++: Our text editor
Version: 6.9.1

2. File format:

A. CSV (Comma-Separated Values): We decided to use this file format because D3 have an excellent parsing function for it.

B. XML (eXtensible Markup Language): We download this file format from the site because our member (Arun) have written a Java application to pre-process it.