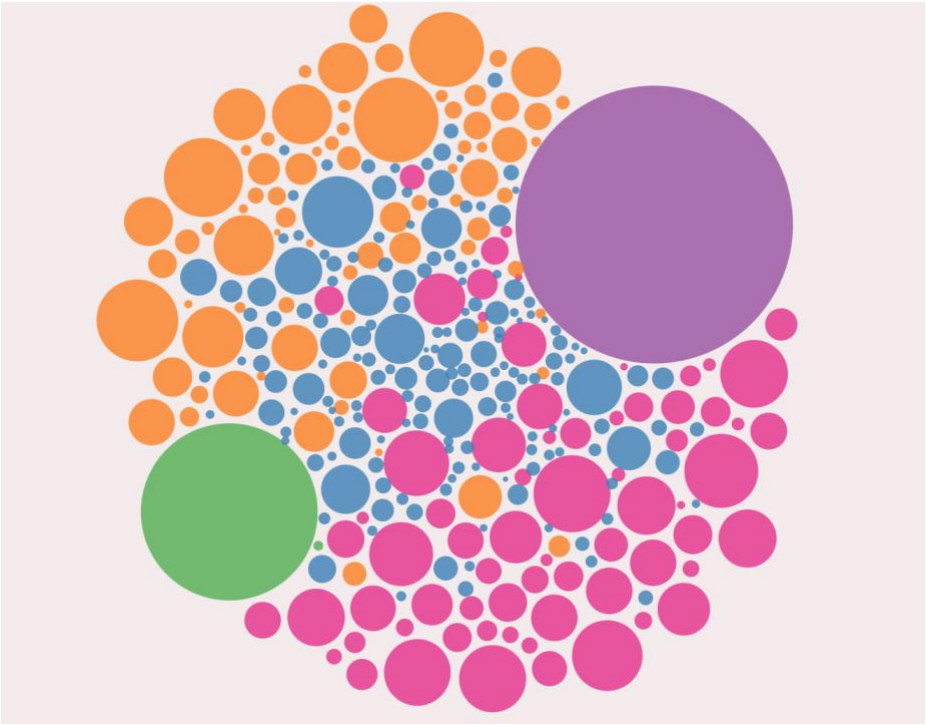
LANGUAGES SPOKEN IN THE U.S.

PROCESS BOOK

BY

JANAAN LAKE, ANDREAS MARTINSON & RACHEL BERGHOUT

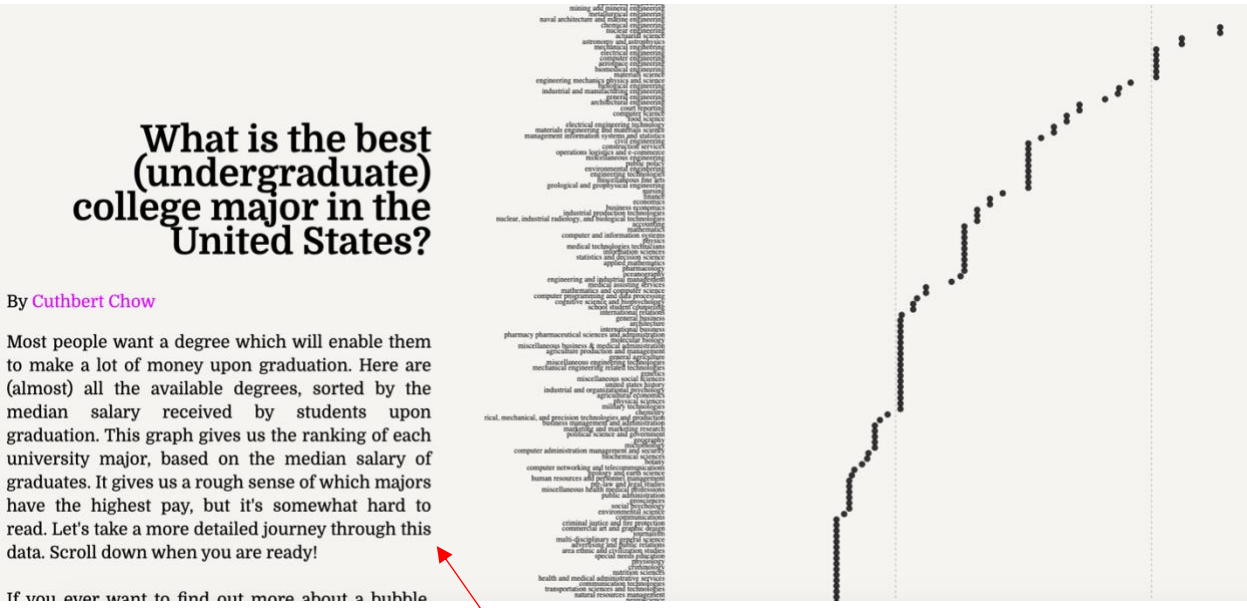


OVERVIEW & MOTIVATION

We were inspired by a tree-based visualization of the 100 most spoken languages around the world found here: https://www.visualcapitalist.com/100-most-spoken-languages/. This led us to decide on our topic of language. Free worldwide language datasets proved to be nearly impossible to find. So, we limited our focus to national datasets since there was ample information for the U.S. available.

RELATED WORK

When thinking about how to present the data, we really liked the storytelling/scrolling approach visualization found at this link: https://cuthchow.github.io/college-majors-visualisation. This influenced the method that we used to present our data. Below are a couple of screen shots of this website. The user scrolls through the left panel of the visualization as it tells a story and changing the view at certain points of the scrolling. The left side presents the storytelling aspect while the We wanted to incorporate a similar storytelling and scrolling aspect into our visualization.



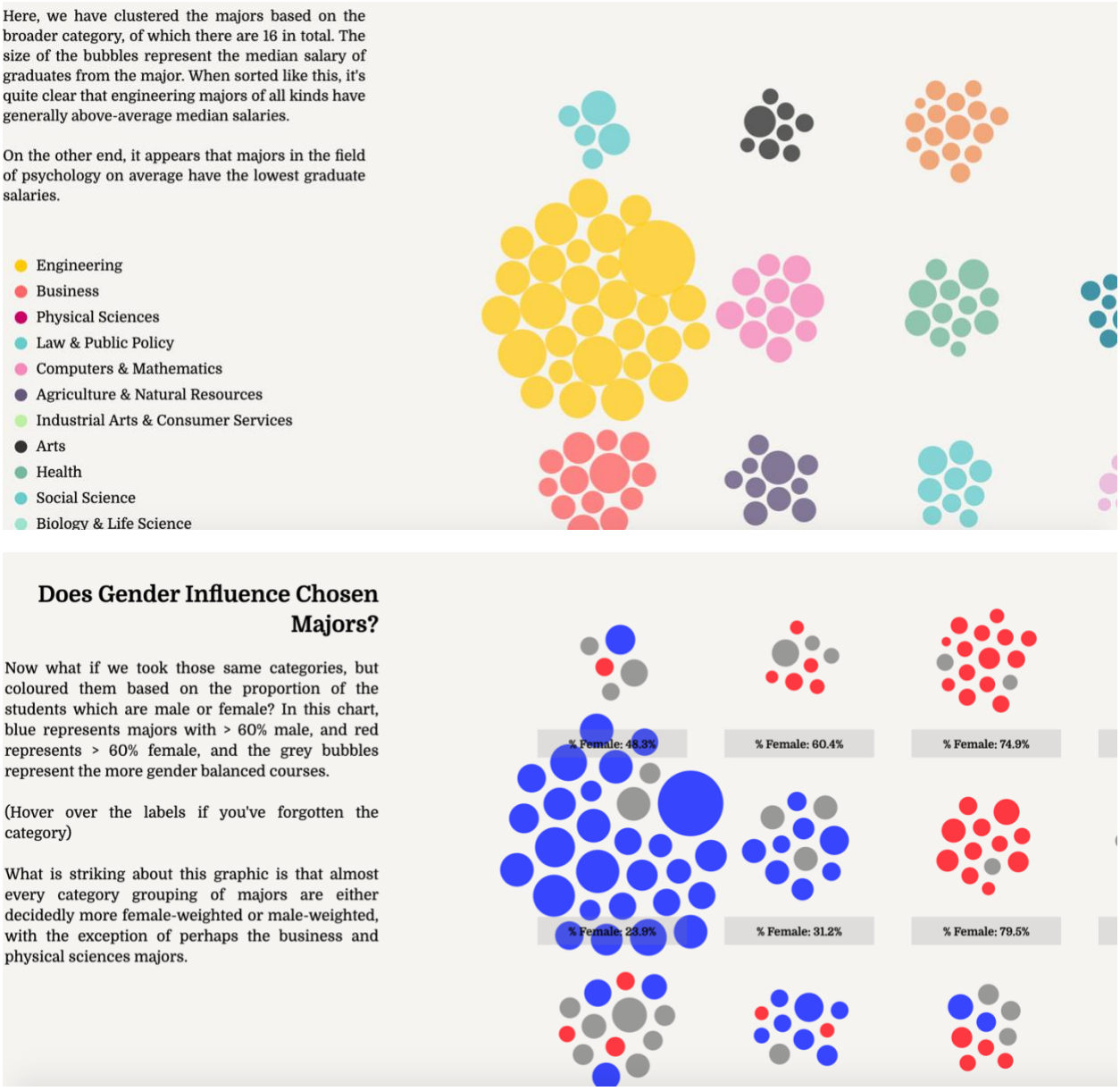


Figure 1: Examples of inspiration for our project

QUESTIONS

We had five primary questions that we wanted to answer in our visualization:

- Which languages are spoken within the U.S.?
- Where are these languages spoken within the U.S.?
- Which states have the most language diversity?
- Which foreign language speakers are the most fluent in English?

- How have the languages spoken changed over time?

We incorporated these questions into our storytelling aspect and created one graphic/view based on each of these questions.

DATA

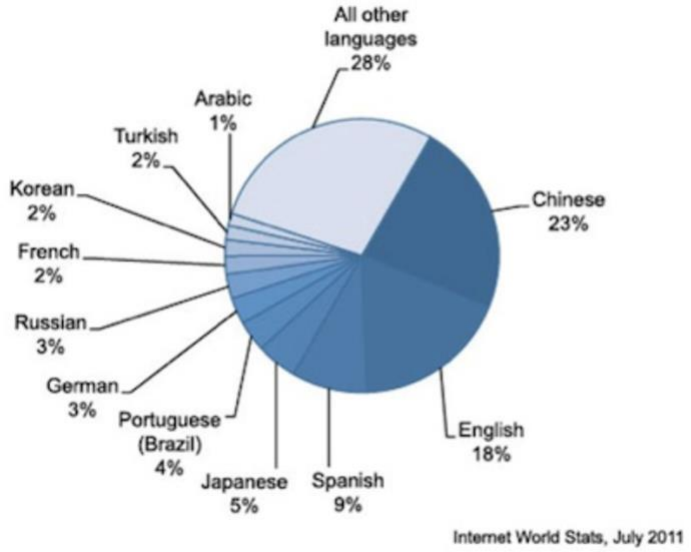
The dataset is from the U.S. Census Bureau and can be found at this link: https://www.census.gov/data/tables/2013/demo/2009-2013-lang-tabl. It represents data from 2009-2013. The data for the time comparison represents data from 1980 to 2010.

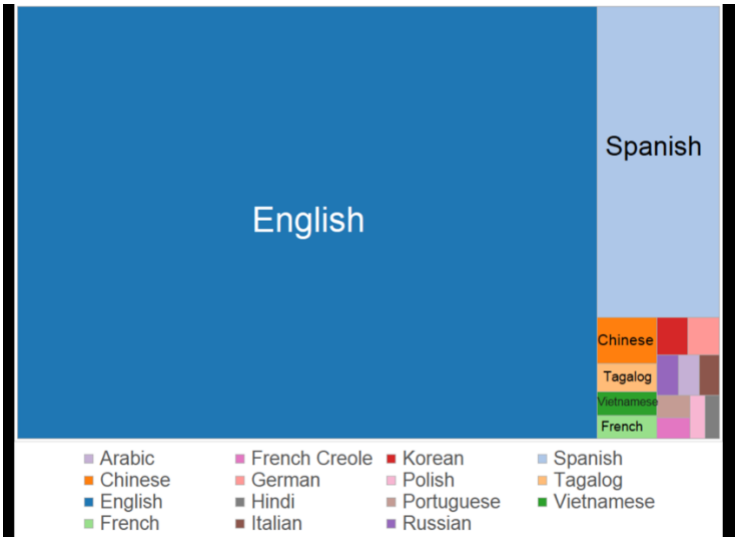
The data we received initially was not formatted in a way that worked well with our visualization. The data had groups, sub-groups, and the individual language rows with the totals and sub-totals as their own row. This would lead to problems with aggregating the data (since it would duplicate number values), and so we took the totals and sub-total rows and created new columns for them. We did the main portion of the data manipulation manually in excel since we decided it would take too long to complete all of it using a programming language and we were eager to start using d3 on the final dataset. We validated the new totals against the old sub- totals and totals to make sure we were not duplicating or deleting values and to help reduce human-error in manipulating the data. There were also some letters being used for null values. We replaced all of the letters with "NA"s so we had a common way to filter out null values in the data.

EXPLORATORY DATA

Our datasets had already been used in other visualizations, so we were able to use these graphics as a starting point to explore our data. Also, taking a cursory look at the data, we knew one of the challenges would be scaling the graphics since we had a large difference in scale (from <100 to 37 million) among our data points. We also wanted our visualizations to be interactive and interesting.

Other graphics of our data are included below:





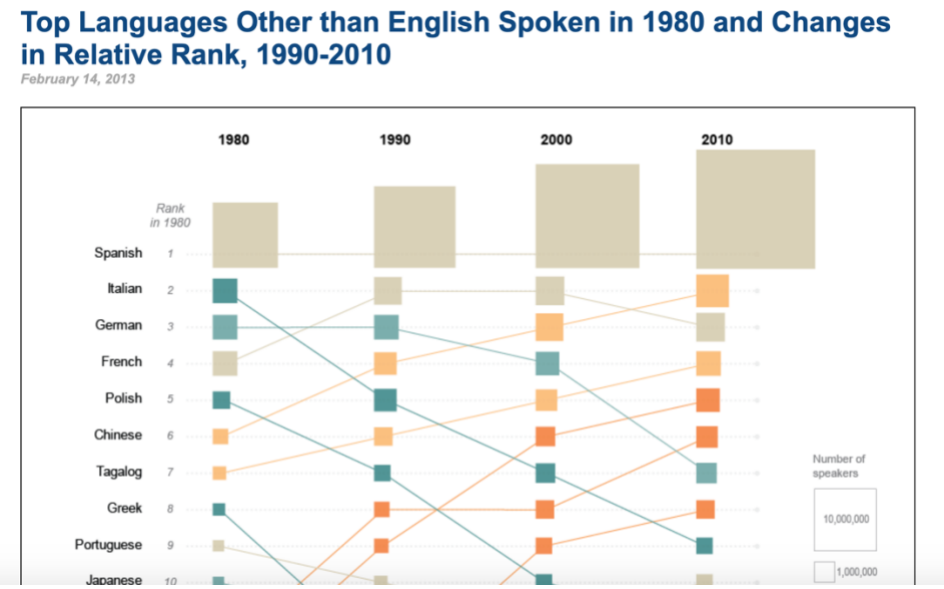


Figure 2: Examples of various visualizations using the same datasets

DESIGN EVOLUTION AND IMPLEMENTATION

Styling

Our aim for the styling was to make the interface clean and user-friendly. We also wanted to incorporate continuity among all the views by using the same color scheme for the same categories, legends, similar tooltip styling and fonts for all views. We aimed for a clean feel without a cluttered look and interactivity on all of the views.

Scrolling

Figuring out how to create a scrolling visualization was complicated initially but became easier to understand once we got through reading the articles and going through the code. The initial inspiration for building a scroller came from this medium post: https://towardsdatascience.com/how-i-created-an-interactive-scrolling-visualisation-with-d3-js-and-how-you-can-too-e116372e2c73. The visualization that was created can be found here: https://cuthchow.github.io/college-majors-visualisation/. We read the medium article first and then consulted Cuthbert's reference for building a scroller: https://vallandingham.me/scroller.html.

The challenges with scrolling was difficult to get right initially since we were trying to coordinate five different views. An initial struggle was getting simulations to start in the middle of the page instead of being created in the top left and then floating to the middle. Another struggle was getting the visualizations to disappear by changing the opacity, so we would have errors like the following:

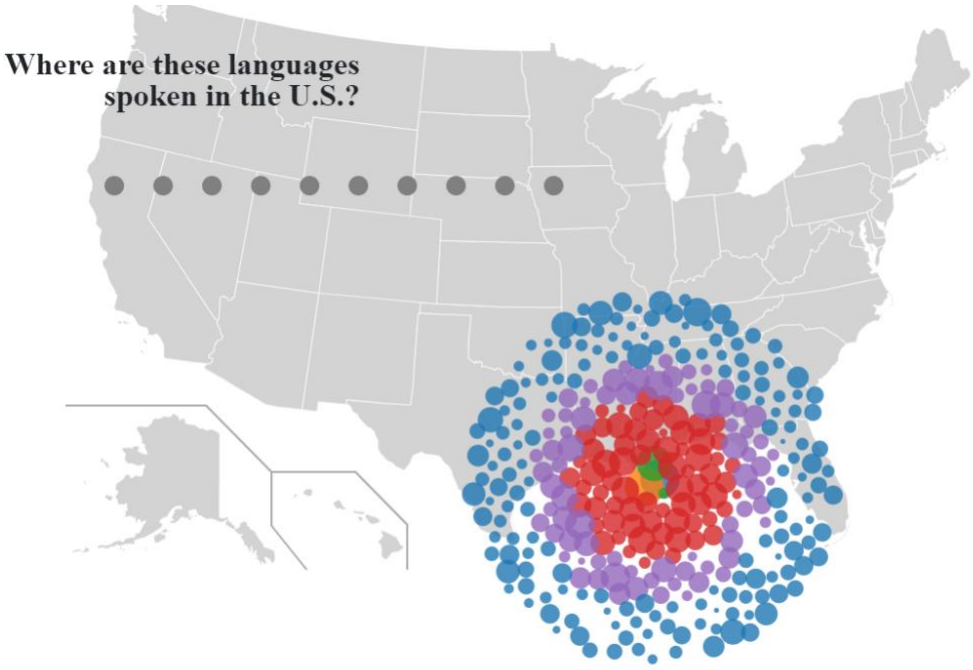


Figure 3: Example of scrolling errors

The basic architecture uses the same svg for all five different visualizations. Essentially, they are all rendered on top of each other. The opacity for each is adjusted based on where the user is in the scrolling. There were several difficulties with rendering tooltips when five different visualizations are on top of each other. To address these challenges we used the d3.raise() function to move the current visualization to be rendered last in the svg and we turned event listeners on and off for each visualization based on where the user was currently scrolling.

Transitions

Originally, we wanted transitions between all our visualizations to create a sense of continuity and for a creative aesthetic. We accomplished this for two of the view changes. The first view change from the cluster to the map shows the clustered circles coming from the first view into the top of the map view. These same circles are used in the interactivity in the map view. The second transition involved the change from the horizontal bar graph to the vertical bar view. We created a custom transition by moving the rectangles/bars from the horizontal view and showing them consolidating into the vertical bars of the same color. We used timed transitions and d3 rendering functions to create this transition.

We struggled to find transitions between the remaining views since the elements were represented in different forms among these views (i.e. circles vs rectangles and rectangles vs. areas). With more time we would have tried to create innovative transitions between these views.

Bubble Chart View

To build the clustered bubble chart view, we had to learn how to use force simulations. The first step was getting the circles to even show. We had always passed 'cx' and 'cy' values explicitly and never derived them from a simulation, so this was a new technique for us. We learned that in a simulation, if the nodes positions aren't explicitly coded, then they will be automatically created. This was what the first example of creating the cluster looked like:

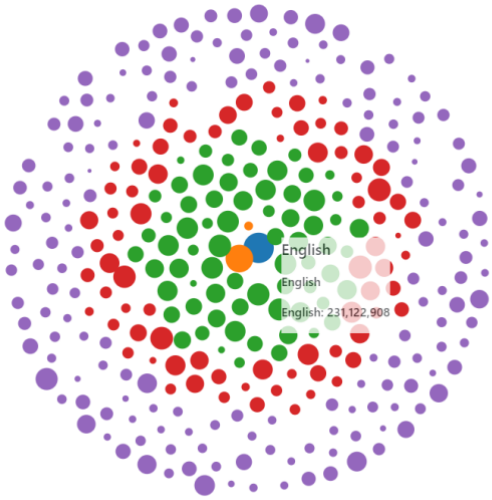


Figure 4: The default phyllotaxis arrangement

The guide that we used to help make this first visual was a combination of the tutorial for the course (http://dataviscourse.net/tutorials/lectures/lecture-d3-layouts/) and the d3 documentation on force layouts (https://github.com/d3/d3-force/blob/master/README.md). The problem with this first view was the circles overlapped. In addition to removing overlap, we wanted to cluster the circles instead of having them in rings. So we first got the collision detection working using this blocks page as reference: https://bl.ocks.org/d3indepth/9d9f03a0016bc9df0f13b0d52978c02f. See the next figure for a picture of what it looked like.

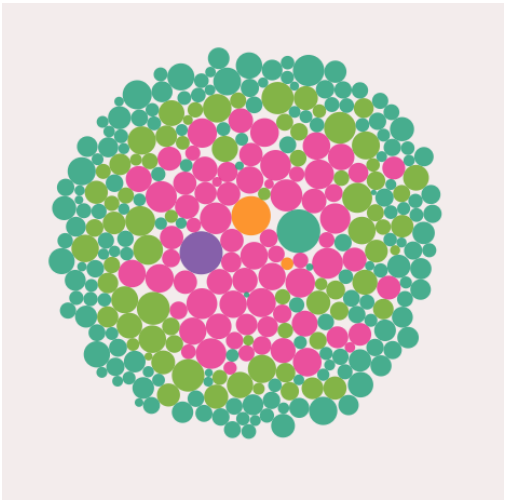


Figure 5: Collision detection was now working

Lastly, we wanted to get the clusters working. However, before this step we changed from a log scale to an exponential scale since it was easier to control the sizing between the nodes. Linear scaling didn't work since the smallest nodes were too small compared to the largest nodes, which is why we eventually decided to do exponential sizing as a happy medium between linear and log scales.

Part of getting the clusters working included using the force testing ground found here to fine-tune the simulation: https://bl.ocks.org/steveharoz/8c3e2524079a8c440df60c1ab72b5d03. The reference we used for clustering was found here: https://bl.ocks.org/pbogden/854425acb57b4e5a4fdf4242c068a127. Once we realized that we needed to have centroids that the circles clustered around and understood the code more clearly, we created this view:

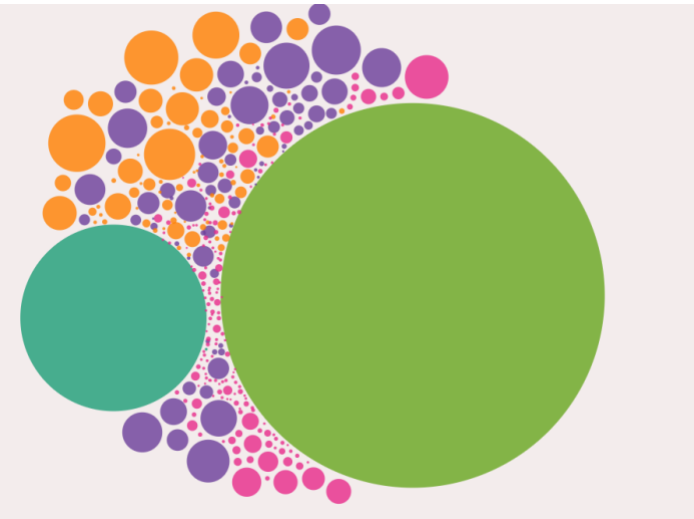


Figure 6: Initial clustering attempt

Finally, after fine-tuning some more, we ended up with the final result for our first visualization in our scrolling visualization.

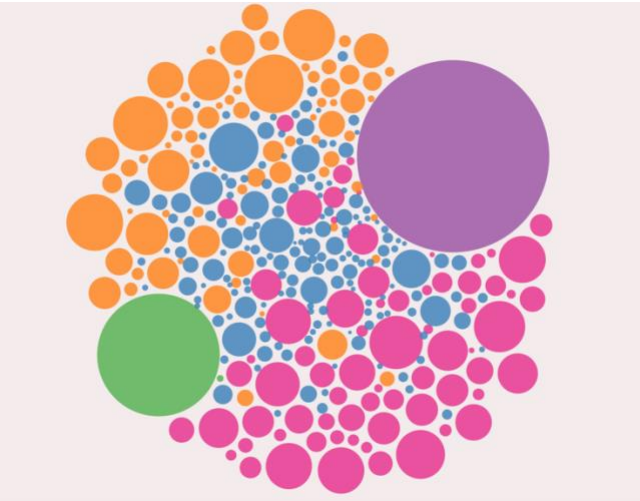
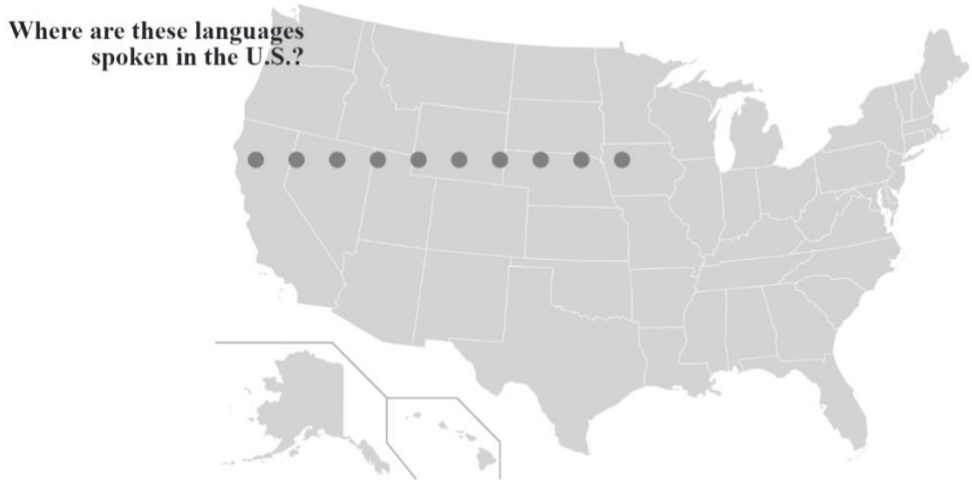


Figure 7: Final view of the clustering

Map View

The map view has gone through multiple changes since the original idea was created. While it still kept the fundamental elements of the idea (such as displaying the density of the speakers per state), the presentation changed a few times over the course of development. At first, we thought the speaker density should be displayed using a regional map, which would be color-coded by the density of the number of speakers given a selected language. We soon realized that this method wouldn't provide the option to view multiple languages at a

Figure 8 - Starting the Map



time. Because of that, we decided to change the speaker density to be displayed as circles.

Once we decided on circles, more details had to be resolved. The first problem was how to place the circles in the state centers (See figure 8). This took a while to figure out, but eventually we created an algorithm that found the height and width of each state block and assigned those locations to the circles. This was very slow to do in real time, so we copied the state centers into a variable which eventually became a json document. This was then tweaked as necessary to produce a clean map (See figure 9).

The next problem was circle overlap. Initially, every circle was drawn in the center of the state. Without added code only the last drawn circle could be seen (as displayed in figure 10). Unfamiliar with how to make circles not overlap, we created an algorithm that moved the circles based on their index within a state (See figure 11). Eventually we used clustering to solve a lot of the overlapping issues. This produced a fairly neat visual. However, the overlapping circles weren't the only challenge. An additional issue was the circle size and how to select the circles.

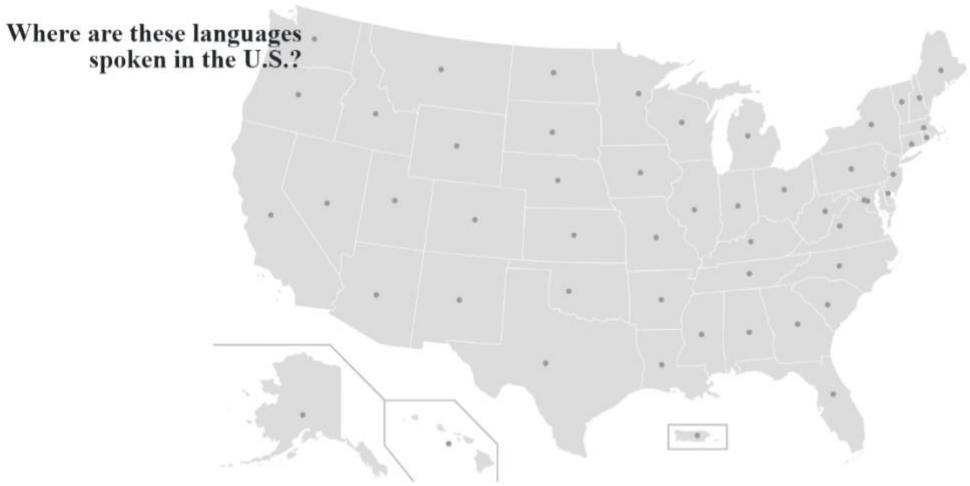


Figure 9 - Putting Circles in the Center of the States

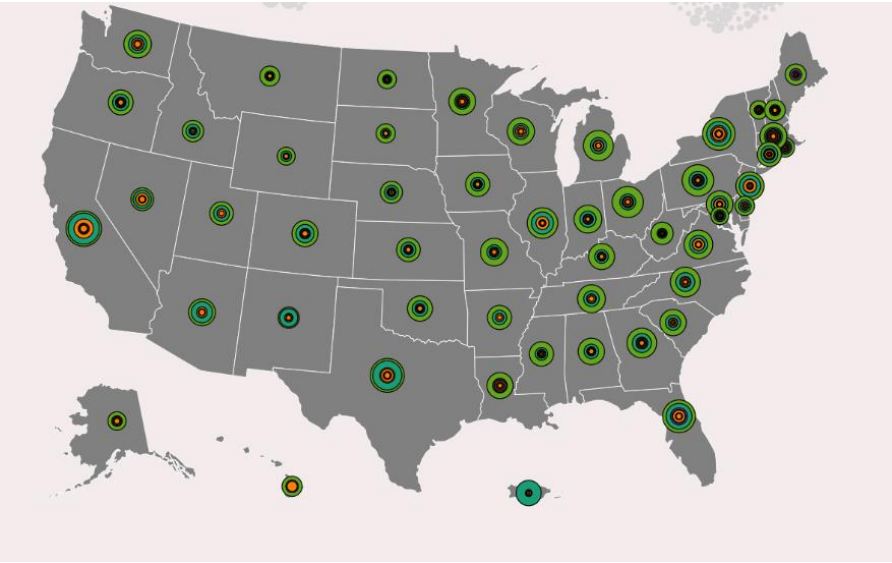


Figure 10 - Problem: Overlapped Circles

The circles, if sized proportionally, looked great when only one or two languages were selected. However, if more than a few languages were selected, the circles would quickly overlap each other and overflow into other states. To combat this, we decided to change the visual to do two things: 1) if there was only one language selected, we would show the state circles with varying sizes, based on

the number of speakers and 2) if there were multiple circles, we would show them with the same size and provide a hover function that would resize the language bubbles according to the language and the state (See figure 12).

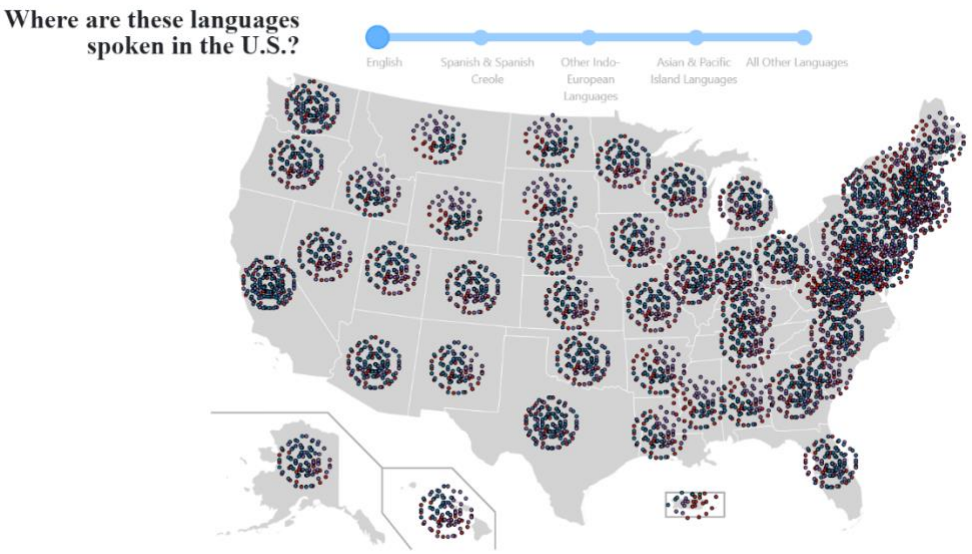


Figure 11 - First Selection and Scatter Circle Attempt: Slider

To address the selection problem, we started of using a slider (see figure 11), but forgot that we decided to use the language filters from the opening slide as filters. Then we implemented a brush to that would select the language circles. (See

Figure 12 - Second Selection Attempt: Brushing

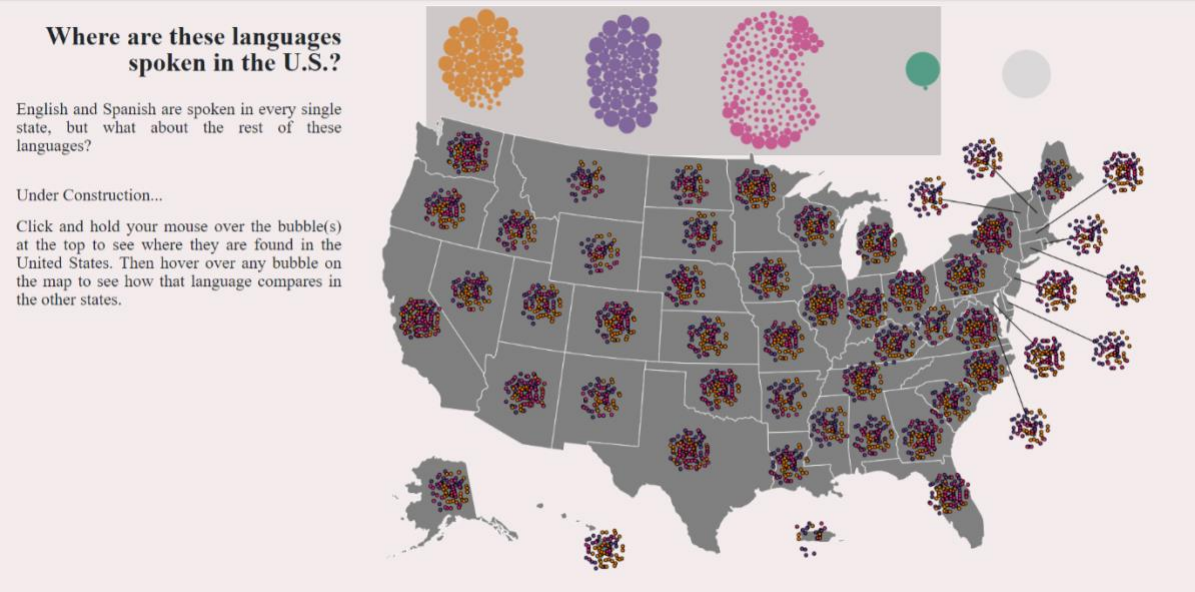


figure 12). We also combined the number of speakers of a language by using a “merge” toggle button. Figures 13 through 18 display the finished product.

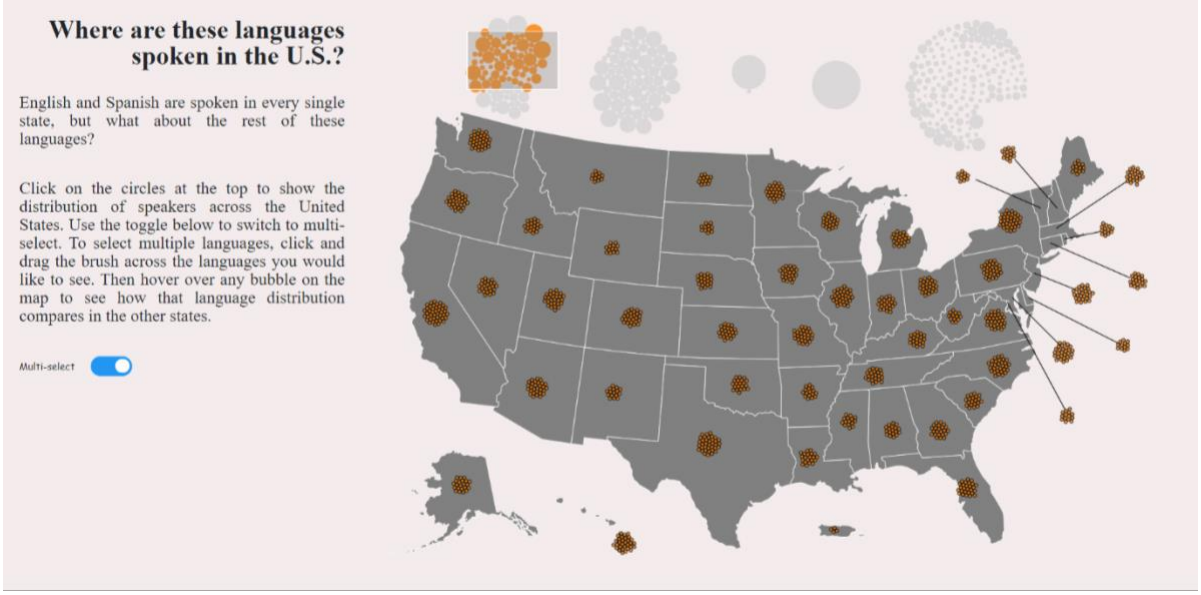


Figure 13 - Second Scattering Attempt: Clustering

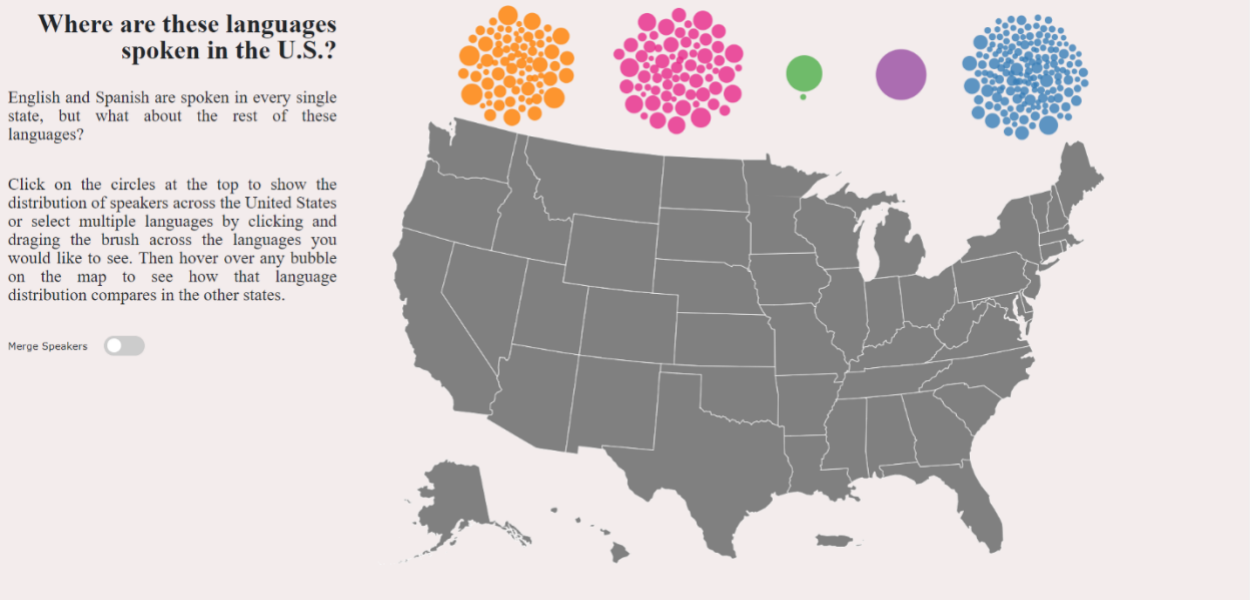


Figure 14 – Final Product: Map Visualization

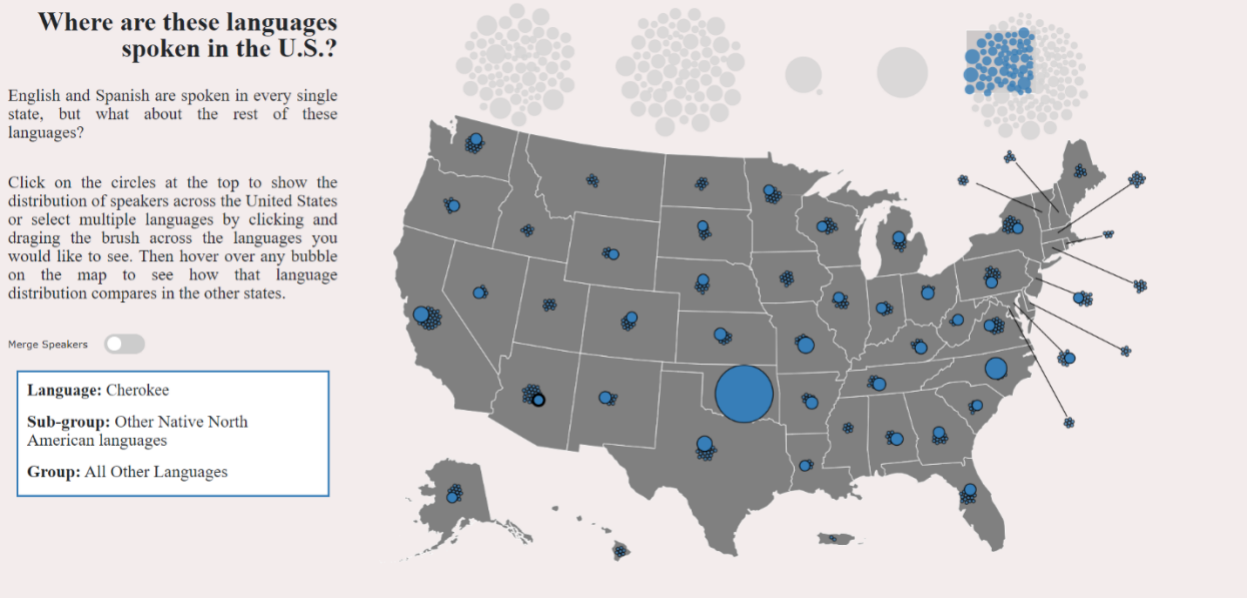


Figure 15 - Final Product: Multi-Select with Expanded Circles on Hover

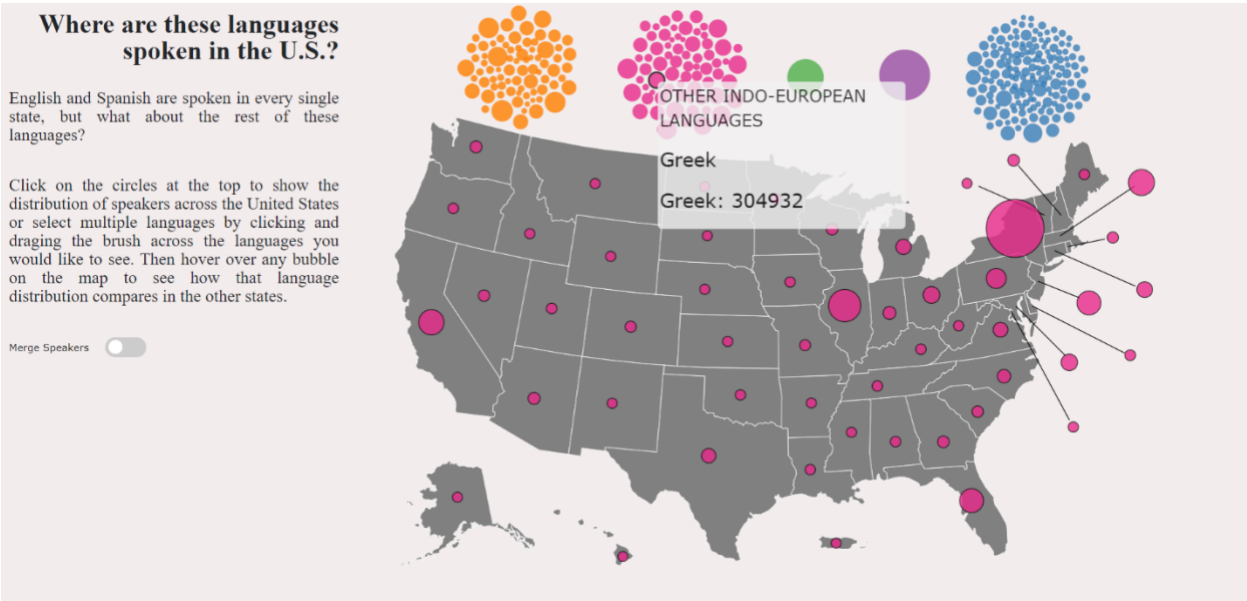


Figure 16 - Final Product: Single Select

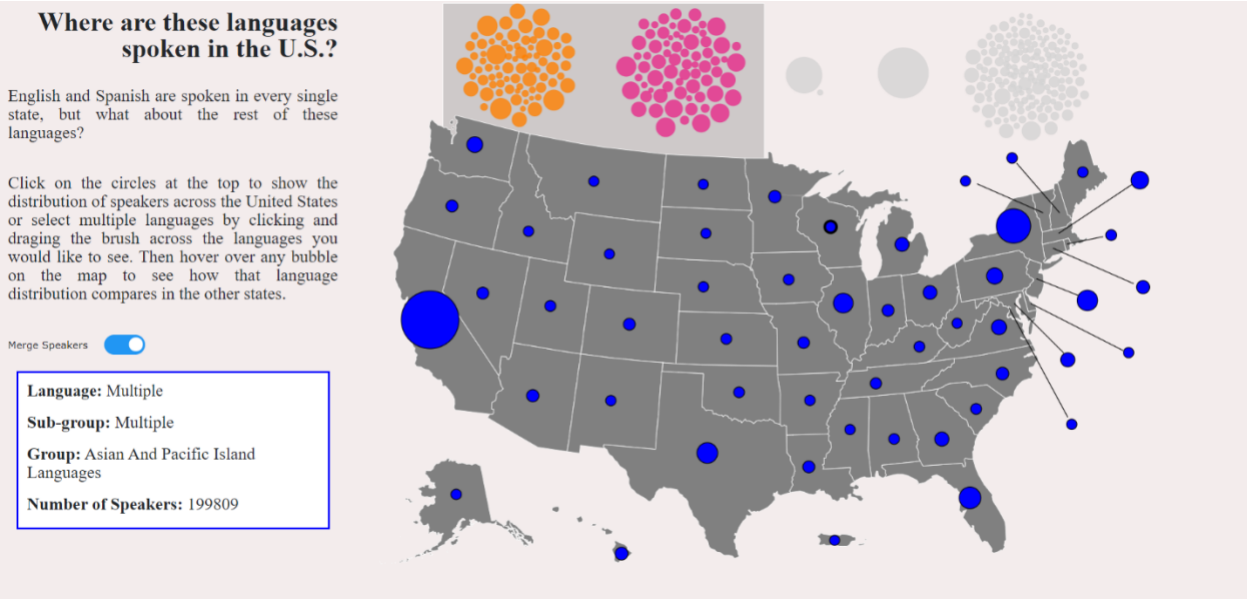


Figure 17 – Final Product: Merge Select with Multiple Groups

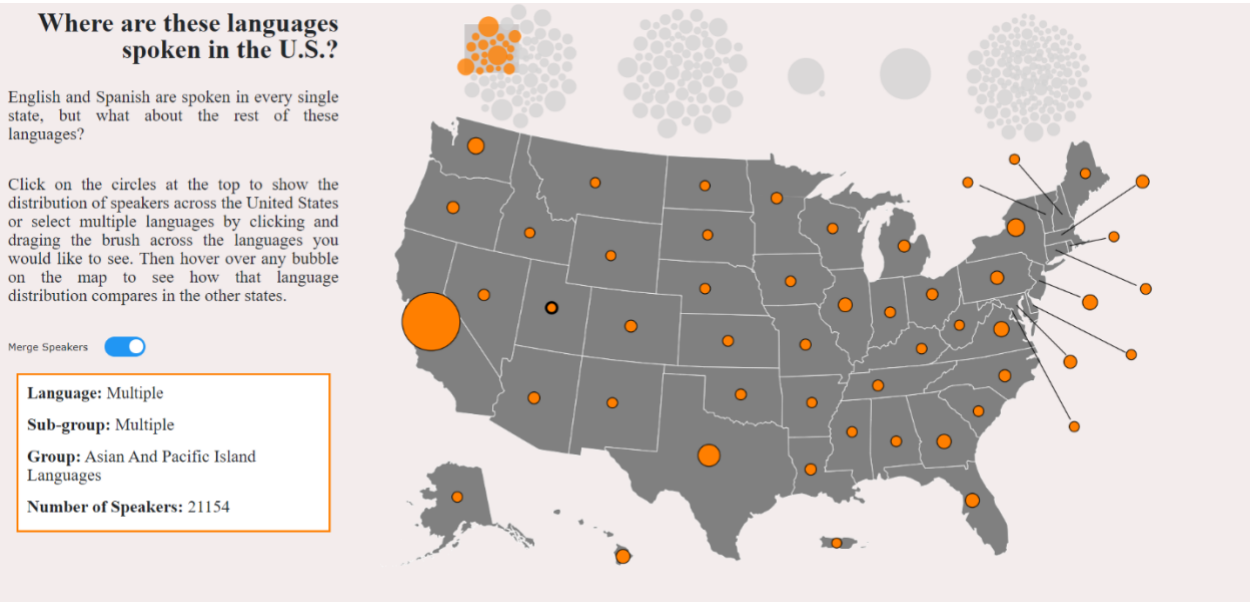


Figure 18 - Final Product: Merge Select with Single Group

Horizontal Bar Graph View

The horizontal bar graph was designed to show the breakout of foreign speakers for each state. The horizontal bars were meant to give the user a relatively accessible comparison of the foreign language diversity in each state. This

visualization used the datasets for all of the states. The languages for each state were summed by category using the d3 rollup methods. The categories are presented as the different colored bars in the graph. The first implementation was done in an html table layout, with the state names as one column and an svg element containing the bars/rects as the other column. Here is the first version of our bar chart:

Figure 19: Initial view of the horizontal bar graph

The next functionality we implemented was clicking the first bar and sorting by that bar. The same was done with the names of the states. The sorting was done by sorting the data bound to either the name column or the first rect in the svg.

Figure 20 First sort implementation of the horizontal bar graph

Sorting the first bar (i.e. the yellow category) was easy. Sorting by the other categories was more difficult since we wanted the sorted bar to appear first. By showing the sorted bar first, the user could make an easier comparison. Otherwise, the sorting didn't appear as intuitive or obvious. Incorporating this feature meant that the x attribute for each rectangle would change with each sort. We approached this the hard way at first before realizing that the d3 library incorporated a stack layout that could have made this process much easier! However, we only realized this once the implementation was complete. To compute the x attribute for each sort, we created an array that contained each possible x value depending upon the order of the bars, which in turn was determined by the group being sorted. When the sorting was finished, a tooltip was added that displays the information for each bar when the user hovers over that bar.

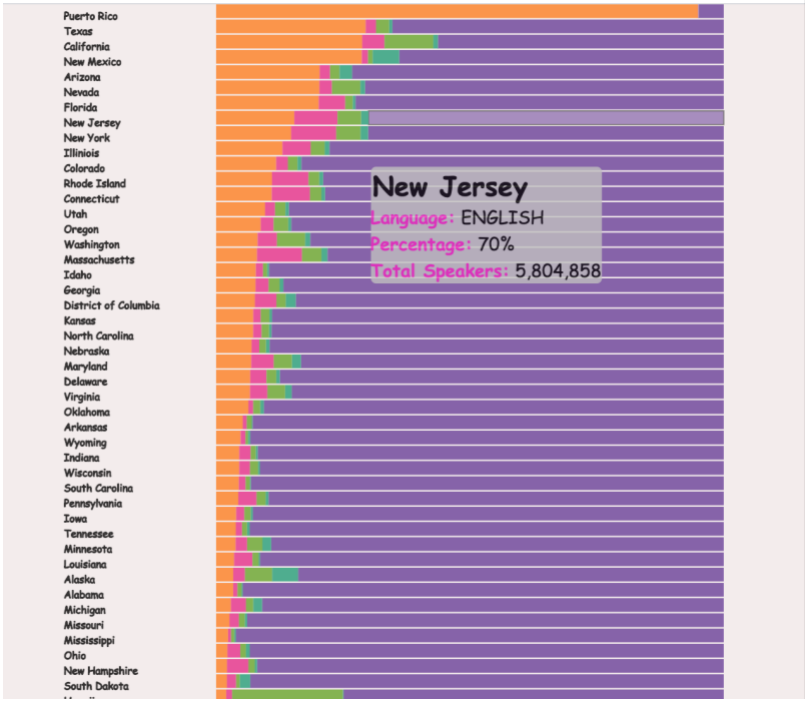


Figure 21: Sorting implementation completed and tooltip added to horizontal bar graph

An added complication came when trying to incorporate this view into the overall visualization. Our visualization was designed so that every view used the same svg, essentially rendering them on top of each other. Because of this, the table element that his graphic was originally rendered in did not work. Therefore, it had to be redone using all svg elements rather than the html table element.

The final version includes storytelling features on the left side in the scrolling section. We wanted to highlight some of the interesting facts about this data while also showcasing some of the interactive tools for the graph. When the user clicks the highlighted links/words in the storytelling section, the graph will highlight the information. We also included a legend so the different language categories could easily be determined, using the same color scheme and categories that were used in the previous views.

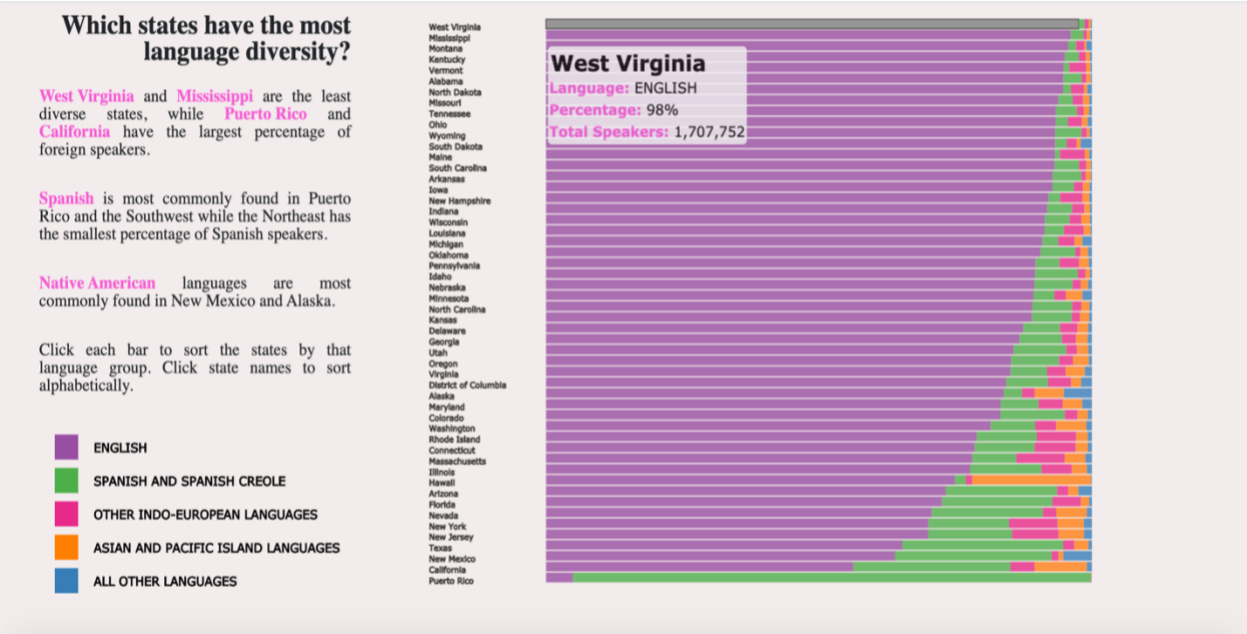


Figure 22: Final version of the horizontal bar graph including storytelling features

Vertical Bar Chart View

The purpose of this visualization is to explore which foreign language speakers are the most fluent in English. This visualization used the national dataset, so the languages were totaled for all of the states. The first step in the implementation was deciding how to represent the data, since we wanted to show the data at all three levels: group, subgroup and language. A tree data structure seemed to be the best fit. We created a Tree Class to store our data. Our data tree had four levels: the root node, the group level, the subgroup level and the language level as the tree leaves. The Tree Class incorporated several recursive methods to traverse the tree/data that would facilitate rendering different views of our graph.

Our original plan was to incorporate a diverging bar chart, with percentages of bilingualism on one side and total number of speakers on the other side. However, due to special constraints we decided to use a vertical bar chart instead. We chose to have the total number of speakers represented by the height of the bar. The percentage of speakers who are bilingual would be represented as a saturated line on the bar, with the height of the line

proportional to the percentage of bilingualism. Our first version is shown here, with the bars representing the top group level:

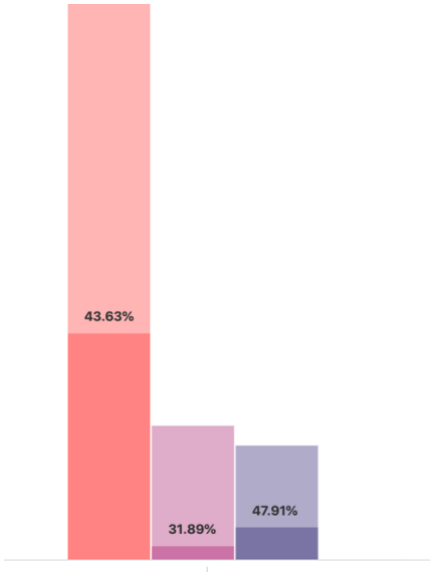


Figure 23: First version of the vertical bar graph

To navigate this graph, we wanted the user to click on a category. The category bar would then break out into subcategories, showing information for each subcategory. The user could continue down the tree by clicking again on a subcategory and that bar would break out into language data. Navigating the tree was one of the first design choices. We wanted to make the navigation as simple and intuitive as possible. We decided upon having the user click once to drill down the tree and double clicking to go back up a level. If the user was at the bottom (hence the language level) and clicked nothing would happen. The same would be for the top (group level) if the user double clicked there. The first challenge encountered was that a double click also fired the click event. We needed to distinguish between the two events. We found an answer online at http://bl.ocks.org/couchand/6394506 that we based our implementation on.

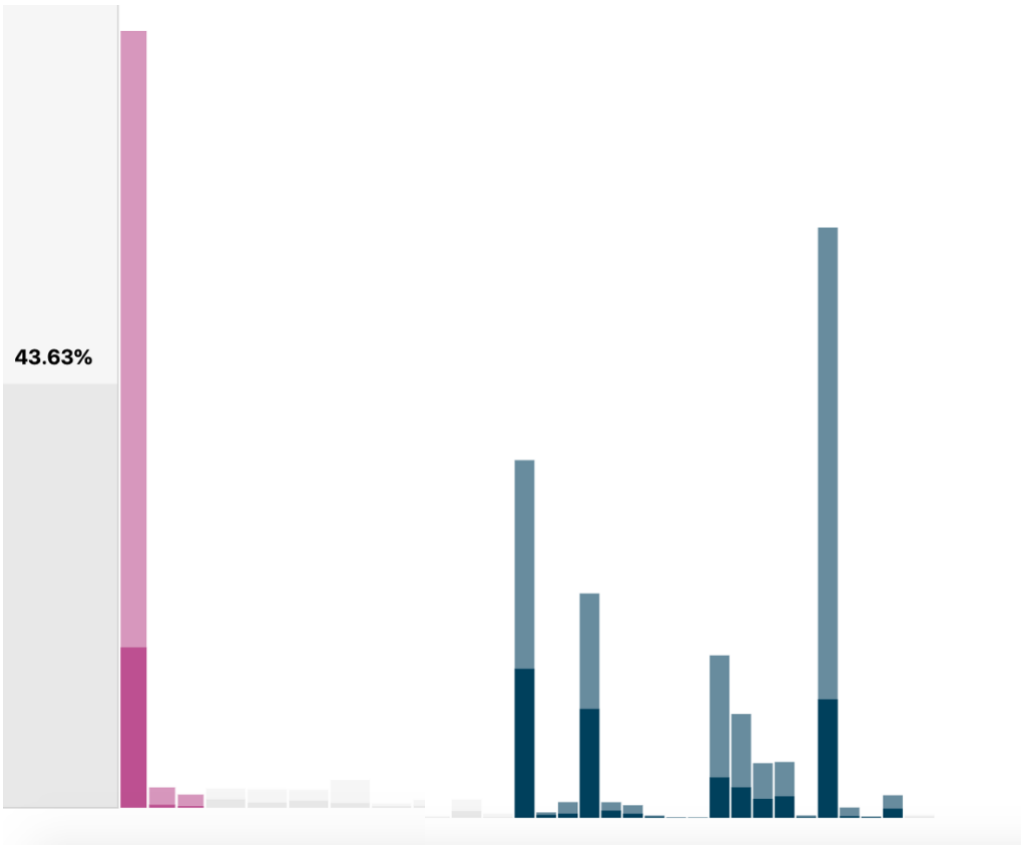


Figure 24: Examples of scaling issues with vertical bar graph

A few views of the graph shown at subgroup and language level are shown above. As can be seen, one of the biggest challenges was the large difference in the scale of the data. Some languages have as little as 100 or less total speakers nationwide. On the other hand, Spanish has over 37 million speakers! Trying to capture this data in a meaningful way on the same graph was challenging. Some of the data was so small that it was meaningless in the graph. Ultimately, we decided that showing the total number of speakers for each language was distracting and complicating the visualization and also duplicating the same information presented in our first view. We decided to only show the percentages for each language within the bar graph. This eliminated the scaling problem and made the graph much easier to see.

Here are some views of the group, subgroup and language levels using only the percentages for the bars. The scaling looks much better! A tooltip was added for additional information when the user hovers over a bar.

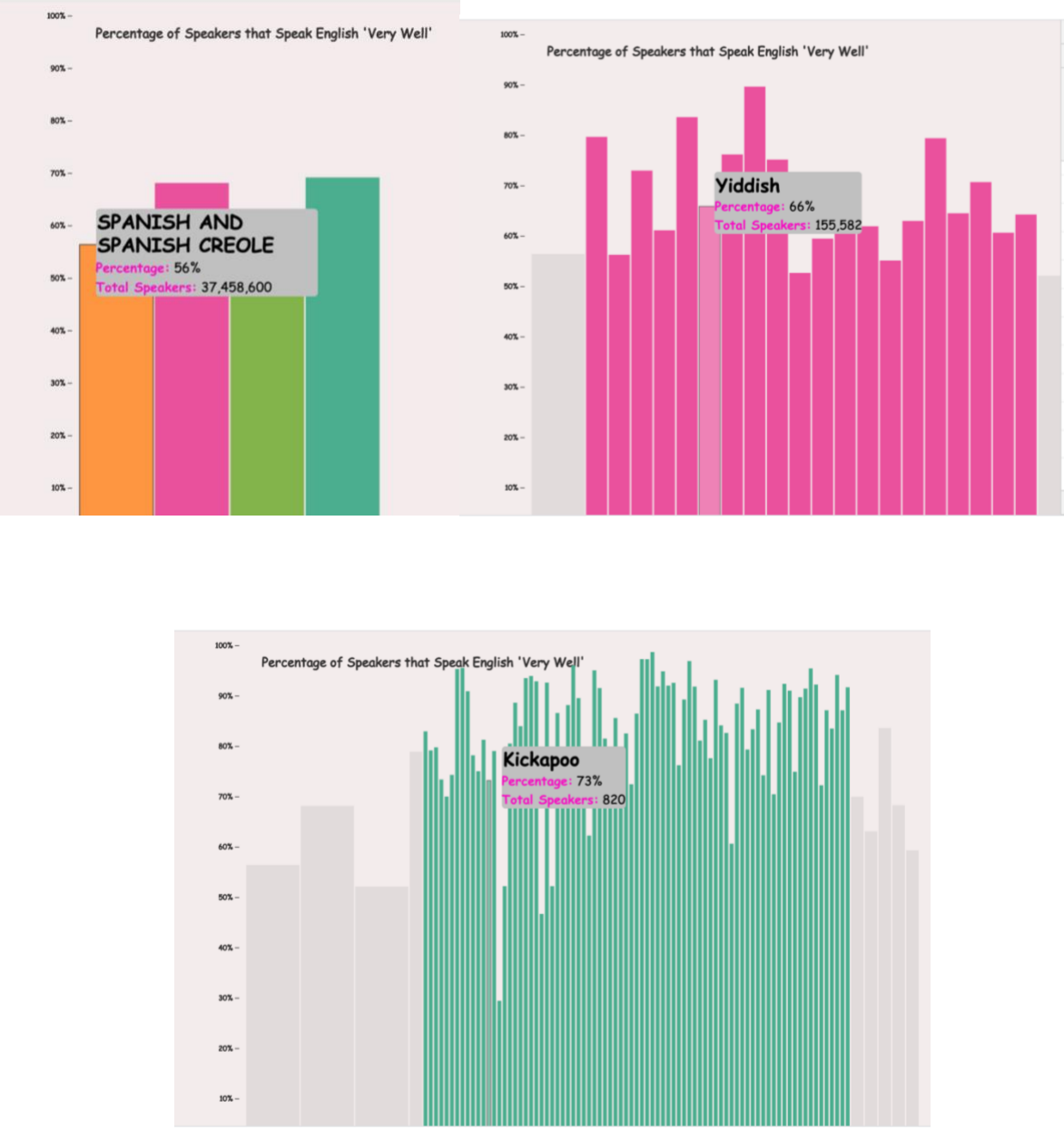


Figure 25: Examples of different levels selected within the vertical bar graph

The next functionality we wanted to incorporate was a sorting function, so that each group, subgroup and language could be compared to all of the other similar items. Below is a view of all the languages sorted.

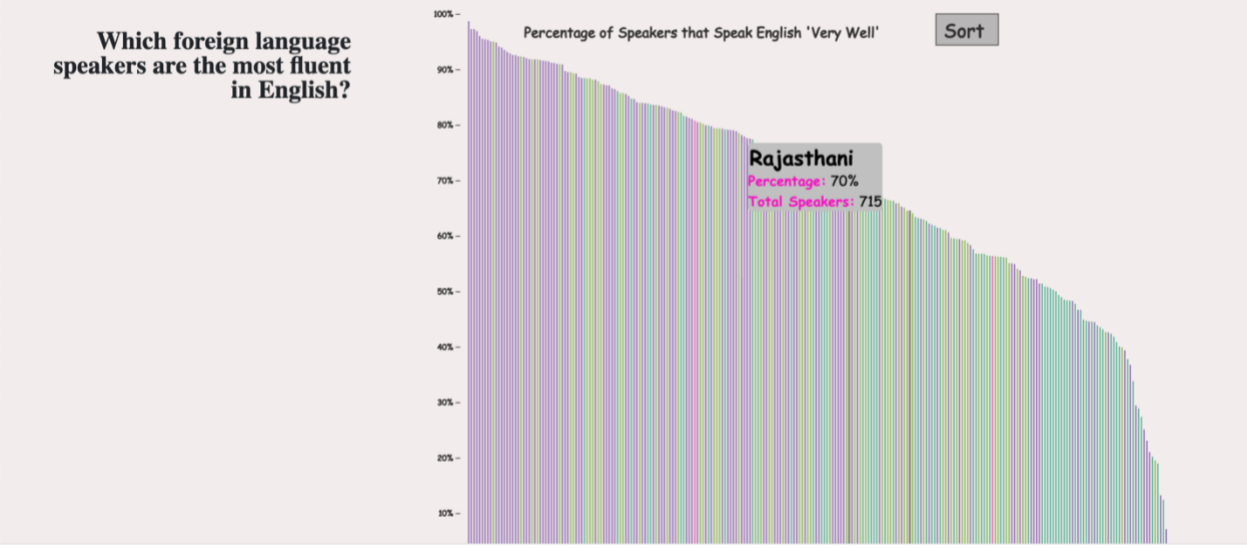


Figure 26: Sorting function implemented into vertical bar graph

For the final version we added some storytelling links and a home button so the user could easily return to the original view. Again, we wanted to highlight some of the interesting facts about this data while also showcasing some of the interactive tools for the graph. Below is a final version of this graph.

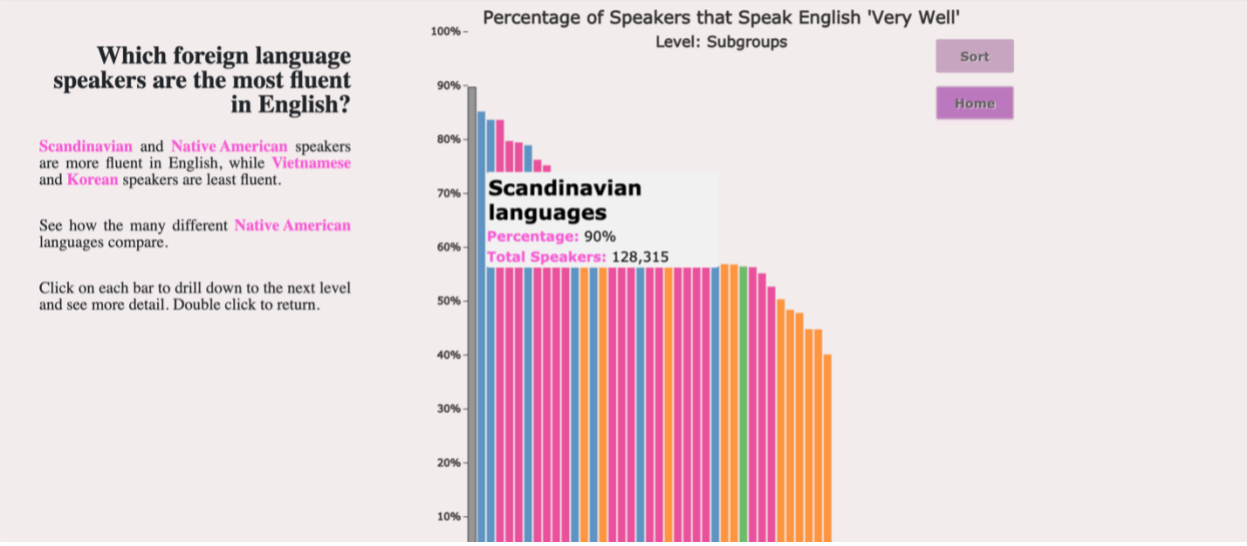


Figure 27: Final version of the vertical bar graph including storytelling features

Area Chart View

This graph is meant to highlight the changes in the distribution of foreign languages within the U.S. over the past 40 years. We would have liked to

incorporate a larger timeline into our graph, but only data from 1980 onwards was available. We considered using a line chart to display the data. We decided to implement an area chart because of its visual appeal and coloring and because we wanted the challenge of learning some new skills in d3!

This graph used a different dataset from the other graphs, although this dataset was fairly small and simple so there wasn't much work to clean and prepare the data. To incorporate the area chart, we used the d3 stack layout feature and the d3 area function. The d3.stack() function calculated coordinates for all of the stacked areas. Then, these coordinates are used in the d3.area() function to creates svg paths to render the stacked areas. After spending some time reading the documentation and several tutorials, we had a working area chart.

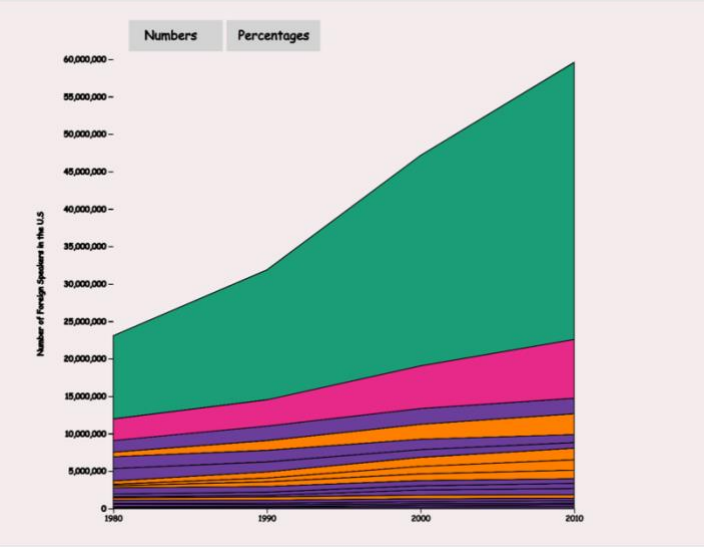


Fig 28: Numbers view of stacked area chart

The first view contained the total number of speakers for each year (1980 – 2010) in 10-year increments. We created a second view that showed the percentage of each particular group compared to the overall total foreign language speakers. This view can provide additional information as to how the overall distribution is changing for each group.

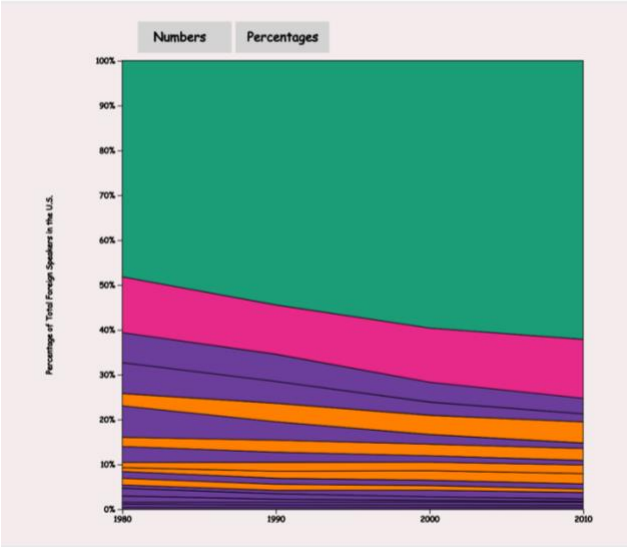


Figure 29: Percentages view of stacked area chart

The next feature we added was a tooltip that shows the name of the language when the user hovers over an area and also greys out the other languages that are not highlighted.

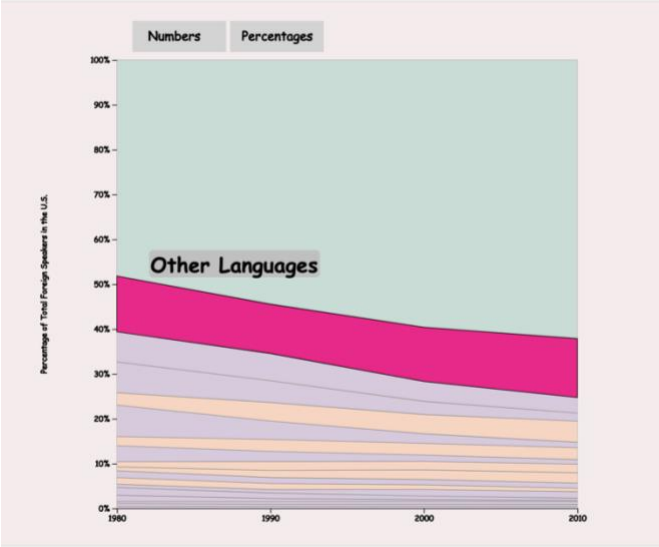


Figure 30: Highlighted area with tooltip in stacked area chart

Next we wanted the user to click an area and have only that area/language shown. To do this, the stacked coordinates had to be recalculated with only one

stack. Then these new coordinates were again fed into the area generator to create the path for rendering. An additional feature with this view is a tooltip that shows information about each year as the user mouses close to that year. This involved creating a "snap" for the tooltip. A d3.bisect() function was used to calculate the snap for the mouse.

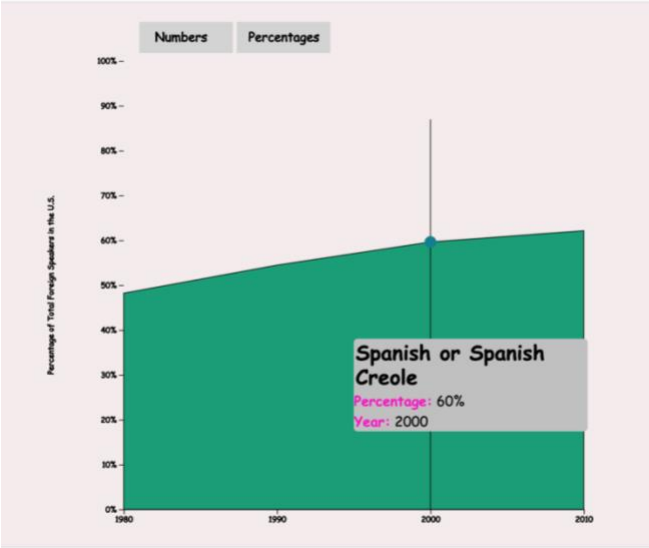


Fig 31: Single area shown with tooltip snap

To finish this view we added storytelling features and a category legend to the left of the view.

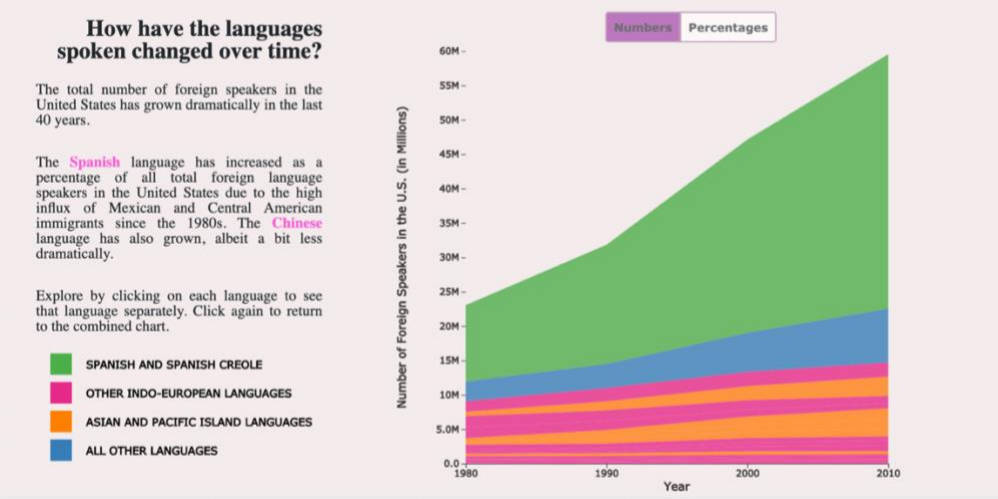


Figure 32: Final view of the area chart with storytelling features

EVALUATION

We learned a couple of things from our visualization. The first is that there are quite a few languages in our country! We didn't realize how many different Native American languages there are (most of them are under the 'All Other Languages' group) and that most of the Native Americans are in Alaska, New Mexico, and Arizona. Second, we learned that the number of English to Spanish speakers is about 6 to 1. Another fact that become even more obvious is the increase in Spanish due to the large influx of immigrants since the 1980s from Mexico and Central America. This trend is most noticeable in the Southern border states. Southern states that are not near the border have the highest amount of English speakers and states that have big cities like New York are more diverse.

We had a set list of questions that we wanted answered when we made a rough draft of our visualization. Then we structured our visualizations around those questions with one question for each view. Those visualizations can each answer multiple questions, but they are all designed around one central question/theme for that view. The storytelling aspect of our visualization guides the user through the visuals.

The visualization works well, and we have refined it to try and eliminate as many errors as possible. To improve it, we would love to use per capita information on the map view so we can compare the states between each other with circle sizes proportional to the population. Another improvement would be adding filters on some of the views so that you can quickly find a state or language that you are looking for. As mentioned above, we would include more transitions to increase the continuity and feel of the experience. In summary, though, we were pleased with the result.

Initial Project Proposal

Basic Information

U.S. Languages Datavisualization
Github Repository Link:
https://github.com/DataVis-Fall-2020-Team/uslanguages.github.io
Group team members:
Janaan Lake
- UID: u0987016
- Email: janaanl@hotmail.com

Andreas Martinson
- UID: u1291396
- Email: andreas.martinson@utah.edu

Rachel Berghout
- UID: u1302108
- Email: u1302108@umail.utah.edu

Background and Motivation

We were inspired by a tree based visualization of the 100 most spoken languages around the world, the link to the viz can be found here: https://www.visualcapitalist.com/100-most-spoken-languages/ This led us to decide on our topic of language. Then after looking at available datasets we found a dataset that was interesting, but only focused on the U.S. That dataset also led us to focus on the nationwide and state levels of detail since the county level data was too sparse.

When we were thinking about how to present the data, we really liked the storytelling scrolling approach visualization found at this link: https://cuthchow.github.io/college-majors-visualisation/ .
This influenced the method that we want to use to present our data.

Project Objectives

We have four primary questions that we would like to answer:

- Which languages are spoken within the U.S.?
- Where are these languages spoken within the U.S.? (i.e. what

is the distribution of the language groups)
- Which states have the greatest language diversity and what languages are spoken in those states?
- Which foreign speakers are the most fluent English speakers?

Another question that we might address if we have time is, “How have the languages in the U.S. changed over time?”

Based on this visualization, we hope that a user would use it to find more about the state they live in and compare it to other states they are interested in. We hope that it will help the user learn the distributions of languages and compare that to their expectations as well. From a technical standpoint, we are excited to learn the scrolling storytelling technique as well as the transitions associated with using it. We also hope to revisit some of the visualizations we have already created (bar charts and maps) and customize them to help tell the story that we want to tell. In the end, we want to also call out some interesting facts to engage the user with the visualization.

Data

The dataset is from the U.S. Census Bureau and can be found at this link:
https://www.census.gov/data/tables/2013/demo/2009-2013-lang-tables.html

The data preprocessing that we would need to do is the following:

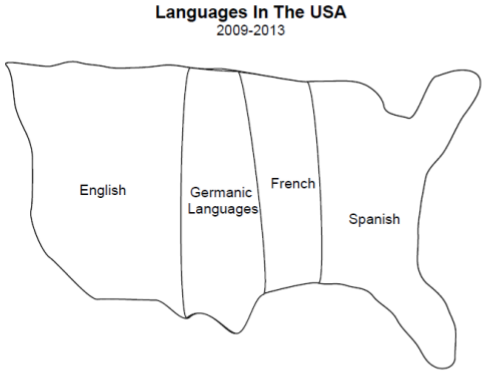
- Combine state data into a single tab
- Add a column for the language grouping instead of having that information bolded and in the same column as the languages
- Change the names of some of the language groupings to avoid parenthesis and punctuation

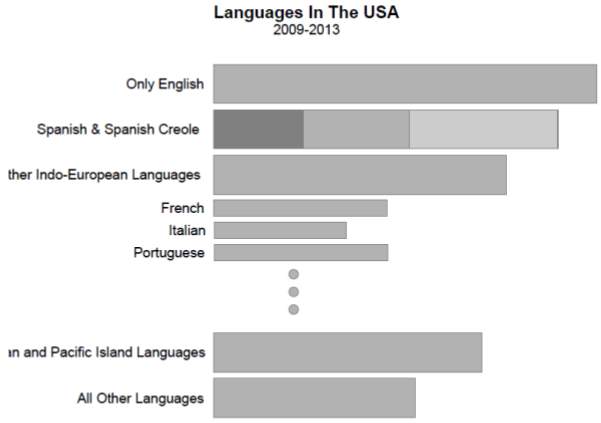
Project Proposal – Initial Sketches

What languages are spoken in the US?

Ξ

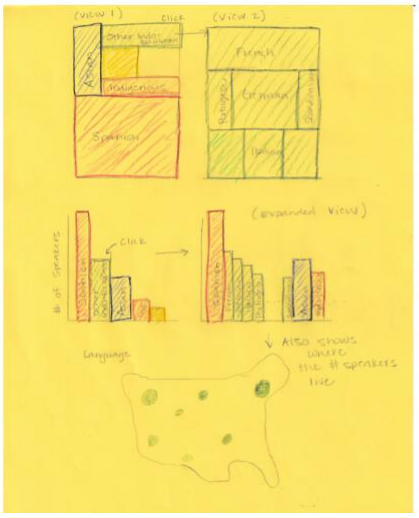
https://www.census.gov/data/tables/2013/demo/2009-2013-lang-tables.html



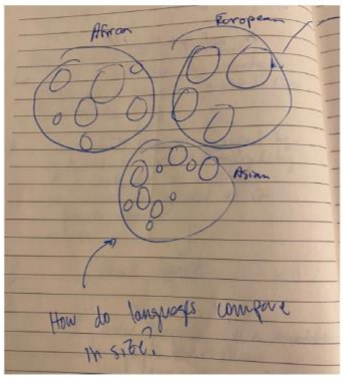


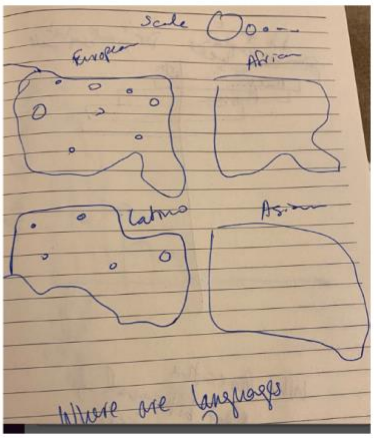
What languages are spoken and where?

Coordinated tree maps/bar chart with map

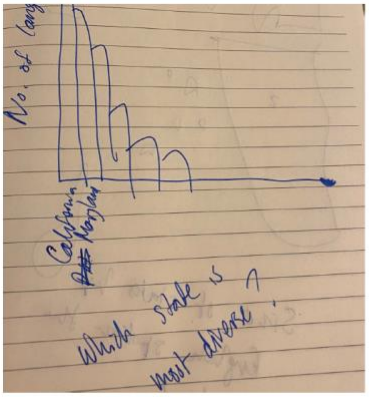


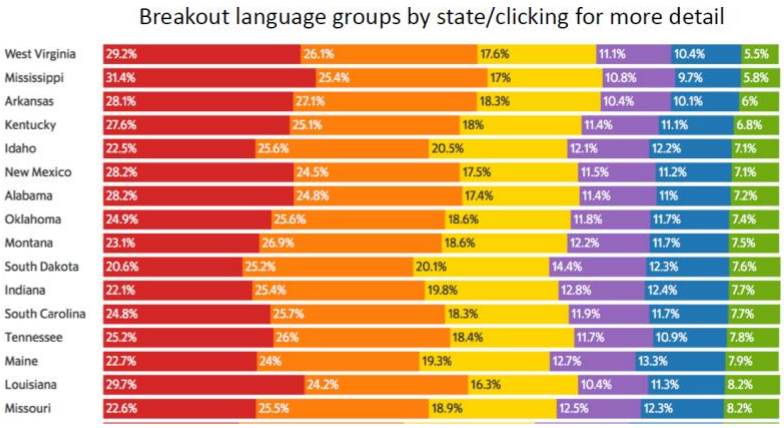
Circle swarms/groups with coordinated maps



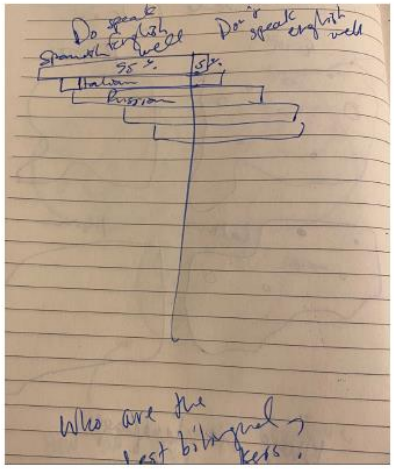


Which states have the most language diversity?

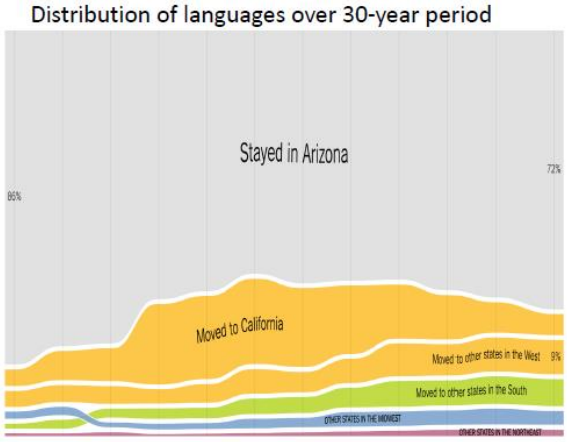




Which speakers are more
bilingual? How have the languages spoken changed over time?

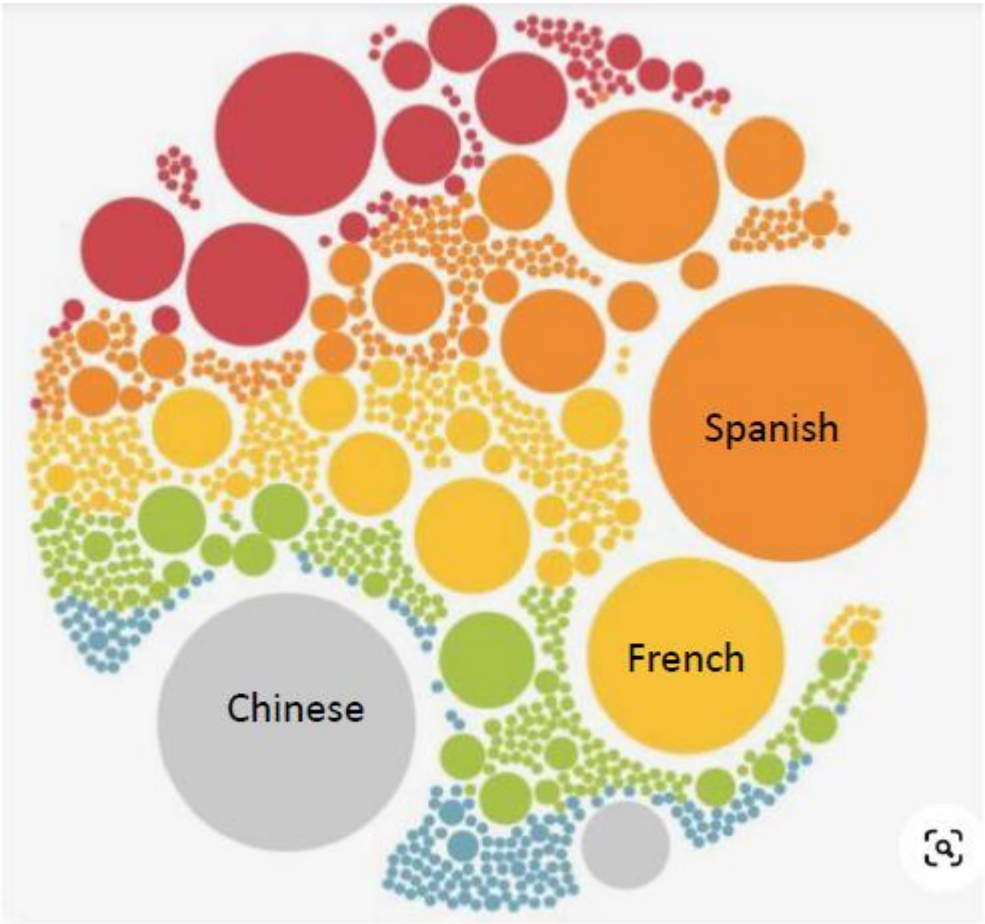


Diverging bar chart with percentages of bilingualism on one side, total number of speakers on the other side



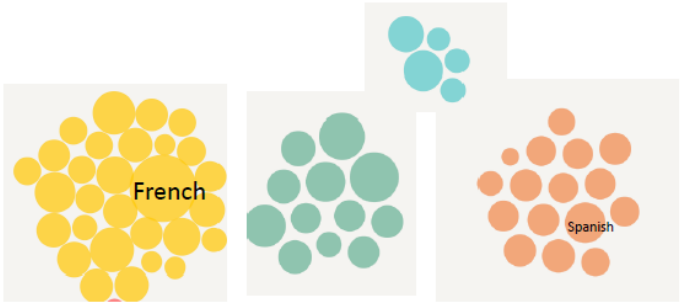
Project Proposal – Final Project Proposal

What languages are spoken in the US?


Spanish
French
Chinese

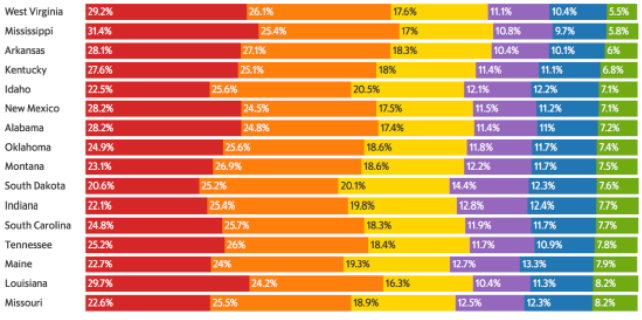
Where are these languages spoken in the United States?

- Animated transition from first view to colored swarms
- Same circles and sizes from first view
- When user clicks on a circle, the name appears and then the map shows where that language is spoken in the US on a chloropleth map. The color will be the same as the clicked circle, and saturation of the color will be the channel represent the percentage of speakers within that state.
- Storytelling feature: Pick a few languages and highlight where they are spoken
- Optional: Brushing and/or toggling on circles for multiple selections



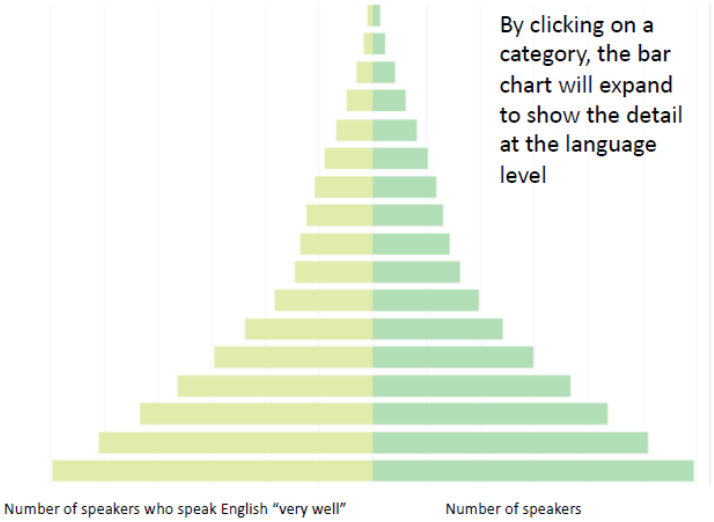
What states have the most language diversity and what are the languages spoken in a specific state?

- Colors are the same colored language groups as before
- Each category/color can be sorted
- Option to include/exclude English
- A tooltip shows the total number of languages on hover
- Storytelling feature: Highligh states with the most and/or least language diversity and why that might be the case.
- Optional: The user can click a category bar within a state for a breakout of the languages represented in that bar and their statistics within that state



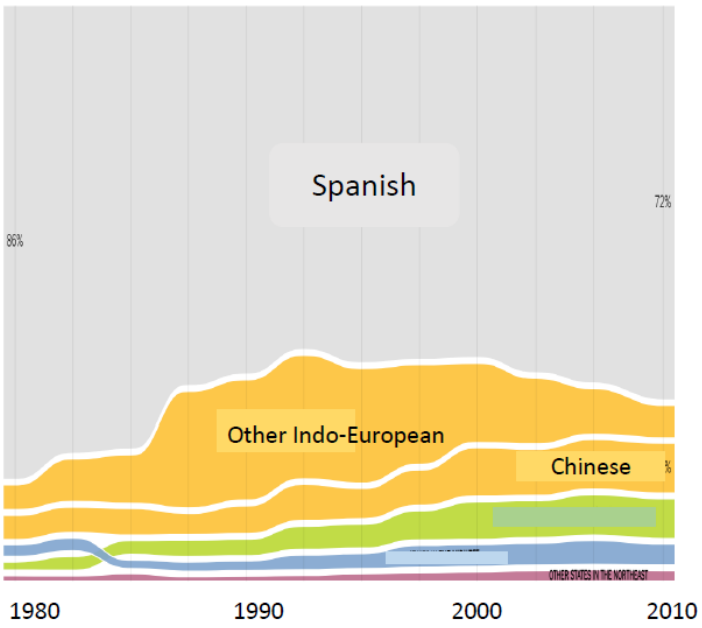
What foreign languages speakers are the most bilingual?

- Same color groupings
- Start at grouping level
- When user clicks on a group, the bar chart expands to be at the language level for that colored group
- Storytelling feature: Highlight extremes and give reasons/explanations for why



(Optional) How have the languages spoken in the US changed over time?

- Dates from 1980 to 2010 with 10-year increments
- Tooltips highlight sections when hovered with more detailed information, such as in the example shown
- Same colored groupings as before
- Storytelling feature: Explain that Spanish and Chinese are increasing due to the large immigration from these areas. Decrease in French, Italian, German as those communities have largely integrated. Dying out of indigenous languages.



Must-Have Features

- View #1 - A bubble chart displaying the percentage of languages in the U.S. colored by the continent the language comes from.
- View #2 - A U.S. map displaying the usage density of the language group chosen (African, Asian, European, etc.). Will use a filter to display the different language groups by continent.
- View #3 - A bar chart displaying the percentage of each language per state.
- View #4 - A bar chart displaying the language speakers by group that speak more than one language fluently. Groups will break out into individual languages.
- Storytelling features in each of the views.

Optional Features

- View #5 - timeline flowmap
- View #2 - Map - toggle button to combine languages into groups
- View #3 - Bar Graph - Transition to break into separate languages vs. pop out bars beneath

Project Schedule

Nov. 5th - Peer Feedback
Nov. 15th - Project Milestone
Dec. 2nd - Project Due
Week 1 - 10/29 - 11/5

- Set up the file structure
  - Style.css
  - Megacluster.js
  - Map_cluster.js
  - Horizontal_bar.js
  - Diverging_bar.js
  - script.js
  - Main.html
- Get the data loaded
- Write script file
- Scrolling layout - Use example as template

Week 2 - 11/6 - 11/13
- Megacluster - Andreas
- Map cluster - Rachel
- Horizontal Bar/Diverging Bar - Janaan

Week 3 - 11/14 - 11/21
- Finish all layouts
- Megacluster - Andreas
- Map cluster - Rachel
- Horizontal Bar/Diverging Bar - Janaan

Week 4 - 11/22 - 11/29
- Focus on transitions/animations

Week 5 - 11/30 - 12/2
- Final polishing

Weekly Scheduled Meetings:
- T,TH @ 8pm-9:30pm

Peer Review

Date: 11/5/20
Peer Reviewer Names: Ricardo Gonzalez, Bao Chau Pham, Lucas Idstrom
Our Group: Rachel Berghout, Andreas Martinson, Janaan Luke

Our Feedback

The other group's visualization was based on immigration worldwide, they have over 40 countries immigration data going back to 1980. The dataset is incomplete; for example, they have immigration data in the U.S., but not emigration data. The main interaction is through clicking on a country and once a country is clicked on, it will show a lot of other related graphs. The critiques that we provided included making sure that their visualization was clear in what it is communicating. When a user clicks on a country, they need to understand the scope of all the different graphs that they were seeing. Another critique that we provided was mentioning that they had some filters at the top that only applied to some of the graphs. Some alternatives we suggested were duplicating some filters and have them be directly over the graphs or group the tables together so it is clear that filter only applies to the specific graphs looked at. A third critique that we gave was suggesting the year filter be a slider instead of a dropdown. This makes it easier to see changes between years and interact with the visualization. Finally, the last

suggestion was recommending an animation to show time lapse between the year or some sort of way to compare countries. They mentioned that those features will be included in their list of optional features.

Peer Review Group Feedback

After looking through and understanding how our visualization will work, the first suggestion that they offered was to make sure that the transitions are thought through when using the scrolling storytelling feature. Another suggestion for improvement was adding annotations to our graphs to point out some interesting points. After looking at our data, they also suggested that we use the 'sub-group' column and incorporate that into our visualization somewhere. Another improvement related to our dataset was suggesting we explain where the data came from and the context.

The visualization that they were most concerned with was our chloropleth map. They mentioned that having circles might be an improvement since with dots we have hue and size as channels, but with a chloropleth we only have color and comparison could be more difficult. Related to this same view, they also suggested that we consider putting more than one language on the map at a time. This feature was part of our 'optional' considerations for improvement.

Analyzing the Feedback

Later in the day, we had a group meeting about how to address the feedback that we received. We agreed that we should think the choropleth map through more and found another visualization that has all dots already drawn and then just changed the size and color once they are chosen.