United States Population Analysis From 1969-2019

Taylor Dawson, KC Gubler, Carson Sorenson

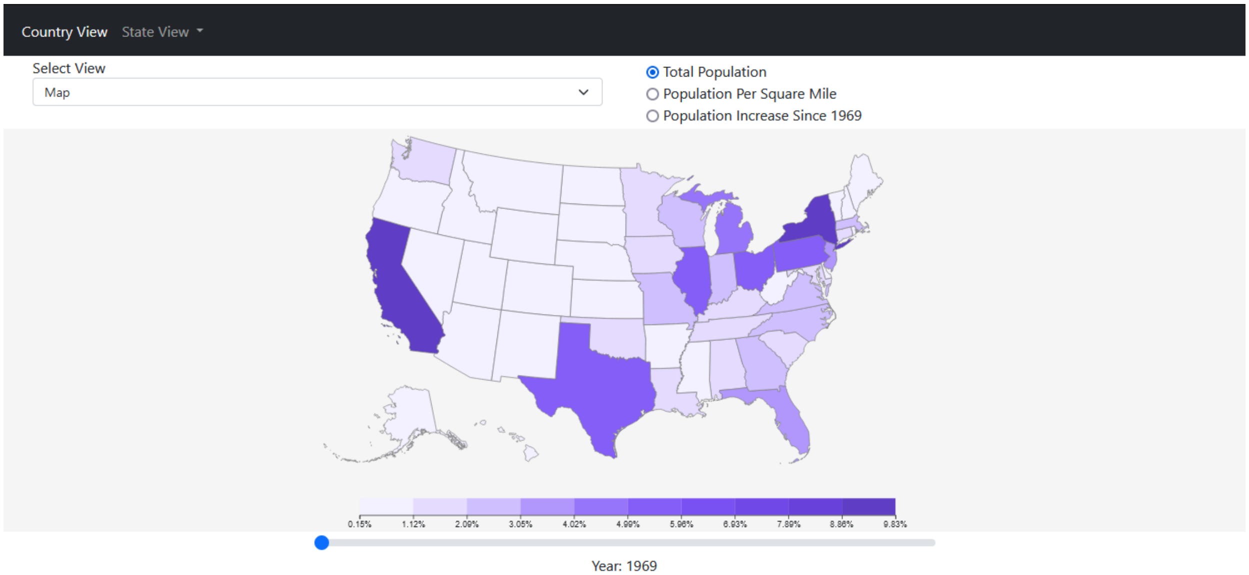

Fig. 1. Webpage that shows interactive United States population change website

Abstract—Being able to explain how and why populations grow and decline becomes exponentially easier when illustrated graphically. This can assist politicians, construction companies and businesses, both large and small, when deciding how to deal with housing shortages, prepare for housing booms and where to move your company to maximize profit. We collected information of population change in the United States from the years 1969-2019 and displayed these changes graphically. We added user interactions such as a slider, toggles and mouse interactions to increase the usability of the data visualization tool to make it easier for the user to decide what it is they want to know. We used Mongo to store our data in the backend and used HTML, CSS and Javascript to display that data in the frontend. We found from one user we tested it on, he really appreciated the availability of the user interactions (slider, toggles, drop-down menu, etc.) so see exactly what it was he wanted.

Index Terms—Population change, D3, MongoDB, Express

1 Introduction

Population growth and degradation affect cities, counties, states and even countries on a variety of levels. For this reason, we chose this project. We are particularly interested in visualizing how the population of the United States has changed in the past fifty years. As a consequence of not having sufficient time to delve deeper into the effects of population change and their causes, we only provide documentation on how we display the results. It is important to note, however, that further research could be done to discover if the following reasons may have caused the change in population:

- What policies were implemented in each state that initiated those increases/decreases in population?
- Were there housing shortages in regions that experienced sharp spikes in population increases?
- What effects on food prices would population decreases have in farming and ranching communities?

We believed it would be particularly advantageous if a user was able to visualize how their state population has changed over the years and be able to compare that data with the surrounding states. We decided to do this project because we thought modeling the changes in the population between the different states would be interesting (shades of color, dots, etc) and that it would be interesting to see the population growth over time. Thus, we found a data set that had population data over an extended period of time.

With our visualization, we will attempt to answer the following primary questions:

- What was the change in total population from the years 1969-2019 in each state?
- By what percentage did each state grow from the year 1969?
- How populated has each state been from the year 1969 (calculating the population per square mile)
- What percentage of the entire US population does each state represent?

- *Taylor Dawson is with Utah State University. E-mail: taylor.dawson@usu.edu.*
- *KC Gubler is with Utah State University. E-mail: cassandra.gubler@usu.edu.*
- *Carson Sorenson is with Utah State University. E-mail: carson.sorenson@usu.edu.*

- How is the population distributed throughout the state? What is the county with the largest population?

The intended benefits of our visualization system:

- Help provide data for someone looking to relocate.
- Provide total population data in addition to more granular statistical data. The user should be able to view which states have grown the most and what states are the most densely populated.
- Allow non-trivial questions about the population of each state to be answered.
- Display a large amount of data in a little amount of space.
- Provide detailed visualizations of each state, such as population by county, and how the state has grown over time.
- Easily identify trends in each states changing population.
- Much faster at identifying mistakes/bad data points (hard to see inconsistencies in numbers).

While working on this project our goal is to learn how to build a visualization system from scratch. We hope to learn how to work well as a team and properly assign work to each other so each component of the project can be completed. Additionally, we hope to learn how to use MongoDB and NodeJS and successfully integrate these frameworks with our front-end web application.

2 Background

Provided below is documentation of the pertinent subject matter.

2.1 Literature

As rivers move, the landscape it flows through changes. So it is with population growth and decay. In a study by D. Pimentel and S. Cooperstein, it was discovered that as populations increase, so does the number of diseases worldwide and the degradation of landscapes. In the same paper, it is also projected that by 2025, 66% of the the world's population will be living in urban regions [6]. Pimentel and Cooperstein help establish the importance of why we should be concerned about how populations are changing.

When geography and how populations change meets big data, answers to decade-old questions are made available to us to use at our disposal. In a special issue of Geographic Data Science by Alex Singleton and Daniel Arribas-Bel, it was argued that geography benefits from data science by developing methodological and technical aspects of working with "Big Data." Data Science benefits from geography by developing a critically reflective perspective for new computational approaches to location problems. Additionally, data scientists benefit from "methodological contributions that better account for some of the key challenges in building models with spatial data" [7].

But, more specifically, how does one use data to answer a question geographically? Up until the recent invention of computers, data had to be displayed statically in books. There was no interactivity of the user. This only allowed the user to learn a limited amount from a graphic. But, with the powerful tool of computers we have, we are able to design tools to allow a user to choose what it is they want to know, make comparisons, choose specific time periods, etc. That being said, if a website is not designed effectively and makes it difficult for the user, there is a higher 'bounce rate'. This is when a user visits the main page, but because the website is difficult to navigate, does not explore additional pages of the website. However, if a website is designed effectively and is easy to use, it has been found attract return users and increase retention (revisit rates) [2].

In the world of big data that we now live in, it is extremely easy to get lost in the details and not know and understand how to use that data to our advantage. Scientists are not well enough trained to come up with effective designs to display their data in the most competent way possible [5]. Thus, before we designed the structure of our project, we researched the best methods to do so.

In a research paper by Stephen R. Midway [5], he lays out 10 principles to make an effective visualization:

1. **Diagram First:** We focused on the data first before we started working on the software to display it. We decided to model population.
2. **Use the Right Software:** There are many pre-maid software packages that can be used to display the data. We created ours from scratch using HTML, CSS and Javascript.
3. **Use an Effective Geometry and Show Data:** Within Javascript, we used the D3 library so create a choropleth map of the United States. This way, a user could interact with the webpage and select a specific state and see what is happening in relation to surrounding states.
4. **Colors Always Mean Something:** Because we are modeling population and don't have a second dimension, we kept our colors simple and started out with black.
5. **Include Uncertainty:** The paper suggests including and uncertainty principle so that users are not left questioning the dynamics of the data. Because of the simplicity of our data, we choose not to focus on including it.
6. **Panel, when Possible (Small Multiples):** This is repeating a figure to focus on differences. This is important because it gives us a reference point to compare what has changed. We did this by creating a side panel when a state was selected that displayed additional graphs.
7. **Data and Models Are Different Things:** Data can be displayed in three different ways: raw (scatterplots), summarized (barcharts) and inferential (regression lines). In our visualization tool we used barcharts and choropleth for the state and county data. Additionally for the county data, we used three more charts: linechart, dotted line chart and a scatter plot.
8. **Simple Visuals, Detailed Captions:** Our objective was not to make a complicated visualization tool. We know they will not return to use it if it is not easy to understand. So we erred on the side of simple graphs and detailed explanations (captions) so that no uncertainly was left.
9. **Consider an Infographic:** Figures are similar to infographics in that they both display numerical data graphically. However, infographics include text, images and other data. Infographics were found to be more memorable. This is what we wanted. Thus, we incorporated this in our webpage.
10. **Get an Opinion:** Before finishing our project, we had users without a technical background try out our webpage. Because, if the general public does not find it useful, we failed in being able to create an effective data visualization tool. Thus, we made changes as requested by the opinions we received from outsiders.

These geographical data visualization techniques have begun being implemented in the real world. In a study done by Miriam Machwitz [4] in her publication in "Computers and Electronics in Agriculture", a web application was developed to help farmers track biomass changes with relation to time and space. With the use of this application tool, farmers are able to determine what the correct quantity of fertilizer, water, pesticides, etc. would be needed to optimize the yield of a certain crop in a field [4]. We hope that with further research, our web application will be used to help farmers and ranchers prepare for the influx/decline of people moving to their state/city by tracking how much produce is needed to supply the given population. This data visualization tool would assist in the impediment of food shortages and supply chain issues.

The most important concept that we found repeated among all of our research publications was how powerful interactivity is for the user. In ”Maps as social constructions: power, communication and visualization” by Jeremy W. Crampton [1], it proposed four major principles to create a Map Communication Model (MCM):

1. **There is a clear separation between the cartographer and the user:** If the only person that is able to understand a ”map” is the cartographer that created it, it is virtually useless. Thus, the cartographer should should separate himself/herself from being the role of being the sole user and create the map in such a way that it useful to most, if not all, and not just themselves.

2. **The map is an intermediary between the cartographer and the user:** The user requests certain information and based on the needs and wants of that individual, the map is the way by which needs are made known and satisfied.

3. **The map communicates information to the user from the cartographer:** Depending on what the user wants to know, the map is created in such a way that graphical information to sent to the one using the map from the one who created it.

4. **It is necessary to know the cognitive and psychophysical parameters of the map user's abilities to comprehend, learn and remember information communicated by the map:** Knowing what the user's intentions are and what they are trying to discover from a map should be our highest priority in designing and creating a data visualization tool.

From these papers we decided that interactivity was going to be one of our highest priorities in displaying information. By continually communicating with those that are not associated with this data visualization tool, we will be able to optimize the effectiveness of creating a superb interactive webpage tool.

2.2 Dataset

Our data comes from the Iowa Community Indicators Program of Iowa State University. Iowa State was able to scrape this data from the US Census Bureau and put it into an excel file format. The data is split into two files. The historical data series, which ranges from the years 1969-2019, and data from the last decade which ranges from 2010-2019. The data consists of rows that contain the state/territory name, and the columns which contain the population that year. Each unit represents the number of residents, as of July 1st each year unless otherwise stated.

In addition to population data on the state level, this website provides data on a county-level basis from the years 2010-2019. This data consists of columns that outline the county, state, and population for that year. This data is in a separate excel tab than the data on the state level.

Important notes about the data:

- The resident population includes all persons who are ”usually resident” in a specified geographic area. It excludes U.S. Armed Forces overseas and civilian U.S. citizens whose usual place of residence is outside the United States.

- The April 1, 2010 estimates base may differ from the 2010 Census count due to legal boundary updates, Count Question Resolution actions or other changes.

- The annual estimates are subject to backward revision with each new data release.

- Last Census Bureau update: December, 2020

In addition to the population dataset, we also included a dataset outlining the total size of each state. This dataset includes the state name and geographical area information about each state. All area measurements are displayed as miles squared. One thing to note is we used the land area metric from this dataset rather than the total area to compute population density.

Another dataset we used came from the US census bureau and includes the geographical area information on a county level basis for each county in the United States. All area measurements are given in miles squared in this dataset as well, and we were only interested in total land area.

The last two datasets we used came from the GitHub page of TopoJSON and include the map data for the United States, and the map data for each individual state. The TopoJSON file for the United States is located here and the individual state files are located here.

2.3 Tasks

Following was an outline of the responsibilities that we needed to complete in order to finish this project:

1. **November 8th: Project Proposal/Peer Review**
   We finalized our project proposal, divided the work that needed to be done amongst each other, finalized our schedule, and had a clear direction on how we developed our visualization system.

2. **November 15th: Create backend, complete processing of data, and set up the API to communicate with the backend, complete frontend components not reliant on the data.**

   Because it was difficult to work on this project without our data, we prioritized setting up the backend and getting our data into MongoDB. Also, during this time period, we set up several front-end components that did not rely on the data. This allowed us to work in parallel while we were working on setting up the backend. In this stage we created the map, toggle buttons, slider, and hover/click events.

3. **November 29th: Set up Javascript to get data from the backend, tie the data into our application, and populate the data in the map.**

   During this stage, worked on having the data from the backend show up in each state's region. We also worked on all the API calls to get data from the database to the front-end.

4. **December 6th: Further styling, fixing bugs that will pop up, and adding any of our optional features.**

   During this stage, we committed ourselves to making our visualization pleasing to the eye, and working on our color scales and overall user experience. Additionally, we thoroughly tested our application during this week and fixed any issues that arose. Finally, we added optional features with the little time we had left over.

5. **December 13th: Final Presentation**

   We presented our data visualization tool to our class.

6. **December 17th: Final Code/Video**

   We created a two minute screen-cast with narration showing a demo of our data visualization tool. We also created a Powerpoint presentation to give an overview of our project.

2.4 Scenarios

This tool can be used for people deciding on a new place to relocate. It could help them avoid populated areas, or find populated areas depending on their interests - this would be particularly true with the county map which gives a much better picture than just the country map.

We also took our data visualization tool to Victor Iverson [3], who is in his second term as chair of the Washington County Board of Commissioners in Utah. As head of this board, he oversees the enactment of local ordinances, approves budget plans, manages water usage among other county-level duties. Upon using our data visualization tool, he said the following:

”Something like this would be extremely important to the water district. They’re trying to project growth in the future, water needs in the future, etc. Something like this would be really good. Water is the number one resource that we worry about and depending on how many people are moving here, we need to plan on where we are getting our water. Right now we have a company in Salt Lake that estimates what our populations will be within the next thirty years and they’ve been wrong in the past and it affects us because they’re saying that we may not need the Lake Powell pipeline in thirty years. But of course, they don’t share how they came to that conclusion because it’s private. So something like this could really help us justify why we do access to certain water resources. As far as the visualization, this seems very user friendly compared to some of the charts we see. I think it has a good look to it. It’s nice that you can go in and break things down. It’s hard to put the data in a way to see where you’ve been in order to project where you’re going but this does a lot better job than what I’m accustomed to seeing.” [3]

3 System Description

Following is a description of our data visualization system.

3.1 System Pipeline and Architecture

The pipeline for our project goes as follows: When a user lands on a page the front end makes an API call to request data. Our API call interfaces with the backend to retrieve the data from the database. The database returns the data to our API, and the API returns the data back to the client.

Going into more detail, we have the following pipelines set up for our two pages. For the country page, the user will land on the page, and we make three calls to the API. The first call is to get the path data to draw the map of the United States, the second call is to get the state and the corresponding land area, and lastly, we retrieve the population for each of the states. Once that data is retrieved and sent to the client side we perform data manipulation to convey the correct data. We calculate the total percentage of population each state contains, the population relative to the land mass, and the population increase since 1969. Finally, we draw our SVG elements and pass this data to the visualizations.

For our state page, the user will land on the page and we make three different API calls. The first call is to get the path data to draw the map of the specific state. The second call is to get the population of each county in the state, and the last call is to get the population of that state over time. We then draw our SVG elements and pass this data to the visualizations.
Our system architecture: we use a Mongo backend. We use Express JS to interact with the mongo backend. On the frontend we use HTML, CSS and Javascript. For our HTML we use EJS as a templating engine, and for our CSS we use Bootstrap. A diagram can be seen below:

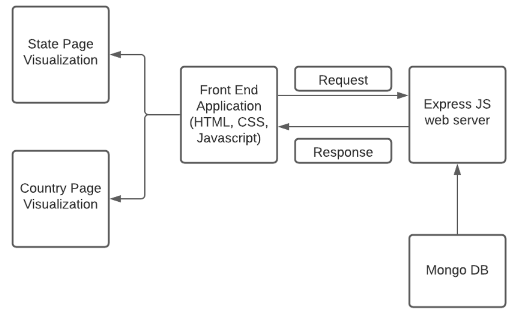

Fig. 2. Shows diagram of System Pipeline

3.2 Data Preprocessing and Analysis

Most of our preprocessing was done directly on the CSV files before they even went into the database. We wrote a JavaScript file called “db-seeder.js” which will read a CSV file and perform some data processing on it and then create a collection in the database and save the data to the database. We did it this way because early on we identified that it would be difficult to stay in sync with the database constantly updating if we had to remember to manually import a different collection every time, we needed to perform a new task. This way, whenever one of us would pull new code down, we would just run this script and the most up to date data/collections were created in our database.

One of the tasks we had to do in this script was combine two CSV files to create one collection. Our collection “states” contains population data for each of the 50 states from 1969 to 2019. To get this data into the database, we had to read two CSV files, one containing data from 1969-2019 and the other containing data from 2010-2019. We then used the name of the state as a key and stored the state in the database with the following document structure:

```
{
   "state": "Utah",
   "years": [
   {
         "year": "1969",
         "population": 1047000,
         ...,
         "year": "2019",
         "population": 3203383
         }
     ]
}
```

Listing 1: Document structure of ”states” collection.

Another section of preprocessing we had to do was on the “counties” collection. Our dataset came with a single column for the county and state. We needed these columns to be separate in order to map the correct county back to the correct state, so we split the first column into two columns and created a new CSV file. We stored the county level data similar to the state level data and the document structure is as follows:

We use the County name as well as the FIPS code to map the data back to the correct county in our GeoJSON data.

Lastly, we had to insert the land area data into the database so we could visualize how densely populated different states/counties are. To do this, for the state data we found a dataset that contained the state name, total area, land area, and water area. We were not interested in total area and water area, we are only looking at the amount of land

```
{
   "state": "Utah",
   "county": "Cache County",
   "fips": "49005",
   "years": [
    {
        "year": "2010",
        "population": 113386
        }, ...
    ]
}
```

Listing 2: Document structure of ”counties” collection.

area in a region. We then took those two values to create the following document structure:

```
{
   "state": "Alaska",
   "square miles": 570641
}
```

Listing 3: Document structure of ”state-land-area” collection.

The county level data was very similar, however we had to include the county name as well as the FIPS code. The document structure of the county level data looks like the following:

```
{
   "county": "Cache",
   "state": "Utah",
   "fips", "49005",
   "square miles": 1164.84
}
```

Listing 4: Document structure of ”county-land-area” collection.

3.3 Visualization

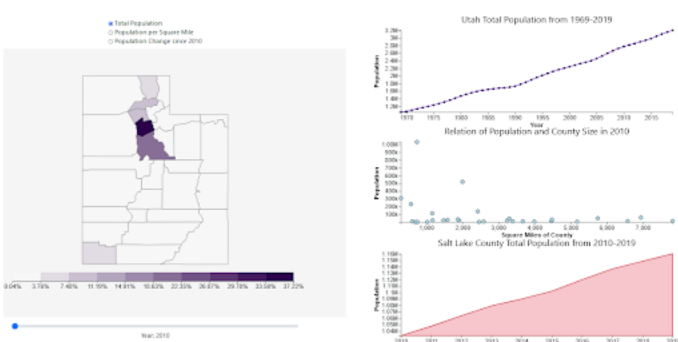

Fig. 3. Shows the State View page

Our final design used a map of the country to display population data for each state based on the year. The year was represented by an interactive slider in which the user could drag to look at years in the range of 1969-2019, as these are the dates for which we were able to obtain population data. Our design displayed a map of the United States with borders between each state, and display the population of each state based on a chosen color display. The higher the population, the darker the shade of the chosen population color. The visualization also included toggles for changing the displayed data in each state to display population with different metrics, such as “the number of square miles per person” or “the percent the population has changed”. Additionally, when a user clicked on a state, the visualization will generate graphs to the side of the main US graph showing the population data for that state, along with a breakdown of population by county for that specific year. Each state also displays a small window of data from the state’s wiki when the state is hovered over.

3.4 Interactions

The following are all the different interactions we created:

3.4.1 Dropdown Menus

For the country view, We created a dropdown menu to choose between the choropleth map and the barchart so they could choose which one they wanted to see. We also created an additional dropdown with the states so that they could choose a state that way in case they forgot where a state was located on the map. In the barchart of the Country View page, an additional dropdown menu was created so that a user could choose if they wanted the bars for the states to be ordered alphabetically or by value.

3.4.2 Toggles

For both the Country and State View page, we created three toggles that switch between total population, population per square mile and population increase since 1969.

3.4.3 Mouse Interactions

For both the Country and State View page, as the mouse hovers over the choropleth map, a popup is provided with data specific to the toggle chosen. On the Country View page, if a state is double clicked, the webpage takes them to that state’s page. On the State View page, if a county is clicked, an area chart for that county appears in third graph on the right-hand side.

3.4.4 Sliders

In the Country View page, this changes the year in the range 1969-2019. In the State View page, this changes the year in the range 2010-2019. As the slider is changed, the color (for the choropleth maps) and arrangement of the states (for the barchart) of the states changes.

3.5 Back-end API

Our database consists of six separate collections and our Back-end API has seven different API calls. The collections in our database are the following:

1. **’counties’:** Contains an entry for each county in the country. Each record in the database contains the county name, state, FIPS code and a list of populations in that county from 2010-2019.

2. **’states’:** Contains an entry for reach state in the country. Each record in the database contains the state name and a list of populations in that state from 1969-2019.

3. **’state-land-area’:** Contains an entry for each state in the country. Each record in the database contains the name of the state and the number of land square miles in that state.

4. **’county-land-area’:** Contains an entry for each county in the county. Each record in the database contains the county name, state name, FIPS code, and the number of land square miles in that county

5. **'map-paths':** This is the JSON data to draw the map of America from the GeoJSON file. We debated on whether or not we should place this data into the database, and we decided it would be easiest if all our data was stored in a single location so we placed this data in the database.

6. **'state-map-paths':** Contains an entry for each state in the county. Each record has the state name and the JSON from the GeoJSON file to draw that specific state.

Our back-end API has seven different calls that do the following:

1. **/topology:** Returns the data from the "map-paths" collection. This allows us to draw the map of the United States.

2. **/topology/:state:** Note, we use :state here to represent a variable. This path could contain any string as the path such as /topology/-Texas, /topology/Utah, etc. This method returns the specific state from the "state-map-paths" collection. This allows us to draw the individual state map.

3. **/population:** This method returns the population of the United States broken up by state. The result of this method will be an array of 50 entries containing population data from 1969-2019.

4. **/population/:state:** This method only returns population data for a specific state. This is useful for our page focusing on a single state because there is not a need to get the data from the other states

5. **/counties/:state:** This method returns a list of each county in a particular state and the population in that county for each year from 2010-2019.

6. **/state-land-area:** This method returns a list of each state and the corresponding land area in square miles of each state. We use this data to calculate the population density of each state.

7. **/county-land-area/:state:** This method returns a list of each county in a particular state and the land area of that county.

4 Use Cases

In the first scenario presented in part 2.4, if a user knew which state they wanted to move to, they click on the state in the country view and then the state specific graphs are displayed. If they want to move to the county that had the lowest population growth, they toggle "Population Increase Since 2010" and search for the counties that have the lightest color.

Mr. Iverson frequently has meetings at the capitol to discuss water shortages with the governor for his county. He could take this tool with him to show the governor how much his county has grown each year since 2010 to secure water rights for his county. Additional funding could be requested to help manage situations similar. He can do this by selecting the state of Utah and then Washington County in the State View page. As he uses the slider he can show how much the population has increased in Washington County. He can use the toggles to switch between totals and percentages of the population and look at the graphs on the right to see the rates of change over the years.

Describe and present at least two use cases based on the proposed scenarios where using your visual analytical system you discover some insight from the dataset.

5 Discussion

Some major insights we had were we thought it was most interesting looking at the population increase since 1969. The map view really helped us understand that there has been a population explosion in the past 50 years in the Western States. Primarily Nevada, Arizona, Utah, and Colorado. It is beneficial how we have the option to look at the bar chart view and the map view. With the bar chart view, we can easily understand just how much the population has grown. However, with the map view we get to see that cluster of growth in the West that gives us an insight we would not have with just a bar chart.

It is actually really fascinating to see how a state can be so highly populated but have it be limited to specific counties. California in particular is the highest populated state in the US and it comes from only about three counties.

Some of the problems that we encountered were instead of referencing counties by their name, we referenced them by their GEOID. This is because in states like Louisiana have parishes, not counties. Additionally, Virginia had many cities that were classified as counties but ended with the name "City". This took us some time to figure out.

While creating the tool seemed to go fairly well, another tedious problem we had was towards the beginning we had to do data preprocessing on each of the files to get them into the correct format to be loaded into the database. We also had to get the TopoJSON files for not only the map of the United States, but each of the 50 states. This ended up being very tedious because we had to download each of these files and rename them to match the correct state

Our main limitation was the data we had. Our main data set only contained data on raw population totals, so we were not able to see why the population was changing or possible implications that may have. We also did not have as much county data as country data. So, the county year slider is limited to 2010-2019. Lastly, we cannot go any deeper than county level, which could be considered a "limitation" if you want to dive deeper into individual cities.

6 Conclusion

Using HTML, CSS, Javascript, D3 and Mongo, we created a data visualization tool that allowed us to see how populations change among states and counties of the United States from the years 1969-2019. We believe that the availability of user interactivity is what really made our project particularly advantageous to everyday users. A limitation that we found is the lack of population data for cities in each county. We believe that with additional research, this limitation could be resolved with more detailed datasets so that the user can go one dimension deeper than just the county.

References

[1] J. W. Crampton. Maps as social constructions: power, communication and visualization. Progress in Human Geography, June 2001.

[2] R. Garett, J. Chiu, L. Zhang, and D. S. D. Young. A literature review: Website design and user engagement. PMC: US National Library of Medicine National Institutes of Health, July 2017.

[3] V. Iverson. Chair of washington county board of commissioners. Personal contact, November 2021.

[4] M. Machwitz, E. Hass, J. Junk, T. Udelhoven, and M. Schlerf. Cropgis – a web application for the spatial and temporal visualization of past, present and future crop biomass development. Computers and Electronics in Agriculture, June 2019.

[5] S. R. Midway. Principles of effective data visualization. Patterns: Volume 1, Issue 9, December 2020.

[6] D. Pimentel, S. Cooperstein, H. Randell, D. Filiberto, S. Sorrentino, B. Kaye, C. Nicklin, J.Yagi, J. Brian, J. O'Hern, A. Habas, and C. Weinstein. Ecology of increasing diseases: Population growth and environmental degradation. Springer Science, July 2007.

[7] A. Singleton and D. Arribas-Bel. Geographic data science. Geographical Analysis, 2021.

7 Appendix

Team members contribution: Initial project idea and data files came from KC Gubler. Carson Sorenson created the backend to get the project set up. Taylor Dawson helped with the front end. KC Gubler helped with styling and focused on the final paper.

US Population Analysis GitHub repository URL