**Lesson 8: Independent exploration of FOSS**

**Amy Farley**

**GEOG 585**

One of the guest speakers at the [2017 TNRIS GIS Forum](https://tnris.org/texas-gis-forum/2017/agenda/#kyle-walker) made a statement during his presentation to the effect that not ALL data belongs on a map. (Walker, n.d.) This perspective resonated with me. The presenter was [Kyle Walker](http://personal.tcu.edu/kylewalker/index.html), a professor and researcher at Texas Christian University. Most of his research revolves around immigration and urban demographics, but he has spent considerable time researching and developing software tools for data acquisition and visualization. During his presentation, he was demonstrating that sometimes data is actually easier to understand when presented as a standard chart. I remember thinking about thematic maps, specifically choropleth maps. I truly appreciate how these types of maps immediately tell an overall story to the viewer about some idea or topic. They are brilliant ways of providing the overall picture or general trend. What is less easy to immediately understand from these types of maps are actual statistics. For example, when viewing a map of the entire country, it is more difficult to immediately pick out the “Top 10” areas of a particular theme or to see a ranking of the areas based on the theme values. This is where basic bar charts and graphs can truly add insight into the mapped data. This is exactly how Mr. Walker presents his project work. There is generally a map along with additional data visualizations in the form of graphs or charts.

My term project involves layering cropland data over a map of Texas counties and allowing viewers to slide through different years. The cropland raster data immediately tells a story about which areas of Texas have more cultivated land, but it not clear which counties have the most land area “allocated” to cultivation. It is also difficult to tell exactly what percentage of the land area is used for crops and where the counties rank in this normalization. A goal for this project (and this class) was to research the D3.js Javascript library so that I could dynamically produce charts to accompany my web map. My goal is to be able to display a chart of Texas counties ranking them by percentage of land allocated to cultivation of crops. As the user moves through each year, the chart will automatically update with the data for the selected year. Lesson 8 provided a great opportunity to really dive into this library. The D3 library allows the developer to programmatically generate content on a web page based on the data provided. It can create, update, or remove standard HTML elements of a page based on user actions and underlying data. This is all accomplished client-side so there is no page refresh resulting from a server call.

I took advantage of several different resources. The starting point was obviously the D3 website <https://d3js.org/>. This website simply describes the library as “a JavaScript library for manipulating documents based on data. **D3** helps you bring data to life using HTML, SVG, and CSS.” (Bostock, D3 Data-Driven Documents, 2017) It provides the basics for getting started. The GitHub repository <https://github.com/d3> supplies links to the API reference, examples, and a visual gallery of the many visualizations that can be achieved with the library. The BEST resource I found was actually a collection of YouTube videos produced back in 2013 by a Swedish man that is identified simply as d3Vienno. Each video was concise and built from a previous video. The examples were relevant and easy to apply to my term project. I referred to these videos the most while working with my own web map. The link to this collection is:

* <https://www.youtube.com/user/d3Vienno/videos>.

Of course, these videos were a bit outdated, and I quickly realized that the more modern releases of this library had significant changes to the API. This is especially true in regards to loading external data as I was trying to do. Back in the GitHub repository, that list of changes and a major change surrounding the use of Javascript Promises instead of asynchronous calls for data loading was explained in the CHANGES.md of the repository found here:

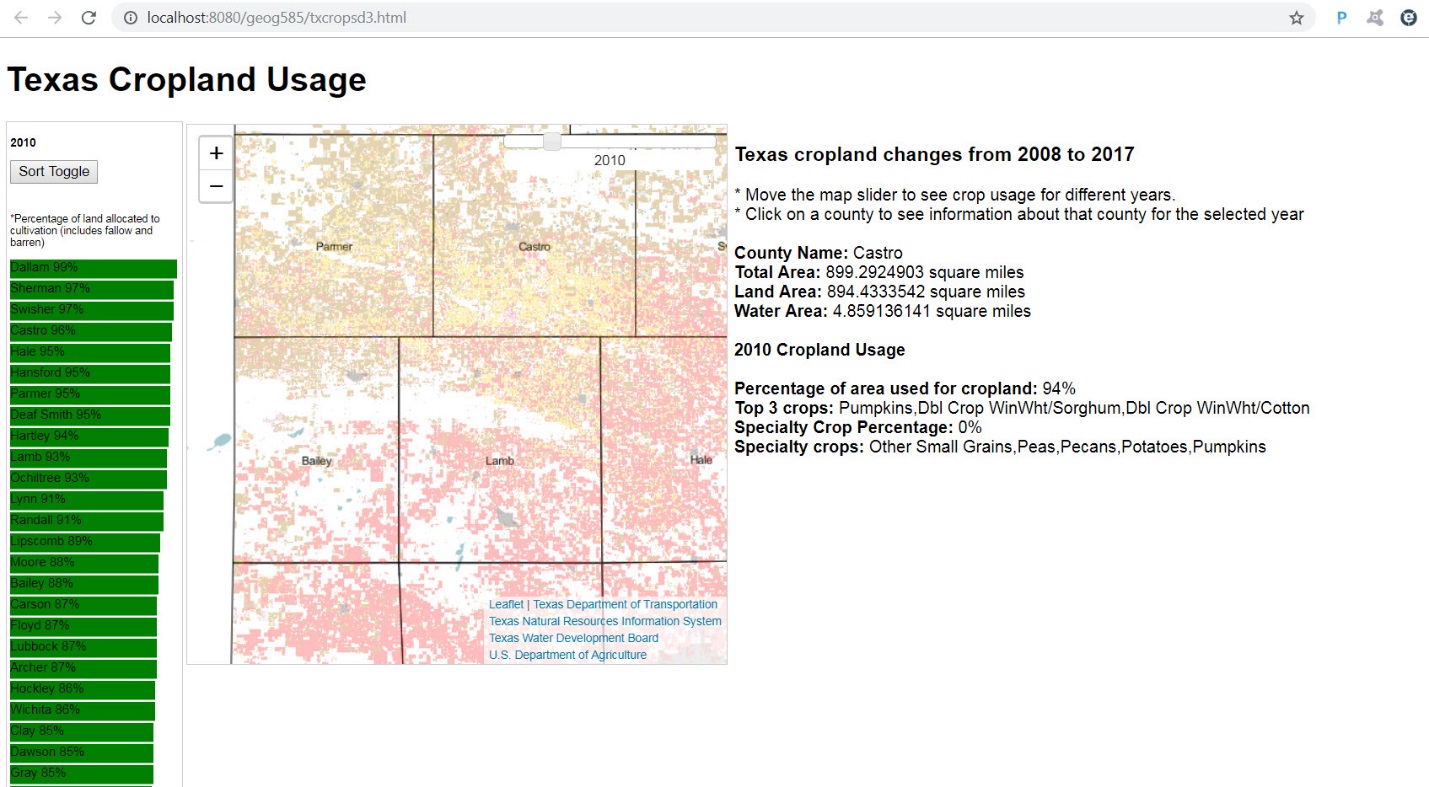
* <https://github.com/d3/d3/blob/master/CHANGES.md#changes-in-d3-50>

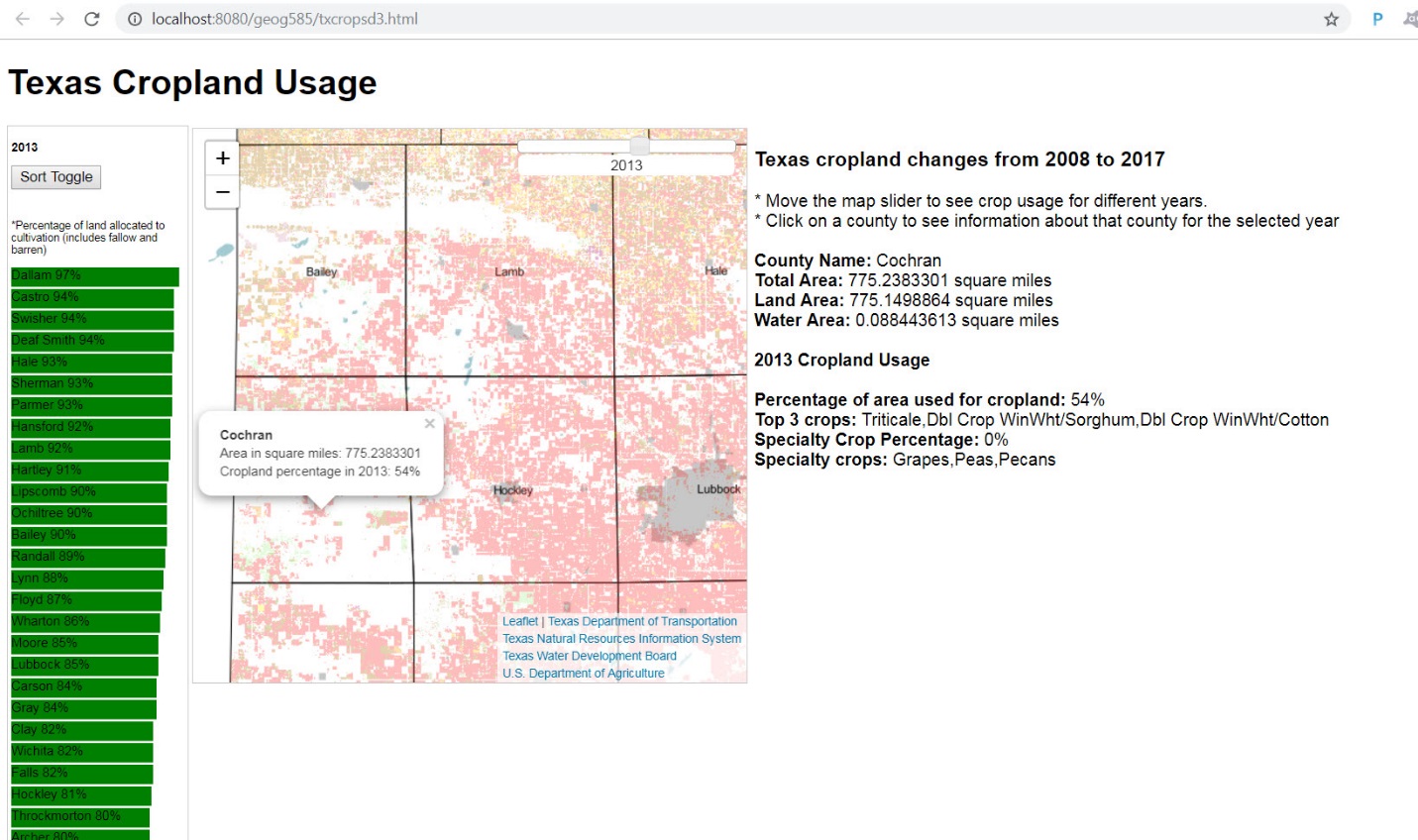
Additional resources included the reference documentation found on the Tutorials Point website <https://www.tutorialspoint.com/d3js/index.htm>. I have used this website many times before for various coding exercises. TutorialTeachers.com had good reference information for data loading in D3 as well at <https://www.tutorialsteacher.com/d3js>.

Last, but certainly not least, Stack Overflow was my resource for all the tedious, not-so-obvious tasks I wanted to accomplish. In some cases, the answer was in the forum post, in other cases, the post led me to another website. Below are a few of the questions and links to the answers or examples I used as a reference.

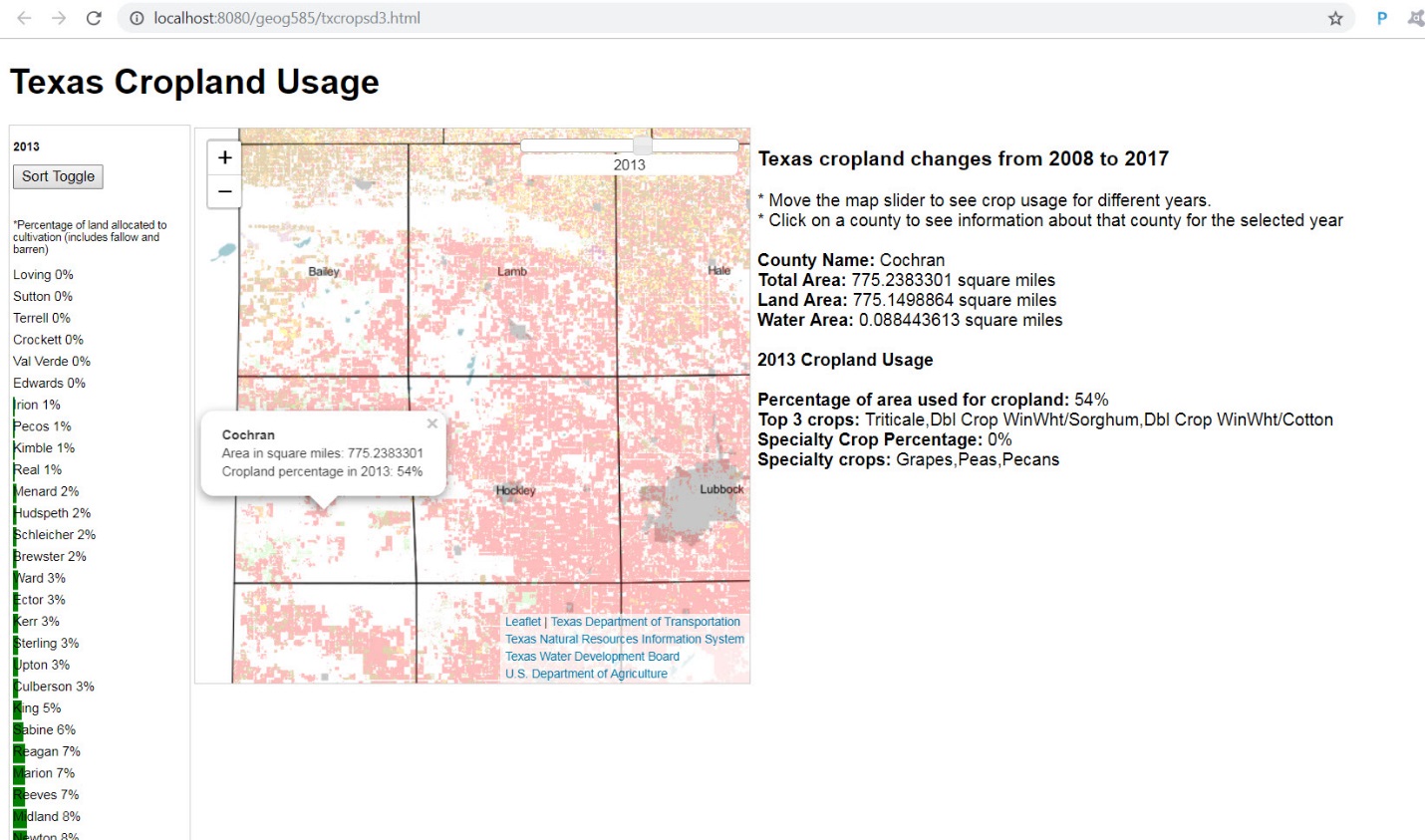
* *How do I clear the D3 content as the user slides to a different year?*
  + <https://stackoverflow.com/questions/10784018/how-can-i-remove-or-replace-svg-content>
* *How do I sort the external data upon each selection?*
  + <https://stackoverflow.com/questions/25175299/d3-csv-import-sort-function-display>
  + <https://stackoverflow.com/questions/29945181/sorting-large-csv-in-d3-js>
  + <http://plnkr.co/edit/z8WdSry2349xassr2mtj?p=preview>
  + <https://bl.ocks.org/anonymous/bc5a9691a3417b403d4e8ade3297afa3/3a2434c1c2849e476791e581754ec27e055db4d6>
* How do I format the text elements attached to my chart “bars”?
  + <https://www.dashingd3js.com/svg-text-element>

I am pleased with how I was able to incorporate this into my web map project and produce the chart visualizations to accompany my map. It took some tinkering with my slider control code to call some new functions for displaying the data, but I was able to dynamically update the chart as the user selected a different year from the map slider control. [Figure 1](#Fig1) and [Figure 2](#Fig2) show the chart in the far-left column updating for each year.

  
**Figure 1:** Screenshot of term project web map with D3 chart accompanying the map data

   
**Figure 2:** Screenshot illustration of D3 chart data update as year selection from slider control moves

[Figure 3](#Fig3) is a screenshot of the Sort Toggle button that toggles the list sort between descending and ascending.

  
**Figure 3:** Screenshot illustration of D3 data sort toggle

The JavaScript code I added to the HTML page to dynamically display the bar chart is displayed in [Figure 4](#Fig4). The entire page can be found in my GitHub repository:

* <https://github.com/atxFarley/GEOG585/blob/master/termProject/code/txcropsd3.html>
* <https://github.com/atxFarley/GEOG585/blob/master/termProject/code/SliderControl.js>

|  |
| --- |
| var dataset;  var sortOrder = "desc";  var selectedYear = "2017";  loadYearData(selectedYear);  function sort() {  console.log("sort(): " + sortOrder);  try {  chart(dataset);  } catch (e) {  console.error("Error in dataset sort toggle: " + e);  }  }  function loadYearData(year) {  selectedYear = year;  sortOrder = "desc";  console.log("loadYearData(year: " + year + ")");  try {  console.log("before loading data: ");  d3.csv("countyData.csv")  .then(data => chart(data))  .catch(function (error) {  console.error("Exception caught getting csv data: " + error);  });  } catch (e) {  console.error("Error caught loading data for D3 transformation : " + e);  }  }  function chart(data) {  console.log("inside chart(): ");  dataset = data;  console.log("data size: " + data.length);  console.log("year: " + selectedYear);  var cropPercentFieldName = "cropct" + selectedYear;  console.log("sortOrder: " + sortOrder);  $("#dataDiv").html('<p style="font-size: 11px; font-weight: bold;">' + selectedYear + '</p><button id="sort" onclick="sort();">Sort Toggle</button><br/><br/><p style="font-size: 10px">\*Percentage of land allocated to cultivation (includes fallow and barren)</p>');  try {  d3.select("svg").remove();  } catch (e) {  console.error("Error removing svg: " + e);  }  try {  if (sortOrder === "desc") {  console.log("perform descending sort...");  data.sort(function (a, b) {  return d3.descending(Math.round(a[cropPercentFieldName] \* 100), Math.round(b[cropPercentFieldName] \* 100))  });  } else {  console.log("perform ascending sort...");  data.sort(function (a, b) {  return d3.ascending(Math.round(a[cropPercentFieldName] \* 100), Math.round(b[cropPercentFieldName] \* 100))  });  }  //toggle the sort variable after actual sorting  if (sortOrder === "desc") {  sortOrder = "asc";  } else {  sortOrder = "desc";  }  } catch (e) {  console.error("error caught sorting: " + e);  }  try {  var width = 160;  var height = 6500;  var widthScale = d3.scaleLinear()  .domain([0, 100])  .range([0, width]);  //console.log("set widthScale: " + widthScale);  var canvas = d3.select("#dataDiv").append("svg")  .attr("width", width)  .attr("height", height)  console.log("canvas created");  canvas.selectAll("rect")  .data(data)  .enter()  .append("rect")  .attr("width", function (d) {  // console.log("d[cropPercentFieldName]: " + d[cropPercentFieldName]);  var cropPercent = Math.round(d[cropPercentFieldName] \* 100);  return widthScale(cropPercent);  })  .attr("height", 18)  .attr("y", function (d, i) {  return i \* 20;  })  .attr("fill", "green")  console.log("added rectangles to canvas");  canvas.selectAll("text")  .data(data)  .enter()  .append("text")  .attr("y", function (d, i) {  return i \* 20 + 11;  })  .attr("fill", "black")  .attr("font-family", "sans-serif")  .attr("font-size", "12px")  .text(function (d) {  //console.log(d["COUNTY"] + " " + d[cropPercentFieldName]);  return (d["COUNTY"] + " " + Math.round(d[cropPercentFieldName] \* 100)) + "%";  })  console.log("added text to rectangles");  } catch (e) {  console.error("Error caught creating chart : " + e);  }  } |

**Figure 4:** JavaScript code using D3 library to create chart visualization from external data

D3 is especially relevant to the Lesson 8 title *Going beyond “dots on a map”* because it introduces different data visualizations that can provide further insight into data displayed on a map. Of course, D3 actually has mapping functionality that CAN create maps from data as well. The d3-path, d3-geo, and d3-geo-projection projects can convert coordinate data into screen coordinates ultimately creating a map. (Bostock, d3/d3 Gallery, 2018) It would be interesting to explore this more to see how useful it really is versus simply using Leaflet with GeoJSON data. I will admit, just wrapping my head around the basic chart creation was heavy. Understanding how D3 converts GeoJSON to SVG path data would take some time. For now, I think it is important to realize that mapping truly does go beyond the dots on a map and even the map itself. Sometimes, it is easier to understand data presented in a different graphical format. What is intriguing is that using the same external data file, I can, theoretically, add a layer to a map, provide feature interactivity on my map and/or HTML page, as well as create complementary charts and graphs of the data all on the same web page. The robustness of the modern web map is only limited by ideas and understanding of available technology.

# References

Bostock, M. (2017). *D3 Data-Driven Documents*. Retrieved from D3 Data-Driven Documents: https://d3js.org/

Bostock, M. (2018, October 27). *d3/d3 Gallery*. Retrieved from GitHub: https://github.com/d3/d3/wiki/Gallery

Quinn, S. (n.d.). *GEOG 585: Open Web Mapping*. Retrieved 2019, from John A. Dutton e-Education Institute, College of Earth and Mineral Sciences, The Pennsylvania State University: https://www.e-education.psu.edu/geog585

Steiniger\*, S., & Hunter, A. J. (2013). The 2012 free and open source GIS software map – A guide to facilitate research, development, and adoption. *Computers, Environment and Urban Systems*(39), 136-150. Retrieved from https://www.journals.elsevier.com/computers-environment-and-urban-systems

Walker, K. (n.d.). *Kyle E. Walker, Ph.D.* Retrieved from Kyle E. Walker, Ph.D.: http://personal.tcu.edu/kylewalker/index.html