Team CAL-Fire Berkeley Ext. Project 2 2020

Mark Sutton
Cruz Flores
Andrew Lamour
Kwasi Ansu
Brian Brown

Overview

In an effort to provide a more detailed view of the forest fires in the state of California the group has created an interactive map supplemented by graphs detailing the the extent of the fires.

https://cruzaflo.github.io/cal_fire/

Overview of Code Functions

The mapping and graphical content of the site was created using the following steps.

- JavaScript utilizing D3.json library was used to request the data from the CAL Fire API
- Mapbox was used to create a map of the state of California
- Leaflet.js library is utilized to label points on the map and to create custom icons and markers.
- Three graphs are created using Python to retrieve the data
- The Pandas library within Python filters and organizes the data
- Plotly is used to plot two graphical visualizations
- Charts.js is used to create a bar chart
- Exported from Jupyter Notebook the acreage information from fires 2016-2020 into a sqlite database called Fires 2016-2020.db in a table called FIRES
- Index.html is served via a Flask app
- Site is hosted at Here
- Files are stored on Github

Process Flow

Challenges/Roadblocks

A major roadblock in our development of the maps was the CORS Policy restrictions.

The way this error was overcome was to add the following lines of code to reroute data to a third party server to avoid the CORS policy restriction.

```
const proxyurl = "https://cors-anywhere.herokuapp.com/";
const APIurl =
"https://www.fire.ca.gov/umbraco/api/IncidentApi/GeoJsonList?inactive=fa
lse";
var queryJSON = proxyurl + APIurl
```

Code

```
//roadblock #1 solution. This will bypass the CORS access restriction for our API url
const proxyurl = "https://cors-anywhere.herokuapp.com/";
const APIurl = "https://www.fire.ca.gov/umbraco/api/IncidentApi/GeoJsonList?inactive=false";
//per the soltuon, we must combine the proxy URL and our API url
var queryJSON = proxyurl + APIurl

//define function for adding popups to each feature/marker.
function onEachFeature (feature, layer){

    //define variables for each popup
    var fireName = feature.properties.Name
    var county = feature.properties.County
    var startDate = feature.properties.Started.split("T")[0]
    var extinguishDate = feature.properties.ExtinguishedDateOnly
    var acresBurned = feature.properties.AcresBurned
    var fireInfoUrl = feature.properties.Url
```

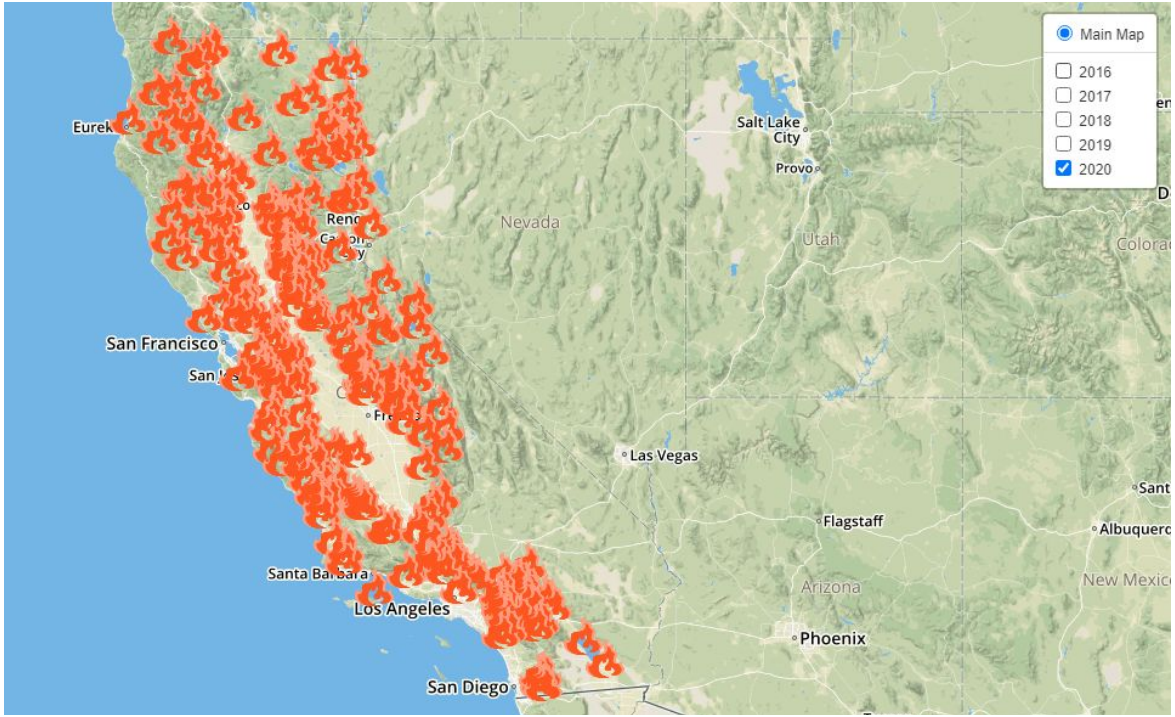
Code

```
//Use Leaflet's geoJSON bindPopup function to add HTML to popups
    layer.bindPopup(
        `<h3>Fire Name: ${fireName}</h3><hr>
        <p>County: ${county}</p>
        <p>Start Date: ${startDate}</p>
        <p>Extinguished Date: ${extinguishDate}</p>
        <p>Acres: ${acresBurned}</p>
        <a href="${fireInfoUrl}" target="_blank">More Information</a>

//load the queryJSON url(proxy url + our API URL)
d3.json(queryJSON, function(data){
    console.log(data)
    var overlayMaps = {}
    var years = [2016, 2017, 2018, 2019, 2020]
```

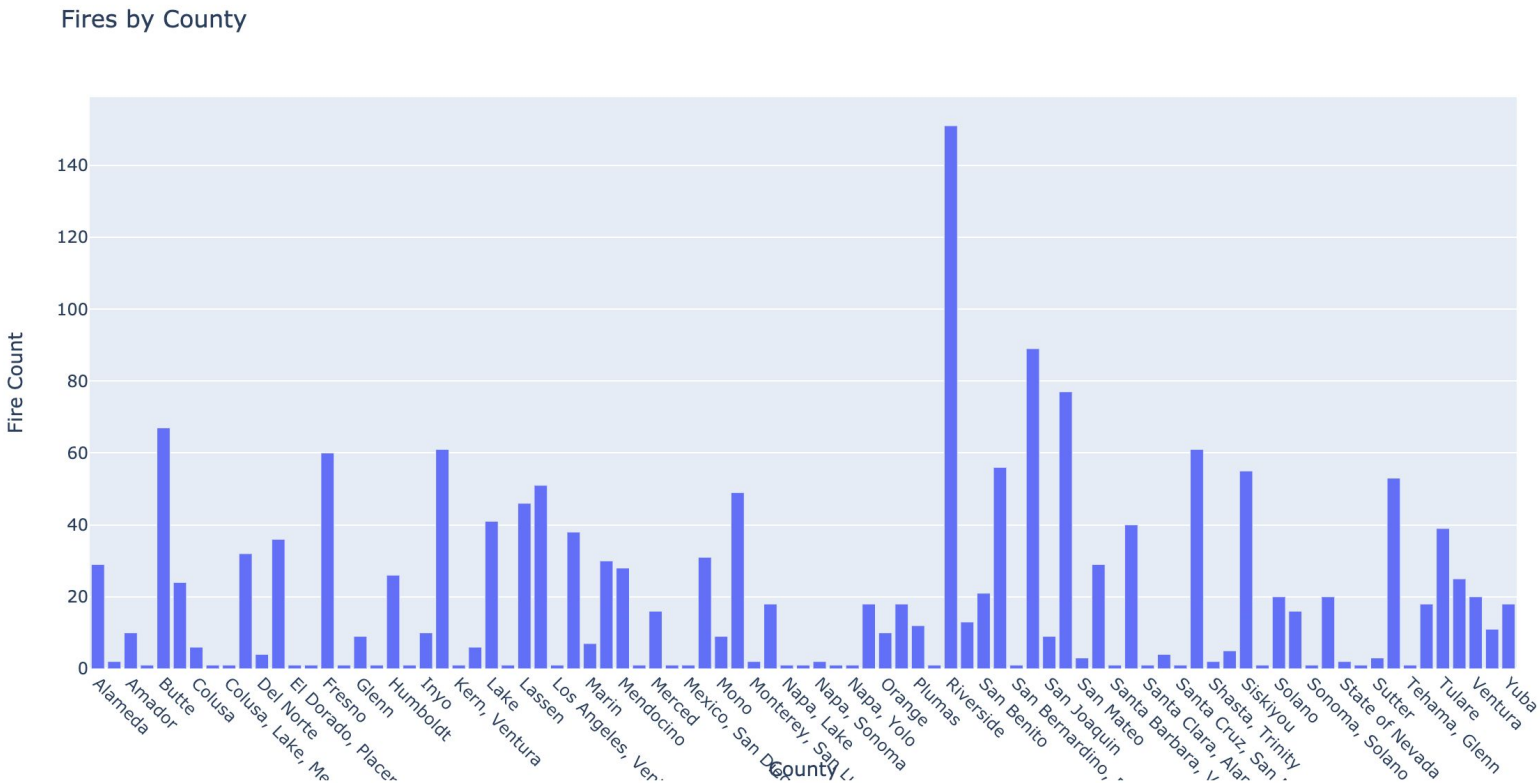
Code

```
// define the layerGroup variable
        var layerGroup = L.geoJSON(data.features, {
            //use the filterByYear and onEachFeature functions that we previously defined.
            filter: filterByYear,
            pointToLayer: customIcon,
            onEachFeature: onEachFeature

//Create map adding the street map and the 2020 features as default layers
    var myMap = L.map("map", {
        center: [36.7783, -119.4179],
        zoom: 6,
        //default layers
        layers: [
            streetMap,
            overlayMaps["2020"]
        ]
```

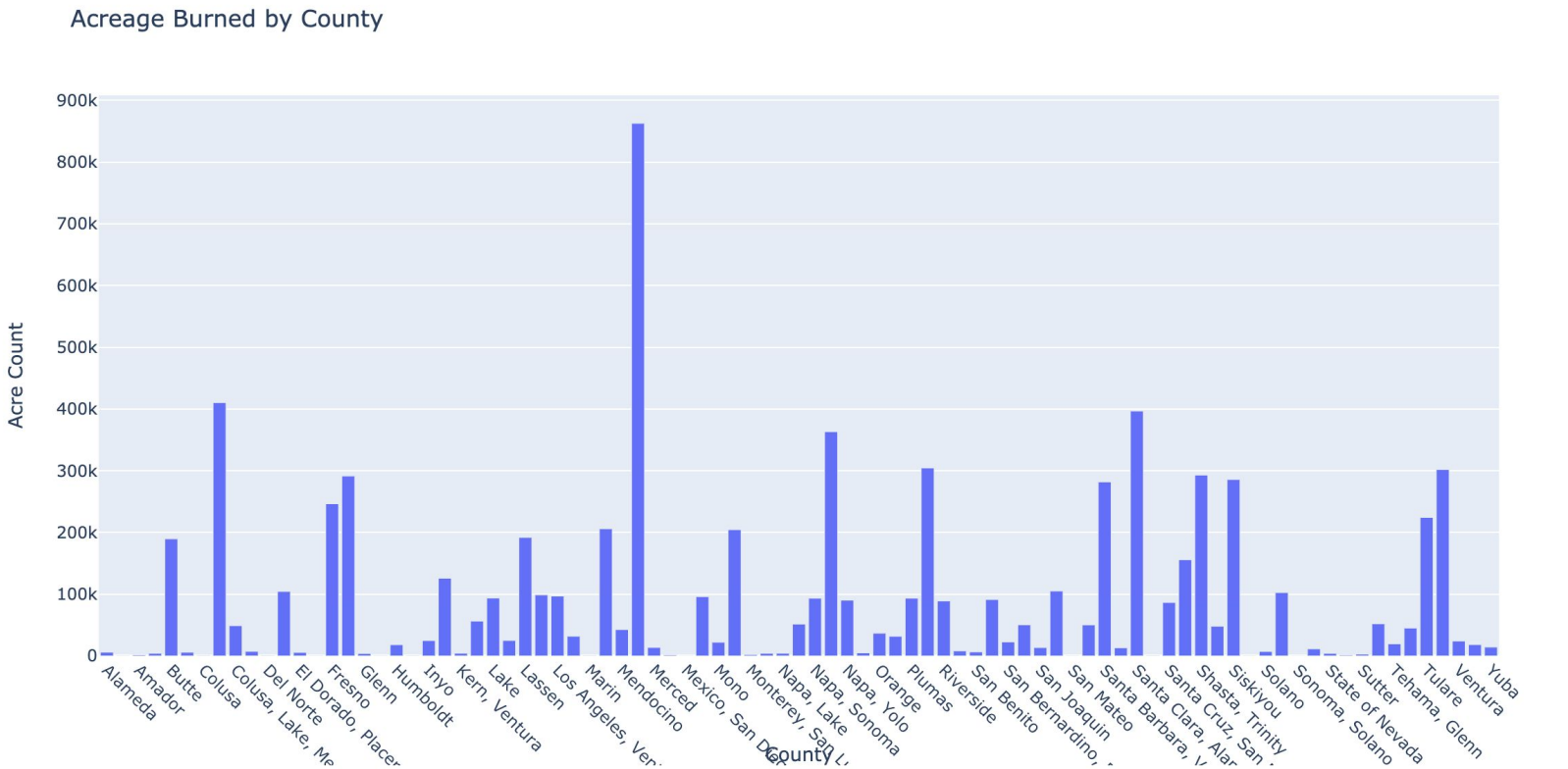
Visualization 1 - Interactive Map



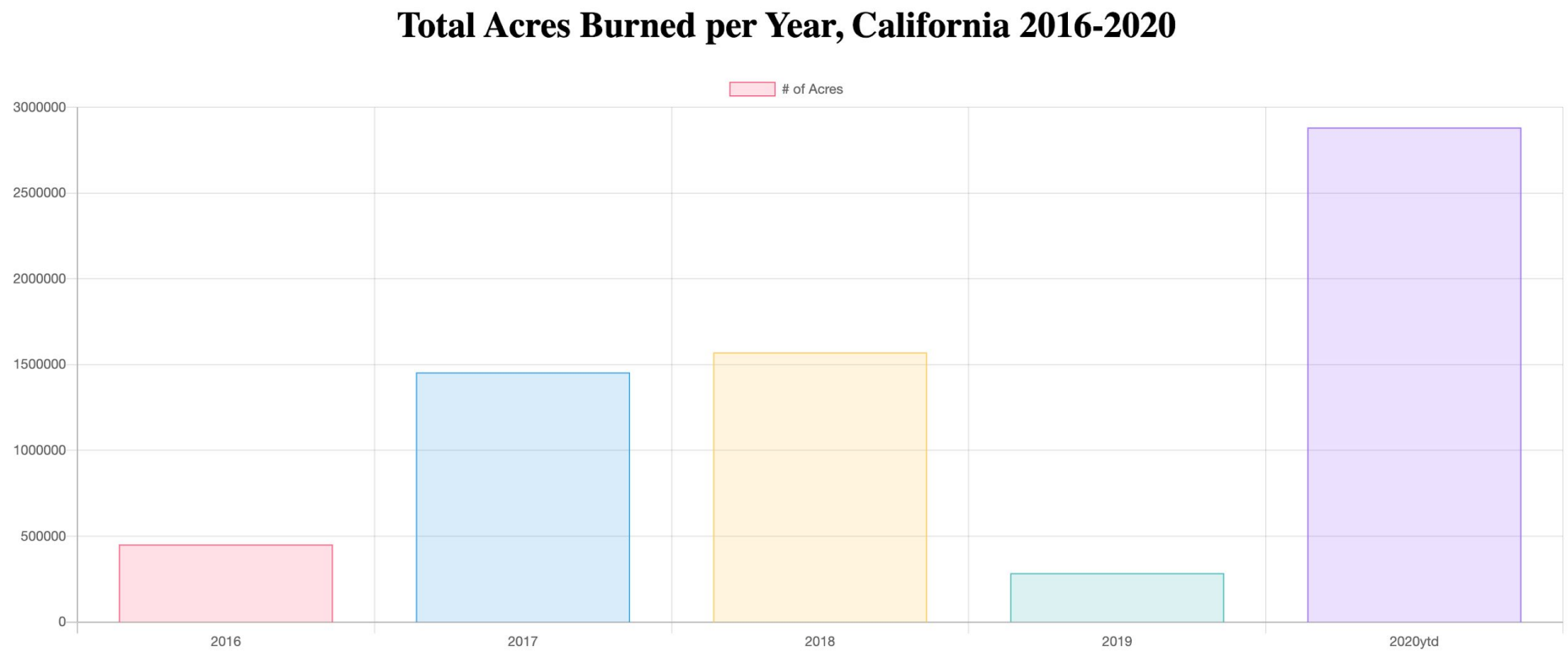
Visualization 2 - Plotly



Visualization 3 - Plotly



Visualization 4 - Charts.js



Insights and Observations from the Data

- There were a large number of fires each year
- Especially when the years are combined, the map gets cluttered
- The icons do not differentiate by size of fire, large fires are noted the same as small fires
- Some fires span across multiple counties and the csv reflects this by creating a unique category combining bordering counties; this obscures the the actual number of fires per county
- Causes of the fire are listed in a separate data set
- CAL Fire provided unique I.D.s for each fire which made identification easier

What Could Be Next

Ideas for further analysis:

- Adjust the size of the icon based on the acreage of fires
- Combine with other data sets to determine causes of fires
- Perimeter outlines could be included
- Include more years and look for long term trends in the data
- Further clean the CSV to breakout fires into individual counties

Questions?

Thank you