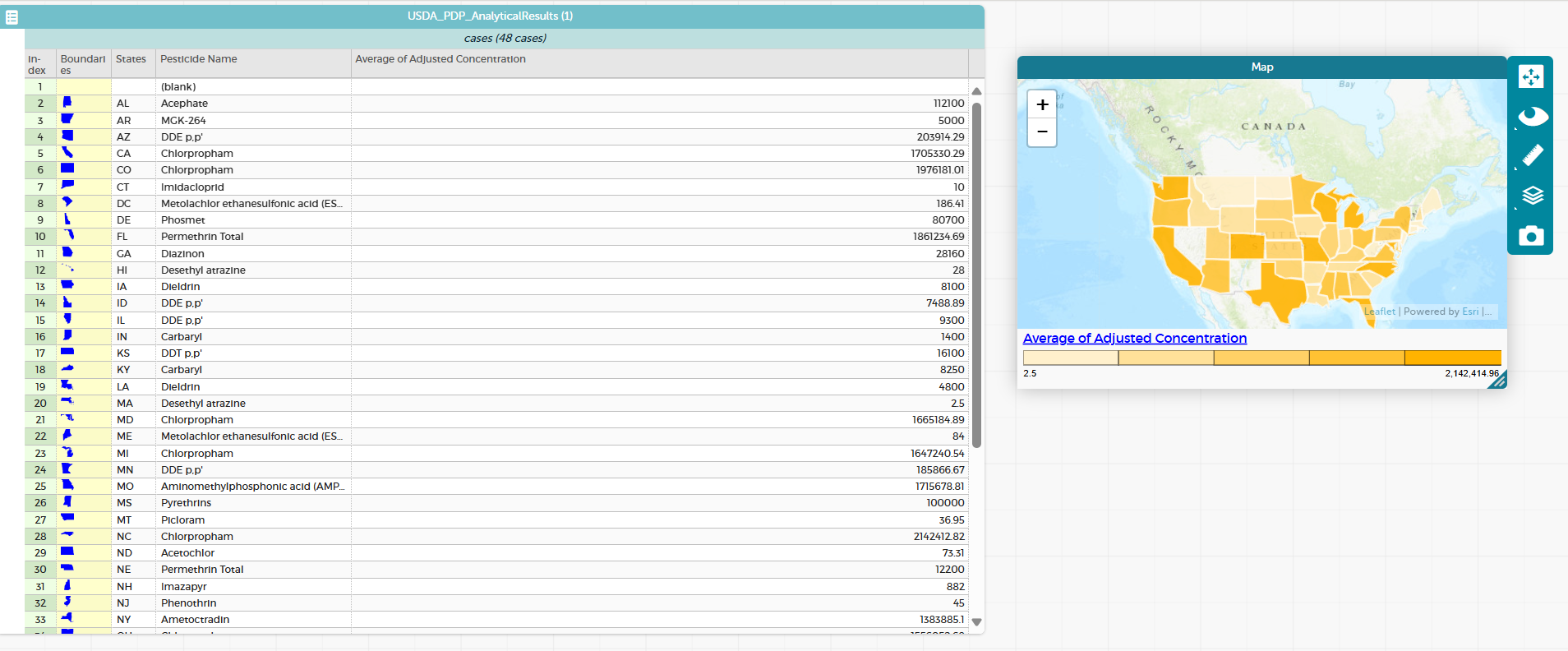
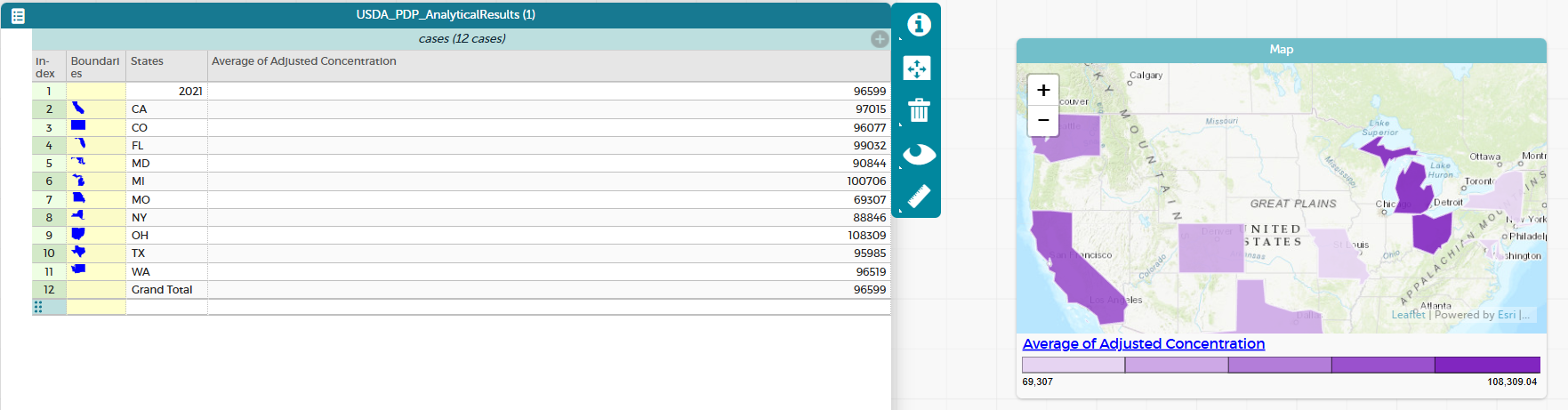


* The Project Starts off by opening the CSV file listed under the USB Drive and downloading it and opening it in an excel sheet
* I did not change the formatting into an excel sheet format so that it would not mess with the overall formatting of the datasets
* Under the “Datathon Dictionary” pdf, its stated the meaning of the numbers found within the Sample ID where it represented years, months and, date as well as the two letters of the state.
* I used the “insert” function to create a separate file where I filled it with separate attributes for States, Years, Months, Date respectively.
* After separating them, I used “filter” to categorize each change in the attributes
* After all of that I did the pivot table function where we reorganized the csv file with only the states, years, and pesticides as well as the average amount of pesticides that is leading for each state as it also factors in the potency of each pesticide as much as possible.



* After creating a new csv file, we go to an app called “Codeap” to upload the csv file and create the table above with the states, pesticides and average amounts of concentrations.
* I created a boundary function and edited its formula so that it can represent the state borders while highlighting the concentration level in each state.



* After the average, I created an average of pesticide amounts for years 2018-2019, in this example I followed the same methods I followed above to create my average pesticides amount overall and created this map representing the average for the year 2018.

import requests

from bs4 import BeautifulSoup

import csv

import matplotlib.pyplot as plt

import pandas as pd

url = "https://journeynorth.org/sightings/querylist.html?season=fall&map=monarch-adult-fall&year=2018&submit=View+Data"

response = requests.get(url)

if response.status\_code == 200:

# Parse the HTML content

soup = BeautifulSoup(response.content, 'html.parser')

table = soup.find('table') # Adjust this if necessary

# Extract the rows from the table

rows = table.find\_all('tr')

# Collect data in a list for sorting

data\_list = []

# Write data rows

for row in rows[1:]:

columns = row.find\_all('td')

if len(columns) > 6: # Check if there are enough columns

state = columns[3].text.strip() # Adjust based on actual table structure

try:

eggs = int(columns[6].text.strip()) # Convert to integer

data\_list.append([state, eggs])

except ValueError:

continue # Skip rows with invalid number formats

# Sort data by state

data\_list.sort(*key*=*lambda* *x*: *x*[0]) # Sort by state name

# Open a CSV file to write the sorted data

with open('monarchadultsighted2018.csv', 'w', *newline*='', *encoding*='utf-8') as csvfile:

writer = csv.writer(csvfile)

# Write the header

headers = ["State/Province", "Number of Monarch Adult Sighted"]

writer.writerow(headers)

# Write sorted data rows

writer.writerows(data\_list)

print("Data has been successfully scraped and saved to 'monarchadultsighted2018.csv'.")

else:

print(*f*"Failed to retrieve the page. Status code: {response.status\_code}")

data = pd.read\_csv('monarchadultsighted2018.csv')

plt.figure(*figsize*=(12, 6))

plt.bar(data['State/Province'], data['Number of Monarch Adult Sighted'], *color*='blue')

plt.xlabel("States")

plt.ylabel("Total Number of Monarch Adult Sighted")

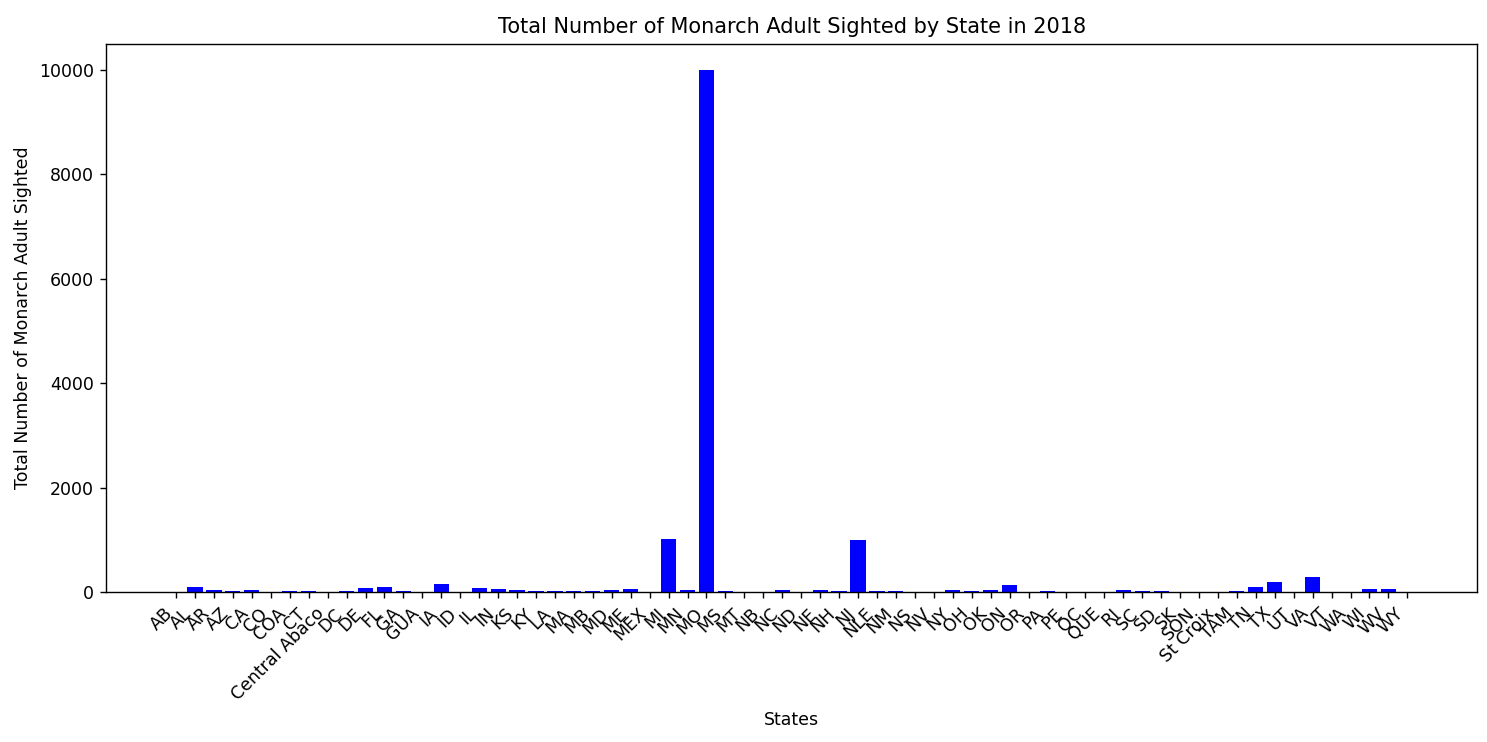
plt.title('Total Number of Monarch Adult Sighted by State in 2018')

plt.xticks(*rotation*=45, *ha*='right') # Rotate x labels for better visibility

plt.tight\_layout() # Adjust layout to prevent clipping of tick-labels

plt.show()

* After finishing the second criteria in the beginner track I start working on extracting the data for monarch butterflies from a website called “Journey North”
* This code extracts the data from that website and creates a separate csv file automatically where it then parses through that file and creates the labels and layouts that I coded in the bottom of my code



* This is what it looks like afterwards.