CMSE 201 Final Project

Ariel Ooms

CMSE 201 - 006

December 6, 2024

COVID-19: 2020 and Beyond

Background and Motivation

COVID-19 played a big role in my life as it did for millions in the United States and worldwide. I wanted to investigate just how quickly the disease spread in its first year, 2020, and how the vaccines were distributed once they were ready for public use. I wanted to see if I could identify any trends in how quickly and where the disease spread. I also wanted to focus on where it affected me which is why I chose to use data only from the United States and focused some of my efforts on the pandemic's spread in Michigan. My main research questions are:

1. How did COVID-19 spread by state, region, and by month in the United States?
2. Which counties in Michigan had the most overall reported cases? Did GDP or population play a role?
3. Which states were the first to rollout vaccine distribution?
4. Which brand was distributed most?
5. How did each brand distribution change over time?

Methodology

I used two datasets to investigate my questions. The first is a dataset from the World Health Organization that includes dates in 2020 as well as the county and state cases were reported in, and how many cases were reported on a specific day. I used this dataset to observe trends in how the disease spread by state, region, and county. I used additional data for the populations of each of Michigan's counties and GDP. The second dataset I used contained all the information on vaccine distribution. The data was provided by the Center for Disease Control and Prevention. I used this data to look at the distribution of vaccines overall as well as brand-specific observations.

In [336…

```
#import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from matplotlib.ticker import ScalarFormatter
```

Cases

In [4]:

```
cases = pd.read_csv("/Users/arielooms/Documents/College/Fall 24/CMSE 201/Data/us_counties_covid19_daily.csv
cases
```

Out[4]:

	date	county	state	fips	cases	deaths
0	2020-01-21	Snohomish	Washington	53061.0	1	0.0
1	2020-01-22	Snohomish	Washington	53061.0	1	0.0
2	2020-01-23	Snohomish	Washington	53061.0	1	0.0
3	2020-01-24	Cook	Illinois	17031.0	1	0.0
4	2020-01-24	Snohomish	Washington	53061.0	1	0.0
...	...	...	...	...	...	...
800432	2020-12-05	Sweetwater	Wyoming	56037.0	2098	10.0
800433	2020-12-05	Teton	Wyoming	56039.0	1739	2.0
800434	2020-12-05	Uinta	Wyoming	56041.0	1187	5.0
800435	2020-12-05	Washakie	Wyoming	56043.0	519	8.0
800436	2020-12-05	Weston	Wyoming	56045.0	419	2.0

800437 rows × 6 columns

Michigan

In [183…

```
#Michigan COVID data only
michigan_mask = cases["state"] == "Michigan"
michigan_cases = cases[michigan_mask]
michigan_cases
```

Out[183…

	date	county	state	fips	cases	deaths
1079	2020-03-10	Oakland	Michigan	26125.0	1	0.0
1080	2020-03-10	Wayne	Michigan	26163.0	1	0.0
1259	2020-03-11	Oakland	Michigan	26125.0	1	0.0
1260	2020-03-11	Wayne	Michigan	26163.0	1	0.0
1477	2020-03-12	Ingham	Michigan	26065.0	1	0.0
...	...	...	...	...	...	...
798511	2020-12-05	Unknown	Michigan	NaN	1751	10.0
798512	2020-12-05	Van Buren	Michigan	26159.0	3189	53.0
798513	2020-12-05	Washtenaw	Michigan	26161.0	11306	150.0
798514	2020-12-05	Wayne	Michigan	26163.0	70257	3313.0
798515	2020-12-05	Wexford	Michigan	26165.0	748	18.0

21186 rows × 6 columns

In [185…

```
#find the name of every MI county
michigan_cases["county"].unique()
```

Out[185…

```
array(['Oakland', 'Wayne', 'Ingham', 'Kent', 'Montcalm', 'St. Clair',
       'Washtenaw', 'Bay', 'Charlevoix', 'Leelanau', 'Macomb', 'Monroe',
       'Ottawa', 'Jackson', 'Otsego', 'Clinton', 'Eaton', 'Genesee',
       'Livingston', 'Midland', 'Allegan', 'Barry', 'Berrien', 'Calhoun',
       'Clare', 'Saginaw', 'Tuscola', 'Wexford', 'Emmet', 'Gladwin',
       'Grand Traverse', 'Roscommon', 'Chippewa', 'Kalamazoo', 'Muskegon',
       'Newaygo', 'Hillsdale', 'Isabella', 'Kalkaska', 'Lapeer',
       'Manistee', 'Unknown', 'Iosco', 'Lenawee', 'Marquette', 'Sanilac',
       'Van Buren', 'Gogebic', 'Ionia', 'Mecosta', 'Missaukee', 'Oceana',
       'Ogemaw', 'Shiawassee', 'Cass', 'Crawford', 'Gratiot', 'Huron',
       'Osceola', 'Cheboygan', 'Antrim', 'Delta', 'Houghton',
       'St. Joseph', 'Arenac', 'Branch', 'Oscoda', 'Mackinac',
       'Dickinson', 'Luce', 'Presque Isle', 'Mason', 'Schoolcraft',
       'Menominee', 'Alpena', 'Lake', 'Montmorency', 'Alcona', 'Baraga',
       'Benzie', 'Iron', 'Alger', 'Keweenaw', 'Ontonagon'], dtype=object)
```

In [187…

```python
#initiate county list, number of cases list, population list, and gdp list
mich_counties = ['Oakland', 'Wayne', 'Ingham', 'Kent', 'Montcalm', 'St. Clair',
       'Washtenaw', 'Bay', 'Charlevoix', 'Leelanau', 'Macomb', 'Monroe',
       'Ottawa', 'Jackson', 'Otsego', 'Clinton', 'Eaton', 'Genesee',
       'Livingston', 'Midland', 'Allegan', 'Barry', 'Berrien', 'Calhoun',
       'Clare', 'Saginaw', 'Tuscola', 'Wexford', 'Emmet', 'Gladwin',
       'Grand Traverse', 'Roscommon', 'Chippewa', 'Kalamazoo', 'Muskegon',
       'Newaygo', 'Hillsdale', 'Isabella', 'Kalkaska', 'Lapeer',
       'Manistee', 'Iosco', 'Lenawee', 'Marquette', 'Sanilac',
       'Van Buren', 'Gogebic', 'Ionia', 'Mecosta', 'Missaukee', 'Oceana',
       'Ogemaw', 'Shiawassee', 'Cass', 'Crawford', 'Gratiot', 'Huron',
       'Osceola', 'Cheboygan', 'Antrim', 'Delta', 'Houghton',
       'St. Joseph', 'Arenac', 'Branch', 'Oscoda', 'Mackinac',
       'Dickinson', 'Luce', 'Presque Isle', 'Mason', 'Schoolcraft',
       'Menominee', 'Alpena', 'Lake', 'Montmorency', 'Alcona', 'Baraga',
       'Benzie', 'Iron', 'Alger', 'Keweenaw', 'Ontonagon']

county_cases = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
county_pops = [1269827, 1743396, 285797, 663788, 69099, 160080, 364568, 102362, 26145, 23155, 875967,154855
               305826, 159090, 25933, 79788, 108786, 401121, 197380, 84316, 122588, 64214, 151687,
               133308, 31325, 187191, 52744, 34226, 34031, 26085, 96544, 24044, 36321, 263538,
               177589, 51488, 45439, 63870, 18736, 89322, 25759, 25273, 96760, 67450, 40254,
               75927, 14071, 65688, 41416, 15390, 27042, 21061, 68083, 51807, 13642, 41632,
               30709, 23381, 26132, 24553, 36799, 38038, 60978, 15167, 45928, 8699, 10740,
               26021, 6596, 13227, 29000, 8124, 22665, 28975, 12817, 9799, 10581, 8337, 18555,
               11775, 8741, 2189, 5891]
```

```
mich_gdp = [104254094, 92979659, 17186665, 42625043, 1554515, 5363330, 24986674, 3545561, 1108451,
            728367, 37893248, 5588368, 14426246, 6374681, 1064333, 2170942, 5200987,15263502,
            6970420, 5689329, 4645409, 1482523, 7491792, 6350135, 1115008, 8243904, 1264535, 1214002,
            1667798, 480574, 4968074, 596684, 1064522, 15017678, 5905749, 1346613, 1520652, 2350168,
            532280, 2227861, 821217, 869368, 3019702, 2610540, 1158417, 2929873, 451217, 1456182,
            1192740, 409810, 616045, 539235, 1610377, 1065278, 406589, 1425014, 1435791, 733271,
            582077, 567251, 1186556, 1057922, 2008481, 419860, 1356763, 189010, 403745, 1280039,
            144842, 334480, 987662, 266970, 652337, 1043571, 234910, 249723, 217818, 267441,
            472157, 308720, 235085, 48015, 123746]
```

In [189…

```
#loops through entire Michigan dataframe
#finds the correct county for the row being iterated through
#adds the number of cases in that row to the county's total
for index, row in michigan_cases.iterrows():
    #print(f"Index: {index}")
    #print(f"Date: {row['date']}, Cases: {row['cases']}, State: {row["state"]}")
    for name in mich_counties:
        if row["county"] == name:
            index = mich_counties.index(name)
            county_cases[index] += row["cases"]
```

United States Cases

In [221…

```
# initiate dictionary containing state, abbrieviation, region, and all months case totals

cases_dict = {'Washington': {"Abbr": "WA", "Region": "West", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May"
               'Illinois': {"Abbr": "IL","Region": "Midwest","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May"
               'California': {"Abbr": "CA","Region": "West","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May":
               'Arizona': {"Abbr": "AZ","Region": "West","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0,
               'Massachusetts': {"Abbr": "MA","Region": "Northeast", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": (
               'Wisconsin': {"Abbr": "WI","Region": "Midwest", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "Ma
               'Texas': {"Abbr": "TX","Region": "South","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, '
               'Nebraska': {"Abbr": "NE","Region": "Midwest","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May"
               'Utah': {"Abbr": "UT","Region": "West","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "J
               'Oregon': {"Abbr": "OR","Region": "West", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0,
               'Florida': {"Abbr": "FL","Region": "South","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0
```

```
        'New York': {"Abbr": "NY","Region": "Northeast","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "Ma
        'Rhode Island': {"Abbr": "RI","Region": "Northeast", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0
        'Georgia': {"Abbr": "GA","Region": "South","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0
        'New Hampshire': {"Abbr": "NH","Region": "Northeast","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0
        'North Carolina': {"Abbr": "NC","Region": "South","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "M
        'New Jersey': {"Abbr": "NJ","Region": "Northeast","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "M
        'Colorado': {"Abbr": "CO","Region": "West","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0
        'Maryland': {"Abbr": "MD","Region": "Northeast", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "Ma
        'Nevada': {"Abbr": "NV","Region": "West", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0,
        'Tennessee': {"Abbr": "TN","Region": "West","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0
        'Hawaii': {"Abbr": "HI","Region": "West","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "
        'Indiana': {"Abbr": "IN","Region": "Midwest","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May":
        'Kentucky': {"Abbr": "KY","Region": "South","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0
        'Minnesota': {"Abbr": "MN","Region": "Midwest","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May"
        'Oklahoma': {"Abbr": "OK","Region": "South", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May":
        'Pennsylvania': {"Abbr": "PA","Region": "Northeast","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0,
        'South Carolina': {"Abbr": "SC","Region": "South","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "M
        'District of Columbia': {"Abbr": "DC","Region": "Other","Jan": 0, "Feb": 0, "Mar": 0, "Apr"
        'Kansas': {"Abbr": "KS","Region": "Midwest","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0
        'Missouri': {"Abbr": "MO","Region": "Midwest","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May"
        'Vermont': {"Abbr": "VT","Region": "Northeast","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May"
        'Virginia': {"Abbr": "VA","Region": "South","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0
        'Connecticut': {"Abbr": "CT","Region": "Northeast","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "
        'Iowa': {"Abbr": "IA","Region": "Midwest","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0,
        'Louisiana': {"Abbr": "LA","Region": "South","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May":
        'Ohio': {"Abbr": "OH","Region": "Midwest","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0,
        'Michigan': {"Abbr": "MI","Region": "Midwest","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May"
        'South Dakota': {"Abbr": "SD","Region": "Midwest","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "M
        'Arkansas': {"Abbr": "AR","Region": "South","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0
        'Delaware': {"Abbr": "DE","Region": "Northeast","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "Ma
        'Mississippi': {"Abbr": "MS","Region": "South", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "Ma
        'New Mexico': {"Abbr": "NM","Region": "West","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May":
        'North Dakota': {"Abbr": "ND","Region": "Midwest","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "M
        'Wyoming': {"Abbr": "WY","Region": "West", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0
        'Alaska': {"Abbr": "AK","Region": "West", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0,
        'Maine': {"Abbr": "ME","Region": "Northeast", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May"
        'Alabama': {"Abbr": "AL","Region": "South","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0
        'Idaho': {"Abbr": "ID","Region": "West","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "J
        'Montana': {"Abbr": "MT","Region": "West", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0
        'Puerto Rico': {"Abbr": "PR","Region": "Other", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "Ma
        'Virgin Islands': {"Abbr": "VI","Region": "Other", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "
        'Guam': {"Abbr": "GU","Region": "Other", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "
```

```
               'West Virginia': {"Abbr": "WV","Region": "South", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "M
               'Northern Mariana Islands': {"Abbr": "MP","Region": "Other", "Jan": 0, "Feb": 0, "Mar": 0, '
              }

#initiates a list of all the US states and territories for easier plotting
states_list = ['Washington', 'Illinois', 'California', 'Arizona', 'Massachusetts', 'Wisconsin', 'Texas', 'N
```

In [558…

```
for index, row in cases.iterrows():  #loops through each index in each row so I can access any row and any
    #print(f"Index: {index}")
    #print(f"Date: {row['date']}, Cases: {row['cases']}, State: {row["state"]}")
    for state in cases_dict:
        if state == row["state"]:    #checks which state currently being iterated through
            if "-01-" in row['date']:    #checks which month the case was in
                cases_dict[state]["Jan"] += row["cases"]   #adds the number of cases in that row to the to
            if "-02-" in row['date']:
                cases_dict[state]["Feb"] += row["cases"]
            if "-03-" in row['date']:
                cases_dict[state]["Mar"] += row["cases"]
            if "-04-" in row['date']:
                cases_dict[state]["Apr"] += row["cases"]
            if "-05-" in row['date']:
                cases_dict[state]["May"] += row["cases"]
            if "-06-" in row['date']:
                cases_dict[state]["Jun"] += row["cases"]
            if "-07-" in row['date']:
                cases_dict[state]["Jul"] += row["cases"]
            if "-08-" in row['date']:
                cases_dict[state]["Aug"] += row["cases"]
            if "-09-" in row['date']:
                cases_dict[state]["Sep"] += row["cases"]
            if "-10-" in row['date']:
                cases_dict[state]["Oct"] += row["cases"]
            if "-11-" in row['date']:
                cases_dict[state]["Nov"] += row["cases"]
            if "-12-" in row['date']:
                cases_dict[state]["Dec"] += row["cases"]


#print(cases_dict["New York"]["Oct"])
```

In [560…

```
#creates lists for each month, each index is a states number of cases in that month
state_names = cases_dict.keys()
state_abbr = []
jan_cases = []
feb_cases = []
mar_cases = []
apr_cases = []
may_cases = []
jun_cases = []
jul_cases = []
aug_cases = []
sep_cases = []
oct_cases = []
nov_cases =[]
dec_cases = []
all_months_cases = [jan_cases, feb_cases, mar_cases, apr_cases, may_cases, jun_cases, jul_cases, aug_cases

for state in state_names:
    state_abbr.append(cases_dict[state]["Abbr"])
    jan_cases.append(cases_dict[state]["Jan"])
    feb_cases.append(cases_dict[state]["Feb"])
    mar_cases.append(cases_dict[state]["Mar"])
    apr_cases.append(cases_dict[state]["Apr"])
    may_cases.append(cases_dict[state]["May"])
    jun_cases.append(cases_dict[state]["Jun"])
    jul_cases.append(cases_dict[state]["Jul"])
    aug_cases.append(cases_dict[state]["Aug"])
    sep_cases.append(cases_dict[state]["Sep"])
    oct_cases.append(cases_dict[state]["Oct"])
    nov_cases.append(cases_dict[state]["Nov"])
    dec_cases.append(cases_dict[state]["Dec"])
```

In [562…

```
#creates lists for each region in the US and each index represents cases in a given month
midwest_cases = [0,0,0,0,0,0,0,0,0,0,0,0]
west_cases = [0,0,0,0,0,0,0,0,0,0,0,0]
northeast_cases = [0,0,0,0,0,0,0,0,0,0,0,0]
south_cases = [0,0,0,0,0,0,0,0,0,0,0,0]
other_cases = [0,0,0,0,0,0,0,0,0,0,0,0]
for state in cases_dict:
    if cases_dict[state]["Region"] == "Midwest":   #checks which region a state is in based on dictionary
        midwest_cases[0] += cases_dict[state]["Jan"]  #adds total region's cases in a month
        midwest_cases[1] += cases_dict[state]["Feb"]
        midwest_cases[2] += cases_dict[state]["Mar"]
        midwest_cases[3] += cases_dict[state]["Apr"]
        midwest_cases[4] += cases_dict[state]["May"]
        midwest_cases[5] += cases_dict[state]["Jun"]
        midwest_cases[6] += cases_dict[state]["Jul"]
        midwest_cases[7] += cases_dict[state]["Aug"]
        midwest_cases[8] += cases_dict[state]["Sep"]
        midwest_cases[9] += cases_dict[state]["Oct"]
        midwest_cases[10] += cases_dict[state]["Nov"]
        midwest_cases[11] += cases_dict[state]["Dec"]
    elif cases_dict[state]["Region"] == "West":
        west_cases[0] += cases_dict[state]["Jan"]
        west_cases[1] += cases_dict[state]["Feb"]
        west_cases[2] += cases_dict[state]["Mar"]
        west_cases[3] += cases_dict[state]["Apr"]
        west_cases[4] += cases_dict[state]["May"]
        west_cases[5] += cases_dict[state]["Jun"]
        west_cases[6] += cases_dict[state]["Jul"]
        west_cases[7] += cases_dict[state]["Aug"]
        west_cases[8] += cases_dict[state]["Sep"]
        west_cases[9] += cases_dict[state]["Oct"]
        west_cases[10] += cases_dict[state]["Nov"]
        west_cases[11] += cases_dict[state]["Dec"]
    elif cases_dict[state]["Region"] == "Northeast":
        northeast_cases[0] += cases_dict[state]["Jan"]
        northeast_cases[1] += cases_dict[state]["Feb"]
        northeast_cases[2] += cases_dict[state]["Mar"]
        northeast_cases[3] += cases_dict[state]["Apr"]
        northeast_cases[4] += cases_dict[state]["May"]
        northeast_cases[5] += cases_dict[state]["Jun"]
        northeast_cases[6] += cases_dict[state]["Jul"]
        northeast_cases[7] += cases_dict[state]["Aug"]
        northeast_cases[8] += cases_dict[state]["Sep"]
        northeast_cases[9] += cases_dict[state]["Oct"]
        northeast_cases[10] += cases_dict[state]["Nov"]
        northeast_cases[11] += cases_dict[state]["Dec"]
    elif cases_dict[state]["Region"] == "South":
        south_cases[0] += cases_dict[state]["Jan"]
        south_cases[1] += cases_dict[state]["Feb"]
        south_cases[2] += cases_dict[state]["Mar"]
        south_cases[3] += cases_dict[state]["Apr"]
        south_cases[4] += cases_dict[state]["May"]
        south_cases[5] += cases_dict[state]["Jun"]
        south_cases[6] += cases_dict[state]["Jul"]
        south_cases[7] += cases_dict[state]["Aug"]
        south_cases[8] += cases_dict[state]["Sep"]
        south_cases[9] += cases_dict[state]["Oct"]
        south_cases[10] += cases_dict[state]["Nov"]
        south_cases[11] += cases_dict[state]["Dec"]
    else:
        other_cases[0] += cases_dict[state]["Jan"]
        other_cases[1] += cases_dict[state]["Feb"]
        other_cases[2] += cases_dict[state]["Mar"]
        other_cases[3] += cases_dict[state]["Apr"]
        other_cases[4] += cases_dict[state]["May"]
        other_cases[5] += cases_dict[state]["Jun"]
        other_cases[6] += cases_dict[state]["Jul"]
        other_cases[7] += cases_dict[state]["Aug"]
        other_cases[8] += cases_dict[state]["Sep"]
        other_cases[9] += cases_dict[state]["Oct"]
        other_cases[10] += cases_dict[state]["Nov"]
        other_cases[11] += cases_dict[state]["Dec"]
```

Highest and Lowest States by Quarterly Case Numbers 2020

In [565…

```
#top 5 highest and lowest states case numbers

import heapq #finds the biggest and smallest n items in a list
def top_5(month):
    #finds the top 5 values in a given month
    #finds the index in the month list
    #finds the state at that index in the states list
    top_5_vals = heapq.nlargest(5, month)
    top_states = []
    top_state_vals = []
    for val in top_5_vals:
        top_state_vals.append(val)
        index = month.index(val)
        state = states_list[index]
        state_abbr = cases_dict[state]["Abbr"]
        top_states.append(state_abbr)
    return top_state_vals, top_states
```

```
def low_5(month):
    #finds the lowest 5 values in a given month
    #finds the index in the month list
    #finds the state at that index in the states list
    low_5_vals = heapq.nsmallest(5, month)
    low_states = []
    low_state_vals = []
    for val in low_5_vals:
        low_state_vals.append(val)
        index = month.index(val)
        state = states_list[index]
        state_abbr = cases_dict[state]["Abbr"]
        low_states.append(state_abbr)
    return low_state_vals, low_states


#uses functions to find the highest and lowest case numbers and their corresponding states
jan_top_vals, jan_top_states = top_5(jan_cases)
jan_low_vals, jan_low_states = low_5(jan_cases)
mar_top_vals, mar_top_states = top_5(mar_cases)
mar_low_vals, mar_low_states = low_5(mar_cases)
jul_top_vals, jul_top_states = top_5(jul_cases)
jul_low_vals, jul_low_states = low_5(jul_cases)
dec_top_vals, dec_top_states = top_5(dec_cases)
dec_low_vals, dec_low_states = low_5(dec_cases)
```

Vaccines

In [859…

```
vaccines = pd.read_csv("/Users/arielooms/Documents/College/Fall 24/CMSE 201/Data/vaccines.csv")
vaccines
#https://www.kaggle.com/datasets/volkanastasia/covid-19-vaccinations-in-the-us
```

Out [859…

	Date	MMWR_week	Location	Distributed	Distributed_Janssen	Distributed_Moderna	Distributed_Pfizer	D
0	05/10/2023	19	NE	5481710	152400	1647380	2905630	
1	05/10/2023	19	LA	10282120	330500	3807980	5164550	
2	05/10/2023	19	GA	28727475	869100	9763000	14773655	
3	05/10/2023	19	WY	1281755	49300	490040	585605	
4	05/10/2023	19	CO	17769135	501900	5402640	9029715	
...	...	...	...	...	...	...	...	
38483	12/13/2020	51	AS	3900	0	0	0	
38484	12/13/2020	51	VI	975	0	0	0	
38485	12/13/2020	51	MP	4875	0	0	0	
38486	12/13/2020	51	US	13650	0	0	0	
38487	12/13/2020	51	GU	3900	0	0	0	

38488 rows × 109 columns

I am focusing primarily on the brands and total amount of vaccines distributed, so while there is much more that can be done with this dataset, I can reduce it to only the columns I will be directly using.

In [862…

```
#get rid of unnecessary cols
vaccines.drop(columns=['Dist_Per_100K',
 'Distributed_Per_100k_5Plus',
 'Distributed_Per_100k_12Plus',
 'Distributed_Per_100k_18Plus',
 'Distributed_Per_100k_65Plus',
 'Administered',
 'Administered_5Plus',
 'Administered_12Plus',
 'Administered_18Plus',
 'Administered_65Plus',
 'Administered_Janssen',
 'Administered_Moderna',
 'Administered_Pfizer',
 'Administered_Novavax',
 'Administered_Unk_Manuf',
 'Admin_Per_100K',
 'Admin_Per_100k_5Plus',
 'Admin_Per_100k_12Plus',
 'Admin_Per_100k_18Plus',
 'Admin_Per_100k_65Plus',
 'Recip_Administered',
 'Administered_Dose1_Recip',
 'Administered_Dose1_Pop_Pct',
 'Administered_Dose1_Recip_5Plus',
 'Administered_Dose1_Recip_5PlusPop_Pct',
 'Administered_Dose1_Recip_12Plus',
 'Administered_Dose1_Recip_12PlusPop_Pct',
 'Administered_Dose1_Recip_18Plus',
 'Administered_Dose1_Recip_18PlusPop_Pct',
 'Administered_Dose1_Recip_65Plus',
 'Administered_Dose1_Recip_65PlusPop_Pct',
 'Series_Complete_Yes',
 'Series_Complete_Pop_Pct',
 'Series_Complete_5Plus',
 'Series_Complete_5PlusPop_Pct',
 'Series_Complete_12Plus',
 'Series_Complete_12PlusPop_Pct',
 'Series_Complete_18Plus',
 'Series_Complete_18PlusPop_Pct',
 'Series_Complete_65Plus',
 'Series_Complete_65PlusPop_Pct',
 'Series_Complete_Janssen',
 'Series_Complete_Moderna',
 'Series_Complete_Pfizer',
 'Series_Complete_Novavax',
 'Series_Complete_Unk_Manuf',
 'Series_Complete_Janssen_5Plus',
 'Series_Complete_Moderna_5Plus',
 'Series_Complete_Pfizer_5Plus',
 'Series_Complete_Unk_Manuf_5Plus',
 'Series_Complete_Janssen_12Plus',
 'Series_Complete_Moderna_12Plus',
 'Series_Complete_Pfizer_12Plus',
 'Series_Complete_Unk_Manuf_12Plus',
 'Series_Complete_Janssen_18Plus',
'Series_Complete_Moderna_18Plus',
'Series_Complete_Pfizer_18Plus',
'Series_Complete_Unk_Manuf_18Plus',
'Series_Complete_Janssen_65Plus',
'Series_Complete_Moderna_65Plus',
'Series_Complete_Pfizer_65Plus',
'Series_Complete_Unk_Manuf_65Plus',
'Additional_Doses',
'Additional_Doses_Vax_Pct',
'Additional_Doses_5Plus',
'Additional_Doses_5Plus_Vax_Pct',
'Additional_Doses_12Plus',
'Additional_Doses_12Plus_Vax_Pct',
'Additional_Doses_18Plus',
'Additional_Doses_18Plus_Vax_Pct',
'Additional_Doses_50Plus',
'Additional_Doses_50Plus_Vax_Pct',
'Additional_Doses_65Plus',
'Additional_Doses_65Plus_Vax_Pct',
'Additional_Doses_Moderna',
'Additional_Doses_Pfizer',
'Additional_Doses_Janssen',
'Additional_Doses_Unk_Manuf',
'Second_Booster',
'Second_Booster_50Plus',
'Second_Booster_50Plus_Vax_Pct',
'Second_Booster_65Plus',
'Second_Booster_65Plus_Vax_Pct',
'Second_Booster_Janssen',
'Second_Booster_Moderna',
'Second_Booster_Pfizer',
'Second_Booster_Unk_Manuf',
'Administered_Bivalent',
'Admin_Bivalent_PFR',
'Admin_Bivalent_MOD',
'Dist_Bivalent_PFR',
'Dist_Bivalent_MOD',
'Bivalent_Booster_5Plus',
'Bivalent_Booster_5Plus_Pop_Pct',
'Bivalent_Booster_12Plus',
'Bivalent_Booster_12Plus_Pop_Pct',
'Bivalent_Booster_18Plus',
'Bivalent_Booster_18Plus_Pop_Pct',
'Bivalent_Booster_65Plus',
'Bivalent_Booster_65Plus_Pop_Pct'])
```

Out[862...

	Date	MMWR_week	Location	Distributed	Distributed_Janssen	Distributed_Moderna	Distributed_Pfizer	D
0	05/10/2023	19	NE	5481710	152400	1647380	2905630	
1	05/10/2023	19	LA	10282120	330500	3807980	5164550	
2	05/10/2023	19	GA	28727475	869100	9763000	14773655	
3	05/10/2023	19	WY	1281755	49300	490040	585605	
4	05/10/2023	19	CO	17769135	501900	5402640	9029715	
...	...	...	...	...	...	...	...	
38483	12/13/2020	51	AS	3900	0	0	0	
38484	12/13/2020	51	VI	975	0	0	0	
38485	12/13/2020	51	MP	4875	0	0	0	
38486	12/13/2020	51	US	13650	0	0	0	
38487	12/13/2020	51	GU	3900	0	0	0	

38488 rows × 9 columns

In [864...

```
#for each date in the vaccines datasets, finds just the year for easier analysis later on
years_list = []
for date in vaccines["Date"]:
    only2020 = date.split('/')[2] #at each '/', the string is split into indices and I want the second inde
    years_list.append(only2020)

vaccines["Year"] = years_list  #creates a new column containing just the years
vaccines
```

```
Out[864…
              Date  MMWR_week  Location  Distributed  Distributed_Janssen  Distributed_Moderna  Distributed_Pfizer  D
    0   05/10/2023         19        NE      5481710               152400              1647380             2905630
    1   05/10/2023         19        LA     10282120               330500              3807980             5164550
    2   05/10/2023         19        GA     28727475               869100              9763000            14773655
    3   05/10/2023         19        WY      1281755                49300               490040              585605
    4   05/10/2023         19        CO     17769135               501900              5402640             9029715
  ...          ...        ...       ...          ...                  ...                  ...                 ...
38483   12/13/2020         51        AS         3900                    0                    0                   0
38484   12/13/2020         51        VI          975                    0                    0                   0
38485   12/13/2020         51        MP         4875                    0                    0                   0
38486   12/13/2020         51        US        13650                    0                    0                   0
38487   12/13/2020         51        GU         3900                    0                    0                   0

38488 rows × 110 columns
```

```
In [866…
#repeats the split above but for months
months_only = []
for date in vaccines["Date"]:
    month_only = date.split('/')[0]
    months_only.append(month_only)

vaccines["Month"] = months_only
vaccines
```

Out [866…

	Date	MMWR_week	Location	Distributed	Distributed_Janssen	Distributed_Moderna	Distributed_Pfizer	D
0	05/10/2023	19	NE	5481710	152400	1647380	2905630	
1	05/10/2023	19	LA	10282120	330500	3807980	5164550	
2	05/10/2023	19	GA	28727475	869100	9763000	14773655	
3	05/10/2023	19	WY	1281755	49300	490040	585605	
4	05/10/2023	19	CO	17769135	501900	5402640	9029715	
...	...	...	...	...	...	...	...	
38483	12/13/2020	51	AS	3900	0	0	0	
38484	12/13/2020	51	VI	975	0	0	0	
38485	12/13/2020	51	MP	4875	0	0	0	
38486	12/13/2020	51	US	13650	0	0	0	
38487	12/13/2020	51	GU	3900	0	0	0	

38488 rows × 111 columns

In [868…

```
#finds all vaccine data for 2020 to be compared to my cases data
only2020_mask = vaccines["Year"] == "2020"
vaccines2020 = vaccines[only2020_mask]
vaccines2020
```

Out[868…

	Date	MMWR_week	Location	Distributed	Distributed_Janssen	Distributed_Moderna	Distributed_Pfizer	D
37350	12/31/2020	53	LTC	2180950	0	0	0	
37351	12/31/2020	53	BP2	7600	0	0	0	
37352	12/31/2020	53	MT	46150	0	0	0	
37353	12/31/2020	53	RI	58750	0	0	0	
37354	12/31/2020	53	KY	174750	0	0	0	
...	...	...	...	...	...	...	...	
38483	12/13/2020	51	AS	3900	0	0	0	
38484	12/13/2020	51	VI	975	0	0	0	
38485	12/13/2020	51	MP	4875	0	0	0	
38486	12/13/2020	51	US	13650	0	0	0	
38487	12/13/2020	51	GU	3900	0	0	0	

1138 rows × 111 columns

In [870…

```
#initiates a dictionary containing state, abbrieviation, and all the month totals for vaccines dsitributed

vaccines_dict = {'Washington': {"Abbr": "WA", "Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, '
               'Illinois': {"Abbr": "IL","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul"
               'California': {"Abbr": "CA","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Ju
               'Arizona': {"Abbr": "AZ","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":
               'Massachusetts': {"Abbr": "MA","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, '
               'Wisconsin': {"Abbr": "WI","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul"
               'Texas': {"Abbr": "TX","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":0,
               'Nebraska': {"Abbr": "NE","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul"
               'Utah': {"Abbr": "UT","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":0, '
               'Oregon': {"Abbr": "OR","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":0
               'Florida': {"Abbr": "FL","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":
               'New York': {"Abbr": "NY","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul"
               'Rhode Island': {"Abbr": "RI","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "
               'Georgia': {"Abbr": "GA","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":
               'New Hampshire': {"Abbr": "NH","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, '
               'North Carolina': {"Abbr": "NC","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0,
               'New Jersey': {"Abbr": "NJ","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Ju
               'Colorado': {"Abbr": "CO","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul"
               'Maryland': {"Abbr": "MD","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul"
               'Nevada': {"Abbr": "NV","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":0
               'Tennessee': {"Abbr": "TN","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul
               'Hawaii': {"Abbr": "HI","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":0
               'Indiana': {"Abbr": "IN","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":
               'Kentucky': {"Abbr": "KY","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul"
               'Minnesota': {"Abbr": "MN","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul
               'Oklahoma': {"Abbr": "OK","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul"
               'Pennsylvania': {"Abbr": "PA","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "
               'South Carolina': {"Abbr": "SC","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0,
               'District of Columbia': {"Abbr": "DC","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Ju
               'Kansas': {"Abbr": "KS","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":0
               'Missouri': {"Abbr": "MO","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul"
               'Vermont': {"Abbr": "VT","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":
               'Virginia': {"Abbr": "VA","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul"
               'Connecticut': {"Abbr": "CT","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Ju
               'Iowa': {"Abbr": "IA","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":0,
               'Louisiana': {"Abbr": "LA","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul
               'Ohio': {"Abbr": "OH","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":0,
               'Michigan': {"Abbr": "MI","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul"
               'South Dakota': {"Abbr": "SD","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "
               'Arkansas': {"Abbr": "AR","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul"
               'Delaware': {"Abbr": "DE","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul"
               'Mississippi': {"Abbr": "MS","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Ju
               'New Mexico': {"Abbr": "NM","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Ju
               'North Dakota': {"Abbr": "ND","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "
               'Wyoming': {"Abbr": "WY","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":
               'Alaska': {"Abbr": "AK","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":0
               'Maine': {"Abbr": "ME","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":0,
               'Alabama': {"Abbr": "AL","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":
               'Idaho': {"Abbr": "ID","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":0,
               'Montana': {"Abbr": "MT","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":
               'Puerto Rico': {"Abbr": "PR","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Ju
               'Virgin Islands': {"Abbr": "VI","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0,
               'Guam': {"Abbr": "GU","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0, "Jul":0,
               'West Virginia': {"Abbr": "WV","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0, "Jun": 0,
               'Northern Mariana Islands': {"Abbr": "MP","Jan": 0, "Feb": 0, "Mar": 0, "Apr": 0, "May": 0,
              }
```

In [872…

```
#iterate through each row in vaccine data in 2020
#checks which state the row is referencing
#adds the number of vaccines to the month december - only month in 2020 with vaccines being distributed
for index, row in vaccines2020.iterrows():
    #print(f"Index: {index}")
    #print(f"Date: {row['date']}, Cases: {row['cases']}, State: {row["state"]}")
    for state in vaccines_dict:
        if vaccines_dict[state]["Abbr"] == row["Location"]:
            if row['Month'] == "12":
                vaccines_dict[state]["Dec"] += row["Distributed"]

dec20_vacs = []
for state in vaccines_dict:
    dec20_vacs.append(vaccines_dict[state]["Dec"])
```

In [874…

```
#initiates total cases for each month
#iniates a list for all months to be put together to chart
all_jan_cases = 0
all_feb_cases = 0
all_mar_cases = 0
all_apr_cases = 0
all_may_cases = 0
all_jun_cases = 0
all_jul_cases = 0
all_aug_cases = 0
all_sep_cases = 0
all_oct_cases = 0
all_nov_cases = 0
all_dec_cases = 0

for state in cases_dict:
    all_jan_cases += (cases_dict[state]["Jan"])
    all_feb_cases += (cases_dict[state]["Feb"])
    all_mar_cases += (cases_dict[state]["Mar"])
    all_apr_cases += (cases_dict[state]["Apr"])
    all_may_cases += (cases_dict[state]["May"])
    all_jun_cases += (cases_dict[state]["Jun"])
    all_jul_cases += (cases_dict[state]["Jul"])
    all_aug_cases += (cases_dict[state]["Aug"])
    all_sep_cases += (cases_dict[state]["Sep"])
    all_oct_cases += (cases_dict[state]["Oct"])
    all_nov_cases += (cases_dict[state]["Nov"])
    all_dec_cases += (cases_dict[state]["Dec"])

all_us_cases_months = [all_jan_cases, all_feb_cases, all_mar_cases, all_apr_cases, all_may_cases, all_jun_
```

In [876…

```
#initiates total vacs for each month
#iniates a list for all months to be put together to chart

all_jan_vacs = 0
all_feb_vacs = 0
all_mar_vacs = 0
all_apr_vacs = 0
all_may_vacs = 0
all_jun_vacs = 0
all_jul_vacs = 0
all_aug_vacs = 0
all_sep_vacs = 0
all_oct_vacs = 0
all_nov_vacs = 0
all_dec_vacs = 0

for state in cases_dict:
    all_jan_vacs += (vaccines_dict[state]["Jan"])
    all_feb_vacs += (vaccines_dict[state]["Feb"])
    all_mar_vacs += (vaccines_dict[state]["Mar"])
    all_apr_vacs += (vaccines_dict[state]["Apr"])
    all_may_vacs += (vaccines_dict[state]["May"])
    all_jun_vacs += (vaccines_dict[state]["Jun"])
    all_jul_vacs += (vaccines_dict[state]["Jul"])
    all_aug_vacs += (vaccines_dict[state]["Aug"])
    all_sep_vacs += (vaccines_dict[state]["Sep"])
    all_oct_vacs += (vaccines_dict[state]["Oct"])
    all_nov_vacs += (vaccines_dict[state]["Nov"])
    all_dec_vacs += (vaccines_dict[state]["Dec"])

all_us_vacs_months = [all_jan_vacs, all_feb_vacs, all_mar_vacs, all_apr_vacs, all_may_vacs, all_jun_vacs,
```

In [878…

```
#finds the total distributed from 2021-2023 of each vaccine brand
janssen = vaccines["Distributed_Janssen"].sum()
moderna = vaccines["Distributed_Moderna"].sum()
pfizer = vaccines["Distributed_Pfizer"].sum()
```

```
novavax = vaccines["Distributed_Novavax"].sum()
other = vaccines["Distributed_Unk_Manuf"].sum()
amounts = [janssen, moderna, pfizer, novavax, other]
brand = ["Janssen", "Moderna", "Pfizer", "Novavax", "Other"]
pfizer, novavax, other
```

Out[878…

```
(334689112325, 83524200.0, 80128700)
```

In [880…

```
#gets just the year 2021 to look at vaccine distribution
only21_mask = vaccines["Year"] == "2021"
only21 = vaccines[only21_mask]
only21
```

Out[880…

	Date	MMWR_week	Location	Distributed	Distributed_Janssen	Distributed_Moderna	Distributed_Pfizer	Di
13696	12/31/2021	52	KS	5195855	249000	1947420	2999435	
13697	12/31/2021	52	AL	8056140	383300	3253600	4419240	
13698	12/31/2021	52	NJ	18675625	898800	6510320	11266505	
13699	12/31/2021	52	GA	18321585	792200	6853800	10675585	
13700	12/31/2021	52	PR	6437230	211200	2275660	3950370	
...	...	...	...	...	...	...	...	
37345	01/01/2021	53	DD2	230525	0	0	0	
37346	01/01/2021	53	AK	54975	0	0	0	
37347	01/01/2021	53	NV	115275	0	0	0	
37348	01/01/2021	53	MA	339350	0	0	0	
37349	01/01/2021	53	VI	4925	0	0	0	

23654 rows × 111 columns

In [882…

```
#finds the number of each vaccine brand distributed in March, June, September, and December
mar_only_mask = only21["Month"] == "03"
mar_only21 = only21[mar_only_mask]
moderna_mar = mar_only21["Distributed_Moderna"].sum()
pfizer_mar = mar_only21["Distributed_Pfizer"].sum()
```

```
janssen_mar = mar_only21["Distributed_Janssen"].sum()
other_mar = mar_only21["Distributed_Unk_Manuf"].sum()
```

In [884…

```
jun_only_mask = only21["Month"] == "06"
jun_only21 = only21[jun_only_mask]
moderna_jun = jun_only21["Distributed_Moderna"].sum()
pfizer_jun = jun_only21["Distributed_Pfizer"].sum()
janssen_jun = jun_only21["Distributed_Janssen"].sum()
other_jun = jun_only21["Distributed_Unk_Manuf"].sum()
```

In [886…

```
sep_only_mask = only21["Month"] == "09"
sep_only21 = only21[sep_only_mask]
moderna_sep = sep_only21["Distributed_Moderna"].sum()
pfizer_sep = sep_only21["Distributed_Pfizer"].sum()
janssen_sep = sep_only21["Distributed_Janssen"].sum()
other_sep = sep_only21["Distributed_Unk_Manuf"].sum()
```

In [888…

```
dec_only_mask = only21["Month"] == "12"
dec_only21 = only21[dec_only_mask]
moderna_dec = dec_only21["Distributed_Moderna"].sum()
pfizer_dec = dec_only21["Distributed_Pfizer"].sum()
janssen_dec = dec_only21["Distributed_Janssen"].sum()
other_dec = dec_only21["Distributed_Unk_Manuf"].sum()
```

In [890…

```
#makes lists of each vaccine brand - each index is one of the four months observed - easier for plotting
moderna_vacs = [moderna_mar, moderna_jun, moderna_sep, moderna_dec]
pfizer_vacs = [pfizer_mar, pfizer_jun, pfizer_sep, pfizer_dec]
janssen_vacs = [janssen_mar, janssen_jun, janssen_sep, janssen_dec]
other_vacs = [other_mar, other_jun, other_sep, other_dec]
month_nums = ["Mar", "Jun", "Sep", "Dec"]  #creates an easy way to label the x-axis for each month observe
```

In [892…

```
#initiates a dictionary to track the amount of vaccine brands distributed in each state
vacs_brands_dict = {'Washington': {"Abbr": "WA", "Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0
               'Illinois': {"Abbr": "IL","Region": "Midwest", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
               'California': {"Abbr": "CA","Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
               'Arizona': {"Abbr": "AZ","Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Massachusetts': {"Abbr": "MA","Region": "Northeast", "Moderna": 0, "Pfizer": 0, "Janssen"
                'Wisconsin': {"Abbr": "WI","Region": "Midwest", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Texas': {"Abbr": "TX","Region": "South", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Nebraska': {"Abbr": "NE","Region": "Midwest", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Utah': {"Abbr": "UT","Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Oregon': {"Abbr": "OR","Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Florida': {"Abbr": "FL","Region": "South", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'New York': {"Abbr": "NY","Region": "Northeast", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Rhode Island': {"Abbr": "RI","Region": "Northeast", "Moderna": 0, "Pfizer": 0, "Janssen":
                'Georgia': {"Abbr": "GA","Region": "South", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'New Hampshire': {"Abbr": "NH","Region": "Northeast", "Moderna": 0, "Pfizer": 0, "Janssen"
                'North Carolina': {"Abbr": "NC","Region": "South", "Moderna": 0, "Pfizer": 0, "Janssen": 0
                'New Jersey': {"Abbr": "NJ","Region": "Northeast", "Moderna": 0, "Pfizer": 0, "Janssen": 0
                'Colorado': {"Abbr": "CO","Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Maryland': {"Abbr": "MD","Region": "Northeast", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Nevada': {"Abbr": "NV","Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Tennessee': {"Abbr": "TN","Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Hawaii': {"Abbr": "HI","Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Indiana': {"Abbr": "IN","Region": "Midwest", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Kentucky': {"Abbr": "KY","Region": "South", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Minnesota': {"Abbr": "MN","Region": "Midwest", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Oklahoma': {"Abbr": "OK","Region": "South", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Pennsylvania': {"Abbr": "PA","Region": "Northeast", "Moderna": 0, "Pfizer": 0, "Janssen":
                'South Carolina': {"Abbr": "SC","Region": "South", "Moderna": 0, "Pfizer": 0, "Janssen": 0
                'District of Columbia': {"Abbr": "DC","Region": "Other", "Moderna": 0, "Pfizer": 0, "Janss
                'Kansas': {"Abbr": "KS","Region": "Midwest", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Missouri': {"Abbr": "MO","Region": "Midwest", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Vermont': {"Abbr": "VT","Region": "Northeast", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Virginia': {"Abbr": "VA","Region": "South", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Connecticut': {"Abbr": "CT","Region": "Northeast", "Moderna": 0, "Pfizer": 0, "Janssen":
                'Iowa': {"Abbr": "IA","Region": "Midwest", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Louisiana': {"Abbr": "LA","Region": "South", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Ohio': {"Abbr": "OH","Region": "Midwest", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Michigan': {"Abbr": "MI","Region": "Midwest", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'South Dakota': {"Abbr": "SD","Region": "Midwest", "Moderna": 0, "Pfizer": 0, "Janssen": 0
                'Arkansas': {"Abbr": "AR","Region": "South", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Delaware': {"Abbr": "DE","Region": "Northeast", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Mississippi': {"Abbr": "MS","Region": "South", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'New Mexico': {"Abbr": "NM","Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'North Dakota': {"Abbr": "ND","Region": "Midwest", "Moderna": 0, "Pfizer": 0, "Janssen": 0
                'Wyoming': {"Abbr": "WY","Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Alaska': {"Abbr": "AK","Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Maine': {"Abbr": "ME","Region": "Northeast", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Alabama': {"Abbr": "AL","Region": "South", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Idaho': {"Abbr": "ID","Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Montana': {"Abbr": "MT","Region": "West", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Puerto Rico': {"Abbr": "PR","Region": "Other", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'Virgin Islands': {"Abbr": "VI","Region": "Other", "Moderna": 0, "Pfizer": 0, "Janssen": 0
                'Guam': {"Abbr": "GU","Region": "Other", "Moderna": 0, "Pfizer": 0, "Janssen": 0},
                'West Virginia': {"Abbr": "WV","Region": "South", "Moderna": 0, "Pfizer": 0, "Janssen": 0}
                'Northern Mariana Islands': {"Abbr": "MP","Region": "Other", "Moderna": 0, "Pfizer": 0, "Ja
                   }
states_list = ['Washington', 'Illinois', 'California', 'Arizona', 'Massachusetts', 'Wisconsin', 'Texas', 'N
```

In [894…

```
#iterates through entire table representing just 2021
for index, row in only21.iterrows():
    #print(f"Index: {index}")
    #print(f"Date: {row['date']}, Cases: {row['cases']}, State: {row["state"]}")
    for state in vacs_brands_dict:
        if vacs_brands_dict[state]["Abbr"] == row["Location"]:  #finds the correct state
            vacs_brands_dict[state]["Moderna"] += row["Distributed_Moderna"]  #adds the number of each bra
            vacs_brands_dict[state]["Pfizer"] += row["Distributed_Pfizer"]
            vacs_brands_dict[state]["Janssen"] += row["Distributed_Janssen"]
```

In [896…

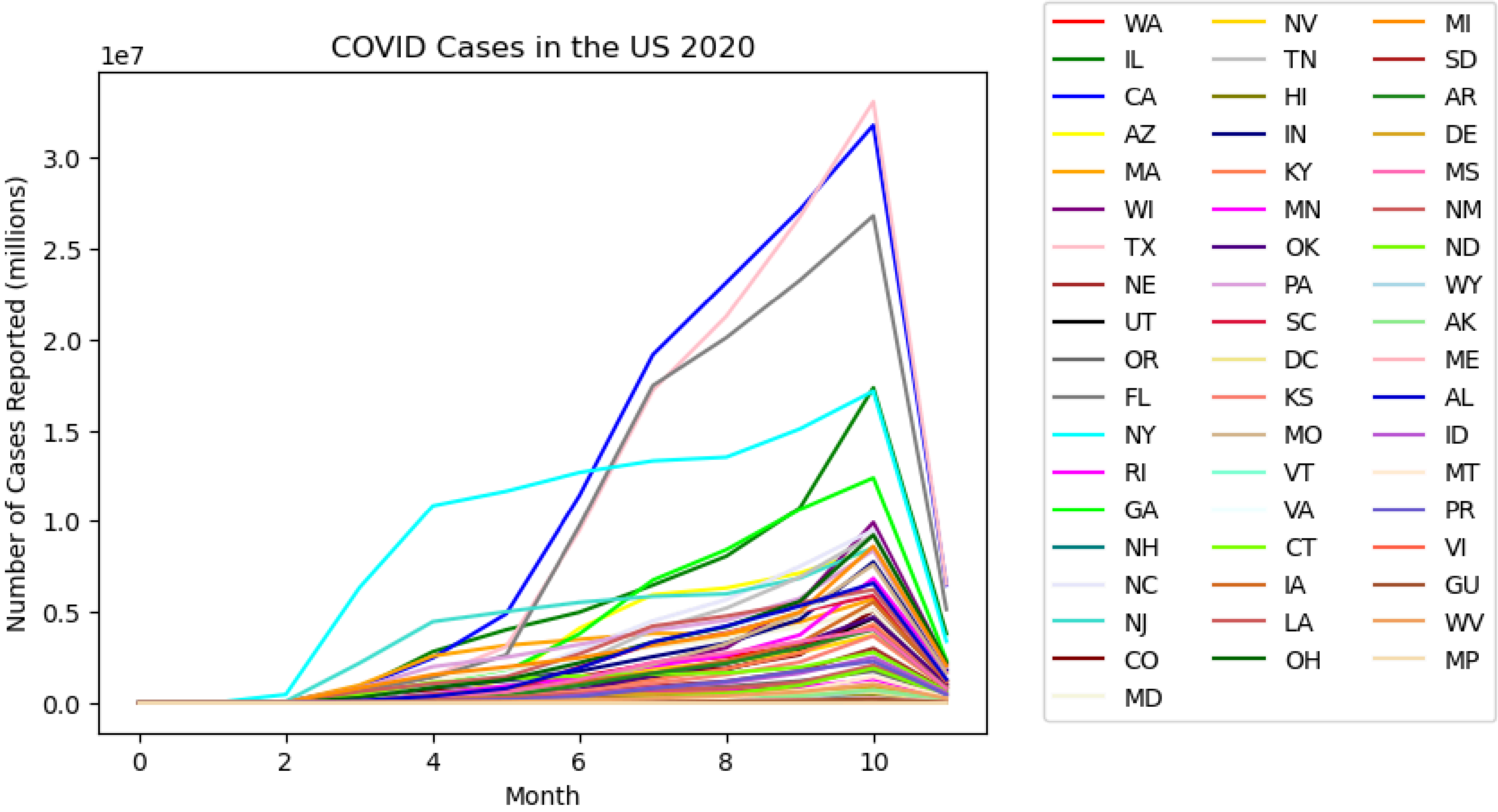
```
#makes new lists where each index is a state and how many of that brand were distributed in that state
moderna_all = []
pfizer_all = []
janssen_all = []
for state in vacs_brands_dict:
    moderna_all.append(vacs_brands_dict[state]["Moderna"])
    pfizer_all.append(vacs_brands_dict[state]["Pfizer"])
    janssen_all.append(vacs_brands_dict[state]["Janssen"])
```

Results

Cases by US State

In [794…

```
#all states cases over 2020
i = 0
color_list = [
    "red", "green", "blue", "yellow", "orange", "purple", "pink", "brown", "black", "dimgrey",
    "gray", "cyan", "magenta", "lime", "teal", "lavender", "turquoise", "maroon", "beige", "gold",
    "silver", "olive", "navy", "coral", "fuchsia", "indigo", "plum", "crimson", "khaki", "salmon",
    "tan", "aquamarine", "azure", "chartreuse", "chocolate", "indianred", "darkgreen", "darkorange",
    "firebrick", "forestgreen", "goldenrod", "hotpink", "indianred", "lawngreen", "lightblue",
    "lightgreen", "lightpink", "mediumblue", "mediumorchid", "papayawhip", "slateblue", "tomato", "sienna"
]

for state in state_names:
    state_cases = [month_cases[i] for month_cases in all_months_cases]  # Cases for one state across months
    plt.plot(state_cases, label=cases_dict[state]["Abbr"], color=color_list[i])  # Plot cases over months
    i += 1

plt.legend(bbox_to_anchor=(1.05, 0), loc="lower left", ncol=3)
plt.xlabel("Month")
plt.ylabel("Number of Cases Reported (millions)")
plt.title("COVID Cases in the US 2020")
plt.show()
```



It is a bit difficult to read this chart, but there are some clear outliers. Texas, California, New York, and Florida all have very high peaks in the month of October. Each state increases after March which makes sense as that is when the pandemic officially started in the United States. There is a plateau in most states from April to June, which could be correlated to the mask mandates and quarantines put into effect. There is also an obvious downturn in cases reported following October which is both expected, but surprising. It is expected because this is when the very first vaccines were beind administered which could account for lower numbers. However, it is also somewhat surprising as November and December are when many people have holiday get-togethers which could have possibly caused a spike in cases, but did not appear to do so.

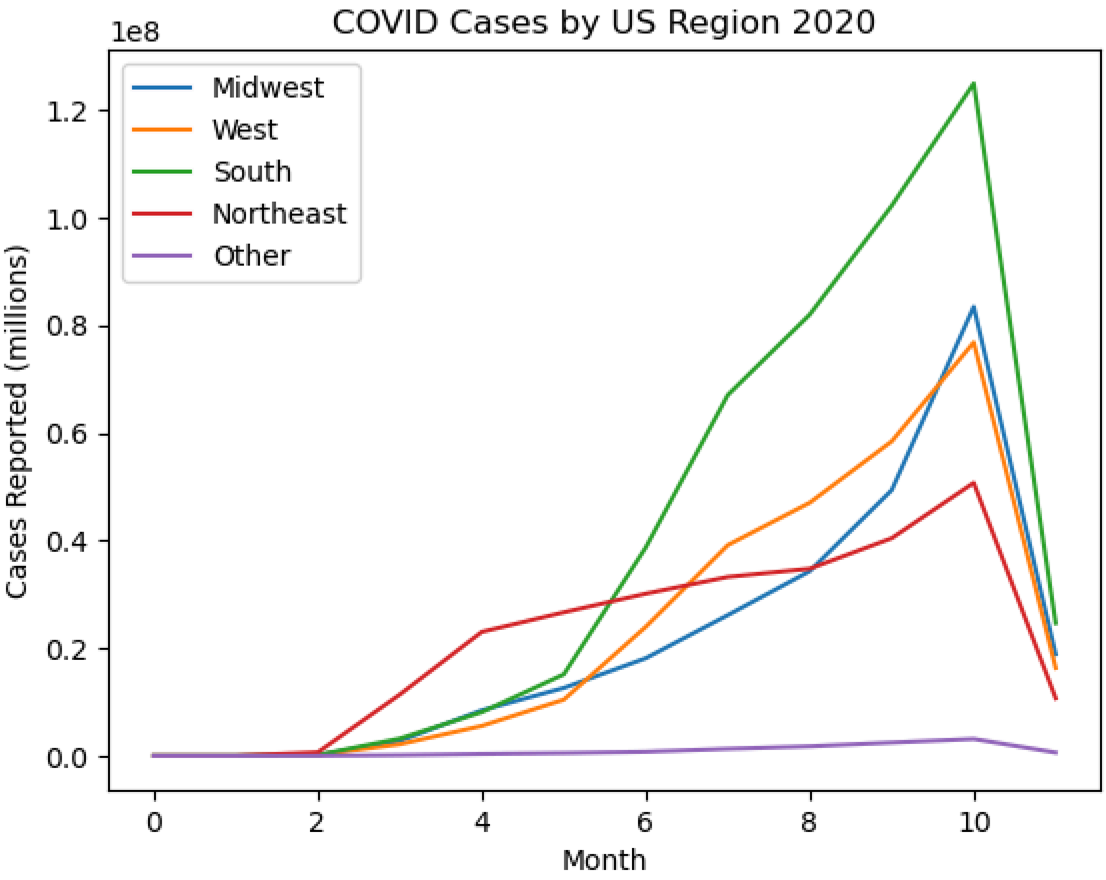
Due to the difficult nature of this graph in terms of readability, I decided to classify the states into five different regions to perhaps see the trends easier.

In [745…

```
plt.plot(midwest_cases, label="Midwest")
plt.plot(west_cases, label="West")
plt.plot(south_cases, label="South")
plt.plot(northeast_cases, label="Northeast")
plt.plot(other_cases, label="Other")
plt.title("COVID Cases by US Region 2020")
plt.xlabel("Month")
plt.ylabel("Cases Reported (millions)")
plt.legend(loc="upper left")
```

Out[745…

```
<matplotlib.legend.Legend at 0x3070d8620>
```



Very similar patterns as the graph above can be seen, but it can also be idenitifed that states in the South have the highest peak in October. Only two of the states identified earlier fall into region of the South, which could indicate that those states are high in case numbers due to population, but they are not representative of the rest of their region.

Quarterly Highest and Lowest States by Cases Reported

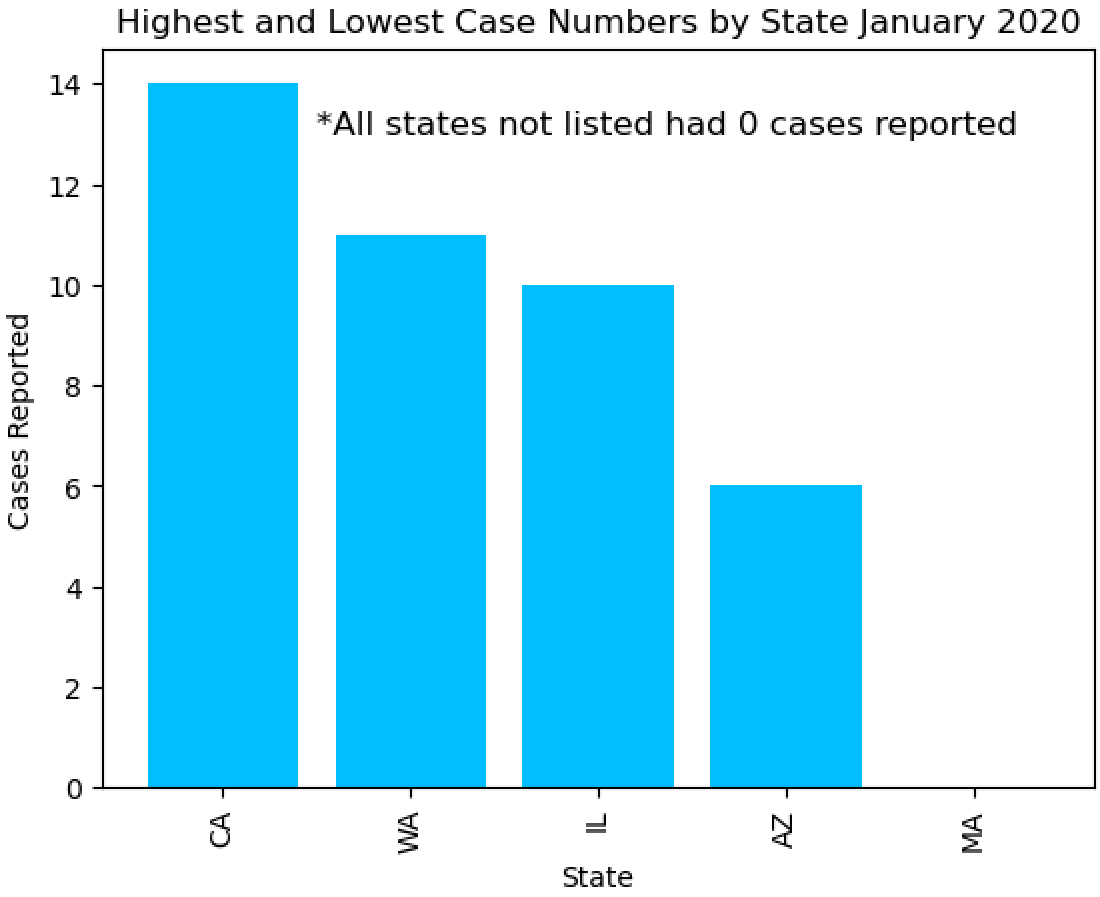
I wanted to identify the states each quarter in 2020 that had the highest number of cases reported and the lowest number reported.

In [798…

```
plt.title("Highest and Lowest Case Numbers by State January 2020")
plt.bar(jan_top_states, jan_top_vals, color="deepskyblue")
plt.xticks(rotation=90)
plt.xlabel("State")
plt.ylabel("Cases Reported")
plt.text(0.5, 13, '*All states not listed had 0 cases reported', fontsize=12)
'''plt.subplot(4,2,2)
plt.bar(jan_low_states, jan_low_vals)
plt.xticks(rotation=90)
plt.title("Highest and Lowest Case Numbers by State January 2020", loc="right")'''
```

Out[798…

```
'plt.subplot(4,2,2)\nplt.bar(jan_low_states, jan_low_vals)\nplt.xticks(rotation=90)\nplt.title("Highest an
d Lowest Case Numbers by State January 2020", loc="right")'
```

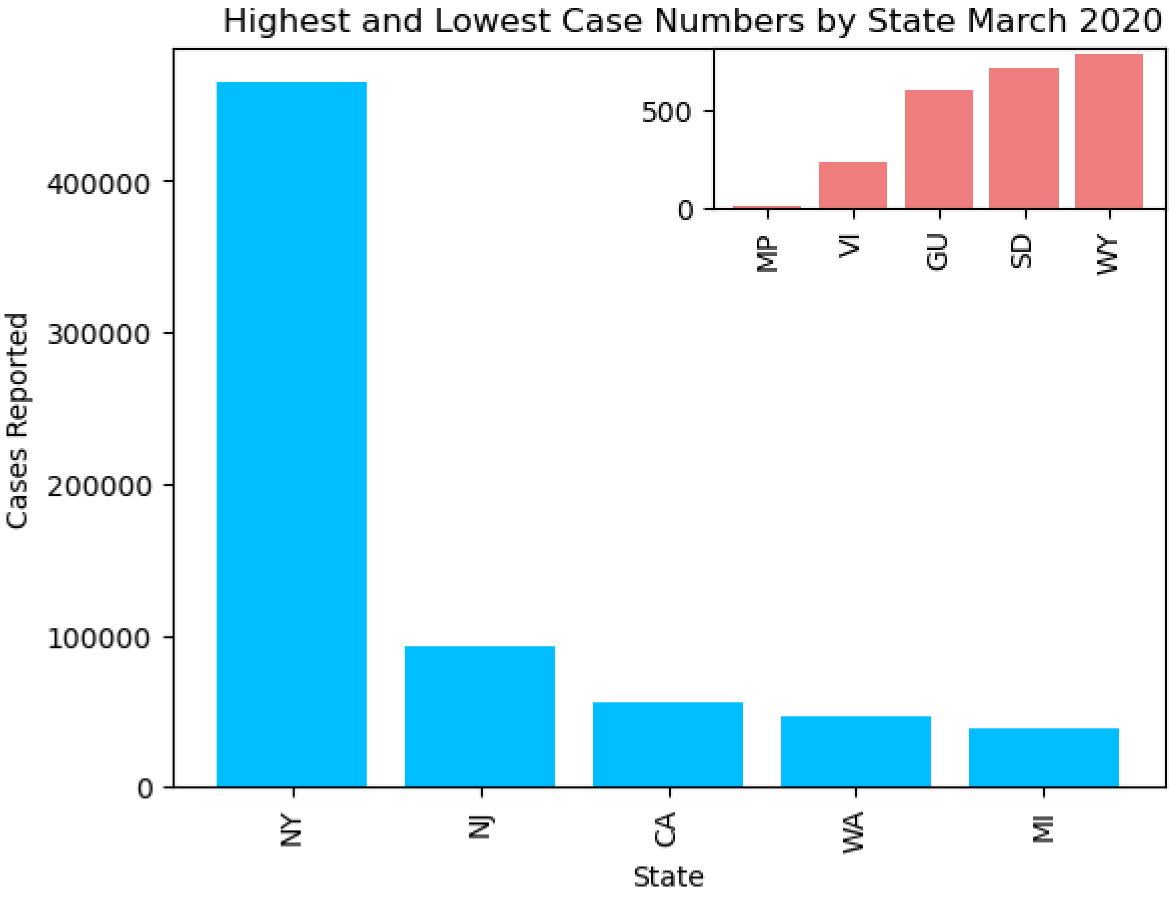


In January 2020, California, Washington, Illinois, and Arizona have the highest number of cases reported. The number is quite low, only reaching 14 cases in California, but these four states make sense as the top 4. California, Washington, and Illinois are very busy hubs for airway transportation and have hundreds of thousands of people coming into the airports everyday, so the disease could more easily travel overseas to these locations.

In [821…
```
plt.xlabel("State")
plt.ylabel("Cases Reported")
plt.bar(mar_top_states, mar_top_vals, color="deepskyblue")
plt.xticks(rotation=90)
plt.subplot(4,2,2)
plt.bar(mar_low_states, mar_low_vals, color="lightcoral")
plt.xticks(rotation=90)
plt.title("Highest and Lowest Case Numbers by State March 2020", loc="right")
```

Out[821…

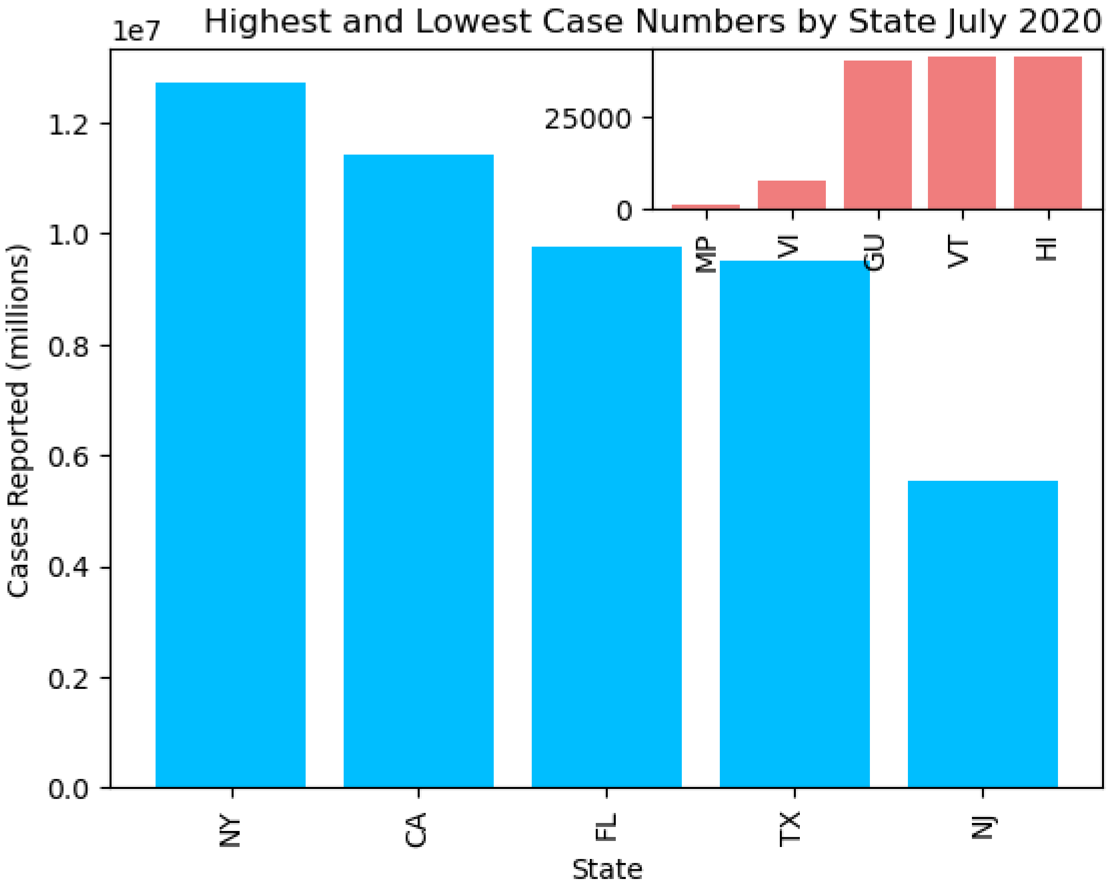
```
Text(1.0, 1.0, 'Highest and Lowest Case Numbers by State March 2020')
```



In March 2020, New York takes the lead in number of cases reported by over 30,000 cases. California and Washington still remain in the top 5, but New Jersey and Michigan also join. On the lowest end, South Dakota and Wyoming as well as some US territories take up the bottom 5. Again, this could be related to the populations of each state/territory as states with higher populations will in turn have more people in closer contact with each other.

In [817…

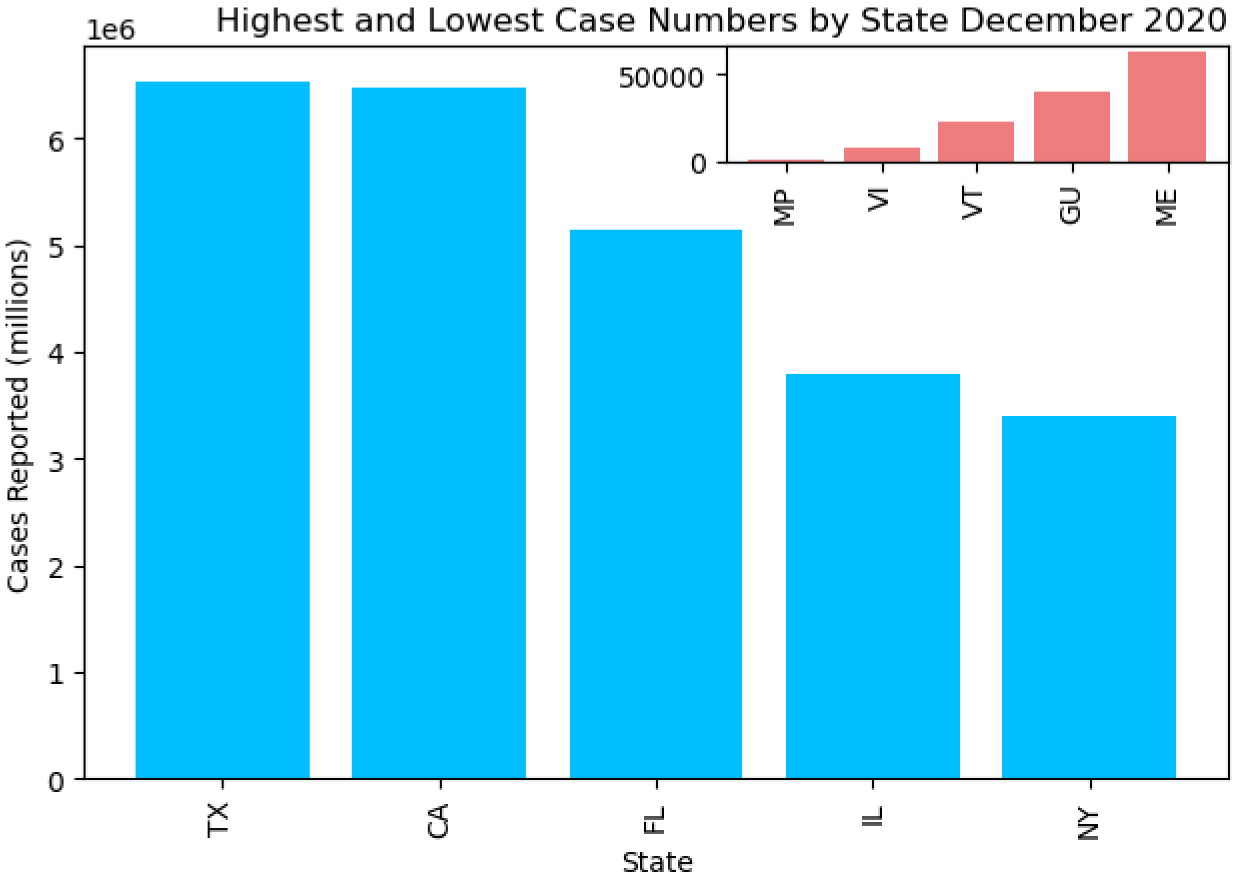
```
plt.xlabel("State")
plt.ylabel("Cases Reported (millions)")
plt.bar(jul_top_states, jul_top_vals, color="deepskyblue")
plt.xticks(rotation=90)
plt.subplot(4,2,2)

plt.bar(jul_low_states, jul_low_vals, color="lightcoral")
plt.xticks(rotation=90)
plt.title("Highest and Lowest Case Numbers by State July 2020", loc="right")
plt.ticklabel_format(style='plain', axis='y')
```



In July, four of the original top 5 from January return. Case numbers continue to rise even following the mask mandates and quarantine orders, which can be seen in the lowest five states which now also have quite a few cases reported. Hawaii and Vermont are now the two US states that fall into the lowest category.

In [824…

```
plt.xlabel("State")
plt.ylabel("Cases Reported (millions)")
plt.bar(dec_top_states, dec_top_vals, color="deepskyblue")
plt.xticks(rotation=90)
plt.subplot(4,2,2)
plt.bar(dec_low_states, dec_low_vals, color="lightcoral")
plt.xticks(rotation=90)
plt.title("Highest and Lowest Case Numbers by State December 2020", loc="right")
plt.tight_layout()
```

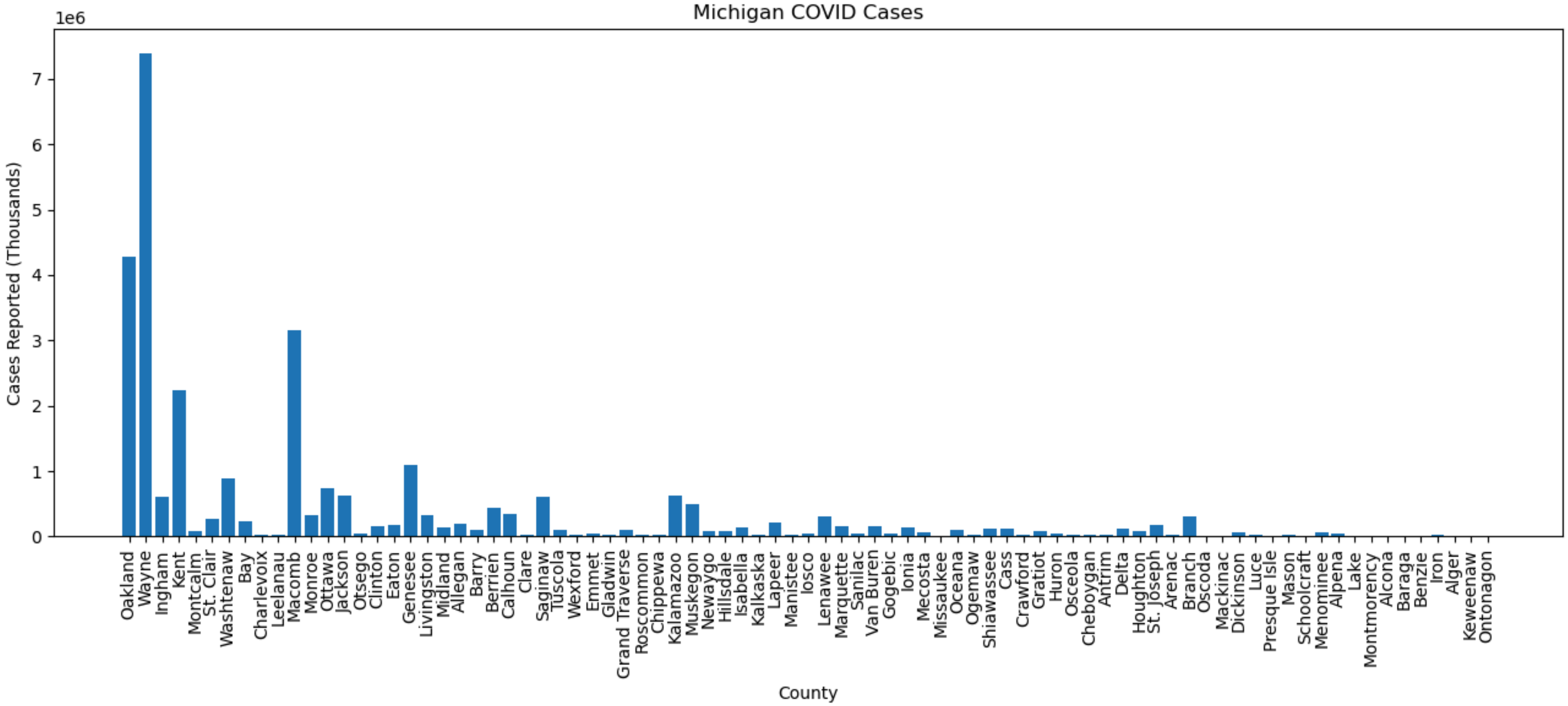


By the end of 2020, the original top 5 remain the same. Overall, throughout the year, there is not much variation in the top 5 and bottom 5 states by number of cases reported.

Cases in Michigan

In [758…

```
plt.figure(figsize=(13, 6))
plt.bar(mich_counties, county_cases)
plt.xticks(rotation=90)
plt.title("Michigan COVID Cases")
plt.xlabel("County")
plt.ylabel("Cases Reported (Thousands)")
plt.tight_layout()
```

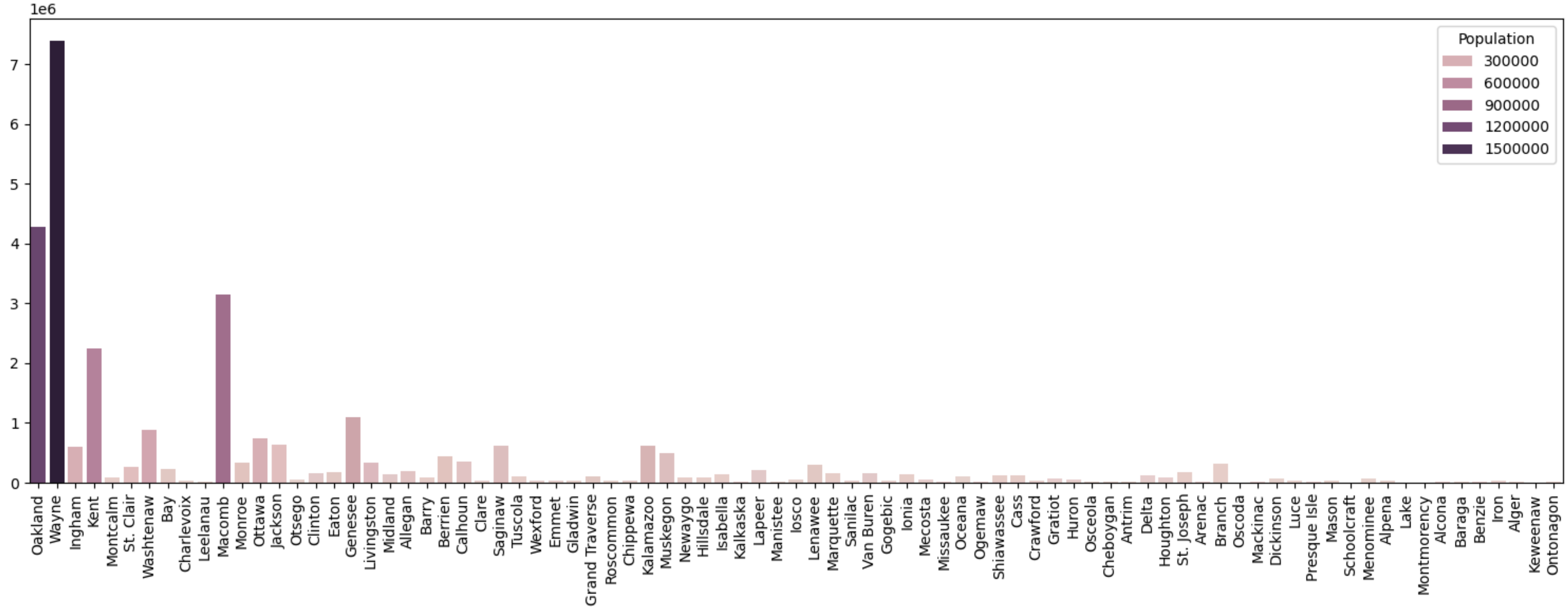


Wayne and Oakland counties have the highest number of cases reported throughout 2020. Macomb and Kent counties follow that, while most of the other counties fall into similar lower ranges.

In [760…

```
plt.figure(figsize=(15, 6))
sns.barplot(x=mich_counties, y=county_cases, hue=county_pops)
```

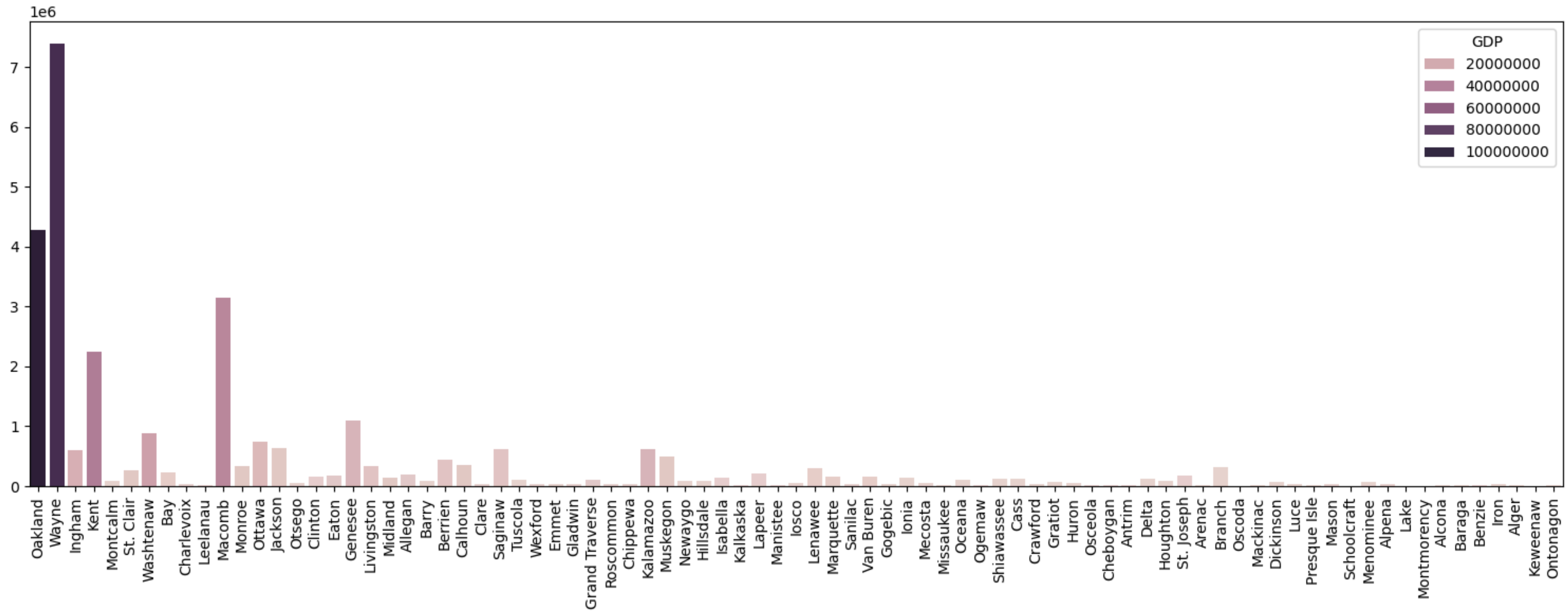
```
plt.xticks(rotation=90);
plt.legend(title='Population')
plt.tight_layout()
```



To see if population had an impact on which counties suffered from the most COVID-19 cases, I used additional data that reported the population for each county. When plotted together, there is an obvious connection between the number of cases and population. Wayne County has the highest population as well as the largest population. Oakland follows Wayne, then Macomb. Most of the counties in Michigan have relatively small populations that correspond with the low number of cases being reported.

In [762…

```
plt.figure(figsize=(15, 6))
sns.barplot(x=mich_counties, y=county_cases, hue=mich_gdp)
plt.xticks(rotation=90);
plt.legend(title='GDP')
plt.tight_layout()
```

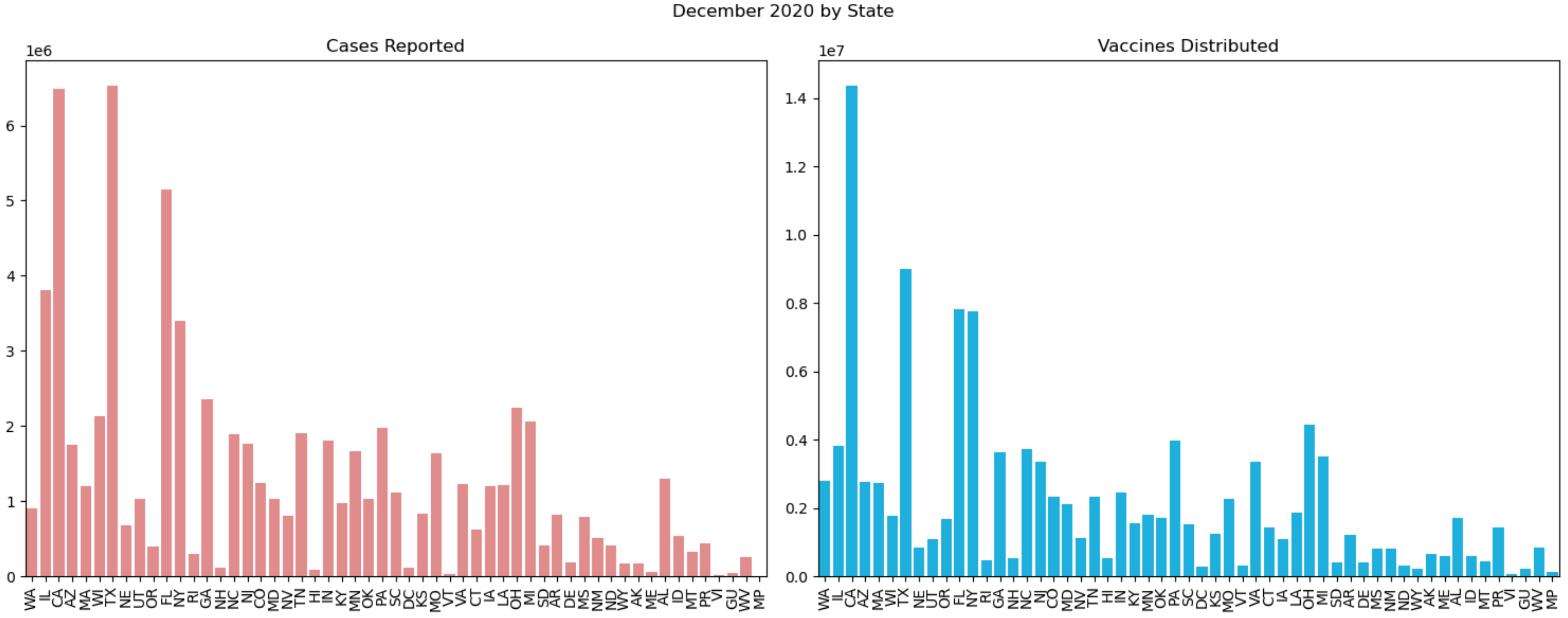


Looking into other variables that could play into the number of cases in different counties, I wanted to see if economic status of a county correlated to the number of cases experienced in that county. Again, Wayne and Oakland have the highest GDP as well as the highest number of cases. These results make sense to me, but are also somewhat surprising in another way. The higher the GDP, the bigger the workforce probably is, which ties back into a bigger population and closer contact between people and likely easier spread of disease including COVID-19. The surprising aspect of these results is that lower income communities most likely have a lower GDP. Lower income communities often suffer more from health problems due to lack of financial access to healthcare. That is not seen in the number of cases reported, but could possibly be seen later on in the pandemic when deaths or long-term health problems begin to be reported.

Cases Reported and Vaccines Distributed in December 2020

In [844…

```
plt.figure(figsize=(15, 6))
plt.suptitle("December 2020 by State")
plt.subplot(1,2,2)
sns.barplot(x=state_abbr, y=dec20_vacs, color="deepskyblue");
plt.xticks(rotation=90);
plt.title("Vaccines Distributed")
plt.subplot(1,2,1)
sns.barplot(x=state_abbr, y=dec_cases, color="lightcoral");
plt.xticks(rotation=90);
plt.title("Cases Reported")
plt.tight_layout()
```

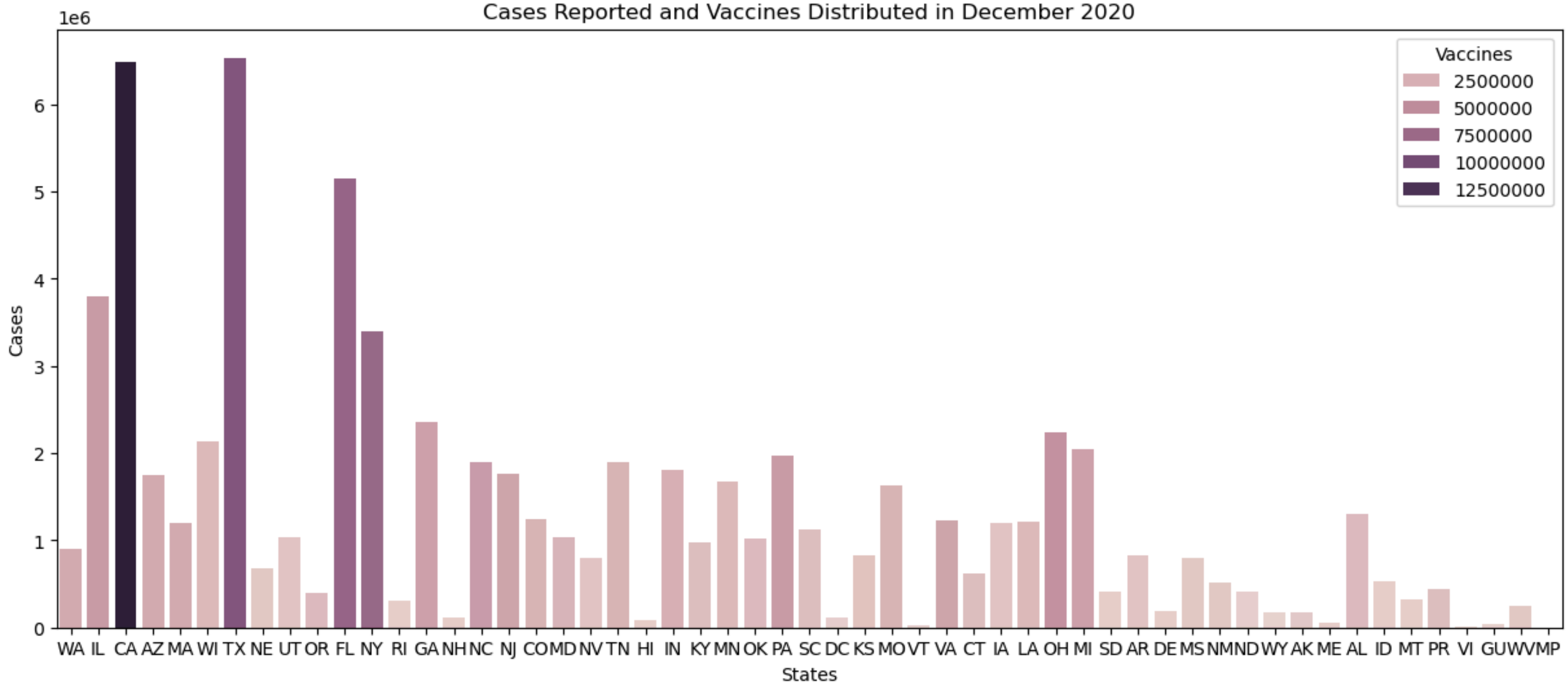


In [768…

```
plt.figure(figsize=(15, 6))
sns.barplot(data=dec_df, x="States", y="Cases", hue="Vaccines")
plt.title("Cases Reported and Vaccines Distributed in December 2020")
```

Out[768…

```
Text(0.5, 1.0, 'Cases Reported and Vaccines Distributed in December 2020')
```



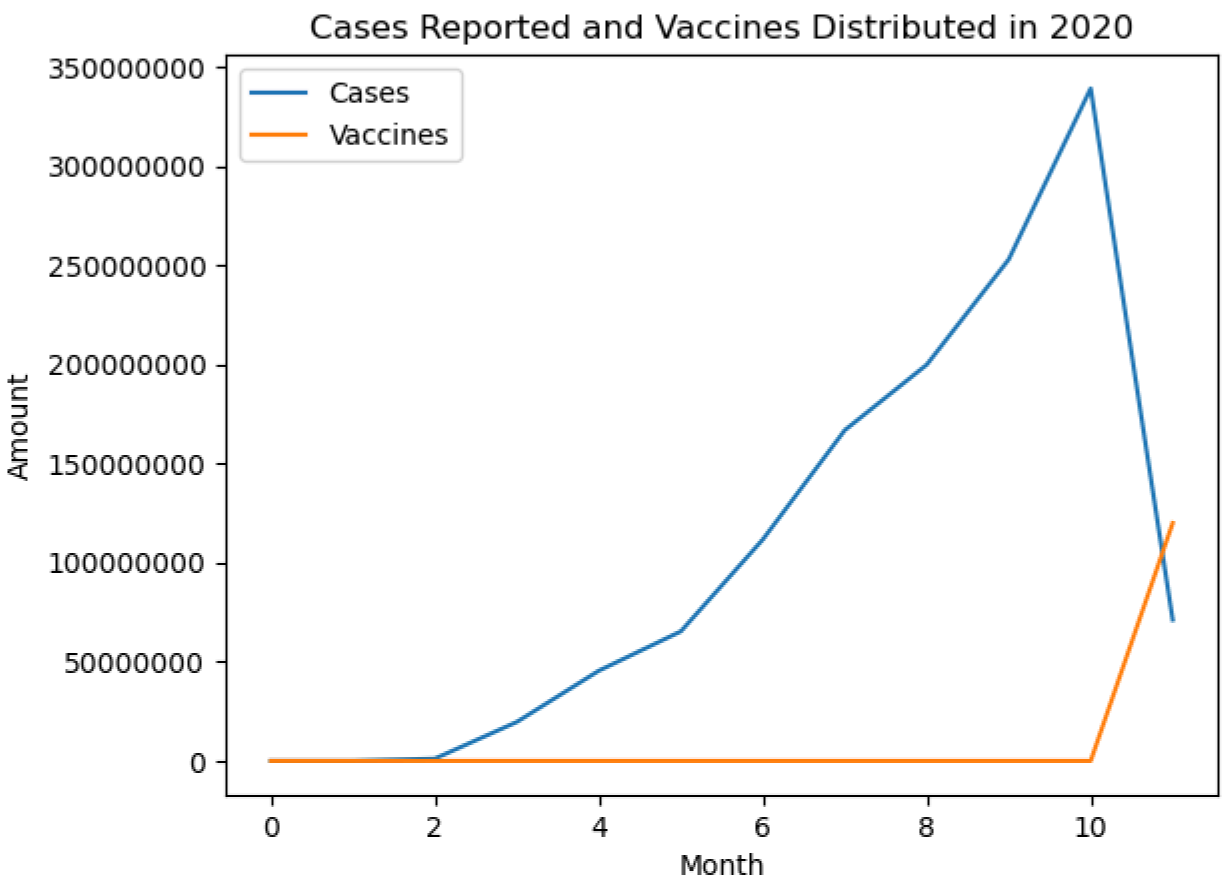
There is a clear correlation between the states that were facing the highest case numbers and the states who had the highest rollout of the vaccines as soon as it came out in December 2020. Shown in both graphs, California, Texas, and Florida faced the highest number of cases and also had the biggest rollout of the vaccines in December.

In [848…

```
plt.plot(all_us_cases_months, label="Cases")
plt.plot(all_us_vacs_months, label="Vaccines")
plt.title("Cases Reported and Vaccines Distributed in 2020")
plt.xlabel("Month")
plt.ylabel("Amount")
plt.ticklabel_format(style='plain', axis='y')
plt.legend()
```

Out[848…

```
<matplotlib.legend.Legend at 0x30b7297c0>
```



The sharp decline around the month of November in 2020 in case reports aligns very well with the public release of the vaccine, indicating that the distribution of the vaccine was reducing the number of COVID-19 cases.

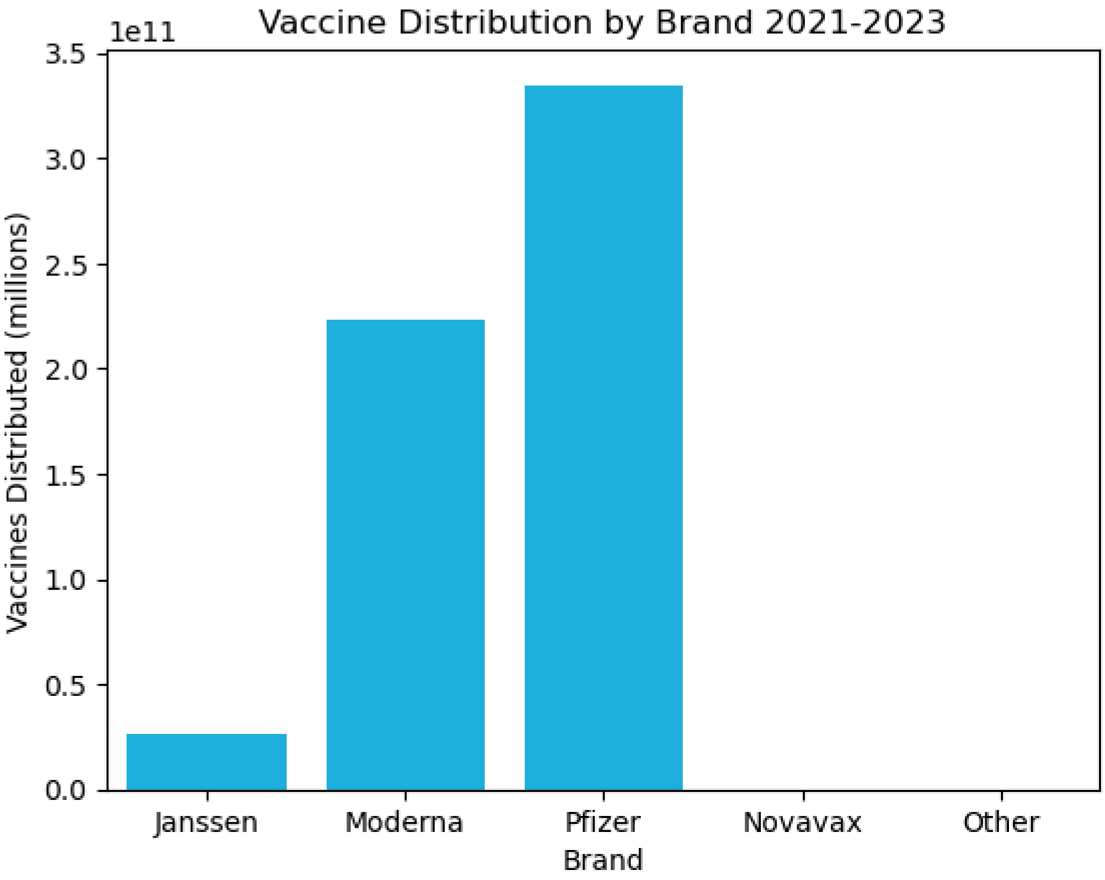
Vaccine Distribution by Brand

In [853…

```
sns.barplot(x=["Janssen", "Moderna", "Pfizer", "Novavax", "Other"], y=[janssen, moderna, pfizer, novavax, 
plt.title("Vaccine Distribution by Brand 2021–2023")
plt.xlabel("Brand")
plt.ylabel("Vaccines Distributed (millions)")
```

Out[853…

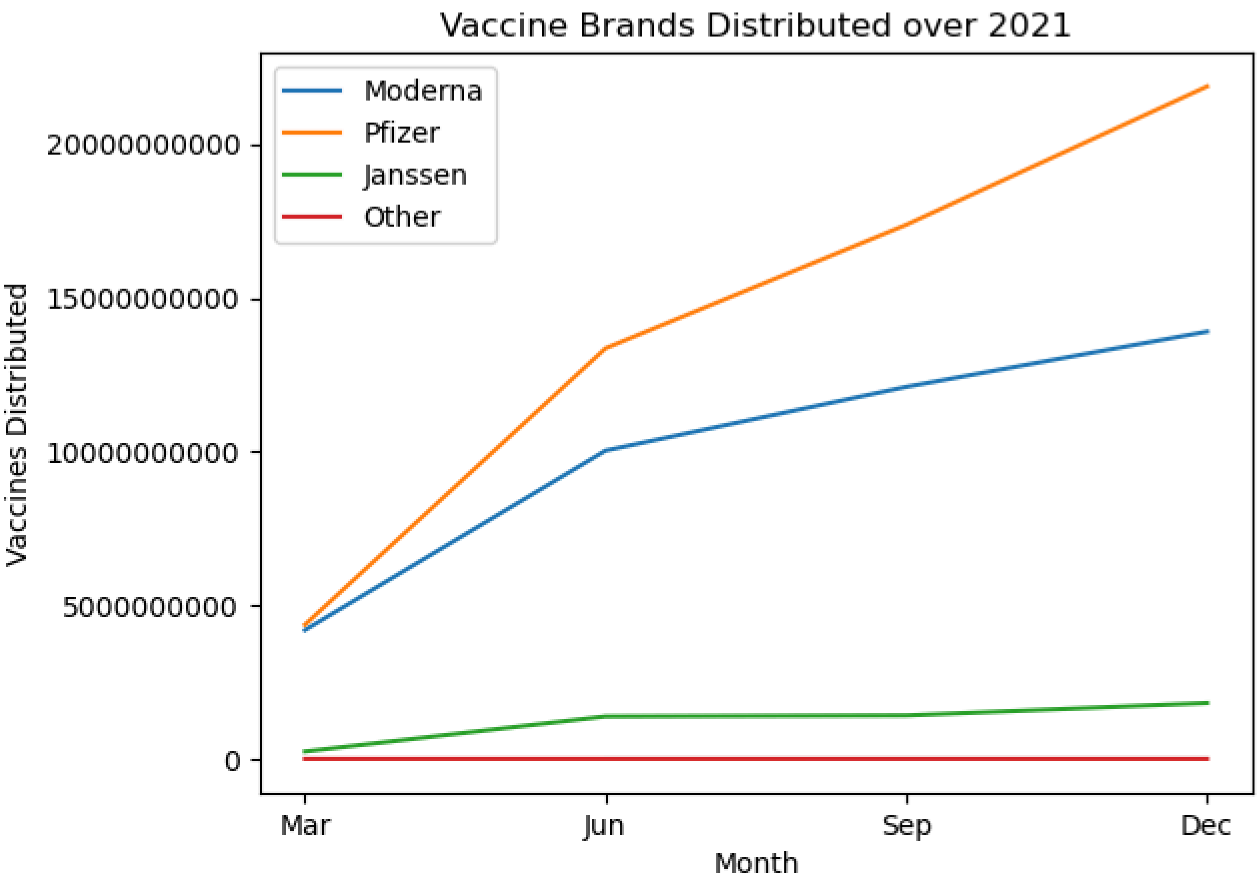
```
Text(0, 0.5, 'Vaccines Distributed (millions)')
```



Overall, Pfizer and Moderna drastically beat the competing brands in the number of vaccines distributed. One thing that could explain this is those two brands were the first to be FDA approved for distribution so most people who wanted to get vaccinated against COVID-19 as soon as possible would choose one of these two brands. Those who were skeptical may have waited longer for other brands to be approved as well before making their decision. Janssen was a third competitor that still did not come close to the top 2, but Novavax and other brands did not distribute enough vaccines to appear on the graph.

In [775…

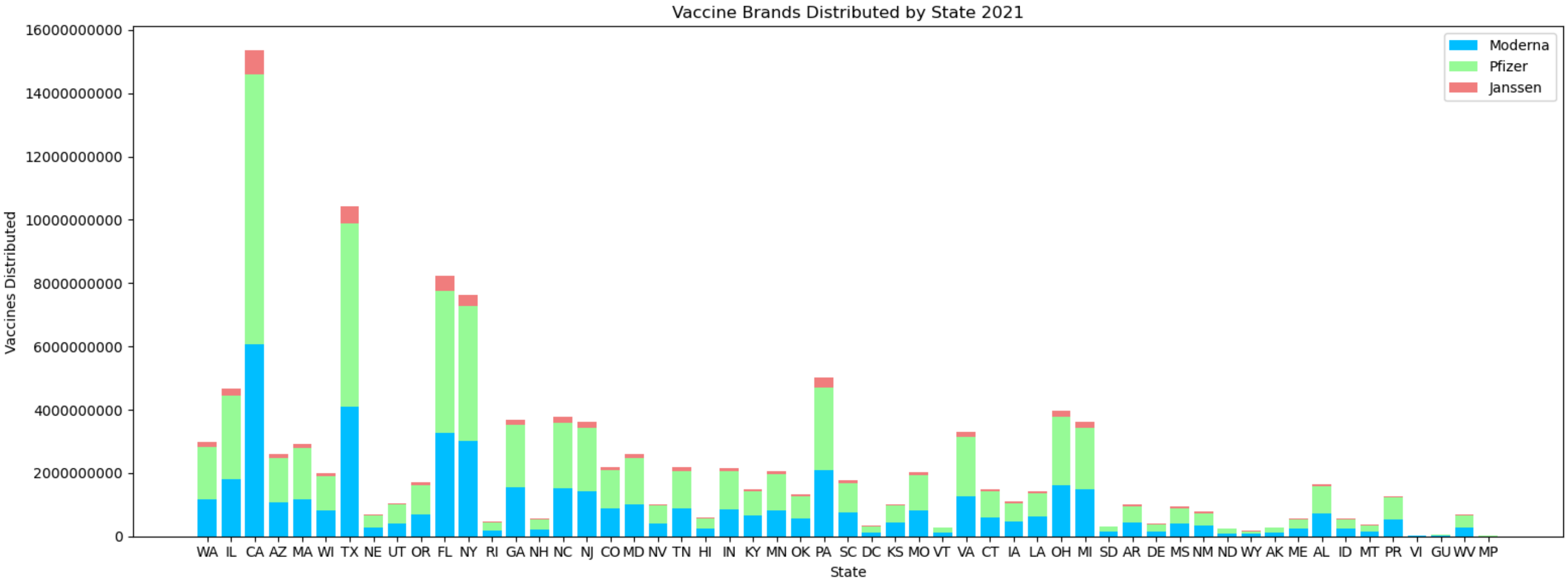
```
plt.plot(month_nums,moderna_vacs, label="Moderna")
plt.plot(month_nums, pfizer_vacs, label="Pfizer")
plt.plot(month_nums, janssen_vacs, label="Janssen")
plt.plot(month_nums, other_vacs, label="Other")
plt.title("Vaccine Brands Distributed over 2021")
plt.xlabel("Month")
plt.ylabel("Vaccines Distributed")
plt.legend()
plt.ticklabel_format(style='plain', axis='y')
```



Once again, Pfizer and Moderna dominate the distributions in 2021, with Pfizer having a steeper incline, Moderna following somewhat close behind, and other brands falling far below either.

In [900…

```
plt.figure(figsize=(15,6))
plt.bar(state_abbr, moderna_all, label="Moderna", color="deepskyblue")
plt.bar(state_abbr, pfizer_all, bottom=moderna_all, label="Pfizer", color="palegreen")
plt.bar(state_abbr, janssen_all, bottom=[sum(x) for x in zip(moderna_all, pfizer_all)], label="Janssen", c
plt.legend()
plt.xlabel("State")
plt.ylabel("Vaccines Distributed")
plt.title("Vaccine Brands Distributed by State 2021")
plt.tight_layout()
plt.ticklabel_format(style='plain', axis='y')
plt.show()
```



Pfizer and Moderna are the two major vaccine brands seen in each state, but Pfizer appears to beat out Moderna in most states. Janssen makes up a very small proportion of each state's brand distrubtions.

Discussion and Conclusion

As seen in the above results, COVID-19 spread in a similar fashion in each state and each region in the US, with an increase after March, plateau in the early summer months, and a spike followed by a sudden downturn around October. The South as a region had the highest peak, but the top 5 states in number of cases reported stayed relatively consistent each quarter in 2020.

In Michigan, Wayne and Oakland Counties had the highest number of cases overall in 2020 and they also had the highest populations and GDPs in the state. This indicates a connection between the spread of the disease, population, and economic status.

The states consistently in the top 5 of highest cases reported were also the first to rollout the vaccine when it became approved in December 2020. This spike in vaccinations aligned with the sudden decrease in cases reported after October 2020.

Overall, Pfizer came out on top as the brand distributed most in the years following the initial 2020 shutdown with Moderna as a somewhat close second. Throughout 2021, Moderna and Pfizer saw steady increases in distribution while other brands were extremely low in comparison.

The main obstacle I encountered was figuring out how to compare vaccine and case data when most of the dates did not overlap. In the end, I only compared in the small time frame where the dates did overlap and the rest of my analysis separated the two datasets. Next time, I would spend more time looking for datasets and trying to do more with prediciting future vaccine and case rates.

References

“GDP by County, Metro, and Other Areas.” GDP by County, Metro, and Other Areas | U.S. Bureau of Economic Analysis (BEA), www.bea.gov/data/gdp/gdp-county-metro-and-other-areas. Accessed Dec. 2024.

Srk. “Covid-19 in USA.” Kaggle, 7 Dec. 2020, www.kaggle.com/datasets/sudalairajkumar/covid19-in-usa.

Volk, Anastasia. “Covid-19 Vaccinations in the US.” Kaggle, 27 Mar. 2024, www.kaggle.com/datasets/volkanastasia/covid-19-vaccinations-in-the-us/data.