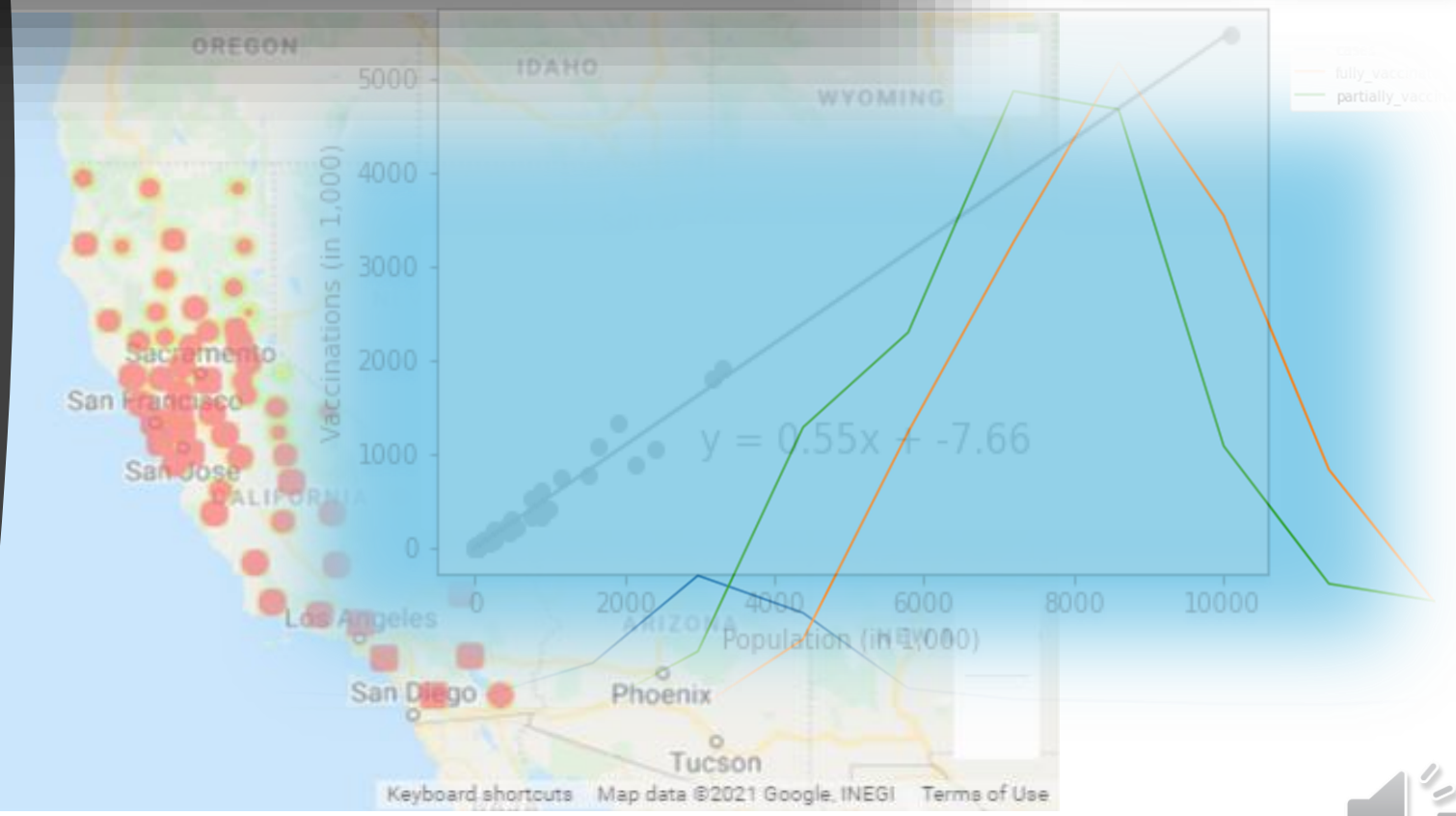
COVID-19 Vaccination

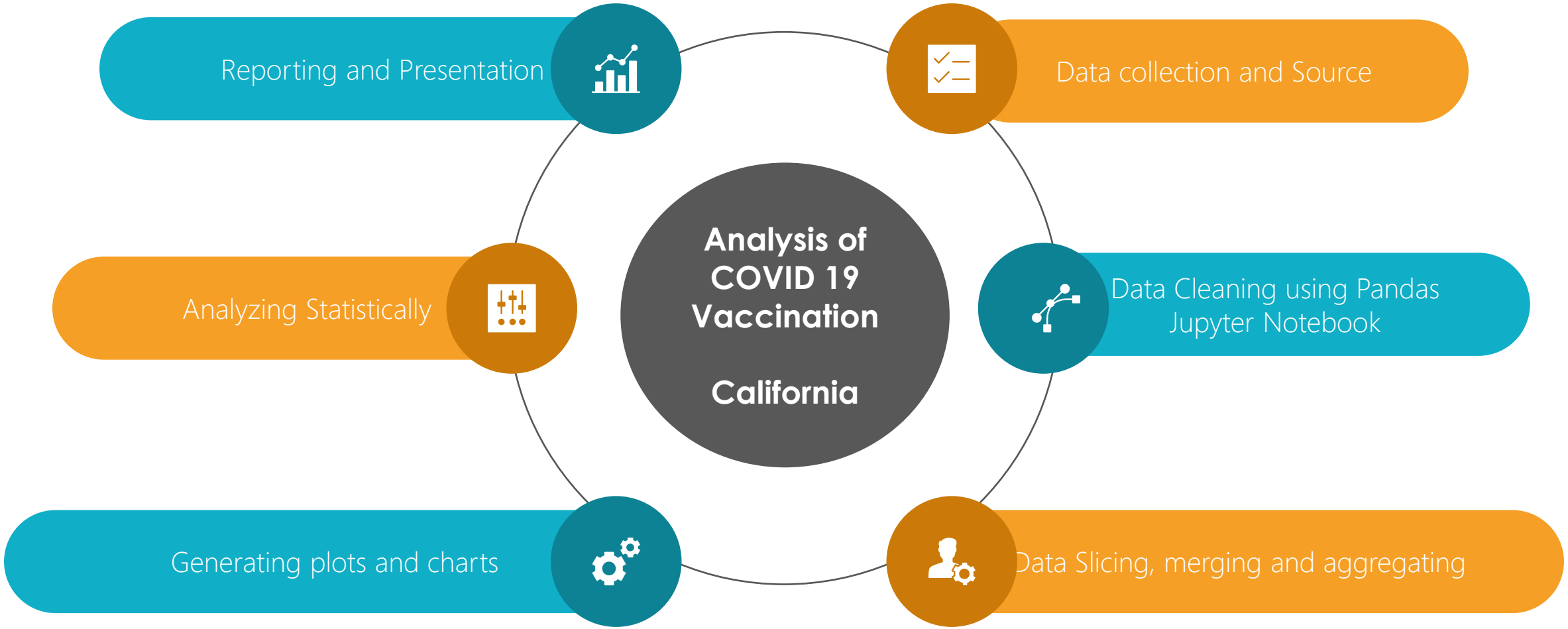
California

Team members:

- Tikaram Subedy
- Kristian Hamilton
- David Koski



Project Outline



Project Plan

California Data

Cases, Vaccination and Demographic data

Data Cleaning and Preparation

Data Cleaning and Preparation using Python and Pandas in Jupyter Notebook

ANALYSIS

Performed Analysis on Overall California, and Counties in Cases, Demographics, and Vaccinations

Trends and Charts

Cases and Vaccination trends were analyzed. Fitted regression models (linear)

Report and Presentation

Prepared data analysis report. Recommendations and further .....

Summarize where and how we found the data used to answer these questions

1. Importing COVID19 data and preparing it for the analysis by slicing and aggregating.
2. Deciding on and calculating a good measure for analysis.
3. Merging datasets and finding correlations among the dataset.
4. Visualizing the analysis results using Pandas and Matplotlib.

What Affects Vaccination in CA?

How is the vaccination trend in overall CA as well counties?

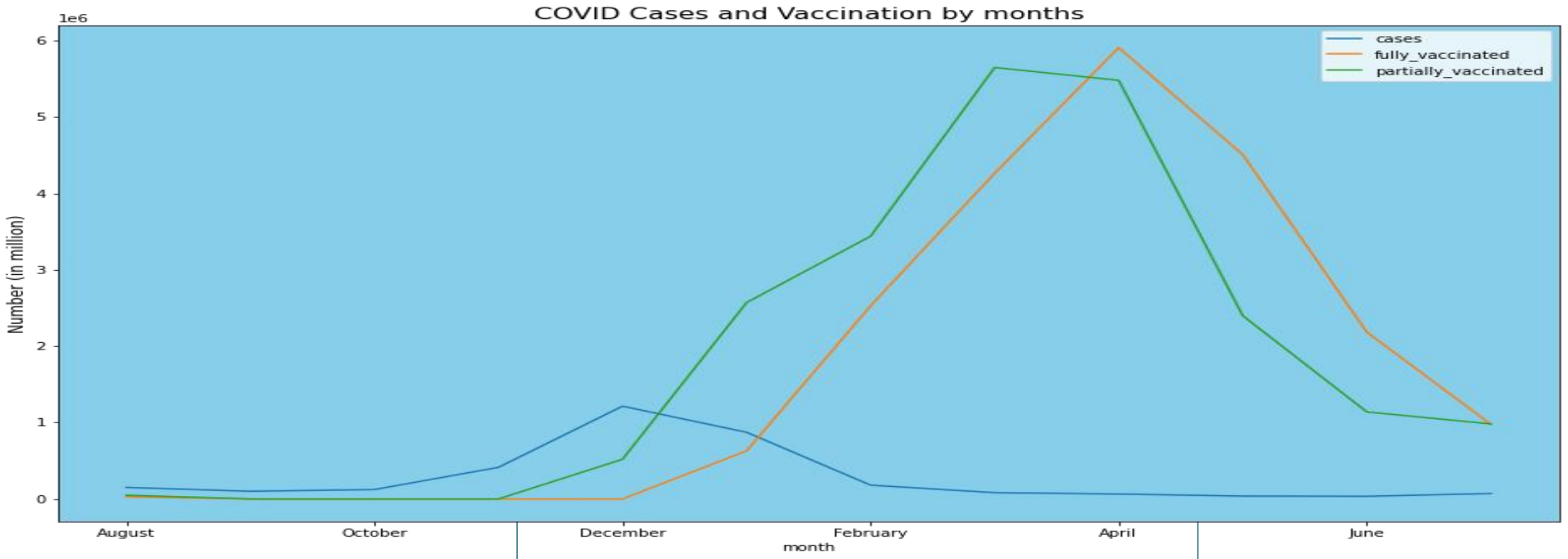
Is there a relationship between cases and deaths that has affected in vaccination?

Do demographics affect on vaccination?

Does location and time affect vaccination?

Analysis

After the vaccination started in December and made available to wider range of population, the cases showed a dropping trend and remaining low.



Total Population in California

39,283,497

Completely Vaccinated

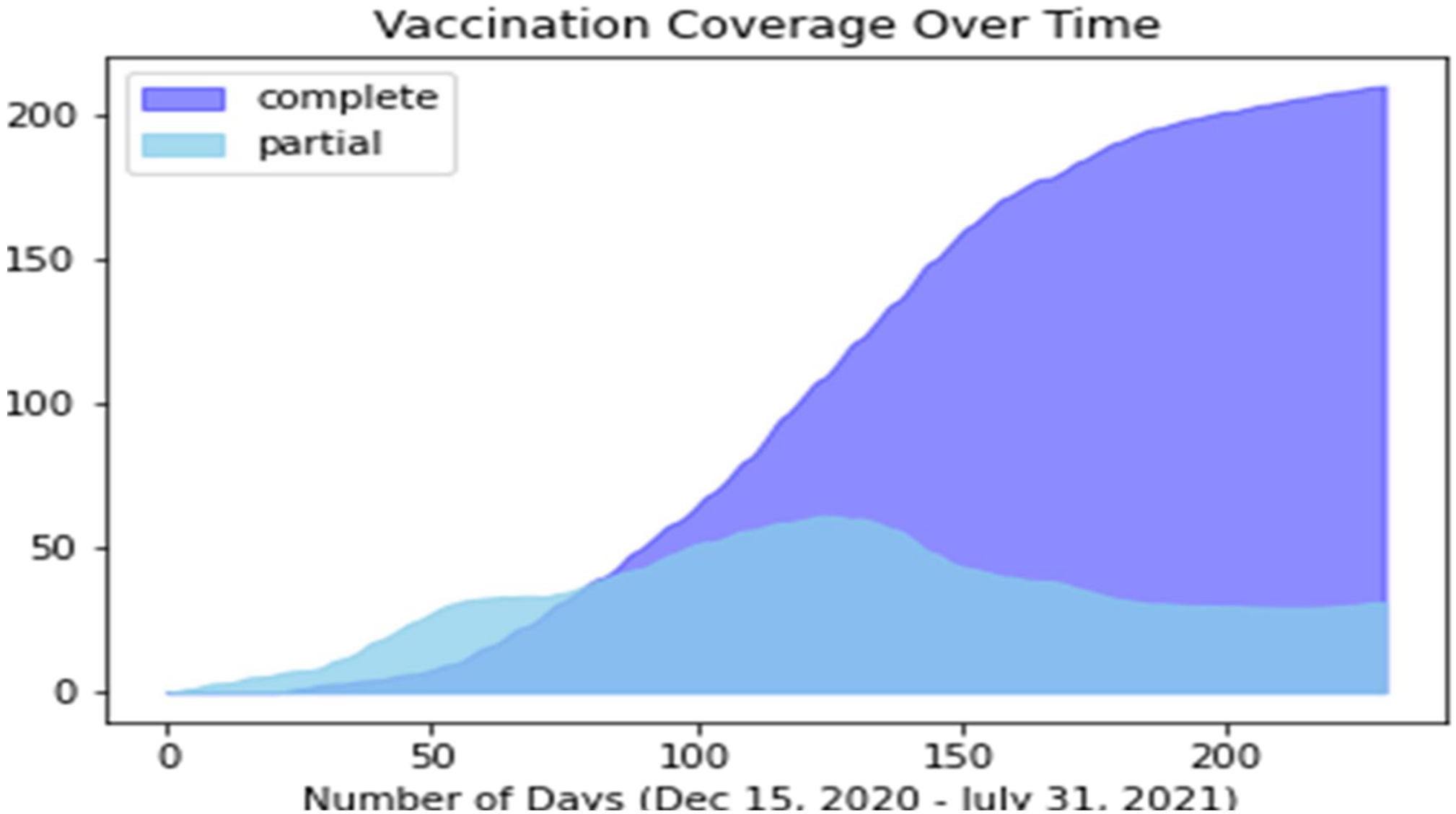
21,025,768

(53.52%)

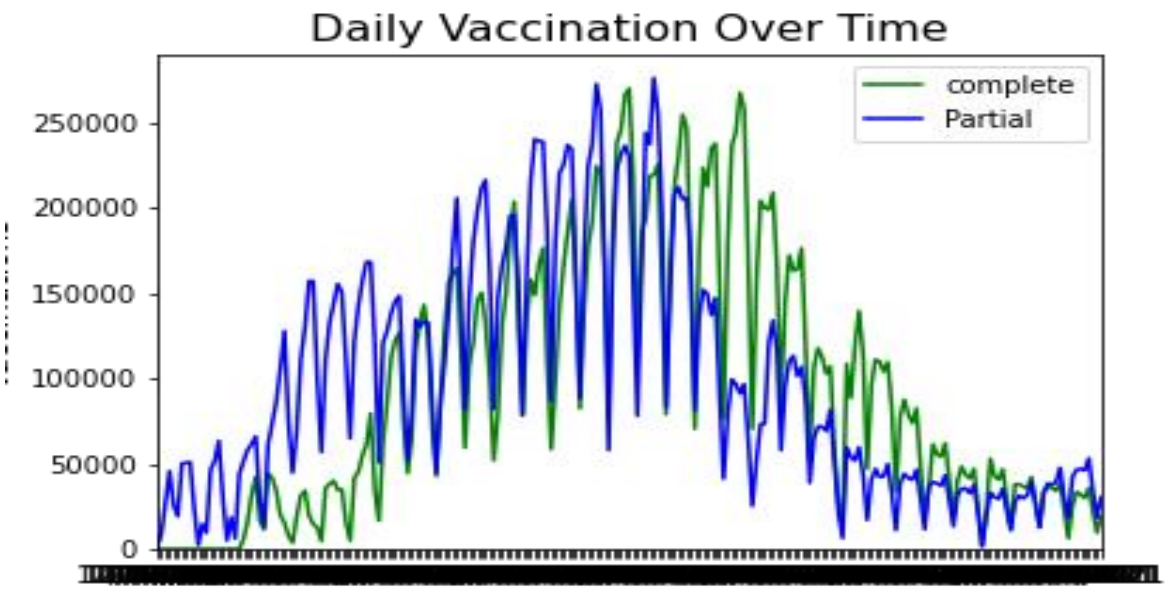
Partially Vaccinated

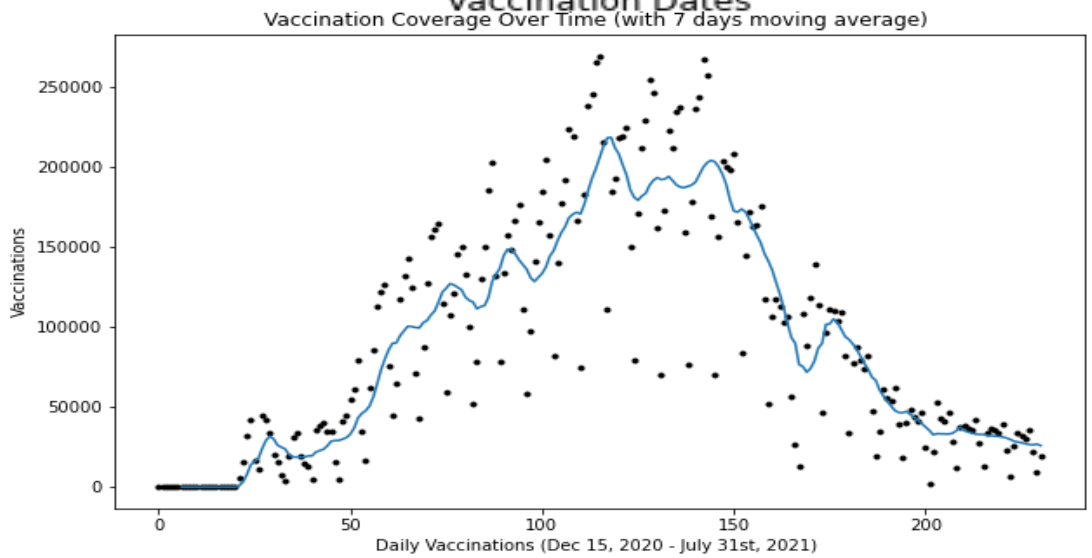
22,224,679

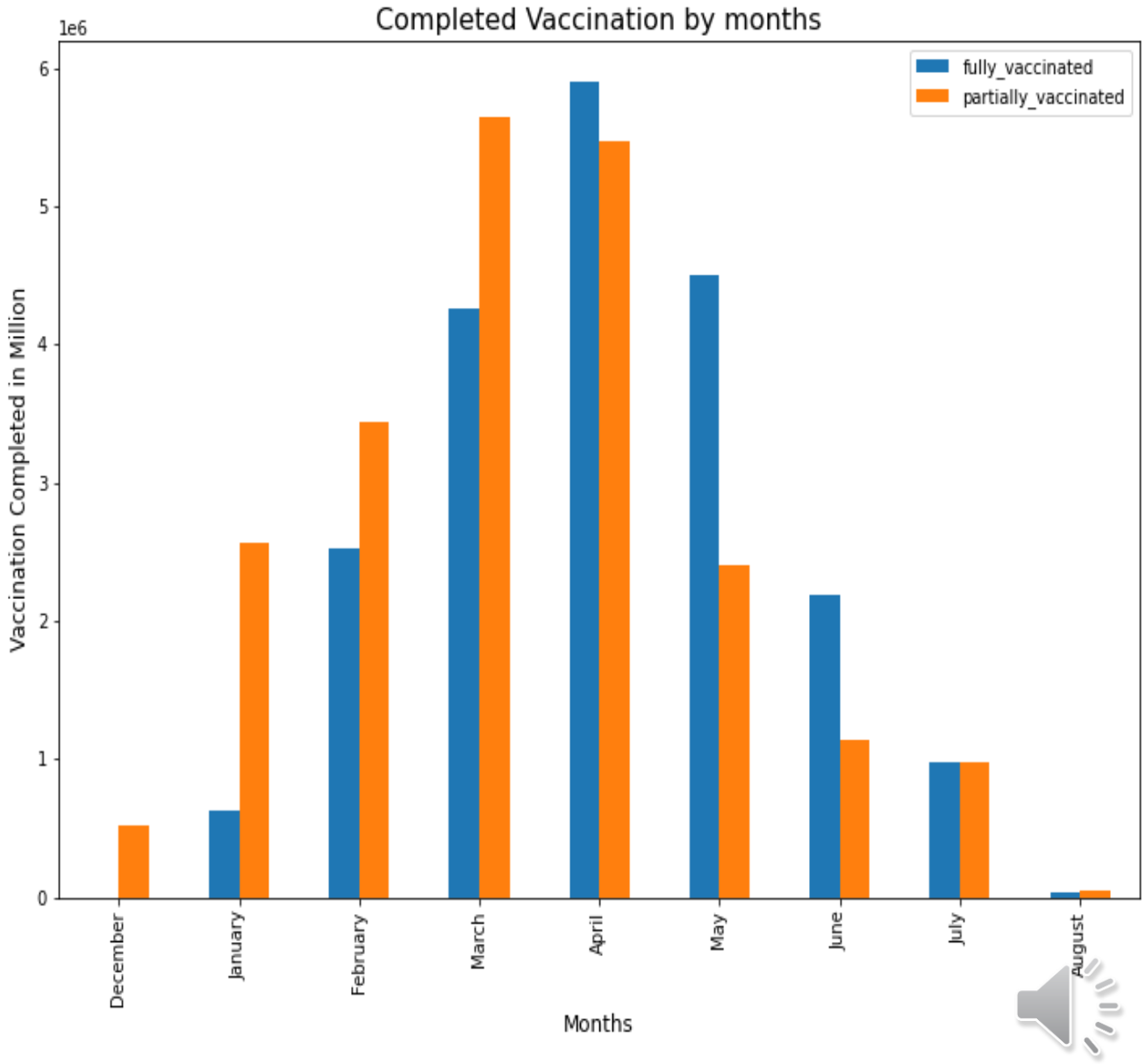
(56.57%)


Vaccination Coverage Over Time
complete
partial
200
150
100
50
0
0
50
100
150
200
Number of Days (Dec 15, 2020 - July 31, 2021)

Analysis

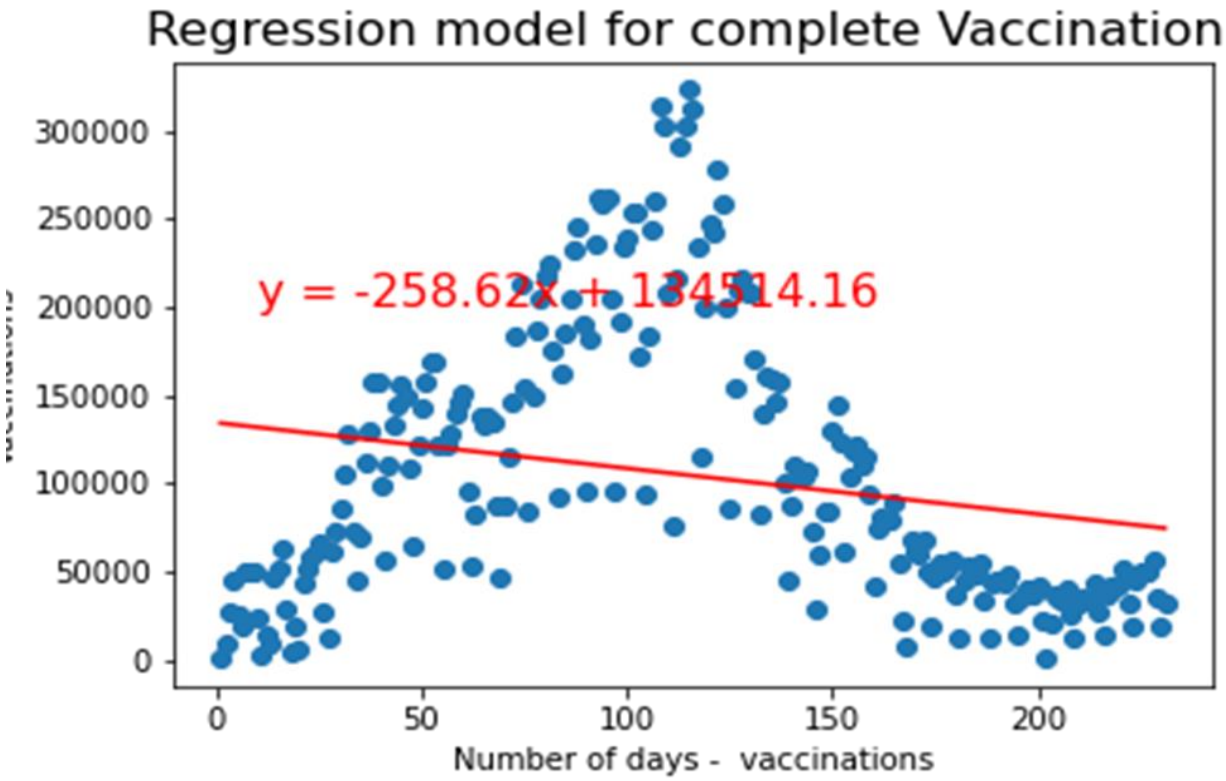

Daily Vaccination Over Time
complete
Partial
250000
200000
150000
100000
50000
0
Vaccination Dates


Vaccination Coverage Over Time (with 7 days moving average)
Vaccinations
250000
200000
150000
100000
50000
0
0
50
100
150
200
Daily Vaccinations (Dec 15, 2020 - July 31st, 2021)

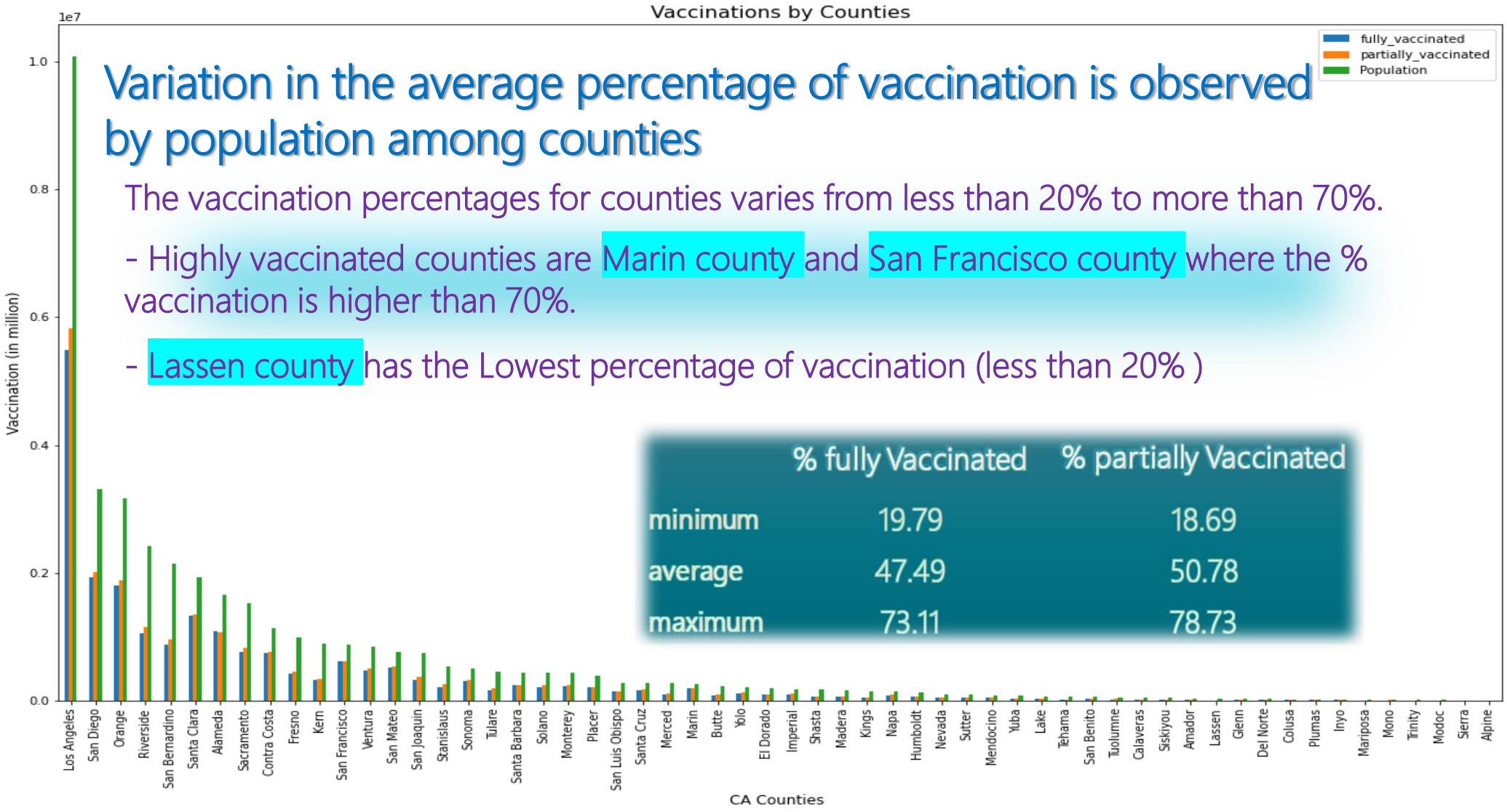

Completed Vaccination by months
1e6
fully_vaccinated
partially_vaccinated
Vaccination Completed in Million
6
5
4
3
2
1
0
December
January
February
March
April
May
June
July
August
Months

Analysis

The statistics shows, it can be estimated that the vaccination in California will be completed in about 521 days from the beginning, while all other compounding factors remain unchanged.



Analysis



Variation in the average percentage of vaccination is observed by population among counties

The vaccination percentages for counties varies from less than 20% to more than 70%.

- Highly vaccinated counties are Marin county and San Francisco county where the % vaccination is higher than 70%.

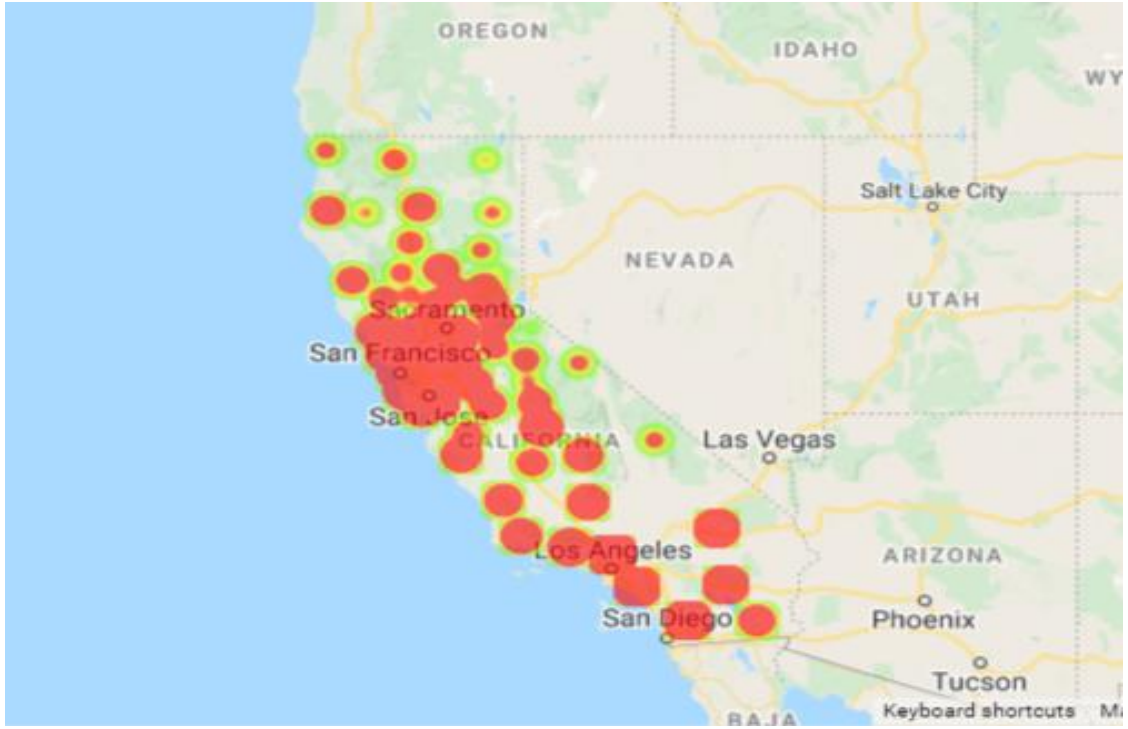
- Lassen county has the Lowest percentage of vaccination (less than 20%)

	% fully Vaccinated	% partially Vaccinated
minimum	19.79	18.69
average	47.49	50.78
maximum	73.11	78.73

Locations and vaccination

Urban and highly populated counties are found to have higher vaccination compared to the rural and less populated counties.

Counties - Complete Vaccination



Counties- higher average vaccination than State average

How strong is the correlation between cases/deaths, and vaccination rates?

CDC

- I downloaded all of the information on COVID cases from the CDC website, and then got to work cleaning it.

USAFacts.org

- I found the data on COVID deaths in California counties from USAFacts.org. I discovered this site after seeing that they were referenced as a trusted data source in an NPR (National Public Radio) article I was reading.

Data Cleanup David

- Describe the data exploration and cleanup process

CDC Cleaning

In [ ]:
```
import pandas as pd
import json
import matplotlib.pyplot as plt
import numpy as np
import os
import csv
```

In [ ]:
```
#read in all CDC COVID data for the US as of early August
csv_cdcpath = "./covid19_cdc_us_data.csv"
cdc_data = pd.read_csv(csv_cdcpath, encoding="utf-8")
cdc_data_df = pd.DataFrame(cdc_data)
cdc_data_df.head()
```

In [ ]:
```
#filter to only include data for CA
ca_cdc_data = cdc_data_df.loc[cdc_data_df['res_state'] == 'CA' ]
ca_cdc_data.head()
```

In [ ]:
```
#filter again to only include month, state, county, age group, current status, sex, race, and ethnicity
ca_cdc_data_short = ca_cdc_data[['case_month', 'res_state', 'res_county',
                                 'age_group', 'current_status', 'sex', 'race', 'ethnicity' ]]
ca_cdc_data_short.head(2)
#write the narrowed results to a csv and start a new notebook so that the computer
#doesn't have to read a 4GB file every time the kernel is restarted

ca_output_data = 'ca_output_data.csv'
ca_cdc_data_short.to_csv(ca_output_data, index = False)
```

CA CDC Data Cleaning

```
#extract only the data for August 2020 on, was originally just for 2021
ca_cases_aug20_on = ca_covid.loc[(ca_covid['case_month'] == '2020-08') |(ca_covid['case_month'] == '2020-09') |
                           (ca_covid['case_month'] == '2020-10') |(ca_covid['case_month'] == '2020-11') |
                            (ca_covid['case_month'] == '2020-12') |
                             (ca_covid['case_month'] == '2021-01') | (ca_covid['case_month'] == '2021-02') |
                             (ca_covid['case_month'] == '2021-03') | (ca_covid['case_month'] == '2021-04')
                           | (ca_covid['case_month'] == '2021-05') | (ca_covid['case_month'] == '2021-06')
                           | (ca_covid['case_month'] == '2021-07') ]

#extract just Dec 2020 on for group members to potentially use
ca_cases_2021 = ca_cases_aug20_on.copy()
ca_cases_2021 = ca_cases_2021.loc[(ca_cases_2021['case_month'] == '2020-12') |
                            (ca_cases_2021['case_month'] == '2021-01') | (ca_cases_2021['case_month'] == '2021-02') |
                             (ca_cases_2021['case_month'] == '2021-03') | (ca_cases_2021['case_month'] == '2021-04')
                           | (ca_cases_2021['case_month'] == '2021-05') | (ca_cases_2021['case_month'] == '2021-06')
                          | (ca_cases_2021['case_month'] == '2021-07') ]
##print(ca_cases_aug20_on.head())
#print(ca_cases_2021.head())
```

```
#write out the modified dataframe
ca_cases_aug20_on.head()
ca_cases_aug20_on.to_csv('./Resources/CA_cases_by_county/ca_aug2020_on_case_data.csv',
                    index=False,header=True)
```

```
#convert the case_month to month names so that it can merge with Tikaram
#use DatetimeIndex so that computer does not time out in for loop
ca_cases_aug20_on['month'] = pd.DatetimeIndex(ca_cases_aug20_on['case_month']).month_name()
ca_cases_2021['month'] = pd.DatetimeIndex(ca_cases_2021['case_month']).month_name()
```

CategoricalDtype Sorting

```
#This cell is will organize the months we are analyzing by calendar placement
# instead of alphabetical order when I plot the data

month_order = CategoricalDtype(['December', 'January', 'February', 'March', 'April',
                                'May', 'June', 'July'], ordered=True)

ca_merged_df['month'] = ca_merged_df['month'].astype(month_order)
ca_merged_df = ca_merged_df.rename(columns={'month':'Month', 'fully_vaccinated':'Full Vaccinations',
                           'partially_vaccinated':'Partial Vaccinations','cumulative_fully_vaccinated':'Cumulative Fully Va
                                        'cumulative_at_least_one_dose':'People with at Least One Dose'})
```

Manually Code Cases per Month

```
#since since using groupby followed by .sum()
#is taking so long, make variables for the case count each month

counts = ca_cases_aug20_on['month'].value_counts()

aug = counts['August']
sept = counts['September']
octo = counts['October']
nov = counts['November']
dec = counts['December']
jan = counts['January']
feb = counts['February']
mar = counts['March']
apr = counts['April']
may = counts['May']
june = counts['June']
july = counts['July']
month_list = counts.tolist()
ordered_month_list = [151345,100714,123809,415153,1216142,873586,181288,82660,65037,37661,34899,71623]
#print(month_list)
print(counts)
```

Finding Monthly Death Totals

In [6]:

```
#create a dataframe for deaths by month for the entire state
#from December 2020 on so that it can be compared with vaccination/cases.
#since the original df is aggregates it is necessary to subtract from the sum of
#each month the sum of the month before it
dec = deaths_by_county['December'].sum() - deaths_by_county['November'].sum()
jan = deaths_by_county['January'].sum() - deaths_by_county['December'].sum()
feb = deaths_by_county['February'].sum() - deaths_by_county['January'].sum()
mar = deaths_by_county['March'].sum() - deaths_by_county['February'].sum()
apr = deaths_by_county['April'].sum() - deaths_by_county['March'].sum()
may = deaths_by_county['May'].sum() - deaths_by_county['April'].sum()
june = deaths_by_county['June'].sum() - deaths_by_county['May'].sum()
july = deaths_by_county['July'].sum() - deaths_by_county['June'].sum()

#print(dec)
#print(jan)
#print(feb)
#print(mar)
#print(apr)
#print(may)
#print(june)

ca_deaths_by_month = pd.DataFrame({
    'month': ['December','January','February',
    'March','April','May','June','July'],
    'Deaths':[dec,jan,feb,mar,apr,may,june,july]
})

ca_deaths_by_month
```

Out[6]:

	month	Deaths
0	December	6171
1	January	15311
2	February	11282
3	March	5799

Merging the Cleaned Dataframes

```
vac_cleaned_grouped = vac_cleaned.groupby('month').sum()[['fully_vaccinated','partially_vaccinated']]
vac_cleaned_grouped_df = pd.DataFrame(vac_cleaned_grouped[['fully_vaccinated','partially_vaccinated']])

ca_merged_df = pd.merge(ca_deaths_by_month,vac_cleaned_grouped_df, on='month')
ca_merged_df = pd.merge(ca_merged_df,cases_by_county,on='month')
ca_merged_df = pd.merge(ca_merged_df,cumulative_fully_vac_grouped_df,on='month')
ca_merged_df
```

Out[11]:

	month	Deaths	fully_vaccinated	partially_vaccinated	Cases	cumulative_fully_vaccinated	cumulative_at_least_one_dose
0	December	6171	75	520210	1216142	77	539074
1	January	15311	631193	2571205	873586	648991	3188197
2	February	11282	2530930	3441366	181288	3237951	6783529
3	March	5799	4278021	5655057	82660	7592057	13100499
4	April	2689	5917068	5482020	65037	13603297	19420686
5	May	1329	4517693	2401439	37661	18173799	22118245
6	June	1078	2187579	1139761	34899	20388307	23475751
7	July	917	981594	986489	71623	21386589	24590667

In [ ]:

```

```

In [12]:

```
#This cell is will organize the months we are analyzing by calendar placement
# instead of alphabetical order when I plot the data

month_order = CategoricalDtype(['December', 'January', 'February', 'March', 'April',
                                'May', 'June', 'July'], ordered=True)

ca_merged_df['month'] = ca_merged_df['month'].astype(month_order)
ca_merged_df = ca_merged_df.rename(columns={'month':'Month', 'fully_vaccinated':'Full Vaccinations',
                            'partially_vaccinated':'Partial Vaccinations','cumulative_fully_vaccinated':'Cumulative Fully Va
                                        'cumulative_at_least_one_dose':'People with at Least One Dose'})

```

Describe the Analysis Process

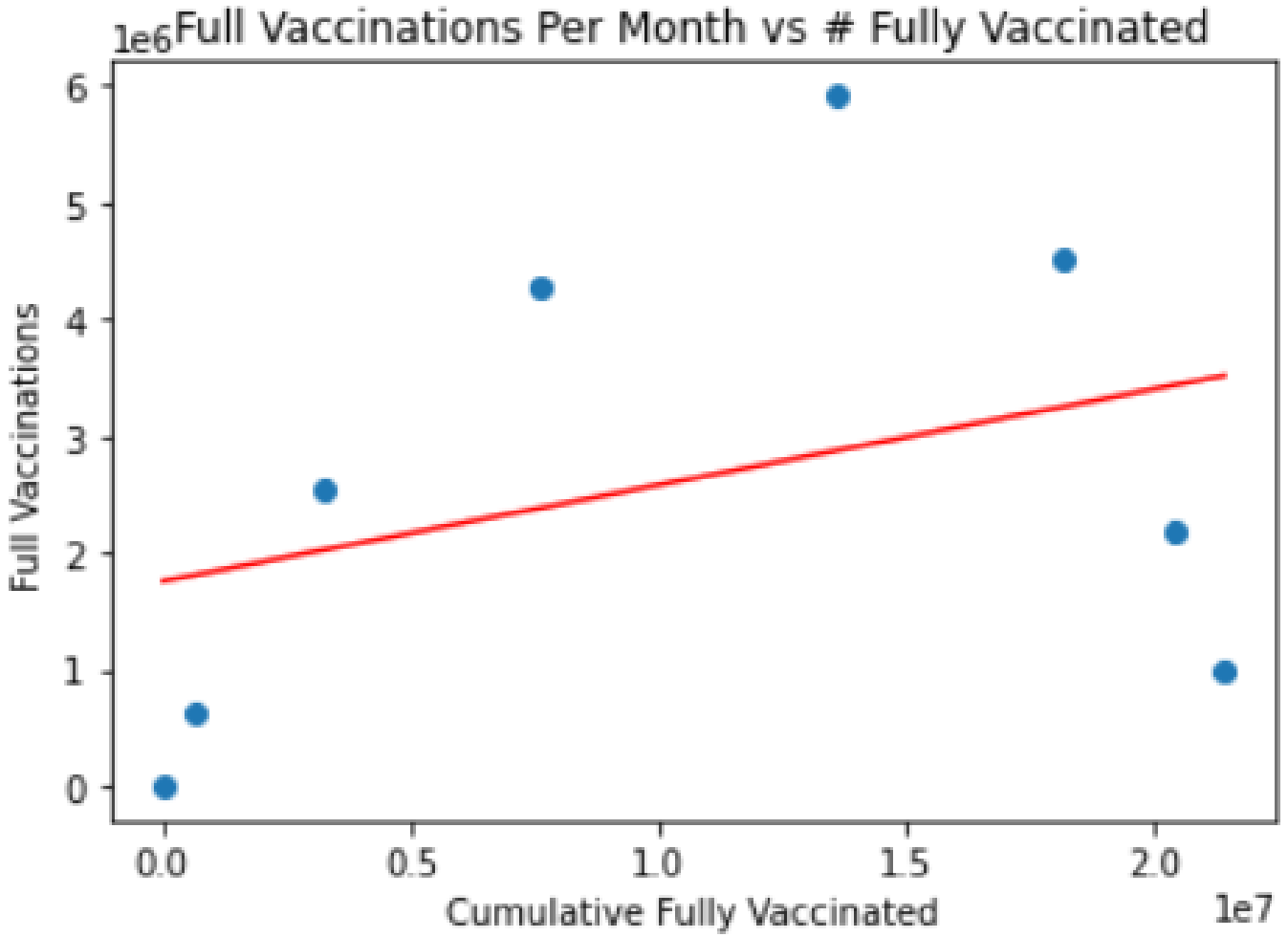
- Screen shots of code and output
- Describe the analysis process

Fully Vaccinated per Month vs. Fully Vaccinated Total Code

```
#show linear regression of vaccinations per month and total full vaccinations in California

x_values = ca_merged_df['Cumulative Fully Vaccinated']
y_values = ca_merged_df['Full Vaccinations']
(slope, intercept, rvalue, pvalue, stderr) = linregress(x_values, y_values)
regress_values = x_values * slope + intercept
line_eq = "y = " + str(round(slope,2)) + "x + " + str(round(intercept,2))
plt.scatter(x_values,y_values)
plt.plot(x_values,regress_values,"r-")
#plt.annotate(line_eq,(6,10),fontsize=15,color="red")
plt.title("Full Vaccinations Per Month vs # Fully Vaccinated")
plt.xlabel('Cumulative Fully Vaccinated')
plt.ylabel('Full Vaccinations')
print(f"The r-value is: {rvalue}")
print(f"The r-squared is: {rvalue**2}")
plt.savefig('./Resources/CA_covid_deaths_cases_vaccines_combo/Images/linreg_fullper_month_#vacced.png')
plt.show()
```

Fully Vaccinated Per Month vs. Fully Vaccinated Total

```
The r-value is: 0.3486571944487115
The r-squared is: 0.1215618392408466 4
```



Cases vs. Total With One or More Shots Code

```
x_values = ca_merged_df['People with at Least One Dose']
y_values = ca_merged_df['Cases']
(slope, intercept, rvalue, pvalue, stderr) = linregress(x_values, y_values)
regress_values = x_values * slope + intercept
line_eq = "y = " + str(round(slope,2)) + "x + " + str(round(intercept,2))
plt.scatter(x_values,y_values)
plt.plot(x_values,regress_values,"r-")
#plt.annotate(line_eq,(6,10),fontsize=15,color="red")
plt.title('COVID Cases vs. People With at Least One Dose')
plt.xlabel('People with at Least One Dose')
plt.ylabel('COVID Cases')
print(f"The r-value is: {rvalue}")
print(f"The r-squared is: {rvalue**2}")
plt.savefig('./Resources/CA_covid_deaths_cases_vaccines_combo/Images/linreg_cases_#dosed.png')
plt.show()
```

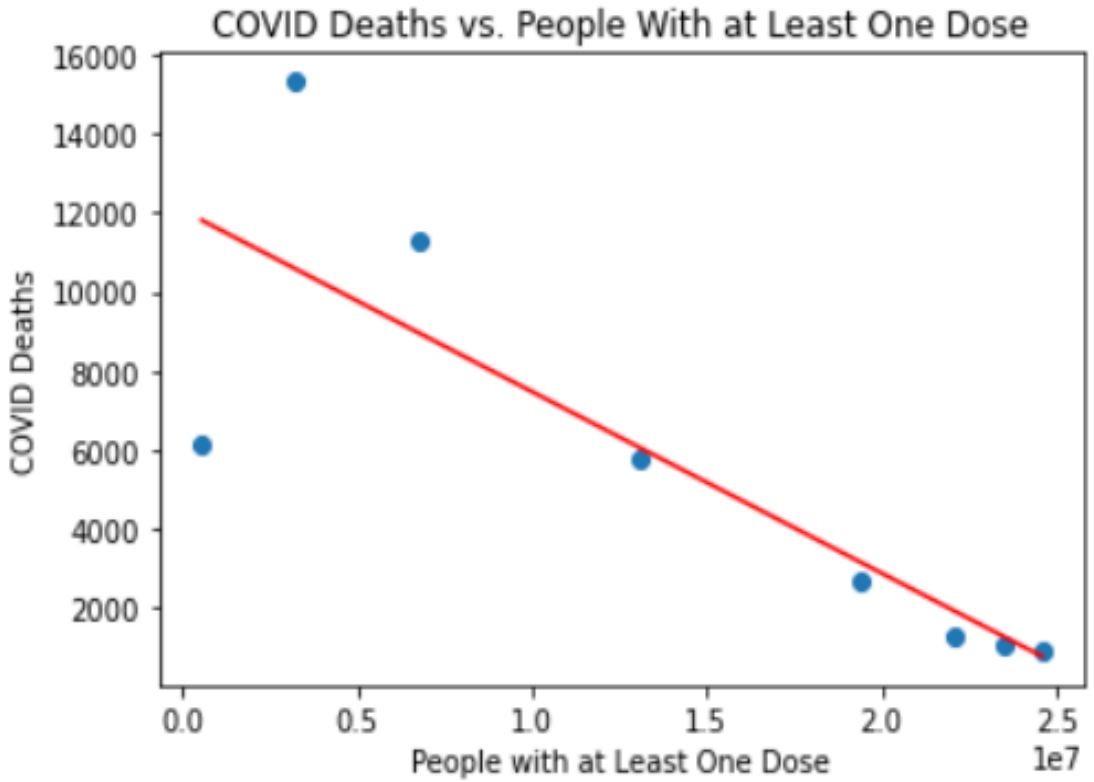
Deaths vs. Total With One or More Shots Code

```python
#show linear regression of deaths and people with at least one shot

x_values = ca_merged_df['People with at Least One Dose']
y_values = ca_merged_df['Deaths']
(slope, intercept, rvalue, pvalue, stderr) = linregress(x_values, y_values)
regress_values = x_values * slope + intercept
line_eq = "y = " + str(round(slope,2)) + "x + " + str(round(intercept,2))
plt.scatter(x_values,y_values)
plt.plot(x_values,regress_values,"r-")
#plt.annotate(line_eq,(6,10),fontsize=15,color="red")
plt.title("COVID Deaths vs. People With at Least One Dose")
plt.xlabel('People with at Least One Dose')
plt.ylabel('COVID Deaths')
print(f"The r-value is: {rvalue}")
print(f"The r-squared is: {rvalue**2}")
plt.savefig('./Resources/CA_covid_deaths_cases_vaccines_combo/Images/linreg_deaths_#dosed.png')
plt.show()
```

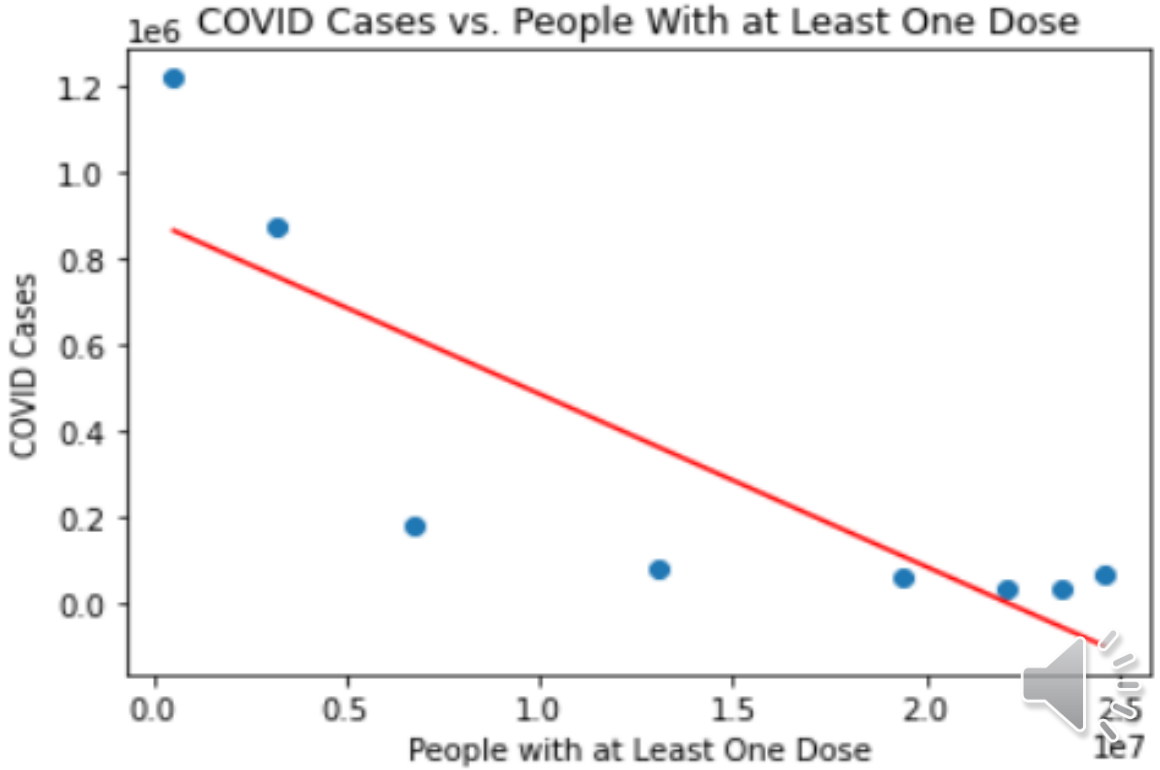
```
The r-value is: -0.8324963312710234
The r-squared is: 0.6930501415797136
```

Cases and Deaths vs. Total With One or More Shots

The r-value is: -0.8324963312710234
The r-squared is: 0.6930501415797136



The r-value is: -0.8364258552960373
The r-squared is: 0.6996082114077076



Do demographics affect vaccination rates?

United States Census Bureau

Search

Advanced Search

All | Tables | Maps | Pages | Microdata | Help | FAQ | Feedback

The 2020 Redistricting Data will be available on data.census.gov no later than September 30th. The redistricting legacy format summary files are now available on the FTP site. Support materials and additic information are available on the Redistricting Data Program's summary file webpage

// Search / Tables / DP02

SELECTED SOCIAL CHARACTERISTICS IN THE UNITED STATES

Survey/Program: American Community Survey **TableID:** DP02 **Product:** 2019: ACS 5-Year Estimates Data Profiles

Notes | Selections | 1 Geo | Years | 6 Topics | Surveys | Codes | Hide | Filter | Transpose | Margin of Error | Restore | Excel | Download | Print | More Data | Map

	California			
Label	**Estimate**	**Margin of Error**	**Percent**	**Percent Margin of Error**
HOUSEHOLDS BY TYPE				
Total households	13,044,266	±20,333	13,044,266	(X)
Married-couple family	6,491,236	±30,996	49.8%	±0.2
With own children of the householder under 18 years	2,801,975	±22,451	21.5%	±0.1
Cohabiting couple household	874,868	±6,012	6.7%	±0.1
With own children of the householder under 18 years	331,408	±4,216	2.5%	±0.1
Male householder, no spouse/partner present	2,260,535	±12,994	17.3%	±0.1
With own children of the householder under 18 years	171,733	±3,524	1.3%	±0.1
Householder living alone	1,390,613	±8,068	10.7%	±0.1
65 years and over	403,763	±4,528	3.1%	±0.1
Female householder, no spouse/partner present	3,417,627	±10,396	26.2%	±0.1
With own children of the householder under 18 years	631,664	±6,030	4.8%	±0.1
Householder living alone	1,715,491	±8,335	13.2%	±0.1
65 years and over	836,525	±5,822	6.4%	±0.1
Households with one or more people under 18 years	4,482,879	±21,313	34.4%	±0.1
Households with one or more people 65 years and over	3,803,822	±9,028	29.2%	±0.1
Average household size	2.95	±0.01	(X)	(X)
Average family size	3.53	±0.01	(X)	(X)
RELATIONSHIP				
Population in households	38,462,235	*****	38,462,235	(X)

Imported .csv as DataFrame:

]:

	GEO_ID	NAME	DP02_0001E	DP02_0001M	DP02_0001PE	DP02_0001PM	DP02_0002E	D
0	id	Geographic Area Name	Estimate!!HOUSEHOLDS BY TYPE!!Total households	Margin of Error!!HOUSEHOLDS BY TYPE!!Total hou...	Percent!!HOUSEHOLDS BY TYPE!!Total households	Percent Margin of Error!!HOUSEHOLDS BY TYPE!!T...	Estimate!!HOUSEHOLDS BY TYPE!!Total households...	Error!!HOU BY TYPE!
1	0500000US06001	Alameda County, California	577177	1744	577177	(X)	292079	
2	0500000US06003	Alpine County, California	350	69	350	(X)	194	
3	0500000US06005	Amador County, California	14594	448	14594	(X)	7954	
4	0500000US06007	Butte County, California	85320	891	85320	(X)	37211	

```
#Fixes header for the census dataframe
newHeader = censusDemo_df.iloc[0]
censusDemo_df = censusDemo_df[1:]
censusDemo_df.columns = newHeader
```

	id	Geographic Area Name	Estimate!!HOUSEHOLDS BY TYPE!!Total households	Margin of Error!!HOUSEHOLDS BY TYPE!!Total households	Percent!!HOUSEHOLDS BY TYPE!!Total households	Percent Margin of Error!!HOUSEHOLDS BY TYPE!!Total households	Estimate!!HOUSEHOLDS BY TYPE!!Total households!!Married-couple family	Erro hou
1	0500000US06001	Alameda County,	577177	1744	577177	(X)	292079	

```
#compile dataframe for use with plots
for x in countyList:
```

```
    county_df.loc[x, "Total Population"] =
        censusDemo_df.loc[
        censusDemo_df['Geographic Area Name'] == (x + ' County, California'), 'Estimate!!ANCESTRY!!Total population'
                          ].sum()*
```

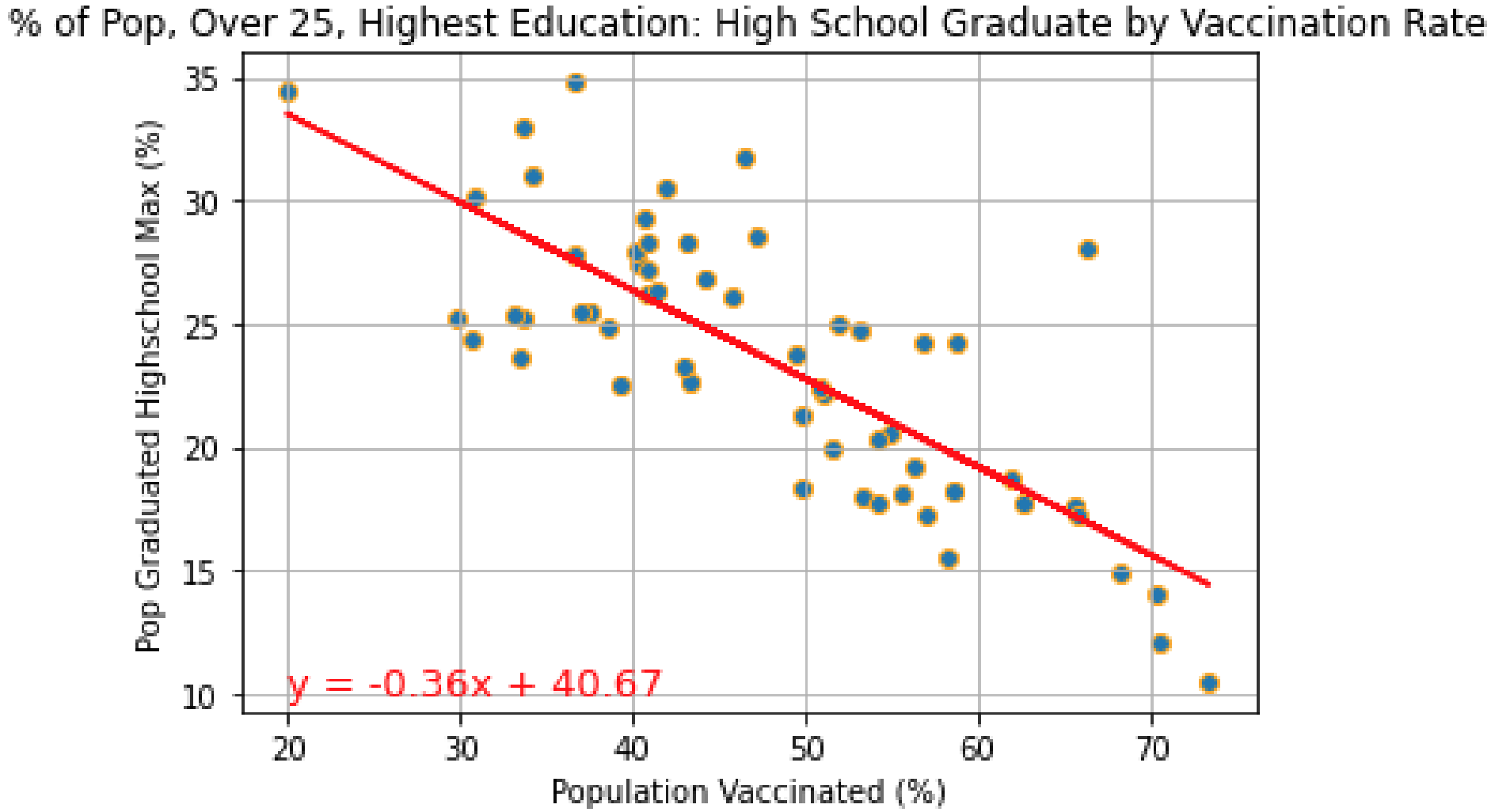
* ValueError: Incompatible indexer with Series

```
#data type for Total Pop needs to be cast as float instead of string to easily calc percentages
county_df['Total Population'] = county_df["Total Population"].apply(pd.to_numeric, downcast = 'float')
county_df.loc[x, "Total Fully Vaccinated"] = CHHS_df.loc[CHHS_df['county'] == x, "fully_vaccinated"].sum()
county_df["% of Pop Fully Vaccinated"] = (county_df["Total Fully Vaccinated"] / county_df["Total Population"]) * 100
```

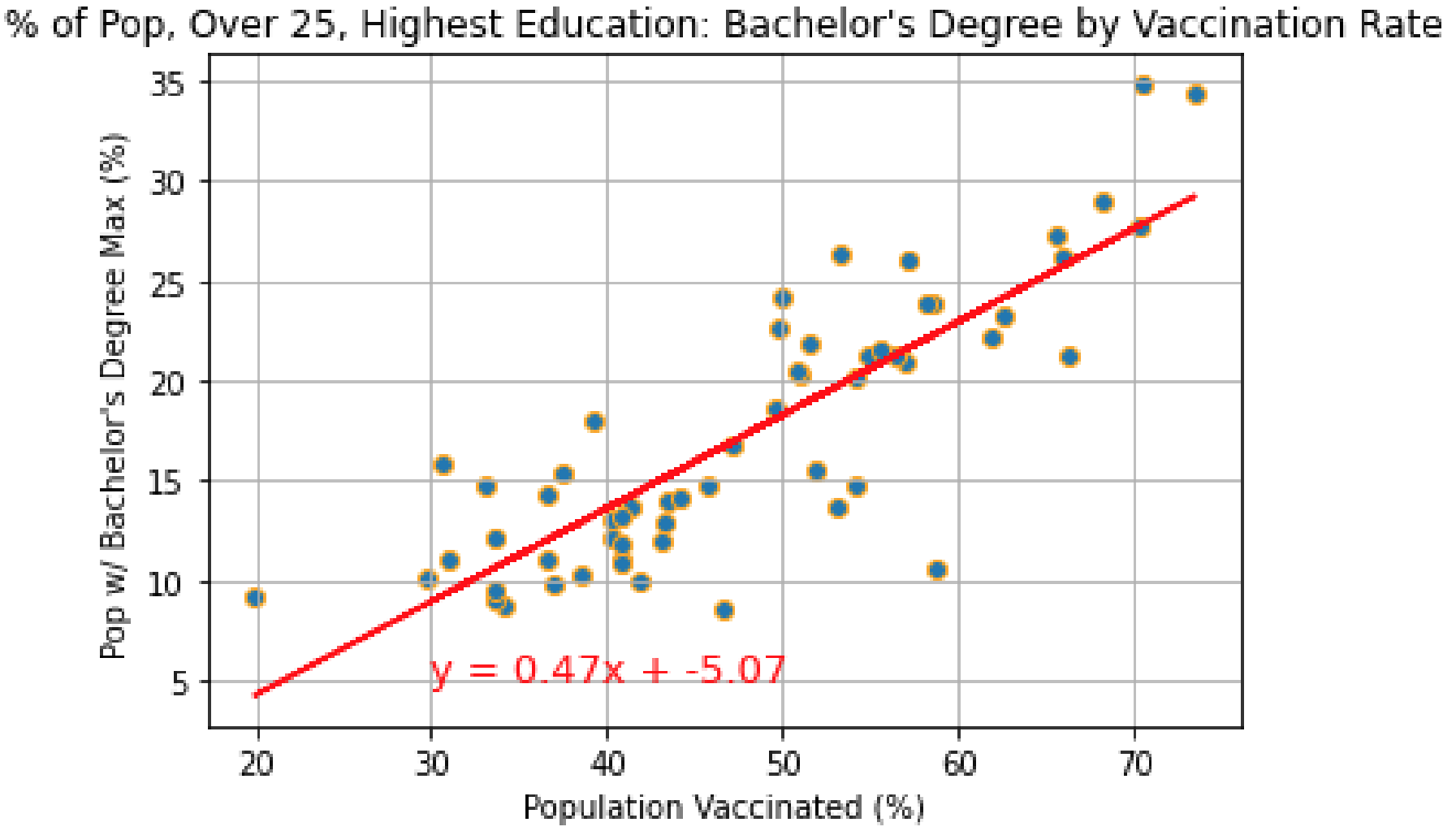
```
county_df
```

County	Total Population	Total Fully Vaccinated	% of Pop Fully Vaccinated	Avg People per Household	Total Households
Alameda	1656754.0	1085847.0	65.540629	2.82	577177
Alpine	1039.0	689.0	66.313763	2.87	350
Amador	38429.0	15481.0	40.284681	2.38	14594
Butte	225817.0	88630.0	39.248595	2.57	85320
Calaveras	45514.0	18304.0	40.216197	2.66	16942
Colusa	21454.0	8758.0	40.822224	2.94	7227

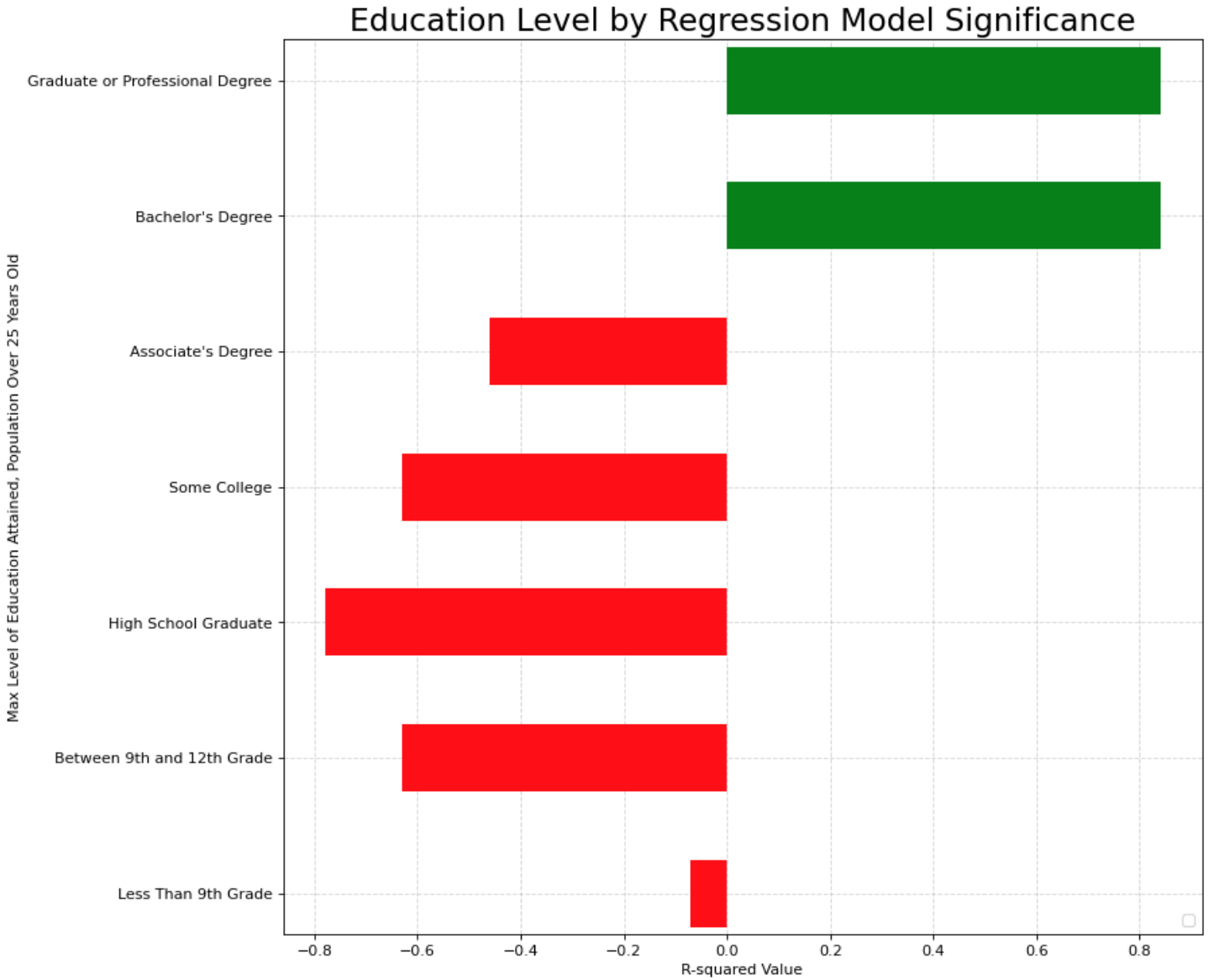
```python
#function for scatter plot and linear regression
def plotScatterLinearRegression(xValues, yValues, title, xLabel, yLabel, textCoordinates):

    #regression formula
    (slope, intercept, rvalue, pvalue, stderr) = linregress(xValues, yValues)
    regressValues = xValues * slope + intercept
    lineEquation = "y = " + str(round(slope, 2)) + "x + " + str(round(intercept, 2))

    #scatter plot and regression line plot
    plt.scatter(xValues, yValues, edgecolor = "orange")
    plt.plot(xValues, regressValues, "r-", color = "red")
    plt.annotate(lineEquation, textCoordinates, fontsize = 13, color = "red")
    plt.xlabel(xLabel)
    plt.ylabel(yLabel)
    plt.title(title)
    print(f"R-squared = {round(rvalue,2)}")
    plt.grid()
    plt.show()
```


% of Pop, Over 25, Highest Education: High School Graduate by Vaccination Rate
Pop Graduated Highschool Max (%)
Population Vaccinated (%)
y = -0.36x + 40.67
35
30
25
20
15
10
20
30
40
50
60
70

$R^2 = -0.78$


% of Pop, Over 25, Highest Education: Bachelor's Degree by Vaccination Rate
Pop w/ Bachelor's Degree Max (%)
Population Vaccinated (%)
y = 0.47x + -5.07
35
30
25
20
15
10
5
20
30
40
50
60
70

$R^2 = 0.84$



Increasing Relationship with Vaccinations

Decreasing Relationship with Vaccinations

.xls available from website

Save As .csv and import as DataFrame

Report of Registration as of July 16, 2021
Registration by County

County	Eligible	Total Registered	Democratic	Republican	American Independent	Green
Alameda	1,078,848	944,570	566,482	103,575	20,359	5,281
Percent		**87.55%**	**59.97%**	**10.97%**	**2.16%**	**0.56%**
Alpine	921	907	394	218	47	7
Percent		**98.48%**	**43.44%**	**24.04%**	**5.18%**	**0.77%**
Amador	26,828	26,268	7,336	12,220	1,294	103
Percent		**97.91%**	**27.93%**	**46.52%**	**4.93%**	**0.39%**
Butte	150,098	125,414	44,403	44,757	5,490	718
Percent		**83.55%**	**35.41%**	**35.69%**	**4.38%**	**0.57%**
Calaveras	36,029	32,406	8,711	14,975	1,688	156
Percent		**89.94%**	**26.88%**	**46.21%**	**5.21%**	**0.48%**
Colusa	12,705	10,008	3,204	4,080	334	22
Percent		**78.77%**	**32.01%**	**40.77%**	**3.34%**	**0.22%**
Contra Costa	759,452	706,597	375,896	131,228	22,414	2,793
Percent		**93.04%**	**53.20%**	**18.57%**	**3.17%**	**0.40%**
Del Norte	18,126	14,879	4,471	5,862	779	96
Percent		**82.09%**	**30.05%**	**39.40%**	**5.24%**	**0.65%**

	County	Eligible	Total Registered	Democratic	Republican	American Independent	Green	Libertarian	Peace and Freedom	Unknown	Other	No Party Preference
0	Alameda	1,078,848	944,570	566,482	103,575	20,359	5,281	5,969	3,748	1	5,495	233,660
1	Percent	NaN	87.55%	59.97%	10.97%	2.16%	0.56%	0.63%	0.40%	0.00%	0.58%	24.74%
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	Alpine	921	907	394	218	47	7	18	2	1	3	217
4	Percent	NaN	98.48%	43.44%	24.04%	5.18%	0.77%	1.98%	0.22%	0.11%	0.33%	23.93%
...	...	...	...	...	...	...	...	...	...	...	...	...
173	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
174	State Total	24,834,943	22,078,290	10,264,695	5,309,040	710,259	88,011	212,686	109,963	118,865	132,121	5,132,650
175	Percent	NaN	88.90%	46.49%	24.05%	3.22%	0.40%	0.96%	0.50%	0.54%	0.60%	23.25%
176	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
177	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
#compile dataframe for use with plots
for x in countyList:
```

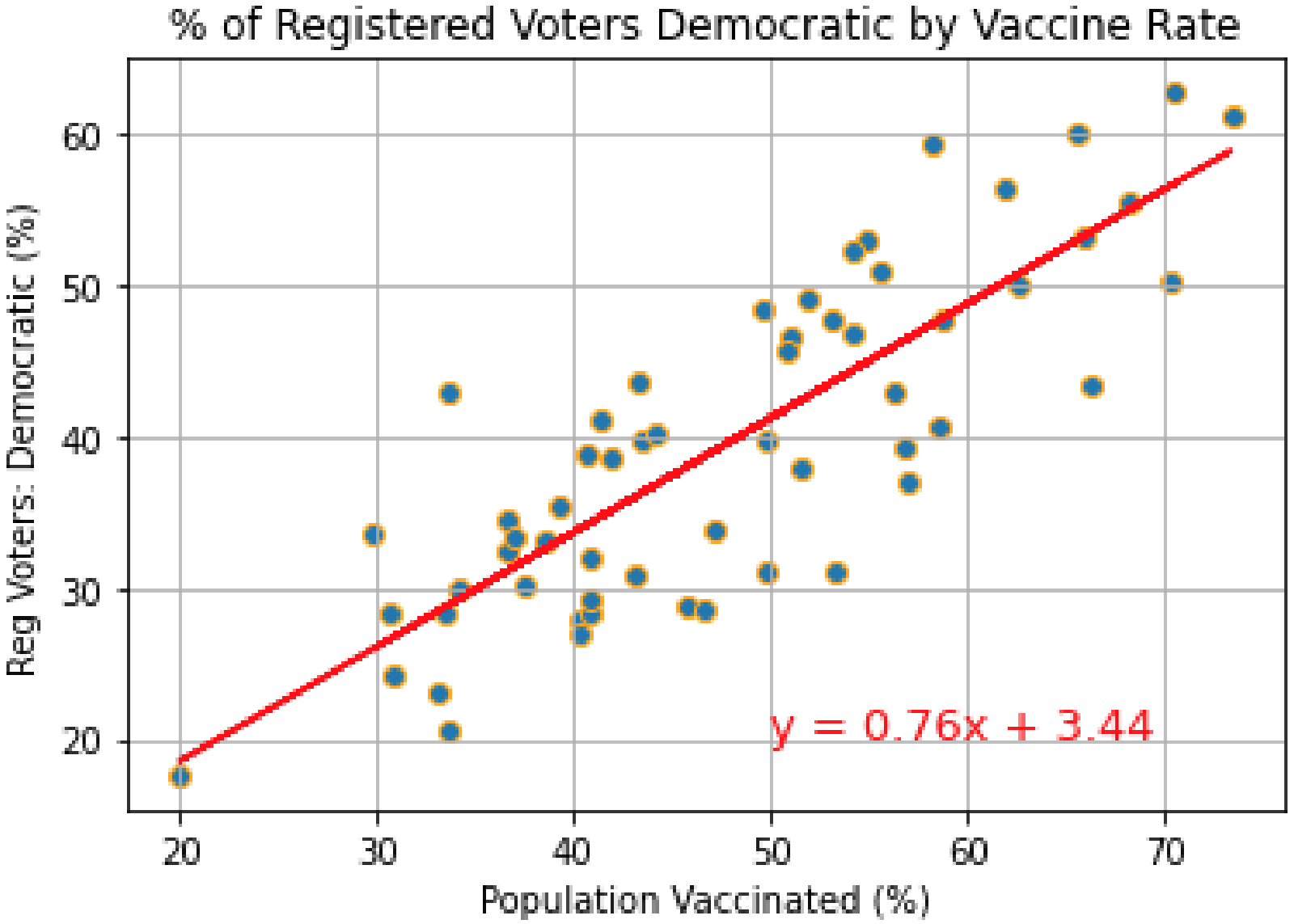
```
    #Voter Registration Pull by County
    county_df.loc[x, "Total Registered Voters"] = caSOS_df.loc[caSOS_df['County'] == x, 'Total Registered'].sum()
    county_df.loc[x, "Democratic"] = caSOS_df.loc[caSOS_df['County'] == x, 'Democratic'].sum()
    county_df.loc[x, "Republican"] = caSOS_df.loc[caSOS_df['County'] == x, 'Republican'].sum()
    county_df.loc[x, "American Independent"] = caSOS_df.loc[caSOS_df['County'] == x, 'American Independent'].sum()
    county_df.loc[x, "Green"] = caSOS_df.loc[caSOS_df['County'] == x, 'Green'].sum()
    county_df.loc[x, "Libertarian"] = caSOS_df.loc[caSOS_df['County'] == x, 'Libertarian'].sum()
    county_df.loc[x, "Peace and Freedom"] = caSOS_df.loc[caSOS_df['County'] == x, 'Peace and Freedom'].sum()
    county_df.loc[x, "No Party Preference"] = caSOS_df.loc[caSOS_df['County'] == x, 'No Party Preference'].sum()
```

```
#Remove commas from dataframe so that it can be converted from string to float
county_df = county_df.replace(',','', regex=True)
```

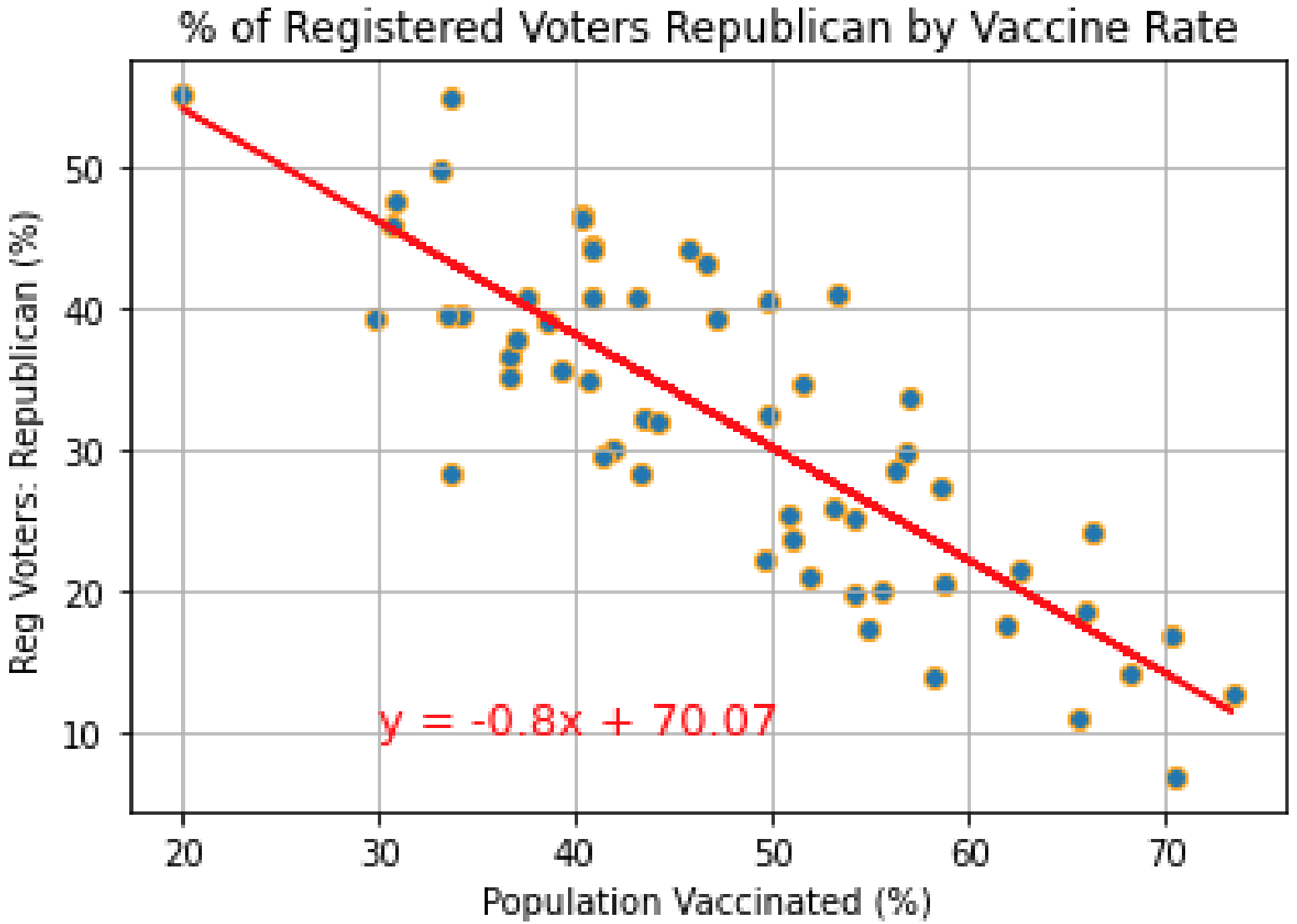
```
county_df["% of Reg Voters: No Party Preference"] =
                                        (county_df["No Party Preference"].apply(pd.to_numeric, downcast = 'float')
                                    / county_df["Total Registered Voters"].apply(pd.to_numeric, downcast = 'float')) * 100
```

```
county_df["% of Reg Voters: No Party Preference"]
```

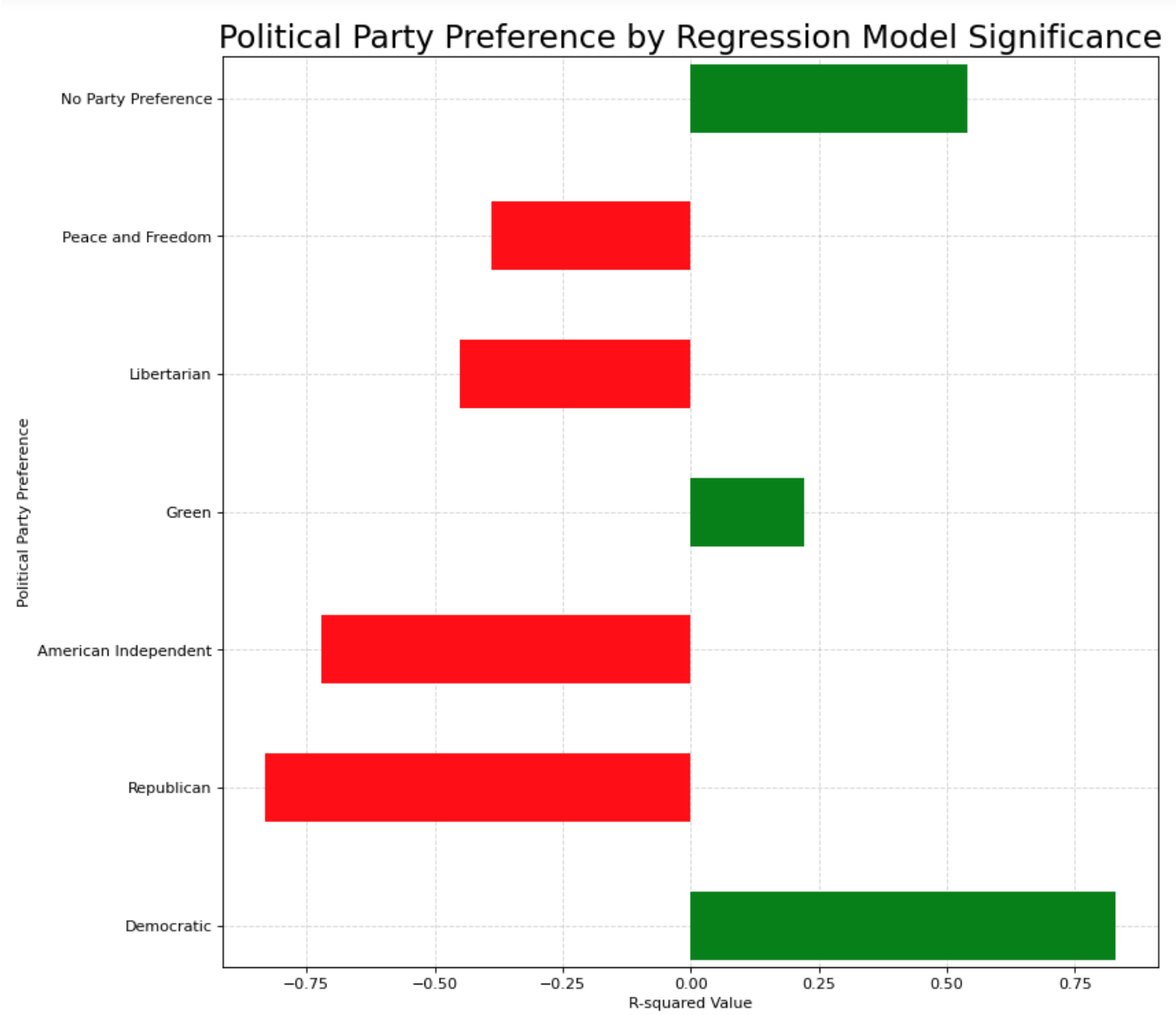
```
County
Alameda          24.737182
Alpine           23.925028
Amador           17.740216
Butte            20.602166
Calaveras        17.728198
Colusa           21.772582
Contra Costa     22.667518
```


% of Registered Voters Democratic by Vaccine Rate
Reg Voters: Democratic (%)
Population Vaccinated (%)
y = 0.76x + 3.44
20
30
40
50
60
70

$R^2 = 0.83$


% of Registered Voters Republican by Vaccine Rate
Reg Voters: Republican (%)
Population Vaccinated (%)
y = -0.8x + 70.07
10
20
30
40
50
20
30
40
50
60
70

$R^2 = -0.83$



Increasing Relationship with Vaccinations

Decreasing Relationship with Vaccinations

Do demographics affect vaccination rates?

Strong R-squared values indicate yes.

Limitations:

Census data is from 2019.
Small sample size with only 58 California Counties.

Recommendations:

It seems like policy encouraging people to take remote college courses may increase vaccination rates slightly in a county. Correlation obviously does not equal causation, however.

Thank You