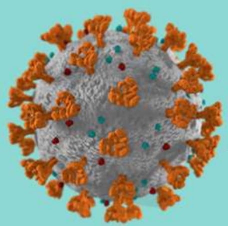
COVID-19 and its Economic Predictors

Joe Rodini

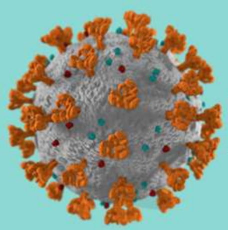
A Project for Data Analytics Bootcamp, University of Oregon, 2022

Project Overview



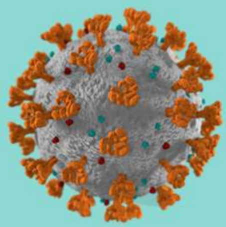
Background

- Coronavirus disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus, first identified in Wuhan, China in December 2019
- It spread to the level of a global pandemic by March 2020
- While a report often cited by the White House in May 2020 estimated a national death toll of 134,000 from the virus, current estimates of the US death toll is over 1,000,000



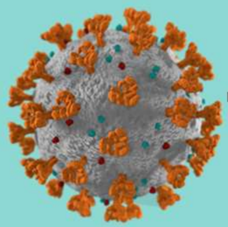
Economic Differences

- In 2021, the top 10 percent of Americans held nearly 70 percent of U.S. wealth, while the bottom 50 percent owned about 2.5 percent of wealth.
- Unemployment rate, defined as the percentage of people of the labor force that is not currently employed but could be, is an indicator of economic health and a signal of potential recession.
- Median Household Income is a well-recognized indicator of poverty, which can affect physical and mental health.
- These two measures will serve as my predictors



Research Question

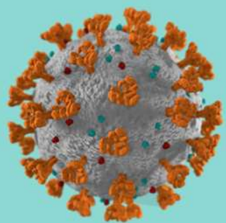
- How well can the economic indicators of unemployment and median income, measured at the county level, predict COVID outcomes?
- COVID outcomes:
  - Cases per 100,000
  - Deaths per 100,000
  - Vaccination Rate



Technologies

- Coding
  - Python, Pandas in Jupyter Notebook; SQLAlchemy
- Database
  - PostgreSQL in PgAdmin
- Visualizations
  - Tableau, Plotly

Working with Data

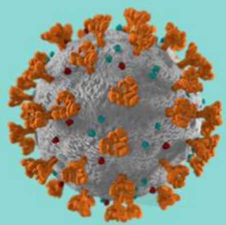


Data Sources

There were four main sources that comprised this analysis:

- Data on county vaccination rates from https://data.cdc.gov/Vaccinations/COVID-19-Vaccinations-in-the-United-States-County/8xkx-amqh
- Data on county cases and deaths numbers from https://github.com/nytimes/covid-19-data
- Data on county economic factors from https://www.ers.usda.gov/data-products/county-level-data-sets/download-data/
- Data on county coordinates and population from https://simplemaps.com/data/us-counties

All four data sources were obtained as CSV files



Data Exploration: Cases/Deaths

Across the four datasets, I included a ST column for consistency. Here, doing so for the cases/deaths data.

```
## Replacing the "state" column with a "ST" column containing the state abbreviation
# 1 extract the old column
Latest_Deaths_df["state"]
```

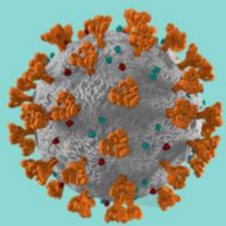
```
94427    Alabama
94428    Alabama
94429    Alabama
```

```
# 2 create the mapping series
# 3 Use series constructor
States_s = pd.Series(
    Lat_Long_df["state_id"].values, index=Lat_Long_df["state_name"]).drop_duplicates(
States_s
```

```
state_name
California      CA
Illinois        IL
```

```
# 4 adjust the code to add the new column to the DataFrame
# 5 Delete the old column from the dataframe
Latest_Deaths_df["ST"] = Latest_Deaths_df["state"].map(States_s)
Latest_Deaths_df.drop(columns="state", inplace=True)
Latest_Deaths_df
```

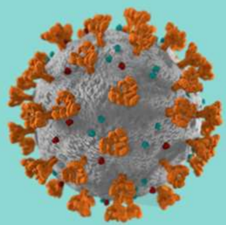
	date	county	fips	cases	deaths	ST
94427	2022-09-14	Autauga	1001.0	18233	226.0	AL
94428	2022-09-14	Baldwin	1003.0	65088	702.0	AL



Data Exploration: Vaccination Rate

I chose to only use data for the 50 states (territories had incomplete data). Here, dropping rows for Puerto Rico and Guam.

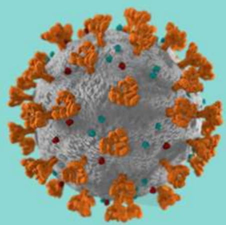
```
# Dropping counties that are not in Lat Long
Latest_Vax_df = Latest_Vax_df[Latest_Vax_df.ST != "PR"]
Latest_Vax_df = Latest_Vax_df[Latest_Vax_df.ST != "GU"]
Latest_Vax_df.drop([92], inplace = True)
Latest_Vax_df
```



Data Exploration: Latitude/Longitude

I ultimately joined the tables under the "fips" column, so here I first renamed the column across the datasets to be consistent.

```
### WORKING ON LAT/LONG dataframe
# Renaming "state_id" to "ST" and "county_fips" to "fips" to be consistent with other datasets
Lat_Long_df = Lat_Long_df.rename(columns={"state_id":"ST", "county_fips":"fips"})
Lat_Long_df
```



Data Exploration: Economic Indicators

Commas had to be dropped from median income data in the economics data.

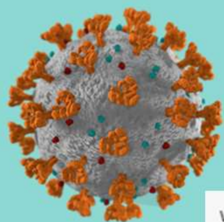
```python
# Converting median income to integer part 1
Econ_df.replace(",","", regex=True, inplace=True)
Econ_df
```

	fips	ST	Unemployment_rate_2021	Median_Household_Income_2020
0	0	US	5.4	67340
1	1000	AL	3.4	53958
2	1001	AL	2.8	67565
3	1003	AL	3.0	71135
4	1005	AL	5.7	38866

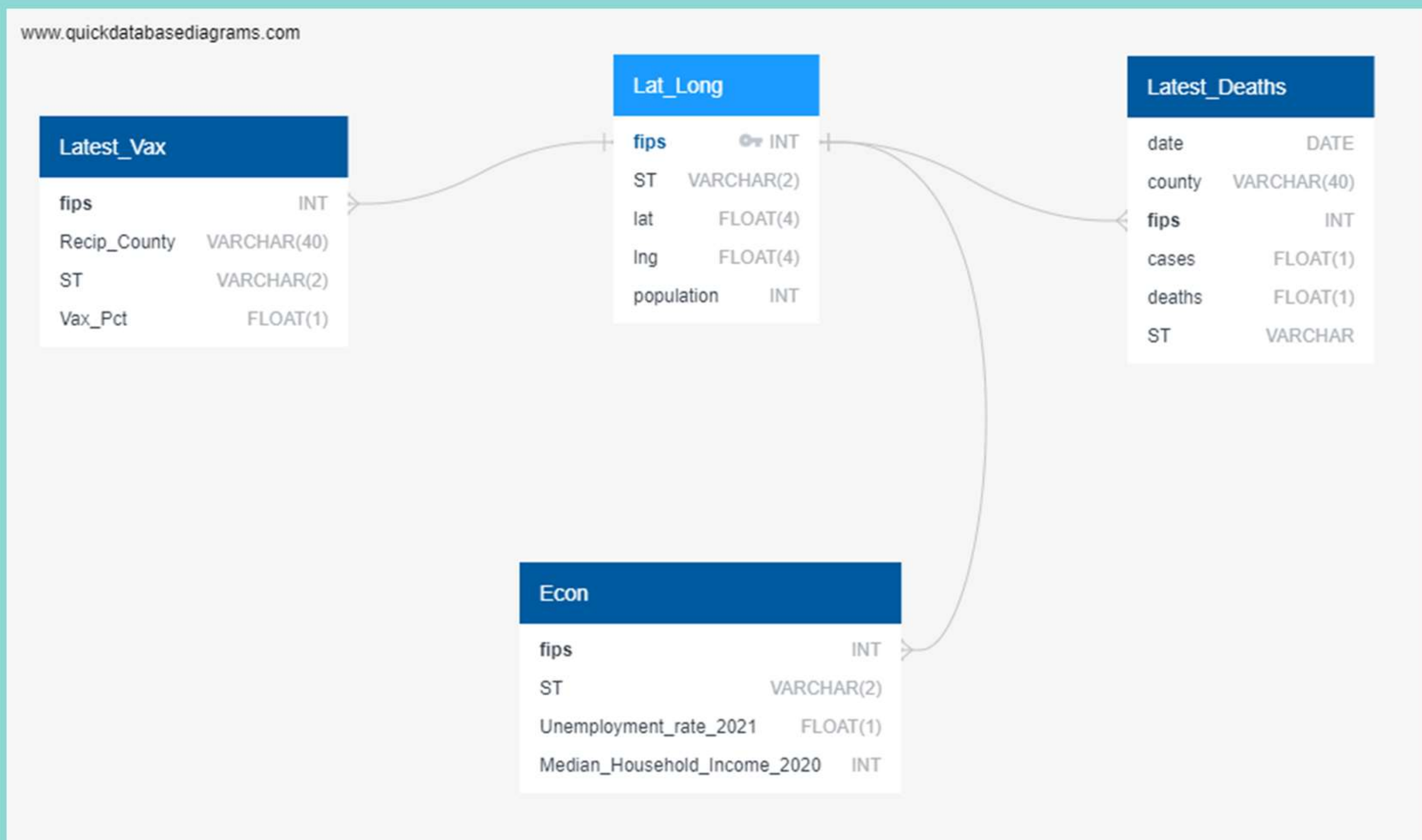
```python
# Converting datatype of median income into integer, part 2
Econ_df['Median_Household_Income_2020'] = Econ_df['Median_Household_Income_2020'].astype(str).astype(int)
Econ_df.dtypes
```

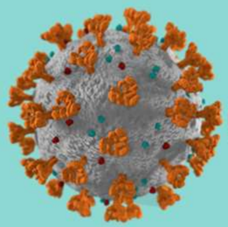
```
fips                              int64
ST                               object
Unemployment_rate_2021          float64
Median_Household_Income_2020      int32
dtype: object
```

Database



Entity Relationship Diagram (ERD)

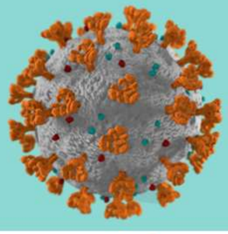




Creating Tables in SQL

```
-- Creating tables for COVID-project
CREATE TABLE Lat_Long (
    fips INT NOT NULL,
    ST VARCHAR(2) NOT NULL,
    lat FLOAT(4) NOT NULL,
    lng FLOAT(4) NOT NULL,
    population INT NOT NULL,
    PRIMARY KEY (fips)
);

CREATE TABLE Latest_Vax (
    fips INT NOT NULL,
    Recip_County VARCHAR(40) NOT NULL,
    ST VARCHAR(2) NOT NULL,
    Vax_Pct FLOAT(1) NOT NULL,
    FOREIGN KEY (fips) REFERENCES Lat_Long (fips)
);
```

```
CREATE TABLE Latest_Deaths (
    date DATE NOT NULL,
    county VARCHAR(40) NOT NULL,
    fips INT NOT NULL,
    cases FLOAT(1) NOT NULL,
    deaths FLOAT(1) NOT NULL,
    ST VARCHAR(2) NOT NULL,
    FOREIGN KEY (fips) REFERENCES Lat_Long (fips)
);

CREATE TABLE Econ (
    fips INT NOT NULL,
    ST VARCHAR(2) NOT NULL,
    Unemployment_rate_2021 FLOAT(1) NOT NULL,
    Median_Household_Income_2020 INT NOT NULL,
    FOREIGN KEY (fips) REFERENCES Lat_Long (fips)
);
```

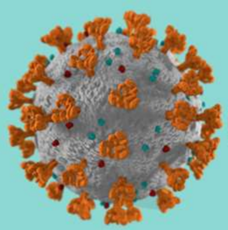


Merging Tables in SQL

```sql
-- Create combined table

-- Joining lat_long and econ
SELECT lat_long.fips,
     lat_long.ST,
     lat_long.lat,
     lat_long.lng,
     lat_long.population,
     econ.Unemployment_rate_2021,
     econ.Median_Household_Income_2020
INTO lat_long_econ
FROM lat_long
LEFT JOIN econ
ON lat_long.fips = econ.fips;
```

```sql
-- Joining the above table with latest_deaths
SELECT lat_long_econ.fips,
     lat_long_econ.ST,
     lat_long_econ.lat,
     lat_long_econ.lng,
     lat_long_econ.population,
     lat_long_econ.Unemployment_rate_2021,
     lat_long_econ.Median_Household_Income_2020,
     latest_deaths.county,
     latest_deaths.cases,
     latest_deaths.deaths
INTO lat_long_econ_deaths
FROM lat_long_econ
LEFT JOIN latest_deaths
ON lat_long_econ.fips = latest_deaths.fips;
```

```sql
-- Combining above table with vax
SELECT lat_long_econ_deaths.fips,
     lat_long_econ_deaths.ST,
     lat_long_econ_deaths.lat,
     lat_long_econ_deaths.lng,
     lat_long_econ_deaths.population,
     lat_long_econ_deaths.Unemployment_rate_2021,
     lat_long_econ_deaths.Median_Household_Income_2020,
     lat_long_econ_deaths.county,
     lat_long_econ_deaths.cases,
     lat_long_econ_deaths.deaths,
     latest_vax.vax_pct
INTO all_tables_merged
FROM lat_long_econ_deaths
LEFT JOIN latest_vax
ON lat_long_econ_deaths.fips = latest_vax.fips;
```

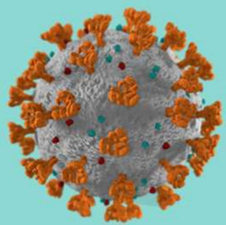


Creating Cases and Deaths per 100,000

To create a rate similar to vaccination rate, I used the population data to calculate cases and deaths per 100,000 people in the counties.

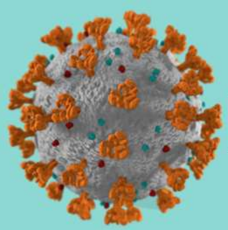
```
-- Creating cases and deaths per 100,000
ALTER TABLE all_tables_merged
    ADD cases_100000 FLOAT(2);

ALTER TABLE all_tables_merged
    ADD deaths_100000 FLOAT(2);

UPDATE all_tables_merged SET cases_100000 = (cases / population * 100000);
UPDATE all_tables_merged SET deaths_100000 = (deaths / population * 100000);
```

Data Analysis



Descriptive Statistics

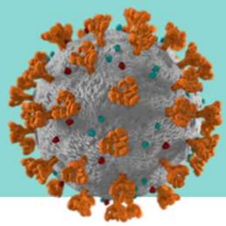
	Unemployment	Median Income	Vaccination Percentage	Cases per 100,000	Deaths per 100,000
Mean	4.64	$57,364.90	52.15%	28,296	395
SD	1.74	$14,545.63	12.43%	7,711	164
Median	4.4	$55,044.00	59.43%	28,025	390



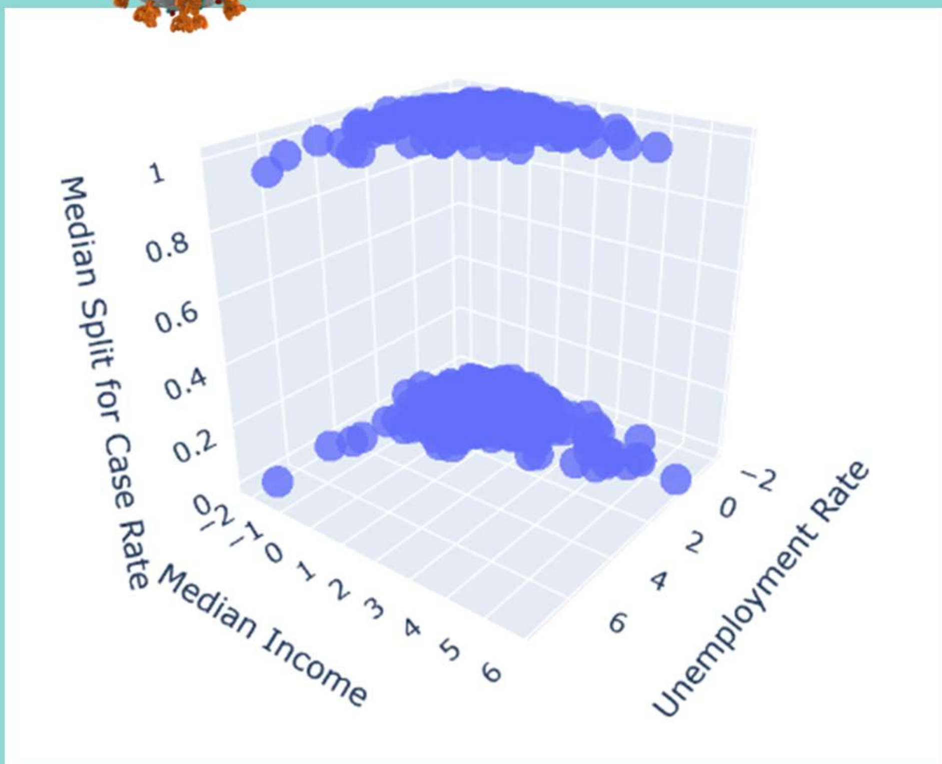
Machine Learning: Logistic Regression

In order to perform logistic regressions, a median split was created for the following three variables: cases per 100,000 people, deaths per 100,000 people, and vaccination rate.

```
# Creating median split codes for cases, deaths, vax
df["median_split_cases"] = (df.cases_100000<df.cases_100000.quantile()).replace({True:0, False:1})
df["median_split_deaths"] = (df.deaths_100000<df.deaths_100000.quantile()).replace({True:0, False:1})
df["median_split_vax_pct"] = (df.vax_pct<df.vax_pct.quantile()).replace({True:0, False:1})
```



Logistic Regression: Case Rate




```
# Get accuracy score

from sklearn.metrics import accuracy_score
print("Accuracy score predicting case rate")
print(accuracy_score(y_cases_test, y_cases_pred))
```

```
Accuracy score predicting case rate
0.5558408215661104
```

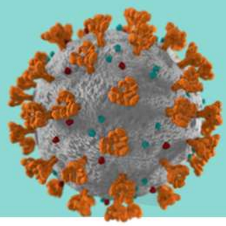
```
# Get confusion matrix
from sklearn.metrics import confusion_matrix, classification_report
cases_matrix = confusion_matrix(y_cases_test, y_cases_pred)

# Create a DataFrame from the confusion matrix.
matrix_cases_df = pd.DataFrame(cases_matrix,
                               index=["Actual 0-low_cases", "Actual 1-high_cases"],
                               columns=["Predicted 0-low_cases", "Predicted 1-high_cases"])

matrix_cases_df
```

	Predicted 0-low_cases	Predicted 1-high_cases
Actual 0-low_cases	211	172
Actual 1-high_cases	174	222

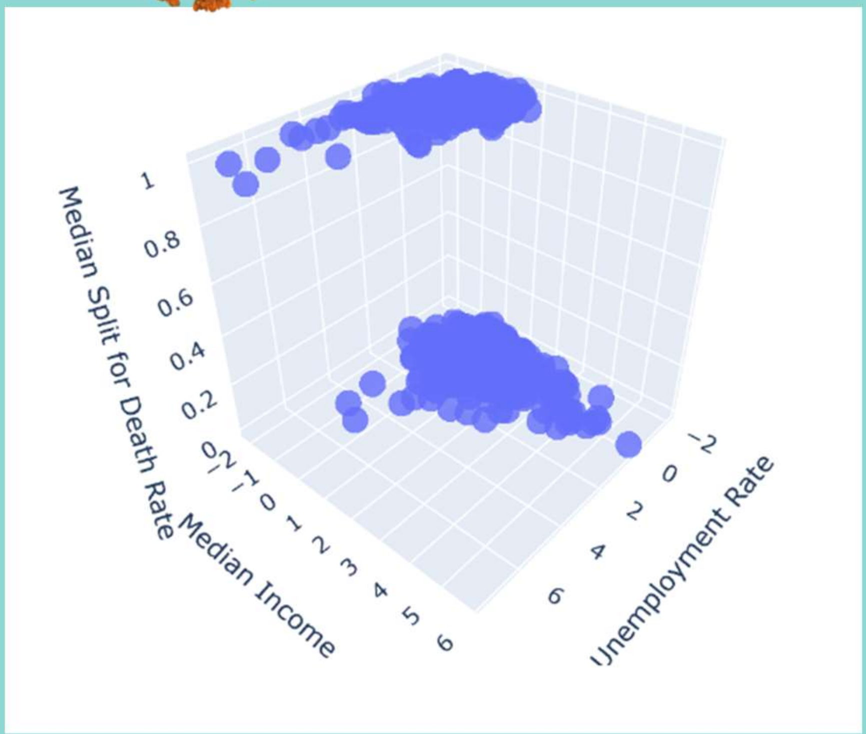
```
# Classification report
cases_report = classification_report(y_cases_test, y_cases_pred)
print(cases_report)
```

```
              precision    recall  f1-score   support

           0       0.55      0.55      0.55       383
           1       0.56      0.56      0.56       396

    accuracy                           0.56       779
   macro avg       0.56      0.56      0.56       779
weighted avg       0.56      0.56      0.56       779
```

The model hardly performed better than chance when predicting case rate.



Logistic Regression: Death Rate




```python
# Get accuracy score

print("Accuracy score predicting death rate")
print(accuracy_score(y_deaths_test, y_deaths_pred))
```

```
Accuracy score predicting death rate
0.7329910141206675
```

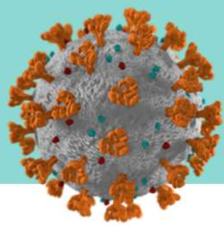
```python
# Get confusion matrix

deaths_matrix = confusion_matrix(y_deaths_test, y_deaths_pred)

# Create a DataFrame from the confusion matrix.
matrix_deaths_df = pd.DataFrame(deaths_matrix,
                           index=["Actual 0-low_deaths", "Actual 1-high_deaths"],
                           columns=["Predicted 0-low_deaths", "Predicted 1-high_deaths"])

matrix_deaths_df
```

	Predicted 0-low_deaths	Predicted 1-high_deaths
Actual 0-low_deaths	270	105
Actual 1-high_deaths	103	301

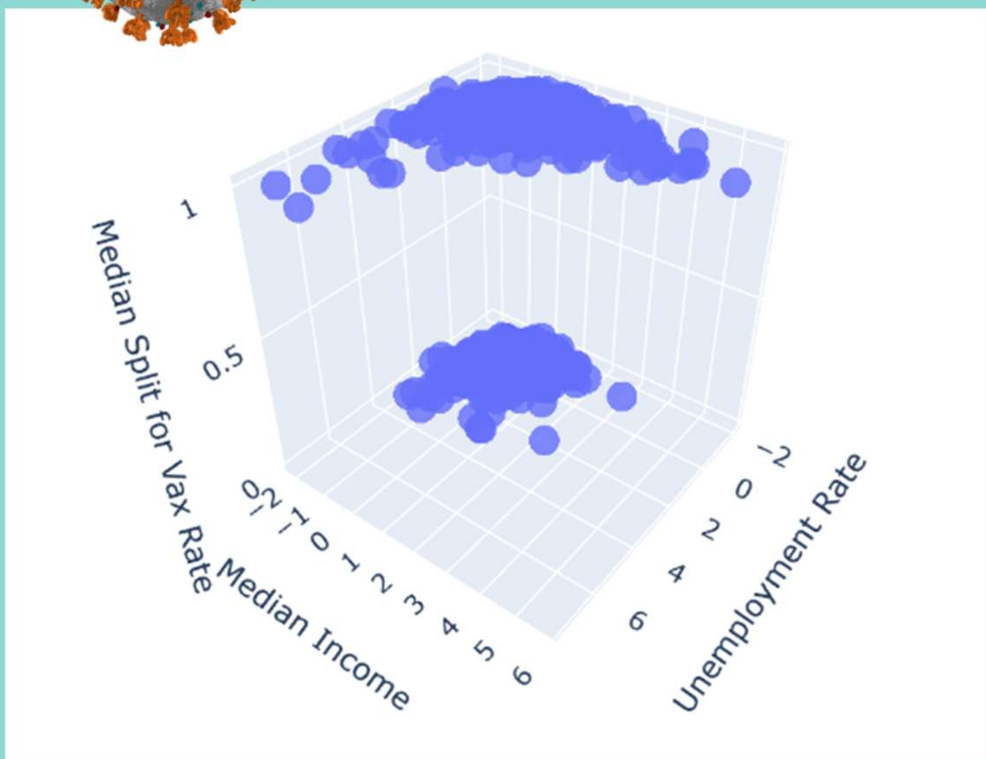
```python
# Classification report
deaths_report = classification_report(y_deaths_test, y_deaths_pred)
print(deaths_report)
```

```
              precision    recall  f1-score   support

           0       0.72      0.72      0.72       375
           1       0.74      0.75      0.74       404

    accuracy                           0.73       779
   macro avg       0.73      0.73      0.73       779
weighted avg       0.73      0.73      0.73       779
```

The model was much more successful at predicting death rate.



Logistic Regression: Vaccination Rate




```python
# Get accuracy score

print("Accuracy score predicting vax pct")
print(accuracy_score(y_vax_test, y_vax_pred))
```

```
Accuracy score predicting vax pct
0.6469833119383825
```

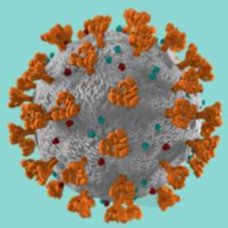
```python
# Get confusion matrix

vax_matrix = confusion_matrix(y_vax_test, y_vax_pred)

# Create a DataFrame from the confusion matrix.
matrix_vax_df = pd.DataFrame(vax_matrix,
                             index=["Actual 0-low_vax", "Actual 1-high_vax"],
                             columns=["Predicted 0-low_vax", "Predicted 1-high_vax"])

matrix_vax_df
```

	Predicted 0-low_vax	Predicted 1-high_vax
Actual 0-low_vax	271	117
Actual 1-high_vax	158	233

```python
# Classification report
vax_report = classification_report(y_vax_test, y_vax_pred)
print(vax_report)
```

```
              precision    recall  f1-score   support

           0       0.63      0.70      0.66       388
           1       0.67      0.60      0.63       391

    accuracy                           0.65       779
   macro avg       0.65      0.65      0.65       779
weighted avg       0.65      0.65      0.65       779
```

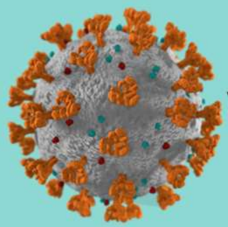
And it was somewhere in the middle when predicting vaccination rate.



Comparison to Support Vector Machines

Outcome Variable	Accuracy score LG	Accuracy score SVM
Cases per 100,000	55.6%	55.3%
Deaths per 100,000	73.3%	72.9%
Vaccination Rate	64.7%	65.0%

Next, SVM models were created in the same manner of the logistic regressions. These models performed nearly identically to the regressions.



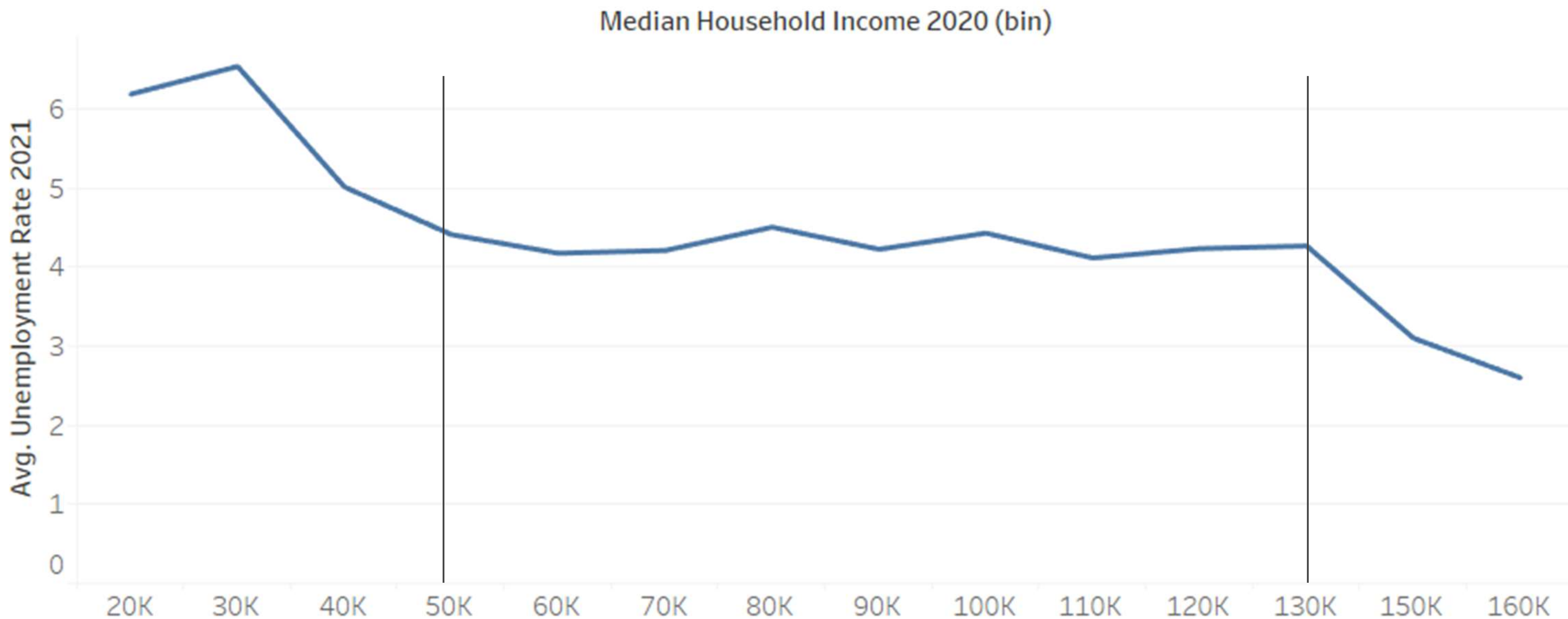
Visualizations with Tableau

Variable	Bin size
Vaccination Rate	15%
Median Income	$10,000
Unemployment	2.5%

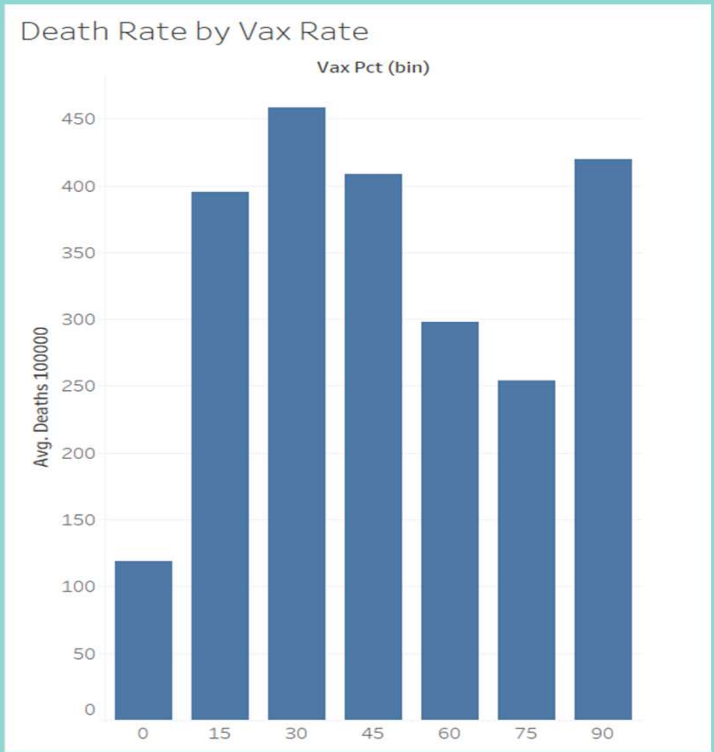
In order to make certain visualizations, bins were created for the continuous variables of vaccination rate, median income, and unemployment.

Visualizations can also be seen at https://public.tableau.com/app/profile/joe.rodini/viz/COVID-projectvisualizations/COVID-project#1

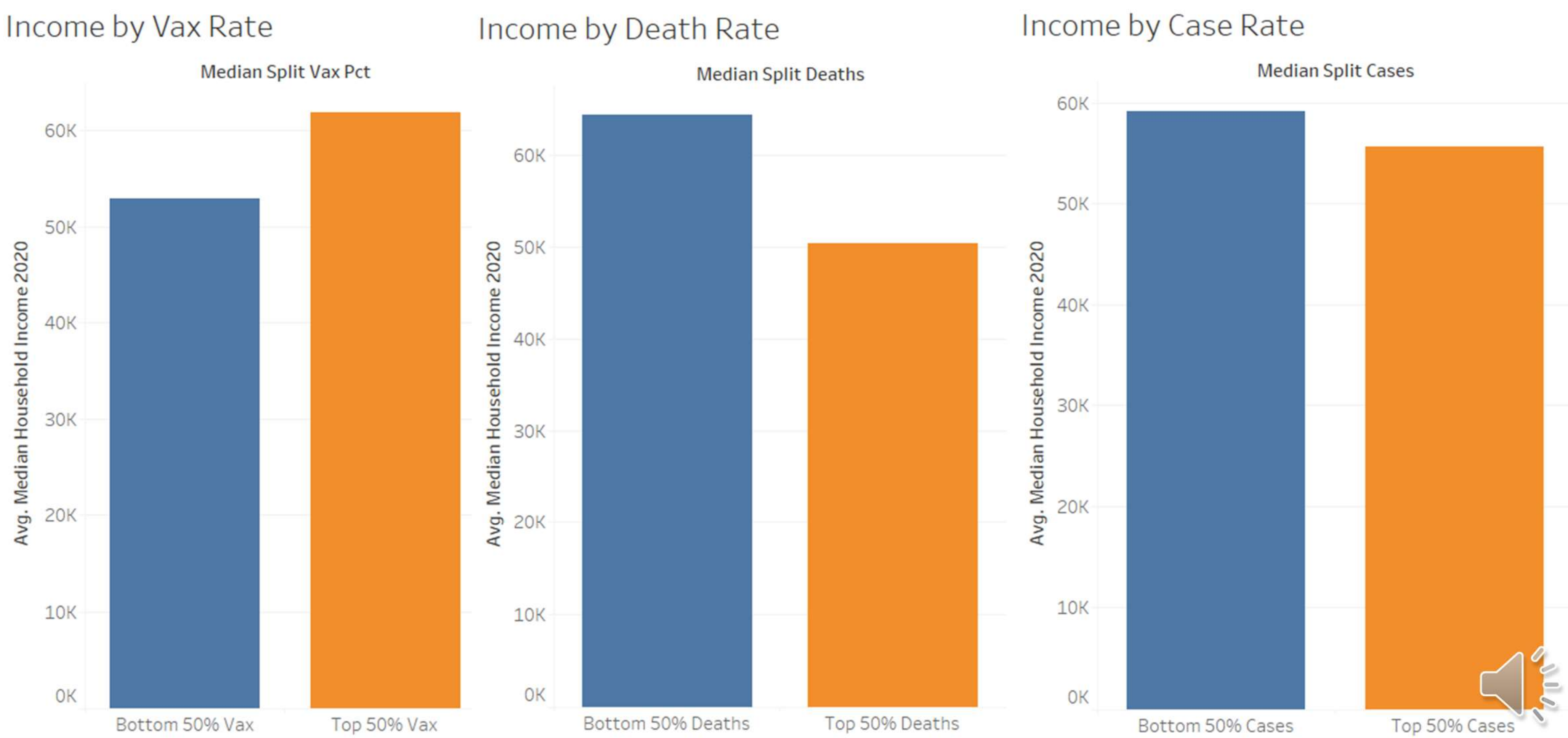
Income and Unemployment



Economic Predictors of COVID Outcomes

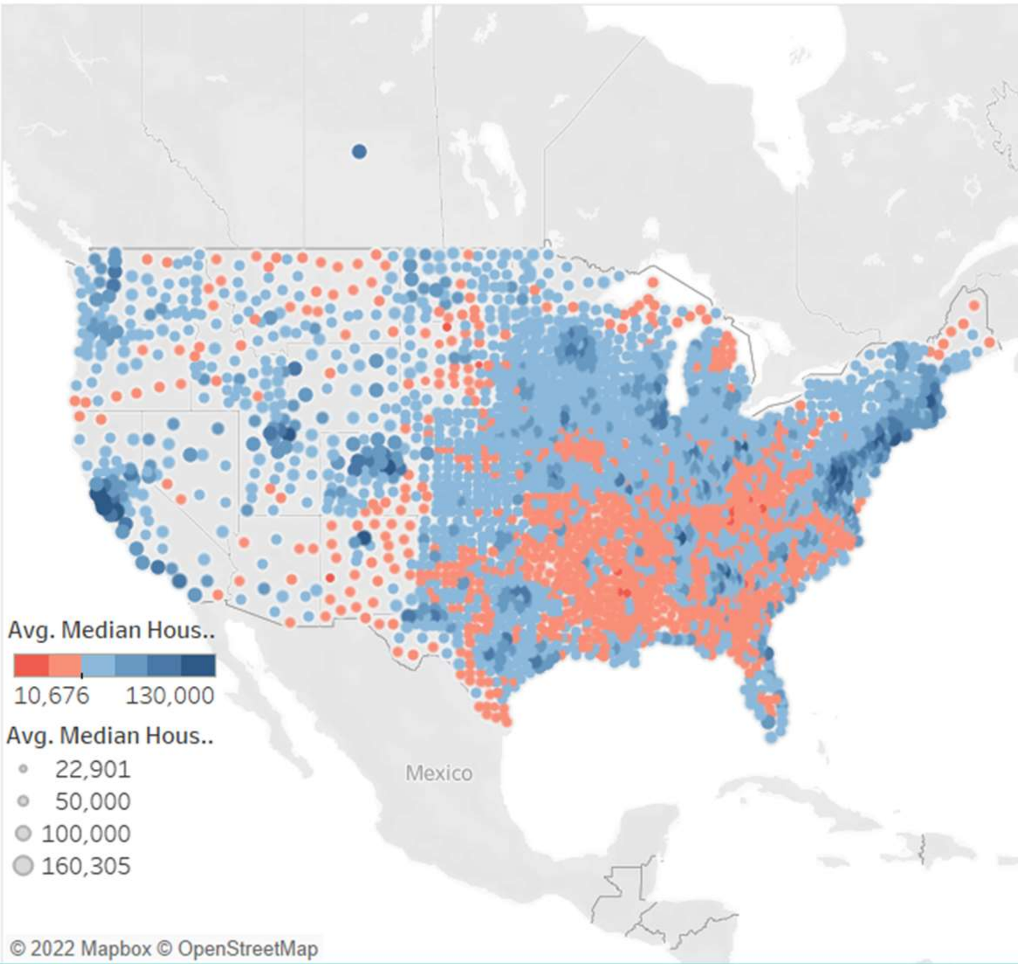


Income and COVID outcomes

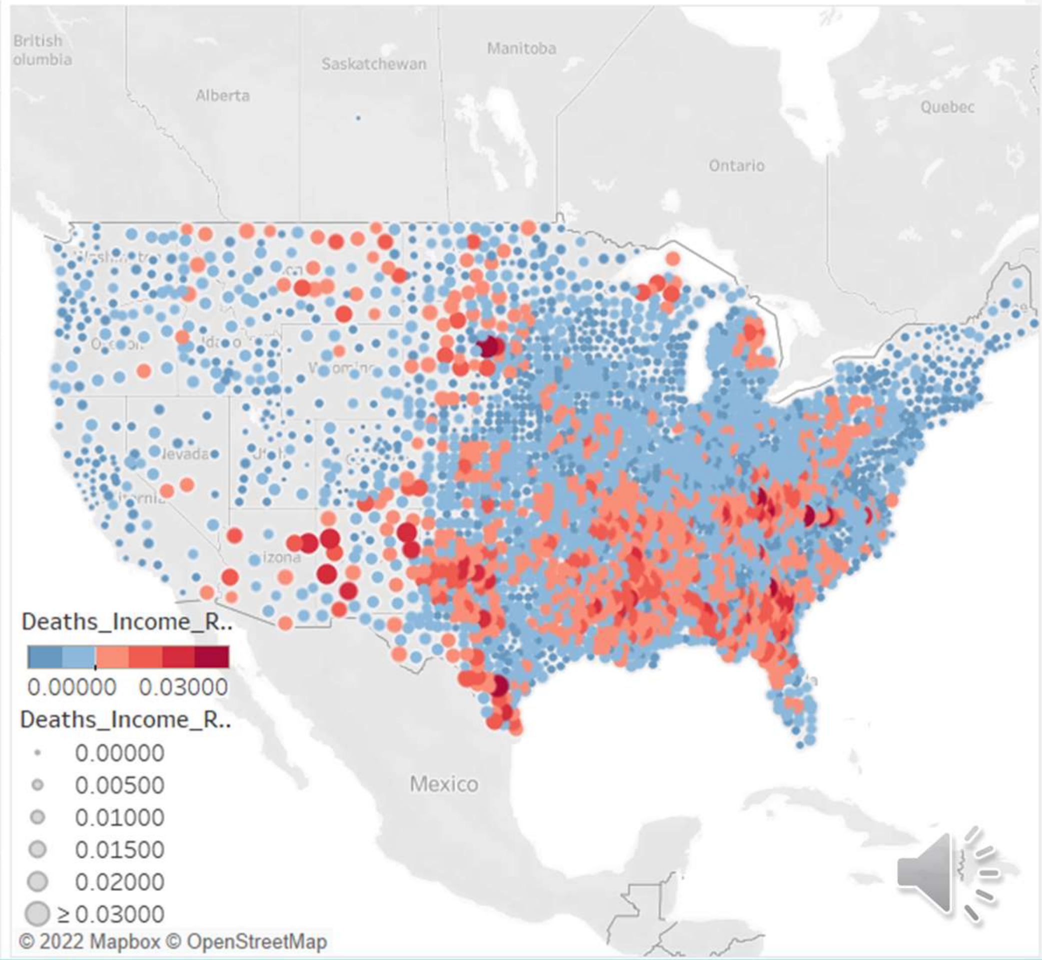


National Overview

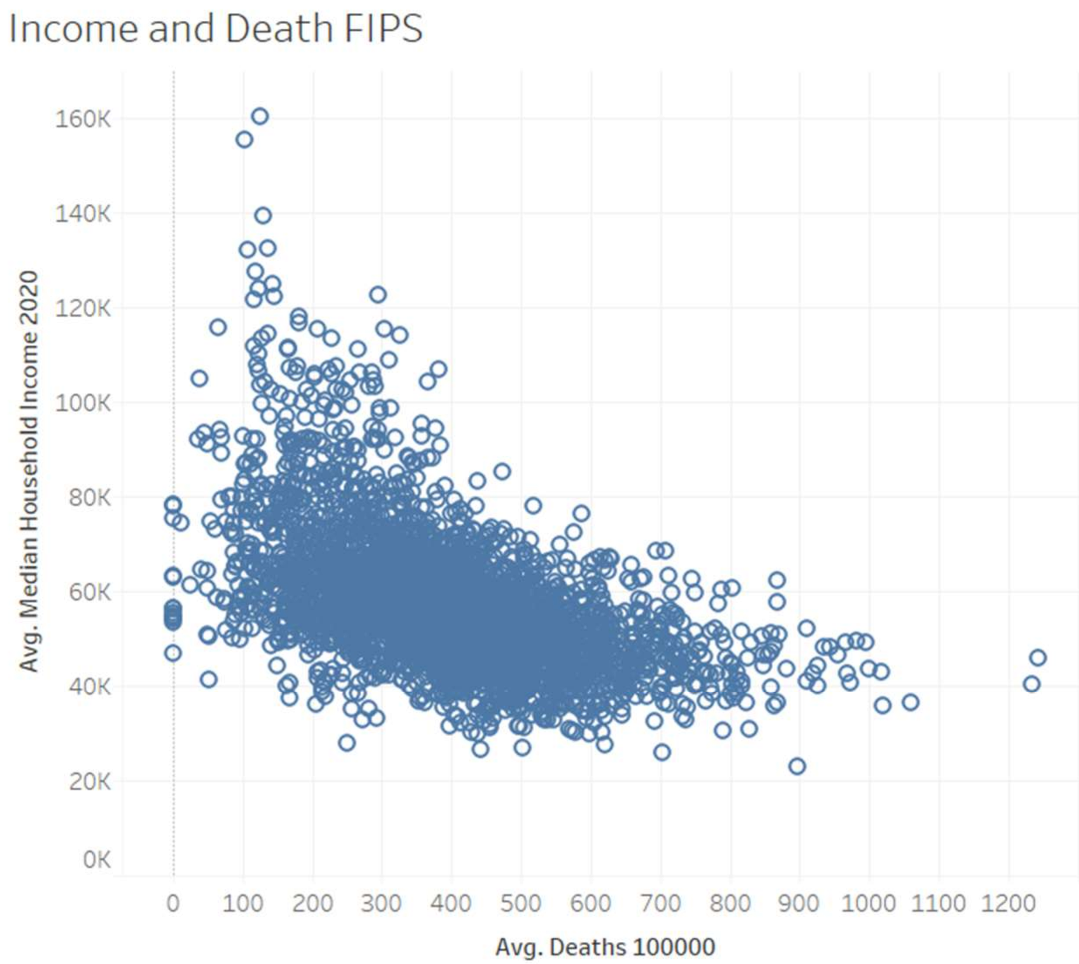
Median Household Income

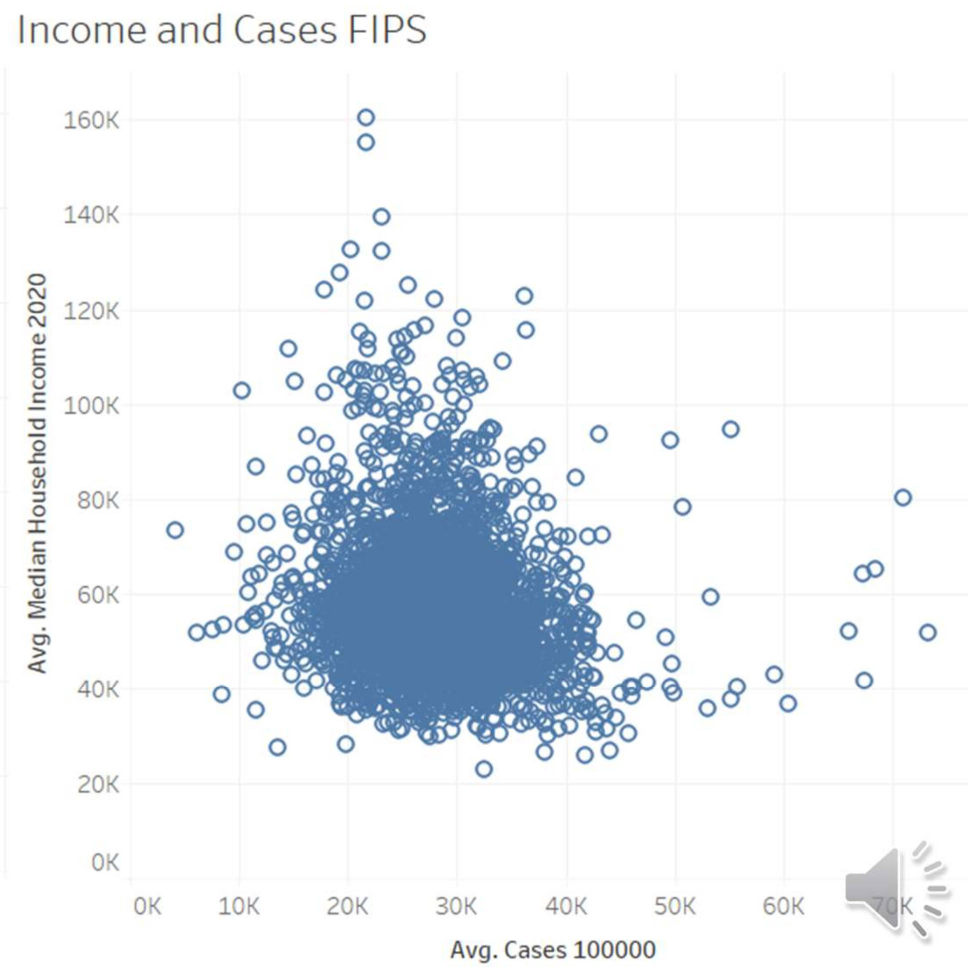


Deaths to Income Ratio

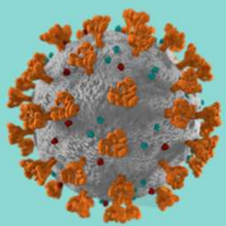


Income Scatterplots



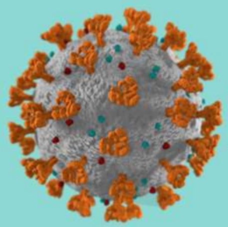


Conclusions



Economic Predictors and COVID Outcomes

- Case rate was not well-predicted by economic indicators, meaning that the spread of COVID was fairly uniform across the country regardless of economic level.
- However, death rate was well predicted by economic indicators, demonstrating that counties with more economic resources were better able to mitigate the pernicious effects of the pandemic.
- Vaccination rate was somewhat well predicted by economic indicators, suggesting that counties with more economic resources did somewhat of a better job getting their populations vaccinated.



Next Steps

- Additional analysis would continue to shed light on this topic.
- Factors that might have obscured the relationship between economic predictors and COVID outcomes might include: population density, political affiliation, education level, and ethnicity
- Correlation is not necessarily causation–it could be that other variables, such as the ones above, cause the economic predictors and COVID outcomes to show a relationship

Thank You!