Comparing Influenza Trends in New York State with Interactive Visualizations

Katherine Huerta

Deliverable 3

Data 606 SP 2020

Methods Overview

1. Influenza Season Subsets
   a. Visualizing the sum of cases for each season at the county-level
2. Interactive Map – Case Total
   a. Outcomes/Questions – how much does population affect this picture?
3. Add population data to the subsets
4. Calculate the prevalence rates per 10,000 people
   a. Add as a new column in subsets
5. Interactive Map – Prevalence Rates

Methods for Seasonal Subsets

1. Sorted data frame by “Season” (2009-2010, 2010-2011, … , 2019-2020)

```
# sort by season
df = df.sort_values(by = ['Season'])
df.head()
df.tail()
```

Season
2019-2020
2019-2020
2019-2020
2019-2020
2019-2020

Season
2009-2010
2009-2010
2009-2010
2009-2010
2009-2010

Methods for Seasonal Subsets

2. For each season:

```
# 2017-2018 Season:
df_17 = df.loc[(df["Season"]=="2017-2018")]
df_17 = df_17.sort_values(by = ['Region','County','CDC_Week'])
df_17 = df_17.reset_index(drop = True)
by_county_17 = df_17.groupby("County").sum().reset_index()
by_county_17 = by_county_17.drop(['FIPS', 'CDC_Week'], axis=1)
by_county_17['FIPS'] = FIPS
by_county_17.head()
```

	County	Count	FIPS
0	ALBANY	1708	36001.0
1	ALLEGANY	205	36003.0
2	BRONX	11749	36005.0
3	BROOME	2214	36007.0
4	CATTARAUGUS	492	36009.0

Season Sub

Group By County (alphabetical) and make it so each county has one row containing the sums for each column.

Reset Index

Drop erroneous FIPS column, and drop unnecessary CDC Week column

ıe correct column

Methods for Seasonal Subsets

3. For each season: `by_county_17.describe()`

	Count
count	62.000000
mean	2068.500000
std	3093.307269
min	28.000000
25%	392.500000
50%	710.000000
75%	1900.750000
max	13511.000000

62 rows = 62 unique counties

Average number of cases per county in 2017-2018 season

The county with the highest number of lab-confirmed cases had a total of 13,511 cases in the 2017-2018 season.

Methods – Interactive Map

- Tools used (Python/Jupyter Notebook):
  - Plotly Express
    - Mapbox Choropleth Maps
    - Urlopen and json to load a GeoJSON file to make the map foundation

```
# Import packages

import pandas as pd
import numpy as np

import json

import plotly.express as px

from urllib.request import urlopen
```

```
with urlopen('https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-fips.json') as response:
    counties = json.load(response)
```

- Two main input types:
  - GeoJSON-formatted geometry information (has all county information in the US)
  - List of values indexed by feature identifier

Methods – Making the Map

```
fig_17 = px.choropleth_mapbox(by_county_17,
                              geojson=counties,
                              locations='FIPS',
                              color='Count',
                              hover_name = "County",
                              color_continuous_scale="Blues",
                              range_color=(0, 8000),
                              mapbox_style="carto-positron",
                              zoom=5.5, center = {"lat": 43.2994, "lon": -74.2179},
                              opacity=0.9,
                              labels={'Count':'Total Count'}
                             )
fig_17.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig_17.show()
```

https://plotly.com/python/mapbox-county-choropleth/

Methods – Making the Map

```
# 2017-2018 Season:
df_17 = df.loc[(df["Season"]=="2017-2018")]
df_17 = df_17.sort_values(by = ['Region','County','CDC_Week'])
df_17 = df_17.reset_index(drop = True)
by_county_17 = df_17.groupby("County").sum().reset_index()
by_county_17 = by_county_17.drop(['FIPS', 'CDC_Week'], axis=1)
by_county_17['FIPS'] = FIPS
```

County Subset

```
fig_17 = px.choropleth_mapbox(by_county_17,
                              geojson=counties,
                              locations='FIPS',
                              color='Count',
                              hover_name = "County",
                              color_continuous_scale="Blues",
                              range_color=(0, 8000),
                              mapbox_style="carto-positron",
                              zoom=5.5, center = {"lat": 43.2994, "lon": -74.2179},
                              opacity=0.9,
                              labels={'Count':'Total Count'}
                             )
fig_17.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig_17.show()
```

Methods – Making the Map

```
fig_17 = px.choropleth_mapbox(by_county_17,
                              geojson=counties,
                              locations='FIPS',
                              color='Count',
                              hover_name = "County",
                              color_continuous_scale="Blues",
                              range_color=(0, 8000),
                              mapbox_style="carto-positron",
                              zoom=5.5, center = {"lat": 43.2994, "lon": -74.2179},
                              opacity=0.9,
                              labels={'Count':'Total Count'}
                             )
fig_17.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig_17.show()
```

2017-2018 Influenza Season Subset

County-level GeoJSON map

Locations identified by their FIPS code

Color by count (darker = higher count)

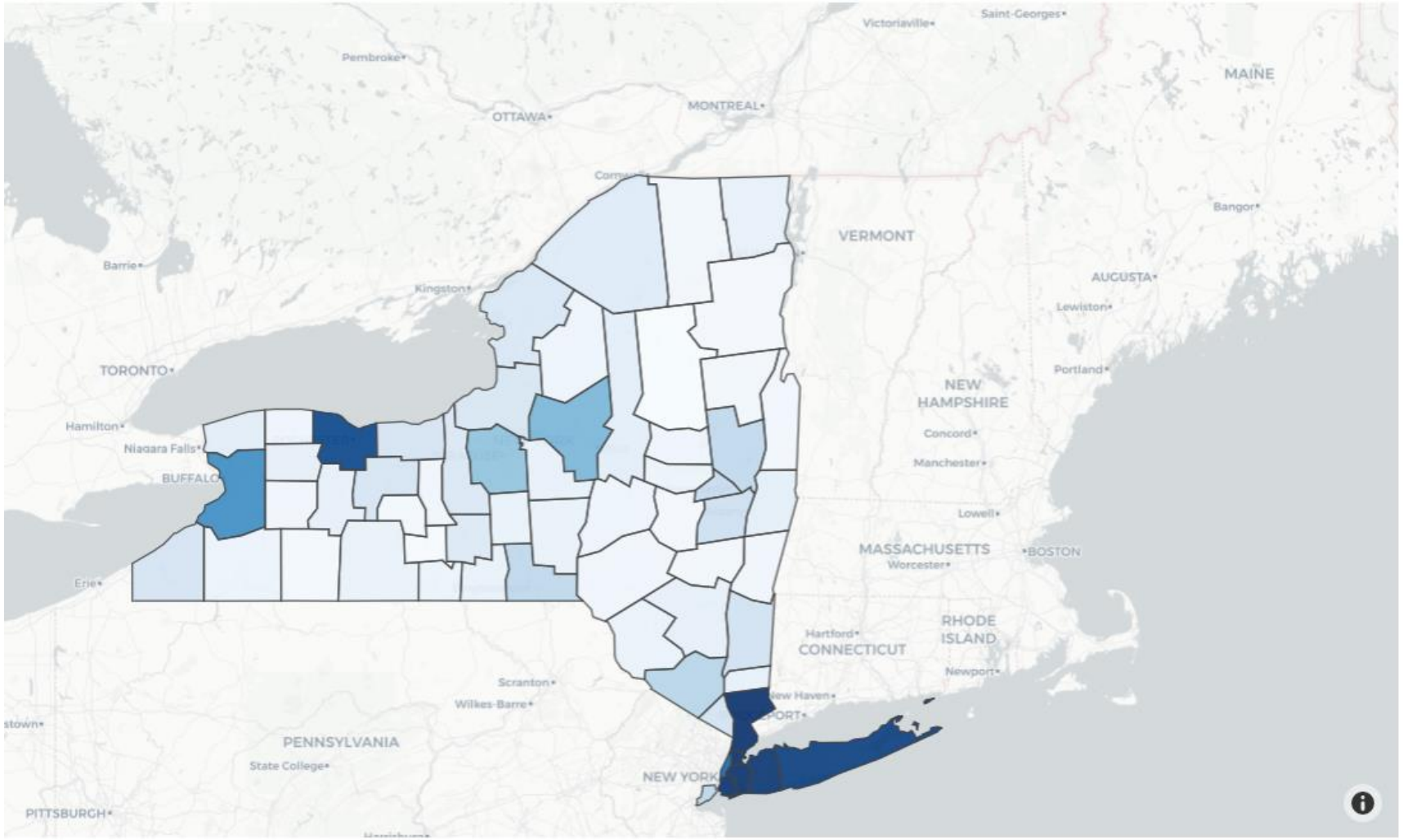
NYS Coordinates

Methods – Making the Map

```
fig_17 = px.choropleth_mapbox(by_county_17,
                              geojson=counties,
                              locations='FIPS',
                              color='Count',
                              hover_name = "County",
                              color_continuous_scale="Blues",
                              range_color=(0, 8000),
                              mapbox_style="carto-positron",
                              zoom=5.5, center = {"lat": 43.2994, "lon": -74.2179},
                              opacity=0.9,
                              labels={'Count':'Total Count'}
                             )
fig_17.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig_17.show()
```

- 2017-2018 Influenza Season Subset
- County-level GeoJSON map
- Locations identified by their FIPS code
- Color by count (darker = higher count)
- Add county name in hover over information
- Color scale light blue → dark blue
- 0 Counts = White, counts > 8,000 = dark blue
- Identifier of base map style
- NYS Coordinates
- Mostly solid (less transparency)
- Labels in legend specified to Total Count
- Setting margin layout (right, top, left, bottom)
- Show figure

2017-2018 Influenza Season Density Map



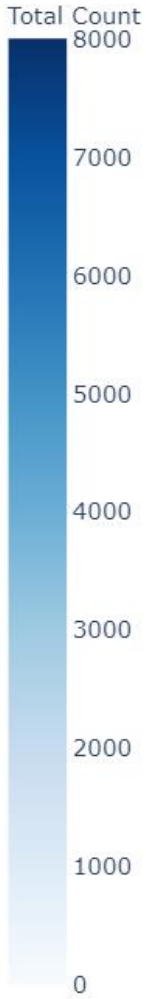


Figure 1. Density map of the sum of all confirmed influenza cases from the 2017-2018 influenza season. The legend on the right indicates that the darker the color, the higher the total count will be (8,000+ = darkest). However, four of the counties surpassed this count (which can be seen in fig. 3).

Model Details

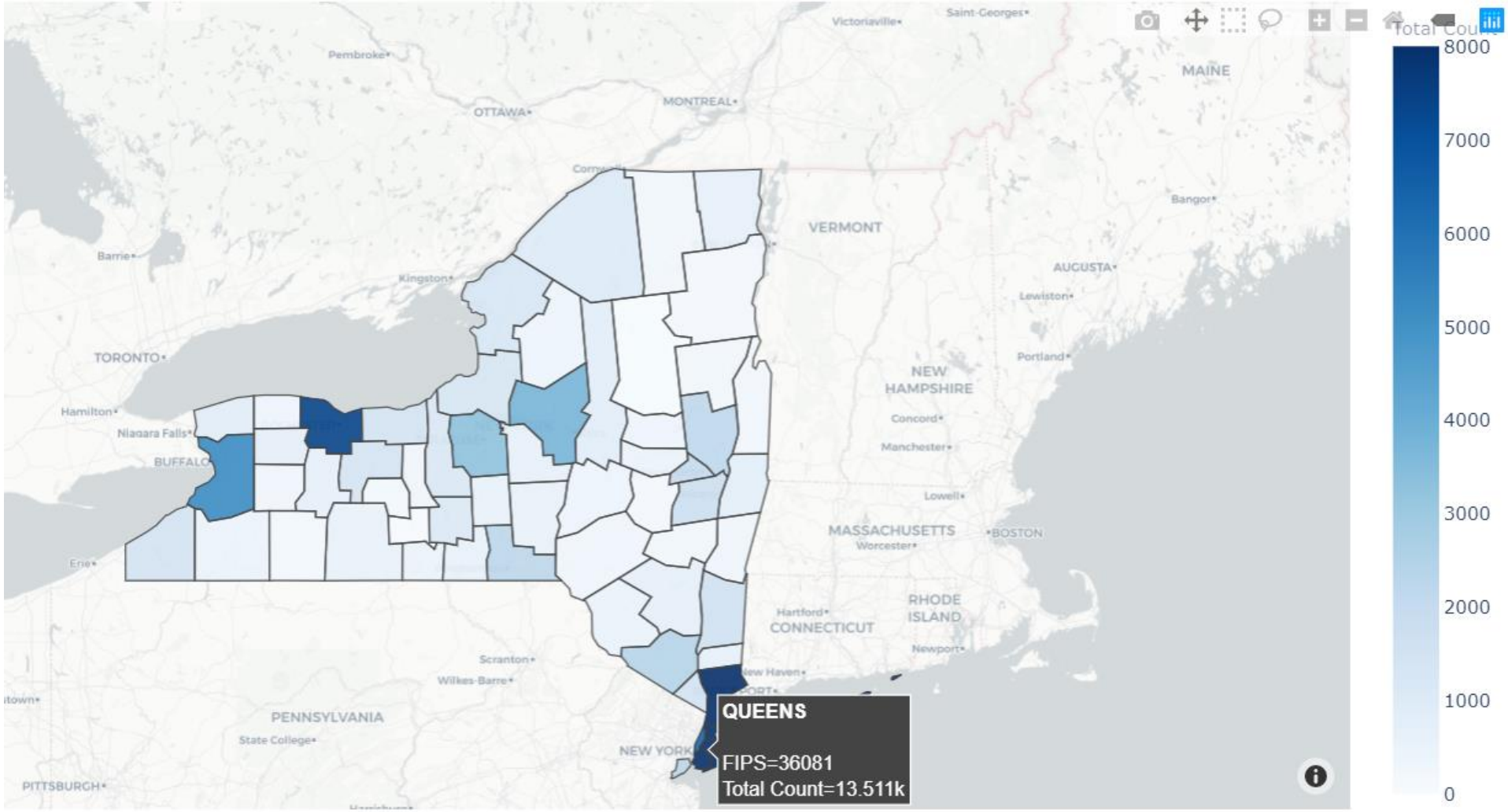


Figure 2. Density map of the sum of all confirmed influenza cases from the 2017-2018 influenza season. This figure shows one of the interactive features of this map (the hover over information). Users can also zoom in, out, and drag the cursor to navigate the map.

2017-2018 Influenza Season Density Map

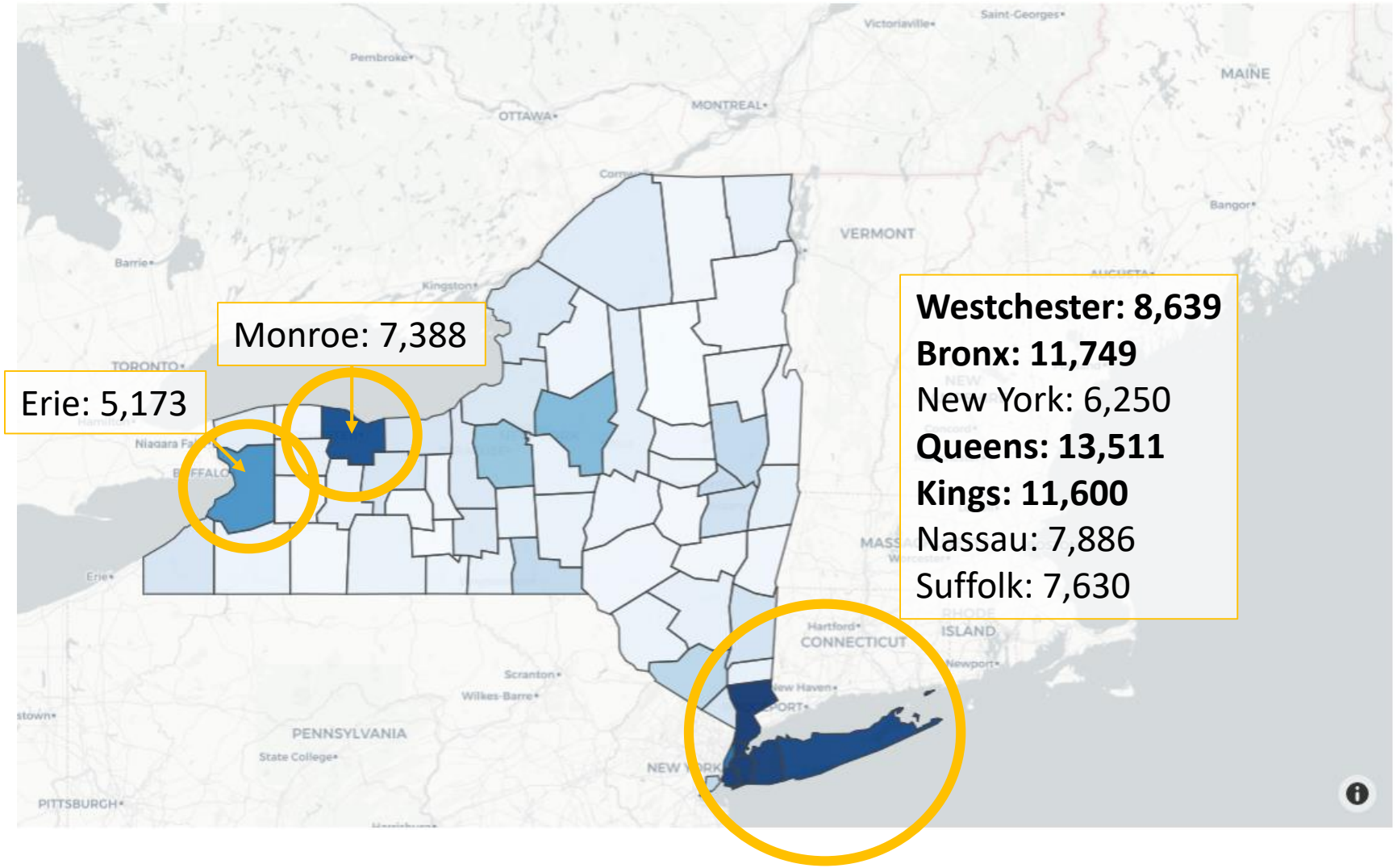


Figure 3. Density map of the sum of all confirmed influenza cases from the 2017-2018 influenza season. This figure outlines some of the counties that had the highest sum of confirmed influenza cases (as well as their number of confirmed cases).

Adding Population Data to Subsets

- Data Source: https://data.ny.gov/Government-Finance/Annual-Population-Estimates-for-New-York-State-and/krt9-ym2k

1. Read in data (pandas) and prepare it (rename columns, check for null/missing data, etc.)

```
cols = ["FIPS Code", "Geography", "Year", "Population"]
pop = pd.read_csv(path+"pop_estimates.csv", usecols=cols)
pop=pop.rename(columns = {'Geography': 'County'})
pop.head()
```

Adding Population Data to Subsets

2. Make two population subsets:
    a. One for the 2009-2010 season (for the decade of 2000)
    b. One for the remaining seasons(2010-2019)
        a. Used 2011 data arbitrarily (population estimates are relatively consistent)

```
pop_county_11 = pop.loc[(pop["Year"]==2011)]
pop_county_11 = pop_county_11.sort_values(by = ['County'])
pop_county_11 = pop_county_11.reset_index(drop = True)
pop_county_11.head()
```

Adding Population Data to Subsets

2. Make two population subsets:
   a. One for the 2009-2010 season (for the decade of 2000)
   b. One for the remaining seasons(2010-2019)
      a. Used 2011 data arbitrarily (population estimates are relatively consistent)

	FIPS Code	County	Year	Population
0	36001	Albany County	2011	304596
1	36003	Allegany County	2011	48800
2	36005	Bronx County	2011	1397335
3	36007	Broome County	2011	199363
4	36009	Cattaraugus County	2011	79815

Adding Population Data to Subsets

3. Remove the extra “New York State” row from the population dataset

```
# Drop the row with NYS
pop_county_11 = pop_county_11.drop(31, axis=0)
pop_county_11 = pop_county_11.reset_index(drop=True)
# Double check that NYS is what was removed

pop_county_11.County.unique()
```

```
'New York County', 'Niagara County', 'Oneida County',
```

```
# Check number of unique values after dropping 31st row
pop_county_11.nunique()
```

```
FIPS Code     62
County        62
Year           1
Population    62
dtype: int64
```

Adding Population Data to Subsets

4. Make the Population column to add to the subsets

```
population_11 = pop_county_11["Population"]
```

Adding Population Data to Subsets

5. Add population column to subsets

```
population = population_11
```

```
# 2017-2018
sum_county_17["Population"] = population
#sum_county_17.head()
#sum_county_17.describe()
```

Adding Population Data to Subsets

6. Define the function to calculate prevalence rate of confirmed influenza cases (10,000 people)

$$Prevalence\ Rate = \left(\frac{10{,}000 x Count}{Population}\right)$$

Adding Population Data to Subsets

```
def CalcPrevalenceNumerator(df):
    numerator=[]

    for index, row in df.iterrows():
        numerator.append(row.Count*10000)

    return numerator
```

```
def CalcPrevalenceRate(df):
    rate=[]

    for index, row in df.iterrows():
        rate.append(row.x/row.Population)

    return rate
```

$$Rate = \left(\frac{10{,}000xCount}{Population}\right)$$

	County	Count	FIPS	Population
0	ALBANY	1708	36001.0	304596
1	ALLEGANY	205	36003.0	48800
2	BRONX	11749	36005.0	1397335
3	BROOME	2214	36007.0	199363
4	CATTARAUGUS	492	36009.0	79815

Adding Population Data to Subsets

```
def CalcPrevalenceNumerator(df):
    numerator=[]

    for index, row in df.iterrows():
        numerator.append(row.Count*10000)

    return numerator
```

```
def CalcPrevalenceRate(df):
    rate=[]

    for index, row in df.iterrows():
        rate.append(row.x/row.Population)

    return rate
```

$$Rate = \left(\frac{10{,}000 x Count}{Population}\right)$$

	County	Count	FIPS	Population
0	ALBANY	1708	36001.0	304596
1	ALLEGANY	205	36003.0	48800
2	BRONX	11749	36005.0	1397335
3	BROOME	2214	36007.0	199363
4	CATTARAUGUS	492	36009.0	79815

Adding Population Data to Subsets

7. Apply calculations to subsets

```
# Save calculation/values/list as variable x
x = CalcPrevalenceNumerator(sum_county_17)

# Add the values as a column with the respective counties
sum_county_17['x'] = x

# See what this looks like
sum_county_17.head(2)
```

	County	Count	FIPS	Population	x
0	ALBANY	1708	36001.0	304596	17080000
1	ALLEGANY	205	36003.0	48800	2050000

"Numerator" value (10,000xCount)

Adding Population Data to Subsets

8. Make new column containing the respective prevalence rates (“Rate”)

```
# Save calculated values as 'rate'
rate = CalcPrevalenceRate(sum_county_17)

# Add new column containing these values
sum_county_17['Rate'] = rate

# Drop column 'x'
sum_county_17 = sum_county_17.drop(['x'], axis=1)

# See the new dataframe containing our new column!
sum_county_17.head()
```

	County	Count	FIPS	Population	Rate
0	ALBANY	1708	36001.0	304596	56.074275
1	ALLEGANY	205	36003.0	48800	42.008197

← Prevalence Rate

Interactive Map – Prevalence Rates

- Same code as before – small changes:

```
fig_17 = px.choropleth_mapbox(sum_county_17,
                              geojson=counties,
                              locations='FIPS',
                              color='Rate',
                              hover_name = "County",
                              color_continuous_scale="Blues",
                              range_color=(0, 110),
                              mapbox_style="carto-positron",
                              zoom=5.5, center = {"lat": 43.2994, "lon": -74.2179},
                              opacity=0.9,
                              labels={'Rate':'Prevalence Rate per 10,000 people'}
                             )
fig_17.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig_17.show()
```

	County	Count	FIPS	Population	Rate
0	ALBANY	1708	36001.0	304596	56.074275
1	ALLEGANY	205	36003.0	48800	42.008197

Interactive Map – Prevalence Rates

- Same code as before – small changes:

```
fig_17 = px.choropleth_mapbox(sum_county_17,
                              geojson=counties,
                              locations='FIPS',
                              color='Rate',
                              hover_name = "County",
                              color_continuous_scale="Blues",
                              range_color=(0, 110),
                              mapbox_style="carto-positron",
                              zoom=5.5, center = {"lat": 43.2994, "lon": -74.2179},
                              opacity=0.9,
                              labels={'Rate':'Prevalence Rate per 10,000 people'}
                             )
fig_17.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig_17.show()
```

Rate
56.074275
42.008197
84.081484
111.053706
61.642548

Interactive Map – Prevalence Rates

- Same code as before – small changes:

```
fig_17 = px.choropleth_mapbox(sum_county_17,
                              geojson=counties,
                              locations='FIPS',
                              color='Rate',
                              hover_name = "County",
                              color_continuous_scale="Blues",
                              range_color=(0, 110),
                              mapbox_style="carto-positron",
                              zoom=5.5, center = {"lat": 43.2994, "lon": -74.2179},
                              opacity=0.9,
                              labels={'Rate':'Prevalence Rate per 10,000 people'}
                             )
fig_17.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig_17.show()
```

	Rate
count	62.000000
mean	79.004159
std	33.347805
min	27.697822
25%	52.446979
50%	66.988705
75%	106.561754
max	159.893774

Prevalence Rates 2017-2018

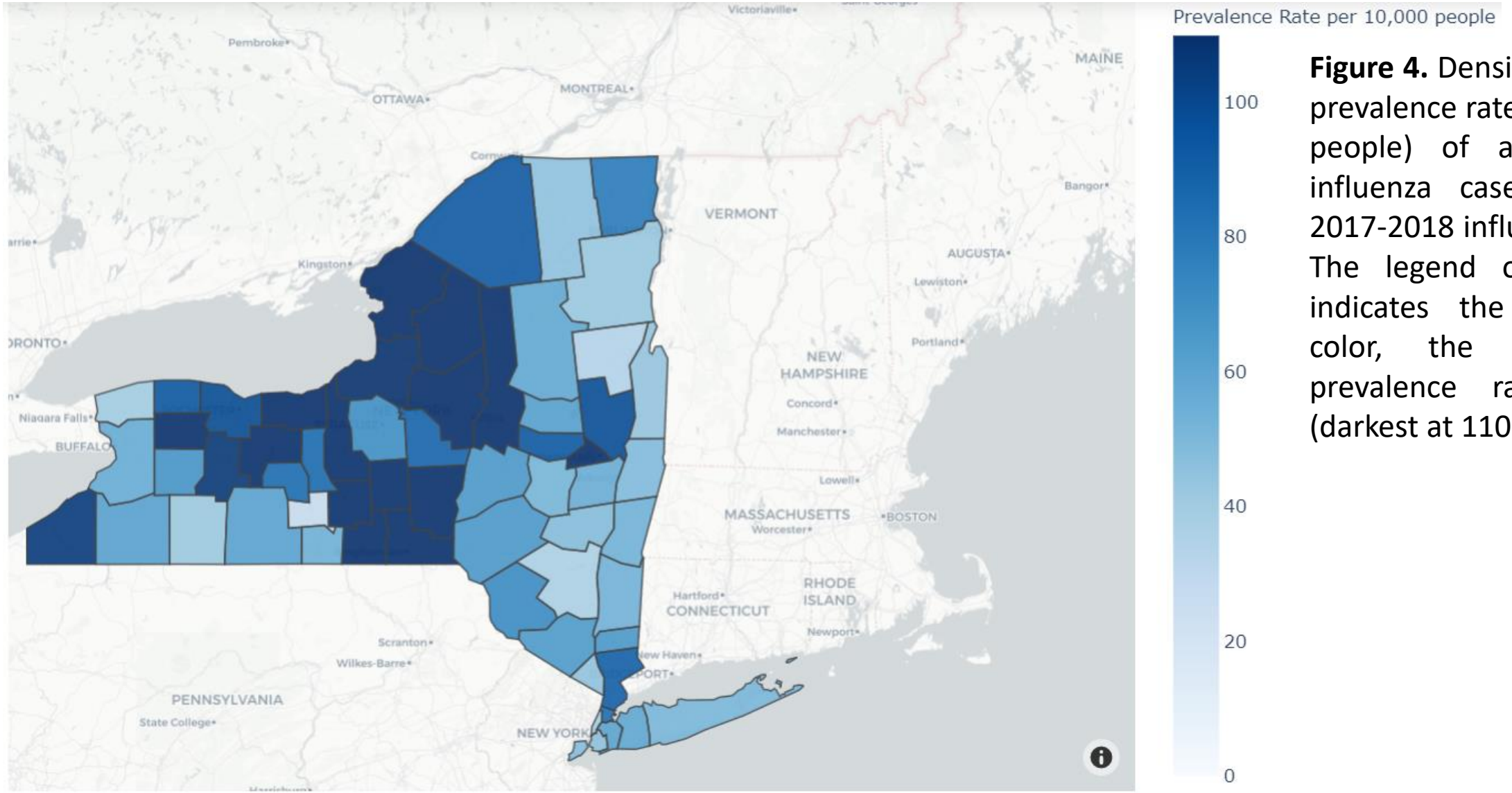


Figure 4. Density map of the prevalence rates (per 10,000 people) of all confirmed influenza cases from the 2017-2018 influenza season. The legend on the right indicates the darker the color, the higher the prevalence rate will be (darkest at 110).

Prevalence Rates 2017-2018

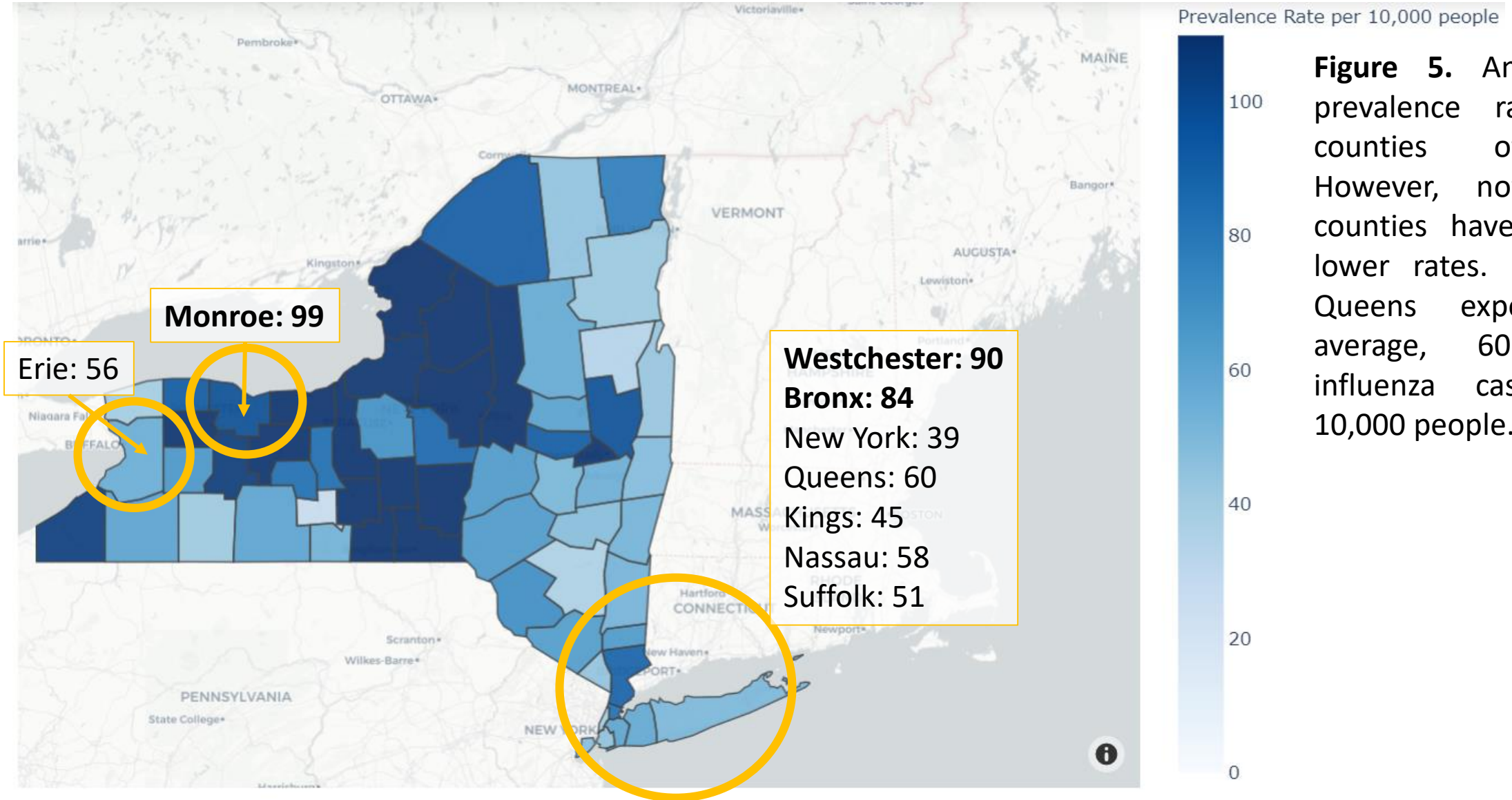


Figure 5. An outline of prevalence rates in the counties of interest. However, now – these counties have a relatively lower rates. For example, Queens experiences on average, 60 confirmed influenza cases out of 10,000 people.

Prevalence Rates 2017-2018

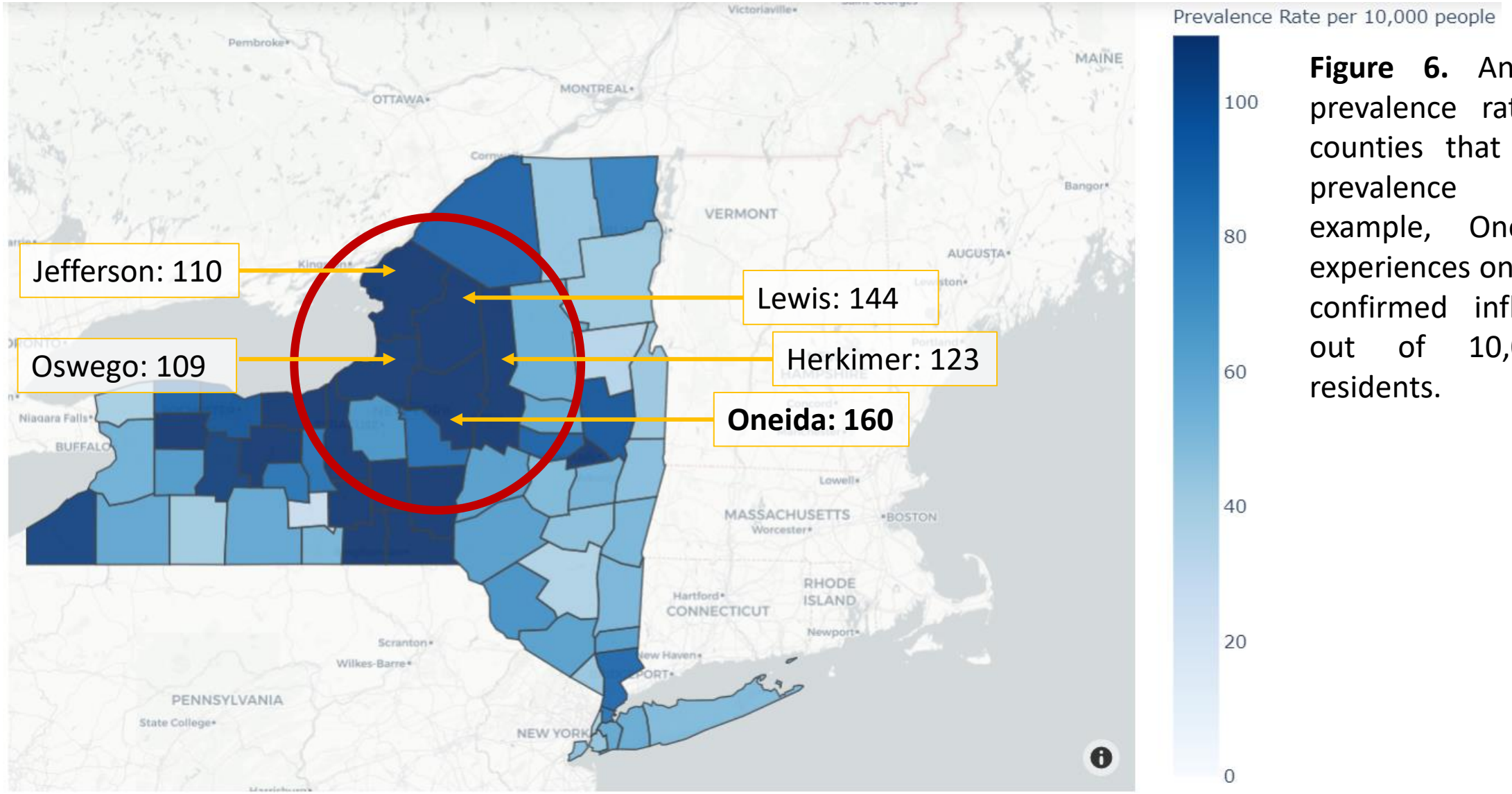


Figure 6. An outline of prevalence rates in some counties that have higher prevalence rates. For example, Oneida county experiences on average, 160 confirmed influenza cases out of 10,000 county residents.

Map Comparison (2017-2018 Season)

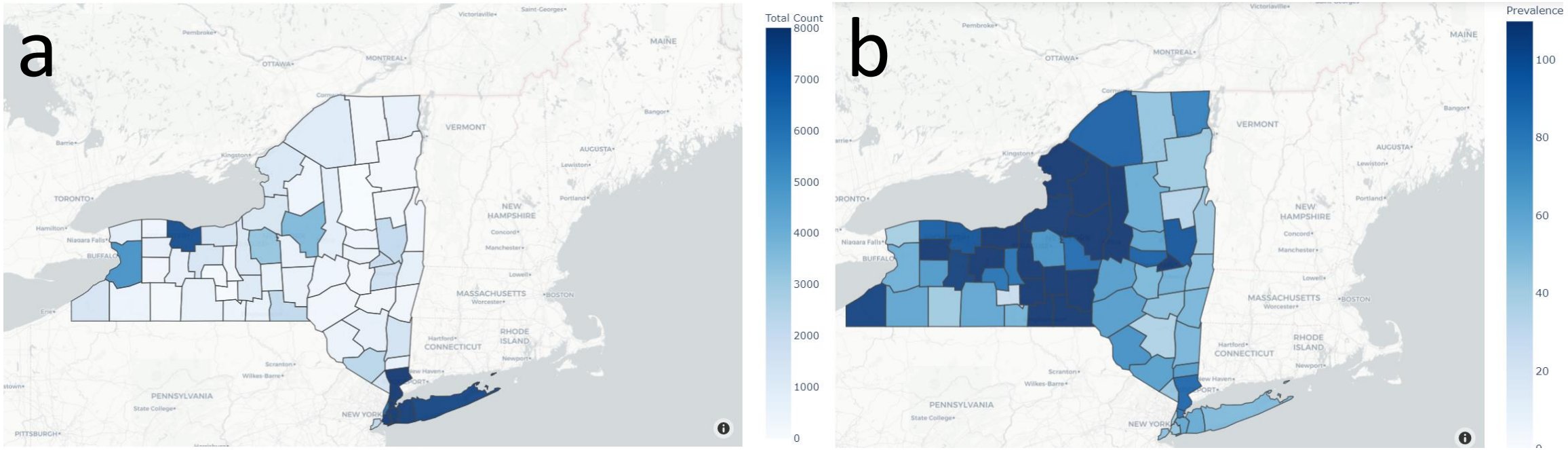


Figure 7. Side by side comparison of the two density maps from the 2017-2018 influenza season. The density map illustrating the total count of confirmed cases (a) is quite different from the map illustrating the prevalence rate of confirmed influenza cases per 10,000 people (b).

Conclusion

- Exploratory Data Analysis and Interactive Map Outcome:
  - Revealed that normalization of confirmed influenza cases with respect to population provides a different perspective that is important for identifying high risk counties
- Questions I would like to address:
  - Is there a trend across all seasons in the prevalence rate per county?
  - What are some potential factors that affect the prevalence rate in counties?

Thank you!
Questions/Feedback?

uv94014@umbc.edu

https://github.com/kathuerta/DATA606/tree/master/Deliverable%203