Kaggle_Ohio_Part1

April 9, 2023

[1]:
```
import matplotlib.pyplot as plt
import pandas as pd
import geopandas as gpd
import os
```

[2]:
```
df_train = pd.read_csv("C:/Users/palit/Downloads/training_data.csv")
df_train
```

[2]:

	county	cases	deaths	date_index_converted	county_data_length	\
0	Richland	0	0	day_10	363	
1	Lawrence	119	0	day_105	256	
2	Wayne	49	0	day_90	769	
3	Fayette	7	0	day_85	36	
4	Trumbull	0	0	day_7	554	
…	…	…	…	…	…	
3136	Summit	105	0	day_81	6121	
3137	Fayette	0	0	day_55	28	
3138	Clark	147	0	day_106	929	
3139	Logan	0	0	day_1	144	
3140	Paulding	0	0	day_60	16	

	core_jaccard	core_cosine	core_intersection	social_jaccard	\
0	0.000000	0.000000	0.000000	0.000011	
1	0.000000	0.000000	0.000000	0.000000	
2	0.000076	0.000370	0.006502	0.000039	
3	0.000000	0.000000	0.000000	0.000000	
4	0.000000	0.000000	0.000000	0.000046	
…	…	…	…	…	
3136	0.000094	0.000491	0.008169	0.000041	
3137	0.000000	0.000000	0.000000	0.000000	
3138	0.000013	0.000105	0.001076	0.000009	
3139	0.000000	0.000000	0.000000	0.000000	
3140	0.000000	0.000000	0.000000	0.000000	

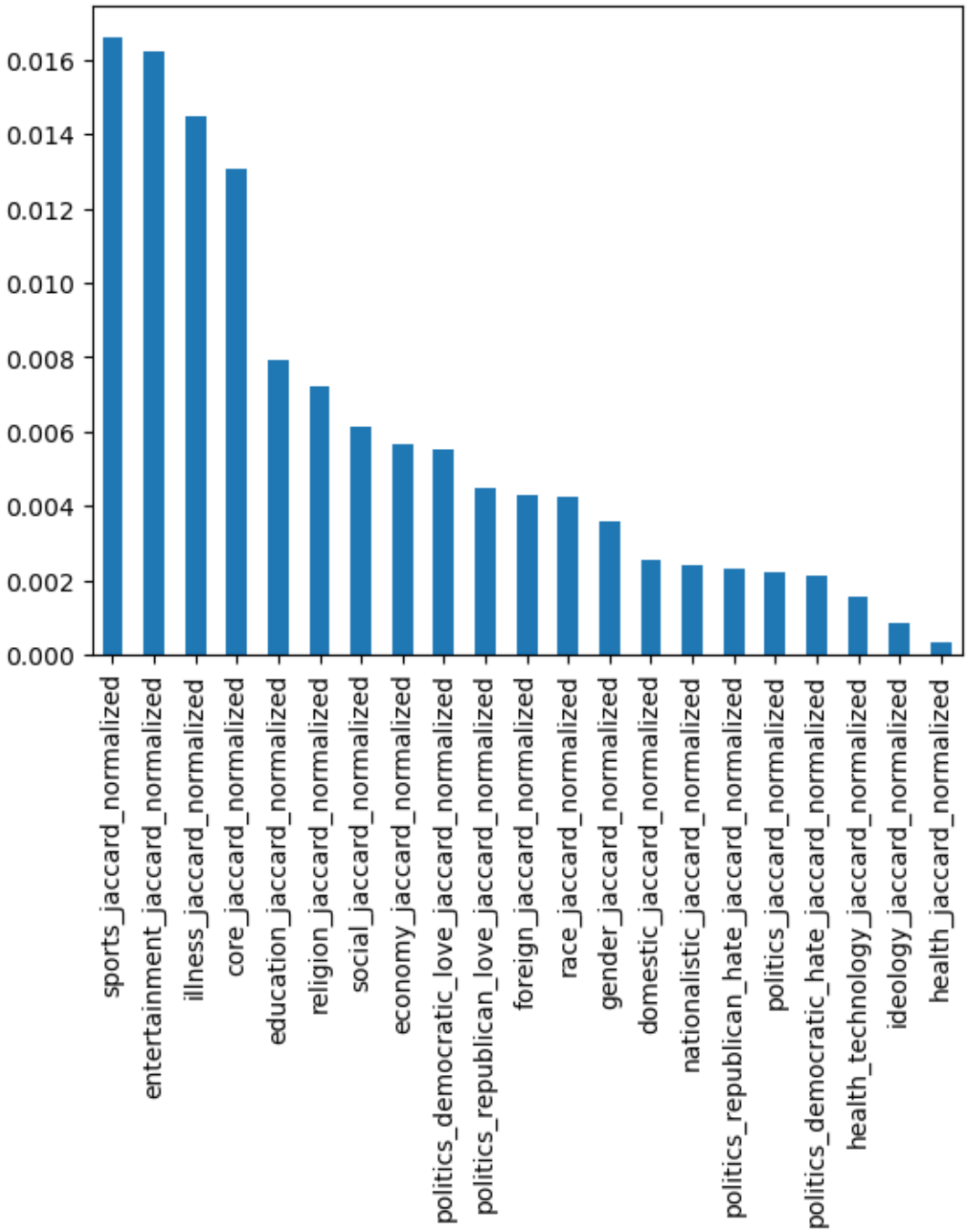
	politics_jaccard	…	labor_force_rate	unemployment_rate	\
0	0.000151	…	55.5	7.5	
1	0.000000	…	53.5	6.5	

```
2            0.000000  …              64.0               4.0
3            0.000000  …              59.3               6.3
4            0.000000  …              56.4               5.9
…               …  …                 …                  …
3136         0.000011  …              64.2               6.4
3137         0.000000  …              59.3               6.3
3138         0.000012  …              60.6               7.7
3139         0.000000  …              62.5               5.8
3140         0.000000  …              61.5               5.0

      median_housing_cost  median_household_earnings  median_worker_earnings  \
0                     675                      41877                   23210
1                     655                      42874                   23510
2                     762                      50383                   26658
3                     732                      40503                   25858
4                     661                      43073                   25800
…                     …                        …                       …
3136                  859                      50765                   28345
3137                  732                      40503                   25858
3138                  736                      43625                   25300
3139                  766                      49783                   28346
3140                  660                      45550                   25476

      percent_insured  percent_married  poverty_rate  median_property_value  \
0                90.5             48.3          15.6                 103700
1                92.2             49.3          18.6                 101500
2                87.2             55.1          13.0                 140100
3                91.3             51.8          17.7                 108900
4                91.7             49.1          17.2                 101600
…                …                …             …                    …
3136             93.2             47.3          13.6                 137000
3137             91.3             51.8          17.7                 108900
3138             92.7             48.1          16.6                 107300
3139             90.7             55.2          13.7                 127200
3140             93.0             57.3          10.7                  92500

      percent_white
0          0.868085
1          0.954027
2          0.950541
3          0.940054
4          0.885724
…             …
3136       0.791435
3137       0.940054
3138       0.865754
3139       0.949363
```

```
3140       0.949607

[3141 rows x 144 columns]
```

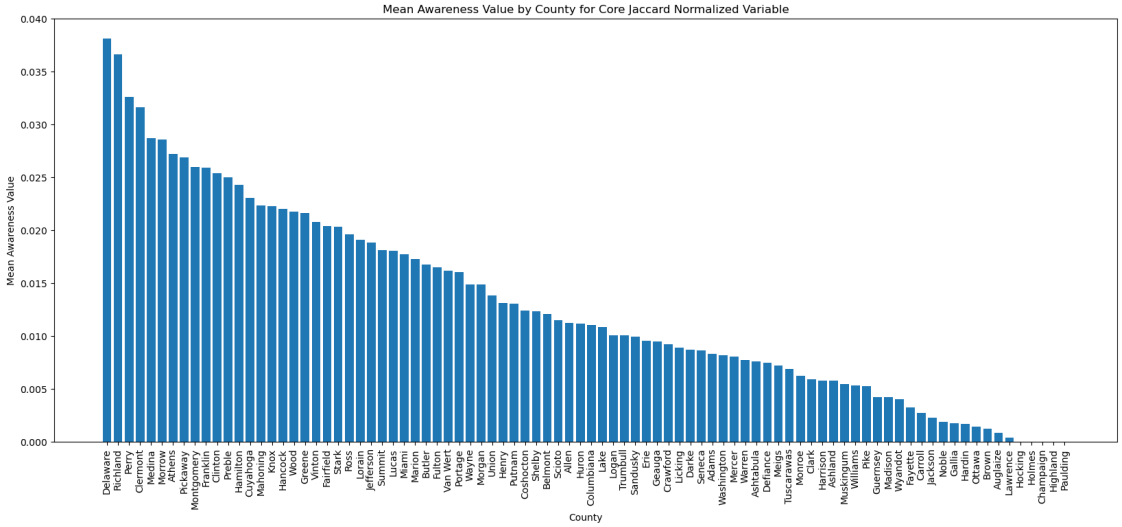
[3]:
```
# select columns with 'jaccard_normalized' in their names
jaccard_cols = [col for col in df_train.columns if 'jaccard_normalized' in col]

# calculate the mean of the selected columns
mean_jaccard = df_train[jaccard_cols].mean()

# create bar chart sorted in descending order
mean_jaccard.sort_values(ascending=False).plot(kind='bar')
```

[3]:
```
<AxesSubplot:>
```



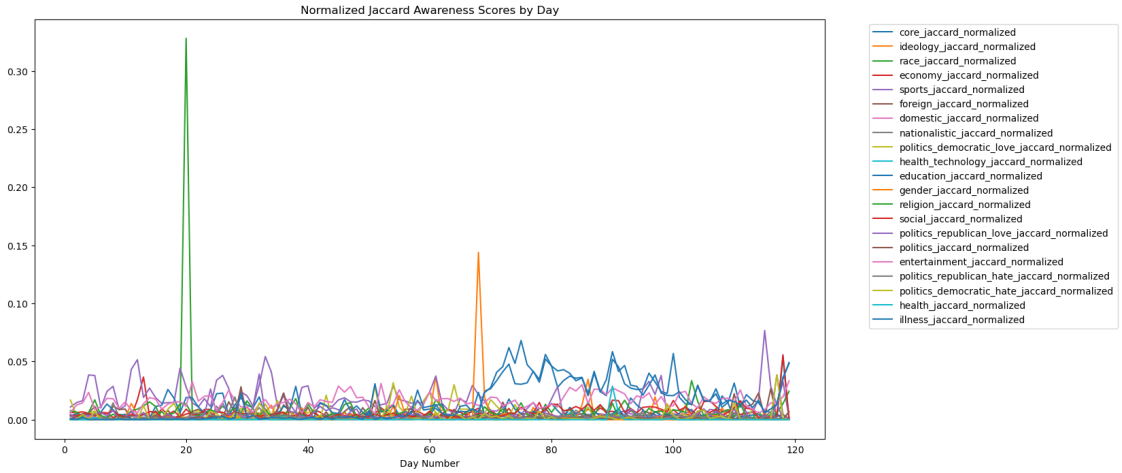
[4]:
```
# calculate mean awareness value for each county
mean_awareness = df_train.groupby('county')['core_jaccard_normalized'].mean().
  ↪sort_values(ascending=False)

# create bar chart
fig, ax = plt.subplots(figsize=(20, 8))
```

```
ax.bar(mean_awareness.index, mean_awareness)
plt.xticks(rotation=90)
plt.xlabel('County')
plt.ylabel('Mean Awareness Value')
plt.title('Mean Awareness Value by County for Core Jaccard Normalized Variable')
plt.show()
```

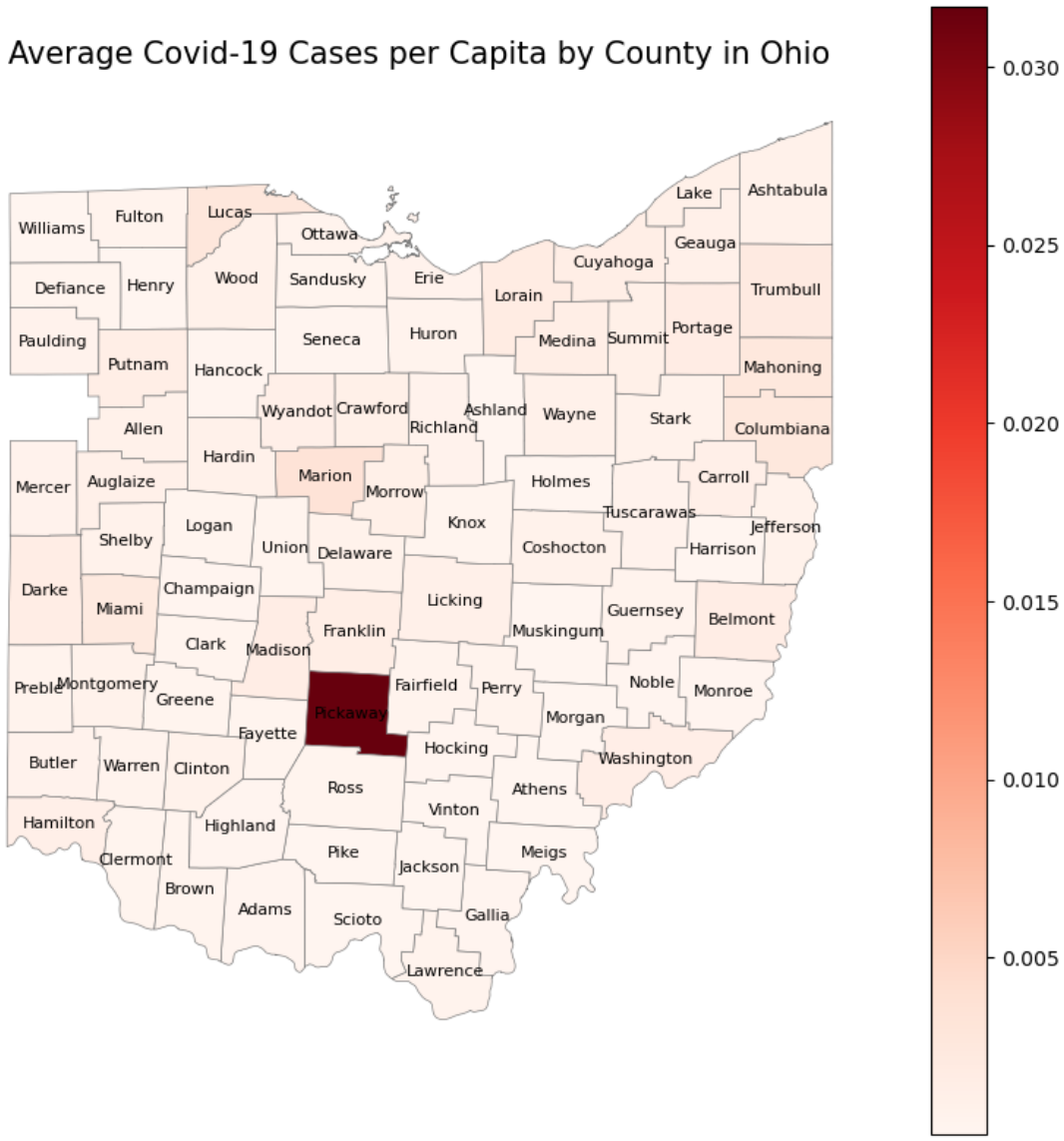


[5]:
```
df_train['date_index_converted'] = df_train['date_index_converted'].str[4:].
  ↪astype(int)
# calculate average normalized Jaccard scores for each day
jaccard_cols = [col for col in df_train.columns if 'jaccard_normalized' in col]
jaccard_averages = df_train.groupby('date_index_converted')[jaccard_cols].mean()

# create line chart with overlapping lines for each topic
jaccard_averages.plot(figsize=(15, 8), title='Normalized Jaccard Awareness␣
  ↪Scores by Day', xlabel = 'Day Number')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
```

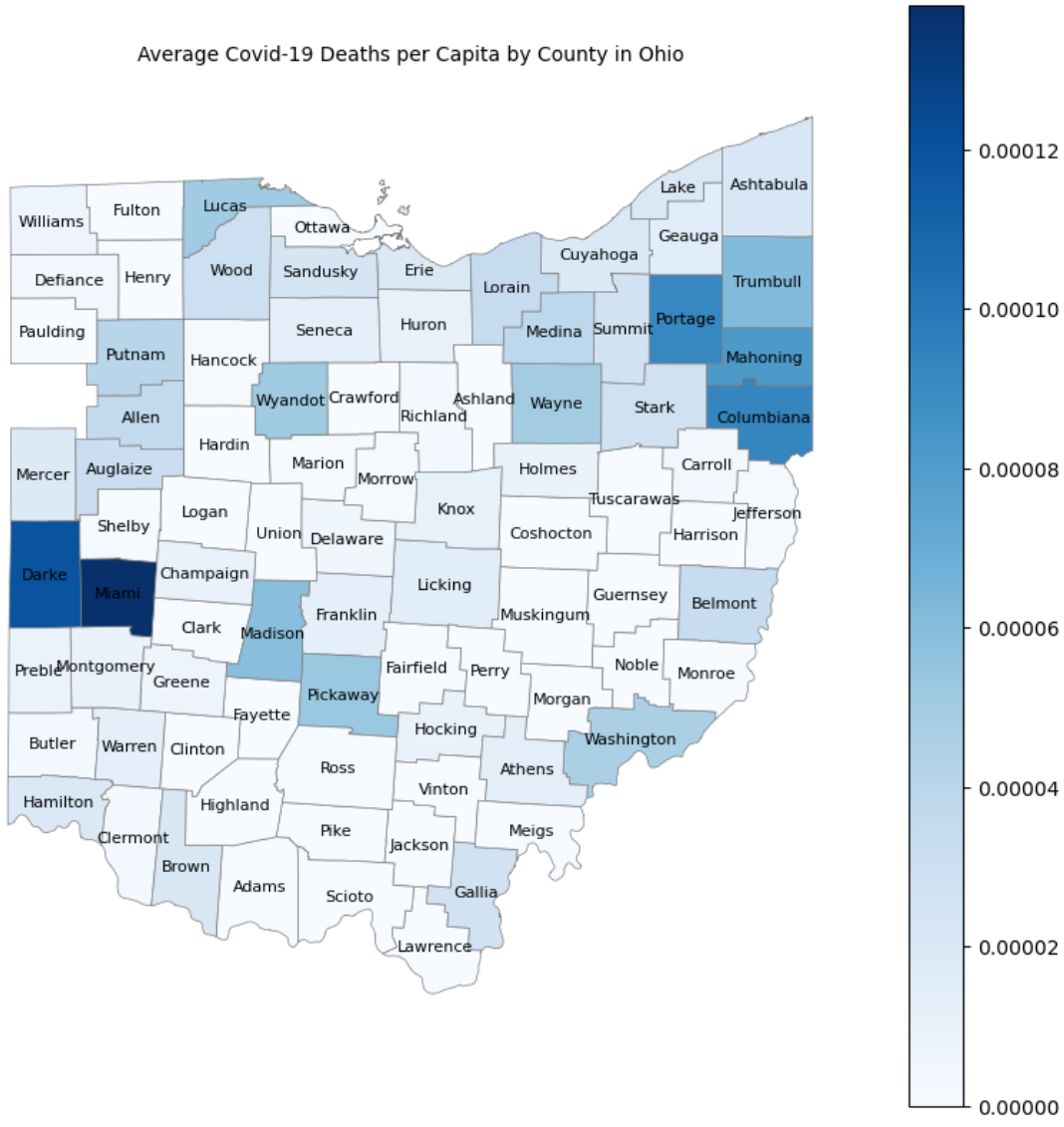
[5]: <matplotlib.legend.Legend at 0x1db680cdb80>



[14]:
```
os.environ['SHAPE_RESTORE_SHX'] = 'YES'

# Load the shapefile
shapefile = gpd.read_file("C:/Users/palit/Downloads/
 ↪tims_shp_datasets_20230409-1951/County.shp")
ohio_counties = shapefile.rename(columns = {'COUNTY':'county'})
ohio_counties['county'] = ohio_counties['county'].str.capitalize()

# Calculate the number of cases and deaths per capita
df_train['cases_per_capita'] = df_train['cases'] / df_train['total_pop']
df_train['deaths_per_capita'] = df_train['deaths'] / df_train['total_pop']

avg_covid_data = df_train.groupby("county")[["cases_per_capita",␣
 ↪"deaths_per_capita"]].mean().reset_index()

# Merge the Covid-19 data with the shapefile based on the county name
merged_data = ohio_counties.merge(avg_covid_data, on='county')

# Plot the map for cases per capita
fig1, ax1 = plt.subplots(figsize=(10, 10))
merged_data.plot(column='cases_per_capita', cmap='Reds', linewidth=0.5,␣
 ↪edgecolor='gray', ax=ax1, legend = True)
ax1.axis('off')
ax1.set_title('Average Covid-19 Cases per Capita by County in Ohio',␣
 ↪fontdict={'fontsize': '15', 'fontweight' : '3'})
for idx, row in merged_data.iterrows():
    ax1.text(row.geometry.centroid.x, row.geometry.centroid.y, row['county'],␣
 ↪ha='center', va='center', fontsize=8)
plt.show()
```



[15]:
```
# Plot the map for deaths per capita
fig2, ax2 = plt.subplots(figsize=(10, 10))
merged_data.plot(column='deaths_per_capita', cmap='Blues', linewidth=0.5,␣
 ↪edgecolor='gray', ax=ax2, legend = True)
ax2.axis('off')
ax2.set_title('Average Covid-19 Deaths per Capita by County in Ohio',␣
 ↪fontdict={'fontsize': '10', 'fontweight' : '3'})
for idx, row in merged_data.iterrows():
    ax2.text(row.geometry.centroid.x, row.geometry.centroid.y, row['county'],␣
 ↪ha='center', va='center', fontsize=8)
plt.show()
```



0.0.1 Interpretations in pdf attached

[ ]:

Part-1

(a) Check out this page: https://en.wikipedia.org/wiki/COVID-19_pandemic_in_Ohio. In around 250 words, summarize the Covid-19 experience of Ohio. Specifically, focus on how Ohio is different or similar to other US states in terms of the intensity of the pandemic (i), the time and the content of the different policies that have been implemented (ii), and if Wikipedia 'thinks' Ohio has dealt with Covid-19 successfully (or not) (iii).

Ohio, a midwestern state in the US, has experienced the Covid-19 pandemic with varying intensity over the past two years. Ohio reported its first positive case on March 9, 2020, and since then, the state has reported over 2.5 million confirmed cases and more than 40,000 deaths as of April 2023.
(i) Compared to some other US states, Ohio has not been hit as hard by the pandemic. According to the Centers for Disease Control and Prevention (CDC), Ohio has reported lower case and death rates than the national average. However, like most US states, Ohio has experienced surges in cases and deaths, particularly between November-March months of 2020 and 2021.
(ii) Ohio has implemented a variety of policies to mitigate the spread of the virus. These policies have included mask mandates, capacity limits on businesses, and social distancing guidelines. Governor Mike DeWine has been particularly proactive in implementing mask mandates in July 2020 which required businesses to post face covering requirement signs at all public entrances.
(iii) Wikipedia does not state whether or not Ohio successfully dealt with Covid-19. The page does, however, mention that Ohio has taken several steps to combat the pandemic, including providing financial assistance to businesses and individuals affected by the pandemic, expanding Covid-19 testing, and implementing a vaccine rollout plan. By April 2023, approximately 60% of Ohio residents had been fully immunized against Covid-19.

(b) Find the average values for all the topic awareness variables. Create a bar chart that shows the average normalized Jaccard similarity-based awareness values for all different types of awareness topics listed above. Order the bars from the biggest to the smallest. Summarize your observations in around 100 words.

The bar chart shows the average normalized Jaccard similarity-based awareness values for all different types of awareness topics. The normalized Jaccard value for Sports is the highest about 0.016 whereas the normalized Jaccard value for Health is the smallest, close to 0. This suggests that the tweets in the Sports topic are more similar to each other compared to the tweets in the Health topic.

(c) Focus on the core_jaccard_normalized variable. Create a bar chart that shows the aggregated mean awareness value for each county. Order the bars from the biggest to the smallest. Which county has the highest awareness? Summarize your observations in around 100 words.

From the plot we can see that the most awareness is in Delaware. This plot is based on the mean awareness value for each county (general awareness on COVID-19). Delaware being the highest indicates that people in Delaware were more aware of COVID-19 and were more active on social media discussions.

(d) Create two county-level maps of Ohio (an example is provided in the first page of the assignment). Using colors, show the number of average Covid-19 cases per capita and the number of average Covid-19 deaths per capita by county. What are the top-5 counties with high number of per capita cases and per capita number of deaths? Summarize your observations in around 100 words.

For this plot we have used a heat map in which dark colors show higher values and light colors show lower values.
From the heat map we can see that the top 5 counties with a high number of per capita cases are: Pickaway, Marion, Lucas, Columbiana and Mahoning. Similarly, the top 5 counties with a high number of per capita deaths are: Miami, Darke, Columbiana, Portage and Mahoning.

(e) Calculate the average normalized Jaccard awareness scores for every day (starting from Day 1). Create a line chart with overlapping lines in which each line represents the evolution of awareness levels for each topic.4 The x-axis of the line chart should correspond to ‘Days’, and the y-axis of the line chart should represent the level of awareness. What are the trends in the graph? Summarize your observations in around 100 words.

From the plot we can see that there is a spike on Day 20 with Jaccard score almost equal to 0.35 where Race related topics have been talked about the most, followed by the Jaccard score of Gender related topics equal to 0.15 on Day 68 and a Jaccard score of 0.075 for Sports on Day 115. The other topics have an almost constant Jaccard score throughout the 120 days.