SI 206 Final

By: Ally Devico and Anders Nelson

Github Link: https://github.com/andersne/SI206FinalProject-andersne-adevico

Original Goals

- Compare COVID information and demographics just in Michigan (we had no intentions in using state parameters) across counties
- Have a preset state hard coded in to find info for
- Simply use one website for demographics and one API for COVID information
- Create at least 2 visualizations
- Create 1 visualization that incorporates two sets of data overlaid on the same axis

Achieved Goals

- The ability to compare COVID information and demographics between any two states by including state parameters in all our functions
- Have the ability to ask the user for input on which two states they would like to compare and visualize
- We ended up using one html and two API's
- We created 5 visualizations
- We created 3 visualizations that showcase comparisons between two varying sets of data (different states). This included a double histogram and two double scatter plots.

Problems Faced

- On our html website, certain state county webpages had 404 errors and therefore we were unable to scrape information for those counties
- For some counties, popular city names was also included in the html text and therefore we had to make sure to remove the city name so that we can sort the counties alphabetically to match the way the counties were listed in the API
- We realized that certain states, such as Michigan, only have 83 counties and therefore we could only collect 83 rows of data. We ended up changing our plan so that we can look at any states including those with over 100 counties and therefore over 100 rows of data.
- When the user inputted a state with multiple words (ex: “New Jersey”), we had to account for the fact that websites and variables can’t have spaces so we had to replace spaces with either “-” or “_”

Calculations From Database

Kansas_covidCalculatedData.csv × visualizations.py

Kansas_covidCalculatedData.csv

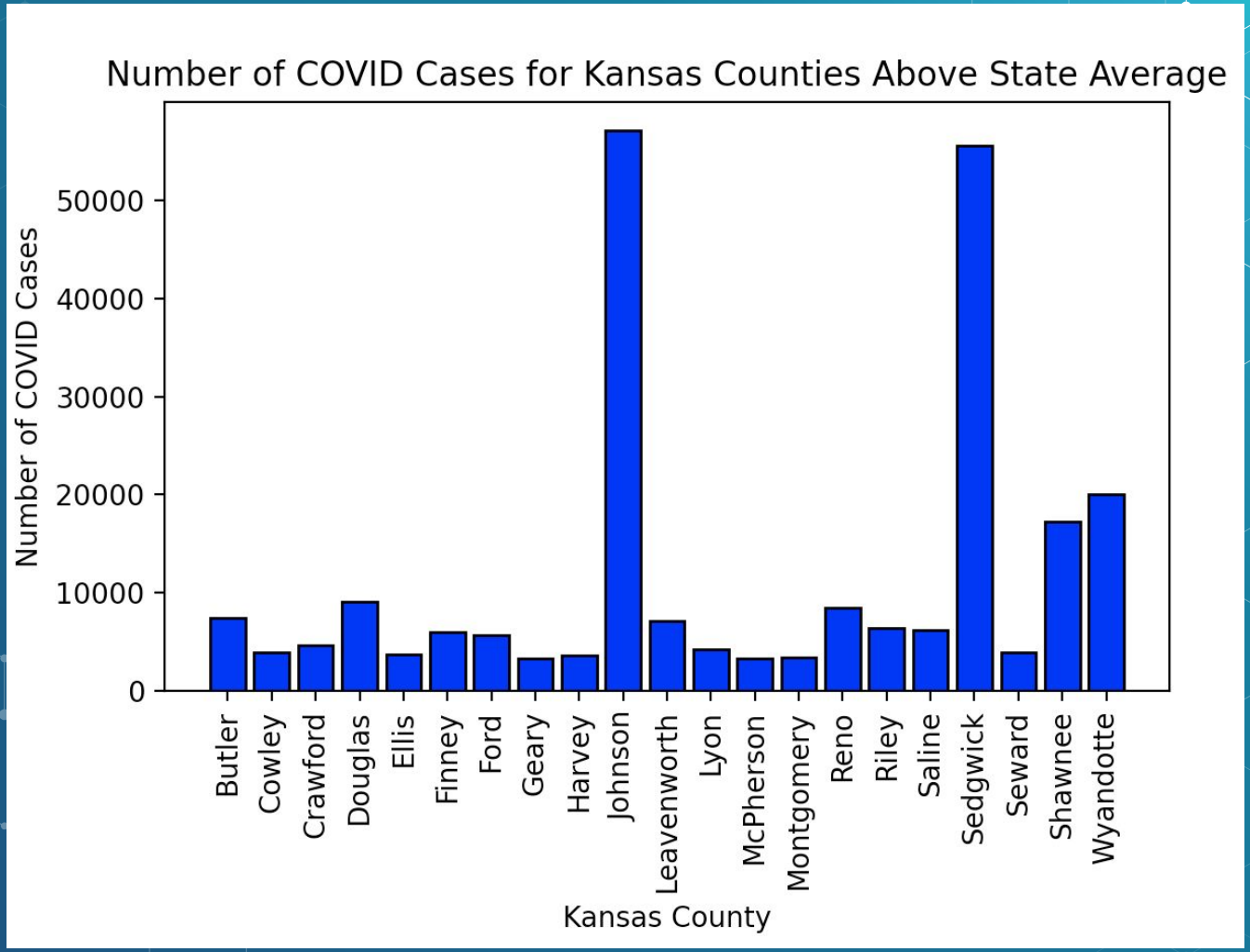
```
County,Positivity Rate,Death Rate
Allen,0.09221664544160611,7.491197842535021e-05
Anderson,0.10463359355326114,0.0
Atchison,0.09428690327566733,0.000772843469472683
Barber,0.07600901454619954,0.0004097520999795124
Barton,0.09413641802951735,0.0007615041520107336
Bourbon,0.09258761917550691,0.00020142339196992077
Brown,0.12443233424159855,0.003229387425572712
Butler,0.11332724783204016,0.00124752776509965
Chase,0.10465116279069768,0.0010901162790697674
Chautauqua,0.07420890506860824,0.0008401008120974517
Cherokee,0.11563663791071513,0.0007542544666006694
Cheyenne,0.13455087588520312,0.003727171077152441
Clark,0.11402298850574713,0.0018390804597701149
Clay,0.09691423207790684,0.0031678986272439284
Cloud,0.10426236513193035,0.00021365238756543105
Coffey,0.08515901060070671,0.0016489988221436984
Comanche,0.08849557522123894,0.004685059864653826
Cowley,0.10787992495309569,0.0002483169628076371
Crawford,0.11919418019026301,0.0020857709721727627
Decatur,0.09169550173010381,0.004152249134948097
Dickinson,0.08714530950217106,0.001868120771483389
Doniphan,0.12263311729571737,0.0008895666539585716
Douglas,0.08020841626637651,0.0007607724494219014
Edwards,0.08815612382234186,0.0
Elk,0.0669409124906507,0.0
Ellis,0.1280712003023951,0.0020961478986976393
Ellsworth,0.18804079110012362,0.0006180469715698393
Finney,0.1622668927100993,0.0013185866903474071
Ford,0.16501802451333814,0.00028839221341023794
Franklin,0.09746467969808399,0.0016257015676407974
Geary,0.0851347465823248,0.0006796141882531301
Gove,0.1414401175606172,0.008082292432035268
```

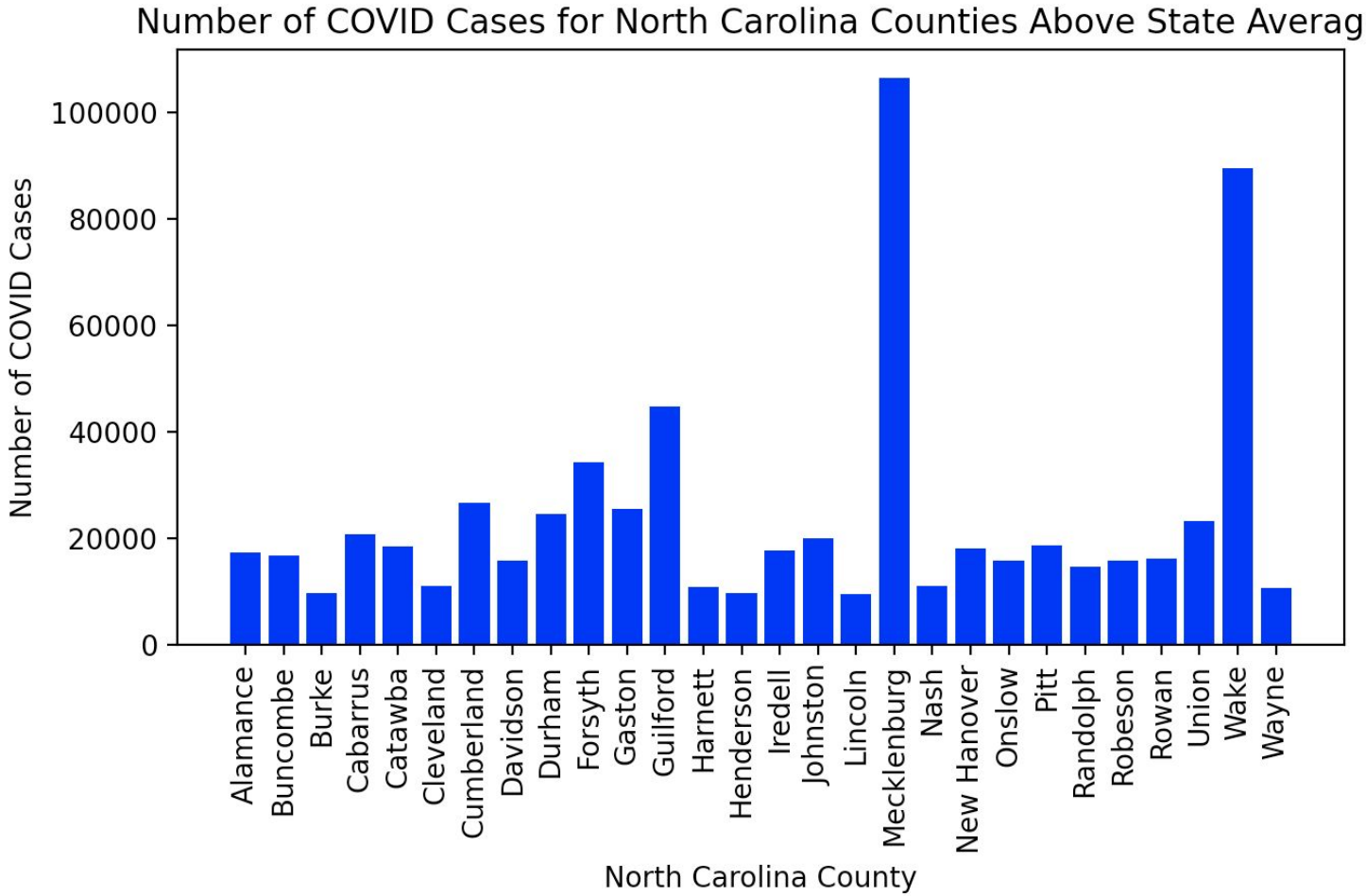
North_Carolina_covidCalculatedData.csv × visualizations.py

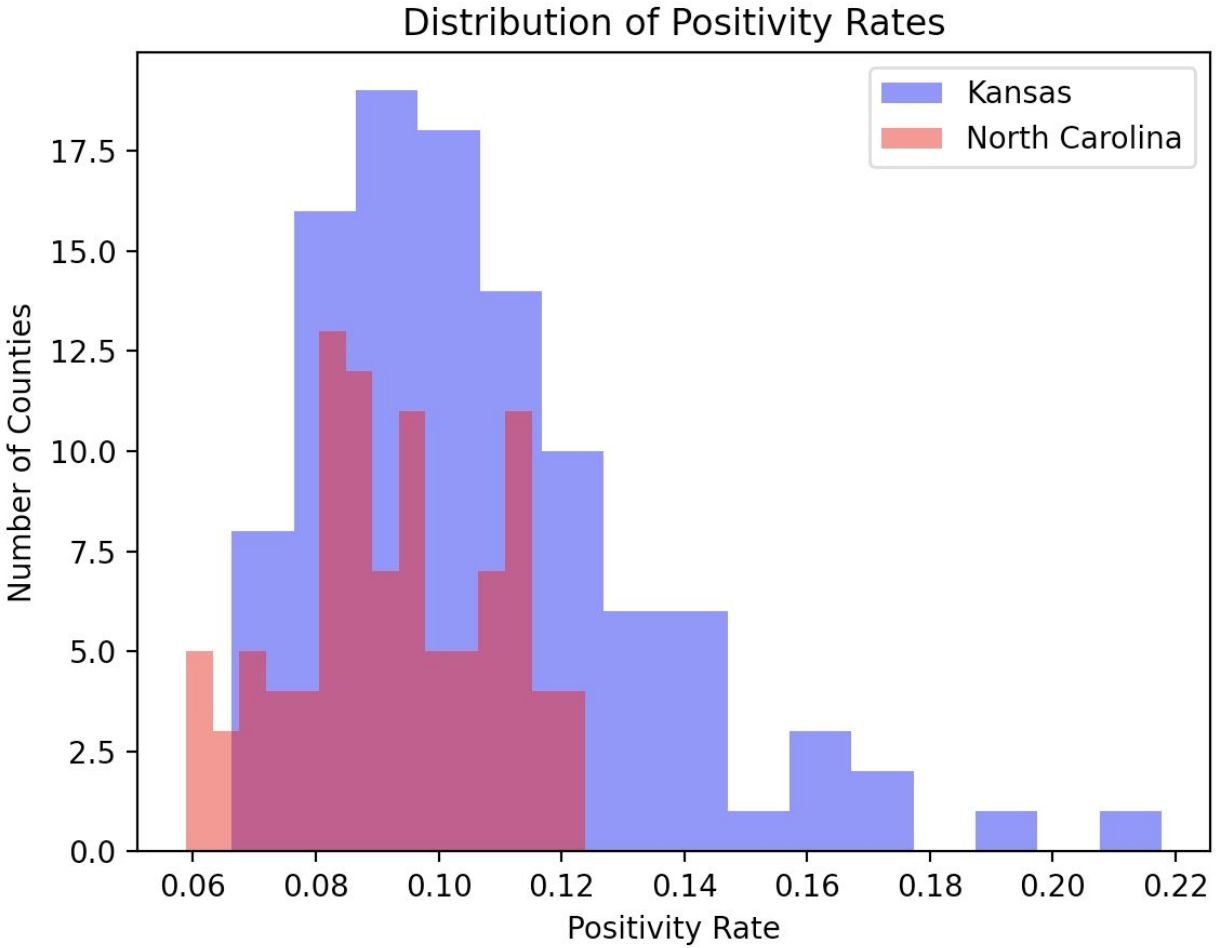
North_Carolina_covidCalculatedData.csv

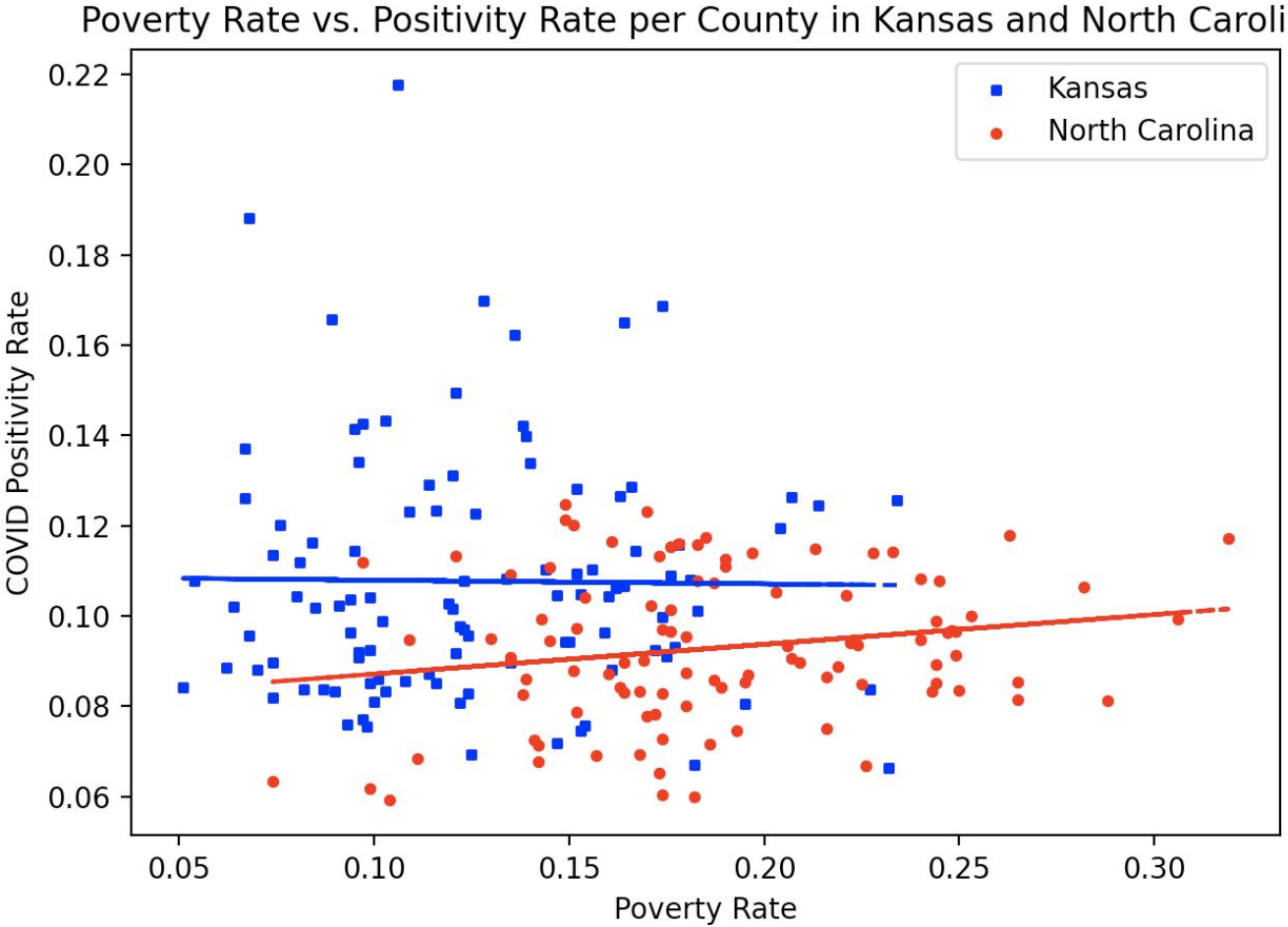
```
County,Positivity Rate,Death Rate
Alamance,0.11280519548128481,0.0017114373470769l
Alexander,0.11509020583700738,0.002245002840063834
Alleghany,0.08863988339254805,0.00045549785916006196
Anson,0.09349979129510871,0.0020870489128372483
Ashe,0.07422680412371134,0.0015095729013254787
Avery,0.11546532070549538,0.0011342369420972041
Beaufort,0.09302276604258367,0.001979655876840132
Bertie,0.08279481074777707,0.0020893056702784123
Bladen,0.08893212773263134,0.0011731715691885086
Brunswick,0.07825521181493267,0.0013101954597716516
Buncombe,0.06874307149197623,0.001313092885981813
Burke,0.10709226137772268,0.0016491057198512484
Cabarrus,0.11214455973459736,0.0013411303502684976
Caldwell,0.11054982063108573,0.001244600629621495
Camden,0.06287306008754477,0.0004974134500596896
Carteret,0.07210332103321034,0.0006937269372693727
Caswell,0.09040622299049265,0.001166810717372515
Catawba,0.11975888949317587,0.0019099785046486934
Chatham,0.06802834640889846,0.001320202127498141
Cherokee,0.08697099733649008,0.000739863865048831
Chowan,0.09543878730201891,0.0014954795731085583
Clay,0.06613433536871771,0.001690934711131987
Cleveland,0.11335180623973727,0.0022885878489326767
Columbus,0.10753042904519647,0.002587119962495442
Craven,0.08386232324768131,0.0011688459784096091
Cumberland,0.08278975019426583,0.0009101857212293O8
Currituck,0.061535269709543566,0.0006224066390041493
Dare,0.05889851377582061,0.0002317698525364313
Davidson,0.09685626356380043,0.0011218872227467232
Davie,0.09440981784799729,0.0008938493501473644
Duplin,0.0985917840179985,0.002133155570369136
Durham,0.08701755878630606,0.0007834629060447176
```

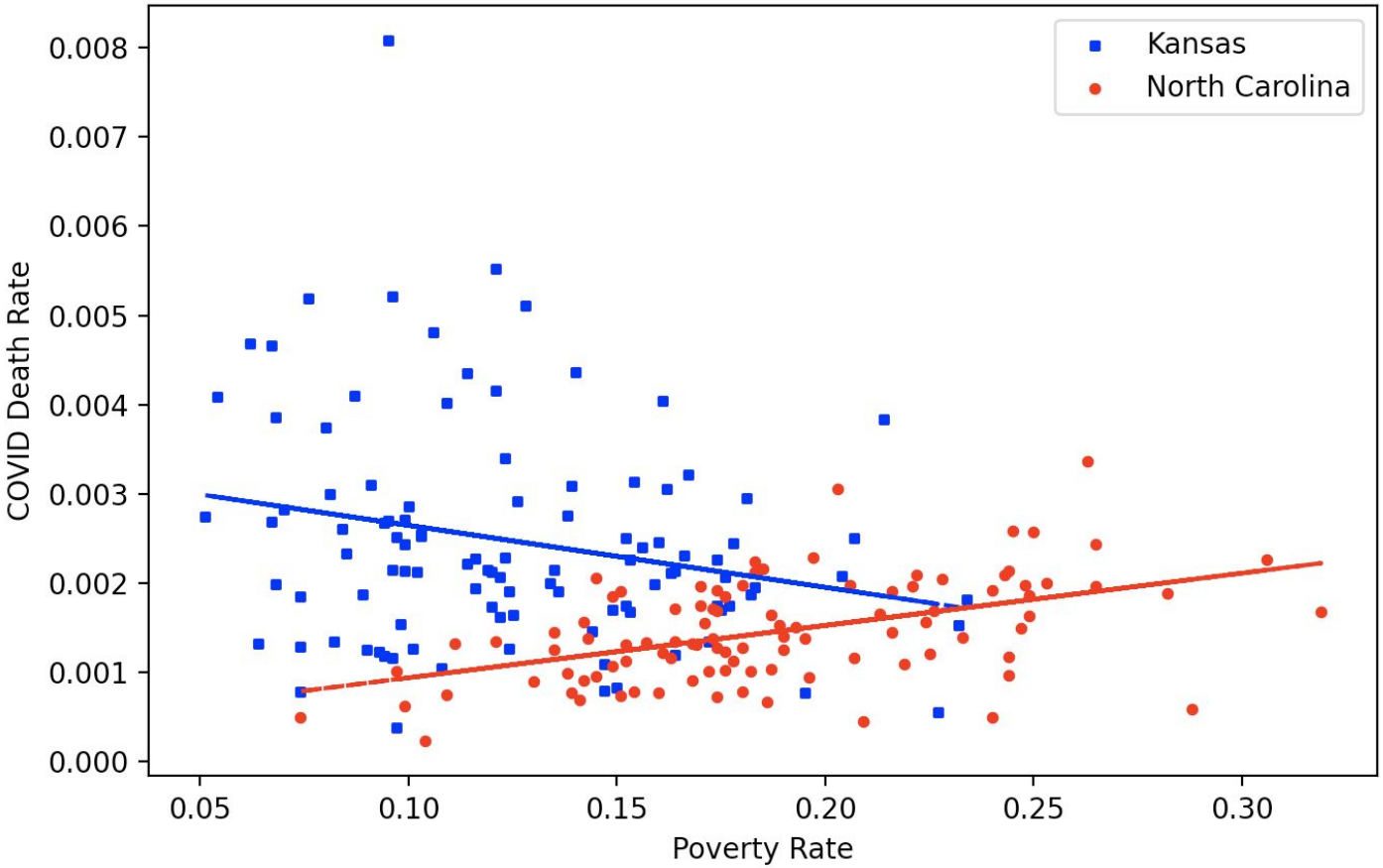
Visualizations


Number of COVID Cases for Kansas Counties Above State Average
Number of COVID Cases
0
10000
20000
30000
40000
50000
Butler
Cowley
Crawford
Douglas
Ellis
Finney
Ford
Geary
Harvey
Johnson
Leavenworth
Lyon
McPherson
Montgomery
Reno
Riley
Saline
Sedgwick
Seward
Shawnee
Wyandotte
Kansas County


Number of COVID Cases for North Carolina Counties Above State Average
Number of COVID Cases
0
20000
40000
60000
80000
100000
Alamance
Buncombe
Burke
Cabarrus
Catawba
Cleveland
Cumberland
Davidson
Durham
Forsyth
Gaston
Guilford
Harnett
Henderson
Iredell
Johnston
Lincoln
Mecklenburg
Nash
New Hanover
Onslow
Pitt
Randolph
Robeson
Rowan
Union
Wake
Wayne
North Carolina County


Distribution of Positivity Rates
Kansas
North Carolina
Number of Counties
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22
Positivity Rate


Poverty Rate vs. Positivity Rate per County in Kansas and North Carolina
Kansas
North Carolina
COVID Positivity Rate
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.05
0.10
0.15
0.20
0.25
0.30
Poverty Rate


Poverty Rate vs. COVID Death Rate per County in Kansas and North Carolina
Kansas
North Carolina
COVID Death Rate
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
0.05
0.10
0.15
0.20
0.25
0.30
Poverty Rate


Tables

KansasCountyNames

	countyID	countyName
	Filter	Filter
1	1	Allen
2	2	Anderson
3	3	Atchison
4	4	Barber
5	5	Barton
6	6	Bourbon
7	7	Brown
8	8	Butler
9	9	Chase
10	10	Chautauqua
11	11	Cherokee
12	12	Cheyenne
13	13	Clark
14	14	Clay
15	15	Cloud
16	16	Coffey
17	17	Comanche
18	18	Cowley
19	19	Crawford
20	20	Decatur
21	21	Dickinson
22	22	Doniphan
23	23	Douglas
24	24	Edwards

KansasCovidInfo

	countyID	covidNumberCases	covidDeaths
	Filter	Filter	Filter
1	1	1231	1
2	2	831	0
3	3	1586	13
4	4	371	2
5	5	2596	21
6	6	1379	3
7	7	1233	32
8	8	7449	82
9	9	288	3
10	10	265	3
11	11	2453	16
12	12	361	10
13	13	248	4
14	14	826	27
15	15	976	2
16	16	723	14
17	17	170	9
18	18	3910	9
19	19	4686	82
20	20	265	12
21	21	1726	37
22	22	965	7
23	23	9067	86
24	24	262	0

KansasDemographics

	countyID	population	povertyRate
	Filter	Filter	Filter
1	1	13349	17.2
2	2	7942	15.3
3	3	16821	15.0
4	4	4881	9.3
5	5	27577	14.9
6	6	14894	17.7
7	7	9909	21.4
8	8	65730	7.4
9	9	2752	14.7
10	10	3571	15.3
11	11	21213	17.8
12	12	2683	9.6
13	13	2175	16.7
14	14	8523	12.3
15	15	9361	16.0
16	16	8490	9.9
17	17	1921	6.2
18	18	36244	18.1
19	19	39314	20.4
20	20	2890	12.1
21	21	19806	11.4
22	22	7869	12.6
23	23	113043	19.5
24	24	2972	16.1

North_CarolinaCountyNames

	countyID	countyName
	Filter	Filter
1	1	Alamance
2	2	Alexander
3	3	Alleghany
4	4	Anson
5	5	Ashe
6	6	Avery
7	7	Beaufort
8	8	Bertie
9	9	Bladen
10	10	Brunswick
11	11	Buncombe
12	12	Burke
13	13	Cabarrus
14	14	Caldwell
15	15	Camden
16	16	Carteret
17	17	Caswell
18	18	Catawba
19	19	Chatham
20	20	Cherokee
21	21	Chowan
22	22	Clay
23	23	Cleveland
24	24	Columbus

North_CarolinaCovidInfo

	countyID	covidNumberCases	covidDeaths
	Filter	Filter	Filter
1	1	17335	263
2	2	4255	83
3	3	973	5
4	4	2464	55
5	5	2016	41
6	6	2036	20
7	7	4417	94
8	8	1704	43
9	9	3108	41
10	10	8780	147
11	11	16805	321
12	12	9676	149
13	13	20654	247
14	14	9060	102
15	15	632	5
16	16	4885	47
17	17	2092	27
18	18	18497	295
19	19	4483	87
20	20	2351	20
21	21	1404	22
22	22	704	18
23	23	11045	223
24	24	6193	149

North_CarolinaDemographics

	countyID	population	povertyRate
	Filter	Filter	Filter
1	1	153672	17.3
2	2	36971	18.3
3	3	10977	20.9
4	4	26353	22.2
5	5	27160	19.3
6	6	17633	17.8
7	7	47483	20.6
8	8	20581	24.3
9	9	34948	24.4
10	10	112197	15.2
11	11	244461	16.8
12	12	90352	18.7
13	13	184173	12.1
14	14	81954	19.0
15	15	10052	7.4
16	16	67750	14.1
17	17	23140	20.7
18	18	154452	15.1
19	19	65899	11.1
20	20	27032	15.1
21	21	14711	24.7
22	22	10645	22.6
23	23	97440	19.7
24	24	57593	24.5

Instructions

1. Delete “covidRawData.db” and any created csv files from your folder before running the code .
2. Open up “Data_Collection_And_Storage.py” and “json.counter.” IMPORTANT: Make sure “counter.json” file has the number 0 in it’s file. If not, change the number to 0 and save the “counter.json” file. Then run “Data_Collection_And_Storage.py”
3. You will be prompted to type in two states you would like to compare COVID info to. Include capitalizations and spaces in each state name. Click enter after each state name. (For the grading rubric, select 2 states that have >= 100 counties to get at least 100 rows of data)
    a. The first time you run the code, 6 tables (3 for each state) will be created and the first 25 counties will be be filled into the first table for State 1 (the 1st table being {stateName}CountyNames).

4. If after running the code you are asked to “Run code again to keep filling the Database”: Run the code again.
    a. When you click run, 25 more items will be inputted into the database
5. Number of times you will have to click run will vary depending on the state and its county quantity. Can range anywhere from 3-15 times to fill all 6 tables.
6. If you receive the message: “Done filling tables and writing files. Time to run visualization.py”
    a. The database is completely filled and it's time to open the “visualizations.py” file.

7. Click run for the visualizations file and enter the two state names like you previously had for the “Data_Collection_And_Storage.py” file (IMPORTANT: Must be the SAME 2 states in the SAME order)
    a. This visualizations file will select data from the database, as well as the calculations saved in each state’s “{stateName}_covidCalculatedData.csv” file
    b. After running the code, 5 visualizations will pop-up. Exit out of one to view the next.
8. In the end, feel free to open “covidRawData.db” to see the 6 tables and “{stateName}_covidCalculatedData.csv” for each state to see the stored calculations
9. IMPORTANT: If you would like to redo the program again with two new states, you MUST return to the “counter.json” file and delete the existing number and replace & save it with the number 0. Also, don’t forget to delete the existing database and csv files that you just created.

Documentation

Data_Collection_And_Storage.py

```
def read_counter():
"""Finds the number of times the code has been executing by reading the counter.json file"""
def write_counter():
"""Updates the number of times the code has been executing bywriting into the counter.json file"""
def get_stateDict(state):
"""Takes in a state name as a string and returns a dictionary of COVID info on that state from a unique API URL link."""
def get_websiteLinks(state):
"""Takes in a state name as a string. Returns a list of county URL links as strings in alphabetical order for a certain state."""
```

Data_Collection_And_Storage.py

```
def get_countyNames(state):
"""Takes in a state name as a string. Returns a list of county names as strings in alphabetical order for a certain state."""
def get_populations(state):
"""Takes in a state name as a string. Returns a dictionary with the keys being county names and the values being the population for each county in the state."""
def get_povertyRates(state):
"""Takes in a state name as a string. Returns a dictionary with the keys being county names and the values being the poverty rate for each county in the state."""
def get_covidNumberCases(state_dic):
"""Utilizes the entire state_dic from the API to grab the number of positive COVID cases per county. Returns a dict with keys as county names and values as COVID case numbers."""
```

Data_Collection_And_Storage.py

```
def get_covidDeaths(state_dic):
"""Utilizes the entire state dic from the API to grab the number of COVID deaths per county. Returns a dictionary with keys as county names and values as COVID death counts."""
def setUpDatabase(db_name):
"""Takes the name of a database, a string, as an input. Returns the cursor and connection to the database."""
def set_up_tables(state1, state2, cur, conn):
"""Takes in the database cursor, connection, and two state names as strings. Sets up three tables for each of the two states passed. The tables include a county name table, a demographic table, and a COVID information table.""”
def fill_countyNamesTable(state, cur, conn):
"""Takes in the database cursor, connection, and the state name as a string. Does not return anything. Fills in the passed state’s county name table with all the counties within that state (countyName) along with a countyID INTEGER PRIMARY KEY."""
```

Data_Collection_And_Storage.py

```
def fill_DemographicTable(state, cur, conn):
"""Takes in the database cursor, connection, and the state name as a string. Does not return anything. Fills in the passed state's county demographic table with countyID, population, and poverty rate."""
def fill_CovidInfoTable(state, cur, conn):
"""Takes in the database cursor, connection, and the state name as a string. Does not return anything. Fills in the passed state's county COVID info table with countyID, covidNumberCases, and covidDeaths."""
def get_percentCovidCases(state, cur, conn):
"""Takes in the database cursor, connection, and the state name as a string. Returns a dictionary with keys as county names and the values as percentages of the number of positive COVID cases vs the population (as a floating decimal) for each county in the state."""
```

Data_Collection_And_Storage.py

```
def get_percentCovidDeaths(state, cur, conn):
"""Takes in the database cursor, connection, and the state name as a string. Returns a dictionary with keys as county names and the values as percentages of the number of COVID deaths vs the number of positive cases (as a floating decimal) for each county in the state."""
def write_calculated_data_to_file(filename, state, cur, conn):
"""Takes in the database cursor and connection. Accepts the filename and state name as strings. Opens the file and writes the county, positivity rate, and death rate for each county in the state"""
def main():
"""Takes nothing as an input and returns nothing. Ask the user for two state names, sets up the database, and calls fill_countyNamesTable(), fill_DemographicTable(), fill_CovidInfoTable(), and write_calculated_data_to_file for each state. Closes the database connection."""
```

visulizations.py

```
def CovidCasesBarGraph(state, cur, conn):
 """Takes in the database cursor, connection, and the state name as a string. For the inputted state, creates a bar graph displaying the COVID case count for each county that has a case count above the state case count average"""
def DoubleHistogram(states, cur, conn):
"""Takes in the database cursor, connection, and a list of state names as a strings. Creates a double histogram for two states, comparing their range of Positivity Rates across their respective counties"""
def PovertyVSDeathScatter(states, cur, conn):
"""Takes in the database cursor, connection, and a list of state names as a strings. Creates a double scatterplot comparing the Poverty Rates and COVID Death Rates for each county within the two passed states"""
```

visulizations.py

```
def PovertyVSPositivityScatter(states, cur, conn):
"""Takes in the database cursor, connection, and a list of state names as a
strings. Creates a double scatter plot comparing the Poverty Rates and Positivity
Rates for each county within the two passed states"""
def main():
"""Takes nothing as an input and returns nothing. Ask the user for two state names
they inputted into the data collection program, establishes the database
connection, and calls CovidCasesBarGraph()for each state, DoubleHistogram(),
PovertyVSDeathScatter(), and PovertyVSPositivityScatter(). Closes the database
connection."""
```

Resources

Date	Issue Description	Location of Resource	Result
4/16	Needed to find a way to capitalize each word in a state. I realized .capitalize() only works for the first word	https://thispointer.com/python-capitalize-the-first-letter-of-each-word-in-a-string/#:~:text=Use%20title()%20to%20capitalize,of%20word%20to%20lower%20case	Implemented .title() on the state name string
4/17	Poverty Rate was scraped and stored as a whole number rather than a complete decimal which was a different scale than Positivity rate (14.6 vs .146)	https://stackoverflow.com/questions/49415175/python-convert-number-expressed-as-percentage-to-decimal	Poverty Rate was set to the same scale as Positivity and Death Rates for standardization of visualization scales
4/18	Needed a way to stop code from executing when certain county websites returned a 404 error	https://stackoverflow.com/questions/19782075/how-to-stop-terminate-a-python-script-from-running/34029481#:~:text=To%20stop%20a%20running%20program,want%20to%20terminate%20the%20program.&text=Ctrl%20%2B%20Z%20should%20do%20it,caught%20in%20the%20python%20shell	Imported sys and added sys.exit() in a try except statement
4/18	Was unsure how to make a trendline on a scatter plot	https://stackoverflow.com/questions/42339602/adding-multiple-trend-lines-4-for-multiple-data-sets-on-a-single-scatter-plot	Imported numpy as np to access ability to add trendlines

Date	Issue Description	Location of Resource	Result
4/20	Needed to find a way to count the number of times the code executed so that the user only had to input the two states on the first execution	https://stackoverflow.com/questions/30961310/how-can-i-count-how-many-times-this-program-has-been-executed-in-python	Created a json.counter file that is read and updated after each execution
4/20	For executions after the first one when the user did not input the two state names, we needed to find a way to receive the state names as they were no longer saved into the Data_Storage_and_Collection.py file	https://www.kite.com/python/answers/how-to-list-tables-using-sqlite3-in-python#:~:text=Cursor%20.-,Call%20sqlite3.,to%20return%20all%20table%20names .	Decided to grab the name of the titles created on the first execution which then let us save variables with the state names