The Impact of Economic and Social Conditions on the Development of the COVID-19 Pandemic

Cyrus Hatam

Thursday, July 16, 2020

Introduction

I am a rising sophomore at Princeton University interested in data science and international affairs, and for the past eight weeks, I have been interning at Syntasa. During my internship, I primarily focused on an end-to-end analytics project centered around COVID-19, combining and manipulating multiple county-level datasets from various sources and using this aggregation as a base for exploratory data analysis, descriptive modeling, and ultimately, forecasting the spread of the disease.

I used Syntasa in conjunction with Google BigQuery, Google Data Studio, and Jupyter Notebook throughout this project, and I created Syntasa apps for aggregating my datasets and updating the dynamic elements on a daily basis in BQ. In addition to automatically reloading my data on a schedule, Syntasa also allowed me to clearly visualize my various data sources that were being consolidated in an interactive workflow.

Before this project, I did not have very much exposure to any of the aforementioned services or to data science in general, but I was able to learn quickly with the help of my supervisor, Adam Neo. We had meetings almost every day to go over aspects of the project and to troubleshoot things that were not working. I also had the opportunity to learn about aspects of data science and machine learning that were not directly related to my project (e.g. neural networks), and through several meetings with people from different branches of the company (from sales to DevOps to product management), I was also able to gain exposure to the ins and outs of the start-up setting.

I greatly enjoyed my time at Syntasa both because I had the freedom to work on my own project and because I had the opportunity to meet and exchange ideas with so many of the company's employees.

Data Sources

The central data source for the project was a COVID-19 dataset from the *New York Times*,[1] which included laboratory confirmed COVID-19 cases as well as confirmed and probable COVID-19 deaths updated daily by county. This dataset was the source of all my dynamic data (data that changed over time) while all my other sources contained static data. The date range is from 01/21/20 (the date of the first US case) to two days ago. As of today, there are ~ 300,000 distinct entries for cases and deaths in the dataset.

My static county-level data sources were the following: the CDC's Social Vulnerability Index (SVI) data,[2] the CDC's diabetes data,[3] Harvard's smoking data,[4] data from the 2016 US presidential election,[5] BQ location data,[6] and BQ Census data.[7]

The SVI has detailed demographic and socioeconomic data and gives each county a social vulnerability percentile (essentially a score). I chose to include data about diabetes and smoking rates as well because these variables are assumed to have effects on COVID-19 deaths; my hypothesis was that counties with higher diabetes and smoking rates would have higher death rates (keeping everything else constant). I also wanted to see if political distribution had an effect

---

[1] https://github.com/nytimes/covid-19-data

[2] https://svi.cdc.gov/data-and-tools-download.html

[3] https://gis.cdc.gov/grasp/diabetes/diabetesatlas.html

[4] https://dataverse.harvard.edu/file.xhtml?persistentId=doi:10.7910/DVN/VZ21KD/WWFBCX&version=1.0

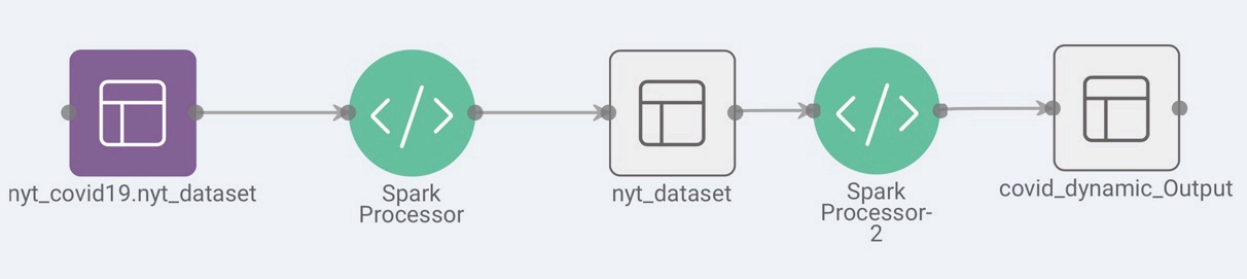
[5] https://github.com/tonmcg/US_County_Level_Election_Results_08-16/blob/master/2016_US_County_Level_Presidential_Results.csv

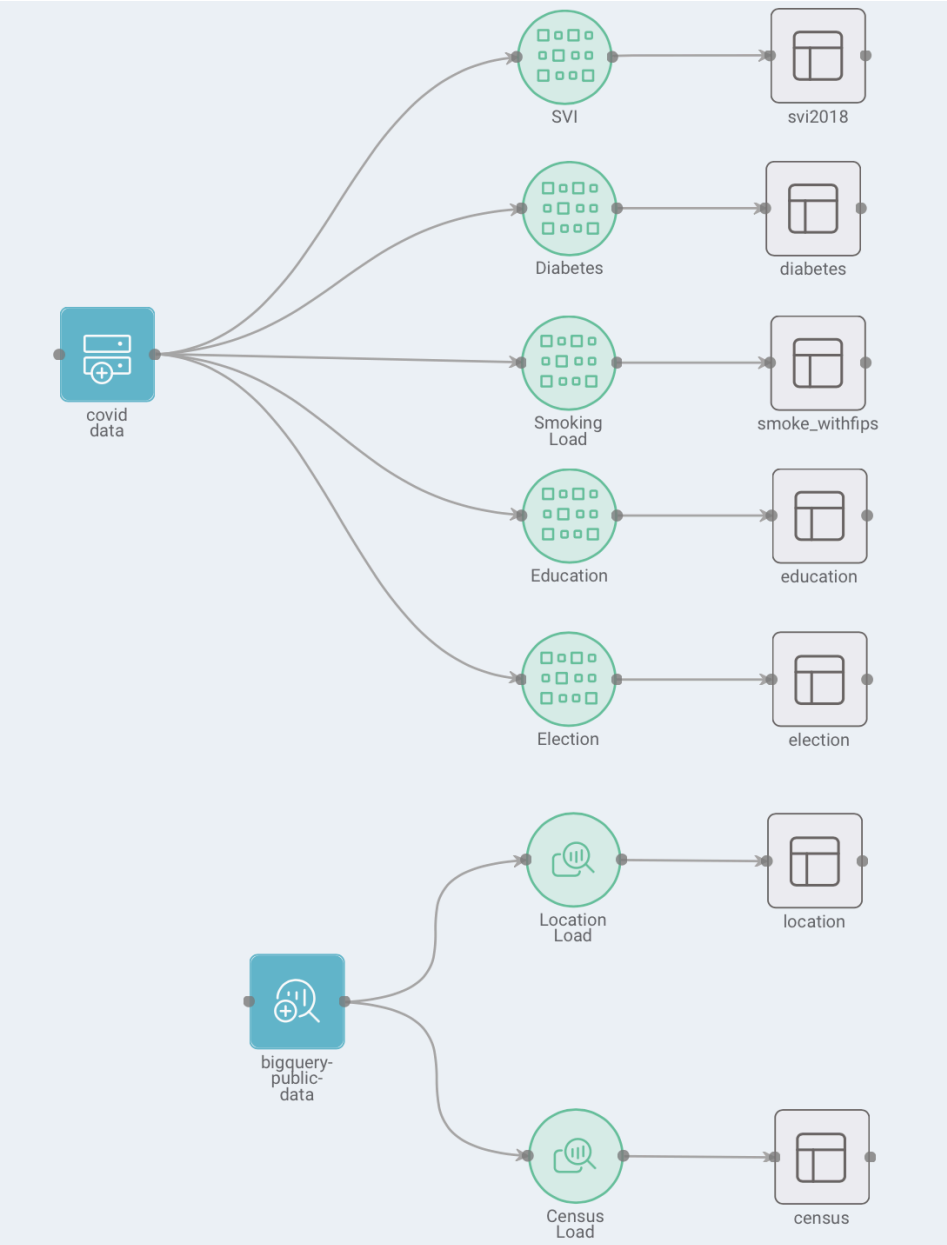
[6] bigquery-public-data.geo_us_boundaries.counties

[7] bigquery-public-data.census_bureau_acs.county_2018_5yr

on the pandemic, and I thought there may be a correlation between a county's political stance and the tendency for people in that county to practice social distancing (or the tendency for them to listen to President Trump). The location data served mainly to facilitate mapping in Data Studio, doing so based on the GPS coordinates of a county's geographic midpoint as opposed to county name or FIPS code, which it could not process. Lastly, I incorporated the BQ Census data because it contained more detailed data concerning breakdowns by age and race. My initial prediction was that counties with higher overall social vulnerability (especially with higher populations of older people and marginalized racial minorities) would be positively correlated with counties that have higher confirmed cases per capita, deaths per capita, and death rates.

Below are visualizations of my Syntasa workflows for my dynamic and static data respectively; I chose to exclude the dataset entitled "education" because this data was already contained within the "svi2018" table:


nyt_covid19.nyt_dataset
Spark Processor
nyt_dataset
Spark Processor-2
covid_dynamic_Output


covid data
SVI
svi2018
Diabetes
diabetes
Smoking Load
smoke_withfips
Education
education
Election
election
bigquery-public-data
Location Load
location
Census Load
census

Data Dictionary

Column	Definition	Table	Source
DATE ds	the date	covid_nyt	https://github.com/nytimes/covid-19-data
STATE	the state name	covid_nyt	https://github.com/nytimes/covid-19-data
COUNTY	the county name	covid_nyt	https://github.com/nytimes/covid-19-data
FIPS	the county FIPS code	covid_nyt	https://github.com/nytimes/covid-19-data
LAT	the latitude of a county's midpoint	counties	bigquery-public-data.geo_us_boundaries.counties
LON	the longitude of a county's midpoint	counties	bigquery-public-data.geo_us_boundaries.counties
E_TOTPOP	the estimated population of a county (2014-2018 ACS)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
M_TOTPOP	^ measure of error	svi2018	https://svi.cdc.gov/data-and-tools-download.html
AREA_SQMI	the tract area In square miles	svi2018	https://svi.cdc.gov/data-and-tools-download.html

CONFIRMED_CASES y	the number of laboratory confirmed COVID-19 cases only (or blank if not available)	covid_nyt	https://github.com/nytimes/covid-19-data
DEATHS	the total number of deaths from COVID-19 (including both confirmed and probable)	covid_nyt	https://github.com/nytimes/covid-19-data
CC_PER_CAPITA	CONFIRMED_CASES / E_TOTPOP		
D_PER_CAPITA	DEATHS / E_TOTPOP		
D_PER_CC	DEATHS / CONFIRMED_CASES		
CCT_1	confirmed cases yesterday		
CCT_2	confirmed cases two days ago		
CCT_3	confirmed cases three days ago		
CCT_7	confirmed cases a week ago		
CCT_14	confirmed cases two weeks ago		
CCT_31	confirmed cases 31 days ago		
DT_1	deaths yesterday		
DT_2	deaths two days ago		
DT_3	deaths three days ago		
DT_7	deaths a week ago		
DT_14	deaths two weeks ago		
DT_31	deaths 31 days ago		

MEDIAN_INCOME	the median income of a county	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
GINI_INDEX	a measure of the income inequality of a county	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
DIABETES_RATE	the estimated percentage of a county population with diabetes (0-100)	diabetes	https://gis.cdc.gov/grasp/diabetes/diabetesatlas.html
SMOKING_RATE	the estimated percentage of a county population that (consistently) smokes cigarettes (0-100)	smoke_withfips	https://dataverse.harvard.edu/file.xhtml?persistentId=doi:10.7910/DVN/VZ21KD/WWFBCX&version=1.0
COMBINED_DIABETES_SMOKING_RATES	DIABETES_RATE + SMOKING_RATE		
PERCENT_DEM	the percentage of a county population that voted democratic in the 2016 election (0-1)	election	https://github.com/tonmcg/US_County_Level_Election_Results_08-16/blob/master/2016_US_County_Level_Presidential_Results.csv
PERCENT_REP	the percentage of a county population that voted republican in the 2016 election (0-1)	election	https://github.com/tonmcg/US_County_Level_Election_Results_08-16/blob/master/2016_US_County_Level_Presidential_Results.csv

RPL_THEME1	the county's percentile ranking for the socioeconomic theme summary (0-1)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
RPL_THEME2	the county's percentile ranking for the household composition theme summary (0-1)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
RPL_THEME3	the county's percentile ranking for the minority status/language theme summary (0-1)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
RPL_THEME4	the county's percentile ranking for the housing type/transportation theme summary (0-1)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
RPL_THEMES	the county's overall percentile ranking for social vulnerability (0-1)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
EP_POV	the estimated percentage of people below poverty in a county (0-100)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
MP_POV	^ measure of error	svi2018	https://svi.cdc.gov/data-and-tools-download.html
EP_UNEMP	the estimated unemployment rate in a county (0-100)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
MP_UNEMP	^ measure of error	svi2018	https://svi.cdc.gov/data-and-tools-download.html

EP_PCI	the estimated per capita income in a county (2014-2018 ACS) (0-100)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
MP_PCI	^ measure of error	svi2018	https://svi.cdc.gov/data-and-tools-download.html
EP_NOHSDP	the estimated percentage of people with no high school diploma in a county (age 25+) (0-100)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
MP_NOHSDP	^ measure of error	svi2018	https://svi.cdc.gov/data-and-tools-download.html
EP_AGE65	the estimated percentage of persons aged 65 and older in a county (2014-2018 ACS) (0-100)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
MP_AGE65	^ measure of error	svi2018	https://svi.cdc.gov/data-and-tools-download.html
EP_AGE17	the estimated percentage of persons aged 17 and younger in a county (2014-2018 ACS) (0-100)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
MP_AGE17	^ measure of error	svi2018	https://svi.cdc.gov/data-and-tools-download.html

EP_DISABL	the estimated percentage of the civilian, noninstitutionalized population with a disability in a county (2014-2018 ACS) (0-100)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
MP_DISABL	^ measure of error	svi2018	https://svi.cdc.gov/data-and-tools-download.html
EP_MNRTY	the estimated minority percentage in a county (all persons except white, non-hispanic; 2014-2018 ACS) (0-100)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
MP_MNRTY	^ measure of error	svi2018	https://svi.cdc.gov/data-and-tools-download.html
EP_CROWD	the estimated percentage of occupied housing units with more people than rooms in a county (0-100)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
MP_CROWD	^ measure of error	svi2018	https://svi.cdc.gov/data-and-tools-download.html
EP_GROUPQ	the estimated percentage of persons in institutionalized group quarters in a county (2014-2018 ACS) (0-100)	svi2018	https://svi.cdc.gov/data-and-tools-download.html

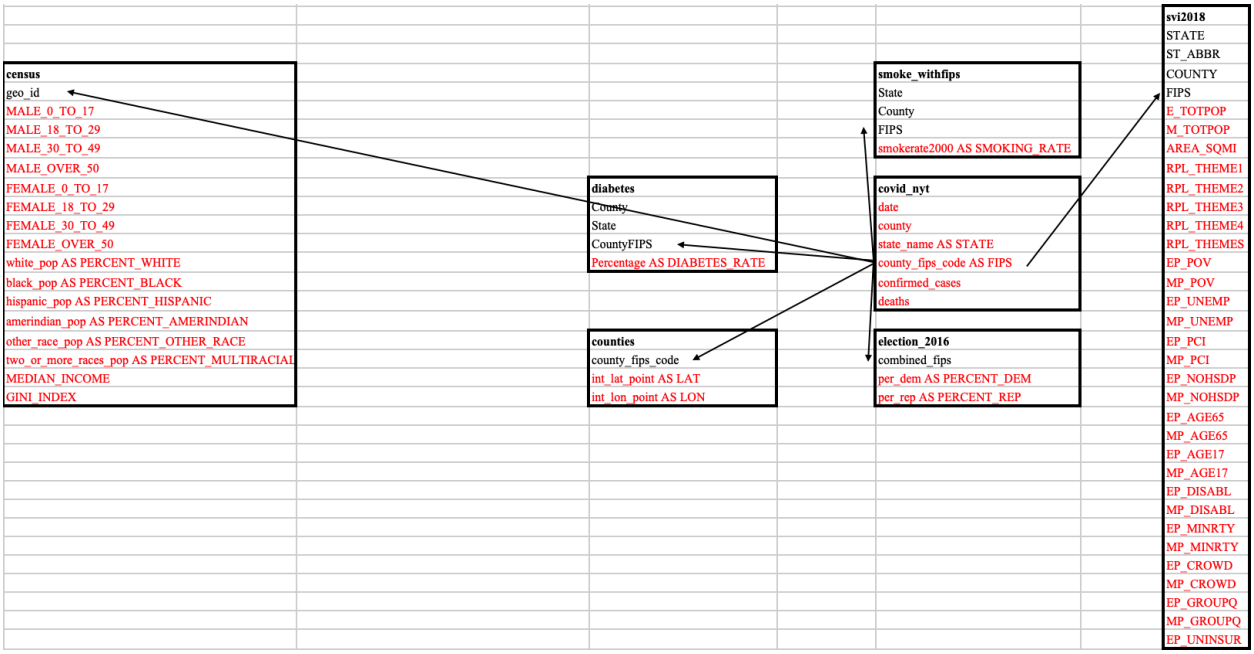
MP_GROUPQ	^ measure of error	svi2018	https://svi.cdc.gov/data-and-tools-download.html
EP_UNINSUR	the estimated uninsured population of noninstitutionalized civilians in a county (2014-2018 ACS) (0-100)	svi2018	https://svi.cdc.gov/data-and-tools-download.html
MP_UNINSUR	^ measure of error	svi2018	https://svi.cdc.gov/data-and-tools-download.html
PERCENT_HISPANIC	the estimated percentage of a county who identify as Hispanic (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
PERCENT_BLACK	the estimated percentage of a county who identify as black (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
PERCENT_ASIAN	the estimated percentage of a county who identify as Asian (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
PERCENT_AMERINDIAN	the estimated percentage of a county who identify as Native American (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
PERCENT_OTHER_RACE	the estimated percentage of a county who identify as another race (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr

PERCENT_MULTIRACIAL	the estimated percentage of a county who identify as multiracial (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
PERCENT_NONCITIZEN	the estimated percentage of a county who are not citizens (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
MALE_0_TO_17	the estimated percentage of a county who is male, age 0-17 (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
MALE_18_TO_29	the estimated percentage of a county who is male, age 18-29 (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
MALE_30_TO_49	the estimated percentage of a county who is male, age 30-49 (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
MALE_OVER_50	the estimated percentage of a county who is male, age 50+ (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
FEMALE_0_TO_17	the estimated percentage of a county who is female, age 0-17 (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
FEMALE_18_TO_29	the estimated percentage of a county who is female, age 18-29 (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr

FEMALE_30_TO_49	the estimated percentage of a county who is female, age 30-49 (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
FEMALE_OVER_50	the estimated percentage of a county who is female, age 50+ (0-1)	census	bigquery-public-data.census_bureau_acs.county_2018_5yr
error	actual cases minus predicted cases		
RMSE	the sum of the square of actual cases minus predicted cases divided by the sample size (the number of cases entries) of a county		

Data Preparation

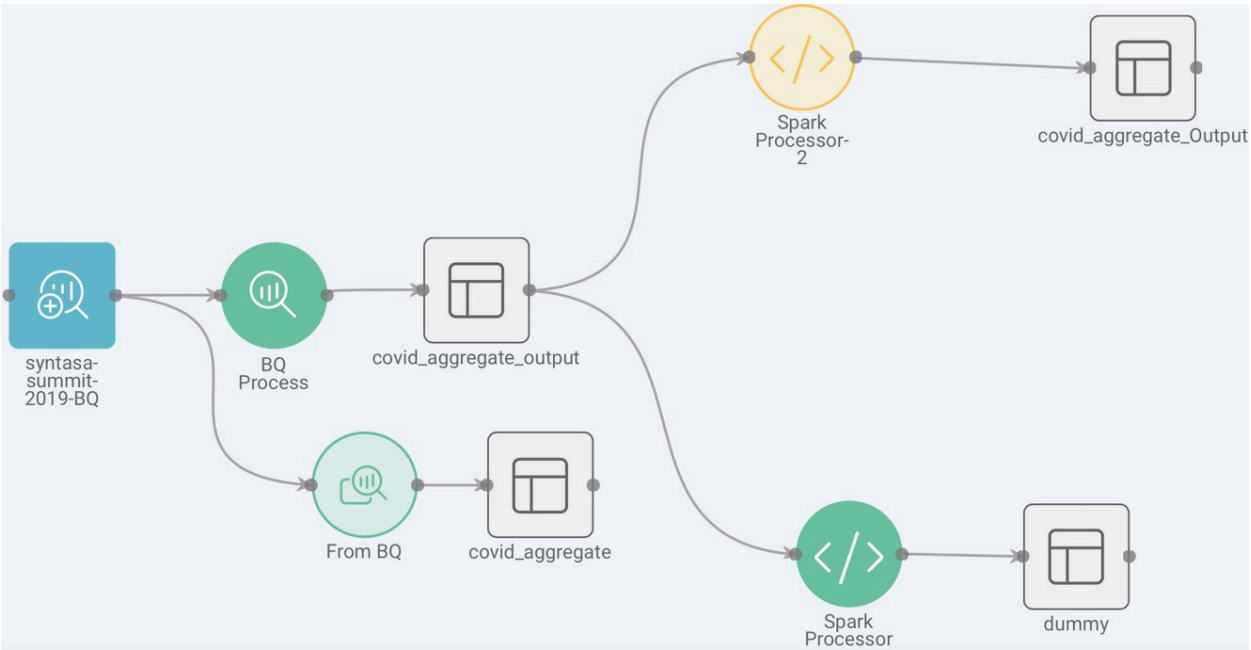
In order to bring all these individual datasets together, I conducted a series of joins in BQ's SQL environment. I recursively left-joined "covid_nyt" with the static datasets on FIPS codes, which all the datasets had in common. Joining on FIPS codes as opposed to county names was much more effective because FIPS codes are guaranteed to be unique, whereas county names can and do have duplicates. My aggregated dataset had a total of 3,061 distinct counties and FIPS codes. In terms of data manipulation, I had to convert all the datasets' FIPS columns to strings and remove leading zeroes and decimal points. I also rounded all float values to the hundredths place and omitted all counties with the name "Unknown," which previously contributed to outliers. Below is the star schema I used to plan out my joins (with items in red representing unique columns) as well as my SQL code (which contains many originally selected columns that were omitted because they were statistically insignificant):



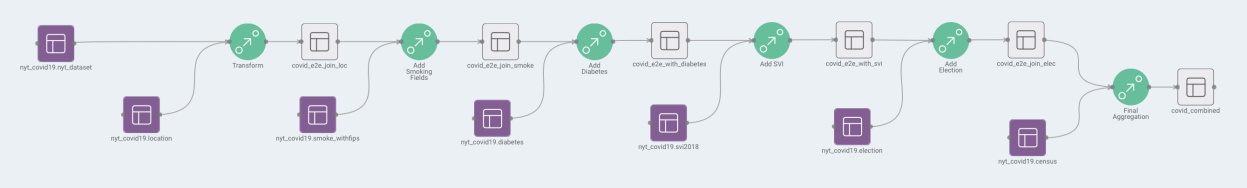
SELECT DATE(NYT._PARTITIONTIME) AS DATE, NYT.STATE AS STATE, NYT.COUNTY AS COUNTY, TRIM(REPLACE(CAST(NYT.FIPS AS STRING), '.0', '')) AS FIPS, INT_POINT_LAT AS LATITUDE, INT_POINT_LON AS LONGITUDE, E_TOTPOP, ROUND(AREA_SQMI, 2) AS AREA_SQMI, NYT.CASES AS CONFIRMED_CASES, DEATHS AS DEATHS, CAST(DIABETES.STRING_FIELD_3 AS FLOAT64) AS DIABETES_RATE, ROUND(SMOKERATE2000 * 100, 2) AS SMOKING_RATE, ROUND(PER_DEM, 2) AS PERCENT_DEM, ROUND(PER_GOP, 2) AS PERCENT_REP, RPL_THEME1, RPL_THEME2, RPL_THEME3, RPL_THEME4, RPL_THEMES, EP_POV, EP_UNEMP, EP_PCI, EP_NOHSDP, EP_AGE65, EP_AGE17, EP_DISABL, EP_MINRTY, EP_CROWD, EP_GROUPQ, EP_UNINSUR, HOUSEHOLDS, MALE_POP, FEMALE_POP, MEDIAN_AGE, MALE_UNDER_5, MALE_5_TO_9, MALE_10_TO_14, MALE_15_TO_17, MALE_18_TO_19, MALE_20, MALE_21, MALE_22_TO_24, MALE_25_TO_29, MALE_30_TO_34, MALE_35_TO_39, MALE_40_TO_44, MALE_45_TO_49, MALE_50_TO_54, MALE_55_TO_59, MALE_60_TO_61, MALE_62_TO_64, MALE_65_TO_66, MALE_67_TO_69, MALE_70_TO_74, MALE_75_TO_79, MALE_80_TO_84, MALE_85_AND_OVER, FEMALE_UNDER_5, FEMALE_5_TO_9, FEMALE_10_TO_14, FEMALE_15_TO_17, FEMALE_18_TO_19, FEMALE_20, FEMALE_21, FEMALE_22_TO_24, FEMALE_25_TO_29, FEMALE_30_TO_34, FEMALE_35_TO_39, FEMALE_40_TO_44, FEMALE_45_TO_49, FEMALE_50_TO_54, FEMALE_55_TO_59, FEMALE_60_TO_61, FEMALE_62_TO_64, FEMALE_65_TO_66, FEMALE_67_TO_69, FEMALE_70_TO_74, FEMALE_75_TO_79, FEMALE_80_TO_84, FEMALE_85_AND_OVER, WHITE_POP, POPULATION_1_YEAR_AND_OVER, POPULATION_3_YEARS_OVER, POP_5_YEARS_OVER, POP_15_AND_OVER, POP_16_OVER, POP_25_YEARS_OVER, POP_25_64, POP_NEVER_MARRIED, POP_NOW_MARRIED, POP_SEPARATED, POP_WIDOWED, POP_DIVORCED, NOT_US_CITIZEN_POP, BLACK_POP, ASIAN_POP, HISPANIC_POP, AMERINDIAN_POP, OTHER_RACE_POP, TWO_OR_MORE_RACES_POP, HISPANIC_ANY_RACE, NOT_HISPANIC_POP, ASIAN_MALE_45_54, ASIAN_MALE_55_64, BLACK_MALE_45_54, BLACK_MALE_55_64, HISPANIC_MALE_45_54, HISPANIC_MALE_55_64, WHITE_MALE_45_54, WHITE_MALE_55_64, MEDIAN_INCOME, INCOME_PER_CAPITA, INCOME_LESS_10000, INCOME_10000_14999, INCOME_15000_19999, INCOME_20000_24999, INCOME_25000_29999, INCOME_30000_34999, INCOME_35000_39999, INCOME_40000_44999, INCOME_45000_49999, INCOME_50000_59999, INCOME_60000_74999, INCOME_75000_99999, INCOME_100000_124999, INCOME_125000_149999, INCOME_150000_199999, INCOME_200000_OR_MORE, POP_DETERMINED_POVERTY_STATUS, POVERTY, GINI_INDEX, HOUSING_UNITS, RENTER_OCCUPIED_HOUSING_UNITS_PAYING_CASH_MEDIAN_GROSS_RENT, OWNER_OCCUPIED_HOUSING_UNITS_LOWER_VALUE_QUARTILE, OWNER_OCCUPIED_HOUSING_UNITS_MEDIAN_VALUE, OWNER_OCCUPIED_HOUSING_UNITS_UPPER_VALUE_QUARTILE, OCCUPIED_HOUSING_UNITS, HOUSING_UNITS_RENTER_OCCUPIED, VACANT_HOUSING_UNITS, VACANT_HOUSING_UNITS_FOR_RENT, VACANT_HOUSING_UNITS_FOR_SALE, DWELLINGS_1_UNITS_DETACHED, DWELLINGS_1_UNITS_ATTACHED, DWELLINGS_2_UNITS, DWELLINGS_3_TO_4_UNITS, DWELLINGS_5_TO_9_UNITS, DWELLINGS_10_TO_19_UNITS, DWELLINGS_20_TO_49_UNITS, DWELLINGS_50_OR_MORE_UNITS, MOBILE_HOMES, HOUSING_BUILT_2005_OR_LATER, HOUSING_BUILT_2000_TO_2004, HOUSING_BUILT_1939_OR_EARLIER, MEDIAN_YEAR_STRUCTURE_BUILT, MARRIED_HOUSEHOLDS, NONFAMILY_HOUSEHOLDS, FAMILY_HOUSEHOLDS, HOUSEHOLDS_PUBLIC_ASST_OR_FOOD_STAMPS, MALE_MALE_HOUSEHOLDS, FEMALE_FEMALE_HOUSEHOLDS, CHILDREN, CHILDREN_IN_SINGLE_FEMALE_HH, MEDIAN_RENT, PERCENT_INCOME_SPENT_ON_RENT, RENT_BURDEN_NOT_COMPUTED, RENT_OVER_50_PERCENT, RENT_40_TO_50_PERCENT, RENT_35_TO_40_PERCENT, RENT_30_TO_35_PERCENT, RENT_25_TO_30_PERCENT, RENT_20_TO_25_PERCENT, RENT_15_TO_20_PERCENT, RENT_10_TO_15_PERCENT, RENT_UNDER_10_PERCENT, OWNER_OCCUPIED_HOUSING_UNITS, MILLION_DOLLAR_HOUSING_UNITS, MORTGAGED_HOUSING_UNITS,

```
DIFFERENT_HOUSE_YEAR_AGO_DIFFERENT_CITY, DIFFERENT_HOUSE_YEAR_AGO_SAME_CITY,
FAMILIES_WITH_YOUNG_CHILDREN, TWO_PARENT_FAMILIES_WITH_YOUNG_CHILDREN,
TWO_PARENTS_IN_LABOR_FORCE_FAMILIES_WITH_YOUNG_CHILDREN,
TWO_PARENTS_FATHER_IN_LABOR_FORCE_FAMILIES_WITH_YOUNG_CHILDREN,
TWO_PARENTS_MOTHER_IN_LABOR_FORCE_FAMILIES_WITH_YOUNG_CHILDREN,
TWO_PARENTS_NOT_IN_LABOR_FORCE_FAMILIES_WITH_YOUNG_CHILDREN,
ONE_PARENT_FAMILIES_WITH_YOUNG_CHILDREN, FATHER_ONE_PARENT_FAMILIES_WITH_YOUNG_CHILDREN,
FATHER_IN_LABOR_FORCE_ONE_PARENT_FAMILIES_WITH_YOUNG_CHILDREN, COMMUTE_LESS_10_MINS,
COMMUTE_10_14_MINS, COMMUTE_15_19_MINS, COMMUTE_20_24_MINS, COMMUTE_25_29_MINS,
COMMUTE_30_34_MINS, COMMUTE_35_44_MINS, COMMUTE_60_MORE_MINS, COMMUTE_45_59_MINS,
COMMUTERS_16_OVER, WALKED_TO_WORK, WORKED_AT_HOME, NO_CAR, NO_CARS, ONE_CAR, TWO_CARS,
THREE_CARS, FOUR_MORE_CARS, AGGREGATE_TRAVEL_TIME_TO_WORK,
COMMUTERS_BY_PUBLIC_TRANSPORTATION, COMMUTERS_BY_BUS, COMMUTERS_BY_CAR_TRUCK_VAN,
COMMUTERS_BY_CARPOOL, COMMUTERS_BY_SUBWAY_OR_ELEVATED, COMMUTERS_DROVE_ALONE,
GROUP_QUARTERS, ASSOCIATES_DEGREE, BACHELORS_DEGREE, HIGH_SCHOOL_DIPLOMA,
LESS_ONE_YEAR_COLLEGE, MASTERS_DEGREE, ONE_YEAR_MORE_COLLEGE,
LESS_THAN_HIGH_SCHOOL_GRADUATE, HIGH_SCHOOL_INCLUDING_GED, BACHELORS_DEGREE_2,
BACHELORS_DEGREE_OR_HIGHER_25_64, GRADUATE_PROFESSIONAL_DEGREE,
SOME_COLLEGE_AND_ASSOCIATES_DEGREE, MALE_45_64_ASSOCIATES_DEGREE,
MALE_45_64_BACHELORS_DEGREE, MALE_45_64_GRADUATE_DEGREE, MALE_45_64_LESS_THAN_9_GRADE,
MALE_45_64_GRADE_9_12, MALE_45_64_HIGH_SCHOOL, MALE_45_64_SOME_COLLEGE, MALE_45_TO_64,
EMPLOYED_POP, UNEMPLOYED_POP, POP_IN_LABOR_FORCE, NOT_IN_LABOR_FORCE, WORKERS_16_AND_OVER,
ARMED_FORCES, CIVILIAN_LABOR_FORCE, EMPLOYED_AGRICULTURE_FORESTRY_FISHING_HUNTING_MINING,
EMPLOYED_ARTS_ENTERTAINMENT_RECREATION_ACCOMMODATION_FOOD, EMPLOYED_CONSTRUCTION,
EMPLOYED_EDUCATION_HEALTH_SOCIAL, EMPLOYED_FINANCE_INSURANCE_REAL_ESTATE,
EMPLOYED_INFORMATION, EMPLOYED_MANUFACTURING, EMPLOYED_OTHER_SERVICES_NOT_PUBLIC_ADMIN,
EMPLOYED_PUBLIC_ADMINISTRATION, EMPLOYED_RETAIL_TRADE,
EMPLOYED_SCIENCE_MANAGEMENT_ADMIN_WASTE, EMPLOYED_TRANSPORTATION_WAREHOUSING_UTILITIES,
EMPLOYED_WHOLESALE_TRADE, OCCUPATION_MANAGEMENT_ARTS,
OCCUPATION_NATURAL_RESOURCES_CONSTRUCTION_MAINTENANCE,
OCCUPATION_PRODUCTION_TRANSPORTATION_MATERIAL, OCCUPATION_SALES_OFFICE, OCCUPATION_SERVICES,
MANAGEMENT_BUSINESS_SCI_ARTS_EMPLOYED, SALES_OFFICE_EMPLOYED, IN_GRADES_1_TO_4,
IN_GRADES_5_TO_8, IN_GRADES_9_TO_12, IN_SCHOOL, IN_UNDERGRAD_COLLEGE,
ROUND((NYT.CASES / E_TOTPOP) * 1000, 2) AS CC_PER_1000, ROUND((DEATHS / E_TOTPOP) * 1000, 2) AS D_PER_1000,
ROUND(DEATHS / NYT.CASES, 2) AS D_PER_CC,
LAG(NYT.CASES,1) OVER (PARTITION BY NYT.COUNTY ORDER BY DATE(NYT._PARTITIONTIME) ASC) AS CT_1,
LAG(NYT.CASES,2) OVER (PARTITION BY NYT.COUNTY ORDER BY DATE(NYT._PARTITIONTIME) ASC) AS CT_2,
LAG(NYT.CASES,3) OVER (PARTITION BY NYT.COUNTY ORDER BY DATE(NYT._PARTITIONTIME) ASC) AS CT_3,
LAG(NYT.CASES,7) OVER (PARTITION BY NYT.COUNTY ORDER BY DATE(NYT._PARTITIONTIME) ASC) AS CT_7,
LAG(DEATHS,1) OVER (PARTITION BY NYT.COUNTY ORDER BY DATE(NYT._PARTITIONTIME) ASC) AS DT_1,
LAG(DEATHS,2) OVER (PARTITION BY NYT.COUNTY ORDER BY DATE(NYT._PARTITIONTIME) ASC) AS DT_2,
LAG(DEATHS,3) OVER (PARTITION BY NYT.COUNTY ORDER BY DATE(NYT._PARTITIONTIME) ASC) AS DT_3,
LAG(DEATHS,7) OVER (PARTITION BY NYT.COUNTY ORDER BY DATE(NYT._PARTITIONTIME) ASC) AS DT_7
FROM `SYNTASA-DEMO-40.PROD_NYT_COVID19.NYT_DATASET` AS NYT

LEFT JOIN `BIGQUERY-PUBLIC-DATA.GEO_US_BOUNDARIES.COUNTIES` AS LOCATION
ON CAST(NYT.FIPS AS STRING) = CAST(CAST(LOCATION.COUNTY_FIPS_CODE AS INT64) AS STRING)

LEFT JOIN `SYNTASA-DEMO-40.PLAYGROUND.SMOKING` AS SMOKING
ON TRIM(REPLACE(CAST(NYT.FIPS AS STRING), '.0', '')) = CAST(SMOKING.FIPS AS STRING)
AND
TRIM(REPLACE(LOWER(NYT.COUNTY), ' CITY', '')) = TRIM(REPLACE(LOWER(SMOKING.COUNTY), ' COUNTY', ''))

LEFT JOIN `SYNTASA-DEMO-40.PLAYGROUND.DIABETES` AS DIABETES
ON CAST(NYT.FIPS AS STRING) = REPLACE(LTRIM(REPLACE(DIABETES.STRING_FIELD_2, '0', ' ')), ' ', '0')

LEFT JOIN `SYNTASA-DEMO-40.PLAYGROUND.SVI` AS SVI
ON CAST(NYT.FIPS AS STRING) = CAST(SVI.FIPS AS STRING)

LEFT JOIN `SYNTASA-DEMO-40.PLAYGROUND.ELECTION_2016` AS ELECT
ON CAST(NYT.FIPS AS STRING) = CAST(ELECT.COMBINED_FIPS AS STRING)

LEFT JOIN `BIGQUERY-PUBLIC-DATA.CENSUS_BUREAU_ACS.COUNTY_2018_5YR` AS CENSUS
ON CAST(NYT.FIPS AS STRING) = REPLACE(LTRIM(REPLACE(CENSUS.GEO_ID, '0', ' ')), ' ', '0')

WHERE NYT.COUNTY != 'UNKNOWN'
-- ORDER BY DATE DESC, STATE, NYT.COUNTY
```

I then used Syntasa to process my SQL code in BQ and generate an aggregated dataset ("covid_aggregate_Output") back into BQ. I scheduled this app to run everyday in order to update the dynamic elements of the dataset. Here is my Syntasa workflow:

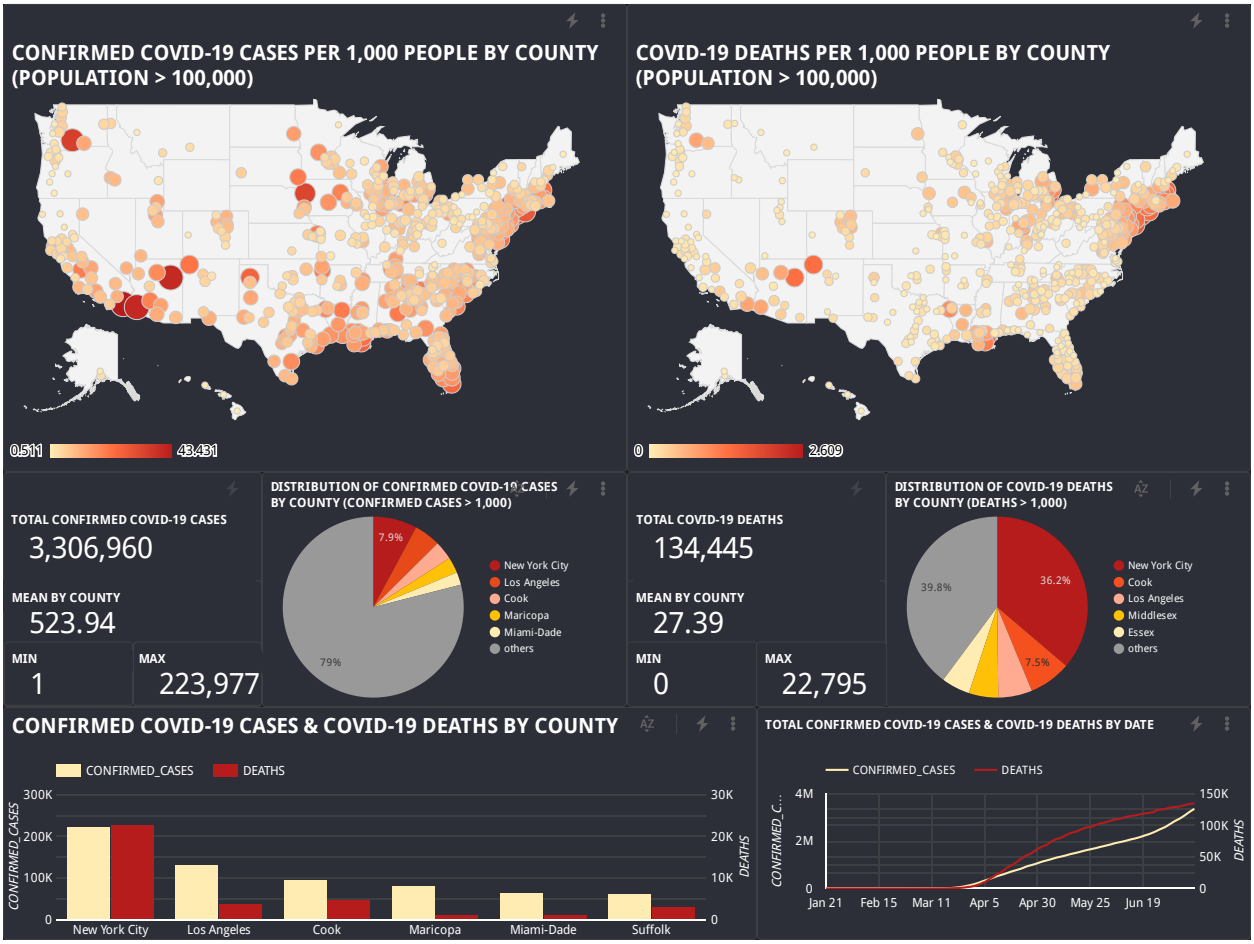


I created a separate workflow in Syntasa to conduct the joins across datasets in a more presentable fashion. Each "Transform" operation conducts a left join between the dynamic dataset and a static dataset on FIPS codes. The following workflow more cleanly represents the prior SQL code:



Data Visualization

Once I had a single finalized dataset, I imported the dataset directly from BQ and used Data Studio to create an interactive visual dashboard to represent the data. Here is the dashboard I generated:



	STATE	COUNTY	CONFIRMED_CASES_PER_1000_PEOPLE ❷	DEATHS_PER_1000_PEOPLE ❶	E_TOTP...	RPL_THEM...	COMBINED_DIABETES_SMOKING_RATES
1.	Georgia	Hancock	26.71	3.87	8,535	0.8	39.4
2.	Georgia	Randolph	29.91	3.53	7,087	0.97	39.7
3.	Georgia	Terrell	27.99	3.16	8,859	0.96	38.8
4.	Georgia	Early	29.57	3	10,348	0.92	39.8
5.	Virginia	Emporia city	27.13	2.79	5,381	0.96	null
6.	Mississippi	Neshoba	35.61	2.62	29,376	0.97	43.6
7.	New Jersey	Essex	24.27	2.61	793,555	0.84	33.5
8.	New Mexico	McKinley	49.88	2.61	72,849	0.99	40.5
9.	New Jersey	Union	30.17	2.43	553,066	0.62	29.3
...	New Jersey	Passaic	34.17	2.42	504,041	0.78	32.1
...	Virginia	Galax city	43.39	2.41	6,638	0.98	null
...	Alabama	Lowndes	47.67	2.34	10,236	0.92	43.7
...	Virginia	Northampton	22.92	2.34	11,957	0.68	37.3
...	Mississippi	Holmes	32.92	2.27	18,075	0.98	41.6
...	Georgia	Turner	23.74	2.26	7,962	0.98	38.5
...	New Jersey	Hudson	28.86	2.22	668,631	0.64	33
...	New Jersey	Bergen	21.66	2.18	929,999	0.24	27.4
...	Louisiana	Bienville	19.39	2.05	13,668	0.94	null
...	Louisiana	St. John the Baptist	24.88	2	43,446	0.75	null
...	New York	Nassau	31.19	1.99	1,356,564	0.24	28.6
...	Mississippi	Leflore	17.92	1.88	29,804	0.97	42.2
...	Nebraska	Dakota	89.04	1.87	20,317	0.86	39.1
...	Texas	Crane	13.23	1.86	4,839	0.47	35
...	Georgia	Mitchell	21.89	1.78	22,432	0.99	41.3
...	Alabama	Tallapoosa	15.75	1.72	40,636	0.89	39.8
...	Georgia	Upson	14.8	1.72	26,216	0.85	40
...	Georgia	Dougherty	23.72	1.71	91,049	0.95	37
...	Georgia	Wilcox	15.26	1.7	8,846	0.82	37.4
...	Louisiana	East Feliciana	18.21	1.69	19,499	0.82	null

I mapped confirmed COVID-19 cases and COVID-19 deaths per 1,000 people by county. I chose to map a per capita metric because it more accurately displayed the spread and impact of the disease by county. I only selected counties with populations higher than 100,000 people to make the visualization less cluttered. The mapping was done based off the latitude and longitude of each county's geographic midpoint.

Regarding confirmed cases, there seems to be a wide range of counties and areas that are severely affected, particularly around Los Angeles, Seattle, New York, and New Jersey. On the other hand, deaths seem to be extremely concentrated around New York and New Jersey and are significantly less everywhere else.

The pie charts display the distributions of confirmed cases and deaths respectively by county, comparing the impact of the five most affected counties with the impact of all other counties.

Roughly 20% of confirmed cases come from five counties, a percentage that has been steadily decreasing as the virus has spread geographically. In comparison, over 60% of deaths come from five counties, confirming that there is a greater right skew for deaths than for confirmed cases. New York City is responsible for 7.7% of confirmed cases but a shocking 36.2% of deaths, almost five times as impactful regarding deaths. New York City is clearly a special case in which the death rate is far above the national average (possibly because they were unprepared for and overwhelmed by such a rapid spike in cases in March and April).

Total confirmed cases in the US are increasing at a faster rate than ever, but while cases and deaths previously mirrored each other's growth, deaths are not climbing nearly as fast as cases. Deaths are still increasing but form a concave function over time, increasing at a decreasing rate; however, cases form a convex function, increasing at an increasing rate. This discrepancy is possibly a result of hospitals being better prepared to handle patients (previously they were somewhat caught by surprise). Also, the increased availability of testing could be contributing to this phenomenon (causing people to catch the virus early and seek prompt medical attention).

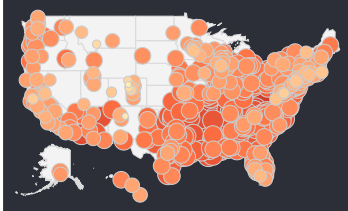
People seem to be more prepared to respond to the sickness than they are to avoid it in the first place.

The table sorts counties by highest deaths per population. 19 of the top 20 counties with the highest deaths per 1,000 people are above the 50th percentile for overall social vulnerability; regarding the top 100 counties, 77 have an above-average social vulnerability. There is a clear positive correlation between COVID-19 deaths and social vulnerability. I created COMBINED_DIABETES_SMOKING_RATES an informal metric to observe the combined effects of diabetes and smoking; however, in order to be valid, diabetes and smoking rates would have to be exclusive from one another, which is not the case.

Data Analysis

I also used Data Studio to conduct exploratory data analysis, looking at correlations between static variables and the effects of politics on the pandemic:

DIABETES RATE BY COUNTY

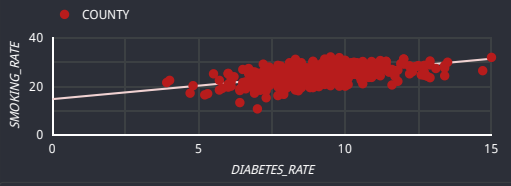


MEAN DIABETES RATE (%)
9.85

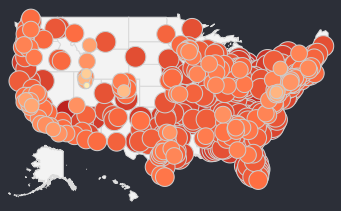
MEAN SMOKING RATE (%)
26.75

MIN	MAX	MIN	MAX
3.4	18.4	10.6	38.3

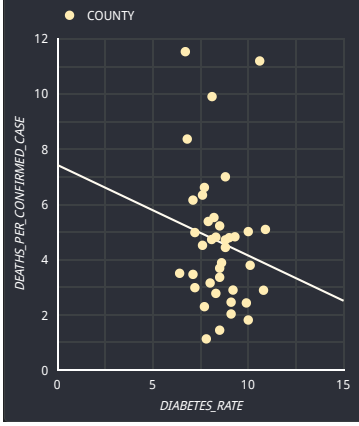
SMOKING RATE V. DIABETES RATE BY COUNTY



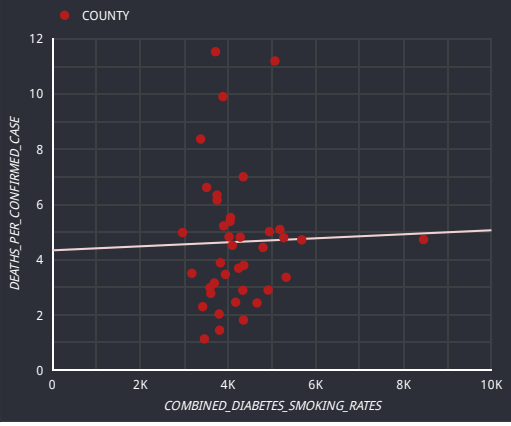
SMOKING RATE BY COUNTY



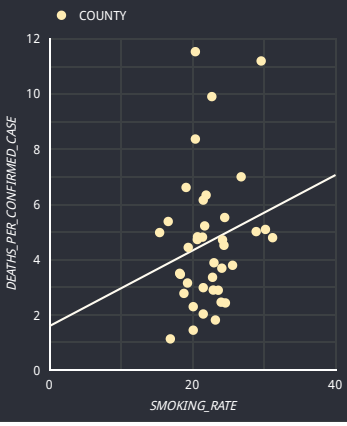
DEATHS PER CONFIRMED COVID-19 CASE V. DIABETES RATE BY COUNTY

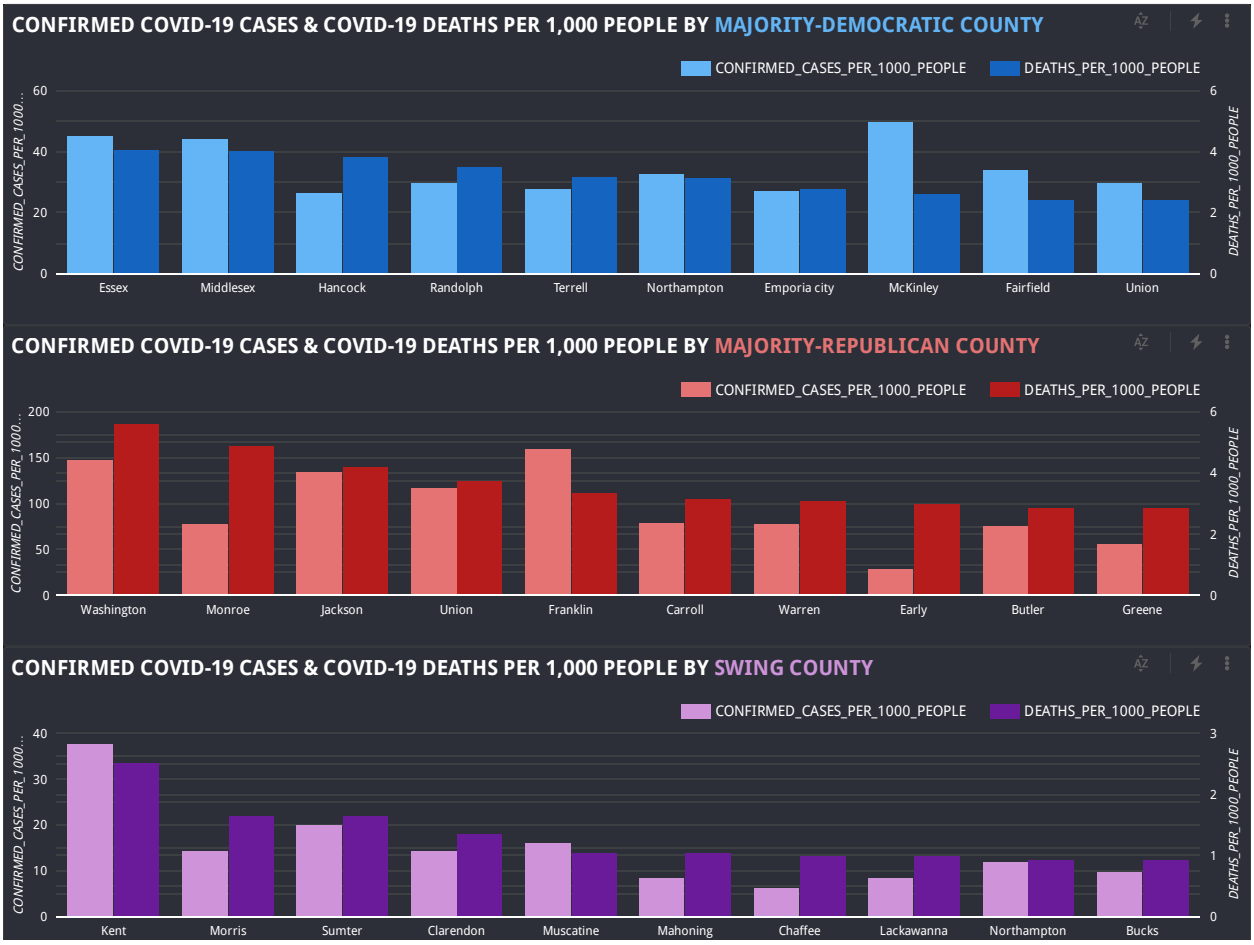


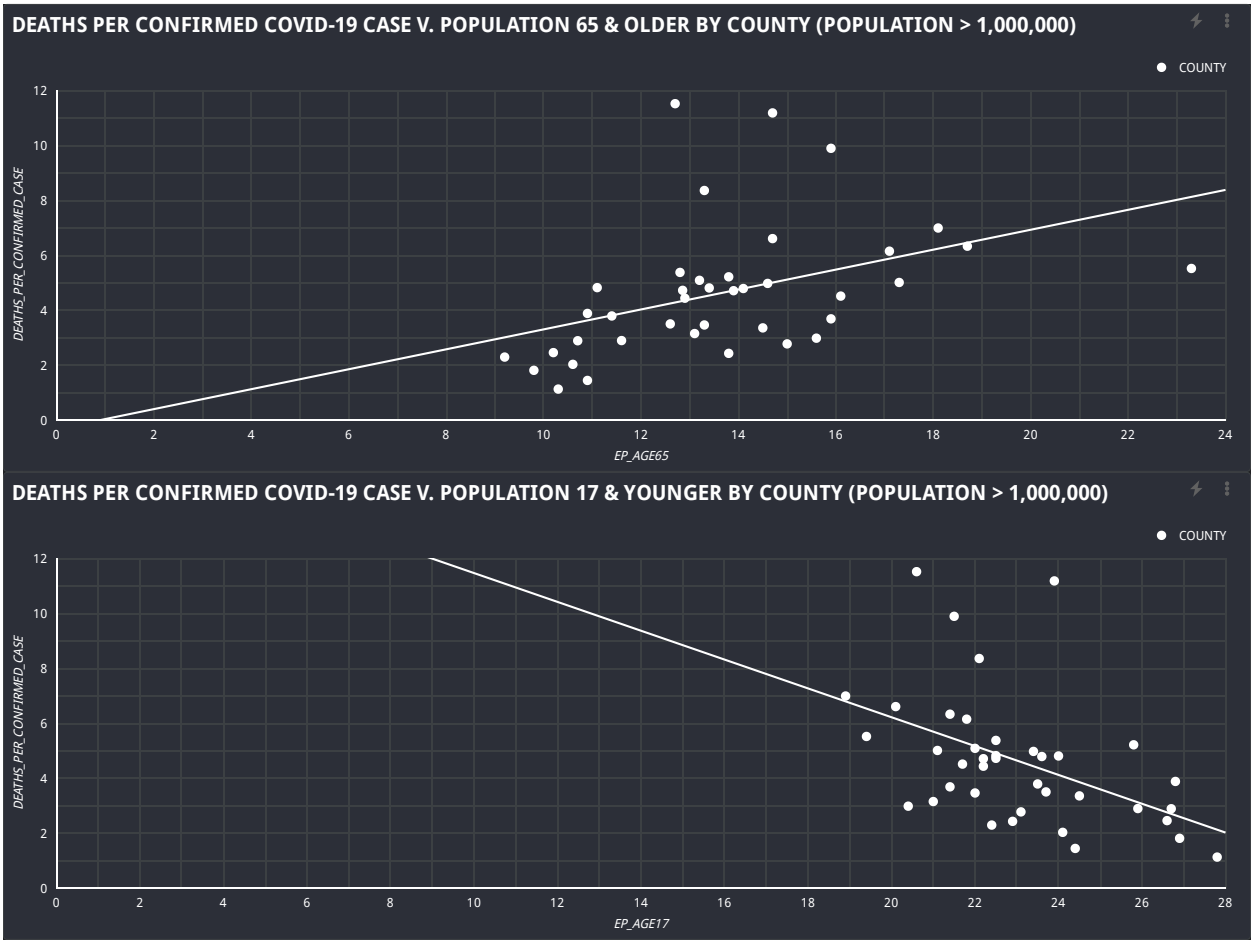
DEATHS PER CONFIRMED COVID-19 CASE V. COMBINED DIABETES-SMOKING RATES BY COUNTY

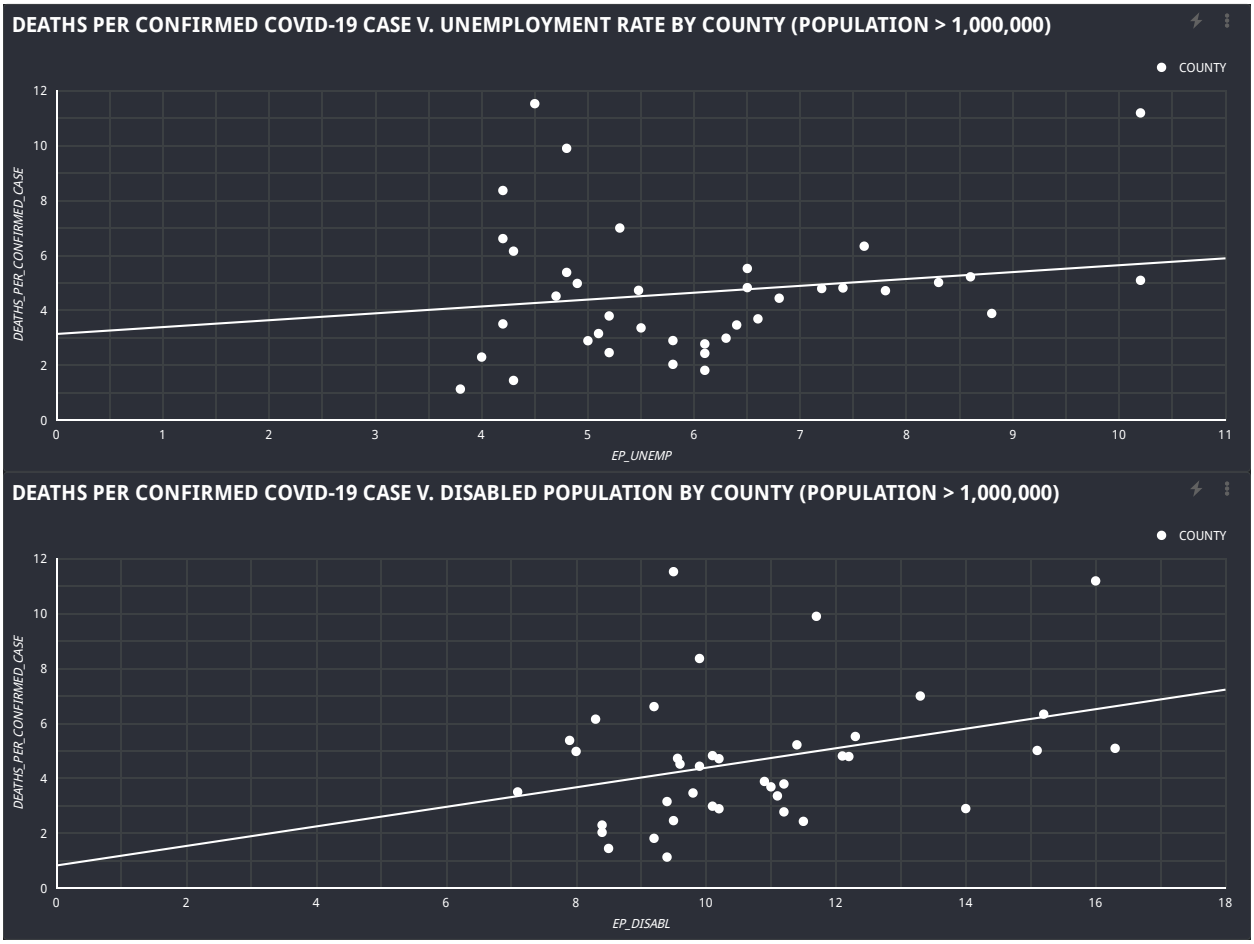


DEATHS PER CONFIRMED COVID-19 CASE V. SMOKING RATE BY COUNTY




CONFIRMED COVID-19 CASES & COVID-19 DEATHS PER 1,000 PEOPLE BY MAJORITY-DEMOCRATIC COUNTY
CONFIRMED_CASES_PER_1000_PEOPLE
DEATHS_PER_1000_PEOPLE
CONFIRMED_CASES_PER_1000...
DEATHS_PER_1000_PEOPLE
0
20
40
60
0
2
4
6
Essex
Middlesex
Hancock
Randolph
Terrell
Northampton
Emporia city
McKinley
Fairfield
Union
CONFIRMED COVID-19 CASES & COVID-19 DEATHS PER 1,000 PEOPLE BY MAJORITY-REPUBLICAN COUNTY
CONFIRMED_CASES_PER_1000_PEOPLE
DEATHS_PER_1000_PEOPLE
CONFIRMED_CASES_PER_1000...
DEATHS_PER_1000_PEOPLE
0
50
100
150
200
0
2
4
6
Washington
Monroe
Jackson
Union
Franklin
Carroll
Warren
Early
Butler
Greene
CONFIRMED COVID-19 CASES & COVID-19 DEATHS PER 1,000 PEOPLE BY SWING COUNTY
CONFIRMED_CASES_PER_1000_PEOPLE
DEATHS_PER_1000_PEOPLE
CONFIRMED_CASES_PER_1000...
DEATHS_PER_1000_PEOPLE
0
10
20
30
40
0
1
2
3
Kent
Morris
Sumter
Clarendon
Muscatine
Mahoning
Chaffee
Lackawanna
Northampton
Bucks


DEATHS PER CONFIRMED COVID-19 CASE V. POPULATION 65 & OLDER BY COUNTY (POPULATION > 1,000,000)
COUNTY
DEATHS_PER_CONFIRMED_CASE
EP_AGE65
0
2
4
6
8
10
12
14
16
18
20
22
24
DEATHS PER CONFIRMED COVID-19 CASE V. POPULATION 17 & YOUNGER BY COUNTY (POPULATION > 1,000,000)
COUNTY
DEATHS_PER_CONFIRMED_CASE
EP_AGE17
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28


DEATHS PER CONFIRMED COVID-19 CASE V. UNEMPLOYMENT RATE BY COUNTY (POPULATION > 1,000,000)
COUNTY
DEATHS_PER_CONFIRMED_CASE
EP_UNEMP
0
1
2
3
4
5
6
7
8
9
10
11
12
DEATHS PER CONFIRMED COVID-19 CASE V. DISABLED POPULATION BY COUNTY (POPULATION > 1,000,000)
COUNTY
DEATHS_PER_CONFIRMED_CASE
EP_DISABL
0
2
4
6
8
10
12
14
16
18

There is a high correlation between diabetes and smoking rates, though the correlations between diabetes and smoking rates with confirmed cases and deaths are less convincing.

From a political standpoint, counties with the highest confirmed cases per 1,000 people that are majority-democratic have less confirmed cases and deaths on average than majority-republican counties (possibly the effect of Trump supporters' ignorance of social distancing policies and other protective measures?); swing counties have significantly less confirmed cases and deaths per 1,000 people than both majority-democratic and -republican counties (though there is also a much smaller sample size of swing counties, or counties with neither a democratic nor a republican majority).

There is a convincing correlation between high elder populations and high deaths per confirmed case and conversely, high younger populations and low deaths per confirmed case in counties with more than 1,000,000 people. The correlations between unemployment and disabilities and death rates are less obvious, and there is not a clear pattern in the data.

I conducted further data analysis using Jupyter Notebook, looking more closely at correlations:

I imported my project into Jupyter from BQ.

In [396]:
```
# Packages

from datetime import datetime
from fbprophet import Prophet
from google.cloud import bigquery
from google.cloud import bigquery_storage_v1beta1
from google.oauth2 import service_account
from patsy import dmatrices
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import MinMaxScaler

credentials = service_account.Credentials.from_service_account_file("/
Users/cyrus.hatam/Desktop/Syntasa-Demo-40-9757e0bf8a9f.json")
project_id = "syntasa-demo-40"
client = bigquery.Client(credentials = credentials, project = project_
id)

import math
import matplotlib.pyplot as plt
import pandas as pd
import sklearn as sk
import statsmodels.formula.api as smf
```

I chose to take the log of CONFIRMED_CASES, DEATHS, E_TOTPOP, and MEDIAN_INCOME because they each had significant right skews. Since OLS regression assumes a normal distribution, I took the log of these variables in order to fit them to such a model more accurately. The following is my SQL code from BQ:

In [433]:
```
# BQ Dataset

sql_query = "SELECT DATE, COUNTY, FIPS, ROUND(LN(CONFIRMED_CASES)) AS
CONFIRMED_CASES, ROUND(LN(DEATHS)) AS DEATHS, CONFIRMED_CASES / E_TOTP
OP AS CC_PER_CAPITA, DEATHS / E_TOTPOP AS D_PER_CAPITA, DEATHS / CONFI
RMED_CASES AS D_PER_CC, ROUND(LN(E_TOTPOP)) AS E_TOTPOP, DIABETES_RATE
, SMOKING_RATE, ROUND(LN(MEDIAN_INCOME)) AS MEDIAN_INCOME, PERCENT_DEM
, PERCENT_REP, RPL_THEMES, LAG(ROUND(LN(CONFIRMED_CASES)),1) OVER (PAR
TITION BY COUNTY ORDER BY DATE ASC) AS CCT_1, LAG(ROUND(LN(CONFIRMED_C
ASES)),2) OVER (PARTITION BY COUNTY ORDER BY DATE ASC) AS CCT_2, LAG(R
OUND(LN(CONFIRMED_CASES)),3) OVER (PARTITION BY COUNTY ORDER BY DATE A
SC) AS CCT_3, LAG(ROUND(LN(CONFIRMED_CASES)),7) OVER (PARTITION BY COU
NTY ORDER BY DATE) AS CCT_7, LAG(ROUND(LN(CONFIRMED_CASES)),14) OVER (
PARTITION BY COUNTY ORDER BY DATE ASC) AS CCT_14, LAG(ROUND(LN(CONFIRM
ED_CASES)),21) OVER (PARTITION BY COUNTY ORDER BY DATE ASC) AS CCT_21,
```

```
LAG(ROUND(LN(CONFIRMED_CASES)),31) OVER (PARTITION BY COUNTY ORDER BY
DATE ASC) AS CCT_31, LAG(DEATHS,1) OVER (PARTITION BY COUNTY ORDER BY
DATE ASC) AS DT_1, LAG(DEATHS,2) OVER (PARTITION BY COUNTY ORDER BY DA
TE ASC) AS DT_2, LAG(DEATHS,3) OVER (PARTITION BY COUNTY ORDER BY DATE
ASC) AS DT_3, LAG(DEATHS,7) OVER (PARTITION BY COUNTY ORDER BY DATE AS
C) AS DT_7, LAG(DEATHS,14) OVER (PARTITION BY COUNTY ORDER BY DATE ASC
) AS DT_14, LAG(DEATHS,21) OVER (PARTITION BY COUNTY ORDER BY DATE ASC
) AS DT_21, LAG(DEATHS,31) OVER (PARTITION BY COUNTY ORDER BY DATE ASC
) AS DT_31, ((MALE_UNDER_5 / E_TOTPOP) + (MALE_5_TO_9 / E_TOTPOP) + (M
ALE_10_TO_14 / E_TOTPOP) + (MALE_15_TO_17 / E_TOTPOP)) AS MALE_0_TO_17
, ((MALE_18_TO_19 / E_TOTPOP) + (MALE_20 / E_TOTPOP) + (MALE_21 / E_TO
TPOP) + (MALE_22_TO_24 / E_TOTPOP) + (MALE_25_TO_29 / E_TOTPOP)) AS MA
LE_18_TO_29, (MALE_30_TO_34 / E_TOTPOP) + (MALE_35_TO_39 / E_TOTPOP) +
(MALE_40_TO_44 / E_TOTPOP) + (MALE_45_TO_49 / E_TOTPOP) AS MALE_30_TO_
49, (MALE_50_TO_54 / E_TOTPOP) + (MALE_55_TO_59 / E_TOTPOP) + (MALE_60
_TO_61 / E_TOTPOP) + (MALE_62_TO_64 / E_TOTPOP) + (MALE_65_TO_66 / E_T
OTPOP) + (MALE_67_TO_69 / E_TOTPOP) + (MALE_70_TO_74 / E_TOTPOP) + (MA
LE_75_TO_79 / E_TOTPOP) + (MALE_80_TO_84 / E_TOTPOP) + MALE_85_AND_OVE
R / E_TOTPOP AS MALE_OVER_50, (FEMALE_UNDER_5 / E_TOTPOP) + (FEMALE_5_
TO_9 / E_TOTPOP) + (FEMALE_10_TO_14 / E_TOTPOP) + (FEMALE_15_TO_17 / E
_TOTPOP) AS FEMALE_0_TO_17, (FEMALE_18_TO_19 / E_TOTPOP) + (FEMALE_20
/ E_TOTPOP) + (FEMALE_21 / E_TOTPOP) + (FEMALE_22_TO_24 / E_TOTPOP) +
(FEMALE_25_TO_29 / E_TOTPOP) AS FEMALE_18_TO_29, (FEMALE_30_TO_34 / E_
TOTPOP) + (FEMALE_35_TO_39 / E_TOTPOP) + (FEMALE_40_TO_44 / E_TOTPOP)
+ (FEMALE_45_TO_49 / E_TOTPOP) AS FEMALE_30_TO_49, (FEMALE_50_TO_54 /
E_TOTPOP) + (FEMALE_55_TO_59 / E_TOTPOP) + (FEMALE_60_TO_61 / E_TOTPOP
) + (FEMALE_62_TO_64 / E_TOTPOP) + (FEMALE_65_TO_66 / E_TOTPOP) + (FEM
ALE_67_TO_69 / E_TOTPOP) + (FEMALE_70_TO_74 / E_TOTPOP) + (FEMALE_75_T
O_79 / E_TOTPOP) + (FEMALE_80_TO_84 / E_TOTPOP) + FEMALE_85_AND_OVER /
E_TOTPOP AS FEMALE_OVER_50, NOT_US_CITIZEN_POP / E_TOTPOP AS PERCENT_N
ONCITIZEN, WHITE_POP / E_TOTPOP AS PERCENT_WHITE, BLACK_POP / E_TOTPOP
AS PERCENT_BLACK, ASIAN_POP / E_TOTPOP AS PERCENT_ASIAN, HISPANIC_POP
/ E_TOTPOP AS PERCENT_HISPANIC, AMERINDIAN_POP / E_TOTPOP AS PERCENT_A
MERINDIAN, OTHER_RACE_POP / E_TOTPOP AS PERCENT_OTHER_RACE, TWO_OR_MOR
E_RACES_POP / E_TOTPOP AS PERCENT_MULTIRACIAL, HISPANIC_ANY_RACE / E_T
OTPOP AS PERCENT_HISPANIC_ANY_RACE, GINI_INDEX, EP_DISABL, EP_MINRTY,
EP_CROWD, EP_GROUPQ, EP_UNINSUR, EP_NOHSDP, CAST(DATE AS STRING) AS ds
, CONFIRMED_CASES AS y FROM prod_nyt_covid19.covid_aggregate_output WH
ERE RPL_THEMES > 0 AND DEATHS > 0"
```

I created a filtered version of the DataFrame because I wanted to eliminate all values that were not floats or ints before conducting correlations and regressions.

In [434]:
```
# Filtered Dataset

df = client.query(sql_query).to_dataframe()
df_filtered = df.drop(columns=['DATE', 'COUNTY', 'FIPS'])
```

I also decided to use MinMaxScaler() to make all variables range from zero to one because not all variables initially did.

In [435]:
```
# Normalized Dataset

scaler = MinMaxScaler()
df_filtered[['CONFIRMED_CASES', 'DEATHS', 'CC_PER_CAPITA', 'D_PER_CAPI
TA', 'D_PER_CC', 'E_TOTPOP', 'DIABETES_RATE', 'SMOKING_RATE', 'MEDIAN_
INCOME', 'PERCENT_DEM', 'RPL_THEMES', 'CCT_1', 'CCT_2', 'CCT_3', 'CCT_
7', 'CCT_14', 'CCT_21', 'CCT_31', 'DT_1', 'DT_2', 'DT_3', 'DT_7', 'DT_
14', 'DT_21', 'DT_31', 'MALE_0_TO_17', 'MALE_18_TO_29', 'MALE_30_TO_49
', 'MALE_OVER_50', 'FEMALE_0_TO_17', 'FEMALE_18_TO_29', 'FEMALE_30_TO_
49', 'FEMALE_OVER_50', 'PERCENT_NONCITIZEN', 'PERCENT_BLACK', 'PERCENT
_ASIAN', 'PERCENT_HISPANIC', 'PERCENT_AMERINDIAN', 'PERCENT_OTHER_RACE
', 'PERCENT_MULTIRACIAL', 'PERCENT_HISPANIC_ANY_RACE', 'GINI_INDEX', '
EP_DISABL', 'EP_MINRTY', 'EP_CROWD', 'EP_GROUPQ', 'EP_UNINSUR', 'EP_NO
HSDP', 'y']] = scaler.fit_transform(df_filtered[['CONFIRMED_CASES', 'D
EATHS', 'CC_PER_CAPITA', 'D_PER_CAPITA', 'D_PER_CC', 'E_TOTPOP', 'DIAB
ETES_RATE', 'SMOKING_RATE', 'MEDIAN_INCOME', 'PERCENT_DEM', 'RPL_THEME
S', 'CCT_1', 'CCT_2', 'CCT_3', 'CCT_7', 'CCT_14', 'CCT_21', 'CCT_31',
'DT_1', 'DT_2', 'DT_3', 'DT_7', 'DT_14', 'DT_21', 'DT_31', 'MALE_0_TO_
17', 'MALE_18_TO_29', 'MALE_30_TO_49', 'MALE_OVER_50', 'FEMALE_0_TO_17
', 'FEMALE_18_TO_29', 'FEMALE_30_TO_49', 'FEMALE_OVER_50', 'PERCENT_NO
NCITIZEN', 'PERCENT_BLACK', 'PERCENT_ASIAN', 'PERCENT_HISPANIC', 'PERC
ENT_AMERINDIAN', 'PERCENT_OTHER_RACE', 'PERCENT_MULTIRACIAL', 'PERCENT
_HISPANIC_ANY_RACE', 'GINI_INDEX', 'EP_DISABL', 'EP_MINRTY', 'EP_CROWD
', 'EP_GROUPQ', 'EP_UNINSUR', 'EP_NOHSDP', 'y']])
df2 = df_filtered
```

The following displays a summary of CONFIRMED_CASES (no log) and DEATHS (no log) to demonstrate the aforementioned right skew. The 75th percentiles for both confirmed cases and deaths do not even amount to 1% of the cases and deaths (respectively) that the max counties have. The means are significantly higher than the 50th percentiles for both metrics.

In [412]:
```
# COVID-19 Summary

df_ccpc[["CONFIRMED_CASES", "DEATHS"]].describe()
```

Out[412]:

	CONFIRMED_CASES	DEATHS
count	167673.000000	167673.000000
mean	877.813285	41.605375
std	3551.386690	184.431646
min	1.000000	1.000000
25%	44.000000	1.000000
50%	133.000000	4.000000
75%	444.000000	17.000000
max	136129.000000	4729.000000

I regrouped the age metrics into larger classifications to make them more digestible. I made this change after running OLS regressions on the original distributions and saw that not all the groupings were statistically significant. Here is a breakdown of counties' age distributions:

In [422]:
```
# Age Summary

df[["MALE_0_TO_17", "FEMALE_0_TO_17", "MALE_18_TO_29", "FEMALE_18_TO_2
9", "MALE_30_TO_49", "FEMALE_30_TO_49", "MALE_OVER_50", "FEMALE_OVER_5
0"]].describe()
```

Out[422]:

	MALE_0_TO_17	FEMALE_0_TO_17	MALE_18_TO_29	FEMALE_18_TO_29	MALE_30_TO_4
count	167673.000000	167673.000000	167673.000000	167673.000000	167673.00000
mean	0.115337	0.109866	0.081398	0.075342	0.12275
std	0.016342	0.015648	0.021687	0.020521	0.01686
min	0.037794	0.035364	0.025911	0.007347	0.04266
25%	0.105539	0.100580	0.069104	0.063935	0.11302
50%	0.115273	0.110101	0.076048	0.071075	0.12054
75%	0.124378	0.118318	0.086819	0.080620	0.12983
max	0.220096	0.200276	0.384230	0.260955	0.25890

Perhaps I could have also tried taking the log of the racial variables (since they almost all have a right skew), but due to limited time, I will save that as something to consider moving forward. Here is the racial breakdown:

In [432]:
```
# Race Summary

df[["PERCENT_WHITE", "PERCENT_HISPANIC", "PERCENT_BLACK", "PERCENT_ASI
AN", "PERCENT_AMERINDIAN", "PERCENT_OTHER_RACE", "PERCENT_MULTIRACIAL"
]].describe()
```

Out[432]:

	PERCENT_WHITE	PERCENT_HISPANIC	PERCENT_BLACK	PERCENT_ASIAN	PERCENT_
count	167673.000000	167673.000000	167673.000000	167673.000000	
mean	0.721772	0.097399	0.127772	0.019533	
std	0.197950	0.128321	0.164290	0.032810	
min	0.007278	0.000000	0.000000	0.000000	
25%	0.592290	0.025563	0.013838	0.004476	
50%	0.765100	0.049563	0.052844	0.009057	
75%	0.890198	0.109242	0.181312	0.021288	
max	0.994990	0.990688	0.874123	0.414846	

Here is another summary of the remaining variables, which also display right skews in most cases (though less significant ones than in the previous tables):

In [438]:
```
# Other Summary

df_ccpc[["E_TOTPOP", "EP_PCI", "EP_POV", "EP_CROWD", "EP_DISABL", "EP_
GROUPQ", "EP_MINRTY", "EP_NOHSDP", "EP_UNINSUR", "RPL_THEMES", "PERCEN
T_DEM", "PERCENT_REP", "PERCENT_NONCITIZEN", "DIABETES_RATE", "SMOKING
_RATE"]].describe()
```

Out[438]:

	E_TOTPOP	EP_PCI	EP_POV	EP_CROWD	EP_DISABL	EP_GROU
count	1.676730e+05	167673.000000	167673.000000	167673.000000	167673.000000	167673.000
mean	1.858528e+05	28059.872502	15.613688	2.438397	15.083568	3.310
std	4.639278e+05	7227.302208	6.253014	1.745420	4.023525	3.774
min	1.328000e+03	10931.000000	2.700000	0.000000	3.800000	0.000
25%	2.507600e+04	23179.000000	11.100000	1.400000	12.300000	1.300
50%	5.679000e+04	26867.000000	14.900000	2.000000	14.800000	2.000
75%	1.620520e+05	31330.000000	19.000000	2.900000	17.600000	3.800
max	1.009805e+07	69775.000000	55.100000	15.500000	31.700000	36.200

I conducted correlations between CONFIRMED_CASES and DEATHS and all other variables.

CONFIRMED_CASES and DEATHS have a correlation of 83.5%, a number which was previously ~ 96% (supporting the theory that people are now more prepared to handle COVID-19 cases and prevent deaths).

E_TOTPOP has a correlation of 67.9% with CONFIRMED_CASES and 61.8% with DEATHS. This high correlation is why it is important to consider CC_PER_CAPITA (which I do later in descriptive modeling).

EP_DISABL has a correlation of -41.3% with CONFIRMED_CASES and -33.8% with DEATHS. These correlations go against my Data Studio results, which state that in counties with populations over 1,000,000 people, there is a positive correlation between EP_DISABL and death rate. Looking at all counties regardless of population changes this relationship.

EP_MINRTY has a correlation of 31.9% with CONFIRMED_CASES and 25.4% with DEATHS. Looking at the racial breakdowns, racial percentages for all races except white and Native American are positively correlated with CONFIRMED_CASES and DEATHS. The strongest correlations are between CONFIRMED_CASES/DEATHS and PERCENT_ASIAN (40.2% and 37.8% respectively) and between CONFIRMED_CASES/DEATHS and PERCENT_NONCITIZEN (43.1% and 33.0% respectively). Perhaps a cause for this trend is that PERCENT_ASIAN and PERCENT_NONCITIZEN are higher in more populated counties, which naturally have higher CONFIRMED_CASES and DEATHS.

According to the table below, PERCENT_DEM is positively correlated with CONFIRMED_CASES and DEATHS, whereas PERCENT_REP is negatively correlated with these variables. In Data Studio, looking at CC_PER_CAPITA revealed that red counties were experiencing higher case counts relative to their populations than blue counties. The reason for this phenomenon might also be attributed to population, which is typically higher in blue counties and thus makes them more prone to cases and deaths.

RPL_THEMES surprisingly had a low correlation with both variables, directly contrasting the Data Studio results; perhaps there is an unknown error here that requires further investigation.

Contrary to my original hypothesis, the rates for diabetes and smoking in counties had negative correlations with cases and deaths.

Looking at the age breakdowns, MALE_OVER_50 and FEMALE_OVER_50 were both negatively correlated with both CONFIRMED_CASES and DEATHS. This result is somewhat surprising because it is known that older people have a higher chance of dying from COVID-19. It is possible that counties higher populations of old people have lower total populations and have not been exposed to the virus as much; also, it could be that counties with smaller populations of old people experience the most elderly deaths because the younger majority spreads the disease to them (maybe older communities are more careful).

In [443]:
```
# Correlation

corr_df = df[["CONFIRMED_CASES", "DEATHS", "E_TOTPOP", "MEDIAN_INCOME"
, "EP_CROWD", "EP_DISABL", "EP_GROUPQ", "EP_MINRTY", "EP_NOHSDP", "EP_
UNINSUR", "RPL_THEMES", "PERCENT_DEM", "PERCENT_REP", "DIABETES_RATE",
"SMOKING_RATE", "PERCENT_WHITE", "PERCENT_HISPANIC", "PERCENT_BLACK",
"PERCENT_ASIAN", "PERCENT_AMERINDIAN", "PERCENT_OTHER_RACE", "PERCENT_
MULTIRACIAL", "PERCENT_NONCITIZEN", "MALE_0_TO_17", "FEMALE_0_TO_17",
"MALE_18_TO_29", "FEMALE_18_TO_29", "MALE_30_TO_49", "FEMALE_30_TO_49"
, "MALE_OVER_50", "FEMALE_OVER_50"]].corr()
corr_df[["CONFIRMED_CASES", "DEATHS"]].head(100)
```

Out[443]:

	CONFIRMED_CASES	DEATHS
CONFIRMED_CASES	1.000000	0.835464
DEATHS	0.835464	1.000000
E_TOTPOP	0.678758	0.617906
MEDIAN_INCOME	0.146140	0.130493
EP_CROWD	0.158024	0.081254
EP_DISABL	-0.412787	-0.337906
EP_GROUPQ	-0.070507	-0.087261
EP_MINRTY	0.319374	0.254689
EP_NOHSDP	-0.106140	-0.139878
EP_UNINSUR	-0.094023	-0.136692
RPL_THEMES	-0.001631	-0.033855
PERCENT_DEM	0.438792	0.419662
PERCENT_REP	-0.441821	-0.414353
DIABETES_RATE	-0.118539	-0.121600
SMOKING_RATE	-0.313857	-0.268886
PERCENT_WHITE	-0.319357	-0.254671
PERCENT_HISPANIC	0.225984	0.145140
PERCENT_BLACK	0.129173	0.118558
PERCENT_ASIAN	0.401815	0.377111
PERCENT_AMERINDIAN	-0.039581	-0.030477
PERCENT_OTHER_RACE	0.229059	0.228421
PERCENT_MULTIRACIAL	0.067010	0.054283

PERCENT_NONCITIZEN	0.430870	0.329861
MALE_0_TO_17	0.106576	0.038342
FEMALE_0_TO_17	0.132421	0.055779
MALE_18_TO_29	0.152525	0.068397
FEMALE_18_TO_29	0.231716	0.166912
MALE_30_TO_49	0.218077	0.133260
FEMALE_30_TO_49	0.421570	0.353427
MALE_OVER_50	-0.392445	-0.266771
FEMALE_OVER_50	-0.306205	-0.175755

I also conducted autocorrelations to better understand the relationships between CONFIRMED_CASES and DEATHS over time.

Cases on a given day are still the most highly correlated with deaths on that day.

CCT_1 (cases yesterday) are most correlated with CCT_3 and CCT_7 (with correlations of 88.2% and 88.4% respectively).

Similarly, DT_1 (deaths yesterday) are most correlated with DT_3 and DT_7 (with correlations of 94.2% and 94.1% respectively).

The autocorrelations for deaths against deaths are stronger than for cases against cases, meaning that more accurate predictions can be made about deaths using death past data than about cases using past case data.

In [410]:
```
# Autocorrelation

sql_query_lag  = "select lag(CONFIRMED_CASES,1) over (partition by COU
NTY order by DATE asc) as CCT_1, lag(CONFIRMED_CASES,2) over (partitio
n by COUNTY order by DATE asc) as CCT_2, lag(CONFIRMED_CASES,3) over (
partition by COUNTY order by DATE asc) as CCT_3, lag(CONFIRMED_CASES,7
) over (partition by COUNTY order by DATE) as CCT_7, lag(CONFIRMED_CAS
ES,14) over (partition by COUNTY order by DATE asc) as CCT_14, lag(CON
FIRMED_CASES,21) over (partition by COUNTY order by DATE asc) as CCT_2
1, lag(CONFIRMED_CASES,31) over (partition by COUNTY order by DATE asc
) as CCT_31, lag(deaths,1) over (partition by COUNTY order by DATE asc
) as DT_1, lag(deaths,2) over (partition by COUNTY order by DATE asc)
as DT_2, lag(deaths,3) over (partition by COUNTY order by DATE asc) as
DT_3, lag(deaths,7) over (partition by COUNTY order by DATE asc) as DT
_7, lag(DEATHS,14) over (partition by COUNTY order by DATE asc) as DT_
14, lag(DEATHS,21) over (partition by COUNTY order by DATE asc) as DT_
21, lag(DEATHS,31) over (partition by COUNTY order by DATE asc) as DT_
31 from prod_nyt_covid19.covid_aggregate_output"
df_lag = client.query(sql_query_lag).to_dataframe()
corr_df_lag = df_lag.corr()
corr_df_lag[["CCT_1", "CCT_2", "CCT_3", "CCT_7", "CCT_14", "CCT_21", "
CCT_31", "DT_1", "DT_2", "DT_3", "DT_7","DT_14", "DT_21", "DT_31"]].he
ad(100)
```

Out[410]:

	CCT_1	CCT_2	CCT_3	CCT_7	CCT_14	CCT_21	CCT_31	DT_1	D
CCT_1	1.000000	0.853979	0.882020	0.884460	0.860776	0.839344	0.801354	0.939660	0.847
CCT_2	0.853979	1.000000	0.852735	0.886281	0.864396	0.853781	0.804083	0.850120	0.940
CCT_3	0.882020	0.852735	1.000000	0.882116	0.876885	0.846249	0.803989	0.867787	0.850
CCT_7	0.884460	0.886281	0.882116	1.000000	0.868011	0.865982	0.821217	0.874510	0.875
CCT_14	0.860776	0.864396	0.876885	0.868011	1.000000	0.858263	0.833629	0.864888	0.867
CCT_21	0.839344	0.853781	0.846249	0.865982	0.858263	1.000000	0.847932	0.849194	0.857
CCT_31	0.801354	0.804083	0.803989	0.821217	0.833629	0.847932	1.000000	0.810453	0.813
DT_1	0.939660	0.850120	0.867787	0.874510	0.864888	0.849194	0.810453	1.000000	0.932
DT_2	0.847434	0.940980	0.850414	0.875232	0.867174	0.857918	0.813362	0.932321	1.000
DT_3	0.862838	0.848225	0.942101	0.872231	0.874479	0.853969	0.815380	0.942092	0.937
DT_7	0.861809	0.865137	0.864242	0.945932	0.869542	0.867567	0.829879	0.940988	0.942
DT_14	0.840912	0.845298	0.854543	0.856706	0.950810	0.864803	0.843584	0.922747	0.925
DT_21	0.818757	0.829474	0.826949	0.846187	0.854382	0.952834	0.856703	0.898585	0.905
DT_31	0.778038	0.782235	0.785611	0.803509	0.824353	0.844788	0.952892	0.850156	0.854

Descriptive Modeling

I did a series of ordinary least squares (OLS) regressions using the following as dependent variables: CONFIRMED_CASES, DEATHS, CC_PER_CAPITA, and error. I chose this regression because it clearly showed me the relationships amongst variables.

I initially wanted to use the forward selection code below to automatically select the variables that would optimize the model, but the factorial runtime would have had me waiting years for the algorithm to run through the hundreds of input variables in my SQL code. I decided to run the regression on all the variables to begin with and removed the statistically insignificant variables thereafter.

In [ ]:
```
# Forward Selection

def forward_selected(data, response):
    remaining = set(data.columns)
    remaining.remove(response)
    selected = []
    current_score, best_new_score = 0.0, 0.0
    while remaining and current_score == best_new_score:
        scores_with_candidates = []
        for candidate in remaining:
            formula = "{} ~ {} + 1".format(response,
                                           ' + '.join(selected + [cand
idate]))
            score = smf.ols(formula, data).fit().rsquared_adj
            scores_with_candidates.append((score, candidate))
        scores_with_candidates.sort()
        best_new_score, best_candidate = scores_with_candidates.pop()
        if current_score < best_new_score:
            remaining.remove(best_candidate)
            selected.append(best_candidate)
            current_score = best_new_score
    formula = "{} ~ {} + 1".format(response,
                                   ' + '.join(selected))
    model = smf.ols(formula, data).fit()
    return model
```

The first regression I did was looking at CONFIRMED_CASES as the dependent variable and had an adjusted R-squared value (R-squared accounting for the number of variables) of 0.769, meaning that 76.9% of the data fits the regression model.

The skew of the dataset is well between -0.5 and 0.5, meaning that it is essentially symmetric. Previously, before I took the log of the highly skewed variables, this value had a magnitude of ~ 3.

The largest positive coefficient value was for total population (0.42), which makes sense because holding everything else constant, counties with higher populations will have more exposure to and higher spread of the disease.

PERCENT_AMERINDIAN in a county has the highest impact on CONFIRMED_CASES out of all the racial demographics (the news mostly talks about black and Hispanic). This result contrasts with my correlations, which suggested that PERCENT_ASIAN was the most highly correlated racial demographic with CONFIRMED_CASES (most likely from omitted variable bias); however, similarly to my correlation results, PERCENT_NONCITIZEN is more impactful than all other race-related variables (with a coefficient of 0.27).

MEDIAN_INCOME and RPL_THEMES surprisingly did not have much of an effect (the latter is possibly due to an unkown error).

All age groups in the model had a lower impact than males 30 to 49 (the base) on CONFIRMED_CASES.

PERCENT_DEM has a very minimal impact on CONFIRMED_CASES (contrary to my observations in Data Studio).

In [445]:
```
# OLS Regression (CONFIRMED_CASES)

results_cc = smf.ols(formula = "CONFIRMED_CASES ~ CCT_1 + CCT_2 + CCT_
3 + CCT_7 + CCT_14 + CCT_31 + E_TOTPOP + MEDIAN_INCOME + EP_CROWD + EP
_DISABL + EP_GROUPQ + EP_NOHSDP + EP_UNINSUR + RPL_THEMES + PERCENT_DE
M + DIABETES_RATE + SMOKING_RATE + PERCENT_HISPANIC + PERCENT_BLACK +
PERCENT_ASIAN + PERCENT_AMERINDIAN + PERCENT_OTHER_RACE + PERCENT_MULT
IRACIAL + PERCENT_NONCITIZEN + MALE_0_TO_17 + FEMALE_0_TO_17 + MALE_18
_TO_29 + FEMALE_18_TO_29 + FEMALE_30_TO_49 + MALE_OVER_50 + FEMALE_OVE
R_50", data=df2).fit()
print(results_cc.summary())
```

```
                            OLS Regression Results
==============================================================================
==========
Dep. Variable:        CONFIRMED_CASES   R-squared:
0.769
Model:                            OLS   Adj. R-squared:
0.769
Method:                 Least Squares   F-statistic:
1.313e+04
```

```
Date:                Thu, 16 Jul 2020   Prob (F-statistic):
0.00
Time:                        00:31:19   Log-Likelihood:
1.4902e+05
No. Observations:              122121   AIC:
-2.980e+05
Df Residuals:                  122089   BIC:
-2.977e+05
Df Model:                          31
Covariance Type:            nonrobust
==============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.4743      0.016     29.708      0.000       0.443       0.506
CCT_1          0.0214      0.002     10.279      0.000       0.017       0.025
CCT_2          0.1025      0.002     49.686      0.000       0.098       0.107
CCT_3          0.1158      0.002     55.032      0.000       0.112       0.120
CCT_7          0.0991      0.002     44.889      0.000       0.095       0.103
CCT_14         0.0841      0.002     38.126      0.000       0.080       0.088
CCT_31         0.0721      0.002     38.924      0.000       0.068       0.076
E_TOTPOP       0.4152      0.002    174.338      0.000       0.411       0.420
MEDIAN_INCOME  0.0220      0.002     12.052      0.000       0.018       0.026
EP_CROWD      -0.0743      0.003    -21.318      0.000      -0.081      -0.067
EP_DISABL     -0.0768      0.003    -24.417      0.000      -0.083      -0.071
EP_GROUPQ     -0.0596      0.004    -15.071      0.000      -0.067      -0.052
EP_NOHSDP      0.0253      0.004      6.291      0.000       0.017       0.033
EP_UNINSUR    -0.1046      0.003    -36.338      0.000      -0.110      -0.099
RPL_THEMES     0.0320      0.002     17.036      0.000       0.028       0.036
PERCENT_DEM   -0.0105      0.002     -4.603      0.000      -0.015      -0.006
DIABETES_RATE -0.0136      0.003     -4.787      0.000      -0.019      -0.008
```

```
SMOKING_RATE             0.0108      0.003      3.545      0.000
0.005       0.017
PERCENT_HISPANIC        -0.0510      0.004    -14.328      0.000
-0.058      -0.044
PERCENT_BLACK            0.1164      0.003     39.922      0.000
0.111       0.122
PERCENT_ASIAN           -0.1031      0.005    -22.706      0.000
-0.112      -0.094
PERCENT_AMERINDIAN       0.1871      0.005     38.490      0.000
0.178       0.197
PERCENT_OTHER_RACE       0.0890      0.006     15.827      0.000
0.078       0.100
PERCENT_MULTIRACIAL     -0.0731      0.003    -21.478      0.000
-0.080      -0.066
PERCENT_NONCITIZEN       0.2621      0.004     60.083      0.000
0.254       0.271
MALE_0_TO_17            -0.0623      0.007     -8.787      0.000
-0.076      -0.048
FEMALE_0_TO_17          -0.1164      0.008    -15.344      0.000
-0.131      -0.102
MALE_18_TO_29           -0.2276      0.013    -17.153      0.000
-0.254      -0.202
FEMALE_18_TO_29         -0.1829      0.007    -26.785      0.000
-0.196      -0.169
FEMALE_30_TO_49         -0.1544      0.007    -22.833      0.000
-0.168      -0.141
MALE_OVER_50            -0.4444      0.013    -34.941      0.000
-0.469      -0.419
FEMALE_OVER_50          -0.0349      0.009     -3.884      0.000
-0.053      -0.017
==============================================================================
Omnibus:                     3227.683   Durbin-Watson:
1.598
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
5888.326
Skew:                          -0.214   Prob(JB):
0.00
Kurtosis:                       3.987   Cond. No.
262.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.
```

I then did a regression looking at DEATHS as the dependent variable and had a slightly lower adjusted R-squared value of 0.647, meaning that 64.7% of the data fits the regression model.

Variables regarding deaths over time are included in this model even though they were excluded in the previous one (because the dead cannot affect the living).

Again, the largest positive coefficient value was for total population (0.62), supporting the fact that deaths occur disproportionately in urban areas. Population is the most likely culprit for the discrepancies between my correlations and regressions (caused by omitted variable bias).

PERCENT_AMERINDIAN in a county also has the highest impact on DEATHS out of all the racial demographics.

Overall, this regression was very similar to the previous model (since CONFIRMED_CASES and DEATHS are so highly related).

In [402]:
```
# OLS Regression (DEATHS)

results_d = smf.ols(formula = "DEATHS ~ CCT_1 + CCT_2 + CCT_3 + CCT_7
+ CCT_14 + CCT_31 + DT_1 + DT_2 + DT_3 + DT_7 + DT_14 + DT_31 + E_TOTP
OP + MEDIAN_INCOME + EP_CROWD + EP_DISABL + EP_GROUPQ + EP_NOHSDP + EP
_UNINSUR + RPL_THEMES + PERCENT_DEM + DIABETES_RATE + SMOKING_RATE + P
ERCENT_HISPANIC + PERCENT_BLACK + PERCENT_ASIAN + PERCENT_AMERINDIAN +
PERCENT_OTHER_RACE + PERCENT_MULTIRACIAL + PERCENT_NONCITIZEN + MALE_0
_TO_17 + FEMALE_0_TO_17 + MALE_18_TO_29 + FEMALE_18_TO_29 + FEMALE_30_
TO_49 + MALE_OVER_50 + FEMALE_OVER_50", data=df2).fit()
print(results_d.summary())
```

```
                            OLS Regression Results
==============================================================================
==========
Dep. Variable:                 DEATHS   R-squared:
0.647
Model:                            OLS   Adj. R-squared:
0.647
Method:                 Least Squares   F-statistic:
6059.
Date:                Wed, 15 Jul 2020   Prob (F-statistic):
0.00
Time:                        12:28:51   Log-Likelihood:
78062.
No. Observations:              122121   AIC:
-1.560e+05
Df Residuals:                  122083   BIC:
-1.557e+05
Df Model:                          37
Covariance Type:            nonrobust
```

```
==============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.2220      0.029     -7.768      0.000      -0.278      -0.166
CCT_1          0.0079      0.004      1.918      0.055      -0.000       0.016
CCT_2          0.0686      0.004     16.981      0.000       0.061       0.077
CCT_3          0.0930      0.004     22.477      0.000       0.085       0.101
CCT_7          0.1067      0.004     24.600      0.000       0.098       0.115
CCT_14         0.1121      0.004     26.037      0.000       0.104       0.121
CCT_31         0.1226      0.004     34.161      0.000       0.116       0.130
DT_1          -0.0164      0.014     -1.171      0.242      -0.044       0.011
DT_2           0.2117      0.013     15.816      0.000       0.185       0.238
DT_3           0.1637      0.014     11.666      0.000       0.136       0.191
DT_7           0.0963      0.015      6.495      0.000       0.067       0.125
DT_14          0.0856      0.015      5.720      0.000       0.056       0.115
DT_31         -0.0218      0.015     -1.494      0.135      -0.050       0.007
E_TOTPOP       0.6244      0.004    146.657      0.000       0.616       0.633
MEDIAN_INCOME  0.0214      0.003      6.564      0.000       0.015       0.028
EP_CROWD      -0.2019      0.006    -32.398      0.000      -0.214      -0.190
EP_DISABL     -0.0433      0.006     -7.689      0.000      -0.054      -0.032
EP_GROUPQ      0.0192      0.007      2.710      0.007       0.005       0.033
EP_NOHSDP      0.0692      0.007      9.634      0.000       0.055       0.083
EP_UNINSUR    -0.1136      0.005    -22.082      0.000      -0.124      -0.104
RPL_THEMES     0.0205      0.003      6.089      0.000       0.014       0.027
PERCENT_DEM    0.0831      0.004     20.350      0.000       0.075       0.091
```

```
DIABETES_RATE          -0.0853      0.005    -16.841      0.000
-0.095      -0.075
SMOKING_RATE            0.0772      0.005     14.219      0.000
0.067       0.088
PERCENT_HISPANIC       -0.0156      0.006     -2.449      0.014
-0.028      -0.003
PERCENT_BLACK           0.1541      0.005     29.548      0.000
0.144       0.164
PERCENT_ASIAN           0.1302      0.008     16.002      0.000
0.114       0.146
PERCENT_AMERINDIAN      0.3538      0.009     40.688      0.000
0.337       0.371
PERCENT_OTHER_RACE      0.2253      0.010     22.377      0.000
0.206       0.245
PERCENT_MULTIRACIAL    -0.1493      0.006    -24.506      0.000
-0.161      -0.137
PERCENT_NONCITIZEN      0.1738      0.008     22.239      0.000
0.158       0.189
MALE_0_TO_17            0.1846      0.013     14.556      0.000
0.160       0.210
FEMALE_0_TO_17         -0.1014      0.014     -7.475      0.000
-0.128      -0.075
MALE_18_TO_29          -0.3596      0.024    -15.154      0.000
-0.406      -0.313
FEMALE_18_TO_29         0.0502      0.012      4.111      0.000
0.026       0.074
FEMALE_30_TO_49        -0.1102      0.012     -9.109      0.000
-0.134      -0.087
MALE_OVER_50           -0.1816      0.023     -7.982      0.000
-0.226      -0.137
FEMALE_OVER_50          0.1300      0.016      8.084      0.000
0.098       0.162
==============================================================================
Omnibus:                       33.250   Durbin-Watson:
1.295
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
32.198
Skew:                           0.026   Prob(JB):
1.02e-07
Kurtosis:                       2.940   Cond. No.
263.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.
```

In order to eliminate the impacts of population on the model, I used CC_PER_CAPITA as the dependent variable and had an adjusted R-squared value of 0.669, meaning that 66.9% of the data fits the regression model.

The most impactful variables in the model after eliminating population were cases from previous days. CCT_3 (cases three days ago) is the variable with the highest coefficient value in the equation for determining CC_PER_CAPITA.

An issue with this model is that it is highly skewed as a result of none of the variables being logged. With more time, I would experiment logging the most highly skewed variables to see how that effects the model.

In part because the variables describing cases from previous days were the most profound in determining CC_PER_CAPITA, I decided to solely focus on CONFIRMED_CASES over time for my forecasting model (which itself reflects the demographic intricacies of counties).

In [444]:
```
# OLS Regression (CC_PER_CAPITA)

sql_query_ccpc = "SELECT DATE, COUNTY, FIPS, CONFIRMED_CASES, DEATHS,
CONFIRMED_CASES / E_TOTPOP AS CC_PER_CAPITA, DEATHS / E_TOTPOP AS D_PE
R_CAPITA, DEATHS / CONFIRMED_CASES AS D_PER_CC, E_TOTPOP, DIABETES_RAT
E, SMOKING_RATE, ROUND(LN(MEDIAN_INCOME)) AS MEDIAN_INCOME, PERCENT_DE
M, PERCENT_REP, RPL_THEMES, lag(CONFIRMED_CASES / E_TOTPOP, 1) over (p
artition by COUNTY order by DATE asc) as CCT_1, lag(CONFIRMED_CASES /
E_TOTPOP, 2) over (partition by COUNTY order by DATE asc) as CCT_2, la
g(CONFIRMED_CASES / E_TOTPOP, 3) over (partition by COUNTY order by DA
TE asc) as CCT_3, lag(CONFIRMED_CASES / E_TOTPOP, 7) over (partition b
y COUNTY order by DATE) as CCT_7, lag(CONFIRMED_CASES / E_TOTPOP, 14)
over (partition by COUNTY order by DATE asc) as CCT_14, lag(CONFIRMED_
CASES / E_TOTPOP, 21) over (partition by COUNTY order by DATE asc) as
CCT_21, lag(CONFIRMED_CASES / E_TOTPOP, 31) over (partition by COUNTY
order by DATE asc) as CCT_31, lag(DEATHS / E_TOTPOP, 1) over (partitio
n by COUNTY order by DATE asc) as DT_1, lag(DEATHS / E_TOTPOP, 2) over
(partition by COUNTY order by DATE asc) as DT_2, lag(DEATHS / E_TOTPOP
, 3) over (partition by COUNTY order by DATE asc) as DT_3, lag(DEATHS
/ E_TOTPOP, 7) over (partition by COUNTY order by DATE asc) as DT_7, l
ag(DEATHS / E_TOTPOP, 14) over (partition by COUNTY order by DATE asc)
as DT_14, lag(DEATHS / E_TOTPOP, 21) over (partition by COUNTY order b
y DATE asc) as DT_21, lag(DEATHS / E_TOTPOP, 31) over (partition by CO
UNTY order by DATE asc) as DT_31, ((MALE_UNDER_5 / E_TOTPOP) + (MALE_5
_TO_9 / E_TOTPOP) + (MALE_10_TO_14 / E_TOTPOP) + (MALE_15_TO_17 / E_TO
TPOP)) as MALE_0_TO_17, ((MALE_18_TO_19 / E_TOTPOP) + (MALE_20 / E_TOT
POP) + (male_21 / E_TOTPOP) + (male_22_to_24 / E_TOTPOP) + (male_25_to
_29 / E_TOTPOP)) as MALE_18_TO_29, (male_30_to_34 / E_TOTPOP) + (male_
35_to_39 / E_TOTPOP) + (male_40_to_44 / E_TOTPOP) + (male_45_to_49 / E
_TOTPOP) as MALE_30_TO_49, (male_50_to_54 / E_TOTPOP) + (male_55_to_59
/ E_TOTPOP) + (male_60_to_61 / E_TOTPOP) + (male_62_to_64 / E_TOTPOP)
+ (male_65_to_66 / E_TOTPOP) + (male_67_to_69 / E_TOTPOP) + (male_70_t
```

```
o_74 / E_TOTPOP) + (male_75_to_79 / E_TOTPOP) + (male_80_to_84 / E_TOT
POP) + male_85_and_over / E_TOTPOP as MALE_OVER_50, (female_under_5 /
E_TOTPOP) + (female_5_to_9 / E_TOTPOP) + (female_10_to_14 / E_TOTPOP)
+ (female_15_to_17 / E_TOTPOP) as FEMALE_0_TO_17, (female_18_to_19 / E
_TOTPOP) + (female_20 / E_TOTPOP) + (female_21 / E_TOTPOP) + (female_2
2_to_24 / E_TOTPOP) + (female_25_to_29 / E_TOTPOP) as FEMALE_18_TO_29,
(female_30_to_34 / E_TOTPOP) + (female_35_to_39 / E_TOTPOP) + (female_
40_to_44 / E_TOTPOP) + (female_45_to_49 / E_TOTPOP) as FEMALE_30_TO_49
, (female_50_to_54 / E_TOTPOP) + (female_55_to_59 / E_TOTPOP) + (femal
e_60_to_61 / E_TOTPOP) + (female_62_to_64 / E_TOTPOP) + (female_65_to_
66 / E_TOTPOP) + (female_67_to_69 / E_TOTPOP) + (female_70_to_74 / E_T
OTPOP) + (female_75_to_79 / E_TOTPOP) + (female_80_to_84 / E_TOTPOP) +
female_85_and_over / E_TOTPOP as FEMALE_OVER_50, not_us_citizen_pop /
E_TOTPOP as PERCENT_NONCITIZEN, black_pop / E_TOTPOP as PERCENT_BLACK,
asian_pop / E_TOTPOP as PERCENT_ASIAN, hispanic_pop / E_TOTPOP as PERC
ENT_HISPANIC, amerindian_pop / E_TOTPOP as PERCENT_AMERINDIAN, other_r
ace_pop / E_TOTPOP as PERCENT_OTHER_RACE, two_or_more_races_pop / E_TO
TPOP as PERCENT_MULTIRACIAL, hispanic_any_race / E_TOTPOP as PERCENT_H
ISPANIC_ANY_RACE, GINI_INDEX, EP_DISABL, EP_MINRTY, EP_CROWD, EP_GROUP
Q, EP_UNINSUR, EP_NOHSDP, EP_PCI, EP_POV, CAST(DATE AS STRING) AS ds,
CONFIRMED_CASES AS y FROM prod_nyt_covid19.covid_aggregate_output WHER
E RPL_THEMES > 0 AND DEATHS > 0"
df_ccpc = client.query(sql_query_ccpc).to_dataframe()
results_ccpc = smf.ols(formula = "CC_PER_CAPITA ~ CCT_1 + CCT_2 + CCT_
3 + CCT_7 + CCT_14 + CCT_31 + MEDIAN_INCOME + EP_CROWD + EP_DISABL + E
P_GROUPQ + EP_NOHSDP + EP_UNINSUR + RPL_THEMES + PERCENT_DEM + DIABETE
S_RATE + SMOKING_RATE + PERCENT_HISPANIC + PERCENT_BLACK + PERCENT_ASI
AN + PERCENT_AMERINDIAN + PERCENT_OTHER_RACE + PERCENT_MULTIRACIAL + P
ERCENT_NONCITIZEN + MALE_0_TO_17 + FEMALE_0_TO_17 + MALE_18_TO_29 + FE
MALE_18_TO_29 + FEMALE_30_TO_49 + MALE_OVER_50 + FEMALE_OVER_50", data
=df_ccpc).fit()
print(results_ccpc.summary())
```

```
                            OLS Regression Results
==============================================================================
==========
Dep. Variable:          CC_PER_CAPITA   R-squared:
0.669
Model:                            OLS   Adj. R-squared:
0.669
Method:                 Least Squares   F-statistic:
8472.
Date:                Thu, 16 Jul 2020   Prob (F-statistic):
0.00
Time:                        00:31:18   Log-Likelihood:
4.9958e+05
No. Observations:              125785   AIC:
-9.991e+05
Df Residuals:                  125754   BIC:
-9.988e+05
```

```
Df Model:                           30
Covariance Type:            nonrobust
==============================================================================
====================
                           coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
--------------------
Intercept                0.0345      0.002     22.285      0.000
0.031       0.038
CCT_1                    0.0632      0.003     25.191      0.000
0.058       0.068
CCT_2                    0.1844      0.003     73.519      0.000
0.179       0.189
CCT_3                    0.2249      0.003     88.178      0.000
0.220       0.230
CCT_7                    0.1702      0.003     62.704      0.000
0.165       0.176
CCT_14                   0.1411      0.003     50.750      0.000
0.136       0.147
CCT_31                   0.0852      0.003     31.782      0.000
0.080       0.090
MEDIAN_INCOME            0.0003   5.73e-05      5.238      0.000
0.000       0.000
EP_CROWD                -0.0002   1.41e-05    -11.625      0.000
-0.000      -0.000
EP_DISABL           -1.301e-05      7e-06     -1.857      0.063   -2
.67e-05    7.19e-07
EP_GROUPQ           -1.735e-05   6.88e-06     -2.521      0.012   -3
.08e-05   -3.86e-06
EP_NOHSDP            7.029e-05   5.18e-06     13.570      0.000    6
.01e-05    8.04e-05
EP_UNINSUR              -0.0002   4.47e-06    -39.377      0.000
-0.000      -0.000
RPL_THEMES               0.0006      0.000      5.062      0.000
0.000       0.001
PERCENT_DEM              0.0009      0.000      5.516      0.000
0.001       0.001
DIABETES_RATE       -2.571e-05   1.19e-05     -2.169      0.030    -
4.9e-05   -2.48e-06
SMOKING_RATE         7.332e-05   6.84e-06     10.723      0.000    5
.99e-05    8.67e-05
PERCENT_HISPANIC        -0.0045      0.000    -20.013      0.000
-0.005      -0.004
PERCENT_BLACK            0.0066      0.000     31.706      0.000
0.006       0.007
PERCENT_ASIAN           -0.0132      0.001    -19.327      0.000
-0.015      -0.012
PERCENT_AMERINDIAN       0.0121      0.000     32.495      0.000
0.011       0.013
```

```
PERCENT_OTHER_RACE      0.0325      0.006      5.050      0.000
0.020       0.045
PERCENT_MULTIRACIAL    -0.0194      0.001    -17.282      0.000
-0.022      -0.017
PERCENT_NONCITIZEN      0.0584      0.001     68.254      0.000
0.057       0.060
MALE_0_TO_17            0.0122      0.002      5.028      0.000
0.007       0.017
FEMALE_0_TO_17         -0.0492      0.003    -17.031      0.000
-0.055      -0.044
MALE_18_TO_29          -0.0320      0.002    -13.765      0.000
-0.037      -0.027
FEMALE_18_TO_29        -0.0536      0.002    -31.437      0.000
-0.057      -0.050
FEMALE_30_TO_49        -0.1138      0.003    -38.903      0.000
-0.120      -0.108
MALE_OVER_50           -0.0448      0.002    -18.584      0.000
-0.049      -0.040
FEMALE_OVER_50         -0.0368      0.002    -22.050      0.000
-0.040      -0.033
==============================================================
==========
Omnibus:                   101723.694   Durbin-Watson:
2.024
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         13
226931.162
Skew:                           3.201   Prob(JB):
0.00
Kurtosis:                      52.827   Cond. No.
1.92e+04
==============================================================
==========

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.
[2] The condition number is large, 1.92e+04. This might indicate tha
t there are
strong multicollinearity or other numerical problems.
```

Forecasting

For forecasting, I initially selected a few demographically and geographically diverse counties to focus on. The counties I selected were the following: Prince George's, Miami-Dade, Westchester, Los Angeles, and Bexar.

First, I had to recursively join COUNTY, ds, and y for every county into one aggregated dataset.

The training period I used was from whenever the county's first case was to 05/31/20 (since this date approximately marked somewhat of a shift in the communities the virus was affecting). The models each predict two weeks ahead.

In [ ]:
```
# Prince George's Join

m_pg = Prophet(changepoint_prior_scale = 0.5, interval_width = 0.95)
df_pg = df[(df.COUNTY == "Prince George's") & (df.ds <= '2020-05-31')]
[["COUNTY", "ds", "y"]]
m_pg.fit(df_pg)
future_pg = m.make_future_dataframe(periods=14)
train_performance_pg = m_pg.predict(df_pg[["ds"]])
train_performance_pg['ds'] = train_performance_pg['ds'].dt.strftime('%
Y-%m-%d')
df_pg_join = pd.merge(df_pg, train_performance_pg, on = 'ds', how = 'l
eft')

# Miami-Dade Join

m_miami = Prophet(changepoint_prior_scale = 0.5, interval_width = 0.95
)
df_miami = df[(df.COUNTY == "Miami-Dade") & (df.ds <= '2020-05-31')][[
"COUNTY", "ds", "y"]]
m_miami.fit(df_miami)
future_miami = m_miami.make_future_dataframe(periods=14)
train_performance_miami = m_miami.predict(df_miami[["ds"]])
train_performance_miami['ds'] = train_performance_miami['ds'].dt.strft
ime('%Y-%m-%d')
df_miami_join = pd.merge(df_miami, train_performance_miami, on = 'ds',
how = 'left')
df_pg_miami = pd.concat([df_pg_join, df_miami_join])

# Westchester Join

m_wc = Prophet(changepoint_prior_scale = 0.5, interval_width = 0.95)
df_wc = df[(df.COUNTY == "Westchester") & (df.ds <= '2020-05-31')][["C
OUNTY", "ds", "y"]]
m_wc.fit(df_wc)
future_wc = m_wc.make_future_dataframe(periods=14)
train_performance_wc = m_wc.predict(df_wc[["ds"]])
train_performance_wc['ds'] = train_performance_wc['ds'].dt.strftime('%
```

```
Y-%m-%d')
df_wc_join = pd.merge(df_wc, train_performance_wc, on = 'ds', how = 'l
eft')
df_pg_miami_wc = pd.concat([df_pg_miami, df_wc_join])

# Los Angeles Join

m_la = Prophet(changepoint_prior_scale = 0.5, interval_width = 0.95)
df_la = df[(df.COUNTY == "Los Angeles") & (df.ds <= '2020-05-31')][["C
OUNTY", "ds", "y"]]
m_la.fit(df_la)
future_la = m_la.make_future_dataframe(periods=14)
train_performance_la = m_la.predict(df_la[["ds"]])
train_performance_la['ds'] = train_performance_la['ds'].dt.strftime('%
Y-%m-%d')
df_la_join = pd.merge(df_la, train_performance_la, on = 'ds', how = 'l
eft')
df_pg_miami_wc_la = pd.concat([df_pg_miami_wc, df_la_join])

# Bexar Join

m_bx = Prophet(changepoint_prior_scale = 0.5, interval_width = 0.95)
df_bx = df[(df.COUNTY == "Bexar") & (df.ds <= '2020-05-31')][["COUNTY"
, "ds", "y"]]
m_bx.fit(df_bx)
future_bx = m_bx.make_future_dataframe(periods=14)
train_performance_bx = m_bx.predict(df_bx[["ds"]])
train_performance_bx['ds'] = train_performance_bx['ds'].dt.strftime('%
Y-%m-%d')
df_bx_join = pd.merge(df_bx, train_performance_bx, on = 'ds', how = 'l
eft')
df_pg_miami_wc_la_bx = pd.concat([df_pg_miami_wc_la, df_bx_join])

""" # Wayne

m_wy = Prophet(changepoint_prior_scale = 0.5, interval_width = 0.95)
df_wy = df[(df.COUNTY == "Wayne") & (df.FIPS == "26163") & (df.DIABETE
S_RATE == 10.6) & (df.ds <= '2020-05-31')][["COUNTY", "FIPS", "ds", "y
"]]
m_wy.fit(df_wy)
future_bx = m_bx.make_future_dataframe(periods=14)
train_performance_wy = m_wy.predict(df_bx[["ds"]])
train_performance_wy['ds'] = train_performance_wy['ds'].dt.strftime('%
Y-%m-%d')
df_wy_join = pd.merge(df_wy, train_performance_wy, on = 'ds', how = 'l
eft')
df_pg_miami_wc_la_bx_wy = pd.concat([df_pg_miami_wc_la_bx, df_wy_join]
) """

m = Prophet(changepoint_prior_scale = 0.5, interval_width = 0.95)
```

```
m.fit(df_pg_miami_wc_la_bx_wy)
future = m.make_future_dataframe(periods=14)
train_performance = m.predict(df_pg_miami_wc_la_bx_wy[["ds"]])
train_performance['ds'] = train_performance['ds'].dt.strftime('%Y-%m-%
d')
```

I then combined the prior aggregated dataset with all my demographic data, adding an error column to keep track of actual - predicted values for cases.

I calculated the root mean square error (RMSE) for each county to gain a sense of the prediction accuracy of each model. Los Angeles had the largest RMSE value (317.48), which makes sense because cases are only starting to pick up now in that region (when they surprisingly did not in March and April).

In [404]:
```
# Aggregate Forecasting Dataset for Selected Counties

df_agg = pd.merge(df_pg_miami_wc_la_bx_wy, df, on = ['COUNTY', 'FIPS',
'ds'], how = 'left')
df_agg["error"] = df_agg.apply(lambda x: x["y_y"] - x["yhat"], axis=1)

# Root Mean Square Error for Selected Counties

print("Root Mean Square Error (RMSE) by County")
print()

rmse_pg = math.sqrt(sum((df_pg_join.y - train_performance_pg.yhat)**2)
/ len(df_pg_join.index))
print("Prince George's:", rmse_pg)

rmse_miami = math.sqrt(sum((df_miami_join.y - train_performance_miami.
yhat)**2) / len(df_miami_join.index))
print("Miami-Dade:", rmse_miami)

rmse_wc = math.sqrt(sum((df_wc_join.y - train_performance_wc.yhat)**2)
/ len(df_wc_join.index))
print("Westchester:", rmse_wc)

rmse_la = math.sqrt(sum((df_la_join.y - train_performance_la.yhat)**2)
/ len(df_la_join.index))
print("Los Angeles:", rmse_la)

rmse_bx = math.sqrt(sum((df_bx_join.y - train_performance_bx.yhat)**2)
/ len(df_bx_join.index))
print("Bexar:", rmse_bx)
```

```
Root Mean Square Error (RMSE) by County

Prince George's: 68.6101745863463
Miami-Dade: 38.10314926934416
Westchester: 57.395530220731345
Los Angeles: 317.48379025010314
Bexar: 24.268622299396277
```

I tried to run an OLS regression on the error values, but because the static data does not change on the level of individual counties, the model was useless, comparing dynamic COVID-19 data with county data that remained constant despite the changes in CONFIRMED_CASES or DEATHS.

In [446]:
```
# OLS Regression (error)

results_fc = smf.ols(formula = "error ~ E_TOTPOP + MEDIAN_INCOME + EP_
CROWD + EP_DISABL + EP_GROUPQ + EP_NOHSDP + EP_UNINSUR + RPL_THEMES +
PERCENT_DEM + DIABETES_RATE + SMOKING_RATE + PERCENT_HISPANIC + PERCEN
T_BLACK + PERCENT_ASIAN + PERCENT_AMERINDIAN + PERCENT_OTHER_RACE + PE
RCENT_MULTIRACIAL + PERCENT_NONCITIZEN + MALE_0_TO_17 + FEMALE_0_TO_17
+ MALE_18_TO_29 + FEMALE_18_TO_29 + FEMALE_30_TO_49 + MALE_OVER_50 + F
EMALE_OVER_50", data=df_agg).fit()
print(results_fc.summary())
```

```
                            OLS Regression Results
==============================================================================
==========
Dep. Variable:                    error   R-squared:
-0.000
Model:                            OLS   Adj. R-squared:
-0.000
Method:                 Least Squares   F-statistic:
nan
Date:                Thu, 16 Jul 2020   Prob (F-statistic):
nan
Time:                        00:55:49   Log-Likelihood:
-372.47
No. Observations:                  71   AIC:
746.9
Df Residuals:                      70   BIC:
749.2
Df Model:                           0
Covariance Type:            nonrobust
==============================================================================
==================
                           coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
------------------
Intercept           -3.933e-05      0.003     -0.013      0.990
-0.006       0.006
E_TOTPOP              -0.0006      0.042     -0.013      0.990
-0.085       0.084
MEDIAN_INCOME         -0.0004      0.033     -0.013      0.990
-0.067       0.066
EP_CROWD            -8.652e-05      0.007     -0.013      0.990
-0.013       0.013
EP_DISABL             -0.0006      0.048     -0.013      0.990
-0.097       0.096
EP_GROUPQ           -5.112e-05      0.004     -0.013      0.990
-0.008       0.008
EP_NOHSDP             -0.0006      0.043     -0.013      0.990
```

```
-0.085       0.084
EP_UNINSUR             -0.0003      0.022     -0.013      0.990
-0.044       0.043
RPL_THEMES           -3.43e-05      0.003     -0.013      0.990
-0.005       0.005
PERCENT_DEM          -2.635e-05     0.002     -0.013      0.990
-0.004       0.004
DIABETES_RATE          -0.0004      0.032     -0.013      0.990
-0.064       0.063
SMOKING_RATE           -0.0012      0.089     -0.013      0.990
-0.179       0.177
PERCENT_HISPANIC    -2.304e-06      0.000     -0.013      0.990
-0.000       0.000
PERCENT_BLACK       -1.522e-05      0.001     -0.013      0.990
-0.002       0.002
PERCENT_ASIAN       -1.275e-06   9.77e-05     -0.013      0.990
-0.000       0.000
PERCENT_AMERINDIAN  -1.004e-07    7.7e-06     -0.013      0.990   -1
.55e-05    1.53e-05
PERCENT_OTHER_RACE  -1.016e-07   7.79e-06     -0.013      0.990   -1
.56e-05    1.54e-05
PERCENT_MULTIRACIAL -8.149e-07   6.25e-05     -0.013      0.990
-0.000       0.000
PERCENT_NONCITIZEN  -1.567e-06      0.000     -0.013      0.990
-0.000       0.000
MALE_0_TO_17         -4.78e-06      0.000     -0.013      0.990
-0.001       0.001
FEMALE_0_TO_17      -4.605e-06      0.000     -0.013      0.990
-0.001       0.001
MALE_18_TO_29       -3.241e-06      0.000     -0.013      0.990
-0.000       0.000
FEMALE_18_TO_29     -3.278e-06      0.000     -0.013      0.990
-0.001       0.000
FEMALE_30_TO_49     -5.048e-06      0.000     -0.013      0.990
-0.001       0.001
MALE_OVER_50         -6.23e-06      0.000     -0.013      0.990
-0.001       0.001
FEMALE_OVER_50      -7.471e-06      0.001     -0.013      0.990
-0.001       0.001
==============================================================================
==========
Omnibus:                        1.054   Durbin-Watson:
2.684
Prob(Omnibus):                  0.590   Jarque-Bera (JB):
0.627
Skew:                          -0.218   Prob(JB):
0.731
Kurtosis:                       3.147   Cond. No.
2.06e+17
==============================================================================
```

```
==========

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.
[2] The smallest eigenvalue is 3.04e-30. This might indicate that th
ere are
strong multicollinearity problems or that the design matrix is singu
lar.
```

I then created visualizations of each model's predictions. In the plots, the black points represent actual data, the red line represents the trendline based on that data, and the blue space represents the uncertainty of the prediction.

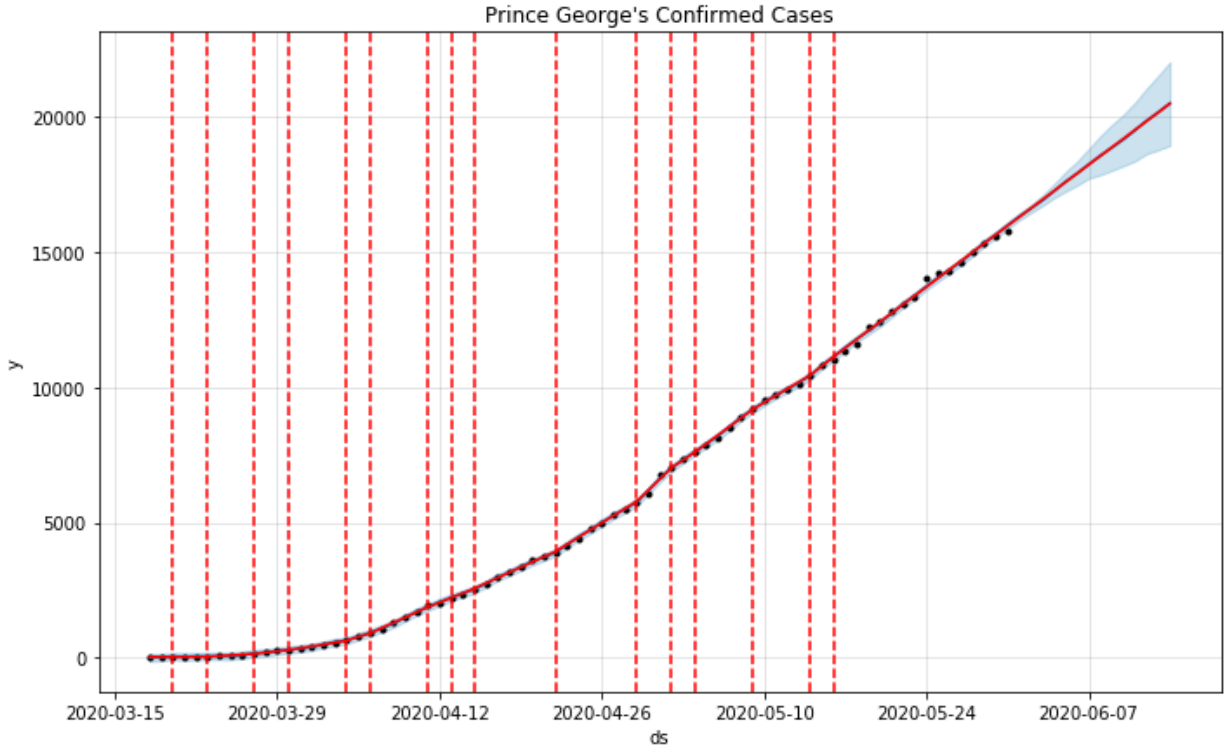
Interestingly, many of the counties displayed have cases reaching a minimum on Wednesdays.

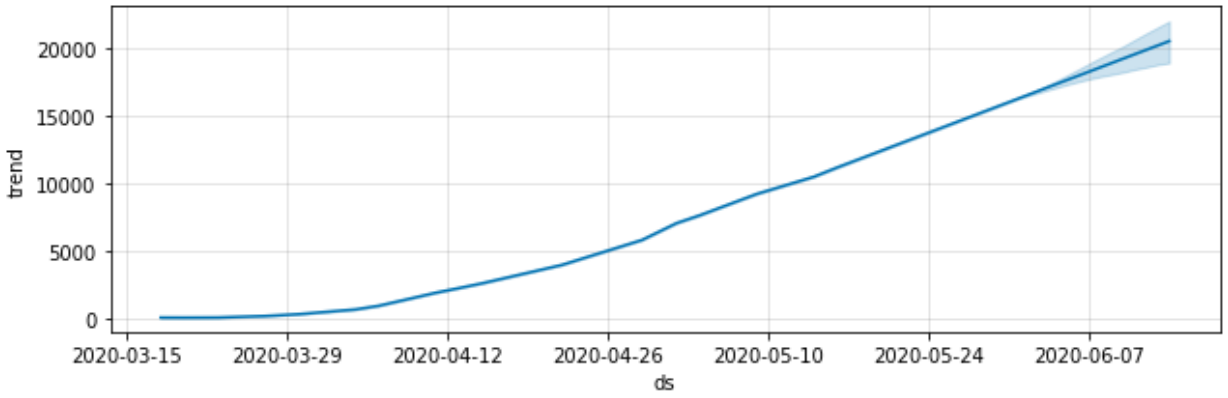
There seem to be different archetypes of counties: generally, those whose plots are concave and those whose are convex. Westchester, for example, is concave while Prince George's and Los Angeles are both convex.

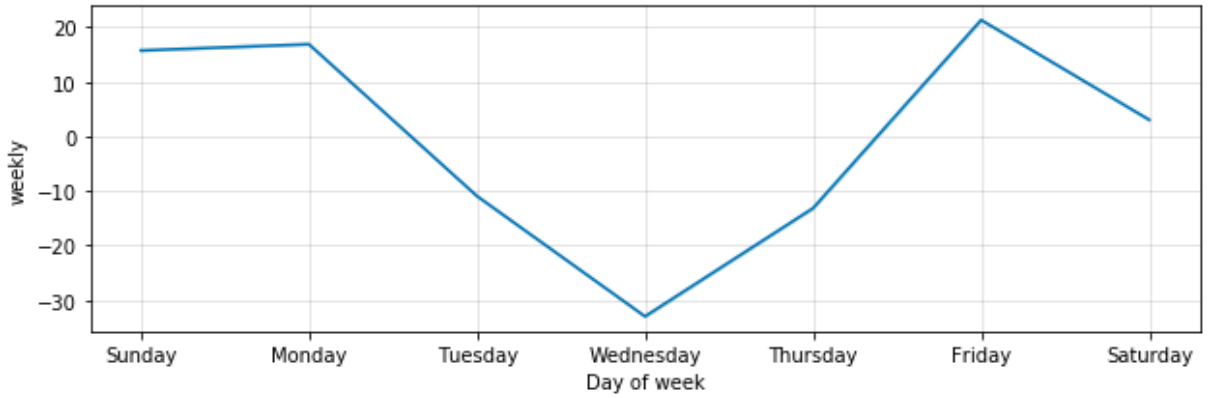
It would be interesting to try to approximate the trendlines on these models to functions and to subsequently take their derivatives to determine the functions' concavities, which could be an interesting way to forecast.

In [405]:
```
# Prince George's Forecast

m_pg = Prophet(changepoint_prior_scale = 0.5, interval_width = 0.95)
df_pg = df[(df.COUNTY == "Prince George's") & (df.ds <= '2020-05-31')]
[["COUNTY", "ds", "y"]]
m_pg.fit(df_pg)
future_pg = m_pg.make_future_dataframe(periods=14)
test_performance_pg = m_pg.predict(future_pg)
fig_pg = m_pg.plot(test_performance_pg)
a_pg = add_changepoints_to_plot(fig_pg.gca(), m_pg, test_performance_p
g)
plt.title("Prince George's Confirmed Cases")
fig_pg_2 = m_pg.plot_components(test_performance_pg)
```

```
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly
_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_s
easonality=True to override this.
```

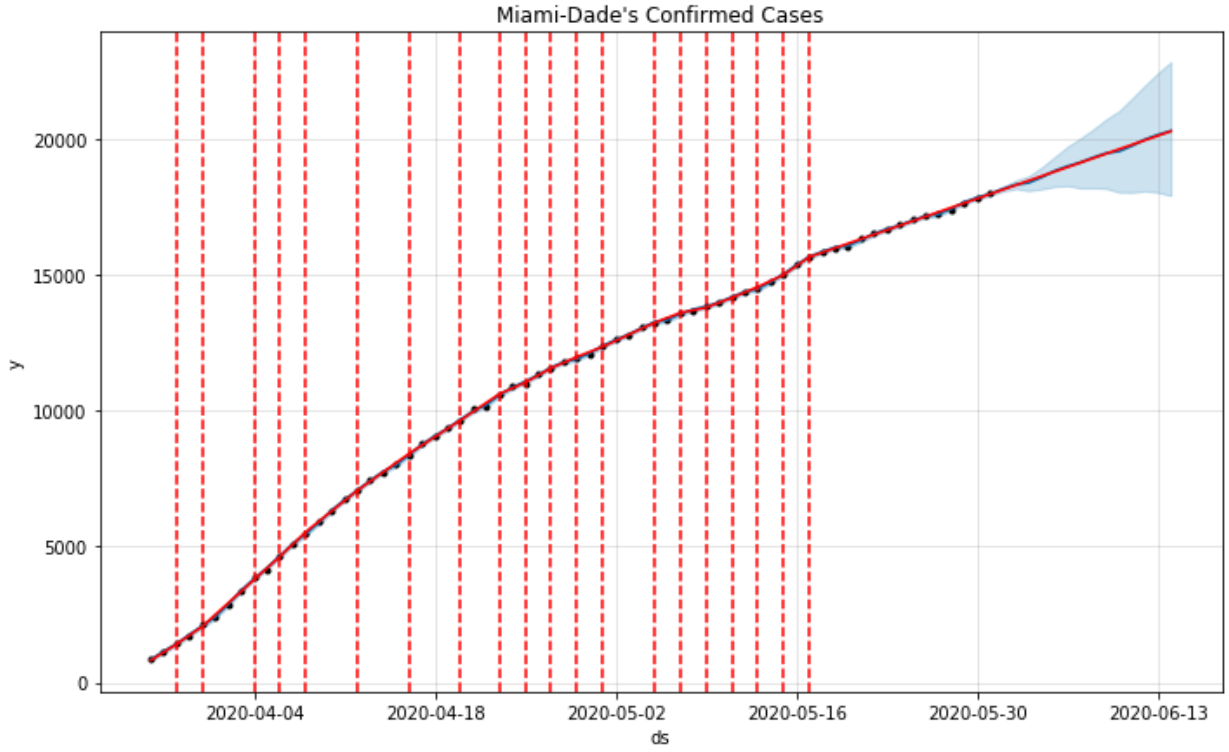

Prince George's Confirmed Cases
20000
15000
10000
5000
0
y
2020-03-15
2020-03-29
2020-04-12
2020-04-26
2020-05-10
2020-05-24
2020-06-07
ds

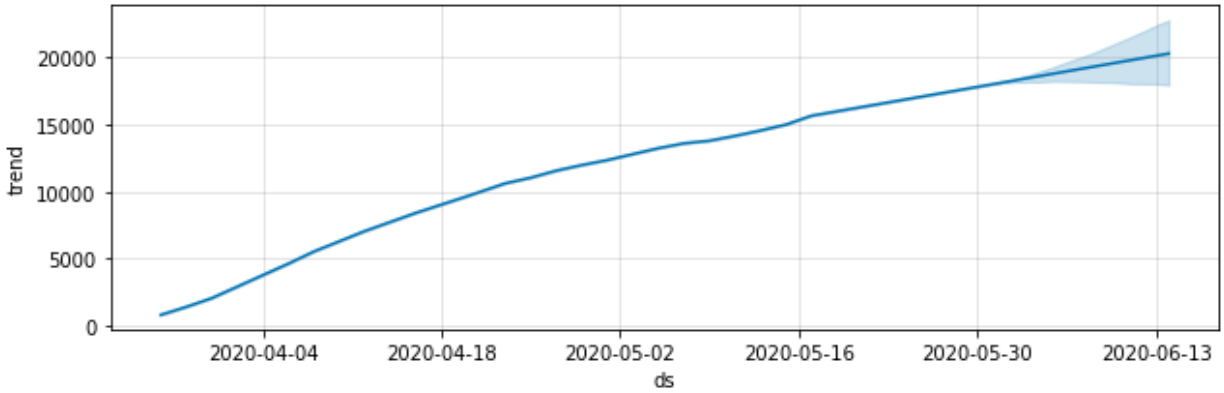

20000
15000
10000
5000
0
trend
2020-03-15
2020-03-29
2020-04-12
2020-04-26
2020-05-10
2020-05-24
2020-06-07
ds

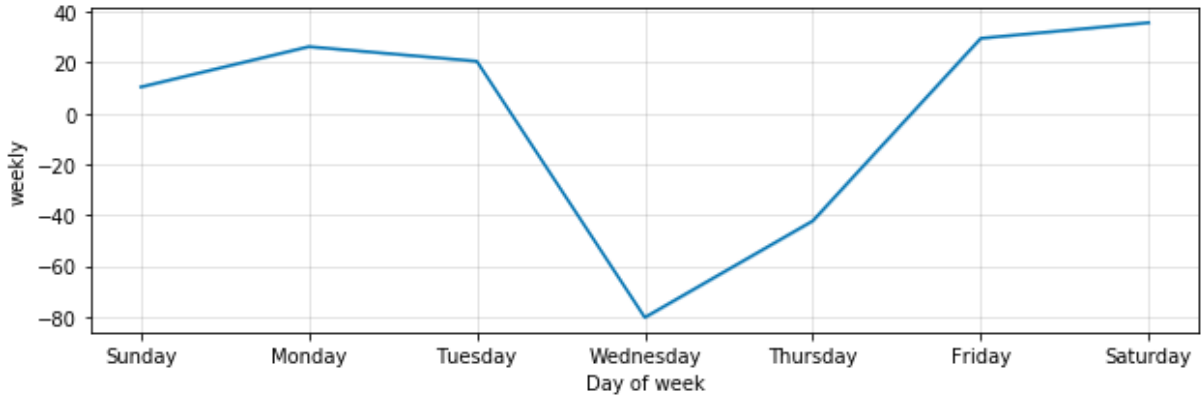

20
10
0
−10
−20
−30
weekly
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of week

In [368]:
```
# Miami-Dade Forecast

m_miami = Prophet(changepoint_prior_scale = 0.5, interval_width = 0.95
)
df_miami = df[(df.COUNTY == "Miami-Dade") & (df.ds <= '2020-05-31')][[
"COUNTY", "ds", "y"]]
m_miami.fit(df_miami)
future_miami = m_miami.make_future_dataframe(periods=14)
test_performance_miami = m_miami.predict(future_miami)
fig_miami = m_miami.plot(test_performance_miami)
a_miami = add_changepoints_to_plot(fig_miami.gca(), m_miami, test_perf
ormance_miami)
plt.title("Miami-Dade's Confirmed Cases")
fig_miami_2 = m_miami.plot_components(test_performance_miami)
```

```
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly
_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_s
easonality=True to override this.
```

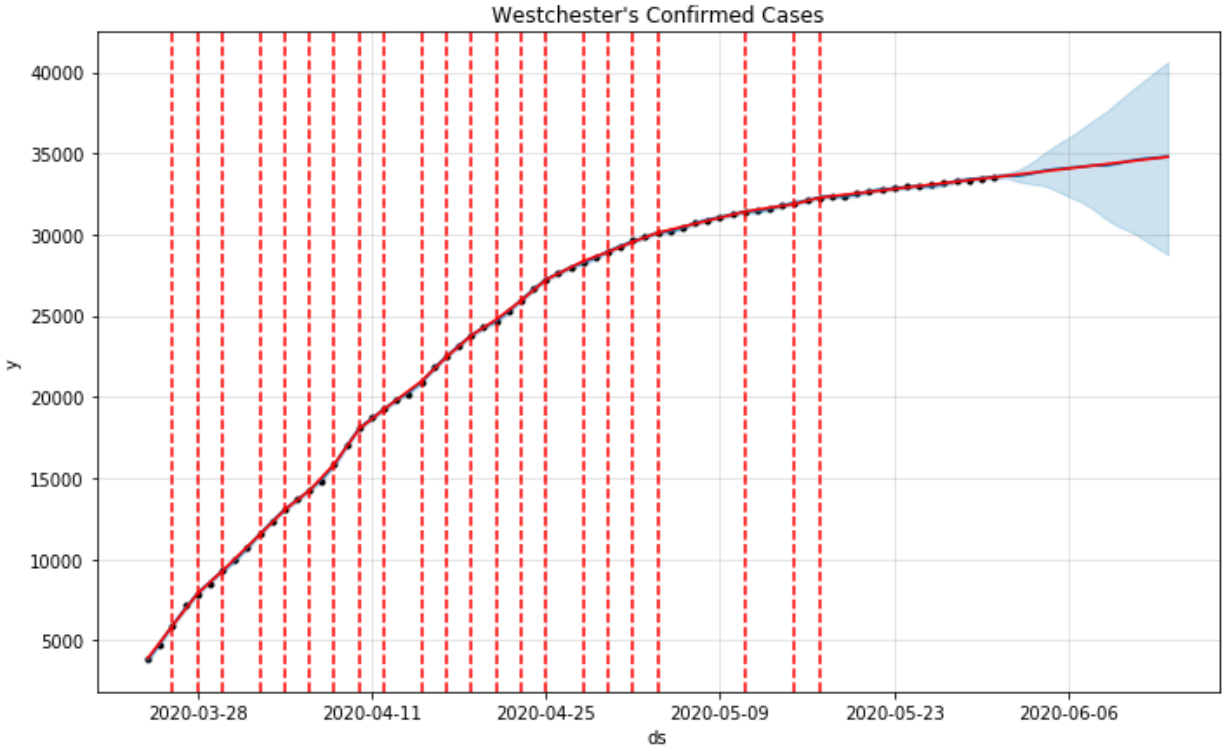


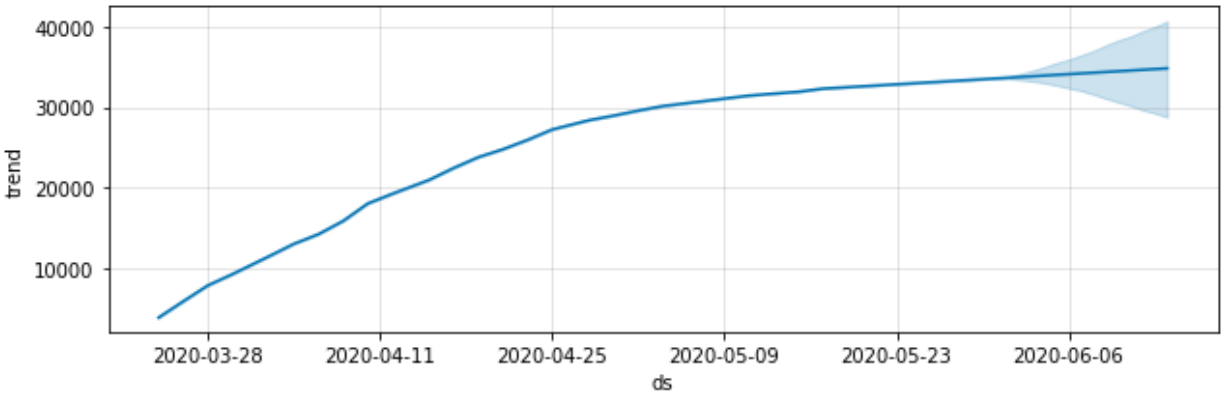


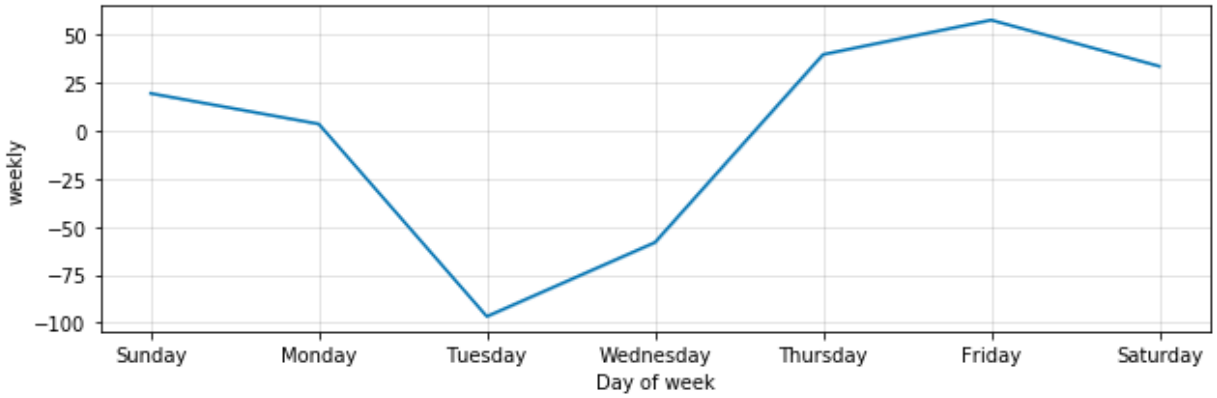


In [369]:
```
# Westchester Forecast

m_wc = Prophet(changepoint_prior_scale = 0.5, interval_width = 0.95)
df_wc = df[(df.COUNTY == "Westchester") & (df.ds <= '2020-05-31')][["C
OUNTY", "ds", "y"]]
m_wc.fit(df_wc)
future_wc = m_wc.make_future_dataframe(periods=14)
test_performance_wc = m_wc.predict(future_wc)
fig_wc = m_wc.plot(test_performance_wc)
a_wc = add_changepoints_to_plot(fig_wc.gca(), m_wc, test_performance_w
c)
plt.title("Westchester's Confirmed Cases")
fig_wc_2 = m_wc.plot_components(test_performance_wc)
```

```
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly
_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_s
easonality=True to override this.
```

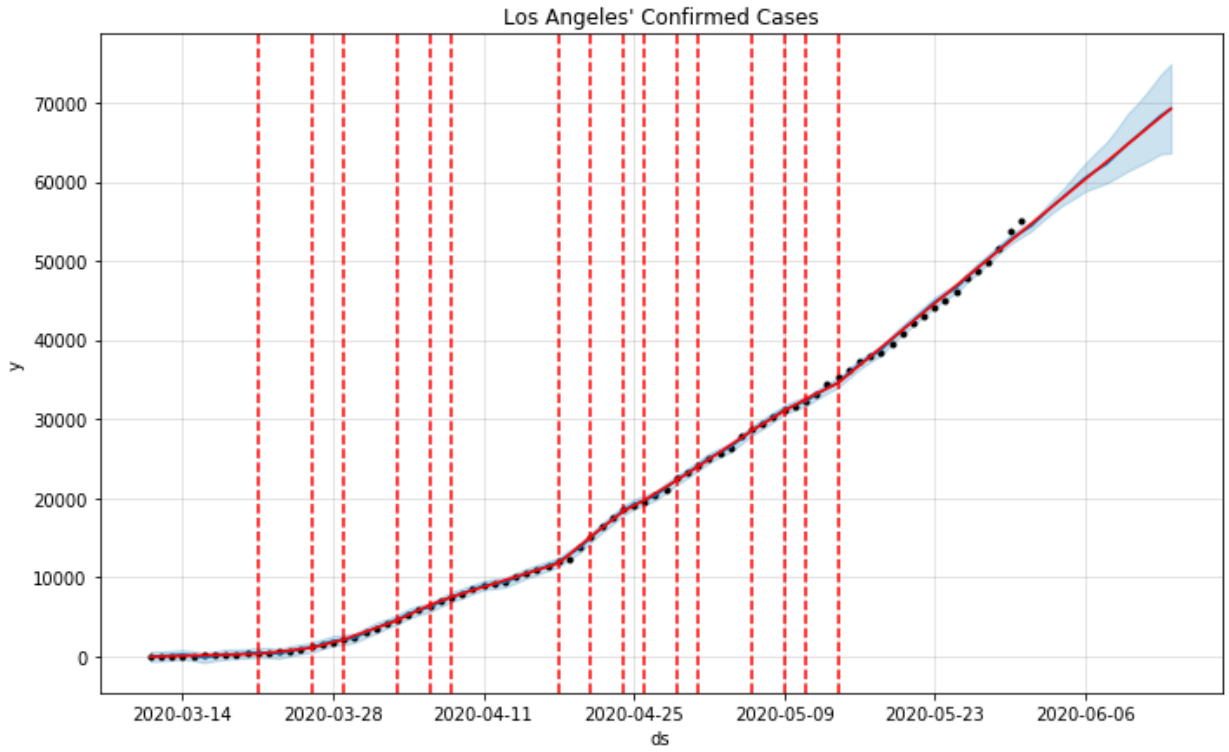

Westchester's Confirmed Cases
40000
35000
30000
25000
20000
15000
10000
5000
y
2020-03-28
2020-04-11
2020-04-25
2020-05-09
2020-05-23
2020-06-06
ds

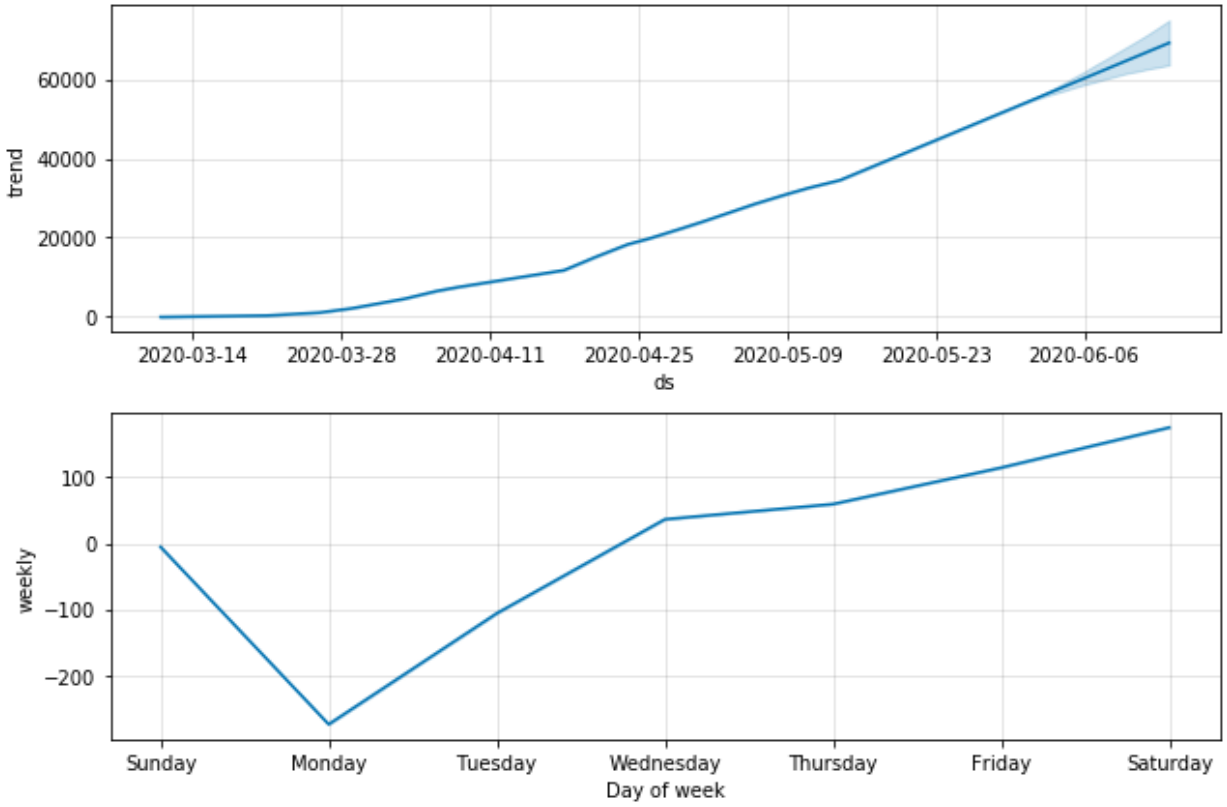

40000
30000
20000
10000
trend
2020-03-28
2020-04-11
2020-04-25
2020-05-09
2020-05-23
2020-06-06
ds


50
25
0
−25
−50
−75
−100
weekly
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of week

In [370]:
```
# Los Angeles Forecast

m_la = Prophet(changepoint_prior_scale = 0.5, interval_width = 0.95)
df_la = df[(df.COUNTY == "Los Angeles") & (df.ds <= '2020-05-31')][["C
OUNTY", "ds", "y"]]
m_la.fit(df_la)
future_la = m_la.make_future_dataframe(periods=14)
test_performance_la = m_la.predict(future_la)
fig_la = m_la.plot(test_performance_la)
a_la = add_changepoints_to_plot(fig_la.gca(), m_la, test_performance_l
a)
plt.title("Los Angeles' Confirmed Cases")
fig_la_2 = m_la.plot_components(test_performance_la)
```

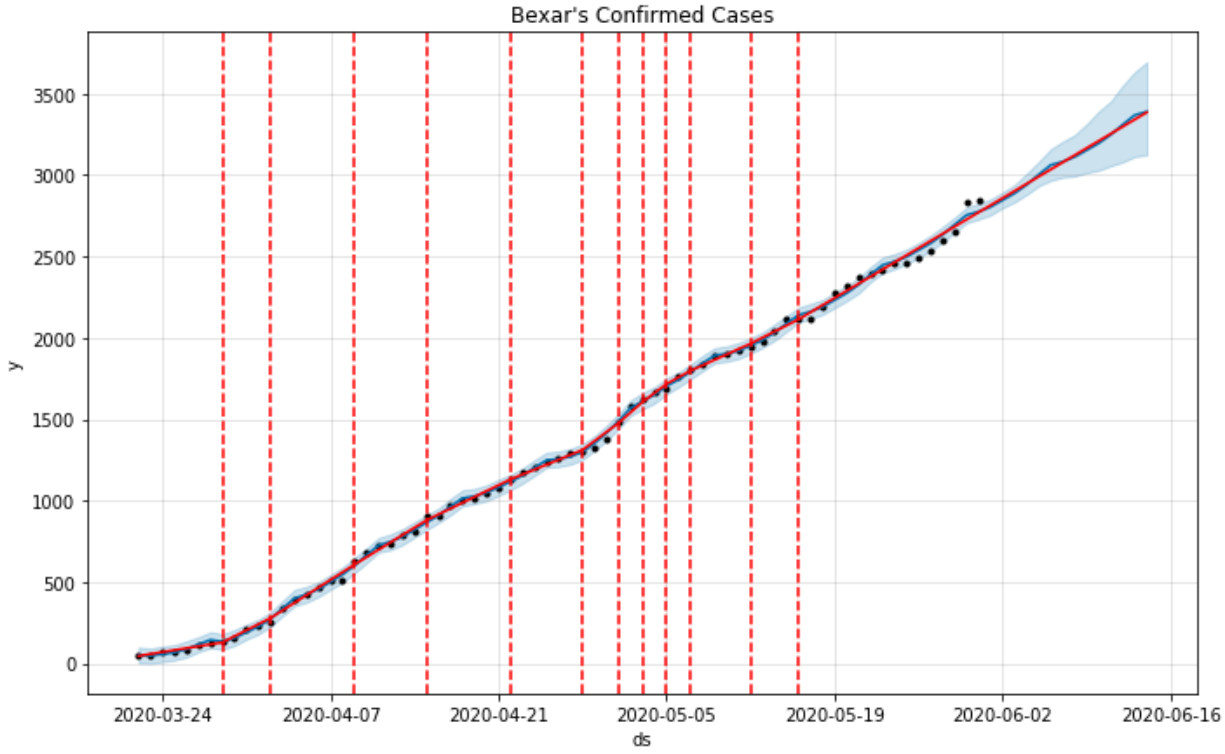
```
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly
_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_s
easonality=True to override this.
```

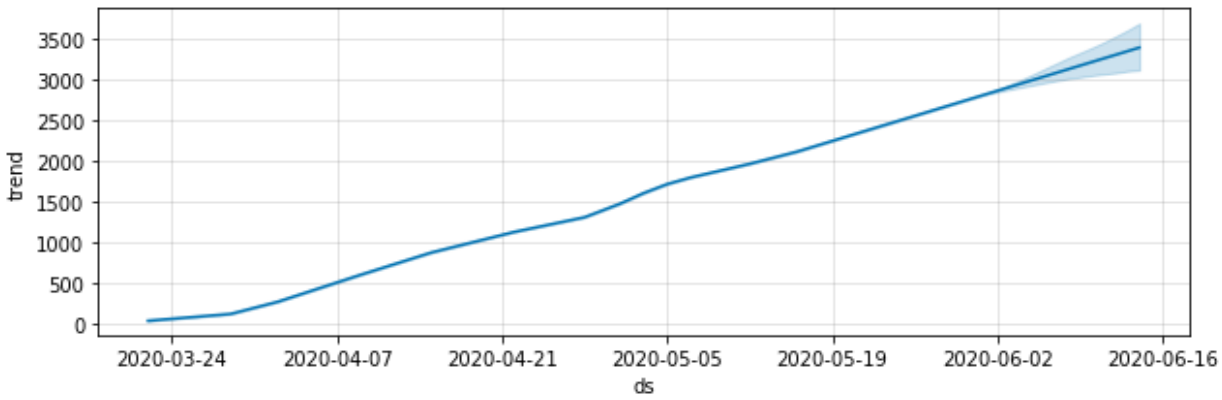


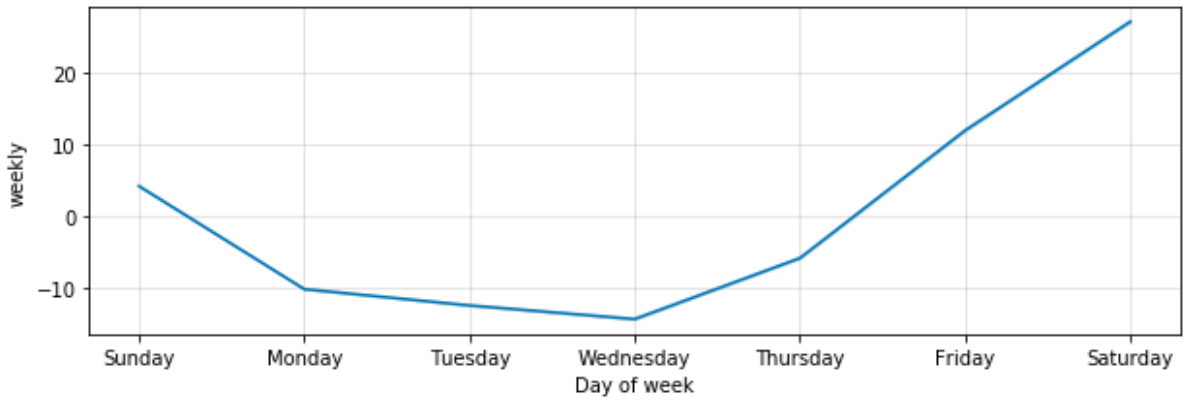


In [372]:
```
# Bexar Forecast

m_bx = Prophet(changepoint_prior_scale = 0.5, interval_width = 0.95)
df_bx = df[(df.COUNTY == "Bexar") & (df.ds <= '2020-05-31')][["COUNTY"
, "ds", "y"]]
m_bx.fit(df_bx)
future_bx = m_bx.make_future_dataframe(periods=14)
test_performance_bx = m_bx.predict(future_bx)
fig_bx = m_bx.plot(test_performance_bx)
a_bx = add_changepoints_to_plot(fig_bx.gca(), m_bx, test_performance_b
x)
plt.title("Bexar's Confirmed Cases")
fig_bx_2 = m_bx.plot_components(test_performance_bx)
```

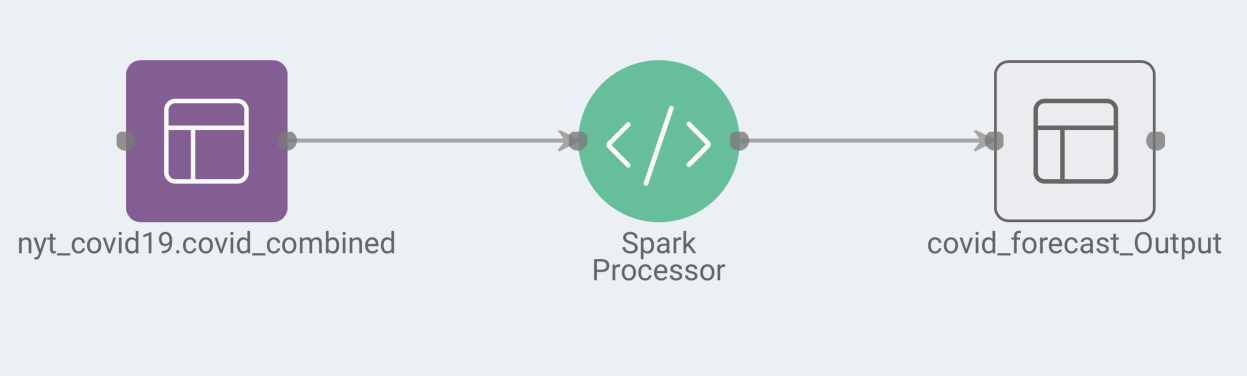
```
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly
_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_s
easonality=True to override this.
```


Bexar's Confirmed Cases
y
ds
0
500
1000
1500
2000
2500
3000
3500
2020-03-24
2020-04-07
2020-04-21
2020-05-05
2020-05-19
2020-06-02
2020-06-16

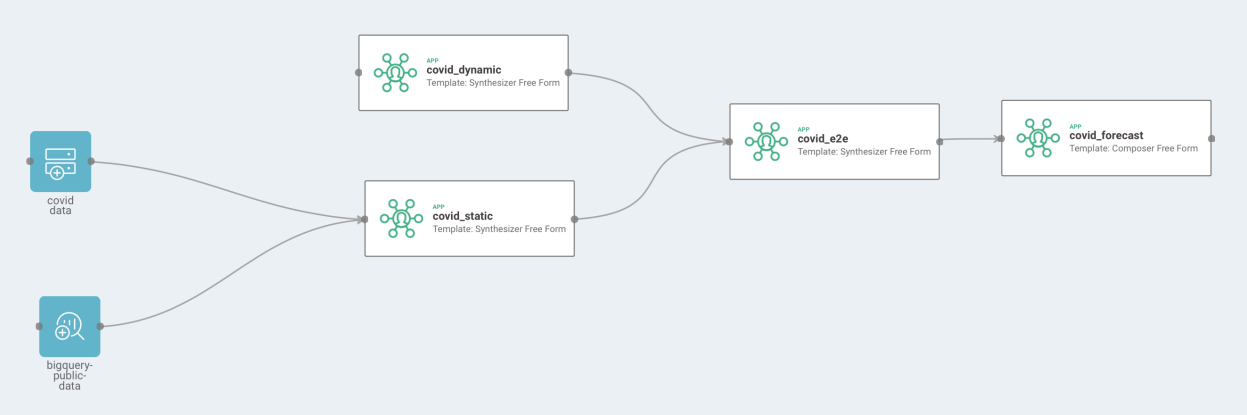

trend
ds
0
500
1000
1500
2000
2500
3000
3500
2020-03-24
2020-04-07
2020-04-21
2020-05-05
2020-05-19
2020-06-02
2020-06-16


weekly
Day of week
−10
0
10
20
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday

In Syntasa, I created an app for recursively creating forecasting models for all counties in the dataset (not just the few I selected in Jupyter). Here is the workflow:



I also combined all my individual apps into one collective workflow using Syntasa, making it easier to visualize:



Conclusion

From an insights standpoint, total population was the variable that had the most effect on cases and deaths, and as a result, many of the variables that were highly correlated with cases and deaths (e.g. PERCENT_DEM, PERCENT_ASIAN) were not very impactful in the descriptive models (since they were masked by E_TOTPOP). Looking at cases and deaths per capita was an

effective way to visualize and interpret the data in Data Studio, and it was also necessary to scale everything by population in descriptive modeling. Making this adjustment made it clear that cases in previous days had the highest impact on CC_PER_CAPITA in the OLS regression, which is why I decided to focus only on cases in my forecasting model; I was unable to bring in anything else from the descriptive models anyways because that data was all static.

I was able to use the Syntasa app to create efficient apps and to visualize everything within interactive workflows. It was an effective tool throughout my project and made the connections between different platforms seamless.

There are many elements of the project that I would enhance if given more time. It would be interesting to look at a logistic regression as well because all the variables range from zero to one. It would also be interesting to split each of my descriptive models into two: one with a date range from the beginning of the pandemic to 05/31/20 and the other from 06/01/20 onwards. Although there technically has not been a second wave of the virus, the pandemic has certainly shifted in recent weeks to communities that were previously unaffected. I would also consider looking at different datasets in the future, particularly hypertension rates by county and dynamic racial data specific to those who contract and die from COVID-19.

Thanks again to Adam Neo for teaching me the tools I needed to complete this project!