DataEng: Data Integration Activity

This week you will gain hands-on experience with Data Integration by combining data from two distinct sources into a unified DataFrame for analysis.

Submit: Make a copy of this document and use it to record your results. Store a PDF copy of the document in your git repository along with any needed code before submitting for this week.

Your job is to integrate **county-level COVID-19 data** with the **ACS Census Tract data for 2017** to build a model that allows you to relate COVID numbers with economic data such as population, per capita income and poverty level. To do this you should build a pandas DataFrame that has a row per USA county (there are more than 3000 counties in the USA) and includes the following columns:

County - name of the county
State - name of the state in which the county resides
TotalCases - total number of COVID cases for this county as of February 20, 2021
Dec2020Cases - number of COVID cases recorded in this county in December of 2020
TotalDeaths - total number of COVID deaths for this county as of February 20, 2021
Dec2020Deaths - number of COVID deaths recorded in this county in December of 2020
Population - population of this county
Poverty - % of people in poverty in this county
PerCapitaIncome - per capita personal income for this county

We hope that you make it all the way through to the end. Regardless, use your time wisely to gain python programming experience and learn as much as you can about building integrated multi-source data models using python and pandas.

For this activity you should use whichever environment is convenient for you to develop with python 3 and pandas. You are not required to use GCP, but you can use it if you prefer.

Submit: In-class Activity Submission Form

A. Aggregate Census Data to County Level

Your integration will use two different dimensions: location (as indicated by state and county) and time. You should greatly simplify your processing and reduce your time by pre-processing your data along each of these dimensions.

The ACS data is separated into "Census Tracts" which are regions within counties that correspond to groups of approximately 4000 people. The Census Bureau defines these

to help organize the actual job of collecting census data, but this grouping can make your Data Engineering job more more challenging. This level of detail is not needed for your county-level analysis, and you can greatly decrease your efforts by aggregating per-tract data to the county level.

Create a python program that produces a one-row-per-county version of the ACS data set. To do this you will need to think about how to properly aggregate Census Tract-level data into County-level summaries.

In this step you can also eliminate unneeded columns from the ACS data.

Question: Show your aggregated county-level data rows for the following counties: Loudoun County Virginia, Washington County Oregon, Harlan County Kentucky, Malheur County Oregon
Results:

State	County	TotalPop	IncomePerCap	Poverty
Virginia	Loudoun	374558	50391.015625	3.884375
Oregon	Washington	572071	34970.817308	10.446154
Kentucky	Harlan	27548	16010.363636	33.318182
Oregon	Malheur	30421	17966.428571	24.414286

B. Simplify the COVID Data

You can simplify the COVID data along the time dimension. The COVID data set contains day-level resolution data from (approximately) March of 2020 through February of 2021. However, you will only need four data points per county: total cases, total deaths, cases reported during December of 2020 and deaths reported during December 2020.

Create a python program that reduces the COVID data to one line per county.

Question: Show your simplified COVID data for the counties listed above.

state	county	cases	deaths	cases_dec	deaths_dec
Virginia	Loudoun	22557	199.0	14169.0	159.0
Oregon	Washington	20866	209.0	16070.0	142.0
Kentucky	Harlan	2352	68.0	1538.0	18.0
Oregon	Malheur	3331	58.0	2914.0	50.0

C. Integrate COVID Data with ACS Data

Create a single pandas DataFrame containing one row per county and using the columns described above. You are free to add additional columns if needed. For example, you might want to normalize all of the COVID data by the population of each county so that you have a consistent "number of cases/deaths per 100000 residents" value for each county.

Question: List your integrated data for all counties in the State of Oregon.

state	county	TotalPop	IncomePerCap	Poverty	cases	deaths	cases_dec	deaths_dec	TotalCases	TotalDeaths	Totalcases	Totaldeaths
Alabama	Autauga	55036	26588.166667	14.558333	6092.0	85.0	4190.0	48.0	11069.118395	154.444364	11069.118395	154.444364
	Baldwin	203360	29130.709677	12.874194	19392.0	262.0	13601.0	161.0	9535.798584	128.835563	9535.798584	128.835563
	Barbour	26201	17891.666667	27.755556	2067.0	50.0	1514.0	32.0	7889.011870	190.832411	7889.011870	190.832411
	Bibb	22580	21799.000000	13.925000	2414.0	58.0	1834.0	46.0	10690.876882	256.864482	10690.876882	256.864482
	Blount	57667	21598.444444	16.422222	6040.0	125.0	4641.0	63.0	10473.927896	216.761753	10473.927896	216.761753

D. Analysis

For each of the following, determine the strength of the correlation between each pair of variables. Compute the correlation strength by calculating the Pearson correlation coefficient R for pairs of columns in your DataFrame. For example, if you have a DataFrame df with each row representing a distinct county, and columns named 'TotalCases' and 'Poverty', then you can compute R like this:

```
R = df['TotalCases'].corr(df['Poverty'])
```

For any R that is > 0.5 or < -0.5 also display a scatter plot (see pandas scatterplot and seaborn documentation for information about how to display scatter plots from DataFrame data).

The COVID numbers should be normalized to population (# of cases per 100,000 residents) so that different sized counties are comparable. So for example, "COVID total cases" below really means "((COVID total cases in county * 100000) / population of county)".

1. Across all of the counties in the State of Oregon
   a. COVID total cases vs. % population in poverty
   b. COVID total deaths vs. % population in poverty
   c. COVID total cases vs. Per Capita Income level
   d. COVID total cases vs. Per Capita Income level
   e. COVID cases during December 2020 vs. % population in poverty
   f. COVID deaths during December 2020 vs. % population in poverty
   g. COVID cases during December 2020 vs. Per Capita Income level
   h. COVID cases during December 2020 vs. Per Capita Income level

Oregondata R values are:

```
R for all corelations respectively 0.19626280807425306
R for all corelations respectively 0.24095099055190802
R for all corelations respectively -0.38298538944796284
R for all corelations respectively -0.5103336402143774
R for all corelations respectively -0.15295719081094342
R for all corelations respectively -0.08031013161838976
R for all corelations respectively 0.5662321434044019
R for all corelations respectively 0.4999023143312516
```

Scatter plots are for only R >0.5 and R<-0.5
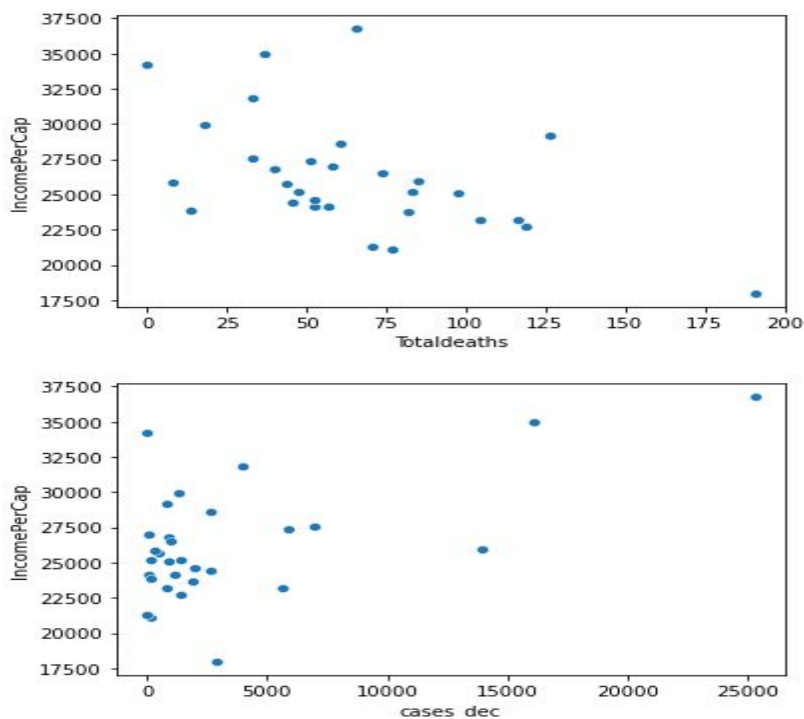
2. Across all of the counties in the entire USA
    a. COVID total cases vs. % population in poverty
    b. COVID total deaths vs. % population in poverty
    c. COVID total cases vs. Per Capita Income level
    d. COVID total cases vs. Per Capita Income level
    e. COVID cases during December 2020 vs. % population in poverty
    f. COVID deaths during December 2020 vs. % population in poverty
    g. COVID cases during December 2020 vs. Per Capita Income level
    h. COVID cases during December 2020 vs. Per Capita Income level

R for all other states:

```
R for all corelations respectively 0.12022994647462465
R for all corelations respectively 0.200497200208987
R for all corelations respectively -0.20541255201485065
R for all corelations respectively -0.23946027915501594
R for all corelations respectively -0.018326019014729845
R for all corelations respectively -0.017508685701395826
R for all corelations respectively 0.18771428825410538
R for all corelations respectively 0.2201914717647978
```

Note that this exercise does not constitute a competent, thorough statistical analysis of the relationships between immunological data and demographic data. It is just an illustration of the types of computations that might be accomplished with an integrated data set.