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: Loudon County Virginia, Washington County Oregon, Harlan County Kentucky, Malheur County Oregon

2968	Virginia	Loudoun County	248.6	374558	50391.015625
2241	Oregon	Washington County	1086.4	572071	34970.817308
1040	Kentucky	Harlan County	366.5	27548	16010.363636
2230	Oregon	Malheur County	170.9	30421	17966.428571

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	Total Cases 02/20/2021	Total Deaths 02/20/2021	Dec Cases	Dec Deaths
0	Alabama	Autauga	6092.0	85.0	96193.0	1212.0
1	Alabama	Baldwin	19392.0	262.0	308290.0	4025.0
2	Alabama	Barbour	2067.0	50.0	36285.0	835.0
3	Alabama	Bibb	2414.0	58.0	41566.0	1110.0
4	Alabama	Blount	6040.0	125.0	107510.0	1406.0
5	Alabama	Bullock	1149.0	33.0	20809.0	557.0

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	Population	Poverty%	PerCapitalIncome	Total Cases 02/20/2021	Total Deaths 02/20/2021	Dec Cases	Dec Deaths
2208	Oregon	Baker County	15980	15.083855	154241.0	629.0	7.0	25078.0	358.0
2209	Oregon	Benton County	88249	22.421152	538668.0	2248.0	16.0	82890.0	1576.0
2210	Oregon	Clackamas County	399962	8.976120	3000217.0	13196.0	172.0	641107.0	12255.0
2211	Oregon	Clatsop County	38021	12.190090	311931.0	766.0	6.0	40368.0	38.0
2212	Oregon	Columbia County	50207	12.315329	281097.0	1208.0	21.0	48525.0	375.0
2213	Oregon	Coos County	62921	17.896488	344348.0	1347.0	18.0	44453.0	169.0
2214	Oregon	Crook County	21717	15.320864	95834.0	765.0	18.0	21778.0	463.0
2215	Oregon	Curry County	22377	15.408656	134496.0	394.0	6.0	12280.0	140.0
2216	Oregon	Deschutes County	175321	12.100898	764025.0	5839.0	58.0	236106.0	2019.0
2217	Oregon	Douglas County	107576	17.025995	554588.0	2312.0	51.0	78247.0	1527.0
2218	Oregon	Gilliam County	1910	9.900000	24178.0	53.0	1.0	2047.0	22.0
2219	Oregon	Grant County	7209	13.635802	47710.0	221.0	1.0	7572.0	40.0
2220	Oregon	Harney County	7195	17.528770	50349.0	266.0	6.0	6747.0	39.0
2221	Oregon	Hood River County	22938	12.123145	116712.0	1057.0	29.0	55308.0	222.0
2222	Oregon	Jackson County	212070	16.858350	1120480.0	8115.0	108.0	329711.0	2258.0
2223	Oregon	Jefferson County	22707	20.694856	136138.0	1918.0	27.0	107683.0	1351.0
2224	Oregon	Josephine County	84514	18.646376	386865.0	2266.0	48.0	56433.0	763.0
2225	Oregon	Klamath County	66018	18.688624	474248.0	2752.0	54.0	91859.0	663.0
2226	Oregon	Lake County	7807	20.139311	42243.0	373.0	6.0	11036.0	70.0
2227	Oregon	Lane County	363471	19.230471	2368975.0	10033.0	121.0	384942.0	4482.0
2228	Oregon	Lincoln County	47307	18.376280	455726.0	1120.0	19.0	98783.0	2172.0
2229	Oregon	Linn County	121074	16.063929	513507.0	3533.0	55.0	153510.0	3492.0
2230	Oregon	Malheur County	30421	24.298225	125765.0	3331.0	58.0	282735.0	4856.0
2231	Oregon	Marion County	330453	16.128516	1502424.0	18171.0	280.0	1083514.0	20753.0
2232	Oregon	Morrow County	11153	14.699050	46343.0	1031.0	13.0	88255.0	876.0
2233	Oregon	Multnomah County	788459	16.474668	6245725.0	31526.0	516.0	1815061.0	33855.0
2234	Oregon	Polk County	79666	15.639958	295607.0	2978.0	42.0	131070.0	3403.0
2235	Oregon	Sherman County	1635	13.700000	34226.0	52.0	0.0	3307.0	0.0
2236	Oregon	Tillamook County	25840	15.512717	206446.0	403.0	2.0	14617.0	0.0
2237	Oregon	Umatilla County	76736	17.825222	348007.0	7580.0	80.0	565929.0	6899.0
2238	Oregon	Union County	25810	17.618597	212071.0	1264.0	19.0	99928.0	618.0
2239	Oregon	Wallowa County	6864	13.748776	80829.0	142.0	4.0	7407.0	272.0
2240	Oregon	Wasco County	25687	13.670818	200718.0	1218.0	25.0	62095.0	1786.0
2241	Oregon	Washington County	572071	10.321202	3636965.0	20866.0	209.0	1134747.0	12829.0
2242	Oregon	Wheeler County	1415	20.600000	21268.0	22.0	1.0	378.0	0.0
2243	Oregon	Yamhill County	102366	13.802658	485841.0	3716.0	62.0	180607.0	3335.0

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

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

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.