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

		TotalPop	Poverty	IncomePerCap
state	county			
Virginia	Loudoun	374558	3.884375	50391.015625
Oregon	Washington	572071	10.446154	34970.817308
Kentucky	Harlan	27548	33.318182	16010.363636
Oregon	Malheur	30421	24.414286	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	cases	deaths	dec2020Deaths	Dec_Cases
Virginia	Loudoun	2496450	35820.0	4729.0	376223
Oregon	Washington	2157339	22455.0	3860.0	424620
Kentucky	Harlan	205984	3994.0	506.0	38959
Oregon	Malheur	453634	7770.0	1465.0	82916

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	Poverty	IncomePerCap	cases	deaths	dec2020Deaths	Dec_Cases
Oregon	Baker	15980	15.000000	25706.833333	55586	663.0	133.0	11688
	Benton	88249	23.644444	29926.000000	180225	2304.0	278.0	34260
	Deschutes	175321	12.208333	31834.375000	509974	4141.0	563.0	102490
	Douglas	107576	16.731818	25208.545455	174952	3983.0	964.0	37590
	Gilliam	1910	9.900000	24178.000000	4691	76.0	25.0	898
	Grant	7209	15.850000	23855.000000	18551	94.0	31.0	4895
	Harney	7195	16.300000	25174.500000	17024	291.0	34.0	3717
	Hood River	22938	12.150000	29178.000000	107383	1444.0	216.0	19348
	Jackson	212070	17.882927	27328.780488	713288	7221.0	1655.0	154535
	Jefferson	22707	20.316667	22689.666667	200346	2630.0	409.0	36278
	Josephine	84514	19.131250	24179.062500	153675	2638.0	407.0	27180
	Klamath	66018	18.930000	23712.400000	224256	2857.0	373.0	45118
	Lake	7807	19.200000	21121.500000	25357	348.0	76.0	5358
	Lane	363471	18.529070	27546.220930	850956	10372.0	2215.0	178816
	Lincoln	47307	17.623529	26807.411765	153979	3117.0	502.0	24041
	Linn	121074	16.923810	24452.714286	324636	5949.0	891.0	66702
	Malheur	30421	24.414286	17966.428571	453634	7770.0	1465.0	82916
	Marion	330453	15.329310	25903.862069	1974030	34089.0	5720.0	365801
	Morrow	11153	13.450000	23171.500000	139209	1447.0	227.0	23219
	Multnomah	788459	15.730588	36739.558824	3374737	58787.0	10244.0	680418
	Polk	79666	18.641667	24633.916667	268036	5480.0	743.0	50986
	Sherman	1635	13.700000	34226.000000	5807	0.0	0.0	855
	Tillamook	25840	15.437500	25805.750000	34370	92.0	0.0	6850
	Umatilla	76736	16.520000	23200.466667	933975	10661.0	1645.0	154995
	Union	25810	17.425000	26508.875000	161223	1533.0	338.0	28227
	Wallowa	6864	14.400000	26943.000000	13017	449.0	93.0	2306
	Wasco	25687	13.037500	25089.750000	121202	3039.0	621.0	22511
	Washington	572071	10.446154	34970.817308	2157339	22455.0	3860.0	424620
	Wheeler	1415	20.600000	21268.000000	1454	53.0	2.0	359
	Yamhill	102366	13.935294	28578.882353	356425	6010.0	812.0	69481

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:

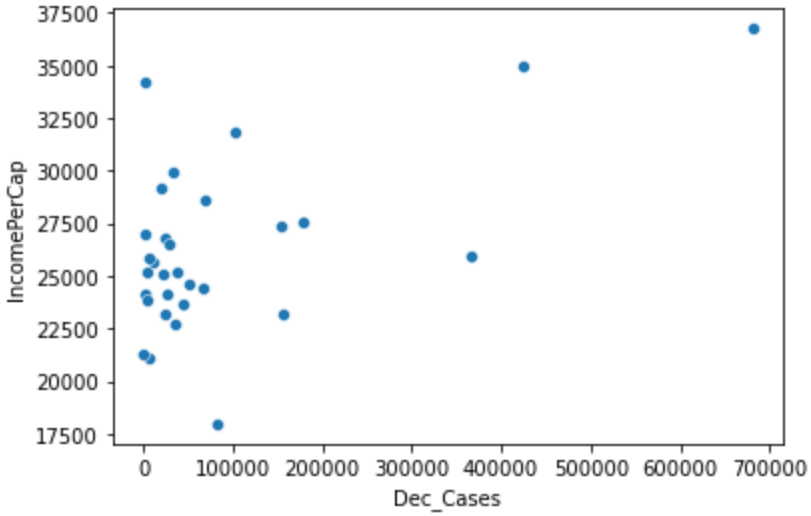
```
R = df['TotalCases'].corr(df['Poverty'])
```

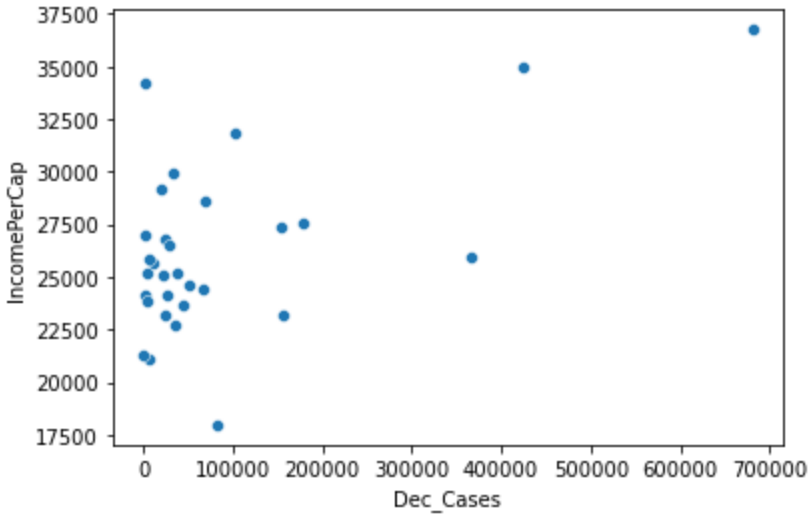
For any R that is > 0.5 or < -0.5 also display a scatter plot (see pandas scatterplot and seaborn documentation for information about how to display scatter plots from DataFrame data).

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 deaths 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 values -0.2202511100275006
R values 0.010357377565565073
R values 0.17177849422507072
R values -0.2742812164208735
R values -0.15042115533686057
R values -0.15042115533686057
R values 0.5634823447732153
R values 0.5634823447732153
```





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 deaths 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 values -0.03191313527808724
R values -0.004417640760717268
R values 0.06346188323981952
R values -0.008939053840965413
R values -0.018485705824012776
R values -0.018485705824012776
R values 0.19638058486619484
R values 0.19638058486619484
```

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.