Forbes Miyasato
2/25/2021

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

State	County	TotalPop	Poverty	IncomePerCap
Virginia	Loudoun County	374558	3.884375	50391.015625
Oregon	Washington County	572071	10.446154	34970.817308
Kentucky	Harlan County	27548	33.318182	16010.363636
Oregon	Malheur County	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.
fi
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_reported_dec_2020	deaths_reported_dec_2020
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.

```
In [69]: df.loc[df.index.get_level_values('State') == 'Oregon']
```

Out[69]:

State	County	cases	deaths	cases_reported_dec_2020	deaths_reported_dec_2020	TotalPop	Poverty	IncomePerCap	Total Cases	Total Deaths	Tota
Oregon	Baker	629	7.0	472.0	5.0	15980.0	15.000000	25706.833333	3936.170213	43.804756	393
	Benton	2248	16.0	1347.0	11.0	88249.0	23.644444	29926.000000	2547.337647	18.130517	254
	Clackamas	13196	172.0	10058.0	114.0	NaN	NaN	NaN	NaN	NaN	
	Clatsop	766	6.0	553.0	3.0	NaN	NaN	NaN	NaN	NaN	
	Columbia	1208	21.0	837.0	14.0	NaN	NaN	NaN	NaN	NaN	
	Coos	1347	18.0	756.0	9.0	NaN	NaN	NaN	NaN	NaN	
	Crook	765	18.0	448.0	7.0	NaN	NaN	NaN	NaN	NaN	
	Curry	394	6.0	278.0	3.0	NaN	NaN	NaN	NaN	NaN	
	Deschutes	5839	58.0	3976.0	22.0	175321.0	12.208333	31834.375000	3330.462409	33.082175	333
	Douglas	2312	51.0	1387.0	39.0	107576.0	16.731818	25208.545455	2149.178255	47.408344	214
	Gilliam	53	1.0	37.0	1.0	1910.0	9.900000	24178.000000	2774.869110	52.356021	277
	Grant	221	1.0	170.0	1.0	7209.0	15.850000	23855.000000	3065.612429	13.871549	306
	Harney	266	6.0	134.0	2.0	7195.0	16.300000	25174.500000	3697.011814	83.391244	369
	Hood River	1057	29.0	816.0	14.0	22938.0	12.150000	29178.000000	4608.073938	126.427762	460
	Jackson	8115	108.0	5884.0	72.0	212070.0	17.882927	27328.780488	3826.566700	50.926581	382
	Jefferson	1918	27.0	1425.0	17.0	22707.0	20.316667	22689.666667	8446.734487	118.906064	844
	Josephine	2266	48.0	1193.0	22.0	84514.0	19.131250	24179.062500	2681.212580	56.795324	268

Klamath	2752	54.0	1910.0	18.0	66018.0	18.930000	23712.400000	4168.560090	81.795874	416
Lake	373	6.0	197.0	4.0	7807.0	19.200000	21121.500000	4777.763546	76.854105	477
Lane	10033	121.0	6929.0	92.0	363471.0	18.529070	27546.220930	2760.330260	33.290139	276
Lincoln	1120	19.0	880.0	17.0	47307.0	17.623529	26807.411765	2367.514321	40.163189	236
Linn	3533	55.0	2650.0	32.0	121074.0	16.923810	24452.714286	2918.050118	45.426764	291
Malheur	3331	58.0	2914.0	50.0	30421.0	24.414286	17966.428571	10949.672923	190.657769	1094
Marion	18171	280.0	13928.0	210.0	330453.0	15.329310	25903.862069	5498.815263	84.732171	549
Morrow	1031	13.0	815.0	8.0	11153.0	13.450000	23171.500000	9244.149556	116.560567	924
Multnomah	31526	516.0	25290.0	394.0	788459.0	15.730588	36739.558824	3998.432385	65.444113	399
Polk	2978	42.0	1977.0	30.0	79666.0	18.641667	24633.916667	3738.106595	52.720106	373
Sherman	52	0.0	31.0	0.0	1635.0	13.700000	34226.000000	3180.428135	0.000000	318
Tillamook	403	2.0	308.0	0.0	25840.0	15.437500	25805.750000	1559.597523	7.739938	155
Umatilla	7580	80.0	5640.0	57.0	76736.0	16.520000	23200.466667	9878.023353	104.253545	987
Union	1264	19.0	980.0	14.0	25810.0	17.425000	26508.875000	4897.326618	73.614878	489
Unknown	1	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Wallowa	142	4.0	76.0	3.0	6864.0	14.400000	26943.000000	2068.764569	58.275058	206
Wasco	1218	25.0	905.0	22.0	25687.0	13.037500	25089.750000	4741.698135	97.325495	474
Washington	20866	209.0	16070.0	142.0	572071.0	10.446154	34970.817308	3647.449355	36.533927	364
Wheeler	22	1.0	17.0	1.0	1415.0	20.600000	21268.000000	1554.770318	70.671378	155
Yamhill	3716	62.0	2641.0	35.0	102366.0	13.935294	28578.882353	3630.111560	60.566985	363

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:

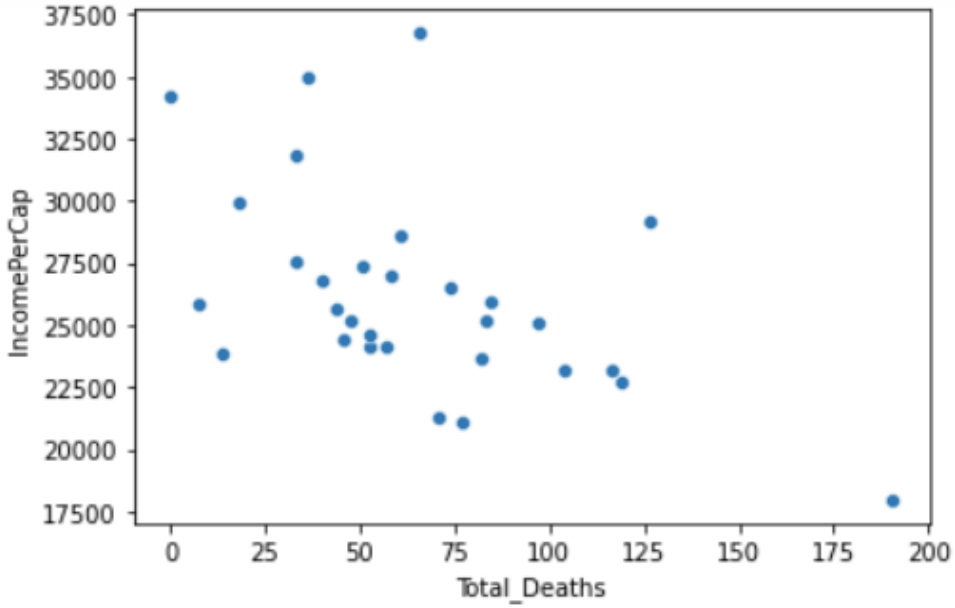
```
R = df['TotalCases'].corr(df['Poverty'])
```

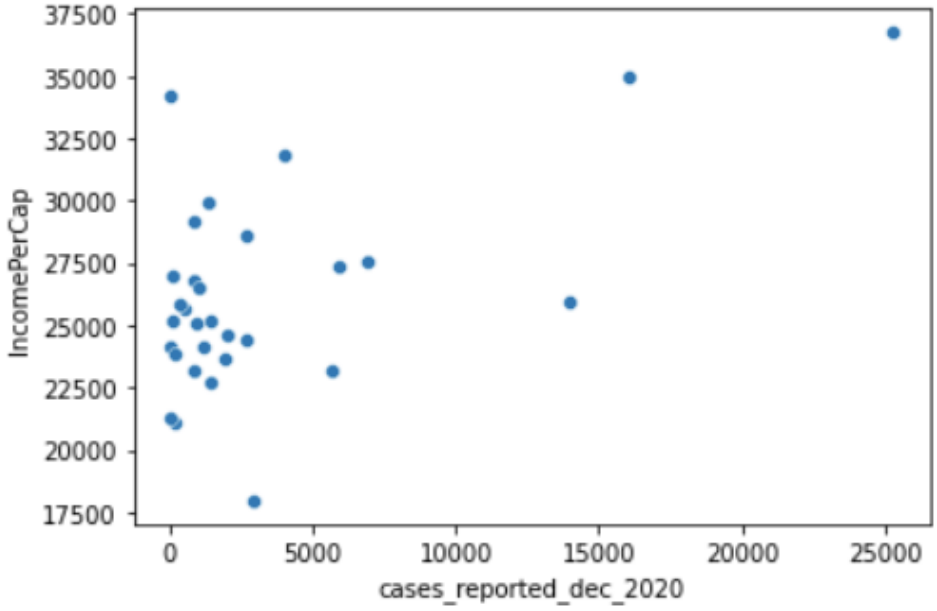
For any R that is > 0.5 or < -0.5 also display a scatter plot (see pandas scatterplot and seaborn documentation for information about how to display scatter plots from DataFrame data).

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

a. **R for Total_Cases vs Poverty = 0.19626280807425306**
b. **R for Total_Deaths vs Poverty = 0.24095099055190802**
c. **R for Total_Cases vs IncomePerCap = -0.38298538944796284**
d. **R for Total_Deaths vs IncomePerCap = -0.5103336402143774**
e. **R for cases_reported_dec_2020 vs Poverty = -0.15295719081094342**
f. **R for deaths_reported_dec_2020 vs Poverty = -0.08031013161838976**
g. **R for cases_reported_dec_2020 vs IncomePerCap = 0.5662321434044019**
h. **R for deaths_reported_dec_2020 vs IncomePerCap = 0.4999023143312516**





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

a. R for Total_Cases vs Poverty = 0.12022994647462464
b. R for Total_Deaths vs Poverty = 0.20049720020898698
c. R for Total_Cases vs IncomePerCap = -0.20541255201485065
d. R for Total_Deaths vs IncomePerCap = -0.23946027915501586
e. R for cases_reported_dec_2020 vs Poverty = -0.018326019014729848
f. R for deaths_reported_dec_2020 vs Poverty = -0.017508685701395826
g. R for cases_reported_dec_2020 vs IncomePerCap = 0.18771428825410538
h. R for deaths_reported_dec_2020 vs IncomePerCap = 0.22019147176479775

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.