

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

For aggregate county data in poverty and PerCapitaIncome columns, the program first calculates how many people are in poverty in the record region, knows the number of poverty people in each county, and then computes the percentage poverty of the whole county. For the PerCapitaIncome column, the process is similar. In this case, the result is relatively precise.

agged.loc[(agged['State'] == 'Virginia') & (agged['County'] == 'Loudoun County')]					
	County	State	Population	Poverty	PerCapitalIncome
1759	Loudoun County	Virginia	374558	3.689598	50455.645745
agged.loc[(agged['State'] == 'Oregon') & (agged['County'] == 'Washington County')]					
	County	State	Population	Poverty	PerCapitalIncome
3046	Washington County	Oregon	572071	10.321202	35369.047499
agged.loc[(agged['State'] == 'Kentucky') & (agged['County'] == 'Harlan County')]					
	County	State	Population	Poverty	PerCapitalIncome
1234	Harlan County	Kentucky	27548	35.669482	15456.971032
agged.loc[(agged['State'] == 'Oregon') & (agged['County'] == 'Malheur County')]					
	County	State	Population	Poverty	PerCapitalIncome
1822	Malheur County	Oregon	30421	24.298225	17567.504323

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.

```
covidCountyAgged.loc[(covidCountyAgged['state'] == 'Virginia') &\n    (covidCountyAgged['county'] == 'Loudoun')]
```

	county	state	casesTotal	deathsTotal	casesDec2020	deathsDec2020
2995	Loudoun	Virginia	22557	199.0	14169.0	159.0

```
covidCountyAgged.loc[(covidCountyAgged['state'] == 'Oregon') &\n    (covidCountyAgged['county'] == 'Washington')]
```

	county	state	casesTotal	deathsTotal	casesDec2020	deathsDec2020
2260	Washington	Oregon	20866	209.0	16070.0	142.0

```
covidCountyAgged.loc[(covidCountyAgged['state'] == 'Kentucky') &\n    (covidCountyAgged['county'] == 'Harlan')]
```

	county	state	casesTotal	deathsTotal	casesDec2020	deathsDec2020
1048	Harlan	Kentucky	2352	68.0	1538.0	18.0

```
covidCountyAgged.loc[(covidCountyAgged['state'] == 'Oregon') &\n    (covidCountyAgged['county'] == 'Malheur')]
```

	county	state	casesTotal	deathsTotal	casesDec2020	deathsDec2020
2249	Malheur	Oregon	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.

	County	State	Population	Poverty	PerCapitalIncome	TotalCases	TotalDeaths	Dec2020Cases	Dec2020Deaths
126	Baker	Oregon	15980	15.083855	25820.273154	629.0	7.0	472.0	5.0
197	Benton	Oregon	88249	22.421152	30872.824361	2248.0	16.0	1347.0	11.0
533	Clackamas	Oregon	399962	8.976120	37550.849108	13196.0	172.0	10058.0	114.0
558	Clatsop	Oregon	38021	12.190090	28114.625523	766.0	6.0	553.0	3.0
631	Columbia	Oregon	50207	12.315329	28459.688051	1208.0	21.0	837.0	14.0
656	Coos	Oregon	62921	17.896488	26007.212997	1347.0	18.0	756.0	9.0
697	Crook	Oregon	21717	15.320864	24238.814477	765.0	18.0	448.0	7.0
718	Curry	Oregon	22377	15.408656	26925.536399	394.0	6.0	278.0	3.0
796	Deschutes	Oregon	175321	12.100898	31574.934092	5839.0	58.0	3976.0	22.0
839	Douglas	Oregon	107576	17.025995	25001.732924	2312.0	51.0	1387.0	39.0
1068	Gilliam	Oregon	1910	9.900000	24178.000000	53.0	1.0	37.0	1.0
1115	Grant	Oregon	7209	13.635802	25154.161742	221.0	1.0	170.0	1.0
1238	Harney	Oregon	7195	17.528770	24397.712578	266.0	6.0	134.0	2.0
1320	Hood	Oregon	22938	12.123145	29594.972796	NaN	NaN	NaN	NaN
1415	Jackson	Oregon	212070	16.858350	27080.538534	8115.0	108.0	5884.0	72.0
1453	Jefferson	Oregon	22707	20.694856	22956.835293	1918.0	27.0	1425.0	17.0
1493	Josephine	Oregon	84514	18.646376	24348.609449	2266.0	48.0	1193.0	22.0
1559	Klamath	Oregon	66018	18.688624	23793.066679	2752.0	54.0	1910.0	18.0
1608	Lake	Oregon	7807	20.139311	21004.589343	373.0	6.0	197.0	4.0
1626	Lane	Oregon	363471	19.230471	27032.412179	10033.0	121.0	6929.0	92.0
1720	Lincoln	Oregon	47307	18.376280	25782.113704	1120.0	19.0	880.0	17.0
1731	Linn	Oregon	121074	16.063929	24448.467359	3533.0	55.0	2650.0	32.0
1822	Malheur	Oregon	30421	24.298225	17567.504323	3331.0	58.0	2914.0	50.0
1848	Marion	Oregon	330453	16.128516	24791.074831	18171.0	280.0	13928.0	210.0
2070	Morrow	Oregon	11153	14.699050	21742.930153	1031.0	13.0	815.0	8.0
2078	Multnomah	Oregon	788459	16.474668	34848.165612	31526.0	516.0	25290.0	394.0
2359	Polk	Oregon	79666	15.639958	25928.364057	2978.0	42.0	1977.0	30.0
2671	Sherman	Oregon	1635	13.700000	34226.000000	52.0	0.0	31.0	0.0
2860	Tillamook	Oregon	25840	15.512717	25458.191138	403.0	2.0	308.0	0.0
2919	Umatilla	Oregon	76736	17.825222	22153.237007	7580.0	80.0	5640.0	57.0
2933	Union	Oregon	25810	17.618597	26585.728710	1264.0	19.0	980.0	14.0
2994	Wallowa	Oregon	6864	13.748776	26897.389860	142.0	4.0	76.0	3.0
3021	Wasco	Oregon	25687	13.670818	24727.506132	1218.0	25.0	905.0	22.0
3046	Washington	Oregon	572071	10.321202	35369.047499	20866.0	209.0	16070.0	142.0
3109	Wheeler	Oregon	1415	20.600000	21268.000000	22.0	1.0	17.0	1.0
3195	Yamhill	Oregon	102366	13.802658	28539.604791	3716.0	62.0	2641.0	35.0

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

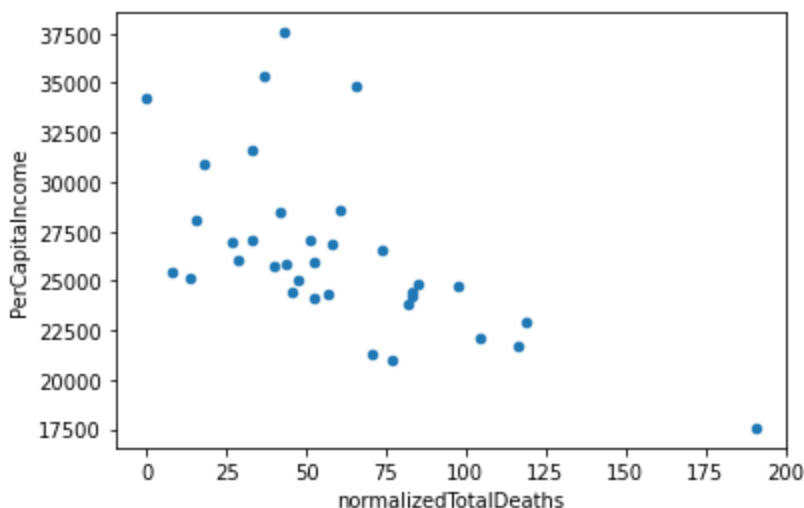
representing a distinct county, and columns named 'TotalCases' and 'Poverty', then you can compute R like this:

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

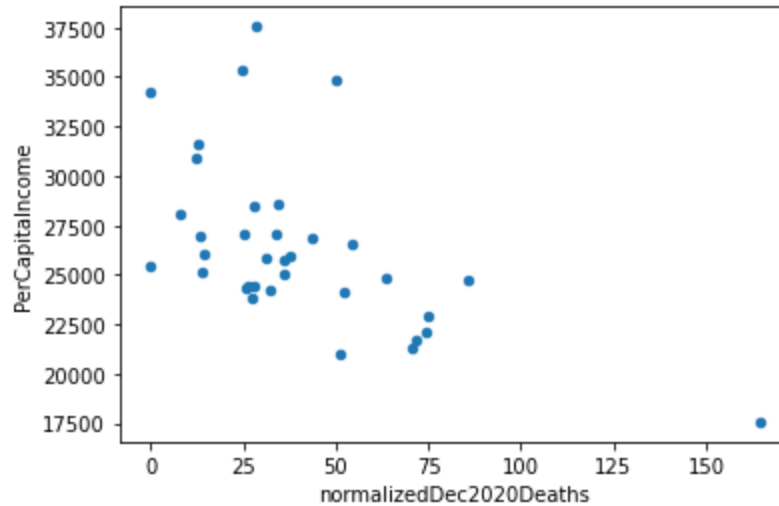
For any R that is > 0.5 or < -0.5 also display a scatter plot (see [pandas scatterplot](#) and [seaborn documentation](#) for information about how to display scatter plots from DataFrame data).

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 " $((\text{COVID total cases in county} * 100000) / \text{population of county})$ ".

1. Across all of the counties in the State of Oregon
 - a. COVID total cases vs. % population in poverty $R = 0.334$
 - b. COVID total deaths vs. % population in poverty $R = 0.446$
 - c. COVID total cases vs. Per Capita Income level $R = -0.423$
 - d. COVID total cases vs. Per Capita Income level **$R = -0.604$**



- e. COVID cases during December 2020 vs. % population in poverty $R = 0.313$
 - f. COVID deaths during December 2020 vs. % population in poverty $R = 0.419$
 - g. COVID cases during December 2020 vs. Per Capita Income level $R = -0.399$
 - h. COVID cases during December 2020 vs. Per Capita Income level **$R = -0.561$**



2. Across all of the counties in the entire USA
 - a. COVID total cases vs. % population in poverty $R = -0.0363$
 - b. COVID total deaths vs. % population in poverty $R = 0.168$
 - c. COVID total cases vs. Per Capita Income level $R = -0.000864$
 - d. COVID total cases vs. Per Capita Income level $R = -0.163$
 - e. COVID cases during December 2020 vs. % population in poverty $R = -0.0469$
 - f. COVID deaths during December 2020 vs. % population in poverty $R = 0.168$
 - g. COVID cases during December 2020 vs. Per Capita Income level $R = -0.00356$
 - h. COVID cases during December 2020 vs. Per Capita Income level $R = -0.110$

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.