Name: Karan Patel

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

Answer:

	State	County	TotalPop	IncomePerCap	Poverty
2968	Virginia	Loudoun County	374558.0	50455.65	3.69

	State	County	TotalPop	IncomePerCap	Poverty
2241	Oregon	Washington County	572071.0	35369.05	10.32

	State	County	TotalPop	IncomePerCap	Poverty
1040	Kentucky	Harlan County	27548.0	15456.97	35.67

	State	County	TotalPop	IncomePerCap	Poverty
2230	Oregon	Malheur County	30421.0	17567.5	24.3

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.

Name: Karan Patel

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

Answer:

	state	county	total-cases	total-deaths	cases-dec-2020	deaths-dec-2020
3017	Virginia	Loudoun	2496450	35820	376223	4729

	state	county	total-cases	total-deaths	cases-dec-2020	deaths-dec-2020
2277	Oregon	Washington	2157339	22455	424620	3860

	state	county	total-cases	total-deaths	cases-dec-2020	deaths-dec-2020
1054	Kentucky	Harlan	205984	3994	38959	506

	state	county	total-cases	total-deaths	cases-dec-2020	deaths-dec-2020
2265	Oregon	Malheur	453634	7770	82916	1465

Name: Karan Patel

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.
Answer:

	State	County	TotalPop	IncomePerCap	Poverty	total-cases	total-deaths	cases-dec-2020	deaths-dec-2020
0	Oregon	Baker	15980.0	25820.27	15.08	55586	663	11688	133
1	Oregon	Benton	88249.0	30872.82	22.42	180225	2304	34260	278
2	Oregon	Clackamas	399962.0	37550.85	8.98	1284402	20040	261810	3125
3	Oregon	Clatsop	38021.0	28114.63	12.19	77666	287	14439	47
4	Oregon	Columbia	50207.0	28459.69	12.32	105324	1363	21459	266
5	Oregon	Coos	62921.0	26007.21	17.90	100097	969	18806	151
6	Oregon	Crook	21717.0	24238.81	15.32	55863	1134	11048	196
7	Oregon	Curry	22377.0	26925.54	15.41	30045	393	6741	72
8	Oregon	Deschutes	175321.0	31574.93	12.10	509974	4141	102490	563
9	Oregon	Douglas	107576.0	25001.73	17.03	174952	3983	37590	964
10	Oregon	Gilliam	1910.0	24178.00	9.90	4691	76	898	25
11	Oregon	Grant	7209.0	25154.16	13.64	18551	94	4895	31
12	Oregon	Harney	7195.0	24397.71	17.53	17024	291	3717	34
13	Oregon	Hood River	22938.0	29594.97	12.12	107383	1444	19348	216
14	Oregon	Jackson	212070.0	27080.54	16.86	713288	7221	154535	1655
15	Oregon	Jefferson	22707.0	22956.84	20.69	200346	2630	36278	409
16	Oregon	Josephine	84514.0	24348.61	18.65	153675	2638	27180	407
17	Oregon	Klamath	66018.0	23793.07	18.69	224256	2857	45118	373
18	Oregon	Lake	7807.0	21004.59	20.14	25357	348	5358	76
19	Oregon	Lane	363471.0	27032.41	19.23	850956	10372	178816	2215
20	Oregon	Lincoln	47307.0	25782.11	18.38	153979	3117	24041	502
21	Oregon	Linn	121074.0	24448.47	16.06	324636	5949	66702	891
22	Oregon	Malheur	30421.0	17567.50	24.30	453634	7770	82916	1465
23	Oregon	Marion	330453.0	24791.07	16.13	1974030	34089	365801	5720
24	Oregon	Morrow	11153.0	21742.93	14.70	139209	1447	23219	227
25	Oregon	Multnomah	788459.0	34848.17	16.47	3374737	58787	680418	10244
26	Oregon	Polk	79666.0	25928.36	15.64	268036	5480	50986	743
27	Oregon	Sherman	1635.0	34226.00	13.70	5807	0	855	0

28	Oregon	Tillamook	25840.0	25458.19	15.51	34370	92	6850	0
29	Oregon	Umatilla	76736.0	22153.24	17.83	933975	10661	154995	1645
30	Oregon	Union	25810.0	26585.73	17.62	161223	1533	28227	338
31	Oregon	Wallowa	6864.0	26897.39	13.75	13017	449	2306	93
32	Oregon	Wasco	25687.0	24727.51	13.67	121202	3039	22511	621
33	Oregon	Washington	572071.0	35369.05	10.32	2157339	22455	424620	3860
34	Oregon	Wheeler	1415.0	21268.00	20.60	1454	53	359	2
35	Oregon	Yamhill	102366.0	28539.60	13.80	356425	6010	69481	812

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
        i. R = 0.2870979669130601
    b. COVID total deaths vs. % population in poverty
        i. r = 0.3605381417925185
    c. COVID total cases vs. Per Capita Income level
        i. R = -0.37568496125710993
    d. COVID total deaths vs. Per Capita Income level
        i. R = -0.46186658814724785
    e. COVID cases during December 2020 vs. % population in poverty
        i. r = 0.2981575324663765
    f. COVID deaths during December 2020 vs. % population in poverty
        i. r = 0.30275469931451643
    g. COVID cases during December 2020 vs. Per Capita Income level
        i. r = -0.3853971317908648
    h. COVID cases during December 2020 vs. Per Capita Income level

Name: Karan Patel

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

Answer for #2:

```
# COVID total cases vs. % population in poverty
total_cases = (us_df['total-cases'] * 100000) / us_df['TotalPop']
r = total_cases.corr(us_df['Poverty'])
print(f'r = {r}')

# COVID total deaths vs. % population in poverty
total_deaths = (us_df['total-deaths'] * 100000) / us_df['TotalPop']
r = total_deaths.corr(us_df['Poverty'])
print(f'r = {r}')

# COVID total cases vs. Per Capita Income level
total_cases = (us_df['total-cases'] * 100000) / us_df['TotalPop']
r = total_cases.corr(us_df['IncomePerCap'])
print(f'r = {r}')

# COVID total deaths vs. Per Capita Income level
total_deaths = (us_df['total-deaths'] * 100000) / us_df['TotalPop']
r = total_deaths.corr(us_df['IncomePerCap'])
print(f'r = {r}')

# COVID cases during December 2020 vs. % population in poverty
total_cases_dec_2020 = (us_df['cases-dec-2020'] * 100000) / us_df['TotalPop']
r = total_cases_dec_2020.corr(us_df['Poverty'])
print(f'r = {r}')

# COVID deaths during December 2020 vs. % population in poverty
total_deaths_dec_2020 = (us_df['cases-dec-2020'] * 100000) / us_df['TotalPop']
r = total_deaths_dec_2020.corr(us_df['Poverty'])
print(f'r = {r}')

# COVID cases during December 2020 vs. Per Capita Income level
total_cases_dec_2020 = (us_df['cases-dec-2020'] * 100000) / us_df['TotalPop']
r = total_deaths_dec_2020.corr(us_df['IncomePerCap'])
print(f'r = {r}')
```

```
r = 0.19742929493170183
r = 0.26803430608102446
r = -0.21366063002097785
r = -0.17345835459087575
r = 0.07035278331532802
r = 0.07035278331532802
r = -0.1593355765234985
```

Name: Karan Patel

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.