DataEng S23: Data Integration In-class Assignment

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 two pandas DataFrames as follows.

County_Info: demographic summaries for each county. This table should have one row per county (there are more than 3000 counties in the USA), and each row must include the following columns:

Name - name of the county
State - name of the state in which the county resides
Population - population of this county
Poverty - % of people in poverty in this county
PerCapitaIncome - per capita personal income for this county
ID - a unique integer ID for the county. You may choose this ID any way you like, it just needs to be unique among all counties and needs to be referenced by the COVID_monthly rows (foreign key lookup).

COVID_monthly: data about COVID cases and deaths for each USA county.

ID - refers to the county found in the County_info table (i.e., this is a foreign key lookup)
Month - integer in range 1..12 indicating the month of the year
Cases - number of COVID cases recorded for the corresponding county during the corresponding month
Deaths - number of COVID-related deaths recorded for the corresponding county during the corresponding month

For this activity you should use whichever development environment is convenient for you to develop with python and pandas. You are not required to use GCP, but you can use it if you prefer.

Submit: use the submission form found on the Materials page of the website.

Discussion Question (do this first)

Within your group, identify two or more sources of data that might be integrated to analyze a problem that could not be easily analyzed or solved otherwise. The data sources and problems do not need to be related to anything we have done in class and do not even need to be serious. You do <u>not</u> need to describe <u>how</u> to integrate the data sets.

For example, you might say something like, "Integrate historic weather/precipitation data with crop yield data to help develop a system for anticipating future prices of corn and wheat."

1. Weather change and the flu data
2. deforestation vs global warming
3. value of country's currency in global market vs inflammation rate in the country

A. Transform the ACS Data

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. I.e., they did it this way for their convenience, not yours, and this grouping makes your Data Engineering job more challenging. Your "dimension of overlap" is the county not the census tract, so you must transform the ACS data.

To properly integrate the ACS with the COVID data, aggregate the per-tract data to the county level of resolution.

Create a python+pandas program that transforms the ACS data into a one-row-per-county ACS DataFrame called County_info. To do this you will need to think about how to properly aggregate Census Tract-level data into County-level summaries. Your transformation code should also filter the data by eliminating unneeded columns from the ACS data.

Also, add an ID column to your County_info dataframe so that the county can be referred to by the COVID data.

Fill the following table using data from your County_info DataFrame:

County	Population	%poverty	PerCapitaIncome
Loudoun County, Virginia	374558	3.884375	50391.015625
Washington County, Oregon	572071	10.446154	34970.817308
Harlan County, Kentucky	27548	33.318182	16010.363636
Malheur County, Oregon	30421	24.414286	17966.428571

Answer the following questions:
- Most populous county in the USA?
  - ANS : County   Los Angeles County
    - State   California
    - TotalPop   10105722

- Least populous county in the USA?
  - ANS: County   Loving County
    - State   Texas
    - TotalPop   74

B. Transform the COVID Data

Simplify the COVID data along the time dimension. The COVID data set contains day-level resolution data from (approximately) January of 2020 through February of 2021. From this you should produce a **COVID_monthly** Data Frame that has just one row per month per county.

Also, add a county ID column that is a foreign key lookup to the corresponding ID column in the County_info DataFrame.

Fill the following table using data from your COVID_monthly DataFrame:

County	Month	# cases	# deaths
Malheur County,	August 2020	28163	459

Oregon			
Malheur County, Oregon	January 2021	96297	1627
Malheur County, Oregon	February 2021	65951	1137

C. Integrate COVID Data with ACS Data

Create a new single pandas DataFrame called COVID_summary containing one row per county. It should have these columns:
ID - integer identifier for the county
Population, Poverty, PerCapitaIncome - these should all be the same as from the County_info DataFrame
TotalCases, TotalDeaths - these two values should come from the COVID_monthly data and summed over all months
TotalCasesPer100K, TotalDeathsPer100K - these two values should be computed by dividing the the TotalCases and TotalDeaths (respectively) by (Population / 100000)

Fill the following table using data from your COVID_summary DataFrame:

County	Poverty %	TotalCasesPer100K
Washington County, Oregon	10.446154	377110.36
Malheur County, Oregon	24.414286	1491187.01
Loudoun County, Virginia	3.884375	666505.59
Harlan County, Kentucky	33.318182	747727.6

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 'TotalCasesPer100K' and 'Poverty', then you can compute R like this:

```
R = df['TotalCasesPer100K'].corr(df['Poverty'])
```

1. Compute the correlation coefficient for the following relationships for all Oregon counties
   a. COVID total cases vs. % population in poverty
      ANS: -0.1179567
   b. COVID total deaths vs. % population in poverty
      ANS: -0.0697110
   c. COVID total cases vs. Per Capita Income level
      ANS: 0.47878583
   d. COVID total deaths vs. Per Capita Income level
      ANS: 0.42613792

2. Across all of the counties in the entire USA
   a. COVID total cases vs. % population in poverty
      ANS: -0.04497796
   b. COVID total deaths vs. % population in poverty
      ANS: -0.02727625
   c. COVID total cases vs. Per Capita Income level
      ANS: 0.21751783
   d. COVID total deaths vs. Per Capita Income level
      ANS: 0.21027121

Fill the following table using data from your analysis. Add more rows to the table if you find additional interesting information:

County	R value
For Oregon Counties only: correlation between % poverty and COVID cases	-0.119567
For all counties: correlation between population and COVID cases	0.9023614
For Oregon counties only: correlation between PerCapitaIncome and COVID deaths	0.4261379
For all USA counties: correlation between PerCapitaIncome and COVID cases	0.2170517
<explore at least one other correlation computation that is made possible by this integrated data set>	

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).

Note that this assignment does not constitute a competent, thorough statistical analysis of the relationships between immunological data and demographic data. It is just an example of the type of work that is often required to integrate disparate data sets.