**Data Analysis Scenario -**

Extract Opioid usage and prescribing rates by County & State and combine the data with unemployment rates by State, County in order to analyze if there is a correlation between unemployment and opioid usage. The ETL portion of the overall scenario is described below

**Extract:**

Datasources : 2 .csv files are the datasources that were leveraged for ETL .Sources are listed below

#1- Extract .csv file from webpage attached below- Read 2016County files from CMS.gov

[https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and- Reports/Medicare-Provider-Charge-Data/OpioidMap.html](https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-%20%20%20%20%20Reports/Medicare-Provider-Charge-Data/OpioidMap.html)

#2 - Extract .csv file from webpage attached below – Read 2016County Unemployment file

[https://www.bls.gov/lau/#tables](%20%20https:/www.bls.gov/lau/%23tables)

**Transform:**

Once relevant CSV files were read into our Python Notebook we had some immediate formatting issues to address. When the files were exported from excel as CSV it incorrectly identified headers because of styling format on the original excel documents. We then correctly identified headers by row with iloc. We dropped null columns and values and reset indices. We then cleaned up by removing unwanted columns that are not relevant for our desired database. This was done for both imported datasets. We then worked on merging the data which required creating related columns to merge on. One State/County column was formatted as: “County” then a State column. The other was formatted as: “County County, State Abbreviation”. This required us to delete County from the column, so we could then split on a single column giving us County State Abbreviation. We then could split that column and remerge to our data frame. Then we could merge the unemployment by county data to the opioid prescribing data. There were many other tedious tweaks that were needed to make this happen and they are explicit in the Jupyter Notebook. We then cleaned and organized our master merged data frame and changed data types of columns, so we could do calculations on columns for our State groupby. We then created a groupby dataframe on “State” with our calculated columns by calling .mean() and .sum() functions. Finally, we established our sql connection in our script and have the data frames loaded to a SQL database titled Opioid\_db.

**Load:**

Output from transformation step was loaded to a MySQL database with SQL Alchemy commands. By storing the data in a relational database we are able to interrogate the data in SQL and analyze the tables in order to identify trends. Also, the ETL script can be repurposed and re-executed with minimal changes to update the database tables on a monthly basis.