**ETL Project - Final Report**

August 2019

Marc Donatiello

Mark Visco

1. **Original data sources**

US Census Demographic Data

Demographic and Economic Data for Tracts and Counties (Income, Poverty etc.)

[https://www.kaggle.com/muonneutrino/us-census-demographic-data#acs2015\_couty\_data.csv](https://www.kaggle.com/muonneutrino/us-census-demographic-data#acs2015_county_data.csv)

United States crime rates by county (County-level crime data of the United States)

<https://www.kaggle.com/mikejohnsonjr/united-states-crime-rates-by-county>

Both data sources are in csv format, which was cleared with Vikas with the stipulation that we create three matplotlib visualizations using our data.

1. **Data Transform: Cleaning**
2. Manually removed the following 10 counties from the crime rates file because they did not have a **match** in the Census demographic data:

**State County**

* Alaska Denali Borough
* Alaska Hoonah-Angoon Census Area
* Alaska Kusilvak Census Area
* Alaska Petersburg Borough
* Alaska Prince of Wales-Hyder Census Area
* Alaska Skagway Municipality
* Alaska Wrangell City and Borough
* Louisiana LaSalle
* New Mexico Dona Ana
* South Dakota Oglala Lakota

1. Removed the following unneeded columns in the crime data file using the ‘drop’ function in Pandas:

Index AG\_OFF

EDITION COVIND

PART INDEX

IDNO MODINDX

CPOPARST population

CPOPCRIM FIPS\_ST

AG\_ARRST FIPS\_CTY

**3**. **Data Transform: Merging**

We created separate Pandas dataframes for each of the files and used the Pandas merge’ operation to create one dataframe with the combined data.

4. **Data Transform: Aggregation**

The following functions were used in the SQL queries in postgreSQL to obtain the following values:

* **Poverty vs Crime Rate** – The Average function was used to obtain the average value for the following fields: Crime per 100k and Poverty Rate.
* **Top and Bottom 10** – In order to get the top 10 wealthiest and poorest counties, we used the fetch
* **Worker Type vs Crime Rate** - The following calculation was used on two fields to turn a percentage value into the actual numerical value:
  + Sum((SelfEmployed/100)\*employed)
  + Sum((unemployement/100)\*TotalPop)

In addition to the above, we also used the ‘sum’ function in several of the queries in postgreSQL to obtain the total number of the following types of crimes in the US:

**Murder, Rape, Robbery, Aggravated Assault, Burglary, Larceny, Auto Theft, Arson.**

This data was later used in Pandas to calculate the percentage of total US crimes for each category in order to create a pie chart of total US crime percentages.

5. **Data Visualizations**

Per an agreement with Vikas, we created three data visualizations using matplotlib as a condition for using two csv files. The following three visualizations are included in the Pandas/Jupyter file in our GitHub repository (<https://github.com/mdonatiello/ETL_Project>):

* 1. US Total Crime Percentage by Type (pie chart)
  2. US Income and Crime Rate by County (scatterplot)
  3. US Poverty and Crime Rate by State (line graph)

6. **Final production database and tables**

Relational - postgreSQL

This type of database was chosen because it would allow us to quickly extract only the data we would need to create the three plots and calculate sums and averages needed for our analysis.

7. **Steps required to reproduce our ETL process:**

1. Download both data files from Kaggle.
2. Read both csv files and Pandas and create separate dataframes for each of them, once called county\_df and one called crime\_df2.
3. Merge the two dataframes on "County" and "State code" into one dataframe called countycrime\_df.
4. Create a summary\_df using countrycrime\_df and drop any unneeded columns. This was performed in Pandas as follows:

summary\_df = countycrime\_df.drop(columns=['ID\_x', 'index', 'EDITION', 'PART', 'IDNO', 'CPOPARST', 'CPOPCRIM', 'AG\_ARRST', 'AG\_OFF', 'COVIND', 'someINDEX', 'MODINDX', 'population', 'FIPS\_ST','FIPS\_CTY', 'IncomeErr', 'IncomePerCapErr'])

summary\_df.head()

e. Export summary\_df to a csv file as follows:

summary\_df.to\_csv("Output/summary.csv", index=False, header=True)

f. Use the summary.csv file to load the data into postgreSQL

g. Pull the data needed to create three plots by creating postgreSQL tables and

running SQL queries.

h. Export the needed data from PostgreSQL into separate csv files.

i. Read the csv files into Pandas and create the three dataframes needed to create

the three plots.

j. Create the three plots using the three dataframes (one dataframe for each plot).

k. Save each of the plots as png files.

8. **GitHub repository**

The following are available in the ETL Project GitHub repository located at <https://github.com/mdonatiello/ETL_Project>:

* Jupyter Notebook Pandas file used to create dataframes and plots.
* The summary.csv file exported from Jupyter/Pandas and loaded into PostgreSQL.
* Three png files for the plots
* The original csv data files.
* SQL queries used to pull data for creating the plots.
* The csv files exported from postgreSQL that were used to create the plots.