Extraction,

Transformation, and Load

Technical Report

Table of Contents

[1.INTRODUCTION 1](#_Toc41838306)

[1.1 Summary 1](#_Toc41838307)

[1.2 Scope 2](#_Toc41838308)

[1.3 Technologies and resource contributions 2](#_Toc41838309)

[1.4 Definitions, Acronyms and Abbreviations 2](#_Toc41838310)

[2. ETL DETAILS 2](#_Toc41838311)

[2.1 Data Import/Extract Sources and Method 2](#_Toc41838312)

[2.2 Data Acquisition 2](#_Toc41838313)

[2.3 Data Transform 3](#_Toc41838314)

[2.4 Data Integrity 3](#_Toc41838315)

[2.5 Data Refresh Frequency 3](#_Toc41838316)

[2.6 Data Security 3](#_Toc41838317)

[2.7 Data Loading and Availability 4](#_Toc41838318)

# 1.INTRODUCTION

The purpose of the Extraction, Transformation, and Load (ETL) Technical Report is to capture details that pertain specifically to ETL portion of the data pipeline that is to be used in a data science project. This however does keep in mind the final target objective while performing the ETL.

## 1.1 Summary

Summary This section summarized the final objective of the project, the business problem definition (problem statement) and the expected outcome of ETL.

## 1.2 Scope

Scope The data sources are the 2014 IRS tax returns that were obtained from the IRS website itself as well as Coronavirus data that was pulled with FIPS data (able to go down to county numbers). These data sources will be integrated together. The eventual outcome of integrating these data sources is to have a connection between fiscal records and outbreak records in order to provide a database which may elucidate trends and highlight potential containment problems based on socioeconomic data.

## 1.3 Technologies and resource contributions

Christine: Data purveyor, data cleaning, coding

Sean: Write up, Data purveyor, coding

Shayon: Data purveyor, write up

Technologies used: PostGreSQL, SQLAlchemy, JupyterLab, Python Pandas, MongoDB,

PyMongo for Python, VSCode

Data provided via the IRS and CDC.gov websites

## 1.4 Definitions, Acronyms and Abbreviations

•ETL: Extract, Transform and Load

•FIPS: Federal Information Processing Standards

•Covid-19: Novel Coronavirus 19

•Coronavirus: Novel Coronavirus 19

•IRS: Internal Revenue Services

•AGI\_STUB: Creates different groups for comparisons

# 2. ETL DETAILS

This section outlines a more detailed description of the processes utilized/proposed to achieve the objectives of this initiative.

## 2.1 Data Import/Extract Sources and Method

Income data was downloaded from the IRS off of Kaggle in a CSV file at this website <https://www.kaggle.com/irs/individual-income-tax-statistics>. Co-Vid 19 data was downloaded from the New York Times off of Kaggle in a CSV file at this website <https://www.kaggle.com/fireballbyedimyrnmom/us-counties-covid-19-dataset>. The CSV file that listed state abbreviations was downloaded from world population review at this website <https://worldpopulationreview.com/states/state-abbreviations/>. There are no restrictions on the CSV files from Kaggle.

## 2.2 Data Acquisition

Data for geographical location of Co-Vid 19 cases was needed along with income information related to geographical location. Both sources of data are static. In order to update or obtain the data again it would have to be downloaded from the IRS and the New York Times again through Kaggle. Data downloaded from Kaggle for the New York Times has had the same columns and would work with the existing code. The data from the IRS was from 2014 and updates rarely. If they updated their data they may use different column names and would need to be coded again.

## 2.3 Data Transform

Our transformation of the data began with our IRS data which had several columns dropped because they were extraneous to our purpose and we needed to optimize the database. Other columns needed to be renamed in order to understandably label the data and call it in the database. The Co-Vid 19 data needed columns dropped that were not useful for our purpose. Finally, we needed the Co-Vid 19 data to be capable of being related to the IRS data. I decided to merge a state abbreviation list to the Co-Vid 19 data which completed our main task. We determined minimal calculations needed to be applied to our data because our data had been presented structurally to our liking.

## 2.4 Data Integrity

The Co-Vid 19 data had fips numbers that were not standardly used with decimal points after them. Also, there were N/As and unknowns in several columns in the Co-Vid data on the county level which we decided to drop completely. We believed our client would need a complete harmonized dataset to quickly and easily run data without our assistance. Due to the frequency of our data we were able to drop multiple columns with redundant data and columns that did not give value to the underlying dataset. Since the data has to be downloaded in a CSV file to load into PANDAS and then uploaded to pgAdmin then it would make sense to notify the team when it is updated. I wouldn’t think that it would be necessary to notify the team. For notifications we will have one service monitoring for change, using SqlDependency/QueryNotifications. This service would push notifications, using WCF for instance, to all our clients running apps. Our client would subscribe to changes such as (“the table COVID19 was changed”).

## 2.5 Data Refresh Frequency

Our IRS data was a challenge to find as we wanted to keep the integrity of the data. With this we ended up keeping the IRS income dataset static and used the most recent year available which is 2014. We thought this would go well with our main dataset the COVID19 geographic data. This data is refreshed daily, however, we have decided it was best to cut down our frequency to weekly as we would like the ability to quality check and perform other tasks as needed. This would provide our client with a greater accuracy and continuation uninterrupted data. It would be necessary to update the local data with the new Co-Vid data at least weekly in order to keep track of changing numbers of Co-Vid cases to states.

## 2.6 Data Security

All data is open sourced from the web and does not have any personal information which could be attached to third parties. The data is purely for knowledge and educational purposes and since it is open source is readily available to be utilized. The data that is pulled from the cdc.gov website had to be specifically requested forand as such is not for making high level decisions. Purely for educational purposes and elucidating potential trends

## 2.7 Data Loading and Availability

We have loaded the data from our CSV into pgAdmin using the graphical user interface of pgAdmin. We created our tables first and imported the data into the system by right clicking on Table name and clicking ‘Import’. For access client we will use an ODBC connector – clients can go to ‘<https://odbc.postgresql.org/>’ to download the library. We will also use ODBC because we would like the client to be able to use our data with a variety of different programs such as R, Python, SPSS, and Excel. We would to not only have our client pull our captures organized data but also be able to do their own inferences if need be and do this seamlessly with an ODBC connector. This will allow them to access the data with most third-party providers. Included in the instructions and set-up for the client we will be providing the necessary packages need to be installed to run their choice of third party provider such as pyodbc for Python and RODBC for R. Since we will need admin rights we will be providing an initial 1on1 consultation to install these drivers on our customers desired portals. Client will be able to run queries on the initial dataset.