Extraction, Transformation, and Load Technical Report

<Breast Cancer & Income>

# TABLE OF CONTENTS

1. Introduction
   1. Summary
   2. Scope
   3. Technologies and resource contributions
   4. Definitions, Acronyms and Abbreviations
2. ETL Details
   1. Data Import/Extract Sources and Method
   2. Data Acquisition
   3. Data Transform
   4. Data Integrity
   5. Data Refresh Frequency
   6. Data Security
   7. Data Loading and Availability
3. Data Quality

# INTRODUCTION

**1.1 Summary**

Breast Cancer has affected the lives of many individuals across the United States. Subsequently, many questions have arisen regarding the cause of such a caustic and life-threatening disease. With so many questions from the public, our client asked us to focus on the correlation between income level and the rate of breast cancer cases being discovered per county in the eastern portion of the United States.

The expected outcome from the ETL project is one supporting a negative correlation between rising incomes and an increasing rate of cancer cases per county in our selected data points, the majority of which are located in the Eastern portion of the United States.

* 1. **Scope**

The purpose of the extraction, transformation, and loading (ETL) of our technical report is to capture details that pertain specifically to the ETL portion of the data pipeline. This portion of the pipeline was the primary focus of our data science project.

The scope of our project involved performing the ETL work to aid in finding a correlation between breast cancer cases and average income per county. The objective of cleaning our data was to focus on the income levels and individual cancer cases only. Factors that may influence the income levels or cancer cases per county, such as job employment or living conditions (such as smoking, diet, etc.), were not considered in the scope of our ETL project.

* 1. **Technologies and Resource Contributions**

Each of our team members contributed to the project. To begin, Arturo, Jackie, and Nate performed project definition and scoping activities. They identified the scope of the project, the in-scope fields, and the data sources. Next, the full team began data investigation to ensure data quality and integrity. Once data integrity was confirmed, the team began to cleanse the data sets. Once the team identified what data cleansing was required, Dean performed the data cleansing in Python. While this was occurring, Conor worked with all members of the team to ensure that we were working to meet our objectives; this meant being a resource to confirm that the coding approach would match the data sets pulled. Additionally, there was manual normalization that needed to occur prior to transformation of the data. Conor identified where the gaps in the data set were and Arturo, Conor, Jackie, and Nate created a normalization file. Dean read this file into Python and joined it to the work in progress data. Dean and Conor then worked together to load the data into SQL while Arturo, Jackie, and Nate began to construct the report. Conor and Dean completed the loading process and helped finalize the report with Arturo, Jackie, and Nate.

The ‘tech stack’ that was used to produce our final deliverable included a number of the tools/programming languages that we have used throughout the boot-camp up until this point. While combing through many different data-sets at the beginning or our ETL project, we used Pandas describe method and Excel pivot tables to perform the initial steps of data discovery. Once we ‘defined’ the data sets to be used, we utilized Python code and pandas within Jupyter notebook to clean our data sets. Once our datasets, which were Excel files, were effectively cleaned, we utilized QDB to develop the table schemas, which we then loaded into tables using Postgres SQL. Throughout the data-retrieval/data-cleansing process, we utilized GitHub to share all of our files amongst our team members, which served to efficiently facilitate the overall workload of the group.

**1.4 Definitions, Acronyms and Abbreviations**

Midway between the extraction and transformation of our data sets, it became very clear that there are a variety of definitions,

mainly regarding variables created in Python and SQL code, that needed to be further explained. Listed below are the variable/field names that needed clarification regarding their various titles and descriptions.

* CaseCount = Number of cancer cases reported (on a per county basis)
* Sum\_w = Household count by zip code
* Mean = The mean household income reported by zip code in the Kaggle dataset
* Pandas = Python library that provides a “dataframe” structure to manipulate data
* Postgres = Platform to host and store structured data tables in SQL
* SQL = Structured Query Language
* Join = Method of consolidating data tables by matching a key in one table to a matching key in a different table
* Inner Join = A type of join in which only the matching keys and values are returned

# 2. ETL DETAILS

**2.1 Data Import/Extract Sources and Method**

The two datasets used in this project were the CDC’s breast cancer database (<https://gis.cdc.gov/cancer/uscs/dataviz.html>) and a Kaggle household income dataset (<https://www.kaggle.com/goldenoakresearch/us-household-income-stats-geo-locations#kaggle_income.csv>). The CDC database was manipulated to pull the following parameters: the years 2011 - 2016, the type of cancer: breast cancer, and the number of new cancer cases. The CDC had APIs available; however, due to time constraints, the dataset was pulled manually, as it was readily available for download.

**2.2 Data Acquisition**

In future projects, the team will work to automate our current data extractions to account for additional years of data. The Kaggle dataset that was used was pulled manually due to the relatively small scope of the project and the rather short time frame that was given to gather necessary data points. In a future project, automating the data pulls through APIs would be our recommended course of action should we partner with the client again.

Our team pulled a dataset of 1,000 counties to match those that we obtained from the CDC breast cancer database. Because of time constraints, this did not amount to a scope of all 50 states. A future project would include all 50 states. Additionally, this process could be easily and efficiently tweaked to examine other cancer types or pull more cancer types in to check all cancers in aggregate.

At the very being of our data extraction, we pulled counties from all southeastern states from both the CDC database and the Kaggle data set. This initially amounted to ~ 1,050 counties which was statistically significant. Upon an initial join of the two tables, the team realized that some instances of the counties and states (the join keys) were not an exact match between the datasets. Consequently, the newly created ‘merged table’ was about 600 counties and was no longer statistically significant. Consequently, the team expanded the scope and doubled the amount of states included in our analysis (now including states from the East coast and Midwest). This simple data expansion step allowed for our dataset to exceed the 1,000 data point minimum. Also, our data set is now statistically significant.

**2.3 Data Transform**

Our team initially performed a review of the Kaggle dataset. The CDC dataset had data at a county level; however, the Kaggle dataset had values at a zip level (with an accompanying county) and was therefore more granular. Because of this, we realized we would need to calculate a weighted average to create values of income by county rather than by zip code. By aggregating the CDC dataset, we could then successfully join that data to the Kaggle data. We calculated the weighted average by weighting the income by zip code against the number of households used in the statistical calculation for that zip code. This would give a weighted average income per county that could be compared to the new rate of breast cancer in the CDC data set.

Next, our team saw that some of the zip codes had incomes of ‘0’ because there was no reported data for that zip code. Those county records were also indicated with a household count of ‘0’. These records were indicated by ‘0’ because they elected not to provide data. The affected records were removed from the dataset as there was no value in maintaining them. These problematic data points would serve as a hindrance during data manipulation.

After we completed this particular data cleansing activity, the team normalized county titles across both data sets. For the most part, these datasets had the same nomenclature for county names. We found 9 instances of counties that needed to be normalized between both data sets. To normalize, we looped through the data frame of the joined datasets and replaced the instances that needed to be replaced with their normalized county name. At this point, our team had two final data sets of cleaned data. Our team then used the QDB tool to create and visualize the table schema and relationships. We created those tables in SQL using PostGres and loaded our cleaned data into the structured tables.

To begin transforming the tables, we queried the tables to create a clearer view of the data. We then selected the in-scope columns from each respective table. We performed an inner join on the county names and state (to avoid common county names across states causing an error). On the inner join, we then performed calculations to calculate the weighted average of the income by county. We took the mean household income and multiplied by the number of households per zip code to get an extended household income by zip code. We then aggregated the extended household income by zip code on a per county basis and divided by the number of households in total in that county. This gave a weighted average household income by county. The team created views to be able to run the final calculation of a weighted income because the two columns were calculated columns and did not hold memory to be able to have calculations ran off of them.

To finalize the transformation, the team ran another query to create a final data table to be outputted and delivered to the client.

**2.4 Data Integrity**

The data used for this ETL project was from two sources: Kaggle and CDC. Regarding the data extracted from Kaggle, and its reliability, this is a static data source from 2017 and not subject to change. The income data within the Kaggle dataset was sourced from the 2017 United States Census. This data source will not change but will continue to update for the following year. If our partnership continued, we could update our data set with this new information should we continue our research. Notification of this data source currently changing is tracked within Kaggle via change control history.

In addition to the Kaggle dataset, the reliability of the CDC data was not affected by missing data for “number of houses polled”. However, dropping these none reporting areas did not influence any of our conclusions. Our aims centered around the number of breast cancer cases and overall average income of counties. The frequency of updates to the CDC cancer data will occur on an annual basis. There is no expectation of this frequency to change; however, notification of updates to this data source are available via the CDC’s “Stay Informed Social Media”.

**2.5 Data Refresh Frequency**

Because both household incomes and breast cancer rates do not rapidly change, a cadence of automated pulls could be set at a mutually agreed upon date. We recommend an annual refresh as we are limited by the availability of the census data.

**2.6 Data Security**

Healthcare and income information are typically personal and very private. However, since we are accessing our data at a high level the data was completely anonymized, thus making it more easily accessible. Therefore, there should be no security requirement at the level we are currently reporting. If the client requests further granularity in the delivered data table, we would potentially need more security clearance due to HIPAA regulations, but any data being manipulated would most likely remain anonymous. If source data remained anonymous, we would not need additional security clearance. The HIPAA Security Rule establishes national standards to protect individuals’ electronic personal health information that is created, received, used, or maintained by a covered entity. The Security Rule requires appropriate administrative, physical and technical safeguards to ensure the confidentiality, integrity, and security of electronic protected health information. This data had no personally identifiable information such as date of births, patient ids, names, etc. There was no need for any additional privacy, Encryption, Data masking, Auditing, Backups etc.

**2.7 Data Loading and Availability**

Our team would create and load the output into an HTML database portal created for the client or send a CSV to the client. The use of frontend technologies, such as Bootstrap’s grid system, would be supported by loading API’s from the defined data sources into a database system or virtual database. Alternatively, data could be inputted via HTML with CSV upload capacity from a Python script.

# 3. DATA QUALITY

Some KPI’s used by the client for this project could include:

* Number of counties compared: > 1,000
* Number of states included in the final database: all 50 states included
* Time to pull the dataset (i.e. level of automation of ETL): upon receiving dataset, team can output final dataset in < 15 minutes
* Accuracy of dataset: < 1% of counties include errors

Our team was able to meet all KPIs except the states KPI. We would like to push back on the client’s request for that. While the client might think he or she wants to look across all 50 states, that does not necessarily equate to statistical significance. We wanted to focus on outputting a statistically significant output through a repeatable and efficient process. We have achieved that. Our process is highly automated with 0 errors and > 1,000 datapoint. Because of the number of datapoints, the client can run statistically significant analyses on the dataset. As the data science professionals, we are recommending the client relax or remove the KPI surrounding the number of states. If not, we are going to need more time (and money) to perform further loads. Because we have created a scalable and efficient process, it should not be much more work.

In terms of site acceptance testing, we would recommend the client test our process through “dummy data”. If the client provides our team with data that they have an accompanying desired output, we can run our processes and output the exact same desired output.