Nicholas Hanashiro

Jordan Roessle

ETL Project

We found 3 data sources as csv files from Kaggle that we thought would be interesting to investigate, which include Forbes’ 2021 Billionaires list, Country Populations, and Country GDP’s. We chose these links with the idea that for a project of larger scope than only ETL, we would have looked at demographics of countries and what kind of impact that had on their quantity of billionaires. And conversely if quantity of billionaires might seem to affect their country. That was the “Extract” portion of our ETL process.

Moving onto how we transformed the data, we dropped columns that were irrelevant to the insights we would try to analyze like: Status (Marital), Age, Self-Made, Citizenship, and a curious column called Geometry that neither of us could figure out the purpose of. From the GDP.csv we dropped all years before 2019, and from the Country.csv we dropped things like yearly change, fert.rate (fertility rate), and urban population.

Loading the data was where we probably had most of our headaches. While initial extraction and cleaning made the previous csv’s seem quite compatible, it is in this loading stage where we realized that certain country’s GDP data was not available in our GDP.csv, and the education column needed 500 characters to be loaded into an SQL database. However, once we were able to load the data, we ended up needing to return to the “Transform” section of ETL.

Transform 2 Electric Bugaloo, led us down a few different difficult rabbit holes to get out of. We were missing gdp values and demographic information from a few countries. Instead of manually searching up our missing country’s GDP’s we felt that a more elegant solution would be to scrape Wikipedia for GDP on the countries we needed, and while we were at it may as well scrape information from Wikipedia on our billionaires that may seem useful. Eventually we needed to abandon grabbing information about the billionaires due to Wikipedia’s ambiguity. If multiple people have the same name the URL of the page changes slightly. Also, there were some tables included at the top of pages that weren’t in others, making using pd.read\_html inconsistent. An attempt to use splinter is included in the repo; however, that code would take hours to execute so it is left unexecuted.

In the end in order to be able to submit the project in a timely manner we decided to scrap the idea of scraping for billionaire info from Wikipedia and only find general GDP data from google for the countries missing GDP. We then merged the population and GDP datasets in order to make them easier to manage.

Then once we appended the newly scraped data, we were able to return to the “Load” section of our project, we began getting a strange error in postgres that would not allow us to import our newly merged gdp and population dataframe.

Due to this error, we changed our approach and decided to create the databases using sql alchemy. Using this method, we circumvented the error that was giving us trouble before and were able to complete the project.