ETL Project

Analysis of World Bank Data relating to Population, GDP & Gender Ratio's

Project Group 1

September 2022

Danny Bruzzese

Lin Huan Jhe

Angela Alexander Smith

Background

With the growth of the global human population from 2.5 billion in the 1950's to an estimated 8 billion now, the viable livable areas and resources required are diminishing putting pressure on society.

We saw an experiment “Universe 25, 1968-1973” which is a series of rodent experiments that showed that even with abundant food and water, personal space is essential to prevent societal collapse. Although some people think the experiment was rigorous and human social networks are not like rodent animals’, we still do notice some similar phenomenon found in Universe 25 happening in our society.

Objective

The objective of this project was to extract the relevant data from the website “https://data.worldbank.org/” and transform the datasets to hold only relevant information from the years of 2000 through to 2020 allowing the following data to be analyzed regarding the following questions.

- What impact does GDP have on Population Growth?
- As country borders have generally remained static for the past 30 years, does the population increase rate slow down as time passes? How about GDP?
- What impact does the Gender Ratio of a country have on the Population Growth?
- Does this also have an impact on the GDP of a country?

Extract

Three data sets were sourced from the following site https://data.worldbank.org/.

- GDP.csv
- Gender_StatsData.csv
- Population.csv

Utilizing Jupyter notebooks the following dependencies were imported

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

All 3 csv files were read into the notebook and dataframes created to hold the relevant information.

"InputData/GDP.csv"
"InputData/Population.csv"
"InputData/Gender_StatsData .csv"

Transformation

population.csv

After reading in the population.csv the irrelevant columns were dropped to only include the data relating to the Country Name and the years of interest (2000 – 2020) this was done by column number rather than column name in the interest of simplicity. The Country Name column was renamed to" countryName" for ease of the load process. All year columns were then converted to an integer using the "astype" method.

This updated data frame was then converted back to an updated csv file.

	countryName	2000	2001	2002	2003	2004	2005	2006
0	Aruba	90866	92892	94992	97016	98744	100028	100830
1	Africa Eastern and Southern	398113044	408522129	419223717	430246635	441630149	453404076	465581372
2	Afghanistan	20779957	21606992	22600774	23680871	24726689	25654274	26433058
3	Africa Western and Central	267214544	274433894	281842480	289469530	297353098	305520588	313985474
4	Angola	16395477	16945753	17519418	18121477	18758138	19433604	20149905

GDP.csv

After reading in the GDP.csv the irrelevant columns were dropped to only include the data relating to the Country Name and the years of interest (2000 – 2020) this was done by column number rather than column name in the interest of simplicity. The Country Name column was renamed to “countryName” for ease of the load process using MongoDB. Year columns were then rounded to two decimal places. This updated data frame was then converted back to an updated csv file.

	countryName	2000	2001	2002	2003	2004	2005	2006	2007
0	Aruba	7.62	4.20	-0.96	1.12	7.28	-0.38	1.14	3.10
1	Africa Eastern and Southern	3.35	3.66	3.89	3.08	5.51	6.12	6.55	6.60
2	Africa Western and Central	3.73	5.21	9.90	5.52	8.01	5.85	5.37	5.53
3	Angola	3.05	4.21	13.67	2.99	10.95	15.03	11.55	14.01
4	Albania	6.95	8.29	4.54	5.53	5.51	5.53	5.90	5.98

Gender_StatsData.csv

After reading in the Gender_StatsData.csv the irrelevant columns were dropped to only include the data relating to the Country Name, Indicator Name and the years of interest (2000 – 2020) this was done by column number rather than column name in the interest of simplicity. The Country Name column was renamed to “countryName “for ease of the load process using MongoDB. This was then further filtered to only include the Indicator Name referencing the population by gender (either Male or Female). The data type was then ascertained and then converted to an integer using the “astype” method. The “reset_index” method was also used to further clean up the final data set. This updated data frame was then converted back to an updated csv file.

	countryName	indicatorName	2000	2001	2002	2003	2004	2005	2006
0	Africa Eastern and Southern	Population, female	200985544	206249128	211655114	217219737	222965267	228908107	235056143
1	Africa Eastern and Southern	Population, male	197127500	202273001	207568603	213026898	218664882	224495969	230525229
2	Africa Western and Central	Population, female	133462296	137042841	140713935	144490048	148390233	152428247	156610597
3	Africa Western and Central	Population, male	133752248	137391053	141128545	144979482	148962865	153092341	157374877
4	Arab World	Population, female	138369759	141291510	144229745	147220575	150310846	153533815	156899292

Load

The data was then loaded by two different methods using SQL_DB and NoSQL_DB

Using PostgreSQL

The required dependencies were imported

```
import psycopg2
from psycopg2 import Error
from psycopg2.extensions import ISOLATION_LEVEL_AUTOCOMMIT
from sqlalchemy import create_engine
from sqlalchemy import inspect
from SQLkeys import password
```

A host port was then created. We then connected to “PostgresSQL” to create a Database and then inputted the data using the create engine method. Tables were then created for the three datasets of GPD, Population & Gender. Following that the tables were then read out into updated CSV files prior to closing the connection.

```
sql_gdp = '''
CREATE TABLE gdp(countryName VARCHAR (255), "2000" float ,\
    "2001" float ,"2002" float ,"2003" float ,\
    "2004" float ,"2005" float ,"2006" float ,\
    "2007" float ,"2008" float ,"2009" float ,\
    "2010" float ,"2011" float ,"2012" float ,\
    "2013" float ,"2014" float ,"2015" float ,\
    "2016" float ,"2017" float ,"2018" float ,\
    "2019" float ,"2020" float );
'''

sql_pop = '''
CREATE TABLE population(countryName VARCHAR (255), "2000" float ,\
    "2001" float ,"2002" float ,"2003" float ,\
    "2004" float ,"2005" float ,"2006" float ,\
    "2007" float ,"2008" float ,"2009" float ,\
    "2010" float ,"2011" float ,"2012" float ,\
    "2013" float ,"2014" float ,"2015" float ,\
    "2016" float ,"2017" float ,"2018" float ,\
    "2019" float ,"2020" float );
'''

sql_gender = '''
CREATE TABLE gender(countryName VARCHAR (255),indicatorName VARCHAR (255),\
    "2000" float ,"2001" float ,"2002" float ,\
    "2003" float ,"2004" float ,"2005" float ,\
    "2006" float ,"2007" float ,"2008" float ,\
    "2009" float ,"2010" float ,"2011" float ,\
    "2012" float ,"2013" float ,"2014" float ,\
    "2015" float ,"2016" float ,"2017" float ,\
    "2018" float , "2019" float ,"2020" float );
```

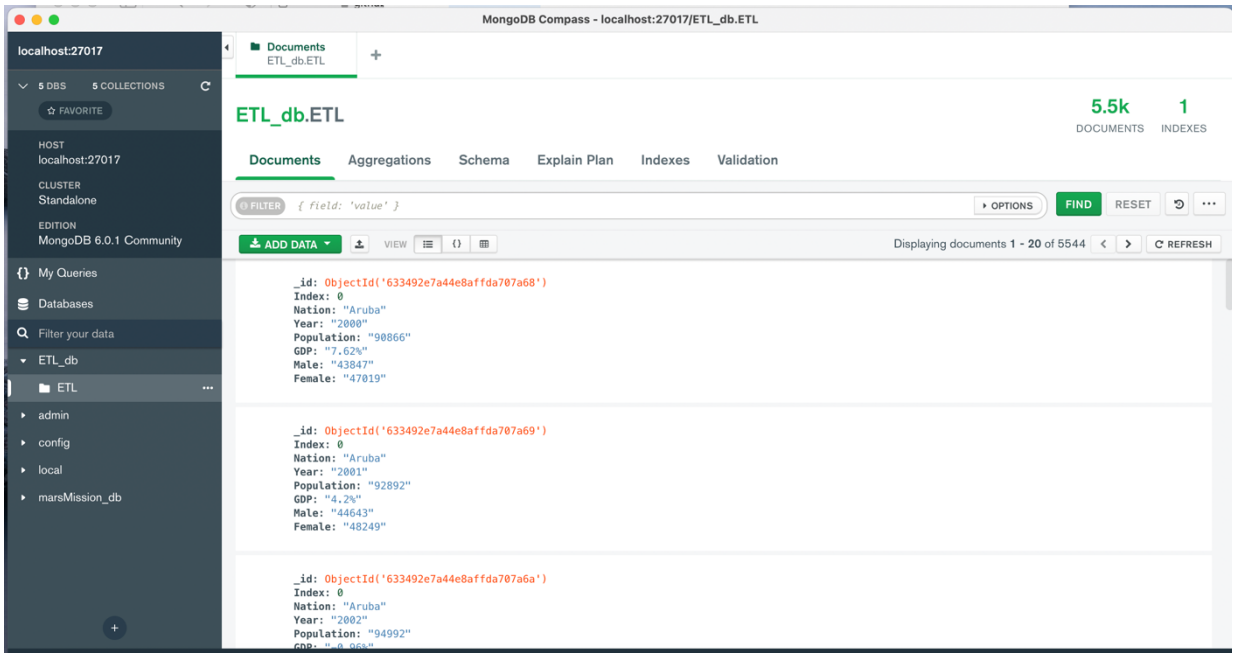
Using MongoDB

The required dependencies were imported

import pymongo

and a connection made to the client. A list of the Country names was created and the years were appended. Using a for loop a collection of the data was created of the key value pairs.

```
# get country list to for loop

countries=cleaned_pop_df['countryName']

#get year list to for loop

cols=cleaned_pop_df.columns.to_list()

years=[]

for i in range(1,len(cols)):

    years.append(cols[i])


#collect MongoDB data storing type
collector=[]

try:
    for i in range(0,len(countries)):
        for j in range(0,len(years)):


            population = cleaned_pop_df.loc[(cleaned_pop_df['countryName']==countries[i]),[years[j]]].squeeze()

            gdp = cleaned_gdp_df.loc[(cleaned_gdp_df['countryName']==countries[i]),[years[j]]].squeeze()

            male = gender_female = gender.loc[((gender['countryName']==countries[i]) & (gender['indicatorName']=='Population, male')),[years[j]]].squeeze()

            female = gender_female = gender.loc[((gender['countryName']==countries[i]) & (gender['indicatorName']=='Population, female')),[years[j]]].squeeze()

            collection={'Index':i,'Nation':countries[i],'Year':years[j],'Population':str(population),'GDP':f'{gdp}%','Male':str(male), 'Female':str(female)}

            collector.append(collection)

except Exception as e:
        print(e)


print(collector)
✓ 15.4s
```



Summary

Using the ETL data integration process we were able to combine the three large csv files from https://data.worldbank.org/ into a consistent data store that will allow the data analysis and visualization for users to be able to answer the following questions as per our initial objective.

What impact does GDP have on Population Growth?
As country borders have generally remained static for the past 30 years, does the population increase rate slow down as time passes? How about GDP?
What impact does the Gender Ratio of a country have on the Population Growth? Does this also have an impact on the GDP of a country?