ETL Project

Student: Yi An Pan (Emma)

Email: emma.pan@nyu.edu

Contents

I. Extraction (E) 1

II. Transformation (T) 1

III. Load (L) 3

I. Extraction (E)

The report is to utilize different data types; therefore, first I found country list as one of my dependency. Country list was found at https://finnhub.io/ that I was getting API key on the website to read data. Following I found population, forex and inflation data which are CSV, CSV, and Excel files. I aimed to merge by chosen same period (from 2012 to 2015) to compare reasonably.

Data Sources

- Country: https://finnhub.io/docs/api#country
- Forex: https://data.world/associatedpress/foreign-exchange-rates
- Population: https://www.kaggle.com/tanuprabhu/population-by-country-2020?select=population_by_country_2020.csv
- Inflation: https://data.worldbank.org/indicator/FP.CPI.TOTL.ZG?end=2019&start=2019&view=bar

II. Transformation (T)

This step is to import data, select columns needed, and clean/organize data. I unified "country" under lower cases for better merging on the next step.

Import and convert data into DataFrame

```
In [160]: csv = "Resources/quarterly-edited.csv"
exchange_rate = pd.read_csv(csv)
```

```
In [161]: exchange_rate = pd.DataFrame(exchange_rate)
exchange_rate.head()
```

Out[161]:

	country	unit	2001Q1	2001Q2	2001Q3	2001Q4	2002Q1	2002Q2	2002Q3	2002Q4	...	2014Q2	2014Q3	2014Q4	2015Q1
0	AFGHANISTAN	AFGHANI	78400.000	73000.000	71500.000	38200.000	35000.000	36807.000	38600.000	39000.000	...	57.390	57.05	57.90	57.34
1	ALBANIA	LEK	142.400	149.000	142.000	136.300	139.810	143.800	139.400	132.880	...	102.620	110.64	115.10	130.25
2	ALGERIA	DINAR	76.539	78.101	76.429	78.078	78.319	80.207	79.969	81.045	...	79.098	82.95	87.81	97.38
3	ANGOLA	KWANZA	19.706	19.706	20.389	30.044	34.320	40.236	46.178	53.307	...	98.000	98.00	104.00	104.00
4	ANTIGUA-BARBUDA	E. CARIBBEAN DOLLAR	2.700	2.700	2.700	2.700	2.700	2.700	2.700	2.700	...	2.700	2.70	2.70	2.70

Clean data (select, rename columns and unify “country” under lower case)

```
In [135]: #clean and rename data
exchange = exchange_rate[['country','2012Q3','2013Q3','2014Q3','2015Q3','2016Q3']].copy()
updated_exchange = exchange.dropna()
updated_exchange.head()
```

Out[135]:

	country	2012Q3	2013Q3	2014Q3	2015Q3	2016Q3
0	AFGHANISTAN	50.500	57.00	57.05	63.800	65.35
1	ALBANIA	107.900	104.55	110.64	123.990	122.77
2	ALGERIA	79.112	81.33	82.95	105.739	109.35
3	ANGOLA	95.000	95.00	98.00	145.000	170.00
4	ANTIGUA-BARBUDA	2.700	2.70	2.70	2.700	2.70

```
In [138]: updated_exchange = updated_exchange.rename(columns={"country":"country",
                                                "2012Q3":"fx_2012Q3",
                                                "2013Q3":"fx_2013Q3",
                                                "2014Q3":"fx_2014Q3",
                                                "2015Q3":"fx_2015Q3",
                                                "2016Q3":"fx_2016Q3"})
updated_exchange['country'] = updated_exchange['country'].str.lower()
```

```
In [139]: updated_exchange.head()
```

Out[139]:

	country	fx_2012Q3	fx_2013Q3	fx_2014Q3	fx_2015Q3	fx_2016Q3
0	afghanistan	50.500	57.00	57.05	63.800	65.35
1	albania	107.900	104.55	110.64	123.990	122.77
2	algeria	79.112	81.33	82.95	105.739	109.35
3	angola	95.000	95.00	98.00	145.000	170.00
4	antigua-barbuda	2.700	2.70	2.70	2.700	2.70

III. Load (L)

The last step is to load data into Postgres and merge the four tables by using SQL skills.

Load data into Postgres

Create database connection

```
In [145]: connection_string = "postgres:postgres@localhost:5432/country_db"
          engine = create_engine(f'postgresql://{connection_string}')

In [146]: engine.table_names()
Out[146]: ['country', 'population', 'inflation', 'forex']
```

Load DataFrames into database

```
In [110]: updated_country.to_sql(name='country', con=engine, if_exists='append', index=False)

In [148]: updated_exchange.to_sql(name='forex', con=engine, if_exists='append', index=False)

In [116]: updated_population.to_sql(name='population', con=engine, if_exists='append', index=False)

In [151]: updated_inflation.to_sql(name='inflation', con=engine, if_exists='append', index=False)
```

Create Table

```
7  CREATE TABLE forex (
8      country TEXT,
9      "fx_2012Q3" DECIMAL(10,2),
10     "fx_2013Q3" DECIMAL(10,2),
11     "fx_2014Q3" DECIMAL(10,2),
12     "fx_2015Q3" DECIMAL(10,2),
13     "fx_2016Q3" DECIMAL(10,2)
14 );
```

Merge Table

```
6  SELECT con.country, con.currency_code, pop.population, fx."fx_2012Q3", inf."cpi_2012",
7      fx."fx_2013Q3", inf."cpi_2013", fx."fx_2014Q3", inf."cpi_2014", fx."fx_2015Q3", inf."cpi_2015"
8  FROM country as con
9  INNER JOIN forex as fx
10 ON con.country = fx.country
11 INNER JOIN population as pop
12 ON con.country = pop.country
13 INNER JOIN inflation as inf
14 ON con.country = inf.country;
```