Maxwell Acha, Gary Whitney, David Griliopoulos

Birmingham University Data Analytics Bootcamp

Project 2 Report:

Abstract

Our team decided that we would compile different data sources on the countries of the world, so that a user could perhaps use this to investigate any statistical connections. We decided to use different countries' 'World Happiness Report Rankings' and 'Population Density'.

This would enable us to analyse the data to find any correlation between the two variables. It would it be interesting to find whether population density has a negative or positive impact on people's happiness, or if it has no connection at all.

Extract

We extracted the data in different formats from two different sources:

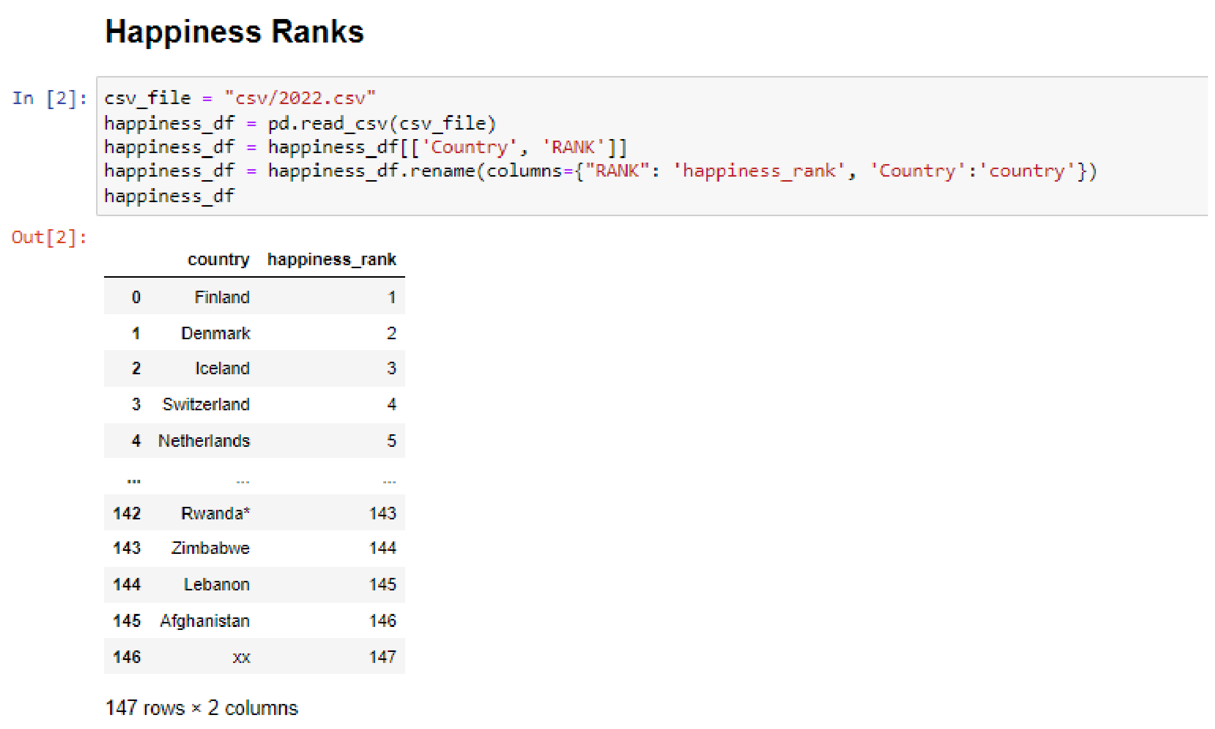
- A .csv regarding the 'happiness score' of various countries
- https://www.kaggle.com/datasets/mathurinache/world-happiness-report?select=2022.csv
- An html table of population density https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population_density

These were both converted into Pandas DataFrames by *pd.read_csv* and *pd.read_html* respectively. We had to search through the list of returned tables to find the correct table.

Transform

There was substantial cleaning and reduction that needed to be done to these two dataframes in order to make them fit together.

Both dataframes have a 'Country' column, however in both dataframes some countries were followed by an asterix ('*').



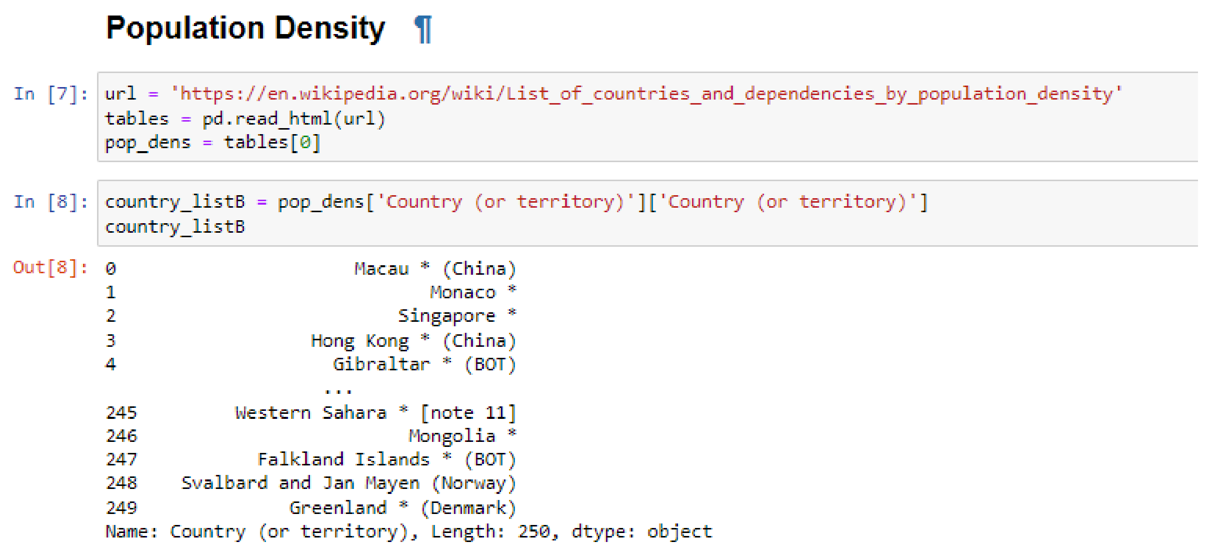
Happiness Ranks

```
In [2]: csv_file = "csv/2022.csv"
        happiness_df = pd.read_csv(csv_file)
        happiness_df = happiness_df[['Country', 'RANK']]
        happiness_df = happiness_df.rename(columns={"RANK": 'happiness_rank', 'Country':'country'})
        happiness_df
```

Out[2]:

	country	happiness_rank
0	Finland	1
1	Denmark	2
2	Iceland	3
3	Switzerland	4
4	Netherlands	5
...	...	...
142	Rwanda*	143
143	Zimbabwe	144
144	Lebanon	145
145	Afghanistan	146
146	xx	147

147 rows × 2 columns



Population Density ¶

```
In [7]: url = 'https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population_density'
        tables = pd.read_html(url)
        pop_dens = tables[0]
```

```
In [8]: country_listB = pop_dens['Country (or territory)']['Country (or territory)']
        country_listB
```

```
Out[8]: 0                        Macau * (China)
        1                              Monaco *
        2                           Singapore *
        3                    Hong Kong * (China)
        4                     Gibraltar * (BOT)
                                   ...
        245         Western Sahara * [note 11]
        246                          Mongolia *
        247            Falkland Islands * (BOT)
        248     Svalbard and Jan Mayen (Norway)
        249              Greenland * (Denmark)
        Name: Country (or territory), Length: 250, dtype: object
```

This, and any following text (some countries had an extra note in brackets afterwards), was removed by reading the countries, stripping off the unnecessary characters by partitioning on them, and then writing that clean country name to a new list. This list was then concatenated with the relevant numerical data from the original dataframe to make a new clean dataframes; columns were then renamed to match the names in the SQL tables.

The Population Density dataframe has more rows than the Happiness dataframe, as it also contains dependent territories (E.g “Svalbard and Jan Mayen (Norway)”) that do not have a separate value in the happiness survey. Users would be able to inner or outer join to keep or remove these anomalies as they see fit.

The happiness dataframe has many more rows than was necessary for our data, all of which were dropped, leaving only ‘Country’ and ‘Rank’, and Rank was also renamed to match the SQL table.

We dropped the Happiness score itself as it was recorded as a formatted string (e.g. “7.587”) not an integer. We could have converted the string to an integer, but we felt that the rank was more of an important measure because it gave more context than the happiness score alone.

Load

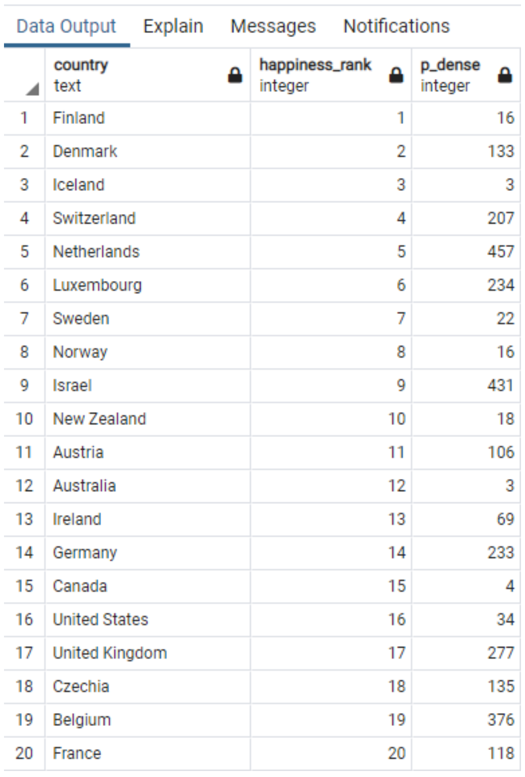
We created a database in PostgreSQL to contain the data, called ‘happy_density’, and inside created two tables that with columns that matched the dataframes:

- create table happiness(country text primary key, happiness_rank int);
- create table pop_dense(country text primary key, p_dense int);

We then loaded the dataframes into PostgreSQL using *sqlalchemy* and the dataframe.to_sql() function.

Using the code

- select happiness.country, happiness.happiness_rank, pop_dense.p_dense
  from happiness
  inner JOIN pop_dense
  ON happiness.country = pop_dense.country;

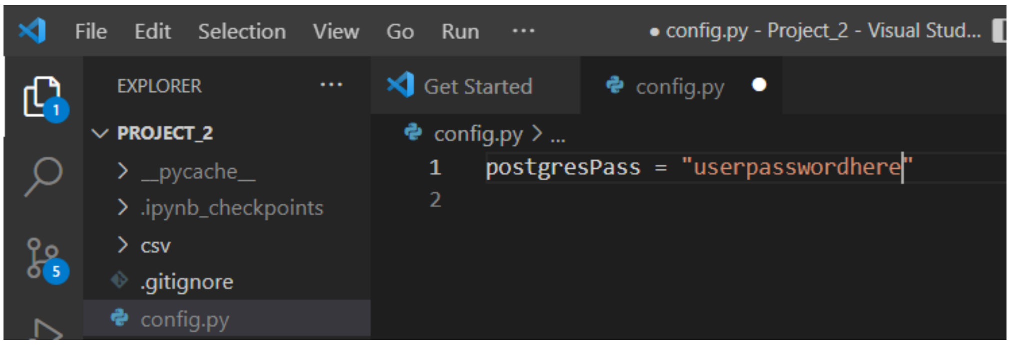


Data Output Explain Messages Notifications

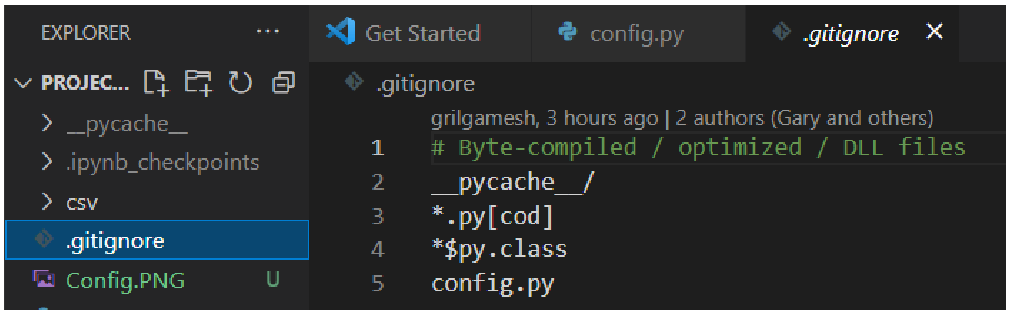
	country text	happiness_rank integer	p_dense integer
1	Finland	1	16
2	Denmark	2	133
3	Iceland	3	3
4	Switzerland	4	207
5	Netherlands	5	457
6	Luxembourg	6	234
7	Sweden	7	22
8	Norway	8	16
9	Israel	9	431
10	New Zealand	10	18
11	Austria	11	106
12	Australia	12	3
13	Ireland	13	69
14	Germany	14	233
15	Canada	15	4
16	United States	16	34
17	United Kingdom	17	277
18	Czechia	18	135
19	Belgium	19	376
20	France	20	118

We then built the relational database with the two tables by joining on the name of the country, returning 141 matched countries. This means there are a handful of countries that did not match, due to alternative naming conventions between the two data sources. Given more time, we would manually fix these errors, however 141 matches is enough to perform statistical analyses.

We elected to use a config.py file to store passwords. This file is ignored by the github repo to keep our own personal data private.



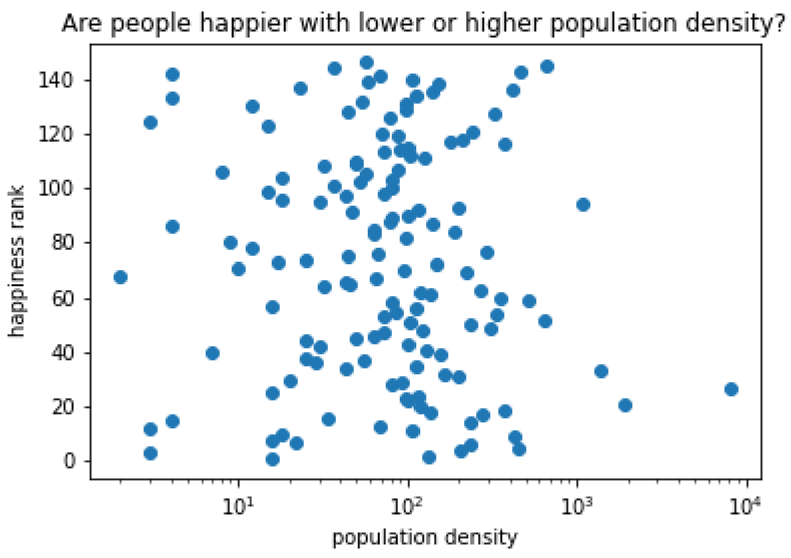
```
postgresPass = "userpasswordhere"
```



```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class
config.py
```

Conclusion

Finally, we created a scatter plot of the data:



There is clearly no correlation on a country-by-country basis between the two variables. Internationally, population density of a country has no predictive power on people’s happiness. However, there might be a connection at a smaller scale within countries and it would be interesting perhaps to do a region-by-region comparison.