**Save the Data (STD)**

**ETL Project**

**Analysis Report**

****

|  |  |
| --- | --- |
| **Prepared by:** | Yann Chye  Eamonn McCallum  Baili Li (Esther)  Michelle Hocking |
| **Date Due:** | 18/08/2020 |
| **Prepared for:** | Monash Analytics Bootcamp |

**Table of Contents**

[**1**](#_heading=h.1fob9te) **Executive Summary 3**

[**2**](#_heading=h.3znysh7) **Data Sources 3**

[**3**](#_heading=h.2et92p0) **Assumptions and Dependencies 4**

[Assumptions 4](#_heading=h.fr2gijlqv08a)

[Dependencies 4](#_heading=h.ozf0xix5nh45)

[**4**](#_heading=h.tyjcwt) **Method 5**

[Extract 5](#_heading=h.m91nisqwsyek)

[Transform 6](#_heading=h.8m9wgcifcjmn)

[Load 10](#_heading=h.9qwqxemdfdgf)

[**5**](#_heading=h.3dy6vkm) **Results 11**

[**6**](#_heading=h.1t3h5sf) **Limitations 11**

[**7**](#_heading=h.4d34og8) **Recommendations & Observations 11**

[**8**](#_heading=h.2s8eyo1) **References 12**

[**9**](#_heading=h.z2a9qiiwkv0f) **Appendix 12**

# Executive Summary

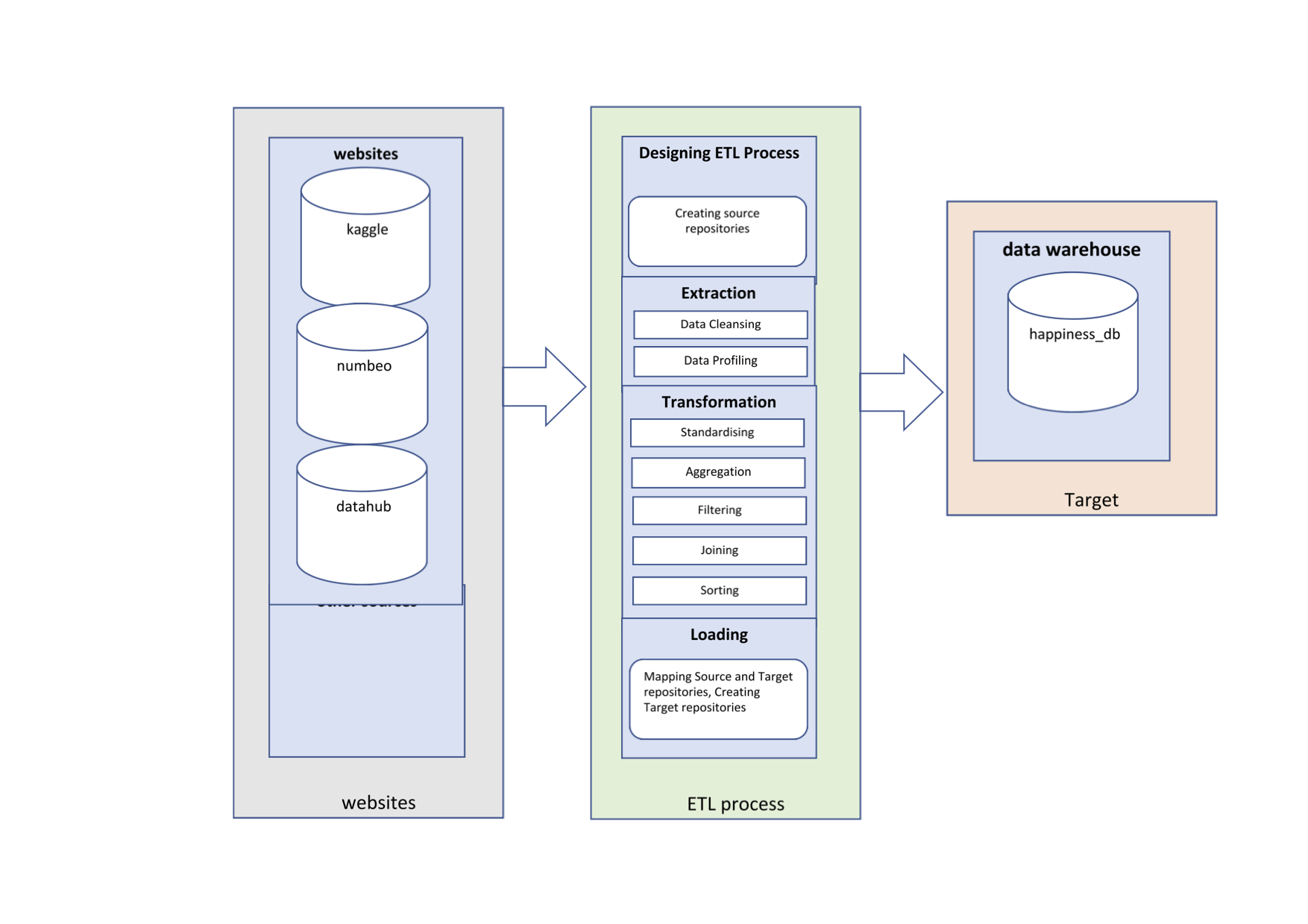
The aim of the project was to enable comparative analysis of a cost of living indices for a range of countries against a happiness score.

Our data was pulled from 2 primary sources: the World Happiness Report (obtained from kaggle[[1]](#footnote-0)), and a Cost of living index (COLI) from Numbeo[[2]](#footnote-1). Looking to enrich the dataset, we included an additional source, a list of countries and continents from datahub.io[[3]](#footnote-2) to enable interrogation by continent. Extraction of the data was done through a combination of flat file downloads, parsing html tables and working with an in-house api module. All data was then imported into pandas dataframes.

Prior to commencing our data transformation, the team created a map of the data source tables and their fields. In reviewing the data, we determined that as it was largely numeric[[4]](#footnote-3) a relational database was the preferred option. This would allow us to query and aggregate the data. We then agreed ‘business rules’ and set about transforming the data.

In bringing our datasets together, we followed a largely iterative and somewhat standardised ETL process (Fig 1.). This in itself was surprising, as from the outset the team worked in pairs, tackling each dataset independently and with little comparison on what was needed or how it was being done. As shown in our ETL diagram the end product was a relational database in PostgreSQL that we were able to successfully query, bringing together fields from all 3 tables.

**Fig 1. ETL diagram**

****

# Data Sources

There were 3 sources of data used for this project. The first source, located at [*https://www.kaggle.com/unsdsn/world-happiness*](https://www.kaggle.com/unsdsn/world-happiness)*,* provided data sets for the Happiness index. These datasets available were in a csv format, each file containing annual data and files were downloaded for a period of 5 years spanning 2015-2019.

The second source, which is for COLI data, can be found at <https://www.numbeo.com/>. Whilst the website does have an API, basic access to the database is at US$220/month. A second option was available, in the form of query-based html tables. A number of different slices of a single database are available and 2 views were selected, The Cost of Living Index by Country and The Property Price Index by Country. These were chosen based on the fields available being a function of disposable income.

The final data source was a listing of country and continent codes found on Datahub ( [*https://datahub.io/JohnSnowLabs/country-and-continent-codes-list*](https://datahub.io/JohnSnowLabs/country-and-continent-codes-list) . This was included to achieve 2 things. Firstly, including a 3-letter country code as a common index eliminates the inherent difficulties of handling country names. In addition, the inclusion of the continent field provides another layer of analysis through the aggregation of the data by continent.

# Assumptions and Dependencies

# Assumptions

The key overriding assumption in using all of our data sources is that they are reliable and quantitatively sound.

More particularly, with the Happiness index significant weight was given to the providence of the report, it being published by the United Nations (UN) (<https://worldhappiness.report/>). We recognised that a weighting by population has been applied to the data prior to publication and that this method is not transparent. We did not believe however that this would adversely affect our use of the data as the country and population features were not within the scope of the analysis.

It was also noted that the report is not comprehensive. The Report indexes 150+ countries of the 195 countries in the world today. We were confident in using this report as the sample size was nearly 80% of the total countries. An assumption was made that countries not covered were most likely too small to impact the end analysis materially.

In relation to the “COLI” data, a number of factors were taken into consideration in assessing if it should be used. Firstly, Numbeo describes themselves as “the world’s largest cost of living database”[[5]](#footnote-4). Data is both manually collected from “websites of supermarkets, taxi company websites, governmental institutions, newspaper articles, other surveys, etc”6, as well as crowd-sourced from global resources. The site does however have over 534,266 contributors1 providing local information, although there is no information about contributor credentials. As data is available city based, it is also “contribution-centric”, reflecting what is contributed. This naturally brings an inherent bias.

Additionally, cost of living indices once aggregated to a country level, may be heavily weighted towards city rather than regional areas. Also, in cities where there are few contributors, it would be difficult to assume the data was a true representation of the ‘real’ cost of living. Numbeo is however seen and referenced as a reputable source for cost of living data including The World Bank, Forbes, The Age and Time Magazine[[6]](#footnote-5) . Given this, an assumption was made that the quality and reliability of crowd wisdom data was of an acceptable level for use.

Extending from this was also a question around the reliability of the index calculations. All indices are stated in relative terms to New York City prices ie New York City is 100%. The database is also an accumulation of both user input and manually collected data from authoritative sources which is entered twice per year at city level. Country aggregation is done using all cities in that country weighted by the number of contributors. Weighing this up with a lack of other available cost of living indexes, the decision was made that there was sufficient rigour and data training techniques applied to the data inputs into the site’s database to take it as reliable categorical data.

From the numbeo tables available, we selected 2 specific tables: the COLI by Country, which listed 135 countries and the Property Index by Country table. The decision to include the property data was seen as an enrichment of the cost of living data, providing indices on affordability and price:income ratios. Both are indexed on the country name and we assumed coming from the same database we would be able to join the different slices 1:1 ie with no reduction in the overall country number.

From an overall perspective it was assumed that a sufficiently high number of the same countries would be in both the happiness index dataset and the COLI datasets to construct a database of ‘adequate’ size.

A standard approach to dropping data was taken for all datasets, in that where there was not 5 years of data available for either the measure or the country, the related data was dropped.

# Dependencies

Dependencies for running the ETL process include -

1. Python version 3.x (minimum)
2. Python modules[[7]](#footnote-6):
   1. Splinter
   2. Webdriver-manager
   3. BeautifulSoup4
   4. pandas
   5. Numpy
   6. kaggle
   7. SQLAlchemy
   8. SQLAlchemy\_utils
   9. os (will be included in python)
   10. time (will be included in python)
3. A [kaggle](https://www.kaggle.com/) account, and an ‘api\_key\_template.py’ file containing your kaggle username and a user-specific kaggle api key (refer to <https://github.com/Kaggle/kaggle-api>). This is needed to be able to interact with the kaggle api to extract their dataset. A template ‘*api\_key\_template.py*’ file is available in the *‘00\_config’* folder. Populate it with your username and api key
4. [postgreSQL](https://www.postgresql.org/), with a ‘password\_template.py’ file containing your password. A template *‘password\_template.py’* file is available in the *‘00\_config’*folder. Populate it with your postgreSQL username and password (default username is ‘postgres’)
5. Required folder structure as shown in the Appendix: Github Repository map. ETL process is done through the respective jupyter notebooks, and should be run in the order of :
   1. 01 (extract) > 02 (transform) > 03 (load)
   2. 02\_transform\_country.ipynb needs to be run before 02\_transform\_coli.ipynb and 02\_transform\_happiness.ipynb

# Method

# Extract

Discussion of the extraction process is best explained in each part as it relates to the 3 separate data sources:

1. **Country list**

Country data was extracted from Datahub using both Splinter and BeautifulSoup to navigate the website and read the country and continent codes into a csv file (refer Appendix).

1. **Happiness index**

Kaggle provides 2 methods for accessing the **Happiness index** data on its site, an in-house api or direct download of csv files. As we chose to use the kaggle api[[8]](#footnote-7), we first needed to install kaggle’s API module. The site provides both zip and separate csv file options and we chose to unzip directly when downloading from the api. A jupyter notebook of the extraction code is available in the Github Repository (refer Appendix).

1. **Cost of Living data**

There were 2 relevant datasets extracted from the Numbeo site, **Cost of Living index** and the **Property index**. In both cases, data was in the form of html tables, which were parsed into Pandas and converted into individual dataframes for each relevant year. A jupyter notebook of the extraction code is available in the Github Repository (refer Appendix).

# Transform

**Referential Integrity**

In any transformation, referential integrity can become a significant issue in bringing datasets together from multiple sources. Understanding this, the team’s first step was to list the source tables and their fields. A draft entity relationship diagram (ERD) in QuickERD showed a natural link (the country) between not only the same source tables (**COLI** and the **Property index**) but also with the **Happiness index**. In addition, the data was predominantly numerical in nature, which would make querying and grouping across tables relatively easy. As such a relational database was chosen for the final load. This decision shaped our approach to the transformation of the data, both within individual datasets, as well as across the group of datasets as a whole. A final consideration before embarking on transformation was whether to include additional fields for summarisation and/ or further aggregated the data. At this point we made the decision to leave data untouched rather than risk changing the meaning of the data or worse still, making the data meaningless.

The transformation processes the team followed can be summarised in the following table.

**Table 1. Transformations within each dataset**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Transformation Action taken** | **Country List** | **World Happiness index** | **Cost of Living index**  **(COLI)** | **Property index** |
| Format standardization | **✓** | **✓** | **✓** | **✓** |
| Cleaning | **✓** | **✓** | **✓** | **✓** |
| Remove duplication | **✓** | **✓** | ‒ | ‒ |
| Constraints implementation | **‒** | **✓** | **✓** | **✓** |
| Decoding of fields | **✓** | **✓** | **‒** | **‒** |
| Merging of information | **‒** | **✓** | **✓** | **✓** |
| Splitting single fields | **✓** | **‒** | **‒** | **‒** |
| Summarisation | **⤬** | **⤬** | **⤬** | **⤬** |
| Inconsistent/missing timing | **‒** | **✓** | **✓** | **✓** |
| Misspelling | **✓** | **‒** | **‒** | **‒** |

**✓** done **⤬** considered but not required‒not required

**Country List**

Once imported into a dataframe, transforming the country data included:

* cleaning the country names. This was done through a process of string splitting, removing trailing whitespace and replacing misspelling/non-alphanumeric character where necessary,
* removing any duplicate country names or country codes,
* dropping any null values for either country names or the country code,
* removing redundant columns, and
* renaming columns to be more meaningful and ‘database friendly’.

**World Happiness index**

Having extracted the Happiness index data from the website, the first step in the transformation was unzipping the data and creating 5 separate pandas dataframes, one for each year. From here, all 5 dataframes were standardised by:

* dropping columns that were not consistent,
* renaming of column names to be the same, and
* reordering columns to reflect the same sequence.

The dataframes were then merged into a single dataframe, with an additional column for the relevant [**Year**] added.

At this point we made the decision to constrain the data, excluding any country with less than 5 years’ worth of data. Midway through the process inconsistencies were identified in 5 country names and these were manually replaced througha find and replace using a hard-coded list.

Following on from this, the transformed Country data was then imported as a separate dataframe. In order to standardise country names across both datasets, the country names in the **Happiness index** data were replaced with their respective 3-letter country code. Further analysis was done of the Happiness index data to ensure all that:

* each country had 5 years’ worth of data,
* all countries listed in the **Happiness index** data were found within the **Country** data, and
* any missing values were dropped.

Some simple statistics confirmed all index values were numerical, within expected ranges and that values were ‘reasonable’.

**COLI and Property index**

Given both the **COLI** and **Property index** data are slices of the same database, it was expected that minimal transformation would be required for both datasets and that the process would be iterative. We first imported the COLI data and discovered however that the total number of countries each year varied between 115 and 122. Handling this in a similar way to **Happiness index** data, countries with less than 5 years of data were dropped, giving a final result of 108 unique countries.

Whilst our source data provided ranking information for countries in each year during scraping it did not load into the dataframe. With these values being easily replicable, the rank information was dropped and recreated referencing the dataframe index. As with **Happiness index**, the following was also done:

* inserting a time reference [Year],
* dropping redundant columns,
* renaming columns to be ‘database friendly’, and
* reordering columns to reflect a logical order.

The same process as the **Happiness index** was also followed for importing and transforming the country name before applying some simple statistical inspections to confirm the integrity of the dataset.

In transforming the **Property index** data, we discovered yet again variation in the total number of countries included each year. On completion of the same transformations applied to the COLI dataset, the total number of countries remaining was 88. This required us to apply an additional filter to the COLI dataset to ensure only data for the same countries were included in each.

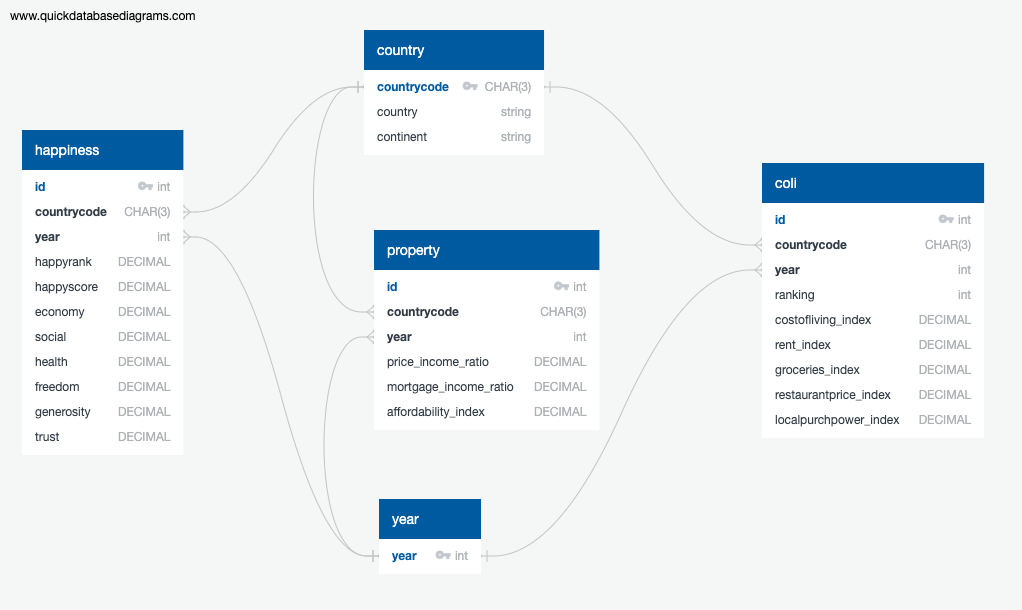
Final outputs for the process were all csv files (refer Appendix for location and further reference).

# Load

On completion of the transformation, the ERD for a relational database in postgreSQL was updated and finalised (**Fig. 2**). Primary keys in the ‘**country**’ and ‘y**ear**’ tables (**‘countrycode**’ and ‘**year**’ respectively) were assigned, which became the foreign keys for the ‘**happiness’**, ‘**property**’, and ‘**coli**’ tables.

The creation of the database, table structure and constraints, and data import was all done within a jupyter notebook with SQLAlchemy. Table columns and their constraints (eg. primary key, foreign key etc.) were created, taking care to import data in the correct order and accounting for headers. Also, worth noting is that where a table already existed, there is provision in the code to drop the table and recreate a new one. Steps taken were:

1. a database named ‘**country\_happyCOL**’ was created
2. the transformed Country data was imported
3. an additional table called ‘years’ (2015, 2016, 2017, 2018, 2019) was also created and imported
4. the transformed Happiness, COLI, and Property data sets were separately imported
5. checks were conducted to ensure the imports were successful

 **Fig. 2 Entity Relationship Diagram (ERD) for the Final Database**

# Results

With the conclusion of the ETL process, using the database generated . The below five questions posed have related SQL queries and result CSV save under folder 04\_results. The questions were:

1. Which continent does the top 10 happiest countries come from in the year 2019?
2. Are the Top 3 happiest countries in 2015 the same in 2019?
3. What is the average happiness, COLI and property affordability per continent from 2015 to 2019
4. What is the correlation between happiness and COLI group by years
5. What is the correlation between GDP from the happiness index and local purchasing power and correlation happiness score and local purchasing power

Based on the results provided, we observed that:

1. In the year 2019, out of the top 10 happiest countries, 8 were from Europe, 1 from North America and 1 from oceania.
2. In the year 2015, the top 3 happiest countries are CHE, ISL, DNK; in the year 2016 are DNK, CHE, ISL, in the year 2017 are: NOR, DNK, ISL; in the year 2018 are:FIN, NOR, DNK; in the year 2019 are: FIN, DNK, NOR, we can see country DNK which is Denmark stays at top 3 ranking consistently.
3. Shows an aggregated table with averaged scores per continent over time. This table could be further analysed from further correlations but was explored in this project. The results are captured in a csv file
4. This shows a positive correlation slightly increase from the lowest point in 2016 of 0.622 to 0.702 in 2019. the The results are captured in a csv file
5. The query showed that there is a positive with higher GDP having greater local purchasing with a coefficient range of 0.748 - 0.807 and a positive correlation with the happiness score and greater local purchasing power coefficient range 0.670 - 0.713 which is better visualised in the related csv

# Key Takeouts & Observations

Having completed the project and reflecting on the process, we made the following observations:

* regardless of differences in how data is sourced, transformation of any dataset follows the same basic approach of cleaning, sort and arranging
* being extremely iterative, transformation lends itself to the use of functions. These were not used extensively in this project however would be an opportunity for future projects
* never under-value the benefits of ‘eye-balling’ the data – there were plenty of ‘sniff tests’ that showed things need to be investigated further
* there will always be a manual element in the cleaning process. In our case it was the 5 country names, and finally
* many datasets can be combined but not all datasets should be combined. Joining data should be done only to give users the ability for more complex interrogation of data (e.g. querying country happiness by continent)

# References

Cost of Living. (n.d.). Retrieved August 11, 2020, from https://www.numbeo.com/

Country and Continent Codes List. (n.d.). Retrieved August 13, 2020, from https://datahub.io/JohnSnowLabs/country-and-continent-codes-list

ETL - Understanding It and Effectively Using It. (2019, January 07). Retrieved August 15, 2020, from <https://medium.com/hashmapinc/etl-understanding-it-and-effectively-using-it-f827a5b3e54d>

Home. (2020, March 20). Retrieved August 14, 2020, from https://worldhappiness.report/

Kaggle API - The Missing Python Documentation. (2020, February 20). Retrieved August 15, 2020, from https://technowhisp.com/kaggle-api-python-documentation/

Kaggle/kaggle-api. (n.d.). Retrieved August 14, 2020, from https://github.com/Kaggle/kaggle-api

World Happiness Report. (2019, November 27). Retrieved August 11, 2020, from <https://www.kaggle.com/unsdsn/world-happiness>

# Appendix

**GitHub Repository**

|  |  |  |
| --- | --- | --- |
| **Directory** | **Description** | **From Notebook(s)** |
| **00\_config** | **Python and Text file to locally store API Keys and Passwords** |  |
| **01\_extract\_Coli** | **Files extracted by scraping Cost of Living Data source** | **01\_extract\_coli.ipynb** |
| **01\_extract\_country** | **Files extracted by scraping country data** | **01\_extract\_country.ipynb** |
| **01\_extract\_happiness** | **Files extracted by scraping and unzipped from Kaggle** | **01\_extract\_happiness.ipynb** |
| **02\_transform\_coli** | **Cost of Living Cleaned CSV Files** | **02\_transform\_coli.ipynb** |
| **02\_transform\_country** | **Country Names/Code Cleaned CSV Files** | **02\_transform\_country.ipynb** |
| **02\_transform\_happiness** | **Happiness Index Cleaned CSV Files** | **02\_transform\_happiness.ipynb** |
| **03\_load** | **Contains our Entity Relationship Diagram, code for loading our database in notebook** | **03\_load.ipynb** |
| **04\_results** | **Result SQL queries and output CSVs** |  |
| **report** | **Contains report Document** |  |

# 

1. World Happiness Report. (2019, November 27). Retrieved August 11, 2020, from https://www.kaggle.com/unsdsn/world-happiness [↑](#footnote-ref-0)
2. Cost of Living. (n.d.). Retrieved August 11, 2020, from <https://www.numbeo.com/> [↑](#footnote-ref-1)
3. Country and Continent Codes List. (n.d.). Retrieved August 13, 2020, from https://datahub.io/JohnSnowLabs/country-and-continent-codes-list [↑](#footnote-ref-2)
4. Indexes and ranking are compound measures known as composite statistics. They are statistical measures of change and the end result of aggregations of other indicators [↑](#footnote-ref-3)
5. Cost of Living. (n.d.). Retrieved August 12, 2020, from https://www.numbeo.com/common/motivation\_and\_methodology.jsp [↑](#footnote-ref-4)
6. About This Website. (n.d.). Retrieved August 12, 2020, from https://www.numbeo.com/common/about.jsp [↑](#footnote-ref-5)
7. See ‘/00\_config/requirements.txt’ for module list. Easiest way to install modules is using PIP package manager - *pip install -r requirements.txt* [↑](#footnote-ref-6)
8. Kaggle. (n.d.). Kaggle/kaggle-api. Retrieved August 14, 2020, from https://github.com/Kaggle/kaggle-api [↑](#footnote-ref-7)