Merged_Data Metadata

Data Sources

World Happiness

Source & Metadata:

- Data collected from: https://www.kaggle.com/mathurinache/world-happiness-report?select=2020.csv
- Source: https://worldhappiness.report/ed/2017/
- Metadata: https://s3.amazonaws.com/happiness-report/2017/StatisticalAppendixWHR2017.pdf

This dataset originating from the World Happiness Report contains life-evaluation indicators by country which reflect how happy citizens perceive themselves to be (Helliwell, J., Layard, R., & Sachs, J. 2017). The data acquired from Kaggle was in csv format and was 155 by 12. It contained 11 indicator columns with 155 unique country entries. The data is licensed by CC0 1.0 i.e. Public Domain Dedication. The data can be downloaded here, https://www.kaggle.com/mathurinache/world-happiness-report?select=2020.csv, the complete written report with metadata can be found here, https://worldhappiness.report/ed/2017/.

Human Freedom Index

Source & Metadata:
HFI Report:
https://www.cato.org/human-freedom-inedex-new
Data Source:
https://www.kaggle.com/gsutters/the-human-freedom-index?select=hfi_cc_2019.csv

This data source was originated from the Human Freedom Index Report and aims to capture a broad but rational accurate picture of overall freedom in the world (Vásquez, I., & Porc̆nik, T., 2019). The dataset we are using includes data from 2011 to 2017 and has 120 columns by 1621 rows. The first four columns deal with the year, country, iso code and region whilst the rest all contain their respective freedoms that make up the overall freedom index for each country, such as homicide, fatalities, criminality etc. The dataset was sourced from Kaggle and is licensed under Open Data Commons Open Database License which means it is free to use.

GNI per capita by Purchasing Power Parity

Source & Metadata:

Data Source:
https://data.worldbank.org/indicator/NY.GNP.PCAP.PP.KD
Source Metadata:
https://datahelpdesk.worldbank.org/knowledgebase/articles/906531-methodologies#:~:text=National%20accounts%20and%20balance%20of,those%20reported%20in%20official%20sources.

This dataset covers the Gross National Income (GNI) per capita in 2017 and originates from current reports gathered by the Bank's country management units and official sources. GNI works to calculate the total income earned by a country's people and business, and includes both investment income and money received from abroad. Sourced from (International Comparison Program, World Bank | World Development Indicators database, World Bank | Eurostat-OECD PPP Programme), the .csv file was procured from Kaggle and contained 264 regional entries and 64 columns denoting countries and years. The data is licenced by CC-BY 4.0, which allows users to copy, modify and distribute data in any format for any purpose, including commercial use.

Utility Dataset: ISO 3166-2 Country Codes

Data Source:
https://www.kaggle.com/juanumusic/countries-iso-codes/metadata
Metadata:
https://www.iso.org/iso-3166-country-codes.html

This ISO Codes dataset contains a universal international standard code system for identifying unique countries and originated from the International Organisation for Standardisation. The dataset is under CC0 1.0 Universal (CC0 1.0) Public Domain Dedication which allows us to freely copy and modify without permission.

Data Wrangling

Initially we had to merge our three data sources of human freedom index, world happiness and GNI per capita. As we were only seeking to compare the data for the year of 2017, we began by extracting only 2017 data from the human freedom index as this contained data up to 2019. Which was then followed by the same process for GNI per capita.

```
#Extract only 2017 Data for Freedom Index
index_not_2017 = freedom_index[freedom_index["year"] != 2017].index
freedom_index.drop(index_not_2017, inplace=True)

#Extract only 2017 Data from GNI/capita PPP
gni_capita_2017 = gni_capita.iloc[:,[0,1,61]]
print(gni_capita_2017)
```

Now that the data had been extracted and established we sought to merge the three sources under one file. Due to each data set having slight differences in the name of their “country” column, e.g “countries” or “country” we uniformly named them “countries” and the ISO code column as “ISO codes.”

However, the same countries often had different names between different datasets. Therefore, we tried to combine based on standard ISO codes. However, whilst both GNI per capita and the Human freedom index contained ISO codes, the world happiness report did not. Therefore by importing a list of ISO codes we were able to create a new column in world happiness that contained the ISO codes matched based on country name. Note that in doing so, countries which did not match the official common name had their entire row removed as it was unrealistic to manually search and change country names.

```
#Extract only ISO code & Countries from iso_code
iso_code = iso_code.iloc[:,[0,2]]

#Change Headings of all datasets to Match Freedom Index Dataset
world_happiness = world_happiness.rename(columns={"Country":"countries"})
gni_capita_2017 = gni_capita_2017.rename(columns={"Country Code":"ISO_code",
                                                  "Country Name":"countries",
                                                  "2017":"GNI_capita"})
iso_code = iso_code.rename(columns={"English short name lower case":"countries",
                                    "Alpha-3 code":"ISO_code"})

#Add ISO Code to World Happiness (It doesn't originally have ISO code)
world_happiness = pd.merge(world_happiness, iso_code, on=["countries"])
```

We then deleted country columns from 2 of the 3 datasets that would cause repeated columns when merged. We changed all the empty values in the datasets from their default value to NaN so that we had a uniform standard in our final merged dataset. When we did this, we carefully ensured that the expected range of columns did not include the empty values that we changed. We then successfully merged all the data sets based on the ISO codes column into one file named merged_data. This step also implicitly removed any country's data that was not present in all three data sets.

```
#Drops duplicate country columns from Freedom Index, GNI/capita
del gni_capita_2017['countries']
del world_happiness['countries']

#Changes all empty values in all datasets to NaN (i.e. empty)
gni_capita_2017 = gni_capita_2017.replace('', np.nan)
world_happiness = world_happiness.replace(0, np.nan)
freedom_index = freedom_index.replace('-', np.nan)

#Merges data based on matching ISO Codes to avoid errors in different spelling of countries
#Implicitly removes country data that is not present in all three sources
merged_data = pd.merge(freedom_index, world_happiness, on=["ISO_code"])
merged_data = pd.merge(merged_data, gni_capita_2017, on=["ISO_code"])
print(merged_data)
```

The next step involved cleaning the merged dataset. Firstly to select the data that was relevant to our analysis we selected columns containing: "year", "ISO_code", "countries", "region", "hf_score", "hf_rank", "Happiness.Rank", "Happiness.Score", "GNI_capita ". "Year", "ISO code", countries and regions all serve to pinpoint where the country is in the world geographically and for the year we are collecting the data for. "Hf_score" and "Happiness.Score" are all values from 0-10. "hf_rank" and "Happiness.rank" order the countries from the highest to lowest in each respective score. Whilst GNI_capita" is a measure of a country's income calculated using the World Bank Atlas Method.

```
#Selects columns that are relevant to our analysis
merged_data = merged_data[["year", "ISO_code", "countries", "region", "hf_score",
                           "hf_rank", "Happiness.Rank", "Happiness.Score", "GNI_capita"]]
```

For any row that had NaN i.e. missing value in any of the columns we decided to remove the entire entry. This is because the columns we have are crucial for what we want to compare and analyse. Whilst the likelihood of duplicate entries were low, we tested for this and removed any duplicate rows which would give us inaccurate summaries. There were 246 unique countries identified by the ISO code dataset and due to limited data available in each of the three remaining datasets the final number of unique entries in our merged data is 133.

```
#Remove row entries that have empty cells in any of the columns
merged_data.dropna(inplace=True)

#Remove duplicates
merged_data.drop_duplicates()
```

After this, we then proceeded to check the remaining data values. We first tested if all the values were within their expected range, and found all values to be appropriate. These ranges were acquired from each of the datasets' metadata. For example, all the columns for scores would require their values to be from 0 to 10, thus if any were not, the console would return false.

```
#Checks if all data values are within their expected ranges: returns True if data is within ranges
print((merged_data["hf_score"].astype(float) >= 0).all() and (merged_data["hf_score"].astype(float) <= 10).all())
#Between 0 and 10 inclusive
print((merged_data["Happiness.Score"].astype(float) >= 0).all() and (merged_data["Happiness.Score"].astype(float) <=
10).all()) #Between 0 and 10 inclusive
print((merged_data["hf_rank"].astype(float) >= 1).all() and (merged_data["hf_rank"].astype(float) <= 159).all())
#Between 1 and 159 inclusive
print((merged_data["Happiness.Rank"].astype(float) >= 1).all() and (merged_data["Happiness.Rank"].astype(float) <=
155).all()) #Between 1 and 155 inclusive
print((merged_data["GNI_capita"].astype(float) >= 0).all()) #Ensuring non-negative values
```

Finally, the data was scanned for any special characters, which ensured that all values were in the expected data type i.e. numeric/alphanumeric.alphabetical.

```
#This function check every item of a given column in a given dataframe for special characters that shouldn't be
present and returns a bool
def check_special(column, dataframe):
    special_char = True
    string_check= re.compile('[@_!#$%^&*()<>?/\|}{~:]')
    for item in dataframe[column]:
        if string_check.search(str(item)) == None:
            special_char = False
        else:
            special_char = True
    return special_char

#The following for loop checks all code to ensure the correct types of characters are present for each column i.e.
alphabetic, numeric
for column_name in merged_data:
    if check_special(column_name, merged_data) == True:
        print(column_name, ": some entries have special characters that are invalid" )
    else:
        if pd.to_numeric(merged_data[column_name], errors='coerce').notnull().all() == True:
            print(column_name, ": is entirely numeric")
        elif not merged_data[column_name].str.isnumeric().all() == True:
            print(column_name, " is entirely alphabetical letters or normal punctuation")
        else:
            print(column_name, ": column contains inappropriately mixed data")
```

Metadata

Column Name	year	ISO_code	countries	region	hf_score
Values Type	Numeric	Alphabetic	Alphabetic	Alphabetic and Special Characters	Numeric
Format	integer	string	string	string	float
Missing Value	n/a	n/a	n/a	n/a	n/a
Meaning	Year that all the data was collected. All cell values read at ‘2017’.	Standardised ISO 3166-1 alpha-3 codes of each country and region. Sourced from the International Organisation for Standardisation.	Standardised names of all countries in English.	Each country is categorised under either a continental or sub-continental region. Regions include ‘Caucasus and Central Asia’, ‘East Asia’, ‘Eastern Europe’, ‘Latin America and the Caribbean’, ‘Middle East and North Africa’, ‘North America’, ‘Oceania’, ‘South Asia’, ‘Sub-Saharan Africa’ and ‘Western Europe’.	This is the human freedom score of each country. The human freedom score is the average of numerous factors and variables measured by the Cato Institute, Fraser Institute, and the Friedrich Naumann Foundation for Freedom. The methods and calculations are described here: https://www.cato.org/human-freedom-inedex-new

Column Name	hf_rank	Happiness.Rank	Happiness.Score	GNI_capita
Values Type	Numeric	Numeric	Numeric	Numeric
Format	integer	integer	float	float
Missing Value	n/a	n/a	n/a	n/a
Meaning	The ordinal rank of countries based on their human freedom score. ‘1’ indicates the highest human freedom score and thus the most ‘free’ country.	The ordinal rank of countries based on their happiness score. ‘1’ indicates the highest happiness score and thus the most ‘happy’ country.	This is a measurement of the happiness of countries around the world. It is the national average of one question in the World Gallup Poll (GWP). The English translation can be found below: “Please imagine a ladder, with steps numbered from 0 at the bottom to 10 at the top. The top of the ladder represents the best possible life for you and the bottom of the ladder represents the worst possible life for you. On which step of the ladder would you say you personally feel you stand at this time?” This was sourced from the World Happiness Report published by Sustainable Development Solutions Network, which can be found here: https://worldhappiness.report/ed/2017/	This is the Gross National Income (GNI) per capita in 2017 and originates from current reports gathered by the Bank's country management units and official sources. It has been converted to international dollars using Purchasing Power Parity (PPP) rates, which compares the world’s income against the US dollar. It can be sourced from the World Bank here: https://data.worldbank.org/indicator/NY.GNP.PCAP.PP.KD