Capstone Project 1: Predictors of global happiness

Data Source

World Happiness
The Sustainable Development Solutions Network (SDSN) has published the World Happiness Report yearly since 2012. The scores and rankings for the countries in the report were based on data from the Gallup World Poll.

Gender Development and Inequality
The United Nations Development Programme produces a report that includes two datasets with statistics on the disparities between the two genders based on various socioeconomic factors..

Population Density
The dataset was sourced and extracted from the World Bank. The full dataset has over 50 years of data from 155 countries.

Economic Freedom
The data from 186 countries was distributed by The Heritage Foundation. The scores are based on 12 freedoms (i.e. property rights, business freedom, monetary freedom, etc.).

Raw Data and Content Structure

All datasets were in either CSV or XLS format. Each row in every dataset contains scores, factors, and identifying information from a unique country. The columns included for each dataset are:

World Happiness (Figure 2.2 tab)

- Country
- Happiness score
- Whisker-high
- Whisker-low
- Explained by: GDP per capita
- Explained by: Social support
- Explained by: Healthy life expectancy
- Explained by: Freedom to make life choices
- Explained by: Generosity
- Explained by:
- Perceptions of corruption
- Dystopia (2.33) + residual

Gender Development

- HDI Rank (2018)
- Country
- 1995
- 2000
- 2005
- 2010
- 2011
- 2012
- 2013
- 2014
- 2015
- 2016
- 2017
- 2018

Gender Inequality

- HDI Rank (2018)
- Country
- 1995
- 2000
- 2005
- 2010
- 2011
- 2012
- 2013
- 2014
- 2015
- 2016
- 2017
- 2018

Population Density

- Country Name
- Country Code
- Indicator Name
- Indicator Code
- 1960
- 1961
- 1962
- 1963
- 1964
- 1965
- 1966
- 1967
- 1968
- 1969
- 1970
- 1971
- 1972
- 1973
- 1974
- 1975
- 1976
- 1977
- 1978
- 1979
- 1980
- 1981
- 1982
- 1983
- 1984
- 1985
- 1986
- 1987
- 1988
- 1989
- 1990
- 1991
- 1992
- 1993
- 1994
- 1995
- 1996
- 1997
- 1998
- 1999
- 2000

- 2001
- 2002
- 2003
- 2004
- 2005
- 2006
- 2007
- 2008
- 2009
- 2010
- 2011
- 2012
- 2013
- 2014
- 2015
- 2016
- 2017
- 2018

Economic Freedom

- CountryID
- Country Name
- WEBNAME
- Region
- World Rank
- Region Rank
- 2019 Score
- Property Rights
- Judicial Effectiveness
- Government Integrity
- Tax Burden
- Govt Spending
- Fiscal Health
- Business Freedom
- Labor Freedom
- Monetary Freedom
- Trade Freedom
- Investment Freedom
- Financial Freedom
- Tariff Rate (%)
- Income Tax Rate (%)
- Corporate Tax Rate (%)
- Tax Burden % of GDP
- Govt Expenditure % of GDP
- Country
- Population (Millions)
- GDP (Billions, PPP)
- GDP Growth Rate (%)
- 5 Year GDP Growth Rate (%)
- GDP per Capita (PPP)
- Unemployment (%)
- Inflation (%)
- FDI Inflow (Millions)
- Public Debt (% of GDP)

Data Wrangling Steps

Note: Please reference my github account for a detailed overview of data wrangling code and steps (http://bit.ly/33try77)

1. Download Data

   All of the datasets were downloaded from their respective site and stored in my Capstone Project directory.

2. Raw Data Inspection

   Since the majority of my data was sourced from UN based sources, it is relatively organized and in a logical and tidy format. There are column headers and each country is represented by a row. Since I am mostly concerned with the countries and their

corresponding score for years 2015-2018, much of the columns in each dataset I won't be using.

3. Import Data

   For XLS files, I used Pandas read_excel function with the sheet_name and usecols parameter to load the datasets as a dataframe from a specific sheet and the specific columns that I needed. CSV files were imported as dataframes using the read_csv function in Pandas.

4. Inspect in Jupyter

   Using the head, tail, and info methods I can see that datasets have differing numbers of countries and null values in several cells.

5. Cleaning and Merging Data
   a. Since I need to merge the dataframes based on Country, I wanted to ensure that there were no leading or trailing spaces in the Country columns that would complicate combining the dataframes. I used the strip() method to accomplish this.
   b. I also wanted to rename all columns so that there are no spaces in the name to facilitate filtering and slicing
   c. The world happiness and economic freedom datasets had to be downloaded individually by year. I wanted to combine all the years for the respective feature into one single dataset to make concatenation easier.
   d. In both the World Happiness and Economic Freedom datasets, a few of the country names varied by year. For example in 2015 Trinidad and Tobago was spelled with "&" instead of "and". I updated the handful of countries in each dataset with the same naming convention.
   e. Once the country names were consistent, I merged the years for the respective features together. Each dataframe had a Country column and 4 additional columns for the score for each year.
   f. I removed all empty rows using the .drop() method.
   g. Since I knew that I wanted to do some analysis based on region and create a map of the scores, I added ISO A3 codes and regions to the world happiness dataset since I planned to merge the other scores to that one. I did this importing the ISO A3 spreadsheet as a dataframe and merging it to my world happiness dataframe.
   h. The gender development and gender inequality datasets had multiple empty columns. I used the .dropna() method to remove those. Additionally, several empty cells were annotated with "..". I used the .replace() method to substitute NaN in those cells.

i. For my purposes, the population density dataset did not need any cleaning or merging at this point. The columns were named appropriately so that I could merge the ones I needed with the rest of my features.

6. Combining Dataframes
    a. I created a function to do the following for each year:
        i. I sliced the 'Country' column in each dataset I needed to merge so that I can compare and remove any countries that are not in the corresponding datasets.
        ii. I chained the concat() and drop_duplicates() methods to review the countries that have no match in the World Happiness dataframes. I also used the keys parameter to create a hierarchical index that allowed me to easily inspect the dataframe and identify which index was associated with their respective dataframe.
        iii. After inspection, I created a list that explicitly identifies the locations of all country names in the dataframe that need to be changed to match the names in the World Happiness dataframe. I have a function in dwfunctions.py to update the countries in the dataframe to match the format of the World Happiness dataframe.
        iv. Next, I removed the countries that do not match in both dataframes by slicing the dataframe using the multiindex and the drop() method.
        v. I then merged the dataframes on the Country column for each feature.
        vi. Lastly, I added a year column
    b. Finally once all the data was merged by year, I concatenated all the years into one dataframe
    c. I used the info method to ensure columns had no empty values. If there were, I removed that row.

7. Export

    I exported the cleaned and combined dataframe using Pandas to_csv and setting the index parameter to False, so that when I import the file for further analysis it does not include an extra column.