

WORLD HAPPINESS REPORT

ENGINEERING CLINIC PROJECT REPORT

Submitted by:

Kritya Shree S - 18BCS093

**Department of Computer Science and Engineering
Kumaraguru College of Technology
Chinnavedampatty
Coimbatore**

DECEMBER 2020

Table of Contents

NO	TITLE	PAGE NO
	Abstract	3
1	Introduction	3
	1.1 Conceptual Study of the project	4
	1.2 Purpose	4
	1.3 DataSet	5
	1.4 DataSet Columns	5
	1.5 Tools Used	6
2	Literature Reviews	7
	2.1 World Happiness Report	7
	2.2 Happiness and Life Satisfaction	8
3	Data Preprocessing	9
	3.1 Data Cleaning	9
	3.2 Data Transformation	11
4	Factors that Impact the Happiness Level	12
5	Happiness Factors on Bases of Continent	13
6	Top 10 Happiest Countries Based on Factors	15
7	Top 10 Unhappiest Countries Based on Factors	17
8	Relation Between Happiness Score and Features	18
9	Happiness Score Trend Over The Years	19
	9.1 Score Trend of India	21
10	Conclusion	22
	References	22
	Appendix	23

Abstract

Our project represents the World Happiness Report created by the United Nations Sustainable Development Solutions Network, which contains the data of 138 countries with the articles and rankings of national happiness based off different factors that contribute to the same. We have represented the dataset with different plots and charts and have come to conclusions of which country stands where and under which category when it comes to their happiness levels and scores. We have also generated a methods to determine the top 10 happiest countries and top 10 unhappiest countries by comparing their progress in the years 2015 and 2020.

1. Introduction

Happiness has always been the uncelebrated and eventual goal in every individual's life. Though most people regard money, power and position to be the ultimate holy grail, happiness still remains as the unsung emperor among them. Happiness is what makes sure that you live and not just exist. All things would mean nothing if your life is deprived of happiness and peace.

Majority masses wouldn't really contemplate how important happiness can be when it comes to a country. When we think of a country the immediate thoughts would be many different things but their collective happiness. The importance of it might not be regarded under the limelight most of the time. But the United Nations Sustainable Development Solutions Network think otherwise. They think that the amount of happiness a country possesses might contribute largely to the development of every aspect of it and the different factors that determine a countries growth and development also contribute largely to their overall happiness.

Boiling the facts down, a country's happiness and factors contributing to development go hand in hand as they both affect each other. The world happiness report is all about 138 countries and how happy they are, the factors that affect their happiness score, who tops the tables, who needs to up their game and different categories that affect their happiness levels.

1.1 Conceptual Study of the project

In statistics, exploratory data processing, mostly using visual tools, is an approach to evaluating data sets to summarise their main features. A predictive model may or should not be used, but EDA is mainly meant to see what the data can teach us beyond the systematic task of simulation or hypothesis testing.

Data scientists use exploratory data analysis (EDA) to evaluate and examine data sets and summarise their key features, sometimes through techniques of data visualisation. In order to get the information you need, it helps to decide how best to exploit data sources, making it easy for data scientists to find trends, spot anomalies, test a theory or check conclusions.

EDA is mostly used to see what details can be revealed beyond the formal task of simulation or hypothesis testing which offers a deeper interpretation of variables in the data set and the interactions between them. It will also help decide whether it is acceptable to use the methodological methods you are evaluating for data analysis. Originally developed in the 1970s by the American mathematician John Tukey, EDA techniques continue to be a commonly used tool today in the process of data exploration.

1.2 Purpose

March 20 is celebrated as International World Happiness Day. United Nations has celebrated this day as a way to recognise the importance of happiness in the lives of people around the world.

World Happiness Report is a landmark survey of the state of global happiness that ranks countries by how happy their citizens perceive themselves to be. Analysing this data helps figure out what the citizens of a country feel and can implement changes in accordance to it. One can also try to predict the happiness score for the following year.

- Factors that impact the happiness level
- Happiness factors on bases of continent
- Top and bottom 5 countries based on factors (2015-2020)
- Relation between happiness score and features
- Country Trends over the years

1.3 DataSet

We have used dataset from [kaggle.com](https://www.kaggle.com) and the original data was provided by Gallup World Poll. The rankings are based on the answers to the main issue of life assessment in the survey. This is called the Cantril ladder: it asks respondents to think of a ladder, with the best life imaginable for them being a 10, and the worst life possible being a 0. They are then asked to score their own current lives on a scale of 0 to 10. The rankings are taken from nationally representative surveys for the years 2015-2020. They are entirely based on survey ratings, using Gallup weights to make the results representative. The sub-bars display the approximate degree to which each of the six factors – GDP levels, life expectancy, kindness, social support, democracy and corruption – contributes to making life assessments higher in each country than in Dystopia, a hypothetical country with values equal to the lowest national average in the world for each of the six factors.

2019_report									
country	happiness_score	gdp_per_capita	family	health	freedom	government_trust	generosity	dystopia_residual	continent
Denmark	7.526	1.441780000000000	1.16374	0.79504	0.579410000000000	0.44453	0.36171	2.73939	Europe
Switzerland	7.50900000000000	1.52733	1.14524	0.86303	0.58557	0.41203	0.28083	2.69463	Europe
Iceland	7.501	1.42666	1.18326	0.867330000000000	0.56624	0.14975	0.47678	2.83137	Europe
Norway	7.49800000000000	1.57744	1.1269	0.79579	0.59609	0.35776	0.37895	2.66465	Europe
Finland	7.41300000000000	1.40598	1.13464	0.81091	0.57104	0.41004	0.25492	2.82595	Europe
Canada	7.404	1.44015	1.0961	0.8276	0.5737	0.31329	0.44834	2.70485	North America
Netherlands	7.33900000000000	1.46468	1.02912	0.812310000000000	0.55211	0.29927	0.47416	2.70749	Europe
New Zealand	7.334	1.36066	1.17278	0.83096	0.581470000000000	0.41904	0.494010000000000	2.47553	Australia
Australia	7.313	1.44443	1.10476	0.8512	0.568370000000000	0.32331	0.47407	2.5465	Australia
Sweden	7.291	1.45181	1.08764	0.83121	0.58218	0.40867	0.38254	2.54734	Europe
Israel	7.267	1.33766	0.99537	0.849170000000000	0.36432	0.08728	0.32288	3.31029	Asia
Austria	7.119	1.45038	1.083830000000000	0.80565	0.54355	0.213480000000000	0.32865	2.69343	Europe
United States	7.104	1.50796	1.04782	0.779	0.48163	0.14868	0.41077	2.72782	North America
Costa Rica	7.08700000000000	1.06879	1.02152	0.76146	0.55225	0.10547	0.22553	3.35168000000000	South America
Germany	6.994	1.44787	1.09774	0.81487	0.53466	0.28551	0.30452	2.50931	Europe
Brazil	6.95200000000000	1.08754	1.03938	0.61415	0.40425	0.141660000000000	0.157760000000000	3.50733000000000	South America
Belgium	6.92900000000000	1.42539	1.05249	0.819590000000000	0.51354	0.26248	0.2424	2.61355	Europe
Ireland	6.907	1.48341	1.16157	0.81455	0.54008	0.29754	0.44963	2.15988	Europe
Luxembourg	6.871	1.69752	1.03999	0.845420000000000	0.5487	0.35329	0.27571	2.11055	Europe
Mexico	6.778	1.11508000000000	0.7146	0.71143	0.37709	0.18355	0.11735	3.55905	North America
Singapore	6.739	1.64555	0.867580000000000	0.94719	0.4877	0.46987	0.32706	1.99375	Asia
United Kingdom	6.725	1.40283	1.08672	0.80891	0.50036	0.27399	0.50156	2.14999000000000	Europe
Chile	6.705	1.2167	0.90587	0.818830000000000	0.37789	0.11451	0.31595	2.95505	South America
Panama	6.70100000000000	1.18306	0.98912	0.70835	0.48927	0.08423	0.2418	3.00559	South America
Argentina	6.65	1.15137	1.066120000000000	0.69711	0.42284	0.07296	0.109890000000000	3.12985	South America

1.4 DataSet Columns

Here are the six key categories thought to explain the variations in happiness scores across the 138 nations

Income: Specifically, the log of the per-capita GDP. Income has a virtuous, or vicious, impact on all of the following factors, but is, in itself, a strong indicator of a person's sense of well-being. **GDP of the country** in the year is converted into a score (By taking natural log on GDP per Capita as this form fits perfectly with the data.)

A sense of trust in businesses and government: There is mounting evidence that economic performance and growth are tied to higher levels of trust within the social framework. Confidence of citizens in the actions of a “government to do what is right and perceived fair”; The score is based on the questions: “Is corruption widespread throughout the government or not?”, “And whether the citizens trust and are happy with the government”

The expectation of good health: There is a long-standing pile of research showing a strong relationship between the expectation of a healthy life and a sense of happiness. This is likely tied to better access to health care, better health behaviors, and better nutrition. How positively their inhabitants evaluate their expected future lives; (expected number of remaining years of life spent in good health from a particular age, typically birth or age sixty-five, assuming current rates of mortality and morbidity of a country.)

A personal network of social support: Social support has been shown to have a strong linkage to happiness. Having someone, or a network of someones, to count on in a time of need has a large impact on a person's sense of, and actual, well-being. Answers to the question: Do you have any social support to help you when you are in trouble?

The perception of freedom: When folks are satisfied with their level of freedom to choose what to do with their lives, this personal liberty is a marked by an improvement to the social fabric. Having a sense of freedom to make key life decisions Answers to the question: “Are you satisfied or dissatisfied with your freedom to choose what you do with your life?”

Giving and generosity: Research has found that when people are benevolent and altruistic, characterized by giving money for a charitable purpose (beyond whatever self-interest they might expect in return), they

tend to feel happier. A willingness to give help or support; Generous spending leads to increased well-being. This score is based on: “Have you donated money to a charity in the past month?” on GDP per capita.

Residuals: This is a statistical sponge that soaks up the myriad unknowable things that contribute to a person's sense of well-being, as well as accounts for an inevitable overlap in the impacts of the above factors. It is a benchmark against which we may compare the evaluation of a country's happiness. The lowest scores observed for the six key variables characterize Dystopia.

Along with these are:

Happiness Score: The sum of all the factors in the data.

Country and Continent.

1.5 Tools Used

- Python - Programming Language
- Jupyter Notebook - Notebook
- Numpy - Python Library
- Pandas - Python Library
- Plotly - Python Library
- Seaborn - Python Library
- Matplotlib - Python Library

2. Literature Review

The research papers based on our dataset are World Happiness Report and Happiness and Life Satisfaction.

2.1 World Happiness Report

A seminal study of the state of global happiness, the World Happiness Report rates 156 countries by how happy their people consider themselves to be. For the first time, the World Happiness Survey 2020 rates cities around the world by their subjective well-being and dives further into how our happiness is affected by societal, urban and natural environments. The

World Happiness Report is a publication of the Sustainable Development Solutions Network, powered by data from the Gallup World Poll.

As a basic text for the UN High Level Meeting: Well-being and Satisfaction: Establishing a New Economic Model, attracting international interest, the first World Happiness Study was issued on April 1, 2012. The study outlined the status of world happiness, the sources of happiness and suffering, and case studies highlighted policy consequences. The second World Happiness Survey was published in 2013 and has been published on an annual basis since then, with the exception of 2014. It was to help national governments give more importance to happiness and well-being in determining how to achieve and measure social and economic development.

The distribution of answers to the Cantril ladder question asking respondents to value their lives today on a 0 to 10 scale, with the worst possible life as a 0 and the best possible life as a 10. This gave us a chance to compare happiness levels and inequality in different parts of the world. The typical annual sample is 1,000 people. If a country had surveys in each year, then the sample size would be 3,000 people.

2.2 Happiness and Life Satisfaction

This research paper was first published in 2013 and had substantive revision May 2017. The researchers discuss data and evidence which will answer to the questions: How content are individuals today? In the past, were people happier? How happy are people in different cultures with their lives? And how is this all affected by our living conditions?

Focus will be on survey-based measures of self-reported happiness and life satisfaction.

- Surveys asking people about life satisfaction and happiness do measure subjective well-being with reasonable accuracy.
- Life satisfaction and happiness vary widely both within and among countries. It only takes a glimpse at the data to see that people are distributed along a wide spectrum of happiness levels.

- Most countries that have experienced sustained economic growth have seen increasing happiness levels. So the evidence suggests that income and life satisfaction tend to go together.
- Important life events such as marriage or divorce do affect our happiness, but have surprisingly little long-term impact. The evidence suggests that people tend to adapt to changes.

3. Data Preprocessing

Data Preprocessing is the stage in which the data is converted, or encoded, in every machine learning process to get it to such a state that the machine can now quickly parse it. In other words, the data's characteristics can now be readily understood by the algorithm.

Steps in Data Preprocessing Involved:

1. **Data Cleaning:** There will be multiple irrelevant and incomplete components in the data. Data cleaning is done to tackle this portion. It includes managing missing data, noisy records, etc.
2. **Data Transformation:** This step is taken to turn the data into appropriate forms suitable for the mining process.
3. **Data Reduction:** Data mining is a method used to manage vast volumes of information. In these cases, research becomes more complicated when dealing with immense data volumes. To get rid of this, we use the strategy of data reduction. It seeks to improve the efficiency of storage and lower the cost of data storage and analysis.

3.1 Data Cleaning

Checking for Null Values: It is very much usual to have missing values in your dataset. It may have happened during data collection thus missing values must be taken into consideration. Since we did not have any missing values in our dataset, this step was not necessary.

Inconsistent Values or Outliers: Outliers are a simple concept—they are values that are notably different from other data points, and they can cause problems in statistical procedures.

Eg: In Voters data, having ages under 18 and over 100 are outliers.

Outliers are values that are notably different from other data points, and they can cause problems in statistical procedures. A histogram was used to see if there were any outliers and a box plot was used to check how many outliers are there.

Histograms emphasize the existence of outliers. Look for isolated bars. Our outlier is the bar far to the right/left. Boxplots display asterisks or other symbols on the graph to indicate explicitly when datasets contain outliers. These graphs use the interquartile method with fences to find outliers.

Since we cannot eliminate any rows in our data set, we have replaced the outliers with mean value.

```
In [5]: df_2015.isnull().sum()
```

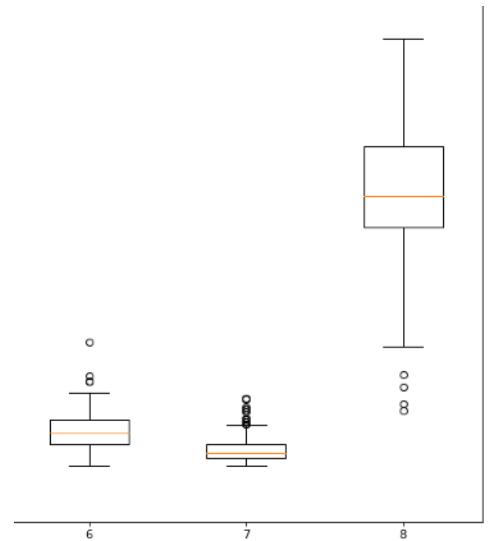
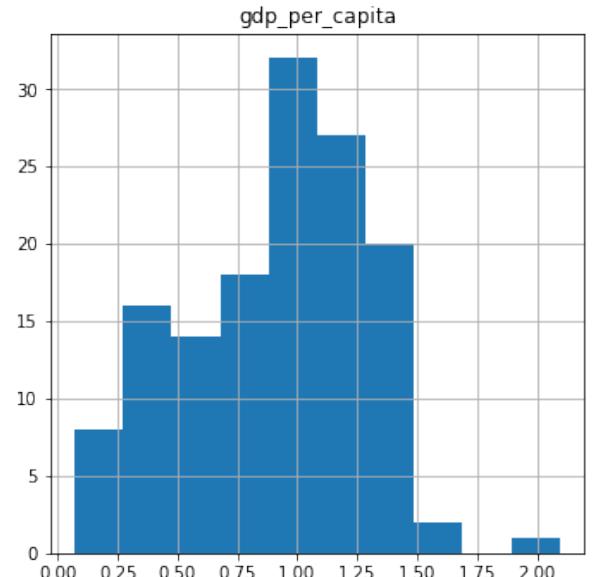
```
Out[5]: country      0  
happiness_score    0  
gdp_per_capita     0  
family              0  
health              0  
freedom             0  
generosity          0  
government_trust   0  
dystopia_residual  0  
continent           0  
dtype: int64
```

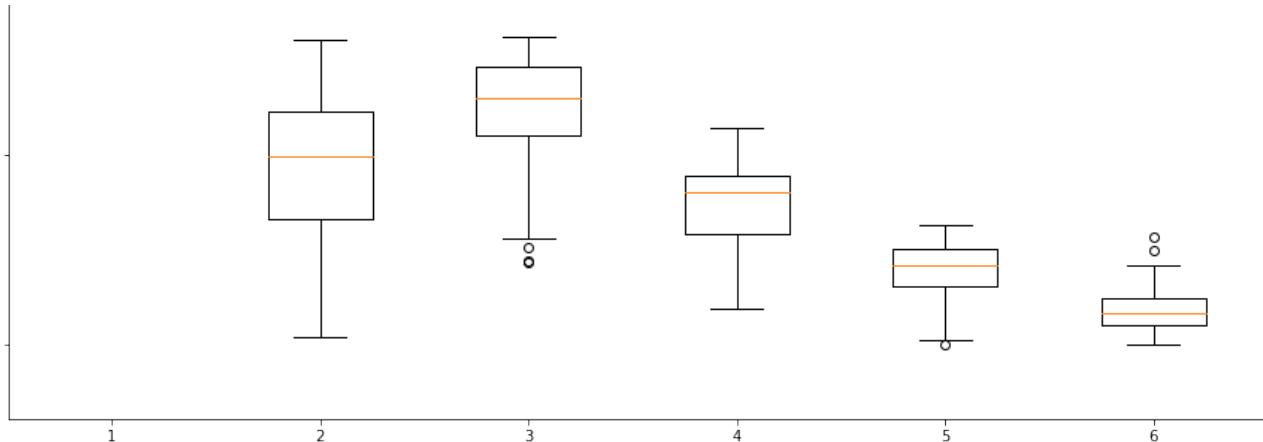
```
In [6]: df_2016.isnull().sum()
```

```
Out[6]: country      0  
happiness_score    0  
gdp_per_capita     0  
family              0  
health              0  
freedom             0  
generosity          0  
government_trust   0  
dystopia_residual  0  
continent           0  
dtype: int64
```

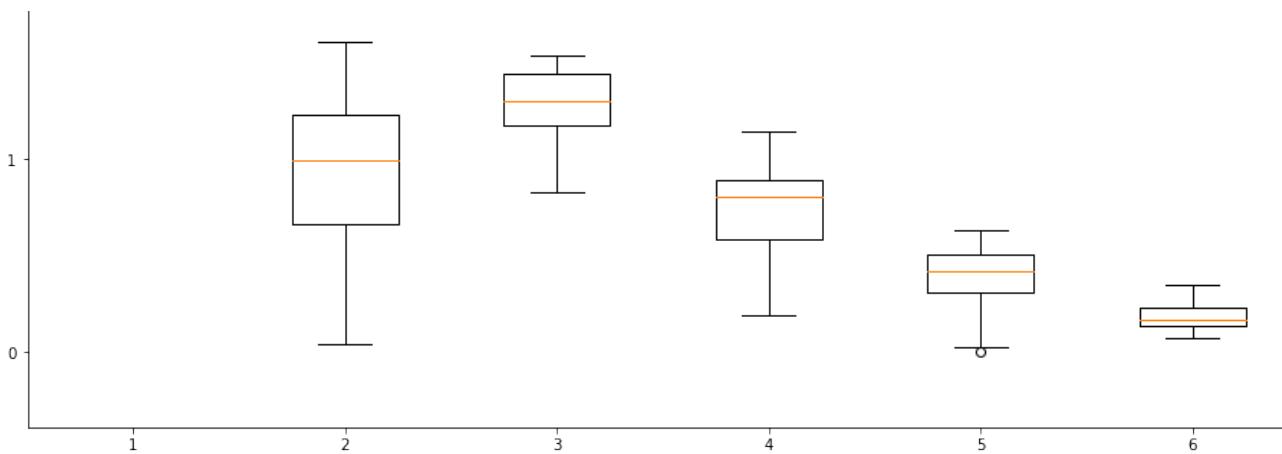
```
In [7]: df_2017.isnull().sum()
```

```
Out[7]: country      0  
happiness_score    0  
gdp_per_capita     0  
family              0  
health              0  
freedom             0  
generosity          0  
government_trust   0  
dystopia_residual  0  
continent           0  
dtype: int64
```





Before Removing Outliers



After Removing Outliers

3.2 Data Transformation

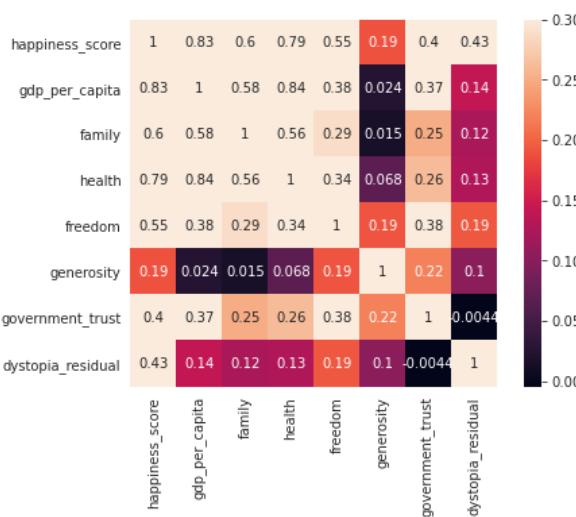
Normalization is a technique often applied as part of data preparation for machine learning. Its goal is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. For machine learning, every dataset does not require normalization. Since our dataset value doesn't require normalization, we skipped this step.

4. Factors that Impact the Happiness Level

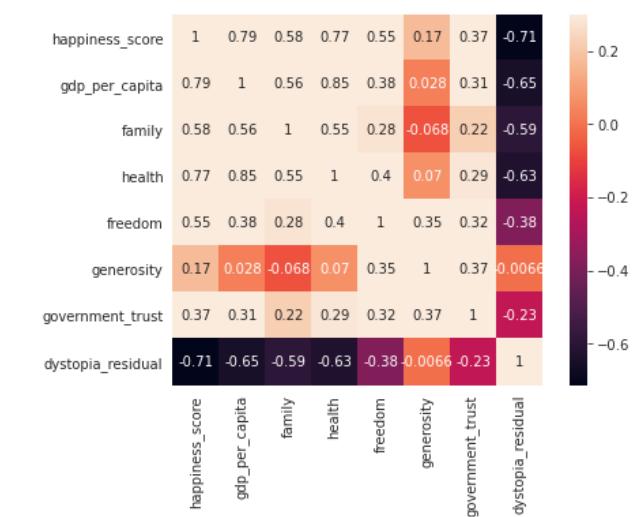
A heatmap is a graphical data representation that uses a color-coding scheme to represent various values. Heatmaps are most often used to explain the actions or association of users between variables.

Inference:

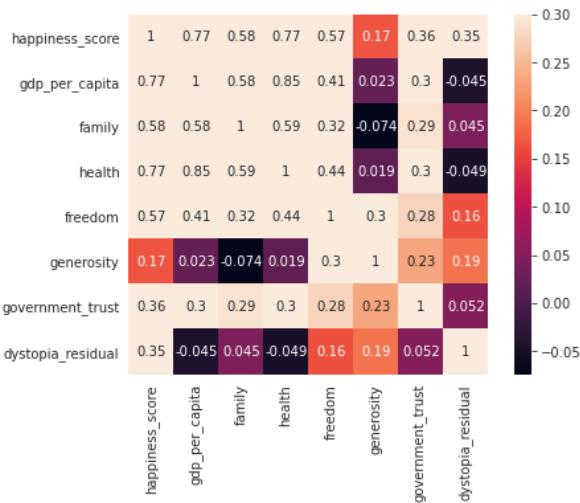
In the year 2015 the main factor seem to be GDP, Family, Health and Freedom but, After the year 2017, Govt Trust and generosity have played a role in a Country's Happiness. In the year 2020, the correlations between generosity, govt trust and other factors have significantly decreased.



2015



2017



2020

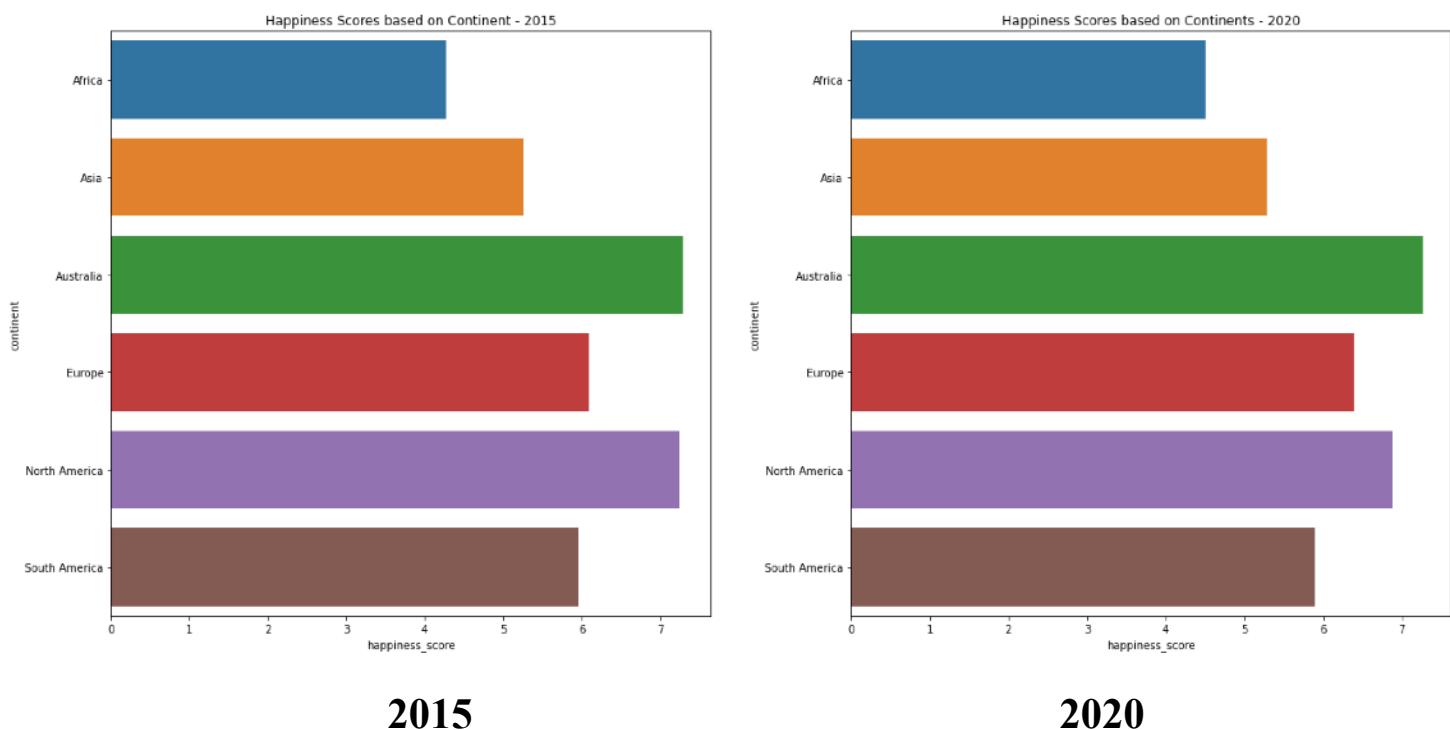
5. Happiness Factors on the Bases of Continent

A bar graph is used to denote this analysis. Bar graph is often used to compare between values of different categories in the data.

Happiness Score:

Inference:

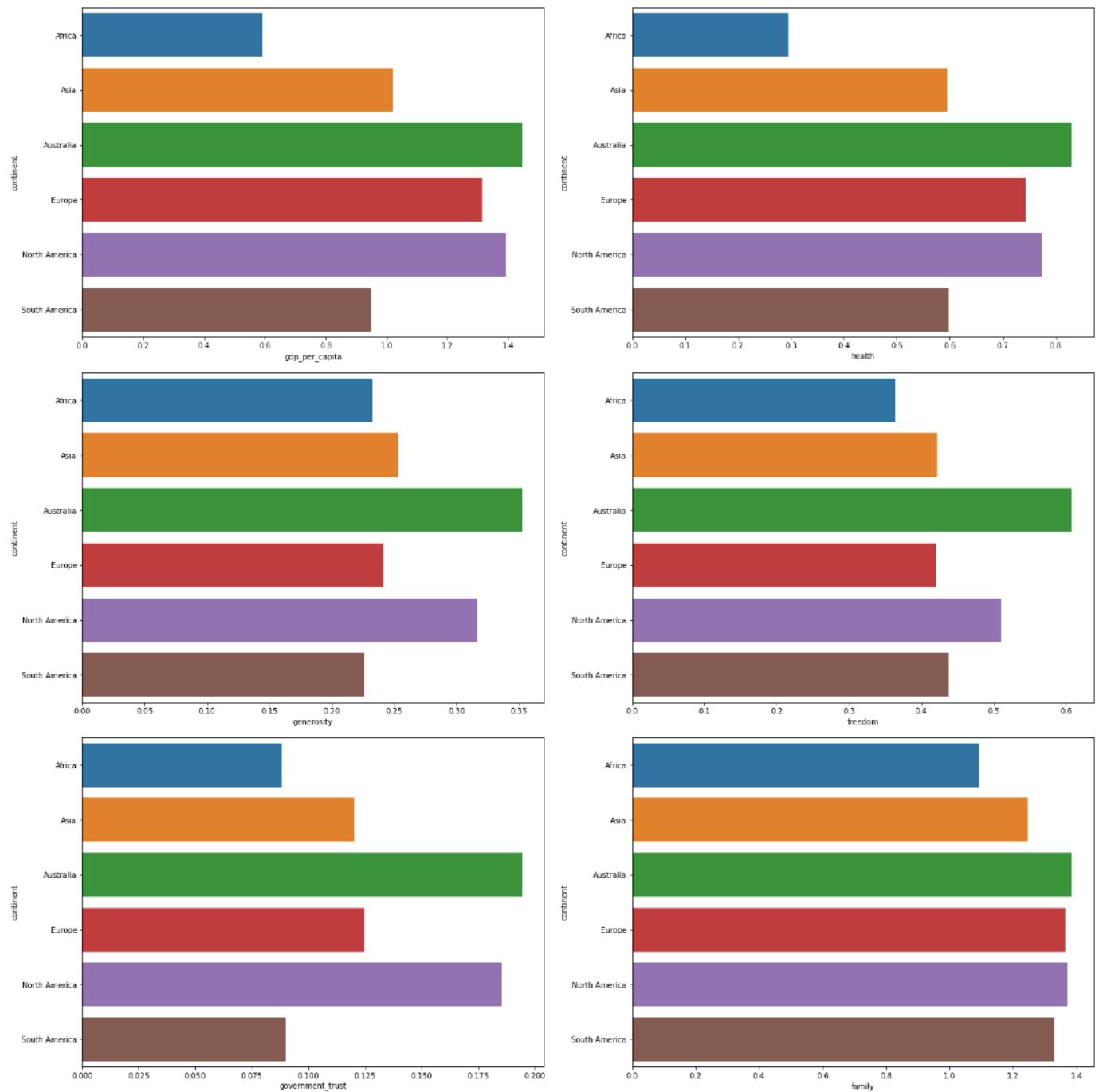
African countries have lowest happiness score, followed by Asia. Australia and North America have the highest followed by Europe and South America. The score of North America has been reduced over the years but scores of Asia and Europe have significantly increased.



All Factors:

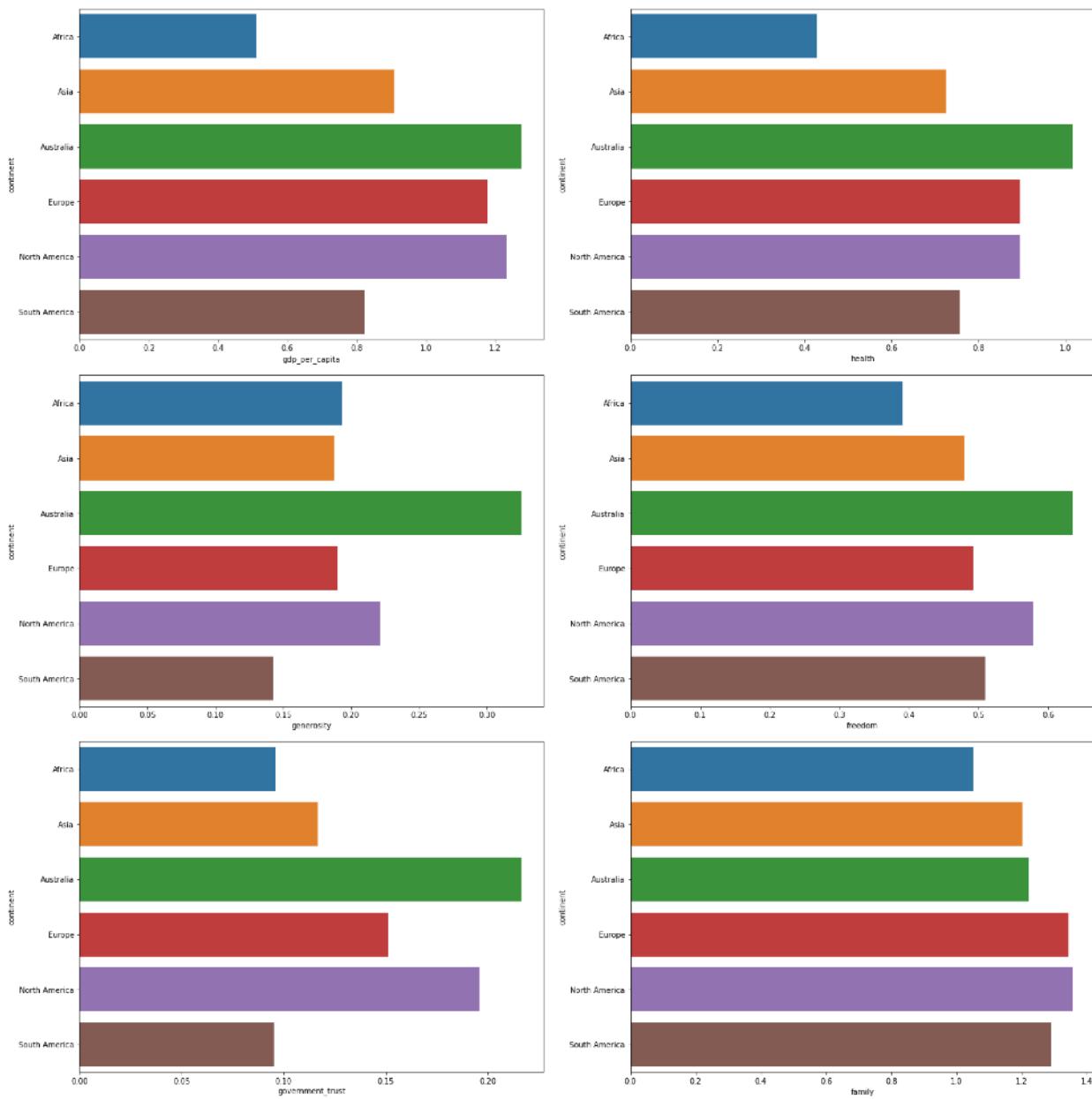
2015 Inference:

Australia gives importance to all factors equally. Africa has the lowest score in all factors. Asia and South America give comparatively more importance to family/social support than the other factors. Europe gives more importance to GDP, Health and Family.



2020 Inference:

There is a drastic decrease in GDP and generosity but an increase in health, freedom and govt trust in all continents. Family value of Australia has decreased over the years. Asia has a decrease in family/social support value. South America has an increase in health and freedom but the Europe gives more importance to GDP, Health and Family.



6. Top 10 Happiest Countries Based on Factors (2015-2020)

Inference:

GDP: Singapore and Luxembourg have retained their position throughout the years and Ireland and Netherlands have increased in position.

Family: UK, Switzerland, Slovenia and Mongolia's position have increased whereas Spain being 1st in 2015 is now not on the list.

Health: Singapore and Japan have retained their positions. Switzerland, Spain and Australia have an increase but South Korea and Italy have a drastic decrease in position.

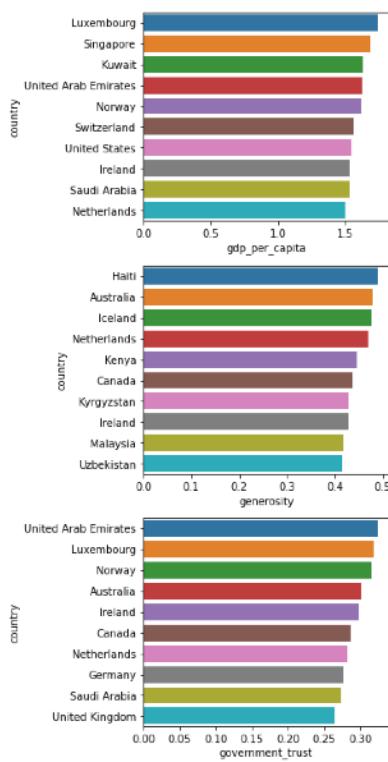
Generosity: Positions of Iceland and Australia have increased but the rest of the positions are different in 2020.

Govt Trust: UK, Netherlands and Ireland have an increase in position.

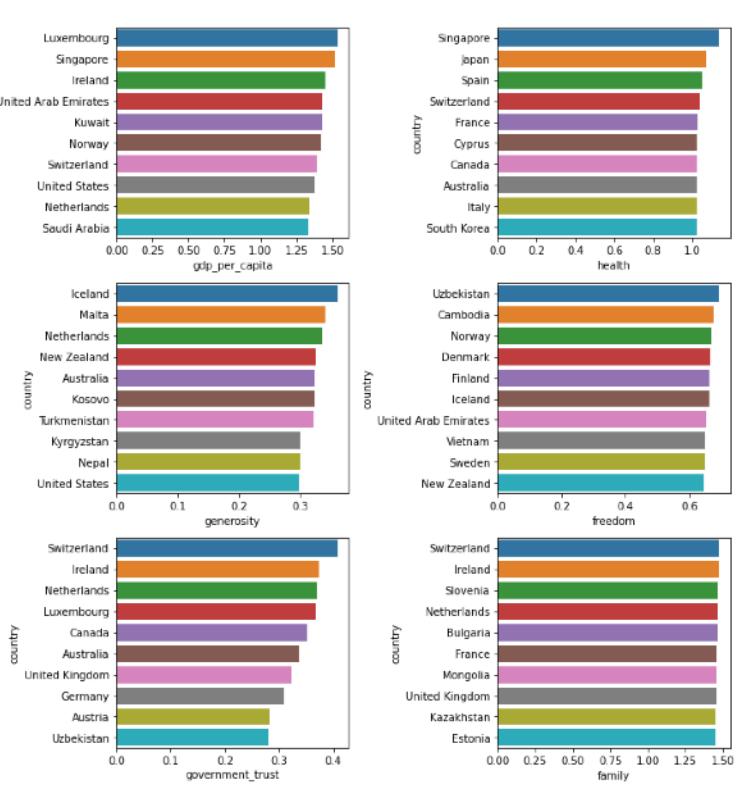
Freedom: Uzbekistan has retained its position as 1 and the top 5 countries are almost the same both years.

Happiness Score: The first 6 countries are the same both years with a few of their positions shuffled. New Zealand has also retained its 7th position in the top 10 and Sweden has increased in 2 positions. Canada and Australia have lost their top 10 positions in 2020.

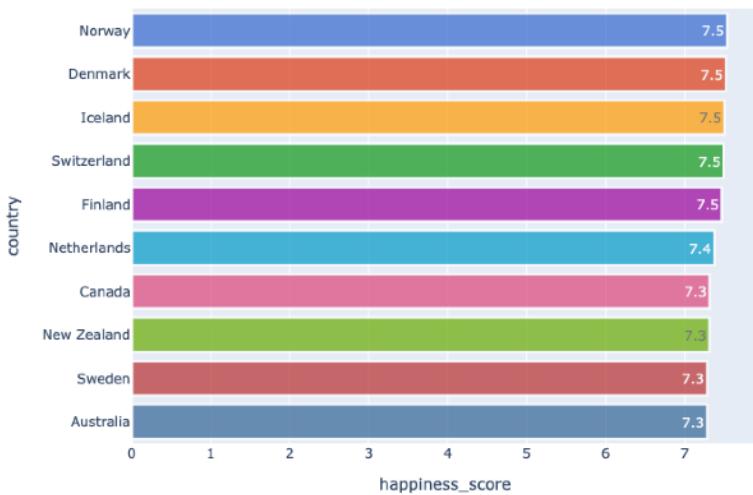
2015



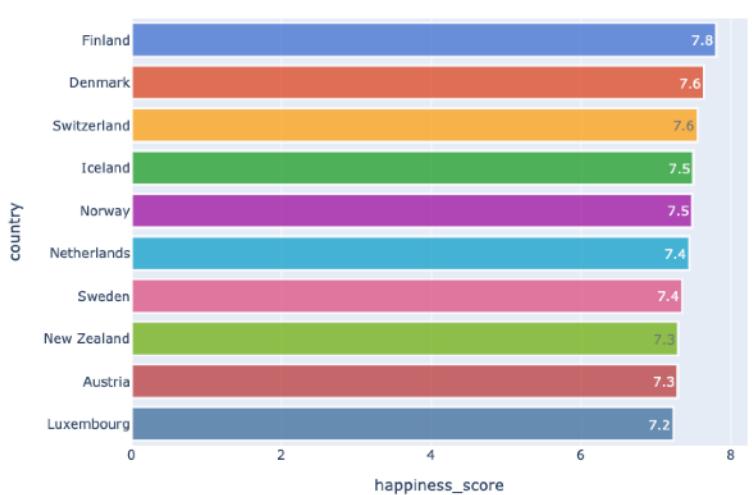
2020



Top 10 Happiest Countries (2015)



Top 10 Happiest Countries (2020)



7. Top 10 Unhappiest Countries Based on Factors (2015-2020)

Inference:

GDP: The first 5 positions are the same countries with their positions swapped and Madagascar. The rest in 2015 have improved by 2020.

Family: The first positions is retained by Congo. Ivory Coast and Madagascar have risen from being in 6th and 7th to 2nd and 3rd positions.

Health: Except South Africa, the same countries as in 2015 is retained in 2020 with changes in positions.

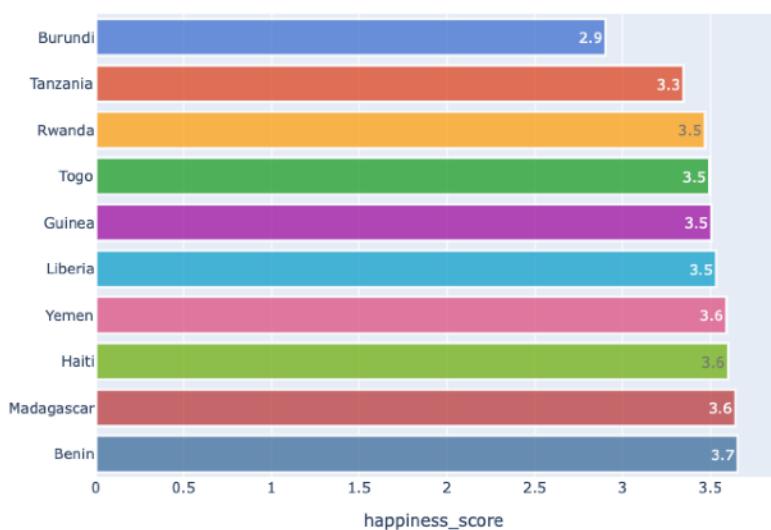
Generosity: Argentina has risen from position 7 to 2 in 2020. The rest of the countries are very different in both years.

Govt Trust: Thailand and Kyrgyzstan have increased in positions in 2020 whereas Mongolia and Russia have come from 2nd and 3rd position to 9th and 10th position.

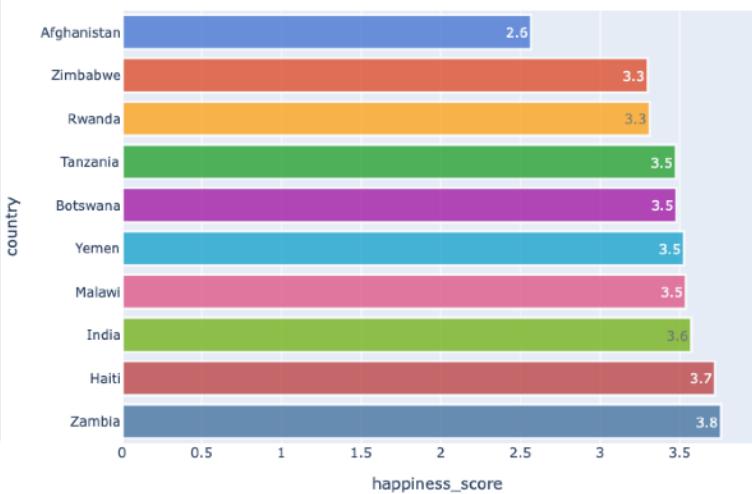
Freedom: Afganistan has 0 freedom in the year 2020 and only very few different countries in the 2020 list.

Happiness Score: Burundi has improved over the years and is not in the top 10 unhappiest counties in 2020. Tanzania has jumped from 2nd to 4th position in 2020. Yemen and Rwanda have maintained their positions. India has joined the list in the year 2020.

Top 10 Unhappiest Countries (2015)

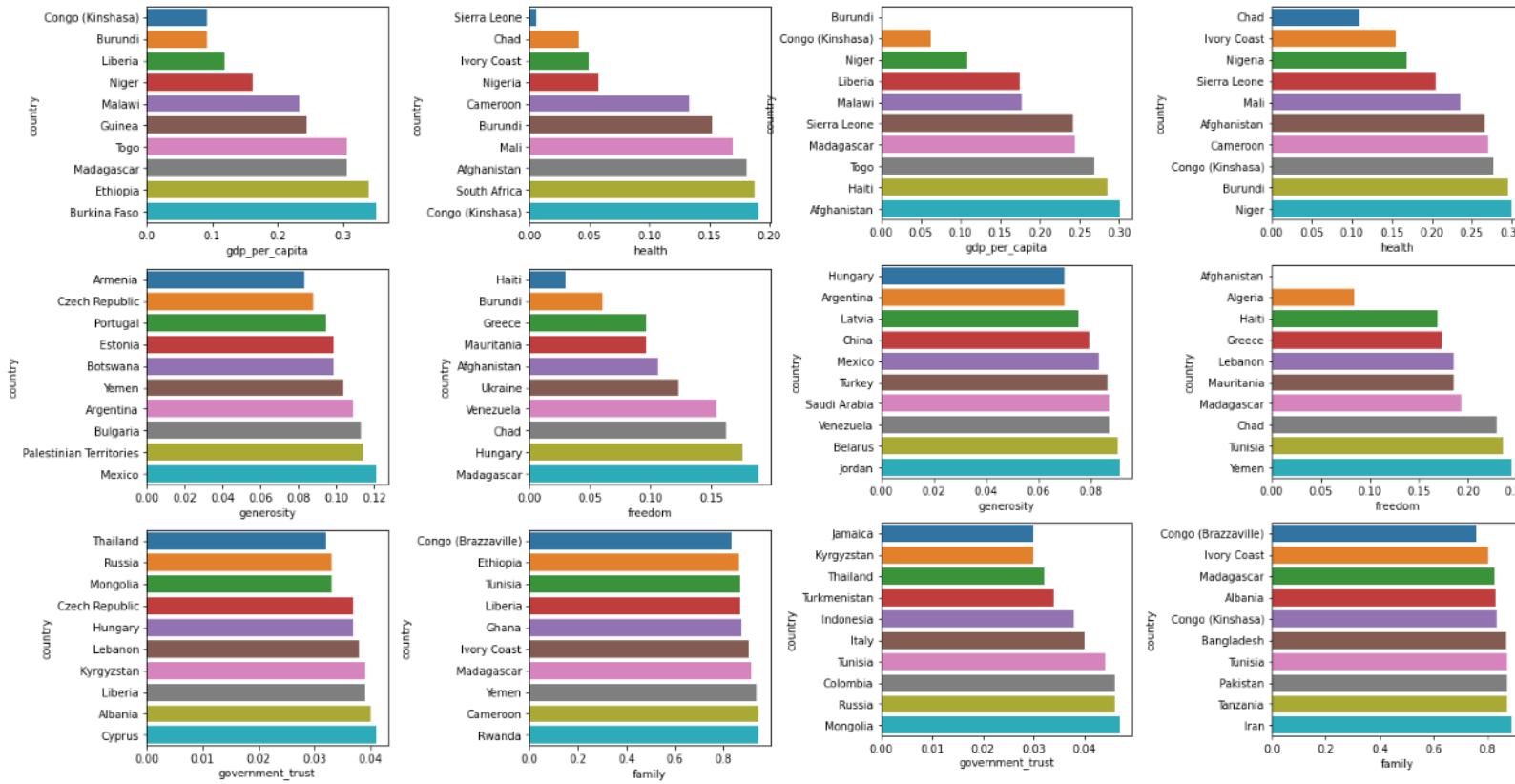


Top 10 Unhappiest Countries (2020)



2015

2020



8. Relation Between Happiness Score and Features

This can be used to predict what a country must focus on to improve their happiness score furthermore.

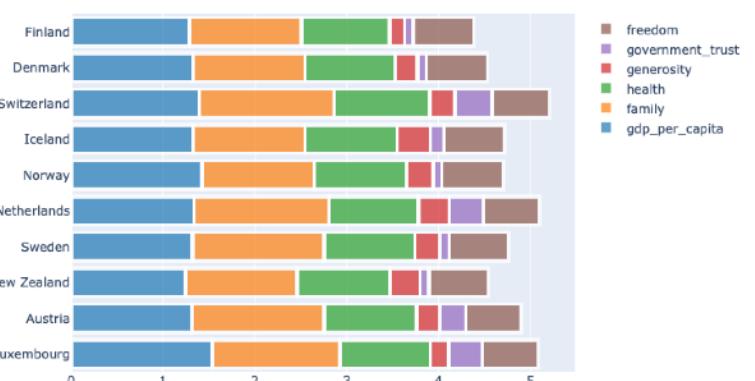
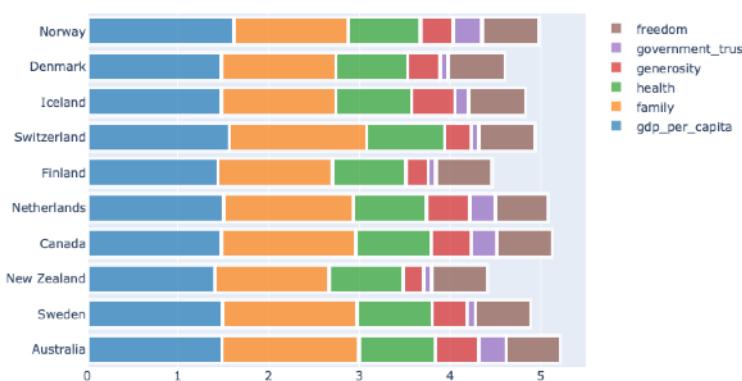
Inference (Top 10 Happiest Countries):

As mentioned in the correlation heatmap; GDP, family, health and freedom are the main factors contributing in the top 10 countries.

Generosity and Govt Trust can be improved in all countries in both years. Health has gained more importance in the year 2020 compared to 2015 but family has decreased in 2020

Contribution to Happiness Score for Top 10 Happiest Countries (2015)

Contribution to Happiness Score for Top 10 Happiest Countries (2020)



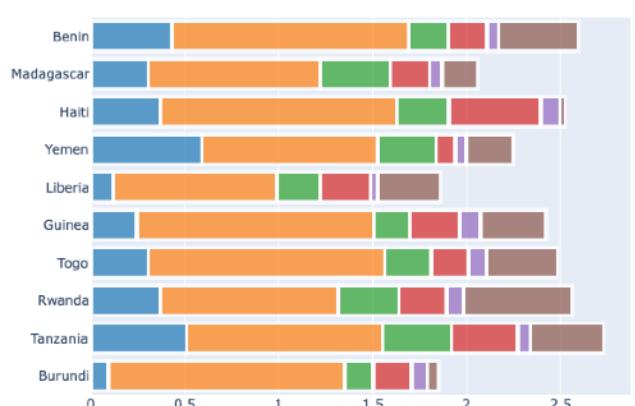
Inference (Top 10 Unhappiest Countries):

This can be used to predict what a country must focus on to improve their happiness score furthermore.

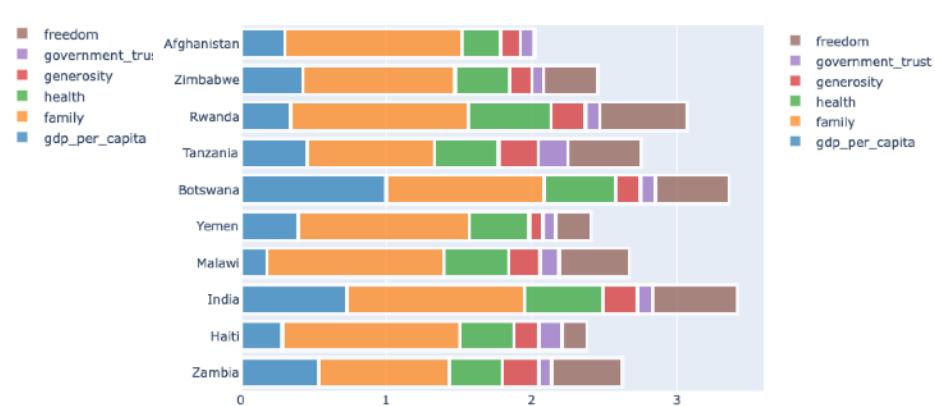
Inference

More importance is given to family and freedom compared to the happiest countries. As mentioned health has improved in 2020. The overall scale has improved from 3 being max in 2015 to 3.5 in 2020.

Contribution to Happiness Score for Top 10 Unhappiest Countries (2015)



Contribution to Happiness Score for Top 10 Unhappiest Countries (2020)



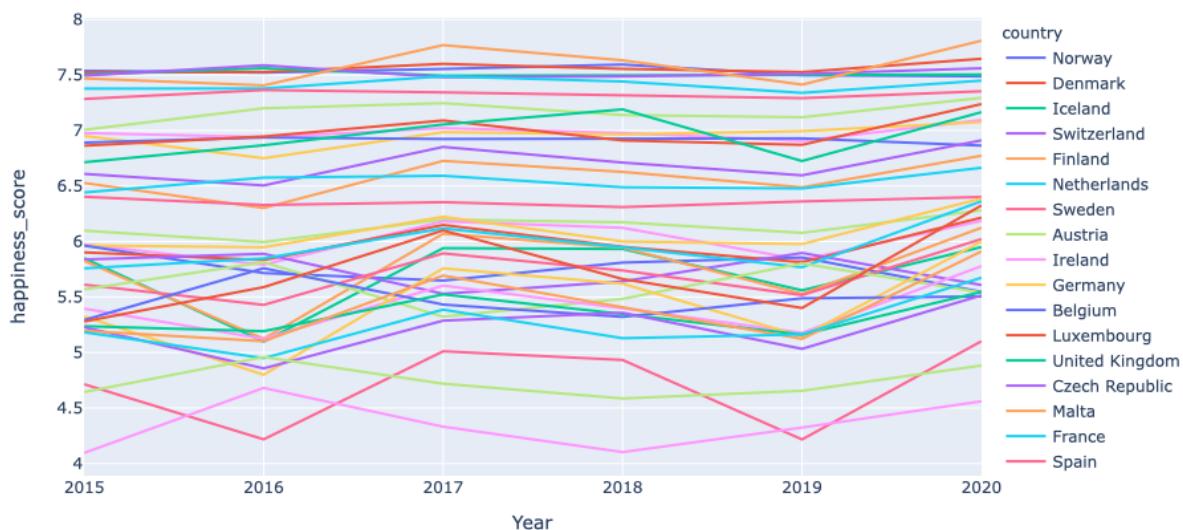
9. Happiness Score Trend Over the Years

With this information we can see how the countries happiness score is throughout the years and also see if it's likely to go up or down the following year.

Happiness Score Trend in Asia



Happiness Score Trend in Europe



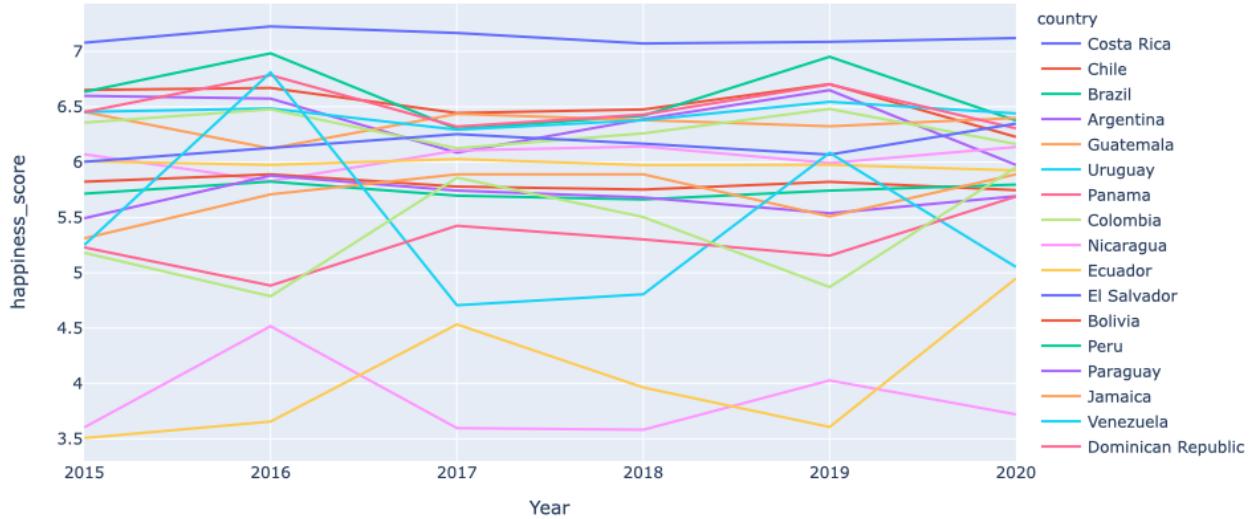
Happiness Score Trend in Africa



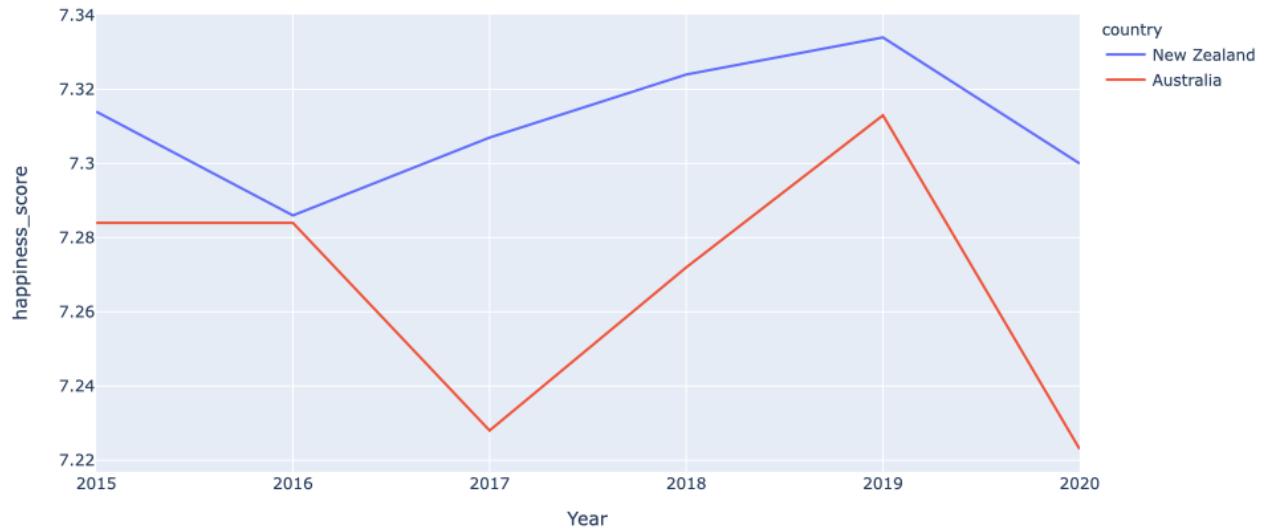
Happiness Score Trend in North America



Happiness Score Trend in South America



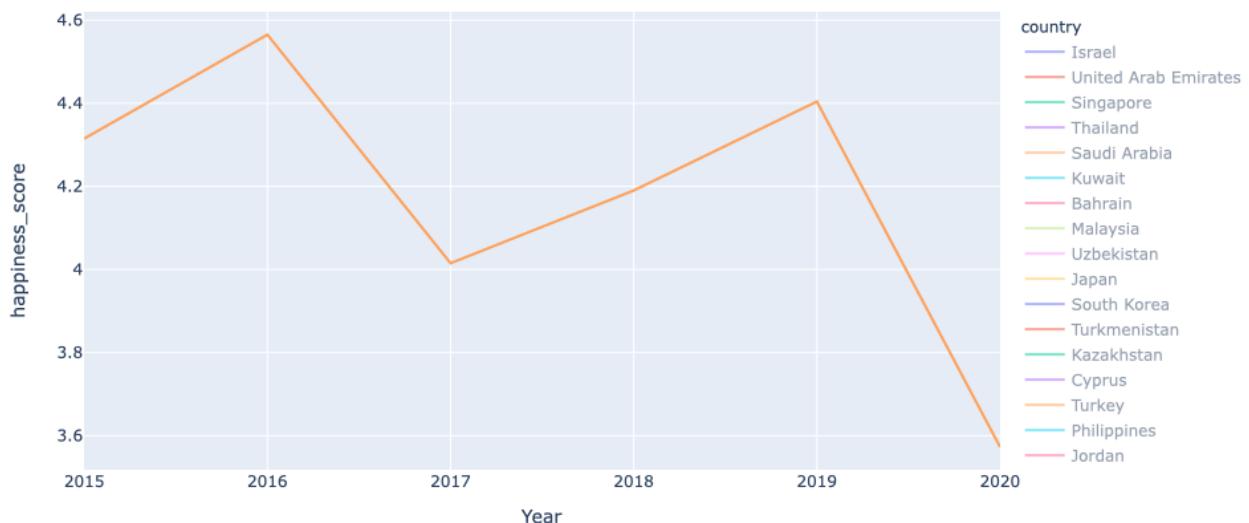
Happiness Score Trend in Australia



9.1 Score Trend of India

The happiness scores throughout the last 6 years were non-linear. It was in its peak in the year 2016 with 4.56 and next peak in 2019 with 4.40. The lowest score is in the year 2020 with 3.57 as the score and has also secured 8th position in “Top 10 unhappiest countries in 2020”.

Happiness Score Trend in Asia



10. Conclusion:

For several of the main variables that have been shown to foster well-being, including jobs, good life expectancy, social support, equality, confidence, and hospitality, the top countries tend to have high values to such an extent that shifts in the top rankings from year to year are to be predicted. In addition to raising the degree of well-being, favourable social conditions often facilitate its spread. We assume that social conditions or the quality of life are of first-order significance.

If the priorities of government really are citizen well-being and satisfaction, so the first step in beginning an evidence-based path towards achieving that aim is to take serious study on systemic and cultural determinants of citizen happiness.

References:

- [1] <https://www.kaggle.com/yamaerenay/world-happiness-report-preprocessed>
- [2] <https://worldhappiness.report/faq/>
- [3] <https://www.arcgis.com/apps/Cascade/index.html?appid=5a333512e79c4c5ab9052c9d0ff8f55b>
- [4] <https://ourworldindata.org/happiness-and-life-satisfaction>

[5] <https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/>

[6] <https://statisticsbyjim.com/basics/outliers/>

[7] <https://www.ibm.com/cloud/learn/exploratory-data-analysis>

Appendix:

Executed Code:

```
Preprocessing Statements and Reading Data

In [8]: import pandas as pd
import numpy as np
import os
from collections import defaultdict

import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
%matplotlib inline

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error

In [9]: df_2015 = pd.read_csv("/Users/kritya/Desktop/DataSet/WHR Data Set/2015_report.csv")
df_2016 = pd.read_csv("/Users/kritya/Desktop/DataSet/WHR Data Set/2016_report.csv")
df_2017 = pd.read_csv("/Users/kritya/Desktop/DataSet/WHR Data Set/2017_report.csv")
df_2018 = pd.read_csv("/Users/kritya/Desktop/DataSet/WHR Data Set/2018_report.csv")
df_2019 = pd.read_csv("/Users/kritya/Desktop/DataSet/WHR Data Set/2019_report.csv")
df_2020 = pd.read_csv("/Users/kritya/Desktop/DataSet/WHR Data Set/2020_report.csv")

Preprocessing

Checking Null Values

In [10]: df_2015.isnull().sum()

Out[10]: country      0
happiness_score      0
gdp_per_capita       0
family                0
health                0
freedom               0
generosity            0
government_trust     0
dystopia_residual    0
continent             0
dtype: int64

In [11]: df_2016.isnull().sum()

Out[11]: country      0
happiness_score      0
gdp_per_capita       0
family                0
health                0
freedom               0
generosity            0
government_trust     0
```

```
In [12]: df_2017.isnull().sum()
```

```
Out[12]: country      0  
happiness_score    0  
gdp_per_capita     0  
family             0  
health             0  
freedom            0  
generosity         0  
government_trust   0  
dystopia_residual  0  
continent          0  
dtype: int64
```

```
In [13]: df_2018.isnull().sum()
```

```
Out[13]: country      0  
happiness_score    0  
gdp_per_capita     0  
family             0  
health             0  
freedom            0  
generosity         0  
government_trust   0  
dystopia_residual  0  
continent          0  
dtype: int64
```

```
In [14]: df_2019.isnull().sum()
```

```
Out[14]: country      0  
happiness_score    0  
gdp_per_capita     0  
family             0  
health             0  
freedom            0  
generosity         0  
government_trust   0  
dystopia_residual  0  
continent          0  
dtype: int64
```

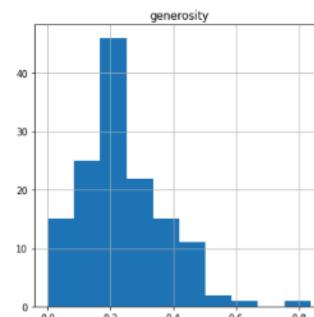
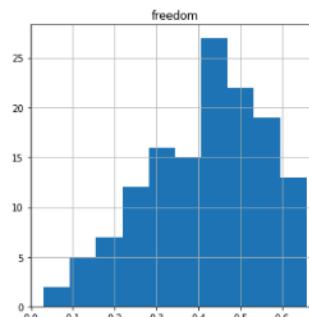
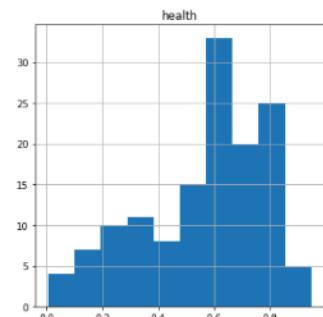
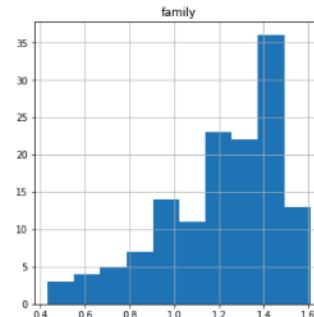
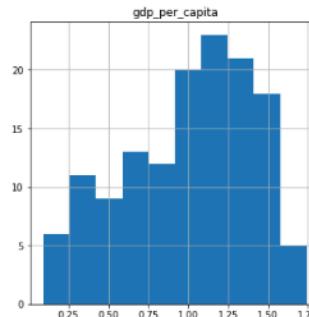
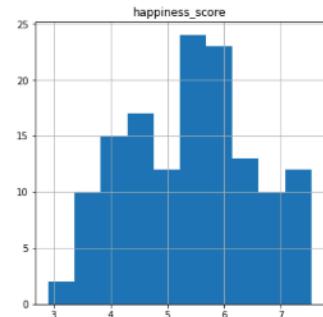
```
In [15]: df_2020.isnull().sum()
```

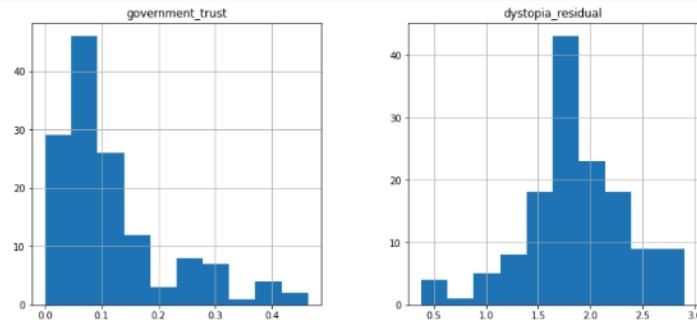
```
Out[15]: country      0  
happiness_score    0  
gdp_per_capita     0  
family             0  
health             0  
freedom            0  
generosity         0  
government_trust   0  
dystopia_residual  0  
continent          0  
dtype: int64
```

Histogram

```
In [16]: plt.rcParams['figure.figsize'] = (20,20)  
df_2015.hist()
```

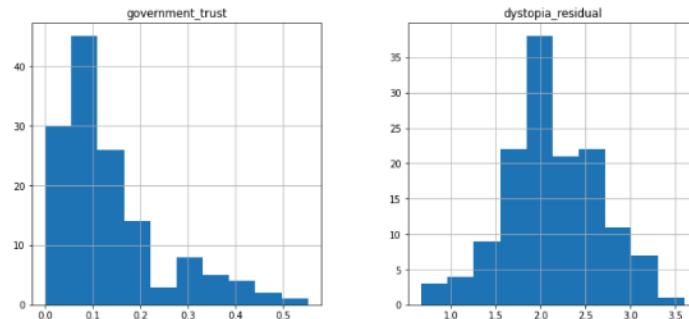
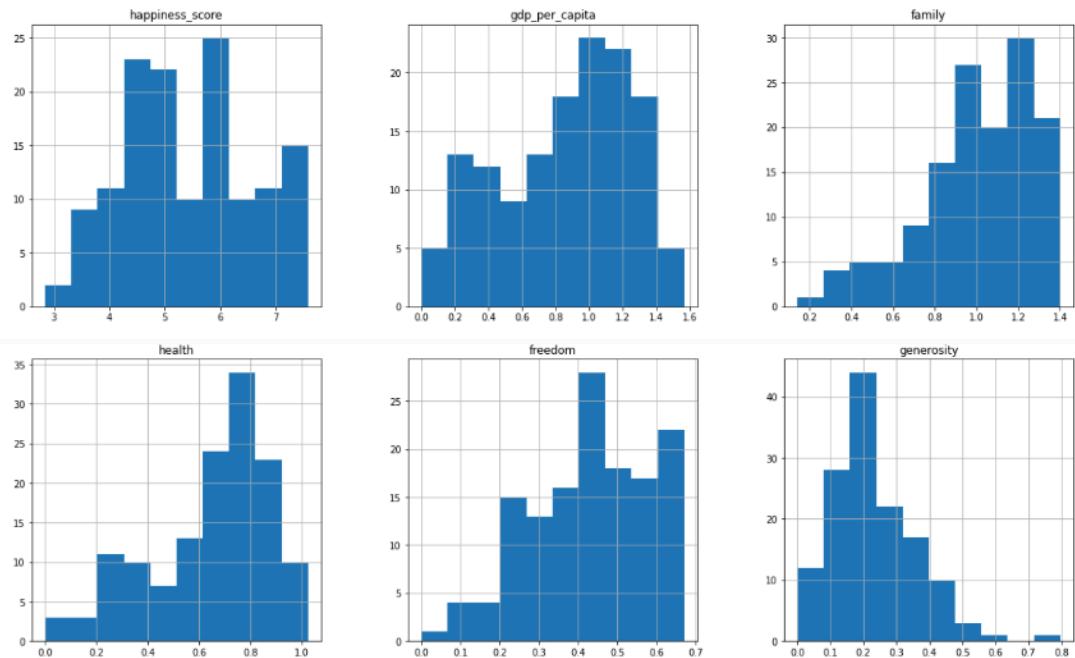
```
Out[16]: array([{'center': 'happiness_score', 'gdp_per_capita': 60, 'family': 14, 'health': 0.2, 'freedom': 0.1, 'generosity': 0.1, 'government_trust': 0.1, 'dystopia_residual': 0.1}, {'center': 'gdp_per_capita', 'gdp_per_capita': 60, 'family': 14, 'health': 0.2, 'freedom': 0.1, 'generosity': 0.1, 'government_trust': 0.1, 'dystopia_residual': 0.1}, {'center': 'family', 'gdp_per_capita': 14, 'family': 60, 'health': 0.2, 'freedom': 0.1, 'generosity': 0.1, 'government_trust': 0.1, 'dystopia_residual': 0.1}, {'center': 'health', 'gdp_per_capita': 14, 'family': 60, 'health': 60, 'freedom': 0.1, 'generosity': 0.1, 'government_trust': 0.1, 'dystopia_residual': 0.1}, {'center': 'freedom', 'gdp_per_capita': 14, 'family': 60, 'health': 0.2, 'freedom': 60, 'generosity': 0.1, 'government_trust': 0.1, 'dystopia_residual': 0.1}, {'center': 'generosity', 'gdp_per_capita': 14, 'family': 60, 'health': 0.2, 'freedom': 0.1, 'generosity': 60, 'government_trust': 0.1, 'dystopia_residual': 0.1}, {'center': 'government_trust', 'gdp_per_capita': 14, 'family': 60, 'health': 0.2, 'freedom': 0.1, 'generosity': 0.1, 'government_trust': 60, 'dystopia_residual': 0.1}, {'center': 'dystopia_residual', 'gdp_per_capita': 14, 'family': 60, 'health': 0.2, 'freedom': 0.1, 'generosity': 0.1, 'government_trust': 0.1, 'dystopia_residual': 60}], dtype=object)
```





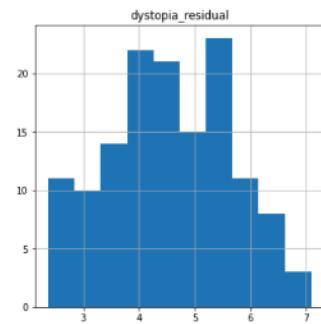
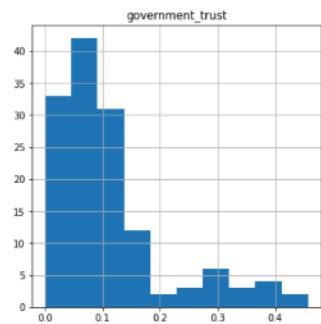
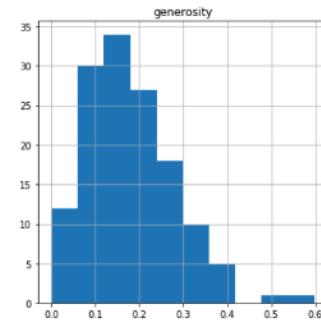
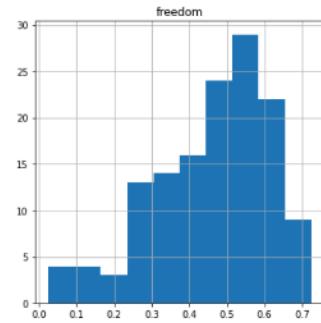
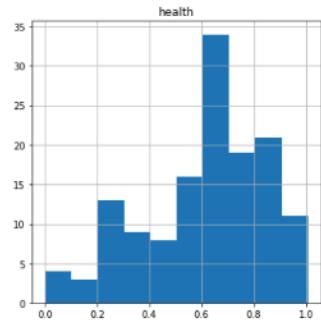
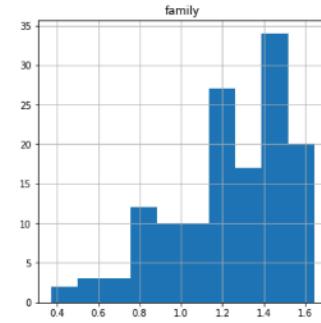
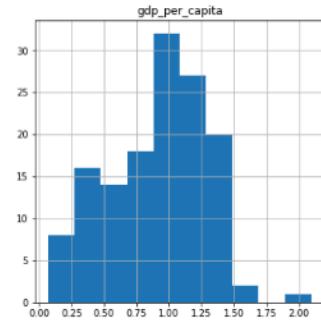
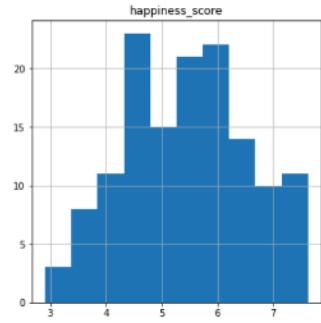
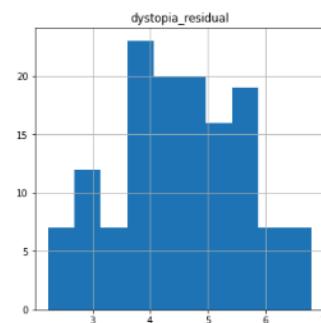
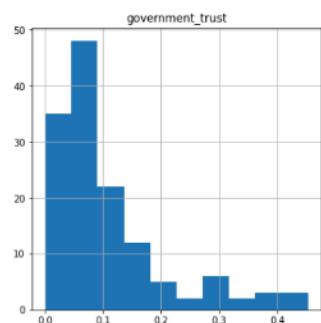
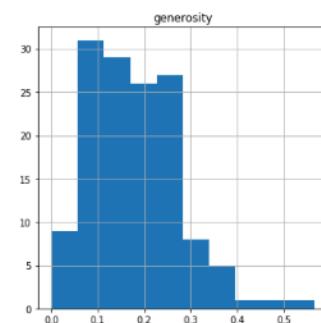
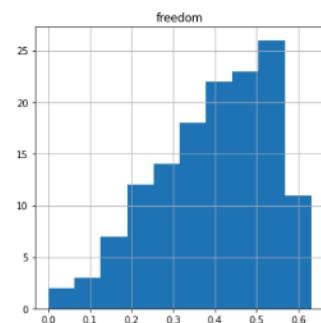
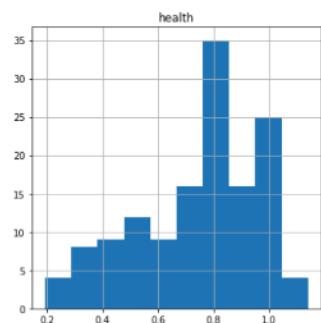
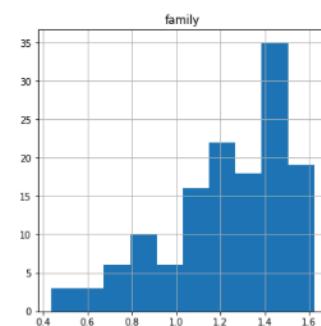
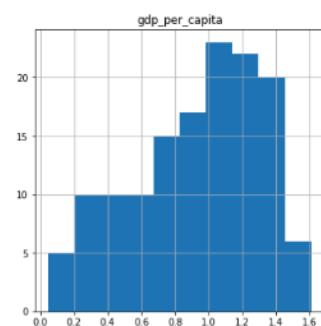
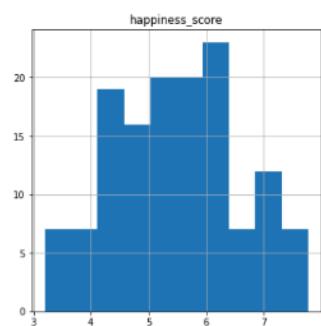
```
In [17]: plt.rcParams['figure.figsize'] = (20,20)
df_2016.hist()
```

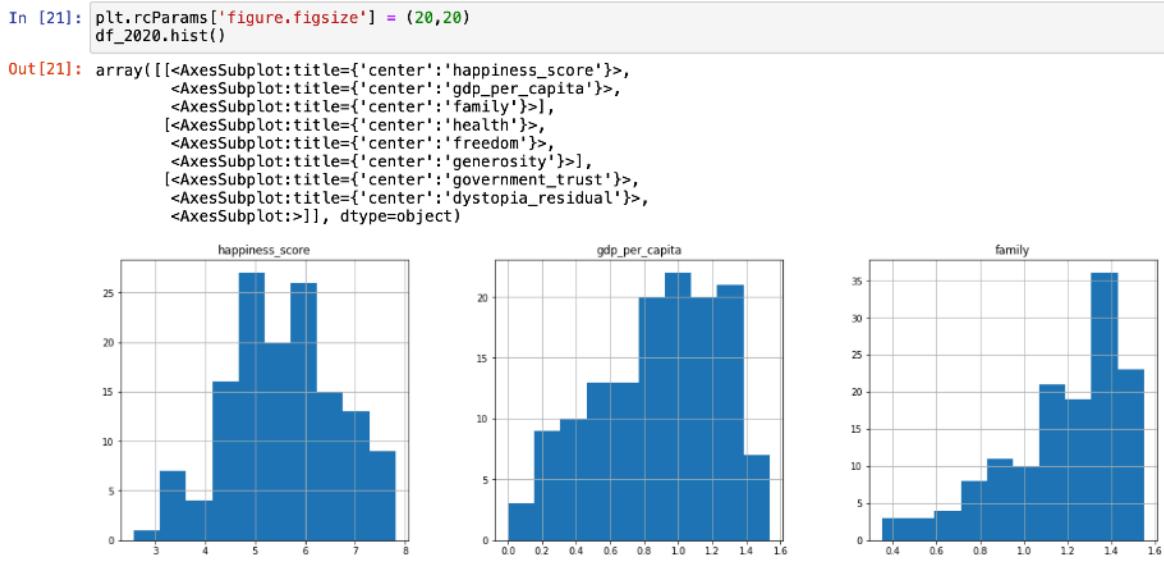
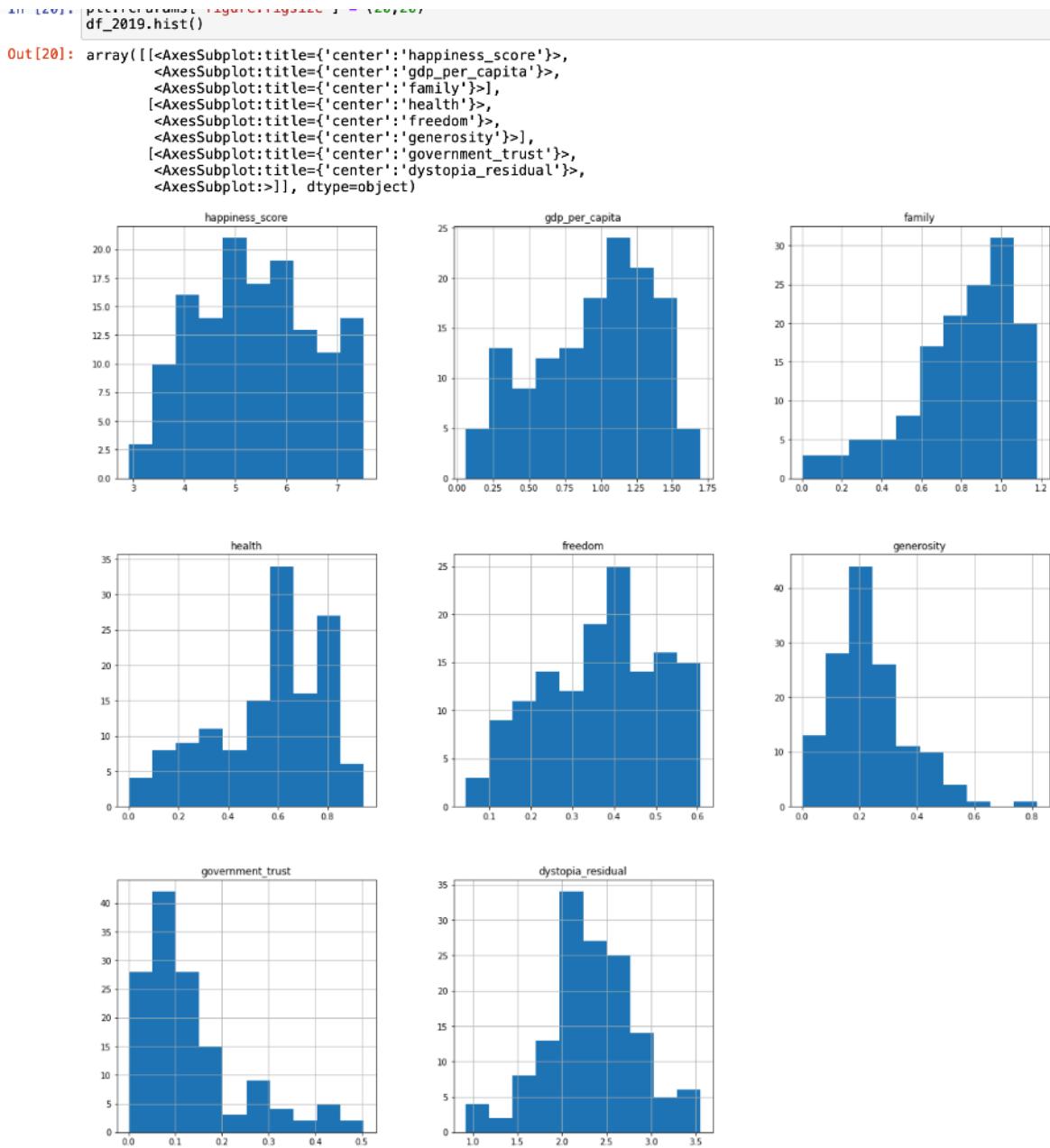
```
Out[17]: array([ [<AxesSubplot:title={'center':'happiness_score'}>,
   <AxesSubplot:title={'center':'gdp_per_capita'}>,
   <AxesSubplot:title={'center':'family'}>],
  [<AxesSubplot:title={'center':'health'}>,
   <AxesSubplot:title={'center':'freedom'}>,
   <AxesSubplot:title={'center':'generosity'}>],
  [<AxesSubplot:title={'center':'government_trust'}>,
   <AxesSubplot:title={'center':'dystopia_residual'}>,
   <AxesSubplot:>]], dtype=object)
```

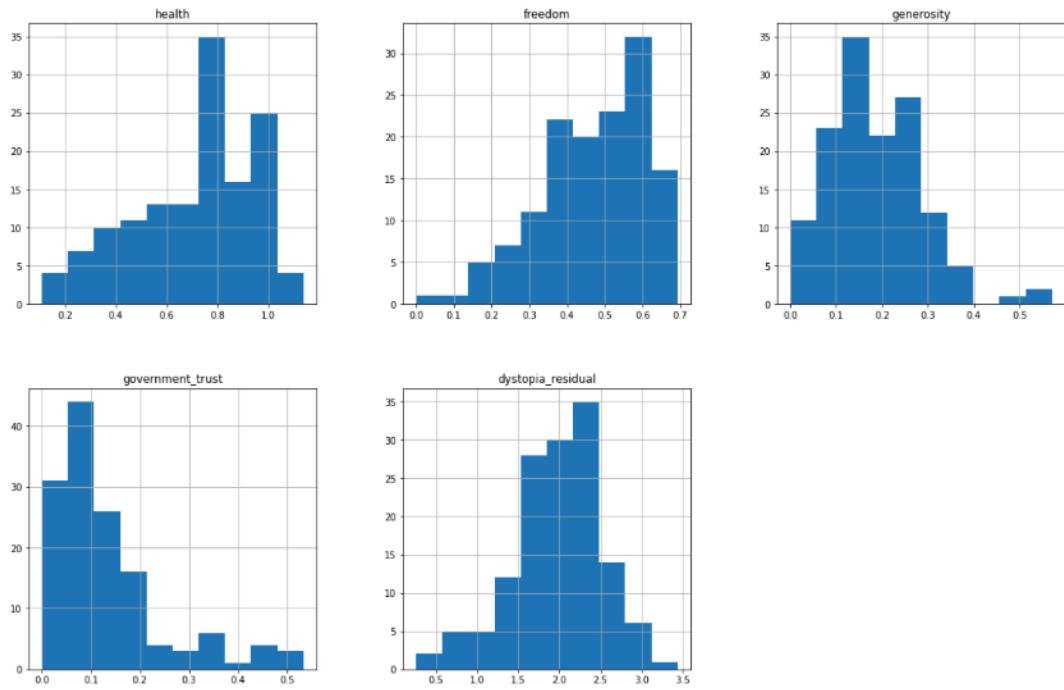


```
In [18]: plt.rcParams['figure.figsize'] = (20,20)
df_2017.hist()
```

```
Out[18]: array([ [<AxesSubplot:title={'center':'happiness_score'}>,
   <AxesSubplot:title={'center':'gdp_per_capita'}>,
   <AxesSubplot:title={'center':'family'}>],
  [<AxesSubplot:title={'center':'health'}>,
   <AxesSubplot:title={'center':'freedom'}>,
   <AxesSubplot:title={'center':'generosity'}>],
  [<AxesSubplot:title={'center':'government_trust'}>,
   <AxesSubplot:title={'center':'dystopia_residual'}>,
   <AxesSubplot:>]], dtype=object)
```

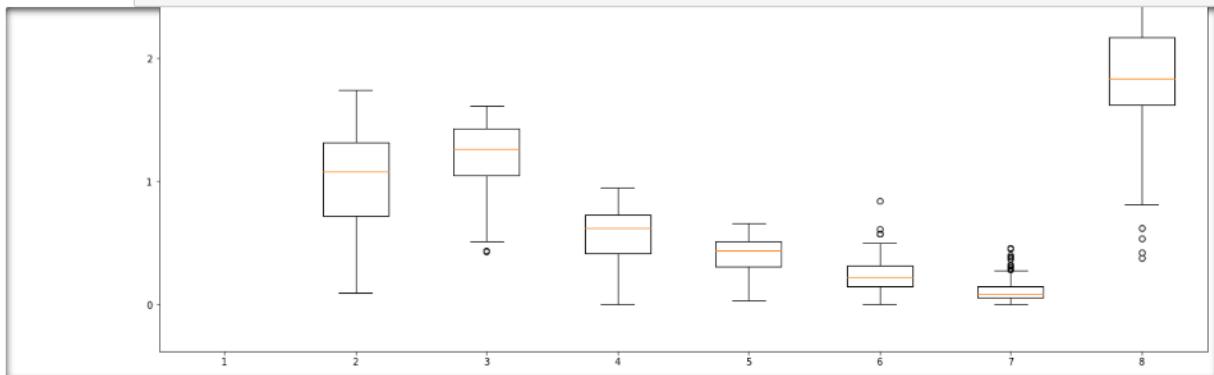




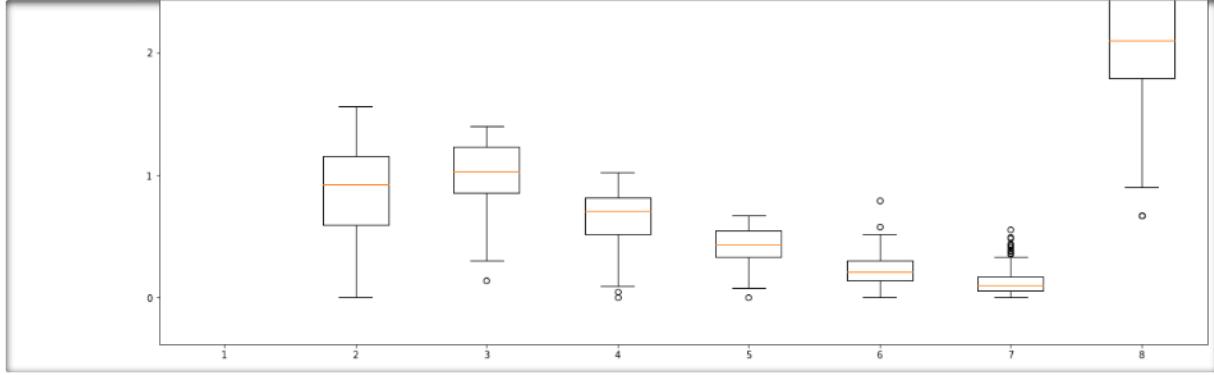


Box Plot and Outliers

```
In [22]: plt.rcParams['figure.figsize'] = (20,20)
plt.boxplot([df_2015.happiness_score, df_2015.gdp_per_capita, df_2015.family, df_2015.health, df_2015.freedom, df_2
```

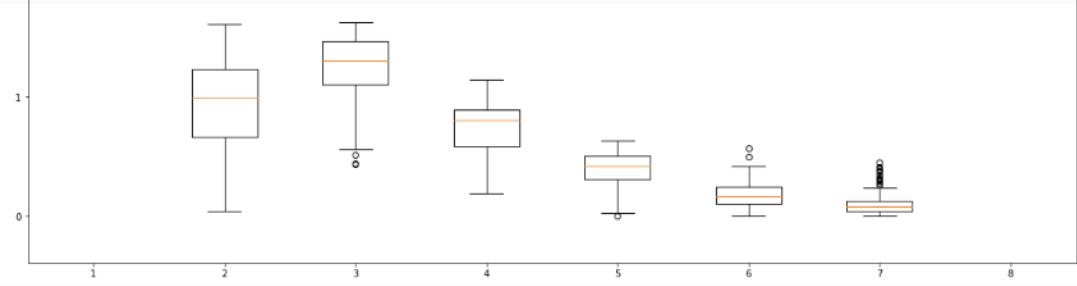


```
In [23]: plt.rcParams['figure.figsize'] = (20,20)
plt.boxplot([df_2016.happiness_score, df_2016.gdp_per_capita, df_2016.family, df_2016.health, df_2016.freedom, df_2
```

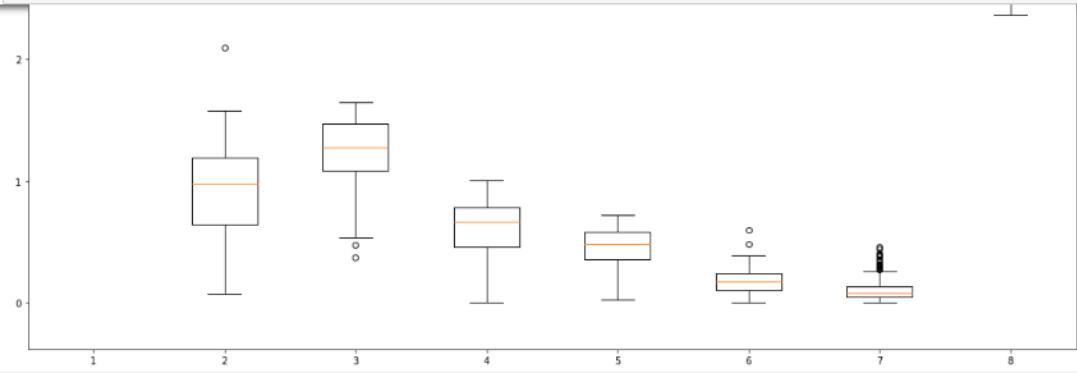


```
In [24]: plt.rcParams['figure.figsize'] = (20,20)
plt.boxplot([df_2017.happiness_score, df_2017.gdp_per_capita, df_2017.family, df_2017.health, df_2017.freedom, df_2
```





```
In [25]: plt.rcParams['figure.figsize'] = (20,20)
plt.boxplot([df_2018.happiness_score, df_2018.gdp_per_capita, df_2018.family, df_2018.health, df_2018.freedom, df_2
```



```
In [27]: df2015 = df_2015
df2016 = df_2016
df2017 = df_2017
df2018 = df_2018
df2019 = df_2019
df2020 = df_2020
```

```
In [28]: x = df2015['dystopia_residual'].median()
y = df2015['dystopia_residual'].quantile(0.10)
z = df2015['dystopia_residual'].quantile(0.95)
print(x,y,z)
df2015['dystopia_residual'] = np.where(df2015['dystopia_residual'] > z, x, df2015['dystopia_residual'])
df2015['dystopia_residual'] = np.where(df2015['dystopia_residual'] < y, x, df2015['dystopia_residual'])

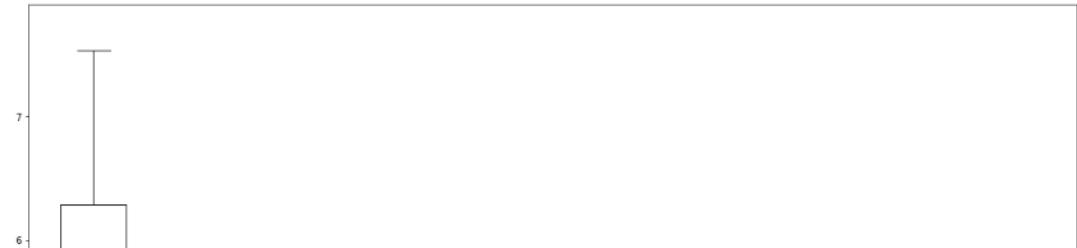
x = df2015['government_trust'].median()
y = df2015['government_trust'].quantile(0.10)
z = df2015['government_trust'].quantile(0.95)
print(x,y,z)
df2015['government_trust'] = np.where(df2015['government_trust'] > z, x, df2015['government_trust'])
df2015['government_trust'] = np.where(df2015['government_trust'] < y, x, df2015['government_trust'])

x = df2015['generosity'].median()
y = df2015['generosity'].quantile(0.10)
z = df2015['generosity'].quantile(0.95)
print(x,y,z)
df2015['generosity'] = np.where(df2015['generosity'] > z, x, df2015['generosity'])
df2015['generosity'] = np.where(df2015['generosity'] < y, x, df2015['generosity'])

x = df2015['family'].median()
y = df2015['family'].quantile(0.10)
z = df2015['family'].quantile(0.95)
print(x,y,z)
df2015['family'] = np.where(df2015['family'] > z, x, df2015['family'])
df2015['family'] = np.where(df2015['family'] < y, x, df2015['family'])
```

```
1.8359999999999999 1.3218 2.7231
0.088000000000000001 0.030800000000000004 0.33045000000000024
0.227 0.0815 0.4892999999999996
1.26 0.8236000000000001 1.5329999999999998
```

```
In [29]: plt.rcParams['figure.figsize'] = (20,20)
plt.boxplot([df_2015.happiness_score, df_2015.gdp_per_capita, df_2015.family, df_2015.health, df_2015.freedom, df_2
'means': []})
```



```
In [30]: x = df2016['dystopia_residual'].median()
y = df2016['dystopia_residual'].quantile(0.10)
z = df2016['dystopia_residual'].quantile(0.95)
print(x,y,z)
df2016['dystopia_residual'] = np.where(df2016['dystopia_residual'] > z, x, df2016['dystopia_residual'])
df2016['dystopia_residual'] = np.where(df2016['dystopia_residual'] < y, x, df2016['dystopia_residual'])

x = df2016['government_trust'].median()
y = df2016['government_trust'].quantile(0.10)
z = df2016['government_trust'].quantile(0.95)
print(x,y,z)
df2016['government_trust'] = np.where(df2016['government_trust'] > z, x, df2016['government_trust'])
df2016['government_trust'] = np.where(df2016['government_trust'] < y, x, df2016['government_trust'])

x = df2016['generosity'].median()
y = df2016['generosity'].quantile(0.10)
z = df2016['generosity'].quantile(0.95)
print(x,y,z)
df2016['generosity'] = np.where(df2016['generosity'] > z, x, df2016['generosity'])
df2016['generosity'] = np.where(df2016['generosity'] < y, x, df2016['generosity'])

x = df2016['family'].median()
y = df2016['family'].quantile(0.10)
z = df2016['family'].quantile(0.95)
print(x,y,z)
df2016['family'] = np.where(df2016['family'] > z, x, df2016['family'])
df2016['family'] = np.where(df2016['family'] < y, x, df2016['family'])

2.0955 1.4566000000000001 3.0431000000000004
0.1045000000000001 0.0279999999999997 0.3902000000000002
0.2125 0.0897999999999999 0.4639500000000001
1.033999999999998 0.636799999999998 1.3204500000000001
```

```
In [31]: x = df2017['dystopia_residual'].median()
y = df2017['dystopia_residual'].quantile(0.10)
z = df2017['dystopia_residual'].quantile(0.95)
print(x,y,z)
df2017['dystopia_residual'] = np.where(df2017['dystopia_residual'] > z, x, df2017['dystopia_residual'])
df2017['dystopia_residual'] = np.where(df2017['dystopia_residual'] < y, x, df2017['dystopia_residual'])

x = df2017['government_trust'].median()
y = df2017['government_trust'].quantile(0.10)
z = df2017['government_trust'].quantile(0.95)
print(x,y,z)
df2017['government_trust'] = np.where(df2017['government_trust'] > z, x, df2017['government_trust'])
df2017['government_trust'] = np.where(df2017['government_trust'] < y, x, df2017['government_trust'])

x = df2017['generosity'].median()
y = df2017['generosity'].quantile(0.10)
z = df2017['generosity'].quantile(0.95)
print(x,y,z)
df2017['generosity'] = np.where(df2017['generosity'] > z, x, df2017['generosity'])
df2017['generosity'] = np.where(df2017['generosity'] < y, x, df2017['generosity'])

x = df2017['family'].median()
y = df2017['family'].quantile(0.10)
z = df2017['family'].quantile(0.95)
print(x,y,z)
df2017['family'] = np.where(df2017['family'] > z, x, df2017['family'])
df2017['family'] = np.where(df2017['family'] < y, x, df2017['family'])

4.505000000000001 2.9352 6.346900000000001
0.0809999999999999 0.0270000000000003 0.3413000000000005
0.172 0.0718 0.3489
1.298 0.8227 1.5395
```

```
In [32]: x = df2018['dystopia_residual'].median()
y = df2018['dystopia_residual'].quantile(0.10)
z = df2018['dystopia_residual'].quantile(0.95)
print(x,y,z)
df2018['dystopia_residual'] = np.where(df2018['dystopia_residual'] > z, x, df2018['dystopia_residual'])
df2018['dystopia_residual'] = np.where(df2018['dystopia_residual'] < y, x, df2018['dystopia_residual'])

x = df2018['government_trust'].median()
y = df2018['government_trust'].quantile(0.10)
z = df2018['government_trust'].quantile(0.95)
print(x,y,z)
df2018['government_trust'] = np.where(df2018['government_trust'] > z, x, df2018['government_trust'])
df2018['government_trust'] = np.where(df2018['government_trust'] < y, x, df2018['government_trust'])

x = df2018['generosity'].median()
y = df2018['generosity'].quantile(0.10)
z = df2018['generosity'].quantile(0.95)
print(x,y,z)
df2018['generosity'] = np.where(df2018['generosity'] > z, x, df2018['generosity'])
df2018['generosity'] = np.where(df2018['generosity'] < y, x, df2018['generosity'])

x = df2018['family'].median()
y = df2018['family'].quantile(0.10)
z = df2018['family'].quantile(0.95)
print(x,y,z)
df2018['family'] = np.where(df2018['family'] > z, x, df2018['family'])
```

```
at2018['family'] = np.where(at2018['family'] > z, x, at2018['family'])
df2018['family'] = np.where(df2018['family'] < y, x, df2018['family'])
```

```
4.5225 3.0314 6.41045
0.0805 0.0287 0.34255000000000013
0.1689999999999998 0.0634 0.35505
1.2735 0.8124 1.5743500000000001
```

```
In [33]: x = df2019['dystopia_residual'].median()
y = df2019['dystopia_residual'].quantile(0.10)
z = df2019['dystopia_residual'].quantile(0.95)
print(x,y,z)
df2019['dystopia_residual'] = np.where(df2019['dystopia_residual'] > z, x, df2019['dystopia_residual'])
df2019['dystopia_residual'] = np.where(df2019['dystopia_residual'] < y, x, df2019['dystopia_residual'])

x = df2019['government_trust'].median()
y = df2019['government_trust'].quantile(0.10)
z = df2019['government_trust'].quantile(0.95)
print(x,y,z)
df2019['government_trust'] = np.where(df2019['government_trust'] > z, x, df2019['government_trust'])
df2019['government_trust'] = np.where(df2019['government_trust'] < y, x, df2019['government_trust'])

x = df2019['generosity'].median()
y = df2019['generosity'].quantile(0.10)
z = df2019['generosity'].quantile(0.95)
print(x,y,z)
df2019['generosity'] = np.where(df2019['generosity'] > z, x, df2019['generosity'])
df2019['generosity'] = np.where(df2019['generosity'] < y, x, df2019['generosity'])

x = df2019['family'].median()
y = df2019['family'].quantile(0.10)
z = df2019['family'].quantile(0.95)
print(x,y,z)
df2019['family'] = np.where(df2019['family'] > z, x, df2019['family'])
df2019['family'] = np.where(df2019['family'] < y, x, df2019['family'])
```

```
2.301 1.7653 3.2023
0.096 0.03240000000000005 0.36565000000000025
0.215 0.08750000000000001 0.4774500000000004
0.861 0.4191000000000001 1.1299000000000001
```

```
In [34]: x = df2020['dystopia_residual'].median()
y = df2020['dystopia_residual'].quantile(0.10)
z = df2020['dystopia_residual'].quantile(0.95)
print(x,y,z)
df2020['dystopia_residual'] = np.where(df2020['dystopia_residual'] > z, x, df2020['dystopia_residual'])
df2020['dystopia_residual'] = np.where(df2020['dystopia_residual'] < y, x, df2020['dystopia_residual'])

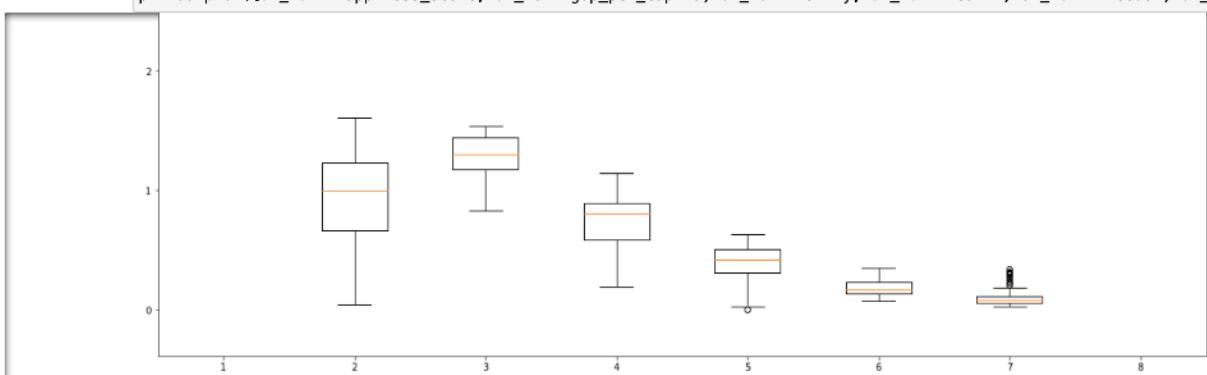
x = df2020['government_trust'].median()
y = df2020['government_trust'].quantile(0.10)
z = df2020['government_trust'].quantile(0.95)
print(x,y,z)
df2020['government_trust'] = np.where(df2020['government_trust'] > z, x, df2020['government_trust'])
df2020['government_trust'] = np.where(df2020['government_trust'] < y, x, df2020['government_trust'])

x = df2020['generosity'].median()
y = df2020['generosity'].quantile(0.10)
z = df2020['generosity'].quantile(0.95)
print(x,y,z)
df2020['generosity'] = np.where(df2020['generosity'] > z, x, df2020['generosity'])
df2020['generosity'] = np.where(df2020['generosity'] < y, x, df2020['generosity'])

x = df2020['family'].median()
y = df2020['family'].quantile(0.10)
z = df2020['family'].quantile(0.95)
print(x,y,z)
df2020['family'] = np.where(df2020['family'] > z, x, df2020['family'])
df2020['family'] = np.where(df2020['family'] < y, x, df2020['family'])
```

```
2.054999999999997 1.2795 2.8099
0.097 0.02940000000000003 0.411900000000000015
0.1705 0.06970000000000001 0.36215
1.2195 0.7559 1.47275
```

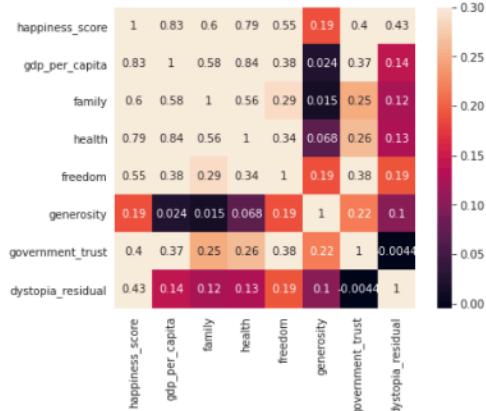
```
In [35]: plt.rcParams['figure.figsize'] = (20,20)
plt.boxplot([df_2017.happiness_score, df_2017.gdp_per_capita, df_2017.family, df_2017.health, df_2017.freedom, df_2
```



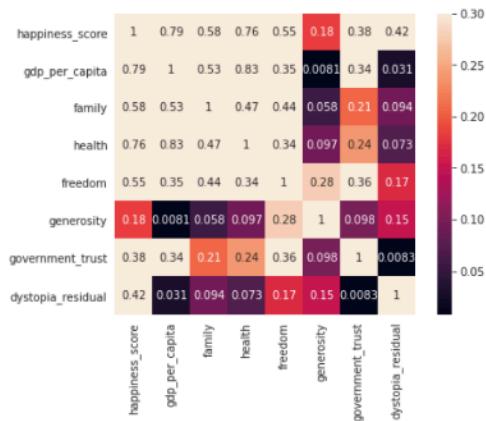
```
In [36]: df2015.columns
```

Correlation Matrix / HeatMap

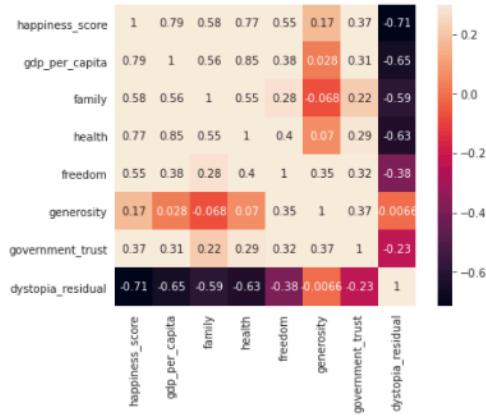
```
In [42]: corr = df2015.corr()
with sns.axes_style("white"):
    f, ax = plt.subplots(figsize=(7, 5))
    ax = sns.heatmap(corr, vmax=.3, square=True, annot=True)
```



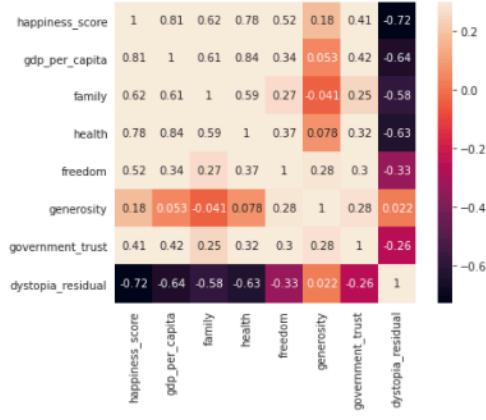
```
In [43]: corr = df2016.corr()
with sns.axes_style("white"):
    f, ax = plt.subplots(figsize=(7, 5))
    ax = sns.heatmap(corr, vmax=.3, square=True, annot=True)
```



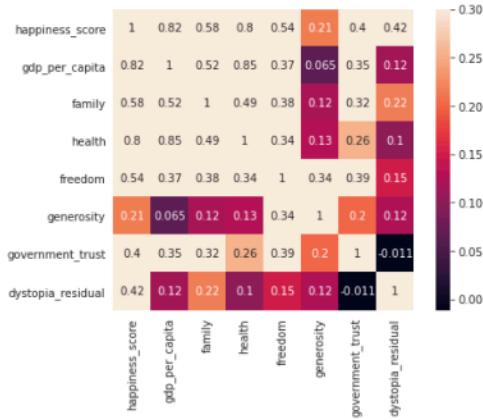
```
In [44]: corr = df2017.corr()
with sns.axes_style("white"):
    f, ax = plt.subplots(figsize=(7, 5))
    ax = sns.heatmap(corr, vmax=.3, square=True, annot=True)
```



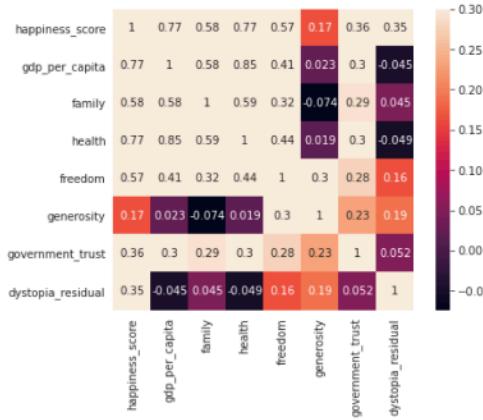
```
In [45]: corr = df2018.corr()
with sns.axes_style("white"):
    f, ax = plt.subplots(figsize=(7, 5))
    ax = sns.heatmap(corr, vmax=.3, square=True, annot=True)
```



```
In [46]: corr = df2019.corr()
with sns.axes_style("white"):
    f, ax = plt.subplots(figsize=(7, 5))
    ax = sns.heatmap(corr, vmax=.3, square=True, annot=True)
```



```
In [47]: corr = df2020.corr()
with sns.axes_style("white"):
    f, ax = plt.subplots(figsize=(7, 5))
    ax = sns.heatmap(corr, vmax=.3, square=True, annot=True)
```



Happiness Score Based On Continent

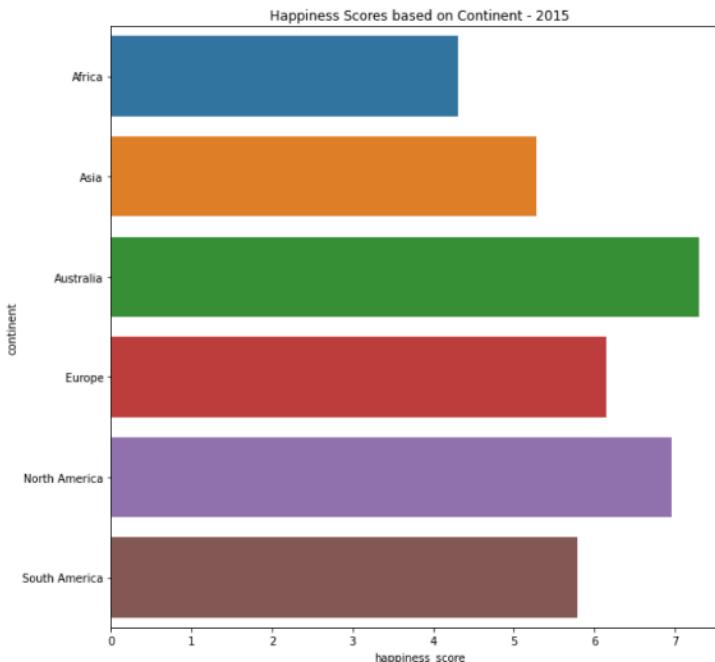
```
In [48]: continent = df2015.groupby("continent").mean().reset_index()
continent
```

Out[48]:

	continent	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual
0	Africa	4.306571	0.592314	1.093514	0.296000	0.363229	0.232543	0.088286	1.867914
1	Asia	5.285179	1.020949	1.246077	0.594846	0.421692	0.252974	0.120513	1.757795
2	Australia	7.299000	1.445000	1.385000	0.830500	0.608000	0.352500	0.194500	2.055500
3	Europe	6.141205	1.312308	1.362692	0.743231	0.419744	0.241128	0.124641	1.956333
4	North America	6.962333	1.392667	1.370867	0.773000	0.510000	0.316667	0.185333	2.080333
5	South America	5.793850	0.948900	1.329400	0.597800	0.437600	0.226250	0.090100	2.106650

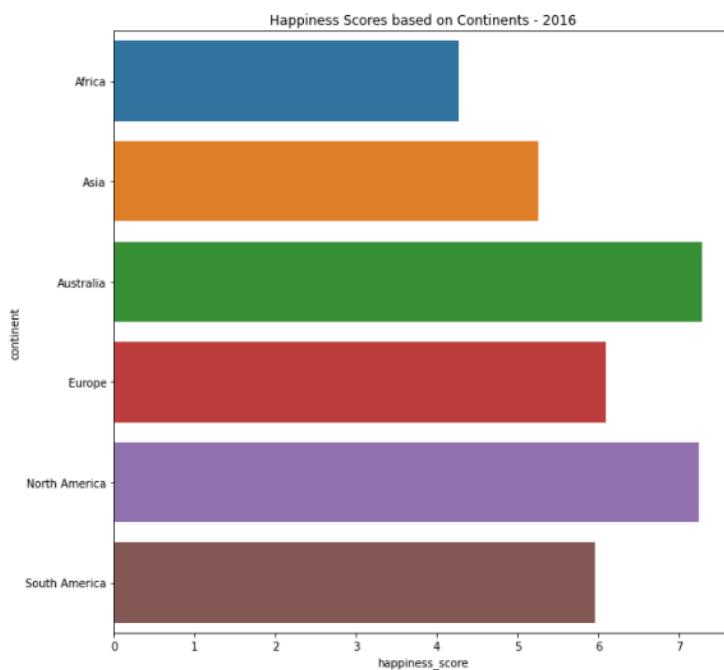
```
In [49]: plt.figure(figsize=(10, 10))
sns.barplot(data=continent, x="happiness_score", y="continent")
plt.title("Happiness Scores based on Continent - 2015")
```

Out[49]: Text(0.5, 1.0, 'Happiness Scores based on Continent - 2015')



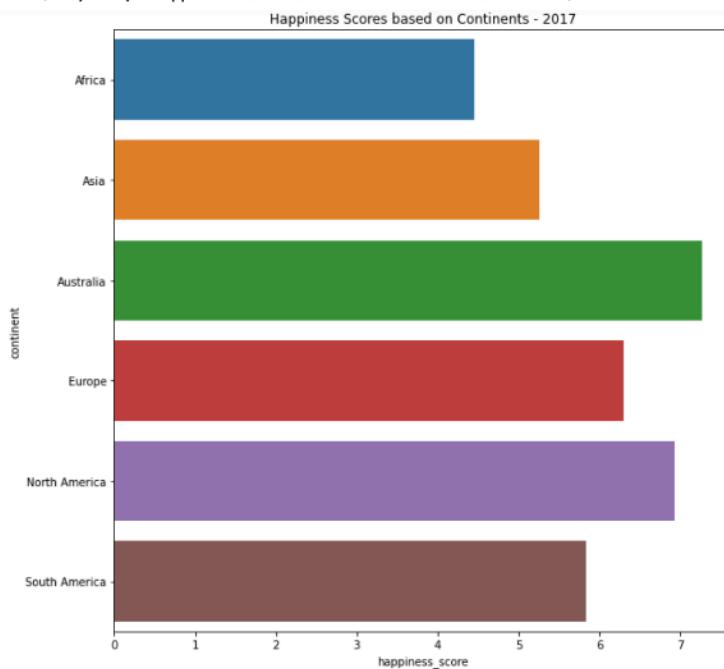
```
In [50]: continent = df2016.groupby("continent").mean().reset_index()
continent
plt.figure(figsize=(10, 10))
sns.barplot(data=continent, x="happiness_score", y="continent")
plt.title("Happiness Scores based on Continents - 2016")
```

Out[50]: Text(0.5, 1.0, 'Happiness Scores based on Continents - 2016')



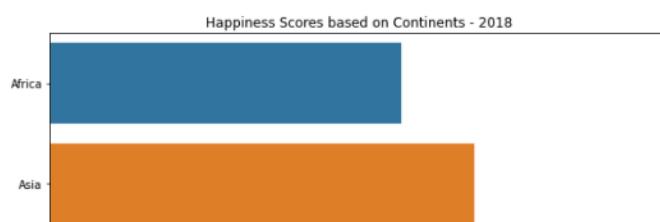
```
In [51]: continent = df2017.groupby("continent").mean().reset_index()
continent
plt.figure(figsize=(10, 10))
sns.barplot(data=continent, x="happiness_score", y="continent")
plt.title("Happiness Scores based on Continents - 2017")
```

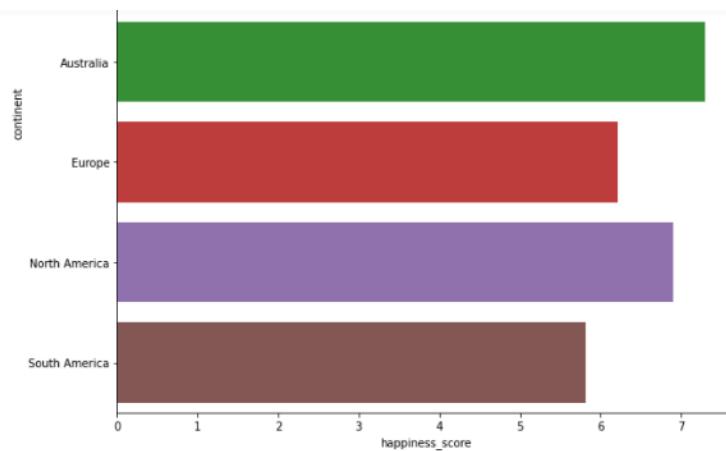
Out[51]: Text(0.5, 1.0, 'Happiness Scores based on Continents - 2017')



```
In [52]: continent = df2018.groupby("continent").mean().reset_index()
continent
plt.figure(figsize=(10, 10))
sns.barplot(data=continent, x="happiness_score", y="continent")
plt.title("Happiness Scores based on Continents - 2018")
```

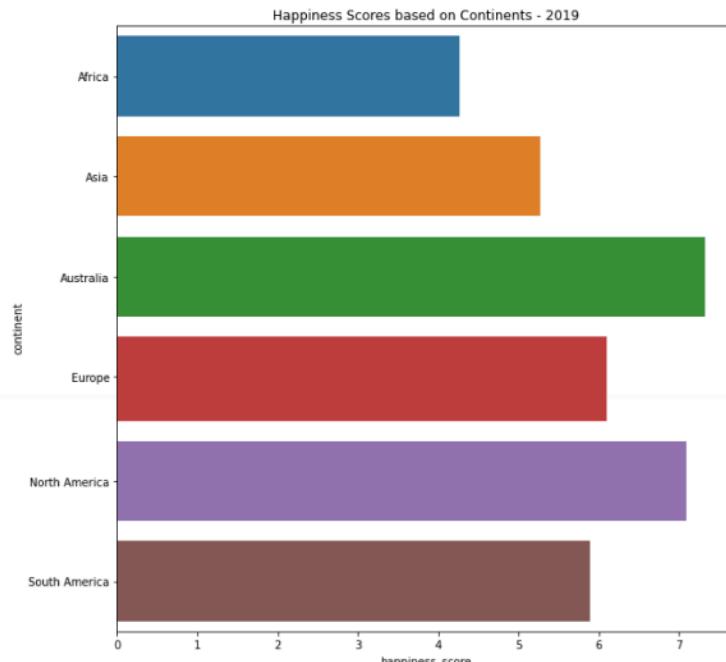
Out[52]: Text(0.5, 1.0, 'Happiness Scores based on Continents - 2018')





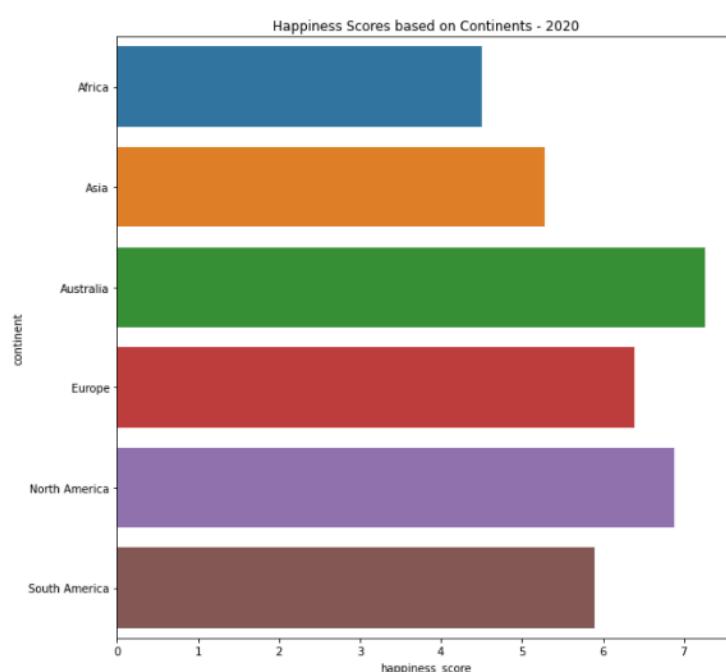
```
In [53]: continent = df2019.groupby("continent").mean().reset_index()
continent
plt.figure(figsize=(10, 10))
sns.barplot(data=continent, x="happiness_score", y="continent")
plt.title("Happiness Scores based on Continents - 2019")
```

Out[53]: Text(0.5, 1.0, 'Happiness Scores based on Continents - 2019')



```
In [54]: continent = df2020.groupby("continent").mean().reset_index()
continent
plt.figure(figsize=(10, 10))
sns.barplot(data=continent, x="happiness_score", y="continent")
plt.title("Happiness Scores based on Continents - 2020")
```

Out[54]: Text(0.5, 1.0, 'Happiness Scores based on Continents - 2020')

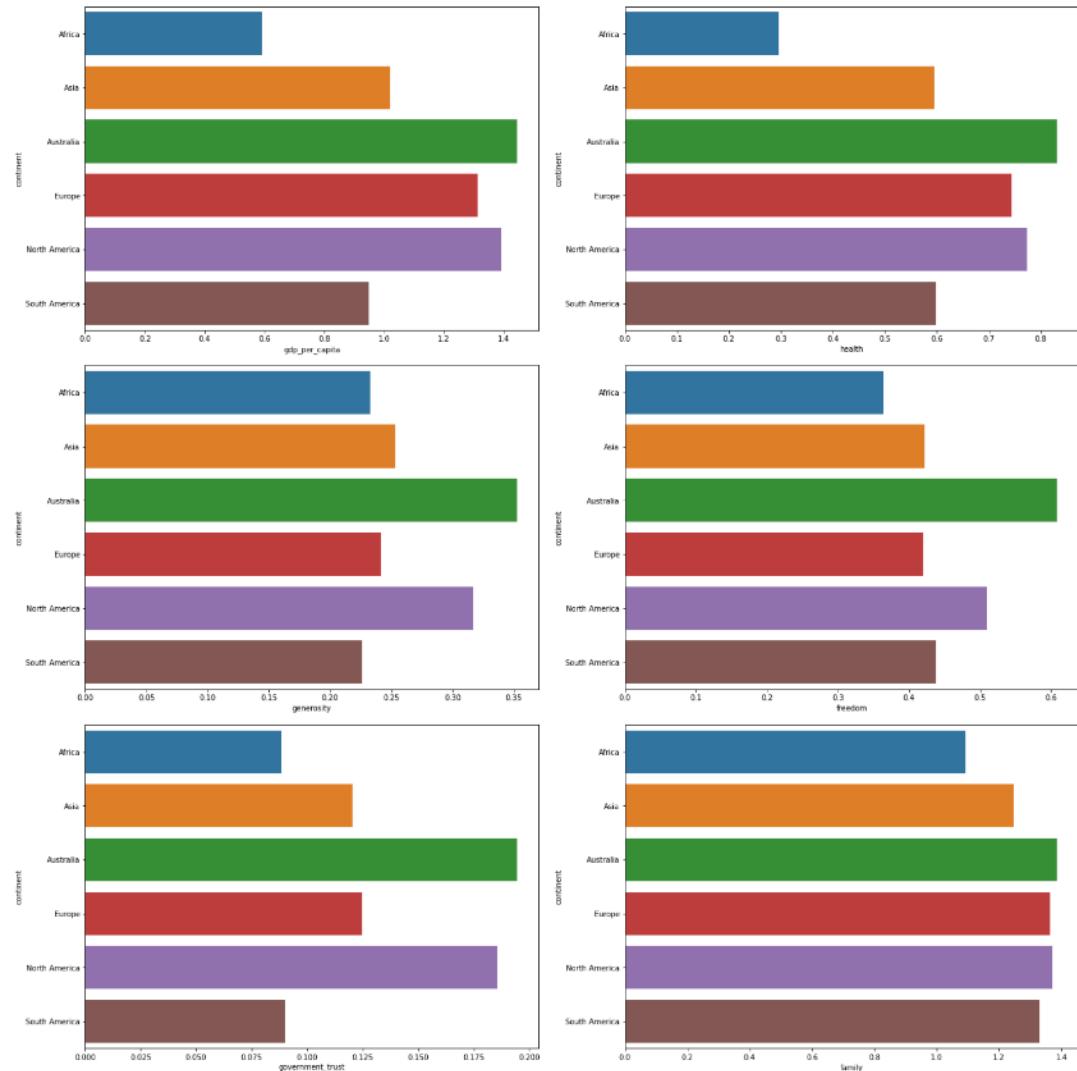


Happiness Factors on the basis on Continent

```
In [55]: continent = df2015.groupby("continent").mean().reset_index()
continent

fig, axes = plt.subplots(nrows=3, ncols=2, constrained_layout=True, figsize=(20,20))
sns.barplot(x='gdp_per_capita',y='continent',data=continent,ax=axes[0,0])
sns.barplot(x='health',y='continent',data=continent,ax=axes[0,1])
sns.barplot(x='generosity' ,y='continent',data=continent,ax=axes[1,0])
sns.barplot(x='freedom' ,y='continent',data=continent,ax=axes[1,1])
sns.barplot(x='government_trust' ,y='continent',data=continent,ax=axes[2,0])
sns.barplot(x='family',y='continent',data=continent,ax=axes[2,1])
```

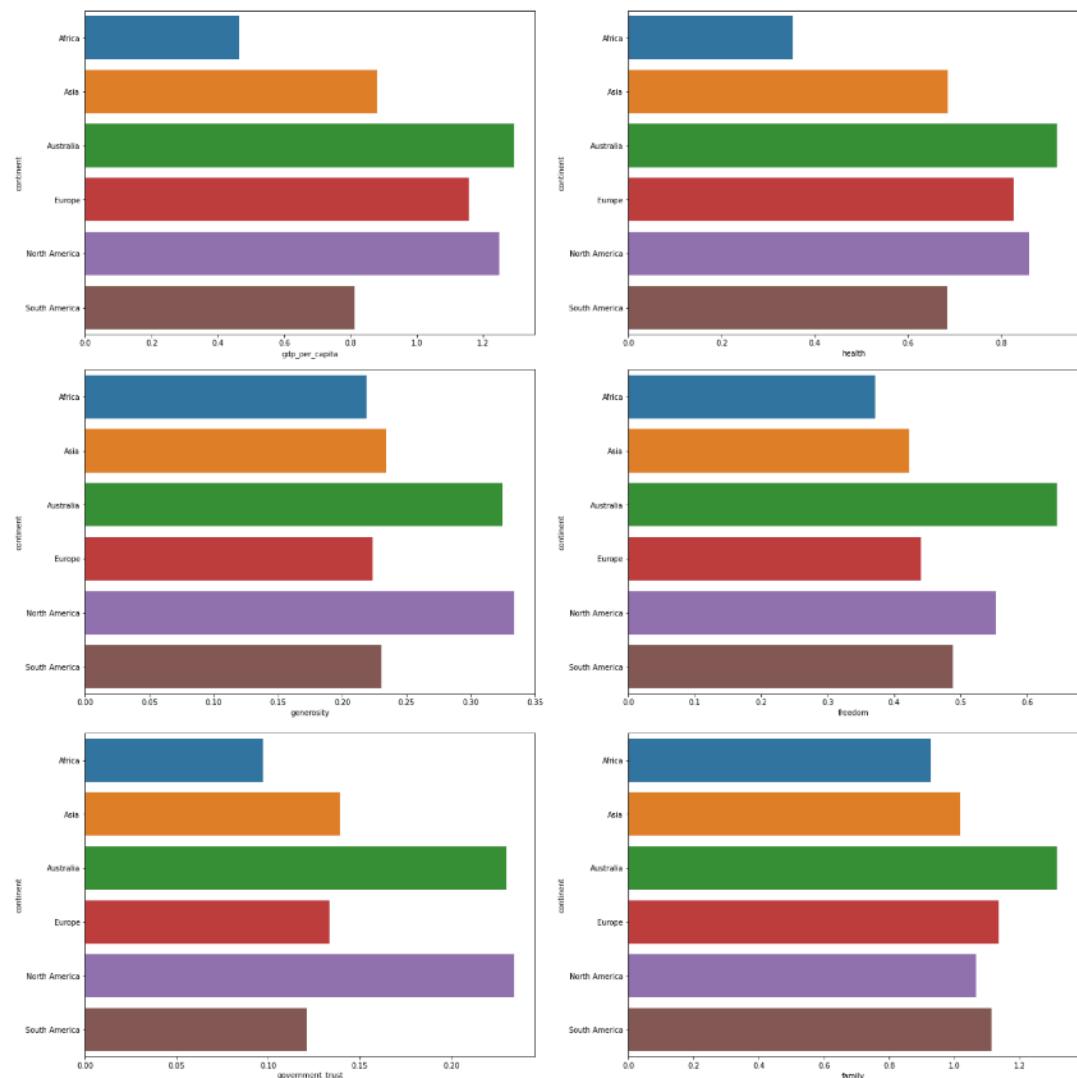
```
Out[55]: <AxesSubplot:xlabel='family', ylabel='continent'>
```



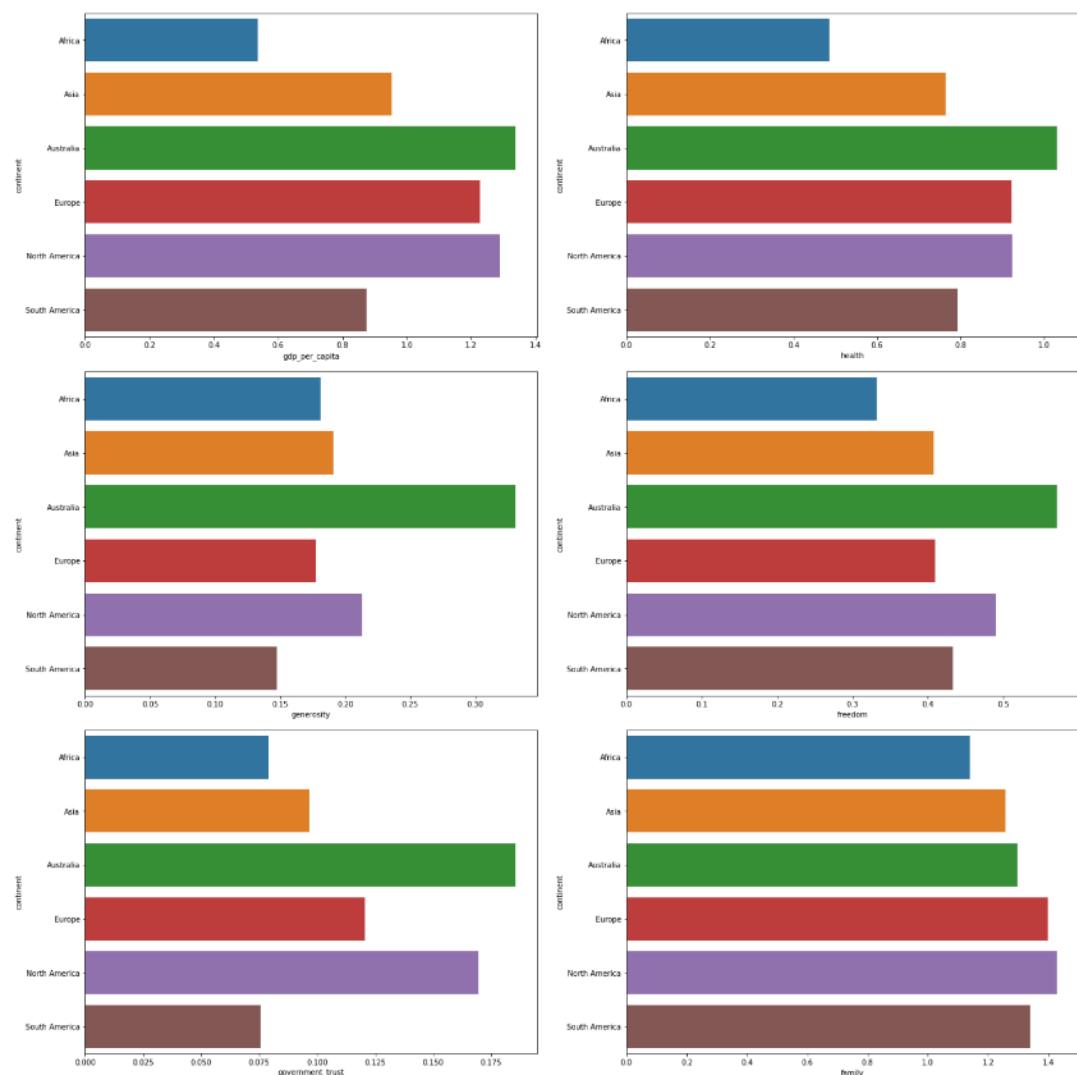
```
In [56]: continent = df2016.groupby("continent").mean().reset_index()
continent
```

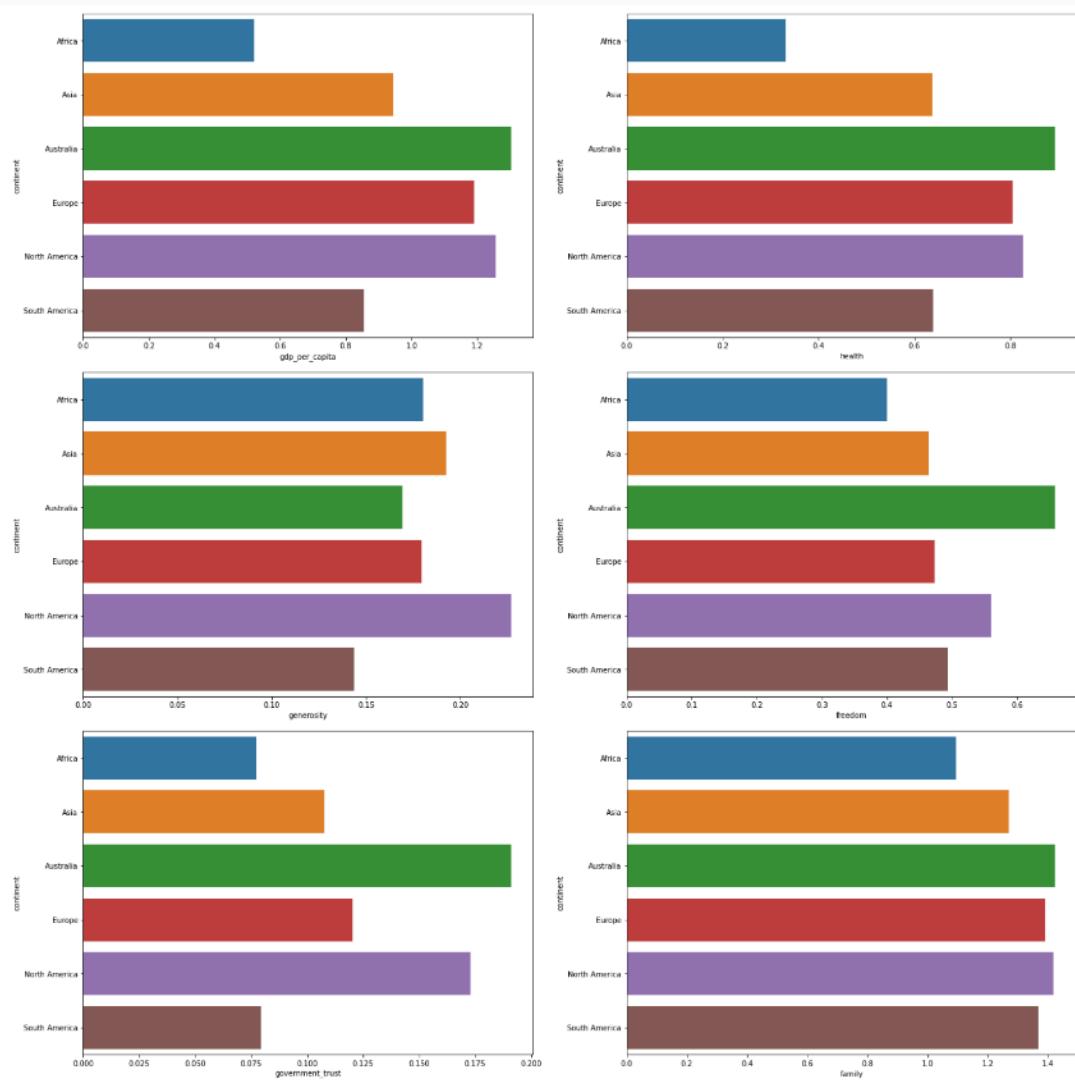
```
fig, axes = plt.subplots(nrows=3, ncols=2, constrained_layout=True, figsize=(20,20))
sns.barplot(x='gdp_per_capita',y='continent',data=continent,ax=axes[0,0])
sns.barplot(x='health',y='continent',data=continent,ax=axes[0,1])
sns.barplot(x='generosity' ,y='continent',data=continent,ax=axes[1,0])
sns.barplot(x='freedom' ,y='continent',data=continent,ax=axes[1,1])
sns.barplot(x='government_trust' ,y='continent',data=continent,ax=axes[2,0])
sns.barplot(x='family',y='continent',data=continent,ax=axes[2,1])
```

```
Out[56]: <AxesSubplot:xlabel='family', ylabel='continent'>
```

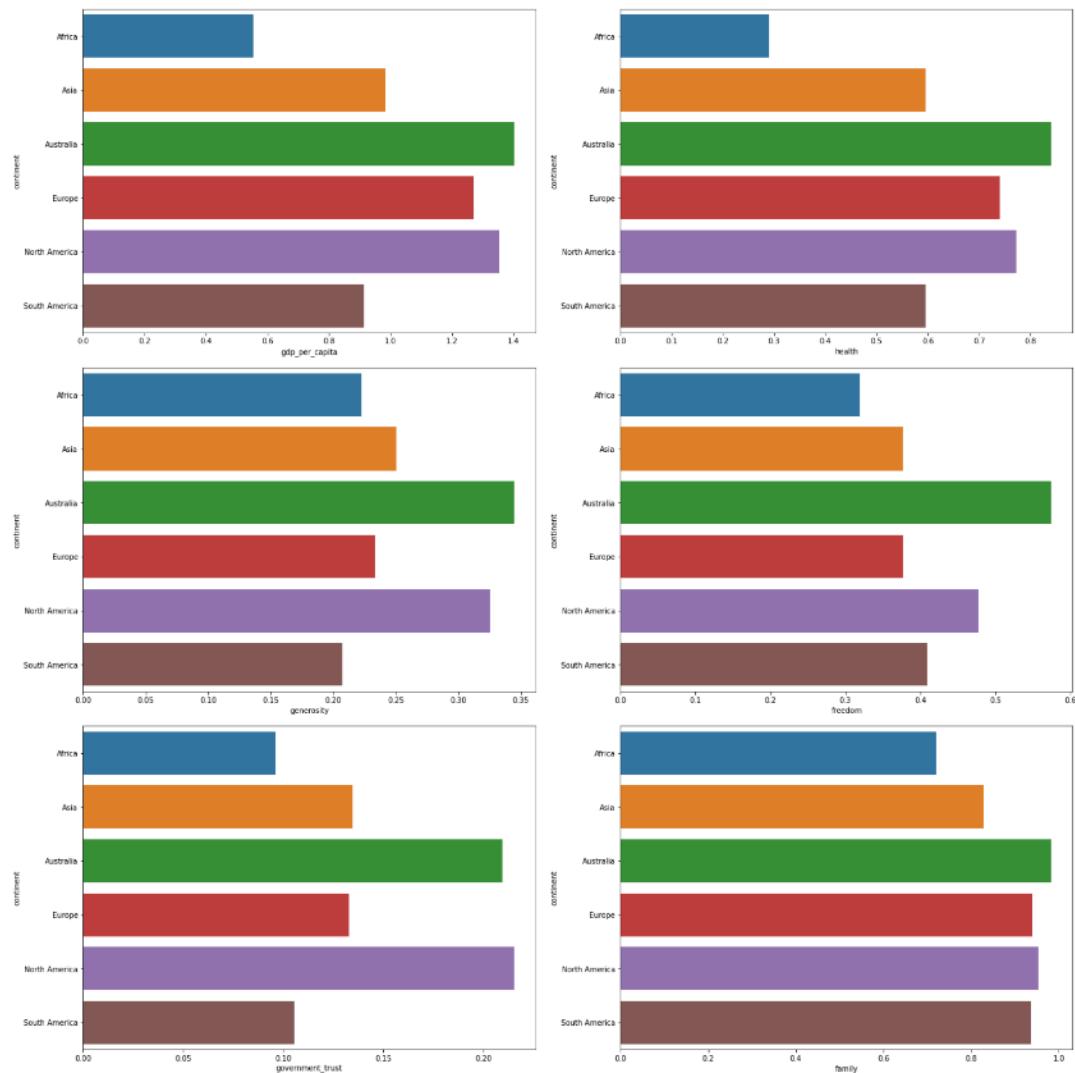


Out[57]: <AxesSubplot:xlabel='family', ylabel='continent'>

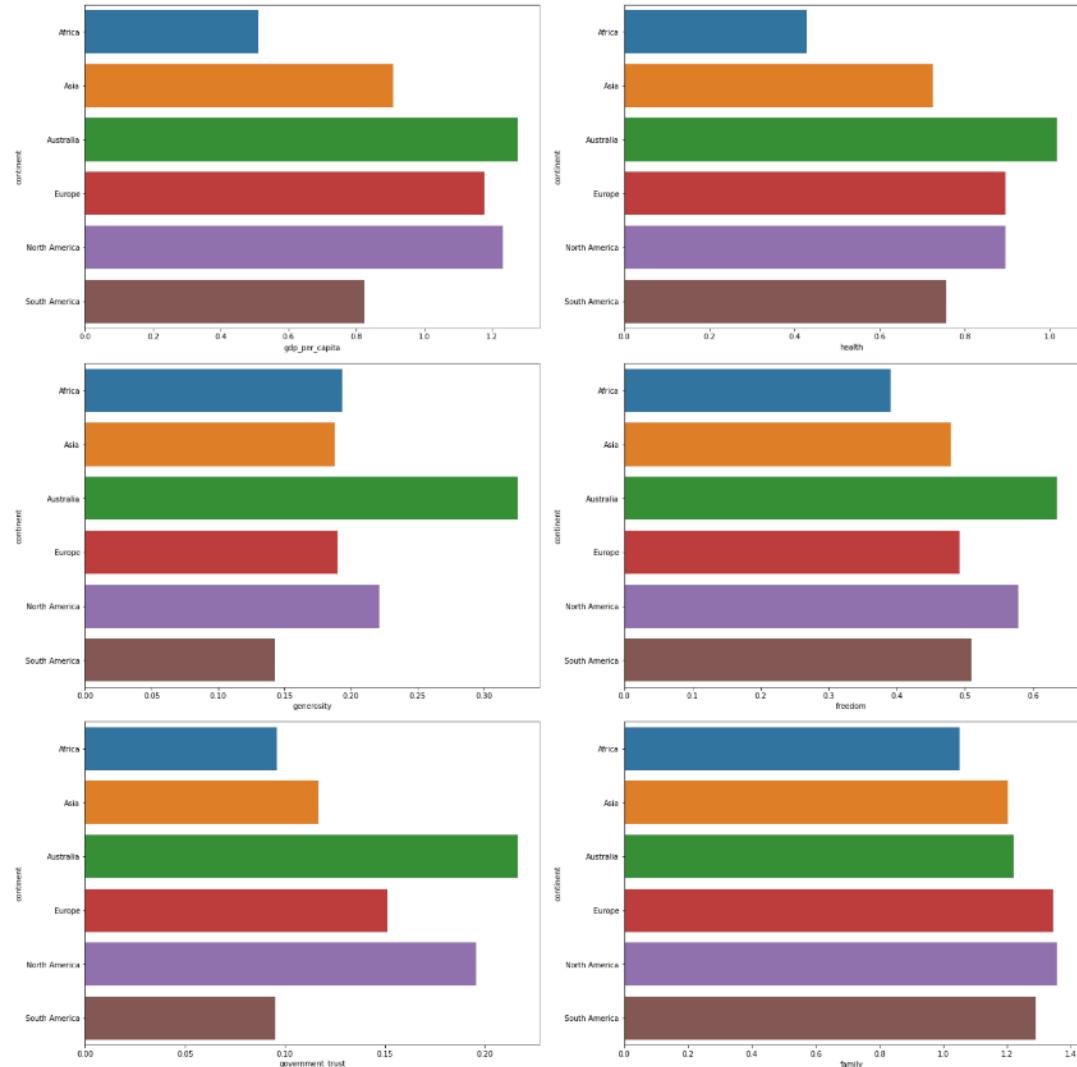




```
In [59]: continent = df2019.groupby("continent").mean().reset_index()
Out[59]: <AxesSubplot:xlabel='family', ylabel='continent'>
```



```
Out[60]: <AxesSubplot:xlabel='family', ylabel='continent'>
```



```
In [61]: from os import chdir
from glob import glob
import pandas as pd
files=list()

for dirname, _, filenames in os.walk('/Users/kritya/Desktop/DataSet/WHR Data Set/'):
    for filename in filenames:
        full_dir = os.path.join(dirname, filename)
        files.append(full_dir)
files.sort()
files
```

```
Out[61]: ['/Users/kritya/Desktop/DataSet/WHR Data Set/.DS_Store',
 '/Users/kritya/Desktop/DataSet/WHR Data Set/2015_report.csv',
 '/Users/kritya/Desktop/DataSet/WHR Data Set/2016_report.csv',
 '/Users/kritya/Desktop/DataSet/WHR Data Set/2017_report.csv',
 '/Users/kritya/Desktop/DataSet/WHR Data Set/2018_report.csv',
 '/Users/kritya/Desktop/DataSet/WHR Data Set/2019_report.csv',
 '/Users/kritya/Desktop/DataSet/WHR Data Set/2020_report.csv']
```

```
In [62]: files.pop(0)
```

```
Out[62]: '/Users/kritya/Desktop/DataSet/WHR Data Set/.DS_Store'
```

```
In [63]: files
```

```
Out[63]: ['/Users/kritya/Desktop/DataSet/WHR Data Set/2015_report.csv',
 '/Users/kritya/Desktop/DataSet/WHR Data Set/2016_report.csv',
 '/Users/kritya/Desktop/DataSet/WHR Data Set/2017_report.csv',
 '/Users/kritya/Desktop/DataSet/WHR Data Set/2018_report.csv',
 '/Users/kritya/Desktop/DataSet/WHR Data Set/2019_report.csv',
 '/Users/kritya/Desktop/DataSet/WHR Data Set/2020_report.csv']
```

```
In [64]: all_df = {}
x=15
for file in files:
    df_name = "World Happiness Report 20{}{}".format(x)
    all_df[df_name] = pd.read_csv(file)
    x += 1
    print(df_name, all_df[df_name].shape)
display(all_df[df_name].head())
```

World Happiness Report 2015 (138, 10)

	country	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual	continent
0	Norway	7.537	1.616	1.534	0.797	0.635	0.362	0.316	2.277	Europe
1	Denmark	7.522	1.482	1.551	0.793	0.626	0.355	0.401	2.314	Europe
2	Iceland	7.504	1.481	1.611	0.834	0.627	0.476	0.154	2.323	Europe
3	Switzerland	7.494	1.565	1.517	0.858	0.620	0.291	0.367	2.277	Europe
4	Finland	7.469	1.444	1.540	0.809	0.618	0.245	0.383	2.430	Europe

World Happiness Report 2016 (138, 10)

	country	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual	continent
0	Switzerland	7.587	1.397	1.350	0.941	0.666	0.297	0.420	2.517	Europe
1	Iceland	7.561	1.302	1.402	0.948	0.629	0.456	0.141	2.702	Europe
2	Denmark	7.527	1.325	1.361	0.875	0.649	0.341	0.484	2.492	Europe
3	Norway	7.522	1.459	1.331	0.885	0.670	0.347	0.365	2.465	Europe
4	Canada	7.427	1.326	1.323	0.906	0.633	0.458	0.330	2.452	North America

World Happiness Report 2017 (138, 10)

	country	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual	continent
0	Finland	7.769	1.340	1.587	0.986	0.596	0.153	0.393	2.231	Europe
1	Denmark	7.600	1.383	1.573	0.996	0.592	0.252	0.410	2.400	Europe
2	Norway	7.554	1.488	1.582	1.028	0.603	0.271	0.341	2.446	Europe
3	Iceland	7.494	1.380	1.624	1.026	0.591	0.354	0.118	2.506	Europe
4	Netherlands	7.488	1.396	1.522	0.999	0.557	0.322	0.298	2.512	Europe

World Happiness Report 2018 (138, 10)

	country	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual	continent
0	Finland	7.632	1.305	1.592	0.874	0.681	0.202	0.393	2.368	Europe
1	Norway	7.594	1.456	1.582	0.861	0.686	0.286	0.340	2.406	Europe
2	Denmark	7.555	1.351	1.590	0.868	0.683	0.284	0.408	2.445	Europe
3	Iceland	7.495	1.343	1.644	0.914	0.677	0.353	0.138	2.505	Europe
4	Switzerland	7.487	1.420	1.549	0.927	0.660	0.256	0.357	2.513	Europe

World Happiness Report 2019 (138, 10)

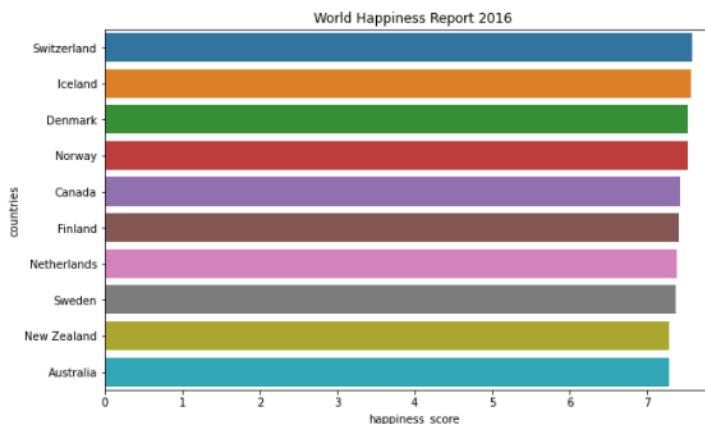
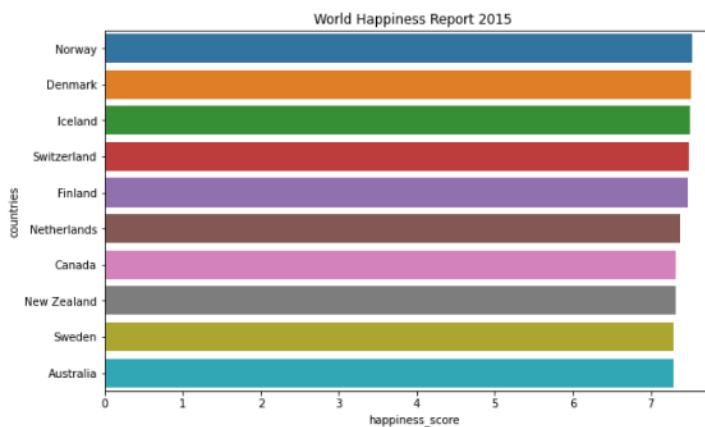
	country	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual	continent
0	Denmark	7.526	1.442	1.164	0.795	0.579	0.362	0.445	2.739	Europe
1	Switzerland	7.509	1.527	1.145	0.863	0.586	0.281	0.412	2.695	Europe
2	Iceland	7.501	1.427	1.183	0.867	0.566	0.477	0.150	2.831	Europe
3	Norway	7.498	1.577	1.127	0.796	0.596	0.379	0.358	2.665	Europe
4	Finland	7.413	1.406	1.135	0.811	0.571	0.255	0.410	2.826	Europe

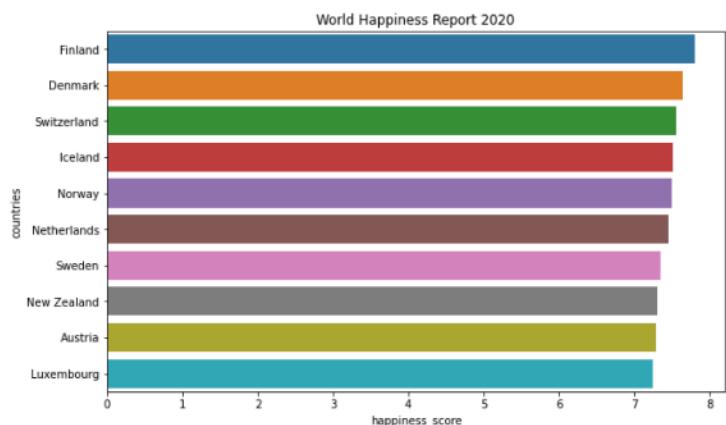
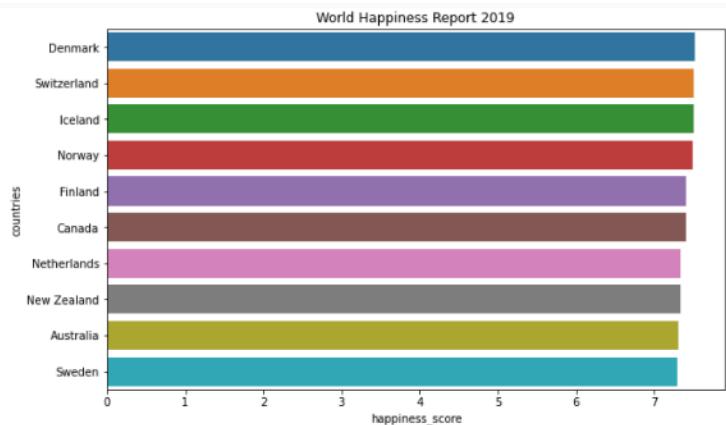
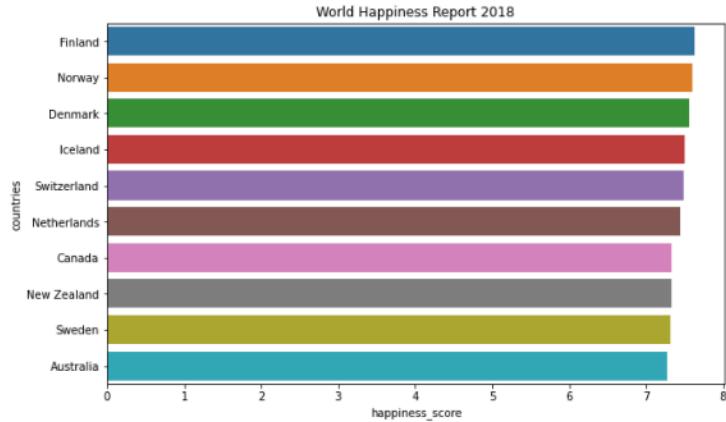
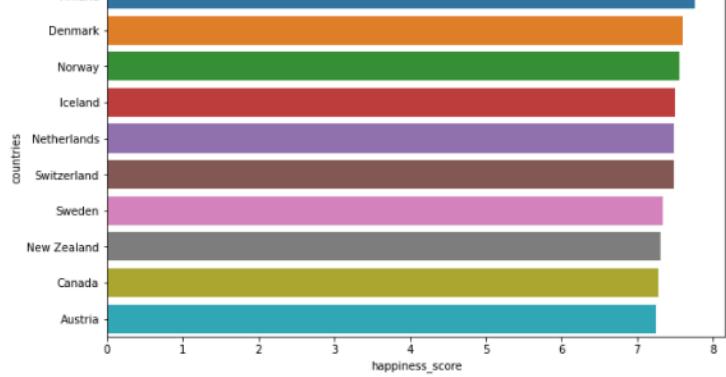
World Happiness Report 2020 (138, 10)

	country	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual	continent
0	Finland	7.809	1.285	1.500	0.961	0.662	0.160	0.478	2.763	Europe
1	Denmark	7.646	1.327	1.503	0.979	0.665	0.243	0.495	2.433	Europe
2	Switzerland	7.560	1.391	1.472	1.041	0.629	0.269	0.408	2.350	Europe
3	Iceland	7.504	1.327	1.548	1.001	0.662	0.362	0.145	2.461	Europe
4	Norway	7.488	1.424	1.495	1.008	0.670	0.288	0.434	2.168	Europe

Top and Bottom 10 countries based on Happiness Score

```
In [65]: for df_name, df in all_df.items():
    fig, ax = plt.subplots(1, 1, figsize = (10, 6))
    plt.title(df_name)
    sns.barplot(x = "happiness_score", y = "country", data=df.head(10))
    ax.set_ylabel('countries')
```

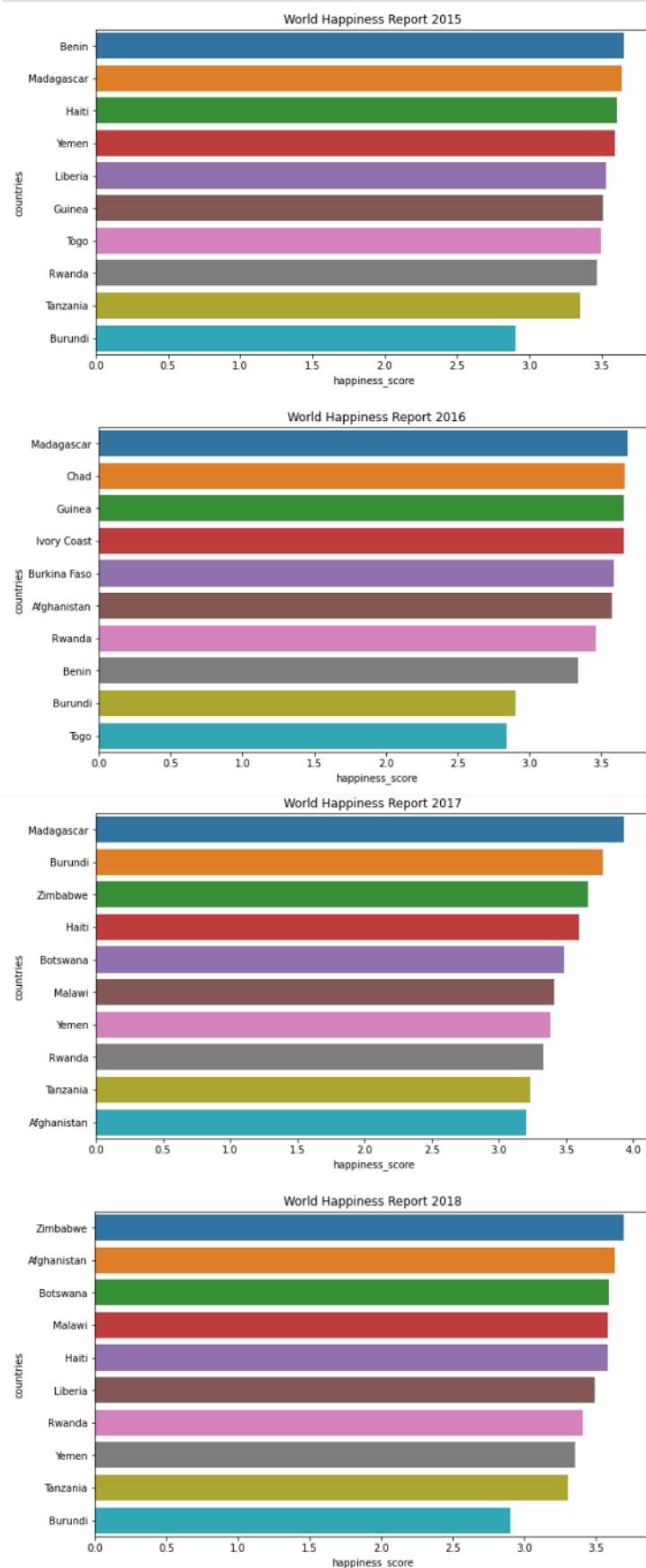


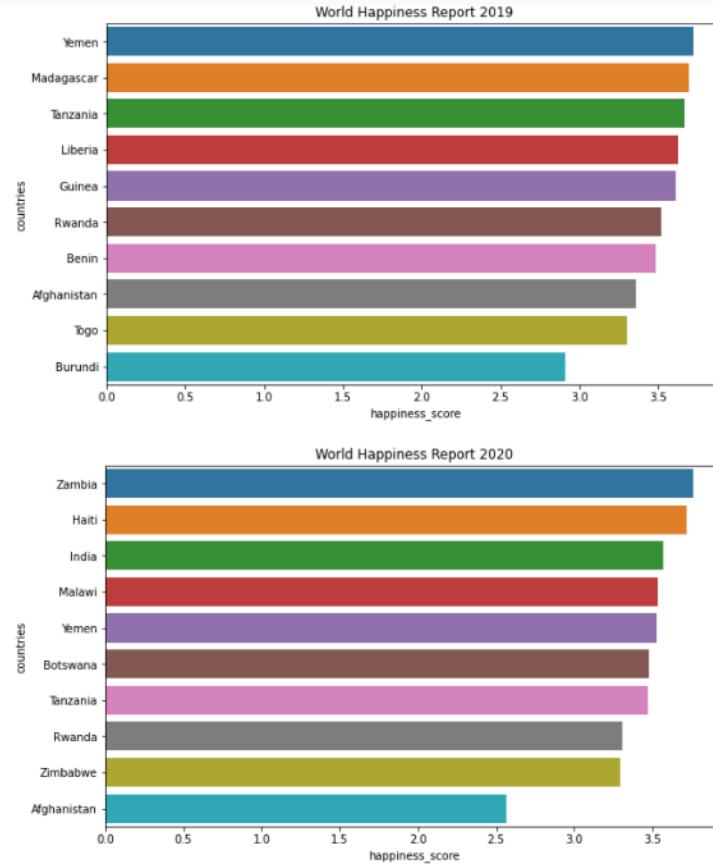


```
In [66]: for df_name, df in all_df.items():
    fig, ax = plt.subplots(1, 1, figsize = (10, 6))
    plt.title(df_name)
    sns.barplot(x = "happiness_score", y = "country", data=df.tail(10))
    ax.set_ylabel('countries')
```



```
In [66]: for df_name, df in all_df.items():
    fig, ax = plt.subplots(1, 1, figsize = (10, 6))
    plt.title(df_name)
    sns.barplot(x = "happiness_score", y = "country", data=df.tail(10))
    ax.set_ylabel('countries')
```



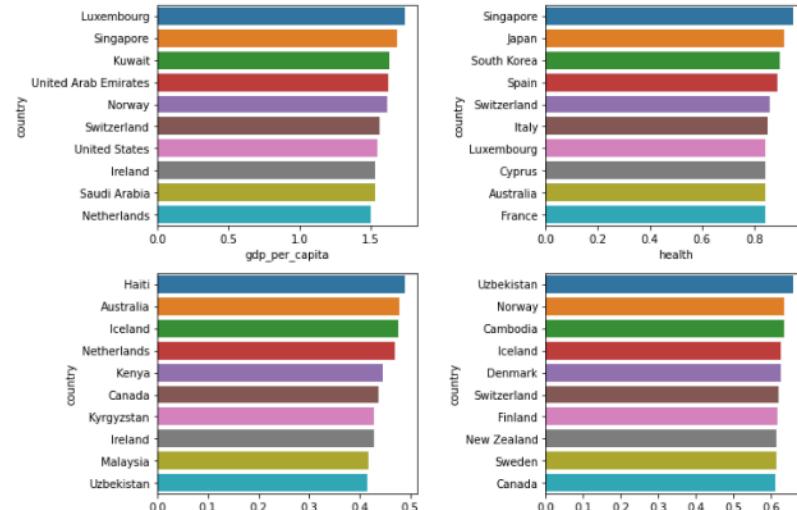


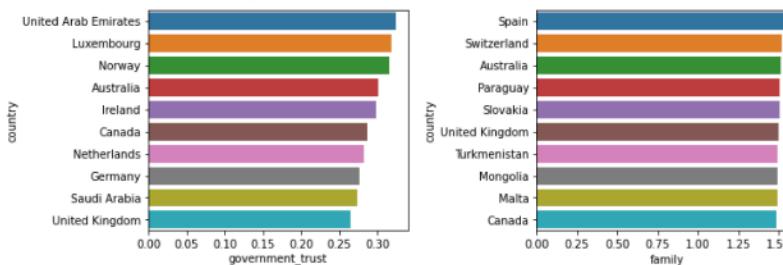
Top 10 Happiest Countries based on factors

2015

```
In [67]: fig, axes = plt.subplots(nrows=3, ncols=2, constrained_layout=True, figsize=(10,10))
sns.barplot(x='gdp_per_capita',y='country',
            data=df2015.nlargest(10,'gdp_per_capita'),
            ax=axes[0,0],palette=None)
sns.barplot(x='health',y='country',
            data=df2015.nlargest(10,'health'),
            ax=axes[0,1],palette=None)
sns.barplot(x='generosity',y='country',
            data=df2015.nlargest(10,'generosity'),
            ax=axes[1,0],palette=None)
sns.barplot(x='freedom',y='country',
            data=df2015.nlargest(10,'freedom'),
            ax=axes[1,1],palette=None)
sns.barplot(x='government_trust',y='country',
            data=df2015.nlargest(10,'government_trust'),
            ax=axes[2,0],palette=None)
sns.barplot(x='family',y='country',
            data=df2015.nlargest(10,'family'),
            ax=axes[2,1],palette=None)
```

Out[67]: <AxesSubplot:xlabel='family', ylabel='country'>

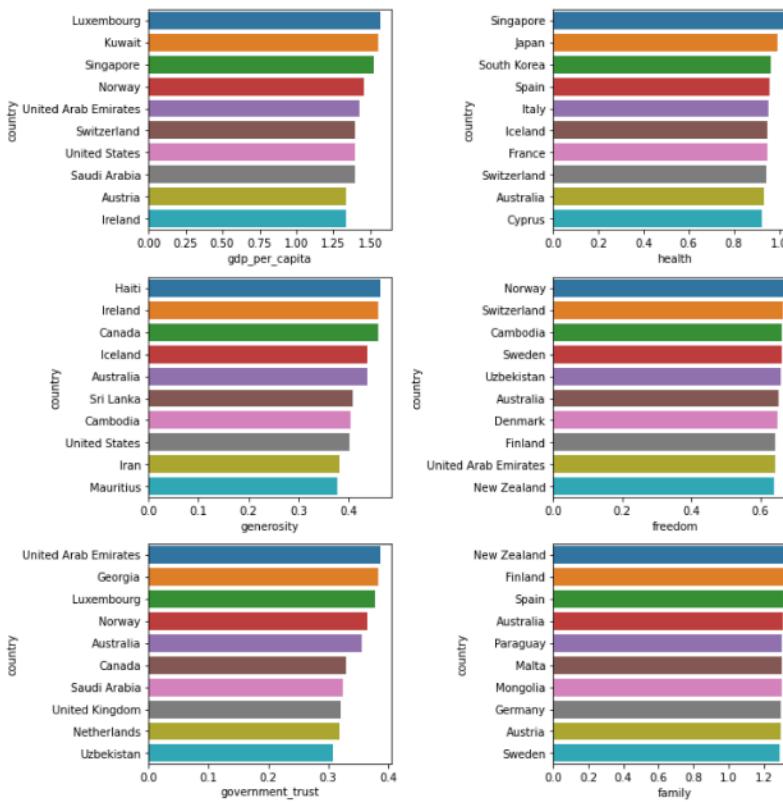




2016

```
In [68]: fig, axes = plt.subplots(nrows=3, ncols=2, constrained_layout=True, figsize=(10,10))
sns.barplot(x='gdp_per_capita',y='country',
            data=df2016.nlargest(10,'gdp_per_capita'),
            ax=axes[0,0],palette=None)
sns.barplot(x='health',y='country',
            data=df2016.nlargest(10,'health'),
            ax=axes[0,1],palette=None)
sns.barplot(x='generosity',y='country',
            data=df2016.nlargest(10,'generosity'),
            ax=axes[1,0],palette=None)
sns.barplot(x='freedom',y='country',
            data=df2016.nlargest(10,'freedom'),
            ax=axes[1,1],palette=None)
sns.barplot(x='government_trust',y='country',
            data=df2016.nlargest(10,'government_trust'),
            ax=axes[2,0],palette=None)
sns.barplot(x='family',y='country',
            data=df2016.nlargest(10,'family'),
            ax=axes[2,1],palette=None)
```

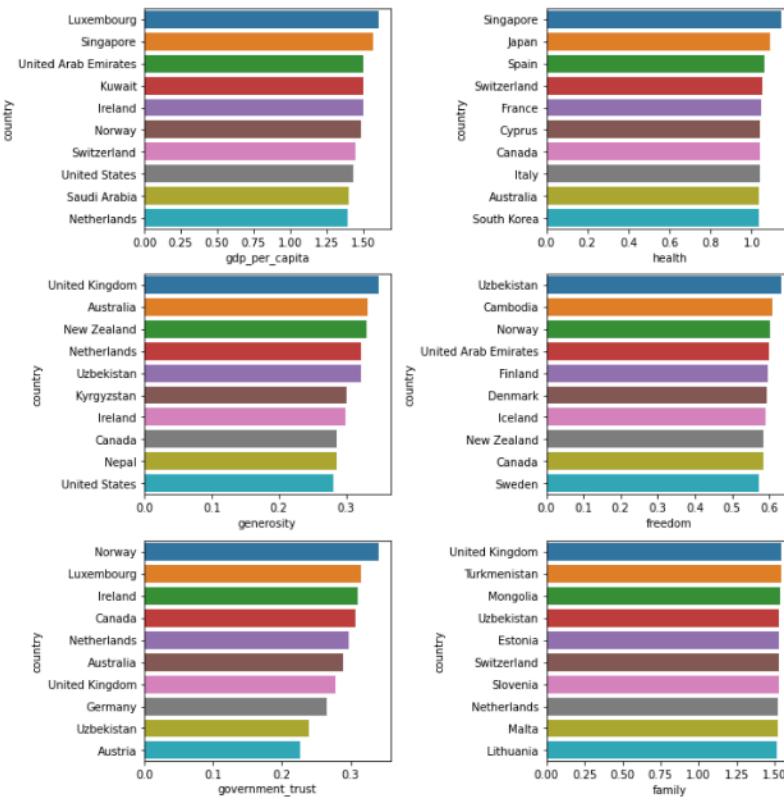
Out[68]: <AxesSubplot:xlabel='family', ylabel='country'>



2017

```
In [69]: fig, axes = plt.subplots(nrows=3, ncols=2, constrained_layout=True, figsize=(10,10))
sns.barplot(x='gdp_per_capita',y='country',
            data=df2017.nlargest(10,'gdp_per_capita'),
            ax=axes[0,0],palette=None)
sns.barplot(x='health',y='country',
            data=df2017.nlargest(10,'health'),
            ax=axes[0,1],palette=None)
sns.barplot(x='generosity',y='country',
            data=df2017.nlargest(10,'generosity'),
            ax=axes[1,0],palette=None)
sns.barplot(x='freedom',y='country',
            data=df2017.nlargest(10,'freedom'),
            ax=axes[1,1],palette=None)
sns.barplot(x='government_trust',y='country',
            data=df2017.nlargest(10,'government_trust'),
            ax=axes[2,0],palette=None)
sns.barplot(x='family',y='country',
            data=df2017.nlargest(10,'family'),
            ax=axes[2,1],palette=None)
```

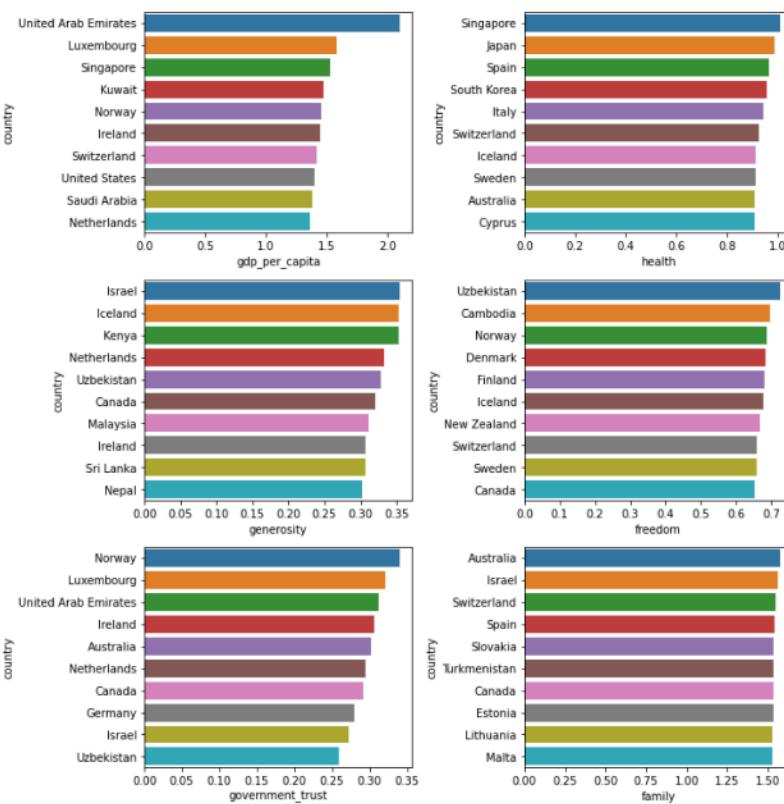
Out[69]: <AxesSubplot:xlabel='family', ylabel='country'>



2018

```
In [70]: fig, axes = plt.subplots(nrows=3, ncols=2, constrained_layout=True, figsize=(10,10))
sns.barplot(x='gdp_per_capita',y='country',
            data=df2018.nlargest(10,'gdp_per_capita'),
            ax=axes[0,0],palette=None)
sns.barplot(x='health',y='country',
            data=df2018.nlargest(10,'health'),
            ax=axes[0,1],palette=None)
sns.barplot(x='generosity',y='country',
            data=df2018.nlargest(10,'generosity'),
            ax=axes[1,0],palette=None)
sns.barplot(x='freedom',y='country',
            data=df2018.nlargest(10,'freedom'),
            ax=axes[1,1],palette=None)
sns.barplot(x='government_trust',y='country',
            data=df2018.nlargest(10,'government_trust'),
            ax=axes[2,0],palette=None)
sns.barplot(x='family',y='country',
            data=df2018.nlargest(10,'family'),
            ax=axes[2,1],palette=None)
```

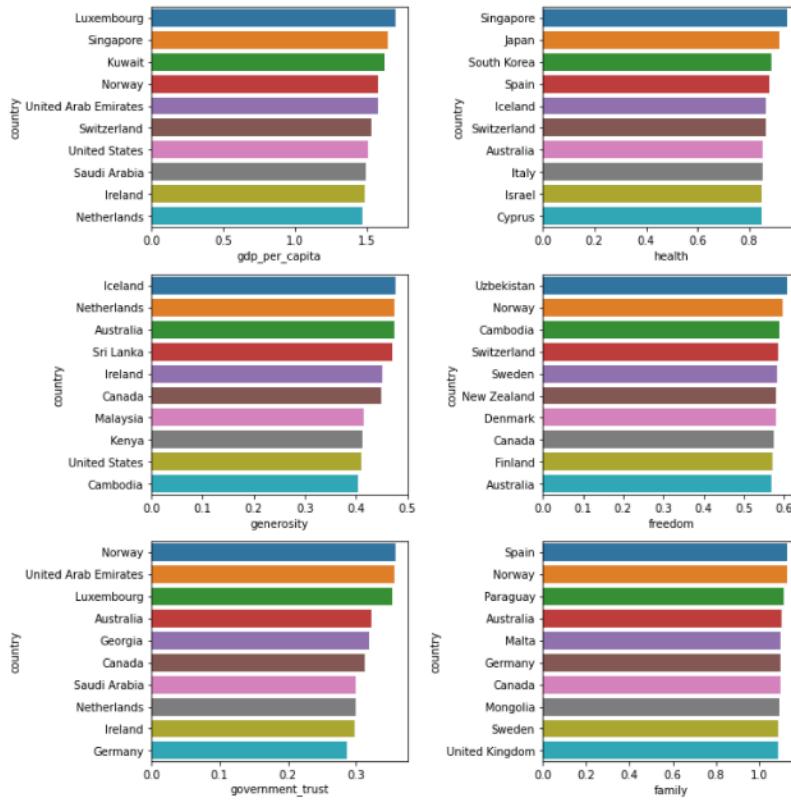
Out[70]: <AxesSubplot:xlabel='family', ylabel='country'>



2019

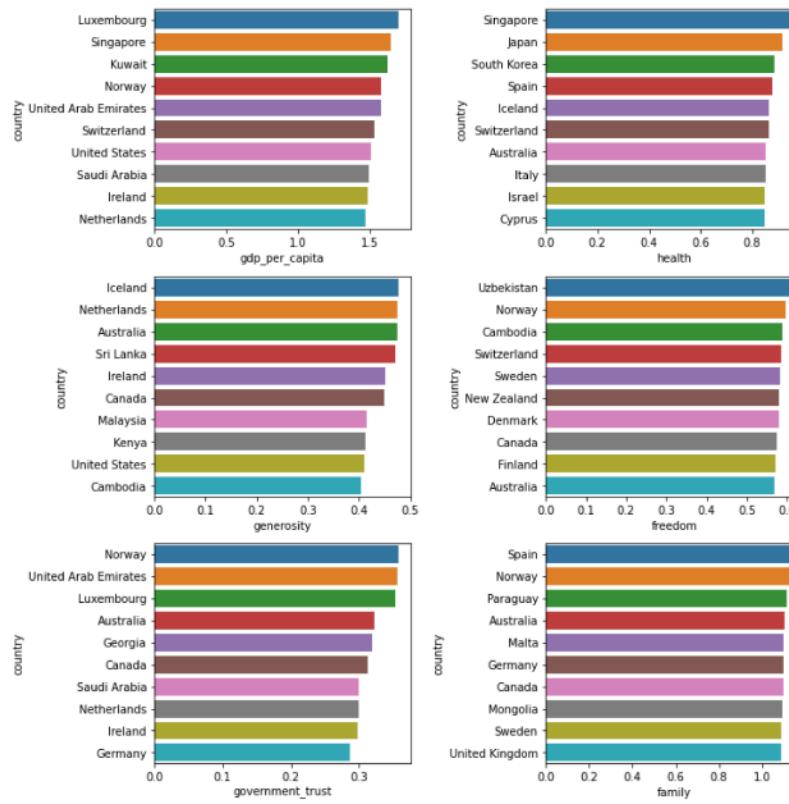
```
In [71]: fig, axes = plt.subplots(nrows=3, ncols=2, constrained_layout=True, figsize=(10,10))
sns.barplot(x='gdp_per_capita',y='country',
            data=df2019.nlargest(10,'gdp_per_capita'),
            ax=axes[0,0],palette=None)
sns.barplot(x='health' ,y='country',
            data=df2019.nlargest(10,'health'),
            ax=axes[0,1],palette=None)
sns.barplot(x='generosity' ,y='country',
            data=df2019.nlargest(10,'generosity'),
            ax=axes[1,0],palette=None)
sns.barplot(x='freedom' ,y='country',
            data=df2019.nlargest(10,'freedom'),
            ax=axes[1,1],palette=None)
sns.barplot(x='government_trust' ,y='country',
            data=df2019.nlargest(10,'government_trust'),
            ax=axes[2,0],palette=None)
sns.barplot(x='family' ,y='country',
            data=df2019.nlargest(10,'family'),
            ax=axes[2,1],palette=None)
```

Out[71]: <AxesSubplot:xlabel='family', ylabel='country'>



```
sns.barplot(x='health' ,y='country',
            data=df2020.nlargest(10,'health'),
            ax=axes[0,1],palette=None)
sns.barplot(x='generosity' ,y='country',
            data=df2020.nlargest(10,'generosity'),
            ax=axes[1,0],palette=None)
sns.barplot(x='freedom' ,y='country',
            data=df2020.nlargest(10,'freedom'),
            ax=axes[1,1],palette=None)
sns.barplot(x='government_trust' ,y='country',
            data=df2020.nlargest(10,'government_trust'),
            ax=axes[2,0],palette=None)
sns.barplot(x='family' ,y='country',
            data=df2020.nlargest(10,'family'),
            ax=axes[2,1],palette=None)
```

Out[72]: <AxesSubplot:xlabel='family', ylabel='country'>

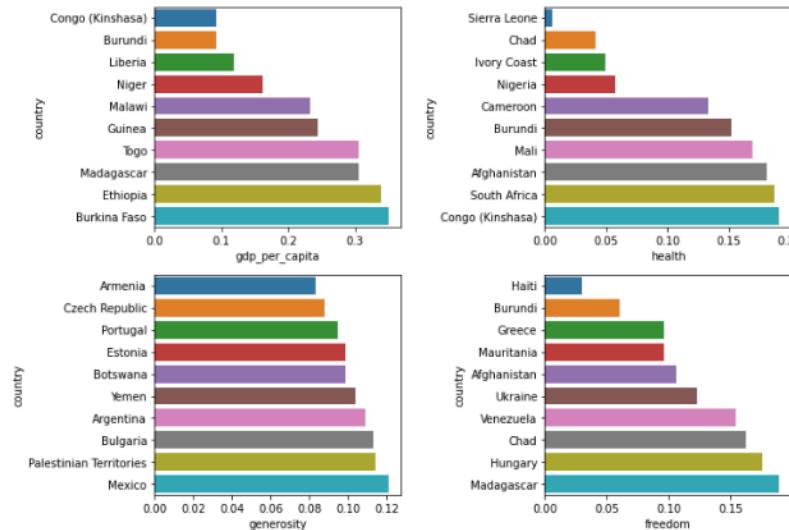


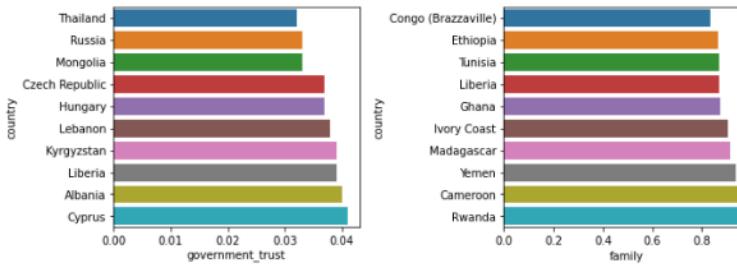
Top 10 Unhappiest Countries based on Factors

2015

```
In [73]: fig, axes = plt.subplots(nrows=3, ncols=2, constrained_layout=True, figsize=(10,10))
sns.barplot(x='gdp_per_capita',y='country',
            data=df2015.nsmallest(10,'gdp_per_capita'),
            ax=axes[0,0],palette=None)
sns.barplot(x='health',y='country',
            data=df2015.nsmallest(10,'health'),
            ax=axes[0,1],palette=None)
sns.barplot(x='generosity',y='country',
            data=df2015.nsmallest(10,'generosity'),
            ax=axes[1,0],palette=None)
sns.barplot(x='freedom',y='country',
            data=df2015.nsmallest(10,'freedom'),
            ax=axes[1,1],palette=None)
sns.barplot(x='government_trust',y='country',
            data=df2015.nsmallest(10,'government_trust'),
            ax=axes[2,0],palette=None)
sns.barplot(x='family',y='country',
            data=df2015.nsmallest(10,'family'),
            ax=axes[2,1],palette=None)
```

Out[73]: <AxesSubplot:xlabel='family', ylabel='country'>

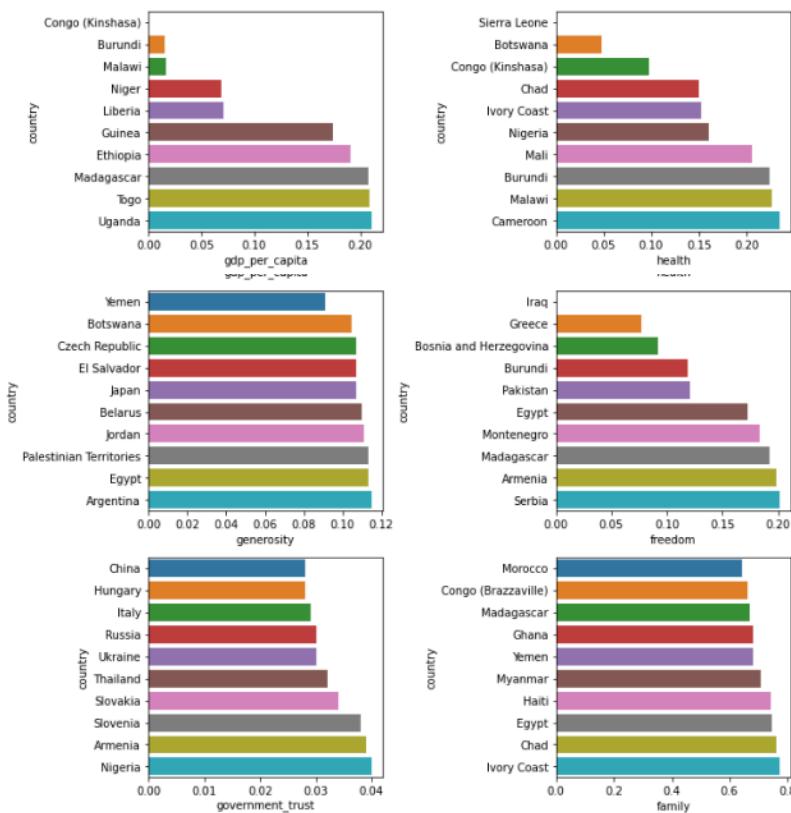




2016

```
In [74]: fig, axes = plt.subplots(nrows=3, ncols=2, constrained_layout=True, figsize=(10,10))
sns.barplot(x='gdp_per_capita',y='country',
            data=df2016.nsmallest(10,'gdp_per_capita'),
            ax=axes[0,0],palette=None)
sns.barplot(x='health',y='country',
            data=df2016.nsmallest(10,'health'),
            ax=axes[0,1],palette=None)
sns.barplot(x='generosity',y='country',
            data=df2016.nsmallest(10,'generosity'),
            ax=axes[1,0],palette=None)
sns.barplot(x='freedom',y='country',
            data=df2016.nsmallest(10,'freedom'),
            ax=axes[1,1],palette=None)
sns.barplot(x='government_trust',y='country',
            data=df2016.nsmallest(10,'government_trust'),
            ax=axes[2,0],palette=None)
sns.barplot(x='family',y='country',
            data=df2016.nsmallest(10,'family'),
            ax=axes[2,1],palette=None)
```

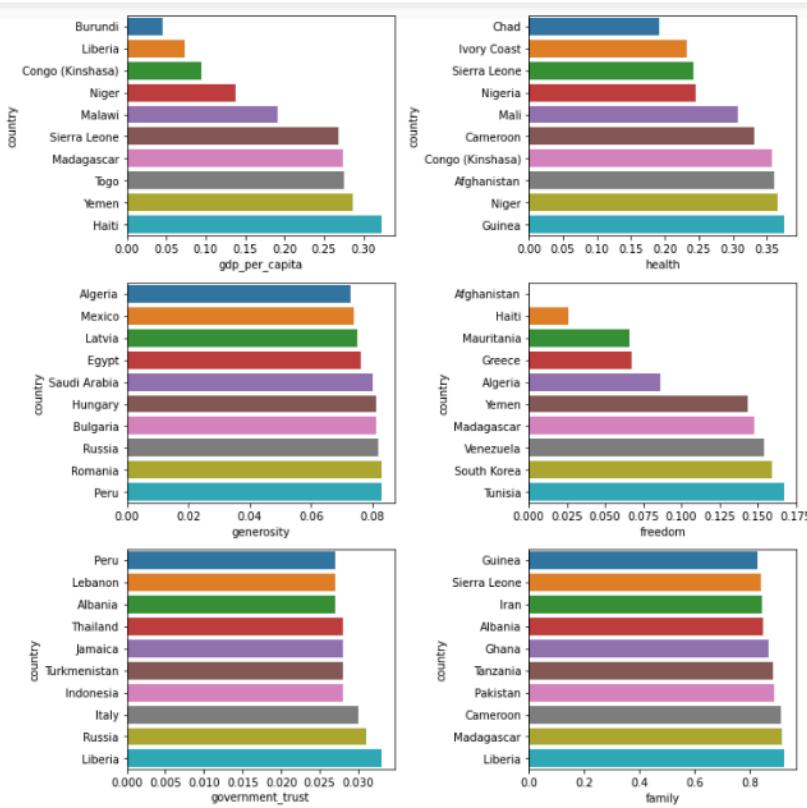
Out[74]: <AxesSubplot:xlabel='family', ylabel='country'>



2017

```
In [75]: fig, axes = plt.subplots(nrows=3, ncols=2, constrained_layout=True, figsize=(10,10))
sns.barplot(x='gdp_per_capita',y='country',
            data=df2017.nsmallest(10,'gdp_per_capita'),
            ax=axes[0,0],palette=None)
sns.barplot(x='health',y='country',
            data=df2017.nsmallest(10,'health'),
            ax=axes[0,1],palette=None)
sns.barplot(x='generosity',y='country',
            data=df2017.nsmallest(10,'generosity'),
            ax=axes[1,0],palette=None)
sns.barplot(x='freedom',y='country',
            data=df2017.nsmallest(10,'freedom'),
            ax=axes[1,1],palette=None)
sns.barplot(x='government_trust',y='country',
            data=df2017.nsmallest(10,'government_trust'),
            ax=axes[2,0],palette=None)
sns.barplot(x='family',y='country',
            data=df2017.nsmallest(10,'family'),
            ax=axes[2,1],palette=None)
```

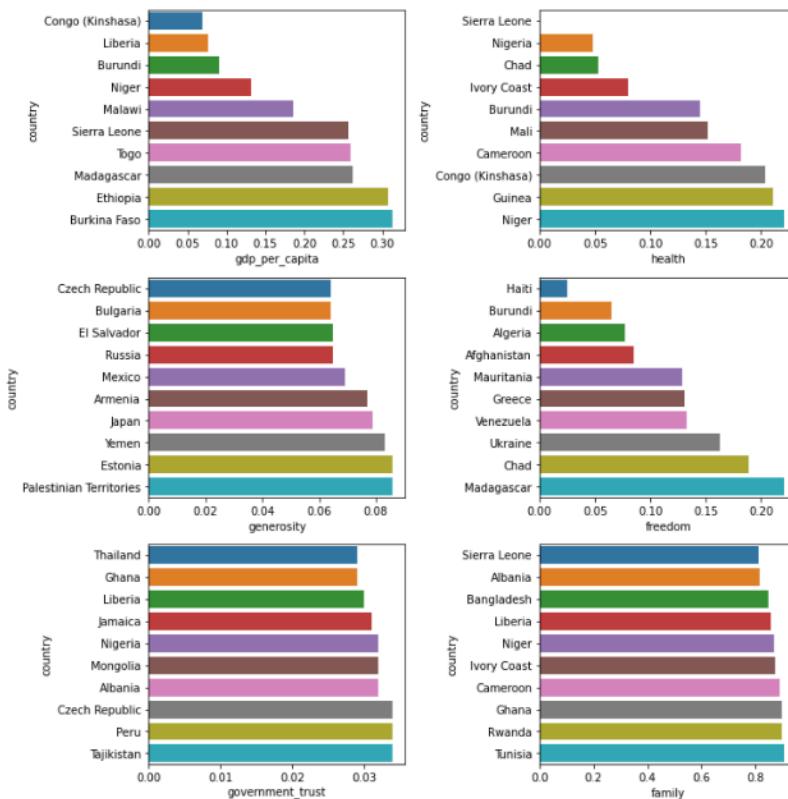
Out[75]: <AxesSubplot:xlabel='family', ylabel='country'>



2018

```
In [76]: fig, axes = plt.subplots(nrows=3, ncols=2, constrained_layout=True, figsize=(10,10))
sns.barplot(x='gdp_per_capita',y='country',
            data=df2018.nsmallest(10,'gdp_per_capita'),
            ax=axes[0,0],palette=None)
sns.barplot(x='health',y='country',
            data=df2018.nsmallest(10,'health'),
            ax=axes[0,1],palette=None)
sns.barplot(x='generosity' ,y='country',
            data=df2018.nsmallest(10,'generosity'),
            ax=axes[1,0],palette=None)
sns.barplot(x='freedom' ,y='country',
            data=df2018.nsmallest(10,'freedom'),
            ax=axes[1,1],palette=None)
sns.barplot(x='government_trust',y='country',
            data=df2018.nsmallest(10,'government_trust'),
            ax=axes[2,0],palette=None)
```

Out[76]: <AxesSubplot:xlabel='family', ylabel='country'>



2019

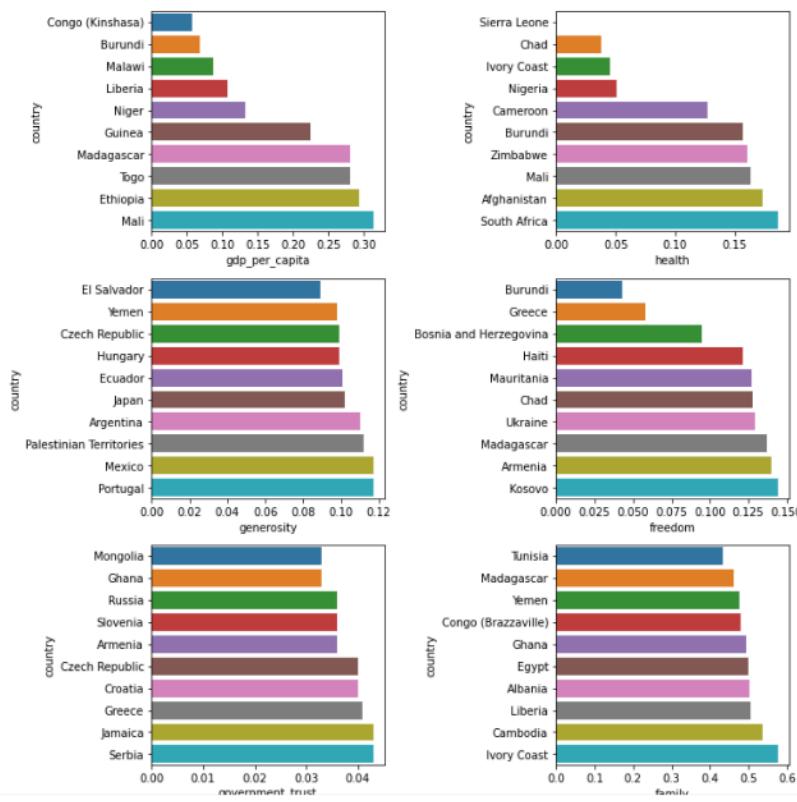
```
In [77]: fig, axes = plt.subplots(nrows=3, ncols=2, constrained_layout=True, figsize=(10,10))
sns.barplot(x='gdp_per_capita',y='country',
            data=df2019.nsmallest(10,'gdp_per_capita'),
            ax=axes[0,0],palette=None)
sns.barplot(x='health',y='country',
            data=df2019.nsmallest(10,'health'),
            ax=axes[0,1],palette=None)
sns.barplot(x='generosity' ,y='country',
            data=df2019.nsmallest(10,'generosity'),
            ax=axes[1,0],palette=None)
sns.barplot(x='freedom' ,y='country',
            data=df2019.nsmallest(10,'freedom'),
            ax=axes[1,1],palette=None)
sns.barplot(x='government_trust',y='country',
            data=df2019.nsmallest(10,'government_trust'),
            ax=axes[2,0],palette=None)
```

```

sns.barplot(x='generosity', y='country',
            data=df2019.nsmallest(10,'generosity'),
            ax=axes[1,0],palette=None)
sns.barplot(x='freedom', y='country',
            data=df2019.nsmallest(10,'freedom'),
            ax=axes[1,1],palette=None)
sns.barplot(x='government_trust', y='country',
            data=df2019.nsmallest(10,'government_trust'),
            ax=axes[2,0],palette=None)
sns.barplot(x='family', y='country',
            data=df2019.nsmallest(10,'family'),
            ax=axes[2,1],palette=None)

```

Out[77]: <AxesSubplot:xlabel='family', ylabel='country'>



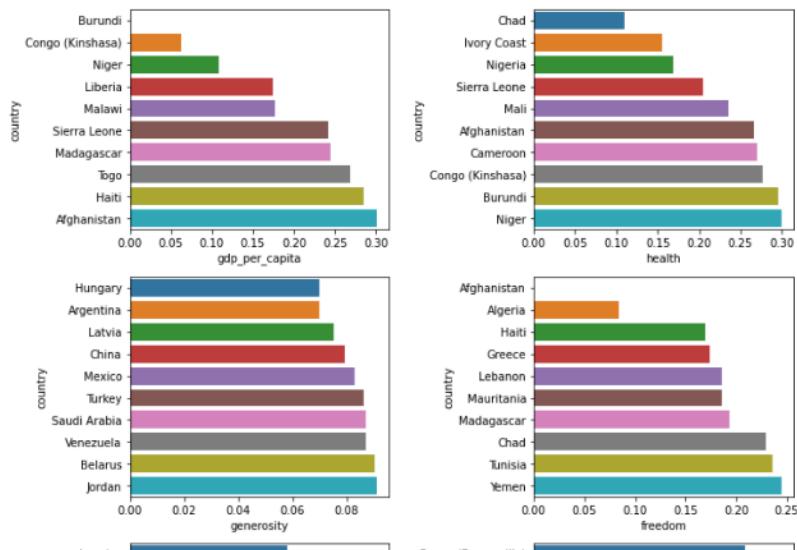
2020

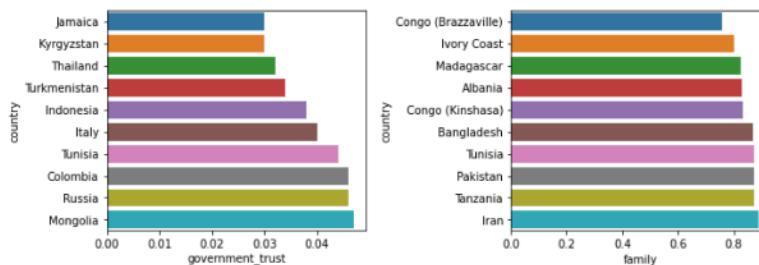
```

In [78]: fig, axes = plt.subplots(nrows=3, ncols=2, constrained_layout=True, figsize=(10,10))
sns.barplot(x='gdp_per_capita',y='country',
            data=df2020.nsmallest(10,'gdp_per_capita'),
            ax=axes[0,0],palette=None)
sns.barplot(x='health',y='country',
            data=df2020.nsmallest(10,'health'),
            ax=axes[0,1],palette=None)
sns.barplot(x='generosity',y='country',
            data=df2020.nsmallest(10,'generosity'),
            ax=axes[1,0],palette=None)
sns.barplot(x='freedom',y='country',
            data=df2020.nsmallest(10,'freedom'),
            ax=axes[1,1],palette=None)
sns.barplot(x='government_trust',y='country',
            data=df2020.nsmallest(10,'government_trust'),
            ax=axes[2,0],palette=None)
sns.barplot(x='family',y='country',
            data=df2020.nsmallest(10,'family'),
            ax=axes[2,1],palette=None)

```

Out[78]: <AxesSubplot:xlabel='family', ylabel='country'>





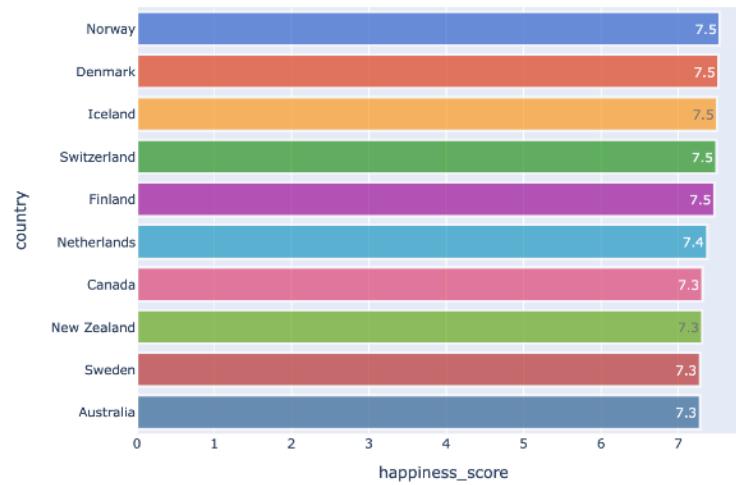
Top 10 countries based on Happiness Score

Top 10 Happiest Countries

```
In [79]: fig_1 = px.bar(data_frame = df2015.nlargest(10,"happiness_score"),
                     y="country",
                     x="happiness_score",
                     orientation='h',
                     color="country",
                     text="happiness_score",
                     color_discrete_sequence=px.colors.qualitative.G10)

fig_1.update_traces(texttemplate='%{text:.2s}',
                     textposition='inside',
                     marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5,
                     opacity=0.7)
fig_1.update_layout(width=750,
                     showlegend=False,
                     title="Top 10 Happiest Countries (2015)",
)
fig_1.show()
```

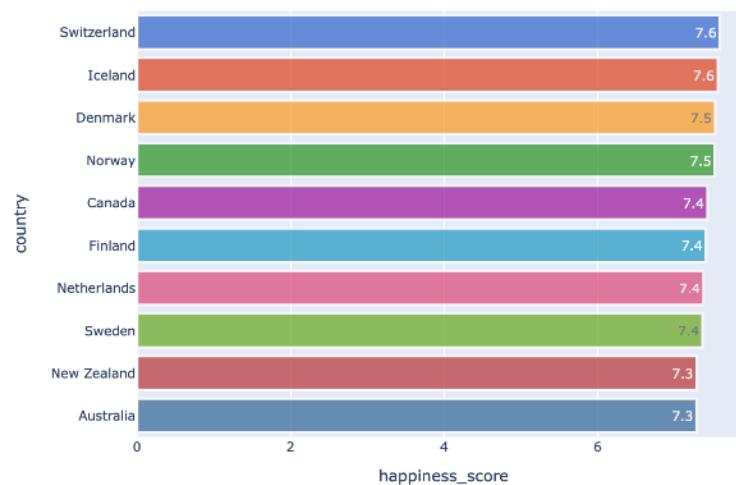
Top 10 Happiest Countries (2015)



```
In [80]: fig_1 = px.bar(data_frame = df2016.nlargest(10,"happiness_score"),
                     y="country",
                     x="happiness_score",
                     orientation='h',
                     color="country",
                     text="happiness_score",
                     color_discrete_sequence=px.colors.qualitative.G10)

fig_1.update_traces(texttemplate='%{text:.2s}',
                     textposition='inside',
                     marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5,
                     opacity=0.7)
fig_1.update_layout(width=750,
                     showlegend=False,
                     title="Top 10 Happiest Countries (2016)",
)
fig_1.show()
```

Top 10 Happiest Countries (2016)



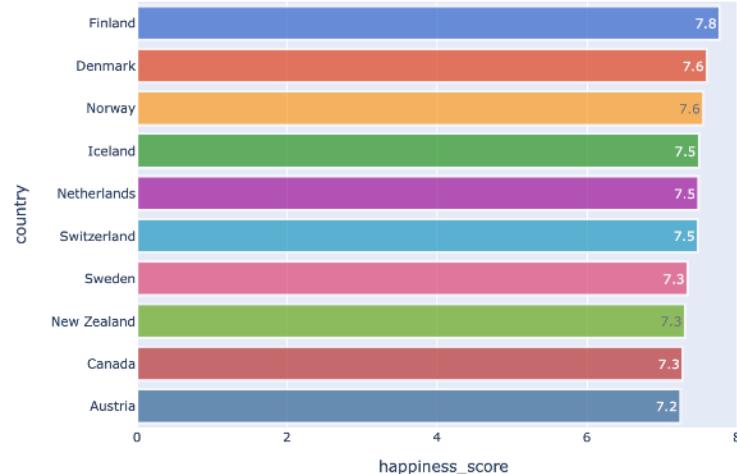
```

        color_discrete_sequence=px.colors.qualitative.G10)

fig_1.update_traces(texttemplate='%{text:.2s}',
                     textposition='inside',
                     marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5,
                     opacity=0.7)
fig_1.update_layout(width=750,
                     showlegend=False,
                     title="Top 10 Happiest Countries (2017)",
)
fig_1.show()

```

Top 10 Happiest Countries (2017)



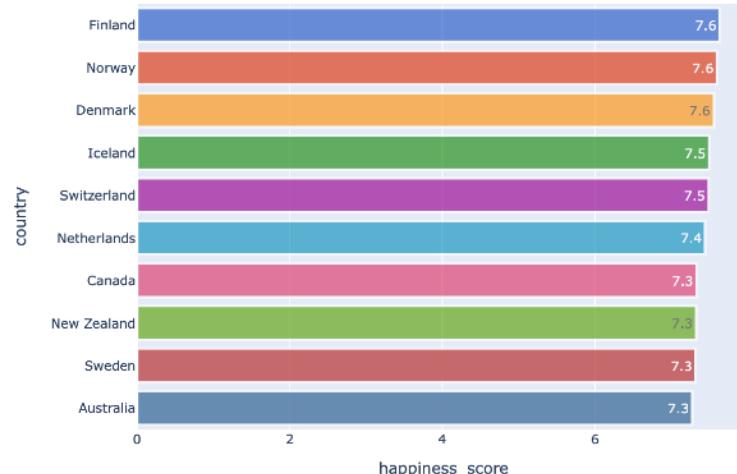
```

In [82]: fig_1 = px.bar(data_frame = df2018.nlargest(10,"happiness_score"),
                     y="country",
                     x="happiness_score",
                     orientation='h',
                     color="country",
                     text="happiness_score",
                     color_discrete_sequence=px.colors.qualitative.G10)

fig_1.update_traces(texttemplate='%{text:.2s}',
                     textposition='inside',
                     marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5,
                     opacity=0.7)
fig_1.update_layout(width=750,
                     showlegend=False,
                     title="Top 10 Happiest Countries (2018)",
)
fig_1.show()

```

Top 10 Happiest Countries (2018)



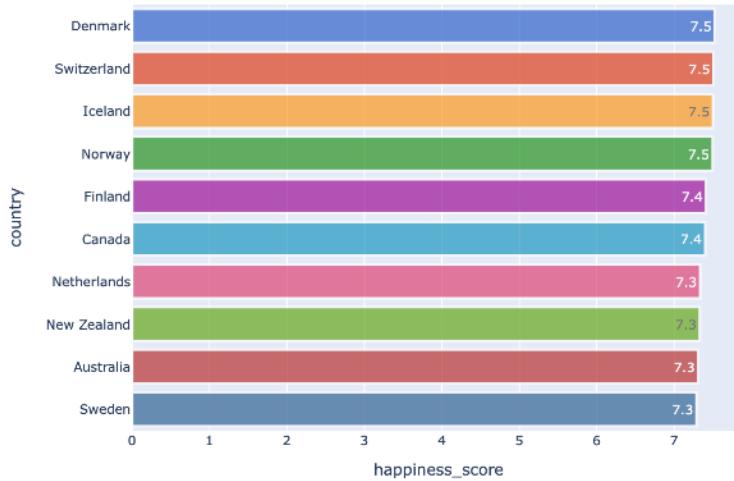
```

In [83]: fig_1 = px.bar(data_frame = df2019.nlargest(10,"happiness_score"),
                     y="country",
                     x="happiness_score",
                     orientation='h',
                     color="country",
                     text="happiness_score",
                     color_discrete_sequence=px.colors.qualitative.G10)

fig_1.update_traces(texttemplate='%{text:.2s}',
                     textposition='inside',
                     marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5,
                     opacity=0.7)
fig_1.update_layout(width=750,
                     showlegend=False,
                     title="Top 10 Happiest Countries (2019)",
)
fig_1.show()

```

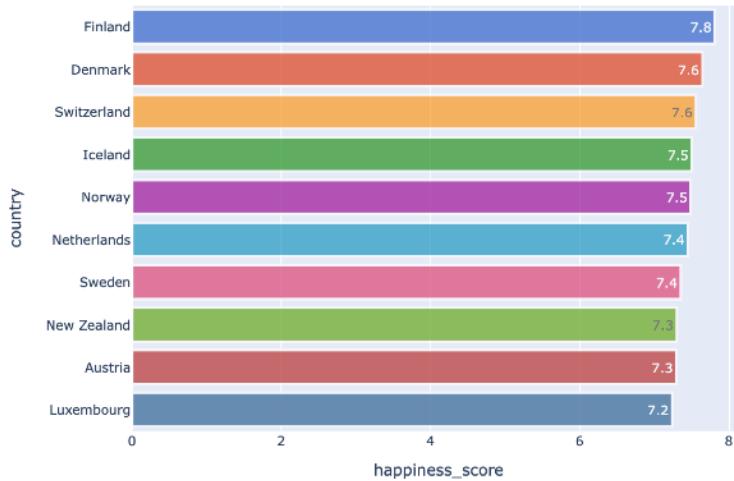
Top 10 Happiest Countries (2019)



```
In [84]: fig_1 = px.bar(data_frame = df2020.nlargest(10,"happiness_score"),
                    y="country",
                    x="happiness_score",
                    orientation="h",
                    color="country",
                    text="happiness_score",
                    color_discrete_sequence=px.colors.qualitative.G10)

fig_1.update_traces(texttemplate='%{text:.2s}',
                     textposition='inside',
                     marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5,
                     opacity=0.7)
fig_1.update_layout(width=750,
                     showlegend=False,
                     title="Top 10 Happiest Countries (2020)",
)
fig_1.show()
```

Top 10 Happiest Countries (2020)

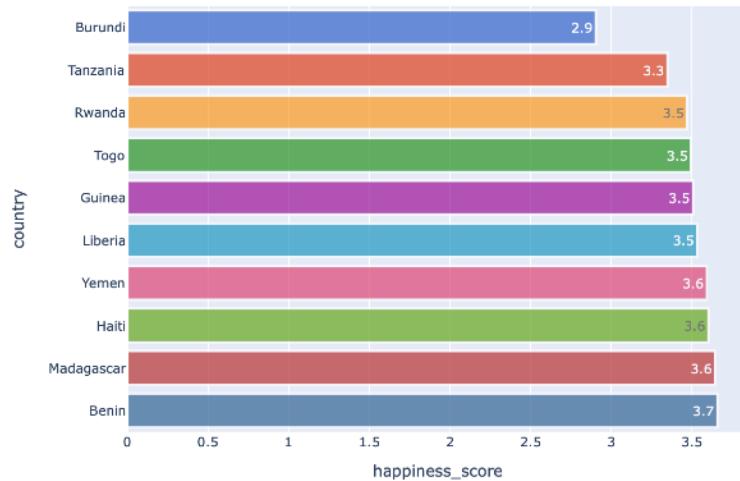


Top 10 Unhappiest Countries

```
In [85]: fig_1 = px.bar(data_frame = df2015.nsmallest(10,"happiness_score"),
                    y="country",
                    x="happiness_score",
                    orientation="h",
                    color="country",
                    text="happiness_score",
                    color_discrete_sequence=px.colors.qualitative.G10)

fig_1.update_traces(texttemplate='%{text:.2s}',
                     textposition='inside',
                     marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5,
                     opacity=0.7)
fig_1.update_layout(width=750,
                     showlegend=False,
                     title="Top 10 Unhappiest Countries (2015)",
)
fig_1.show()
```

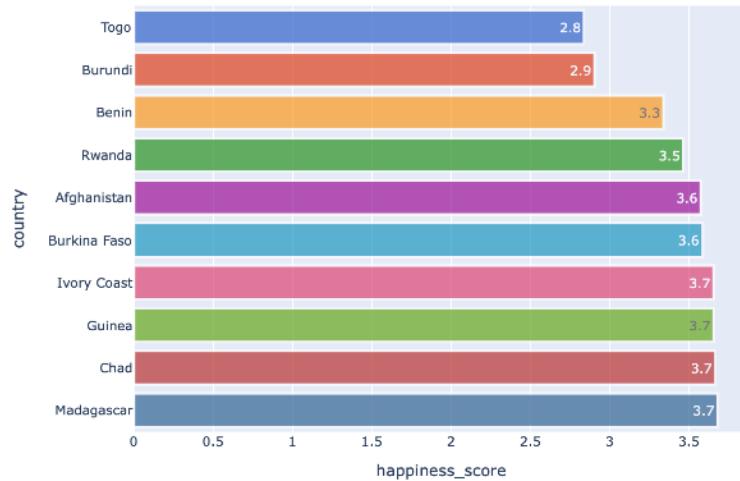
Top 10 Unhappiest Countries (2015)



```
In [86]: fig_1 = px.bar(data_frame = df2016.nsmallest(10,"happiness_score"),
                     y="country",
                     x="happiness_score",
                     orientation='h',
                     color="country",
                     text="happiness_score",
                     color_discrete_sequence=px.colors.qualitative.G10)

fig_1.update_traces(texttemplate='%{text:.2s}',
                     textposition='inside',
                     marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5,
                     opacity=0.7)
fig_1.update_layout(width=750,
                     showlegend=False,
                     title="Top 10 Unhappiest Countries (2016)",
)
fig_1.show()
```

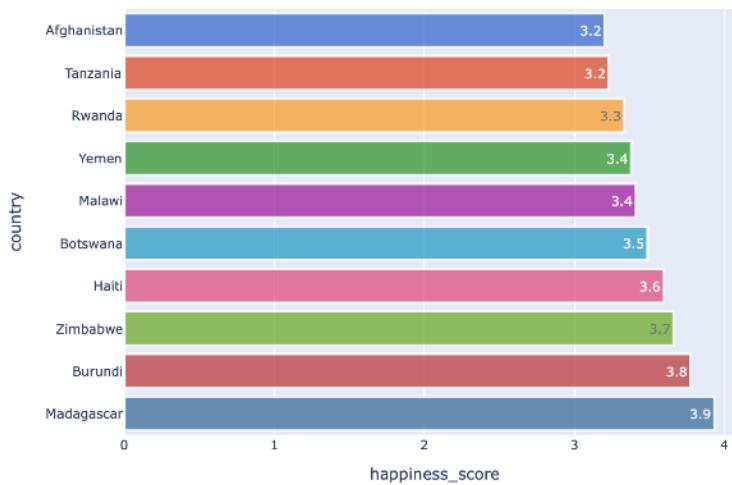
Top 10 Unhappiest Countries (2016)



```
In [87]: fig_1 = px.bar(data_frame = df2017.nsmallest(10,"happiness_score"),
                     y="country",
                     x="happiness_score",
                     orientation='h',
                     color="country",
                     text="happiness_score",
                     color_discrete_sequence=px.colors.qualitative.G10)

fig_1.update_traces(texttemplate='%{text:.2s}',
                     textposition='inside',
                     marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5,
                     opacity=0.7)
fig_1.update_layout(width=750,
                     showlegend=False,
                     title="Top 10 Unhappiest Countries (2017)",
)
fig_1.show()
```

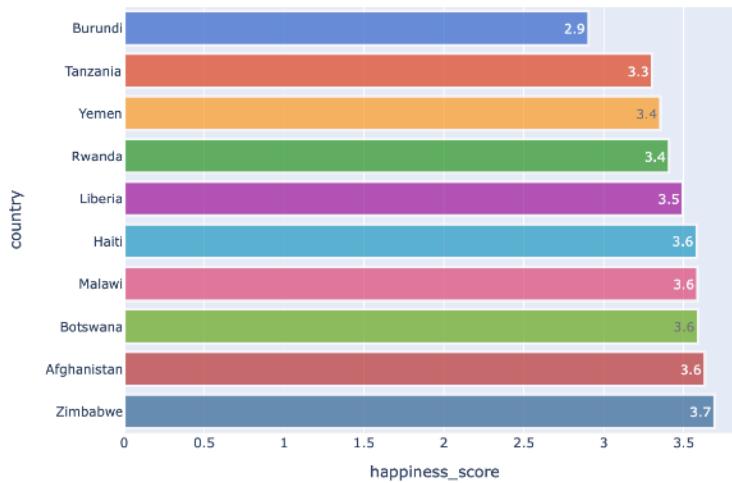
Top 10 Unhappiest Countries (2017)



```
In [88]: fig_1 = px.bar(data_frame = df2018.nsmallest(10,"happiness_score"),
                    y="country",
                    x="happiness_score",
                    orientation='h',
                    color="country",
                    text="happiness_score",
                    color_discrete_sequence=px.colors.qualitative.G10)

fig_1.update_traces(texttemplate='%{text:.2s}',
                     textposition='inside',
                     marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5,
                     opacity=0.7)
fig_1.update_layout(width=750,
                     showlegend=False,
                     title="Top 10 Unhappiest Countries (2018)",
)
fig_1.show()
```

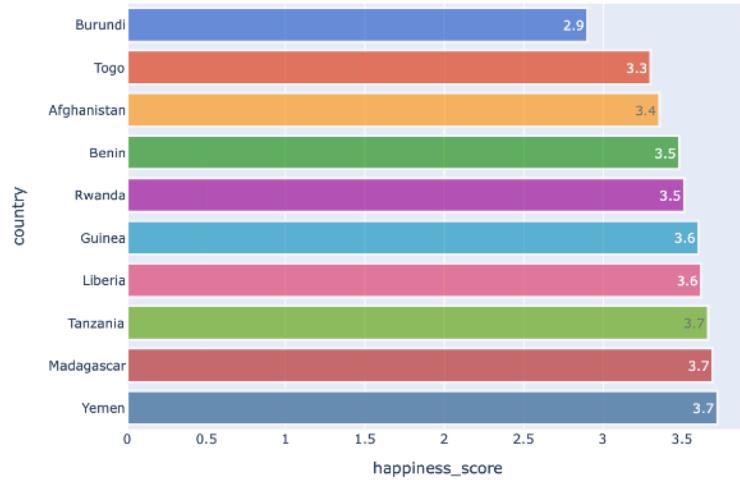
Top 10 Unhappiest Countries (2018)



```
In [89]: fig_1 = px.bar(data_frame = df2019.nsmallest(10,"happiness_score"),
                    y="country",
                    x="happiness_score",
                    orientation='h',
                    color="country",
                    text="happiness_score",
                    color_discrete_sequence=px.colors.qualitative.G10)

fig_1.update_traces(texttemplate='%{text:.2s}',
                     textposition='inside',
                     marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5,
                     opacity=0.7)
fig_1.update_layout(width=750,
                     showlegend=False,
                     title="Top 10 Unhappiest Countries (2019)",
)
fig_1.show()
```

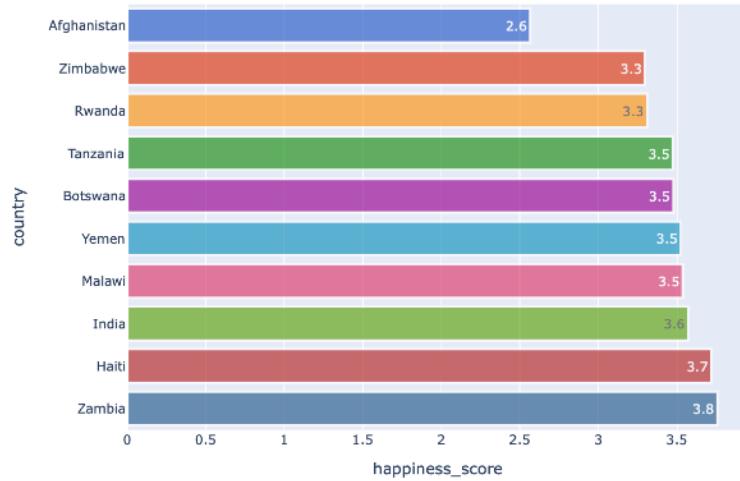
Top 10 Unhappiest Countries (2019)



```
In [90]: fig_1 = px.bar(data_frame = df2020.nsmallest(10,"happiness_score"),
                     y="country",
                     x="happiness_score",
                     orientation='h',
                     color="country",
                     text="happiness_score",
                     color_discrete_sequence=px.colors.qualitative.G10)

fig_1.update_traces(texttemplate='%{text:.2s}',
                     textposition='inside',
                     marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5,
                     opacity=0.7)
fig_1.update_layout(width=750,
                     showlegend=False,
                     title="Top 10 Unhappiest Countries (2020)",
)
fig_1.show()
```

Top 10 Unhappiest Countries (2020)

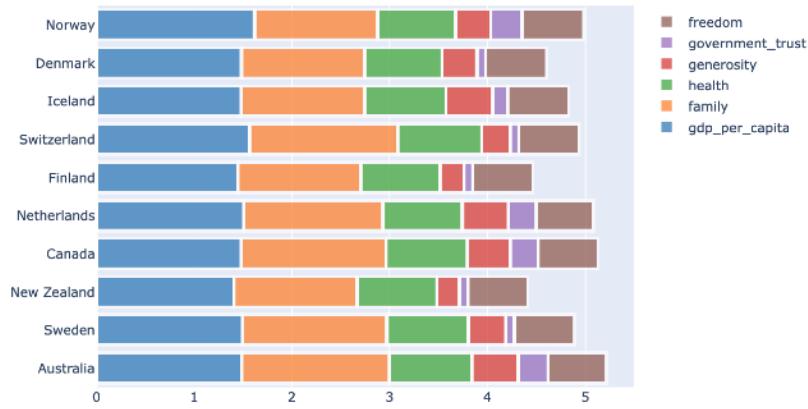


Relationship between Happiness Score and features

Happiest Countries

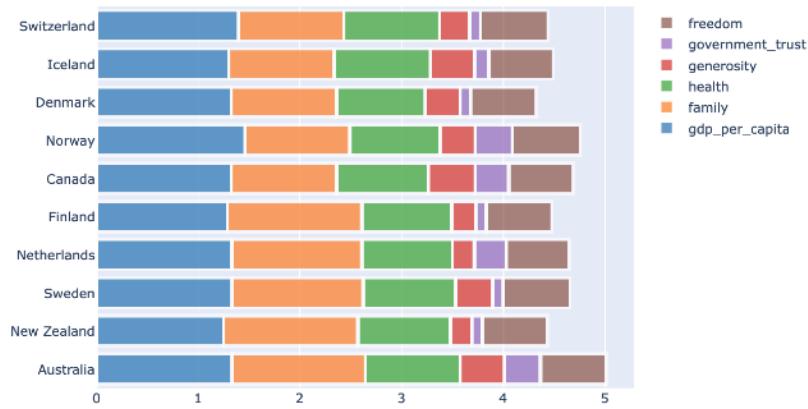
```
In [92]: features = ["gdp_per_capita", "family", "health", "generosity", "government_trust", "freedom"]
my_list = []
for f,c in zip(features, px.colors.qualitative.D3):
    my_list.append(go.Bar(y=df2015.nlargest(10,"happiness_score")["country"].values,
                          x=df2015.nlargest(10,"happiness_score")[f].values,
                          name=f,
                          marker=dict(color=c),
                          orientation="h"))
fig_9 = go.Figure(data=my_list)
fig_9.update_traces(marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5, opacity=0.7)
fig_9.update_layout(
    width=900,
    barmode='stack',
    yaxis=dict(autorange="reversed"),
)
fig_9.update_layout(width=750,
                    yaxis=dict(autorange="reversed"),
                    title="Contribution to Happiness Score for Top 10 Happiest Countries (2015)")
fig_9.show()
```

Contribution to Happiness Score for Top 10 Happiest Countries (2015)



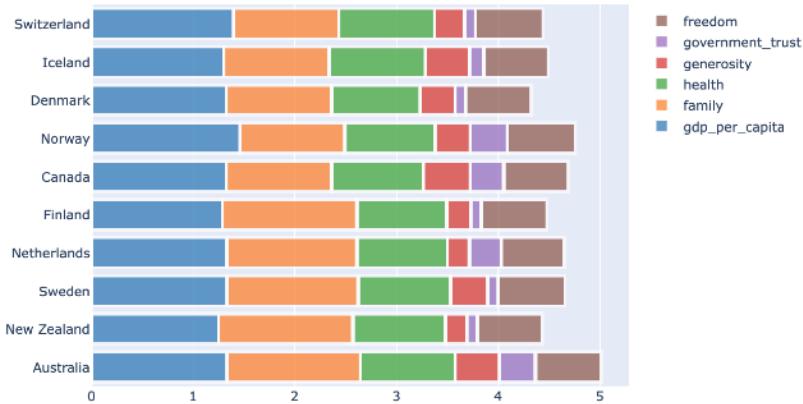
```
In [93]: features = ["gdp_per_capita", "family", "health", "generosity", "government_trust", "freedom"]
my_list = []
for f,c in zip(features, px.colors.qualitative.D3):
    my_list.append(go.Bar(y=df2016.nlargest(10,"happiness_score")["country"].values,
                          x=df2016.nlargest(10,"happiness_score")[f].values,
                          name=f,
                          marker=dict(color=c),
                          orientation="h"))
fig_9 = go.Figure(data=my_list)
fig_9.update_traces(marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5, opacity=0.7)
fig_9.update_layout(
    width=900,
    barmode='stack',
    yaxis=dict(autorange="reversed"),
)
fig_9.update_layout(width=750,
                    yaxis=dict(autorange="reversed"),
                    title="Contribution to Happiness Score for Top 10 Happiest Countries (2016)")
fig_9.show()
```

Contribution to Happiness Score for Top 10 Happiest Countries (2016)



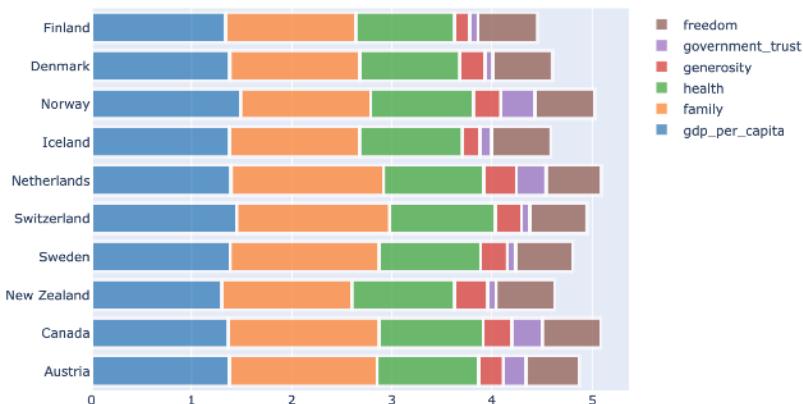
```
In [93]: features = ["gdp_per_capita", "family", "health", "generosity", "government_trust", "freedom"]
my_list = []
for f,c in zip(features, px.colors.qualitative.D3):
    my_list.append(go.Bar(y=df2016.nlargest(10,"happiness_score")["country"].values,
                          x=df2016.nlargest(10,"happiness_score")[f].values,
                          name=f,
                          marker=dict(color=c),
                          orientation="h"))
fig_9 = go.Figure(data=my_list)
fig_9.update_traces(marker_line_color='rgb(255,255,255)', marker_line_width=2.5, opacity=0.7)
fig_9.update_layout(
    width=900,
    barmode='stack',
    yaxis=dict(autorange="reversed"),
)
fig_9.update_layout(width=750,
                    yaxis=dict(autorange="reversed"),
                    title="Contribution to Happiness Score for Top 10 Happiest Countries (2016)")
fig_9.show()
```

Contribution to Happiness Score for Top 10 Happiest Countries (2016)



```
In [94]: features = ["gdp_per_capita", "family", "health", "generosity", "government_trust", "freedom"]
my_list = []
for f,c in zip(features, px.colors.qualitative.D3):
    my_list.append(go.Bar(y=df2017.nlargest(10,"happiness_score")["country"].values,
                          x=df2017.nlargest(10,"happiness_score")[f].values,
                          name=f,
                          marker=dict(color=c),
                          orientation="h"))
fig_9 = go.Figure(data=my_list)
fig_9.update_traces(marker_line_color='rgb(255,255,255)', marker_line_width=2.5, opacity=0.7)
fig_9.update_layout(
    width=900,
    barmode='stack',
    yaxis=dict(autorange="reversed"),
)
fig_9.update_layout(width=750,
                    yaxis=dict(autorange="reversed"),
                    title="Contribution to Happiness Score for Top 10 Happiest Countries (2017)")
fig_9.show()
```

Contribution to Happiness Score for Top 10 Happiest Countries (2017)



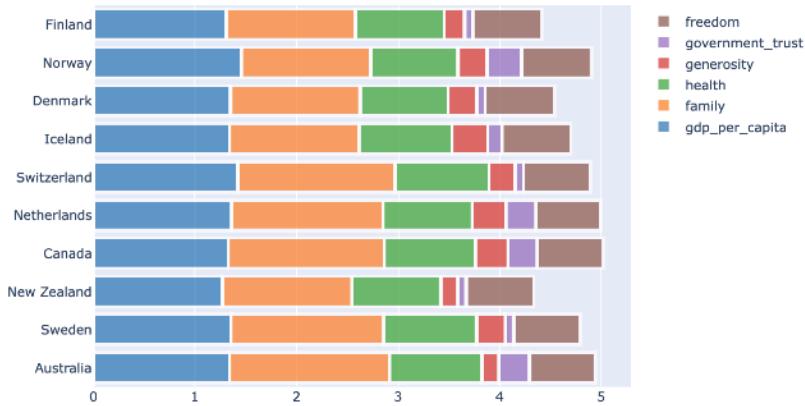
```
In [95]: features = ["gdp_per_capita", "family", "health", "generosity", "government_trust", "freedom"]
my_list = []
for f,c in zip(features, px.colors.qualitative.D3):
    my_list.append(go.Bar(y=df2018.nlargest(10,"happiness_score")["country"].values,
                          x=df2018.nlargest(10,"happiness_score")[f].values,
                          name=f,
```

```

        marker=dict(color=c),
        orientation="h"))
fig_9 = go.Figure(data=my_list)
fig_9.update_traces(marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5, opacity=0.7)
fig_9.update_layout(
    width=900,
    barmode='stack',
    yaxis=dict(autorange="reversed"),
)
fig_9.update_layout(width=750,
                    yaxis=dict(autorange="reversed"),
                    title="Contribution to Happiness Score for Top 10 Happiest Countries (2018)")
fig_9.show()

```

Contribution to Happiness Score for Top 10 Happiest Countries (2018)

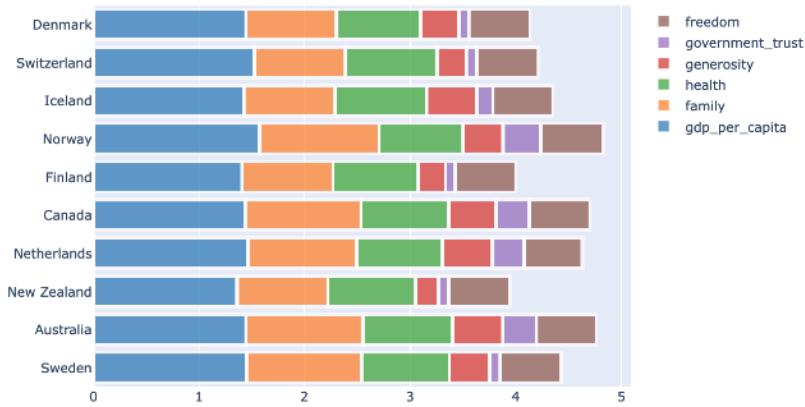


```

In [96]: features = ["gdp_per_capita", "family", "health", "generosity", "government_trust", "freedom"]
my_list = []
for f,c in zip(features, px.colors.qualitative.D3):
    my_list.append(go.Bar(y=df2019.nlargest(10,"happiness_score")["country"].values,
                          x=df2019.nlargest(10,"happiness_score")[f].values,
                          name=f,
                          marker=dict(color=c),
                          orientation="h"))

```

Contribution to Happiness Score for Top 10 Happiest Countries (2019)

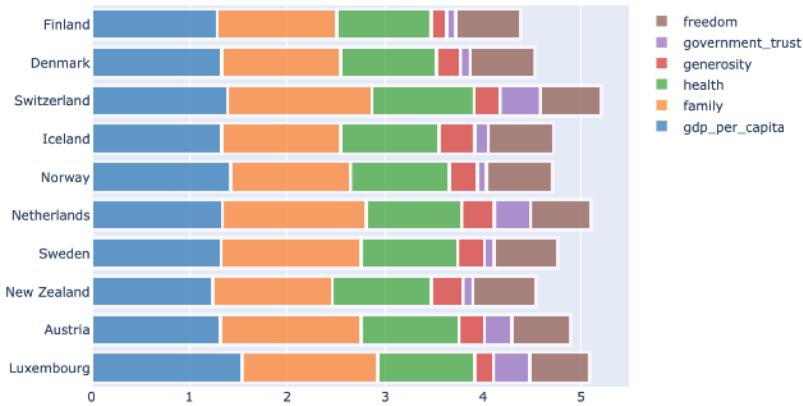


```

In [97]: features = ["gdp_per_capita", "family", "health", "generosity", "government_trust", "freedom"]
my_list = []
for f,c in zip(features, px.colors.qualitative.D3):
    my_list.append(go.Bar(y=df2020.nlargest(10,"happiness_score")["country"].values,
                          x=df2020.nlargest(10,"happiness_score")[f].values,
                          name=f,
                          marker=dict(color=c),
                          orientation="h"))
fig_9 = go.Figure(data=my_list)
fig_9.update_traces(marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5, opacity=0.7)
fig_9.update_layout(
    width=900,
    barmode='stack',
    yaxis=dict(autorange="reversed"),
)
fig_9.update_layout(width=750,
                    yaxis=dict(autorange="reversed"),
                    title="Contribution to Happiness Score for Top 10 Happiest Countries (2020)")
fig_9.show()

```

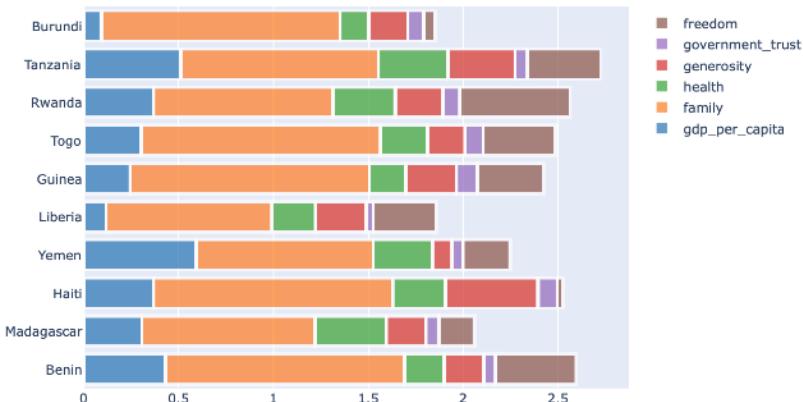
Contribution to Happiness Score for Top 10 Happiest Countries (2020)



Unhappiest Countries

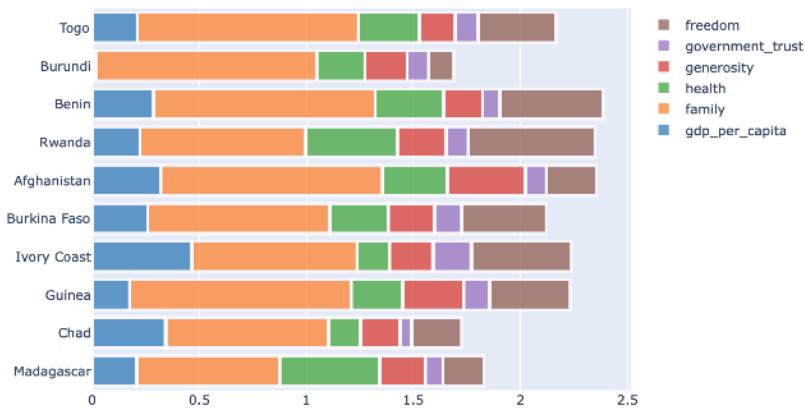
```
In [98]: features = ["gdp_per_capita", "family", "health", "generosity", "government_trust", "freedom"]
my_list = []
for f,c in zip(features, px.colors.qualitative.D3):
    my_list.append(go.Bar(y=df2015.nsmallest(10,"happiness_score")["country"].values,
                          x=df2015.nsmallest(10,"happiness_score")[f].values,
                          name=f,
                          marker=dict(color=c),
                          orientation="h"))
fig_9 = go.Figure(data=my_list)
fig_9.update_traces(marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5, opacity=0.7)
fig_9.update_layout(
    width=900,
    barmode='stack',
    yaxis=dict(autorange="reversed"),
)
fig_9.update_layout(width=750,
                     yaxis=dict(autorange="reversed"),
                     title="Contribution to Happiness Score for Top 10 Unhappiest Countries (2015)")
fig_9.show()
```

Contribution to Happiness Score for Top 10 Unhappiest Countries (2015)



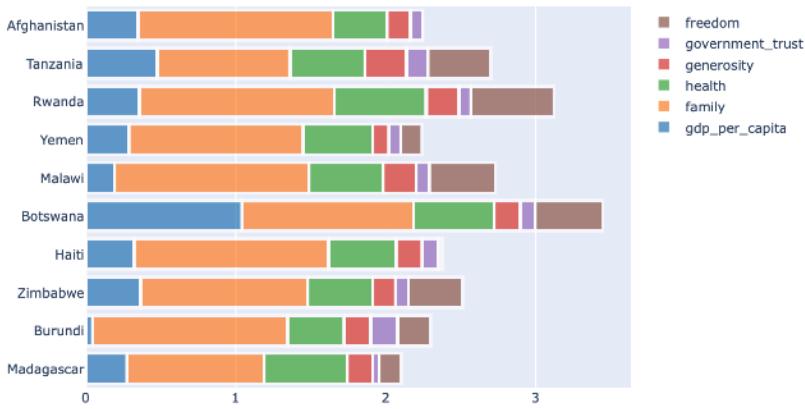
```
In [99]: features = ["gdp_per_capita", "family", "health", "generosity", "government_trust", "freedom"]
my_list = []
for f,c in zip(features, px.colors.qualitative.D3):
    my_list.append(go.Bar(y=df2016.nsmallest(10,"happiness_score")["country"].values,
                          x=df2016.nsmallest(10,"happiness_score")[f].values,
                          name=f,
                          marker=dict(color=c),
                          orientation="h"))
fig_9 = go.Figure(data=my_list)
fig_9.update_traces(marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5, opacity=0.7)
fig_9.update_layout(
    width=900,
    barmode='stack',
    yaxis=dict(autorange="reversed"),
)
fig_9.update_layout(width=750,
                     yaxis=dict(autorange="reversed"),
                     title="Contribution to Happiness Score for Top 10 Unhappiest Countries (2016)")
fig_9.show()
```

Contribution to Happiness Score for Top 10 Unhappiest Countries (2016)



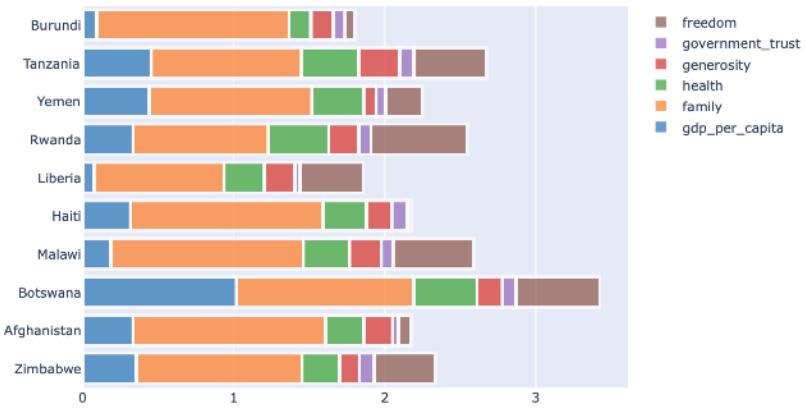
```
In [100]: features = ["gdp_per_capita", "family", "health", "generosity", "government_trust", "freedom"]
my_list = []
for f,c in zip(features, px.colors.qualitative.D3):
    my_list.append(go.Bar(y=df2017.nsmallest(10,"happiness_score")["country"].values,
                          x=df2017.nsmallest(10,"happiness_score")[f].values,
                          name=f,
                          marker=dict(color=c),
                          orientation="h"))
fig_9 = go.Figure(data=my_list)
fig_9.update_traces(marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5, opacity=0.7)
fig_9.update_layout(
    width=900,
    barmode='stack',
    yaxis=dict(autorange="reversed"),
)
fig_9.update_layout(width=750,
                    yaxis=dict(autorange="reversed"),
                    title="Contribution to Happiness Score for Top 10 Unhappiest Countries (2017)")
fig_9.show()
```

Contribution to Happiness Score for Top 10 Unhappiest Countries (2017)



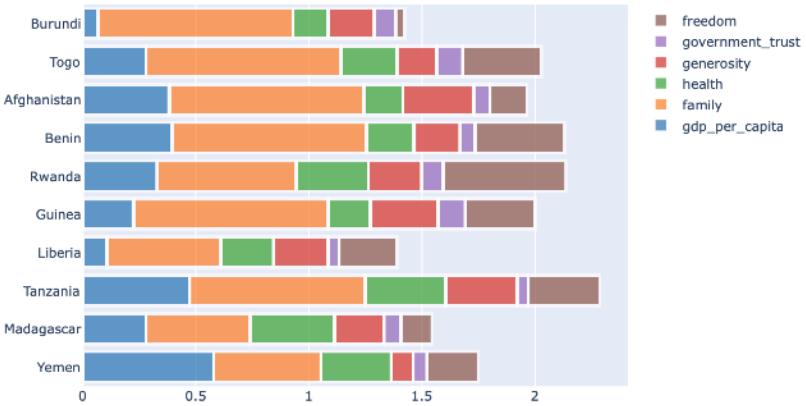
```
In [101]: features = ["gdp_per_capita", "family", "health", "generosity", "government_trust", "freedom"]
my_list = []
for f,c in zip(features, px.colors.qualitative.D3):
    my_list.append(go.Bar(y=df2018.nsmallest(10,"happiness_score")["country"].values,
                          x=df2018.nsmallest(10,"happiness_score")[f].values,
                          name=f,
                          marker=dict(color=c),
                          orientation="h"))
fig_9 = go.Figure(data=my_list)
fig_9.update_traces(marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5, opacity=0.7)
fig_9.update_layout(
    width=900,
    barmode='stack',
    yaxis=dict(autorange="reversed"),
)
fig_9.update_layout(width=750,
                    yaxis=dict(autorange="reversed"),
                    title="Contribution to Happiness Score for Top 10 Unhappiest Countries (2018)")
fig_9.show()
```

Contribution to Happiness Score for Top 10 Unhappiest Countries (2018)



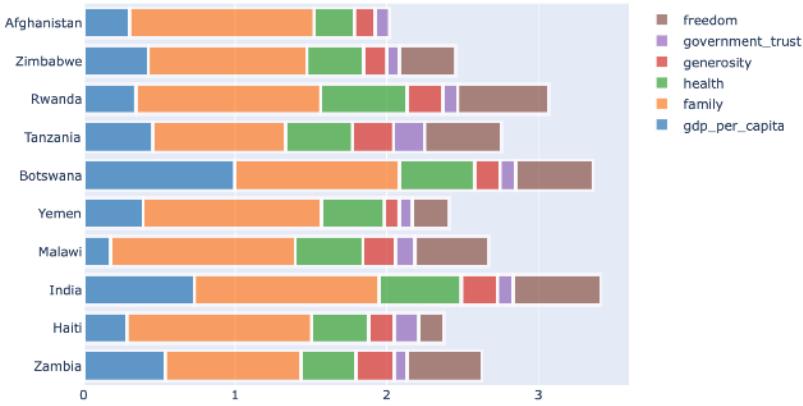
```
In [102]: features = ["gdp_per_capita", "family", "health", "generosity", "government_trust", "freedom"]
my_list = []
for f,c in zip(features, px.colors.qualitative.D3):
    my_list.append(go.Bar(y=df2019.nsmallest(10,"happiness_score")["country"].values,
                          x=df2019.nsmallest(10,"happiness_score")[f].values,
                          name=f,
                          marker=dict(color=c),
                          orientation="h"))
fig_9 = go.Figure(data=my_list)
fig_9.update_traces(marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5, opacity=0.7)
fig_9.update_layout(
    width=900,
    barmode='stack',
    yaxis=dict(autorange="reversed"),
)
fig_9.update_layout(width=750,
                    yaxis=dict(autorange="reversed"),
                    title="Contribution to Happiness Score for Top 10 Unhappiest Countries (2019)")
fig_9.show()
```

Contribution to Happiness Score for Top 10 Unhappiest Countries (2019)



```
In [103]: features = ["gdp_per_capita", "family", "health", "generosity", "government_trust", "freedom"]
my_list = []
for f,c in zip(features, px.colors.qualitative.D3):
    my_list.append(go.Bar(y=df2020.nsmallest(10,"happiness_score")["country"].values,
                          x=df2020.nsmallest(10,"happiness_score")[f].values,
                          name=f,
                          marker=dict(color=c),
                          orientation="h"))
fig_9 = go.Figure(data=my_list)
fig_9.update_traces(marker_line_color='rgb(255,255,255)',
                     marker_line_width=2.5, opacity=0.7)
fig_9.update_layout(
    width=900,
    barmode='stack',
    yaxis=dict(autorange="reversed"),
)
fig_9.update_layout(width=750,
                    yaxis=dict(autorange="reversed"),
                    title="Contribution to Happiness Score for Top 10 Unhappiest Countries (2020)")
fig_9.show()
```

Contribution to Happiness Score for Top 10 Unhappiest Countries (2020)



```
In [104]: df2015['Year']='2015'
df2016['Year']='2016'
df2017['Year']='2017'
df2018['Year']='2018'
df2019['Year']='2019'
df2020['Year']='2020'
```

```
In [105]: data1=df2015.filter(['country','gdp_per_capita','happiness_score','Year','continent','family','generosity','governe
data2=df2016.filter(['country','gdp_per_capita','happiness_score','Year','continent','family','generosity','governe
data3=df2017.filter(['country','gdp_per_capita','happiness_score','Year','continent','family','generosity','governe
data4=df2018.filter(['country','gdp_per_capita','happiness_score','Year','continent','family','generosity','governe
data5=df2019.filter(['country','gdp_per_capita','happiness_score','Year','continent','family','generosity','governe
data6=df2020.filter(['country','gdp_per_capita','happiness_score','Year','continent','family','generosity','governe
df_data = data1
df_data=df_data.append([data2,data3,data4,data5,data6])
```

Trends over the years

```
In [111]: df = df_data.query("continent=='Asia'")
fig = px.line(df, x="Year", y="happiness_score", color='country', title='Happiness Score Trend in Asia')
fig.show()

df = df_data.query("continent=='Australia'")
fig = px.line(df, x="Year", y="happiness_score", color='country', title='Happiness Score Trend in Australia')
fig.show()

df = df_data.query("continent=='North America'")
fig = px.line(df, x="Year", y="happiness_score", color='country', title='Happiness Score Trend in North America')
fig.show()

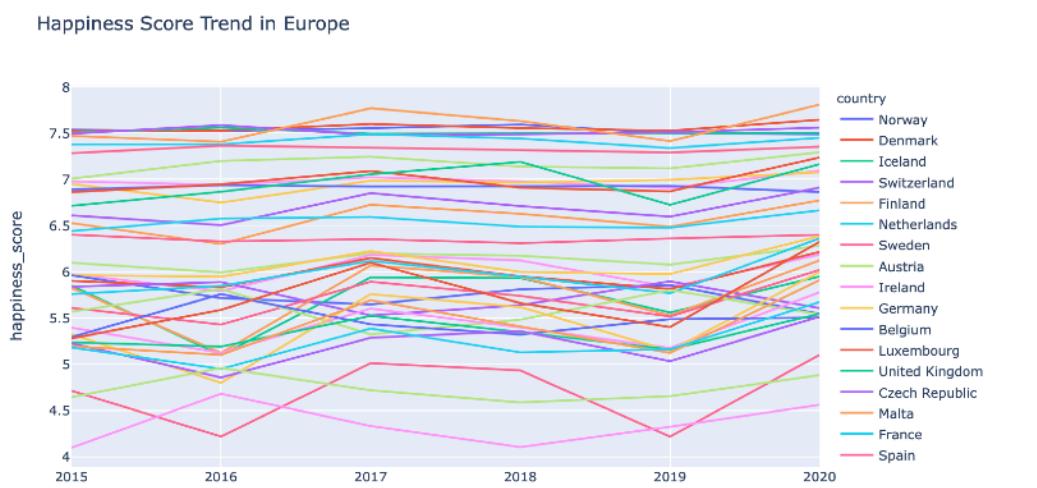
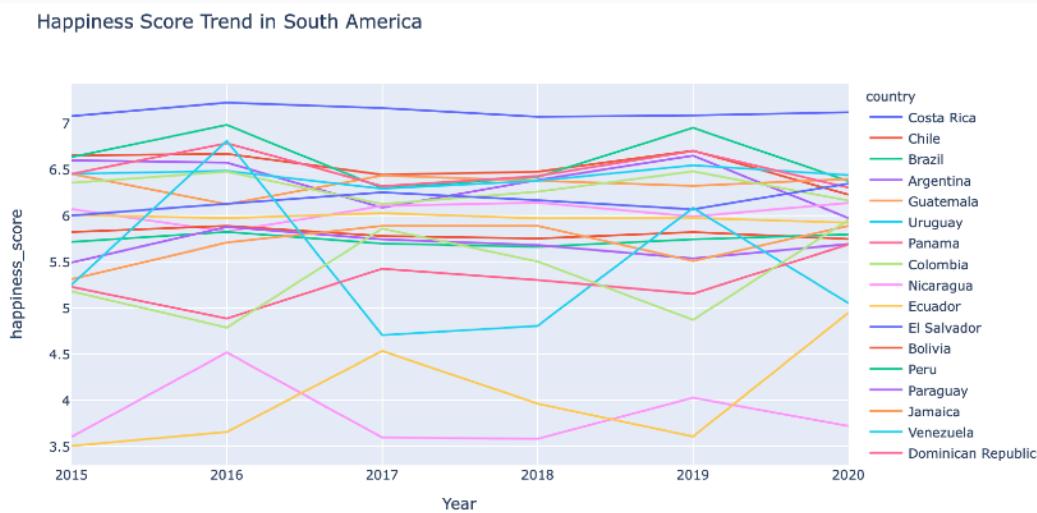
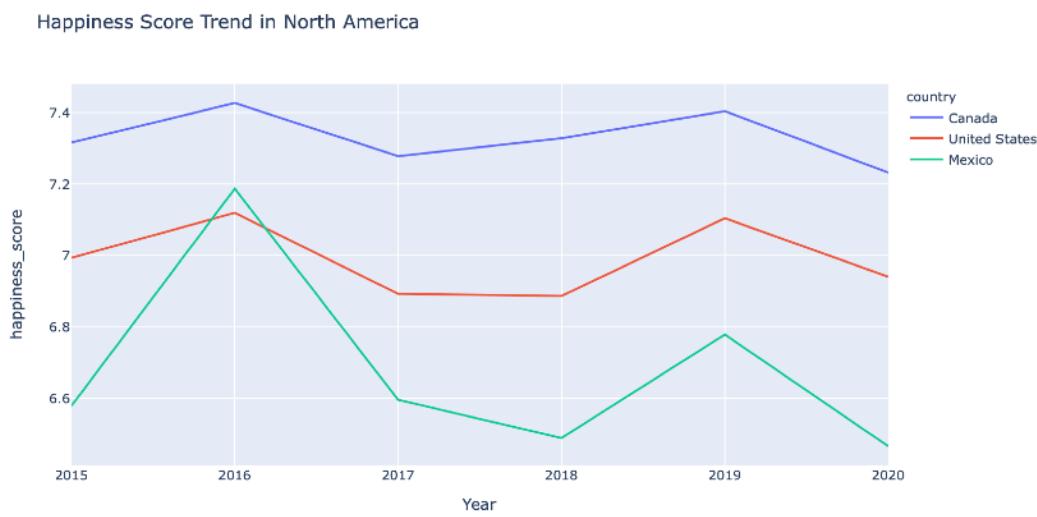
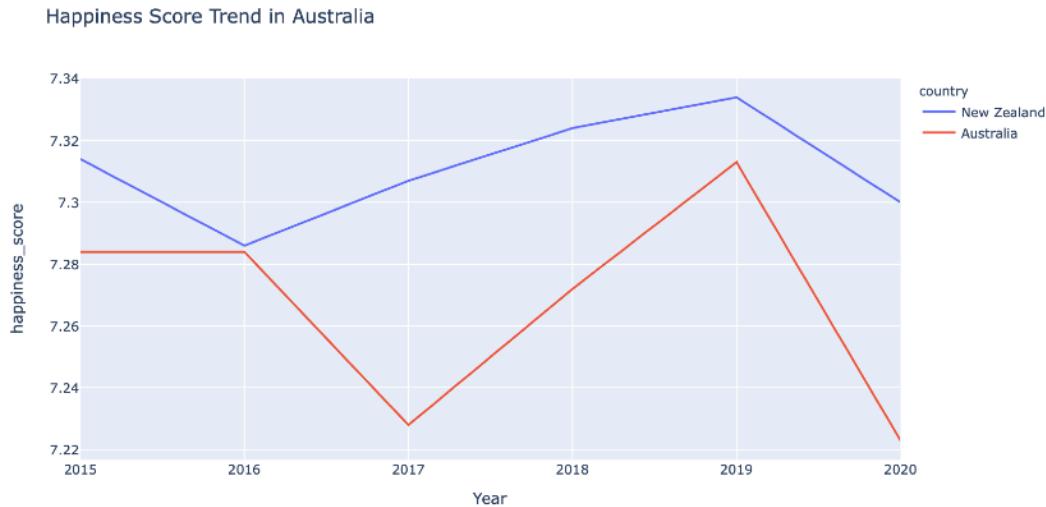
df = df_data.query("continent=='South America'")
fig = px.line(df, x="Year", y="happiness_score", color='country', title='Happiness Score Trend in South America')
fig.show()

df = df_data.query("continent=='Europe'")
fig = px.line(df, x="Year", y="happiness_score", color='country', title='Happiness Score Trend in Europe')
fig.show()

df = df_data.query("continent=='Africa'")
fig = px.line(df, x="Year", y="happiness_score", color='country', title='Happiness Score Trend in Africa')
fig.show()
```

Happiness Score Trend in Asia





Happiness Score Trend in Africa



```
In [108]: df = df_data.query("continent=='Asia'")
fig = px.line(df, x="Year", y="gdp_per_capita", color='country', title='GDP Trend in Asia')
fig.show()

df = df_data.query("continent=='Australia'")
fig = px.line(df, x="Year", y="gdp_per_capita", color='country', title='GDP Trend in Australia')
fig.show()

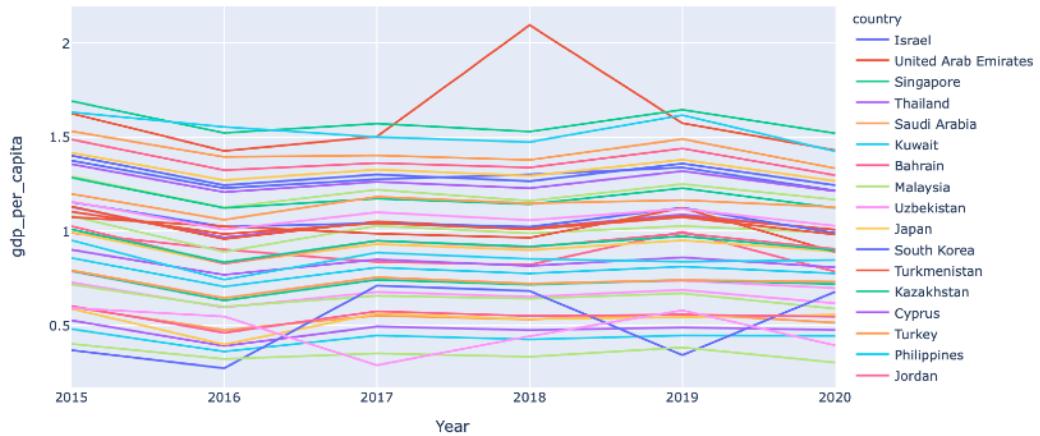
df = df_data.query("continent=='North America'")
fig = px.line(df, x="Year", y="gdp_per_capita", color='country', title='GDP Trend in North America')
fig.show()

df = df_data.query("continent=='South America'")
fig = px.line(df, x="Year", y="gdp_per_capita", color='country', title='GDP Trend in South America')
fig.show()

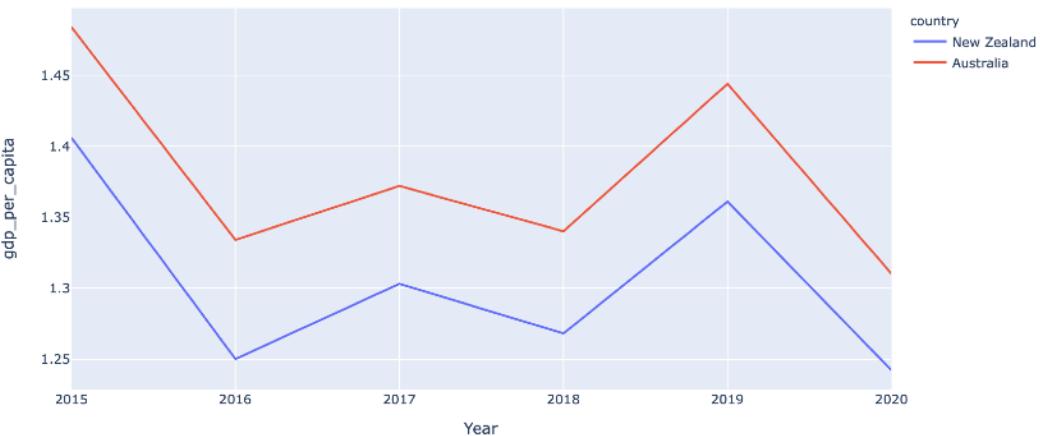
df = df_data.query("continent=='Europe'")
fig = px.line(df, x="Year", y="gdp_per_capita", color='country', title='GDP Trend in Europe')
fig.show()

df = df_data.query("continent=='Africa'")
fig = px.line(df, x="Year", y="gdp_per_capita", color='country', title='GDP Trend in Africa')
fig.show()
```

GDP Trend in Asia



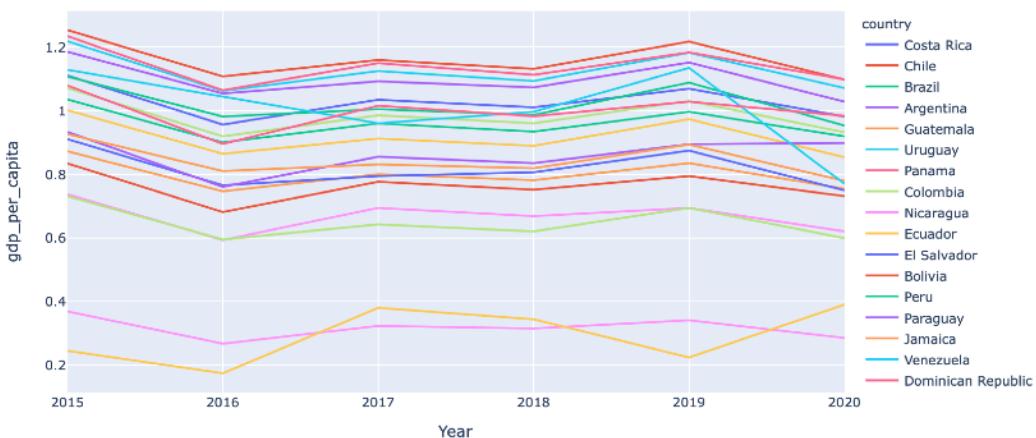
GDP Trend in Australia



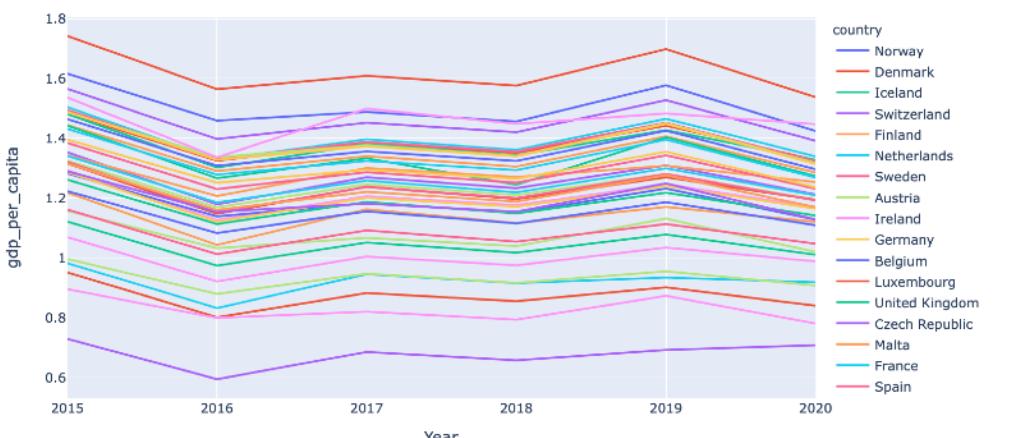
GDP Trend in North America



GDP Trend in South America



GDP Trend in Europe



GDP Trend in Africa



```
In [109]: df = df_data.query("continent=='Asia'")
fig = px.line(df, x="Year", y="freedom", color='country', title='freedom Trend in Asia')
fig.show()

df = df_data.query("continent=='Australia'")
fig = px.line(df, x="Year", y="freedom", color='country', title='freedom Trend in Australia')
fig.show()

df = df_data.query("continent=='North America'")
fig = px.line(df, x="Year", y="freedom", color='country', title='freedom Trend in North America')
fig.show()

df = df_data.query("continent=='South America'")
fig = px.line(df, x="Year", y="freedom", color='country', title='freedom Trend in South America')
fig.show()

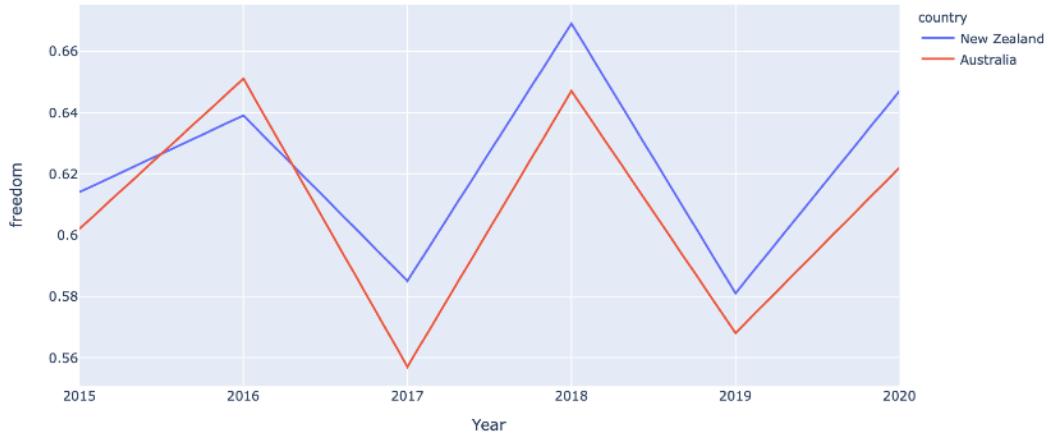
df = df_data.query("continent=='Europe'")
fig = px.line(df, x="Year", y="freedom", color='country', title='freedom Trend in Europe')
fig.show()

df = df_data.query("continent=='Africa'")
fig = px.line(df, x="Year", y="freedom", color='country', title='freedom Trend in Africa')
fig.show()
```

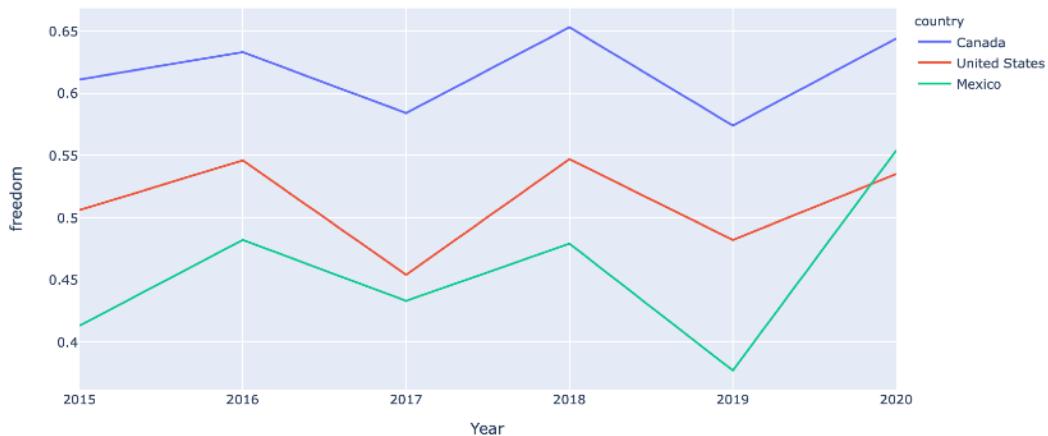
freedom Trend in Asia



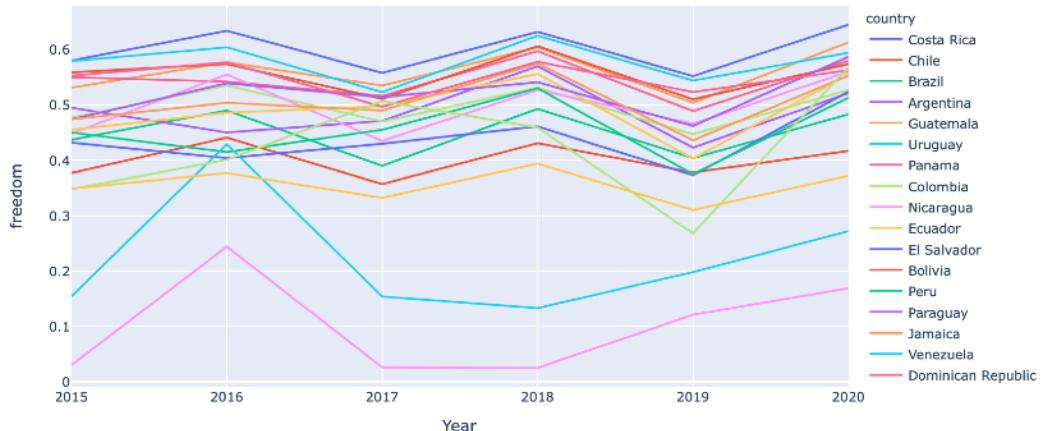
freedom Trend in Australia



freedom Trend in North America



freedom Trend in South America



freedom Trend in Europe

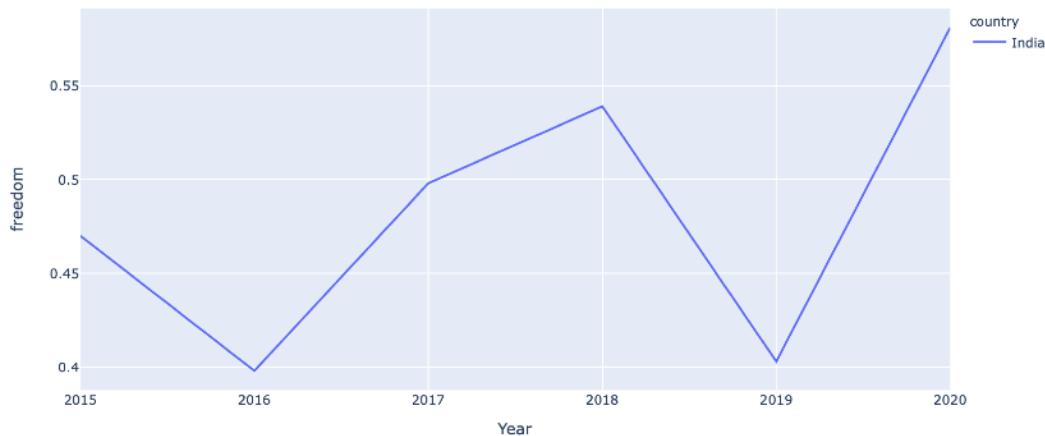


freedom Trend in Africa



```
In [110]: df = df_data.query("country=='India'")
fig = px.line(df, x="Year", y="freedom", color='country', title='GDP Trend in Asia')
fig.show()
```

GDP Trend in Asia



Dataset:

2015

2015_report										
1	country	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual	continent
2	Norway	7.537	1.616	1.534	0.797	0.635	0.362	0.316	2.277	Europe
3	Denmark	7.522	1.482	1.551	0.793	0.626	0.355	0.401	2.314	Europe
4	Iceland	7.504	1.481	1.611	0.834	0.627	0.476	0.154	2.323	Europe
5	Switzerland	7.494	1.565	1.517	0.858	0.620	0.291	0.367	2.277	Europe
6	Finland	7.469	1.444	1.540	0.809	0.618	0.245	0.383	2.430	Europe
7	Netherlands	7.377	1.504	1.429	0.811	0.585	0.470	0.283	2.295	Europe
8	Canada	7.316	1.479	1.481	0.835	0.611	0.436	0.287	2.187	North America
9	New Zealand	7.314	1.406	1.548	0.817	0.614	0.500	0.383	2.046	Australia
10	Sweden	7.284	1.494	1.478	0.831	0.613	0.385	0.384	2.098	Europe
11	Australia	7.284	1.484	1.510	0.844	0.602	0.478	0.301	2.065	Australia
12	Israel	7.213	1.375	1.376	0.838	0.406	0.330	0.085	2.802	Asia
13	Costa Rica	7.079	1.110	1.416	0.760	0.580	0.215	0.100	2.899	South America
14	Austria	7.006	1.487	1.460	0.815	0.568	0.316	0.221	2.139	Europe
15	United States	6.993	1.546	1.420	0.774	0.506	0.393	0.136	2.218	North America
16	Ireland	6.977	1.536	1.558	0.810	0.573	0.428	0.298	1.774	Europe
17	Germany	6.951	1.488	1.473	0.799	0.563	0.336	0.277	2.016	Europe
18	Belgium	6.891	1.464	1.462	0.818	0.540	0.232	0.251	2.124	Europe
19	Luxembourg	6.863	1.742	1.458	0.845	0.597	0.283	0.319	1.620	Europe
20	United Kingdom	6.714	1.442	1.496	0.805	0.508	0.493	0.265	1.704	Europe
21	Chile	6.652	1.253	1.284	0.819	0.377	0.327	0.082	2.510	South America
22	United Arab Emirates	6.648	1.626	1.266	0.727	0.608	0.361	0.324	1.735	Asia
23	Brazil	6.635	1.107	1.431	0.617	0.437	0.162	0.111	2.769	South America
24	Czech Republic	6.609	1.353	1.434	0.754	0.491	0.088	0.037	2.452	Europe
25	Argentina	6.599	1.185	1.440	0.695	0.495	0.109	0.060	2.614	South America
26	Mexico	6.578	1.153	1.211	0.710	0.413	0.121	0.133	2.837	North America
27	Singapore	6.572	1.692	1.354	0.949	0.550	0.346	0.464	1.216	Asia
28	Malta	6.527	1.343	1.488	0.822	0.589	0.575	0.153	1.557	Europe
29	Guatemala	6.454	0.872	1.256	0.540	0.531	0.283	0.077	2.894	South America
30	Uruguay	6.454	1.218	1.412	0.719	0.579	0.175	0.178	2.172	South America
31	Panama	6.452	1.234	1.373	0.706	0.550	0.211	0.071	2.307	South America

2016

2016_report

1	country	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual	continent
2	Switzerland	7.587	1.397	1.350	0.941	0.666	0.297	0.420	2.517	Europe
3	Iceland	7.561	1.302	1.402	0.948	0.629	0.436	0.141	2.702	Europe
4	Denmark	7.527	1.325	1.361	0.875	0.649	0.341	0.484	2.492	Europe
5	Norway	7.522	1.459	1.331	0.885	0.670	0.347	0.365	2.465	Europe
6	Canada	7.427	1.326	1.323	0.906	0.633	0.458	0.330	2.452	North America
7	Finland	7.406	1.290	1.318	0.889	0.642	0.234	0.414	2.620	Europe
8	Netherlands	7.378	1.329	1.280	0.893	0.616	0.476	0.318	2.466	Europe
9	Sweden	7.364	1.332	1.289	0.911	0.660	0.363	0.438	2.371	Europe
10	New Zealand	7.286	1.250	1.320	0.908	0.639	0.475	0.429	2.264	Australia
11	Australia	7.284	1.334	1.309	0.932	0.651	0.436	0.356	2.266	Australia
12	Israel	7.278	1.229	1.224	0.914	0.413	0.332	0.078	3.089	Asia
13	Costa Rica	7.226	0.956	1.238	0.860	0.634	0.255	0.106	3.177	South America
14	Austria	7.200	1.337	1.297	0.890	0.624	0.331	0.187	2.533	Europe
15	Mexico	7.187	1.021	0.915	0.814	0.482	0.141	0.213	3.602	North America
16	United States	7.119	1.395	1.247	0.862	0.546	0.401	0.159	2.510	North America
17	Brazil	6.983	0.981	1.233	0.697	0.490	0.146	0.175	3.260	South America
18	Luxembourg	6.946	1.564	1.220	0.919	0.616	0.280	0.378	1.970	Europe
19	Ireland	6.940	1.336	1.369	0.895	0.618	0.459	0.287	1.976	Europe
20	Belgium	6.937	1.308	1.286	0.897	0.585	0.223	0.225	2.415	Europe
21	United Arab Emirates	6.901	1.427	1.126	0.809	0.642	0.264	0.386	2.247	Asia
22	United Kingdom	6.867	1.266	1.285	0.909	0.596	0.519	0.321	1.970	Europe
23	Venezuela	6.810	1.044	1.256	0.721	0.429	0.058	0.111	3.191	South America
24	Singapore	6.798	1.522	1.020	1.025	0.543	0.311	0.492	1.885	Asia
25	Panama	6.786	1.064	1.199	0.797	0.542	0.244	0.093	2.848	South America
26	Germany	6.750	1.328	1.299	0.892	0.615	0.282	0.218	2.116	Europe
27	Chile	6.670	1.107	1.124	0.859	0.441	0.334	0.129	2.676	South America
28	France	6.575	1.278	1.260	0.946	0.550	0.123	0.206	2.211	Europe
29	Argentina	6.574	1.054	1.248	0.787	0.450	0.115	0.085	2.836	South America
30	Czech Republic	6.505	1.179	1.206	0.845	0.464	0.107	0.027	2.678	Europe
31	Uruguay	6.485	1.062	1.209	0.812	0.604	0.232	0.246	2.321	South America
32	Colombia	6.477	0.919	1.240	0.691	0.535	0.184	0.051	2.857	South America
33	Thailand	6.455	0.967	1.265	0.739	0.557	0.576	0.032	2.319	Asia
34	Saudi Arabia	6.411	1.395	1.084	0.720	0.310	0.137	0.325	2.439	Asia
35	Spain	6.329	1.230	1.314	0.956	0.460	0.182	0.064	2.124	Europe
36	Malta	6.302	1.207	1.302	0.887	0.604	0.518	0.136	1.649	Europe
37	Kuwait	6.295	1.554	1.166	0.725	0.555	0.162	0.256	1.876	Asia
38	El Salvador	6.130	0.765	1.025	0.677	0.404	0.107	0.118	3.035	South America
39	Guatemala	6.123	0.746	1.044	0.644	0.577	0.275	0.095	2.743	South America
40	Uzbekistan	6.003	0.632	1.340	0.598	0.658	0.228	0.308	2.237	Asia

2017

2017_report

country	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual	continent
Finland	7.769	1.340	1.587	0.986	0.596	0.153	0.393	2.231	Europe
Denmark	7.600	1.383	1.573	0.996	0.592	0.252	0.410	2.400	Europe
Norway	7.554	1.488	1.582	1.028	0.603	0.271	0.341	2.446	Europe
Iceland	7.494	1.380	1.624	1.026	0.591	0.354	0.118	2.506	Europe
Netherlands	7.488	1.396	1.522	0.999	0.557	0.322	0.298	2.512	Europe
Switzerland	7.480	1.452	1.526	1.052	0.572	0.263	0.343	2.520	Europe
Sweden	7.343	1.387	1.487	1.009	0.574	0.267	0.373	2.657	Europe
New Zealand	7.307	1.303	1.557	1.026	0.585	0.330	0.380	2.693	Australia
Canada	7.278	1.365	1.505	1.039	0.584	0.285	0.308	2.722	North America
Austria	7.246	1.376	1.475	1.016	0.532	0.244	0.226	2.754	Europe
Australia	7.228	1.372	1.548	1.036	0.557	0.332	0.290	2.772	Australia
Costa Rica	7.167	1.034	1.441	0.963	0.558	0.144	0.093	2.833	South America
Israel	7.139	1.276	1.455	1.029	0.371	0.261	0.082	2.861	Asia
Luxembourg	7.090	1.609	1.479	1.012	0.526	0.194	0.316	2.910	Europe
United Kingdom	7.054	1.333	1.538	0.996	0.450	0.348	0.278	2.946	Europe
Ireland	7.021	1.499	1.553	0.999	0.516	0.298	0.310	2.979	Europe
Germany	6.985	1.373	1.454	0.987	0.495	0.261	0.265	3.015	Europe
Belgium	6.923	1.356	1.504	0.986	0.473	0.160	0.210	3.077	Europe
United States	6.892	1.433	1.457	0.874	0.454	0.280	0.128	3.108	North America
Czech Republic	6.852	1.269	1.487	0.920	0.457	0.046	0.036	3.148	Europe
United Arab Emirates	6.825	1.503	1.310	0.825	0.598	0.262	0.182	3.175	Asia
Malta	6.726	1.300	1.520	0.999	0.564	0.375	0.151	3.274	Europe
Mexico	6.595	1.070	1.323	0.861	0.433	0.074	0.073	3.405	North America
France	6.592	1.324	1.472	1.045	0.436	0.111	0.183	3.408	Europe
Chile	6.444	1.159	1.369	0.920	0.357	0.187	0.056	3.556	South America
Guatemala	6.436	0.800	1.269	0.746	0.535	0.175	0.078	3.564	South America
Saudi Arabia	6.375	1.403	1.357	0.795	0.439	0.080	0.132	3.625	Asia
Spain	6.354	1.286	1.484	1.062	0.362	0.153	0.079	3.646	Europe
Panama	6.321	1.149	1.442	0.910	0.516	0.109	0.054	3.679	South America
Brazil	6.300	1.004	1.439	0.802	0.390	0.099	0.086	3.700	South America

2018

2018_report

1	country	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual	continent
2	Finland	7.632	1.305	1.592	0.874	0.681	0.202	0.393	2.368	Europe
3	Norway	7.594	1.456	1.582	0.861	0.686	0.286	0.340	2.406	Europe
4	Denmark	7.555	1.351	1.590	0.868	0.683	0.284	0.408	2.445	Europe
5	Iceland	7.495	1.343	1.644	0.914	0.677	0.353	0.138	2.505	Europe
6	Switzerland	7.487	1.420	1.549	0.927	0.660	0.256	0.357	2.513	Europe
7	Netherlands	7.441	1.361	1.488	0.878	0.638	0.333	0.295	2.559	Europe
8	Canada	7.328	1.330	1.532	0.896	0.653	0.321	0.291	2.672	North America
9	New Zealand	7.324	1.268	1.601	0.876	0.669	0.365	0.389	2.676	Australia
10	Sweden	7.314	1.355	1.501	0.913	0.659	0.285	0.383	2.686	Europe
11	Australia	7.272	1.340	1.573	0.910	0.647	0.361	0.302	2.728	Australia
12	United Kingdom	7.190	1.244	1.433	0.888	0.464	0.262	0.082	2.810	Europe
13	Austria	7.139	1.341	1.504	0.891	0.617	0.242	0.224	2.861	Europe
14	Costa Rica	7.072	1.010	1.459	0.817	0.632	0.143	0.101	2.928	South America
15	Ireland	6.977	1.448	1.583	0.876	0.614	0.307	0.306	3.023	Europe
16	Germany	6.965	1.340	1.474	0.861	0.586	0.273	0.280	3.035	Europe
17	Belgium	6.927	1.324	1.483	0.894	0.583	0.188	0.240	3.073	Europe
18	Luxembourg	6.910	1.576	1.520	0.896	0.632	0.196	0.321	3.090	Europe
19	United States	6.886	1.398	1.471	0.819	0.547	0.291	0.133	3.114	North America
20	Israel	6.814	1.301	1.559	0.883	0.533	0.354	0.272	3.186	Asia
21	United Arab Emirates	6.774	2.096	0.776	0.670	0.284	0.186	0.312	3.226	Asia
22	Czech Republic	6.711	1.233	1.489	0.854	0.543	0.064	0.034	3.289	Europe
23	Malta	6.627	1.270	1.525	0.884	0.645	0.376	0.142	3.373	Europe
24	France	6.489	1.293	1.466	0.908	0.520	0.098	0.176	3.511	Europe
25	Mexico	6.488	1.038	1.252	0.761	0.479	0.069	0.095	3.512	North America
26	Chile	6.476	1.131	1.331	0.808	0.431	0.197	0.061	3.524	South America
27	Panama	6.430	1.112	1.438	0.759	0.597	0.125	0.063	3.570	South America
28	Brazil	6.419	0.986	1.474	0.675	0.493	0.110	0.088	3.581	South America
29	Argentina	6.388	1.073	1.468	0.744	0.570	0.062	0.054	3.612	South America
30	Guatemala	6.382	0.781	1.268	0.608	0.604	0.179	0.071	3.618	South America
31	Uruguay	6.379	1.093	1.459	0.771	0.625	0.130	0.155	3.621	South America
32	Saudi Arabia	6.371	1.379	1.331	0.633	0.509	0.098	0.127	3.629	Asia
33	Singapore	6.343	1.529	1.451	1.008	0.631	0.261	0.457	3.657	Asia
34	Malaysia	6.322	1.161	1.258	0.669	0.356	0.311	0.059	3.678	Asia
35	Spain	6.310	1.251	1.538	0.965	0.449	0.142	0.074	3.690	Europe
36	Colombia	6.260	0.960	1.439	0.635	0.531	0.099	0.039	3.740	South America
37	Slovakia	6.173	1.210	1.537	0.776	0.354	0.118	0.014	3.827	Europe
38	El Salvador	6.167	0.806	1.231	0.639	0.461	0.065	0.082	3.833	South America
39	Nicaragua	6.141	0.668	1.319	0.700	0.527	0.208	0.128	3.859	South America

2019

		2019_report									
1	country	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual	continent	
2	Denmark	7.526	1.442	1.164	0.795	0.579	0.362	0.445	2.739	Europe	
3	Switzerland	7.509	1.527	1.145	0.863	0.586	0.281	0.412	2.695	Europe	
4	Iceland	7.501	1.427	1.183	0.867	0.566	0.477	0.150	2.831	Europe	
5	Norway	7.498	1.577	1.127	0.796	0.596	0.379	0.358	2.665	Europe	
6	Finland	7.413	1.406	1.135	0.811	0.571	0.255	0.410	2.826	Europe	
7	Canada	7.404	1.440	1.096	0.828	0.574	0.448	0.313	2.705	North America	
8	Netherlands	7.339	1.465	1.029	0.812	0.552	0.474	0.299	2.707	Europe	
9	New Zealand	7.334	1.361	1.173	0.831	0.581	0.494	0.419	2.476	Australia	
10	Australia	7.313	1.444	1.105	0.851	0.568	0.474	0.323	2.547	Australia	
11	Sweden	7.291	1.452	1.088	0.831	0.582	0.383	0.409	2.547	Europe	
12	Israel	7.267	1.338	0.995	0.849	0.364	0.323	0.087	3.310	Asia	
13	Austria	7.119	1.450	1.084	0.806	0.544	0.329	0.213	2.693	Europe	
14	United States	7.104	1.508	1.048	0.779	0.482	0.411	0.149	2.728	North America	
15	Costa Rica	7.087	1.069	1.022	0.761	0.552	0.226	0.105	3.352	South America	
16	Germany	6.994	1.448	1.098	0.815	0.535	0.305	0.286	2.509	Europe	
17	Brazil	6.952	1.088	1.039	0.614	0.404	0.158	0.142	3.507	South America	
18	Belgium	6.929	1.425	1.052	0.820	0.514	0.242	0.262	2.614	Europe	
19	Ireland	6.907	1.483	1.162	0.815	0.540	0.450	0.298	2.160	Europe	
20	Luxembourg	6.871	1.698	1.040	0.845	0.549	0.276	0.353	2.111	Europe	
21	Mexico	6.778	1.115	0.715	0.711	0.377	0.117	0.184	3.559	North America	
22	Singapore	6.739	1.646	0.868	0.947	0.488	0.327	0.470	1.994	Asia	
23	United Kingdom	6.725	1.403	1.087	0.810	0.500	0.502	0.274	2.150	Europe	
24	Chile	6.705	1.217	0.906	0.819	0.378	0.316	0.115	2.955	South America	
25	Panama	6.701	1.183	0.989	0.708	0.489	0.242	0.084	3.006	South America	
26	Argentina	6.650	1.151	1.066	0.697	0.423	0.110	0.073	3.130	South America	
27	Czech Republic	6.596	1.309	1.008	0.764	0.414	0.099	0.040	2.962	Europe	
28	United Arab Emirates	6.573	1.574	0.871	0.730	0.562	0.266	0.356	2.215	Asia	
29	Uruguay	6.545	1.182	1.031	0.722	0.544	0.181	0.214	2.671	South America	
30	Malta	6.488	1.308	1.099	0.803	0.550	0.562	0.176	1.990	Europe	
31	Colombia	6.481	1.030	1.022	0.597	0.447	0.156	0.054	3.175	South America	
32	France	6.478	1.395	1.005	0.838	0.466	0.122	0.178	2.474	Europe	
33	Thailand	6.474	1.089	1.045	0.649	0.496	0.587	0.028	2.580	Asia	
34	Saudi Arabia	6.379	1.490	0.848	0.593	0.379	0.155	0.300	2.615	Asia	
35	Spain	6.361	1.343	1.129	0.879	0.375	0.177	0.061	2.397	Europe	
36	Algeria	6.355	1.053	0.833	0.618	0.210	0.070	0.162	3.409	Africa	
37	Guatemala	6.324	0.835	0.871	0.540	0.504	0.288	0.087	3.199	South America	
38	Kuwait	6.239	1.617	0.878	0.636	0.432	0.160	0.237	2.281	Asia	
39	Bahrain	6.218	1.440	0.944	0.657	0.474	0.171	0.258	2.274	Asia	
40	Venezuela	6.084	1.134	1.033	0.619	0.198	0.043	0.083	2.975	South America	

2020

		2020_report									
1	country	happiness_score	gdp_per_capita	family	health	freedom	generosity	government_trust	dystopia_residual	continent	
2	Finland	7.809	1.285	1.500	0.961	0.662	0.160	0.478	2.763	Europe	
3	Denmark	7.646	1.327	1.503	0.979	0.665	0.243	0.495	2.433	Europe	
4	Switzerland	7.560	1.391	1.472	1.041	0.629	0.269	0.408	2.350	Europe	
5	Iceland	7.504	1.327	1.548	1.001	0.662	0.362	0.145	2.461	Europe	
6	Norway	7.488	1.424	1.495	1.008	0.670	0.288	0.434	2.168	Europe	
7	Netherlands	7.449	1.339	1.464	0.976	0.614	0.336	0.369	2.352	Europe	
8	Sweden	7.353	1.322	1.433	0.986	0.650	0.273	0.442	2.246	Europe	
9	New Zealand	7.300	1.242	1.487	1.008	0.647	0.326	0.461	2.128	Australia	
10	Austria	7.294	1.317	1.437	1.001	0.603	0.256	0.281	2.398	Europe	
11	Luxembourg	7.238	1.537	1.388	0.986	0.610	0.196	0.367	2.154	Europe	
12	Canada	7.232	1.302	1.435	1.023	0.644	0.282	0.352	2.195	North America	
13	Australia	7.223	1.310	1.477	1.023	0.622	0.325	0.336	2.130	Australia	
14	United Kingdom	7.165	1.273	1.458	0.976	0.525	0.373	0.323	2.237	Europe	
15	Israel	7.129	1.216	1.403	1.008	0.421	0.267	0.100	2.713	Asia	
16	Costa Rica	7.121	0.981	1.375	0.940	0.645	0.131	0.096	2.953	South America	
17	Ireland	7.094	1.447	1.471	0.976	0.588	0.295	0.373	1.944	Europe	
18	Germany	7.076	1.314	1.369	0.972	0.564	0.252	0.309	2.295	Europe	
19	United States	6.940	1.374	1.405	0.832	0.535	0.298	0.152	2.344	North America	
20	Czech Republic	6.911	1.212	1.405	0.895	0.506	0.046	0.050	2.797	Europe	□
21	Belgium	6.864	1.296	1.399	0.965	0.500	0.147	0.209	2.349	Europe	
22	United Arab Emirates	6.791	1.431	1.251	0.788	0.653	0.281	0.220	2.167	Asia	
23	Malta	6.773	1.253	1.443	0.972	0.633	0.341	0.179	1.952	Europe	
24	France	6.664	1.268	1.459	1.030	0.514	0.113	0.227	2.053	Europe	
25	Mexico	6.465	1.024	1.226	0.832	0.554	0.083	0.083	2.663	North America	
26	Uruguay	6.440	1.071	1.425	0.857	0.594	0.132	0.193	2.167	South America	
27	Saudi Arabia	6.406	1.334	1.310	0.760	0.548	0.087	0.163	2.203	Asia	
28	Spain	6.401	1.231	1.421	1.051	0.426	0.165	0.110	1.997	Europe	
29	Guatemala	6.399	0.754	1.174	0.706	0.613	0.171	0.098	2.883	South America	
30	Italy	6.387	1.236	1.347	1.023	0.321	0.170	0.040	2.250	Europe	
31	Singapore	6.377	1.520	1.395	1.138	0.635	0.219	0.533	0.937	Asia	
32	Brazil	6.376	0.953	1.363	0.766	0.483	0.132	0.107	2.572	South America	
33	Slovenia	6.363	1.209	1.465	0.933	0.647	0.146	0.077	1.889	Europe	
34	El Salvador	6.348	0.749	1.149	0.753	0.524	0.119	0.117	2.937	South America	
35	Kosovo	6.325	0.840	1.184	0.673	0.557	0.325	0.009	2.737	Europe	
36	Panama	6.305	1.098	1.376	0.879	0.580	0.097	0.054	2.221	South America	
37	Slovakia	6.281	1.195	1.424	0.853	0.424	0.117	0.011	2.256	Europe	
38	Uzbekistan	6.258	0.697	1.434	0.717	0.693	0.363	0.280	2.073	Asia	
39	Chile	6.228	1.097	1.323	0.889	0.417	0.156	0.063	2.283	South America	