Global Happiness Score since 2008

Contributors: Hanya Ansari, Daren Aguilera, Nikhita Kalluri

Chosen dataset

World Happiness Report 2023: indices related to happiness and wellbeing by country 2008-present

Data description

For the purpose of this course project, an analysis was conducted on the 2023 World Happiness Report. The World Happiness Report was published as a means to measure the success of countries as a result of an individual's happiness score. The dataset used for this analysis comprises information from 160 countries, ranging the years 2005 to 2022. 2200 observations were recorded in this dataset, of which analysis was conducted on 2153 observations, from which 47 observations were removed as a result of NaN values. While 11 variables were included in the original dataset, analysis was specifically conducted on Country Name, Year, Life Ladder, Log GDP per capita, Social Support, Freedom to make life choices, Positive Effect, Negative Effect, and the added variables of Continent and Net Effect. The Happiness score, or Life ladder, represents the national average response to the question of life evaluations; The scores on the ladder ranges from the top to the bottom, representing a spectrum for the best possible life to the worst possible life. The Log GDP per capita represents the measure of the economic output of a nation per person specific to a country based on its population. Social Support measures a respondents likelihood of having someone to count on in times of trouble. Freedom to make life choices examines respondents satisfaction, or lack of, in their freedom to choose what they want to do with their life. Additionally, the variables Positive Effect and Negative Effect measure feelings of laughter, enjoyment, sadness, and anger, respectively (Helliwell et al. 1-2). Finally, the variable of Continent was added to the dataset to categorize each Country Name with its respective Continent. The variables chosen for analysis provide a comprehensive view of a continent's overall happiness and offers insights into various emotional and social dimensions.

Question(s) of Interest

What factors most influence the Happiness score?

- Has the average Happiness Score improved globally since 2008?
- Which continent has seen the largest net positive effect on the Happiness Score over time?
- What are the top factors that affect the Happiness Score?

Data Analysis

Tidy Data

Based on the distribution of missing values, we noticed that the variables Log GDP Per Capita and Positive Effect were skewed to the left while Negative Effect was skewed to the right. Other options to address missingness such as imputation would introduce bias by not accounting for the lack of normality of the distribution of values.

We chose to tidy the dataset by dropping missing observations. Since there was only a maximum of 24 missing observations, our tidy dataset still contains over 2000 observations so this was a reasonable choice. We also introduced a Net Effect variable into the dataset by subtracting Negative Effect from Positive Effect. Moreover, we removed observations before 2008 to get a uniform distribution with similar amounts of observations per year.

EDA and Data Visualization

In order to answer our questions of interest, we began the exploratory data analysis process to understand the distributions and impacts of our variables on happiness scores. We focused on understanding positive effects, negative effects, and net effects over time, both globally and regionally when grouped by. The net effects were calculated by taking the difference between positive and negative effects for each specific year within a continent. Subsequently, we applied this analysis to 'Life ladder', the variable representing the happiness score in our dataset. It's worth noting that the change in net effect on happiness and the change in Life ladder yielded different results. Given that the Life ladder is considered the happiness score in our original dataset, we chose to use this variable to carry out the analysis of this project.

To identify the top factors influencing the happiness score, we constructed a correlation matrix. This matrix enabled us to assess the significance of the relationships between explanatory variables and the Life ladder. We determined that Log GDP per capita, Social Support, Perceptions of Corruption, and Freedom to Make Life choices were the most important factors in our analysis. We then visualized these variables in relation to continents and explored their contribution to the Life ladder.

Linear Regression

We used our variables of interest from the correlation matrix and implemented them into a fitted multilinear regression model.

$$\beta_{Happiness} = \beta_0 + \beta_{Economy} + \beta_{Social\ support} + \beta_{Freedom} + \beta_{Corruption}$$

Our model suggests that the predictors selected have a statistically significant relationship with the measurement of Happiness. We obtained positive coefficients for Log GDP per capita, Social support, and freedom to make life choices, implying that an increase in these predictors will likely increase the mean value for Happiness. On the other hand, the negative coefficient obtained for Perception of corruption implies an increase here is associated with a decrease in average Happiness. Our most notable coefficient obtained was from Social support, suggesting the most influence on Happiness scores. This supports evidence for the importance of Social support, even over the role of the economy.

Cluster Analysis

The final step in our project was cluster analysis, conducted through K-means clustering. We determined that the optimal number of clusters was 3 by plotting the WCSS against the number of clusters ranging from 1-11. We then standardized the data and generated 3 clusters which we visualized through a plot. Each of the three clusters was visibly different in the results when comparing each distinct explanatory variable to the Life ladder suggesting that it performed well because there was low variance.

Summary of Findings

Our project was designed with the purpose to determine what factors most influenced the Happiness Score. We addressed this question by conducting analysis on the average Happiness Score, the largest net positive effect on the Happiness Score, and by examining various emotional and social dimensions. The findings of our project are detailed below:

Has the average Happiness Score improved globally since 2008?

- From 2008 to 2022 the overall Happiness score has seen a significant increase across the world. Between the years of 2008 and 2018, the Life Ladder score remained below 5.60. In 2020, the Happiness Score peaked at an all time high of 5.75; This was due in large part to the fact that “Respondents had been able to discover and share the capacity to care for each other in difficult times” during the COVID-19 pandemic (Booth, 2023).”

Which continent has seen the largest net positive effect on the Happiness Score over time?

- Positive Effects: Europe has the highest change in positive effects with a change of 7.671 followed by Africa with a change of 2.327. The least change in positive effects occurred in Asia with a change of -8.592.
- Negative Effects: Africa has the highest change in negative effects with a change of 4.715 followed by Europe with a smaller change of 2.962. The least change in negative effects occurred in Asia with a change of -3.035.
- Net Effects: Europe has the highest change in net effects with a change of 4.709 followed by Oceania with a change of -0.127. The least change in net effects occurred in Asia with a change of -5.557.
- Happiness score (from Life Ladder): South America has the greatest positive increase in Happiness score throughout the years of 0.366 followed by Asia with a change of 0.3079. Oceaniar has the highest negative change from 2008 to 2023 with a decrease in happiness score of -0.332, followed by Europe with a decrease of -0.148.

What are the top five factors that affect the Happiness Score?

- Creating a correlation matrix, we identified four key factors that significantly affect the happiness score, measured by the Life ladder—Log GDP per Capita, Social Support, Freedom to Make Life Choices, and Perception of Corruption.
- Globally, the strongest positive correlation with Life ladder was observed for Log GDP per capita, which exhibits the steepest slope in Fig_12. Social support and freedom to make life choices also exhibited positive correlations, however to a smaller extent. However, Perceptions of Corruption was found to have a negative correlation, indicating that higher perceived corruption in a country's government is associated with lower happiness scores.
- Fig_13 illustrates the distribution of Life ladder by continent based on these explanatory factors. Log GDP per Capita is seen as the dominant variable across all continents, exerting the most influence on Life ladder. Africa has the lowest Life ladder, and was characterized by the lowest Log GDP per capita, Social Support, and Freedom to Make Life Choices, though it has the second-highest Perception of Corruption. Asia shows a moderate Life ladder, while South America and North America had similar scores, with North

America having a higher Log GDP per capita and South America showing higher values for Social Support and Perception of Corruption. Oceania followed by Europe have the highest Life ladders, with the highest values for all explanatory variables except for Perception of Corruption.

Citations

Booth, Robert. “Covid Has Not Affected People’s Happiness around World, Study Reveals.” The Guardian, Guardian News and Media, 20 Mar. 2023.

Helliwell, John F., et al. "Statistical Appendix for 'World happiness, trust and social connections in times of crisis,' Chapter 2 of World Happiness Report 2023." World Happiness Report 2023, 13 Mar. 2023.

Code Appendix

```
import warnings
warnings.filterwarnings("ignore")
```

```
!pip install pycountry-convert
```

```
Requirement already satisfied: pycountry-convert in /usr/local/lib/python3.10/dist-packages (0.7.2)
Requirement already satisfied: pprintpp>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (0.4.0)
Requirement already satisfied: pycountry>=16.11.27.1 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (23.12.11)
Requirement already satisfied: pytest>=3.4.0 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (7.4.3)
Requirement already satisfied: pytest-mock>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (3.12.0)
Requirement already satisfied: pytest-cov>=2.5.1 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (4.1.0)
Requirement already satisfied: repoze.lru>=0.7 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (0.7)
Requirement already satisfied: wheel>=0.30.0 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (0.42.0)
Requirement already satisfied: iniconfig in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry-convert) (2.0.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry-convert) (23.2)
Requirement already satisfied: pluggy<2.0,>=0.12 in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry-convert) (1.3
Requirement already satisfied: exceptiongroup>=1.0.0rc8 in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry-conver
Requirement already satisfied: tomli>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry-convert) (2.0.1)
Requirement already satisfied: coverage[toml]>=5.2.1 in /usr/local/lib/python3.10/dist-packages (from pytest-cov>=2.5.1->pycountry-conve
```

Tidying Data

```
import numpy as np
import pandas as pd
import altair as alt
import statsmodels.api as sm
import matplotlib.pyplot as plt
import pycountry_convert as pc
import plotly.express as px
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score

alt.renderers.enable('colab')
```

```
RendererRegistry.enable('colab')
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
data = pd.read_csv('/content/drive/Shareddrives/F23 PSTAT 100/whr-2023.csv')
data
```

	Country name	year	Life Ladder	Log GDP per capita	Social support	Healthy life expectancy at birth	Freedom to make life choices	Generosity	Perceptions of corruption	Positive affect	Negative affect
0	Afghanistan	2008	3.724	7.350	0.451	50.500	0.718	0.168	0.882	0.414	0.258
1	Afghanistan	2009	4.402	7.509	0.552	50.800	0.679	0.191	0.850	0.481	0.237
2	Afghanistan	2010	4.758	7.614	0.539	51.100	0.600	0.121	0.707	0.517	0.275
3	Afghanistan	2011	3.832	7.581	0.521	51.400	0.496	0.164	0.731	0.480	0.267
4	Afghanistan	2012	3.783	7.661	0.521	51.700	0.531	0.238	0.776	0.614	0.268
...	...	...	...	...	...	...	...	...	...	...	...
2194	Zimbabwe	2018	3.616	7.783	0.775	52.625	0.763	-0.051	0.844	0.658	0.212
2195	Zimbabwe	2019	3.694	7.698	0.759	53.100	0.632	-0.047	0.831	0.658	0.235

```
country_dict = {
    'Taiwan Province of China': 'Taiwan',
    'Hong Kong S.A.R. of China': 'Hong Kong',
    'Turkiye': 'Turkey',
    'Somaliland region':'Somaliland',
    'Congo (Kinshasa)': 'Democratic Republic of the Congo',
    'State of Palestine': 'Palestine'
}
data['Country name'] = data['Country name'].replace(country_dict)

def get_continent(country):
    try:
        country_code = pc.country_name_to_country_alpha2(country, cn_name_format="default")
        continent_code = pc.country_alpha2_to_continent_code(country_code)
        continent_name = pc.convert_continent_code_to_continent_name(continent_code)
        return continent_name
    except:
        return "Unknown"

data['Continent'] = data['Country name'].apply(get_continent)
data

# manually configure last two countries outisde of the function library to appropriate continents
data.loc[data['Country name'] == 'Kosovo', 'Continent'] = 'Europe'
data.loc[data['Country name'] == 'Congo (Brazzaville)', 'Continent'] = 'Africa'

# manually check unknown values
unknown = data[data['Continent']=="Unknown"]['Country name']
# unknown.unique() Check which countries attriuted to 'Unknown'
# no unknown values for continent now, all countries have been appropriately matched to respect their continent


data_subset = data[['Country name', 'year','Life Ladder', 'Log GDP per capita', 'Social support', 'Freedom to make life choices', 'Perceptic
# shape: [2199, 9]

# observe null values within relevant columns of interest
data_subset.isna().sum()
```

```
Country name                      0
year                              0
Life Ladder                       0
Log GDP per capita               20
Social support                   13
Freedom to make life choices     33
Perceptions of corruption       116
Positive affect                  24
Negative affect                  16
Continent                         0
dtype: int64
```

```
# slice rows of observations which contain missing values to inspect randomness
mask_null = data_subset.isna()
missing_rows = data_subset[mask_null.any(axis=1)]
missing_rows
# observations appear to have missing values completely at random (MCAR)
```

	Country name	year	Life Ladder	Log GDP per capita	Social support	Freedom to make life choices	Perceptions of corruption	Positive affect	Negative affect	Continent
13	Afghanistan	2022	1.281	NaN	0.228	0.368	0.733	0.206	0.576	Asia
29	Algeria	2010	5.464	9.306	NaN	0.593	0.618	NaN	NaN	Africa
32	Algeria	2014	6.355	9.355	0.818	NaN	NaN	0.558	0.177	Africa
33	Algeria	2016	5.341	9.383	0.749	NaN	NaN	0.565	0.377	Africa
126	Bahrain	2014	6.165	10.802	NaN	NaN	NaN	NaN	NaN	Asia
...	...	...	...	...	...	...	...	...	...	...
2147	Vietnam	2015	5.076	8.999	0.849	NaN	NaN	0.583	0.232	Asia
2149	Vietnam	2017	5.175	9.111	NaN	NaN	NaN	NaN	NaN	Asia
2155	Yemen	2007	4.477	8.212	0.825	0.673	NaN	0.524	0.379	Asia

```
filtered_df = data_subset[data_subset['year'] >= 2008]
clean_data = filtered_df.dropna()
clean_data['Net Effect'] = clean_data['Positive affect'] - clean_data['Negative affect']

# 46 observations dropped
clean_data
# new dim shape [ 2153 x 7 ]
```

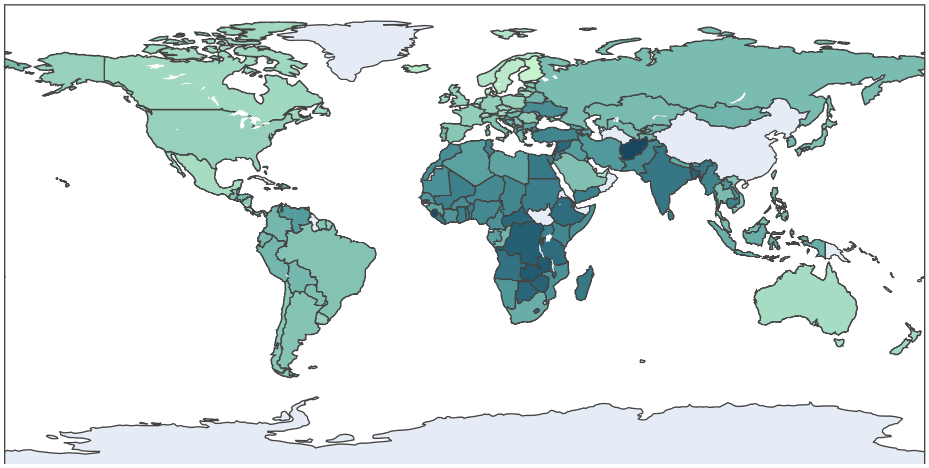
```
<ipython-input-121-ce2fce961550>:3: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
```

	Country name	year	Life Ladder	Log GDP per capita	Social support	Freedom to make life choices	Perceptions of corruption	Positive affect	Negative affect	Continent	Net Effect
0	Afghanistan	2008	3.724	7.350	0.451	0.718	0.882	0.414	0.258	Asia	0.156
1	Afghanistan	2009	4.402	7.509	0.552	0.679	0.850	0.481	0.237	Asia	0.244
2	Afghanistan	2010	4.758	7.614	0.539	0.600	0.707	0.517	0.275	Asia	0.242
3	Afghanistan	2011	3.832	7.581	0.521	0.496	0.731	0.480	0.267	Asia	0.213
4	Afghanistan	2012	3.783	7.661	0.521	0.531	0.776	0.614	0.268	Asia	0.346
...	...	...	...	...	...	...	...	...	...	...	...
2194	Zimbabwe	2018	3.616	7.783	0.775	0.763	0.844	0.658	0.212	Africa	0.446
2195	Zimbabwe	2019	2.694	7.698	0.759	0.632	0.831	0.658	0.235	Africa	0.423
2196	Zimbabwe	2020	3.160	7.596	0.717	0.643	0.789	0.661	0.346	Africa	0.315
2197	Zimbabwe	2021	3.155	7.657	0.685	0.668	0.757	0.610	0.242	Africa	0.368
2198	Zimbabwe	2022	3.296	7.670	0.666	0.652	0.753	0.641	0.191	Africa	0.450

Exploratory Data Analysis

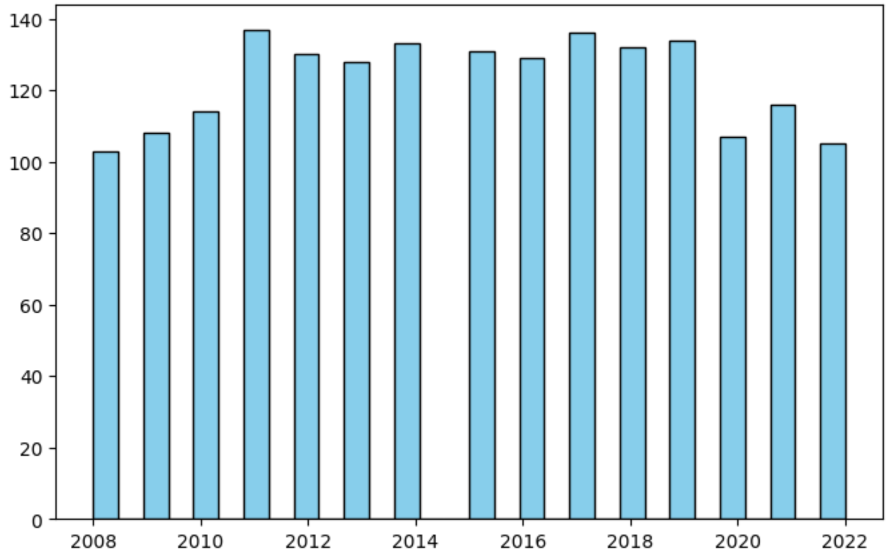
```
# If comparing by continents, what are the weights of each continent by the number of countries it contains?
#continents_n = data_subset.loc[:,['Country name','']]
#cont_n = data_subset['Continent'].value_counts()

fig = px.choropleth(clean_data, locations="Country name", locationmode='country names',
                    color="Life Ladder", hover_name="Country name",
                    title="World Happiness Report: Happiness score by country",
                  color_continuous_scale=px.colors.sequential.Darkmint_r)
# match color to sequential scale, monotone gradient showing change over one variable, Lecture: 'Principal figure design'
fig.show()
```

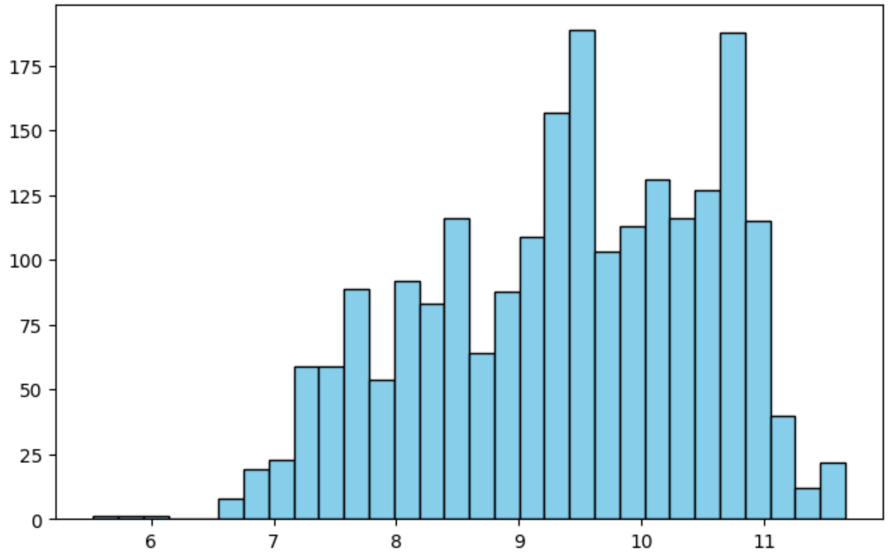
World Happiness Report: Happiness score by country



```
plt.figure(figsize=(8, 5))
plt.hist(clean_data['year'], bins=30, edgecolor='black', label='year', color='skyblue')
# approximately uniform distributed

plt.figure(figsize=(8, 5))
plt.hist(data_subset['Log GDP per capita'], bins=30, edgecolor='black',label='GDP', color='skyblue')
#

plt.figure(figsize=(8, 5))
plt.hist(data_subset['Positive affect'], bins=30, edgecolor='black', color='skyblue')

plt.figure(figsize=(8, 5))
plt.hist(data_subset['Negative affect'], bins=30, edgecolor='black', color='skyblue')
```

```
(array([  9.,  33.,  47., 100., 157., 195., 209., 221., 207., 209., 176.,
        126., 121.,  89.,  85.,  56.,  46.,  36.,  16.,  15.,   8.,   5.,
          6.,   4.,   3.,   2.,   0.,   1.,   0.,   1.]),
 array([0.083     , 0.10373333, 0.12446667, 0.1452    , 0.16593333,
        0.18666667, 0.2074    , 0.22813333, 0.24886667, 0.2696    ,
        0.29033333, 0.31106667, 0.3318    , 0.35253333, 0.37326667,
        0.394     , 0.41473333, 0.43546667, 0.4562    , 0.47693333,
        0.49766667, 0.5184    , 0.53913333, 0.55986667, 0.5806    ,
        0.60133333, 0.62206667, 0.6428    , 0.66353333, 0.68426667,
        0.705     ]),
 <BarContainer object of 30 artists>)
```

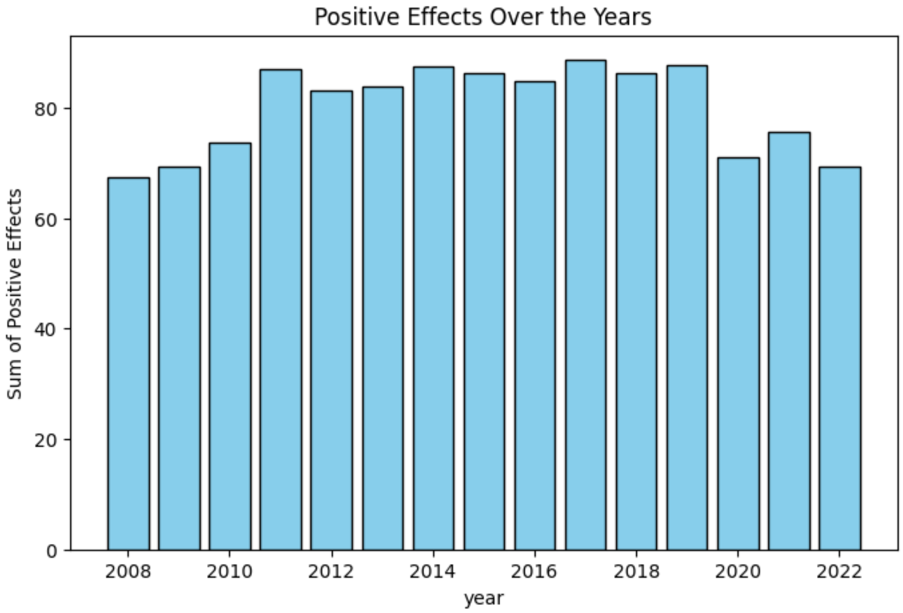




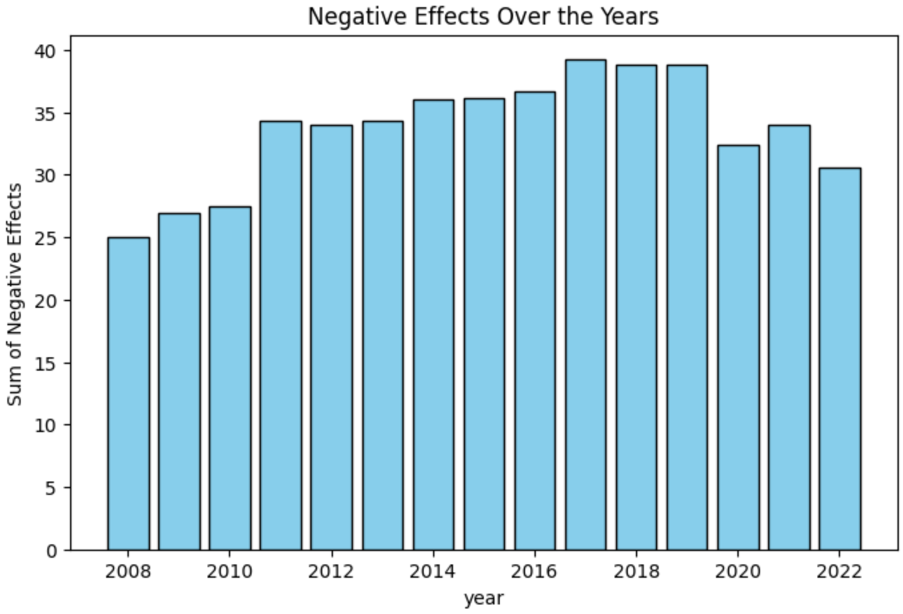
Question 1:

Has the average Happiness Score improved globally since 2008?

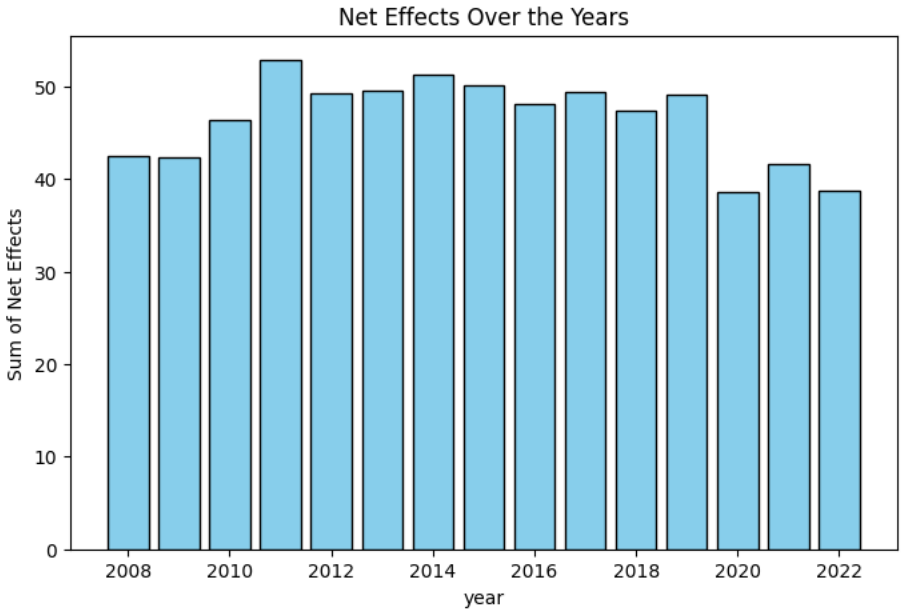
```
fig_1 = clean_data.groupby('year')['Positive affect'].sum().reset_index()

# Plot histogram of positive effects on happiness thorughout years
plt.figure(figsize=(8, 5))
plt.bar(fig_1['year'], fig_1['Positive affect'], color='skyblue', edgecolor='black')
plt.title('Positive Effects Over the Years')
plt.xlabel('year')
plt.ylabel('Sum of Positive Effects')
plt.show()
```



```
fig_2 = clean_data.groupby('year')['Negative affect'].sum().reset_index()

# Plot histogram of positive effects on happiness thorughout years
plt.figure(figsize=(8, 5))
plt.bar(fig_2['year'], fig_2['Negative affect'], color='skyblue', edgecolor='black')
plt.title('Negative Effects Over the Years')
plt.xlabel('year')
plt.ylabel('Sum of Negative Effects')
plt.show()
```



```
fig_3 = clean_data.groupby('year')['Net Effect'].sum().reset_index()

# Plot histogram of positive effects on happiness thorughout years
plt.figure(figsize=(8, 5))
plt.bar(fig_3['year'], fig_3['Net Effect'], color='skyblue', edgecolor='black')
plt.title('Net Effects Over the Years')
plt.xlabel('year')
plt.ylabel('Sum of Net Effects')
plt.show()
```



```
# Calculate Annual Average Happiness Score from 2008 to 2022
plt.figure(figsize = (8, 5))

# Gather the mean Happiness Score per year across every country
annual_score = clean_data.groupby('year')['Life Ladder'].mean()

plt.plot(annual_score.index, annual_score, marker = 'o', linestyle = '-')
plt.xlabel('Year') # Year range from 2008 to 2022
plt.ylabel('Happiness Score') # Life Ladder variable to determine the Happiness Score
plt.title('Trend of Happiness Score globally since 2008')
plt.grid(True)

fig_4 = plt.show
fig_4
# Shows an initial decline in the Life Ladder but then spikes up at 2020
```

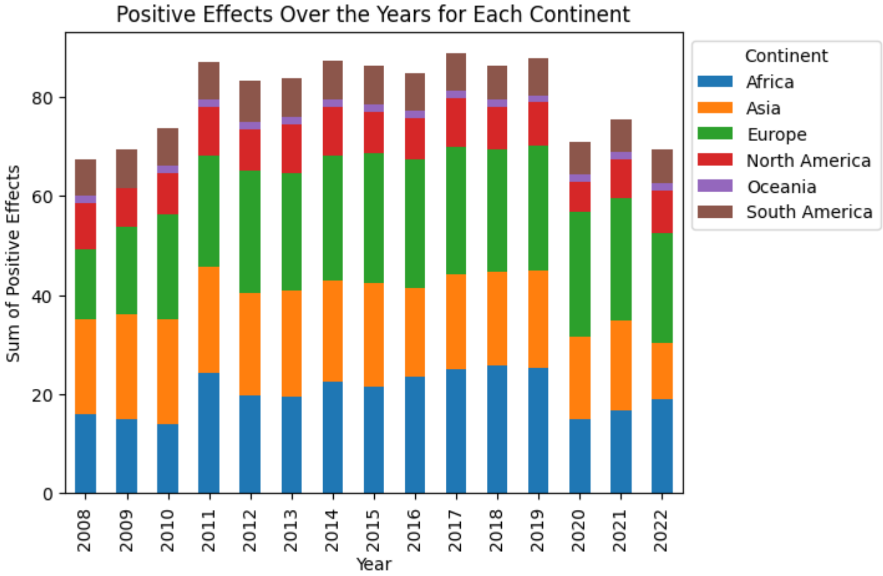
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

Question 2:

Which continent has seen the greatest improvement in Happiness Score over time?

```
positive_effect = clean_data.groupby(['year', 'Continent'])['Positive affect'].sum().reset_index()

pivot_data = positive_effect.pivot(index='year', columns='Continent', values='Positive affect')

# Plot a stacked bar chart
plt.figure(figsize=(8, 5))
pivot_data.plot(kind='bar', stacked=True)

plt.title('Positive Effects Over the Years for Each Continent')
plt.xlabel('Year')
plt.ylabel('Sum of Positive Effects')
plt.legend(title='Continent', loc='upper left', bbox_to_anchor=(1, 1))
fig_5 = plt.show()
fig_5

pos_score_change = pivot_data.diff()

# Sum the changes over all years to get the overall improvement for each continent
overall_improvement = pos_score_change.sum()
overall_improvement
```

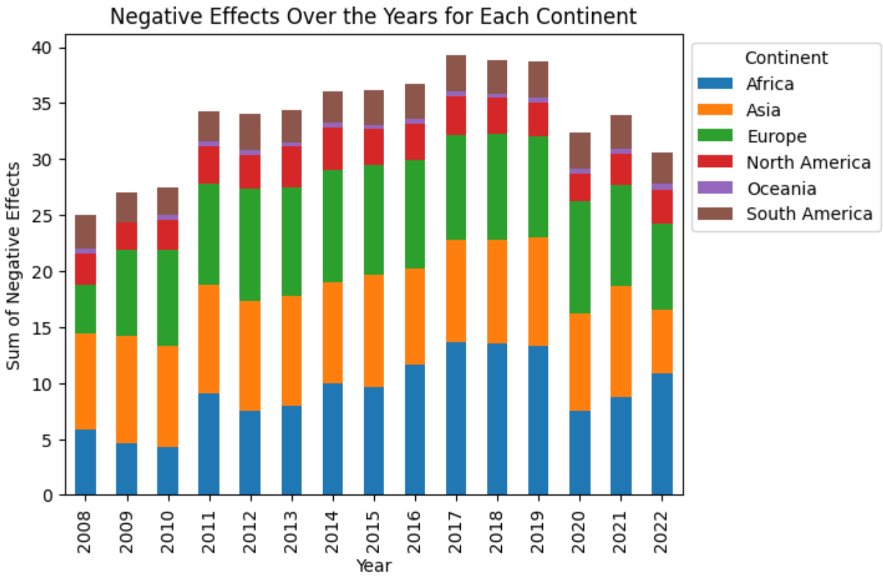
```
<Figure size 800x500 with 0 Axes>
```



```
Continent
Africa            3.071
Asia             -7.976
Europe            8.171
North America    -0.585
Oceania          -0.128
South America    -0.640
dtype: float64
```

```
negative_effect = clean_data.groupby(['year', 'Continent'])['Negative affect'].sum().reset_index()

pivot_data = negative_effect.pivot(index='year', columns='Continent', values='Negative affect')

# Plot a stacked bar chart
plt.figure(figsize=(8, 5))
pivot_data.plot(kind='bar', stacked=True)

plt.title('Negative Effects Over the Years for Each Continent')
plt.xlabel('Year')
plt.ylabel('Sum of Negative Effects')
plt.legend(title='Continent', loc='upper left', bbox_to_anchor=(1, 1))
fig_6 = plt.show()
fig_6

neg_score_change = pivot_data.diff()

# Sum the changes over all years to get the overall improvement for each continent
overall_improvement = neg_score_change.sum()
overall_improvement
```

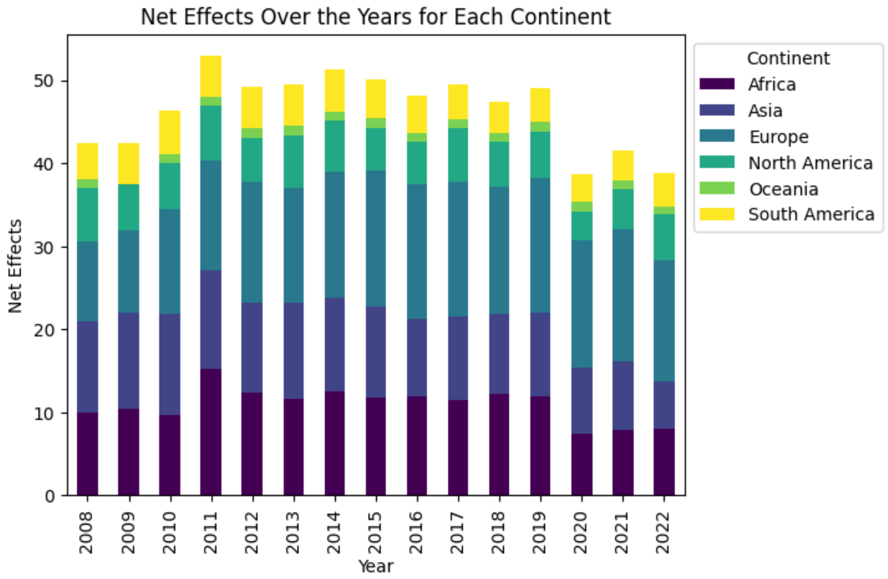
```
<Figure size 800x500 with 0 Axes>
```



```
Continent
Africa           5.007
Asia            -2.817
Europe           3.280
North America    0.309
Oceania         -0.001
South America   -0.147
dtype: float64
```

```
net_effect = clean_data.groupby(['year', 'Continent'])[['Net Effect']].sum().reset_index()
pivot_data = net_effect.pivot(index='year', columns='Continent', values='Net Effect')

plt.figure(figsize=(8, 5))
pivot_data.plot(kind='bar', stacked=True, colormap='viridis')

plt.title('Net Effects Over the Years for Each Continent')
plt.xlabel('Year')
plt.ylabel('Net Effects')
plt.legend(title='Continent', loc='upper left', bbox_to_anchor=(1, 1))
fig_7 = plt.show()
fig_7

net_score_change = pivot_data.diff()

# Sum the changes over all years to get the overall improvement for each continent
overall_improvement = net_score_change.sum()
overall_improvement
```

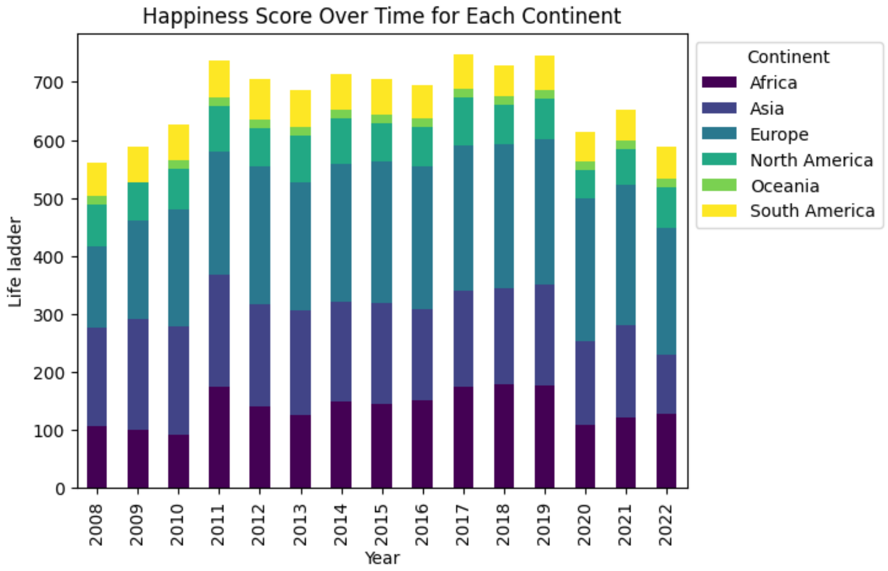
```
<Figure size 800x500 with 0 Axes>
```



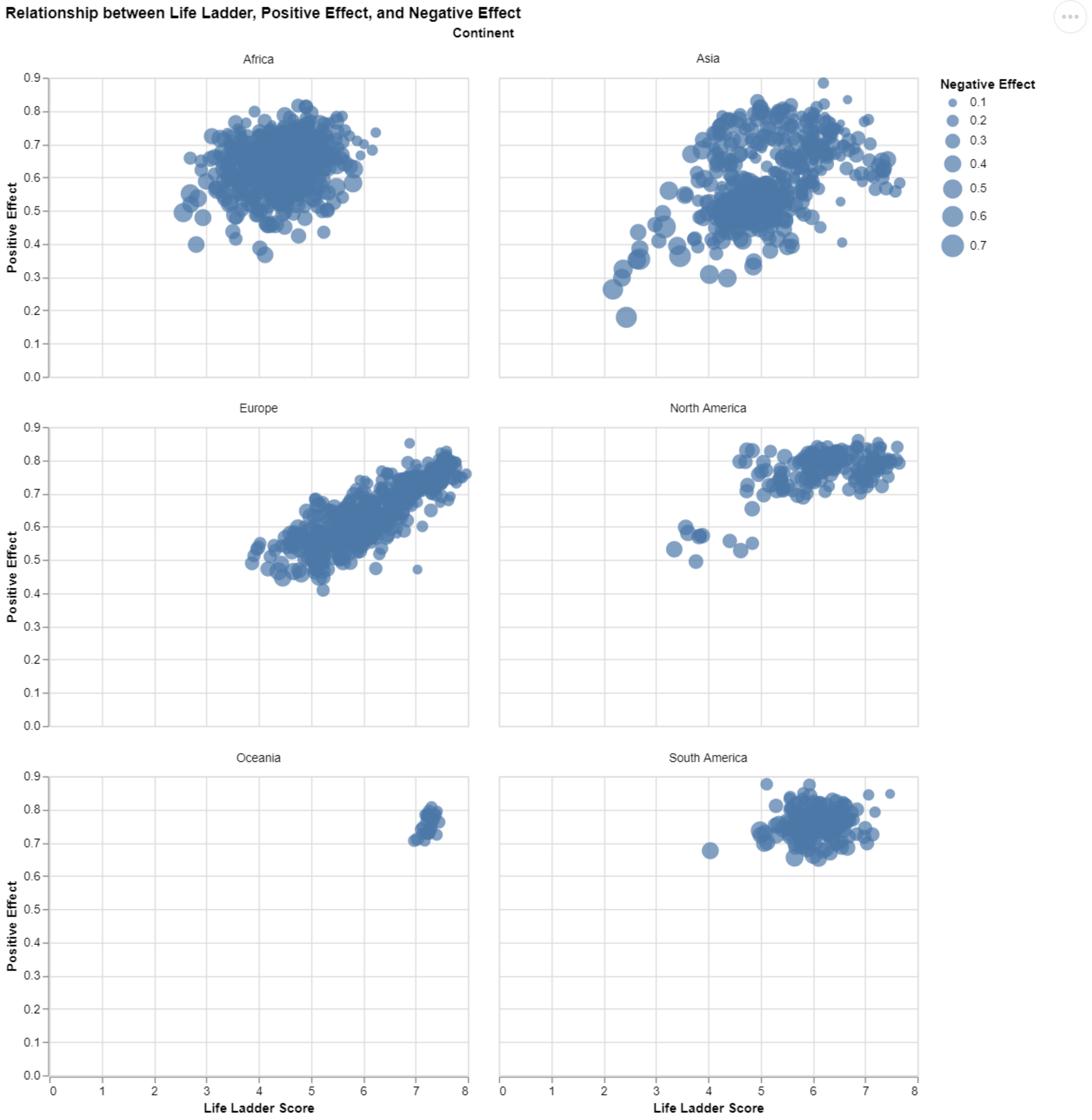
```
Continent
Africa          -1.936
Asia            -5.159
Europe           4.891
North America   -0.894
Oceania         -0.127
South America   -0.493
dtype: float64
```

```
score = clean_data.groupby(['year', 'Continent'])[['Life Ladder']].sum().reset_index()

pivot_data = score.pivot(index='year', columns='Continent', values='Life Ladder')

plt.figure(figsize=(8, 5))
pivot_data.plot(kind='bar', stacked=True, colormap='viridis')

plt.title('Happiness Score Over Time for Each Continent')
plt.xlabel('Year')
plt.ylabel('Life ladder')
plt.legend(title='Continent', loc='upper left', bbox_to_anchor=(1, 1))
fig_8 = plt.show()
fig_8
```

```
<Figure size 800x500 with 0 Axes>
```



```
# scatterplot of Happiness Score vs Positive effect

fig_9 = alt.Chart(clean_data).mark_circle().encode(
    x=alt.X('Life Ladder:Q', title='Life Ladder Score'),
    x=alt.X('Life Ladder:Q', title='Life Ladder Score'),
    y=alt.Y('Positive affect:Q', title='Positive Effect'),
    size=alt.Size('Negative affect:Q', title='Negative Effect'),
    tooltip=['Country name:N', 'year:O'],  # Adding tooltip for more information on hover,
    facet=alt.Facet('Continent', columns=2)
).properties(
    width=350,
    height=250,
    title='Relationship between Life Ladder, Positive Effect, and Negative Effect'
)#.facet('Continent')

# Display the plot
fig_9
```

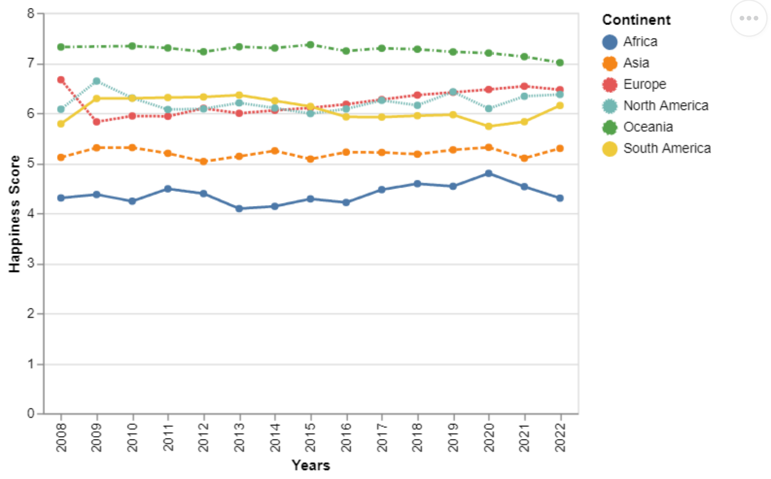


```
# Group the data and reset index to convert it into a DataFrame
grouped_data = clean_data.groupby(['year', 'Continent'], observed=False)['Life Ladder'].mean().reset_index()

# Group by continent and calculate the average Happiness Score for each continent and year

average_happiness = clean_data.groupby(['Continent', 'year'])['Life Ladder'].mean().reset_index()

# Pivot the DataFrame to have continents as columns and years as index
pivot_data = average_happiness.pivot(index='year', columns='Continent', values='Life Ladder')

# Calculate the change in Happiness Score for each continent over time
score_change = pivot_data.diff()
# Sum the changes over all years to get the overall improvement for each continent
```

```
overall_improvement = score_change.sum()
print(overall_improvement)

# Construct the plot using the grouped DataFrame
fig_11 = alt.Chart(grouped_data).mark_line(point=True).encode(
    x=alt.X('year:O', title='Years'),  # ":O" denotes ordinal data
    y=alt.Y('Life Ladder:Q', title='Happiness Score'),  # ":Q" denotes quantitative data
    color='Continent:N',  # ":N" denotes nominal (categorical) data
    strokeDash='Continent:N'  # Additional encoding for shape
).properties(
    width=600,  # Adjust the width as needed
    height=400  # Adjust the height as needed
)
```

```
Continent
Africa          -0.004320
Asia             0.180325
Europe          -0.199514
North America    0.294833
Oceania         -0.332000
South America    0.366611
dtype: float64
```

```
# Construct the plot using the grouped DataFrame
fig_11 = alt.Chart(grouped_data).mark_line(point=True).encode(
    x=alt.X('year:O', title='Years'),  # ":O" denotes ordinal data
    y=alt.Y('Life Ladder:Q', title='Happiness Score'),  # ":Q" denotes quantitative data
    color='Continent:N',  # ":N" denotes nominal (categorical) data
    strokeDash='Continent:N'  # Additional encoding for shape
).properties(
    width=400,  # Adjust the width as needed
    height=300  # Adjust the height as needed
)

# Display the plot
fig_11
```

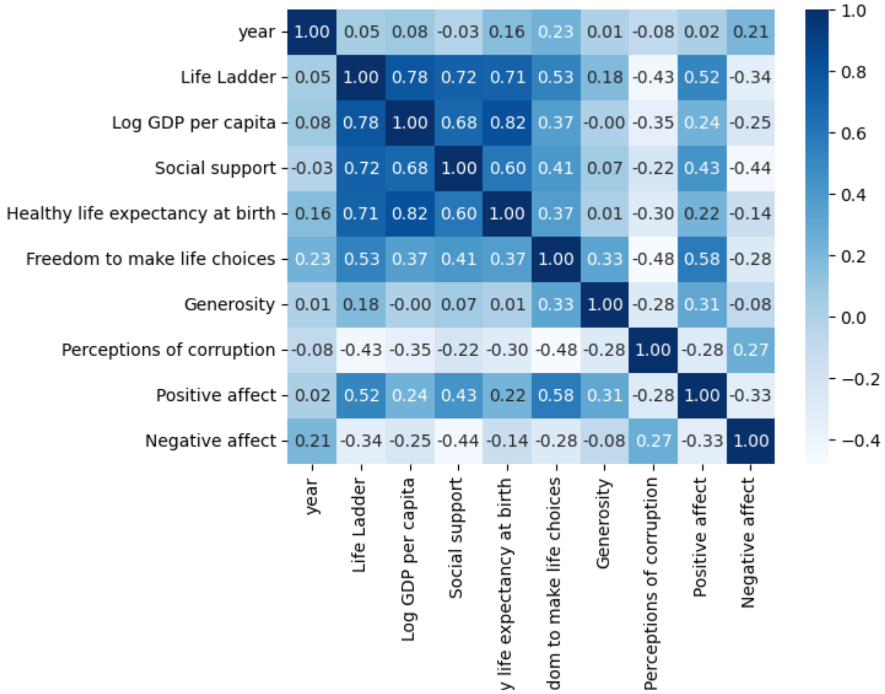


Question 3:

What are the top factors that affect the Happiness Score?

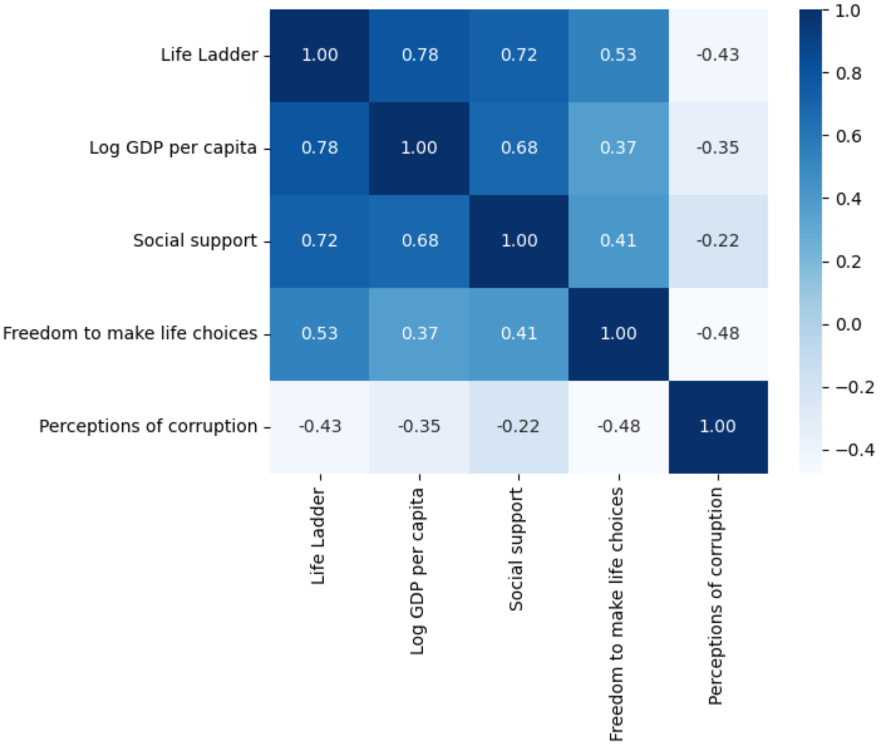
```
## Create a correlation matrix analyzing the relation of all the variables
#corr_matrix_all = data.loc[:,].corr(numeric_only = True)
#sns.heatmap(corr_matrix_all, annot=True, cmap="Blues", fmt=".2f")
```

```
<Axes: >
```



```
## Create a correlation matrix analyzing the relation of just the variables of interest
corr_matrix = data.loc[:,['Life Ladder','Log GDP per capita','Social support','Freedom to make life choices','Perceptions of corruption']].c
sns.heatmap(corr_matrix, annot=True, cmap="Blues", fmt=".2f")
```

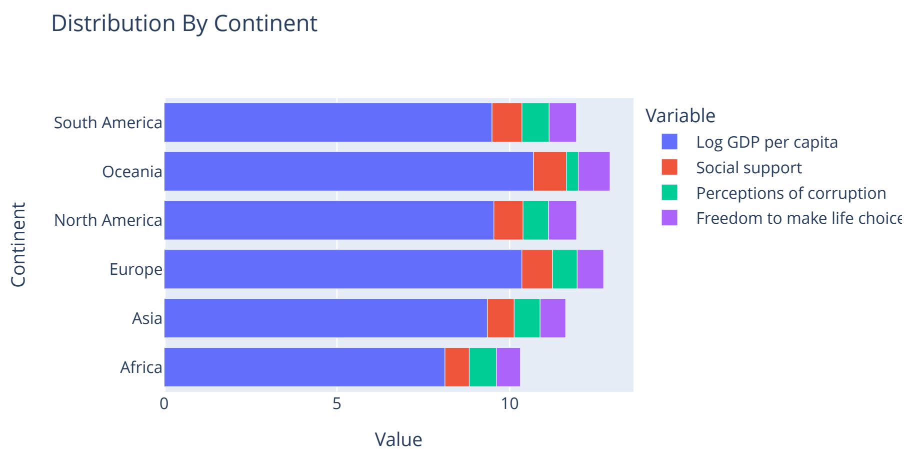
```
<Axes: >
```



```
# group by continent and find the mean of each explanatory variable
continent_dist = clean_data.groupby('Continent')[['Log GDP per capita', 'Social support','Perceptions of corruption', 'Freedom to make life
continent_dist
```

	Log GDP per capita	Social support	Perceptions of corruption	Freedom to make life choices
Continent				
Africa	8.125040	0.705284	0.785575	0.694509
Asia	9.354766	0.775917	0.749425	0.739515
Europe	10.351315	0.890487	0.712017	0.765028

```
# create a bar plot for each continent based on the above grouped mean variables per continent
fig_13 = px.bar(continent_dist.reset_index().melt(id_vars=['Continent'], var_name='Variable', value_name='Value'),
    y="Continent", x="Value", color="Variable",
    title="Distribution By Continent",
            height = 400,
            width = 700
)

fig_13.show()
```



Linear Regression

Since all of our predictor variables are quantitative and continuous, minimal preprocessing is required to begin fitting our model.

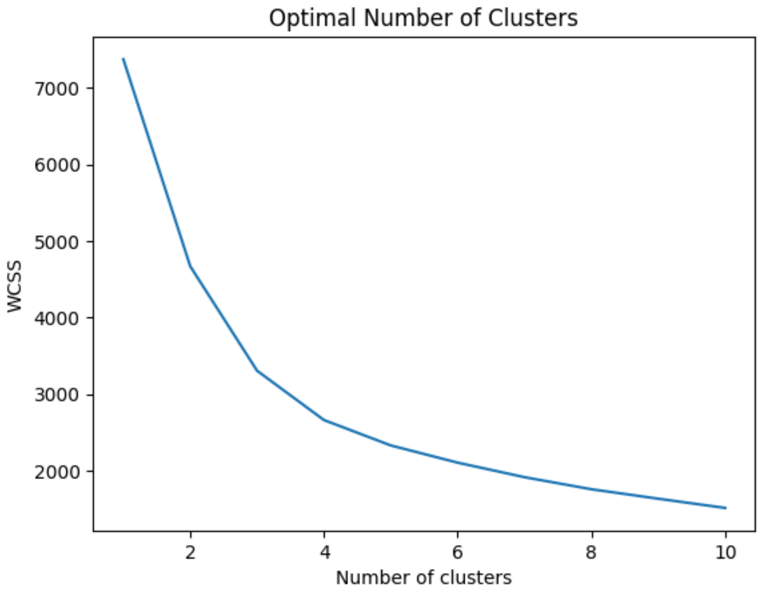
$$\beta_{Happiness} = \beta_0 + \beta_{Economy} + \beta_{Social\ support} + \beta_{Freedom} + \beta_{Corruption}$$

```
# select relevant variables with higher correlation from matrix
factor = clean_data[['Log GDP per capita', 'Social support','Freedom to make life choices','Perceptions of corruption']]

# add intercept of explanatory variable matrix
X = factor

# Add constant to independent variables
X = sm.add_constant(X)

# response variable matrix
y = clean_data['Life Ladder']


# Fit regression model by Ordinary Least Squares model
mdl = sm.OLS(y, X)
rslt = mdl.fit()

# Get parameter estimates and standard errors
coef_tbl = pd.DataFrame({'Coefficient': rslt.params, 'Standard Error': rslt.bse})

# Add error variance estimate
coef_tbl.loc['Error Variance'] = rslt.mse_resid

# Display the result
coef_tbl
```

	Coefficient	Standard Error
const	-1.725829	0.173588
Log GDP per capita	0.458935	0.017368
Social support	2.801337	0.162523
Freedom to make life choices	1.517871	0.119206
Perceptions of corruption	-0.669525	0.086883
Error Variance	0.350248	0.350248

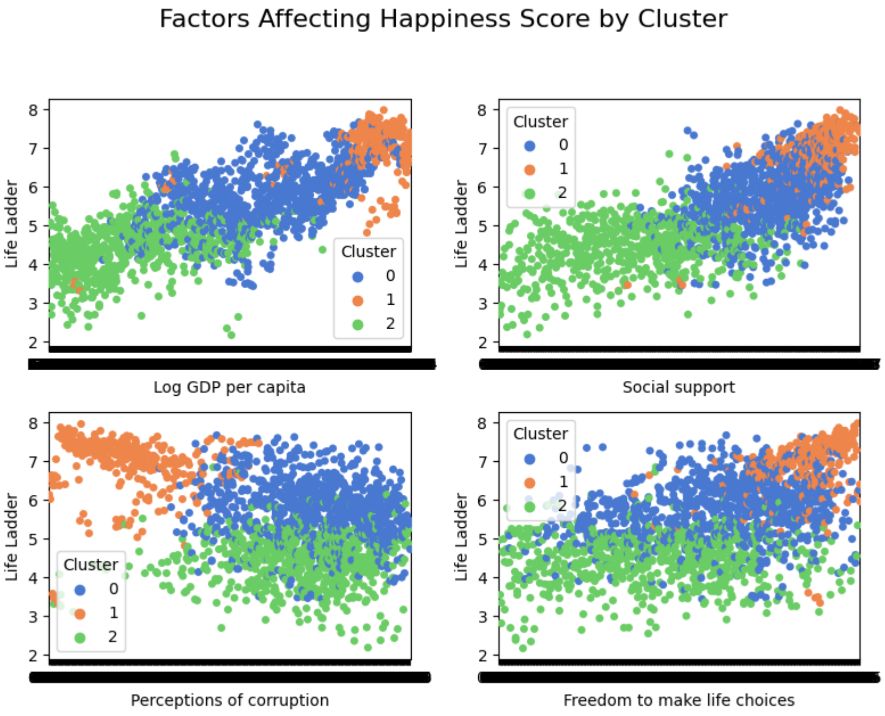
K-Means Cluster Analysis

```
# select the columns for clustering
X = clean_data[['Log GDP per capita', 'Social support', 'Perceptions of corruption', 'Freedom to make life choices']]

# standardize data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# find the optimal number of clusters by looping through 11 clusters and finding
# the correpsonding wcss
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

# label and plot
plt.plot(range(1, 11), wcss)
plt.title('Optimal Number of Clusters')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
fig_14 =plt.show()
fig_14
```



```
# apply k-means clustering
kmeans = KMeans(n_clusters=3, init='k-means++', random_state=42)
kmeans.fit(X_scaled)
clean_data['Cluster'] = kmeans.labels_

# Choose variables to use and a 2 x 2 plot
variables = ['Log GDP per capita', 'Social support',
             'Perceptions of corruption', 'Freedom to make life choices']
fig, axes = plt.subplots(2, 2, figsize=(8, 6))

# Loop through each variable and creating the corresponding line plots for each variable
for var, ax in zip(variables, axes.flatten()):
```

```
for var, ax in zip(variables, axes.flatten()):
    sns.stripplot(x=clean_data[var], y=clean_data["Life Ladder"], hue=clean_data['Cluster'], ax=ax, jitter=True, dodge=True,palette="muted")
    ax.set_xlabel(var)
    ax.set_ylabel("Life Ladder")

# add title and display the plot
plt.suptitle('Factors Affecting Happiness Score by Cluster', fontsize=16, y=1.05)
plt.tight_layout()
fig_15 = plt.show()
fig_15
```



```
cluster_groups = clean_data.groupby(['Cluster'])['Log GDP per capita', 'Social support',
             'Perceptions of corruption', 'Freedom to make life choices'].mean().reset_index()
cluster_groups
```

	Cluster	Log GDP per capita	Social support	Perceptions of corruption	Freedom to make life choices
0	0	9.780113	0.860711	0.821410	0.762189
1	1	10.758706	0.924802	0.404703	0.895608
2	2	8.199166	0.680558	0.789267	0.663086

```
# visual depiction of clustered grouping

# bar plot of the correpsonding explanatory variables for each cluster
fig_16 = px.bar(cluster_groups.#reset_index().
                melt(id_vars=['Cluster'], var_name='Variable', value_name='Value'),
    y="Cluster", x="Value", color="Variable",
    color_discrete_sequence=["#2c3e50", "#808b96","#273746","#1b4f72","#5499C7","#A9CCE3"],orientation='h',
    title="Cluster Characteristics")

# show figure
fig_16.show()
```