PSTAT100_FinalProject_WHR

December 14, 2023

[1]:
```
## PSTAT 100 FINAL PROJECT: WORLD HAPPINESS REPORT
## Anthony Clark, Samuel Lepoutre, Samuel Metta, Ajith Nair

import pandas as pd
import numpy as np
import altair as alt
import warnings
from vega_datasets import data # used for geomapping
import statsmodels.api as sm



# exporting graphics
alt.data_transformers.disable_max_rows()
alt.renderers.enable('mimetype')
warnings.filterwarnings('ignore')

# data import
whr = pd.read_csv("data/whr-2023.csv")
```

1 Background

1.1 Dataset Description

https://worldhappiness.report/data/

The World Happiness Report (WHR) dataset is a collection of years of Gallup World Poll survey data that reflects the happiness and well-being of people and is based on answers to main life questions that are either binary response or on a scale of 1-10. The set is evaluated on mainly the opinions of those they interview, with the only "measurable" attribute being log GDP per capita. Summarized on the WHR website, the purpose of this dataset is to "demand more attention to happiness and well-being as a criteria for government policy".

Name	Variable description
Country name	Country Name
year	Year of report
Life Ladder	National average "happiness score", ranging from 0 (lowest) to 10 (highest)

Name	Variable description
Log GDP per capita	Log transformation of the GDP per capita
Social Support	National average of a binary response on whether an individual has "someone to count on in times of trouble"
Healthy life Expectancy	Life expectancy in years, extracted from the World Health Organization data repo
Freedom to make life choices	National average of responses to the question of whether one is satisfied with their "freedom to choose what to do with their life"
Generosity	The residual of regressing national average of response to the question "Have you donated money to charity in the past month?"
Perception of corruption	National average of the binary responses to whether corruption is widespread in government and businesses.
Positive affect	Measured averages to questions on whether or not an individual "smiled, enjoyed, or learned a lot yesterday"
Negative affect	Measured averages to questions on whether or not an individual was "worried, angry, or sad a lot yesterday"

2 Tidying Data

2.1 Renaming Columns

[2]:
```
whr_clean = whr.rename(columns = {
    "Country name": "Country",
    "Life Ladder": "Score",
    "Log GDP per capita": "GDP",
    "Social support": "Support",
    "Healthy life expectancy at birth": "Lifespan",
    "Freedom to make life choices": "Freedom",
    "Perceptions of corruption": "Corruption",
    "Positive affect": "Positive",
    "Negative affect": "Negative"})
whr_clean.head()
```

[2]:

	Country	year	Score	GDP	Support	Lifespan	Freedom	Generosity	\
0	Afghanistan	2008	3.724	7.350	0.451	50.5	0.718	0.168	
1	Afghanistan	2009	4.402	7.509	0.552	50.8	0.679	0.191	
2	Afghanistan	2010	4.758	7.614	0.539	51.1	0.600	0.121	
3	Afghanistan	2011	3.832	7.581	0.521	51.4	0.496	0.164	
4	Afghanistan	2012	3.783	7.661	0.521	51.7	0.531	0.238	

	Corruption	Positive	Negative
0	0.882	0.414	0.258
1	0.850	0.481	0.237
2	0.707	0.517	0.275
3	0.731	0.480	0.267
4	0.776	0.614	0.268

2.2 Missing Data Imputation

[3]:
```
# Imputing missing data...

# Defining unique countries
unique_countries = whr_clean["Country"].unique()

# Looping for each column
for col in whr_clean.columns.drop("Country"):
    # Filtering to columns with missing values only
    if (whr_clean[col].isna().any() == True):
        # Filtering for each country
        for country in unique_countries:
            # Filtering to the individual country
            filtered = whr_clean["Country"] == country
            # Collecting country-specific and column-specific mean
            local_mean = whr_clean.loc[filtered, col].mean()
            # Function to check whether mean is NaN
            check_mean = (local_mean > 0) or (local_mean <= 0)
            # checking to make sure a mean value exists for the specific country
            if check_mean == True:
                # Imputation based on country-/column-specific mean
                whr_clean.loc[filtered, col] = whr_clean.loc[filtered, col].
 ↪fillna(local_mean)
### Note: elif statement optional based on whether to impute country-specific␣
 ↪data
### with the average of ALL countries; could lead to inaccurate conclusions
#             elif check_mean == False:
#                 # if no valid mean, imputing individual country based on␣
 ↪entire dataset (global mean)
#                 # NOTE: We will not impute corruption for a country based on␣
 ↪other countries' data here
#                 if col != "Corruption":
#                     global_mean = whr_clean[col].mean()
#                     whr_clean.loc[filtered, col] = whr_clean.loc[filtered,␣
 ↪col].fillna(global_mean)

# whr_clean.isna().sum() -- note that some missing data remains in order to␣
 ↪avoid inaccurate data analysis
```

```
# (e.g., imputing corrpution data for China from various other countries could␣
 ↪create some interesting data implications)
```

3 Question of Interest

Has the geographic distribution of happiness changed in the last 18 years? If so, why have levels of happiness changed, and what can we attribute these changes to?

The goal of our analysis is to determine why happiness varies geographically by analyzing several predictors in the data set and their impact not only on various countries in the dataset, but also on the data set as a whole. Through this analysis, we hope to gather valuable information that countries with lower happiness scores can apply in an attempt to increase the average level of life satisfaction within their national population.

3.1 Does happiness vary geographically?

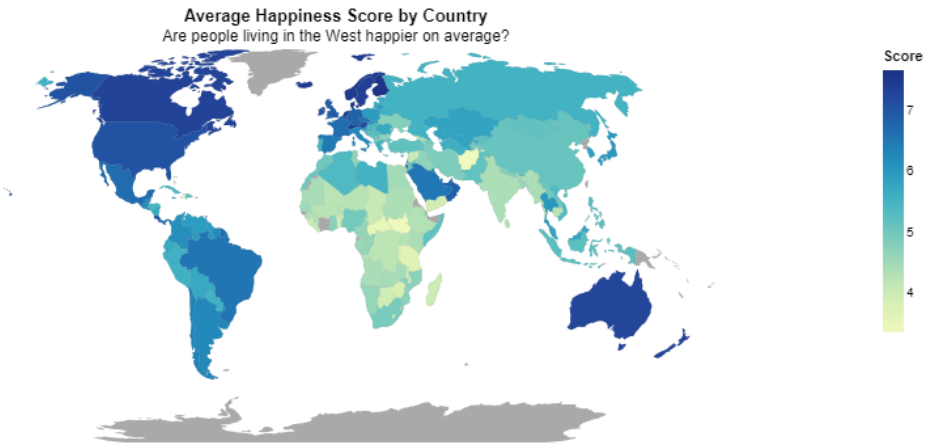
[4]:
```
### Global Happiness Map code

# grouping data by country
country_groups = whr_clean.groupby("Country")

# calculating average happiness score over the years
avg_country_score = country_groups["Score"].mean().to_frame().reset_index()

# importing vega country data
countries = alt.topo_feature(data.world_110m.url, 'countries')

# importing country ISO data for mapping
# SOURCE: https://github.com/lukes/ISO-3166-Countries-with-Regional-Codes/blob/
 ↪master/all/all.csv
iso_url = "https://raw.githubusercontent.com/lukes/
 ↪ISO-3166-Countries-with-Regional-Codes/master/all/all.csv"
iso_df = pd.read_csv(iso_url)

# dictionary of country names to match WHR dataset for ID mapping
updated_country_names = {
    "Bolivia (Plurinational State of)": "Bolivia",
    "United States of America": "United States",
    "Congo": "Congo (Brazzaville)",
    "Congo, Democratic Republic of the": "Congo (Kinshasa)",
    "Hong Kong": "Hong Kong S.A.R. of China",
    "Iran (Islamic Republic of)": "Iran",
    # NOTE: Kosovo has no code - will not be mapped
    "Lao People's Democratic Republic": "Laos",
    "Moldova, Republic of": "Moldova",
    "Russian Federation": "Russia",
    # NOTE: Somaliland region has no code - will not be mapped
```

```
    "Korea, Republic of": "South Korea",
    "Palestine, State of": "State of Palestine",
    "Syrian Arab Republic": "Syria",
    "Tanzania, United Republic of": "Tanzania",
    "Turkey": "Turkiye",
    "United Kingdom of Great Britain and Northern Ireland": "United Kingdom",
    "Venezuela (Bolivarian Republic of)": "Venezuela",
    "Viet Nam": "Vietnam"}

# tidying and merging country ID and country's mean happiness
iso_df_fixed = iso_df[["name", "country-code"]].rename(
    columns = {
        "name": "Country",
        "country-code": "id"}
).replace(updated_country_names)
# merging...
avg_country_score_with_ID = avg_country_score.merge(iso_df_fixed)

# creating base for world map
map_base = alt.Chart(countries).mark_geoshape(color = "darkgrey").project(
    "naturalEarth1").properties(width = 800)

## Overall average happiness map
avg_mean_happiness = alt.Chart(countries).mark_geoshape().encode(
    color = alt.Color("Score:Q")).transform_lookup(
    lookup = "id",
    from_=alt.LookupData(avg_country_score_with_ID, "id", ["Score"])
    ).project("naturalEarth1").properties(width = 800)
# combining with base
overall_avg_happiness_map = (map_base + avg_mean_happiness).properties(
    title = alt.Title("Average Happiness Score by Country",
    subtitle = "Are people living in the West happier on average?"))

## Overall happiness delta (change) map

# setting up data:
# minimum score
country_min_score = country_groups.year.min().to_frame().reset_index(
).merge(whr_clean).rename(columns = {"Score": "Oldest Score", "year": "Old␣
 ↪Year"})[["Country", "Old Year", "Oldest Score"]]
# maximum score
country_max_score = country_groups.year.max().to_frame().reset_index(
).merge(whr_clean).rename(columns = {"Score": "Newest Score", "year": "New␣
 ↪Year"})[["Country", "New Year", "Newest Score"]]
# merging scores
country_minmax = country_min_score.merge(country_max_score)
# calculating change over time
```

```
country_minmax["Delta"] = -(country_minmax["Oldest Score"].
 ↪sub(country_minmax["Newest Score"]))
country_minmax_with_ID = country_minmax.merge(iso_df_fixed)

# mapping:
delta_map = alt.Chart(countries).mark_geoshape().encode(
    color = alt.Color("Delta:Q").scale(scheme = "redblue", domainMid = 0)).
 ↪transform_lookup(
    lookup = "id",
    from_=alt.LookupData(country_minmax_with_ID, "id", ["Delta"])
    ).project("naturalEarth1").properties(width = 800)
# combining map with base
happiness_delta_map = (map_base + delta_map).properties(
    title = alt.Title("Happiness Delta by Country",
    subtitle = "How has happiness changed in each country from the oldest to␣
 ↪most recent year it was recorded?"))
```

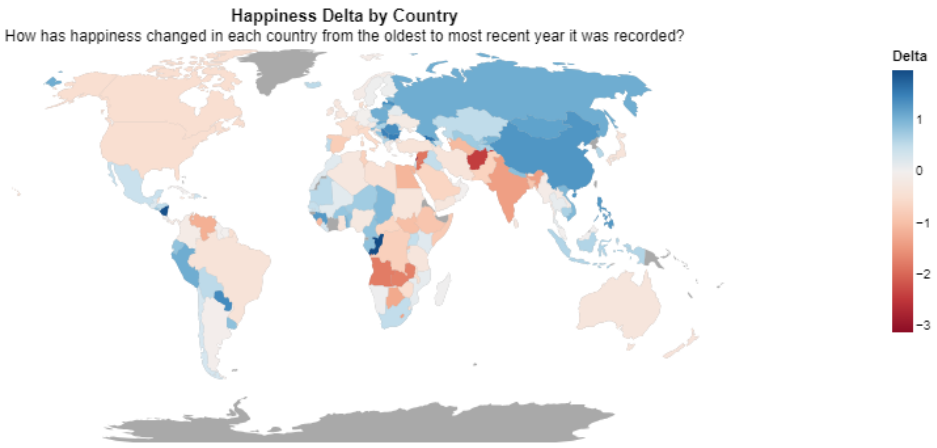
[5]:
```
overall_avg_happiness_map
```

[5]:



[6]:
```
happiness_delta_map
```

[6]:



[7]:
```
country_minmax_with_ID.sort_values("Delta", ascending = False)
```

[7]:

	Country	Old Year	Oldest Score	New Year	Newest Score	Delta	id
32	Congo (Brazzaville)	2008	3.820	2022	5.805	1.985	178
105	Nicaragua	2006	4.460	2022	6.392	1.932	558
52	Georgia	2006	3.675	2022	5.293	1.618	268
20	Bulgaria	2007	3.844	2022	5.378	1.534	100
124	Serbia	2007	4.750	2021	6.245	1.495	688
..	…	…	…	…	…	…	…
3	Angola	2011	5.589	2014	3.795	-1.794	24
140	Syria	2008	5.323	2015	3.462	-1.861	760
73	Jordan	2005	6.295	2022	4.356	-1.939	400
0	Afghanistan	2008	3.724	2022	1.281	-2.443	4
80	Lebanon	2005	5.491	2022	2.352	-3.139	422

[161 rows x 7 columns]

3.1.1 Geographic Analysis

The **Happiness Delta by Country** figure and corresponding table reveal quite insightful data on the change in global happiness over the past 15 or so years. According to the data, the Republic of the Congo (Brazzaville) has experienced the largest increase in happiness in the data set, with a 1.985 increase in happiness score from 2008 to 2022. On the other hand, the country with the largest *decrease* in happiness score is Lebanon, which experienced a decrease in happiness score by 3.139 points from 2005 to 2022. Considering the ongoing crises in Lebanon, this large decrease in overall happiness seems reasonable. Additionally, it seems like despite Western countries' generally high average happiness score in recent years, as showcased in the **Average Happiness Score by Country** figure, their happiness score appears to be decreasing over time; examples include the U.S., Canada, and the United Kingdom.

3.2 Why have levels of happiness changed and what can we attribute these changes to?

3.2.1 Correlation Matrix

[8]:
```
happy_corr = whr_clean[['Score', 'GDP', 'Support', 'Lifespan', 'Freedom',␣
 ↪'Generosity',
                        'Corruption', 'Positive', 'Negative']]
happy_corr = happy_corr.corr()
happy_corr
```

[8]:

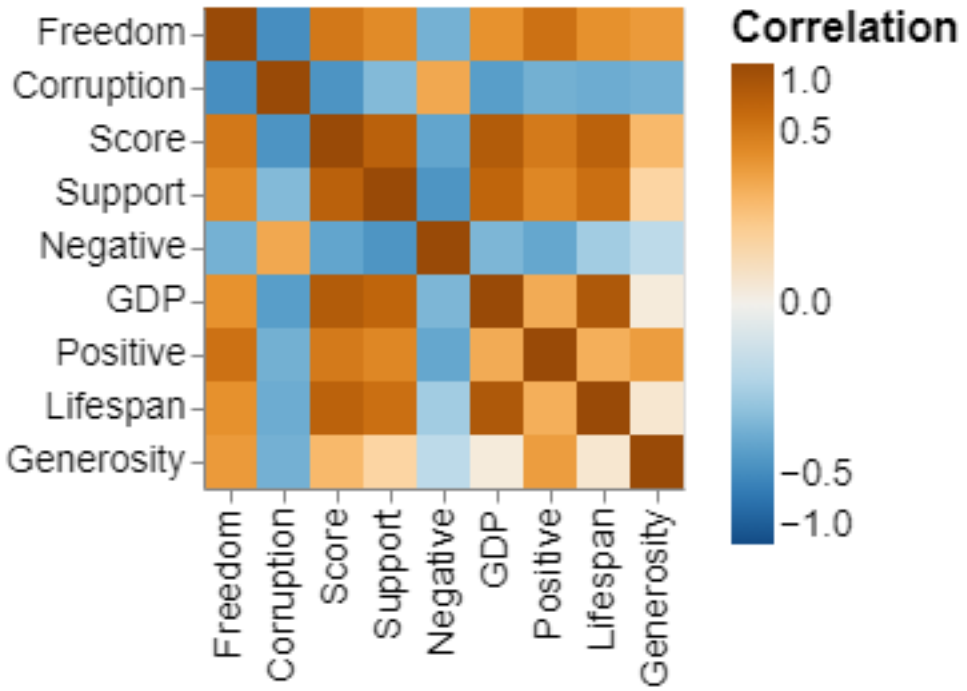
	Score	GDP	Support	Lifespan	Freedom	Generosity	\
Score	1.000000	0.784716	0.722054	0.714336	0.529458	0.182302	
GDP	0.784716	1.000000	0.682068	0.818523	0.365758	0.004244	
Support	0.722054	0.682068	1.000000	0.598180	0.408843	0.070864	
Lifespan	0.714336	0.818523	0.598180	1.000000	0.370864	0.011911	
Freedom	0.529458	0.365758	0.408843	0.370864	1.000000	0.321412	
Generosity	0.182302	0.004244	0.070864	0.011911	0.321412	1.000000	
Corruption	-0.446241	-0.383950	-0.230294	-0.301051	-0.482434	-0.278804	
Positive	0.517589	0.240776	0.432617	0.225065	0.577521	0.306648	
Negative	-0.339696	-0.249064	-0.440981	-0.146700	-0.275800	-0.078627	

	Corruption	Positive	Negative
Score	-0.446241	0.517589	-0.339696
GDP	-0.383950	0.240776	-0.249064
Support	-0.230294	0.432617	-0.440981
Lifespan	-0.301051	0.225065	-0.146700
Freedom	-0.482434	0.577521	-0.275800
Generosity	-0.278804	0.306648	-0.078627
Corruption	1.000000	-0.283000	0.254964
Positive	-0.283000	1.000000	-0.329495
Negative	0.254964	-0.329495	1.000000

3.2.2 Correlation Plot

[9]:
```
corr_mx_long = happy_corr.reset_index().rename(
    columns = {'index': 'row'}
).melt(
    id_vars = 'row',
    var_name = 'col',
    value_name = 'Correlation'
)

# visualize
alt.Chart(corr_mx_long).mark_rect().encode(
    x = alt.X('col', title = '', sort = {'field': 'Correlation', 'order':␣
 ↪'ascending'}),
    y = alt.Y('row', title = '', sort = {'field': 'Correlation', 'order':␣
 ↪'ascending'}),
    color = alt.Color('Correlation',
                      scale = alt.Scale(scheme = 'blueorange',
                                        domain = (-1, 1),
                                        type = 'sqrt'),
                      legend = alt.Legend(tickCount = 5))
).configure_axis(
    labelFontSize = 14
).configure_legend(
    labelFontSize = 14,
    titleFontSize = 16
)
```

[9]:



3.2.3 Correlation Findings

We would like narrow our focus to predictors that appear to significantly affect happiness score in order to determine why changes in happiness score have occurred in various countries throughout the past 18 years. Through our correlation analysis, we have determined that signficant factors will be defined by a correlation between the predictor variable and `Score` that is either greater than 0.4 or less than −0.4. This leaves us with the following signficant factors, in order from highest to lowest correlation: * Significantly Positive (≥ 0.4): `GDP`, `Support`, `Lifespan`, `Freedom`, `Positive Affect` * Significantly Negative (≤ -0.4): `Corruption`

In order to analyze how these factors are related to changes in happiness, we will proceed with exploratory analysis of the data.

4 Exploratory Analysis

4.1 Which Year experienced the best worldwide happiness rating?

[10]:
```
whr_clean['year'] = pd.to_datetime(whr_clean['year'], format='%Y')

# Calculate the maximum and minimum happiness scores for each year worldwide
max_scores_by_year = whr_clean.groupby('year')['Score'].max()
min_scores_by_year = whr_clean.groupby('year')['Score'].min()

# Find the corresponding years for the maximum and minimum scores
best_years_worldwide = max_scores_by_year.idxmax().year
worst_years_worldwide = min_scores_by_year.idxmin().year

print(f"The year with the best worldwide happiness rating:␣
 ↪{best_years_worldwide}")
print(f"The year with the worst worldwide happiness rating:␣
 ↪{worst_years_worldwide}")
```

```
The year with the best worldwide happiness rating: 2005
The year with the worst worldwide happiness rating: 2022
```

4.2 Which Countries had the best / worst happiness rating worldwide by year?

[11]:
```
best_countries_by_year = whr_clean.loc[whr_clean.groupby('year')['Score'].
 ↪idxmax()][['year', 'Country', 'Score']]
worst_countries_by_year = whr_clean.loc[whr_clean.groupby('year')['Score'].
 ↪idxmin()][['year', 'Country', 'Score']]

# Display the results
print("\nCountry with the best recorded happiness score for each year:")
print(best_countries_by_year)
```

```
print("\nCountry with the worst recorded happiness score for each year:")
print(worst_countries_by_year)
```

```
Country with the best recorded happiness score for each year:
           year      Country  Score
505  2005-01-01      Denmark  8.019
623  2006-01-01      Finland  7.672
506  2007-01-01      Denmark  7.834
507  2008-01-01      Denmark  7.971
508  2009-01-01      Denmark  7.683
509  2010-01-01      Denmark  7.771
510  2011-01-01      Denmark  7.788
1868 2012-01-01  Switzerland  7.776
334  2013-01-01       Canada  7.594
513  2014-01-01      Denmark  7.508
1477 2015-01-01       Norway  7.603
631  2016-01-01      Finland  7.660
632  2017-01-01      Finland  7.788
633  2018-01-01      Finland  7.858
634  2019-01-01      Finland  7.780
635  2020-01-01      Finland  7.889
636  2021-01-01      Finland  7.794
637  2022-01-01      Finland  7.729

Country with the worst recorded happiness score for each year:
           year                    Country  Score
1980 2005-01-01                    Turkiye  4.719
1950 2006-01-01                       Togo  3.202
2183 2007-01-01                   Zimbabwe  3.280
1951 2008-01-01                       Togo  2.808
1919 2009-01-01                   Tanzania  3.408
1920 2010-01-01                   Tanzania  3.229
1952 2011-01-01                       Togo  2.936
1882 2012-01-01                      Syria  3.164
1883 2013-01-01                      Syria  2.688
1953 2014-01-01                       Togo  2.839
1126 2015-01-01                    Liberia  2.702
347  2016-01-01  Central African Republic  2.693
9    2017-01-01                Afghanistan  2.662
10   2018-01-01                Afghanistan  2.694
11   2019-01-01                Afghanistan  2.375
1115 2020-01-01                    Lebanon  2.634
1116 2021-01-01                    Lebanon  2.179
13   2022-01-01                Afghanistan  1.281
```

There are several years where Finland has the best recorded happiness score, while Lebanon has

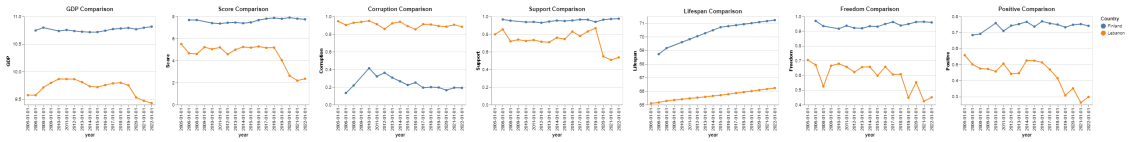
the worst happiness differential from 2005 to 2022 (evident from the Happiness Delta by Country Map). Since there is an argument to be made for Finland and Lebanon having the respective best and worst happiness levels in recent years, let's further anaylze how these two countries differ from factor to factor to see if this difference in happiness can be explained by predictors in the data set.

4.3 Finland vs. Lebanon

[12]:
```
fin_leb = whr_clean[whr_clean['Country'].isin(['Finland', 'Lebanon'])]
fin_leb
fin_leb['year'] = fin_leb['year'].astype(str)

gdp_compare = alt.Chart(fin_leb).mark_line(point = True).encode(
    x = 'year:O',
    y = alt.Y('GDP:Q', scale = alt.Scale(zero = False)),
    color = 'Country:N'
).properties(
    width = 300,
    height = 200,
    title = 'GDP Comparison'
)

score_compare = alt.Chart(fin_leb).mark_line(point = True).encode(
    x = 'year:O',
    y = alt.Y('Score:Q'),
    color = 'Country:N'
).properties(
    width = 300,
    height = 200,
    title = 'Score Comparison'
)

corruption_compare = alt.Chart(fin_leb).mark_line(point = True).encode(
    x = 'year:O',
    y = alt.Y('Corruption:Q'),
    color = 'Country:N'
).properties(
    width = 300,
    height = 200,
    title = 'Corruption Comparison'
)

support_compare = alt.Chart(fin_leb).mark_line(point = True).encode(
    x = 'year:O',
    y = alt.Y('Support:Q'),
    color = 'Country:N'
).properties(
    width = 300,
```

```
    height = 200,
    title = 'Support Comparison'
)

lifespan_compare = alt.Chart(fin_leb).mark_line(point = True).encode(
    x = 'year:O',
    y = alt.Y('Lifespan:Q', scale = alt.Scale(zero = False)),
    color = 'Country:N'
).properties(
    width = 300,
    height = 200,
    title = 'Lifespan Comparison'
)

lifespan_compare = alt.Chart(fin_leb).mark_line(point = True).encode(
    x = 'year:O',
    y = alt.Y('Lifespan:Q', scale = alt.Scale(zero = False)),
    color = 'Country:N'
).properties(
    width = 300,
    height = 200,
    title = 'Lifespan Comparison'
)

freedom_compare = alt.Chart(fin_leb).mark_line(point = True).encode(
    x = 'year:O',
    y = alt.Y('Freedom:Q', scale = alt.Scale(zero = False)),
    color = 'Country:N'
).properties(
    width = 300,
    height = 200,
    title = 'Freedom Comparison'
)

positive_compare = alt.Chart(fin_leb).mark_line(point = True).encode(
    x = 'year:O',
    y = alt.Y('Positive:Q', scale = alt.Scale(zero = False)),
    color = 'Country:N'
).properties(
    width = 300,
    height = 200,
    title = 'Positive Comparison'
)
#score_compare
gdp_compare | score_compare | corruption_compare | support_compare |␣
 ↪lifespan_compare | freedom_compare | positive_compare
```

[12]:



4.3.1 Finland vs. Lebanon Analysis

It appears that our comparison of several predictors explains a significant amount of the difference in happiness `Score` between Finland and Lebanon. Finland has higher scores for `GDP`, `Support`, `Freedom`, and `Positive`, with a far lower `Corruption` rating, while Lebanon is the opposite. Since Finland is also consistently happier than Lebanon in all of the years where both countries have a recorded score, it is reasonable to conclude that the differences between the two countries in predictors such as `GDP`, `Support`, `Freedom`, and `Positive` affect contribute largely to the difference between the two overall happiness scores.

Additionally, analyzing the slopes of each graph for each country will show us why happiness could be decreasing or increasing over time. We observe that Lebanese `Support`, `Freedom` and `Positive` affect have a negative trend over time while their Finnish counterparts have a slope of nearly zero. Contrary to the numerous negative predictor trends, `Lifespan` in Lebanon has indeed increased; however, this is at a slower pace than the increase of lifespan in Finland. Although we know from the data that Finland has, on average, a higher happiness score than Lebanon, the positive and negative trends within the Finland vs. Lebanon predictor graphs in Section 4.3 likely reveal why Lebanon's happiness score has diminished over time.

5 Regression Analysis

In order to do regression analysis, there can be no missing data. Therefore, we will create a new dataframe with the missing data being imputed through averages. The difference between the imputing method above and this method, however, is that our initial method avoids using the global averages to prevent reaching inaccurate conclusions. However, in order to run our regression models, we determined that it would better to use global averages to impute missing data for countries that have no relevant data to average, such as the lack of corruption data for China. This allows us to fit models that will still have a significant level of viability in explaining our dataset without dropping several countries' worth of data.

[13]:
```
whr2 = whr_clean.copy()
unique_countries = whr2["Country"].unique()

for col in whr2.columns.drop("Country"):
    # Filtering to columns with missing values only
    if (whr2[col].isna().any() == True):
        # Filtering for each country
        for country in unique_countries:
            # Filtering to the individual country
            filtered = whr2["Country"] == country
            # Collecting country-specific and column-specific mean
```

```
            local_mean = whr2.loc[filtered, col].mean()
            # Function to check whether mean is NaN
            check_mean = (local_mean > 0) or (local_mean <= 0)
            # If/else determined by whether the mean is not NaN
            if check_mean == True:
                # Imputation based on country-/column-specific mean
                whr2.loc[filtered, col] = whr2.loc[filtered, col].
 ↪fillna(local_mean)
            elif check_mean == False:
                # if no valid mean, imputing individual country based on entire␣
 ↪dataset (global mean)
                global_mean = whr2[col].mean()
                whr2.loc[filtered, col] = whr2.loc[filtered, col].
 ↪fillna(global_mean)

#whr2.isnull().sum()    # 0 -- no missing data
```

5.1 First Model: Score as a model of GDP, Support, Lifespan, Freedom, Positive Affect, and Corruption

[14]:
```
# regression
# idea, model score as a function of all other significant factors

whr_reg = whr2.drop(columns = ['Country', 'year', 'Score', 'Generosity',␣
 ↪'Negative'])
x = sm.tools.add_constant(whr_reg)
y = whr2.Score

mlr = sm.OLS(endog = y, exog = x)
rslt = mlr.fit()

coef_tbl = pd.DataFrame(
    {'estimate':rslt.params.values, 'standard error': np.sqrt(rslt.cov_params().
 ↪values.diagonal())},
    index = x.columns
)
coef_tbl.loc['error variance', 'estimate'] = rslt.scale

# display
coef_tbl

# Support and Positive have the largest influence on score by a substantial␣
 ↪margin although standard error is large
# Corruption and Negative have a negative effect on happiness, as expected
# lifespan doesn't seem to have an effect on happiness
```

```
# rslt.rsquared ~= 0.7716; about 77.16% of the variance is explained by these␣
 ↪factors.
# If we add Generosity and Negative, r^2 increases to 77.4%, showing that they␣
 ↪have little to no effect on Score
```

[14]:

	estimate	standard error
const	-2.887638	0.158449
GDP	0.369982	0.019616
Support	1.983049	0.142304
Lifespan	0.028751	0.002934
Freedom	0.433091	0.113166
Corruption	-0.650101	0.073115
Positive	2.485814	0.140209
error variance	0.290098	NaN

5.2 Second Model: Score as a Model of Geographic Location by Continent

[15]:
```
#pip install country_converter # package that maps country to continent
```

[16]:
```
# categorizing countries by continent


whr_reg1 = whr_clean.copy()

import country_converter as coco

# Sample DataFrame with a 'Country' column
# Use country_converter to convert country names to continents
cc = coco.CountryConverter()
whr_reg1['Continent'] = whr_reg1['Country'].apply(lambda x: cc.convert(x,␣
 ↪to='continent'))

whr_reg1['Continent_Code'] = pd.factorize(whr_reg1['Continent'])[0]

asia = list(whr_reg1.loc[whr_reg1.Continent_Code == 0,:].Country.value_counts().
 ↪index)
europe = list(whr_reg1.loc[whr_reg1.Continent_Code == 1,:].Country.
 ↪value_counts().index)
africa = list(whr_reg1.loc[whr_reg1.Continent_Code == 2,:].Country.
 ↪value_counts().index)
americas = list(whr_reg1.loc[whr_reg1.Continent_Code == 3,:].Country.
 ↪value_counts().index)
australia = list(whr_reg1.loc[whr_reg1.Continent_Code == 4,:].Country.
 ↪value_counts().index)
```

```
north_am = ['United States', 'Canada', 'Mexico']
south_am = [country for country in americas if country not in north_am]
south_am

def categorize_americas(country):
    if country in north_am:
        return 'North America'
    elif country in south_am:
        return 'South America'
    elif country in asia:
        return 'Asia'
    elif country in europe:
        return 'Europe'
    elif country in africa:
        return 'Africa'
    elif country in australia:
        return 'Australia'



# Apply the custom mapping function
whr_reg1['Continent'] = whr_reg1['Country'].apply(categorize_americas)

# indicator variable
whr_reg1['Continent_Code'] = pd.factorize(whr_reg1['Continent'])[0]

# 0 = Asia, 1 = Europe, 2 = Africa, 3 = South America, 4 = Australia, 5 = North␣
 ↪America
whr_reg1.head()
#whr_reg1.Continent_Code.value_counts()
```

[16]:

	Country	year	Score	GDP	Support	Lifespan	Freedom	Generosity	Corruption	Positive	Negative	Continent	Continent_Code
0	Afghanistan	2008-01-01	3.724	7.350	0.451	50.5	0.718	0.168	0.882	0.414	0.258	Asia	0
1	Afghanistan	2009-01-01	4.402	7.509	0.552	50.8	0.679	0.191	0.850	0.481	0.237	Asia	0
2	Afghanistan	2010-01-01	4.758	7.614	0.539	51.1	0.600	0.121	0.707	0.517	0.275	Asia	0
3	Afghanistan	2011-01-01	3.832	7.581	0.521	51.4	0.496	0.164	0.731	0.480	0.267	Asia	0
4	Afghanistan	2012-01-01	3.783	7.661	0.521	51.7	0.531	0.238	0.776	0.614	0.268	Asia	0

[17]:

```
whr_reg2 = whr_reg1[['Continent_Code']]
indicators = pd.get_dummies(whr_reg2, columns=['Continent_Code'],␣
 ↪drop_first=False)
indicators = indicators.astype(int)

x1 = sm.tools.add_constant(indicators)
y1 = whr_clean.Score

# regression model of continent

mlr1 = sm.OLS(endog = y1, exog = x1)
rslt1 = mlr1.fit()

coef_tbl1 = pd.DataFrame(
    {'estimate':rslt1.params.values, 'standard error': np.sqrt(rslt1.
 ↪cov_params().values.diagonal())},
    index = x1.columns
)
coef_tbl1.loc['error variance', 'estimate'] = rslt1.scale

# display
coef_tbl1
# it appears that Asia, Australia, and North America have the largest positive␣
 ↪effects on happiness scores
# Africa has a negative effect on happiness scores
# rslt1.rsquared ~= 0.4627; About 46.27% of the variation is explained by the␣
 ↪continents
```

[17]:

	estimate	standard error
const	5.158885	0.028677
Continent_Code_0	0.126885	0.039634
Continent_Code_1	1.034405	0.040297
Continent_Code_2	-0.778936	0.041470
Continent_Code_3	0.807345	0.048726
Continent_Code_4	2.108365	0.126690
Continent_Code_5	1.860821	0.101869
error variance	0.682218	NaN

5.3 Regression Model Interpretations

5.3.1 First Model

From this regressive model, it is evident that `Support` (having someone to count on in times of trouble) and `Positive` affect have the largest influence on the happiness score. When all other factors are accounted for, a 0.1 increase in `Support` corresponds to a 0.198 increase in happiness score, and a 0.1 increase in `Positive` affect corresponds to a 0.249 increase in happiness score. Additionally, `Corruption` is the only factor that has a significant negative effect on the happiness score (the effect of `Negative` affect is insignificant). When `Corruption` increases by 0.1, the hap-

piness score decreases by .065. Furthermore, the factor with the lowest influence on the happiness score is `Lifespan`.

These results make intuitive sense. Using a psychological lens, one would believe that having someone to count on in times of trouble (support), as well as positive behavior such as smiling and learning something interesting, would have a positive effect on your happiness. Conversely, one would not expect people who are living in a corrupt environment to voluntarily report increases in happiness. Furthermore, if we compare these results to the correlation matrix, it is evident that the regression communicates adequate information about the data.

5.3.2 Second Model

From this regressive model, it is evident that living in Australia, North America, and Europe will increase the predicted happiness score for an individual. Specifically, at a base happiness score of 5.158, living in Australia will increase the predicted score by 2.11; living in North America will increase the predicted score by 1.86; and living in Europe will increase the predicted score by 1.03. The only continent with a negative effect on happiness score is Africa, which decreases the predicted score by -0.779.

Once again, these results are aligned with what we have observed previously. As shown by the map of Average Happiness by Country, it is evident that the countries with the greatest happiness scores seem to be generally clustered within Australia, North America, and Europe. Similarly, the continent with the lowest average score appears to be Africa. There are several economic, political, and social factors that can be attributed to these findings; however, for the sake of a concise analysis, we will let the data speak for itself.

6 Summary of Findings

According to our analysis, the geographic distribution of happiness **has indeed changed** in the last 18 years. The world map of the happiness delta from the oldest to most recent recorded year indicates that happiness is trending downward slightly in various parts of the West, such as multiple countries in western Europe and the United States and Canada in North America, while eastern countries like Russia and China appear to be increasing in happiness score over time. Additionally, countries like Lebanon have seen their happiness decrease significantly in recent years, while countries like Finland have been consistently very happy for numerous years.

It appears that a significant number of our predictors are correlated with happiness scores in each country, with our correlation plot indicating several predictors that are positively correlated with happiness: `GDP`, `Support`, `Lifespan`, `Freedom`, and `Positive` affect (`Corruption` is notably negatively correlated with happiness `Score`). Through our country-specific analyses of Finland and Lebanon, we were able to determine that the trends of many of these aforementioned predictors were reflected in the overall happiness score of each of the countries, further verifying that these changes in score can be attributed to many of our predictors.

Appendix Link to download the Jupyter Notebook for this assignment: https://drive.google.com/file/d/1bhfW8Hq6CzdJPNjBYig1zWMSGUC4T4gg/view?usp=sharing