World Happiness Index

November 29, 2020

1 World Happiness Index

1.1 preparing dataset

[1]:
```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
import seaborn as sns
sns.set_style('darkgrid', {'legend.frameon':True})
```

[2]:
```
df2015 = pd.read_csv('2015.csv')
df2016 = pd.read_csv('2016.csv')
df2017 = pd.read_csv('2017.csv')
df2018 = pd.read_csv('2018.csv')
df2019 = pd.read_csv('2019.csv')

df2015.drop(['Standard Error', 'Dystopia Residual'],axis=1,inplace=True)
df2015.rename(columns = {'Economy (GDP per Capita)':'GDP_per_capita',␣
 ↪'Happiness Rank':'Happiness_Rank', 'Happiness Score':'Happiness_Score',␣
 ↪'Family':'Social_Support', 'Health (Life Expectancy)':'Life_Expectancy',␣
 ↪'Trust (Government Corruption)':'Corruption'}, inplace = True)
#get all regions and proper column order for later on
country_region = df2015[['Country', 'Region']].copy()
cols = df2015.columns.tolist()

df2016.drop(['Lower Confidence Interval', 'Upper Confidence Interval',␣
 ↪'Dystopia Residual'],axis=1,inplace=True)
df2016.rename(columns = {'Economy (GDP per Capita)':'GDP_per_capita',␣
 ↪'Happiness Rank':'Happiness_Rank', 'Happiness Score':'Happiness_Score',␣
 ↪'Family':'Social_Support', 'Health (Life Expectancy)':'Life_Expectancy',␣
 ↪'Trust (Government Corruption)':'Corruption' }, inplace = True)

df2017.drop(['Whisker.high', 'Whisker.low', 'Dystopia.
 ↪Residual'],axis=1,inplace=True)
```

```
df2017.rename(columns = {'Happiness.Rank':'Happiness_Rank', 'Happiness.Score':
 ↪'Happiness_Score', 'Economy..GDP.per.Capita.':'GDP_per_capita', 'Family':␣
 ↪'Social_Support', 'Health..Life.Expectancy.': 'Life_Expectancy', 'Trust..
 ↪Government.Corruption.': 'Corruption'},inplace=True)
df2017 = df2017.merge(country_region, on='Country') #add the missing region for␣
 ↪year 2019
df2017 = df2017[cols] #sort columns

df2018.rename(columns = {'Overall rank':'Happiness_Rank', 'Country or region':
 ↪'Country', 'GDP per capita':'GDP_per_capita', 'Healthy life expectancy':
 ↪'Life_Expectancy', 'Perceptions of corruption':'Corruption', 'Social␣
 ↪support':'Social_Support', 'Freedom to make life choices':'Freedom', 'Score':
 ↪'Happiness_Score'},inplace=True)
df2018 = df2018.merge(country_region, on='Country') #add the missing region for␣
 ↪year 2019
df2018 = df2018[cols] #sort columns

df2019.rename(columns = {'Overall rank':'Happiness_Rank', 'Country or region':
 ↪'Country', 'GDP per capita':'GDP_per_capita', 'Healthy life expectancy':
 ↪'Life_Expectancy', 'Perceptions of corruption':'Corruption',  'Social␣
 ↪support':'Social_Support', 'Freedom to make life choices':'Freedom', 'Score':
 ↪'Happiness_Score'},inplace=True)
df2019 = df2019.merge(country_region, on='Country') #add the missing region for␣
 ↪year 2019
df2019 = df2019[cols] #sort columns
```

now all into one df

[3]:
```
df2015["year"] = str(2015)
df2016["year"] = str(2016)
df2017["year"] = str(2017)
df2018["year"] = str(2018)
df2019["year"] = str(2019)
df_all = df2015.append([df2016,df2017,df2018,df2019])
```

2 First lets look at correlation between happiness & the specific attributes

Considered in dataset contributing to happiness, values are in relation to Dystopia, the most unhappiest place on earth - GDP per Capita - Family - Life Expectancy - Freedom - Generosity - Trust Government Corruption

2.0.1 Correlation: influence of seperate factors regarding Happiness Rank

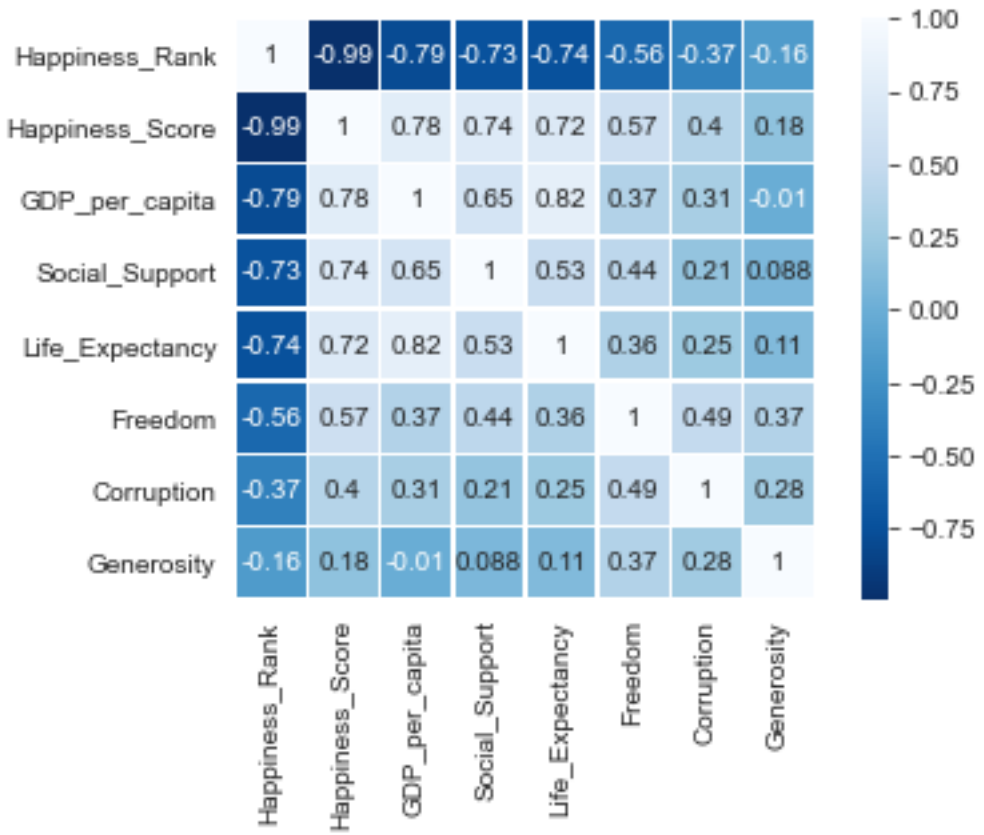
We are using a heatmap to show the correlation.

2.0.2 Year 2015

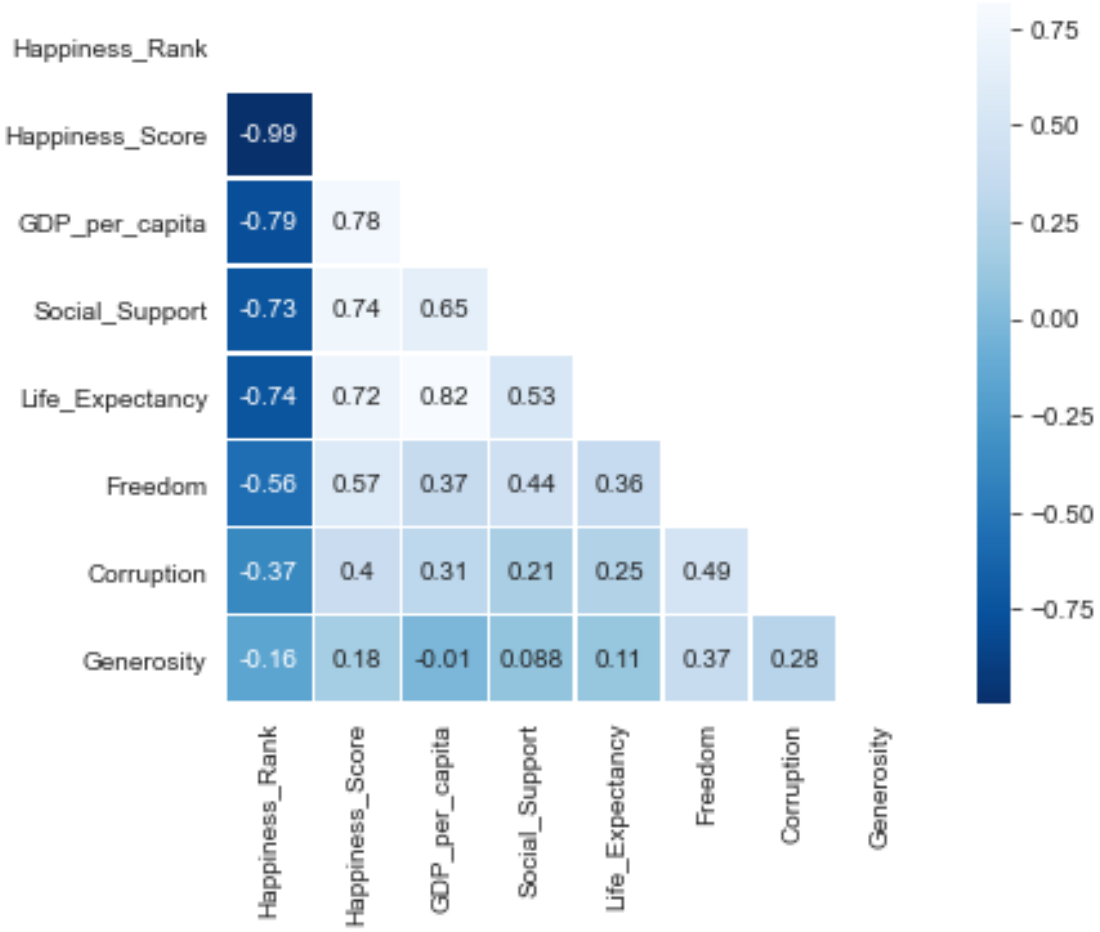
[4]:
```
corr2015 = df2015.corr()
corr2015
```

[4]:
```
                Happiness_Rank  Happiness_Score  GDP_per_capita  \
Happiness_Rank        1.000000        -0.992105       -0.785267
Happiness_Score      -0.992105         1.000000        0.780966
GDP_per_capita       -0.785267         0.780966        1.000000
Social_Support       -0.733644         0.740605        0.645299
Life_Expectancy      -0.735613         0.724200        0.816478
Freedom              -0.556886         0.568211        0.370300
Corruption           -0.372315         0.395199        0.307885
Generosity           -0.160142         0.180319       -0.010465

                Social_Support  Life_Expectancy   Freedom  Corruption  \
Happiness_Rank       -0.733644        -0.735613 -0.556886   -0.372315
Happiness_Score       0.740605         0.724200  0.568211    0.395199
GDP_per_capita        0.645299         0.816478  0.370300    0.307885
Social_Support        1.000000         0.531104  0.441518    0.205605
Life_Expectancy       0.531104         1.000000  0.360477    0.248335
Freedom               0.441518         0.360477  1.000000    0.493524
Corruption            0.205605         0.248335  0.493524    1.000000
Generosity            0.087513         0.108335  0.373916    0.276123

                Generosity
Happiness_Rank   -0.160142
Happiness_Score   0.180319
GDP_per_capita   -0.010465
Social_Support    0.087513
Life_Expectancy   0.108335
Freedom           0.373916
Corruption        0.276123
Generosity        1.000000
```

[5]:
```
sns.heatmap(corr2015, annot=True, linewidths=.5, square = True, cmap =␣
↪'Blues_r');
```



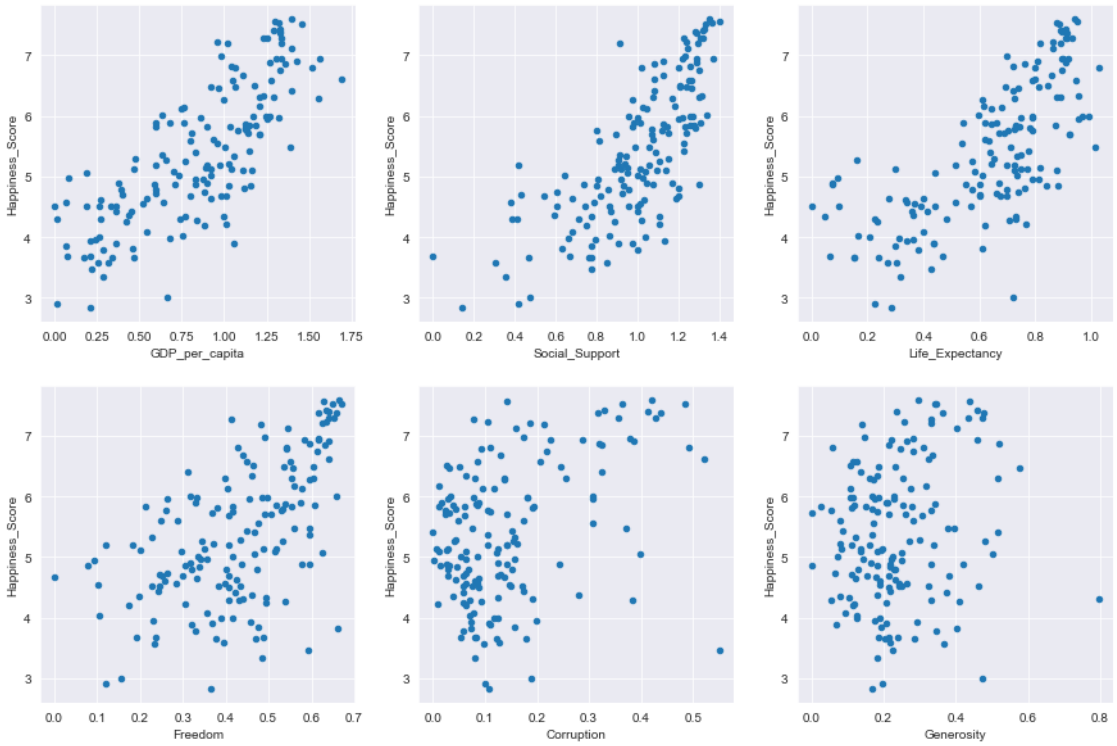
[6]:
```
mask = np.zeros_like(corr2015)

mask[np.triu_indices_from(mask)] = True

with sns.axes_style("white"):

    f, ax = plt.subplots(figsize=(7, 5))

    ax = sns.heatmap(corr2015, mask = mask, annot=True, linewidths=.5, square =␣
↪True, cmap = 'Blues_r')
```



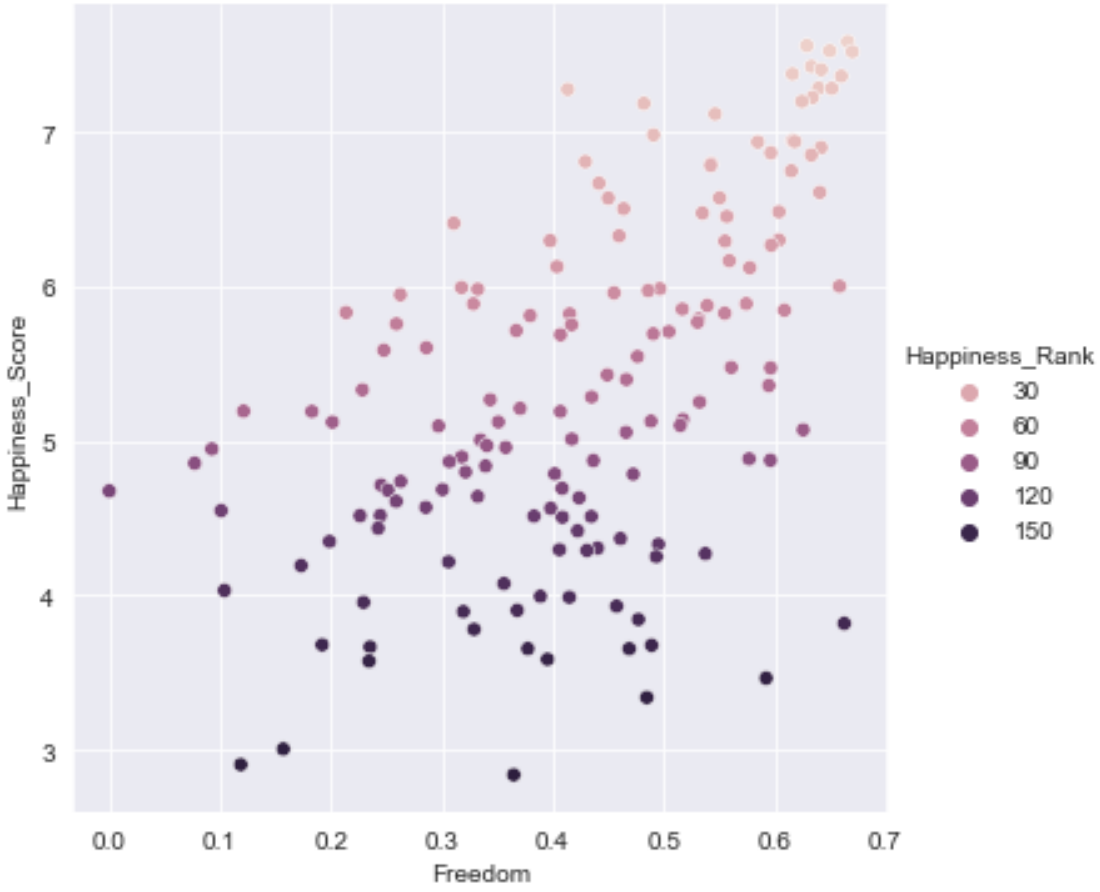
Conclusion: We can see that Happiness is highly dependent on GDP as well as Social Support through family and Healthy life expectancy and quite independent from corruption or generosity

Scatterplot to highlight correlation: also shows, how gdp and social support attribute the most to the happiness factor while generosity and preception of corruption show least correlation to felt happiness.

[7]:
```
fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(15,10));

df2015.plot.scatter(ax=axes[0,0], x = 'GDP_per_capita', y = 'Happiness_Score');
df2015.plot.scatter(ax=axes[0,1], x = 'Social_Support', y = 'Happiness_Score');
df2015.plot.scatter(ax=axes[0,2], x = 'Life_Expectancy', y = 'Happiness_Score');
df2015.plot.scatter(ax=axes[1,0], x = 'Freedom', y = 'Happiness_Score');
df2015.plot.scatter(ax=axes[1,1], x = 'Corruption', y = 'Happiness_Score');
df2015.plot.scatter(ax=axes[1,2], x = 'Generosity', y = 'Happiness_Score');

plt.show();
```



[8]:
```
sns.relplot(x="Freedom", y="Happiness_Score", hue="Happiness_Rank",␣
↪data=df2015);
```

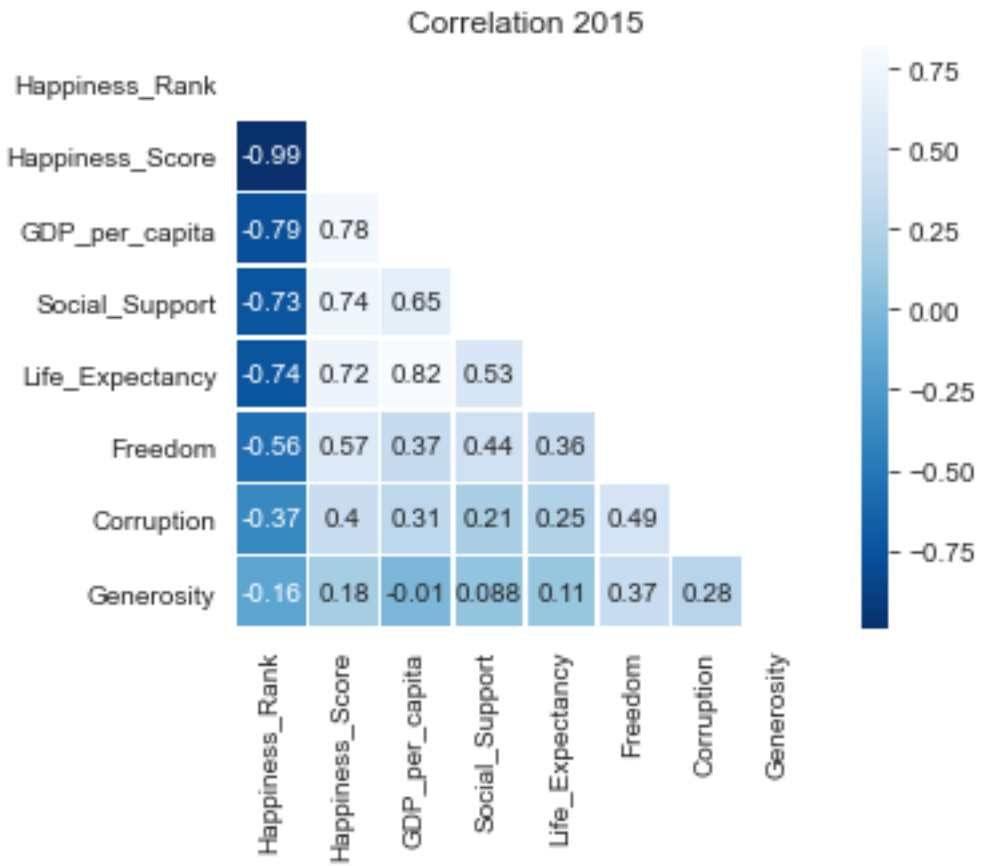


2.0.3 Change in correlation from 2015-2019

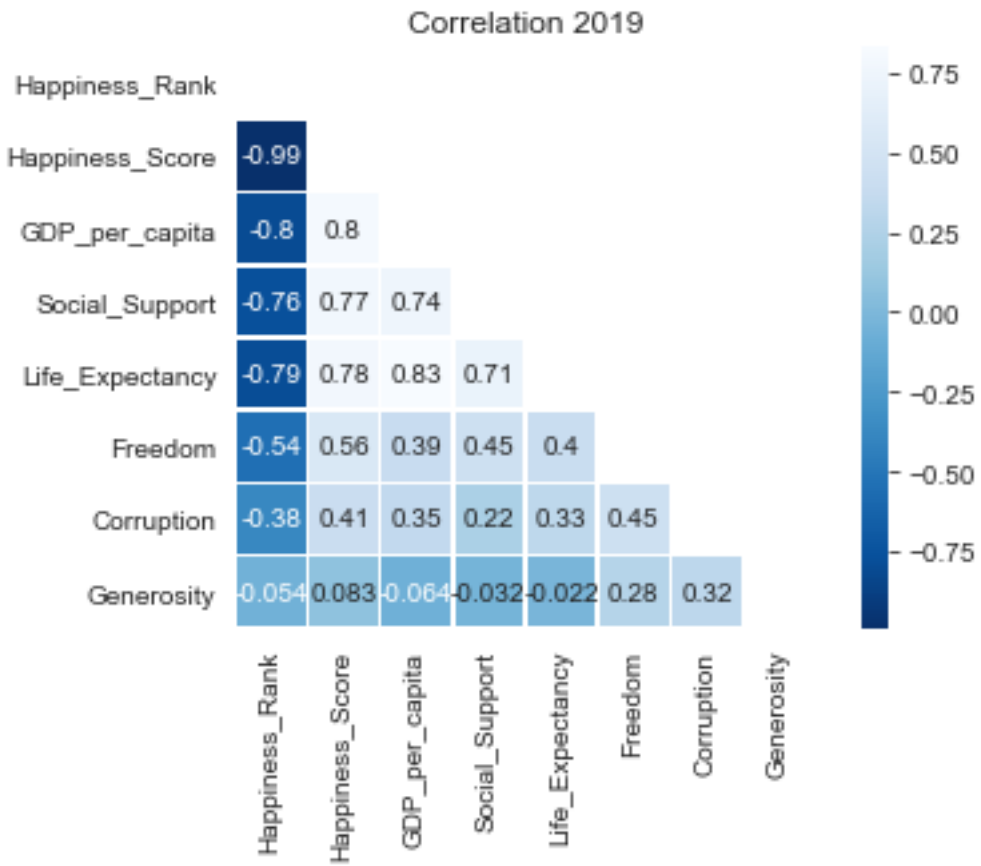
Does correlation change over the years and are other factors more important than others in 2019? Correlation in general:

[9]:
```
corr2019 = df2019.corr()
corr2015 = df2015.corr()
```

[10]:
```
mask = np.zeros_like(corr2015)

mask[np.triu_indices_from(mask)] = True

with sns.axes_style("white"):

    f, ax = plt.subplots()

    ax = sns.heatmap(corr2015, mask = mask, annot=True, linewidths=.5, square =␣
↪True, cmap = 'Blues_r').set_title('Correlation 2015')
```

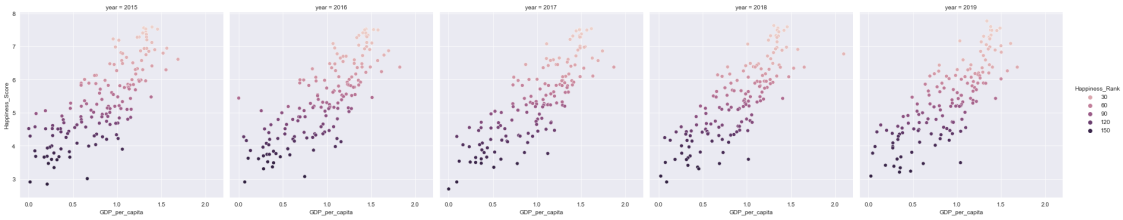


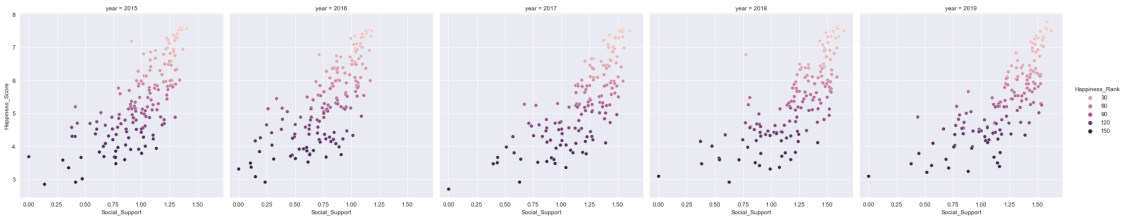
[11]:
```
mask = np.zeros_like(corr2019)

mask[np.triu_indices_from(mask)] = True

with sns.axes_style("white"):

    f, ax = plt.subplots()

    ax = sns.heatmap(corr2019, mask = mask, annot=True, linewidths=.5, square =␣
↪True, cmap = 'Blues_r').set_title('Correlation 2019')
```

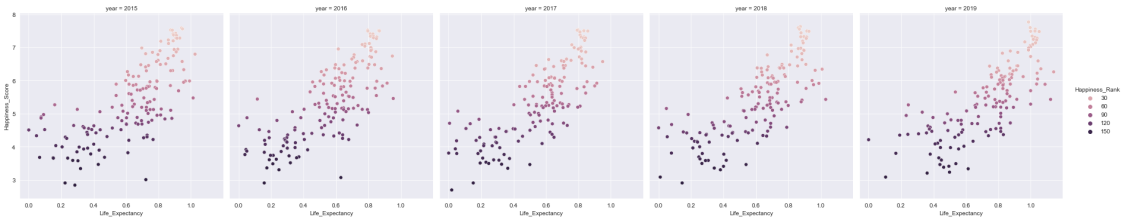


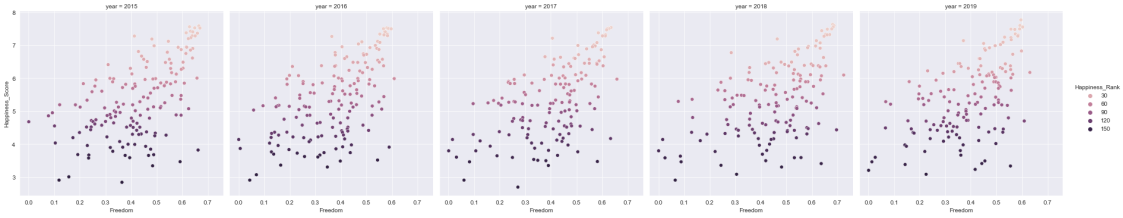
And in more detail specific development of attributes over time from year 2015 to 2019

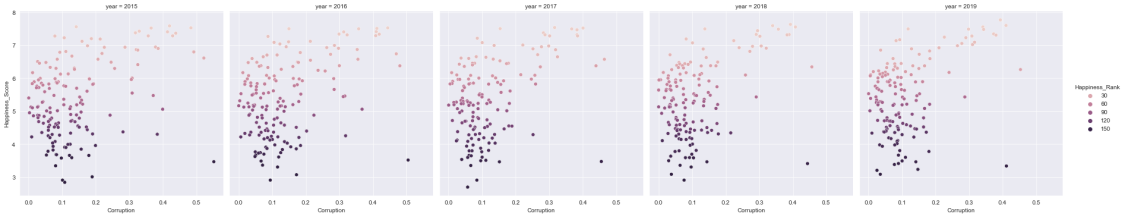
[12]:
```
sns.relplot(data=df_all, x="GDP_per_capita", y="Happiness_Score",␣
 ↪hue="Happiness_Rank", col="year");
sns.relplot(data=df_all, x="Social_Support", y="Happiness_Score",␣
 ↪hue="Happiness_Rank", col="year");
sns.relplot(data=df_all, x="Life_Expectancy", y="Happiness_Score",␣
 ↪hue="Happiness_Rank", col="year");
sns.relplot(data=df_all, x="Freedom", y="Happiness_Score",␣
 ↪hue="Happiness_Rank", col="year");
sns.relplot(data=df_all, x="Corruption", y="Happiness_Score",␣
 ↪hue="Happiness_Rank", col="year");
sns.relplot(data=df_all, x="Generosity", y="Happiness_Score",␣
 ↪hue="Happiness_Rank", col="year");
```

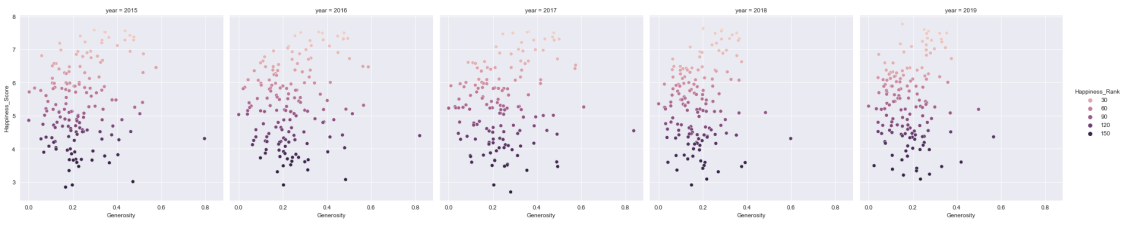

year = 2015
year = 2016
year = 2017
year = 2018
year = 2019
Happiness_Score
GDP_per_capita
Happiness_Rank
30
60
90
120
150


year = 2015
year = 2016
year = 2017
year = 2018
year = 2019
Happiness_Score
Social_Support
Happiness_Rank
30
60
90
120
150


year = 2015
year = 2016
year = 2017
year = 2018
year = 2019
Happiness_Score
Life_Expectancy
Happiness_Rank
30
60
90
120
150


year = 2015
year = 2016
year = 2017
year = 2018
year = 2019
Happiness_Score
Freedom
Happiness_Rank
30
60
90
120
150


year = 2015
year = 2016
year = 2017
year = 2018
year = 2019
Happiness_Score
Corruption
Happiness_Rank
30
60
90
120
150



Conclusion: Therefore we can see that correlation in 2015 and 2019 is still depending on the same attributes, mainly GDP, social support aswell as life expectancy and less on corruption or generosity. Therefore we will focus on 2015 in the following exploration of our data.

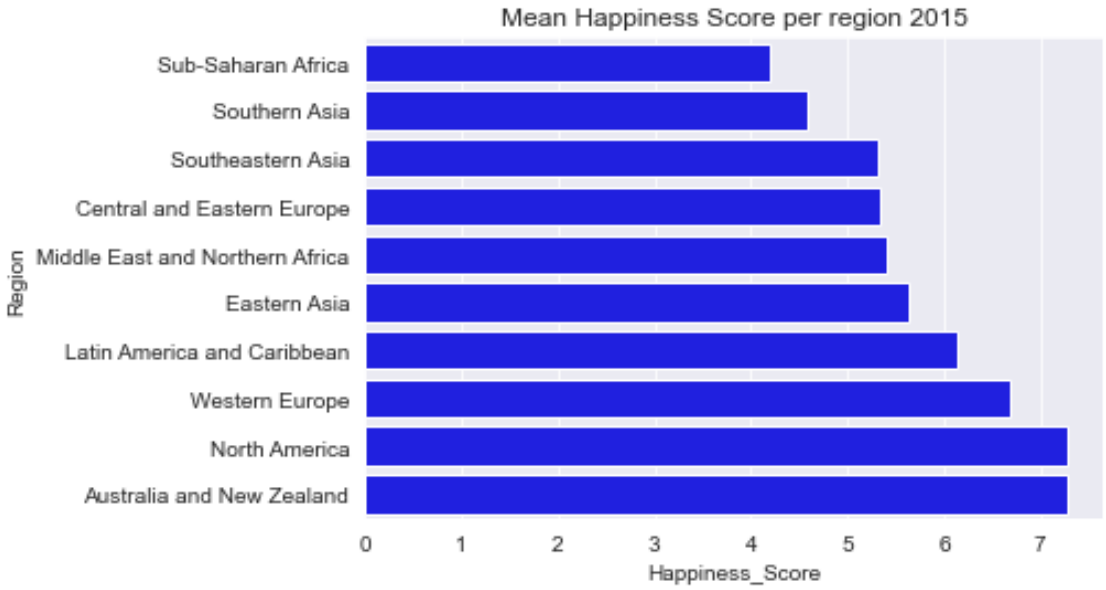
2.1 Mean happiness score per region in year 2015

Now that we found out that on which factors happiness depends the most, we take a closer look at mean value of the happiness score in each region

[13]:
```
df2015_mean_happiness = df2015.copy()
location_mean_byregion = df2015.groupby(['Region']).mean()
mean_happiness_scores = location_mean_byregion['Happiness_Score'].to_dict()
df2015_mean_happiness['Mean_Happiness_Score'] = df2015_mean_happiness['Region'].
↪map(mean_happiness_scores)
```

[14]:
```
df2015_mean_happiness_sorted = df2015_mean_happiness.
↪sort_values(['Mean_Happiness_Score'])
```

[15]:
```
g = sns.barplot(data=df2015_mean_happiness_sorted, x='Happiness_Score',␣
↪y='Region', color='blue',ci = None);
g.set_title("Mean Happiness Score per region 2015")
plt.show()
```

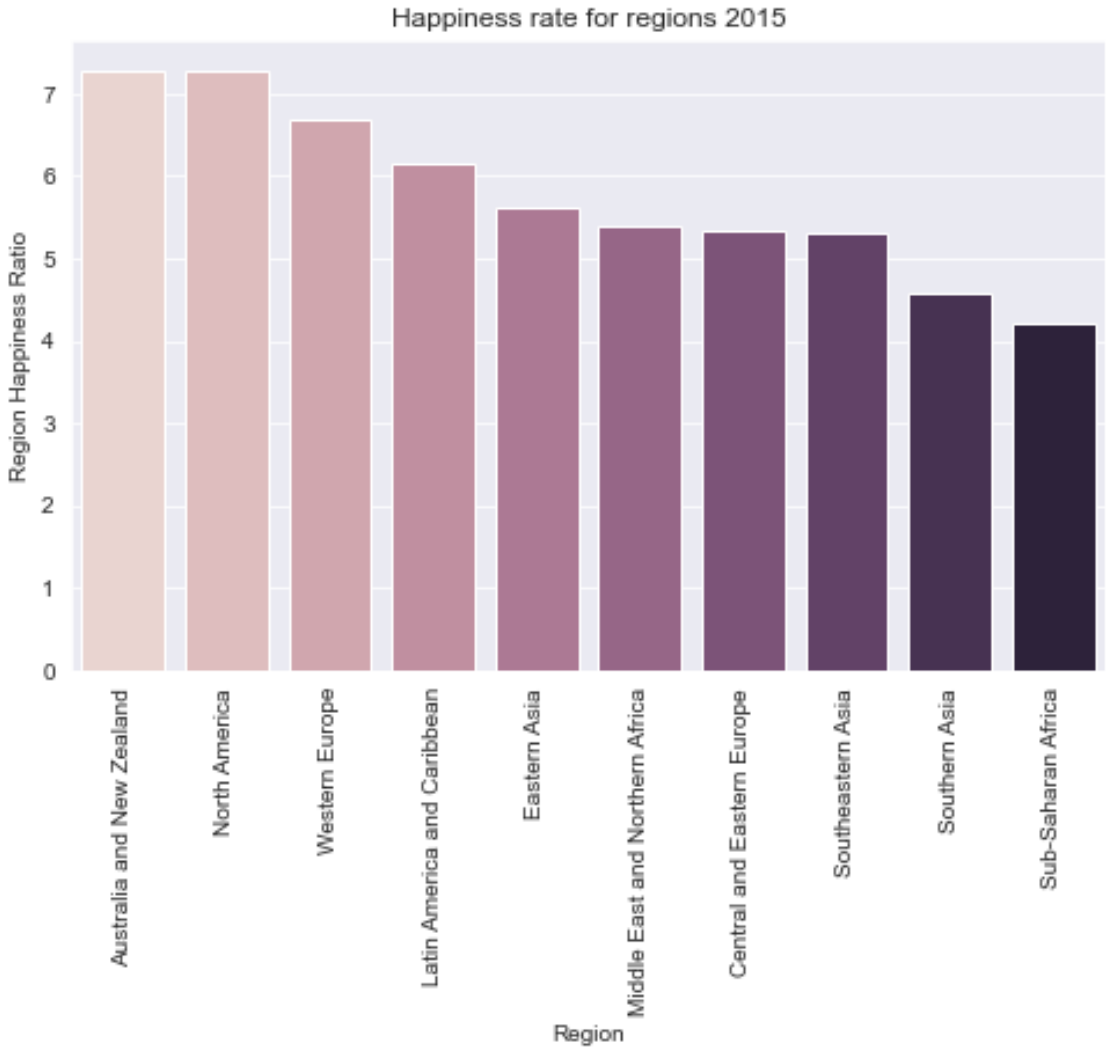


[16]:
```
region_lists=list(df2015['Region'].unique())
region_happiness_ratio=[]
for each in region_lists:
    region=df2015[df2015['Region']==each]
    region_happiness_rate=sum(region['Happiness_Score']/len(region))
    region_happiness_ratio.append(region_happiness_rate)

data=pd.DataFrame({'region':region_lists,'region_happiness_ratio':
 ↪region_happiness_ratio})
new_index=(data['region_happiness_ratio'].sort_values(ascending=False)).index.
 ↪values
sorted_data = data.reindex(new_index)

sorted_data
```

[16]:

	region	region_happiness_ratio
2	Australia and New Zealand	7.285000
1	North America	7.273000
0	Western Europe	6.689619
4	Latin America and Caribbean	6.144682
7	Eastern Asia	5.626167
3	Middle East and Northern Africa	5.406900
6	Central and Eastern Europe	5.332931
5	Southeastern Asia	5.317444
9	Southern Asia	4.580857
8	Sub-Saharan Africa	4.202800

[17]:
```
plt.figure(figsize=(8,5))
sns.barplot(x=sorted_data['region'],␣
 ↪y=sorted_data['region_happiness_ratio'],palette=sns.
 ↪cubehelix_palette(len(sorted_data['region'])))
plt.xticks(rotation= 90)
plt.xlabel('Region')
plt.ylabel('Region Happiness Ratio')
plt.title('Happiness rate for regions 2015')
plt.show()
```



2.2 Distribution of happiness per region

Let's look at distribution of Happiness per region. Are there countries with only good ranks or bad ranks? Are there regions with broad spectrums from very happy to very unhappy? Let's have a look at the maximum and minimum values of Happiness Score per region

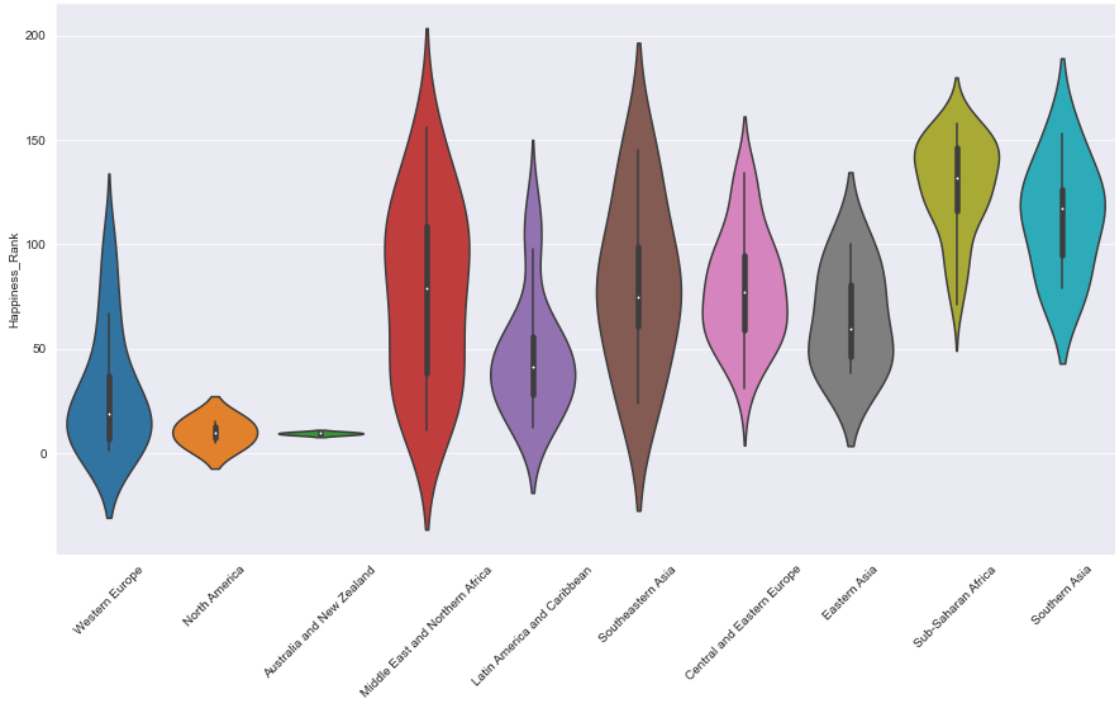
[18]:
```
df2015.groupby([df2015.Region])["Happiness_Score"].agg(["max", "min"]).
↪rename_axis(["region"])
```

[18]:

region	max	min
Australia and New Zealand	7.286	7.284
Central and Eastern Europe	6.505	4.218
Eastern Asia	6.298	4.874
Latin America and Caribbean	7.226	4.518
Middle East and Northern Africa	7.278	3.006
North America	7.427	7.119
Southeastern Asia	6.798	3.819
Southern Asia	5.253	3.575
Sub-Saharan Africa	5.477	2.839
Western Europe	7.587	4.857

Violin Plot

[19]:
```
fig = plt.gcf()
fig.set_size_inches(15, 8)

ax = sns.violinplot(x="Region", y="Happiness_Rank", scale="width", data=df2015)
for item in ax.get_xticklabels():
    item.set_rotation(45)
ax.set(xlabel=None)

plt.show()
```

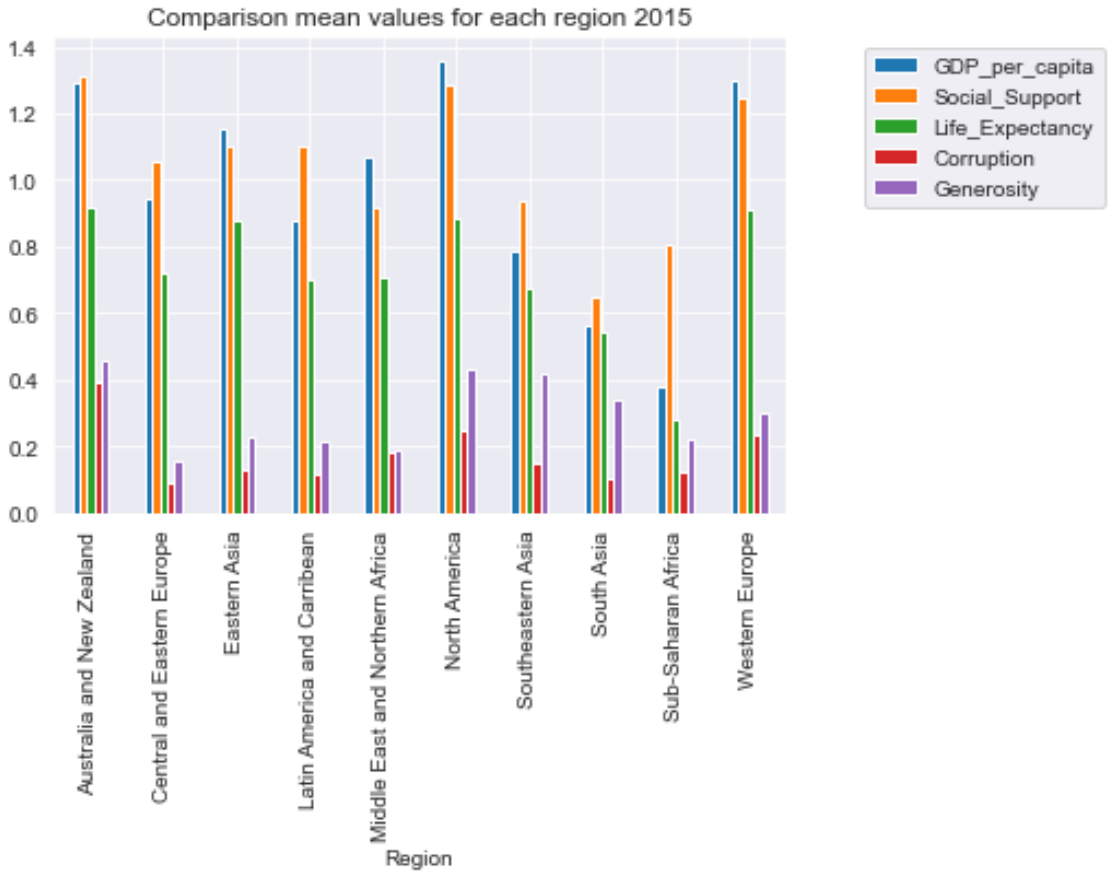


Conclusion: Australia and New Zealand obviously is only 2 countries, but we can see that Western Europe is mainly happy, in the middle east the band is pretty wide and in Sub Saharan Africa aswell as Southern Asia Happiness is distributed mainly in the bottom part of the rank.

The information of this plot is useful to detect differences in Regions and might be useful for the final poster.

[20]:
```
df2015_mean = pd.DataFrame(location_mean_byregion.to_dict())
df2015_mean['Region'] = ['Australia and New Zealand', 'Central and Eastern␣
 ↪Europe', 'Eastern Asia', 'Latin America and Carribean', 'Middle East and␣
 ↪Northern Africa', 'North America', 'Southeastern Asia', 'South Asia',␣
 ↪'Sub-Saharan Africa', 'Western Europe']
```

[21]:
```
df2015_mean.plot(x='Region', y=['GDP_per_capita', 'Social_Support',␣
 ↪'Life_Expectancy', 'Corruption', 'Generosity'], kind='bar');
plt.legend(bbox_to_anchor=(1.45,1), loc='upper right')
plt.title('Comparison mean values for each region 2015')
plt.show()
```



[22]:
```
df2015_mean2 = df2015_mean.copy()
df2015_mean2.drop(['Happiness_Rank', 'Happiness_Score',␣
 ↪'Region'],axis=1,inplace=True)
df2015_mean2
```

[22]:

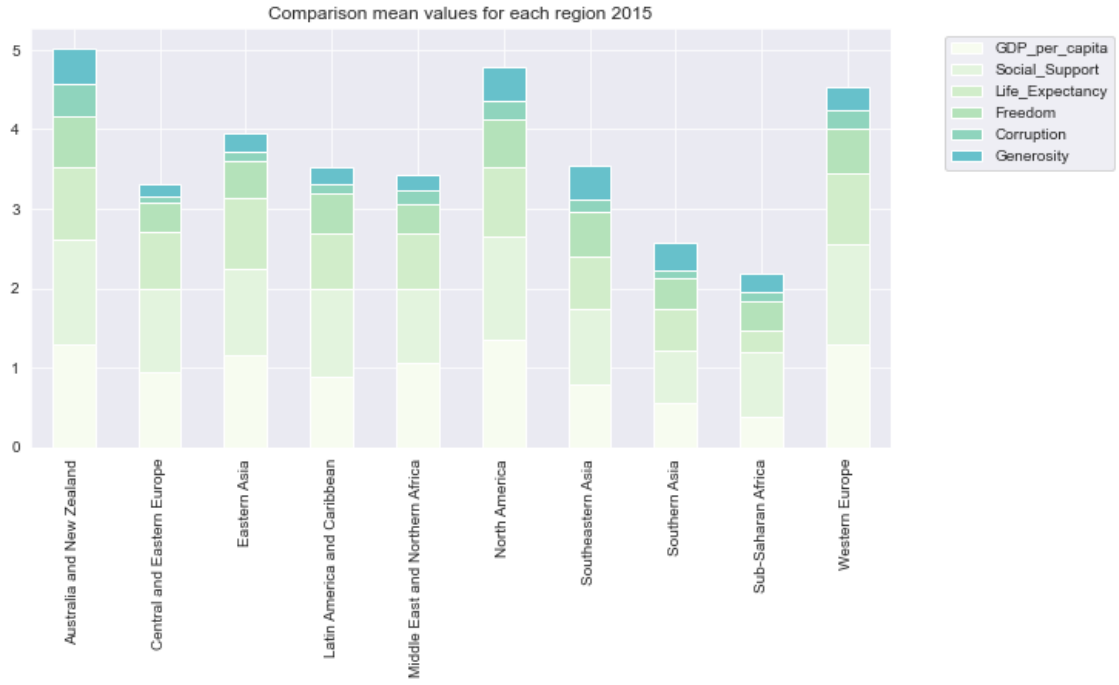
	GDP_per_capita	Social_Support	\
Australia and New Zealand	1.291880	1.314450	
Central and Eastern Europe	0.942438	1.053042	
Eastern Asia	1.151780	1.099427	
Latin America and Caribbean	0.876815	1.104720	
Middle East and Northern Africa	1.066973	0.920490	
North America	1.360400	1.284860	
Southeastern Asia	0.789054	0.940468	
Southern Asia	0.560486	0.645321	
Sub-Saharan Africa	0.380473	0.809085	
Western Europe	1.298596	1.247302	

	Life_Expectancy	Freedom	Corruption	\
Australia and New Zealand	0.919965	0.645310	0.392795	
Central and Eastern Europe	0.718774	0.358269	0.086674	
Eastern Asia	0.877388	0.462490	0.127695	
Latin America and Caribbean	0.703870	0.501740	0.117172	
Middle East and Northern Africa	0.705616	0.361751	0.181702	
North America	0.883710	0.589505	0.244235	
Southeastern Asia	0.677357	0.557104	0.151276	
Southern Asia	0.540830	0.373337	0.102536	
Sub-Saharan Africa	0.282332	0.365944	0.123878	
Western Europe	0.909148	0.549926	0.231463	

	Generosity
Australia and New Zealand	0.455315
Central and Eastern Europe	0.152264
Eastern Asia	0.225885
Latin America and Caribbean	0.217788
Middle East and Northern Africa	0.190375
North America	0.429580
Southeastern Asia	0.419261
Southern Asia	0.341429
Sub-Saharan Africa	0.221137
Western Europe	0.302109

[23]:
```
colors = plt.cm.GnBu(np.linspace(0, 1, 10))
plt.rc('axes', axisbelow=True)

ax = df2015_mean2.plot(kind='bar', stacked=True, figsize=(10, 5), color=colors)
plt.legend(bbox_to_anchor=(1.27,1), loc='upper right')
plt.title('Comparison mean values for each region 2015')
```

```
plt.show()
```



2.2.1 Top 10 Countries from 2015 next to each other regarding different attributes using stacked bar charts

Therefore we focus now mainly on year 2015. And evaluate the seperate attributes. Are there differences in the top 10 most happiest countries in the world? Does one country derive its happiness more from generosity compared to others?

[ ]:
```
df2015_top10 = df2015.head(10)
df2015_top10.drop(['Happiness_Rank', 'Region', 'Happiness_Score',␣
↪'year'],axis=1,inplace=True)
```

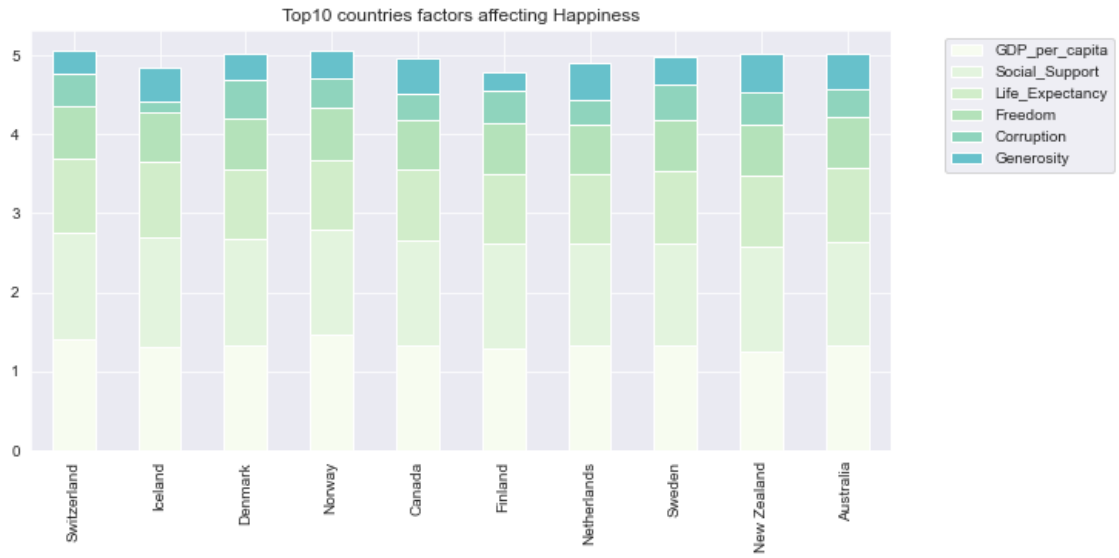
[25]:
```
df2015_top10
```

[25]:

	Country	GDP_per_capita	Social_Support	Life_Expectancy	Freedom	\
0	Switzerland	1.39651	1.34951	0.94143	0.66557	
1	Iceland	1.30232	1.40223	0.94784	0.62877	
2	Denmark	1.32548	1.36058	0.87464	0.64938	
3	Norway	1.45900	1.33095	0.88521	0.66973	
4	Canada	1.32629	1.32261	0.90563	0.63297	
5	Finland	1.29025	1.31826	0.88911	0.64169	
6	Netherlands	1.32944	1.28017	0.89284	0.61576	
7	Sweden	1.33171	1.28907	0.91087	0.65980	

```
8  New Zealand         1.25018         1.31967          0.90837  0.63938
9    Australia         1.33358         1.30923          0.93156  0.65124

   Corruption  Generosity
0     0.41978     0.29678
1     0.14145     0.43630
2     0.48357     0.34139
3     0.36503     0.34699
4     0.32957     0.45811
5     0.41372     0.23351
6     0.31814     0.47610
7     0.43844     0.36262
8     0.42922     0.47501
9     0.35637     0.43562
```

[26]:
```
colors = plt.cm.GnBu(np.linspace(0, 1, 10))

ax = df2015_top10.plot(kind='bar', stacked=True, figsize=(10, 5), color=colors)
ax.set_xticks(df2015_top10.index)
ax.set_xticklabels(df2015_top10.Country, rotation=90)
plt.legend(bbox_to_anchor=(1.27,1), loc='upper right')
plt.title('Top10 countries factors affecting Happiness')

plt.show()
```



2.3 Maps

2.3.1 global

Let's get an overview by looking at a world map to visualize the distribution of happiness around the world and the development and changes from 2015-2019.

[27]:
```
import plotly.express as px

fig = px.choropleth(df_all, locationmode = 'country names', locations="Country",
                    color="Happiness_Rank",
                    hover_name="Country",
                    animation_frame="year",
                    color_continuous_scale=px.colors.sequential.Plasma)
fig.show()
```

2.3.2 Regional Averages for 2015

[28]:
```
fig = px.choropleth(df2015_mean_happiness, locationmode = 'country names',␣
↪locations="Country",
                    color="Mean_Happiness_Score",
                    hover_name="Region",
                    animation_frame="year",
                    color_continuous_scale=px.colors.sequential.Plasma)
fig.show()
```

Based on the analysis of just the mean values for each region, the expected result of an overall happy North America, Australia and Western Europe can be seen clearly. Additionally, the Latin America and Carribean region is happier on average than all of Asia and Africa. Sub-Saharan Africa is on average the unhappiest place, followed closely by Southern Asia, where India, Pakistan and Afghanistan are.

3 Regional exploration

[29]:
```
set(df_all.Region)
```

[29]:
```
{'Australia and New Zealand',
 'Central and Eastern Europe',
 'Eastern Asia',
 'Latin America and Caribbean',
 'Middle East and Northern Africa',
 'North America',
 'Southeastern Asia',
 'Southern Asia',
 'Sub-Saharan Africa',
 'Western Europe'}
```

[30]:
```
df_all_oceania = df_all[df_all.Region == 'Australia and New Zealand']
df_all_EuropeCentralEast = df_all[df_all.Region == 'Central and Eastern Europe']
df_all_EuropeWestern = df_all[df_all.Region == 'Western Europe']
df_all_AmericaSouth = df_all[df_all.Region == 'Latin America and Caribbean']
df_all_AmericaNorth = df_all[df_all.Region == 'North America']
df_all_AfricaMiddleEastNorth = df_all[df_all.Region == 'Middle East and␣
↪Northern Africa']
df_all_AfricaSubSahara = df_all[df_all.Region == 'Sub-Saharan Africa']
df_all_AsiaEast = df_all[df_all.Region == 'Eastern Asia']
df_all_AsiaSouthEast = df_all[df_all.Region == 'Southeastern Asia']
df_all_AsiaSouth = df_all[df_all.Region == 'Southern Asia']
```

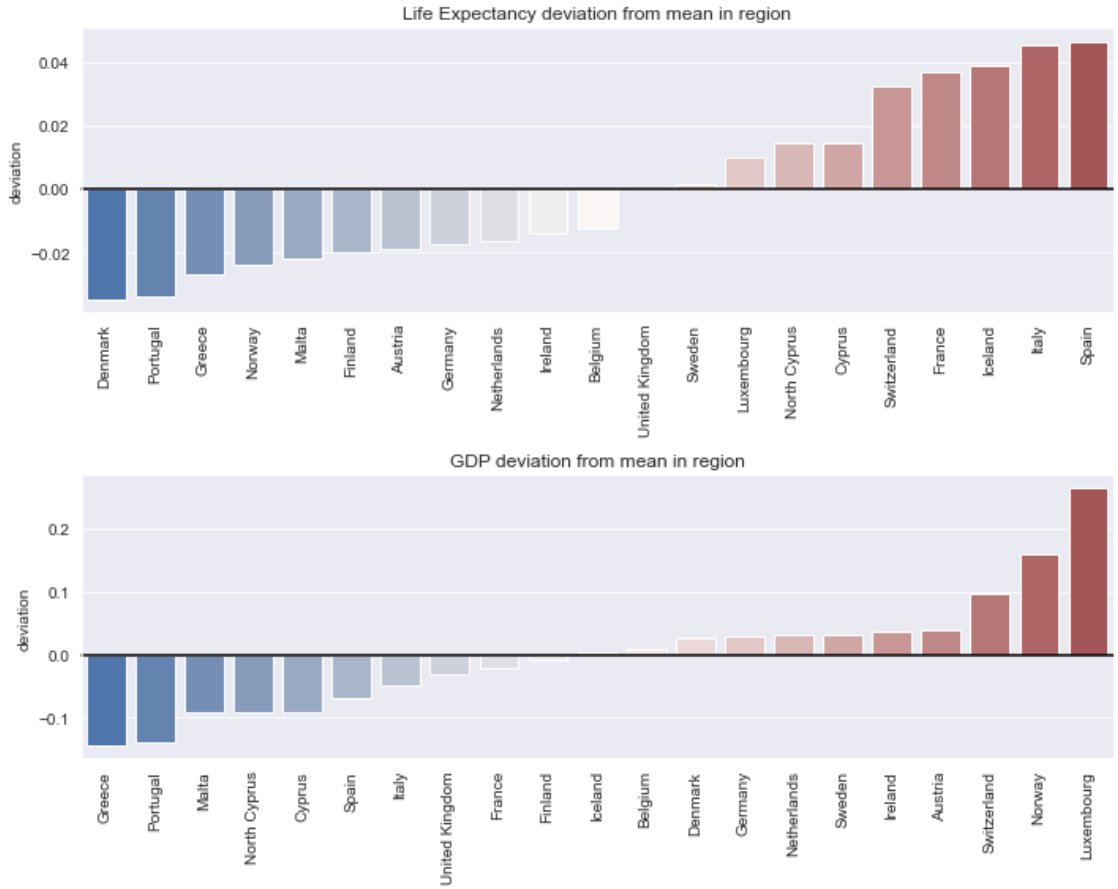
Deviation from mean regarding GDP and Life Expectancy in Europe in year 2015

[31]:
```
df2015_WesternEurope = df2015[df2015.Region == 'Western Europe']
```

[ ]:
```
df2015_WesternEurope['deviation_GDP']=df2015_WesternEurope.GDP_per_capita -␣
↪df2015_WesternEurope.GDP_per_capita.mean()
df2015_WesternEurope['deviation_Life']=df2015_WesternEurope.Life_Expectancy -␣
↪df2015_WesternEurope.Life_Expectancy.mean()
```

[33]:
```
#### Deviation from mean each region
f, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8))

df2015_WesternEurope.sort_values(['deviation_Life'], inplace=True)
y1 = df2015_WesternEurope.deviation_Life
sns.barplot(x=df2015_WesternEurope.Country, y=y1, palette="vlag", ax=ax1).
↪set_title('Life Expectancy deviation from mean in region')
ax1.axhline(0, color="k", clip_on=False)
ax1.set_ylabel("deviation")
ax1.set(xlabel=None)
for item in ax1.get_xticklabels():
    item.set_rotation(90)

df2015_WesternEurope.sort_values(['deviation_GDP'], inplace=True)
y2 = df2015_WesternEurope.GDP_per_capita - df2015_WesternEurope.GDP_per_capita.
↪mean()
sns.barplot(x=df2015_WesternEurope.Country, y=y2, palette="vlag", ax=ax2).
↪set_title('GDP deviation from mean in region')
ax2.axhline(0, color="k", clip_on=False)
ax2.set_ylabel("deviation")
ax2.set(xlabel=None)
for item in ax2.get_xticklabels():
    item.set_rotation(90)

f.tight_layout()
plt.show()
```

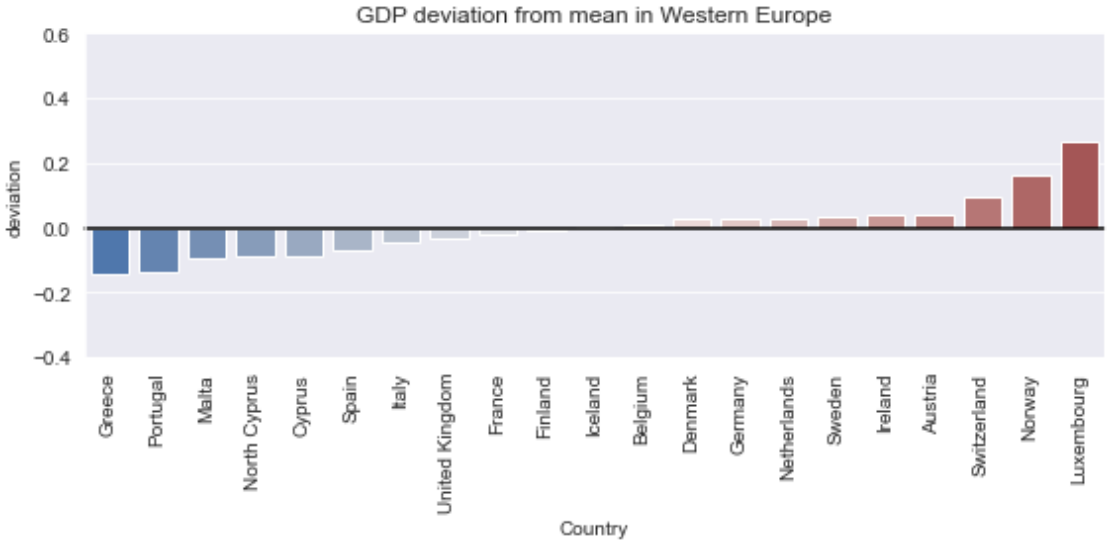
```
C:\Users\r-sut\anaconda3\envs\TSM_InfVis\lib\site-
packages\ipykernel_launcher.py:4: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

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

C:\Users\r-sut\anaconda3\envs\TSM_InfVis\lib\site-
packages\ipykernel_launcher.py:13: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

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

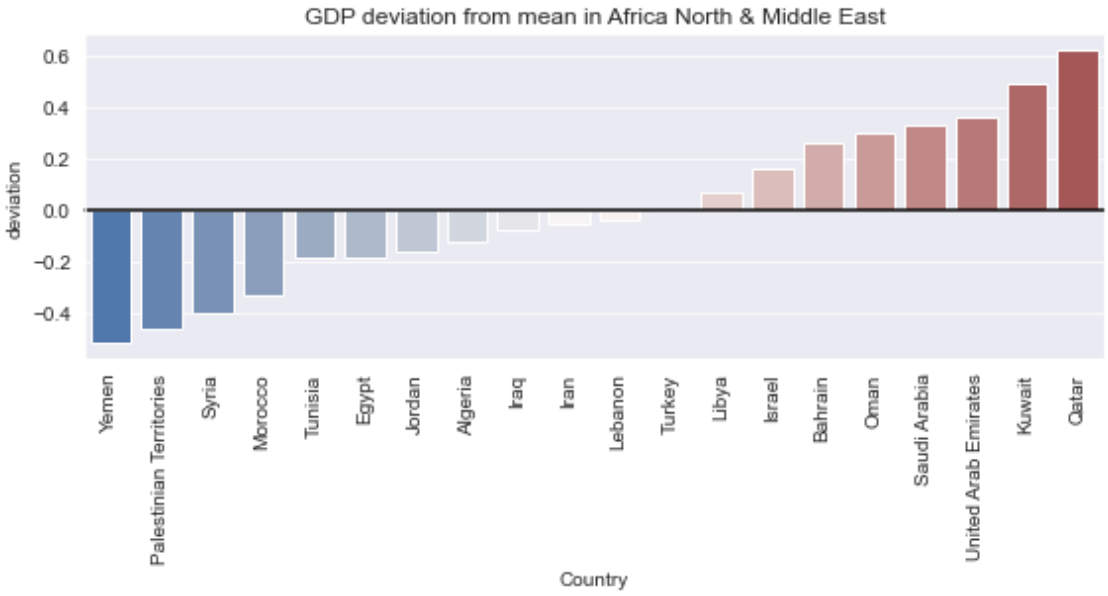


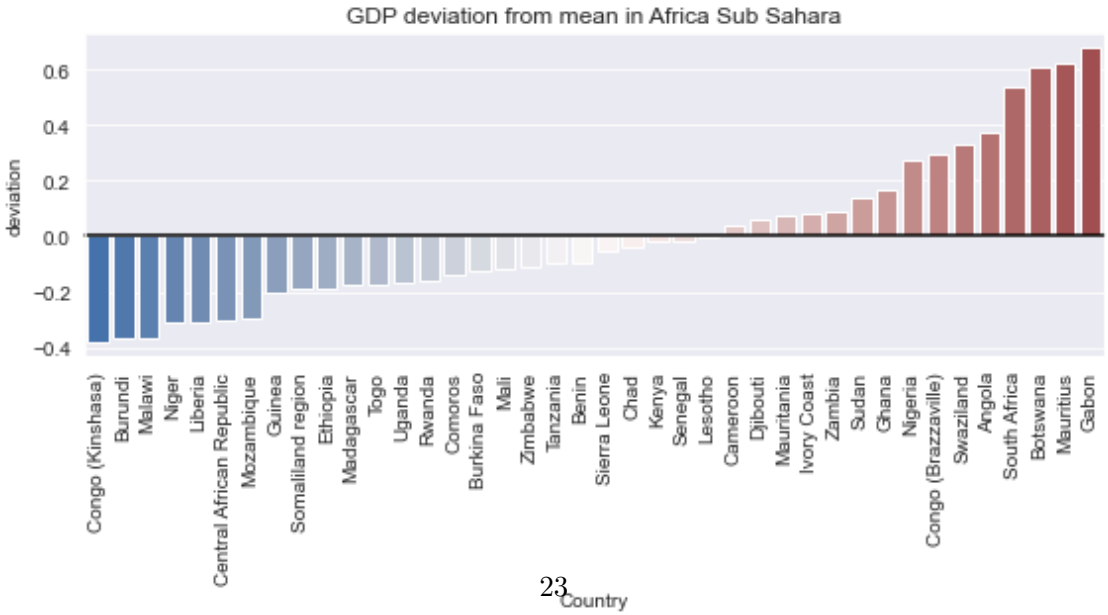
3.0.1 Regions Africa Middle East, Sub Sahara an Europe in comparison GDP deviation from mean in 2015

[ ]:
```
df2015_AfricaMiddleEastNorth = df2015[df2015.Region == 'Middle East and␣
↪Northern Africa']
df2015_AfricaSubSahara = df2015[df2015.Region == 'Sub-Saharan Africa']
df2015_AfricaMiddleEastNorth['deviation_GDP']=df2015_AfricaMiddleEastNorth.
↪GDP_per_capita - df2015_AfricaMiddleEastNorth.GDP_per_capita.mean()
df2015_AfricaSubSahara['deviation_GDP']=df2015_AfricaSubSahara.GDP_per_capita -␣
↪df2015_AfricaSubSahara.GDP_per_capita.mean()
df2015_AfricaSubSahara.sort_values(['deviation_GDP'], inplace=True)
df2015_AfricaMiddleEastNorth.sort_values(['deviation_GDP'], inplace=True)
```

[35]:
```
#### Deviation from mean each region
f, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(8, 13))

y1 = df2015_WesternEurope.GDP_per_capita - df2015_WesternEurope.GDP_per_capita.
↪mean()
sns.barplot(x=df2015_WesternEurope.Country, y=y1, palette="vlag", ax=ax1).
↪set_title('GDP deviation from mean in Western Europe')
ax1.axhline(0, color="k", clip_on=False)
ax1.set_ylabel("deviation")
ax1.set_ylim([-0.4, 0.6])
for item in ax1.get_xticklabels():
    item.set_rotation(90)

y2 = df2015_AfricaMiddleEastNorth.GDP_per_capita - df2015_AfricaMiddleEastNorth.
↪GDP_per_capita.mean()
sns.barplot(x=df2015_AfricaMiddleEastNorth.Country, y=y2, palette="vlag",␣
↪ax=ax2).set_title('GDP deviation from mean in Africa North & Middle East')
ax2.axhline(0, color="k", clip_on=False)
ax2.set_ylabel("deviation")
for item in ax2.get_xticklabels():
    item.set_rotation(90)

y3 = df2015_AfricaSubSahara.GDP_per_capita - df2015_AfricaSubSahara.
↪GDP_per_capita.mean()
sns.barplot(x=df2015_AfricaSubSahara.Country, y=y3, palette="vlag", ax=ax3).
↪set_title('GDP deviation from mean in Africa Sub Sahara')
ax3.axhline(0, color="k", clip_on=False)
ax3.set_ylabel("deviation")
for item in ax3.get_xticklabels():
    item.set_rotation(90)

f.tight_layout()
plt.show()
```


GDP deviation from mean in Western Europe
deviation
0.6
0.4
0.2
0.0
−0.2
−0.4
Greece
Portugal
Malta
North Cyprus
Cyprus
Spain
Italy
United Kingdom
France
Finland
Iceland
Belgium
Denmark
Germany
Netherlands
Sweden
Ireland
Austria
Switzerland
Norway
Luxembourg
Country


GDP deviation from mean in Africa North & Middle East
deviation
0.6
0.4
0.2
0.0
−0.2
−0.4
Yemen
Palestinian Territories
Syria
Morocco
Tunisia
Egypt
Jordan
Algeria
Iraq
Iran
Lebanon
Turkey
Libya
Israel
Bahrain
Oman
Saudi Arabia
United Arab Emirates
Kuwait
Qatar
Country


GDP deviation from mean in Africa Sub Sahara
deviation
0.6
0.4
0.2
0.0
−0.2
−0.4
Congo (Kinshasa)
Burundi
Malawi
Niger
Liberia
Central African Republic
Mozambique
Guinea
Somaliland region
Ethiopia
Madagascar
Togo
Uganda
Rwanda
Comoros
Burkina Faso
Mali
Zimbabwe
Tanzania
Benin
Sierra Leone
Chad
Kenya
Senegal
Lesotho
Cameroon
Djibouti
Mauritania
Ivory Coast
Zambia
Sudan
Ghana
Nigeria
Congo (Brazzaville)
Swaziland
Angola
South Africa
Botswana
Mauritius
Gabon
Country

3.0.2 Map visualization development of happiness rank from 2015-2019 per country

[36]:
```
fig = px.choropleth(df_all, locationmode = 'country names', locations="Country",
                    color="Happiness_Rank",
                    hover_name="Country",
                    animation_frame="year",
                    scope="north america",
                    color_continuous_scale=px.colors.sequential.Plasma)
fig.show()
```

[37]:
```
fig = px.choropleth(df_all, locationmode = 'country names', locations="Country",
                    color="Happiness_Rank",
                    hover_name="Country",
                    animation_frame="year",
                    scope="europe",
                    color_continuous_scale=px.colors.sequential.Plasma)
fig.show()
```

[38]:
```
fig = px.choropleth(df_all, locationmode = 'country names', locations="Country",
                    color="Happiness_Rank",
                    hover_name="Country",
                    animation_frame="year",
                    scope="asia",
                    color_continuous_scale=px.colors.sequential.Plasma)
fig.show()
```

[39]:
```
fig = px.choropleth(df_all, locationmode = 'country names', locations="Country",
                    color="Happiness_Rank",
                    hover_name="Country",
                    animation_frame="year",
                    scope="africa",
                    color_continuous_scale=px.colors.sequential.Plasma)
fig.show()
```

[40]:
```
fig = px.choropleth(df_all, locationmode = 'country names', locations="Country",
                    color="Happiness_Rank",
                    hover_name="Country",
                    animation_frame="year",
                    scope="south america",
                    color_continuous_scale=px.colors.sequential.Plasma)
fig.show()
```

4 Top 10 changes in happiness rank from 2015 to 2019, are there any reasons for this?

Which country made the biggest gain or loss in Happiness Rank from 2015 to 2019.

[41]:
```
df2015_ranks = df2015[['Country','Happiness_Rank']]
df2015_ranks.rename(columns = {'Happiness_Rank':'2015'}, inplace = True)

df_ranks_change = df2019[['Country','Happiness_Rank']]
df_ranks_change.rename(columns = {'Happiness_Rank':'2019'}, inplace = True)

df_ranks_change = df_ranks_change.merge(df2015_ranks, on='Country')
df_ranks_change['change']=df_ranks_change['2015']-df_ranks_change['2019']
df_ranks_change
```

```
C:\Users\r-sut\anaconda3\envs\TSM_InfVis\lib\site-
packages\pandas\core\frame.py:4304: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

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

[41]:

	Country	2019	2015	change
0	Finland	1	6	5
1	Denmark	2	3	1
2	Norway	3	4	1
3	Iceland	4	2	-2
4	Netherlands	5	7	2
..	…	…	…	…
144	Yemen	151	136	-15
145	Rwanda	152	154	2
146	Tanzania	153	146	-7
147	Afghanistan	154	153	-1
148	Central African Republic	155	148	-7

[149 rows x 4 columns]

[42]:
```
change_top10 = df_ranks_change.sort_values('change', ascending = False).head(10)
change_bottom10 = df_ranks_change.sort_values('change', ascending = True).
↪head(10)
```

4.1 Best performers: who jumped the most?

[43]:
```
change_top10
```

[43]:

	Country	2019	2015	change
98	Benin	102	155	53
95	Ivory Coast	99	151	52
57	Honduras	59	105	46
60	Hungary	62	104	42
100	Gabon	104	143	39
46	Romania	48	86	38
109	Burkina Faso	115	152	37
92	Cameroon	96	133	37
93	Bulgaria	97	134	37
99	Congo (Brazzaville)	103	139	36

4.1.1 Let's look at change in detail from 2015-2019 for Benin, and Honduras

Benin

[44]:
```
benin = df_all[df_all.Country == 'Benin']
benin
```

[44]:

	Country	Region	Happiness_Rank	Happiness_Score	\
154	Benin	Sub-Saharan Africa	155	3.340	
152	Benin	Sub-Saharan Africa	153	3.484	
137	Benin	Sub-Saharan Africa	143	3.657	
130	Benin	Sub-Saharan Africa	136	4.141	
98	Benin	Sub-Saharan Africa	102	4.883	

	GDP_per_capita	Social_Support	Life_Expectancy	Freedom	Corruption	\
154	0.286650	0.35386	0.31910	0.484500	0.080100	
152	0.394990	0.10419	0.21028	0.397470	0.066810	
137	0.431085	0.43530	0.20993	0.425963	0.060929	
130	0.378000	0.37200	0.24000	0.440000	0.067000	
98	0.393000	0.43700	0.39700	0.349000	0.082000	

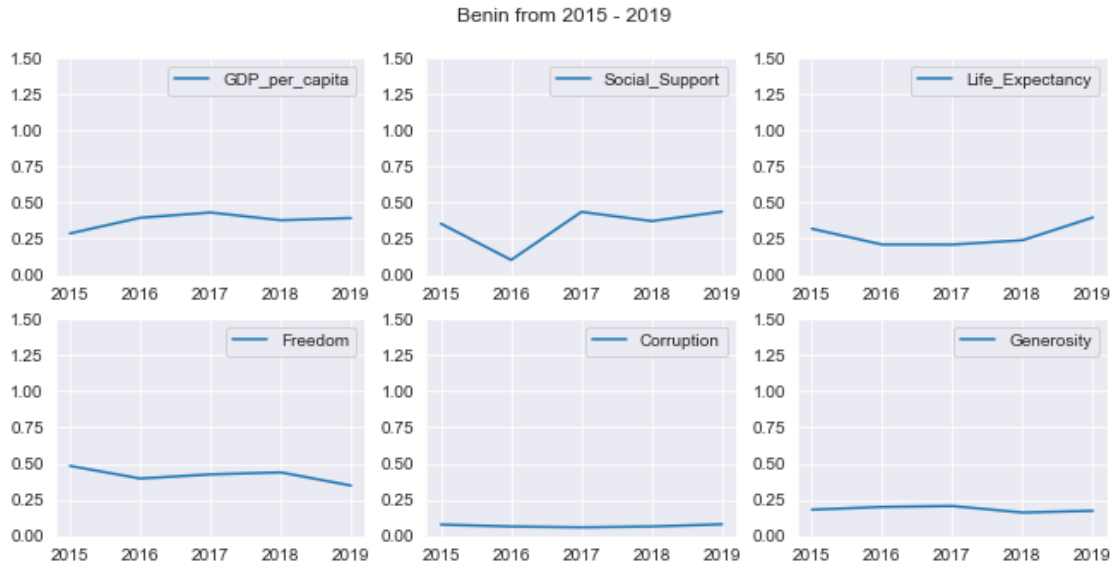
	Generosity	year
154	0.182600	2015
152	0.201800	2016
137	0.207948	2017
130	0.163000	2018
98	0.175000	2019

[45]:
```
plt.plot(benin.year, benin.Happiness_Rank);
plt.title('Development Happiness in Benin')
plt.xlabel('year')
plt.ylabel('Happiness Rank')
```

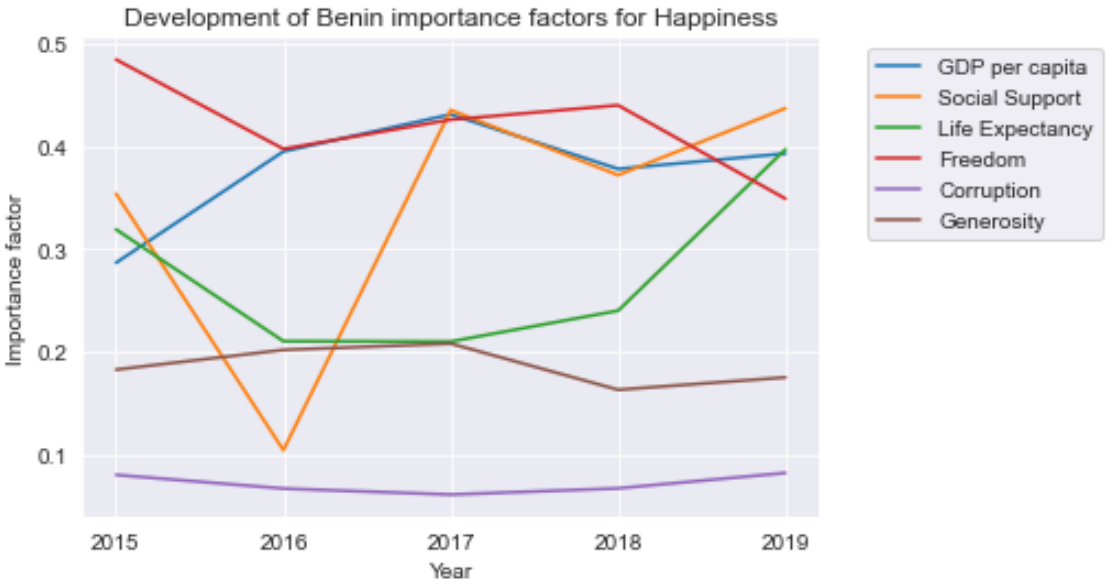
```
plt.gca().invert_yaxis()
plt.show();
```

[46]:
```
fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(11,5));

benin.plot(ax=axes[0,0], x = 'year', y = 'GDP_per_capita');
axes[0][0].set_ylim([0.0,1.5])
axes[0][0].set(xlabel=None)
benin.plot(ax=axes[0,1], x = 'year', y = 'Social_Support');
axes[0,1].set_ylim([0.0,1.5])
axes[0][1].set(xlabel=None)
benin.plot(ax=axes[0,2], x = 'year', y = 'Life_Expectancy');
axes[0,2].set_ylim([0.0,1.5])
axes[0][2].set(xlabel=None)
benin.plot(ax=axes[1,0], x = 'year', y = 'Freedom');
axes[1,0].set_ylim([0.0,1.5])
axes[1][0].set(xlabel=None)
benin.plot(ax=axes[1,1], x = 'year', y = 'Corruption');
axes[1,1].set_ylim([0.0,1.5])
axes[1][1].set(xlabel=None)
benin.plot(ax=axes[1,2], x = 'year', y = 'Generosity');
axes[1,2].set_ylim([0.0,1.5])
axes[1][2].set(xlabel=None)
fig.suptitle('Benin from 2015 - 2019', fontsize=12)
```

```
plt.subplots_adjust(top=0.9)
#fig.tight_layout()
plt.show()
```



[47]:
```
plt.figure()
plt.plot(benin.year, benin.GDP_per_capita, label='GDP per capita')
plt.plot(benin.year, benin.Social_Support, label='Social Support')
plt.plot(benin.year, benin.Life_Expectancy, label='Life Expectancy')
plt.plot(benin.year, benin.Freedom, label='Freedom')
plt.plot(benin.year, benin.Corruption, label='Corruption')
plt.plot(benin.year, benin.Generosity, label='Generosity')
plt.xlabel('Year')
plt.ylabel('Importance factor')
plt.legend(bbox_to_anchor=(1.4,1), loc='upper right')
plt.title('Development of Benin importance factors for Happiness')

fig.tight_layout()
plt.show()
```



and now honduras

[48]:
```
honduras = df_all[df_all.Country == 'Honduras']
honduras
```

[48]:

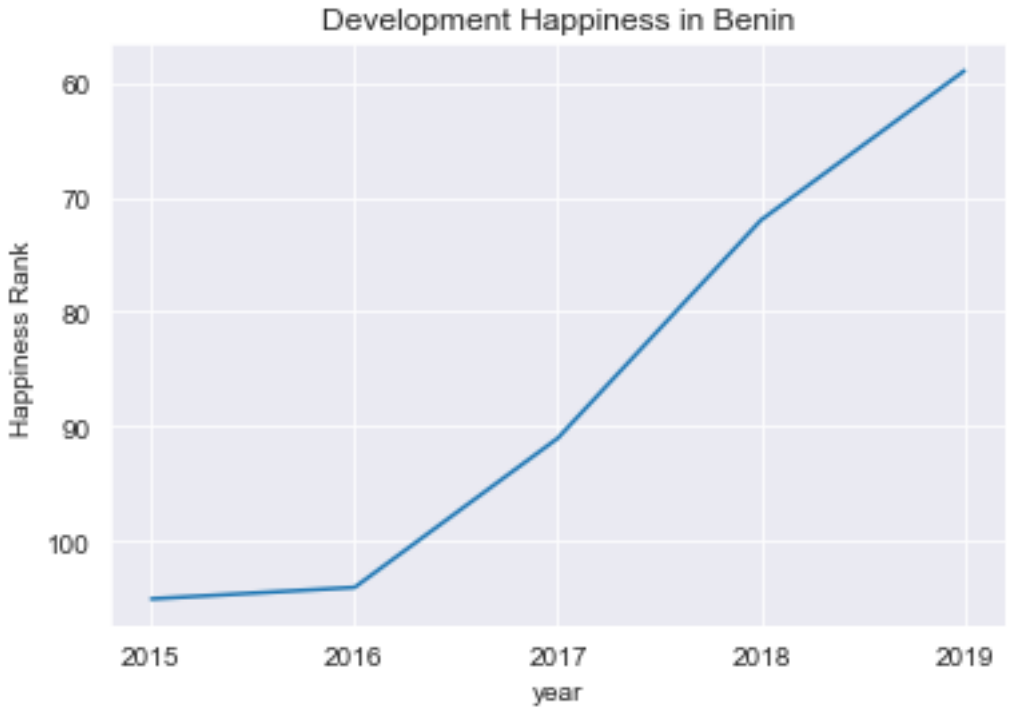
	Country	Region	Happiness_Rank	Happiness_Score	\
104	Honduras	Latin America and Caribbean	105	4.788	
103	Honduras	Latin America and Caribbean	104	4.871	
87	Honduras	Latin America and Caribbean	91	5.181	
68	Honduras	Latin America and Caribbean	72	5.504	
57	Honduras	Latin America and Caribbean	59	5.860	

	GDP_per_capita	Social_Support	Life_Expectancy	Freedom	Corruption	\
104	0.595320	0.953480	0.695100	0.40148	0.068250	
103	0.694290	0.755960	0.583830	0.26755	0.069060	
87	0.730573	1.143945	0.582569	0.34808	0.073345	
68	0.620000	1.205000	0.622000	0.45900	0.074000	
57	0.642000	1.236000	0.828000	0.50700	0.078000	

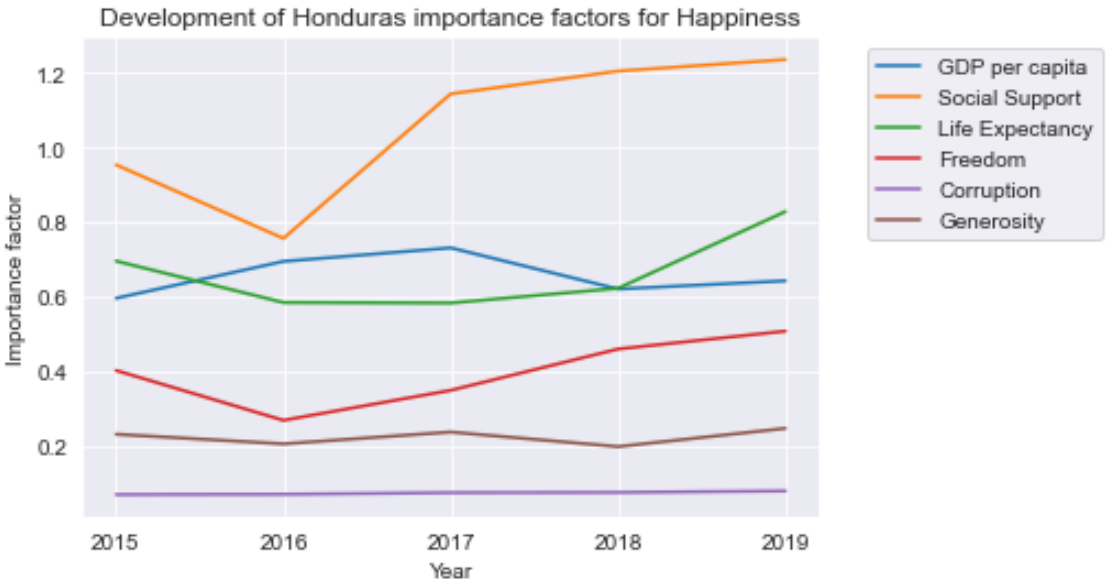
	Generosity	year
104	0.230270	2015
103	0.204400	2016
87	0.236189	2017
68	0.197000	2018
57	0.246000	2019

[49]:
```
plt.plot(honduras.year, honduras.Happiness_Rank);
plt.title('Development Happiness in Benin')
```

```
plt.xlabel('year')
plt.ylabel('Happiness Rank')
plt.gca().invert_yaxis()
plt.show();
```



[50]:
```
plt.figure()
plt.plot(honduras.year, honduras.GDP_per_capita, label='GDP per capita')
plt.plot(honduras.year, honduras.Social_Support, label='Social Support')
plt.plot(honduras.year, honduras.Life_Expectancy, label='Life Expectancy')
plt.plot(honduras.year, honduras.Freedom, label='Freedom')
plt.plot(honduras.year, honduras.Corruption, label='Corruption')
plt.plot(honduras.year, honduras.Generosity, label='Generosity')
plt.xlabel('Year')
plt.ylabel('Importance factor')
plt.legend(bbox_to_anchor=(1.4,1), loc='upper right')
plt.title('Development of Honduras importance factors for Happiness')

plt.show()
```



4.2 Worst performers: who slipped the most?

[51]:
```
change_bottom10
```

[51]:

	Country	2019	2015	change
104	Venezuela	108	23	-85
131	Zambia	138	85	-53
137	Lesotho	144	97	-47
128	Swaziland	135	101	-34
139	Zimbabwe	146	115	-31
116	Mozambique	123	94	-29
140	Haiti	147	119	-28
134	Liberia	141	116	-25
133	India	140	117	-23
126	Ukraine	133	111	-22

There are quite heavy changes regarding happiness rank, performing the wors being Venezuela, with the country being in a long crisis.

4.3 Let's look at change in detail from 2015-2019 for Venezuela, and Ukraine

4.3.1 Venezuela

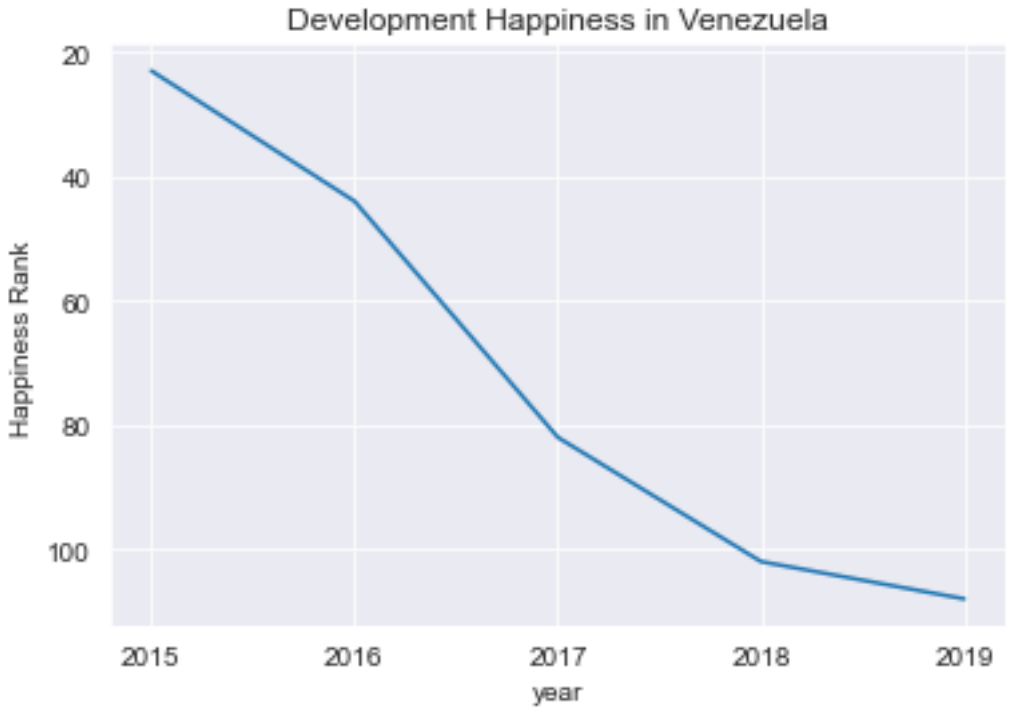
[52]:
```
venezuela = df_all[df_all.Country == 'Venezuela']
venezuela
```

[52]:

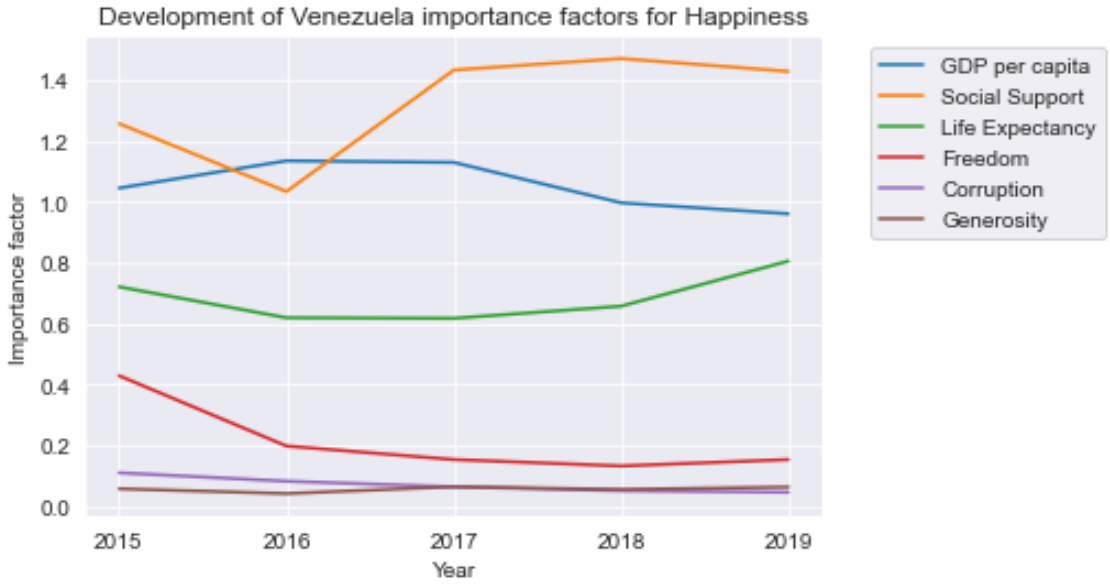
	Country	Region	Happiness_Rank	Happiness_Score	\
22	Venezuela	Latin America and Caribbean	23	6.810	

```
43   Venezuela  Latin America and Caribbean              44            6.084
78   Venezuela  Latin America and Caribbean              82            5.250
97   Venezuela  Latin America and Caribbean             102            4.806
104  Venezuela  Latin America and Caribbean             108            4.707

     GDP_per_capita  Social_Support  Life_Expectancy   Freedom  Corruption  \
22         1.044240        1.255960         0.720520  0.429080    0.110690
43         1.133670        1.033020         0.619040  0.198470    0.083040
78         1.128431        1.431338         0.617144  0.153997    0.064491
97         0.996000        1.469000         0.657000  0.133000    0.052000
104        0.960000        1.427000         0.805000  0.154000    0.047000

     Generosity  year
22      0.05841  2015
43      0.04250  2016
78      0.06502  2017
97      0.05600  2018
104     0.06400  2019
```

[53]:
```
plt.plot(venezuela.year, venezuela.Happiness_Rank);
plt.gca().invert_yaxis()
plt.title('Development Happiness in Venezuela')
plt.xlabel('year')
plt.ylabel('Happiness Rank')
plt.show();
```



[54]:
```
plt.figure()
plt.plot(venezuela.year, venezuela.GDP_per_capita, label='GDP per capita')
plt.plot(venezuela.year, venezuela.Social_Support, label='Social Support')
plt.plot(venezuela.year, venezuela.Life_Expectancy, label='Life Expectancy')
plt.plot(venezuela.year, venezuela.Freedom, label='Freedom')
plt.plot(venezuela.year, venezuela.Corruption, label='Corruption')
plt.plot(venezuela.year, venezuela.Generosity, label='Generosity')
plt.xlabel('Year')
plt.ylabel('Importance factor')
plt.legend(bbox_to_anchor=(1.4,1), loc='upper right')
plt.title('Development of Venezuela importance factors for Happiness')
plt.show()
```



Analysis of development in Venezuela

The Crisis in Venezuela during the Bolivarian Revolution is an ongoing socioeconomic and political crisis that began in Venezuela on 2 June 2010 during the presidency of Hugo Chávez and continues into the presidency of Nicolás Maduro. It is marked by hyperinflation, escalating starvation, disease, crime and mortality rates, resulting in massive emigration from the country. According to economists interviewed by The New York Times, the situation is the worst economic crisis in Venezuela's history and the worst facing a country in peace time since the mid-20th century and is more severe than that of the United States during the Great Depression, of the 1985–1994 Brazilian economic crisis, or of the 2008–2009 hyperinflation in Zimbabwe. Other American writers have also compared aspects of the crisis such as unemployment and GDP contraction to Bosnia and Herzegovina after the 1992–1995 Bosnian War as well as Russia, Cuba and Albania following the collapse of the Eastern Bloc in 1989.

[source: https://en.wikipedia.org/wiki/Crisis_in_Venezuela_during_the_Bolivarian_Revolution]

4.3.2 Ukraine

[55]:
```
ukraine = df_all[df_all.Country == 'Ukraine']
ukraine
```

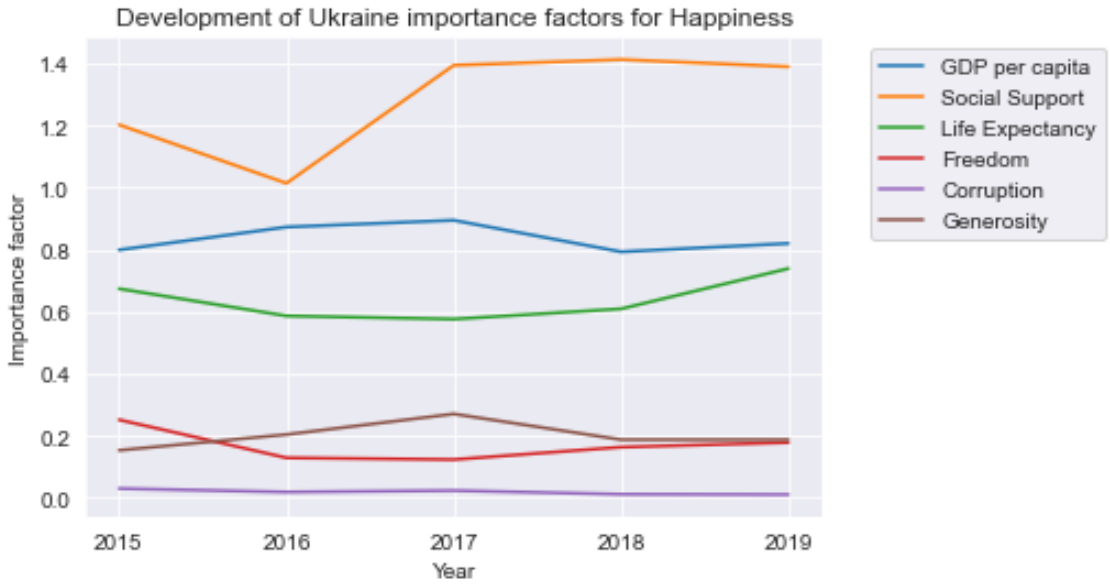
[55]:

	Country	Region	Happiness_Rank	Happiness_Score	GDP_per_capita	Social_Support	Life_Expectancy	Freedom	Corruption	Generosity	year
110	Ukraine	Central and Eastern Europe	111	4.681	0.799070	1.202780	0.673900	0.251230	0.029610	0.152750	2015
122	Ukraine	Central and Eastern Europe	123	4.324	0.872870	1.014130	0.586280	0.128590	0.018290	0.203630	2016
126	Ukraine	Central and Eastern Europe	132	4.096	0.894652	1.394538	0.575904	0.122975	0.023029	0.270061	2017
132	Ukraine	Central and Eastern Europe	138	4.103	0.793000	1.413000	0.609000	0.163000	0.011000	0.187000	2018
126	Ukraine	Central and Eastern Europe	133	4.332	0.820000	1.390000	0.739000	0.178000	0.010000	0.187000	2019

[56]:
```
plt.plot(ukraine.year, ukraine.Happiness_Rank);
plt.gca().invert_yaxis()
plt.title('Development Happiness in Ukraine')
plt.xlabel('year')
plt.ylabel('Happiness Rank')
plt.show();
```

[57]:
```
fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(11,5));

ukraine.plot(ax=axes[0,0], x = 'year', y = 'GDP_per_capita');
axes[0,0].set_ylim([0.0,1.5])
axes[0,0].set(xlabel=None)
ukraine.plot(ax=axes[0,1], x = 'year', y = 'Social_Support');
axes[0,1].set_ylim([0.0,1.5])
axes[0,1].set(xlabel=None)
ukraine.plot(ax=axes[0,2], x = 'year', y = 'Life_Expectancy');
axes[0,2].set_ylim([0.0,1.5])
axes[0,2].set(xlabel=None)
ukraine.plot(ax=axes[1,0], x = 'year', y = 'Freedom');
axes[1,0].set_ylim([0.0,1.5])
axes[1,0].set(xlabel=None)
ukraine.plot(ax=axes[1,1], x = 'year', y = 'Corruption');
axes[1,1].set_ylim([0.0,1.5])
axes[1,1].set(xlabel=None)
ukraine.plot(ax=axes[1,2], x = 'year', y = 'Generosity');
axes[1,2].set_ylim([0.0,1.5])
axes[1,2].set(xlabel=None)
fig.suptitle('Ukraine from 2015 - 2019', fontsize=12)
plt.subplots_adjust(top=0.9)
#fig.tight_layout()
plt.show()
```

[58]:
```
plt.figure()
plt.plot(ukraine.year, ukraine.GDP_per_capita, label='GDP per capita')
plt.plot(ukraine.year, ukraine.Social_Support, label='Social Support')
plt.plot(ukraine.year, ukraine.Life_Expectancy, label='Life Expectancy')
plt.plot(ukraine.year, ukraine.Freedom, label='Freedom')
plt.plot(ukraine.year, ukraine.Corruption, label='Corruption')
plt.plot(ukraine.year, ukraine.Generosity, label='Generosity')
plt.xlabel('Year')
plt.ylabel('Importance factor')
plt.legend(bbox_to_anchor=(1.4,1), loc='upper right')
plt.title('Development of Ukraine importance factors for Happiness')
plt.show()
```



Analysis of development in Ukraine

In March of 2014, the current crisis erupted when Russian special forces occupied Ukraine's Crimean peninsula. Russia claimed it was protecting its port access to the Black Sea. Between 2014–2018, a military conflict between Ukrainian soldiers and Russian-backed separatists continued in eastern Ukraine. More than 10,000 people were killed. The Ukraine crisis is a power struggle between factions within Ukraine. One of the factions wants to align with the European Union and the other with Russia.

This not yet solved conflict is clearly the reason for the development of life satisfaction in the Ukraine.

[source: https://www.thebalance.com/ukraine-crisis-summary-and-explanation-3970462]

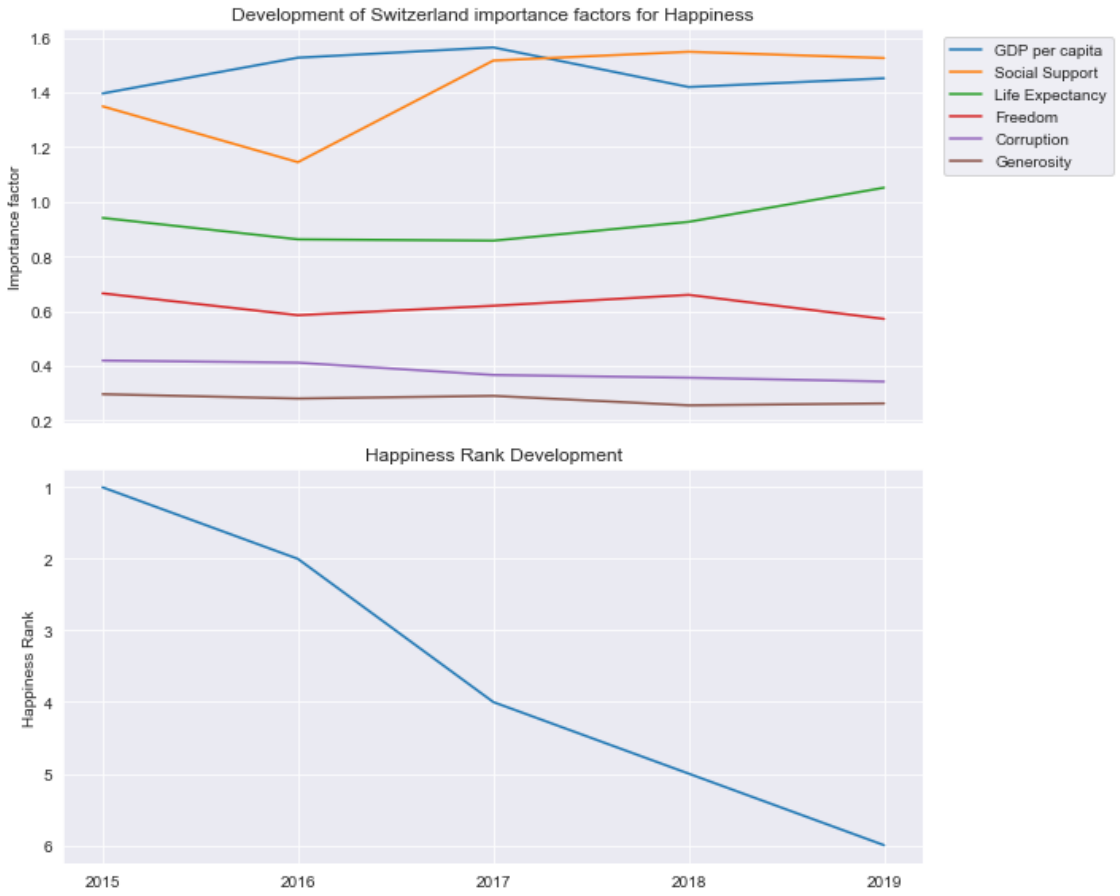
5 Development of Switzerland from 2015-2019

[59]:
```
switzerland = df_all[df_all.Country == 'Switzerland']
switzerland = switzerland.drop(columns=['Country', 'Region'])
switzerland
```

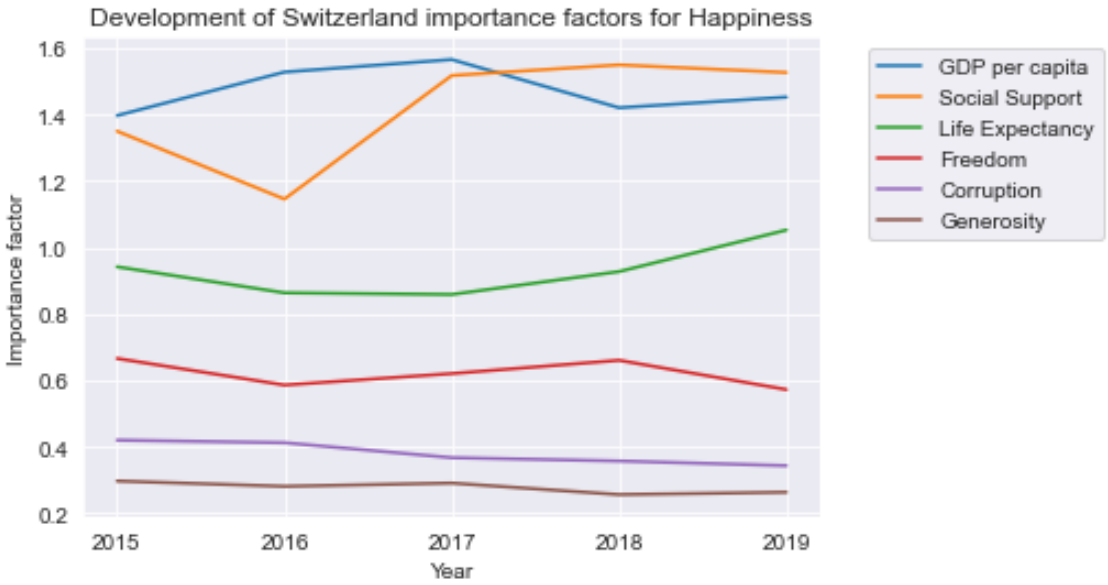
[59]:
```
    Happiness_Rank  Happiness_Score  GDP_per_capita  Social_Support  \
0                1            7.587         1.39651        1.349510
1                2            7.509         1.52733        1.145240
3                4            7.494         1.56498        1.516912
4                5            7.487         1.42000        1.549000
5                6            7.480         1.45200        1.526000

   Life_Expectancy   Freedom  Corruption  Generosity  year
```

```
0          0.941430  0.665570    0.419780    0.296780  2015
1          0.863030  0.585570    0.412030    0.280830  2016
3          0.858131  0.620071    0.367007    0.290549  2017
4          0.927000  0.660000    0.357000    0.256000  2018
5          1.052000  0.572000    0.343000    0.263000  2019
```

[60]:
```
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8), sharex = True)

ax1.plot(switzerland.year, switzerland.GDP_per_capita, label='GDP per capita')
ax1.plot(switzerland.year, switzerland.Social_Support, label='Social Support')
ax1.plot(switzerland.year, switzerland.Life_Expectancy, label='Life Expectancy')
ax1.plot(switzerland.year, switzerland.Freedom, label='Freedom')
ax1.plot(switzerland.year, switzerland.Corruption, label='Corruption')
ax1.plot(switzerland.year, switzerland.Generosity, label='Generosity')
ax1.legend(bbox_to_anchor=(1.23,1), loc='upper right')
ax1.title.set_text('Development of Switzerland importance factors for␣
 ↪Happiness')
ax1.set_ylabel('Importance factor')

ax2.plot(switzerland.year, switzerland.Happiness_Rank)
ax2.invert_yaxis()
ax2.title.set_text('Happiness Rank Development')
ax2.set_ylabel('Happiness Rank')

fig.tight_layout()
plt.savefig("Switzerland.png")
plt.show()
```



[61]:
```
#fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(11,5));
plt.figure()
plt.plot(switzerland.year, switzerland.GDP_per_capita, label='GDP per capita')
plt.plot(switzerland.year, switzerland.Social_Support, label='Social Support')
plt.plot(switzerland.year, switzerland.Life_Expectancy, label='Life Expectancy')
plt.plot(switzerland.year, switzerland.Freedom, label='Freedom')
plt.plot(switzerland.year, switzerland.Corruption, label='Corruption')
plt.plot(switzerland.year, switzerland.Generosity, label='Generosity')
plt.xlabel('Year')
plt.ylabel('Importance factor')
plt.legend(bbox_to_anchor=(1.4,1), loc='upper right')
plt.title('Development of Switzerland importance factors for Happiness')
plt.show()
```



5.0.1 Analysis of development in Switzerland from 2015-2019

The visualizations of the different factors for the happiness calculations in Switzerland between 2015 and 2019 show an estimate on the importance of these different factors for each year.

General Analysis:

The two most important factors are 'GDP per capita' and 'Social Support' which makes sense for a well functioning country with a lot of socialitarian structures. Corruption and Generosity are a pretty low importance factor, since most of the population doesn't feel to either be suppressed by the government or has to rely on others to survive.

Life Expectancy and Freedom are in between of these 4 factors and are somewhat important but are overall not threatening for most people and therefore not as important as Social Support or money itself.

Yearly Development: The most unstable factor is Social Support which proves to be an important factor and rose over the years but also experienced a downfall from 2015 to 2016 (Google why)

The stable income for Swiss is important and has been overall stable. Money will probably always be of importance for Switzerland and its Population.

Life Expectancy is rising overall. (Why is health getting more important?)

Freedom, Generosity and Corruption are almost stable and only fall off slowly. The analysis for this is that these factors didn't prove to have any negative influences over the years and therefore fall off slowly.