ABSTRACT

Data mining is a process used by companies to turn raw data into useful information. By using software to look for patterns in large batches of data, businesses can learn more about their customers to develop more effective marketing strategies, increase sales and decrease costs. Data mining depends on effective data collection, warehousing, and computer processing, artificial intelligence (e.g., machine learning) and business intelligence. Data mining involves exploring and analysing large blocks of information to glean meaningful patterns and trends. It can be used in a variety of ways, such as database marketing, credit risk management, fraud detection, spam Email filtering, or even to discern the sentiment or opinion of users.

The data mining process breaks down into five steps. First, organizations collect data and load it into their data warehouses. Next, they store and manage the data, either on in-house servers or the cloud. Business analysts, management teams and information technology professionals access the data and determine how they want to organize it. Then, application software sorts the data based on the user's results, and finally, the end-user presents the data in an easy-to-share format, such as a graph or table

In this project we have performed data mining on a dataset of World Happiness and predicted some important results related to it. The World Happiness Report is a landmark survey of the state of global happiness. The first report was published in 2012, the second in 2013, the third in 2015, and the fourth in the 2016 Update. The World Happiness 2017, which ranks 155 countries by their happiness levels, was released at the United Nations at an event celebrating International Day of Happiness on March 20th. The report continues to gain global recognition as governments, organizations and civil society increasingly use happiness indicators to inform their policy-making decisions. Leading experts across fields – economics, psychology, survey analysis, national statistics, health, public policy and more – describe how measurements of well-being can be used effectively to assess the progress of nations. The reports review the state of happiness in the world today and show how the new science of happiness explains personal and national variations in happiness.

1.1 Project Description

The World Happiness Report is a landmark survey of the state of global happiness. The first report was published in 2012, the second in 2013, the third in 2015, and the fourth in the 2016 Update. The World Happiness 2017, which ranks 155 countries by their happiness levels, was released at the United Nations at an event celebrating International Day of Happiness on March 20th. The report continues to gain global recognition as governments, organizations and civil society increasingly use happiness indicators to inform their policy-making decisions. Leading experts across fields – economics, psychology, survey analysis, national statistics, health, public policy and more – describe how measurements of well-being can be used effectively to assess the progress of nations. The reports review the state of happiness in the world today and show how the new science of happiness explains personal and national variations in happiness.

The happiness scores and rankings use data from the Gallup World Poll. The scores are based on answers to the main life evaluation question asked in the poll. This question, known as the Cantril ladder, asks respondents to think of a ladder with the best possible life for them being a 10 and the worst possible life being a 0 and to rate their own current lives on that scale. The scores are from nationally representative samples for the years 2013-2016 and use the Gallup weights to make the estimates representative. The columns following the happiness score estimate the extent to which each of six factors – economic production, social support, life expectancy, freedom, absence of corruption, and generosity – contribute to making life evaluations higher in each country than they are in Dystopia, a hypothetical country that has values equal to the world's lowest national averages for each of the six factors. They have no impact on the total score reported for each country, but they do explain why some countries rank higher than others.

This notebook is inspired from the Kaggle kernel: https://www.kaggle.com/javadzabihi/happiness-2017-visualization-prediction in which Visualization + Prediction is done using R language. In this notebook Visualization + Prediction is done using Python language.

Dataset - Dataset is taken from https://www.kaggle.com/unsdsn/world-happiness but updated with a column named "Continent".

This project is also submitted at this Kaggle kernel - https://www.kaggle.com/gurkanwalkang/happiness-visualization-prediction-using-python

How it works?

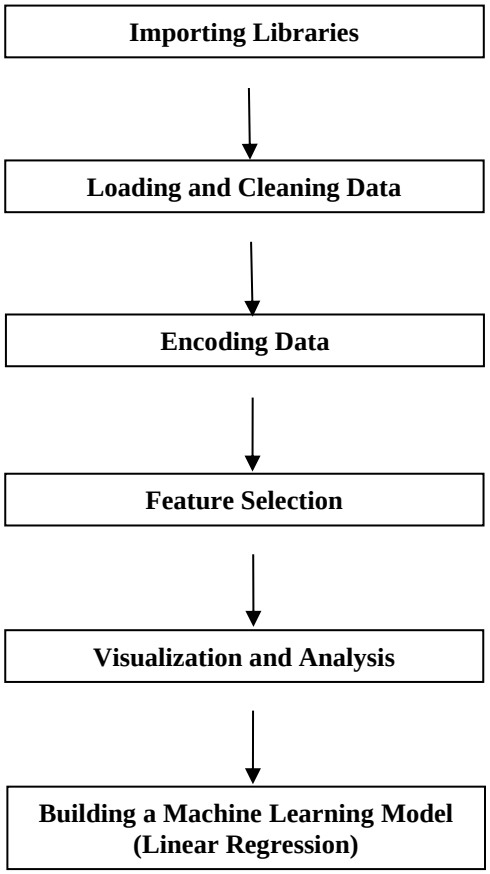


Figure 8. Workflow of World Happiness Report

1.2 Importing Libraries

First load the required libraries.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from math import sqrt
```

1.3 Loading Data

Now, it's moment to load data, here I have used that location where dataset is stored in my system. The name of dataset is "2017.csv". For displaying in more representative way it is converted to data frame using Pandas library of Python after processing data.

```
In [2]: data = pd.DataFrame(pd.read_csv("world_happiness.csv"))
        data.head()
```

Out[2]:

	Continent	Country	Happiness.Rank	Happiness.Score	Whisker.high	Whisker.low	Economy..GDP.per.Capita.	Family	Health..Life.Expectancy.	Freedom
0	Europe	Norway	1	7.537	7.594445	7.479556	1.616463	1.533524	0.796667	0.635423
1	Europe	Denmark	2	7.522	7.581728	7.462272	1.482383	1.551122	0.792566	0.626007
2	Europe	Iceland	3	7.504	7.622030	7.385970	1.480633	1.610574	0.833552	0.627163
3	Europe	Switzerland	4	7.494	7.561772	7.426227	1.564980	1.516912	0.858131	0.620071
4	Europe	Finland	5	7.469	7.527542	7.410458	1.443572	1.540247	0.809158	0.617951

1.4 Cleaning Data

Data Cleansing is a form of data management. Over time, persons and businesses mount up a lot of personal information! Ultimately, information becomes out-of-date. For example, over 10 years you may transform your address, or your name, and then change your address again! Data cleansing is a method in which you go through all of the data within a database and either eradicate or modernize information that is incomplete, mistaken, improperly formatted, irrelevant, or duplicated. Data cleansing usually involves cleaning up data compiled in one area. For example, data from a single spreadsheet.

Though data cleansing does and can engross deleting information, it is focused more on updating, correcting, and consolidating data to ensure your system is as effective as possible.

```
In [3]: data.columns = data.columns.str.strip().str.lower().str.replace('.', '_')
        data.head(2)
```

Out[3]:

	continent	country	happiness_rank	happiness_score	whisker_high	whisker_low	economy__gdp_per_capita_	family	health__life_expectancy_	free
0	Europe	Norway	1	7.537	7.594445	7.479556	1.616463	1.533524	0.796667	0.63
1	Europe	Denmark	2	7.522	7.581728	7.462272	1.482383	1.551122	0.792566	0.62

```
In [4]: data.shape
```

Out[4]: (155, 13)

head() and .tail() - Head and tail functions are capable of 5 rows per time. But you can change this situation. So you can enter the desired value in the parameter section. The first function, ie head (), returns the initial values. The second function returns the last values.

Renaming some of the columns and removing unnecessary columns:

We have observed the variables inside our dataset, their class, and the first few observations of each. In fact, the dataset has 155 observations and 12 variables. I believe some of the variable

names are not clear enough and I decided to change the name of several of them a little bit. Also, I will remove whisker low and whisker high variables from my dataset because these variables give only the lower and upper confidence interval of happiness score and there is no need to use them for visualization and prediction.

1. Renaming some of the columns

```
In [5]: data.columns = ['continent','country', 'happiness_rank', 'happiness_score', 'whisker_high', 'whisker_low', 'economy','family'
        data.head(2)
```

Out[5]:

	continent	country	happiness_rank	happiness_score	whisker_high	whisker_low	economy	family	health	freedom	generosity	trust	dystc
0	Europe	Norway	1	7.537	7.594445	7.479556	1.616463	1.533524	0.796667	0.635423	0.362012	0.315964	
1	Europe	Denmark	2	7.522	7.581728	7.462272	1.482383	1.551122	0.792566	0.626007	0.355280	0.400770	

2. Removing unnecessary columns:

```
In [6]: data.drop(data.columns[[4,5]], axis = 1, inplace = True)
        data.head(2)
```

Out[6]:

	continent	country	happiness_rank	happiness_score	economy	family	health	freedom	generosity	trust	dystopia_residual
0	Europe	Norway	1	7.537	1.616463	1.533524	0.796667	0.635423	0.362012	0.315964	2.277027
1	Europe	Denmark	2	7.522	1.482383	1.551122	0.792566	0.626007	0.355280	0.400770	2.313707

describe() - Describe function includes analysis of all our numerical data. For this, count, mean, std, min,% 25,% 50,% 75, max values are given. The reason this section is important is that you can estimate the probability that the values found here are deviant data.

info() - Then, the data has what informations. We are learning the information for all data The info function shows the data types and numerical values of the features in our data set. In short, this information about our data set. :)

```
In [7]: data.shape
Out[7]: (155, 11)

In [8]: data.columns
Out[8]: Index(['continent', 'country', 'happiness_rank', 'happiness_score', 'economy',
               'family', 'health', 'freedom', 'generosity', 'trust',
               'dystopia_residual'],
              dtype='object')
```

```
In [9]: data.describe()
```

Out[9]:

	happiness_rank	happiness_score	economy	family	health	freedom	generosity	trust	dystopia_residual
count	155.000000	155.000000	155.000000	155.000000	155.000000	155.000000	155.000000	155.000000	155.000000
mean	78.000000	5.354019	0.984718	1.188898	0.551341	0.408786	0.246883	0.123120	1.850238
std	44.888751	1.131230	0.420793	0.287263	0.237073	0.149997	0.134780	0.101661	0.500028
min	1.000000	2.693000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.377914
25%	39.500000	4.505500	0.663371	1.042635	0.369866	0.303677	0.154106	0.057271	1.591291
50%	78.000000	5.279000	1.064578	1.253918	0.606042	0.437454	0.231538	0.089848	1.832910
75%	116.500000	6.101500	1.318027	1.414316	0.723008	0.516561	0.323762	0.153296	2.144654
max	155.000000	7.537000	1.870766	1.610574	0.949492	0.658249	0.838075	0.464308	3.117485

```
In [10]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 155 entries, 0 to 154
Data columns (total 11 columns):
continent             155 non-null object
country               155 non-null object
happiness_rank        155 non-null int64
happiness_score       155 non-null float64
economy               155 non-null float64
family                155 non-null float64
health                155 non-null float64
freedom               155 non-null float64
generosity            155 non-null float64
trust                 155 non-null float64
dystopia_residual     155 non-null float64
dtypes: float64(8), int64(1), object(2)
memory usage: 13.4+ KB
```

```
In [11]: print('Number of Null values in Columns')
         data.isnull().sum()
```

```
Number of Null values in Columns
```

Out[11]:

```
continent             0
country               0
happiness_rank        0
happiness_score       0
economy               0
family                0
health                0
freedom               0
generosity            0
trust                 0
dystopia_residual     0
dtype: int64
```

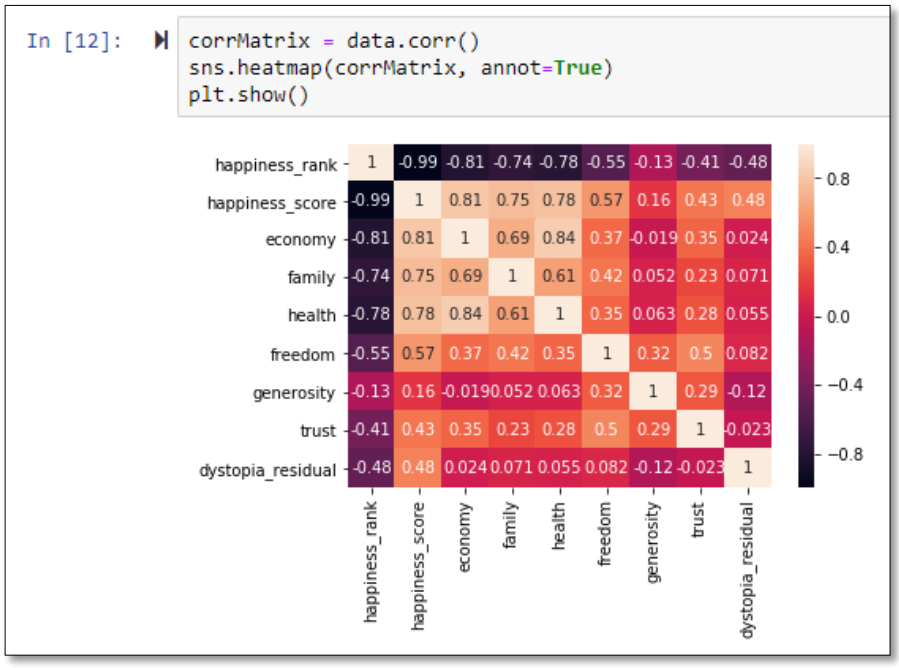
No null values in the entire data set

There are no null values in any column.

1.5 Visualization

In this section, we will play with different variables to find out how they correlate with each other.

Correlation plot:



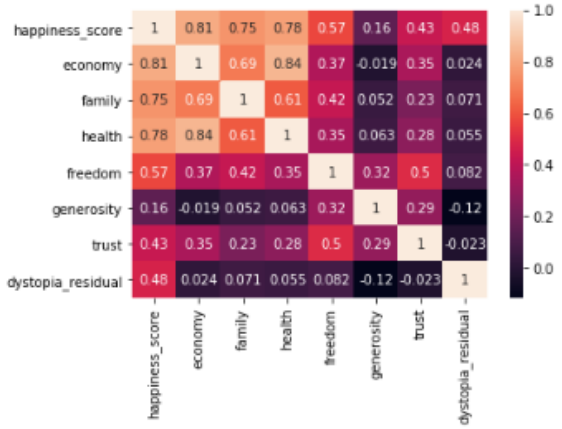
Obviously, there is an inverse correlation between “Happiness Rank” and all the other numerical variables. In other words, the lower the happiness rank, the higher the happiness score, and the higher the other seven factors that contribute to happiness. So let’s remove the happiness rank, and see the correlation again.

```
In [13]:
data.drop(data.columns[[2]], axis = 1, inplace = True)
data.head(2)
```

Out[13]:

	continent	country	happiness_score	economy	family	health	freedom	generosity	trust	dystopia_residual
0	Europe	Norway	7.537	1.616463	1.533524	0.796667	0.635423	0.362012	0.315964	2.277027
1	Europe	Denmark	7.522	1.482383	1.551122	0.792566	0.626007	0.355280	0.400770	2.313707

```
In [14]:
corrMatrix = data.corr()
sns.heatmap(corrMatrix, annot=True)
plt.show()
```



According to the above correlation plot, Economy, life expectancy, and family play the most significant role in contributing to happiness. Trust and generosity have the lowest impact on the happiness score

Correlation plot for each continent:

Before that creating separate data frames for each continent.

Asia:

```
In [16]:
#asia
asia = data["continent"] == "Asia"
asia_df = data[asia]
asia_data = asia_df.mean()
asia_mean = asia_data['happiness_score']
print("Mean of happiness score -", asia_mean)
asia_df.head(2)
```

```
Mean of happiness score - 5.2841249763499984
```

Out[16]:

	continent	country	happiness_score	economy	family	health	freedom	generosity	trust	dystopia_residual
10	Asia	Israel	7.213	1.375382	1.37629	0.838404	0.405989	0.330083	0.085242	2.801757
20	Asia	United Arab Emirates	6.648	1.626343	1.26641	0.726798	0.608345	0.360942	0.324490	1.734704

Africa:

In [17]:
```
#africa
africa = data["continent"] == "Africa"
africa_df = data[africa]
africa_data = africa_df.mean()
africa_mean = africa_data['happiness_score']
print("Mean of happiness score -", africa_mean)
africa_df.head(2)
```

```
Mean of happiness score - 4.376326531755102
```

Out[17]:

	continent	country	happiness_score	economy	family	health	freedom	generosity	trust	dystopia_residual
33	Africa	Spain	6.403	1.384398	1.532091	0.888961	0.408781	0.190134	0.070914	1.927758
52	Africa	Algeria	5.872	1.091864	1.146217	0.617585	0.233336	0.069437	0.146096	2.567604

Europe:

In [18]:
```
#europe
europe = data["continent"] == "Europe"
europe_df = data[europe]
europe_data = europe_df.mean()
europe_mean = europe_data['happiness_score']
print("Mean of happiness score -", europe_mean)
europe_df.head(2)
```

```
Mean of happiness score - 6.102225017600002
```

Out[18]:

	continent	country	happiness_score	economy	family	health	freedom	generosity	trust	dystopia_residual
0	Europe	Norway	7.537	1.616463	1.533524	0.796667	0.635423	0.362012	0.315964	2.277027
1	Europe	Denmark	7.522	1.482383	1.551122	0.792566	0.626007	0.355280	0.400770	2.313707

North America:

In [19]:
```
#north america
nm = data["continent"] == "North America"
nm_df = data[nm]
nm_data = nm_df.mean()
nm_mean = nm_data['happiness_score']
print("Mean of happiness score -", nm_mean)
nm_df.head(2)
```

```
Mean of happiness score - 6.028214301428572
```

Out[19]:

	continent	country	happiness_score	economy	family	health	freedom	generosity	trust	dystopia_residual
6	North America	Canada	7.316	1.479204	1.481349	0.834558	0.611101	0.435540	0.287372	2.187264
11	North America	Costa Rica	7.079	1.109706	1.416404	0.759509	0.580132	0.214613	0.100107	2.898639

South America:

In [20]:
```
#south america
sm = data["continent"] == "South America"
sm_df = data[sm]
sm_data = sm_df.mean()
sm_mean = sm_data['happiness_score']
print("Mean of happiness score -", sm_mean)
sm_df.head(2)
```

```
Mean of happiness score - 6.0986000061
```

Out[20]:

	continent	country	happiness_score	economy	family	health	freedom	generosity	trust	dystopia_residual
19	South America	Chile	6.652	1.252785	1.284025	0.819480	0.376895	0.326662	0.082288	2.509586
21	South America	Brazil	6.635	1.107353	1.431306	0.616552	0.437454	0.162350	0.111093	2.769267

Australia:

```
In [21]:
#Australia
aus = data["continent"] == "Australia"
aus_df = data[aus]
aus_data = aus_df.mean()
aus_mean = aus_data['happiness_score']
print("Mean of happiness score -", aus_mean)
aus_df.head(2)
```

```
Mean of happiness score - 7.299000025
```

Out[21]:

	continent	country	happiness_score	economy	family	health	freedom	generosity	trust	dystopia_residual
7	Australia	New Zealand	7.314	1.405706	1.548195	0.816760	0.614062	0.500005	0.382817	2.046456
9	Australia	Australia	7.284	1.484415	1.510042	0.843887	0.601607	0.477699	0.301184	2.065211

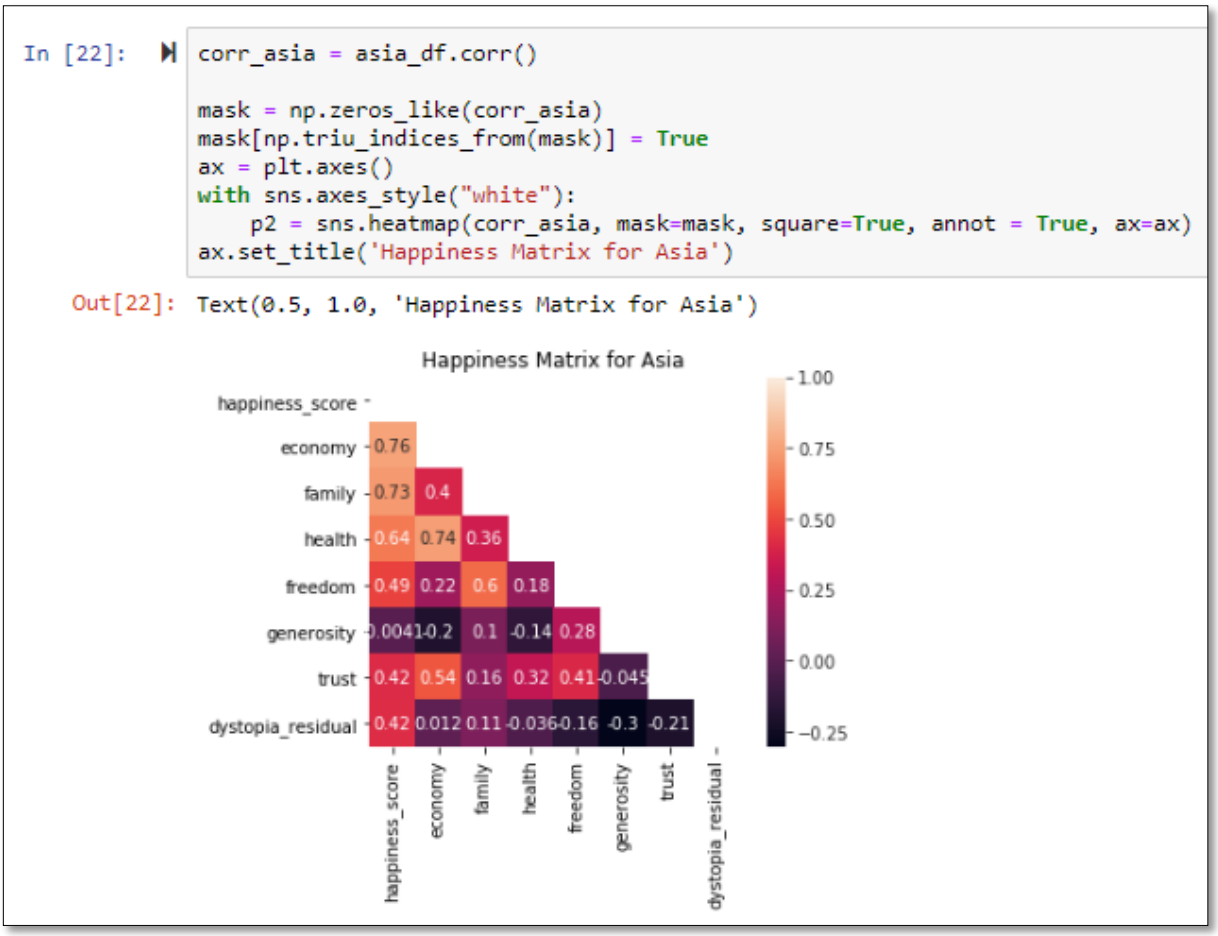
Correlation between "Happiness Score" and the other variables in Asia:

Economy > Family > Life.Expectancy > Freedom > Trust > Dystopia.Residual

There is no correlation between happiness score and generosity.

```
In [22]:
corr_asia = asia_df.corr()

mask = np.zeros_like(corr_asia)
mask[np.triu_indices_from(mask)] = True
ax = plt.axes()
with sns.axes_style("white"):
    p2 = sns.heatmap(corr_asia, mask=mask, square=True, annot = True, ax=ax)
ax.set_title('Happiness Matrix for Asia')
```

```
Out[22]: Text(0.5, 1.0, 'Happiness Matrix for Asia')
```

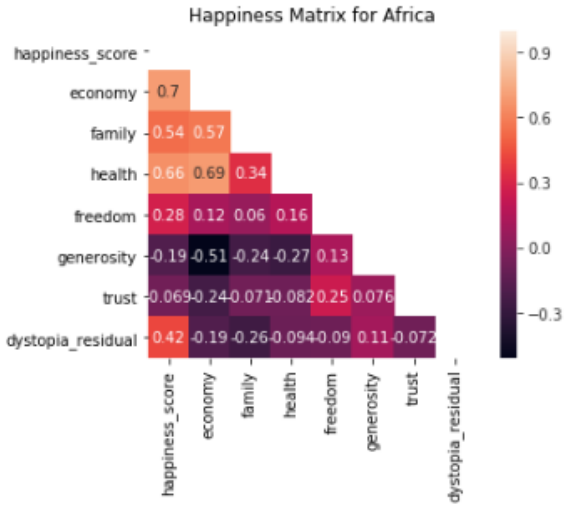


Correlation between "Happiness Score" and the other variables in Africa:

Economy > Family > Life.Expectancy > Dystopia.Residual > Freedom

There is no correlation between happiness score and trust. There is an inverse correlation between happiness score and generosity

```
In [23]: corr_africa = africa_df.corr()

         mask = np.zeros_like(corr_africa)
         mask[np.triu_indices_from(mask)] = True
         ax = plt.axes()
         with sns.axes_style("white"):
             p2 = sns.heatmap(corr_africa, mask=mask, square=True, annot = True, ax=ax)
         ax.set_title('Happiness Matrix for Africa')

Out[23]: Text(0.5, 1.0, 'Happiness Matrix for Africa')
```

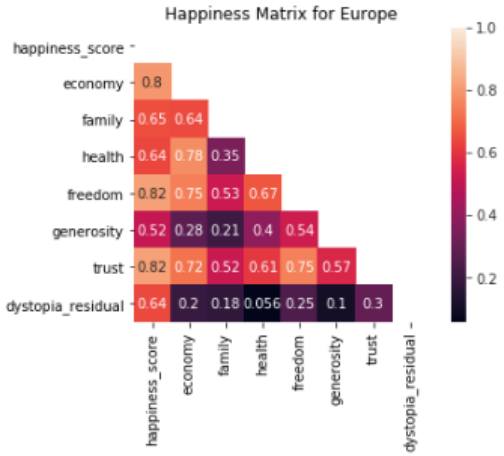


Correlation between "Happiness Score" and the other variables in Europe:

Freedom > Trust > Economy > Family > Dystopia.Residual > Life.Expectancy > Generosity

The highest correlation between generosity and happiness score took place in Europe.

```
In [24]: corr_europe = europe_df.corr()

         mask = np.zeros_like(corr_europe)
         mask[np.triu_indices_from(mask)] = True
         ax = plt.axes()
         with sns.axes_style("white"):
             p2 = sns.heatmap(corr_europe, mask=mask, square=True, annot = True, ax=ax)
         ax.set_title('Happiness Matrix for Europe')

Out[24]: Text(0.5, 1.0, 'Happiness Matrix for Europe')
```

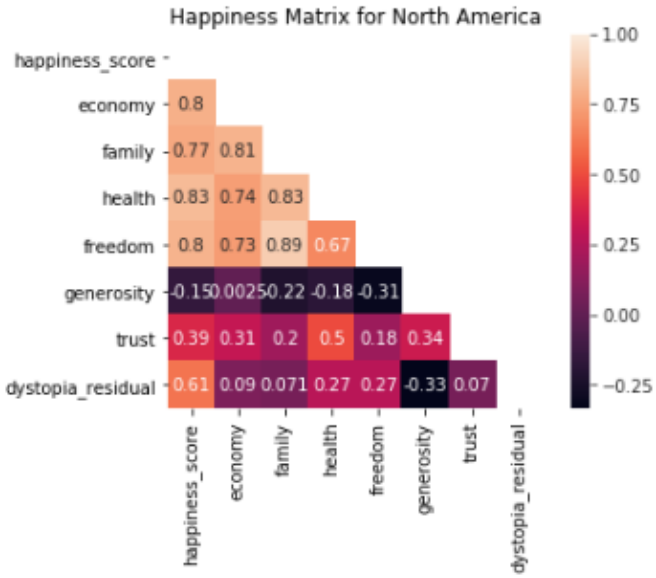


Correlation between "Happiness Score" and the other variables in North America:

Life.Expectancy > Economy > Freedom > Family > Dystopia.Residual > Trust

There is an inverse correlation between happiness score and generosity.

```
In [25]: corr_nm = nm_df.corr()

         mask = np.zeros_like(corr_nm)
         mask[np.triu_indices_from(mask)] = True
         ax = plt.axes()
         with sns.axes_style("white"):
             p2 = sns.heatmap(corr_nm, mask=mask, square=True, annot = True, ax=ax)
         ax.set_title('Happiness Matrix for North America')

Out[25]: Text(0.5, 1.0, 'Happiness Matrix for North America')
```

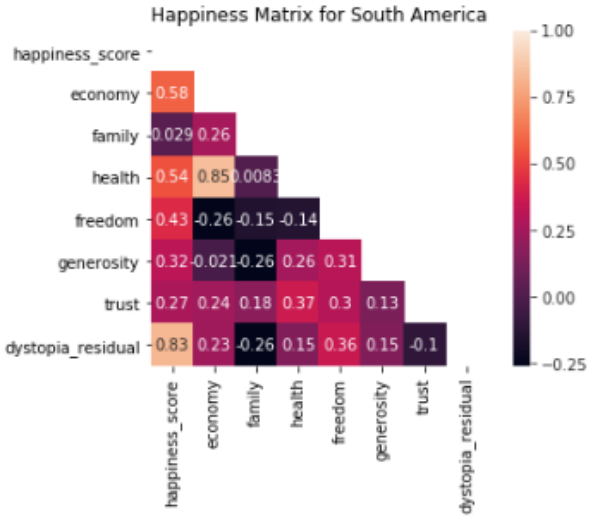


Correlation between "Happiness Score" and the other variables in South America:

Dystopia.Residual > Economy > Life.Expectancy > Freedom > Generosity > Trust > Family

The family is the least significant factor in South America

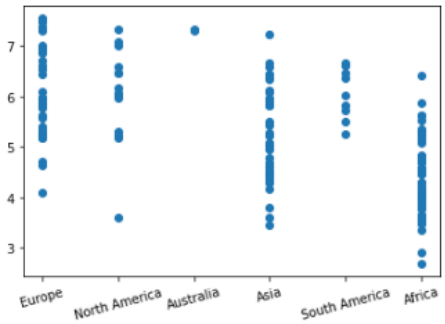
```
In [26]:
corr_sm = sm_df.corr()

mask = np.zeros_like(corr_sm)
mask[np.triu_indices_from(mask)] = True
ax = plt.axes()
with sns.axes_style("white"):
    p2 = sns.heatmap(corr_sm, mask=mask, square=True, annot = True, ax=ax)
ax.set_title('Happiness Matrix for South America')

Out[26]: Text(0.5, 1.0, 'Happiness Matrix for South America')
```



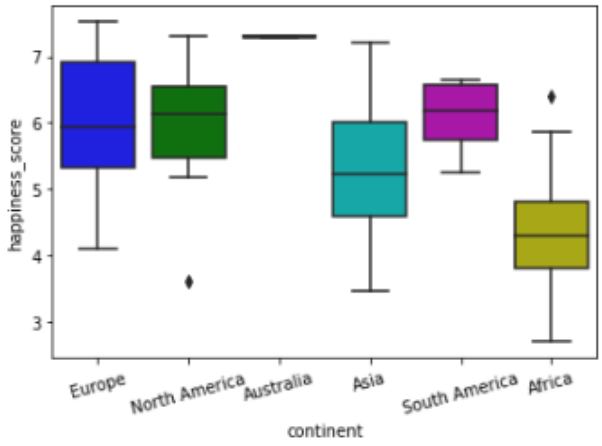
Happiness score comparison on different continents

We will use scatter plot, box plot, and violin plot to see the happiness score distribution in different countries, how this score is populated in these continents and also will calculate the mean and median of happiness score for each of these continents.

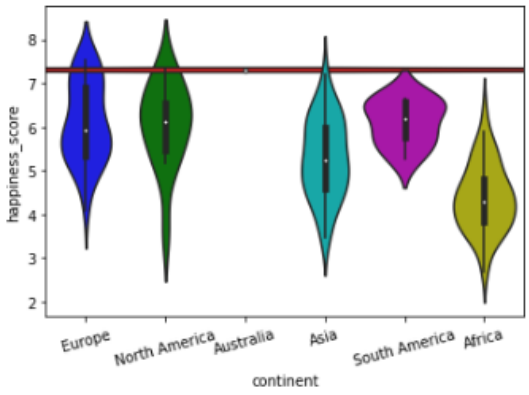
```
In [28]:
clr = {"Europe": "b", "North America": "g", "Australia":"r", "Asia": "c", "South America": "m", "Africa":"y"}

In [29]:
plt.scatter(data["continent"], data["happiness_score"], cmap=clr )
plt.xticks(rotation=15)

Out[29]: ([0, 1, 2, 3, 4, 5], <a list of 6 Text xticklabel objects>)
```



```
In [30]: sns.boxplot( x=data["continent"], y=data["happiness_score"], palette=clr )
         plt.xticks(rotation=15)

Out[30]: (array([0, 1, 2, 3, 4, 5]), <a list of 6 Text xticklabel objects>)
```




```
In [31]: sns.violinplot( x=data["continent"], y=data["happiness_score"], width=25, palette=clr )
         plt.xticks(rotation=15)

C:\Users\71guk\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futu
rray index, `arr[np.array(seq)]`, which will result either in an error or a different resu
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

Out[31]: (array([0, 1, 2, 3, 4, 5]), <a list of 6 Text xticklabel objects>)
```



Continent	Mean of happiness score	Median of happiness score
Africa	4.239500	4.1850
Asia	5.284721	5.2620
Australia	7.299000	7.2990
Europe	6.097929	5.9325
North America	6.028214	6.1195
South America	6.098600	6.1825

As we have seen before, Australia has the highest median happiness score. Europe, South America, and North America are in the second place regarding median happiness score. Asia

has the lowest median after Africa. We can see the range of happiness score for different continents, and also the concentration of happiness score.

```
In [84]:  grid = sns.pairplot(data, hue = 'continent', diag_kind = 'kde',
                              plot_kws = {'alpha': 0.6, 's': 80, 'edgecolor': 'k'},
                              size = 4)
          grid
          grid.savefig('PairPlot.png')
```

1.6 Encoding Data

In machine learning, we usually deal with datasets which contains multiple labels in one or more than one columns. These labels can be in the form of words or numbers. To make the data understandable or in human readable form, the training data is often labeled in words. Label Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

```
In [33]: data.head(2)
```

Out[33]:

	continent	country	happiness_score	economy	family	health	freedom	generosity	trust	dystopia_residual
0	Europe	Norway	7.537	1.616463	1.533524	0.796667	0.635423	0.362012	0.315964	2.277027
1	Europe	Denmark	7.522	1.482383	1.551122	0.792566	0.626007	0.355280	0.400770	2.313707

```
In [34]: #ENCODING
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
data[data.columns[0]] = labelencoder.fit_transform(data[data.columns[0]])
data[data.columns[1]] = labelencoder.fit_transform(data[data.columns[1]])
```

```
In [35]: data.head(2)
```

Out[35]:

	continent	country	happiness_score	economy	family	health	freedom	generosity	trust	dystopia_residual
0	3	104	7.537	1.616463	1.533524	0.796667	0.635423	0.362012	0.315964	2.277027
1	3	37	7.522	1.482383	1.551122	0.792566	0.626007	0.355280	0.400770	2.313707

1.7 Building the Model

Modeling in machine learning is an interactive phase where a data scientist constantly train and test machine learning models to find out the most excellent one for the given task.

Different machine learning models exist and the selection of which one to use generally depends on the problem at hand. No machine learning model works best for all kind of problems. So, our job in this stage is to test multiple models and fine tune parameters to compress out every bit of accuracy.

Here, we have used **Linear Regression** - as the output variable (Happiness Score) is a Continuous value.

Splitting Data:- In this step dataset is split into two parts, that is, training set and testing set.

- **Training Set:-** A subset to train model.
- **Testing Set:-** A subset to test the trained model.

You could picturize slicing the single data set as follows:

- **Predicting Happiness Score of a country**

```
In [36]: X = data[data.columns[[0,1,3,4,5,6,7,8,9]]]
         y = data[data.columns[2]]

In [37]: from sklearn.model_selection import train_test_split
         X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)

         reg=LinearRegression()

In [38]: reg.fit(X_train,y_train)

Out[38]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
                          normalize=False)
```

```
In [40]: pdt = reg.predict(X_test)
         pdt

Out[40]: array([6.071419  , 3.47064853, 4.49694209, 6.08368375, 3.79389656,
                7.53703128, 4.46557343, 5.22745217, 7.49427357, 5.83854038,
                4.19013666, 5.01065226, 5.52497072, 6.86277714, 3.87457228,
                6.35706543, 5.95564048, 4.53482526, 3.46215345, 6.95066582,
                3.60317994, 3.65685097, 5.32437925, 6.57763819, 4.44023948,
                5.26233293, 7.28361633, 5.62901359, 4.31496788, 6.16781067,
                5.3952197 ])
```

Accuracy of our model predicting happiness score is – 99.9%

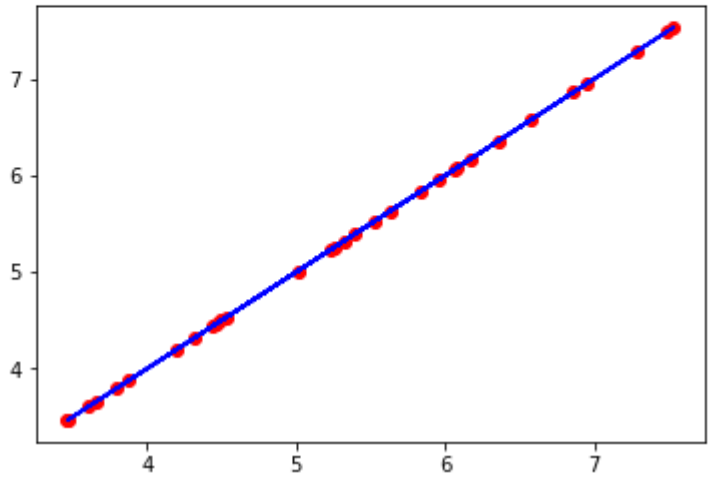
```
In [39]: reg.score(X_train,y_train)

Out[39]: 0.999999936858735
```

In [41]:
```
from sklearn.metrics import mean_squared_error
from math import sqrt

rms = sqrt(mean_squared_error(y_test,pdt))
rms
```

Out[41]:
```
0.00029603296172245303
```

In [72]:
```
plt.scatter(pdt, y_test, color = 'red')

# plot predicted data
plt.plot(pdt, y_test, color = 'blue')
```

Out[72]:
```
[<matplotlib.lines.Line2D at 0x1f01b565d30>]
```



CONCLUSION

- There is an inverse correlation between “Happiness Rank” and all the other numerical variables. In other words, *the lower the happiness rank, the higher the happiness score, and the higher the other seven factors that contribute to happiness.*
- Economy, life expectancy, and family play the most significant role in contributing to happiness. Trust and generosity have the lowest impact on the happiness score
- Economy > Family > Life.Expectancy > Freedom > Trust > Dystopia.Residual
  There is no correlation between happiness score and generosity.
- Economy > Family > Life.Expectancy > Dystopia.Residual > Freedom
- There is no correlation between happiness score and trust. There is an inverse correlation between happiness score and generosity
- Freedom > Trust > Economy > Family > Dystopia.Residual > Life.Expectancy > Generosity
  The highest correlation between generosity and happiness score took place in Europe.
- Life.Expectancy > Economy > Freedom > Family > Dystopia.Residual > Trust
  There is an inverse correlation between happiness score and generosity.
- Dystopia.Residual > Economy > Life.Expectancy > Freedom > Generosity > Trust > Family
  The family is the least significant factor in South America
- Australia has the highest median happiness score. Europe, South America, and North America are in the second place regarding median happiness score. Asia has the lowest median after Africa.
- We have seen the range of happiness score for different continents, and also the concentration of happiness score.

REFERENCES

- www.kaggle.com
- https://www.kaggle.com/unsdsn/world-happiness
- www.datacamp.com
- www.towardsdatascence.com
- www.machinelearningmastery.com