Jared Faucher

CSCI E-83 – Final Project

5/13/2018

What matters most when it comes to being happy?

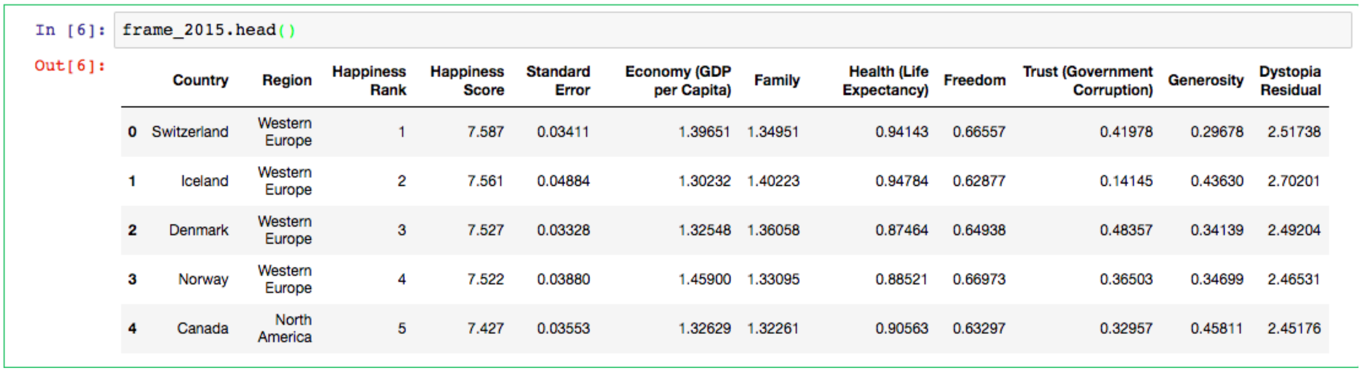
## Overview and summary

For my final project I decided to do an analysis of the World Happiness Report for the years 2015-2017, found on Kaggle (https://www.kaggle.com/unsdsn/world-happiness). The World Happiness Report is comprised of survey data from the Gallup World Poll which contains the contribution that six factors have on a country's Happiness Score. These six factors are Economy (GDP per Capita), Family, Health (Life Expectancy), Freedom, Trust (Government Corruption), Generosity and a Dystopia Residual. This report has gained a lot of recognition since it was first published in 2012, gaining recognition from governments, universities and other organizations worldwide. Given its increase in exposure over the last few years since its inception and the fact that we have data over a period of time, I was immediately interested in examining this dataset and seeing what kinds of conclusions I could draw from it.

Before going into the clean-up and exploration of this dataset, it is important to explain some of the content of this dataset. The different categorical scores are based on answers to a serious of questions asked in a poll, called a Cantril ladder. This exercise is where someone who is surveyed is asked to rate their own lives on a scale of 0 to 10, for these different categories. The answers to these questions are then aggregated and converted into weights which represent the extent to which each factor contributes to the overall happiness of a country. It is also worth explaining the Dystopia Residual, which is the residual compared to an imaginary country "Dystopia" that contains the world's most unhappy people.

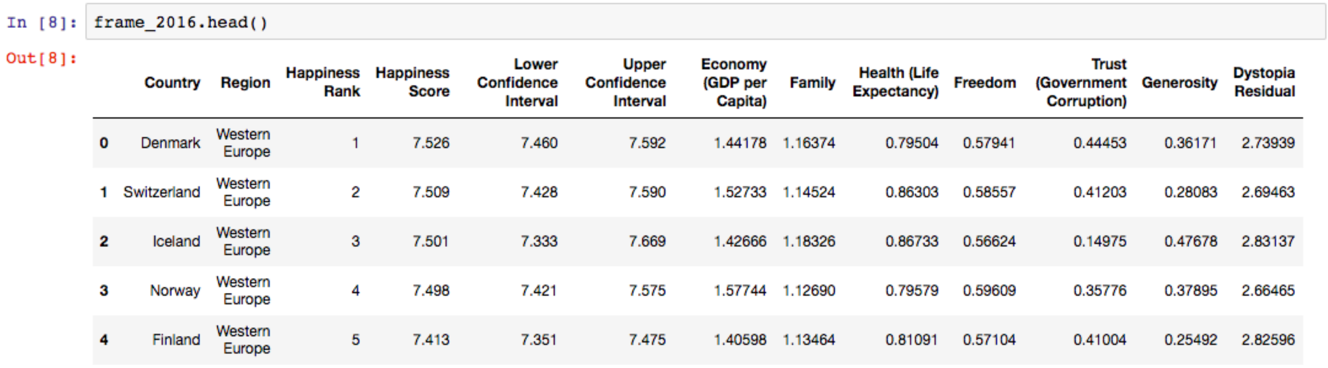
## Load and explore the data set

The first step was the loading up our dataset from Kaggle and looking for any differences between the different year's datasets.



```
In [6]: frame_2015.head()
Out[6]:
```

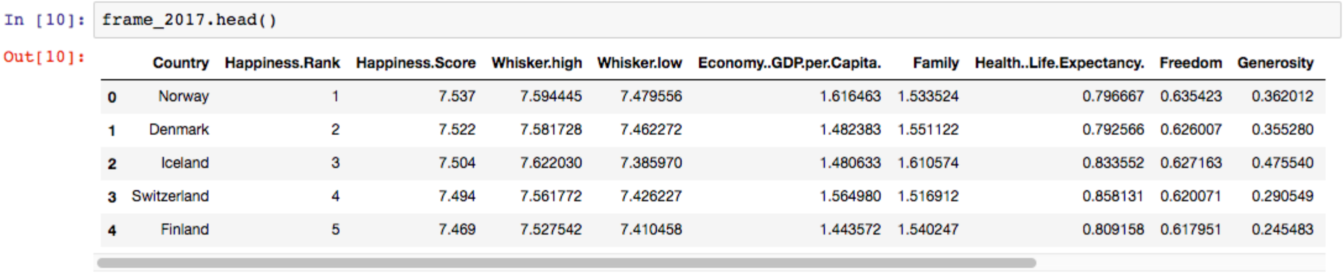
| | Country | Region | Happiness Rank | Happiness Score | Standard Error | Economy (GDP per Capita) | Family | Health (Life Expectancy) | Freedom | Trust (Government Corruption) | Generosity | Dystopia Residual |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Switzerland | Western Europe | 1 | 7.587 | 0.03411 | 1.39651 | 1.34951 | 0.94143 | 0.66557 | 0.41978 | 0.29678 | 2.51738 |
| 1 | Iceland | Western Europe | 2 | 7.561 | 0.04884 | 1.30232 | 1.40223 | 0.94784 | 0.62877 | 0.14145 | 0.43630 | 2.70201 |
| 2 | Denmark | Western Europe | 3 | 7.527 | 0.03328 | 1.32548 | 1.36058 | 0.87464 | 0.64938 | 0.48357 | 0.34139 | 2.49204 |
| 3 | Norway | Western Europe | 4 | 7.522 | 0.03880 | 1.45900 | 1.33095 | 0.88521 | 0.66973 | 0.36503 | 0.34699 | 2.46531 |
| 4 | Canada | North America | 5 | 7.427 | 0.03553 | 1.32629 | 1.32261 | 0.90563 | 0.63297 | 0.32957 | 0.45811 | 2.45176 |



```
In [8]: frame_2016.head()
```

Out[8]:

| | Country | Region | Happiness Rank | Happiness Score | Lower Confidence Interval | Upper Confidence Interval | Economy (GDP per Capita) | Family | Health (Life Expectancy) | Freedom | Trust (Government Corruption) | Generosity | Dystopia Residual |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Denmark | Western Europe | 1 | 7.526 | 7.460 | 7.592 | 1.44178 | 1.16374 | 0.79504 | 0.57941 | 0.44453 | 0.36171 | 2.73939 |
| 1 | Switzerland | Western Europe | 2 | 7.509 | 7.428 | 7.590 | 1.52733 | 1.14524 | 0.86303 | 0.58557 | 0.41203 | 0.28083 | 2.69463 |
| 2 | Iceland | Western Europe | 3 | 7.501 | 7.333 | 7.669 | 1.42666 | 1.18326 | 0.86733 | 0.56624 | 0.14975 | 0.47678 | 2.83137 |
| 3 | Norway | Western Europe | 4 | 7.498 | 7.421 | 7.575 | 1.57744 | 1.12690 | 0.79579 | 0.59609 | 0.35776 | 0.37895 | 2.66465 |
| 4 | Finland | Western Europe | 5 | 7.413 | 7.351 | 7.475 | 1.40598 | 1.13464 | 0.81091 | 0.57104 | 0.41004 | 0.25492 | 2.82596 |

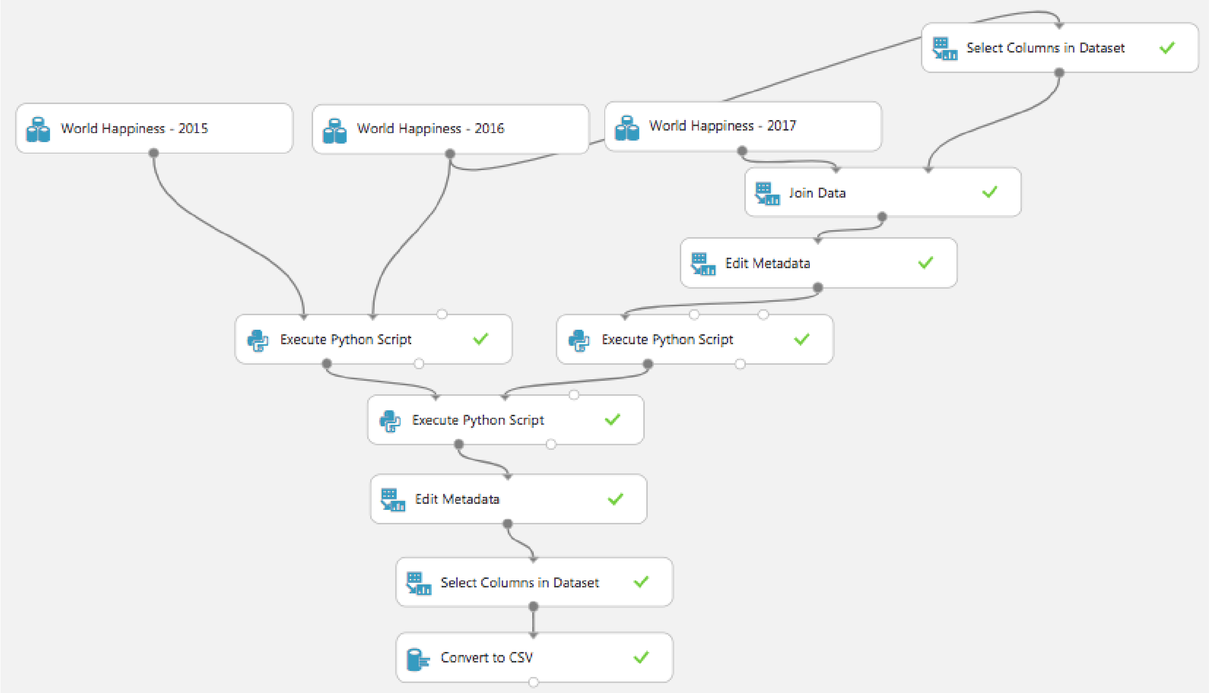


```
In [10]: frame_2017.head()
```

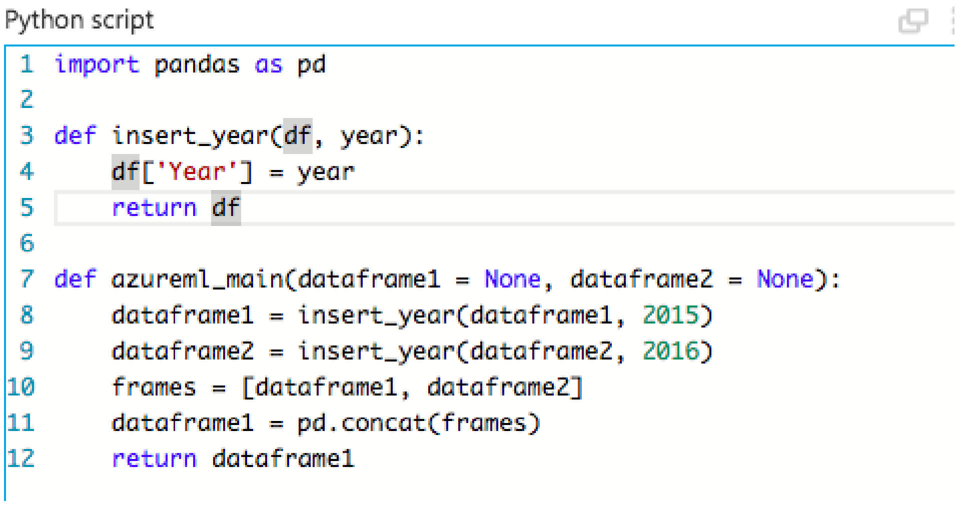
Out[10]:

| | Country | Happiness.Rank | Happiness.Score | Whisker.high | Whisker.low | Economy..GDP.per.Capita. | Family | Health..Life.Expectancy. | Freedom | Generosity |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Norway | 1 | 7.537 | 7.594445 | 7.479556 | 1.616463 | 1.533524 | 0.796667 | 0.635423 | 0.362012 |
| 1 | Denmark | 2 | 7.522 | 7.581728 | 7.462272 | 1.482383 | 1.551122 | 0.792566 | 0.626007 | 0.355280 |
| 2 | Iceland | 3 | 7.504 | 7.622030 | 7.385970 | 1.480633 | 1.610574 | 0.833552 | 0.627163 | 0.475540 |
| 3 | Switzerland | 4 | 7.494 | 7.561772 | 7.426227 | 1.564980 | 1.516912 | 0.858131 | 0.620071 | 0.290549 |
| 4 | Finland | 5 | 7.469 | 7.527542 | 7.410458 | 1.443572 | 1.540247 | 0.809158 | 0.617951 | 0.245483 |

The first thing we see here is that the 2015 dataset has a column "Standard Error", the 2016 dataset has two columns "Lower Confidence Interval" and "Upper Confidence Interval" and the 2017 dataset has "Whiskey.high" and "Whisker.low" columns, is missing the "Region" column and follows a different naming convention for some of the columns compared to the previous two years. Due to these irregularities, I decided to do some cleanup in Azure ML in order to get all three years into one dataset for further analysis.

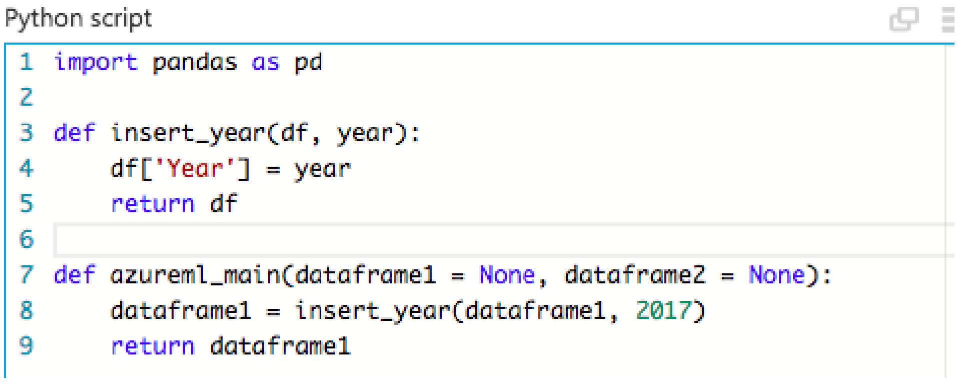


Within this Azure ML flow I used a Select Columns in Dataset module to get the Country and Region columns from the 2016 dataset and did an inner join of those columns on the 2017 dataset. I then used an Edit Metadata module to rename the 2017 columns to the same names as the other two years. I then used the following Execute Python Scripts modules to add a "Year" column to the datasets and combine the three datasets together.



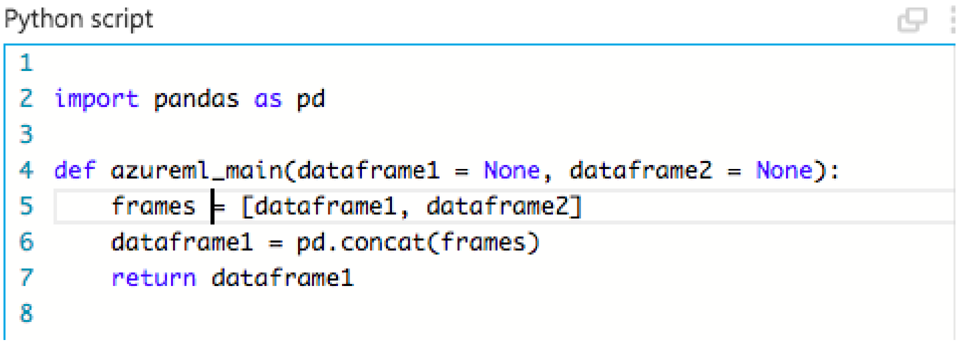
Python script

```python
import pandas as pd

def insert_year(df, year):
    df['Year'] = year
    return df

def azureml_main(dataframe1 = None, dataframe2 = None):
    dataframe1 = insert_year(dataframe1, 2015)
    dataframe2 = insert_year(dataframe2, 2016)
    frames = [dataframe1, dataframe2]
    dataframe1 = pd.concat(frames)
    return dataframe1
```



Python script

```python
import pandas as pd

def insert_year(df, year):
    df['Year'] = year
    return df

def azureml_main(dataframe1 = None, dataframe2 = None):
    dataframe1 = insert_year(dataframe1, 2017)
    return dataframe1
```

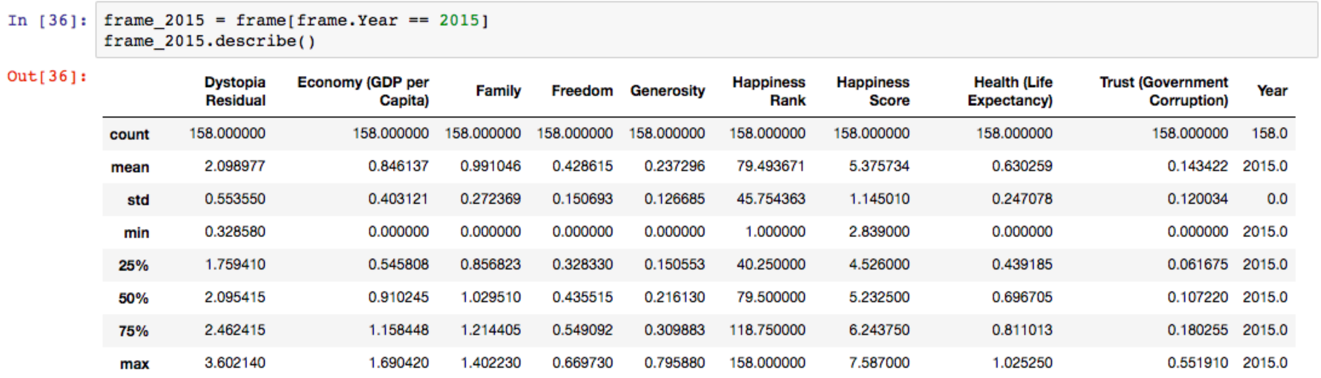


Python script

```python

import pandas as pd

def azureml_main(dataframe1 = None, dataframe2 = None):
    frames = [dataframe1, dataframe2]
    dataframe1 = pd.concat(frames)
    return dataframe1

```

After this I used an Edit Metadata module to convert the "Country", "Region" and "Year" columns to categorical and used a Select Columns in Dataset to remove the "Standard Error", "Lower Confidence Interval" and "Upper Confidence Interval" columns as they were not shared by all 3 datasets. I then used a Convert to CSV module and opened up the resulting dataset in a Jupyter Notebook for further analysis.

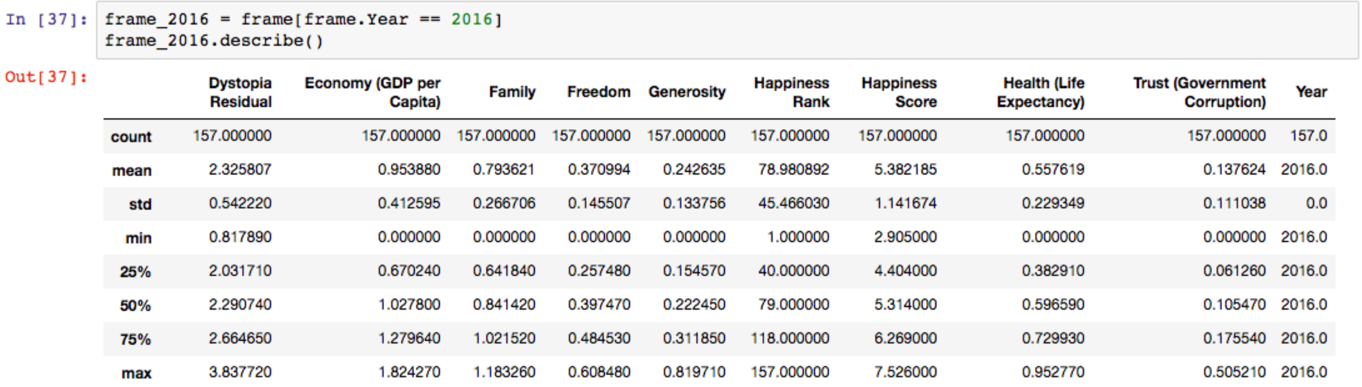
First thing I did in the notebook was split our data frame by year in order to analyze each year independently and ran some basic statistics



```
In [36]: frame_2015 = frame[frame.Year == 2015]
         frame_2015.describe()
```

Out[36]:

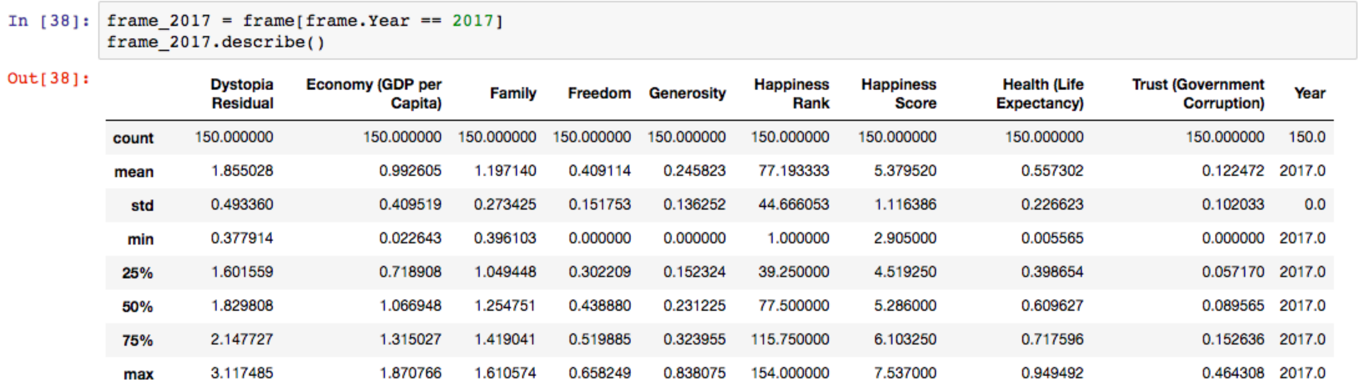
| | Dystopia Residual | Economy (GDP per Capita) | Family | Freedom | Generosity | Happiness Rank | Happiness Score | Health (Life Expectancy) | Trust (Government Corruption) | Year |
|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 158.000000 | 158.000000 | 158.000000 | 158.000000 | 158.000000 | 158.000000 | 158.000000 | 158.000000 | 158.000000 | 158.0 |
| **mean** | 2.098977 | 0.846137 | 0.991046 | 0.428615 | 0.237296 | 79.493671 | 5.375734 | 0.630259 | 0.143422 | 2015.0 |
| **std** | 0.553550 | 0.403121 | 0.272369 | 0.150693 | 0.126685 | 45.754363 | 1.145010 | 0.247078 | 0.120034 | 0.0 |
| **min** | 0.328580 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 | 2.839000 | 0.000000 | 0.000000 | 2015.0 |
| **25%** | 1.759410 | 0.545808 | 0.856823 | 0.328330 | 0.150553 | 40.250000 | 4.526000 | 0.439185 | 0.061675 | 2015.0 |
| **50%** | 2.095415 | 0.910245 | 1.029510 | 0.435515 | 0.216130 | 79.500000 | 5.232500 | 0.696705 | 0.107220 | 2015.0 |
| **75%** | 2.462415 | 1.158448 | 1.214405 | 0.549092 | 0.309883 | 118.750000 | 6.243750 | 0.811013 | 0.180255 | 2015.0 |
| **max** | 3.602140 | 1.690420 | 1.402230 | 0.669730 | 0.795880 | 158.000000 | 7.587000 | 1.025250 | 0.551910 | 2015.0 |



```
In [37]: frame_2016 = frame[frame.Year == 2016]
         frame_2016.describe()
```

Out[37]:

| | Dystopia Residual | Economy (GDP per Capita) | Family | Freedom | Generosity | Happiness Rank | Happiness Score | Health (Life Expectancy) | Trust (Government Corruption) | Year |
|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 157.000000 | 157.000000 | 157.000000 | 157.000000 | 157.000000 | 157.000000 | 157.000000 | 157.000000 | 157.000000 | 157.0 |
| **mean** | 2.325807 | 0.953880 | 0.793621 | 0.370994 | 0.242635 | 78.980892 | 5.382185 | 0.557619 | 0.137624 | 2016.0 |
| **std** | 0.542220 | 0.412595 | 0.266706 | 0.145507 | 0.133756 | 45.466030 | 1.141674 | 0.229349 | 0.111038 | 0.0 |
| **min** | 0.817890 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 | 2.905000 | 0.000000 | 0.000000 | 2016.0 |
| **25%** | 2.031710 | 0.670240 | 0.641840 | 0.257480 | 0.154570 | 40.000000 | 4.404000 | 0.382910 | 0.061260 | 2016.0 |
| **50%** | 2.290740 | 1.027800 | 0.841420 | 0.397470 | 0.222450 | 79.000000 | 5.314000 | 0.596590 | 0.105470 | 2016.0 |
| **75%** | 2.664650 | 1.279640 | 1.021520 | 0.484530 | 0.311850 | 118.000000 | 6.269000 | 0.729930 | 0.175540 | 2016.0 |
| **max** | 3.837720 | 1.824270 | 1.183260 | 0.608480 | 0.819710 | 157.000000 | 7.526000 | 0.952770 | 0.505210 | 2016.0 |



```
In [38]: frame_2017 = frame[frame.Year == 2017]
         frame_2017.describe()
```

Out[38]:

| | Dystopia Residual | Economy (GDP per Capita) | Family | Freedom | Generosity | Happiness Rank | Happiness Score | Health (Life Expectancy) | Trust (Government Corruption) | Year |
|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 150.000000 | 150.000000 | 150.000000 | 150.000000 | 150.000000 | 150.000000 | 150.000000 | 150.000000 | 150.000000 | 150.0 |
| **mean** | 1.855028 | 0.992605 | 1.197140 | 0.409114 | 0.245823 | 77.193333 | 5.379520 | 0.557302 | 0.122472 | 2017.0 |
| **std** | 0.493360 | 0.409519 | 0.273425 | 0.151753 | 0.136252 | 44.666053 | 1.116386 | 0.226623 | 0.102033 | 0.0 |
| **min** | 0.377914 | 0.022643 | 0.396103 | 0.000000 | 0.000000 | 1.000000 | 2.905000 | 0.005565 | 0.000000 | 2017.0 |
| **25%** | 1.601559 | 0.718908 | 1.049448 | 0.302209 | 0.152324 | 39.250000 | 4.519250 | 0.398654 | 0.057170 | 2017.0 |
| **50%** | 1.829808 | 1.066948 | 1.254751 | 0.438880 | 0.231225 | 77.500000 | 5.286000 | 0.609627 | 0.089565 | 2017.0 |
| **75%** | 2.147727 | 1.315027 | 1.419041 | 0.519885 | 0.323955 | 115.750000 | 6.103250 | 0.717596 | 0.152636 | 2017.0 |
| **max** | 3.117485 | 1.870766 | 1.610574 | 0.658249 | 0.838075 | 154.000000 | 7.537000 | 0.949492 | 0.464308 | 2017.0 |

One of the first things I noticed at this point was the different country counts. After some analysis I found the following differences:

- In 2015 & missing from 2016: Lesotho, Mozambique, Oman, Swaziland, Djibouti, Central African Republic
- In 2016 & missing from 2015: Puerto Rico, Belize, Namibia, South Sudan
- In 2015 & missing from 2017: Oman, Taiwan, Suriname, Hong Kong, Somaliland region, Laos, Swaziland, Djbouti, Comoros
- In 2017 & missing from 2015: Taiwan Province of China, Belize, Hong Kong S.A.R., China, Somalia, Nambia, South Sudan

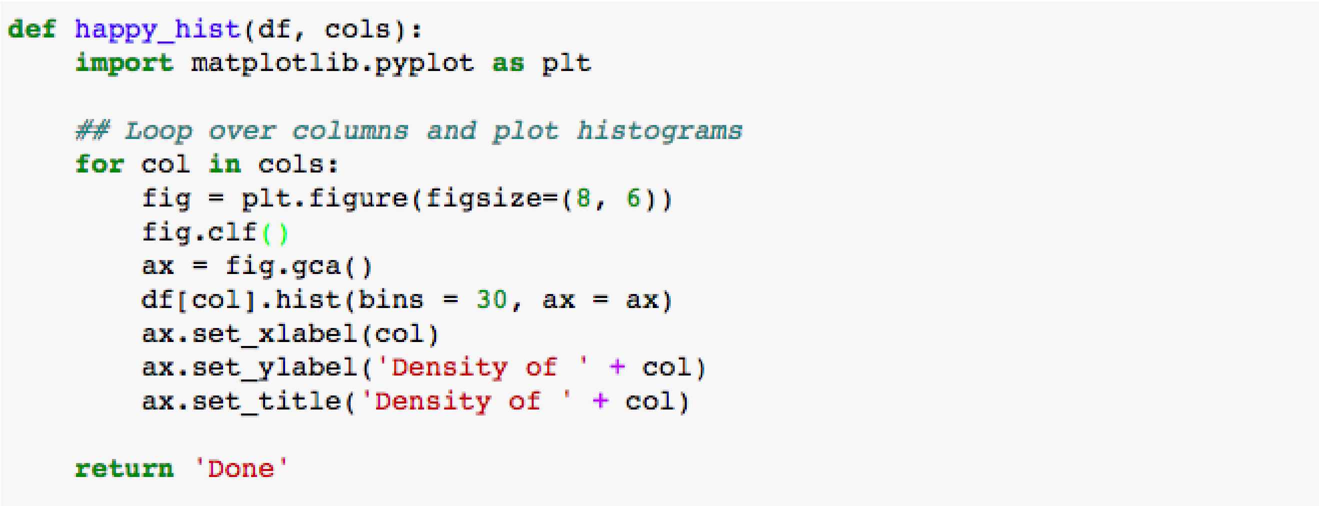
- In 2016 & missing from 2017: Puerto Rico, Taiwan, Suriname, Hong Kong, Somaliland region, Laos, Comoros
- In 2017 & missing from 2016: Taiwan Province of China, Hong Kong S.A.R., China, Mozambique, Lesotho, Central African Republic

Some interesting takeaways from this are the following:

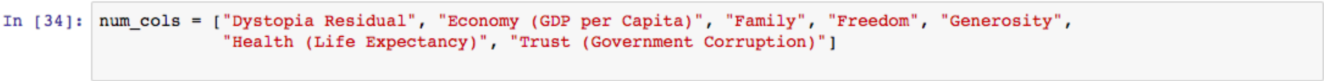
- Puerto Rico was only surveyed in 2016
- Oman was only surveyed in 2015
- Taiwan and Hong Kong were named differently in the 2017 dataset
- Mozambique, Lesotho and Central African Republic were not surveyed in 2016
- Belize was not surveyed in 2015

### What contributes most to happiness?

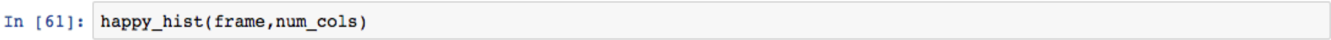
The obvious first question to ask when given a dataset on world happiness would be to find out what contributes most to happiness?  In order to answer this question, I decided to use a simple function to plot histograms for the different numerical columns in our dataset.



```python
def happy_hist(df, cols):
    import matplotlib.pyplot as plt

    ## Loop over columns and plot histograms
    for col in cols:
        fig = plt.figure(figsize=(8, 6))
        fig.clf()
        ax = fig.gca()
        df[col].hist(bins = 30, ax = ax)
        ax.set_xlabel(col)
        ax.set_ylabel('Density of ' + col)
        ax.set_title('Density of ' + col)

    return 'Done'
```

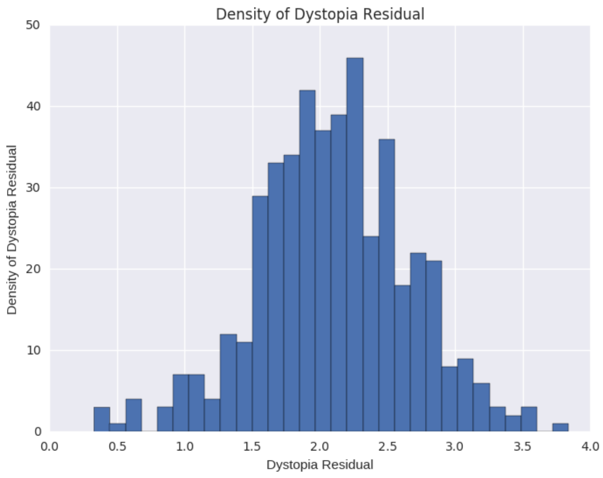


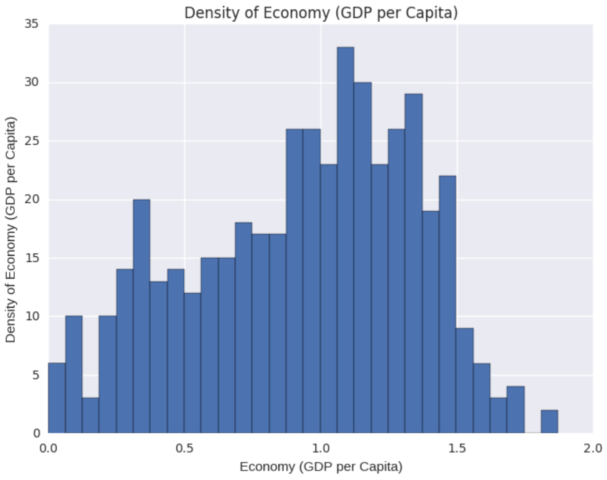
```
In [34]: num_cols = ["Dystopia Residual", "Economy (GDP per Capita)", "Family", "Freedom", "Generosity",
                     "Health (Life Expectancy)", "Trust (Government Corruption)"]
```

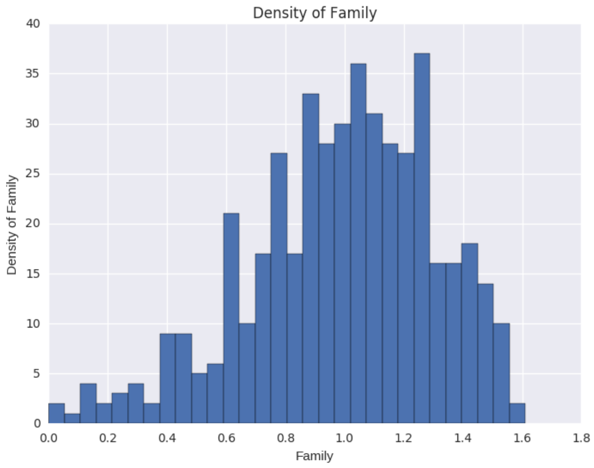


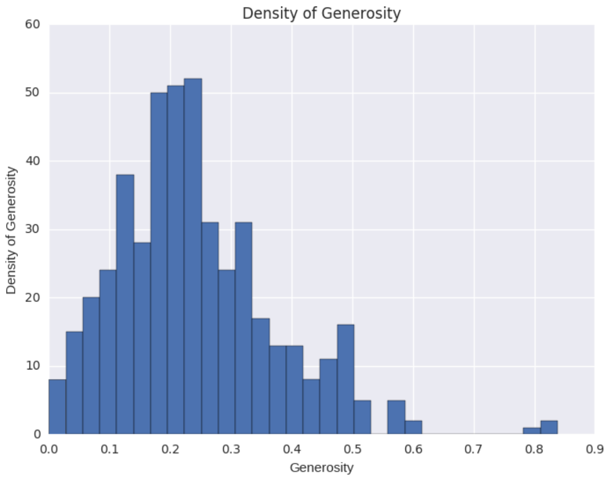
```
In [61]: happy_hist(frame,num_cols)
```

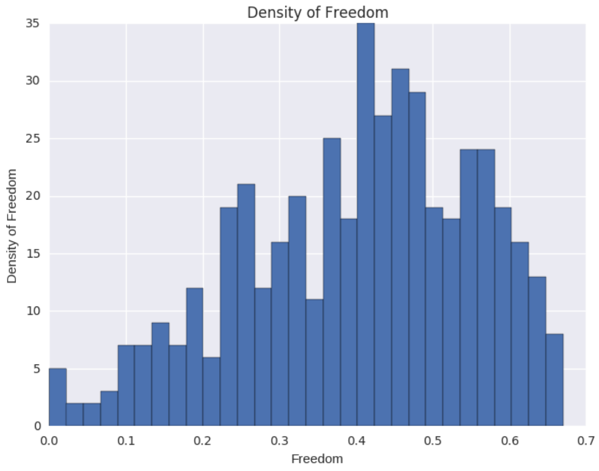
The result of this function call is the following histograms:

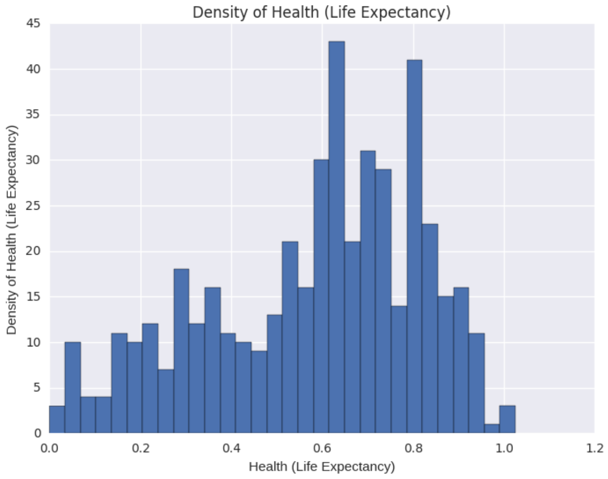

Density of Dystopia Residual
Density of Dystopia Residual
0
10
20
30
40
50
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Dystopia Residual

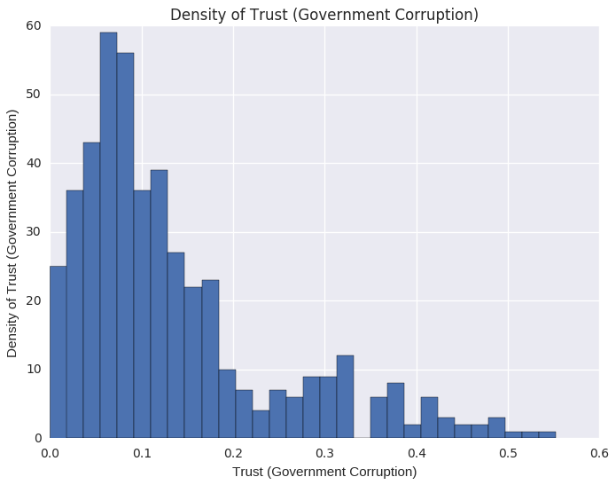

Density of Economy (GDP per Capita)
Density of Economy (GDP per Capita)
0
5
10
15
20
25
30
35
0.0
0.5
1.0
1.5
2.0
Economy (GDP per Capita)


Density of Family
Density of Family
0
5
10
15
20
25
30
35
40
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
Family


Density of Generosity
Density of Generosity
0
10
20
30
40
50
60
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Generosity


Density of Freedom
Density of Freedom
0
5
10
15
20
25
30
35
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Freedom

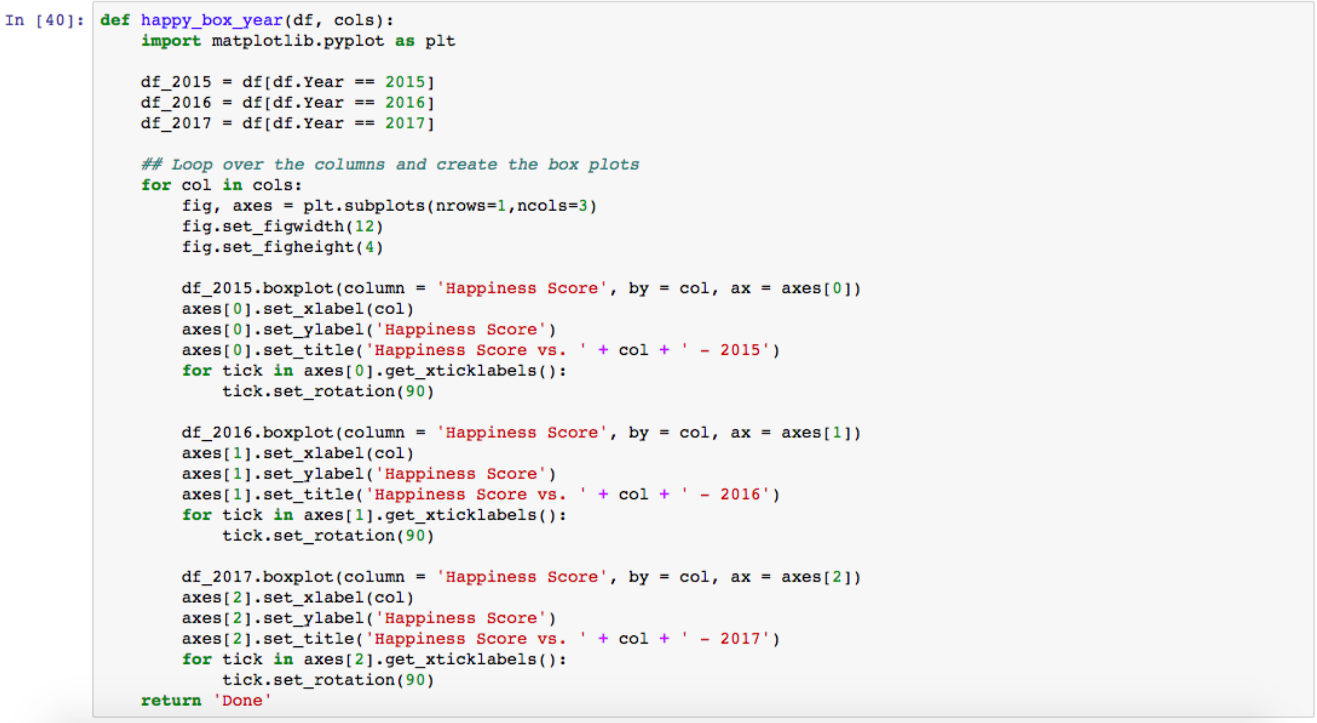

Density of Health (Life Expectancy)
Density of Health (Life Expectancy)
0
5
10
15
20
25
30
35
40
45
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Health (Life Expectancy)



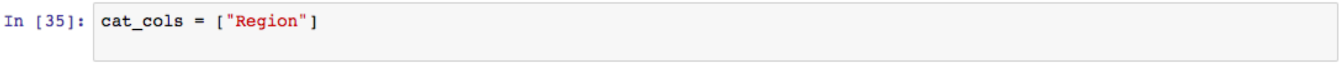
As you can see most of these factors are normally distributed, although some of the distributions are skewed. Based on the medians, we can see that Dystopia Residual, GDP and Family are the most significant factors that contribute to happiness overall.

### Trends by Region

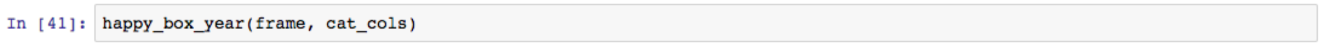
After examining the overall distribution of the different factors the next logical step was to examine the differences by region. Using the following script I was able to look at some box plots of the Happiness Score of our data grouped by Region and Year.



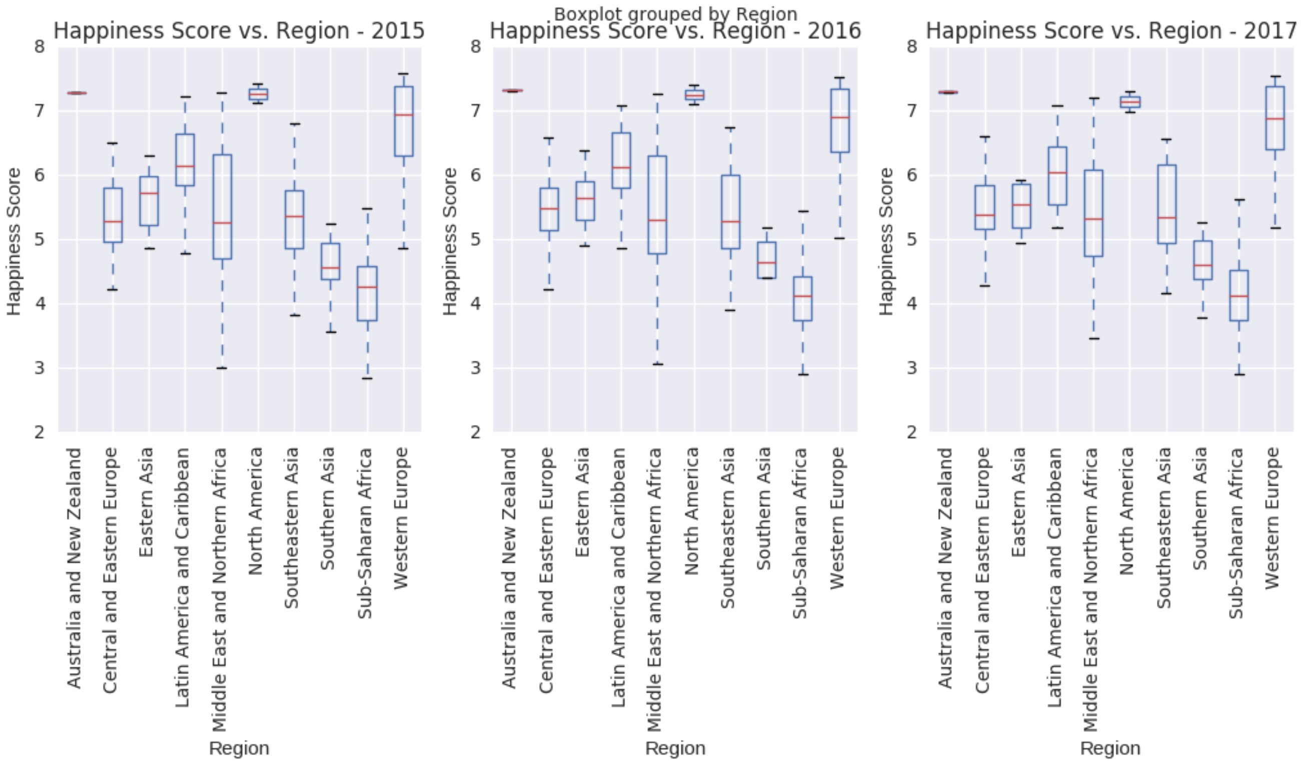
```
In [40]: def happy_box_year(df, cols):
             import matplotlib.pyplot as plt

             df_2015 = df[df.Year == 2015]
             df_2016 = df[df.Year == 2016]
             df_2017 = df[df.Year == 2017]

             ## Loop over the columns and create the box plots
             for col in cols:
                 fig, axes = plt.subplots(nrows=1,ncols=3)
                 fig.set_figwidth(12)
                 fig.set_figheight(4)

                 df_2015.boxplot(column = 'Happiness Score', by = col, ax = axes[0])
                 axes[0].set_xlabel(col)
                 axes[0].set_ylabel('Happiness Score')
                 axes[0].set_title('Happiness Score vs. ' + col + ' - 2015')
                 for tick in axes[0].get_xticklabels():
                     tick.set_rotation(90)

                 df_2016.boxplot(column = 'Happiness Score', by = col, ax = axes[1])
                 axes[1].set_xlabel(col)
                 axes[1].set_ylabel('Happiness Score')
                 axes[1].set_title('Happiness Score vs. ' + col + ' - 2016')
                 for tick in axes[1].get_xticklabels():
                     tick.set_rotation(90)

                 df_2017.boxplot(column = 'Happiness Score', by = col, ax = axes[2])
                 axes[2].set_xlabel(col)
                 axes[2].set_ylabel('Happiness Score')
                 axes[2].set_title('Happiness Score vs. ' + col + ' - 2017')
                 for tick in axes[2].get_xticklabels():
                     tick.set_rotation(90)
             return 'Done'
```



```
In [35]: cat_cols = ["Region"]
```



```
In [41]: happy_box_year(frame, cat_cols)
```



As you can see between 2015 and 2017 the general trend of happiness by region has stayed the same. North America, Australia and New Zealand and Western Europe are the happiest regions. In the middle we have Latin America and Caribbean, Eastern Asia, Central and Eastern Europe along with the Middle East and Northern Africa and Southeastern Asia, which have approximately the same median over the three years but with variable standard deviation. Finally, we have Southern Asia and Sub-Saharan Africa as the unhappiest regions.

Some interesting things to note from this visualization are that although Australia/New Zealand, North America and Western Europe have approximately the same median happiness score, Western Europe has a much larger standard deviation, most likely due to the fact that there are simply more different countries located in Western Europe. The other very interesting observation is that it seems as though the Middle East/Northern Africa's median Happiness is approximately the same over time compared with Southeastern Asia, the Middle East/Africa's standard deviation is decreasing slightly while Southeastern Asia's standard deviation is increasing. Therefore it seems as the differences in happiness by country within the Middle East/Africa is decreasing while the differences in happiness by country within Southeastern Asia is increasing over time.

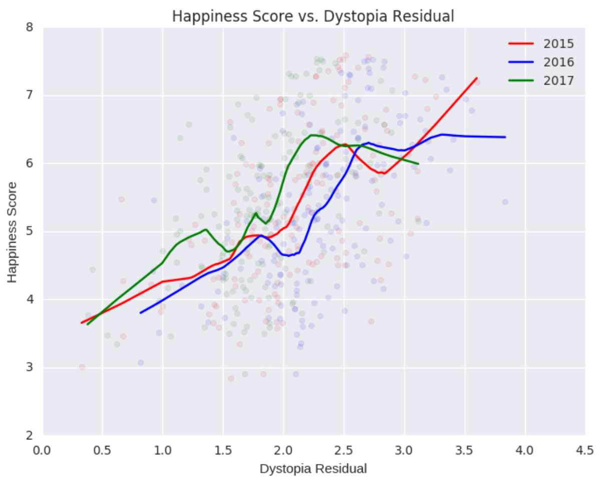
**Trends over time**

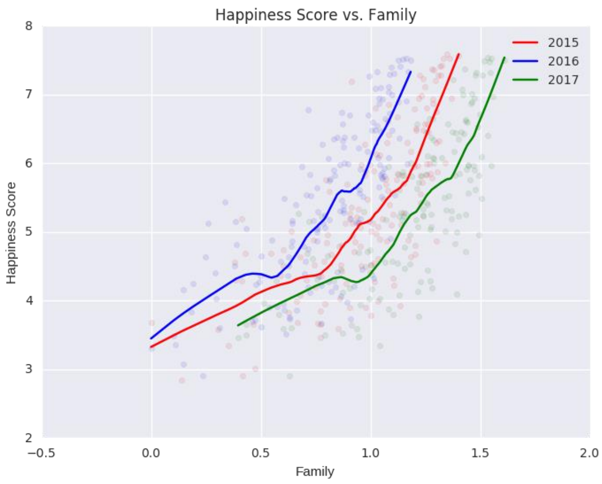
After examining some of the differences of Happiness by Region I decided to look at how the different columns interacted with happiness and to see if there were any notable trends over time. To achieve that I used a simple script to plot scatter plots and trend lines for the various columns compared to the Happiness Score.

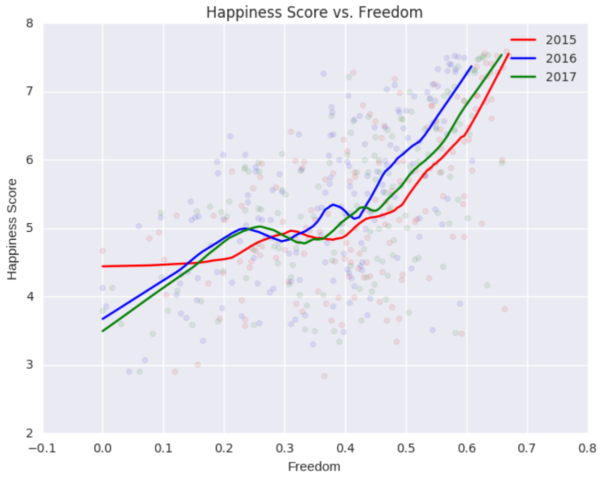
```
def happy_scatter_year(df, cols):
    import matplotlib.pyplot as plt
    import statsmodels.nonparametric.smoothers_lowess as lw

    ## Loop over the columns and create the scatter plots
    for col in cols:
        df_2015 = df[df.Year == 2015]
        df_2016 = df[df.Year == 2016]
        df_2017 = df[df.Year == 2017]
        ## first compute a lowess fit to the data
        los_2015 = lw.lowess(df_2015['Happiness Score'], df_2015[col], frac = 0.3)
        los_2016 = lw.lowess(df_2016['Happiness Score'], df_2016[col], frac = 0.3)
        los_2017 = lw.lowess(df_2017['Happiness Score'], df_2017[col], frac = 0.3)

        ## Now make the plots
        fig = plt.figure(figsize=(8, 6))
        fig.clf()
        ax = fig.gca()
        df_2015.plot(kind = 'scatter', x = col, y = 'Happiness Score', ax = ax, alpha = 0.1, color = 'red')
        plt.plot(los_2015[:, 0], los_2015[:, 1], axes = ax, color = 'red', label = '2015')

        df_2016.plot(kind = 'scatter', x = col, y = 'Happiness Score', ax = ax, alpha = 0.1, color = 'blue')
        plt.plot(los_2016[:, 0], los_2016[:, 1], axes = ax, color = 'blue', label = '2016')

        df_2017.plot(kind = 'scatter', x = col, y = 'Happiness Score', ax = ax, alpha = 0.1, color = 'green')
        plt.plot(los_2017[:, 0], los_2017[:, 1], axes = ax, color = 'green', label = '2017')
        ax.set_xlabel(col)
        ax.set_ylabel('Happiness Score')
        ax.set_title('Happiness Score vs. ' + col)
        ax.legend()
    return 'Done'
```

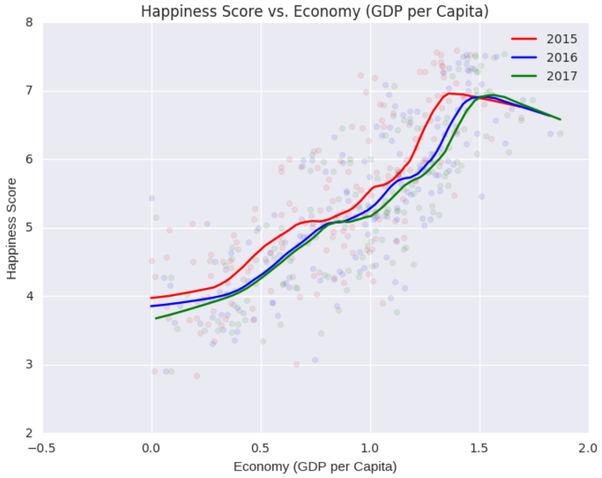
```
In [38]: %matplotlib inline
         happy_scatter_year(frame, num_cols)
```

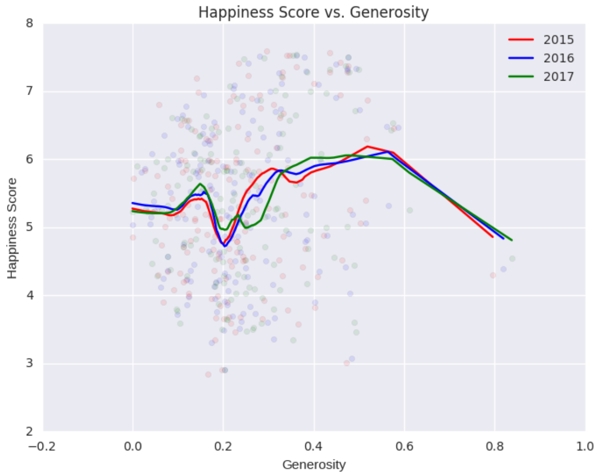
```
/home/nbuser/anaconda3_23/lib/python3.4/site-packages/matplotlib/artist.py:210: MatplotlibDeprecationWarning: This ha
s been deprecated in mpl 1.5, please use the
axes property.  A removal date has not been set.
  warnings.warn(_get_axes_msg, mplDeprecation, stacklevel=1)
```

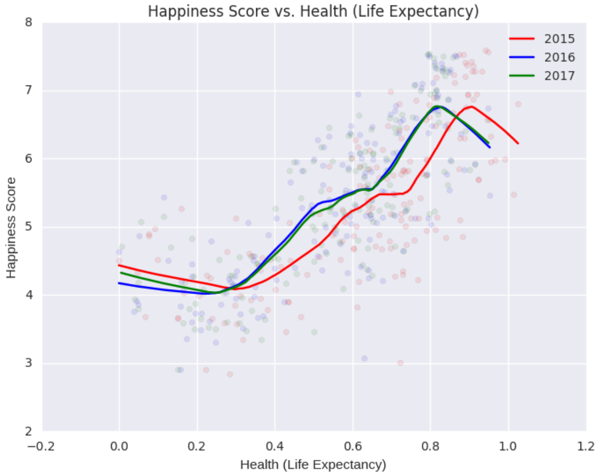


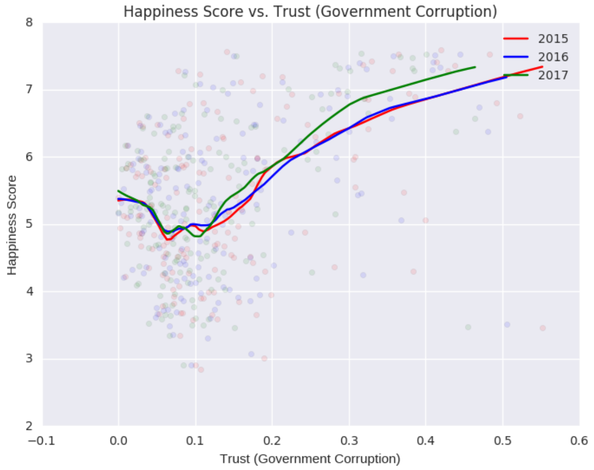












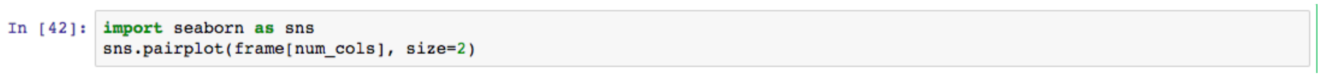
As you can see there is a clear positive correction between most attributes and the Happiness Score, except for Generosity which seems to be quite a flat relationship as well as being multimodal, hinting at some other factor interacting with Generosity. Most of the trend lines plotted for these relationships show the same general shape over the three separate years, but there are some interesting conclusions that can be drawn from these plots.

One of these such conclusions is that Dystopia played a varying part in the Happiness Score over these years. In 2016 the Dystopia Score was more influential overall compared to 2015 but less influential in 2017 compared to 2015.  Conversely Family was less influential in 2016 compared to 2015 but was more influential in 2017 compared to 201.

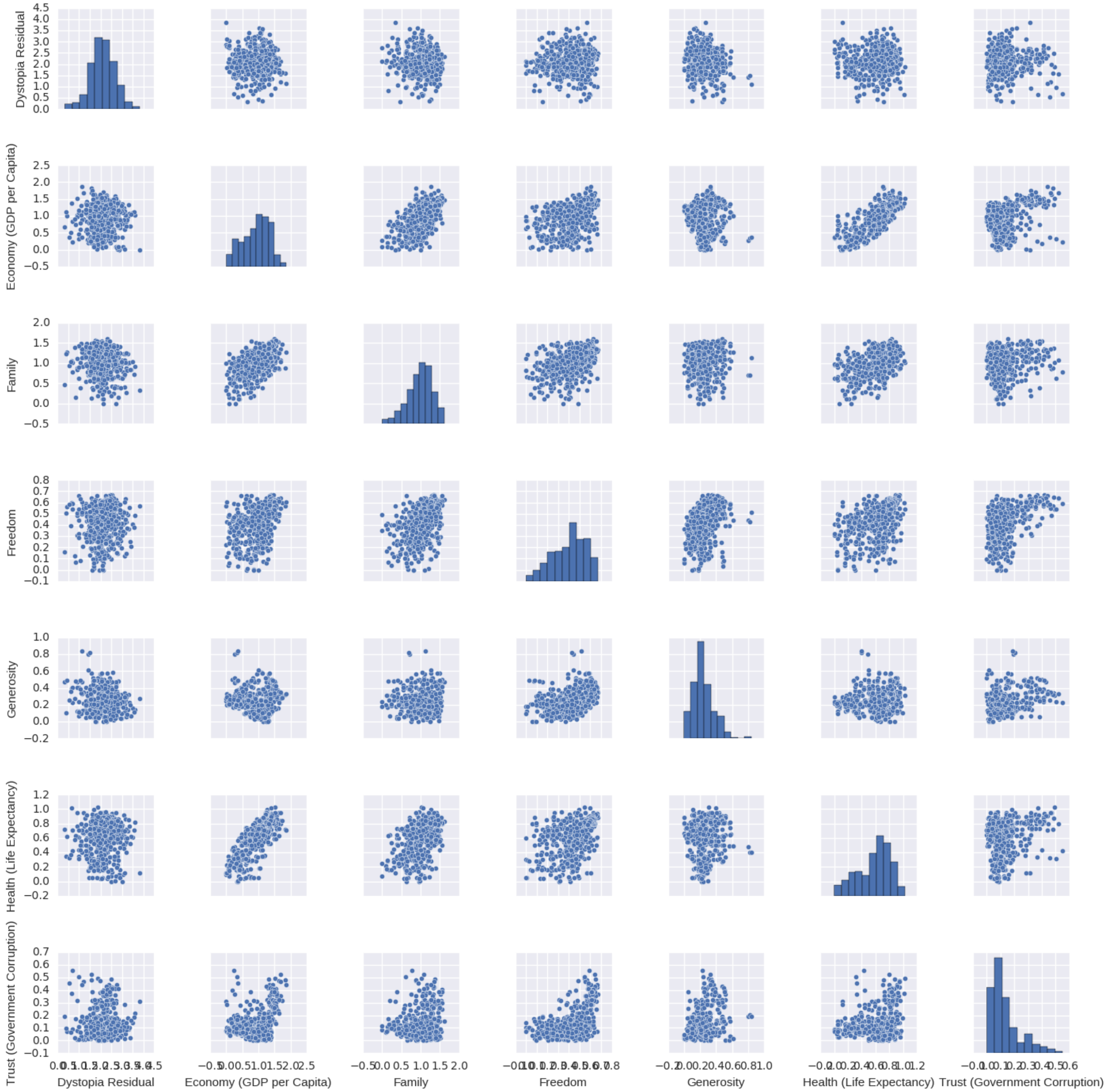
In addition to these observations we can see that GDP has steadily increased in importance each year from 2015 to 2017, following the same curve.

### Meaningful correlations

Another thing that seemed worth examining was to see if there were any correlations and relationships between the different factors towards the Happiness Score.  In order to examine these potential relationships I used the following code to create a pairwise plot.



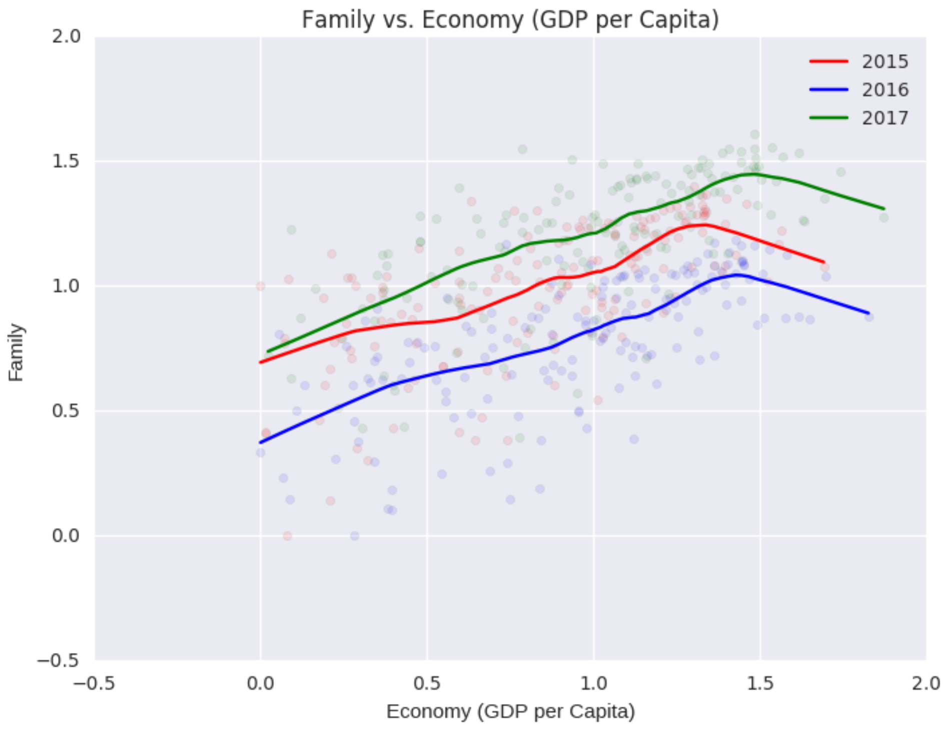
```
In [42]: import seaborn as sns
         sns.pairplot(frame[num_cols], size=2)
```

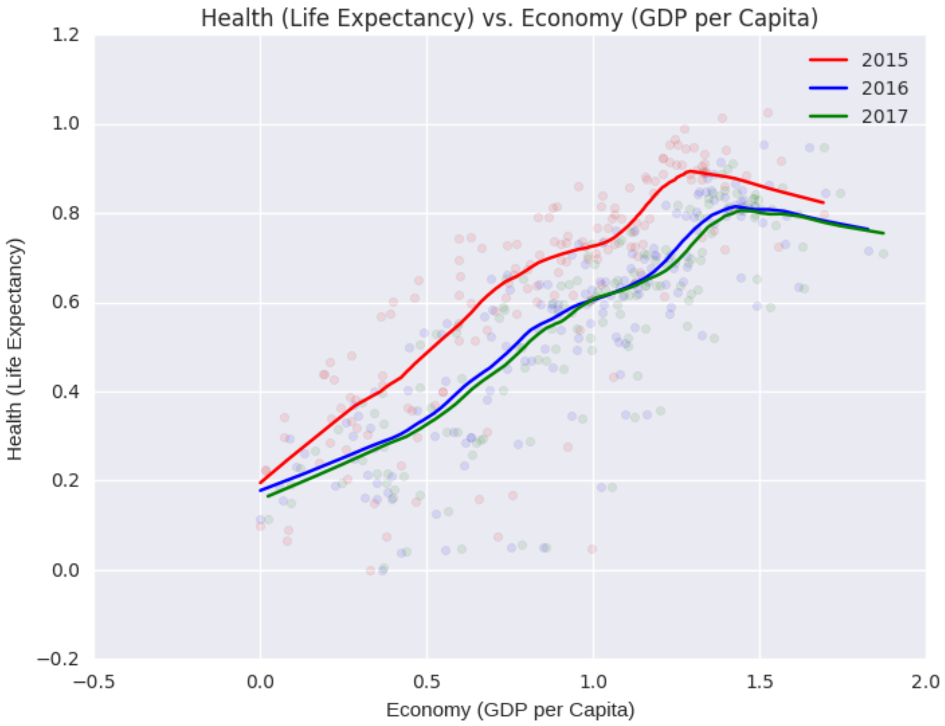


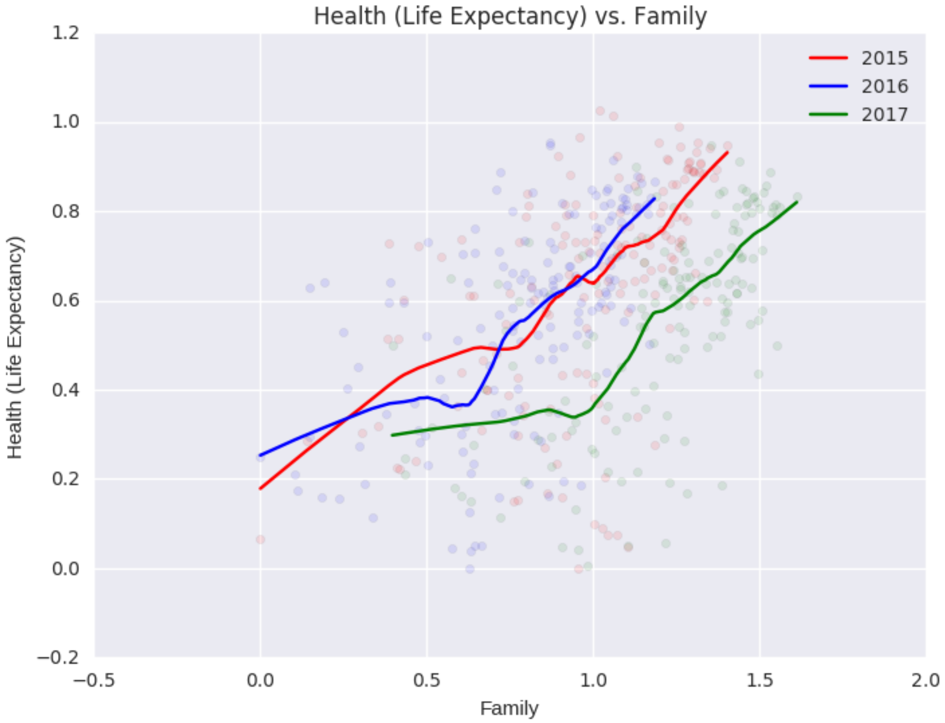
Although there does not seem to be a relationship between all of these factors, this pairwise plot was useful to identify some potential relationships that warranted further investigation.  Among these I found the most interesting relationships to be the positive correlations between Family vs GDP, Health vs GDP and Health vs Family.  I decided to investigate these correlations further to see if there were any interesting trends overtime, using the following code.

In [54]:

```
def generic_scatter_year(df, to_compare, cols):
    import matplotlib.pyplot as plt
    import statsmodels.nonparametric.smoothers_lowess as lw

    ## Loop over the columns and create the scatter plots
    for col in cols:
        df_2015 = df[df.Year == 2015]
        df_2016 = df[df.Year == 2016]
        df_2017 = df[df.Year == 2017]
        ## first compute a lowess fit to the data
        los_2015 = lw.lowess(df_2015[to_compare], df_2015[col], frac = 0.3)
        los_2016 = lw.lowess(df_2016[to_compare], df_2016[col], frac = 0.3)
        los_2017 = lw.lowess(df_2017[to_compare], df_2017[col], frac = 0.3)

        ## Now make the plots
        fig = plt.figure(figsize=(8, 6))
        fig.clf()
        ax = fig.gca()
        df_2015.plot(kind = 'scatter', x = col, y = to_compare, ax = ax, alpha = 0.1, color = 'red')
        plt.plot(los_2015[:, 0], los_2015[:, 1], axes = ax, color = 'red', label = '2015')

        df_2016.plot(kind = 'scatter', x = col, y = to_compare, ax = ax, alpha = 0.1, color = 'blue')
        plt.plot(los_2016[:, 0], los_2016[:, 1], axes = ax, color = 'blue', label = '2016')

        df_2017.plot(kind = 'scatter', x = col, y = to_compare, ax = ax, alpha = 0.1, color = 'green')
        plt.plot(los_2017[:, 0], los_2017[:, 1], axes = ax, color = 'green', label = '2017')
        ax.set_xlabel(col)
        ax.set_ylabel(to_compare)
        ax.set_title(to_compare + ' vs. ' + col)
        ax.legend()
    return 'Done'
```



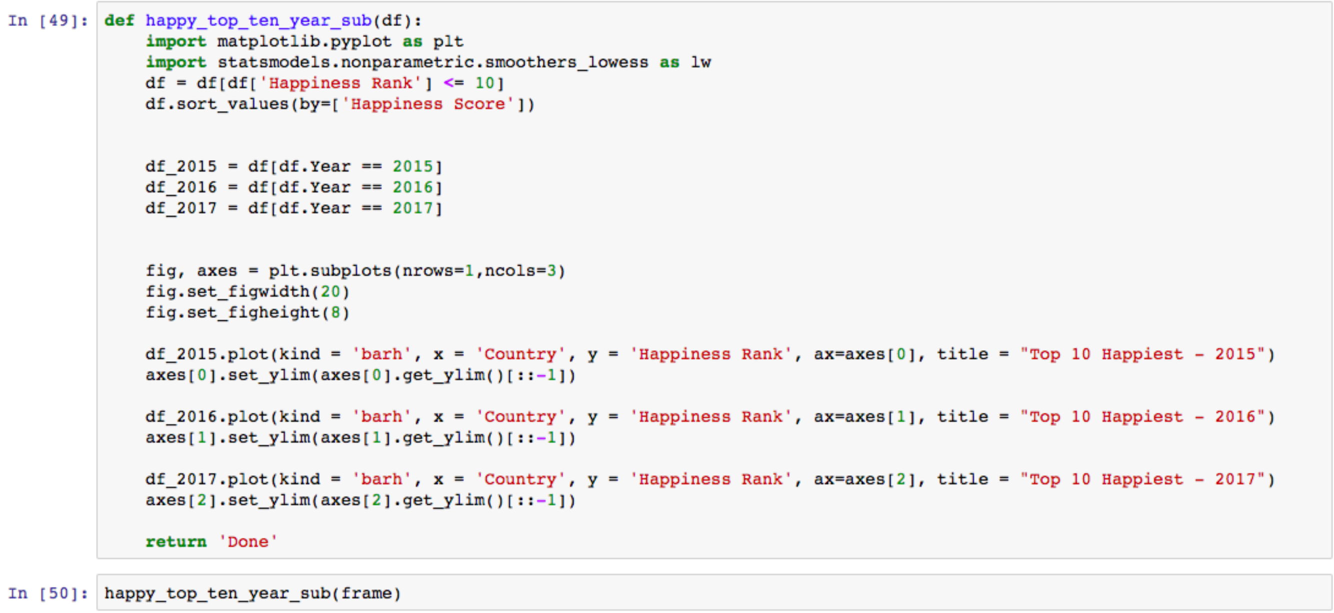




As you can see there is a definitely positive correlation between these factors and that the general curve of these relationships have remained consistent over time. The Health vs Family relationship seems to be less significant in 2017 compared to previous years. Similarly the Health vs GDP relationship also seems to be less significant in 2017 and 2016 compared to 2015. However the Family vs GDP relationship does not seem to be following a particular trend over time but does seem to fluctuate in significance.

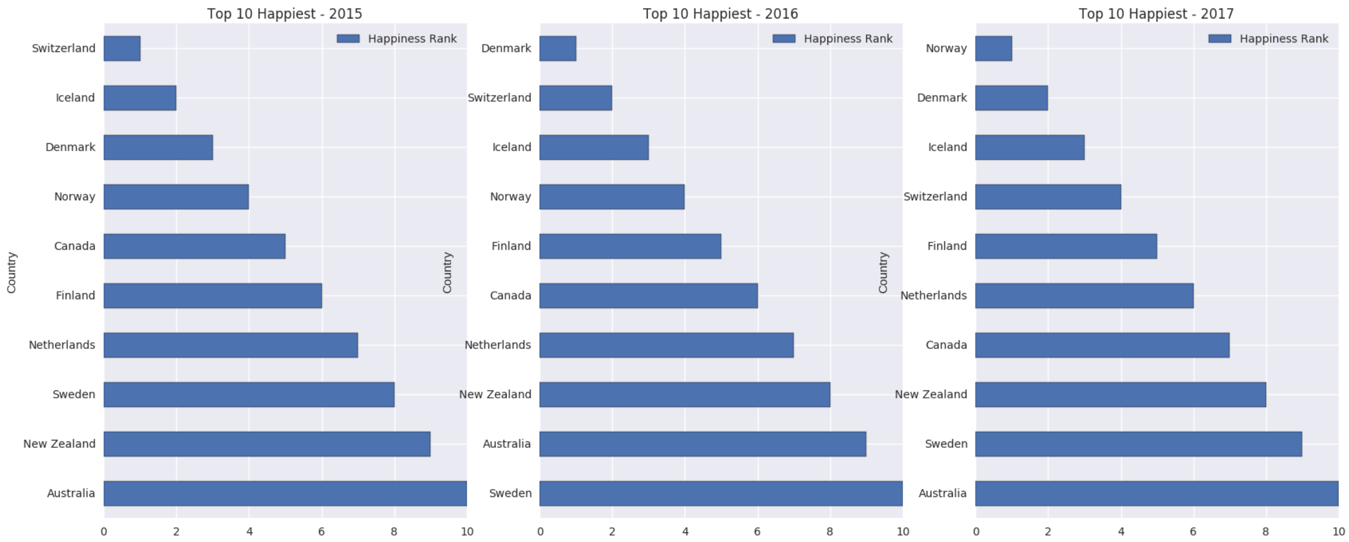
**Happiest and Unhappiest Countries**

Lastly another interesting statistic I decided to explore was to examine what the top 10 happiest and unhappiest countries were over these three years.  In order to determine who were the 10 happiest countries per year I used the code below.



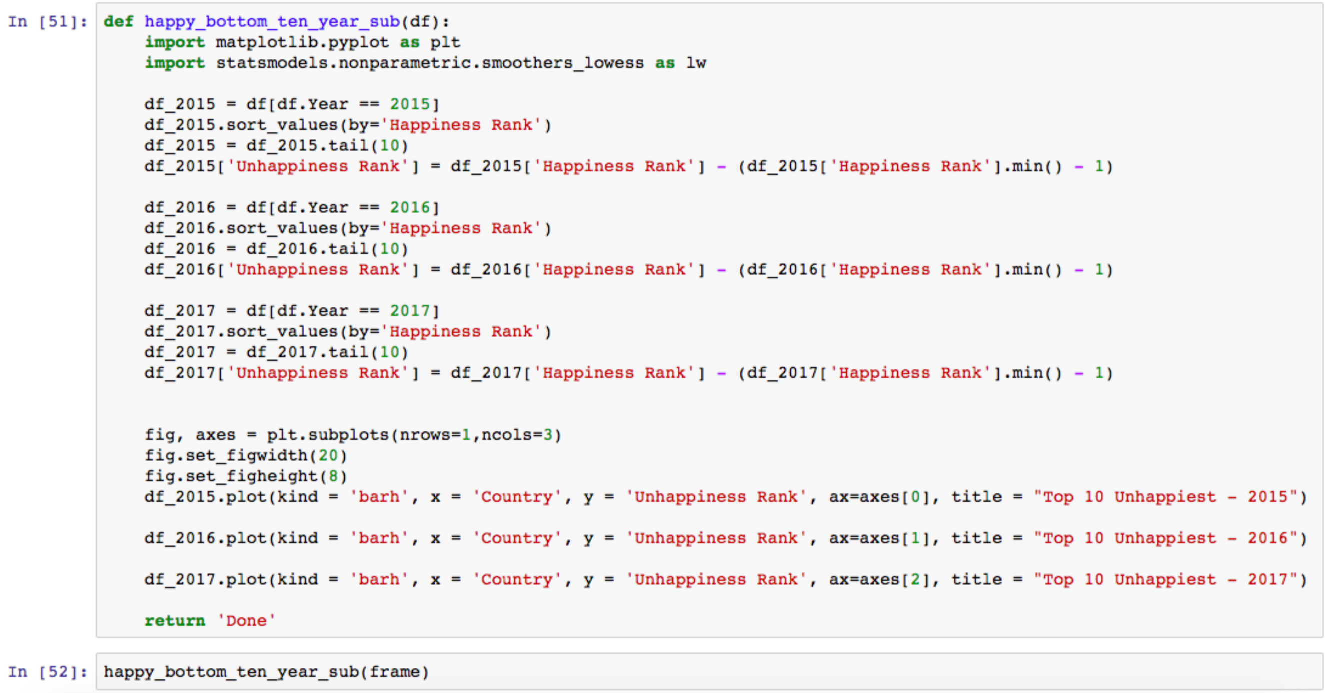
```
In [49]: def happy_top_ten_year_sub(df):
             import matplotlib.pyplot as plt
             import statsmodels.nonparametric.smoothers_lowess as lw
             df = df[df['Happiness Rank'] <= 10]
             df.sort_values(by=['Happiness Score'])


             df_2015 = df[df.Year == 2015]
             df_2016 = df[df.Year == 2016]
             df_2017 = df[df.Year == 2017]


             fig, axes = plt.subplots(nrows=1,ncols=3)
             fig.set_figwidth(20)
             fig.set_figheight(8)

             df_2015.plot(kind = 'barh', x = 'Country', y = 'Happiness Rank', ax=axes[0], title = "Top 10 Happiest - 2015")
             axes[0].set_ylim(axes[0].get_ylim()[::-1])

             df_2016.plot(kind = 'barh', x = 'Country', y = 'Happiness Rank', ax=axes[1], title = "Top 10 Happiest - 2016")
             axes[1].set_ylim(axes[1].get_ylim()[::-1])

             df_2017.plot(kind = 'barh', x = 'Country', y = 'Happiness Rank', ax=axes[2], title = "Top 10 Happiest - 2017")
             axes[2].set_ylim(axes[2].get_ylim()[::-1])

             return 'Done'
```

```
In [50]: happy_top_ten_year_sub(frame)
```

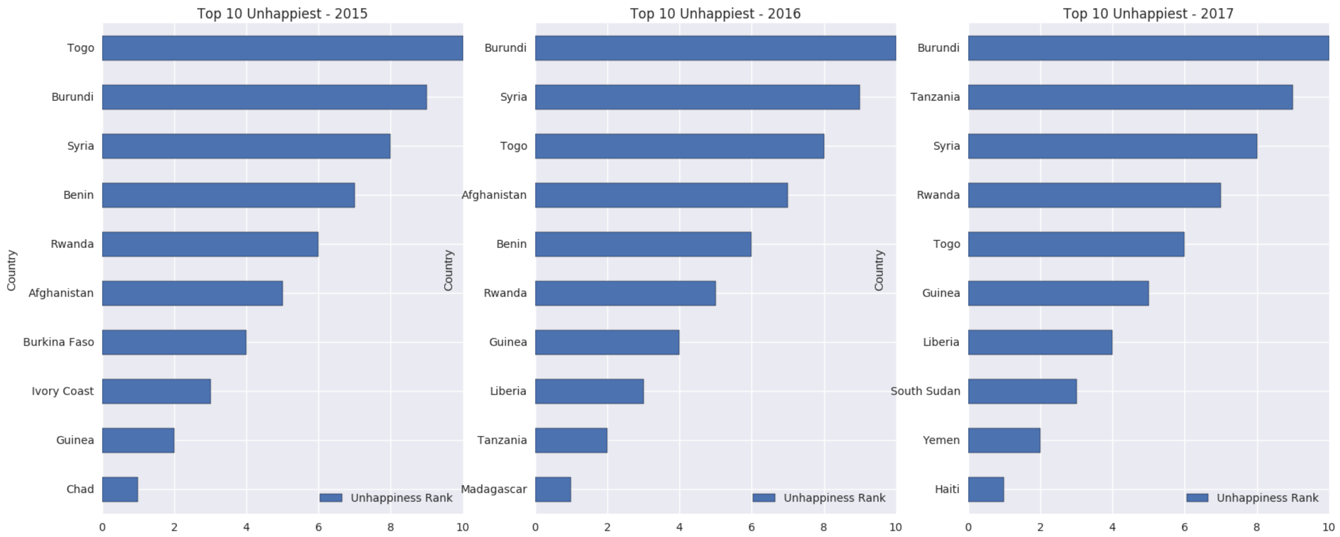


One of the most interesting this about this analysis is that we can see that there are the same 10 countries in the top 10 for each year. Some other interesting observations from these charts are that Switzerland and Canada seem to be decreasing in rank over time while most other countries hover around the same approximate ranking, only moving up or down one or two rankings.

Similarly to the top 10 happiest countries, I used the following code to calculate the top 10 unhappiest countries over these 3 years.



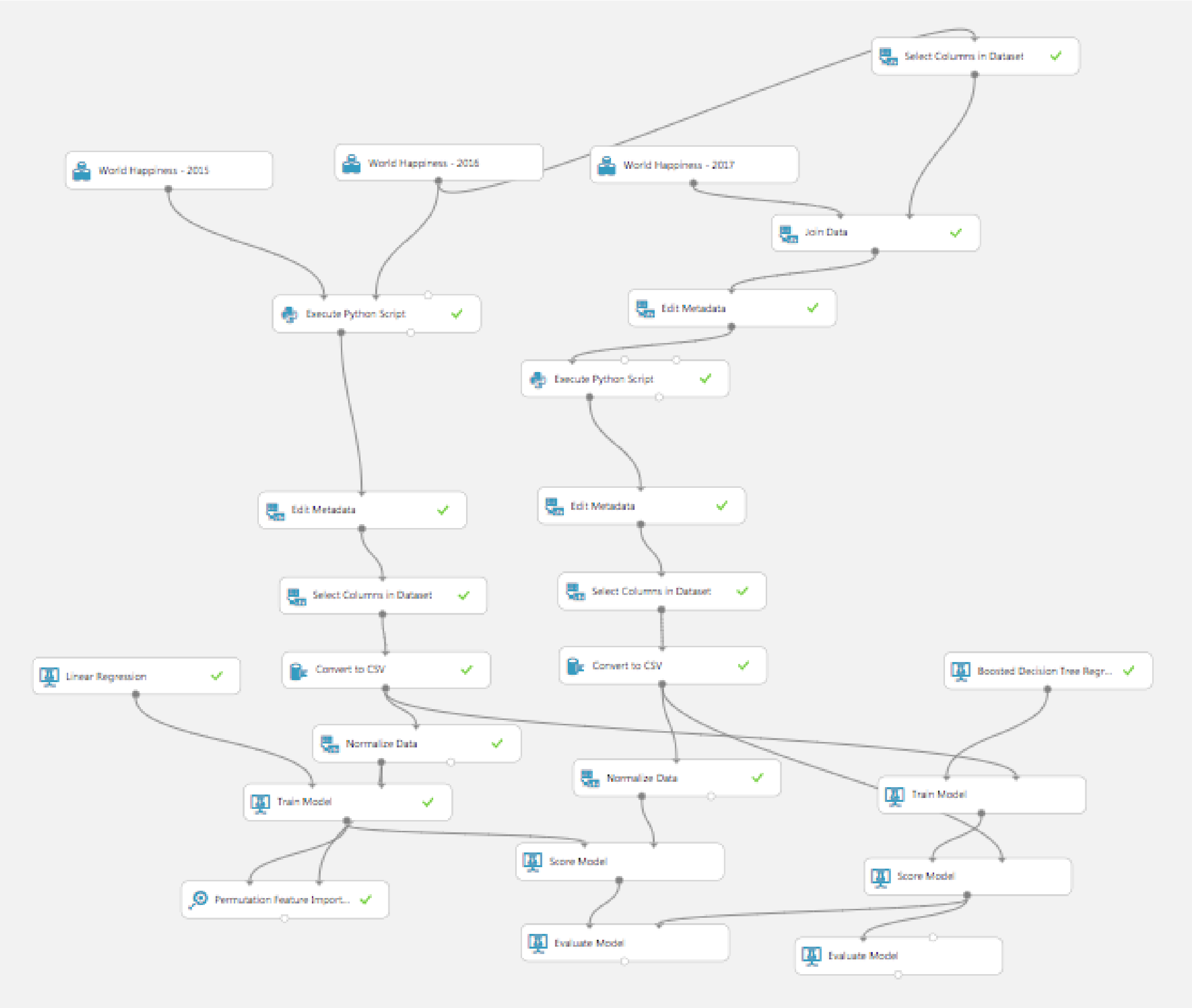
```
In [51]: def happy_bottom_ten_year_sub(df):
             import matplotlib.pyplot as plt
             import statsmodels.nonparametric.smoothers_lowess as lw

             df_2015 = df[df.Year == 2015]
             df_2015.sort_values(by='Happiness Rank')
             df_2015 = df_2015.tail(10)
             df_2015['Unhappiness Rank'] = df_2015['Happiness Rank'] - (df_2015['Happiness Rank'].min() - 1)

             df_2016 = df[df.Year == 2016]
             df_2016.sort_values(by='Happiness Rank')
             df_2016 = df_2016.tail(10)
             df_2016['Unhappiness Rank'] = df_2016['Happiness Rank'] - (df_2016['Happiness Rank'].min() - 1)

             df_2017 = df[df.Year == 2017]
             df_2017.sort_values(by='Happiness Rank')
             df_2017 = df_2017.tail(10)
             df_2017['Unhappiness Rank'] = df_2017['Happiness Rank'] - (df_2017['Happiness Rank'].min() - 1)


             fig, axes = plt.subplots(nrows=1,ncols=3)
             fig.set_figwidth(20)
             fig.set_figheight(8)
             df_2015.plot(kind = 'barh', x = 'Country', y = 'Unhappiness Rank', ax=axes[0], title = "Top 10 Unhappiest - 2015")

             df_2016.plot(kind = 'barh', x = 'Country', y = 'Unhappiness Rank', ax=axes[1], title = "Top 10 Unhappiest - 2016")

             df_2017.plot(kind = 'barh', x = 'Country', y = 'Unhappiness Rank', ax=axes[2], title = "Top 10 Unhappiest - 2017")

             return 'Done'

In [52]: happy_bottom_ten_year_sub(frame)
```



As you can see the 10 unhappiest countries are not the same 10 countries over these three years. However, there are 6 countries which are all in the 10 unhappiest countries over these years: Guinea, Rwanda, Syria, Burndi and Togo. Togo looks as if it is dropping in the unhappiness ranks over these three years while Guinea is steadily increasing in unhappiness, whereas most of the other repeating countries are staying in approximately the same rank.

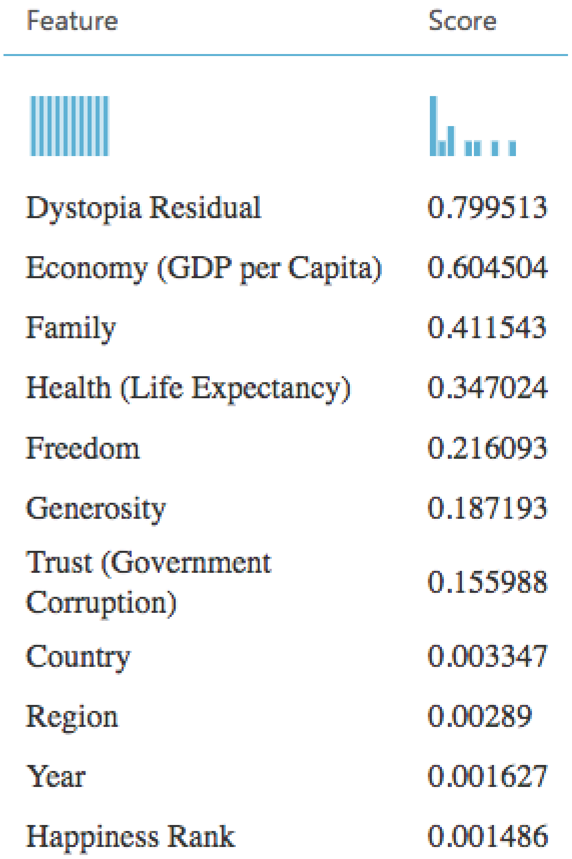
**Predicting Happiness**

During my exploration of the dataset, I thought it would also be interesting to see if I could create a regression model which could predict the happiness score of the 2017 countries after being trained with the 2015-2016 data. The model can be viewed at https://gallery.azure.ai/Experiment/World-Happiness-Report-2015-2017-Regression.



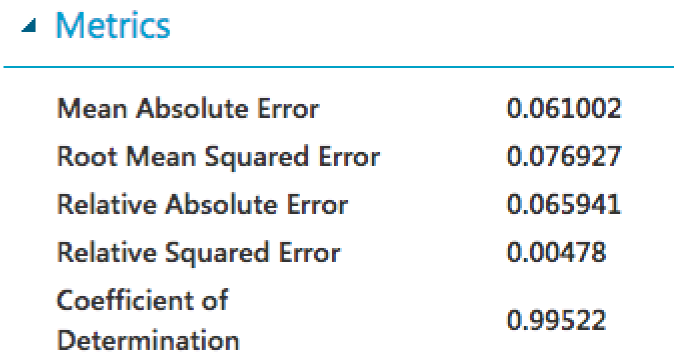
The experiment used to build this predictive model is very similar to the experiment used to prepare and combine the datasets for analysis and visualization, but with a few slight differences. Firstly, I decided to normalize the numerical columns used for predicting (Dystopia, GDO, Family, Freedom, Generosity, Health and Trust) using a ZScore normalization module, in order to minimize the effect of any outliers in the dataset. Besides this, the main difference worth noting between this regression and some of other regressions built during the semester is that rather than splitting my data, using one portion to train and one portion to score the model, I am using the 2015-2016 data to train my models and the 2017 data to score my models. I decided to do this as it is basically a 66/33 split, which is pretty close to the typical 70/30 split used during this class, as well as the fact that predicting a future happiness score based on data from previous years seems the most interesting way to examine this problem.

After updating normalizing the data, I first decided to attempt a Linear Regression model as the data is highly numerical. I configured the Linear Regression model using the Ordinary Least Squares Solution method and also attached a Permutation Feature Importance module using the Root Mean Squared Error metric, in order to determine if there were any features that were less significant than others.



| Feature | Score |
|---|---|
| Dystopia Residual | 0.799513 |
| Economy (GDP per Capita) | 0.604504 |
| Family | 0.411543 |
| Health (Life Expectancy) | 0.347024 |
| Freedom | 0.216093 |
| Generosity | 0.187193 |
| Trust (Government Corruption) | 0.155988 |
| Country | 0.003347 |
| Region | 0.00289 |
| Year | 0.001627 |
| Happiness Rank | 0.001486 |

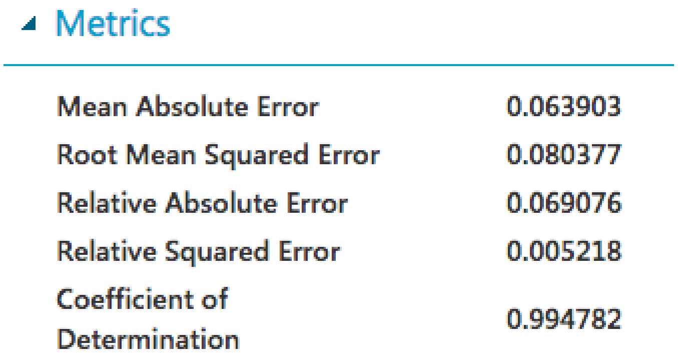
As you can see from the Permutation Feature Importance module output, the features that are most important are Dystopia Residual, Economy, Family and Health, which confirmed my previous analysis' conclusion that these were the most influential features.  However, I was surprised that Region did not play as much of a role, as the data seemed to show that there were some differences per region.  I supposed although there were differences by region, those differences were also expressed within the other features which in turn were more significant than Region itself. As you can see, Year has very little significance towards our predictive model. Here is the output from the Evaluation Model module from my Linear Regression Model.



Metrics

| | |
|---|---|
| Mean Absolute Error | 0.061002 |
| Root Mean Squared Error | 0.076927 |
| Relative Absolute Error | 0.065941 |
| Relative Squared Error | 0.00478 |
| Coefficient of Determination | 0.99522 |

With really no updates to the default parameters of the Linear Regression model, we can see that my model is highly predictive for the Happiness Score feature. With a .9952 Coefficient of Determination, we can see that the model explains almost all of the variance in the data. I was very happy with this first model, but in the interest of exploring more options I also decided to compare this Linear Regression model with a Boosted Decision Tree Regression module, which can be viewed below.



Metrics

| | |
|---|---|
| Mean Absolute Error | 0.063903 |
| Root Mean Squared Error | 0.080377 |
| Relative Absolute Error | 0.069076 |
| Relative Squared Error | 0.005218 |
| Coefficient of Determination | 0.994782 |

As you can see similarly to the Linear Regression model, this Boosted Decision Tree model also performed extremely well with no changes to the default parameters of the model. I suspect that given the fact that this dataset has been prepared before being published for mass consumption and is highly linear, our models had no trouble predicting the happiness score, although the Decision Tree is slightly less predictive, with a higher RMSE and lower Coefficient of Determination.

**Conclusion**

In conclusion there is a wide range of factors that contribute to a country's happiness, including Economy (GDP per Capita), Family, Health (Life Expectancy), Freedom, Trust (Government Corruption) and Generosity.  The Happiest regions of the world seem to be Australia/New Zealand, North America and Western Europe, while the unhappiest regions seem to be Southeast Asia and Sub-Saharan Africa. The factors that contribute most to happiness seem to be GDP, Family and the country's Dystopia Residual, which Trust (Government Corruption) and Generosity seem to have little effect.  There are some meaningful correlations between Family importance vs GDP, Health vs GDP and Health vs Family. In addition to this there were several interesting trends over these three years, including an increased importance of GDP on overall Happiness.  Lastly, we determined that the 10 happiest countries have not changed at all and the 10 unhappiest countries have changed very little over these three years, indicating that not much is changing at the extreme ends of the spectrum.

I was also able to build a few different predictive models trained with 2015-2016 data in order to predict the 2017 Happiness Score. Through analysis using the Permutation Feature Importance module I was able to confirm my previous conclusions that the most significant factors to Happiness Score were the Dystopia Residual, GDP, Family and Health. I was also able to compare a Linear Regression Model to a Boosted Decision Tree model, which both performed well on this highly linear data.

Given the extreme impact that these factors have on world happiness and the importance that happiness plays in the human experience, this is just the tip of the iceberg for the knowledge that can be extracted from this data. I would be interested in extending this analysis to the years 2012-2014 as well as in the future to identify any additional meaningful information about how these factors play into happiness.