EDAHappinessNotebook

November 16, 2019

1 The World Happiness Report - (2015 & 2016)

The following analyzation will be looking at data from the World Happiness Report years 2015 and 2016.

I hypothesize that regions with greater freedom from the government and with generous citizens will tend to have a higher happiness score than the other regions.

1.1 Modules & Packages

[465]:
```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import thinkplot
from matplotlib import pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
import random
import scipy.stats
import thinkstats2
```

1.1.1 Dataframes

[441]:
```
df15 = pd.read_csv("2015-Copy1.csv") # Data from 2015
df16 = pd.read_csv("2016-Copy1.csv") # Data from 2016
```

1.2 2015 Dataset

First five rows of 2015 datset shown below.

1.2.1 Variables:

Country: Name of country

Region : Region the country pertains to

Happiness Rank : Rank of the country based on the Happiness Score

Happiness Score : A metric measured in 2015 by asking the sampled people: "How would you rate your happiness on a scale of 0 to 10 where 10 is the happiest."

Economy (GDP per Capita): real GDP per capita

Family : social support

Health (Life Expectancy): healthy life expectancy

Freedom : freedom to make life choices

Trust (Government Corruption): perceptions of corruption

Generosity : perceptions of generosity

Dystopia : each country is compared against a hypothetical nation that represents the lowest national averages for each key variable and is, along with residual error, used as a regression benchmark

[429]:
```
df15.head(5)
```

[429]:

	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

[43]:
```
df15.info() # variables in dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 12 columns):
Country                          158 non-null object
Region                           158 non-null object
Happiness Rank                   158 non-null int64
Happiness Score                  158 non-null float64
Standard Error                   158 non-null float64
Economy (GDP per Capita)         158 non-null float64
Family                           158 non-null float64
Health (Life Expectancy)         158 non-null float64
Freedom                          158 non-null float64
Trust (Government Corruption)    158 non-null float64
Generosity                       158 non-null float64
Dystopia Residual                158 non-null float64
dtypes: float64(9), int64(1), object(2)
memory usage: 14.9+ KB
```

[49]:
```
print("\n\nRegions:\n")
print("Distinct values found: {} \n".format(len(df15.Region.unique())))
print(df15.Region.unique())
```

```
Regions:

Distinct values found: 10

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

[442]:
```
df15.describe() # count, mean, min, and max found here
```

[442]:

	Happiness Rank	Happiness Score	Standard Error	\
count	158.000000	158.000000	158.000000	
mean	79.493671	5.375734	0.047885	
std	45.754363	1.145010	0.017146	
min	1.000000	2.839000	0.018480	
25%	40.250000	4.526000	0.037268	
50%	79.500000	5.232500	0.043940	
75%	118.750000	6.243750	0.052300	
max	158.000000	7.587000	0.136930	

	Economy (GDP per Capita)	Family	Health (Life Expectancy)	\
count	158.000000	158.000000	158.000000	

```
mean                   0.846137    0.991046                   0.630259
std                    0.403121    0.272369                   0.247078
min                    0.000000    0.000000                   0.000000
25%                    0.545808    0.856823                   0.439185
50%                    0.910245    1.029510                   0.696705
75%                    1.158448    1.214405                   0.811013
max                    1.690420    1.402230                   1.025250

          Freedom  Trust (Government Corruption)  Generosity  \
count  158.000000                     158.000000  158.000000
mean     0.428615                       0.143422    0.237296
std      0.150693                       0.120034    0.126685
min      0.000000                       0.000000    0.000000
25%      0.328330                       0.061675    0.150553
50%      0.435515                       0.107220    0.216130
75%      0.549092                       0.180255    0.309883
max      0.669730                       0.551910    0.795880

       Dystopia Residual
count         158.000000
mean            2.098977
std             0.553550
min             0.328580
25%             1.759410
50%             2.095415
75%             2.462415
max             3.602140
```

1.2.2 Histograms

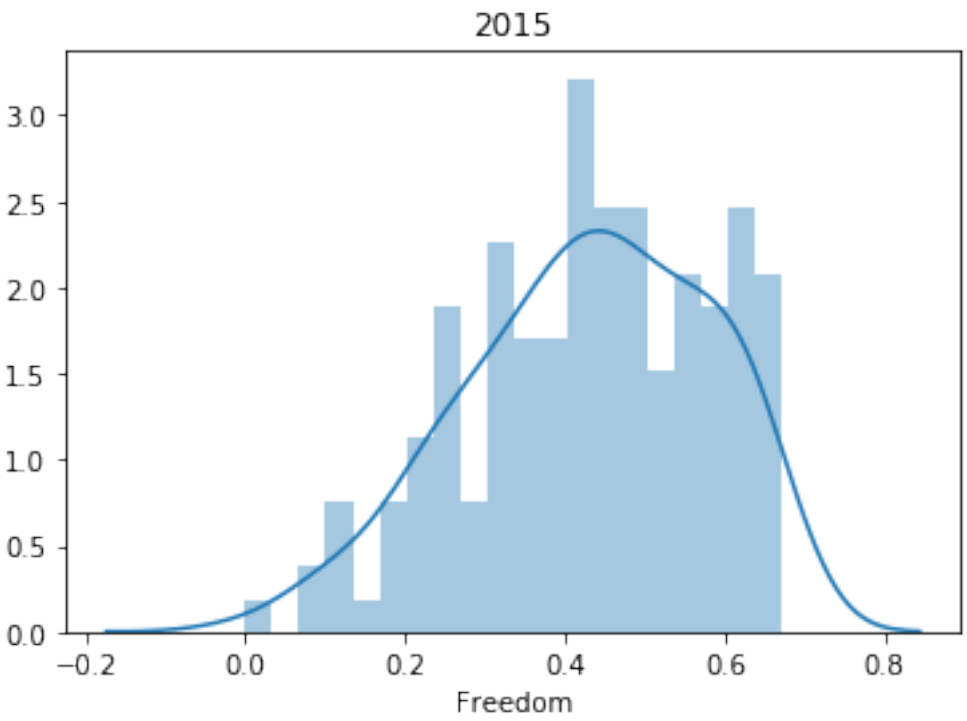
[ ]:
```
# Chapter 2
# Add Titles + Labels
# Descriptive Stats of Variables: Mean, Mode, Spread, and Tails
```

[187]:
```
sns.distplot(df15['Happiness Score'], bins = 10).set_title(2015) #density plot␣
↪of happiness score
# add titles and labels
```

[187]:
```
Text(0.5, 1.0, '2015')
```

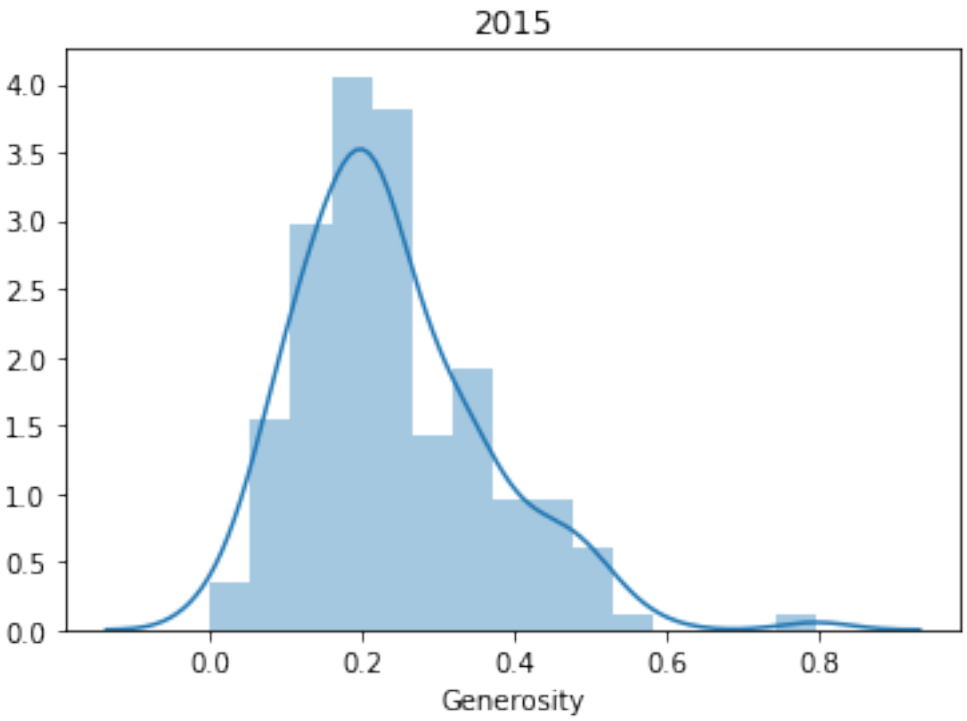
[149]:
```
sns.distplot(df15['Freedom'], bins = 20).set_title(2015)
```

[149]:
```
Text(0.5, 1.0, '2015')
```



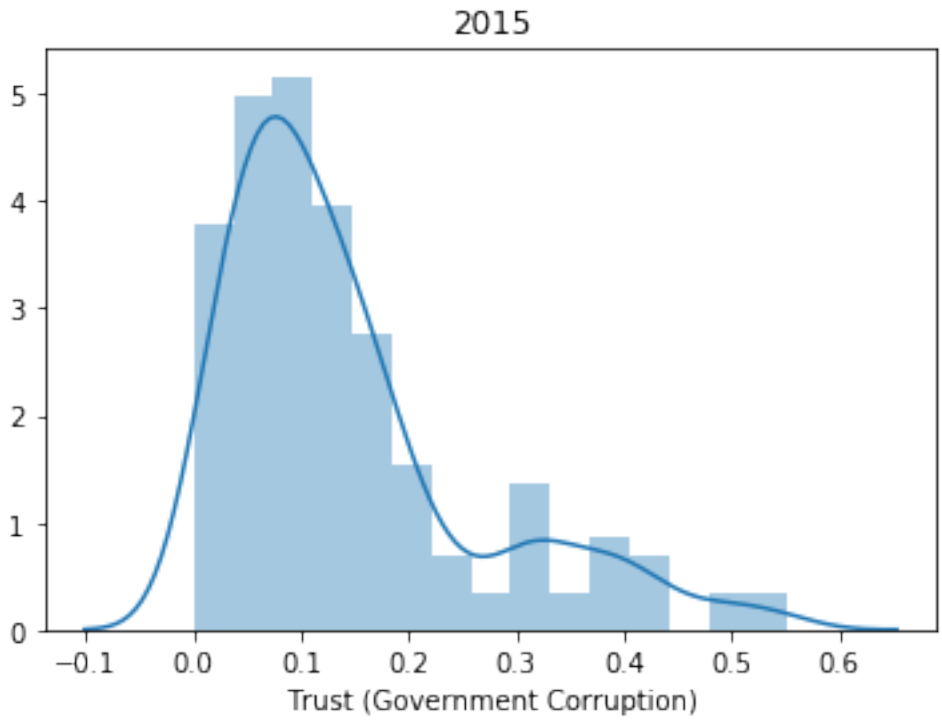
[156]:
```
sns.distplot(df15['Generosity'], bins = 15).set_title(2015)
```

[156]:
```
Text(0.5, 1.0, '2015')
```



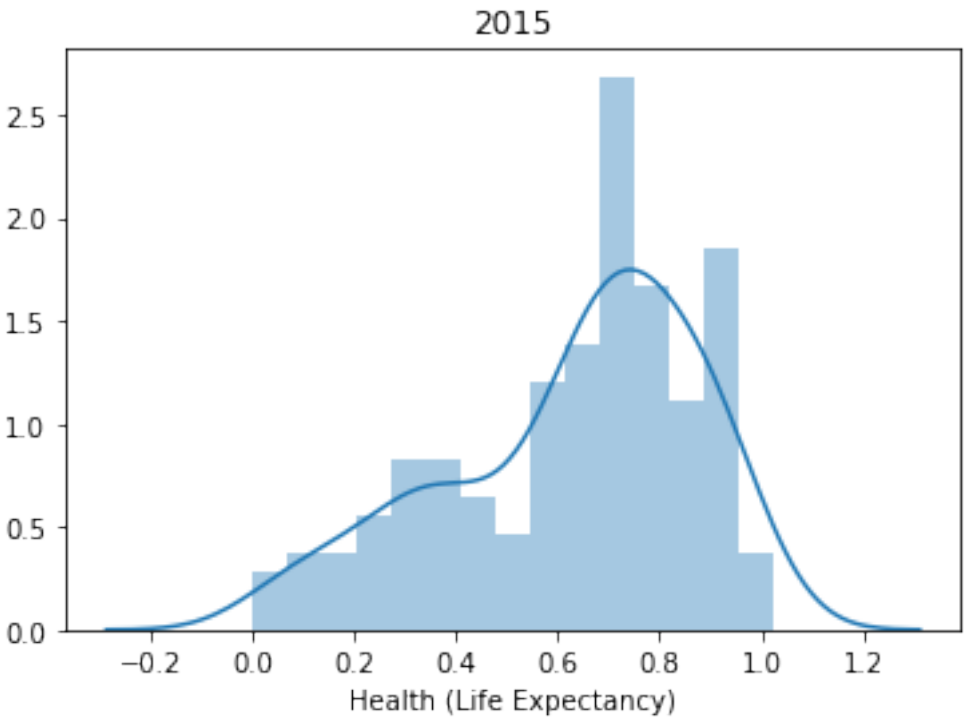
[155]:
```
sns.distplot(df15['Trust (Government Corruption)'], bins = 15).set_title(2015)
# most gov corruption lays between 0-0.1
```

[155]:
```
Text(0.5, 1.0, '2015')
```



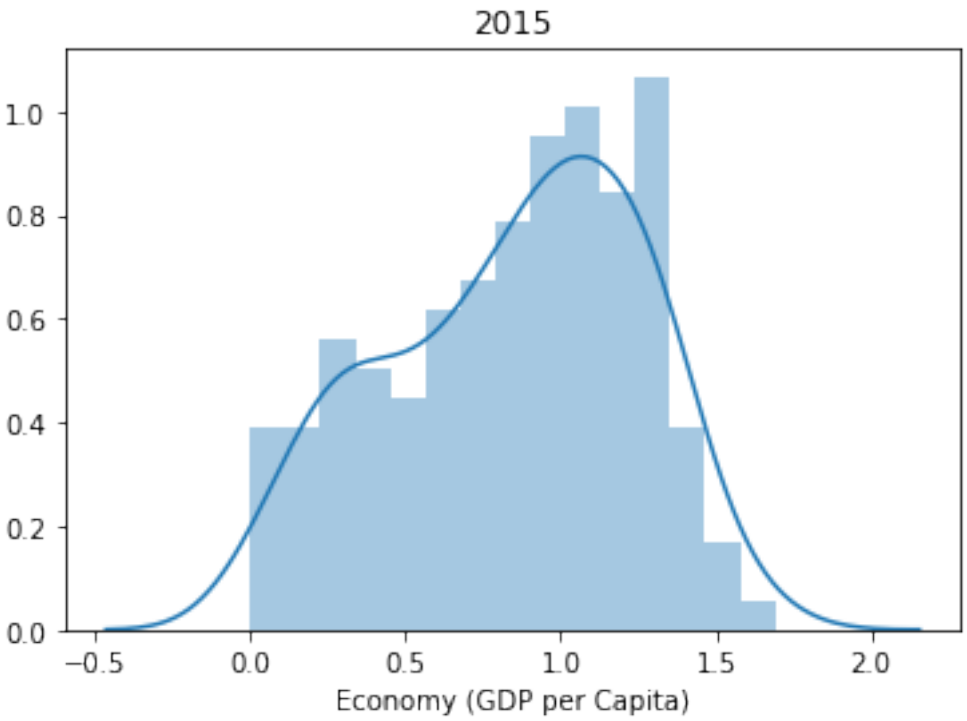
[158]:
```
sns.distplot(df15['Health (Life Expectancy)'], bins = 15).set_title(2015)
# most health exp lays around 0.7 - 0.8
```

[158]: Text(0.5, 1.0, '2015')



[160]:
```
sns.distplot(df15['Economy (GDP per Capita)'], bins = 15).set_title(2015)
```

[160]: Text(0.5, 1.0, '2015')



1.2.3 Probability Mass Function

1.2.4 Region: PMF

[ ]:
```
# Chapter 3: Probability Mass Function of Regions in Dataset
# two scenarios in your data using a PMF, tells you number of regions and␣
 ↪happines scores
# Region & Country
```

[359]:
```
ddf15 = pd.DataFrame(df15["Region"].value_counts())
ddf15 # datafram with counts of region
```

[359]:

	Region
Sub-Saharan Africa	40
Central and Eastern Europe	29
Latin America and Caribbean	22
Western Europe	21
Middle East and Northern Africa	20
Southeastern Asia	9
Southern Asia	7
Eastern Asia	6
Australia and New Zealand	2
North America	2

[360]:
```
lenght = len(df15) # lenght of dataframe
lenght
```

[360]:
```
158
```

[364]:
```
ddf15 = pd.DataFrame(df15["Region"].value_counts())
ddf15
```

[364]:

	Region
Sub-Saharan Africa	40
Central and Eastern Europe	29
Latin America and Caribbean	22
Western Europe	21
Middle East and Northern Africa	20
Southeastern Asia	9
Southern Asia	7
Eastern Asia	6
Australia and New Zealand	2
North America	2

[365]:
```
ddf15.columns = ["Counts"]
ddf15
```

[365]:

	Counts
Sub-Saharan Africa	40
Central and Eastern Europe	29
Latin America and Caribbean	22
Western Europe	21
Middle East and Northern Africa	20
Southeastern Asia	9
Southern Asia	7
Eastern Asia	6
Australia and New Zealand	2
North America	2

[383]:
```
ddf15["Prob"] = ddf15["Counts"]/lenght
ddf15
```

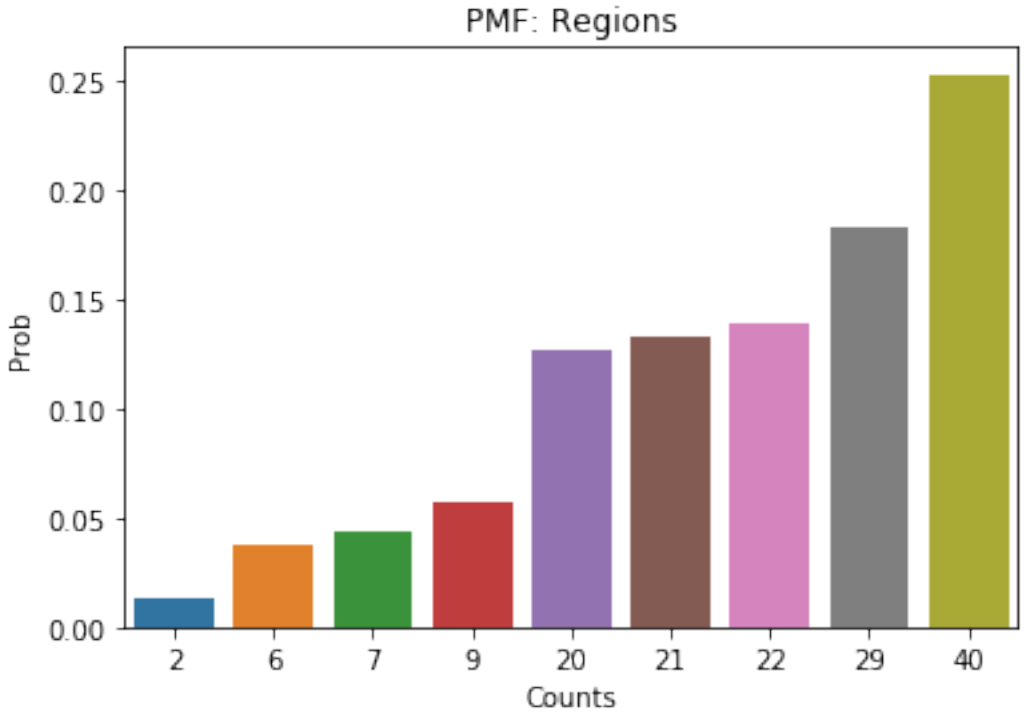
[383]:

	Counts	Prob
Sub-Saharan Africa	40	0.254777
Central and Eastern Europe	29	0.184713
Latin America and Caribbean	22	0.140127
Western Europe	21	0.133758
Middle East and Northern Africa	20	0.127389
Southeastern Asia	9	0.057325
Southern Asia	7	0.044586
Eastern Asia	6	0.038217

```
Australia and New Zealand              2  0.012739
North America                          2  0.012739
```

[376]:
```
sns.barplot(ddf15["Counts"], ddf15["Prob"]).set_title('PMF: Regions')
```

[376]:
```
Text(0.5, 1.0, 'PMF: Regions')
```



1.2.5 Happiness Score: PMF

[388]:
```
ddf2 = pd.DataFrame(df15["Happiness Score"].value_counts())
ddf2 # dataframe with counts of region
```

[388]:
```
        Happiness Score
5.192                 2
4.642                 1
5.098                 1
5.129                 1
5.889                 1
…                     …
4.252                 1
4.633                 1
3.931                 1
7.200                 1
```

```
6.750                1

[157 rows x 1 columns]
```

[379]:
```
lenght = len(ddf2) # lenght of dataframe
lenght
```

[379]:
```
157
```

[380]:
```
ddf2 = pd.DataFrame(df15["Happiness Score"].value_counts())
ddf2
```

[380]:
```
        Happiness Score
5.192                2
4.642                1
5.098                1
5.129                1
5.889                1
…                  …
4.252                1
4.633                1
3.931                1
7.200                1
6.750                1

[157 rows x 1 columns]
```

[381]:
```
ddf2.columns = ["Counts"]
ddf2
```

[381]:
```
        Counts
5.192        2
4.642        1
5.098        1
5.129        1
5.889        1
…          …
4.252        1
4.633        1
3.931        1
7.200        1
6.750        1

[157 rows x 1 columns]
```

[382]:
```
ddf2["Prob"] = ddf2["Counts"]/lenght
ddf2
```

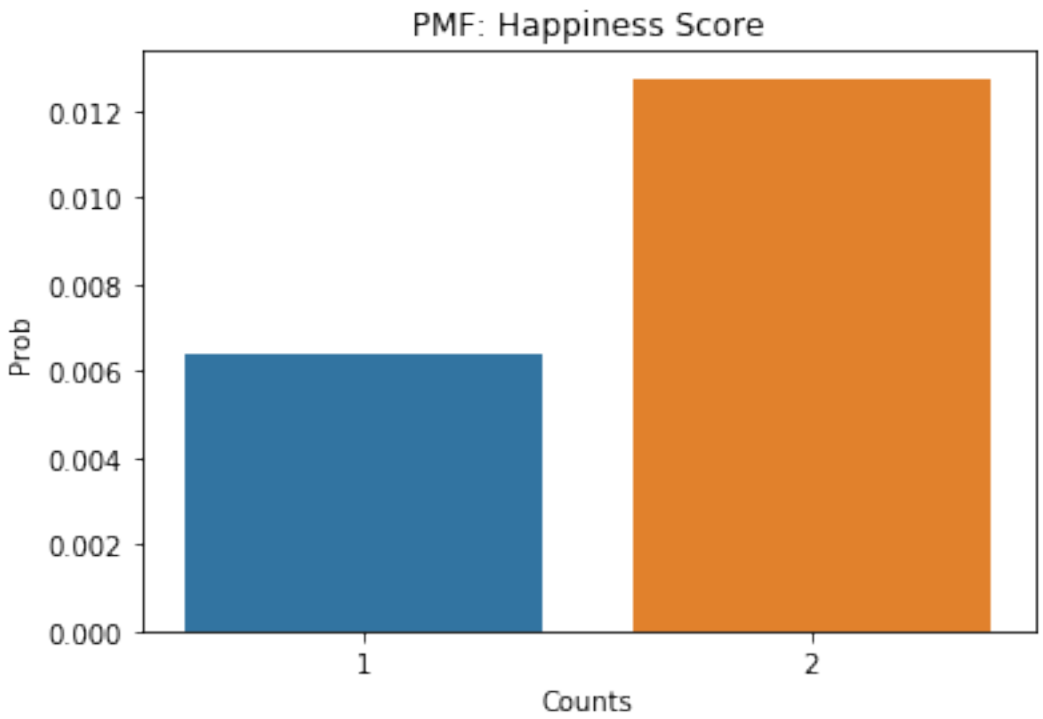
[382]:

	Counts	Prob
5.192	2	0.012739
4.642	1	0.006369
5.098	1	0.006369
5.129	1	0.006369
5.889	1	0.006369
…	…	…
4.252	1	0.006369
4.633	1	0.006369
3.931	1	0.006369
7.200	1	0.006369
6.750	1	0.006369

[157 rows x 2 columns]

[384]:
```
sns.barplot(ddf2["Counts"], ddf2["Prob"]).set_title('PMF: Happiness Score')
```

[384]: Text(0.5, 1.0, 'PMF: Happiness Score')



1.2.6 CDF

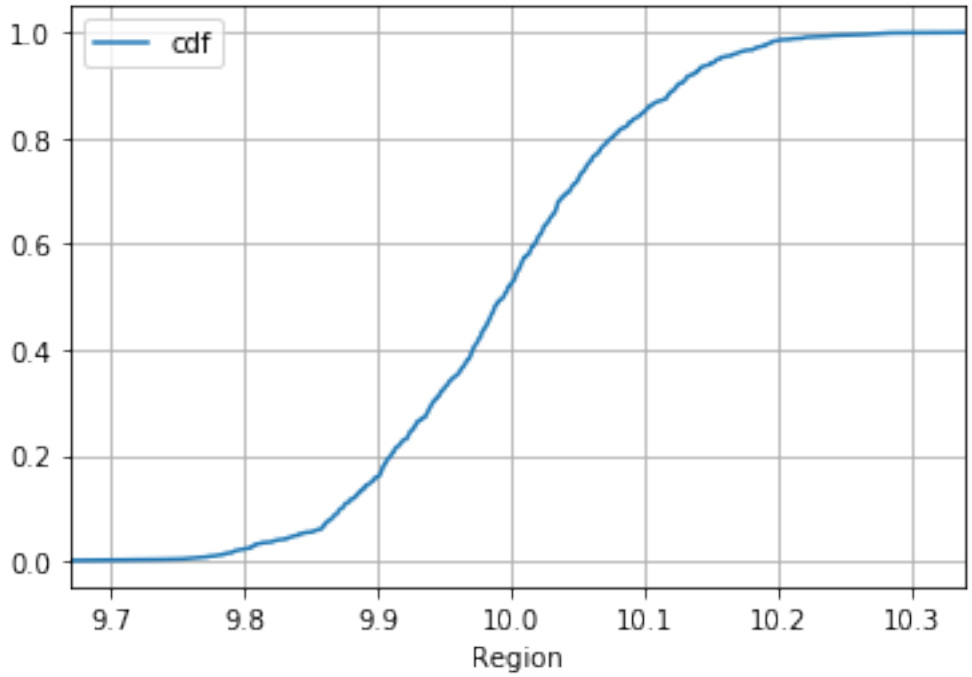
[406]:
```
# Chapter 4: Cumulative Distribution Function
# CDF one of your variables
# Pg 41-44
```

[472]:
```
s = pd.Series(np.random.normal(loc = 10, scale = 0.1, size = 1000), name =␣
↪'Region')
df15 = pd.DataFrame(s)
```

[473]:
```
df155['cdf'] = df155.rank(method = 'average', pct = True)
```

[474]:
```
df155.sort_values('Region').plot(x = 'Region', y = 'cdf', grid = True)
```

[474]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a36d04590>
```

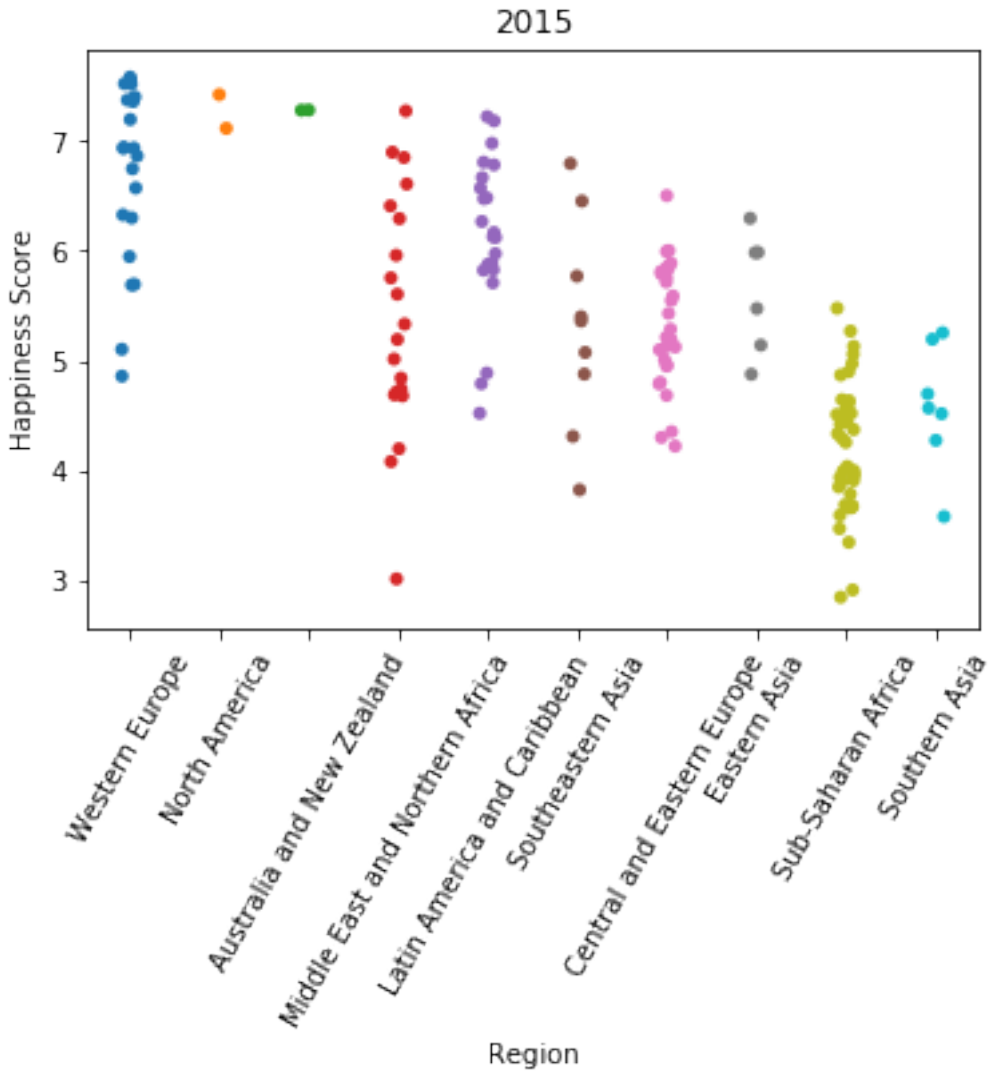


1.2.7 Scatter Plots

[ ]:
```
# Chapter 7
# 2 scatter plots compare 2 variables, correlation/causation
# covariance, Pearson's correlation, and Non-Linear Relationships should also␣
↪be considered during your analysis (Chapter 7)
```

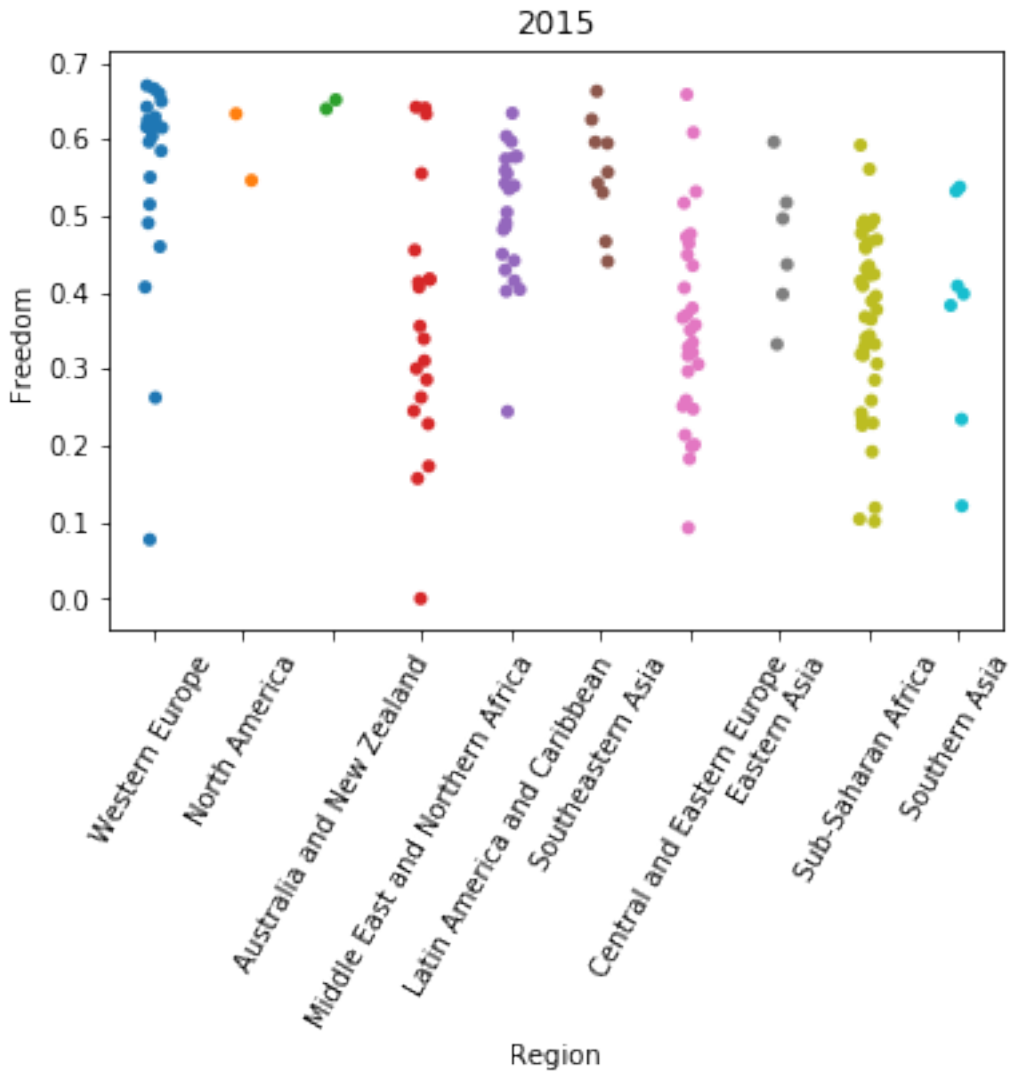
[195]:
```
sns.stripplot(x= "Region",y="Happiness Score", data=df15).set_title(2015)
plt.xticks(rotation = 60)
```

[195]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]), <a list of 10 Text xticklabel objects>)



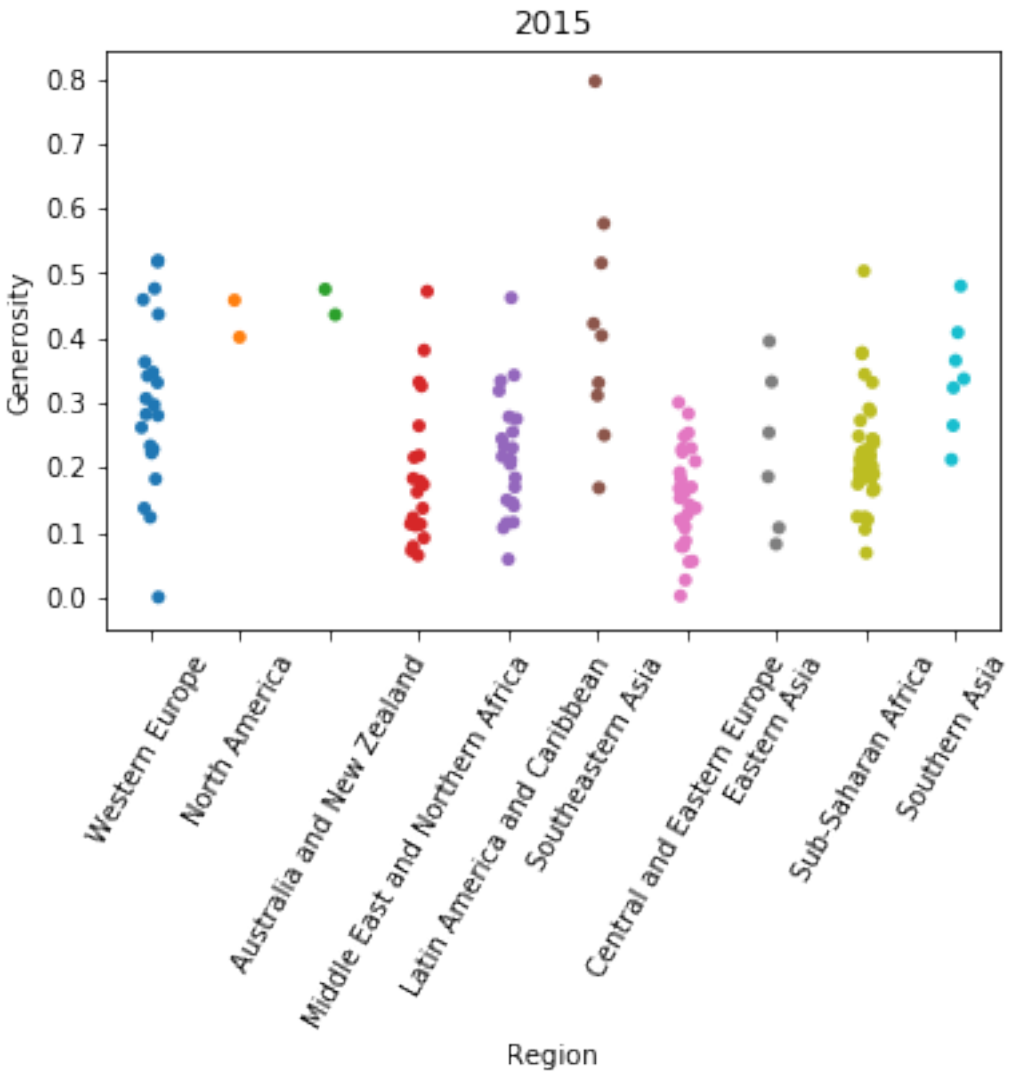
[194]:
```
sns.stripplot(x= "Region",y="Freedom", data=df15).set_title(2015)
plt.xticks(rotation = 60)
```

[194]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]), <a list of 10 Text xticklabel objects>)



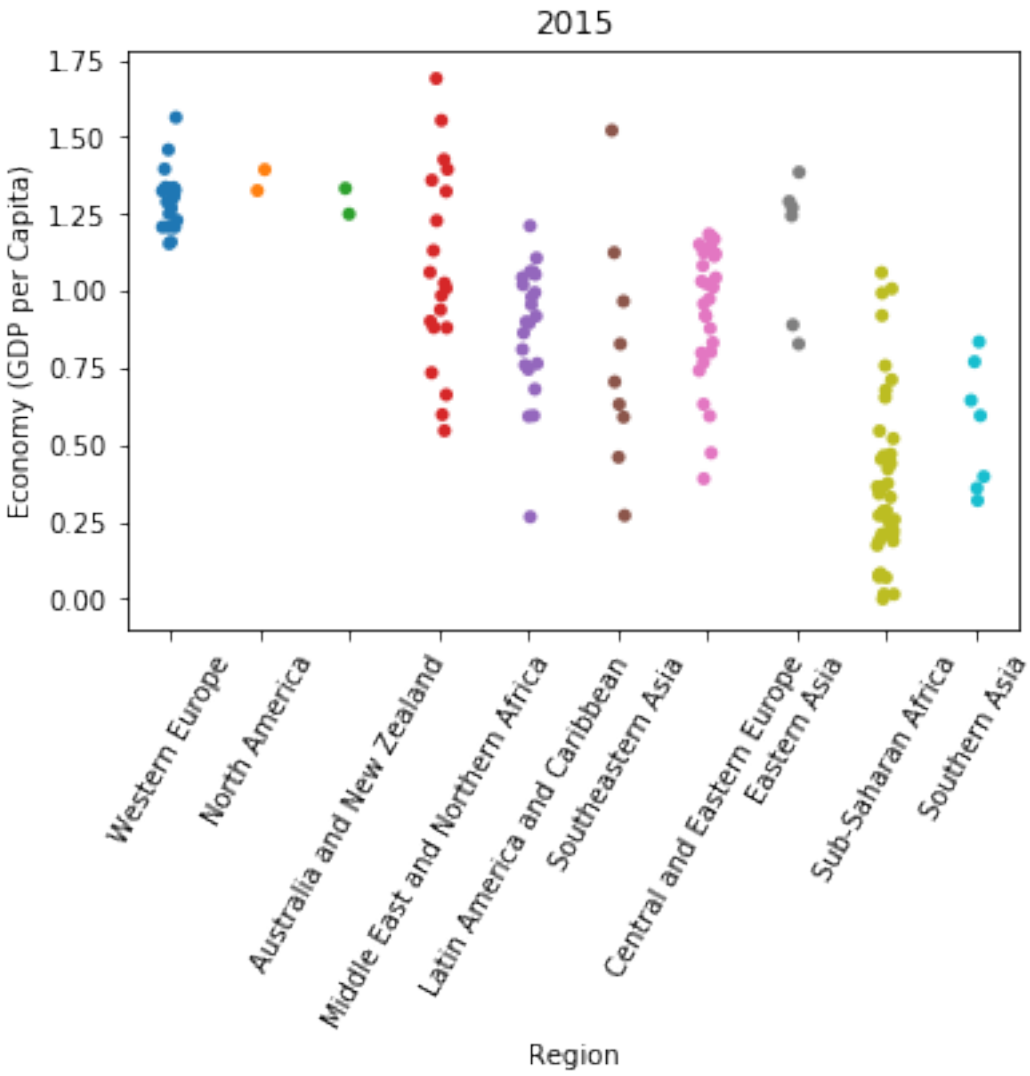
[193]:
```
sns.stripplot(x= "Region",y="Generosity", data=df15).set_title(2015)
plt.xticks(rotation = 60)
```

[193]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]), <a list of 10 Text xticklabel objects>)



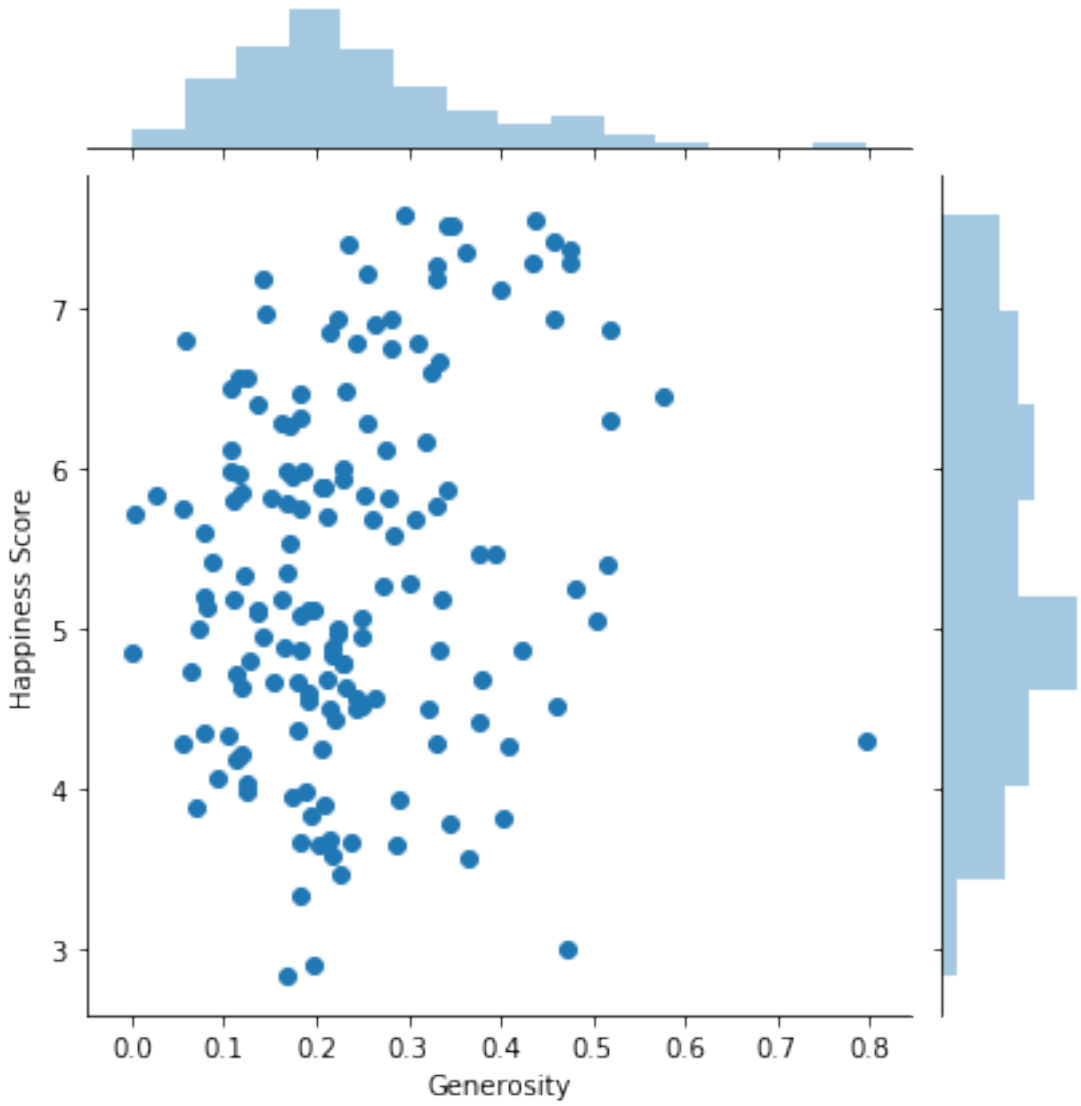
[192]:
```
sns.stripplot(x= "Region",y="Economy (GDP per Capita)", data=df15).
↪set_title(2015)
plt.xticks(rotation = 60)
```

[192]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]), <a list of 10 Text xticklabel objects>)



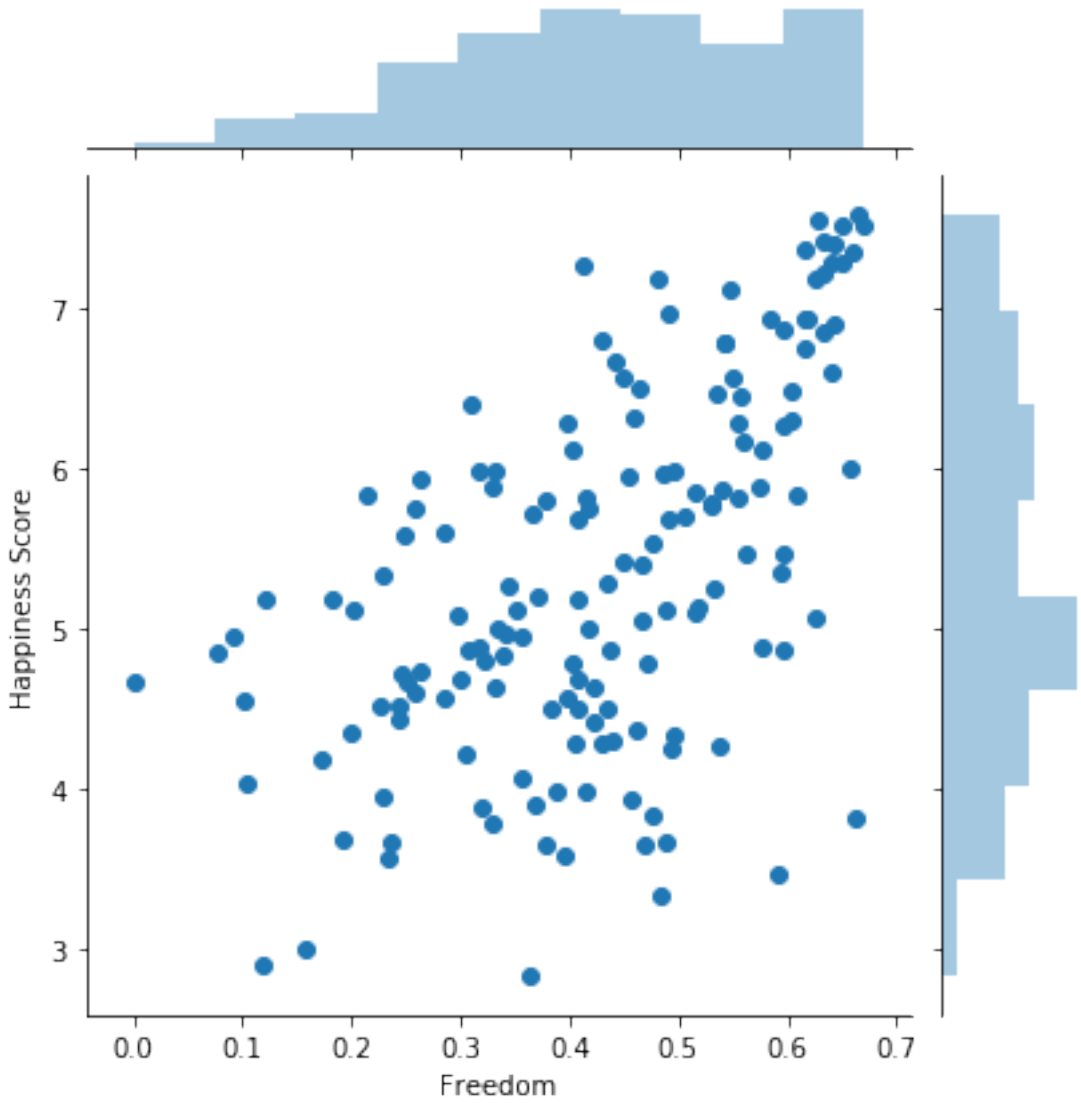
[209]:
```
sns.jointplot(x="Generosity", y="Happiness Score", data=df15) # correlation
```

[209]: <seaborn.axisgrid.JointGrid at 0x1a29ba1ed0>



[208]:
```
sns.jointplot(x="Freedom", y="Happiness Score", data=df15) # linear correlation
```

[208]:
```
<seaborn.axisgrid.JointGrid at 0x1a23644c90>
```

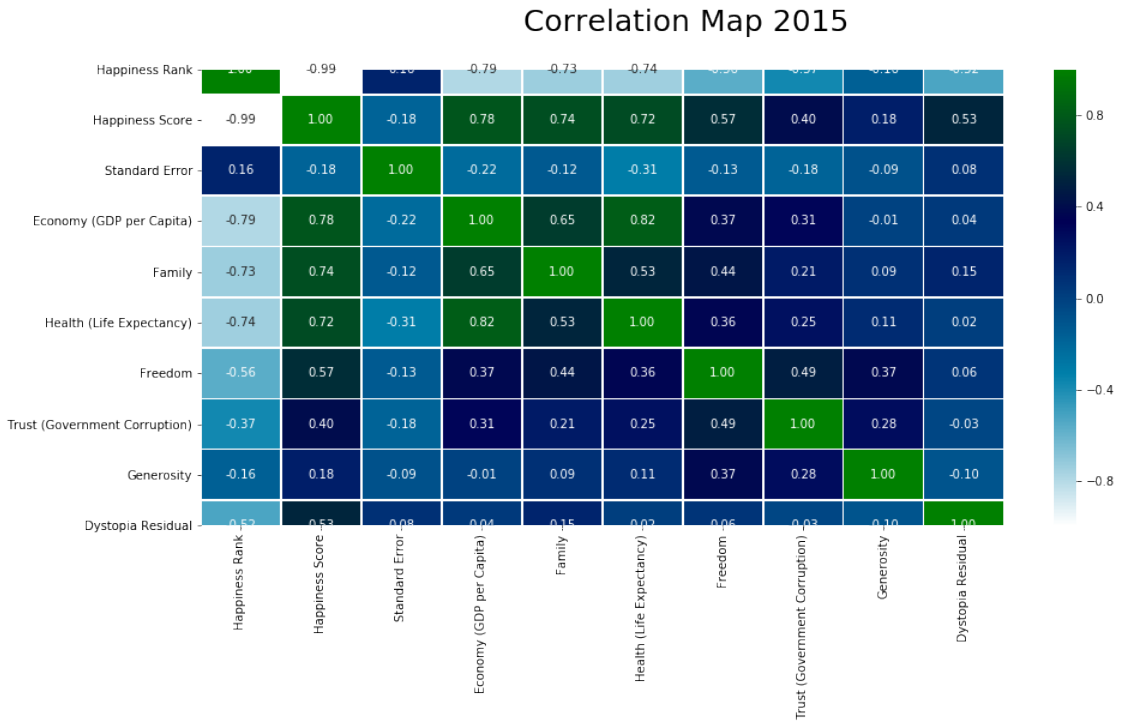


1.2.8 Correlation & Correlation Test

1.2.9 Pearson Correlation Test

[430]:
```
f, ax = plt.subplots(figsize=(15, 7))
sns.heatmap(df15.corr(), annot=True, linewidths=.7, cmap="ocean_r", fmt=".2f",␣
↪ax=ax)  # seaborn has very simple solution for heatmap

plt.suptitle("Correlation Map 2015", fontsize=25)

plt.show()   # whitest(-) and greenest(+) = most correlated
```

```
# Calculated using Pearson: measures the degree of the relationship btw␣
 ↪linearly related variables
# Highest correlation: Economy, Family, Health Life Exp, Freedom.
# Lowest correlation: Rank, Generosity
```



1.2.10 Kendall Correlation Test

[334]:
```
df15.corr(method="kendall")
# measures strenght of dependence between 2 variables
```

[334]:

	Happiness Rank	Happiness Score	\
Happiness Rank	1.000000	-1.000000	
Happiness Score	-1.000000	1.000000	
Standard Error	0.147823	-0.147823	
Economy (GDP per Capita)	-0.592945	0.592945	
Family	-0.577303	0.577303	
Health (Life Expectancy)	-0.554185	0.554185	
Freedom	-0.413465	0.413465	
Trust (Government Corruption)	-0.193920	0.193920	
Generosity	-0.112719	0.112719	
Dystopia Residual	-0.371377	0.371377	

	Standard Error	Economy (GDP per Capita)	\

Happiness Rank	0.147823	-0.592945
Happiness Score	-0.147823	0.592945
Standard Error	1.000000	-0.158381
Economy (GDP per Capita)	-0.158381	1.000000
Family	-0.148381	0.485286
Health (Life Expectancy)	-0.201210	0.652772
Freedom	-0.125640	0.286302
Trust (Government Corruption)	-0.097339	0.142552
Generosity	-0.071449	-0.005241
Dystopia Residual	-0.018548	0.053455

	Family	Health (Life Expectancy)	Freedom	\
Happiness Rank	-0.577303	-0.554185	-0.413465	
Happiness Score	0.577303	0.554185	0.413465	
Standard Error	-0.148381	-0.201210	-0.125640	
Economy (GDP per Capita)	0.485286	0.652772	0.286302	
Family	1.000000	0.407176	0.369830	
Health (Life Expectancy)	0.407176	1.000000	0.271074	
Freedom	0.369830	0.271074	1.000000	
Trust (Government Corruption)	0.123523	0.107805	0.325580	
Generosity	0.096348	0.070631	0.283399	
Dystopia Residual	0.125695	0.052248	0.062807	

	Trust (Government Corruption)	Generosity	\
Happiness Rank	-0.193920	-0.112719	
Happiness Score	0.193920	0.112719	
Standard Error	-0.097339	-0.071449	
Economy (GDP per Capita)	0.142552	-0.005241	
Family	0.123523	0.096348	
Health (Life Expectancy)	0.107805	0.070631	
Freedom	0.325580	0.283399	
Trust (Government Corruption)	1.000000	0.142068	
Generosity	0.142068	1.000000	
Dystopia Residual	0.039831	0.005079	

	Dystopia Residual
Happiness Rank	-0.371377
Happiness Score	0.371377
Standard Error	-0.018548
Economy (GDP per Capita)	0.053455
Family	0.125695
Health (Life Expectancy)	0.052248
Freedom	0.062807
Trust (Government Corruption)	0.039831
Generosity	0.005079
Dystopia Residual	1.000000

[438]:
```python
df15.cov() # covariance how each column relates to another
```

[438]:
```
                               Happiness Rank  Happiness Score  \
Happiness Rank                    2093.461743       -51.975613
Happiness Score                    -51.975613         1.311048
Standard Error                       0.124358        -0.003480
Economy (GDP per Capita)           -14.483883         0.360476
Family                              -9.142720         0.230969
Health (Life Expectancy)            -8.316021         0.204881
Freedom                             -3.839647         0.098042
Trust (Government Corruption)       -2.044785         0.054316
Generosity                          -0.928243         0.026156
Dystopia Residual                  -13.220847         0.336225

                               Standard Error  Economy (GDP per Capita)  \
Happiness Rank                       0.124358                -14.483883
Happiness Score                     -0.003480                  0.360476
Standard Error                       0.000294                 -0.001504
Economy (GDP per Capita)            -0.001504                  0.162506
Family                              -0.000564                  0.070852
Health (Life Expectancy)            -0.001315                  0.081323
Freedom                             -0.000335                  0.022495
Trust (Government Corruption)       -0.000367                  0.014898
Generosity                          -0.000192                 -0.000534
Dystopia Residual                    0.000797                  0.008939

                                 Family  Health (Life Expectancy)   Freedom  \
Happiness Rank                -9.142720                 -8.316021 -3.839647
Happiness Score                0.230969                  0.204881  0.098042
Standard Error                -0.000564                 -0.001315 -0.000335
Economy (GDP per Capita)       0.070852                  0.081323  0.022495
Family                         0.074185                  0.035741  0.018122
Health (Life Expectancy)       0.035741                  0.061047  0.013422
Freedom                        0.018122                  0.013422  0.022708
Trust (Government Corruption)  0.006722                  0.007365  0.008927
Generosity                     0.003020                  0.003391  0.007138
Dystopia Residual              0.022332                  0.002596  0.005237

                               Trust (Government Corruption)  Generosity  \
Happiness Rank                                     -2.044785   -0.928243
Happiness Score                                     0.054316    0.026156
Standard Error                                     -0.000367   -0.000192
Economy (GDP per Capita)                            0.014898   -0.000534
Family                                              0.006722    0.003020
Health (Life Expectancy)                            0.007365    0.003391
Freedom                                             0.008927    0.007138
Trust (Government Corruption)                       0.014408    0.004199
```

```
Generosity                                      0.004199    0.016049
Dystopia Residual                              -0.002200   -0.007104
```

	Dystopia Residual
Happiness Rank	-13.220847
Happiness Score	0.336225
Standard Error	0.000797
Economy (GDP per Capita)	0.008939
Family	0.022332
Health (Life Expectancy)	0.002596
Freedom	0.005237
Trust (Government Corruption)	-0.002200
Generosity	-0.007104
Dystopia Residual	0.306417

1.2.11 Regression Analysis

Regression analysis will be applied to the dependet variable of Happiness score using freedom as a predictor.

[281]:
```
# simple linear regression
# dependent: Happiness Score
# independet/predictor: Freedom
```

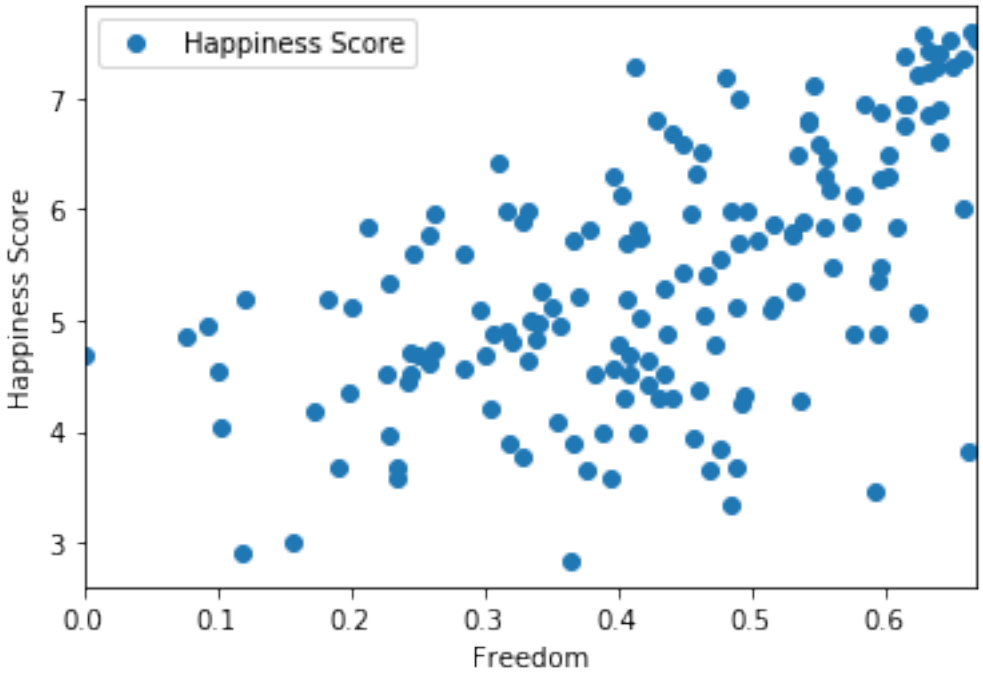
[282]:
```
data = df15.loc[:,['Freedom','Happiness Score']]
data.head(5)
```

[282]:

	Freedom	Happiness Score
0	0.66557	7.587
1	0.62877	7.561
2	0.64938	7.527
3	0.66973	7.522
4	0.63297	7.427

[283]:
```
data.plot(x="Freedom", y="Happiness Score", style="o")
plt.xlabel("Freedom")
plt.ylabel("Happiness Score")
plt.show
```

[283]:
```
<function matplotlib.pyplot.show(*args, **kw)>
```



[284]:
```
X = pd.DataFrame(data['Freedom'])
y = pd.DataFrame(data['Happiness Score'])
```

[285]:
```
# splitting data to create train and test data sets
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,␣
 ↪random_state=1)
```

[286]:
```
# Shape of train and test data sets
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(126, 1)
(32, 1)
(126, 1)
(32, 1)
```

[304]:
```
# Training algorithm
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

[304]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

[305]:
```
print(regressor.intercept_)
```

```
[3.63176702]
```

[306]:
```
print(regressor.coef_)
```

```
[[4.01561069]]
```

[311]:
```
# Comparing predicted values to actual values

y_pred = regressor.predict(X_test)
y_pred = pd.DataFrame(y_pred, columns=['Predicted'])
```

[312]:
```
y_pred # predicted values
```

[312]:

	Predicted
0	5.437748
1	4.615551
2	5.824451
3	5.264554
4	5.326957
5	6.208062
6	5.503483
7	6.176700
8	5.875851
9	4.003010
10	4.862873
11	4.975270
12	5.270939
13	5.066665
14	5.155209
15	4.823239
16	5.950100
17	5.705790
18	4.949610
19	4.952020
20	5.764940
21	5.476980
22	5.546892
23	6.208544
24	4.428745
25	6.055670
26	6.104701
27	5.768313
28	4.612580

```
29   5.305915
30   5.169023
31   4.107617
```

[293]:
```
y_test # actual values
```

[293]:
```
     Happiness Score
29              6.574
107             4.715
14              7.119
81              5.192
124             4.419
19              6.901
73              5.399
11              7.226
40              6.168
95              4.949
100             4.867
92              5.007
108             4.694
94              4.959
58              5.813
88              5.098
42              6.123
53              5.855
51              5.889
145             3.781
59              5.791
35              6.329
143             3.845
5               7.406
126             4.350
31              6.485
16              6.946
78              5.253
118             4.518
91              5.013
120             4.514
156             2.905
```

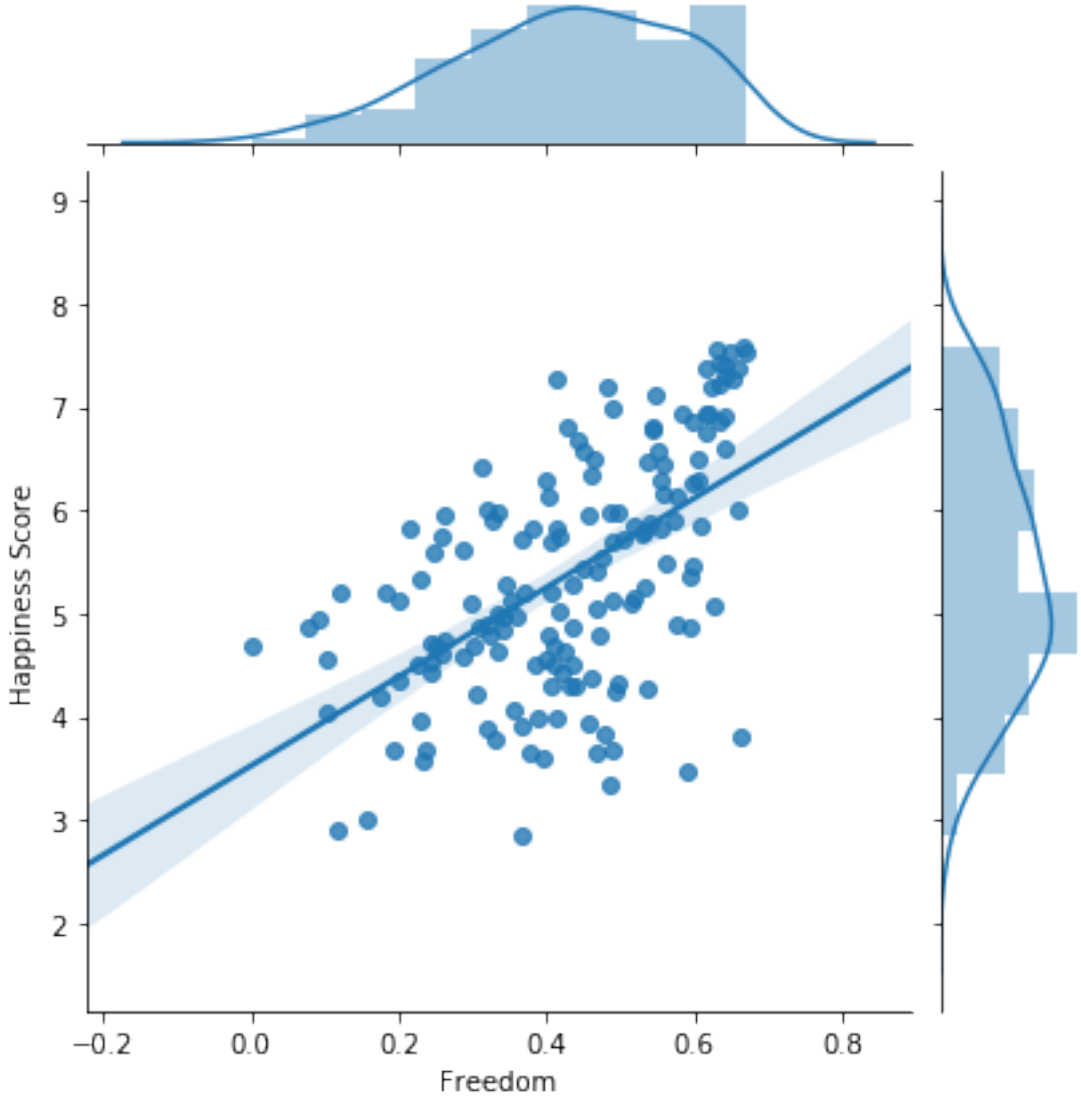
[315]:
```
# Evaluating the algorithm,
# MAE: measures accuracy for the Freedom variable on Happiness score, average␣
 ↪diff.
# between predicted and actual is minimal at 0.58
# MeanSquareError: how close a fitted line is to data points
# RMSE: measures average magnitude of error = small variation 0.74
# lower values for both is better
```

```
from sklearn import metrics
print("Mean Absolute Error:", metrics.mean_absolute_error(y_test, y_pred))
print("Mean Squared Error:", metrics.mean_squared_error(y_test, y_pred))
print("Root Mean Squared Error:", np.sqrt(metrics.mean_squared_error(y_test,␣
 ↪y_pred)))
```

```
Mean Absolute Error: 0.5802626440433369
Mean Squared Error: 0.554039874779413
Root Mean Squared Error: 0.7443385484975321
```

[255]:
```
sns.jointplot(x="Freedom", y="Happiness Score", data=df15, kind='reg')
#linear regression Happiness score w/ freedom (predictor)
```

[255]: <seaborn.axisgrid.JointGrid at 0x1a2f31b110>



1.3 2016 Dataset

First five rows of 2016 datset shown below.

1.3.1 Variables:

Country: Name of country

Region : Region the country pertains to

Happiness Rank : Rank of the country based on the Happiness Score

Happiness Score : A metric measured in 2015 by asking the sampled people: "How would you rate your happiness on a scale of 0 to 10 where 10 is the happiest."

Lower Confidence Interval: the lower confidence limit of an estimate of a parameter

Upper Confidence Interval: the upper confidence limit of an estimate of a parameter

Economy (GDP per Capita): real GDP per capita

Family : social support

Health (Life Expectancy): healthy life expectancy

Freedom : freedom to make life choices

Trust (Government Corruption): perceptions of corruption

Generosity : perceptions of generosity

Dystopia Residual: each country is compared against a hypothetical nation that represents the lowest national averages for each key variable and is, along with residual error, used as a regression benchmark

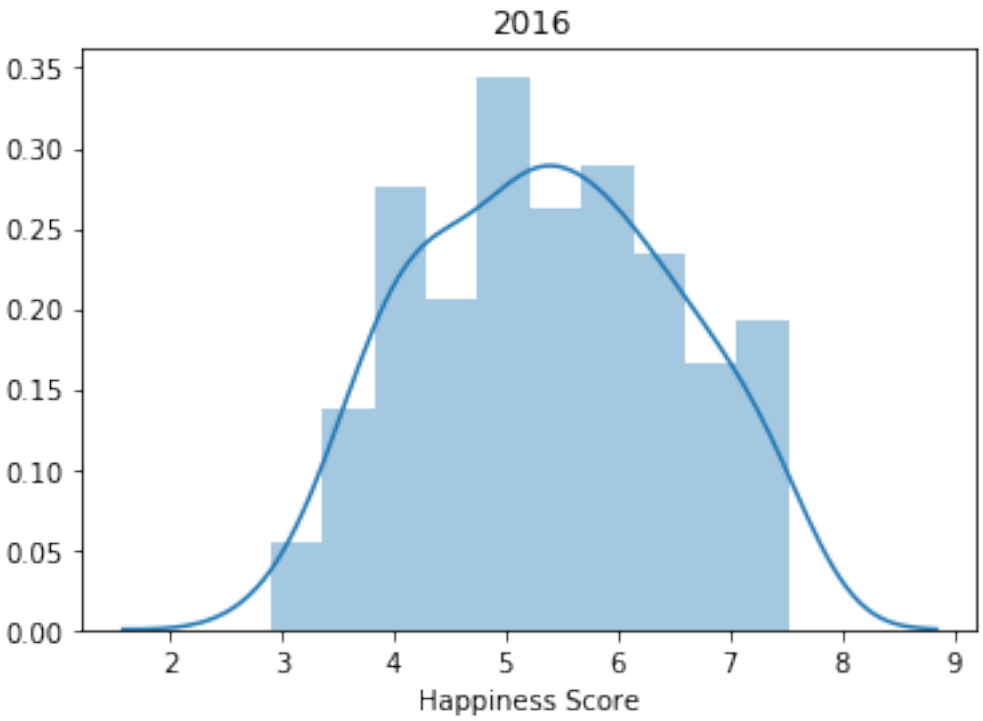
[89]:
```
df16.head(5)
```

[89]:
```
       Country          Region  Happiness Rank  Happiness Score  \
0      Denmark  Western Europe               1            7.526
1  Switzerland  Western Europe               2            7.509
2      Iceland  Western Europe               3            7.501
3       Norway  Western Europe               4            7.498
4      Finland  Western Europe               5            7.413

   Lower Confidence Interval  Upper Confidence Interval  \
0                      7.460                      7.592
1                      7.428                      7.590
2                      7.333                      7.669
3                      7.421                      7.575
4                      7.351                      7.475

   Economy (GDP per Capita)   Family  Health (Life Expectancy)  Freedom  \
```

```
0                    1.44178  1.16374                    0.79504  0.57941
1                    1.52733  1.14524                    0.86303  0.58557
2                    1.42666  1.18326                    0.86733  0.56624
3                    1.57744  1.12690                    0.79579  0.59609
4                    1.40598  1.13464                    0.81091  0.57104

   Trust (Government Corruption)  Generosity  Dystopia Residual
0                        0.44453     0.36171            2.73939
1                        0.41203     0.28083            2.69463
2                        0.14975     0.47678            2.83137
3                        0.35776     0.37895            2.66465
4                        0.41004     0.25492            2.82596
```

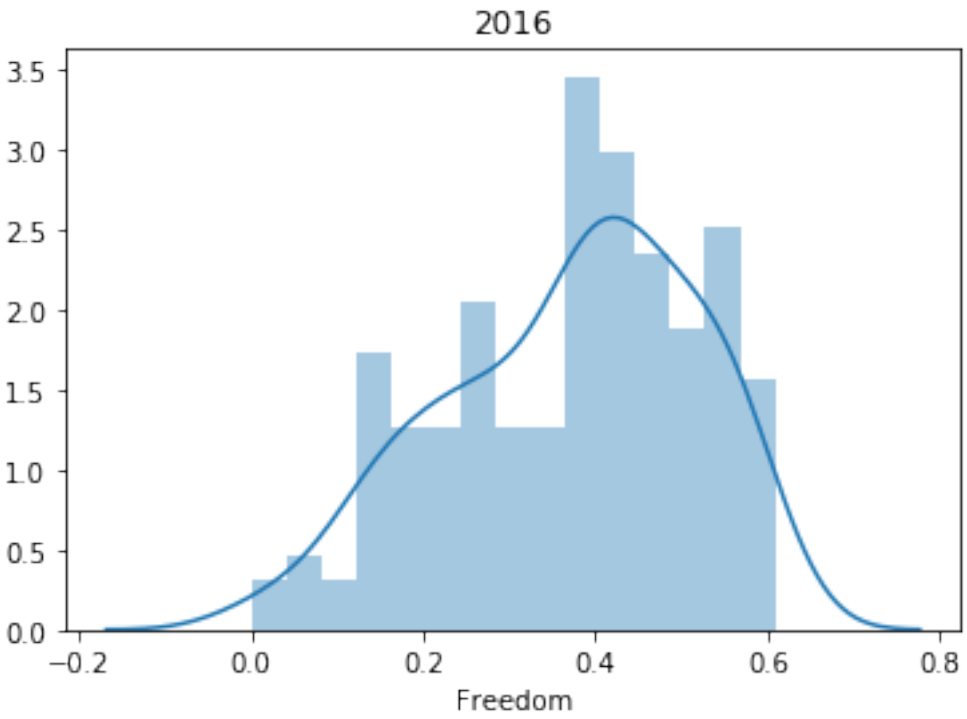
[140]:
```
sns.distplot(df16['Happiness Score'], bins = 10).set_title(2016)
```

[140]:
```
Text(0.5, 1.0, '2016')
```



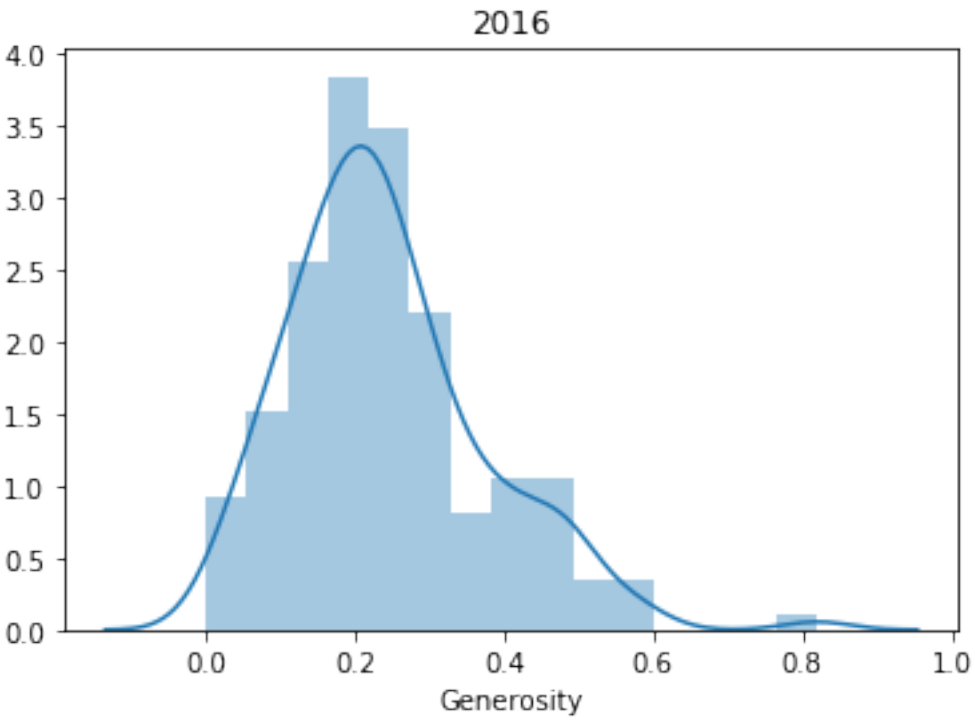
[145]:
```
sns.distplot(df16['Freedom'], bins = 15).set_title(2016)
```

[145]:
```
Text(0.5, 1.0, '2016')
```



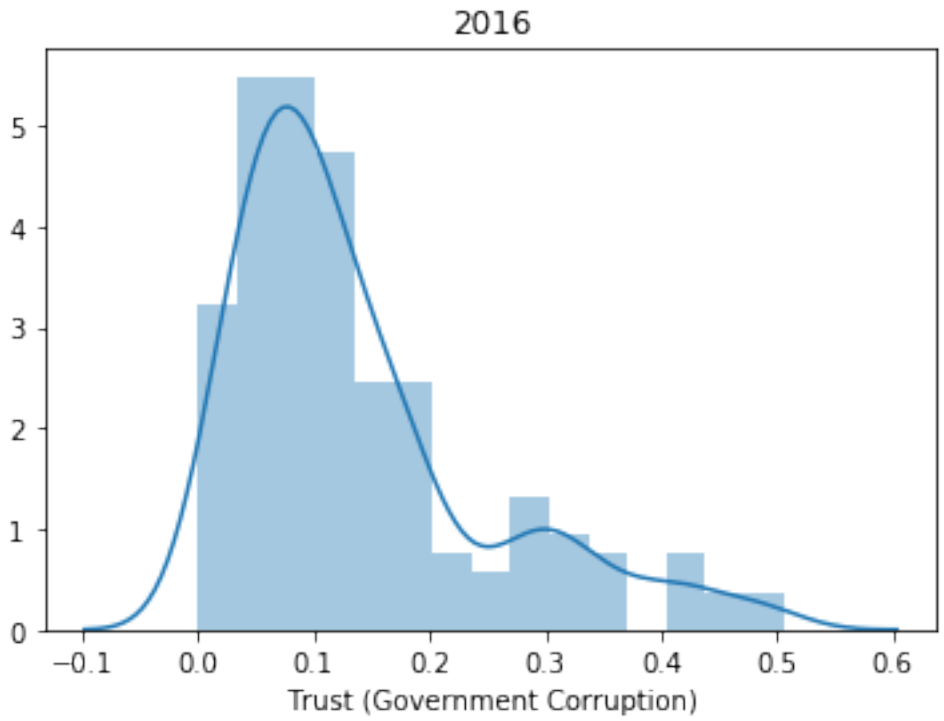
[162]:
```
sns.distplot(df16['Generosity'], bins = 15).set_title(2016)
```

[162]:
```
Text(0.5, 1.0, '2016')
```



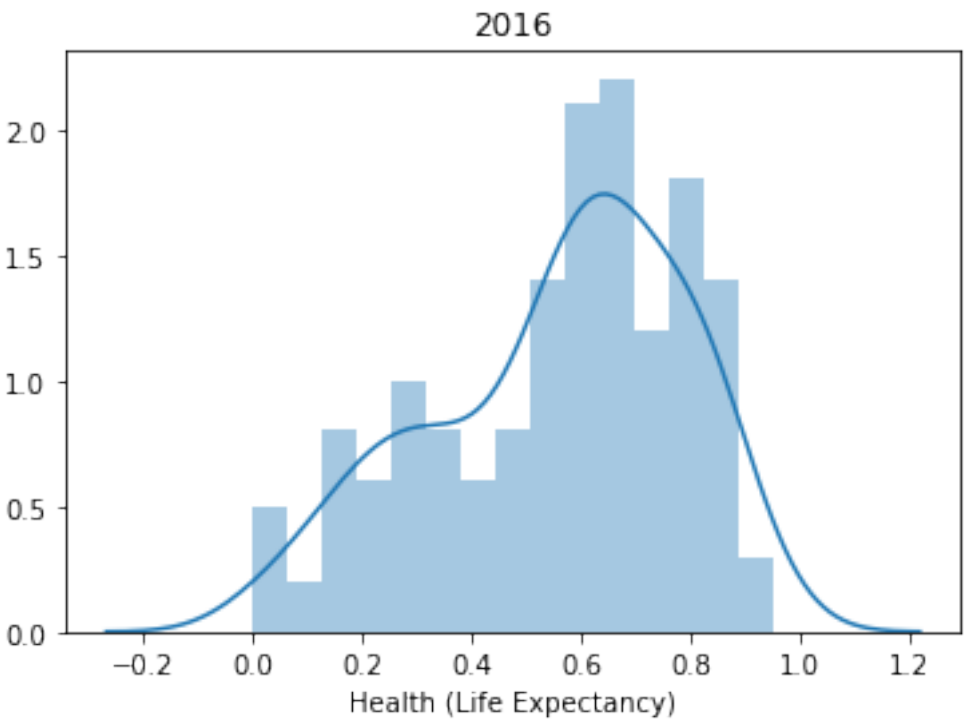
[163]:
```
sns.distplot(df16['Trust (Government Corruption)'], bins = 15).set_title(2016)
```

[163]:
```
Text(0.5, 1.0, '2016')
```



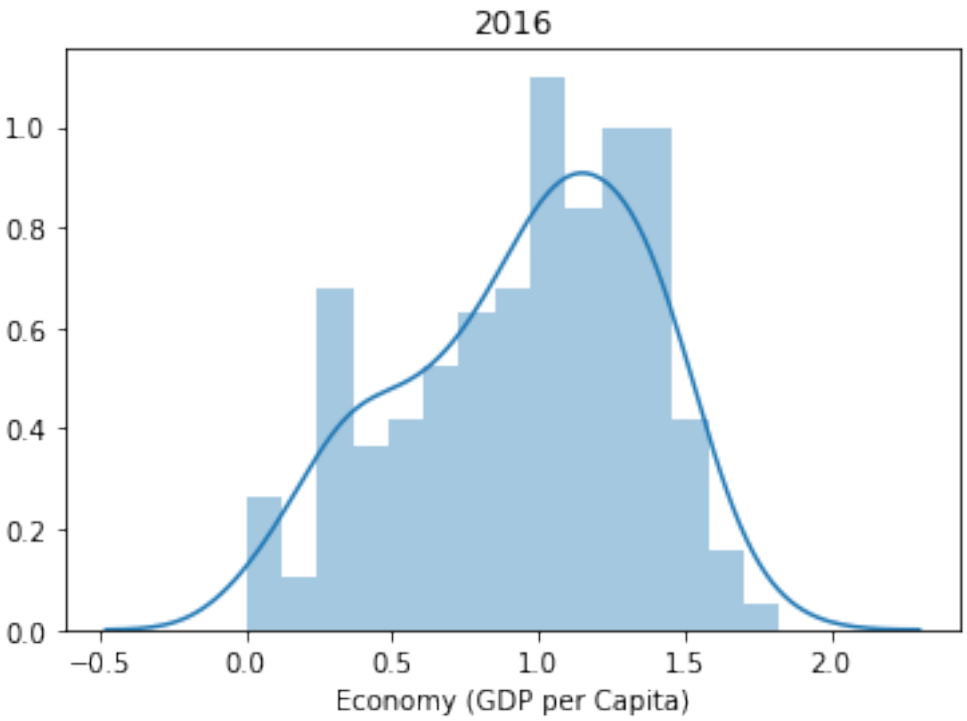
[164]:
```
sns.distplot(df16['Health (Life Expectancy)'], bins = 15).set_title(2016)
```

[164]: Text(0.5, 1.0, '2016')



[432]:
```
sns.distplot(df16['Economy (GDP per Capita)'], bins = 15).set_title(2016)
```

[432]:
```
Text(0.5, 1.0, '2016')
```

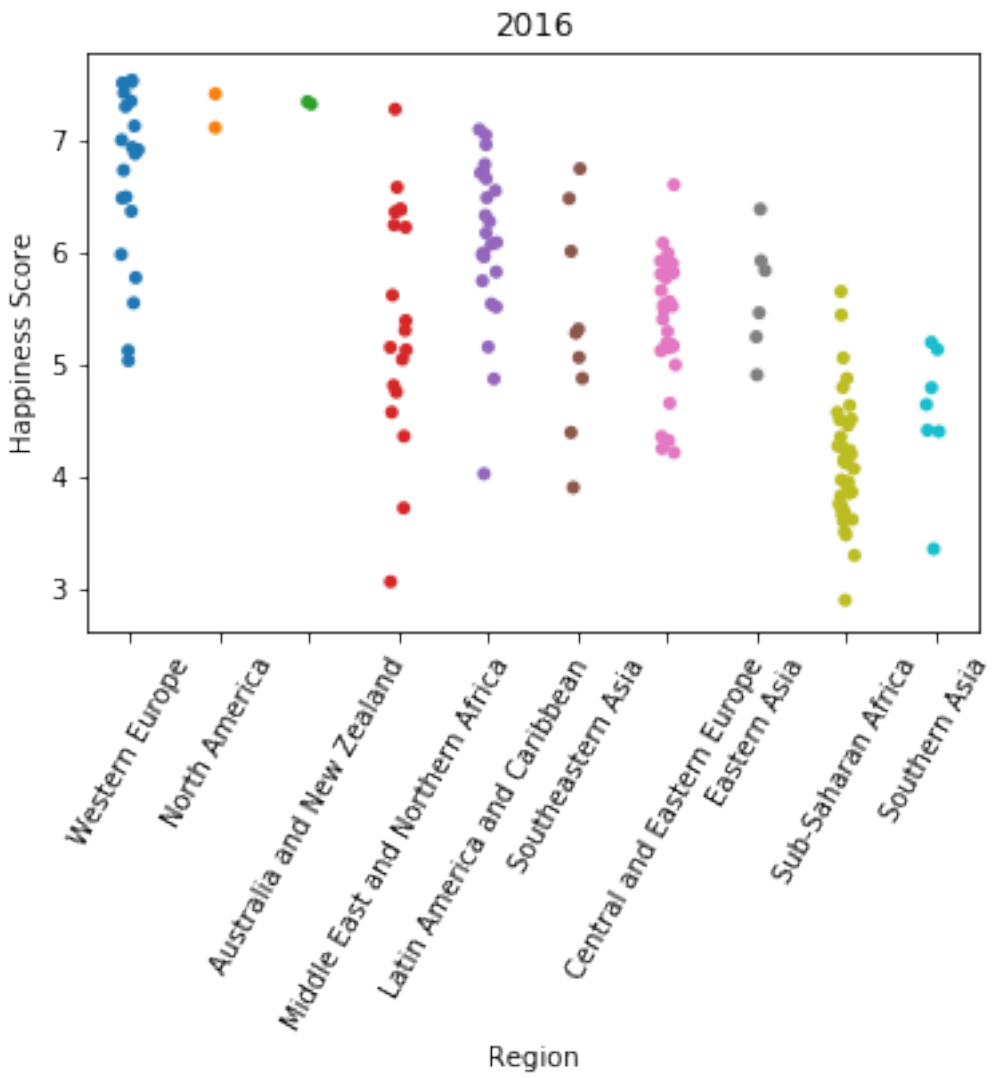


[433]:
```
#s = pd.Series(np.random.normal(loc = 10, scale = 0.1, size = 1000), name =␣
↪'Region')
#df16 = pd.DataFrame(s)
# df166['cdf'] = df166.rank(method = 'average', pct = True
# df166.sort_values('Region').plot(x = 'Region', y = 'cdf', grid = True)
```

1.3.2 Scatter Plots

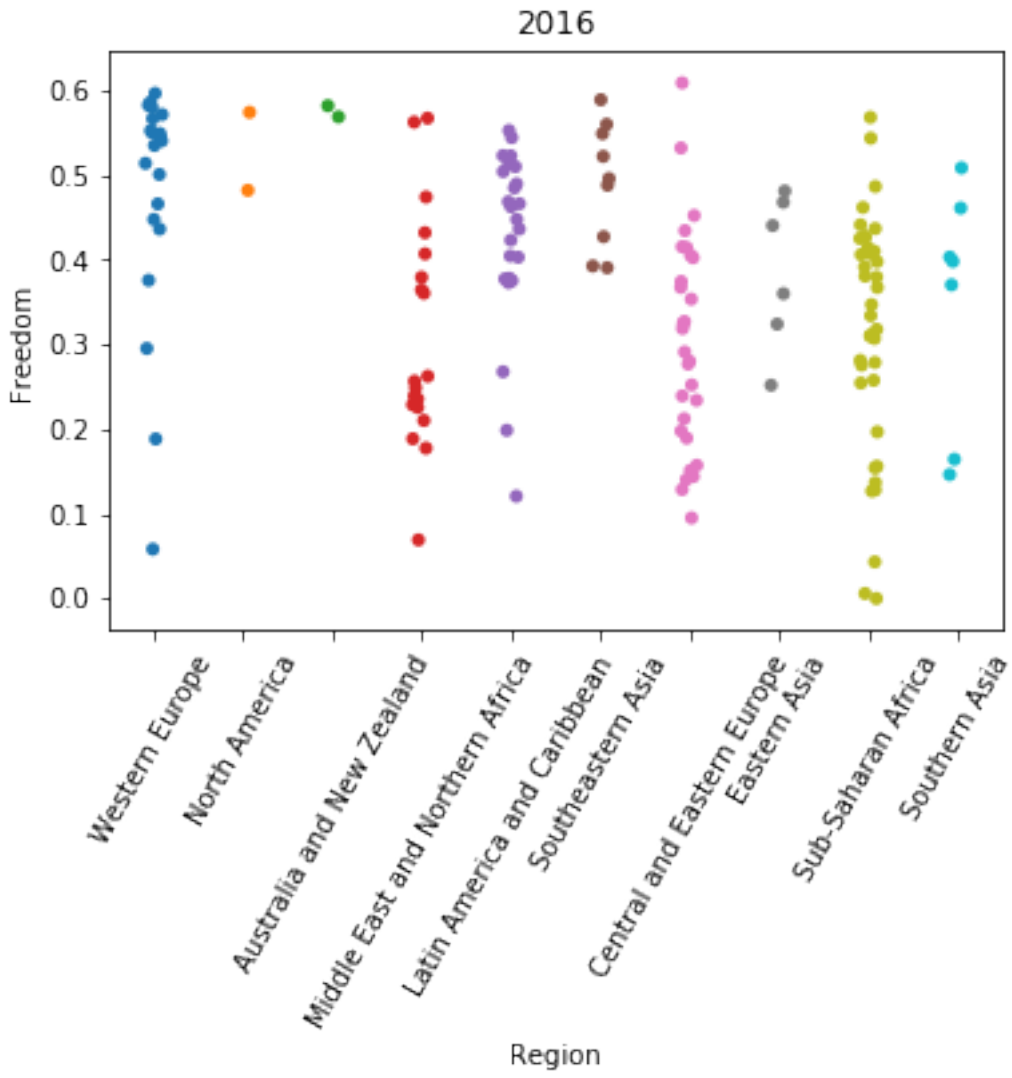
[196]:
```
sns.stripplot(x= "Region",y="Happiness Score", data=df16).set_title(2016)
plt.xticks(rotation = 60)
```

[196]:
```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]), <a list of 10 Text xticklabel objects>)
```



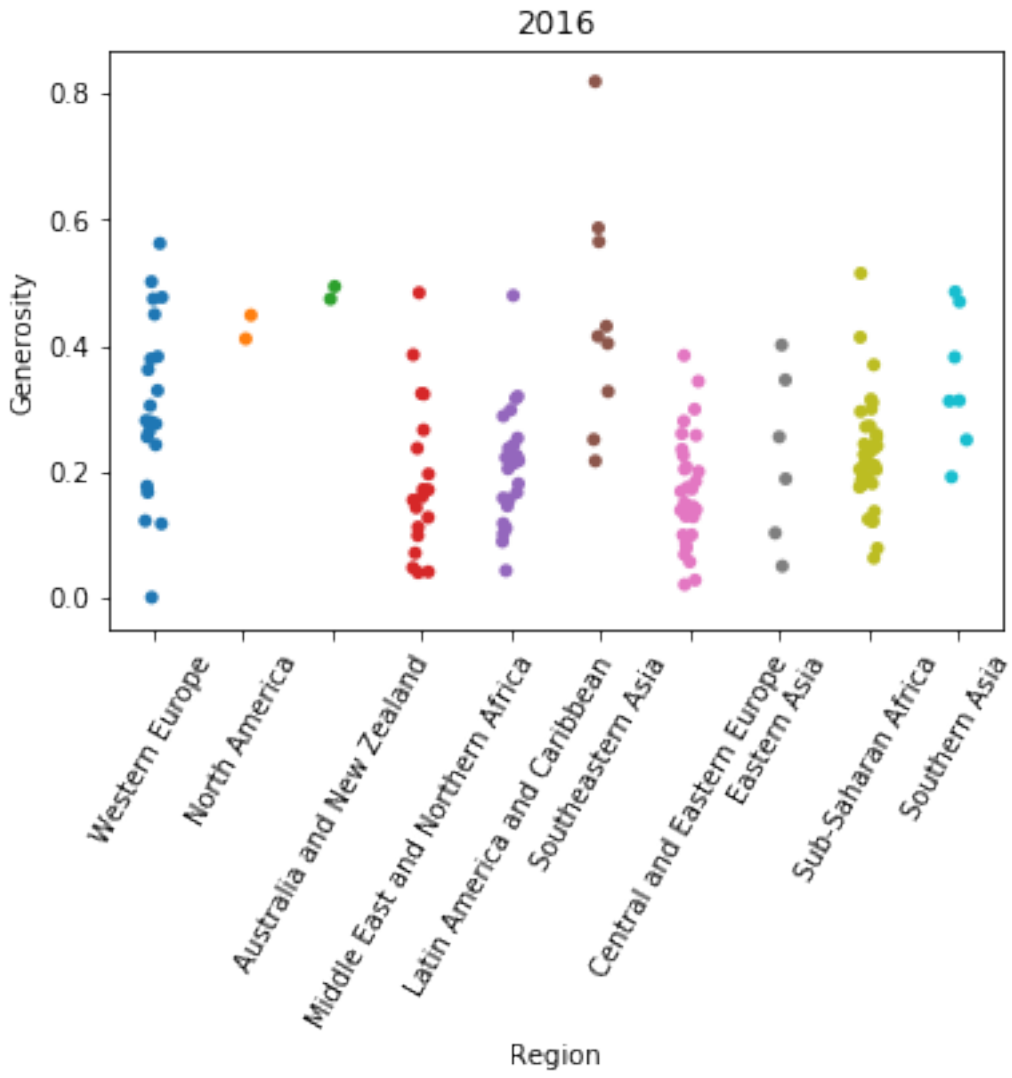
[197]:
```
sns.stripplot(x= "Region",y="Freedom", data=df16).set_title(2016)
plt.xticks(rotation = 60)
```

[197]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]), <a list of 10 Text xticklabel objects>)



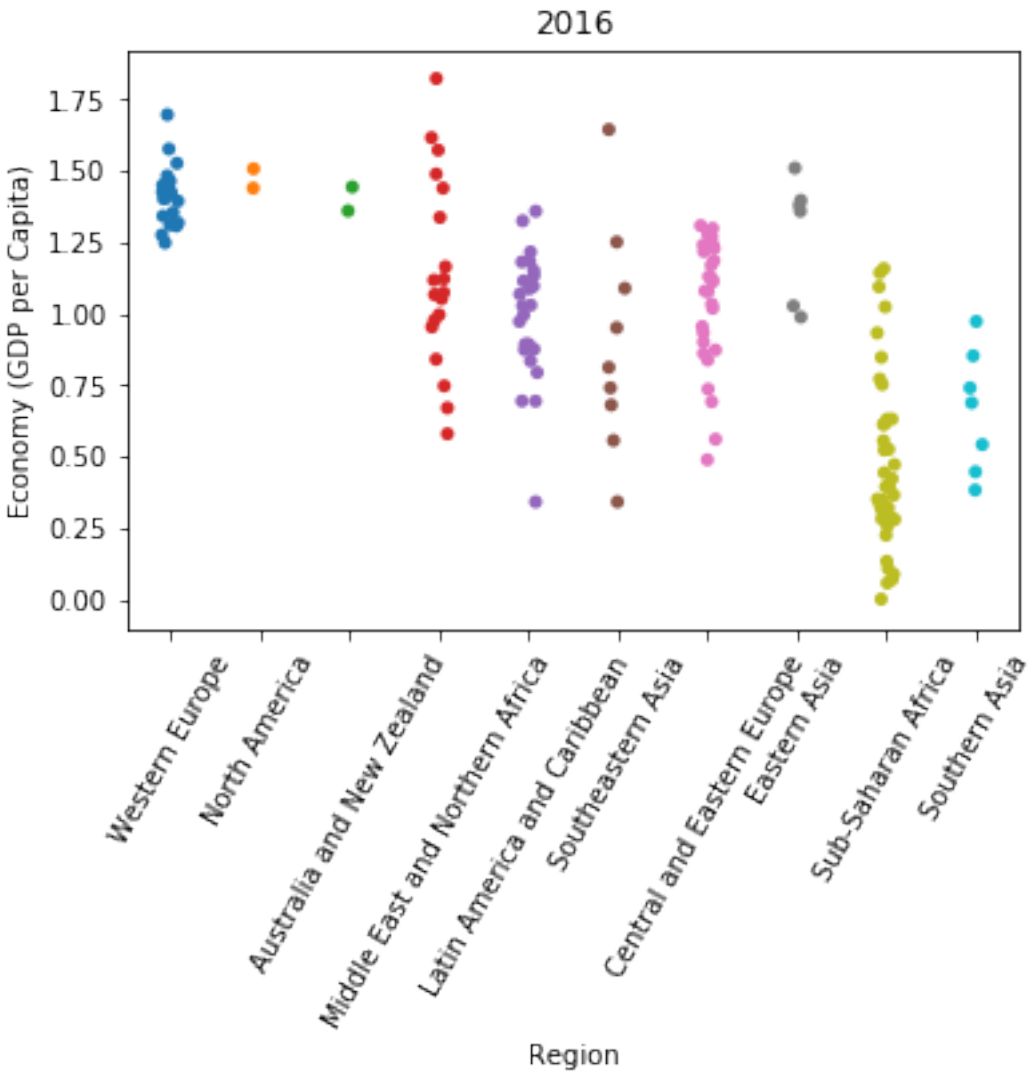
[198]:
```
sns.stripplot(x= "Region",y="Generosity", data=df16).set_title(2016)
plt.xticks(rotation = 60)
```

[198]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]), <a list of 10 Text xticklabel objects>)



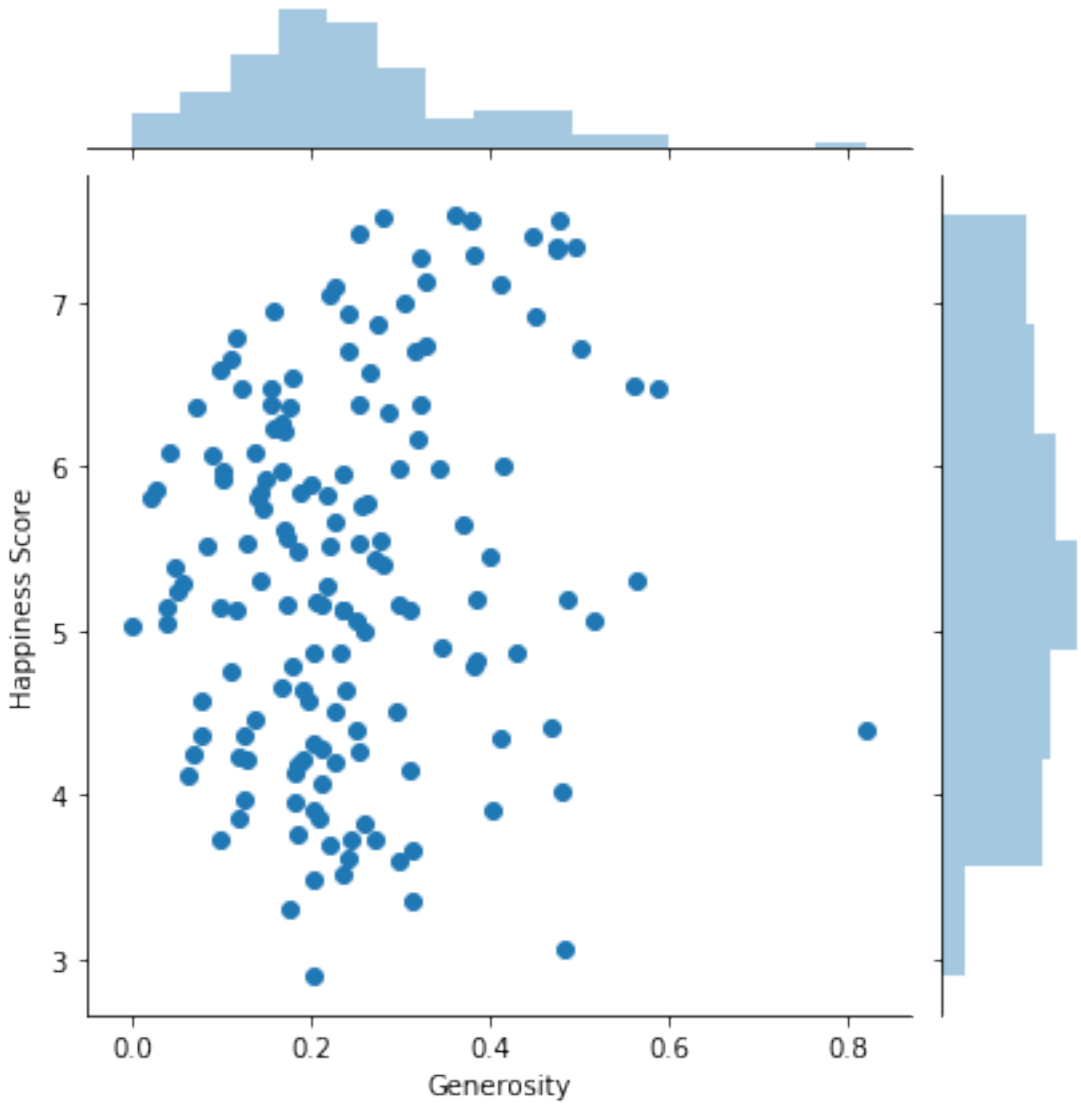
[201]:
```
sns.stripplot(x= "Region",y="Economy (GDP per Capita)", data=df16).
 ↪set_title(2016)
plt.xticks(rotation = 60)
```

[201]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]), <a list of 10 Text xticklabel objects>)



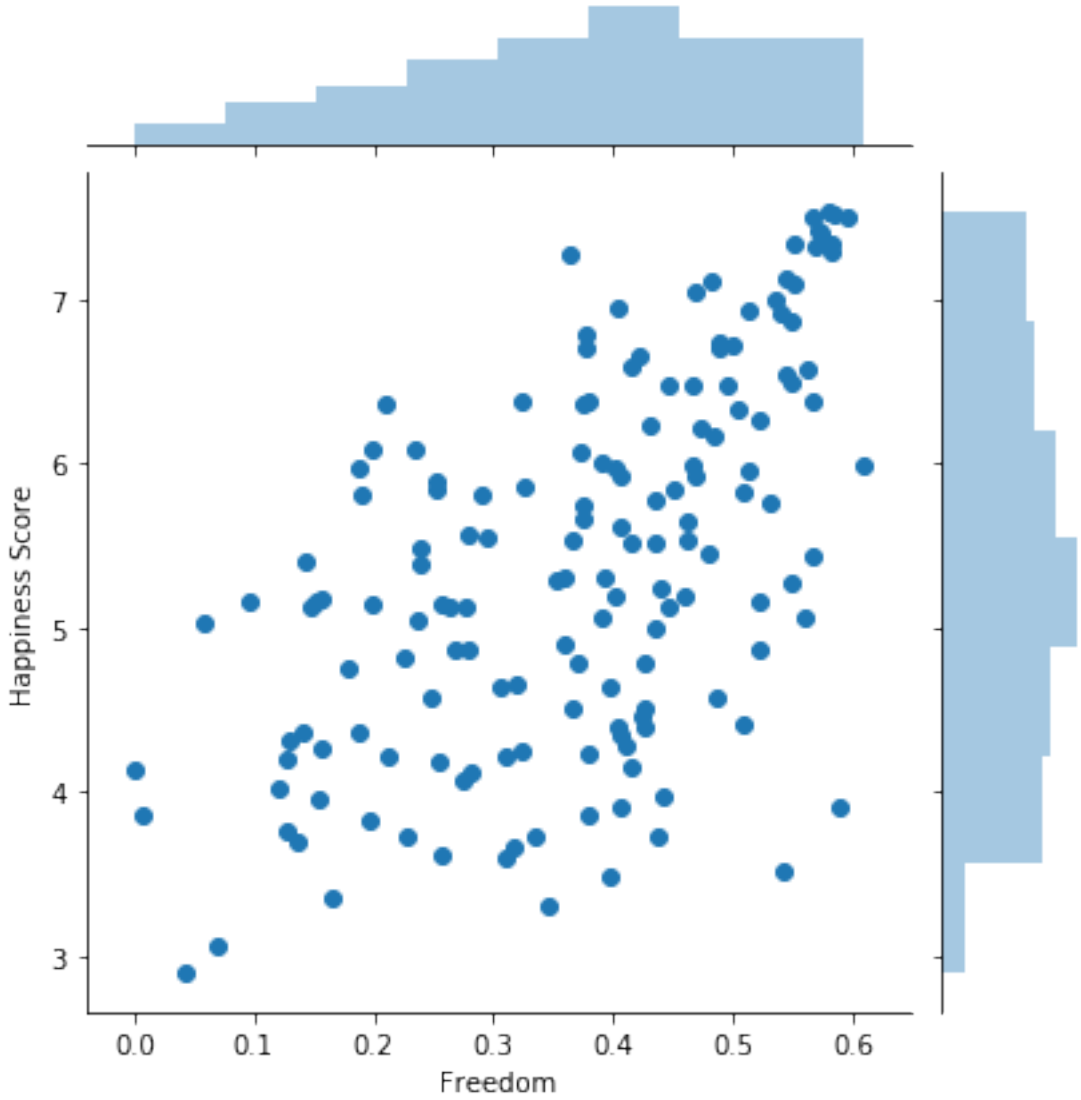
[202]:
```
sns.jointplot(x="Generosity", y="Happiness Score", data=df16)
```

[202]:
```
<seaborn.axisgrid.JointGrid at 0x1a29a04d90>
```



[203]:
```
sns.jointplot(x="Freedom", y="Happiness Score", data=df16)
```

[203]:
```
<seaborn.axisgrid.JointGrid at 0x1a2c683e50>
```

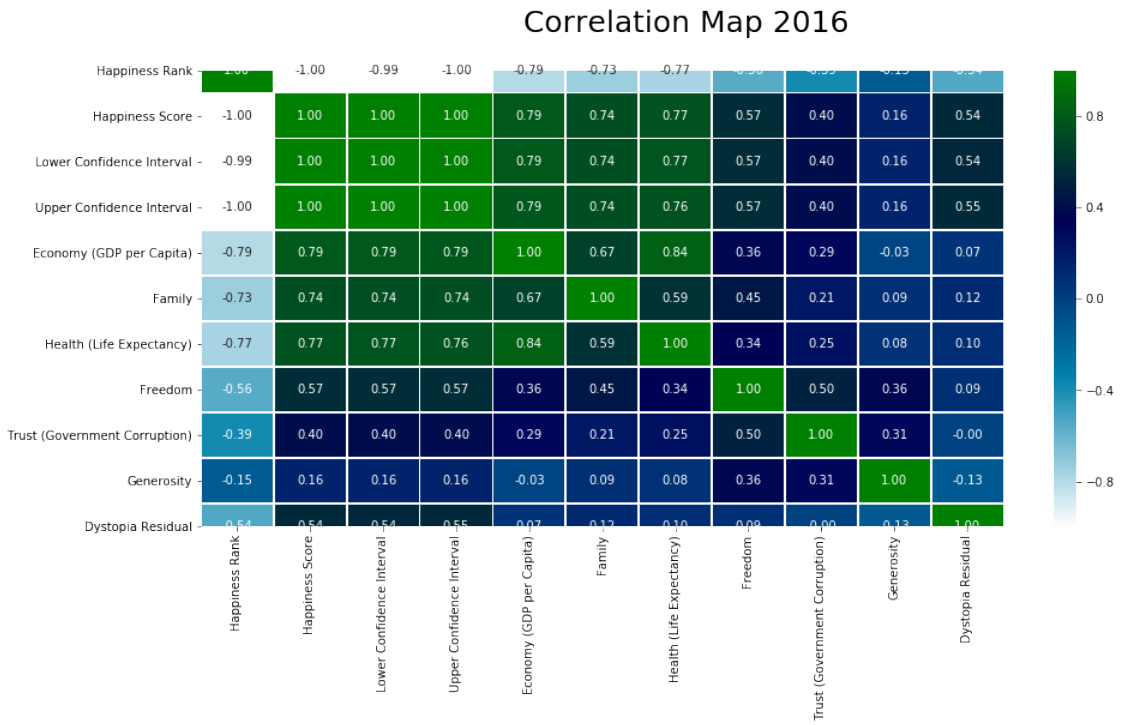


1.3.3 Correlation & Correlation Test

1.3.4 Pearsons Correlation Test

[177]:
```
f, ax = plt.subplots(figsize=(15, 7))
sns.heatmap(df16.corr(), annot=True, linewidths=.7, cmap="ocean_r", fmt=".2f",␣
↪ax=ax)  # seaborn has very simple solution for heatmap

plt.suptitle("Correlation Map 2016", fontsize = 25)

plt.show()   # whitest(-) and greenest(+) = most correlated

# Calculate Pearson (supports scatterplots visuals)
```

```
#Happines Score: Highest correlation with Economy, Health Life Expectancy,␣
 ↪Family,
#follwed by freedom.
# Lowest correlation to score: Happiness score, Generosity, Governmmet
```



1.3.5 Kendall Correlation Test

[434]:
```
df16.corr(method="kendall")
# measures strenght of dependence between 2 variables
```

[434]:

	Happiness Rank	Happiness Score	\
Happiness Rank	1.000000	-1.000000	
Happiness Score	-1.000000	1.000000	
Lower Confidence Interval	-0.987013	0.987013	
Upper Confidence Interval	-0.984073	0.984073	
Economy (GDP per Capita)	-0.611622	0.611622	
Family	-0.568827	0.568827	
Health (Life Expectancy)	-0.568932	0.568932	
Freedom	-0.405162	0.405162	
Trust (Government Corruption)	-0.206959	0.206959	
Generosity	-0.097105	0.097105	
Dystopia Residual	-0.392421	0.392421	

	Lower Confidence Interval
Happiness Rank	-0.987013
Happiness Score	0.987013
Lower Confidence Interval	1.000000
Upper Confidence Interval	0.970840
Economy (GDP per Capita)	0.612438
Family	0.569644
Health (Life Expectancy)	0.573995
Freedom	0.401731
Trust (Government Corruption)	0.206632
Generosity	0.093838
Dystopia Residual	0.390134

	Upper Confidence Interval
Happiness Rank	-0.984073
Happiness Score	0.984073
Lower Confidence Interval	0.970840
Upper Confidence Interval	1.000000
Economy (GDP per Capita)	0.609171
Family	0.566540
Health (Life Expectancy)	0.565175
Freedom	0.407285
Trust (Government Corruption)	0.206468
Generosity	0.094491
Dystopia Residual	0.397158

	Economy (GDP per Capita)	Family
Happiness Rank	-0.611622	-0.568827
Happiness Score	0.611622	0.568827
Lower Confidence Interval	0.612438	0.569644
Upper Confidence Interval	0.609171	0.566540
Economy (GDP per Capita)	1.000000	0.502205
Family	0.502205	1.000000
Health (Life Expectancy)	0.661386	0.438283
Freedom	0.278295	0.362241
Trust (Government Corruption)	0.143481	0.118656
Generosity	-0.011759	0.077413
Dystopia Residual	0.088029	0.122326

	Health (Life Expectancy)	Freedom
Happiness Rank	-0.568932	-0.405162
Happiness Score	0.568932	0.405162
Lower Confidence Interval	0.573995	0.401731
Upper Confidence Interval	0.565175	0.407285
Economy (GDP per Capita)	0.661386	0.278295
Family	0.438283	0.362241

```
Health (Life Expectancy)                        1.000000  0.239843
Freedom                                         0.239843  1.000000
Trust (Government Corruption)                   0.109106  0.325916
Generosity                                      0.046956  0.273559
Dystopia Residual                               0.098567  0.067614

                                Trust (Government Corruption)  Generosity  \
Happiness Rank                                      -0.206959   -0.097105
Happiness Score                                      0.206959    0.097105
Lower Confidence Interval                            0.206632    0.093838
Upper Confidence Interval                            0.206468    0.094491
Economy (GDP per Capita)                             0.143481   -0.011759
Family                                               0.118656    0.077413
Health (Life Expectancy)                             0.109106    0.046956
Freedom                                              0.325916    0.273559
Trust (Government Corruption)                        1.000000    0.170267
Generosity                                           0.170267    1.000000
Dystopia Residual                                    0.040260   -0.031357

                                Dystopia Residual
Happiness Rank                          -0.392421
Happiness Score                          0.392421
Lower Confidence Interval                0.390134
Upper Confidence Interval                0.397158
Economy (GDP per Capita)                 0.088029
Family                                   0.122326
Health (Life Expectancy)                 0.098567
Freedom                                  0.067614
Trust (Government Corruption)            0.040260
Generosity                              -0.031357
Dystopia Residual                        1.000000
```

[437]:
```
df16.cov()
```

[437]:
```
                                Happiness Rank  Happiness Score  \
Happiness Rank                     2067.159889       -51.686413
Happiness Score                     -51.686413         1.303418
Lower Confidence Interval           -51.932217         1.310015
Upper Confidence Interval           -51.440609         1.296822
Economy (GDP per Capita)            -14.886773         0.372281
Family                               -8.891747         0.225095
Health (Life Expectancy)             -8.008299         0.200410
Freedom                              -3.686014         0.094162
Trust (Government Corruption)        -1.954264         0.050965
Generosity                           -0.884037         0.023951
Dystopia Residual                   -13.376885         0.336594
```

	Lower Confidence Interval
Happiness Rank	-51.932217
Happiness Score	1.310015
Lower Confidence Interval	1.318002
Upper Confidence Interval	1.302027
Economy (GDP per Capita)	0.374524
Family	0.227047
Health (Life Expectancy)	0.202526
Freedom	0.094692
Trust (Government Corruption)	0.051607
Generosity	0.024266
Dystopia Residual	0.335394

	Upper Confidence Interval
Happiness Rank	-51.440609
Happiness Score	1.296822
Lower Confidence Interval	1.302027
Upper Confidence Interval	1.291617
Economy (GDP per Capita)	0.370037
Family	0.223144
Health (Life Expectancy)	0.198293
Freedom	0.093631
Trust (Government Corruption)	0.050323
Generosity	0.023637
Dystopia Residual	0.337795

	Economy (GDP per Capita)	Family
Happiness Rank	-14.886773	-8.891747
Happiness Score	0.372281	0.225095
Lower Confidence Interval	0.374524	0.227047
Upper Confidence Interval	0.370037	0.223144
Economy (GDP per Capita)	0.170235	0.073677
Family	0.073677	0.071132
Health (Life Expectancy)	0.079210	0.035990
Freedom	0.021750	0.017471
Trust (Government Corruption)	0.013478	0.006324
Generosity	-0.001409	0.003197
Dystopia Residual	0.015352	0.017306

	Health (Life Expectancy)	Freedom
Happiness Rank	-8.008299	-3.686014
Happiness Score	0.200410	0.094162
Lower Confidence Interval	0.202526	0.094692
Upper Confidence Interval	0.198293	0.093631
Economy (GDP per Capita)	0.079210	0.021750
Family	0.035990	0.017471
Health (Life Expectancy)	0.052601	0.011386

```
Freedom                                        0.011386  0.021172
Trust (Government Corruption)                  0.006356  0.008112
Generosity                                     0.002331  0.007041
Dystopia Residual                              0.012542  0.007229

                               Trust (Government Corruption)  Generosity  \
Happiness Rank                                     -1.954264   -0.884037
Happiness Score                                     0.050965    0.023951
Lower Confidence Interval                           0.051607    0.024266
Upper Confidence Interval                           0.050323    0.023637
Economy (GDP per Capita)                            0.013478   -0.001409
Family                                              0.006324    0.003197
Health (Life Expectancy)                            0.006356    0.002331
Freedom                                             0.008112    0.007041
Trust (Government Corruption)                       0.012329    0.004544
Generosity                                          0.004544    0.017891
Dystopia Residual                                  -0.000175   -0.009646

                               Dystopia Residual
Happiness Rank                       -13.376885
Happiness Score                        0.336594
Lower Confidence Interval              0.335394
Upper Confidence Interval              0.337795
Economy (GDP per Capita)               0.015352
Family                                 0.017306
Health (Life Expectancy)               0.012542
Freedom                                0.007229
Trust (Government Corruption)         -0.000175
Generosity                            -0.009646
Dystopia Residual                      0.294003
```

1.4 Results

[ ]:
```
250 - 500 words

Statistical/Hypothetical Question:

I hypothesize that regions with greater freedom from the
government and with generous citizens will tend to have a higher
happiness score than the other regions.

Outcome of your EDA
What do you feel was missed during the analysis?
Were there any variables you felt could have helped in the analysis?
Were there any assumptions made you felt were incorrect?
What challenges did you face, what did you not fully understand?
```

```
Submit a link to your repository to the assignment link during the final week␣
↪of class.
```

[ ]:

[ ]:

1.4.1 Sources Cited

[ ]: