Data Bootcamp Final Project : Looking At Happiness across the world

Author: William John

Email: wj552@nyu.edu

It is hard to quantify Happiness, or rather, it is hard to quantify reasons for happiness.

"Money cannot buy happiness" is a phrase repeated all over social media. However, locigally speaking, having money allows citizens of a country to purchase, food, water, and shelter, which would not necessarily make someone more happy, but it would help in ensuring that one is not unhappy. This project uses thee World Happiness Report, released annualy to analyze if welath begets happiness.

Data Report

This data is pulled from the World Happiness Report (http://worldhappiness.report/) and also from the World Bank (http://www.worldbank.org/). The World Happiness report is a survey of the happiness of countries. For the purposes of this project, I will be using the report of WHR 2017, released at the United Nations event celebrating the International Day of Happiness on March 20th. which looks at 155 countries and ranks them according to happiness.
Happiness scores are based on the answers to the main life evaluation question asked in the Gallup World Poll, which asks participants to rate their state of life from a scale of 0 to 10 (0 being the worst possible state of life, and 10 being the best).

All of the data is formatted as an excel file, which pandas can directly read with read_excel. The excel file from the World Happiness Report has 4 sheets but I only need to use Data behind Table 2.1 WHR 2017 and Figure2.2 WHR 2017.

- Data behind Table 2.1 WHR 2017 is the excel sheet that has Life evauations, Log GDP, Social support, Healthy Life Expectancy, Freedom to make Life Choices, GINI, etc on countries from the year 2008 to 2015.
- Figure2.2 WHR 2017 is a simple excel sheet that has the Happiness score of each countries evaluated by the World Happiness Report 2017.

Packages Used

In [351]:
```
import pandas as pd             # data package
import matplotlib.pyplot as plt # graphics
import plotly.graph_objs as go #  i use plotly to make nice looking graphs and
 such
import plotly.plotly as py
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
import gzip                     # Library for reading g zip files.
import requests, io             # internet and input tools
import zipfile as zf            # zip file tools
import shutil                   # file management tools
import os                       # operating system tools (check files)
import numpy as np
```

Data Collection

(Failed zip file grabbing)

Below I try to pull data from kaggle, but it was reading the file as a gzip, and there were a lot of complicaions afterwards, because the type was a gzip, but gzip methods could not be called on the object.

In [149]:
```
url = "https://www.kaggle.com/unsdsn/world-happiness/downloads/world-happiness
-report.zip"

r = requests.get(url)

# describe response
print('Response status code:', r.status_code)
print('Response type:', type(r))
print('Response .content:', type(r.content))
print('Response headers:\n', r.headers, sep='')
```

```
Response status code: 200
Response type: <class 'requests.models.Response'>
Response .content: <class 'bytes'>
Response headers:
{'Cache-Control': 'private', 'Content-Length': '17800', 'Content-Type': 'tex
t/html; charset=utf-8', 'Content-Encoding': 'gzip', 'Vary': 'Accept-Encodin
g', 'Set-Cookie': '__RequestVerificationToken=mqz-FbdmjJyQH_bWAosSv9a0LyglUDc
nmnVjc-pN7K0nTRBFYTj6boBVRz2_zMseZAOBtTbcv0_NYGkQDbPGR4K6j6Q1; path=/; secur
e; HttpOnly, TempData=; expires=Mon, 20-Nov-2017 23:51:09 GMT; path=/; secur
e; HttpOnly', 'X-Frame-Options': 'SAMEORIGIN', 'Date': 'Wed, 20 Dec 2017 23:5
1:08 GMT'}
```

In [150]:
```
# convert bytes to zip file
mlz = gzip.open(io.BytesIO(r.content))
print('Type of zipfile object:', type(mlz))
```

```
Type of zipfile object: <class 'gzip.GzipFile'>
```

2017 data from World Happiness Report website (4th Excel Sheet)

I have url to the data and I pull in the data from the excel file online and I only read into my data dataframe the sheet that is named "Figure2.2 WHR 2017".

In [151]:
```
location ="http://worldhappiness.report/wp-content/uploads/sites/2/2017/03/online-data-chapter-2-whr-2017.xlsx"

data = pd.read_excel(location,sheet_name = "Figure2.2 WHR 2017")

data.dtypes
```

Out[151]:
```
Country                                       object
Happiness score                              float64
Whisker-high                                 float64
Whisker-low                                  float64
Explained by: GDP per capita                 float64
Explained by: Social support                 float64
Explained by: Healthy life expectancy        float64
Explained by: Freedom to make life choices   float64
Explained by: Generosity                     float64
Explained by: Perceptions of corruption      float64
Dystopia (1.85) + residual                   float64
dtype: object
```

In [152]:
```
data.shape
```

Out[152]:
```
(155, 11)
```

This is quite a small data set, because only 155 countries are looked at by the WHR

Just for simplicity sake, I want to make sure every column is a string and also show the first 10 rows of the data.

In [153]:
```
for col in data.columns:
    data[col] = data[col].astype(str)
data.head(10)
```

Out[153]:

	Country	Happiness score	Whisker-high	Whisker-low	Explained by: GDP per capita	Explainec Social sup
0	Norway	7.53700017929	7.59444482058	7.479555538	1.61646318436	1.5335235
1	Denmark	7.52199983597	7.58172806486	7.46227160707	1.48238301277	1.5511215
2	Iceland	7.50400018692	7.62203047305	7.38596990079	1.4806330204	1.6105740
3	Switzerland	7.49399995804	7.56177242041	7.42622749567	1.56497955322	1.5169117
4	Finland	7.46899986267	7.52754207581	7.41045764953	1.44357192516	1.5402467
5	Netherlands	7.37699985504	7.42742584124	7.32657386884	1.50394463539	1.4289392
6	Canada	7.31599998474	7.38440283537	7.24759713411	1.47920441628	1.4813489
7	New Zealand	7.3140001297	7.37951044187	7.24848981753	1.40570604801	1.5481951
8	Sweden	7.28399991989	7.34409487739	7.22390496239	1.49438726902	1.4781621
9	Australia	7.28399991989	7.35665122494	7.21134861484	1.48441493511	1.5100419

here I notice that the countries are ordered from highest happiness score, to lowest so finding the rank is just assignning the index +1 to a new column called "Happiness Rank".

In [439]:
```
data = data.set_index(data.index +1)

data["Happiness Rank"] = data.index

data.head(10)
```

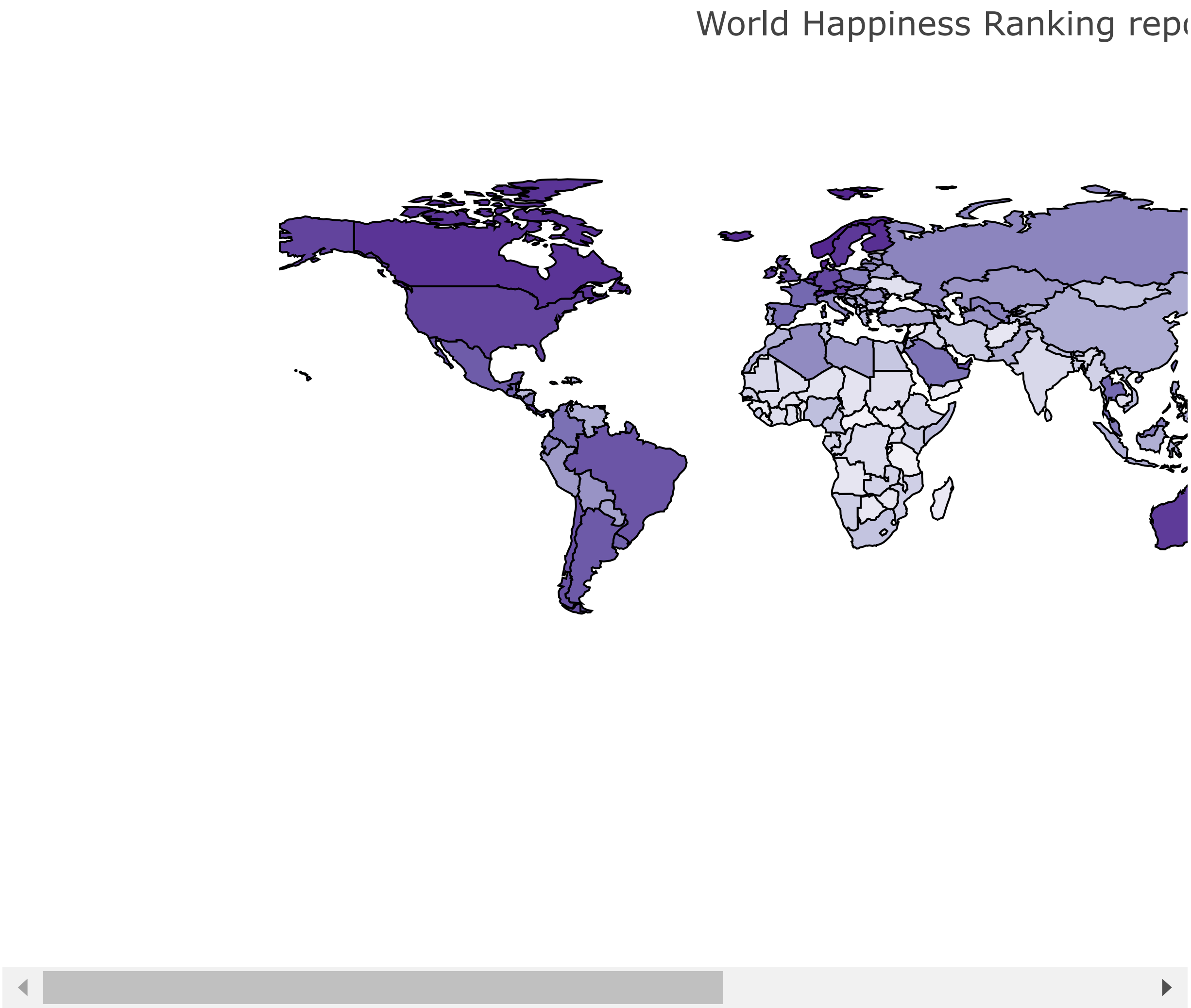
Out[439]:

	Country	Happiness score	Whisker-high	Whisker-low	Explained by: GDP per capita	Explaine Social su
2	Norway	7.53700017929	7.59444482058	7.479555538	1.61646318436	1.533523
3	Denmark	7.52199983597	7.58172806486	7.46227160707	1.48238301277	1.551121
4	Iceland	7.50400018692	7.62203047305	7.38596990079	1.4806330204	1.610574
5	Switzerland	7.49399995804	7.56177242041	7.42622749567	1.56497955322	1.516911
6	Finland	7.46899986267	7.52754207581	7.41045764953	1.44357192516	1.540246
7	Netherlands	7.37699985504	7.42742584124	7.32657386884	1.50394463539	1.428939
8	Canada	7.31599998474	7.38440283537	7.24759713411	1.47920441628	1.481348
9	New Zealand	7.3140001297	7.37951044187	7.24848981753	1.40570604801	1.548195
10	Sweden	7.28399991989	7.34409487739	7.22390496239	1.49438726902	1.478162
11	Australia	7.28399991989	7.35665122494	7.21134861484	1.48441493511	1.510041

I didnt really get any useful information about gdp, because the World Happiness Report's explained by columns are not very clear and they do not do much graphically with the data. Therefore I used the plotly sites help to create a chloropleth map that graphs all the rankings, and happiness scores of the countries in the dataframe. https://plot.ly/python/choropleth-maps/#new-to-plotly (https://plot.ly/python/choropleth-maps/#new-to-plotly)

Looking at the Data with chloropleth

In [441]:
```
scl = [[0.0, 'rgb(242,240,247)'],[0.2, 'rgb(218,218,235)'],[0.4, 'rgb(188,189,
220)'],
            [0.6, 'rgb(158,154,200)'],[0.8, 'rgb(117,107,177)'],[1.0, 'rgb(84,
39,143)']]
# I defined a scale for the gradient to look like.  This uses perentages of th
e whole scale to partition different slightly
#colors into a gradient

#define what I want to map
vis = dict(type = "choropleth",
            colorscale = scl, #use my previously defined scale to define the g
radient
            autocolorscale = False,
            locations = data["Country"], #this is how I label the locations
            locationmode = "country names",# define the map
            z = data["Happiness Rank"], #this is how the map will get colored
 with the scale i made
            text = data["Happiness score"],#when howevering over a country, th
is is what shows on the however
            reversescale = True, # i wanted the lighter countries to be the le
ss happy, which needs to flip the default scale setup
            colorbar = dict(
            title = "Happiness Rank<br>(Lighter means less happy)") # title of
 map
          )
#define my layout
layout = dict(
        title = "World Happiness Ranking reports", #Title of the map
        geo = dict(
        showframe = False,#show frame no and show coastlines no
        showcoastlines = False,
        projection = dict(
            type = 'Mercator'
        )
        )
)
#after formatting everything, actually print the map out to console
choromap = go.Figure(data = [vis], layout=layout)
iplot(choromap)
```



The darker, happier countries are more prevalent in the west and in eastern europe/australia. This is aligns with richer countires being happier because they are usually more wealthy nations. However, I need to pull more data to actually quantitavely explore this

2017 data from World Happiness Report website (1st Excel Sheet)

In [287]:
```
location ="http://worldhappiness.report/wp-content/uploads/sites/2/2017/03/onl
ine-data-chapter-2-whr-2017.xlsx"

data2 = pd.read_excel(location,sheet_name = "Data behind Table 2.1 WHR 2017")
data2.columns
```

Out[287]:
```
Index(['WP5 Country', 'country', 'year', 'Life Ladder', 'Log GDP per capita',
       'Social support', 'Healthy life expectancy at birth',
       'Freedom to make life choices', 'Generosity',
       'Perceptions of corruption', 'Positive affect', 'Negative affect',
       'Confidence in national government', 'Democratic Quality',
       'Delivery Quality', 'Standard deviation of ladder by country-year',
       'Standard deviation/Mean of ladder by country-year',
       'GINI index (World Bank estimate)',
       'GINI index (World Bank estimate), average 2000-13',
       'gini of household income reported in Gallup, by wp5-year',
       'Most people can be trusted, Gallup',
       'Most people can be trusted, WVS round 1981-1984',
       'Most people can be trusted, WVS round 1989-1993',
       'Most people can be trusted, WVS round 1994-1998',
       'Most people can be trusted, WVS round 1999-2004',
       'Most people can be trusted, WVS round 2005-2009',
       'Most people can be trusted, WVS round 2010-2014'],
      dtype='object')
```

- Life Ladder is what the people rated how happy they were from a scale of 0 to 10 using the Cantril Ladder.
- Social support is the national average of the responses (yes or no, 1 or 0) of the question, “If you were in trouble, do you have relatives or friends you can count on to help you whenever you need them, or not?” asked in the Callup World Poll.
- Freedom to make life choices is the average of the responses to the Gallup World Poll question, "Are you satisfied or disatisfied with your freedom to choose what you do with your life?"(yes or no, 1 or 0)
- Perceptions of corruption are the average of binary answers to the questions, “Is corruption widespread throughout the government or not?” and “Is corruption widespread within businesses or not?”.

In [288]:
```
data2.shape
```

Out[288]:
```
(1420, 27)
```

In [289]:
```
data2.head(10)
```

Out[289]:

	WP5 Country	country	year	Life Ladder	Log GDP per capita	Social support	Healthy life expectancy at birth	Freedom to make life choices	Ge
0	Afghanistan	Afghanistan	2008	3.723590	7.197130	0.450662	47.550438	0.718114	0.1
1	Afghanistan	Afghanistan	2009	4.401778	7.362664	0.552308	47.859673	0.678896	0.2
2	Afghanistan	Afghanistan	2010	4.758381	7.416260	0.539075	48.159512	0.600127	0.1
3	Afghanistan	Afghanistan	2011	3.831719	7.445761	0.521104	48.451160	0.495901	0.1
4	Afghanistan	Afghanistan	2012	3.782938	7.549241	0.520637	48.738346	0.530935	0.2
5	Afghanistan	Afghanistan	2013	3.572100	7.536999	0.483552	49.023087	0.577955	0.0
6	Afghanistan	Afghanistan	2014	3.130896	7.519704	0.525568	49.305813	0.508514	0.1
7	Afghanistan	Afghanistan	2015	3.982855	7.506759	0.528597	49.588539	0.388928	0.0
8	Afghanistan	Afghanistan	2016	4.220169	7.497288	0.559072	49.871265	0.522566	0.0
9	Albania	Albania	2007	4.634252	8.984322	0.821372	67.169853	0.528605	-0.

10 rows × 27 columns

This excel sheet is marketly bigger as it has more years of data and more columns. With this sheet, I can average the Life Ladder, Log GDP per capita, Social Support, Health life expectancy, Confidence in Government, over the years and graph those values to see how they affect happiness

In [290]:
```
grouped = data2.groupby("country")

AverageLifeLadder = grouped['Life Ladder'].agg(["count","mean"])

AverageLifeLadder["Number of Years"]=AverageLifeLadder["count"]

AverageLifeLadder["Life Ladder Average"]=AverageLifeLadder["mean"]

del AverageLifeLadder["count"]
del AverageLifeLadder["mean"]

AverageLifeLadder.head()
```

Out[290]:

	Number of Years	Life Ladder Average
country		
Afghanistan	9	3.933825
Albania	9	5.027596
Algeria	5	5.625685
Angola	4	4.420299
Argentina	11	6.439476

Calculate the average, and then concatenate them into one data frame for easy graphing

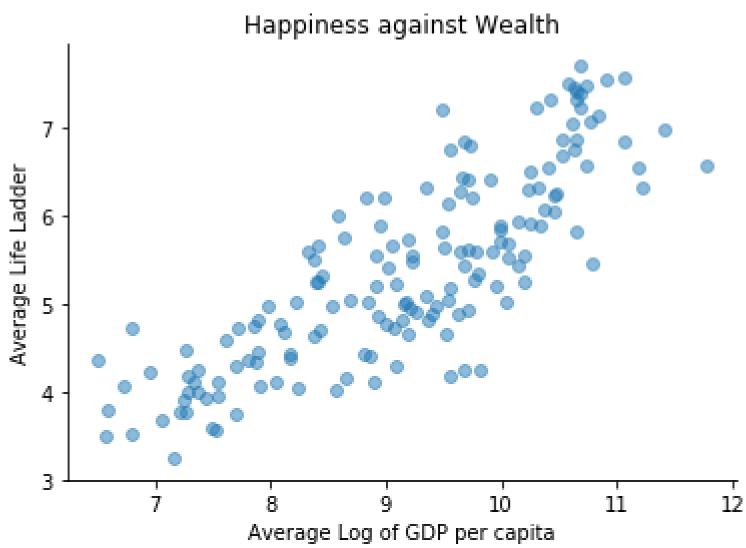
In [291]:
```
AverageGDP = grouped['Log GDP per capita'].agg(["mean"])
AverageGDP["Average Log GDP per capita"]=AverageGDP["mean"]
del AverageGDP["mean"]

AverageSocialSupp = grouped['Social support'].agg(["mean"])
AverageSocialSupp["Average Social Support"]=AverageSocialSupp["mean"]
del AverageSocialSupp["mean"]

AverageLife = grouped['Healthy life expectancy at birth'].agg(["mean"])
AverageLife["Average Life Expectancy"]=AverageLife["mean"]
del AverageLife["mean"]

AverageConf = grouped['Perceptions of corruption'].agg(["mean"])
AverageConf["Average Perceptions of Corruption"]=AverageConf["mean"]
del AverageConf["mean"]

bigdata = pd.concat([AverageLifeLadder, AverageGDP,AverageSocialSupp,AverageLi
fe,AverageConf], axis = 1)
bigdata.head()
```

Out[291]:

	Number of Years	Life Ladder Average	Average Log GDP per capita	Average Social Support	Average Life Expectancy	Average Perceptions of Corruption
country						
Afghanistan	9	3.933825	7.447978	0.520064	48.727537	0.812616
Albania	9	5.027596	9.172559	0.732704	67.999568	0.857864
Algeria	5	5.625685	9.498361	0.803993	64.117065	0.648712
Angola	4	4.420299	8.816016	0.737973	44.572942	0.867018
Argentina	11	6.439476	9.658498	0.906024	66.701826	0.844310

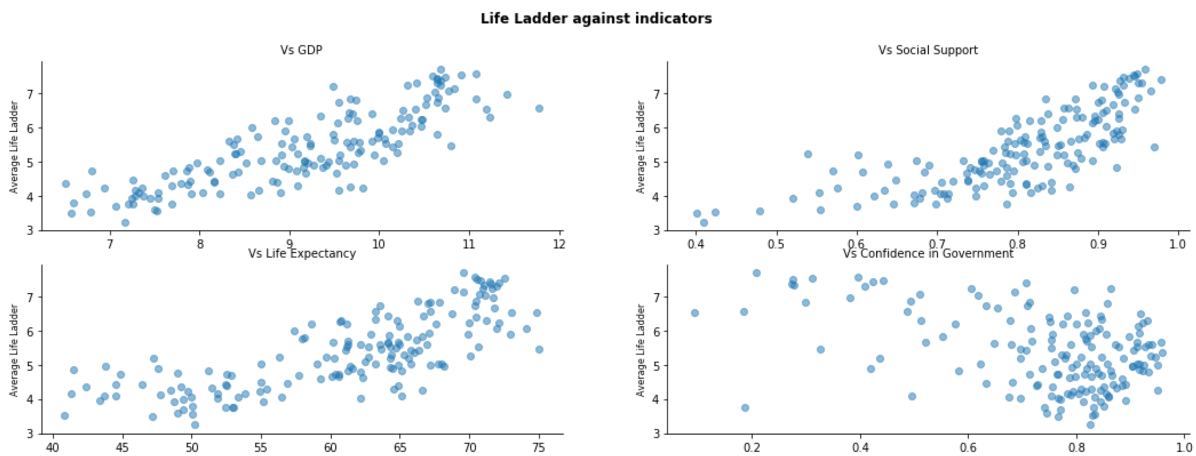
Plot the average Life Ladder results and the Average Log GDP per capita

In [292]:
```
fig, ax = plt.subplots()

ax.scatter(bigdata["Average Log GDP per capita"], bigdata["Life Ladder Averag
e"],
            alpha= 0.50)
ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)

ax.set_title("Happiness against Wealth")
ax.set_ylabel("Average Life Ladder")
ax.set_xlabel("Average Log of GDP per capita")


plt.show()
```



From this graph it would seem that a higher GDP is correlated to a higher Happiness "Life Ladder" to see the other indicators and their plots against the Life Ladder, I plot them side by side below.

In [293]:
```
fig, ax = plt.subplots(nrows = 2, ncols = 2, sharex = False, figsize = (18,6))

ax = ax.ravel()

var_list = ["Average Log GDP per capita","Average Social Support", "Average Li
fe Expectancy", "Average Perceptions of Corruption"]
nice_name = ["Vs GDP", "Vs Social Support", "Vs Life Expectancy", "Vs Confiden
ce in Government"]

count = 0

for x in ax:

    x.scatter(bigdata[var_list[count]],bigdata["Life Ladder Average"], alpha=
0.50)

    x.set_title(nice_name[count], fontsize = 10)

    x.spines["right"].set_visible(False)

    x.spines["top"].set_visible(False)

    x.set_ylabel("Average Life Ladder", fontsize = 8)

    count = count + 1

fig.suptitle("Life Ladder against indicators", fontsize = 12, fontweight = "bo
ld") # big titel


plt.show()
```



These graphs indicate that the life expectancy, GDP, and Social Support are directly correlated to the average Life Ladder reported by people of a specific country between 2008 and 2017.

I need a sanity check to see the correlations.

In [294]:
```
corr_mat = bigdata.corr()
corr_mat
```

Out[294]:

	Number of Years	Life Ladder Average	Average Log GDP per capita	Average Social Support	Average Life Expectancy	Average Perceptions of Corruption
Number of Years	1.000000	0.150697	0.123899	0.201111	0.312062	0.069416
Life Ladder Average	0.150697	1.000000	0.824259	0.759044	0.762103	-0.438427
Average Log GDP per capita	0.123899	0.824259	1.000000	0.713051	0.829720	-0.382046
Average Social Support	0.201111	0.759044	0.713051	1.000000	0.627476	-0.217053
Average Life Expectancy	0.312062	0.762103	0.829720	0.627476	1.000000	-0.298834
Average Perceptions of Corruption	0.069416	-0.438427	-0.382046	-0.217053	-0.298834	1.000000

The correlation of Average Log GDP per capita is indeed quite high and the highest out of all the other indicators.

Maybe finding the average is not accurate enough, we need more information. For example, people in an area can be "sad", as in they are more inclined to say that they are sad in surveys. Also, it is hard to quantify how happy you are if you are not sure of the boundaries, everyone has different base cases and levels throughout the day.

Look at Percentage change over years

I want to read in the data again and also clean it up a bit so I can look at only the things I want.

In [391]:
```
location ="http://worldhappiness.report/wp-content/uploads/sites/2/2017/03/online-data-chapter-2-whr-2017.xlsx"

data2 = pd.read_excel(location,sheet_name = "Data behind Table 2.1 WHR 2017")
data2.columns

data2 = data2.set_index("year")
del data2["country"]
s = data2.iloc[:,0:4]
s.head(10)
```

Out[391]:

	WP5 Country	Life Ladder	Log GDP per capita	Social support
year				
2008	Afghanistan	3.723590	7.197130	0.450662
2009	Afghanistan	4.401778	7.362664	0.552308
2010	Afghanistan	4.758381	7.416260	0.539075
2011	Afghanistan	3.831719	7.445761	0.521104
2012	Afghanistan	3.782938	7.549241	0.520637
2013	Afghanistan	3.572100	7.536999	0.483552
2014	Afghanistan	3.130896	7.519704	0.525568
2015	Afghanistan	3.982855	7.506759	0.528597
2016	Afghanistan	4.220169	7.497288	0.559072
2007	Albania	4.634252	8.984322	0.821372

In [392]:
```
s["Life Ladder Growth"] = s["Life Ladder"].pct_change()
s["Log GDP per capita Growth"] = s["Log GDP per capita"].pct_change()

s
```

Out[392]:

	WP5 Country	Life Ladder	Log GDP per capita	Social support	Life Ladder Growth	Log GDP per capita Growth
year						
2008	Afghanistan	3.723590	7.197130	0.450662	NaN	NaN
2009	Afghanistan	4.401778	7.362664	0.552308	0.182133	0.023000
2010	Afghanistan	4.758381	7.416260	0.539075	0.081013	0.007279
2011	Afghanistan	3.831719	7.445761	0.521104	-0.194743	0.003978
2012	Afghanistan	3.782938	7.549241	0.520637	-0.012731	0.013898
2013	Afghanistan	3.572100	7.536999	0.483552	-0.055734	-0.001622
2014	Afghanistan	3.130896	7.519704	0.525568	-0.123514	-0.002295
2015	Afghanistan	3.982855	7.506759	0.528597	0.272114	-0.001721
2016	Afghanistan	4.220169	7.497288	0.559072	0.059584	-0.001262
2007	Albania	4.634252	8.984322	0.821372	0.098120	0.198343
2009	Albania	5.485470	9.104285	0.833047	0.183680	0.013353
2010	Albania	5.268937	9.145679	0.733152	-0.039474	0.004547
2011	Albania	5.867422	9.173692	0.759434	0.113587	0.003063
2012	Albania	5.510124	9.189353	0.784502	-0.060895	0.001707
2013	Albania	4.550648	9.201632	0.759477	-0.174130	0.001336
2014	Albania	4.813763	9.222470	0.625587	0.057819	0.002265
2015	Albania	4.606651	9.249300	0.639356	-0.043025	0.002909
2016	Albania	4.511101	9.282300	0.638411	-0.020742	0.003568
2010	Algeria	5.463567	9.464850	NaN	0.211138	0.019667
2011	Algeria	5.317194	9.474717	0.810234	-0.026791	0.001042
2012	Algeria	5.604596	9.488671	0.839397	0.054051	0.001473
2014	Algeria	6.354898	9.514430	0.818189	0.133873	0.002715
2016	Algeria	5.388171	9.549138	0.748150	-0.152123	0.003648
2011	Angola	5.589001	8.783317	0.723094	0.037272	-0.080198
2012	Angola	4.360250	8.800270	0.752593	-0.219852	0.001930
2013	Angola	3.937107	8.833123	0.721591	-0.097046	0.003733
2014	Angola	3.794838	8.847354	0.754615	-0.036135	0.001611
2006	Argentina	6.312925	9.520737	0.938463	0.663556	0.076111
2007	Argentina	6.073158	9.594051	0.862206	-0.037980	0.007701
2008	Argentina	5.961034	9.650693	0.892195	-0.018462	0.005904

	WP5 Country	Life Ladder	Log GDP per capita	Social support	Life Ladder Growth	Log GDP per capita Growth
year						
...	...	...	...	...	...	...
2007	Yemen	4.477133	8.354671	0.824969	-0.115587	-0.038822
2009	Yemen	4.809259	8.378407	0.756430	0.074183	0.002841
2010	Yemen	4.350313	8.383639	0.726612	-0.095430	0.000624
2011	Yemen	3.746256	8.193192	0.662680	-0.138854	-0.022717
2012	Yemen	4.060601	8.191209	0.681678	0.083909	-0.000242
2013	Yemen	4.217679	8.206121	0.693905	0.038683	0.001821
2014	Yemen	3.967958	8.286581	0.638252	-0.059208	0.009805
2015	Yemen	2.982674	7.843260	0.668683	-0.248310	-0.053499
2016	Yemen	3.825631	NaN	0.775407	0.282618	NaN
2006	Zambia	4.824455	7.865655	0.797665	0.261087	0.002855
2007	Zambia	3.998293	7.917436	0.687989	-0.171245	0.006583
2008	Zambia	4.730263	7.963219	0.624418	0.183071	0.005783
2009	Zambia	5.260361	8.021870	0.781926	0.112065	0.007365
2011	Zambia	4.999114	8.114495	0.864023	-0.049663	0.011547
2012	Zambia	5.013375	8.157069	0.780023	0.002853	0.005247
2013	Zambia	5.243996	8.176512	0.761312	0.046001	0.002384
2014	Zambia	4.345837	8.194842	0.706223	-0.171274	0.002242
2015	Zambia	4.843164	8.195832	0.691483	0.114438	0.000121
2016	Zambia	4.347544	8.198633	0.767047	-0.102334	0.000342
2006	Zimbabwe	3.826268	7.364544	0.821656	-0.119901	-0.101735
2007	Zimbabwe	3.280247	7.314472	0.828113	-0.142703	-0.006799
2008	Zimbabwe	3.174264	7.105295	0.843475	-0.032309	-0.028598
2009	Zimbabwe	4.055914	7.146843	0.805781	0.277750	0.005848
2010	Zimbabwe	4.681570	7.236320	0.856638	0.154257	0.012520
2011	Zimbabwe	4.845642	7.328846	0.864694	0.035046	0.012786
2012	Zimbabwe	4.955101	7.407775	0.896476	0.022589	0.010770
2013	Zimbabwe	4.690188	7.429061	0.799274	-0.053463	0.002873
2014	Zimbabwe	4.184451	7.443748	0.765839	-0.107829	0.001977
2015	Zimbabwe	3.703191	7.431285	0.735800	-0.115011	-0.001674
2016	Zimbabwe	3.735400	7.422308	0.768425	0.008698	-0.001208

1420 rows × 6 columns

Mistakes were made because pandas has no idea that there were different countries in the rows and it calculated percentage changes between countires through a leap back in time(2016 Afghanistan to 2007 Albania).I can try to iterate over the dataset to do percentages for each country.

Below I go through the list of countries and then I add the percentage change of each country's GDP to a series, after that I add that to the dataframe we had before named s

In [399]:
```
listOfCountries=s["WP5 Country"].unique()

a = s[s['WP5 Country'] == "Afghanistan"]["Log GDP per capita"].pct_change()

for i in listOfCountries:
    if i == "Afghanistan":
        continue
    else:
        a = pd.concat([a,s[s['WP5 Country'] == i]["Log GDP per capita"].pct_ch
ange()], axis=0)
s['Log GDP per capita Growth'] = pd.Series(a, index=s.index)

b = s[s['WP5 Country'] == "Afghanistan"]["Life Ladder"].pct_change()

for i in listOfCountries:
    if i == "Afghanistan":
        continue
    else:
        b = pd.concat([b,s[s['WP5 Country'] == i]["Life Ladder"].pct_change
()], axis=0)
s['Life Ladder Growth'] = pd.Series(b, index=s.index)
s
```

Out[399]:

	WP5 Country	Life Ladder	Log GDP per capita	Social support	Life Ladder Growth	Log GDP per capita Growth
year						
2008	Afghanistan	3.723590	7.197130	0.450662	NaN	NaN
2009	Afghanistan	4.401778	7.362664	0.552308	0.182133	0.023000
2010	Afghanistan	4.758381	7.416260	0.539075	0.081013	0.007279
2011	Afghanistan	3.831719	7.445761	0.521104	-0.194743	0.003978
2012	Afghanistan	3.782938	7.549241	0.520637	-0.012731	0.013898
2013	Afghanistan	3.572100	7.536999	0.483552	-0.055734	-0.001622
2014	Afghanistan	3.130896	7.519704	0.525568	-0.123514	-0.002295
2015	Afghanistan	3.982855	7.506759	0.528597	0.272114	-0.001721
2016	Afghanistan	4.220169	7.497288	0.559072	0.059584	-0.001262
2007	Albania	4.634252	8.984322	0.821372	NaN	NaN
2009	Albania	5.485470	9.104285	0.833047	0.183680	0.013353
2010	Albania	5.268937	9.145679	0.733152	-0.039474	0.004547
2011	Albania	5.867422	9.173692	0.759434	0.113587	0.003063
2012	Albania	5.510124	9.189353	0.784502	-0.060895	0.001707
2013	Albania	4.550648	9.201632	0.759477	-0.174130	0.001336
2014	Albania	4.813763	9.222470	0.625587	0.057819	0.002265
2015	Albania	4.606651	9.249300	0.639356	-0.043025	0.002909
2016	Albania	4.511101	9.282300	0.638411	-0.020742	0.003568
2010	Algeria	5.463567	9.464850	NaN	NaN	NaN
2011	Algeria	5.317194	9.474717	0.810234	-0.026791	0.001042
2012	Algeria	5.604596	9.488671	0.839397	0.054051	0.001473
2014	Algeria	6.354898	9.514430	0.818189	0.133873	0.002715
2016	Algeria	5.388171	9.549138	0.748150	-0.152123	0.003648
2011	Angola	5.589001	8.783317	0.723094	NaN	NaN
2012	Angola	4.360250	8.800270	0.752593	-0.219852	0.001930
2013	Angola	3.937107	8.833123	0.721591	-0.097046	0.003733
2014	Angola	3.794838	8.847354	0.754615	-0.036135	0.001611
2006	Argentina	6.312925	9.520737	0.938463	NaN	NaN
2007	Argentina	6.073158	9.594051	0.862206	-0.037980	0.007701
2008	Argentina	5.961034	9.650693	0.892195	-0.018462	0.005904

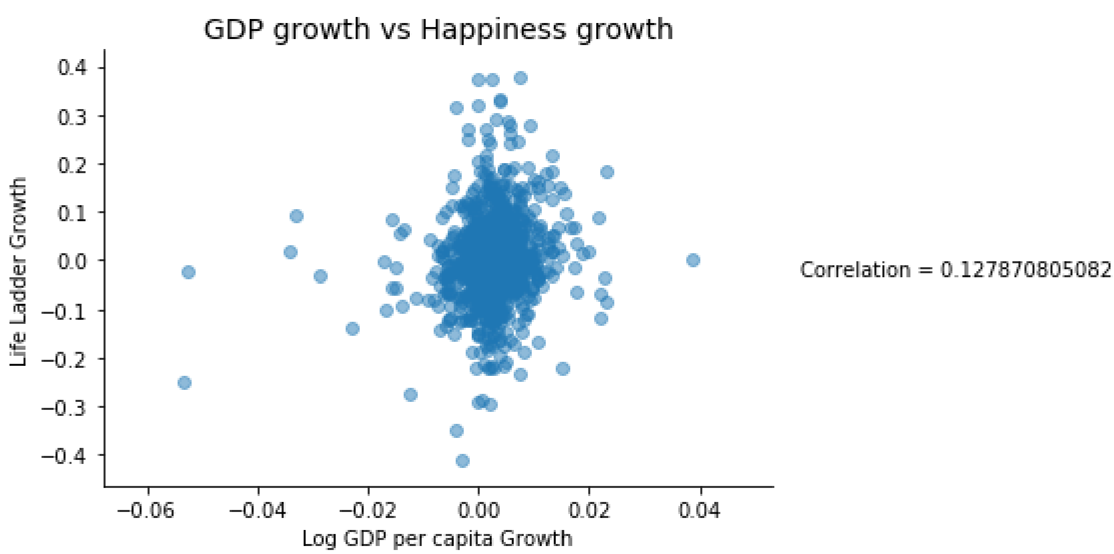
	WP5 Country	Life Ladder	Log GDP per capita	Social support	Life Ladder Growth	Log GDP per capita Growth
year						
...	...	...	...	...	...	...
2007	Yemen	4.477133	8.354671	0.824969	NaN	NaN
2009	Yemen	4.809259	8.378407	0.756430	0.074183	0.002841
2010	Yemen	4.350313	8.383639	0.726612	-0.095430	0.000624
2011	Yemen	3.746256	8.193192	0.662680	-0.138854	-0.022717
2012	Yemen	4.060601	8.191209	0.681678	0.083909	-0.000242
2013	Yemen	4.217679	8.206121	0.693905	0.038683	0.001821
2014	Yemen	3.967958	8.286581	0.638252	-0.059208	0.009805
2015	Yemen	2.982674	7.843260	0.668683	-0.248310	-0.053499
2016	Yemen	3.825631	NaN	0.775407	0.282618	NaN
2006	Zambia	4.824455	7.865655	0.797665	NaN	NaN
2007	Zambia	3.998293	7.917436	0.687989	-0.171245	0.006583
2008	Zambia	4.730263	7.963219	0.624418	0.183071	0.005783
2009	Zambia	5.260361	8.021870	0.781926	0.112065	0.007365
2011	Zambia	4.999114	8.114495	0.864023	-0.049663	0.011547
2012	Zambia	5.013375	8.157069	0.780023	0.002853	0.005247
2013	Zambia	5.243996	8.176512	0.761312	0.046001	0.002384
2014	Zambia	4.345837	8.194842	0.706223	-0.171274	0.002242
2015	Zambia	4.843164	8.195832	0.691483	0.114438	0.000121
2016	Zambia	4.347544	8.198633	0.767047	-0.102334	0.000342
2006	Zimbabwe	3.826268	7.364544	0.821656	NaN	NaN
2007	Zimbabwe	3.280247	7.314472	0.828113	-0.142703	-0.006799
2008	Zimbabwe	3.174264	7.105295	0.843475	-0.032309	-0.028598
2009	Zimbabwe	4.055914	7.146843	0.805781	0.277750	0.005848
2010	Zimbabwe	4.681570	7.236320	0.856638	0.154257	0.012520
2011	Zimbabwe	4.845642	7.328846	0.864694	0.035046	0.012786
2012	Zimbabwe	4.955101	7.407775	0.896476	0.022589	0.010770
2013	Zimbabwe	4.690188	7.429061	0.799274	-0.053463	0.002873
2014	Zimbabwe	4.184451	7.443748	0.765839	-0.107829	0.001977
2015	Zimbabwe	3.703191	7.431285	0.735800	-0.115011	-0.001674
2016	Zimbabwe	3.735400	7.422308	0.768425	0.008698	-0.001208

1420 rows × 6 columns

It took a substantial amount of time but now we have the percentage change that also accounts for the countries changing throughout the data set.(you can see the NaN value when a new country starts)

Lets graph it now!

In [428]:
```
fig, ax = plt.subplots()

ax.scatter(s["Log GDP per capita Growth"], s["Life Ladder Growth"],
            alpha= 0.50)

corr_mat = s.corr()
cr = corr_mat["Life Ladder Growth"]["Log GDP per capita Growth"]

message = "Correlation = " + str(cr)
ax.text(.115,-.03, message, horizontalalignment='right')
ax.set_title('GDP growth vs Happiness growth',  fontsize=14,)


ax.set_xlabel("Log GDP per capita Growth")
ax.set_ylabel("Life Ladder Growth")

ax.spines["right"].set_visible(False)

ax.spines["top"].set_visible(False)

plt.show()
```



In our Graph here we do not see much correlation between growth of GDP and growth of Happiness. This means that a change in GDP whether positive or negative will not affect the overall happiness of a person. The graphs below showed that more GDP meant more Happiness, and this shows that changing GDP does not really affect change in Happiness. This leads me to believe in a few theories; that GDP is related to previously accumulated wealth, accumulated steadily, so, already having a higher GDP or already having wealth contributes more to happiness than quickly acquiring it. It also makes me believe that living in a richer country will make you more happy then living in a rapidly developing country.

Conclusion

In Conclusion, a higher GDP generally correlates with a higher happiness score and ranking

In the graph of Log GDP per capita vs Life Ladder, there was a clear association between a high GDP and higher Happines score. The correlation calculated was 0.824259 which was rather high.
However, Social Support and Life Expectancy also had high, but not as high correlations to GDP. Furthermore, Social support and Life Expectany are more factors that are effected by and affect GDP.
This does not detract from the fact that more rich countries based on GDP per capita are more happy.