Minh Ta

Dr. Stonedahl

DATA-360

```
In [ ]: import pandas as pd
```

Today I am going to analyze countries' happiness score and how it correlates with multiple other factors (e.g. GDP, Birth/death rate,...)

First, the 2017 happiness data set (which can be found here: https://www.kaggle.com/unsdsn/world-happiness (https://www.kaggle.com/unsdsn/world-happiness)) ranks the happiness of countries based on the data from the Gallup World Poll. We will merge this dataset with the world countries information dataset (here: https://www.kaggle.com/fernandol/countries-of-the-world (https://www.kaggle.com/fernandol/countries-of-the-world)), which originated from the CIA's website, to gain more insights into countries presented in this happiness dataset.

Let's see what the datasets look like:

```
In [106]: happiness2017 = pd.read_csv('happiness/2017.csv')
          happiness2017[0:5]
```

Out[106]:

	Country	Happiness.Rank	Happiness.Score	Whisker.high	Whisker.low	Economy..GDP.per.Capita.	Famil
0	Norway	1	7.537	7.594445	7.479556	1.616463	1.53352
1	Denmark	2	7.522	7.581728	7.462272	1.482383	1.55112
2	Iceland	3	7.504	7.622030	7.385970	1.480633	1.61057
3	Switzerland	4	7.494	7.561772	7.426227	1.564980	1.51691
4	Finland	5	7.469	7.527542	7.410458	1.443572	1.54024

```
In [107]: countries = pd.read_csv('countries of the world.csv')
          countries[0:5]
```

Out[107]:

	Country	Region	Population	Area (sq. mi.)	Pop. Density (per sq. mi.)	Coastline (coast/area ratio)	Net migration	Infant mortality (per 1000 births)	GDP ($ per capita)	Literacy (%)
0	Afghanistan	ASIA (EX. NEAR EAST)	31056997	647500	48,0	0,00	23,06	163,07	700.0	36,0
1	Albania	EASTERN EUROPE	3581655	28748	124,6	1,26	-4,93	21,52	4500.0	86,5
2	Algeria	NORTHERN AFRICA	32930091	2381740	13,8	0,04	-0,39	31	6000.0	70,0
3	American Samoa	OCEANIA	57794	199	290,4	58,29	-20,71	9,27	8000.0	97,0
4	Andorra	WESTERN EUROPE	71201	468	152,1	0,00	6,6	4,05	19000.0	100,0

Before our analysis, we will have to clean the data and merge them together, based on `Country`. Notice that I had to `strip()` both datasets so that

In [108]:

```
happiness2017["Country"] = happiness2017["Country"].str.strip()
countries["Country"] = countries["Country"].str.strip()
mergedDat = happiness2017.merge(countries, on="Country")
mergedDat[0:5]
```

Out[108]:

	Country	Happiness.Rank	Happiness.Score	Whisker.high	Whisker.low	Economy..GDP.per.Capita.	Famil
0	Norway	1	7.537	7.594445	7.479556	1.616463	1.53352
1	Denmark	2	7.522	7.581728	7.462272	1.482383	1.55112
2	Iceland	3	7.504	7.622030	7.385970	1.480633	1.61057
3	Switzerland	4	7.494	7.561772	7.426227	1.564980	1.51691
4	Finland	5	7.469	7.527542	7.410458	1.443572	1.54024

5 rows × 31 columns

We will extract some columns of data for our analysis. Here I picked Region. Happiness.Score, GDP, Literacy, Phones, and Net Migration. Also we will drop all the NaN values from the dataset.

In [109]:

```
graphDat = mergedDat[['Region', 'Happiness.Score', 'GDP ($ per capita)', 'Literacy
(%)', 'Phones (per 1000)', 'Net migration']]
graphDat = graphDat.dropna()
```

I figured out that the second dataset has commas as decimal point. So we will have to convert that back to dots.

In [110]:

```
graphDat['GDP ($ per capita)'] = pd.to_numeric(graphDat['GDP ($ per capita)'].asty
pe(str).str.replace(',','.'))
graphDat['Literacy (%)'] = pd.to_numeric(graphDat['Literacy (%)'].astype(str).str.
replace(',','.'))
graphDat['Phones (per 1000)'] = pd.to_numeric(graphDat['Phones (per 1000)'].astype
(str).str.replace(',','.'))
graphDat['Net migration'] = pd.to_numeric(graphDat['Net migration'].astype(str).st
r.replace(',','.'))
```

Here is what `graphDat` dataset looks like so far

In [111]:

```
graphDat[0:5]
```

Out[111]:

	Region	Happiness.Score	GDP ($ per capita)	Literacy (%)	Phones (per 1000)	Net migration
0	WESTERN EUROPE	7.537	37800.0	100.0	461.7	1.74
1	WESTERN EUROPE	7.522	31100.0	100.0	614.6	2.48
2	WESTERN EUROPE	7.504	30900.0	99.9	647.7	2.38
3	WESTERN EUROPE	7.494	32700.0	99.0	680.9	4.05
4	WESTERN EUROPE	7.469	27400.0	100.0	405.3	0.95

VISUALIZING DATA

We are going to import `seaborn` , `matplotlib` , and `numpy` for our visualization

In [112]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
# for graphs to display in the notebook
%matplotlib inline
```

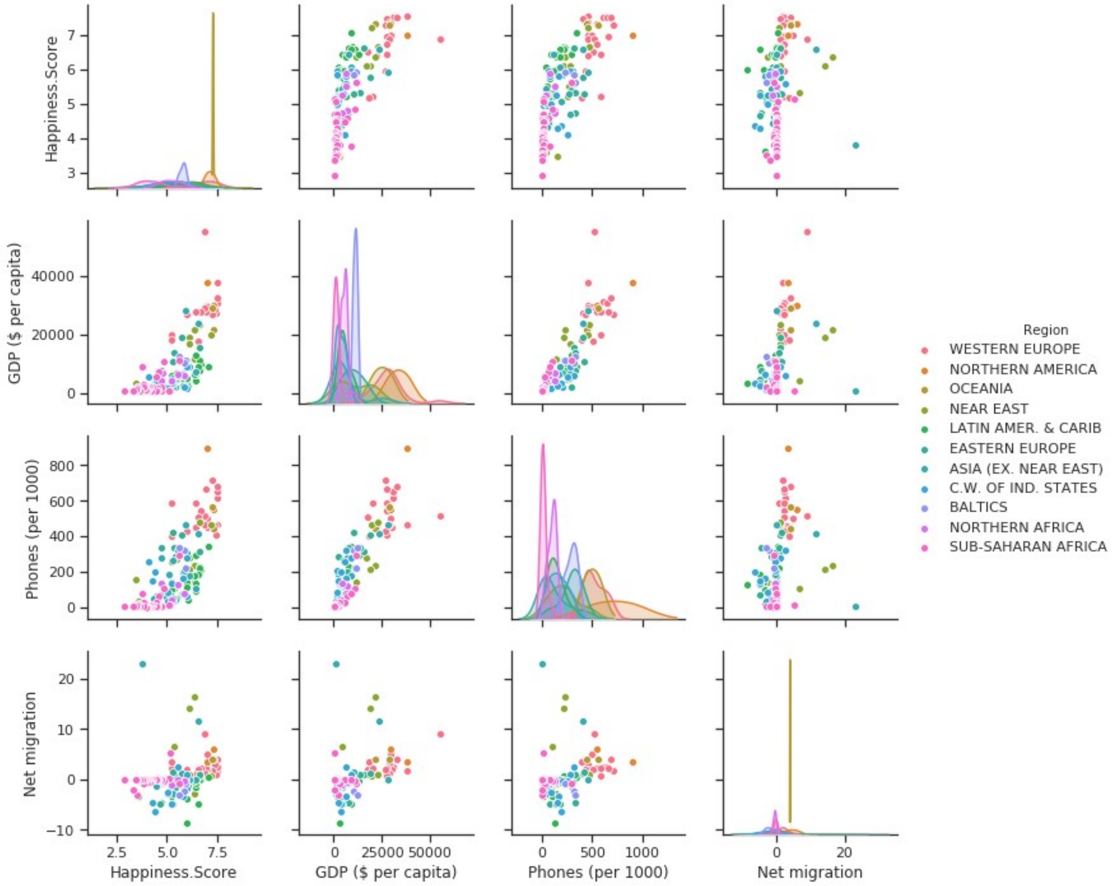
For this, I tried to manipulate Literacy data because it was throwing errors earlier, but I haven't got it done yet. I will try again later. We will skip Literacy for now.

In [113]:
```
graphDat['Literacy (%)'] = graphDat['Literacy (%)'] * 100
```

We will create a pairplot of all the data we have as follow:

In [114]:
```
sns.set(style="ticks")

sns.pairplot(graphDat, hue='Region', vars=['Happiness.Score', 'GDP ($ per capita)'
, 'Phones (per 1000)', 'Net migration']);
```



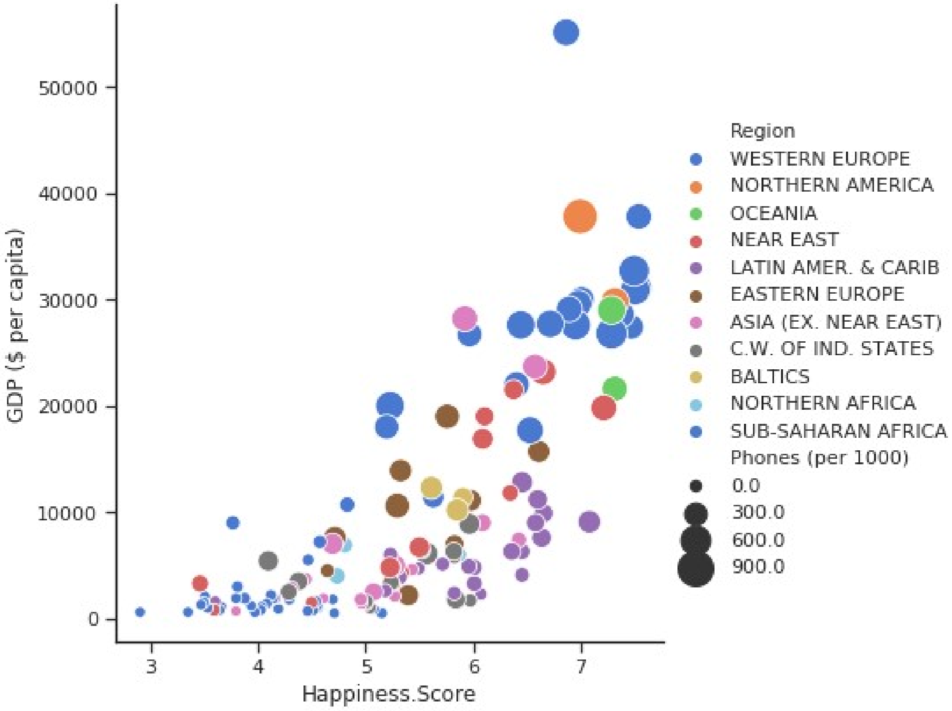
We will focus on the first column of graphs. Here we can see that there is somewhat a correlation between happiness score and GDP and the ammount of Phones per person.

In [115]:

```
sns.relplot(x="Happiness.Score", y="GDP ($ per capita)", hue="Region", size="Phones (per 1000)",
            sizes=(40, 400), alpha=1, palette="muted",
            height=6, data=graphDat)
```

Out[115]:

```
<seaborn.axisgrid.FacetGrid at 0x7f3e5b55d850>
```



Looking closer into Happiness, GDP, and Phones, we also see this log correlation as when GDP and Phones increase, Happiness increases.

In [ ]: