

Geospatial Analysis With A Choropleth Map

Objective of Script

1. Import libraries
2. Import data
3. Dimension Checks
4. Data wrangling
5. Data consistency checks
6. Plotting a choropleth map of WHR_AllYears
7. Exporting Dataframes
8. Exporting Map
9. Discussion of Results

01. Import Libraries

Importing necessary libraries

```
import pandas as pd
import numpy as np
import matplotlib
import seaborn as sns
import os
import folium
import json
```

Verify matplotlib version

```
matplotlib.__version__
'3.5.2'
```

Plot graphs will display in notebook

```
%matplotlib inline
```

02. Import Dataframe

Import dataframes, creating shortcut to data file and verify

```
path = r'/Users/tatumzeliadt/Library/CloudStorage/OneDrive-Personal/Data Analytics/Data Immersion/6  
Advanced Analytics and Dashboard Design/World Happiness Report Jun2020'
```

```
print(path)
```

```
/Users/tatumzeliadt/Library/CloudStorage/OneDrive-Personal/Data Analytics/Dat  
a Immersion/6 Advanced Analytics and Dashboard Design/World Happiness Report  
Jun2020
```

Import merged WHR dataframe

```
WHR_ALLYEARS = pd.read_pickle(os.path.join(path, '02 Data','Prepared Data', 'WHR_ALLYEARS.pkl')) # with  
all columns
```

Import ".json" file for entire globe from <https://geojson-maps.ash.ms>

```
country_geo = r'/Users/tatumzeliadt/Library/CloudStorage/OneDrive-Personal/Data Analytics/Data Immer  
sion/6 Advanced Analytics and Dashboard Design/World Happiness Report Jun2020/02 Data/Original Data/  
custom.geo.json'
```

To view the JSON file contents within Python:

```
o = open(r'/Users/tatumzeliadt/Library/CloudStorage/OneDrive-Personal/Data Analytics/Data Immersion/  
6 Advanced Analytics and Dashboard Design/World Happiness Report Jun2020/02 Data/Original Data/custo  
m.geo.json')
```

returns JSON object as a dictionary

```
data = json.load(o)
```

Iterating through the json list

```
for i in data['features']:  
    print(i)
```

IOPub data rate exceeded.

The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.

To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:

NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)

NotebookApp.rate_limit_window=3.0 (secs)

03. Dimension Check on Imported Dataframe

Checks on WHR 2015-2019

Dimension Check Summary of WHR_ALLYEARS

```
def check_df(WHR_ALLYEARS, head=5):
    print('----- WHR_ALLYEARS -----')
    print('***** SHAPE *****')
    print(WHR_ALLYEARS.shape)
    print(' ')
    print('***** COLUMNS *****')
    print(WHR_ALLYEARS.columns.to_list())
    print(' ')
    print('***** DATA TYPES *****')
    print(WHR_ALLYEARS.dtypes)
    print(' ')
    print('***** FIRST FIVE ROWS *****')
    print(WHR_ALLYEARS.head(head))
    print(' ')
    print('***** LAST FIVE ROWS *****')
    print(WHR_ALLYEARS.tail(head))
```

```
check_df(WHR_ALLYEARS)
```

```
----- WHR_ALLYEARS -----
```

```
***** SHAPE *****
```

```
(777, 11)
```

```
***** COLUMNS *****
```

```
['Year', 'Country', 'Region', 'Happiness Rank', 'Happiness Score', 'Economy',  
'Family', 'Freedom', 'Generosity', 'Health', 'Trust']
```

```
***** DATA TYPES *****
```

```
Year                int64
Country             object
Region              object
Happiness Rank      int64
Happiness Score     float64
Economy              float64
Family              float64
Freedom              float64
Generosity           float64
Health               float64
Trust                float64
dtype: object
```

```
***** FIRST FIVE ROWS *****
```

```
Year      Country      Region  Happiness Rank  Happiness Score  \
```

0	2015	Switzerland	Western Europe	1	7.587
1	2015	Iceland	Western Europe	2	7.561
2	2015	Denmark	Western Europe	3	7.527
3	2015	Norway	Western Europe	4	7.522
4	2015	Canada	North America	5	7.427

	Economy	Family	Freedom	Generosity	Health	Trust
0	1.3965	1.3495	0.6656	0.2968	0.9414	0.4198
1	1.3023	1.4022	0.6288	0.4363	0.9478	0.1415
2	1.3255	1.3606	0.6494	0.3414	0.8746	0.4836
3	1.4590	1.3310	0.6697	0.3470	0.8852	0.3650
4	1.3263	1.3226	0.6330	0.4581	0.9056	0.3296

***** LAST FIVE ROWS *****

	Year	Country	Region	Happiness Rank \
151	2019	Rwanda	Sub-Saharan Africa	152
152	2019	Tanzania	Sub-Saharan Africa	153
153	2019	Afghanistan	Southern Asia	154
154	2019	Central African Republic	Sub-Saharan Africa	155
155	2019	South Sudan	Sub-Saharan Africa	156

	Happiness Score	Economy	Family	Freedom	Generosity	Health	Trust
151	3.334	0.359	0.711	0.555	0.217	0.614	0.411
152	3.231	0.476	0.885	0.417	0.276	0.499	0.147
153	3.203	0.350	0.517	0.000	0.158	0.361	0.025
154	3.083	0.026	0.000	0.225	0.235	0.105	0.035
155	2.853	0.306	0.575	0.010	0.202	0.295	0.091

04. Data Wrangling

Select only the necessary columns for map and put them in a list called 'columns' to further
Columns 'Freedom', 'Generosity', 'Trust' were not included because of only weak to moderate relationship

```
columns = ['Country', 'Happiness Rank', 'Happiness Score', 'Economy', 'Family', 'Health']
```

Create a subset of the dataframe with the limited columns

```
WHR_CORRVARS = WHR_ALLYEARS[columns]
```

Dimension Check of Cleaned WHR_CORRVARS

```
def check_df(WHR_CORRVARS):
    print('----- WHR_CORRVARS Check -----')
    print('***** SHAPE *****')
    print(WHR_CORRVARS.shape)
    print(' ')
    print('***** COLUMNS *****')
```

```

print(WHR_CORRVARs.columns.to_list())
print(' ')
print('***** FIRST FIVE ROWS *****')
print(WHR_CORRVARs.head(5))

check_df(WHR_CORRVARs)
----- WHR_CORRVARs Check -----
***** SHAPE *****
(777, 6)

***** COLUMNS *****
['Country', 'Happiness Rank', 'Happiness Score', 'Economy', 'Family', 'Health']

***** FIRST FIVE ROWS *****
   Country  Happiness Rank  Happiness Score  Economy  Family  Health
0  Switzerland           1           7.587    1.3965  1.3495  0.9414
1    Iceland           2           7.561    1.3023  1.4022  0.9478
2    Denmark           3           7.527    1.3255  1.3606  0.8746
3    Norway            4           7.522    1.4590  1.3310  0.8852
4    Canada            5           7.427    1.3263  1.3226  0.9056

WHR_CORRVARs.dtypes
Country           object
Happiness Rank    int64
Happiness Score   float64
Economy           float64
Family           float64
Health           float64
dtype: object
# Get the type of an object

type(WHR_CORRVARs)
pandas.core.frame.DataFrame

```

05. Data Consistency Checks

Duplicates

Checking for duplicates

```
WHR_CORRVARs.duplicated()
```

No duplicates observed

```
0    False
1    False
```

```
2      False
3      False
4      False
...
151    False
152    False
153    False
154    False
155    False
Length: 777, dtype: bool
```

Missing Values

Check for missing values

```
WHR_CORRVARs.isnull().sum()
```

No missing values observed

```
Country          0
Happiness Rank   0
Happiness Score   0
Economy          0
Family           0
Health           0
dtype: int64
```

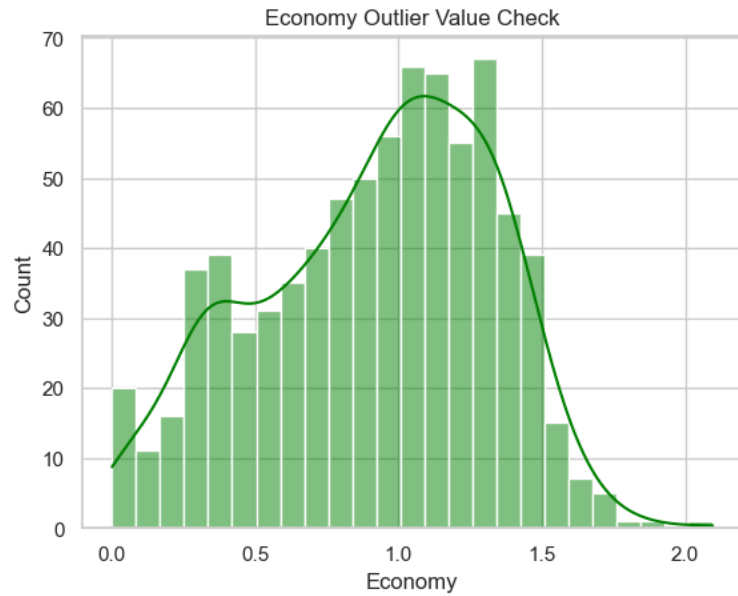
Outlier Value Check by Histogram View

Histograms for Happiness_Score variables to view outlier values

```
sns.histplot(WHR_CORRVARs['Happiness Score'], bins=25, kde = True, color = 'gold').set(title = 'Happiness Score Outlier Value Check')
[Text(0.5, 1.0, 'Happiness Score Outlier Value Check')]
```

Histograms for Economy variables to view outlier values

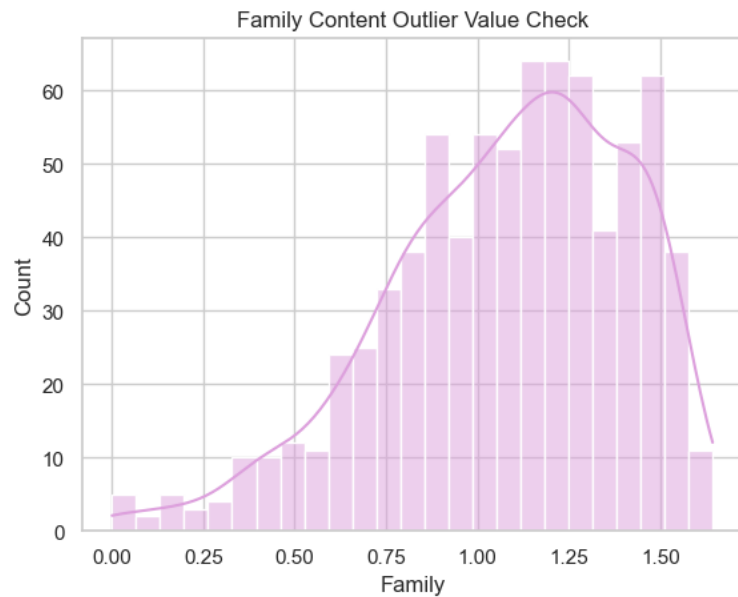
```
sns.histplot(WHR_CORRVARs['Economy'], bins=25, kde = True, color = 'green').set(title = 'Economy Outlier Value Check')
[Text(0.5, 1.0, 'Economy Outlier Value Check')]
```



Histograms for Family variables to view outlier values

```
sns.histplot(WHR_CORRVARs['Family'], bins=25, kde = True, color = 'plum').set(title = 'Family Content Outlier Value Check')
```

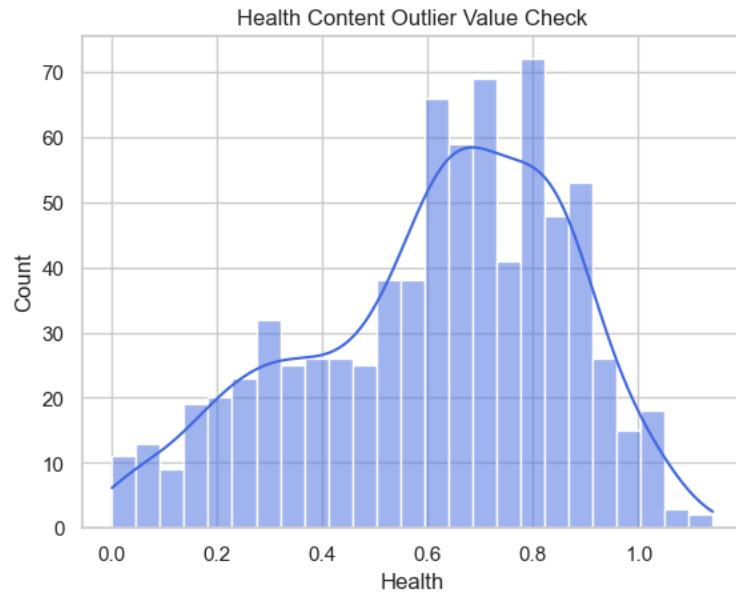
```
[Text(0.5, 1.0, 'Family Content Outlier Value Check')]
```



Histograms for Health variable to view outlier values

```
sns.histplot(WHR_CORRVARs['Health'], bins=25, kde = True, color = 'royalblue').set(title = 'Health Content Outlier Value Check')
```

```
[Text(0.5, 1.0, 'Health Content Outlier Value Check')]
```



No outlier observed in any of the variable histograms above

06. Plotting a Cloropleth Map

Map individual variables

Cloropleth map of Happiness Score

```
plot_hs = WHR_CORRVARS[['Country', 'Happiness Score']]
plot_hs.head(5)
```

	Country	Happiness Score
0	Switzerland	7.587
1	Iceland	7.561
2	Denmark	7.527
3	Norway	7.522
4	Canada	7.427

To create the choropleth map of Happiness Score

```
map = folium.Map(location = [0, 0], zoom_start = 1.25, titles = 'WHR Happiness Score')
```

```
folium.Choropleth(
    geo_data = country_geo,
    data = plot_hs,
    columns = ['Country', 'Happiness Score'],
    key_on = 'feature.properties.name',
    fill_color = 'YlOrRd', fill_opacity=0.6, line_opacity=0.1,
```



```
legend_name = 'Happiness Score').add_to(map)
folium.LayerControl().add_to(map)
```

map

Make this Notebook Trusted to load map: File -> Trust Notebook

Choropleth map of Economy

```
plot_econ = WHR_CORRVAR[['Country', 'Economy']]
plot_econ.head(5)
```

	Country	Economy
0	Switzerland	1.3965
1	Iceland	1.3023
2	Denmark	1.3255
3	Norway	1.4590
4	Canada	1.3263

To create the choropleth map of Economy

```
map = folium.Map(location = [0, 0], zoom_start = 1.25, titles = 'WHR Economy')
```

```
folium.Choropleth(
    geo_data = country_geo,
    data = plot_econ,
    columns = ['Country', 'Economy'],
    key_on = 'feature.properties.name',
    fill_color = 'Greens', fill_opacity=0.6, line_opacity=0.1,
    legend_name = 'Economy').add_to(map)
folium.LayerControl().add_to(map)
```

map

Make this Notebook Trusted to load map: File -> Trust Notebook

Choropleth map of Family

```
plot_fam = WHR_CORRVAR[['Country', 'Family']]
plot_fam.head(5)
```

	Country	Family
0	Switzerland	1.3495
1	Iceland	1.4022
2	Denmark	1.3606

	Country	Family
3	Norway	1.3310
4	Canada	1.3226

To create the choropleth map of Family

```
map = folium.Map(location = [0, 0], zoom_start = 1.25, titles = 'WHR Family')
```

```
folium.Choropleth(
    geo_data = country_geo,
    data = plot_fam,
    columns = ['Country', 'Family'],
    key_on = 'feature.properties.name',
    fill_color = 'Purples', fill_opacity=0.6, line_opacity=0.1,
    legend_name = 'Economy').add_to(map)
folium.LayerControl().add_to(map)
```

map

Make this Notebook Trusted to load map: File -> Trust Notebook

Choropleth map of Health

```
plot_hlth = WHR_CORRVARs[['Country', 'Health']]
plot_hlth.head(5)
```

	Country	Health
0	Switzerland	0.9414
1	Iceland	0.9478
2	Denmark	0.8746
3	Norway	0.8852
4	Canada	0.9056

To create the choropleth map of Health

```
map = folium.Map(location = [0, 0], zoom_start = 1.25, titles = 'WHR Health')
```

```
folium.Choropleth(
    geo_data = country_geo,
    data = plot_hlth,
    columns = ['Country', 'Health'],
    key_on = 'feature.properties.name',
    fill_color = 'Blues', fill_opacity=0.6, line_opacity=0.1,
    legend_name = 'Health').add_to(map)
```

```
folium.LayerControl().add_to(map)
```

map

Make this Notebook Trusted to load map: File -> Trust Notebook

07. Exporting Dataframes

Exporting the data plot dataframes

```
plot_hs.to_csv(os.path.join(path, '02 Data', 'Prepared Data', 'plot_hs.csv'))
plot_econ.to_csv(os.path.join(path, '02 Data', 'Prepared Data', 'plot_econ.csv'))
plot_fam.to_csv(os.path.join(path, '02 Data', 'Prepared Data', 'plot_fam.csv'))
plot_hlth.to_csv(os.path.join(path, '02 Data', 'Prepared Data', 'plot_hlth.csv'))
```

08. Exporting the Choropleth Map

Saving maps

```
map.save('plot_data.html')
```

09. Discussion of Results

The three variables Economy, Family and Health are consistent in contributing to the Happiness Score

Does the analysis answer any of your existing research questions?

Questions to Explore from 6.2

1. Which country is the happiest per year?

2. What variable contributes most to defining a country's happiness?

a. Does the variable change per year?

3. How much does each variable contribute to a country's happiness?

4. Which country is the least happy per year?

1. Unable to determine with the Map, does not go to the granularity to distinguish the higher happy countries

2. The three variables Economy, Family and Health seem to contribute in similar degrees to the happiness score

2.a. Further investigation needed

3. Further investigation is needed

4. Further investigation is needed, however the continent of Africa and Southern Asia seems to have the lower variable scores

Does the analysis lead you to any new research questions?

1. The amount in which each variable contributes to the Happiness Rank?

2. Further investigation of how much each variable contributes to the country's Happiness Ranking from 2015 to 2019

****Submitting individual years in separate file***