3_Life_Expectancy_EDA

January 29, 2023

Table of Contents

1 Life_Expectancy_WHO_UN_Analysis_Modeling

1.1 Exploratory_Data_Analysis

To: Magnimind

From: Matt Curcio, matt.curcio.ri@gmail.com

Date: 2023-01-29

Re: NOTEBOOK #3

---

[1]:
```
# Common Python Libraries
import numpy as np
```

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

import warnings
warnings.simplefilter(action='ignore', category=UserWarning)

#pip install -U pandas-profiling
from pandas_profiling import ProfileReport

import plotly.express as px
```

[2]:
```
!ls *.csv
```

```
Clean_LE_Data_FEng_4.csv      Life_Expectancy_Data.csv  y_test.csv
Clean_LE_Data_Post_EDA_3.csv  x_test.csv                y_train.csv
Clean_LE_Data_w_Means_2.csv   x_train.csv
```

[3]:
```
filename = 'Life_Expectancy_Data.csv'

column_names = ['Country','Year','Status','LifeExpectancy','AdultMort',
                'InfD','EtOH','PercExpen','HepB','Measles',
                'BMI','lt5yD','Polio','TotalExpen','DTP','HIV',
                'GDP','Population','Thin1_19y','Thin5_9y','Income',
                'Education']

df = pd.read_csv(filename, names=column_names, header=0)

print(f'\nOriginal File "{filename}" has ', df.shape[0], 'observations &', df.
 ↪shape[1], 'features.\n')
```

```
Original File "Life_Expectancy_Data.csv" has  2938 observations & 22 features.
```

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Country          2938 non-null   object
 1   Year             2938 non-null   int64
 2   Status           2938 non-null   object
 3   LifeExpectancy   2928 non-null   float64
 4   AdultMort        2928 non-null   float64
```

```
 5   InfD             2938 non-null   int64
 6   EtOH             2744 non-null   float64
 7   PercExpen        2938 non-null   float64
 8   HepB             2385 non-null   float64
 9   Measles          2938 non-null   int64
 10  BMI              2904 non-null   float64
 11  lt5yD            2938 non-null   int64
 12  Polio            2919 non-null   float64
 13  TotalExpen       2712 non-null   float64
 14  DTP              2919 non-null   float64
 15  HIV              2938 non-null   float64
 16  GDP              2490 non-null   float64
 17  Population       2286 non-null   float64
 18  Thin1_19y        2904 non-null   float64
 19  Thin5_9y         2904 non-null   float64
 20  Income           2771 non-null   float64
 21  Education        2775 non-null   float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
```

1.2 Descriptive statistics of np.numeric data

[5]:
```
df.describe(include=[np.number], percentiles=None).T
```

[5]:

	count	mean	std	min	25%	50%	75%	max
Year	2938.0	2.007519e+03	4.613841e+00	2000.00000	2004.000000	2008.000000	2012.000000	2015.000000
LifeExpectancy	2928.0	6.922493e+01	9.523867e+00	36.30000	63.100000	72.100000	75.700000	89.000000
AdultMort	2928.0	1.647964e+02	1.242921e+02	1.00000	74.000000	144.000000	228.000000	723.000000
InfD	2938.0	3.030395e+01	1.179265e+02	0.00000	0.000000	0.000000	3.000000	1800.000000
EtOH	2744.0	4.602861e+00	4.052413e+00	0.01000	0.877500	3.755000	7.702500	17.870000
PercExpen	2938.0	7.382513e+02	1.987915e+03	0.00000	4.685343	64.912906	441.534144	17602.252760
HepB	2385.0	8.094046e+01	2.507002e+01	1.00000	77.000000	92.000000	97.000000	99.000000
Measles	2938.0	2.419592e+03	1.146727e+04	0.00000	0.000000	17.000000	360.250000	212183.000000
BMI	2904.0	3.832125e+01	2.004403e+01	1.00000	19.300000	43.500000	56.200000	77.100000
lt5yD	2938.0	4.203574e+01	1.604455e+02	0.00000	0.000000	4.000000	28.000000	2500.000000
Polio	2919.0	8.255019e+01	2.342805e+01	3.00000	78.000000	93.000000	97.000000	99.000000
TotalExpen	2712.0	5.938190e+00	2.498320e+00	0.37000	4.260000	5.755000	7.492500	17.600000
DTP	2919.0	8.232408e+01	2.371691e+01	2.00000	78.000000	93.000000	97.000000	99.000000
HIV	2938.0	1.742103e+00	5.077785e+00	0.10000	0.100000	0.100000	0.800000	50.600000
GDP	2490.0	7.483158e+03	1.427017e+04	1.68135	463.935626	1766.947595	5910.806335	119172.741800
Population	2286.0	1.275338e+07	6.101210e+07	34.00000	195793.250000	1386542.000000	7420359.000000	1293859294.000000
Thin1_19y	2904.0	4.839704e+00	4.420195e+00	0.10000	1.600000	3.400000	7.200000	27.700000
Thin5_9y	2904.0	4.870317e+00	4.508882e+00	0.10000	1.500000	3.300000	7.200000	28.600000
Income	2771.0	6.275511e-01	2.109036e-01	0.00000	0.493000	0.677000	0.779000	0.948000
Education	2775.0	1.199279e+01	3.358920e+00	0.00000	10.100000	12.300000	14.300000	20.700000

```
Year             2.008000e+03  2.012000e+03  2.015000e+03
LifeExpectancy   7.210000e+01  7.570000e+01  8.900000e+01
AdultMort        1.440000e+02  2.280000e+02  7.230000e+02
InfD             3.000000e+00  2.200000e+01  1.800000e+03
EtOH             3.755000e+00  7.702500e+00  1.787000e+01
PercExpen        6.491291e+01  4.415341e+02  1.947991e+04
HepB             9.200000e+01  9.700000e+01  9.900000e+01
Measles          1.700000e+01  3.602500e+02  2.121830e+05
BMI              4.350000e+01  5.620000e+01  8.730000e+01
lt5yD            4.000000e+00  2.800000e+01  2.500000e+03
Polio            9.300000e+01  9.700000e+01  9.900000e+01
TotalExpen       5.755000e+00  7.492500e+00  1.760000e+01
DTP              9.300000e+01  9.700000e+01  9.900000e+01
HIV              1.000000e-01  8.000000e-01  5.060000e+01
GDP              1.766948e+03  5.910806e+03  1.191727e+05
Population       1.386542e+06  7.420359e+06  1.293859e+09
Thin1_19y        3.300000e+00  7.200000e+00  2.770000e+01
Thin5_9y         3.300000e+00  7.200000e+00  2.860000e+01
Income           6.770000e-01  7.790000e-01  9.480000e-01
Education        1.230000e+01  1.430000e+01  2.070000e+01
```

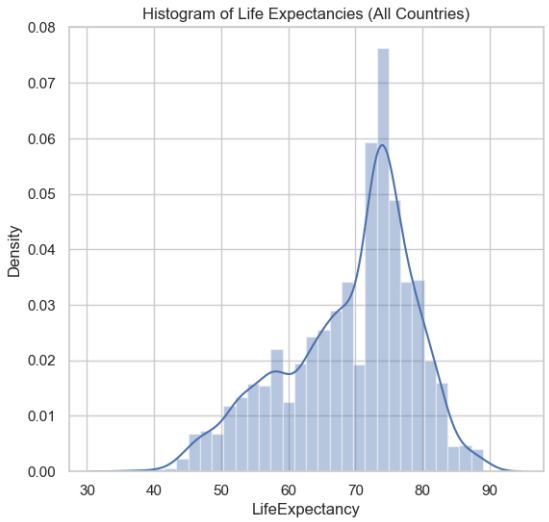
1.3 Counts of object data

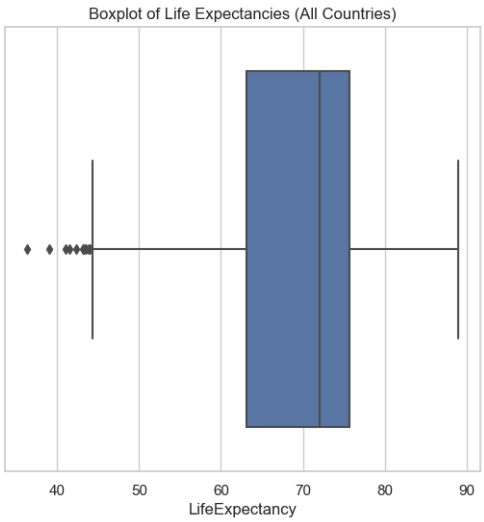
[6]:
```
df.describe(include=[object]).T
```

[6]:
```
         count unique          top  freq
Country   2938    193  Afghanistan    16
Status    2938      2   Developing  2426
```

1.4 Visualizing life expectancy for 193 countries

[7]:
```
# Histogram and Boxplot of Life Expectancy

plt.figure(figsize=(14,6))
sns.set_theme(style="whitegrid")

plt.subplot(1,2,1)
plt.title('Histogram of Life Expectancies (All Countries)')
sns.distplot(df['LifeExpectancy'])

plt.subplot(1,2,2)
plt.title('Boxplot of Life Expectancies (All Countries)')
sns.boxplot(x=df['LifeExpectancy'])

plt.show()
```





1.4.1 NOTE 1: Inference

1. **Skewness= -0.6386047359**
2. **Kurtosis = -0.2344773942**
   - Whenever the Kurtosis is less than zero or negative, it refers to distributions that are wider than the standard normal. It also refers to distributions that have thinner tails than the standard normal curve. This simply means that more data values are located near the mean and less data values are located at the extremes.
3. The Life Expectancy between "*developing nations*" and "*developed nations*" has mean (**69.2 y**) and the median (**72.1 y**) are not appreciably different.

[8]:
```
import plotly.express as px

plt.figure(figsize=(12,8))
fig = px.violin(df, y="LifeExpectancy",
                color="Status",
                box=True,
                points="all",
                hover_data=df.columns
               )

fig.update_layout(margin=dict(l=20, r=20, t=40, b=10),
                  paper_bgcolor="Lightyellow",
                  title='Plots of Developing (0) vs Developed (1) Nation Status'
                 )

fig.show()
```

```
<Figure size 1200x800 with 0 Axes>
```

1.5 Exploratory Data Analysis With Pandas Profiling

1.5.1 NOTE 2: Using `Pandas Profiling`

- For this report, I use the `Pandas Profiling` library, pandas-profiling.ydata.ai.
- Report(s) are customizable by changing `config_default.yaml`. *Be warned:* my first report was approximately 400 in PDF format.
- Useful aspects are `Alerts` and standardized reporting. *Question: Is a template useful and customizable too?*
- View Pandas-Profiling HTML

[9]:
```
from pandas_profiling import ProfileReport

profile = ProfileReport(df,
                        title="Life Expecteancy WHO UN Data: EDA Report Using␣
 ↪Pandas Profiling",
                        config_file="/home/mcc/Desktop/
 ↪Life_Expectancy_Analysis_Modeling/reports/default_ON.yaml")

profile.to_file("Life Expecteancy WHO UN Data: EDA Report Using Pandas␣
 ↪Profiling.html")

profile.to_notebook_iframe()
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]

Export report to file:   0%|          | 0/1 [00:00<?, ?it/s]

<IPython.core.display.HTML object>
```

1.5.2 NOTE 3: Pandas Profiling Alerts

24 Alerts were obtained from `WHO Life Expecteancy: IDA_EDA_Report`, a list of all alerts can be found on the Report.html

1. `Infant Death` and `Number of less than 5yr deaths` are highly correlated, **correlation = 0.997**.
2. `Thinness at 1-19yr` and `Thinness at 5-9yr` are highly correlated, **correlation = 0.939**.
3. `Perc_Expen` is highly correlated with `GDP`, **correlation = 0.899**.
4. `HepB` is highly correlated with `Polio`, **correlation = 0.674**
5. `Polio` is highly correlated with `DTP`, **correlation = 0.674**.

6. `Measles` is highly correlated with `Infant Death`, and `Number deaths less than 5yr`.
7. `BMI` is highly correlated with a countries `Status`, `GDP`, `Education` and `EtOH`.

1.5.3 NOTE 4: Further Questions & Possible Directions**

1. What is difference in Life Expectancy between Developing and Developed countries?
2. What is Percent Expenditure versus Total Expenditure????? `Percent Expenditure` and `Total Expenditure`, What is difference?
3. Carry out the most simple linear model first. Get an idea of the factors and coefficients involved.
4. Did Life Expectancy change over the time/course of the study? 2000-2015?

1.6 Drop Highly Correlated Features

The *WHO Life Expecteancy: IDA_EDA_Report* found six variables were highly correlated.

Variables	Correlation
(InfD) Infant Death & (lt5yD) Number of less than 5yr deaths	0.997
Thinness at 1-19yr & Thinness at 5-9yr	0.939
Perc_Expen & GDP	0.899

- Since GDP was dropped due to 22% missing values ANY comparsion IS NOT Necessary
- In order to determine which ones to remove, we will compare the correlations after each removal.
- It is reccommened that removing descriptors with absolute correlations above 0.75 is done by a 'bake-off' method, for reference, see Max Kuhn
- BTW, my co-worker Peter Brown used the 'bake-off' idea alot when doing scientific comparisons, but I don't think anyone will ever care or read this far.

https://topepo.github.io/caret/pre-processing.html#identifying-correlated-predictors

1.6.1 Compare Correlation coeff of (InfD) Infant Death Vs. (lt5yD) Number of less than 5yr deaths

[10]:
```
# Use: `Clean_LE_Data_w_Means_2.csv`

filename = 'Clean_LE_Data_w_Means_2.csv'

df = pd.read_csv(filename, header=0)
```

[11]:
```
# Generate Corr Coeff for InfD vs LifeExpectancy

np.corrcoef(df['InfD'], df['LifeExpectancy'])
```

[11]:
```
array([[ 1.        , -0.19655718],
       [-0.19655718,  1.        ]])
```

[12]:
```
# Generate Corr Coeff for lt5yD vs LifeExpectancy

np.corrcoef(df['lt5yD'], df['LifeExpectancy'])
```

[12]:
```
array([[ 1.        , -0.22252912],
       [-0.22252912,  1.        ]])
```

1.6.2 NOTE 5:

- The correlation of `lt5yd` = -0.22252912
- The correlation of `InfD` = -0.19655718
- Therefore DELETE `InfD`

1.6.3 Compare Correlation coeff of (Thin1_19y) Thinness at 1-19yr & (Thin5_9y) Thinness at 5-9yr

[13]:
```
np.corrcoef(df['Thin1_19y'], df['LifeExpectancy'])
```

[13]:
```
array([[ 1.        , -0.47277841],
       [-0.47277841,  1.        ]])
```

[14]:
```
np.corrcoef(df['Thin5_9y'], df['LifeExpectancy'])
```

[14]:
```
array([[ 1.        , -0.46723051],
       [-0.46723051,  1.        ]])
```

1.6.4 NOTE 6:

- The correlation of `Thin1_19y` = -0.47277841
- The correlation of `Thin5_9y` = -0.46723051
- Therefore DELETE `Thin5_9y`

[15]:
```
# Drop TWO Features

df.drop(['Thin5_9y','InfD'], axis=1, inplace=True)
```

1.7 Save Intermediate dataframe

[16]:
```
df.to_csv("Clean_LE_Data_Post_EDA_3.csv", index=False)
```

[17]:
```
!ls *.csv
```

```
Clean_LE_Data_FEng_4.csv      Life_Expectancy_Data.csv  y_test.csv
Clean_LE_Data_Post_EDA_3.csv  x_test.csv                y_train.csv
Clean_LE_Data_w_Means_2.csv   x_train.csv
```

[18]:
```
df.head(3)
```

[18]:

	Country	Year	Status	LifeExpectancy	AdultMort	EtOH	PercExpen	\
0	Afghanistan	2015	0	65.0	263.0	0.01	71.279624	
1	Afghanistan	2014	0	59.9	271.0	0.01	73.523582	
2	Afghanistan	2013	0	59.9	268.0	0.01	73.219243	

	Measles	BMI	lt5yD	Polio	TotalExpen	DTP	HIV	Thin1_19y	Income	\
0	1154	19.1	83	6.0	8.16	65.0	0.1	17.2	0.479	
1	492	18.6	86	58.0	8.18	62.0	0.1	17.5	0.476	
2	430	18.1	89	62.0	8.13	64.0	0.1	17.7	0.470	

	Education
0	10.1
1	10.0
2	9.9

[ ]:
```

```