Hands on Activity 10.1 Data Analysis using Python

Name: Calingo, Christian Lei

Section: CPE22S3

Course: Computational Thinking with Python

Course Code: CPE311

Intended Learning Outcomes

- Perform descriptive and correlation analysis to to analyze the dataset
- Interpret the results of descriptive and correlation analysis

Resources

- Personal Computer
- Jupyter Notebook
- Internet Connection

Instruction

1. Gather a dataset regarding your identified problem for the ASEAN Data Science Explorer. Make sure that the dataset includes multiple variables.
2. Load the dataset into pandas dataframe.
3. Prepare the data by applying appropriate data preprocessing techniques.
4. Analyze the data using descriptive analysis.
5. Perform correlation analysis.
6. Interpret the results based on the descriptive and correlation analysis.
7. Submit the PDF file.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
data = pd.read_csv('data/CO2_emission.csv')
data
```

	Country Name	country_code	Region	Indicator Name	1990	1991	1992	199
0	Aruba	ABW	Latin America & Caribbean	CO2 emissions (metric tons per capita)	NaN	NaN	NaN	Nal
1	Afghanistan	AFG	South Asia	CO2 emissions (metric tons per capita)	0.191745	0.167682	0.095958	0.08472
2	Angola	AGO	Sub-Saharan Africa	CO2 emissions (metric tons per capita)	0.553662	0.544539	0.543557	0.70898
3	Albania	ALB	Europe & Central Asia	CO2 emissions (metric tons per capita)	1.819542	1.242810	0.683700	0.63830
4	Andorra	AND	Europe & Central Asia	CO2 emissions (metric tons per capita)	7.521832	7.235379	6.963079	6.72417
...	...	...	...	...	...	...	...	.
210	Samoa	WSM	East Asia & Pacific	CO2 emissions (metric tons per capita)	0.552836	0.609756	0.604266	0.65822
211	Yemen, Rep.	YEM	Middle East & North Africa	CO2 emissions (metric tons per	0.567037	0.690937	0.704793	0.62710

```
#putting all the ASEAN countries
asean = ['Vietnam', 'Indonesia', 'Philippines', 'Thailand', 'Myanmar', 'Cambodia', 'Malaysia', 'Lao PDR', 'Singapore', 'Brunei Darussalam']
```

```
data_new = data[data['Country Name'].isin(asean)] #creating a new dataframe for the ASEAN Countries
data_new.drop(['Region', 'country_code', 'Indicator Name', '2019.1'], axis = 1, inplace = True) #dropping unecessary columns in the datafra
data_new.set_index('Country Name', inplace = True) #setting the country name as the index
data_new
```

```
<ipython-input-207-de09a7e22ca8>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user
  data_new.drop(['Region', 'country_code', 'Indicator Name', '2019.1'], axis = 1, inpla
```

Country Name	1990	1991	1992	1993	1994	1995	1996	
Brunei Darussalam	12.600787	12.696839	13.107548	13.951804	14.752014	15.482377	15.855820	1
Indonesia	0.818738	0.879779	0.913552	0.969813	1.027138	1.135811	1.184185	
Cambodia	0.140381	0.139946	0.140276	0.138405	0.143440	0.137949	0.139307	
Lao PDR	0.119761	0.125593	0.133323	0.132036	0.135140	0.138245	0.157538	
Myanmar	0.098705	0.095965	0.100939	0.112147	0.130234	0.158081	0.164446	
Malaysia	3.029425	3.515130	3.534768	3.748544	3.991489	4.212791	4.693205	
Philippines	0.663703	0.640614	0.677021	0.717783	0.755491	0.884872	0.941434	
Singapore	9.507301	9.719041	9.722357	10.913027	11.108355	10.662487	10.493900	1
Thailand	1.577490	1.713538	1.867291	2.103611	2.327972	2.619592	2.905529	
Vietnam	0.284311	0.285151	0.293580	0.335268	0.362248	0.419167	0.460505	

10 rows × 30 columns

```
data_new = data_new.round(4) # rounding the values to 4 decimal
data_new
```

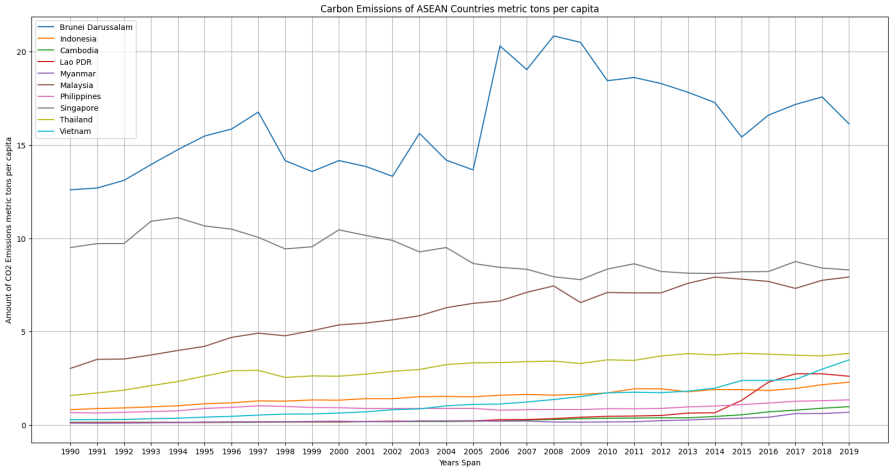
Country Name	1990	1991	1992	1993	1994	1995	1996	1997	1998
Brunei Darussalam	12.6008	12.6968	13.1075	13.9518	14.7520	15.4824	15.8558	16.7649	14.1633
Indonesia	0.8187	0.8798	0.9136	0.9698	1.0271	1.1358	1.1842	1.2876	1.2769
Cambodia	0.1404	0.1399	0.1403	0.1384	0.1434	0.1379	0.1393	0.1452	0.1621
Lao PDR	0.1198	0.1256	0.1333	0.1320	0.1351	0.1382	0.1575	0.1643	0.1652
Myanmar	0.0987	0.0960	0.1009	0.1121	0.1302	0.1581	0.1644	0.1610	0.1734
Malaysia	3.0294	3.5151	3.5348	3.7485	3.9915	4.2128	4.6932	4.9168	4.7792
Philippines	0.6637	0.6406	0.6770	0.7178	0.7555	0.8849	0.9414	1.0241	0.9939
Singapore	9.5073	9.7190	9.7224	10.9130	11.1084	10.6625	10.4939	10.0526	9.4342
Thailand	1.5775	1.7135	1.8673	2.1036	2.3280	2.6196	2.9055	2.9244	2.5459
Vietnam	0.2843	0.2852	0.2936	0.3353	0.3622	0.4192	0.4605	0.5295	0.5821

Plotting the Carbon Emissions of each ASEAN country is crucial, with that, we can identify if the Carbon Emissions of each ASEAN country throughout the years. And also, we can identify which country contributes the most carbon emissions throught the ASEAN Countries

```
plt.figure(figsize = (20,10)) #setting the size of the figure

for asean in data_new.index: #iterating each index in the dataframe
  plt.plot(data_new.columns, data_new.loc[asean], label = asean) #plotting the values of each values throughout the years

plt.title('Carbon Emissions of ASEAN Countries metric tons per capita')
plt.xlabel('Years Span')
plt.ylabel('Amount of CO2 Emissions metric tons per capita')
plt.legend()
plt.grid(True)
plt.show()
```



Importing another data to see the air pollutants for each country

Now, let's get the cause of those Carbon Emissions, we are going to get the data about the how much air pollutant is getting burned with their peak year. To do it, we are going to get their max value and find the year where it belongs

```
data2 = pd.read_csv('data/air-pollution.csv')
asean2 = ['Vietnam', 'Indonesia', 'Philippines', 'Thailand', 'Myanmar', 'Cambodia', 'Malaysia', 'Laos', 'Singapore', 'Brunei']
data2 = data2[data2['Country'].isin(asean2)]
data2.set_index('Country', inplace = True)
data2
```

Country	Year	Nitrogen Oxide	Sulphur Dioxide	Carbon Monoxide	Organic Carbon	NMVOCs	Black Carbon	Ammonia
Brunei	1750	1.60	0.15	675.04	41.98	98.92	10.32	56.67
Brunei	1760	1.62	0.15	684.57	42.58	100.36	10.47	57.72
Brunei	1770	1.65	0.15	693.94	43.16	101.77	10.61	58.79
Brunei	1780	1.68	0.15	703.09	43.73	103.16	10.75	59.88
Brunei	1790	1.70	0.16	711.99	44.28	104.52	10.89	60.99
...	...	...	...	...	...	...	...	...
Vietnam	2015	917325.10	508118.33	10231802.65	214056.91	2292300.32	75755.97	648530.61
Vietnam	2016	959776.72	557030.51	9604751.28	214258.27	2196311.06	77354.00	649176.57
Vietnam	2017	947335.92	561289.11	10256054.90	212595.19	2234926.07	77942.18	654878.46
Vietnam	2018	1052623.14	695156.64	10914872.55	219213.88	2262785.93	83608.08	661310.60
Vietnam	2019	1177238.84	852061.42	11708923.29	226891.68	2296905.23	90055.98	667801.76

2250 rows × 8 columns

Next steps: View recommended plots

In this code, we are cleaning the dataframe so that only the values where they reached their peak carbon emissions is only shown.

```
import pandas as pd

peak_year = data_new.idxmax(axis=1) #getting the column of the max value of each row
peak_year_list = [] #list for the year of the peak values

for _, x in peak_year.items(): #iterations for the peak_year #using (_) placeholder to ignore the country
    x2 = int(x) #changing the datatype of the year
    peak_year_list.append(x2) #appending it in a list

asean2 = ['Vietnam', 'Indonesia', 'Philippines', 'Thailand', 'Myanmar', 'Cambodia', 'Malaysia', 'Laos', 'Singapore', 'Brunei'] #accessing the
asean2.sort() # sorting values

results_dict = {} # creating an empty dictionary to store results for each country

for country, year_p in zip(asean2, peak_year_list): #zipping the 2 list for simultaneous iterations
    values = (data2.index == country) & (data2['Year'] == year_p) #locating the values in the dataframe and checking if its in the list and a
    results_dict[country] = data2[values] # assigning a values to the keys of the dictionary


combined_df = pd.concat(results_dict.values()) #combine all results into one dataframe

data3 = pd.DataFrame(combined_df)
data3.round(2)
```

Country	Year	Nitrogen Oxide	Sulphur Dioxide	Carbon Monoxide	Organic Carbon	NMVOCs	Black Carbon	Ammonia
Brunei	2008	14745.08	13148.01	14556.97	3362.02	53171.81	1418.86	8306.35
Cambodia	2019	211588.46	67094.55	2594187.26	49428.24	562514.70	14641.60	179382.08
Indonesia	2019	5038852.43	2595484.38	23309812.17	755683.79	7602169.19	269256.86	2003211.60
Laos	2018	115993.80	254805.24	369870.40	8453.94	92352.67	2360.22	99160.65
Malaysia	2019	896638.04	377090.03	3839309.87	41831.40	1555014.58	33978.93	284254.01
Myanmar	2019	573915.39	141558.14	3272663.92	164578.99	1050164.41	52538.17	458705.28
Philippines	2019	669617.54	545617.99	3266633.22	87735.67	1248417.22	42677.71	514186.96
Singapore	1994	168603.39	851815.32	93147.35	10009.80	209003.88	6483.81	9787.24
Thailand	2015	1082939.05	457175.78	11387595.78	221487.63	3244526.66	99281.14	705405.56
Vietnam	2019	1177238.84	852061.42	11708923.29	226891.68	2296905.23	90055.98	667801.76

```
plt.figure(figsize = (20,10))

for asean2 in data3.index:
  plt.plot(data3.columns, data3.loc[asean2], label = asean2)

plt.xticks(data3.columns[1:])

plt.title('Air Pollutants on their Peak Year of Carbon Emission')
plt.xlabel('Years Span')
plt.ylabel('Amount of Air Pollutants')
plt.legend()
plt.grid(True)
plt.show()
```