Hands-on Activity 10.1 Data Analysis using Python

Name: Cuadra, Audrick Zander G.

Section: CPE22S3

Date: April 3, 2024

Submitted to: Engr. Roman Richard

Intended Learning Outcome

1. Perform descriptive and correlation analysis to to analyze the dataset.
2. Interpret the results of descriptive and correlation analysis

Resources

- Personal Computer
- Google Colab
- 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.

```
import pandas as pd
import numpy as np

co2e = pd.read_csv(
    '/content/Total GHG by Sector.csv',
)
co2e
```

	Year	Agriculture	Commercial	Electricity	Energy (own-use)	Industry	Residential	Transp
0	1990	0.15	0.33	10.21	1.17	9.09	2.20	13
1	1991	0.14	0.37	13.48	1.08	6.98	2.17	13
2	1992	0.13	0.41	14.30	1.26	7.35	2.43	15
3	1993	0.12	0.47	14.76	1.26	8.96	2.68	17
4	1994	0.11	0.52	15.49	1.24	8.24	2.89	17
5	1995	0.12	0.61	16.60	1.34	11.31	3.31	21
6	1996	0.13	0.65	18.20	1.56	10.64	3.55	23
7	1997	1.17	0.81	21.76	1.50	12.46	3.81	23
8	1998	1.04	1.16	22.09	1.62	11.30	3.92	24
9	1999	0.99	1.47	19.44	1.81	10.88	4.20	24
10	2000	0.86	1.69	21.44	2.19	8.89	3.94	24
11	2001	0.81	2.04	22.48	1.83	8.37	3.68	24
12	2002	0.87	2.07	21.45	1.79	7.85	3.67	25
13	2003	0.94	1.92	22.55	2.15	9.06	3.35	24
14	2004	0.88	1.64	23.95	1.71	8.88	3.50	25
15	2005	0.86	1.47	26.53	1.59	9.33	2.88	24
16	2006	0.76	1.58	23.12	1.15	9.51	2.60	22
17	2007	0.62	1.54	25.00	1.18	10.02	2.51	23
18	2008	0.76	1.53	27.76	0.98	11.71	2.41	21
19	2009	0.61	2.44	28.27	0.90	10.09	2.52	22
20	2010	0.64	2.78	31.28	1.02	11.68	2.50	22
21	2011	0.56	2.89	32.32	0.94	11.38	2.46	22
22	2012	0.54	2.84	34.58	1.04	10.54	2.41	23
23	2013	0.57	3.30	40.18	0.89	12.16	2.36	24
24	2014	0.51	4.22	43.07	1.05	12.68	2.31	25
25	2015	0.58	3.79	46.89	0.91	12.99	2.60	29
26	2016	0.69	4.80	50.95	0.63	15.05	2.99	32
27	2017	0.87	6.06	58.24	0.68	16.36	3.08	33
28	2018	0.62	6.51	63.76	0.74	13.99	3.34	34
29	2019	0.69	6.93	69.40	1.00	12.96	3.49	35

Next steps: View recommended plots

3. Prepare the data by applying appropriate data preprocessing techniques.

```
# filtering out the datas that are lower than 2001
co2e2000s = co2e.query('Year > 2000')
co2e2000s
```

	Year	Agriculture	Commercial	Electricity	Energy (own-use)	Industry	Residential	Transp
11	2001	0.81	2.04	22.48	1.83	8.37	3.68	24
12	2002	0.87	2.07	21.45	1.79	7.85	3.67	25
13	2003	0.94	1.92	22.55	2.15	9.06	3.35	24
14	2004	0.88	1.64	23.95	1.71	8.88	3.50	25
15	2005	0.86	1.47	26.53	1.59	9.33	2.88	24
16	2006	0.76	1.58	23.12	1.15	9.51	2.60	22
17	2007	0.62	1.54	25.00	1.18	10.02	2.51	23
18	2008	0.76	1.53	27.76	0.98	11.71	2.41	21
19	2009	0.61	2.44	28.27	0.90	10.09	2.52	22
20	2010	0.64	2.78	31.28	1.02	11.68	2.50	22
21	2011	0.56	2.89	32.32	0.94	11.38	2.46	22
22	2012	0.54	2.84	34.58	1.04	10.54	2.41	23
23	2013	0.57	3.30	40.18	0.89	12.16	2.36	24
24	2014	0.51	4.22	43.07	1.05	12.68	2.31	25
25	2015	0.58	3.79	46.89	0.91	12.99	2.60	29
26	2016	0.69	4.80	50.95	0.63	15.05	2.99	32
27	2017	0.87	6.06	58.24	0.68	16.36	3.08	33
28	2018	0.62	6.51	63.76	0.74	13.99	3.34	34
29	2019	0.69	6.93	69.40	1.00	12.96	3.49	35

Next steps: View recommended plots

```
# transforming Year column into index
co2e2000s.set_index('Year', inplace=True)
co2e2000s
```

	Agriculture	Commercial	Electricity	Energy (own-use)	Industry	Residential	Transport
Year							
2001	0.81	2.04	22.48	1.83	8.37	3.68	24.19
2002	0.87	2.07	21.45	1.79	7.85	3.67	25.08
2003	0.94	1.92	22.55	2.15	9.06	3.35	24.47
2004	0.88	1.64	23.95	1.71	8.88	3.50	25.34
2005	0.86	1.47	26.53	1.59	9.33	2.88	24.12
2006	0.76	1.58	23.12	1.15	9.51	2.60	22.65
2007	0.62	1.54	25.00	1.18	10.02	2.51	23.45
2008	0.76	1.53	27.76	0.98	11.71	2.41	21.71
2009	0.61	2.44	28.27	0.90	10.09	2.52	22.74
2010	0.64	2.78	31.28	1.02	11.68	2.50	22.96
2011	0.56	2.89	32.32	0.94	11.38	2.46	22.75
2012	0.54	2.84	34.58	1.04	10.54	2.41	23.68
2013	0.57	3.30	40.18	0.89	12.16	2.36	24.75
2014	0.51	4.22	43.07	1.05	12.68	2.31	25.69
2015	0.58	3.79	46.89	0.91	12.99	2.60	29.71
2016	0.69	4.80	50.95	0.63	15.05	2.99	32.15
2017	0.87	6.06	58.24	0.68	16.36	3.08	33.20
2018	0.62	6.51	63.76	0.74	13.99	3.34	34.36
2019	0.69	6.93	69.40	1.00	12.96	3.49	35.57

Next steps: View recommended plots

```
# checking if the Year column turned into index
co2e2000s.dtypes
```

```
Agriculture          float64
Commercial           float64
Electricity          float64
Energy (own-use)     float64
Industry             float64
Residential          float64
Transport            float64
dtype: object
```

```
# adds a column "Total" that adds the sum of the rows
co2e2000s['Total'] = co2e2000s.sum(axis=1)
co2e2000s
```

```
<ipython-input-101-0d683eddf107>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  co2e2000s['Total'] = co2e2000s.sum(axis=1)
```

	Agriculture	Commercial	Electricity	Energy (own-use)	Industry	Residential	Transport
Year							
2001	0.81	2.04	22.48	1.83	8.37	3.68	24.19
2002	0.87	2.07	21.45	1.79	7.85	3.67	25.08
2003	0.94	1.92	22.55	2.15	9.06	3.35	24.47
2004	0.88	1.64	23.95	1.71	8.88	3.50	25.34
2005	0.86	1.47	26.53	1.59	9.33	2.88	24.12
2006	0.76	1.58	23.12	1.15	9.51	2.60	22.65
2007	0.62	1.54	25.00	1.18	10.02	2.51	23.45
2008	0.76	1.53	27.76	0.98	11.71	2.41	21.71
2009	0.61	2.44	28.27	0.90	10.09	2.52	22.74
2010	0.64	2.78	31.28	1.02	11.68	2.50	22.96
2011	0.56	2.89	32.32	0.94	11.38	2.46	22.75
2012	0.54	2.84	34.58	1.04	10.54	2.41	23.68
2013	0.57	3.30	40.18	0.89	12.16	2.36	24.75
2014	0.51	4.22	43.07	1.05	12.68	2.31	25.69
2015	0.58	3.79	46.89	0.91	12.99	2.60	29.71
2016	0.69	4.80	50.95	0.63	15.05	2.99	32.15
2017	0.87	6.06	58.24	0.68	16.36	3.08	33.20
2018	0.62	6.51	63.76	0.74	13.99	3.34	34.36
2019	0.69	6.93	69.40	1.00	12.96	3.49	35.57

Next steps: View recommended plots

```
# sort the dataframe by descending order of the total column
sort_co2 = co2e2000s.sort_values(by='Total', ascending=False)
sort_co2
```

	Agriculture	Commercial	Electricity	Energy (own-use)	Industry	Residential	Transport
Year							
2019	0.69	6.93	69.40	1.00	12.96	3.49	35.57
2018	0.62	6.51	63.76	0.74	13.99	3.34	34.36
2020	0.63	7.25	70.01	0.77	10.62	3.29	27.44
2017	0.87	6.06	58.24	0.68	16.36	3.08	33.20
2016	0.69	4.80	50.95	0.63	15.05	2.99	32.15
2015	0.58	3.79	46.89	0.91	12.99	2.60	29.71
2014	0.51	4.22	43.07	1.05	12.68	2.31	25.69
2013	0.57	3.30	40.18	0.89	12.16	2.36	24.75
2012	0.54	2.84	34.58	1.04	10.54	2.41	23.68
2011	0.56	2.89	32.32	0.94	11.38	2.46	22.75
2010	0.64	2.78	31.28	1.02	11.68	2.50	22.96
2009	0.61	2.44	28.27	0.90	10.09	2.52	22.74
2008	0.76	1.53	27.76	0.98	11.71	2.41	21.71
2005	0.86	1.47	26.53	1.59	9.33	2.88	24.12
2004	0.88	1.64	23.95	1.71	8.88	3.50	25.34
2003	0.94	1.92	22.55	2.15	9.06	3.35	24.47
2007	0.62	1.54	25.00	1.18	10.02	2.51	23.45
2001	0.81	2.04	22.48	1.83	8.37	3.68	24.19
2002	0.87	2.07	21.45	1.79	7.85	3.67	25.08

Next steps: View recommended plots

```
# displays the top 10 years that produced the most CO2
sort_head_co2 = sort_co2.head(10)
sort_head_co2
```

	Agriculture	Commercial	Electricity	Energy (own-use)	Industry	Residential	Transport
Year							
2019	0.69	6.93	69.40	1.00	12.96	3.49	35.57
2018	0.62	6.51	63.76	0.74	13.99	3.34	34.36
2020	0.63	7.25	70.01	0.77	10.62	3.29	27.44
2017	0.87	6.06	58.24	0.68	16.36	3.08	33.20
2016	0.69	4.80	50.95	0.63	15.05	2.99	32.15
2015	0.58	3.79	46.89	0.91	12.99	2.60	29.71
2014	0.51	4.22	43.07	1.05	12.68	2.31	25.69
2013	0.57	3.30	40.18	0.89	12.16	2.36	24.75
2012	0.54	2.84	34.58	1.04	10.54	2.41	23.68

Next steps: View recommended plots

4. Analyze the data using descriptive analysis.

```
# displays the descriptive data analysis of the top 10 years and its total
sort_head_co2.describe()
```

	Agriculture	Commercial	Electricity	Energy (own-use)	Industry	Residential	Transpo
count	10.000000	10.000000	10.000000	10.000000	10.00000	10.000000	10.0000
mean	0.626000	4.859000	50.940000	0.865000	12.87300	2.833000	28.930
std	0.104478	1.703346	13.866627	0.151089	1.86967	0.453727	4.699
min	0.510000	2.840000	32.320000	0.630000	10.54000	2.310000	22.750
25%	0.562500	3.422500	40.902500	0.747500	11.57500	2.422500	24.985
50%	0.600000	4.510000	48.920000	0.900000	12.82000	2.795000	28.575
75%	0.675000	6.397500	62.380000	0.985000	13.74000	3.237500	32.937

5. Perform correlation analysis.

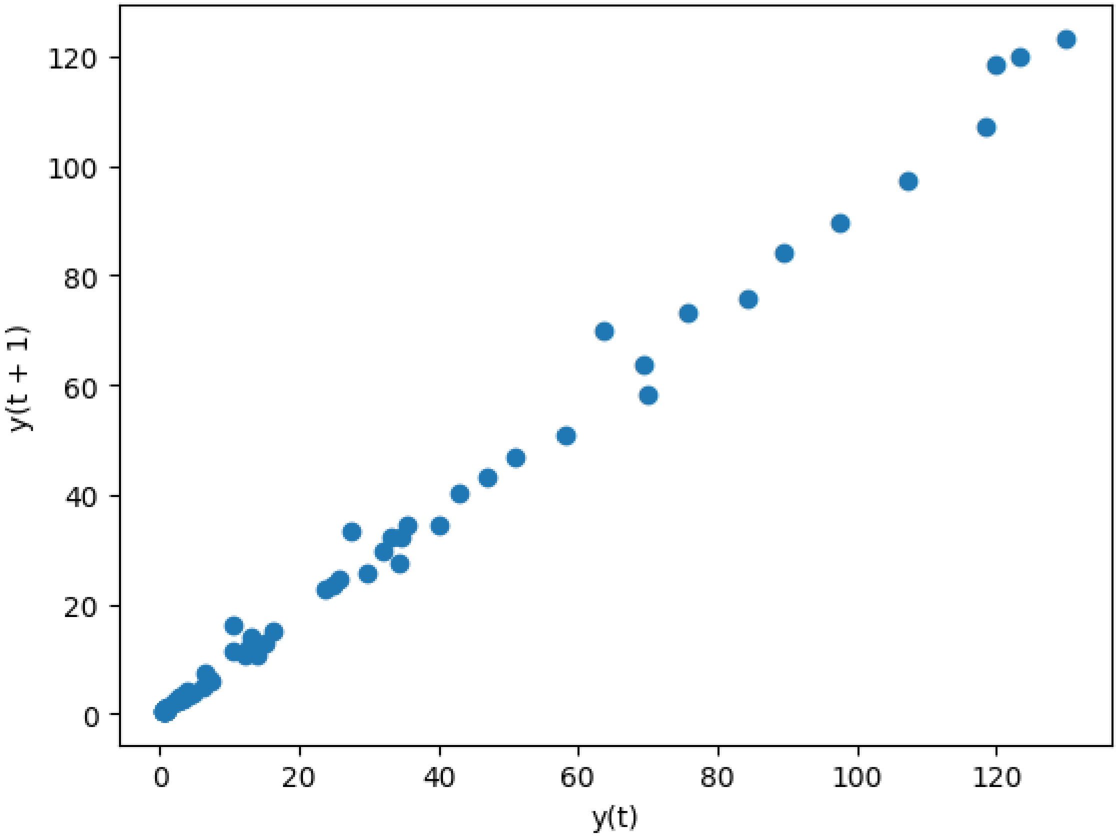
```
sort_head_co2.corr()
```

	Agriculture	Commercial	Electricity	Energy (own-use)	Industry	Residential	T
Agriculture	1.000000	0.584184	0.559892	-0.676435	0.760002	0.632902	
Commercial	0.584184	1.000000	0.985932	-0.461336	0.329937	0.932609	
Electricity	0.559892	0.985932	1.000000	-0.447692	0.313381	0.938850	
Energy (own-use)	-0.676435	-0.461336	-0.447692	1.000000	-0.644693	-0.521495	-
Industry	0.760002	0.329937	0.313381	-0.644693	1.000000	0.360400	
Residential	0.632902	0.932609	0.938850	-0.521495	0.360400	1.000000	
Transport	0.694308	0.758131	0.789944	-0.506827	0.721773	0.834375	

```
# displaying the correlation in a plot
from pandas.plotting import lag_plot

lag_plot(sort_head_co2)
```

```
<Axes: xlabel='y(t)', ylabel='y(t + 1)'>
```



6. Interpret the results based on the descriptive and correlation analysis.

- It displays a strong positive correlation of the greenhouse emissions. It also shows the averages of different greenhouse emissions per year as well as the fluctuation of values in the years were CO2 emissions are in their all time high. The standard deviation provides insights on the amount of emissions that deviate from their respective averages during its peak. This benefits us in a way that we can identify the consistency of greenhouse emission during these pivotal years.