

## Assignment 2

Complete the code cells in this Jupyter notebook, and submit the final .ipynb notebook to Gradescope

- Make sure you save your changes in VS Code before uploading it (beware of the little "dot" in the tab above)
- You should **execute all code cells** before saving and submitting. We need to see the outputs/figures generated by Python

### (a) Import the pandas and matplotlib packages

```
In [1]: # Write your answer

import pandas as pd
import matplotlib.pyplot as plt
```

### (b) Open the dataset

"united\_states\_co2\_emissions.csv" in the "data" folder using pandas

- Assign it to a variable named "us\_emissions"
- Note: This contains CO2 emissions (in kilotons) by the United States from 1960–2019.

It is compiled from various sources, including the United Nations Framework Convention

on Climate Change (UNFCCC) and the International Energy Agency (IEA).

```
In [2]: # Write your answer
us_emissions = pd.read_csv('data/united_states_co2_emissions.csv')
print(type(us_emissions))
```

```
<class 'pandas.core.frame.DataFrame'>
```

### (c) View the dataset and create a list of variable names

- The us\_emissions dataset has two named column headings.
- Create a list containing two strings, one for each of the column headings.

- The strings must match the headings **exactly** (the "#" symbols shown in DataWrangler are not part of the name). Otherwise you will run into a problem in part d)
- Assign this list to a variable called "columns"
- You do not have to "extract" the names from the dataset through a Python command. You can simply type them manually.

```
In [3]: # Write your answer
year = "Year"
total_emissions = "Total Emissions"
print(type(year))
print(type(total_emissions))
columns = year + total_emissions
print(type(columns))
```

```
<class 'str'>
<class 'str'>
<class 'str'>
```

## (d) Compute descriptive statistics for the 2nd variable in the dataset

- The direct way to do this is to execute the command:  
`us_emissions["total_emissions"].describe()`
- Do something similar in the code cell below, but typing the string "total\_emissions" is **not allowed**
- Instead of typing "total\_emissions" manually, use the variable "columns" you created in the previous cell, extracting the list's 2nd element
- **Remember:** The numbering in Python starts at zero
- For example, if

```
list_colors = ["red","green","yellow"],
```

then `list_colors[2]` will be "yellow"

```
In [4]: # Write your answer
columns_list = ["year", "total_emissions"]
columns_emissions = columns_list[1]
```

```
print(columns_emissions)
us_emissions[columns_emissions].describe()
```

total\_emissions

```
Out[4]: count    6.000000e+01
        mean     4.733944e+06
        std      7.568200e+05
        min      2.880506e+06
        25%      4.457976e+06
        50%      4.862075e+06
        75%      5.160723e+06
        max      5.775810e+06
        Name: total_emissions, dtype: float64
```

**Hint:** If you did everything correctly above, that cell should have the same output as the one below

```
In [5]: # run this cell to compare with your answer
        us_emissions["total_emissions"].describe()
```

```
Out[5]: count    6.000000e+01
        mean     4.733944e+06
        std      7.568200e+05
        min      2.880506e+06
        25%      4.457976e+06
        50%      4.862075e+06
        75%      5.160723e+06
        max      5.775810e+06
        Name: total_emissions, dtype: float64
```

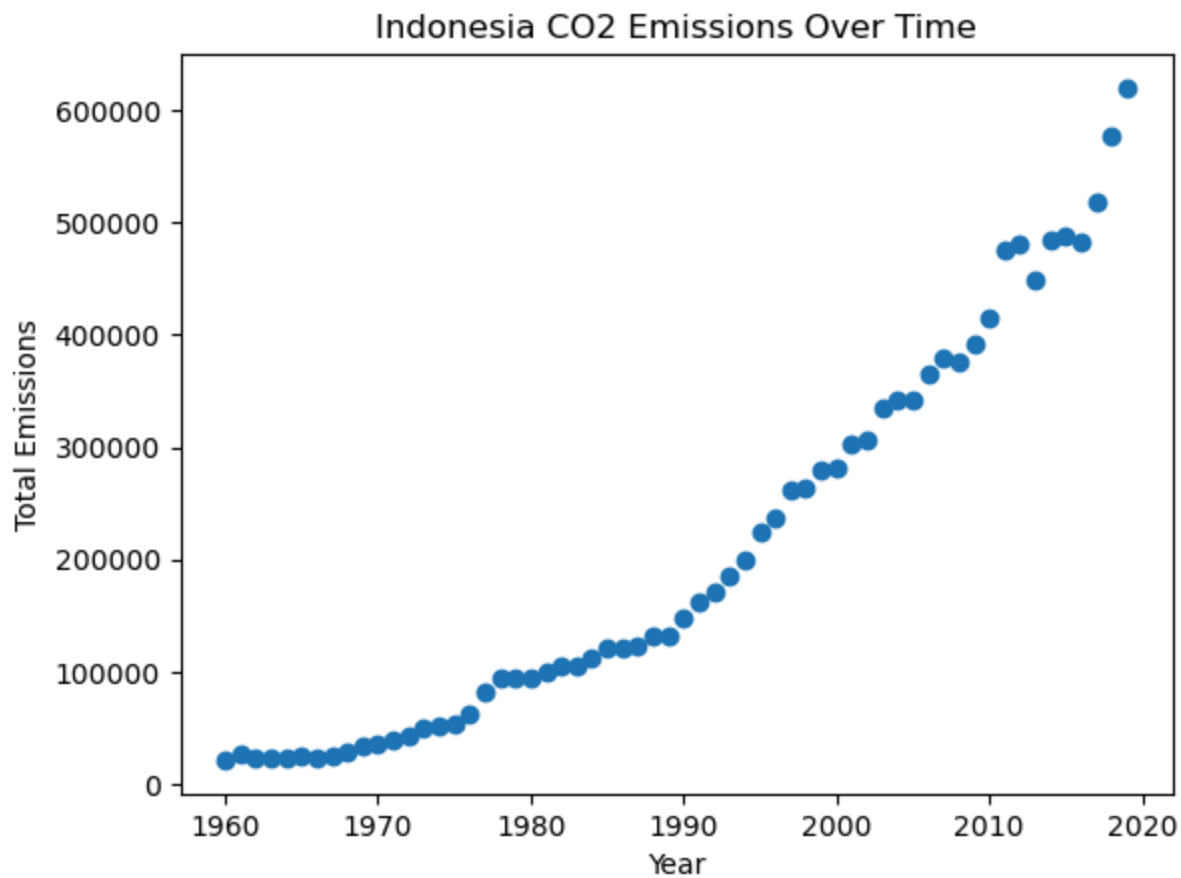
(e) Choose any 2 of the other countries in the "data" folder and import the 2 datasets as you did in part (b). Create 2 **scatter plots** that have "year" on the x-axis (horizontal axis) and "total\_emissions" in the y-axis.

- The "data" folder contains the datasets for the top 10 CO2 emitting countries
- For **full points**, label the axes in each plot.
- Use "plt.title("Relevant Title for Plot")" to create a title for the plot (substituting a more appropriate string, of course)
- Use "plt.show()" after creating each plot to ensure both are displayed
- Note: This question does not depend on (d)

```
In [44]: # Write your answer
indonesia_emissions = pd.read_csv('data/indonesia_co2_emissions.csv')
print(type(indonesia_emissions))
indonesia_emissions
table = pd.crosstab(index = indonesia_emissions['year'], columns = indonesia_emissions['total_emissions'])
indonesia_emissions.describe()
plt.scatter(x = indonesia_emissions['year'], y = indonesia_emissions['total_emissions'])
plt.title("Indonesia CO2 Emissions Over Time")
plt.xlabel(year)
plt.ylabel(total_emissions)
plt.show()

china_emissions = pd.read_csv('data/china_co2_emissions.csv')
print(type(china_emissions))
china_emissions
table = pd.crosstab(index = china_emissions['year'], columns = china_emissions['total_emissions'])
china_emissions.describe()
plt.scatter(x = china_emissions['year'], y = china_emissions['total_emissions'])
plt.title("China CO2 Emissions Over Time")
plt.xlabel(year)
plt.ylabel(total_emissions)
plt.show()

<class 'pandas.core.frame.DataFrame'>
```



```
<class 'pandas.core.frame.DataFrame'>
```

