

Data Visualization – Part 2

Visualize UN Carbon Dioxide Data

This project uses data from United Nation's Data website. The link is:

<https://data.un.org/>

Data: "CO2 Emissions Estimate"

Project Requirement:

1. Load CO2 emission data into data frame.
2. Reorganize the data to show country's CO2 Emission Total and Emission Per Capita.
3. Show the data in different chart types.

Code Reference – Part 1

When using method `read_csv`, we could specify the column names, the “,” in the number as thousand separators, and encoding. Otherwise, the imported data might not have a correct format.

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4
5 columns = ['region_ID', 'Country/Region', 'Year', 'Note', 'Emission_Value']
6 df_emission = pd.read_csv("SYB64_310_202110_Carbon Dioxide Emission Estimates.csv",
7 header = 1, names = columns, thousands = ",", encoding = "iso-8859-1")
8
9 df_emission_per_capita = df_emission.loc[df_emission['Note'].str.contains("Emissions per capita")]
10 df_emission_total = df_emission.loc[~df_emission['Note'].str.contains("Emissions per capita")]
```

Use `.loc` method and conditional filtering to separate the data frame into two ones.

Code Reference – Part 2

```
12 df_merged = pd.merge(df_emission_per_capita, df_emission_total, how = "left",
13 left_on = ['Country/Region', 'Year'], right_on = ['Country/Region', 'Year'],
14 suffixes = ('_per_capita', '_total'))
15
16 df_final = df_merged[['Country/Region', "Year", 'Emission_Value_per_capita',
17 "Emission_Value_total"]]
18
19 print(df_final.head(10))
20
21 # create a category chart: bar chart
22 df_emission_2018 = df_final[df_final["Year"] == 2018].sort_values('Emission_Value_total',
23 ascending = False).head(10)
24 print(df_emission_2018)
```

Pick out several columns to form a more compact data frame.

Merge the two data frames together in column direction, so each country has both CO2 emission total and emission per capita data.

Find out the top 10 countries in terms of emission total value.

Illustration of Result

C:\Python\Python39\python.exe

	Country/Region	Year	Emission_Value_per_capita	Emission_Value_total
0	Albania	1975	1.8	4338.0
1	Albania	1985	2.3	6930.0
2	Albania	2005	1.3	3825.0
3	Albania	2010	1.3	3930.0
4	Albania	2015	1.3	3825.0
5	Albania	2016	1.3	3674.0
6	Albania	2017	1.5	4342.0
7	Albania	2018	1.5	4303.0
8	Algeria	1975	0.8	13553.0
9	Algeria	1985	1.9	42073.0
	Country/Region	Year	Emission_Value_per_capita	Emission_Value_total
181	China	2018	6.8	9528214.0
1035	United States of America	2018	15.0	4921126.0
452	India	2018	1.7	2307780.0
829	Russian Federation	2018	11.0	1587024.0
516	Japan	2018	8.5	1080716.0
380	Germany	2018	8.4	696131.0
809	Republic of Korea	2018	11.7	605775.0
468	Iran (Islamic Republic of)	2018	7.1	579555.0
165	Canada	2018	15.3	565232.0
460	Indonesia	2018	2.0	542876.0

df_final.head(10)

df_emission_2018.head(10)

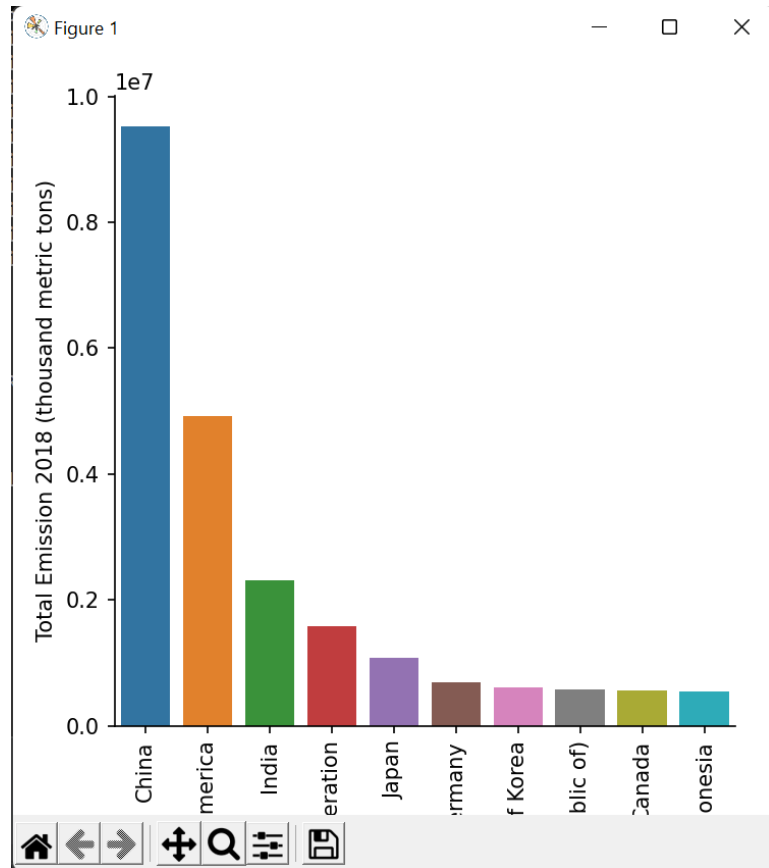
Code Reference – Part 3

```
27 # bar chart
28 sns.catplot(x = "Country/Region", y = "Emission_Value_total", data = df_emission_2018,
29 kind = 'bar')
30 plt.ylabel("Total Emission 2018 (thousand metric tons)")
31 plt.xticks(rotation = 90)
32 plt.show()
33
34 # same as above, but use plt directly.
35 plt.bar(df_emission_2018["Country/Region"], df_emission_2018["Emission_Value_total"],
36 color = 'maroon', width = 0.6)
37 plt.ylabel("Total Emission 2018 (thousand metric tons)")
38 plt.xticks(rotation = 90)
39 plt.show()
40
41 sns.barplot(x = "Emission_Value_total", y = "Country/Region", data= df_emission_2018)
42 plt.show()
```

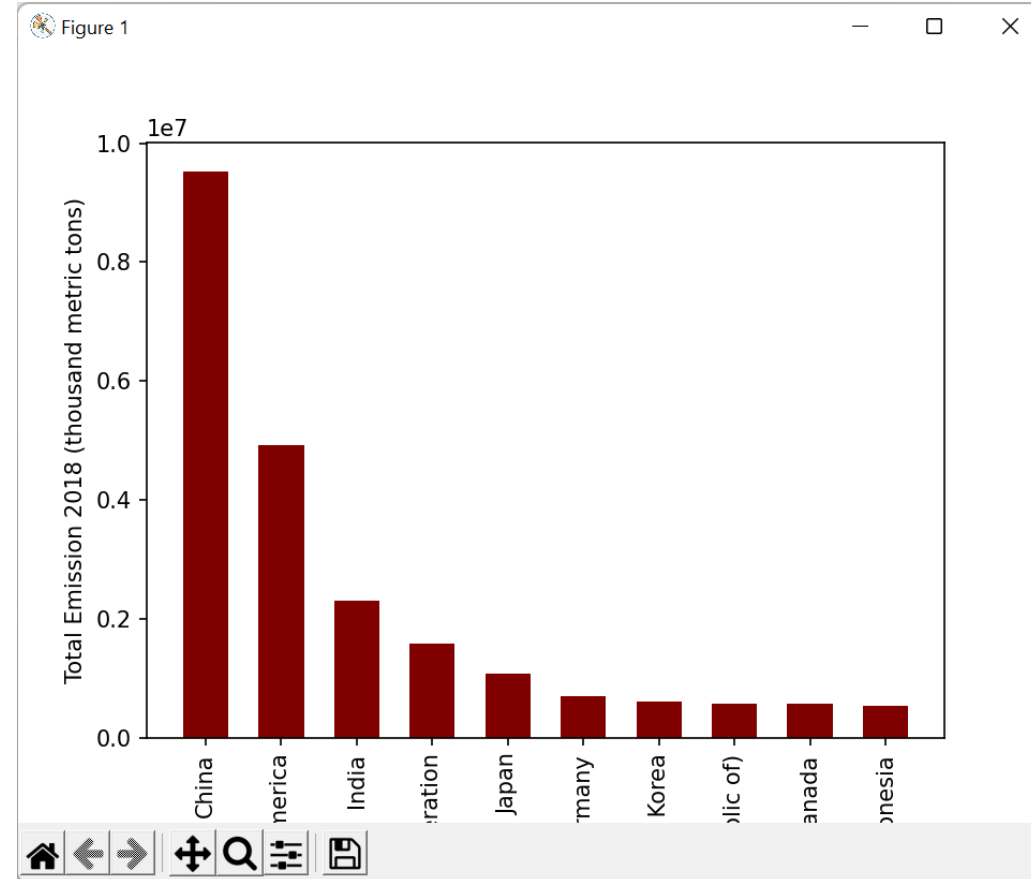
Use seaborn's catplot() method to draw the bar chart.

Use matplotlib's bar() method to draw the bar chart.

Illustration

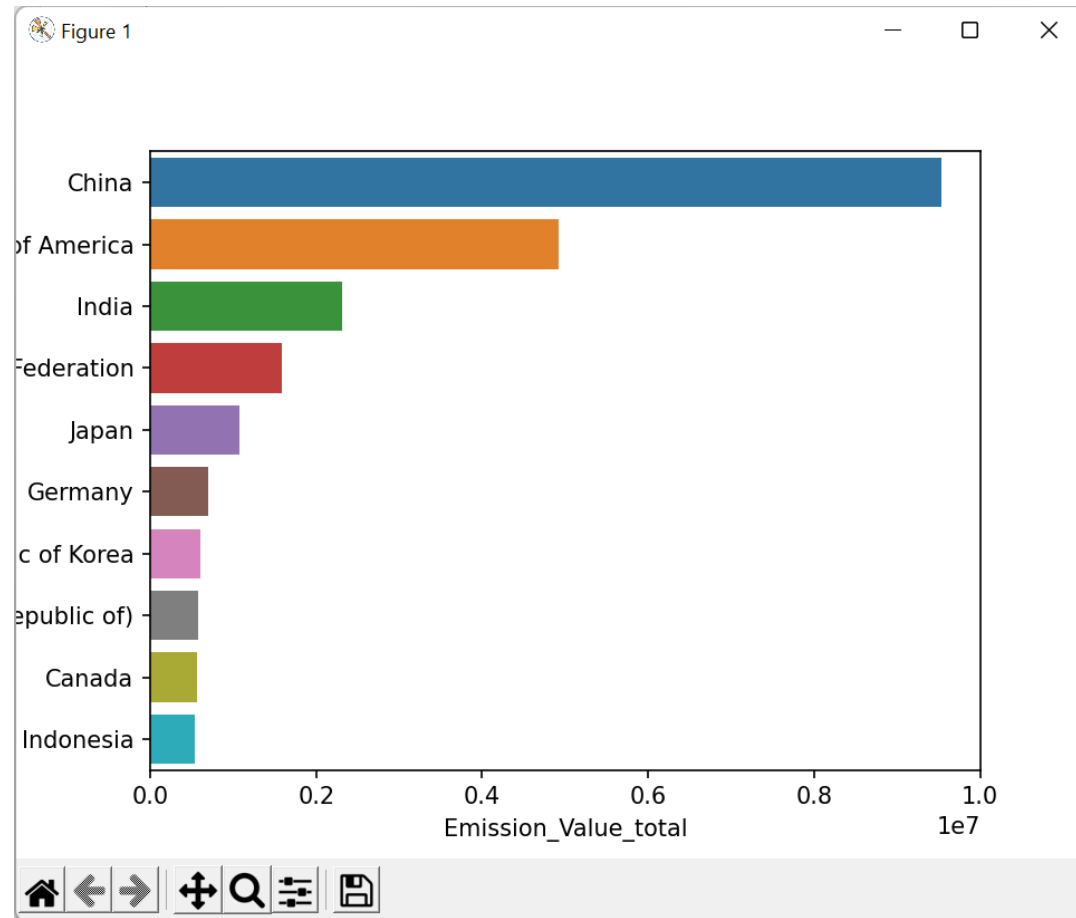


Use seaborn's catplot to draw the top 10 countries of total CO2 emission.



Use matplotlib's bar() method to draw the top 10 countries of total CO2 emission.

Illustration



Use seaborn's `barplot()` method to draw horizontal bars.

Code Reference – Part 4

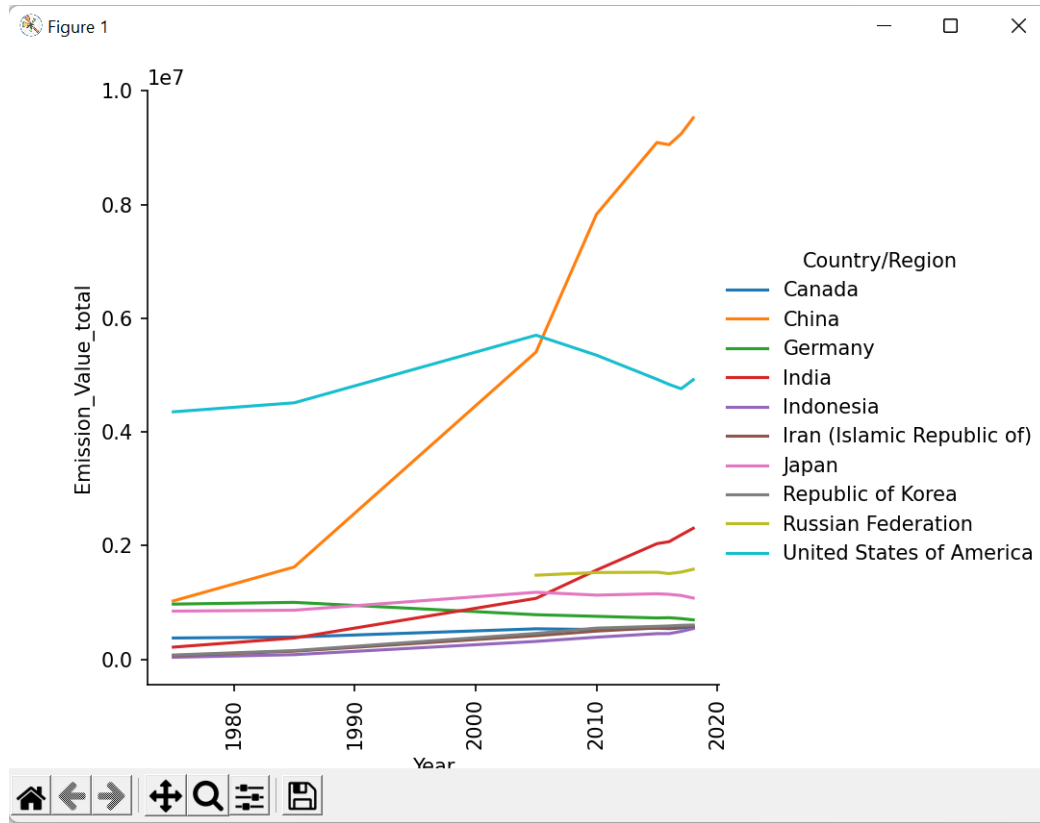
Pick out the annual emission data for those top 10 countries of total emission value.

```
45 top_emission_countries = df_emission_2018['Country/Region'].tolist()
46 df_emission_top_countries = df_final[df_final['Country/Region'].isin(top_emission_countries)]
47 sns.relplot(x = "Year", y = "Emission_Value_total", data = df_emission_top_countries,
48 kind = "line", hue = "Country/Region")
49 plt.xticks(rotation = 90)
50 plt.show()
51
52 # use matplotlib to do it directly
53 df_china_emission = df_emission_top_countries.loc[df_emission_top_countries['Country/Region'] == 'China']
54 df_usa_emission = df_emission_top_countries.loc[df_emission_top_countries['Country/Region'] == 'United States']
55 plt.bar(df_china_emission["Year"]-0.5, df_china_emission["Emission_Value_total"],
56 color = 'r', width = 0.5)
57 plt.bar(df_usa_emission["Year"], df_usa_emission["Emission_Value_total"],
58 color = 'b', width = 0.5)
59 plt.show()
```

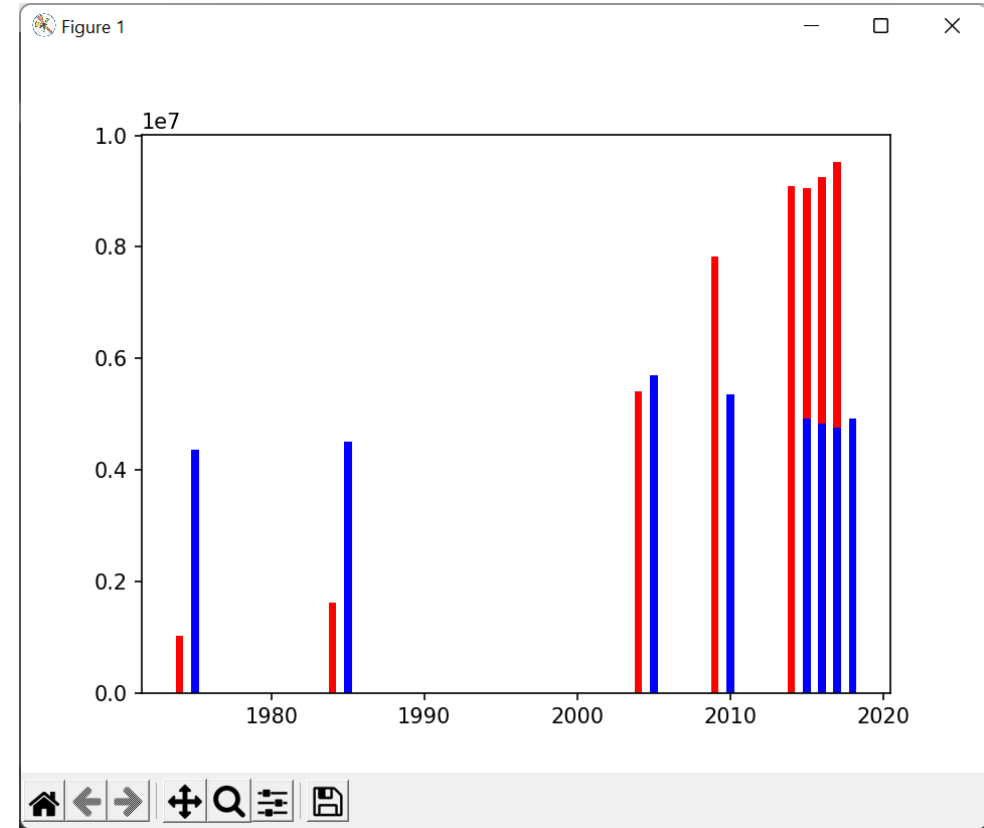
Use seaborn's relplot() method to draw line chart. Please note the "kind" parameter should be "line".

Use matplotlib's bar() method to draw side-by-side bar chart.

Illustration



Use `sns.catplot(... kind= "line" ...)` to draw line chart.



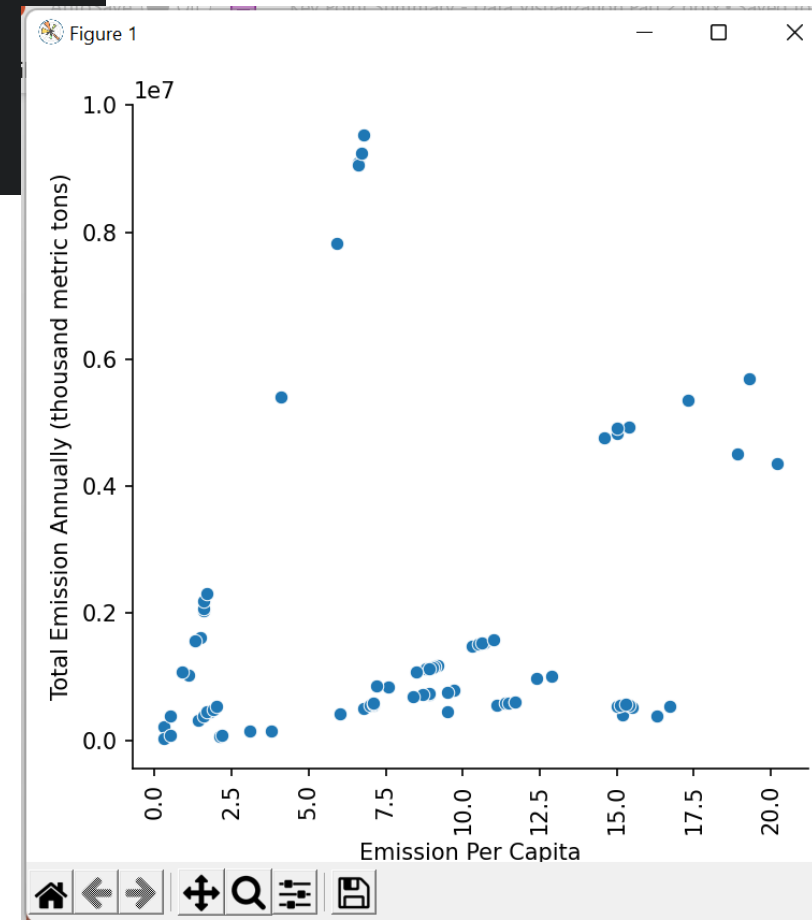
Use `plt.bar` method to draw side-by-side bar chart.

Code Reference – Part 5

```
61 # create scatter chart
62 sns.relplot(x = "Emission_Value_per_capita", y = "Emission_Value_total",
63 data = df_emission_top_countries, kind = "scatter")
64 plt.xticks(rotation = 90)
65 plt.xlabel("Emission Per Capita")
66 plt.ylabel("Total Emission Annually (thousand metric tons)")
67 plt.show()
```

Use seaborn's relplot() method to draw scatter chart by setting "kind" parameter to "scatter".

The chart shows that Emission Total and Emission Per Capita has not apparent linear relationship.



**Enjoy the Coding
and Have Fun!**