Пошуковий аналіз даних

Ознайомитись з методами перевірки статистичних гіпотез. Після завершення цієї лабораторної роботи ви зможете:

- Досліджувати дані за допомогою візуалізацій
- Робити описовий аналіз
- Групувати дані для аналізу
- Знаходити зв'язок між ознаками
- Перевіряти гіпотези про значущість коефіцієнта кореляції та про вигляд закону розподілу
- Робити дисперсійний аналіз

1. Скачати дані із файлу 'clean_data2.csv', який зберегли наприкінці попередньої роботи (Data2.csv з виправленими помилками та заповненими пропусками). Записати дані у dataframe. Дослідити ознаки, побудувавши їх візуалізації
2. Порахувати кореляцію між всіма кількісними ознаками
3. Побудувати діаграми розсіювання для кількісних ознак та 'CO2 emission'. Побудувати діаграму розмаху для 'CO2 emission' по регіонам. Візуально оцініть наявність та силу зв'язку між цими ознаками.

4. Які кількісні ознаки можуть бути предикторами кількості викидів CO2?
5. Виконати дисперсійний аналіз для кількості викидів CO2, згрупувати дані по регіонам

Завдання #1:

Зчитую дані з файлу у датафрейм

```
# Напишіть ваш код нижче та натисніть Shift+Enter для виконання
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

df = pd.read_csv("clean_data2.csv", encoding='cp1252')
print(df)
```

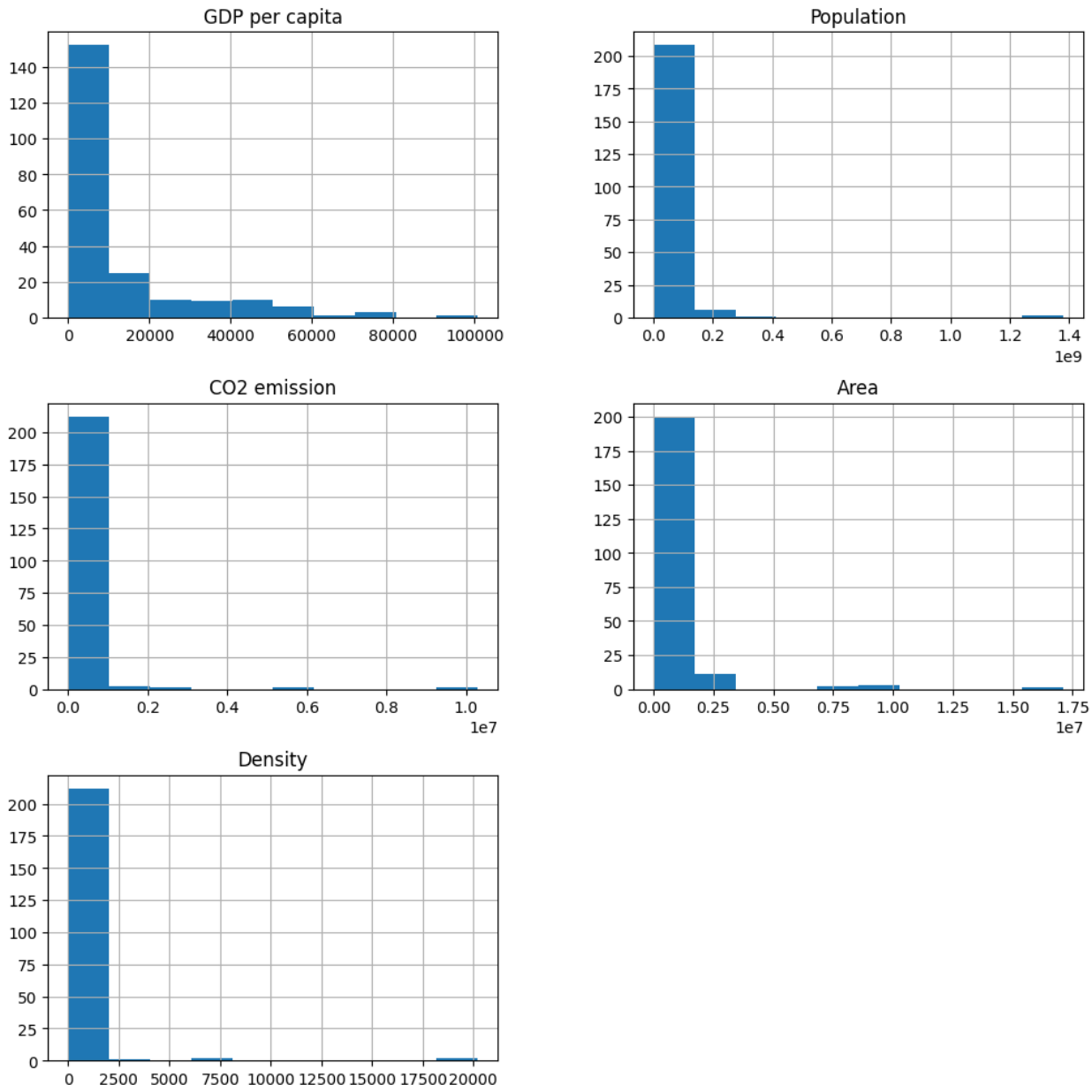
```
              Country Name                     Region  GDP per capita
\
0              Afghanistan                 South Asia      561.778746

1                  Albania       Europe & Central Asia     4124.982390
```

```
2                  Algeria  Middle East & North Africa     3916.881571

3           American Samoa         East Asia & Pacific    11834.745230

4                  Andorra       Europe & Central Asia    36988.622030

..                     ...                         ...             ...

212  Virgin Islands (U.S.)   Latin America & Caribbean        6.327732

213     West Bank and Gaza  Middle East & North Africa     2943.404534

214            Yemen, Rep.  Middle East & North Africa      990.334774

215                 Zambia          Sub-Saharan Africa     1269.573537

216               Zimbabwe          Sub-Saharan Africa     1029.076649


     Population   CO2 emission       Area     Density
0      34656032    9809.225000   652860.0   53.083405
1       2876101    5716.853000    28750.0  100.038296
2      40606052  145400.217000  2381740.0   17.048902
3         55599      31.100793      200.0  277.995000
4         77281     462.042000      470.0  164.427660
..          ...            ...        ...         ...
212      102951      57.577071      350.0  294.145714
213     4551566    2540.270209     6020.0  756.074086
214    27584213   22698.730000   527970.0   52.245796
215    16591390    4503.076000   752610.0   22.045136
216    16150362   12020.426000   390760.0   41.330643

[217 rows x 7 columns]
```

Будую графіки

```
# Напишіть ваш код нижче та натисніть Shift+Enter для виконання
df.hist(figsize=(12, 12))
plt.suptitle('Гістограми')
plt.show()
```

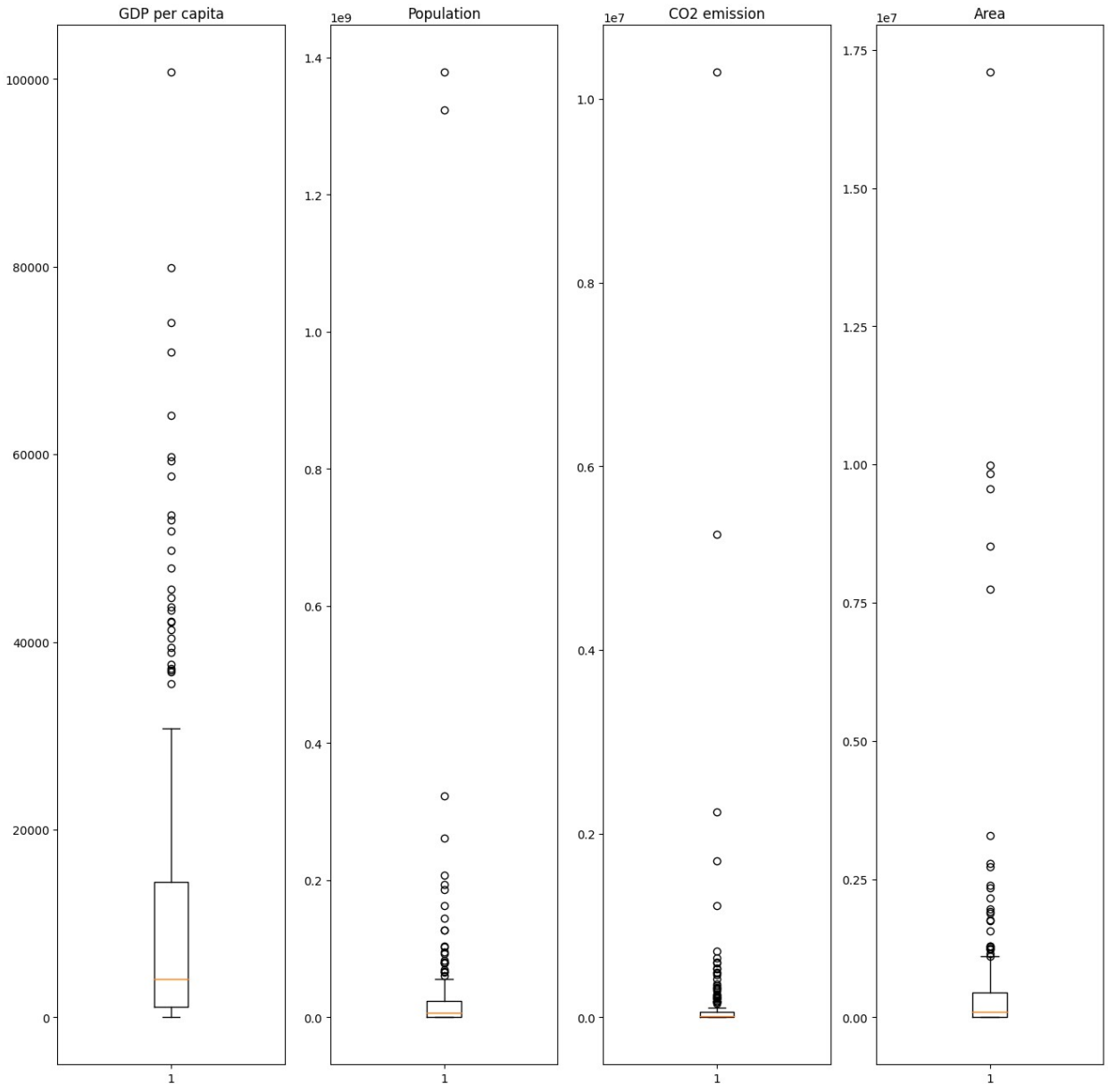
Гістограми




```
fig, axs = plt.subplots(1, 4, figsize=(16, 16))
fig.suptitle('Діаграми розмаху', fontsize=16)
axs[0].set_title('GDP per capita')
axs[0].boxplot(df['GDP per capita'])
axs[1].set_title('Population')
axs[1].boxplot(df['Population'])
axs[2].set_title('CO2 emission')
axs[2].boxplot(df['CO2 emission'])
```

```
axs[3].set_title('Area')
axs[3].boxplot(df['Area'])

plt.show()
```

Діаграми розмаху



Завдання #2:

Рахую кореляцію між всіма кількісними ознаками

```
# Напишіть ваш код нижче та натисніть Shift+Enter для виконання
df.iloc[:, 2:].corr()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 5,\n  \"fields\": [\n
{\n      \"column\": \"GDP per capita\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.41472330275464725,\n
\"min\": -0.025193068178011636,\n        \"max\": 1.0,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          -
0.025193068178011636,\n          0.20369894612587758,\n
0.0997205691937224\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.46960814377433235,\n        \"min\": -
0.0282322394377628,\n        \"max\": 1.0,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          1.0,\n
-0.0282322394377628,\n          0.8042756753281473\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"CO2 emission\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.44359901700443544,\n        \"min\": -0.025994744487178145,\n
\"max\": 1.0,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          0.8042756753281473,\n          -0.025994744487178145,\n
1.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Area\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0.4221393478914457,\n        \"min\": -0.06396864925528256,\n
\"max\": 1.0,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          0.45373505170699885,\n          -0.06396864925528256,\n
0.5886823760434484\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Density\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 0.4503872073954158,\n        \"min\": -
0.06396864925528256,\n        \"max\": 1.0,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          -
0.0282322394377628,\n          1.0,\n          -0.025994744487178145\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```


```
<google.colab._quickchart_helpers.SectionTitle at 0x7aa41d1725c0>
```

```
from matplotlib import pyplot as plt
_df_12['GDP per capita'].plot(kind='hist', bins=20, title='GDP per
capita')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_13['Population'].plot(kind='hist', bins=20, title='Population')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_14['CO2 emission'].plot(kind='hist', bins=20, title='CO2
emission')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_15['Area'].plot(kind='hist', bins=20, title='Area')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7aa41d170820>
```

```
from matplotlib import pyplot as plt
_df_16.plot(kind='scatter', x='GDP per capita', y='Population', s=32,
alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_17.plot(kind='scatter', x='Population', y='CO2 emission', s=32,
alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_18.plot(kind='scatter', x='CO2 emission', y='Area', s=32,
alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_19.plot(kind='scatter', x='Area', y='Density', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7aa416472650>
```

```
from matplotlib import pyplot as plt
_df_20['GDP per capita'].plot(kind='line', figsize=(8, 4), title='GDP
per capita')
plt.gca().spines[['top', 'right']].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_21['Population'].plot(kind='line', figsize=(8, 4),
title='Population')
plt.gca().spines[['top', 'right']].set_visible(False)
```

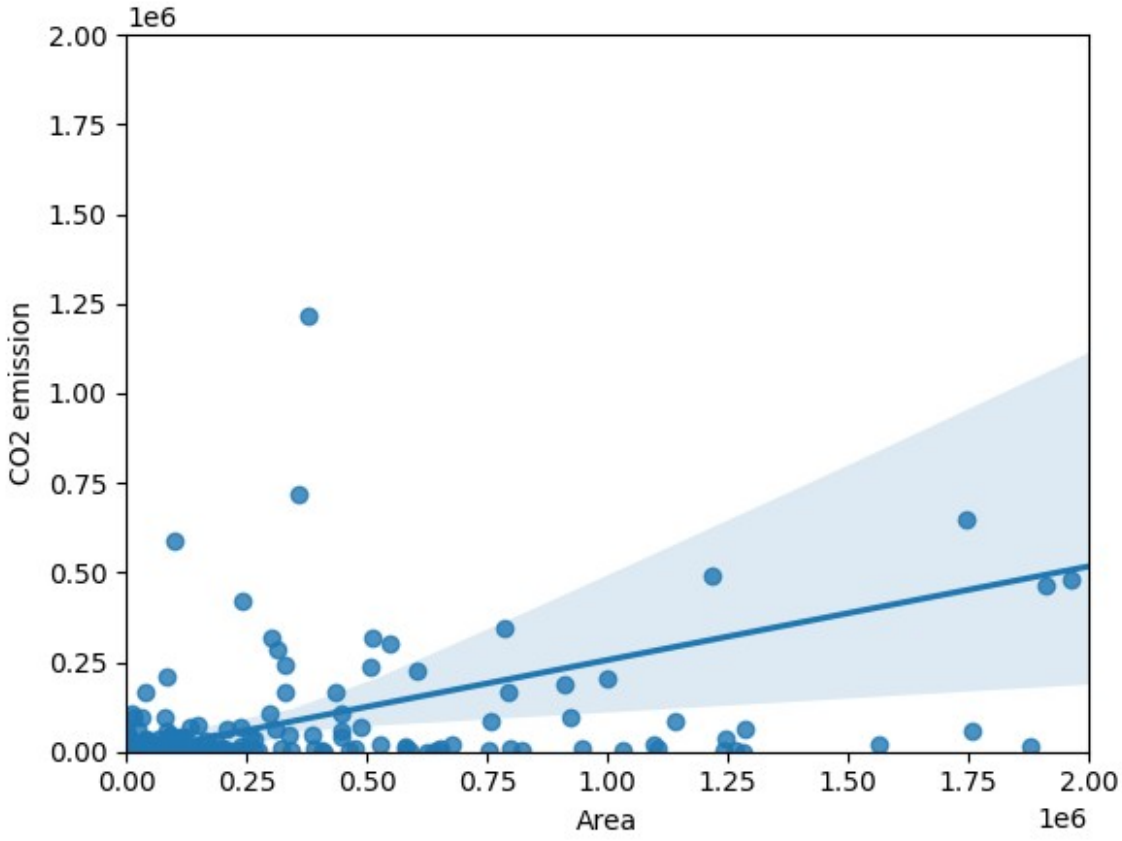
```
from matplotlib import pyplot as plt
_df_22['CO2 emission'].plot(kind='line', figsize=(8, 4), title='CO2
emission')
plt.gca().spines[['top', 'right']].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_23['Area'].plot(kind='line', figsize=(8, 4), title='Area')
plt.gca().spines[['top', 'right']].set_visible(False)
```

Завдання #3:

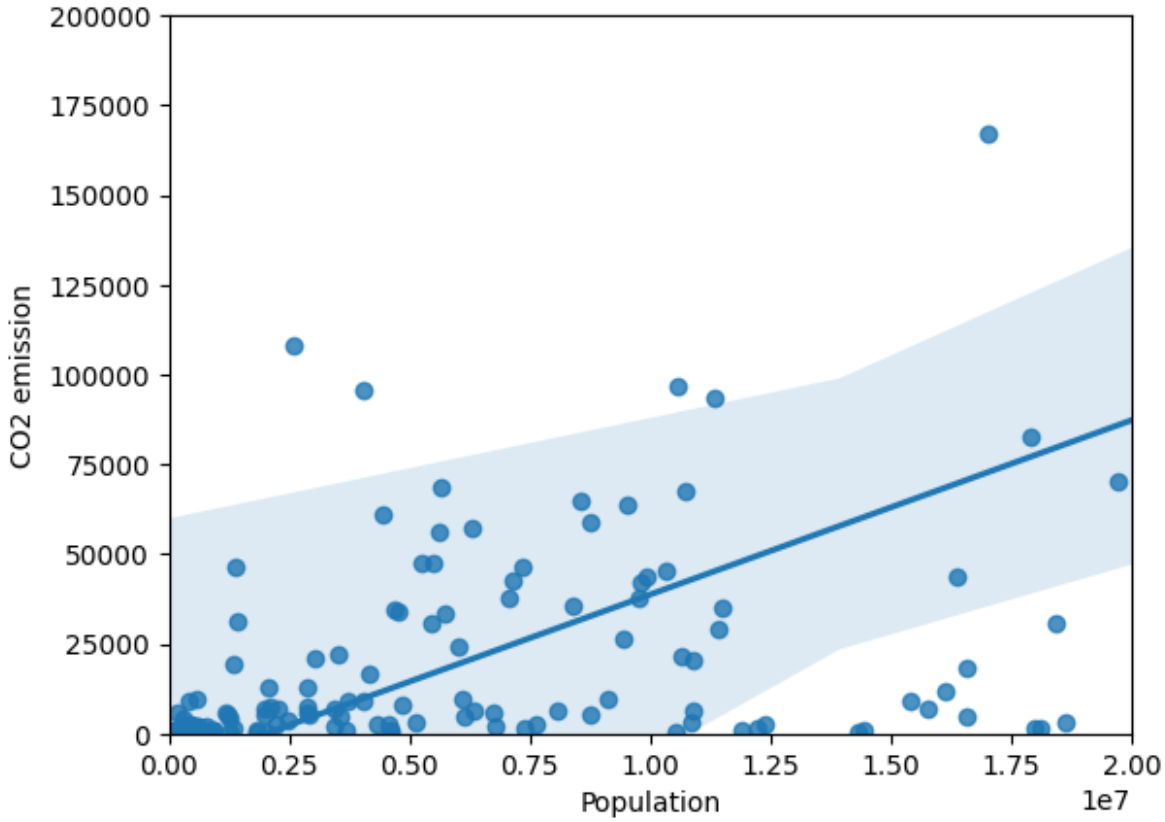
Будую діаграму розсіювання для кількісних ознак та 'CO2 emission'

```
# Напишіть ваш код нижче та натисніть Shift+Enter для виконання

axes = sns.regplot(x=df['Area'], y=df['CO2 emission'])
axes.set_ylim(0, 2000000)
axes.set_xlim(0, 2000000)

plt.show()
```

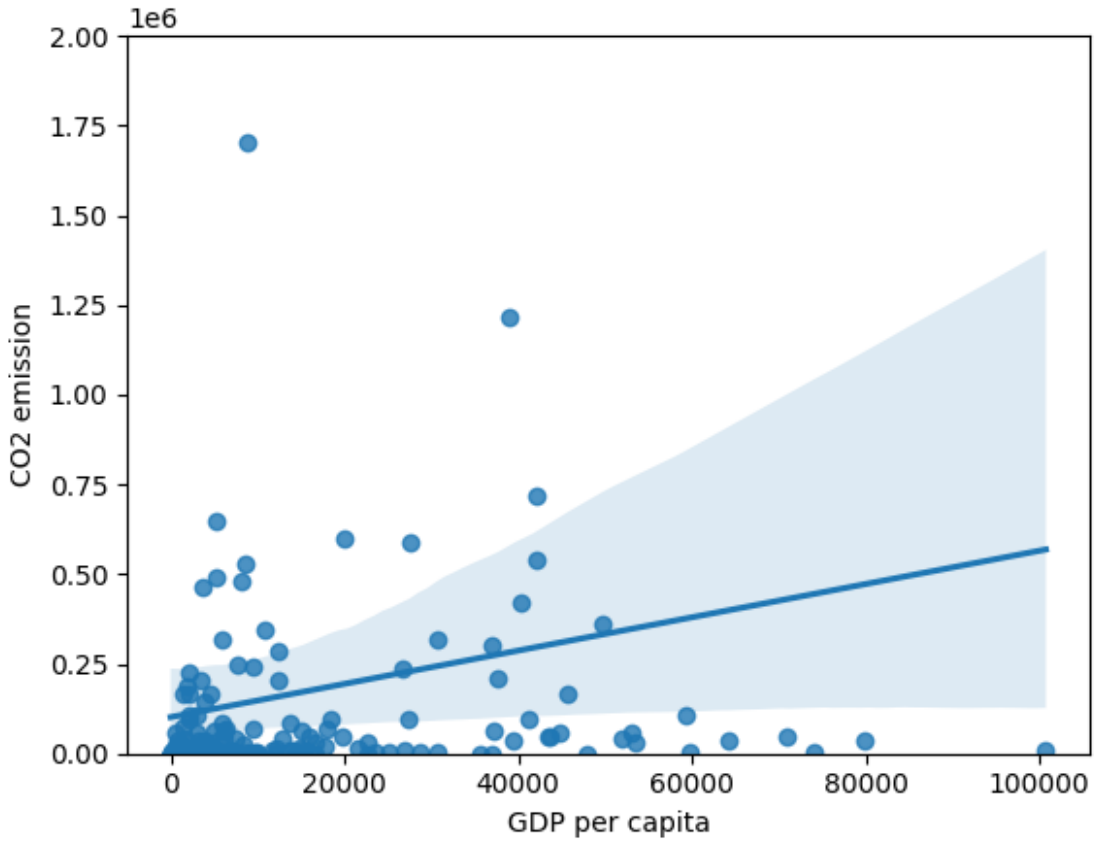



```
axes = sns.regplot(x=df['Population'], y=df['CO2 emission'])

axes.set_ylim(0, 200000)
axes.set_xlim(0, 20000000)
```

```
plt.show()
```




```
axes = sns.regplot(x=df['GDP per capita'], y=df['CO2 emission'])

axes.set_ylim(0, 2000000)
# axes.set_xlim(0, 200000)

plt.show()
```



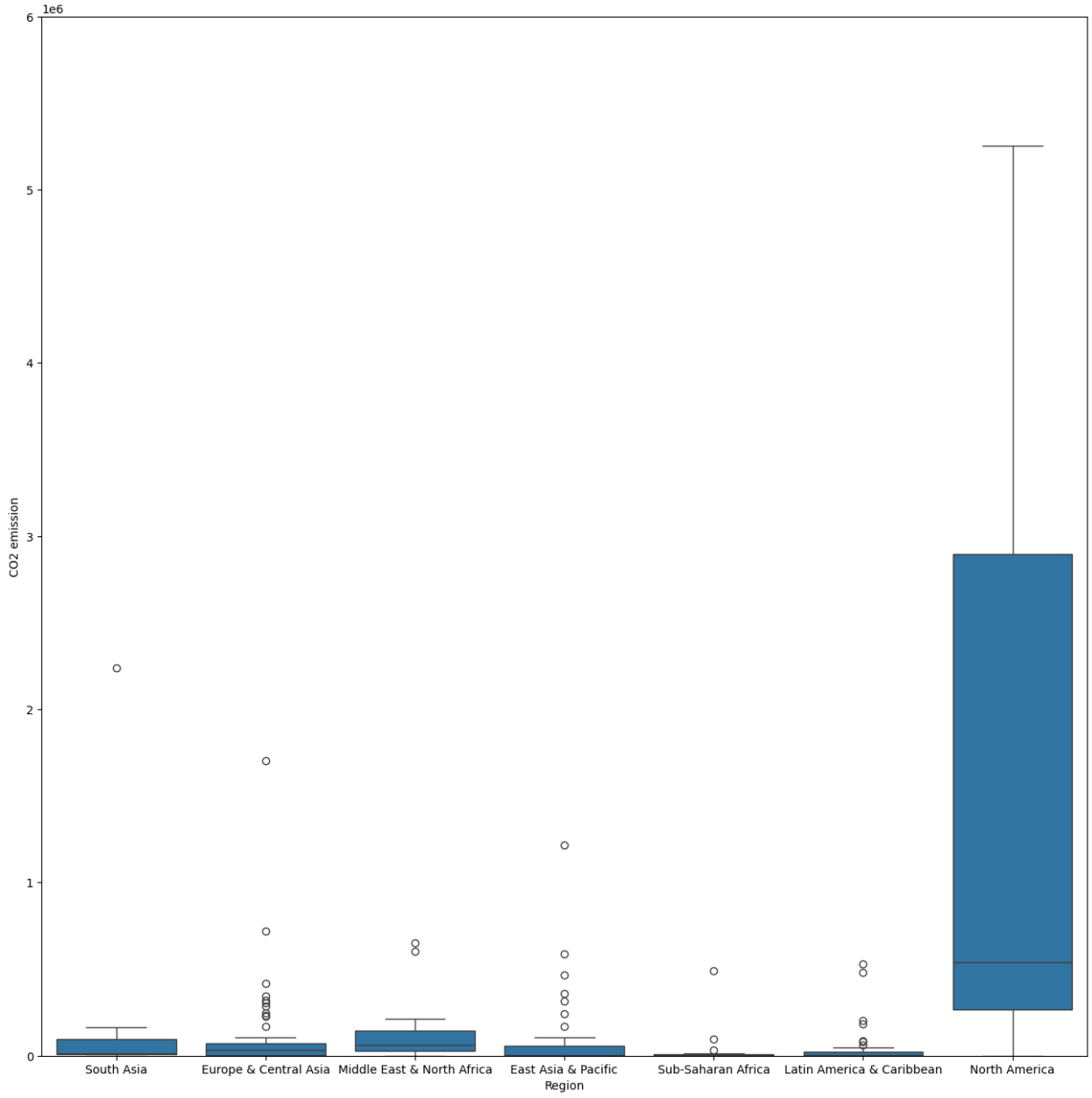
Візуально помітний прямий зв'язок між CO2 emission та іншими кількісними ознаками.

Найсильніший зв'язок між CO2 emission та Population

Будую діаграму розмаху для 'CO2 emission' по регіонам

```
# Напишіть ваш код нижче та натисніть Shift+Enter для виконання
plt.figure(figsize=(16, 16))

axes = sns.boxplot(x='Region', y='CO2 emission', data=df)

axes.set_ylim(0, 6000000)

plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Завдання #4:

Обчислюю коефіцієнт кореляції Пірсона та P-value для всіх кількісних змінних та 'CO2 emission'

```
from scipy import stats

# Напишіть ваш код нижче та натисніть Shift+Enter для виконання
# data =
pearson_area, p_value_area = stats.pearsonr(df['Area'], df['CO2
```

```
emission'])
pearson_population, p_value_population =
stats.pearsonr(df['Population'], df['CO2 emission'])
pearson_gdp, p_value_gdp = stats.pearsonr(df['GDP per capita'],
df['CO2 emission'])

df_corr = pd.DataFrame({'Variable': ['Area', 'Population', 'GDP per
capita'],
                        'Pearson': [pearson_area, pearson_population,
pearson_gdp],
                        'p_value': [p_value_area, p_value_population,
p_value_gdp]})

print(df_corr)
```

```
         Variable   Pearson       p_value
0            Area  0.588682  1.250796e-21
1      Population  0.804276  1.710419e-50
2  GDP per capita  0.099721  1.431548e-01
```

Оскільки p-value < 0.001 для всіх кількісних ознак, кореляція між рівнем викиду CO2 та площею/кількістю населення/ВВП на душу населення є статистично значущою.

Лінійний зв'язок між CO2 emission та Area (~0.588) помірний.

Лінійний зв'язок між CO2 emission та Area (~0.804) дуже сильний.

Лінійний зв'язок між CO2 emission та Area (~0.099) дуже слабкий.

Завдання #5:

Групую дані, щоб побачити чи впливає 'Region' на 'CO2 emission'.

```
# Напишіть ваш код нижче та натисніть Shift+Enter для виконання
grouped=df[['Region', 'CO2 emission']].groupby(['Region'])

print(grouped.head(2), '\n\n')

grouped.get_group('South Asia')['CO2 emission']
```

```
                        Region   CO2 emission
0                   South Asia    9809.225000
1        Europe & Central Asia    5716.853000
2   Middle East & North Africa  145400.217000
3          East Asia & Pacific      31.100793
4        Europe & Central Asia     462.042000
5           Sub-Saharan Africa   34763.160000
6    Latin America & Caribbean     531.715000
7    Latin America & Caribbean  204024.546000
```

```
10         East Asia & Pacific  361261.839000
14  Middle East & North Africa   31338.182000
15                South Asia   73189.653000
20          Sub-Saharan Africa    6318.241000
21               North America     575.719000
35               North America  537193.498000
```

```
0         9809.225
15       73189.653
22        1001.091
88     2238377.137
121       1334.788
138       8030.730
148     166298.450
179      18393.672
Name: CO2 emission, dtype: float64
```

Скористаюсь функцією f_oneway з модуля "stats" для отримання F-test score та P-value.

```
print('\n\nУнікальні записи у полі Region:\n', df['Region'].unique())
```

```
Унікальні записи у полі Region:
 ['South Asia' 'Europe & Central Asia' 'Middle East & North Africa'
 'East Asia & Pacific' 'Sub-Saharan Africa' 'Latin America &
Caribbean'
 'North America']
```

```
# Напишіть ваш код нижче та натисніть Shift+Enter для виконання
f_val, p_val = stats.f_oneway(grouped.get_group('South Asia')['CO2
emission'],
                              grouped.get_group('Europe & Central
Asia')['CO2 emission'],
                              grouped.get_group('Middle East & North
Africa')['CO2 emission'],
                              grouped.get_group('East Asia & Pacific')
['CO2 emission'],
                              grouped.get_group('Sub-Saharan Africa')
['CO2 emission'],
                              grouped.get_group('Latin America &
Caribbean')['CO2 emission'],
                              grouped.get_group('North America')['CO2
emission'],)

print("F-statistic:", f_val)
print("p-value:", p_val)
```

```
F-statistic: 3.5508824714043836
p-value: 0.002270432690210372
```

p-value < 0.05, це означає що оцінка є статистично значущою, а F показник вказує на (насправді не дуже великий) рівень кореляції.

Але чи означає це, що всі групи корелюють між собою?

Розглянемо їх окремо.

```
# Напишіть ваш код нижче та натисніть Shift+Enter для виконання

regions = df['Region'].unique()

for i in range(len(regions)-1):
  for j in range(i+1, len(regions)):
    f_val, p_val = stats.f_oneway(grouped.get_group(regions[i])['CO2
emission'],
                                  grouped.get_group(regions[j])['CO2
emission'])
    if (p_val>0.1):
      print('!!!!!!!!!!!!') #відмічено завелике значення p
    print(f"Між {regions[i]} і {regions[j]}\n"
          f"F-statistic: {f_val}\n"
          f"p-value: {p_val}\n")

!!!!!!!!!!
Між South Asia і Europe & Central Asia
F-statistic: 2.4563284325681862
p-value: 0.12198356892674468

!!!!!!!!!!
Між South Asia і Middle East & North Africa
F-statistic: 1.1676868056769816
p-value: 0.28943512933225635

!!!!!!!!!!
Між South Asia і East Asia & Pacific
F-statistic: 0.010779237880269849
p-value: 0.9177924654224876

Між South Asia і Sub-Saharan Africa
F-statistic: 7.300524087267293
p-value: 0.009192711202178152

Між South Asia і Latin America & Caribbean
F-statistic: 4.901414243471789
p-value: 0.031619114432338585

!!!!!!!!!!
```

```
Mіж South Asia і North America
F-statistic: 2.4462636855311657
p-value: 0.1522433374176963

!!!!!!!!!!
Mіж Europe & Central Asia і Middle East & North Africa
F-statistic: 0.07107110672565103
p-value: 0.7904962203272123

!!!!!!!!!!
Mіж Europe & Central Asia і East Asia & Pacific
F-statistic: 1.4465924181848755
p-value: 0.23212866889855782

Mіж Europe & Central Asia і Sub-Saharan Africa
F-statistic: 5.842896019941366
p-value: 0.01738064211148622

!!!!!!!!!!
Mіж Europe & Central Asia і Latin America & Caribbean
F-statistic: 2.2613927410805226
p-value: 0.1358506720763068

Mіж Europe & Central Asia і North America
F-statistic: 27.542942507369514
p-value: 2.192558279480627e-06

!!!!!!!!!!
Mіж Middle East & North Africa і East Asia & Pacific
F-statistic: 0.47071503086045974
p-value: 0.49548927919429053

Mіж Middle East & North Africa і Sub-Saharan Africa
F-statistic: 12.667677209560903
p-value: 0.0006894524567808801

Mіж Middle East & North Africa і Latin America & Caribbean
F-statistic: 4.508747692054755
p-value: 0.037786578610934526

Mіж Middle East & North Africa і North America
F-statistic: 10.870039824858475
p-value: 0.0032855685573012923

!!!!!!!!!!
Mіж East Asia & Pacific і Sub-Saharan Africa
F-statistic: 2.195045163818428
p-value: 0.1422417679455547

!!!!!!!!!!
```

```
Між East Asia & Pacific і Latin America & Caribbean
F-statistic: 1.6264217369678258
p-value: 0.2060330582252482

!!!!!!!!!!
Між East Asia & Pacific і North America
F-statistic: 2.1223376115912314
p-value: 0.15337905025775542

!!!!!!!!!!
Між Sub-Saharan Africa і Latin America & Caribbean
F-statistic: 2.0307483518164124
p-value: 0.15768181172333404

Між Sub-Saharan Africa і North America
F-statistic: 29.889184991922956
p-value: 1.5319132629213873e-06

Між Latin America & Caribbean і North America
F-statistic: 24.830715550606573
p-value: 1.0700462806894996e-05
```

Між групами відміченими знаками оклику значення p > 0.1, тому між цими групами розрахований показник F не є статистично значущим.

Додаткове завдання:

1. По результатам дисперсійного аналізу для кількості викидів CO2 по регіонам, вкажіть пару регіонів, що відрізняються найсильніше.
2. Створіть якісну ознаку 'Rich country', згрупувавши дані 'GDP per capita' в кілька категорій (багаті-бідні країни, 3-5 категорій). Побудуйте діаграму розмаху для 'CO2 emission' по категоріям 'Rich country'. Візуально оцініть наявність зв'язку між цими ознаками.
3. Виконайте дисперсійний аналіз для 'CO2 emission', згрупувавши дані по категоріям 'Rich country'.

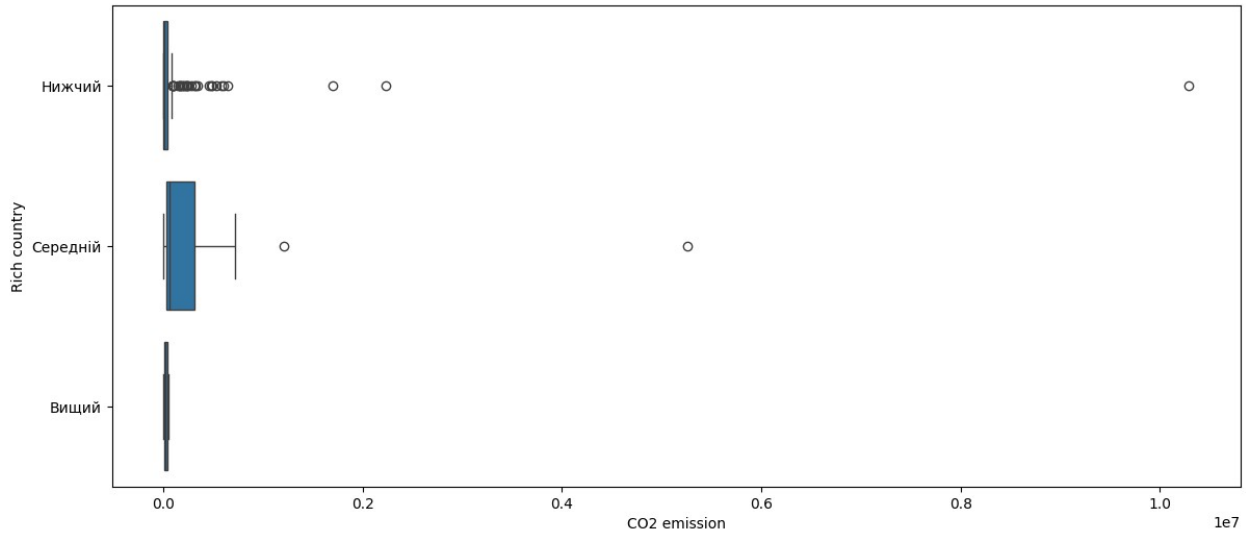
1) Щоб знайти пару регіонів, що відрозняються найбільше, упустимо пари у яких показник F не є статистично значущим (p > 0.1).

```
#1
max_f = 0

for i in range(len(regions)-1):
  for j in range(i+1, len(regions)):
    f_val, p_val = stats.f_oneway(grouped.get_group(regions[i])['CO2
emission'],
                                  grouped.get_group(regions[j])['CO2
```

```
emission'])
    if (p_val<0.1 and f_val > max_f):
      max_reg = [regions[i], regions[j]]
      max_f = f_val

print(f"Між регіонами {max_reg} найбільша різниця, що складає:
{max_f}")
```

```
Між регіонами ['Sub-Saharan Africa', 'North America'] найбільша
різниця, що складає: 29.889184991922956
```

```
#2
df['Rich country'] = pd.cut(df['GDP per capita'], bins=3,
labels=['Нижчий', 'Середній', 'Вищий'])

plt.figure(figsize=(14, 6))
sns.boxplot(x='CO2 emission', y='Rich country', data=df)
plt.show()
```



Візуально присутня різниця між першим і третім квартилями (особливо тут виділяється середній рівень багатства), а також максимальними і мінімальними значеннями (вусиками коробки), однак медіанні значення для всіх країн знаходяться на одному рівні.

```
#3

unique_categories = df['Rich country'].unique()

data_by_category = (df[df['Rich country'] == category]['CO2 emission']
for category in unique_categories)

f_val, p_val = stats.f_oneway(*data_by_category)
```

```
print("F-statistic:", f_val)
print("p-value:", p_val)
```

```
F-statistic: 1.3471041186853305
p-value: 0.2621876608725918
```

Значення p є занадто великим, тож кореляція не є статистично значущою.