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

```
import pandas as pd

df = pd.read_csv("../Data2-clean.csv", sep=';' ,encoding='cp1252')
df
```

```
    Country Name                        Region  GDP per capita
Population  \
0    Afghanistan                    South Asia      561.778746
34656032.0
1        Albania       Europe & Central Asia     4124.982390
2876101.0
2        Algeria  Middle East & North Africa     3916.881571
40606052.0
3        Andorra       Europe & Central Asia    36988.622030
77281.0
4         Angola          Sub-Saharan Africa     3308.700233
```

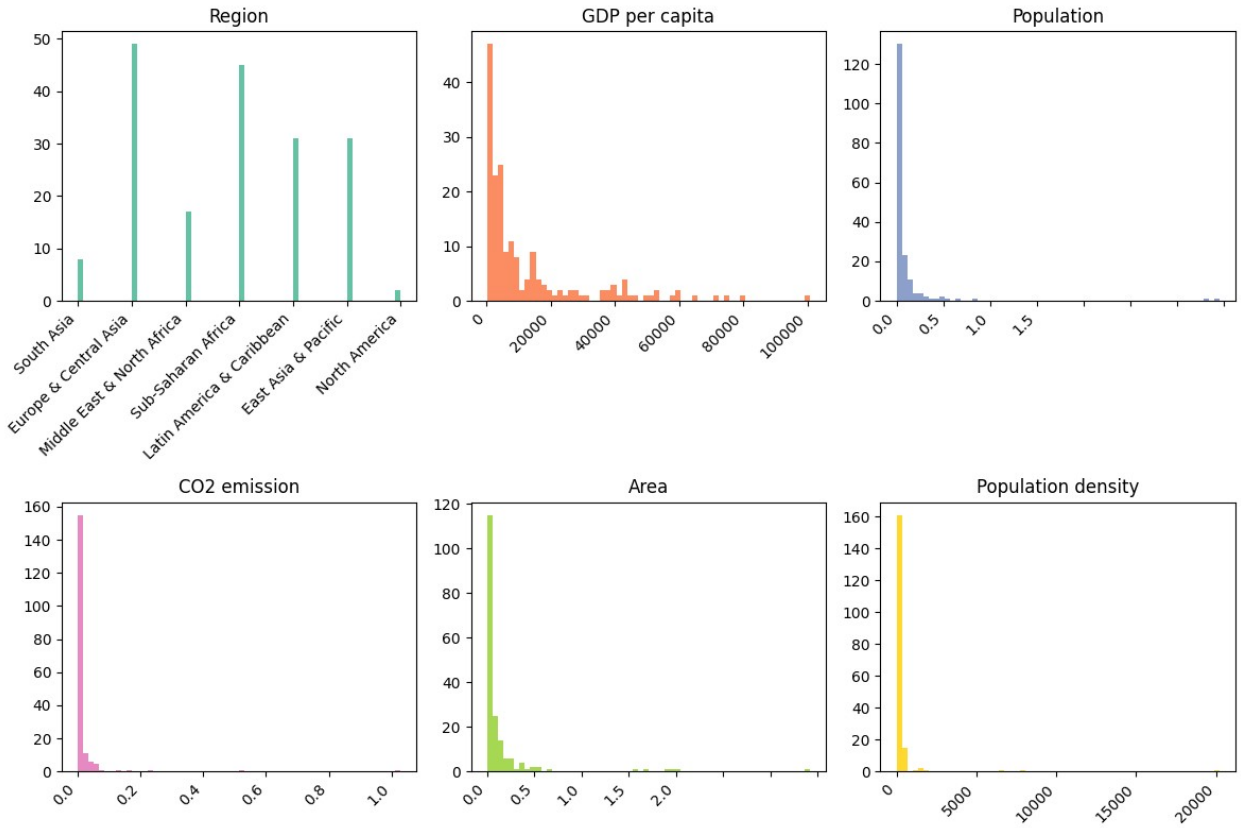
```
28813463.0
..           ...                           ...             ...          .
..
178      Vanuatu         East Asia & Pacific     2860.566475
270402.0
179      Vietnam         East Asia & Pacific     2170.648054
92701100.0
180  Yemen, Rep.  Middle East & North Africa      990.334774
27584213.0
181       Zambia          Sub-Saharan Africa     1269.573537
16591390.0
182     Zimbabwe          Sub-Saharan Africa     1029.076649
16150362.0

     CO2 emission       Area  Population density
0         9809.225   652860.0           53.083405
1         5716.853    28750.0          100.038296
2       145400.217  2381740.0           17.048902
3          462.042      470.0          164.427660
4        34763.160  1246700.0           23.111786
..             ...        ...                 ...
178        154.014    12190.0           22.182281
179     166910.839   330967.0          280.091671
180      22698.730   527970.0           52.245796
181       4503.076   752610.0           22.045136
182      12020.426   390760.0           41.330643

[183 rows x 7 columns]

df.dtypes

Country Name           object
Region                 object
GDP per capita        float64
Population            float64
CO2 emission          float64
Area                  float64
Population density    float64
dtype: object
```

Будую графіки

```
import matplotlib.pyplot as plt
import seaborn as sns

fig, axes = plt.subplots(2, 3, figsize=(12, 8))
clrs = sns.color_palette("Set2", 7)

for i, col in enumerate(df.columns[1:]):
    row = i // 3
```

```
    col_index = i % 3
    axes[row, col_index].hist(df[col], bins=60, color=clrs[i])
    axes[row, col_index].set_title(col)
    labels = axes[row, col_index].get_xticklabels()
    axes[row, col_index].set_xticklabels(labels, rotation=45,
ha='right')

plt.tight_layout()
```

```
C:\Users\local_gud2i5y\AppData\Local\Temp\
ipykernel_24976\3045389681.py:13: UserWarning: set_ticklabels() should
only be used with a fixed number of ticks, i.e. after set_ticks() or
using a FixedLocator.
  axes[row, col_index].set_xticklabels(labels, rotation=45,
ha='right')
C:\Users\local_gud2i5y\AppData\Local\Temp\
ipykernel_24976\3045389681.py:13: UserWarning: set_ticklabels() should
only be used with a fixed number of ticks, i.e. after set_ticks() or
using a FixedLocator.
  axes[row, col_index].set_xticklabels(labels, rotation=45,
ha='right')
C:\Users\local_gud2i5y\AppData\Local\Temp\
ipykernel_24976\3045389681.py:13: UserWarning: set_ticklabels() should
only be used with a fixed number of ticks, i.e. after set_ticks() or
using a FixedLocator.
  axes[row, col_index].set_xticklabels(labels, rotation=45,
ha='right')
C:\Users\local_gud2i5y\AppData\Local\Temp\
ipykernel_24976\3045389681.py:13: UserWarning: set_ticklabels() should
only be used with a fixed number of ticks, i.e. after set_ticks() or
using a FixedLocator.
  axes[row, col_index].set_xticklabels(labels, rotation=45,
ha='right')
C:\Users\local_gud2i5y\AppData\Local\Temp\
ipykernel_24976\3045389681.py:13: UserWarning: set_ticklabels() should
only be used with a fixed number of ticks, i.e. after set_ticks() or
using a FixedLocator.
  axes[row, col_index].set_xticklabels(labels, rotation=45,
ha='right')
C:\Users\local_gud2i5y\AppData\Local\Temp\
ipykernel_24976\3045389681.py:13: UserWarning: set_ticklabels() should
only be used with a fixed number of ticks, i.e. after set_ticks() or
using a FixedLocator.
  axes[row, col_index].set_xticklabels(labels, rotation=45,
ha='right')
```

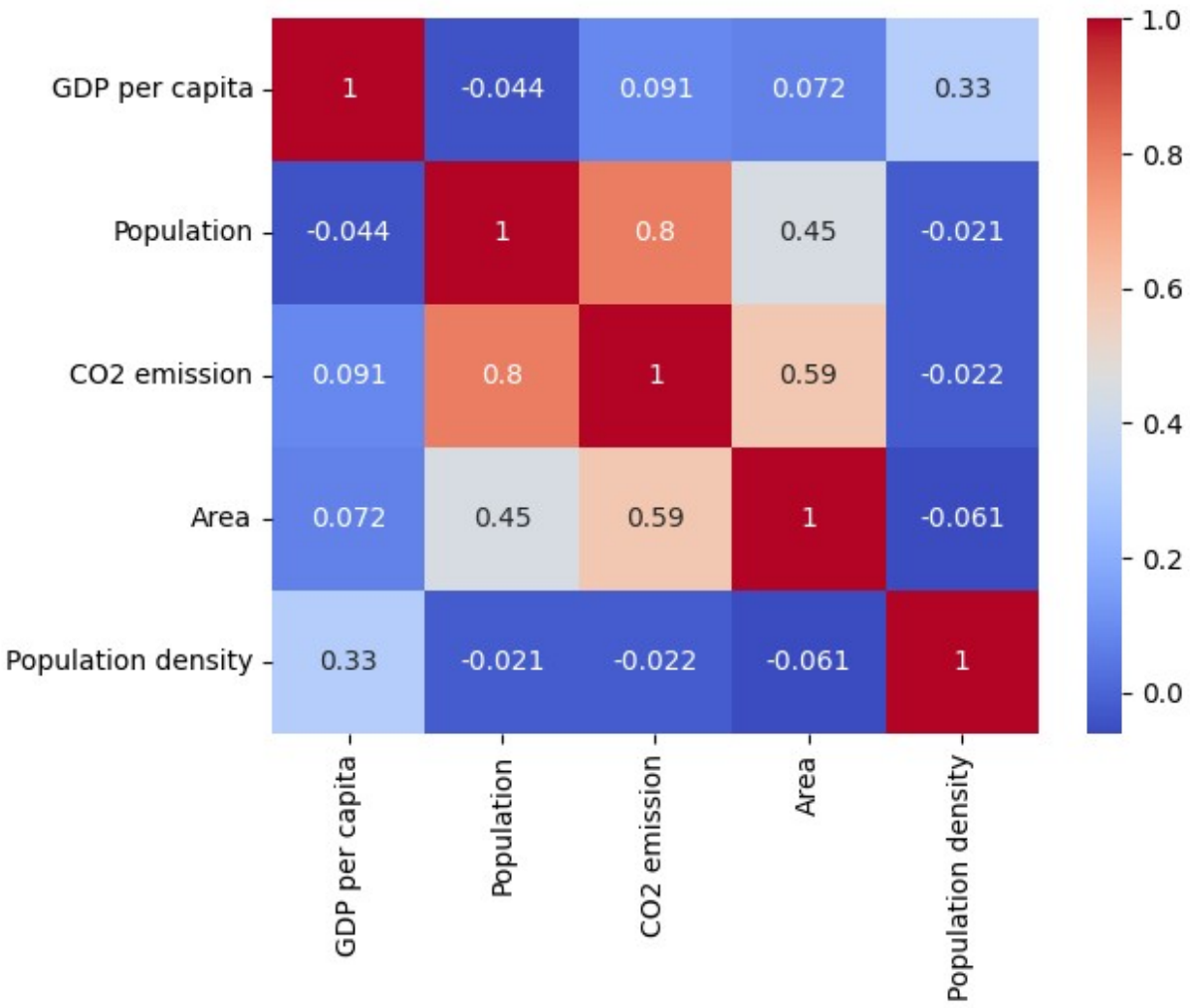


Завдання #2:

Рахую кореляцію між всіма кількісними ознаками

```
df_numeric = df.select_dtypes(include=['float64'])

correlation_matrix = df_numeric.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
```

```
<Axes: >
```

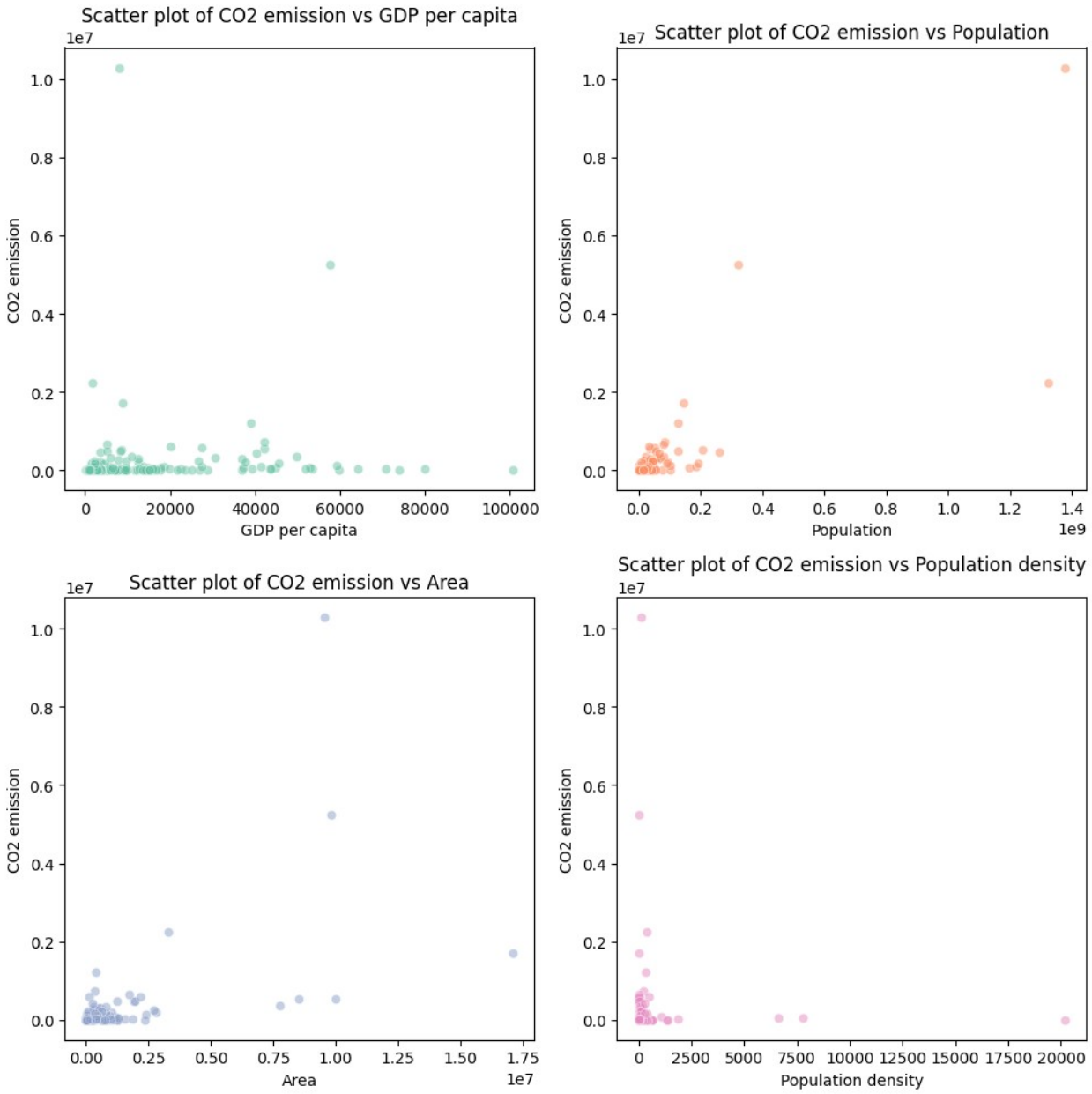


Завдання #3:

Будую діаграму розсіювання для кількісних ознак та `CO2 emission`

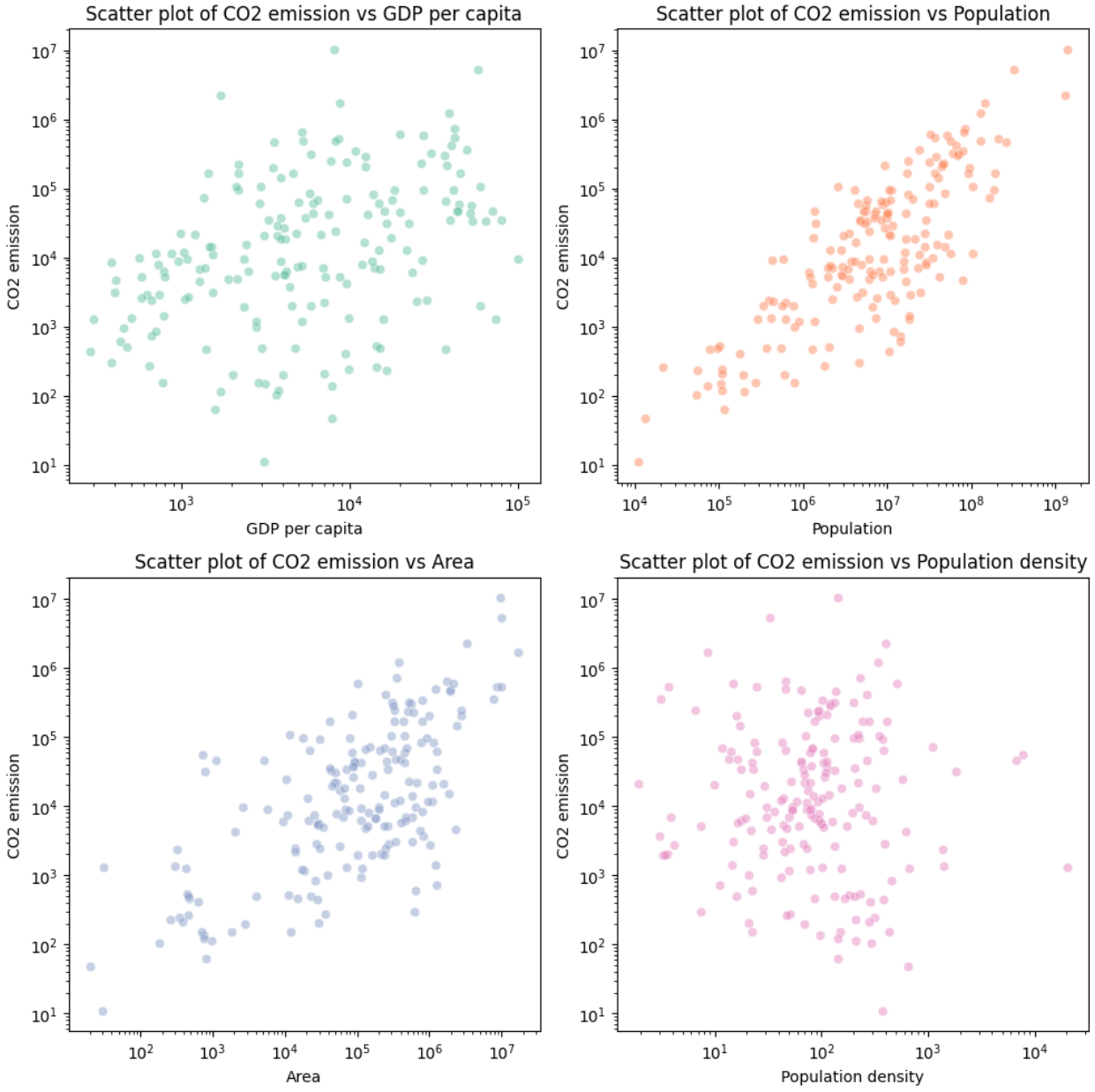
```
quantitative_columns = ['GDP per capita', 'Population', 'Area',
'Population density']

fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10, 10))

for i, col in enumerate(quantitative_columns):
    row = i // 2
    col_index = i % 2
    sns.scatterplot(data=df, x=col, y='CO2 emission', color=clrs[i],
alpha=0.5, ax=axes[row, col_index])
    axes[row, col_index].set_title(f'Scatter plot of CO2 emission vs
{col}')
```

```
plt.tight_layout()
```




```
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10, 10))

for i, col in enumerate(quantitative_columns):
    row = i // 2
    col_index = i % 2
    sns.scatterplot(data=df, x=col, y='CO2 emission', color=clrs[i],
alpha=0.5, ax=axes[row, col_index])
    axes[row, col_index].set(xscale="log", yscale="log")
    axes[row, col_index].set_title(f'Scatter plot of CO2 emission vs
```

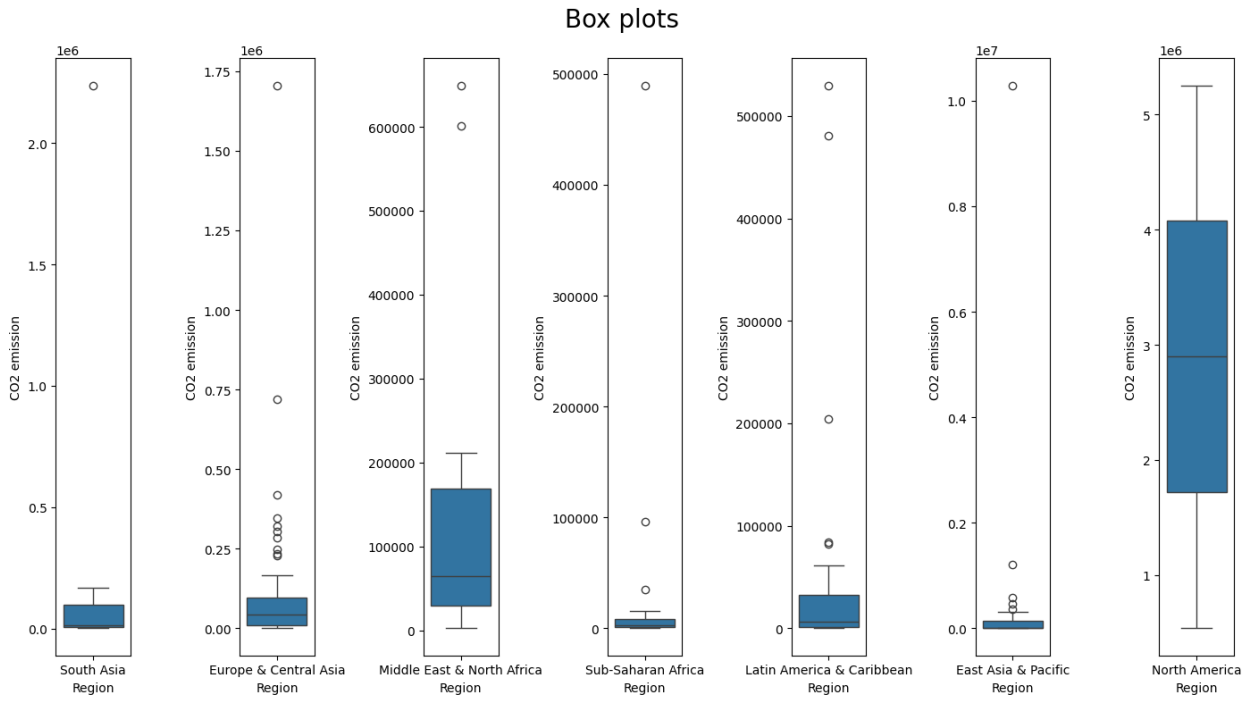
```
{col}')

plt.tight_layout()
```



Будую діаграму розмаху для `CO2 emission` по регіонам

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

fig, axes = plt.subplots(1, ncols=len(regions),
figsize=(2*len(regions), 8))
fig.suptitle('Box plots', fontsize=20)
```

```
for i, region in enumerate(regions):
    region_data = df[df['Region'] == region]
    sns.boxplot(ax=axes[i], data=region_data, x='Region', y='CO2
emission')

plt.tight_layout()
```



Завдання #4:

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

```
import scipy.stats
from termcolor import colored

quantitative_columns = ['GDP per capita', 'Population', 'Area',
'Population density']

for col in quantitative_columns:
    correlation_coefficient, p_value = scipy.stats.pearsonr(df[col],
df['CO2 emission'])

    # Green for significant, yellow for non-significant
    p_value_color = 'green' if p_value < 0.05 else 'yellow'

    print(f"For {colored(col, 'blue')} and CO2 emission:")
```

```
    print(f"\tThe Pearson correlation coefficient is
{correlation_coefficient}")
    print(f"\tThe P-value is {colored(p_value, p_value_color)}")
    print()
```

```
For GDP per capita and CO2 emission:
     The Pearson correlation coefficient is 0.0913637432897542
     The P-value is 0.21867791078062213

For Population and CO2 emission:
     The Pearson correlation coefficient is 0.8030020569760477
     The P-value is 1.523799314848454e-42

For Area and CO2 emission:
     The Pearson correlation coefficient is 0.5870512066109349
     The P-value is 2.4632499671008433e-18

For Population density and CO2 emission:
     The Pearson correlation coefficient is -0.022018057694006544
     The P-value is 0.7673448766598259
```

Завдання #5:

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

```
region_co2_mean = df.groupby('Region')['CO2 emission'].mean()
region_co2_mean
```

```
Region
East Asia & Pacific           4.502644e+05
Europe & Central Asia         1.272658e+05
Latin America & Caribbean     5.363957e+04
Middle East & North Africa    1.471921e+05
North America                 2.895736e+06
South Asia                    3.145543e+05
Sub-Saharan Africa            1.804637e+04
Name: CO2 emission, dtype: float64
```

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

```
regions = df['Region'].unique()
data = [df[df['Region'] == region]['CO2 emission'] for region in
regions]

f_value, p_value = scipy.stats.f_oneway(*data)
p_value_color = 'green' if p_value < 0.05 else 'yellow'
```

```
print(f"\tThe F-test score: {f_value}")
print(f"\tThe P-value is {colored(p_value, p_value_color)}")
```

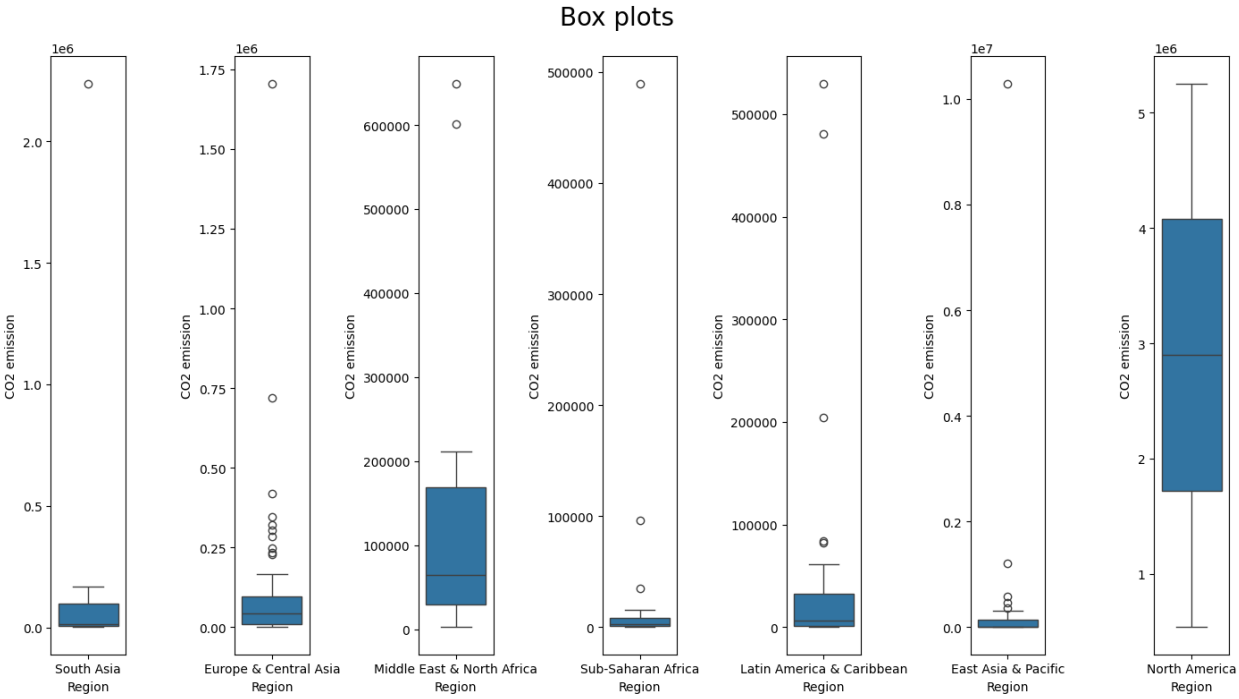
```
	The F-test score: 4.559032591227754
	The P-value is 0.0002520786181145583
```

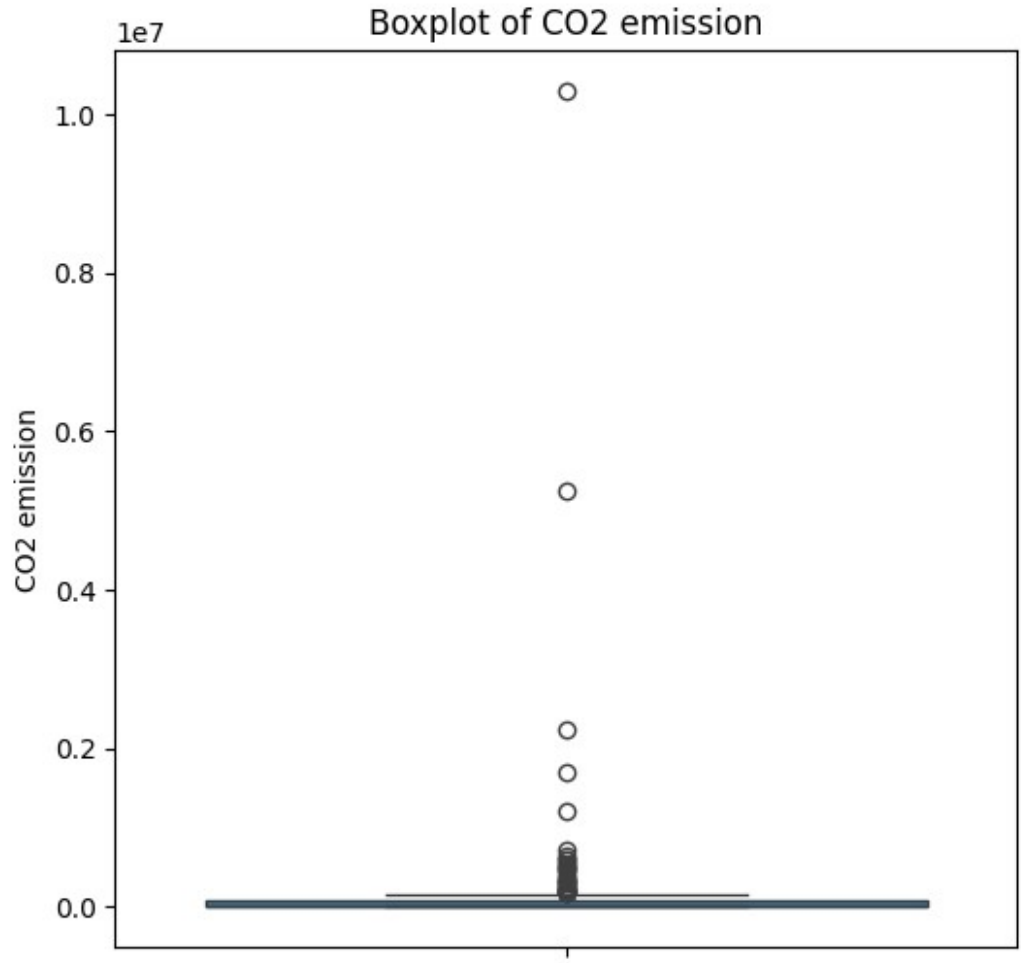
Результат із **4.559032591227754** показником тесту, який показує високу варіацію між групами у порівнянні з варіацією всередині групи, і P-value **0.0002520786181145583** , що вказує на дуже низьку ймовірність отримати такі або ще більш екстримальні значення. Але чи означає це, що досліджувані групи статистично значущо корелюють між собою?

Розглянемо їх окремо.

```
import matplotlib.pyplot as plt
import seaborn as sns

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

fig, axes = plt.subplots(1, ncols=len(regions),
figsize=(2*len(regions), 8))
fig.suptitle('Box plots', fontsize=20)

for i, region in enumerate(regions):
    region_data = df[df['Region'] == region]
    sns.boxplot(ax=axes[i], data=region_data, x='Region', y='CO2
emission')

plt.tight_layout()

plt.figure(figsize=(6, 6))
sns.boxplot(y='CO2 emission', data=df)
plt.title('Boxplot of CO2 emission')
```

```
Text(0.5, 1.0, 'Boxplot of CO2 emission')
```


Box plots
1e6
CO2 emission
South Asia
Region
1e6
CO2 emission
Europe & Central Asia
Region
CO2 emission
Middle East & North Africa
Region
CO2 emission
Sub-Saharan Africa
Region
CO2 emission
Latin America & Caribbean
Region
1e7
CO2 emission
East Asia & Pacific
Region
1e6
CO2 emission
North America
Region


Boxplot of CO2 emission
1e7
CO2 emission

Отже, на основі цих результатів, ми можемо зробити висновок, що `CO2 emission` статистично значущо корелює з `Region`.

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. Регіони з найбільшою кількістю викидів CO2 та найменшою.

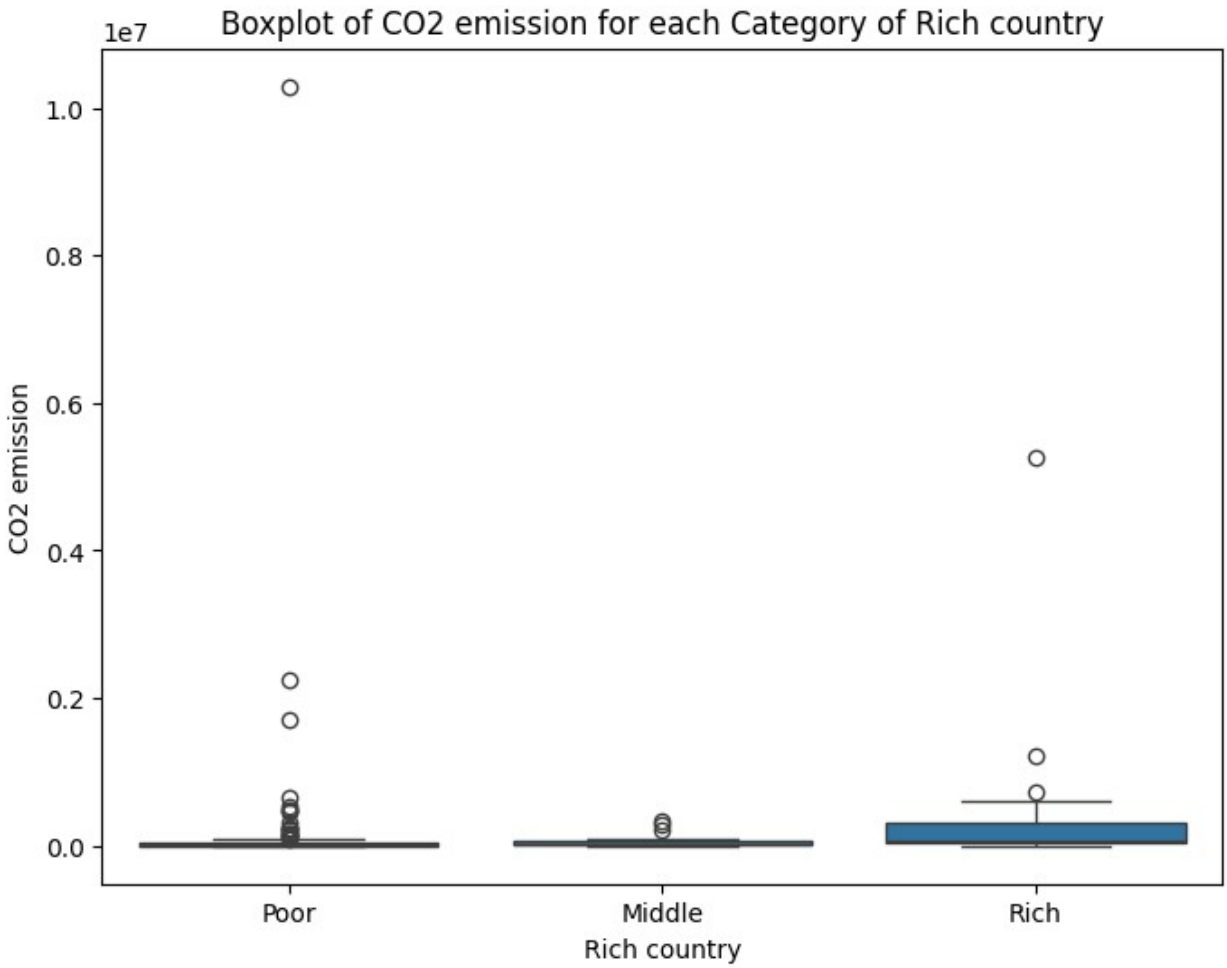
```
print(f"The most different regions: {region_co2_mean.idxmax()} та
{ region_co2_mean.idxmin()}")
```

```
The most different regions: North America та Sub-Saharan Africa
```

1. Створення якісної ознаки `Rich country`. Створення діаграми розмаху для `CO2 emission` за цією категорією.

```
import numpy as np

df['Rich country'] = pd.cut(df['GDP per capita'], bins=[0, 10000,
20000, np.inf], labels=['Poor', 'Middle', 'Rich'])

plt.figure(figsize=(8, 6))
sns.boxplot(x='Rich country', y='CO2 emission', data=df)
plt.title('Boxplot of CO2 emission for each Category of Rich country')
```

```
Text(0.5, 1.0, 'Boxplot of CO2 emission for each Category of Rich
country')
```



1. Дисперсійний аналіз для `CO2 emission`, згрупованих по категоріям `Rich country`.

```
categories = df['Rich country'].unique()
data = [df[df['Rich country'] == category]['CO2 emission'].dropna()
for category in categories]

f_value, p_value = scipy.stats.f_oneway(*data)
p_value_color = 'green' if p_value < 0.05 else 'yellow'

print(f"\tThe F-test score: {f_value}")
print(f"\tThe P-value is {colored(p_value, p_value_color)}")
```

```
    The F-test score: 0.6882720195506884
    The P-value is 0.5037608679303966
```