МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Практикум №3

з курсу «Аналіз даних в інформаційнних системах» на тему: «Описова статистика»

Викладач:

Олійник Ю.О.

Виконав:

студент 2 курсу групи ІП-15 ФІОТ  
Мєшков Андрій Ігорович

Київ-2023

**Практикум №3**

**Описова статистика**

**Мета роботи**: ознайомитись з методикою первинної обробки статистичних даних; проаналізувати вплив способу представлення даних на їх інформативність.

**Завдання:**

Скачати потрібні дані.

**Основне завдання**

Скачати дані із файлу Data2.csv

1. Записати дані у data frame
2. Дослідити структуру даних
3. Виправити помилки в даних
4. Побудувати діаграми розмаху та гістограми
5. Додати стовпчик із щільністю населення

**Додаткове завдання**

Відповісти на питання (файл Data2.csv):

1. Чи є пропущені значення? Якщо є, замінити середніми
2. Яка країна має найбільший ВВП на людину (GDP per capita)? Яка має найменшу площу?
3. В якому регіоні середня площа країни найбільша?
4. Знайдіть країну з найбільшою щільністю населення у світі? У Європі та центральній Азії?
5. Чи співпадає в якомусь регіоні середнє та медіана ВВП?
6. Вивести топ 5 країн та 5 останніх країн по ВВП та кількості СО2 на душу населення.

**Хід роботи:**

**Основне завдання:**

Інсталюємо pandas:

pip3 install pandas

Імпортуємо бібліотеку

import pandas as pd

1. Записати дані у data frame

data\_path = 'Data2.csv'

df = read\_dataset(data\_path)

def read\_dataset(path):

df = pd.read\_csv(path, sep=';', encoding='cp1252')

*return* df

1. Дослідити структуру даних

def print\_exploring(df):

print('Data frame info:')

df.info()

print('\nFirst 5 rows:')

print(df.head())

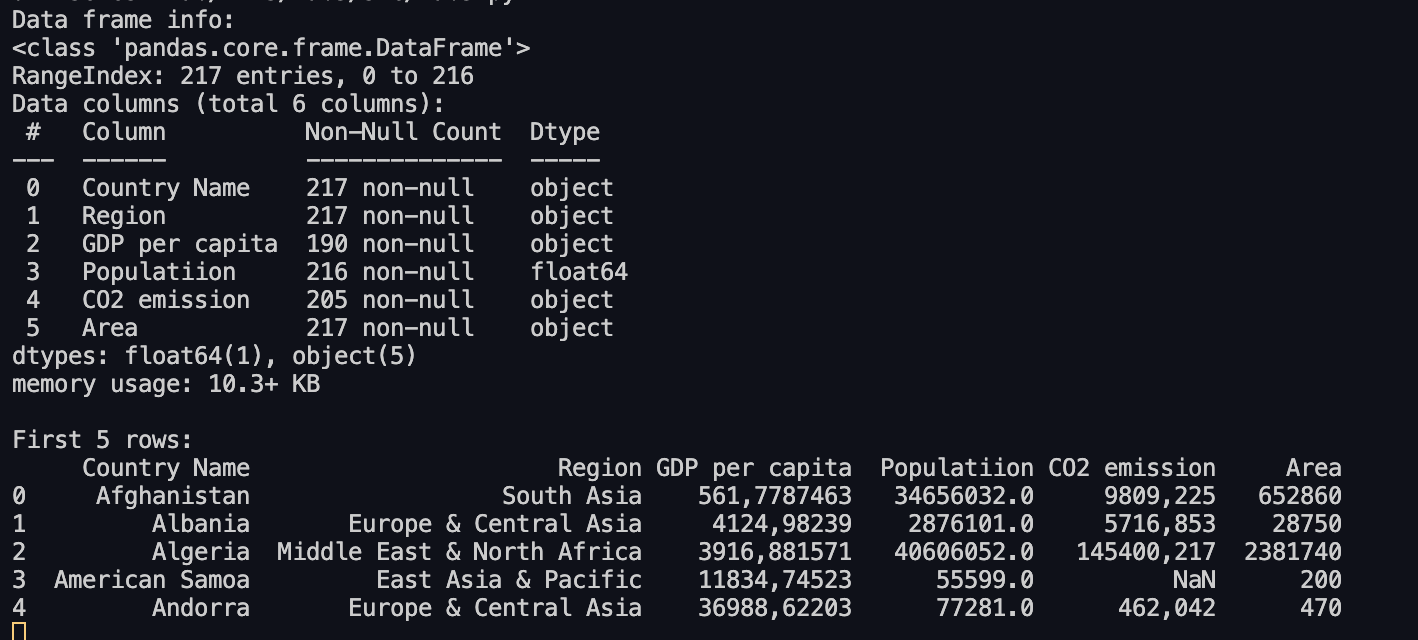


Рисунок 1. Дослідження структури

1. Виправити помилки в даних

Виправляємо помилки

def remove\_typo(df):

df.rename(columns={"Populatiion": "Population"}, inplace=True)

*return* df

Форматування типу даних

def clean\_up(df):

df['Area'] = df['Area'].str.replace(',', '.').astype(float)

df["GDP per capita"] = df["GDP per capita"].str.replace(',', '.').astype(float)

df["CO2 emission"] = df["CO2 emission"].str.replace(',', '.').astype(float)

*return* df

Виправити негативні дані

def fix\_negative(df):

fix\_gdp = df[df['GDP per capita'] < 0]

area\_gdp = df[df['Area'] < 0]

fix\_gdp['GDP per capita'] \*= -1

area\_gdp['Area'] \*= -1

df[df['GDP per capita'] < 0] = fix\_gdp

df[df['Area'] < 0] = area\_gdp

*return* df

Виправити пропущені дані середніми

def fix\_NaN(df):

df = df.fillna(df.mean())

*return* df

1. Побудувати діаграми розмаху та гістограми

Інсталюємо бібліотеку Matplotlib

pip3 install matplotlib

Імпортуємо бібліотеку

import matplotlib.pyplot as plt

Створюємо діаграми розмаху

def create\_boxplot(df):

fig, axs = plt.subplots(1, 4, figsize=(16, 4))

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'])

Створюємо гістограми

def create\_hist(df):

fig, axs = plt.subplots(1, 4, figsize=(16, 4))

fig.suptitle('Гістограми', fontsize=16)

axs[0].set\_title('GDP per capita')

axs[0].hist(df['GDP per capita'])

axs[1].set\_title('Population')

axs[1].hist(df['Population'])

axs[2].set\_title('CO2 emission')

axs[2].hist(df['CO2 emission'])

axs[3].set\_title('Area')

axs[3].hist(df['Area'])

Показуємо діаграми

plt.show()

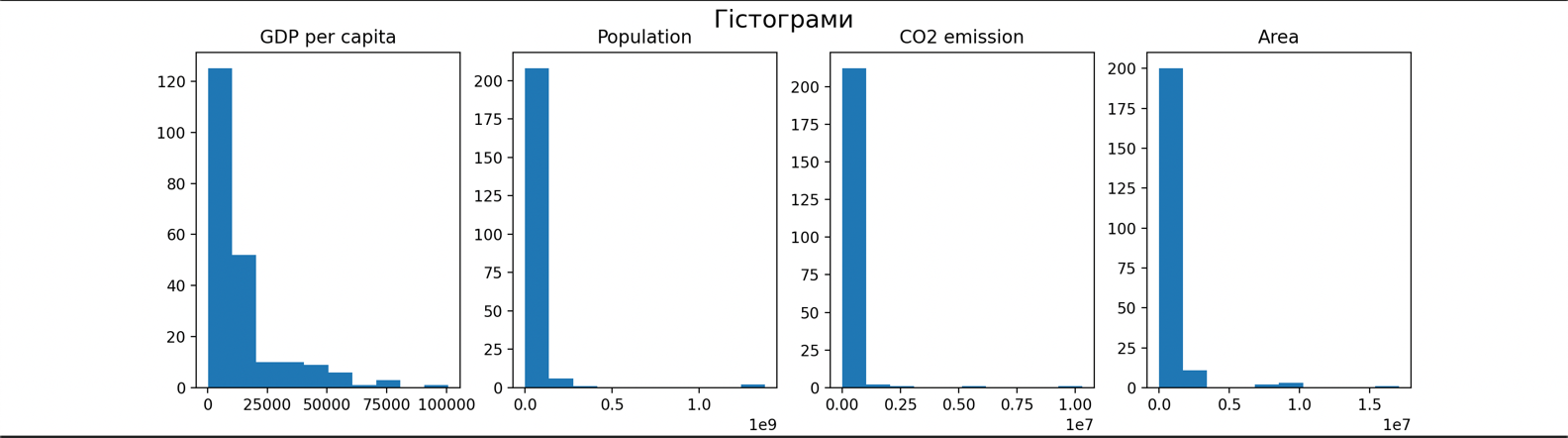


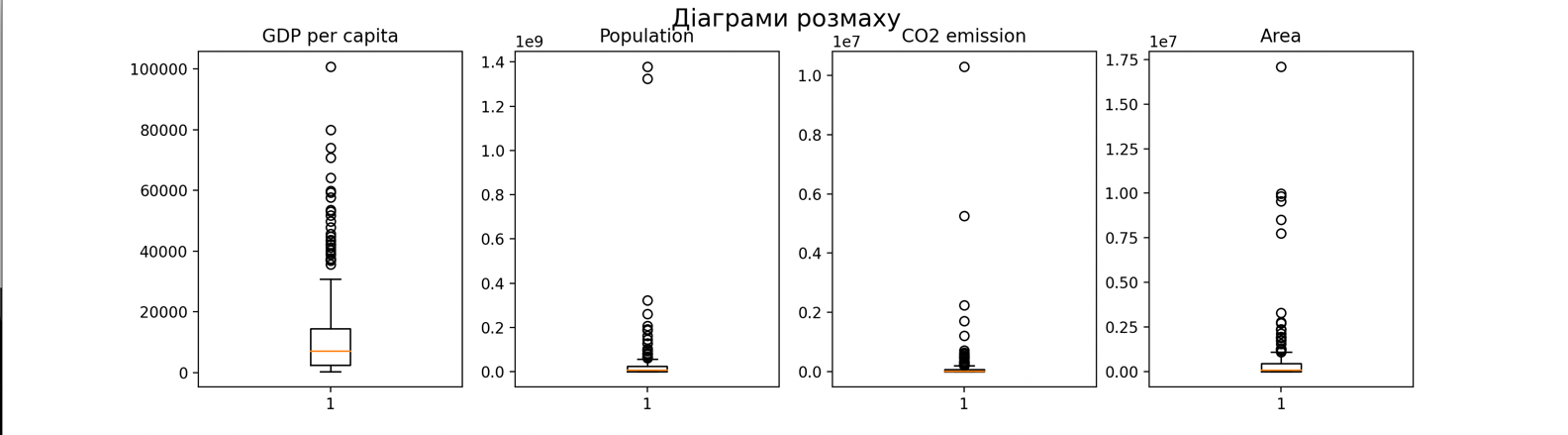
Рисунок 2. Гістограми

Рисунок 3. Діаграми розмаху

1. Додати стовпчик із щільністю населення

def add\_population\_density(df):

df["Population\_density"] = df["Population"] / df["Area"]

print(df.head())

*return* df

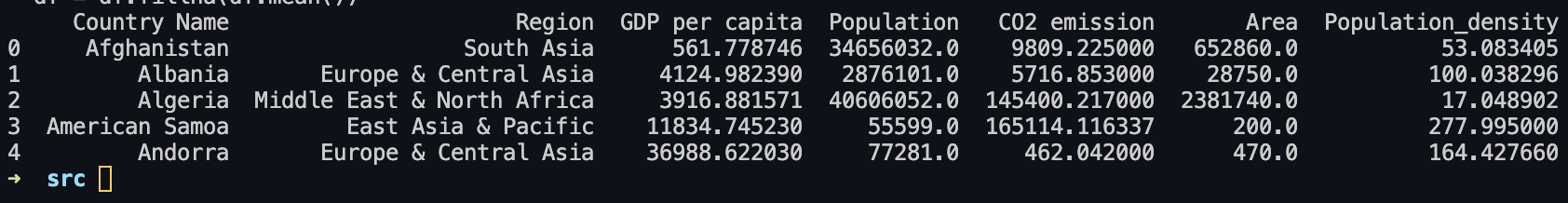


Рисунок 4. Щільність населення

**Додаткове завдання:**

1. Чи є пропущені значення? Якщо є, замінити середніми

Пропущені значення є, вони були замінені раніше за допомогою функції fix\_NaN(df).

1. Яка країна має найбільший ВВП на людину (GDP per capita)? Яка має найменшу площу?

highest\_gdp\_per\_capita = df.loc[df['GDP per capita'].idxmax()]

print("Country with the highest GDP per capita:", highest\_gdp\_per\_capita['Country Name'])

smallest\_area = df.loc[df['Area'].idxmin()]

print("Country with the smallest area:", smallest\_area['Country Name'])

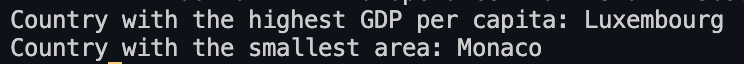


Рисунок 5. Відповідь на питання№2

1. В якому регіоні середня площа країни найбільша?

mean\_area\_by\_region = df.groupby('Region')['Area'].mean()

region\_with\_highest\_mean\_area = mean\_area\_by\_region.idxmax()

print("Region with the highest average area per country:", region\_with\_highest\_mean\_area)



Рисунок 6. Відповідь на питання№3

1. Знайдіть країну з найбільшою щільністю населення у світі? У Європі та центральній Азії?

highest\_pop\_density = df.loc[df['Population\_density'].idxmax()]

print("Country with the highest population density in the world:", highest\_pop\_density['Country Name'])

europe\_and\_central\_asia = df[df['Region'].isin(['Europe & Central Asia'])]

highest\_pop\_density\_in\_europe\_and\_central\_asia = europe\_and\_central\_asia.loc[europe\_and\_central\_asia['Population\_density'].idxmax()]

print("Country with the highest population density in Europe and Central Asia:", highest\_pop\_density\_in\_europe\_and\_central\_asia['Country Name'])

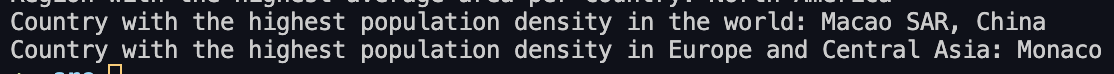


Рисунок 7. Відповідь на питання№4

1. Чи співпадає в якомусь регіоні середнє та медіана ВВП?

*for* region *in* df['Region'].unique():

region\_data = df[df['Region'] == region]

mean\_gdp = region\_data['GDP per capita'].mean()

median\_gdp = region\_data['GDP per capita'].median()

print('\nMean GDP per capita in', region, ' - ' ,mean\_gdp)

print('Median GDP per capita in', region, ' - ' , median\_gdp)

*if* mean\_gdp == median\_gdp:

print(f"In the {region} region, the mean and median GDP per capita are the same: {mean\_gdp}")

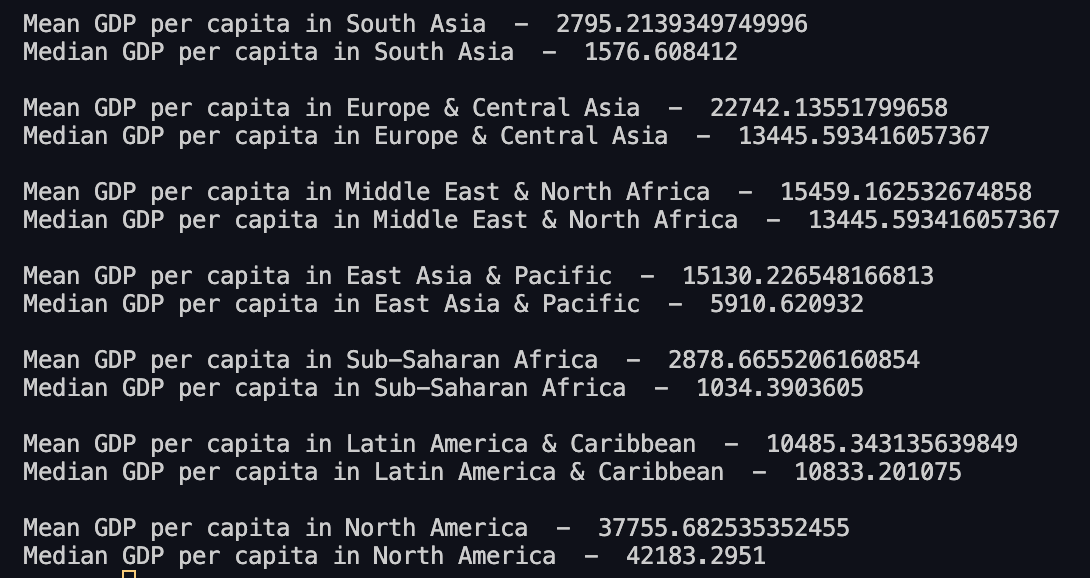


Рисунок 8. Відповідь на питання№5. Середнє і медіана ВВП ніде не співпадає.

1. Вивести топ 5 країн та 5 останніх країн по ВВП та кількості СО2 на душу населення.

df\_sorted\_gdp = df.sort\_values(by='GDP per capita', ascending=False)

top5\_gdp = df\_sorted\_gdp.head(5)

bottom5\_gdp = df\_sorted\_gdp.tail(5)

print("\nTop 5 countries by GDP per capita:")

print(top5\_gdp)

print("\nBottom 5 countries by GDP per capita:")

print(bottom5\_gdp)

pd.set\_option("display.max\_columns", None)

df['CO2 emission per citizen'] = df['CO2 emission'] / df['Population']

df\_sorted\_co2 = df.sort\_values(by='CO2 emission per citizen', ascending=False)

top5\_co2 = df\_sorted\_co2.head(5)

bottom5\_co2 = df\_sorted\_co2.tail(5)

print("\n\nTop 5 countries by CO2 emissions per citizen:")

print(top5\_co2)

print("\nBottom 5 countries by CO2 emissions per citizen:")

print(bottom5\_co2)

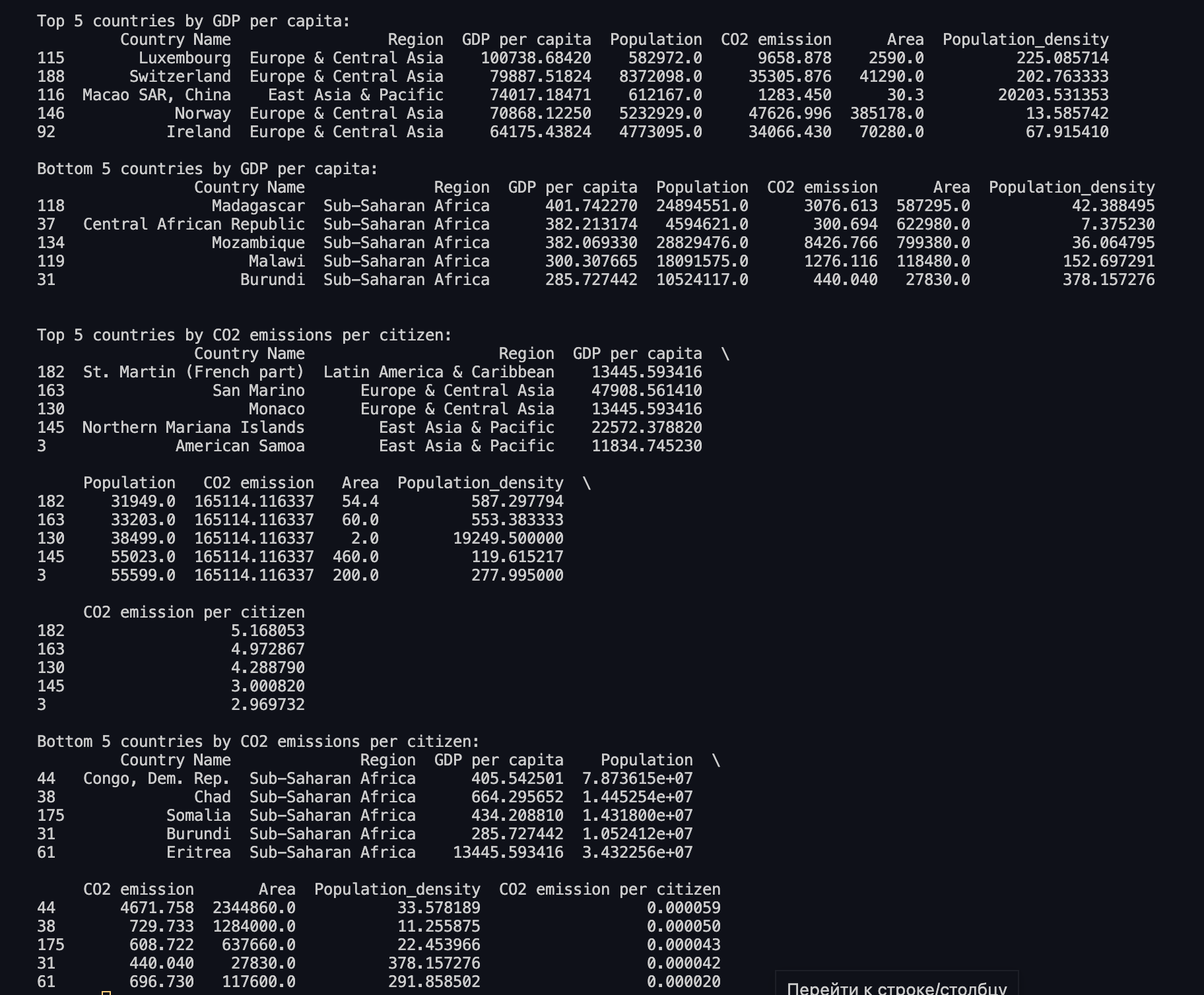
****

Рисунок 9. Відповідь на питання№6.

Висновок

За отриманими даними можна зробити висновок, що країни західної Європи мають високий ВВП, тоді як країни Африки мають найнижчі показники. Значення викидів вуглекислого газу на одну особу в країнах Африки є дуже малим. Проте, щодо країн з найбільшим значенням цього показника, не можна бути впевненим, оскільки серед топ-5 країн, які були отримані, всі мали пропущене значення в початковому датасеті. Це призвело до того, що їм було присвоєно середнє значення даного показника, яке може бути далеким від істинного. Крім того, всі ці топ-5 країн мають порівняно невелику кількість населення, що спричиняє великий показник викидів на одну особу. Тому необхідно здійснити додатковий збір даних для отримання достовірних результатів.