Додаток 1

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

Звіт

з комп'ютерного практикуму № 3 з дисципліни

«Аналіз даних в інформаційних системах»

на тему: «Описова статистика»

Виконав студент ІП-13, Шиманська Ганна Артурівна
(шифр, прізвище, ім'я, по батькові)

Перевірила Ліхоузова Тетяна Анатолівна
(прізвище, ім'я, по батькові)

Київ 2023

Комп’ютерний практикум 3

Тема – Описова статистика.

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

Завдання

Основне:

1. Скачати дані із файлу Data2.csv
2. Записати дані у data frame
3. Дослідити структуру даних
4. Виправити помилки в даних
5. Побудувати діаграми розмаху та гістограми
6. Додати стовпчик із щільністю населення

Додаткове:

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 CO2 на душу населення.

Імпортуємо усі необхідні пакети.

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

Зчитуємо датасет, правильно вказавши його кодування.

```
def read_dataset(path: str):
    data = pd.read_csv(path, sep=";", encoding='cp1252')
    return data
df = read_dataset("Data2.csv")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 217 entries, 0 to 216
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Country Name    217 non-null    object
 1   Region          217 non-null    object
 2   GDP per capita  190 non-null    object
 3   Populatiion     216 non-null    float64
 4   CO2 emission    205 non-null    object
 5   Area            217 non-null    object
dtypes: float64(1), object(5)
memory usage: 10.3+ KB
```

При дослідженні даних датасету можна помітити, що певна колонка містять помилку в назві. Також, що деякі колонки з числовими даними містять тип даних object. Щоб коректно привести їх у тип float, необхідно усюди замінити коми на крапки. Також є пропущені дані. Ще є дані, значення яких менше нуля.

```
df = df.rename(columns={'Populatiion': 'Population'})
print(df.columns)
```

```
Index(['Country Name', 'Region', 'GDP per capita', 'Population',
       'CO2 emission', 'Area'],
      dtype='object')
```

Одразу приведемо дані до типу float задля подальших операцій.

```
def correct_floats_in_columns(df: DataFrame, list_of_columns: list[str]):
    for column in list_of_columns:
        df[column] = df[column].astype(str).str.replace(',', '.').astype(float)
    return df
correct_floats_in_columns(df, ['GDP per capita', 'CO2 emission', 'Area'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 217 entries, 0 to 216
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Country Name    217 non-null    object
 1   Region          217 non-null    object
 2   GDP per capita  190 non-null    float64
 3   Population      216 non-null    float64
```

```
 4   CO2 emission    205 non-null    float64
 5   Area            217 non-null    float64
dtypes: float64(4), object(2)
memory usage: 10.3+ KB
```

Тепер знайдемо рядки з пропущеною інформацією.

```
def get_missing_values(df:DataFrame):
    return df[df.isna().any(axis = 1)]
print(get_missing_values(df))
```

```
                 Country Name                     Region  GDP per capita  \
3              American Samoa       East Asia & Pacific    11834.745230
9                       Aruba  Latin America & Caribbean             NaN
21                    Bermuda             North America             NaN
27      British Virgin Islands  Latin America & Caribbean             NaN
36              Cayman Islands  Latin America & Caribbean             NaN
39             Channel Islands      Europe & Central Asia             NaN
49                        Cuba  Latin America & Caribbean             NaN
50                    Curacao  Latin America & Caribbean             NaN
54                   Djibouti  Middle East & North Africa             NaN
61                    Eritrea         Sub-Saharan Africa             NaN
64               Faroe Islands      Europe & Central Asia             NaN
68           French Polynesia       East Asia & Pacific             NaN
74                 Gibraltar      Europe & Central Asia             NaN
76                 Greenland      Europe & Central Asia             NaN
78                      Guam       East Asia & Pacific    35562.567530
93                 Isle of Man      Europe & Central Asia             NaN
102  Korea, Dem. People’s Rep.       East Asia & Pacific             NaN
104                    Kosovo      Europe & Central Asia     3661.429847
112                    Libya  Middle East & North Africa             NaN
113              Liechtenstein      Europe & Central Asia             NaN
130                   Monaco      Europe & Central Asia             NaN
140             New Caledonia       East Asia & Pacific             NaN
143                     Niger         Sub-Saharan Africa             NaN
145   Northern Mariana Islands       East Asia & Pacific    22572.378820
157               Puerto Rico  Latin America & Caribbean    30790.104790
163                San Marino      Europe & Central Asia    47908.561410
171  Sint Maarten (Dutch part)  Latin America & Caribbean             NaN
177               South Sudan         Sub-Saharan Africa             NaN
182   St. Martin (French part)  Latin America & Caribbean             NaN
189       Syrian Arab Republic  Middle East & North Africa             NaN
200   Turks and Caicos Islands  Latin America & Caribbean             NaN
210             Venezuela, RB  Latin America & Caribbean             NaN
212      Virgin Islands (U.S.)  Latin America & Caribbean             NaN
213         West Bank and Gaza  Middle East & North Africa     2943.404534

     Population  CO2 emission       Area
3       55599.0           NaN      200.0
9      104822.0       872.746      180.0
21      65331.0       575.719       50.0
27      30661.0       179.683      150.0
36      60765.0       542.716      264.0
39     164541.0           NaN      190.0
49   11475982.0     34836.500   109880.0
50     159999.0      5881.868      444.0
54     942333.0       722.399    23200.0
61          NaN       696.730   117600.0
64      49117.0       597.721     1396.0
```

```
68     280208.0       803.073     4000.0
74      34408.0       528.048       10.0
76      56186.0       506.046   410450.0
78     162896.0           NaN      540.0
93      83737.0           NaN      570.0
102  25368620.0     40527.684   120540.0
104   1816200.0           NaN    10887.0
112   6293253.0     56996.181  1759540.0
113     37666.0        44.004      160.0
130     38499.0           NaN        2.0
140    278000.0      4290.390    18580.0
143  20672987.0      2126.860  1267000.0
145     55023.0           NaN      460.0
157   3411307.0           NaN     8870.0
163     33203.0           NaN       60.0
171     40005.0       733.400       34.0
177  12230730.0      1496.136   644330.0
182     31949.0           NaN       54.4
189  18430453.0     30703.791   185180.0
200     34900.0       205.352      950.0
210  31568179.0    185220.170   912050.0
212    102951.0           NaN      350.0
213   4551566.0           NaN     6020.0
```

Знайдемо рядки з від'ємними значеннями.

```
def get_negative_rows(df: DataFrame):
    return df.loc[(df['GDP per capita'] < 0) | (df['CO2 emission'] < 0) | (df['Area']
< 0)]
print(get_negative_rows(df))
```

```
          Country Name                     Region  GDP per capita  \
56   Dominican Republic  Latin America & Caribbean    -6722.223536
135             Myanmar        East Asia & Pacific     1195.515372

     Population  CO2 emission      Area
56   10648791.0     21539.958   48670.0
135  52885223.0     21631.633 -676590.0
```

Виправляємо усі некоректні дані. Для цього:

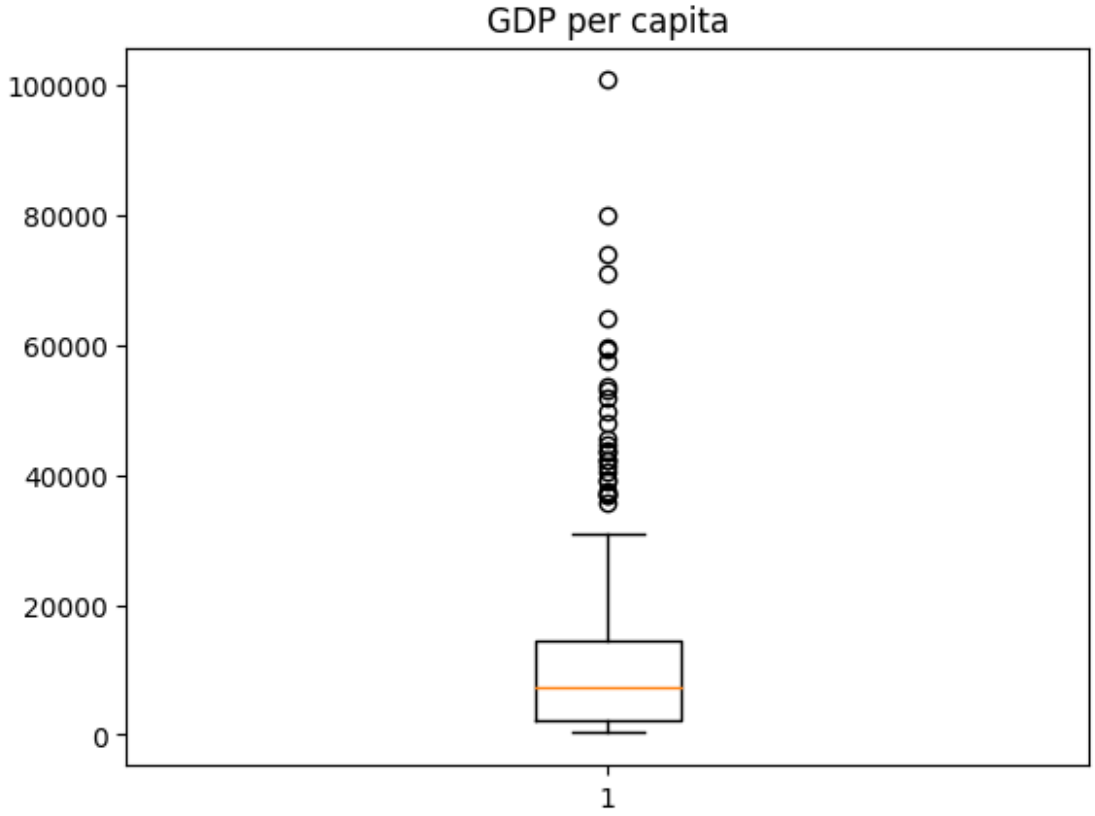
- Ми уже привели дані до типу float.
- Для кожної колонки беремо модуль усіх чисел.
- Заповнюємо пропущені дані середніми значеннями.

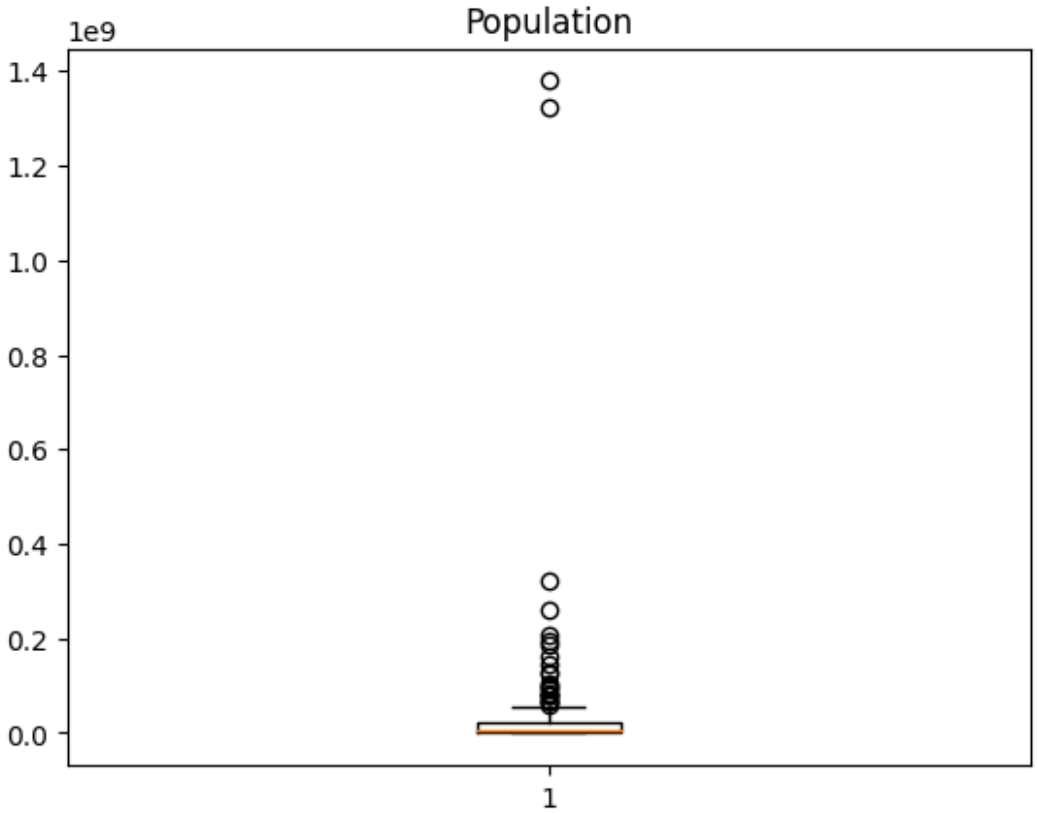
```
def correct_data(df: DataFrame, list_of_columns: list[str]):
    for column in list_of_columns:
        df[column] = df[column].abs()
    df = df.fillna(df.mean(numeric_only=True))
    return df
df = correct_data(df, ['GDP per capita', 'CO2 emission', 'Area'])
print(get_negative_rows(df))
print(get_missing_values(df))
```

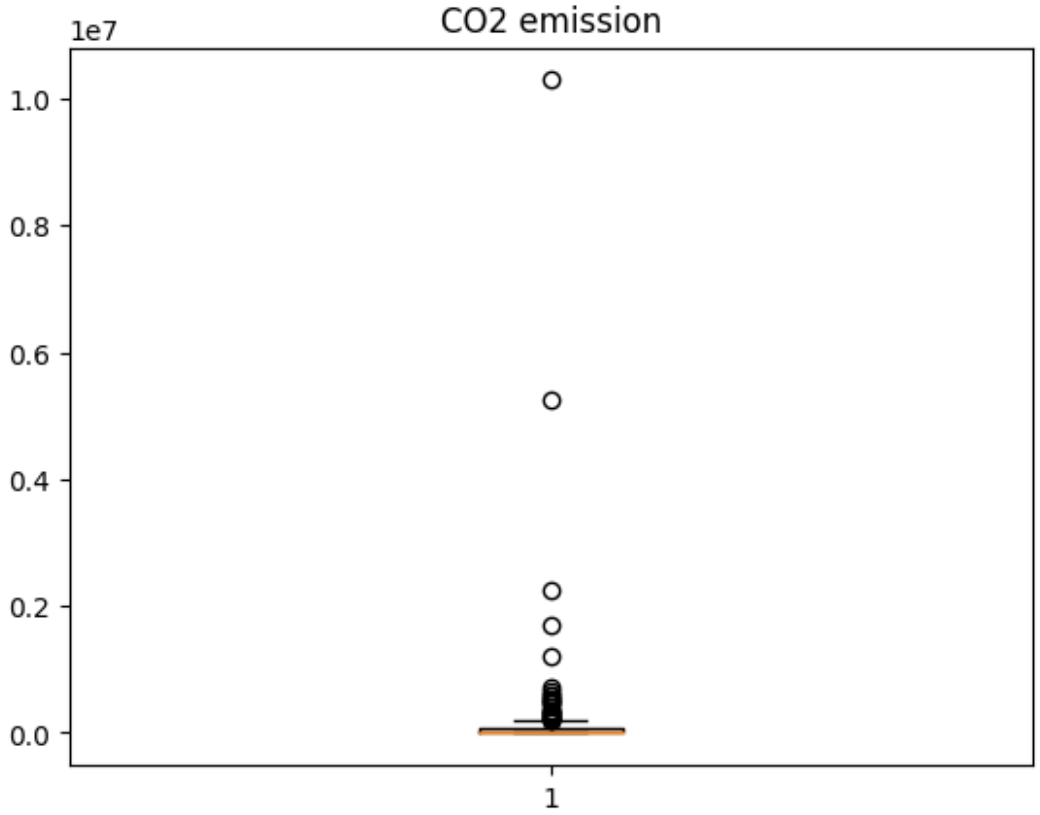
```
Empty DataFrame
Columns: [Country Name, Region, GDP per capita, Population, CO2 emission, Area]
Index: []
Empty DataFrame
Columns: [Country Name, Region, GDP per capita, Population, CO2 emission, Area]
Index: []
```

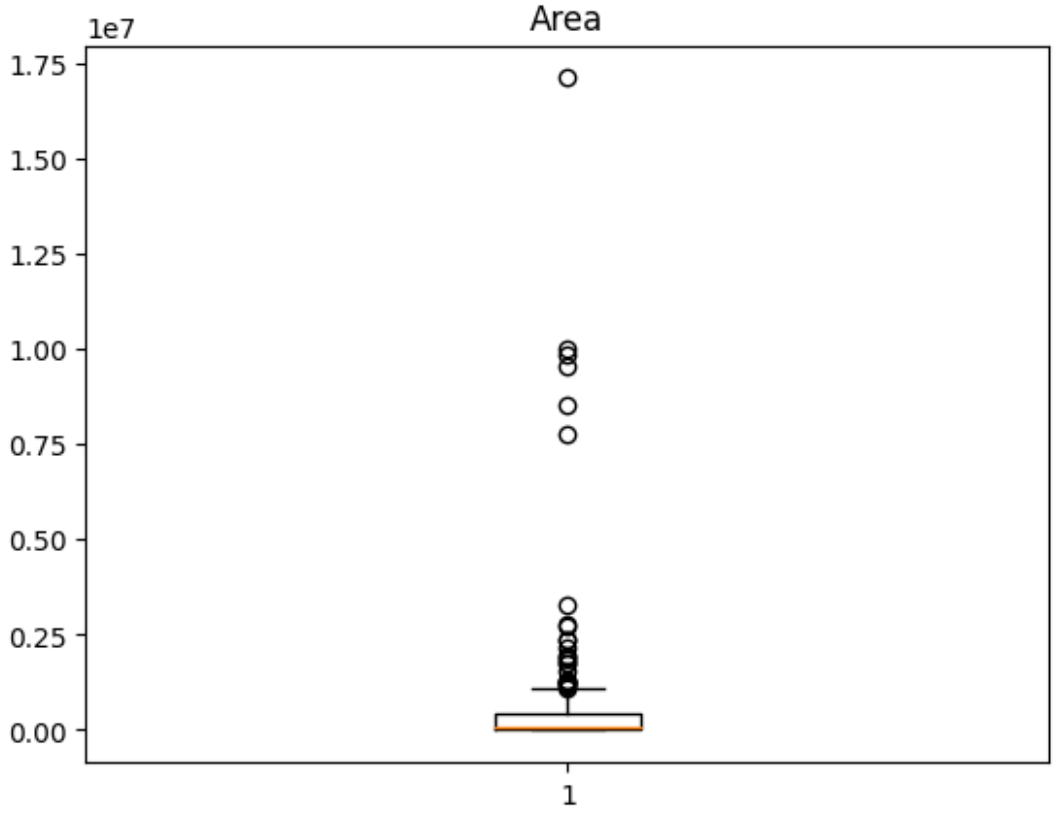
Побудуємо діаграми розмаху для кожного стовпця з числовими значеннями.

```
for column in df.columns:
    if df[column].dtype == float:
        plt.figure()
        plt.boxplot(df[column])
        plt.title(column)
        plt.show()
```



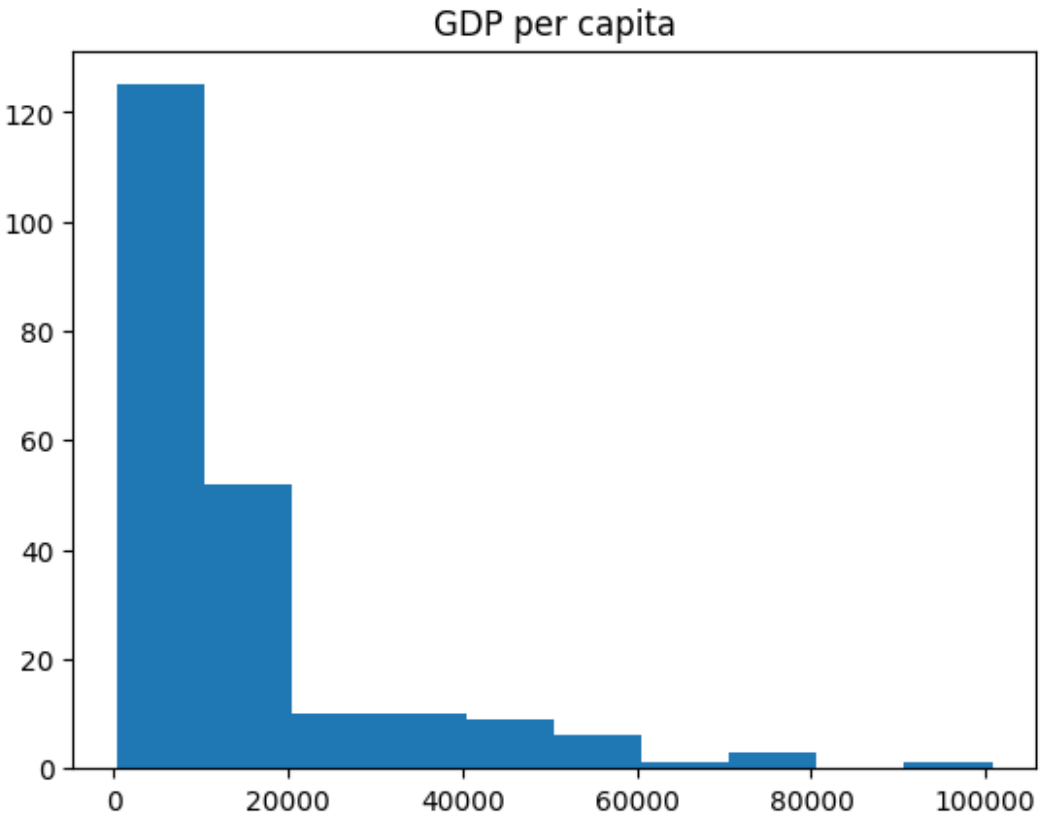


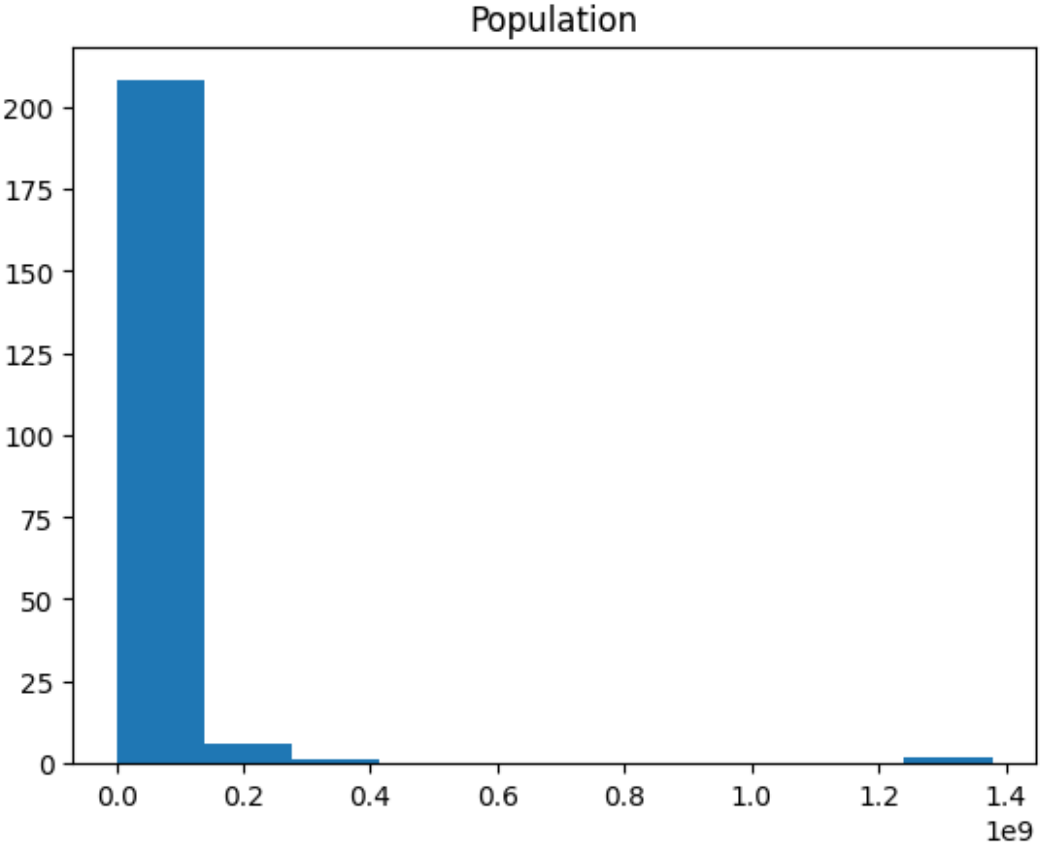


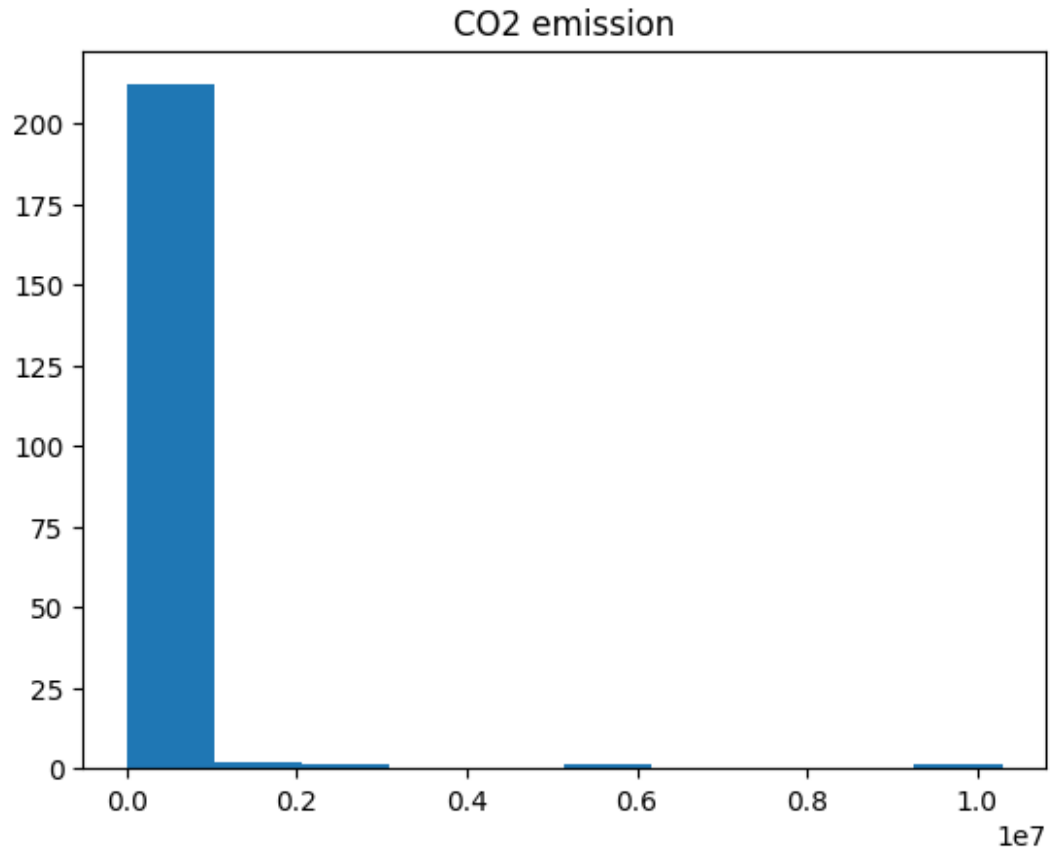


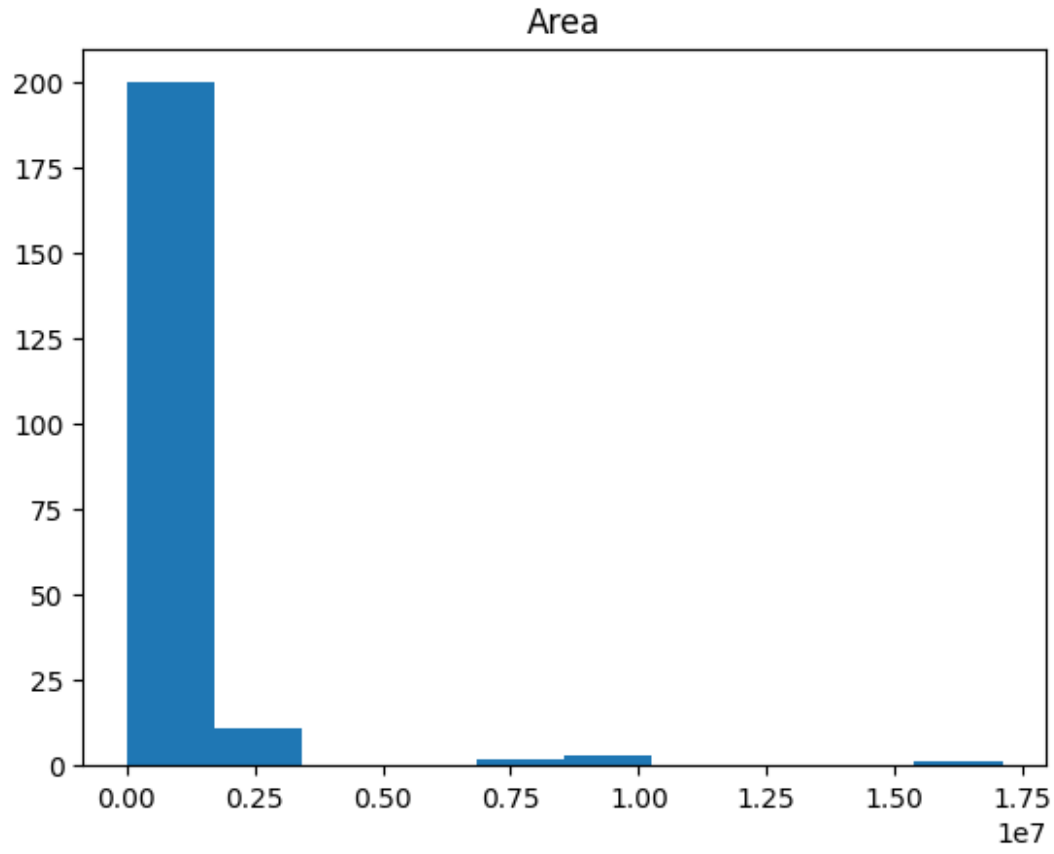
Побудуємо гістограми для кожної числової колонки

```
for column in df.columns:
    if df[column].dtype == float:
        plt.hist(df[column])
        plt.title(column)
        plt.show()
```


GDP per capita
120
100
80
60
40
20
0
0
20000
40000
60000
80000
100000


Population
200
175
150
125
100
75
50
25
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e9





Створимо стовпчик із щільністю населення.

```
df["Population Density"] = df["Population"] / df["Area"]
df.head()
```

```
     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  American Samoa         East Asia & Pacific    11834.745230     55599.0
4         Andorra       Europe & Central Asia    36988.622030     77281.0

    CO2 emission       Area  Population Density
0    9809.225000   652860.0           53.083405
1    5716.853000    28750.0          100.038296
2  145400.217000  2381740.0           17.048902
3  165114.116337      200.0          277.995000
4     462.042000      470.0          164.427660
```

Пропущені значення уже були замінені на середні. Тепер знайдемо країну з найбільшим ВВП на людину та країну з найменшою площею.

```
row_with_max_gdp = df.loc[df['GDP per capita'].idxmax()]
max_gdp_country = row_with_max_gdp['Country Name']
print("Країна з найбільшим ВВП на людину:", max_gdp_country)

row_with_min_area = df.loc[df['Area'].idxmin()]
min_area_country_name = row_with_min_area['Country Name']
print("Країна з найменшою площею:", min_area_country_name)
```

```
Країна з найбільшим ВВП на людину: Luxembourg
Країна з найменшою площею: Monaco
```

Знайдемо регіон, в якому середня площа країни найбільша.

```
avg_areas_in_region = df.groupby(['Region']).mean(numeric_only=True)['Area']
region_with_max_avg_areas = avg_areas_in_region.idxmax()
print("Регіон, в якому середня площа країни найбільша:", region_with_max_avg_areas)
```

```
Регіон, в якому середня площа країни найбільша: North America
```

Знайдемо країну з найбільшою щільністю населення у світі, у Європі та центральній Азії.

```
max_population_density_index_world = df['Population Density'].idxmax()
max_population_density_country_world = df.loc[max_population_density_index_world,
'Country Name']
print("Країна з найбільшою щільністю населення у світі:",
max_population_density_country_world)

df_eu_and_ca_countries = df[df['Region'] == 'Europe & Central Asia']
max_population_density_eu_ca_index = df_eu_and_ca_countries['Population
Density'].idxmax()
max_population_density_eu_ca_country = df.loc[max_population_density_eu_ca_index,
'Country Name']
print("Країна з найбільшою щільністю населення у Європі та центральній Азії:",
max_population_density_eu_ca_country)
```

```
Країна з найбільшою щільністю населення у світі: Macao SAR, China
Країна з найбільшою щільністю населення у Європі та центральній Азії: Monaco
```

Перевіримо, чи співпадає в якомусь регіоні середнє та медіана ВВП.

```
mean_gdp_region = df.groupby(['Region']).mean(numeric_only=True)['GDP per capita']
median_gdp_region = df.groupby(['Region']).median(numeric_only=True)['GDP per
capita']

print('Середнє ВВП за регіоном:\n', mean_gdp_region)
print('Медіана за регіоном:\n', median_gdp_region)
```

```
Середнє ВВП за регіоном:
 Region
East Asia & Pacific           15130.226548
Europe & Central Asia         22742.135518
Latin America & Caribbean     10485.343136
Middle East & North Africa    15459.162533
North America                 37755.682535
South Asia                     2795.213935
Sub-Saharan Africa             2878.665521
Name: GDP per capita, dtype: float64
Медіана за регіоном:
 Region
East Asia & Pacific            5910.620932
Europe & Central Asia         13445.593416
Latin America & Caribbean     10833.201075
Middle East & North Africa    13445.593416
North America                 42183.295100
South Asia                     1576.608412
Sub-Saharan Africa             1034.390361
Name: GDP per capita, dtype: float64
```

```
compare_mean_median = mean_gdp_region.compare(median_gdp_region,
align_axis=1).rename(columns={'self': 'mean', 'other': 'median'}, level=-1)
compare_mean_median['diff'] = compare_mean_median['mean'] -
compare_mean_median['median']
print(compare_mean_median)
print("\nРядки, в яких різниця між середнім та медіаною дорівнює нулю:")
print(compare_mean_median.loc[(compare_mean_median['diff'] == 0)])
```

```
                                mean        median         diff
Region
East Asia & Pacific         15130.226548   5910.620932  9219.605616
Europe & Central Asia       22742.135518  13445.593416  9296.542102
Latin America & Caribbean   10485.343136  10833.201075  -347.857939
Middle East & North Africa  15459.162533  13445.593416  2013.569117
North America               37755.682535  42183.295100 -4427.612565
South Asia                   2795.213935   1576.608412  1218.605523
Sub-Saharan Africa           2878.665521   1034.390361  1844.275160

Рядки, в яких різниця між середнім та медіаною дорівнює нулю:
Empty DataFrame
Columns: [mean, median, diff]
Index: []
```

Тепер знайдемо регіони з найменшою різницею між mean та median.

```
print(compare_mean_median.iloc[(compare_mean_median['diff']).abs().argsort()].head(3)
)
```

```
                              mean        median         diff
Region
Latin America & Caribbean  10485.343136  10833.201075  -347.857939
South Asia                  2795.213935   1576.608412  1218.605523
Sub-Saharan Africa          2878.665521   1034.390361  1844.275160
```

Виведемо топ 5 країн та 5 останніх країн по ВВП та кількості CO2 на душу населення.

```
sorted_df_by_gdp = df.sort_values(by='GDP per capita', ascending=False)
print('Топ-5 країн по ВВП на душу населення:')
print(sorted_df_by_gdp.head())
```

```
Топ-5 країн по ВВП на душу населення:
         Country Name                 Region  GDP per capita  Population  \
115        Luxembourg  Europe & Central Asia    100738.68420    582972.0
188       Switzerland  Europe & Central Asia     79887.51824   8372098.0
116  Macao SAR, China    East Asia & Pacific     74017.18471    612167.0
146            Norway  Europe & Central Asia     70868.12250   5232929.0
92            Ireland  Europe & Central Asia     64175.43824   4773095.0

     CO2 emission      Area  Population Density  CO2 emission per capita
115      9658.878    2590.0          225.085714                 0.016568
188     35305.876   41290.0          202.763333                 0.004217
116      1283.450      30.3        20203.531353                 0.002097
146     47626.996  385178.0           13.585742                 0.009101
92      34066.430   70280.0           67.915410                 0.007137
```

```
print('Топ-5 країн з найменшим ВВП на душу населення:')
print(sorted_df_by_gdp.tail())
```

```
Топ-5 країн з найменшим ВВП на душу населення:
                Country Name              Region  GDP per capita  Population  \
118               Madagascar  Sub-Saharan Africa      401.742270  24894551.0
37   Central African Republic  Sub-Saharan Africa      382.213174   4594621.0
134               Mozambique  Sub-Saharan Africa      382.069330  28829476.0
119                   Malawi  Sub-Saharan Africa      300.307665  18091575.0
31                   Burundi  Sub-Saharan Africa      285.727442  10524117.0

     CO2 emission      Area  Population Density  CO2 emission per capita
118      3076.613  587295.0           42.388495                 0.000124
37        300.694  622980.0            7.375230                 0.000065
134      8426.766  799380.0           36.064795                 0.000292
119      1276.116  118480.0          152.697291                 0.000071
31        440.040   27830.0          378.157276                 0.000042
```

```
df['CO2 emission per capita'] = df['CO2 emission'] / df['Population']

sorted_df_by_CO2 = df.sort_values(['CO2 emission per capita'], ascending=False)
print('Топ-5 країн з найбільшою кількістю CO2 на душу населення:')
print(sorted_df_by_CO2.head())
```

```
Топ-5 країн з найбільшою кількістю CO2 на душу населення:
                Country Name                     Region  GDP per capita  \
182  St. Martin (French part)  Latin America & Caribbean    13445.593416
163               San Marino      Europe & Central Asia    47908.561410
130                   Monaco      Europe & Central Asia    13445.593416
145  Northern Mariana Islands        East Asia & Pacific    22572.378820
3              American Samoa        East Asia & Pacific    11834.745230

     Population   CO2 emission   Area  Population Density  \
182     31949.0  165114.116337   54.4          587.297794
163     33203.0  165114.116337   60.0          553.383333
130     38499.0  165114.116337    2.0        19249.500000
145     55023.0  165114.116337  460.0          119.615217
3       55599.0  165114.116337  200.0          277.995000

     CO2 emission per capita
182                 5.168053
163                 4.972867
130                 4.288790
145                 3.000820
3                   2.969732
```

```
print('Топ-5 країн з найменшою кількістю CO2 на душу населення:')
print(sorted_df_by_CO2.tail())
```

```
Топ-5 країн з найменшою кількістю CO2 на душу населення:
         Country Name              Region  GDP per capita    Population  \
44   Congo, Dem. Rep.  Sub-Saharan Africa      405.542501  7.873615e+07
38               Chad  Sub-Saharan Africa      664.295652  1.445254e+07
175           Somalia  Sub-Saharan Africa      434.208810  1.431800e+07
31            Burundi  Sub-Saharan Africa      285.727442  1.052412e+07
61            Eritrea  Sub-Saharan Africa    13445.593416  3.432256e+07

     CO2 emission       Area  Population Density  CO2 emission per capita
44       4671.758  2344860.0           33.578189                 0.000059
38        729.733  1284000.0           11.255875                 0.000050
175       608.722   637660.0           22.453966                 0.000043
31        440.040    27830.0          378.157276                 0.000042
61        696.730   117600.0          291.858502                 0.000020
```

Висновок

У цій лабораторній роботі я використала Pandas у роботі з даними. Вхідні дані було приведено до коректного формату після трансформації у датафрейм, були побудовані гістограми та діаграми розмаху. Також я проаналізувала дані, визначивши:

1. Яка країна має найбільший ВВП на людину (GDP per capita), яка має найменшу площу.
2. В якому регіоні середня площа країни найбільша.
3. Країну з найбільшою щільністю населення у світі. У Європі та центральній Азії.
4. Чи співпадає в якомусь регіоні середнє та медіана ВВП.
5. Топ 5 країн та 5 останніх країн по ВВП та кількості CO2 на душу населення.

Я вкотре переконалася, що Pandas – потужний ісінтрумент у роботі з даними.