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, Бондаренко Максим Вікторович
(шифр, прізвище, ім'я, по батькові)

Перевірив Олійник Юрій Олександрович
(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.

Основне завдання

DataFrame та його структура

За допомогою Python бібліотеки Pandas завантажимо дані з даного csv файлу в dataframe та досліджуємо структуру наших даних, використовуючи скрипти нижче.

```
[1]: import pandas as pd
     import matplotlib.pyplot as plt

     df = pd.read_csv('Data2.csv', delimiter=';', decimal=',')
     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   Populatiion     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
```

На даному рисунку можна помітити загальну інформацію про датафрейм: кількість рядків та колонок, назви всіх колонок, кількість записів в кожній з них, тип даних колонки та використання пам’яті.

Виправлення помилок

Щодо базової обробки датафрейму, змінимо назву колонки ‘Populatiion’ на ‘Population’ скриптом нижче, оскільки вона містить помилку в назві.

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

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

Знайдемо рядки, поля яких містять від’ємні елементи та виведемо їх.

```
[4]: mask = (df.select_dtypes(include=[float]) < 0).any(axis=1)
     result = df[mask]
     print(result)
```

```
          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
```

Виправимо всі існуючі від'ємні значення та виведемо їх знову, щоб перевірити внесені зміни.

```
[5]: for col in df.columns:
         if df[col].dtype == float:
             df[col] = df[col].abs()

     print(df.loc[[56,135]])
```

```
          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
```

Також в даних наявні пусті елементи, які потрібно замінити середніми по стовпчику.

```
[3]: for col in df.columns:
         if df[col].dtype == float:
             col_mean = df[col].mean()
             df[col] = df[col].fillna(col_mean)

     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    13374.833168
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
0    34656032.0    9809.225000   652860.0
1     2876101.0    5716.853000    28750.0
2    40606052.0  145400.217000  2381740.0
3       55599.0  165114.116337      200.0
4       77281.0     462.042000      470.0
..          ...            ...        ...
212    102951.0  165114.116337      350.0
213   4551566.0  165114.116337     6020.0
214  27584213.0   22698.730000   527970.0
215  16591390.0    4503.076000   752610.0
216  16150362.0   12020.426000   390760.0

[217 rows x 6 columns]
```

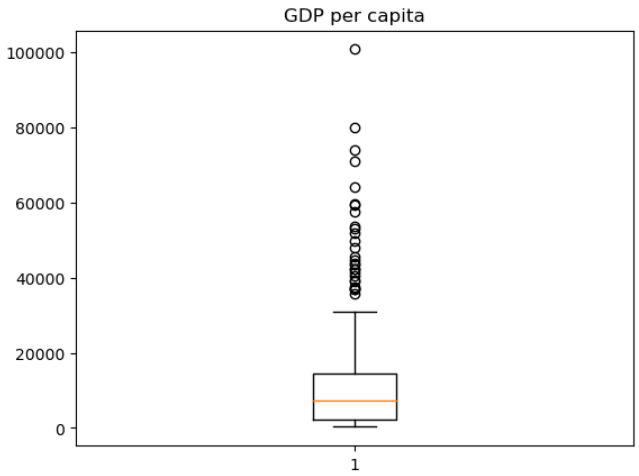
Діаграми розмаху та гістограми

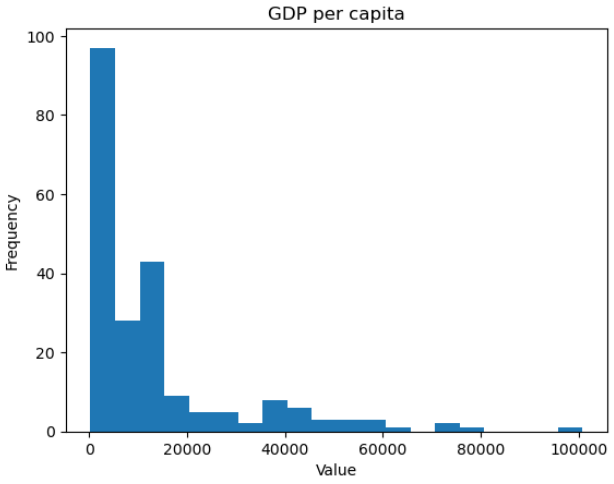
Виведемо діаграми розмаху та гістограми для кожного стовпця з чисельними даними.

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

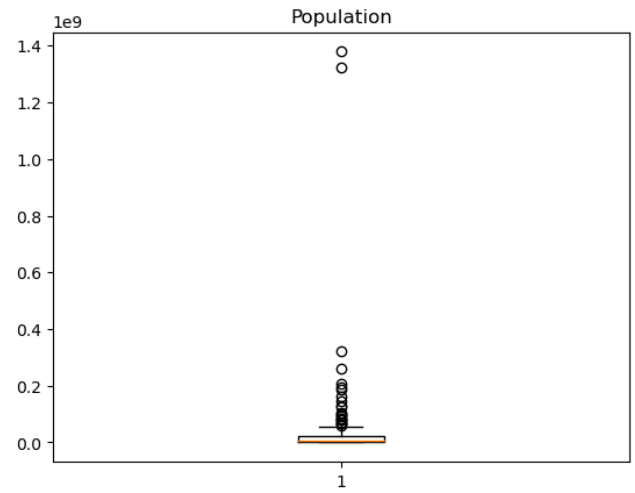
```
[7]: for column in df.columns:
         if df[column].dtype == float:
             plt.hist(df[column].dropna(), bins=20)
             plt.title(column)
             plt.xlabel('Value')
             plt.ylabel('Frequency')
             plt.show()
```

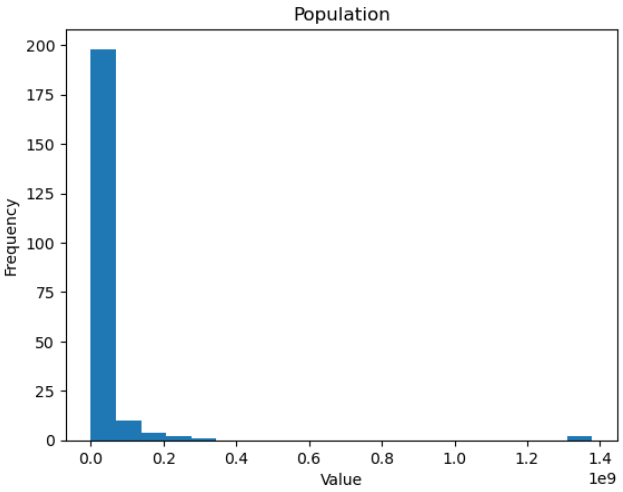
Діаграма розмаху та гістограма для ВВП на душу населення



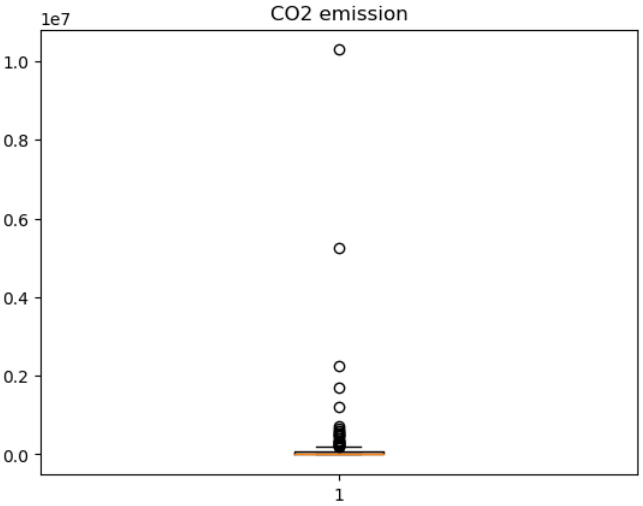


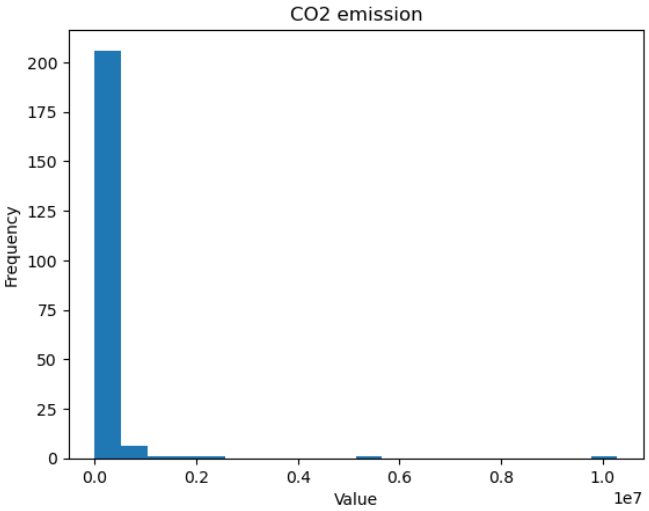
Діаграма розмаху та гістограма для кількості населення



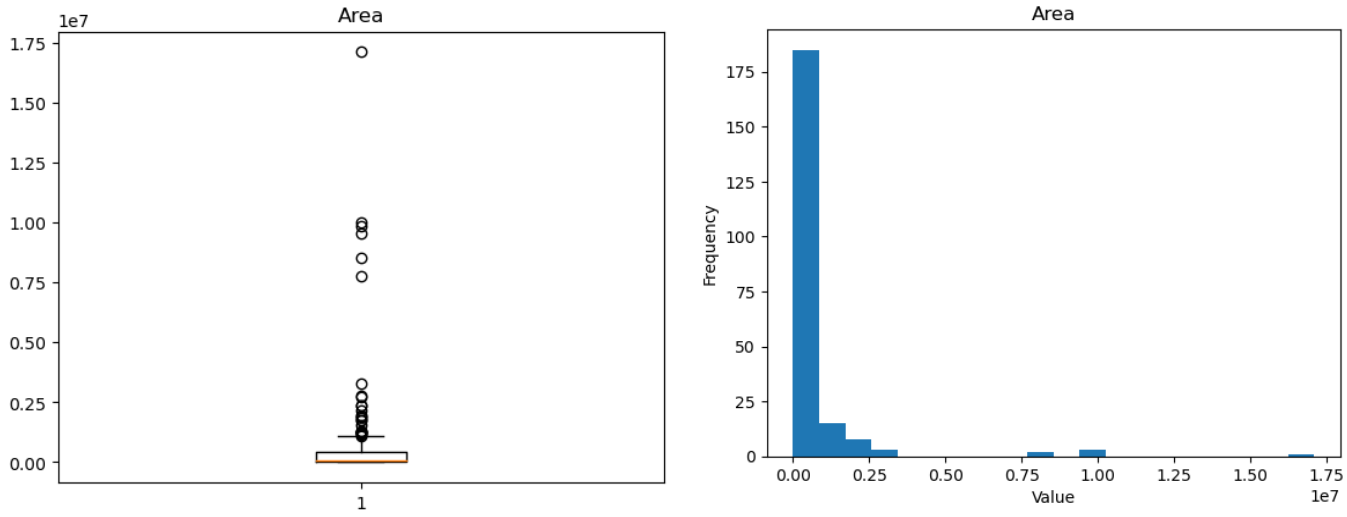


Діаграма розмаху та гістограма для кількості викидів CO_2





Діаграма розмаху та гістограма для площі країн



Додавання стовпчику із щільністю населення

Додаємо стовпчик із щільністю населення кожної країни, який є просто представленням кількості населення поділеного на площу країни.

[8]:
```
df["Population Density"] = df["Population"] / df["Area"]
df.head(1)
```

[8]:

	Country Name	Region	GDP per capita	Population	CO2 emission	Area	Population Density
0	Afghanistan	South Asia	561.778746	34656032.0	9809.225	652860.0	53.083405

Додаткове завдання

Заміна пропущених значень

Демонстрація заміни пропущених значень описана в розділі «Виправлення помилок».

Країна з найбільшим ВВП на людину, з найменшою площею

Виведемо країну з найбільшим ВВП на душу населення та країну з найменшою площею.

[9]:
```
max_gdp_index = df['GDP per capita'].idxmax()
max_gdp_country = df.loc[max_gdp_index, 'Country Name']
print(f"The country with the largest GDP per capita in the world is {max_gdp_country}.")

min_area_index = df['Area'].idxmin()
min_area_country = df.loc[min_area_index, 'Country Name']
print(f"The country with the smallest area is {min_area_country}.")
```

```
The country with the largest GDP per capita in the world is Luxembourg.
The country with the smallest area is Monaco.
```

Регіон з найбільшою середньою площею країн

[10]:
```
region_mean_area = df.groupby('Region')['Area'].mean()
largest_region = region_mean_area.idxmax()
print("The region with the largest average area is", largest_region)
```

```
The region with the largest average area is North America
```

Країна з найбільшою щільністю населення у світі, у Європі та центральній Азії

[11]:
```
df_sorted = df.sort_values(by='Population Density', ascending=False)
print('Country with highest population density is', df_sorted.iloc[0]['Country Name'])

df_eu_ca = df[df['Region'] == 'Europe & Central Asia']
df_eu_ca_sorted = df_eu_ca.sort_values(by='Population Density', ascending=False)
print('Country with highest population density in Europe and Central Asia is', df_eu_ca_sorted.iloc[0]['Country Name'])
```

```
Country with highest population density is Macao SAR, China
Country with highest population density in Europe and Central Asia is Monaco
```

Співпадіння середнього та медіани ВВП по регіонам

Для початку розрахуємо загальне ВВП для кожної країни та створимо окрему колонку для цих даних.

[12]:
```
df["Total GDP"] = df["GDP per capita"] * df["Population"]
df.head(1)
```

[12]:

	Country Name	Region	GDP per capita	Population	CO2 emission	Area	Population Density	Total GDP
0	Afghanistan	South Asia	561.778746	34656032.0	9809.225	652860.0	53.083405	1.946902e+10

Розрахуємо середнє та медіану для кожного регіону окремо та порівняємо їх. Не існує жодного регіону, де ці параметри були б рівними.

[13]:
```
region_stats = df.groupby('Region')['Total GDP'].agg(['mean', 'median'])

for region in region_stats.index:
    mean = region_stats.loc[region, 'mean']
    median = region_stats.loc[region, 'median']
    if mean == median:
        print(f"The mean and median GDP in {region} are equal ({mean}).")
    else:
        print(f"The mean and median GDP in {region} are different (mean = {mean}, median = {median}).")
```

```
The mean and median GDP in East Asia & Pacific are different (mean = 601314797021.4976, median = 11400653732.56196).
The mean and median GDP in Europe & Central Asia are different (mean = 349091144622.72107, median = 49052249268.26028).
The mean and median GDP in Latin America & Caribbean are different (mean = 128573963145.39444, median = 13643876718.90971).
The mean and median GDP in Middle East & North Africa are different (mean = 161162758088.2565, median = 102047824411.42694).
The mean and median GDP in North America are different (mean = 6718676588591.594, median = 1530680973899.0176).
The mean and median GDP in South Asia are different (mean = 361745128122.7743, median = 52017740706.313446).
The mean and median GDP in Sub-Saharan Africa are different (mean = 49945863170.15772, median = 10981369640.35254).
```

Топ 5 країн та 5 останніх країн по ВВП та кількості CO2 на душу населення

Для початку розрахуємо кількість викидів CO2 на душу населення для кожної країни

[14]:
```
df['CO2 emission per capita'] = df['CO2 emission'] / df['Population']
df.head(1)
```

[14]:

	Country Name	Region	GDP per capita	Population	CO2 emission	Area	Population Density	Total GDP	CO2 emission per capita
0	Afghanistan	South Asia	561.778746	34656032.0	9809.225	652860.0	53.083405	1.946902e+10	0.000283

Виведемо 5 країн з найбільшою кількістю ВВП на душу населення та 5 з найменшою

[15]:
```
df_sorted = df.sort_values(['GDP per capita'], ascending=False)
print('Top 5 countries by GDP per capita:\n', df_sorted.head()[['Country Name', 'GDP per capita']],
      '\nThe last 5 countries by GDP per capita:\n', df_sorted.tail()[['Country Name', 'GDP per capita']])
```

```
Top 5 countries by GDP per capita:
          Country Name  GDP per capita
115        Luxembourg    100738.68420
188       Switzerland     79887.51824
116  Macao SAR, China     74017.18471
146            Norway     70868.12250
92            Ireland     64175.43824
The last 5 countries by GDP per capita:
                  Country Name  GDP per capita
118                Madagascar      401.742270
37   Central African Republic      382.213174
134                Mozambique      382.069330
119                    Malawi      300.307665
31                    Burundi      285.727442
```

Виведемо 5 країн з найбільшою кількістю викидів CO2 на душу населення та 5 з найменшою

[16]:
```
df_sorted = df.sort_values(['CO2 emission per capita'], ascending=False)
print('Top 5 countries by CO2 emission per capita:\n', df_sorted.head()[['Country Name', 'CO2 emission per capita']],
    '\nThe last 5 countries by CO2 emission per capita:\n', df_sorted.tail()[['Country Name', 'CO2 emission per capita']])
```

```
Top 5 countries by CO2 emission per capita:
                  Country Name  CO2 emission per capita
182  St. Martin (French part)                 5.168053
163               San Marino                 4.972867
130                   Monaco                 4.288790
145  Northern Mariana Islands                3.000820
3             American Samoa                 2.969732
The last 5 countries by CO2 emission per capita:
          Country Name  CO2 emission per capita
44   Congo, Dem. Rep.                 0.000059
38               Chad                 0.000050
175           Somalia                 0.000043
31            Burundi                 0.000042
61            Eritrea                 0.000020
```

Висновок

У цьому комп’ютерному практикуму було вивчено можливості Python, а саме Pandas у роботі з даними. Вхідні дані було записано в DataFrame, структуру якого було вивчено та помічено нецілісність даних, тому я почистив дані від від’ємних значень, нульові замінив середніми для більш об’єктивної побудови гістограм та діаграм розмаху. На діаграмах розмаху було помічено великий розмах між даними. Наприклад, на діаграмі населення є дві країни з кількістю населення значно більшою за всі інші, так само і з викидами CO2, дані з ВВП на душу населення є найбільш кучними. Було визначено країну з найбільшим ВВП на душу населення у світі, з найменшою площею території, регіон з найбільшою середньою площею країн, країни з найбільшою густиною населення у світі та окремо в регіоні «Європа та центральна Азія». Регіонів з однаковими середньою та медіаною ВВП країн не виявилось, усі мають різні. Також було виведено 5 країн з найбільшим та найменшим ВВП на душу населення та 5 з найбільшою та найменшою кількістю викидів CO2.