

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 лабораторної роботи №3 з дисципліни
«Аналіз даних в інформаційних системах»

„Описова статистика”

Виконав(ла)

ІП-11 Тарасюнок Дмитро Євгенович
(шифр, прізвище, ім'я, по батькові)

Перевірила

Ліхоузова Т. А.
(прізвище, ім'я, по батькові)

Київ 2023

ЗМІСТ

1	Мета лабораторної роботи.....	3
2	Завдання.....	4
2.1	Основне завдання.....	4
2.2	Додаткове завдання.....	4
3	Виконання основного завдання	5
3.1	Записати дані у data frame	5
3.2	Дослідити структуру даних.....	5
3.3	Виправити помилки в даних	5
3.4	Побудувати діаграми розмаху та гістограми	7
3.5	Додати стовпчик із щільністю населення	9
4	Виконання додаткового завдання.....	10
4.1	Чи є пропущені значення? Якщо є, замінити середніми	10
4.2	Яка країна має найбільший ВВП на людину (GDP per capita)? Яка має найменшу площу?.....	12
4.3	В якому регіоні середня площа країни найбільша?	13
4.4	Знайдіть країну з найбільшою щільністю населення у світі? У Європі та центральній Азії?	13
4.5	Чи співпадає в якомусь регіоні середнє та медіана ВВП?.....	14
4.6	Вивести топ 5 країн та 5 останніх країн по ВВП та кількості CO2 на душу населення.	16
5	Висновок.....	19

1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

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

2 ЗАВДАННЯ

2.1 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

2.2 Додаткове завдання

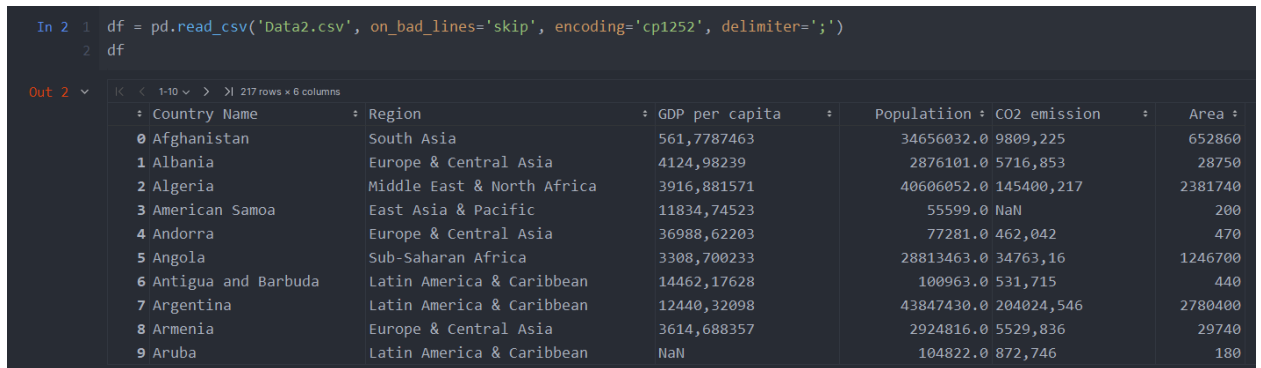
Відповісти на питання (файл Data2.csv):

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 CO₂ на душу населення.

3 ВИКОНАННЯ ОСНОВНОГО ЗАВДАННЯ

3.1 Записати дані у data frame

Зробити дану операцію можна за допомогою функції `read_csv` пакету `pandas`. На рисунку 3.1 наведено завантажений data frame.



```
In 2 1 df = pd.read_csv('Data2.csv', on_bad_lines='skip', encoding='cp1252', delimiter=',')
2 df
```

Out 2

	Country Name	Region	GDP per capita	Population	CO2 emission	Area
0	Afghanistan	South Asia	561,7787463	34656032.0	9809,225	652860
1	Albania	Europe & Central Asia	4124,98239	2876101.0	5716,853	28750
2	Algeria	Middle East & North Africa	3916,881571	40606052.0	145400,217	2381740
3	American Samoa	East Asia & Pacific	11834,74523	55599.0	NaN	200
4	Andorra	Europe & Central Asia	36988,62203	77281.0	462,042	470
5	Angola	Sub-Saharan Africa	3308,700233	28813463.0	34763,16	1246700
6	Antigua and Barbuda	Latin America & Caribbean	14462,17628	100963.0	531,715	440
7	Argentina	Latin America & Caribbean	12440,32098	43847430.0	204024,546	2780400
8	Armenia	Europe & Central Asia	3614,688357	2924816.0	5529,836	29740
9	Aruba	Latin America & Caribbean	NaN	104822.0	872,746	180

Рис. 3.1 – Data Frame

3.2 Дослідити структуру даних

Було детально досліджено структуру даних: data frame містить 6 стовпців: назву країни, регіон, ВВП на душу населення, кількість населення, кількість викидів CO2, а також площу країни. Було помічено, що дійсні числа записані через роздільник «кома», що сприймається пакетом як рядок, було виявлено некоректно названий стовпець населення: «Populatiion» містить дві літери «і». Також помічено, що стовпці ВВП на душу населення й площі мають від'ємні значення, які, очевидно, бути такими не можуть.

3.3 виправити помилки в даних

Спочатку було перейменовано стовпці: записано їх у форматі `snake_case` для зручності доступу до них як до атрибутів об'єкту, а також виправлено помилку в написанні назви стовпця кількості населення. На рисунку 3.2 наведено код та результат.

```
In 3 1 df.columns = ['country_name', 'region', 'gdp_per_capita', 'population', 'co2_emission', 'area']
      2 df
```

Out 3

	country_name	region	gdp_per_capita	population	co2_emission	area
0	Afghanistan	South Asia	561,7787463	34656032.0	9809,225	652860
1	Albania	Europe & Central Asia	4124,98239	2876101.0	5716,853	28750
2	Algeria	Middle East & North Africa	3916,881571	40606052.0	145400,217	2381740
3	American Samoa	East Asia & Pacific	11834,74523	55599.0	NaN	200
4	Andorra	Europe & Central Asia	36988,62203	77281.0	462,042	470
5	Angola	Sub-Saharan Africa	3308,700233	28813463.0	34763,16	1246700
6	Antigua and Barbuda	Latin America & Caribbean	14462,17628	100963.0	531,715	440
7	Argentina	Latin America & Caribbean	12440,32098	43847430.0	204024,546	2780400
8	Armenia	Europe & Central Asia	3614,688357	2924816.0	5529,836	29740
9	Aruba	Latin America & Caribbean	NaN	104822.0	872,746	180

Рис. 3.2 – Перейменування стовпців

Далі було встановлено назву країни в якості індексу, оскільки пакет pandas за замовчуванням в якості індексу встановлює номер рядку.

```
In 4 1 df.set_index('country_name', inplace=True)
      2 df
```

Out 4

	country_name	region	gdp_per_capita	population	co2_emission	area
	Afghanistan	South Asia	561,7787463	34656032.0	9809,225	652860
	Albania	Europe & Central Asia	4124,98239	2876101.0	5716,853	28750
	Algeria	Middle East & North Africa	3916,881571	40606052.0	145400,217	2381740
	American Samoa	East Asia & Pacific	11834,74523	55599.0	NaN	200
	Andorra	Europe & Central Asia	36988,62203	77281.0	462,042	470
	Angola	Sub-Saharan Africa	3308,700233	28813463.0	34763,16	1246700
	Antigua and Barbuda	Latin America & Caribbean	14462,17628	100963.0	531,715	440
	Argentina	Latin America & Caribbean	12440,32098	43847430.0	204024,546	2780400
	Armenia	Europe & Central Asia	3614,688357	2924816.0	5529,836	29740
	Aruba	Latin America & Caribbean	NaN	104822.0	872,746	180

Рис. 3.3 – Встановлення індексу

Після цього було замінено коми в значеннях на крапки й приведено до дійсночисельного типу.

```
In 88 1 df.gdp_per_capita = df.gdp_per_capita.astype(str).str.replace(',', '.').astype(float)
        2 df.co2_emission = df.co2_emission.astype(str).str.replace(',', '.').astype(float)
        3 df.area = df.area.astype(str).str.replace(',', '.').astype(float)
        4 df
```

Out 88

	country_name	region	gdp_per_capita	population	co2_emission	area
	Afghanistan	South Asia	561.778746	34656032.0	9809.225	652860.0
	Albania	Europe & Central Asia	4124.982390	2876101.0	5716.853	28750.0
	Algeria	Middle East & North Africa	3916.881571	40606052.0	145400.217	2381740.0
	American Samoa	East Asia & Pacific	11834.745230	55599.0	NaN	200.0
	Andorra	Europe & Central Asia	36988.622030	77281.0	462.042	470.0

	Virgin Islands (U.S.)	Latin America & Caribbean	NaN	102951.0	NaN	350.0
	West Bank and Gaza	Middle East & North Africa	2943.404534	4551566.0	NaN	6020.0
	Yemen, Rep.	Middle East & North Africa	990.334774	27584213.0	22698.730	527970.0
	Zambia	Sub-Saharan Africa	1269.573537	16591390.0	4503.076	752610.0
	Zimbabwe	Sub-Saharan Africa	1029.076649	16150362.0	12020.426	390760.0

Рис. 3.4 – Виправлення дійсних чисел

Можемо перевірити типи стовпців за допомогою виклику методу «dtypes».

```
In 89 1 df.dtypes

Out 89  region      object
        gdp_per_capita  float64
        population     float64
        co2_emission   float64
        area           float64
        dtype: object
```

Рис. 3.5 – Виведення типів даних кожного стовпця

Після цього від’ємні значення було перетворено на додатні шляхом виклику методу модуля

```
In 7 1 df.gdp_per_capita = df.gdp_per_capita.abs()
      2 df.area = df.area.abs()
      3 df

Out 7  |< 1-10 >| 217 rows x 5 columns
      country_name      : region      :      gdp_per_capita :      population :      co2_emission :      area :
Afghanistan      South Asia      561.778746      34656032.0      9809.225      652860.0
Albania      Europe & Central Asia      4124.982390      2876101.0      5716.853      28750.0
Algeria      Middle East & North Africa      3916.881571      40606052.0      145400.217      2381740.0
American Samoa      East Asia & Pacific      11834.745230      55599.0      NaN      200.0
Andorra      Europe & Central Asia      36988.622030      77281.0      462.042      470.0
Angola      Sub-Saharan Africa      3308.700233      28813463.0      34763.160      1246700.0
Antigua and Barbuda      Latin America & Caribbean      14462.176280      100963.0      531.715      440.0
Argentina      Latin America & Caribbean      12440.320980      43847430.0      204024.546      2780400.0
Armenia      Europe & Central Asia      3614.688357      2924816.0      5529.836      29740.0
Aruba      Latin America & Caribbean      NaN      104822.0      872.746      180.0
```

Рис. 3.6 – Заміна від’ємних значень ВВП на душу населення та площі на значення по модулю

3.4 Побудувати діаграми розмаху та гістограми

Побудувати діаграми об’єктів класу DataFrame пакету pandas можна дуже легко: треба просто викликати метод об’єкту. Для побудови гістограм таким методом є `.hist()`, а для діаграми розмаху - `.boxplot()`. Для обох діаграм задається розмір: 20 по висоті й 12 по ширині для кращого відображення діаграм.

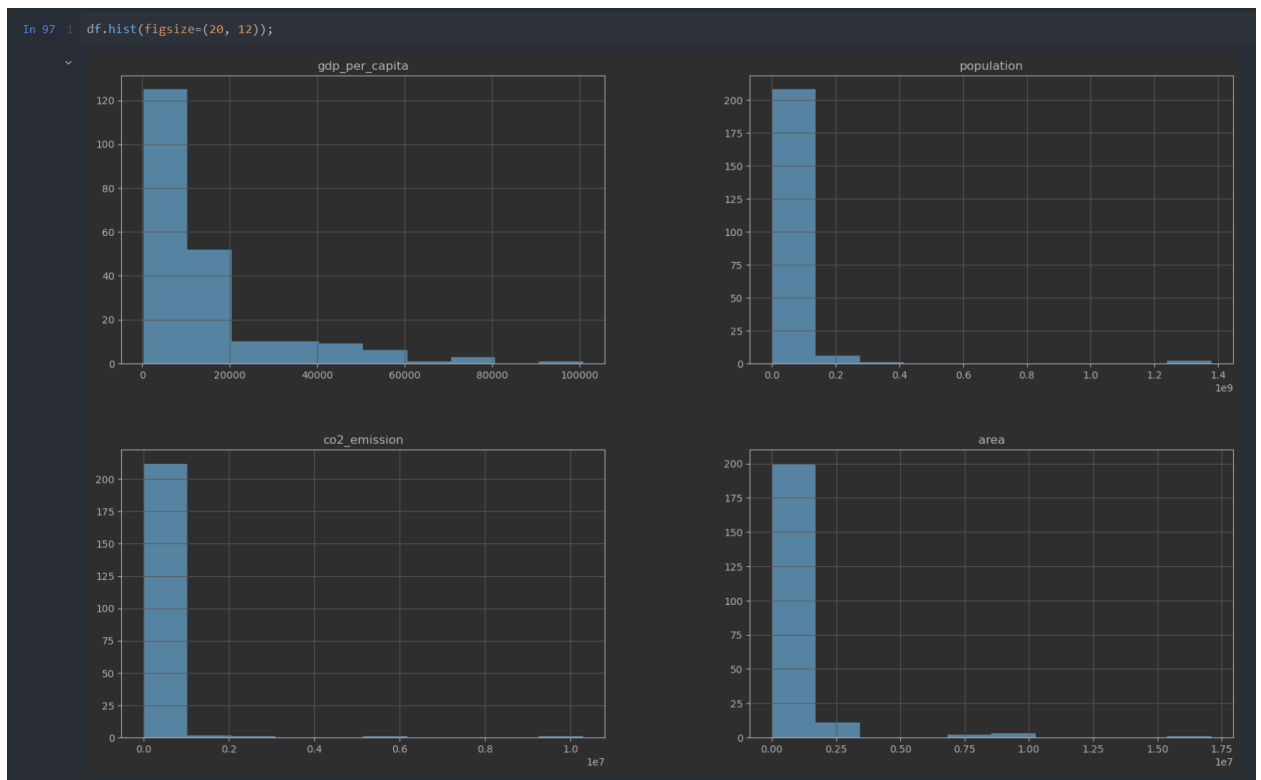


Рис. 3.7 – Побудова гістограм

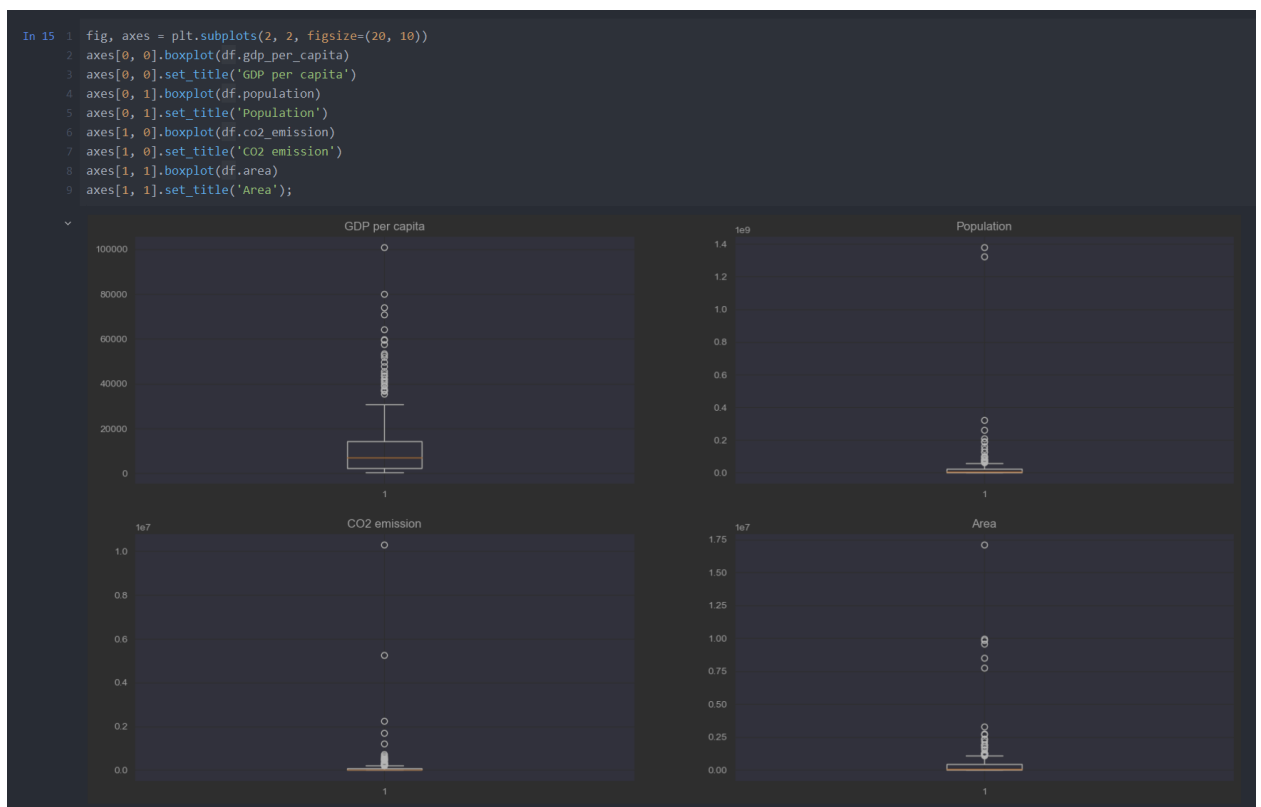


Рис. 3.8 – Побудова діаграм розмаху

3.5 Додати стовпчик із щільністю населення

Зробити це можна дуже легко: треба просто провести операцію над стовпцями населення та площі країн і вказати назву нового стовпця так, ніби ми просто до нього звертаємося.

```
In 16 1 df['density'] = df.population / df.area
      2 df
```

Out 16

country_name	region	gdp_per_capita	population	co2_emission	area	density
Afghanistan	South Asia	561.778746	34656032	9809.225000	652860.0	53.083405
Albania	Europe & Central Asia	4124.982390	2876101	5716.853000	28750.0	100.038296
Algeria	Middle East & North Africa	3916.881571	40606052	145400.217000	2381740.0	17.048902
American Samoa	East Asia & Pacific	11834.745230	55599	165114.116337	200.0	277.995000
Andorra	Europe & Central Asia	36988.622030	77281	462.042000	470.0	164.427660
Angola	Sub-Saharan Africa	3308.700233	28813463	34763.160000	1246700.0	23.111786
Antigua and Barbuda	Latin America & Caribbean	14462.176280	100963	531.715000	440.0	229.461364
Argentina	Latin America & Caribbean	12440.320980	43847430	204024.546000	2780400.0	15.770188
Armenia	Europe & Central Asia	3614.688357	2924816	5529.836000	29740.0	98.346200
Aruba	Latin America & Caribbean	13445.593416	104822	872.746000	180.0	582.344444

Рис. 3.9 – Додавання стовпця із щільністю населення

4 ВИКОНАННЯ ДОДАТКОВОГО ЗАВДАННЯ

4.1 Чи є пропущені значення? Якщо є, замінити середніми

Варто зазначити, що виконання таких операцій, як побудова діаграм розмаху та створення стовпця щільності населення було б неможливим без заміни пропущених значень. Перевірити, чи є пропущені значення в кожному зі стовпців, можна просто викликавши метод `.isna().any()` об'єкту типу `DataFrame`.

```
In 10 1 df.isna().any()
```

Out 10 ▾

	data
region	False
gdp_per_capita	True
population	True
co2_emission	True
area	False

Рис. 3.10 – Перевірка, які стовпці містять пропущені значення

Як бачимо, пропущенні значення є в стовпцях ВВП на душу населення, кількості населення та кількості викидів CO2. Замінити пропущенні значення можна шляхом виклику методу `.fillna()` із вказанням значення, на яке треба замінити пропущене значення: за постановкою задачі, це середнє, а отже обрахуємо його шляхом виклику методу `.mean()`. У методі `.fillna()` встановимо аргумент `inplace=True`, щоб ці значення замінилися безпосередньо в `DataFrame` без зайвих операцій присвоєння.

In 11

```

1 df.gdp_per_capita.fillna(df.gdp_per_capita.mean(), inplace=True)
2 df.population.fillna(df.population.mean(), inplace=True)
3 df.co2_emission.fillna(df.co2_emission.mean(), inplace=True)
4 df

```

Out 11

country_name	region	gdp_per_capita	population	co2_emission	area
Afghanistan	South Asia	561.778746	34656032.0	9809.225000	652860.0
Albania	Europe & Central Asia	4124.982390	2876101.0	5716.853000	28750.0
Algeria	Middle East & North Africa	3916.881571	40606052.0	145400.217000	2381740.0
American Samoa	East Asia & Pacific	11834.745230	55599.0	165114.116337	200.0
Andorra	Europe & Central Asia	36988.622030	77281.0	462.042000	470.0
Angola	Sub-Saharan Africa	3308.700233	28813463.0	34763.160000	1246700.0
Antigua and Barbuda	Latin America & Caribbean	14462.176280	100963.0	531.715000	440.0
Argentina	Latin America & Caribbean	12440.320980	43847430.0	204024.546000	2780400.0
Armenia	Europe & Central Asia	3614.688357	2924816.0	5529.836000	29740.0
Aruba	Latin America & Caribbean	13445.593416	104822.0	872.746000	180.0

Рис. 3.11 – Заміна пропущених значень середніми значеннями

Можна після цього перевірити, що пропущенні значення відсутні викликавши вже згадані методи

In 12

```

1 df.isna().any()

```

Out 12

	data
region	False
gdp_per_capita	False
population	False
co2_emission	False
area	False

Рис. 3.12 – Перевірка, які стовпці містять пропущені значення

Також оскільки кількість населення – значення завжди ціле, а пропущенні значення відсутні (вони повинні обов’язково мати тип дійсного числа), можемо встановити тип стовпця населення як ціле число.

In 13 1 df.population = df.population.astype(int)
2 df

Out 13 217 rows x 5 columns

country_name	region	gdp_per_capita	population	co2_emission	area
Afghanistan	South Asia	561.778746	34656032	9809.225000	652860.0
Albania	Europe & Central Asia	4124.982390	2876101	5716.853000	28750.0
Algeria	Middle East & North Africa	3916.881571	40606052	145400.217000	2381740.0
American Samoa	East Asia & Pacific	11834.745230	55599	165114.116337	200.0
Andorra	Europe & Central Asia	36988.622030	77281	462.042000	470.0
Angola	Sub-Saharan Africa	3308.700233	28813463	34763.160000	1246700.0
Antigua and Barbuda	Latin America & Caribbean	14462.176280	100963	531.715000	440.0
Argentina	Latin America & Caribbean	12440.320980	43847430	204024.546000	2780400.0
Armenia	Europe & Central Asia	3614.688357	2924816	5529.836000	29740.0
Aruba	Latin America & Caribbean	13445.593416	104822	872.746000	180.0

Рис. 3.13 – Перетворення типу кількості населення

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

Взнати, яка країна має найбільший ВВП на душу населення, можна викликавши спочатку метод `.idxmax()` для стовпця, за яким шукаємо максимальне значення, а потім викликати метод `.loc[]` нашого data frame, щоб відобразити повну інформацію щодо цієї країни.

In 17 1 df.loc[[df.gdp_per_capita.idxmax()]]

Out 17 1 row x 6 columns

country_name	region	gdp_per_capita	population	co2_emission	area	density
Luxembourg	Europe & Central Asia	100738.6842	582972	9658.878	2590.0	225.085714

Рис. 3.14 – Виведення країни з найвищим ВВП на душу населення

Як бачимо, найвищий показник ВВП на душу населення в Люксембурзі. Проведемо такі ж операції для знаходження країни з найменшою площею, зміниться тільки метод: `.idxmax()` на `.idxmin()`.

In 18 1 df.loc[[df.area.idxmin()]]

Out 18 1 row x 6 columns

country_name	region	gdp_per_capita	population	co2_emission	area	density
Monaco	Europe & Central Asia	13445.593416	38499	165114.116337	2.0	19249.5

Рис. 3.15 – Виведення країни з найменшою площею

Як бачимо, Монако має найменшу площу серед нашого набору даних. Одразу ж можна зробити висновок про неповноту нашого набору даних, оскільки насправді країною з найменшою площею є Ватикан, а не Монако, але ця країна пропущена в нашому наборі даних.

4.3 В якому регіоні середня площа країни найбільша?

Ця операція вже є складнішою. Треба спочатку згрупувати країни за регіоном (метод `.groupby()`), знайти середнє значення площі (`.area.mean()`), а потім вивести інформацію про цей континент, використавши вже згадані методи `.loc[]` та `.idxmax()`.

```
In 19 1 group = df.groupby('region').area.mean()
      2 group.loc[[group.idxmax()]]
```

Out 19 ▾

< < 1 row ▾ > > Length: 1, dtype: float64	
region	area
North America	6605410.0

Рис. 3.16 – Виведення регіону із найбільшою середньою площею країн
Як бачимо, середня площа країн є найбільшою в Північній Америці.

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

Для таких операцій треба знову ж таки використати метод `.idxmax()` для стовпця щільності населення, а потім метод `.loc[]` для виведення повної інформації про цю країну

```
In 20 1 df.loc[[df.density.idxmax()]]
```

Out 20 ▾

< < 1 row ▾ > > 1 rows x 6 columns						
country_name	region	gdp_per_capita	population	co2_emission	area	density
Macao SAR, China	East Asia & Pacific	74017.18471	612167	1283.45	30.3	20203.531353

Рис. 3.17 – Виведення країни з найвищою щільністю населення

Як бачимо, Макао має найбільшу щільність населення у світі. Відфільтруємо країни за регіоном «Європа та Центральна Азія» й знайдемо країну з найбільшою щільністю населення, використавши ті ж самі вже згадані методи.

```
In 104: df.loc[[df[df.region == 'Europe & Central Asia'].density.idxmax()]]
```

Out 104: 1 row x 6 columns

country_name	region	gdp_per_capita	population	co2_emission	area	density
Monaco	Europe & Central Asia	13445.593416	38499	165114.116337	2.0	19249.5

Рис. 3.18 – Виведення країни з регіону «Європа та Центральна Азія» із найбільшою щільністю населення

Робимо висновок, що Монако має найбільшу щільність населення в даному регіоні.

4.5 Чи співпадає в якомусь регіоні середнє та медіана ВВП?

Зазначимо, що наш набір даних не містить стовпця із значенням ВВП, а отже його треба згенерувати за відомою формулою: кількість населення, помножена на ВВП на душу населення.

```
In 21: df['gdp'] = df.population * df.gdp_per_capita
df
```

Out 21: 217 rows x 7 columns

country_name	region	gdp_per_capita	population	co2_emission	area	density	gdp
Afghanistan	South Asia	561.778746	34656032	9809.225000	652860.0	53.083405	1.946902e+10
Albania	Europe & Central Asia	4124.982390	2876101	5716.853000	28750.0	100.038296	1.186387e+10
Algeria	Middle East & North Africa	3916.881571	40606052	145400.217000	2381740.0	17.048902	1.590491e+11
American Samoa	East Asia & Pacific	11834.745230	55599	165114.116337	200.0	277.995000	6.580000e+08
Andorra	Europe & Central Asia	36988.622030	77281	462.042000	470.0	164.427660	2.858518e+09
Angola	Sub-Saharan Africa	3308.700233	28813463	34763.160000	1246700.0	23.111786	9.535511e+10
Antigua and Barbuda	Latin America & Caribbean	14462.176280	100963	531.715000	440.0	229.461364	1.460145e+09
Argentina	Latin America & Caribbean	12440.320980	43847430	204024.546000	2780400.0	15.770188	5.454761e+11
Armenia	Europe & Central Asia	3614.688357	2924816	5529.836000	29740.0	98.346200	1.057230e+10
Aruba	Latin America & Caribbean	13445.593416	104822	872.746000	180.0	582.344444	1.409394e+09

Рис. 3.19 – Обрахунок ВВП країн

Далі згрупуємо країни за регіоном, викликавши вже згаданий метод `.groupby()` із аргументом `region` (стовпець групування), потім оберемо стовпець, для якого обраховуємо середнє та медіану – у нашому випадку це ВВП й обраховуємо дані метрики.

In 22	1	gdp_group = df.groupby('region').gdp.aggregate(['mean', 'median'])	
	2	gdp_group	
Out 22	▼	<div> <div> <</div> <div><</div> <div>7 rows ▼</div> <div>></div> <div>> </div> <div>7 rows × 2 columns</div> </div>	
		region	mean
		East Asia & Pacific	6.013644e+11
		Europe & Central Asia	3.490917e+11
		Latin America & Caribbean	1.286474e+11
		Middle East & North Africa	1.612492e+11
		North America	6.718678e+12
		South Asia	3.617451e+11
		Sub-Saharan Africa	5.004497e+10

Рис. 3.20 – Обрахунок середніх та медіанних значень ВВП для кожного з регіонів

Далі для зручності додамо стовпець різниці між середнім та медіаною шляхом обчислення модуля різниці середнього та медіани для кожного зі стовпців

In 23

1

gdp_group['difference'] = (gdp_group['mean'] - gdp_group['median']).abs()

2

gdp_group

Out 23

|<

<

7 rows ▼

>

>|

7 rows × 3 columns

region	mean	median	difference
East Asia & Pacific	6.013644e+11	1.140065e+10	5.899637e+11
Europe & Central Asia	3.490917e+11	4.905225e+10	3.000395e+11
Latin America & Caribbean	1.286474e+11	1.364388e+10	1.150036e+11
Middle East & North Africa	1.612492e+11	1.020478e+11	5.920142e+10
North America	6.718678e+12	1.530681e+12	5.187997e+12
South Asia	3.617451e+11	5.201774e+10	3.097274e+11
Sub-Saharan Africa	5.004497e+10	1.098137e+10	3.906360e+10

Рис 3.21 – Обчислення різниці між середнім та медіаною

Після цього необхідно порівняти стовпці середнього та медіани й вивести ті, у яких вони рівні.

```
In 27 1 gdp_group[gdp_group['mean'] == gdp_group['median']]
```

Out 27 0 rows x 3 columns

region	mean	median	difference
--------	------	--------	------------

Рис. 3.22 – Виведення регіонів, середнє яких дорівнює медіані

Очікувано, такі регіони відсутні, але можемо подивитися, де різниця найменша.

```
In 26 1 gdp_group.loc[[gdp_group['difference'].idxmin()]]
```

Out 26 1 row x 3 columns

region	mean	median	difference
Sub-Saharan Africa	5.004497e+10	1.098137e+10	3.906360e+10

Рис. 3.23 – Виведення регіону із найменшою різницею між середнім та медіаною

Як бачимо, найменша різниця між середнім та медіаною є в Субсахарській Африці.

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

Знову ж зазначимо, що в наборі даних відсутній стовпець кількості CO2 на душу населення, натомість є стовпці кількості CO2 на країну та кількість населення. Проведемо такі ж дії, як і з обрахунком щільності населення.

```
In 28 1 df['co2_emission_per_capita'] = df.co2_emission / df.population
```

Out 28 217 rows x 8 columns

country_name	region	gdp_per_capita	population	co2_emission	area	density	gdp	co2_emission_per_capita
Afghanistan	South Asia	561.778746	34656032	9809.225000	652860.0	53.083405	1.946902e+10	0.000283
Albania	Europe & Central Asia	4124.982390	2876101	5716.853000	28750.0	100.038296	1.186387e+10	0.001988
Algeria	Middle East & North Africa	3916.881571	40606052	145400.217000	2381740.0	17.048902	1.590491e+11	0.003581
American Samoa	East Asia & Pacific	11834.745230	55599	165114.116337	200.0	277.995000	6.580000e+08	2.969732
Andorra	Europe & Central Asia	36988.622030	77281	462.042000	470.0	164.427660	2.858518e+09	0.005979
Angola	Sub-Saharan Africa	3308.700233	28813463	34763.160000	1246700.0	23.111786	9.533511e+10	0.001206
Antigua and Barbuda	Latin America & Caribbean	14462.176280	100963	531.715000	440.0	229.461364	1.460145e+09	0.005266
Argentina	Latin America & Caribbean	12440.320980	43847430	204024.546000	2780400.0	15.770188	5.454761e+11	0.004653
Armenia	Europe & Central Asia	3614.688357	2924816	5529.836000	29740.0	98.346200	1.057230e+10	0.001891
Aruba	Latin America & Caribbean	13445.593416	104822	872.746000	180.0	582.344444	1.400394e+09	0.008326

Рис. 3.24 – Обрахунок кількості викидів CO2 на душу населення

Далі відсортуємо набір даних за ВВП, викликавши метод `.sort_values()` та вказавши стовпець, за яким іде сортування в якості аргументу `by`


```
In 29: sorted_gpd = df.sort_values(by='gdp', ascending=False)
      sorted_gpd
```

Out 29: 217 rows x 8 columns

country_name	region	gdp_per_capita	population	co2_emission	area	density	gdp	co2_emission_per_capita
United States	North America	57638.159090	323127513	5.254279e+06	9831510.0	32.866519	1.862448e+13	0.016261
China	East Asia & Pacific	8123.180873	1378665000	1.029193e+07	9562911.0	144.167921	1.119915e+13	0.007465
Japan	East Asia & Pacific	38900.569310	126994511	1.214048e+06	377962.0	335.998092	4.940159e+12	0.009560
Germany	Europe & Central Asia	42161.319660	82667685	7.198834e+05	357380.0	231.315924	3.485379e+12	0.008708
United Kingdom	Europe & Central Asia	40367.037840	65637239	4.198202e+05	243610.0	269.435733	2.649581e+12	0.006396
France	Europe & Central Asia	36857.119230	66896109	3.032756e+05	549087.0	121.831529	2.465598e+12	0.004534
India	South Asia	1709.591808	1324171354	2.238377e+06	3287259.0	402.819295	2.263792e+12	0.001690
Italy	Europe & Central Asia	30661.221810	60600590	3.204115e+05	301340.0	201.103703	1.858088e+12	0.005287
Brazil	Latin America & Caribbean	8649.948492	207652865	5.298082e+05	8515770.0	24.384508	1.796187e+12	0.002551
Canada	North America	42183.295100	36286425	5.371935e+05	9984670.0	3.634214	1.530681e+12	0.014804

Рис. 3.25 – Сортуювання набору даних за ВВП

Після цього треба вивести 5 перших рядків, викликавши метод `.head()` із зазначенням кількості рядків (у нашому випадку – 5).

```
In 30: sorted_gpd.head(5)
```

Out 30: 5 rows x 8 columns

country_name	region	gdp_per_capita	population	co2_emission	area	density	gdp	co2_emission_per_capita
United States	North America	57638.159090	323127513	5.254279e+06	9831510.0	32.866519	1.862448e+13	0.016261
China	East Asia & Pacific	8123.180873	1378665000	1.029193e+07	9562911.0	144.167921	1.119915e+13	0.007465
Japan	East Asia & Pacific	38900.569310	126994511	1.214048e+06	377962.0	335.998092	4.940159e+12	0.009560
Germany	Europe & Central Asia	42161.319660	82667685	7.198834e+05	357380.0	231.315924	3.485379e+12	0.008708
United Kingdom	Europe & Central Asia	40367.037840	65637239	4.198202e+05	243610.0	269.435733	2.649581e+12	0.006396

Рис. 3.26 – Виведення ТОП-5 країн за ВВП

Далі виведемо 5 останніх рядків, викликавши метод `.tail()` із зазначенням кількості рядків (у нашому випадку – 5) із поєднанням із методом `.sort_values()` для того, щоб найвище стояв рядок із найменшим значенням ВВП.

```
In 31: sorted_gpd.tail(5).sort_values(by='gdp')
```

Out 31: 5 rows x 8 columns

country_name	region	gdp_per_capita	population	co2_emission	area	density	gdp	co2_emission_per_capita
Tuvalu	East Asia & Pacific	3083.615251	11097	11.001	30.0	369.900000	3.421888e+07	0.000991
Nauru	East Asia & Pacific	7821.298918	13049	47.671	20.0	652.450000	1.020601e+08	0.003653
Kiribati	East Asia & Pacific	1587.057869	114395	62.339	810.0	141.228395	1.815515e+08	0.000545
Marshall Islands	East Asia & Pacific	3665.207477	53066	102.676	180.0	294.811111	1.944979e+08	0.001935
Palau	East Asia & Pacific	14428.140260	21503	260.357	460.0	46.745652	3.102483e+08	0.012108

Рис. 3.27 – Виведення 5 країн із найменшим ВВП

Такі ж операції проведемо й для стовпця викидів CO2 на душу населення.

In 34

```
1 sorted_co2 = df.sort_values(by='co2_emission_per_capita', ascending=False)
2 sorted_co2
```

Out 34

1-10

217 rows x 8 columns

CSV

country_name	region	gdp_per_capita	population	co2_emission	area	density	gdp	co2_emission_per_capita
St. Martin (French part)	Latin America & Caribbean	13445.593416	31949	165114.116337	54.4	587.297794	4.295733e+08	5.168053
San Marino	Europe & Central Asia	47908.561410	33203	165114.116337	60.0	553.383333	1.590708e+09	4.972867
Monaco	Europe & Central Asia	13445.593416	38499	165114.116337	2.0	19249.500000	5.176419e+08	4.288790
Northern Mariana Islands	East Asia & Pacific	22572.378820	55023	165114.116337	460.0	119.615217	1.242000e+09	3.000820
American Samoa	East Asia & Pacific	11834.745230	55599	165114.116337	200.0	277.995000	6.580000e+08	2.969732
Isle of Man	Europe & Central Asia	13445.593416	83737	165114.116337	570.0	146.907018	1.125894e+09	1.971818
Virgin Islands (U.S.)	Latin America & Caribbean	13445.593416	102951	165114.116337	350.0	294.145714	1.384237e+09	1.603813
Guam	East Asia & Pacific	35562.567530	162896	165114.116337	540.0	301.659259	5.793000e+09	1.013617
Channel Islands	Europe & Central Asia	13445.593416	164541	165114.116337	190.0	866.005263	2.212351e+09	1.003483
Kosovo	Europe & Central Asia	3661.429847	1816200	165114.116337	10887.0	166.822816	6.649889e+09	0.090912

In 35

```
1 sorted_co2.head(5)
```

Out 35

5 rows

5 rows x 8 columns

CSV

country_name	region	gdp_per_capita	population	co2_emission	area	density	gdp	co2_emission_per_capita
St. Martin (French part)	Latin America & Caribbean	13445.593416	31949	165114.116337	54.4	587.297794	4.295733e+08	5.168053
San Marino	Europe & Central Asia	47908.561410	33203	165114.116337	60.0	553.383333	1.590708e+09	4.972867
Monaco	Europe & Central Asia	13445.593416	38499	165114.116337	2.0	19249.500000	5.176419e+08	4.288790
Northern Mariana Islands	East Asia & Pacific	22572.378820	55023	165114.116337	460.0	119.615217	1.242000e+09	3.000820
American Samoa	East Asia & Pacific	11834.745230	55599	165114.116337	200.0	277.995000	6.580000e+08	2.969732

In 36

```
1 sorted_co2.tail(5).sort_values(by='co2_emission_per_capita')
```

Out 36

5 rows

5 rows x 8 columns

CSV

country_name	region	gdp_per_capita	population	co2_emission	area	density	gdp	co2_emission_per_capita
Eritrea	Sub-Saharan Africa	13445.593416	34322559	696.730	117600.0	291.858495	4.614872e+11	0.000020
Burundi	Sub-Saharan Africa	285.727442	10524117	440.040	27830.0	378.157276	3.007029e+09	0.000042
Somalia	Sub-Saharan Africa	434.208810	14317996	608.722	637660.0	22.453966	6.217000e+09	0.000043
Chad	Sub-Saharan Africa	664.295652	14452543	729.733	1284000.0	11.255875	9.600761e+09	0.000050
Congo, Dem. Rep.	Sub-Saharan Africa	405.542501	78736153	4671.758	2344860.0	33.578189	3.193080e+10	0.000059

Рис. 3.28 – Виведення перших та останніх п’яти країн за викидами CO2 на душу населення

5 ВИСНОВОК

У ході даної лабораторної роботи я набув навичок в первинній обробці статистичних даних. Було проведено велику кількість операцій над набором даних, що містить інформацію про ВВП на душу населення, кількість населення, кількість викидів CO₂ та площу: виправлено помилки, побудовано діаграми, додано стовпчики щільності населення, ВВП, викидів CO₂ на душу населення, знайдено пропущені значення, замінено їх на середні, знайдено країни з найбільшою ВВП на душу населення (Люксембург), країну з найменшою площею (Монако), регіон із найбільшою середньою площею країн (Північна Америка), країну з найбільшою щільністю населення у світі (Макао), країну з найбільшою щільністю населення у Європі та Центральній Азії (Монако), перевірено, чи співпадає в якомусь із регіонів середнє та медіана ВВП (ні), виведено регіон із найменшою різницею між цими показниками (Субсахарська Африка), а також виведено ТОП-5 та 5 останніх країн за ВВП та кількістю CO₂ на душу населення. Загалом отримані навички значно допоможуть у проведенні складнішого аналізу даних.