

*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 ПІ-13, Бабашев Олексій Дмитрович  
(шифр, прізвище, ім'я, по батькові)

Перевірив Олійник Юрій Олександрович  
(прізвище, ім'я, по батькові)

## 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 завантажено дані з даного csv файлу в dataframe.

```
In 143: 1 import pandas as pd
        2 import matplotlib.pyplot as plt
        3 #TASK1.1
        4 df = pd.read_csv('Data2.csv', sep=';', encoding='cp1252', decimal=',')
        5 df
```

Executed in 99ms, 25 Apr at 23:56:08

Out 143: 217 rows x 6 columns pd.DataFrame

	Country Name	Region	GDP per capita	Population	CO2 emission	Area
0	Afghanistan	South Asia	561.778746	34656032.0	9889.225	652860.0
1	Albania	Europe & Central Asia	4124.982390	2876101.0	5716.853	28750.0
2	Algeria	Middle East & North Africa	3916.881571	40606052.0	145400.217	2381740.0
3	American Samoa	East Asia & Pacific	11834.745230	55599.0	NaN	200.0
4	Andorra	Europe & Central Asia	36988.622030	77281.0	462.042	470.0
5	Angola	Sub-Saharan Africa	3308.700233	28813463.0	34763.160	1246700.0
6	Antigua and Barbuda	Latin America & Caribbean	14462.176280	100963.0	531.715	440.0
7	Argentina	Latin America & Caribbean	12440.320980	43847430.0	204024.546	2780400.0
8	Armenia	Europe & Central Asia	3614.688357	2924816.0	5529.836	29740.0

Досліджено структуру даних.

```
10 #convert_column_to_float(df, 'Area')
11 df.info()
```

Executed in 78ms, 25 Apr at 23:56:08

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

## Виправлення помилок

Змінено назву колонки 'Populatiion' на 'Population'.

```
1 #TASK1.3
2 df = df.rename(columns={'Populatiion': 'Population'})
3 df
```

Executed in 156ms, 27 Apr at 17:12:24

Знайдено рядки, поля яких містять від'ємні елементи.

```
1 df.describe()
```

Executed in 166ms, 27 Apr at 17:12:24

8 rows x 4 columns pd.DataFrame

	GDP per capita	Population	CO2 emission	Area
count	190.000000	2.160000e+02	2.050000e+02	2.170000e+02
mean	13374.833168	3.432256e+07	1.651141e+05	6.126082e+05
std	18091.785849	1.347600e+08	8.335357e+05	1.829940e+06
min	-6722.223536	1.109700e+04	1.100100e+01	-6.765900e+05
25%	1926.540477	7.900265e+05	1.334788e+03	1.045000e+04
50%	5226.289415	6.221590e+06	9.108828e+03	9.222500e+04
75%	16003.299818	2.350337e+07	5.986378e+04	4.474000e+05
max	100738.684200	1.378665e+09	1.029193e+07	1.709825e+07

Виправлено від'ємні значення.

```
In 129 1 #1 variant for every not object value
2 df1 = df
3 df1 = df1.apply(lambda x: x if x.dtype == "object" else x.abs())
4 df1.describe()
Executed in 137ms, 27 Apr at 17:12:24
```

Out 129 ▾ |< < 8 rows ▾ > >| 8 rows × 4 columns [pd.DataFrame](#) ▸

	GDP per capita ▴	Population ▴	CO2 emission ▴	Area ▴
count	190.000000	2.160000e+02	2.050000e+02	2.170000e+02
mean	13445.593416	3.432256e+07	1.651141e+05	6.188441e+05
std	18038.981506	1.347600e+08	8.335357e+05	1.827830e+06
min	285.727442	1.109700e+04	1.100100e+01	2.000000e+00
25%	2031.779671	7.900265e+05	1.334788e+03	1.088700e+04
50%	5235.308547	6.221590e+06	9.108828e+03	9.303000e+04
75%	16003.299818	2.350337e+07	5.986378e+04	4.474200e+05
max	100738.684200	1.378665e+09	1.029193e+07	1.709825e+07

Перевірено на нульові значення.

```
In 133 1 df = df4
2 #TASK1.4
3 df.isna().any()
Executed in 151ms, 27 Apr at 17:12:24
```

Out 133 ▾ |< < 6 rows ▾ > >| Length: 6, dtype: bool [pd.Series](#) ▸

	<unnamed> ▴
Country Name	False
Region	False
GDP per capita	True
Population	True
CO2 emission	True
Area	False

Замінено нульові на середнє арифметичне всіх значень.

```
In 134 1 df = df.fillna(df.mean(numeric_only=True))
2 df.isna().any()
Executed in 132ms, 27 Apr at 17:12:24
```

Out 134 ▾ |< < 6 rows ▾ > >| Length: 6, dtype: bool [pd.Series](#) ▸

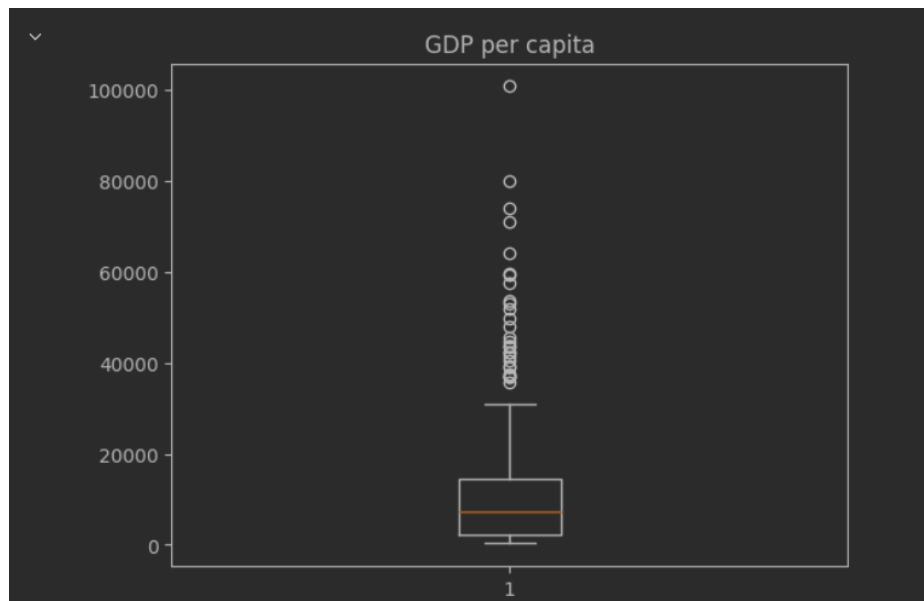
	<unnamed> ▴
Country Name	False
Region	False
GDP per capita	False
Population	False
CO2 emission	False
Area	False

## Діаграми розмаху та гістограми

Виведено діаграми розмаху для кожного стовпця з чисельними даними.

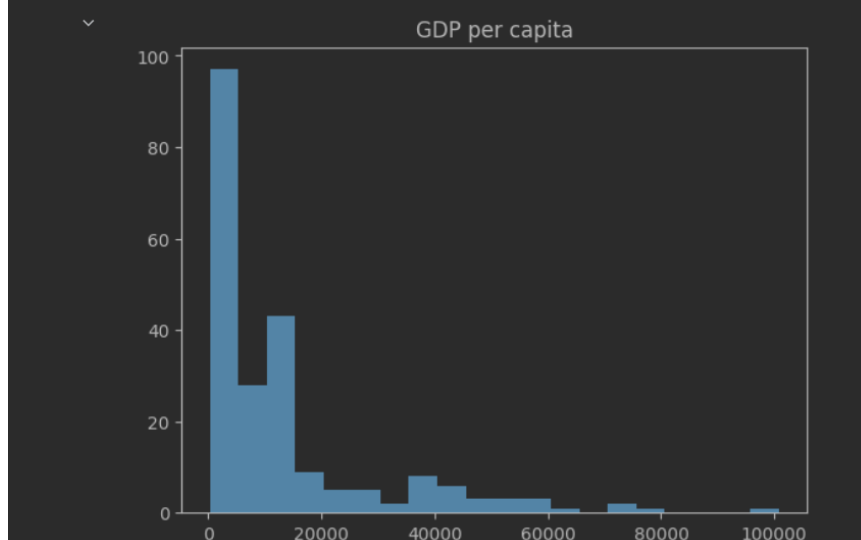
```
In 89 1 #TASK1.4
2 for column in df.columns:
3     if df[column].dtype == float or df[column].dtype == int:
4         plt.figure()
5         plt.title(column)
6         plt.boxplot(df[column])
7     plt.show()
Executed in 666ms, 27 Apr at 17:41:23
```

Маємо подібні діаграми розмаху для кожного стовпця.



Аналогічно з гістограмою.

```
In 90 1 for column in df.columns:
2     if df[column].dtype == float or df[column].dtype == int:
3         plt.figure()
4         plt.title(column)
5         plt.hist(df[column], bins = 20)
6         plt.show()
7
Executed in 751ms, 27 Apr at 17:41:24
```



### Додавання стовпчику із щільністю населення

Додано стовпчик з щільністю населення.

```
In 92 1 #TASK1.5
      2 df['Population density'] = df['Population']/df['Area']
      3 df
      Executed in 43ms, 27 Apr at 17:41:24
```

### Додаткове завдання

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

```
In 93 1 #ADDITIONAL TASKS
      2 #1 is done df = df.fillna(df.mean(numeric_only = True))
      3 #2
      4 max_gdp_row = df.loc[df['GDP per capita'].idxmax()]
      5 max_gdp_country_name = max_gdp_row['Country Name']
      6 print(max_gdp_country_name)
      7 print(f"The biggest gdp per capita is in {max_gdp_row['Country Name']} = {max_gdp_row['GDP per capita']}")
      Executed in 12ms, 27 Apr at 17:41:24

      Luxembourg
      The biggest gdp per capita is in Luxembourg = 100738.6842

In 94 1 min_area_row_df = df.loc[df['Area'].idxmin()]
      2 min_area_country_name = min_area_row_df['Country Name']
      3 print(min_area_country_name)
      4 print(f"The smallest area is in {min_area_row_df['Country Name']} = {min_area_row_df['GDP per capita']}")
      Executed in 22ms, 27 Apr at 17:41:24

      Monaco
      The smallest area is in Monaco = 13445.593416057367
```

Виведено регіон з найбільшою середньою площею країн.

```
In 95 1 #3
      2 #first variant
      3 #work with dataframe
      4 df_grouped = df.groupby(['Region'], as_index=False).agg({'Area': 'mean'})
      5 max_area_row_df = df_grouped.loc[df_grouped['Area'].idxmax()]
      6 name = max_area_row_df['Region']
      7 print(f"The biggest mean of countries area is in {name}")
      Executed in 68ms, 27 Apr at 17:41:24

      The biggest mean of countries area is in North America
```

Виведено назву країни з найбільшою щільністю населення у світі, та у Європі та центральній Азії.

```
In 123 1 #4
      2 #world
      3 max_den_row_df = df.loc[df['Population density'].idxmax()]
      4 max_den_name = max_den_row_df['Country Name']
      5 print(f"The biggest pop density in the world is in {max_den_name}.")
      Executed in 150ms, 27 Apr at 17:58:47

      The biggest pop density in the world is in Macao SAR, China.

In 124 1 #Europe & Central Asia
      2 max_den_row_df = df.loc[df[df['Region'] == 'Europe & Central Asia']['Population density'].idxmax()]
      3 max_den_name = max_den_row_df['Country Name']
      4 print(f"The biggest pop density in Europe & Central Asia is in {max_den_name}.")
      Executed in 152ms, 27 Apr at 17:58:47

      The biggest pop density in Europe & Central Asia is in Monaco.
```

## Аналіз даних в інформаційних системах

Виведено Співпадіння середнього та медіани ВВП по регіонам.

```
In 125 1 #5
      2 #1variant
      3 mean_reg_gdp = df.groupby(['Region'])['GDP per capita'].mean()
      4 median_reg_gdp = df.groupby(['Region'])['GDP per capita'].median()
      5 print(pd.merge(mean_reg_gdp, median_reg_gdp, how='inner'))
      6 |
      Executed in 138ms, 27 Apr at 17:58:47

Empty DataFrame
Columns: [GDP per capita]
Index: []
```

Виведено 5 найбільших назв країн по показниках ВВП та CO2 на душу населення та 5 найменших відповідно.

ВВП

```
In 127 1 #6
      2 sorted_by_gdp = df.sort_values(by='GDP per capita', ascending=False)
      3 sorted_by_gdp.head(5)
      Executed in 197ms, 27 Apr at 17:58:47

Out 127 5 rows x 7 columns pd.DataFrame
      Country Name  Region  GDP per capita  Population  CO2 emission  Area  Population
115 Luxembourg    Europe & Central Asia  100738.68420  582972.0  9658.878  2590.0
188 Switzerland    Europe & Central Asia  79887.51824  8372098.0  35305.876  41290.0
116 Macao SAR, China East Asia & Pacific  74017.18471  612167.0  1283.450  30.3  20
146 Norway          Europe & Central Asia  70868.12250  5232929.0  47626.996  385178.0
92 Ireland          Europe & Central Asia  64175.43924  4773005.0  36066.430  70200.0

In 128 1 sorted_by_gdp.tail(5)
      Executed in 169ms, 27 Apr at 17:58:47

Out 128 5 rows x 7 columns pd.DataFrame
      Country Name  Region  GDP per capita  Population  CO2 emission  Area  Population
118 Madagascar    Sub-Saharan Africa  401.742270  24894551.0  3076.613  587295.0
37 Central African Republic Sub-Saharan Africa  382.213174  4594621.0  300.694  622980.0
134 Mozambique     Sub-Saharan Africa  382.069330  28829476.0  8426.766  799380.0
119 Malawi          Sub-Saharan Africa  300.307665  18091575.0  1276.116  118480.0
71 Burundi          Sub-Saharan Africa  285.727442  10524117.0  440.040  27830.0
```

CO2

```
In 129 1 df['CO2_per_capita'] = df['CO2 emission']/df['Population']
      2 sorted_by_co2 = df.sort_values(by='CO2_per_capita', ascending=False)
      3 sorted_by_co2.head(5)
      Executed in 153ms, 27 Apr at 17:58:47

Out 129 5 rows x 8 columns pd.DataFrame
      Country Name  Region  GDP per capita  Population  CO2 emission  Area  Population de
102 St. Martin (French part) Latin America & Caribbean  13445.593416  51747.0  165114.116337  34.4
163 San Marino      Europe & Central Asia  47908.561410  33203.0  165114.116337  60.0
130 Monaco           Europe & Central Asia  13445.593416  38499.0  165114.116337  2.0
145 Northern Mariana Islands East Asia & Pacific  22572.378820  55023.0  165114.116337  460.0
3 American Samoa    East Asia & Pacific  11834.745230  55599.0  165114.116337  200.0

In 130 1 sorted_by_co2.tail(5)
      Executed in 152ms, 27 Apr at 17:58:47

Out 130 5 rows x 8 columns pd.DataFrame
      Country Name  Region  GDP per capita  Population  CO2 emission  Area  Population de
44 Congo, Dem. Rep. Sub-Saharan Africa  405.942901  7.873619e+07  4671.730  2344000.0
38 Chad            Sub-Saharan Africa  664.295652  1.445254e+07  729.733  1284000.0
175 Somalia         Sub-Saharan Africa  434.208810  1.431800e+07  608.722  637660.0
31 Burundi          Sub-Saharan Africa  285.727442  1.052412e+07  440.040  27830.0
61 Eritrea           Sub-Saharan Africa  13445.593416  3.432256e+07  696.730  117600.0
```

## **Висновок**

У цьому комп'ютерному практикумі був ознайомлений з основними інструментами роботи в python бібліотеках для роботи з даними pandas та matplotlib. Використав на практиці опанований матеріал.