Підготовка даних до аналізу

Ознайомитись з методикою первинної обробки даних. Після завершення цієї лабораторної роботи ви зможете:

- Досліджувати структуру завантажених даних
- Виправляти формати даних
- Знаходити та заповнювати пропуски в даних
- Знаходити викиди та некоректні значення
- Будувати прості візуалізації

Завдання, що оцінюються

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

Завдання #1:

Зчитую дані з файлу у датафрейм

```
import pandas as pd

df = pd.read_csv("../Data2.csv", sep=';' ,encoding='cp1252')
df
```

```
               Country Name                         Region GDP per capita
\
0               Afghanistan                     South Asia    561,7787463

1                   Albania       Europe & Central Asia     4124,98239

2                   Algeria  Middle East & North Africa    3916,881571

3            American Samoa         East Asia & Pacific    11834,74523

4                   Andorra       Europe & Central Asia    36988,62203

..                      ...                          ...            ...
```

```
212  Virgin Islands (U.S.)   Latin America & Caribbean            NaN

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


     Populatiion CO2 emission     Area
0     34656032.0     9809,225   652860
1      2876101.0     5716,853    28750
2     40606052.0   145400,217  2381740
3        55599.0          NaN      200
4        77281.0      462,042      470
..           ...          ...      ...
212     102951.0          NaN      350
213    4551566.0          NaN     6020
214   27584213.0     22698,73   527970
215   16591390.0     4503,076   752610
216   16150362.0    12020,426   390760

[217 rows x 6 columns]
```

Досліджую структуру даних

```
df.dtypes

df['GDP per capita'].count()

df['Populatiion'].count()

df['CO2 emission'].count()

df['Area'].sort_values()

df['GDP per capita'].sort_values()
```

Бачу наступні проблеми в даних:

1. Дані стовпців із числами, що мають дробову частину, записуються з використання коми замість крапки та представлені у вигляді об'єктів
2. Присутні від'ємні дані *(площа, ВВП)* для деяких країн
3. Назва стовпця з кількістю населення містить описку. (Populati **i** on)
4. Є пропущені значення в стовпцях з даними про ВВП, населення та викиди CO2 для багатьох країн

Завдання #2:

Проблема 1

Виправлю описку в назві стовпця

```
df.columns = ['Country Name', 'Region', 'GDP per capita',
'Population', 'CO2 emission', 'Area']
df.columns
```

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

Проблема 2

Використаю регулярні вирази для того, щоб замінити коми на крапки. Також приведу дані до типу float.

```
cols_to_update = ['GDP per capita', 'CO2 emission', 'Area']

df_clean = df.copy()
df_clean[cols_to_update] = df_clean[cols_to_update].apply(lambda x:
pd.to_numeric(x.str.replace(',', '.', regex=True), errors='coerce'))
df_clean
```

```
               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             NaN

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      9809.225   652860.0
1     2876101.0      5716.853    28750.0
2    40606052.0    145400.217  2381740.0
3       55599.0           NaN      200.0
4       77281.0       462.042      470.0
..          ...           ...        ...
212    102951.0           NaN      350.0
213   4551566.0           NaN     6020.0
214  27584213.0     22698.730   527970.0
215  16591390.0      4503.076   752610.0
216  16150362.0     12020.426   390760.0

[217 rows x 6 columns]
```

Проблема 3

Виправлю значення від'ємних даних на додатні, за допомогою команди `abs()`

```
df_clean['Area'] = df_clean['Area'].abs()
df_clean['Area'].sort_values()
```

```
130           2.0
74           10.0
137          20.0
201          30.0
116          30.3
          ...
26      8515770.0
41      9562911.0
206     9831510.0
35      9984670.0
160    17098250.0
Name: Area, Length: 217, dtype: float64
```

```
df_clean['GDP per capita'] = df_clean['GDP per capita'].abs()
df_clean['GDP per capita'].sort_values()
```

```
31     285.727442
119    300.307665
134    382.069330
37     382.213174
118    401.742270
           ...
182           NaN
189           NaN
200           NaN
210           NaN
212           NaN
Name: GDP per capita, Length: 217, dtype: float64
```

Завдання #3:

Через важливість та унікальність даних їхнє прогнозування або відтворення на основі інших країн неможливе. Країни з відсутніми даними видалятимуться зі списку. Будь-які інші можливі способи можуть спотворити наше уявлення про країни та створити неправильну картину

```
df_clean.dropna(inplace=True)
```

Як ми бачимо, кільість стовпців зменшилася

```
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 183 entries, 0 to 216
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Country Name    183 non-null    object 
 1   Region          183 non-null    object 
 2   GDP per capita  183 non-null    float64
 3   Population      183 non-null    float64
 4   CO2 emission    183 non-null    float64
 5   Area            183 non-null    float64
dtypes: float64(4), object(2)
memory usage: 10.0+ KB
```

Завдання #4:

Щільність населення розрахую по формулі:

p/a, де p - кількість населення, a - площа країни

```
df_clean['Population density'] = df_clean['Population'] /
df_clean['Area']
df_clean
```

```
    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
```

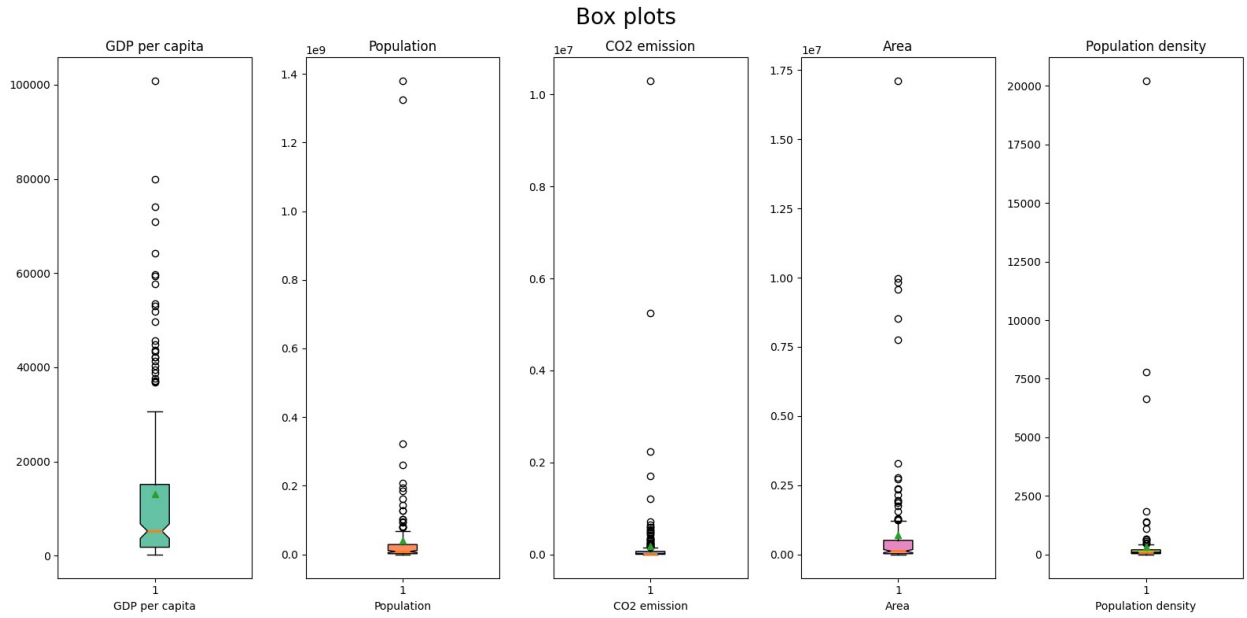
```
4        Andorra       Europe & Central Asia    36988.622030
77281.0
5         Angola          Sub-Saharan Africa     3308.700233
28813463.0
..           ...                         ...             ...          .
..
209      Vanuatu         East Asia & Pacific     2860.566475
270402.0
211      Vietnam         East Asia & Pacific     2170.648054
92701100.0
214  Yemen, Rep.  Middle East & North Africa      990.334774
27584213.0
215       Zambia          Sub-Saharan Africa     1269.573537
16591390.0
216     Zimbabwe          Sub-Saharan Africa     1029.076649
16150362.0

     CO2 emission       Area  Population density
0         9809.225   652860.0           53.083405
1         5716.853    28750.0          100.038296
2       145400.217  2381740.0           17.048902
4          462.042      470.0          164.427660
5        34763.160  1246700.0           23.111786
..             ...        ...                 ...
209        154.014    12190.0           22.182281
211     166910.839   330967.0          280.091671
214      22698.730   527970.0           52.245796
215       4503.076   752610.0           22.045136
216      12020.426   390760.0           41.330643

[183 rows x 7 columns]
```

Завдання #5:

Для побудови графіків скористайтесь бібліотекою Matplotlib. Спробуйте погратись з кольорами, розмірами та підписами.

```
import matplotlib.pyplot as plt
import seaborn as sns

df_cols = ['GDP per capita', 'Population', 'CO2 emission', 'Area',
'Population density']

fig, axs = plt.subplots(1, 5, figsize=(16, 8))
fig.suptitle('Box plots', fontsize=20)

colors = sns.color_palette("Set2", 5)
```

```
for i, col in enumerate(df_cols):
  axs[i].set_title(col)
  axs[i].boxplot(df_clean[col], patch_artist=True, notch=True,
vert=True, showmeans=True,
                 medianprops={"linewidth": 2}, boxprops={"facecolor":
colors[i]})
  axs[i].set_xlabel(col)

plt.tight_layout()
```



Додаткове завдання:

1. Яка країна має найбільший ВВП на людину (GDP per capita)?
2. Яка країна має найменшу площу?
3. Знайдіть країну з найбільшою щільністю населення у світі? У Європі та центральній Азії?
4. Покажіть топ 5 країн та 5 останніх країн по ВВП на людину.

1. Країна з найбільшим ВВП на душу населення

```
df_clean.loc[df_clean['GDP per capita'].idxmax()]
```

```
Country Name                        Luxembourg
Region                 Europe & Central Asia
GDP per capita                    100738.6842
Population                           582972.0
CO2 emission                         9658.878
Area                                   2590.0
```

```
Population density                225.085714
Name: 115, dtype: object
```

1. Країна з найменшою площею

```
df_clean.loc[df_clean['Area'].idxmin()]
```

```
Country Name                            Nauru
Region                 East Asia & Pacific
GDP per capita                 7821.298918
Population                         13049.0
CO2 emission                        47.671
Area                                  20.0
Population density                  652.45
Name: 137, dtype: object
```

1. Країни з найбільшою щільністю населення

Європа та центральна Азія

```
df_europe_and_central_asia = df_clean.loc[(df_clean['Region'] ==
'Europe & Central Asia')]
df_clean.loc[df_europe_and_central_asia['Population
density'].idxmax()]
```

```
Country Name                      Netherlands
Region                 Europe & Central Asia
GDP per capita                  45637.88675
Population                       17018408.0
CO2 emission                     167303.208
Area                                41540.0
Population density                409.687241
Name: 139, dtype: object
```

Світ

```
df_clean.loc[df_clean['Population density'].idxmax()]
```

```
Country Name             Macao SAR, China
Region                 East Asia & Pacific
GDP per capita                 74017.18471
Population                        612167.0
CO2 emission                       1283.45
Area                                  30.3
Population density             20203.531353
Name: 116, dtype: object
```

1. Країни з найбільшими та найменшими показниками ВВП на душу населення

```
df_sorted_by_gdp = df_clean.sort_values(by='GDP per capita',
ascending=False)
df_sorted_by_gdp
```

```
                  Country Name                  Region  GDP per
capita  \
115                 Luxembourg  Europe & Central Asia   100738.684200

188                Switzerland  Europe & Central Asia    79887.518240

116          Macao SAR, China    East Asia & Pacific    74017.184710

146                    Norway  Europe & Central Asia    70868.122500

92                     Ireland  Europe & Central Asia    64175.438240

..                         ...                    ...             ...

118                Madagascar     Sub-Saharan Africa      401.742270

37   Central African Republic     Sub-Saharan Africa      382.213174

134                Mozambique     Sub-Saharan Africa      382.069330

119                    Malawi     Sub-Saharan Africa      300.307665

31                     Burundi     Sub-Saharan Africa      285.727442


     Population  CO2 emission      Area  Population density
115    582972.0      9658.878    2590.0          225.085714
188   8372098.0     35305.876   41290.0          202.763333
116    612167.0      1283.450      30.3        20203.531353
146   5232929.0     47626.996  385178.0           13.585742
92    4773095.0     34066.430   70280.0           67.915410
..          ...           ...       ...                ...
118  24894551.0      3076.613  587295.0           42.388495
37    4594621.0       300.694  622980.0            7.375230
134  28829476.0      8426.766  799380.0           36.064795
119  18091575.0      1276.116  118480.0          152.697291
31   10524117.0       440.040   27830.0          378.157276

[183 rows x 7 columns]
```

Збережіть дані у новий файл 'clean_data2.csv':

```
df_clean.to_csv('../Data2-clean.csv', sep=';', index=False)
```