Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та комп’ютерних технологій

Звіт

Про виконання лабораторної роботи №3

З курсу «Системи опрацювання даних**»**

«Статистичний, візуальний та кореляційний аналіз даних»

Виконав:

Студент групи ФеС-21

Шавало Андрій

Львів-2025

Мета роботи

Ознайомитися з методами статистичного аналізу даних.

Виконати візуалізацію розподілу та взаємозв'язків між змінними. Для dataset побудувати графіки: гісторам та матриці розсіяння, boxplot, violin plot, а також лінійний графік, паралельний графік та (або) інші, які відображають особливості dataset

Використати кореляційний аналіз для виявлення залежностей.

Завдання

1. Завантажити набір даних dataset у форматі CSV (наприклад, data.csv).

2. Завантажити дані у pandas DataFrame та вивести перші 5 рядків. Зрозуміти і пояснити про що дані.

```
df = pd.read_csv("data.csv")
display(df.head())
```

✓ 0.0s Python

	Overall rank	Country or region	Score	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
0	1	Finland	7.632	1.305	1.592	0.874	0.681	0.202	0.393
1	2	Norway	7.594	1.456	1.582	0.861	0.686	0.286	0.340
2	3	Denmark	7.555	1.351	1.590	0.868	0.683	0.284	0.408
3	4	Iceland	7.495	1.343	1.644	0.914	0.677	0.353	0.138
4	5	Switzerland	7.487	1.420	1.549	0.927	0.660	0.256	0.357

- **Overall rank** – загальний рейтинг країни за рівнем щастя.
- **Country or region** – назва країни або регіону.
- **Score** – індекс щастя.
- **GDP per capita** – рівень ВВП на душу населення.
- **Social support** – рівень соціальної підтримки.
- **Healthy life expectancy** – очікувана тривалість здорового життя.
- **Freedom to make life choices** – рівень свободи у виборі життєвого шляху.
- **Generosity** – рівень благодійності.
- **Perceptions of corruption** – рівень сприйняття корупції.

3. Відобразити загальну інформацію про датасет (.info() …)

```
df.info()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 9 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Overall rank                  156 non-null    int64
 1   Country or region             156 non-null    object
 2   Score                         156 non-null    float64
 3   GDP per capita                156 non-null    float64
 4   Social support                156 non-null    float64
 5   Healthy life expectancy       156 non-null    float64
 6   Freedom to make life choices  156 non-null    float64
 7   Generosity                    156 non-null    float64
 8   Perceptions of corruption     156 non-null    float64
dtypes: float64(7), int64(1), object(1)
memory usage: 11.1+ KB
```

4. Перевірити наявність пропущених значень та обробити їх.

```
display(df.isnull().sum())
df.fillna(df.mean(numeric_only=True), inplace=True)
```

✓ 0.0s

```
Overall rank                    0
Country or region               0
Score                           0
GDP per capita                  0
Social support                  0
Healthy life expectancy         0
Freedom to make life choices    0
Generosity                      0
Perceptions of corruption       0
dtype: int64
```

Статистичний аналіз даних

- Використати метод describe() для визначення статистичних характеристик.

```
display(df.describe())
```

✓ 0.0s Python

	Overall rank	Score	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
count	156.000000	156.000000	156.000000	156.000000	156.000000	156.000000	156.000000	156.00000
mean	78.500000	5.375917	0.891449	1.213237	0.597346	0.454506	0.181006	0.11200
std	45.177428	1.119506	0.391921	0.302372	0.247579	0.162424	0.098471	0.09618
min	1.000000	2.905000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	39.750000	4.453750	0.616250	1.066750	0.422250	0.356000	0.109500	0.05100
50%	78.500000	5.378000	0.949500	1.255000	0.644000	0.487000	0.174000	0.08200
75%	117.250000	6.168500	1.197750	1.463000	0.777250	0.578500	0.239000	0.13650
max	156.000000	7.632000	2.096000	1.644000	1.030000	0.724000	0.598000	0.45700

- Обчислити середнє значення, медіану, моду, стандартне відхилення для числових змінних. Визначити мінімальні та максимальні значення.

```
print('Середнє значення:')
display(df.mean(numeric_only=True))
print('Медіана:')
display(df.median(numeric_only=True))
print('Мода:')
display(df.mode(numeric_only=True).loc[0])
print('Стандартне відхилення:')
display(df.std(numeric_only=True))
print('Мінімальне значення:')
display(df.min(numeric_only=True))
print('Максимальне значення:')
display(df.max(numeric_only=True))
```

```
Середнє значення:

Overall rank                    78.500000
Score                            5.375917
GDP per capita                   0.891449
Social support                   1.213237
Healthy life expectancy          0.597346
Freedom to make life choices     0.454506
Generosity                       0.181006
Perceptions of corruption        0.112000
dtype: float64

Медіана:

Overall rank                    78.5000
Score                            5.3780
GDP per capita                   0.9495
Social support                   1.2550
Healthy life expectancy          0.6440
Freedom to make life choices     0.4870
Generosity                       0.1740
Perceptions of corruption        0.0820
dtype: float64

Мода:

Overall rank                    1.000
Score                           5.358
GDP per capita                  0.332
Social support                  0.896
Healthy life expectancy         0.343
Freedom to make life choices    0.312
Generosity                      0.092
Perceptions of corruption       0.082
Name: 0, dtype: float64
```

```
Стандартне відхилення:

Overall rank                    45.177428
Score                            1.119506
GDP per capita                   0.391921
Social support                   0.302372
Healthy life expectancy          0.247579
Freedom to make life choices     0.162424
Generosity                       0.098471
Perceptions of corruption        0.096180
dtype: float64

Мінімальне значення:

Overall rank                    1.000
Score                           2.905
GDP per capita                  0.000
Social support                  0.000
Healthy life expectancy         0.000
Freedom to make life choices    0.000
Generosity                      0.000
Perceptions of corruption       0.000
dtype: float64

Максимальне значення:

Overall rank                    156.000
Score                             7.632
GDP per capita                    2.096
Social support                    1.644
Healthy life expectancy           1.030
Freedom to make life choices      0.724
Generosity                        0.598
Perceptions of corruption         0.457
dtype: float64
```

- Виявити аномальні значення (використати Z-score або IQR).

```
z_scores = np.abs(zscore(numeric_df))
outlier_rows = df[(z_scores > 3).any(axis=1)]
display(outlier_rows)
```

✓ 0.0s Python

	Overall rank	Country or region	Score	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
2	3	Denmark	7.555	1.351	1.590	0.868	0.683	0.284	0.408
19	20	United Arab Emirates	6.774	2.096	0.776	0.670	0.284	0.186	0.112
33	34	Singapore	6.343	1.529	1.451	1.008	0.631	0.261	0.457
95	96	Indonesia	5.093	0.899	1.215	0.522	0.538	0.484	0.018
129	130	Myanmar	4.308	0.682	1.174	0.429	0.580	0.598	0.178
150	151	Rwanda	3.408	0.332	0.896	0.400	0.636	0.200	0.444
154	155	Central African Republic	3.083	0.024	0.000	0.010	0.305	0.218	0.038

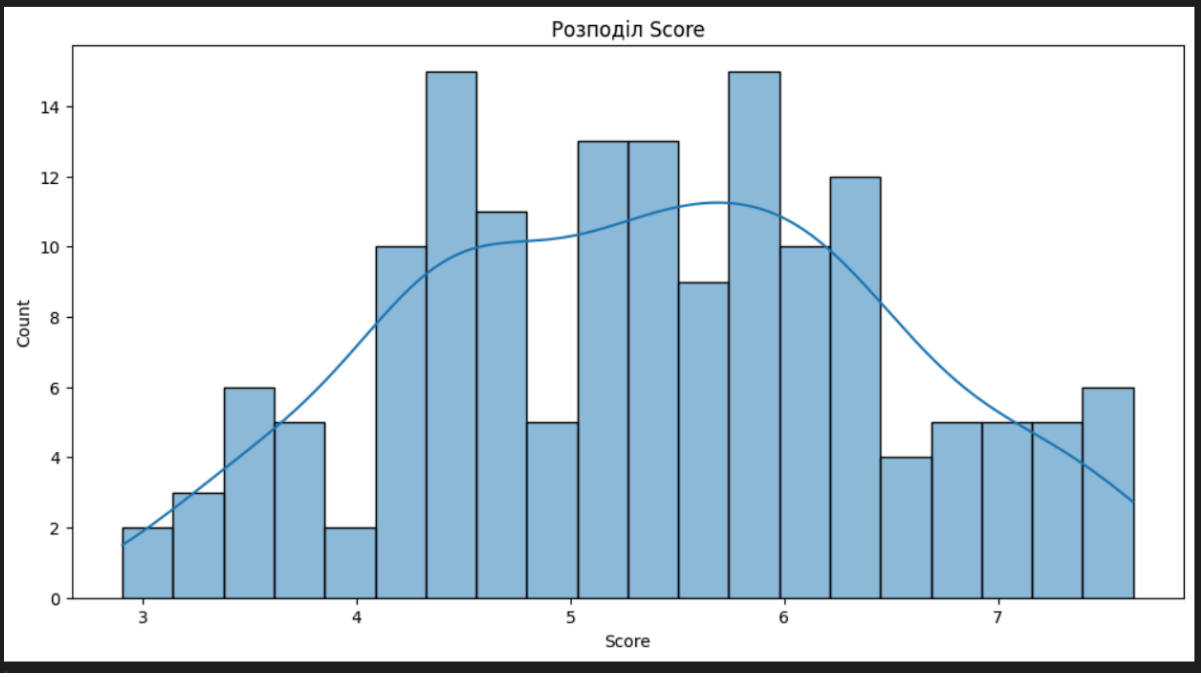
Візуальний аналіз даних

Побудувати графіки за допомогою бібліотек Matplotlib, Pandas Visualization, Seaborn. Пояснити, що зображено на графіках, що відкладено по осях, як називається графік. Див. код вище. Теоретичні відомості.

Для dataset побудувати щонайменше 8 видів графіків:

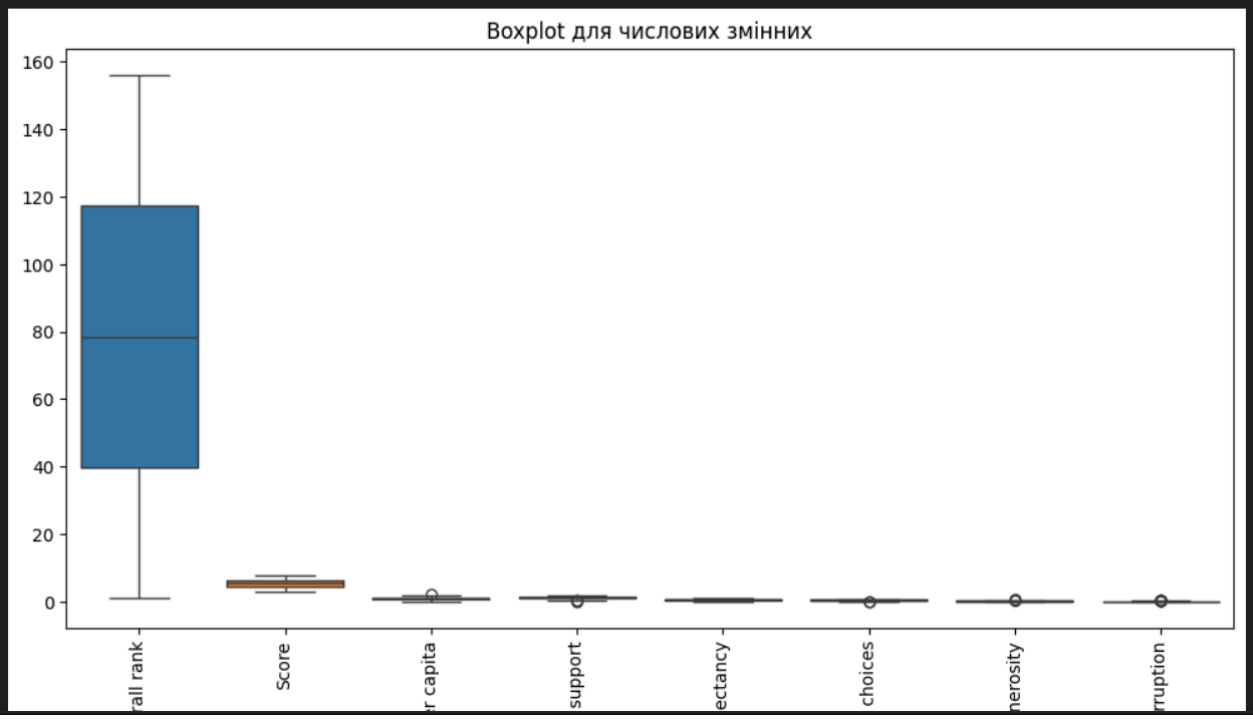
```
plt.figure(figsize=(12, 6))
sns.histplot(df['Score'], kde=True, bins=20)
plt.title("Розподіл Score")
plt.show()
```

✓ 0.1s



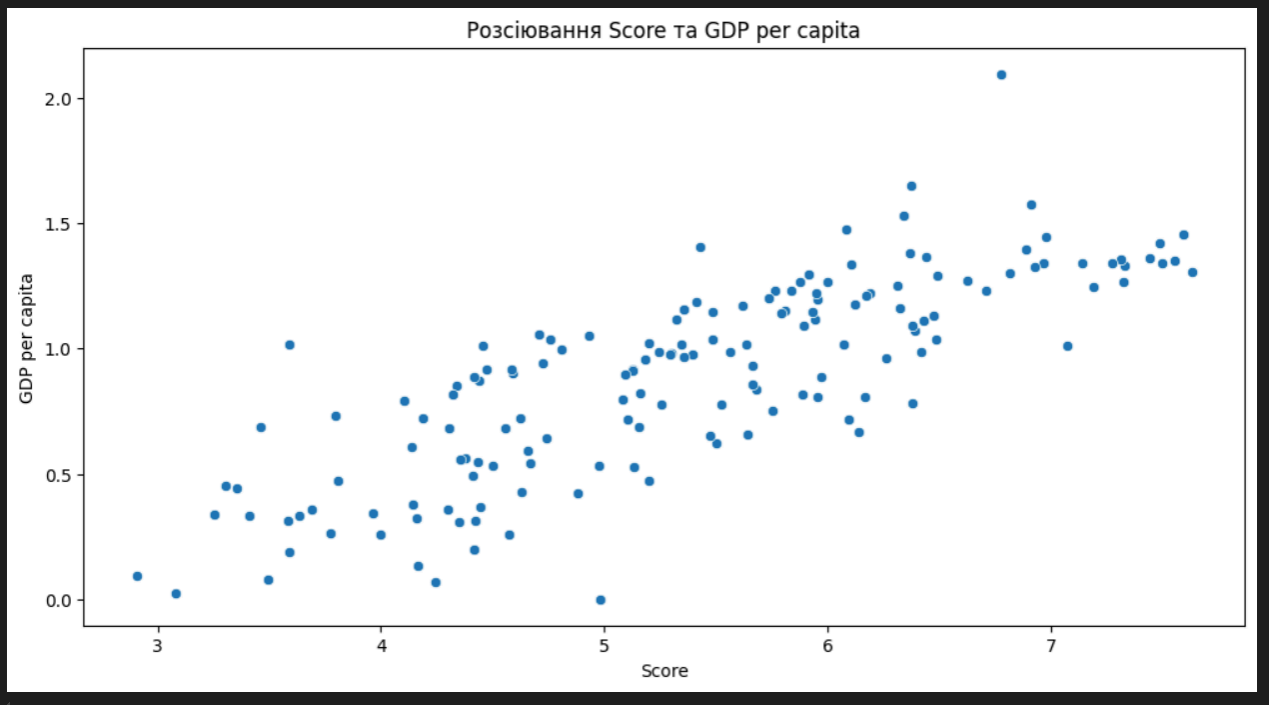

```
plt.figure(figsize=(12, 6))
sns.boxplot(data=df[numeric_cols])
plt.xticks(rotation=90)
plt.title("Boxplot для числових змінних")
plt.show()
```

✓ 0.2s



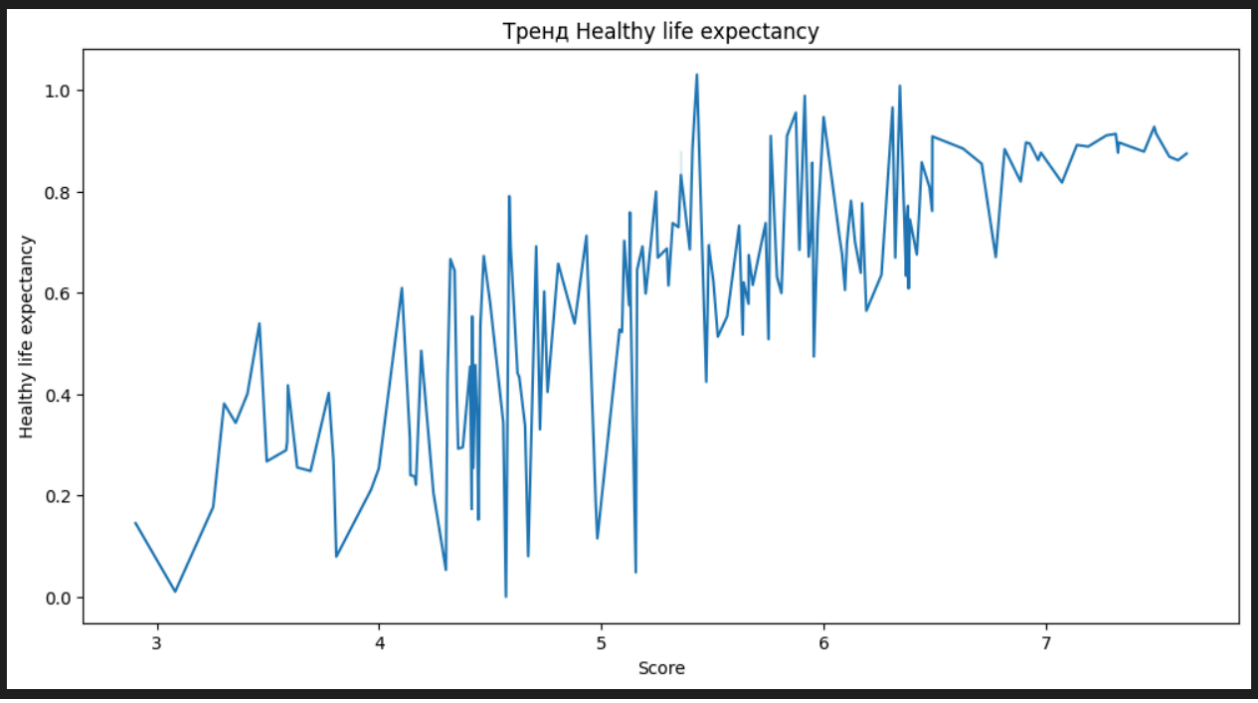
```python
plt.figure(figsize=(12, 6))
sns.scatterplot(x='Score', y='GDP per capita', data=df)
plt.title("Розсіювання Score та GDP per capita")
plt.show()
```

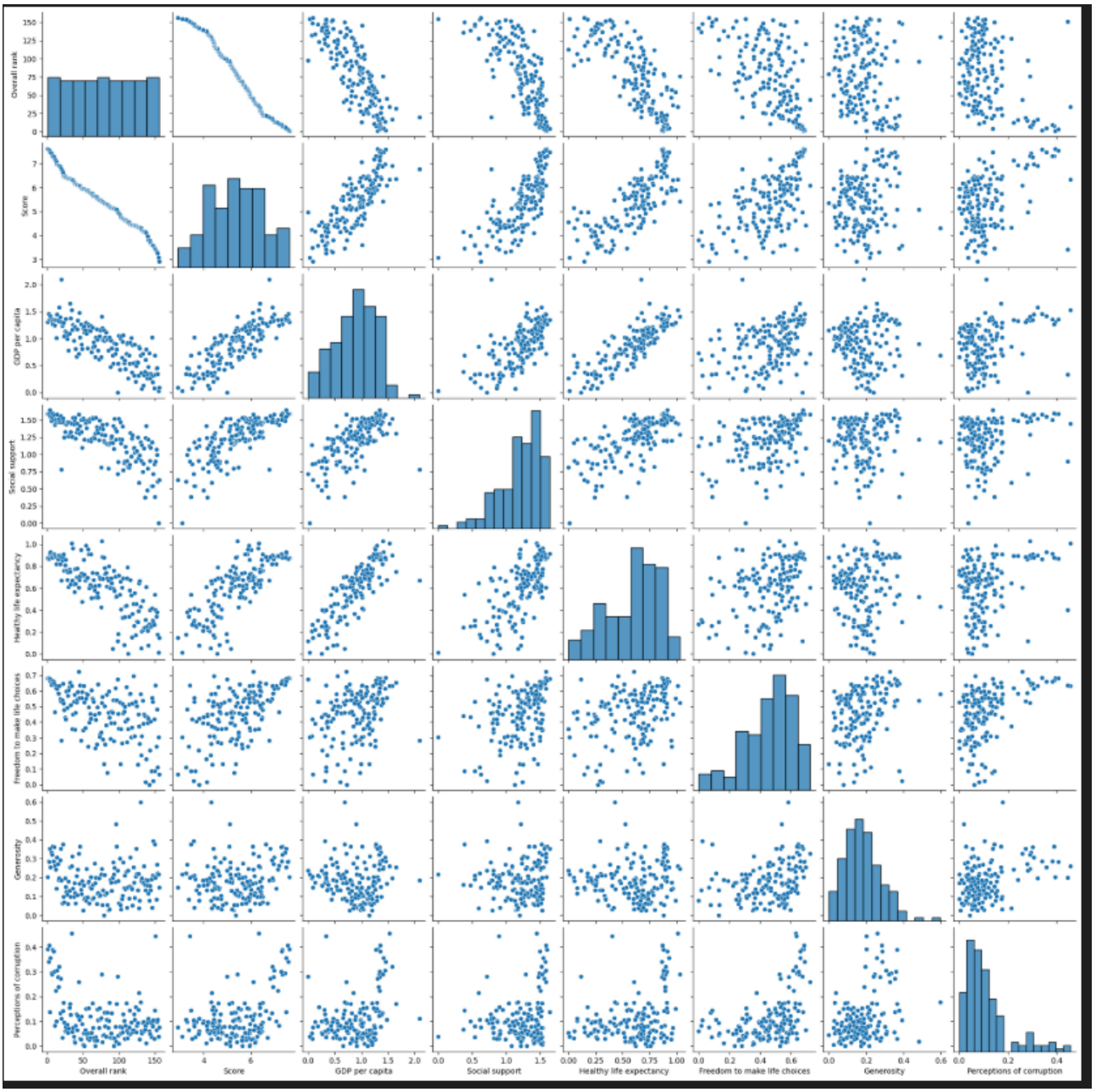
✓ 0.1s




```python
plt.figure(figsize=(12, 6))
sns.lineplot(x='Score', y='Healthy life expectancy', data=df)
plt.title("Тренд Healthy life expectancy")
plt.show()
```

✓ 0.2s



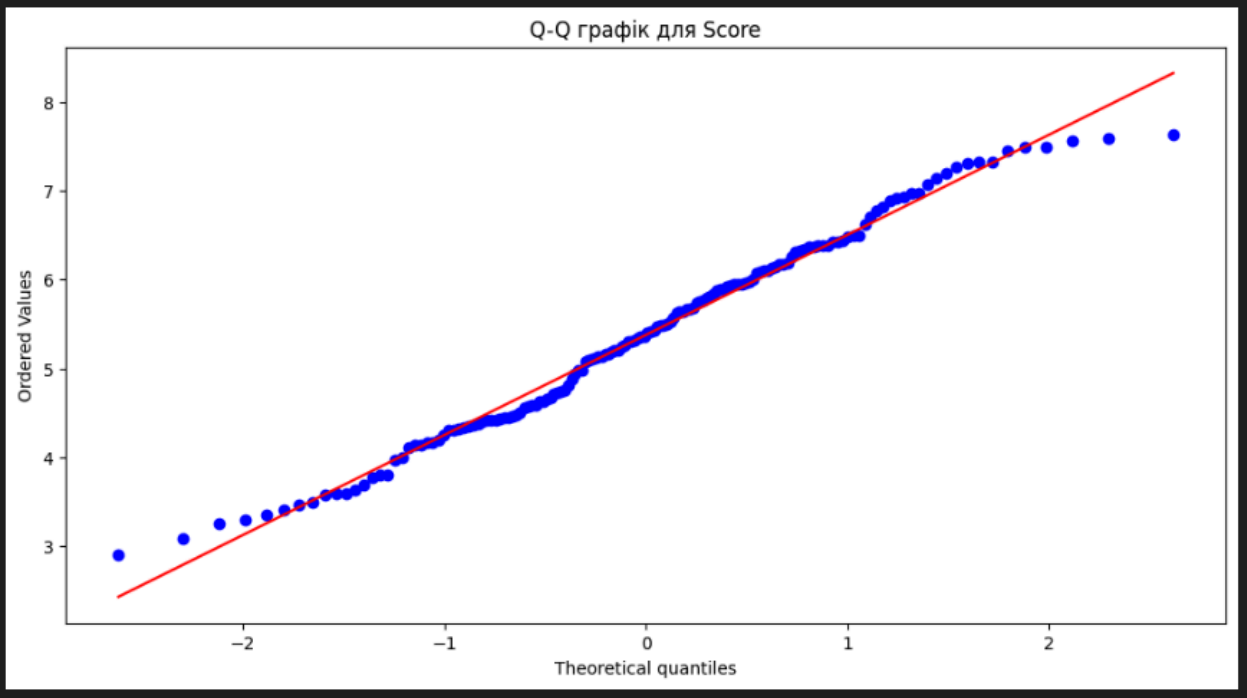

Overall rank
Score
GDP per capita
Social support
Healthy life expectancy
Freedom to make life choices
Generosity
Perceptions of corruption

Перевірка нормальності розподілу даних у вибраних колонках Вибираємо колонку для аналізу Виконуємо Тест Шапіро-Уїлка або (Андерсона-Дарлінга чи Колмогорова-Смірнова). Через порівняння статистичних параметрів, візуалізації та результатів тестів робимо висновок про нормальний розподіл даних в колонці.
Додатково можна використати Q-Q графік змінної з вибраної колонки.

```python
import scipy.stats as stats
from scipy.stats import shapiro

stat, p_value = shapiro(df['Score'])

print(f"Stat: {stat}, p-value: {p_value}")

if p_value > 0.05:
    print("Розподіл ймовірно нормальний")
else:
    print("Розподіл не є нормальним")

plt.figure(figsize=(12, 6))
stats.probplot(df['Score'], dist="norm", plot=plt)
plt.title("Q-Q графік для Score")
plt.show()
```

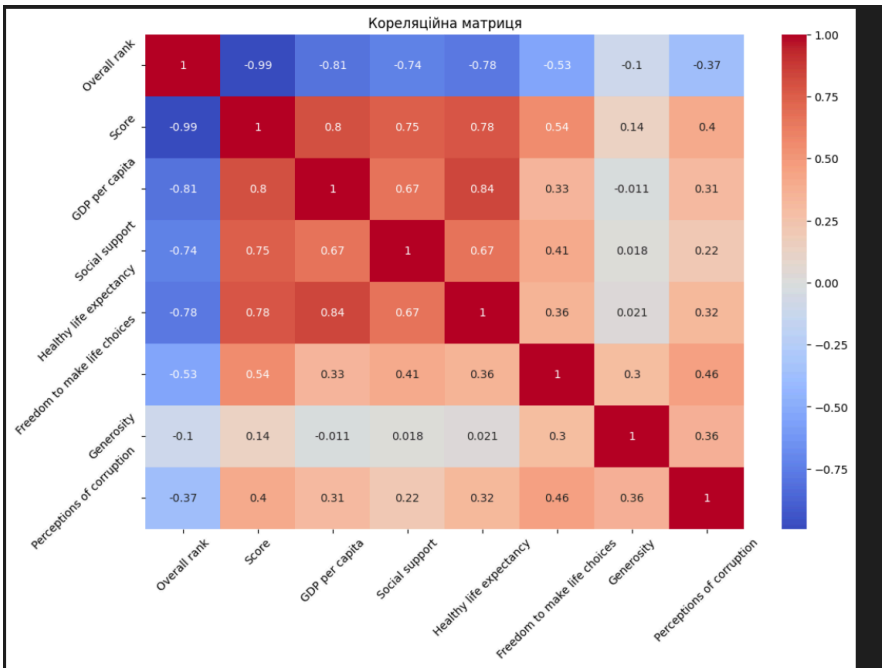
✓ 0.1s

```
Stat: 0.9847009168993887, p-value: 0.08279644536822624
Розподіл ймовірно нормальний
```



Кореляційний аналіз

1. Побудувати кореляційну матрицю (df.corr()).
2. Візуалізувати її за допомогою теплової карти (heatmap).



3. Визначити найсильніші позитивні та негативні кореляції.

```
Найсильніші позитивні кореляції:
Overall rank                  Overall rank                  1.000000
Score                         Score                         1.000000
GDP per capita                GDP per capita                1.000000
Perceptions of corruption     Perceptions of corruption     1.000000
Healthy life expectancy       Healthy life expectancy       1.000000
Freedom to make life choices  Freedom to make life choices  1.000000
Generosity                    Generosity                    1.000000
Social support                Social support                1.000000
GDP per capita                Healthy life expectancy       0.844273
Healthy life expectancy       GDP per capita                0.844273
Score                         GDP per capita                0.802124
GDP per capita                Score                         0.802124
Healthy life expectancy       Score                         0.775814
Score                         Healthy life expectancy       0.775814
                              Social support                0.745760
Social support                Score                         0.745760
dtype: float64
Найсильніші негативні кореляції:
Overall rank            Score                     -0.991749
Score                   Overall rank              -0.991749
GDP per capita          Overall rank              -0.805897
Overall rank            GDP per capita            -0.805897
Healthy life expectancy Overall rank              -0.778700
Overall rank            Healthy life expectancy   -0.778700
                        Social support            -0.737500
Social support          Overall rank              -0.737500
dtype: float64
```

Кореляція та коваріація
df.corr(method='pearson') – кореляція Пірсона
df.corr(method='spearman') – кореляція Спірмена
df.corr(method='kendall') – кореляція Кендалла
df.cov() – коваріація

Висновок

Згідно з результатами проведеного аналізу даних, було виконано кілька важливих кроків для оцінки їх якості та нормальності розподілу. Спершу були перевірені наявні пропущені значення, які були заповнені середнім значенням для числових змінних. Далі було проведено статистичний аналіз, включаючи обчислення основних статистичних параметрів, таких як середнє, медіана, мінімум та максимум, а також виявлення аномальних значень через Z-score. Візуалізація даних включала кілька типів графіків, таких як гістограма, boxplot, теплова карта та графік парних порівнянь. Перевірка нормальності розподілу даних, проведена за допомогою тесту Шапіро-Уїлка, показала, чи можна припустити, що розподіл є нормальним, що має значення для подальшого статистичного аналізу. У результаті можна зробити висновок про нормальність або ненормальність розподілу вибраних змінних, що допомагає при виборі відповідних методів аналізу для подальшої роботи з даними.