КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

ЗВІТ

Лабораторній роботі №1

Студентки групи САТР-3

Кулеш Ріти Вікторівни

Київ 2025

ЗМІСТ

1 Опис вхідної інформації

Обраний датасет відображає якість червоного вина.

Джерело даних:
https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009

Змінні:

1. chlorides - кількість хлоридів
2. pH - рівень кислоти
3. sulphates - кількість сульфатів

Усі змінні є кількісними скалярними і неперервними.

2 Постановка задачі

Мета: провести попередній аналіз обраного набору даних

Для кожної використаної скалярної змінної виконати усе, що має сенс, з переліку:

- дати її класифікацію,
- провести обробку недопустимих та пропущених спостережень, якщо потрібно,
- графічно представити (емпіричну функцію щільності)/(полігон частот),
- побудувати зображення "скринька з вусами" та "стебло-листок",
- виявити та вилучити аномальні спостереження, якщо потрібно,
- підрахувати вибіркові значення: мінімального та максимального спостережень вибірки, медіани, квартилів, децилів,
- підрахувати вибіркові значення усіх характеристик положення центру значень, з якими Ви були ознайомлені і не тільки (тобто також тих характеристик, які на Вашу думку будуть корисні при подальшому аналізі цих даних),
- підрахувати вибіркові значення усіх характеристик розсіювання значень, з якими Ви були ознайомлені і не тільки (тобто також тих характеристик, які на Вашу думку будуть корисні при подальшому аналізі цих даних),
- провести аналіз скошеності та гостроверхості розподілу,
- з'ясувати чи є нормально розподіленими змінні, які Ви досліджуєте,
- у випадку, коли Ваші змінні виявляться не нормально розподіленими, спробувати знайти перетворення, яке дозволяє перейти до дослідження нормально розподілених змінних,
- провести інші процедури попереднього аналізу, які Ви вважаєте доцільними і будуть корисні при подальшому аналізі цих даних,
- провести аналіз отриманих результатів передньої обробки даних та сформулювати відповідні висновки для кожного кроку обробки.

3 Опис додаткового теоретичного матеріалу

Трансформація до нормального розподілу [ред. | ред. код]

Багато методів статистичного висновування вимагають використовування нормально розподілених даних.

Нормальність даних можна досягти через степеневу трансформацію. Для оцінки відповідності даних параметрам нормального розподілу зазвичай використовують графічний метод. Одним із методів перевірки нормальності даних є граф — Гістограма. Нормально розподілені данні матимуть вигляд симетричної Гаусової кривої.

Якщо ж Гістограма виявила несиметричність — дані можна спробувати трансформувати одним із наступних методів:

- Логарифмічна трансформація $Yi = \log(Xi)$
- Взяти корінь із числа $\mathrm{Yi}=\sqrt{(\mathrm{Xi})}$

Якщо стандартне відхилення пропорційно середньому арифметичному, або ж Гістограма показує що данні позитивно асиметричні, може допомогти логарифмічна трансформація. Якщо ж дисперсія пропорційна середньому арифметичному, то підійде коренева трансформація.[1]

4 Покроковий опис обробки даних

1. Підключення бібліотек та завантаження вхідниз даних

   Підключаю всі необхідні бібліотеки:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import statistics
from scipy.stats import skew, kurtosis, kstest, shapiro, boxcox, probplot
```

Завантажую вхідні дані. Датасет має 1599 рядків, що свідчить про достатню кількість даних для подальшої роботи з ним, а також про те, що основні тенденції будуть достатньо точні.

```
[2] all_data = pd.read_csv('winequality-red.csv', encoding='utf-8')

    all_data
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

1599 rows × 12 columns

Відділила від усіх стовбців лише ті, що були опрацьовані під час виконання лабораторної.

```
[3] data = all_data[['chlorides', 'pH', 'sulphates']]
    data
```

	chlorides	pH	sulphates
0	0.076	3.51	0.56
1	0.098	3.20	0.68
2	0.092	3.26	0.65
3	0.075	3.16	0.58
4	0.076	3.51	0.56
...	...	...	...
1594	0.090	3.45	0.58
1595	0.062	3.52	0.76
1596	0.076	3.42	0.75
1597	0.075	3.57	0.71
1598	0.067	3.39	0.66

1599 rows × 3 columns

2. Обробка недопустимих та пропущених значень

```
missing_data = data.isna().sum()
missing_data
```

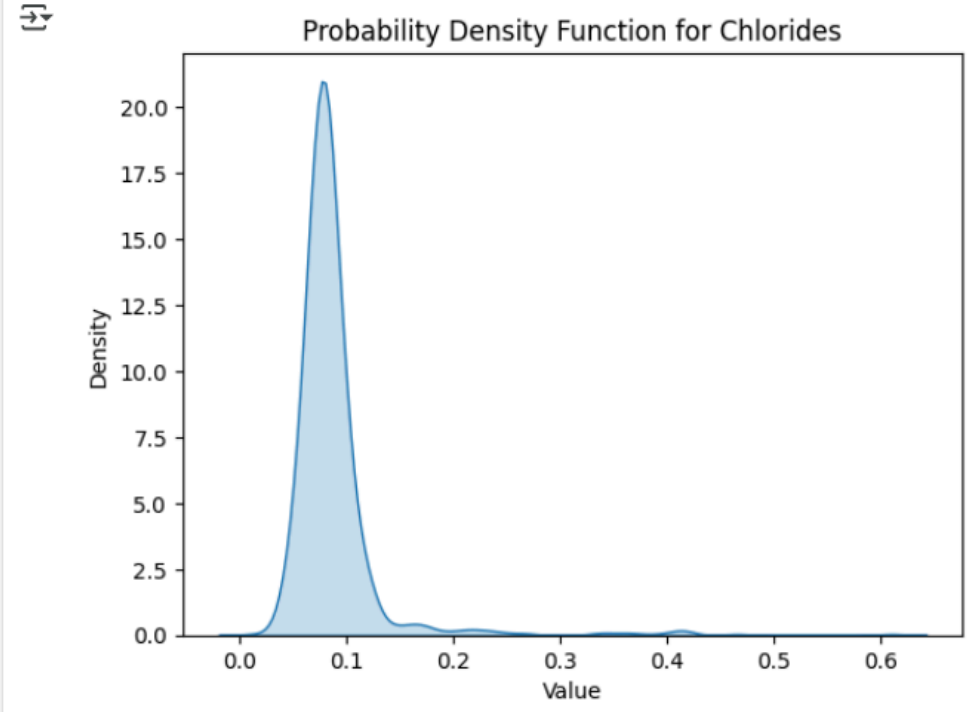
	0
chlorides	0
pH	0
sulphates	0

Перевірка показала, що пропущених недопустимих значень в таблиці нема, що полегшує роботу з даними.

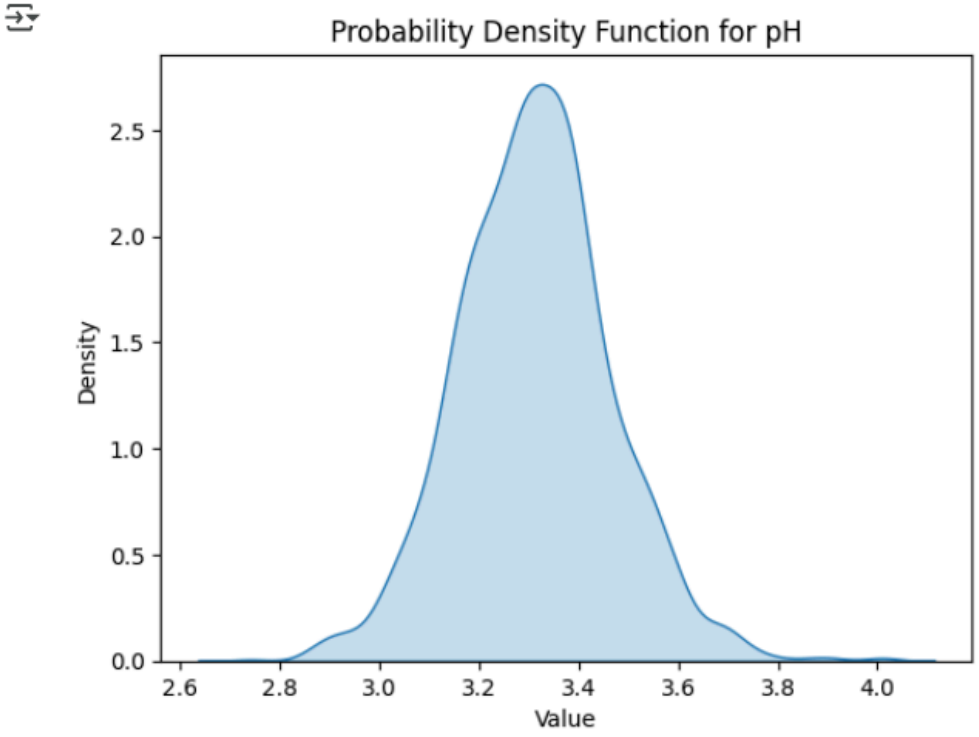
3. Емпірична функція щільності

Графічно зображую функцію щільності для кожної змінної.

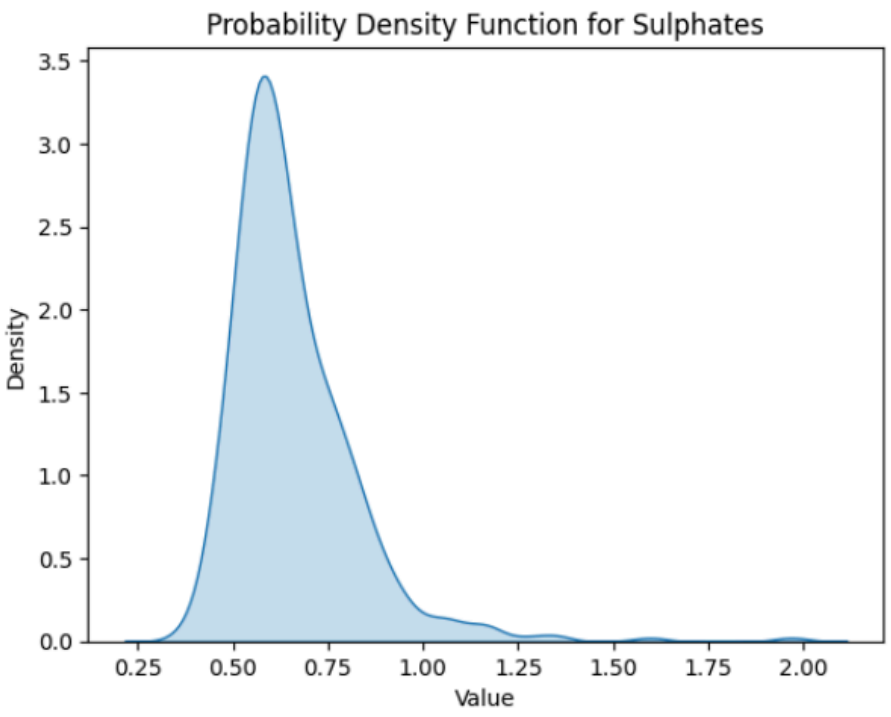
```
sns.kdeplot(data['chlorides'], fill=True)
plt.title('Probability Density Function for Chlorides')
plt.xlabel('Value')
plt.ylabel('Density')
plt.show()
```



```
sns.kdeplot(data['pH'], fill=True)
plt.title('Probability Density Function for pH')
plt.xlabel('Value')
plt.ylabel('Density')
plt.show()
```



```
sns.kdeplot(data['sulphates'], fill=True)
plt.title('Probability Density Function for Sulphates')
plt.xlabel('Value')
plt.ylabel('Density')
plt.show()
```



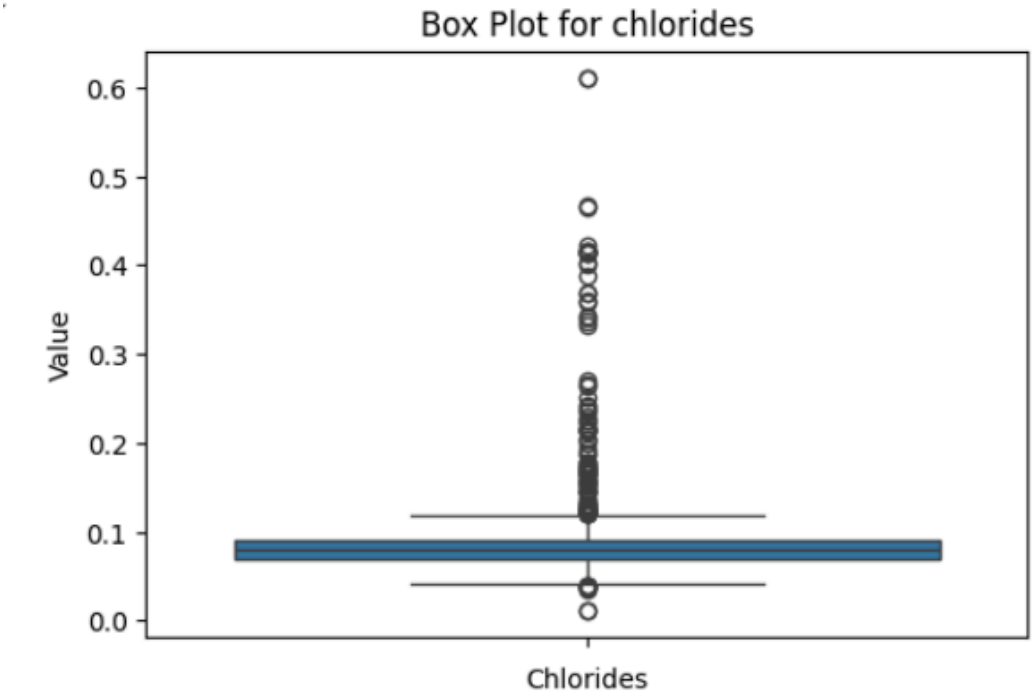
З отриманих графіків можна сказати, що лише змінна pH має наближений до нормального розподіл.

4. Побудова зображень “скринька з вусами” та “стебло-листок”

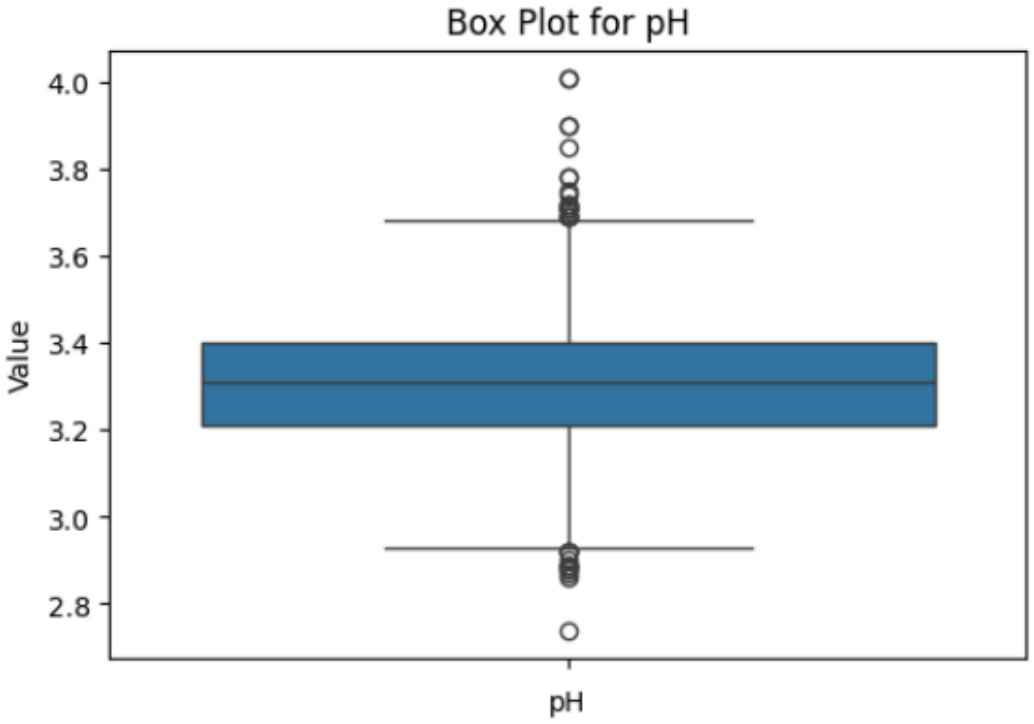
   Будуємо графік "скринька з вусами" з якого одразу можемо зробити висновок про основні характеристики змінних, а саме медіану, 1 та 3 квартилі, наявність викидів.

   Зміст "скриньки з вусами":

   - проекція середньої вертикальної лінії скриньки на вісь абсцис – значення медіани;
   - проекція лівої границі скриньки на вісь абсцис – значення нижнього квартилю, правої границі – верхнього квартилю;
   - проекція лівого кінця лівого вуса – найменше значення у вибірці, проекція правого кінця правого вуса – найбільше значення у вибірці;
   - аномальні спостереження виділяються окремо у формі кружечків.

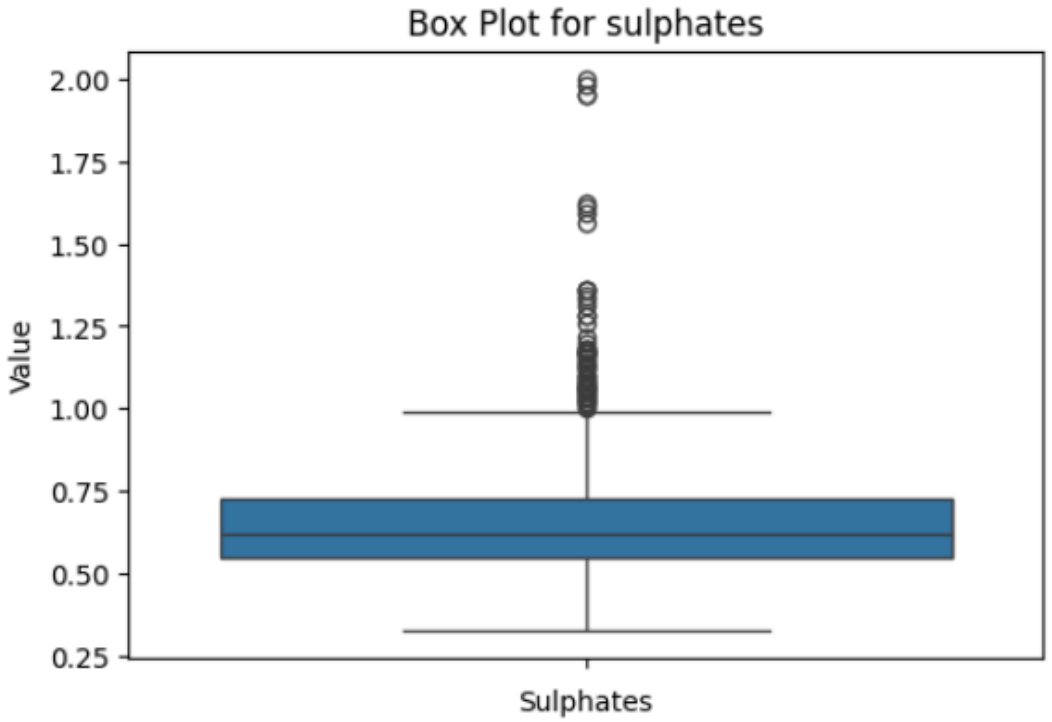


Лінія медіани приблизно ділить прямокутник навпіл, отже є можливість, що розподіл симетричний. Також спостерігаємо наявність аномальних значень.



Лінія медіани ділить прямокутник приблизно порівну, отже можливо розподіл є

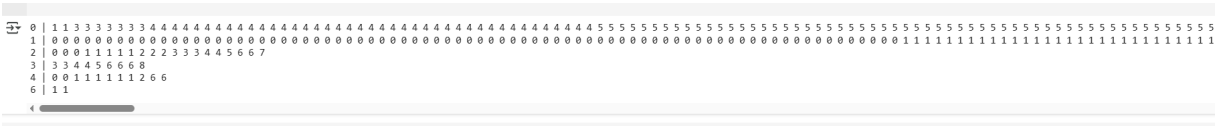
нормальним та симетричним. Також бачимо, що наявні аномальні значення.



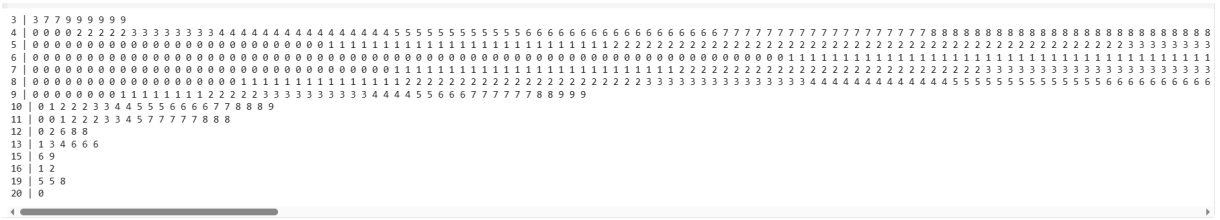
Лінія медіани відхиляється від середини прямокутника, отже розподіл є не симетричним. Також спостерігаємо наявність аномальних значень. Графіки ”стебло-листок” також дають приблизне розуміння розподілу.

```python
def stem_and_leaf(data):
    data = sorted(data.tolist())

    stem_dict = {}

    for num in data:
        stem, leaf = divmod(int(num*1000), 10)
        if stem not in stem_dict:
            stem_dict[stem] = []
        stem_dict[stem].append(leaf)


    for stem in sorted(stem_dict.keys()):
        leaves = " ".join(str(leaf) for leaf in stem_dict[stem])
        print(f"{stem} | {leaves}")

stem_and_leaf(data['chlorides'])
```

pH:



sulphates:



5. Виявити та вилучити аномальні спостереження, якщо потрібно.

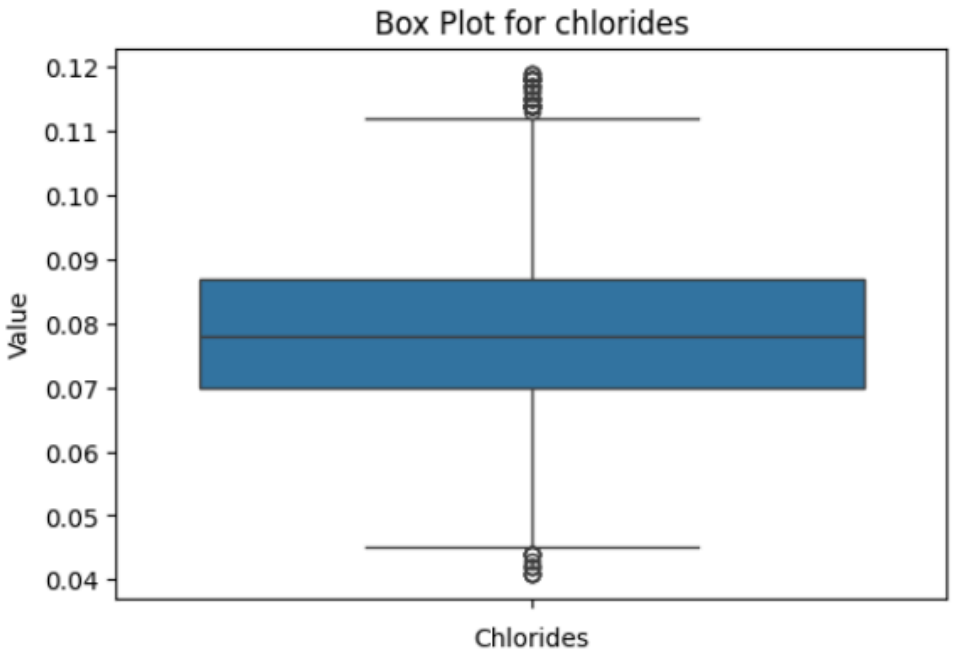
   Для виявлення аномальних змінних використала порівняння із квантилями нормального розподілу.

```
def remove_outliers(x):
    Q1 = x.quantile(0.25)
    Q3 = x.quantile(0.75)

    # Calculate the IQR
    IQR = Q3 - Q1

    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR

    x_fix = x[(x > lower) & (x < upper)]

    return x_fix
```

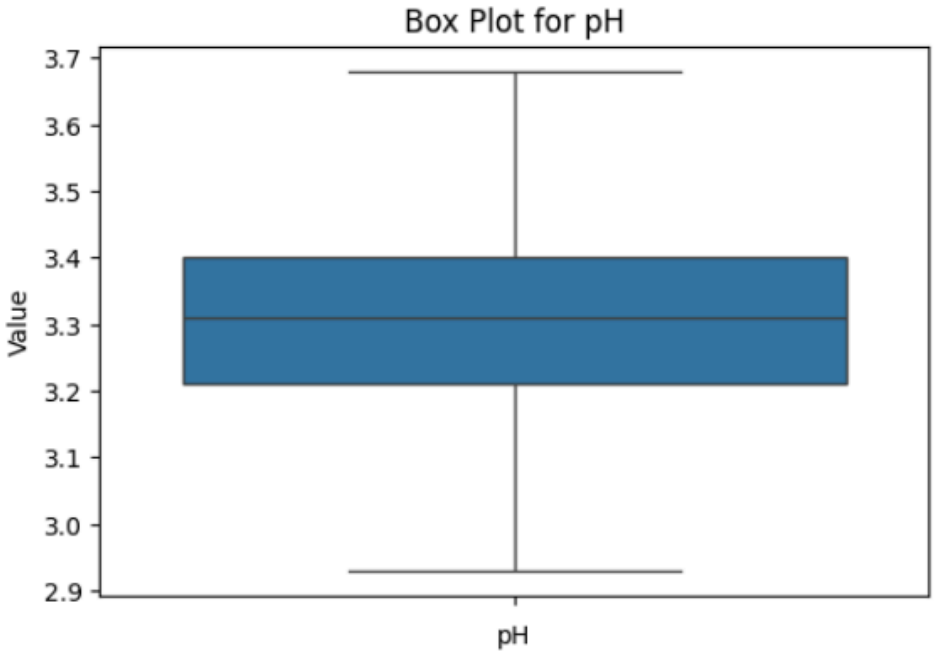
   Зберігаю очищенні дані в нові змінні

```
chlorides = remove_outliers(data['chlorides'])
pH = remove_outliers(data['pH'])
sulphates = remove_outliers(data['sulphates'])
```

Будую ще раз графіки ”скринька з вусами” аби побачити різницю.

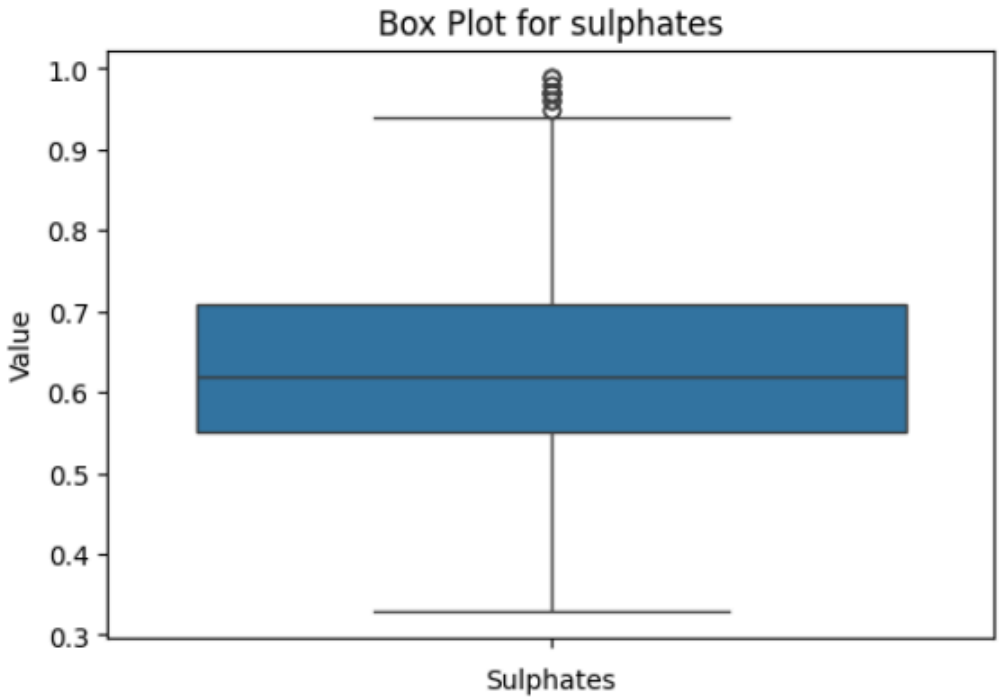


З діаграми бачимо, що кількість аномальних змінних помітно зменшилась, а також медіана ділить прямокутник порівну, що може означати, що розподіл став близьким до нормального.



Бачимо, що розподіл став нормальним за цією діаграмою, але потрібні

додаткові перевірки аби впевнитись в цьому.



Помітно, що кількість аномальних значень зменшилась, але лінія медіана відхиляється від середини прямокутника, отже, розподіл є несиметричним.

6. Обчислення вибіркових значень

```
def general_info(x):
  print("-"*14)
  print("Max: ", max(x))
  print("Min: ", min(x))
  print("Median: ", statistics.median(x))

  print("-"*14)

  for i in range(1, 4):
    q = 0.25 * i
    print("Quantile ", q, " : ", x.quantile(q))

  print("-"*14)

  for i in range(1, 10):
    d = 0.1 * i
    print("Decile ", round(d, 1), " : ", x.quantile(d))

  print("-"*14)

print("Chlorides: ")
general_info(chlorides)
print("pH: ")
general_info(pH)
print("Sulphates: ")
general_info(sulphates)
```

```
Chlorides:
--------------
Max:  0.119
Min:  0.041
Median:  0.078
--------------
Quantile  0.25  :  0.07
Quantile  0.5  :  0.078
Quantile  0.75  :  0.087
--------------
Decile  0.1  :  0.06
Decile  0.2  :  0.067
Decile  0.3  :  0.071
Decile  0.4  :  0.075
Decile  0.5  :  0.078
Decile  0.6  :  0.081
Decile  0.7  :  0.085
Decile  0.8  :  0.09
Decile  0.9  :  0.098
--------------
```

```
pH:
--------------
Max:  3.68
Min:  2.93
Median:  3.31
--------------
Quantile  0.25  :  3.21
Quantile  0.5  :  3.31
Quantile  0.75  :  3.4
--------------
Decile  0.1  :  3.13
Decile  0.2  :  3.19
Decile  0.3  :  3.23
Decile  0.4  :  3.28
Decile  0.5  :  3.31
Decile  0.6  :  3.35
Decile  0.7  :  3.38
Decile  0.8  :  3.42
Decile  0.9  :  3.5
--------------

Sulphates:
--------------
Max:  0.99
Min:  0.33
Median:  0.62
--------------
Quantile  0.25  :  0.55
Quantile  0.5  :  0.62
Quantile  0.75  :  0.71
--------------
Decile  0.1  :  0.5
Decile  0.2  :  0.54
Decile  0.3  :  0.56
Decile  0.4  :  0.59
Decile  0.5  :  0.62
Decile  0.6  :  0.65
Decile  0.7  :  0.69
Decile  0.8  :  0.74
Decile  0.9  :  0.82
--------------
```

7. Обчислення вибіркових значень положень центра

```
def central_char(x):
  print("-"*14)
  print("Expected value: ", statistics.mean(x))
  print("Geometric mean: ", statistics.geometric_mean(x))
  print("Harmonic mean: ", statistics.harmonic_mean(x))
  print("Moda: ", statistics.multimode(x))
  print("-"*14)

print("Chlorides: ")
central_char(chlorides)
print("pH: ")
central_char(pH)
print("Sulphates: ")
central_char(sulphates)
```

```
Chlorides:
--------------
Expected value:  0.07875588433086751
Geometric mean:  0.0773199242756907
Harmonic mean:  0.07582915787835671
Moda:  [0.08]
--------------
pH:
--------------
Expected value:  3.308772378516624
Geometric mean:  3.305787827126788
Harmonic mean:  3.3027990109549887
Moda:  [3.3]
--------------
Sulphates:
--------------
Expected value:  0.6364155844155844
Geometric mean:  0.6253477973353126
Harmonic mean:  0.6146673873274582
Moda:  [0.6]
--------------
```

8. Обчислення вибіркових значень розсіювання

```
def dispersal_char(x):
  print("-"*14)
  print("Variance: ", statistics.variance(x))
  print("Standard Deviation: ", statistics.stdev(x))
  print("Coefficient of Variance: ", (statistics.stdev(x)/statistics.mean(x) * 100), "%")
  print("Probabilistic Deviation: ", 0.5*(x.quantile(0.75) - x.quantile(0.25)))
  print("Range of Sample: ", max(x) - min(x))
  print("Concentration Interval of Distribution: ", (statistics.mean(x) - 3 * statistics.stdev(x)),
        "-", (statistics.mean(x) + 3 * statistics.stdev(x)))
  print("-"*14)

print("Chlorides: ")
dispersal_char(chlorides)
print("pH: ")
dispersal_char(pH)
print("Sulphates: ")
dispersal_char(sulphates)
```

```
Chlorides:
--------------
Variance:  0.00022097334186548103
Standard Deviation:  0.014865172110186987
Coefficient of Variance:  18.874998657542527 %
Probabilistic Deviation:  0.008499999999999994
Range of Sample:  0.0780000000000001
Concentration Interval of Distribution:  0.03416036800030655 - 0.12335140066142847
--------------
pH:
--------------
Variance:  0.019739758775912934
Standard Deviation:  0.14049825186070086
Coefficient of Variance:  4.246235031848534 %
Probabilistic Deviation:  0.09499999999999997
Range of Sample:  0.75
Concentration Interval of Distribution:  2.8872776229345214 - 3.730267134098727
--------------
Sulphates:
--------------
Variance:  0.014632107879125422
Standard Deviation:  0.12096325011806447
Coefficient of Variance:  19.006959144337124 %
Probabilistic Deviation:  0.07999999999999996
Range of Sample:  0.6599999999999999
Concentration Interval of Distribution:  0.273525834061391 - 0.9993053347697778
--------------
```

9. Аналіз скошеності та гостроверхості розподілу

Щоб проаналізувати скошеність знаходжу коефіцієнти асиметрії.

```
print("Skewness for Chlorides: ", skew(data["chlorides"]))
print("Skewness for pH: ", skew(data["pH"]))
print("Skewness for Sulphates: ", skew(data["sulphates"]))
```

```
Skewness for Chlorides:  5.675016527504258
Skewness for pH:  0.19350175891005525
Skewness for Sulphates:  2.426393455449087
```

Для всіх змінних коефіцієнт є більшим 0, що свідчить про те, що розподіли скошенні ліворуч. Також помітно, що коефіцієнт змінної pH близька до нуля, розподіл наближений до нормального.

Щоб проаналізувати гостроверхість знаходжу для змінних коефіцієнт ексцесу.

```
print("Kurtosis for Chlorides: ", skew(chlorides))
print("Kurtosis for pH: ", skew(pH))
print("Kurtosis for Sulphates: ", skew(sulphates))
```

```
Kurtosis for Chlorides:  0.19149117942695665
Kurtosis for pH:  0.030755054823969594
Kurtosis for Sulphates:  0.5858272358069182
```

Для всіх змінних коефіцієнт більший 0, отже розподіли є більш гостроверхими ніж нормальний із відповідними параметрами.

10. Перевірка на нормальну розподіленість

Використала два тести: Колмогорова-Смірнова та Шапіро-Вілка

Колмогорова-Смірнова:

```
stat1, p_val1 = kstest(chlorides, "norm")
stat2, p_val2 = kstest(pH, "norm")
stat3, p_val3 = kstest(sulphates, "norm")

print("For chlorides: ")
print("Statistic: ", stat1, "\np-value: ", p_val1)
print("-"*14)
print("For pH: ")
print("Statistic: ", stat2, "\np-value: ", round(p_val2, 5))
print("-"*14)
print("For sulphates: ")
print("Statistic: ", stat3, "\np-value: ", p_val3)
```

```
For chlorides:
Statistic:  0.5163520520682413
p-value:  0.0
--------------
For pH:
Statistic:  0.9983051899807227
p-value:  0.0
--------------
For sulphates:
Statistic:  0.6547138677162306
p-value:  0.0
```

Значення $p<0.05$, отже, відкидаю гіпотезу про нормальність розподілу.

Шапіро-Вілка:

```
s1, p1 = shapiro(chlorides)
s2, p2 = shapiro(pH)
s3, p3 = shapiro(sulphates)

print("For chlorides: ")
print("Statistic: ", s1, "\np-value: ", p1)
print("-"*14)
print("For pH: ")
print("Statistic: ", s2, "\np-value: ", p2)
print("-"*14)
print("For sulphates: ")
print("Statistic: ", s3, "\np-value: ", p3)
```

```
For chlorides:
Statistic:  0.9912129630795583
p-value:  9.07582990238058e-08
--------------
For pH:
Statistic:  0.9970899577278729
p-value:  0.0052645327698861875
--------------
For sulphates:
Statistic:  0.9691276388368203
p-value:  1.2259063185779397e-17
```

Для цього тесту усі значення p_value < 0.05.

Отже, двома тестами відкинули гіпотезу про те, що розподіли є нормальними.

11. Переведення розподілів до нормально розподілених

Спершу я спробувала використати логарифмічне перетворення, але це не дало очікуваного результату, це помітно з отриманих значень тесту Шапіро-Вілка

```
[39] chl_norm1 = np.log(chlorides)
     pH_norm1 = np.log(pH)
     sul_norm1 = np.log(sulphates)
```

```
s1, p1 = shapiro(chl_norm1)
s2, p2 = shapiro(pH_norm1)
s3, p3 = shapiro(sul_norm1)

print("For chlorides: ")
print("Statistic: ", s1, "\np-value: ", p1)
print("-"*14)
print("For pH: ")
print("Statistic: ", s2, "\np-value: ", p2)
print("-"*14)
print("For sulphates: ")
print("Statistic: ", s3, "\np-value: ", p3)
```

```
For chlorides:
Statistic:  0.9832911008942912
p-value:  4.127570529818875e-12
--------------
For pH:
Statistic:  0.996806873177082
p-value:  0.0026196678112443807
--------------
For sulphates:
Statistic:  0.9913527921872384
p-value:  7.035006273913902e-08
```

Також я спробувала використати перетворення квадратним коренем, але це також не дало очікуваного результату.

```
[43] chl_norm2 = np.sqrt(chlorides)
     pH_norm2 = np.sqrt(pH)
     sul_norm2 = np.sqrt(sulphates)
```

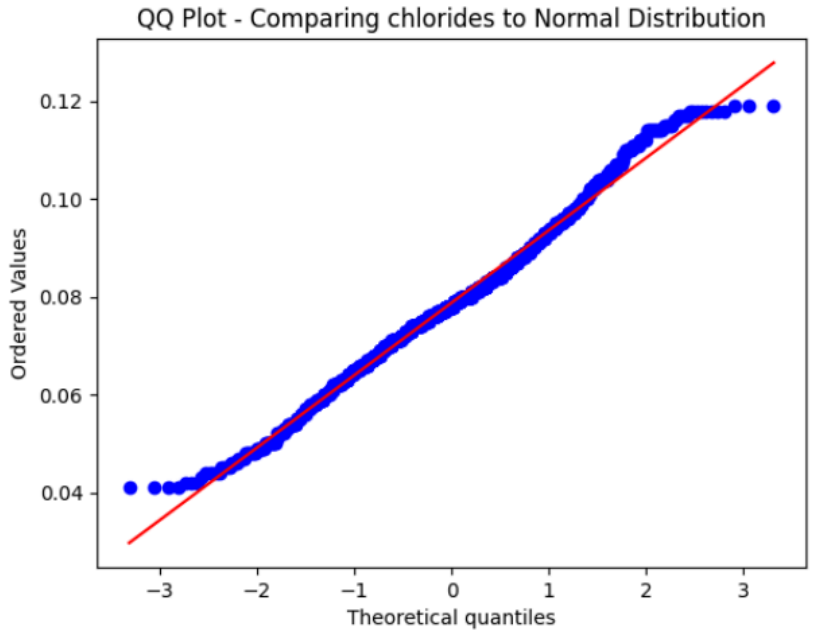
```
[44] s1, p1 = shapiro(chl_norm2)
     s2, p2 = shapiro(pH_norm2)
     s3, p3 = shapiro(sul_norm2)

     print("For chlorides: ")
     print("Statistic: ", s1, "\np-value: ", p1)
     print("-"*14)
     print("For pH: ")
     print("Statistic: ", s2, "\np-value: ", p2)
     print("-"*14)
     print("For sulphates: ")
     print("Statistic: ", s3, "\np-value: ", p3)
```

```
For chlorides:
Statistic:  0.9924890498223529
p-value:  7.085288607185494e-07
--------------
For pH:
Statistic:  0.9971421682260191
p-value:  0.005997019782284366
--------------
For sulphates:
Statistic:  0.9831922445114037
p-value:  1.981661613133929e-12
```

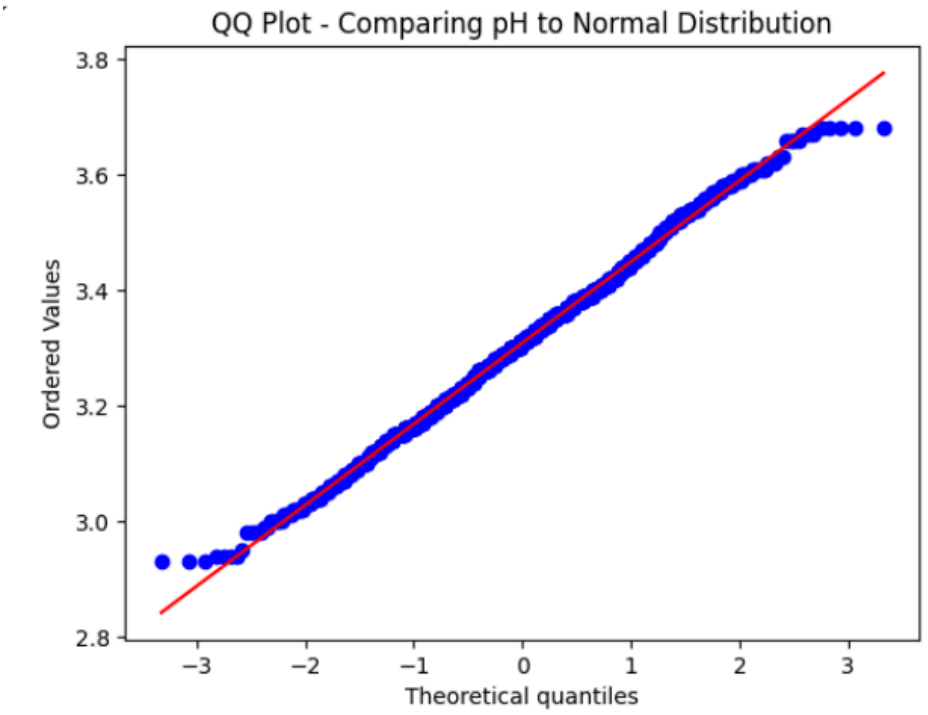
Також були спроби використати метод схожий до наявного в мові R scale(), але це також не дало позитивного результату.

12. В результаті вирішила побудувати QQ діаграми аби побачити візуально наскільки суттєвими є відхилення від нормального розподілу.

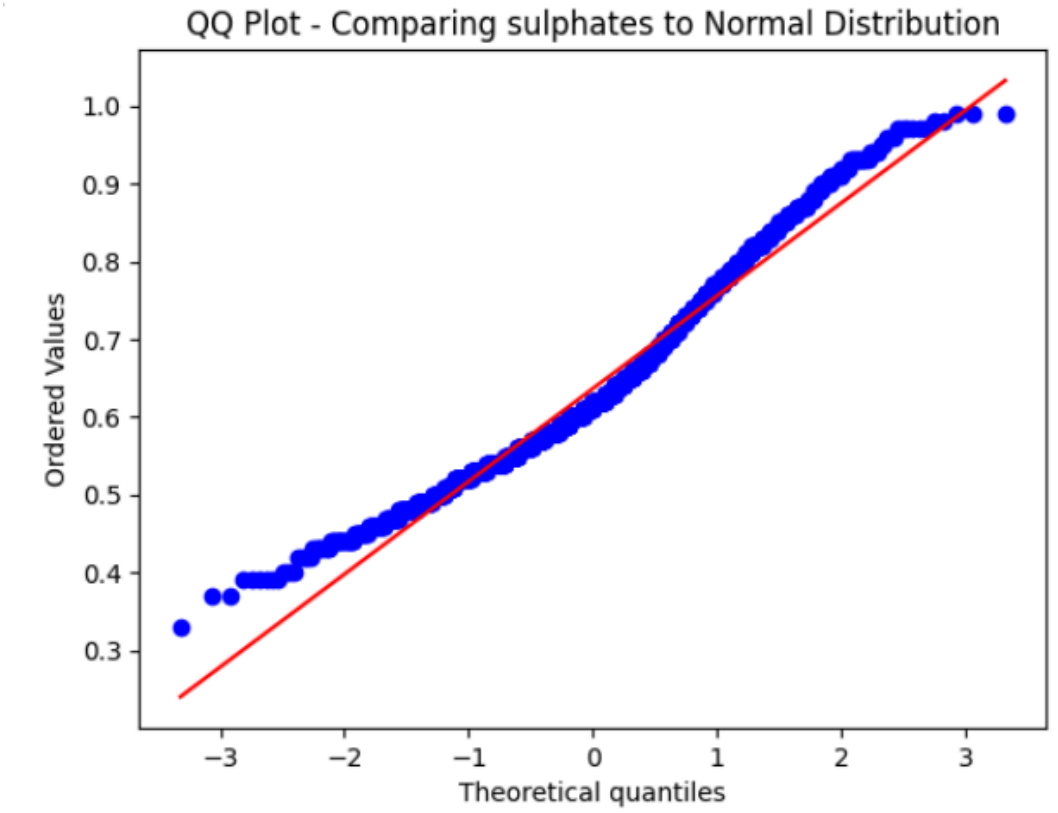
```
probplot(chlorides.to_numpy(), dist="norm", plot=plt)
plt.title('QQ Plot - Comparing chlorides to Normal Distribution')
plt.show()
```



```
] probplot(pH.to_numpy(), dist="norm", plot=plt)
  plt.title('QQ Plot - Comparing pH to Normal Distribution')
  plt.show()
```




```
probplot(sulphates.to_numpy(), dist="norm", plot=plt)
plt.title('QQ Plot - Comparing sulphates to Normal Distribution')
plt.show()
```



З отриманих діаграм бачимо, що всі розподіли далекі від прямої, а тому

є малі шанси отримати нормальні розподіли використовуючи перетворення.

5 Висновки

В даній лабораторній роботі був проведений попередній аналіз обраних 3 змінних з датасету, який відображає інформацію про якість вина.

Для кожної змінної була проведена обробка недопустимих та пропущених спостережень, виявлені та вилучені аномалії, підраховані основні вибіркові значення, вибіркові значення усіх характеристик положення центру значень та вибіркові значення усіх характеристик розсіювання значень. Проведений аналіз скошеності та гостроверхості розподілу та були підібрані перетворення, які дають змогу перейти до нормально розподілу, але всі вони не дали очікуваного результату.

Побудовані графіки функції щільності, "скринька з вусами" та "стебло-листок" одразу дають нам базове розуміння розподілу та його зміни після перетворень.

Також були побудовані Q-Q діаграми для порівняння розподілу з нормальним – змінні не відповідають нормальному розподілу.

6 Додаток. Програмна реалізація

```
[1] import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    import numpy as np
    import statistics
    from scipy.stats import skew, kurtosis, kstest, shapiro, boxcox, probplot
```

```
all_data = pd.read_csv('winequality-red.csv', encoding='utf-8')

all_data
```

```
data = all_data[['chlorides', 'pH', 'sulphates']]
data
```

```
[4] data.loc[:, 'chlorides'] = data['chlorides'].astype('Float64')
    data.loc[:, 'pH'] = data['pH'].astype('Float64')
    data.loc[:, 'sulphates'] = data['sulphates'].astype('Float64')
```

```
[5] missing_data = data.isna().sum()
    missing_data
```

```
sns.kdeplot(data['chlorides'], fill=True)
plt.title('Probability Density Function for Chlorides')
plt.xlabel('Value')
plt.ylabel('Density')
plt.show()
```

```
sns.kdeplot(data['pH'], fill=True)
plt.title('Probability Density Function for pH')
plt.xlabel('Value')
plt.ylabel('Density')
plt.show()
```

```
] sns.kdeplot(data['sulphates'], fill=True)
  plt.title('Probability Density Function for Sulphates')
  plt.xlabel('Value')
  plt.ylabel('Density')
  plt.show()
```

```
plt.figure(figsize=(6, 4))
sns.boxplot(data=chlorides)
plt.title('Box Plot for chlorides')
plt.xlabel('Chlorides')
plt.ylabel('Value')
plt.show()

plt.figure(figsize=(6, 4))
sns.boxplot(data=pH)
plt.title('Box Plot for pH')
plt.xlabel('pH')
plt.ylabel('Value')
plt.show()

plt.figure(figsize=(6, 4))
sns.boxplot(data=sulphates)
plt.title('Box Plot for sulphates')
plt.xlabel('Sulphates')
plt.ylabel('Value')
plt.show()
```

```
[15] def stem_and_leaf(data):
         data = sorted(data.tolist())

         stem_dict = {}

         for num in data:
             stem, leaf = divmod(int(num*100), 10)
             if stem not in stem_dict:
                 stem_dict[stem] = []
             stem_dict[stem].append(leaf)


         for stem in sorted(stem_dict.keys()):
             leaves = " ".join(str(leaf) for leaf in stem_dict[stem])
             print(f"{stem} | {leaves}")

     stem_and_leaf(data['sulphates'])
```

```
7] def remove_outliers(x):
       Q1 = x.quantile(0.25)
       Q3 = x.quantile(0.75)

       # Calculate the IQR
       IQR = Q3 - Q1

       lower = Q1 - 1.5 * IQR
       upper = Q3 + 1.5 * IQR

       x_fix = x[(x > lower) & (x < upper)]

       return x_fix

   chlorides = remove_outliers(data['chlorides'])
   pH = remove_outliers(data['pH'])
   sulphates = remove_outliers(data['sulphates'])

   print(chlorides, "\n", pH, "\n", sulphates)
```

```
def general_info(x):
  print("-"*14)
  print("Max: ", max(x))
  print("Min: ", round(min(x), 3))
  print("Median: ", statistics.median(x))

  print("-"*14)

  for i in range(1, 4):
    q = 0.25 * i
    print("Quantile ", q, " : ", x.quantile(q))

  print("-"*14)

  for i in range(1, 10):
    d = 0.1 * i
    print("Decile ", round(d, 1), " : ", x.quantile(d))

  print("-"*14)

print("Chlorides: ")
general_info(chlorides)
print("pH: ")
general_info(pH)
print("Sulphates: ")
general_info(sulphates)
```

```
[29] def central_char(x):
       print("-"*14)
       print("Expected value: ", statistics.mean(x))
       print("Geometric mean: ", statistics.geometric_mean(x))
       print("Harmonic mean: ", statistics.harmonic_mean(x))
       print("Moda: ", statistics.multimode(x))
       print("-"*14)

     print("Chlorides: ")
     central_char(chlorides)
     print("pH: ")
     central_char(pH)
     print("Sulphates: ")
     central_char(sulphates)
```

```
def dispersal_char(x):
  print("-"*14)
  print("Variance: ", statistics.variance(x))
  print("Standard Deviation: ", statistics.stdev(x))
  print("Coefficient of Variance: ", (statistics.stdev(x)/statistics.mean(x) * 100), "%")
  print("Probabilistic Deviation: ", 0.5*(x.quantile(0.75) - x.quantile(0.25)))
  print("Range of Sample: ", max(x) - min(x))
  print("Concentration Interval of Distribution: ", (statistics.mean(x) - 3 * statistics.stdev(x)),
        "-", (statistics.mean(x) + 3 * statistics.stdev(x)))
  print("-"*14)

print("Chlorides: ")
dispersal_char(chlorides)
print("pH: ")
dispersal_char(pH)
print("Sulphates: ")
dispersal_char(sulphates)
```

```
print("Skewness for Chlorides: ", skew(data["chlorides"]))
print("Skewness for pH: ", skew(data["pH"]))
print("Skewness for Sulphates: ", skew(data["sulphates"]))
```

```
print("Kurtosis for Chlorides: ", skew(chlorides))
print("Kurtosis for pH: ", skew(pH))
print("Kurtosis for Sulphates: ", skew(sulphates))
```

```
stat1, p_val1 = kstest(chlorides, "norm")
stat2, p_val2 = kstest(pH, "norm")
stat3, p_val3 = kstest(sulphates, "norm")

print("For chlorides: ")
print("Statistic: ", stat1, "\np-value: ", p_val1)
print("-"*14)
print("For pH: ")
print("Statistic: ", stat2, "\np-value: ", round(p_val2, 5))
print("-"*14)
print("For sulphates: ")
print("Statistic: ", stat3, "\np-value: ", p_val3)
```

```
] chl_norm1 = np.log(chlorides)
  pH_norm1 = np.log(pH)
  sul_norm1 = np.log(sulphates)
```

```
] s1, p1 = shapiro(chl_norm1)
  s2, p2 = shapiro(pH_norm1)
  s3, p3 = shapiro(sul_norm1)

  print("For chlorides: ")
  print("Statistic: ", s1, "\np-value: ", p1)
  print("-"*14)
  print("For pH: ")
  print("Statistic: ", s2, "\np-value: ", p2)
  print("-"*14)
  print("For sulphates: ")
  print("Statistic: ", s3, "\np-value: ", p3)
```

```
] chl_norm2 = np.sqrt(chlorides)
  pH_norm2 = np.sqrt(pH)
  sul_norm2 = np.sqrt(sulphates)
```

```
] s1, p1 = shapiro(chl_norm2)
  s2, p2 = shapiro(pH_norm2)
  s3, p3 = shapiro(sul_norm2)

  print("For chlorides: ")
  print("Statistic: ", s1, "\np-value: ", p1)
  print("-"*14)
  print("For pH: ")
  print("Statistic: ", s2, "\np-value: ", p2)
  print("-"*14)
  print("For sulphates: ")
  print("Statistic: ", s3, "\np-value: ", p3)
```

```
  probplot(chlorides.to_numpy(), dist="norm", plot=plt)
  plt.title('QQ Plot - Comparing chlorides to Normal Distribution')
  plt.show()
```

```
probplot(pH.to_numpy(), dist="norm", plot=plt)
plt.title('QQ Plot - Comparing pH to Normal Distribution')
plt.show()
```

```
probplot(sulphates.to_numpy(), dist="norm", plot=plt)
plt.title('QQ Plot - Comparing sulphates to Normal Distribution')
plt.show()
```

7 Список використаних джерел

1. О.С Слабоспицький. Аналіз даних. Попередня обробка: Навчальний посібник. – К.: Видавничо-поліграфічний центр “Київський університет”, 2001
2. https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85_(%D1%81%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0)
3. Лекції Аналіз Даних
4. https://www.rdocumentation.org/
5. https://docs.python.org/3/