# K03-T1-IF2220-13521004-13521007

April 17, 2023

## 1 Tugas Besar 1 - IF2220 Probabilitas dan Statistika

### 1.1 Dibuat Oleh


| NIM | Nama |
|---|---|
| 13521004 | Henry Anand Septian Radityo |
| 13521007 | Matthew Mahendra |


## 2 Setup

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats
from scipy import stats
from scipy.stats import norm
from scipy.stats import shapiro, probplot
from IPython.display import display, Markdown, Latex

df = pd.read_csv('anggur.csv')

# Cleanup untuk data NaN jika ada
df = df.dropna()
```

## 3 SOAL

Diberikan sebuah data anggur.csv yang dapat diakses pada utas berikut: Dataset Tugas Besar IF2220 merupakan data metrik kualitas wine (minuman anggur) yang mengandung 12 kolom sebagai berikut: 1. fixed acidity 2. volatile acidity 3. citric acid 4. residual sugar 5. chlorides 6. free sulfur dioxide 7. total sulfur dioxide 8. density 9. pH 10. sulphates 11. alcohol 12. quality

Kolom 1-11 adalah kolom atribut (non-target), sedangkan kolom 12 adalah kolom target. Anda diminta untuk melakukan analisis statistika sebagai berikut: 1. Menulis deskripsi statistika (Descriptive Statistics) dari semua kolom pada data yang bersifat numerik, terdiri dari mean, median,

modus, standar deviasi, variansi, range, nilai minimum, maksimum, kuartil, IQR, skewness dan kurtosis. Boleh juga ditambahkan deskripsi lain. 2. Membuat Visualisasi plot distribusi, dalam bentuk histogram dan boxplot untuk setiap kolom numerik. Berikan uraian penjelasan kondisi setiap kolom berdasarkan kedua plot tersebut. 3. Menentukan setiap kolom numerik berdistribusi normal atau tidak. Gunakan normality test yang dikaitkan dengan histogram plot. 4. Melakukan test hipotesis 1 sampel, - Nilai rata-rata pH di atas 3.29? - Nilai rata-rata Residual Sugar tidak sama dengan 2.50? - Nilai rata-rata 150 baris pertama kolom sulphates bukan 0.65? - Nilai rata-rata total sulfur dioxide di bawah 35 - Proporsi nilai total Sulfat Dioxide yang lebih dari 40, adalah tidak sama dengan 50% ?

5. Melakukan test hipotesis 2 sampel,

- Data kolom fixed acidity dibagi 2 sama rata: bagian awal dan bagian akhir kolom. Benarkah rata-rata kedua bagian tersebut sama?
- Data kolom chlorides dibagi 2 sama rata: bagian awal dan bagian akhir kolom. Benarkah rata-rata bagian awal lebih besar daripada bagian akhir sebesar 0.001?
- Benarkah rata-rata sampel 25 baris pertama kolom Volatile Acidity sama dengan rata-rata 25 baris pertama kolom Sulphates ?
- Bagian awal kolom residual sugar memiliki variansi yang sama dengan bagian akhirnya?
- Proporsi nilai setengah bagian awal alcohol yang lebih dari 7, adalah lebih besar daripada, proporsi nilai yang sama di setengah bagian akhir alcohol?

---

# 4 SOAL 1

Mean of Data

[2]:
```
# Mean dari setiap kolom
df.mean()
```

[2]:
```
fixed acidity           7.152530
volatile acidity        0.520839
citric acid             0.270517
residual sugar          2.567104
chlorides               0.081195
free sulfur dioxide    14.907679
total sulfur dioxide   40.290150
density                 0.995925
pH                      3.303610
sulphates               0.598390
alcohol                10.592280
quality                 7.958000
dtype: float64
```

Median of Data

[3]:
```
# Median dari setiap kolom
df.median()
```

[3]:
```
fixed acidity           7.150000
volatile acidity        0.524850
citric acid             0.272200
residual sugar          2.519430
chlorides               0.082167
free sulfur dioxide    14.860346
total sulfur dioxide   40.190000
density                 0.996000
pH                      3.300000
sulphates               0.595000
alcohol                10.610000
quality                 8.000000
dtype: float64
```

Standar Deviation of Data

[4]:
```
# Standar Deviation dari setiap kolom
df.std()
```

[4]:
```
fixed acidity           1.201598
volatile acidity        0.095848
citric acid             0.049098
residual sugar          0.987915
chlorides               0.020111
free sulfur dioxide     4.888100
total sulfur dioxide    9.965767
density                 0.002020
pH                      0.104875
sulphates               0.100819
alcohol                 1.510706
quality                 0.902802
dtype: float64
```

Variance of Data

[5]:
```
# Variance
df.std()**2
```

[5]:
```
fixed acidity           1.443837
volatile acidity        0.009187
citric acid             0.002411
residual sugar          0.975977
chlorides               0.000404
free sulfur dioxide    23.893519
total sulfur dioxide   99.316519
density                 0.000004
pH                      0.010999
sulphates               0.010164
```

```
alcohol                 2.282233
quality                 0.815051
dtype: float64
```

Range of Data

[6]:
```
# Range
df.max() - df.min()
```

[6]:
```
fixed acidity           8.170000
volatile acidity        0.665200
citric acid             0.292900
residual sugar          5.518200
chlorides               0.125635
free sulfur dioxide    27.267847
total sulfur dioxide   66.810000
density                 0.013800
pH                      0.740000
sulphates               0.670000
alcohol                 8.990000
quality                 5.000000
dtype: float64
```

Quantiles of Data

[7]:
```
# Q1
df.quantile(0.25)
```

[7]:
```
fixed acidity           6.377500
volatile acidity        0.456100
citric acid             0.237800
residual sugar          1.896330
chlorides               0.066574
free sulfur dioxide    11.426717
total sulfur dioxide   33.785000
density                 0.994600
pH                      3.230000
sulphates               0.530000
alcohol                 9.560000
quality                 7.000000
Name: 0.25, dtype: float64
```

[8]:
```
# Q2
df.quantile(0.5)
```

[8]:
```
fixed acidity           7.150000
volatile acidity        0.524850
citric acid             0.272200
```

```
residual sugar           2.519430
chlorides                0.082167
free sulfur dioxide     14.860346
total sulfur dioxide    40.190000
density                  0.996000
pH                       3.300000
sulphates                0.595000
alcohol                 10.610000
quality                  8.000000
Name: 0.5, dtype: float64
```

[9]:
```
# Q3
df.quantile(0.75)
```

[9]:
```
fixed acidity            8.000000
volatile acidity         0.585375
citric acid              0.302325
residual sugar           3.220873
chlorides                0.095312
free sulfur dioxide     18.313098
total sulfur dioxide    47.022500
density                  0.997200
pH                       3.370000
sulphates                0.670000
alcohol                 11.622500
quality                  9.000000
Name: 0.75, dtype: float64
```

Inter-Quantile Range of Data

[10]:
```
# IQR
df.quantile(0.75) - df.quantile(0.25)
```

[10]:
```
fixed acidity            1.622500
volatile acidity         0.129275
citric acid              0.064525
residual sugar           1.324544
chlorides                0.028738
free sulfur dioxide      6.886381
total sulfur dioxide    13.237500
density                  0.002600
pH                       0.140000
sulphates                0.140000
alcohol                  2.062500
quality                  2.000000
dtype: float64
```

Skewness of Data

[11]:
```
# Skewness
df.skew()
```

[11]:
```
fixed acidity          -0.028879
volatile acidity       -0.197699
citric acid            -0.045576
residual sugar          0.132638
chlorides              -0.051319
free sulfur dioxide     0.007130
total sulfur dioxide   -0.024060
density                -0.076883
pH                      0.147673
sulphates               0.149199
alcohol                -0.018991
quality                -0.089054
dtype: float64
```

Kurtosis of Data

[12]:
```
# Kurtosis
df.kurtosis()
```

[12]:
```
fixed acidity          -0.019292
volatile acidity        0.161853
citric acid            -0.104679
residual sugar         -0.042980
chlorides              -0.246508
free sulfur dioxide    -0.364964
total sulfur dioxide    0.063950
density                 0.016366
pH                      0.080910
sulphates               0.064819
alcohol                -0.131732
quality                 0.108291
dtype: float64
```

Mode of Data

[13]:
```
df.mode(numeric_only=True, dropna=True)
```

[13]:

| | fixed acidity | volatile acidity | citric acid | residual sugar | chlorides | \ |
|---|---|---|---|---|---|---|
| 0 | 6.54 | 0.5546 | 0.3019 | 0.032555 | 0.015122 | |
| 1 | NaN | NaN | NaN | 0.033333 | 0.020794 | |
| 2 | NaN | NaN | NaN | 0.051774 | 0.024259 | |
| 3 | NaN | NaN | NaN | 0.077156 | 0.027209 | |
| 4 | NaN | NaN | NaN | 0.084744 | 0.032111 | |
| .. | ... | ... | ... | ... | ... | |
| 995 | NaN | NaN | NaN | 5.210260 | 0.131425 | |

```
996            NaN               NaN          NaN        5.217429   0.133656
997            NaN               NaN          NaN        5.252864   0.135368
998            NaN               NaN          NaN        5.299524   0.135790
999            NaN               NaN          NaN        5.550755   0.140758

     free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0               0.194679                 35.20   0.9959  3.34       0.59
1               0.621628                 37.25   0.9961   NaN        NaN
2               0.860177                 39.64   0.9965   NaN        NaN
3               3.032139                 40.61   0.9970   NaN        NaN
4               3.129885                 41.05      NaN   NaN        NaN
..                   ...                   ...      ...   ...        ...
995            26.630490                   NaN      NaN   NaN        NaN
996            26.665773                   NaN      NaN   NaN        NaN
997            26.822626                   NaN      NaN   NaN        NaN
998            27.006307                   NaN      NaN   NaN        NaN
999            27.462525                   NaN      NaN   NaN        NaN

     alcohol  quality
0       9.86      8.0
1      10.31      NaN
2        NaN      NaN
3        NaN      NaN
4        NaN      NaN
..       ...      ...
995      NaN      NaN
996      NaN      NaN
997      NaN      NaN
998      NaN      NaN
999      NaN      NaN

[1000 rows x 12 columns]
```

[14]:
```python
# Mode
df['fixed acidity'].mode()

for cols in df:
    temp = "Modus dari " + cols + ":"
    display(Markdown(f"**{temp}**"))
    modus = []
    for i in df[cols].mode():
        modus.append(i)
    display(pd.DataFrame(modus))
```

**Modus dari fixed acidity:**

```
      0
0  6.54
```

**Modus dari volatile acidity:**

```
        0
0  0.5546
```

**Modus dari citric acid:**

```
        0
0  0.3019
```

**Modus dari residual sugar:**

```
            0
0    0.032555
1    0.033333
2    0.051774
3    0.077156
4    0.084744
..        ...
995  5.210260
996  5.217429
997  5.252864
998  5.299524
999  5.550755

[1000 rows x 1 columns]
```

**Modus dari chlorides:**

```
            0
0    0.015122
1    0.020794
2    0.024259
3    0.027209
4    0.032111
..        ...
995  0.131425
996  0.133656
997  0.135368
998  0.135790
999  0.140758

[1000 rows x 1 columns]
```

**Modus dari free sulfur dioxide:**

```
             0
0     0.194679
1     0.621628
2     0.860177
3     3.032139
4     3.129885
```

```
..          ...
995  26.630490
996  26.665773
997  26.822626
998  27.006307
999  27.462525

[1000 rows x 1 columns]
```

**Modus dari total sulfur dioxide:**

```
       0
0  35.20
1  37.25
2  39.64
3  40.61
4  41.05
5  41.59
6  44.51
```

**Modus dari density:**

```
        0
0  0.9959
1  0.9961
2  0.9965
3  0.9970
```

**Modus dari pH:**

```
      0
0  3.34
```

**Modus dari sulphates:**

```
      0
0  0.59
```

**Modus dari alcohol:**

```
       0
0   9.86
1  10.31
```

**Modus dari quality:**

```
   0
0  8
```

---

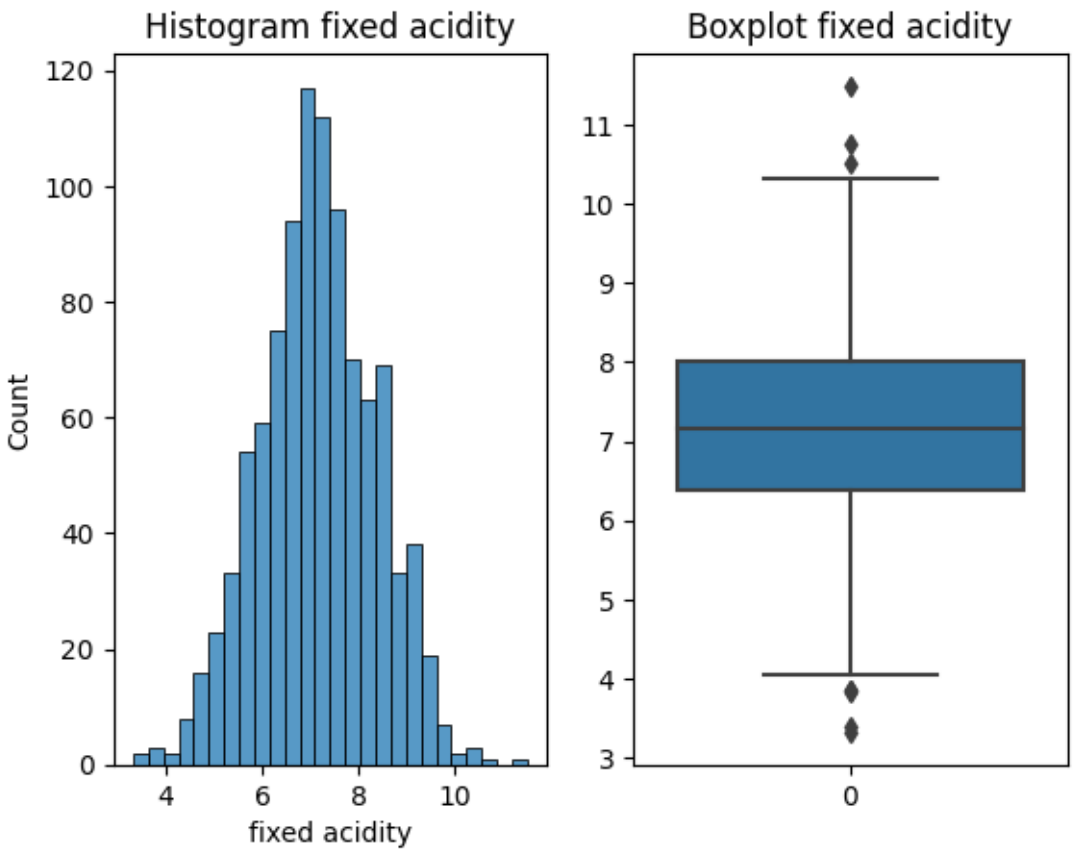
# 5 SOAL 2

Membuat Visualisasi plot distribusi, dalam bentuk histogram dan boxplot untuk setiap kolom numerik. Berikan uraian penjelasan kondisi setiap kolom berdasarkan kedua plot tersebut.

## 5.1 Kolom "fixed acidity"

[15]:
```
var = df['fixed acidity']
plt.figure(figsize=(7,6))
fig,ax = plt.subplots(1,2)
# df[var].hist(ax = ax[0]) matplotlib
sns.histplot(var, ax=ax[0])
ax[0].set_title("Histogram fixed acidity")
ax[1].set_title("Boxplot fixed acidity")
# df.boxplot(var,ax=ax[1]) matplotlib
sns.boxplot(var,ax=ax[1])
plt.show()
```

```
<Figure size 700x600 with 0 Axes>
```




Dari histogram tersebut dapat dilihat bahwa persebaran data untuk kolom fixed acidity cenderung

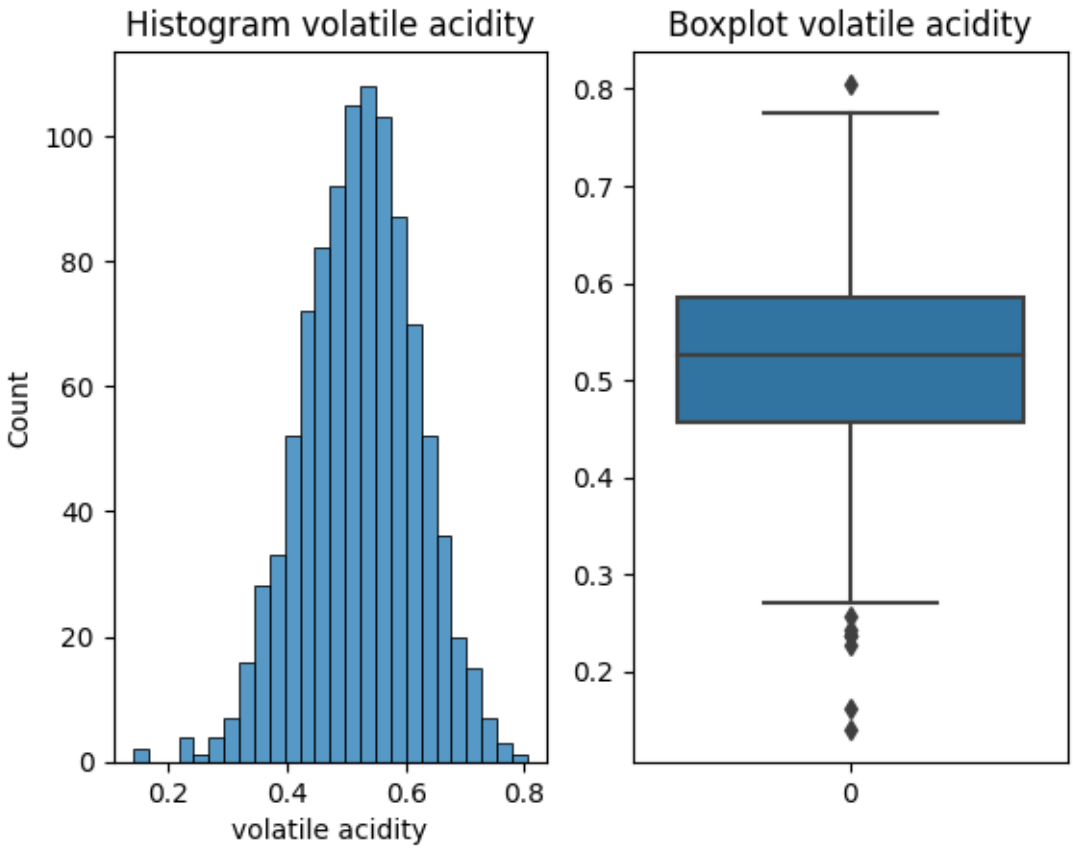
terdistribusi secara merata apabila dilihat dari grafik yang berbentuk simetris seperti lonceng dan titik tertinggi berada di bagian tengah dari data dan rata-rata data yang ada di angka 7.15.

Grafik Box Plot juga menunjukkan data yang cenderung **terdistribusi normal** yang ditunjukkan dari interquartile range yang berada di tengah nilai minimum dan maksimum. Dari box plot juga dapat dilihat terdapat beberapa data outlier yaitu data yang berada diluar batas atas dan batas bawah. Terdapat 3 data yang berada di bawah batas bawah dan 3 data berada diatas batas atas.

## 5.2 Kolom "folatile acidity"

[16]:
```
var = df['volatile acidity']
plt.figure(figsize=(7,6))
fig,ax = plt.subplots(1,2)
# df[var].hist(ax = ax[0]) matplotlib
sns.histplot(var, ax=ax[0])
ax[0].set_title("Histogram volatile acidity")
ax[1].set_title("Boxplot volatile acidity")
# df.boxplot(var,ax=ax[1]) matplotlib
sns.boxplot(var,ax=ax[1])
plt.show()
```

```
<Figure size 700x600 with 0 Axes>
```



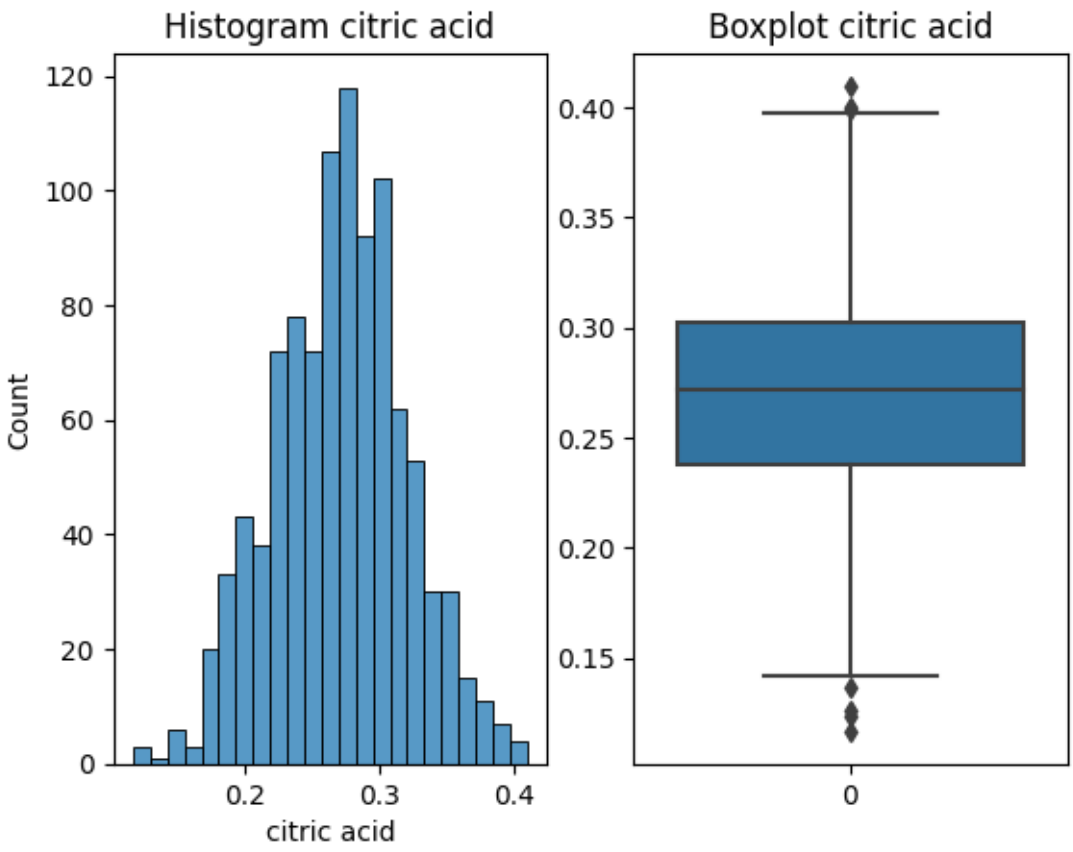
Dari histogram tersebut dapat dilihat bahwa persebaran data untuk kolom folatile acidity memiliki persebaran paling banyak pada nilai di sekitar 0.5. Grafik Histogram terlihat simetris dengan persebaran yang cukup merata.

Grafik Box Plot juga menunjukkan data outlier yaitu data yang berada diluar batas atas dan batas bawah. Terdapat 6 data yang berada di bawah batas bawah dan 1 data berada diatas batas atas.

### 5.3 Kolom "citric acid"

[17]:
```
var = df['citric acid']
plt.figure(figsize=(7,6))
fig,ax = plt.subplots(1,2)
# df[var].hist(ax = ax[0]) matplotlib
sns.histplot(var, ax=ax[0])
ax[0].set_title("Histogram citric acid")
ax[1].set_title("Boxplot citric acid")
# df.boxplot(var,ax=ax[1]) matplotlib
sns.boxplot(var,ax=ax[1])
plt.show()
```

```
<Figure size 700x600 with 0 Axes>
```



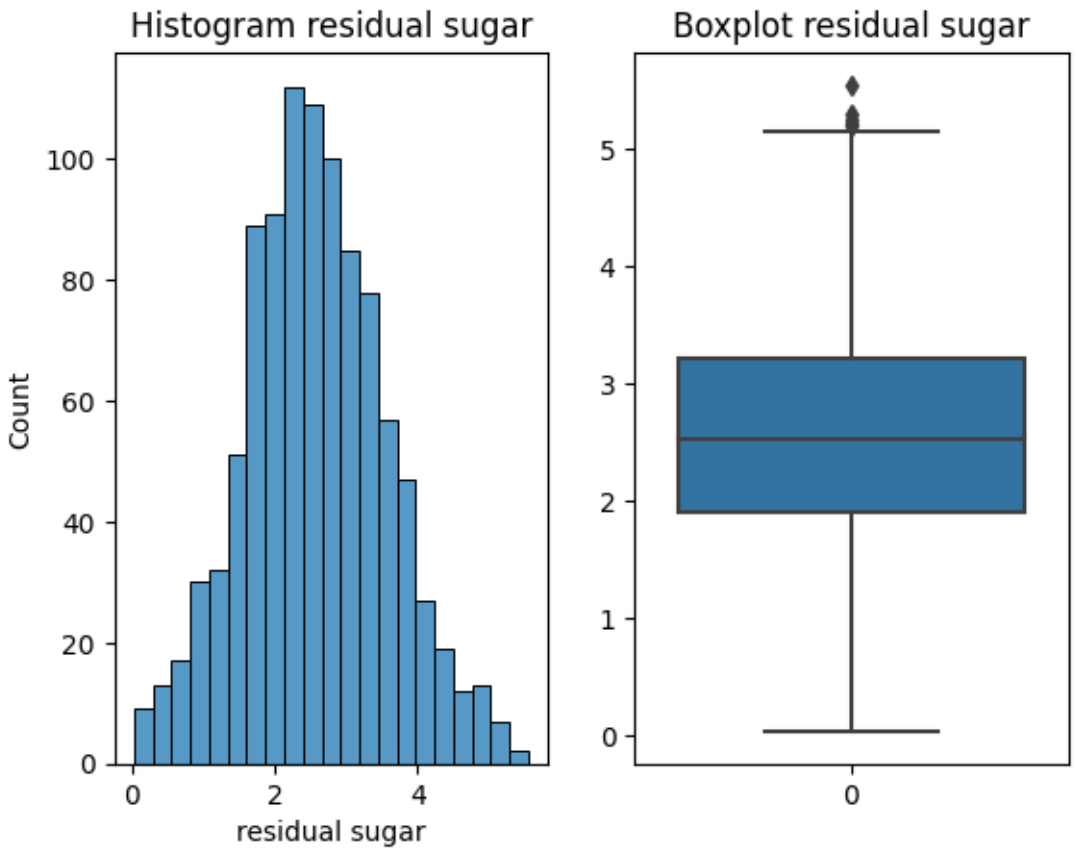
Dari histogram tersebut dapat dilihat bahwa persebaran data untuk kolom citric acid **terdistribusi normal**, apabila dilihat dari grafik yang berbentuk simetris seperti lonceng dan titik tertinggi berada di bagian tengah dari data dan rata-rata data yang ada di angka 0.27. Distribusi normal juga dapat dilihat dari kurva bagian samping yang memiliki frekuensi lebih sedikit dibanding dengan bagian tengah.

Grafik Box Plot juga menunjukkan data yang **terdistribusi normal** yang ditunjukkan dari interquartile range yang berada di tengah nilai minimum dan maksimum. Dari box plot juga dapat dilihat terdapat beberapa data outlier yaitu data yang berada diluar batas atas dan batas bawah. Terdapat 4 data yang berada di bawah batas bawah dan 2 data berada diatas batas atas.

### 5.4 Kolom "residual sugar"

[18]:
```
var = df['residual sugar']
plt.figure(figsize=(7,6))
fig,ax = plt.subplots(1,2)
# df[var].hist(ax = ax[0]) matplotlib
sns.histplot(var, ax=ax[0])
ax[0].set_title("Histogram residual sugar")
ax[1].set_title("Boxplot residual sugar")
# df.boxplot(var,ax=ax[1]) matplotlib
sns.boxplot(var,ax=ax[1])
plt.show()
```

```
<Figure size 700x600 with 0 Axes>
```




Dari histogram tersebut dapat dilihat bahwa persebaran data untuk kolom residual sugar **terdistribusi normal**, apabila dilihat dari grafik yang berbentuk simetris seperti lonceng dan titik tertinggi berada di bagian tengah dari data dan rata-rata data yang ada di angka 2.57. Distribusi normal juga dapat dilihat dari kurva bagian samping yang memiliki frekuensi lebih sedikit dibanding dengan bagian tengah.

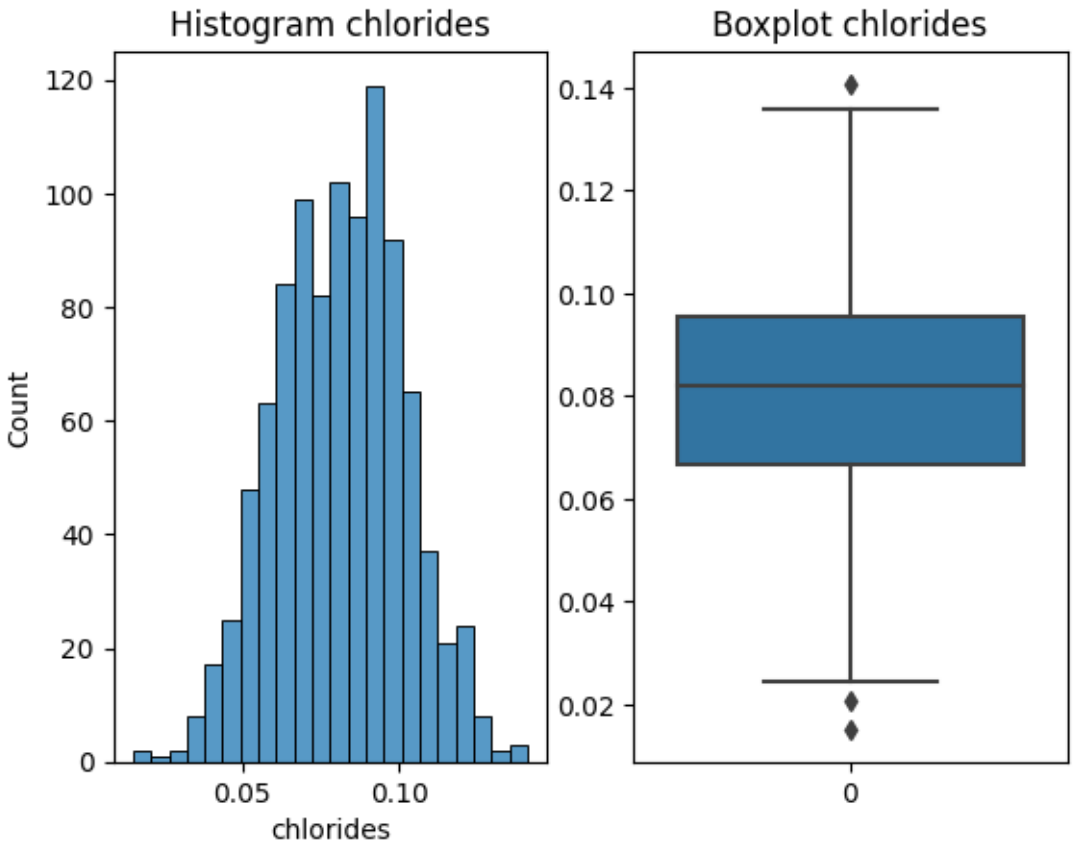
Grafik Box Plot juga menunjukkan data yang **terdistribusi normal** yang ditunjukkan dari *interquartile range* yang berada di tengah nilai minimum dan maksimum. Dari box plot juga dapat dilihat terdapat beberapa data outlier yaitu data yang berada diluar batas atas dan batas bawah. Terdapat beberapa data outlier yang memiliki nilai lebih dari batas atas.

### 5.5 Kolom "chlorides"

[19]:
```
var = df['chlorides']
plt.figure(figsize=(7,6))
fig,ax = plt.subplots(1,2)
# df[var].hist(ax = ax[0]) matplotlib
sns.histplot(var, ax=ax[0])
ax[0].set_title("Histogram chlorides")
ax[1].set_title("Boxplot chlorides")
# df.boxplot(var,ax=ax[1]) matplotlib
```

```
sns.boxplot(var,ax=ax[1])
plt.show()

# # plot histogram
# sns.histplot(var, kde=True)
# plt.show()

# plot QQ plot
# probplot(var, dist="norm", plot=plt)
# plt.show()
```

```
<Figure size 700x600 with 0 Axes>
```




Dari histogram tersebut dapat dilihat bahwa persebaran data untuk kolom chlorides **terdistribusi normal**, apabila dilihat dari grafik yang berbentuk simetris seperti lonceng dan titik tertinggi berada di bagian tengah dari data dan rata-rata data yang ada di angka 0.08. Distribusi normal juga dapat dilihat dari kurva bagian samping yang memiliki frekuensi lebih sedikit dibanding dengan bagian tengah.

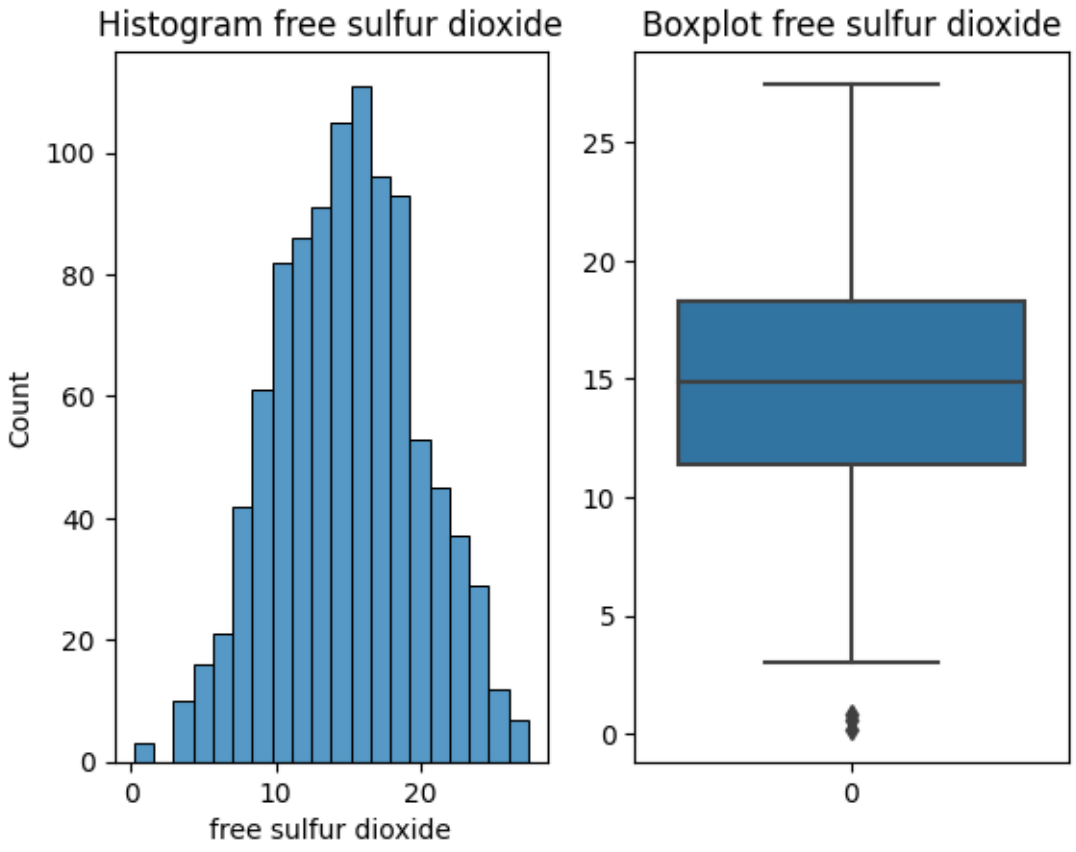
Grafik Box Plot juga menunjukkan data yang **terdistribusi normal** yang ditunjukkan dari *interquartile range* yang berada di tengah nilai minimum dan maksimum. Dari box plot juga dapat

dilihat terdapat beberapa data outlier yaitu data yang berada diluar batas atas dan batas bawah. Terdapat beberapa data outlier yang memiliki nilai lebih dari batas atas yaitu 1 data dan terdapat pula 2 data kurang dari batas bawah.

### 5.6 Kolom "free sulfur dioxide"

[20]:
```
var = df['free sulfur dioxide']
plt.figure(figsize=(7,6))
fig,ax = plt.subplots(1,2)
# df[var].hist(ax = ax[0]) matplotlib
sns.histplot(var, ax=ax[0])
ax[0].set_title("Histogram free sulfur dioxide")
ax[1].set_title("Boxplot free sulfur dioxide")
# df.boxplot(var,ax=ax[1]) matplotlib
sns.boxplot(var,ax=ax[1])
plt.show()

# probplot(var, dist="norm", plot=plt)
# plt.show()
```

```
<Figure size 700x600 with 0 Axes>
```



Grafik Box Plot juga menunjukkan data yang cukup merata yang ditunjukkan dari *interquartile range* yang berada di tengah nilai minimum dan maksimum. Dari box plot juga dapat dilihat terdapat beberapa data outlier yaitu data yang berada diluar batas atas dan batas bawah. Terdapat beberapa data outlier yang memiliki nilai kurang dari batas bawah.

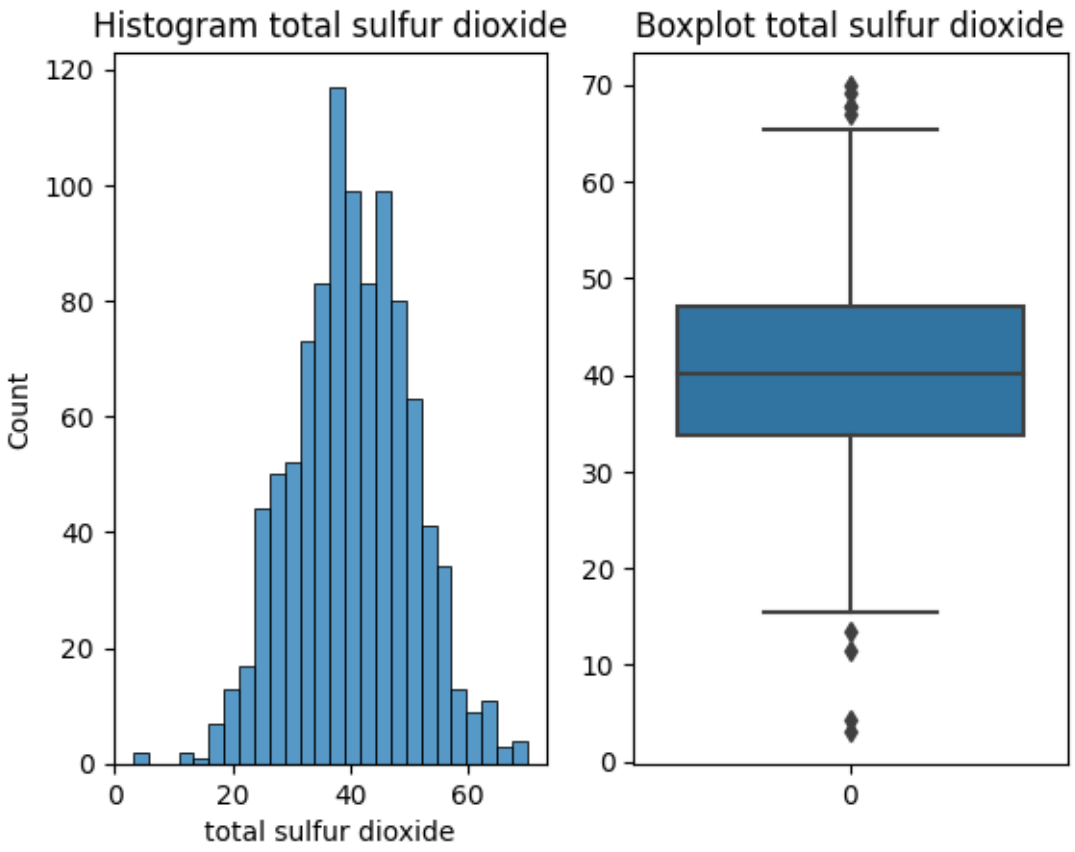
Dari histogram tersebut dapat dilihat bahwa persebaran data untuk kolom free sulfur dioxide memiliki persebaran paling banyak pada nilai di sekitar 14.9. Grafik Histogram terlihat cukup simetris dengan persebaran yang cukup merata.

Grafik Box Plot juga menunjukkan data outlier yaitu data yang berada diluar batas atas dan batas bawah. Terdapat 6 data yang berada di bawah batas bawah dan 1 data berada diatas batas atas.

### 5.7 Kolom "total sulfur dioxide"

[21]:
```
var = df['total sulfur dioxide']
plt.figure(figsize=(7,6))
fig,ax = plt.subplots(1,2)
# df[var].hist(ax = ax[0]) matplotlib
sns.histplot(var, ax=ax[0])
ax[0].set_title("Histogram total sulfur dioxide")
ax[1].set_title("Boxplot total sulfur dioxide")
# df.boxplot(var,ax=ax[1]) matplotlib
sns.boxplot(var,ax=ax[1])
plt.show()
```

```
<Figure size 700x600 with 0 Axes>
```




Dari histogram tersebut dapat dilihat bahwa persebaran data untuk kolom total sulfur dioxide **terdistribusi normal**, apabila dilihat dari grafik yang berbentuk simetris seperti lonceng dan titik tertinggi berada di bagian tengah dari data dan rata-rata data yang ada di angka 40.29. Distribusi normal juga dapat dilihat dari kurva bagian samping yang memiliki frekuensi lebih sedikit dibanding dengan bagian tengah.

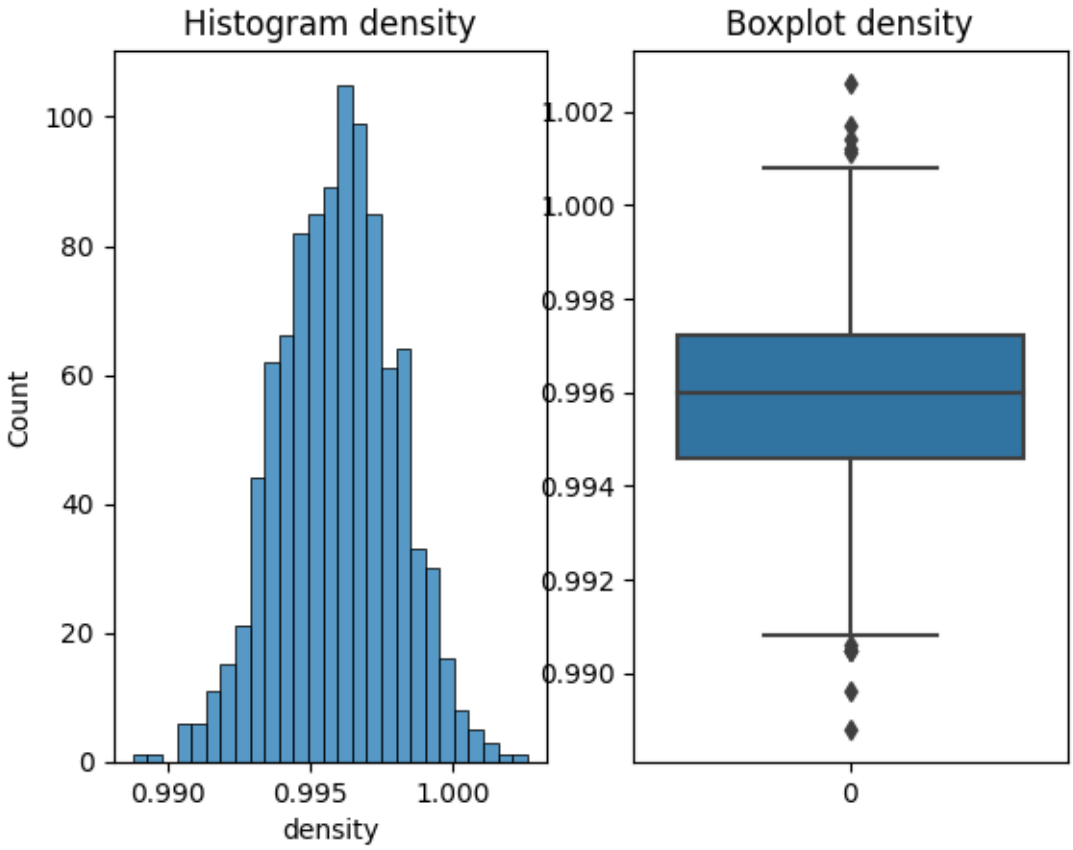
Grafik Box Plot juga menunjukkan data yang **terdistribusi normal** yang ditunjukkan dari *interquartile range* yang berada di tengah nilai minimum dan maksimum. Dari box plot juga dapat dilihat terdapat beberapa data outlier yaitu data yang berada diluar batas atas dan batas bawah. Terdapat beberapa data outlier yang memiliki nilai kurang dari batas bawah dan lebih dari batas atas.

### 5.8 Kolom "density"

[22]:
```
var = df['density']
plt.figure(figsize=(7,6))
fig,ax = plt.subplots(1,2)
# df[var].hist(ax = ax[0]) matplotlib
sns.histplot(var, ax=ax[0])
ax[0].set_title("Histogram density")
```

```
ax[1].set_title("Boxplot density")
# df.boxplot(var,ax=ax[1]) matplotlib
sns.boxplot(var,ax=ax[1])
plt.show()
```

```
<Figure size 700x600 with 0 Axes>
```



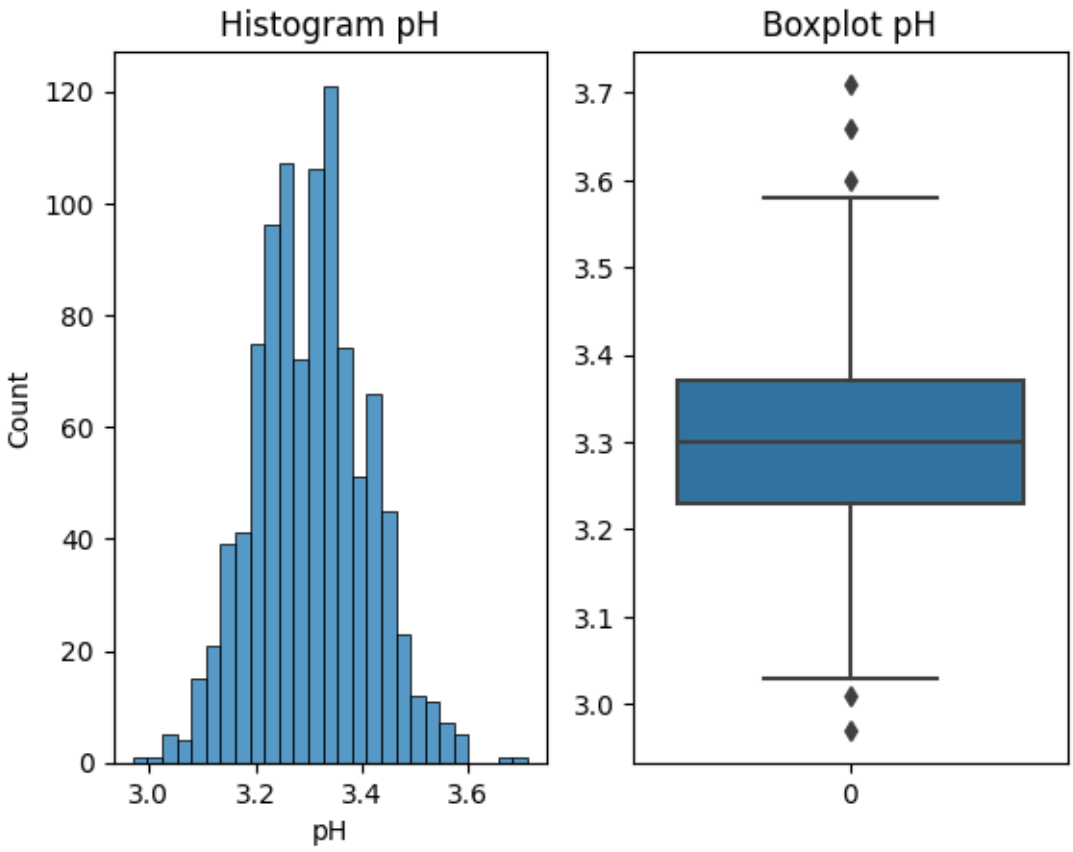

Dari histogram tersebut dapat dilihat bahwa persebaran data untuk kolom density **terdistribusi normal**, apabila dilihat dari grafik yang berbentuk simetris seperti lonceng dan titik tertinggi berada di bagian tengah dari data dan rata-rata data yang ada di angka 0.99. Distribusi normal juga dapat dilihat dari kurva bagian samping yang memiliki frekuensi lebih sedikit dibanding dengan bagian tengah.

Grafik Box Plot juga menunjukkan data yang **terdistribusi normal** yang ditunjukkan dari *interquartile range* yang berada di tengah nilai minimum dan maksimum. Dari box plot juga dapat dilihat terdapat beberapa data outlier yaitu data yang berada diluar batas atas dan batas bawah. Terdapat beberapa data outlier yang memiliki nilai kurang dari batas bawah dan lebih dari batas atas.

### 5.9 Kolom "pH"

[23]:
```
var = df['pH']
plt.figure(figsize=(7,6))
fig,ax = plt.subplots(1,2)
# df[var].hist(ax = ax[0]) matplotlib
sns.histplot(var, ax=ax[0])
ax[0].set_title("Histogram pH")
ax[1].set_title("Boxplot pH")
# df.boxplot(var,ax=ax[1]) matplotlib
sns.boxplot(var,ax=ax[1])
plt.show()
```

```
<Figure size 700x600 with 0 Axes>
```



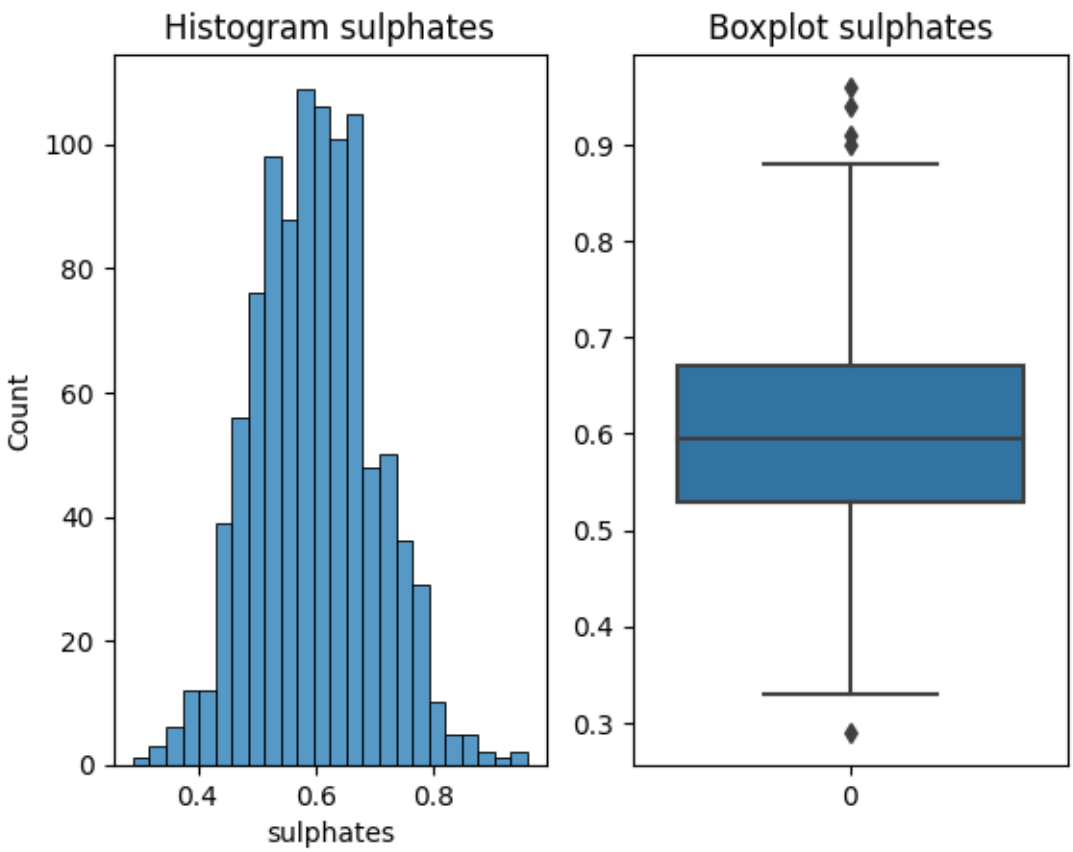

Dari histogram tersebut dapat dilihat bahwa persebaran data untuk kolom pH **terdistribusi normal**, apabila dilihat dari grafik yang berbentuk simetris seperti lonceng dan titik tertinggi berada di bagian tengah dari data dan rata-rata data yang ada di angka 3.30. Distribusi normal juga dapat dilihat dari kurva bagian samping yang memiliki frekuensi lebih sedikit dibanding dengan bagian tengah.

Grafik Box Plot juga menunjukkan data yang **terdistribusi normal** yang ditunjukkan dari *interquartile range* yang berada di tengah nilai minimum dan maksimum. Dari box plot juga dapat dilihat terdapat beberapa data outlier yaitu data yang berada diluar batas atas dan batas bawah. Terdapat 2 data outlier yang memiliki nilai kurang dari batas bawah dan 3 data yang memiliki nilai lebih dari batas atas.

### 5.10 Kolom "sulphates"

[24]:
```
var = df['sulphates']
plt.figure(figsize=(7,6))
fig,ax = plt.subplots(1,2)
# df[var].hist(ax = ax[0]) matplotlib
sns.histplot(var, ax=ax[0])
ax[0].set_title("Histogram sulphates")
ax[1].set_title("Boxplot sulphates")
# df.boxplot(var,ax=ax[1]) matplotlib
sns.boxplot(var,ax=ax[1])
plt.show()
```

```
<Figure size 700x600 with 0 Axes>
```



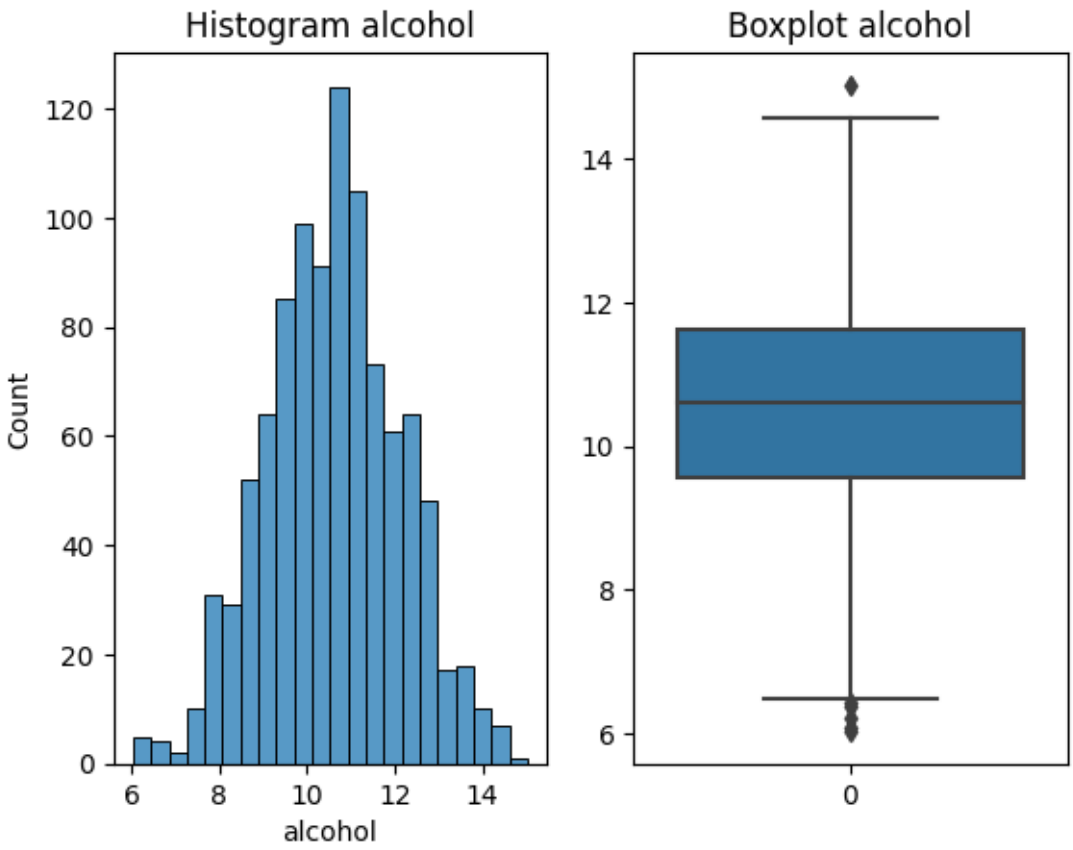
Dari histogram tersebut dapat dilihat bahwa persebaran data untuk kolom sulphates **terdistribusi normal**, apabila dilihat dari grafik yang berbentuk simetris seperti lonceng dan titik tertinggi berada di bagian tengah dari data dan rata-rata data yang ada di angka 0.60. Distribusi normal juga dapat dilihat dari kurva bagian samping yang memiliki frekuensi lebih sedikit dibanding dengan bagian tengah.

Grafik Box Plot juga menunjukkan data yang **terdistribusi normal** yang ditunjukkan dari *interquartile range* yang berada di tengah nilai minimum dan maksimum. Dari box plot juga dapat dilihat terdapat beberapa data outlier yaitu data yang berada diluar batas atas dan batas bawah. Terdapat 1 data outlier yang memiliki nilai kurang dari batas bawah dan 4 data yang memiliki nilai lebih dari batas atas.

### 5.11 Kolom "alcohol"

[25]:
```
var = df['alcohol']
plt.figure(figsize=(7,6))
fig,ax = plt.subplots(1,2)
# df[var].hist(ax = ax[0]) matplotlib
sns.histplot(var, ax=ax[0])
ax[0].set_title("Histogram alcohol")
ax[1].set_title("Boxplot alcohol")
# df.boxplot(var,ax=ax[1]) matplotlib
sns.boxplot(var,ax=ax[1])
plt.show()
```

```
<Figure size 700x600 with 0 Axes>
```




[26]:
```
var.mean()
```

[26]:
```
10.592279999999999
```

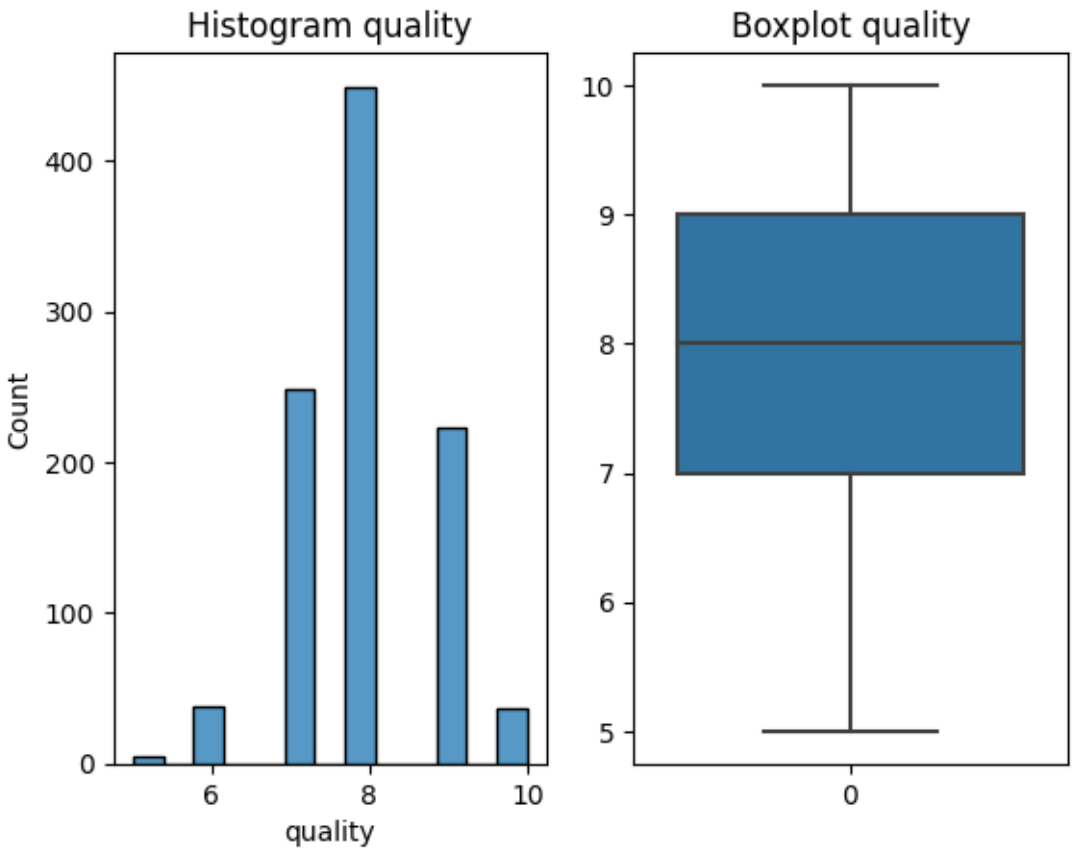
Dari histogram tersebut dapat dilihat bahwa persebaran data untuk kolom alcohol **terdistribusi normal**, apabila dilihat dari grafik yang berbentuk simetris seperti lonceng dan titik tertinggi berada di bagian tengah dari data dan rata-rata data yang ada di angka 10.59. Distribusi normal juga dapat dilihat dari kurva bagian samping yang memiliki frekuensi lebih sedikit dibanding dengan bagian tengah.

Grafik Box Plot juga menunjukkan data yang **terdistribusi normal** yang ditunjukkan dari *interquartile range* yang berada di tengah nilai minimum dan maksimum. Dari box plot juga dapat dilihat terdapat beberapa data outlier yaitu data yang berada diluar batas atas dan batas bawah. Terdapat beberapa data outlier yang memiliki nilai kurang dari batas bawah dan 1 data yang memiliki nilai lebih dari batas atas.

[27]:
```
var = df['quality']
plt.figure(figsize=(7,6))
fig,ax = plt.subplots(1,2)
# df[var].hist(ax = ax[0]) matplotlib
sns.histplot(var, ax=ax[0])
```

```
ax[0].set_title("Histogram quality")
ax[1].set_title("Boxplot quality")
# df.boxplot(var,ax=ax[1]) matplotlib
sns.boxplot(var,ax=ax[1])
plt.show()
```

```
<Figure size 700x600 with 0 Axes>
```




# 6 SOAL 3

Dibawah ini terdefinisi prosedur untuk pengecekan suatu data terdistribusi normal atau tidak. Menggunakan D'Agostino Test dan Pearson Test yang menggabungkan tes skewness dan kurtosis dengan hasil

$$s^2 + k^2$$

dengan s nilai z dari skewness test dan k nilai z dari kurtosis test

Parameter yang digunakan:

$H_0$: Data berdistribusi normal

$H_1$: Data tidak berdistribusi normal

$\alpha$: 0.05

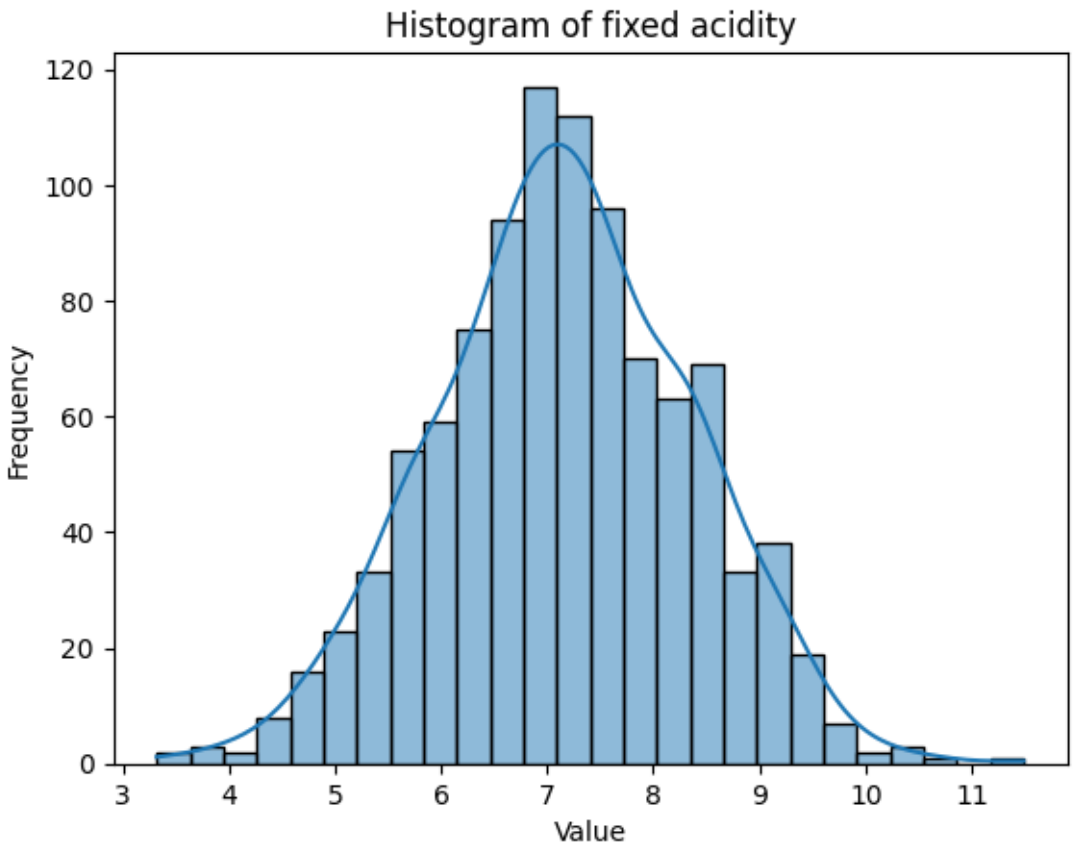
Jika nilai $p < \alpha$, maka $H_0$ ditolak, sebaliknya $p \geq \alpha$, maka $H_0$ diterima

[28]:
```
from scipy.stats import normaltest

def normal_test(col):
    stat, p = normaltest(df[col])

    if p > 0.05:
        display(Markdown("Kolom " + col +" **terdistribusi normal**"))
    else:
        display(Markdown("Kolom " + col +" **tidak terdistribusi normal**"))

    sns.histplot(df[col], bins="auto", kde = True)
    plt.title('Histogram of ' + col)
    plt.xlabel('Value')
    plt.ylabel('Frequency')
    plt.show()
```

### 6.1 Kolom Fixed Acidity

[29]:
```
normal_test('fixed acidity')
```

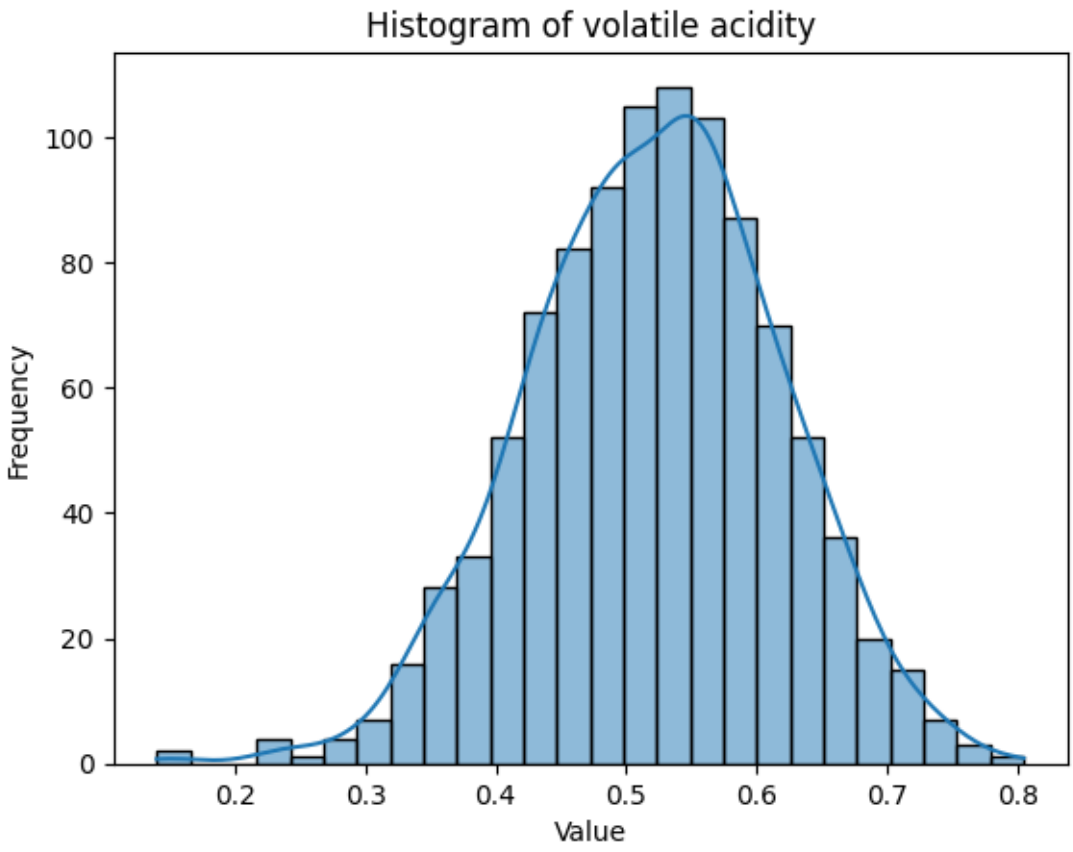
Kolom fixed acidity **terdistribusi normal**




### 6.2 Kolom Volatile Acidity

[30]:
```
normal_test('volatile acidity')
```

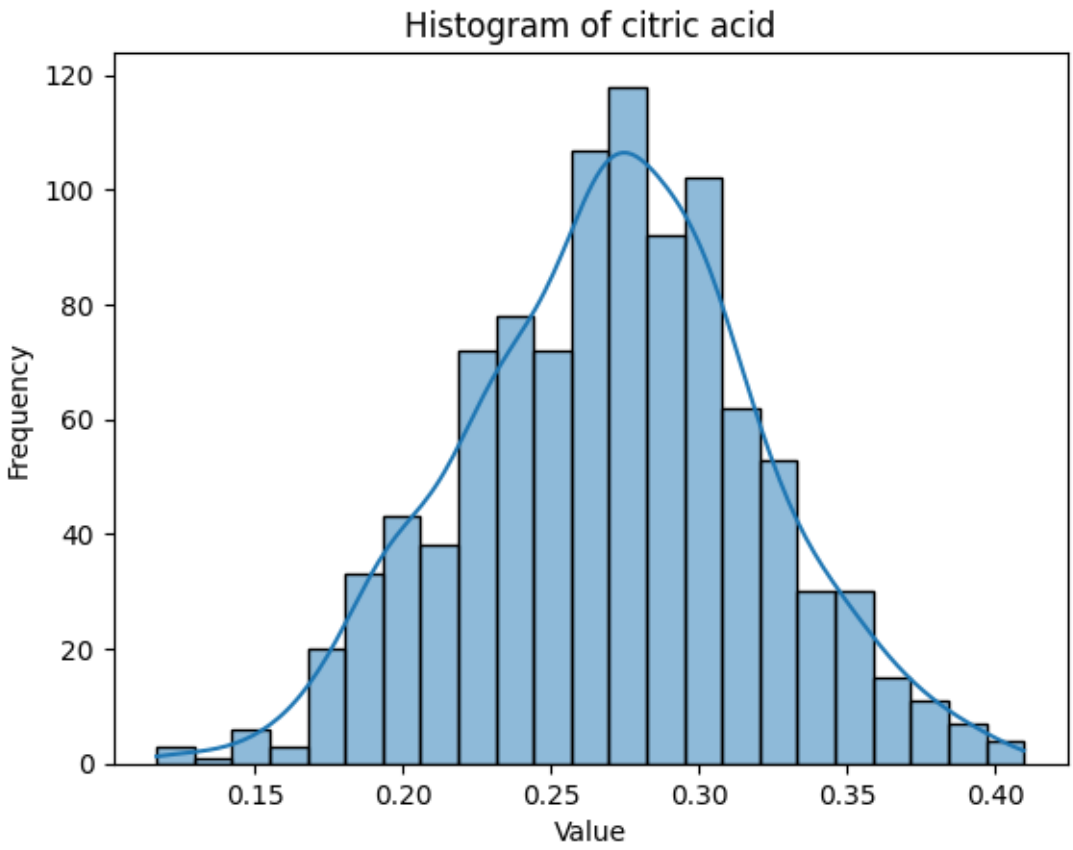
Kolom volatile acidity **tidak terdistribusi normal**




### 6.3 Kolom Citric Acid

[31]:
```
normal_test('citric acid')
```

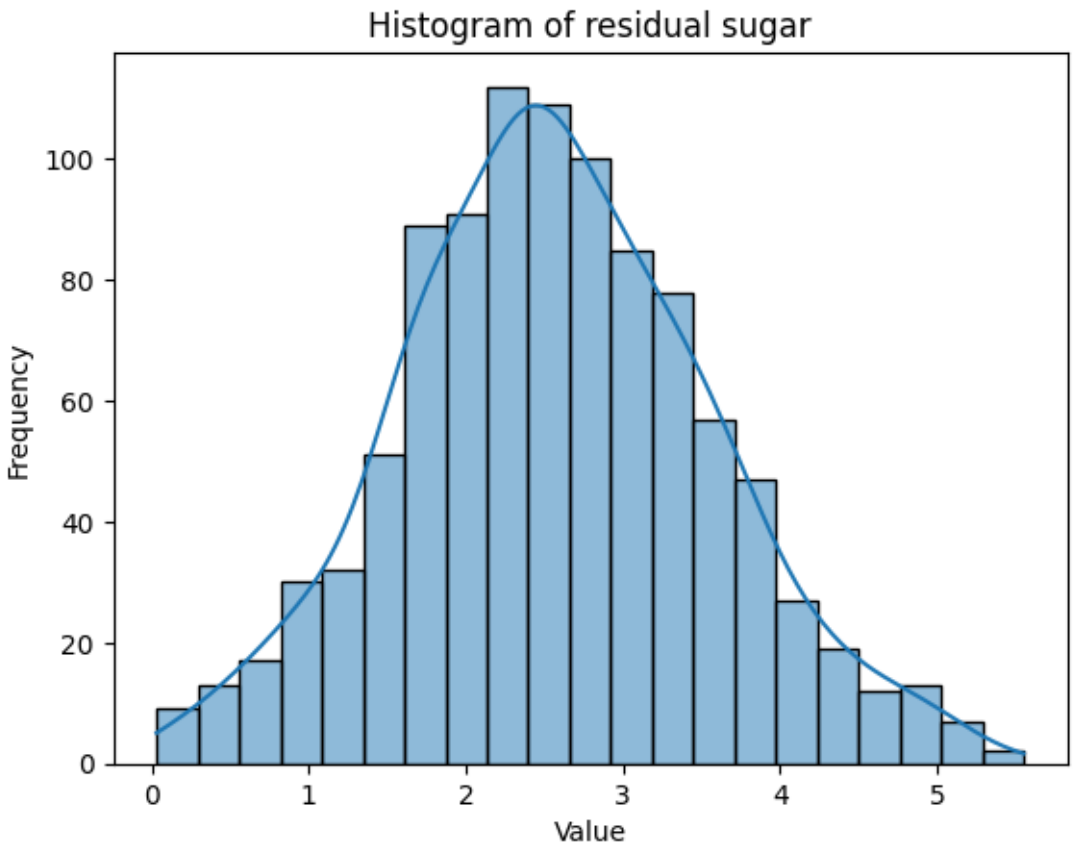
Kolom citric acid **terdistribusi normal**




### 6.4 Kolom Residual Sugar

[32]:
```
normal_test('residual sugar')
```

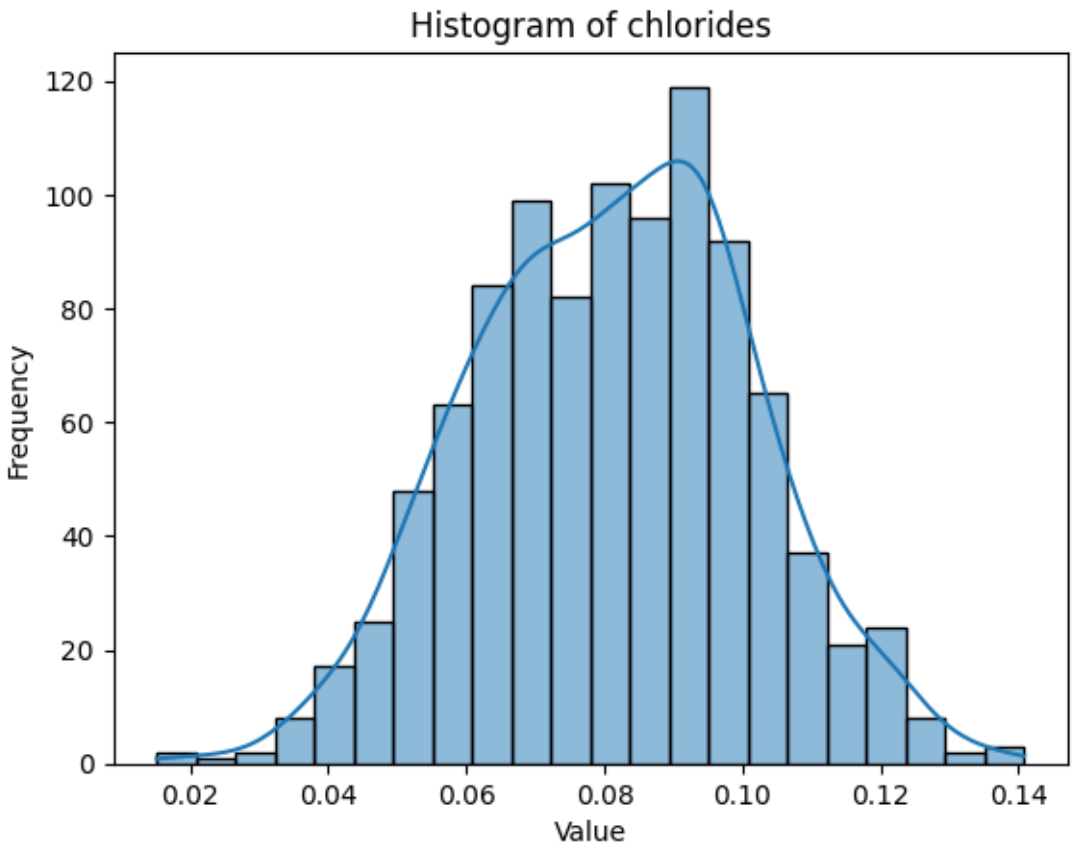
Kolom residual sugar **terdistribusi normal**




### 6.5 Kolom Chlorides

[33]:
```
normal_test('chlorides')
```

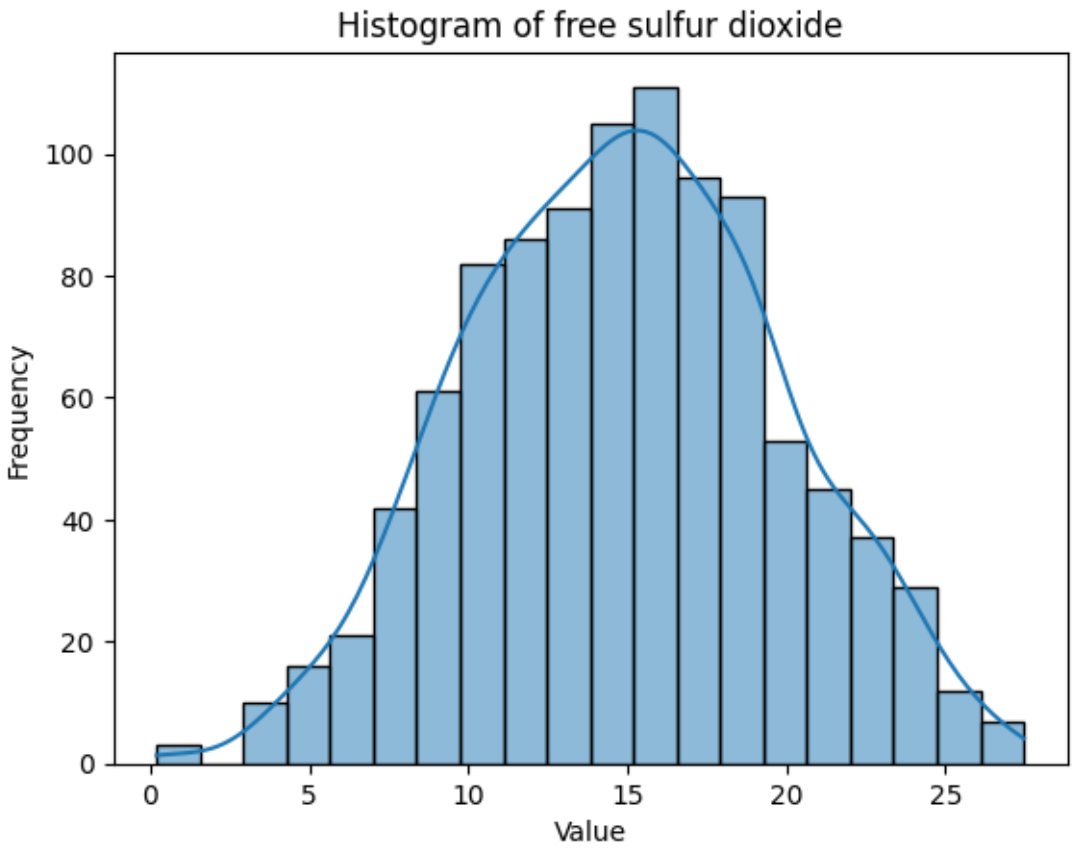
Kolom chlorides **terdistribusi normal**




### 6.6 Kolom Free Sulfur Dioxide

[34]:
```
normal_test('free sulfur dioxide')
```

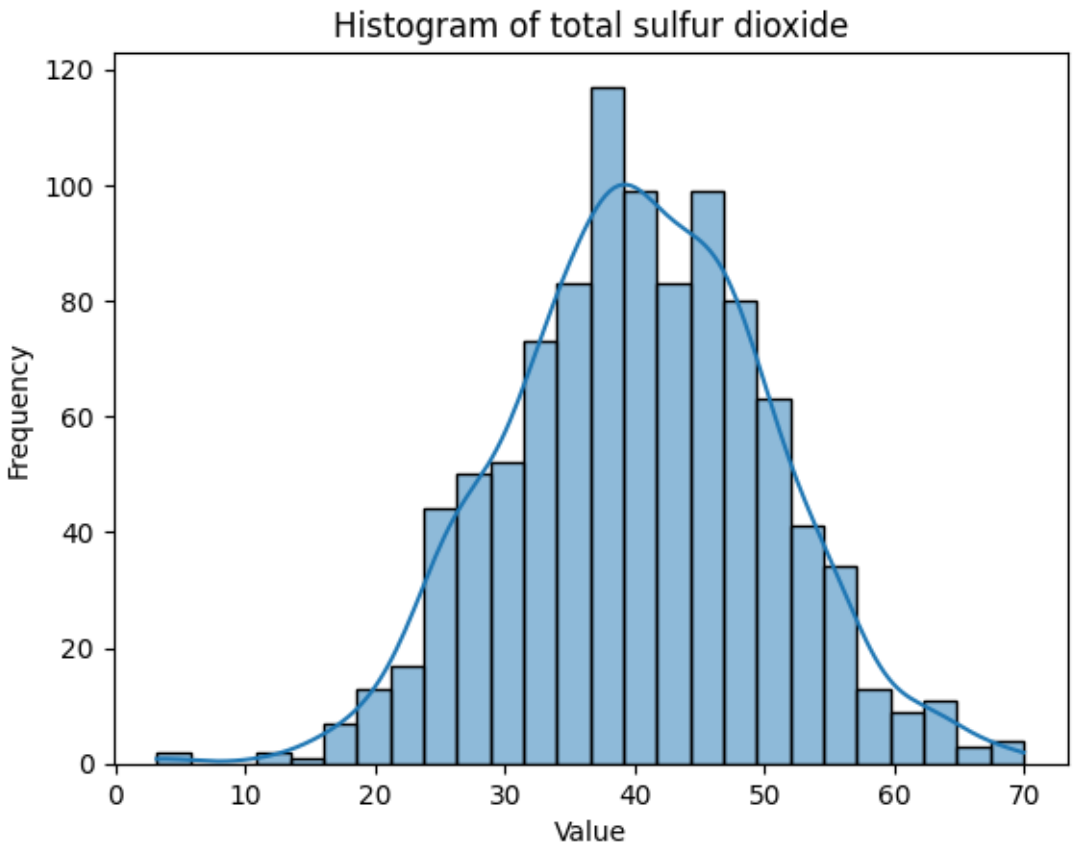
Kolom free sulfur dioxide **tidak terdistribusi normal**




### 6.7 Kolom Total Sulfur Dioxide

[35]:
```
normal_test('total sulfur dioxide')
```

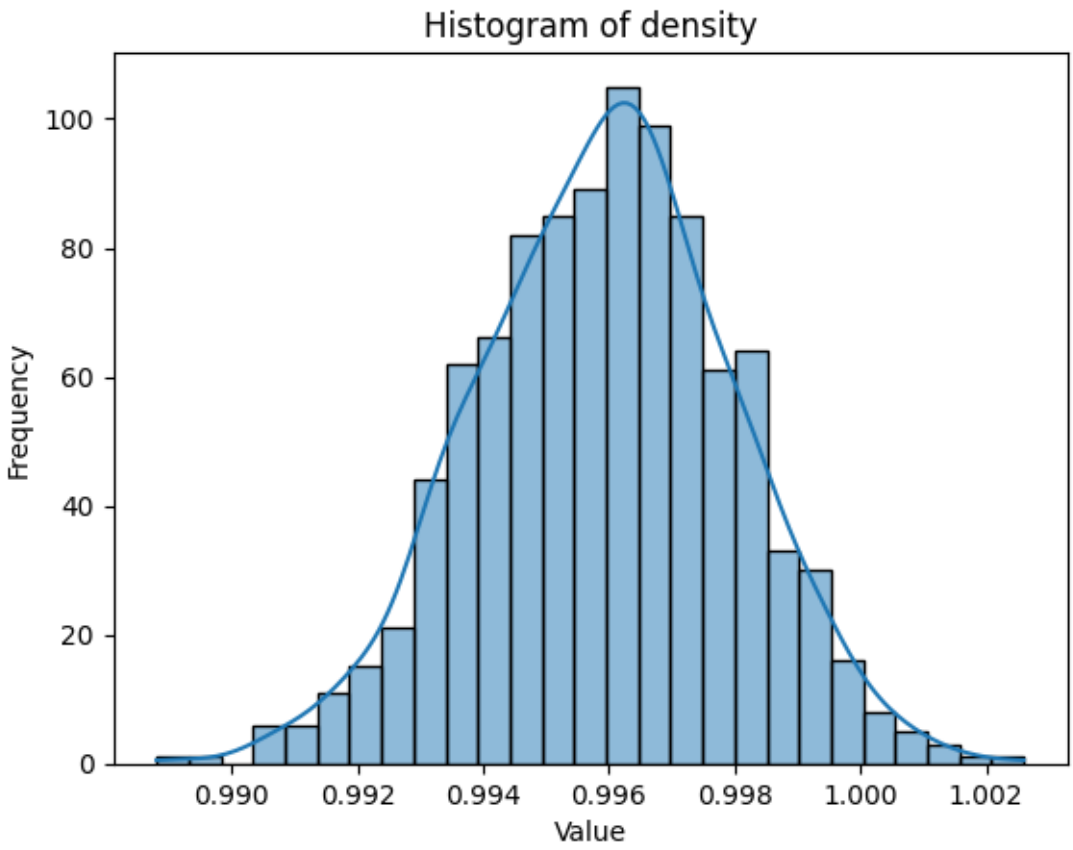
Kolom total sulfur dioxide **terdistribusi normal**




### 6.8 Kolom Density

[36]:
```
normal_test('density')
```

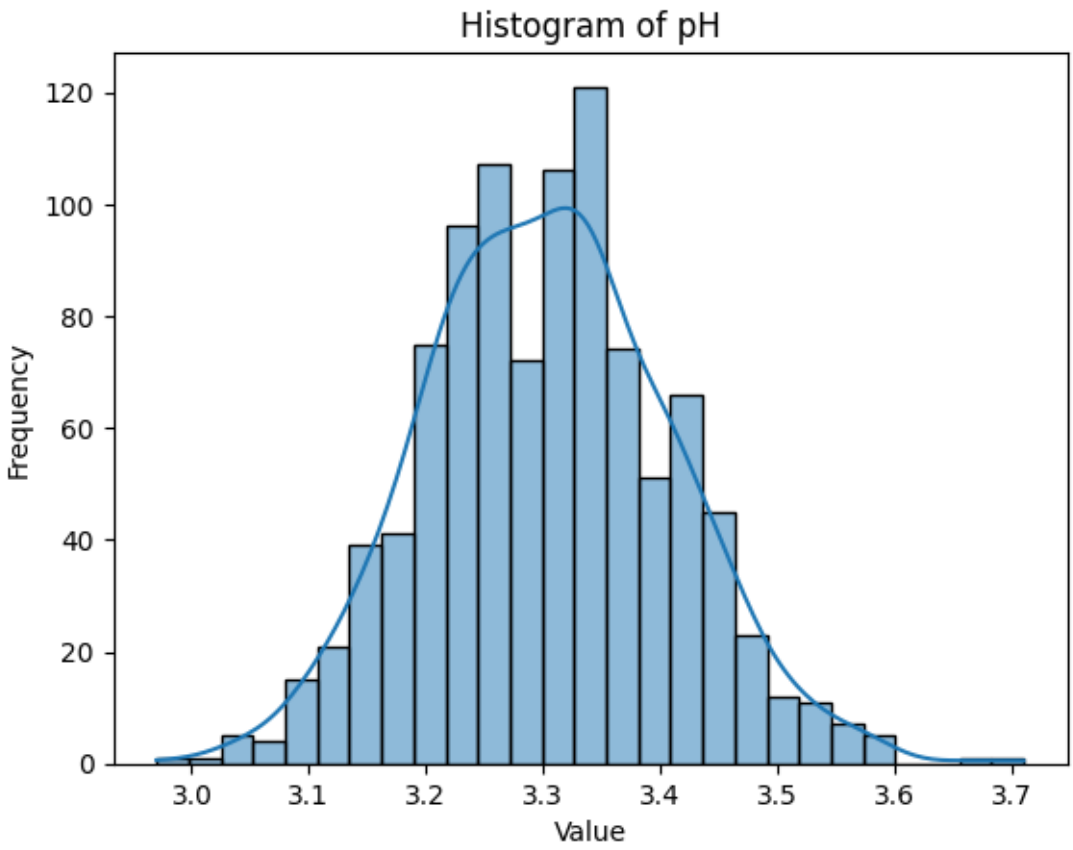
Kolom density **terdistribusi normal**




### 6.9 Kolom pH

[37]:
```
normal_test('pH')
```

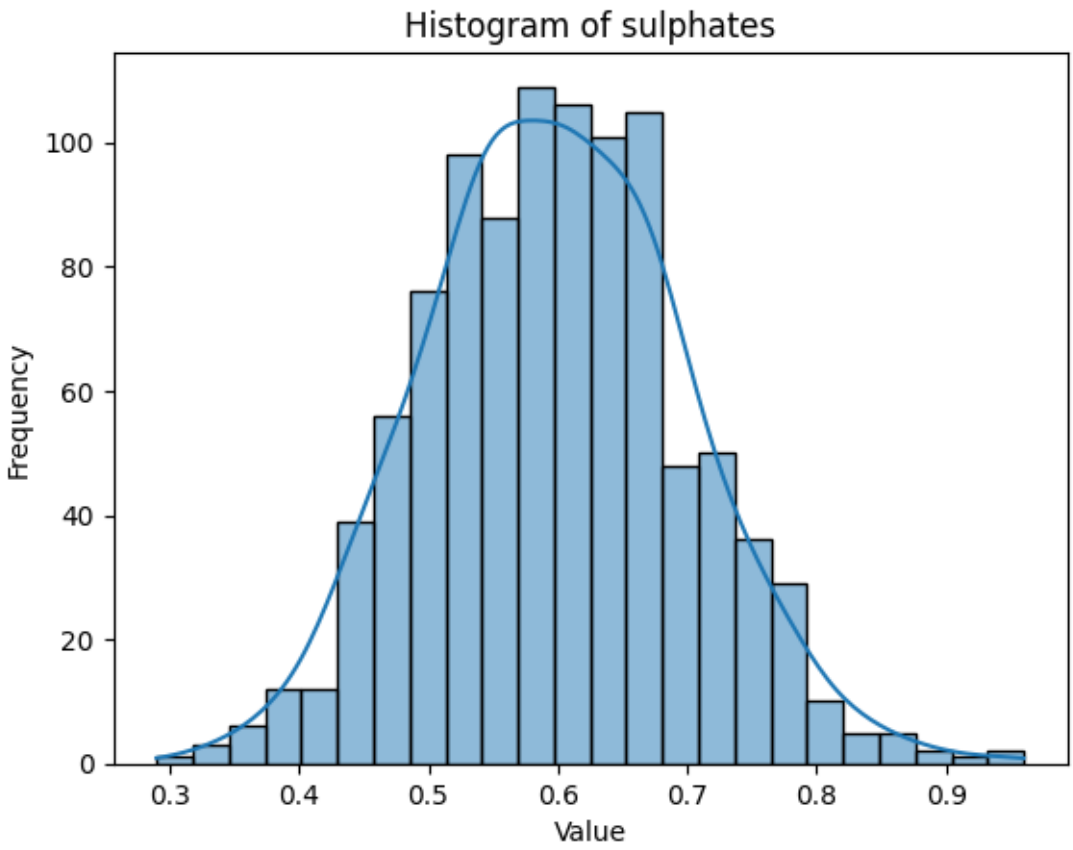
Kolom pH **terdistribusi normal**




### 6.10 Kolom Sulphates

[38]:
```
normal_test('sulphates')
```

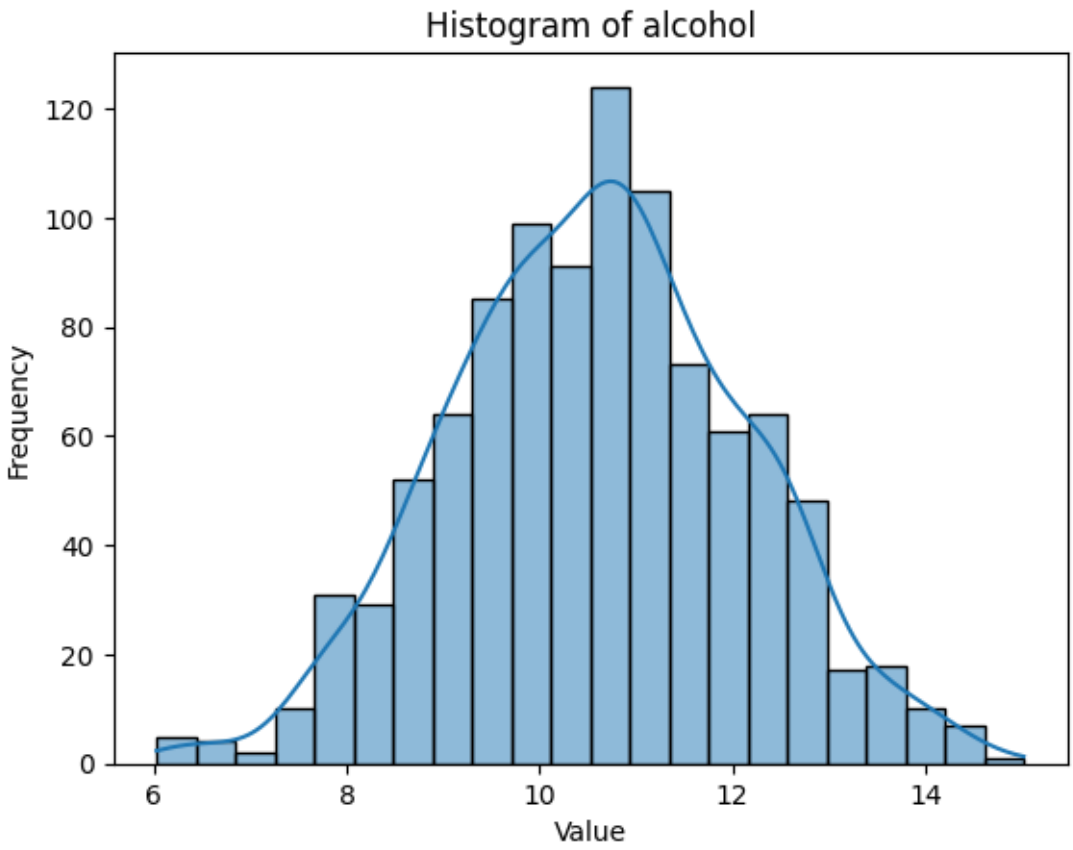
Kolom sulphates **terdistribusi normal**




### 6.11 Kolom Alcohol

[39]:
```
normal_test('alcohol')
```

Kolom alcohol **terdistribusi normal**




# 7 SOAL 4

## 7.1 a. Nilai rata-rata pH di atas 3.29?

**Tentukan Hipotesis null**

$H_0 : \mu = 3.29$

**Tentukan Hipotesis alternatif**

$H_1 : \mu > 3.29$

**Tentukan tingkat signifikan**

$\alpha = 0.05$

**Menggunakan one-tailed test dan perhitungan z**

Nilai z didapatkan dari

$$z_{Hit} = \frac{\bar{x} - \mu}{\sigma}\sqrt{N}$$

Daerah kritis yang diambil = $z > 1.645$

**Perhitungan $p$-value**

$p$-value didapatkan dengan menghitung $P(Z > z_{Hit})$

**Pengambilan Keputusan**

$H_0$ diterima jika $p \geq \alpha$ dan $z_{Hit} \leq z_{Tab}$

$H_0$ ditolak jika $p < \alpha$ atau $z_{Hit} > z_{Tab}$

[40]:
```
# Nilai Kepercayaan
sig = 0.05

# cari nilai z
zHit = (df['pH'].mean() - 3.29)/(df['pH'].std()/np.sqrt(1000))

# cari nilai z dari tabel
zTab = norm.ppf(1-sig)

#cari Pvalue
Pval = norm.sf(zHit)

display(Markdown(f"Nilai p = {round(Pval,3)}"))
display(Markdown(f"Nilai z = {round(zHit,3)}"))
display(Markdown(f"Nilai alpha = {round(sig,3)}"))
display(Markdown(f"Nilai z tabel = {round(zTab,3)}"))

display(Markdown("Kesimpulan"))
if (Pval >= sig and zHit <= zTab):
    display(Markdown("H0 diterima: Rata-rata pH-nya 3.29 ke bawah"))
else:
    display(Markdown("H0 ditolak: Rata-rata pH-nya lebih dari 3.29"))
```

Nilai p = 0.0

Nilai z = 4.104

Nilai alpha = 0.05

Nilai z tabel = 1.645

Kesimpulan

H0 ditolak: Rata-rata pH-nya lebih dari 3.29

## 7.2 b. Nilai rata-rata Residual Sugar tidak sama dengan 2.50?

**Tentukan Hipotesis null**

$H_0 : \mu = 2.5$

**Tentukan Hipotesis alternatif**

$H_1 : \mu \neq 2.5$

**Tentukan tingkat signifikan**

$\alpha = 0.05$

**Menggunakan two-tailed test dan perhitungan z**

Nilai z didapatkan dari

$$z_{Hit} = \frac{\bar{x} - \mu}{\sigma}\sqrt{N}$$

daerah kritis yang diambil $z < -1.96$ atau $z > 1.96$

**Perhitungan $p$-value**

$p$-value didapatkan dari $2P(Z < -z_{Hit})$

**Pengambilan Keputusan**

$H_0$ ditolak jika $p < \alpha$ dan untuk nilai $z$, $z < -1.96$ atau $z > 1.96$

$H_0$ diterima jika $p \geq \alpha$ dan untuk nilai $z$, $-1.96 \leq z \leq 1.96$

[41]:
```
#4.b
# Nilai Kepercayaan = 0.05
sig = 0.05
# Hitung nilai ZHit
zHit = (df['residual sugar'].mean() - 2.5)/(df['residual sugar'].std()/np.
 ↪sqrt(1000))
# Cari nilai Z Tabel
zTab = norm.ppf(1 - sig/2)

#cari p value untuk two tailed
Pval = 2*norm.sf(zHit)

display(Markdown(f"Nilai p = {round(Pval,3)}"))
display(Markdown(f"Nilai z = {round(zHit,3)}"))
display(Markdown(f"Nilai alpha = {round(sig,3)}"))
display(Markdown(f"Nilai z tabel = {round(zTab,3)}"))

# Pengambilan Keputusan
if (Pval<sig and (zHit < -1*(zTab) or zHit > zTab)):
    print("H0 ditolak: Residual Sugar tidak sama dengan 2.5")
else:
    print("H0 diterima: Residual Sugar sama dengan 2.5")
```

Nilai p = 0.032

Nilai z = 2.148

Nilai alpha = 0.05

Nilai z tabel = 1.96

```
H0 ditolak: Residual Sugar tidak sama dengan 2.5
```

## 7.3 c. Nilai rata-rata 150 baris pertama kolom sulphates bukan 0.65?

**Tentukan Hipotesis null**

$H_0 : \mu = 0.65$

**Tentukan Hipotesis alternatif**

$H_1 : \mu \neq 0.65$

**Tentukan tingkat signifikan**

$\alpha = 0.05$

**Menggunakan two-tailed test dan perhitungan z**

Nilai z didapatkan dari

$$z_{Hit} = \frac{\bar{x} - \mu}{\sigma}\sqrt{N}$$

daerah kritis yang diambil $z < -1.96$ atau $z > 1.96$

**Perhitungan $p$-value**

$p$-value didapatkan dari $2P(Z > z_{Hit})$

**Pengambilan Keputusan**

$H_0$ ditolak jika $p < \alpha$ dan untuk nilai $z$, $z < -1.96$ atau $z > 1.96$

$H_0$ diterima jika $p \geq \alpha$ dan untuk nilai $z$, $-1.96 \leq z \leq 1.96$

[42]:
```
# Ambil 150 data pertama
testSulphate = df.head(150).copy()

# Nilai Kepercayaan = 0.05
sig = 0.05

# Hitung nilai ZHit
zHit = (testSulphate['sulphates'].mean() - 0.65)/(testSulphate['sulphates'].
 ↪std()/np.sqrt(150))

# Hitung nilai Z Tabel
zTab = norm.ppf(1-sig/2)

# Cari nilai p
PVal = norm.sf(abs(zHit))*2

display(Markdown(f"Nilai p = {round(PVal,3)}"))
display(Markdown(f"Nilai z = {round(zHit,3)}"))
display(Markdown(f"Nilai alpha = {round(sig,3)}"))
display(Markdown(f"Nilai z tabel = {round(zTab,3)}"))
display(Markdown("Kesimpulan"))

# Pengambilan Keputusan
```

```
if (PVal<sig and (zHit < -zTab or zHit > zTab)):
    print("H0 ditolak: Rata-Rata sulphates tidak sama dengan 0.65")
else:
    print("H0 diterima: Rata-Rata sulphates sama dengan 0.65")
```

```
0.6058666666666667
```

Nilai p = 0.0

Nilai z = -4.965

Nilai alpha = 0.05

Nilai z tabel = 1.96

Kesimpulan

```
H0 ditolak: Rata-Rata sulphates tidak sama dengan 0.65
```

## 7.4 d. Nilai rata-rata total sulfur dioxide di bawah 35?

**Tentukan Hipotesis null**

$H_0 : \mu = 35$

**Tentukan Hipotesis alternatif**

$H_1 : \mu < 35$

**Tentukan tingkat signifikan**

$\alpha = 0.05$

**Menggunakan one-tailed test yang didekati dari kiri dan perhitungan z**

Nilai z didapatkan dari

$$z_{Hit} = \frac{\bar{x} - \mu}{\sigma}\sqrt{N}$$

daerah kritis yang diambil $z < -1.645$

**Perhitungan $p$-value**

$p$-value didapatkan dari $P(Z < z_{Hit})$

**Pengambilan Keputusan**

$H_0$ ditolak jika $p < \alpha$ dan untuk nilai $z$, $z < -1.645$

$H_0$ diterima jika $p \geq \alpha$ dan untuk nilai $z$, $z \geq -1.645$

[43]:
```
# Nilai Kepercayaan = 0.05
sig = 0.05

# Hitung nilai ZHitung
zHit = (df['total sulfur dioxide'].mean() - 35)/(df['total sulfur dioxide'].
 ↪std()/np.sqrt(1000))
```

```
# Hitung nilai Z tabel
zTab = norm.ppf(sig)

# Hitung nilai p
PVal = norm.sf(abs(zHit))

display(Markdown(f"Nilai p = {round(PVal,3)}"))
display(Markdown(f"Nilai z = {round(zHit,3)}"))
display(Markdown(f"Nilai alpha = {round(sig,3)}"))
display(Markdown(f"Nilai z tabel = {round(zTab,3)}"))
display(Markdown("Kesimpulan"))

# Pengambilan Keputusan
if (PVal < sig and zHit < zTab):
    print("H0 ditolak: rata-rata total sulfur dioxide di bawah 35")
else:
    print("H0 diterima: rata-rata total sulfur dioxide di atas 35")
```

Nilai p = 0.0

Nilai z = 16.786

Nilai alpha = 0.05

Nilai z tabel = -1.645

Kesimpulan

`H0 diterima: rata-rata total sulfur dioxide di atas 35`

## 7.5 e. Proporsi nilai total Sulfat Dioxide yang lebih dari 40, adalah tidak sama dengan 50%

**Tentukan Hipotesis null**

$H_0 : p = 0.5$

**Tentukan Hipotesis alternatif**

$H_1 : p \neq 0.05$

**Tentukan tingkat signifikan**

$\alpha = 0.05$

**Menggunakan two-tailed test dan perhitungan z**

Nilai z didapatkan dari

$$z_{Hit} = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0 q_0}{n}}}$$

daerah kritis yang diambil $z_{Hit} < -z_{Tab}$ or $z_{Hit} > z_{Tab}$

[44]:
```
# Pecah data untuk nilai total sulfur dioxide yang lebih dari 40
xMoreThan40 = df.loc[df['total sulfur dioxide'] > 40]
pTopi = len(xMoreThan40)/1000

sig = 0.05
zHit = (pTopi-0.5)/(np.sqrt(0.5**2/1000))
zTab = norm.ppf(1-sig/2)
PVal = scipy.stats.norm.sf(abs(zHit)) * 2

display(Markdown(f"Nilai p = {round(PVal,3)}"))
display(Markdown(f"Nilai z = {round(zHit,3)}"))
display(Markdown(f"Nilai alpha = {round(sig,3)}"))
display(Markdown(f"Nilai z tabel = {round(zTab,3)}"))
display(Markdown("Kesimpulan"))

if (PVal < sig and (zHit < -zTab or zHit> zTab)):
    print("H0 ditolak : p != 0.5")
else:
    print("H0 diterima : p = 0.5")
```

Nilai p = 0.448

Nilai z = 0.759

Nilai alpha = 0.05

Nilai z tabel = 1.96

Kesimpulan

```
H0 diterima : p = 0.5
```

# 8 SOAL 5

## 8.1 a. Data kolom fixed acidity dibagi 2 sama rata: bagian awal dan bagian akhir kolom. Benarkah rata-rata kedua bagian tersebut sama?

**Tentukan Hipotesis null**

$H_0 : \mu_1 - \mu_2 = 0$

**Tentukan Hipotesis alternatif**

$H_1 : \mu_1 - \mu_2 \neq 0$

**Tentukan tingkat signifikan**

$\alpha = 0.05$

**Menggunakan two-tailed test dan perhitungan z**

Nilai z didapatkan dari

$$z_{Hit} = \frac{\bar{x}_1 - \bar{x}_2 - \bar{d}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Daerah Kritis = $z < -1.96$ atau $z > 1.96$

[45]:
```
fixedAcidityAwal = df[0:500].copy()
fixedAcidityAkhir = df[500:1000].copy()
m1 = fixedAcidityAwal['fixed acidity'].mean()
m2 = fixedAcidityAkhir['fixed acidity'].mean()

# H0: m1 - m2 = 0
# H1: m1 - m2 != 0
# Derajat Kepercayaan = 0.05
# Daerah Kritis = z < -1.96 atau z > 1.96

# Computation
sig = 0.05
z = (m1-m2-0)/(np.sqrt((fixedAcidityAwal['fixed acidity'].std()**2/500) +␣
 ↪(fixedAcidityAkhir['fixed acidity'].std()**2/500)))
zTab = norm.ppf(1-sig/2)
PVal = scipy.stats.norm.sf(abs(z))*2

display(Markdown(f"Nilai p = {round(PVal,3)}"))
display(Markdown(f"Nilai z = {round(z,3)}"))
display(Markdown(f"Nilai alpha = {round(sig,3)}"))
display(Markdown(f"Nilai z tabel = {round(zTab,3)}"))
display(Markdown("Kesimpulan"))

if (PVal < sig and (z < -zTab or z > zTab)):
    print("H0 ditolak, bagian awal tidak sama dengan bagian akhir rata2nya")
else:
    print("H0 diterima, bagian awal sama dengan bagian akhir rata2nya")
```

Nilai p = 0.979

Nilai z = 0.026

Nilai alpha = 0.05

Nilai z tabel = 1.96

Kesimpulan

```
H0 diterima, bagian awal sama dengan bagian akhir rata2nya
```

## 8.2 b. Data kolom chlorides dibagi 2 sama rata: bagian awal dan bagian akhir kolom. Benarkah rata-rata bagian awal lebih besar daripada bagian akhir sebesar 0.001?

**Tentukan Hipotesis null**

$H_0 : \mu_1 - \mu_2 = 0.001$

**Tentukan Hipotesis alternatif**

$H_1 : \mu_1 - \mu_2 > 0.001$

**Tentukan tingkat signifikan**

$\alpha = 0.05$

**Menggunakan one-tailed test dan perhitungan z**

Nilai z didapatkan dari

$$z_{Hit} = \frac{\bar{x}_1 - \bar{x}_2 - \bar{d}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Ambil daerah kritis $z > 1.645$

[46]:
```
# ini emang ada 2 ya critical valuenya?
chloridesAwal = df[0:500].copy()
chloridesAkhir = df[500:1000].copy()
m1 = chloridesAwal['chlorides'].mean()
m2 = chloridesAkhir['chlorides'].mean()

# H0: m1 - m2 = 0.001
# H1: m1 - m2 > 0.001
# Derajat Kepercayaan: 0.05
# Derajat Kebebasan = 1000 - 2 = 998
# Critical Value z > 1.645

# Computation
sig = 0.05
zHit = (m1-m2-0.001)/(np.sqrt((chloridesAwal['chlorides'].std()**2/500) +␣
 ↪(fixedAcidityAkhir['chlorides'].std()**2/500)))
zTab = norm.ppf(1-sig)
PVal = norm.sf(zHit) * 2

display(Markdown(f"Nilai p = {round(PVal,3)}"))
display(Markdown(f"Nilai z = {round(zHit,3)}"))
display(Markdown(f"Nilai alpha = {round(sig,3)}"))
display(Markdown(f"Nilai z tabel = {round(zTab,3)}"))
display(Markdown("Kesimpulan"))

if((zHit > zTab) and PVal < 0.05):
    print("H0 ditolak, bagian awal lebih besar dari bagian akhirnya ")
```

```
else:
    print("H0 diterima, bagian awal tidak lebih besar dari bagian akhirnya")
```

Nilai p = 1.36

Nilai z = -0.467

Nilai alpha = 0.05

Nilai z tabel = 1.645

Kesimpulan

```
H0 diterima, bagian awal tidak lebih besar dari bagian akhirnya
```

### 8.3 c. Benarkah rata-rata sampel 25 baris pertama kolom Volatile Acidity sama dengan rata-rata 25 baris pertama kolom Sulphates ?

**Tentukan Hipotesis null**

$H_0 : \mu_{VA} - \mu_{VS} = 0$

**Tentukan Hipotesis alternatif**

$H_1 : \mu_{VA} - \mu_{VS} \neq 0$

**Tentukan tingkat signifikan**

$\alpha = 0.05$

**Menggunakan two tailed test dan perhitungan spooled t**

Nilai $t$ didapatkan dari

$$t = \frac{\bar{x}_1 - \bar{x}_2 - d_0}{s_p^2 \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

dan $s_p^2$

$$s_p^2 = \frac{s_1^2(n_1 - 1) + s_2^2(n_2 - 1)}{n_1 + n_2 - 2}$$

dengan critical value $t < -t_{tab}$ atau $t > t_{tab}$

**Perhitungan $p$-value**

$p$-value didapatkan dari $2P(T > t)$

[47]:
```
splitTwentyFive = df.head(25).copy()
mVA = splitTwentyFive['volatile acidity'].mean()
mS = splitTwentyFive['sulphates'].mean()

# H0: mVA - mVS = 0
# H1: mVA - mVS != 0
# Derajat Kepercayaan: 0.05
# Derajat Kebebasan = 25 + 25 - 2 = 48
# Critical Value t < -2.011 atau t > 2.011
```

```
# Computation
sp = (splitTwentyFive['volatile acidity'].std()**2 * (24) +␣
 ↪splitTwentyFive['sulphates'].std()**2 * (24))/48
t = (mVA - mS - 0)/((np.sqrt(sp)*np.sqrt((1/25 + 1/25))))

display(Markdown(f"t = {round(t,3)}"))
ttab = scipy.stats.t.ppf(q=1-0.05/2, df = 48)
display(Markdown(f"t tabel = {round(ttab,3)}"))

#Cari nilai P
PVal = scipy.stats.t.sf(abs(t), df=48) * 2

display(Markdown(f"Nilai alpha = 0.05"))
display(Markdown(f"P Value = {round(PVal,3)}"))

display(Markdown("Kesimpulan:"))

if (PVal < sig and (t < -ttab or t > ttab)):
    print("H0 ditolak, rata-rata sampel 25 baris pertama kolom Volatile Acidity␣
 ↪tidak sama dengan rata-rata 25 baris pertama kolom Sulphates")
else:
    print("H0 diterima, rata-rata sampel 25 baris pertama kolom Volatile Acidity␣
 ↪sama dengan rata-rata 25 baris pertama kolom Sulphates")
```

t = -2.637

t tabel = 2.011

Nilai alpha = 0.05

P Value = 0.011

Kesimpulan:

H0 ditolak, rata-rata sampel 25 baris pertama kolom Volatile Acidity tidak sama dengan rata-rata 25 baris pertama kolom Sulphates

## 8.4 d. Bagian awal kolom residual sugar memiliki variansi yang sama dengan bagian akhirnya?

**Tentukan Hipotesis null**

$H_0 : v1 = v2$

**Tentukan Hipotesis alternatif**

$H_1 : v1 \neq v2$

**Tentukan tingkat signifikan**

$\alpha = 0.05$

**Menggunakan uji two tailed f test**

Perhitungan nilai $f$

$$f = \frac{s_1^2}{s_2^2}$$

ambil daerah kritis $f < f_{1-\alpha/2}(v_1, v_2)$ dan $f > f_{\alpha/2}(v_1, v_2)$

degan $v_1 = n_1 - 1, v_2 = n_2 - 1$

[48]:
```
# H0 : v1 = v2
# H1 : v1 != v2

rs = df['residual sugar'].copy()
r1 = rs[:500]
r2 = rs[500:1000]

#cari critical region
batas1 = scipy.stats.f.ppf(q=1-0.975, dfn=500-1, dfd=500-1)
# print(batas1)
batas2 = scipy.stats.f.ppf(q=0.975, dfn=500-1, dfd=500-1)
# print(batas2)

#cari f nya
s1 = r1.var()
s2 = r2.var()
f = s1/s2

display(Markdown(f"Nilai f: {round(f,3)}"))
display(Markdown("Kesimpulan:"))
if (f<batas1 or f>batas2):
    print("H0 ditolak : v1 != v2")
else:
    print("H0 diterima : v1 = v2")
```

Nilai f: 0.942

Kesimpulan:

```
H0 diterima : v1 = v2
```

## 8.5 e. Proporsi nilai setengah bagian awal alcohol yang lebih dari 7, adalah lebih besar daripada, proporsi nilai yang sama di setengah bagian akhir alcohol?

**Tentukan Hipotesis null**

$H_0 : p1 = p2$

**Tentukan Hipotesis alternatif**

$H_1 : p1 \neq p2$

**Tentukan tingkat signifikan**

$\alpha = 0.05$

**Two tailed test dengan z untuk testing p1 = p2**

Nilai $z$

$$z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}\hat{q}(1/n_1 + 1/n_2)}}$$

dengan $\hat{p}_1 = \frac{x_1}{n_1}, \hat{p}_2 = \frac{x_2}{n_2}$ dan $\hat{p} = \frac{x_1+x_2}{n_1+n_2}, \hat{q} = 1 - \hat{p}$

daerah kritis: $ z < -z_{tabel}$ or $ z > z_{tabel}$

[49]:
```
# H0 : p1 = p2
# H1 : p1 != p2
x1MoreThan7 = df.head(500).loc[df['alcohol']>7, "alcohol"]
pTopi1 = len(x1MoreThan7)/1000
x2MoreThan7 = df.tail(500).loc[df['alcohol']>7, "alcohol"]
pTopi2 = len(x2MoreThan7)/1000

sig = 0.05
#hitung p
p = (len(x1MoreThan7)+len(x2MoreThan7))/(1000 + 1000)
# display(p)

# z tabel
zTab = norm.ppf(1-sig/2)
display(Markdown(f"z tabel: {round(zTab,3)}"))

#hitung z
z = (pTopi1 - pTopi2)/np.sqrt(p*(1-p)*(1/500+1/500))
display(Markdown(f"z: {z}"))

PVal = norm.sf(z)

display(Markdown(f"Nilai p = {round(PVal,3)}"))

if (PVal<0.05 and (z < -zTab or z > zTab)):
    print("H0 ditolak : p1 = p2")
else:
    print("H1 diterima : p1 != p2")
```

z tabel: 1.96

z: 0.0

Nilai p = 0.5

```
H1 diterima : p1 != p2
```