In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df= pd.read_csv('just data.csv')
df
```

Out[2]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.00100	3
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.99400	3
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.99510	3
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3
...	...	...	...	...	...	...	...	...	
4893	6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3
4894	6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3
4895	6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2
4896	5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3
4897	6.0	0.21	0.38	0.8	0.020	22.0	98.0	0.98941	3

4898 rows × 12 columns

In [3]:
```
df.describe()
```

Out[3]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free
count	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.
mean	6.854788	0.278241	0.334192	6.391415	0.045772	35.
std	0.843868	0.100795	0.121020	5.072058	0.021848	17.
min	3.800000	0.080000	0.000000	0.600000	0.009000	2.
25%	6.300000	0.210000	0.270000	1.700000	0.036000	23.
50%	6.800000	0.260000	0.320000	5.200000	0.043000	34.
75%	7.300000	0.320000	0.390000	9.900000	0.050000	46.
max	14.200000	1.100000	1.660000	65.800000	0.346000	289.

In [4]:
```
df.head()
```

Out[4]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4898 entries, 0 to 4897
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         4898 non-null   float64
 1   volatile acidity      4898 non-null   float64
 2   citric acid           4898 non-null   float64
 3   residual sugar        4898 non-null   float64
 4   chlorides             4898 non-null   float64
 5   free sulfur dioxide   4898 non-null   float64
 6   total sulfur dioxide  4898 non-null   float64
 7   density               4898 non-null   float64
 8   pH                    4898 non-null   float64
 9   sulphates             4898 non-null   float64
 10  alcohol               4898 non-null   float64
 11  quality               4898 non-null   int64  
dtypes: float64(11), int64(1)
memory usage: 459.3 KB
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

In [7]:
```
df.columns
```

Out[7]:
```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual suga
r',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'densit
y',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

In [8]:
```
df.count
```

Out[8]:
```
<bound method DataFrame.count of       fixed acidity  volatile acidity  cit
ric acid  residual sugar  chlorides  \
0               7.0              0.27         0.36            20.7      0.0
45
1               6.3              0.30         0.34             1.6      0.0
49
2               8.1              0.28         0.40             6.9      0.0
50
3               7.2              0.23         0.32             8.5      0.0
58
4               7.2              0.23         0.32             8.5      0.0
58
...             ...               ...          ...             ...
...
4893            6.2              0.21         0.29             1.6      0.0
39
4894            6.6              0.32         0.36             8.0      0.0
47
4895            6.5              0.24         0.19             1.2      0.0
41
4896            5.5              0.29         0.30             1.1      0.0
22
4897            6.0              0.21         0.38             0.8      0.0
20

      free sulfur dioxide  total sulfur dioxide  density    pH  sulphates
\
0                    45.0                 170.0  1.00100  3.00       0.45
1                    14.0                 132.0  0.99400  3.30       0.49
2                    30.0                  97.0  0.99510  3.26       0.44
3                    47.0                 186.0  0.99560  3.19       0.40
4                    47.0                 186.0  0.99560  3.19       0.40
...                   ...                   ...      ...   ...        ...
4893                 24.0                  92.0  0.99114  3.27       0.50
4894                 57.0                 168.0  0.99490  3.15       0.46
4895                 30.0                 111.0  0.99254  2.99       0.46
4896                 20.0                 110.0  0.98869  3.34       0.38
4897                 22.0                  98.0  0.98941  3.26       0.32

      alcohol  quality
0         8.8        6
1         9.5        6
2        10.1        6
3         9.9        6
4         9.9        6
...       ...      ...
4893     11.2        6
4894      9.6        5
4895      9.4        6
4896     12.8        7
4897     11.8        6

[4898 rows x 12 columns]>
```

In [9]:
```
df.tail()
```

Out[9]:

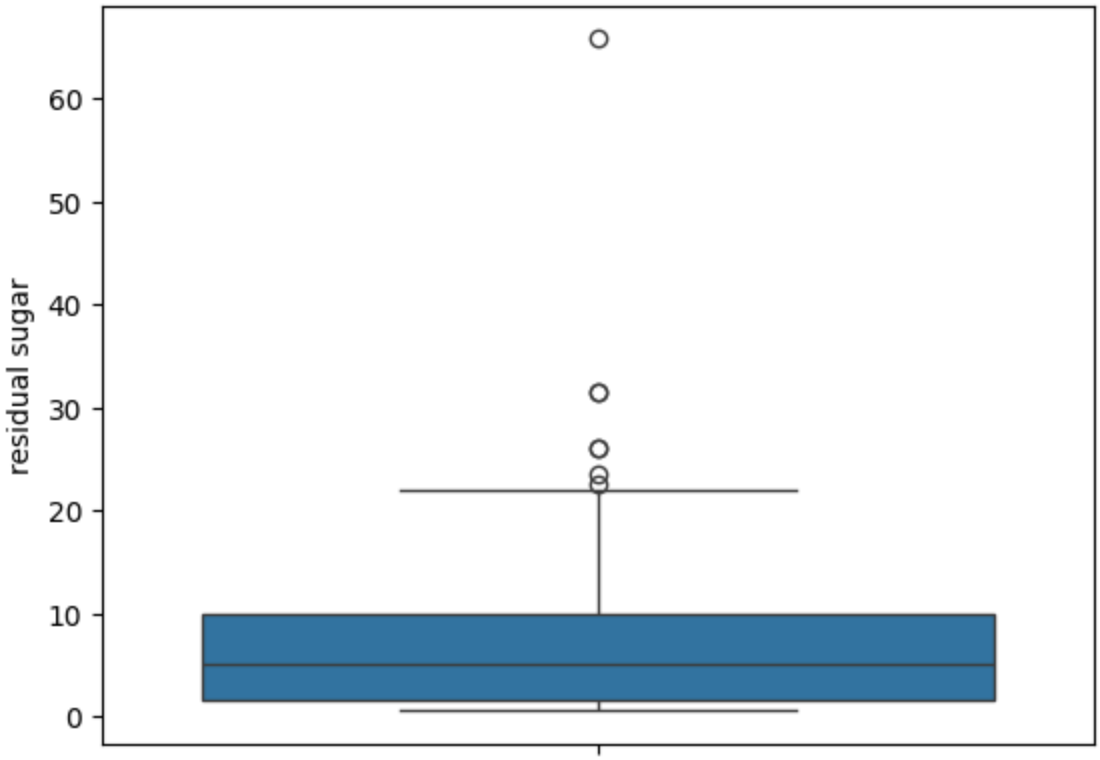
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	
4893	6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3
4894	6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3
4895	6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2
4896	5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3
4897	6.0	0.21	0.38	0.8	0.020	22.0	98.0	0.98941	3

In [10]:
```
df.describe
```

Out[10]:
```
<bound method NDFrame.describe of       fixed acidity  volatile acidity  ci
tric acid  residual sugar  chlorides  \
0                7.0              0.27         0.36            20.7      0.0
45
1                6.3              0.30         0.34             1.6      0.0
49
2                8.1              0.28         0.40             6.9      0.0
50
3                7.2              0.23         0.32             8.5      0.0
58
4                7.2              0.23         0.32             8.5      0.0
58
...              ...               ...          ...             ...
...
4893             6.2              0.21         0.29             1.6      0.0
39
4894             6.6              0.32         0.36             8.0      0.0
47
4895             6.5              0.24         0.19             1.2      0.0
41
4896             5.5              0.29         0.30             1.1      0.0
22
4897             6.0              0.21         0.38             0.8      0.0
20

      free sulfur dioxide  total sulfur dioxide  density    pH  sulphates
\
0                    45.0                 170.0  1.00100  3.00       0.45
1                    14.0                 132.0  0.99400  3.30       0.49
2                    30.0                  97.0  0.99510  3.26       0.44
3                    47.0                 186.0  0.99560  3.19       0.40
4                    47.0                 186.0  0.99560  3.19       0.40
...                   ...                   ...      ...   ...        ...
4893                 24.0                  92.0  0.99114  3.27       0.50
4894                 57.0                 168.0  0.99490  3.15       0.46
4895                 30.0                 111.0  0.99254  2.99       0.46
4896                 20.0                 110.0  0.98869  3.34       0.38
4897                 22.0                  98.0  0.98941  3.26       0.32

      alcohol  quality
0         8.8        6
1         9.5        6
2        10.1        6
3         9.9        6
4         9.9        6
...       ...      ...
4893     11.2        6
4894      9.6        5
4895      9.4        6
4896     12.8        7
4897     11.8        6

[4898 rows x 12 columns]>
```

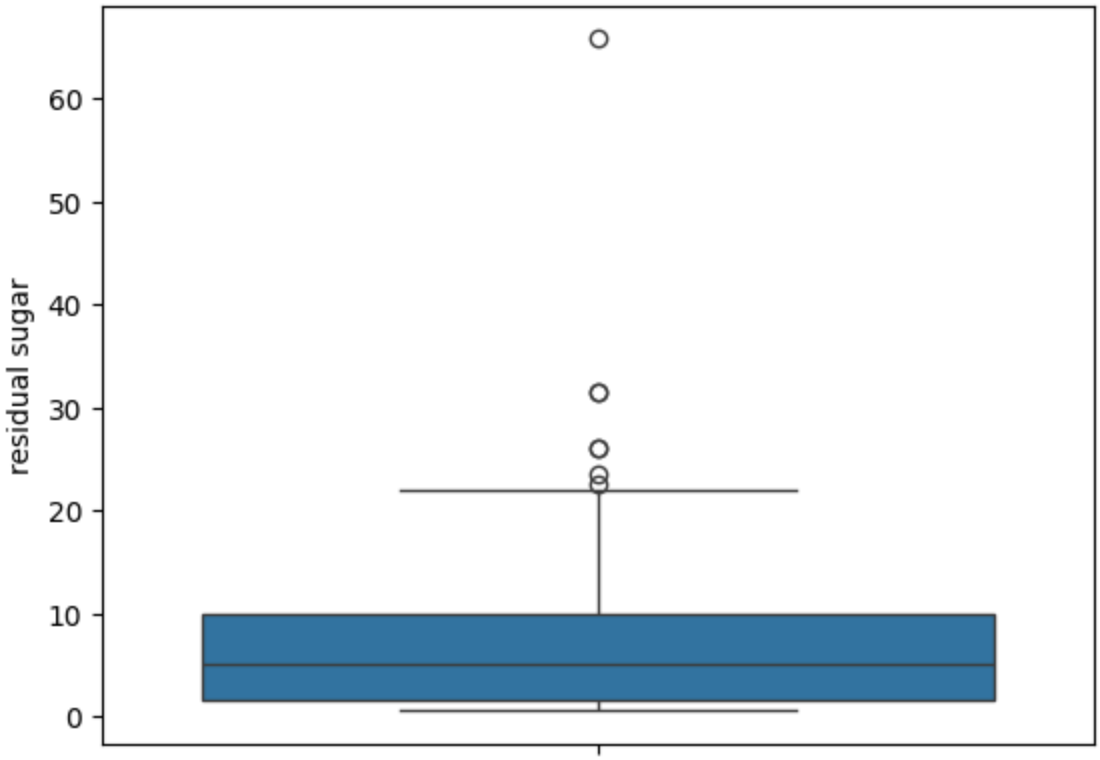
In [11]:
```
sns.boxplot(df['residual sugar'])
```

Out[11]: <Axes: ylabel='residual sugar'>



In [12]:
```
sns.boxplot(df['residual sugar'])
```

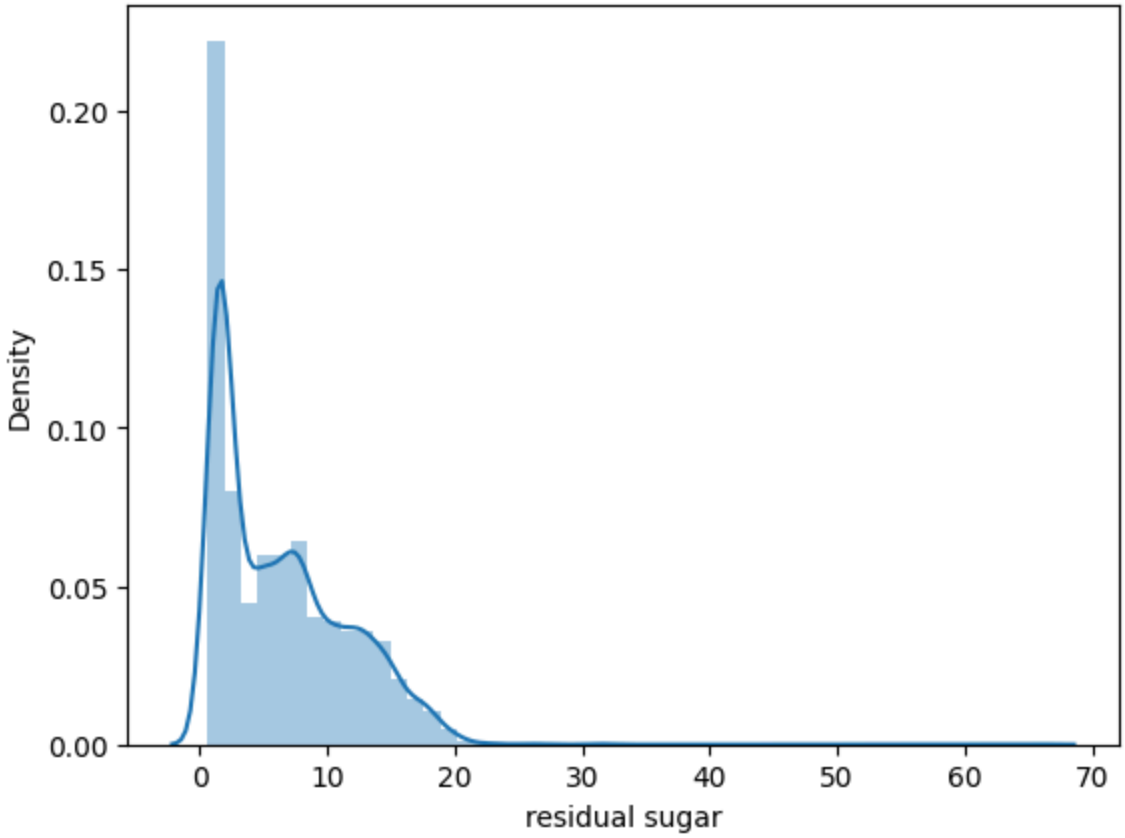
Out[12]: <Axes: ylabel='residual sugar'>



In [19]:
```
sns.distplot(df['residual sugar'])
```

```
C:\Users\shaw3\AppData\Local\Temp\ipykernel_10188\107078400.py:1: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['residual sugar'])
```

Out[19]: <Axes: xlabel='residual sugar', ylabel='Density'>



In [21]:
```
# Select only numerical columns
num_cols = df.select_dtypes(include=['float64', 'int64']).columns
num_cols
```

Out[21]:
```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual suga
r',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'densit
y',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

1st method

z-score method

In [23]:
```
# find the limits
upper_limit= df['residual sugar'].mean() + 3*df['residual sugar'].std()
lower_limit= df['residual sugar'].mean() - 3*df['residual sugar'].std()
print(upper_limit)
print(lower_limit)
```

```
21.607588215254115
-8.824758488835169
```

find the outliers

In [27]:
```
df.loc[(df['residual sugar']> upper_limit) |  (df['residual sugar']< lower_l
```

Out[27]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	
182	6.8	0.280	0.40	22.00	0.048	48.0	167.0	1.00100	2
191	6.8	0.280	0.40	22.00	0.048	48.0	167.0	1.00100	2
1608	6.9	0.270	0.49	23.50	0.057	59.0	235.0	1.00240	2
1653	7.9	0.330	0.28	31.60	0.053	35.0	176.0	1.01030	3
1663	7.9	0.330	0.28	31.60	0.053	35.0	176.0	1.01030	3
2781	7.8	0.965	0.60	65.80	0.074	8.0	160.0	1.03898	3
3619	6.8	0.450	0.28	26.05	0.031	27.0	122.0	1.00295	3
3623	6.8	0.450	0.28	26.05	0.031	27.0	122.0	1.00295	3
4480	5.9	0.220	0.45	22.60	0.120	55.0	122.0	0.99636	3

trimming - detect the outlier data

In [30]:
```
new_df= df.loc[(df['residual sugar']< upper_limit) &  (df['residual sugar']>
new_df
```

Out[30]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.00100	3
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.99400	3
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.99510	3
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3
...	...	...	...	...	...	...	...	...	
4893	6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3
4894	6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3
4895	6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2
4896	5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3
4897	6.0	0.21	0.38	0.8	0.020	22.0	98.0	0.98941	3

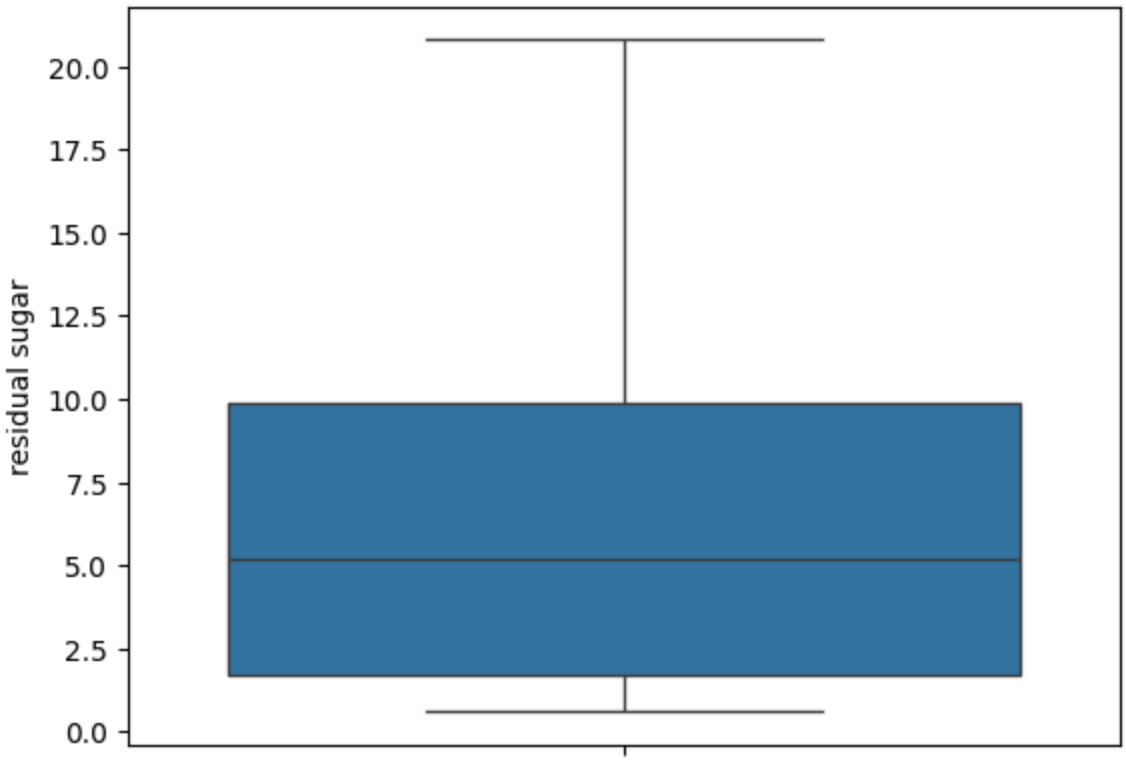
4889 rows × 12 columns

In [33]:
```
print('before remove outliers', len(df))
print('after removing outliers', len(new_df))
print('outliers', len(df)- len(new_df))
```

```
before remove outliers 4898
after removing outliers 4889
outliers 9
```

In [34]:
```
sns.boxplot(new_df['residual sugar'])
```

Out[34]:
```
<Axes: ylabel='residual sugar'>
```



In [35]:
```
new_df
```

Out[35]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.00100	3
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.99400	3
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.99510	3
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3
...	...	...	...	...	...	...	...	...	
4893	6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3
4894	6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3
4895	6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2
4896	5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3
4897	6.0	0.21	0.38	0.8	0.020	22.0	98.0	0.98941	3

4889 rows × 12 columns

In [36]:
```
df
```

Out[36]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.00100	3
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.99400	3
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.99510	3
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3
...	...	...	...	...	...	...	...	...	
4893	6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3
4894	6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3
4895	6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2
4896	5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3
4897	6.0	0.21	0.38	0.8	0.020	22.0	98.0	0.98941	3

4898 rows × 12 columns

2nd method

IQR METHOD

In [37]:
```
q1= df['residual sugar'].quantile(0.25) # 25%
q3= df['residual sugar'].quantile(0.75) #75%
iqr= q3- q1
```

In [38]:
```
q1, q3, iqr
```

Out[38]:
```
(np.float64(1.7), np.float64(9.9), np.float64(8.200000000000001))
```

In [40]:
```
upper_limit= q3 + (1.5* iqr)
lower_limit= q1 - (1.5*iqr)
```

In [42]:
```
upper_limit
#lower_limit
```

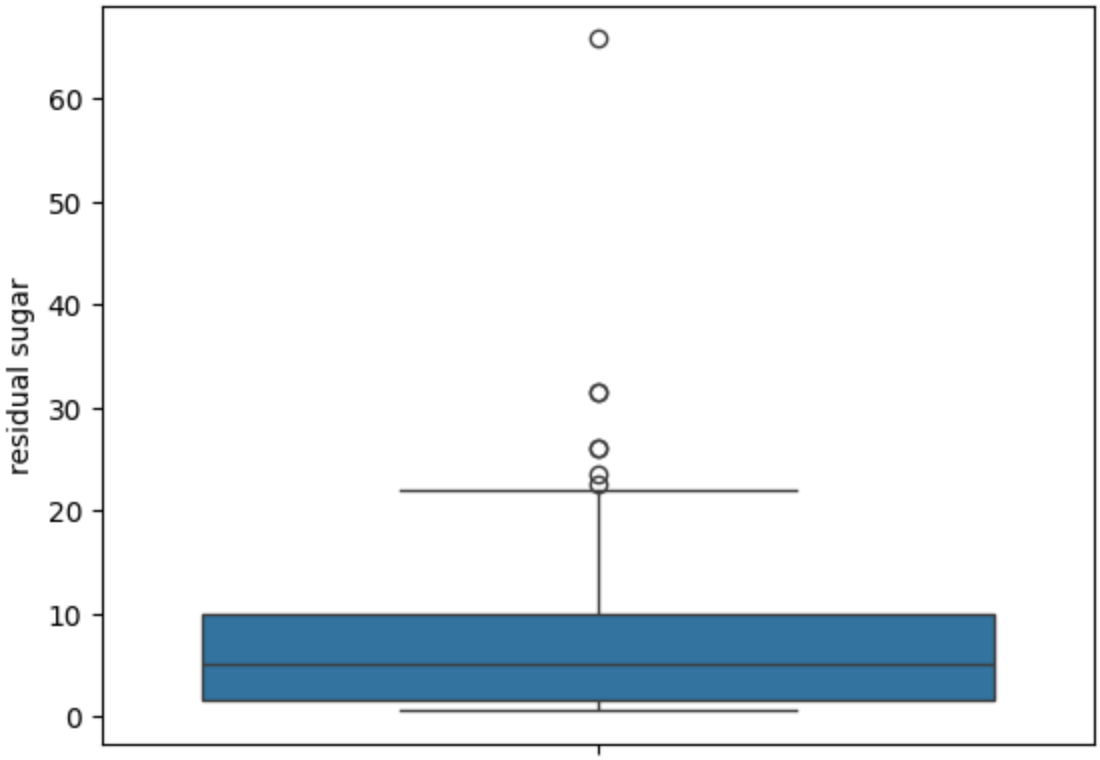
Out[42]:
```
np.float64(22.200000000000003)
```

In [43]:
```
lower_limit
```

Out[43]:
```
np.float64(-10.600000000000001)
```

In [44]:
```
sns.boxplot(df['residual sugar'])
```

Out[44]: <Axes: ylabel='residual sugar'>



In [46]:
```
new1_df= df.loc[(df['residual sugar']< upper_limit) &  (df['residual sugar']
new1_df
```

Out[46]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.00100	3
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.99400	3
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.99510	3
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3
...	...	...	...	...	...	...	...	...	
4893	6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3
4894	6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3
4895	6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2
4896	5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3
4897	6.0	0.21	0.38	0.8	0.020	22.0	98.0	0.98941	3

4891 rows × 12 columns

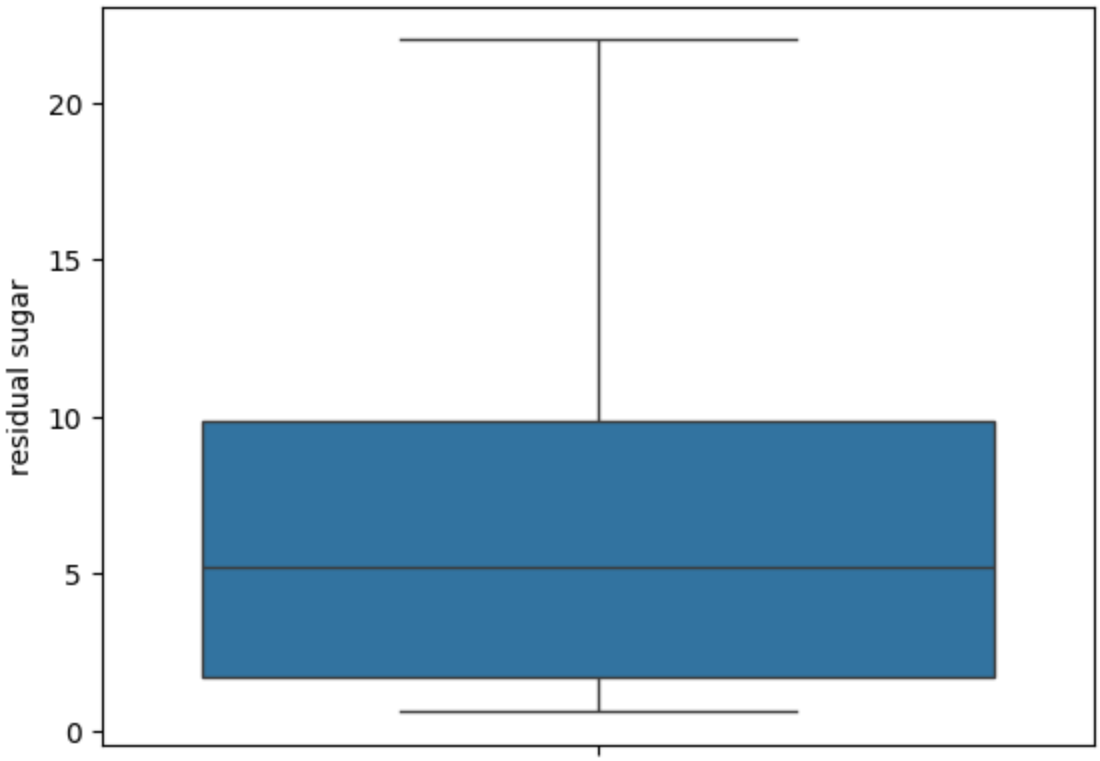
In [47]:
```
print('before remove outliers', len(df))
print('after removing outliers', len(new1_df))
```

```
print('outliers', len(df)- len(new1_df))
```

```
before remove outliers 4898
after removing outliers 4891
outliers 7
```

In [49]:
```
sns.boxplot(new1_df['residual sugar'])
```

Out[49]: <Axes: ylabel='residual sugar'>



3rd method

parcentile method

In [50]:
```
upper_limit= df['residual sugar'].quantile(0.99) # 99% for check upper limit
lower_limit= df['residual sugar'].quantile(0.01) # 1% for check lower limit
```

In [53]:
```
upper_limit
```

Out[53]: np.float64(18.8)

In [54]:
```
lower_limit
```

Out[54]: np.float64(0.9)

In [55]:
```
df.loc[(df['residual sugar']> upper_limit) |  (df['residual sugar']< lower_l
```

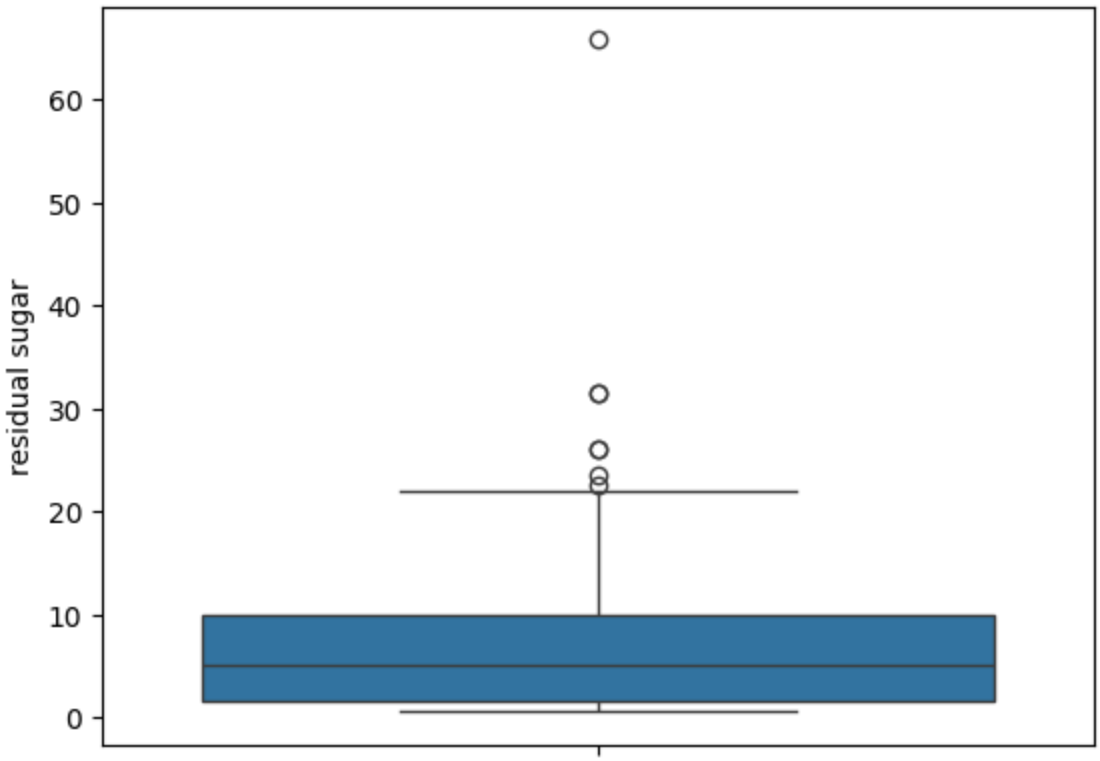
Out[55]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	
0	7.0	0.270	0.36	20.70	0.045	45.0	170.0	1.00100	3
7	7.0	0.270	0.36	20.70	0.045	45.0	170.0	1.00100	3
14	8.3	0.420	0.62	19.25	0.040	41.0	172.0	1.00020	2
103	7.5	0.305	0.40	18.90	0.059	44.0	170.0	1.00000	2
172	7.6	0.480	0.37	0.80	0.037	4.0	100.0	0.99020	3
...	...	...	...	...	...	...	...	...	
4694	6.9	0.190	0.31	19.25	0.043	38.0	167.0	0.99954	2
4778	5.8	0.315	0.19	19.40	0.031	28.0	106.0	0.99704	2
4779	6.0	0.590	0.00	0.80	0.037	30.0	95.0	0.99032	3
4877	5.9	0.540	0.00	0.80	0.032	12.0	82.0	0.99286	3
4897	6.0	0.210	0.38	0.80	0.020	22.0	98.0	0.98941	3

81 rows × 12 columns

In [56]:
```
sns.boxplot(df['residual sugar'])
```

Out[56]:
```
<Axes: ylabel='residual sugar'>
```



In [60]:
```
new2_df= df.loc[(df['residual sugar']<= upper_limit) &  (df['residual sugar'
new2_df
```

Out[60]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.99400	3
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.99510	3
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3
5	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.99510	3
...	...	...	...	...	...	...	...	...	
4892	6.5	0.23	0.38	1.3	0.032	29.0	112.0	0.99298	3
4893	6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3
4894	6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3
4895	6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2
4896	5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3

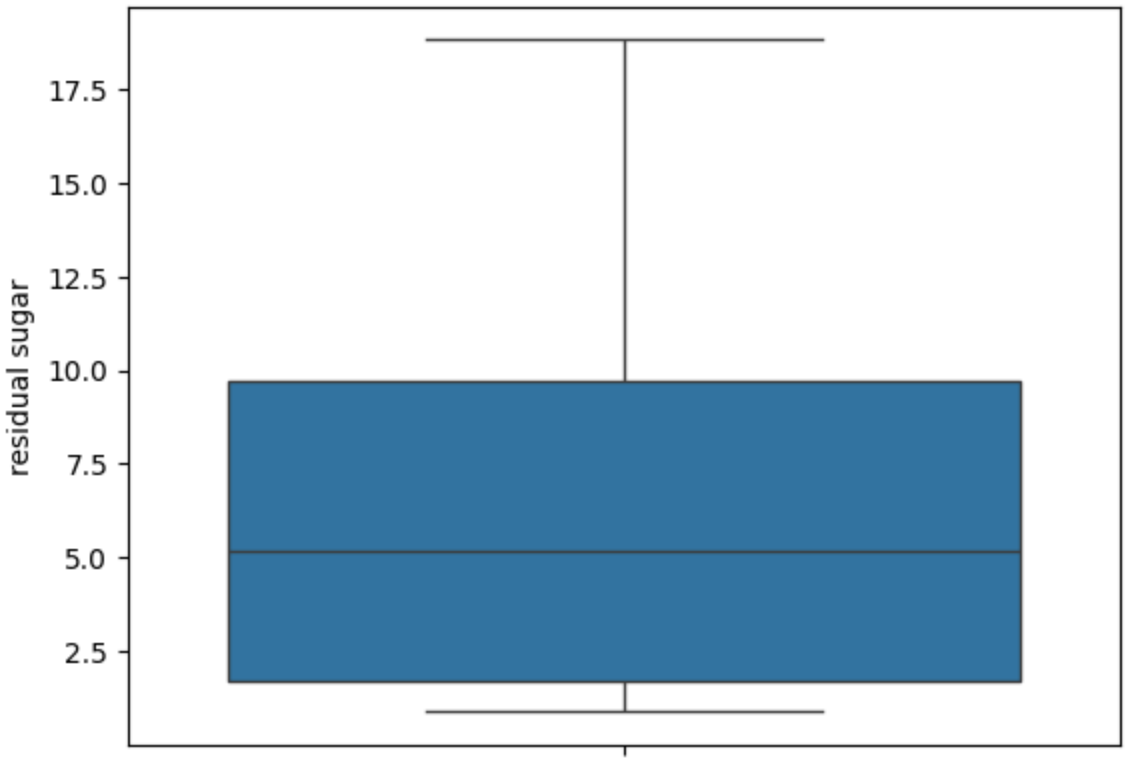
4817 rows × 12 columns

In [61]:
```
print('before remove outliers', len(df))
print('after removing outliers', len(new2_df))
print('outliers', len(df)- len(new2_df))
```

```
before remove outliers 4898
after removing outliers 4817
outliers 81
```

In [63]:
```
sns.boxplot(new2_df['residual sugar'])
```

Out[63]:
```
<Axes: ylabel='residual sugar'>
```



special method

capping method

In [13]:
```
upper_limit= df['residual sugar'].mean() + 3*df['residual sugar'].std()
lower_limit= df['residual sugar'].mean()-  3*df['residual sugar'].std()
```

In [16]:
```
df['residual sugar']= np.where(
    df['residual sugar']> upper_limit,
    upper_limit,
    np.where(
        df['residual sugar']< lower_limit,
        lower_limit,
        df['residual sugar']
    )
)
```

In [18]:
```
df['residual sugar'].describe()
```

Out[18]:
```
count    4898.000000
mean        6.375749
std         4.979963
min         0.600000
25%         1.700000
50%         5.200000
75%         9.900000
max        21.607588
Name: residual sugar, dtype: float64
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: