# 1. Introduction to Statistics and Descriptive Statistics

### Descriptive statistics summarize and organize characteristics of a dataset. It includes:

1. Central Tendency: Measures that represent the center of a dataset (mean, median, mode).
2. Dispersion: Describes the spread or variability (range, variance, standard deviation).
3. Outliers: Data points significantly different from others.
4. Symmetry: Describes the shape (skewness, kurtosis).

# 2. Measures of Central Tendency

5. Mean: The average of all values.
6. Median: The middle value of sorted data.
7. Mode: The most frequently occurring value(s).

In [1]:

```
x=[8,10,7,7,15,4]
```

In [2]:

```
import numpy as np
import statistics as st
x = [1, 2, 3, 4, 5, 5, 6, 8, 9]

# Mean
mean = np.mean(x)

# Median
median = np.median(x)

# Mode
mode = st.mode(x)
```

```
In [3]: mean
```

```
Out[3]: 4.777777777777778
```

```
In [4]: median
```

```
Out[4]: 5.0
```

```
In [5]: mode
```

```
Out[5]: 5
```

In [6]:

```
import seaborn as sns

sns.get_dataset_names()
```

```
Out[6]: ['anagrams',
         'anscombe',
         'attention',
         'brain_networks',
         'car_crashes',
         'diamonds',
         'dots',
         'dowjones',
         'exercise',
         'flights',
         'fmri',
         'geyser',
         'glue',
         'healthexp',
         'iris',
         'mpg',
         'penguins',
         'planets',
         'seaice',
         'taxis',
         'tips',
         'titanic',
         'anagrams',
         'anagrams',
         'anscombe',
         'anscombe',
         'attention',
         'attention',
         'brain_networks',
         'brain_networks',
         'car_crashes',
         'car_crashes',
         'diamonds',
         'diamonds',
         'dots',
         'dots',
         'dowjones',
         'dowjones',
         'exercise',
         'exercise',
         'flights',
         'flights',
         'fmri',
'fmri',
'geyser',
'geyser',
'glue',
'glue',
'healthexp',
'healthexp',
'iris',
'iris',
'mpg',
'mpg',
'penguins',
'penguins',
'planets',
'planets',
'seaice',
'seaice',
'taxis',
'taxis',
'tips',
'tips',
'titanic',
'titanic',
'anagrams',
'anscombe',
'attention',
'brain_networks',
'car_crashes',
'diamonds',
'dots',
'dowjones',
'exercise',
'flights',
'fmri',
'geyser',
'glue',
'healthexp',
'iris',
'mpg',
'penguins',
'planets',
'seaice',
'taxis',
 'tips',
 'titanic']
```

In [7]:

```
data=sns.load_dataset("tips")
data
```

Out[7]:

| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| **1** | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| **2** | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 |
| **3** | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 |
| **4** | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **239** | 29.03 | 5.92 | Male | No | Sat | Dinner | 3 |
| **240** | 27.18 | 2.00 | Female | Yes | Sat | Dinner | 2 |
| **241** | 22.67 | 2.00 | Male | Yes | Sat | Dinner | 2 |
| **242** | 17.82 | 1.75 | Male | No | Sat | Dinner | 2 |
| **243** | 18.78 | 3.00 | Female | No | Thur | Dinner | 2 |

244 rows × 7 columns

In [8]:

```
data["tip"].median()
```

Out[8]:

```
2.9
```

In [9]:

```
st.median(data["total_bill"])
```

Out[9]:

```
17.795
```

In [10]:

```
data.describe()
```

Out[10]:

| | total_bill | tip | size |
|---|---|---|---|
| **count** | 244.000000 | 244.000000 | 244.000000 |
| **mean** | 19.785943 | 2.998279 | 2.569672 |
| **std** | 8.902412 | 1.383638 | 0.951100 |
| **min** | 3.070000 | 1.000000 | 1.000000 |
| **25%** | 13.347500 | 2.000000 | 2.000000 |
| **50%** | 17.795000 | 2.900000 | 2.000000 |
| **75%** | 24.127500 | 3.562500 | 3.000000 |
| **max** | 50.810000 | 10.000000 | 6.000000 |

In [ ]:

```
```

# 3. Measures of Dispersion

1. Range: Difference between the maximum and minimum values.
2. Variance: Average of squared deviations from the mean.
3. Standard Deviation: Square root of variance, indicating the spread of data around the mean.

In [11]:

```
# Range

x = [1, 2, 3, 4, 5, 5, 6, 8, 9]

range_val = np.ptp(x)

range_val
```

Out[11]:

```
8
```

In [12]:

```
# Variance
variance = np.var(data)

variance
```

```
C:\Users\HP\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3721: FutureWarning: Dropping of nuisance
columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will rais
e TypeError.  Select only valid columns before calling the reduction.
  return var(axis=axis, dtype=dtype, out=out, ddof=ddof, **kwargs)
```

Out[12]:

```
total_bill    78.928131
tip            1.906609
size           0.900883
dtype: float64
```

In [13]:

```
# Standard Deviation
std_dev = np.std(data)

std_dev
```

```
C:\Users\HP\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3579: FutureWarning: Dropping of nuisance
columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will rais
e TypeError.  Select only valid columns before calling the reduction.
  return std(axis=axis, dtype=dtype, out=out, ddof=ddof, **kwargs)
```

Out[13]:

```
total_bill    8.884151
tip           1.380800
size          0.949149
dtype: float64
```

```
In [14]: data[["tip","size"]]
```

```
Out[14]:
          tip  size
      0  1.01     2
      1  1.66     3
      2  3.50     3
      3  3.31     2
      4  3.61     4
    ...   ...   ...
    239  5.92     3
    240  2.00     2
    241  2.00     2
    242  1.75     2
    243  3.00     2

244 rows × 2 columns
```

```
In [15]: np.std(data["tip"])
```

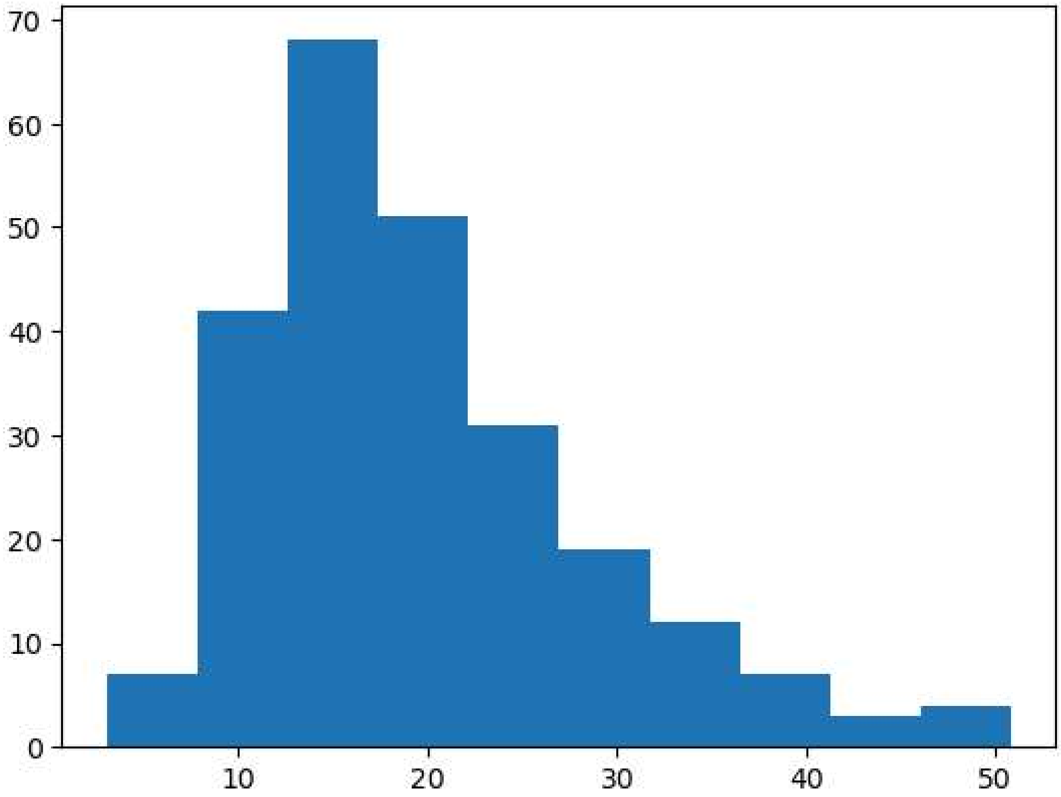
```
Out[15]: 1.3807999538298958
```

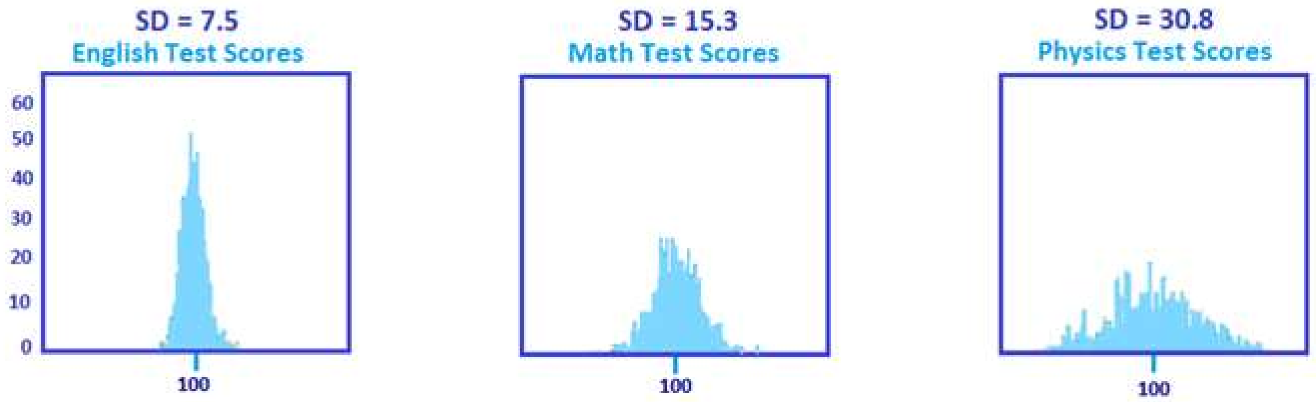
```
In [16]: import matplotlib.pyplot as plt
```

plt.hist(data["tip"])

```
In [17]: plt.hist(data["total_bill"])
```

```
Out[17]: (array([ 7., 42., 68., 51., 31., 19., 12.,  7.,  3.,  4.]),
          array([ 3.07 ,  7.844, 12.618, 17.392, 22.166, 26.94 , 31.714, 36.488,
                 41.262, 46.036, 50.81 ]),
          <BarContainer object of 10 artists>)
```





## 4. Handling Outliers

Outliers can skew data analysis. We often use:

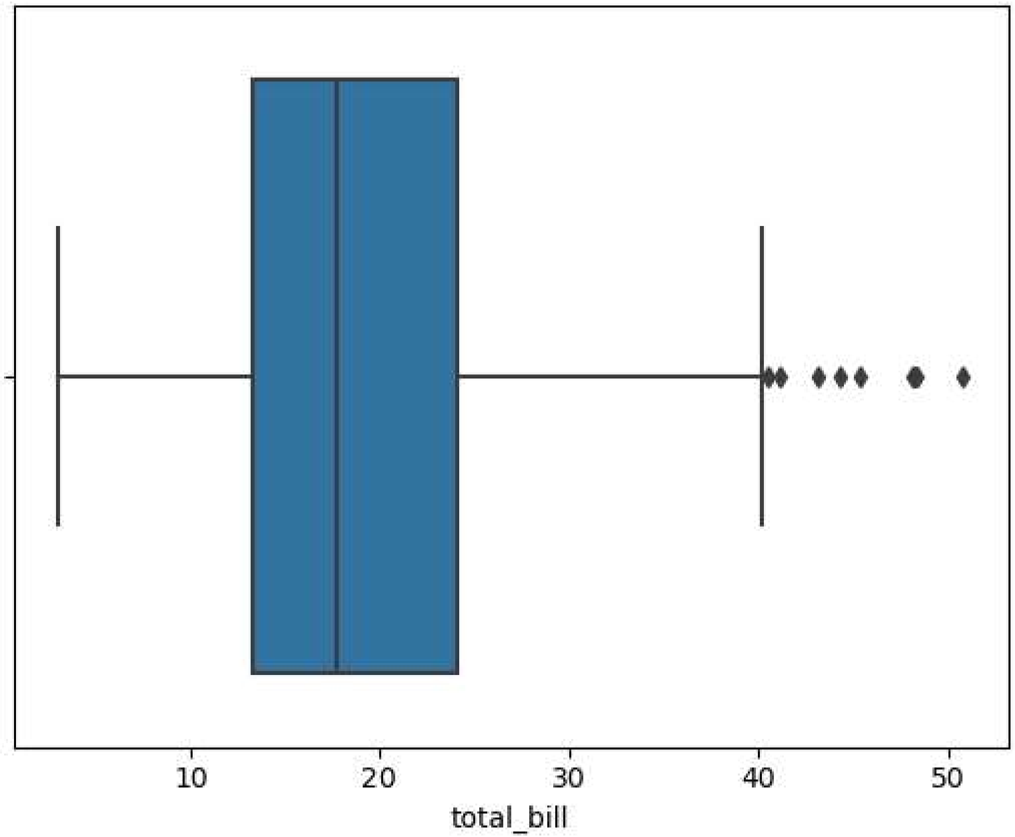
1. Percentiles and Quartiles: Percentiles divide data into 100 equal parts; quartiles into four.
2. Interquartile Range (IQR): Range between Q1 (25th percentile) and Q3 (75th percentile).

Welcome
Home

```
In [18]: sns.boxplot(data["total_bill"])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
e as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

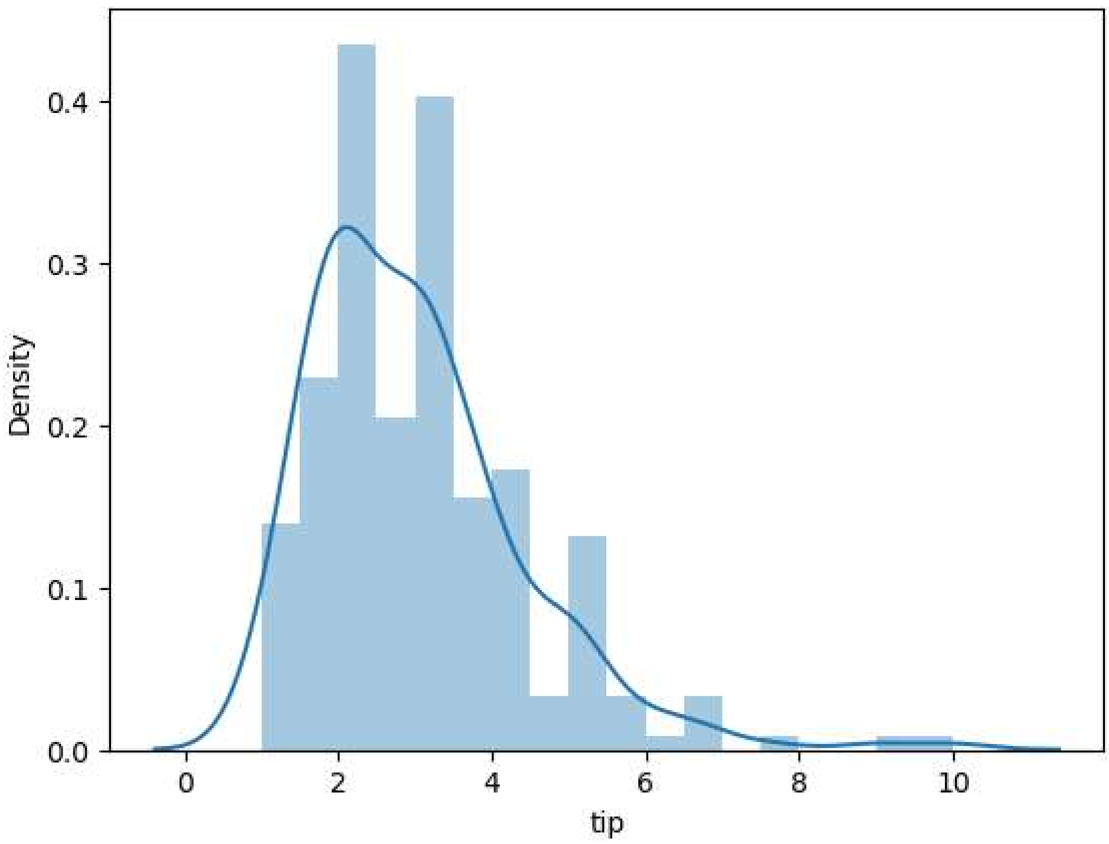
```
Out[18]: <AxesSubplot:xlabel='total_bill'>
```

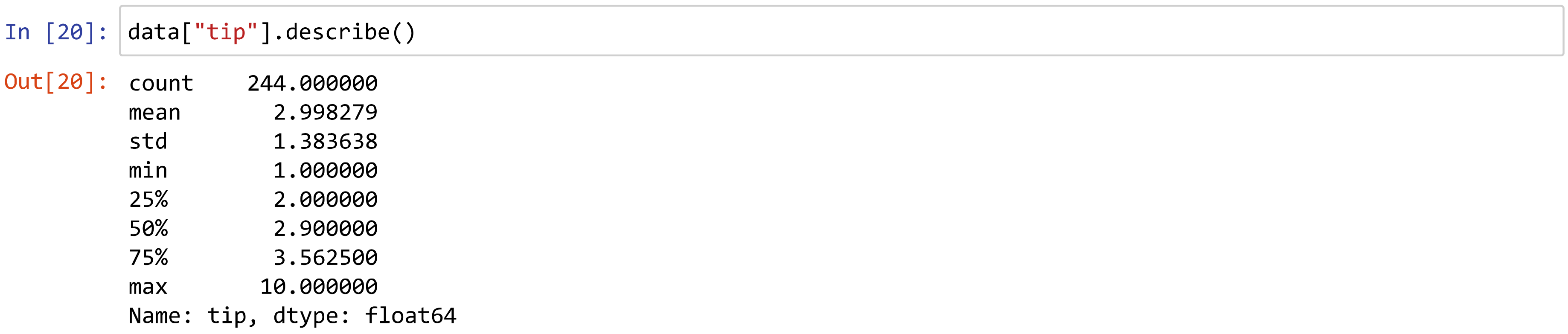


In [19]:
```
sns.distplot(data["tip"])
```

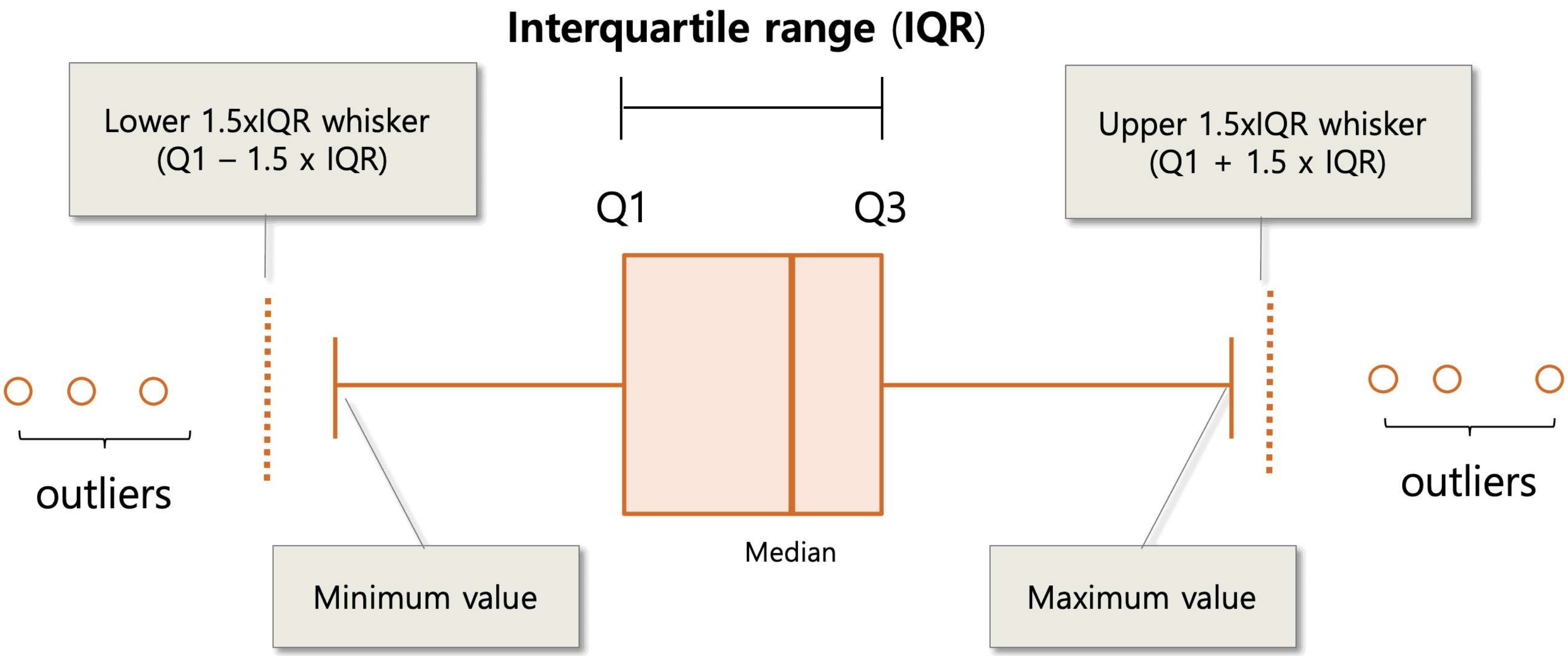
```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprec
ated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

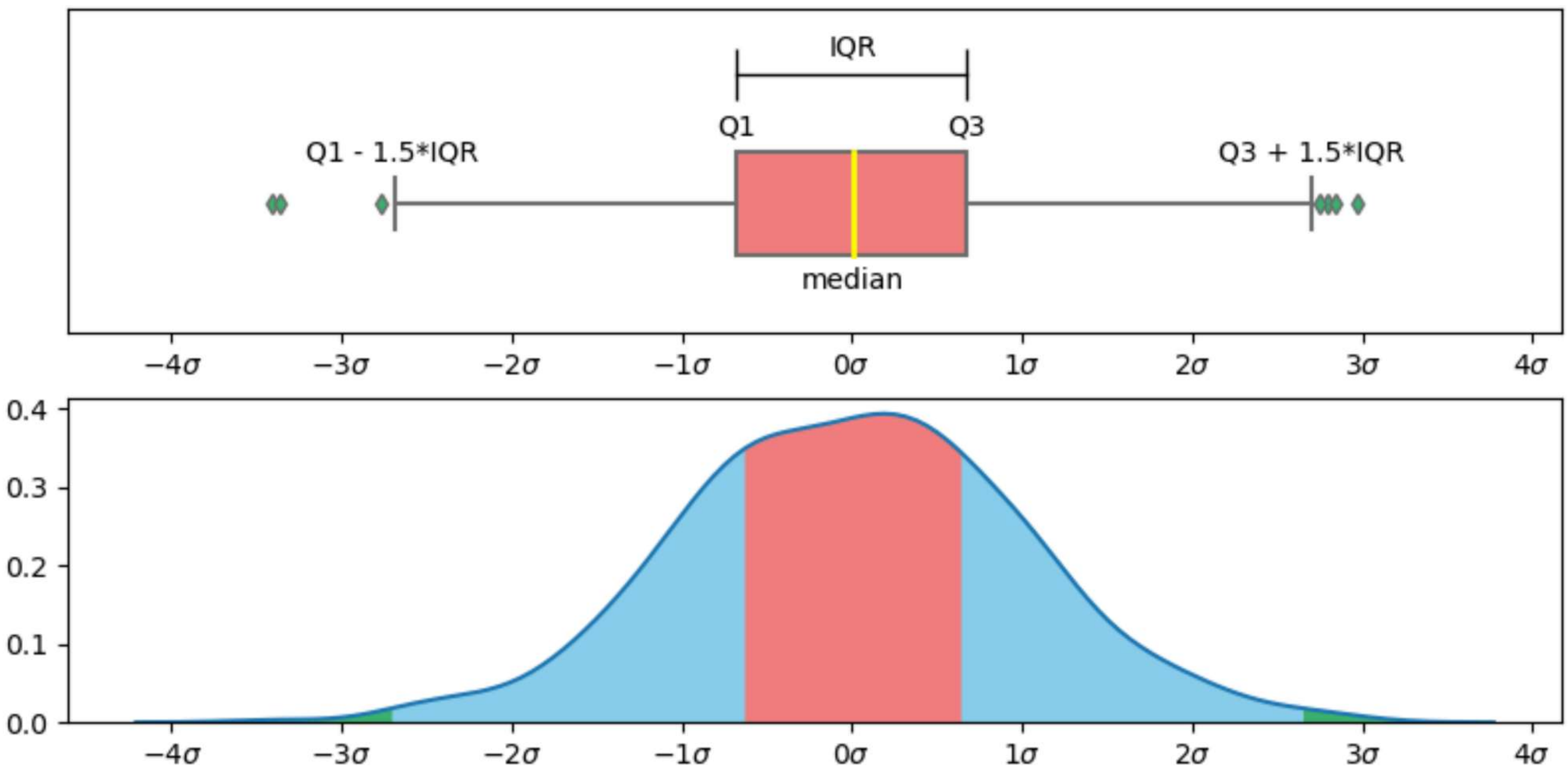
Out[19]:
```
<AxesSubplot:xlabel='tip', ylabel='Density'>
```





```
In [20]: data["tip"].describe()

Out[20]: count    244.000000
         mean       2.998279
         std        1.383638
         min        1.000000
         25%        2.000000
         50%        2.900000
         75%        3.562500
         max       10.000000
         Name: tip, dtype: float64
```



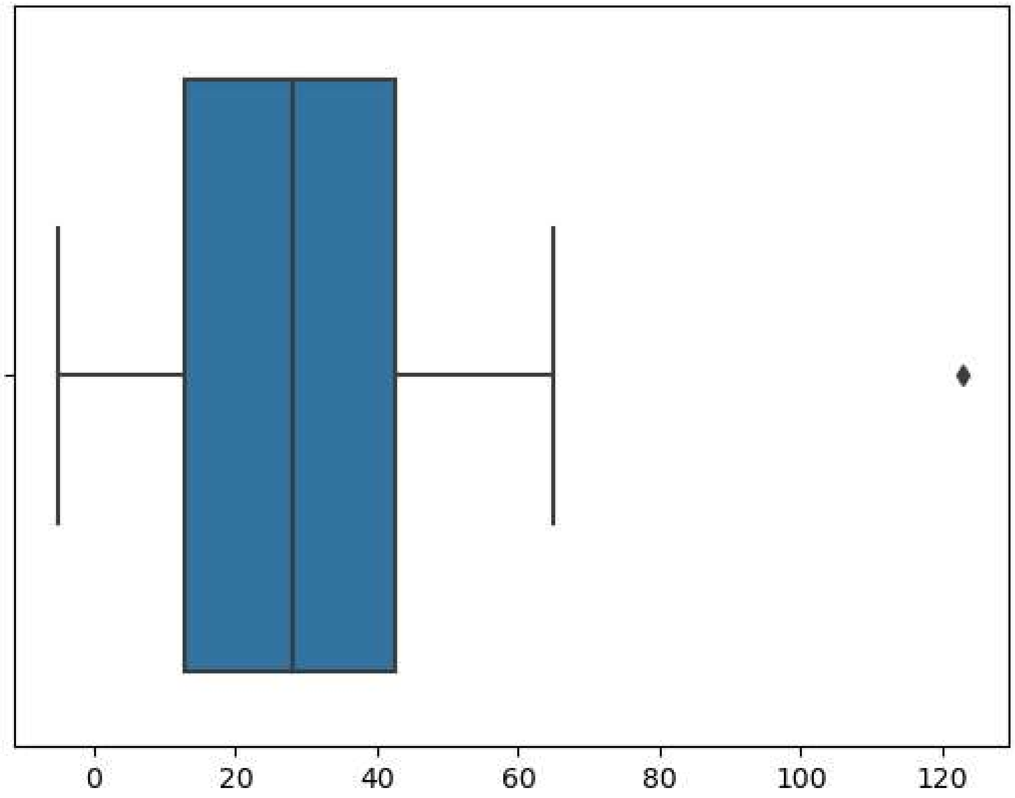


In [21]:
```
x=[-5,12,45,35,22,10,34,65,15,123]
```

In [22]:
```
sns.boxplot(x)
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
e as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[22]:
```
<AxesSubplot:>
```



In [23]:
```
# Percentiles and Quartiles
q1 = np.quantile(x, .25)
q3 = np.quantile(x, .75)
q1
```

Out[23]:
```
12.75
```

In [24]:
```
q3
```

Out[24]:
```
42.5
```

In [25]:
```
iqr=q3-q1
iqr
```

Out[25]:
```
29.75
```

In [26]:
```
lower_bound = q1 - (1.5 * iqr)
upper_bound = q3 + (1.5 * iqr)
```

In [27]:
```
lower_bound
```

Out[27]:
```
-31.875
```

In [28]:
```
data["tip"]
```

Out[28]:
```
0      1.01
1      1.66
2      3.50
3      3.31
4      3.61
       ...
239    5.92
240    2.00
241    2.00
242    1.75
243    3.00
Name: tip, Length: 244, dtype: float64
```

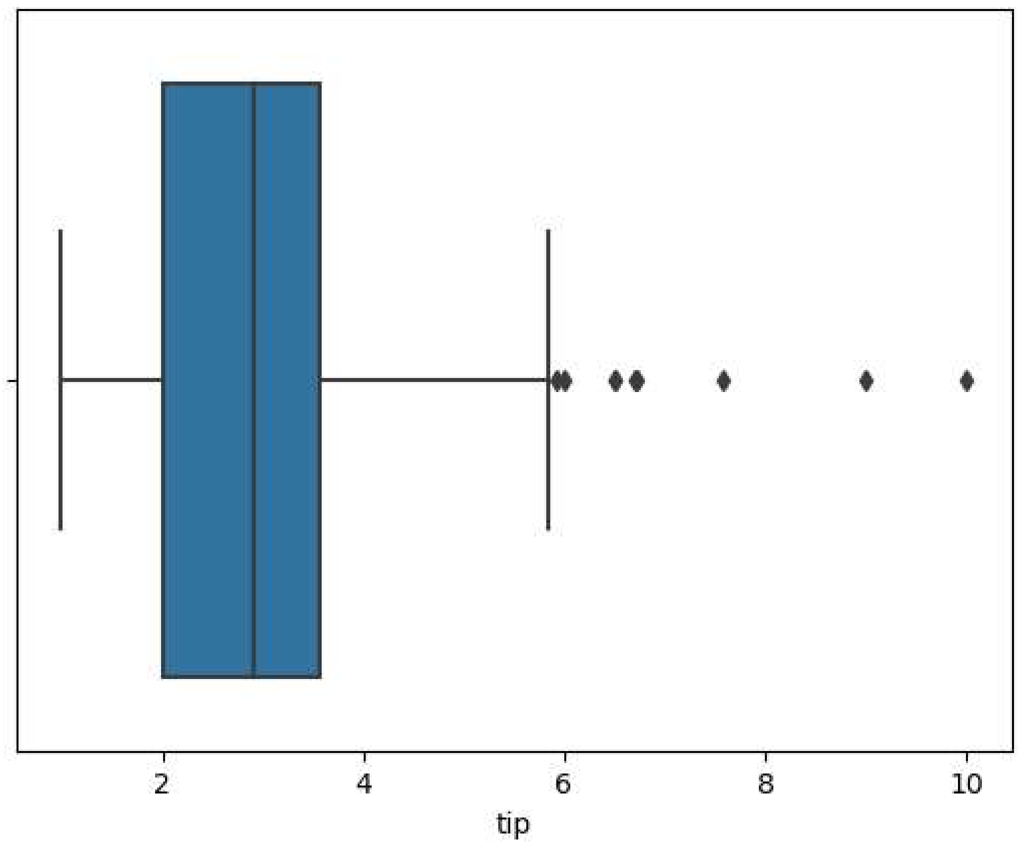
In [29]:

```
q1=np.quantile(data["tip"],.25)
q3=np.quantile(data["tip"],.75)
iqr=q3-q1
lower_bound = q1 - (1.5 * iqr)
upper_bound = q3 + (1.5 * iqr)
print(upper_bound)
print(lower_bound)

sns.boxplot(data["tip"])
```

```
5.90625
-0.34375

C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
e as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[29]:

```
<AxesSubplot:xlabel='tip'>
```


2
4
6
8
10
tip

# 5. Measures of Symmetry: Skewness and Kurtosis

### Skewness: Measures asymmetry. Positive skew (right skew) indicates a tail to the right; negative skew (left skew) indicates a tail to the left.

In [30]:

```
data
```

Out[30]:

| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| **1** | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| **2** | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 |
| **3** | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 |
| **4** | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **239** | 29.03 | 5.92 | Male | No | Sat | Dinner | 3 |
| **240** | 27.18 | 2.00 | Female | Yes | Sat | Dinner | 2 |
| **241** | 22.67 | 2.00 | Male | Yes | Sat | Dinner | 2 |
| **242** | 17.82 | 1.75 | Male | No | Sat | Dinner | 2 |
| **243** | 18.78 | 3.00 | Female | No | Thur | Dinner | 2 |

244 rows × 7 columns

In [31]:

```
nu=data.select_dtypes(include="number")
nu
```

Out[31]:

|  | total_bill | tip | size |
|---|---|---|---|
| **0** | 16.99 | 1.01 | 2 |
| **1** | 10.34 | 1.66 | 3 |
| **2** | 21.01 | 3.50 | 3 |
| **3** | 23.68 | 3.31 | 2 |
| **4** | 24.59 | 3.61 | 4 |
| **...** | ... | ... | ... |
| **239** | 29.03 | 5.92 | 3 |
| **240** | 27.18 | 2.00 | 2 |
| **241** | 22.67 | 2.00 | 2 |
| **242** | 17.82 | 1.75 | 2 |
| **243** | 18.78 | 3.00 | 2 |

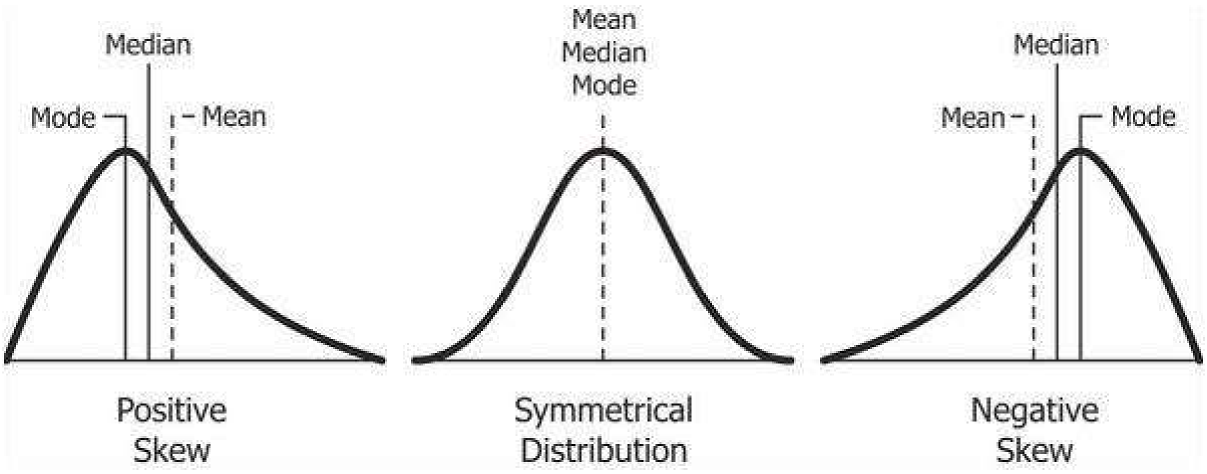
244 rows × 3 columns

In [32]:

```
nu.skew()
```

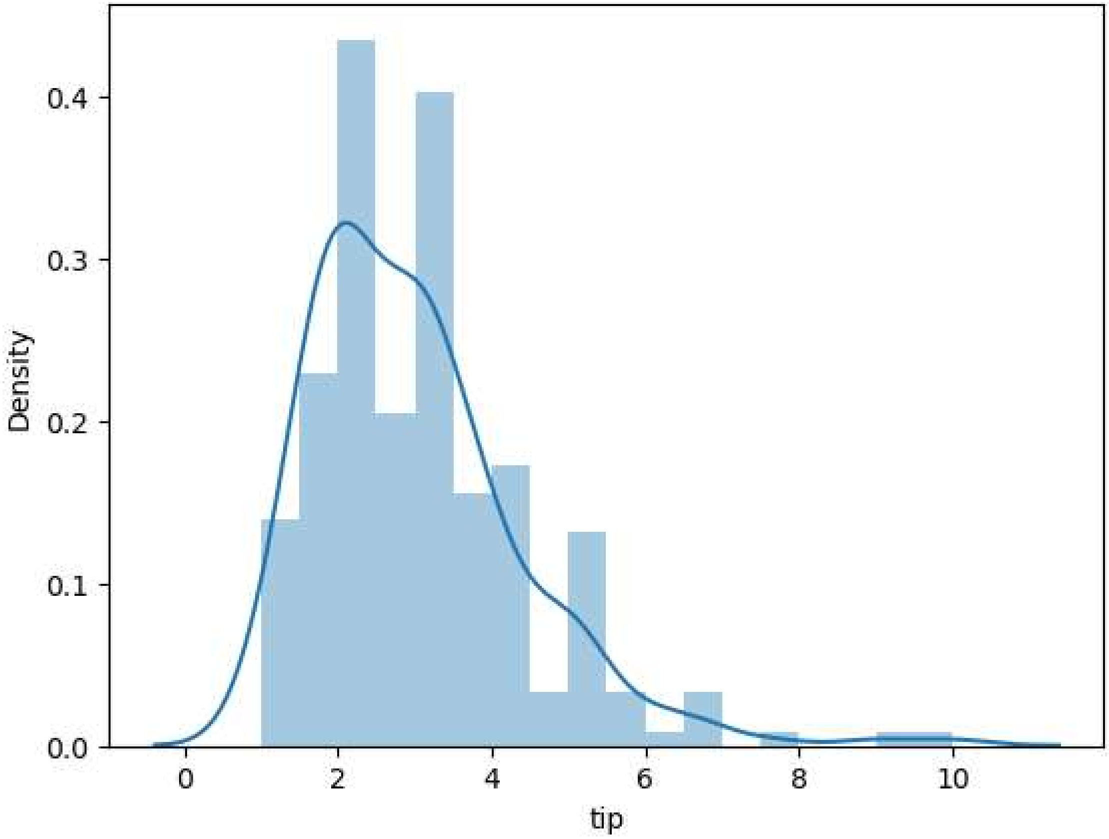
Out[32]:

```
total_bill    1.133213
tip           1.465451
size          1.447882
dtype: float64
```

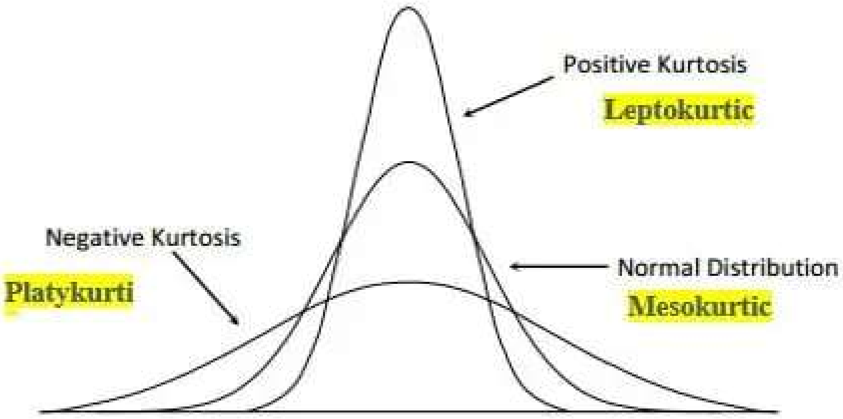

Median
Mode
Mean
Positive Skew
Mean
Median
Mode
Symmetrical Distribution
Median
Mean
Mode
Negative Skew

In [33]:
```
sns.distplot(data["tip"])

plt.show()
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprec
ated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



Kurtosis: Measures the "tailedness" of the distribution. High kurtosis (leptokurtic) indicates heavy tails, low kurtosis (platykurtic) indicates light tails.



| Type of Kurtosis | Tailedness | Outlier | Kurtosis Value |
|---|---|---|---|

```
In [34]: data["tip"].kurt()

Out[34]: 3.648375873352852
```

```
In [35]: nu.kurt()

Out[35]: total_bill    1.218484
         tip           3.648376
         size          1.731700
         dtype: float64
```

```
In [ ]:
```