```
import numpy as np
import pandas as pd
```

```
iris = 'https://gist.githubusercontent.com/curran/a08a1080b88344b0c8a7/raw/639388c2cbc2120a14dcf466e85730eb8be498bb/iris.csv'
df_iris = pd.read_csv(iris,sep =',')
print(type(df_iris))
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
spotify = '/content/spotify_top_songs_audio_features.csv'
df_spotify = pd.read_csv(spotify,sep = ',' )
print(type(df_spotify))
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
s=pd.Series([1,3,5,6,8])
print(type(s))
s
```

```
<class 'pandas.core.series.Series'>
0    1
1    3
2    5
3    6
4    8
dtype: int64
```

```
d=pd.DataFrame({'col':[1,2,3,4,5,6],'col2':[1,2,3,4,5,6],'col3':['1','2','3','4','5',None]})
print(d)
```

```
   col  col2  col3
0    1     1     1
1    2     2     2
2    3     3     3
3    4     4     4
4    5     5     5
5    6     6  None
```

```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   col     6 non-null      int64 
 1   col2    6 non-null      int64 
 2   col3    5 non-null      object
dtypes: int64(2), object(1)
memory usage: 272.0+ bytes
```

```
df_iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
df_iris.head(10)
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
9	4.9	3.1	1.5	0.1	setosa

```
df_spotify.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6513 entries, 0 to 6512
Data columns (total 19 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   id                6513 non-null   object 
 1   artist_names      6513 non-null   object 
 2   track_name        6513 non-null   object 
 3   source            6513 non-null   object 
 4   key               6513 non-null   object 
 5   mode              6513 non-null   object 
 6   time_signature    6513 non-null   object 
 7   danceability      6513 non-null   float64
 8   energy            6513 non-null   float64
 9   speechiness       6513 non-null   float64
 10  acousticness      6513 non-null   float64
 11  instrumentalness  6513 non-null   float64
 12  liveness          6513 non-null   float64
 13  valence           6513 non-null   float64
 14  loudness          6513 non-null   float64
 15  tempo             6513 non-null   float64
 16  duration_ms       6513 non-null   int64  
 17  weeks_on_chart    6513 non-null   int64  
 18  streams           6513 non-null   int64  
dtypes: float64(9), int64(3), object(7)
memory usage: 966.9+ KB
```

```
df_spotify.head(10)
```

	id	artist_names	track_name	source	key	mode	time_signature	danceability	energy	sp
0	000xQL6tZNLJzlrtlgxqSl	ZAYN, PARTYNEXTDOOR	Still Got Time (feat. PARTYNEXTDOOR)	RCA Records Label	G	Major	4 beats	0.748	0.627	
1	003eolwxETJujVWmNFMoZy	Alessia Cara	Growing Pains	Def Jam Recordings	C#/Db	Minor	4 beats	0.353	0.755	
2	003vvx7Niy0yvhvHt4a68B	The Killers	Mr. Brightside	Island Records	C#/Db	Major	4 beats	0.352	0.911	
3	00B7TZ0Xawar6NZ00JFomN	Cardi B, Chance the Rapper	Best Life (feat. Chance The Rapper)	Atlantic/KSR	A	Major	4 beats	0.620	0.625	
4	00Blm7zeNqgYLPtW6zg8cj	Post Malone, The Weeknd	One Right Now (with The Weeknd)	Republic Records	C#/Db	Major	4 beats	0.687	0.781	
5	00EPlEnX1JFjff8sC6bccd	Thalia, NATTI NATASHA	No Me Acuerdo	Sony Music Latin	G	Minor	4 beats	0.836	0.799	
6	00ETaeHUQ6lops3oWU1Wrt	Kygo, Donna Summer	Hot Stuff	RCA Records Label	F	Major	4 beats	0.681	0.773	
7	00ZKeP47bZtswtANkvxz2j	Tropa do Bruxo, DJ Ws da Igrejinha, SMU, Triz,...	Baile do Bruxo	Tropa Do Bruxo	G	Minor	5 beats	0.734	0.228	
8	00gpGR84M27moP7AFuqHlx	YBN Nahmir	Bounce Out With That	2018	G#/Ab	Major	4 beats	0.857	0.560	
9	00imgaPlYRrMGn9o83hfmk	Brent Faiyaz	LOOSE CHANGE	Lost Kids LLC., Marketed by Venice / Stem	C#/Db	Minor	4 beats	0.574	0.369	

```
df_iris.tail()
```

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

```
df_spotify.tail()
```

	id	artist_names	track_name	source	key	mode	time_signature	danceability
6508	7zgqtptZvhf8GEmdsM2vp2	Taylor Swift	...Ready For It?	Big Machine Records, LLC	D	Major	4 beats	0.615
6509	7zjEyeBsaw9gV0jofJLfOM	Young Thug, A$AP Rocky, Post Malone	Livin It Up (with Post Malone & A$AP Rocky)	300 Entertainment/Atl	G	Major	4 beats	0.767
6510	7zl7kehxesNEo2pYkKXTSe	Eminem, Jack Harlow, Cordae	Killer (feat. Jack Harlow & Cordae) – Remix	Shady/Aftermath/Interscope Records	B	Minor	4 beats	0.924
6511	7zvfDihYiJ8RQ1nRcpKBF5	Kendrick Lamar, Tanna Leone	Mr. Morale	pgLang/Top Dawg Entertainment/Aftermath/Inters...	A	Major	3 beats	0.727
6512	7zxRMhXxJMQCeDDg0rKAVo	NAV, The Weeknd	Some Way	XO Records	C	Major	4 beats	0.744

```
df_iris.describe()
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```
df_spotify.describe()
```

	danceability	energy	speechiness	acousticness	instrumentalness	liveness	valence	loudness	tempo	
count	6513.000000	6513.000000	6513.000000	6513.000000	6513.000000	6513.000000	6513.000000	6513.000000	6513.000000	
mean	0.681731	0.636522	0.121933	0.236761	0.012469	0.180168	0.492412	−6.350667	122.117244	2
std	0.141787	0.164813	0.113441	0.244784	0.075151	0.138054	0.227001	2.536114	29.416097	
min	0.150000	0.021800	0.023200	0.000008	0.000000	0.019700	0.032000	−34.475000	46.718000	
25%	0.591000	0.534000	0.044000	0.044400	0.000000	0.097400	0.316000	−7.564000	98.007000	1
50%	0.698000	0.651000	0.072200	0.145000	0.000000	0.124000	0.489000	−5.983000	120.034000	1
75%	0.785000	0.759000	0.163000	0.356000	0.000041	0.219000	0.669000	−4.673000	142.025000	2
max	0.985000	0.989000	0.966000	0.994000	0.953000	0.977000	0.982000	1.509000	212.117000	6

```
df_iris.dtypes
```

```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
species          object
dtype: object
```

```
df_spotify.dtypes
```

```
id                    object
artist_names          object
track_name            object
source                object
key                   object
mode                  object
time_signature        object
danceability         float64
energy               float64
speechiness          float64
acousticness         float64
instrumentalness     float64
liveness             float64
valence              float64
loudness             float64
tempo                float64
duration_ms            int64
weeks_on_chart         int64
streams                int64
dtype: object
```

```
print(df_iris.index)
```

```
RangeIndex(start=0, stop=150, step=1)
```

```
print(df_spotify.index)
```

```
RangeIndex(start=0, stop=6513, step=1)
```

```
df_iris.columns
```

```
    Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
           'species'],
          dtype='object')
```

```
df_spotify.columns
```

```
    Index(['id', 'artist_names', 'track_name', 'source', 'key', 'mode',
           'time_signature', 'danceability', 'energy', 'speechiness',
           'acousticness', 'instrumentalness', 'liveness', 'valence', 'loudness',
           'tempo', 'duration_ms', 'weeks_on_chart', 'streams'],
          dtype='object')
```

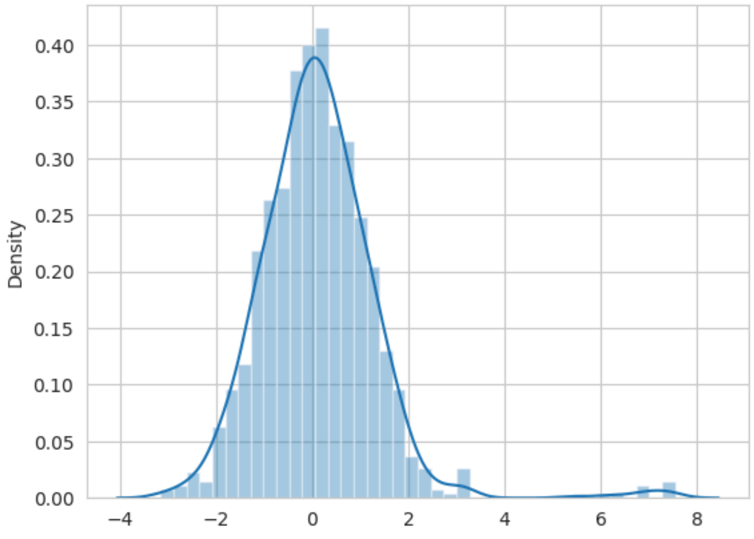
```
attributes = ['sepal_length','sepal_width','petal_length','petal_width','additional']
df_iris.columns = attributes
df_iris.head(1)#重新命名
```

	sepal_length	sepal_width	petal_length	petal_width	additional
0	5.1	3.5	1.4	0.2	setosa

```
concat#用于左右合并
duplicated(keep=False去重)
```

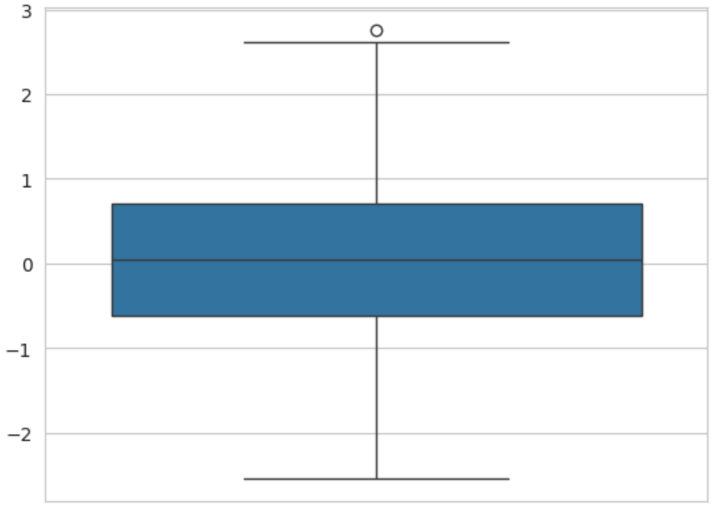
```
import seaborn as sns #处理outlier的时候需要使用seaborn函数
dt_outlier = np.concatenate([np.random.randn(1000),np.random.normal(7,1,10)])
sns.set_style('whitegrid')
sns.distplot(dt_outlier)
```

```
    <ipython-input-35-aabf72eeb118>:4: UserWarning:

    `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(dt_outlier)
    <Axes: ylabel='Density'>
```



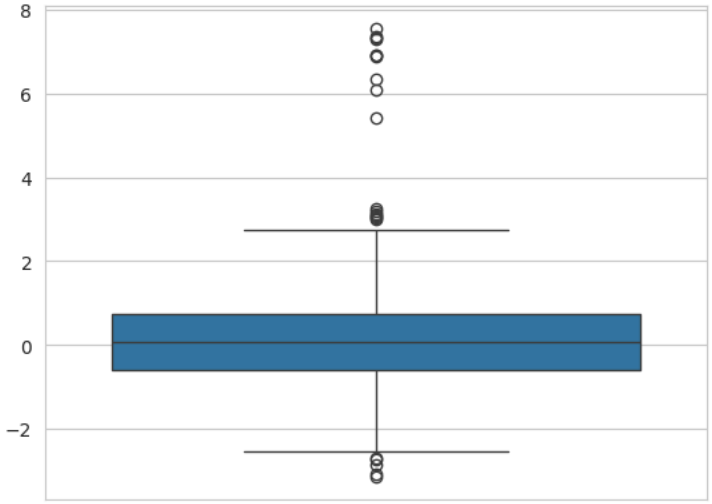
```
def iqr_outlier_rm(dt_input):
  lq,uq = np.percentile(dt_input,[25,75])
  lower_l = lq - 1.5*(uq-lq)
  upper_l = uq + 1.5*(uq-lq)
  return dt_input[(dt_input >= lower_l)&(dt_input <= upper_l)]

dt_outlier_ws = iqr_outlier_rm(dt_outlier)
sns.boxplot(dt_outlier_ws,orient='v')
```

```
<Axes: >
```



```
sns.boxplot(dt_outlier,orient = 'v')
```

```
<Axes: >
```



```
raw_data = {'name': ['Jason', np.nan, 'Mike', 'Rayman', 'Alex', 'Meimei'],
        'age': [36, np.nan, 36, 18, 36, 16],
        'gender': ['m', np.nan, 'm', np.nan, 'f', 'f'],
        'preMLScore': [1, np.nan, np.nan, 2, 3, 90],
        'postMLScore': [65, np.nan, np.nan, 62, 70, 100]}

# create a dataframe by passing a dictionary
df = pd.DataFrame(raw_data, columns = ['name', 'age', 'gender', 'preMLScore', 'postMLScore'])
```

```
df
```

	name	age	gender	preMLScore	postMLScore
0	Jason	36.0	m	1.0	65.0
1	NaN	NaN	NaN	NaN	NaN
2	Mike	36.0	m	NaN	NaN
3	Rayman	18.0	NaN	2.0	62.0
4	Alex	36.0	f	3.0	70.0
5	Meimei	16.0	f	90.0	100.0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   name         5 non-null      object 
 1   age          5 non-null      float64
 2   gender       4 non-null      object 
 3   preMLScore   4 non-null      float64
 4   postMLScore  4 non-null      float64
dtypes: float64(3), object(2)
memory usage: 368.0+ bytes
```

```
df.isnull()
```

	name	age	gender	preMLScore	postMLScore
0	False	False	False	False	False
1	True	True	True	True	True
2	False	False	False	True	True
3	False	False	True	False	False
4	False	False	False	False	False
5	False	False	False	False	False

```
df.isnull().sum()
```

```
name           1
age            1
gender         2
preMLScore     2
postMLScore    2
dtype: int64
```

```
df.isnull().any(axis = 0)
```

```
name           True
age            True
gender         True
preMLScore     True
postMLScore    True
dtype: bool
```

```
df
```

	name	age	gender	preMLScore	postMLScore
0	Jason	36.0	m	1.0	65.0
1	NaN	NaN	NaN	NaN	NaN
2	Mike	36.0	m	NaN	NaN
3	Rayman	18.0	NaN	2.0	62.0
4	Alex	36.0	f	3.0	70.0
5	Meimei	16.0	f	90.0	100.0

```
df.dropna(axis = 0,how = 'any')#去除NA缺失值
```

	name	age	gender	preMLScore	postMLScore
0	Jason	36.0	m	1.0	65.0
4	Alex	36.0	f	3.0	70.0
5	Meimei	16.0	f	90.0	100.0

```
df.dropna(axis =1,how = 'any')
```

```
0
1
2
3
4
5
```

```
df=df.dropna(how = 'all',inplace = False)
df
```

```
     name  age  gender  preMLScore  postMLScore
```