In [1]:

```
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
import statistics
```

In [3]:

```
#mean,median,mode
```

In [4]:

```
df= sns.load_dataset('tips')
```

In [5]:

```
df.head()
```

Out[5]:

	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

In [6]:

```
np.mean(df['total_bill'])
```

Out[6]:

```
19.785942622950824
```

In [7]:

```
np.median(df['total_bill'])
```

Out[7]:

```
17.795
```

In [8]:

```
statistics.mode(df['total_bill'])
```

Out[8]:

```
13.42
```

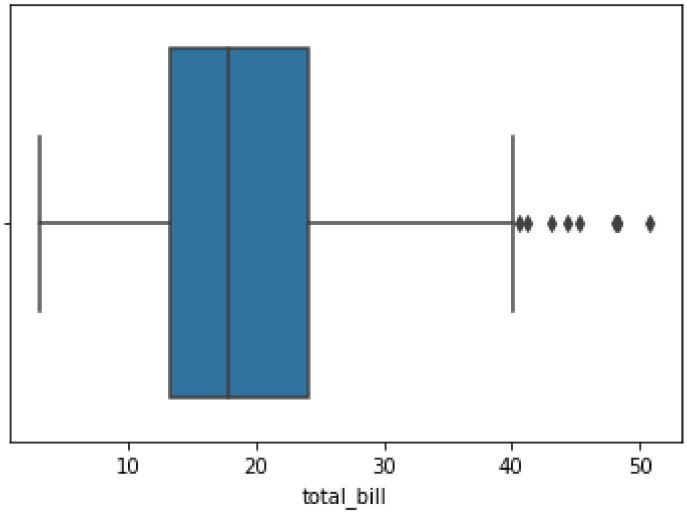
In [9]:

```
sns.boxplot(df['total_bill'])
```

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

Out[9]:

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

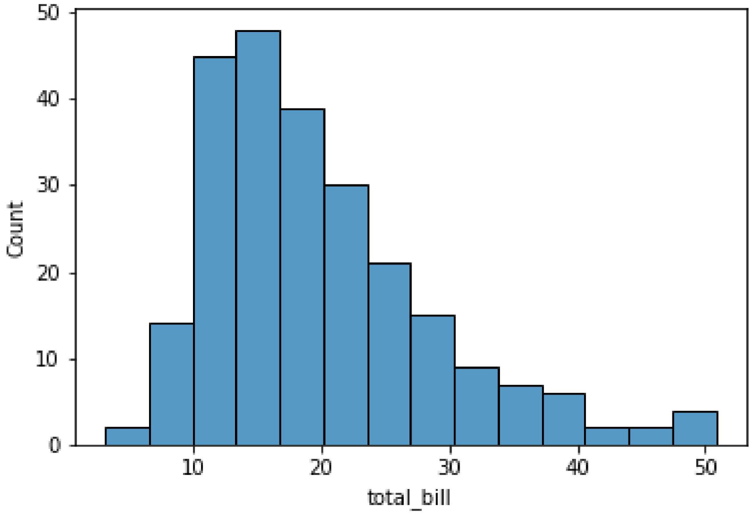


In [10]:

```
sns.histplot(df['total_bill'])
```

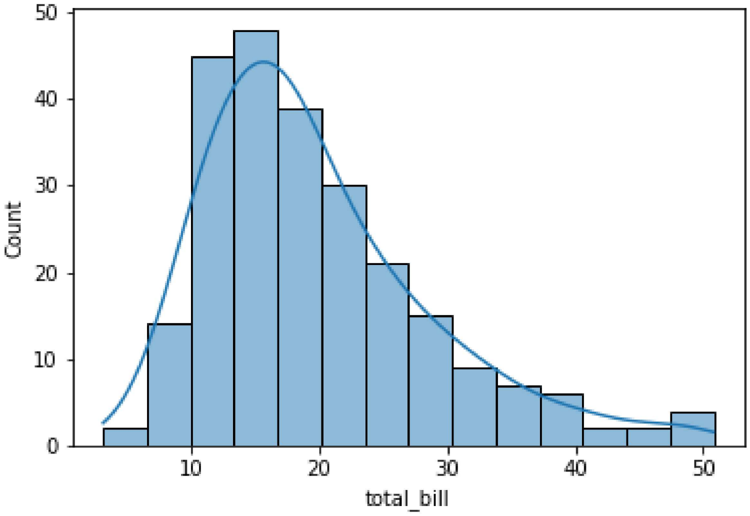
Out[10]:

```
<AxesSubplot:xlabel='total_bill', ylabel='Count'>
```



In [11]:
```
sns.histplot(df['total_bill'],kde=True)
```

Out[11]:
```
<AxesSubplot:xlabel='total_bill', ylabel='Count'>
```



In [12]:
```
df1=sns.load_dataset('iris')
```

In [13]:
```
df1.head()
```

Out[13]:

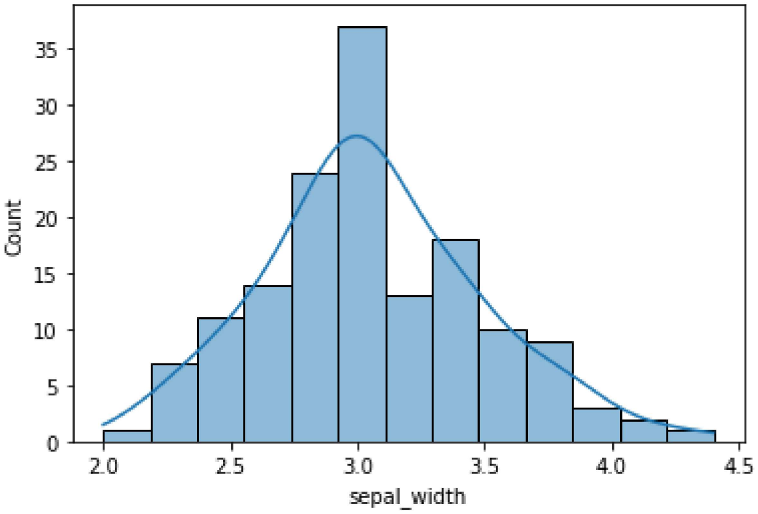
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [14]:

```python
sns.histplot(df1['sepal_width'],kde=True)
```

Out[14]:

```
<AxesSubplot:xlabel='sepal_width', ylabel='Count'>
```



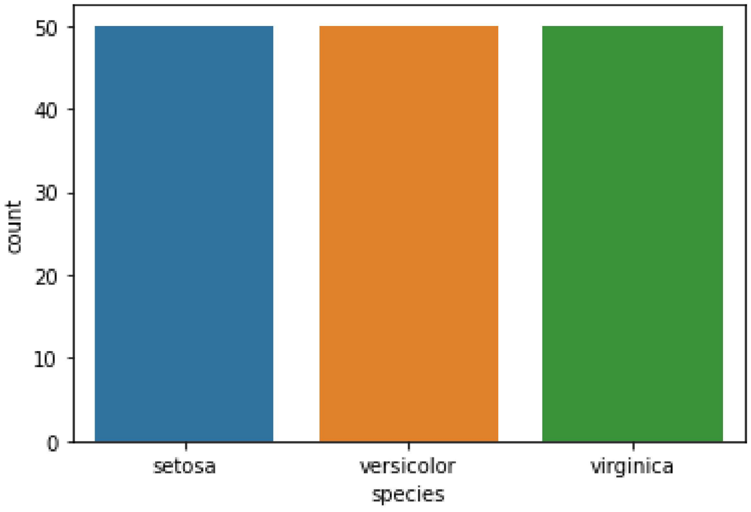
In [15]:

```python
sns.countplot(df1['species'])
```

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

Out[15]:

```
<AxesSubplot:xlabel='species', ylabel='count'>
```



In [16]:

```python
np.percentile(df1['sepal_length'],[25,75])
```

Out[16]:

```
array([5.1, 6.4])
```

In [17]:

```python
### Outliers
```

In [18]:

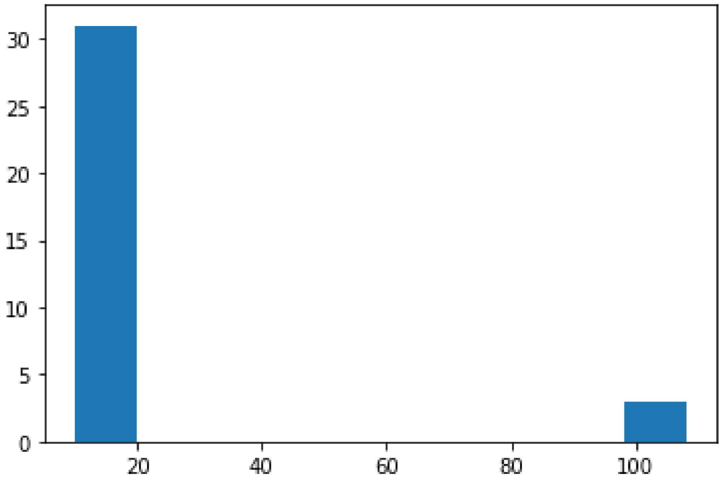
```python
### Define our dataset
dataset =[11,10,12,14,12,15,14,13,15,102,12,14,17,19,107, 10,13,12,14,12,108
```

In [19]:

```
plt.hist(dataset)
```

Out[19]:

```
(array([31.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  3.]),
 array([ 10. ,  19.8,  29.6,  39.4,  49.2,  59. ,  68.8,  78.6,  88.4,
         98.2, 108. ]),
 <BarContainer object of 10 artists>)
```



In [20]:

```
## Z score
outliers = []

def detect_outliers(data):
 threshold = 3  ## 3 std deviation
 mean = np.mean(dataset)
 std = np.std(dataset)

 for i in data:
    z_score=(i-mean)//std
    if np.abs(z_score)>threshold:
     outliers.append(i)

 return outliers
```

In [21]:

```
detect_outliers(dataset)
```

Out[21]:

```
[]
```

#IQR

1. Sort the data
2. Calculate Q1 andQ3
3. Find IQR(Q3-Q1)
4. Find the Lower fence(Q1-1.5(iqr))
5. FInd the Upper fence(Q3-1.5(iqr))

In [22]:

```
dataset=sorted(dataset)
```

In [23]:

```
dataset
```

Out[23]:

```
[10,
 10,
 10,
 10,
 10,
 11,
 11,
 12,
 12,
 12,
 12,
 12,
 12,
 12,
 13,
 13,
 13,
 13,
 14,
 14,
 14,
 14,
 14,
 14,
 15,
 15,
 15,
 15,
 15,
 17,
 19,
 102,
 107,
 108]
```

In [24]:

```
q1,q3=np.percentile(dataset,[25,75])
```

In [25]:

```
print(q1,q3)
```

```
12.0 15.0
```

In [26]:

```
iqr=q3-q1
print(iqr)
```

```
3.0
```

In [27]:

```
## Find the lower fence and higher fence
lower_fence=q1-(1.5*iqr)
higher_fence=q3+(1.5*iqr)
```

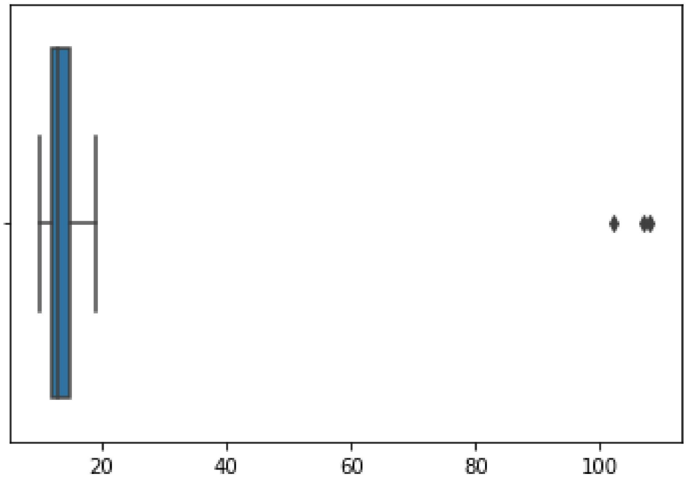
In [28]:
```
print(lower_fence,higher_fence)
```

```
7.5 19.5
```

In [29]:
```
sns.boxplot(dataset)
```

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

Out[29]:
```
<AxesSubplot:>
```



Suppose the IQ in a certain population is normally distributed with a mean of μ = 100 and standard deviation of σ = 15.

A researcher wants to know if a new drug affects IQ levels, so he recruits 20 patients to try it and records their IQ levels.

The following code shows how to perform a one sample z-test in Python to determine if the new drug causes a significant difference in IQ levels:

In [30]:
```
from statsmodels.stats.weightstats import ztest as ztest

#enter IQ levels for 20 patients
data = [88, 92, 94, 94, 96, 97, 97, 97, 99, 99,
        105, 109, 109, 109, 110, 112, 112, 113, 114, 115]

ztest(data,value=110)
```

Out[30]:
```
(-3.640487595530384, 0.00027212221833431376)
```

In [31]:
```
## t -test

ages=[10,20,35,50,28,40,55,18,16,55,30,25,43,18,30,28,14,24,16,17,32,35,26,2
```

In [32]:

```
import numpy as np
ages_mean=np.mean(ages)
ages_mean
```

Out[32]:

```
30.34375
```

In [33]:

```
sample_size=10
age_sample=np.random.choice(ages,sample_size)
```

In [34]:

```
age_sample
```

Out[34]:

```
array([20, 16, 65, 65, 40, 55, 10, 23, 16, 30])
```

In [35]:

```
np.mean(age_sample)
```

Out[35]:

```
34.0
```

In [36]:

```
from scipy.stats import ttest_1samp
```

In [37]:

```
ttest_1samp(age_sample,30)
```

Out[37]:

```
Ttest_1sampResult(statistic=0.6033274769885615, pvalue=0.561185830671401)
```

In [38]:

```
ttest_1samp(age_sample,31)
```

Out[38]:

```
Ttest_1sampResult(statistic=0.4524956077414211, pvalue=0.6616217915232363)
```

In [39]:

```
ttest_1samp(age_sample,28)
```

Out[39]:

```
Ttest_1sampResult(statistic=0.9049912154828422, pvalue=0.38905355118870044)
```

In [40]:

```
ttest_1samp(age_sample,26)
```

Out[40]:

```
Ttest_1sampResult(statistic=1.206654953977123, pvalue=0.25832360566613216)
```

Consider anothe example

#ages of the college students(population) #1 class student mean of all the ages

In [41]:

```
import numpy as np
import pandas as pd
import scipy.stats as stats
import math
np.random.seed(6)
school_ages=stats.poisson.rvs(loc=18,mu=35,size=1500)
classA_ages=stats.poisson.rvs(loc=18,mu=30,size=60)
```

In [42]:

```
school_ages
```

Out[42]:

```
array([62, 59, 44, ..., 45, 52, 50])
```

In [43]:
```
classA_ages
```

Out[43]:
```
array([52, 46, 40, 40, 47, 50, 51, 45, 44, 52, 46, 53, 43, 44, 51, 50, 54,
       42, 54, 45, 61, 53, 49, 46, 47, 41, 45, 51, 43, 45, 48, 50, 40, 52,
       44, 55, 54, 40, 45, 46, 54, 42, 46, 35, 51, 51, 46, 48, 47, 35, 52,
       52, 39, 44, 48, 40, 42, 46, 47, 45])
```

In [44]:
```
classA_ages.mean()
```

Out[44]:
```
46.9
```

In [45]:
```
ttest_1samp(classA_ages,popmean=school_ages.mean())
```

Out[45]:
```
Ttest_1sampResult(statistic=-9.604796510704091, pvalue=1.139027071016194e-13)
```

In [46]:
```
school_ages.mean()
```

Out[46]:
```
53.303333333333335
```

In [47]:
```
if p_value<=0.05:
    print("Reject H0  ")
else:
    print("Accept Ho")
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\ARYASH~1\AppData\Local\Temp/ipykernel_14664/4054940971.py in <module>
----> 1 if p_value<=0.05:
      2     print("Reject H0  ")
      3 else:
      4     print("Accept Ho")

NameError: name 'p_value' is not defined
```

In [48]:
```
import seaborn as sns
```

In [49]:
```
df=sns.load_dataset('iris')
df.head()
```

Out[49]:

	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

In [50]:

```
df.corr()
```

Out[50]:

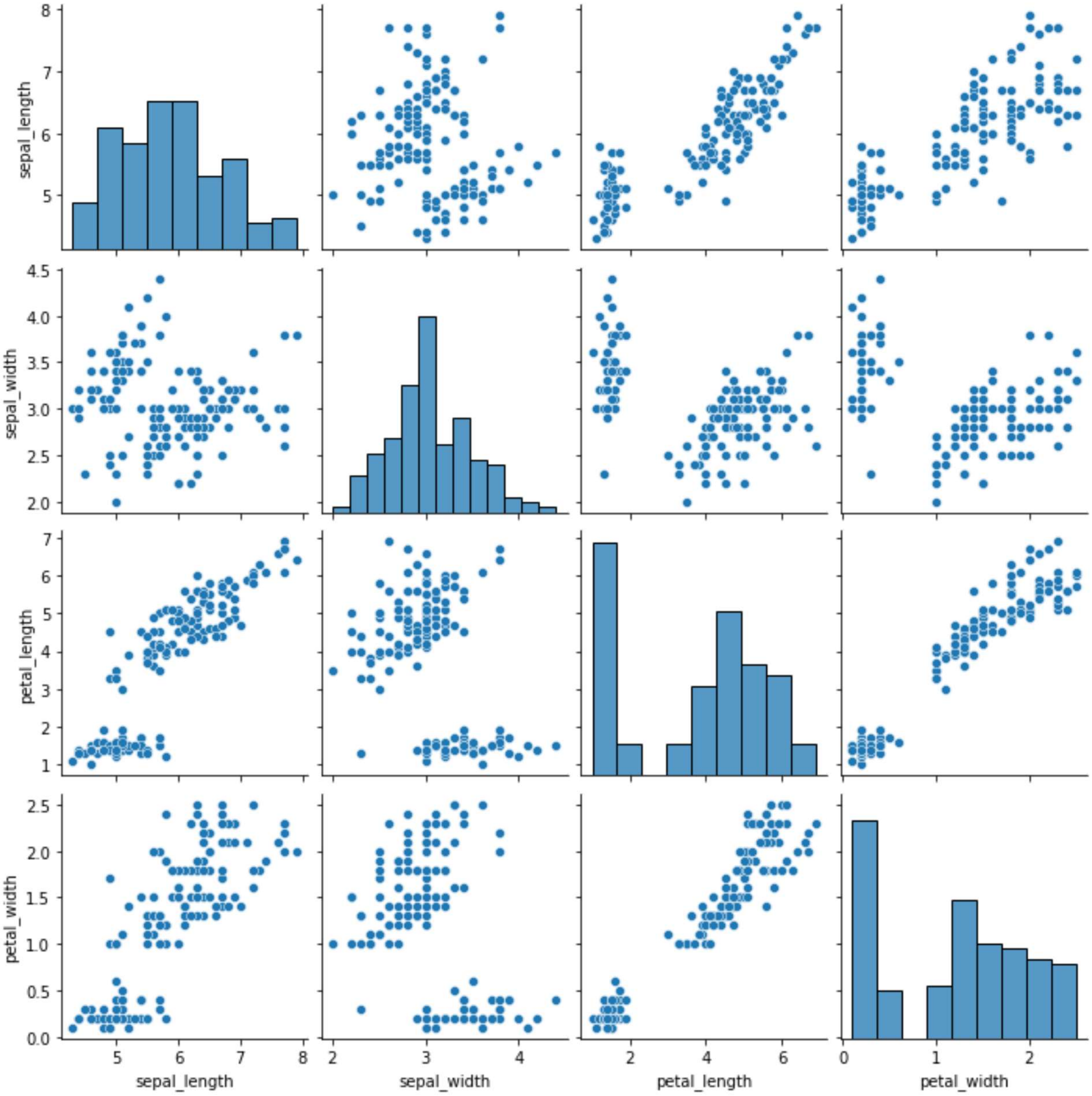
	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

In [51]:

```
sns.pairplot(df)
```

Out[51]:

```
<seaborn.axisgrid.PairGrid at 0x1f08aad65e0>
```


sepal_length
sepal_width
petal_length
petal_width
sepal_length
sepal_width
petal_length
petal_width