In [77]:
```
#import libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stat
import pylab as pl
import statistics as st
import math
```

In [78]:
```
#assign data
dataset=[1,5,6,7,8,9,10,12,13,14,15,16,18,19,20,21,22,23,24,25,26,27,28 ]   #assu
dataset
```

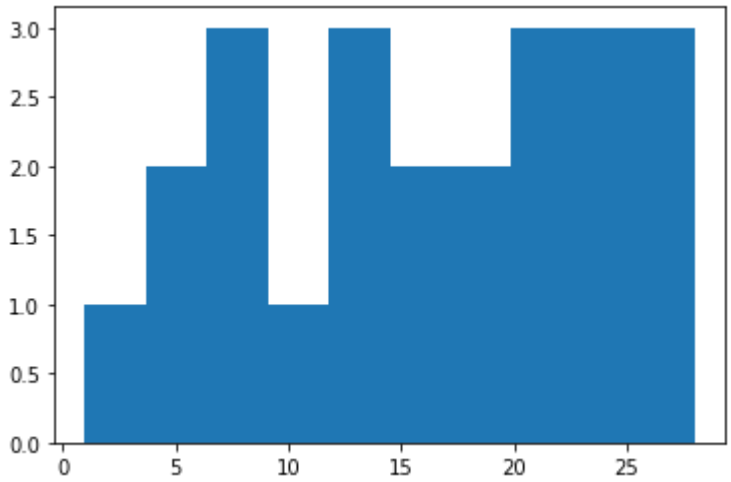
Out[78]:
```
[1,
 5,
 6,
 7,
 8,
 9,
 10,
 12,
 13,
 14,
 15,
 16,
 18,
 19,
 20,
 21,
 22,
 23,
 24,
 25,
 26,
 27,
 28]
```

In [79]:
```
sample_data=[1,5,6,7,8,9,10,12,13,14]
sample_data              #we take subdata from dataset that is called sample dat
```

Out[79]:
```
[1, 5, 6, 7, 8, 9, 10, 12, 13, 14]
```

In [80]:
```
plt.hist(dataset)   #continuous histogram
```

Out[80]:
```
(array([1., 2., 3., 1., 3., 2., 2., 3., 3., 3.]),
 array([ 1. ,  3.7,  6.4,  9.1, 11.8, 14.5, 17.2, 19.9, 22.6, 25.3, 28. ]),
 <BarContainer object of 10 artists>)
```

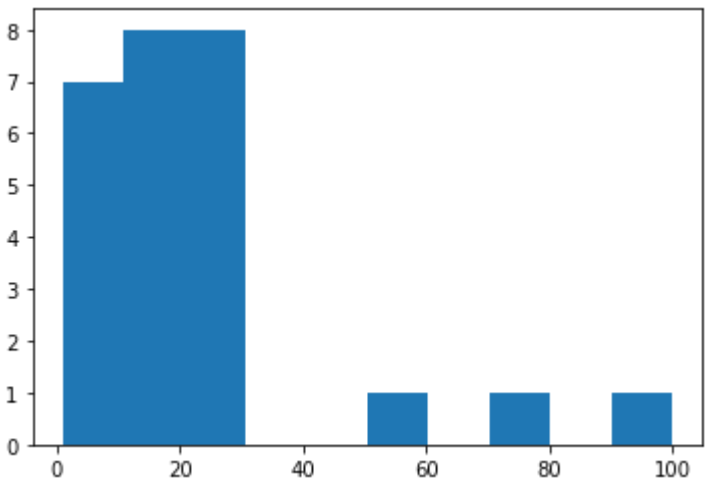


In [81]:
```
dataset1=[1,5,6,7,8,9,10,12,13,14,15,16,18,19,20,21,22,23,24,25,26,27,28,60,80,10
dataset1
```

Out[81]:
```
[1,
 5,
 6,
 7,
 8,
 9,
 10,
 12,
 13,
 14,
 15,
 16,
 18,
 19,
 20,
 21,
 22,
 23,
 24,
 25,
 26,
 27,
 28,
 60,
 80,
 100]
```

In [82]:
```
plt.hist(dataset1) #discrete histogram
```

Out[82]:
```
(array([7., 8., 8., 0., 0., 1., 0., 1., 0., 1.]),
 array([  1. ,  10.9,  20.8,  30.7,  40.6,  50.5,  60.4,  70.3,  80.2,
         90.1, 100. ]),
 <BarContainer object of 10 artists>)
```



In [83]:
```
dataset=[1,5,6,7,8,9,10,12,13,14,15,16,18,19,20,21,22,23,24,25,26,27,28 ]
```

In [84]:
```
#mean of dataset
print(np.mean(dataset))
print(st.mean(dataset))
```

```
16.043478260869566
16.043478260869566
```

In [85]:
```
#median of dataset
print(np.median(dataset))
print(st.median(dataset))
```

```
16.0
16
```

In [86]:
```
#mode of dataset
st.mode(dataset)
```

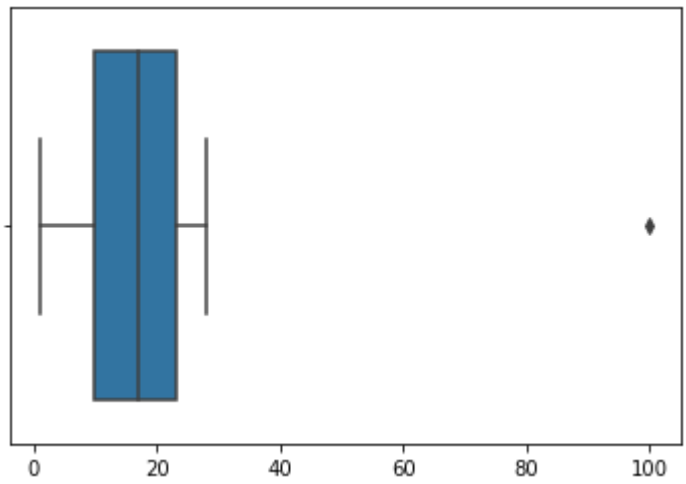
Out[86]:
```
1
```

In [87]:
```
dataset=[1,5,6,7,8,9,10,12,13,14,15,16,18,19,20,21,22,23,24,25,26,27,28,100]
```

In [88]:
```
 sns.boxplot(dataset)        # boxplot hepls finding the outliers in data
```

```
C:\Users\ASUS\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the on
ly valid positional argument will be `data`, and passing other arguments withou
t an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[88]:
```
<AxesSubplot:>
```



Five Number summary for finding the outliers

In [89]:
```
Q1=np.percentile(dataset,25)  # 25% of percentile of the dataset(1st quartile)
Q1
```

Out[89]: 9.75

In [90]:
```
Q3=np.percentile(dataset,75)           # 75% of percentile of the dataset(3rd quc
Q3
```

Out[90]: 23.25

In [91]:
```
IQR=Q3-Q1      # interquartile range
lower_fence=Q1-1.5*IQR
upper_fence=Q3+1.5*IQR
```

In [92]:
```
# 5 number summary
lower_fence,min(dataset),Q1,np.median(dataset),Q3,max(dataset),upper_fence
```

Out[92]: (-10.5, 1, 9.75, 17.0, 23.25, 100, 43.5)

measure of dispersion

In [93]:
```
st.variance(dataset)    # return sample varience of the data
```

Out[93]: 352.606884057971

In [94]:
```
st.pvariance(dataset)  # return population varience of data population varience
```

Out[94]: 337.91493055555554

In [95]:
```
np.var(dataset,axis=0)  # population varience
```

Out[95]: 337.9149305555556

In [96]:
```
math.sqrt(st.pvariance(dataset))           # return the standard deviation
```

Out[96]: 18.382462581372376

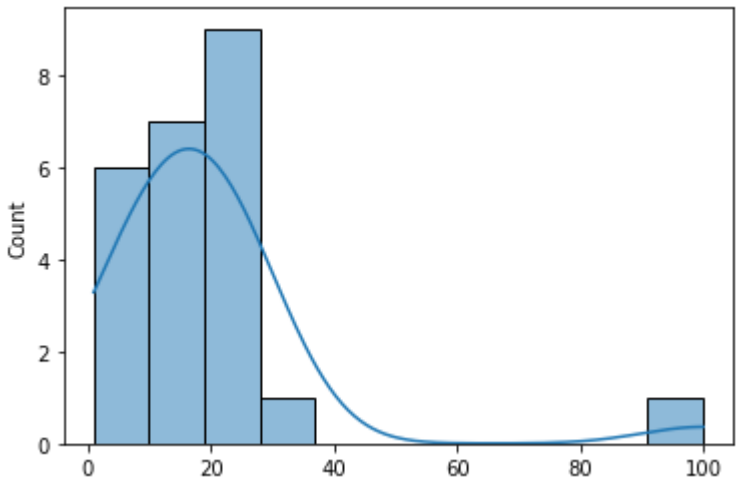
In [97]:
```
# population varience
def variance(data):
    n=len(data)
    #mean of data
    mean=sum(data)/n
    #varience of data
    deviation=[(x-mean)**2 for x in data]
    var=sum(deviation)/n
    return var
variance(dataset)
```

Out[97]: 337.91493055555554

Histogram & PDF

In [98]:
```
sns.histplot(dataset,kde=True) #compute a kernel density estimate to smooth the d
```

Out[98]: <AxesSubplot:ylabel='Count'>



Working with 'Titanic' dataset

In [99]:
```
df=sns.load_dataset('titanic')   #Load titanic data
df
```

Out[99]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True
...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True

891 rows × 15 columns

In [100]:
```
df.isnull().sum().sum()    #null values in df
```

Out[100]:
```
869
```

In [101]:
```
df1=df.dropna() #drop na values from the data (but it is not a proper way i just
df1
```

Out[101]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False
6	0	1	male	54.0	0	0	51.8625	S	First	man	True
10	1	3	female	4.0	1	1	16.7000	S	Third	child	False
11	1	1	female	58.0	0	0	26.5500	S	First	woman	False
...	...	...	...	...	...	...	...	...	...	...	...
871	1	1	female	47.0	1	1	52.5542	S	First	woman	False
872	0	1	male	33.0	0	0	5.0000	S	First	man	True
879	1	1	female	56.0	0	1	83.1583	C	First	woman	False
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True

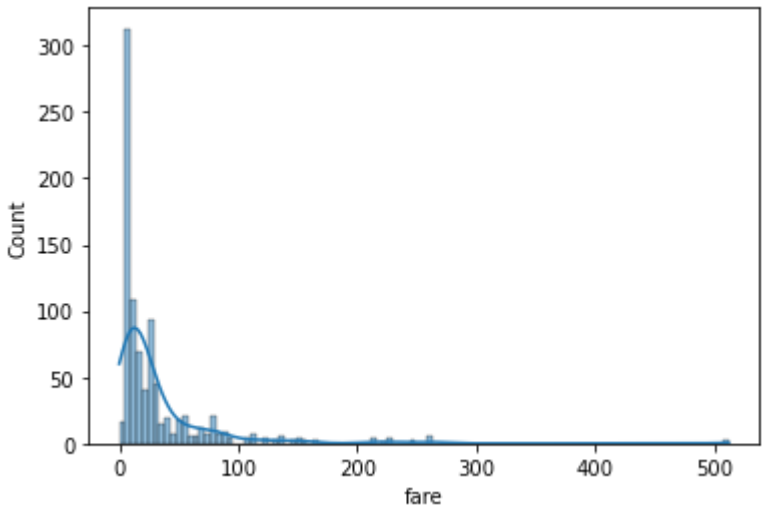
182 rows × 15 columns

In [102]:
```
df1.isnull().sum().sum()
```

Out[102]:
```
0
```

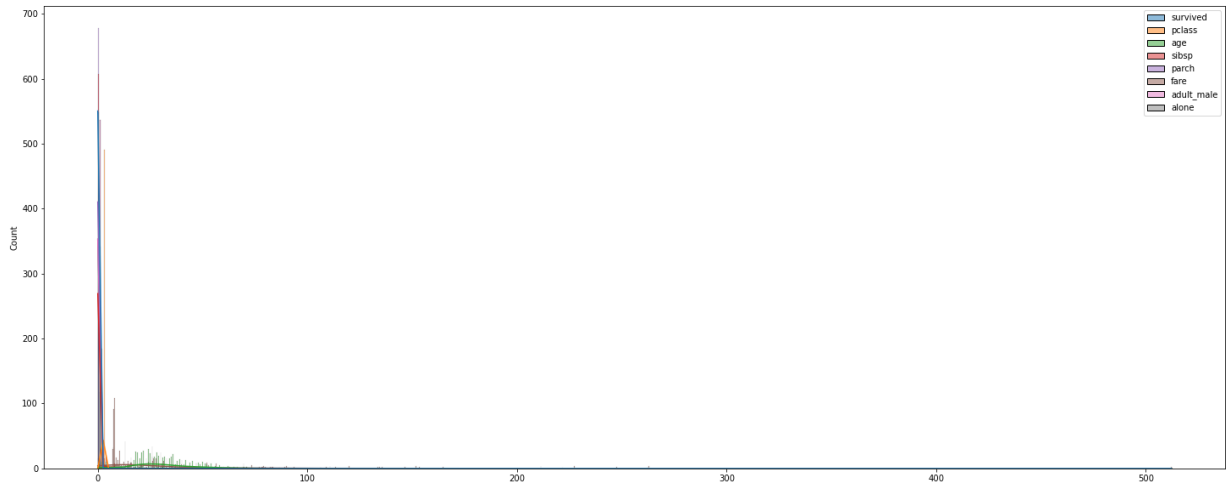
In [103]:
```
sns.histplot(df['fare'],kde=True)
```

Out[103]:
```
<AxesSubplot:xlabel='fare', ylabel='Count'>
```



In [104]:
```
plt.figure(figsize=(25,10))
sns.histplot(df,kde=True)
```

Out[104]: <AxesSubplot:ylabel='Count'>

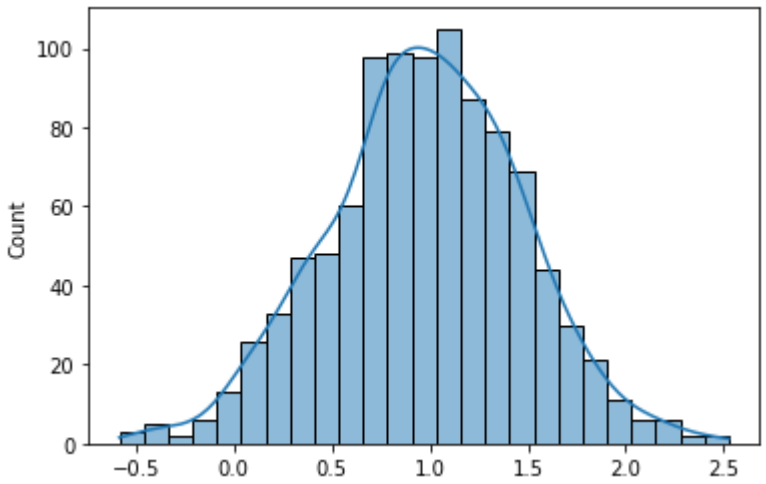


Creat a Normal distribution dataset

In [105]:
```
nd=np.random.normal(1,0.5,1000)  # mean 1, std .5, no of values=1000
```

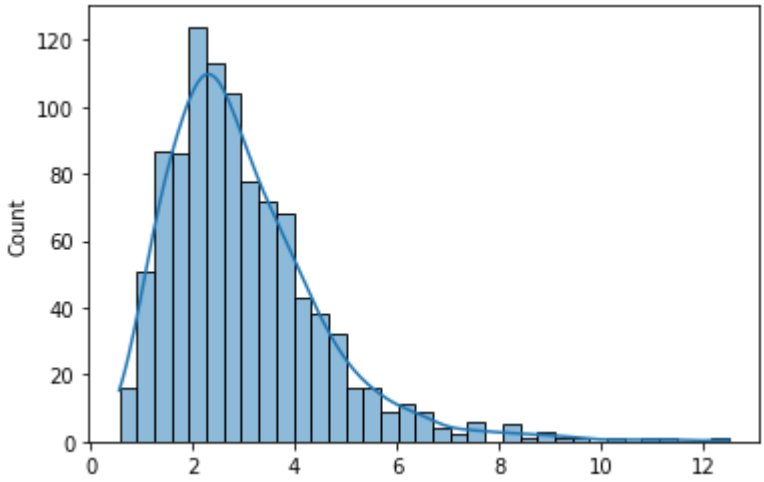
In [106]:
```
sns.histplot(nd,kde=True)  # normal distribution plot
```

Out[106]: <AxesSubplot:ylabel='Count'>



In [107]:
```
sns.histplot(np.exp(nd),kde=True)  # normal distribution convert into lognormal d
```

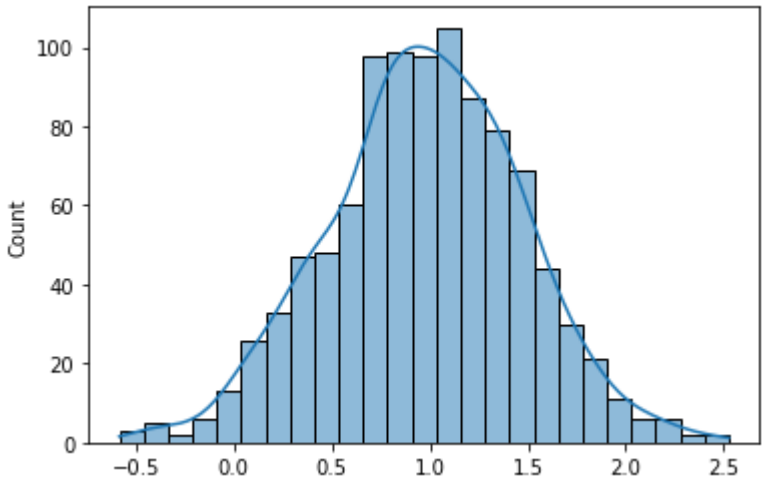
Out[107]: <AxesSubplot:ylabel='Count'>



In [108]:
```
lnd=np.exp(nd)
```

In [109]:
```
sns.histplot(np.log(lnd),kde=True)    # lognormal distribution convert into normd
```

Out[109]: <AxesSubplot:ylabel='Count'>

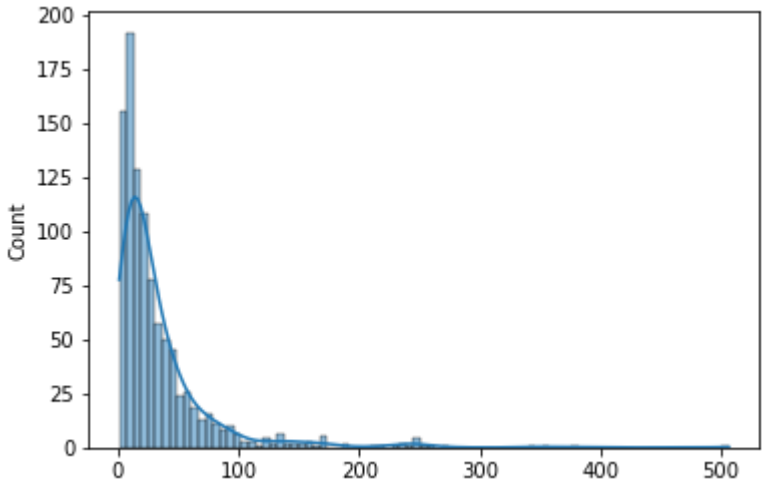


In [110]:
```
lognormal=np.random.lognormal(3,1,1000)
lognormal
```

Out[110]:
```
array([168.17548703,  13.98748008,  51.32728738,  73.18942197,
        57.48266454,  14.58882744,  13.87588109,  65.896135  ,
        25.66974415,  18.36995074,   3.8086528 ,  45.73099284,
         7.50359618,  29.14710172,  86.09491279,  24.17870402,
         2.77031697,  51.89195558,   9.48085055,   7.38463552,
         7.20306459,  25.17440957,   3.83354238,  18.20555149,
       105.98556152,   8.66259106,   5.57219628,  96.26762843,
        32.1306059 ,  19.12632027,   2.27340206,  19.63750687,
        56.13573289,  33.14933314,  12.76617396,  18.89787364,
        20.36378223,  16.41396739,  42.18144945,  15.07208784,
         5.51760352,  11.07452805,  92.4952617 ,   2.54104879,
        14.88131009,  44.30041446,  24.53207257,  26.70667758,
        11.93441954,  19.34112019,  11.38285639,  43.77676396,
        46.60764095,   8.04247876,  23.17308025,  37.95588495,
        10.06108008,  76.17423547,  19.49917879,  14.18369886,
        29.25436297,  19.72418037,  12.40843897,   9.73369037,
        54.13134426,  11.13254916,  21.70062796,   9.47195042,
        18.67291943,   7.76936176,  28.30137542,   8.83329392,
         4.59510078,   7.90930383,  48.152169  ,  24.05553359,
```

In [111]:
```
sns.histplot(lognormal,kde=True)   # lognormal distribution
```

Out[111]:
```
<AxesSubplot:ylabel='Count'>
```



person & spearman rank correlation

In [112]:
```
df=sns.load_dataset('iris')
df
```

Out[112]:

	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
...	...	...	...	...	...
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

150 rows × 5 columns

In [113]:
```
df.isnull().sum()
```

Out[113]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [114]:
```
df.corr()   # finding correlation ( positive value define the same direction and
```

Out[114]:

	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 [115]:
```
df.cov()    # finding covariance
```

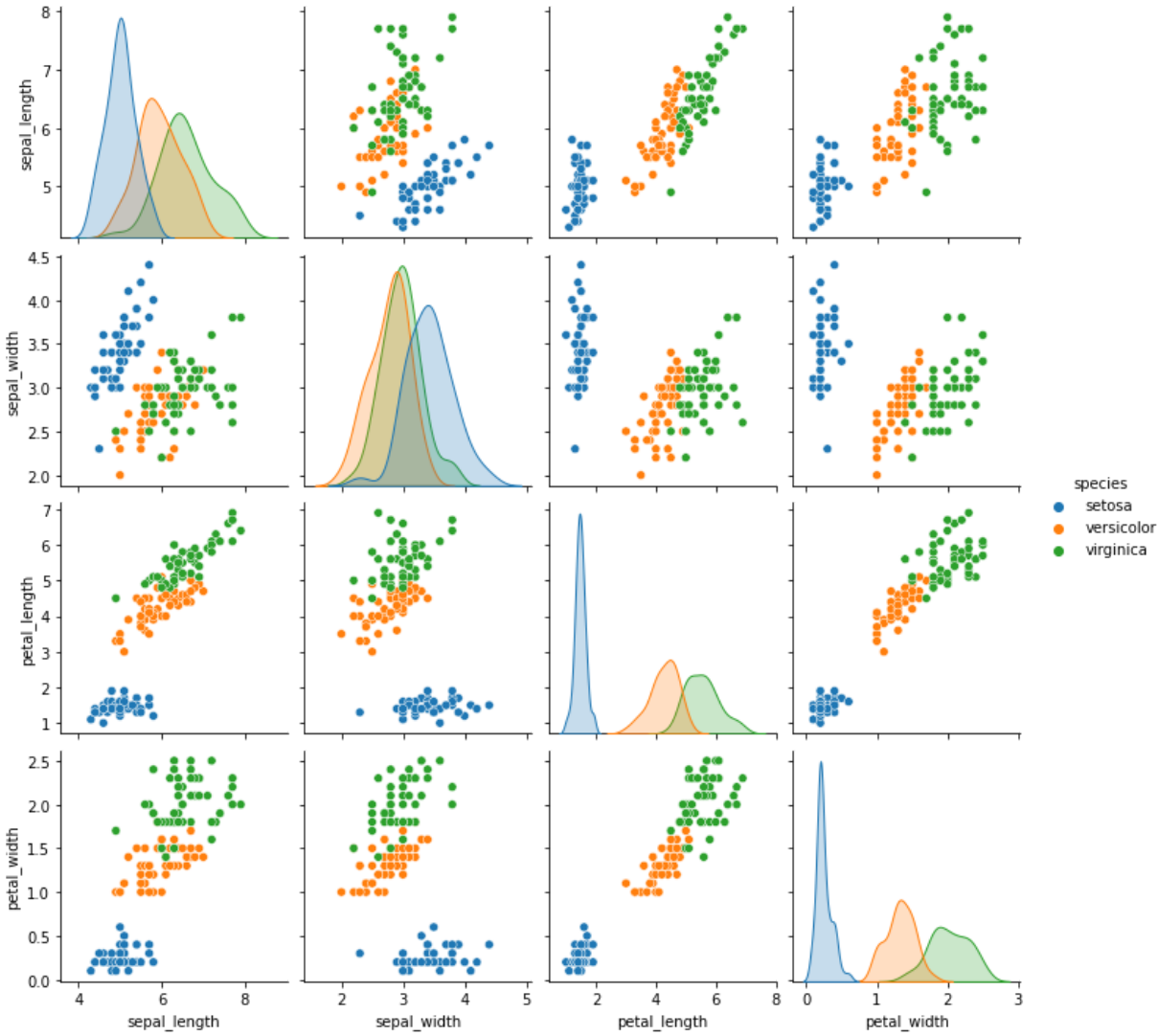
Out[115]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	0.685694	-0.042434	1.274315	0.516271
sepal_width	-0.042434	0.189979	-0.329656	-0.121639
petal_length	1.274315	-0.329656	3.116278	1.295609
petal_width	0.516271	-0.121639	1.295609	0.581006

pair plot

In [117]:
```
sns.pairplot(df,hue="species")
```

Out[117]:
```
<seaborn.axisgrid.PairGrid at 0x286613fc820>
```



In [118]:
```
print('Thank you')
```

```
Thank you
```

In [ ]:
```

```