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

```
df=pd.read_csv("/content/iris.csv")
df
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
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

```
df.head(5)
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
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

```
df.tail(5)
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
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

```
df.shape
```

```
(150, 5)
```

```
df.dtypes
```

```
sepal length (cm)    float64
sepal width (cm)     float64
petal length (cm)    float64
petal width (cm)     float64
target                object
dtype: object
```

```
df.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 (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
 4   target             150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
df.describe()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
min	4.300000	2.000000	1.000000	0.100000
75%	6.400000	3.300000	5.100000	1.800000

```
import seaborn as sb
```

```
import statistics as stat
```

```
stat.mean(df['sepal length (cm)'])
```

```
5.843333333333334
```

```
stat.median(df['sepal length (cm)'])
```

```
5.8
```

```
stat.mode(df['sepal length (cm)'])
```

```
5.0
```

```
np.percentile(df['sepal length (cm)'], 50)
```

```
5.8
```

```
df['sepal length (cm)'].quantile([0.05,0.25,0.5,0.75])
```

```
0.05    4.6
0.25    5.1
0.50    5.8
0.75    6.4
Name: sepal length (cm), dtype: float64
```

```
stat.variance(df['sepal length (cm)'])
```

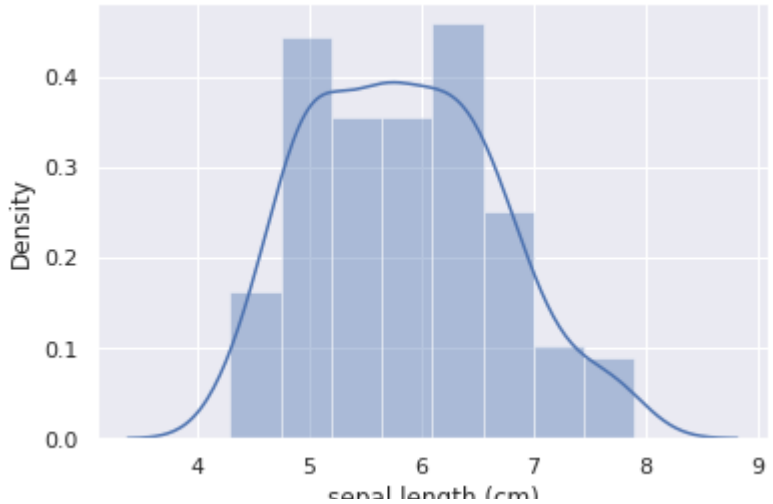
```
0.6856935123042506
```

```
xstd= np.std(df['sepal length (cm)'])
print(xstd)
```

```
0.8253012917851409
```

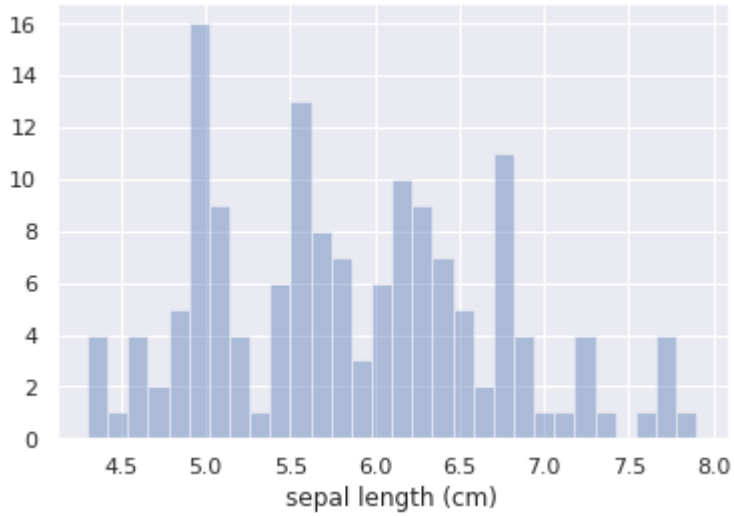
```
sb.set(style='darkgrid')
sb.distplot(df['sepal length (cm)'])
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `di
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f4367e087f0>
```




```
sb.distplot(df['sepal length (cm)'],kde=False,bins=30)
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `di
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f4362ef8d60>
```




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

```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
target               0
dtype: int64
```

```
X=df.drop('target',axis=1)
```

```
Y=df['target']
print(X)
print(Y)
```

```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0                  5.1               3.5                1.4               0.2
1                  4.9               3.0                1.4               0.2
2                  4.7               3.2                1.3               0.2
3                  4.6               3.1                1.5               0.2
4                  5.0               3.6                1.4               0.2
..                 ...               ...                ...               ...
145                6.7               3.0                5.2               2.3
146                6.3               2.5                5.0               1.9
147                6.5               3.0                5.2               2.0
148                6.2               3.4                5.4               2.3
149                5.9               3.0                5.1               1.8

[150 rows x 4 columns]
0         setosa
1         setosa
2         setosa
3         setosa
4         setosa
          ...
145    virginica
146    virginica
147    virginica
148    virginica
149    virginica
Name: target, Length: 150, dtype: object
```

```
data = Y.value_counts()
data
```

```
setosa        50
versicolor    50
virginica     50
Name: target, dtype: int64
```

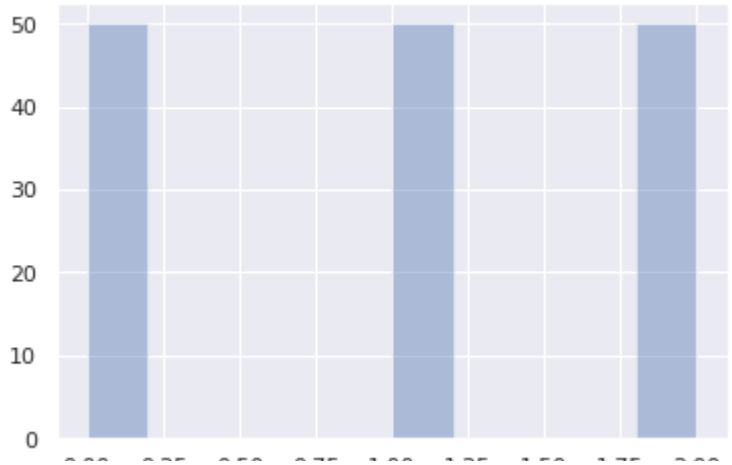
```
from sklearn.preprocessing import LabelEncoder


lb = LabelEncoder()


Y = lb.fit_transform(Y)


sb.set(style='darkgrid')
sb.distplot(Y,kde=False,bins=10)
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `di
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f43625aec40>
```




```
from sklearn.model_selection import train_test_split


X_train, X_test, Y_train, Y_test =train_test_split(X,Y,test_size=0.2)


from sklearn.naive_bayes import GaussianNB


gnb = GaussianNB()
gnb.fit(X_train, Y_train)
```

```
GaussianNB()
```

```
gnb.score(X_test,Y_test)*100
```

```
93.33333333333333
```

```
from sklearn.metrics import accuracy_score



X_train_prediction=gnb.predict(X_train)
training_data_accuracy=accuracy_score(X_train_prediction,Y_train)

print('Accuracy score of the training data: ',training_data_accuracy*100)
```

```
Accuracy score of the training data:  95.83333333333334
```

```
X_test_prediction=gnb.predict(X_test)
testing_data_accuracy=accuracy_score(X_test_prediction,Y_test)

print('Accuracy score of the testing data: ',testing_data_accuracy*100)
```

```
Accuracy score of the testing data:  93.33333333333333
```

Colab paid products - Cancel contracts here

✓ 0s completed at 7:22 PM