In [1]:
```
import pandas as pd
import scipy.stats as stats

std_marks=[34,36,36,38,38,39,39,40,40,41,41,41,41,42,42,45,49,56]
```
executed in 2.17s, finished 16:05:17 2022-03-28

In [2]:
```
std_marks=pd.DataFrame(std_marks)
```
executed in 13ms, finished 16:05:17 2022-03-28

In [3]:
```
std_marks.mean()
```
executed in 16ms, finished 16:05:17 2022-03-28

Out[3]:
```
0    41.0
dtype: float64
```

In [4]:
```
std_marks.describe()
```
executed in 23ms, finished 16:05:17 2022-03-28

Out[4]:

	0
count	18.000000
mean	41.000000
std	5.052664
min	34.000000
25%	38.250000
50%	40.500000
75%	41.750000
max	56.000000

In [5]:
```
std_marks.median()
```
executed in 16ms, finished 16:05:17 2022-03-28

Out[5]:
```
0    40.5
dtype: float64
```

In [6]:
```
std_marks.var()
```
executed in 15ms, finished 16:05:17 2022-03-28

Out[6]:
```
0    25.529412
dtype: float64
```

In [7]:
```
std_marks.std()
```
executed in 8ms, finished 16:05:17 2022-03-28

Out[7]:
```
0    5.052664
dtype: float64
```

In [8]:
```
1 Q9=pd.read_csv("Q9_a.csv")
```
executed in 18ms, finished 16:05:17 2022-03-28

In [9]:
```
1 Q9.skew()
```
executed in 22ms, finished 16:05:17 2022-03-28

Out[9]:
```
Index    0.000000
speed   -0.117510
dist     0.806895
dtype: float64
```

In [10]:
```
1 import seaborn as sns
```
executed in 585ms, finished 16:05:18 2022-03-28

In [11]:
```
1  import matplotlib.pyplot
2  plt.figure(figsize = (6,4))
3
4  plt.hist(Q9["dist"],
5          bins = 10
6          color="#108A99",
7          )
8  plt.title("speed and distance",fontsize = 14)
9  plt.xlabel("speed")
10 plt.yalbel("distance")
11 sns.despine()
12 plt.show()
```
executed in 8ms, finished 16:05:18 2022-03-28

```
SyntaxError: invalid syntax (Temp/ipykernel_40268/4073283644.py, line 6)
```

In [ ]:
```
1 import numpy as np
2 import scipy.stats as st
```
executed in 0ms, finished 16:05:18 2022-03-28

In [ ]:
```
1 #confidence interval for 94%
2 st.t.interval(alpha= 0.97 , df = 2000, loc = 200 , scale = 30)
```
executed in 0ms, finished 16:05:18 2022-03-28

In [ ]:
```
1 #confidence interval for 98%
2 st.t.interval(alpha= 0.99 , df = 2000, loc = 200 , scale = 30)
```
executed in 0ms, finished 16:05:18 2022-03-28

In [ ]:
```
1 # confidence interval for 96%
2 st.t.interval(alpha=0.98 , df = 2000 , loc = 200 , scale = 30)
```
executed in 0ms, finished 16:05:18 2022-03-28

Q.20

cars.csv

In [ ]:
```
1 import seaborn as sns
```
executed in 0ms, finished 16:05:18 2022-03-28

In [ ]:
```
1 cars = pd.read_csv("Cars (1).csv")
```
executed in 0ms, finished 16:05:18 2022-03-28

In [ ]:
```
1 cars
```
executed in 0ms, finished 16:05:18 2022-03-28

In [ ]:
```
1 sns.boxplot(cars.MPG)
```
executed in 0ms, finished 16:05:18 2022-03-28

In [ ]:
```
1 # A) P(mpg>38)
2 print("probablity mpg>38 = ",(1-stats.norm.cdf(38,cars.MPG.mean(),cars.MPG.s
```
executed in 0ms, finished 16:05:18 2022-03-28

In [ ]:
```
1 # B) P(mpg>40)
2 print("probablity mpg>40 = ",(stats.norm.cdf(40,cars.MPG.mean(),cars.MPG.std
```
executed in 0ms, finished 16:05:18 2022-03-28

In [ ]:
```
1 P(20<mpg<50)
t("probablity mpg>50 = ",(stats.norm.cdf(50,cars.MPG.mean(),cars.MPG.std())-stats
```
executed in 0ms, finished 16:05:18 2022-03-28

In [ ]:
```
1 sns.kdeplot(cars["MPG"])
```
executed in 0ms, finished 16:05:18 2022-03-28

21b

waist adipose tissue

In [14]:
```
1 at=pd.read_csv("wc-at.csv")
```
executed in 30ms, finished 16:06:36 2022-03-28

In [16]:
```
1 at
```

executed in 32ms, finished 16:06:54 2022-03-28

Out[16]:

	Waist	AT
0	74.75	25.72
1	72.60	25.89
2	81.80	42.60
3	83.95	42.80
4	74.65	29.84
...	...	...
104	100.10	124.00
105	93.30	62.20
106	101.80	133.00
107	107.90	208.00
108	108.50	208.00

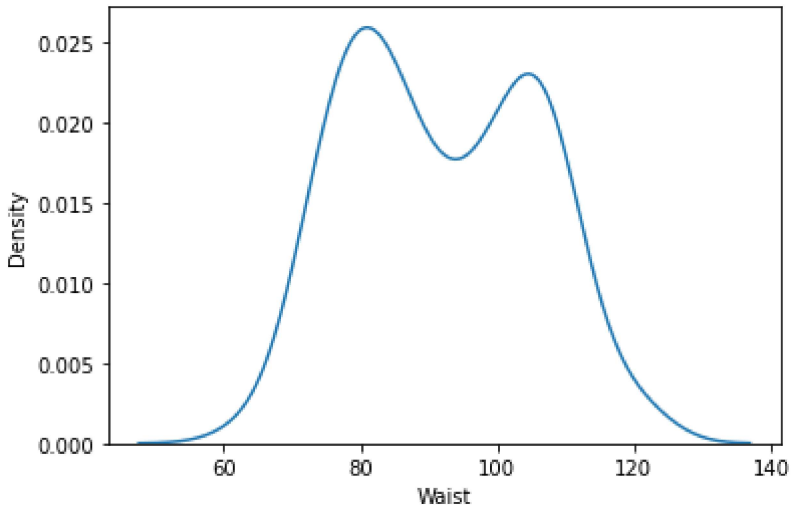
109 rows × 2 columns

In [ ]:
```
1
```

In [20]:
```
1 sns.kdeplot(at["Waist"])
```

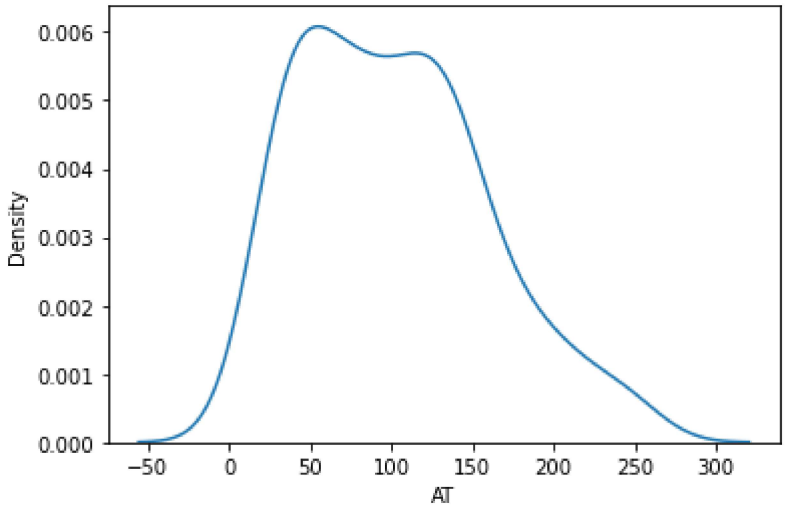
executed in 339ms, finished 16:07:48 2022-03-28

Out[20]:
```
<AxesSubplot:xlabel='Waist', ylabel='Density'>
```



In [21]:
```
1 sns.kdeplot(at["AT"])
```
executed in 377ms, finished 16:08:07 2022-03-28

Out[21]: <AxesSubplot:xlabel='AT', ylabel='Density'>



22)

calculation of z scores

In [22]:
```
1 # z score of 90%
2 stats.norm.ppf(0.95)
```
executed in 18ms, finished 16:16:12 2022-03-28

Out[22]: 1.6448536269514722

In [23]:
```
1 # z scores of 94%
2 stats.norm.ppf(0.97)
```
executed in 15ms, finished 16:16:55 2022-03-28

Out[23]: 1.8807936081512509

In [24]:
```
1 # z scores of 60%
2 stats.norm.ppf(.8)
```
executed in 27ms, finished 16:17:58 2022-03-28

Out[24]: 0.8416212335729143

23)

confidence interval

In [25]:
```
# t score of 95 sample size of 25
stats.t.ppf(0.975,24)
```
executed in 19ms, finished 16:26:15 2022-03-28

Out[25]: 2.0638985616280205

In [26]:
```
# t score of 96 sample size of 25
stats.t.ppf(0.98,24)
```
executed in 18ms, finished 16:27:03 2022-03-28

Out[26]: 2.1715446760080677

In [28]:
```
# t score of 96 sample size of 25
stats.t.ppf(0.995,24)
```
executed in 19ms, finished 16:28:38 2022-03-28

Out[28]: 2.796939504772804

24)

In [32]:
```
t= (260-270)/(90/18**0.5)
t
```
executed in 12ms, finished 16:38:27 2022-03-28

Out[32]: -0.4714045207910317

In [34]:
```
# pvalue =
Pvalue = 1-stats.t.cdf(abs(-0.4714),df=17)
Pvalue
```
executed in 16ms, finished 16:40:34 2022-03-28

Out[34]: 0.32167411684460556