Importing file

In [1]:

```
import numpy as np
import pandas as pd
df=pd.read_csv("Automobile_data.csv")
df
```

Out[1]:

	make	fuel-type	aspiration	body-style	drive-wheels	wheel-base	length	width	height	num-of-cylinders	eng
0	alfa-romero	gas	std	convertible	rwd	88.6	168.8	64.1	48.8	four	
1	alfa-romero	gas	std	convertible	rwd	88.6	168.8	64.1	48.8	four	
2	alfa-romero	gas	std	hatchback	rwd	94.5	171.2	65.5	52.4	six	
3	audi	gas	std	sedan	fwd	99.8	176.6	66.2	54.3	four	
4	audi	gas	std	sedan	4wd	99.4	176.6	66.4	54.3	five	
...	...	...	...	...	...	...	...	...	...	...	
193	volvo	gas	std	sedan	rwd	109.1	188.8	68.9	55.5	four	
194	volvo	gas	turbo	sedan	rwd	109.1	188.8	68.8	55.5	four	
195	volvo	gas	std	sedan	rwd	109.1	188.8	68.9	55.5	six	
196	volvo	diesel	turbo	sedan	rwd	109.1	188.8	68.9	55.5	six	
197	volvo	gas	turbo	sedan	rwd	109.1	188.8	68.9	55.5	four	

198 rows × 15 columns

For One Column only!!

In [2]:

```
data1 = df["engine-size"]
data1
```

Out[2]:

```
0      130
1      130
2      152
3      109
4      136
      ...
193    141
194    141
195    173
196    145
197    141
Name: engine-size, Length: 198, dtype: int64
```

COUNT

In [4]:
```
data1.count()
```
Out[4]:
```
198
```

MIN

In [5]:
```
data1.min()
```
Out[5]:
```
61
```

MAX

In [6]:
```
data1.max()
```
Out[6]:
```
326
```

SUM

In [7]:
```
data1.sum()
```
Out[7]:
```
25415
```

NEGATIVE VALUES

In [9]:
```
print("Negative values in data are :-> ")
data1[data1 < 0]
```
```
Negative values in data are :->
```
Out[9]:
```
Series([], Name: engine-size, dtype: int64)
```

COUNT OF ZERO'S

In [12]:
```
print("Count of Zeros in data are :-> ")
data1[data1 == 0]
```
```
Count of Zeros in data are :->
```
Out[12]:
```
Series([], Name: engine-size, dtype: int64)
```

In [13]:
```
print("Count of Zeros in data are :-> ")
data1[data1 == 0].count()
```
```
Count of Zeros in data are :->
```
Out[13]:
```
0
```

CENTRAL TENDENCY

mean

In [14]:
```
data1.mean()
```

Out[14]:
```
128.35858585858585
```

median

In [15]:
```
data1.median()
```

Out[15]:
```
120.0
```

mode

In [16]:
```
data1.mode()
```

Out[16]:
```
0     92
1    122
Name: engine-size, dtype: int64
```

In [18]:
```
data1.mode()[1]
```

Out[18]:
```
122
```

SPREAD

In [19]:
```
from statistics import variance
```

variance

In [20]:
```
variance(data1)
```

Out[20]:
```
1719.2159411372609
```

standard deviation

In [21]:
```
data1.std()
```

Out[21]:
```
41.46342896019651
```

In [23]:
```
import math
standard_deviation = math.sqrt(variance(data1))
standard_deviation
```

Out[23]:
```
41.46342896019649
```

IQR

In [24]:
```
q1 = np.median(data1[:99])
q3 = np.median(data1[99:])
iqr = q3-q1
```

In [25]:
```
q1
```

Out[25]:
```
119.0
```

In [26]:
```
q3
```

Out[26]:
```
120.0
```

In [27]:
```
iqr
```

Out[27]:
```
1.0
```

In [28]:
```
from scipy.stats import skew
from scipy.stats import kurtosis
```

SKEWNESS

In [30]:
```
print("Skewness is",skew(data1,axis=0,bias=True))
```

```
Skewness is 1.9770271847769776
```

KURTOSIS

In [31]:
```
print("kurtosis is",kurtosis(data1,axis=0,bias=True))
```

```
kurtosis is 5.231952518680528
```

In [32]:
```
data1.skew()
```

Out[32]:
```
1.992151110757244
```

In [33]:
```
data1.kurt()
```

Out[33]:
```
5.397442035317445
```

with range specification

In [35]:
```
x = data1.skew()
print("Skewness = ",x)

if(-0.5 < x < 0.5):
    print("Skewness is Low")
elif( 1 < x < -0.5 or 0.5 < x < 1):
    print("Skewness is Moderate")
elif (x < 1 or x> 1):
    print("Skewness is High")
else:
    print("Not matched")
```

```
Skewness =  1.992151110757244
Skewness is High
```

In [36]:
```
x = data1.kurt()
print("kurtosis = ",x)

if(-0.5 < x < 0.5):
    print("kurtosis is Low")
elif( 1 < x < -0.5 or 0.5 < x < 1):
    print("kurtosis is Moderate")
elif (x < 1 or x> 1):
    print("kurtosis is High")
```

```
else:
    print("Not matched")
```

```
kurtosis =  5.397442035317445
kurtosis is High
```

In [3]:

```
import matplotlib.pyplot as plt
```

For Whole Data

In [4]:

```
df0=pd.read_csv("Automobile_data_new.csv")
df0
```

Out[4]:

	wheel-base	length	width	height	stroke
0	88.6	168.8	64.1	48.8	2.68
1	88.6	168.8	64.1	48.8	2.68
2	94.5	171.2	65.5	52.4	3.47
3	99.8	176.6	66.2	54.3	3.40
4	99.4	176.6	66.4	54.3	3.40
...	...	...	...	...	...
193	109.1	188.8	68.9	55.5	3.15
194	109.1	188.8	68.8	55.5	3.15
195	109.1	188.8	68.9	55.5	2.87
196	109.1	188.8	68.9	55.5	3.40
197	109.1	188.8	68.9	55.5	3.15

198 rows × 5 columns

In [51]:

```
df0.count()
```

Out[51]:

```
wheel-base    198
length        198
width         198
height        198
stroke        198
dtype: int64
```

In [52]:

```
df0.min()
```

Out[52]:

```
wheel-base     86.60
length        141.10
width          60.30
height         47.80
stroke          2.07
dtype: float64
```

In [53]:

```
df0.max()
```

Out[53]:

```
wheel-base    120.90
length        208.10
width          72.30
height         59.80
stroke          4.17
dtype: float64
```

In [54]:

```
df0.sum()
```

Out[54]:

```
wheel-base    19575.40
length        34514.10
width         13053.20
height        10659.20
stroke          644.72
dtype: float64
```

In [55]:

```
print("Negative values in data are :-> ")
df0[df0 < 0]
```

```
Negative values in data are :->
```

Out[55]:

	wheel-base	length	width	height	stroke
0	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...
193	NaN	NaN	NaN	NaN	NaN
194	NaN	NaN	NaN	NaN	NaN
195	NaN	NaN	NaN	NaN	NaN
196	NaN	NaN	NaN	NaN	NaN
197	NaN	NaN	NaN	NaN	NaN

198 rows × 5 columns

In [56]:

```
print("Count of Zeros in data are :-> ")
df0[df0 == 0].count()
```

Out[56]:

```
Count of Zeros in data are :->
wheel-base    0
length        0
width         0
height        0
stroke        0
dtype: int64
```

In [57]:

```
df0.mean()
```

Out[57]:

```
wheel-base     98.865657
length        174.313636
width          65.925253
height         53.834343
stroke          3.256162
dtype: float64
```

In [58]:
```
df0.mode()
```

Out[58]:

	wheel-base	length	width	height	stroke
0	93.7	157.3	63.8	50.8	3.4

In [59]:
```
df0.median()
```

Out[59]:
```
wheel-base     97.00
length        173.20
width          65.50
height         54.10
stroke          3.29
dtype: float64
```

In [69]:
```
variance(df0["wheel-base"])
```

Out[69]:
```
37.10896682561658
```

In [65]:
```
variance(df0["length"])
```

Out[65]:
```
153.59763036455925
```

In [66]:
```
variance(df0["width"])
```

Out[66]:
```
4.690018971440295
```

In [67]:
```
variance(df0["height"])
```

Out[67]:
```
5.7801343383069295
```

In [68]:
```
variance(df0["stroke"])
```

Out[68]:
```
0.10151615648874533
```

In [70]:
```
df0.std()
```

Out[70]:
```
wheel-base     6.091713
length        12.393451
width          2.165645
height         2.404191
stroke         0.318616
dtype: float64
```

In [73]:
```
df0.skew()
```

Out[73]:
```
wheel-base    1.003837
length        0.128112
width         0.894204
height        0.039841
stroke       -0.687836
dtype: float64
```

In [74]:
```
df0.kurt()
```

Out[74]:
```
wheel-base    0.894089
length       -0.094075
width         0.646716
height       -0.375159
stroke        2.036843
dtype: float64
```

```
In [77]: q1 = np.median(df0[:109])
```

```
In [78]: q1
```

```
Out[78]: 65.4
```

```
In [79]: q3 = np.median(df0[109:])
```

```
In [80]: q3
```

```
Out[80]: 65.5
```

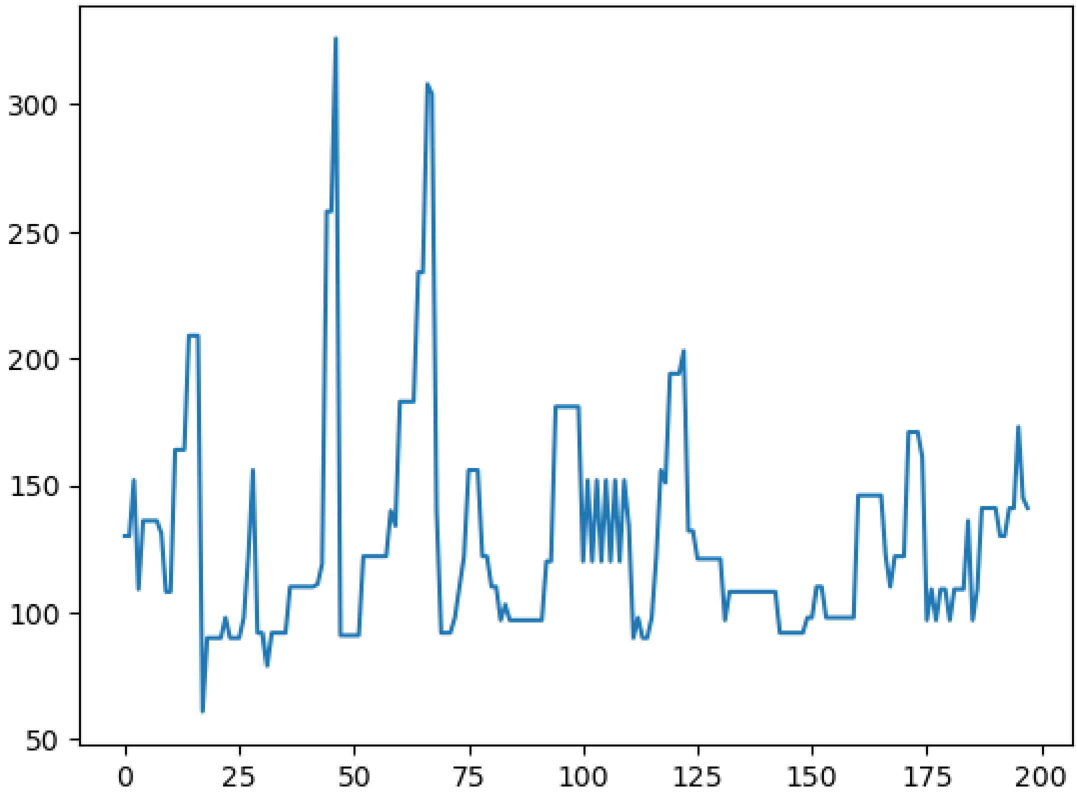
```
In [81]: iqr = q3-q1
         iqr
```

```
Out[81]: 0.09999999999999432
```

Line plot

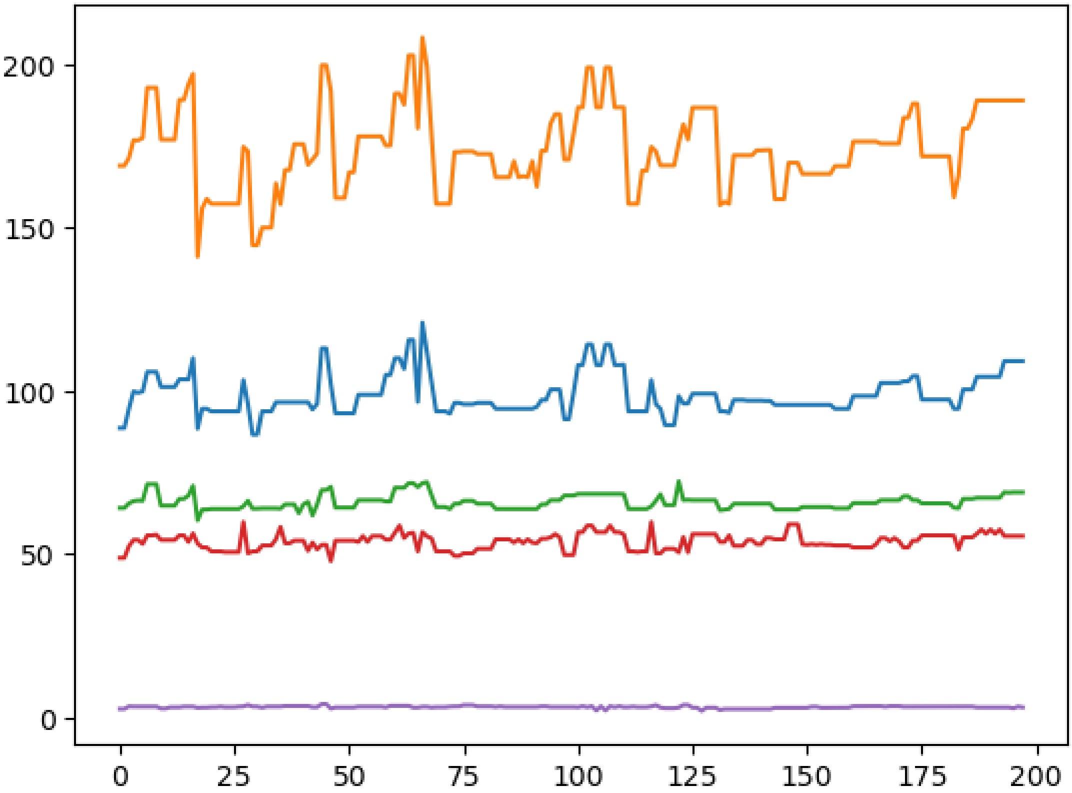
```
In [6]: plt.plot(data1)
        plt.show
```

```
Out[6]: <function matplotlib.pyplot.show(close=None, block=None)>
```



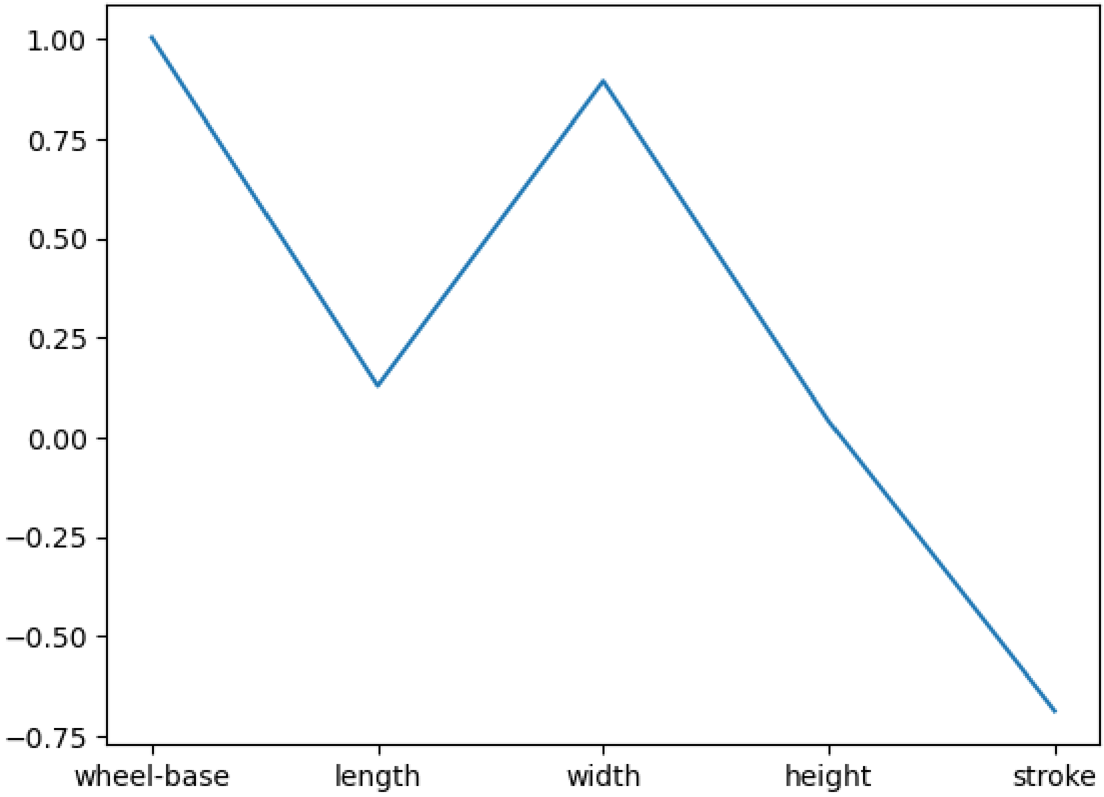
```
In [7]: plt.plot(df0)
        plt.show
```

```
Out[7]: <function matplotlib.pyplot.show(close=None, block=None)>
```



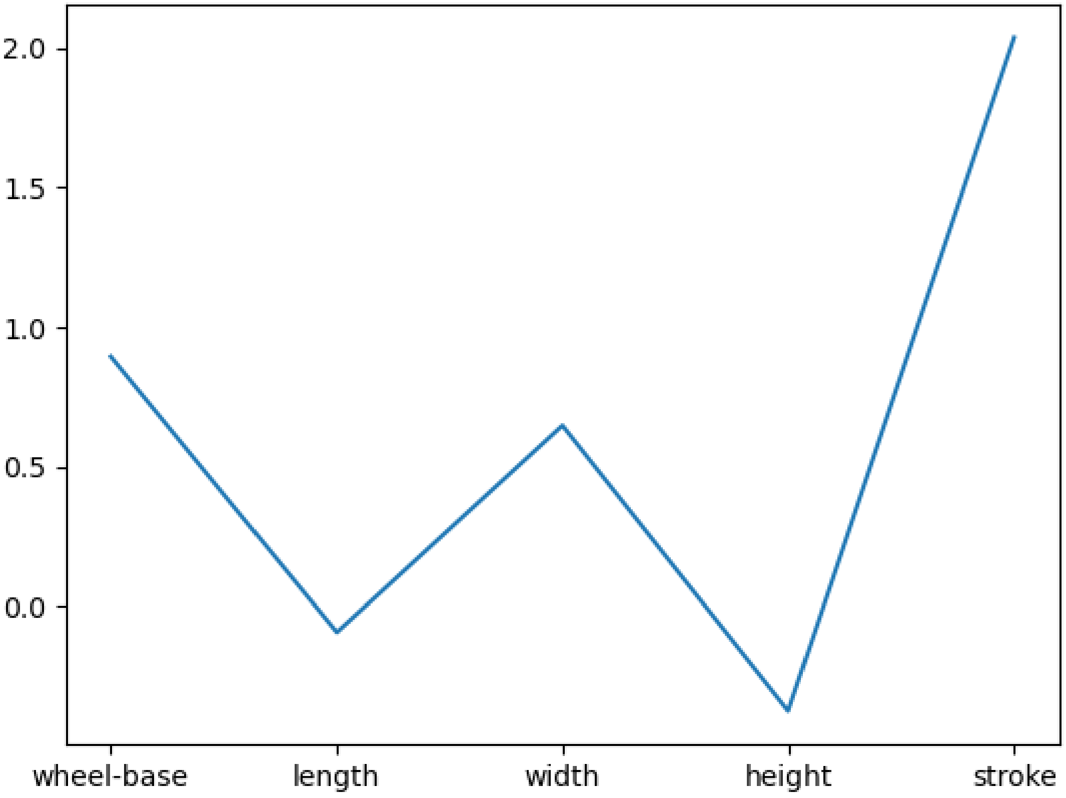
In [9]:
```
plt.plot(df0.skew())
plt.show
```

Out[9]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [10]:
```
plt.plot(df0.kurt())
plt.show
```

Out[10]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Bell Plots

In [13]:
```
def pdf(x):
    mean = np.mean(x)
    std = np.std(x)
    y_out = 1/(std * np.sqrt(2 * np.pi)) * np.exp( - (x - mean)**2 / (2 * std**2))
    return y_out

plt.style.use('seaborn')
plt.figure(figsize = (6, 6))
```

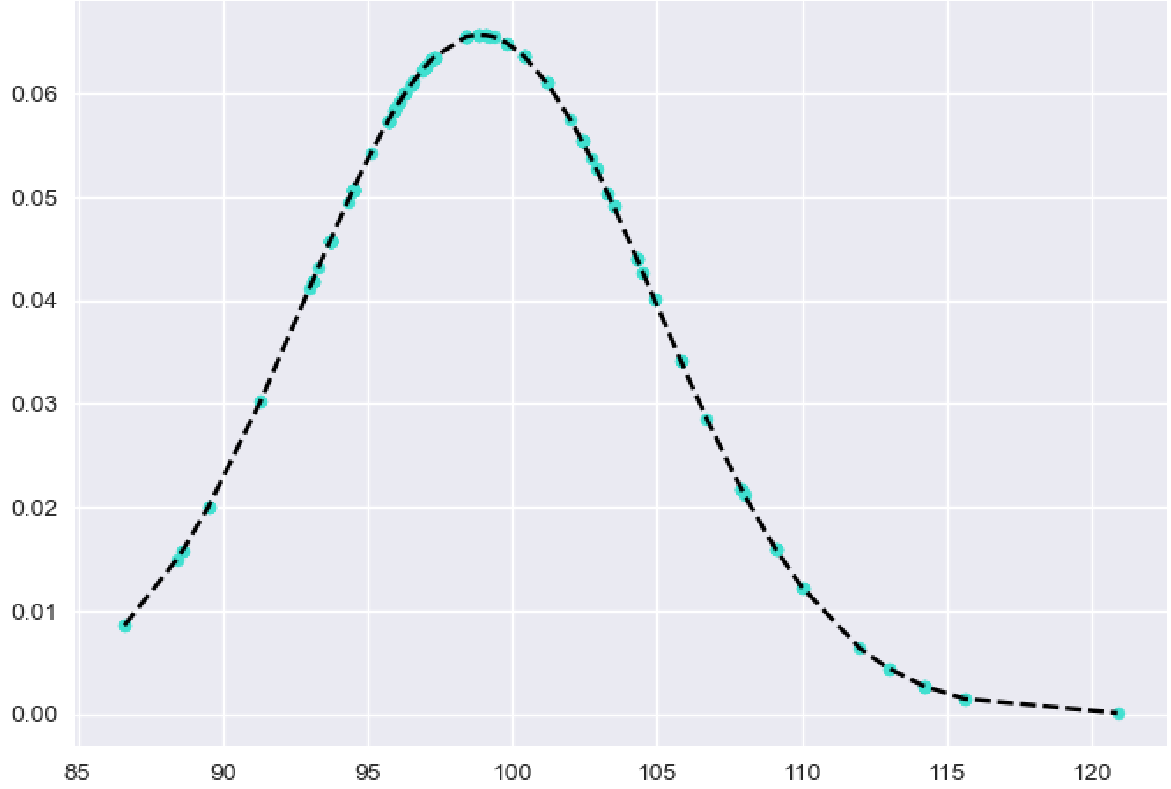
Out[13]:
```
<Figure size 600x600 with 0 Axes>

<Figure size 600x600 with 0 Axes>
```

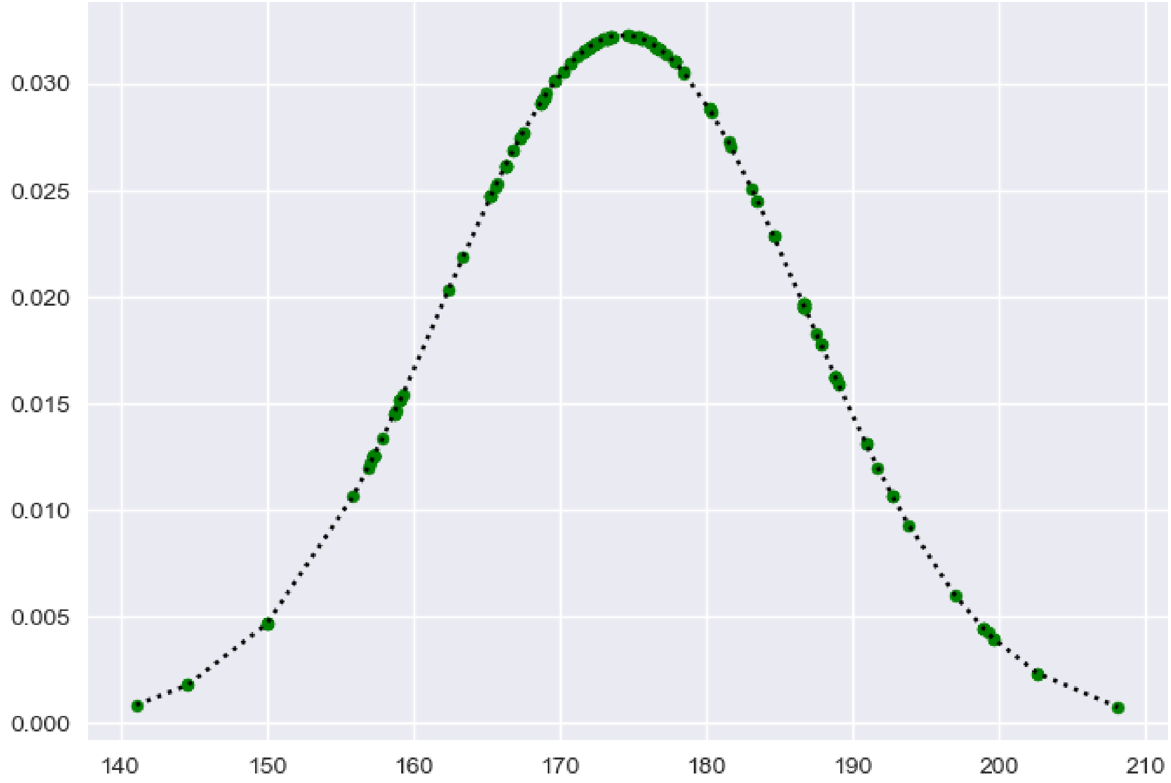
In [22]:
```
a0 = sorted(df0["wheel-base"])
a1 = sorted(df0["length"])
a2 = sorted(df0["width"])
a3 = sorted(df0["height"])
a4 = sorted(df0["stroke"])

b0 = pdf(a0)
b1 = pdf(a1)
b2 = pdf(a2)
b3 = pdf(a3)
b4 = pdf(a4)
```

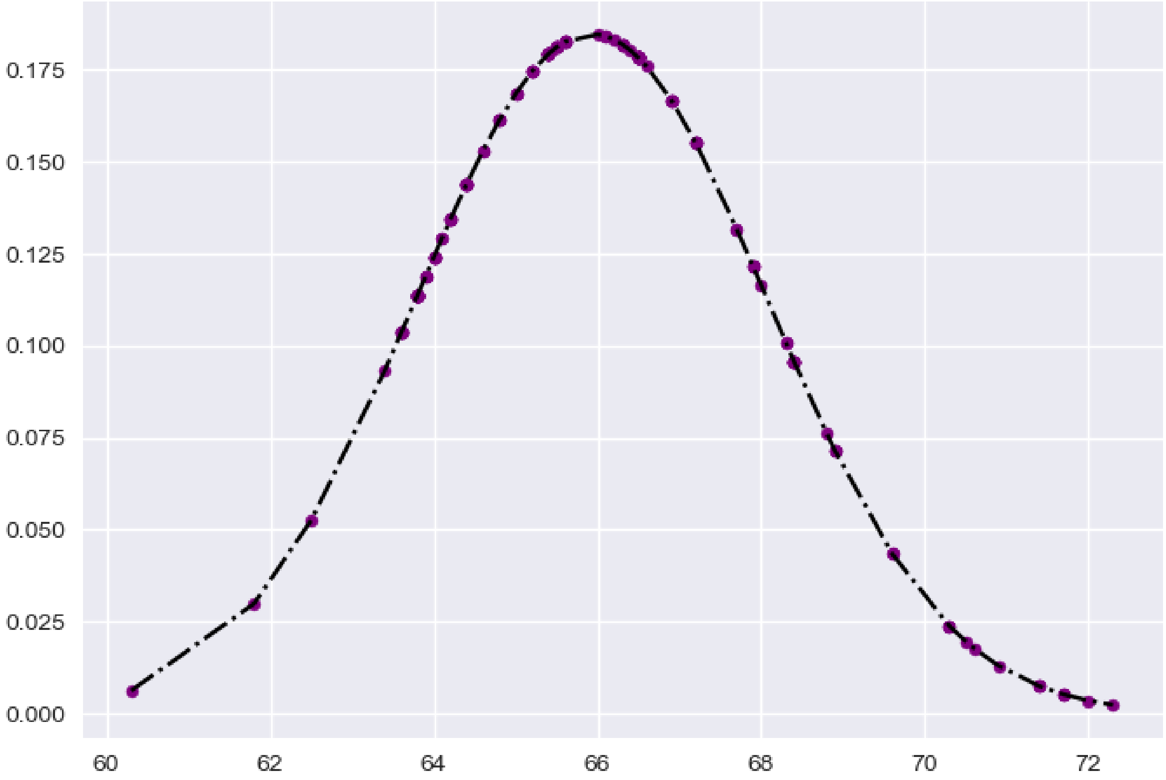
In [21]:
```
plt.plot(a0, b0, color = 'black',linestyle = 'dashed')
plt.scatter( a0, b0, marker = 'o', s = 25, color = 'turquoise')
plt.show()
```



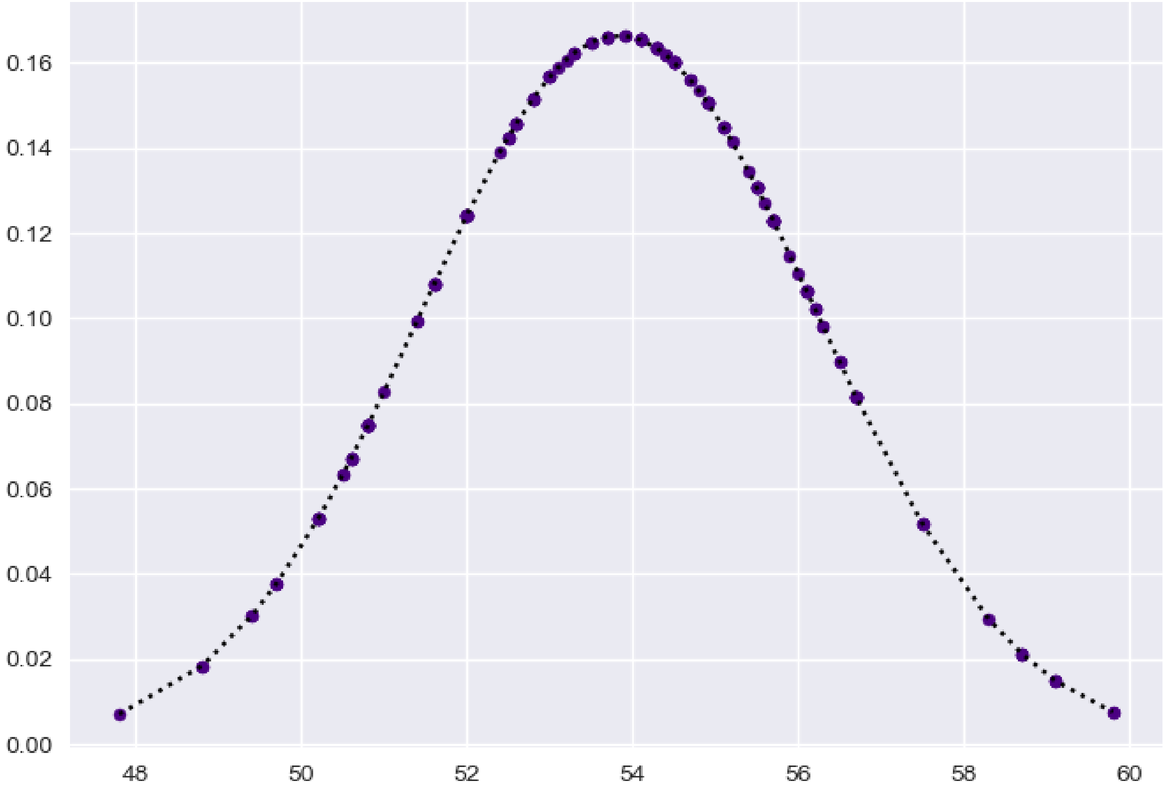
In [23]:
```
plt.plot(a1, b1, color = 'black',linestyle = 'dotted')
plt.scatter( a1, b1, marker = 'o', s = 25, color = 'green')
plt.show()
```



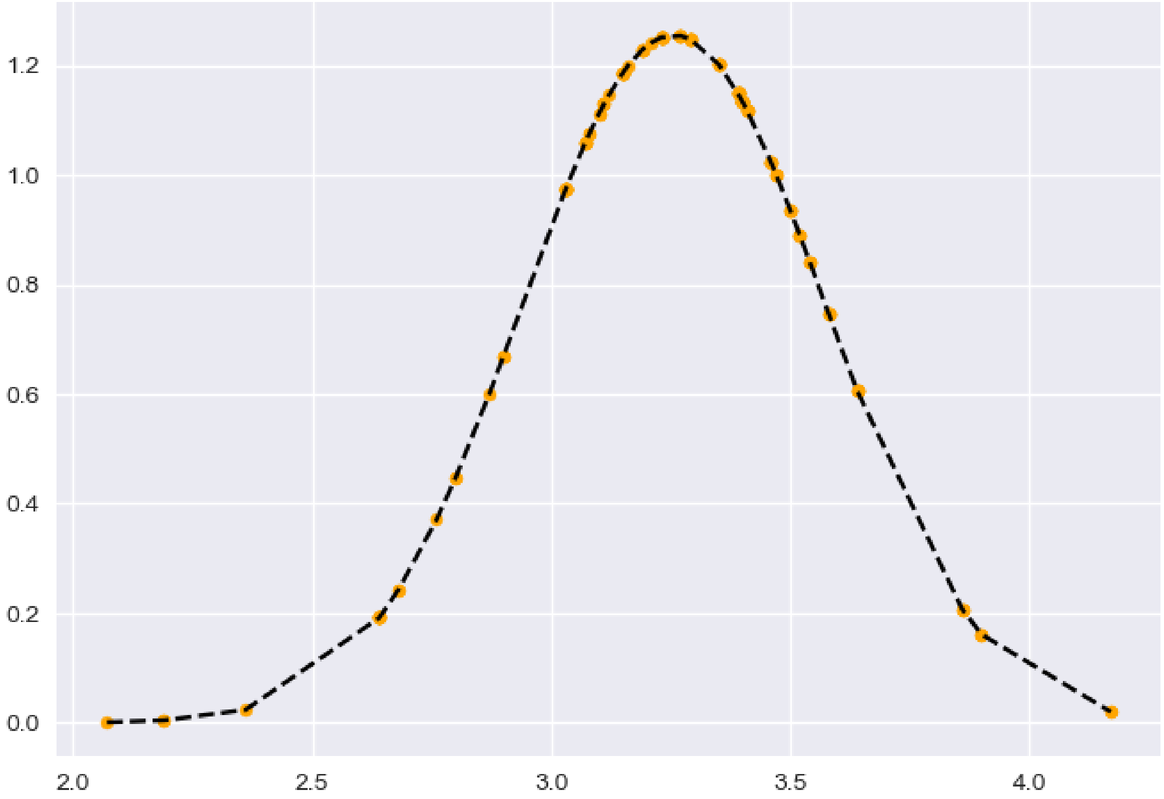
In [34]:
```
plt.plot(a2, b2, color = 'black',linestyle = 'dashdot')
plt.scatter( a2, b2, marker = 'o', s = 25, color = 'purple')
plt.show()
```



In [39]:
```
plt.plot(a3, b3, color = 'black',linestyle = 'dotted')
plt.scatter( a3, b3, marker = 'o', s = 25, color = 'indigo')
plt.show()
```



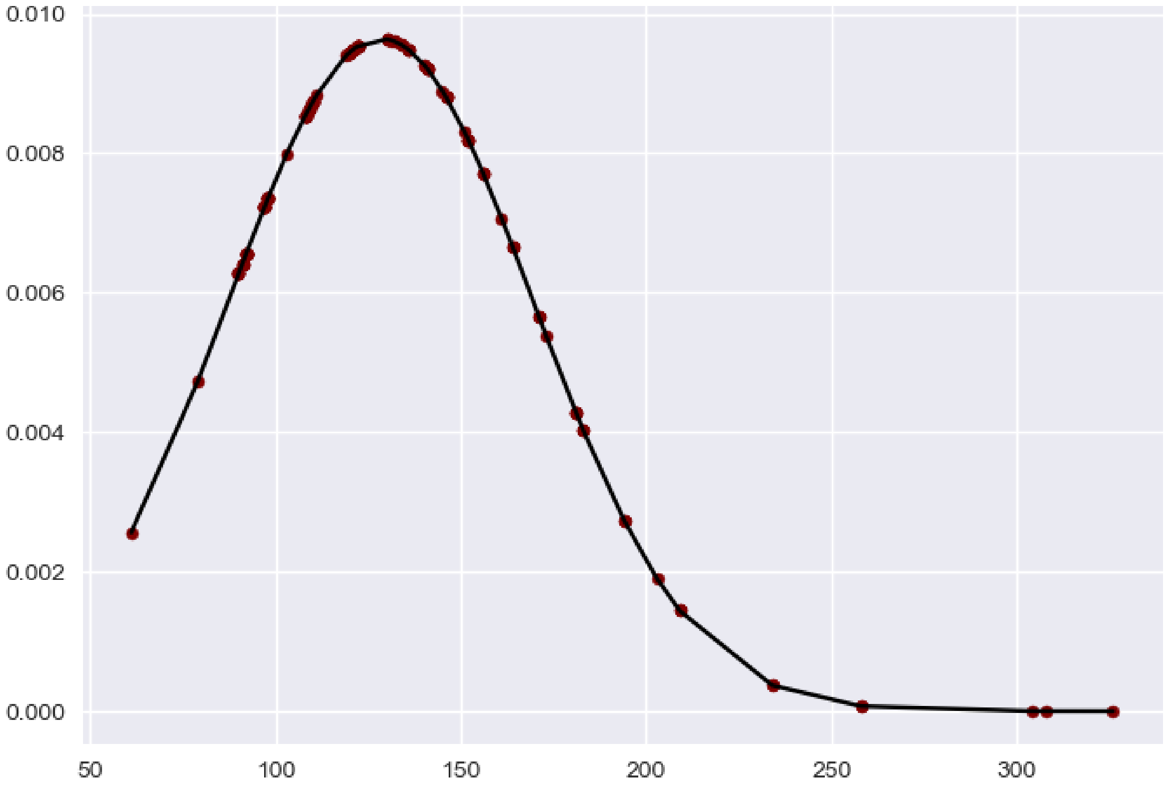
In [28]:
```
plt.plot(a4, b4, color = 'black',linestyle = 'dashed')
plt.scatter( a4, b4, marker = 'o', s = 25, color = 'orange')
plt.show()
```



In [33]:

```
a = np.sort(data1)
b = pdf(a)

plt.plot(a, b, color = 'black',linestyle = 'solid')
plt.scatter( a, b, marker = 'o', s = 25, color = 'maroon')
plt.show()
```


0.010
0.008
0.006
0.004
0.002
0.000
50
100
150
200
250
300