Case Study on Normal Distribution

Global Tech Company (GTC) exports fastener screws to a company in Frankfurt in Germany. GTC manufactures the screws in its factory at Pune. Due to varying weather conditions and wear & tear, the length of the fastener screw is normally distributed. Forthcoming table presents data on the monthly production of the screws.

The importer will buy only those screws with lengths in the interval 2.00 ± 0.02 inch. In other words, the importer wants the length to be 2.00 inch but will accept up to 0.02 inch deviation on either side.

What percentage of the screws will be acceptable to the customer?

In order to improve percentage accepted, the production manager discusses with the production engineers to adjust the machine. What mean value should it be adjusted to?

Suppose the mean cannot be adjusted, but the standard deviation can be reduced. What maximum values of the standard deviation would make 95% and 99% of the parts acceptable to the importer?

Month	Average Length of the Screw (Inches)
1	2.01
2	2.02
3	2.03
4	1.99
5	1.98
6	2.01
7	2.04
8	2.03
9	2.03
10	1.98
11	2.02
12	2.05

13	2.03
14	2.03
15	2.01
16	2.05
17	2.01
18	2.02
19	2.04
20	2.05
21	1.99
22	2.00
23	1.93
24	1.93
25	1.92
26	1.92
27	2.03
28	1.99
29	2.03
30	2.01
31	2.04
32	2.10
33	2.00
34	2.01
35	1.99
Average	Standard Deviation
2.01	0.038632677

Ans) Assumptions:

Programming Language Used - **Python**

The number of screws delivered each month is same and the SD is 0.038632677

Part A)

Analysis of data

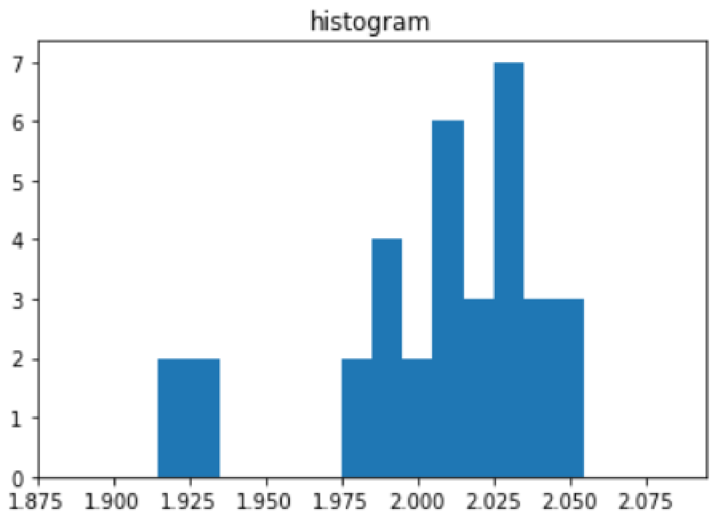
```
In [59]: #list l1 stores the values of average length of screws
         freq_dict = dict()
         for i in l1:
           freq_dict[i] = freq_dict.get(i, 0) + 1
```

```
In [60]: #Frequency distribution
         freq_dict

Out[60]: {2.01: 6,
          2.02: 3,
          2.03: 7,
          1.99: 4,
          1.98: 2,
          2.04: 3,
          2.05: 3,
          2.0: 2,
          1.93: 2,
          1.92: 2,
          2.1: 1}
```

```
In [64]: rnge=[x/100 for x in range(189,210,1) ]
```

```
In [65]: import numpy as np
         import matplotlib.pyplot as plt
         #import seaborn as sns
         data=np.array(t1)
         plt.hist(data, bins =rnge, density=False, align='left')
         plt.title("histogram")
         #sns.distplot(data)
         plt.show()
```



As shown above, the values are not normally distributed.

In order to simplify calculations, we are using mean as 2.01

Percentage of acceptable screws->

```
In [115]: import scipy.stats as st
          sigma=0.038632677
          inc=0.01
          pz1=0.01/sigma
          pz2=0.03/sigma
          val=round(st.norm.cdf(pz1),4)+round(st.norm.cdf(pz2),4)-sn1
          print(val)

          0.38339999999999996
```

sd=0.038632677

Z1 = (2.02-2.01)/sd

Z2 = (2.00-1.98)/sd

P(Z1) + P(Z2) = 0.38334

Part B) To improve the percentage accepted, the mean value should be 2.00 to maintain symmetry

Part C)

Adjust Standard deviation(SD) to get 95% acceptable screws-

Mathematically->

Z1 = (2.02-2.01)/sd

Z2 = (2.00-1.98)/sd

P(Z1) + P(Z2) =0.95

We find SD using the following code->

In [ ]:

```
import scipy.stats as st
sigma=0.0386
inc=0.00001
pz1=0.01/sigma
pz2=0.03/sigma
val=round(st.norm.cdf(pz1),4)+round(st.norm.cdf(pz2),4)-1

while val<0.95:
    sigma=sigma-inc
    pz1=0.01/sigma
    pz2=0.03/sigma
    val=round(st.norm.cdf(pz1),4)+round(st.norm.cdf(pz2),4)-1

print(sigma)
print(val)
#round(st.norm.ppf(0.95),4)
```

n [114]:

```
import scipy.stats as st
sigma=0.00608
inc=0.01
pz1=0.01/sigma
pz2=0.03/sigma
val=round(st.norm.cdf(pz1),4)+round(st.norm.cdf(pz2),4)-1
print(val)
```

```
0.95
```

When **SD=0.00608 , 95%** screws are acceptable

Mathematically->

Z1 = (2.02-2.01)/sd

Z2 = (2.00-1.98)/sd

P(Z1) + P(Z2) = 0.99

```
In [ ]: import scipy.stats as st
        sigma=0.00604
        inc=0.00001
        pz1=0.01/sigma
        pz2=0.03/sigma
        val=round(st.norm.cdf(pz1),4)+round(st.norm.cdf(pz2),4)-1

        while val<0.95:
            sigma=sigma-inc
            pz1=0.01/sigma
            pz2=0.03/sigma
            val=round(st.norm.cdf(pz1),4)+round(st.norm.cdf(pz2),4)-1


        print(sigma)
        print(val)
```

```
In [111]: import scipy.stats as st
          sigma=0.00430
          inc=0.01
          pz1=0.01/sigma
          pz2=0.03/sigma
          val=round(st.norm.cdf(pz1),4)+round(st.norm.cdf(pz2),4)-1
          print(val)

          0.99
```

When **SD=0.00430 , 99%** screws are acceptable

Hence, value of SD should be between **SD=0.00608 and SD=0.00430**