Lab-9

MIL-STD 414

1. An inspector for a military agency desires a variable sampling plan for use with an AQL of 1.5%, assuming that lots are of size 7000. If the standard deviation of the lot or process is unknown, derive a sampling plan using Procedure 1 from MIL-STD 414. How does the sample size find? Compare the sample size with what would have been used under MIL-STD 105E?

Solution:

- AQL=1.5 percent Lot-size=N=7000
- The K technique is implied by Procedure 1. As we are deriving a normal inspection plan, the general inspection level equals IV. Under MIL-STD 414, we first looked at a variable single sample strategy.
- As a result, the Sample size code letter= M for that general inspection level (from table1, MIL-STD-414)
- a) Standard inspection plan: n =50, k=1.80
- b) Strict sampling plan: n =50, k=1.80
- c) Reduced sampling plan: n =20, k=1.51
- Under MIL-STD 105E, we've taken into account a flexible single sample design.
- As we are deriving a normal inspection plan, the general inspection level is equal to II. So, sample size code letter= L for that general inspection level (from table1, MIL-STD-105E)
- a) Standard inspection plan: n =200, Ac =7, Re =8
- b) Tightened sampling plan: n =200, Ac =5, Re =6
- c) Reduced sampling plan: n =80, Ac =5, Re =6
- When we compared MIL-STD 105E and MIL-STD 414, we discovered that MIL-STD 105E has a substantially larger sample size for each sampling strategy than MIL-STD 414.

2. A standard of 0.3 ppm has been established for formaldehyde emission levels in wood products. Suppose that the standard deviation of emissions in an individual board is σ=0.10 ppm. Any lot that contains 1% of its items above 0.3 ppm is considered acceptable. Any lot that has 8% or more of its items above 0.3 ppm is considered unacceptable. Good lots are to be accepted with a probability of 0.95, and bad lots are to be rejected with a probability of 0.90.

- A) Derive a variable sampling plan for this situation.

In [3]:
```
from scipy.stats import norm
norm.ppf(0.05)
```

Out[3]:
```
-1.6448536269514729
```

In [2]:

```
import numpy as np
# sigma = 0.10
# aql = 0.01
# rql = 0.08
# mean = 0.3
# alpha = 0.05
# beta = 0.10
# Variable Sampling for this situation Using the K method
n = ((norm.ppf(0.05) - norm.ppf(0.9))/(norm.ppf(0.01) - norm.ppf(0.08)))**2
k = (norm.ppf(0.9)/np.sqrt(n)) - (norm.ppf(0.08))
print('Using the Z table we have: ')
print('Z alpha : ', norm.ppf(0.05))
print('Z one minus beta : ', norm.ppf(0.9))
print('Z AQL : ', norm.ppf(0.01))
print('Z RQL : ', norm.ppf(0.08))
print(f'Sample size : {n} or {np.ceil(n)}')
print(f'K value : {k} or {np.ceil(k)}')
print()
```

```
Using the Z table we have:
Z alpha :  -1.6448536269514729
Z one minus beta :  1.2815515655446004
Z AQL :  -2.3263478740408408
Z RQL :  -1.4050715603096329
Sample size : 10.089952146199433 or 11.0
K value : 1.808523244030281 or 2.0
```

B) Using the 1% nonconformance level as an AQL, and assuming that lots consist of 5000 panels, find an appropriate set of sampling plans from MIL-STD 414, assuming σ unknown. Compare the sample sizes and the protection that both producer and consumer obtain from this plan with the plan derived in part (A).

- Find a suitable set of sampling plans from MIL-STD 414, assuming unknown, using the 1 percent nonconformance level as an AQL and lots consisting of 5000 panels. Compare the sample sizes and the protection that this provides to both the producer and the customer. Plan based on the plan derived in part (A).
- AQL = 1% = 0.01
- lot size = 5000.
- As we are developing a standard inspection plan, the general inspection level is IV.
- First, we investigated a variable single sampling strategy in accordance with MIL-STD 414. As a result, the Sample size code letter= M for that general inspection level (from table1, MIL-STD-414)
- In this case, we assumed that the variance is unknown.
- a) Normal inspection plan: n =50
- b) Tightened sampling plan: n =50
- c) Reduced sampling plan: n =20

- Conclusion: In this situation, the sample size is larger than the sample size for the derived plan in A.

In [ ]:

```

```