DWDM Tutorial 1

U18CO021: Sahil Bondre

[1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
```

[2]:

```
df = pd.read_csv('Histogram.csv')
df.head()
```

[2]:

```
           A          B          C          D  Left Skew  Multimodal  \
0  48.916926  67.223785  55.917225  45.561471       23.1   37.632318
1  47.692726  68.175751  30.174288  47.825783       18.2   49.244001
2  48.629579  61.753451  43.641583  59.699370       14.6   37.780203
3  58.544034  69.783507  53.738745  45.704638       21.2   56.827208
4  44.821338  70.730153  67.829659  44.254419       24.5   54.513731

         IQ20       IQ100
0  120.459951   93.041368
1  107.418864   93.806158
2   95.006312  135.339681
3   96.522192  100.772632
4  108.878563   91.600053
```

1. Generate the histograms for the frequency of values in the dataset uploaded to the classroom and study statistical characteristics like Mean, Mode, Median, Variance of any sample (Histograms can be generated in Excel/Python/Orange, etc).

[3]:

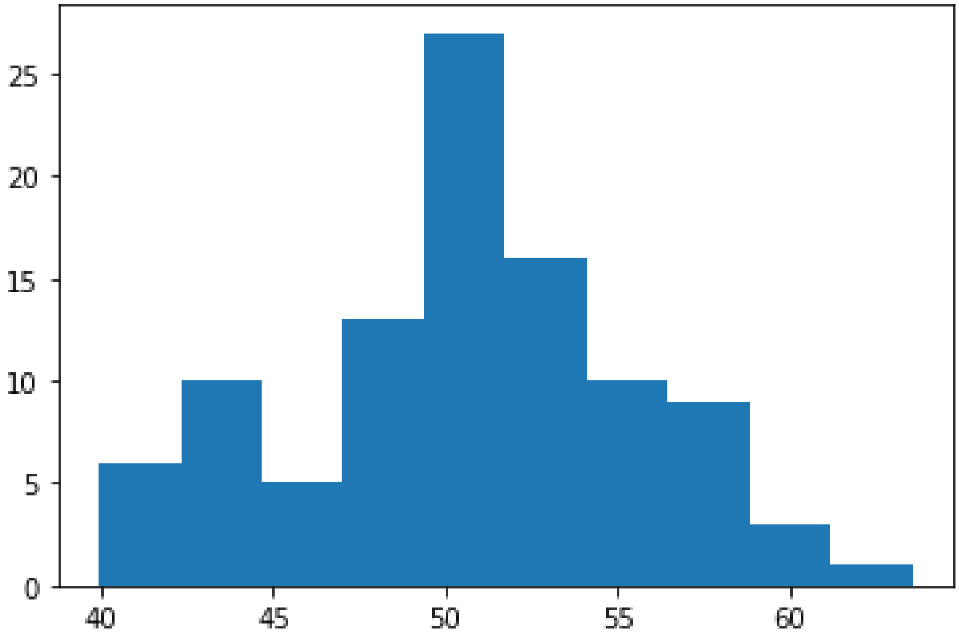
```
df.describe()
```

[3]:

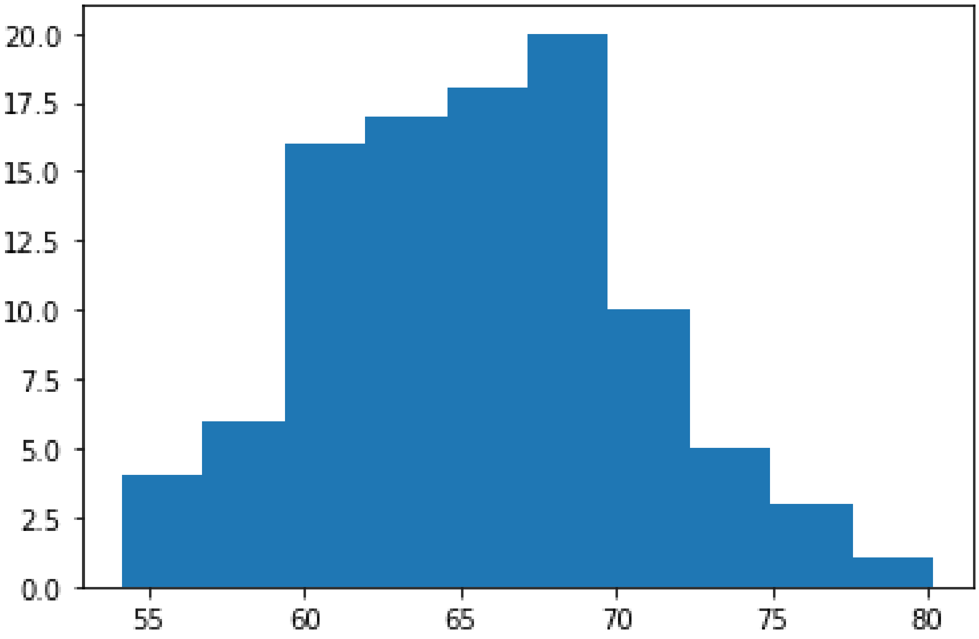
```
                A           B           C           D  Left Skew  Multimodal  \
count  100.000000  100.000000  100.000000  100.000000  92.000000  200.000000
mean    50.632133   65.544513   50.851334   50.211539  20.107609   59.734576
std      5.063123    5.085469   15.342335    5.228720   7.047410   11.513170
min     39.935450   54.142510   15.381702   39.081231   1.000000   33.555815
25%     47.693309   61.819282   42.188371   46.852570  15.025000   49.592572
50%     50.673711   65.898797   51.654882   49.726685  21.500000   60.602041
75%     53.820237   68.821663   61.308291   53.196049  25.925000   69.521137
max     63.531483   80.184730   90.095257   71.200000  31.400000   81.929535
```

```
             IQ20        IQ100
count   20.000000   100.000000
mean   102.132401   102.925179
std     15.550922    15.223586
min     78.284920    69.763146
25%     91.681628    92.096983
50%    105.608402   101.426575
75%    108.952938   114.041076
max    133.448312   138.871933
```

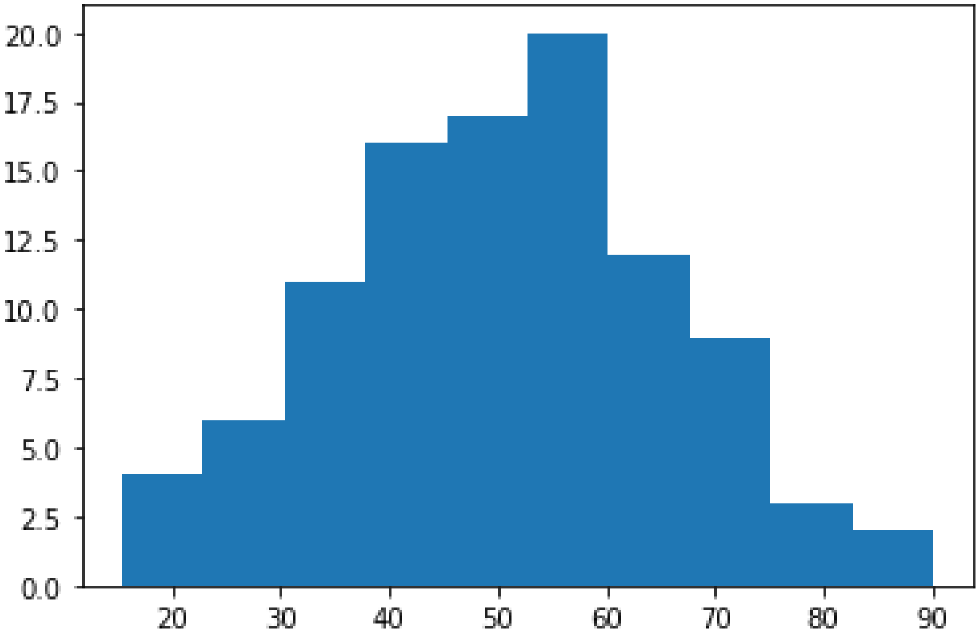
[4]:
```
plt.hist(df['A']);
```



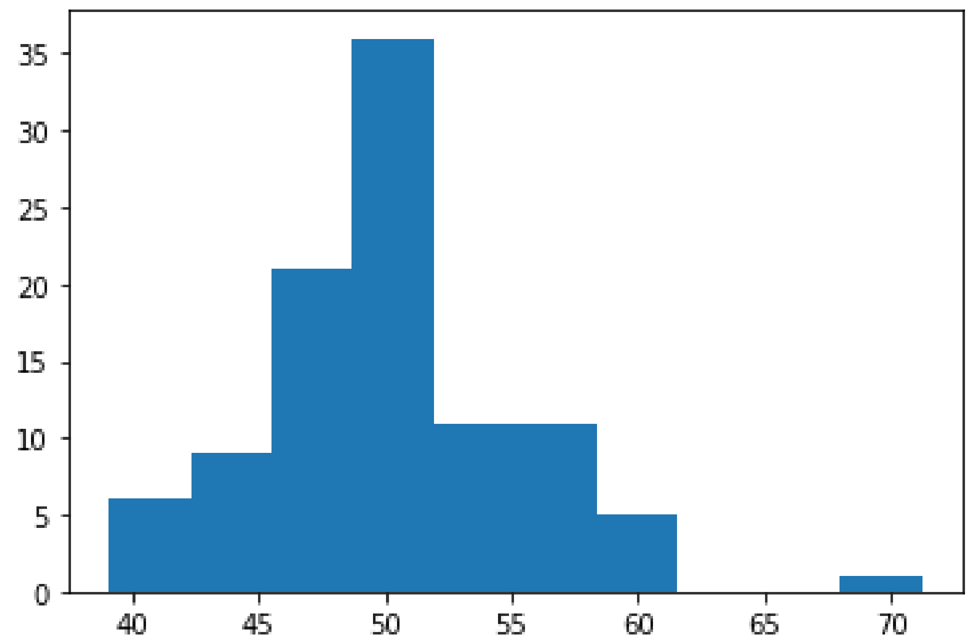
[5]:
```
plt.hist(df['B']);
```



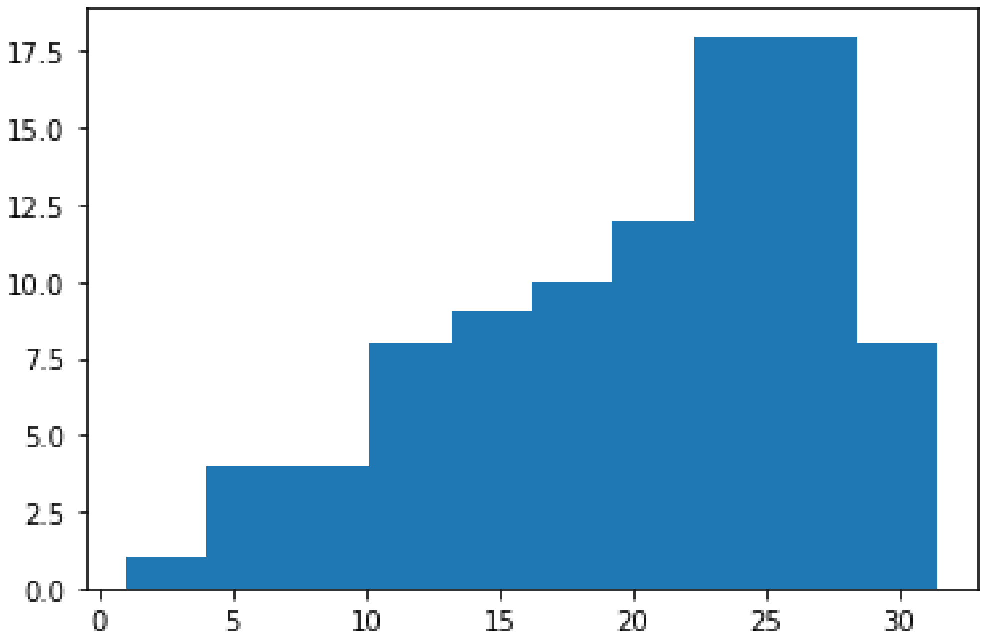
```
[6]: plt.hist(df['C']);
```



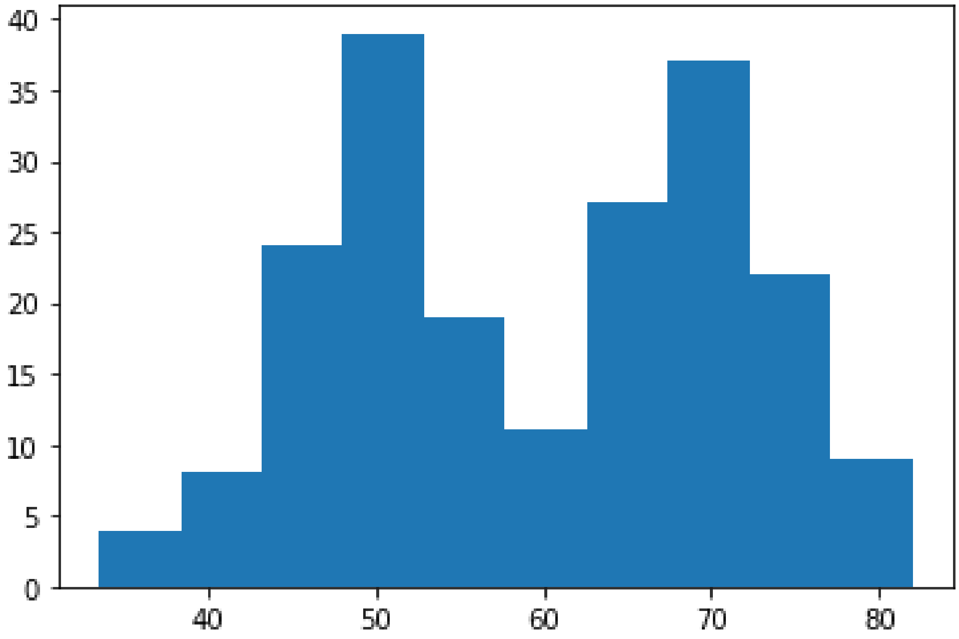
```
[7]: plt.hist(df['D']);
```



```
[8]: plt.hist(df['Left Skew']);
```

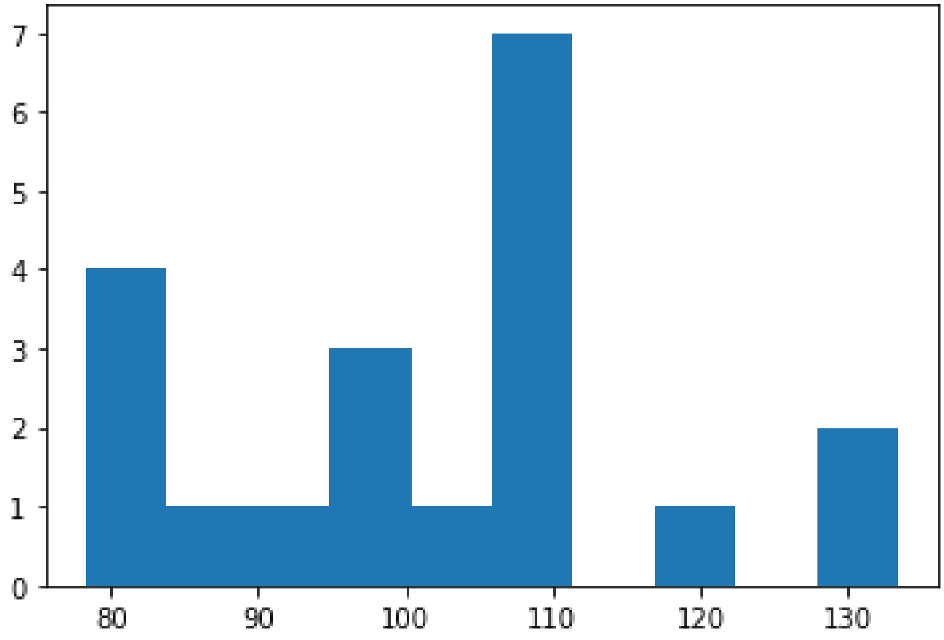


```
[9]: plt.hist(df['Multimodal']);
```



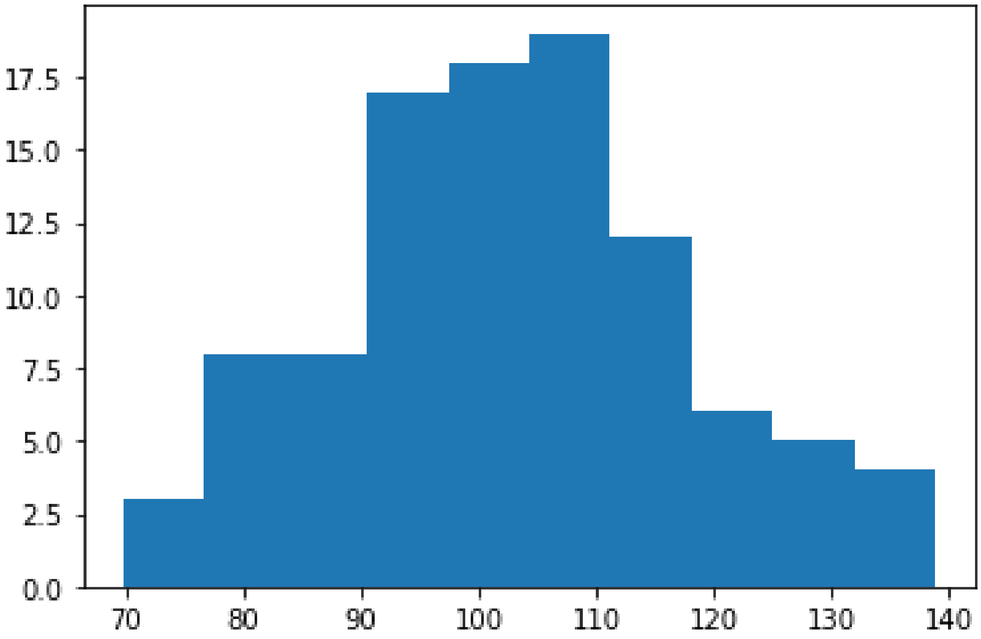
[10]:

```
plt.hist(df['IQ20']);
```



[11]:

```
plt.hist(df['IQ100']);
```



2. Perform skewness analysis for the data and decide the suitable missing value replacement for the ratio scale and interval scale numerical data attributes.

```
[12]: df = pd.read_csv('Histogram.csv')
      df.head()
```

```
[12]:            A          B          C          D  Left Skew  Multimodal  \
      0  48.916926  67.223785  55.917225  45.561471       23.1   37.632318
      1  47.692726  68.175751  30.174288  47.825783       18.2   49.244001
      2  48.629579  61.753451  43.641583  59.699370       14.6   37.780203
      3  58.544034  69.783507  53.738745  45.704638       21.2   56.827208
      4  44.821338  70.730153  67.829659  44.254419       24.5   54.513731

               IQ20       IQ100
      0  120.459951   93.041368
      1  107.418864   93.806158
      2   95.006312  135.339681
      3   96.522192  100.772632
      4  108.878563   91.600053
```

Column A

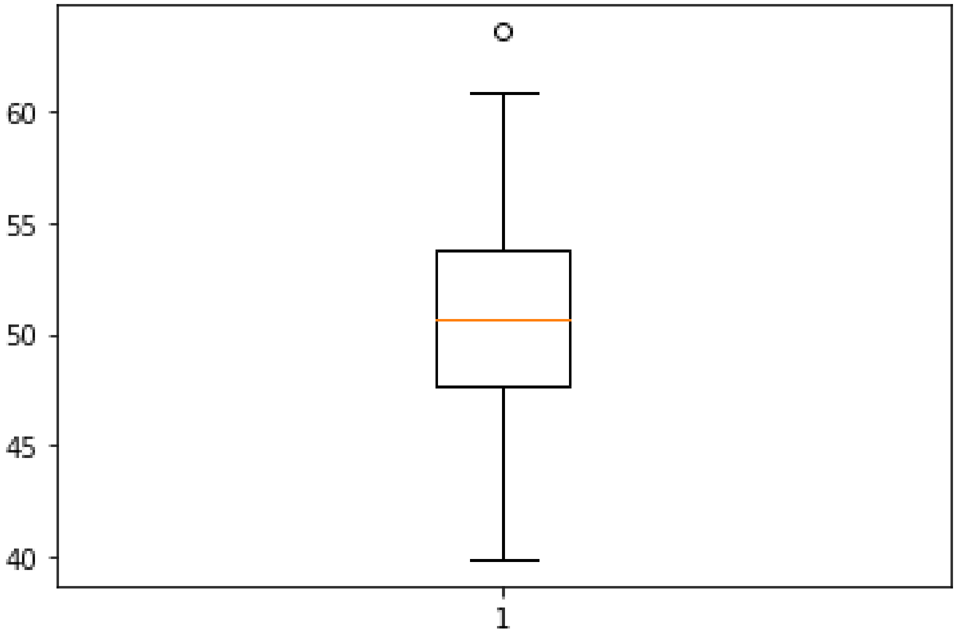
```
[13]: A = df['A'].dropna()
      # insert random N/A with probability 10%
      A = A.mask(np.random.random(A.shape) < .1)
      A.head()
```

```
[13]: 0    48.916926
      1    47.692726
      2    48.629579
      3    58.544034
      4    44.821338
      Name: A, dtype: float64
```

```
[14]: f"Number of N/A values in A: {A.isna().sum()}"
```

```
[14]: 'Number of N/A values in A: 7'
```

```
[15]: plt.boxplot(A.dropna());
```



The median bar is in the center. This is a non-skewed distribution, thus we will replace NA by mean

```
[16]: A.fillna(value=A.mean(), inplace=True)
      f"Number of N/A values in A: {A.isna().sum()}"
```

```
[16]: 'Number of N/A values in A: 0'
```

Column B

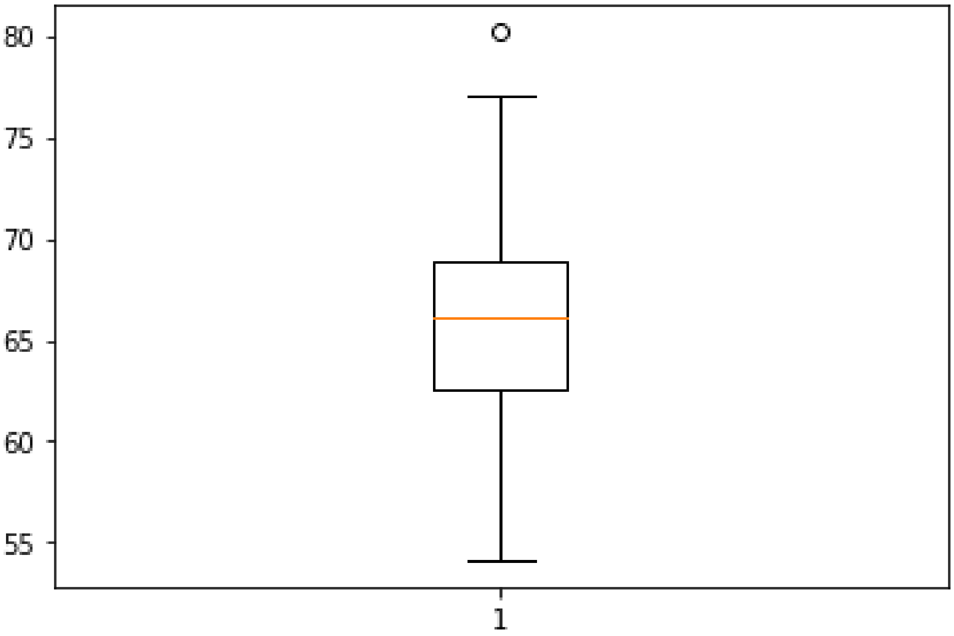
```
[17]: B = df['B'].dropna()
      # insert random N/A with probability 10%
      B = B.mask(np.random.random(B.shape) < .1)
      B.head()
```

```
[17]: 0    67.223785
      1    68.175751
      2    61.753451
      3          NaN
      4    70.730153
      Name: B, dtype: float64
```

```
[18]: f"Number of N/A values in B: {B.isna().sum()}"
```

```
[18]: 'Number of N/A values in B: 9'
```

```
[19]: plt.boxplot(B.dropna());
```



Median is above the center so the data is skewed left. We will fill NA values by median.

```
[20]: B.fillna(value=B.median(), inplace=True)
      f"Number of N/A values in B: {B.isna().sum()}"
```

```
[20]: 'Number of N/A values in B: 0'
```

Column C

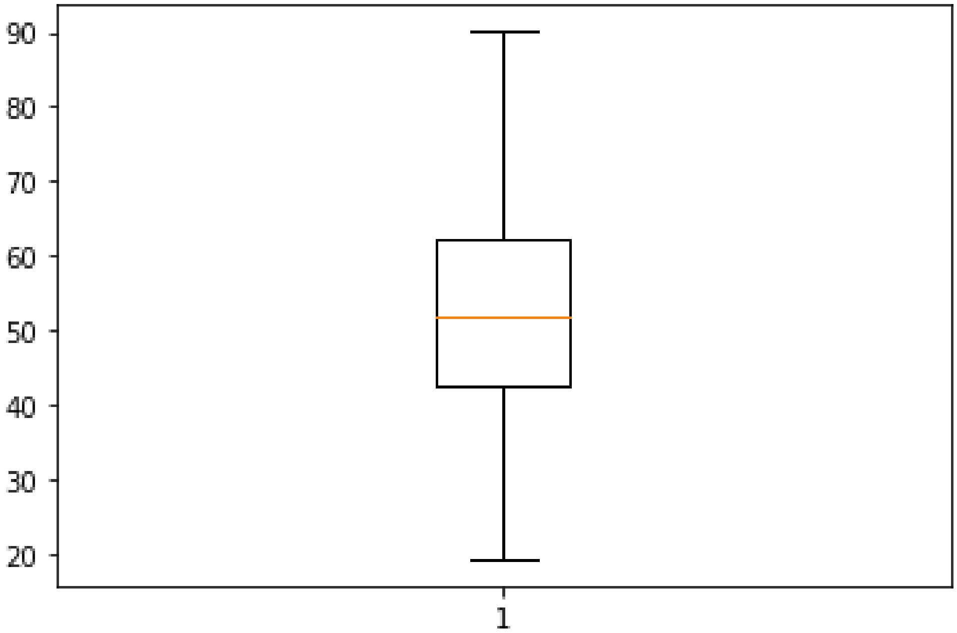
```
[21]: C = df['C'].dropna()
      # insert random N/A with probability 10%
      C = C.mask(np.random.random(C.shape) < .1)
      C.head()
```

```
[21]: 0    55.917225
      1    30.174288
      2    43.641583
      3    53.738745
      4    67.829659
      Name: C, dtype: float64
```

```
[22]: f"Number of N/A values in C: {C.isna().sum()}"
```

```
[22]: 'Number of N/A values in C: 10'
```

```
[23]: plt.boxplot(C.dropna());
```



The median bar is in the center. This is a non-skewed distribution, thus we will replace NA by mean

```
[24]: C.fillna(value=C.mean(), inplace=True)
      f"Number of N/A values in C: {C.isna().sum()}"
```

```
[24]: 'Number of N/A values in C: 0'
```

Column D

```
[25]: D = df['D'].dropna()
      # insert random N/A with probability 10%
      D = D.mask(np.random.random(D.shape) < .1)
      D.head()
```

[25]:

```
0    45.561471
1    47.825783
2    59.699370
3          NaN
4    44.254419
Name: D, dtype: float64
```

[26]:

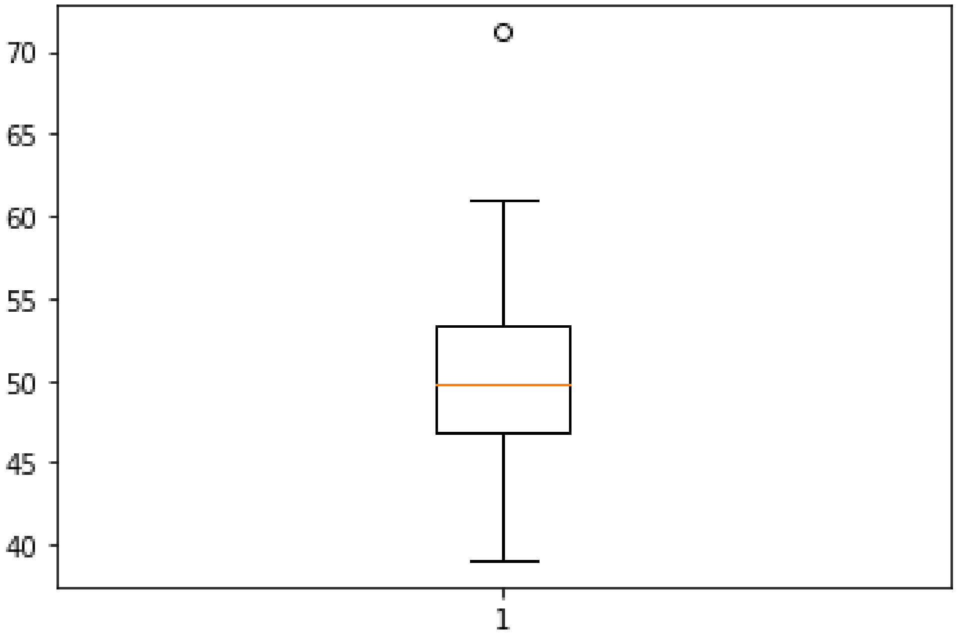
```
f"Number of N/A values in D: {D.isna().sum()}"
```

[26]:

```
'Number of N/A values in D: 13'
```

[27]:

```
plt.boxplot(D.dropna());
```



The median bar is in the center. This is a non-skewed distribution, thus we will replace NA by mean

[28]:

```
D.fillna(value=D.mean(), inplace=True)
f"Number of N/A values in D: {D.isna().sum()}"
```

[28]:

```
'Number of N/A values in D: 0'
```

3. Perform Missing value replacement by Mean, Mode, Median on the A attributes. Intentionally remove two values from that attribute and find the value of the X and Y for given data using mean value replacement (perform the operation on first 12 records).

[29]:

```
df = pd.read_csv('Histogram.csv')
data = df['A'].iloc[0:12]
```

```
data
```

[29]:

```
0     48.916926
1     47.692726
2     48.629579
3     58.544034
4     44.821338
5     47.693504
6     43.954434
7     52.849055
8     47.934716
9     63.531483
10    49.804099
11    52.183024
Name: A, dtype: float64
```

[30]:

```
# 2 NA
x = data[1]
y = data[7]
data[1] = np.nan
data[7] = np.nan
data
```

[30]:

```
0     48.916926
1           NaN
2     48.629579
3     58.544034
4     44.821338
5     47.693504
6     43.954434
7           NaN
8     47.934716
9     63.531483
10    49.804099
11    52.183024
Name: A, dtype: float64
```

[31]:

```
mean = data.mean()
median = data.median()
mode = data.mode()[data.size / 2]
[mean, median, mode]
```

[31]:

```
[50.601313826, 48.773252885, 49.80409903]
```

[32]:

```
data_mean = data.fillna(value=mean)
data_mean
```

[32]:
```
0     48.916926
1     50.601314
2     48.629579
3     58.544034
4     44.821338
5     47.693504
6     43.954434
7     50.601314
8     47.934716
9     63.531483
10    49.804099
11    52.183024
Name: A, dtype: float64
```

[33]:
```
data_median = data.fillna(value=median)
data_median
```

[33]:
```
0     48.916926
1     48.773253
2     48.629579
3     58.544034
4     44.821338
5     47.693504
6     43.954434
7     48.773253
8     47.934716
9     63.531483
10    49.804099
11    52.183024
Name: A, dtype: float64
```

[34]:
```
data_mode = data.fillna(value=mode)
data_mode
```

[34]:
```
0     48.916926
1     49.804099
2     48.629579
3     58.544034
4     44.821338
5     47.693504
6     43.954434
7     49.804099
8     47.934716
9     63.531483
10    49.804099
11    52.183024
Name: A, dtype: float64
```

[35]:

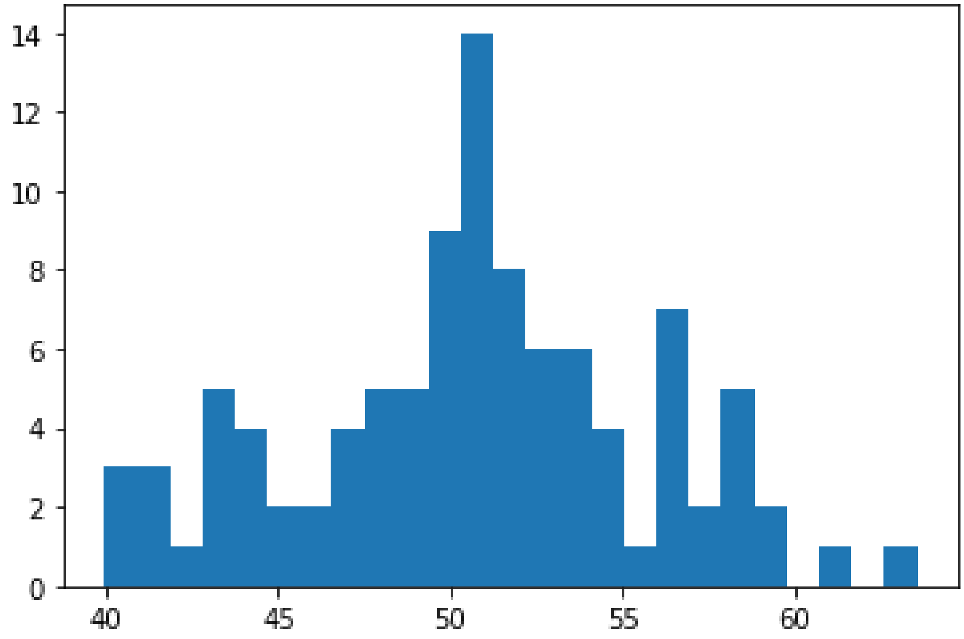
```
new_x = data_mean[1]
new_y = data_mean[7]
change_x = abs(x - new_x) * 100 / x
change_y = abs(y - new_y) * 100 / y
print('Change in x: %0.2f%%' % (change_x))
print('Change in y: %0.2f%%' % (change_y))
```

```
Change in x: 6.10%
Change in y: 4.25%
```

4. Perform Noise identification, Outlier detection using histogram and try to remove the outliers and check the statistical characteristics again

[36]:

```
df = pd.read_csv('Histogram.csv')
plt.hist(df['A'], 25);
```



Note the outliers at 40 with extra bars and at 60 with isolated bars

[37]:

```
new_data = []
data = df['A'].to_numpy()
rows = df.shape[0]
for row in range(rows):
    if data[row] <= 60 and data[row] >= 40:
        new_data.append(data[row])
new_data = np.asarray(new_data)
new_df = pd.DataFrame(new_data.reshape(-1, 1), columns=['A'])
new_df.describe()
```

```
[37]:                A
      count  97.000000
      mean   50.504802
      std     4.735186
      min    40.206318
      25%    47.693504
      50%    50.656375
      75%    53.414551
      max    59.630779
```