Mean Absolute Deviation

Deviation primarily implies the differnece between observed value and estimate of location (mean of the data points). Mean absolute deviation is nothing but finding the differenece of each data point with the mean of data and finding the average of all these differences . This average is called mean absolute deviation or 'MAD'. Let's take an example of a exam score sheet.

In [22]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
score = pd.DataFrame(data = [['harry',68],
                             ['barry',75],
                             ['larry',74],
                             ['cherry',79],
                             ['ron',71],
                             ['park',73]], columns = ['Name','Score'])
```

In [3]:

```
score.head()
```

Out[3]:

	Name	Score
0	harry	68
1	barry	75
2	larry	74
3	cherry	79
4	ron	71

In [4]:

```
# first let's find the mean of the score
score_avg = round(np.mean(score["Score"]))
print(f'the average of the scores here is {score_avg}')
```

```
the average of the scores here is 73
```

In [5]:

```
# Now let's find all the differences of data points with average extracted above
# Then using a for loop let's take all these differences into a list
# we will take absolute values here as few data points may come out as negative
diff_lst = []
for i in score.Score:
    diff_lst.append(abs(score_avg-i))
```

In [6]:

```
diff_lst
```

Out[6]:

```
[5, 2, 1, 6, 2, 0]
```

In [7]:

```
# Now let's find the average of all these differences
print(f'the mean absolute deviation is {round(np.mean(diff_lst))}')
```

```
the mean absolute deviation is 3
```

In [8]:

```
# Now let's try the MAD function given by pandas as well
print(f'the mean absolute deviation is {round(score["Score"].mad())}')
```

```
the mean absolute deviation is 3
```

Standard Deviation

Now the MAD concept is not useful always , because there are times when we get the same mad score even if one of my data set is widely spread and the other dataset is not. let's try to understand this by plotting the previously created data and one more data that we will create now

In [9]:

```
score1 = pd.DataFrame(data = [['harry',83],
                              ['barry',63],
                              ['larry',70],
                              ['cherry',70],
                              ['ron',70],
                              ['park',70]], columns = ['Name','Score'])
```

In [10]:

```
score1_avg = round(np.mean(score1["Score"]))
print(f'the average of the scores here is {score_avg}')
```

```
the average of the scores here is 73
```

In [11]:

```
# Let's try to find out the mad score for new dataset as well
print(f'the mean absolute deviation is {round(score1["Score"].mad())}')
```

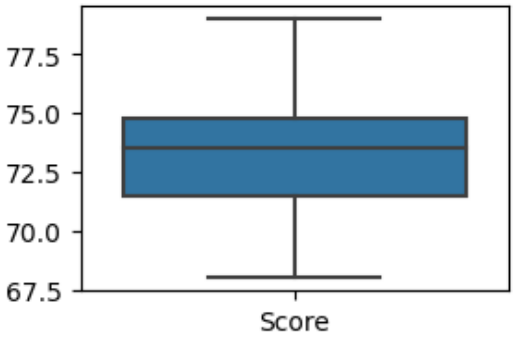
```
the mean absolute deviation is 4
```

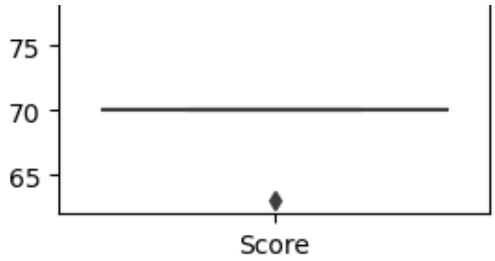
In [12]:

```
#importing library for plotting
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [13]:

```
plt.figure(figsize= [3,2])
sns.boxplot(score)
plt.show()
plt.figure(figsize= [3,2])
sns.boxplot(score1)
plt.show()
```





In [14]:

```
# now as we can clearly see with the plotting that even if in the score1 data
# the extreme values are widely spread from the mean
# we are still getting a near close mad score for both score and score1 dataframe
print(f'the mean absolute deviation for score dataframe is {round(score["Score"].mad())}'
)
print(f'the mean absolute deviation for score1 dataframe is {round(score1["Score"].mad())
}')
```

```
the mean absolute deviation for score dataframe is 3
the mean absolute deviation for score1 dataframe is 4
```

In [15]:

```
# for scenarios like this we use standard deviation
# standard deviation performs a square operation in all of the (data point - avg) or the
difference in
# data points , then performs average on all these squared differences
# Then perform square root operation in (average of squared differences)
# Now let's find all the differences of data points with average extracted above
# Then using a for loop let's take all these differences into a list
# we will take absolute values here as few data points may come out as negative
diff_lst1 = []
for i in score1.Score:
    diff_lst1.append((abs(score_avg-i))**2)
diff_lst1
```

Out[15]:

```
[100, 100, 9, 9, 9, 9]
```

In [16]:

```
# Now let's find the average of all these differences and perform square root on the same
# This would be our standard deviation as well
avg_mean_score1 = round(np.mean(diff_lst1))
print(f'the standard deviation is {int(np.sqrt(avg_mean_score1))}')
```

```
the standard deviation is 6
```

In [17]:

```
# Let's perform standard deviation with stdev function
import statistics
print(f'the standard deviation is {int(statistics.stdev(score1["Score"]))}')
```

```
the standard deviation is 6
```

In [19]:

```
# The difference in standard deviation and mean absolute deviation is clear now
print(f'the mean absolute deviation for score1 dataframe is {int(score1["Score"].mad())}'
)
print(f'the standard deviation for score1 dataframe is {int(statistics.stdev(score1["Scor
e"]))}')
```

```
the mean absolute deviation for score1 dataframe is 4
the standard deviation for score1 dataframe is 6
```

In [ ]: