Measures of Dispersion

Measures of dispersion are statistical values that help in understanding how much the data in a dataset varies or is spread out from its central tendency. Dispersion indicates how much the values in a dataset deviate from the mean or median. In other words, it provides a measure of the diversity or variability of the dataset. The most common measures of dispersion are the range, variance, and standard deviation. This article aims to provide a comprehensive overview of the components of measures of dispersion.

- **Range:** Range is the simplest measure of dispersion and it represents the difference between the highest and lowest value in a dataset. It provides a rough idea of how far the data spreads. To calculate the range, subtract the lowest value from the highest value. For instance, consider a dataset with values 5, 7, 10, 12, 15. The range of this dataset would be 15-5=10. However, the range does not account for the variability between the values in the dataset.

Python Code for Range:

Range

`Range = Largest data value - Smallest data value`

- We used inbuilt functions like `max` to find maximum value and `min` to find minimum value

In [8]:
```
# Sample Data
arr = [1, 2, 3, 4, 5]
arr2= [1,2,3,4,5,6,7,8,9,10]
#Finding Max
Maximum = max(arr)
Max     = max(arr2)
# Finding Min
Minimum = min(arr)
Min     = min(arr2)

# Difference Of Max and Min
Range = Maximum-Minimum
Range2= Max - Min
print("Maximum = {}, Minimum = {} and Range = {}".format(Maximum, Minimum, Range))
print("Maximum = {}, Minimum = {} and Range = {}".format(Max, Min, Range2))
```

```
Maximum = 5, Minimum = 1 and Range = 4
Maximum = 10, Minimum = 1 and Range = 9
```

In [10]:
```
x = range(0,1001)
for i in x:
    maximum = max(x)
    minimum = min(x)
Range = maximum - minimum
print("Maximum = {}, Minimum = {} and Range = {}".format(maximum, minimum, Range))
```

```
Maximum = 1000, Minimum = 0 and Range = 1000
```

- **Variance:** The variance is the average of the squared deviation of each data point from the mean. It measures how far the data points are from the mean. Variance is calculated by squaring the difference between each data point and the mean, adding all of these squared differences, and then dividing the sum by the total number of data points in the dataset. The formula for variance is:

$$\text{Variance} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$$

where,

x is the data point

n is the total number of data points in the dataset

x_bar is the mean of the dataset.

The variance provides a more accurate measure of the variability in the dataset than the range as it considers all the values in the dataset. However, the variance is influenced by outliers or extreme values, making it sensitive to extreme values.

Python Code for Variance:

Variance

$$\sigma^2 = \frac{\sum(x-\mu)^2}{N}$$

- N = number of terms
- μ = mean

In [12]:
```
# sample data
arr = [1, 2, 3, 4, 5]
arr2= [1,2,3,4,5,6,7,8,9,10]
# variance
print("Var = ", (statistics.variance(arr)))
print("Var = ", (statistics.variance(arr2)))
```

```
Var =  2.5
Var =  9.166666666666666
```

- **Standard Deviation:** The standard deviation is the square root of the variance. It is a widely used measure of dispersion as it is easy to interpret and has desirable mathematical properties. It indicates how much the data points deviate from the mean in terms of standard deviations. A small standard deviation indicates that the data points are tightly clustered around the mean, while a large standard deviation indicates that the data points are spread out from the mean.The formula for standard deviation is:

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

where,

x is the data point

n is the total number of data points in the dataset

x_bar is the mean of the dataset.

The standard deviation is preferred over the variance as it provides a measure of dispersion that is in the same unit as the data. It is also less sensitive to outliers than the variance.

Python code for standard deviation:

Standard Deviation

$$\sigma = \sqrt{\frac{\Sigma (x - u)^2}{N}}$$

- `N = number of terms`
- `μ = mean`

]:
```
# sample data
arr = [1, 2, 3, 4, 5]
arr2= [1,2,3,4,5,6,7,8,9,10]
# variance
print("Var = ", (statistics.stdev(arr)))
print("Var = ", (statistics.stdev(arr2)))
```

```
Var =  1.5811388300841898
Var =  3.0276503540974917
```

- **Coefficient of Variation:** The coefficient of variation is a measure of dispersion that expresses the standard deviation as a percentage of the mean. It is used to compare the variability of datasets with different means. The formula for coefficient of variation is:

$$\text{Coefficient of Variation} = \frac{\text{Standard Deviation}}{\bar{x}} \times 100$$

where,

x_bar is the mean of the dataset.

A higher coefficient of variation indicates that the data is more dispersed, while a lower coefficient of variation indicates that the data is less dispersed. The coefficient of variation is useful in comparing the variability of datasets that have different means, such as income levels in different countries.

- **Interquartile Range**

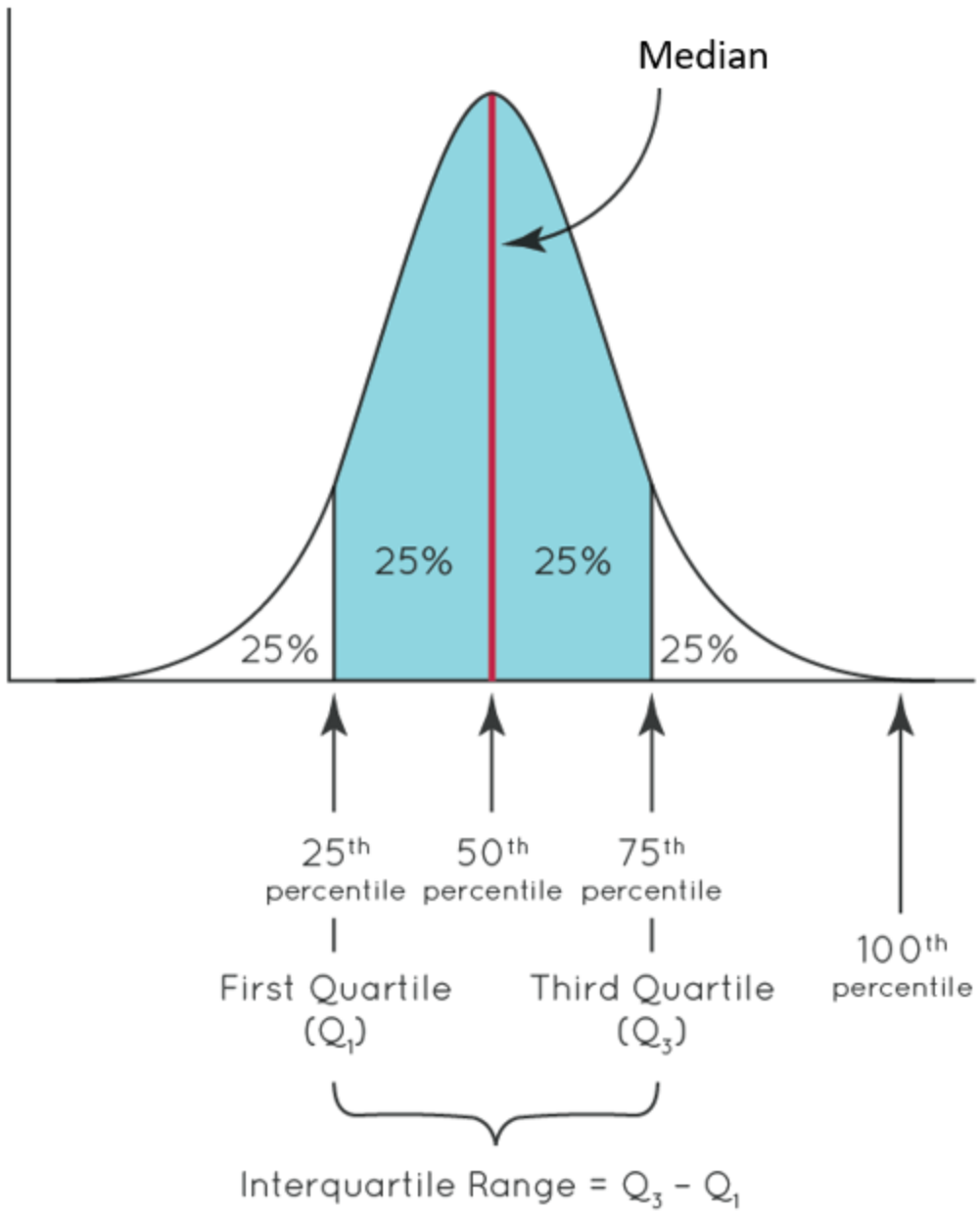
Quantile: A quantile determines how many values in a distribution are crossing a threshold, i.e., how many values are above and below a certain limit.

Quartiles (Quarter), Quintile (Fifth part) and percentiles (Hundredth) are some types of quantiles that we use.

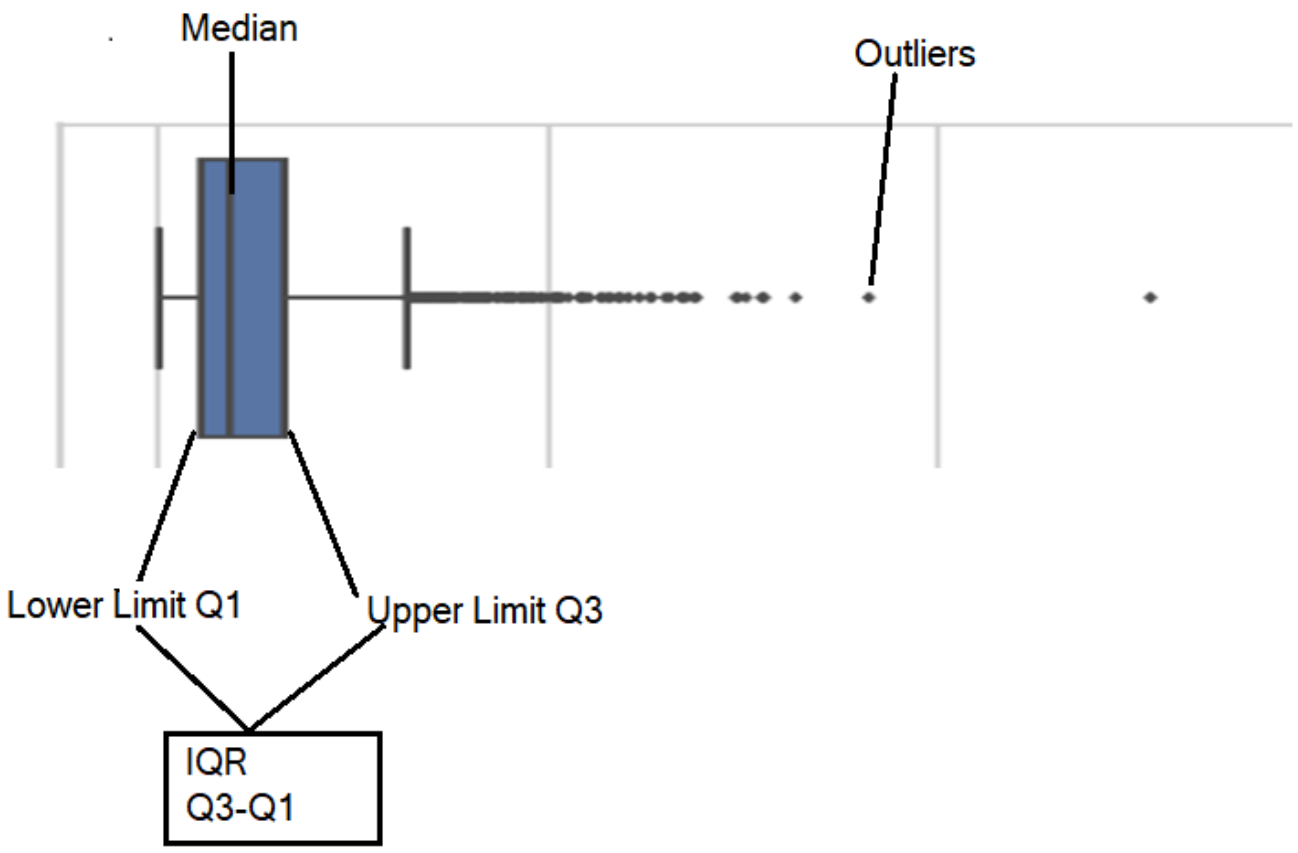
The interquartile range is a measure of dispersion that is based on the quartiles of the dataset. The quartiles divide the dataset into four equal parts, with each part representing 25% of the data. The interquartile range is the difference between the upper quartile (Q3) and the lower quartile (Q1). It represents the range of the middle 50% of the dataset, which is less affected by outliers than the range.

If we divide a distribution into four equal portions, we will speak of four quartiles. The first quartile includes all values that are smaller than a quarter of all values. In a graphical representation, it corresponds to 25% of the total area of a distribution. The two lower quartiles comprise 50% of all distribution values.

The Interquartile range is the distance between the 25^{th} and 75^{th} percentile, which is also the height of the box in a box plot.



Now coming to the real question on how we can remove the outliers' using stats and boxplot methods.



To remove outliers using Boxplot:

Q1 = quantile(0.25) / 25th quantile/percentile

Q3 = quantile(0.75)/ 75th quantile/percentile

IQR = Q3 – Q1

Now we have to find upper bound and lower bound of boxplot

Upper_Bound = Q3 + (1.5 * IQR)

Lower_Bound = Q1 – (1.5 * IQR)

To remove outlier data should be less than Upper_Bound and Greater than Lower_Bound

BOXPLOT METHOD

```
In [32]:
q1 = df['mileage'].quantile(0.25)
q1
```

```
Out[32]: 21466.5
```

```
In [33]:
q3 = df['mileage'].quantile(0.75)
q3
```

```
Out[33]: 63472.5
```

```
In [34]:
IQR = q3 - q1
IQR
```

```
Out[34]: 42006.0
```

```
In [35]:
u_bound = q3 + (1.5 * IQR)
l_bound = q1 - (1.5 * IQR)
```

```
In [36]:
print(q1,q3,IQR,u_bound,l_bound)
```

```
21466.5 63472.5 42006.0 126481.5 -41542.5
```

```
In [37]:
filtered_data = df[(df['mileage'] < u_bound) & (df['mileage'] > l_bound)]
```

```
In [38]:
filtered_data.shape
```

```
Out[38]: (2310, 10)
```

```
In [39]:
df.shape
```

```
Out[39]: (2499, 10)
```

`Information loss for Boxplot method`

```
In [40]:
2499-2310
```

```
Out[40]: 189
```

```
In [41]:
round((189/2499)*100,2)
```

```
Out[41]: 7.56
```

- Approximately 7.6% Info Loss

STATS METHOD

`Static`

- static threshold of 95 , 99 %ile (value beyond this)
- static threshold of 0.05 or 0.01

```
In [42]:
u_thresh = df['mileage'].quantile(0.99)
l_thresh = df['mileage'].quantile(0.01)
u_thresh = round(u_thresh,2)
l_thresh = round(l_thresh,2)
```

```
In [43]:
print(u_thresh,'and',l_thresh)
```

```
235380.64 and 665.6
```

```
In [44]:
filtered_data_2 = df[(df['mileage'] < u_thresh) & (df['mileage'] > l_thresh)]
```

```
In [45]:
filtered_data_2.shape
```

```
Out[45]: (2449, 10)
```

```
In [46]:
2499-2449
```

```
Out[46]: 50
```

```
In [47]:
round((50/2499)*100,2)
```

```
Out[47]: 2.0
```

`Approximately 2% loss in our data`

Mark as Read

Report An Issue

If you are facing any issue on this page. Please let us know.