1. Basic statistical functions for data Exploration.

November 15, 2022

[8]:
```
#importing libraries
import math
import numpy as np
import pandas as pd
```

1 Mean

Mean is only the average of all numbers in a particular numeric variable. when data contains outliers then finding mean and using it in any kind of manipulation is not suggested because a single outlier affects mean badly. so its solution is median.

[22]:
```
x = [8.0, 1, 2.5, 4, 28.0]
y = [5.0, 2 , 6.0, 7.2, 4.3]
```

[23]:
```
print("mean of x")
np.mean(x)
```

```
mean of x
```

[23]:
```
8.7
```

[24]:
```
print("mean of x")
np.mean(y)
```

```
mean of x
```

[24]:
```
4.9
```

[25]:
```
dataset=pd.read_csv('Iris.csv')
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             150 non-null    int64
 1   SepalLengthCm  150 non-null    float64
```

```
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

[26]:
```
np.mean(dataset['SepalLengthCm'])
```

[26]: 5.843333333333335

[27]:
```
np.mean(dataset['SepalWidthCm'])
```

[27]: 3.0540000000000007

2 Median

Median is a centre value after sorting all the numbers. if the total number is even then it is the average of centre 2 values. It does not depend on or affected outliers till half of the data does not become outliers.

[28]:
```
np.median(x)
```

[28]: 4.0

[29]:
```
np.median(y)
```

[29]: 5.0

[30]:
```
np.median(dataset['PetalLengthCm'])
```

[30]: 3.7586666666666693

[31]:
```
np.median(dataset['PetalWidthCm'])
```

[31]: 1.3

3 Mode

Mode represents the most frequent observation in a numeric variable. To find mode we do not have a function in Numpy, but we have a function in scipy.

[34]:
```
#find mode of petal length of each class
from scipy import stats
print(stats.mode(dataset['PetalLengthCm']))
print(stats.mode(dataset['PetalWidthCm']))
print(stats.mode(dataset['Species']))
```

```
ModeResult(mode=array([1.5]), count=array([14]))
ModeResult(mode=array([0.2]), count=array([28]))
ModeResult(mode=array(['Iris-setosa'], dtype=object), count=array([50]))
```

4 Variance

Variance measure how far is data point is from the mean, only the difference from MAD and variance is we take square here. The variance is computed by finding the difference between each data point and mean, squaring them, summing them up, and take the average of all those numbers. the numpy has a direct function to calculate variance.

[41]:
```
np.var(x)
```

[41]:
```
98.55999999999999
```

[42]:
```
np.var(y)
```

[42]:
```
3.056
```

[35]:
```
np.var(dataset['PetalLengthCm'])
```

[35]:
```
3.0924248888888854
```

[37]:
```
np.var(dataset['PetalWidthCm'])
```

[37]:
```
0.5785315555555559
```

[39]:
```
np.var(dataset['SepalLengthCm'])
```

[39]:
```
0.6811222222222222
```

[40]:
```
np.var(dataset['SepalWidthCm'])
```

[40]:
```
0.1867506666666667
```

5 Standard Deviation

Standard deviation is simply the square root of variance so we get again the value in the same measurement. And we can directly calculate the standard deviation using Numpy.

[43]:
```
np.std(x)
```

[43]:
```
9.927738916792684
```

[44]:
```
np.std(y)
```

[44]:
```
1.7481418706729726
```

[45]:
```
np.std(dataset['PetalLengthCm'])
```

[45]: 1.7585291834055201

[46]:
```
np.std(dataset['PetalWidthCm'])
```

[46]: 0.760612618588172