da-lab-1

March 18, 2023

#Central tendency

[ ]:
```
import pandas as pd
import numpy as np
```

[ ]:
```
ds=pd.read_excel('person.xlsx')
ds.head()
```

[ ]:

	City	Age	Gender	Marital Status	Income
0	New York	32	Male	Single	55000
1	Toronto	45	Female	Married	75000
2	Paris	28	Male	Single	45000
3	Berlin	40	Female	Married	120000
4	London	31	Male	Single	50000

1.Mean

[ ]:
```
mn= ds['Income'].mean()
print("Mean is ",mn)

# Gives the averge income of person which is basically the arthimetic means
```

```
Mean is  73200.0
```

2.Median

[ ]:
```
md= ds['Income'].median()
print("Median is ",md)

# Gives the mid value of Income of a person. Since there is one median value
```

```
Median is  75000.0
```

3.Mode

[ ]:
```
from statistics import mode
gd= mode(ds['Gender'])
print("Mode is ",gd)

# Gives the highest frequency of Gender of Person
```

```
Mode is  Male
```

1 Measures of variance

1.Range

[ ]:
```
mx=max(ds['Income'])
mn=min(ds['Income'])
rg=mx-mn
print('Range is',rg)

# Gives the difference between upper and lower limit of Income
```

```
Range is 140000
```

2.Variance

[ ]:
```
var=ds['Age'].var()
print("Variance is ",var)

# Since the value of variance is low therefore the distribution is not
# covers wide area
```

```
Variance is  68.60666666666667
```

3.Standard Deviation

[ ]:
```
sd=ds['Income'].std()
print("Standard Deviation is ",sd)

# Baically gives square root of variance and concises distribution of data
```

```
Standard Deviation is  31087.24282831571
```

4.Quartile

[ ]:
```
q1 = np.percentile(ds['Income'], 25)
q2 = np.percentile(ds['Income'], 50)
q3 = np.percentile(ds['Income'], 75)

print("First Quartile: ", q1)
print("Second Quartile: ", q2)
print("Third Quartile: ", q3)

# First quartile gives the Income from lowest to highest (25%)
# Second quartile gives the median value of Income (50%)
# Third quartile gives the Income above the median (75%)
```

```
First Quartile:  50000.0
Second Quartile:  75000.0
```

```
Third Quartile:  90000.0
```

5.Coefficient of Variance

```
[ ]: cov = sd/mn
print("Coefficient of variance is: ", cov)

# Gives ratio of standard deviation and mean which basically
# shows how the data is relatively spread with respect to mean
```

```
Coefficient of variance is:  3.108724282831571
```

6.Skewness

```
[ ]: from scipy.stats import skew
sn = skew(ds['Income'])
print("Skewness is: ", sn)


# skew is positive hence the distribution is positive and
# has long right tail i.e., right directed
```

```
Skewness is:  0.3765461418499572
```

7.Kurtosis

```
[ ]: from scipy.stats import kurtosis
kt = kurtosis(ds['Income'])
print("Kurtosis is: ", kt)

# kurtosis is postive hence the distribution has high peak
```

```
Kurtosis is:  0.13747064669867948
```

OTHER STATISTICAL FUNCTIONS

1.Correlation

```
[ ]: corelate = ds.corr()
print("Correlation of data is: \n\n", corelate)

# The value of correlation is positive which shows that the
# unit age and income are related to each other and
# hence varying with respect to each other
```

```
Correlation of data is:

              Age    Income
Age      1.000000  0.373117
Income   0.373117  1.000000
```

2.Sum of square

[ ]:
```
sos = np.sum((ds['Income']-mn)**2)
print("Sum of square is: ", sos)

# There are two types of sum of squares i.e., total sum of squares and residual
# sum of squares. In the above, total sum of squares is given for the Income
# and this helps in finding the coefficient of determination, which measures the
# proportion of the total variation in the dependent variable that is explained
# by independent variable
```

```
Sum of square is:  123050000000
```

3.Z-score

[ ]:
```
zs = (ds['Income']-mn)/sd
print("Z- Score is:\n", zs)
# All are positive hence all deviations are above mean
```

```
Z- Score is:
 0     1.447539
1     2.090890
2     1.125864
3     3.538429
4     1.286701
5     0.965026
6     2.734241
7     2.573403
8     1.608377
9     0.804188
10    2.251728
11    1.769214
12    0.000000
13    1.930052
14    0.965026
15    2.412565
16    2.734241
17    1.447539
18    2.573403
19    4.503455
20    2.895078
21    1.286701
22    2.251728
23    2.090890
24    3.538429
Name: Income, dtype: float64
```

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```