Group A Assignment No:3

**Descriptive Statistics - Measures of Central Tendency and variability**

Perform the following operations on any open source dataset (e.g., data.csv)

1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable.

2. Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of ‘Iris-setosa’, ‘Iris-versicolor’ and ‘Iris-versicolor’ of iris.csv dataset.

Provide the codes with outputs and explain everything that you do in this step.

Let's start by loading the required libraries and the data.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

df = pd.read\_csv(r"D:\dsbldlab\demo1.csv")

print(df.shape)

print(df.info())

utput:

1 (29, 6)

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 29 entries, 0 to 28

Data columns (total 6 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 math score 29 non-null int64

1 reading score 29 non-null int64

2 writing score 29 non-null int64

3 placement score 29 non-null int64

4 placement offer count 29 non-null int64

5 club join year 29 non-null int64

dtypes: int64(6)

memory usage: 1.5 KB

None

**Measures of Central Tendency**

### Mean

Mean represents the arithmetic average of the data. The line of code below prints the mean of the numerical variables in the data

The command **df.mean(axis = 0)** will also give the same output.

1 df.mean()

Output:

1 math score 84.482759

reading score 65.896552

writing score 73.793103

placement score 82.724138

placement offer count 2.448276

club join year 2019.448276

dtype: float64

t is also possible to calculate the mean of a particular variable in a data, as shown below

print(df.loc[:,'math score'].mean())

Output:

84.48275862068965

### Median

In simple terms, median represents the 50th percentile, or the middle value of the data, that separates the distribution into two halves. The line of code below prints the median of the numerical variables in the data. The command **df.median(axis = 0)** will also give the same output.

1df.median()

python

Output:

1 math score 77.0

reading score 68.0

writing score 78.0

placement score 85.0

placement offer count 3.0

club join year 2019.0

dtype: float64

It is also possible to calculate the median of a particular variable in a data, as shown in the first two lines of code below. We can also calculate the median of the rows by specifying the **(axis = 1)** argument. The third line below calculates the median of the first five rows.

1#to calculate a median of a particular column

2print(df.loc[:,'Age'].median())

3print(df.loc[:,'Income'].median())

4

5df.median(axis = 1)[0:5]

Output:

1 77.0

0 75.0

1 78.0

2 78.0

3 75.0

4 74.5

dtype: float64

## Measures of Dispersion

In the previous sections, we have discussed the various measures of central tendency. However, as we have seen in the data, the values of these measures differ for many variables. This is because of the extent to which a distribution is stretched or squeezed. In statistics, this is measured by dispersion which is also referred to as variability, scatter, or spread. The most popular measures of dispersion are standard deviation, variance, and the interquartile range.

### Standard Deviation

Standard deviation is a measure that is used to quantify the amount of variation of a set of data values from its mean. A low standard deviation for a variable indicates that the data points tend to be close to its mean, and vice versa. The line of code below prints the standard deviation of all the numerical variables in the data.

1df.std()

python

Output:

math score 25.973366

reading score 20.098587

writing score 17.341897

placement score 26.891178

placement offer count 0.631676

club join year 0.985111

dtype: float64

It is also possible to calculate the standard deviation of a particular variable, as shown in the first two lines of code below. The third line calculates the standard deviation for the first five rows.

1print(df.loc[:,'math score'].std())

2#calculate the standard deviation of the first five rows

3df.std(axis = 1)[0:5]

Output:

25.973366426915053

1 799.477496

2 803.328430

3 801.493107

4 799.602401

dtype: float64

### Variance

Variance is another measure of dispersion. It is the square of the standard deviation and the covariance of the random variable with itself. The line of code below prints the variance of all the numerical variables in the dataset. The interpretation of the variance is similar to that of the standard deviation.

1df.var()

python

Output:

math score 674.615764

reading score 403.953202

writing score 300.741379

placement score 723.135468

placement offer count 0.399015

club join year 0.970443

dtype: float64

### Skewness

Another useful statistic is skewness, which is the measure of the symmetry, or lack of it, for a real-valued random variable about its mean. The skewness value can be positive, negative, or undefined.

print(df.skew())

Output:

1 math score 2.879976

reading score -1.118335

writing score -2.551347

placement score 0.704166

placement offer count -0.705767

club join year 0.395053

dtype: float64

**Putting Everything Together**

 It is important to analyse these individually, however, because there are certain useful functions in python that can be called upon to find these values. One such important function is the **.describe()** function that prints the summary statistic of the numerical variables. The line of code below performs this operation on the data.

df.describe()

Write a Python program to view some basic statistical details like percentile, mean, std etc. of the species of ‘Iris-setosa’, ‘Iris-versicolor’ and ‘Iris-virginica’.

**Sample Solution:**

**Python Code:**

import pandas as pd

data = pd.read\_csv("iris.csv")

print('Iris-setosa')

setosa = data['Species'] == 'Iris-setosa'

print(data[setosa].describe())

print('\nIris-versicolor')

setosa = data['Species'] == 'Iris-versicolor'

print(data[setosa].describe())

print('\nIris-virginica')

setosa = data['Species'] == 'Iris-virginica'

print(data[setosa].describe())

Iris-setosa

Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm

count 50.00000 50.00000 50.000000 50.000000 50.00000

mean 25.50000 5.00600 3.418000 1.464000 0.24400

std 14.57738 0.35249 0.381024 0.173511 0.10721

min 1.00000 4.30000 2.300000 1.000000 0.10000

25% 13.25000 4.80000 3.125000 1.400000 0.20000

50% 25.50000 5.00000 3.400000 1.500000 0.20000

75% 37.75000 5.20000 3.675000 1.575000 0.30000

max 50.00000 5.80000 4.400000 1.900000 0.60000

Iris-versicolor

Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm

count 50.00000 50.000000 50.000000 50.000000 50.000000

mean 75.50000 5.936000 2.770000 4.260000 1.326000

std 14.57738 0.516171 0.313798 0.469911 0.197753

min 51.00000 4.900000 2.000000 3.000000 1.000000

25% 63.25000 5.600000 2.525000 4.000000 1.200000

50% 75.50000 5.900000 2.800000 4.350000 1.300000

75% 87.75000 6.300000 3.000000 4.600000 1.500000

max 100.00000 7.000000 3.400000 5.100000 1.800000

Iris-virginica

Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm

count 50.00000 50.00000 50.000000 50.000000 50.00000

mean 125.50000 6.58800 2.974000 5.552000 2.02600

std 14.57738 0.63588 0.322497 0.551895 0.27465

min 101.00000 4.90000 2.200000 4.500000 1.40000

25% 113.25000 6.22500 2.800000 5.100000 1.80000

50% 125.50000 6.50000 3.000000 5.550000 2.00000

75% 137.75000 6.90000 3.175000 5.875000 2.30000

max 150.00000 7.90000 3.800000 6.900000 2.50000