EXPERIMENT NO.1

AIM – Write a python program to open Comma Separated Value (CSV) and perform given statistical operations.

THEORY –

a) **CSV –**

A **CSV(Comma Separated Values)** file is a delimited text file that uses a comma to separate values. Each line of the file is a data record.

b) **Categorical variables –**

Categorical variables represent groupings of some kind. They are sometimes recorded as numbers, but the numbers represent categories rather than actual amounts of things.

There are three types of categorical variables: **binary**, **nominal**, and **ordinal** variables.

I) Binary variables – They represent data either YES or NO like Heads/tails in a coin flip.

II) Nominal variables – Data has group or order between them like Species names.

III) Ordinal variables – Data are ranked in specific order like Rating scale responses in a survey.

c) **Numerical variables –**

The data variable that takes on any value within a finite or infinite interval. They are also called **Continuous variables** because it exabits the features of Continuous Data.

There are two types of numerical variables, namely; interval and ratio variables.

I) Interval Variable - The Interval variable is an extension of the ordinal variable, with a standardized difference between variables in the interval scale.

II) Normal Distribution - A real-valued random variable is said to be normally distributed if its distribution is unknown.

OPERATIONS –

1. Identification of Categorical and Numerical Variables.

Code –

```
import pandas as pd

data = pd.read_csv('Data Set.csv')
var = data.dtypes
print(var)
```

Output –

```
"C:\Users\Sabyasachi Singh\AppData\Local\Pr
Id                int64
SepalLengthCm     float64
SepalWidthCm      float64
PetalLengthCm     float64
PetalWidthCm      float64
Species            object
dtype: object

Process finished with exit code 0
```

q

Species contains data as string so its **categorical** whereas rest of the columns are **Numerical Variables**.

3. Mean, Median, Mode, Variance, Standard Deviation, Quartile Range

Mean –

The mean is the average of a data set.

CODE –

```
import csv


def mean_cal():
    SepalLengthCm = 0
    SepalWidthCm = 0
    PetalLengthCm = 0
    PetalWidthCm = 0
    n = 0

    with open('Data Set.csv', 'r') as csv_file:
        csv_r = csv.reader(csv_file)
        (next(csv_r))
        for line in csv_r:
            SepalLengthCm = SepalLengthCm +
float(line[1])
            SepalWidthCm = SepalWidthCm + float(line[2])
            PetalLengthCm = PetalLengthCm +
float(line[3])
            PetalWidthCm = PetalWidthCm + float(line[4])
            n = n + 1

        mean = SepalLengthCm / n
        mean2 = SepalWidthCm / n
        mean3 = PetalLengthCm / n
        mean4 = PetalWidthCm / n
```

```
        print("Mean of SepalLengthCm :" +
str(round(mean, 4)))
        print("Mean of SepalWidthCm :" +
str(round(mean2, 4)))
        print("Mean of PetalLengthCm :" +
str(round(mean3, 4)))
        print("Mean of PetalWidthCm :" +
str(round(mean4, 4)))


mean_cal()
```

Output –

```
"C:\Users\Sabyasachi Singh\AppData\Local\Programs\P
Mean of SepalLengthCm :5.8433
Mean of SepalWidthCm :3.054
Mean of PetalLengthCm :3.7587
Mean of PetalWidthCm :1.1987

Process finished with exit code 0
```

Median –

The median is the middle of the set of numbers.

```
import csv


def Meadian_cal():
    n = 0
    l = []

    with open('Data Set.csv', 'r') as csv_file:
        csv_r = csv.reader(csv_file)
```

```
        (next(csv_r))
        for line in csv_r:
            a = line[1]
            l.append(a)
            n = n + 1

        median = int(n / 2)
        p = l[median]

        print('Median is :'+ str(median) + '  and the
value is :' + str(p))


Meadian_cal()
```

Output –

```
"C:\Users\Sabyasachi Singh\AppData\Local\Programs\Pyt
Median is :75  and the value is :6.6

Process finished with exit code 0
```

Mode –

The mode is the value that appears most often in a set of data values. If X is a discrete random variable, the mode is the value x at which the probability mass function takes its maximum value.

```
import csv

l = []
l2 = []
l3 = []
l4 = []

with open('Data Set.csv', 'r') as csv_file:
    csv_r = csv.reader(csv_file)
    (next(csv_r))
    for line in csv_r:
        a = line[1]
        b = line[2]
        c = line[3]
        d = line[4]
        l2.append(b)
        l.append(a)
        l3.append(c)
        l4.append(d)


def most_frequent(List):
    return max(set(List), key=List.count)


print("SepalLengthCm :" + most_frequent(l))
print("SepalWidthCm :" + most_frequent(l2))
print("PetalLengthCm :" + most_frequent(l3))
print("PetalWidthCm :" + most_frequent(l4))
```

Output –

```
"C:\Users\Sabyasachi Singh\AppData\Local\Progr
SepalLengthCm :5.0
SepalWidthCm :3.0
PetalLengthCm :1.5
PetalWidthCm :0.2

Process finished with exit code 0
```

Variance –

Variance measures how far each number in the set is from the mean and thus from every other number in the set.

```
import csv


def Variance_cal():
    SepalLengthCm = 0
    SepalWidthCm = 0
    PetalLengthCm = 0
    PetalWidthCm = 0
    n = 0
    l = []
    z = []
    z2 = []
    z3 = []
    z4 = []
    l2 = []
    l3 = []
    l4 = []

    with open('Data Set.csv', 'r') as csv_file:
```

```
        csv_r = csv.reader(csv_file)
        (next(csv_r))
        for line in csv_r:
            SepalLengthCm = SepalLengthCm +
float(line[1])
            SepalWidthCm = SepalWidthCm + float(line[2])
            PetalLengthCm = PetalLengthCm +
float(line[3])
            PetalWidthCm = PetalWidthCm + float(line[4])
            l.append(float(line[1]))
            l2.append(float(line[2]))
            l3.append(float(line[3]))
            l4.append(float(line[4]))
            n = n + 1
            mean = SepalLengthCm / n
            mean2 = SepalWidthCm / n
            mean3 = PetalLengthCm / n
            mean4 = PetalWidthCm / n

        for i in l:
            m = (i - mean) ** 2 / len(l)
            m2 = (i - mean2) ** 2 / len(l)
            m3 = (i - mean3) ** 2 / len(l)
            m4 = (i - mean4) ** 2 / len(l)

            z.append(m)
            z2.append(m2)
            z3.append(m3)
            z4.append(m4)

        print( "Variance of SepalLengthCm :" +
str(sum(z)))
        print( "Variance of SepalWidthCm :" +
str(sum(z2)))
        print("Variance of PetalLengthCm :" +
str(sum(z3)))
        print("Variance of PetalWidthCm :" +
str(sum(z4)))


Variance_cal()
```

Output –

```
"C:\Users\Sabyasachi Singh\AppData\Local\Programs\Pyt
Variance of SepalLengthCm :0.6811222222222217
Variance of SepalWidthCm :8.46150266666666
Variance of PetalLengthCm :5.02695733333332
Variance of PetalWidthCm :22.25405066666666
```

Standard Deviation-

The standard deviation is a measure of the amount of variation or dispersion of a set of values. A low standard deviation indicates that the values tend to be close to the mean of the set, while a high standard deviation indicates that the values are spread out over a wider range.

Code –

```
import csv


def Variance_cal():
    SepalLengthCm = 0
    SepalWidthCm = 0
    PetalLengthCm = 0
    PetalWidthCm = 0
    n = 0
    l = []
    z = []
    z2 = []
    z3 = []
    z4 = []
    l2 = []
    l3 = []
    l4 = []

    with open('Data Set.csv', 'r') as csv_file:
```

```
        csv_r = csv.reader(csv_file)
        (next(csv_r))
        for line in csv_r:
            SepalLengthCm = SepalLengthCm +
float(line[1])
            SepalWidthCm = SepalWidthCm + float(line[2])
            PetalLengthCm = PetalLengthCm +
float(line[3])
            PetalWidthCm = PetalWidthCm + float(line[4])
            l.append(float(line[1]))
            l2.append(float(line[2]))
            l3.append(float(line[3]))
            l4.append(float(line[4]))
            n = n + 1
            mean = SepalLengthCm / n
            mean2 = SepalWidthCm / n
            mean3 = PetalLengthCm / n
            mean4 = PetalWidthCm / n

        for i in l:
            m = ((i - mean) ** 2 / len(l))
            m2 = (i - mean2) ** 2 / len(l)
            m3 = (i - mean3) ** 2 / len(l)
            m4 = (i - mean4) ** 2 / len(l)

            z.append(m)
            z2.append(m2)
            z3.append(m3)
            z4.append(m4)

        print( "Standard Deviation of SepalLengthCm :" +
str(sum(z)**0.5))
        print( "Standard Deviation of SepalWidthCm :" +
str(sum(z2)**0.5))
        print("Standard Deviation of PetalLengthCm :" +
str(sum(z3)**0.5))
        print("Standard Deviation of PetalWidthCm :" +
str(sum(z4)**0.5))


Variance_cal()
```

Output –

```
"C:\Users\Sabyasachi Singh\AppData\Local\Programs\Python\Pytho
Standard Deviation of SepalLengthCm :0.8253012917851406
Standard Deviation of SepalWidthCm :2.9088662167013903
Standard Deviation of PetalLengthCm :2.2420877175822804
Standard Deviation of PetalWidthCm :4.717419916296053
```

Quartile Range –

The **interquartile range** is a measure of where the "middle fifty" is in a data set. Where a range is a measure of where the beginning and end are in a set, **an interquartile range is a measure of where the bulk of the values lie.**

Code –

```
import csv


def Quartile_cal():
    l = []

    l2 = []
    l3 = []
    l4 = []

    with open('Data Set.csv', 'r') as csv_file:
        csv_r = csv.reader(csv_file)
        (next(csv_r))
        for line in csv_r:
            l.append(float(line[1]))
            l2.append(float(line[2]))
            l3.append(float(line[3]))
            l4.append(float(line[4]))
```

```
        m = max(l) - min(l)
        m2 = max(l2) - min(l2)
        m3 = max(l3) - min(l3)
        m4 = max(l4) - min(l4)

        print("Quartile Range :" + str(m))
        print("Quartile Range :" + str(m2))
        print("Quartile Range :" + str(m3))
        print("Quartile Range :" + str(m4))


Quartile_cal()
```

Output –

```
"C:\Users\Sabyasachi Singh\AppData\Local\Programs
Quartile Range :3.6000000000000005
Quartile Range :2.4000000000000004
Quartile Range :5.9
Quartile Range :2.4

Process finished with exit code 0
```

4. Show categorical variables.

a. Show Binary data –

There is no Binary data in the given data set.

b. Nominal data –

Sepal length , Sepal Width , Petal length , Petal Width are Nominal data because they doesn't follow any order.

c. Show Ordinal data –

Species are Ordinal data because they follow a certain order.

CONCLUSION –

Therefore the given statistical operations were done successfully without using any pre defined libraries after applying the concepts of categorical and numerical variables.

Also during our experiment we found that contingency tables cannot be formed without having 2 categorical variables.