DATA ANALYTICS LAB

EXPERIMENT NO: 07

TITLE OF THE EXPERIMENT: Descriptive statistical analysis in Python

NAME: Maitreyi Erande

ROLL NO: SEETC317

DATE OF PERFORMANCE:

SIGNATURE OF COURSE FACULTY WITH DATE:

Pimpri Chinchwad Education Trust’s

Pimpri Chinchwad College of Engineering

Department of Electronics and Telecommunication Engineering

Experiment No: 7

Academic Year: 2020- 2021 **Year: SE (A, B, C)** **Semester: II**

Course: Data Analytics Lab **Course Code:204198**
Name: Maitreyi Erande **Roll No.:** SEETC317

TITLE: Descriptive statistical analysis in Python

AIM:

1. Perform different measures of central tendency on data set in python Pandas.
2. Perform different measures of central tendency and dispersion on Panda’s Series.

OBJECTIVES:

1. To understand the Descriptive statistical analysis with python Pandas

CO and PO MAPPED:

SOFTWARES:

1. Spyder (Python 3.7)

THEORY:

Statistics is a branch of mathematics which deals with the collection, analysis, interpretation and presentation of masses of numerical data. Statistics is a tool used to communicate our understanding of data. It helps us understand the world better, make assertions, and communicate our confidence in the statements we are making. Two main statistical methods are used in data analysis:

1. **Descriptive statistics:** This method is used to summarize data from a sample using measures such as the mean or standard deviation
2. **Inferential statistics:** With this method, you can conclude data that are subject to random variation (e.g., observational errors, sampling variation).

Descriptive statistics can be defined as the measures that summarize a given data, and these measures can be broken down further into the measures of central tendency and the measures of dispersion. Measures of central tendency include mean, median, and the mode, while the measures of dispersion include standard deviation and variance.

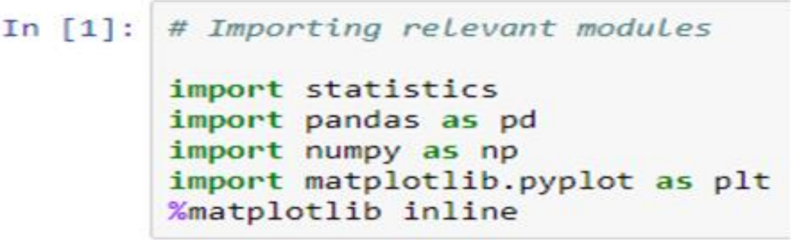
- **Measures of Central Tendency**

1. Mean
2. Median
3. Mode

- **Measures of Dispersion**

1. Variation
2. Standard Deviation

First, we need to import the Python statistics module.



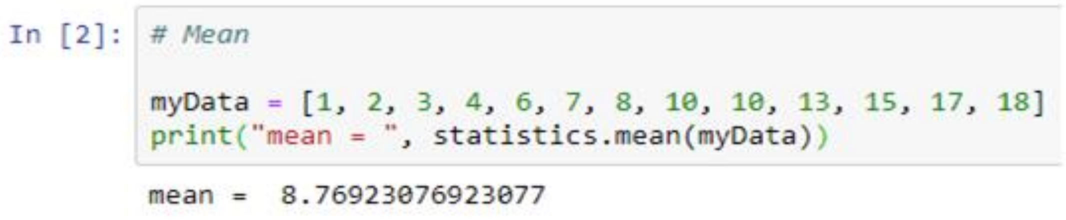
```
In [1]: # Importing relevant modules

        import statistics
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        %matplotlib inline
```

Measures of Central Tendency

Mean:

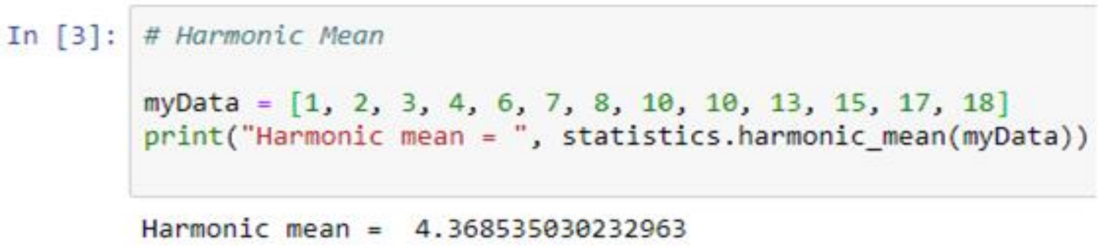
The arithmetic mean is the sum of data divided by the number of data-points. It is a measure of the central location of data in a set of values that vary in range. In Python, we usually do this by dividing the sum of given numbers with the count of the number present. Python mean function can be used to calculate the mean/average of the given list of numbers. It returns the mean of the data set passed as parameters.

- mean( ): Arithmetic mean ("average") of data.



```
In [2]: # Mean

        myData = [1, 2, 3, 4, 6, 7, 8, 10, 10, 13, 15, 17, 18]
        print("mean = ", statistics.mean(myData))

        mean =  8.76923076923077
```

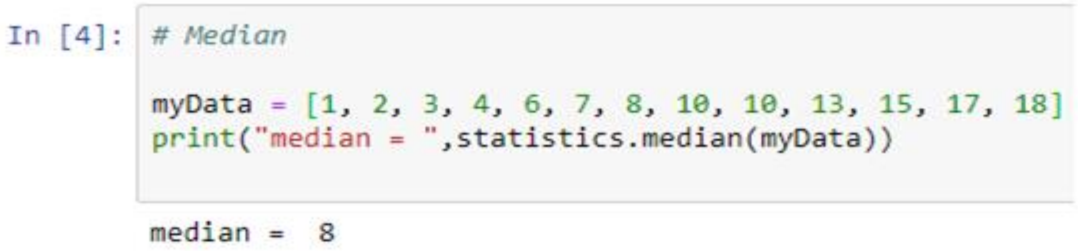
- harmonic_mean( ): It is the reciprocal of the arithmetic mean of the reciprocals of the data (say for three numbers a, b and c, 1/mean = 3/(1/a + 1/b + 1/c)).
-



```
In [3]: # Harmonic Mean

        myData = [1, 2, 3, 4, 6, 7, 8, 10, 10, 13, 15, 17, 18]
        print("Harmonic mean = ", statistics.harmonic_mean(myData))

        Harmonic mean =  4.368535030232963
```

Median:

median( ): Median or middle value of data is calculated as the mean of middle two. When the number of data points is odd, the middle data point is returned. The median is a robust measure of a central location and is less affected by the presence of outliers in your data compared to the mean.

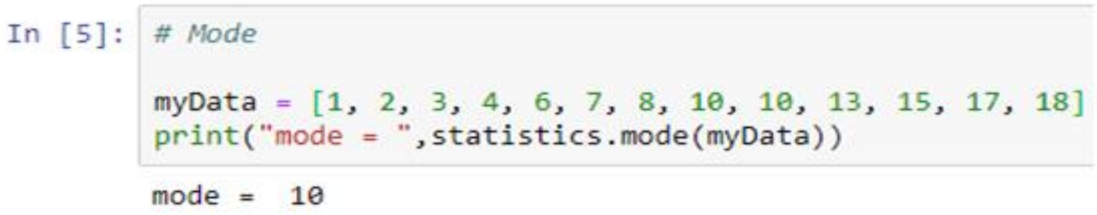


```
In [4]: # Median

        myData = [1, 2, 3, 4, 6, 7, 8, 10, 10, 13, 15, 17, 18]
        print("median = ",statistics.median(myData))

        median =  8
```

- median_low( ): Low median of data is calculated when the number of data points is odd. Here the middle value is usually returned. When it is even, the smaller of the two middle values is returned.
- median_high( ): High median of data is calculated when the number of data points is odd. Here, the middle value is usually returned. When it is even, the larger of the two middle values is returned.

Mode:

- mode( ): Mode (most common value) of discrete data. The mode (when it exists) is the most typical value and is a robust measure of central location.



```
In [5]: # Mode

        myData = [1, 2, 3, 4, 6, 7, 8, 10, 10, 13, 15, 17, 18]
        print("mode = ",statistics.mode(myData))

        mode =  10
```

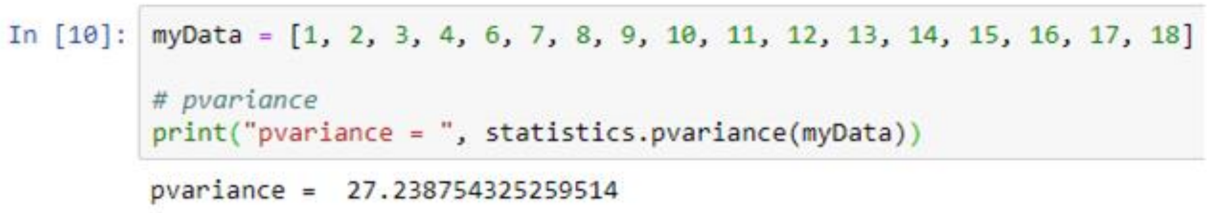
Measures of Dispersion

Measures of dispersion are statistics that describe how data varies, usually relative to the typical value. While measures of centre give us an idea of the typical value, measures of spread give us a sense of how much the data tends to diverge from the typical value.

These following functions (from the statistics module in python) calculate a measure of how much the population or sample tends to deviate from the typical or average values.

Population Variance:

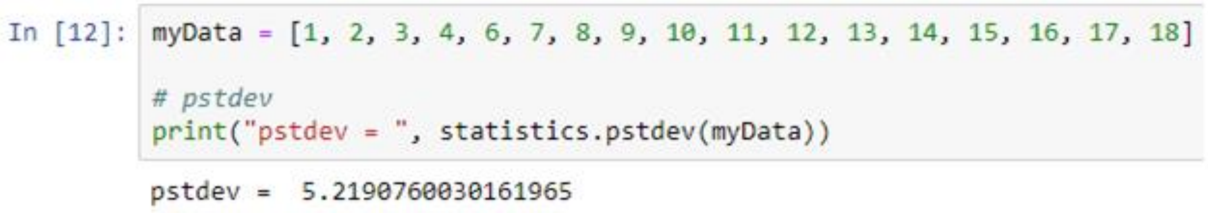
- pvariance( ): Returns the population variance of data. Use this function to calculate the variance from the entire population. To estimate the variance from a sample, the variance ( ) function is usually a better choice. When called with the entire population, this gives the population variance σ^2. When called on a sample instead, this is the biased sample variance s^2, also known as variance with N degrees of freedom.



```
In [10]: myData = [1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]

         # pvariance
         print("pvariance = ", statistics.pvariance(myData))

         pvariance =  27.238754325259514
```

Population Standard Deviation:

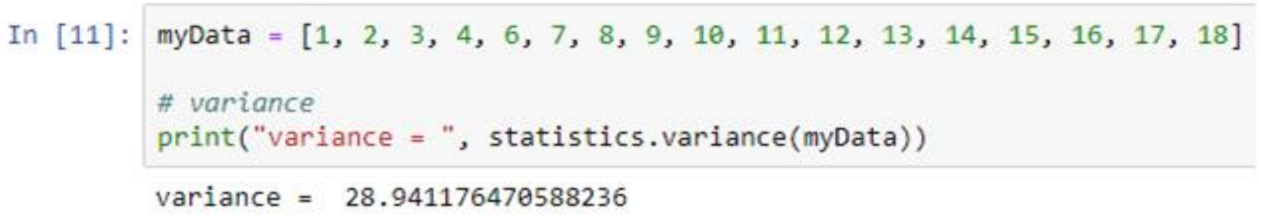
- pstdev( ): Return the population standard deviation (the square root of the population variance)



```
In [12]: myData = [1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]

         # pstdev
         print("pstdev = ", statistics.pstdev(myData))

         pstdev =  5.2190760030161965
```

Sample Variance:

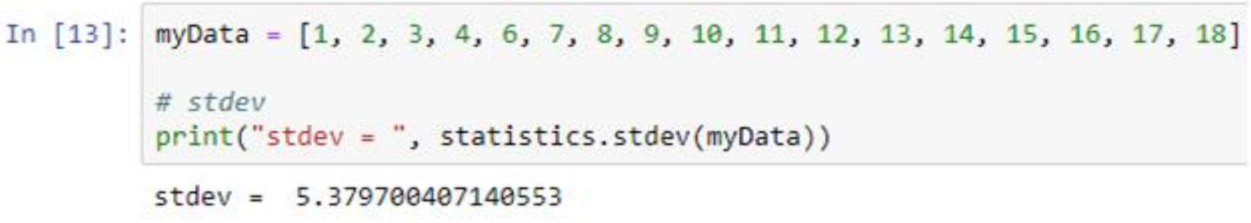
- variance ( ): Returns the sample variance of data, an iterable of at least two real-valued numbers. Variance, or second moment about the mean, is a measure of the variability (spread or dispersion) of data. A large variance indicates that the data is spread out; a small variance indicates it is clustered closely around the mean. If the optional second argument is given to the function, it should be the mean of data. This is the sample variance s^2 with Bessel's correction, also known as variance with N-1 degrees of freedom.



```
In [11]: myData = [1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]

         # variance
         print("variance = ", statistics.variance(myData))

         variance =  28.941176470588236
```

Sample Standard Deviation:

- stdev( ): Returns the sample standard deviation (the square root of the sample variance)



```
In [13]: myData = [1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]

         # stdev
         print("stdev = ", statistics.stdev(myData))

         stdev =  5.379700407140553
```

PROBLEM STATEMENT:

1. Write a program to create series in python pandas and perform following operation:

    a. Perform all Measures of Central Tendency (Mean, Mode, Median)

    b. Perform all Measures of Dispersion (Variance, Standard deviation)

2. Write program to import given data set(Toyota.csv) into python and perform following operation:

    a. Import csv file into pandas dataframe by removing index column and with replacing all junk values with NaN value

    b. Perform Mean, Mode and Median operation on few columns

SOLUTION:

1. **<u>Problem 1</u>**

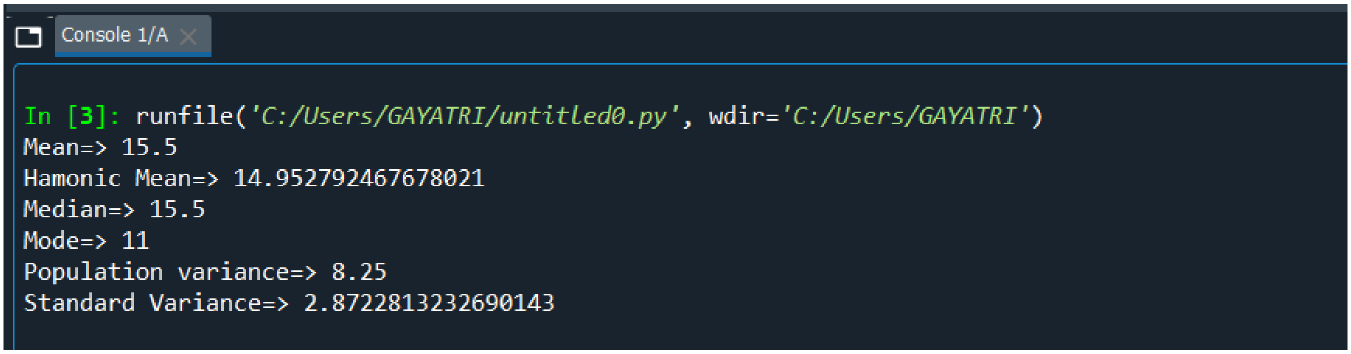
Input Code:

#Practical Assignment 7

#Problem Statement 1

```
import statistics

series=[11 ,12,13,14,15,16,17,18,19,20]

#Measures of Central Tendancy:

print("Mean=>",statistics.mean(series))

print("Hamonic Mean=>",statistics.harmonic_mean(series))

print("Median=>",statistics.median(series))

print("Mode=>",statistics.mode(series))

#Measures of Dispersion:

print("Population variance=>",statistics.pvariance(series))

print("Standard Variance=>",statistics.pstdev(series))
```

Output:



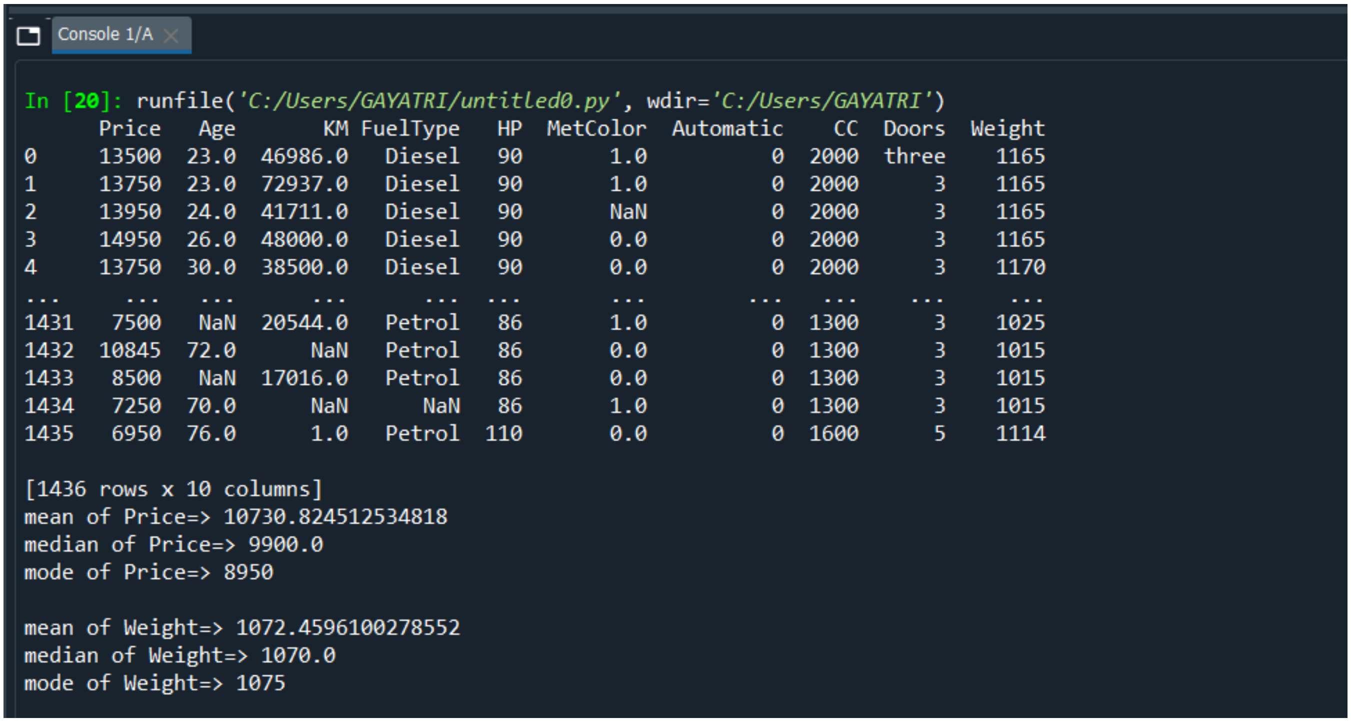
```
In [3]: runfile('C:/Users/GAYATRI/untitled0.py', wdir='C:/Users/GAYATRI')
Mean=> 15.5
Hamonic Mean=> 14.952792467678021
Median=> 15.5
Mode=> 11
Population variance=> 8.25
Standard Variance=> 2.8722813232690143
```

2. Problem 2

Input Code:

```
#Practical Assignment 7
#Problem Statement 2
import statistics
import os
import pandas as pd
os.chdir(r'C:/Users/GAYATRI/Downloads/dal/maitreyi')
dataset=pd.read_csv('Toyota.csv',index_col=0,na_values=["??"])
print(dataset)
print("mean of Price=>",statistics.mean(dataset['Price']))
print("median of Price=>",statistics.median(dataset['Price']))
print("mode of Price=>",statistics.mode(dataset['Price']),"\n")
print("mean of Weight=>",statistics.mean(dataset['Weight']))
print("median of Weight=>",statistics.median(dataset['Weight']))
print("mode of Weight=>",statistics.mode(dataset['Weight']))
```

Output:



```
In [20]: runfile('C:/Users/GAYATRI/untitled0.py', wdir='C:/Users/GAYATRI')
      Price   Age       KM FuelType   HP  MetColor  Automatic    CC  Doors  Weight
0     13500  23.0  46986.0   Diesel   90       1.0          0  2000  three    1165
1     13750  23.0  72937.0   Diesel   90       1.0          0  2000      3    1165
2     13950  24.0  41711.0   Diesel   90       NaN          0  2000      3    1165
3     14950  26.0  48000.0   Diesel   90       0.0          0  2000      3    1165
4     13750  30.0  38500.0   Diesel   90       0.0          0  2000      3    1170
...     ...   ...      ...      ...  ...       ...        ...   ...    ...     ...
1431   7500   NaN  20544.0   Petrol   86       1.0          0  1300      3    1025
1432  10845  72.0      NaN   Petrol   86       0.0          0  1300      3    1015
1433   8500   NaN  17016.0   Petrol   86       0.0          0  1300      3    1015
1434   7250  70.0      NaN      NaN   86       1.0          0  1300      3    1015
1435   6950  76.0      1.0   Petrol  110       0.0          0  1600      5    1114

[1436 rows x 10 columns]
mean of Price=> 10730.824512534818
median of Price=> 9900.0
mode of Price=> 8950

mean of Weight=> 1072.4596100278552
median of Weight=> 1070.0
mode of Weight=> 1075
```

CONCLUSION: From this practical experiment we understood the Descriptive statistical analysis with python Pandas and performed different measures of central tendency on data set in python Pandas and dispersion on Panda’s Series.

REFERENCE:

1. Wes McKinney and O’Reilly, “Python for Data Analysis”, 2nd Edition.
2. Jake Vander Plas and O’Reilly, “Python Data Science Handbook: Essential Tools for Working with Data”
3. https://swayam.gov.in/nd1_noc20_cs46/
4. https://m.dexlabanalytics.com/blog/python-statistics-fundamentals-how-to-describe-your-data-part-i
5. https://data-flair.training/blogs/python-descriptive-statistics/#:~:text=Descriptive%20Statistics%20in%20Python&text=Python%20Central%20tendency%20characterizes%20one,center%20and%20from%20each%20other.