

Dharmsinh Desai University, Nadiad
Department of Information Technology
DAIE, IT704
B.Tech. IT, Sem: VII

Submitted By

Roll No: IT076

Name: Dishant Modh

Experiment – 01 (Introduction to Python Programming)

Aim: Introduction to python programming for Data Mining

Tools/Apparatus: Visual Studio Code, Jupyter Notebook Extension

Procedure:

1. Python Programming Language Introduction:

1.1. Variables:

```
: ## variable
a,b = 1,2
name = "DAIE"
print(a+b)
print(name)

3
DAIE
```

1.2 Logical Operator

```
## Logical Operator
name = "DAIE"
age = 0
if name == "DAIE" or name == "DAIE2" and age >= 2 :
    print("Hello! Welcome.")
else :
    print("Good Bye!!")

Hello! Welcome.
```

1.4 List Data Type

BTech-IT, Sem-7, Term work, DAIE, IT704

1.3. Mathematical Operators:

```
## Mathematical Operator  
a, b = 10, 20  
print(b // a)  
print(a**5)
```

```
2  
100000
```

```
## List Data Type  
listitem1 = [10, 6.4, True, "DAIE", 6.4, 10]  
listitem1.append('DAIE LAB')  
print(listitem1)  
listitem2 = [60, 5.01, 76]  
listitem2.insert(1, 56)  
print(listitem2)  
listitem2.sort()  
print(listitem2)
```

```
[10, 6.4, True, 'DAIE', 6.4, 10, 'DAIE LAB']  
[60, 56, 5.01, 76]  
[5.01, 56, 60, 76]
```

1.5. Tuple Data Type:

```
## Truple Data Type
tuple1 = (9, 8, 7, 'DAIE', 'DAIE Lab', 6)
print(f'index: {tuple1.index("DAIE")}')
tuple2 = (70, 69, 60)
print(f'adding: {tuple1.__add__(tuple2)}')
print(f'length: {len(tuple1)}')
```

```
index: 3
adding: (9, 8, 7, 'DAIE', 'DAIE Lab', 6, 70, 69, 60)
length: 6
```

2 | Page

BTech-IT, Sem-7, Term work, DAIE, IT704

1.6 Dictionary Data Type

```
## Dictoionary Data Type
subject = ["abc", "def", "ghi"]
ratings = [7.7, 8.1, 6.8]

subject_choice_index = subject.index("def")
print(ratings[subject_choice_index])
```

```
8.1
```

1.7. Conditions:

```
## Conditions
hour = int(input())
print(f"Enter Hours: {hour}")

if hour < 8:
    print("It's morning")
elif hour < 18:
    print("It's the day")
else:
    print("It's the evening")
```

```
12
Enter Hours: 12
It's the day
```

1.8 For Loop

```
## For Loop
numbers = [70, 69, 60, 10, 10]

sum = 0

for val in numbers:
    sum = sum+val

print("The sum is", sum)

The sum is 219
```

3 | Page

1.10. Functions:

BTech-IT, Sem-7, Term work, DAIE, IT704

1.9 While Loop

```
## While Loop
n = int(input("Enter n: "))
sum = 0
i = 1

while i <= n:
    sum = sum + i
    i = i+1

print("The sum is", sum)

Enter n: 20
The sum is 210
```

```

## Function
def compute_hcf(x, y):

    if x > y:
        smaller = y
    else:
        smaller = x
    for i in range(1, smaller+1):
        if((x % i == 0) and (y % i == 0)):
            hcf = i
    return hcf

num1 = 54
num2 = 24

print("The H.C.F. is", compute_hcf(num1, num2))

```

The H.C.F. is 6

2. Handling multi-dimensional data and element-wise operators using

Numpy: 2.1. Import Numpy:

```

## Import Numpy
import numpy as np
print(f'Numpy version: {np.__version__}')

Numpy version: 1.19.5

```

2.2 Creating Data Vector

```
## Creating Data Vector and Accessing Data
import numpy as np

# Horizontal
list1 = [1, 2, 3]

# Vertical
list2 = [[10],
         [20],
         [30]]

vector1 = np.array(list1)
vector2 = np.array(list2)

print("Horizontal Vector")
print(vector1)

print("-----")

print("Vertical Vector")
print(vector2)
```

```
Horizontal Vector
[1 2 3]
-----
Vertical Vector
[[10]
 [20]
 [30]]
```

2.3. Element Wise Operators:

```

## Element Wise
import numpy as np

list1 = [9, 8, 7]
list2 = [4, 5, 6]

vector1 = np.array(list1)
print("First Vector      : " + str(vector1))
vector2 = np.array(list2)
print("Second Vector     : " + str(vector2))

addition = vector1 + vector2
print("Vector Addition   : " + str(addition))
subtraction = vector1 - vector2
print("Vector Subtraction : " + str(subtraction))
multiplication = vector1 * vector2
print("Vector Multiplication : " + str(multiplication))
division = vector1 / vector2
print("Vector Division     : " + str(division))

First Vector      : [9 8 7]
Second Vector     : [4 5 6]
Vector Addition   : [13 13 13]
Vector Subtraction : [5 3 1]
Vector Multiplication : [36 40 42]
Vector Division     : [2.25      1.6      1.16666667]

```

2.4 Matrix Operation

```

## Matrix Operator
import numpy as np

A = np.array([[1, 1], [2, 1], [3, -3]])
print(A.transpose())

[[ 1  2  3]
 [ 1  1 -3]]

```

2.5. Mathematical Operators:

```

## Mathematical Operator
a, b = 10, 20
print(b // a)
print(a**5)

2
100000

```

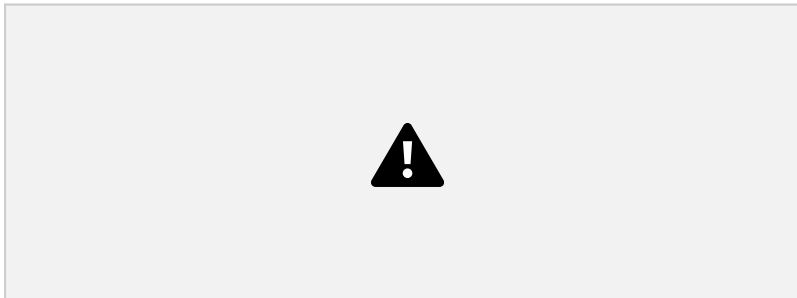
Experiment – 02 (Data pre-processing)

Aim: To perform data pre-processing using pandas package in python

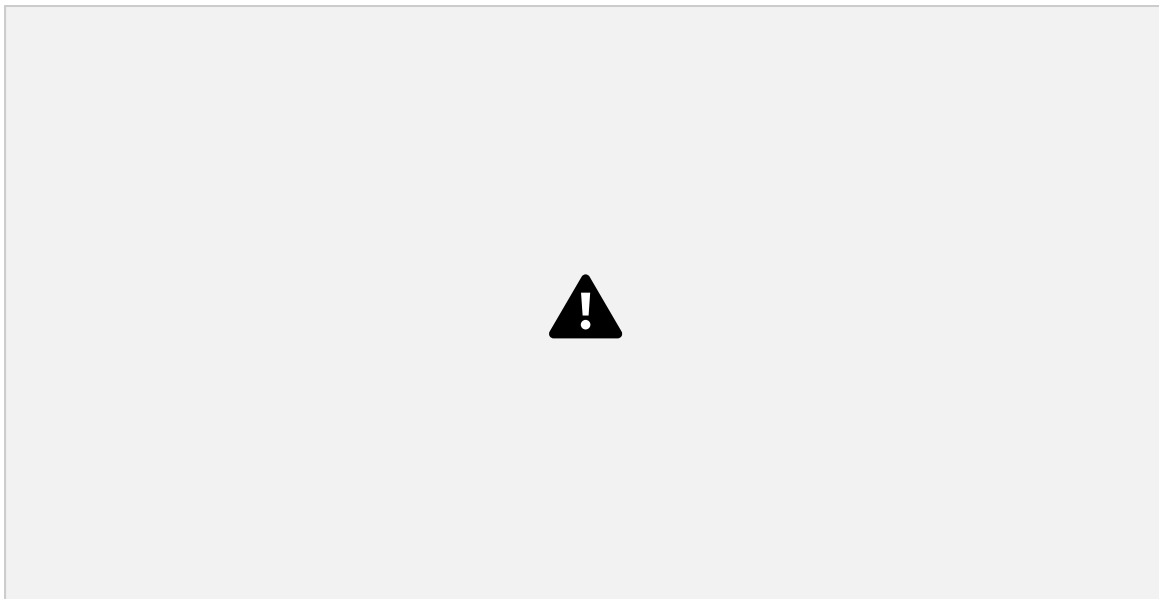
Tools/Apparatus: Visual Studio Code, Jupyter Notebook Extension

Procedure:

1 Import Pandas



2 Create Data Series



7 |

Page

BTech-IT, Sem-7, Term work, DAIE, IT704

3. Create Data Frame:



4. Reading Data from CSV file:



5. Reading inbuilt Data Sets:





6. Performing Mathematical Operation:



7. Handling Missing Values:





8. Data Normalization:



8.1 Min- Max Normalization



8.2. Z-score Normalization:



Experiment – 03 (Statistical Analysis and Visualization of Data)

Aim: To perform statistical analysis and visualization of data using pandas and matplotlib in python. (mean, mod, median, standard deviation, variance, correlation, bar chart, line chart, pie graph, histogram, box plot, scatter plot)

Tools/Apparatus: Visual Studio Code, Jupyter Notebook Extension

Procedure:

1.1 Import Numpy



1.2. Mean, Median, Mode, Standard Deviation and Variance of Data:



1.3. Correlation Coefficient and Percentile Data:



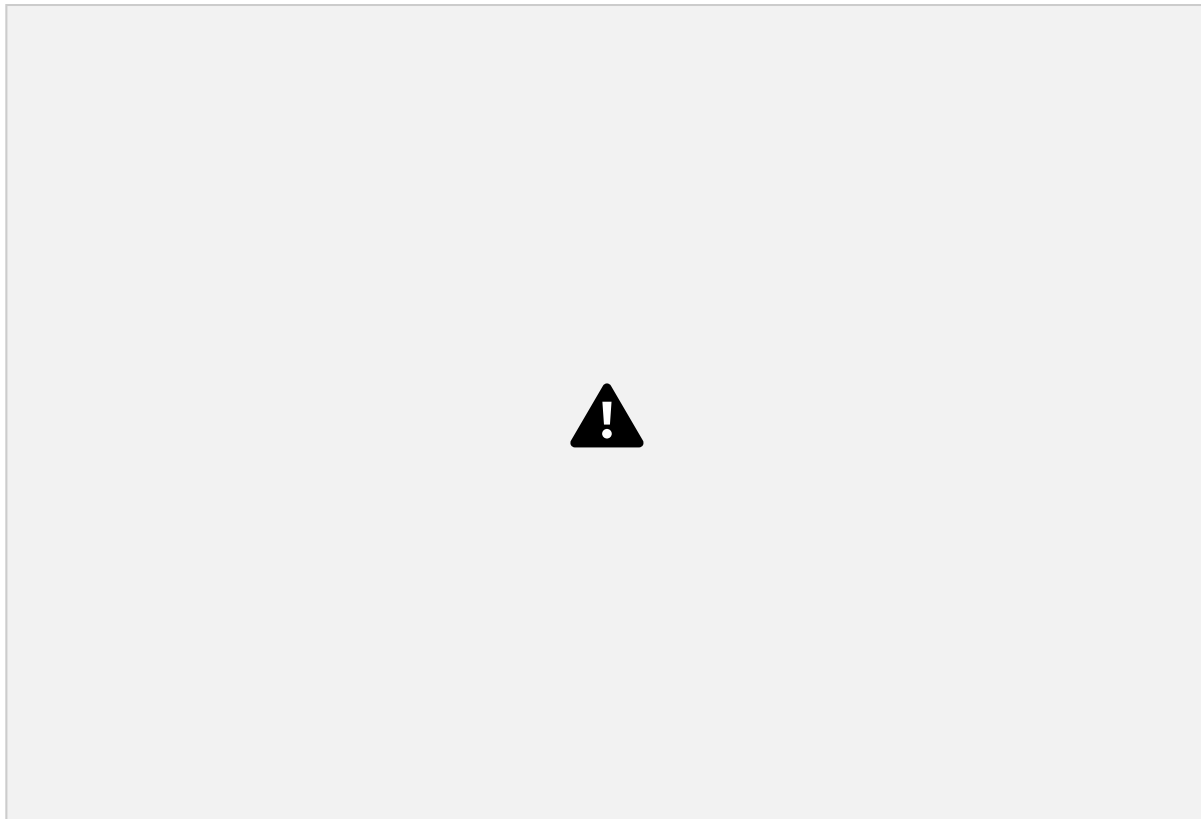
2.2 Line Chart

BTech-IT, Sem-7, Term work, DAIE, IT704

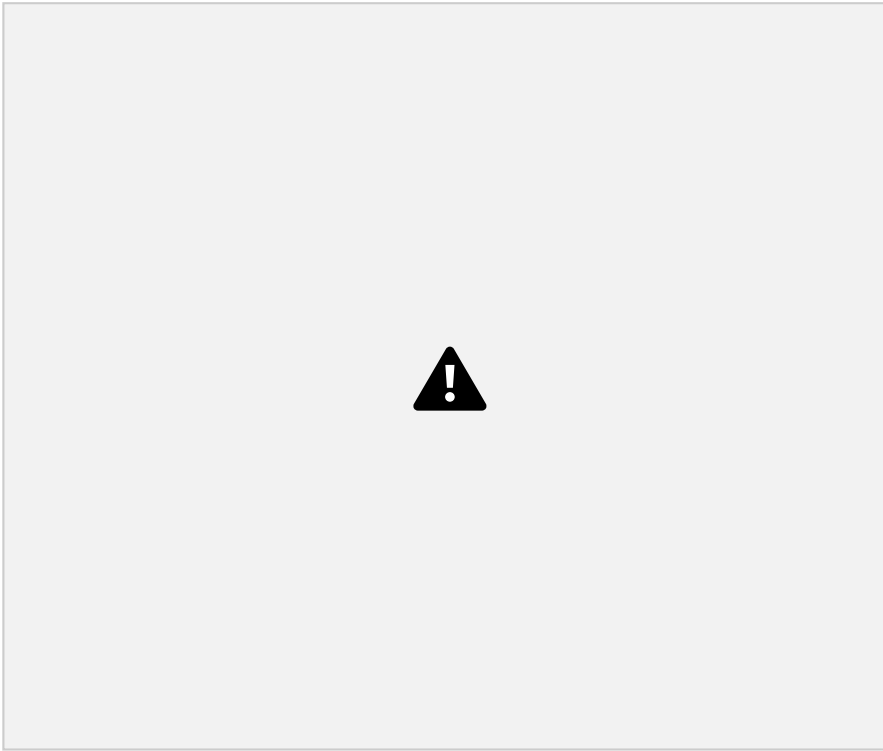
2.1. Import Matplotlib for PyPlot:



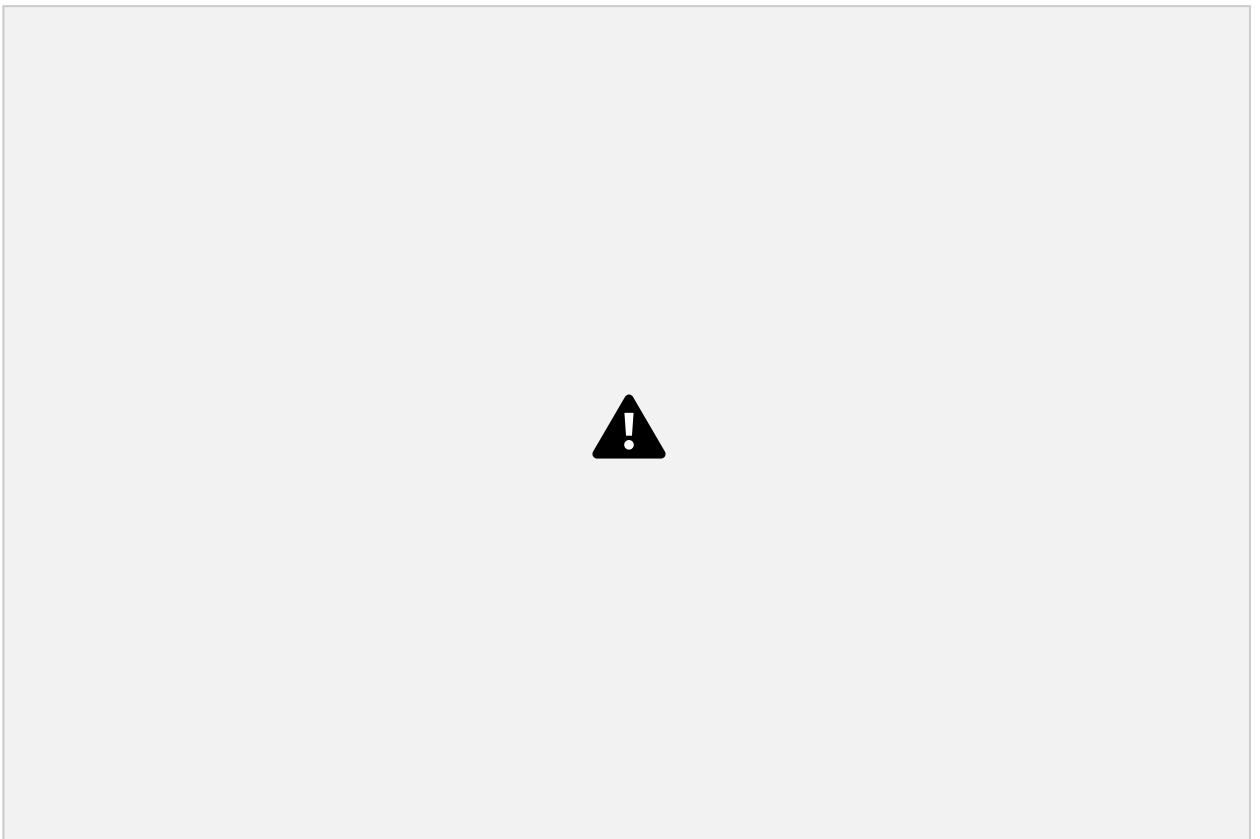
2.3. Scatter Plot:



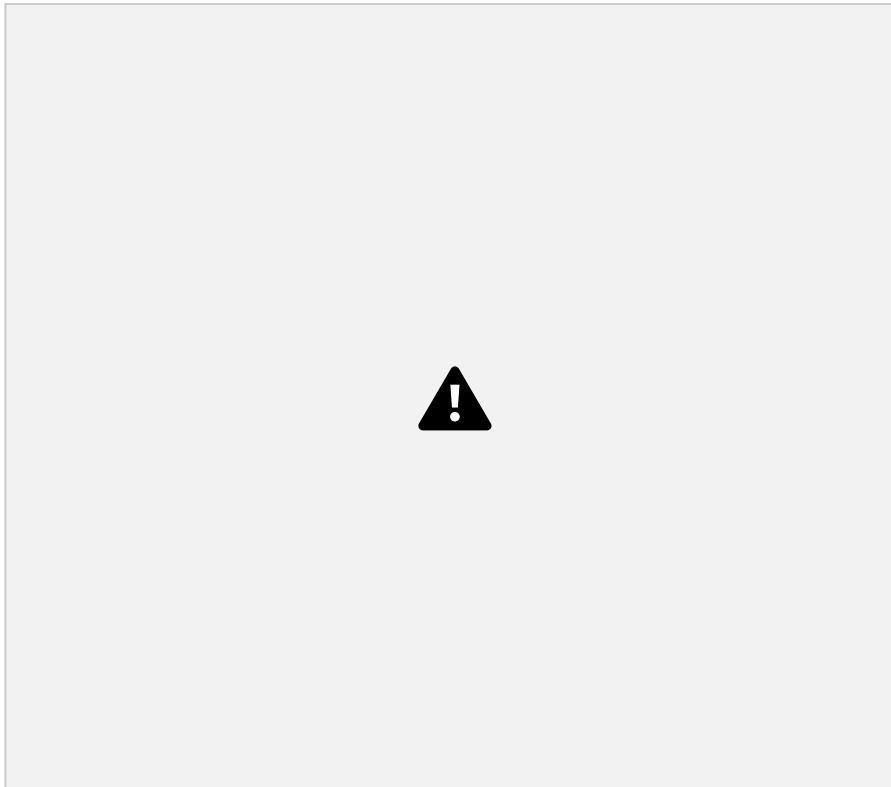
2.4 Bar Chart



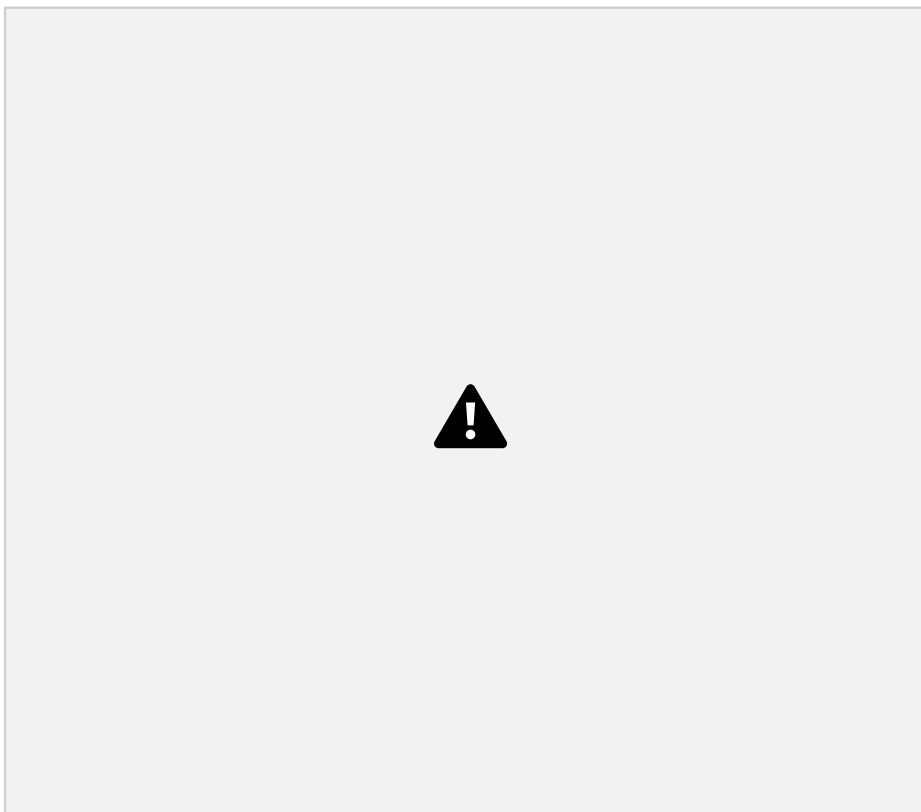
2.5. Pie Chart:



2.6 Box Plot



2.7. Sub Plot:



2.8 Plotting in Pandas



