

PART-B_Assignment No. 1

Aim: Perform the following operations using Python on the Facebook metrics data sets

- a) Create data subsets
- b) Merge Data
- c) Sort Data
- d) Transposing Data
- e) Shape and Reshape data

Objective: 1) To study data analysis operations using Python

.

Outcomes: 1) To apply the Analytical concept of Big data using Python.

Theory: Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library

Pseudo Code

:

1. Open jupyter notebook C:\Users\IT>jupyter notebook
 2. Upload CSV file
 3. Create new file
- ```
import numpy as np
import pandas as pd
a=pd.read_csv('1.csv')
```
4. Display output

### Creating a Dataframe

To create subsets of a dataframe, we need to create a dataframe

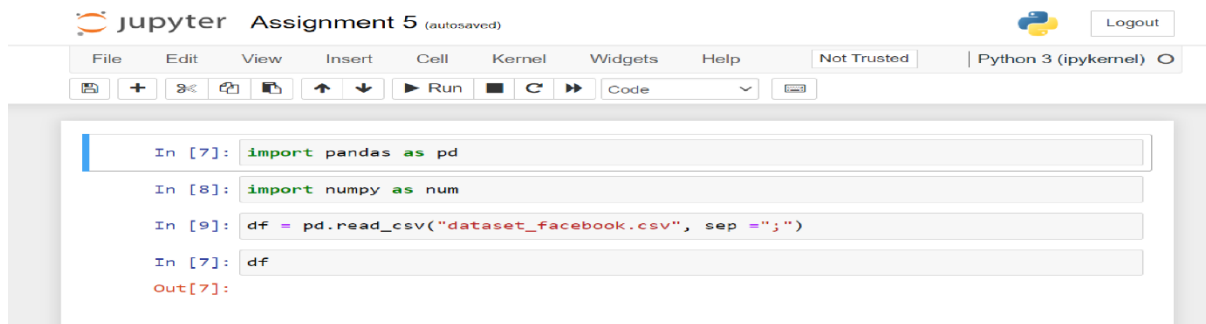
```
import pandas as pd
data = {"Roll-num": [10,20,30,40,50,60,70], "Age":[12,14,13,12,14,13,15],
"NAME":["John','Camili','Rheana','Joseph','Amanti','Alexa','Siri']}
block = pd.DataFrame(data)
print("Original Data frame:\n")
print(block)
```

Output:

Original Data frame:

Roll-num Age NAME

```
0 10 12 John
1 20 14 Camili
2 30 13 Rheana
3 40 12 Joseph
4 50 14 Amanti
5 60 13 Alexa
6 70 15 Siri
```



The image shows a Jupyter Notebook interface titled "Assignment 5 (autosaved)". The interface includes a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, Help. Below the menu bar is a toolbar with icons for file operations and execution. The main area contains a code cell with the following code:

```
In [7]: import pandas as pd
In [8]: import numpy as num
In [9]: df = pd.read_csv("dataset_facebook.csv", sep = ";")
In [7]: df
Out[7]:
```

Create a subset of a Python dataframe using the loc() function

Python loc() function enables us to form a subset of a data frame according to a specific row or column or a combination of both.

Syntax: pandas.dataframe.loc[]

Example 1:

Extract data of specific rows of a dataframe

block.loc[[0,1,3]]

Output:

Roll-num Age NAME

```
0 10 12 John
1 20 14 Camili
3 40 12 Joseph
```

Example 2: Create a subset of rows using slicing

block.loc[0:3]

Output:

Roll-num Age NAME

```
0 10 12 John
1 20 14 Camili
2 30 13 Rheana
3 40 12 Joseph
```

Example 3: Create a subset of particular columns using labels

block.loc[0:2,['Age','NAME']]

Output: Age NAME

```
0 12 John
1 14 Camili
2 13 Rheana
```

## **2. Using Python `iloc()` function to create a subset of a dataframe**

[Python `iloc\(\)` function](#) enables us to create subset choosing specific values from rows and columns based on indexes.

That is, unlike `loc()` function which works on labels, `iloc()` function works on index values. We can choose and create a subset of a Python dataframe from the data providing the index numbers of the rows and columns.

### **Syntax:**

```
pandas.dataframe.iloc[]
```

### **Example:**

```
block.iloc[[0,1,3,6],[0,2]]
```

Here, we have created a subset which includes the data of the rows 0,1,3 and 6 as well as column number 0 and 2 i.e. 'Roll-num' and 'NAME'.

### **Output:**

```
Roll-num NAME
0 10 John
1 20 Camili
2 40 Joseph
3 70 Siri
```

### 3. Indexing operator to create a subset of a dataframe

In a simple manner, we can make use of an indexing operator i.e. square brackets to create a subset of the data.

#### Syntax:

```
dataframe[['col1','col2','colN']]
```

#### Shape

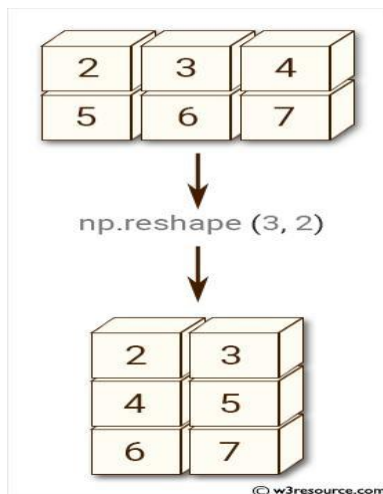
The *shape* attribute shows the number of items in each dimension. Checking a

DataFrame's [shape](#) returns a tuple with two integers. The first is the number of rows and the second is the number of columns.

```
df_hurricanes.shape(3, 2)
```

#### numpy.reshape

Gives a new shape to an array without changing its data.



1 ) import numpy as np

```
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
```

```
newarr = arr.reshape(4, 3)
```

```
print(newarr)
```

2) import numpy as np

```
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
```

```
newarr = arr.reshape(2, 3, 2)
```

```
print(newarr)
```

```
import numpy as np
```

```
x=np.arange(12)
```

```
y=np.reshape(x, (4,3))
```

```
x
```

```
y
```

```
#read the ".csv" file into dataframe "df"
```

```
df = pd.read_csv('50_Startups.csv')
```

```
#creating subsets and concat
```

```
df2 = df[:25]
```

```
df3 = df[25:]
```

```
#joining subsets horizontally
```

```
df6 = pd.concat([df2,df3],axis=0)
```

```
df4 = df.iloc[:,2:]
```

```
df5 = df.iloc[:,2:]
```

```
#joining subsets vertically
```

```
df6 = pd.concat([df4,df5],axis=1)
```

```
#merge
```

```
df2 = df.iloc[:,3:]
```

```
df3 = df.iloc[:,2:]
```

```
df7 = df2.merge(df3, on=['Marketing Spend'],how='inner')
```

```
#sort
```

```
x = df.sort_values(by = ['Marketing Spend'],ascending=True)
```

```
#transpose
```

```
x = df.T
```

```
#Casting data to wide format
```

```
df = pd.read_csv('abc.txt', sep=" ", names=['date', 'name', 'dollars'])
```

```
df2=df.pivot(index='date',columns='name',values='dollars').reset_index()
```

```
#Melting Data to long format
```

```
df3=pd.melt(df2,id_vars='date',value_vars=['George','Lisa','Michael'])
```

**Conclusion:** Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional.