Python for Data Science (Part – II) Financial Mathematics Department

Designed by Lecturer Engineer Syed Umaid Ahmed

1. For viewing all contents in a folder (Directory)

```
import os

path = “./Sales_Data”

os.listdir(path)
```

2. If the file name is not starting with dot (.) , than don’t take it

```
if not file startswith(‘.’)
```

3. **A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.**

```
df = pd.DataFrame()
```

4. If you want to read the CSV file in Python and concatenate (mix) multiple files in one

```
current_data = pd.read_csv(‘Filename.csv’)

all_months_data = pd.concat([all_months_data, current data ])
```

5. Make the CSV file of all data stored in variable ‘all_months_data’

```
all_months_data.to_csv(‘all_data.csv’)
```

6. **Check the complete code:**

```
for file in files:
    current_data = pd.read_csv(‘Filename.csv’)
    all_months_data = pd.concat([all_months_data, current data ])
all_months_data.to_csv(‘all_data.csv’)
```

7. How to read 10, 20, columns using head( ) the commands in Python:

```
all_data = pd.read_csv(‘all_data.csv’)
all_data.head()
```

8. Find rows in Python having NAN data (null data) and see them visually:

```
nan_df = all_data[all_data.isna().any(axis=1)]
display(nan_df.head())
```

9. Delete all rows in Python having NAN data (null data) and Check:

```
all_data = all_data.dropna(how=’all’)
all_data.head()
```

10. Extract the “Month Number” from the column ‘Order Date written in this format. On some places it is written ‘Or’ instead of the original month

04/09/19 08:46

```
all_data = all_data[all_data[‘Order Date’].str[0:2]!=’Or’]
```

11. For or calculation and manipulation we've to convert all the data extracted from column to numeric and "Integer" for calculation of Sales. The function of pandas is used.

```
pd.to_numeric( ) and .astype('int32')
```

12. For breaking any name (string) into the list we can use the function

```
A = "Osman"

A.split('m')
```

Output:

```
['Os', 'an']
```

13. For removing spaces (empty) we use strip(" "), It will remove empty space.

If something is in decimal digits before multiplying must convert them to float like this:

```
data.astype('int32') * data.astype('float')
```

14. For data analysis, "groupby( )" is very widely used function. It groups the data on given value.

The sum( ) function will provide the total addition

15. For converting the data into 'datetime' format to extract "HOUR & MINUTE" seperately, we can use.

```
all_data['Hour'] = pd.to_datatime(all_data['Order Data']).dt.hour

all_data['Minute'] = pd.to_datatime(all_data['Order Data']).dt.minute
```

16. For counting use the function count( )
17. For deleting duplicates use, .drop_duplicates( )
18. Explore other functions like lambda and apply( ) similarly.