**Pandas**

**Create Series**

import pandas as pd

series =pd.Series()

print (series)

arr1 =np.array ([10,12,14,16,18,20])

series =pd.Series(arr1)

print (series)

arr1 =np.array ([10,12,14,16,18,20])

series =pd.Series(arr1)

print (series)

print (series[2:5])

print (series[2:])

print (series[:5])

dict1 = {'Q1':1,'Q2':2,'Q3':3,'Q4':4}

series =pd.Series (dict1)

print (series)

dict1 = {'Q1':1,'Q2':2,'Q3':3,'Q4':4}

series =pd.Series (dict1)

print (series)

print (series ['Q1':'Q3'])

**Update Values in the Series**

dict1 = {'Q1':1,'Q2':2,'Q3':3,'Q4':4}

series =pd.Series (dict1)

print (series)

series[1] =100

print (series)

**Data Frames**

import numpy as np

import pandas as pd

list1 = [10,20,30,40,50,60]

S1= pd.Series (list1)

print (S1)

dataframe1 = pd.DataFrame(list1)

print (dataframe1)

**Dictionary List**

list2 = [{'Name': 'John','Age':25}, {'Name':'Sam','Age':20},

{'Name':'Eric','Age':30},{'Name':'Ken','Age':28}]

dataframe2=pd.DataFrame(list2)

print (dataframe2)

**List with one dictionary**

list2 = [{'Name': ['John','Sam','Eric','Ken'],'Age':[25,20,30,28] }]

dataframe2=pd.DataFrame(list2)

print (dataframe2)

**Data Frame from Dictionary**

dict = {'Name': ['John','Sam','Eric','Ken'],'Age':[25,20,30,28] }

dataframe2=pd.DataFrame(dict)

print (dataframe2)

**Data Frame from List with NaN**

list2= [['Sam', 28,5362],['Ken','Tom',32,7500], ['Eric',45,8512]]

df2 = pd.DataFrame(list2)

print (df2)

**Data Frame NaN**

list2 = [{'Name': 'John','Age':25}, {'Name':'Sam','Age':20},

{'Name':'Eric','Age':30},{'Name':'Ken','Age':28,'Last\_Name':'Soto'}]

dataframe2=pd.DataFrame(list2)

print (dataframe2)

**Row Index**

list2 = [{'Name': 'John','Age':25}, {'Name':'Sam','Age':20},

{'Name':'Eric','Age':30},{'Name':'Ken','Age':28,'Last\_Name':'Soto'}]

dataframe2=pd.DataFrame(list2,index = ['Row 1','Row 2','Row 3','Row 4'])

print (dataframe2)

**Row Index**

data1 = {'a': pd.Series ([10,11,12],index = ['a','b','c']),'b': pd.Series ([10,11,12,13],index = ['a','b','c','d'])}

tab1 = pd.DataFrame(data1)

print (tab1)

**Addition of new column to Data Frame[Column]**

data = {'one':pd.Series ([1,2,3], index = ['a','b','c']),

'two':pd.Series ([10,15,20,30], index = ['a','b','c','d']) }

Ser= pd.DataFrame(data)

Ser['three'] = pd.Series ([10,20,50],index = ['a','b','c'])

print (Ser)

**If more column values it won’t be added**

data = {'one':pd.Series ([1,2,3], index = ['a','b','c']),

'two':pd.Series ([10,15,20,30], index = ['a','b','c','d']) }

Ser= pd.DataFrame(data)

Ser['three'] = pd.Series ([10,20,50,60,70],index = ['a','b','c','d','e'])

print (Ser)

**print one column**

list2 = [{'Name': 'John','Age':25}, {'Name':'Sam','Age':20},

{'Name':'Eric','Age':30},{'Name':'Ken','Age':28,'Last\_Name':'Soto'}]

dataframe2=pd.DataFrame(list2,index = ['Row1','Row2','Row3','Row4'])

print (dataframe2['Name'])

**print multiple columns**

list2 = [{'Name': 'John','Age':25}, {'Name':'Sam','Age':20},

{'Name':'Eric','Age':30},{'Name':'Ken','Age':28,'Last\_Name':'Soto'}]

dataframe2=pd.DataFrame(list2,index = ['Row1','Row2','Row3','Row4'])

print (dataframe2 [['Name','Age']])

print (dataframe2.columns [1])

print (dataframe2.columns [1:3])

print (dataframe2[dataframe2.columns [1]])

print (dataframe2[dataframe2.columns [0:2]])

print (dataframe2[dataframe2.columns [:2]])

print (dataframe2[dataframe2.columns [1:]])

**Print Rows**

print (dataframe2.iloc [1])

print (dataframe2.iloc [1:3])

print (dataframe2.iloc [:3])

print (dataframe2.iloc [1:])

print (dataframe2.index.values)

print (dataframe2.index.values[1])

print (dataframe2.index.values[1:3])

print (dataframe2.index.values[:3])

print (dataframe2.index)

**print with column name**

print (dataframe2.loc['Row1'])

print (dataframe2.loc['Row2':])

print (dataframe2.loc[:'Row3'])

**Print Rows and colums**

dataframe2.iloc[1:2,0:1]

**Addition of Rows**

row1=pd.DataFrame([['Bob',32],['Mike',28]],columns = ['Name', 'Age'])

dataframe2 = dataframe2.append (row1)

print (dataframe2)

**Drop Rows**

Dataframe2=dataframe2.drop ('Row1')

Dataframe2=dataframe2.drop (0)

**Del a column**

del(Ser['three'])

print (Ser)

**Pop is to assign value**

x= Ser.pop('one')

print (Ser)

print (x)

**Read a csv**

df1 = pd.read\_csv("C:\\Users\\Noble\\Desktop\\Car Sales Dataset.csv")

print (df1)

df1 = pd.read\_csv("C:/Users/Noble/Desktop/Car Sales Dataset.csv")

print (df1)

**read excel**

df2 = pd.read\_excel("C:\\Users\\Noble\\Desktop\\Boston Hosuing .xlsx")

print (df2)

**Write to csv**

df1.to\_csv("C:/Users/Noble/Desktop/Car Sales Dataset1.csv")

**Update Values in the data Frame**

list2 = [{'Name': 'John','Age':25}, {'Name':'Sam','Age':20},

{'Name':'Eric','Age':30},{'Name':'Ken','Age':28}]

df1=pd.DataFrame(list2)

print (df1)

df1.iloc[0:1,0:1] = 'Jacob'

print (df1)

**Update multiple rows and columns**

df1.iloc[0:3,0:2] = ['Jacob',40]

print (df1)

**Example 2 – Reading data from file**

df2 = pd.read\_excel("C:\\Users\\Noble\\Desktop\\Boston Hosuing .xlsx")

x = df2.iloc[0:10,0:5]

print (x)

x.iloc[1:2,2:3] =100

print (x)

**Joins**

**Create Table from dictionary**

Student\_Data = {'Student\_id': ['AAA-123-2211','BBB-542-8542','CCC-854-8541','AAA-856-7458',

'DDD-745-7841'], 'First\_Name': ['Sam','Jim','Rick','Ben','Kevin'],

'Last\_Name': ['Edison','Mon','Dany','Denil','Pocket']}

student= pd.DataFrame (Student\_Data, columns = ['Student\_id','First\_Name','Last\_Name'])

print (student)

**Single Bracket – Series , Double bracket - Data Frame**

print (student ['Fist\_Name'])

print (type(student ['Fist\_Name']))

print (student [['Fist\_Name']])

print (type(student [['Fist\_Name']]))

**to extract (1st row 2nd Column)**

student.iloc[0:2, 0:2]

**Extract with Loc**

student.loc[0:2, 'Fist\_Name' :'Last\_Name']

**second table from dictionary**

Student\_Course= {'stu\_id':['AAA-123-2211','BBB-542-8542','CCC-854-7458','AAA-856-7458',

'DDD-745-7841'],'course':['ML','DS','RPA','BI','DL'],

'duration':[3,4,2,5,2]}

course = pd.DataFrame(Student\_Course)

course

**concat both tables / Union**

print (pd.concat([student,course]))

**Axis = 0 – Union**

print (pd.concat([student,course],axis =0))

**Index Join**

print (pd.concat([student,course],axis =1))

**Merge to Join two tables**

pd.merge(student,course,left\_on= 'Student\_id',right\_on ='stu\_id', how = 'inner')

**Left Outer**

pd.merge(student,course,left\_on= 'Student\_id',right\_on ='stu\_id', how = 'left')

**Full Outer**

pd.merge(student,course,left\_on= 'Student\_id',right\_on ='stu\_id', how = 'outer')

**Example for Join**

left = pd.DataFrame({

'id':[1,2,3,4,5],

'Name': ['Alex', 'Amy', 'Allen', 'Alice', 'Ayoung'],

'subject\_id':['sub1','sub2','sub4','sub6','sub5']})

right = pd.DataFrame({

'id':[1,2,3,4,5],

'Name': ['Billy', 'Brian', 'Bran', 'Bryce', 'Betty'],

'subject\_id':['sub2','sub4','sub3','sub6','sub5']})

print (left)

print ('\n')

print (right)

**# Join with two columns**

left = pd.DataFrame({

'id':[1,2,3,4,5],

'Name': ['Alex', 'Amy', 'Allen', 'Alice', 'Ayoung'],

'subject\_id':['sub1','sub2','sub4','sub6','sub5']})

right = pd.DataFrame({

'id':[1,2,3,4,5],

'Name': ['Billy', 'Brian', 'Bran', 'Bryce', 'Betty'],

'subject\_id':['sub2','sub4','sub3','sub6','sub5']})

print (pd.merge(left,right,on=['id','subject\_id']))

**# Left Join**

left = pd.DataFrame({

'id':[1,2,3,4,5],

'Name': ['Alex', 'Amy', 'Allen', 'Alice', 'Ayoung'],

'subject\_id':['sub1','sub2','sub4','sub6','sub5']})

right = pd.DataFrame({

'id':[1,2,3,4,5],

'Name': ['Billy', 'Brian', 'Bran', 'Bryce', 'Betty'],

'subject\_id':['sub2','sub4','sub3','sub6','sub5']})

print (pd.merge(left, right, on='subject\_id', how='left'))

**Import Libraries**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import os

**Read Data**

stats = pd.read\_csv('C:\\Noble\\Training\\Top Mentor\\Training\\Data Set\\DemographicData.csv')

stats

**Get current working directory**

os.getcwd()

**Change directory**

os.chdir ('C:\\Noble\\Training\\Top Mentor\\Training\\Data Set\\')

stats = pd.read\_csv ('DemographicData.csv')

stats

**other commands**

print (len (stats))

print (stats.columns)

print (len(stats.columns))

**assign column name to List**

col\_names = list(stats.columns)

print (col\_names)

**Top 5 records**

print (stats.head())

print (stats .head (10))

**bottom records**

print (stats.tail())

print (stats.tail(10))

stats.info()

stats.describe()

**Rename Columns - All**

stats.columns = ['C\_Name','C\_Code','B\_Rate','I\_Users','Income\_Grp']

stats.head()

**Rename – Individual Columns**

stats.rename (columns = {'C\_Name':'Country Name','C\_Code':'Country Code'},inplace= True)

stats.head()

**addition of two columns column name no space**

stats ['Add'] = stats.BirthRate+stats.InternetUsers

**addition of two columns column name with space**

stats ['Add1'] = stats ['Birth rate']+stats['Internet users']

print (stats [['Birth rate','Internet users','Add1']])

**delete column**

del stats['Add1']

**Drop Column – 1 for column**

stats =stats.drop ('Add',1)

**Filter the data**

stats[stats.InternetUsers <20]

stats[(stats.InternetUsers <10) & (stats.BirthRate >40)]

stats[(stats.InternetUsers <10) | (stats.BirthRate >40)]

stats[(stats.IncomeGroup =='Upper middle income')]

stats[(stats ["Internet users"] <10) & (stats ["Birth rate"] >40)]

stats[stats['Country Name'].str.contains('Af', case=False, na=False)]

stats[stats['Country Name'].str.startswith('Af', na=False)]

**Distinct Values**

stats.IncomeGroup.unique()

**Create new Data Frame based on condition**

tips1 = tips[tips.tip<=5]

tips1

**SQL s**

**Select three columns**

tips[['total\_bill', 'smoker', 'time']].head (25)

stats[(stats ["Internet users"] <10) & (stats ["Birth rate"] >40)]

stats[['CountryName','CountryCode']].head(10).loc[1:2]

**Not Null**

tips[tips.time.notna()]

**NULL records**

tips[tips.time.isna()]

**Group by**

tips.groupby('time').size()

**Count ()**

tips.groupby('time').count()

**Aggregate**

tips.groupby ('day').agg({'tip':np.mean,'day':np.size})

**Group by and size same column**

tips.groupby (['day','smoker'] ).agg({'tip':[np.size,np.mean]})

tips.groupby(['day','sex']).agg({'tip':[np.size,np.sum,np.mean]})

tips.groupby (['day','smoker'] ).agg({'tip':[np.size,np.mean,np.sum, np.min, np.max]})

**Delete Records Based on the condition**

tips = tips.drop(tips[tips.day == 'Sun'].index)

tips

**Sort the Values in a Data Frame**

stats = pd.read\_csv ('C:\\Noble\\Training\\Top Mentor\\Training\\Data Set\\DemographicData.csv')

stats

**Sort based on Birth Rate Columns**

stats.sort\_values(by=['Birth rate'])

**Sort based on Birth Rate Descending Order**

stats.sort\_values(by=['Birth rate'], ascending=False)

**Sorting Data frames by multiple columns**

stats.sort\_values(by=['Income Group', 'Birth rate', 'Internet users'])

**Change Data Type of column Birth Rate from float 64 to float 32**

stats['Birth rate'] =stats['Birth rate'].astype('float32')

stats.info()