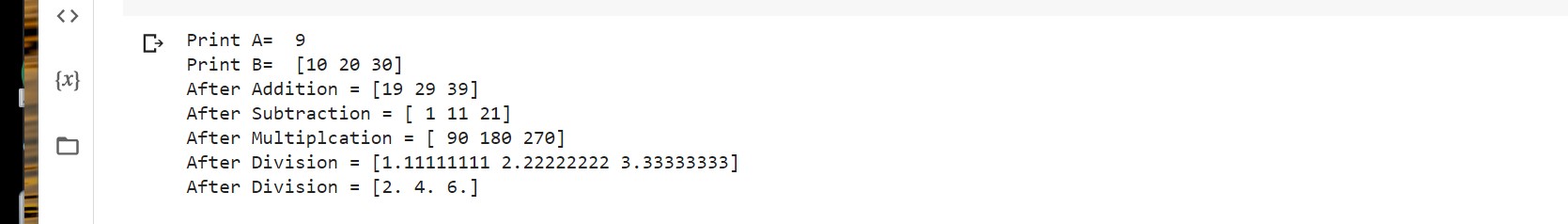
**NUMPY ARITHEMATIC OPERATIONS:**

# PROGRAM-1

import numpy as np a=np.array(9)

print("Print A= ",a) b=np.array([10,20,30]) print("Print B= ", b) addition= np.add(a,b) print("After Addition =", addition) sub=np.subtract(b,a) print("After Subtraction =",sub) mul=np.multiply (a,b) print("After Multiplcation =",mul) div=np.divide (b,a) print("After Division =",div) div1=np.divide (b,5) print("After Division =",div1)

# OUTPUT



# PROGRAM-2

a= np.array([0.25,1.33,1,111])

print(a) rec=np.reciprocal(a)

print(rec)

# OUTPUT



# PROGRAM-3

a=np.array([10,100,1000])

print(a) pow=np.power(a,2) print("after ^2 = ", pow) b=np.array([2,3,1]) pow1=np.power(a,b)

print("after b array elements as ^ = ", pow1)

# OUTPUT

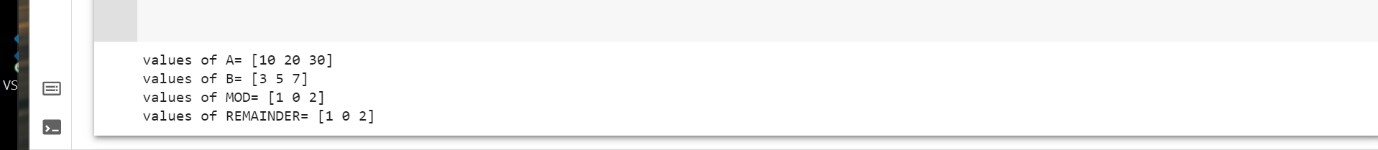


# PROGRAM-4

a= np.array ([10,20,30])

b= np.array ([3,5,7]) print("values of A=", a) print("values of B=", b) mm=np.mod(a,b) rm=np.remainder(a,b) print("values of MOD=", mm) print("values of REMAINDER=", rm)

# OUTPUT



# PROGRAM-5

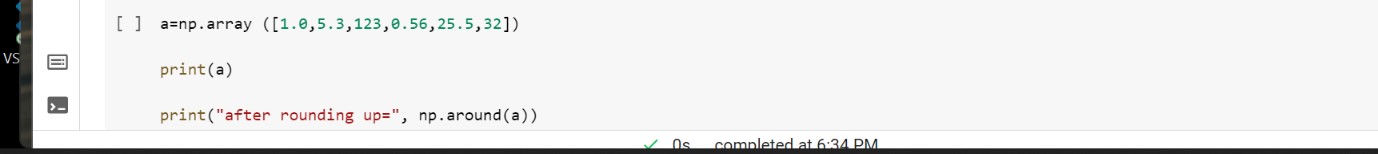
a=np.array([-5.6j, 0.2j,11, 1+1j])

print(a)

print("real=",np.real(a)) print("imaginary=",np.imag(a))

print("Conjugate=", np.conj(a))

# OUTPUT



# PROGRAM-6

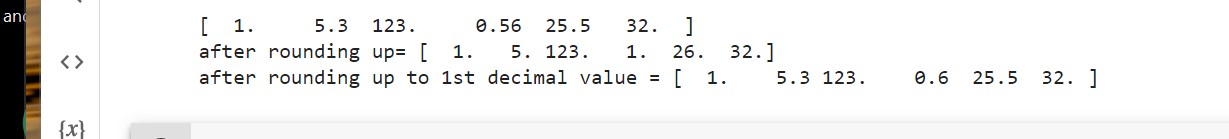
a=np.array ([1.0,5.3,123,0.56,25.5,32])

print(a)

print("after rounding up=", np.around(a))

print("after rounding up to 1st decimal value =", np.around(a,decimals=1))

# OUTPUT



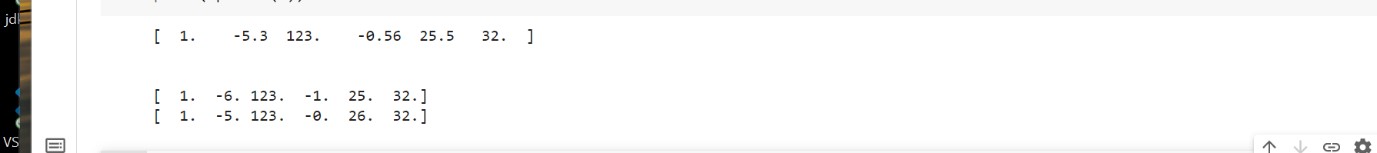
# PROGRAM-7

a=np.array ([1.0,-5.3,123,-0.56,25.5,32])

print(a) print('\n') print(np.floor(a))

print(np.ceil(a))

# OUTPUT



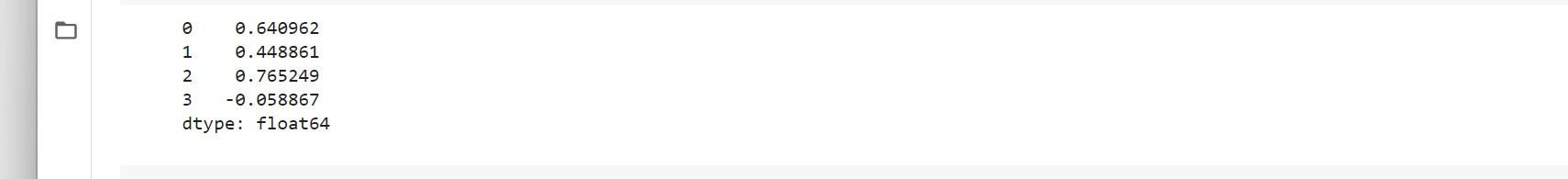
**PANDAS:**

# PROGRAM-1

import pandas as pd import numpy as np

s = pd.Series(np.random.randn(4)) print (s)

# OUTPUT

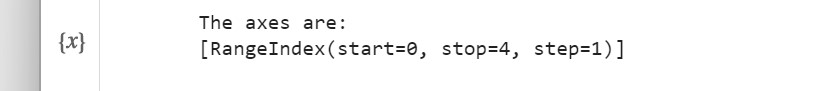


# PROGRAM-2

import pandas as pd import numpy as np s = pd.Series(np.random.randn(4))

print ("The axes are:") print (s.axes)

# OUTPUT



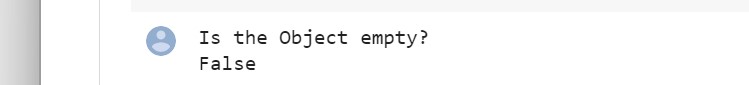
# PROGRAM-3

import pandas as pd import numpy as np

s = pd.Series(np.random.randn(4)) print ("Is the Object empty?")

print (s.empty)

# OUTPUT



# PROGRAM-4

import pandas as pd import numpy as np s = pd.Series(np.random.randn(4)) print (s)

print ("The dimensions of the object:") print (s.ndim)

# OUTPUT



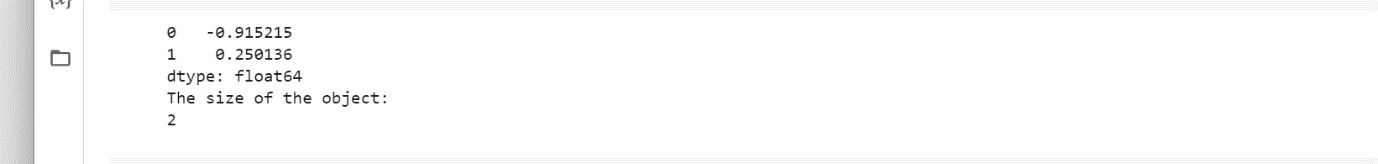
# PROGRAM-5

import pandas as pd import numpy as np s = pd.Series(np.random.randn(2)) print (s)

print ("The size of the object:")

print (s.size)

# OUTPUT

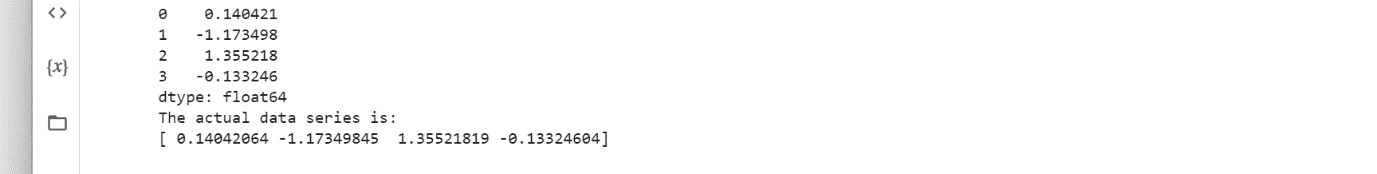


# PROGRAM-6

import pandas as pd import numpy as np s = pd.Series(np.random.randn(4)) print (s)

print ("The actual data series is:") print (s.values)

# OUTPUT



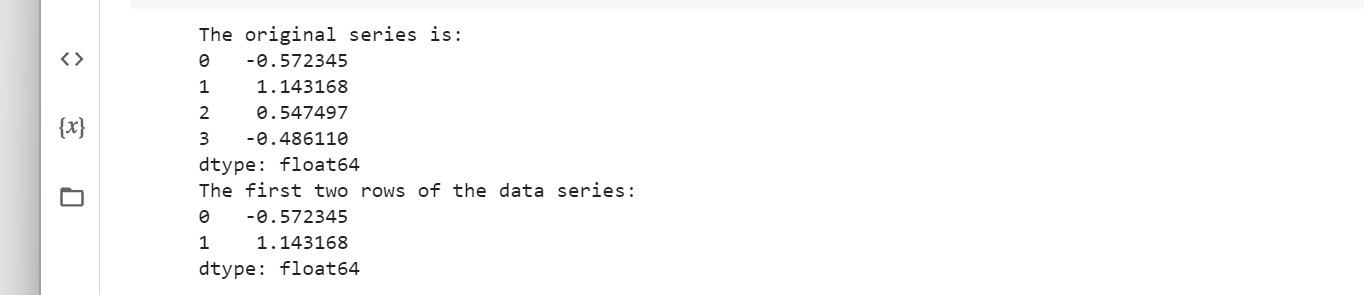
# PROGRAM-7

import pandas as pd import numpy as np s = pd.Series(np.random.randn(4)) print ("The original series is:")

print (s)

print ("The first two rows of the data series:") print (s.head(2))

# OUTPUT



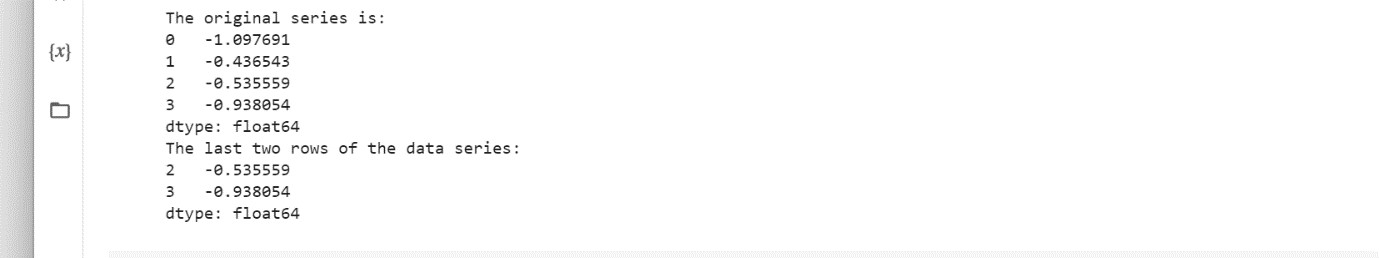
# PROGRAM-8

import pandas as pd import numpy as np s = pd.Series(np.random.randn(4)) print ("The original series is:")

print (s)

print ("The last two rows of the data series:") print (s.tail(2))

# OUTPUT



# PROGRAM-9

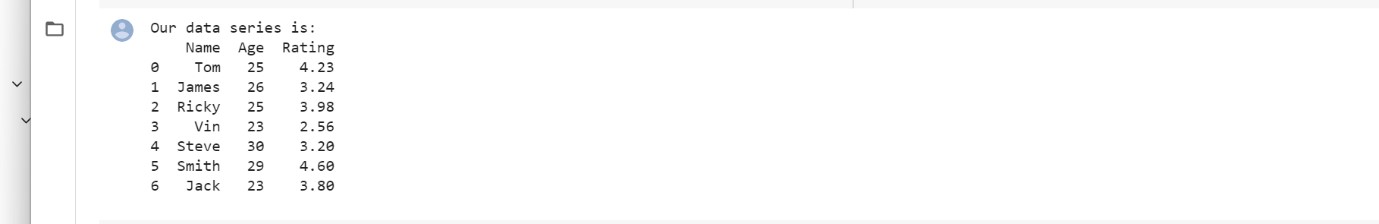
import pandas as pd import numpy as np

d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack']),

'Age':pd.Series([25,26,25,23,30,29,23]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])} df = pd.DataFrame(d) print ("Our data series is:") print (df)

# OUTPUT



# PROGRAM-10

import pandas as pd import numpy as np

d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack']),

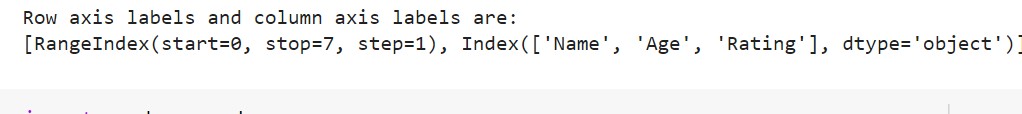
'Age':pd.Series([25,26,25,23,30,29,23]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])} df = pd.DataFrame(d)

print ("Row axis labels and column axis labels are:")

print (df.axes)

# OUTPUT



# PROGRAM-11

import pandas as pd import numpy as np

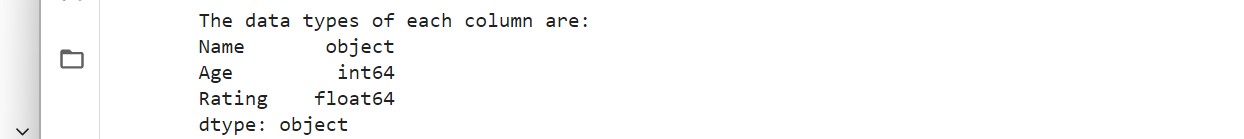
d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack']),

'Age':pd.Series([25,26,25,23,30,29,23]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])} df = pd.DataFrame(d)

print ("The data types of each column are:") print (df.dtypes)

# OUTPUT



# PROGRAM-12

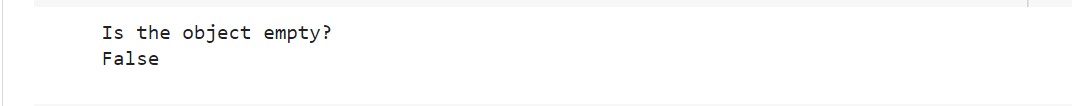
import pandas as pd import numpy as np

d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack']),

'Age':pd.Series([25,26,25,23,30,29,23]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])} df = pd.DataFrame(d) print ("Is the object empty?") print (df.empty)

# OUTPUT



# PROGRAM-13

import pandas as pd import numpy as np

d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack']),

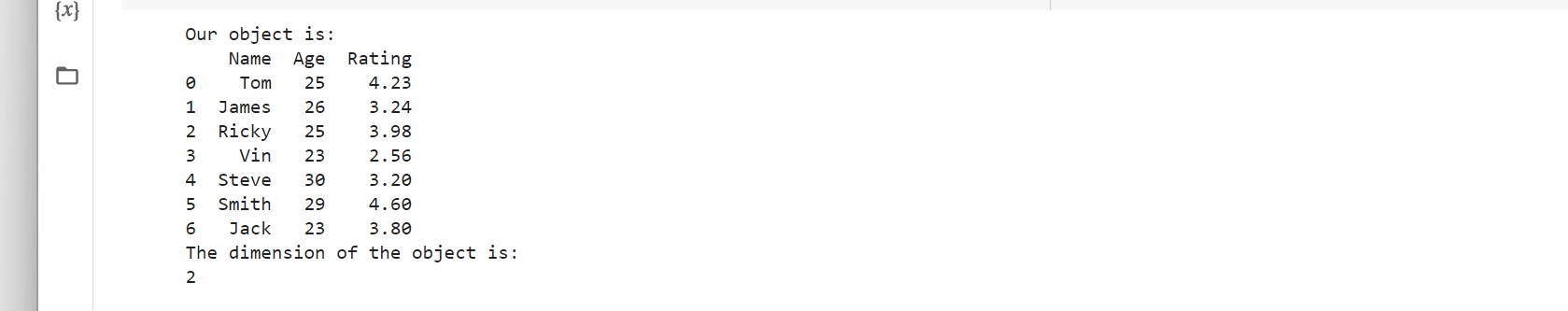
'Age':pd.Series([25,26,25,23,30,29,23]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])}

df = pd.DataFrame(d) print ("Our object is:") print (df)

print ("The dimension of the object is:") print (df.ndim)

# OUTPUT



# PROGRAM-14

import pandas as pd import numpy as np

d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack']),

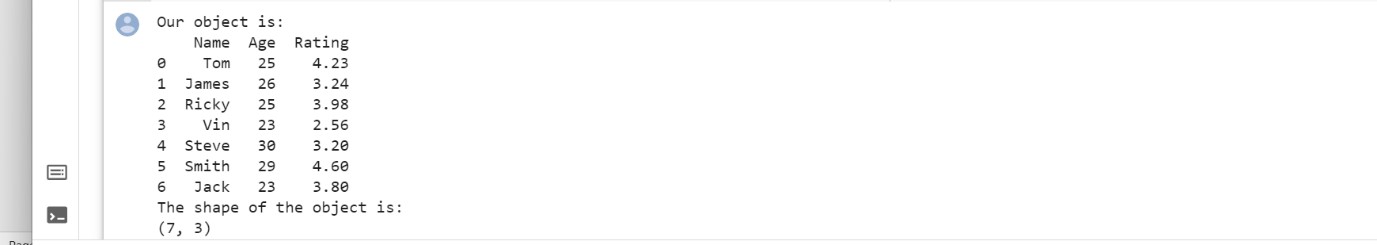
'Age':pd.Series([25,26,25,23,30,29,23]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])}

df = pd.DataFrame(d) print ("Our object is:") print (df)

print ("The shape of the object is:") print (df.shape)

# OUTPUT



# PROGRAM-15

import pandas as pd

import numpy as np

d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack']),

'Age':pd.Series([25,26,25,23,30,29,23]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])}

df = pd.DataFrame(d) print ("Our data frame is:")

print (df)

print ("The last two rows of the data frame is:") print (df.tail(2))

# OUTPUT



**DATA FRAMES:**

# PROGRAM-1

import pandas as pd data = [1,2,3,4,5]

df = pd.DataFrame(data)

print (df)

# OUTPUT

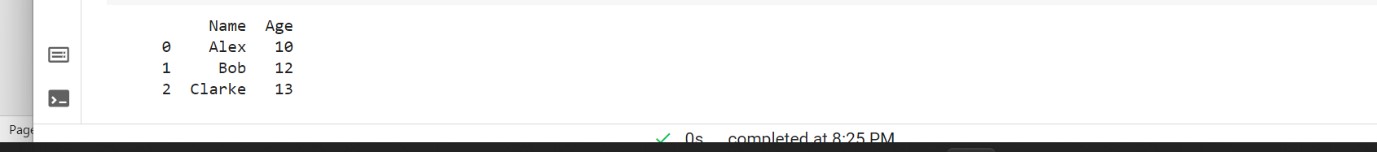


# PROGRAM-2

import pandas as pd data = [['Alex',10],['Bob',12],['Clarke',13]]

df = pd.DataFrame(data,columns=['Name','Age']) print (df)

# OUTPUT



# PROGRAM-3

import pandas as pd

data = [['Alex',10],['Bob',12],['Clarke',13]]

df = pd.DataFrame(data,columns=['Name','Age']) print (df)

# OUTPUT



# PROGRAM-4

import pandas as pd

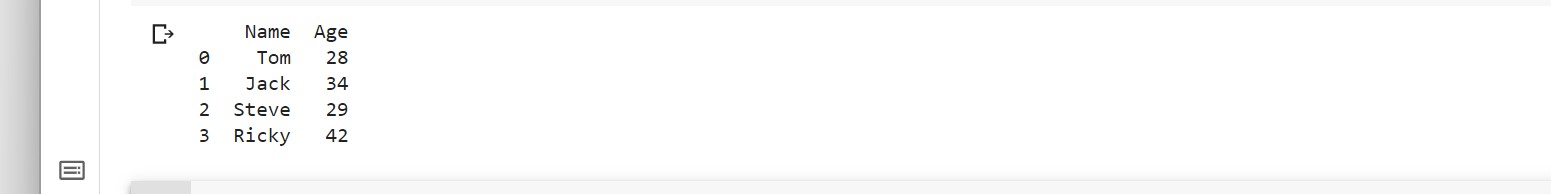
data = {'Name':['Tom', 'Jack', 'Steve', 'Ricky'],

'Age':[28,34,29,42]}

df = pd.DataFrame(data)

print (df)

# OUTPUT

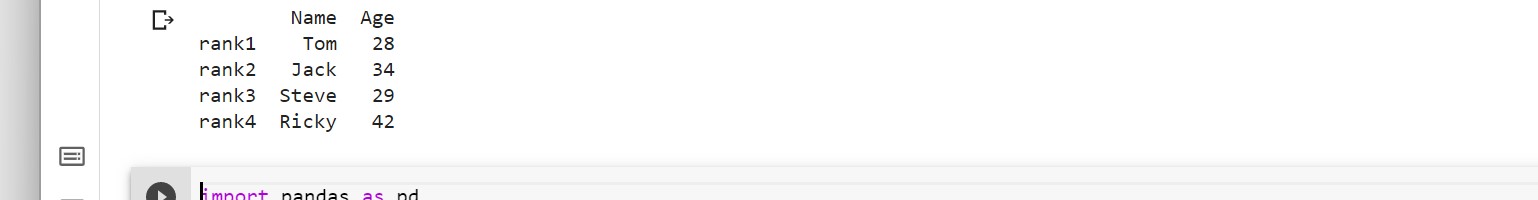


# PROGRAM-5

import pandas as pd

data = {'Name':['Tom', 'Jack', 'Steve', 'Ricky'],'Age':[28,34,29,42]} df = pd.DataFrame(data, index=['rank1','rank2','rank3','rank4']) print (df)

# OUTPUT



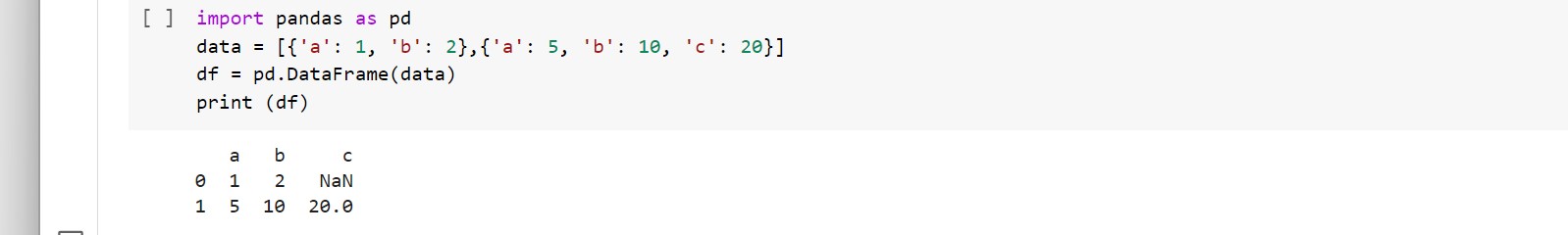
# PROGRAM-6

import pandas as pd

data = [{'a': 1, 'b': 2},{'a': 5, 'b': 10, 'c': 20}] df = pd.DataFrame(data)

print (df)

# OUTPUT



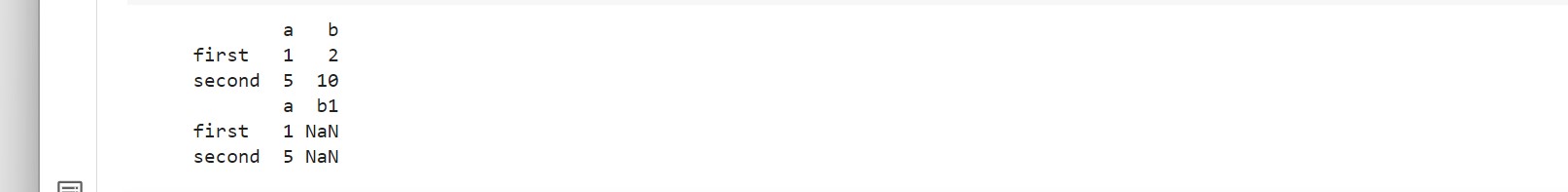
# PROGRAM-7

import pandas as pd

data = [{'a': 1, 'b': 2},{'a': 5, 'b': 10, 'c': 20}]

df1 = pd.DataFrame(data, index=['first', 'second'], columns=['a', 'b']) df2 = pd.DataFrame(data, index=['first', 'second'], columns=['a', 'b1']) print (df1) print (df2)

# OUTPUT



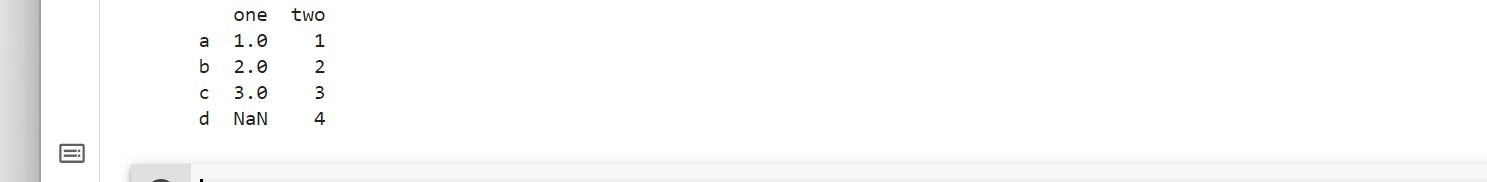
# PROGRAM-8

import pandas as pd

d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']), 'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}

df = pd.DataFrame(d) print (df)

# OUTPUT



# PROGRAM-9

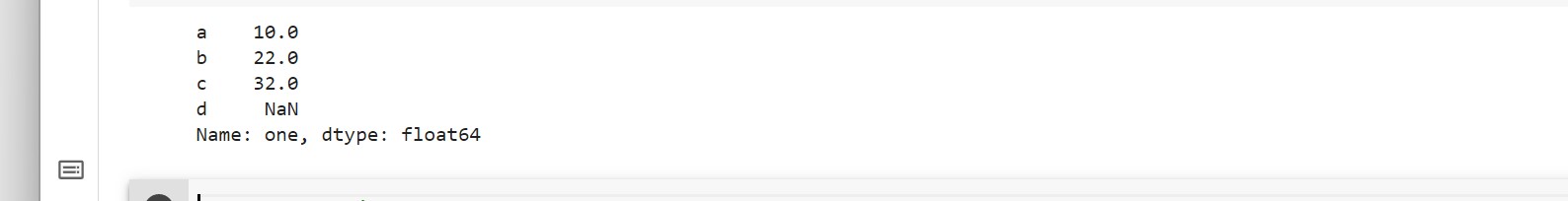
import pandas as pd

d = {'one' : pd.Series([10, 22, 32], index=['a', 'b', 'c']),

'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}

df = pd.DataFrame(d) print (df['one'])

# OUTPUT



# PROGRAM-10

import pandas as pd

de = {'one' : pd.Series([10, 22, 32,44], index=['a', 'b', 'c','d']), 'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}

df = pd.DataFrame(de) print (df['one'])

# OUTPUT



# PROGRAM-11

import pandas as pd

d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']), 'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}

df = pd.DataFrame(d)

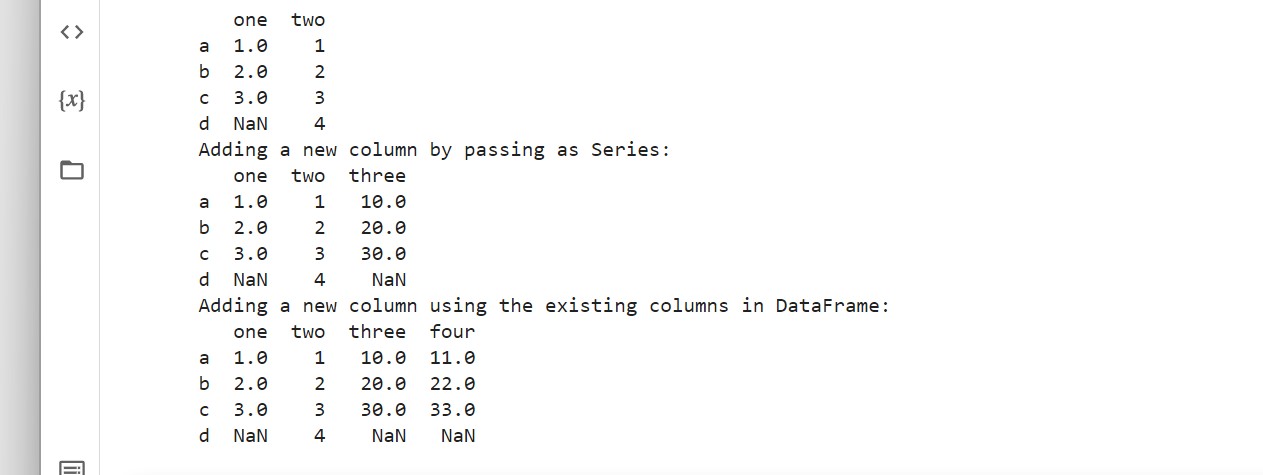
print (df)

print ("Adding a new column by passing as Series:") df['three']=pd.Series([10,20,30],index=['a','b','c']) print (df)

print ("Adding a new column using the existing columns in DataFrame:") df['four']=df['two']+df['three']

print (df)

# OUTPUT



# PROGRAM-12

import pandas as pd

d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']), 'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd']),

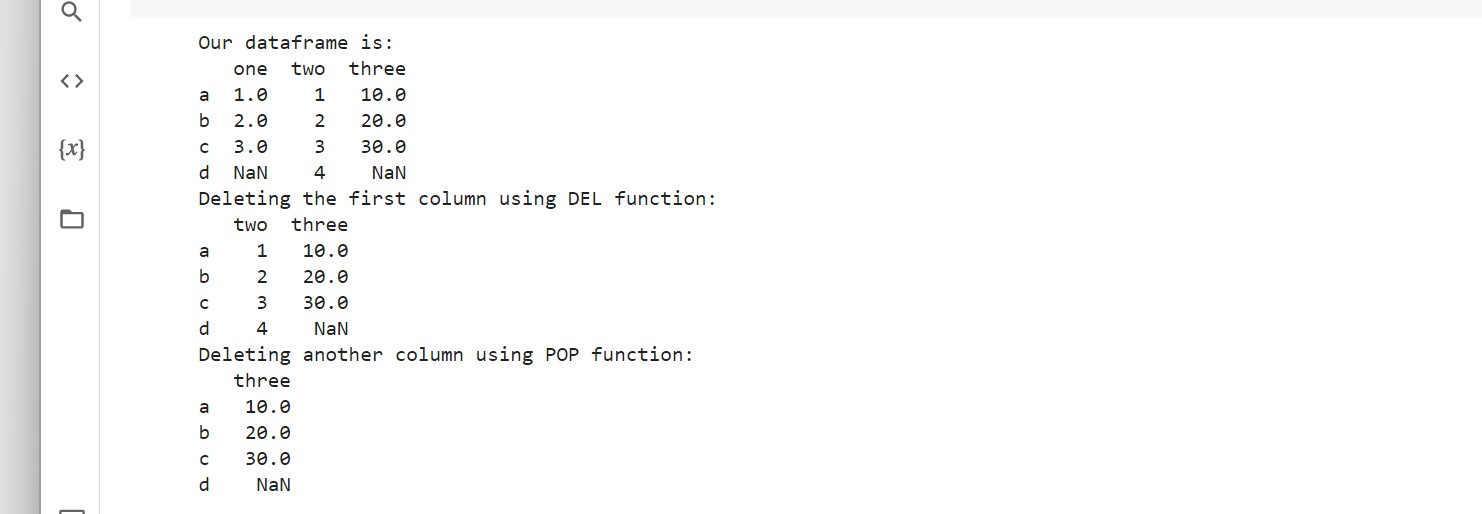
'three' : pd.Series([10,20,30], index=['a','b','c'])}

df = pd.DataFrame(d) print ("Our dataframe is:") print (df)

print ("Deleting the first column using DEL function:") del df['one'] print (df)

print ("Deleting another column using POP function:") df.pop('two') print (df)

# OUTPUT



# PROGRAM-13

import pandas as pd

d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']), 'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}

df = pd.DataFrame(d) print (df.loc['b'])

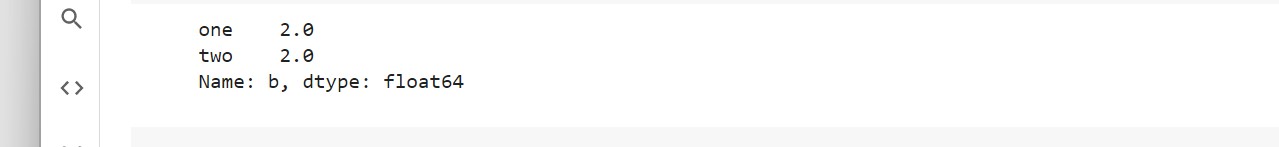
# PROGRAM-14

import pandas as pd

d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']), 'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}

df = pd.DataFrame(d) print (df.iloc[2])

# OUTPUT



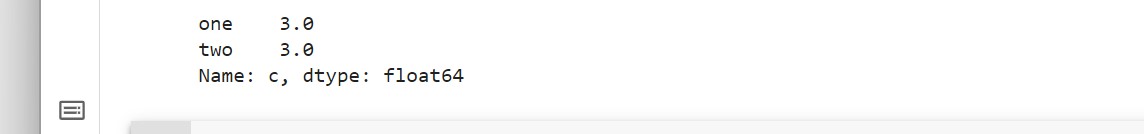
# PROGRAM-15

import pandas as pd

d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']), 'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}

df = pd.DataFrame(d) print (df[2:4])

# OUTPUT



# PROGRAM-16

import pandas as pd

df = pd.DataFrame([[1, 2], [3, 4]], columns = ['a','b']) df2 = pd.DataFrame([[5, 6], [7, 8]], columns = ['a','b']) print (df) print("after appending") df = df.append(df2)

print (df)

# OUTPUT



# PROGRAM-17

import pandas as pd

df = pd.DataFrame([[1, 2], [3, 4]], columns = ['a','b']) df2 = pd.DataFrame([[5, 6], [7, 8]], columns = ['a','b'])

df = df.append(df2)

df = df.drop(0)

print (df)

# OUTPUT



**DATA WRANGLING:**

# PROGRAM-1

import pandas as pd

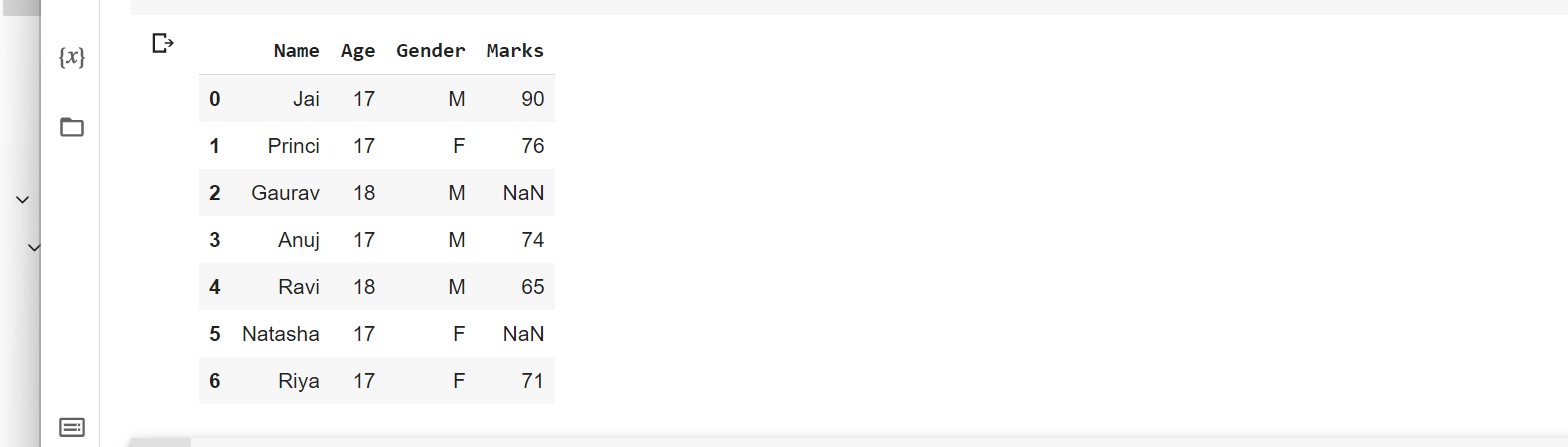
data = {'Name': ['Jai', 'Princi', 'Gaurav','Anuj', 'Ravi', 'Natasha', 'Riya'],

'Age': [17, 17, 18, 17, 18, 17, 17],

'Gender': ['M', 'F', 'M', 'M', 'M', 'F', 'F'], 'Marks': [90, 76, 'NaN', 74, 65, 'NaN', 71]} df = pd.DataFrame(data)

Df

# OUTPUT



# PROGRAM-2

c = avg = 0 for ele in df['Marks']: if str(ele).isnumeric():

c += 1 avg += ele avg /= c

df = df.replace(to\_replace="NaN",

value=avg)

Df

# OUTPUT



# PROGRAM-3

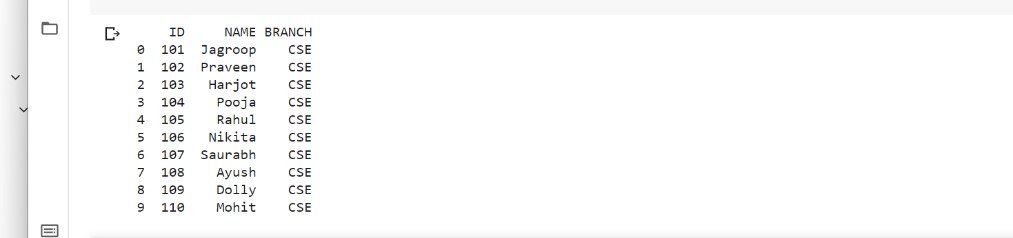
import pandas as pd details = pd.DataFrame({

'ID': [101, 102, 103, 104, 105, 106,107, 108, 109, 110],

'NAME': ['Jagroop', 'Praveen', 'Harjot','Pooja', 'Rahul', 'Nikita', 'Saurabh', 'Ayush', 'Dolly', "Mohit"],

'BRANCH': ['CSE', 'CSE', 'CSE', 'CSE', 'CSE','CSE', 'CSE', 'CSE', 'CSE', 'CSE']}) print(details)

# OUTPUT



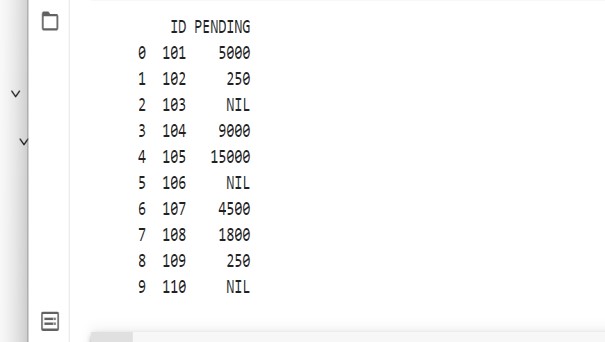
# PROGRAM-4

import pandas as pd fees\_status = pd.DataFrame(

{'ID': [101, 102, 103, 104, 105,106, 107, 108, 109, 110],

'PENDING': ['5000', '250', 'NIL','9000', '15000', 'NIL','4500', '1800', '250', 'NIL']}) print(fees\_status)

# OUTPUT



# PROGRAM-5

import pandas as pd details = pd.DataFrame({

'ID': [101, 102, 103, 104, 105,106, 107, 108, 109, 110],

'NAME': ['Jagroop', 'Praveen', 'Harjot','Pooja', 'Rahul', 'Nikita','Saurabh', 'Ayush', 'Dolly', "Mohit"],

'BRANCH': ['CSE', 'CSE', 'CSE', 'CSE', 'CSE', 'CSE', 'CSE', 'CSE', 'CSE', 'CSE']})

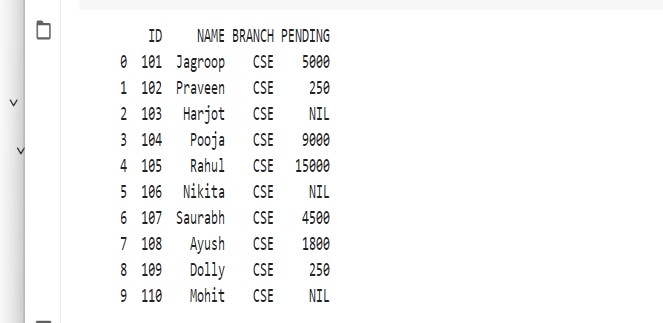
fees\_status = pd.DataFrame(

{'ID': [101, 102, 103, 104, 105,106, 107, 108, 109, 110],

'PENDING': ['5000', '250', 'NIL','9000', '15000', 'NIL', '4500', '1800', '250', 'NIL']})

print(pd.merge(details, fees\_status, on='ID'))

# OUTPUT



# PROGRAM-6

import pandas as pd

car\_selling\_data = {'Brand': ['Maruti', 'Maruti', 'Maruti','Maruti', 'Hyundai', 'Hyundai','Toyota', 'Mahindra', ' Mahindra', 'Ford', 'Toyota', 'Ford'],

'Year': [2010, 2011, 2009, 2013,2010, 2011, 2011, 2010,2013, 2010, 2010, 2011],

'Sold': [6, 7, 9, 8, 3, 5,2, 8, 7, 2, 4, 2]}

df = pd.DataFrame(car\_selling\_data) print(df)

# OUTPUT



# PROGRAM-7

import pandas as pd

car\_selling\_data = {'Brand': ['Maruti', 'Maruti', 'Maruti','Maruti', 'Hyundai', 'Hyundai','Toyota', 'Mahindra', ' Mahindra', 'Ford', 'Toyota', 'Ford'],

'Year': [2010, 2011, 2009, 2013,2010, 2011, 2011, 2010,2013, 2010, 2010, 2011],

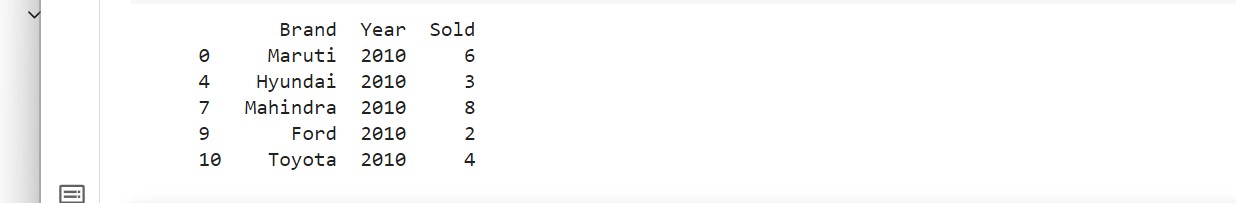
'Sold': [6, 7, 9, 8, 3, 5,2, 8, 7, 2, 4, 2]}

df = pd.DataFrame(car\_selling\_data)

grouped = df.groupby('Year')

print(grouped.get\_group(2010))

# OUTPUT



# PROGRAM-8

import pandas as pd

student\_data = {'Name': ['Amit', 'Praveen', 'Jagroop','Rahul', 'Vishal', 'Suraj', 'Rishab', 'Satyapal', 'Amit', 'Rahul', 'Praveen', 'Amit'],

'Roll\_no': [23, 54, 29, 36, 59, 38,12, 45, 34, 36, 54, 23],

'Email': ['xxxx@gmail.com', 'xxxxxx@gmail.com', 'xxxxxx@gmail.com', 'xx@gmail.com', 'xxxx@gmail.com', 'xxxxx@gmail.com', 'xxxxx@gmail.com', 'xxxxx@gmail.com','xxxxx@gmail.com', ' xxxxxx@gmail.com', 'xxxxxxxxxx@gmail.com', 'xxxxxxxxxx@gmail.com']}

df = pd.DataFrame(student\_data)

print(df)

# OUTPUT



# PROGRAM-9

import pandas as pd

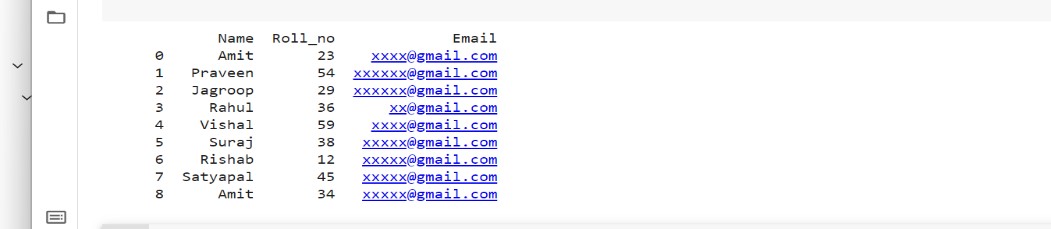
student\_data = {'Name': ['Amit', 'Praveen', 'Jagroop','Rahul', 'Vishal', 'Suraj', 'Rishab', 'Satyapal', 'Amit', 'Rahul', 'Praveen', 'Amit'],

'Roll\_no': [23, 54, 29, 36, 59, 38,12, 45, 34, 36, 54, 23],

'Email': ['xxxx@gmail.com', 'xxxxxx@gmail.com', 'xxxxxx@gmail.com', 'xx@gmail.com', 'xxxx@gmail.com', 'xxxxx@gmail.com', 'xxxxx@gmail.com', 'xxxxx@gmail.com','xxxxx@gmail.com', ' xxxxxx@gmail.com', 'xxxxxxxxxx@gmail.com', 'xxxxxxxxxx@gmail.com']}

df = pd.DataFrame(student\_data) non\_duplicate = df[~df.duplicated('Roll\_no')] print(non\_duplicate)

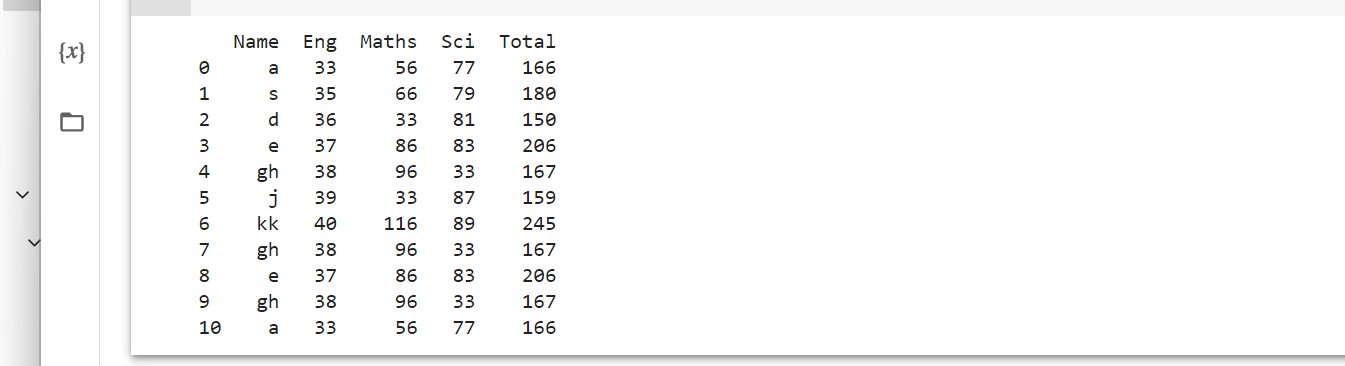
# OUTPUT



**PANDAS-DATA FILTERING:**

d=pd.read\_excel("C:\\Users\Shabnam\Desktop\duplicate.xlsx") df=pd.DataFrame(d) print(df)

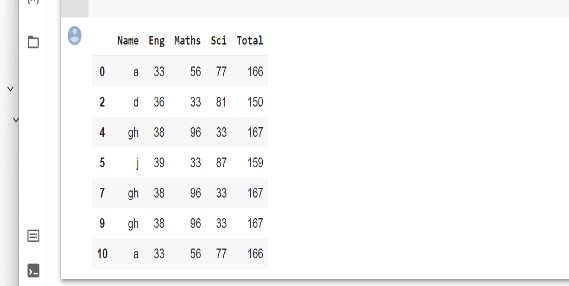
# OUTPUT



**PROGRAM-2**

df.loc[df['Total']<170]

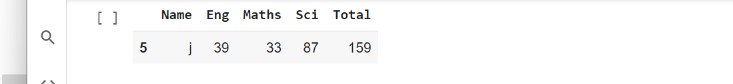
# OUTPUT



**PROGRAM-3**

df.loc[(df['Eng']> 38) & (df['Maths']<50)]

# OUTPUT



**PROGRAM-4**

df.loc[(df['Eng']> 38) | (df['Maths']<50)]

# OUTPUT



**PROGRAM-5**

df.loc[df['Name'].str.contains("k")]

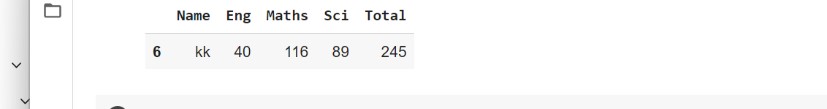
# OUTPUT



**PROGRAM-6**

df.loc[df['Name'].str.startswith("k")]

# OUTPUT



**PROGRAM-7**

df.loc[df['Name'].str.endswith("k")]

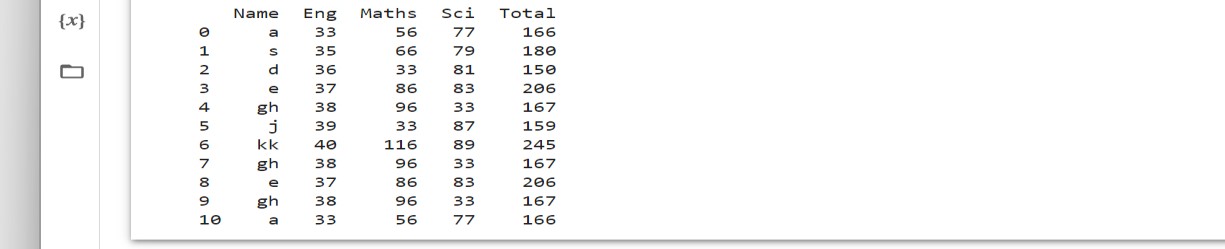
# OUTPUT



**PANDAS-DUPLICATE**

d=pd.read\_excel("C:\\Users\Shabnam\Desktop\duplicate.xlsx") df=pd.DataFrame(d) print(df)

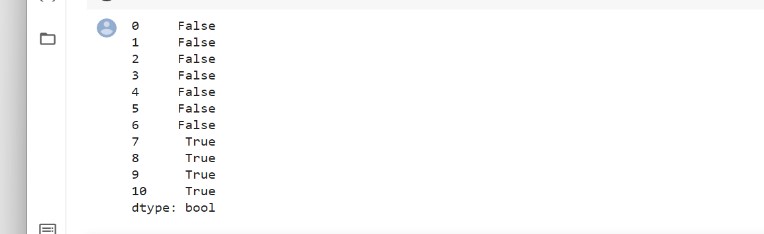
# OUTPUT



**PROGRAM-2**

df.duplicated()

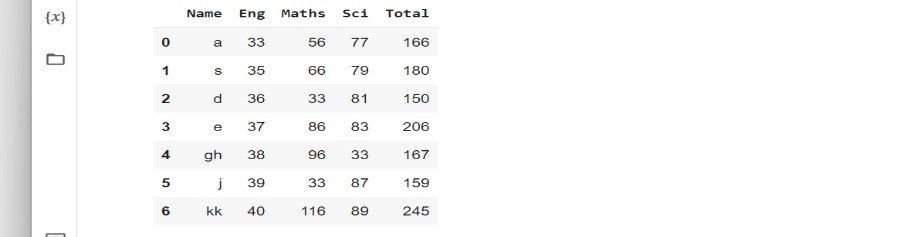
# OUTPUT



**PROGRAM-3**

df.drop\_duplicates()

# OUTPUT



# PROGRAM-4

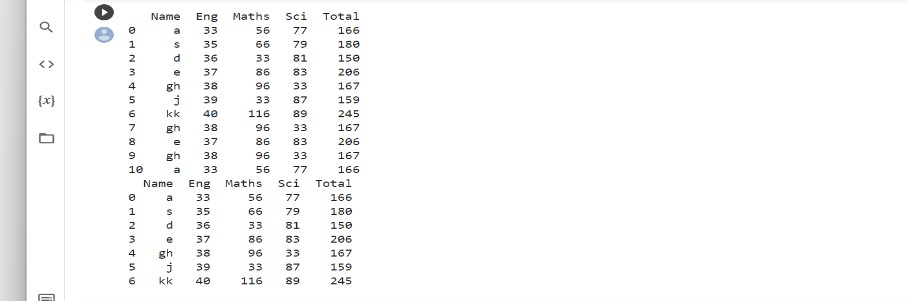
import pandas as pd

d=pd.read\_excel("C:\\Users\Shabnam\Desktop\duplicate.xlsx") df=pd.DataFrame(d) print(df)

df.drop\_duplicates(inplace=True)

print(df)

# OUTPUT



**DESCRIPTIVE STATS:**

import numpy as np

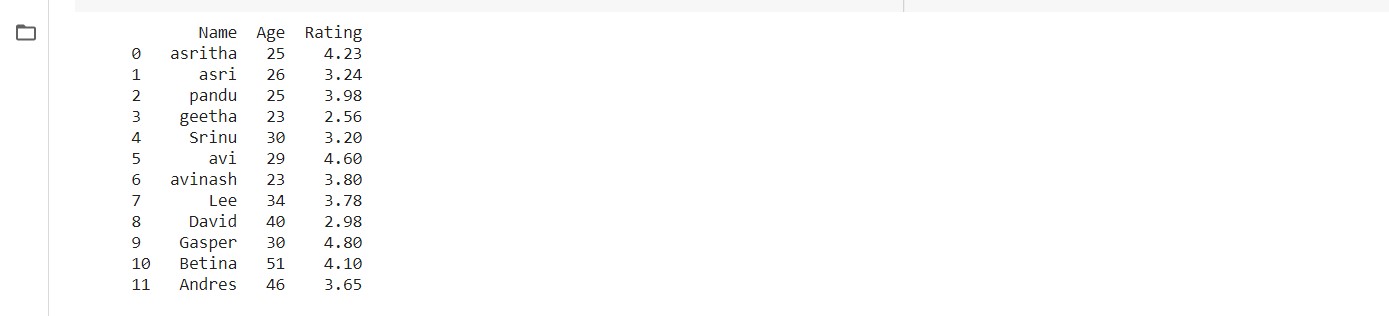
d = {'Name':pd.Series(['asritha','asri','pandu','geetha','Srinu','avi','avinash' 'Lee','David','Gasper','Betina', 'Andres']),

'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65]) }

df = pd.DataFrame(d) print (df)

# OUTPUT



# PROGRAM-2

import pandas as pd import numpy as np

d = {'Name':pd.Series(['asritha','asri','pandu','geetha','Srinu','avi','avinash',

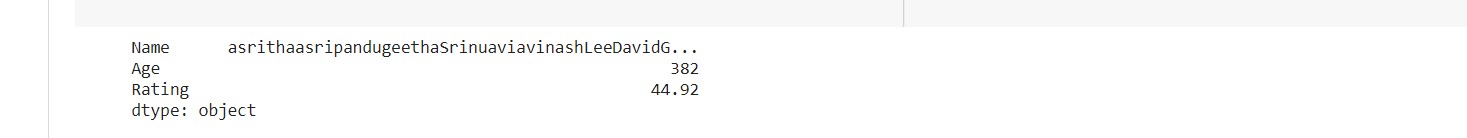
'Lee','David','Gasper','Betina','Andres']),

'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65]) }

df = pd.DataFrame(d) print (df.sum())

# OUTPUT



# PROGRAM-3

import pandas as pd import numpy as np

d = {'Name':pd.Series(['asritha','asri','pandu','geetha','Srinu','avinash','avi',

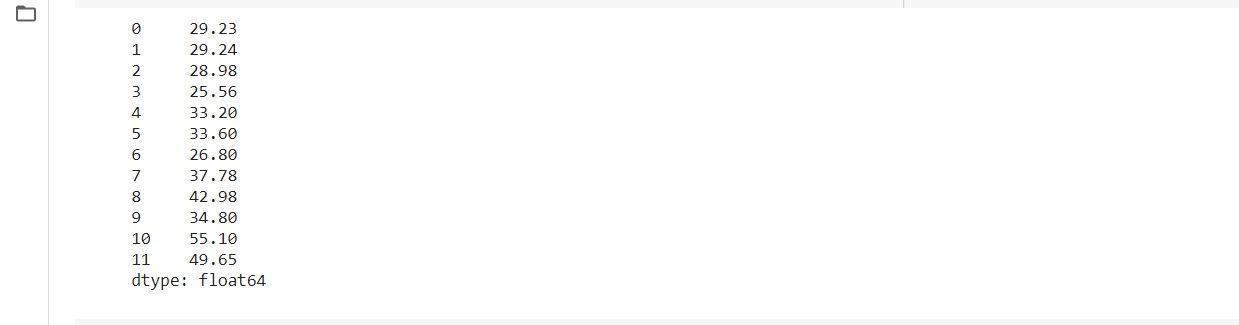
'Lee','David','Gasper','Betina','Andres']),

'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65]) }

df = pd.DataFrame(d) print (df.sum(1))

# OUTPUT



# PROGRAM-4

import pandas as pd import numpy as np

d = {'Name':pd.Series(['asritha','asri','pandu','geetha','Srinu','avinash','avi',

'Lee','David','Gasper','Betina','Andres']),

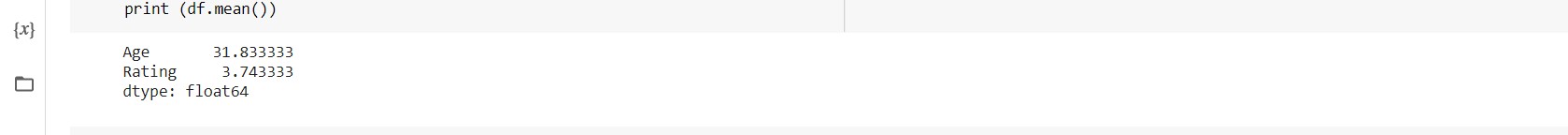
'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65]) }

df = pd.DataFrame(d)

print (df.sum(1))

# OUTPUT



# PROGRAM-5

import pandas as pd import numpy as np

d = {'Name':pd.Series(['asritha','asri','pandu','geetha','srinu','avinash','avi',

'Lee','David','Gasper','Betina','Andres']),

'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65]) }

df = pd.DataFrame(d) print (df.mean())

# OUTPUT



# PROGRAM-6

import pandas as pd import numpy as np

d = {'Name':pd.Series(['asritha','asri','pandu','geetha','Srinu','avi','avinash',

'Lee','David','Gasper','Betina','Andres']),

'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65]) }

df = pd.DataFrame(d) print (df.std())

# OUTPUT



# PROGRAM-7

import pandas as pd import numpy as np

d = {'Name':pd.Series(['asritha','asri','pandu','srinu','geetha','avinash','avi',

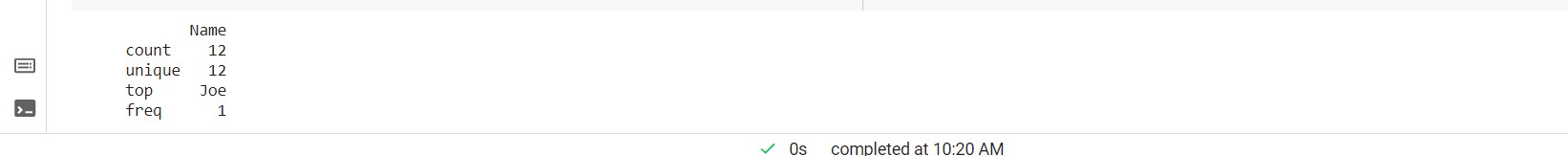
'Lee','David','Gasper','Betina','Andres']),

'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65]) }

df = pd.DataFrame(d) print (df.describe())

# OUTPUT



# PROGRAM-8

import pandas as pd import numpy as np

d = {'Name':pd.Series(['asritha','asri','pandu','geetha','srinu','avi','avinash',

'Lee','David','Joe','Tom','Vin']),

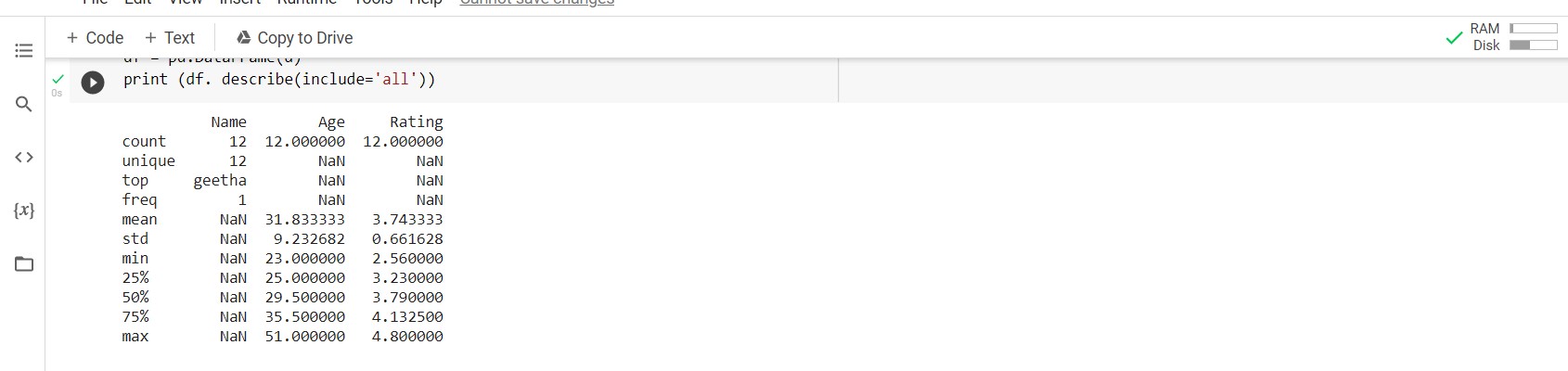
'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65]) }

df = pd.DataFrame(d)

print (df.describe(include=['object']))

# OUTPUT



# PROGRAM-9

import pandas as pd import numpy as np

d = {'Name':pd.Series(['asritha','asri','pandu','geetha','Srinu','avinash','avi',

'Lee','David','Gasper','Betina','Andres']),

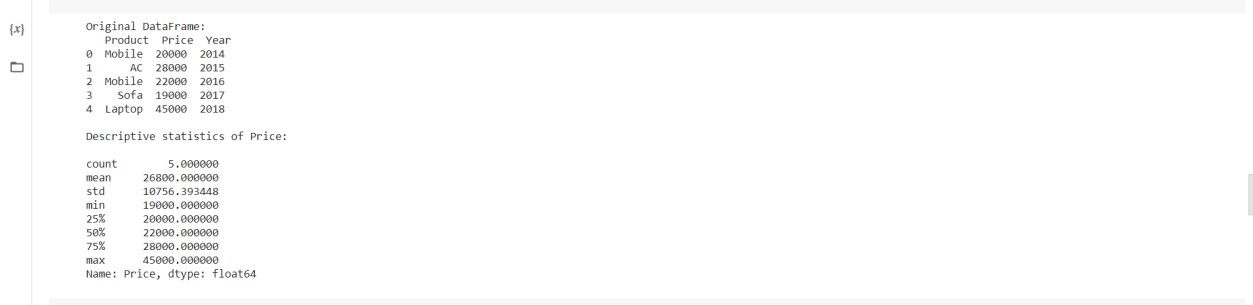
'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),

'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65]) }

df = pd.DataFrame(d)

print (df. describe(include='all'))

# OUTPUT



# PROGRAM-10

from pandas import DataFrame

# Create DataFrame

cart = {'Product': ['Mobile', 'AC', 'Mobile', 'Sofa', 'Laptop'],

'Price': [20000, 28000, 22000, 19000, 45000],

'Year': [2014, 2015, 2016, 2017, 2018]

}

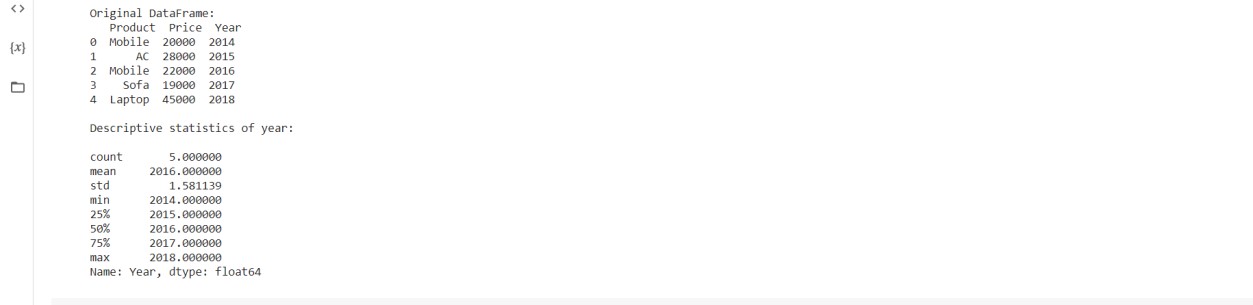
df = DataFrame(cart, columns = ['Product', 'Price', 'Year'])

# Original DataFrame

print("Original DataFrame:\n", df)

# Describing descriptive statistics of Price print("\nDescriptive statistics of Price:\n") stats = df['Price'].describe() print(stats)

# OUTPUT



# PROGRAM-11

cart = {'Product': ['Mobile', 'AC', 'Mobile', 'Sofa', 'Laptop'],

'Price': [20000, 28000, 22000, 19000, 45000],

'Year': [2014, 2015, 2016, 2017, 2018]

}

df = DataFrame(cart, columns = ['Product', 'Price', 'Year']) print("Original DataFrame:\n", df) print("\nDescriptive statistics of year:\n") stats = df['Year'].describe() print(stats)

# OUTPUT



# PROGRAM-12

cart = {'Product': ['Mobile', 'AC', 'Mobile', 'Sofa', 'Laptop'],

'Price': [20000, 28000, 22000, 19000, 45000],

'Year': [2014, 2015, 2016, 2017, 2018]

}

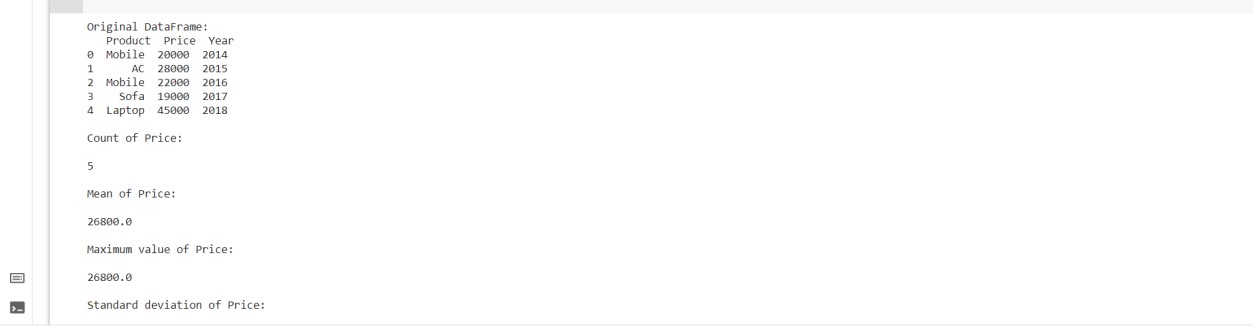
df = DataFrame(cart, columns = ['Product', 'Price', 'Year'])

print("Original DataFrame:\n", df)

print("\nDescriptive statistics of whole dataframe:\n")

stats = df.describe(include = 'all') print(stats)

# OUTPUT



# PROGRAM-13

cart = {'Product': ['Mobile', 'AC', 'Mobile', 'Sofa', 'Laptop'],

'Price': [20000, 28000, 22000, 19000, 45000],

'Year': [2014, 2015, 2016, 2017, 2018]

}

df = DataFrame(cart, columns = ['Product', 'Price', 'Year'])

print("Original DataFrame:\n", df)

print("\nCount of Price:\n") counts = df['Price'].count() print(counts)

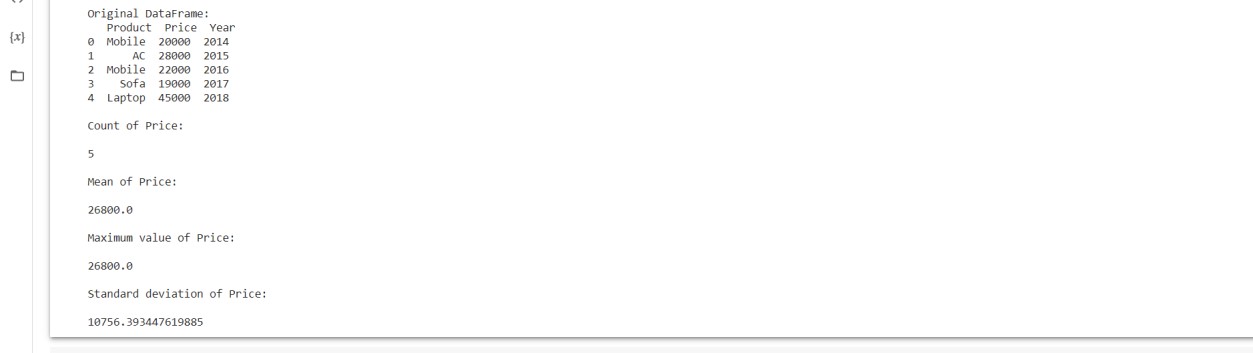
print("\nMean of Price:\n") m = df['Price'].mean() print(m)

print("\nMaximum value of Price:\n") mx = df['Price'].max() print(m)

print("\nStandard deviation of Price:\n")

sd = df['Price'].std() print(sd)

# OUTPUT



**PANDAS FUNCTIONS:**

# PROGRAM-1

import pandas as pd import numpy as np

def adder(ele1,ele2): return ele1+ele2

df = pd.DataFrame(np.random.randn(5,3),columns=['col1','col2','col3']) print(df)

print("printing the results of pipe") print(df.pipe(adder,2))

# OUTPUT



# PROGRAM-2

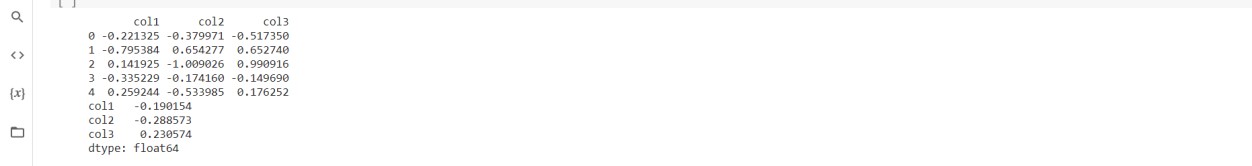
import pandas as pd import numpy as np

def adder(ele1,ele2): return ele1+ele2

df = pd.DataFrame(np.random.randn(5,3),columns=['col1','col2','col3']) print(df)

print (df.apply(np.mean))

# OUTPUT



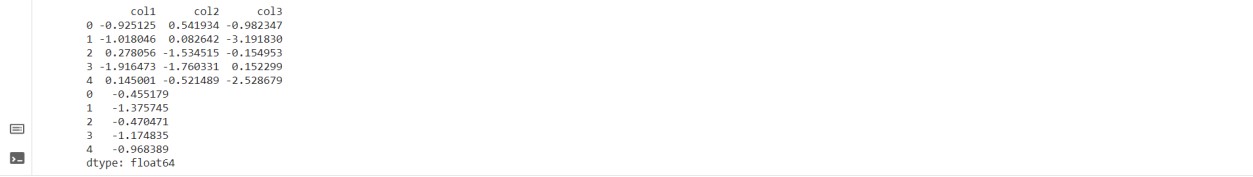
# PROGRAM-3

import pandas as pd import numpy as np

df = pd.DataFrame(np.random.randn(5,3),columns=['col1','col2','col3']) print(df)

print (df.apply(np.mean,axis=1))

# OUTPUT



# PROGRAM-4

import pandas as pd import numpy as np

df = pd.DataFrame(np.random.randn(5,3),columns=['col1','col2','col3']) print(df)

print(df['col1'].map(lambda x:x\*100))

# OUTPUT



# PROGRAM-5

df=pd.DataFrame({'id':[1,2,3,4,5],'name':['abc','defg','ghs','eeee','wwww'],'age':[11,22,33,44,55], 'income':[

9999,8888,7777,6666,5555]}) print(df)

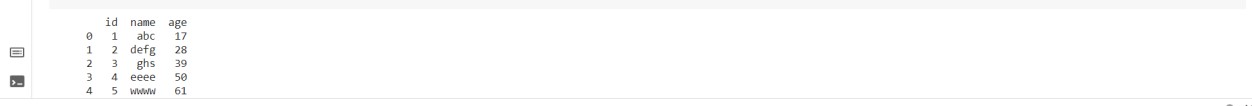
# OUTPUT



# PROGRAM-6

df['age']=df.apply(lambda x: x['age']+3, axis=1) print(df)

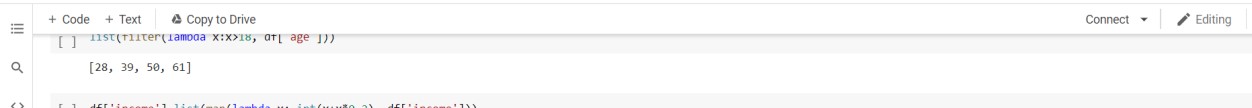
# OUTPUT



**PROGRAM-7**

list(filter(lambda x:x>18, df['age']))

# OUTPUT



# PROGRAM-8

df['income']=list(map(lambda x: int(x+x\*0.2), df['income'])) print(df)

# OUTPUT

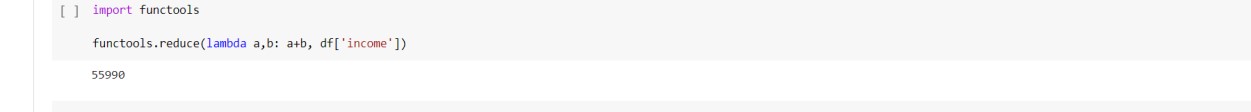


# PROGRAM-9

import functools

functools.reduce(lambda a,b: a+b, df['income'])

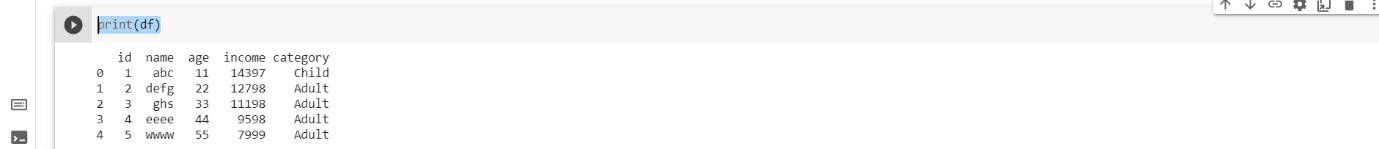
# OUTPUT



# PROGRAM-10

df['category']=df['age'].apply(lambda x: 'Adult' if x>=18 else 'Child') print(df)

# OUTPUT



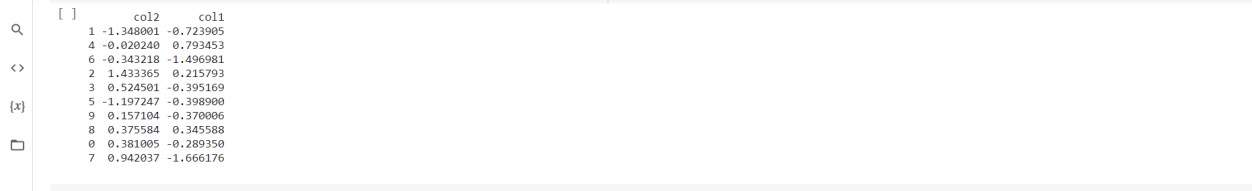
**PANDAS SORTING:**

# PROGRAM-1

import pandas as pd import numpy as np

unsorted\_df=pd.DataFrame(np.random.randn(10,2),index=[1,4,6,2,3,5,9,8,0,7],columns=['col2','col1']) print (unsorted\_df)

# OUTPUT



# PROGRAM-2

import pandas as pd import numpy as np

unsorted\_df = pd.DataFrame(np.random.randn(10,2),index=[1,4,6,2,3,5,9,8,0,7],columns = ['col2','col1'])

sorted\_df=unsorted\_df.sort\_index() print (sorted\_df)

# OUTPUT



# PROGRAM-3

import pandas as pd import numpy as np

unsorted\_df = pd.DataFrame(np.random.randn(10,2),index=[1,4,6,2,3,5,9,8,0,7],columns = ['col2','col1'])

sorted\_df = unsorted\_df.sort\_index(ascending=False) print (sorted\_df)

# OUTPUT



# PROGRAM-4

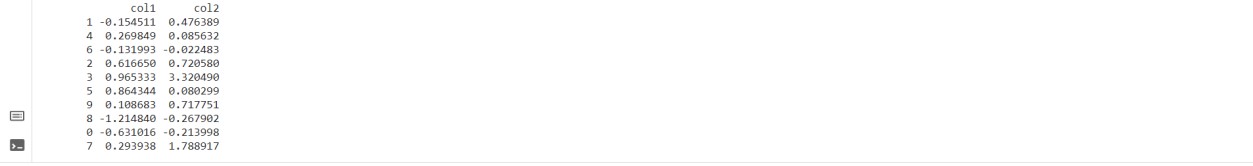
import pandas as pd import numpy as np

unsorted\_df = pd.DataFrame(np.random.randn(10,2),index=[1,4,6,2,3,5,9,8,0,7],columns = ['col2','col1'])

sorted\_df=unsorted\_df.sort\_index(axis=1)

print (sorted\_df)

# OUTPUT



# PROGRAM-5

import pandas as pd import numpy as np

unsorted\_df = pd.DataFrame({'col1':[2,1,1,1],'col2':[1,3,2,4]}) sorted\_df = unsorted\_df.sort\_values(by='col1')

print (sorted\_df)

# OUTPUT



# PROGRAM-6

import pandas as pd import numpy as np

unsorted\_df = pd.DataFrame({'col1':[2,1,1,1],'col2':[1,3,2,4]}) sorted\_df = unsorted\_df.sort\_values(by=['col1','col2'])

print (sorted\_df)

# OUTPUT



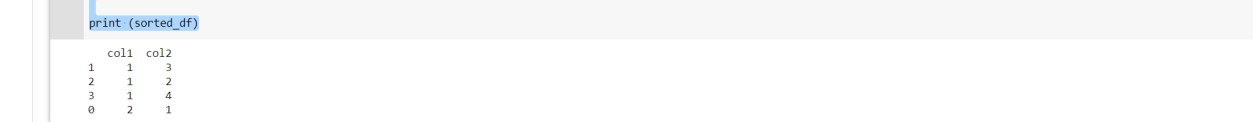
# PROGRAM-7

import pandas as pd import numpy as np

unsorted\_df = pd.DataFrame({‘col1’:[2,1,1,1],’col2’:[1,3,2,4]}) sorted\_df = unsorted\_df.sort\_values(by=’col1’ ,kind=’mergesort’)

print (sorted\_df)

# OUTPUT



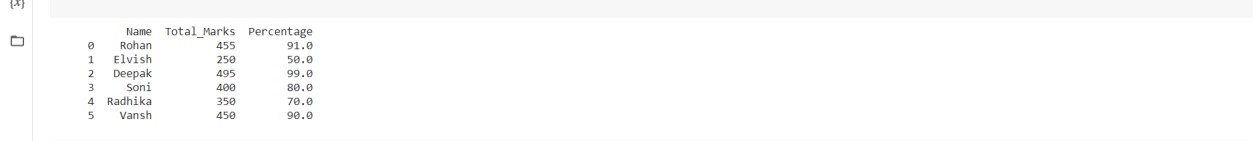
**PANDAS DATALAMBDA**

# PROGRAM-1

import pandas as pd values= [['Rohan',455],['Elvish',250],['Deepak',495], ['Soni',400],['Radhika',350],['Vansh',450]]

df = pd.DataFrame(values,columns=['Name','Total\_Marks']) df = df.assign(Percentage = lambda x: (x['Total\_Marks'] /500 \* 100)) print(df)

# OUTPUT



# PROGRAM-2

import pandas as pd

values\_list = [[15, 2.5, 100], [20, 4.5, 50], [25, 5.2, 80],

[45, 5.8, 48], [40, 6.3, 70], [41, 6.4, 90],

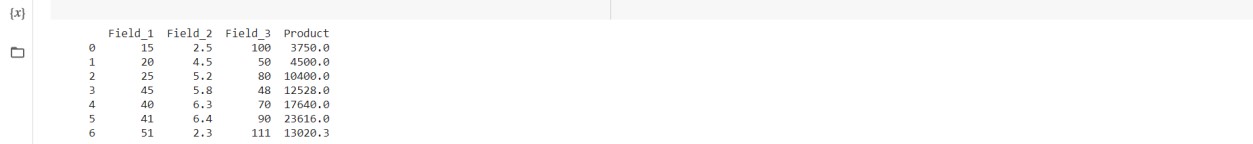
[51, 2.3, 111]]

df = pd.DataFrame(values\_list, columns=['Field\_1', 'Field\_2', 'Field\_3'])

df = df.assign(Product=lambda x: (x['Field\_1'] \* x['Field\_2'] \* x['Field\_3']))

print(df)

# OUTPUT



# PROGRAM-3

import pandas as pd import numpy as np

values\_list = [[15, 2.5, 100], [20, 4.5, 50], [25, 5.2, 80],

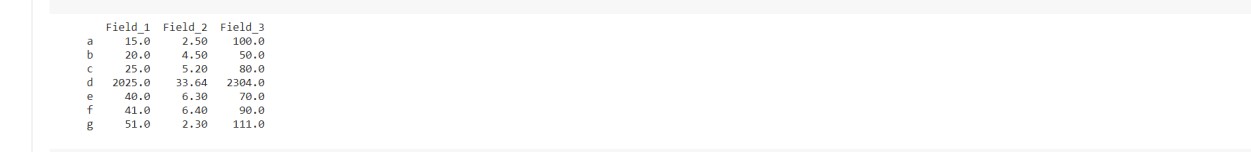
[45, 5.8, 48], [40, 6.3, 70], [41, 6.4, 90],

[51, 2.3, 111]]

df = pd.DataFrame(values\_list, columns=['Field\_1', 'Field\_2', 'Field\_3'], index=['a', 'b', 'c', 'd', 'e', 'f', 'g'])

df = df.apply(lambda x: np.square(x) if x.name == 'd' else x, axis=1) print(df)

# OUTPUT



# PROGRAM-4

import pandas as pd import numpy as np

values\_list = [[15, 2.5, 100], [20, 4.5, 50], [25, 5.2, 80],

[45, 5.8, 48], [40, 6.3, 70], [41, 6.4, 90],

[51, 2.3, 111]]

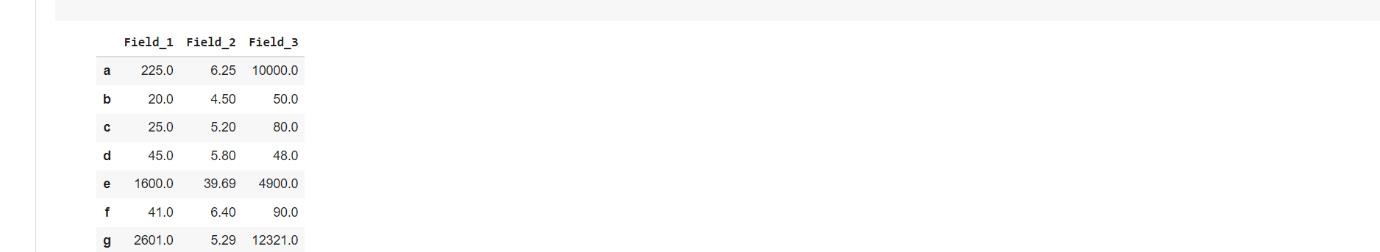
df = pd.DataFrame(values\_list, columns=['Field\_1', 'Field\_2', 'Field\_3'], index=['a', 'b', 'c', 'd', 'e', 'f', 'g'])

df = df.apply(lambda x: np.square(x) if x.name in [

'a', 'e', 'g'] else x, axis=1)

df

# OUTPUT



# PROGRAM-5

import pandas as pd import numpy as np

values\_list = [[1.5, 2.5, 10.0], [2.0, 4.5, 5.0], [2.5, 5.2, 8.0],

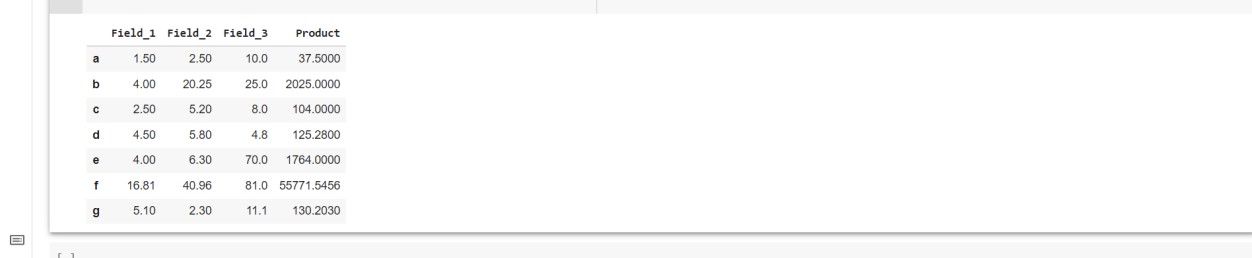
[4.5, 5.8, 4.8], [4.0, 6.3, 70], [4.1, 6.4, 9.0],

[5.1, 2.3, 11.1]]

df = pd.DataFrame(values\_list, columns=['Field\_1', 'Field\_2', 'Field\_3'], index=['a', 'b', 'c', 'd', 'e', 'f', 'g'])

df = df.apply(lambda x: np.square(x) if x.name in ['b', 'f'] else x, axis=1) df = df.assign(Product=lambda x: (x['Field\_1'] \* x['Field\_2'] \* x['Field\_3'])) df

# OUTPUT



**INDEX AND SELECTDATA**

# PROGRAM-1

import pandas as pd import numpy as np

df = pd.DataFrame(np.random.randn(8, 4),

index = ['a','b','c','d','e','f','g','h'], columns = ['A', 'B', 'C', 'D']) print (df.loc[:,'A'])

# OUTPUT



# PROGRAM-2

import pandas as pd import numpy as np

df = pd.DataFrame(np.random.randn(8, 4),

index = ['a','b','c','d','e','f','g','h'], columns = ['A', 'B', 'C', 'D']) print (df.loc[:,['A','C']])

# OUTPUT



# PROGRAM-3

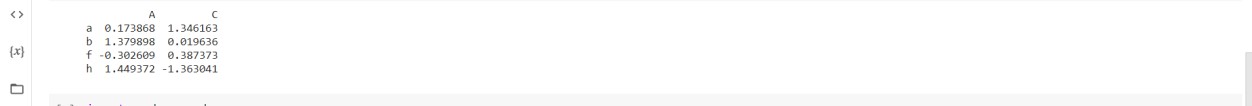
import pandas as pd import numpy as np

df = pd.DataFrame(np.random.randn(8, 4),

index = ['a','b','c','d','e','f','g','h'], columns = ['A', 'B', 'C', 'D'])

print (df.loc[['a','b','f','h'],['A','C']])

# OUTPUT



# PROGRAM-4

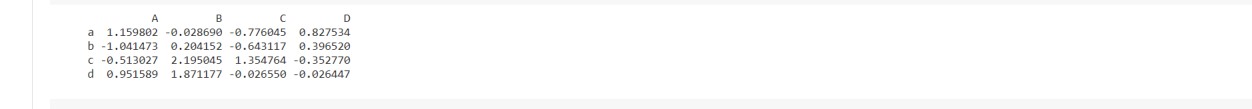
import pandas as pd import numpy as np

df = pd.DataFrame(np.random.randn(8, 4),

index = ['a','b','c','d','e','f','g','h'], columns = ['A', 'B', 'C', 'D'])

print (df.loc['a':'d'])

# OUTPUT



# PROGRAM-5

import pandas as pd import numpy as np

df = pd.DataFrame(np.random.randn(8, 4),

index = ['a','b','c','d','e','f','g','h'], columns = ['A', 'B', 'C', 'D']) print(df)

print (df.loc['a']>0)

# OUTPUT

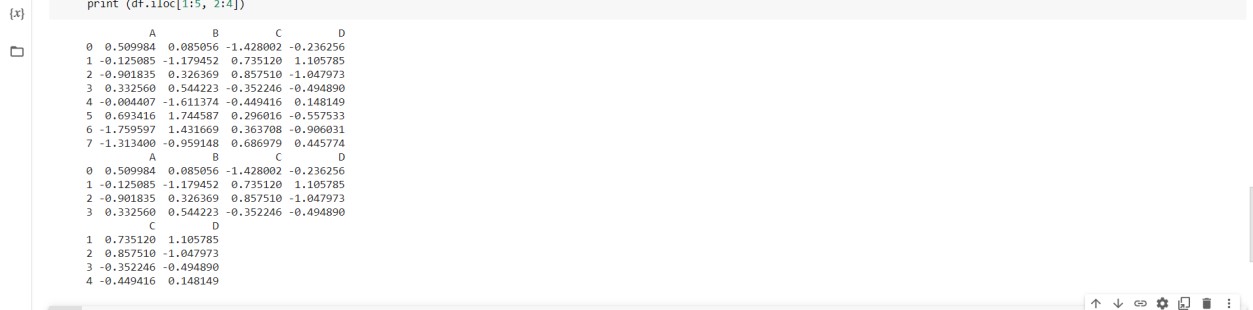


# PROGRAM-6

import pandas as pd import numpy as np

df = pd.DataFrame(np.random.randn(8, 4), columns = ['A', 'B', 'C', 'D']) print(df) print (df.iloc[:4])

# OUTPUT

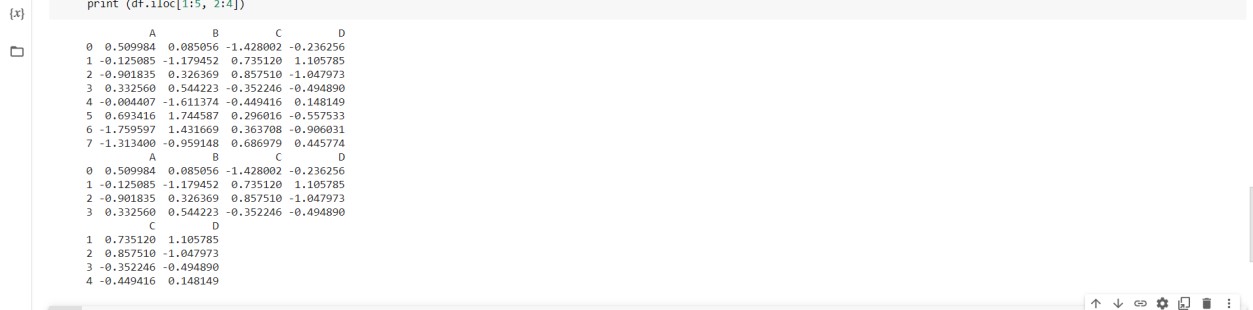


# PROGRAM-7

import pandas as pd import numpy as np

df = pd.DataFrame(np.random.randn(8, 4), columns = ['A', 'B', 'C', 'D']) print(df) print (df.iloc[:4]) print (df.iloc[1:5, 2:4])

# OUTPUT



# PROGRAM-8

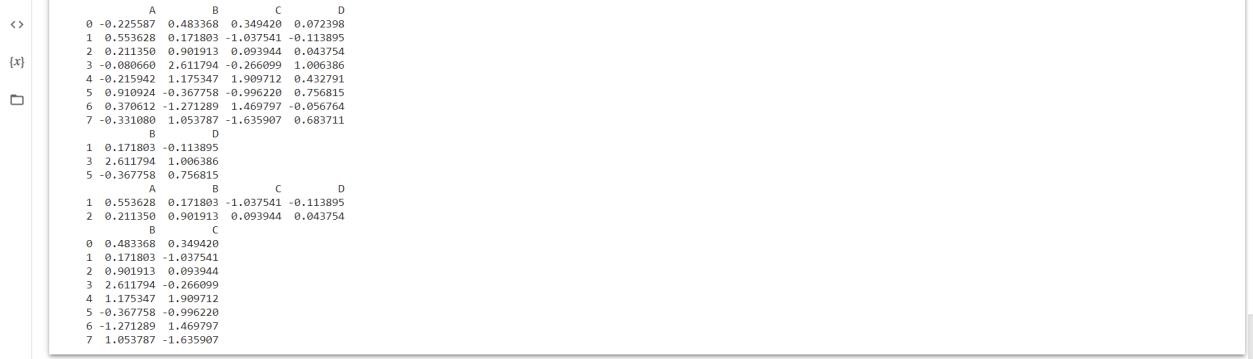
import pandas as pd import numpy as np

df = pd.DataFrame(np.random.randn(8, 4), columns = ['A', 'B', 'C', 'D']) print(df)

print (df.iloc[[1, 3, 5], [1, 3]]) print (df.iloc[1:3, :])

print (df.iloc[:,1:3])

# OUTPUT



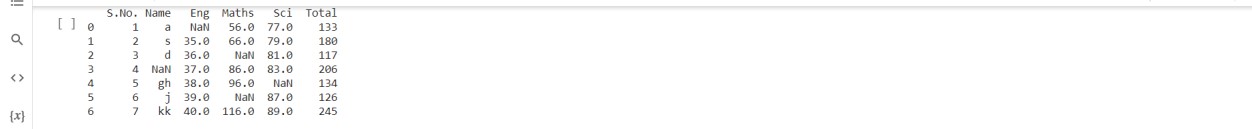
**MISSING DATA**

# PROGRAM-1

import pandas as pd

d=pd.read\_excel("C:\\Users\Shabnam\Desktop\missing.xlsx") df=pd.DataFrame(d) print(df)

# OUTPUT



**PROGRAM- 2**

df.dropna()

# OUTPUT



# PROGRAM-3

import pandas as pd

d=pd.read\_excel("C:\\Users\Shabnam\Desktop\missing.xlsx") df=pd.DataFrame(d) print(df)

# OUTPUT



**PROGRAM-4**

df['Name'].dropna()

# OUTPUT



# PROGRAM-5

import pandas as pd

d=pd.read\_excel("C:\\Users\Shabnam\Desktop\missing.xlsx") df=pd.DataFrame(d) print(df)

# OUTPUT



**PROGRAM-6**

df.loc[:,['Name','Eng']].dropna()

# OUTPUT

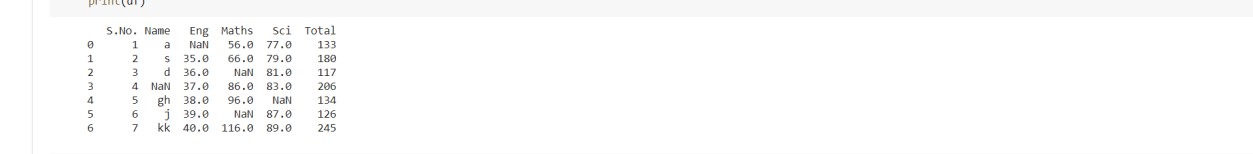


# PROGRAM-7

import pandas as pd

d=pd.read\_excel("C:\\Users\Shabnam\Desktop\missing.xlsx") df=pd.DataFrame(d) print(df)

# OUTPUT



**PROGRAM-8**

df.fillna("\*")

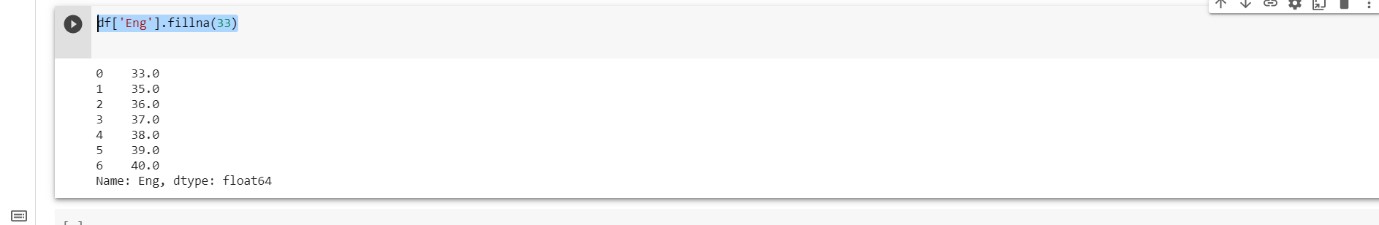
# OUTPUT



**PROGRAM-9**

df['Eng'].fillna(33)

# OUTPUT



**IAT LAB-2**

US Baby Names 1880–2010: The United States Social Security Administration (SSA) has made available data on the frequency of baby names from 1880 to 2010

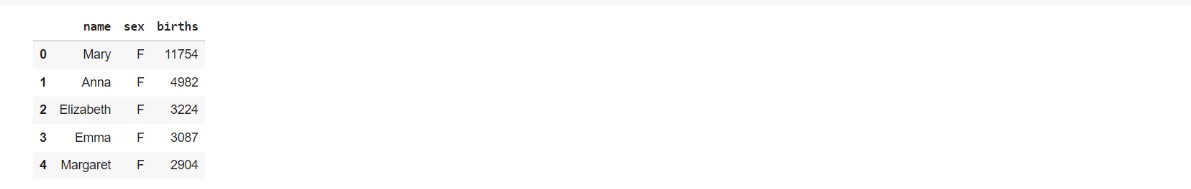
i. Use Data Wrangling to load this dataset

import pandas as pd import numpy as np

names1888 = pd.read\_csv('yob1888.txt',

names=['name', 'sex', 'births']) names1888.head()

# OUTPUT



ii. Find sum of the birth’s column by sex as the total number of births in that year years = [1888,1889,1900,1910,1928,1940,1969,1980,2000,2010] pieces = [] columns = ['name', 'sex', 'births'] for year in years:

path = 'yob%d.txt' % year

frame = pd.read\_csv(path, names=columns)

frame['year'] = year

pieces.append(frame)

result = pd.concat(pieces, ignore\_index=True) result

# OUTPUT



iii. Assemble all of the data into a single Data Frame and further add a year field

total\_births = result.pivot\_table('births', index='year', columns='sex', aggfunc='sum') total\_births

# OUTPUT



iv. Visualize total births by sex and year

line\_graph = total\_births.plot(title='Total births by sex and year') print(line\_graph)

# OUTPUT

