DAP LAB BOOK 2

ABDUR RAHMAN

18BBTCS003

- **PANDAS :**

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays.

- **DATAFRAME :**

DataFrame is a 2-dimensional labeled data structure with columns of potentially different types. You can think of it like a spreadsheet or SQL table, or a dictionary of Series objects. It is generally the most commonly used pandas object.

- **DESTRICTIVE STATISTICS :**

A large number of methods collectively compute descriptive statistics and other related operations on DataFrame. Most of these are aggregations like sum(), mean(), but some of them, like sumsum(), produce an object of the same size.

- **LAMBDA FUNCTION :**

Lambda Function, also referred to as 'Anonymous function' is same as a regular python function but can be defined without a name. While normal functions are defined using the def keyword, anonymous functions are defined using the lambda keyword.

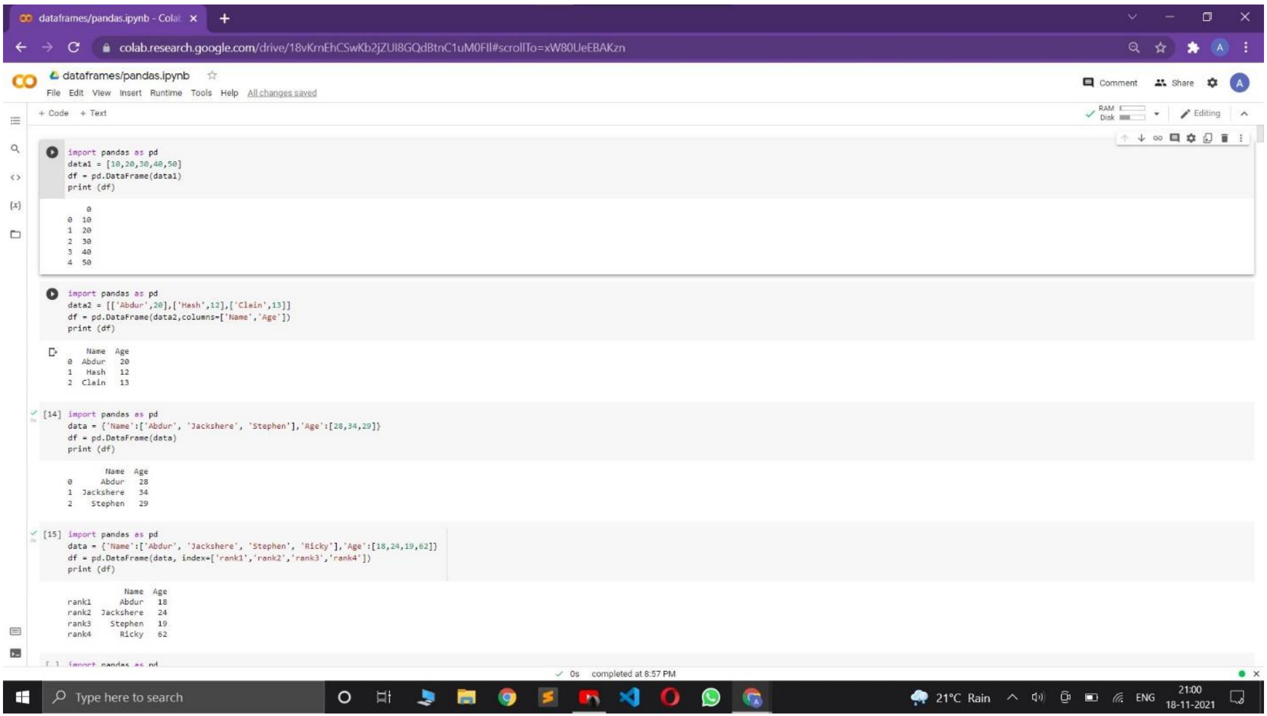
- **SORTING :**

The sort() method is a built-in Python method that, by default, sorts the list in ascending order. However, you can modify the order from ascending to descending by specifying the sorting criteria.

- **INDEXING/SELECTING DATA :**

Indexing in Python is a way to refer the individual items within an iterable by its position. In other words, you can directly access your elements of choice within an iterable and do various operations depending on your needs.

.


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FII#scrollTo=xW80UeEBAKzn
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
✓ 0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:00
18-11-2021

```
import pandas as pd
data1 = [10,20,30,40,50]
df = pd.DataFrame(data1)
print (df)
```

```
    0
0  10
1  20
2  30
3  40
4  50
```

```
import pandas as pd
data2 = [['Abdur',20],['Hash',12],['Clain',13]]
df = pd.DataFrame(data2,columns=['Name','Age'])
print (df)
```

```
    Name  Age
0  Abdur   20
1   Hash   12
2  Clain   13
```

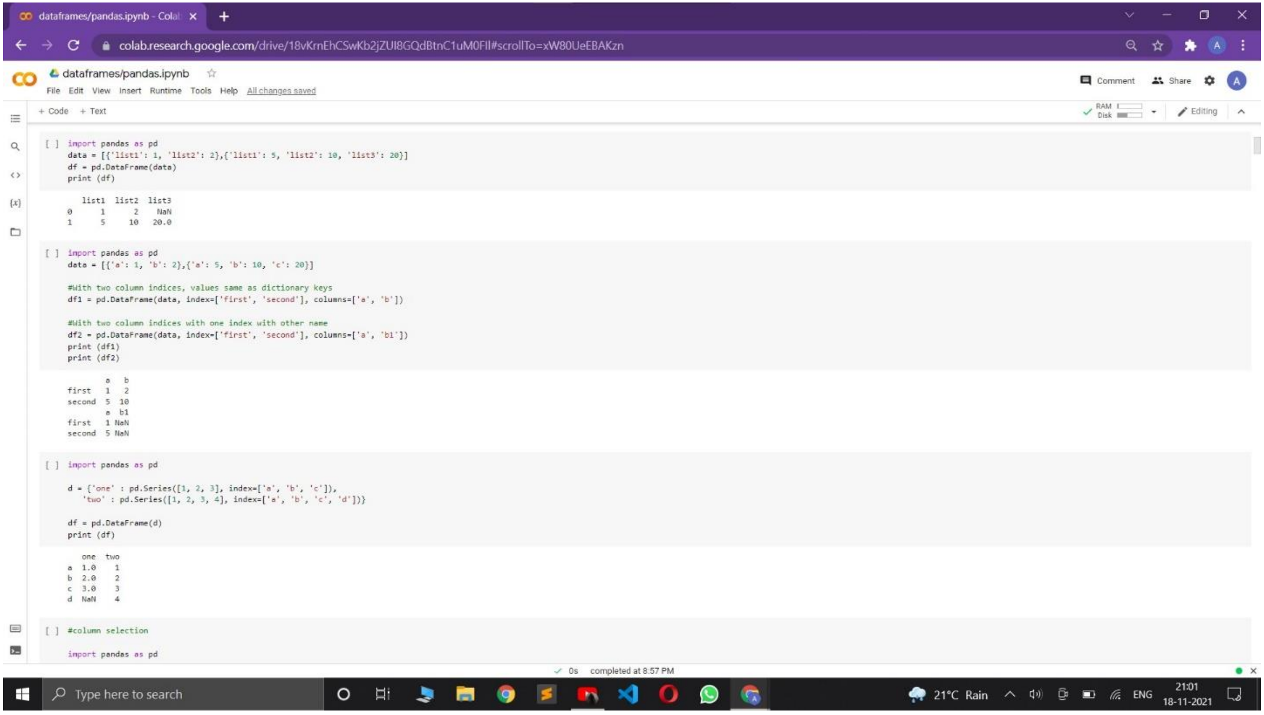
```
[14] import pandas as pd
data = {'Name':['Abdur', 'Jackshere', 'Stephen'],'Age':[28,34,29]}
df = pd.DataFrame(data)
print (df)
```

```
        Name  Age
0      Abdur   28
1  Jackshere   34
2    Stephen   29
```

```
[15] import pandas as pd
data = {'Name':['Abdur', 'Jackshere', 'Stephen', 'Ricky'],'Age':[18,24,19,62]}
df = pd.DataFrame(data, index=['rank1','rank2','rank3','rank4'])
print (df)
```

```
            Name  Age
rank1      Abdur   18
rank2  Jackshere   24
rank3    Stephen   19
rank4      Ricky   62
```

```
[ ] import pandas as pd
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FII#scrollTo=xW80UeEBAKzn
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
✓ 0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:01
18-11-2021

```
[ ] import pandas as pd
data = [{'list1': 1, 'list2': 2},{'list1': 5, 'list2': 10, 'list3': 20}]
df = pd.DataFrame(data)
print (df)
```

```
   list1  list2  list3
0      1      2    NaN
1      5     10   20.0
```

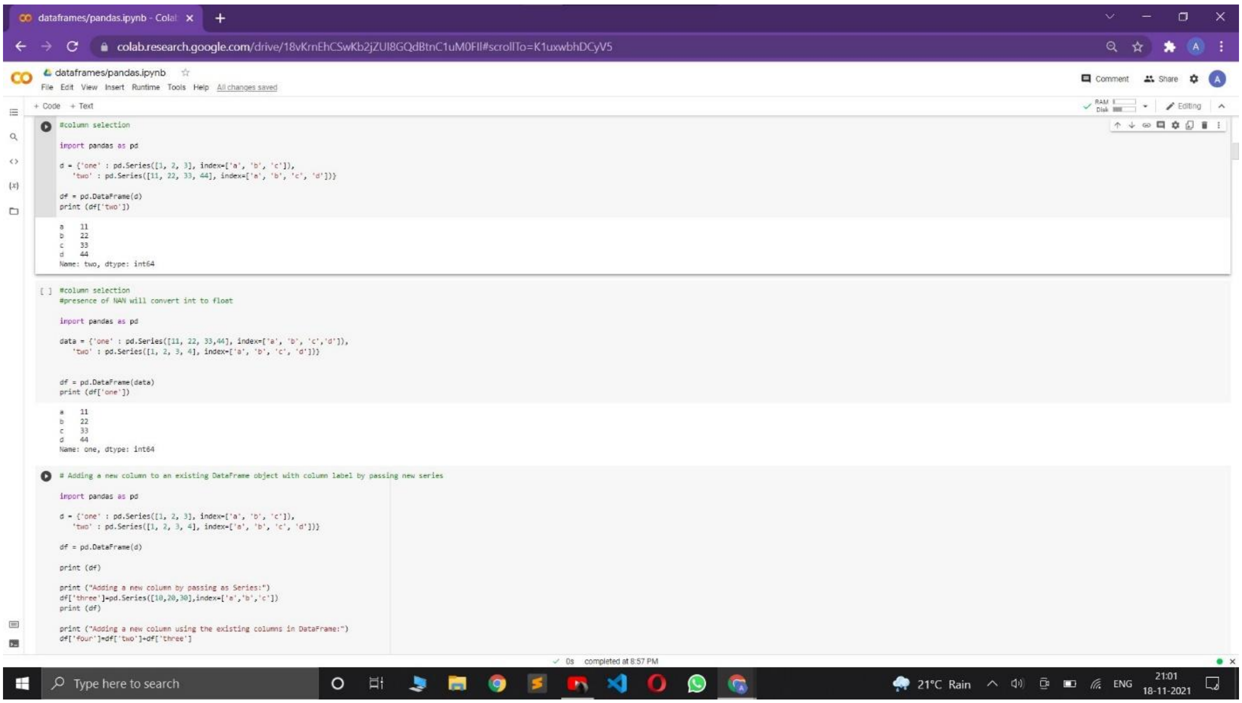
```
[ ] import pandas as pd
data = [{'a': 1, 'b': 2},{'a': 5, 'b': 10, 'c': 20}]

#With two column indices, values same as dictionary keys
df1 = pd.DataFrame(data, index=['first', 'second'], columns=['a', 'b'])

#With two column indices with one index with other name
df2 = pd.DataFrame(data, index=['first', 'second'], columns=['a', 'b1'])
print (df1)
print (df2)
```

```
        a   b
first   1   2
second  5  10
        a  b1
first   1 NaN
second  5 NaN
```

```
[ ] 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)
```

```
   one  two
a  1.0    1
b  2.0    2
c  3.0    3
d  NaN    4
```

```
[ ] #column selection

import pandas as pd
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FII#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment
Share
Editing
✓ 0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:01
18-11-2021

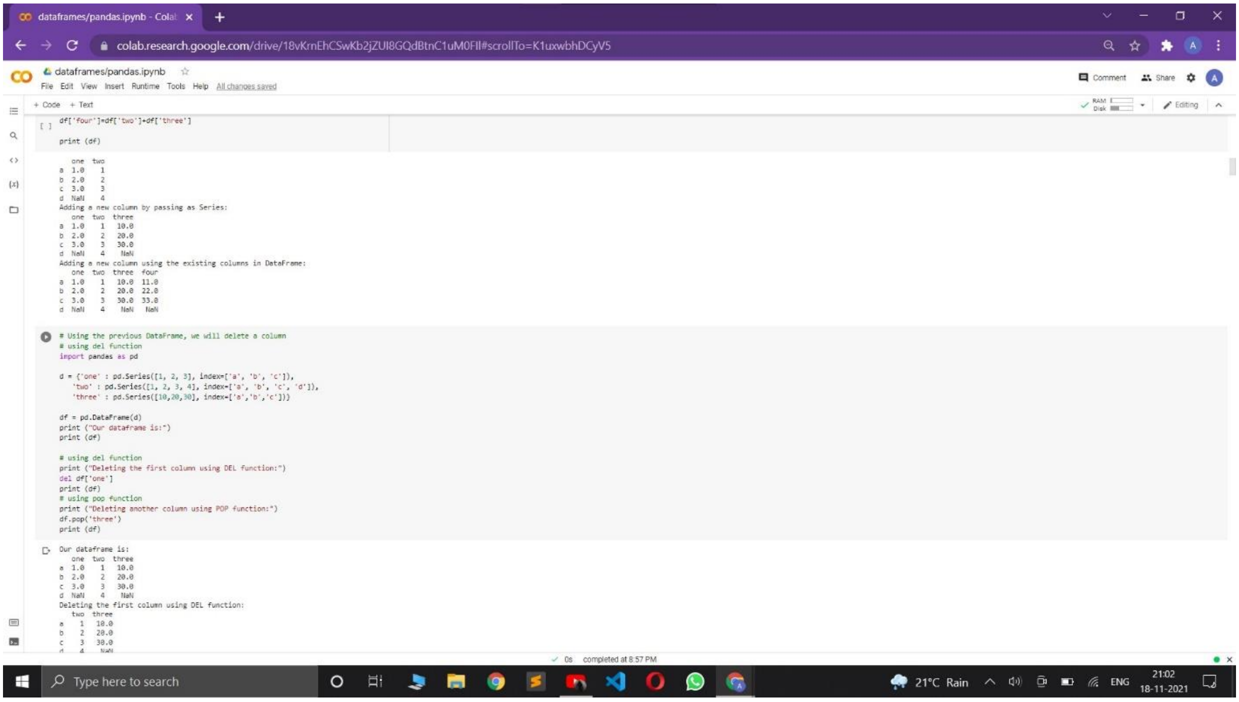
```
#column selection

import pandas as pd

d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']),
   'two' : pd.Series([11, 22, 33, 44], index=['a', 'b', 'c', 'd'])}

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

```
a    11
b    22
c    33
d    44
Name: two, dtype: int64
```

```
#column selection
#presence of NAN will convert int to float

import pandas as pd

data = {'one' : pd.Series([11, 22, 33,44], index=['a', 'b', 'c','d']),
   'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}


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

```
a    11
b    22
c    33
d    44
Name: one, dtype: int64
```

```
# Adding a new column to an existing DataFrame object with column label by passing new series

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']
```

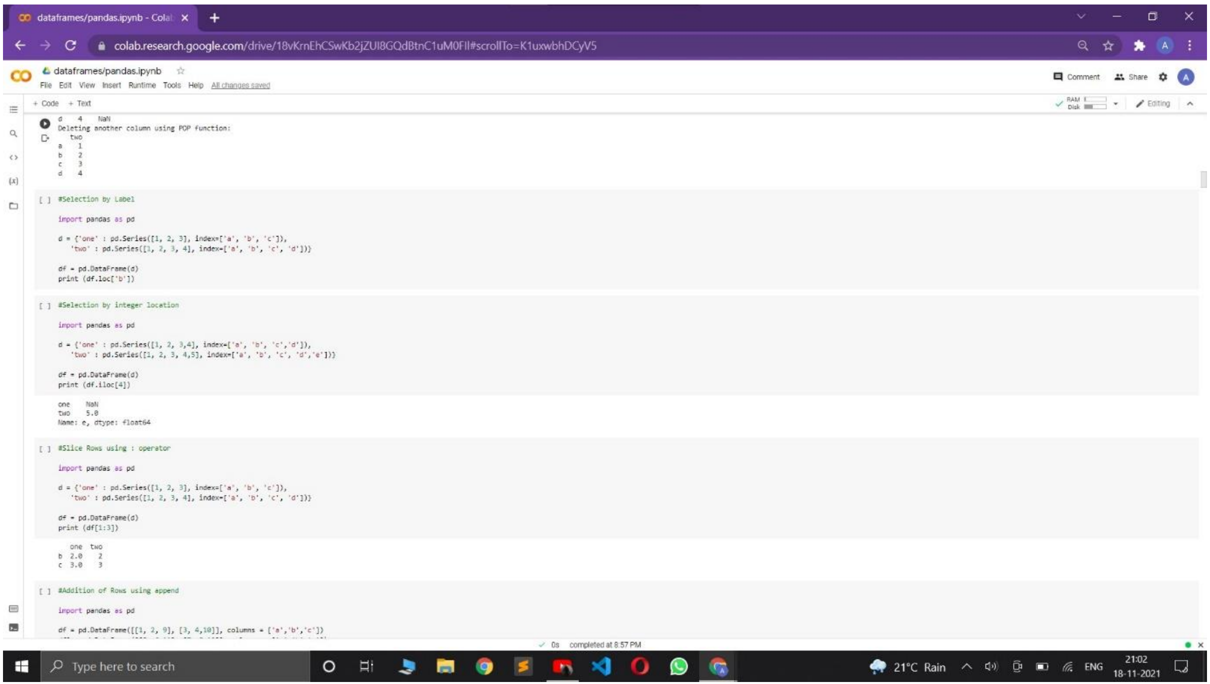

dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FII#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment
Share
Editing
✓ 0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:02
18-11-2021

```
df['four']=df['two']+df['three']

print (df)
```

```
   one  two
a  1.0    1
b  2.0    2
c  3.0    3
d  NaN    4
Adding a new column by passing as Series:
   one  two  three
a  1.0    1   10.0
b  2.0    2   20.0
c  3.0    3   30.0
d  NaN    4    NaN
Adding a new column using the existing columns in DataFrame:
   one  two  three  four
a  1.0    1   10.0  11.0
b  2.0    2   20.0  22.0
c  3.0    3   30.0  33.0
d  NaN    4    NaN   NaN
```

```
# Using the previous DataFrame, we will delete a column
# using del function
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)

# using del function
print ("Deleting the first column using DEL function:")
del df['one']
print (df)
# using pop function
print ("Deleting another column using POP function:")
df.pop('three')
print (df)
```

```
Our dataframe is:
   one  two  three
a  1.0    1   10.0
b  2.0    2   20.0
c  3.0    3   30.0
d  NaN    4    NaN
Deleting the first column using DEL function:
   two  three
a    1   10.0
b    2   20.0
c    3   30.0
d    4    NaN
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment Share
RAM Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:02
18-11-2021

```
d    4    NaN
Deleting another column using POP function:
   two
a    1
b    2
c    3
d    4
```

```
#Selection by Label

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'])
```

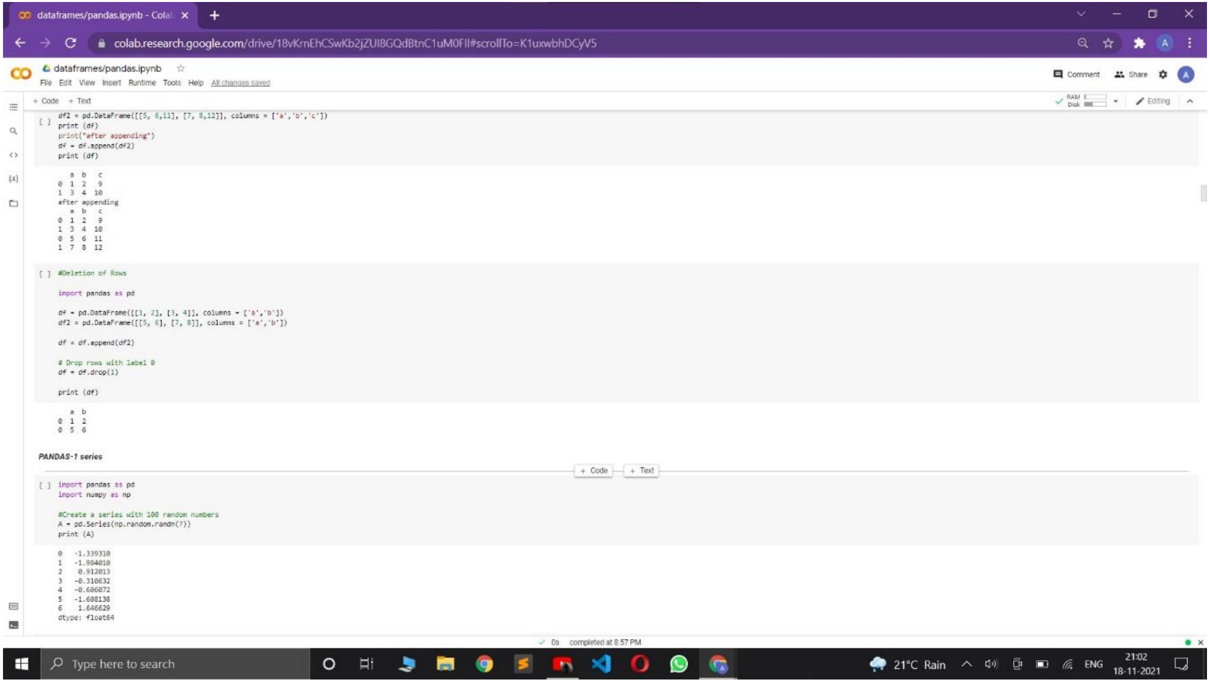
```
#Selection by integer location

import pandas as pd

d = {'one' : pd.Series([1, 2, 3,4], index=['a', 'b', 'c','d']),
   'two' : pd.Series([1, 2, 3, 4,5], index=['a', 'b', 'c', 'd','e'])}

df = pd.DataFrame(d)
print (df.iloc[4])
```

```
one    NaN
two    5.0
Name: e, dtype: float64
```

```
#Slice Rows using : operator

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[1:3])
```

```
   one  two
b  2.0    2
c  3.0    3
```

```
#Addition of Rows using append

import pandas as pd

df = pd.DataFrame([[1, 2, 9], [3, 4,10]], columns = ['a','b','c'])
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment Share
RAM Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:02
18-11-2021

```
df2 = pd.DataFrame([[5, 6,11], [7, 8,12]], columns = ['a','b','c'])
print (df)
print("after appending")
df = df.append(df2)
print (df)
```

```
   a  b   c
0  1  2   9
1  3  4  10
after appending
   a  b   c
0  1  2   9
1  3  4  10
0  5  6  11
1  7  8  12
```

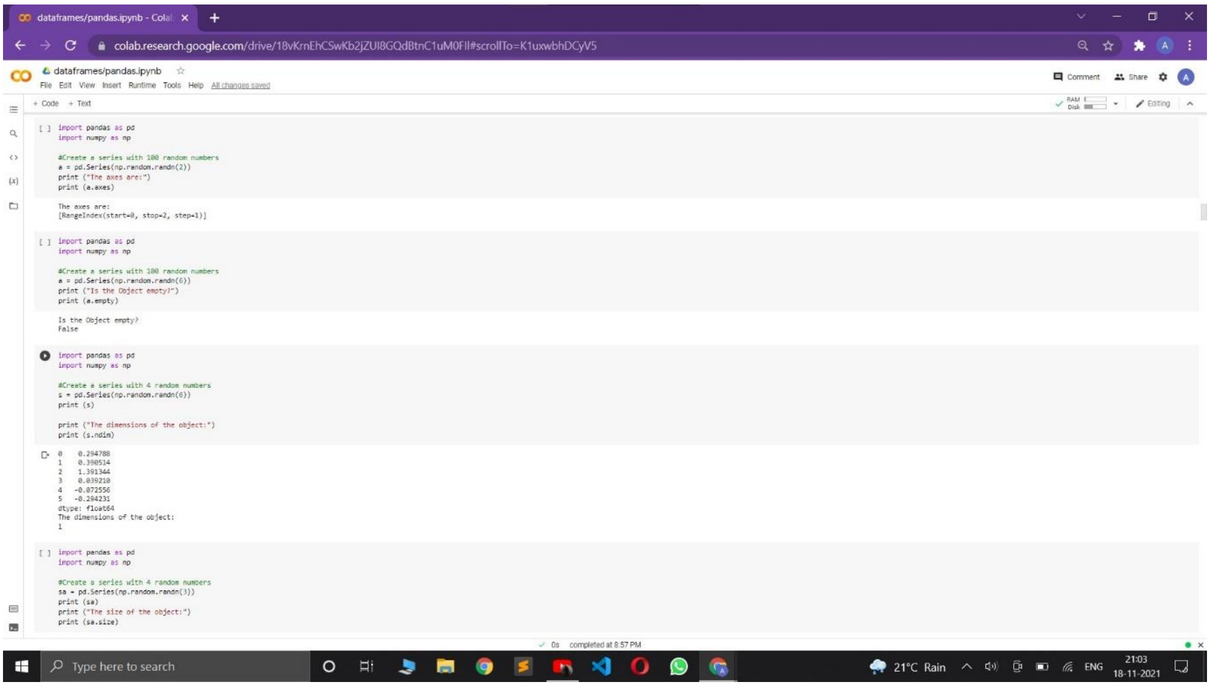
```
#Deletion of Rows

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)

# Drop rows with label 0
df = df.drop(1)

print (df)
```

```
   a  b
0  1  2
0  5  6
```

PANDAS-1 series

```
import pandas as pd
import numpy as np

#Create a series with 100 random numbers
A = pd.Series(np.random.randn(7))
print (A)
```

```
0   -1.339310
1   -1.984010
2    0.912013
3   -0.310632
4   -0.606072
5   -1.608138
6    1.646629
dtype: float64
```



dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment
Share
Editing
RAM
Disk

```
import pandas as pd
import numpy as np

#Create a series with 100 random numbers
a = pd.Series(np.random.randn(2))
print ("The axes are:")
print (a.axes)
```

```
The axes are:
[RangeIndex(start=0, stop=2, step=1)]
```

```
import pandas as pd
import numpy as np

#Create a series with 100 random numbers
a = pd.Series(np.random.randn(6))
print ("Is the Object empty?")
print (a.empty)
```

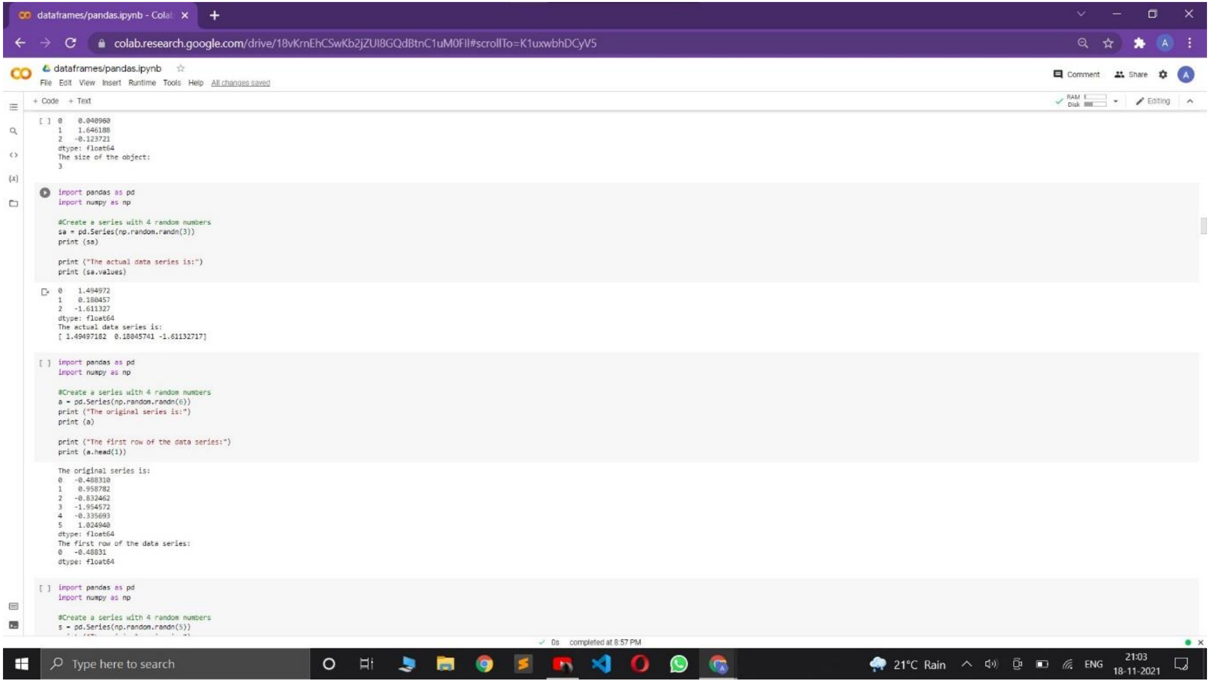
```
Is the Object empty?
False
```

```
import pandas as pd
import numpy as np

#Create a series with 4 random numbers
s = pd.Series(np.random.randn(6))
print (s)

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

```
0    0.294788
1    0.390514
2    1.391344
3    0.039210
4   -0.072556
5   -0.294231
dtype: float64
The dimensions of the object:
1
```

```
import pandas as pd
import numpy as np

#Create a series with 4 random numbers
sa = pd.Series(np.random.randn(3))
print (sa)
print ("The size of the object:")
print (sa.size)
```

✓ 0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:03
18-11-2021



dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment
Share
Editing
RAM
Disk

```
0    0.040960
1    1.646188
2   -0.123721
dtype: float64
The size of the object:
3
```

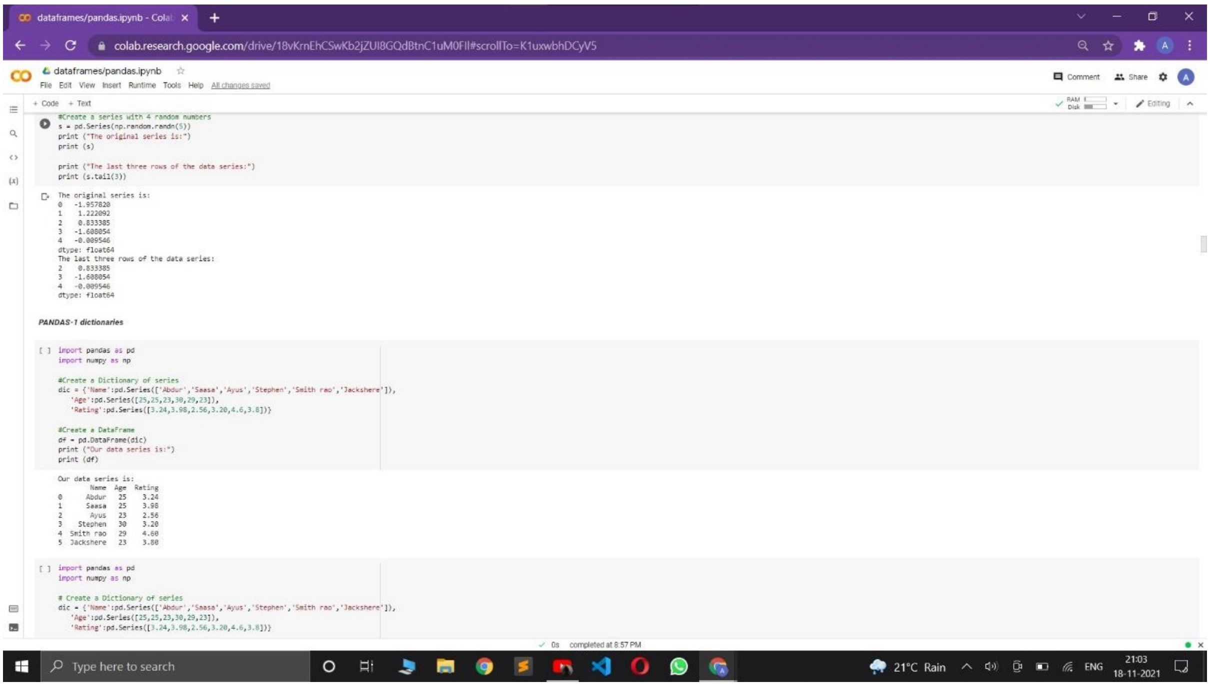
```
import pandas as pd
import numpy as np

#Create a series with 4 random numbers
sa = pd.Series(np.random.randn(3))
print (sa)

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

```
0    1.494972
1    0.180457
2   -1.611327
dtype: float64
The actual data series is:
[ 1.49497182  0.18045741 -1.61132717]
```

```
import pandas as pd
import numpy as np

#Create a series with 4 random numbers
a = pd.Series(np.random.randn(6))
print ("The original series is:")
print (a)

print ("The first row of the data series:")
print (a.head(1))
```

```
The original series is:
0   -0.488310
1    0.958782
2   -0.832462
3   -1.954572
4   -0.335693
5    1.024940
dtype: float64
The first row of the data series:
0   -0.48831
dtype: float64
```

```
import pandas as pd
import numpy as np

#Create a series with 4 random numbers
s = pd.Series(np.random.randn(5))
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
✓ 0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:03
18-11-2021

```
#Create a series with 4 random numbers
s = pd.Series(np.random.randn(5))
print ("The original series is:")
print (s)

print ("The last three rows of the data series:")
print (s.tail(3))
```

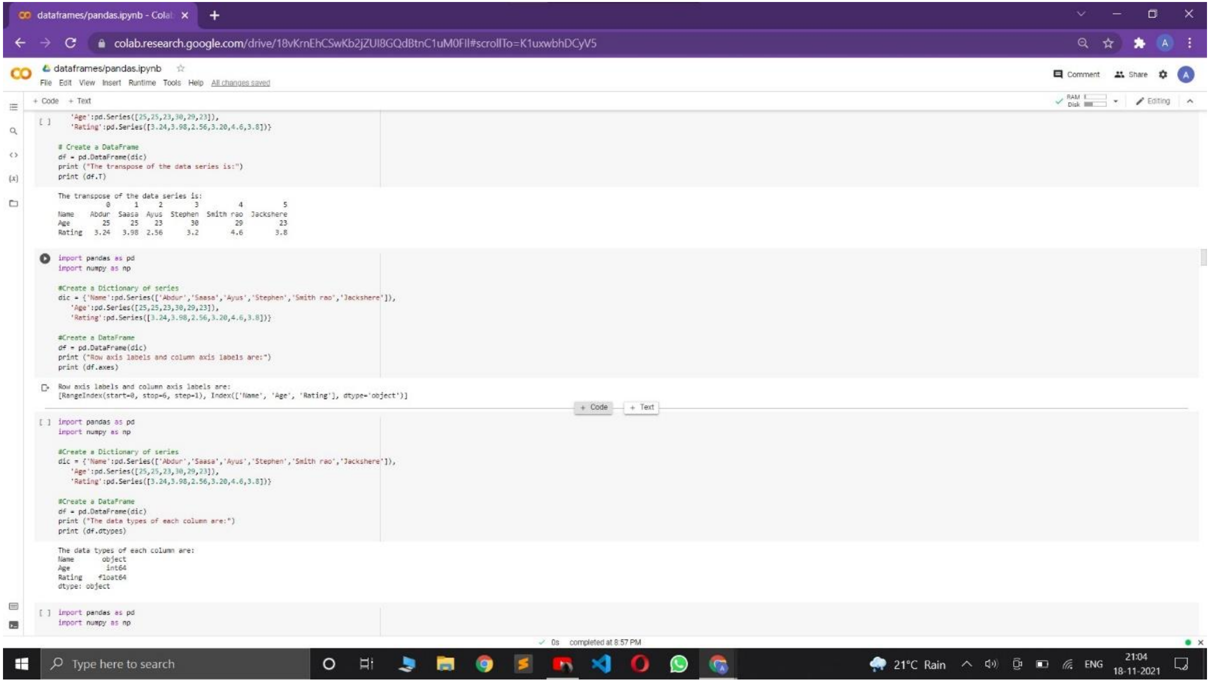
```
The original series is:
0   -1.957820
1    1.222092
2    0.833385
3   -1.608054
4   -0.009546
dtype: float64
The last three rows of the data series:
2    0.833385
3   -1.608054
4   -0.009546
dtype: float64
```

PANDAS-1 dictionaries

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
dic = {'Name':pd.Series(['Abdur','Saasa','Ayus','Stephen','Smith rao','Jackshere']),
   'Age':pd.Series([25,25,23,30,29,23]),
   'Rating':pd.Series([3.24,3.98,2.56,3.20,4.6,3.8])}

#Create a DataFrame
df = pd.DataFrame(dic)
print ("Our data series is:")
print (df)
```

```
Our data series is:
        Name  Age  Rating
0      Abdur   25    3.24
1      Saasa   25    3.98
2       Ayus   23    2.56
3    Stephen   30    3.20
4  Smith rao   29    4.60
5  Jackshere   23    3.80
```

```
import pandas as pd
import numpy as np

# Create a Dictionary of series
dic = {'Name':pd.Series(['Abdur','Saasa','Ayus','Stephen','Smith rao','Jackshere']),
   'Age':pd.Series([25,25,23,30,29,23]),
   'Rating':pd.Series([3.24,3.98,2.56,3.20,4.6,3.8])}
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
+ Code + Text
✓ 0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:04
18-11-2021

```
   'Age':pd.Series([25,25,23,30,29,23]),
   'Rating':pd.Series([3.24,3.98,2.56,3.20,4.6,3.8])}

# Create a DataFrame
df = pd.DataFrame(dic)
print ("The transpose of the data series is:")
print (df.T)
```

```
The transpose of the data series is:
            0      1     2        3          4          5
Name    Abdur  Saasa  Ayus  Stephen  Smith rao  Jackshere
Age        25     25    23       30         29         23
Rating   3.24   3.98  2.56      3.2        4.6        3.8
```

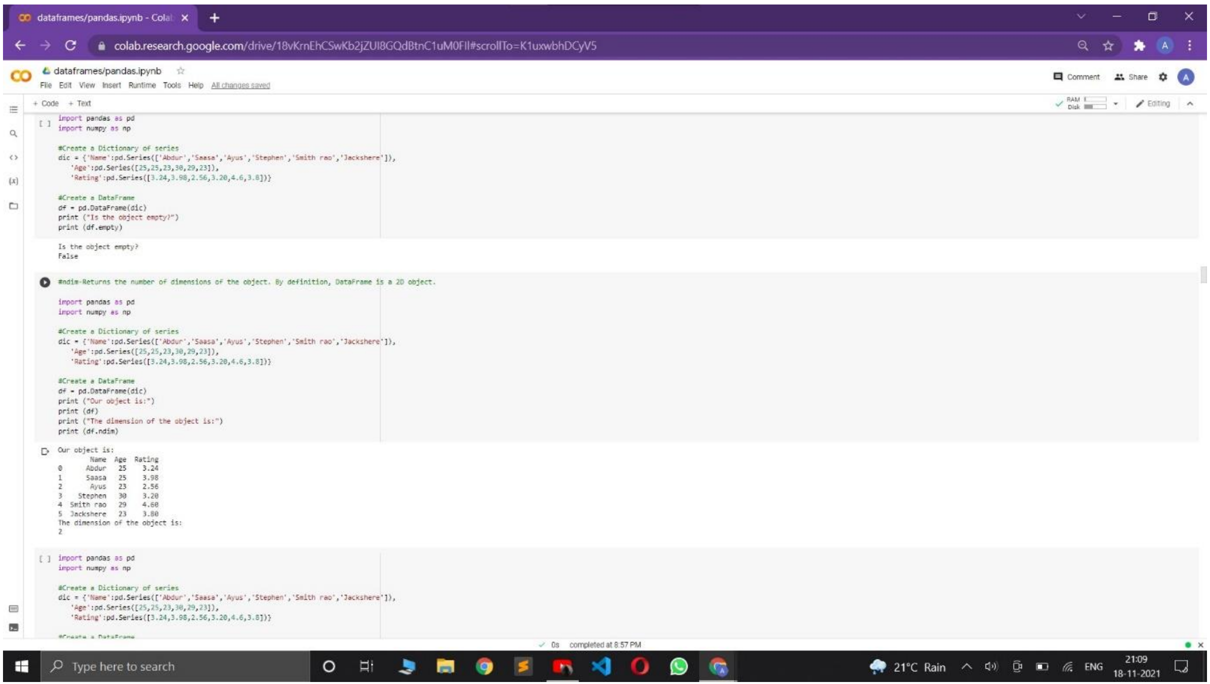
```
import pandas as pd
import numpy as np

#Create a Dictionary of series
dic = {'Name':pd.Series(['Abdur','Saasa','Ayus','Stephen','Smith rao','Jackshere']),
   'Age':pd.Series([25,25,23,30,29,23]),
   'Rating':pd.Series([3.24,3.98,2.56,3.20,4.6,3.8])}

#Create a DataFrame
df = pd.DataFrame(dic)
print ("Row axis labels and column axis labels are:")
print (df.axes)
```

```
Row axis labels and column axis labels are:
[RangeIndex(start=0, stop=6, step=1), Index(['Name', 'Age', 'Rating'], dtype='object')]
```

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
dic = {'Name':pd.Series(['Abdur','Saasa','Ayus','Stephen','Smith rao','Jackshere']),
   'Age':pd.Series([25,25,23,30,29,23]),
   'Rating':pd.Series([3.24,3.98,2.56,3.20,4.6,3.8])}

#Create a DataFrame
df = pd.DataFrame(dic)
print ("The data types of each column are:")
print (df.dtypes)
```

```
The data types of each column are:
Name       object
Age         int64
Rating    float64
dtype: object
```

```
import pandas as pd
import numpy as np
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:09
18-11-2021

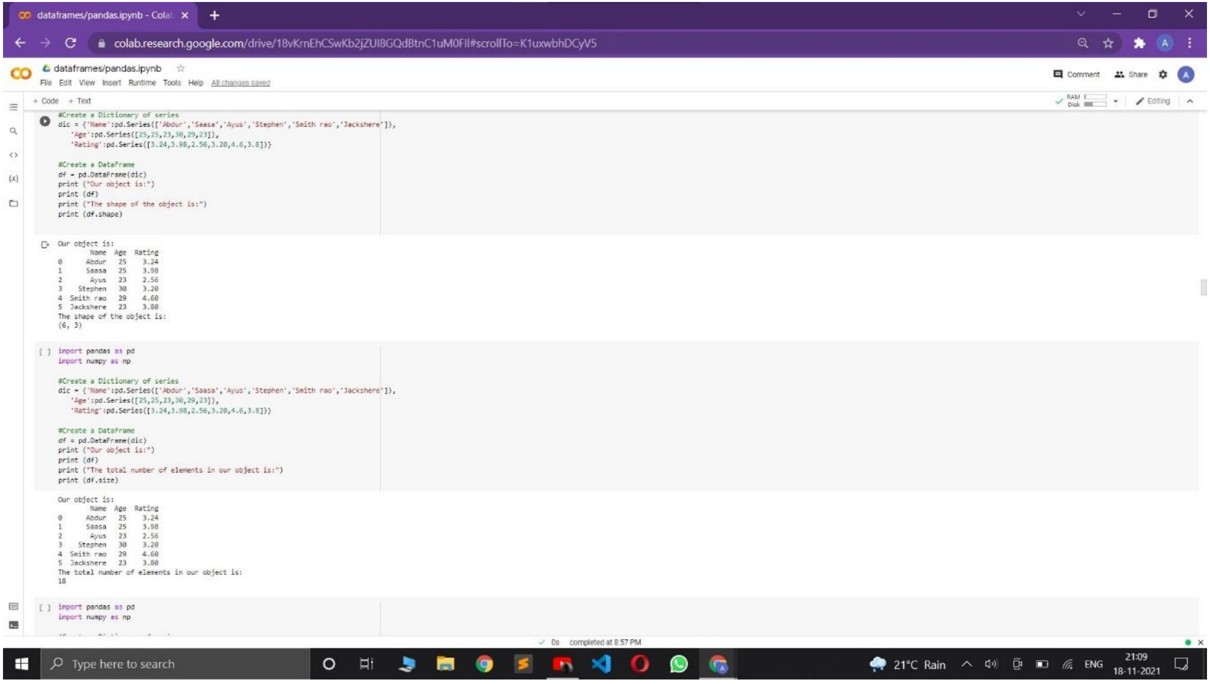
```
import pandas as pd
import numpy as np

#Create a Dictionary of series
dic = {'Name':pd.Series(['Abdur','Saasa','Ayus','Stephen','Smith rao','Jackshere']),
   'Age':pd.Series([25,25,23,30,29,23]),
   'Rating':pd.Series([3.24,3.98,2.56,3.20,4.6,3.8])}

#Create a DataFrame
df = pd.DataFrame(dic)
print ("Is the object empty?")
print (df.empty)
```

```
Is the object empty?
False
```

```
#ndim-Returns the number of dimensions of the object. By definition, DataFrame is a 2D object.

import pandas as pd
import numpy as np

#Create a Dictionary of series
dic = {'Name':pd.Series(['Abdur','Saasa','Ayus','Stephen','Smith rao','Jackshere']),
   'Age':pd.Series([25,25,23,30,29,23]),
   'Rating':pd.Series([3.24,3.98,2.56,3.20,4.6,3.8])}

#Create a DataFrame
df = pd.DataFrame(dic)
print ("Our object is:")
print (df)
print ("The dimension of the object is:")
print (df.ndim)
```

```
Our object is:
        Name  Age  Rating
0      Abdur   25    3.24
1      Saasa   25    3.98
2       Ayus   23    2.56
3    Stephen   30    3.20
4  Smith rao   29    4.60
5  Jackshere   23    3.80
The dimension of the object is:
2
```

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
dic = {'Name':pd.Series(['Abdur','Saasa','Ayus','Stephen','Smith rao','Jackshere']),
   'Age':pd.Series([25,25,23,30,29,23]),
   'Rating':pd.Series([3.24,3.98,2.56,3.20,4.6,3.8])}

#Create a DataFrame
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:09
18-11-2021

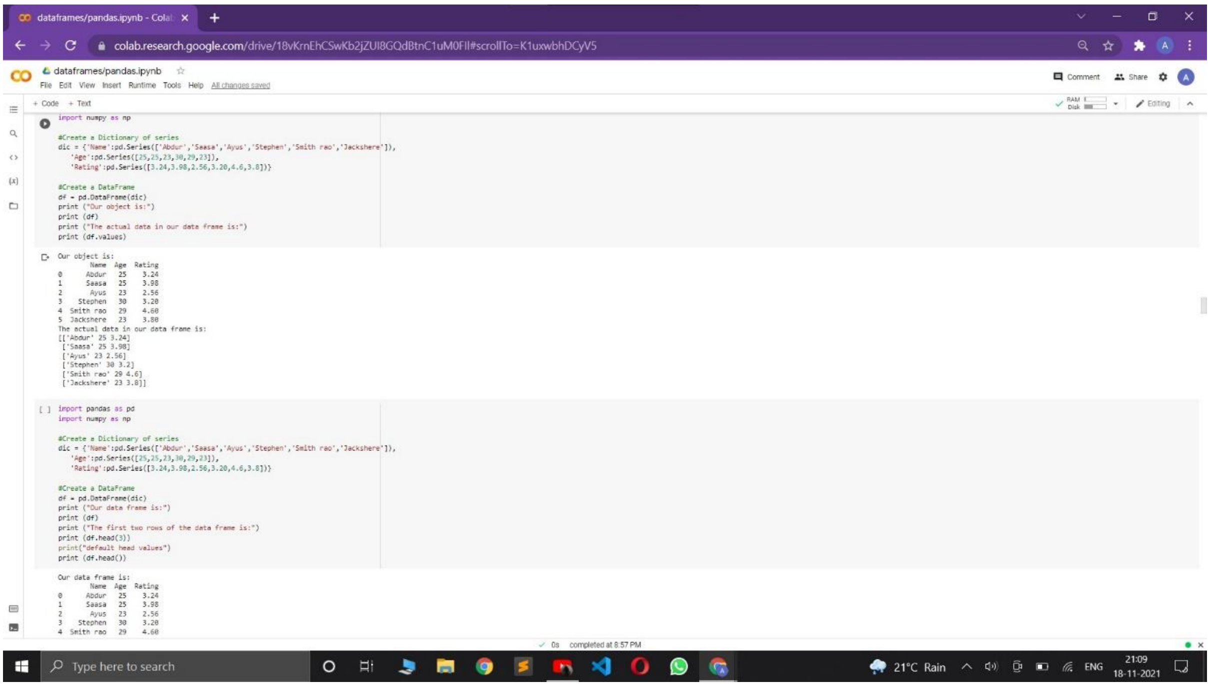
```
#Create a Dictionary of series
dic = {'Name':pd.Series(['Abdur','Saasa','Ayus','Stephen','Smith rao','Jackshere']),
   'Age':pd.Series([25,25,23,30,29,23]),
   'Rating':pd.Series([3.24,3.98,2.56,3.20,4.6,3.8])}

#Create a DataFrame
df = pd.DataFrame(dic)
print ("Our object is:")
print (df)
print ("The shape of the object is:")
print (df.shape)
```

```
Our object is:
        Name  Age  Rating
0      Abdur   25    3.24
1      Saasa   25    3.98
2       Ayus   23    2.56
3    Stephen   30    3.20
4  Smith rao   29    4.60
5  Jackshere   23    3.80
The shape of the object is:
(6, 3)
```

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
dic = {'Name':pd.Series(['Abdur','Saasa','Ayus','Stephen','Smith rao','Jackshere']),
   'Age':pd.Series([25,25,23,30,29,23]),
   'Rating':pd.Series([3.24,3.98,2.56,3.20,4.6,3.8])}

#Create a DataFrame
df = pd.DataFrame(dic)
print ("Our object is:")
print (df)
print ("The total number of elements in our object is:")
print (df.size)
```

```
Our object is:
        Name  Age  Rating
0      Abdur   25    3.24
1      Saasa   25    3.98
2       Ayus   23    2.56
3    Stephen   30    3.20
4  Smith rao   29    4.60
5  Jackshere   23    3.80
The total number of elements in our object is:
18
```

```
import pandas as pd
import numpy as np
```

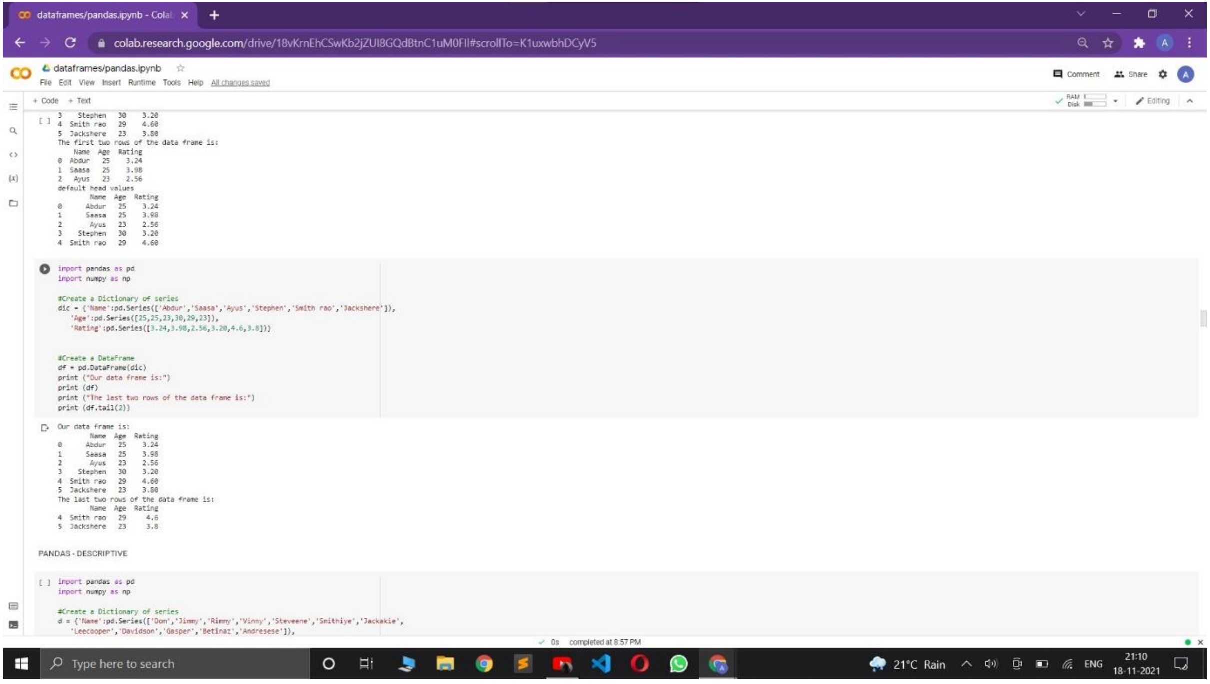

dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:09
18-11-2021

```
import numpy as np

#Create a Dictionary of series
dic = {'Name':pd.Series(['Abdur','Saasa','Ayus','Stephen','Smith rao','Jackshere']),
   'Age':pd.Series([25,25,23,30,29,23]),
   'Rating':pd.Series([3.24,3.98,2.56,3.20,4.6,3.8])}

#Create a DataFrame
df = pd.DataFrame(dic)
print ("Our object is:")
print (df)
print ("The actual data in our data frame is:")
print (df.values)
```

```
Our object is:
        Name  Age  Rating
0      Abdur   25    3.24
1      Saasa   25    3.98
2       Ayus   23    2.56
3    Stephen   30    3.20
4  Smith rao   29    4.60
5  Jackshere   23    3.80
The actual data in our data frame is:
[['Abdur' 25 3.24]
 ['Saasa' 25 3.98]
 ['Ayus' 23 2.56]
 ['Stephen' 30 3.2]
 ['Smith rao' 29 4.6]
 ['Jackshere' 23 3.8]]
```

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
dic = {'Name':pd.Series(['Abdur','Saasa','Ayus','Stephen','Smith rao','Jackshere']),
   'Age':pd.Series([25,25,23,30,29,23]),
   'Rating':pd.Series([3.24,3.98,2.56,3.20,4.6,3.8])}

#Create a DataFrame
df = pd.DataFrame(dic)
print ("Our data frame is:")
print (df)
print ("The first two rows of the data frame is:")
print (df.head(3))
print("default head values")
print (df.head())
```

```
Our data frame is:
        Name  Age  Rating
0      Abdur   25    3.24
1      Saasa   25    3.98
2       Ayus   23    2.56
3    Stephen   30    3.20
4  Smith rao   29    4.60
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:10
18-11-2021

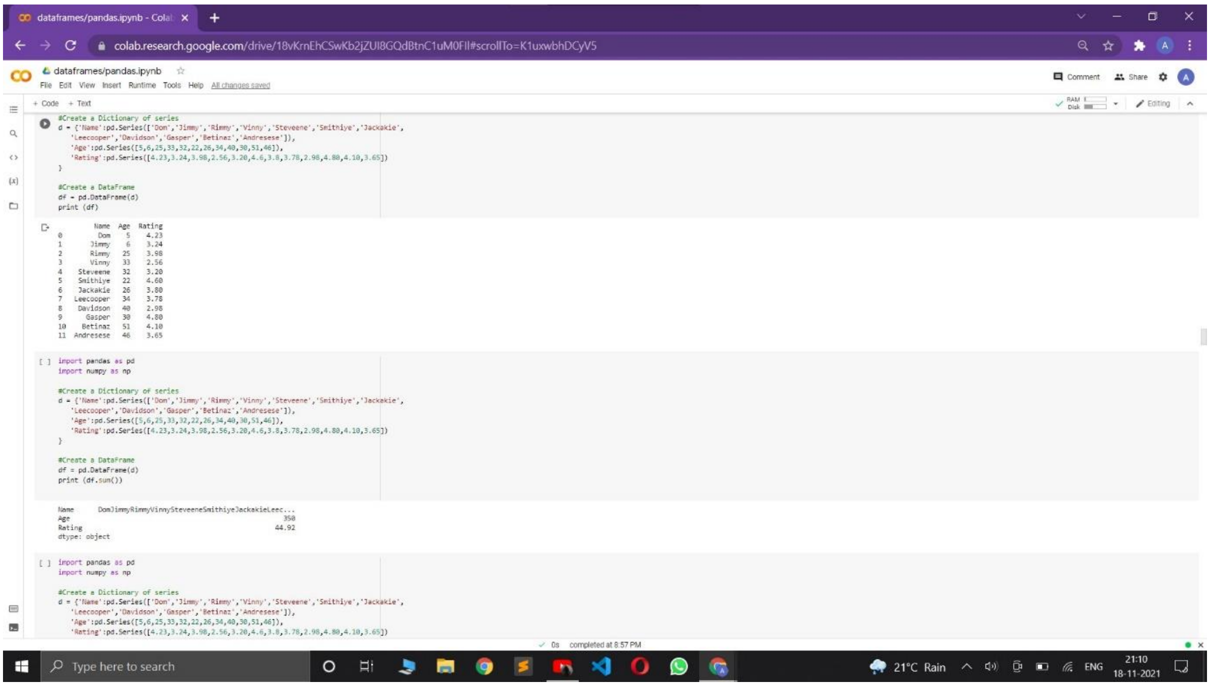
```
3    Stephen   30    3.20
4  Smith rao   29    4.60
5  Jackshere   23    3.80
The first two rows of the data frame is:
    Name  Age  Rating
0  Abdur   25    3.24
1  Saasa   25    3.98
2   Ayus   23    2.56
default head values
        Name  Age  Rating
0      Abdur   25    3.24
1      Saasa   25    3.98
2       Ayus   23    2.56
3    Stephen   30    3.20
4  Smith rao   29    4.60
```

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
dic = {'Name':pd.Series(['Abdur','Saasa','Ayus','Stephen','Smith rao','Jackshere']),
   'Age':pd.Series([25,25,23,30,29,23]),
   'Rating':pd.Series([3.24,3.98,2.56,3.20,4.6,3.8])}


#Create a DataFrame
df = pd.DataFrame(dic)
print ("Our data frame is:")
print (df)
print ("The last two rows of the data frame is:")
print (df.tail(2))
```

```
Our data frame is:
        Name  Age  Rating
0      Abdur   25    3.24
1      Saasa   25    3.98
2       Ayus   23    2.56
3    Stephen   30    3.20
4  Smith rao   29    4.60
5  Jackshere   23    3.80
The last two rows of the data frame is:
        Name  Age  Rating
4  Smith rao   29     4.6
5  Jackshere   23     3.8
```

PANDAS - DESCRIPTIVE

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Dom','Jimmy','Rimmy','Vinny','Steveene','Smithiye','Jackakie',
   'Leecooper','Davidson','Gasper','Betinaz','Andresese']),
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment
Share
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:10
18-11-2021

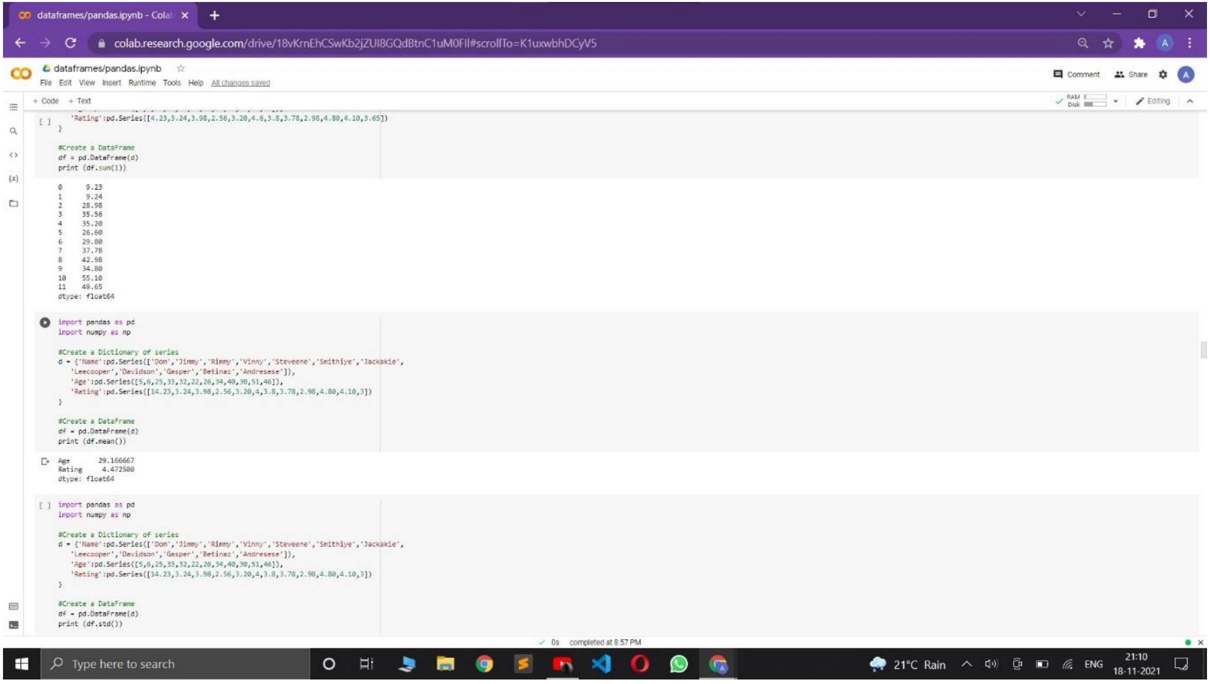
```
#Create a Dictionary of series
d = {'Name':pd.Series(['Dom','Jimmy','Rimmy','Vinny','Steveene','Smithiye','Jackakie',
   'Leecooper','Davidson','Gasper','Betinaz','Andresese']),
   'Age':pd.Series([5,6,25,33,32,22,26,34,40,30,51,46]),
   'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65])
}

#Create a DataFrame
df = pd.DataFrame(d)
print (df)
```

```
         Name  Age  Rating
0         Dom    5    4.23
1       Jimmy    6    3.24
2       Rimmy   25    3.98
3       Vinny   33    2.56
4    Steveene   32    3.20
5    Smithiye   22    4.60
6    Jackakie   26    3.80
7   Leecooper   34    3.78
8    Davidson   40    2.98
9      Gasper   30    4.80
10    Betinaz   51    4.10
11  Andresese   46    3.65
```

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Dom','Jimmy','Rimmy','Vinny','Steveene','Smithiye','Jackakie',
   'Leecooper','Davidson','Gasper','Betinaz','Andresese']),
   'Age':pd.Series([5,6,25,33,32,22,26,34,40,30,51,46]),
   '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])
}

#Create a DataFrame
df = pd.DataFrame(d)
print (df.sum())
```

```
Name      DomJimmyRimmyVinnySteveeneSmithiyeJackakieLeec...
Age                                                     350
Rating                                                44.92
dtype: object
```

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Dom','Jimmy','Rimmy','Vinny','Steveene','Smithiye','Jackakie',
   'Leecooper','Davidson','Gasper','Betinaz','Andresese']),
   'Age':pd.Series([5,6,25,33,32,22,26,34,40,30,51,46]),
   '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])
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment
Share
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:10
18-11-2021

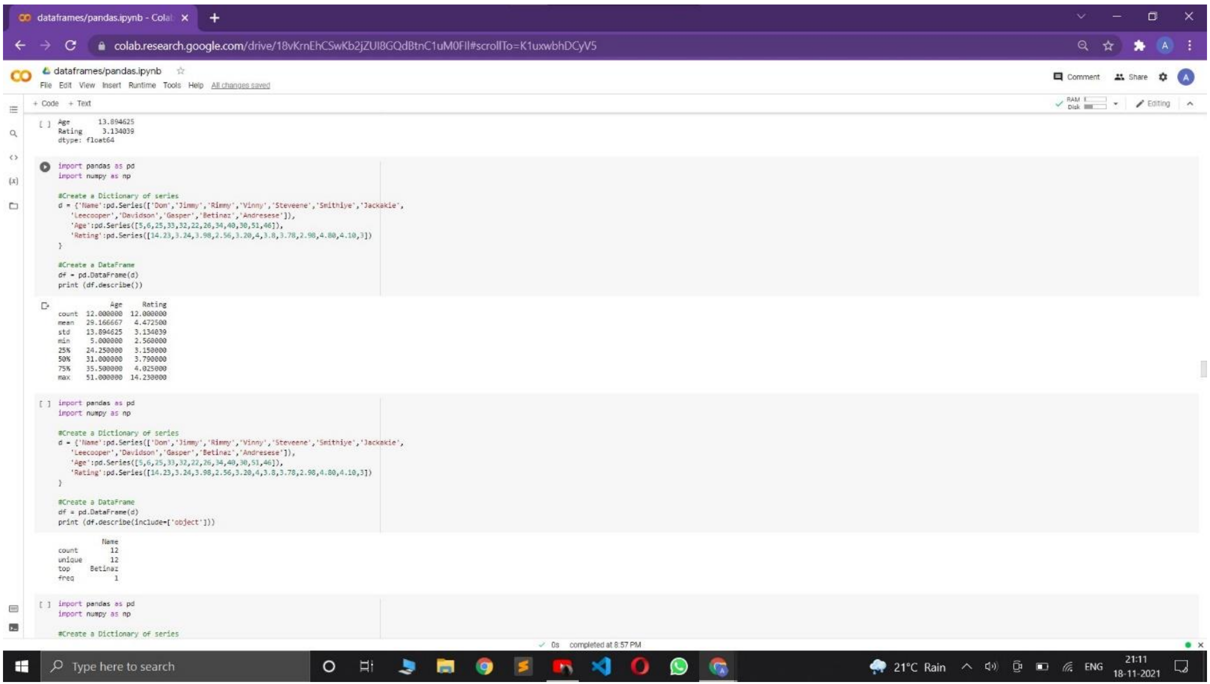
```
   'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65])
}

#Create a DataFrame
df = pd.DataFrame(d)
print (df.sum(1))
```

```
0      9.23
1      9.24
2     28.98
3     35.56
4     35.20
5     26.60
6     29.80
7     37.78
8     42.98
9     34.80
10    55.10
11    49.65
dtype: float64
```

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Dom','Jimmy','Rimmy','Vinny','Steveene','Smithiye','Jackakie',
   'Leecooper','Davidson','Gasper','Betinaz','Andresese']),
   'Age':pd.Series([5,6,25,33,32,22,26,34,40,30,51,46]),
   'Rating':pd.Series([14.23,3.24,3.98,2.56,3.20,4,3.8,3.78,2.98,4.80,4.10,3])
}

#Create a DataFrame
df = pd.DataFrame(d)
print (df.mean())
```

```
Age       29.166667
Rating     4.472500
dtype: float64
```

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Dom','Jimmy','Rimmy','Vinny','Steveene','Smithiye','Jackakie',
   'Leecooper','Davidson','Gasper','Betinaz','Andresese']),
   'Age':pd.Series([5,6,25,33,32,22,26,34,40,30,51,46]),
   'Rating':pd.Series([14.23,3.24,3.98,2.56,3.20,4,3.8,3.78,2.98,4.80,4.10,3])
}

#Create a DataFrame
df = pd.DataFrame(d)
print (df.std())
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment Share
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:11
18-11-2021

```
Age       13.894625
Rating     3.134039
dtype: float64
```

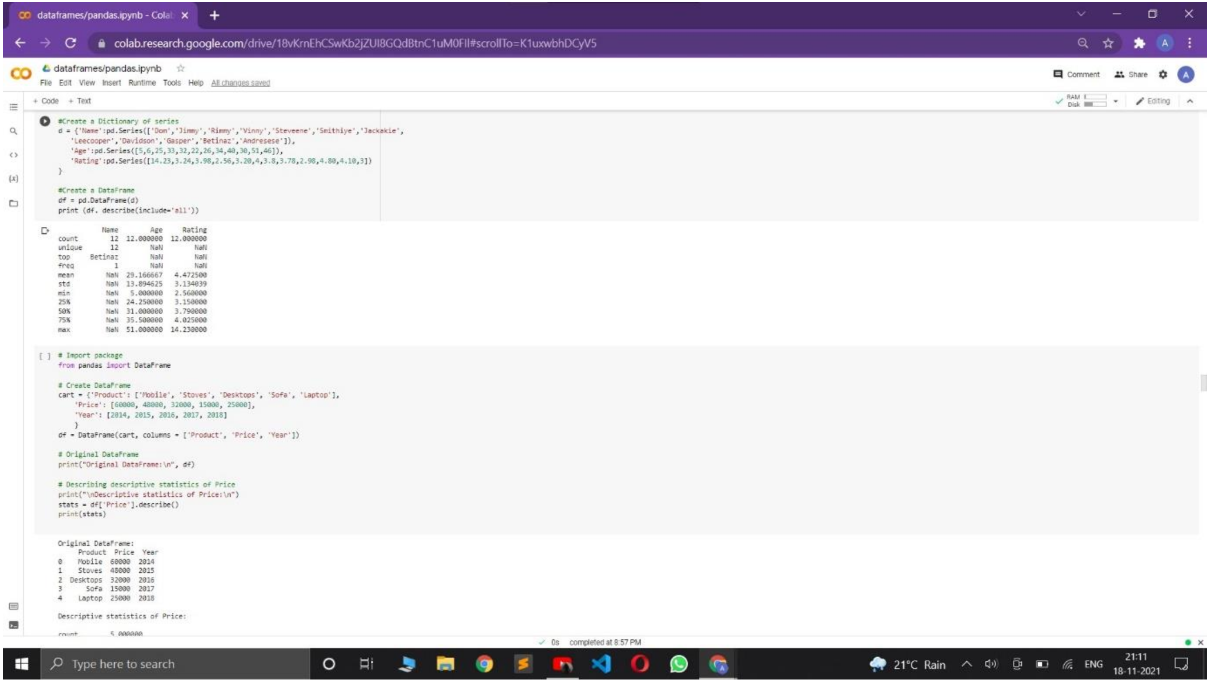
```python
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Dom','Jimmy','Rimmy','Vinny','Steveene','Smithiye','Jackakie',
   'Leecooper','Davidson','Gasper','Betinaz','Andresese']),
   'Age':pd.Series([5,6,25,33,32,22,26,34,40,30,51,46]),
   'Rating':pd.Series([14.23,3.24,3.98,2.56,3.20,4,3.8,3.78,2.98,4.80,4.10,3])
}

#Create a DataFrame
df = pd.DataFrame(d)
print (df.describe())
```

```
             Age     Rating
count  12.000000  12.000000
mean   29.166667   4.472500
std    13.894625   3.134039
min     5.000000   2.560000
25%    24.250000   3.150000
50%    31.000000   3.790000
75%    35.500000   4.025000
max    51.000000  14.230000
```

```python
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Dom','Jimmy','Rimmy','Vinny','Steveene','Smithiye','Jackakie',
   'Leecooper','Davidson','Gasper','Betinaz','Andresese']),
   'Age':pd.Series([5,6,25,33,32,22,26,34,40,30,51,46]),
   'Rating':pd.Series([14.23,3.24,3.98,2.56,3.20,4,3.8,3.78,2.98,4.80,4.10,3])
}

#Create a DataFrame
df = pd.DataFrame(d)
print (df.describe(include=['object']))
```

```
           Name
count        12
unique       12
top     Betinaz
freq          1
```

```python
import pandas as pd
import numpy as np

#Create a Dictionary of series
```

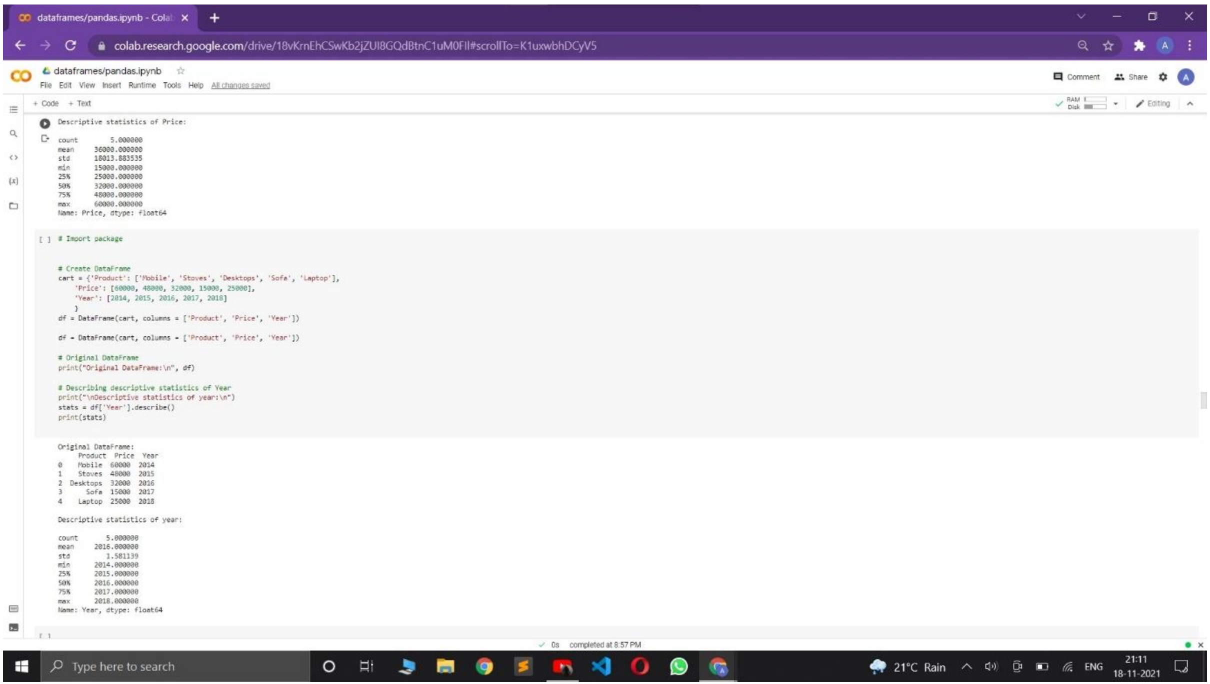

dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment Share
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:11
18-11-2021

```python
#Create a Dictionary of series
d = {'Name':pd.Series(['Dom','Jimmy','Rimmy','Vinny','Steveene','Smithiye','Jackakie',
   'Leecooper','Davidson','Gasper','Betinaz','Andresese']),
   'Age':pd.Series([5,6,25,33,32,22,26,34,40,30,51,46]),
   'Rating':pd.Series([14.23,3.24,3.98,2.56,3.20,4,3.8,3.78,2.98,4.80,4.10,3])
}

#Create a DataFrame
df = pd.DataFrame(d)
print (df. describe(include='all'))
```

```
           Name        Age     Rating
count        12  12.000000  12.000000
unique       12        NaN        NaN
top     Betinaz        NaN        NaN
freq          1        NaN        NaN
mean        NaN  29.166667   4.472500
std         NaN  13.894625   3.134039
min         NaN   5.000000   2.560000
25%         NaN  24.250000   3.150000
50%         NaN  31.000000   3.790000
75%         NaN  35.500000   4.025000
max         NaN  51.000000  14.230000
```

```python
# Import package
from pandas import DataFrame

# Create DataFrame
cart = {'Product': ['Mobile', 'Stoves', 'Desktops', 'Sofa', 'Laptop'],
    'Price': [60000, 48000, 32000, 15000, 25000],
    '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)
```

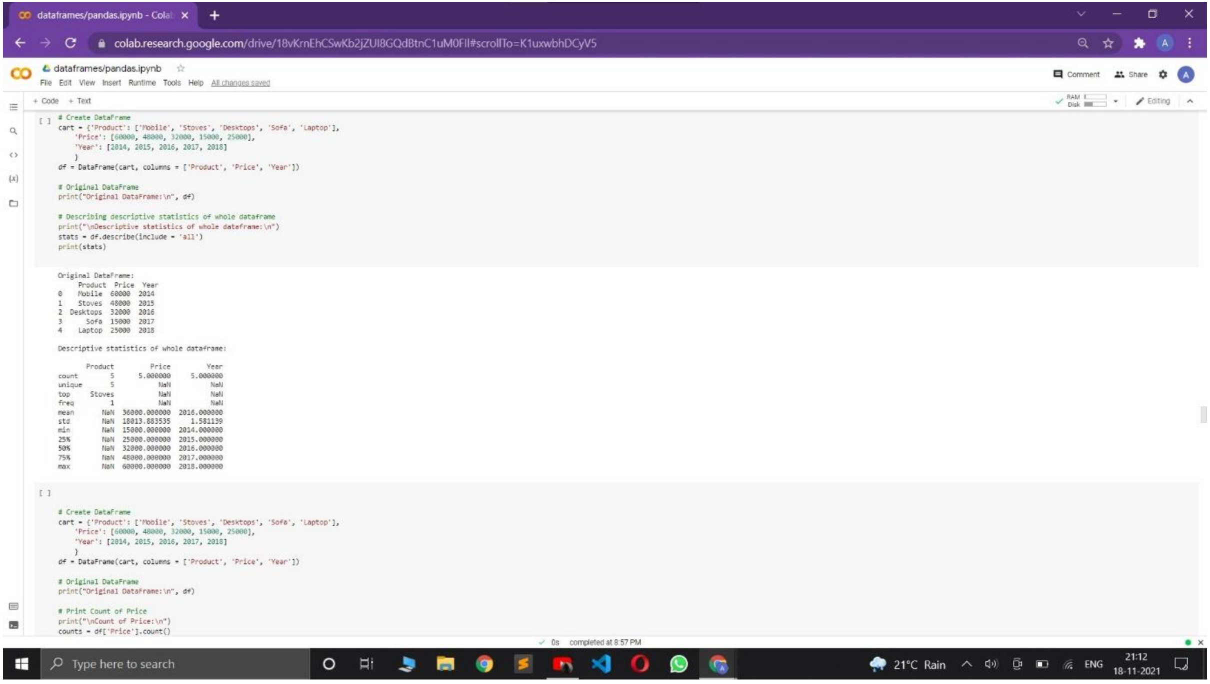
```
Original DataFrame:
     Product  Price  Year
0    Mobile  60000  2014
1    Stoves  48000  2015
2  Desktops  32000  2016
3      Sofa  15000  2017
4    Laptop  25000  2018

Descriptive statistics of Price:

count        5.000000
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:11
18-11-2021

```
Descriptive statistics of Price:

count        5.000000
mean     36000.000000
std      18013.883535
min      15000.000000
25%      25000.000000
50%      32000.000000
75%      48000.000000
max      60000.000000
Name: Price, dtype: float64
```

```
[ ] # Import package


    # Create DataFrame
    cart = {'Product': ['Mobile', 'Stoves', 'Desktops', 'Sofa', 'Laptop'],
        'Price': [60000, 48000, 32000, 15000, 25000],
        'Year': [2014, 2015, 2016, 2017, 2018]
        }
    df = DataFrame(cart, columns = ['Product', 'Price', 'Year'])

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

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

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

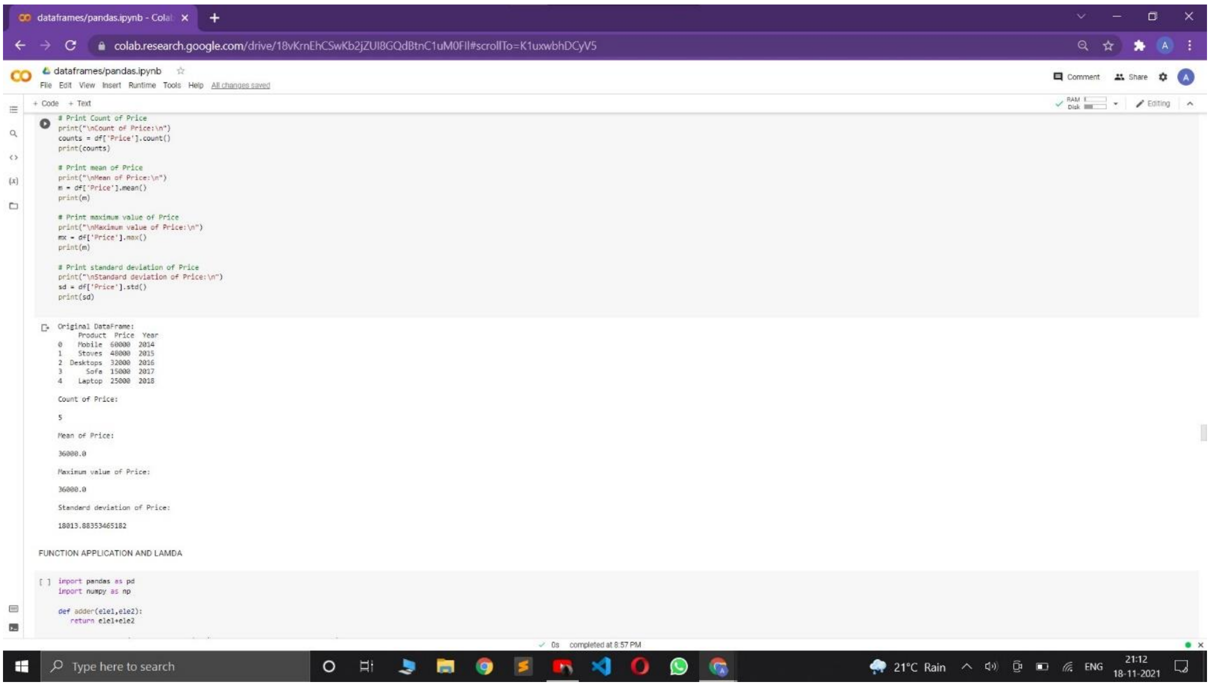
```
Original DataFrame:
     Product  Price  Year
0    Mobile  60000  2014
1    Stoves  48000  2015
2  Desktops  32000  2016
3      Sofa  15000  2017
4    Laptop  25000  2018

Descriptive statistics of year:

count       5.000000
mean     2016.000000
std         1.581139
min      2014.000000
25%      2015.000000
50%      2016.000000
75%      2017.000000
max      2018.000000
Name: Year, dtype: float64
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:12
18-11-2021

```
[ ] # Create DataFrame
    cart = {'Product': ['Mobile', 'Stoves', 'Desktops', 'Sofa', 'Laptop'],
        'Price': [60000, 48000, 32000, 15000, 25000],
        'Year': [2014, 2015, 2016, 2017, 2018]
        }
    df = DataFrame(cart, columns = ['Product', 'Price', 'Year'])

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

    # Describing descriptive statistics of whole dataframe
    print("\nDescriptive statistics of whole dataframe:\n")
    stats = df.describe(include = 'all')
    print(stats)
```

```
Original DataFrame:
     Product  Price  Year
0    Mobile  60000  2014
1    Stoves  48000  2015
2  Desktops  32000  2016
3      Sofa  15000  2017
4    Laptop  25000  2018

Descriptive statistics of whole dataframe:

       Product         Price         Year
count        5      5.000000     5.000000
unique       5           NaN          NaN
top     Stoves           NaN          NaN
freq         1           NaN          NaN
mean       NaN  36000.000000  2016.000000
std        NaN  18013.883535     1.581139
min        NaN  15000.000000  2014.000000
25%        NaN  25000.000000  2015.000000
50%        NaN  32000.000000  2016.000000
75%        NaN  48000.000000  2017.000000
max        NaN  60000.000000  2018.000000
```

```
[ ]


    # Create DataFrame
    cart = {'Product': ['Mobile', 'Stoves', 'Desktops', 'Sofa', 'Laptop'],
        'Price': [60000, 48000, 32000, 15000, 25000],
        'Year': [2014, 2015, 2016, 2017, 2018]
        }
    df = DataFrame(cart, columns = ['Product', 'Price', 'Year'])

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

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

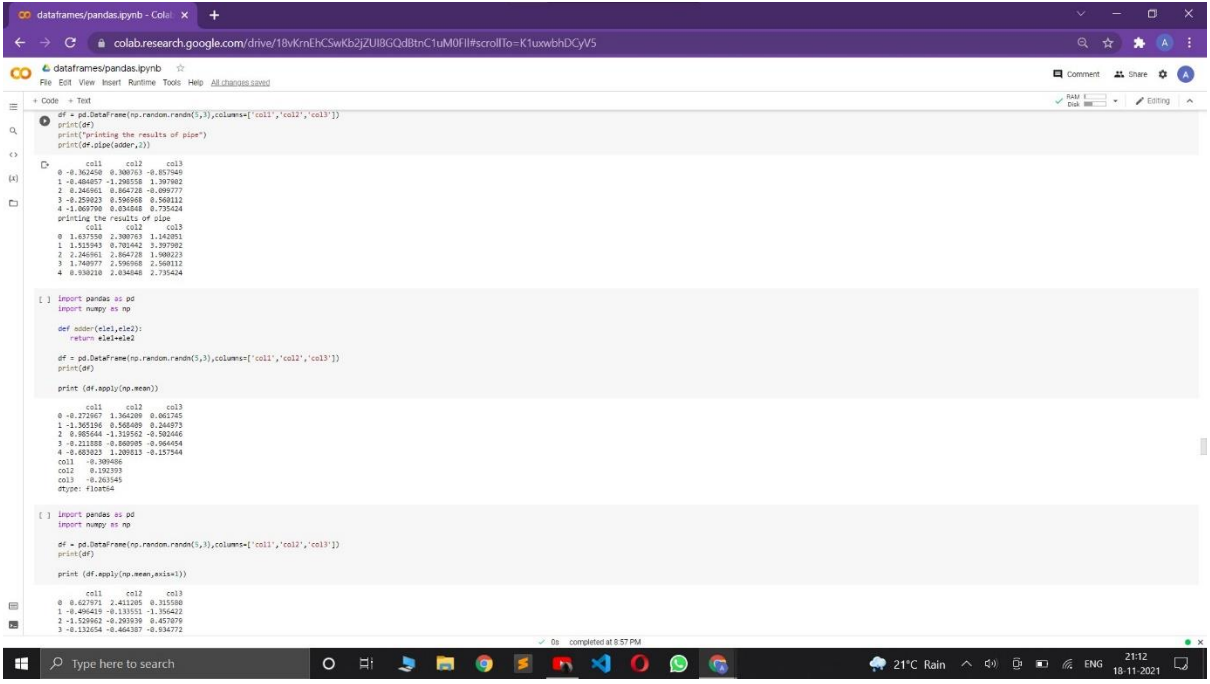

dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FlI#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:12
18-11-2021

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

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

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

# Print standard deviation of Price
print("\nStandard deviation of Price:\n")
sd = df['Price'].std()
print(sd)
```

```
Original DataFrame:
    Product  Price  Year
0    Mobile  60000  2014
1    Stoves  48000  2015
2  Desktops  32000  2016
3      Sofa  15000  2017
4    Laptop  25000  2018

Count of Price:

5

Mean of Price:

36000.0

Maximum value of Price:

36000.0

Standard deviation of Price:

18013.88353465182
```

FUNCTION APPLICATION AND LAMDA

```
import pandas as pd
import numpy as np

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


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FlI#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:12
18-11-2021

```
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))
```

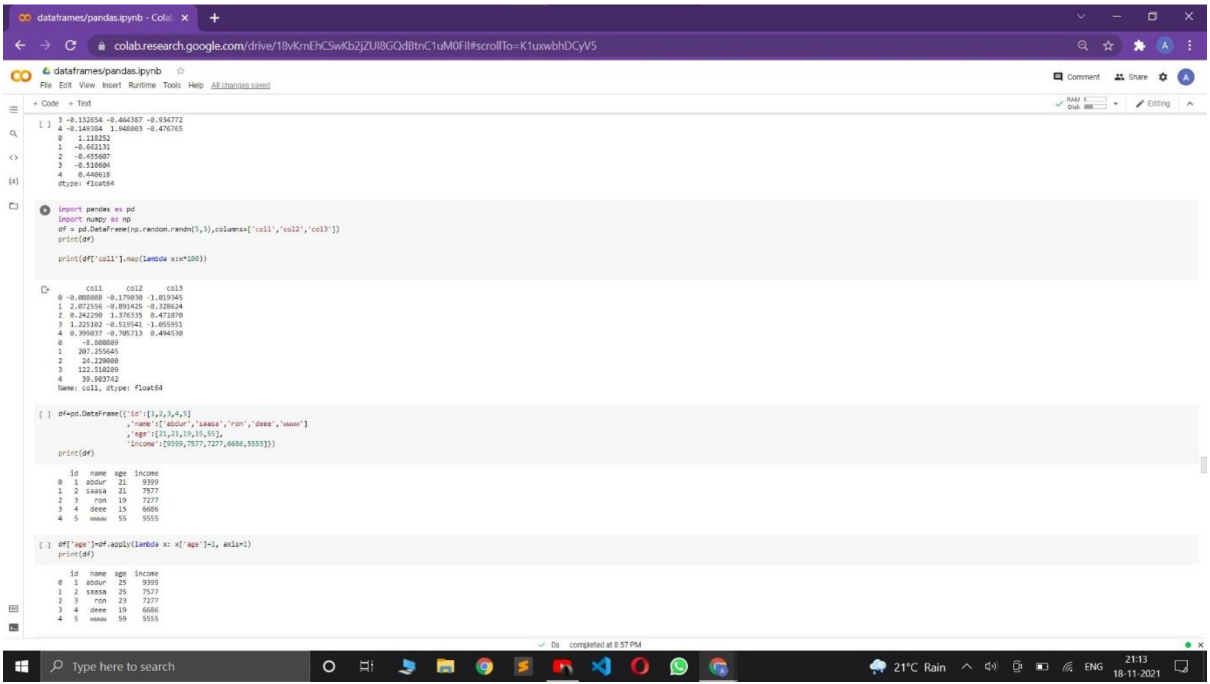
```
       col1      col2      col3
0 -0.362450  0.300763 -0.857949
1 -0.484057 -1.298558  1.397902
2  0.246961  0.864728 -0.099777
3 -0.259023  0.596968  0.560112
4 -1.069790  0.034848  0.735424
printing the results of pipe
       col1      col2      col3
0  1.637550  2.300763  1.142051
1  1.515943  0.701442  3.397902
2  2.246961  2.864728  1.900223
3  1.740977  2.596968  2.560112
4  0.930210  2.034848  2.735424
```

```
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))
```

```
       col1      col2      col3
0 -0.272967  1.364209  0.061745
1 -1.365196  0.568409  0.244973
2  0.985644 -1.319562 -0.502446
3 -0.211888 -0.860905 -0.964454
4 -0.683023  1.209813 -0.157544
col1   -0.309486
col2    0.192393
col3   -0.263545
dtype: float64
```

```
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))
```

```
       col1      col2      col3
0  0.627971  2.411205  0.315580
1 -0.496419 -0.133551 -1.356422
2 -1.529962 -0.293939  0.457079
3 -0.132654 -0.464387 -0.934772
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Editing
completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:13
18-11-2021

```
3 -0.132654 -0.464387 -0.934772
4 -0.149384  1.948003 -0.476765
0    1.118252
1   -0.662131
2   -0.455607
3   -0.510604
4    0.440618
dtype: float64
```

```
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))
```

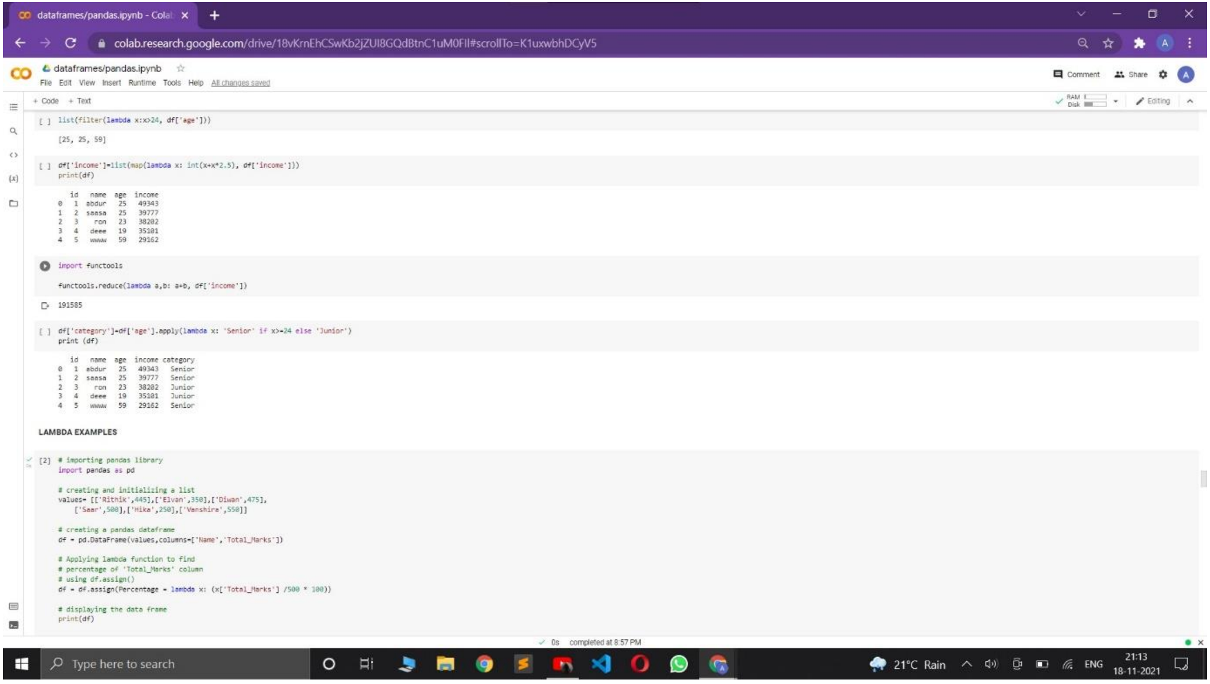
```
       col1      col2      col3
0 -0.088088 -0.179030 -1.019345
1  2.072556 -0.891425 -0.328624
2  0.242290  1.376335  0.471070
3  1.225102 -0.519541 -1.055951
4  0.399037 -0.705713  0.494530
0     -8.808809
1    207.255645
2     24.229000
3    122.510209
4     39.903742
Name: col1, dtype: float64
```

```
df=pd.DataFrame({'id':[1,2,3,4,5]
                 ,'name':['abdur','saasa','ron','deee','wwww']
                 ,'age':[21,21,19,15,55],
                 'income':[9399,7577,7277,6686,5555]})
print(df)
```

```
   id   name  age  income
0   1  abdur   21    9399
1   2  saasa   21    7577
2   3    ron   19    7277
3   4   deee   15    6686
4   5   wwww   55    5555
```

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

```
   id   name  age  income
0   1  abdur   25    9399
1   2  saasa   25    7577
2   3    ron   23    7277
3   4   deee   19    6686
4   5   wwww   59    5555
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Editing
completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:13
18-11-2021

```
list(filter(lambda x:x>24, df['age']))
```

```
[25, 25, 59]
```

```
df['income']=list(map(lambda x: int(x+x*2.5), df['income']))
print(df)
```

```
   id   name  age  income
0   1  abdur   25   49343
1   2  saasa   25   39777
2   3    ron   23   38202
3   4   deee   19   35101
4   5   wwww   59   29162
```

```
import functools

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

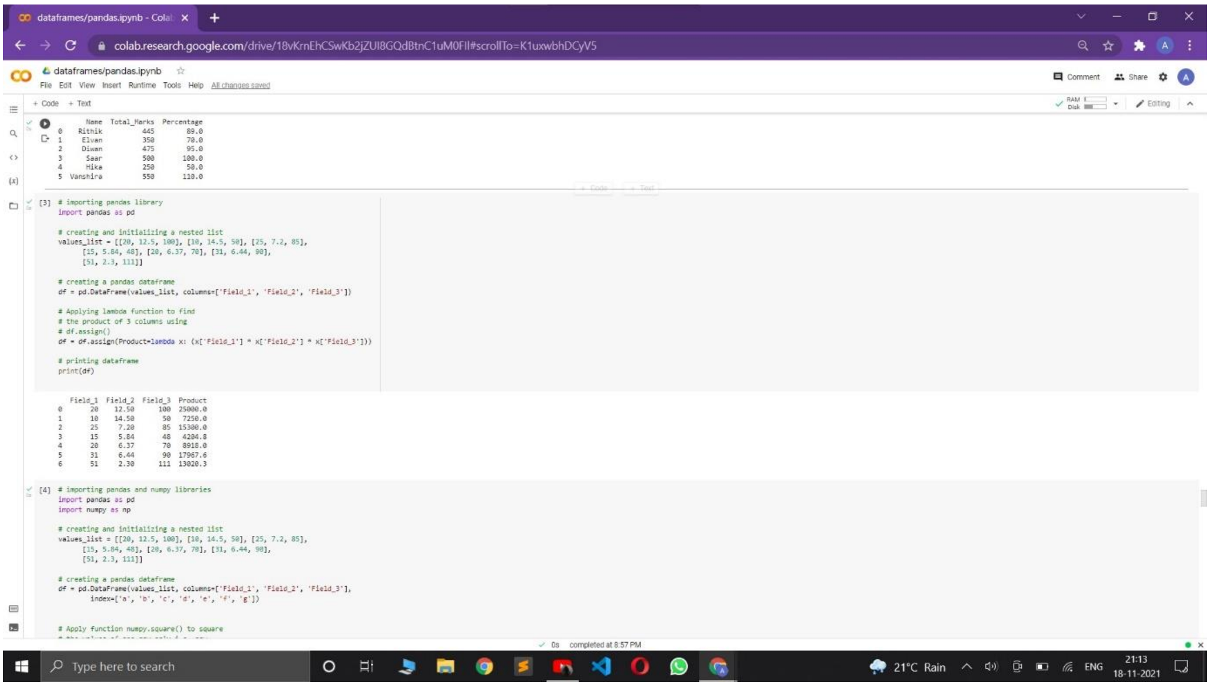
```
191585
```

```
df['category']=df['age'].apply(lambda x: 'Senior' if x>=24 else 'Junior')
print (df)
```

```
   id   name  age  income category
0   1  abdur   25   49343   Senior
1   2  saasa   25   39777   Senior
2   3    ron   23   38202   Junior
3   4   deee   19   35101   Junior
4   5   wwww   59   29162   Senior
```

LAMBDA EXAMPLES

```
# importing pandas library
import pandas as pd

# creating and initializing a list
values= [['Rithik',445],['Elvan',350],['Diwan',475],
    ['Saar',500],['Hika',250],['Vanshira',550]]

# creating a pandas dataframe
df = pd.DataFrame(values,columns=['Name','Total_Marks'])

# Applying lambda function to find
# percentage of 'Total_Marks' column
# using df.assign()
df = df.assign(Percentage = lambda x: (x['Total_Marks'] /500 * 100))

# displaying the data frame
print(df)
```

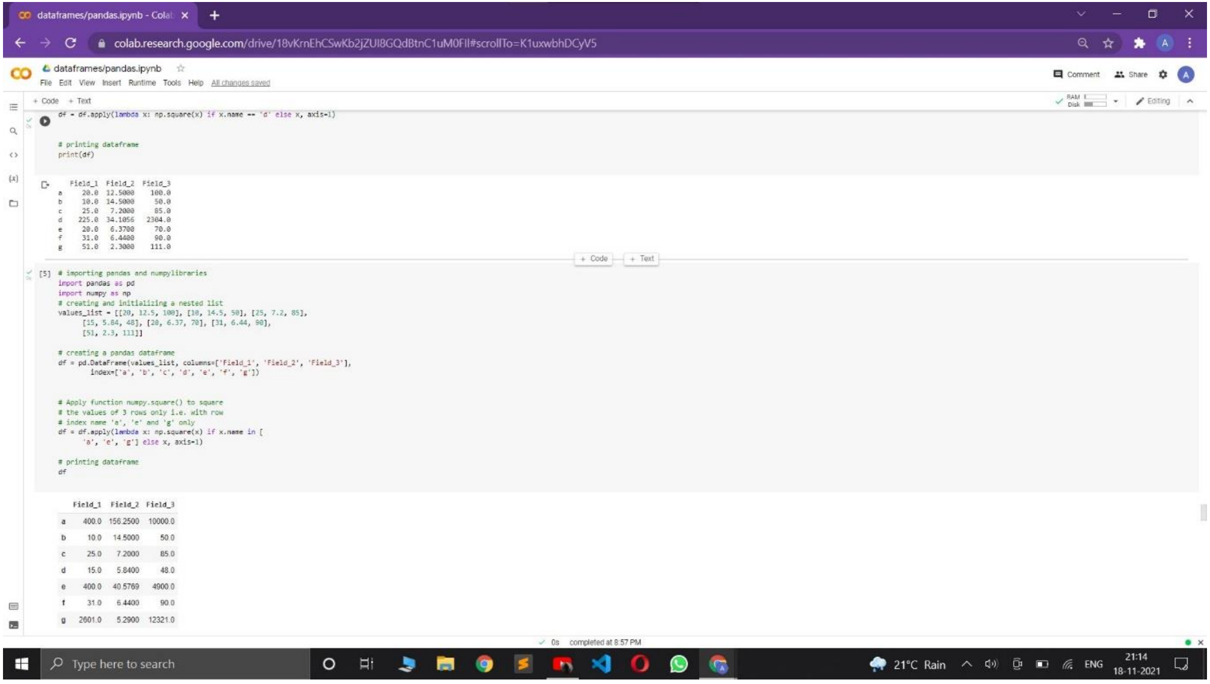

dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment
Share
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:13
18-11-2021

```
       Name  Total_Marks  Percentage
0    Rithik          445        89.0
1     Elvan          350        70.0
2     Diwan          475        95.0
3      Saar          500       100.0
4      Hika          250        50.0
5  Vanshira          550       110.0
```

```
[3] # importing pandas library
    import pandas as pd

    # creating and initializing a nested list
    values_list = [[20, 12.5, 100], [10, 14.5, 50], [25, 7.2, 85],
          [15, 5.84, 48], [20, 6.37, 70], [31, 6.44, 90],
          [51, 2.3, 111]]

    # creating a pandas dataframe
    df = pd.DataFrame(values_list, columns=['Field_1', 'Field_2', 'Field_3'])

    # Applying lambda function to find
    # the product of 3 columns using
    # df.assign()
    df = df.assign(Product=lambda x: (x['Field_1'] * x['Field_2'] * x['Field_3']))

    # printing dataframe
    print(df)
```

```
   Field_1  Field_2  Field_3  Product
0       20    12.50      100  25000.0
1       10    14.50       50   7250.0
2       25     7.20       85  15300.0
3       15     5.84       48   4204.8
4       20     6.37       70   8918.0
5       31     6.44       90  17967.6
6       51     2.30      111  13020.3
```

```
[4] # importing pandas and numpy libraries
    import pandas as pd
    import numpy as np

    # creating and initializing a nested list
    values_list = [[20, 12.5, 100], [10, 14.5, 50], [25, 7.2, 85],
          [15, 5.84, 48], [20, 6.37, 70], [31, 6.44, 90],
          [51, 2.3, 111]]

    # creating a pandas dataframe
    df = pd.DataFrame(values_list, columns=['Field_1', 'Field_2', 'Field_3'],
           index=['a', 'b', 'c', 'd', 'e', 'f', 'g'])


    # Apply function numpy.square() to square
```

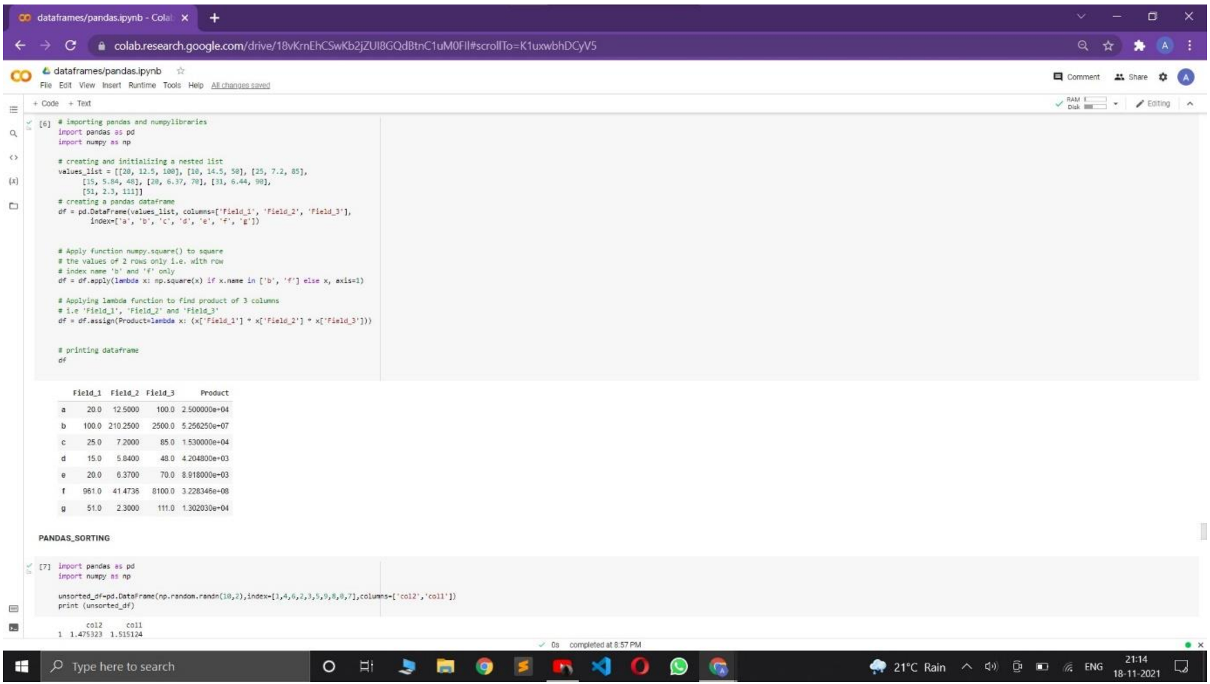

dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment
Share
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:14
18-11-2021

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


    # printing dataframe
    print(df)
```

```
   Field_1  Field_2  Field_3
a     20.0  12.5000    100.0
b     10.0  14.5000     50.0
c     25.0   7.2000     85.0
d    225.0  34.1056   2304.0
e     20.0   6.3700     70.0
f     31.0   6.4400     90.0
g     51.0   2.3000    111.0
```

```
[5] # importing pandas and numpylibraries
    import pandas as pd
    import numpy as np
    # creating and initializing a nested list
    values_list = [[20, 12.5, 100], [10, 14.5, 50], [25, 7.2, 85],
          [15, 5.84, 48], [20, 6.37, 70], [31, 6.44, 90],
          [51, 2.3, 111]]

    # creating a pandas dataframe
    df = pd.DataFrame(values_list, columns=['Field_1', 'Field_2', 'Field_3'],
           index=['a', 'b', 'c', 'd', 'e', 'f', 'g'])


    # Apply function numpy.square() to square
    # the values of 3 rows only i.e. with row
    # index name 'a', 'e' and 'g' only
    df = df.apply(lambda x: np.square(x) if x.name in [
          'a', 'e', 'g'] else x, axis=1)

    # printing dataframe
    df
```

	Field_1	Field_2	Field_3
a	400.0	156.2500	10000.0
b	10.0	14.5000	50.0
c	25.0	7.2000	85.0
d	15.0	5.8400	48.0
e	400.0	40.5769	4900.0
f	31.0	6.4400	90.0
g	2601.0	5.2900	12321.0


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FlI#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
✓ 0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:14
18-11-2021

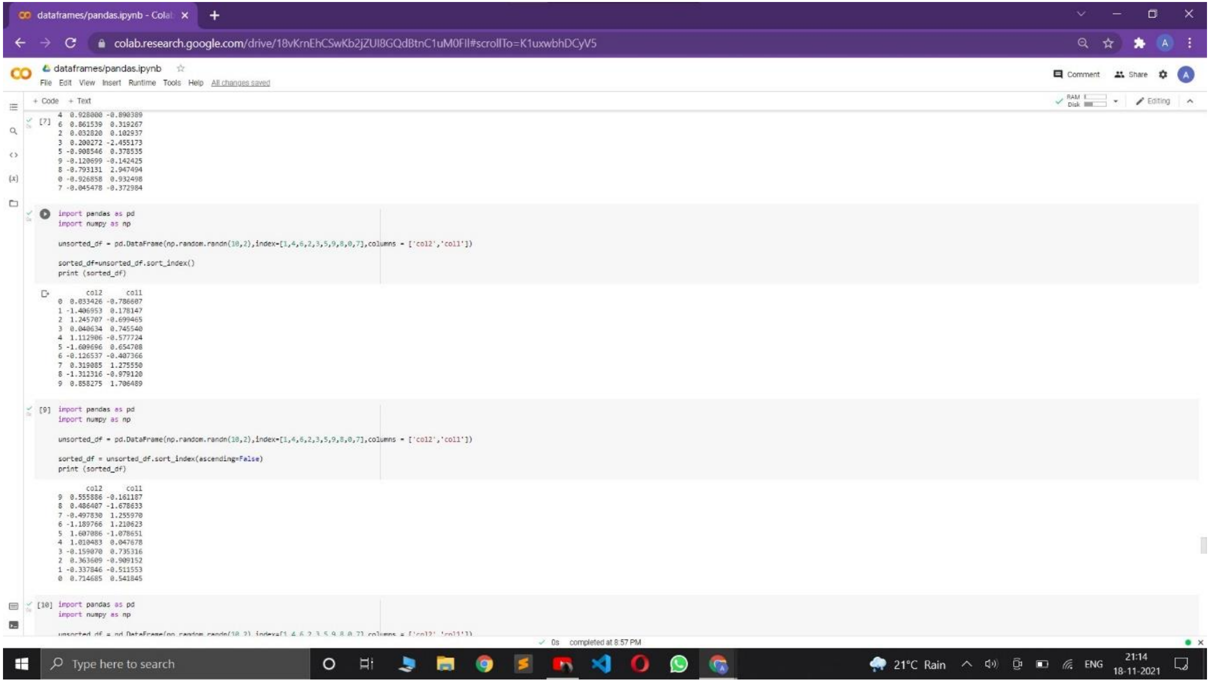
```
[6] # importing pandas and numpylibraries
    import pandas as pd
    import numpy as np

    # creating and initializing a nested list
    values_list = [[20, 12.5, 100], [10, 14.5, 50], [25, 7.2, 85],
          [15, 5.84, 48], [20, 6.37, 70], [31, 6.44, 90],
          [51, 2.3, 111]]
    # creating a pandas dataframe
    df = pd.DataFrame(values_list, columns=['Field_1', 'Field_2', 'Field_3'],
            index=['a', 'b', 'c', 'd', 'e', 'f', 'g'])


    # Apply function numpy.square() to square
    # the values of 2 rows only i.e. with row
    # index name 'b' and 'f' only
    df = df.apply(lambda x: np.square(x) if x.name in ['b', 'f'] else x, axis=1)

    # Applying lambda function to find product of 3 columns
    # i.e 'Field_1', 'Field_2' and 'Field_3'
    df = df.assign(Product=lambda x: (x['Field_1'] * x['Field_2'] * x['Field_3']))


    # printing dataframe
    df
```

	Field_1	Field_2	Field_3	Product
a	20.0	12.5000	100.0	2.500000e+04
b	100.0	210.2500	2500.0	5.256250e+07
c	25.0	7.2000	85.0	1.530000e+04
d	15.0	5.8400	48.0	4.204800e+03
e	20.0	6.3700	70.0	8.918000e+03
f	961.0	41.4736	8100.0	3.228346e+08
g	51.0	2.3000	111.0	1.302030e+04

PANDAS_SORTING

```
[7] 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)
```

```
       col2      col1
1  1.475323  1.515124
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FlI#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
✓ 0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:14
18-11-2021

```
4  0.928000 -0.890389
6  0.861539  0.319267
2  0.032820  0.102937
3  0.200272 -2.455173
5 -0.908546  0.378535
9 -0.120699 -0.142425
8 -0.793131  2.947494
0 -0.926858  0.932498
7 -0.045478 -0.372984
```

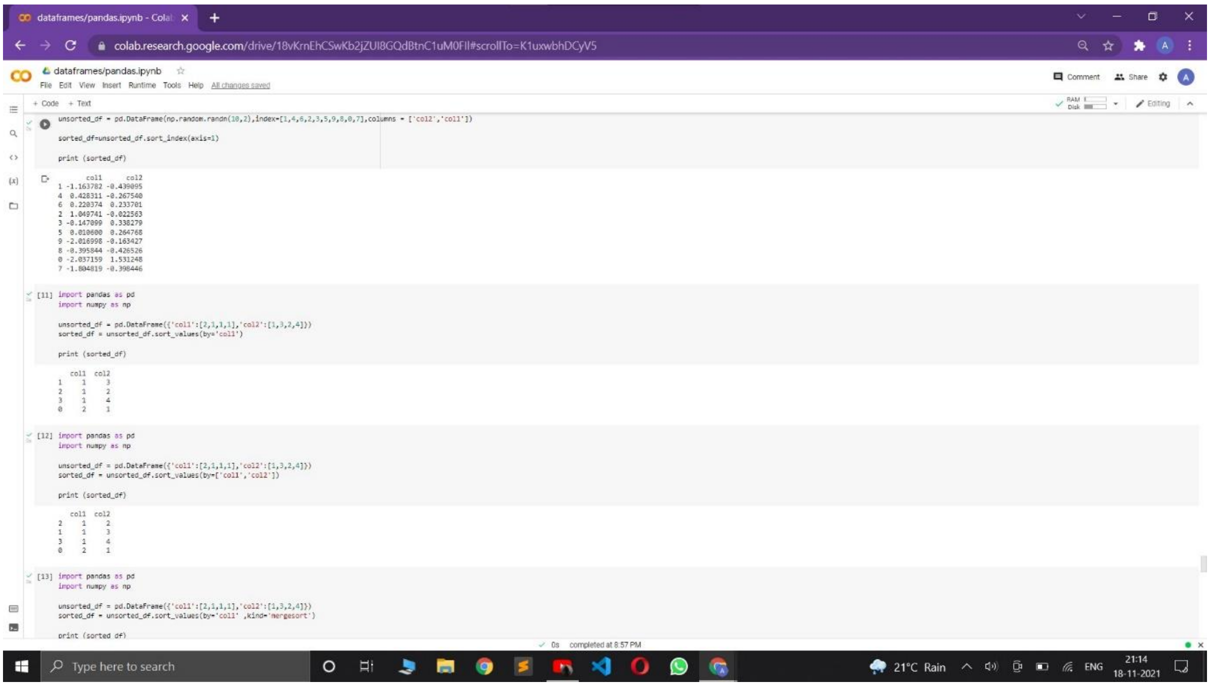
```
import pandas as pd
import numpy as np

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)
```

```
       col2      col1
0  0.033426 -0.786607
1 -1.406953  0.178147
2  1.245707 -0.699465
3  0.040634  0.745540
4  1.112906 -0.577724
5 -1.609696  0.654708
6 -0.126537 -0.407366
7  0.319085  1.275550
8 -1.312316 -0.979120
9  0.858275  1.706489
```

```
[9] 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)
```

```
       col2      col1
9  0.555886 -0.161187
8  0.486407 -1.678633
7 -0.497830  1.255970
6 -1.189766  1.210623
5  1.607086 -1.078651
4  1.010483  0.047678
3 -0.159070  0.735316
2  0.363609 -0.909152
1 -0.337846 -0.511553
0  0.714685  0.541845
```

```
[10] 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'])
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:14
18-11-2021

```
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)
```

```
       col1      col2
1 -1.163782 -0.439095
4  0.428311 -0.267540
6  0.220374  0.233701
2  1.049741 -0.022563
3 -0.147099  0.338279
5  0.010600  0.264768
9 -2.016998 -0.163427
8 -0.395844 -0.426526
0 -2.037159  1.531248
7 -1.804819 -0.398446
```

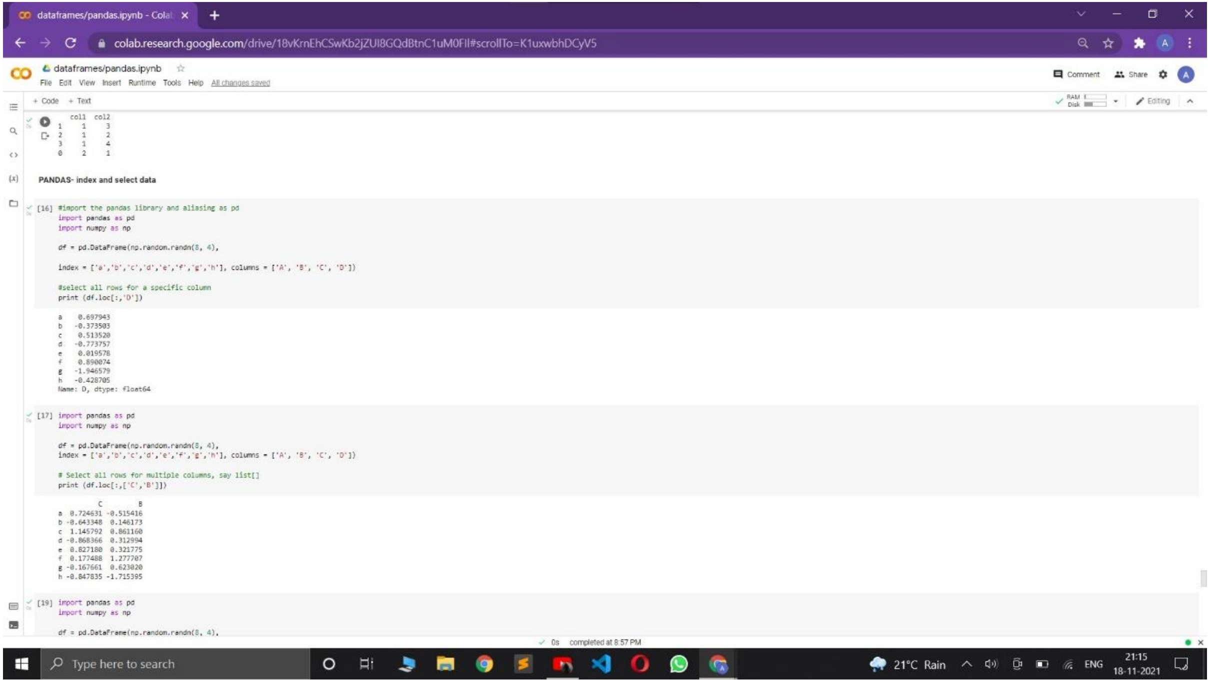
```
[11] import pandas as pd
     import numpy as np

     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)
```

```
   col1  col2
1     1     3
2     1     2
3     1     4
0     2     1
```

```
[12] 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)
```

```
   col1  col2
2     1     2
1     1     3
3     1     4
0     2     1
```

```
[13] 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)
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:15
18-11-2021

```
   col1  col2
1     1     3
2     1     2
3     1     4
0     2     1
```

PANDAS- index and select data

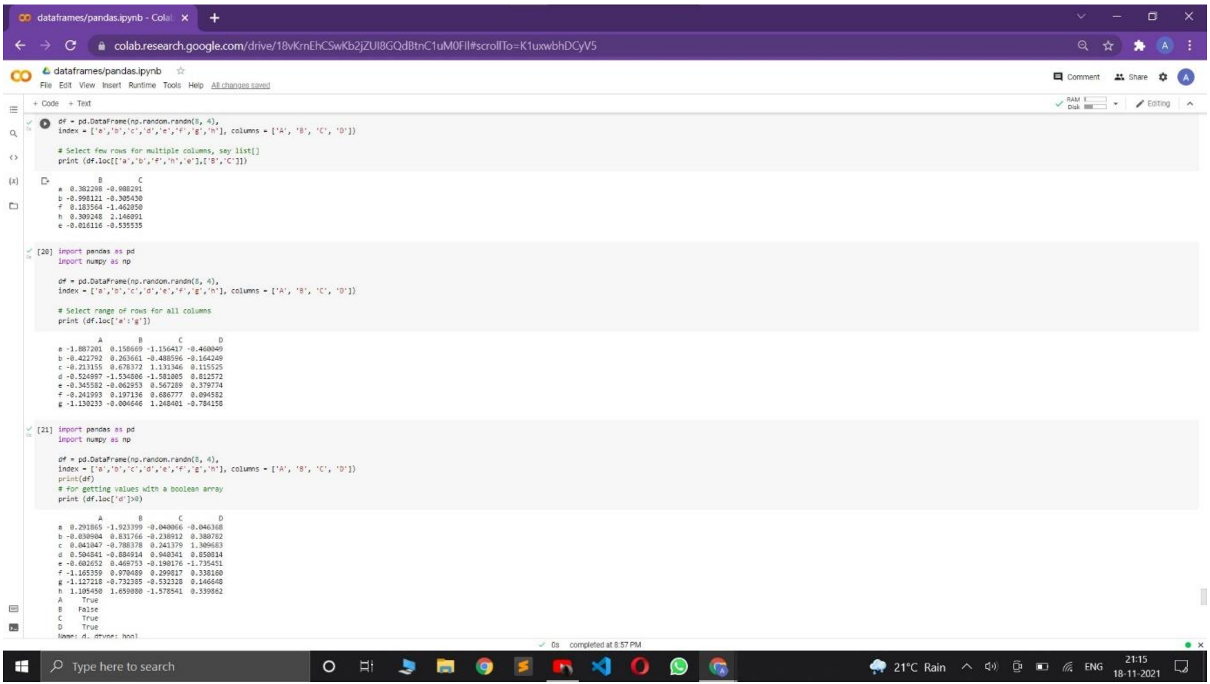
```
[16] #import the pandas library and aliasing as pd
     import pandas as pd
     import numpy as np

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

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

     #select all rows for a specific column
     print (df.loc[:,'D'])
```

```
a    0.697943
b   -0.373503
c    0.513520
d   -0.773757
e    0.019578
f    0.890074
g   -1.946579
h   -0.428705
Name: D, dtype: float64
```

```
[17] 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'])

     # Select all rows for multiple columns, say list[]
     print (df.loc[:,['C','B']])
```

```
          C         B
a  0.724631 -0.515416
b -0.643348  0.146173
c  1.145792  0.861160
d -0.868366  0.312994
e  0.827180  0.321775
f  0.177488  1.277707
g -0.167661  0.623020
h -0.847835 -1.715395
```

```
[19] import pandas as pd
     import numpy as np

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


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FlI#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:15
18-11-2021

```
df = pd.DataFrame(np.random.randn(8, 4),
index = ['a','b','c','d','e','f','g','h'], columns = ['A', 'B', 'C', 'D'])

# Select few rows for multiple columns, say list[]
print (df.loc[['a','b','f','h','e'],['B','C']])
```

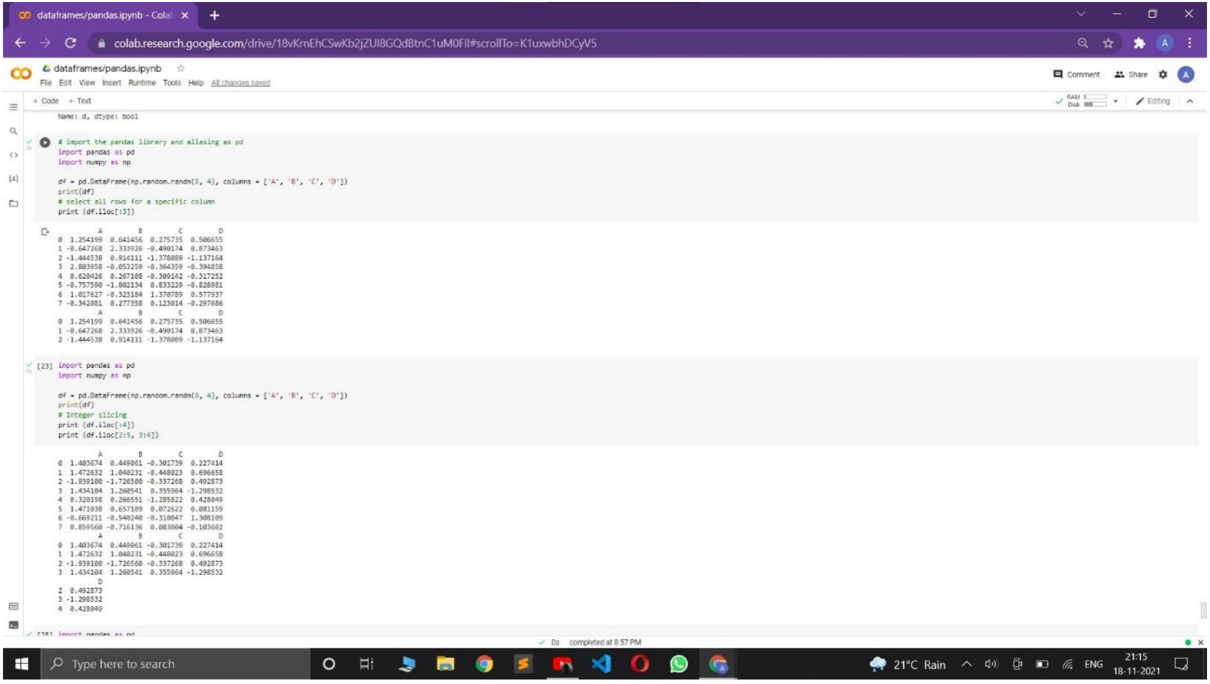
```
          B         C
a  0.382298 -0.988291
b -0.998121 -0.305430
f  0.183564 -1.462050
h  0.309248  2.146091
e -0.016116 -0.535535
```

```
[20] 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'])

     # Select range of rows for all columns
     print (df.loc['a':'g'])
```

```
          A         B         C         D
a -1.887201  0.158669 -1.156417 -0.460049
b -0.422792  0.263661 -0.488596 -0.164249
c -0.213155  0.678372  1.131346  0.115525
d -0.524997 -1.534806 -1.581005  0.812572
e -0.345582 -0.062953  0.567289  0.379774
f -0.241993  0.197136  0.686777  0.094582
g -1.130233 -0.004646  1.248401 -0.784158
```

```
[21] 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)
     # for getting values with a boolean array
     print (df.loc['d']>0)
```

```
          A         B         C         D
a  0.291865 -1.923399 -0.040066 -0.046368
b -0.030904  0.831766 -0.238912  0.380782
c  0.041047 -0.788378  0.241379  1.309683
d  0.504841 -0.884914  0.940341  0.850814
e -0.602652  0.469753 -0.190176 -1.735451
f -1.165359  0.970489  0.299817  0.338160
g -1.127218 -0.732385 -0.532328  0.146648
h  1.105450  1.659080 -1.578541  0.339862
A     True
B    False
C     True
D     True
Name: d, dtype: bool
```


dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FlI#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:15
18-11-2021

```
Name: d, dtype: bool
```

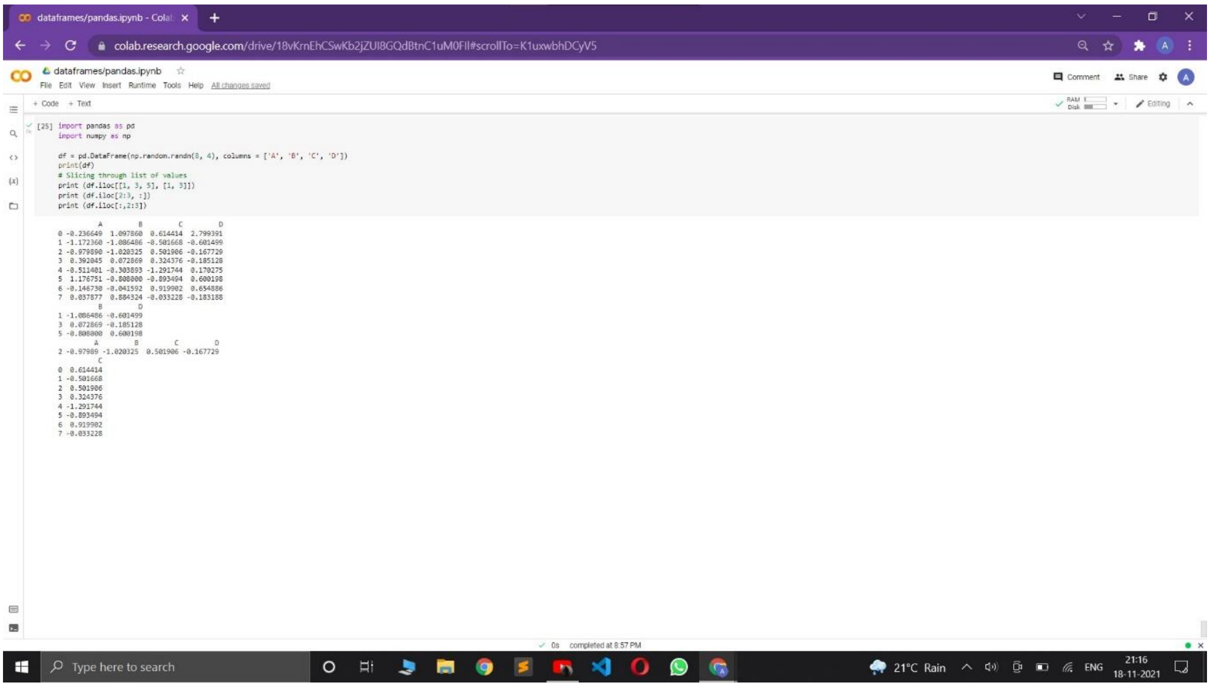
```
# import the pandas library and aliasing as pd
import pandas as pd
import numpy as np

df = pd.DataFrame(np.random.randn(8, 4), columns = ['A', 'B', 'C', 'D'])
print(df)
# select all rows for a specific column
print (df.iloc[:3])
```

```
          A         B         C         D
0  1.254199  0.641456  0.275735  0.506655
1 -0.647268  2.333926 -0.490174  0.873463
2 -1.444538  0.914111 -1.378089 -1.137164
3  2.803958 -0.053259 -0.364359 -0.394858
4  0.620426  0.267108 -0.309142 -0.317252
5 -0.757590 -1.802134  0.833229 -0.828981
6  1.017627 -0.323184  1.370789  0.577937
7 -0.342081  0.277358  0.123014 -0.297686
          A         B         C         D
0  1.254199  0.641456  0.275735  0.506655
1 -0.647268  2.333926 -0.490174  0.873463
2 -1.444538  0.914111 -1.378089 -1.137164
```

```
[23] import pandas as pd
     import numpy as np

     df = pd.DataFrame(np.random.randn(8, 4), columns = ['A', 'B', 'C', 'D'])
     print(df)
     # Integer slicing
     print (df.iloc[:4])
     print (df.iloc[2:5, 3:4])
```

```
          A         B         C         D
0  1.403674  0.449061 -0.301739  0.227414
1  1.472632  1.040231 -0.448023  0.696658
2 -1.939100 -1.726560 -0.337268  0.492873
3  1.434104  1.260541  0.355964 -1.298532
4  0.320198  0.266551 -1.285822  0.428049
5  1.471038  0.657189  0.072622  0.081159
6 -0.669211 -0.540240 -0.310847  1.308109
7  0.859560 -0.716136  0.083804 -0.103602
          A         B         C         D
0  1.403674  0.449061 -0.301739  0.227414
1  1.472632  1.040231 -0.448023  0.696658
2 -1.939100 -1.726560 -0.337268  0.492873
3  1.434104  1.260541  0.355964 -1.298532
          D
2  0.492873
3 -1.298532
4  0.428049
```

```
[25] import pandas as pd
```

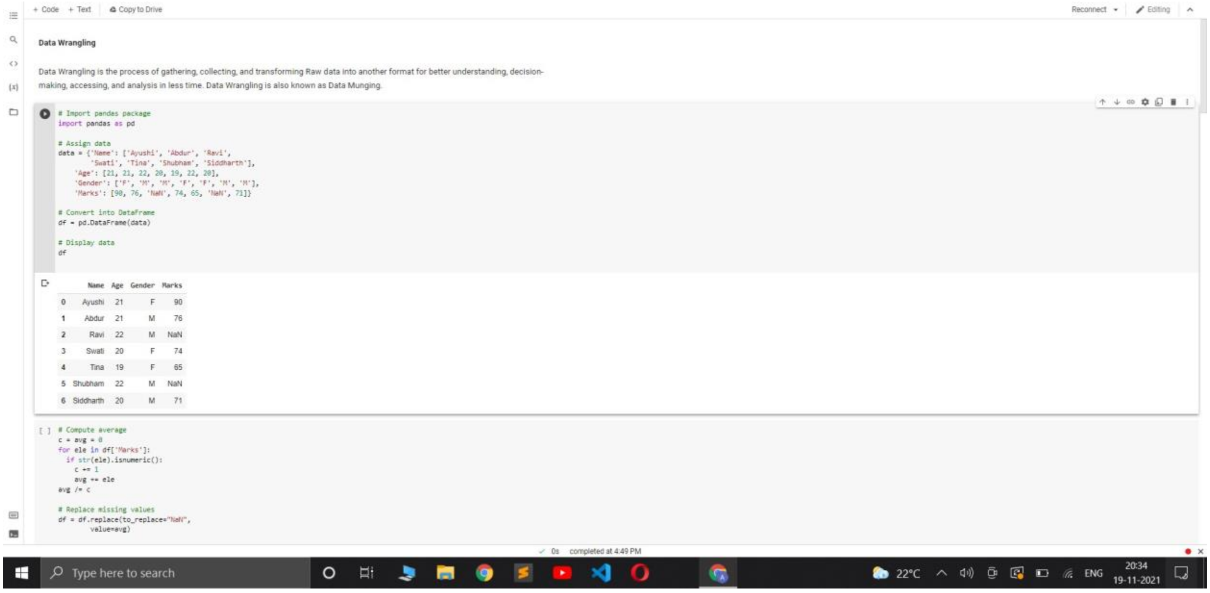

dataframes/pandas.ipynb - Colab
colab.research.google.com/drive/18vKrnEhCSwKb2jZUI8GQdBtnC1uM0FIl#scrollTo=K1uxwbhDCyV5
dataframes/pandas.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
[25]
0s completed at 8:57 PM
Type here to search
21°C Rain
ENG
21:16
18-11-2021

```
import pandas as pd
import numpy as np

df = pd.DataFrame(np.random.randn(8, 4), columns = ['A', 'B', 'C', 'D'])
print(df)
# Slicing through list of values
print (df.iloc[[1, 3, 5], [1, 3]])
print (df.iloc[2:3, :])
print (df.iloc[:,2:3])
```

```
          A         B         C         D
0 -0.236649  1.097860  0.614414  2.799391
1 -1.172360 -1.086486 -0.501668 -0.601499
2 -0.979890 -1.020325  0.501906 -0.167729
3  0.392045  0.072869  0.324376 -0.185128
4 -0.511401 -0.303893 -1.291744  0.170275
5  1.176751 -0.808000 -0.893494  0.600198
6 -0.146730 -0.041592  0.919902  0.654886
7  0.037877  0.884324 -0.033228 -0.183188
          B         D
1 -1.086486 -0.601499
3  0.072869 -0.185128
5 -0.808000  0.600198
         A         B         C         D
2 -0.97989 -1.020325  0.501906 -0.167729
          C
0  0.614414
1 -0.501668
2  0.501906
3  0.324376
4 -1.291744
5 -0.893494
6  0.919902
7 -0.033228
```

DATA WRANGLING:

Data Wrangling is the process of gathering, collecting, and transforming Raw data into another format for better understanding, decision-making, accessing, and analysis in less time. Data Wrangling is also known as Data Munging.



Data Wrangling

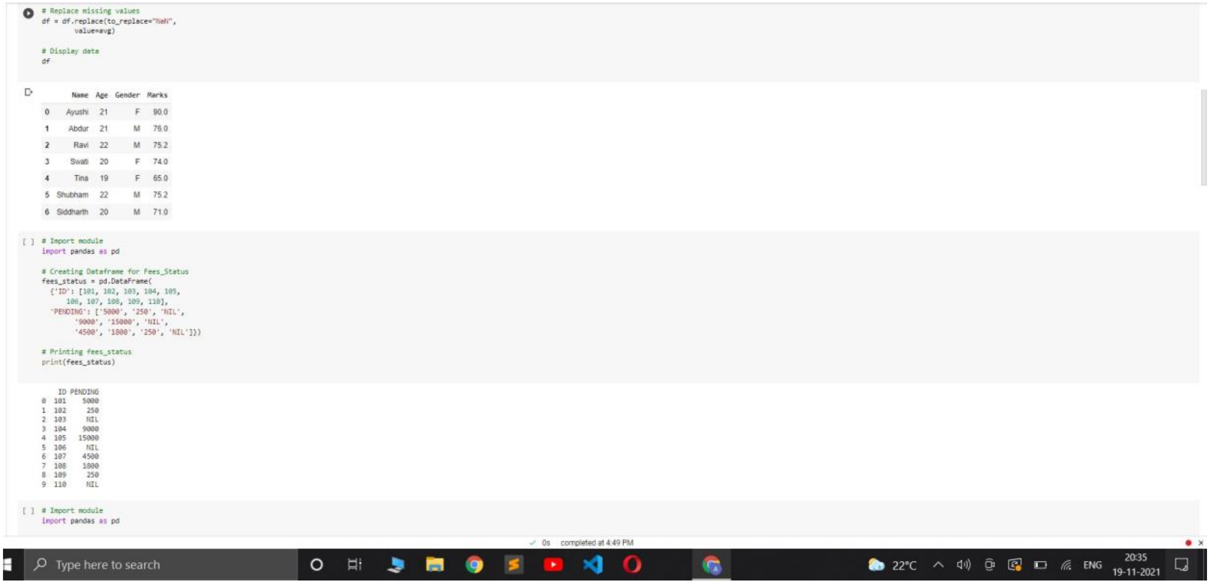
Data Wrangling is the process of gathering, collecting, and transforming Raw data into another format for better understanding, decision-making, accessing, and analysis in less time. Data Wrangling is also known as Data Munging.

```
# Import pandes package
import pandas as pd

# Assign data
data = {'Name': ['Ayushi', 'Abdur', 'Ravi',
         'Swati', 'Tina', 'Shubham', 'Siddharth'],
    'Age': [21, 21, 22, 20, 19, 22, 20],
    'Gender': ['F', 'M', 'M', 'F', 'F', 'M', 'M'],
    'Marks': [90, 76, 'NaN', 74, 65, 'NaN', 71]}

# Convert into DataFrame
df = pd.DataFrame(data)

# Display data
df
```

	Name	Age	Gender	Marks
0	Ayushi	21	F	90
1	Abdur	21	M	76
2	Ravi	22	M	NaN
3	Swati	20	F	74
4	Tina	19	F	65
5	Shubham	22	M	NaN
6	Siddharth	20	M	71

```
# Compute average
c = avg = 0
for ele in df['Marks']:
  if str(ele).isnumeric():
    c += 1
    avg += ele
avg /= c

# Replace missing values
df = df.replace(to_replace="NaN",
        value=avg)
```

✓ 0s completed at 4:49 PM



```
# Replace missing values
df = df.replace(to_replace="NaN",
        value=avg)

# Display data
df
```

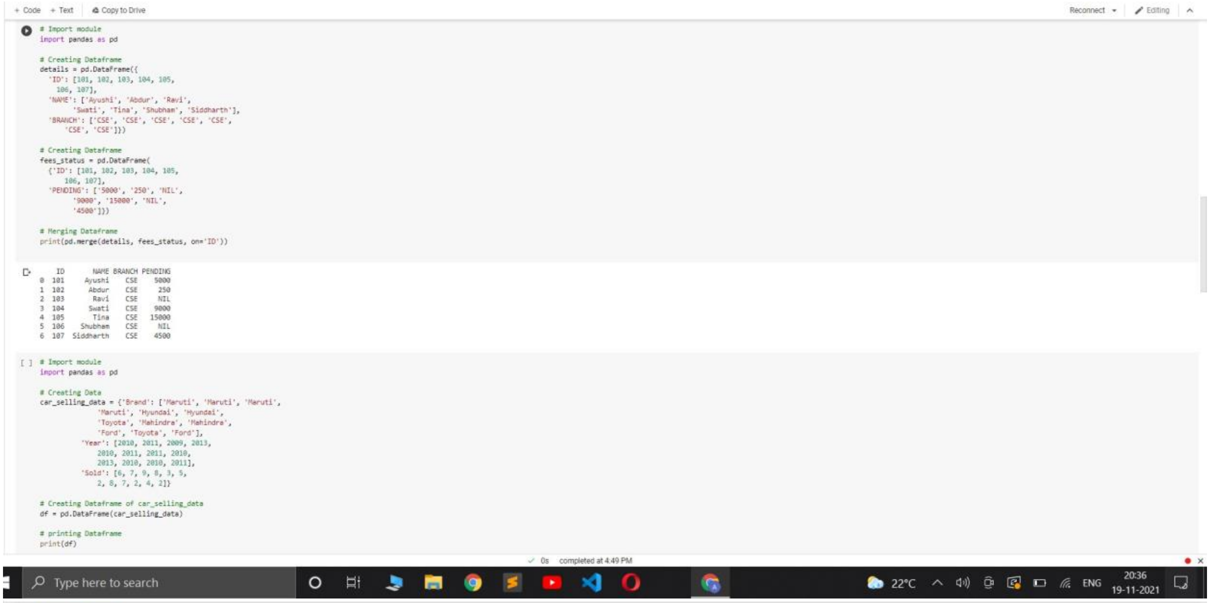
	Name	Age	Gender	Marks
0	Ayushi	21	F	90.0
1	Abdur	21	M	76.0
2	Ravi	22	M	75.2
3	Swati	20	F	74.0
4	Tina	19	F	65.0
5	Shubham	22	M	75.2
6	Siddharth	20	M	71.0

```
# Import module
import pandas as pd

# Creating Dataframe for Fees_Status
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']})

# Printing fees_status
print(fees_status)
```

```
    ID PENDING
0  101    5000
1  102     250
2  103     NIL
3  104    9000
4  105   15000
5  106     NIL
6  107    4500
7  108    1800
8  109     250
9  110     NIL
```

```
# Import module
import pandas as pd
```

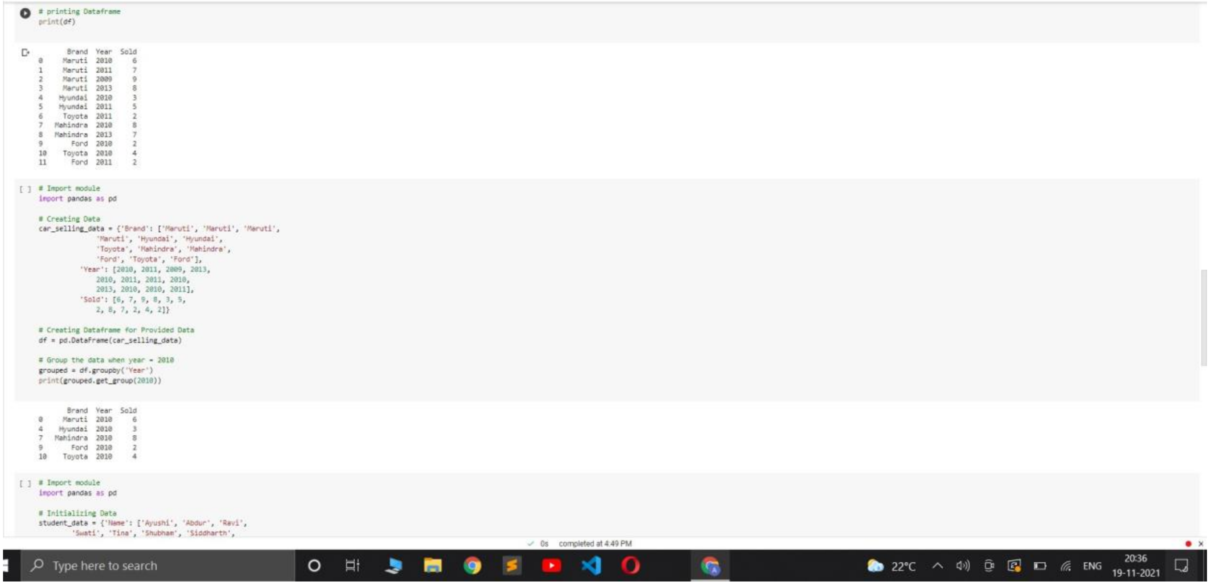
✓ 0s completed at 4:49 PM



```
# Import module
import pandas as pd

# Creating Dataframe
details = pd.DataFrame({
  'ID': [101, 102, 103, 104, 105,
    106, 107],
  'NAME': ['Ayushi', 'Abdur', 'Ravi',
        'Swati', 'Tina', 'Shubham', 'Siddharth'],
  'BRANCH': ['CSE', 'CSE', 'CSE', 'CSE', 'CSE',
      'CSE', 'CSE']})

# Creating Dataframe
fees_status = pd.DataFrame(
  {'ID': [101, 102, 103, 104, 105,
      106, 107],
  'PENDING': ['5000', '250', 'NIL',
        '9000', '15000', 'NIL',
        '4500']})

# Merging Dataframe
print(pd.merge(details, fees_status, on='ID'))
```

```
    ID       NAME BRANCH PENDING
0  101     Ayushi    CSE    5000
1  102      Abdur    CSE     250
2  103       Ravi    CSE     NIL
3  104      Swati    CSE    9000
4  105       Tina    CSE   15000
5  106    Shubham    CSE     NIL
6  107  Siddharth    CSE    4500
```

```
# Import module
import pandas as pd

# Creating Data
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]}

# Creating Dataframe of car_selling_data
df = pd.DataFrame(car_selling_data)

# printing Dataframe
print(df)
```



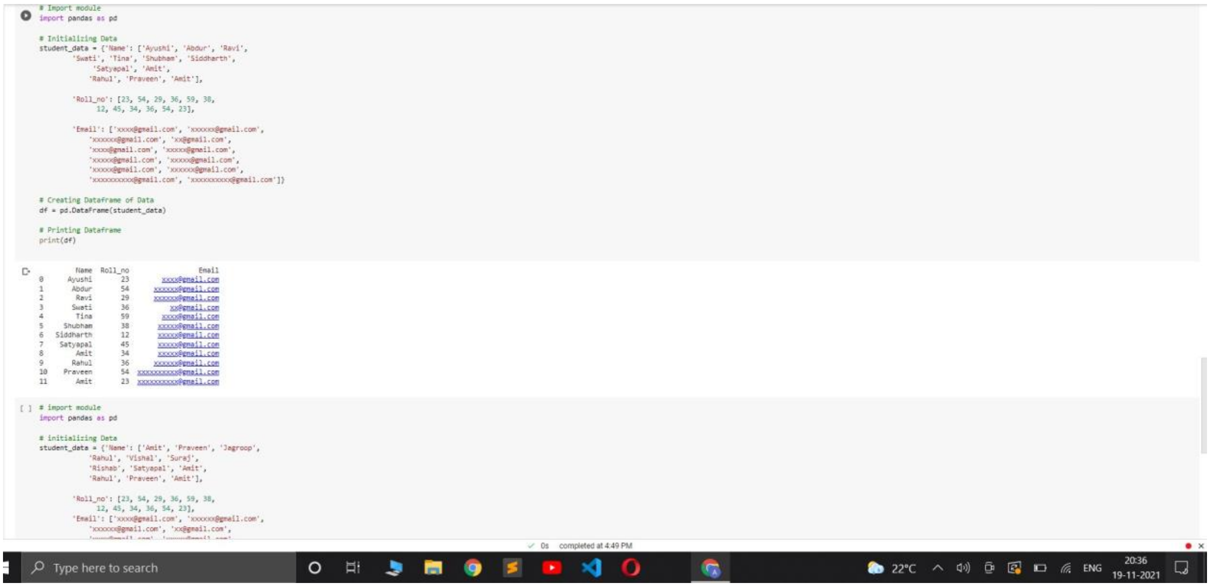
```
# printing Dataframe
print(df)
```

```
       Brand  Year  Sold
0     Maruti  2010     6
1     Maruti  2011     7
2     Maruti  2009     9
3     Maruti  2013     8
4    Hyundai  2010     3
5    Hyundai  2011     5
6     Toyota  2011     2
7   Mahindra  2010     8
8   Mahindra  2013     7
9       Ford  2010     2
10    Toyota  2010     4
11      Ford  2011     2
```

```
# Import module
import pandas as pd

# Creating Data
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]}

# Creating Dataframe for Provided Data
df = pd.DataFrame(car_selling_data)

# Group the data when year = 2010
grouped = df.groupby('Year')
print(grouped.get_group(2010))
```

```
       Brand  Year  Sold
0     Maruti  2010     6
4    Hyundai  2010     3
7   Mahindra  2010     8
9       Ford  2010     2
10    Toyota  2010     4
```

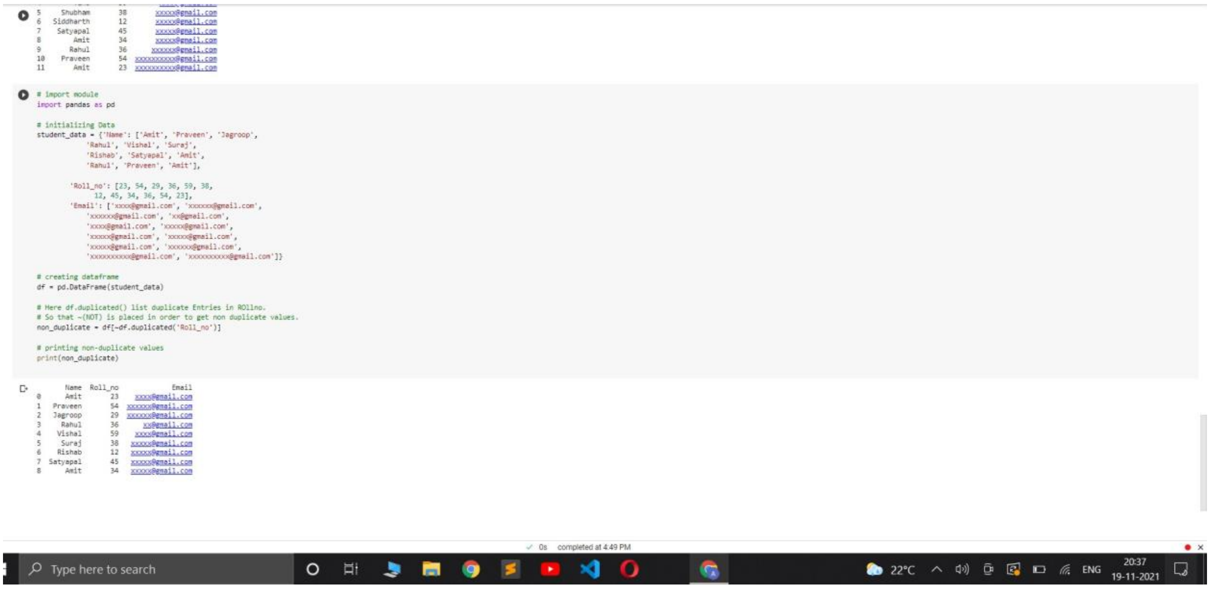
```
# Import module
import pandas as pd

# Initializing Data
student_data = {'Name': ['Ayushi', 'Abdur', 'Ravi',
      'Swati', 'Tina', 'Shubham', 'Siddharth',
```



```
# Import module
import pandas as pd

# Initializing Data
student_data = {'Name': ['Ayushi', 'Abdur', 'Ravi',
        'Swati', 'Tina', 'Shubham', 'Siddharth',
             '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']}

# Creating Dataframe of Data
df = pd.DataFrame(student_data)

# Printing Dataframe
print(df)
```

```
         Name  Roll_no                 Email
0      Ayushi       23        xxxx@gmail.com
1       Abdur       54      xxxxxx@gmail.com
2        Ravi       29      xxxxxx@gmail.com
3       Swati       36          xx@gmail.com
4        Tina       59        xxxx@gmail.com
5     Shubham       38       xxxxx@gmail.com
6   Siddharth       12       xxxxx@gmail.com
7    Satyapal       45       xxxxx@gmail.com
8        Amit       34       xxxxx@gmail.com
9       Rahul       36      xxxxxx@gmail.com
10    Praveen       54  xxxxxxxxxx@gmail.com
11       Amit       23  xxxxxxxxxx@gmail.com
```

```
[ ] # import module
    import pandas as pd

    # initializing Data
    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',
```



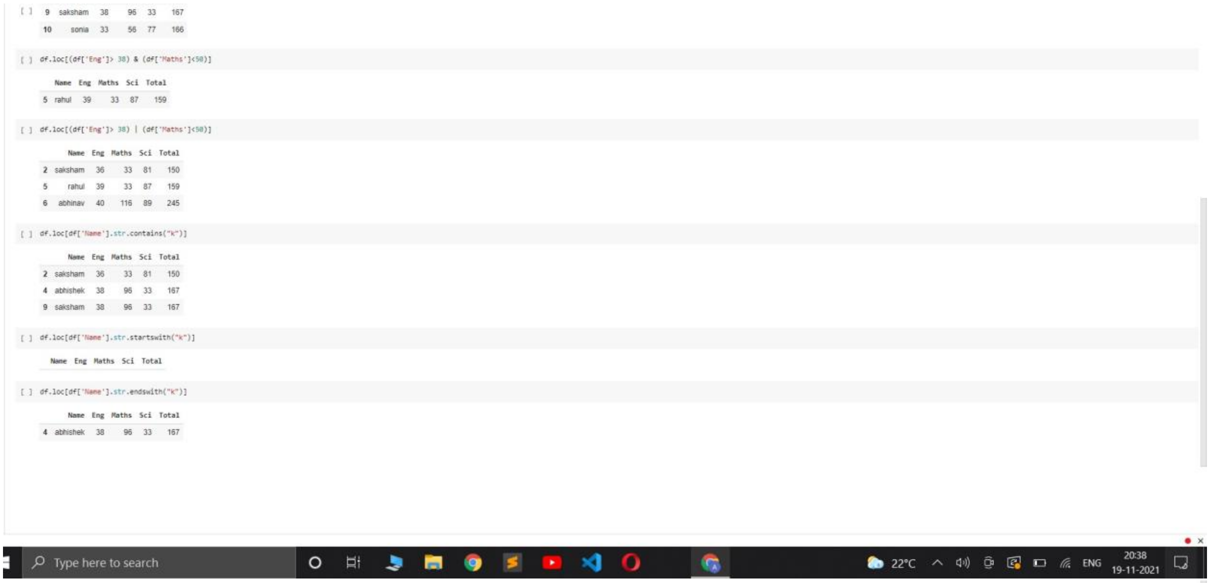
```
5     Shubham       38       xxxxx@gmail.com
6   Siddharth       12       xxxxx@gmail.com
7    Satyapal       45       xxxxx@gmail.com
8        Amit       34       xxxxx@gmail.com
9       Rahul       36      xxxxxx@gmail.com
10    Praveen       54  xxxxxxxxxx@gmail.com
11       Amit       23  xxxxxxxxxx@gmail.com
```

```
# import module
import pandas as pd

# initializing Data
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']}

# creating dataframe
df = pd.DataFrame(student_data)

# Here df.duplicated() list duplicate Entries in ROllno.
# So that ~(NOT) is placed in order to get non duplicate values.
non_duplicate = df[~df.duplicated('Roll_no')]

# printing non-duplicate values
print(non_duplicate)
```

```
       Name  Roll_no             Email
0      Amit       23    xxxx@gmail.com
1   Praveen       54  xxxxxx@gmail.com
2   Jagroop       29  xxxxxx@gmail.com
3     Rahul       36      xx@gmail.com
4    Vishal       59    xxxx@gmail.com
5     Suraj       38   xxxxx@gmail.com
6    Rishab       12   xxxxx@gmail.com
7  Satyapal       45   xxxxx@gmail.com
8      Amit       34   xxxxx@gmail.com
```

DATA FILTERING:

Data Filtering is one of the most frequent data manipulation operation. It is similar to WHERE clause in SQL or you must have used filter in MS Excel for selecting specific rows based on some conditions. In terms of speed, python has an efficient way to perform filtering and aggregation.



9	saksham	38	96	33	167
10	sonia	33	56	77	166

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

	Name	Eng	Maths	Sci	Total
5	rahul	39	33	87	159

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

	Name	Eng	Maths	Sci	Total
2	saksham	36	33	81	150
5	rahul	39	33	87	159
6	abhinav	40	116	89	245

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

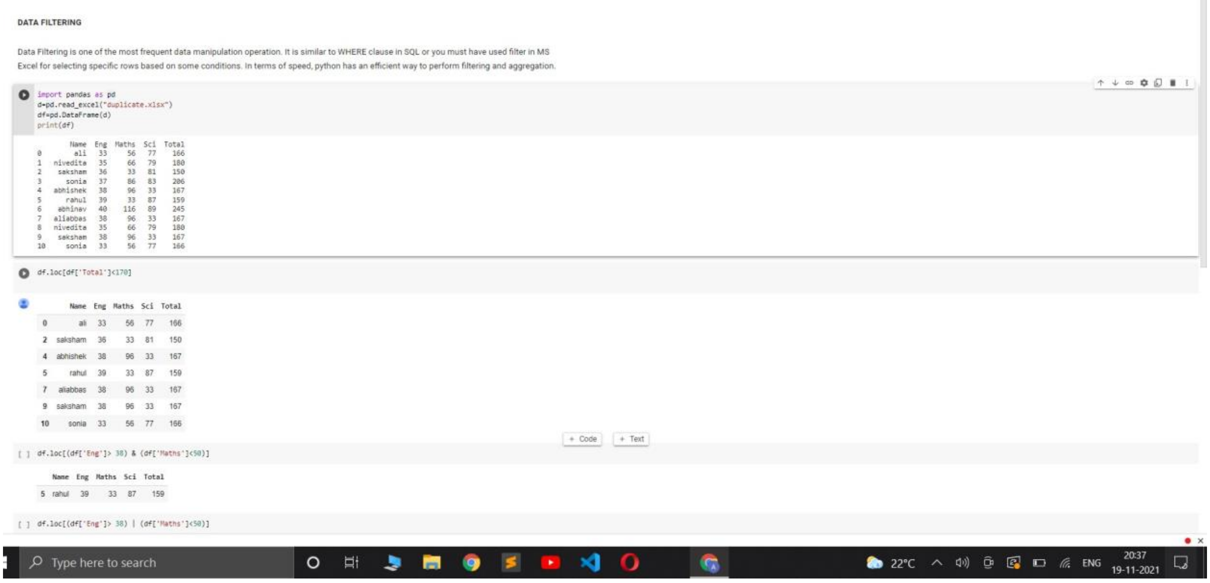
	Name	Eng	Maths	Sci	Total
2	saksham	36	33	81	150
4	abhishek	38	96	33	167
9	saksham	38	96	33	167

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

Name	Eng	Maths	Sci	Total

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

	Name	Eng	Maths	Sci	Total
4	abhishek	38	96	33	167



DATA FILTERING

Data Filtering is one of the most frequent data manipulation operation. It is similar to WHERE clause in SQL or you must have used filter in MS Excel for selecting specific rows based on some conditions. In terms of speed, python has an efficient way to perform filtering and aggregation.

```
import pandas as pd
d=pd.read_excel("duplicate.xlsx")
df=pd.DataFrame(d)
print(df)
```

```
        Name  Eng  Maths  Sci  Total
0        ali   33     56   77    166
1   nivedita   35     66   79    180
2    saksham   36     33   81    150
3      sonia   37     86   83    206
4   abhishek   38     96   33    167
5      rahul   39     33   87    159
6    abhinav   40    116   89    245
7   aliabbas   38     96   33    167
8   nivedita   35     66   79    180
9    saksham   38     96   33    167
10     sonia   33     56   77    166
```

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

	Name	Eng	Maths	Sci	Total
0	ali	33	56	77	166
2	saksham	36	33	81	150
4	abhishek	38	96	33	167
5	rahul	39	33	87	159
7	aliabbas	38	96	33	167
9	saksham	38	96	33	167
10	sonia	33	56	77	166

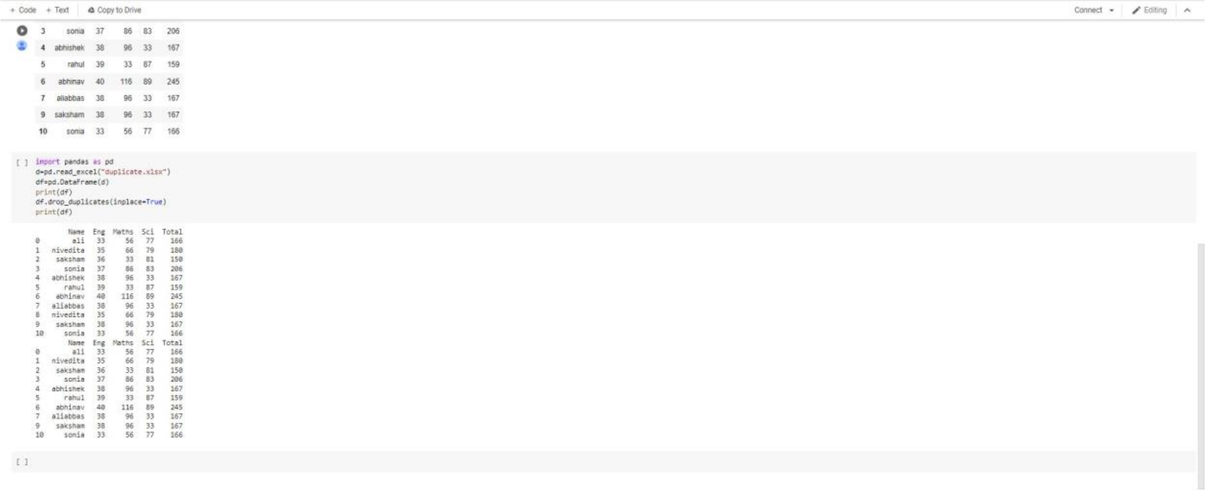
```
df.loc[(df['Eng']> 38) & (df['Maths']<50)]
```

	Name	Eng	Maths	Sci	Total
5	rahul	39	33	87	159

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

PANDAS DUPLICATION:

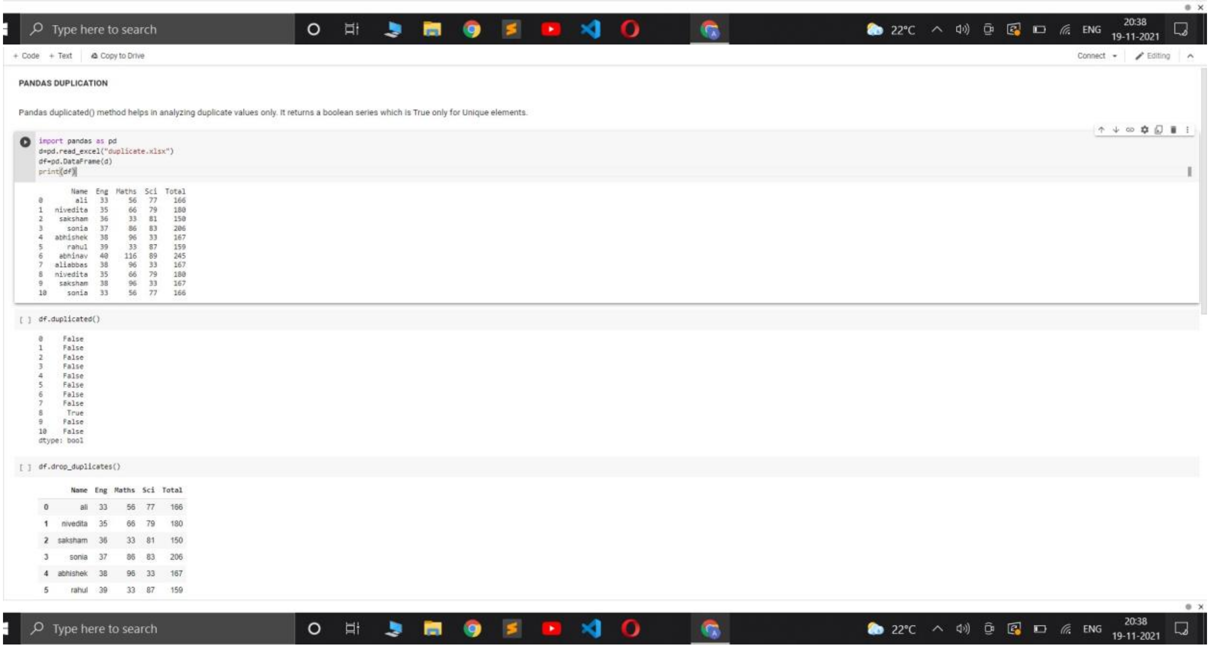
Pandas duplicated() method helps in analyzing duplicate values only. It returns a boolean series which is True only for Unique elements.



```
3      sonia  37   86  83   206
4   abhishek  38   96  33   167
5      rahul  39   33  87   159
6    abhinav  40  116  89   245
7   aliabbas  38   96  33   167
9    saksham  38   96  33   167
10     sonia  33   56  77   166
```

```
import pandas as pd
d=pd.read_excel("duplicate.xlsx")
df=pd.DataFrame(d)
print(df)
df.drop_duplicates(inplace=True)
print(df)
```

```
        Name  Eng  Maths  Sci  Total
0        ali   33     56   77    166
1   nivedita   35     66   79    180
2    saksham   36     33   81    150
3      sonia   37     86   83    206
4   abhishek   38     96   33    167
5      rahul   39     33   87    159
6    abhinav   40    116   89    245
7   aliabbas   38     96   33    167
8   nivedita   35     66   79    180
9    saksham   38     96   33    167
10     sonia   33     56   77    166
        Name  Eng  Maths  Sci  Total
0        ali   33     56   77    166
1   nivedita   35     66   79    180
2    saksham   36     33   81    150
3      sonia   37     86   83    206
4   abhishek   38     96   33    167
5      rahul   39     33   87    159
6    abhinav   40    116   89    245
7   aliabbas   38     96   33    167
9    saksham   38     96   33    167
10     sonia   33     56   77    166
```



PANDAS DUPLICATION

Pandas duplicated() method helps in analyzing duplicate values only. It returns a boolean series which is True only for Unique elements.

```
import pandas as pd
d=pd.read_excel("duplicate.xlsx")
df=pd.DataFrame(d)
print(df)
```

```
        Name  Eng  Maths  Sci  Total
0        ali   33     56   77    166
1   nivedita   35     66   79    180
2    saksham   36     33   81    150
3      sonia   37     86   83    206
4   abhishek   38     96   33    167
5      rahul   39     33   87    159
6    abhinav   40    116   89    245
7   aliabbas   38     96   33    167
8   nivedita   35     66   79    180
9    saksham   38     96   33    167
10     sonia   33     56   77    166
```

```
df.duplicated()
```

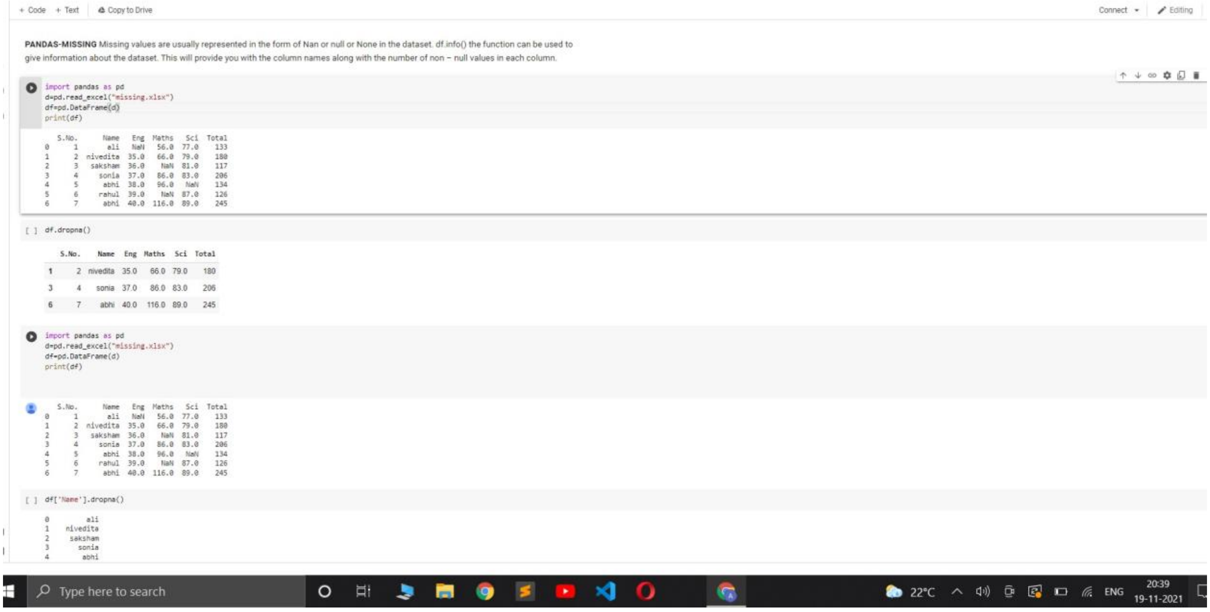
```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8      True
9     False
10    False
dtype: bool
```

```
df.drop_duplicates()
```

```
       Name  Eng  Maths  Sci  Total
0       ali   33     56   77    166
1  nivedita   35     66   79    180
2   saksham   36     33   81    150
3     sonia   37     86   83    206
4  abhishek   38     96   33    167
5     rahul   39     33   87    159
```

PANDAS-MISSING:

Missing values are usually represented in the form of Nan or null or None in the dataset. df.info() the function can be used to give information about the dataset. This will provide you with the column names along with the number of non – null values in each column.



PANDAS-MISSING Missing values are usually represented in the form of Nan or null or None in the dataset. df.info() the function can be used to give information about the dataset. This will provide you with the column names along with the number of non – null values in each column.

```
import pandas as pd
d=pd.read_excel("missing.xlsx")
df=pd.DataFrame(d)
print(df)
```

```
   S.No.      Name   Eng  Maths   Sci  Total
0      1       ali   NaN   56.0  77.0    133
1      2  nivedita  35.0   66.0  79.0    180
2      3   saksham  36.0    NaN  81.0    117
3      4     sonia  37.0   86.0  83.0    206
4      5      abhi  38.0   96.0   NaN    134
5      6     rahul  39.0    NaN  87.0    126
6      7      abhi  40.0  116.0  89.0    245
```

```
[ ] df.dropna()
```

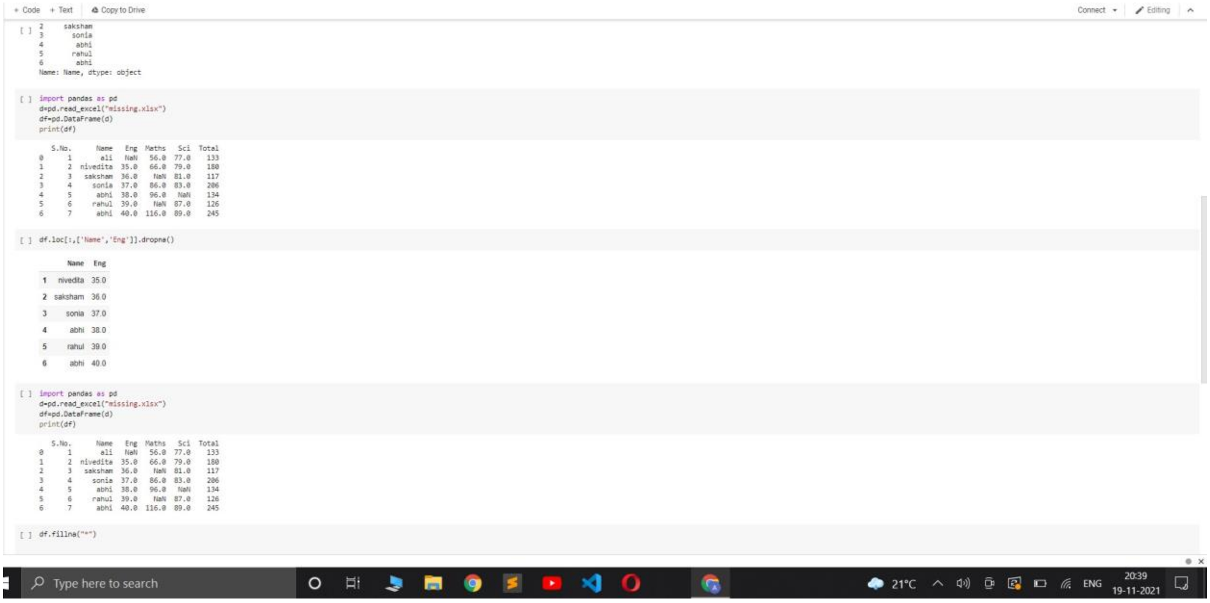
	S.No.	Name	Eng	Maths	Sci	Total
1	2	nivedita	35.0	66.0	79.0	180
3	4	sonia	37.0	86.0	83.0	206
6	7	abhi	40.0	116.0	89.0	245

```
import pandas as pd
d=pd.read_excel("missing.xlsx")
df=pd.DataFrame(d)
print(df)
```

```
   S.No.      Name   Eng  Maths   Sci  Total
0      1       ali   NaN   56.0  77.0    133
1      2  nivedita  35.0   66.0  79.0    180
2      3   saksham  36.0    NaN  81.0    117
3      4     sonia  37.0   86.0  83.0    206
4      5      abhi  38.0   96.0   NaN    134
5      6     rahul  39.0    NaN  87.0    126
6      7      abhi  40.0  116.0  89.0    245
```

```
[ ] df['Name'].dropna()
```

```
0         ali
1    nivedita
2     saksham
3       sonia
4        abhi
```



```
[ ] 2     saksham
    3       sonia
    4        abhi
    5       rahul
    6        abhi
    Name: Name, dtype: object
```

```
[ ] import pandas as pd
    d=pd.read_excel("missing.xlsx")
    df=pd.DataFrame(d)
    print(df)
```

```
   S.No.      Name   Eng  Maths   Sci  Total
0      1       ali   NaN   56.0  77.0    133
1      2  nivedita  35.0   66.0  79.0    180
2      3   saksham  36.0    NaN  81.0    117
3      4     sonia  37.0   86.0  83.0    206
4      5      abhi  38.0   96.0   NaN    134
5      6     rahul  39.0    NaN  87.0    126
6      7      abhi  40.0  116.0  89.0    245
```

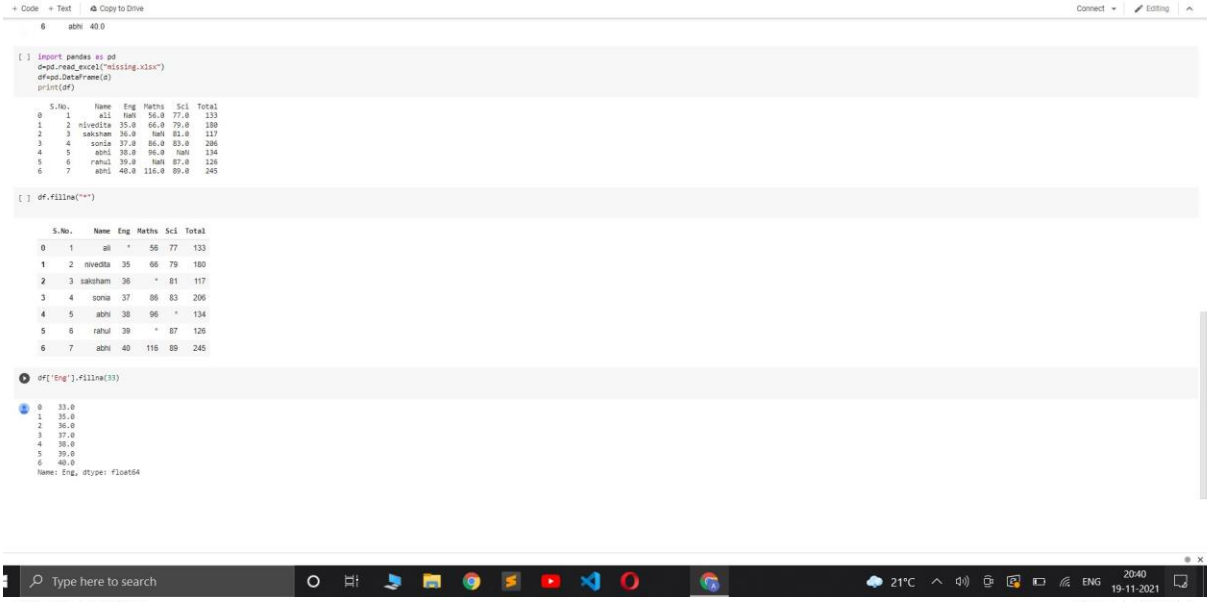
```
[ ] df.loc[:,['Name','Eng']].dropna()
```

	Name	Eng
1	nivedita	35.0
2	saksham	36.0
3	sonia	37.0
4	abhi	38.0
5	rahul	39.0
6	abhi	40.0

```
[ ] import pandas as pd
    d=pd.read_excel("missing.xlsx")
    df=pd.DataFrame(d)
    print(df)
```

```
   S.No.      Name   Eng  Maths   Sci  Total
0      1       ali   NaN   56.0  77.0    133
1      2  nivedita  35.0   66.0  79.0    180
2      3   saksham  36.0    NaN  81.0    117
3      4     sonia  37.0   86.0  83.0    206
4      5      abhi  38.0   96.0   NaN    134
5      6     rahul  39.0    NaN  87.0    126
6      7      abhi  40.0  116.0  89.0    245
```

```
[ ] df.fillna("*")
```



```
6       abhi  40.0
```

```
[ ] import pandas as pd
    d=pd.read_excel("missing.xlsx")
    df=pd.DataFrame(d)
    print(df)
```

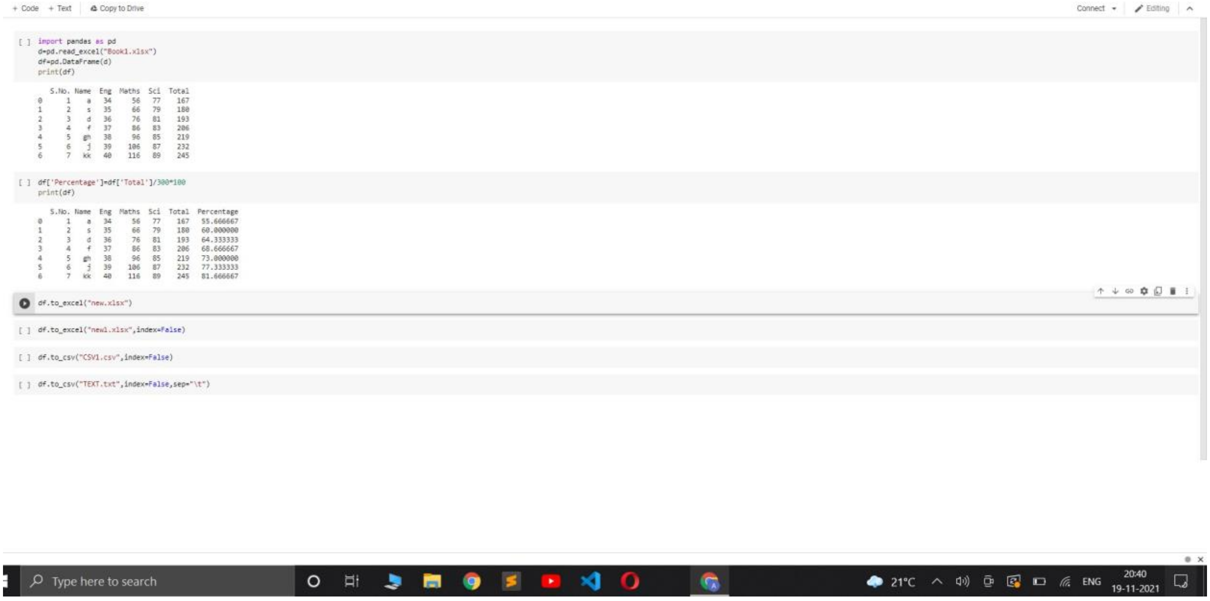
```
   S.No.      Name   Eng  Maths   Sci  Total
0      1       ali   NaN   56.0  77.0    133
1      2  nivedita  35.0   66.0  79.0    180
2      3   saksham  36.0    NaN  81.0    117
3      4     sonia  37.0   86.0  83.0    206
4      5      abhi  38.0   96.0   NaN    134
5      6     rahul  39.0    NaN  87.0    126
6      7      abhi  40.0  116.0  89.0    245
```

```
[ ] df.fillna("*")
```

	S.No.	Name	Eng	Maths	Sci	Total
0	1	ali	*	56	77	133
1	2	nivedita	35	66	79	180
2	3	saksham	36	*	81	117
3	4	sonia	37	86	83	206
4	5	abhi	38	96	*	134
5	6	rahul	39	*	87	126
6	7	abhi	40	116	89	245

```
df['Eng'].fillna(33)
```

```
0    33.0
1    35.0
2    36.0
3    37.0
4    38.0
5    39.0
6    40.0
Name: Eng, dtype: float64
```



```
[ ] import pandas as pd
    d=pd.read_excel("Book1.xlsx")
    df=pd.DataFrame(d)
    print(df)
```

```
   S.No. Name  Eng  Maths  Sci  Total
0      1    a   34     56   77    167
1      2    s   35     66   79    180
2      3    d   36     76   81    193
3      4    f   37     86   83    206
4      5   gh   38     96   85    219
5      6    j   39    106   87    232
6      7   kk   40    116   89    245
```

```
[ ] df['Percentage']=df['Total']/300*100
    print(df)
```

```
   S.No. Name  Eng  Maths  Sci  Total  Percentage
0      1    a   34     56   77    167   55.666667
1      2    s   35     66   79    180   60.000000
2      3    d   36     76   81    193   64.333333
3      4    f   37     86   83    206   68.666667
4      5   gh   38     96   85    219   73.000000
5      6    j   39    106   87    232   77.333333
6      7   kk   40    116   89    245   81.666667
```

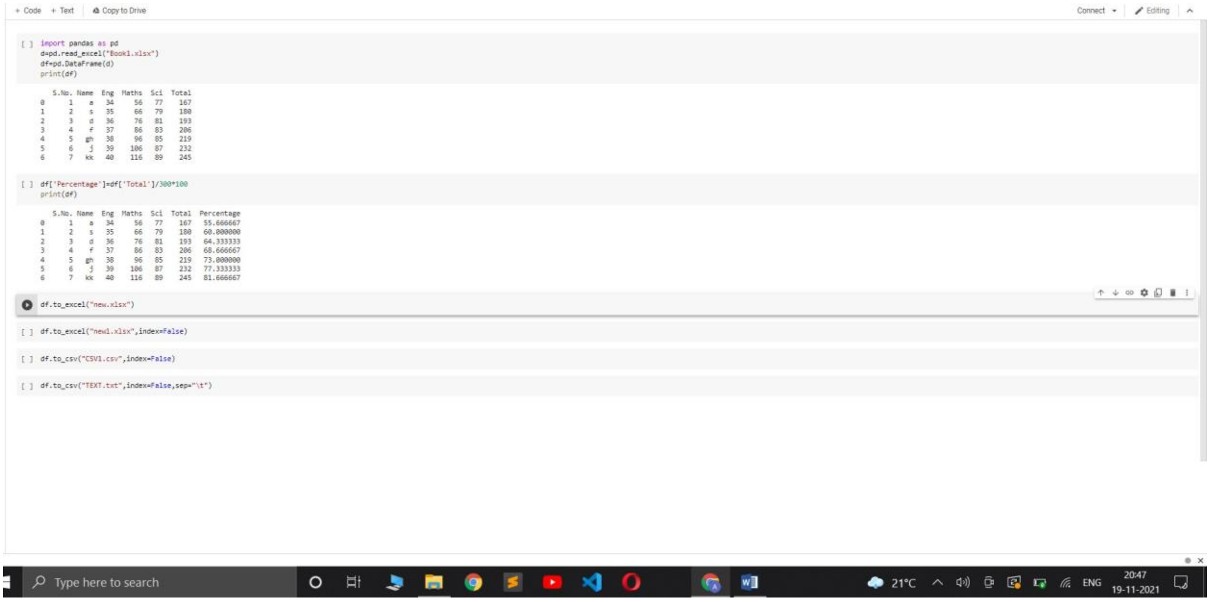
```
df.to_excel("new.xlsx")
```

```
[ ] df.to_excel("new1.xlsx",index=False)
```

```
[ ] df.to_csv("CSV1.csv",index=False)
```

```
[ ] df.to_csv("TEXT.txt",index=False,sep="\t")
```

PANDAS EXPORT:



```
[ ] import pandas as pd
    d=pd.read_excel("Book1.xlsx")
    df=pd.DataFrame(d)
    print(df)
```

```
   S.No. Name  Eng  Maths  Sci  Total
0      1    a   34     56   77    167
1      2    s   35     66   79    180
2      3    d   36     76   81    193
3      4    f   37     86   83    206
4      5   gh   38     96   85    219
5      6    j   39    106   87    232
6      7   kk   40    116   89    245
```

```
[ ] df['Percentage']=df['Total']/300*100
    print(df)
```

```
   S.No. Name  Eng  Maths  Sci  Total  Percentage
0      1    a   34     56   77    167   55.666667
1      2    s   35     66   79    180   60.000000
2      3    d   36     76   81    193   64.333333
3      4    f   37     86   83    206   68.666667
4      5   gh   38     96   85    219   73.000000
5      6    j   39    106   87    232   77.333333
6      7   kk   40    116   89    245   81.666667
```

```
df.to_excel("new.xlsx")
```

```
[ ] df.to_excel("new1.xlsx",index=False)
```

```
[ ] df.to_csv("CSV1.csv",index=False)
```

```
[ ] df.to_csv("TEXT.txt",index=False,sep="\t")
```