1. **In this question (1), we will use attached dataset as nba.csv.**

Find the output of the following sections:

i)Write a syntax to load given data

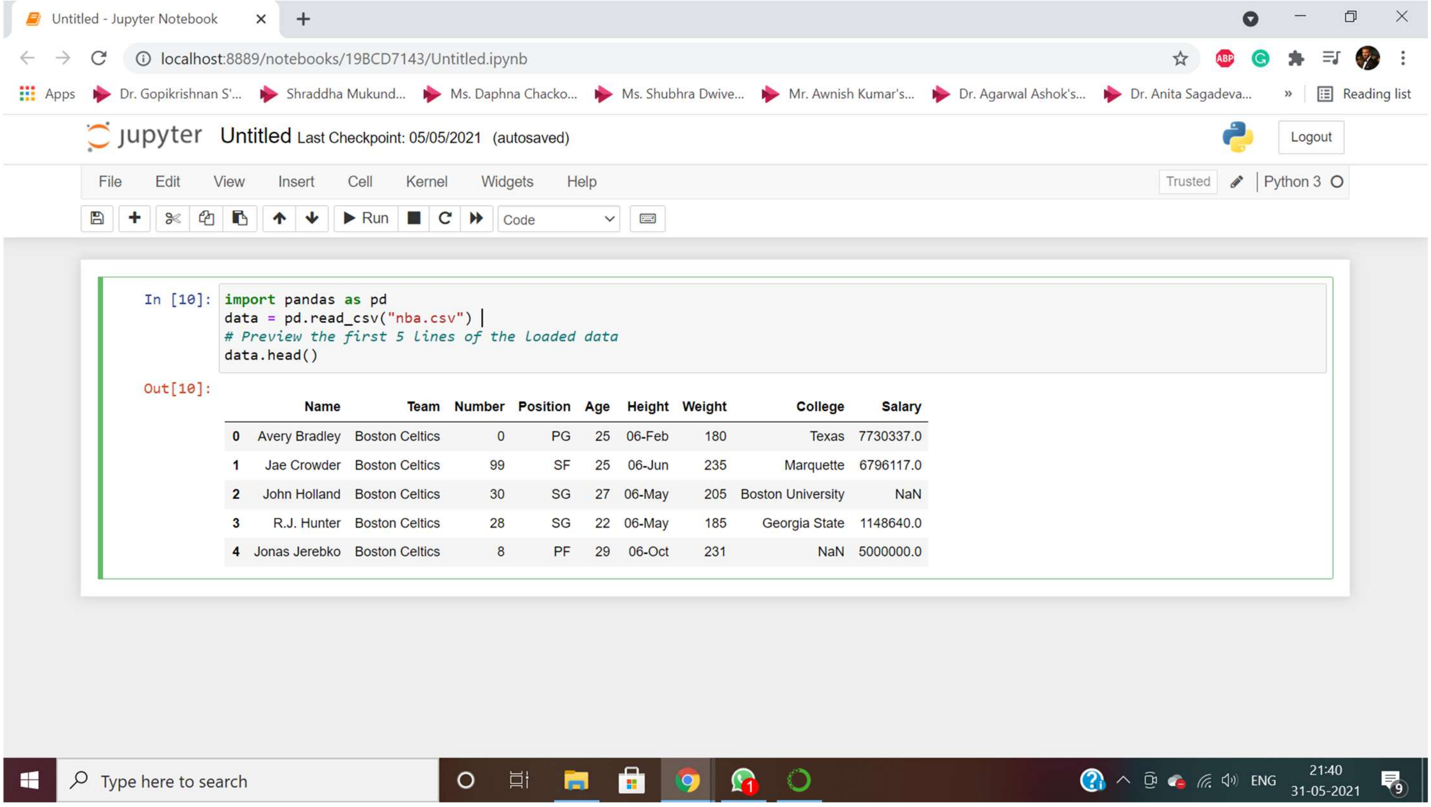
Answer:

import pandas as pd

data = pd.read_csv("nba.csv")

data.head()

<u>OUTPUT</u>



In [10]:

```
import pandas as pd
data = pd.read_csv("nba.csv")
# Preview the first 5 lines of the loaded data
data.head()
```

Out[10]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	06-Feb	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	06-Jun	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	06-May	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	06-May	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	06-Oct	231	NaN	5000000.0

ii)Summarize the data

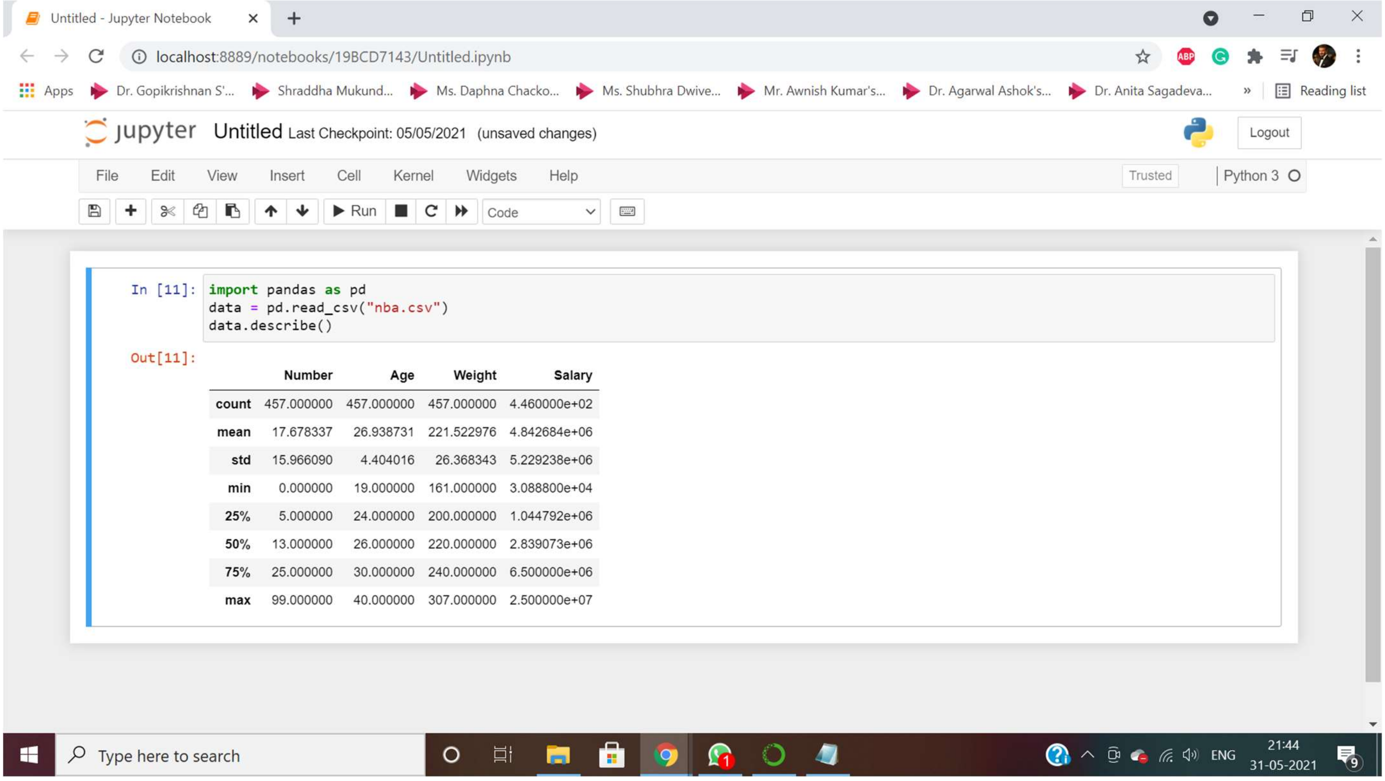
Answer:

import pandas as pd

data = pd.read_csv("nba.csv")

data.describe()

OUTPUT



```
In [11]: import pandas as pd
         data = pd.read_csv("nba.csv")
         data.describe()
```

Out[11]:

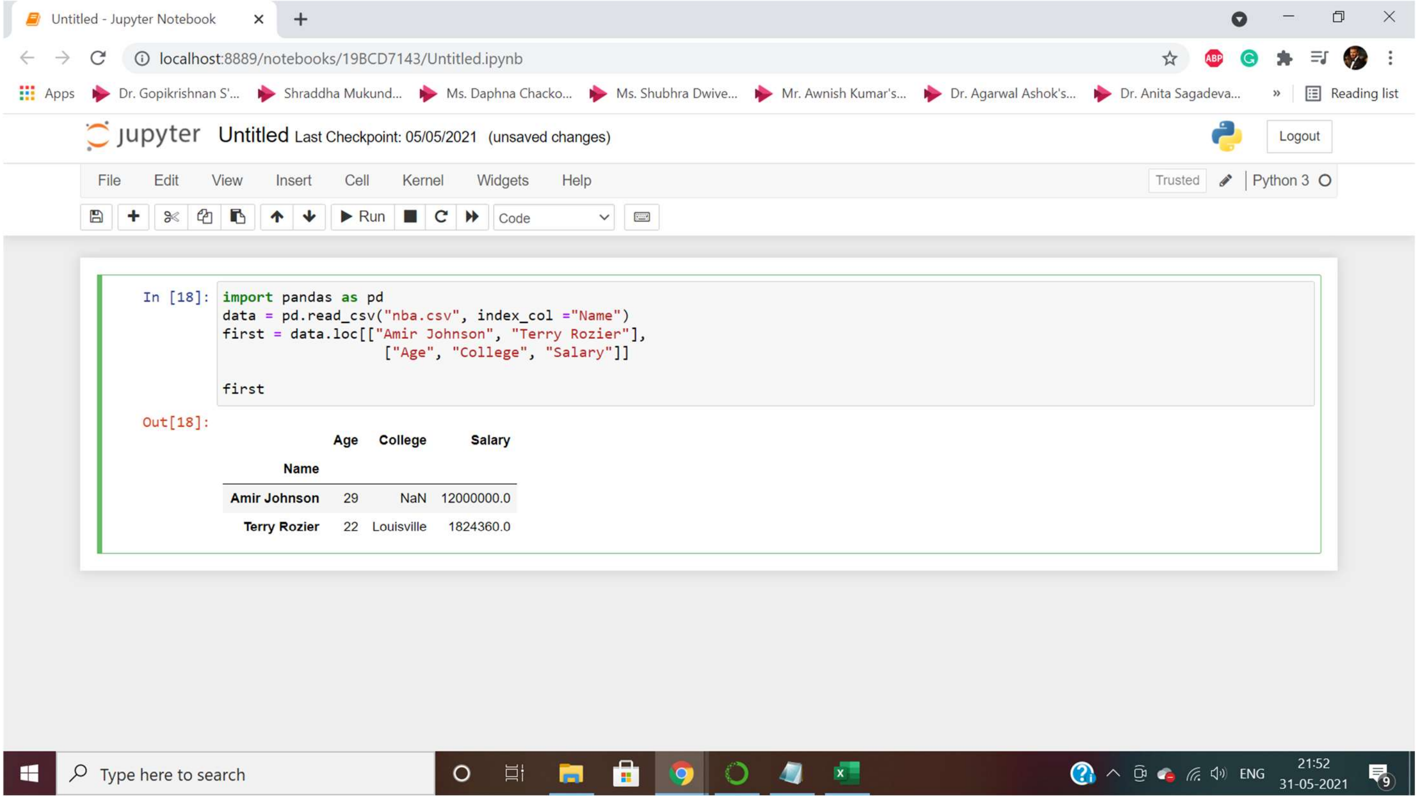
	Number	Age	Weight	Salary
count	457.000000	457.000000	457.000000	4.460000e+02
mean	17.678337	26.938731	221.522976	4.842684e+06
std	15.966090	4.404016	26.368343	5.229238e+06
min	0.000000	19.000000	161.000000	3.088800e+04
25%	5.000000	24.000000	200.000000	1.044792e+06
50%	13.000000	26.000000	220.000000	2.839073e+06
75%	25.000000	30.000000	240.000000	6.500000e+06
max	99.000000	40.000000	307.000000	2.500000e+07

iii) To select columns Age, College and Salary for only rows with a labels Amir Johnson and Terry Rozier.

Answer:

```
import pandas as pd
data = pd.read_csv("nba.csv", index_col ="Name")
first = data.loc[["Amir Johnson", "Terry Rozier"],
            ["Age", "College", "Salary"]]
first
```

OUTPUT


Untitled - Jupyter Notebook
localhost:8889/notebooks/19BCD7143/Untitled.ipynb
jupyter Untitled Last Checkpoint: 05/05/2021 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
ENG
21:52
31-05-2021

```
In [18]: import pandas as pd
         data = pd.read_csv("nba.csv", index_col ="Name")
         first = data.loc[["Amir Johnson", "Terry Rozier"],
                          ["Age", "College", "Salary"]]

         first
```

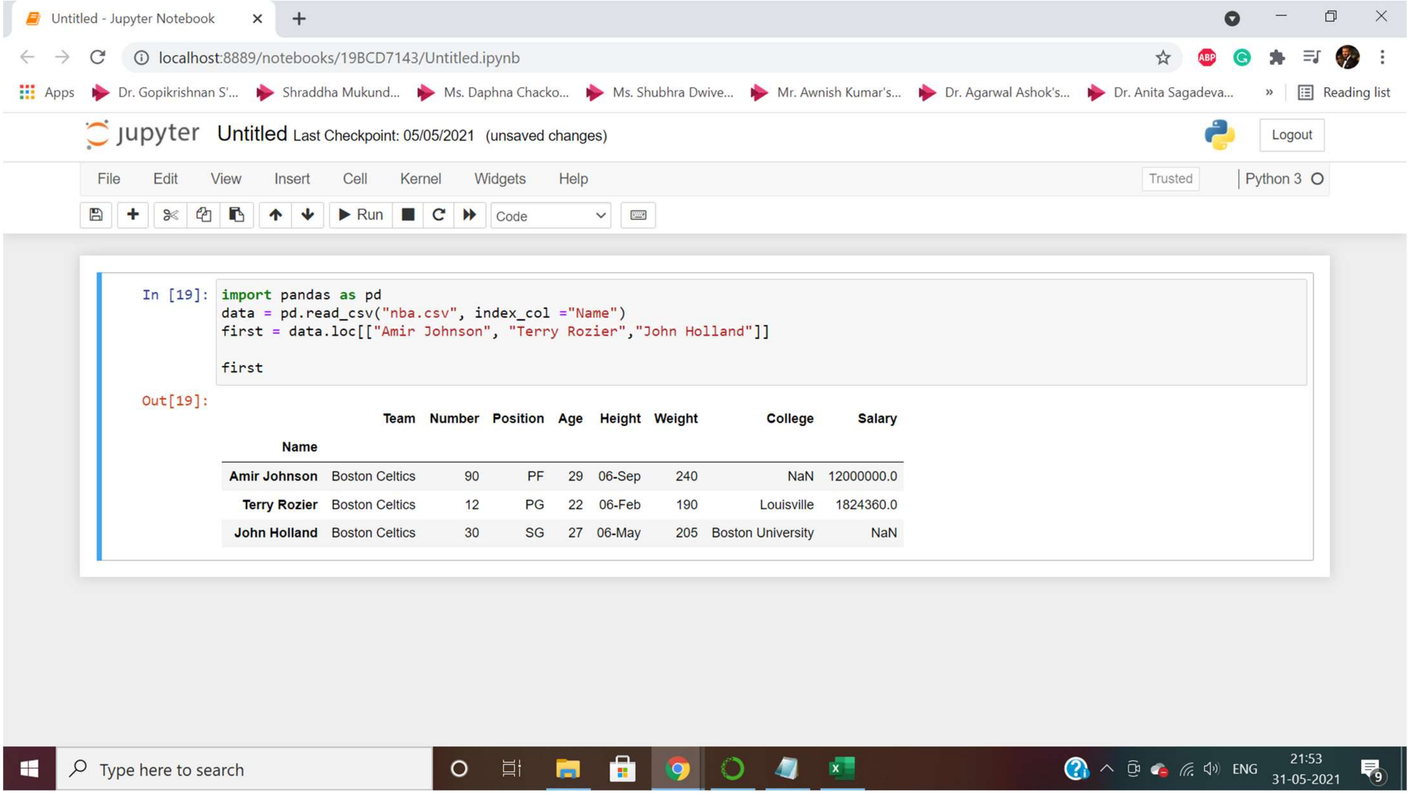
```
Out[18]:
                  Age   College      Salary
         Name
         Amir Johnson   29       NaN  12000000.0
          Terry Rozier   22  Louisville   1824360.0
```

iv) To select row Amir Jhonson, Terry Rozier and John Holland with all columns in a dataframe.

Answer:

```
import pandas as pd
data = pd.read_csv("nba.csv", index_col ="Name")
first = data.loc[["Amir Johnson", "Terry Rozier","John Holland"]]

first
```

OUTPUT



```
In [19]: import pandas as pd
         data = pd.read_csv("nba.csv", index_col ="Name")
         first = data.loc[["Amir Johnson", "Terry Rozier","John Holland"]]

         first
```

Out[19]:

	Team	Number	Position	Age	Height	Weight	College	Salary
Name								
Amir Johnson	Boston Celtics	90	PF	29	06-Sep	240	NaN	12000000.0
Terry Rozier	Boston Celtics	12	PG	22	06-Feb	190	Louisville	1824360.0
John Holland	Boston Celtics	30	SG	27	06-May	205	Boston University	NaN

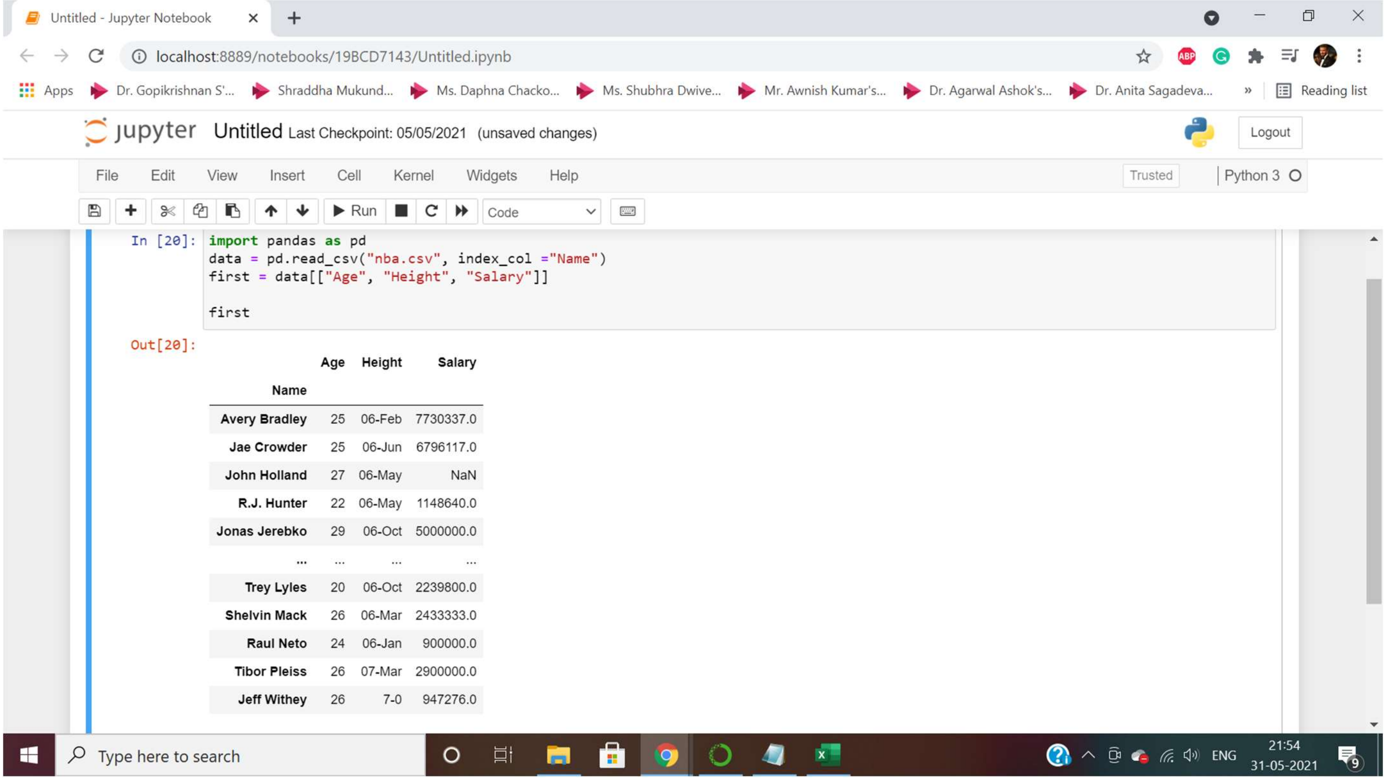
v) To select columns Age, Height and Salary with all rows in a dataframe.

Answer:

import pandas as pd
data = pd.read_csv("nba.csv", index_col ="Name")
first = data[["Age", "Height", "Salary"]]

first

OUTPUT


Untitled - Jupyter Notebook
localhost:8889/notebooks/19BCD7143/Untitled.ipynb
jupyter Untitled Last Checkpoint: 05/05/2021 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
ENG
21:54
31-05-2021

```
In [20]: import pandas as pd
         data = pd.read_csv("nba.csv", index_col ="Name")
         first = data[["Age", "Height", "Salary"]]

         first
```

Out[20]:

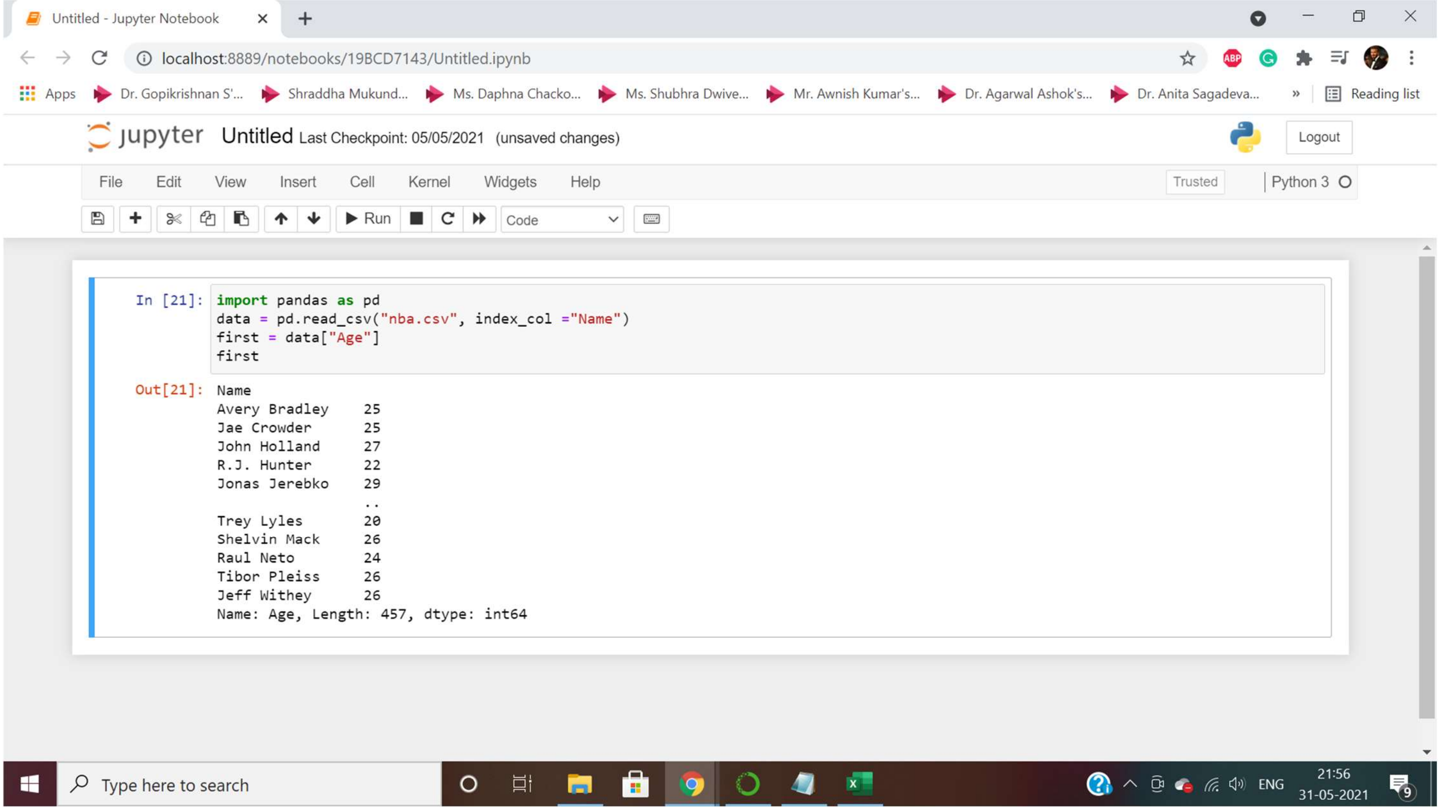
Name	Age	Height	Salary
Avery Bradley	25	06-Feb	7730337.0
Jae Crowder	25	06-Jun	6796117.0
John Holland	27	06-May	NaN
R.J. Hunter	22	06-May	1148640.0
Jonas Jerebko	29	06-Oct	5000000.0
...	...	...	...
Trey Lyles	20	06-Oct	2239800.0
Shelvin Mack	26	06-Mar	2433333.0
Raul Neto	24	06-Jan	900000.0
Tibor Pleiss	26	07-Mar	2900000.0
Jeff Withey	26	7-0	947276.0

vi) To select column Age, with all rows in a dataframe with label is Name.

Asnwer:

```
import pandas as pd
data = pd.read_csv("nba.csv", index_col ="Name")
first = data["Age"]
first
```

OUTPUT



```
In [21]: import pandas as pd
         data = pd.read_csv("nba.csv", index_col ="Name")
         first = data["Age"]
         first
```

```
Out[21]: Name
         Avery Bradley    25
         Jae Crowder      25
         John Holland     27
         R.J. Hunter      22
         Jonas Jerebko    29
                          ..
         Trey Lyles       20
         Shelvin Mack     26
         Raul Neto        24
         Tibor Pleiss     26
         Jeff Withey      26
         Name: Age, Length: 457, dtype: int64
```

2.Create a dataframe as df = pd.DataFrame(np.random.randn(6, 3), index = ['a','b','c','d','e','f'], columns = ['A', 'B', 'C'])

Find the output of the following syntax:

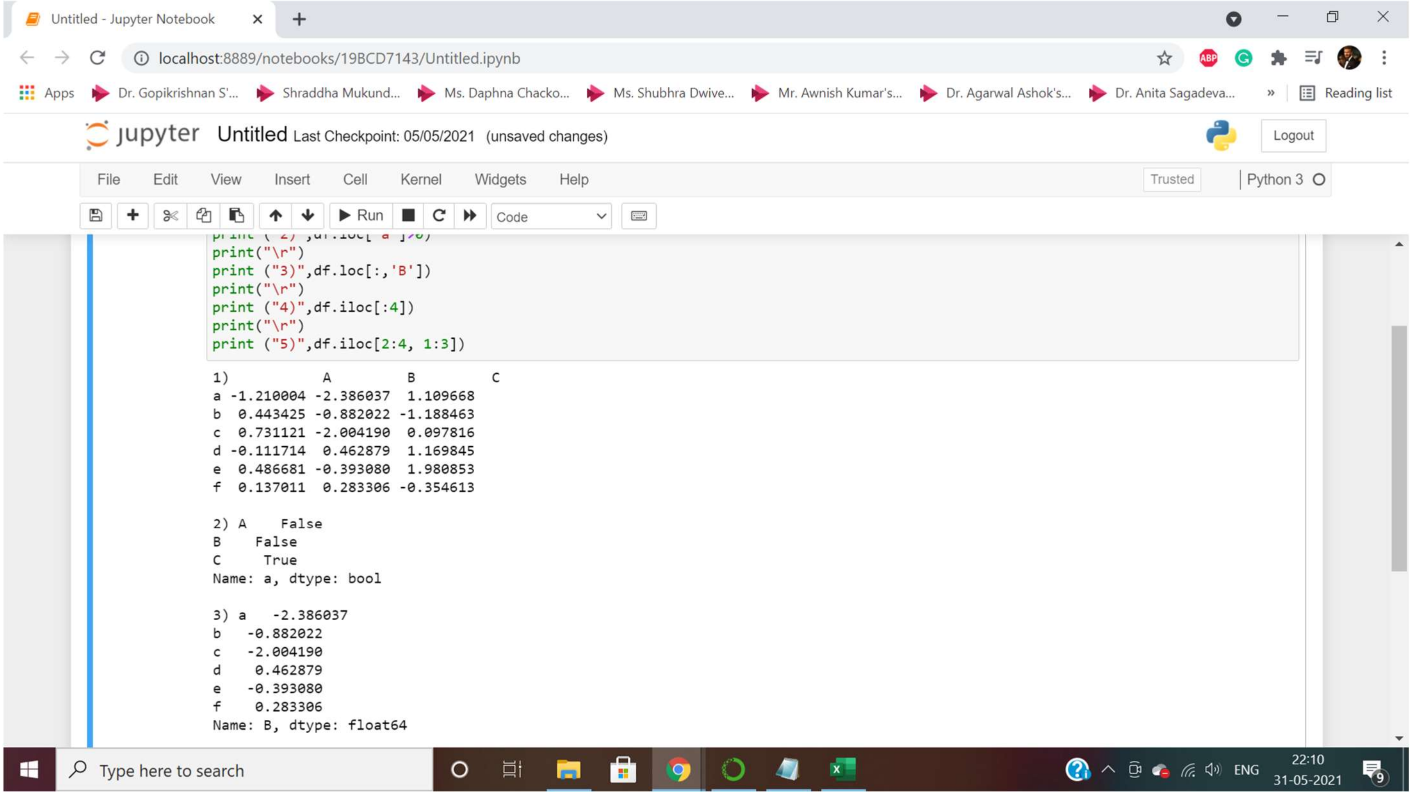
- **print (df.loc['a':'f'])**
- **print (df.loc['a']>0)**
- **print df.loc[:,'B']**
- **print (df1.iloc[:4])**
- **print (df1.iloc[2:4, 1:3])**

Answer:

import pandas as pd

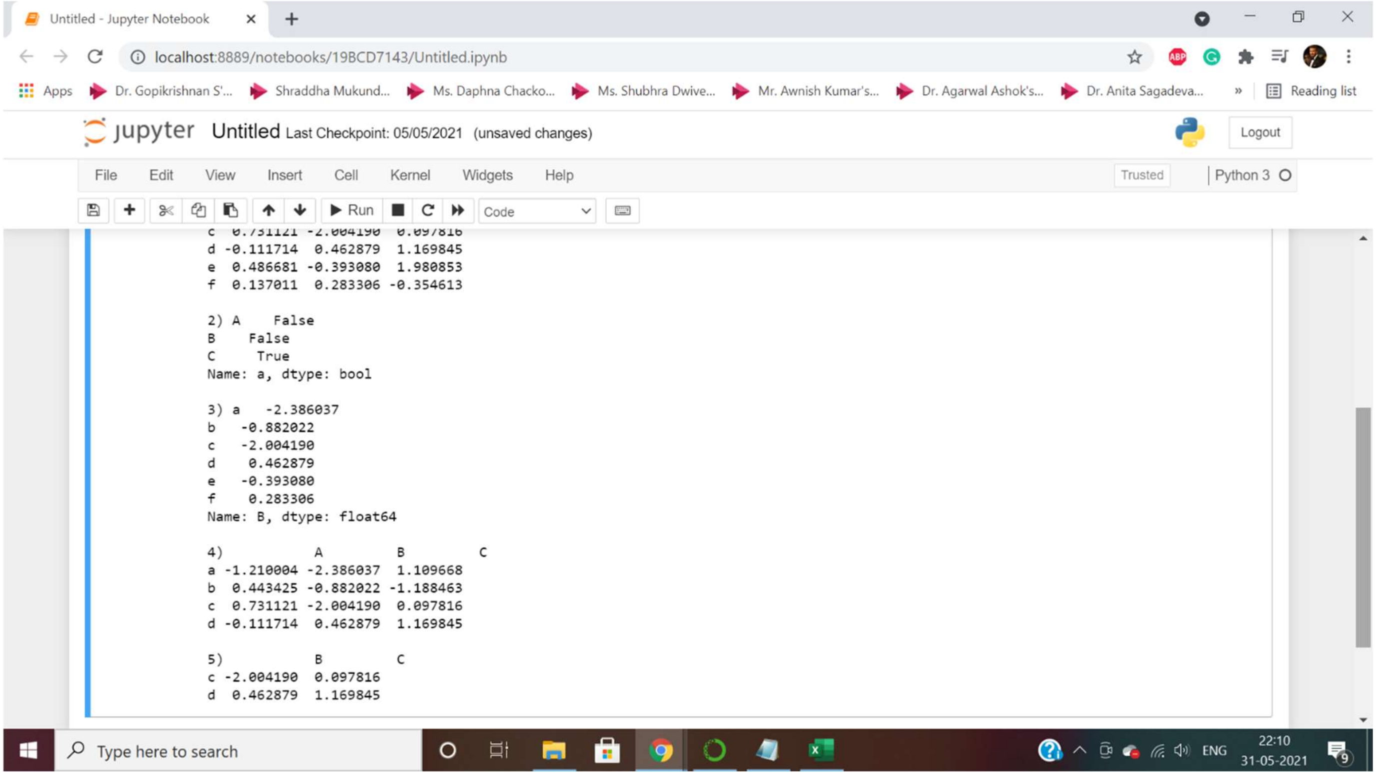
```
import numpy as np
df =pd.DataFrame(np.random.randn(6, 3), index =
['a','b','c','d','e','f'],columns = ['A', 'B', 'C'])
print ("1)",df.loc['a':'f'])
print("\r")
print ("2)",df.loc['a']>0)
print("\r")
print ("3)",df.loc[:,'B'])
print("\r")
print ("4)",df.iloc[:4])
print("\r")
print ("5)",df.iloc[2:4, 1:3])
```

OUTPUT



```
print ("2)",df.loc['a']>0)
print("\r")
print ("3)",df.loc[:,'B'])
print("\r")
print ("4)",df.iloc[:4])
print("\r")
print ("5)",df.iloc[2:4, 1:3])
```

```
1)           A         B         C
a -1.210004 -2.386037  1.109668
b  0.443425 -0.882022 -1.188463
c  0.731121 -2.004190  0.097816
d -0.111714  0.462879  1.169845
e  0.486681 -0.393080  1.980853
f  0.137011  0.283306 -0.354613

2) A    False
B    False
C     True
Name: a, dtype: bool

3) a   -2.386037
b   -0.882022
c   -2.004190
d    0.462879
e   -0.393080
f    0.283306
Name: B, dtype: float64
```



Untitled - Jupyter Notebook
localhost:8889/notebooks/19BCD7143/Untitled.ipynb
jupyter Untitled Last Checkpoint: 05/05/2021 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
22:10
31-05-2021

```
c  0.731121 -2.004190  0.097816
d -0.111714  0.462879  1.169845
e  0.486681 -0.393080  1.980853
f  0.137011  0.283306 -0.354613

2) A    False
B    False
C     True
Name: a, dtype: bool

3) a   -2.386037
b   -0.882022
c   -2.004190
d    0.462879
e   -0.393080
f    0.283306
Name: B, dtype: float64

4)          A         B         C
a -1.210004 -2.386037  1.109668
b  0.443425 -0.882022 -1.188463
c  0.731121 -2.004190  0.097816
d -0.111714  0.462879  1.169845

5)          B         C
c -2.004190  0.097816
d  0.462879  1.169845
```

3.Create a dataframe as df5 = pd.DataFrame({'a': ['one', 'one', 'two', 'two', 'two'], 'b': ['x', 'y', 'x', 'y', 'x'], 'c': np.random.randn(5)})

Find the output of the following syntax:

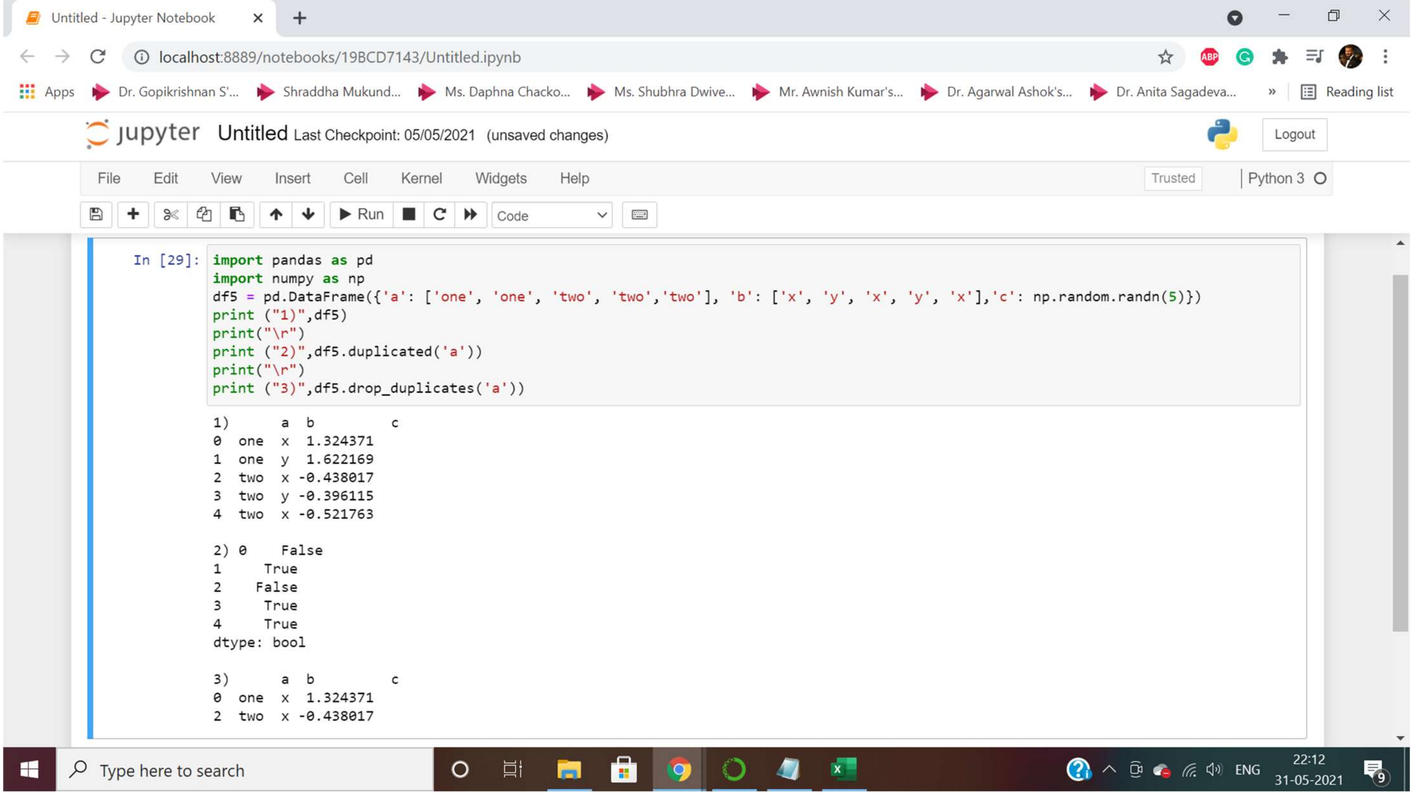
- **Print df5**
- **Print df5.duplicated('a')**
- **Print df5.drop_duplicates('a')**

Answer:

```
import pandas as pd
import numpy as np
df5 = pd.DataFrame({'a': ['one', 'one', 'two', 'two','two'], 'b': ['x', 'y', 'x',
'y', 'x'],'c': np.random.randn(5)})
print ("1)",df5)
print("\r")
print ("2)",df5.duplicated('a'))
print("\r")
```

print ("3)",df5.drop_duplicates('a'))

<u>OUTPUT</u>



```
In [29]: import pandas as pd
         import numpy as np
         df5 = pd.DataFrame({'a': ['one', 'one', 'two', 'two','two'], 'b': ['x', 'y', 'x', 'y', 'x'],'c': np.random.randn(5)})
         print ("1)",df5)
         print("\r")
         print ("2)",df5.duplicated('a'))
         print("\r")
         print ("3)",df5.drop_duplicates('a'))
```

```
1)      a  b         c
0  one  x  1.324371
1  one  y  1.622169
2  two  x -0.438017
3  two  y -0.396115
4  two  x -0.521763

2) 0    False
1     True
2    False
3     True
4     True
dtype: bool

3)      a  b         c
0  one  x  1.324371
2  two  x -0.438017
```

4. Create the following series data = pd.Series(['a', 'b', 'c'], index=[1, 3, 5])

Find the output of the following syntax:

· Print data · Print data[1]

· Print data[1:3]

· Print data.loc[1]

· Print data.loc[1:3]

· Print data.iloc[1:3]

Answer:

import pandas as pd

import numpy as np

data = pd.Series(['a', 'b', 'c'], index=[1, 3, 5])

print ("1)",data)

print("\r")

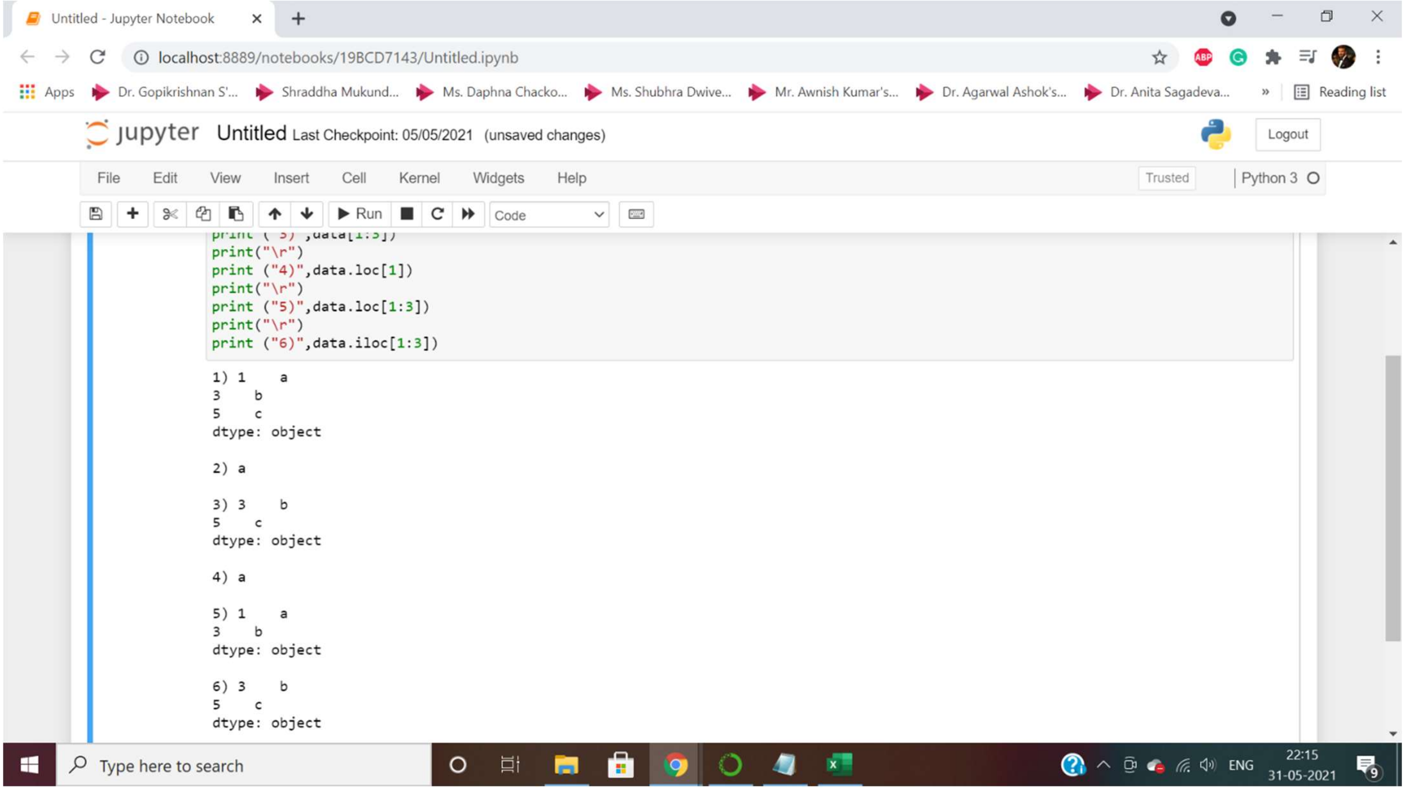
print ("2)",data[1])

print("\r")

print ("3)",data[1:3])

```
print("\r")
print ("4)",data.loc[1])
print("\r")
print ("5)",data.loc[1:3])
print("\r")
print ("6)",data.iloc[1:3])
```

OUTPUT



```
print ("3)",data[1:3])
print("\r")
print ("4)",data.loc[1])
print("\r")
print ("5)",data.loc[1:3])
print("\r")
print ("6)",data.iloc[1:3])
```

```
1) 1    a
3    b
5    c
dtype: object

2) a

3) 3    b
5    c
dtype: object

4) a

5) 1    a
3    b
dtype: object

6) 3    b
5    c
dtype: object
```

5. Create the following series area = pd.Series({'California': 423967, 'Texas': 695662, 'New York': 141297, 'Florida': 170312, 'Illinois': 149995}) pop = pd.Series({'California': 38332521, 'Texas': 26448193, 'New York': 19651127, 'Florida': 19552860, 'Illinois': 12882135})

Find the output of the following syntax:

· Create dataframe by following syntax as: data = pd.DataFrame({'area':area, 'pop':pop})

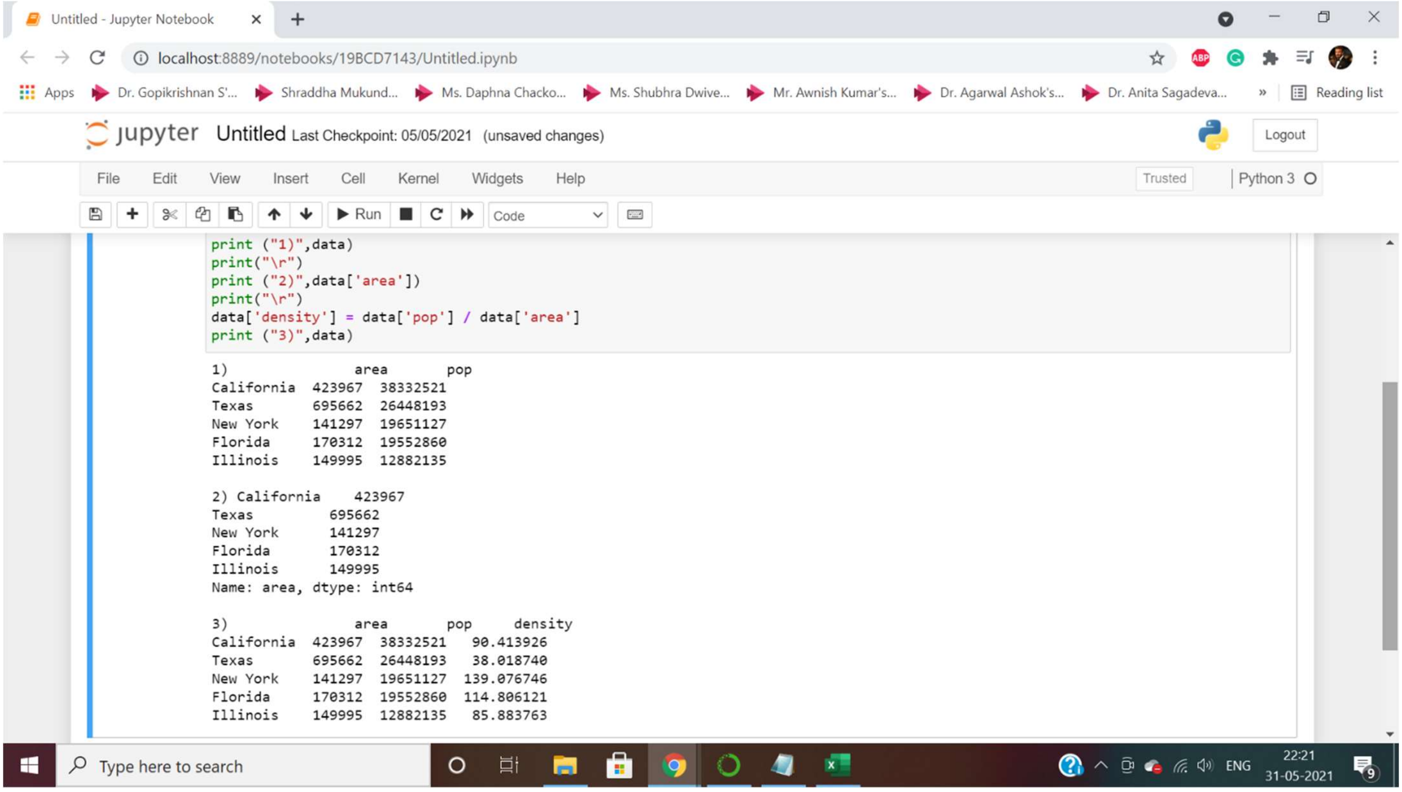
· Print data · Print data['area']

· Print data['density'] = data['pop'] / data['area']

Answer:

```
import pandas as pd
import numpy as np
area = pd.Series({'California': 423967, 'Texas': 695662,
'New York': 141297, 'Florida': 170312,
'Illinois': 149995})
pop = pd.Series({'California': 38332521, 'Texas': 26448193,
'New York': 19651127, 'Florida': 19552860,
'Illinois': 12882135})

data =pd.DataFrame({'area':area, 'pop':pop})

print ("1)",data)
print("\r")
print ("2)",data['area'])
print("\r")
data['density'] = data['pop'] / data['area']
print ("3)",data)
```

OUTPUT



```
print ("1)",data)
print("\r")
print ("2)",data['area'])
print("\r")
data['density'] = data['pop'] / data['area']
print ("3)",data)
```

```
1)               area       pop
California  423967  38332521
Texas       695662  26448193
New York    141297  19651127
Florida     170312  19552860
Illinois    149995  12882135

2) California    423967
Texas         695662
New York      141297
Florida       170312
Illinois      149995
Name: area, dtype: int64

3)               area       pop     density
California  423967  38332521   90.413926
Texas       695662  26448193   38.018740
New York    141297  19651127  139.076746
Florida     170312  19552860  114.806121
Illinois    149995  12882135   85.883763
```

6. . Create the following series Create as index = [('California', 2000), ('California', 2010), ('New York', 2000), ('New York', 2010), ('Texas', 2000), ('Texas', 2010)] populations = [33871648, 37253956, 18976457, 19378102, 20851820, 25145561] pop = pd.Series(populations, index=index)

Find the output of the following syntax:

· Print pop

· Print pop[('California', 2010):('Texas', 2000)]

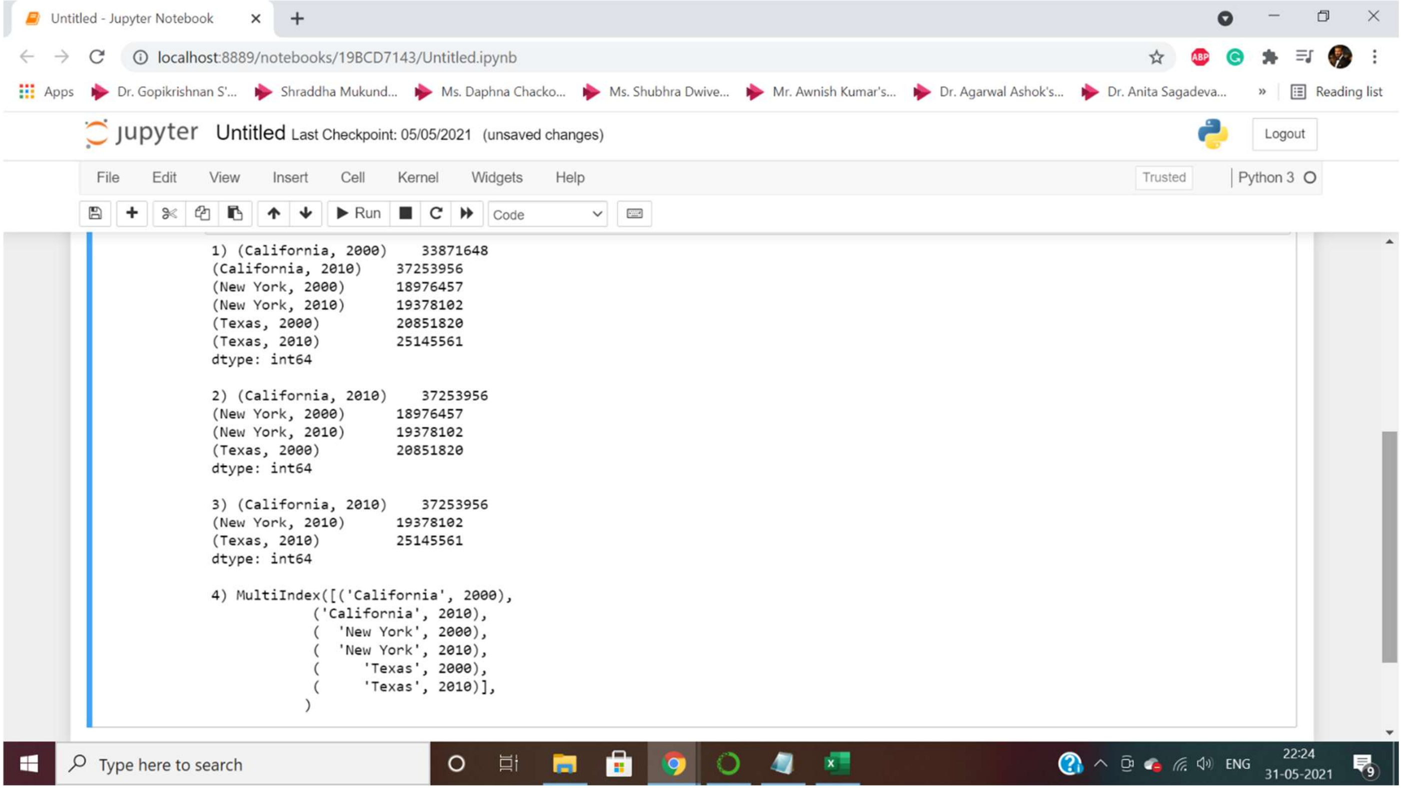
· Print pop[[i for i in pop.index if i[1] == 2010]]

· Print index = pd.MultiIndex.from_tuples(index)

Answer:

```
import pandas as pd
import numpy as np
index = [('California', 2000), ('California', 2010),
('New York', 2000), ('New York', 2010),
('Texas', 2000), ('Texas', 2010)]
populations = [33871648, 37253956,
18976457, 19378102,
20851820, 25145561]
pop = pd.Series(populations, index=index)

print ("1)",pop)
print("\r")
print ("2)",pop[('California', 2010):('Texas', 2000)])
print("\r")
print ("3)",pop[[i for i in pop.index if i[1] == 2010]])
print("\r")
index = pd.MultiIndex.from_tuples(index)
print ("4)",index)
```

OUTPUT



Untitled - Jupyter Notebook
localhost:8889/notebooks/19BCD7143/Untitled.ipynb
jupyter Untitled Last Checkpoint: 05/05/2021 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
ENG
22:24
31-05-2021

```
1) (California, 2000)    33871648
(California, 2010)    37253956
(New York, 2000)      18976457
(New York, 2010)      19378102
(Texas, 2000)         20851820
(Texas, 2010)         25145561
dtype: int64

2) (California, 2010)    37253956
(New York, 2000)      18976457
(New York, 2010)      19378102
(Texas, 2000)         20851820
dtype: int64

3) (California, 2010)    37253956
(New York, 2010)      19378102
(Texas, 2010)         25145561
dtype: int64

4) MultiIndex([('California', 2000),
            ('California', 2010),
            (  'New York', 2000),
            (  'New York', 2010),
            (     'Texas', 2000),
            (     'Texas', 2010)],
           )
```

7. Create a dataframe as df = pd.DataFrame([[1, np.nan, 2], [2, 3, 5], [np.nan, 4, 6]])

Find the output of the following syntax:

· Print df · Print df.dropna()

· Print df.dropna(axis='columns')

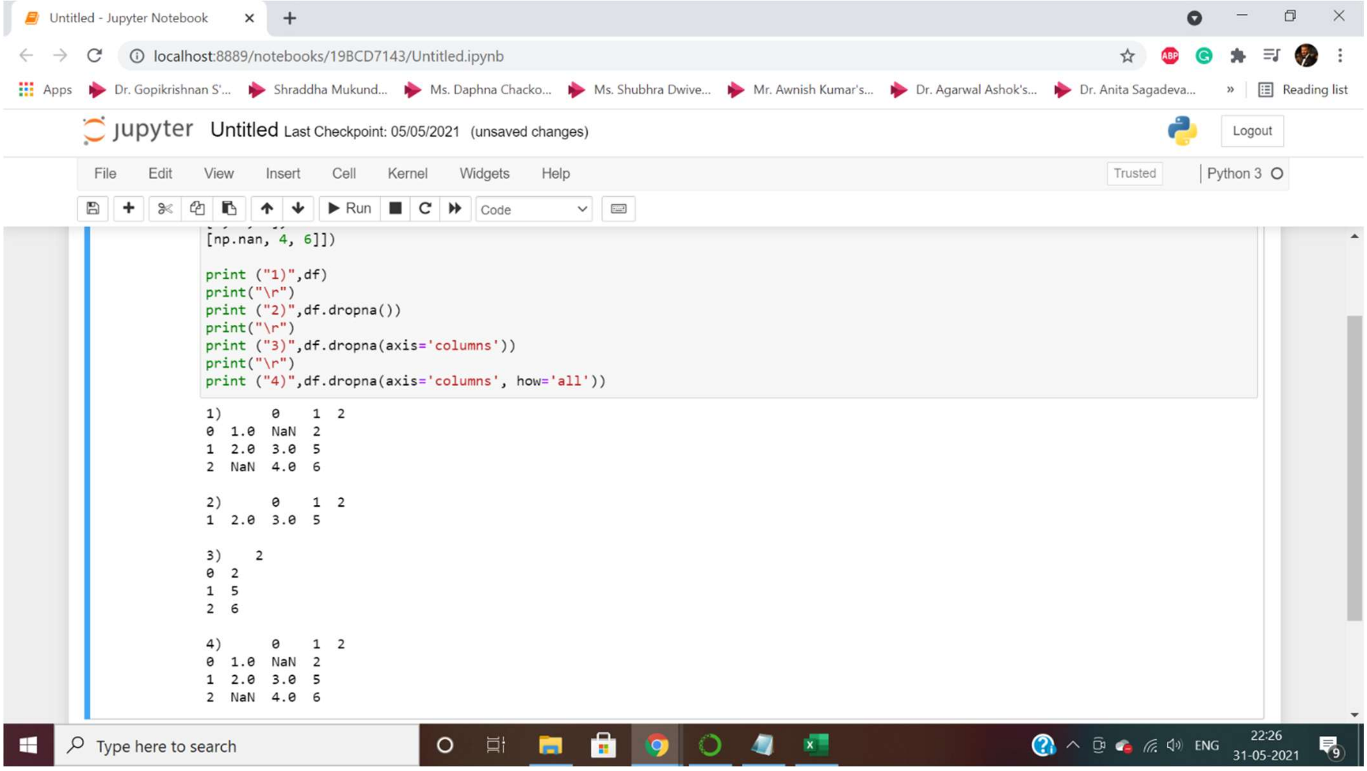
· Print df.dropna(axis='columns', how='all')

Answer:

```
import pandas as pd
import numpy as np
df = pd.DataFrame([[1, np.nan, 2],
[2, 3, 5],
[np.nan, 4, 6]])

print ("1)",df)
```

```
print("\r")
print ("2)",df.dropna())
print("\r")
print ("3)",df.dropna(axis='columns'))
print("\r")
print ("4)",df.dropna(axis='columns', how='all'))
```

<u>OUTPUT</u>


Untitled - Jupyter Notebook
localhost:8889/notebooks/19BCD7143/Untitled.ipynb
jupyter Untitled Last Checkpoint: 05/05/2021 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Type here to search
ENG
22:26
31-05-2021

```
[np.nan, 4, 6]])

print ("1)",df)
print("\r")
print ("2)",df.dropna())
print("\r")
print ("3)",df.dropna(axis='columns'))
print("\r")
print ("4)",df.dropna(axis='columns', how='all'))
```

```
1)      0    1  2
0  1.0  NaN  2
1  2.0  3.0  5
2  NaN  4.0  6

2)      0    1  2
1  2.0  3.0  5

3)    2
0  2
1  5
2  6

4)      0    1  2
0  1.0  NaN  2
1  2.0  3.0  5
2  NaN  4.0  6
```

8.

a. Consider the following dataframe : student_df and write a statement of below mentioned frame to get the minimum value of the column marks:

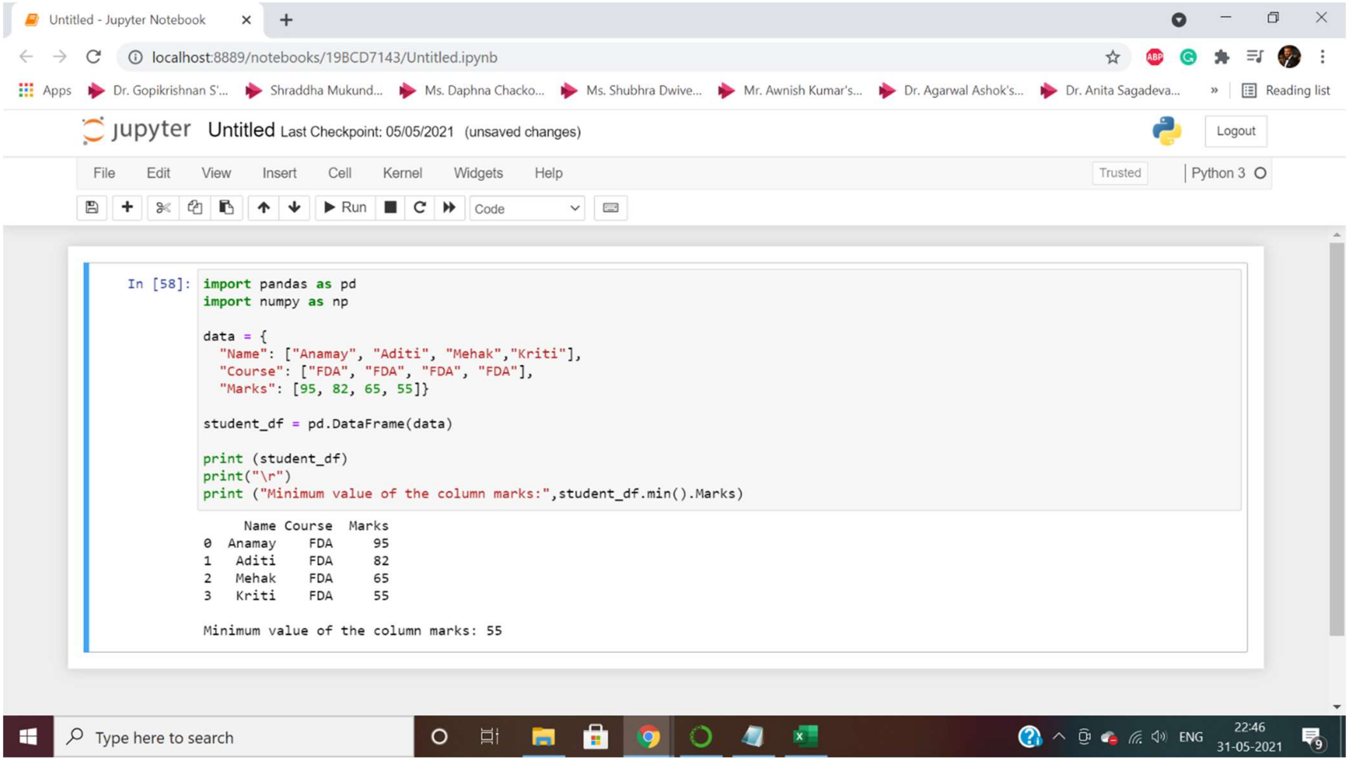
Name Course Marks
Anamay FDA 95
Aditi FDA 82
Mehak FDA 65
Kriti FDA 55

Answer:

import pandas as pd

```
import numpy as np

data = {
  "Name": ["Anamay", "Aditi", "Mehak","Kriti"],
  "Course": ["FDA", "FDA", "FDA", "FDA"],
  "Marks": [95, 82, 65, 55]}

student_df = pd.DataFrame(data)

print (student_df)
print("\r")
print ("Minimum value of the column marks:",student_df.min().Marks)
```

OUTPUT


Untitled - Jupyter Notebook
localhost:8889/notebooks/19BCD7143/Untitled.ipynb
jupyter Untitled Last Checkpoint: 05/05/2021 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
ENG
22:46
31-05-2021

```
In [58]: import pandas as pd
         import numpy as np

         data = {
           "Name": ["Anamay", "Aditi", "Mehak","Kriti"],
           "Course": ["FDA", "FDA", "FDA", "FDA"],
           "Marks": [95, 82, 65, 55]}

         student_df = pd.DataFrame(data)

         print (student_df)
         print("\r")
         print ("Minimum value of the column marks:",student_df.min().Marks)

             Name Course  Marks
         0  Anamay    FDA     95
         1   Aditi    FDA     82
         2   Mehak    FDA     65
         3   Kriti    FDA     55

         Minimum value of the column marks: 55
```

b. Write the output of the following syntax:

```
 import numpy as np
 array1=np.array([10,12,14,16,18,20,22])
print(array1[1:5:2])
```

Answer:

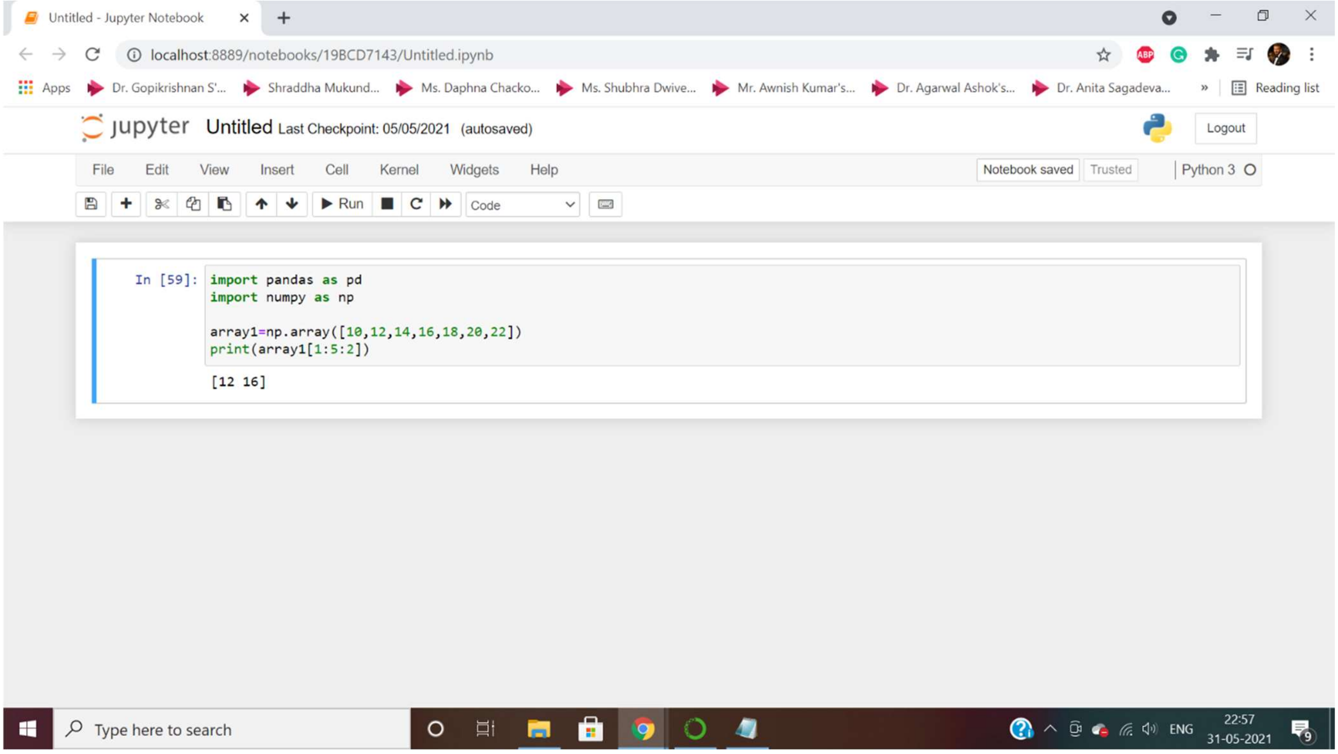
import pandas as pd

import numpy as np

array1=np.array([10,12,14,16,18,20,22])

print(array1[1:5:2])

OUTPUT



```
In [59]: import pandas as pd
         import numpy as np

         array1=np.array([10,12,14,16,18,20,22])
         print(array1[1:5:2])

         [12 16]
```

Untitled - Jupyter Notebook
localhost:8889/notebooks/19BCD7143/Untitled.ipynb
jupyter Untitled Last Checkpoint: 05/05/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Notebook saved Trusted Python 3
Run Code
Type here to search
ENG 22:57 31-05-2021

c. Consider the following dataframe, and answer the questions given below: import pandas as pd df = pd.DataFrame({“Quarter1":[2000, 4000, 5000, 4400, 10000], "Quarter2":[5800, 2500, 5400, 3000, 2900], "Quarter3":[20000, 16000, 7000, 3600, 8200], "Quarter4":[1400, 3700, 1700, 2000, 6000]})

(i) Write the code to find mean value from above dataframe df over the index and column axis.

(ii) Use sum() function to find the sum of all the values over the index axis.

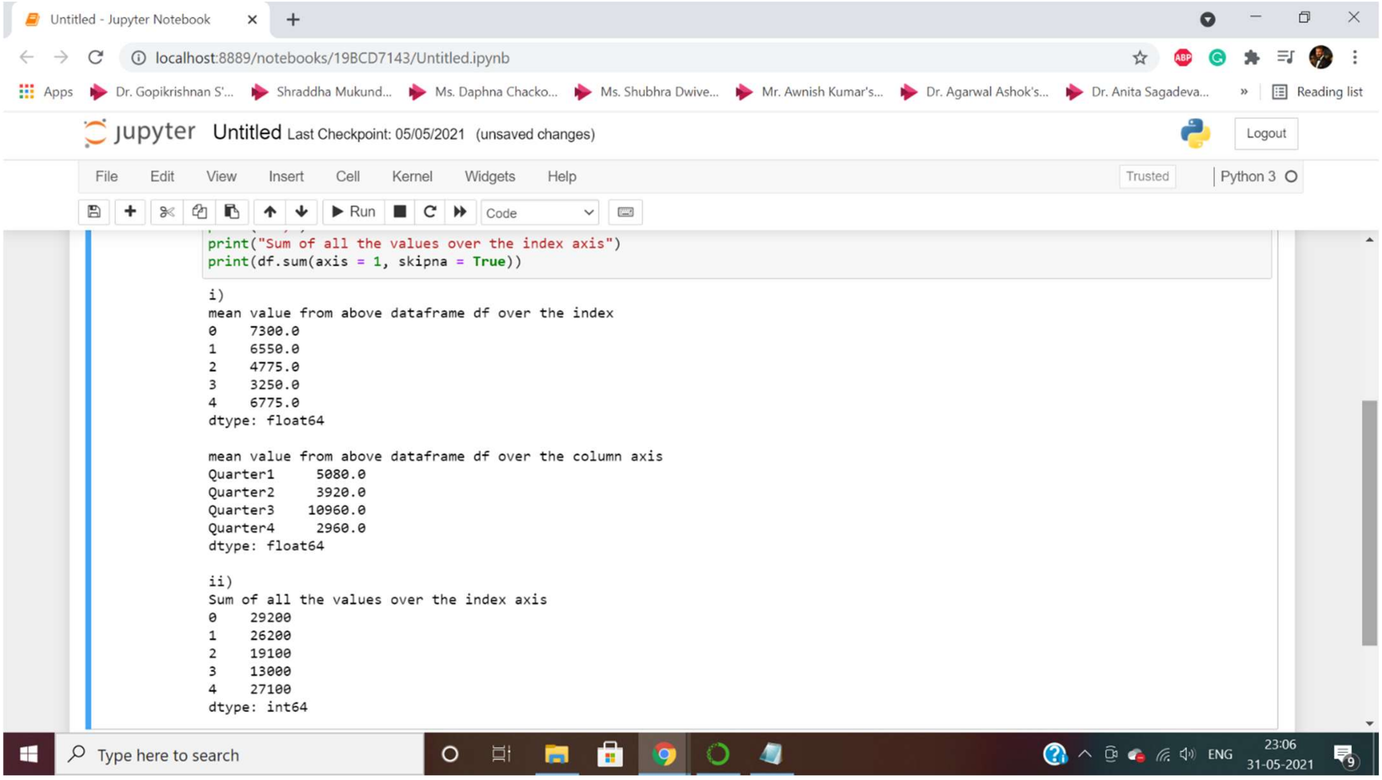
Answer:

import pandas as pd

import numpy as np

```
df = pd.DataFrame({"Quarter1":[2000, 4000, 5000, 4400, 10000],
"Quarter2":[5800, 2500, 5400, 3000, 2900], "Quarter3":[20000, 16000,
7000,
3600, 8200], "Quarter4":[1400, 3700, 1700, 2000, 6000]})

print("i)")
print("mean value from above dataframe df over the index")
print(df.mean(axis = 1, skipna = True))
print("\r")
print("mean value from above dataframe df over the column axis")
print(df.mean(axis = 0, skipna = True))
print("\r")
print("ii)")
print("Sum of all the values over the index axis")
print(df.sum(axis = 1, skipna = True))
```

OUTPUT



```
print("Sum of all the values over the index axis")
print(df.sum(axis = 1, skipna = True))
```

```
i)
mean value from above dataframe df over the index
0    7300.0
1    6550.0
2    4775.0
3    3250.0
4    6775.0
dtype: float64

mean value from above dataframe df over the column axis
Quarter1     5080.0
Quarter2     3920.0
Quarter3    10960.0
Quarter4     2960.0
dtype: float64

ii)
Sum of all the values over the index axis
0    29200
1    26200
2    19100
3    13000
4    27100
dtype: int64
```

9. In this question (9), we will use attached dataset as pima-indians-diabetes.csv.

· load the dataset

· summarize the dataset

· print the first 20 rows of data

· count the number of missing values for each column

· fill missing values with mean column values

· replace '0' values with 'nan

· count the number of nan values in each column

· drop rows with missing values

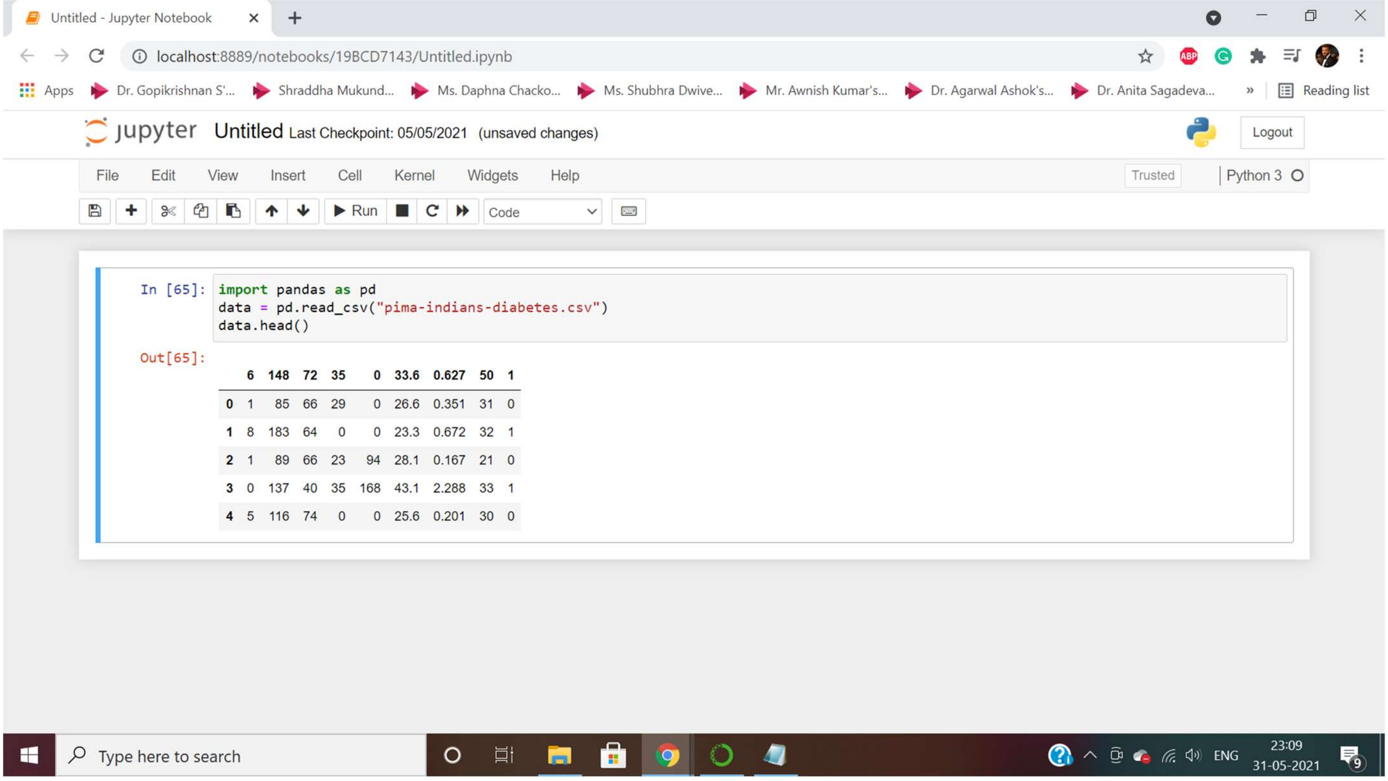
Answer:

1)import pandas as pd

data = pd.read_csv("pima-indians-diabetes.csv")

data.head()

OUTPUT



```
In [65]: import pandas as pd
         data = pd.read_csv("pima-indians-diabetes.csv")
         data.head()
```

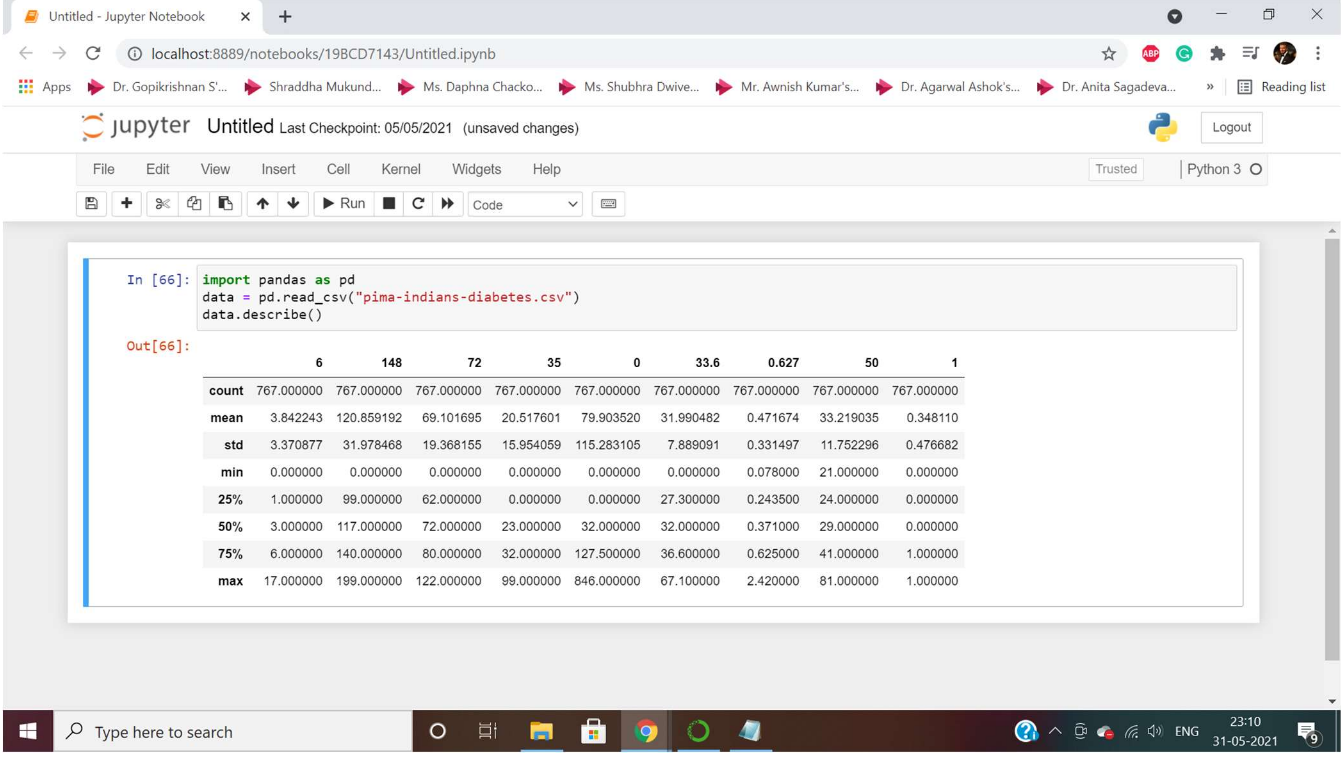
```
Out[65]:
      6  148  72  35    0  33.6  0.627  50  1
   0  1   85  66  29    0  26.6  0.351  31  0
   1  8  183  64   0    0  23.3  0.672  32  1
   2  1   89  66  23   94  28.1  0.167  21  0
   3  0  137  40  35  168  43.1  2.288  33  1
   4  5  116  74   0    0  25.6  0.201  30  0
```

2)import pandas as pd

data = pd.read_csv("pima-indians-diabetes.csv")

data.describe()

OUTPUT



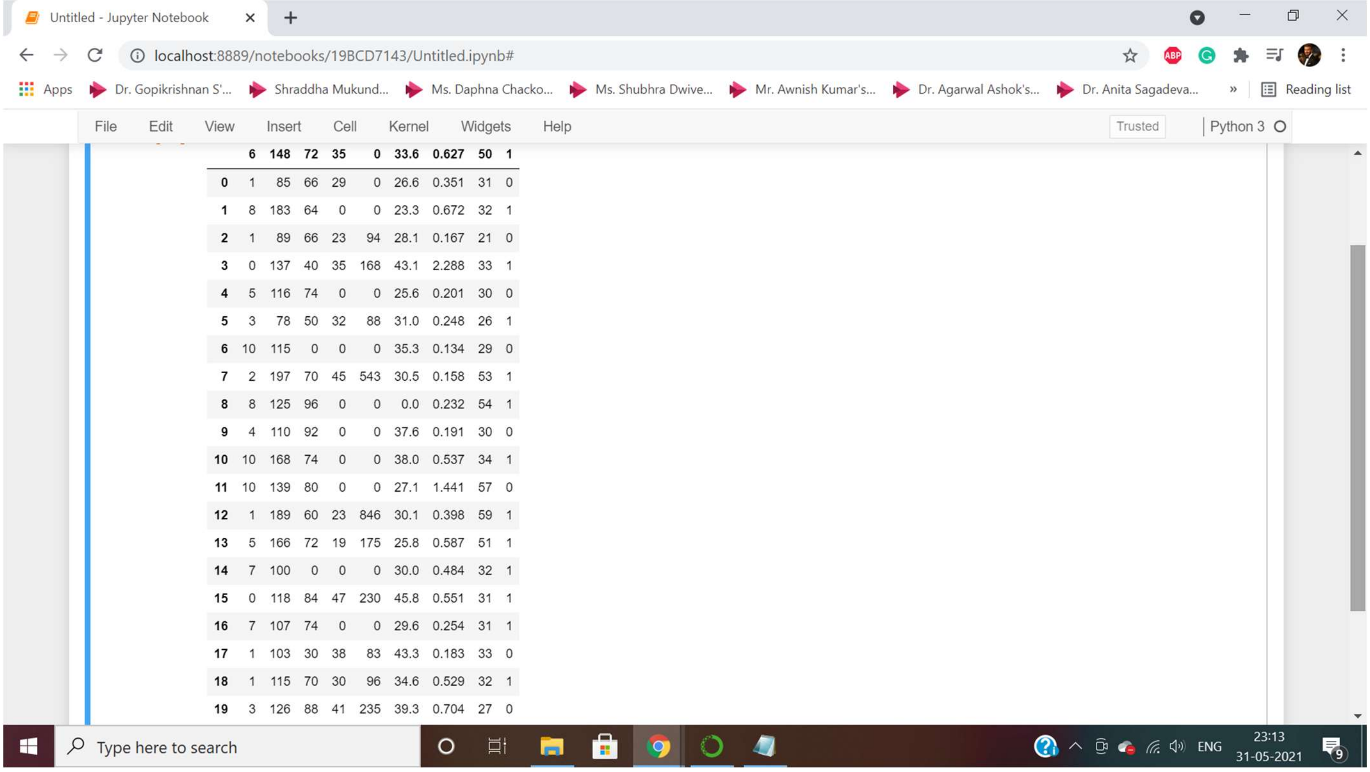
```
In [66]: import pandas as pd
         data = pd.read_csv("pima-indians-diabetes.csv")
         data.describe()
```

```
Out[66]:
                6         148          72          35           0        33.6       0.627          50           1
count  767.000000  767.000000  767.000000  767.000000  767.000000  767.000000  767.000000  767.000000  767.000000
mean     3.842243  120.859192   69.101695   20.517601   79.903520   31.990482    0.471674   33.219035    0.348110
std      3.370877   31.978468   19.368155   15.954059  115.283105    7.889091    0.331497   11.752296    0.476682
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.078000   21.000000    0.000000
25%      1.000000   99.000000   62.000000    0.000000    0.000000   27.300000    0.243500   24.000000    0.000000
50%      3.000000  117.000000   72.000000   23.000000   32.000000   32.000000    0.371000   29.000000    0.000000
75%      6.000000  140.000000   80.000000   32.000000  127.500000   36.600000    0.625000   41.000000    1.000000
max     17.000000  199.000000  122.000000   99.000000  846.000000   67.100000    2.420000   81.000000    1.000000
```

3)import pandas as pd

```
data = pd.read_csv("pima-indians-diabetes.csv")
data.head(20)
```

OUTPUT



```
     6  148  72  35    0  33.6  0.627  50  1
0    1   85  66  29    0  26.6  0.351  31  0
1    8  183  64   0    0  23.3  0.672  32  1
2    1   89  66  23   94  28.1  0.167  21  0
3    0  137  40  35  168  43.1  2.288  33  1
4    5  116  74   0    0  25.6  0.201  30  0
5    3   78  50  32   88  31.0  0.248  26  1
6   10  115   0   0    0  35.3  0.134  29  0
7    2  197  70  45  543  30.5  0.158  53  1
8    8  125  96   0    0   0.0  0.232  54  1
9    4  110  92   0    0  37.6  0.191  30  0
10  10  168  74   0    0  38.0  0.537  34  1
11  10  139  80   0    0  27.1  1.441  57  0
12   1  189  60  23  846  30.1  0.398  59  1
13   5  166  72  19  175  25.8  0.587  51  1
14   7  100   0   0    0  30.0  0.484  32  1
15   0  118  84  47  230  45.8  0.551  31  1
16   7  107  74   0    0  29.6  0.254  31  1
17   1  103  30  38   83  43.3  0.183  33  0
18   1  115  70  30   96  34.6  0.529  32  1
19   3  126  88  41  235  39.3  0.704  27  0
```

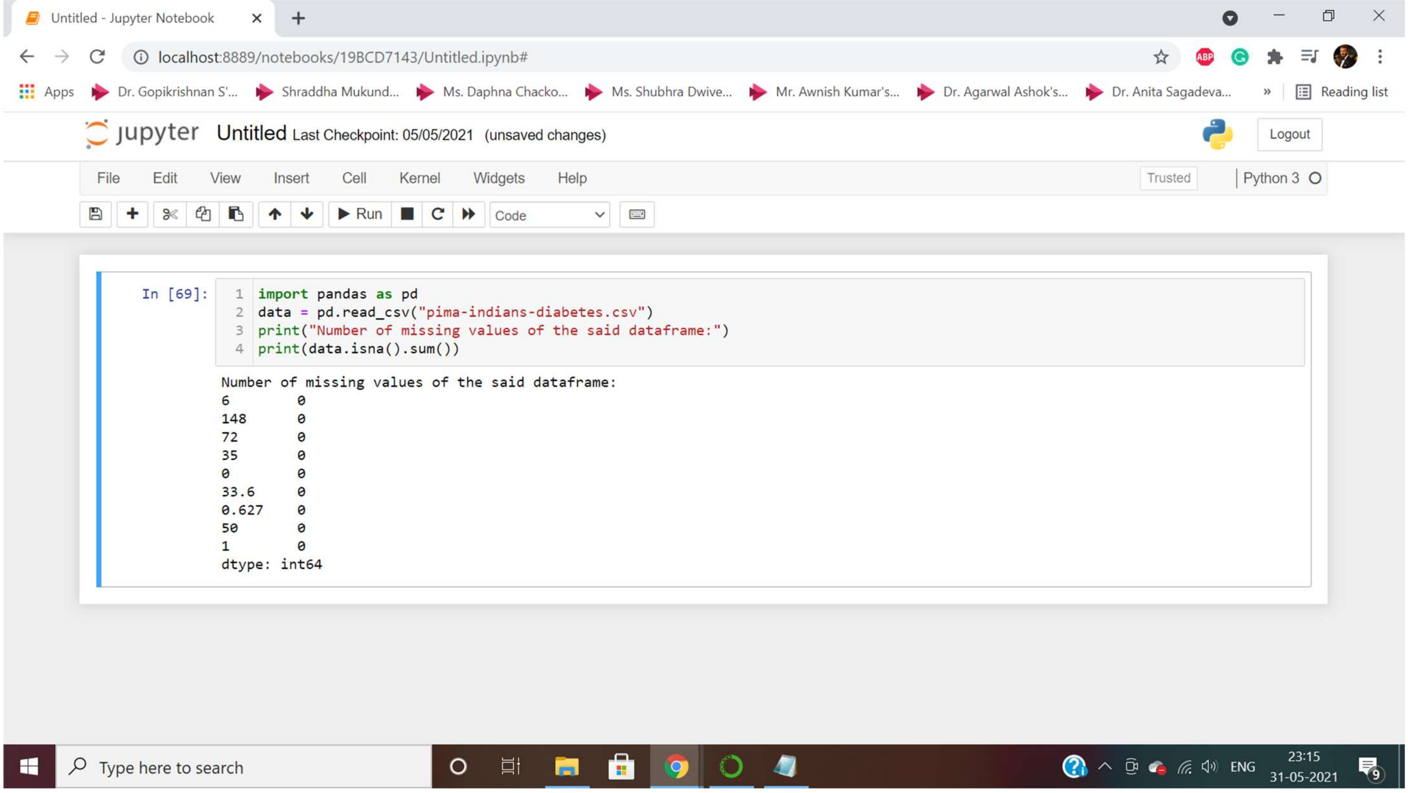
4)import pandas as pd

data = pd.read_csv("pima-indians-diabetes.csv")

print("Number of missing values of the said dataframe:")

print(data.isna().sum())

OUTPUT



```
import pandas as pd
data = pd.read_csv("pima-indians-diabetes.csv")
print("Number of missing values of the said dataframe:")
print(data.isna().sum())
```

```
Number of missing values of the said dataframe:
6        0
148      0
72       0
35       0
0        0
33.6     0
0.627    0
50       0
1        0
dtype: int64
```

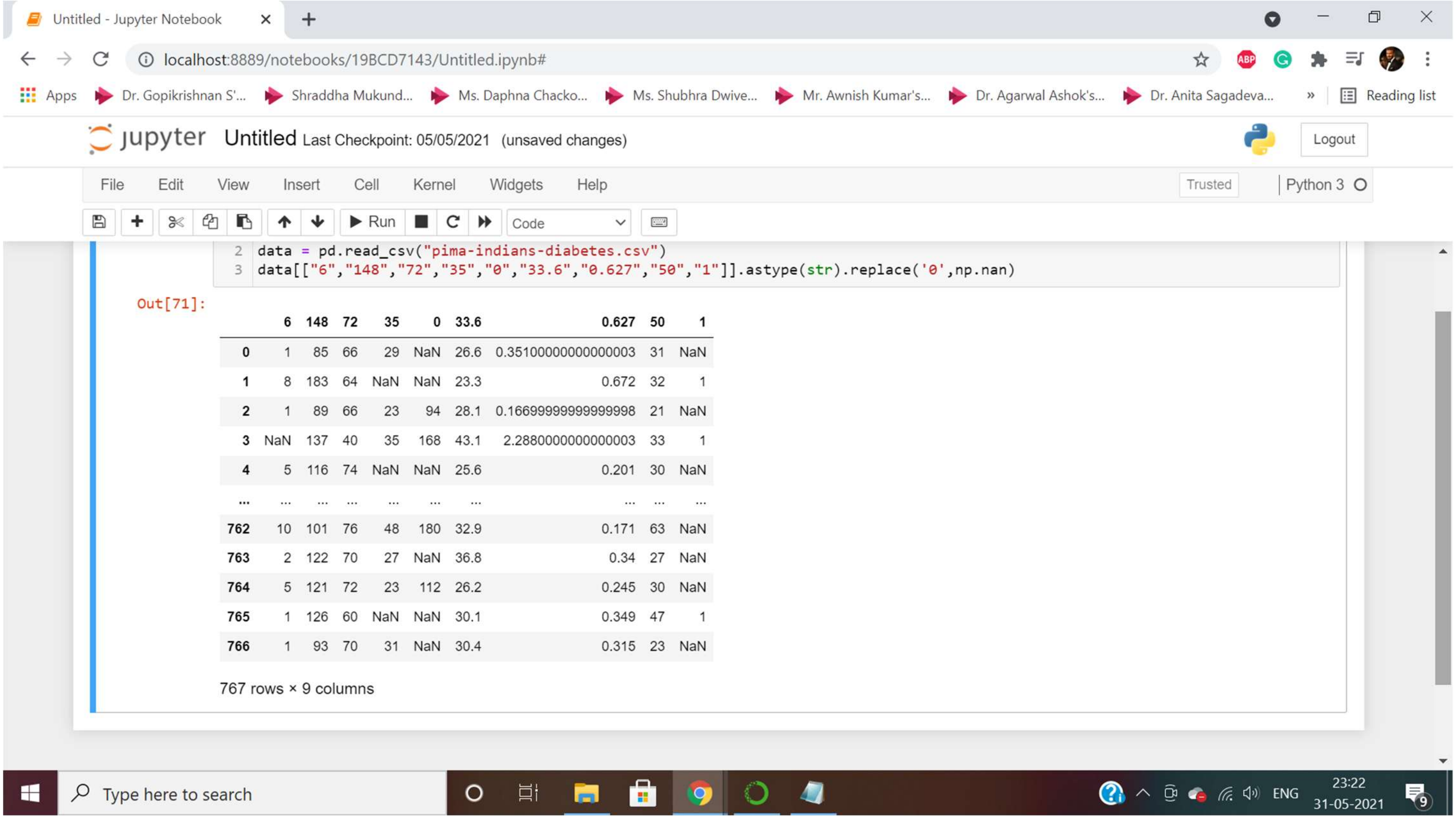
5)There are no missing values,so we cant fill it with mean values

6)import pandas as pd

data = pd.read_csv("pima-indians-diabetes.csv")

data[["6","148","72","35","0","33.6","0.627","50","1"]].astype(str).replace('0', np.nan)

OUTPUT


Untitled - Jupyter Notebook
localhost:8889/notebooks/19BCD7143/Untitled.ipynb#
jupyter Untitled Last Checkpoint: 05/05/2021 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Type here to search
ENG
23:22
31-05-2021

```
data = pd.read_csv("pima-indians-diabetes.csv")
data[["6","148","72","35","0","33.6","0.627","50","1"]].astype(str).replace('0',np.nan)
```

Out[71]:

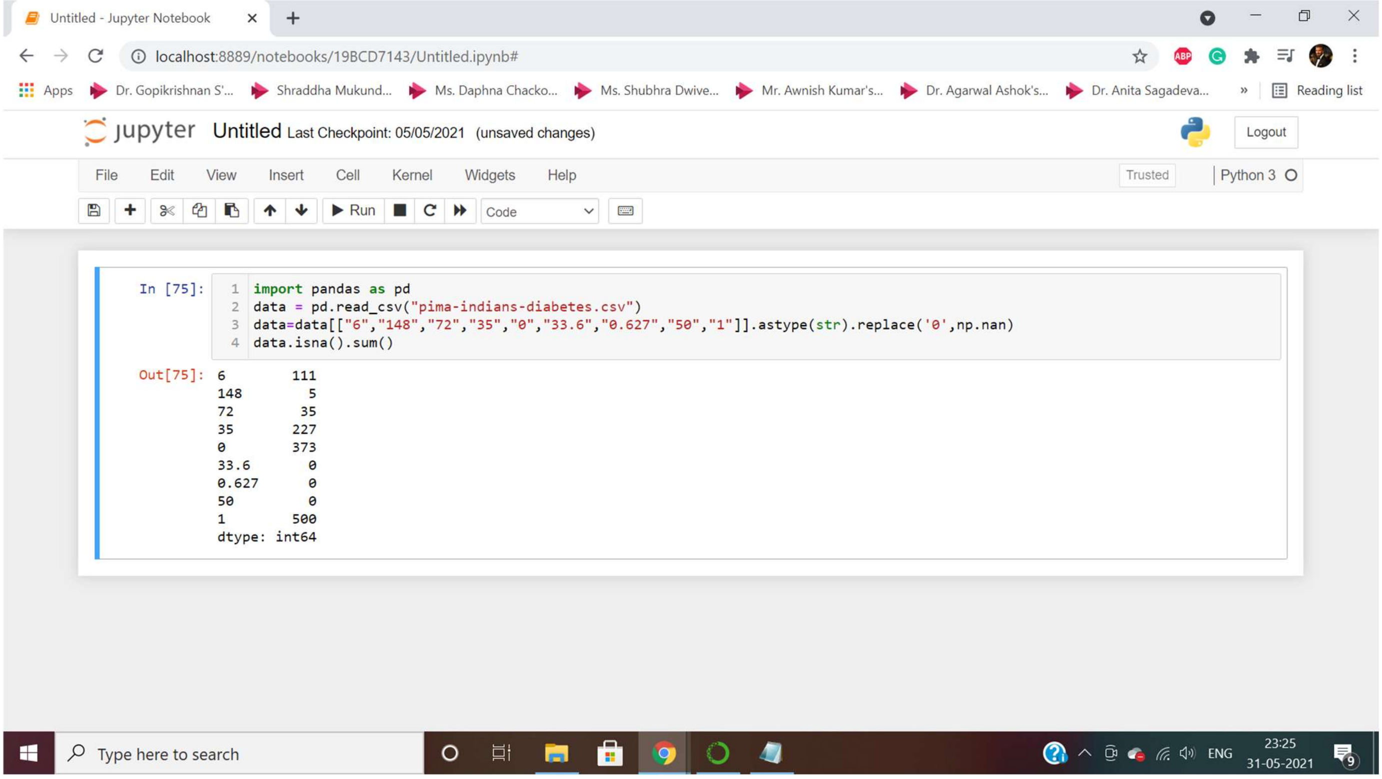
	6	148	72	35	0	33.6	0.627	50	1
0	1	85	66	29	NaN	26.6	0.35100000000000003	31	NaN
1	8	183	64	NaN	NaN	23.3	0.672	32	1
2	1	89	66	23	94	28.1	0.16699999999999998	21	NaN
3	NaN	137	40	35	168	43.1	2.2880000000000003	33	1
4	5	116	74	NaN	NaN	25.6	0.201	30	NaN
...	...	...	...	...	...	...	...	...	...
762	10	101	76	48	180	32.9	0.171	63	NaN
763	2	122	70	27	NaN	36.8	0.34	27	NaN
764	5	121	72	23	112	26.2	0.245	30	NaN
765	1	126	60	NaN	NaN	30.1	0.349	47	1
766	1	93	70	31	NaN	30.4	0.315	23	NaN

767 rows × 9 columns

7)import pandas as pd

```
data = pd.read_csv("pima-indians-diabetes.csv")
data=data[["6","148","72","35","0","33.6","0.627","50","1"]].astype(str).
replace('0',np.nan)
data.isna().sum()
```

OUTPUT



In [75]:

```python
import pandas as pd
data = pd.read_csv("pima-indians-diabetes.csv")
data=data[["6","148","72","35","0","33.6","0.627","50","1"]].astype(str).replace('0',np.nan)
data.isna().sum()
```

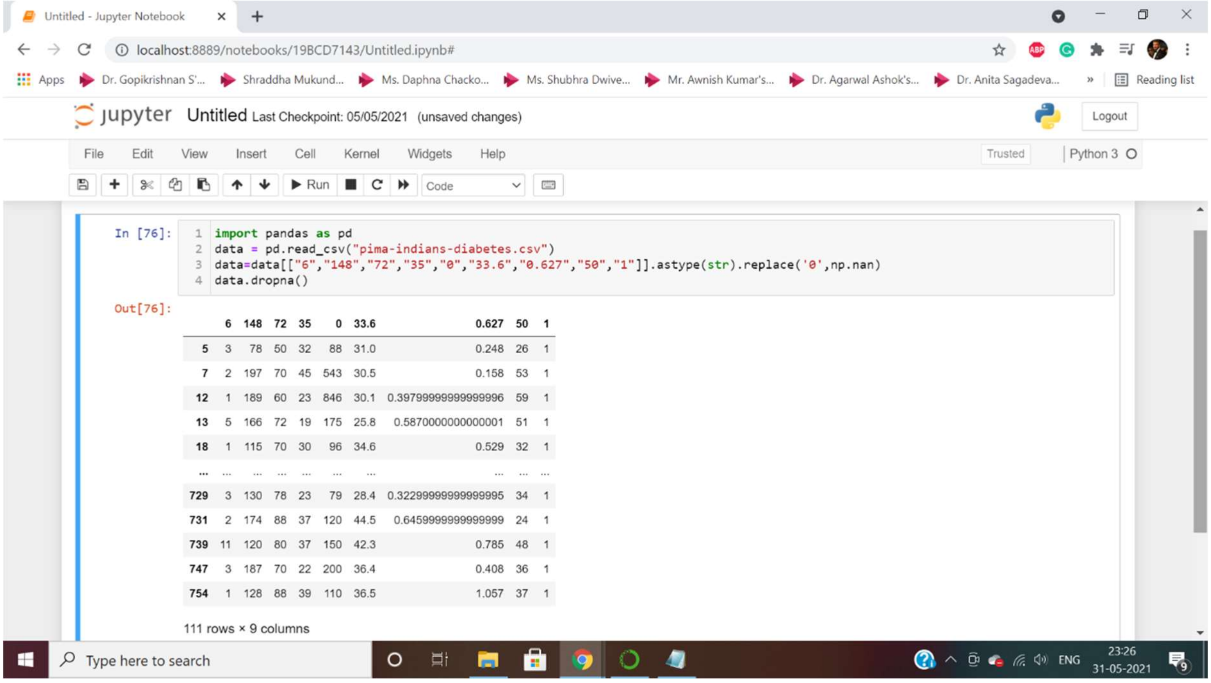
Out[75]:

```
6        111
148        5
72        35
35       227
0        373
33.6       0
0.627      0
50         0
1        500
dtype: int64
```

8)import pandas as pd   data = pd.read_csv("pima-indians-diabetes.csv")

data=data[["6","148","72","35","0","33.6","0.627","50","1"]].astype(str).
replace('0',np.nan)
data.dropna()

OUTPUT



In [76]:

```python
import pandas as pd
data = pd.read_csv("pima-indians-diabetes.csv")
data=data[["6","148","72","35","0","33.6","0.627","50","1"]].astype(str).replace('0',np.nan)
data.dropna()
```

Out[76]:

	6	148	72	35	0	33.6	0.627	50	1
5	3	78	50	32	88	31.0	0.248	26	1
7	2	197	70	45	543	30.5	0.158	53	1
12	1	189	60	23	846	30.1	0.39799999999999996	59	1
13	5	166	72	19	175	25.8	0.5870000000000001	51	1
18	1	115	70	30	96	34.6	0.529	32	1
...	...	...	...	...	...	...	...	...	...
729	3	130	78	23	79	28.4	0.32299999999999995	34	1
731	2	174	88	37	120	44.5	0.6459999999999999	24	1
739	11	120	80	37	150	42.3	0.785	48	1
747	3	187	70	22	200	36.4	0.408	36	1
754	1	128	88	39	110	36.5	1.057	37	1

111 rows × 9 columns

10. Consider the following dataframe, and answer the questions given below:
df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f', 'h'], columns=['one', 'two', 'three']) df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])

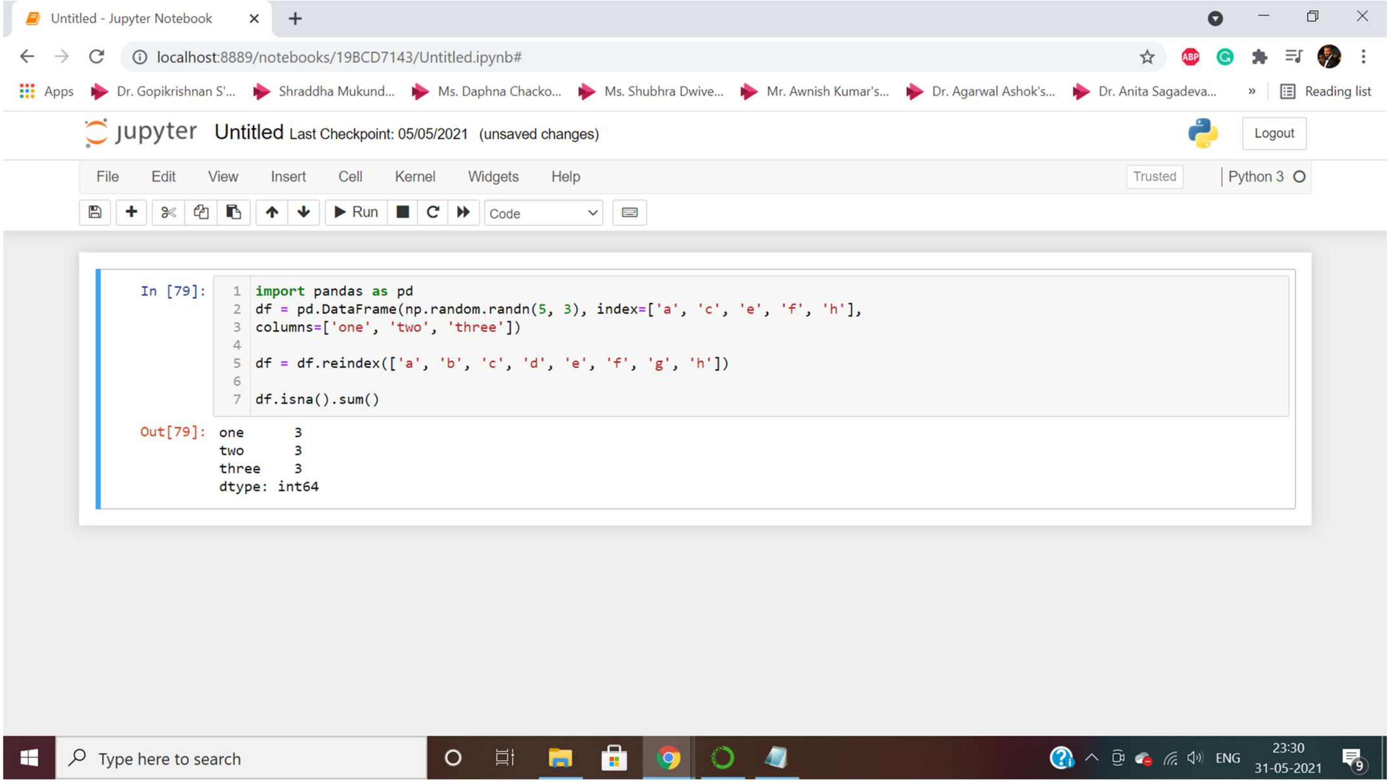
· Count the number of nan values in each column

· Replace nan values with 22

Answer:

```
1)import pandas as pd
df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f', 'h'],
columns=['one', 'two', 'three'])

df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])

df.isna().sum()
```

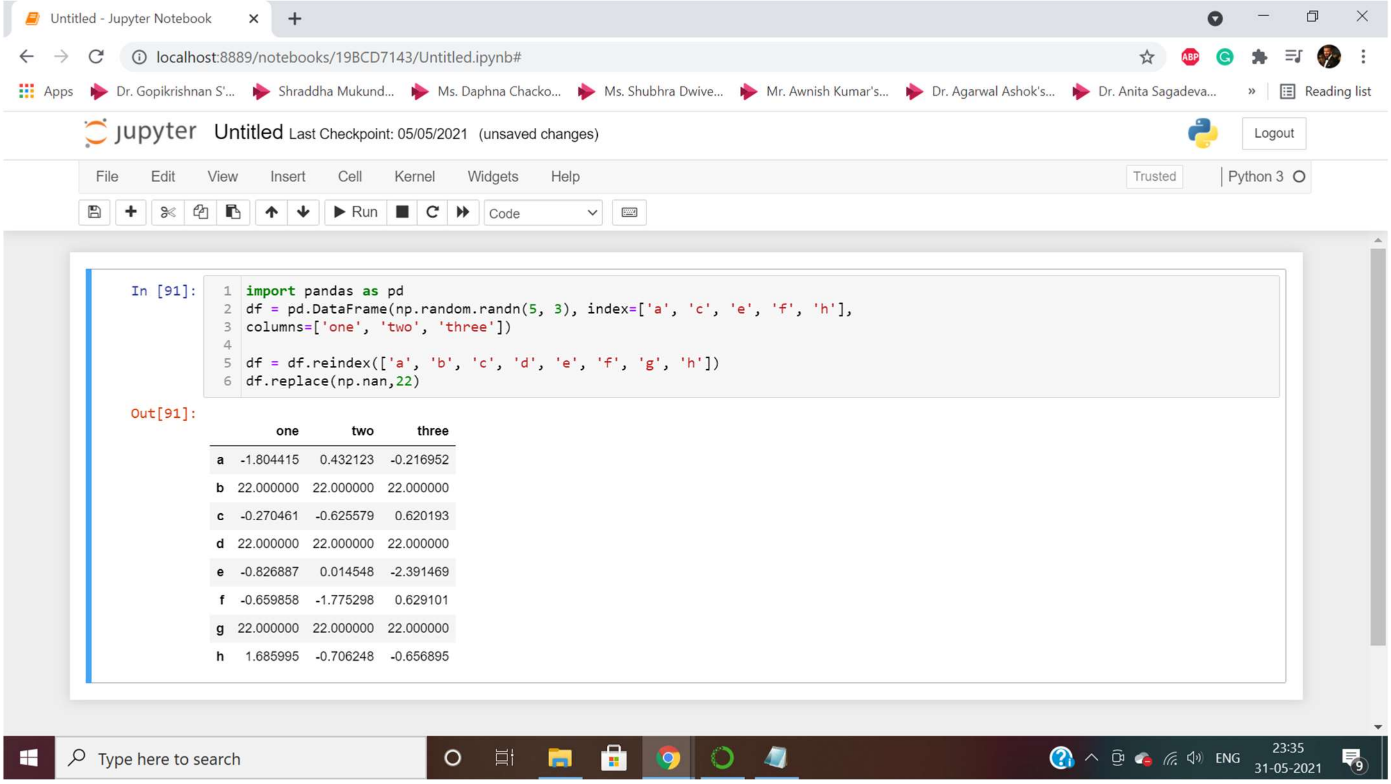
<u>OUTPUT</u>


Untitled - Jupyter Notebook
localhost:8889/notebooks/19BCD7143/Untitled.ipynb#
jupyter Untitled Last Checkpoint: 05/05/2021 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Type here to search
ENG
23:30
31-05-2021

```
In [79]:  import pandas as pd
          df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f', 'h'],
          columns=['one', 'two', 'three'])

          df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])

          df.isna().sum()

Out[79]: one      3
         two      3
         three    3
         dtype: int64
```

```
2)import pandas as pd
df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f', 'h'],
columns=['one', 'two', 'three'])

df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
df.replace(np.nan,22)
```

OUTPUT



In [91]:

```
import pandas as pd
df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f', 'h'],
columns=['one', 'two', 'three'])

df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
df.replace(np.nan,22)
```

Out[91]:

	one	two	three
a	-1.804415	0.432123	-0.216952
b	22.000000	22.000000	22.000000
c	-0.270461	-0.625579	0.620193
d	22.000000	22.000000	22.000000
e	-0.826887	0.014548	-2.391469
f	-0.659858	-1.775298	0.629101
g	22.000000	22.000000	22.000000
h	1.685995	-0.706248	-0.656895