

NUMPY

NumPy stands for numeric python which is a python package for the computation and processing of the multidimensional and single dimensional array elements.

NumPy provides a convenient and efficient way to handle the vast amount of data. NumPy is also very convenient with Matrix multiplication and data reshaping. NumPy is fast which makes it reasonable to work with a large set of data.

There are the following advantages of using NumPy for data analysis.

- 1. NumPy performs array-oriented computing.*
- 2. It efficiently implements the multidimensional arrays.*
- 3. It performs scientific computations.*
- 4. It is capable of performing Fourier Transform and reshaping the data stored in multidimensional arrays.*
- 5. NumPy provides the in-built functions for linear algebra and random number generation.*

Nowadays, NumPy in combination with SciPy and Matplotlib is used as the replacement to MATLAB as Python is more complete and easier programming language than MATLAB.

```

import numpy as np

mylist =[1,2,3,4,5,6,7]
myarray =np.array(mylist)

print(myarray)

[1 2 3 4 5 6 7]

myarray.shape

(7,)

x1=np.array([[1,2],[4,5]] , dtype ='g')

x1.dtype

dtype('float64')

x1=np.array([[1,2],[4,5]] , dtype =np.uint8)
x1.dtype

dtype('uint8')

x1=np.array([[1,2],[4,5]] , dtype =np.longdouble)
x1.dtype

dtype('float64')

x1=np.array([[1,2],[4,5]] , dtype =np.csingle)
x1.dtype

dtype('complex64')

x1=np.array([[1,2],[4,5]] , dtype =np.ubyte)
x1.dtype

dtype('uint8')

x=np.array([[1,2,3],[4,5,6],[7,8,9],[1,2,3]])

x.shape

(4, 3)

x

array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9],
       [1, 2, 3]])

x.dtype

dtype('int32')

```

```

x=np.array([1,2,3,4,5,6])
x.shape

(6,)

y=np.array([[1,2,3,4,5,6]])
y.shape

(1, 6)

arr1d =np.array([1,2,3,4,5])

arr2d =np.array([[1,2,3],[4,5,6],[7,8,9],[1,2,3],[4,5,6],[7,8,9]])
arr2d

array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9],
       [1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])

arr1d[2:]

array([3, 4, 5])

arr2d[1:,1:2]

array([[5],
       [8],
       [2],
       [5],
       [8]])

y=np.array([[1,2,3,4,5,6,7,8,9,10,11,12]])
k=y.reshape(4,3)

k

array([[ 1,  2,  3],
       [ 4,  5,  6],
       [ 7,  8,  9],
       [10, 11, 12]])

k.flatten()

array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])

np.transpose(k)

unsort_arr = np.array([[12,1,13],[15,13,0], [17,1,9]])

unsort_arr

```

```

a=np.random.randint(1,10,(5,2))

a
array([[3, 7],
       [5, 5],
       [3, 8],
       [9, 3],
       [2, 5]])

a
array([[3, 7],
       [5, 5],
       [3, 8],
       [9, 3],
       [2, 5]])

arr = np.array([-2, -1, 0, 1, 2])
result = np.piecewise(arr, [arr < 0, arr >= 0], [lambda x: x**2,
lambda x: x**3])
print(result)
[4 1 0 1 8]

arr = np.array([1, 2, 3, 4, 5])
condition = arr % 2 == 0
result = np.extract(condition, arr)
print(result)
[2 4]

arr = np.array([[1, 2], [3, 4], [5, 6]])
indices = np.array([[1], [0], [1]])
result = np.take_along_axis(arr, indices, axis=1)
print(result)
[[2]
 [3]
 [6]]

start_date = np.datetime64('2023-08-01')
end_date = np.datetime64('2023-08-07')
count = np.busday_count(start_date, end_date)
print(count)
4

data = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
bin_edges = np.histogram_bin_edges(data, bins='auto')
print(bin_edges)
[ 1.   2.8  4.6  6.4  8.2 10. ]

```

PANDAS

Pandas is an **open-source library** in **Python** that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library of Python. Pandas is fast and it has high performance & productivity for users.

- Fast and efficient for manipulating and analyzing data.
- Data from different file objects can be easily loaded.
- Flexible reshaping and pivoting of data sets
- Provides time-series functionality.

Pandas Data Structures:

Pandas generally provide two data structures for manipulating data, They are:

- **Series**
- **DataFrame**

The diagram illustrates a Pandas DataFrame with 7 rows and 5 columns. The columns are labeled 'Name', 'Team', 'Number', 'Position', and 'Age'. The rows are indexed from 0 to 6. Annotations include: 'Columns' pointing to the column headers, 'Rows' pointing to the row indices, and 'Data' pointing to the cell contents. A pink box highlights the data for the row where 'Team' is 'Boston Celtics' and 'Number' is '8.0'.

	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

Pandas library of python is very useful for the manipulation of mathematical data and is widely used in the field of machine learning. It comprises many methods for its proper functioning. `loc()` and `iloc()` are one of those methods. These are used in slicing data from the Pandas DataFrame. They help in the convenient selection of data from the DataFrame in Python. They are used in filtering the data according to some conditions.

Pandas loc vs. iloc

- `loc` selects rows and columns with specific labels
- `iloc` selects rows and columns at specific integer positions

```
import pandas as pd
file="sample.csv"
df =pd.read_csv(file)
df
```

	id	name	salary	department
0	1	john	2000	sales
1	2	Andrew	5000	finance
2	3	Mark	8000	hr
3	4	Rey	5000	marketing
4	5	Tan	4000	IT
5	6	Sarah	3500	sales
6	7	Michael	6000	finance
7	8	Emily	7000	hr
8	9	David	5500	marketing
9	10	Lisa	4500	IT
10	11	Chris	3000	sales
11	12	Michelle	6500	finance
12	13	Alex	7500	hr
13	14	Oliver	4800	marketing
14	15	Natalie	4200	IT
15	16	James	3200	sales
16	17	Grace	5800	finance
17	18	William	8200	hr
18	19	Samantha	5200	marketing
19	20	Benjamin	3900	IT
20	21	Ava	2800	sales
21	22	Daniel	6700	finance
22	23	Emma	7200	hr
23	24	Liam	5100	marketing
24	25	Sophia	4100	IT
25	26	Matthew	3100	sales
26	27	Chloe	5400	finance
27	28	Joshua	7900	hr
28	29	Ashley	5300	marketing
29	30	Ethan	4700	IT
30	31	Anna	2600	sales
31	32	William	6200	finance
32	33	Isabella	7800	hr
33	34	Lucas	5900	marketing
34	35	Mia	4300	IT
35	36	Aiden	3400	sales
36	37	Grace	5600	finance
37	38	Jack	8400	hr
38	39	Sophia	5700	marketing
39	40	Logan	4400	IT
40	41	Amelia	2700	sales
41	42	Owen	6900	finance

42	43	Charlotte	7600	hr
43	44	Harper	5000	marketing
44	45	Liam	4600	IT
45	46	Mia	3300	sales
46	47	James	5500	finance
47	48	Abigail	8100	hr
48	49	Emily	5900	marketing
49	50	Noah	4200	IT

```
df[['name', 'department']]
```

	name	department
0	john	sales
1	Andrew	finance
2	Mark	hr
3	Rey	marketing
4	Tan	IT
5	Sarah	sales
6	Michael	finance
7	Emily	hr
8	David	marketing
9	Lisa	IT
10	Chris	sales
11	Michelle	finance
12	Alex	hr
13	Oliver	marketing
14	Natalie	IT
15	James	sales
16	Grace	finance
17	William	hr
18	Samantha	marketing
19	Benjamin	IT
20	Ava	sales
21	Daniel	finance
22	Emma	hr
23	Liam	marketing
24	Sophia	IT
25	Matthew	sales
26	Chloe	finance
27	Joshua	hr
28	Ashley	marketing
29	Ethan	IT
30	Anna	sales
31	William	finance
32	Isabella	hr
33	Lucas	marketing
34	Mia	IT
35	Aiden	sales
36	Grace	finance
37	Jack	hr
38	Sophia	marketing

```

39     Logan      IT
40   Amelia    sales
41     Owen    finance
42 Charlotte     hr
43   Harper  marketing
44     Liam      IT
45     Mia     sales
46   James    finance
47   Abigail     hr
48     Emily  marketing
49     Noah      IT

```

```
df.iloc[44]
```

```

id          45
name        Liam
salary      4600
department   IT
Name: 44, dtype: object

```

```
df.iloc[44][3]
```

```
'IT'
```

```
df.iloc[::-1]
```

```

      id    name  salary department
49  50    Noah   4200         IT
48  49    Emily   5900  marketing
47  48  Abigail   8100         hr
46  47    James   5500    finance
45  46     Mia   3300     sales
44  45     Liam   4600         IT
43  44    Harper   5000  marketing
42  43  Charlotte   7600         hr
41  42     Owen   6900    finance
40  41    Amelia   2700     sales
39  40    Logan   4400         IT
38  39    Sophia   5700  marketing
37  38     Jack   8400         hr
36  37    Grace   5600    finance
35  36    Aiden   3400     sales
34  35     Mia   4300         IT
33  34    Lucas   5900  marketing
32  33  Isabella   7800         hr
31  32    William   6200    finance
30  31     Anna   2600     sales
29  30     Ethan   4700         IT
28  29    Ashley   5300  marketing
27  28    Joshua   7900         hr
26  27     Chloe   5400    finance

```


25	26	Matthew	3100	sales
24	25	Sophia	4100	IT
23	24	Liam	5100	marketing
22	23	Emma	7200	hr
21	22	Daniel	6700	finance
20	21	Ava	2800	sales
19	20	Benjamin	3900	IT
18	19	Samantha	5200	marketing
17	18	William	8200	hr
16	17	Grace	5800	finance
15	16	James	3200	sales
14	15	Natalie	4200	IT
13	14	Oliver	4800	marketing
12	13	Alex	7500	hr
11	12	Michelle	6500	finance
10	11	Chris	3000	sales
9	10	Lisa	4500	IT
8	9	David	5500	marketing
7	8	Emily	7000	hr
6	7	Michael	6000	finance
5	6	Sarah	3500	sales
4	5	Tan	4000	IT
3	4	Rey	5000	marketing
2	3	Mark	8000	hr
1	2	Andrew	5000	finance
0	1	john	2000	sales

```
df.iloc[44,[2,3]]
```

```
salary      4600
department   IT
Name: 44, dtype: object
```

```
df.iloc[[44,45,46],[0,2,3]]
```

	id	salary	department
44	45	4600	IT
45	46	3300	sales
46	47	5500	finance

```
df.iloc[44:,1:]
```

	name	salary	department
44	Liam	4600	IT
45	Mia	3300	sales
46	James	5500	finance
47	Abigail	8100	hr
48	Emily	5900	marketing
49	Noah	4200	IT

```
df.loc[44:,'salary']
```

```

44    4600
45    3300
46    5500
47    8100
48    5900
49    4200
Name: salary, dtype: int64

df.loc[44:48,'salary']

44    4600
45    3300
46    5500
47    8100
48    5900
Name: salary, dtype: int64

df.iloc[44:48,0:2]

   id  name
44  45  Liam
45  46   Mia
46  47  James
47  48 Abigail

df[df.name == 'Liam']

   id  name  salary department
23  24  Liam    5100  marketing
44  45  Liam    4600         IT

df.name == 'Liam'

0    False
1    False
2    False
3    False
4    False
5    False
6    False
7    False
8    False
9    False
10   False
11   False
12   False
13   False
14   False
15   False
16   False
17   False
18   False

```

```
19     False
20     False
21     False
22     False
23     True
24     False
25     False
26     False
27     False
28     False
29     False
30     False
31     False
32     False
33     False
34     False
35     False
36     False
37     False
38     False
39     False
40     False
41     False
42     False
43     False
44     True
45     False
46     False
47     False
48     False
49     False
Name: name, dtype: bool
```

```
df[df['salary']==7500]
```

```
   id  name  salary department
12  13  Alex   7500         hr
```

```
df[(df['salary']>4000)&(df['department']=='IT')]
```

```
   id  name  salary department
9   10   Lisa   4500         IT
14  15  Natalie  4200         IT
24  25   Sophia  4100         IT
29  30   Ethan   4700         IT
34  35    Mia   4300         IT
39  40   Logan   4400         IT
44  45    Liam   4600         IT
49  50   Noah   4200         IT
```

```
df.rename(columns = {'salary':'bonus'},inplace=True)
```

df

	id	name	bonus	department
0	1	john	2000	sales
1	2	Andrew	5000	finance
2	3	Mark	8000	hr
3	4	Rey	5000	marketing
4	5	Tan	4000	IT
5	6	Sarah	3500	sales
6	7	Michael	6000	finance
7	8	Emily	7000	hr
8	9	David	5500	marketing
9	10	Lisa	4500	IT
10	11	Chris	3000	sales
11	12	Michelle	6500	finance
12	13	Alex	7500	hr
13	14	Oliver	4800	marketing
14	15	Natalie	4200	IT
15	16	James	3200	sales
16	17	Grace	5800	finance
17	18	William	8200	hr
18	19	Samantha	5200	marketing
19	20	Benjamin	3900	IT
20	21	Ava	2800	sales
21	22	Daniel	6700	finance
22	23	Emma	7200	hr
23	24	Liam	5100	marketing
24	25	Sophia	4100	IT
25	26	Matthew	3100	sales
26	27	Chloe	5400	finance
27	28	Joshua	7900	hr
28	29	Ashley	5300	marketing
29	30	Ethan	4700	IT
30	31	Anna	2600	sales
31	32	William	6200	finance
32	33	Isabella	7800	hr
33	34	Lucas	5900	marketing
34	35	Mia	4300	IT
35	36	Aiden	3400	sales
36	37	Grace	5600	finance
37	38	Jack	8400	hr
38	39	Sophia	5700	marketing
39	40	Logan	4400	IT
40	41	Amelia	2700	sales
41	42	Owen	6900	finance
42	43	Charlotte	7600	hr
43	44	Harper	5000	marketing
44	45	Liam	4600	IT
45	46	Mia	3300	sales
46	47	James	5500	finance

```

47 48    Abigail    8100    hr
48 49      Emily    5900 marketing
49 50      Noah    4200    IT

```

```
df['salary']=df['bonus']*12
```

```
df
```

	id	name	bonus	department	salary
0	1	john	2000	sales	24000
1	2	Andrew	5000	finance	60000
2	3	Mark	8000	hr	96000
3	4	Rey	5000	marketing	60000
4	5	Tan	4000	IT	48000
5	6	Sarah	3500	sales	42000
6	7	Michael	6000	finance	72000
7	8	Emily	7000	hr	84000
8	9	David	5500	marketing	66000
9	10	Lisa	4500	IT	54000
10	11	Chris	3000	sales	36000
11	12	Michelle	6500	finance	78000
12	13	Alex	7500	hr	90000
13	14	Oliver	4800	marketing	57600
14	15	Natalie	4200	IT	50400
15	16	James	3200	sales	38400
16	17	Grace	5800	finance	69600
17	18	William	8200	hr	98400
18	19	Samantha	5200	marketing	62400
19	20	Benjamin	3900	IT	46800
20	21	Ava	2800	sales	33600
21	22	Daniel	6700	finance	80400
22	23	Emma	7200	hr	86400
23	24	Liam	5100	marketing	61200
24	25	Sophia	4100	IT	49200
25	26	Matthew	3100	sales	37200
26	27	Chloe	5400	finance	64800
27	28	Joshua	7900	hr	94800
28	29	Ashley	5300	marketing	63600
29	30	Ethan	4700	IT	56400
30	31	Anna	2600	sales	31200
31	32	William	6200	finance	74400
32	33	Isabella	7800	hr	93600
33	34	Lucas	5900	marketing	70800
34	35	Mia	4300	IT	51600
35	36	Aiden	3400	sales	40800
36	37	Grace	5600	finance	67200
37	38	Jack	8400	hr	100800
38	39	Sophia	5700	marketing	68400
39	40	Logan	4400	IT	52800
40	41	Amelia	2700	sales	32400
41	42	Owen	6900	finance	82800

```

42 43 Charlotte 7600 hr 91200
43 44 Harper 5000 marketing 60000
44 45 Liam 4600 IT 55200
45 46 Mia 3300 sales 39600
46 47 James 5500 finance 66000
47 48 Abigail 8100 hr 97200
48 49 Emily 5900 marketing 70800
49 50 Noah 4200 IT 50400

```

```
df['CTC']=df['bonus']+df['salary']
```

```
df
```

	id	name	bonus	department	salary	CTC
0	1	John	2000	sales	24000	26000
1	2	Andrew	5000	finance	60000	65000
2	3	Mark	8000	hr	96000	104000
3	4	Rey	5000	marketing	60000	65000
4	5	Tan	4000	IT	48000	52000
5	6	Sarah	3500	sales	42000	45500
6	7	Michael	6000	finance	72000	78000
7	8	Emily	7000	hr	84000	91000
8	9	David	5500	marketing	66000	71500
9	10	Lisa	4500	IT	54000	58500
10	11	Chris	3000	sales	36000	39000
11	12	Michelle	6500	finance	78000	84500
12	13	Alex	7500	hr	90000	97500
13	14	Oliver	4800	marketing	57600	62400
14	15	Natalie	4200	IT	50400	54600
15	16	James	3200	sales	38400	41600
16	17	Grace	5800	finance	69600	75400
17	18	William	8200	hr	98400	106600
18	19	Samantha	5200	marketing	62400	67600
19	20	Benjamin	3900	IT	46800	50700
20	21	Ava	2800	sales	33600	36400
21	22	Daniel	6700	finance	80400	87100
22	23	Emma	7200	hr	86400	93600
23	24	Liam	5100	marketing	61200	66300
24	25	Sophia	4100	IT	49200	53300
25	26	Matthew	3100	sales	37200	40300
26	27	Chloe	5400	finance	64800	70200
27	28	Joshua	7900	hr	94800	102700
28	29	Ashley	5300	marketing	63600	68900
29	30	Ethan	4700	IT	56400	61100
30	31	Anna	2600	sales	31200	33800
31	32	William	6200	finance	74400	80600
32	33	Isabella	7800	hr	93600	101400
33	34	Lucas	5900	marketing	70800	76700
34	35	Mia	4300	IT	51600	55900
35	36	Aiden	3400	sales	40800	44200
36	37	Grace	5600	finance	67200	72800

37	38	Jack	8400	hr	100800	109200
38	39	Sophia	5700	marketing	68400	74100
39	40	Logan	4400	IT	52800	57200
40	41	Amelia	2700	sales	32400	35100
41	42	Owen	6900	finance	82800	89700
42	43	Charlotte	7600	hr	91200	98800
43	44	Harper	5000	marketing	60000	65000
44	45	Liam	4600	IT	55200	59800
45	46	Mia	3300	sales	39600	42900
46	47	James	5500	finance	66000	71500
47	48	Abigail	8100	hr	97200	105300
48	49	Emily	5900	marketing	70800	76700
49	50	Noah	4200	IT	50400	54600

```
df['increment']=df['bonus']+df['salary']*10/100
```

```
df
```

	id	name	bonus	department	salary	CTC	increment
0	1	john	2000	sales	24000	26000	4400.0
1	2	Andrew	5000	finance	60000	65000	11000.0
2	3	Mark	8000	hr	96000	104000	17600.0
3	4	Rey	5000	marketing	60000	65000	11000.0
4	5	Tan	4000	IT	48000	52000	8800.0
5	6	Sarah	3500	sales	42000	45500	7700.0
6	7	Michael	6000	finance	72000	78000	13200.0
7	8	Emily	7000	hr	84000	91000	15400.0
8	9	David	5500	marketing	66000	71500	12100.0
9	10	Lisa	4500	IT	54000	58500	9900.0
10	11	Chris	3000	sales	36000	39000	6600.0
11	12	Michelle	6500	finance	78000	84500	14300.0
12	13	Alex	7500	hr	90000	97500	16500.0
13	14	Oliver	4800	marketing	57600	62400	10560.0
14	15	Natalie	4200	IT	50400	54600	9240.0
15	16	James	3200	sales	38400	41600	7040.0
16	17	Grace	5800	finance	69600	75400	12760.0
17	18	William	8200	hr	98400	106600	18040.0
18	19	Samantha	5200	marketing	62400	67600	11440.0
19	20	Benjamin	3900	IT	46800	50700	8580.0
20	21	Ava	2800	sales	33600	36400	6160.0
21	22	Daniel	6700	finance	80400	87100	14740.0
22	23	Emma	7200	hr	86400	93600	15840.0
23	24	Liam	5100	marketing	61200	66300	11220.0
24	25	Sophia	4100	IT	49200	53300	9020.0
25	26	Matthew	3100	sales	37200	40300	6820.0
26	27	Chloe	5400	finance	64800	70200	11880.0
27	28	Joshua	7900	hr	94800	102700	17380.0
28	29	Ashley	5300	marketing	63600	68900	11660.0
29	30	Ethan	4700	IT	56400	61100	10340.0
30	31	Anna	2600	sales	31200	33800	5720.0
31	32	William	6200	finance	74400	80600	13640.0

13 14	Oliver	4800	marketing	57600	62400	10560.0
72960.0						
14 15	Natalie	4200	IT	50400	54600	9240.0
63840.0						
15 16	James	3200	sales	38400	41600	7040.0
48640.0						
16 17	Grace	5800	finance	69600	75400	12760.0
88160.0						
17 18	William	8200	hr	98400	106600	18040.0
124640.0						
18 19	Samantha	5200	marketing	62400	67600	11440.0
79040.0						
19 20	Benjamin	3900	IT	46800	50700	8580.0
59280.0						
20 21	Ava	2800	sales	33600	36400	6160.0
42560.0						
21 22	Daniel	6700	finance	80400	87100	14740.0
101840.0						
22 23	Emma	7200	hr	86400	93600	15840.0
109440.0						
23 24	Liam	5100	marketing	61200	66300	11220.0
77520.0						
24 25	Sophia	4100	IT	49200	53300	9020.0
62320.0						
25 26	Matthew	3100	sales	37200	40300	6820.0
47120.0						
26 27	Chloe	5400	finance	64800	70200	11880.0
82080.0						
27 28	Joshua	7900	hr	94800	102700	17380.0
120080.0						
28 29	Ashley	5300	marketing	63600	68900	11660.0
80560.0						
29 30	Ethan	4700	IT	56400	61100	10340.0
71440.0						
30 31	Anna	2600	sales	31200	33800	5720.0
39520.0						
31 32	William	6200	finance	74400	80600	13640.0
94240.0						
32 33	Isabella	7800	hr	93600	101400	17160.0
118560.0						
33 34	Lucas	5900	marketing	70800	76700	12980.0
89680.0						
34 35	Mia	4300	IT	51600	55900	9460.0
65360.0						
35 36	Aiden	3400	sales	40800	44200	7480.0
51680.0						
36 37	Grace	5600	finance	67200	72800	12320.0
85120.0						
37 38	Jack	8400	hr	100800	109200	18480.0
127680.0						

38	39	Sophia	5700	marketing	68400	74100	12540.0	
86640.0	39	40	Logan	4400	IT	52800	57200	9680.0
66880.0	40	41	Amelia	2700	sales	32400	35100	5940.0
41040.0	41	42	Owen	6900	finance	82800	89700	15180.0
104880.0	42	43	Charlotte	7600	hr	91200	98800	16720.0
115520.0	43	44	Harper	5000	marketing	60000	65000	11000.0
76000.0	44	45	Liam	4600	IT	55200	59800	10120.0
69920.0	45	46	Mia	3300	sales	39600	42900	7260.0
50160.0	46	47	James	5500	finance	66000	71500	12100.0
83600.0	47	48	Abigail	8100	hr	97200	105300	17820.0
123120.0	48	49	Emily	5900	marketing	70800	76700	12980.0
89680.0	49	50	Noah	4200	IT	50400	54600	9240.0
63840.0								

```
df['New CTC in float ']=df['New CTC'].astype(float)
```

df

id	name	bonus	department	salary	CTC	increment	New
0	John	2000	sales	24000	26000	4400.0	
1	Andrew	5000	finance	60000	65000	11000.0	
2	Mark	8000	hr	96000	104000	17600.0	
3	Rey	5000	marketing	60000	65000	11000.0	
4	Tan	4000	IT	48000	52000	8800.0	
5	Sarah	3500	sales	42000	45500	7700.0	
6	Michael	6000	finance	72000	78000	13200.0	
7	Emily	7000	hr	84000	91000	15400.0	
8	David	5500	marketing	66000	71500	12100.0	
9	Lisa	4500	IT	54000	58500	9900.0	

10	11	Chris	3000	sales	36000	39000	6600.0
45600.0							
11	12	Michelle	6500	finance	78000	84500	14300.0
98800.0							
12	13	Alex	7500	hr	90000	97500	16500.0
114000.0							
13	14	Oliver	4800	marketing	57600	62400	10560.0
72960.0							
14	15	Natalie	4200	IT	50400	54600	9240.0
63840.0							
15	16	James	3200	sales	38400	41600	7040.0
48640.0							
16	17	Grace	5800	finance	69600	75400	12760.0
88160.0							
17	18	William	8200	hr	98400	106600	18040.0
124640.0							
18	19	Samantha	5200	marketing	62400	67600	11440.0
79040.0							
19	20	Benjamin	3900	IT	46800	50700	8580.0
59280.0							
20	21	Ava	2800	sales	33600	36400	6160.0
42560.0							
21	22	Daniel	6700	finance	80400	87100	14740.0
101840.0							
22	23	Emma	7200	hr	86400	93600	15840.0
109440.0							
23	24	Liam	5100	marketing	61200	66300	11220.0
77520.0							
24	25	Sophia	4100	IT	49200	53300	9020.0
62320.0							
25	26	Matthew	3100	sales	37200	40300	6820.0
47120.0							
26	27	Chloe	5400	finance	64800	70200	11880.0
82080.0							
27	28	Joshua	7900	hr	94800	102700	17380.0
120080.0							
28	29	Ashley	5300	marketing	63600	68900	11660.0
80560.0							
29	30	Ethan	4700	IT	56400	61100	10340.0
71440.0							
30	31	Anna	2600	sales	31200	33800	5720.0
39520.0							
31	32	William	6200	finance	74400	80600	13640.0
94240.0							
32	33	Isabella	7800	hr	93600	101400	17160.0
118560.0							
33	34	Lucas	5900	marketing	70800	76700	12980.0
89680.0							
34	35	Mia	4300	IT	51600	55900	9460.0
65360.0							

35	36	Aiden	3400	sales	40800	44200	7480.0
51680.0							
36	37	Grace	5600	finance	67200	72800	12320.0
85120.0							
37	38	Jack	8400	hr	100800	109200	18480.0
127680.0							
38	39	Sophia	5700	marketing	68400	74100	12540.0
86640.0							
39	40	Logan	4400	IT	52800	57200	9680.0
66880.0							
40	41	Amelia	2700	sales	32400	35100	5940.0
41040.0							
41	42	Owen	6900	finance	82800	89700	15180.0
104880.0							
42	43	Charlotte	7600	hr	91200	98800	16720.0
115520.0							
43	44	Harper	5000	marketing	60000	65000	11000.0
76000.0							
44	45	Liam	4600	IT	55200	59800	10120.0
69920.0							
45	46	Mia	3300	sales	39600	42900	7260.0
50160.0							
46	47	James	5500	finance	66000	71500	12100.0
83600.0							
47	48	Abigail	8100	hr	97200	105300	17820.0
123120.0							
48	49	Emily	5900	marketing	70800	76700	12980.0
89680.0							
49	50	Noah	4200	IT	50400	54600	9240.0
63840.0							

New CTC in float

0	30400.0
1	76000.0
2	121600.0
3	76000.0
4	60800.0
5	53200.0
6	91200.0
7	106400.0
8	83600.0
9	68400.0
10	45600.0
11	98800.0
12	114000.0
13	72960.0
14	63840.0
15	48640.0
16	88160.0
17	124640.0

18	79040.0
19	59280.0
20	42560.0
21	101840.0
22	109440.0
23	77520.0
24	62320.0
25	47120.0
26	82080.0
27	120080.0
28	80560.0
29	71440.0
30	39520.0
31	94240.0
32	118560.0
33	89680.0
34	65360.0
35	51680.0
36	85120.0
37	127680.0
38	86640.0
39	66880.0
40	41040.0
41	104880.0
42	115520.0
43	76000.0
44	69920.0
45	50160.0
46	83600.0
47	123120.0
48	89680.0
49	63840.0