

Ex.No-7

Data Aggregation and Grouping

Aim:

To perform Data Aggregation and Grouping functions

Description:

1. Create a DataFrame
2. Implement Data Aggregation and Grouping functions

Program:

```
import pandas as pd
import numpy as np

df = pd.DataFrame({'A' : ['foo', 'foo', 'bar', 'foo', 'bar', 'foo'],
                   'B' : ['one', 'two', 'three','two', 'two', 'one'],
                   'C' : np.random.randint(5, size=6),
                   'D' : np.random.randint(5, size=6),
                   'E' : np.random.randint(5, size=6)})

print("\nOriginal DataFrame\n",df)

#Group by (multi-column): with one column sum
df1 = df.groupby(['A', 'B'], as_index=False)['C'].sum()
print("\nOriginal DataFrame\n",df)
print("\nGroup by (multi-column): with one column sum\n",df1)

#Group by (multi-column): with multi columns sum
df2 = df.groupby(['A', 'B'], as_index=False).sum()
print("\nOriginal DataFrame\n",df)
print("\nGroup by (multi-column): with multi columns sum\n",df2)

#Combined Groupby and Aggregate function
df3 = df.groupby(['A', 'B'], as_index=False)['C'].agg('sum')
print("\nOriginal DataFrame\n",df)
```

```
print("\nCombined Groupby and Aggregate function\n",df3)

#Combined Groupby and Aggregate function- separate column headings
df4 = (df.groupby(['A', 'B'])
        .agg(['average', 'mean'], ('total', 'sum'))))
print("\nOriginal DataFrame\n",df)
print("\nCombined Groupby and Aggregate function- separate column headings\n",df4)

df5 = df.groupby(['A', 'B'], as_index=False).sum()
df6 = (df.groupby(['A', 'B']).agg(['sum']))
print("\nOriginal DataFrame\n",df)
print("\nGroupby - sum function\n",df5)
print("\nGroupby and Aggregate - sum function\n",df6)
```

Output:

Original DataFrame

	A	B	C	D	E
0	foo	one	3	0	3
1	foo	two	2	3	0
2	bar	three	1	2	1
3	foo	two	2	1	4
4	bar	two	4	0	3
5	foo	one	1	2	3

Original DataFrame

	A	B	C	D	E
0	foo	one	3	0	3
1	foo	two	2	3	0
2	bar	three	1	2	1
3	foo	two	2	1	4
4	bar	two	4	0	3

```
5 foo one 1 2 3
```

Group by (multi-column): with one column sum

```
  A  B C
```

```
0 bar three 1
```

```
1 bar two 4
```

```
2 foo one 4
```

```
3 foo two 4
```

Original DataFrame

```
  A  B C D E
```

```
0 foo one 3 0 3
```

```
1 foo two 2 3 0
```

```
2 bar three 1 2 1
```

```
3 foo two 2 1 4
```

```
4 bar two 4 0 3
```

```
5 foo one 1 2 3
```

Group by (multi-column): with multi columns sum

```
  A  B C D E
```

```
0 bar three 1 2 1
```

```
1 bar two 4 0 3
```

```
2 foo one 4 2 6
```

```
3 foo two 4 4 4
```

Original DataFrame

```
  A  B C D E
```

```
0 foo one 3 0 3
```

```
1 foo two 2 3 0
```

```
2 bar three 1 2 1
```

```
3 foo two 2 1 4
```

```
4 bar two 4 0 3
5 foo one 1 2 3
```

Combined Groupby and Aggregate function

```
  A  B C
0 bar three 1
1 bar two 4
2 foo one 4
3 foo two 4
```

Original DataFrame

```
  A  B C D E
0 foo one 3 0 3
1 foo two 2 3 0
2 bar three 1 2 1
3 foo two 2 1 4
4 bar two 4 0 3
5 foo one 1 2 3
```

Combined Groupby and Aggregate function- separate column headings

```
      C      D      E
average total average total average total
A B
bar three  1.0  1    2.0  2    1.0  1
      two  4.0  4    0.0  0    3.0  3
foo one   2.0  4    1.0  2    3.0  6
      two  2.0  4    2.0  4    2.0  4
```

Original DataFrame

```
  A  B C D E
0 foo one 3 0 3
```

```
1 foo two 2 3 0
2 bar three 1 2 1
3 foo two 2 1 4
4 bar two 4 0 3
5 foo one 1 2 3
```

Groupby - sum function

```
   A  B C D E
0 bar three 1 2 1
1 bar two 4 0 3
2 foo one 4 2 6
3 foo two 4 4 4
```

Groupby and Aggregate - sum function

```
   C  D  E
sum sum sum
A  B
bar three 1 2 1
    two 4 0 3
foo one 4 2 6
    two 4 4 4
```

Result:

Hence the programs were run successfully.