

# Session5-Pandas (Groupby, Operations)

DAwithPython S-5

Training Clarusway

Pear Deck - January 24, 2022 at 9:01PM

## Part 1 - Summary

Use this space to summarize your thoughts on the lesson

## Part 2 - Responses

Slide 1



Use this space to take notes:

## Slide 2



Use this space to take notes:

## Slide 3

### ► Table of Contents

- ▶ Basic Aggregation Methods
- ▶ Grouping Data ( DataFrame.groupby() )
- ▶ DataFrame/Series Operations
  - Aggregate
  - Filter
  - Transform

Use this space to take notes:

## Slide 4

### Your Response

You Chose

I've completed the pre-class content?

True

False

Students choose an option

Pear Deck Interactive Slide  
Do not remove this bar

- True

Other Choices

- False

Use this space to take notes:

Slide 5	Your Response
<p>Write down three of the Aggregation Methods in Pandas?</p> <p>Students, write your response!</p> <p>Pear Deck Interactive Slide Do not remove this bar</p>	<p><b>Answer 1:</b> sum min max median std group by</p>

Use this space to take notes:

Slide 6
---------

## ► Basic Aggregation Methods ➤

- ▶ count()
- ▶ min()
- ▶ mean()
- ▶ max()
- ▶ median()
- ▶ std()
- ▶ idxmin()
- ▶ var()
- ▶ idxmax()
- ▶ sum()
- ▶ corr()

8

Use this space to take notes:

## Slide 7

### ► Grouping Data ( DataFrame.groupby ) ➤

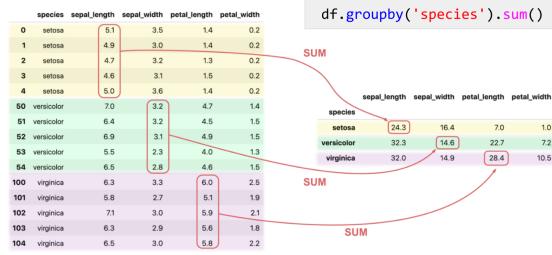
- ▶ DataFrame.groupby operation involves some combination of **splitting the object, applying a function, and combining the results.**
- ▶ This can be used to group large amounts of data and compute operations on these groups.

9

Use this space to take notes:

## Slide 8

## ▶ Grouping Data ( DataFrame.groupby() ) ➤



Use this space to take notes:

## Slide 9

## ▶ DataFrame/Series Operations ➤

### DataFrame.aggregate/agg

<pre>&gt;&gt;&gt; df = pd.DataFrame([[1, 2, 3], ...                   [4, 5, 6], ...                   [7, 8, 9], ...                   [np.nan, np.nan, np.nan]], ...                   columns=['A', 'B', 'C'])</pre>	<p>Aggregate different functions over the columns and rename the index</p> <pre>&gt;&gt;&gt; df.agg(x=('A', max), y=('B', 'min'), z=('C', np.mean))    A    B         C x  7.0  Null  Null y  Null  2.0  Null z  Null  Null  6.0</pre>
<p>Aggregate these functions over the rows.</p> <pre>&gt;&gt;&gt; df.agg(['sum', 'sin'])    A    B sum 12.0  15.0 sin  1.0  2.0 min  1.0  3.0</pre>	<p>Aggregate over the columns.</p> <pre>&gt;&gt;&gt; df.agg("mean", axis="columns")    A    B 0  2.0 1  5.0 2  8.0 3  Null</pre>
<p>Different aggregations per column.</p> <pre>&gt;&gt;&gt; df.agg({'A' : ['sum', 'min'], 'B' : ['min', 'max']})    A    B sum 12.0  Null min  1.0  2.0 max  Null  8.0</pre>	

Use this space to take notes:

## Slide 10

## ► DataFrame/Series Operations ➞

### DataFrame.groupby.aggregate/agg

```
>>> df
   A   B       C
0  1  0.227877
1  2  0.362838
2  2  1.277077
3  2  -0.562868

The aggregation is for each column.

>>> df.groupby('A').agg(['min'])
          B       C
A
1  1  0.227877
2  3  -0.562868

Multiple aggregations

>>> df.groupby('A').agg(['min', 'max'])
          B       C
A  min  max  min  max
1  1  2  0.227877  0.362838
2  3  4 -0.562868  1.277077
```

Select a column for aggregation

```
>>> df.groupby('A').B.agg(['min', 'max'])
          min  max
A
1  1  2
2  3  4
```

Different aggregations per column

```
>>> df.groupby('A').agg({'B': ['min', 'max'], 'C': 'sum'})
          B       C
A  min  max  sum
1  1  2  0.59815
2  3  4  0.784907
```

10

Use this space to take notes:

## Slide 11

## ► DataFrame/Series Operations ➞

### DataFrame.filter

- Subset the dataframe rows or columns according to the specified index labels.

```
>>> df = pd.DataFrame(np.array([[1, 2, 3], [4, 5, 6]]),
...                   index=['mouse', 'rabbit'],
...                   columns=['one', 'two', 'three'])
>>> df
   one  two  three
mouse    1    2    3
rabbit   4    5    6
```

# select columns by name

```
>>> df.filter(items=['one', 'three'])
          one  three
mouse    1    3
rabbit   4    6
```

# select columns by regular expression

```
>>> df.filter(regex='e$', axis=1)
          one  three
mouse    1    3
rabbit   4    6
```

# select rows containing 'bbi'

```
>>> df.filter(like='bbi', axis=0)
          one  two  three
rabbit   4    5    6
```

11

Use this space to take notes:

## Slide 12

## ► DataFrame/Series Operations ➤

### DataFrame.groupby.filter

- ▶ Return a copy of a DataFrame excluding filtered elements.

```
df = pd.DataFrame({'A' : ['foo', 'bar', 'foo', 'bar',
                           'foo', 'bar'],
                   'B' : [1, 2, 3, 4, 5, 6],
                   'C' : [2.0, 5., 0., 1., 2., 9.]})  
  
df  
  
A B C  
0 foo 1 2.0  
1 bar 2 5.0  
2 foo 3 8.0  
3 bar 4 1.0  
4 foo 5 2.0  
5 bar 6 9.0
```

```
df.groupby('A').mean()  
  
B C  
A  
bar 4 5.0  
foo 3 4.0  
  
df.groupby('A').filter(lambda x: x['B'].mean() > 3.)  
  
A B C  
1 bar 2 5.0  
3 bar 4 1.0  
5 bar 6 9.0
```

12

Use this space to take notes:

## Slide 13

## ► DataFrame/Series Operations ➤

### DataFrame.transform

- ▶ Produced DataFrame will have same axis length as self.

```
>>> df = pd.DataFrame({'A': range(3), 'B': range(1, 4)})  
>>> df  
A B  
0 0 1  
1 1 2  
2 2 3  
>>> df.transform(lambda x: x + 1)  
A B  
0 1 2  
1 2 3  
2 3 4
```

```
>>> s = pd.Series(range(3))  
>>> s  
0 0  
1 1  
2 2  
dtype: int64  
>>> s.transform([np.sqrt, np.exp])  
sqrt exp  
0 0.000000 1.000000  
1 1.000000 2.718282  
2 1.414214 7.389056
```

13

Use this space to take notes:

## Slide 14

## ▶ DataFrame/Series Operations ➤

### DataFrame.groupby.transform

df2		
groups	var1	var2
0	A	100
1	B	253
2	C	333
3	A	262
4	B	111
5	C	99
6	A	405
7	B	578
8	C	760

df2.groupby("groups")["var1"].mean()		
groups		
A	36.000000	
B	39.333333	
C	59.000000	
Name: var1, dtype: float64		
df2.groupby("groups")["var1"].transform("mean")		
0	36.000000	
1	39.333333	
2	59.000000	
3	36.000000	
4	39.333333	
5	59.000000	
6	36.000000	
7	39.333333	
8	59.000000	

14

Use this space to take notes:

Slide 15

## ▶ Data Analysis with Python ➤



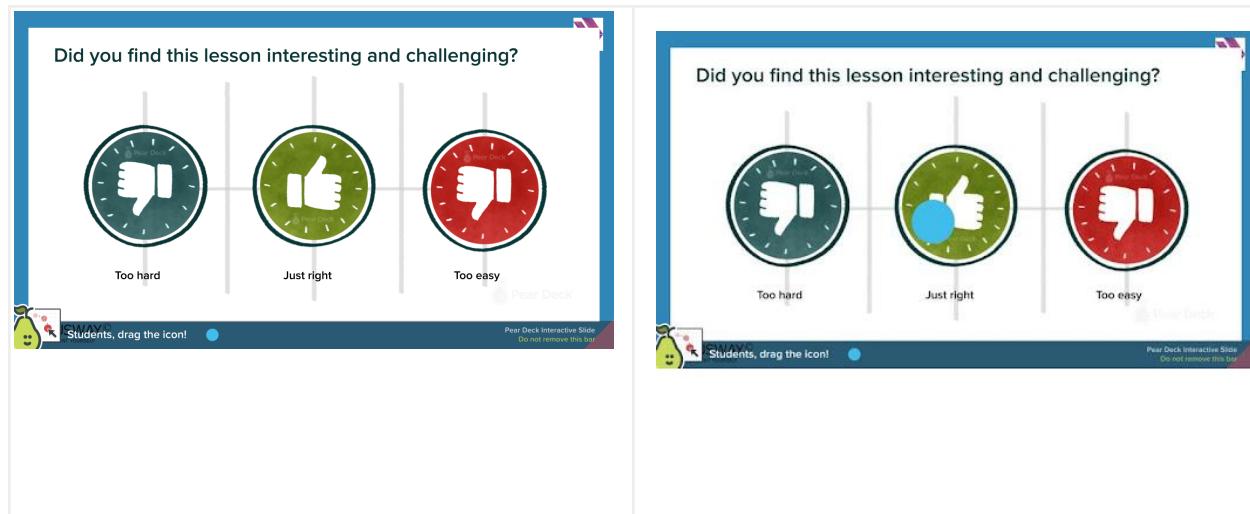
let's start the  
hands-on phase

15

Use this space to take notes:

Slide 16

Your Response



Use this space to take notes:

## Slide 17

# THANKS!

**Any questions?**

You can find us at:

steve\_w@clarusway.com  
michael\_g@clarusway.com

CLARUSWAY®  
way to success



17

Use this space to take notes: