

# Data Aggregation: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2019

## Syntax

---

### GROUPBY OBJECTS

- Create a GroupBy object:

```
df.groupby('col_to_groupby')
```

- Select one column from a GroupBy object:

```
df.groupby('col_to_groupby')['col_selected']
```

- Select multiple columns from a GroupBy object:

```
df.groupby('col_to_groupby')[['col_selected1', 'col_selected2']]
```

---

### COMMON AGGREGATION METHODS

- `mean()` : Calculates the mean of groups
- `sum()` : Calculates the sum of group values
- `size()` : Calculates the size of groups
- `count()` : Calculates the count of values in groups
- `min()` : Calculates the minimum of group values
- `max()` : Calculates the maximum of group values

---

### GROUPBY.AGG() METHOD

- Apply one function to a GroupBy object:

```
df.groupby('col_to_groupby').agg(function_name)
```

- Apply multiple functions to a GroupBy object:

```
df.groupby('col_to_groupby').agg([function_name1, function_name2, function_name3])
```

- Apply a custom function to a GroupBy object:

```
df.groupby('col_to_groupby').agg(custom_function)
```

---

## AGGREGATION WITH THE `DATAFRAME.PIVOT_TABLE` METHOD

- Apply only one function:

```
df.pivot_table(values='Col_to_aggregate', index='Col_to_group_by',  
aggfunc=function_name)
```

- Apply multiple functions:

```
df.pivot_table('Col_to_aggregate', 'Col_to_group_by', aggfunc=[function_name1,  
function_name2, function_name3])
```

- Aggregate multiple columns:

```
df.pivot_table(['Col_to_aggregate1', 'Col_to_aggregate2'], 'Col_to_group_by', aggfunc =  
function_name)
```

- Calculate the grand total for the aggregation column:

```
df.pivot_table('Col_to_aggregate', 'Col_to_group_by', aggfunc=function_name,  
margins=True)
```

## Concepts

- **Aggregation** is applying a statistical operation to groups of data. It reduces dimensionality so that the dataframe returned will contain just one value for each group. The aggregation process can be broken down into three steps:
  - Split the dataframe into groups.
  - Apply a function to each group.
  - Combine the results into one data structure.
- The `groupby` operation optimizes the split-apply-combine process. It can be broken down into two steps:
  - Create a GroupBy object.
  - Call an aggregation function.
- Creating the GroupBy object is an intermediate step that allows us to optimize our work. It contains information on how to group the dataframe, but nothing is actually computed until a function is called.

- The `DataFrame.pivot_table()` method can also be used to aggregate data. We can also use it to calculate the grand total for the aggregation column.



Takeaways by Dataquest Labs, Inc. - All rights reserved © 2019