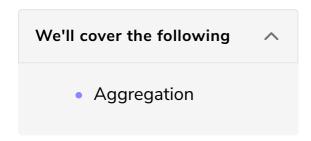
Data Aggregation

In this lesson, an explanation is provided on data aggregation and its various techniques.



The grouping and aggregating functions can sometimes seem similar, but they are actually consecutive steps in obtaining important insights about data. Data is first grouped using the <code>groupby()</code> clause explained in the previous lesson. Then, aggregating functions and techniques can be applied to fetch the required information from the data.

Aggregation

It is a process of applying operations on groups of data to extract useful insights. Let's understand this with an example. Some random animals are defined with hypothetical legs, weight, height, and protein values. Our task is to compute the average amount of legs, weight, height, and protein a certain animal class can have.

```
import numpy as np
                                                                                              C
import pandas as pd
import random
# Declaring a DataFrame with values
df = pd.DataFrame({'Animal_type':[random.choice(['Chicken','Duck', 'Goat', 'Turkey']) for i in range
                   'legs':[random.choice(range(1,4)) for i in range(1,16)],
                   'weight':[random.choice(range(10,20)) for i in range(1,16)],
                   'height':[random.choice(range(4,15)) for i in range(1,16)],
                   'protein':abs(np.random.randn(15)),
                    })
print("The Original DataFrame:")
print(df, '\n********
Aw = df.groupby('Animal_type') # Grouping with Animal_type column
# Computing mean of individual groups
print("Average properties an animal can have:")
print(Aw.agg('mean'), '\n******
```



On **line 14**, the data is first grouped and stored in the Aw variable. Then, on **line 17**, the agg('mean') function is used to calculate the *average* of all the property columns according to the specific animal class.

It can be seen from the output that we have clearly obtained how many <code>legs</code>, <code>weight</code>, <code>height</code>, and <code>protein</code> a certain animal can have on average. Keep in mind that these values are randomly generated, but the operations performed on it are completely viable. This is only one example. The <code>agg()</code> function provides a lot of other functionalities, some of which are listed in the table below.

Functionality	Description
max	Computes the max of the grouped values
min	Computes the min of the grouped values
last	Computes the last of the grouped values
first	Computes the first of the grouped values
sum	Computes the sum of the grouped values
size	Computes the size or length of the grouped values
count	Computes the count of the grouped values
var	Computes the variance of the grouped values

std	Computes the standard deviation of the grouped values
describe	Provides the descriptive statistics about the grouped values

More information regarding pandas *aggregate* function can be found here.

In the next lesson, the *split-apply-combine* technique of pandas is explained.