Ironhack Student Portal

preview.my.ironhack.com/lms/courses/course-

Lesson Goals

In this lesson we will learn to perform calculations on existing columns in a Pandas DataFrame and store them in a new column

Introduction

There are cases where we might want to augment our Pandas DataFrame with calculated columns. Pandas enables us to perform these calculations and easily store them in a new column.

Working with Constants

We can add new calculated columns using an existing column and a constant value.

Recall our animals dataset. We will use this dataset and create a new column that converts the body weight in pounds to kilograms.

```
import numpy as np
import pandas as pd

animals = pd.read_csv('./data/animals.csv')

animals['bodywtkg'] = animals['bodywt'] * 0.45359237
animals.head()
        brainwt     bodywt      animal          bodywtkg
0       3.385       44.500      Arctic_fox      20.184860
1       0.480       15.499      Owl_monkey      7.030228
2       1.350       8.100       Beaver          3.674098
3       464.983     423.012     Cow             191.875016
4       36.328      119.498     Gray_wolf       54.203381
```

Note that we used the `head` function to look at the first 5 rows for every column. We do this to confirm that the changes we made to the DataFrame worked as expected.

Combining Two (or More) Columns

We can perform calculations using a combination of two or more column. We write an equation that correctly refers to the columns in the DataFrame and assign the calculation to a new column.

For example, we can compute the ratio of body weight to brain weight for all animals in our data and assign this value to a new column.

```
animals['wtratio'] = animals['bodywt'] / animals['brainwt']
animals.head()
```

	brainwt	bodywt	animal	bodywtkg	wtratio
0	3.385	44.500	Arctic_fox	20.184860	13.146233
1	0.480	15.499	Owl_monkey	7.030228	32.289583
2	1.350	8.100	Beaver	3.674098	6.000000
3	464.983	423.012	Cow	191.875016	0.909736
4	36.328	119.498	Gray_wolf	54.203381	3.289419

Conditional Calculations

It is possible to perform more complex calculations. For example, you may have noticed that we used division in the previous example without checking whether the denominator is zero. This can cause quite a bit of problems. Therefore, we can introduce a condition in our assignment. If the brain weight is zero then the ratio will be zero, otherwise, store the ratio in the new column. We can create conditional functions using the `where` function in numpy. We pass 3 arguments to the function. The first argument is the condition, the second is the value in case the condition is true, and the third is the value in case the condition is false.

```
animals['wtratiozerocheck'] = np.where(animals['brainwt'] != 0, animals['bodywt'] / animals['brainwt'], 0)
animals.head()
```

	brainwt	bodywt	animal	bodywtkg	wtratio	wtratiozerocheck
0	3.385	44.500	Arctic_fox	20.184860	13.146233	13.146233
1	0.480	15.499	Owl_monkey	7.030228	32.289583	32.289583
2	1.350	8.100	Beaver	3.674098	6.000000	6.000000
3	464.983	423.012	Cow	191.875016	0.909736	0.909736
4	36.328	119.498	Gray_wolf	54.203381	3.289419	3.289419

Calculations Using Functions

As we have learned in a previous lesson, Pandas DataFrames have 3 components: rows, columns and data. The rows and columns are also called axes. Axis zero is the row axis and axis one is the column axis. Therefore, we can apply functions to the column axis in order to summarize all columns at once.

Let's say we want to take a sum of all numeric columns in the animals DataFrame. We can do this by using the sum function and passing `axis=1` as an argument to the function.

```
animals['sum'] = animals.sum(axis=1)
animals['sum']
0        94.362327
1        87.588395
2        25.124098
3      1081.689489
4       216.608218
          ...
57      407.773558
58       10.457118
59       32.265027
60       51.814904
61      101.297708
Name: sum, Length: 62, dtype: float64
```

Summary

In this lesson we learned different ways to create calculated columns. We computed a new column by combining existing data with a constant. We also computed a calculated column using two existing columns as well as using a conditional function to create a calculated column. Finally, we applied one function to all columns at once by specifying to apply the function to `axis=1` .