learn-co-curriculum / dsc-lambda-functions-lab Public

View license

0 stars 207 forks

Star Watch

Code Issues Pull requests 1 Actions Projects Security Insights

solution ...

This branch is 11 commits ahead, 19 commits behind master. Contribute

hoffm386 fix objectives spacing ... on Aug 1 13

View code

README.md

Lambda Functions - Lab

Introduction

In this lab, you'll get some hands-on practice creating and using lambda functions.

Objectives

In this lab you will:

- Create lambda functions to use as arguments of other functions
- Use the `.map()` or `.apply()` method to apply a function to a pandas series or DataFrame

Lambda Functions

```
import pandas as pd
df = pd.read_csv('Yelp_Reviews.csv', index_col=0)
df.head(2)
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	business_id	cool	date	funny	review_id
1	pomGBqfbxcqPv14c3XH-ZQ	0	2012-11-13	0	dDl8zu1vWPdKGihJrwQbpv
2	jtQARsP6P-LbkyjbO1qNGg	1	2014-10-23	1	LZp4UX5zK3e-c5ZGSeo3kA

Simple arithmetic

Use a lambda function to create a new column called `'stars_squared'` by squaring the stars column.

```
df['stars_squared'] = df['stars'].map(lambda x: x**2)
df.head(2)
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	business_id	cool	date	funny	review_id
1	pomGBqfbxcqPv14c3XH-ZQ	0	2012-11-13	0	dDI8zu1vWPdKGihJrwQbpv
2	jtQARsP6P-LbkyjbO1qNGg	1	2014-10-23	1	LZp4UX5zK3e-c5ZGSeo3kA

Dates

Select the month from the date string using a lambda function.

```
df['date'].map(lambda x: x[5:7]).head()
```

```
1     11
2     10
4     09
5     02
10    06
Name: date, dtype: object
```

What is the average number of words for a yelp review?

Do this with a single line of code!

```
df['text'].map(lambda x: len(x.split())).mean()
```

```
77.06551724137931
```

Create a new column for the number of words in the review

```
df['Review_num_words'] = df['text'].map(lambda x: len(x.split()))
df.head(2)
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	business_id	cool	date	funny	review_id

	business_id	cool	date	funny	review_id
1	pomGBqfbxcqPv14c3XH-ZQ	0	2012-11-13	0	dDl8zu1vWPdKGihJrwQbpv
2	jtQARsP6P-LbkyjbO1qNGg	1	2014-10-23	1	LZp4UX5zK3e-c5ZGSeo3kA

Rewrite the following as a lambda function

Create a new column `'Review_Length'` by applying this lambda function to the `'Review_num_words'` column.

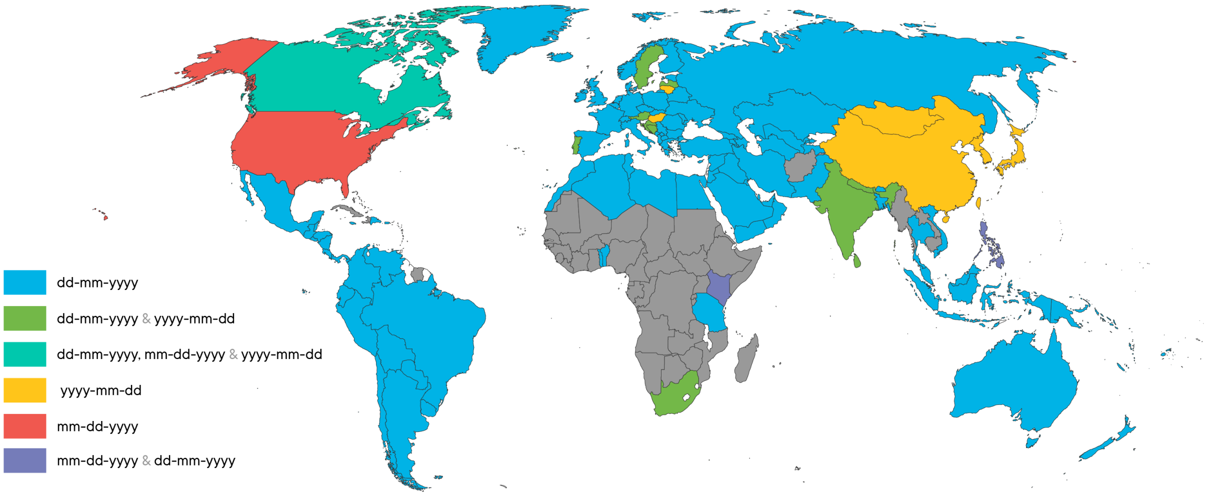
```
# Rewrite the following function as a lambda function
def rewrite_as_lambda(value):
    if len(value) > 50:
        return 'Short'
    elif len(value) < 80:
        return 'Medium'
    else:
        return 'Long'
# Hint: nest your if, else conditionals

df['Review_length'] = df['Review_num_words'].map(lambda x: 'Short' if x < 50 else ('
df['Review_length'].value_counts(normalize=True)
```

```
Short     0.493103
Long      0.294636
```

```
Medium    0.212261
Name: Review_length, dtype: float64
```

Level Up: Dates Advanced!


🟦 dd-mm-yyyy
🟩 dd-mm-yyyy & yyyy-mm-dd
🟩 dd-mm-yyyy, mm-dd-yyyy & yyyy-mm-dd
🟨 yyyy-mm-dd
🟥 mm-dd-yyyy
🟪 mm-dd-yyyy & dd-mm-yyyy

Print the first five rows of the `'date'` column.

```
df['date'].head()
```

```
1     2012-11-13
2     2014-10-23
4     2014-09-05
5     2011-02-25
10    2016-06-15
Name: date, dtype: object
```

Overwrite the `'date'` column by reordering the month and day from `YYYY-MM-DD` to `DD-MM-YYYY` . Try to do this using a lambda function.

```
df['date'] = df['date'].map(lambda x: '{}-{}-{}'.format(x[-2:], x[5:7], x[:4]))
df['date'].head()
```

```
1     13-11-2012
2     23-10-2014
4     05-09-2014
5     25-02-2011
10    15-06-2016
Name: date, dtype: object
```

Summary

Great! Hopefully, you're getting the hang of lambda functions now! It's important not to overuse them - it will often make more sense to define a function so that it's reusable elsewhere. But whenever you need to quickly apply some simple processing to a collection of data you have a new technique that will help you to do just that. It'll also be useful if you're reading someone else's code that happens to use lambdas.

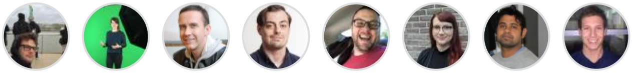
Releases

No releases published

Packages

No packages published

Contributors 8



Languages

● **Jupyter Notebook** 100.0%