Map, Apply, and Filter

Key Functions for Data Manipulation

Mpussa Doumbia, Ph.D.

Howard University

January 29, 2024

Introduction

- Data manipulation is fundamental in data analysis.
- Map, Apply, and Filter are essential tools for transforming data.

Map Function

- Applies a function to all items in an input list.
- Returns a map object.

```
# Define a function to square numbers
def square(x):
    return x * x

# Apply the square function to each element in the list
numbers = [1, 2, 3, 4]
squared = map(square, numbers)

# Convert the map object to a list (for visualization)
print(list(squared))
```

Output: [1, 4, 9, 16]

Applying Functions to Non-DataFrame Objects

Using map Function:

```
numbers = [1, 2, 3, 4]

# Define a function to square numbers
def square(x):
    return x * x

# Apply the square function to each element
squared_numbers = map(square, numbers)

# Convert to list for display
print(list(squared_numbers))
```

Output: [1, 4, 9, 16]

Applying Functions to Non-DataFrame Objects (cont.)

Using List Comprehension:

```
numbers = [1, 2, 3, 4]

# Using list comprehension to square numbers
squared_numbers = [x * x for x in numbers]

print(squared_numbers)
```

Output: [1, 4, 9, 16]

Applying Functions to Non-DataFrame Objects (cont.)

Using apply from the toolz library:

```
from toolz.functoolz import apply

# Define a function to add two numbers
def add(x, y):
    return x + y

# Apply the add function to a tuple of numbers
result = apply(add, (1, 2))

print(result)
```

Output: 3

Apply Function (Pandas)

- Applies a function along an axis of the DataFrame or Series.

```
import pandas as pd

# Create a DataFrame
df = pd.DataFrame({
    'A': [1, 2, 3],
    'B': [10, 20, 30]
})

# Define a function to add a constant to each element
def add_five(x):
    return x + 5

# Apply function to each element of the DataFrame
df = df.applymap(add_five)
print(df)
```

Output:

```
   A   B
0  6  15
```

Filter Function

- Constructs an iterator from elements of an iterable for which a function returns true.

```
# Define a function to check if a number is even
def is_even(x):
    return x % 2 == 0

# Filter out only even numbers from the list
numbers = [1, 2, 3, 4, 5, 6]
even_numbers = filter(is_even, numbers)

# Convert the filter object to a list (for visualization)
print(list(even_numbers))
```

Output: [2, 4, 6]

Conclusion

- Understanding and utilizing Map, Apply, and Filter are crucial for efficient data manipulation.
- These functions enable concise and readable code, important for large-scale data analysis.

QA

Any Questions?