How to combine together Pandas methods

SHARP SIGHT

What you'll learn

- How to "chain" commands together

- How to perform complex data manipulation
  - use multiple Pandas methods

The syntax for "chaining" Pandas methods

It's possible to "chain" Pandas methods together

- Similar to:
  - pipes in Unix
  - dplyr pipes in R

- Allows you to send the output of a command to the input of another

Syntax: Pandas method chains

```
(bank
   .query("loan == 'no'")
   .groupby('education')
   .agg('mean')
   .filter(['age','balance'])
)
```

Here we are 'chaining' together multiple Pandas methods.

The output of one method becomes the input of the next.

To chain methods together, enclose the operations in parenthesis

The whole expression is enclosed by parenthesis

```
(bank
   .query("loan == 'no'")
   .groupby('education')
   .agg('mean')
   .filter(['age','balance'])
)
```

Note: The parenthesis are necessary if you write the syntax this way, with the methods on separate lines

You can read each line as as “then”

```
(bank
  .query("loan == 'no'")
  .groupby('education')
  .agg('mean')
  .filter(['age','balance'])
)
```

Take the bank dataset .....THEN
keep rows where loan is 'no’ .....THEN
group by “education” .....THEN
compute the mean by group .....THEN
keep only the mean age and balance

Example: chaining multiple methods together to analyze bank data

```
(bank
   .query("loan == 'no'")
   .groupby('education')
   .agg('mean')
   .filter(['age','balance'])
)
```

education	mean age	mean balance
primary	47	1551
secondary	40	1288
tertiary	40	1845
unknown	45	1602

We can perform fairly complex data manipulation with this technique

- You can chain together many methods to get different effects

- Also good for data analysis
  - i.e., group and aggregate

Recap

Key Takeaways

- You can use the combine methods together in a "chain"
  - wrap the syntax in parenthesis
  - different methods can be on different lines

- Combining syntax is excellent for complex data wrangling

- Also very useful for data analysis