CSCI 1010 Class 2

Profs. Michael Linderman and Phil Chodrow
Department of Computer Science
Middlebury College

Choose the best match each Pandas row accessor (left) to its primary behavior (right).

Accessor:

1. `.loc`
2. `.at`
3. `.iloc`
4. `.iat`

Behavior:

i. Vector access by integer position (like Python slice semantics)

ii. Vector access by label (end inclusive)

iii. Scalar access by label for a single value

iv. Scalar access by integer position for a single value

Options:

a. 1-i, 2-ii, 3-iii, 4-iv

b. 1-ii, 2-iii, 3-i, 4-iv

c. 1-ii, 2-i, 3-iii, 4-iv

Slide 1 Notes

Answer: B

Recall that `loc` is for label indexing and `iloc` for positional indexing. Although we didn't discuss `at`, we can make inferences based the naming.

Assume `df` is a Pandas DataFrame where “A” is a column of strings. What does this code do?

```
# lambda x: len(x) > 3 creates an anonymous function with
# a single argument
df.groupby('A').filter(lambda x: len(x) > 3)
```

a. Keep rows where the value in column “A” has more than 3 characters

b. Removes groups where column “A” has more than 3 unique values

c. Filters each group to keep the first 3 rows

d. Returns only the groups that have more than 3 rows

Slide 2 Notes

Answer: D

Recall that `groupby` returns a view into the underlying data. The aggregation functions are applied to those groups. So here `len` is reporting the size of the groups, not of the underlying value.

Assume `df` is a Pandas DataFrame where “A” and “B” are columns of strings. Approximately how many rows do you expect in the resulting DataFrame? Assume the number of unique values in each column are greater than 1, i.e., $|A| > 1$ and $|B| > 1$.

```
1 df.groupby(['A', 'B']).mean()
```

a. 2

b. $|A|$

c. $\max(|A|, |B|)$

d. $|A| + |B|$

e. $|A| \cdot |B|$

Ross single
double

Coffrin single
double

count
10
20

Slide 3 Notes

Answer: E

`groupby` with multiple columns creates entries for each combination of values observed across those columns. That will approximate be the product of the cardinalities of the grouping columns. Must it be the product? No. Not all combinations may be observed in the data, and depending on the data types and the arguments supplied to `groupby`, we might not get unobserved combinations in the output.