Trong pandas, iloc và loc là hai phương thức quan trọng để truy cập dữ liệu trong DataFrame/Series, nhưng chúng hoạt động khác nhau về cách **chỉ mục (indexing)**. Dưới đây là bảng so sánh chi tiết:

| **Tiêu chí** | .loc | .iloc |
| --- | --- | --- |
| **Cơ chế** | Truy cập bằng **label (nhãn)** của index/column. | Truy cập bằng **vị trí số nguyên (integer position)** của index/column. |
| **Phạm vi slice** | **Bao gồm cả endpoint** (ví dụ: a:b sẽ lấy từ a đến b). | **Không bao gồm endpoint** (giống Python list, ví dụ: a:b lấy từ a đến b-1). |
| **Boolean Indexing** | Hỗ trợ (ví dụ: df.loc[df['column'] > 5]). | Không hỗ trợ trực tiếp, phải kết hợp với mảng boolean. |
| **Index phức tạp** | Làm việc tốt với index không phải số (ví dụ: chuỗi, ngày tháng). | Bỏ qua label, chỉ quan tâm đến vị trí thứ tự. |
| **Kết hợp row/column** | Cú pháp: df.loc[row\_labels, column\_labels]. | Cú pháp: df.iloc[row\_positions, column\_positions]. |

**Ví dụ minh họa**:

Giả sử có DataFrame sau:

python

Copy

Download

import pandas as pd

data = {

"A": [10, 20, 30, 40],

"B": ["X", "Y", "Z", "W"]

}

df = pd.DataFrame(data, index=["row1", "row2", "row3", "row4"])

**1. Truy cập bằng**.loc**(theo label)**:

* Lấy hàng có index là "row2":

python

Copy

Download

df.loc["row2"] # Output: A=20, B='Y'

* Lấy hàng từ "row2" đến "row4" và cột "A":

python

Copy

Download

df.loc["row2":"row4", "A"] # Output: row2:20, row3:30, row4:40

* Boolean indexing:

python

Copy

Download

df.loc[df["A"] > 25] # Lấy các hàng có cột A > 25

**2. Truy cập bằng**.iloc**(theo vị trí số)**:

* Lấy hàng ở vị trí 1 (hàng thứ 2):

python

Copy

Download

df.iloc[1] # Output: A=20, B='Y'

* Lấy hàng từ vị trí 1 đến 3 (không bao gồm 3):

python

Copy

Download

df.iloc[1:3] # Output: row2 và row3

* Lấy cột ở vị trí 0 (cột A):

python

Copy

Download

df.iloc[:, 0] # Output: Tất cả hàng của cột A

**Khi nào dùng**.loc**vs**.iloc:

* Dùng .loc khi:
  + Làm việc với index có **label rõ ràng** (ví dụ: chuỗi, ngày tháng).
  + Cần truy vấn dữ liệu bằng **điều kiện logic** (boolean indexing).
* Dùng .iloc khi:
  + Chỉ quan tâm đến **vị trí thứ tự** của dữ liệu.
  + Làm việc với index số nguyên đơn giản (giống list).

**Lưu ý**:

* Nếu index của DataFrame là số nguyên, việc dùng loc với số nguyên vẫn hợp lệ, nhưng iloc sẽ luôn hiểu là **vị trí thứ tự**.
* Tránh nhầm lẫn giữa slice trong .loc (bao gồm endpoint) và .iloc (không bao gồm endpoint).

**Ví dụ nhầm lẫn phổ biến**:

python

Copy

Download

# Giả sử index là số nguyên [0, 1, 2, 3]

df.loc[0:2] # Lấy hàng có label 0, 1, 2

df.iloc[0:2] # Lấy hàng ở vị trí 0 và 1 (không bao gồm 2)

Hy vọng qua ví dụ này, bạn đã phân biệt rõ cách sử dụng loc và iloc! 🚀