

レシピ°30 スカラー選択の高速化（P81）

- スカラーインデクサ `.iat`、`.at` でより高速にスカラー要素を取得できる
- `iloc`や`loc`よりどれだけ高速化を比較する

(1) 校名をインデックスにしてcollegeデータセットを読み込む。`.loc` に校名とカラム名を渡してスカラー値を取得する

```
In [14]: college = pd.read_csv('college.csv', index_col='INSTNM')
cn = 'Texas A & M University-College Station'
college.loc[cn, 'UGDS_WHITE']

Out[14]: 0.6609999999999999
```

(2) 同じ結果を `.at` で取得する

```
In [15]: college.at[cn, 'UGDS_WHITE']

Out[15]: 0.6609999999999999
```

(3) 両者の速度を比較する（`.loc` vs `.at`）

```
In [18]: %timeit college.loc[cn, 'UGDS_WHITE']

5.98 μs ± 97.2 ns per loop (mean ± std. dev. of 7 runs, 100000 loops each)

In [19]: %timeit college.at[cn, 'UGDS_WHITE']

3.29 μs ± 47.8 ns per loop (mean ± std. dev. of 7 runs, 100000 loops each)
```

- 結果、`.at` が約45%高速

(4) 今度は `.iolc` と `.iat` の速度を比較する

```
In [28]: row_num = college.index.get_loc(cn)
col_num = college.columns.get_loc('UGDS_WHITE')
row_num, col_num

Out[28]: (3765, 10)

In [29]: %timeit college.iloc[row_num, col_num]

19.5 μs ± 346 ns per loop (mean ± std. dev. of 7 runs, 10000 loops each)

In [30]: %timeit college.iat[row_num, col_num]

17.5 μs ± 790 ns per loop (mean ± std. dev. of 7 runs, 10000 loops each)
```

- 結果、`.iat` が約10%高速
- スカラーインデクサが速い。大して速くないと思うかもしれないがプログラムでスカラー選択が繰り返される場合、大きな違いになる

(補足) `.iat`、`.at` はSeriesにも使える。スカラー値を渡せば、スカラーが返される

```
In [32]: state = college['STABBR']
state.iat[1000]

Out[32]: 'IL'

In [33]: state.at['Stanford University']

Out[33]: 'CA'
```