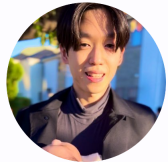


read_csv コードリーディング



@kazukiigeta

read_csvを呼び出し後の内部動作

1. 渡された実引数の中身を**チェック**する
2. 引数で指定された"エンジン"でファイルやバッファを**読み取る**
3. `pd.DataFrame` を**作成**する
4. 条件によって `pd.DataFrame` から `pd.Series` を**作成**する
5. `pd.DataFrame` または `pd.Series` を**返す**



1. 渡された実引数の中身をチェックする



引数は全部で52個もある

```
def read_csv(  
    filepath_or_buffer: FilePath | ReadCsvBuffer[bytes] | ReadCsvBuffer[str],  
    sep=lib.no_default,  
    delimiter=None,  
    # Column and Index Locations and Names  
    header="infer",  
    names=lib.no_default,  
    index_col=None,  
    usecols=None,  
    squeeze: bool | None = None,  

```

引数の定義はまだまだ続く



引数チェックの概要

1. 引数の値の組合せから矛盾を検知してエラーを起こす
2. ?



引数の値の組合せから矛盾を検知してエラーを起こす

```
if delimiter and (sep is not lib.no_default):  
    raise ValueError(  
        "Specified a sep and a delimiter; you can only specify one.")
```

この例は、 `sep` と `delimiter` の両方が
引数で指定されていたらエラーを起こす



2. 引数で指定された"エンジン"でファイル やバッファを**読み取る**



3. `pd.DataFrame` を作成する



4. `pd.DataFrame` または `pd.Series` を返す



pd.Series を返す条件

- squeeze 引数が True &
- pd.DataFrame カラム数が 1

このとき、生成した pd.DataFrame を
pd.Series に変換して返す（v1.4.0から非推奨）

それ以外の場合、pd.DataFrame を返す



Appendix

沢山の引数を別の関数に渡すときに便利な方法



引数が多い関数は `locals()`

[pandas/io/parsers/readers.py#L908](#)

```
kwds = locals().copy()
```

`locals()` 呼び出し個所のローカル変数を
辞書型で取得できる

以下の様に、`read_csv()` の数十個の引数を全て書かずに、`kwds` の形で渡すことができるようになる

```
return _read(lepath_or_buffer, kwds)
```

おわり

