what’s new and awesome in pandas
pandas?

```
In [13]: foo
Out[13]:
        methyl1    age  edu    something  indic
   0    38.36  30to39  geCollege     1    False
   1    37.85  lt30   geCollege     1    False
   2    38.57  30to39  geCollege     1    False
   3    39.75  30to39  geCollege     1     True
   4    43.83  30to39  geCollege     1     True
   5    39.08  30to39   ltHS        1     True
```

Size-mutable “labeled arrays” that can handle heterogeneous data
Kinda like a structured array??

- Automatic data alignment with lots of reshaping and indexing methods

- Implicit and explicit handling of missing data

- Easy time series functionality
  - Far less fuss than scikits.timeseries

- Lots of in-memory SQL-like operations (group by, join, etc.)
pandas?

- Extremely good for financial data
  - StackOverflow: “this is a beast of a financial analysis tool”

- One of the better relational data munging tools in any language?

- But also has maybe 60+% of what R users expect when they come to Python
1. Heavily redesigned internals

• Merged old DataFrame and DataMatrix into a single DataFrame: retain optimal performance where possible

• Internal BlockManager class manages homogeneous ndarrays for optimal performance and reshaping
1. Heavily redesigned internals

- Better handling of missing data for non-floating point dtypes

- Soon: DataFrame variant with N-dim “hyperslabs”
2. Fancier indexing

Mix boolean / integer / label / slice-based indexing

\[
df.\text{ix}[0]
df.\text{ix}[\text{date1:}\text{date2}]
df.\text{ix}[:5, \text{"A":}\text{"F"}]
\]

Setting works too

\[
df.\text{ix}[df[\text{"A"}] > 0, [\text{"B"}, \text{"C"}, \text{"D"}]] = \text{nan}
\]
3. More robust IO

data_frame = read_csv('mydata.csv')
data_frame2 = read_table('mydata.txt', sep='\t',
                         skiprows=[1,2],
                         na_values=['#N/A NA'])

store = HDFStore('pytables.h5')
store['a'] = data_frame
store['b'] = data_frame2
### 4. Better pivoting / reshaping

<table>
<thead>
<tr>
<th></th>
<th>foo</th>
<th>bar</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>one</td>
<td>a</td>
<td>-0.0524</td>
<td>1.664</td>
<td>1.171</td>
</tr>
<tr>
<td>1</td>
<td>one</td>
<td>a</td>
<td>0.2514</td>
<td>0.8306</td>
<td>-1.396</td>
</tr>
<tr>
<td>2</td>
<td>one</td>
<td>b</td>
<td>0.1256</td>
<td>0.3897</td>
<td>0.5227</td>
</tr>
<tr>
<td>3</td>
<td>one</td>
<td>b</td>
<td>-0.9301</td>
<td>0.6513</td>
<td>-0.2313</td>
</tr>
<tr>
<td>4</td>
<td>one</td>
<td>c</td>
<td>2.037</td>
<td>1.938</td>
<td>-0.3454</td>
</tr>
<tr>
<td>5</td>
<td>two</td>
<td>a</td>
<td>0.2073</td>
<td>0.7857</td>
<td>0.9051</td>
</tr>
<tr>
<td>6</td>
<td>two</td>
<td>a</td>
<td>-1.032</td>
<td>-0.8615</td>
<td>1.028</td>
</tr>
<tr>
<td>7</td>
<td>two</td>
<td>b</td>
<td>-0.7319</td>
<td>-1.846</td>
<td>0.9294</td>
</tr>
<tr>
<td>8</td>
<td>two</td>
<td>b</td>
<td>0.1004</td>
<td>-1.19</td>
<td>0.6043</td>
</tr>
<tr>
<td>9</td>
<td>two</td>
<td>c</td>
<td>-1.008</td>
<td>-0.3339</td>
<td>0.09522</td>
</tr>
</tbody>
</table>
4. Better pivoting / reshaping

In [29]: pivoted = df.pivot('bar', 'foo')

In [30]: pivoted['B']
Out[30]:
    one  two
  a  1.664  0.7857
  b  0.8306 -0.8615
  c  0.3897 -1.846
  d  0.6513 -1.19
  e  1.938  -0.3339
4. Better pivoting / reshaping

In [31]: pivoted.major_xs('a')
Out[31]:
   A   B   C
one -0.0524 1.664 1.171
two  0.2073 0.7857 0.9051

In [32]: pivoted.minor_xs('one')
Out[32]:
   A   B   C
   a -0.0524 1.664 1.171
   b  0.2514 0.8306 -1.396
   c  0.1256 0.3897 0.5227
d -0.9301 0.6513 -0.2313
e  2.037  1.938 -0.3454
4. Better pivoting / reshaping

In [30]: pivoted['B']
Out[30]:
    one    two  
  a  1.664  0.7857
  b  0.8306 -0.8615
  c  0.3897 -1.846
  d  0.6513 -1.19
  e  1.938  -0.3339
4. Some other things

- “Sparse” (mostly NA) versions of data structures
- Time zone support in DateRange
- Generic moving window function rolling_apply
Near future

• More powerful Group By

• Flexible, fast frequency (time series) conversions

• More integration with statsmodels
Thanks!

• Hack: github.com/wesm/pandas

• Twitter: @wesmckinn

• Blog: blog.wesmckinney.com