pandas-dfquery-demo

March 24, 2015

0.1 pandas-dfquery

Provides keyword-style queries on Pandas DataFrames – see examples below.

0.2 Why?

Ever got tired of writing code like this:

```
# standard subsetting syntax
df[df.YEAR == 2015 & df.MONTH == 1]
df[df.YEAR == 2015 & df.PRODUCT.str.contains('Fab')]
# .query() style
df.query('YEAR==2015 & MONTH==1')
# -- uups, string functions raise an exception (Node call not implemented)
df.query('df.YEAR == 2015 & df.PRODUCT.str.contains("Fab")'
```

and wish you could instead write:

```
df.query(YEAR=2015, MONTH=1)
df.query(PRODUCT__contains='Fab')
```

Then pandas-dfquery is for you. See the tutorial below.

0.3 Tutorial

```
In [1]: from dfquery import QDataFrame, Q, Filter
        import pandas as pd
        import numpy as np

        # basic filtering
        iris = QDataFrame(pd.read_csv('https://raw.github.com/pydata/pandas/master/pandas/tests/data/ir
        df = iris.query(SepalLength__gte=6.0, Name__contains='versicolor')
        df
```

Out[1]:

	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
50	7.0	3.2	4.7	1.4	Iris-versicolor
51	6.4	3.2	4.5	1.5	Iris-versicolor
52	6.9	3.1	4.9	1.5	Iris-versicolor
54	6.5	2.8	4.6	1.5	Iris-versicolor
56	6.3	3.3	4.7	1.6	Iris-versicolor
58	6.6	2.9	4.6	1.3	Iris-versicolor
62	6.0	2.2	4.0	1.0	Iris-versicolor
63	6.1	2.9	4.7	1.4	Iris-versicolor
65	6.7	3.1	4.4	1.4	Iris-versicolor
68	6.2	2.2	4.5	1.5	Iris-versicolor

71	6.1	2.8	4.0	1.3	Iris-versicolor
72	6.3	2.5	4.9	1.5	Iris-versicolor
73	6.1	2.8	4.7	1.2	Iris-versicolor
74	6.4	2.9	4.3	1.3	Iris-versicolor
75	6.6	3.0	4.4	1.4	Iris-versicolor
76	6.8	2.8	4.8	1.4	Iris-versicolor
77	6.7	3.0	5.0	1.7	Iris-versicolor
78	6.0	2.9	4.5	1.5	Iris-versicolor
83	6.0	2.7	5.1	1.6	Iris-versicolor
85	6.0	3.4	4.5	1.6	Iris-versicolor
86	6.7	3.1	4.7	1.5	Iris-versicolor
87	6.3	2.3	4.4	1.3	Iris-versicolor
91	6.1	3.0	4.6	1.4	Iris-versicolor
97	6.2	2.9	4.3	1.3	Iris-versicolor

In [14]:
```
# create Q objects as query terms, which are combinable by logical &, |
q_versi = Q(SepalLength__lt=6.0, Name__contains='versi')
q_setosa = Q(SepalLength__lt=6.0, Name__contains='setosa')
iris.query(q_versi & ~q_setosa)
```

Out[14]:

	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
53	5.5	2.3	4.0	1.3	Iris-versicolor
55	5.7	2.8	4.5	1.3	Iris-versicolor
57	4.9	2.4	3.3	1.0	Iris-versicolor
59	5.2	2.7	3.9	1.4	Iris-versicolor
60	5.0	2.0	3.5	1.0	Iris-versicolor
61	5.9	3.0	4.2	1.5	Iris-versicolor
64	5.6	2.9	3.6	1.3	Iris-versicolor
66	5.6	3.0	4.5	1.5	Iris-versicolor
67	5.8	2.7	4.1	1.0	Iris-versicolor
69	5.6	2.5	3.9	1.1	Iris-versicolor
70	5.9	3.2	4.8	1.8	Iris-versicolor
79	5.7	2.6	3.5	1.0	Iris-versicolor
80	5.5	2.4	3.8	1.1	Iris-versicolor
81	5.5	2.4	3.7	1.0	Iris-versicolor
82	5.8	2.7	3.9	1.2	Iris-versicolor
84	5.4	3.0	4.5	1.5	Iris-versicolor
88	5.6	3.0	4.1	1.3	Iris-versicolor
89	5.5	2.5	4.0	1.3	Iris-versicolor
90	5.5	2.6	4.4	1.2	Iris-versicolor
92	5.8	2.6	4.0	1.2	Iris-versicolor
93	5.0	2.3	3.3	1.0	Iris-versicolor
94	5.6	2.7	4.2	1.3	Iris-versicolor
95	5.7	3.0	4.2	1.2	Iris-versicolor
96	5.7	2.9	4.2	1.3	Iris-versicolor
98	5.1	2.5	3.0	1.1	Iris-versicolor
99	5.7	2.8	4.1	1.3	Iris-versicolor

In [17]:
```
# create Q objects as query terms, which are combinable by logical &, |
q_versi = Q(SepalLength__gt=6.0, Name__contains='versi')
q_setosa = Q(SepalLength__lt=6.0, Name__contains='setosa')
iris.query(q_versi | q_setosa)
```

Out[17]:

	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
0	5.1	3.5	1.4	0.2	Iris-setosa

1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa
10	5.4	3.7	1.5	0.2	Iris-setosa
11	4.8	3.4	1.6	0.2	Iris-setosa
12	4.8	3.0	1.4	0.1	Iris-setosa
13	4.3	3.0	1.1	0.1	Iris-setosa
14	5.8	4.0	1.2	0.2	Iris-setosa
15	5.7	4.4	1.5	0.4	Iris-setosa
16	5.4	3.9	1.3	0.4	Iris-setosa
17	5.1	3.5	1.4	0.3	Iris-setosa
18	5.7	3.8	1.7	0.3	Iris-setosa
19	5.1	3.8	1.5	0.3	Iris-setosa
20	5.4	3.4	1.7	0.2	Iris-setosa
21	5.1	3.7	1.5	0.4	Iris-setosa
22	4.6	3.6	1.0	0.2	Iris-setosa
23	5.1	3.3	1.7	0.5	Iris-setosa
24	4.8	3.4	1.9	0.2	Iris-setosa
25	5.0	3.0	1.6	0.2	Iris-setosa
26	5.0	3.4	1.6	0.4	Iris-setosa
27	5.2	3.5	1.5	0.2	Iris-setosa
28	5.2	3.4	1.4	0.2	Iris-setosa
29	4.7	3.2	1.6	0.2	Iris-setosa
..	...	...	...	...	...
40	5.0	3.5	1.3	0.3	Iris-setosa
41	4.5	2.3	1.3	0.3	Iris-setosa
42	4.4	3.2	1.3	0.2	Iris-setosa
43	5.0	3.5	1.6	0.6	Iris-setosa
44	5.1	3.8	1.9	0.4	Iris-setosa
45	4.8	3.0	1.4	0.3	Iris-setosa
46	5.1	3.8	1.6	0.2	Iris-setosa
47	4.6	3.2	1.4	0.2	Iris-setosa
48	5.3	3.7	1.5	0.2	Iris-setosa
49	5.0	3.3	1.4	0.2	Iris-setosa
50	7.0	3.2	4.7	1.4	Iris-versicolor
51	6.4	3.2	4.5	1.5	Iris-versicolor
52	6.9	3.1	4.9	1.5	Iris-versicolor
54	6.5	2.8	4.6	1.5	Iris-versicolor
56	6.3	3.3	4.7	1.6	Iris-versicolor
58	6.6	2.9	4.6	1.3	Iris-versicolor
63	6.1	2.9	4.7	1.4	Iris-versicolor
65	6.7	3.1	4.4	1.4	Iris-versicolor
68	6.2	2.2	4.5	1.5	Iris-versicolor
71	6.1	2.8	4.0	1.3	Iris-versicolor
72	6.3	2.5	4.9	1.5	Iris-versicolor
73	6.1	2.8	4.7	1.2	Iris-versicolor
74	6.4	2.9	4.3	1.3	Iris-versicolor
75	6.6	3.0	4.4	1.4	Iris-versicolor

```
76          6.8         2.8          4.8         1.4  Iris-versicolor
77          6.7         3.0          5.0         1.7  Iris-versicolor
86          6.7         3.1          4.7         1.5  Iris-versicolor
87          6.3         2.3          4.4         1.3  Iris-versicolor
91          6.1         3.0          4.6         1.4  Iris-versicolor
97          6.2         2.9          4.3         1.3  Iris-versicolor

[70 rows x 5 columns]
```

In [13]:
```
# note you can apply query objects to other dataframes, too
versi_l7 = df.query(~q_setosa & Q(SepalLength__gte=7.0))
versi_l7
```

Out[13]:

	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
50	7	3.2	4.7	1.4	Iris-versicolor

In [27]:
```
# lazy evaluation -- query() returns self instead of a new dataframe
# calls to .query() build up a filter object which is only evaluated
# on repr() or when accesing the .value property
df = QDataFrame(iris).lazy()
df.query(~Q(Name__contains='versicolor') & ~Q(Name__contains='setosa'))
df.query(SepalLength=5.8)
df.value
```

Out[27]:

	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
101	5.8	2.7	5.1	1.9	Iris-virginica
114	5.8	2.8	5.1	2.4	Iris-virginica
142	5.8	2.7	5.1	1.9	Iris-virginica