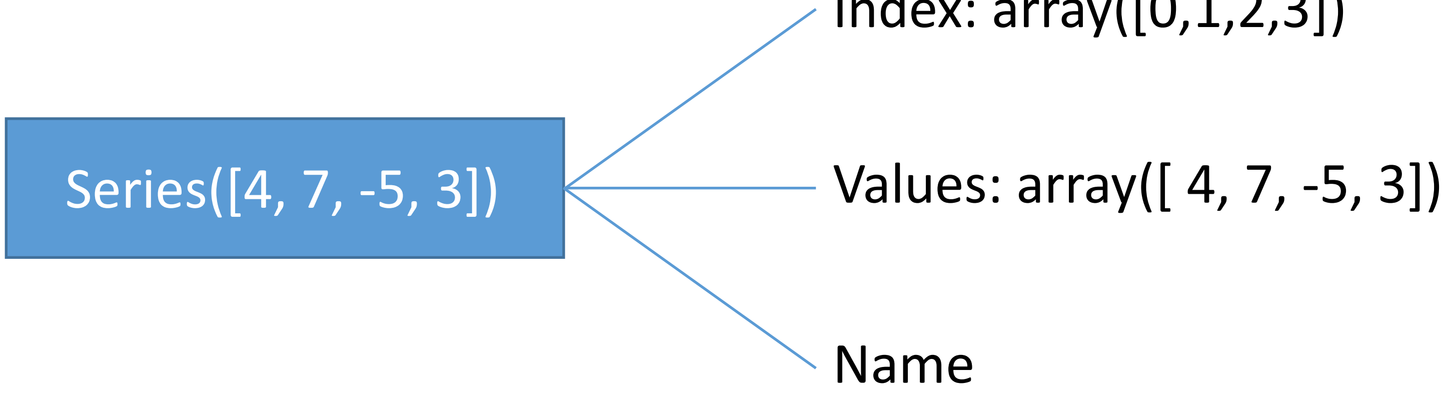
More Data Wrangling

Agenda

- Pandas
- Finance
- Missing Values
- Group Assignment
- Input / Output

Pandas

Series : pandas 1-D vectors



Series([4, 7, -5, 3])

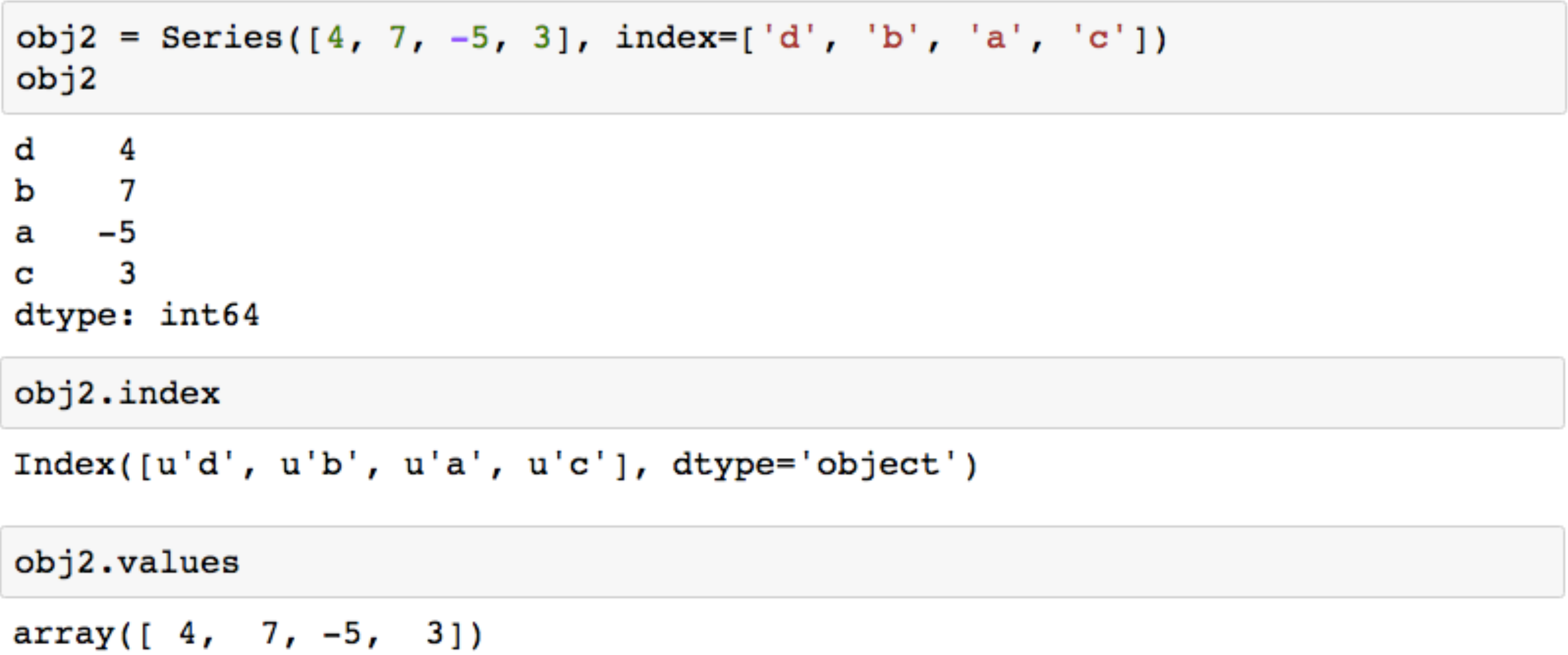
Index: array([0,1,2,3])

Values: array([4, 7, -5, 3])

Name

Series: Index, Values

2 main Series attribues: Index, Values



```
obj2 = Series([4, 7, -5, 3], index=['d', 'b', 'a', 'c'])
obj2
```

```
d    4
b    7
a   -5
c    3
dtype: int64
```

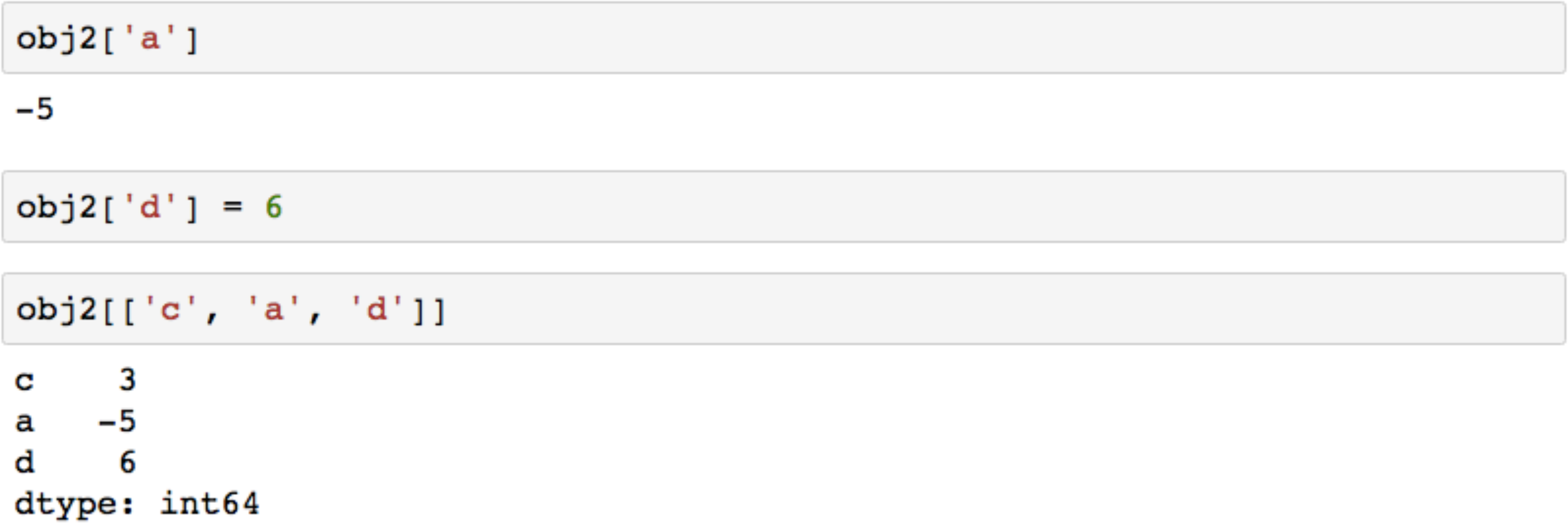
```
obj2.index
```

```
Index([u'd', u'b', u'a', u'c'], dtype='object')
```

```
obj2.values
```

```
array([ 4,  7, -5,  3])
```

Series: element selection



```
obj2['a']
```

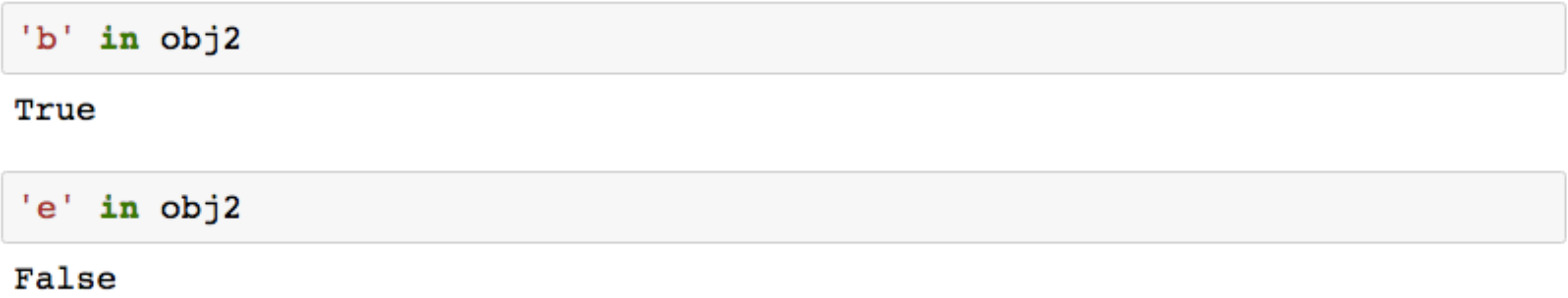
```
-5
```

```
obj2['d'] = 6
```

```
obj2[['c', 'a', 'd']]
```

```
c    3
a   -5
d    6
dtype: int64
```

Series: membership



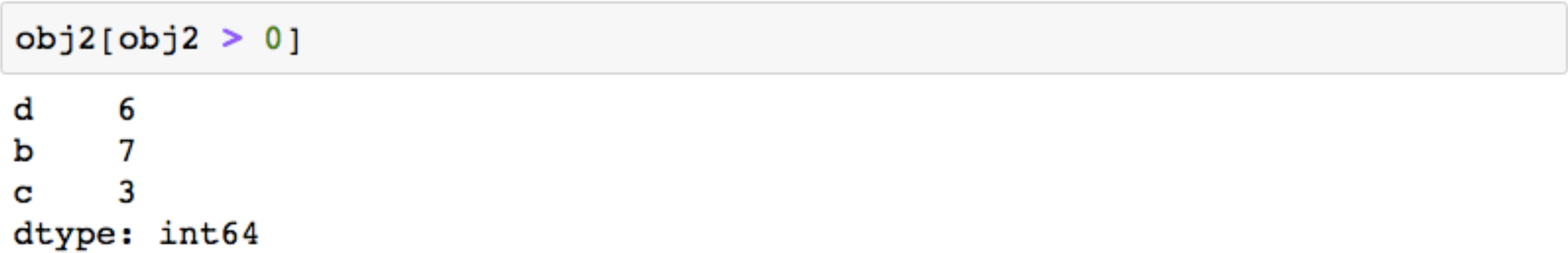
```
'b' in obj2
```

```
True
```

```
'e' in obj2
```

```
False
```

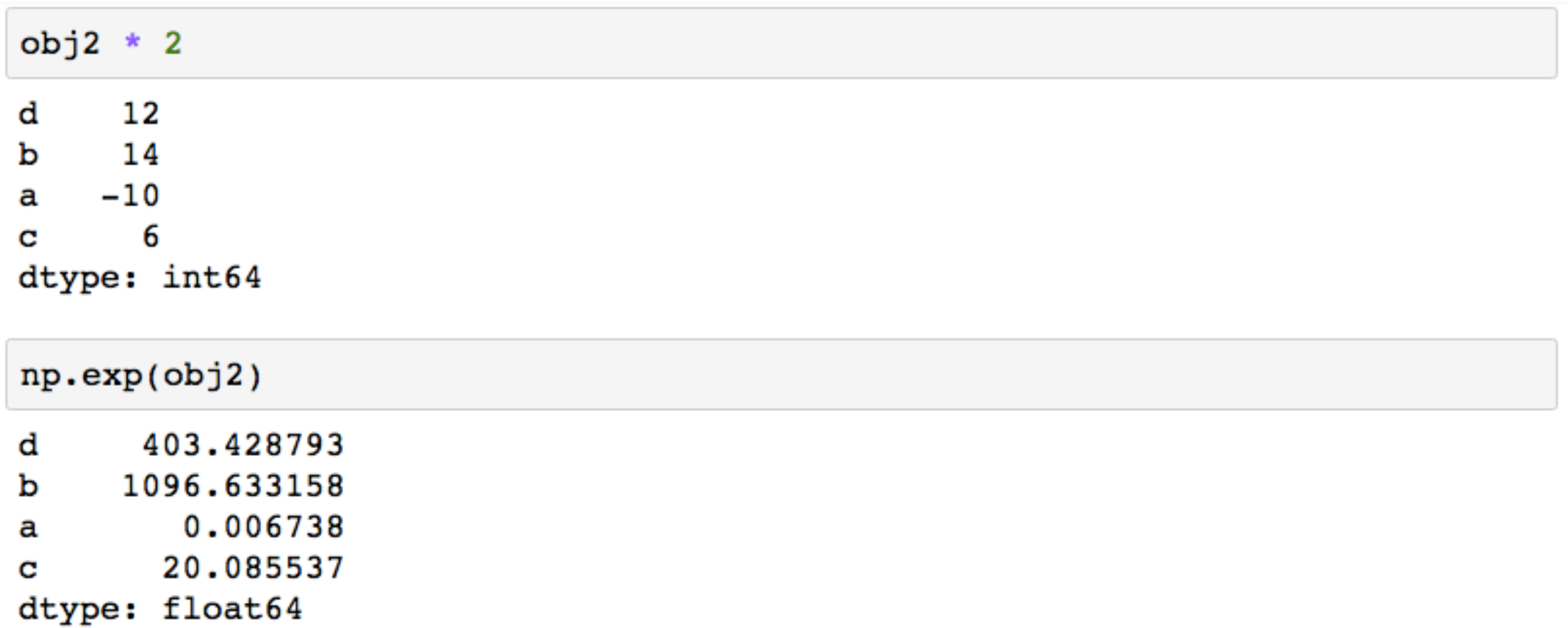
Series: element filtering



```
obj2[obj2 > 0]
```

```
d     6
b     7
c     3
dtype: int64
```

Series: scalar operations



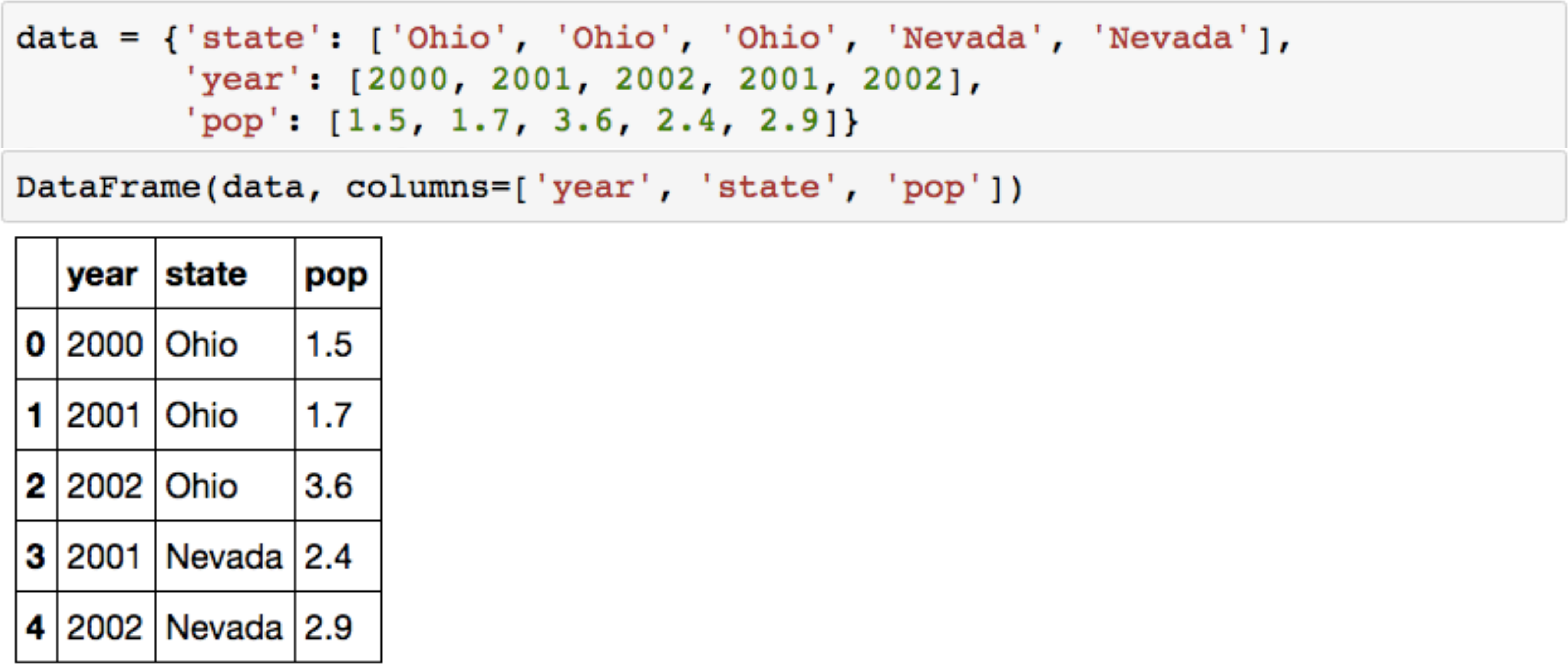
```
obj2 * 2
```

```
d    12
b    14
a   -10
c     6
dtype: int64
```

```
np.exp(obj2)
```

```
d     403.428793
b    1096.633158
a       0.006738
c      20.085537
dtype: float64
```

DataFrame: table in pandas

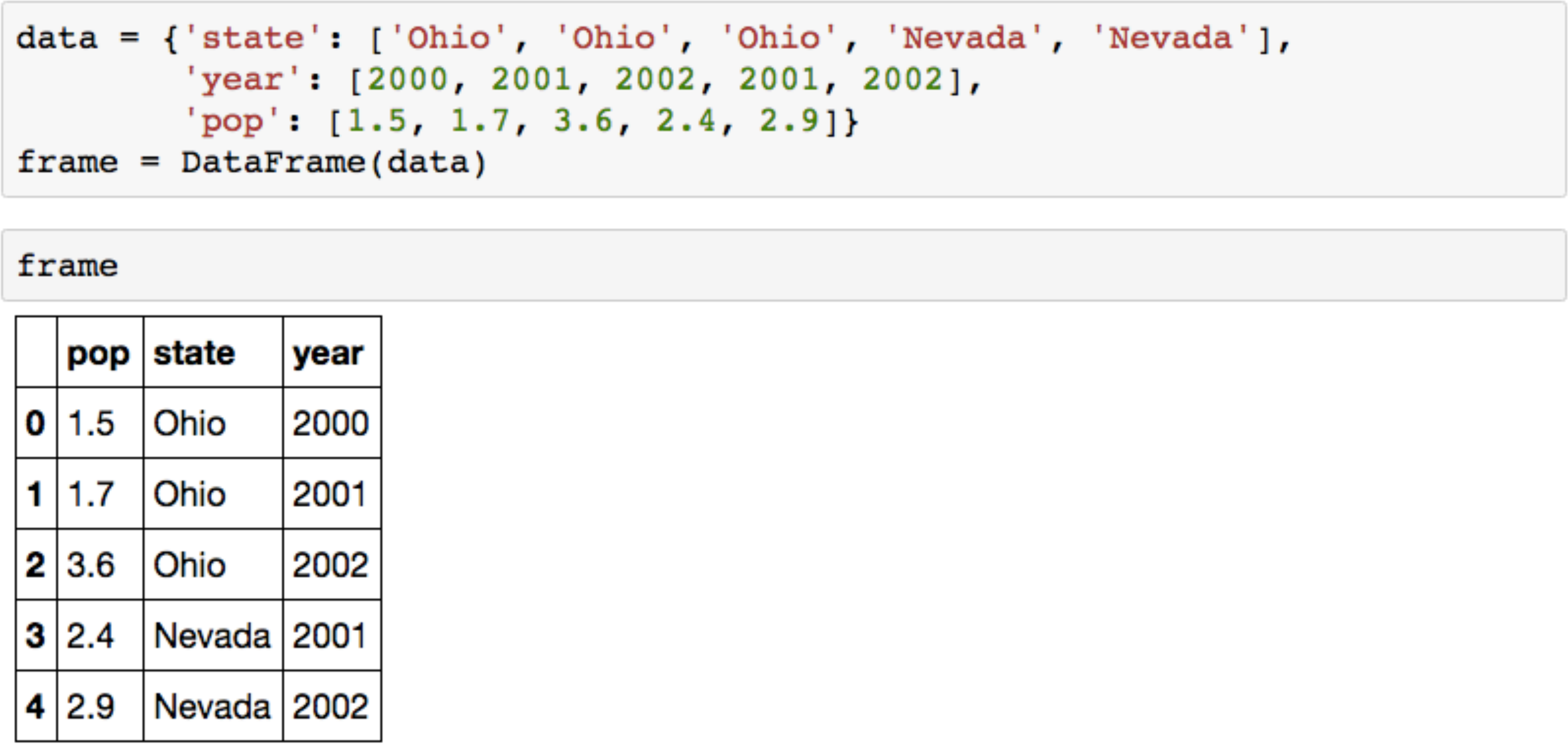


```
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
        'year': [2000, 2001, 2002, 2001, 2002],
        'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
```

```
DataFrame(data, columns=['year', 'state', 'pop'])
```

	year	state	pop
0	2000	Ohio	1.5
1	2001	Ohio	1.7
2	2002	Ohio	3.6
3	2001	Nevada	2.4
4	2002	Nevada	2.9

DataFrame: table in pandas

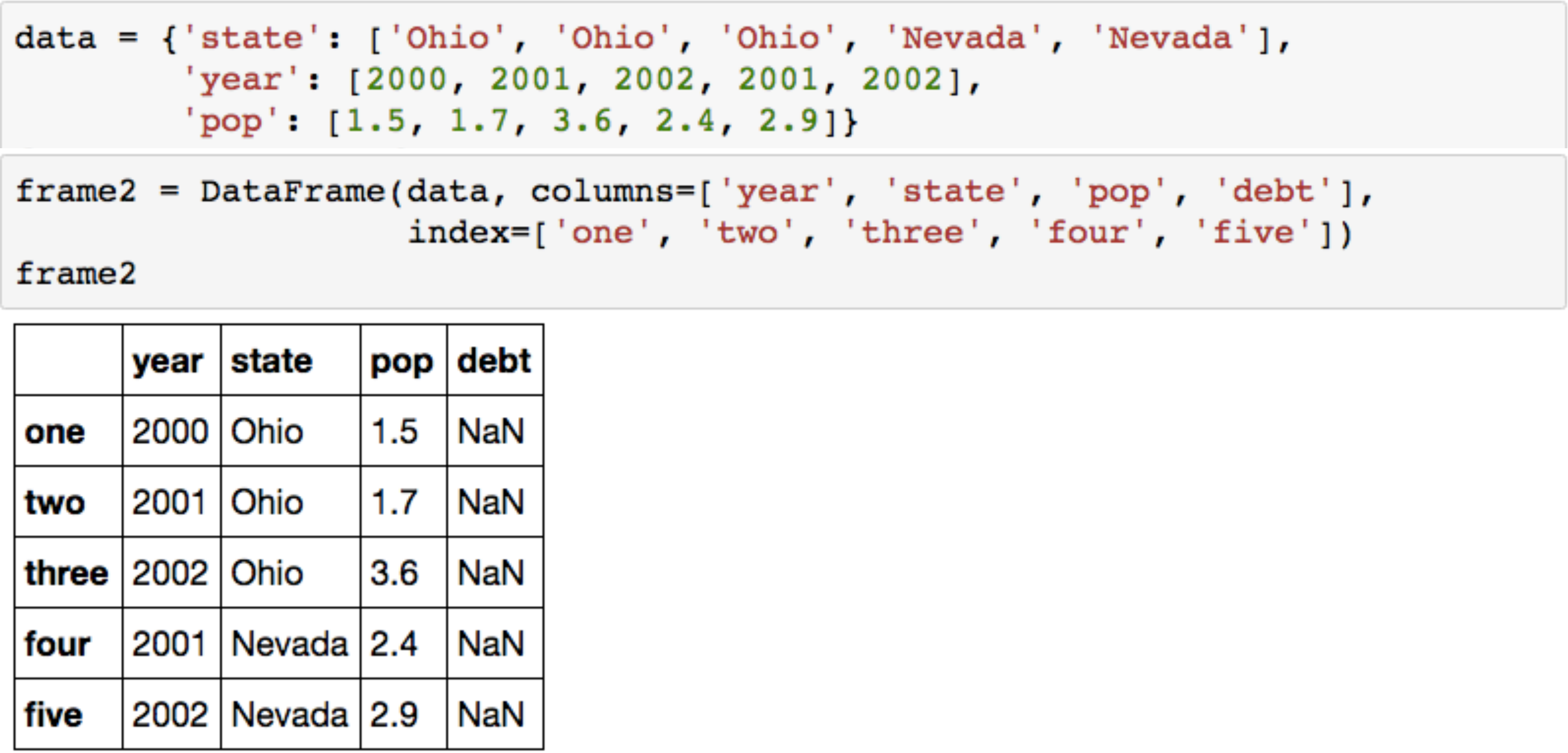


```
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
        'year': [2000, 2001, 2002, 2001, 2002],
        'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
frame = DataFrame(data)
```

```
frame
```

	pop	state	year
0	1.5	Ohio	2000
1	1.7	Ohio	2001
2	3.6	Ohio	2002
3	2.4	Nevada	2001
4	2.9	Nevada	2002

DataFrame: columns of lists with indices

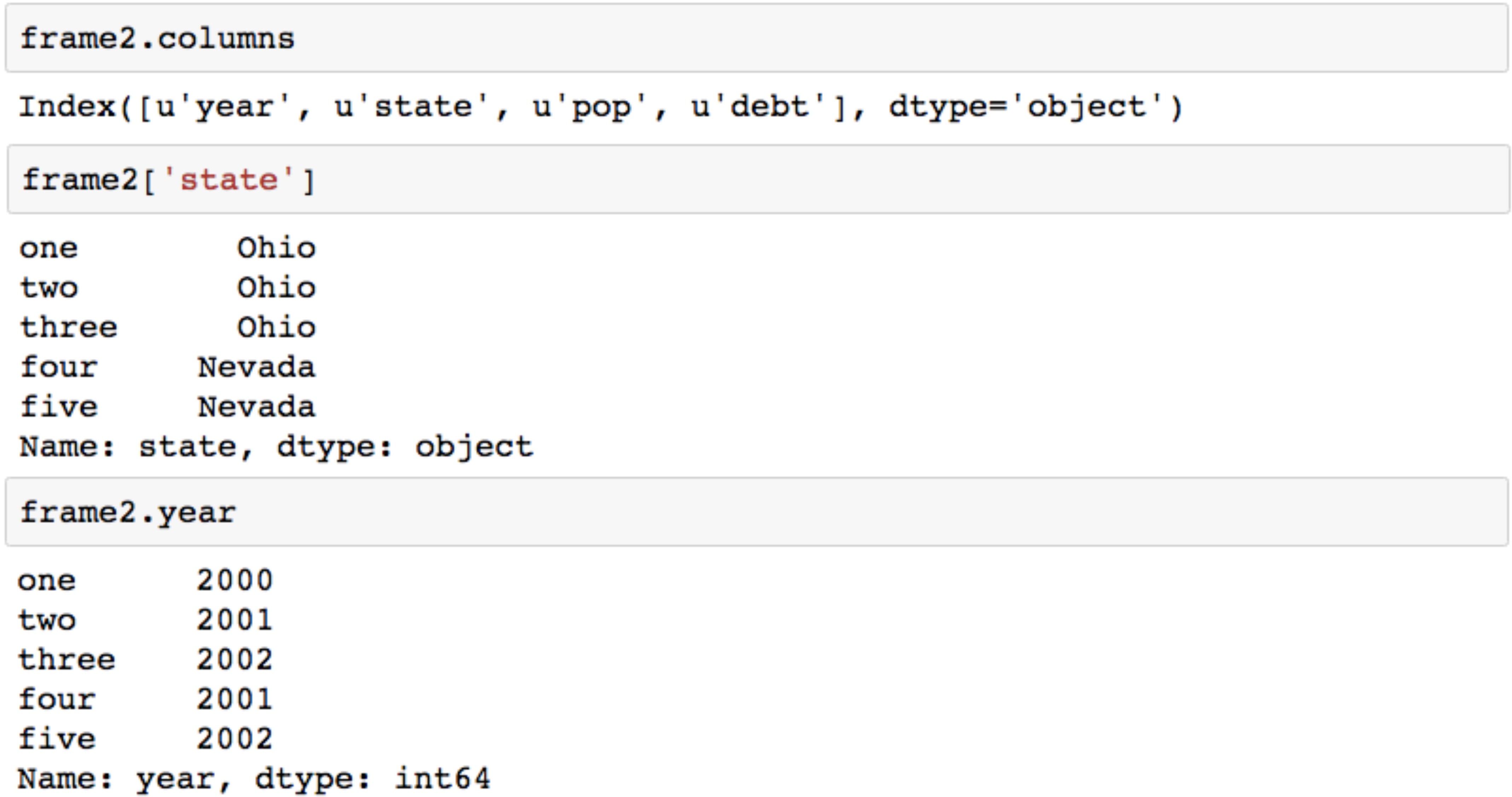


```
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
        'year': [2000, 2001, 2002, 2001, 2002],
        'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
```

```
frame2 = DataFrame(data, columns=['year', 'state', 'pop', 'debt'],
                   index=['one', 'two', 'three', 'four', 'five'])
frame2
```

	year	state	pop	debt
one	2000	Ohio	1.5	NaN
two	2001	Ohio	1.7	NaN
three	2002	Ohio	3.6	NaN
four	2001	Nevada	2.4	NaN
five	2002	Nevada	2.9	NaN

DataFrame: columns



```
frame2.columns
```

```
Index([u'year', u'state', u'pop', u'debt'], dtype='object')
```

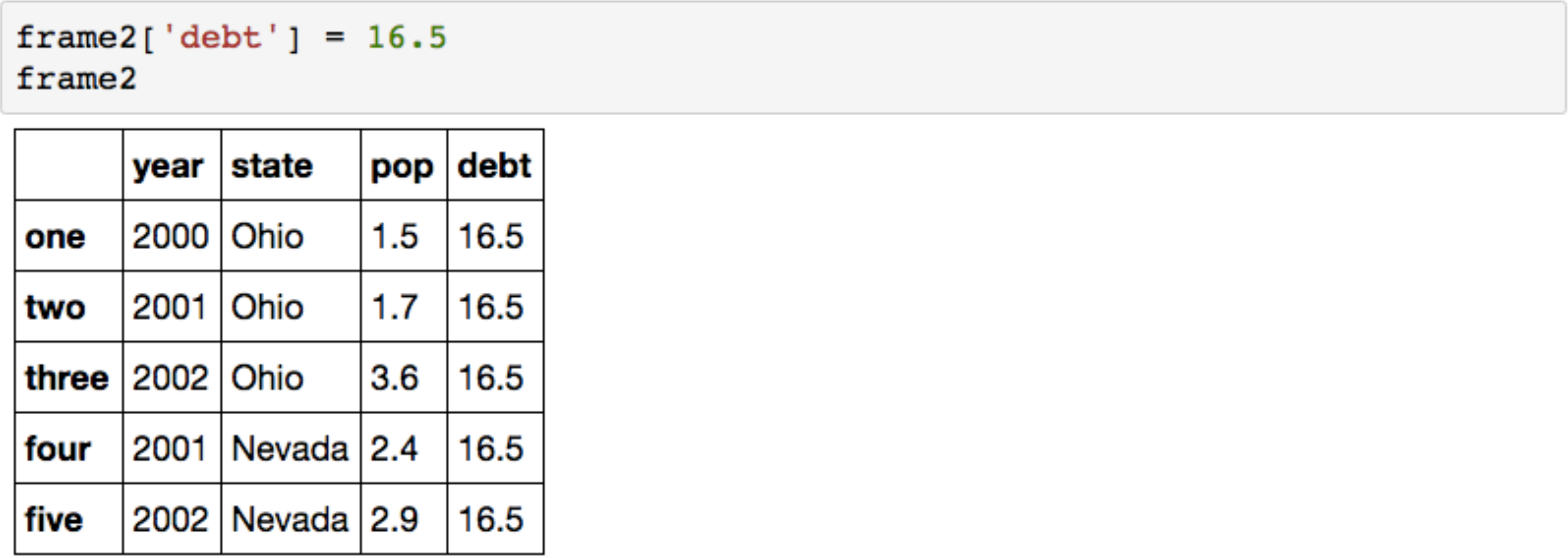
```
frame2['state']
```

```
one        Ohio
two        Ohio
three      Ohio
four     Nevada
five     Nevada
Name: state, dtype: object
```

```
frame2.year
```

```
one      2000
two      2001
three    2002
four     2001
five     2002
Name: year, dtype: int64
```

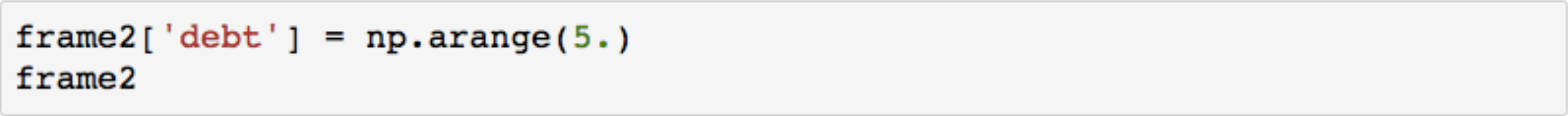
DataFrame: inserting data



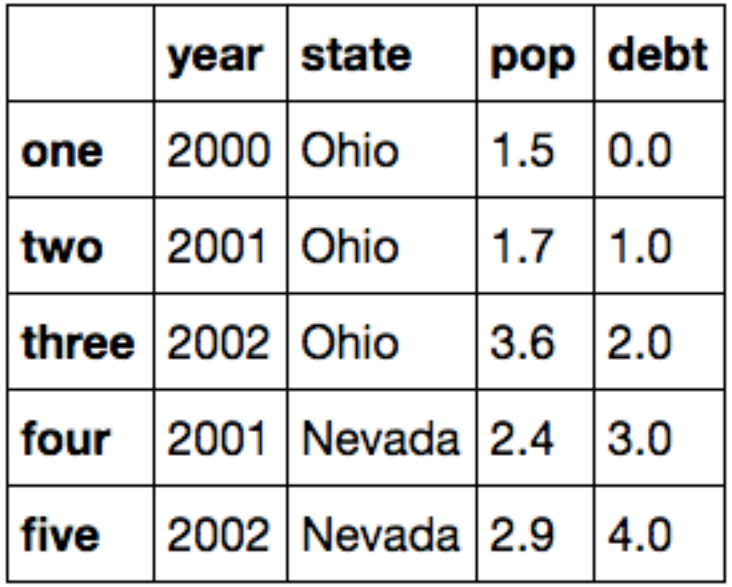
```
frame2['debt'] = 16.5
frame2
```

	year	state	pop	debt
one	2000	Ohio	1.5	16.5
two	2001	Ohio	1.7	16.5
three	2002	Ohio	3.6	16.5
four	2001	Nevada	2.4	16.5
five	2002	Nevada	2.9	16.5
]

DataFrame: inserting data

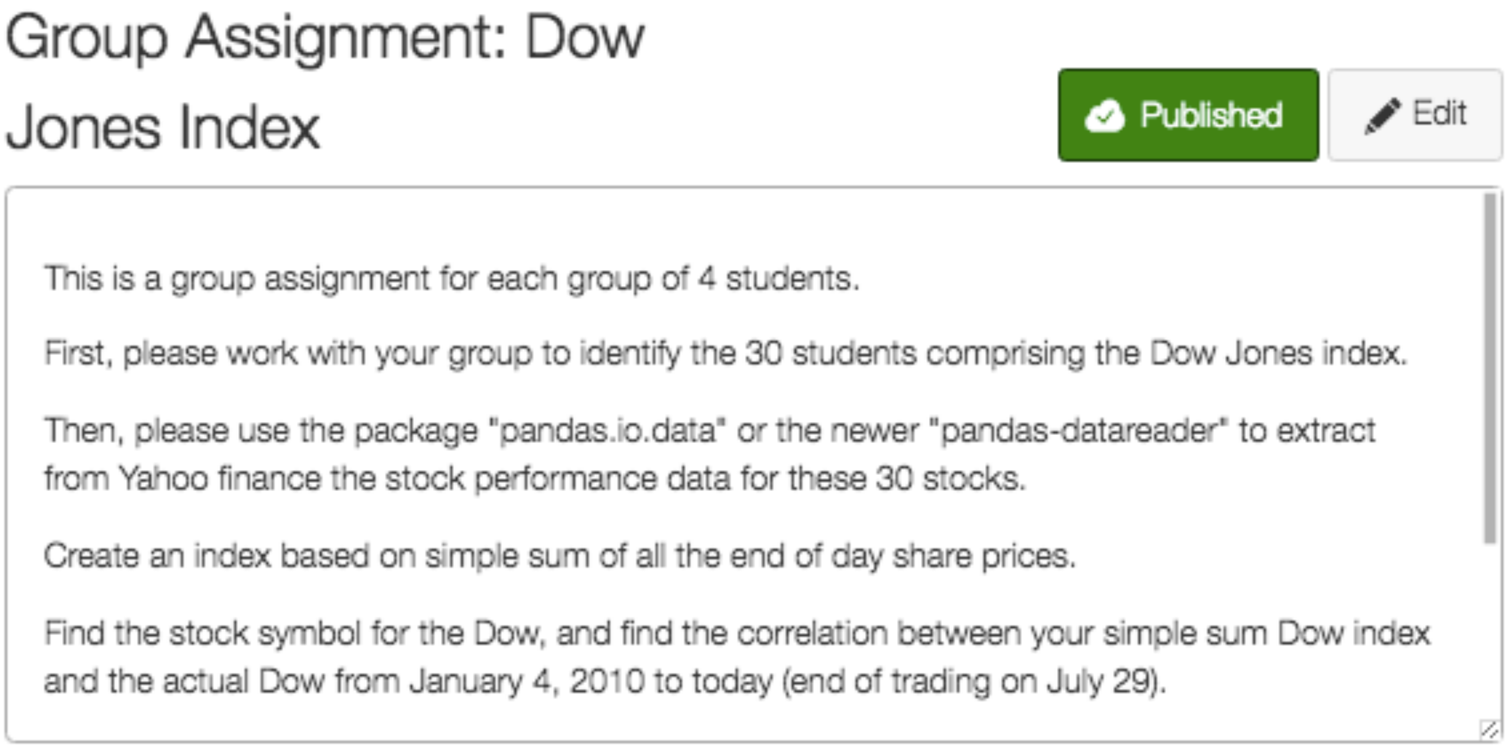


```
frame2['debt'] = np.arange(5.)
frame2
```



	year	state	pop	debt
one	2000	Ohio	1.5	0.0
two	2001	Ohio	1.7	1.0
three	2002	Ohio	3.6	2.0
four	2001	Nevada	2.4	3.0
five	2002	Nevada	2.9	4.0

Group Assignment: Dow Jones



Group Assignment: Dow Jones Index

This is a group assignment for each group of 4 students.

First, please work with your group to identify the 30 students comprising the Dow Jones index.

Then, please use the package "pandas.io.data" or the newer "pandas-datareader" to extract from Yahoo finance the stock performance data for these 30 stocks.

Create an index based on simple sum of all the end of day share prices.

Find the stock symbol for the Dow, and find the correlation between your simple sum Dow index and the actual Dow from January 4, 2010 to today (end of trading on July 29).