```
Last login: Sun Mar 25 22:25:15 on ttys001
Hansen@Hansens-MacBook-Pro:~ $ " ls
Applications         Library              Public               curl
Creative Cloud Files MDIGB0_ideal.xbgf    PycharmProjects      experiments
Desktop              Mathematica          Untitled.ipynb       nohup.out
Developer            Movies               amber_test           requirement.txt
Documents            Music                anaconda             seaborn-data
Downloads            Parallels            android              solarized
Dropbox              Pictures             bin                  texput.log
Hansen@Hansens-MacBook-Pro:~ $ " top
Hansen@Hansens-MacBook-Pro:~ $ " ipython
Python 2.7.12 |Anaconda custom (x86_64)| (default, Jul  2 2016, 17:43:17)
Type "copyright", "credits" or "license" for more information.

IPython 4.2.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: import pandas as pd

In [2]: import numpy as np

In [3]: import matpoltlib.pyplot as plt
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
<ipython-input-3-522f058eb800> in <module>()
----> 1 import matpoltlib.pyplot as plt

ImportError: No module named matpoltlib.pyplot

In [4]: import matpoltlib.pylot as plt
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
<ipython-input-4-e2b5c9d99bfc> in <module>()
----> 1 import matpoltlib.pylot as plt

ImportError: No module named matpoltlib.pylot

In [5]: import matplotlib.pyplot as plt

In [6]: %matplotlib
Using matplotlib backend: MacOSX

In [7]: ipython
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-7-aa842c094c41> in <module>()
----> 1 ipython

NameError: name 'ipython' is not defined

In [8]: obj3 = pd.Series(['blue', 'purple', 'yellow'], index = [0, 2, 4])

In [9]: obj3.reindex(range(6), method = 'ffill')
Out[9]:
0      blue
1      blue
2    purple
3    purple
4    yellow
5    yellow
dtype: object

In [10]: obj3 = pd.Series(['blue', 'purple', 'yellow'], index = [0, 2, 4])

In [11]: obj3.reindex(range(6))
```

```
Out[11]:
0      blue
1       NaN
2    purple
3       NaN
4    yellow
5       NaN
dtype: object

In [12]: obj3.reindex(method = 'bfill')
Out[12]:
0      blue
2    purple
4    yellow
dtype: object

In [13]: obj3
Out[13]:
0      blue
2    purple
4    yellow
dtype: object

In [14]: obj3.reindex(range(6), method = 'bfill')
Out[14]:
0      blue
1    purple
2    purple
3    yellow
4    yellow
5       NaN
dtype: object

In [15]: obj3
Out[15]:
0      blue
2    purple
4    yellow
dtype: object

In [16]: data = pd.DataFrame(np.arange(16).reshape((4, 4)), index = ['Ohio', 'Colorado', 'Utah', 'New
 York'], columns = ['one', 'two', 'three', 'four'])

In [17]: data
Out[17]:
          one  two  three  four
Ohio        0    1      2     3
Colorado    4    5      6     7
Utah        8    9     10    11
New York   12   13     14    15

In [18]: data['two']
Out[18]:
Ohio         1
Colorado     5
Utah         9
New York    13
Name: two, dtype: int64

In [19]: data[['three':]]
  File "<ipython-input-19-9b4aaba1ba09>", line 1
    data[['three':]]
                 ^
SyntaxError: invalid syntax


In [20]: data['three':]
---------------------------------------------------------------------------
```

```
KeyError                                  Traceback (most recent call last)
<ipython-input-20-5da59067fa16> in <module>()
----> 1 data['three':]

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/frame.pyc in __getitem__(self, key)
   1983 
   1984         # see if we can slice the rows
-> 1985         indexer = convert_to_index_sliceable(self, key)
   1986         if indexer is not None:
   1987             return self._getitem_slice(indexer)

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/indexing.pyc in convert_to_index_slice
able(obj, key)
   1756     idx = obj.index
   1757     if isinstance(key, slice):
-> 1758         return idx._convert_slice_indexer(key, kind='getitem')
   1759 
   1760     elif isinstance(key, compat.string_types):

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/indexes/base.pyc in _convert_slice_indexer(
self, key, kind)
   1077         else:
   1078             try:
-> 1079                 indexer = self.slice_indexer(start, stop, step, kind=kind)
   1080             except Exception:
   1081                 if is_index_slice:

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/indexes/base.pyc in slice_indexer(self, sta
rt, end, step, kind)
   2783         """
   2784         start_slice, end_slice = self.slice_locs(start, end, step=step,
-> 2785                                                  kind=kind)
   2786 
   2787         # return a slice

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/indexes/base.pyc in slice_locs(self, start,
 end, step, kind)
   2962         start_slice = None
   2963         if start is not None:
-> 2964             start_slice = self.get_slice_bound(start, 'left', kind)
   2965         if start_slice is None:
   2966             start_slice = 0

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/indexes/base.pyc in get_slice_bound(self, l
abel, side, kind)
   2911             except ValueError:
   2912                 # raise the original KeyError
-> 2913                 raise err
   2914 
   2915         if isinstance(slc, np.ndarray):

KeyError: 'three'

In [21]: data[:2]
Out[21]:
          one  two  three  four
Ohio        0    1      2     3
Colorado    4    5      6     7

In [22]: data < 5
Out[22]:
            one    two  three   four
Ohio       True   True   True   True
Colorado   True  False  False  False
Utah      False  False  False  False
New York  False  False  False  False

In [23]: data
```

```
Out[23]:
          one  two  three  four
Ohio        0    1      2     3
Colorado    4    5      6     7
Utah        8    9     10    11
New York   12   13     14    15

In [24]: data.ix[data.three > 5]
Out[24]:
          one  two  three  four
Colorado    4    5      6     7
Utah        8    9     10    11
New York   12   13     14    15

In [25]: data.three
Out[25]:
Ohio         2
Colorado     6
Utah        10
New York    14
Name: three, dtype: int64

In [26]: df = pd.Da
pd.DataFrame      pd.DateOffset     pd.DatetimeIndex

In [26]: df = pd.DataFrame(np.ran)
np.random  np.rank

In [26]: df = pd.DataFrame(np.random.r)
np.random.rand             np.random.random           np.random.ranf
np.random.randint          np.random.random_integers  np.random.rayleigh
np.random.randn            np.random.random_sample

In [26]: df = pd.DataFrame(np.random.randn(4,3), index = ['a', 'a', 'b', 'b'])

In [27]: df
Out[27]:
          0         1         2
a  1.659823 -0.765419  0.357108
a  1.894311  0.102078 -0.570970
b  0.345480  2.302504 -0.876686
b -1.196642 -0.511981 -0.268261

In [28]: df
Out[28]:
          0         1         2
a  1.659823 -0.765419  0.357108
a  1.894311  0.102078 -0.570970
b  0.345480  2.302504 -0.876686
b -1.196642 -0.511981 -0.268261

In [29]: df.ix['b']
Out[29]:
          0         1         2
b  0.345480  2.302504 -0.876686
b -1.196642 -0.511981 -0.268261

In [30]: df.describe
Out[30]:
<bound method DataFrame.describe of           0         1         2
a  1.659823 -0.765419  0.357108
a  1.894311  0.102078 -0.570970
b  0.345480  2.302504 -0.876686
b -1.196642 -0.511981 -0.268261>

In [31]: df.describe()
Out[31]:
              0         1         2
```

```
count  4.000000  4.000000  4.000000
mean   0.675743  0.281796 -0.339702
std    1.422231  1.395505  0.526778
min   -1.196642 -0.765419 -0.876686
25%   -0.040051 -0.575340 -0.647399
50%    1.002651 -0.204951 -0.419615
75%    1.718445  0.652185 -0.111919
max    1.894311  2.302504  0.357108

In [32]: df.describe(axis = 1)
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-32-4bde61cc5202> in <module>()
----> 1 df.describe(axis = 1)

TypeError: describe() got an unexpected keyword argument 'axis'

In [33]: df.describe?
Signature: df.describe(percentiles=None, include=None, exclude=None)
Docstring:
Generate various summary statistics, excluding NaN values.

Parameters
----------
percentiles : array-like, optional
    The percentiles to include in the output. Should all
    be in the interval [0, 1]. By default `percentiles` is
    [.25, .5, .75], returning the 25th, 50th, and 75th percentiles.
include, exclude : list-like, 'all', or None (default)
    Specify the form of the returned result. Either:

    - None to both (default). The result will include only
      numeric-typed columns or, if none are, only categorical columns.
    - A list of dtypes or strings to be included/excluded.
      To select all numeric types use numpy numpy.number. To select
      categorical objects use type object. See also the select_dtypes
      documentation. eg. df.describe(include=['O'])
    - If include is the string 'all', the output column-set will
      match the input one.

Returns
-------
summary: NDFrame of summary statistics

Notes
-----
The output DataFrame index depends on the requested dtypes:

For numeric dtypes, it will include: count, mean, std, min,
max, and lower, 50, and upper percentiles.

For object dtypes (e.g. timestamps or strings), the index
will include the count, unique, most common, and frequency of the
most common. Timestamps also include the first and last items.

For mixed dtypes, the index will be the union of the corresponding
output types. Non-applicable entries will be filled with NaN.
Note that mixed-dtype outputs can only be returned from mixed-dtype
inputs and appropriate use of the include/exclude arguments.

If multiple values have the highest count, then the
`count` and `most common` pair will be arbitrarily chosen from
among those with the highest count.

The include, exclude arguments are ignored for Series.

See Also
--------
```

```
DataFrame.select_dtypes
File:      ~/anaconda/lib/python2.7/site-packages/pandas/core/generic.py
Type:      instancemethod

In [34]: obj = pd.Series(['a', 'a', 'b', 'c'] * 4)

In [35]: obj
Out[35]:
0     a
1     a
2     b
3     c
4     a
5     a
6     b
7     c
8     a
9     a
10    b
11    c
12    a
13    a
14    b
15    c
dtype: object

In [36]: obj.describe()
Out[36]:
count     16
unique     3
top        a
freq       8
dtype: object

In [37]: import pandas.io.data as web
/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py:35: FutureWarning:
The pandas.io.data module is moved to a separate package (pandas-datareader) and will be removed from
 pandas in a future version.
After installing the pandas-datareader package (https://github.com/pydata/pandas-datareader), you can
 change the import ``from pandas.io import data, wb`` to ``from pandas_datareader import data, wb``.
  FutureWarning)

In [38]: from pandas.io import data

In [39]: all_data = [{}
   ....:
   ....:
   ....:
   ....:
   ....:
   ....:
   ....:
   ....:
   ....: ]

In [40]: all_data = {}

In [41]: for ticker in ['AAPL', 'IBM', 'MSFT', 'GOOG']:
   ....:        all_data[ticker] = data.get_
data.get_components_yahoo  data.get_data_google       data.get_quote_yahoo
data.get_data_famafrench   data.get_data_yahoo
data.get_data_fred         data.get_quote_google
   ....:        all_data[ticker] = data.get_d
data.get_data_famafrench  data.get_data_google
data.get_data_fred        data.get_data_yahoo
   ....:        all_data[ticker] = data.get_data_yahoo(ticker, '1/1/2000', '1/1/2000')
   ....:
---------------------------------------------------------------------------
```

```
IOError                                   Traceback (most recent call last)
<ipython-input-41-478352023485> in <module>()
      1 for ticker in ['AAPL', 'IBM', 'MSFT', 'GOOG']:
----> 2         all_data[ticker] = data.get_data_yahoo(ticker, '1/1/2000', '1/1/2000')
      3 

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in get_data_yahoo(symbols, start
, end, retry_count, pause, adjust_price, ret_index, chunksize, interval)
    438         raise ValueError("Invalid interval: valid values are 'd', 'w', 'm' and 'v'")
    439     return _get_data_from(symbols, start, end, interval, retry_count, pause,
--> 440                           adjust_price, ret_index, chunksize, 'yahoo')
    441 
    442 

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_data_from(symbols, start
, end, interval, retry_count, pause, adjust_price, ret_index, chunksize, source)
    379     # If a single symbol, (e.g., 'GOOG')
    380     if isinstance(symbols, (compat.string_types, int)):
--> 381         hist_data = src_fn(symbols, start, end, interval, retry_count, pause)
    382     # Or multiple symbols, (e.g., ['GOOG', 'AAPL', 'MSFT'])
    383     elif isinstance(symbols, DataFrame):

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_hist_yahoo(sym, start, e
nd, interval, retry_count, pause)
    222            '&g=%s' % interval +
    223            '&ignore=.csv')
--> 224     return _retry_read_url(url, retry_count, pause, 'Yahoo!')
    225 
    226 

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _retry_read_url(url, retry_co
unt, pause, name)
    199 
    200     raise IOError("after %d tries, %s did not "
--> 201                   "return a 200 for url %r" % (retry_count, name, url))
    202 
    203 

IOError: after 3 tries, Yahoo! did not return a 200 for url 'http://ichart.finance.yahoo.com/table.cs
v?s=AAPL&a=0&b=1&c=2000&d=0&e=1&f=2000&g=d&ignore=.csv'

In [42]: for ticker in ['AAPL', 'IBM', 'MSFT', 'GOOG']:
        all_data[ticker] = data.get_data_yahoo(ticker, '1/1/2000', '1/1/2010')
   ....:
---------------------------------------------------------------------------
IOError                                   Traceback (most recent call last)
<ipython-input-42-3d541f0ff5a5> in <module>()
      1 for ticker in ['AAPL', 'IBM', 'MSFT', 'GOOG']:
----> 2         all_data[ticker] = data.get_data_yahoo(ticker, '1/1/2000', '1/1/2010')
      3 

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in get_data_yahoo(symbols, start
, end, retry_count, pause, adjust_price, ret_index, chunksize, interval)
    438         raise ValueError("Invalid interval: valid values are 'd', 'w', 'm' and 'v'")
    439     return _get_data_from(symbols, start, end, interval, retry_count, pause,
--> 440                           adjust_price, ret_index, chunksize, 'yahoo')
    441 
    442 

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_data_from(symbols, start
, end, interval, retry_count, pause, adjust_price, ret_index, chunksize, source)
    379     # If a single symbol, (e.g., 'GOOG')
    380     if isinstance(symbols, (compat.string_types, int)):
--> 381         hist_data = src_fn(symbols, start, end, interval, retry_count, pause)
    382     # Or multiple symbols, (e.g., ['GOOG', 'AAPL', 'MSFT'])
    383     elif isinstance(symbols, DataFrame):

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_hist_yahoo(sym, start, e
nd, interval, retry_count, pause)
    222             '&g=%s' % interval +
    223             '&ignore=.csv')
--> 224     return _retry_read_url(url, retry_count, pause, 'Yahoo!')
    225
    226

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _retry_read_url(url, retry_co
unt, pause, name)
    199
    200     raise IOError("after %d tries, %s did not "
--> 201                   "return a 200 for url %r" % (retry_count, name, url))
    202
    203

IOError: after 3 tries, Yahoo! did not return a 200 for url 'http://ichart.finance.yahoo.com/table.cs
v?s=AAPL&a=0&b=1&c=2000&d=0&e=1&f=2010&g=d&ignore=.csv'

In [43]: for ticker in ['AAPL', 'IBM', 'MSFT', 'GOOG']:
        all_data[ticker] = data.get_data_(ticker, '1/1/2000', '1/1/2010')
data.get_data_famafrench  data.get_data_google
data.get_data_fred        data.get_data_yahoo

In [43]: for ticker in ['AAPL', 'IBM', 'MSFT', 'GOOG']:
        all_data[ticker] = data.get_data_google(ticker, '1/1/2000', '1/1/2010')
   ....:
^C---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-43-77df4bcb8e8b> in <module>()
      1 for ticker in ['AAPL', 'IBM', 'MSFT', 'GOOG']:
----> 2         all_data[ticker] = data.get_data_google(ticker, '1/1/2000', '1/1/2010')
      3

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in get_data_google(symbols, star
t, end, retry_count, pause, adjust_price, ret_index, chunksize)
    474     """
    475     return _get_data_from(symbols, start, end, None, retry_count, pause,
--> 476                           adjust_price, ret_index, chunksize, 'google')
    477
    478

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_data_from(symbols, start
, end, interval, retry_count, pause, adjust_price, ret_index, chunksize, source)
    379     # If a single symbol, (e.g., 'GOOG')
    380     if isinstance(symbols, (compat.string_types, int)):
--> 381         hist_data = src_fn(symbols, start, end, interval, retry_count, pause)
    382     # Or multiple symbols, (e.g., ['GOOG', 'AAPL', 'MSFT'])
    383     elif isinstance(symbols, DataFrame):

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_hist_google(sym, start,
end, interval, retry_count, pause)
    243                                "enddate": end.strftime('%b %d, %Y'),
    244                                "output": "csv"}))
--> 245     return _retry_read_url(url, retry_count, pause, 'Google')
    246
    247

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _retry_read_url(url, retry_co
unt, pause, name)
    177         # kludge to close the socket ASAP
    178         try:
--> 179             with urlopen(url) as resp:
    180                 lines = resp.read()
    181         except _network_error_classes:

/Users/Hansen/anaconda/lib/python2.7/contextlib.pyc in __enter__(self)
     15     def __enter__(self):
     16         try:
```

```
---> 17             return self.gen.next()
     18         except StopIteration:
     19             raise RuntimeError("generator didn't yield")

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/common.pyc in urlopen(*args, **kwargs)
     49     @contextmanager
     50     def urlopen(*args, **kwargs):
---> 51         with closing(_urlopen(*args, **kwargs)) as f:
     52             yield f
     53 

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in urlopen(url, data, timeout, cafile, capath, cadef
ault, context)
    152     else:
    153         opener = _opener
--> 154     return opener.open(url, data, timeout)
    155 
    156 def install_opener(opener):

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in open(self, fullurl, data, timeout)
    427             req = meth(req)
    428 
--> 429         response = self._open(req, data)
    430 
    431         # post-process response

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in _open(self, req, data)
    445         protocol = req.get_type()
    446         result = self._call_chain(self.handle_open, protocol, protocol +
--> 447                                   '_open', req)
    448         if result:
    449             return result

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in _call_chain(self, chain, kind, meth_name, *args)
    405             func = getattr(handler, meth_name)
    406 
--> 407             result = func(*args)
    408             if result is not None:
    409                 return result

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in http_open(self, req)
   1226 
   1227     def http_open(self, req):
-> 1228         return self.do_open(httplib.HTTPConnection, req)
   1229 
   1230     http_request = AbstractHTTPHandler.do_request_

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in do_open(self, http_class, req, **http_conn_args)
   1193 
   1194         try:
-> 1195             h.request(req.get_method(), req.get_selector(), req.data, headers)
   1196         except socket.error, err: # XXX what error?
   1197             h.close()

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in request(self, method, url, body, headers)
   1055     def request(self, method, url, body=None, headers={}):
   1056         """Send a complete request to the server."""
-> 1057         self._send_request(method, url, body, headers)
   1058 
   1059     def _set_content_length(self, body, method):

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in _send_request(self, method, url, body, headers)
   1095         for hdr, value in headers.iteritems():
   1096             self.putheader(hdr, value)
-> 1097         self.endheaders(body)
   1098 
   1099     def getresponse(self, buffering=False):
```

```
/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in endheaders(self, message_body)
   1051         else:
   1052             raise CannotSendHeader()
-> 1053         self._send_output(message_body)
   1054 
   1055     def request(self, method, url, body=None, headers={}):

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in _send_output(self, message_body)
    895             msg += message_body
    896             message_body = None
--> 897         self.send(msg)
    898         if message_body is not None:
    899             #message_body was not a string (i.e. it is a file) and

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in send(self, data)
    857         if self.sock is None:
    858             if self.auto_open:
--> 859                 self.connect()
    860             else:
    861                 raise NotConnected()

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in connect(self)
    834         """Connect to the host and port specified in __init__."""
    835         self.sock = self._create_connection((self.host,self.port),
--> 836                                             self.timeout, self.source_address)
    837 
    838         if self._tunnel_host:

/Users/Hansen/anaconda/lib/python2.7/socket.pyc in create_connection(address, timeout, source_address
)
    564             if source_address:
    565                 sock.bind(source_address)
--> 566             sock.connect(sa)
    567             return sock
    568 

/Users/Hansen/anaconda/lib/python2.7/socket.pyc in meth(name, self, *args)
    226 
    227 def meth(name,self,*args):
--> 228     return getattr(self._sock,name)(*args)
    229 
    230 for _m in _socketmethods:

KeyboardInterrupt: 

In [44]: for ticker in ['AAPL', 'IBM', 'MSFT', 'GOOG']:
        all_data[ticker] = data.get_data_google(ticker, '1/1/2000', '1/1/2010')
   ....: 
^C---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-44-77df4bcb8e8b> in <module>()
      1 for ticker in ['AAPL', 'IBM', 'MSFT', 'GOOG']:
----> 2         all_data[ticker] = data.get_data_google(ticker, '1/1/2000', '1/1/2010')
      3 

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in get_data_google(symbols, star
t, end, retry_count, pause, adjust_price, ret_index, chunksize)
    474     """
    475     return _get_data_from(symbols, start, end, None, retry_count, pause,
--> 476                           adjust_price, ret_index, chunksize, 'google')
    477 
    478 

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_data_from(symbols, start
, end, interval, retry_count, pause, adjust_price, ret_index, chunksize, source)
    379     # If a single symbol, (e.g., 'GOOG')
    380     if isinstance(symbols, (compat.string_types, int)):
--> 381         hist_data = src_fn(symbols, start, end, interval, retry_count, pause)
```

```
    382     # Or multiple symbols, (e.g., ['GOOG', 'AAPL', 'MSFT'])
    383     elif isinstance(symbols, DataFrame):

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_hist_google(sym, start,
end, interval, retry_count, pause)
    243                                        "enddate": end.strftime('%b %d, %Y'),
    244                                        "output": "csv"}))
--> 245     return _retry_read_url(url, retry_count, pause, 'Google')
    246 
    247 

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _retry_read_url(url, retry_co
unt, pause, name)
    177         # kludge to close the socket ASAP
    178         try:
--> 179             with urlopen(url) as resp:
    180                 lines = resp.read()
    181         except _network_error_classes:

/Users/Hansen/anaconda/lib/python2.7/contextlib.pyc in __enter__(self)
     15     def __enter__(self):
     16         try:
---> 17             return self.gen.next()
     18         except StopIteration:
     19             raise RuntimeError("generator didn't yield")

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/common.pyc in urlopen(*args, **kwargs)
     49     @contextmanager
     50     def urlopen(*args, **kwargs):
---> 51         with closing(_urlopen(*args, **kwargs)) as f:
     52             yield f
     53 

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in urlopen(url, data, timeout, cafile, capath, cadef
ault, context)
    152     else:
    153         opener = _opener
--> 154     return opener.open(url, data, timeout)
    155 
    156 def install_opener(opener):

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in open(self, fullurl, data, timeout)
    427             req = meth(req)
    428 
--> 429         response = self._open(req, data)
    430 
    431         # post-process response

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in _open(self, req, data)
    445         protocol = req.get_type()
    446         result = self._call_chain(self.handle_open, protocol, protocol +
--> 447                                   '_open', req)
    448         if result:
    449             return result

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in _call_chain(self, chain, kind, meth_name, *args)
    405             func = getattr(handler, meth_name)
    406 
--> 407             result = func(*args)
    408             if result is not None:
    409                 return result

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in http_open(self, req)
   1226 
   1227     def http_open(self, req):
-> 1228         return self.do_open(httplib.HTTPConnection, req)
   1229 
   1230     http_request = AbstractHTTPHandler.do_request_
```

```
/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in do_open(self, http_class, req, **http_conn_args)
   1193 
   1194         try:
-> 1195             h.request(req.get_method(), req.get_selector(), req.data, headers)
   1196         except socket.error, err: # XXX what error?
   1197             h.close()

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in request(self, method, url, body, headers)
   1055     def request(self, method, url, body=None, headers={}):
   1056         """Send a complete request to the server."""
-> 1057         self._send_request(method, url, body, headers)
   1058 
   1059     def _set_content_length(self, body, method):

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in _send_request(self, method, url, body, headers)
   1095         for hdr, value in headers.iteritems():
   1096             self.putheader(hdr, value)
-> 1097         self.endheaders(body)
   1098 
   1099     def getresponse(self, buffering=False):

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in endheaders(self, message_body)
   1051         else:
   1052             raise CannotSendHeader()
-> 1053         self._send_output(message_body)
   1054 
   1055     def request(self, method, url, body=None, headers={}):

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in _send_output(self, message_body)
    895             msg += message_body
    896             message_body = None
--> 897         self.send(msg)
    898         if message_body is not None:
    899             #message_body was not a string (i.e. it is a file) and

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in send(self, data)
    857         if self.sock is None:
    858             if self.auto_open:
--> 859                 self.connect()
    860             else:
    861                 raise NotConnected()

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in connect(self)
    834         """Connect to the host and port specified in __init__."""
    835         self.sock = self._create_connection((self.host,self.port),
--> 836                                             self.timeout, self.source_address)
    837 
    838         if self._tunnel_host:

/Users/Hansen/anaconda/lib/python2.7/socket.pyc in create_connection(address, timeout, source_address
)
    564             if source_address:
    565                 sock.bind(source_address)
--> 566             sock.connect(sa)
    567             return sock
    568 

/Users/Hansen/anaconda/lib/python2.7/socket.pyc in meth(name, self, *args)
    226 
    227 def meth(name,self,*args):
--> 228     return getattr(self._sock,name)(*args)
    229 
    230 for _m in _socketmethods:

KeyboardInterrupt: 

In [45]: for ticker in ['AAPL', 'IBM', 'MSFT']:
        all_data[ticker] = data.get_data_google(ticker, '1/1/2000', '1/1/2010')
   ....:
 :q
^C---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-45-8ef27acaa610> in <module>()
      1 for ticker in ['AAPL', 'IBM', 'MSFT']:
----> 2         all_data[ticker] = data.get_data_google(ticker, '1/1/2000', '1/1/2010')
      3

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in get_data_google(symbols, star
t, end, retry_count, pause, adjust_price, ret_index, chunksize)
    474     """
    475     return _get_data_from(symbols, start, end, None, retry_count, pause,
--> 476                           adjust_price, ret_index, chunksize, 'google')
    477
    478

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_data_from(symbols, start
, end, interval, retry_count, pause, adjust_price, ret_index, chunksize, source)
    379     # If a single symbol, (e.g., 'GOOG')
    380     if isinstance(symbols, (compat.string_types, int)):
--> 381         hist_data = src_fn(symbols, start, end, interval, retry_count, pause)
    382     # Or multiple symbols, (e.g., ['GOOG', 'AAPL', 'MSFT'])
    383     elif isinstance(symbols, DataFrame):

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_hist_google(sym, start,
end, interval, retry_count, pause)
    243                                "enddate": end.strftime('%b %d, %Y'),
    244                                "output": "csv"}))
--> 245     return _retry_read_url(url, retry_count, pause, 'Google')
    246
    247

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _retry_read_url(url, retry_co
unt, pause, name)
    177         # kludge to close the socket ASAP
    178         try:
--> 179             with urlopen(url) as resp:
    180                 lines = resp.read()
    181         except _network_error_classes:

/Users/Hansen/anaconda/lib/python2.7/contextlib.pyc in __enter__(self)
     15     def __enter__(self):
     16         try:
---> 17             return self.gen.next()
     18         except StopIteration:
     19             raise RuntimeError("generator didn't yield")

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/common.pyc in urlopen(*args, **kwargs)
     49     @contextmanager
     50     def urlopen(*args, **kwargs):
---> 51         with closing(_urlopen(*args, **kwargs)) as f:
     52             yield f
     53

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in urlopen(url, data, timeout, cafile, capath, cadef
ault, context)
    152     else:
    153         opener = _opener
--> 154     return opener.open(url, data, timeout)
    155
    156 def install_opener(opener):

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in open(self, fullurl, data, timeout)
    427             req = meth(req)
    428
--> 429         response = self._open(req, data)
```

```
    430
    431         # post-process response

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in _open(self, req, data)
    445         protocol = req.get_type()
    446         result = self._call_chain(self.handle_open, protocol, protocol +
--> 447                                   '_open', req)
    448         if result:
    449             return result

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in _call_chain(self, chain, kind, meth_name, *args)
    405             func = getattr(handler, meth_name)
    406
--> 407             result = func(*args)
    408             if result is not None:
    409                 return result

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in http_open(self, req)
   1226
   1227     def http_open(self, req):
-> 1228         return self.do_open(httplib.HTTPConnection, req)
   1229
   1230     http_request = AbstractHTTPHandler.do_request_

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in do_open(self, http_class, req, **http_conn_args)
   1193
   1194         try:
-> 1195             h.request(req.get_method(), req.get_selector(), req.data, headers)
   1196         except socket.error, err: # XXX what error?
   1197             h.close()

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in request(self, method, url, body, headers)
   1055     def request(self, method, url, body=None, headers={}):
   1056         """Send a complete request to the server."""
-> 1057         self._send_request(method, url, body, headers)
   1058
   1059     def _set_content_length(self, body, method):

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in _send_request(self, method, url, body, headers)
   1095         for hdr, value in headers.iteritems():
   1096             self.putheader(hdr, value)
-> 1097         self.endheaders(body)
   1098
   1099     def getresponse(self, buffering=False):

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in endheaders(self, message_body)
   1051         else:
   1052             raise CannotSendHeader()
-> 1053         self._send_output(message_body)
   1054
   1055     def request(self, method, url, body=None, headers={}):

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in _send_output(self, message_body)
    895             msg += message_body
    896             message_body = None
--> 897         self.send(msg)
    898         if message_body is not None:
    899             #message_body was not a string (i.e. it is a file) and

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in send(self, data)
    857         if self.sock is None:
    858             if self.auto_open:
--> 859                 self.connect()
    860             else:
    861                 raise NotConnected()

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in connect(self)
    834         """Connect to the host and port specified in __init__."""
    835         self.sock = self._create_connection((self.host,self.port),
--> 836                                            self.timeout, self.source_address)
    837 
    838         if self._tunnel_host:

/Users/Hansen/anaconda/lib/python2.7/socket.pyc in create_connection(address, timeout, source_address
)
    564             if source_address:
    565                 sock.bind(source_address)
--> 566             sock.connect(sa)
    567             return sock
    568 

/Users/Hansen/anaconda/lib/python2.7/socket.pyc in meth(name, self, *args)
    226 
    227 def meth(name,self,*args):
--> 228     return getattr(self._sock,name)(*args)
    229 
    230 for _m in _socketmethods:

KeyboardInterrupt: 

In [46]: from pandas.io import data as web

In [47]: for ticker in ['AAPL', 'IBM', 'MSFT']:
   ....:         all_data[ticker] = web.get_data_yahoo(ticker, '1/1/2000', '1/1/2010')
   ....: 
---------------------------------------------------------------------------
IOError                                   Traceback (most recent call last)
<ipython-input-47-7a0d1f560d3e> in <module>()
      1 for ticker in ['AAPL', 'IBM', 'MSFT']:
----> 2         all_data[ticker] = web.get_data_yahoo(ticker, '1/1/2000', '1/1/2010')
      3 

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in get_data_yahoo(symbols, start
, end, retry_count, pause, adjust_price, ret_index, chunksize, interval)
    438         raise ValueError("Invalid interval: valid values are 'd', 'w', 'm' and 'v'")
    439     return _get_data_from(symbols, start, end, interval, retry_count, pause,
--> 440                           adjust_price, ret_index, chunksize, 'yahoo')
    441 
    442 

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_data_from(symbols, start
, end, interval, retry_count, pause, adjust_price, ret_index, chunksize, source)
    379     # If a single symbol, (e.g., 'GOOG')
    380     if isinstance(symbols, (compat.string_types, int)):
--> 381         hist_data = src_fn(symbols, start, end, interval, retry_count, pause)
    382     # Or multiple symbols, (e.g., ['GOOG', 'AAPL', 'MSFT'])
    383     elif isinstance(symbols, DataFrame):

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_hist_yahoo(sym, start, e
nd, interval, retry_count, pause)
    222            '&g=%s' % interval +
    223            '&ignore=.csv')
--> 224     return _retry_read_url(url, retry_count, pause, 'Yahoo!')
    225 
    226 

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _retry_read_url(url, retry_co
unt, pause, name)
    199 
    200     raise IOError("after %d tries, %s did not "
--> 201                   "return a 200 for url %r" % (retry_count, name, url))
    202 
    203 

IOError: after 3 tries, Yahoo! did not return a 200 for url 'http://ichart.finance.yahoo.com/table.cs
v?s=AAPL&a=0&b=1&c=2000&d=0&e=1&f=2010&g=d&ignore=.csv'
```

```
In [48]: for ticker in ['AAPL', 'IBM', 'MSFT']:
        all_data[ticker] = web.get_data(ticker, '1/1/2000', '1/1/2010')
web.get_data_famafrench  web.get_data_fred        web.get_data_google      web.get_data_yahoo

In [48]: for ticker in ['AAPL', 'IBM', 'MSFT']:
        all_data[ticker] = web.get_data_google(ticker, '1/1/2000', '1/1/2010')
   ....:
^C---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-48-e8a98b9d1495> in <module>()
      1 for ticker in ['AAPL', 'IBM', 'MSFT']:
----> 2         all_data[ticker] = web.get_data_google(ticker, '1/1/2000', '1/1/2010')
      3

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in get_data_google(symbols, star
t, end, retry_count, pause, adjust_price, ret_index, chunksize)
    474     """
    475     return _get_data_from(symbols, start, end, None, retry_count, pause,
--> 476                           adjust_price, ret_index, chunksize, 'google')
    477
    478

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_data_from(symbols, start
, end, interval, retry_count, pause, adjust_price, ret_index, chunksize, source)
    379     # If a single symbol, (e.g., 'GOOG')
    380     if isinstance(symbols, (compat.string_types, int)):
--> 381         hist_data = src_fn(symbols, start, end, interval, retry_count, pause)
    382     # Or multiple symbols, (e.g., ['GOOG', 'AAPL', 'MSFT'])
    383     elif isinstance(symbols, DataFrame):

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_hist_google(sym, start,
end, interval, retry_count, pause)
    243                                  "enddate": end.strftime('%b %d, %Y'),
    244                                  "output": "csv"}))
--> 245     return _retry_read_url(url, retry_count, pause, 'Google')
    246
    247

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _retry_read_url(url, retry_co
unt, pause, name)
    177         # kludge to close the socket ASAP
    178         try:
--> 179             with urlopen(url) as resp:
    180                 lines = resp.read()
    181         except _network_error_classes:

/Users/Hansen/anaconda/lib/python2.7/contextlib.pyc in __enter__(self)
     15     def __enter__(self):
     16         try:
---> 17             return self.gen.next()
     18         except StopIteration:
     19             raise RuntimeError("generator didn't yield")

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/common.pyc in urlopen(*args, **kwargs)
     49     @contextmanager
     50     def urlopen(*args, **kwargs):
---> 51         with closing(_urlopen(*args, **kwargs)) as f:
     52             yield f
     53

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in urlopen(url, data, timeout, cafile, capath, cadef
ault, context)
    152     else:
    153         opener = _opener
--> 154     return opener.open(url, data, timeout)
    155
    156 def install_opener(opener):
```

```
/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in open(self, fullurl, data, timeout)
    427             req = meth(req)
    428 
--> 429         response = self._open(req, data)
    430 
    431         # post-process response

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in _open(self, req, data)
    445         protocol = req.get_type()
    446         result = self._call_chain(self.handle_open, protocol, protocol +
--> 447                                   '_open', req)
    448         if result:
    449             return result

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in _call_chain(self, chain, kind, meth_name, *args)
    405             func = getattr(handler, meth_name)
    406 
--> 407             result = func(*args)
    408             if result is not None:
    409                 return result

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in http_open(self, req)
   1226 
   1227     def http_open(self, req):
-> 1228         return self.do_open(httplib.HTTPConnection, req)
   1229 
   1230     http_request = AbstractHTTPHandler.do_request_

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in do_open(self, http_class, req, **http_conn_args)
   1193 
   1194         try:
-> 1195             h.request(req.get_method(), req.get_selector(), req.data, headers)
   1196         except socket.error, err: # XXX what error?
   1197             h.close()

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in request(self, method, url, body, headers)
   1055     def request(self, method, url, body=None, headers={}):
   1056         """Send a complete request to the server."""
-> 1057         self._send_request(method, url, body, headers)
   1058 
   1059     def _set_content_length(self, body, method):

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in _send_request(self, method, url, body, headers)
   1095         for hdr, value in headers.iteritems():
   1096             self.putheader(hdr, value)
-> 1097         self.endheaders(body)
   1098 
   1099     def getresponse(self, buffering=False):

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in endheaders(self, message_body)
   1051         else:
   1052             raise CannotSendHeader()
-> 1053         self._send_output(message_body)
   1054 
   1055     def request(self, method, url, body=None, headers={}):

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in _send_output(self, message_body)
    895             msg += message_body
    896             message_body = None
--> 897         self.send(msg)
    898         if message_body is not None:
    899             #message_body was not a string (i.e. it is a file) and

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in send(self, data)
    857         if self.sock is None:
    858             if self.auto_open:
--> 859                 self.connect()
```

```
    860             else:
    861                 raise NotConnected()

/Users/Hansen/anaconda/lib/python2.7/httplib.pyc in connect(self)
    834         """Connect to the host and port specified in __init__."""
    835         self.sock = self._create_connection((self.host,self.port),
--> 836                                             self.timeout, self.source_address)
    837 
    838         if self._tunnel_host:

/Users/Hansen/anaconda/lib/python2.7/socket.pyc in create_connection(address, timeout, source_address
)
    564             if source_address:
    565                 sock.bind(source_address)
--> 566             sock.connect(sa)
    567             return sock
    568 

/Users/Hansen/anaconda/lib/python2.7/socket.pyc in meth(name, self, *args)
    226 
    227 def meth(name,self,*args):
--> 228     return getattr(self._sock,name)(*args)
    229 
    230 for _m in _socketmethods:

KeyboardInterrupt:

In [49]: for ticker in ['AAPL', 'IBM', 'MSFT']:
        all_data[ticker] = web.get_data_(ticker, '1/1/2000', '1/1/2010')
web.get_data_famafrench  web.get_data_fred        web.get_data_google      web.get_data_yahoo

In [49]: for ticker in ['AAPL', 'IBM', 'MSFT']:
        all_data[ticker] = web.get_data_f(ticker, '1/1/2000', '1/1/2010')
web.get_data_famafrench  web.get_data_fred

In [49]: for ticker in ['AAPL', 'IBM', 'MSFT']:
        all_data[ticker] = web.get_data_fred(ticker, '1/1/2000', '1/1/2010')
   ....:
---------------------------------------------------------------------------
HTTPError                                 Traceback (most recent call last)
<ipython-input-49-8d38eaaae762> in <module>()
      1 for ticker in ['AAPL', 'IBM', 'MSFT']:
----> 2         all_data[ticker] = web.get_data_fred(ticker, '1/1/2000', '1/1/2010')
      3 

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in get_data_fred(name, start, en
d)
    513                                   "a valid FRED series.".format(name))
    514             raise
--> 515     df = concat([fetch_data(url, n) for url, n in zip(urls, names)],
    516                 axis=1, join='outer')
    517     return df

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in fetch_data(url, name)
    502 
    503     def fetch_data(url, name):
--> 504         with urlopen(url) as resp:
    505             data = read_csv(resp, index_col=0, parse_dates=True,
    506                             header=None, skiprows=1, names=["DATE", name],

/Users/Hansen/anaconda/lib/python2.7/contextlib.pyc in __enter__(self)
     15     def __enter__(self):
     16         try:
---> 17             return self.gen.next()
     18         except StopIteration:
     19             raise RuntimeError("generator didn't yield")

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/common.pyc in urlopen(*args, **kwargs)
```

```
     49     @contextmanager
     50     def urlopen(*args, **kwargs):
---> 51         with closing(_urlopen(*args, **kwargs)) as f:
     52             yield f
     53 

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in urlopen(url, data, timeout, cafile, capath, cadef
ault, context)
    152     else:
    153         opener = _opener
--> 154     return opener.open(url, data, timeout)
    155 
    156 def install_opener(opener):

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in open(self, fullurl, data, timeout)
    433         for processor in self.process_response.get(protocol, []):
    434             meth = getattr(processor, meth_name)
--> 435             response = meth(req, response)
    436 
    437         return response

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in http_response(self, request, response)
    546         if not (200 <= code < 300):
    547             response = self.parent.error(
--> 548                 'http', request, response, code, msg, hdrs)
    549 
    550         return response

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in error(self, proto, *args)
    465             http_err = 0
    466         args = (dict, proto, meth_name) + args
--> 467         result = self._call_chain(*args)
    468         if result:
    469             return result

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in _call_chain(self, chain, kind, meth_name, *args)
    405             func = getattr(handler, meth_name)
    406 
--> 407             result = func(*args)
    408             if result is not None:
    409                 return result

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in http_error_302(self, req, fp, code, msg, headers)
    652         fp.close()
    653 
--> 654         return self.parent.open(new, timeout=req.timeout)
    655 
    656     http_error_301 = http_error_303 = http_error_307 = http_error_302

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in open(self, fullurl, data, timeout)
    433         for processor in self.process_response.get(protocol, []):
    434             meth = getattr(processor, meth_name)
--> 435             response = meth(req, response)
    436 
    437         return response

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in http_response(self, request, response)
    546         if not (200 <= code < 300):
    547             response = self.parent.error(
--> 548                 'http', request, response, code, msg, hdrs)
    549 
    550         return response

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in error(self, proto, *args)
    465             http_err = 0
    466         args = (dict, proto, meth_name) + args
--> 467         result = self._call_chain(*args)
    468         if result:
```

```
    469             return result

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in _call_chain(self, chain, kind, meth_name, *args)
    405             func = getattr(handler, meth_name)
    406 
--> 407             result = func(*args)
    408             if result is not None:
    409                 return result

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in http_error_302(self, req, fp, code, msg, headers)
    652         fp.close()
    653 
--> 654         return self.parent.open(new, timeout=req.timeout)
    655 
    656     http_error_301 = http_error_303 = http_error_307 = http_error_302

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in open(self, fullurl, data, timeout)
    433         for processor in self.process_response.get(protocol, []):
    434             meth = getattr(processor, meth_name)
--> 435             response = meth(req, response)
    436 
    437         return response

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in http_response(self, request, response)
    546         if not (200 <= code < 300):
    547             response = self.parent.error(
--> 548                 'http', request, response, code, msg, hdrs)
    549 
    550         return response

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in error(self, proto, *args)
    471         if http_err:
    472             args = (dict, 'default', 'http_error_default') + orig_args
--> 473             return self._call_chain(*args)
    474 
    475 # XXX probably also want an abstract factory that knows when it makes

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in _call_chain(self, chain, kind, meth_name, *args)
    405             func = getattr(handler, meth_name)
    406 
--> 407             result = func(*args)
    408             if result is not None:
    409                 return result

/Users/Hansen/anaconda/lib/python2.7/urllib2.pyc in http_error_default(self, req, fp, code, msg, hdrs
)
    554 class HTTPDefaultErrorHandler(BaseHandler):
    555     def http_error_default(self, req, fp, code, msg, hdrs):
--> 556         raise HTTPError(req.get_full_url(), code, msg, hdrs, fp)
    557 
    558 class HTTPRedirectHandler(BaseHandler):

HTTPError: HTTP Error 404: Not Found

In [50]: for ticker in ['AAPL', 'IBM', 'MSFT']:
        all_data[ticker] = web.get_data_(ticker, '1/1/2000', '1/1/2010')
web.get_data_famafrench  web.get_data_fred        web.get_data_google      web.get_data_yahoo

In [50]: for ticker in ['AAPL', 'IBM', 'MSFT']:
        all_data[ticker] = web.get_data_famafrench(ticker, '1/1/2000', '1/1/2010')
   ....:
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-50-02481d233382> in <module>()
      1 for ticker in ['AAPL', 'IBM', 'MSFT']:
----> 2         all_data[ticker] = web.get_data_famafrench(ticker, '1/1/2000', '1/1/2010')
      3 
```

```
TypeError: get_data_famafrench() takes exactly 1 argument (3 given)

In [51]: web.get_d
web.get_data_famafrench  web.get_data_fred        web.get_data_google      web.get_data_yahoo

In [51]: web.get_data_google?
Signature: web.get_data_google(symbols=None, start=None, end=None, retry_count=3, pause=0.001, adjust
_price=False, ret_index=False, chunksize=25)
Docstring:
Returns DataFrame/Panel of historical stock prices from symbols, over date
range, start to end. To avoid being penalized by Google Finance servers,
pauses between downloading 'chunks' of symbols can be specified.

Parameters
----------
symbols : string, array-like object (list, tuple, Series), or DataFrame
    Single stock symbol (ticker), array-like object of symbols or
    DataFrame with index containing stock symbols.
start : string, (defaults to '1/1/2010')
    Starting date, timestamp. Parses many different kind of date
    representations (e.g., 'JAN-01-2010', '1/1/10', 'Jan, 1, 1980')
end : string, (defaults to today)
    Ending date, timestamp. Same format as starting date.
retry_count : int, default: 3
    Number of times to retry query request.
pause : numeric, default: 0.001
    Time, in seconds, to pause between consecutive queries of chunks. If
    single value given for symbol, represents the pause between retries.
chunksize : int, default: 25
    Number of symbols to download consecutively before intiating pause.
ret_index : bool, default: False
    If True, includes a simple return index 'Ret_Index' in hist_data.

Returns
-------
hist_data : DataFrame (str) or Panel (array-like object, DataFrame)
File:      ~/anaconda/lib/python2.7/site-packages/pandas/io/data.py
Type:      function

In [52]: for ticker in ['AAPL', 'IBM', 'MSFT']:
        all_data[ticker] = web.get_data_yahoo(ticker, '1/1/2000', '1/1/2010')
   ....:
---------------------------------------------------------------------------
IOError                                   Traceback (most recent call last)
<ipython-input-52-7a0d1f560d3e> in <module>()
      1 for ticker in ['AAPL', 'IBM', 'MSFT']:
----> 2         all_data[ticker] = web.get_data_yahoo(ticker, '1/1/2000', '1/1/2010')
      3

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in get_data_yahoo(symbols, start
, end, retry_count, pause, adjust_price, ret_index, chunksize, interval)
    438         raise ValueError("Invalid interval: valid values are 'd', 'w', 'm' and 'v'")
    439     return _get_data_from(symbols, start, end, interval, retry_count, pause,
--> 440                           adjust_price, ret_index, chunksize, 'yahoo')
    441
    442

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_data_from(symbols, start
, end, interval, retry_count, pause, adjust_price, ret_index, chunksize, source)
    379     # If a single symbol, (e.g., 'GOOG')
    380     if isinstance(symbols, (compat.string_types, int)):
--> 381         hist_data = src_fn(symbols, start, end, interval, retry_count, pause)
    382     # Or multiple symbols, (e.g., ['GOOG', 'AAPL', 'MSFT'])
    383     elif isinstance(symbols, DataFrame):

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _get_hist_yahoo(sym, start, e
nd, interval, retry_count, pause)
    222             '&g=%s' % interval +
    223             '&ignore=.csv')
--> 224     return _retry_read_url(url, retry_count, pause, 'Yahoo!')
    225
    226

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/io/data.py in _retry_read_url(url, retry_co
unt, pause, name)
    199
    200     raise IOError("after %d tries, %s did not "
--> 201                   "return a 200 for url %r" % (retry_count, name, url))
    202
    203

IOError: after 3 tries, Yahoo! did not return a 200 for url 'http://ichart.finance.yahoo.com/table.cs
v?s=AAPL&a=0&b=1&c=2000&d=0&e=1&f=2010&g=d&ignore=.csv'

In [53]: frame = pd.DataFrame(np.arange(12).reshape((4,3)), index = [['a', 'a', 'b', 'b'], [1, 2, 1,
2]], columns = [['Ohio', 'Ohio', 'Colorado'], ['Green', 'Red', 'Green']])

In [54]: frame
Out[54]:
     Ohio     Colorado
    Green Red    Green
a 1     0   1        2
  2     3   4        5
b 1     6   7        8
  2     9  10       11

In [55]: frame.index
Out[55]:
MultiIndex(levels=[[u'a', u'b'], [1, 2]],
           labels=[[0, 0, 1, 1], [0, 1, 0, 1]])

In [56]: frame.columns
Out[56]:
MultiIndex(levels=[[u'Colorado', u'Ohio'], [u'Green', u'Red']],
           labels=[[1, 1, 0], [0, 1, 0]])

In [57]: frame.index.n
frame.index.name     frame.index.nbytes   frame.index.nlevels
frame.index.names    frame.index.ndim     frame.index.nunique

In [57]: frame.index.na
frame.index.name   frame.index.names

In [57]: frame.index.name = ['key1', 'key2']

In [58]: frame.columns.names = ['state', 'color']

In [59]: frame
Out[59]:
state  Ohio     Colorado
color Green Red    Green
a 1       0   1        2
  2       3   4        5
b 1       6   7        8
  2       9  10       11

In [60]: frame.index.names = ['key1', 'key2']

In [61]: frame
Out[61]:
state      Ohio     Colorado
color     Green Red    Green
key1 key2
a    1        0   1        2
     2        3   4        5
b    1        6   7        8
```

```
     2        9  10       11

In [62]: frame.sum(level = 'key2' )
Out[62]:
state  Ohio     Colorado
color Green Red    Green
key2
1         6   8       10
2        12  14       16

In [63]: frame.sum(level = 'key2', axis = 1 )
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
<ipython-input-63-a495465d562e> in <module>()
----> 1 frame.sum(level = 'key2', axis = 1 )

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/generic.pyc in stat_func(self, axis, s
kipna, level, numeric_only, **kwargs)
   5306         if level is not None:
   5307             return self._agg_by_level(name, axis=axis, level=level,
-> 5308                                       skipna=skipna)
   5309         return self._reduce(f, name, axis=axis, skipna=skipna,
   5310                             numeric_only=numeric_only)

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/generic.pyc in _agg_by_level(self, nam
e, axis, level, skipna, **kwargs)
   5016
   5017     def _agg_by_level(self, name, axis=0, level=0, skipna=True, **kwargs):
-> 5018         grouped = self.groupby(level=level, axis=axis)
   5019         if hasattr(grouped, name) and skipna:
   5020             return getattr(grouped, name)(**kwargs)

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/generic.pyc in groupby(self, by, axis,
 level, as_index, sort, group_keys, squeeze, **kwargs)
   3776         return groupby(self, by=by, axis=axis, level=level, as_index=as_index,
   3777                        sort=sort, group_keys=group_keys, squeeze=squeeze,
-> 3778                        **kwargs)
   3779
   3780     def asfreq(self, freq, method=None, how=None, normalize=False):

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/groupby.pyc in groupby(obj, by, **kwds
)
   1425         raise TypeError('invalid type: %s' % type(obj))
   1426
-> 1427     return klass(obj, by, **kwds)
   1428
   1429

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/groupby.pyc in __init__(self, obj, key
s, axis, level, grouper, exclusions, selection, as_index, sort, group_keys, squeeze, **kwargs)
    352                                                     level=level,
    353                                                     sort=sort,
--> 354                                                     mutated=self.mutated)
    355
    356         self.obj = obj

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/groupby.pyc in _get_grouper(obj, key,
axis, level, sort, mutated)
   2400                         sort=sort,
   2401                         in_axis=in_axis) \
-> 2402             if not isinstance(gpr, Grouping) else gpr
   2403
   2404         groupings.append(ping)

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/groupby.pyc in __init__(self, index, g
rouper, obj, name, level, sort, in_axis)
   2125             if not isinstance(level, int):
   2126                 if level not in index.names:
```

```
-> 2127                     raise AssertionError('Level %s not in index' % str(level))
   2128                 level = index.names.index(level)
   2129 

AssertionError: Level key2 not in index
```

```
In [64]: frame.sum(level = 'color')
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
<ipython-input-64-d0ad6da94717> in <module>()
----> 1 frame.sum(level = 'color')

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/generic.pyc in stat_func(self, axis, s
kipna, level, numeric_only, **kwargs)
   5306         if level is not None:
   5307             return self._agg_by_level(name, axis=axis, level=level,
-> 5308                                       skipna=skipna)
   5309         return self._reduce(f, name, axis=axis, skipna=skipna,
   5310                             numeric_only=numeric_only)

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/generic.pyc in _agg_by_level(self, nam
e, axis, level, skipna, **kwargs)
   5016 
   5017     def _agg_by_level(self, name, axis=0, level=0, skipna=True, **kwargs):
-> 5018         grouped = self.groupby(level=level, axis=axis)
   5019         if hasattr(grouped, name) and skipna:
   5020             return getattr(grouped, name)(**kwargs)

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/generic.pyc in groupby(self, by, axis,
 level, as_index, sort, group_keys, squeeze, **kwargs)
   3776         return groupby(self, by=by, axis=axis, level=level, as_index=as_index,
   3777                        sort=sort, group_keys=group_keys, squeeze=squeeze,
-> 3778                        **kwargs)
   3779 
   3780     def asfreq(self, freq, method=None, how=None, normalize=False):

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/groupby.pyc in groupby(obj, by, **kwds
)
   1425         raise TypeError('invalid type: %s' % type(obj))
   1426 
-> 1427     return klass(obj, by, **kwds)
   1428 
   1429 

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/groupby.pyc in __init__(self, obj, key
s, axis, level, grouper, exclusions, selection, as_index, sort, group_keys, squeeze, **kwargs)
    352                                                     level=level,
    353                                                     sort=sort,
--> 354                                                     mutated=self.mutated)
    355 
    356         self.obj = obj

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/groupby.pyc in _get_grouper(obj, key,
axis, level, sort, mutated)
   2400                         sort=sort,
   2401                         in_axis=in_axis) \
-> 2402             if not isinstance(gpr, Grouping) else gpr
   2403 
   2404         groupings.append(ping)

/Users/Hansen/anaconda/lib/python2.7/site-packages/pandas/core/groupby.pyc in __init__(self, index, g
rouper, obj, name, level, sort, in_axis)
   2125             if not isinstance(level, int):
   2126                 if level not in index.names:
-> 2127                     raise AssertionError('Level %s not in index' % str(level))
   2128                 level = index.names.index(level)
   2129 
```