Chapter 3 - Pandas

In [1]:
```
from os import path
fname = path.expanduser(r'C:\Users\danal\Desktop\Ex_Files_Data_Science_Python\Exe
```

In [2]:
```
print(r'C:\Users\danal\Desktop\Ex_Files_Data_Science_Python\Exercise Files\Ch04\0
```

```
C:\Users\danal\Desktop\Ex_Files_Data_Science_Python\Exercise Files\Ch04\04_02\t
rack.csv
```

In [6]:
```
path.getsize(fname)   # The file size is in bytes.
```

Out[6]:
```
43844
```

In [11]:
```
path.getsize(fname) / (1<<10)     # The file size is in kilobytes.
```

Out[11]:
```
42.81640625
```

In [12]:
```
!head "$fname"    # It doesn't work in this version.
```

```
'head' is not recognized as an internal or external command,
operable program or batch file.
```

In [13]:
```
!ls -lh "$fname"     # It doesn't work in this version.
```

```
'ls' is not recognized as an internal or external command,
operable program or batch file.
```

In [14]:
```
with open(fname) as fp:
    for lnum, line in enumerate(fp):    # Enumerate is a function that gives us t
        if lnum > 10:
            break
        print(line[:-1])
```

```
time,lat,lng,height
2015-08-20 03:48:07.235,35.015021,32.519585,136.1999969482422
2015-08-20 03:48:24.734,35.014954,32.519606,126.5999984741211
2015-08-20 03:48:25.660,35.014871,32.519612,123.0
2015-08-20 03:48:26.819,35.014824,32.519654,120.5
2015-08-20 03:48:27.828,35.014776,32.519689,118.9000015258789
2015-08-20 03:48:29.720,35.014704,32.519691,119.9000015258789
2015-08-20 03:48:30.669,35.014657,32.519734,120.9000015258789
2015-08-20 03:48:33.793,35.014563,32.519719,121.69999694824219
2015-08-20 03:48:34.869,35.014549,32.519694,121.19999694824219
2015-08-20 03:48:37.708,35.014515,32.519625,121.69999694824219
```

In [15]:
```
!wc -l "$fname"      # This command doesn't work in this version.
```

```
'wc' is not recognized as an internal or external command,
operable program or batch file.
```

In [16]:
```
with open(fname) as fp:
    print(sum(1 for line in fp))
```

```
741
```

In [17]:
```
import pandas as pd
```

In [18]:
```
df = pd.read_csv(fname)
```

In [19]:
```
len(df)
```

Out[19]:
```
740
```

In [20]:
```
df.columns
```

Out[20]:
```
Index(['time', 'lat', 'lng', 'height'], dtype='object')
```

In [21]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 740 entries, 0 to 739
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   time    740 non-null    object 
 1   lat     740 non-null    float64
 2   lng     740 non-null    float64
 3   height  740 non-null    float64
dtypes: float64(3), object(1)
memory usage: 23.2+ KB
```

In [22]:
```
df.head()
```

Out[22]:

	time	lat	lng	height
0	2015-08-20 03:48:07.235	35.015021	32.519585	136.199997
1	2015-08-20 03:48:24.734	35.014954	32.519606	126.599998
2	2015-08-20 03:48:25.660	35.014871	32.519612	123.000000
3	2015-08-20 03:48:26.819	35.014824	32.519654	120.500000
4	2015-08-20 03:48:27.828	35.014776	32.519689	118.900002

In [23]:
```
df.dtypes
```

Out[23]:
```
time       object
lat       float64
lng       float64
height    float64
dtype: object
```

In [27]:
```
df = pd.read_csv(fname, parse_dates=['time'])
```

In [28]:
```
df.dtypes
```

Out[28]:
```
time      datetime64[ns]
lat              float64
lng              float64
height           float64
dtype: object
```

In [29]:
```
df['lat']
```

Out[29]:
```
0      35.015021
1      35.014954
2      35.014871
3      35.014824
4      35.014776
         ...
735    35.014387
736    35.014355
737    35.014279
738    35.014264
739    35.014212
Name: lat, Length: 740, dtype: float64
```

In [30]:
```
df.lat
```

Out[30]:
```
0      35.015021
1      35.014954
2      35.014871
3      35.014824
4      35.014776
         ...
735    35.014387
736    35.014355
737    35.014279
738    35.014264
739    35.014212
Name: lat, Length: 740, dtype: float64
```

In [32]:
```
df[['lat', 'lng']]
```

Out[32]:

	lat	lng
0	35.015021	32.519585
1	35.014954	32.519606
2	35.014871	32.519612
3	35.014824	32.519654
4	35.014776	32.519689
...	...	...
735	35.014387	32.517020
736	35.014355	32.517035
737	35.014279	32.517087
738	35.014264	32.517098
739	35.014212	32.517142

740 rows × 2 columns

In [34]:
```
df['lat'][0]
```

Out[34]:
```
35.015021000000004
```

In [35]:
```
df.loc[0]
```

Out[35]:
```
time      2015-08-20 03:48:07.235000
lat                           35.015
lng                          32.5196
height                         136.2
Name: 0, dtype: object
```

In [36]:
```
df.loc[2:7]
```

Out[36]:

	time	lat	lng	height
2	2015-08-20 03:48:25.660	35.014871	32.519612	123.000000
3	2015-08-20 03:48:26.819	35.014824	32.519654	120.500000
4	2015-08-20 03:48:27.828	35.014776	32.519689	118.900002
5	2015-08-20 03:48:29.720	35.014704	32.519691	119.900002
6	2015-08-20 03:48:30.669	35.014657	32.519734	120.900002
7	2015-08-20 03:48:33.793	35.014563	32.519719	121.699997

In [37]:
```
df.loc[2:7]
```

Out[37]:

	time	lat	lng	height
2	2015-08-20 03:48:25.660	35.014871	32.519612	123.000000
3	2015-08-20 03:48:26.819	35.014824	32.519654	120.500000
4	2015-08-20 03:48:27.828	35.014776	32.519689	118.900002
5	2015-08-20 03:48:29.720	35.014704	32.519691	119.900002
6	2015-08-20 03:48:30.669	35.014657	32.519734	120.900002
7	2015-08-20 03:48:33.793	35.014563	32.519719	121.699997

In [39]:
```
df[['lat', 'lng']][2:7]
```

Out[39]:

	lat	lng
2	35.014871	32.519612
3	35.014824	32.519654
4	35.014776	32.519689
5	35.014704	32.519691
6	35.014657	32.519734

In [40]:
```
df.index
```

Out[40]:
```
RangeIndex(start=0, stop=740, step=1)
```

In [41]:
```
import numpy as np
```

In [45]:
```
df1 = pd.DataFrame(np.arange(10).reshape((5,2)), columns=['x', 'y'], index=['a',
df1
```

Out[45]:

	x	y
a	0	1
b	2	3
c	4	5
d	6	7
e	8	9

In [46]:
```
df1.loc['a']
```

Out[46]:
```
x    0
y    1
Name: a, dtype: int32
```

In [47]:
```
df1.loc['b':'d']
```

Out[47]:

	x	y
b	2	3
c	4	5
d	6	7

In [48]:
```
df.index
```

Out[48]:
```
RangeIndex(start=0, stop=740, step=1)
```

In [49]:
```
df.index = df['time']
df.index
```

Out[49]:
```
DatetimeIndex(['2015-08-20 03:48:07.235000', '2015-08-20 03:48:24.734000',
               '2015-08-20 03:48:25.660000', '2015-08-20 03:48:26.819000',
               '2015-08-20 03:48:27.828000', '2015-08-20 03:48:29.720000',
               '2015-08-20 03:48:30.669000', '2015-08-20 03:48:33.793000',
               '2015-08-20 03:48:34.869000', '2015-08-20 03:48:37.708000',
               ...
               '2015-08-20 04:20:18.844000', '2015-08-20 04:20:21.996000',
               '2015-08-20 04:20:22.897000', '2015-08-20 04:20:24.905000',
               '2015-08-20 04:20:25.835000', '2015-08-20 04:20:28.982000',
               '2015-08-20 04:20:29.923000', '2015-08-20 04:20:32.863000',
               '2015-08-20 04:20:33.994000', '2015-08-20 04:20:42.329000'],
              dtype='datetime64[ns]', name='time', length=740, freq=None)
```

In [51]:
```
df.loc['2015-08-20 04:18:54']
```

Out[51]:

	time	lat	lng	height
time				
2015-08-20 04:18:54.007	2015-08-20 04:18:54.007	35.015942	32.515209	117.099998
2015-08-20 04:18:54.893	2015-08-20 04:18:54.893	35.015937	32.515240	117.500000

In [52]:
```
df.loc['2015-08-20 03:48']
```

Out[52]:

time	time	lat	lng	height
2015-08-20 03:48:07.235	2015-08-20 03:48:07.235	35.015021	32.519585	136.199997
2015-08-20 03:48:24.734	2015-08-20 03:48:24.734	35.014954	32.519606	126.599998
2015-08-20 03:48:25.660	2015-08-20 03:48:25.660	35.014871	32.519612	123.000000
2015-08-20 03:48:26.819	2015-08-20 03:48:26.819	35.014824	32.519654	120.500000
2015-08-20 03:48:27.828	2015-08-20 03:48:27.828	35.014776	32.519689	118.900002
2015-08-20 03:48:29.720	2015-08-20 03:48:29.720	35.014704	32.519691	119.900002
2015-08-20 03:48:30.669	2015-08-20 03:48:30.669	35.014657	32.519734	120.900002
2015-08-20 03:48:33.793	2015-08-20 03:48:33.793	35.014563	32.519719	121.699997
2015-08-20 03:48:34.869	2015-08-20 03:48:34.869	35.014549	32.519694	121.199997
2015-08-20 03:48:37.708	2015-08-20 03:48:37.708	35.014515	32.519625	121.699997
2015-08-20 03:48:38.839	2015-08-20 03:48:38.839	35.014505	32.519599	121.800003
2015-08-20 03:48:41.980	2015-08-20 03:48:41.980	35.014481	32.519514	122.599998
2015-08-20 03:48:42.725	2015-08-20 03:48:42.725	35.014472	32.519486	123.000000
2015-08-20 03:48:45.896	2015-08-20 03:48:45.896	35.014439	32.519405	122.699997
2015-08-20 03:48:46.662	2015-08-20 03:48:46.662	35.014432	32.519379	122.699997
2015-08-20 03:48:49.829	2015-08-20 03:48:49.829	35.014414	32.519309	122.699997
2015-08-20 03:48:50.665	2015-08-20 03:48:50.665	35.014400	32.519287	123.300003
2015-08-20 03:48:53.692	2015-08-20 03:48:53.692	35.014372	32.519211	122.300003
2015-08-20 03:48:54.662	2015-08-20 03:48:54.662	35.014365	32.519187	122.599998
2015-08-20 03:48:58.869	2015-08-20 03:48:58.869	35.014337	32.519106	122.000000
2015-08-20 03:48:59.663	2015-08-20 03:48:59.663	35.014331	32.519084	121.800003

In [53]:
```
import pytz
```

In [54]:
```
ts = df.index[0]
```

In [56]:
```
ts.tz_localize(pytz.UTC)
```

Out[56]:
```
Timestamp('2015-08-20 03:48:07.235000+0000', tz='UTC')
```

In [57]:
```
ts.tz_localize(pytz.UTC).tz_convert(pytz.timezone('Asia/Jerusalem'))
```

Out[57]:
```
Timestamp('2015-08-20 06:48:07.235000+0300', tz='Asia/Jerusalem')
```

In [60]:
```
df.index = df.index.tz_localize(pytz.UTC).tz_convert(pytz.timezone('Asia/Jerusale
df.index[:10]
```

Out[60]:
```
DatetimeIndex(['2015-08-20 06:48:07.235000+03:00',
               '2015-08-20 06:48:24.734000+03:00',
               '2015-08-20 06:48:25.660000+03:00',
               '2015-08-20 06:48:26.819000+03:00',
               '2015-08-20 06:48:27.828000+03:00',
               '2015-08-20 06:48:29.720000+03:00',
               '2015-08-20 06:48:30.669000+03:00',
               '2015-08-20 06:48:33.793000+03:00',
               '2015-08-20 06:48:34.869000+03:00',
               '2015-08-20 06:48:37.708000+03:00'],
              dtype='datetime64[ns, Asia/Jerusalem]', name='time', freq=None)
```

In [61]:
```
%pwd
```

Out[61]:
```
'C:\\Users\\danal'
```

In [62]:
```
import geo
```

In [63]:
```
import sys
sys.path
```

Out[63]:
```
['C:\\Users\\danal',
 'C:\\Users\\danal\\anaconda3\\python37.zip',
 'C:\\Users\\danal\\anaconda3\\DLLs',
 'C:\\Users\\danal\\anaconda3\\lib',
 'C:\\Users\\danal\\anaconda3',
 '',
 'C:\\Users\\danal\\anaconda3\\lib\\site-packages',
 'C:\\Users\\danal\\anaconda3\\lib\\site-packages\\win32',
 'C:\\Users\\danal\\anaconda3\\lib\\site-packages\\win32\\lib',
 'C:\\Users\\danal\\anaconda3\\lib\\site-packages\\Pythonwin',
 'C:\\Users\\danal\\anaconda3\\lib\\site-packages\\IPython\\extensions',
 'C:\\Users\\danal\\.ipython']
```

In [69]:
```
??geo
```

In [66]:
```
from geo import circle_dist
```

In [67]:
```
lat1, lng1 = df.iloc[0].lat, df.iloc[0].lng
lat2, lng2 = df.iloc[1].lat, df.iloc[1].lng
```

In [68]:
```
circle_dist(lat1, lng1, lat2, lng2)
```

Out[68]:
```
0.007693931535344109
```

In [70]:
```
s = pd.Series(np.arange(5))
```

In [71]:
```
s
```

Out[71]:
```
0    0
1    1
2    2
3    3
4    4
dtype: int32
```

In [72]:
```
s.shift()
```

Out[72]:
```
0    NaN
1    0.0
2    1.0
3    2.0
4    3.0
dtype: float64
```

In [73]:
```
s.shift(-1)
```

Out[73]:
```
0    1.0
1    2.0
2    3.0
3    4.0
4    NaN
dtype: float64
```

In [74]:
```
dist = circle_dist(df['lat'], df['lng'], df['lat'].shift(), df['lng'].shift())
```

In [75]:
```
dist[:10]
```

Out[75]:
```
time
2015-08-20 06:48:07.235000+03:00         NaN
2015-08-20 06:48:24.734000+03:00    0.007694
2015-08-20 06:48:25.660000+03:00    0.009248
2015-08-20 06:48:26.819000+03:00    0.006479
2015-08-20 06:48:27.828000+03:00    0.006219
2015-08-20 06:48:29.720000+03:00    0.008010
2015-08-20 06:48:30.669000+03:00    0.006533
2015-08-20 06:48:33.793000+03:00    0.010545
2015-08-20 06:48:34.869000+03:00    0.002759
2015-08-20 06:48:37.708000+03:00    0.007336
dtype: float64
```

In [77]:
```
dist.sum()
```

Out[77]:
```
4.688135968432568
```

In [79]:
```
dt = df['time'] - df['time'].shift()
```

In [80]:
```
dt[:10]
```

Out[80]:
```
time
2015-08-20 06:48:07.235000+03:00                NaT
2015-08-20 06:48:24.734000+03:00   00:00:17.499000
2015-08-20 06:48:25.660000+03:00   00:00:00.926000
2015-08-20 06:48:26.819000+03:00   00:00:01.159000
2015-08-20 06:48:27.828000+03:00   00:00:01.009000
2015-08-20 06:48:29.720000+03:00   00:00:01.892000
2015-08-20 06:48:30.669000+03:00   00:00:00.949000
2015-08-20 06:48:33.793000+03:00   00:00:03.124000
2015-08-20 06:48:34.869000+03:00   00:00:01.076000
2015-08-20 06:48:37.708000+03:00   00:00:02.839000
Name: time, dtype: timedelta64[ns]
```

In [81]:
```
dt.sum()
```

Out[81]:
```
Timedelta('0 days 00:32:35.094000')
```

In [82]:
```
dt[1].total_seconds()
```

Out[82]:
```
17.499
```

In [83]:
```
dt[1] / np.timedelta64(1, 'h')
```

Out[83]:
```
0.004860833333333333
```

In [84]:
```
dt[1].total_seconds()/3600
```

Out[84]:
```
0.004860833333333333
```

In [89]:
```
speed = dist / (dt / np.timedelta64(1, 'h'))
```

In [91]:
```
speed[:10]
```

Out[91]:
```
time
2015-08-20 06:48:07.235000+03:00          NaN
2015-08-20 06:48:24.734000+03:00     1.582842
2015-08-20 06:48:25.660000+03:00    35.954340
2015-08-20 06:48:26.819000+03:00    20.123165
2015-08-20 06:48:27.828000+03:00    22.187213
2015-08-20 06:48:29.720000+03:00    15.241680
2015-08-20 06:48:30.669000+03:00    24.783839
2015-08-20 06:48:33.793000+03:00    12.151207
2015-08-20 06:48:34.869000+03:00     9.230505
2015-08-20 06:48:37.708000+03:00     9.302060
dtype: float64
```

In [92]:
```
df['dist'] = dist
df['dt'] = dt
```

In [93]:
```
df1m = df.resample('1min').sum()
```

In [94]:
```
df1m.index
```

Out[94]:
```
DatetimeIndex(['2015-08-20 06:48:00+03:00', '2015-08-20 06:49:00+03:00',
               '2015-08-20 06:50:00+03:00', '2015-08-20 06:51:00+03:00',
               '2015-08-20 06:52:00+03:00', '2015-08-20 06:53:00+03:00',
               '2015-08-20 06:54:00+03:00', '2015-08-20 06:55:00+03:00',
               '2015-08-20 06:56:00+03:00', '2015-08-20 06:57:00+03:00',
               '2015-08-20 06:58:00+03:00', '2015-08-20 06:59:00+03:00',
               '2015-08-20 07:00:00+03:00', '2015-08-20 07:01:00+03:00',
               '2015-08-20 07:02:00+03:00', '2015-08-20 07:03:00+03:00',
               '2015-08-20 07:04:00+03:00', '2015-08-20 07:05:00+03:00',
               '2015-08-20 07:06:00+03:00', '2015-08-20 07:07:00+03:00',
               '2015-08-20 07:08:00+03:00', '2015-08-20 07:09:00+03:00',
               '2015-08-20 07:10:00+03:00', '2015-08-20 07:11:00+03:00',
               '2015-08-20 07:12:00+03:00', '2015-08-20 07:13:00+03:00',
               '2015-08-20 07:14:00+03:00', '2015-08-20 07:15:00+03:00',
               '2015-08-20 07:16:00+03:00', '2015-08-20 07:17:00+03:00',
               '2015-08-20 07:18:00+03:00', '2015-08-20 07:19:00+03:00',
               '2015-08-20 07:20:00+03:00'],
              dtype='datetime64[ns, Asia/Jerusalem]', name='time', freq='T')
```

In [95]:
```
df1m.columns
```

Out[95]:
```
Index(['lat', 'lng', 'height', 'dist'], dtype='object')
```

In [96]:
```
df['dt'] = dt / np.timedelta64(1, 'h')
df1m = df.resample('1min').sum()
speed1m = df1m['dist'] / df1m['dt']
```

In [98]:
```
speed1m[:10]
```

Out[98]:
```
time
2015-08-20 06:48:00+03:00     8.127118
2015-08-20 06:49:00+03:00     7.579874
2015-08-20 06:50:00+03:00     9.127972
2015-08-20 06:51:00+03:00    10.220818
2015-08-20 06:52:00+03:00    10.114279
2015-08-20 06:53:00+03:00     9.687690
2015-08-20 06:54:00+03:00    10.856446
2015-08-20 06:55:00+03:00    10.892145
2015-08-20 06:56:00+03:00    10.270580
2015-08-20 06:57:00+03:00     6.629397
Freq: T, dtype: float64
```

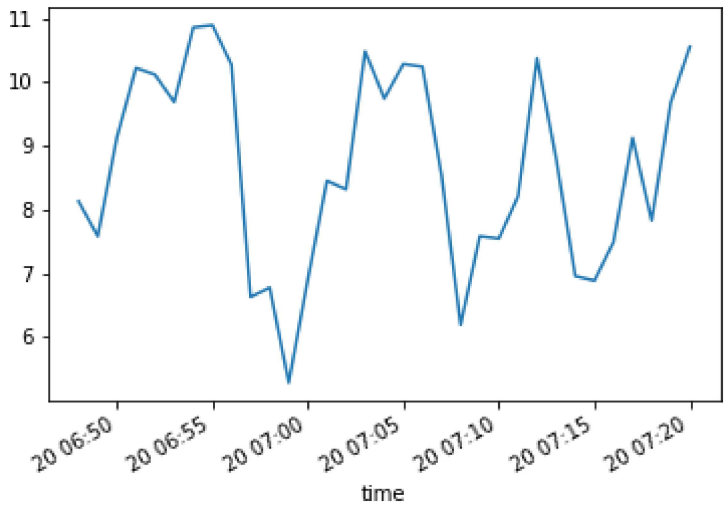
In [100]:
```
speed1m.describe()
```

Out[100]:
```
count    33.000000
mean      8.658214
std       1.543214
min       5.285595
25%       7.543402
50%       8.538120
75%      10.220818
max      10.892145
dtype: float64
```

In [102]:
```
%matplotlib inline   # We need to tell MatPlotLib to display the charts in our nc
```

In [103]:
```
speed1m.plot()
```

Out[103]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x13dcc2d6b48>
```



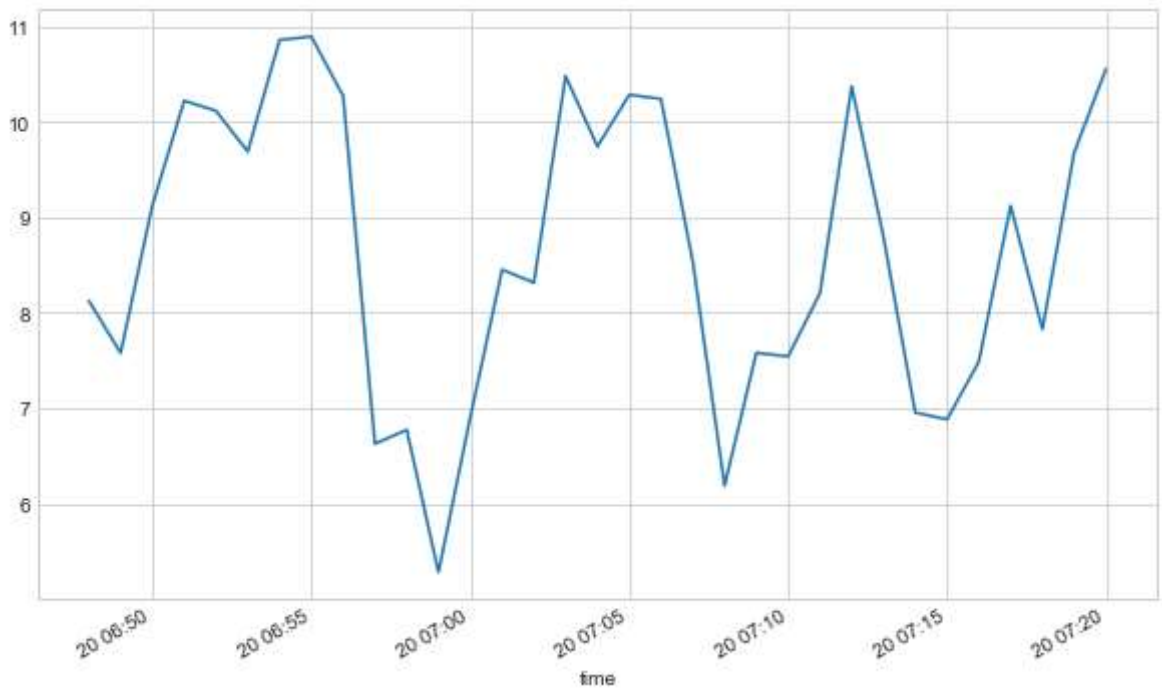
In [104]:
```
import matplotlib.pyplot as plt
```

In [105]:
```
plt.rcParams['figure.figsize'] = [10,6]
```

In [108]:
```
plt.style.use('seaborn-whitegrid')
```

In [109]:
```
speed1m.plot()
```

Out[109]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x13dd0a62a88>
```



In [110]:
```
speed1m.plot.box()
```

Out[110]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x13dd0c281c8>
```