Importing Necessary Libraries

In [1]:
```
%pylab inline
import pandas
import seaborn
```

```
Populating the interactive namespace from numpy and matplotlib
```

Loading CSV file into memory

In [2]:
```
data = pandas.read_csv(r"D:\DATA SCIENCE\Uber Data Trip\RawUberData.csv")
```

In [3]:
```
data.head()
```

Out[3]:

	Date/Time	Lat	Lon	Base
0	04-01-2014	40.7690	-73.9549	B02512
1	04-01-2014	40.7267	-74.0345	B02512
2	04-01-2014	40.7316	-73.9873	B02512
3	04-01-2014	40.7588	-73.9776	B02512
4	04-01-2014	40.7594	-73.9722	B02512

Convert Date Time

In [4]:
```
data ['Date/Time'] = data ['Date/Time'].map(pandas.to_datetime)
```

In [5]:
```
data.head()
```

Out[5]:

	Date/Time	Lat	Lon	Base
0	2014-04-01	40.7690	-73.9549	B02512
1	2014-04-01	40.7267	-74.0345	B02512
2	2014-04-01	40.7316	-73.9873	B02512
3	2014-04-01	40.7588	-73.9776	B02512
4	2014-04-01	40.7594	-73.9722	B02512

In [6]:
```
def get_dom(dt):
    return dt.day
data['dom'] = data['Date/Time'].map(get_dom)
```

In [7]:
```
data.head()
```

Out[7]:

	Date/Time	Lat	Lon	Base	dom
0	2014-04-01	40.7690	-73.9549	B02512	1
1	2014-04-01	40.7267	-74.0345	B02512	1
2	2014-04-01	40.7316	-73.9873	B02512	1
3	2014-04-01	40.7588	-73.9776	B02512	1
4	2014-04-01	40.7594	-73.9722	B02512	1

In [8]:
```
def get_weekday(dt):
    return dt.weekday()

data['weekday'] = data['Date/Time'].map(get_weekday)

def get_hour(dt):
    return dt.hour

data['hour'] = data['Date/Time'].map(get_hour)

data.tail()
```

Out[8]:

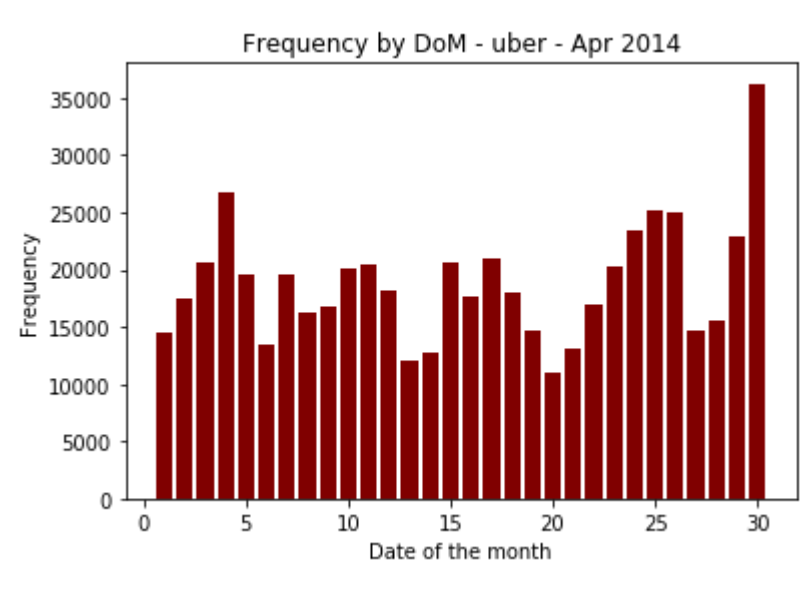
	Date/Time	Lat	Lon	Base	dom	weekday	hour
564511	2014-04-30 23:22:00	40.7640	-73.9744	B02764	30	2	23
564512	2014-04-30 23:26:00	40.7629	-73.9672	B02764	30	2	23
564513	2014-04-30 23:31:00	40.7443	-73.9889	B02764	30	2	23
564514	2014-04-30 23:32:00	40.6756	-73.9405	B02764	30	2	23
564515	2014-04-30 23:48:00	40.6880	-73.9608	B02764	30	2	23

ANALYSIS

Analysis of the DoM

In [9]:
```
hist(data.dom, bins=30, rwidth=.8, color='#800000', range=(0.5, 30.5))
xlabel('Date of the month')
ylabel('Frequency')
title('Frequency by DoM - uber - Apr 2014')
```

Out[9]:
```
Text(0.5, 1.0, 'Frequency by DoM - uber - Apr 2014')
```

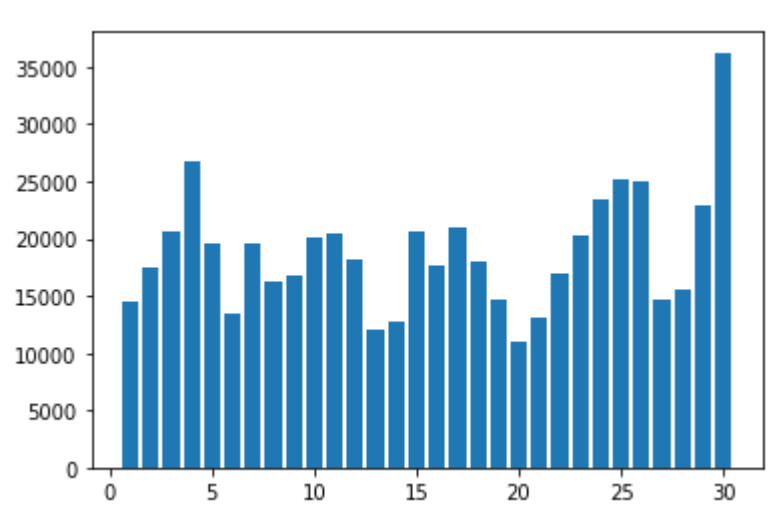


In [10]:
```
#for k, rows in data.groupby('dom'):
#print((k, len(rows)))

def count_rows(rows):
    return len(rows)

by_date = data.groupby('dom').apply(count_rows)
by_date
```

Out[10]:
```
dom
1     14546
2     17474
3     20701
4     26714
5     19521
6     13445
7     19550
8     16188
9     16843
10    20041
11    20420
12    18170
13    12112
14    12674
15    20641
16    17717
17    20973
18    18074
19    14602
20    11017
21    13162
22    16975
23    20346
24    23352
25    25095
26    24925
27    14677
28    15475
29    22835
30    36251
dtype: int64
```

In [11]:
```
bar(range(1, 31), by_date)
```

Out[11]:
```
<BarContainer object of 30 artists>
```

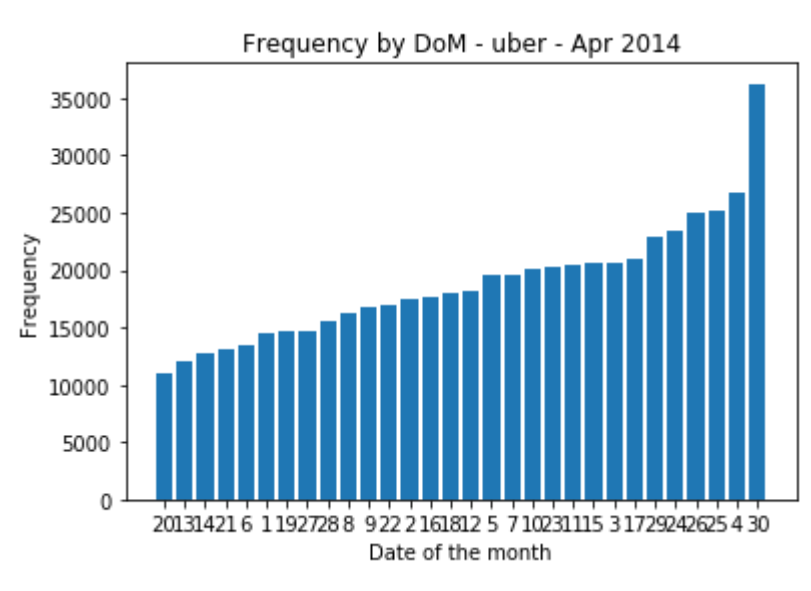


In [12]:
```
by_date_sorted = by_date.sort_values()
by_date_sorted
```

Out[12]:
```
dom
20    11017
13    12112
14    12674
21    13162
6     13445
1     14546
19    14602
27    14677
28    15475
8     16188
9     16843
22    16975
2     17474
16    17717
18    18074
12    18170
5     19521
7     19550
10    20041
23    20346
11    20420
15    20641
3     20701
17    20973
29    22835
24    23352
26    24925
25    25095
4     26714
30    36251
dtype: int64
```

In [13]:
```
bar(range(1, 31), by_date_sorted)
xticks(range(1,31), by_date_sorted.index)
xlabel('Date of the month')
ylabel('Frequency')
title('Frequency by DoM - uber - Apr 2014')
```

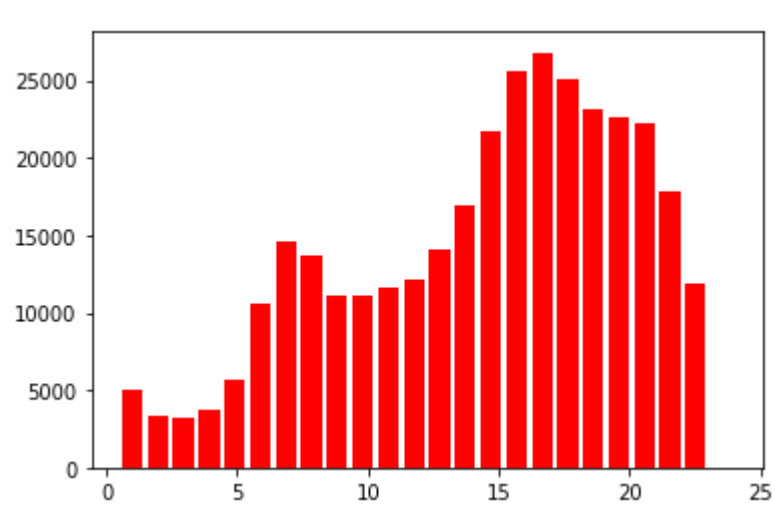
Out[13]:
```
Text(0.5, 1.0, 'Frequency by DoM - uber - Apr 2014')
```



Analysis of the hour

In [14]:
```
hist(data.hour, bins=24, range=(.5, 24), rwidth=.8, color='#FF0000')
```

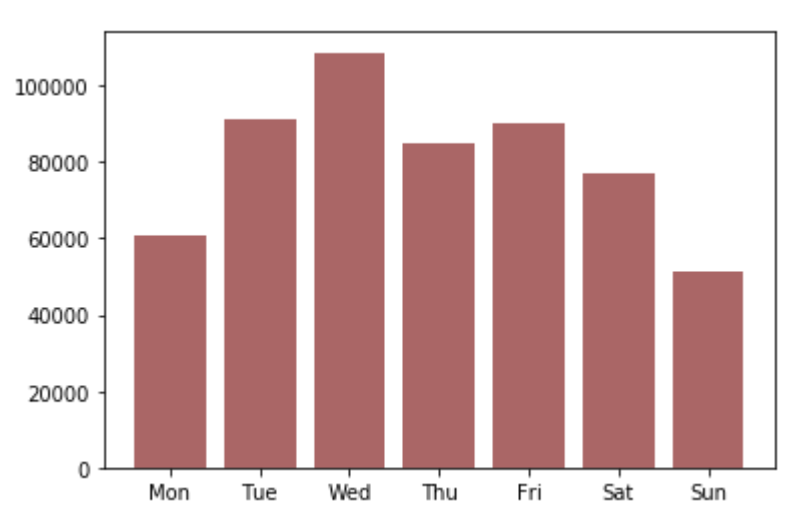
Out[14]:
```
(array([ 5009.,  3275.,  3146.,  3689.,  5675., 10637., 14562., 13687.,
        11070., 11081., 11628., 12120., 14044., 16970., 21684., 25601.,
        26802., 25123., 23175., 22583., 22276., 17870., 11851.,     0.]),
 array([ 0.5       ,  1.47916667,  2.45833333,  3.4375    ,  4.41666667,
         5.39583333,  6.375     ,  7.35416667,  8.33333333,  9.3125    ,
        10.29166667, 11.27083333, 12.25      , 13.22916667, 14.20833333,
        15.1875    , 16.16666667, 17.14583333, 18.125     , 19.10416667,
        20.08333333, 21.0625    , 22.04166667, 23.02083333, 24.        ]),
 <a list of 24 Patch objects>)
```



Analysis of the Weekday

In [15]:
```
hist(data.weekday, bins=7, range =(-.5,6.5), rwidth=.8, color='#AA6666')
xticks(range(7), 'Mon Tue Wed Thu Fri Sat Sun'.split())
```

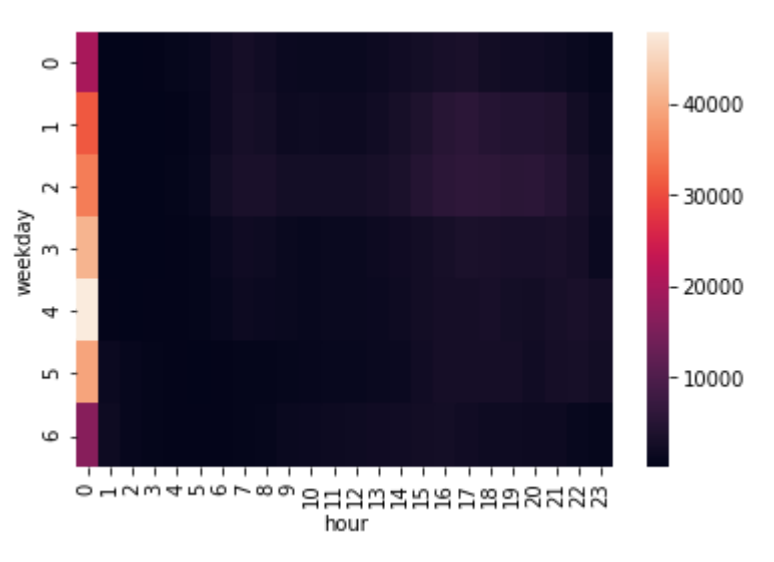
Out[15]:
```
([<matplotlib.axis.XTick at 0x2846f1a5248>,
  <matplotlib.axis.XTick at 0x2846f19f948>,
  <matplotlib.axis.XTick at 0x2846f19f608>,
  <matplotlib.axis.XTick at 0x2846f1c9288>,
  <matplotlib.axis.XTick at 0x2846f1c9688>,
  <matplotlib.axis.XTick at 0x2846f1c9f08>,
  <matplotlib.axis.XTick at 0x2846f2a4548>],
 <a list of 7 Text xticklabel objects>)
```



Color Analysis (hour,dow)

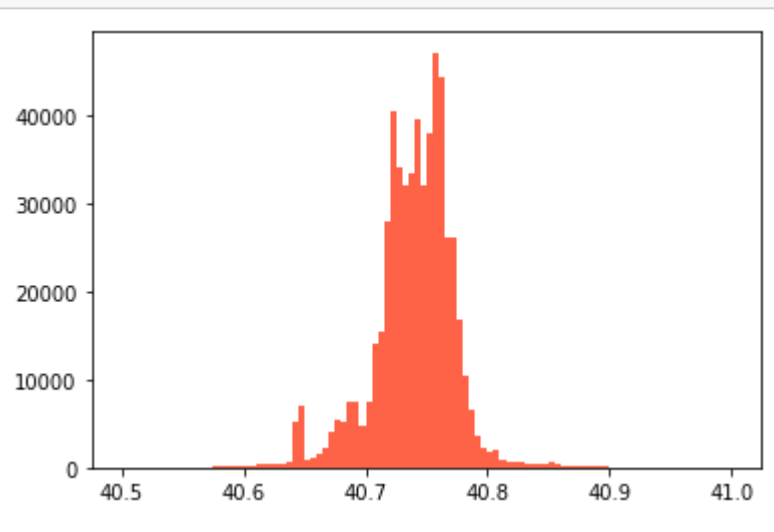
In [16]:
```
by_cross = data.groupby('weekday hour'.split()).apply(count_rows).unstack()
seaborn.heatmap(by_cross)
```

Out[16]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2846dc3e548>
```

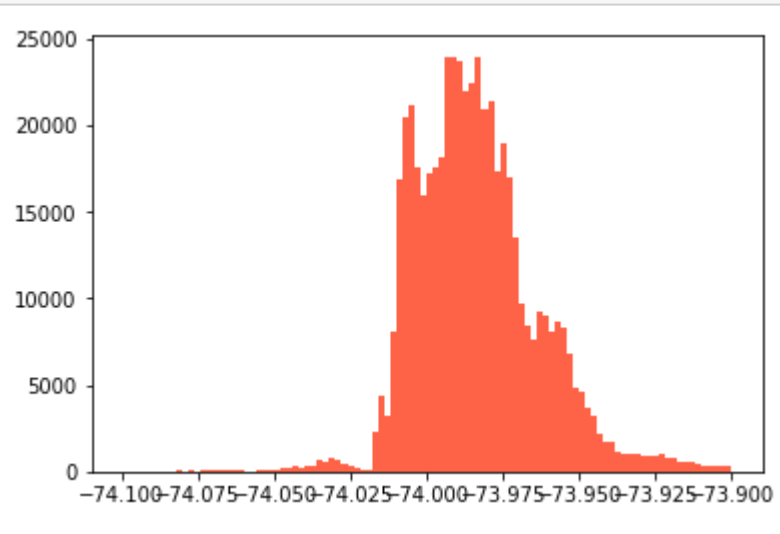


Lat and Lon

In [17]:
```
hist(data['Lat'], bins=100, range = (40.5, 41), color='#FF6347');
```

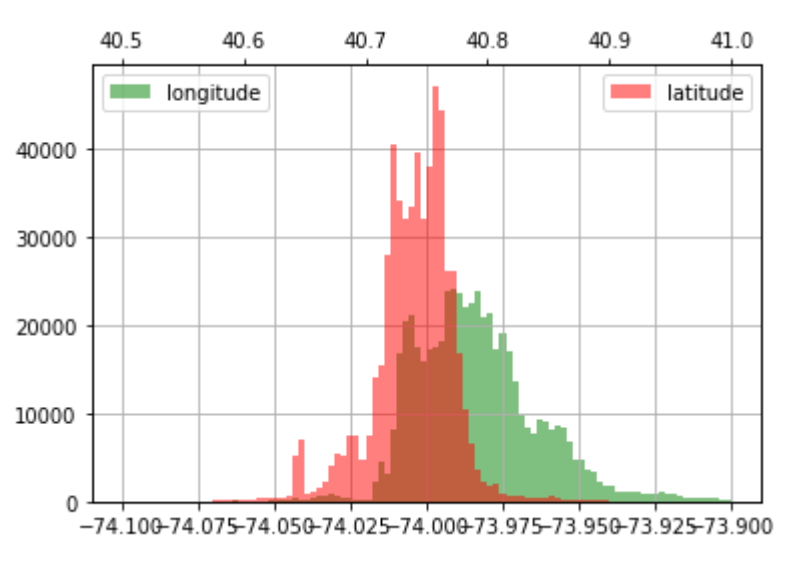


In [18]:
```
hist(data['Lon'], bins=100, range = (-74.1, -73.9), color='#FF6347');
```



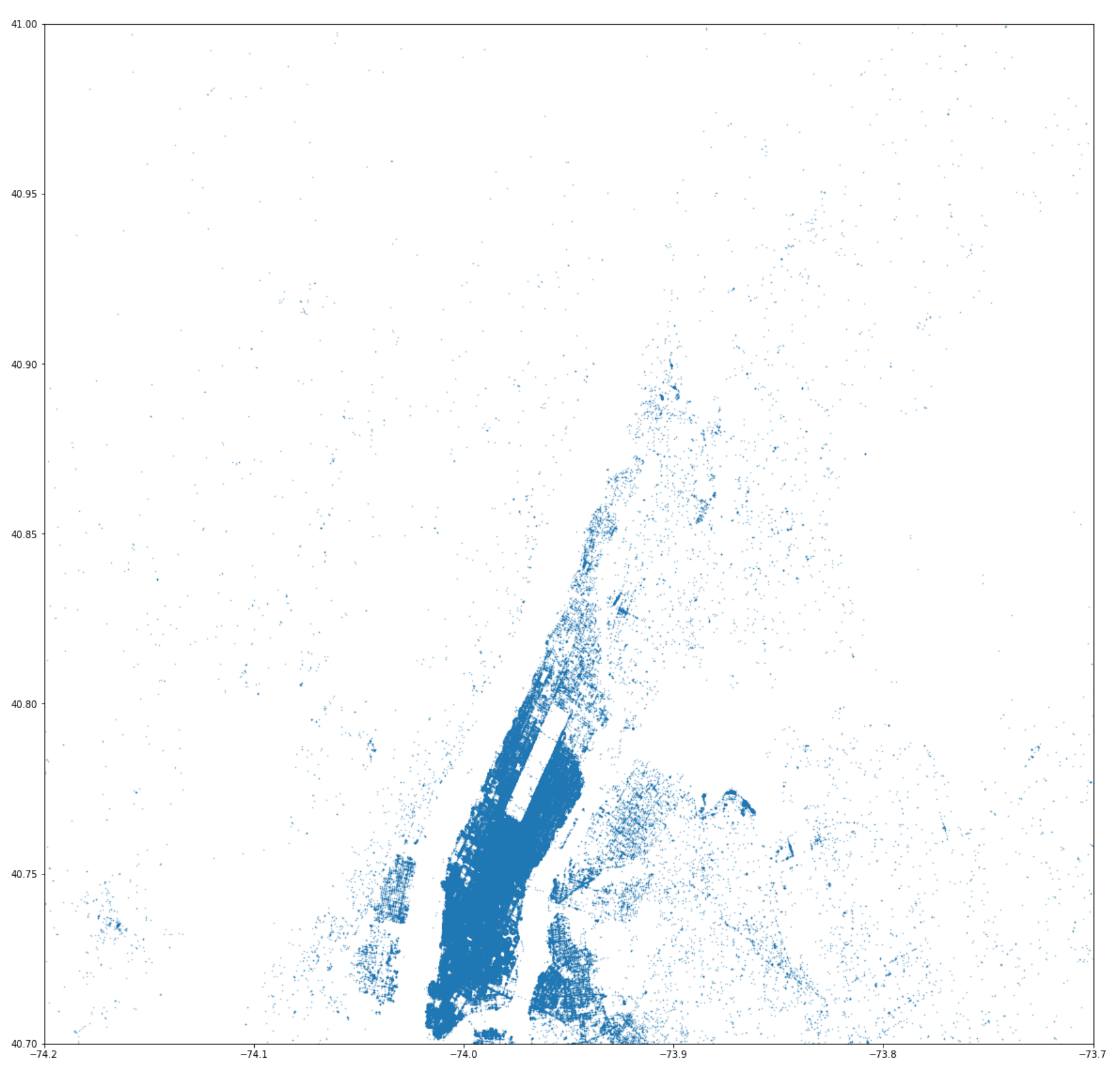
In [19]:
```
hist(data['Lon'], bins=100, range = (-74.1, -73.9), color='g', alpha=.5, label = 'longitude')
grid()
legend(loc='upper left')
twiny()
hist(data['Lat'], bins=100, range = (40.5, 41), color='r', alpha=.5, label = 'latitude')
legend(loc='best')
;
```

Out[19]:
```
''
```



In [20]:
```
figure(figsize=(20, 20))
plot(data['Lon'], data['Lat'], '.', ms=1, alpha=.5)
xlim(-74.2, -73.7)
ylim(40.7, 41)
```

Out[20]:
```
(40.7, 41)
```



In [ ]: