In [85]:
```
%pylab inline
import pandas
import seaborn
```

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

Loading Dataset Into Memory

In [3]:
```
data = pandas.read_csv('Desktop/Anushka/Uber_data_analytics_Python/Uber-dataset.csv')
```

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

Out[7]:

	Date/Time	Lat	Lon	Base
564511	4/30/2014 23:22:00	40.7640	-73.9744	B02764
564512	4/30/2014 23:26:00	40.7629	-73.9672	B02764
564513	4/30/2014 23:31:00	40.7443	-73.9889	B02764
564514	4/30/2014 23:32:00	40.6756	-73.9405	B02764
564515	4/30/2014 23:48:00	40.6880	-73.9608	B02764

Data Preparation

Converting datetime and adding some useful columns

In [17]:
```
data['Date/Time']= data['Date/Time'].map(pandas.to_datetime)
```

In [18]:
```
data.tail()
```

Out[18]:

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

In [19]:
```
def get_dom(dt): #creating seperate column for day of the month i.e. DOM
    return dt.day

data['dom']=data['Date/Time'].map(get_dom) #getting the day of the month
```

In [20]:
```
data.tail()
```

Out[20]:

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

In [27]:
```
def get_weekday(dt): #creating seperate column for weekday
    return dt.weekday()

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

def get_hour(dt):  #creating seperate column for hour
    return dt.hour

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

data.tail()
```

Out[27]:

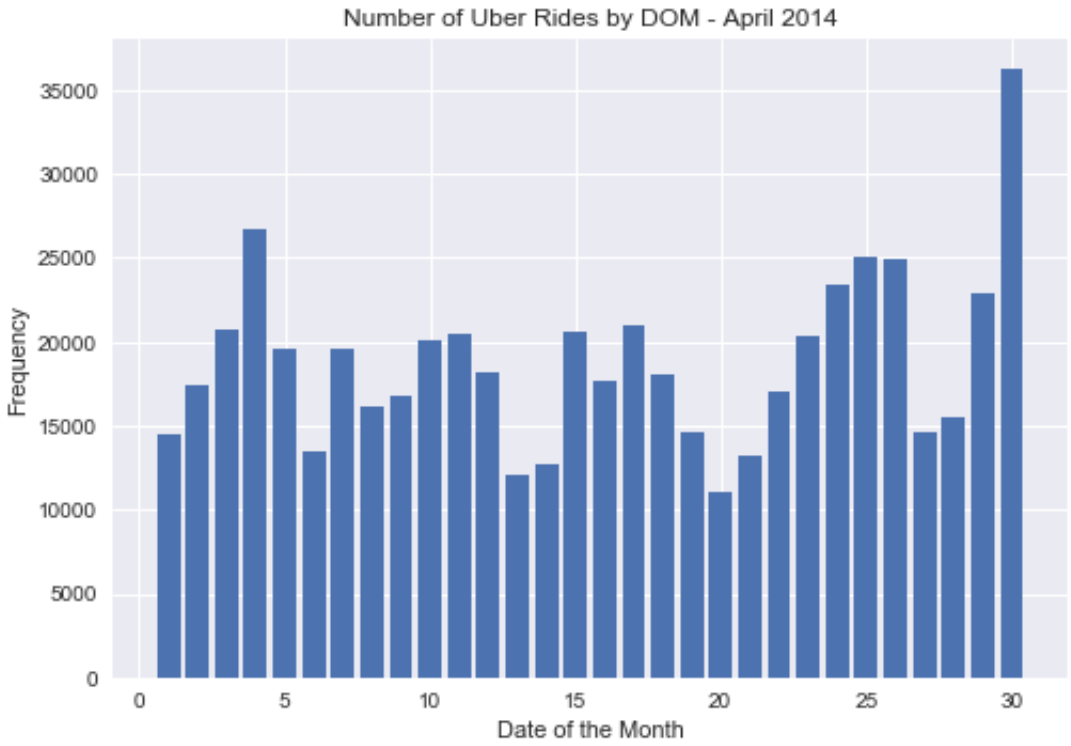
	Date/Time	Lat	Lon	Base	dom	weekday	hour
564511	2014-04-30 23:22:00	40.7640	-73.9744	B02764	30	2	23
564512	2014-04-30 23:26:00	40.7629	-73.9672	B02764	30	2	23
564513	2014-04-30 23:31:00	40.7443	-73.9889	B02764	30	2	23
564514	2014-04-30 23:32:00	40.6756	-73.9405	B02764	30	2	23
564515	2014-04-30 23:48:00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

Data Analysis

Analysing the Day of the Month Data (Histogram)

In [33]:
```
hist(data.dom, bins=30, rwidth=.8, range=(0.5, 30.5))
xlabel('Date of the Month')
ylabel('Frequency')
title('Number of Uber Rides by DOM - April 2014')
```

Out[33]: <matplotlib.text.Text at 0x11dcbc390>

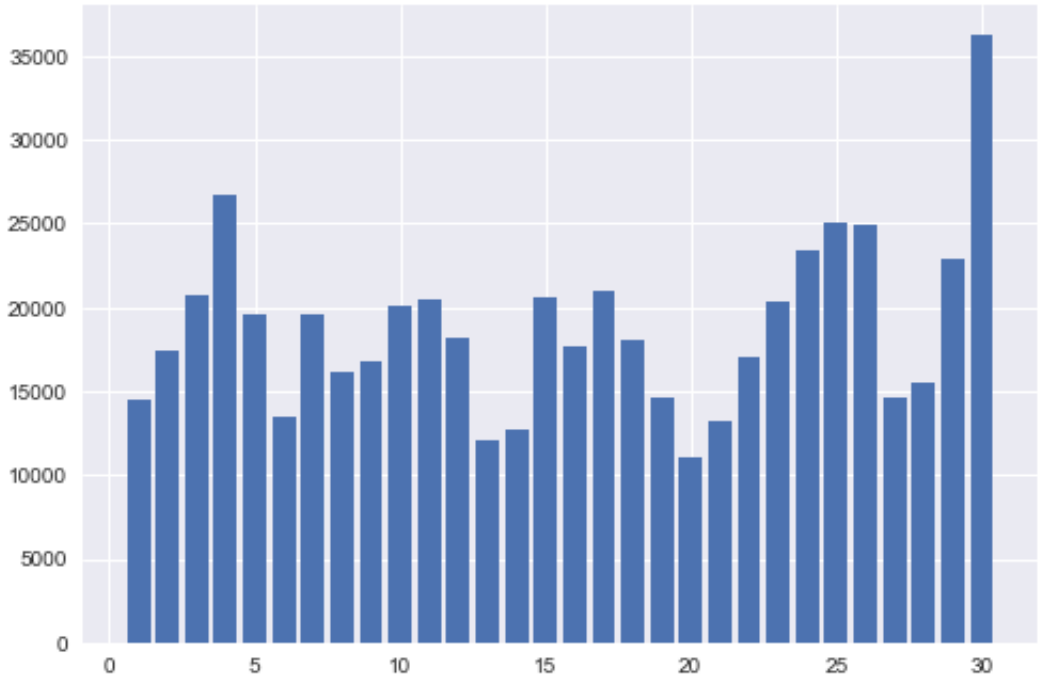


In [35]:
```
#for k, rows in data.groupby('dom'):
 #   print((k, len(rows)))

def  count_rows(rows):
    return len(rows)

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

```
Out[35]: 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 [40]:
```
bar(range(1,31),(by_date))
```

Out[40]: <Container object of 30 artists>

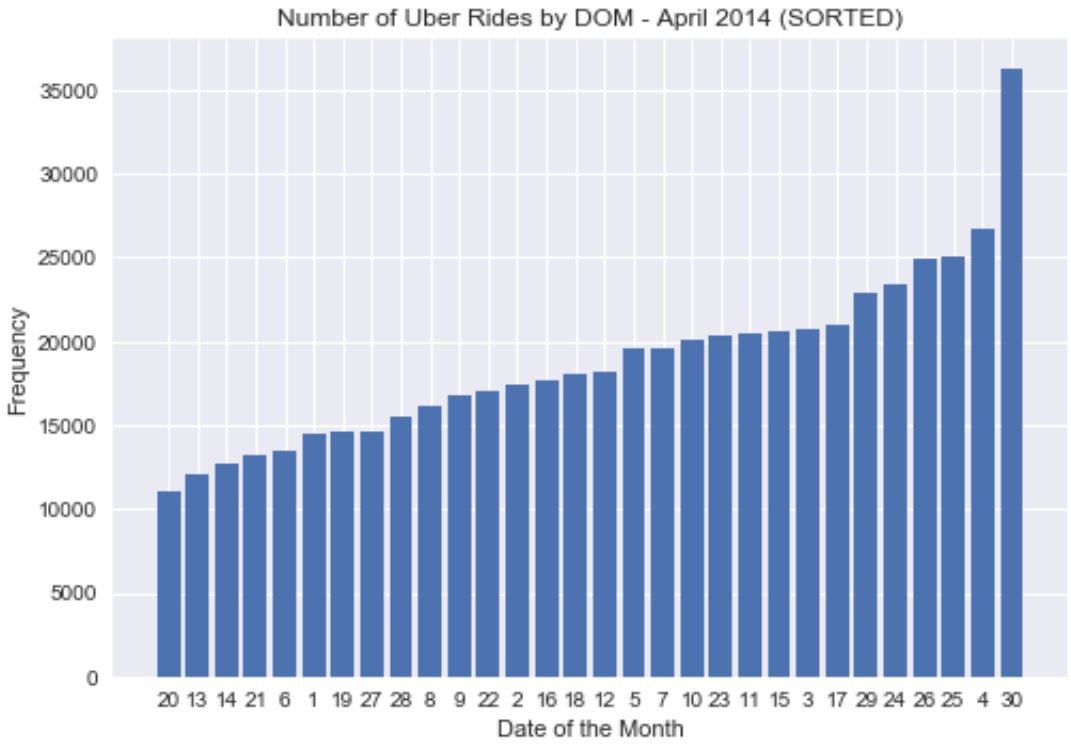


In [42]:
```
by_date_sorted = by_date.sort_values()
by_date_sorted
```

Out[42]:
```
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 [45]:
```
bar(range(1, 31), by_date_sorted)
xticks(range(1,31),by_date_sorted.index)
xlabel('Date of the Month')
ylabel('Frequency')
title('Number of Uber Rides by DOM - April 2014 (SORTED)')
;
```

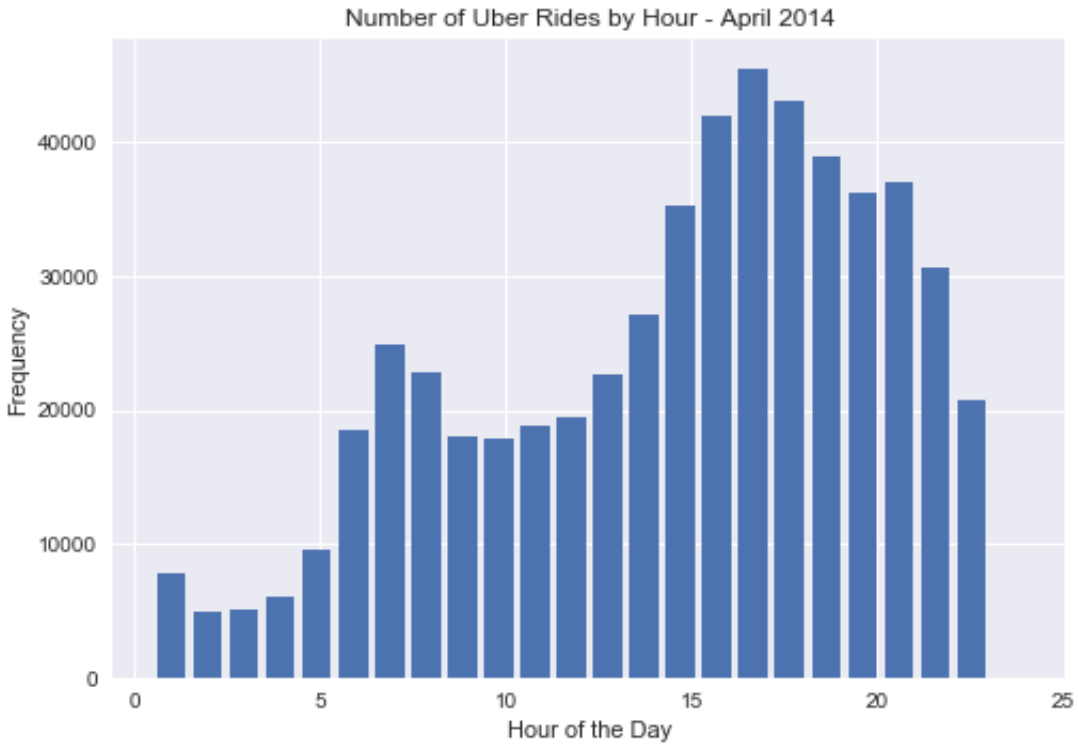
Out[45]: ''



Analyzing by Hour (Histogram)

In [48]:
```
hist(data.hour, bins=24, rwidth=.8, range=(.5, 24))
xlabel('Hour of the Day')
ylabel('Frequency')
title('Number of Uber Rides by Hour - April 2014')
```

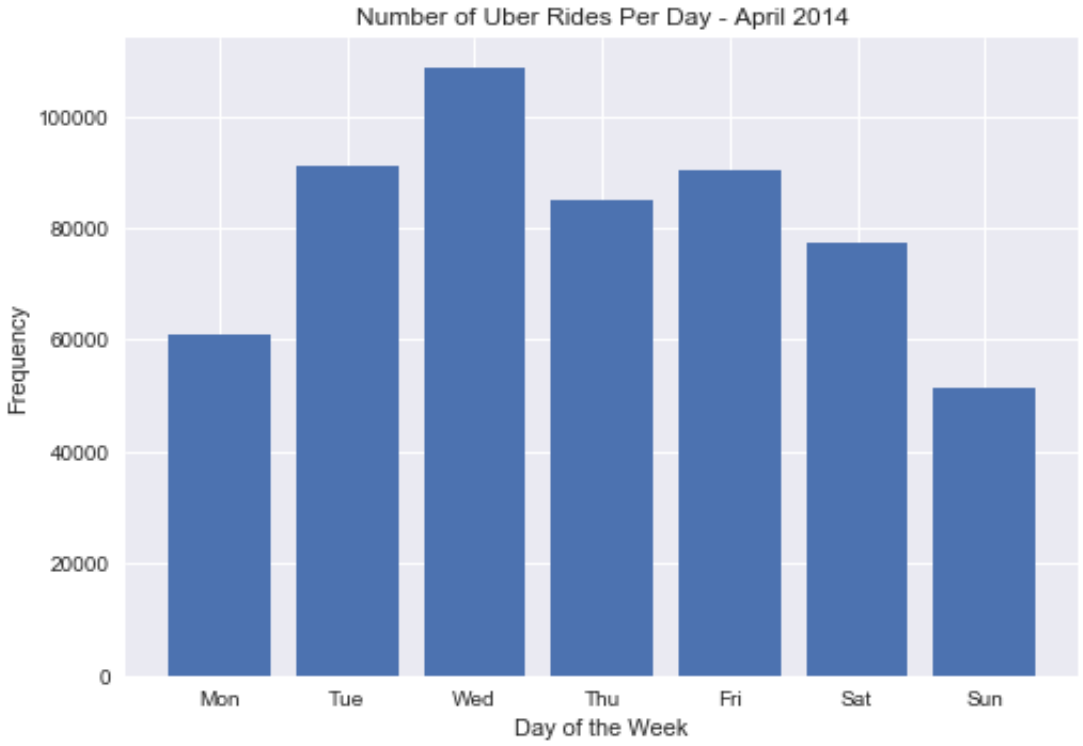
Out[48]: <matplotlib.text.Text at 0x11e1ee828>



Analyzing by Weekday (Histogram)

In [59]:
```
hist(data.weekday, bins=7, range=(-.5,6.5), rwidth=.8)
xticks(range(7), 'Mon Tue Wed Thu Fri Sat Sun'.split())
xlabel('Day of the Week')
ylabel('Frequency')
title('Number of Uber Rides Per Day - April 2014')
```

Out[59]: <matplotlib.text.Text at 0x11fd4eac8>

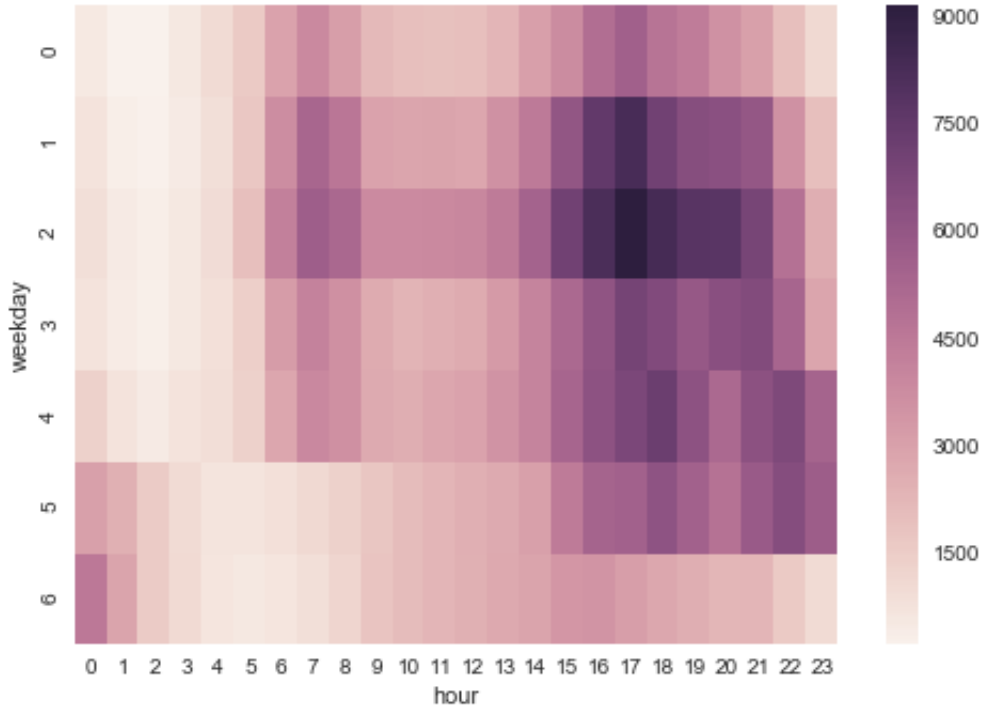


Analysis of Hour and DOW (CROSS ANALYSIS)

In [64]:
```
by_cross=data.groupby('weekday hour'.split()).apply(count_rows).unstac
k()
```

In [65]:
```
seaborn.heatmap(by_cross)
```

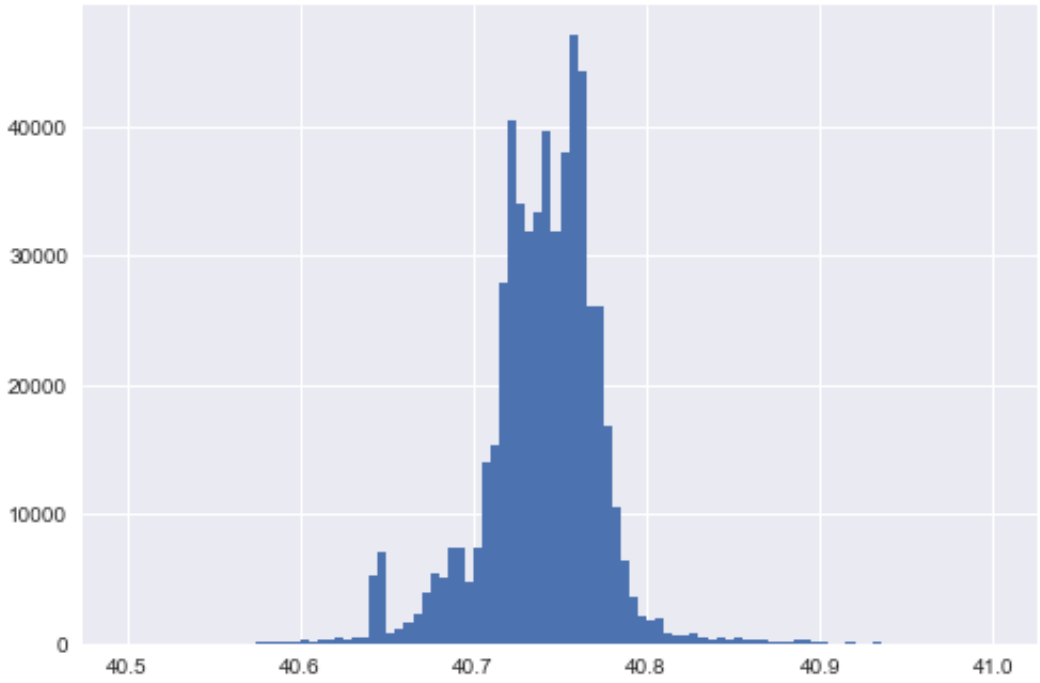
Out[65]: <matplotlib.axes._subplots.AxesSubplot at 0x11fd89f98>



Analysis by Latitude and Longitude

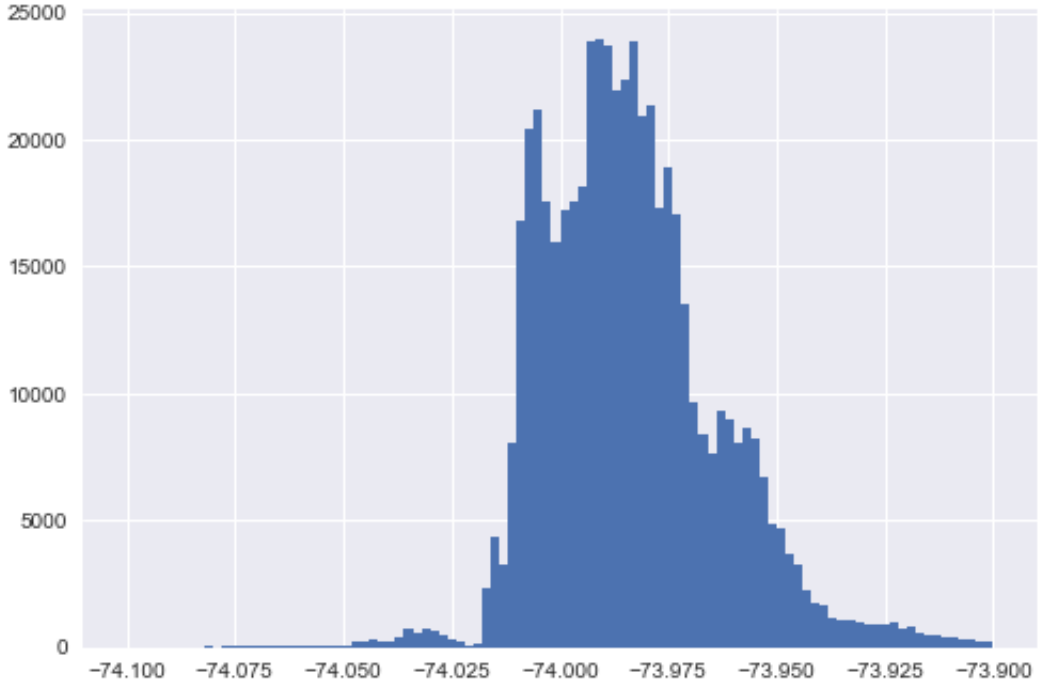
In [66]:
```
hist(data['Lat'],bins=100, range = (40.5,41))
;
```

Out[66]: ''



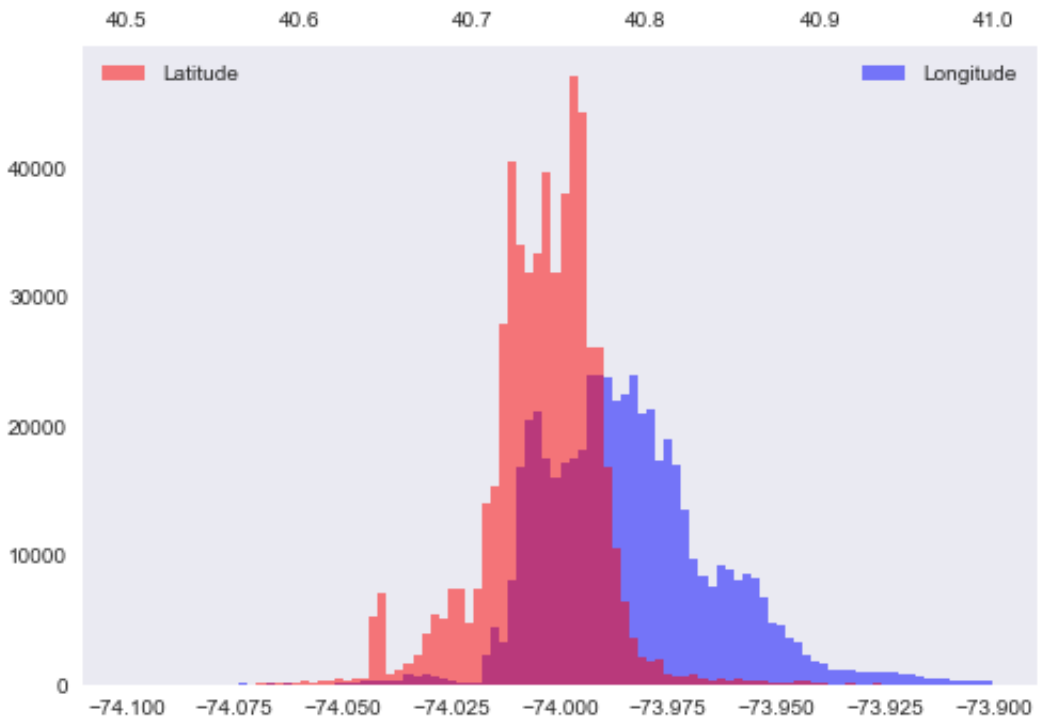
In [68]:
```
hist(data['Lon'],bins=100, range = (-74.1,-73.9))
;
```

Out[68]: ''



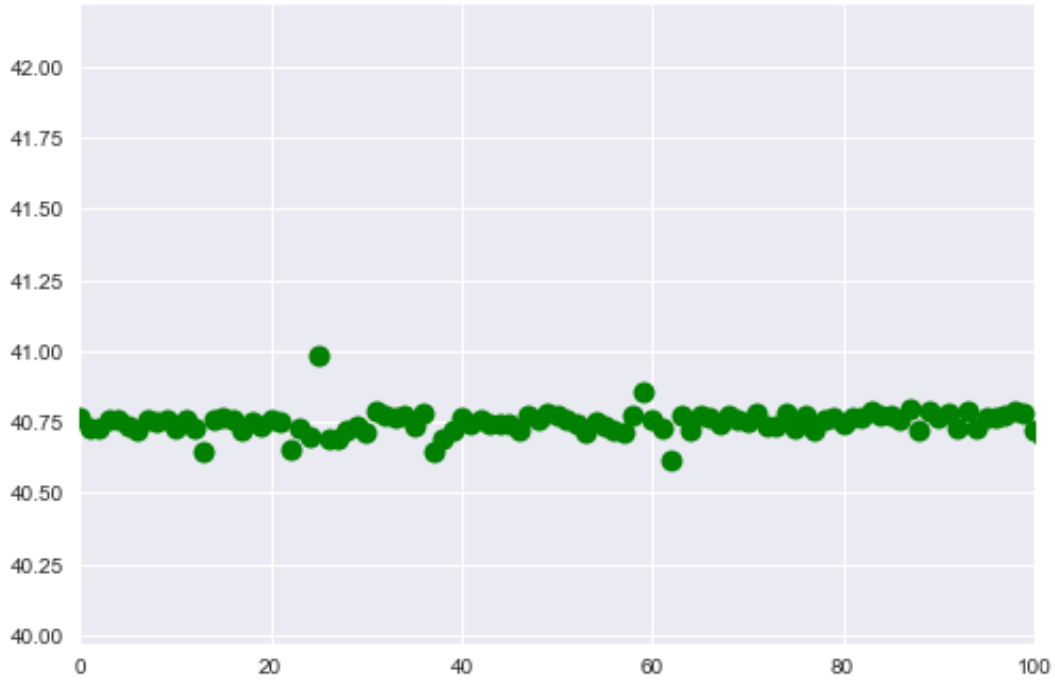
In [79]:
```
hist(data['Lon'],bins=100, range = (-74.1,-73.9), color='b', alpha=.5,
label='Longitude')
grid()
legend(loc='best')
twiny()
hist(data['Lat'],bins=100, range = (40.5,41), color='r', alpha=.5, lab
el='Latitude')
grid()
legend(loc='upper left')
;
```

Out[79]: ''



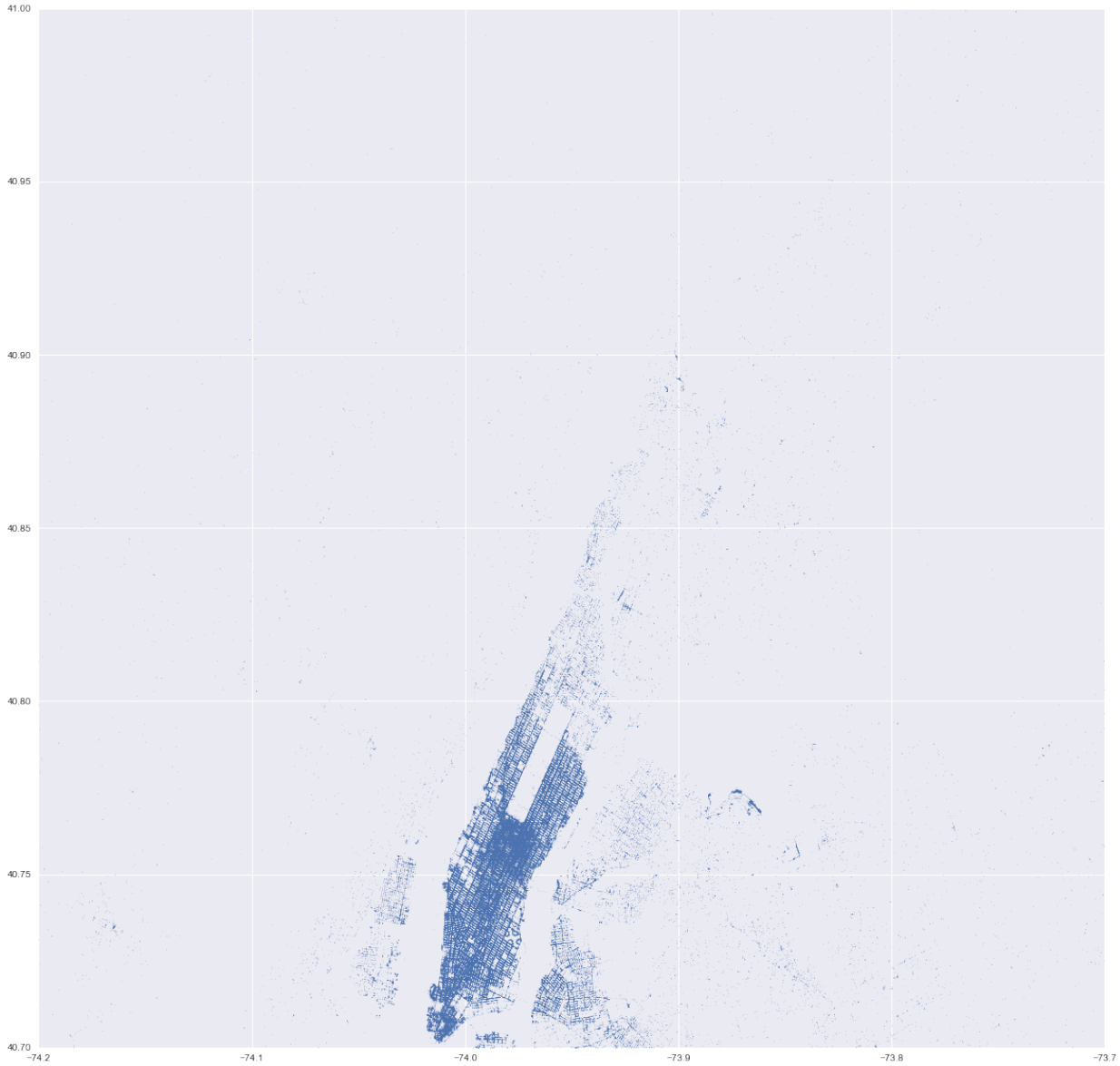
In [80]:
```
plot(data['Lat'], '.', ms=20, color='green')
xlim(0,100)
```

Out[80]: (0, 100)



In [84]:
```
figure(figsize=(20,20))
plot(data['Lon'], data['Lat'],'.', ms=1, alpha=.5)
xlim(-74.2, -73.7)
ylim(40.7, 41)
```

Out[84]: (40.7, 41)



In [ ]: