In [1]:

```
%pylab inline
import pandas
import seaborn
```

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

This is my first attempt at using Jupyter Notebook, Im just practicing with relativley simple data sets to master the basics. I used two data sets, one from September's Uber Data and one from April's Uber Data. I chose those two as there would clearly be differences in frequency of rides because of the change in temperature, allowing for a simple analysis for my first time!

In [2]:

```
datasept = pandas.read_csv('uber-raw-data-sep14.csv')
dataapr = pandas.read_csv('uber-raw-data-apr14.csv')
```

In [3]:

```
datasept['Date/Time']= datasept['Date/Time'].map(pandas.to_datet
ime)
```

In [4]:

```
dataapr['Date/Time']= dataapr['Date/Time'].map(pandas.to_datetim
e)
```

Here I converted the Date/Time into a more manageable format. I also created a number of graphs to give a general visualization of the key points of the data.

In [5]:

```
datasept.head()
```

Out[5]:

	Date/Time	Lat	Lon	Base
0	2014-09-01 00:01:00	40.2201	-74.0021	B02512
1	2014-09-01 00:01:00	40.7500	-74.0027	B02512
2	2014-09-01 00:03:00	40.7559	-73.9864	B02512
3	2014-09-01 00:06:00	40.7450	-73.9889	B02512
4	2014-09-01 00:11:00	40.8145	-73.9444	B02512

In [6]:

```
dataapr.head()
```

Out[6]:

	Date/Time	Lat	Lon	Base
0	2014-04-01 00:11:00	40.7690	-73.9549	B02512
1	2014-04-01 00:17:00	40.7267	-74.0345	B02512
2	2014-04-01 00:21:00	40.7316	-73.9873	B02512
3	2014-04-01 00:28:00	40.7588	-73.9776	B02512
4	2014-04-01 00:33:00	40.7594	-73.9722	B02512

In [29]:

```
def get_dom(dt):
    return dt.day
datasept['dom'] = datasept['Date/Time'].map(get_dom)
dataapr['dom'] = dataapr['Date/Time'].map(get_dom)
```

In [30]:

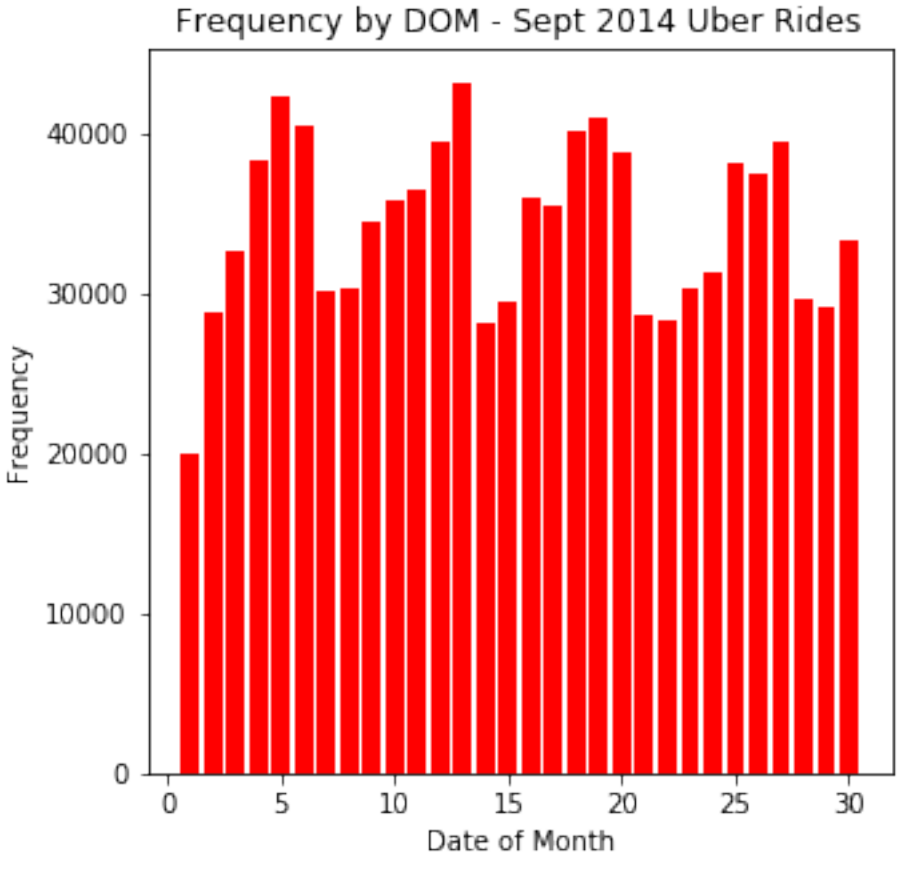
```
def get_weekday(dt):
    return dt.weekday()
datasept['weekday'] = datasept['Date/Time'].map(get_weekday)
dataapr['weekday'] = dataapr['Date/Time'].map(get_weekday)
```

In [31]:

```
def get_hour(dt):
    return dt.hour
datasept['hour'] = datasept['Date/Time'].map(get_hour)
dataapr['hour'] = dataapr['Date/Time'].map(get_hour)
```

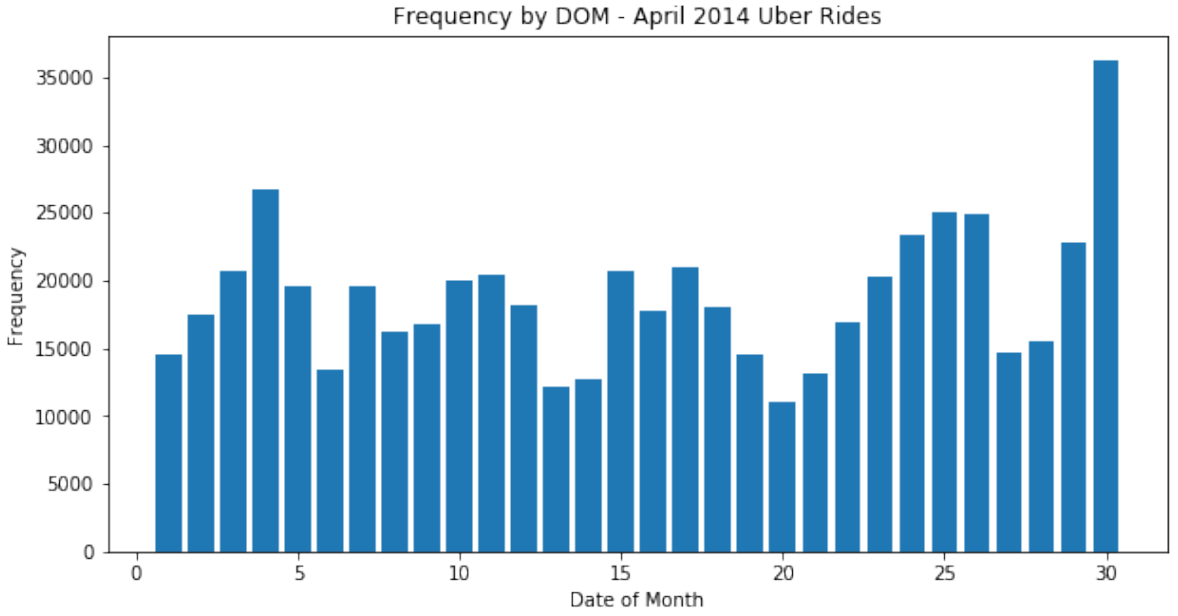
In [44]:

```
hist(datasept.dom, bins=30, rwidth=.8, range=(.5,30.5),color = '
red')
xlabel('Date of Month')
ylabel('Frequency')
title('Frequency by DOM - Sept 2014 Uber Rides')
plt.rcParams["figure.figsize"] = (5,5)
```



In [43]:

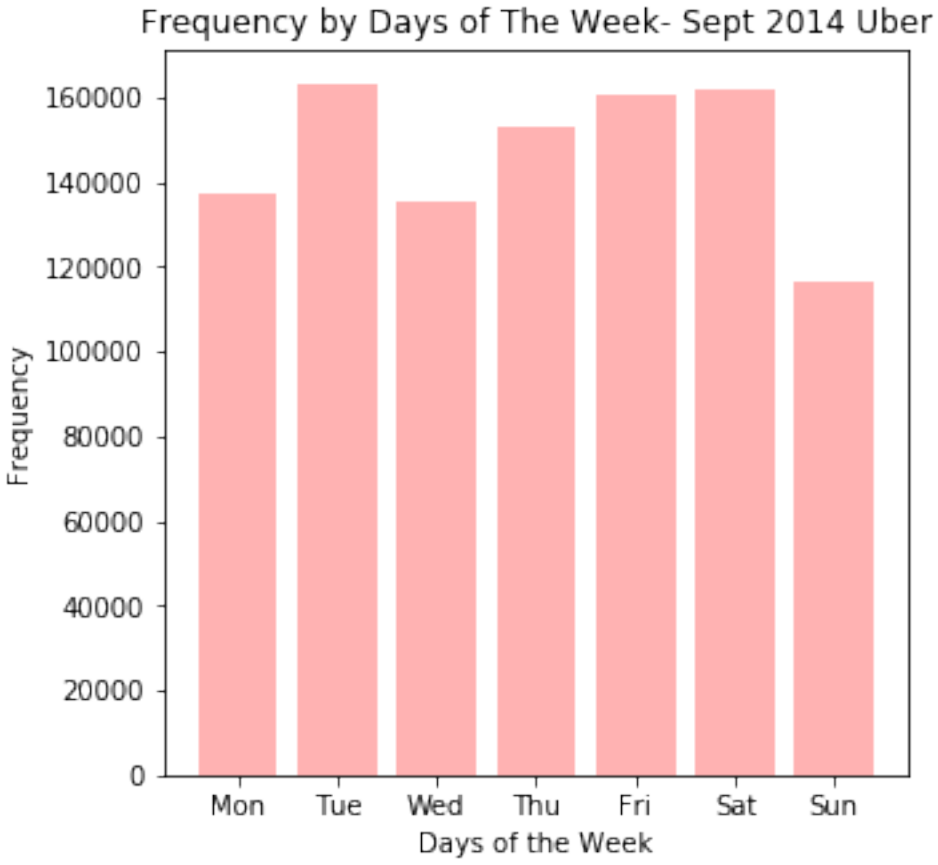
```
hist(dataapr.dom, bins=30, rwidth=.8, range=(.5,30.5))
xlabel('Date of Month')
ylabel('Frequency')
title('Frequency by DOM - April 2014 Uber Rides')

plt.rcParams["figure.figsize"] = (5,5)
```



GRAPHING SECTION

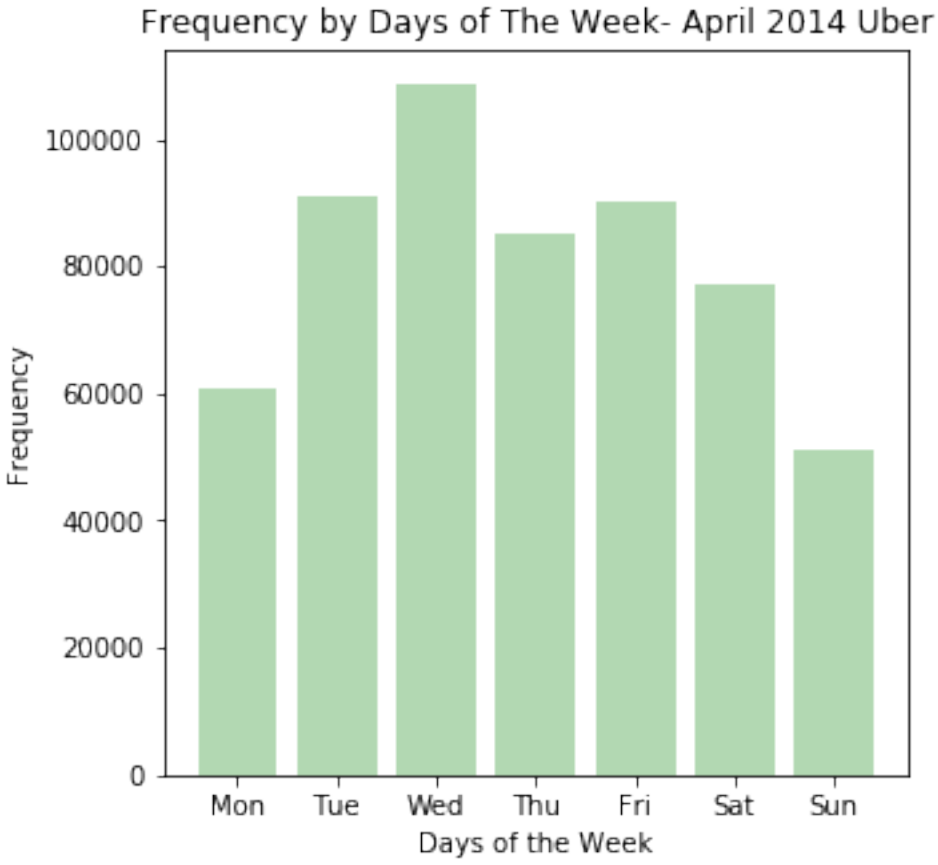
In [45]:

```
hist(datasept.weekday, bins=7, range=(-.5,6.5), rwidth=.8, color
='Red', alpha=.3)
xticks(range(7), 'Mon Tue Wed Thu Fri Sat Sun'.split())
xlabel('Days of the Week')
ylabel('Frequency')
title('Frequency by Days of The Week- Sept 2014 Uber')
plt.rcParams["figure.figsize"] = (5,5)
```



In [46]:

```
hist(dataapr.weekday, bins=7, range=(-.5,6.5), rwidth=.8, color=
'green', alpha=.3)
xticks(range(7), 'Mon Tue Wed Thu Fri Sat Sun'.split())
xlabel('Days of the Week')
ylabel('Frequency')
title('Frequency by Days of The Week- April 2014 Uber')
plt.rcParams["figure.figsize"] = (5,5)
```



Cross ANALYSIS SECTION

In [36]:

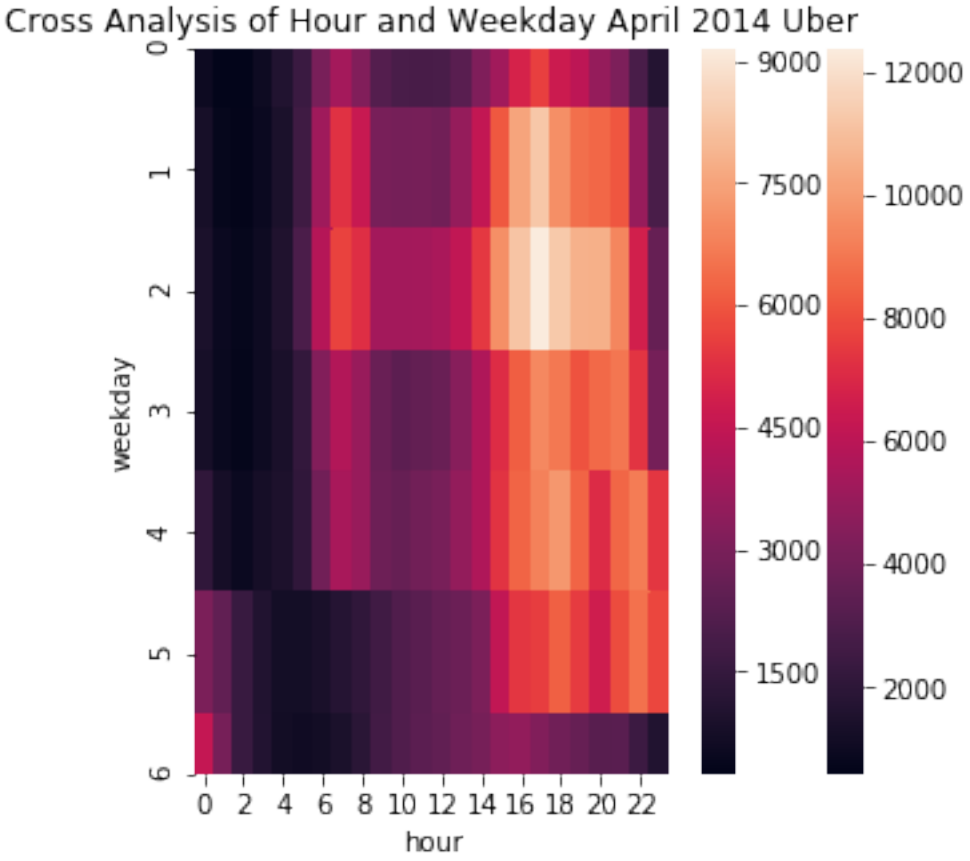
```
def count_rows(rows):
    return len(rows)
by_cross_hw_sept = datasept.groupby('weekday hour'.split()).appl
y(count_rows).unstack()
```

In [37]:

```
by_cross_hw_apr = dataapr.groupby('weekday hour'.split()).apply(
count_rows).unstack()
```

In [47]:

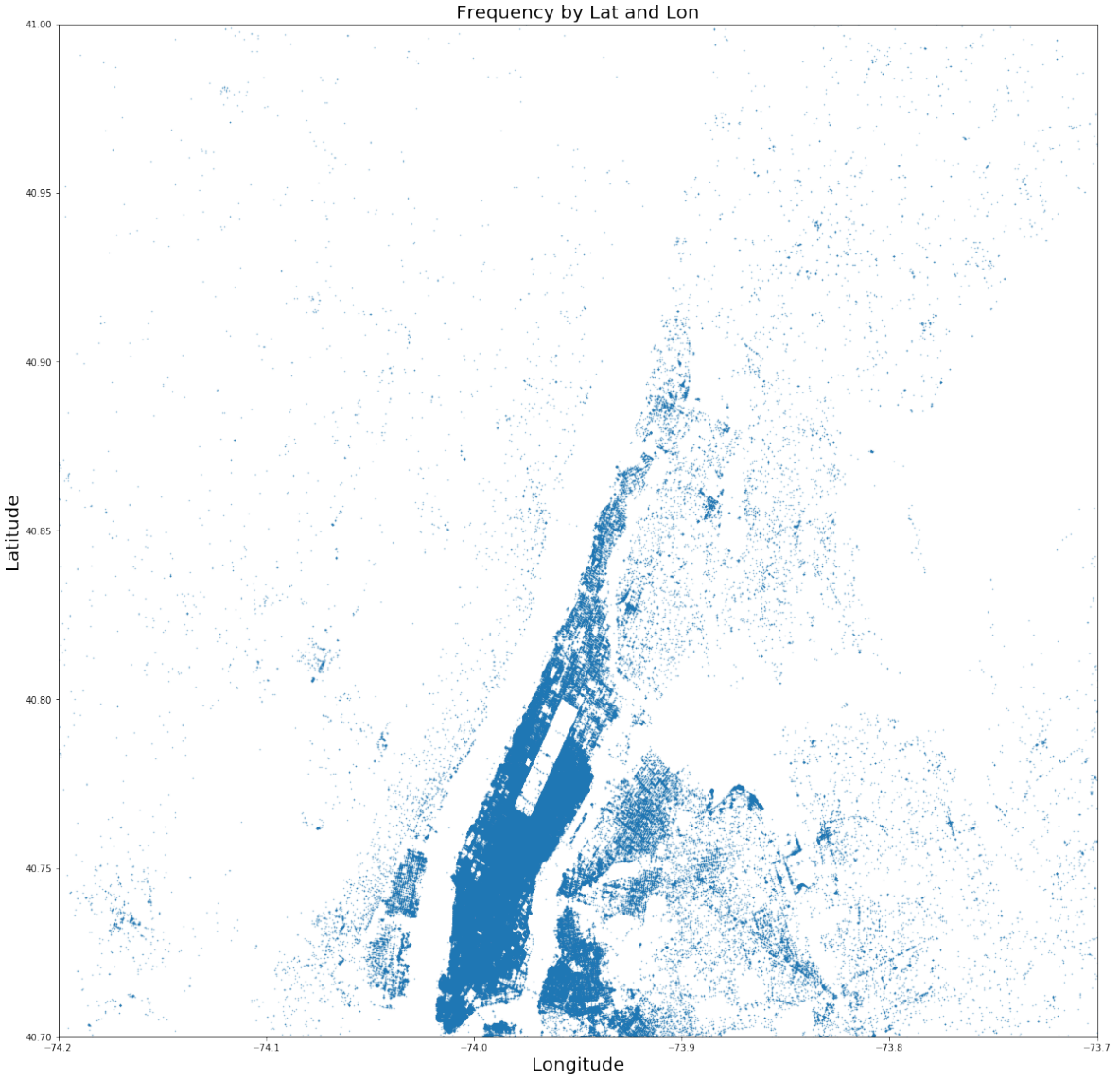
```
 seaborn.heatmap(by_cross_hw_sept)
 title('Cross Analysis of Hour and Weekday Sept 2014 Uber')

 seaborn.heatmap(by_cross_hw_apr)
 title('Cross Analysis of Hour and Weekday April 2014 Uber')
plt.rcParams["figure.figsize"] = (5,5)
```



Using seaborns heatmap was not a good choice as it didnt indicate anything useful and is hard to understand in this context. I used it simply to get an understadning of seaborns builtin functions

In [48]:

```
figure(figsize(20,20))
plot(datasept['Lon'],datasept['Lat'], '.', ms=1, alpha=.5)
xlim(-74.2, -73.7)
ylim(40.70,41)
xlabel('Longitude', size=20)
ylabel('Latitude',size=20)
title('Frequency by Lat and Lon',size=20)
```

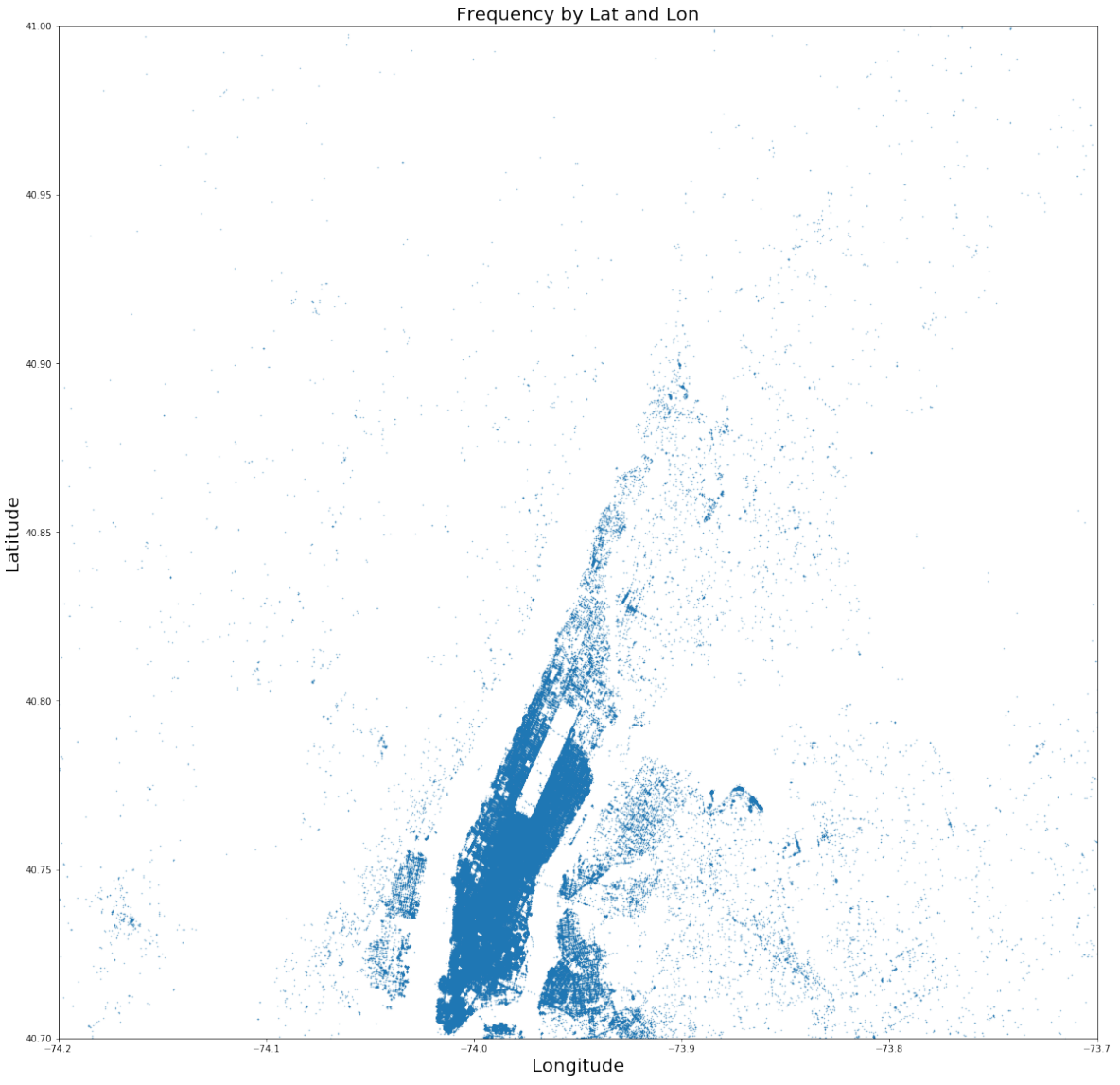


In [52]:

```
figure(figsize(20,20))
plot(dataapr['Lon'],dataapr['Lat'], '.', ms=1, alpha=.5)
xlim(-74.2, -73.7)
ylim(40.70,41)
xlabel('Longitude', size=20)
ylabel('Latitude',size=20)
title('Frequency by Lat and Lon',size=20)
```

Out[52]:

```
Text(0.5, 1.0, 'Frequency by Lat and Lon')
```



Overall from this analysis I was able to solidify assumptions I already had regarding frequency of rides and the increase in radius and frequency of rides that is correlated with harsher weather. Thanks for taking a look at my first time! I hope to get better soon!