In [1]:

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

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

# Load CSV file into memory

In [2]:

```
data=pandas.read_csv('Desktop/uber-raw-data-apr14.csv')
data.head()
```

Out[2]:

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

# Convert DateTime and add some useful columns

In [ ]:

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

In [5]:

```
data.tail()
```

Out[5]:

| | 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 [6]:

```
def get_dom(dt):
    return dt.day

data['dom'] = data['Date/Time'].map(get_dom)

data.tail()
```

Out[6]:

| | 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 [8]:

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

Out[8]:

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

In [9]:

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

Out[9]:

| | 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 |

# Let's start the analysis

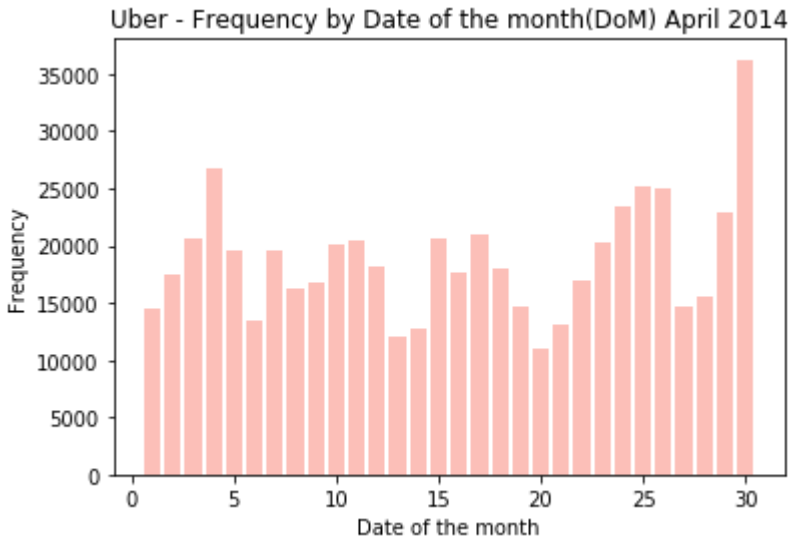
## Analyze the Date of Month(DoM)

In [15]:

```
hist(data.dom, bins=30, rwidth=.8 ,range=(0.5,30.5), color=('salmon'), alpha=(0.5))
xlabel('Date of the month')
ylabel('Frequency')
title('Uber - Frequency by Date of the month(DoM) April 2014')
```

Out[15]:

```
Text(0.5, 1.0, 'Uber - Frequency by Date of the month(DoM) April 201
4')
```



In [16]:

```
def count_rows(rows):
    return len(rows)

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

Out[16]:

```
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 [20]:

```
by_date_sorted = by_date.sort_values()
by_date_sorted
```

Out[20]:

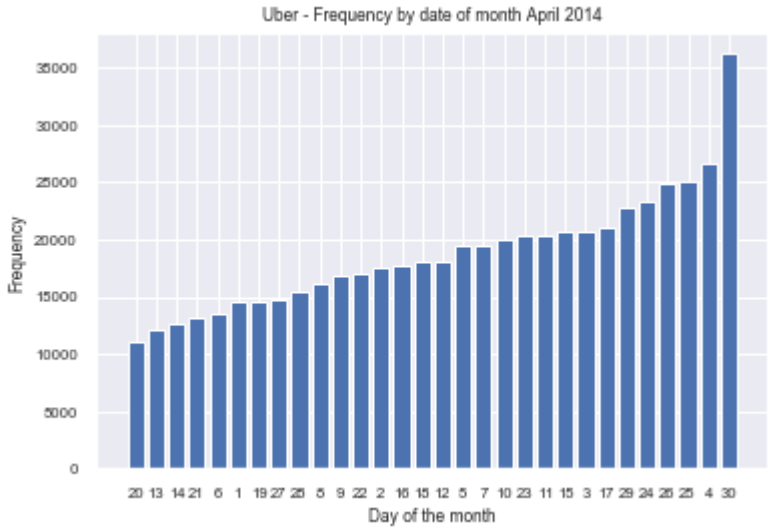
```
dom
20    11017
13    12112
14    12674
21    13162
6     13445
1     14546
19    14602
27    14677
28    15475
8     16188
9     16843
22    16975
2     17474
16    17717
18    18074
12    18170
5     19521
7     19550
10    20041
23    20346
11    20420
15    20641
3     20701
17    20973
29  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

In [89]:

```
bar(range(1,31),by_date_sorted)
seaborn.set(font_scale=0.7)
xticks(range(1,31), by_date_sorted.index)
xlabel('Day of the month')
ylabel('Frequency')
title('Uber - Frequency by date of month April 2014')
;
```

Out[89]:

```
''
```



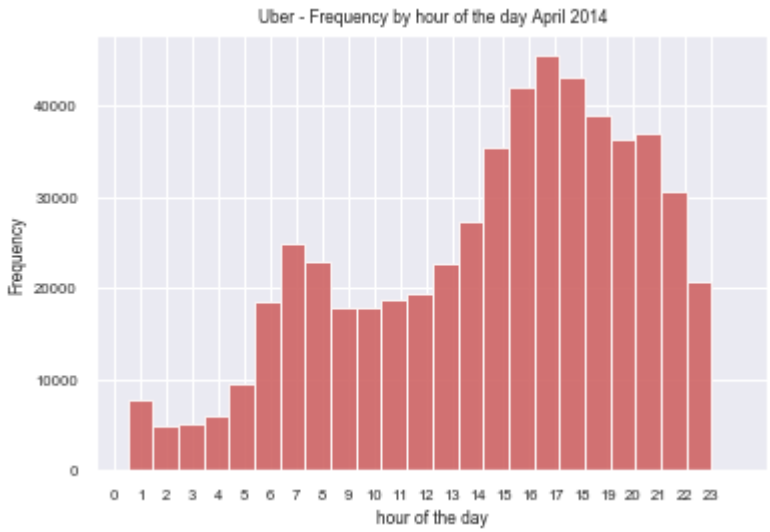

# Analyze the Hour

In [134]:

```
hist(data.hour,bins=24,range=(0.5,24),color=('indianred'), alpha=0.85)
xticks(range (0,24))
xlabel('hour of the day')
ylabel('Frequency')
title('Uber - Frequency by hour of the day April 2014')
;
```

Out[134]:

```
''
```



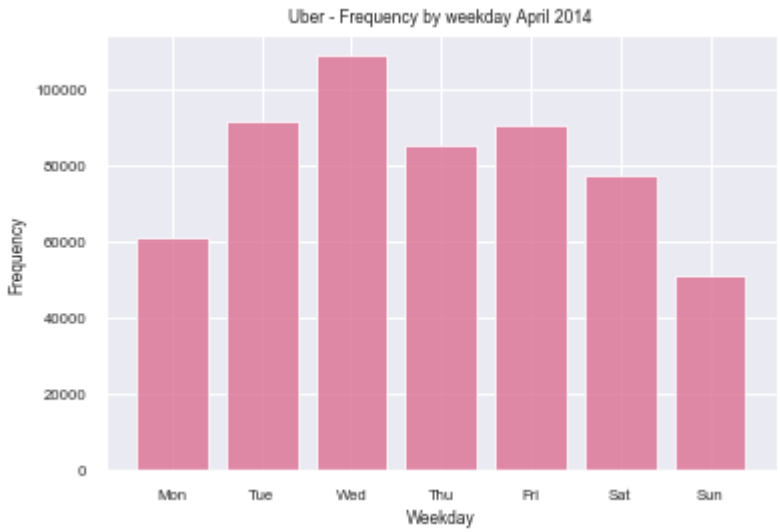

# Analyze the weekday

In [102]:

```
hist(data.weekday,bins=7, range=(-.5,6.5), rwidth=.8, color='palevioletred', alpha=(
xticks(range(7), 'Mon Tue Wed Thu Fri Sat Sun'.split())
xlabel('Weekday')
ylabel('Frequency')
title('Uber - Frequency by weekday April 2014')
;
```

Out[102]:

```
''
```




## Cross analysis (hour, day of the week)

In [104]:

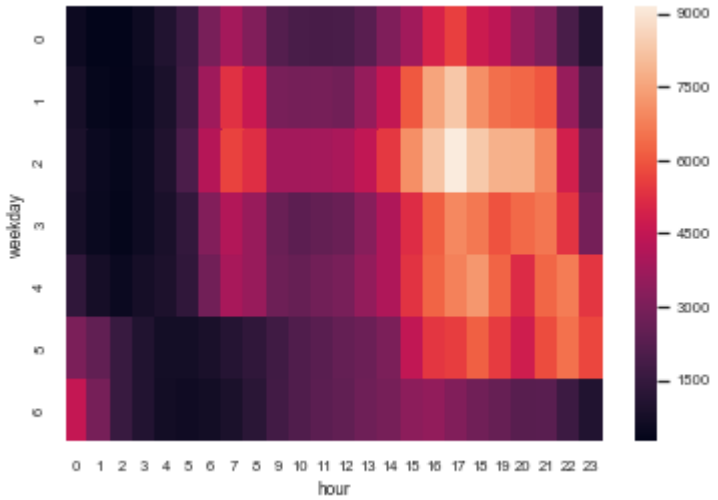
```
by_cross = data.groupby('weekday hour'.split()).apply(count_rows).unstack()
```

In [106]:

```
seaborn.heatmap(by_cross)
;
```

Out[106]:

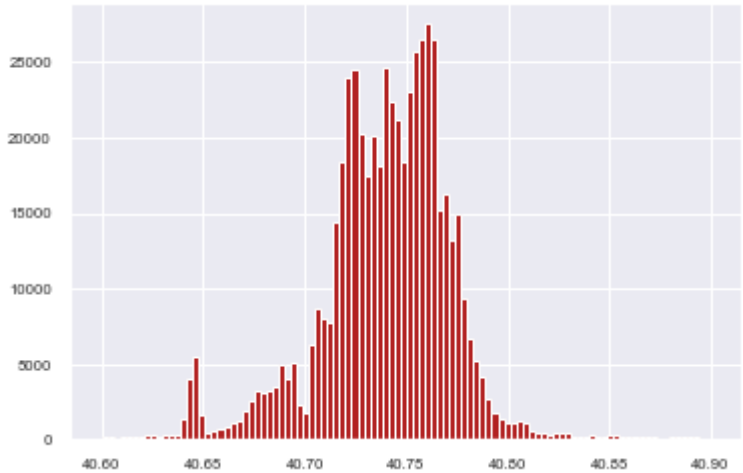
```
''
```




# Latitude and longitude analysis

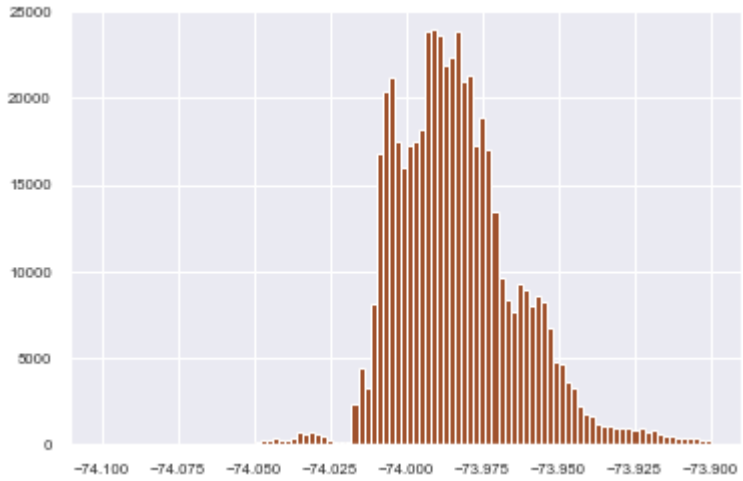
In [126]:

```
hist(data['Lat'], bins=100, range=(40.6,40.9), color='firebrick');
```



In [125]:

```
hist(data['Lon'], bins=100, range=(-74.1,-73.9), color='sienna');
```

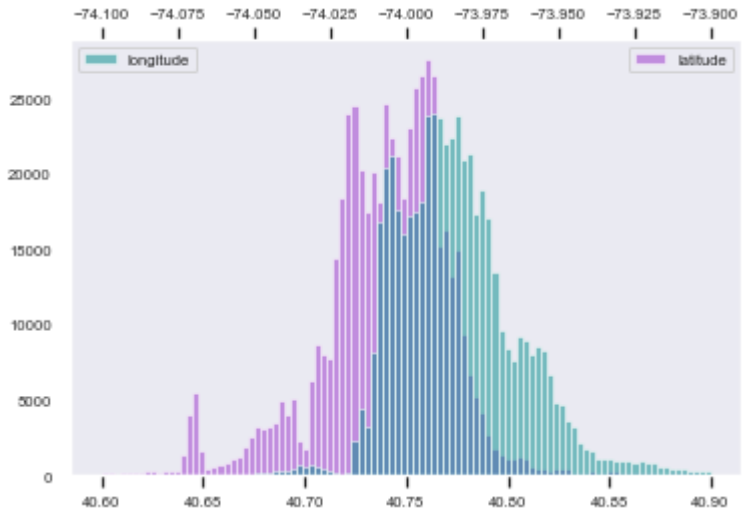



In [136]:

```
hist(data['Lat'], bins=100, range=(40.6,40.9), color='darkorchid', alpha=0.5, label=
grid()
legend(loc='best')
twiny()
hist(data['Lon'], bins=100, range=(-74.1,-73.9), color='darkcyan', alpha=0.5, label=
grid()
legend(loc='upper left')
;
```

Out[136]:

```
''
```

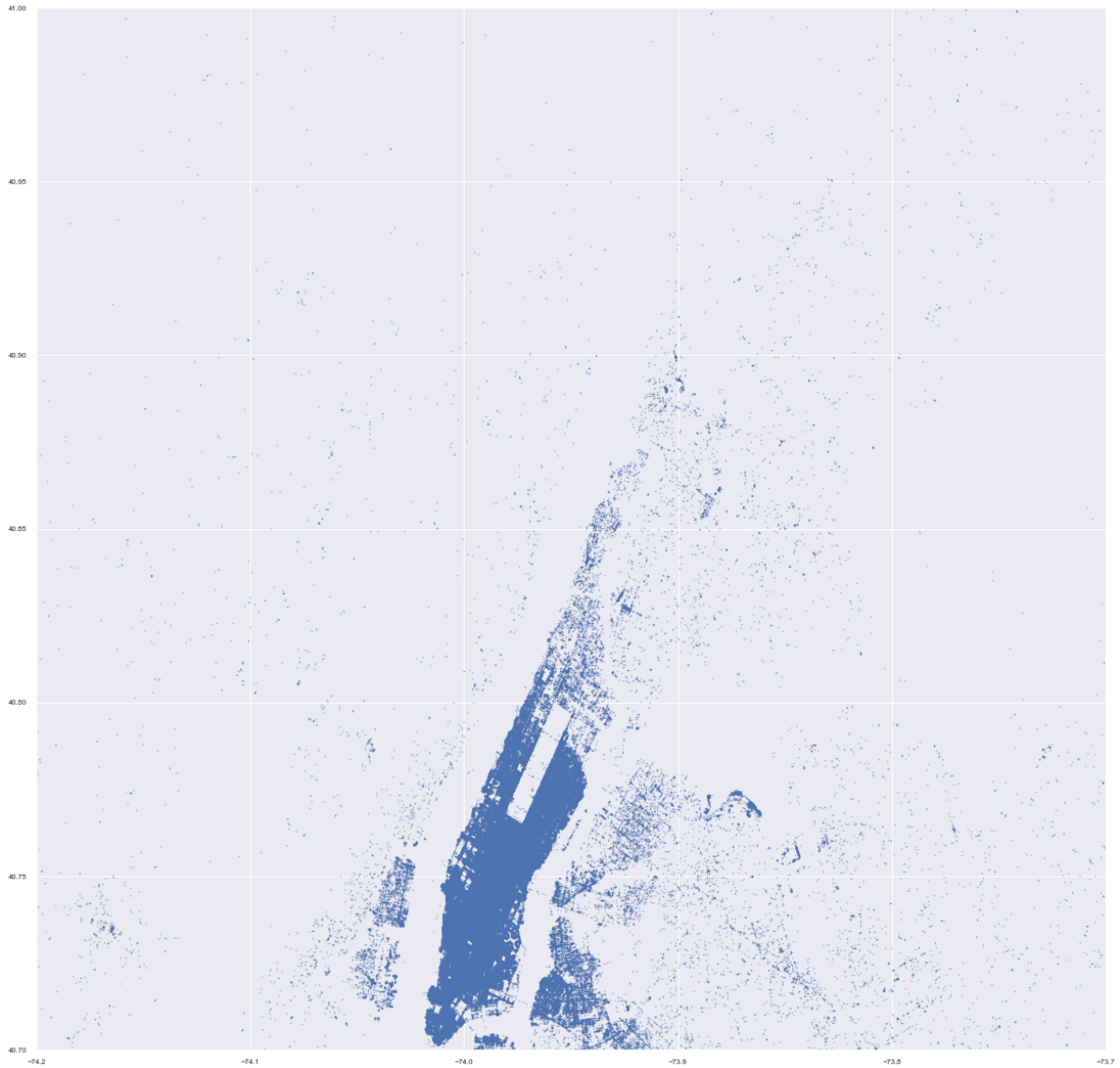


In [144]:

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

Out[144]:

```
(40.7, 41)
```




In [ ]: