Luis Santulli Uber Data Analysis

In [7]:
```
#Importing modules
%pylab inline
import pandas as luis
import seaborn as santulli
```

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

Loading CSV into memory from Uber Raw Data

In [8]:
```
data = luis.read_csv(r"C:\Users\Luis Santulli\Desktop\uber-raw-data-apr14.csv")
```

In [9]:
```
data.tail()
```

Out[9]:

	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

In [10]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 564516 entries, 0 to 564515
Data columns (total 4 columns):
Date/Time    564516 non-null object
Lat          564516 non-null float64
Lon          564516 non-null float64
Base         564516 non-null object
dtypes: float64(2), object(2)
memory usage: 17.2+ MB
```

Converting to Date/Time format

In [11]:
```
#Using the mapping function because it is a series

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

#This took 8 minutes to convert
```

In [12]:
```
#Displaying correct date/time format
data.tail()
```

Out[12]:

	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

Defining functions and adding useful columns

In [13]:
```
def get_dom(dt):
    return dt.day

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


def get_weekdate(dt):
    return dt.weekday()

data['Weekday'] = data['Date/Time'].map(get_weekdate)


def get_hour(dt):
    return dt.hour

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

In [14]:
```
data.tail()
```

Out[14]:

	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

In [15]:
```
data.head()
```

Out[15]:

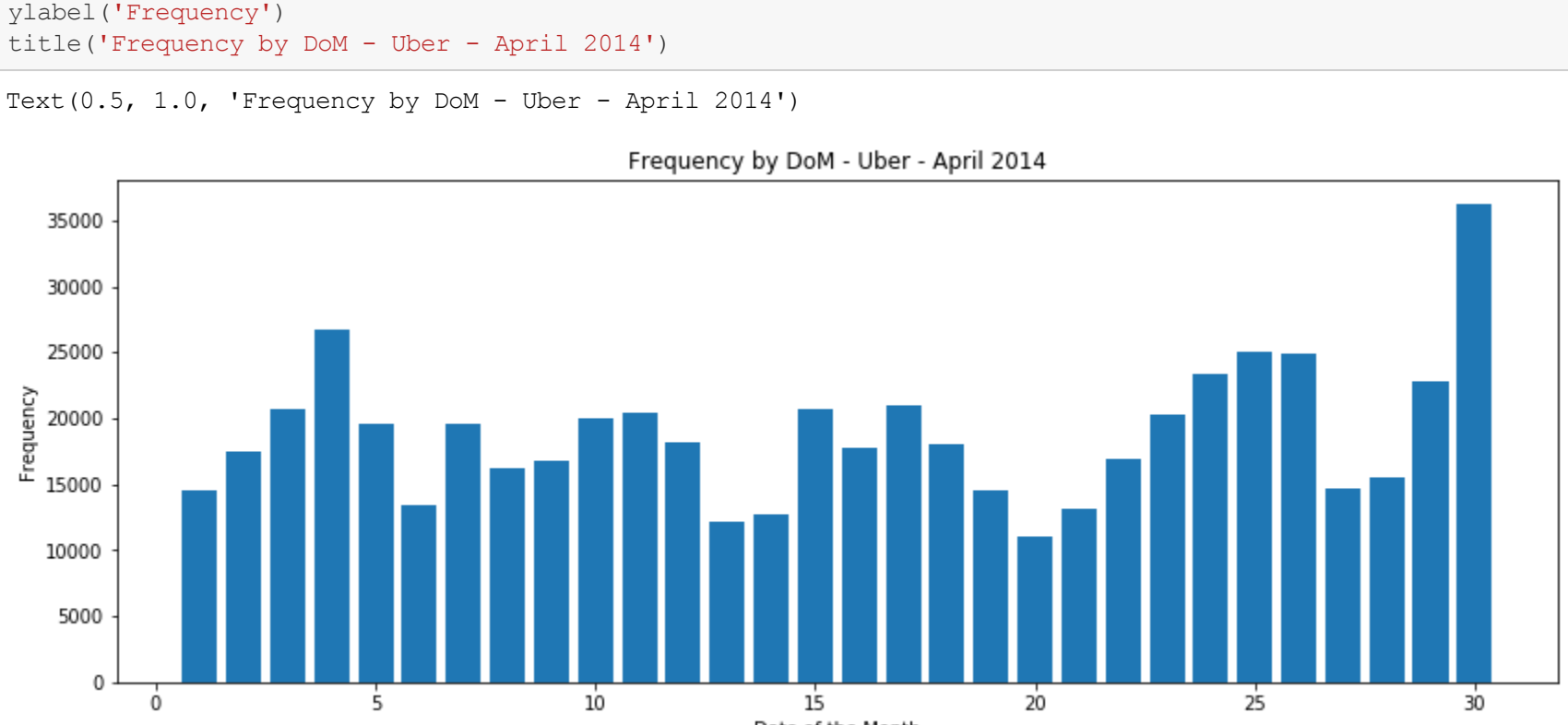
	Date/Time	Lat	Lon	Base	DoM	Weekday	Hour
0	2014-04-01 00:11:00	40.7690	-73.9549	B02512	1	1	0
1	2014-04-01 00:17:00	40.7267	-74.0345	B02512	1	1	0
2	2014-04-01 00:21:00	40.7316	-73.9873	B02512	1	1	0
3	2014-04-01 00:28:00	40.7588	-73.9776	B02512	1	1	0
4	2014-04-01 00:33:00	40.7594	-73.9722	B02512	1	1	0

Analysis

Analyzing DoM

In [16]:
```
figure(figsize=(14, 5))
hist(data.DoM, bins=30, rwidth=.8, range=(0.5, 30.5))
xlabel('Date of the Month')
ylabel('Frequency')
title('Frequency by DoM - Uber - April 2014')
```

Out[16]: `Text(0.5, 1.0, 'Frequency by DoM - Uber - April 2014')`

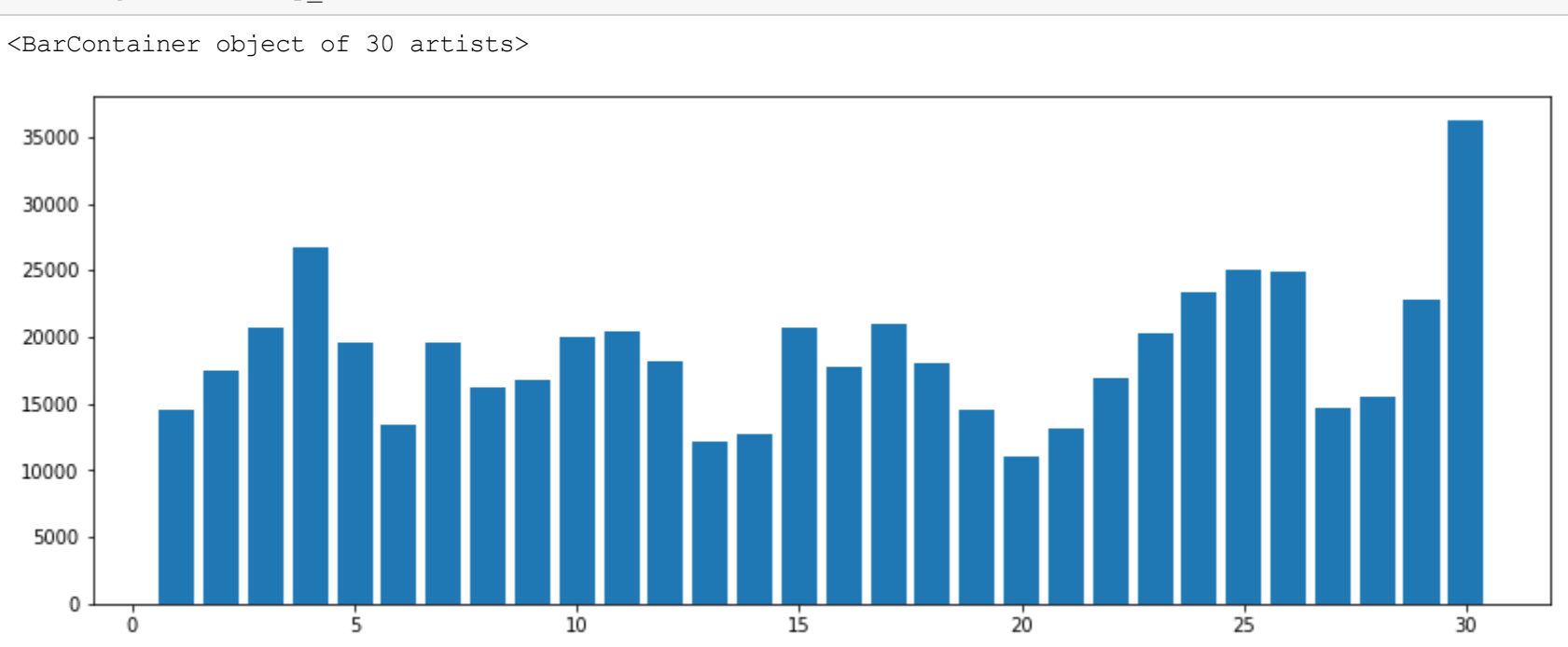


In [17]:
```
#Date of the month frequency function

def count_rows(rows):
    return len(rows)

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

Out[17]:
```
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 [18]:
```
figure(figsize=(14, 5))
bar(range(1, 31), by_date)
```

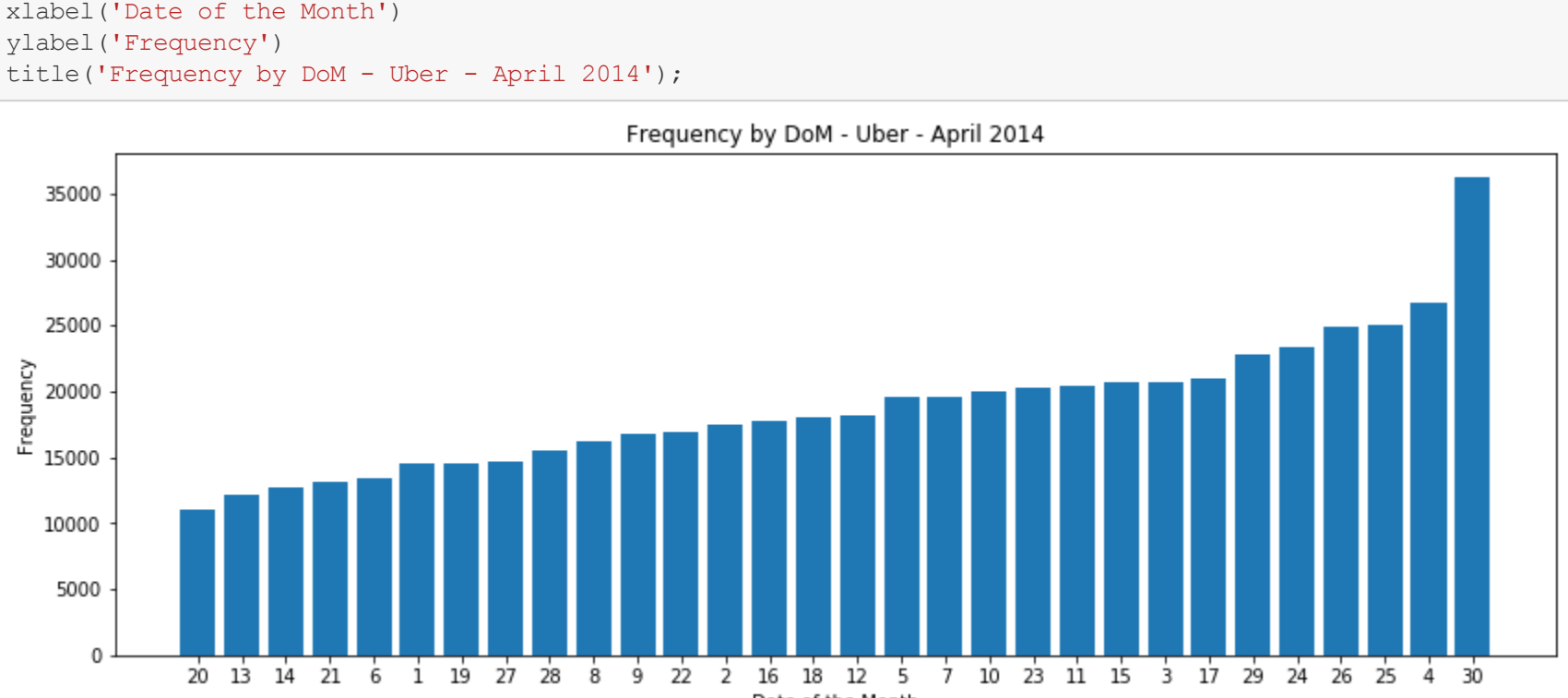
Out[18]: `<BarContainer object of 30 artists>`



In [19]:
```
#sorting by ascending values

by_date_sorted = by_date.sort_values()
by_date_sorted
```

Out[19]:
```
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
```

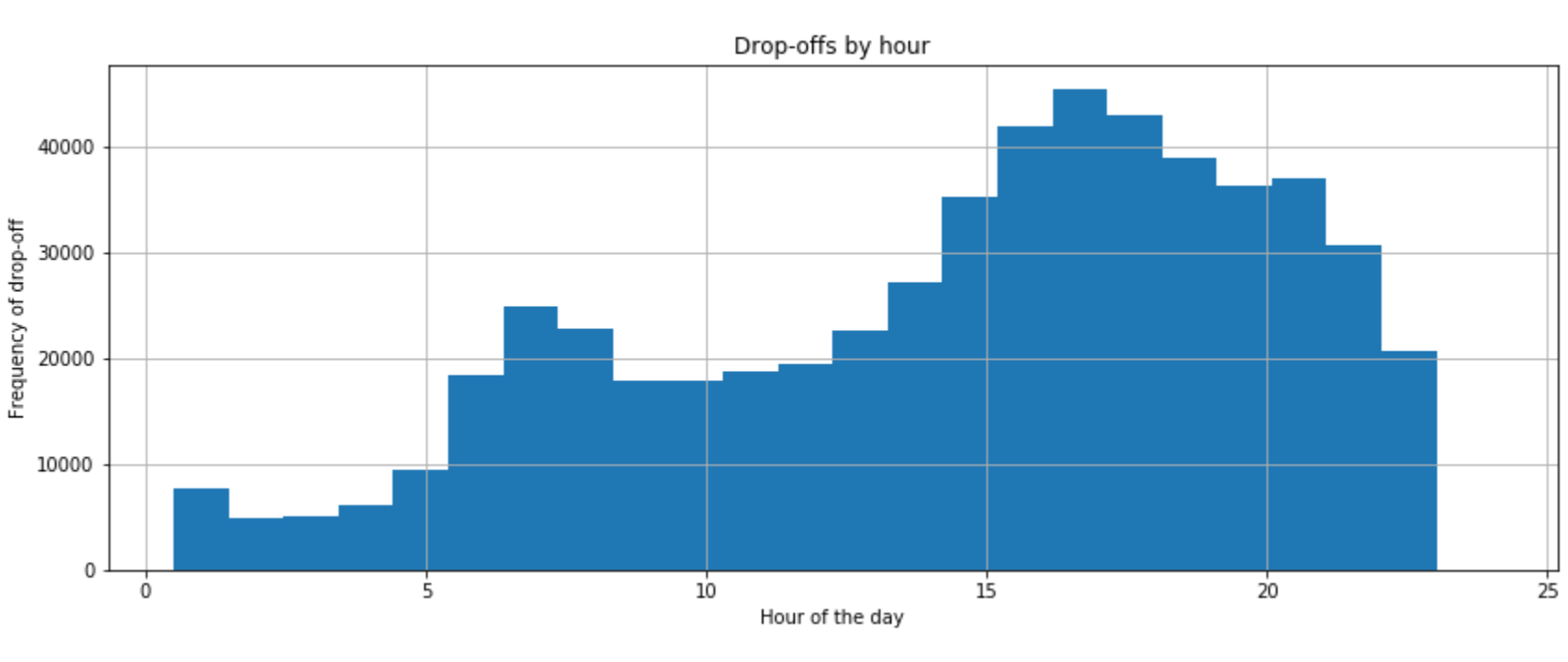
In [20]:
```
#Graphing by date sorted

figure(figsize=(14, 5))
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 - April 2014');
```



Analyzing the hour

In [21]:
```
figure(figsize=(14, 5))
hist(data.Hour, bins=24, range=(.5, 24))
grid()
xlabel('Hour of the day')
ylabel('Frequency of drop-off')
title('Drop-offs by hour')
```

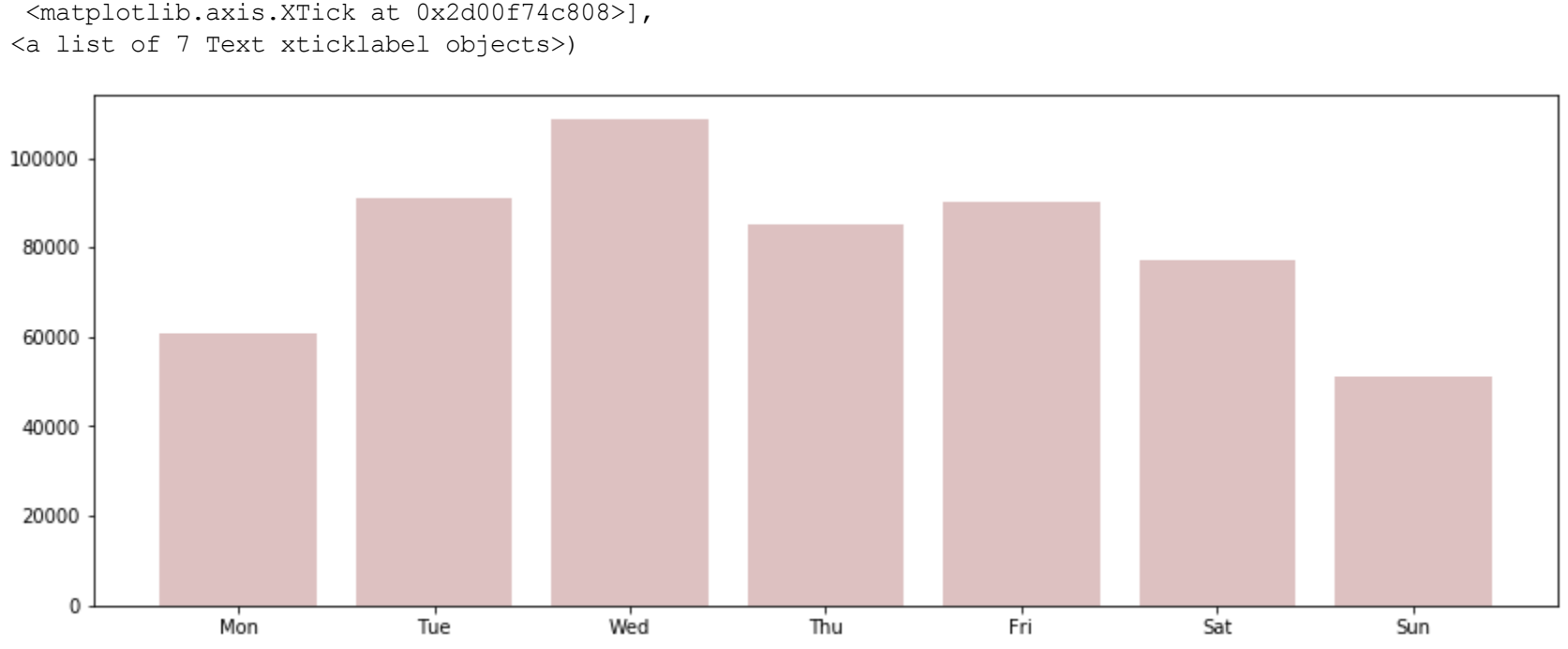
Out[21]: `Text(0.5, 1.0, 'Drop-offs by hour')`



Analyzing the weekday

In [22]:
```
figure(figsize=(14, 5))
hist(data.Weekday, bins=7, range =(-.5,6.5), rwidth=.8, color='#AA6666', alpha=.4)
xticks(range(7), 'Mon Tue Wed Thu Fri Sat Sun'.split())
```

Out[22]:
```
([<matplotlib.axis.XTick at 0x2d00f019a88>,
  <matplotlib.axis.XTick at 0x2d00f019048>,
  <matplotlib.axis.XTick at 0x2d00ef78408>,
  <matplotlib.axis.XTick at 0x2d00f7462c8>,
  <matplotlib.axis.XTick at 0x2d00f746908>,
  <matplotlib.axis.XTick at 0x2d00f74c288>,
  <matplotlib.axis.XTick at 0x2d00f74c808>],
 <a list of 7 Text xticklabel objects>)
```

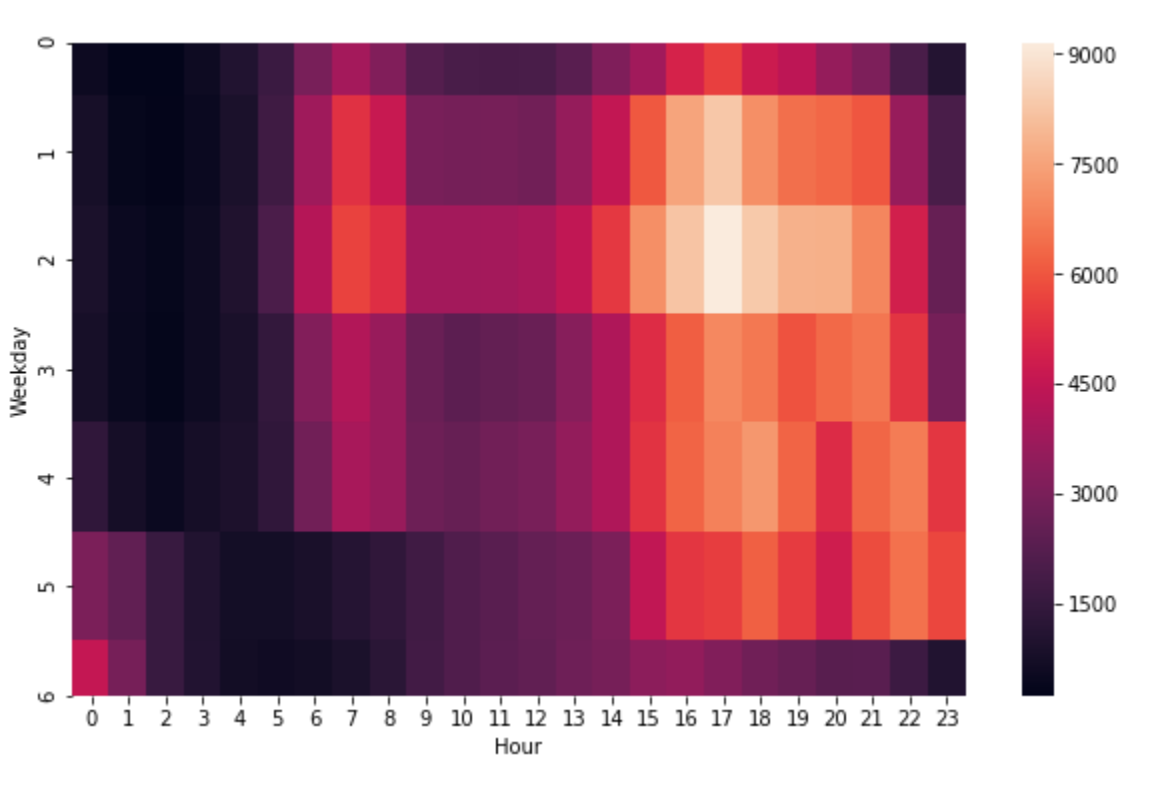


Cross Analysis (Hour, DoW)

In [23]:
```
by_cross = data.groupby('Weekday Hour'.split()).apply(count_rows).unstack()
```

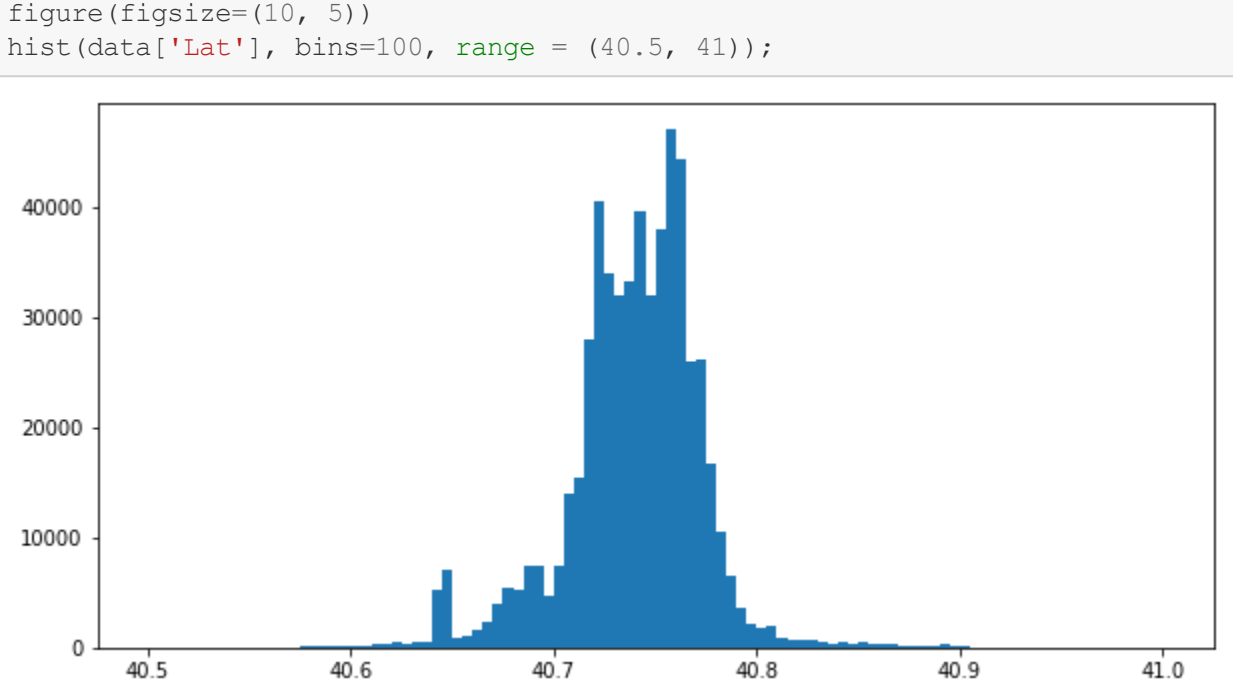
In [24]:
```
figure(figsize=(10, 6))
santulli.heatmap(by_cross)
```

Out[24]: `<matplotlib.axes._subplots.AxesSubplot at 0x2d00d865ec8>`

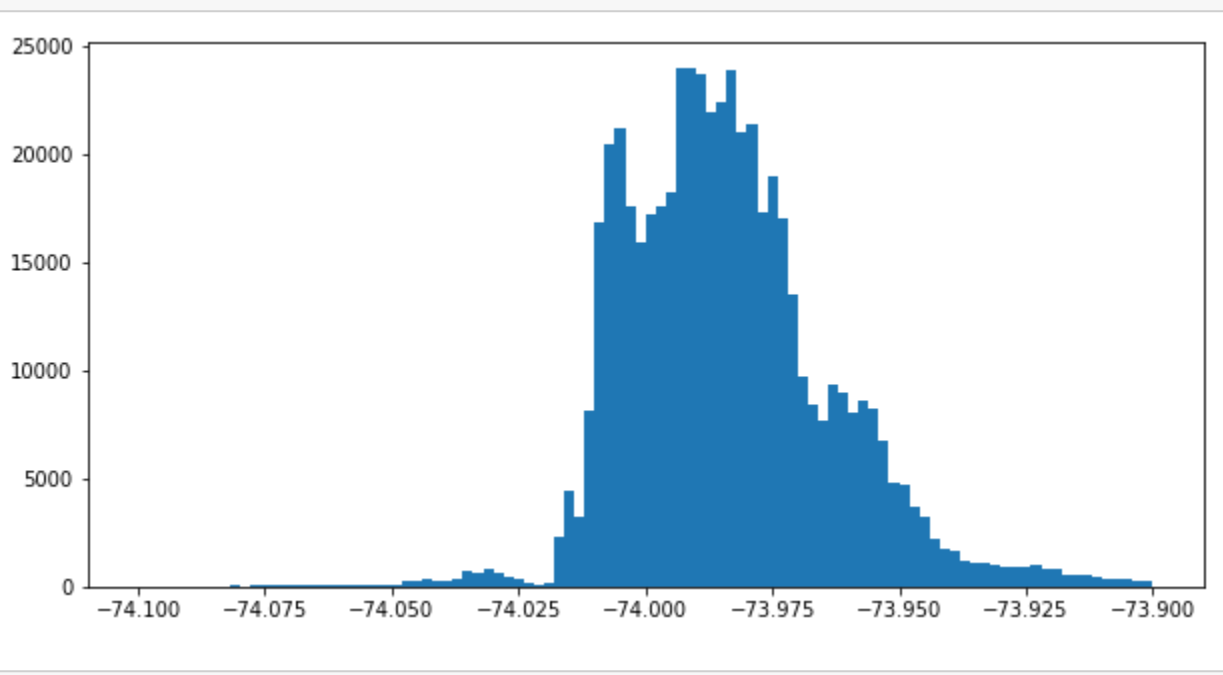


By Lat and Lon

In [25]:
```
figure(figsize=(10, 5))
hist(data['Lat'], bins=100, range = (40.5, 41));
```

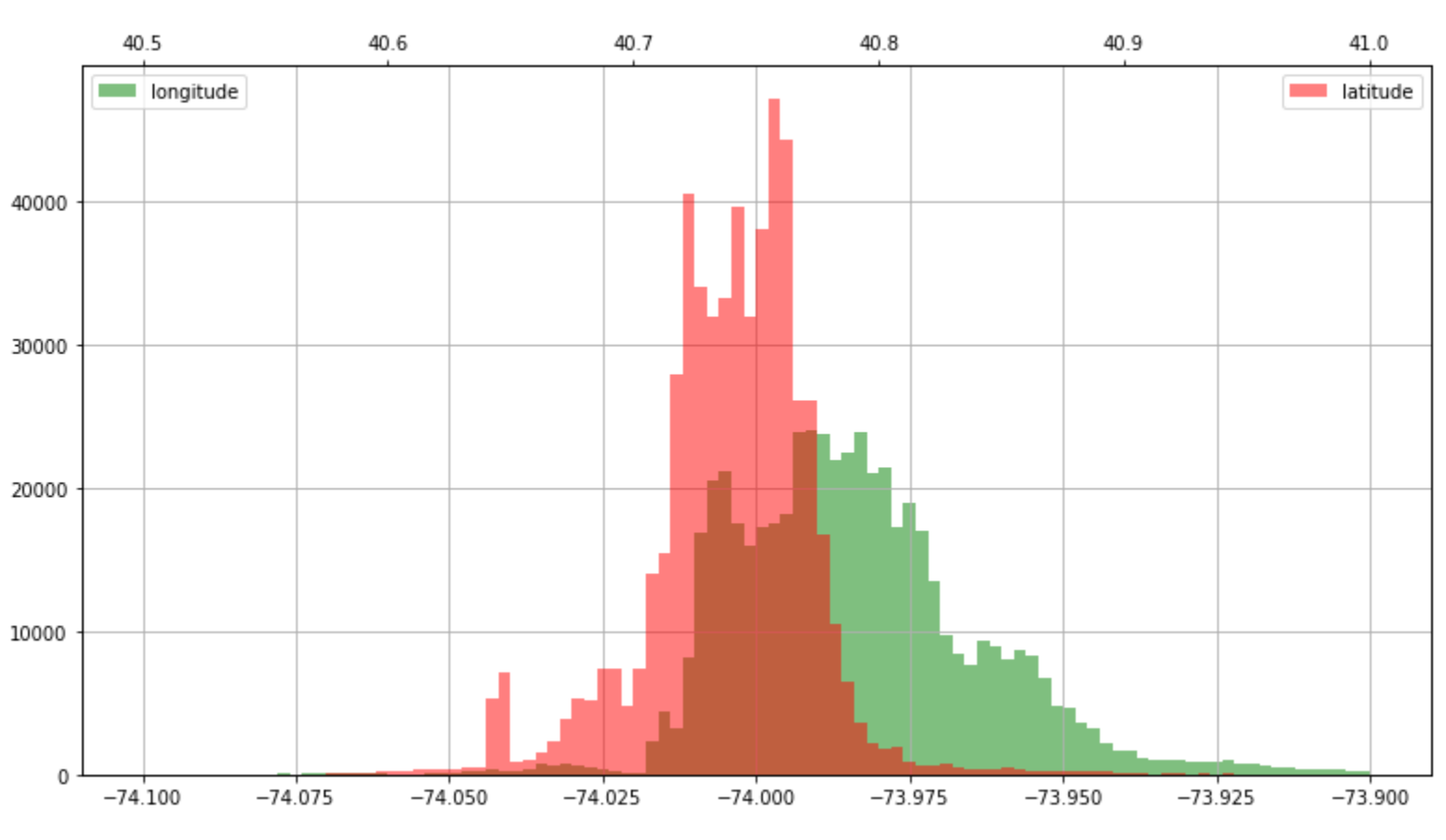


In [26]:
```
figure(figsize=(10, 5))
hist(data['Lon'], bins=100, range = (-74.1, -73.9));
```



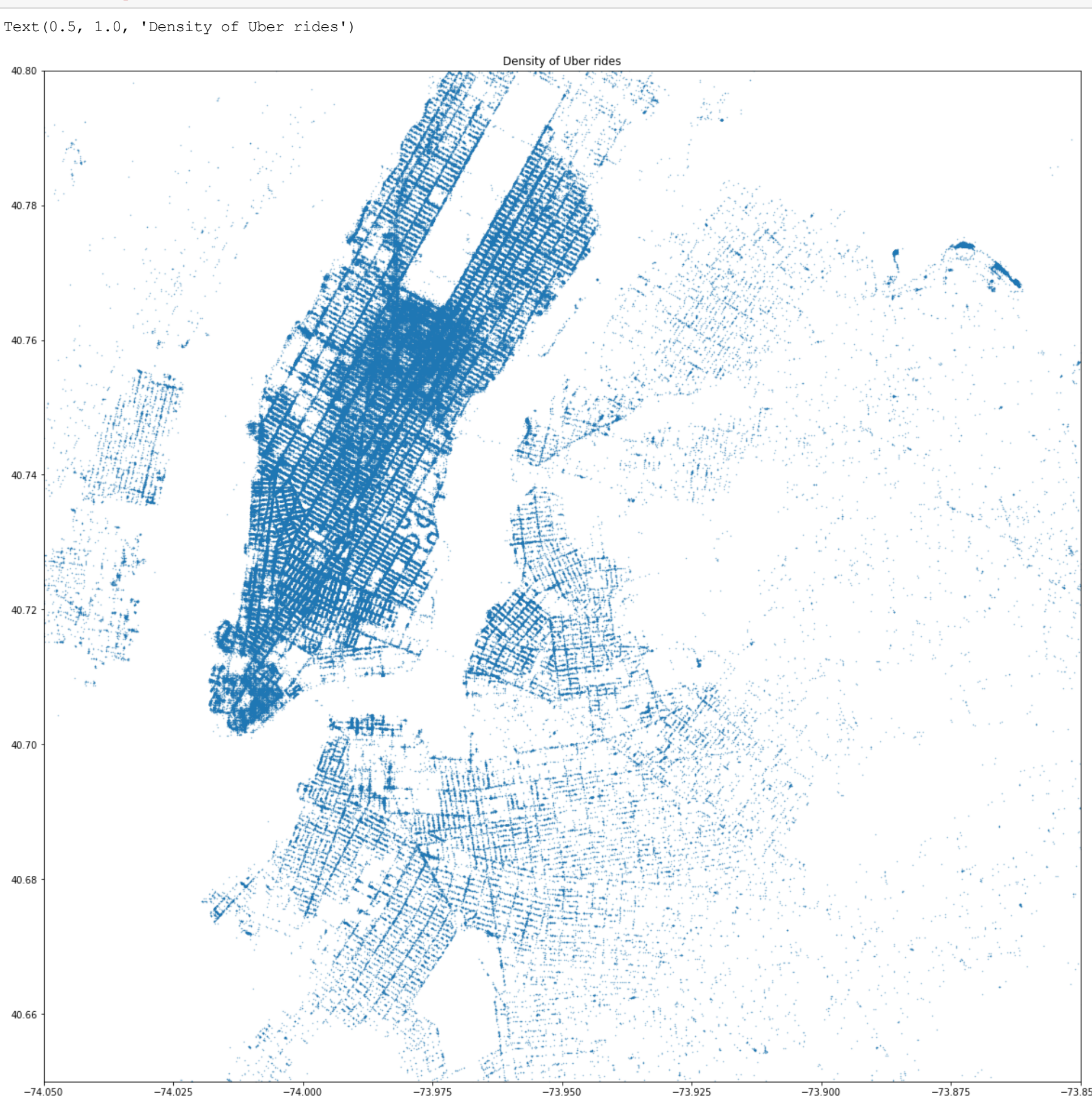
In [27]:
```
figure(figsize=(13, 7))
hist(data['Lon'], bins=100, range = (-74.1, -73.9), color='g', alpha=.5, label = 'longitude')
grid()
legend(loc='upper left')
twiny()
hist(data['Lat'], bins=100, range = (40.5, 41), color='r', alpha=.5, label = 'latitude')
legend(loc='upper right');
```

Out[27]: `''`



In [47]:
```
#plotting density of Uber rides in manhattan
figure(figsize=(20, 20))
plot(data['Lon'], data['Lat'], '.', ms=1, alpha=.5)
ylim(40.65, 40.8)
xlim(-74.05, -73.85)
title('Density of Uber rides')
```

Out[47]: `Text(0.5, 1.0, 'Density of Uber rides')`



This analysis will be continued