Entrée [1]:

```
import pandas as pd
import os
path = os.getcwd()
print(path)
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
```

```
C:\Users\hp\Documents\PSB_BI\Analysis1\Uber
```

Entrée [2]:

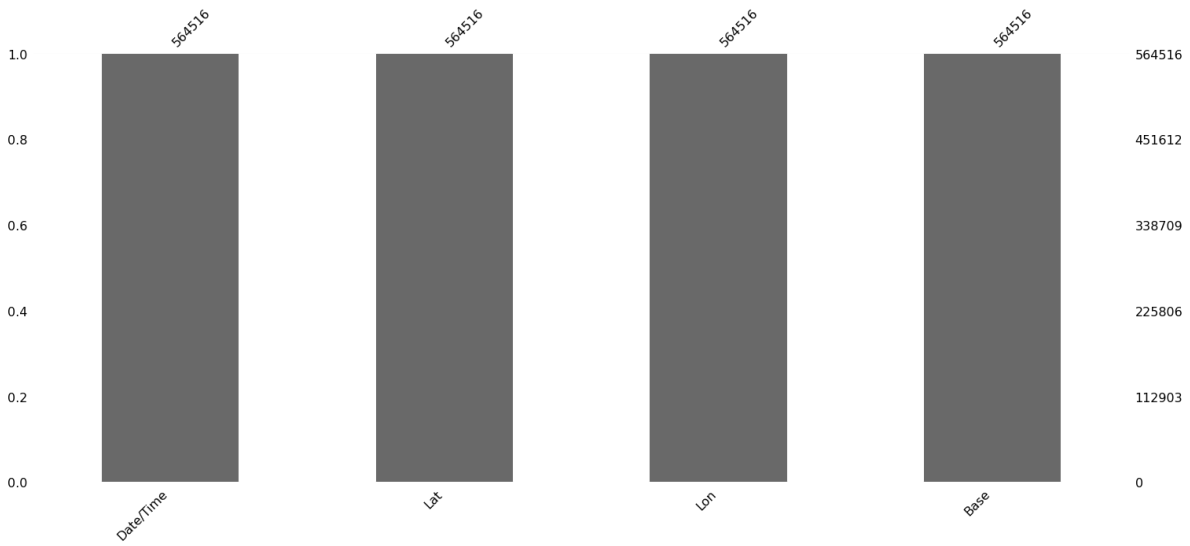
```
df = pd.read_csv("uber-raw-data-apr14.csv", delimiter=",")
```

Entrée [3]:

```
msno.bar(df)
```

Out[3]:

```
<AxesSubplot:>
```



Entrée [4]:

```
df.describe()
```

Out[4]:

	Lat	Lon
count	564516.000000	564516.000000
mean	40.740005	-73.976817
std	0.036083	0.050426
min	40.072900	-74.773300
25%	40.722500	-73.997700
50%	40.742500	-73.984800
75%	40.760700	-73.970000
max	42.116600	-72.066600

Entrée [5]:

```
df
```

Out[5]:

	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
...	...	...	...	...
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

564516 rows × 4 columns

Entrée [6]:

```
df.info()
```

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

Entrée [36]:

```
df["Date/Time"] = df["Date/Time"].map(pd.to_datetime)
```

Entrée [37]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 564516 entries, 0 to 564515
Data columns (total 7 columns):
 #   Column     Non-Null Count   Dtype         
---  ------     --------------   -----         
 0   Date/Time  564516 non-null  datetime64[ns]
 1   Lat        564516 non-null  float64       
 2   Lon        564516 non-null  float64       
 3   Base       564516 non-null  object        
 4   day        564516 non-null  int64         
 5   Weekday    564516 non-null  int64         
 6   hour       564516 non-null  int64         
dtypes: datetime64[ns](1), float64(2), int64(3), object(1)
memory usage: 30.1+ MB
```

Entrée [38]:

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

Entrée [39]:

```
df["day"] = df["Date/Time"].map(get_dom)
```

Entrée [40]:

```
df
```

Out[40]:

	Date/Time	Lat	Lon	Base	day	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
...	...	...	...	...	...	...	...
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

564516 rows × 7 columns

Entrée [41]:

```
def get_weekday (dt):
    return dt.weekday() #Weekday is a method
```

Entrée [42]:

```
df["Weekday"] = df["Date/Time"].map(get_weekday)
```

Entrée [43]:

```
df
```

Out[43]:

	Date/Time	Lat	Lon	Base	day	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
...	...	...	...	...	...	...	...
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

564516 rows × 7 columns

Entrée [15]:

```
def get_hour (dt):
    return dt.hour #hour is an attribut
```

Entrée [16]:

```
df["hour"] = df["Date/Time"].map(get_hour)
```

Entrée [17]:

```
df
```

Out[17]:

	Date/Time	Lat	Lon	Base	day	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
...	...	...	...	...	...	...	...
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

564516 rows × 7 columns

Entrée [18]:

```
df.head()
```

Out[18]:

	Date/Time	Lat	Lon	Base	day	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

Entrée [19]:

```
df.describe()
```

Out[19]:

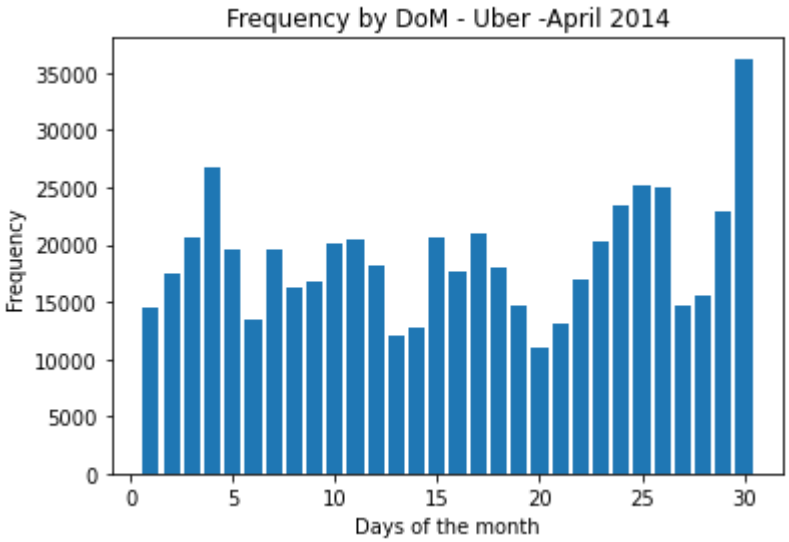
	Lat	Lon	day	Weekday	hour
count	564516.000000	564516.000000	564516.000000	564516.00000	564516.000000
mean	40.740005	-73.976817	16.117127	2.86698	14.465043
std	0.036083	0.050426	9.048139	1.82081	5.873925
min	40.072900	-74.773300	1.000000	0.00000	0.000000
25%	40.722500	-73.997700	8.000000	1.00000	10.000000
50%	40.742500	-73.984800	16.000000	3.00000	16.000000
75%	40.760700	-73.970000	24.000000	4.00000	19.000000
max	42.116600	-72.066600	30.000000	6.00000	23.000000

Entrée [20]:

```
hist = df["day"].plot.hist(bins=30, rwidth=0.8, range=(0.5,30.5), title = "Frequency by DoM
plt.xlabel("Days of the month")
```

Out[20]:

```
Text(0.5, 0, 'Days of the month')
```



Entrée [21]:

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

Entrée [22]:

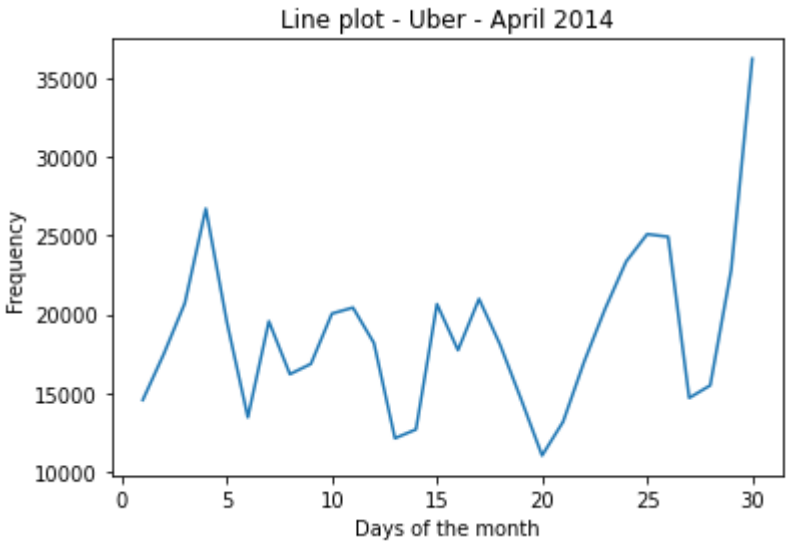
```
by_date = df.groupby("day").apply(count_rows)
by_date
```

Out[22]:

```
day
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
```

Entrée [23]:

```
plt.title("Line plot - Uber - April 2014");
plt.xlabel("Days of the month")
plt.ylabel("Frequency")
plt.plot(by_date);
```



Entrée [24]:

```
df.shape
```

Out[24]:

```
(564516, 7)
```

Entrée [25]:

```
df
```

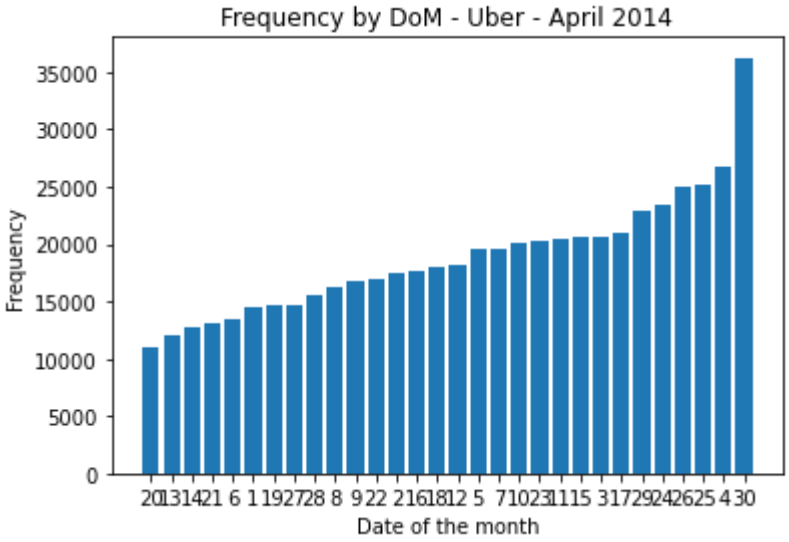
Out[25]:

	Date/Time	Lat	Lon	Base	day	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
...	...	...	...	...	...	...	...
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

564516 rows × 7 columns

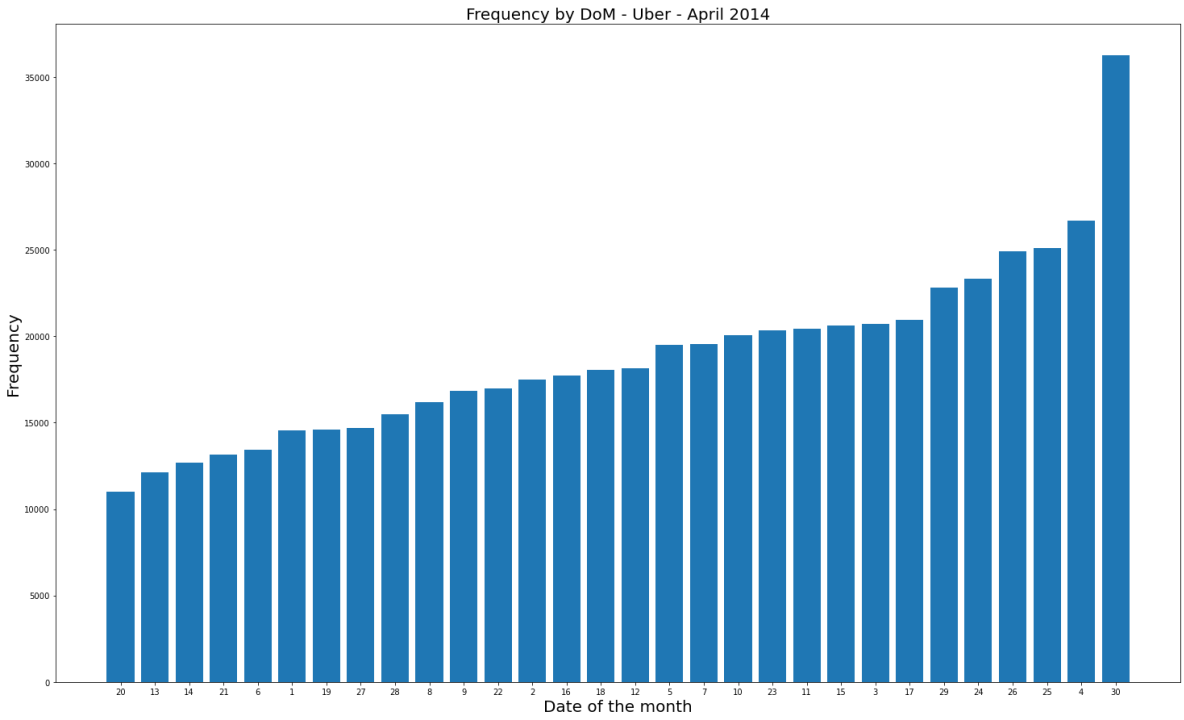
Entrée [26]:

```
plt.bar(range(1,31), by_date.sort_values())
plt.xticks(range(1,31), by_date.sort_values().index)
plt.xlabel("Date of the month")
plt.ylabel("Frequency")
plt.title("Frequency by DoM - Uber - April 2014");
```



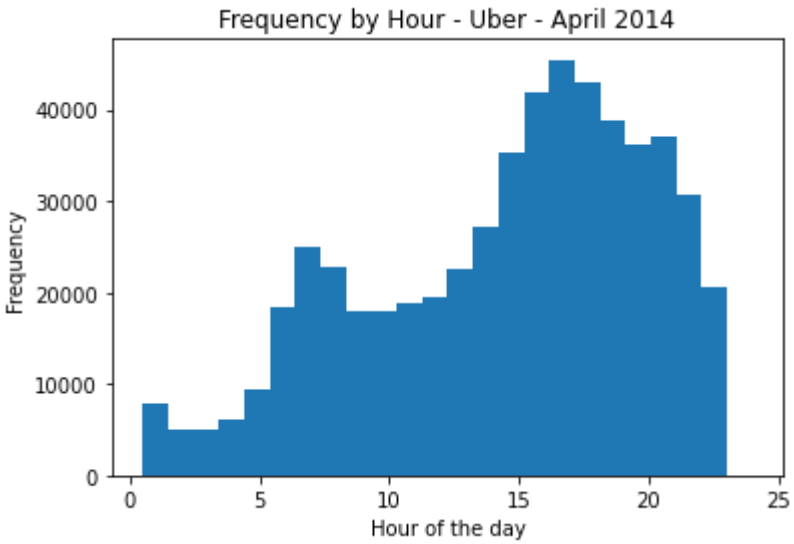
Entrée [27]:

```
plt.figure(figsize = (25, 15))
plt.bar(range(1,31), by_date.sort_values())
plt.xticks(range(1,31), by_date.sort_values().index)
plt.xlabel(("Date of the month"), fontsize =20)
plt.ylabel(("Frequency"), fontsize =20)
plt.title(("Frequency by DoM - Uber - April 2014"),fontsize =20);
```



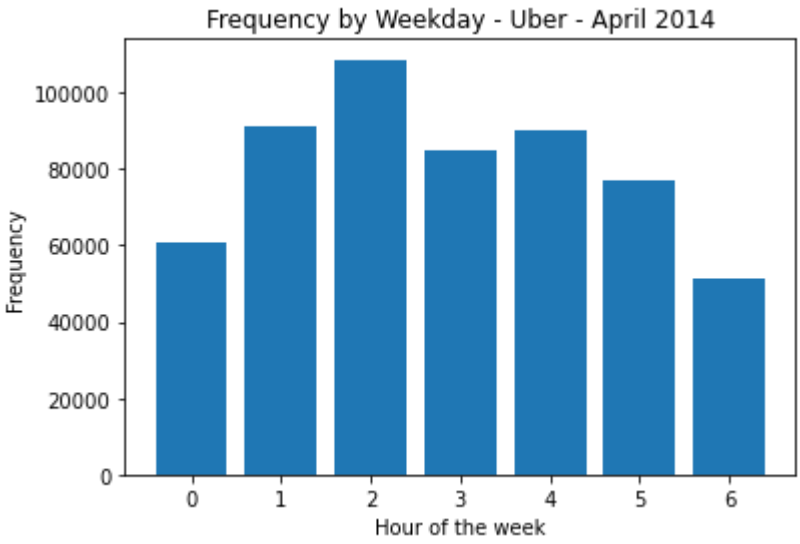
Entrée [28]:

```
plt.hist(df.hour, bins=24, range=(0.5,24))
plt.xlabel("Hour of the day")
plt.ylabel("Frequency")
plt.title("Frequency by Hour - Uber - April 2014");
```



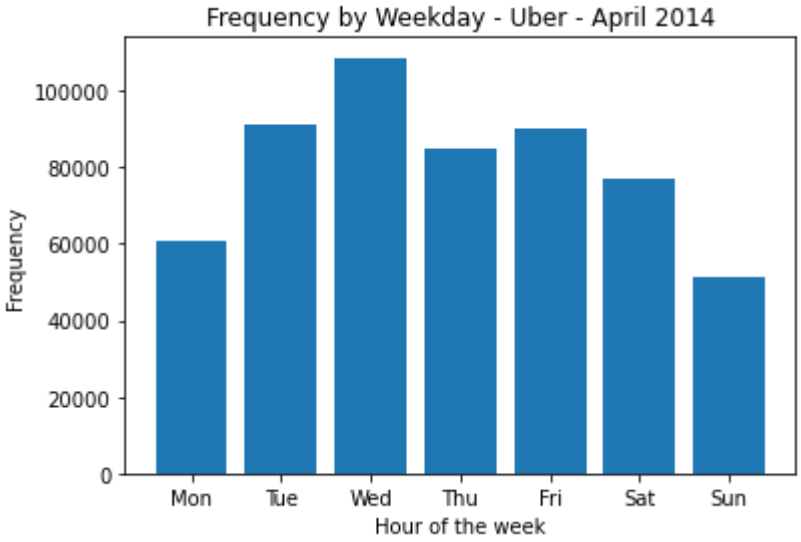
Entrée [29]:

```
plt.hist(df.Weekday, bins=7, rwidth = 0.8, range=(-.5,6.5))
plt.xlabel("Hour of the week")
plt.ylabel("Frequency")
plt.title("Frequency by Weekday - Uber - April 2014");
```



Entrée [30]:

```
plt.hist(df.Weekday, bins=7, rwidth = 0.8, range=(-.5,6.5))
plt.xlabel("Hour of the week")
plt.ylabel("Frequency")
plt.title("Frequency by Weekday - Uber - April 2014");
plt.xticks(np.arange(7), "Mon Tue Wed Thu Fri Sat Sun".split())
plt.show()
```



Entrée [31]:

```
df2 = df.groupby(["Weekday","hour"]).apply(count_rows).unstack()
df2.head()
```

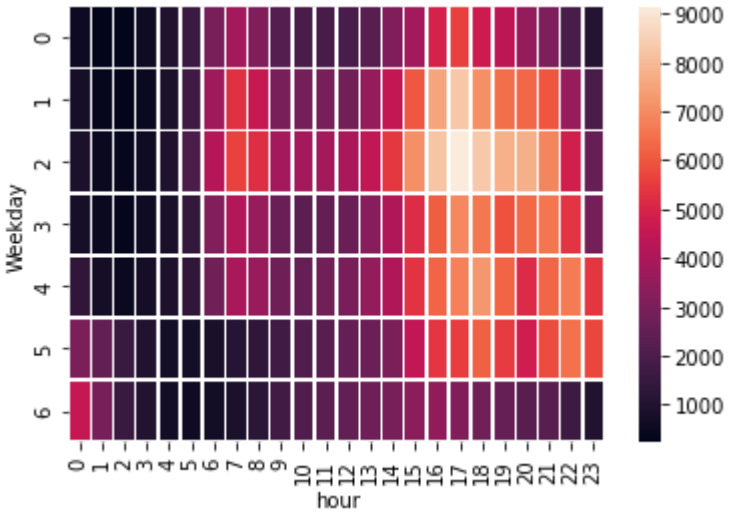
Out[31]:

hour	0	1	2	3	4	5	6	7	8	9	...	14	15	16	
Weekday															
0	518	261	238	571	1021	1619	2974	3888	3138	2211	...	3117	3818	4962	557
1	765	367	304	516	887	1734	3766	5304	4594	2962	...	4489	6042	7521	829
2	899	507	371	585	1003	1990	4230	5647	5242	3846	...	5438	7071	8213	915
3	792	459	342	567	861	1454	3179	4159	3616	2654	...	4083	5182	6149	695
4	1367	760	513	736	932	1382	2836	3943	3648	2732	...	4087	5354	6259	679

5 rows × 24 columns

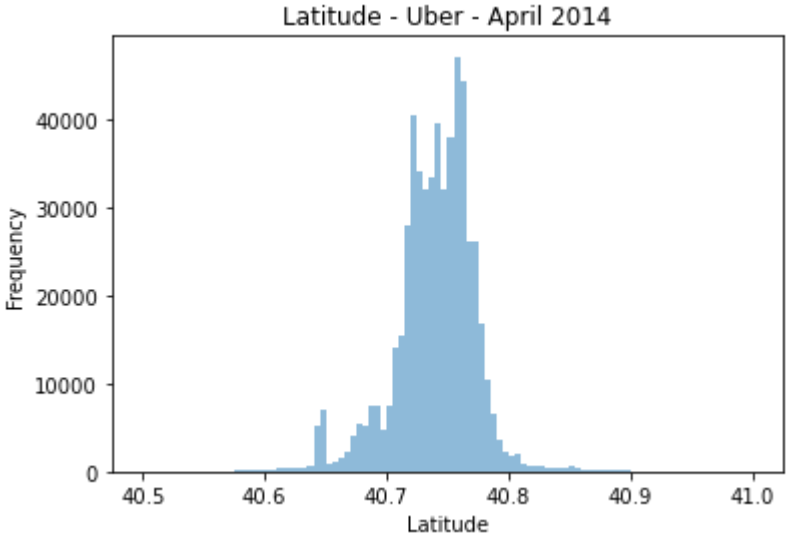
Entrée [32]:

```
heatmap = sns.heatmap(df2, linewidths = .5);
```



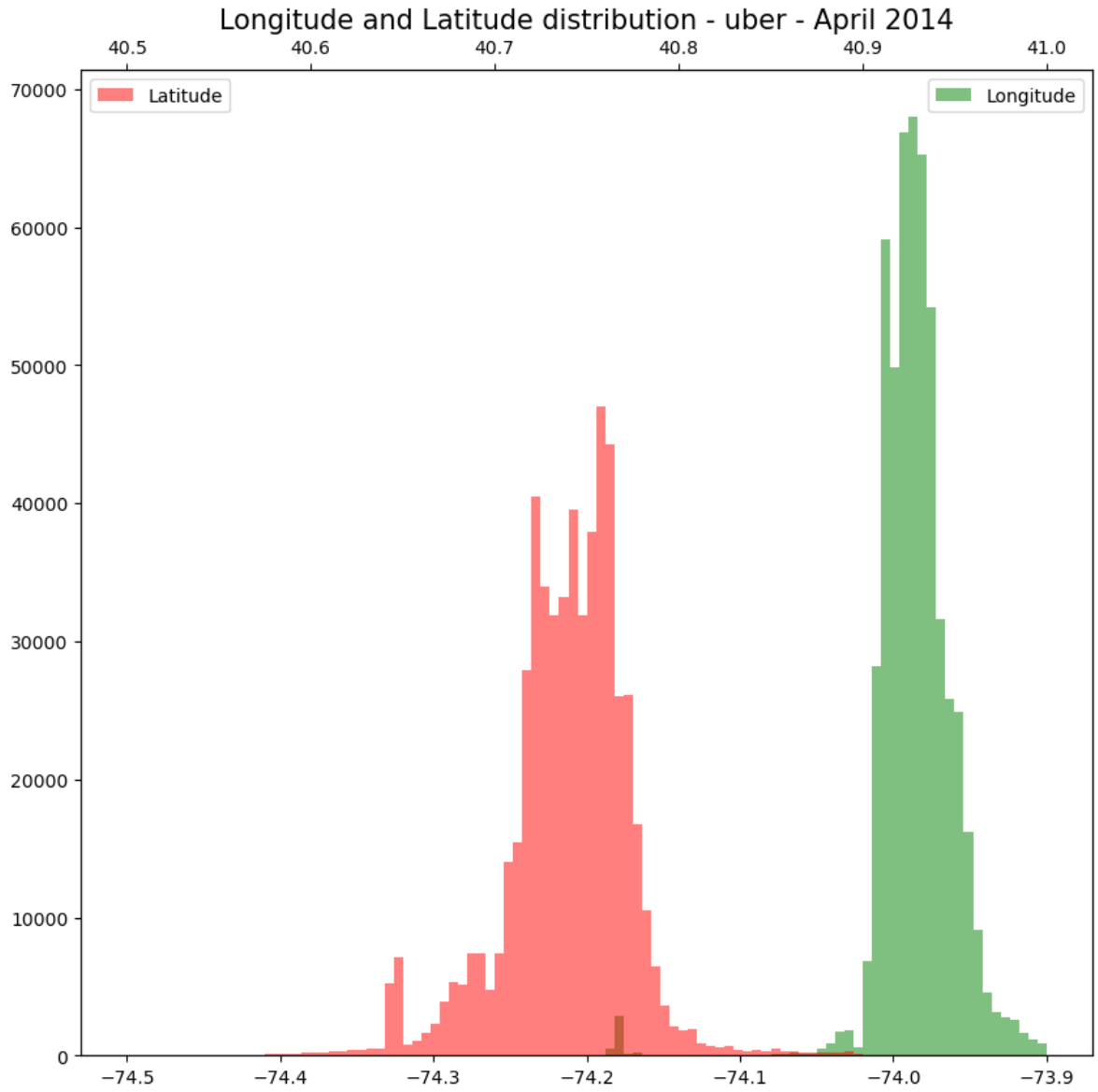
Entrée [33]:

```
plt.hist(df["Lat"], bins=100, range=(40.5,41), alpha = 0.5, label =  "Latitude")
plt.xlabel("Latitude")
plt.ylabel("Frequency")
plt.title("Latitude - Uber - April 2014");
plt.show()
```



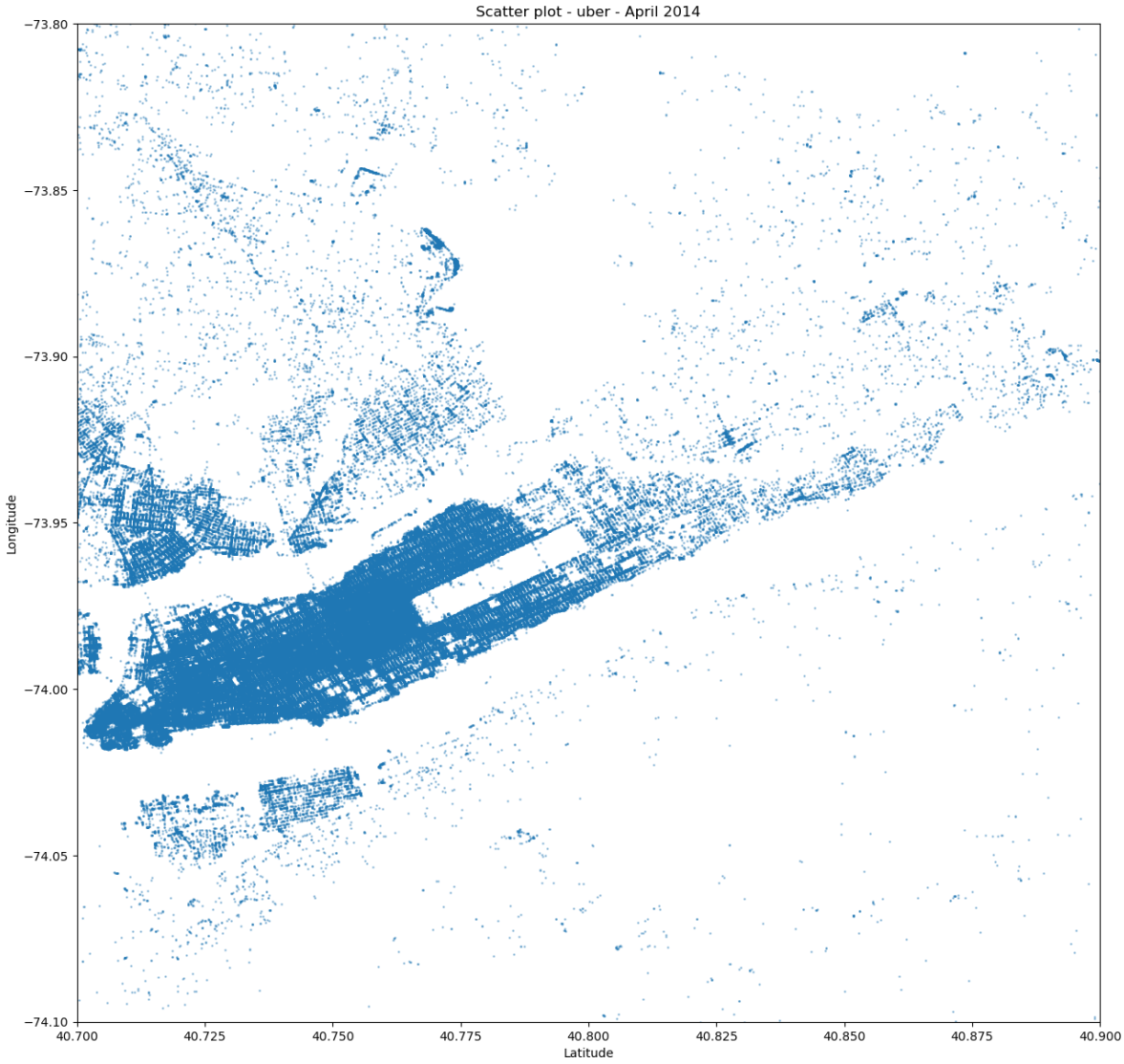
Entrée [34]:

```
plt.figure(figsize = (10, 10), dpi=100)
plt.title('Longitude and Latitude distribution - uber - April 2014', fontsize=15)
plt.hist(df["Lon"], bins=100, range=(-74.5, -73.9),color = 'g', alpha = 0.5, label =  "Long
plt.legend(loc = "best")
plt.twiny()
plt.hist(df["Lat"], bins=100, range=(40.5, 41),color = 'r', alpha = 0.5, label =  "Latitude
plt.legend(loc = 'upper left')
plt.show()
```



Entrée [35]:

```
plt.figure(figsize = (15, 15), dpi=100)
plt.title('Scatter plot - uber - April 2014')
plt.xlabel("Latitude")
plt.ylabel("Longitude")
plt.scatter(df["Lat"], df['Lon'], s=0.8, alpha = 0.4)
plt.ylim (-74.1, -73.8)
plt.xlim(40.7, 40.9);
```



Entrée [44]:

```
pip install nbconvert
```

```
Requirement already satisfied: nbconvert in c:\programdata\anaconda3\lib\sit
e-packages (6.0.7)
Requirement already satisfied: nbformat>=4.4 in c:\programdata\anaconda3\lib
\site-packages (from nbconvert) (5.1.3)
Requirement already satisfied: bleach in c:\programdata\anaconda3\lib\site-p
ackages (from nbconvert) (3.3.0)
Requirement already satisfied: jupyter-core in c:\programdata\anaconda3\lib
\site-packages (from nbconvert) (4.7.1)
Requirement already satisfied: mistune<2,>=0.8.1 in c:\programdata\anaconda3
\lib\site-packages (from nbconvert) (0.8.4)
Requirement already satisfied: pandocfilters>=1.4.1 in c:\programdata\anacon
da3\lib\site-packages (from nbconvert) (1.4.3)
Requirement already satisfied: jinja2>=2.4 in c:\programdata\anaconda3\lib\s
ite-packages (from nbconvert) (2.11.3)
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in c:\programdata\anac
onda3\lib\site-packages (from nbconvert) (0.5.3)
Requirement already satisfied: pygments>=2.4.1 in c:\programdata\anaconda3\l
ib\site-packages (from nbconvert) (2.8.1)
Requirement already satisfied: traitlets>=4.2 in c:\programdata\anaconda3\li
b\site-packages (from nbconvert) (5.0.5)
Requirement already satisfied: entrypoints>=0.2.2 in c:\programdata\anaconda
3\lib\site-packages (from nbconvert) (0.3)
Requirement already satisfied: jupyterlab-pygments in c:\programdata\anacond
a3\lib\site-packages (from nbconvert) (0.1.2)
Requirement already satisfied: defusedxml in c:\programdata\anaconda3\lib\si
te-packages (from nbconvert) (0.7.1)
Requirement already satisfied: testpath in c:\programdata\anaconda3\lib\site
-packages (from nbconvert) (0.4.4)
Requirement already satisfied: MarkupSafe>=0.23 in c:\programdata\anaconda3
\lib\site-packages (from jinja2>=2.4->nbconvert) (1.1.1)
Requirement already satisfied: async-generator in c:\programdata\anaconda3\l
ib\site-packages (from nbclient<0.6.0,>=0.5.0->nbconvert) (1.10)
Requirement already satisfied: jupyter-client>=6.1.5 in c:\programdata\anaco
nda3\lib\site-packages (from nbclient<0.6.0,>=0.5.0->nbconvert) (6.1.12)
Requirement already satisfied: nest-asyncio in c:\programdata\anaconda3\lib
\site-packages (from nbclient<0.6.0,>=0.5.0->nbconvert) (1.5.1)
Requirement already satisfied: python-dateutil>=2.1 in c:\programdata\anacon
da3\lib\site-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->n
bconvert) (2.8.1)
Requirement already satisfied: tornado>=4.1 in c:\programdata\anaconda3\lib
\site-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconver
t) (6.1)
Requirement already satisfied: pyzmq>=13 in c:\programdata\anaconda3\lib\sit
e-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert)
(20.0.0)
Requirement already satisfied: pywin32>=1.0 in c:\programdata\anaconda3\lib
\site-packages (from jupyter-core->nbconvert) (227)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in c:\programdata\ana
conda3\lib\site-packages (from nbformat>=4.4->nbconvert) (3.2.0)
Requirement already satisfied: ipython-genutils in c:\programdata\anaconda3
\lib\site-packages (from nbformat>=4.4->nbconvert) (0.2.0)
Requirement already satisfied: pyrsistent>=0.14.0 in c:\programdata\anaconda
3\lib\site-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert)
(0.17.3)
Requirement already satisfied: attrs>=17.4.0 in c:\programdata\anaconda3\lib
\site-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert) (20.
3.0)
```

```
Requirement already satisfied: setuptools in c:\programdata\anaconda3\lib\si
te-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert) (52.0.
0.post20210125)
Requirement already satisfied: six>=1.11.0 in c:\programdata\anaconda3\lib\s
ite-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert) (1.15.
0)
Requirement already satisfied: packaging in c:\programdata\anaconda3\lib\sit
e-packages (from bleach->nbconvert) (20.9)
Requirement already satisfied: webencodings in c:\programdata\anaconda3\lib
\site-packages (from bleach->nbconvert) (0.5.1)
Requirement already satisfied: pyparsing>=2.0.2 in c:\programdata\anaconda3
\lib\site-packages (from packaging->bleach->nbconvert) (2.4.7)
Note: you may need to restart the kernel to use updated packages.
```

Entrée [ ]: