In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
import os
```

In [3]:
```
os.listdir(r"/Users/haleigh/Desktop/Udemy Courses/Data Analysis Projects/Uber Project/Datasets")
```

Out[3]:
```
['other-Lyft_B02510.csv',
 'other-FHV-services_jan-aug-2015.csv',
 'other-Firstclass_B01536.csv',
 'other-Skyline_B00111.csv',
 'uber-raw-data-janjune-15_sample.csv',
 'uber-raw-data-janjune-15.csv',
 'other-American_B01362.csv',
 'uber-raw-data-apr14.csv',
 'Uber-Jan-Feb-FOIL.csv',
 'other-Highclass_B01717.csv',
 'uber-raw-data-aug14.csv',
 'uber-raw-data-sep14.csv',
 'uber-raw-data-jul14.csv',
 'other-Federal_02216.csv',
 'uber-raw-data-jun14.csv',
 'other-Carmel_B00256.csv',
 'other-Diplo_B01196.csv',
 'other-Dial7_B00887.csv',
 'uber-raw-data-may14.csv',
 'other-Prestige_B01338.csv']
```

In [4]:
```
# obtaining the dataset
uber_15 = pd.read_csv(r"/Users/haleigh/Desktop/Udemy Courses/Data Analysis Projects/Uber Project/Datasets/ube
```

In [5]:
```
# showing rows and columns shape
uber_15.shape
```

Out[5]:
```
(100000, 4)
```

In [6]:
```
# checking type
type(uber_15)
```

Out[6]:
```
pandas.core.frame.DataFrame
```

In [7]:
```
# checking for duplications
uber_15.duplicated().sum()
```

Out[7]:
```
54
```

In [8]:
```
uber_15.drop_duplicates(inplace=True)
```

In [9]:
```
uber_15.duplicated().sum()
```

Out[9]:
```
0
```

In [10]:
```
uber_15.shape
```

Out[10]:
```
(99946, 4)
```

In [11]:
```
uber_15.dtypes
```

Out[11]:
```
Dispatching_base_num    object
Pickup_date             object
Affiliated_base_num     object
locationID               int64
dtype: object
```

In [12]:
```
uber_15.isnull().sum()
```

Out[12]:
```
Dispatching_base_num       0
Pickup_date                0
Affiliated_base_num     1116
locationID                 0
dtype: int64
```

In [13]:
```
# find the first pick up date
uber_15['Pickup_date'][0]
```

Out[13]:
```
'2015-05-02 21:43:00'
```

In [14]:
```
uber_15['Pickup_date'] = pd.to_datetime(uber_15['Pickup_date'])
```

In [15]:
```
uber_15['Pickup_date'].dtype
```

Out[15]:
```
dtype('<M8[ns]')
```

In [16]:
```
uber_15['Pickup_date'][0]
```

Out[16]:
```
Timestamp('2015-05-02 21:43:00')
```

In [17]:
```
type(uber_15['Pickup_date'][0])
```

Out[17]:
```
pandas._libs.tslibs.timestamps.Timestamp
```

In [18]:
```
uber_15.dtypes
```

Out[18]:
```
Dispatching_base_num            object
Pickup_date             datetime64[ns]
Affiliated_base_num             object
locationID                       int64
dtype: object
```

In [19]:
```
# analysing which month has the maximum uber pickups in NYC?
# extract the month feature
uber_15
```

Out[19]:

	Dispatching_base_num	Pickup_date	Affiliated_base_num	locationID
0	B02617	2015-05-02 21:43:00	B02764	237
1	B02682	2015-01-20 19:52:59	B02682	231
2	B02617	2015-03-19 20:26:00	B02617	161
3	B02764	2015-04-10 17:38:00	B02764	107
4	B02764	2015-03-23 07:03:00	B00111	140
...	...	...	...	...
99995	B02764	2015-04-13 16:12:00	B02764	234
99996	B02764	2015-03-06 21:32:00	B02764	24
99997	B02598	2015-03-19 19:56:00	B02598	17
99998	B02682	2015-05-02 16:02:00	B02682	68
99999	B02764	2015-06-24 16:04:00	B02764	125

99946 rows × 4 columns

uber_15['month'] = uber_15['Pickup_date'].dt.month_name()

In [20]:
```
uber_15['month'] = uber_15['Pickup_date'].dt.month_name()
```

In [21]:
```
uber_15['month']
```

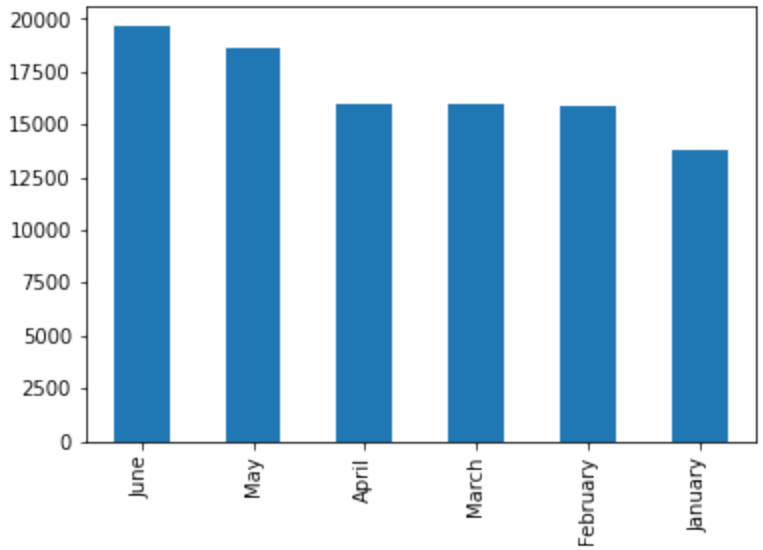
Out[21]:
```
0           May
1       January
2         March
3         April
4         March
          ...
99995     April
99996     March
99997     March
99998       May
99999      June
Name: month, Length: 99946, dtype: object
```

In [22]:
```
#freq. table
uber_15['month'].value_counts()
```

Out[22]:
```
June         19620
May          18660
April        15982
March        15969
February     15896
January      13819
Name: month, dtype: int64
```

In [23]:
```
uber_15['month'].value_counts().plot(kind='bar')
```

Out[23]:
```
<AxesSubplot:>
```



In [24]:
```
uber_15['Weekday'] = uber_15['Pickup_date'].dt.day_name()
uber_15['Day'] = uber_15['Pickup_date'].dt.day
uber_15['Hour'] = uber_15['Pickup_date'].dt.hour
uber_15['Minute'] = uber_15['Pickup_date'].dt.minute
```

In [25]:
```
uber_15.head(4)
```

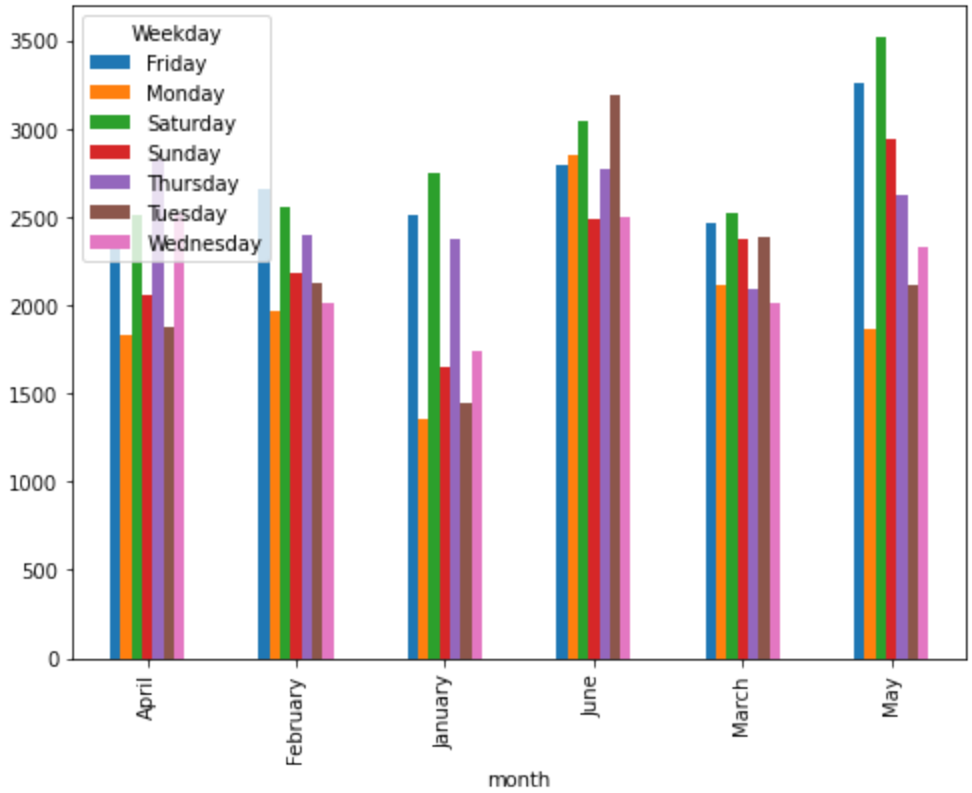
Out[25]:

	Dispatching_base_num	Pickup_date	Affiliated_base_num	locationID	month	Weekday	Day	Hour	Minute
0	B02617	2015-05-02 21:43:00	B02764	237	May	Saturday	2	21	43
1	B02682	2015-01-20 19:52:59	B02682	231	January	Tuesday	20	19	52
2	B02617	2015-03-19 20:26:00	B02617	161	March	Thursday	19	20	26
3	B02764	2015-04-10 17:38:00	B02764	107	April	Friday	10	17	38

In [26]:
```
pivot = pd.crosstab(index=uber_15['month'], columns=uber_15['Weekday'])
```

In [27]:
```
# create grouped bar chart
pivot.plot(kind='bar', figsize = (8,6))
```

Out[27]:
```
<AxesSubplot:xlabel='month'>
```



In [31]:
```
# finding hourly rush in NYC for all days
summary = uber_15.groupby(['Weekday', 'Hour'], as_index = False).size()
```

In [32]:
```
summary
```

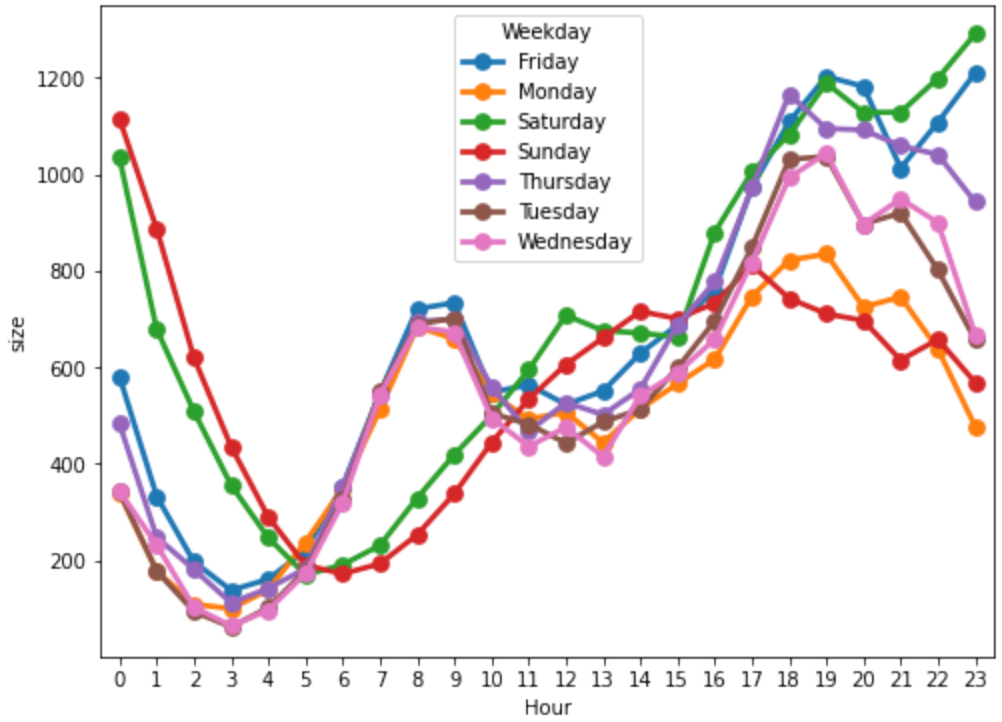
Out[32]:

	Weekday	Hour	size
0	Friday	0	581
1	Friday	1	333
2	Friday	2	197
3	Friday	3	138
4	Friday	4	161
...	...	...	...
163	Wednesday	19	1044
164	Wednesday	20	897
165	Wednesday	21	949
166	Wednesday	22	900
167	Wednesday	23	669

168 rows × 3 columns

In [33]:
```
# making point plot
plt.figure(figsize=(8,6))
sns.pointplot(x="Hour", y="size", hue="Weekday", data=summary)
```

Out[33]:
```
<AxesSubplot:xlabel='Hour', ylabel='size'>
```



In [36]:
```
# show which base number has the MOST number of active vehicles
# pull from data set to get the base_number and active vehicles
uber_foil = pd.read_csv(r"/Users/haleigh/Desktop/Udemy Courses/Data Analysis Projects/Uber Project/Datasets/U
```

In [37]:
```
uber_foil.shape
```

Out[37]:
```
(354, 4)
```

In [38]:
```
uber_foil.head(3)
```

Out[38]:

	dispatching_base_number	date	active_vehicles	trips
0	B02512	1/1/2015	190	1132
1	B02765	1/1/2015	225	1765
2	B02764	1/1/2015	3427	29421

In [39]:
```
# making box plot
!pip install chart_studio
```

```
!pip install plotly
```

```
Collecting chart_studio
  Downloading chart_studio-1.1.0-py3-none-any.whl.metadata (1.3 kB)
Collecting plotly (from chart_studio)
  Downloading plotly-5.24.1-py3-none-any.whl.metadata (7.3 kB)
Requirement already satisfied: requests in /Users/haleigh/Library/Python/3.10/lib/python/site-packages (from
chart_studio) (2.32.3)
Collecting retrying>=1.3.3 (from chart_studio)
  Downloading retrying-1.3.4-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: six in /Users/haleigh/Library/Python/3.10/lib/python/site-packages (from chart
_studio) (1.16.0)
Collecting tenacity>=6.2.0 (from plotly->chart_studio)
  Downloading tenacity-9.0.0-py3-none-any.whl.metadata (1.2 kB)
Requirement already satisfied: packaging in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.1
0/site-packages (from plotly->chart_studio) (21.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /Users/haleigh/Library/Python/3.10/lib/python/site
-packages (from requests->chart_studio) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /Users/haleigh/Library/Python/3.10/lib/python/site-packages (f
rom requests->chart_studio) (3.8)
Requirement already satisfied: urllib3<3,>=1.21.1 in /Library/Frameworks/Python.framework/Versions/3.10/lib/p
ython3.10/site-packages (from requests->chart_studio) (1.26.8)
Requirement already satisfied: certifi>=2017.4.17 in /Library/Frameworks/Python.framework/Versions/3.10/lib/p
ython3.10/site-packages (from requests->chart_studio) (2024.7.4)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /Library/Frameworks/Python.framework/Versions/3.1
0/lib/python3.10/site-packages (from packaging->plotly->chart_studio) (3.0.7)
Downloading chart_studio-1.1.0-py3-none-any.whl (64 kB)
Downloading retrying-1.3.4-py3-none-any.whl (11 kB)
Downloading plotly-5.24.1-py3-none-any.whl (19.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 19.1/19.1 MB 44.8 MB/s eta 0:00:0000:010:01
Downloading tenacity-9.0.0-py3-none-any.whl (28 kB)
Installing collected packages: tenacity, retrying, plotly, chart_studio
Successfully installed chart_studio-1.1.0 plotly-5.24.1 retrying-1.3.4 tenacity-9.0.0
Requirement already satisfied: plotly in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/si
te-packages (5.24.1)
Requirement already satisfied: tenacity>=6.2.0 in /Library/Frameworks/Python.framework/Versions/3.10/lib/pyth
on3.10/site-packages (from plotly) (9.0.0)
Requirement already satisfied: packaging in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.1
0/site-packages (from plotly) (21.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /Library/Frameworks/Python.framework/Versions/3.1
0/lib/python3.10/site-packages (from packaging->plotly) (3.0.7)
```

In [40]:

```
import chart_studio.plotly as py
import plotly.graph_objs as go
import plotly.express as px
```

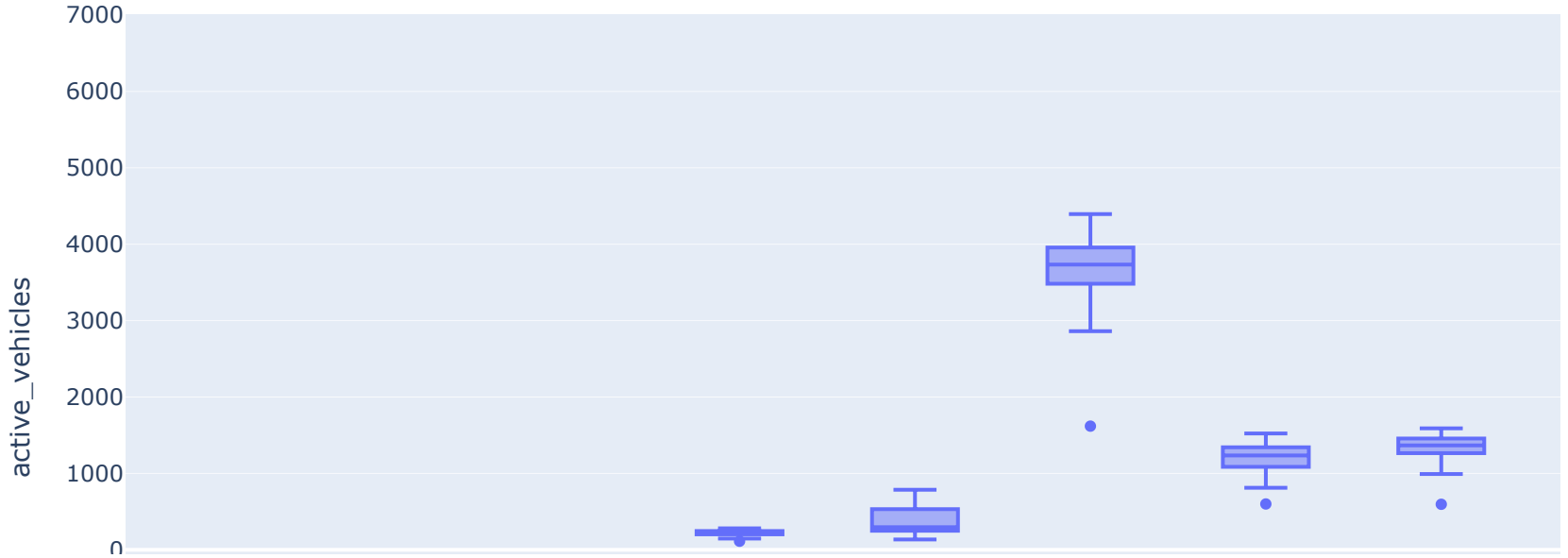
```
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
```

In [41]:
```
init_notebook_mode(connected=True)
```

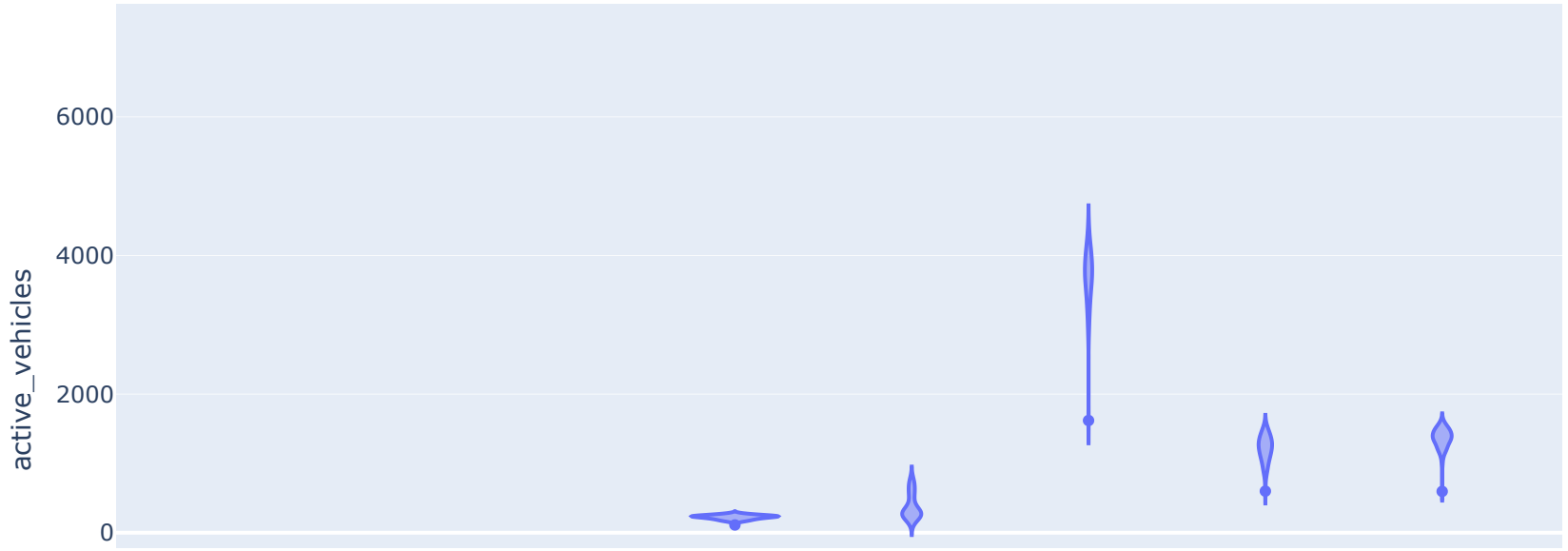
In [42]:
```
uber_foil.columns
```

Out[42]:
```
Index(['dispatching_base_number', 'date', 'active_vehicles', 'trips'], dtype='object')
```

In [43]:
```
px.box(x='dispatching_base_number', y='active_vehicles', data_frame = uber_foil)
```



In [45]:
```
px.violin(x='dispatching_base_number', y='active_vehicles', data_frame = uber_foil)
```



In [46]:
```
# collect entire data for data analysis
# Big Data --> new .csv file
files = [
    'uber-raw-data-apr14.csv', 'uber-raw-data-aug14.csv',
    'uber-raw-data-jul14.csv', 'uber-raw-data-jun14.csv',
    'uber-raw-data-may14.csv', 'uber-raw-data-sep14.csv'
]
# concatenate these files
```

In [51]:
```
final = pd.DataFrame()
path = r"/Users/haleigh/Desktop/Udemy Courses/Data Analysis Projects/Uber Project/Datasets"
```

```
for file in files:
    current_df = pd.read_csv(path+'/'+file)
    final = pd.concat([current_df, final])
```

In [49]:
```
final.shape
```

Out[49]:
```
(18904806, 8)
```

In [52]:
```
final.duplicated().sum()
```

Out[52]:
```
1080806
```

In [53]:
```
final.drop_duplicates(inplace=True)
```

In [54]:
```
final.shape
```

Out[54]:
```
(17824000, 8)
```

In [55]:
```
final.head(3)
```

Out[55]:

	Date/Time	Lat	Lon	Base	Dispatching_base_num	Pickup_date	Affiliated_base_num	locationID
0	9/1/2014 0:01:00	40.2201	-74.0021	B02512	NaN	NaN	NaN	NaN
1	9/1/2014 0:01:00	40.7500	-74.0027	B02512	NaN	NaN	NaN	NaN
2	9/1/2014 0:03:00	40.7559	-73.9864	B02512	NaN	NaN	NaN	NaN

In [61]:
```
# spatial analysis / locations for uber pick up rushes
# map based visualization
rush_uber = final.groupby(['Lat', 'Lon'], as_index=False).size()
```

In [62]:
```
rush_uber.head(6)
```

Out[62]:

	Lat	Lon	size
0	39.6569	-74.2258	1
1	39.6686	-74.1607	1
2	39.7214	-74.2446	1
3	39.8416	-74.1512	1
4	39.9055	-74.0791	1
5	39.9196	-74.1112	1

In [63]:
```
# create base map
!pip install folium
```

```
Collecting folium
  Downloading folium-0.17.0-py2.py3-none-any.whl.metadata (3.8 kB)
Collecting branca>=0.6.0 (from folium)
  Downloading branca-0.8.0-py3-none-any.whl.metadata (1.5 kB)
Requirement already satisfied: jinja2>=2.9 in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.
10/site-packages (from folium) (3.0.3)
Requirement already satisfied: numpy in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/sit
e-packages (from folium) (1.22.1)
Requirement already satisfied: requests in /Users/haleigh/Library/Python/3.10/lib/python/site-packages (from
folium) (2.32.3)
Collecting xyzservices (from folium)
  Downloading xyzservices-2024.9.0-py3-none-any.whl.metadata (4.1 kB)
Requirement already satisfied: MarkupSafe>=2.0 in /Library/Frameworks/Python.framework/Versions/3.10/lib/pyth
on3.10/site-packages (from jinja2>=2.9->folium) (2.0.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /Users/haleigh/Library/Python/3.10/lib/python/site
-packages (from requests->folium) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /Users/haleigh/Library/Python/3.10/lib/python/site-packages (f
rom requests->folium) (3.8)
Requirement already satisfied: urllib3<3,>=1.21.1 in /Library/Frameworks/Python.framework/Versions/3.10/lib/p
ython3.10/site-packages (from requests->folium) (1.26.8)
Requirement already satisfied: certifi>=2017.4.17 in /Library/Frameworks/Python.framework/Versions/3.10/lib/p
ython3.10/site-packages (from requests->folium) (2024.7.4)
Downloading folium-0.17.0-py2.py3-none-any.whl (108 kB)
Downloading branca-0.8.0-py3-none-any.whl (25 kB)
Downloading xyzservices-2024.9.0-py3-none-any.whl (85 kB)
Installing collected packages: xyzservices, branca, folium
Successfully installed branca-0.8.0 folium-0.17.0 xyzservices-2024.9.0
```

In [64]:
```
import folium
```

In [65]:
```
basemap = folium.Map()
```

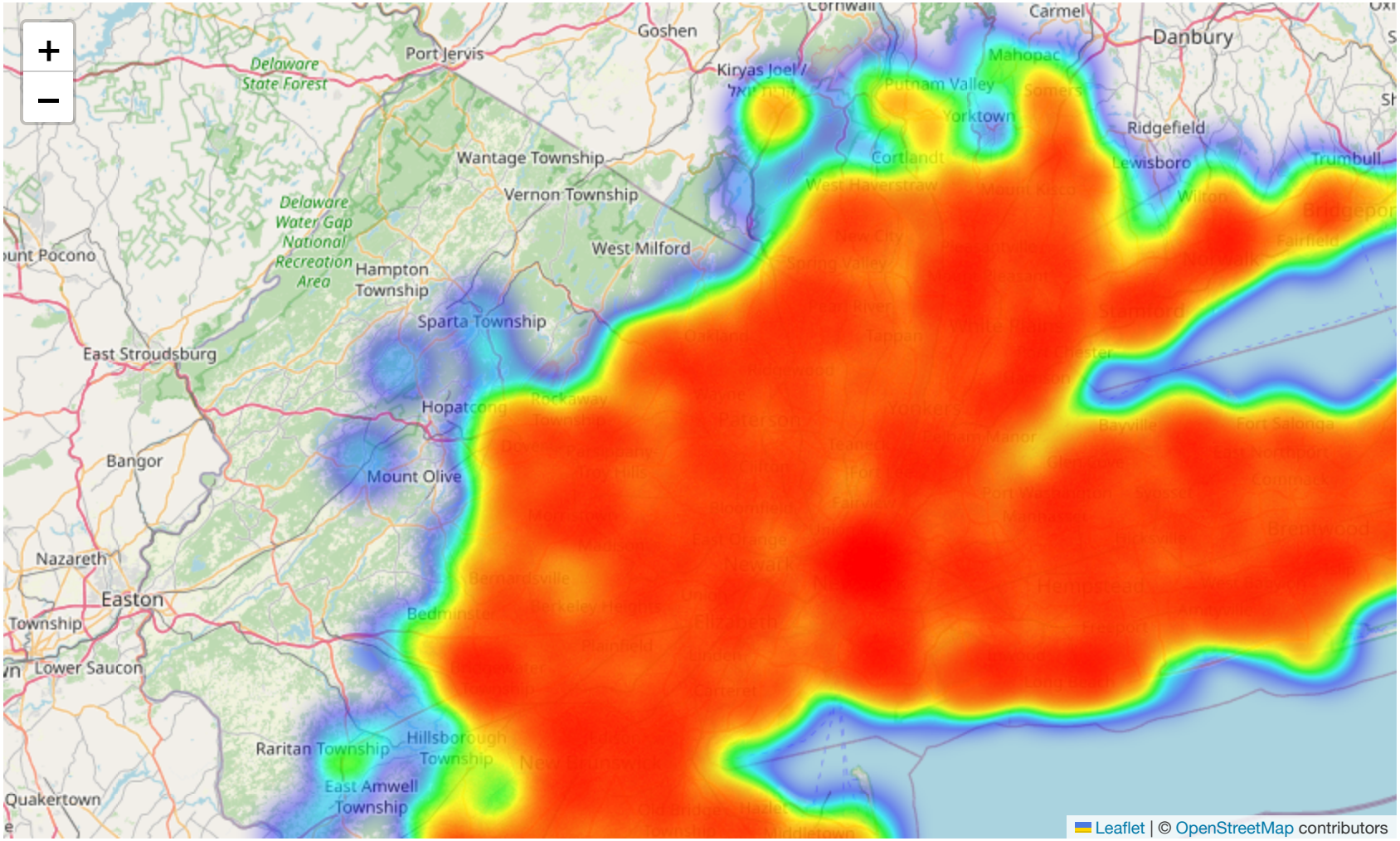
In [66]:
```
from folium.plugins import HeatMap
```

In [67]:
```
HeatMap(rush_uber).add_to(basemap)
```

Out[67]:
```
<folium.plugins.heat_map.HeatMap at 0x31a98e2f0>
```

In [68]:
```
# zoom in to see NYC and Manhattan
basemap
```

Out[68]:



In [71]:
```
final['Date/Time'][0]
```

Out[71]:
```
0    9/1/2014 0:01:00
0    5/1/2014 0:02:00
0    6/1/2014 0:00:00
0    7/1/2014 0:03:00
0                 NaN
0                 NaN
0    8/1/2014 0:03:00
0    4/1/2014 0:11:00
Name: Date/Time, dtype: object
```

In [73]:
```
# pair wise analysis to figure out the rush (on hr and weekday)
final['Date/Time'] = pd.to_datetime(final['Date/Time'], format="%m/%d/%Y %H:%M:%S")
```

In [74]:
```
final['Date/Time'].dtype
```

Out[74]:
```
dtype('<M8[ns]')
```

In [79]:
```
final['Day'] = final['Date/Time'].dt.day
final['Hour'] = final['Date/Time'].dt.hour
```

In [80]:
```
final.head(4)
```

Out[80]:

	Date/Time	Lat	Lon	Base	Dispatching_base_num	Pickup_date	Affiliated_base_num	locationID	Day	Hour
0	2014-09-01 00:01:00	40.2201	-74.0021	B02512	NaN	NaN	NaN	NaN	1.0	0.0
1	2014-09-01 00:01:00	40.7500	-74.0027	B02512	NaN	NaN	NaN	NaN	1.0	0.0
2	2014-09-01 00:03:00	40.7559	-73.9864	B02512	NaN	NaN	NaN	NaN	1.0	0.0
3	2014-09-01 00:06:00	40.7450	-73.9889	B02512	NaN	NaN	NaN	NaN	1.0	0.0

In [82]:
```
pivot = final.groupby(['Day', 'Hour']).size().unstack()
```

In [83]:
```
pivot
```

Out[83]:

Hour	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	...	14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0
Day																			
1.0	3178	1944	1256	1308	1429	2126	3664	5380	5292	4617	...	6933	7910	8633	9511	8604	8001	7315	7803
2.0	2435	1569	1087	1414	1876	2812	4920	6544	6310	4712	...	6904	8449	10109	11100	11123	9474	8759	8357
3.0	3354	2142	1407	1467	1550	2387	4241	5663	5386	4657	...	7226	8850	10314	10491	11239	9599	9026	8531
4.0	2897	1688	1199	1424	1696	2581	4592	6029	5704	4744	...	7158	8515	9492	10357	10259	9097	8358	8649
5.0	2733	1541	1030	1253	1617	2900	4814	6261	6469	5530	...	6955	8312	9609	10699	10170	9430	9354	9610
6.0	4537	2864	1864	1555	1551	2162	3642	4766	4942	4401	...	7235	8612	9444	9929	9263	8405	8117	8567
7.0	3645	2296	1507	1597	1763	2422	4102	5575	5376	4639	...	7276	8474	10393	11013	10573	9472	8691	8525
8.0	2830	1646	1123	1483	1889	3224	5431	7361	7357	5703	...	7240	8775	9851	10673	9687	8796	8604	8367
9.0	2657	1724	1222	1480	1871	3168	5802	7592	7519	5895	...	7877	9220	10270	11910	11449	9804	8909	8665
10.0	3296	2126	1464	1434	1591	2594	4664	6046	6158	5072	...	7612	9578	11045	11875	10934	9613	9687	9240
11.0	3036	1665	1095	1424	1842	2520	4954	6876	6871	5396	...	7503	8920	10125	10898	10361	9327	8824	8730
12.0	3227	2147	1393	1362	1757	2710	4576	6250	6231	5177	...	7743	9390	10734	11713	12216	10393	9965	10310
13.0	5408	3509	2262	1832	1705	2327	4196	5685	6060	5631	...	8200	9264	10534	11826	11450	9921	8705	8423
14.0	3748	2349	1605	1656	1756	2629	4257	5781	5520	4824	...	6963	8192	9511	10115	9553	9146	9182	8589
15.0	2497	1515	1087	1381	1862	2980	5050	6837	6729	5201	...	7633	8505	10285	11959	11728	11032	10509	9105
16.0	2547	1585	1119	1395	1818	2966	5558	7517	7495	5958	...	7597	9290	10804	11773	10855	10924	10142	10374
17.0	3155	2048	1500	1488	1897	2741	4562	6315	5882	4934	...	7472	8997	10323	11236	11089	9919	9935	9823
18.0	3390	2135	1332	1626	1892	2959	4688	6618	6451	5377	...	7534	9040	10274	10692	10338	9551	9310	9285
19.0	3217	2188	1604	1675	1810	2639	4733	6159	6014	5006	...	7374	8898	9893	10741	10429	9701	10051	10049
20.0	4475	3190	2100	1858	1618	2143	3584	4900	5083	4765	...	7462	8630	9448	10046	9272	8592	8614	8703
21.0	4294	3194	1972	1727	1926	2615	4185	5727	5529	4707	...	7064	8127	9483	9817	9291	8317	8107	8245
22.0	2787	1637	1175	1468	1934	3151	5204	6872	6850	5198	...	7337	9148	10574	10962	9884	8980	8772	8430
23.0	2546	1580	1136	1429	1957	3132	5204	6890	6436	5177	...	7575	9309	9980	10341	10823	11347	11447	10347
24.0	3200	2055	1438	1493	1798	2754	4484	6013	5913	5146	...	7083	8706	10366	10786	9772	9080	9213	8831
25.0	2405	1499	1072	1439	1943	2973	5356	7627	7078	5994	...	7298	8732	9922	10504	10673	9048	8751	9508

Hour	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	...	14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0
Day																			
26.0	3810	3065	2046	1806	1730	2337	3776	5172	5071	4808	...	7269	8815	9885	10697	10867	10122	9820	10441
27.0	5196	3635	2352	2055	1723	2336	3539	4937	5053	4771	...	7519	8803	9793	9838	9228	8267	7908	8507
28.0	4123	2646	1843	1802	1883	2793	4290	5715	5671	5206	...	7341	8584	9671	9975	9132	8255	8309	7949
29.0	2678	1827	1409	1678	1948	3056	5213	6852	6695	5481	...	7630	9249	10105	11113	10411	9301	9270	9114
30.0	2401	1510	1112	1403	1841	3216	5757	7596	7611	6064	...	8396	10243	11554	12126	12561	11024	10836	10042
31.0	2174	1394	1087	919	773	997	1561	2169	2410	2525	...	4104	5099	5386	5308	5350	4898	4819	5064

In [84]:
```
pivot.style.background_gradient()
```

Out[84]:

Day \ Hour	0.000000	1.000000	2.000000	3.000000	4.000000	5.000000	6.000000	7.000000	8.000000	9.000000	10.0000
1.000000	3178	1944	1256	1308	1429	2126	3664	5380	5292	4617	46
2.000000	2435	1569	1087	1414	1876	2812	4920	6544	6310	4712	47
3.000000	3354	2142	1407	1467	1550	2387	4241	5663	5386	4657	47
4.000000	2897	1688	1199	1424	1696	2581	4592	6029	5704	4744	47
5.000000	2733	1541	1030	1253	1617	2900	4814	6261	6469	5530	51
6.000000	4537	2864	1864	1555	1551	2162	3642	4766	4942	4401	48
7.000000	3645	2296	1507	1597	1763	2422	4102	5575	5376	4639	49
8.000000	2830	1646	1123	1483	1889	3224	5431	7361	7357	5703	52
9.000000	2657	1724	1222	1480	1871	3168	5802	7592	7519	5895	54
10.000000	3296	2126	1464	1434	1591	2594	4664	6046	6158	5072	49
11.000000	3036	1665	1095	1424	1842	2520	4954	6876	6871	5396	52
12.000000	3227	2147	1393	1362	1757	2710	4576	6250	6231	5177	51
13.000000	5408	3509	2262	1832	1705	2327	4196	5685	6060	5631	54
14.000000	3748	2349	1605	1656	1756	2629	4257	5781	5520	4824	49
15.000000	2497	1515	1087	1381	1862	2980	5050	6837	6729	5201	53
16.000000	2547	1585	1119	1395	1818	2966	5558	7517	7495	5958	56
17.000000	3155	2048	1500	1488	1897	2741	4562	6315	5882	4934	50
18.000000	3390	2135	1332	1626	1892	2959	4688	6618	6451	5377	51
19.000000	3217	2188	1604	1675	1810	2639	4733	6159	6014	5006	50
20.000000	4475	3190	2100	1858	1618	2143	3584	4900	5083	4765	51
21.000000	4294	3194	1972	1727	1926	2615	4185	5727	5529	4707	49
22.000000	2787	1637	1175	1468	1934	3151	5204	6872	6850	5198	52
23.000000	2546	1580	1136	1429	1957	3132	5204	6890	6436	5177	50
24.000000	3200	2055	1438	1493	1798	2754	4484	6013	5913	5146	49
25.000000	2405	1499	1072	1439	1943	2973	5356	7627	7078	5994	54

Hour	0.000000	1.000000	2.000000	3.000000	4.000000	5.000000	6.000000	7.000000	8.000000	9.000000	10.0000
Day											
26.000000	3810	3065	2046	1806	1730	2337	3776	5172	5071	4808	50
27.000000	5196	3635	2352	2055	1723	2336	3539	4937	5053	4771	51
28.000000	4123	2646	1843	1802	1883	2793	4290	5715	5671	5206	52
29.000000	2678	1827	1409	1678	1948	3056	5213	6852	6695	5481	52
30.000000	2401	1510	1112	1403	1841	3216	5757	7596	7611	6064	59

In [89]:
```
# automating analysis
def gen_pivot_table(df, col1, col2):
    pivot = final.groupby([col1, col2]).size().unstack()
    return pivot.style.background_gradient()
```

In [90]:
```
final.columns
```

Out[90]:
```
Index(['Date/Time', 'Lat', 'Lon', 'Base', 'Dispatching_base_num',
       'Pickup_date', 'Affiliated_base_num', 'locationID', 'Day', 'Hour'],
      dtype='object')
```

In [91]:
```
gen_pivot_table(final, "Day", "Hour")
```

Out[91]:

Day \ Hour	0.000000	1.000000	2.000000	3.000000	4.000000	5.000000	6.000000	7.000000	8.000000	9.000000	10.0000
1.000000	3178	1944	1256	1308	1429	2126	3664	5380	5292	4617	46
2.000000	2435	1569	1087	1414	1876	2812	4920	6544	6310	4712	47
3.000000	3354	2142	1407	1467	1550	2387	4241	5663	5386	4657	47
4.000000	2897	1688	1199	1424	1696	2581	4592	6029	5704	4744	47
5.000000	2733	1541	1030	1253	1617	2900	4814	6261	6469	5530	51
6.000000	4537	2864	1864	1555	1551	2162	3642	4766	4942	4401	48
7.000000	3645	2296	1507	1597	1763	2422	4102	5575	5376	4639	49
8.000000	2830	1646	1123	1483	1889	3224	5431	7361	7357	5703	52
9.000000	2657	1724	1222	1480	1871	3168	5802	7592	7519	5895	54
10.000000	3296	2126	1464	1434	1591	2594	4664	6046	6158	5072	49
11.000000	3036	1665	1095	1424	1842	2520	4954	6876	6871	5396	52
12.000000	3227	2147	1393	1362	1757	2710	4576	6250	6231	5177	51
13.000000	5408	3509	2262	1832	1705	2327	4196	5685	6060	5631	54
14.000000	3748	2349	1605	1656	1756	2629	4257	5781	5520	4824	49
15.000000	2497	1515	1087	1381	1862	2980	5050	6837	6729	5201	53
16.000000	2547	1585	1119	1395	1818	2966	5558	7517	7495	5958	56
17.000000	3155	2048	1500	1488	1897	2741	4562	6315	5882	4934	50
18.000000	3390	2135	1332	1626	1892	2959	4688	6618	6451	5377	51
19.000000	3217	2188	1604	1675	1810	2639	4733	6159	6014	5006	50
20.000000	4475	3190	2100	1858	1618	2143	3584	4900	5083	4765	51
21.000000	4294	3194	1972	1727	1926	2615	4185	5727	5529	4707	49
22.000000	2787	1637	1175	1468	1934	3151	5204	6872	6850	5198	52
23.000000	2546	1580	1136	1429	1957	3132	5204	6890	6436	5177	50
24.000000	3200	2055	1438	1493	1798	2754	4484	6013	5913	5146	49
25.000000	2405	1499	1072	1439	1943	2973	5356	7627	7078	5994	54

Hour	0.000000	1.000000	2.000000	3.000000	4.000000	5.000000	6.000000	7.000000	8.000000	9.000000	10.0000
Day											
26.000000	3810	3065	2046	1806	1730	2337	3776	5172	5071	4808	50
27.000000	5196	3635	2352	2055	1723	2336	3539	4937	5053	4771	51
28.000000	4123	2646	1843	1802	1883	2793	4290	5715	5671	5206	52
29.000000	2678	1827	1409	1678	1948	3056	5213	6852	6695	5481	52
30.000000	2401	1510	1112	1403	1841	3216	5757	7596	7611	6064	59