In [139…
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
uber_15=pd.read_csv(r'C:\Users\Admin\OneDrive\Desktop\data analysis\uber-pickups-in-new-york-city/uber-raw-data-janjune-15.csv',encoding='utf-8')
```

In [140…
```python
uber_15.head(2)
```

Out[140]:

	Dispatching_base_num	Pickup_date	Affiliated_base_num	locationID
0	B02617	2015-05-17 09:47:00	B02617	141
1	B02617	2015-05-17 09:47:00	B02617	65

In [141…
```python
uber_15.shape
```

Out[141]:
```
(14270479, 4)
```

In [142…
```python
uber_15.duplicated().sum()
```

Out[142]:
```
898225
```

In [143…
```python
uber_15.drop_duplicates(inplace=True)
```

In [144…
```python
uber_15.shape
```

Out[144]:
```
(13372254, 4)
```

In [145…
```python
uber_15.dtypes
```

Out[145]:
```
Dispatching_base_num    object
Pickup_date             object
Affiliated_base_num     object
locationID               int64
dtype: object
```

In [146…
```python
uber_15['Pickup_date']=pd.to_datetime(uber_15['Pickup_date'],format='%Y-%m-%d %H:%M:%S')
```

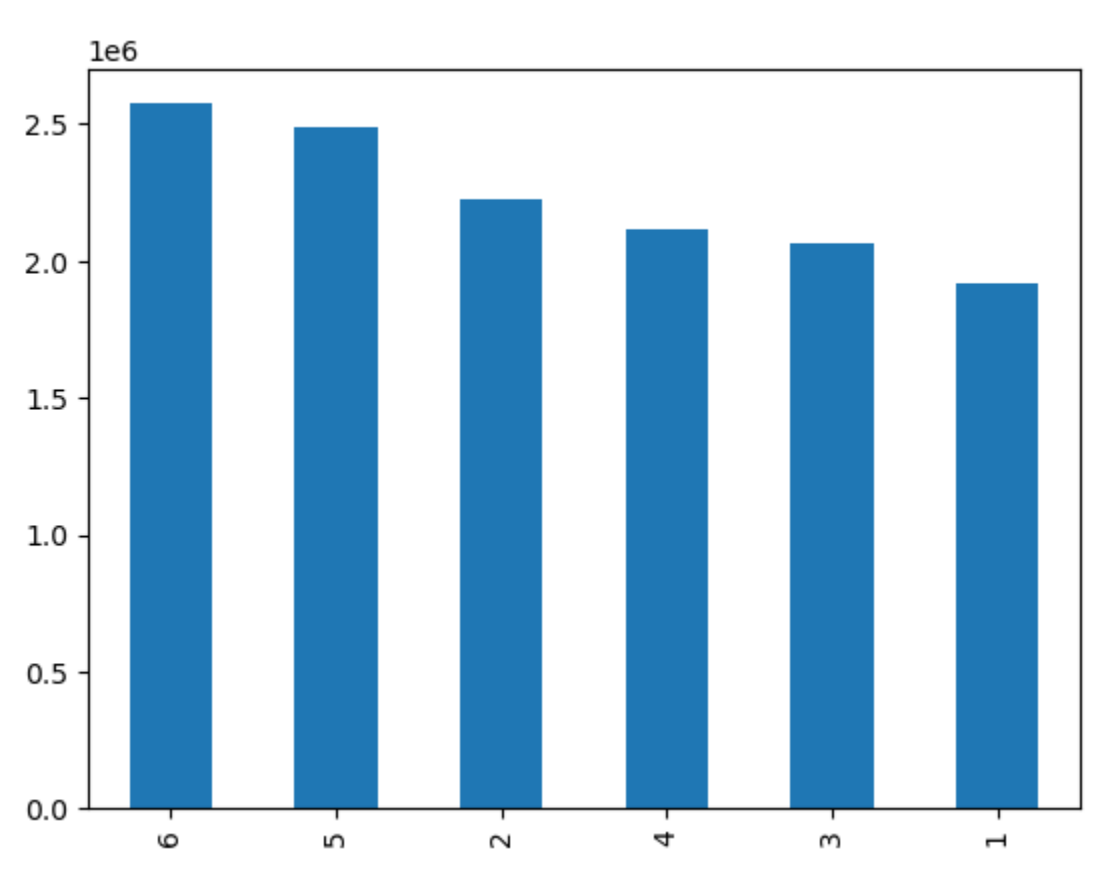
In [147…
```python
uber_15['Pickup_date'].dtype
```

Out[147]:
```
dtype('<M8[ns]')
```

In [148…
```python
uber_15['month']=uber_15['Pickup_date'].dt.month
```

In [149…
```python
uber_15['month'].value_counts().plot(kind='bar')
```

Out[149]:
```
<AxesSubplot:>
```



In [150…
```python
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['month']=uber_15['Pickup_date'].dt.month
uber_15['minute']=uber_15['Pickup_date'].dt.minute
uber_15.head(2)
```

Out[150]:

	Dispatching_base_num	Pickup_date	Affiliated_base_num	locationID	month	weekday	day	hour	minute
0	B02617	2015-05-17 09:47:00	B02617	141	5	Sunday	17	9	47
1	B02617	2015-05-17 09:47:00	B02617	65	5	Sunday	17	9	47

In [151…
```python
uber_15.groupby(['month','weekday']).size()
```

Out[151]:
```
month  weekday
1      Friday       339285
       Monday       190606
       Saturday     386049
       Sunday       230487
       Thursday     330319
       Tuesday      196574
       Wednesday    245650
2      Friday       373550
       Monday       274948
       Saturday     368311
       Sunday       296130
       Thursday     335603
       Tuesday      287260
       Wednesday    286387
3      Friday       309631
       Monday       269931
       Saturday     314785
       Sunday       313865
       Thursday     277026
       Tuesday      320634
       Wednesday    256767
4      Friday       315002
       Monday       238429
       Saturday     324545
       Sunday       273560
       Thursday     372522
       Tuesday      250632
       Wednesday    338015
5      Friday       430134
       Monday       255501
       Saturday     464298
       Sunday       390391
       Thursday     337607
       Tuesday      290004
       Wednesday    316045
6      Friday       371225
       Monday       375312
       Saturday     399377
       Sunday       334434
       Thursday     357782
       Tuesday      405500
       Wednesday    328141
dtype: int64
```

In [152…
```python
temp=uber_15.groupby(['month','weekday'],as_index=False).size()
temp['month'].unique()
```

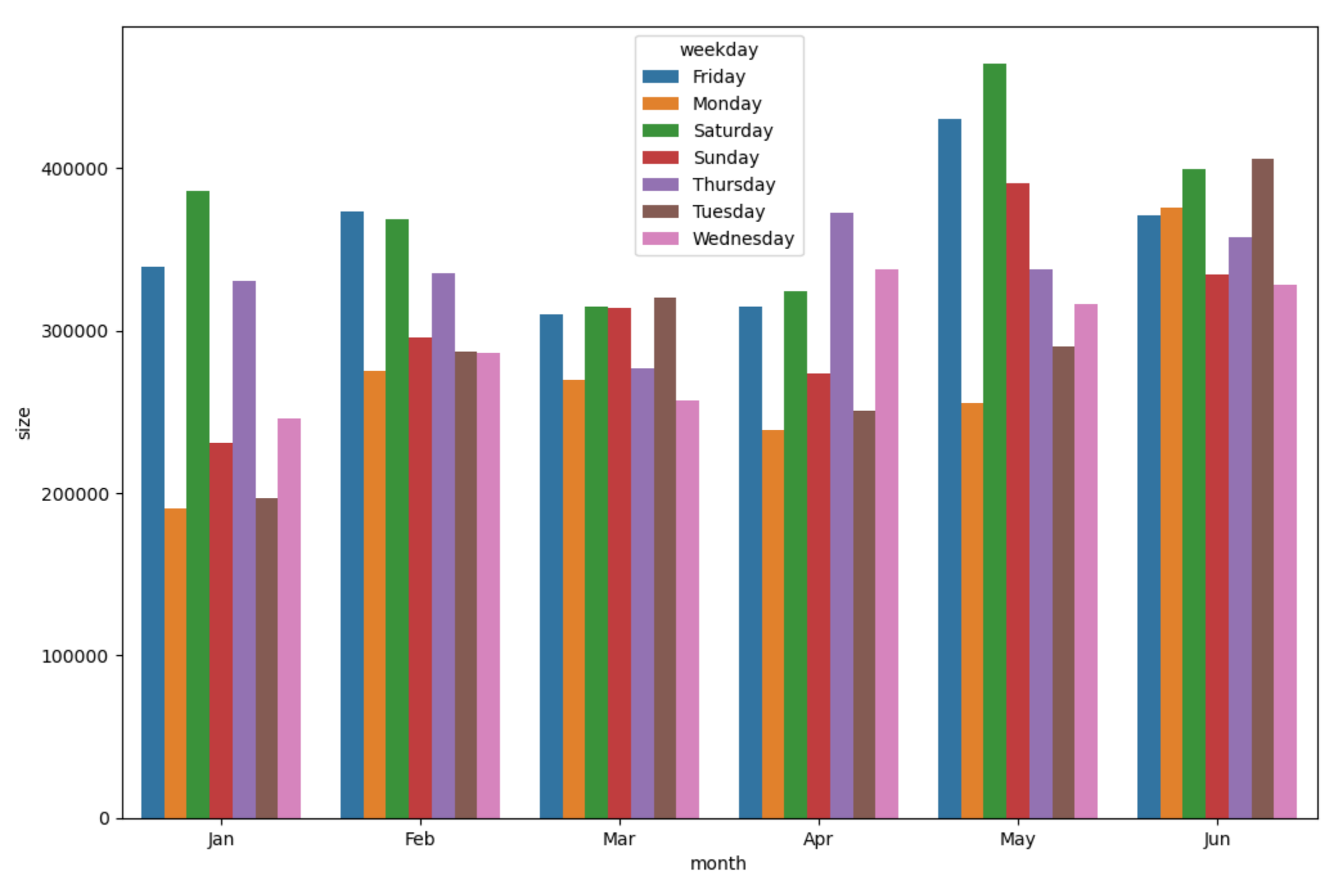
Out[152]:
```
array([1, 2, 3, 4, 5, 6], dtype=int64)
```

In [153…
```python
dict_month={1:'Jan', 2:'Feb', 3:'Mar', 4:'Apr', 5:'May', 6:'Jun'}
temp['month']=temp['month'].map(dict_month)
temp['month']
```

Out[153]:
```
0     Jan
1     Jan
2     Jan
3     Jan
4     Jan
5     Jan
6     Jan
7     Feb
8     Feb
9     Feb
10    Feb
11    Feb
12    Feb
13    Feb
14    Mar
15    Mar
16    Mar
17    Mar
18    Mar
19    Mar
20    Mar
21    Apr
22    Apr
23    Apr
24    Apr
25    Apr
26    Apr
27    Apr
28    May
29    May
30    May
31    May
32    May
33    May
34    May
35    Jun
36    Jun
37    Jun
38    Jun
39    Jun
40    Jun
41    Jun
Name: month, dtype: object
```

In [154…
```python
plt.figure(figsize=(12,8))
sns.barplot(x='month',y='size',hue='weekday',data=temp)
```

Out[154]:
```
<AxesSubplot:xlabel='month', ylabel='size'>
```



In [155…
```python
summary=uber_15.groupby(['weekday','hour'],as_index=False).size()
```

In [156…
```python
summary
```

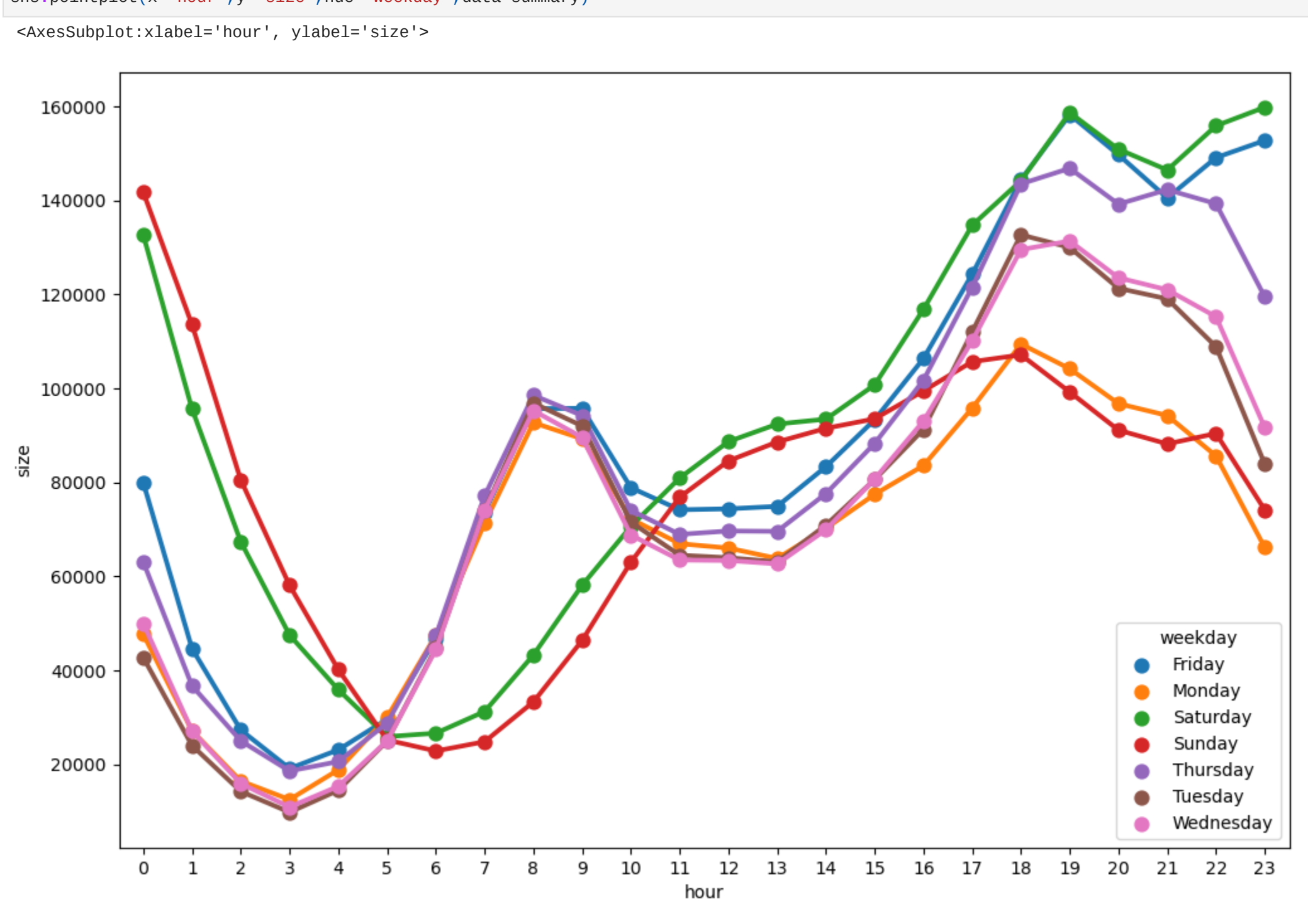
Out[156]:

	weekday	hour	size
0	Friday	0	79879
1	Friday	1	44563
2	Friday	2	27252
3	Friday	3	19076
4	Friday	4	23049
...	...	...	...
163	Wednesday	19	131317
164	Wednesday	20	123490
165	Wednesday	21	120941
166	Wednesday	22	115208
167	Wednesday	23	91631

168 rows × 3 columns

In [157…
```python
plt.figure(figsize=(12,8))
sns.pointplot(x='hour',y='size',hue='weekday',data=summary)
```

Out[157]:
```
<AxesSubplot:xlabel='hour', ylabel='size'>
```



In [158…
```python
uber_foil=pd.read_csv(r'C:\Users\Admin\OneDrive\Desktop\data analysis\uber-pickups-in-new-york-city/Uber-Jan-Feb-FOIL.csv')
```

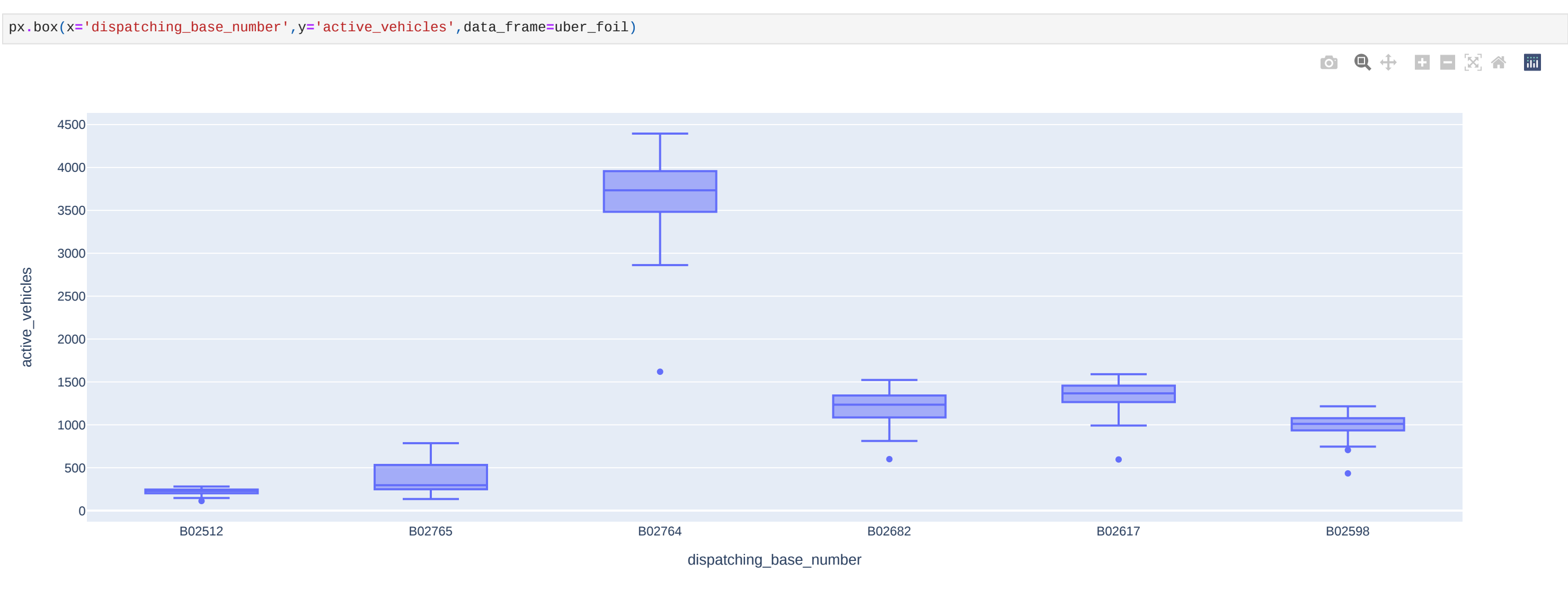
In [159…
```python
uber_foil.head()
```

Out[159]:

	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
3	B02682	1/1/2015	945	7679
4	B02617	1/1/2015	1228	9537

In [160…
```python
import chart_studio.plotly as py
import plotly.graph_objs as go
import plotly.express as px
from plotly.offline import download_plotlyjs, plot, iplot, init_notebook_mode
init_notebook_mode(connected=True)
```

In [161…
```python
px.box(x='dispatching_base_number',y='active_vehicles',data_frame=uber_foil)
```



In [162…
```python
import os
files=os.listdir(r'C:\Users\Admin\OneDrive\Desktop\data analysis\uber-pickups-in-new-york-city')[-7:]
```

In [163…
```python
files
```

Out[163]:
```
['uber-raw-data-apr14.csv',
 'uber-raw-data-aug14.csv',
 'uber-raw-data-janjune-15.csv',
 'uber-raw-data-jul14.csv',
 'uber-raw-data-jun14.csv',
 'uber-raw-data-may14.csv',
 'uber-raw-data-sep14.csv']
```

In [164…
```python
files.remove('uber-raw-data-janjune-15.csv')
```

In [165…
```python
files
```

Out[165]:
```
['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']
```

In [166…
```python
path=r'C:\Users\Admin\OneDrive\Desktop\data analysis\uber-pickups-in-new-york-city'

final=pd.DataFrame()
for file in files:
        current_df=pd.read_csv(path+'/'+file,encoding='utf-8')
        final=pd.concat([current_df,final])
```

In [167…
```python
final.shape
```

Out[167]:
```
(4534327, 4)
```

In [168…
```python
final.head(5)
```

Out[168]:

	Date/Time	Lat	Lon	Base
0	9/1/2014 0:01:00	40.2201	-74.0021	B02512
1	9/1/2014 0:01:00	40.7500	-74.0027	B02512
2	9/1/2014 0:03:00	40.7559	-73.9864	B02512
3	9/1/2014 0:06:00	40.7450	-73.9889	B02512
4	9/1/2014 0:11:00	40.8145	-73.9444	B02512

In [169…
```python
final.duplicated().sum()
```

Out[169]:
```
82581
```

In [170…
```python
final.drop_duplicates(inplace=True)
```

In [171…
```python
final.shape
```

Out[171]:
```
(4451746, 4)
```

In [172…
```python
rush_uber=final.groupby(['Lat','Lon'],as_index=False).size()
```

In [173…
```python
rush_uber
```

Out[173]:

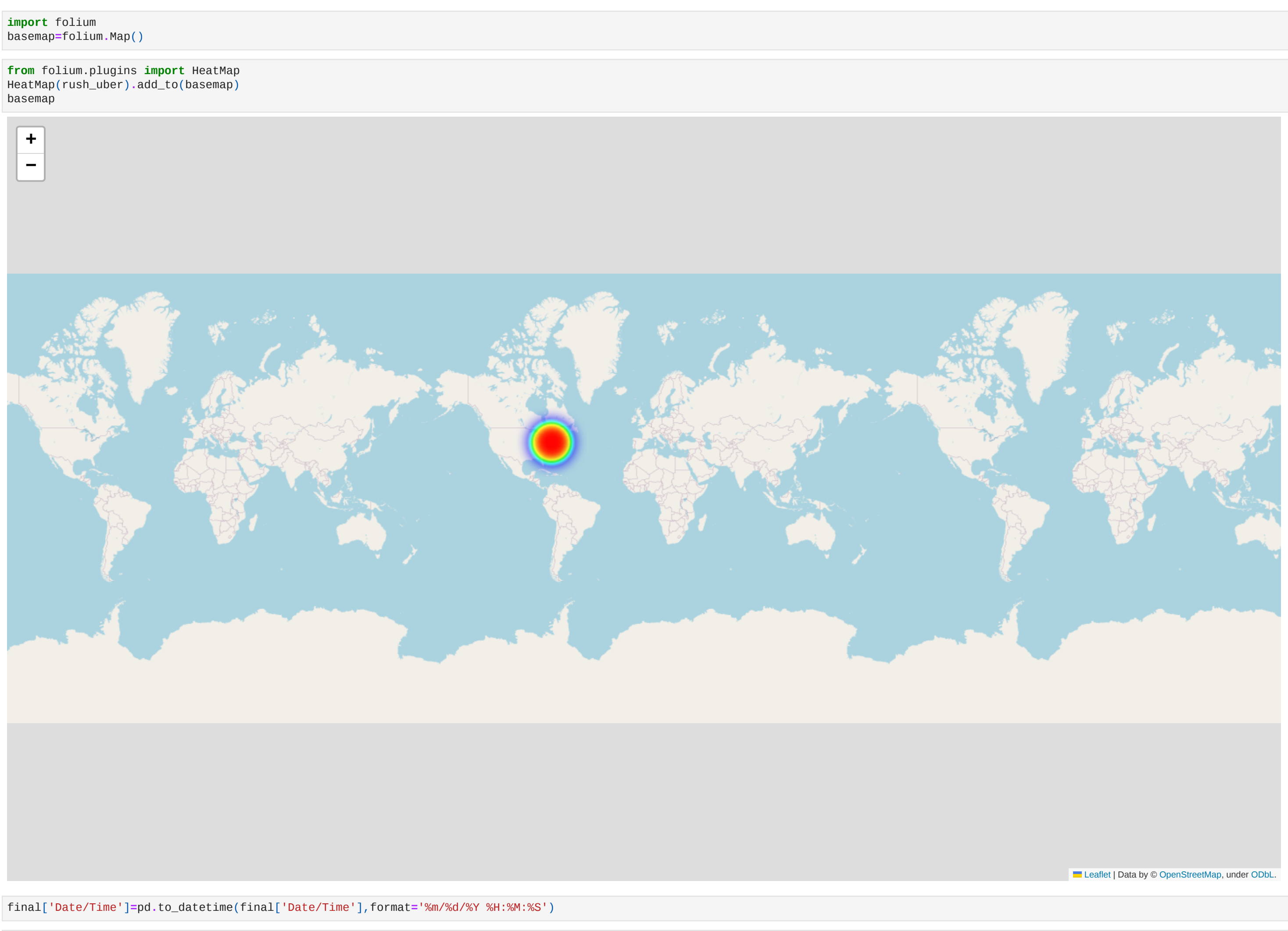
	Lat	Lon	size
0	39.6569	-74.2258	1
1	39.6686	-74.1607	1
2	39.7214	-74.2446	1
3	39.8416	-74.1512	1
4	39.9055	-74.0791	1
...	...	...	...
574553	41.3730	-72.9237	1
574554	41.3737	-73.7988	1
574555	41.5016	-72.8987	1
574556	41.5276	-72.7734	1
574557	42.1166	-72.0666	1

574558 rows × 3 columns

In [174…
```python
import folium
basemap=folium.Map()
```

In [175…
```python
from folium.plugins import HeatMap
HeatMap(rush_uber).add_to(basemap)
basemap
```

Out[175]:



In [176…
```python
final['Date/Time']=pd.to_datetime(final['Date/Time'],format='%m/%d/%Y %H:%M:%S')
```

In [177…
```python
final.head(4)
```

Out[177]:

	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

In [178…
```python
final['weekday']=final['Date/Time'].dt.day
final['hour']=final['Date/Time'].dt.hour
```

In [179…
```python
final.head(4)
```

Out[179]:

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

In [180…
```python
pivot=final.groupby(['weekday','hour']).size().unstack()
```

In [181…
```python
pivot.style.background_gradient()
```

Out[181]:

hour weekday	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	3178	1944	1256	1308	1429	2126	3664	5380	5292	4617	4607	4729	4930	5794	6933	7910	8633	9511	8604	8001	7315	7803	6268	4050
2	2435	1569	1087	1414	1876	2812	4920	6544	6310	4712	4797	4975	5188	5695	6904	8449	10109	11100	11123	9474	8759	8357	6998	5160
3	3354	2142	1407	1467	1550	2387	4241	5663	5386	4657	4788	5065	5384	6093	7226	8850	10314	10491	11239	9599	9026	8531	7142	4686
4	2897	1688	1199	1424	1696	2581	4592	6029	5704	4744	4743	4975	5193	6175	7158	8515	9492	10357	10259	9097	8358	8649	7706	5130
5	2733	1541	1030	1253	1617	2900	4814	6261	6469	5530	5141	5011	5047	5690	6955	8312	9609	10699	10170	9430	9354	9610	8853	6518
6	4537	2864	1864	1555	1551	2162	3642	4766	4942	4401	4801	5174	5426	6258	7235	8612	9444	9929	9263	8405	8117	8567	7852	5946
7	3645	2296	1507	1597	1763	2422	4102	5575	5376	4639	4905	5166	5364	6214	7276	8474	10393	11013	10573	9472	8691	8525	7194	4801
8	2830	1646	1123	1483	1889	3224	5431	7361	7357	5703	5288	5350	5483	6318	7240	8775	9851	10673	9687	8796	8604	8367	6795	4256
9	2657	1724	1222	1480	1871	3168	5802	7592	7519	5895	5406	5443	5496	6419	7877	9220	10270	11910	11449	9804	8909	8665	7499	5203
10	3296	2126	1464	1434	1591	2594	4664	6046	6158	5072	4976	5415	5506	6527	7612	9578	11045	11875	10934	9613	9687	9240	7766	5496
11	3036	1665	1095	1424	1842	2520	4954	6876	6871	5396	5215	5423	5513	6486	7503	8920	10125	10898	10361	9327	8824	8730	7771	5360
12	3227	2147	1393	1362	1757	2710	4576	6250	6231	5177	5157	5319	5570	6448	7743	9390	10734	11713	12216	10393	9965	10310	9992	7945
13	5408	3509	2262	1832	1705	2327	4196	5685	6060	5631	5442	5720	5914	6678	8200	9264	10534	11826	11450	9921	8705	8423	7363	5936
14	3748	2349	1605	1656	1756	2629	4257	5781	5520	4824	4911	5118	5153	5747	6963	8192	9511	10115	9553	9146	9182	8589	6891	4460
15	2497	1515	1087	1381	1862	2980	5050	6837	6729	5201	5347	5517	5503	6997	7633	8505	10285	11959	11728	11032	10509	9105	7153	4480
16	2547	1585	1119	1395	1818	2966	5558	7517	7495	5958	5626	5480	5525	6198	7597	9290	10804	11773	10855	10924	10142	10374	8094	5380
17	3155	2048	1500	1488	1897	2741	4562	6315	5882	4934	5004	5306	5634	6507	7472	8997	10323	11236	11089	9919	9935	9823	8362	5699
18	3390	2135	1332	1626	1892	2959	4688	6618	6451	5377	5150	5487	5490	6383	7534	9040	10274	10692	10338	9551	9310	9285	8015	5492
19	3217	2188	1604	1675	1810	2639	4733	6159	6014	5006	5092	5240	5590	6367	7374	8898	9893	10741	10429	9701	10051	10049	9090	6666
20	4475	3190	2100	1858	1618	2143	3584	4900	5083	4765	5135	5650	5745	6656	7462	8630	9448	10046	9272	8592	8614	8703	7787	5907
21	4294	3194	1972	1727	1926	2615	4185	5727	5529	4707	4911	5212	5465	6085	7064	8127	9483	9817	9291	8317	8107	8245	7362	5231
22	2787	1637	1175	1468	1934	3151	5204	6872	6850	5198	5277	5352	5512	6342	7337	9148	10574	10962	9884	8980	8772	8430	6784	4530
23	2546	1580	1136	1429	1957	3132	5204	6890	6436	5177	5066	5304	5504	6232	7575	9309	9980	10341	10823	11347	11447	10347	8637	5577
24	3200	2055	1438	1493	1798	2754	4484	6013	5913	5146	4947	5311	5229	5974	7083	8706	10366	10786	9772	9080	9213	8831	7480	4456
25	2405	1499	1072	1439	1943	2973	5356	7627	7078	5994	5432	5504	5694	6204	7298	8732	9922	10504	10673	9048	8751	9508	8522	6605
26	3810	3065	2046	1806	1730	2337	3776	5172	5071	4808	5061	5179	5381	6166	7269	8815	9885	10697	10867	10122	9820	10441	9486	7593
27	5196	3635	2352	2055	1723	2336	3539	4937	5053	4771	5198	5732	5839	6820	7519	8803	9793	9838	9228	8267	7908	8507	7720	6046
28	4123	2646	1843	1802	1883	2793	4290	5715	5671	5206	5247	5500	5486	6120	7341	8584	9671	9975	9132	8255	8309	7949	6411	4461
29	2678	1827	1409	1678	1948	3056	5213	6852	6695	5481	5234	5163	5220	6305	7630	9249	10105	11113	10411	9301	9270	9114	6992	4323
30	2401	1510	1112	1403	1841	3216	5757	7596	7611	6064	5987	6090	6423	7249	8396	10243	11554	12126	12561	11024	10836	10042	8275	4723
31	2174	1394	1087	919	773	997	1561	2169	2410	2525	2564	2777	2954	3280	4104	5099	5386	5308	5350	4898	4819	5064	5164	3961

In [ ]: