In [8]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [9]:

```
import os
```

In [3]:

```
os.listdir(r"C:\Users\jopau\Documents\Courses\Python Data Analysis\Uber Analysis\Datasets")
```

Out[3]:

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

In [45]:

```
uber = pd.read_csv(r"C:\Users\jopau\Documents\Courses\Python Data Analysis\Uber Analysis\Datasets\uber-raw-data
```

In [47]:

```
uber.shape
```

Out[47]:

```
(100000, 4)
```

In [44]:

```
uber
```

Out[44]:

	Dispatching_base_num	Pickup_date	Affiliated_base_num	locationID	Month	Day_name	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
4	B02764	2015-03-23 07:03:00	B00111	140	March	Monday	23	7	3
...	...	...	...	...	...	...	...	...	...
99995	B02764	2015-04-13 16:12:00	B02764	234	April	Monday	13	16	12
99996	B02764	2015-03-06 21:32:00	B02764	24	March	Friday	6	21	32
99997	B02598	2015-03-19 19:56:00	B02598	17	March	Thursday	19	19	56
99998	B02682	2015-05-02 16:02:00	B02682	68	May	Saturday	2	16	2
99999	B02764	2015-06-24 16:04:00	B02764	125	June	Wednesday	24	16	4

99946 rows × 9 columns

In [11]:

```
type(uber)
```

Out[11]:

```
pandas.core.frame.DataFrame
```

In [ ]:

```

```

In [12]:

```
uber.duplicated()
```

Out[12]:

```
0        False
1        False
2        False
3        False
4        False
         ...  
99995    False
99996    False
99997    False
99998    False
99999    False
Length: 100000, dtype: bool
```

In [13]:

```
uber.duplicated().sum()
```

```
Out[13]: 54
```

```
In [14]: uber.drop_duplicates(inplace=True)
```

```
In [15]: uber.duplicated().sum()

Out[15]: 0
```

```
In [16]: uber.shape

Out[16]: (99946, 4)
```

```
In [17]: uber.dtypes

Out[17]: Dispatching_base_num    object
         Pickup_date             object
         Affiliated_base_num     object
         locationID               int64
         dtype: object
```

```
In [18]: uber.isnull()
```

Out[18]:

	Dispatching_base_num	Pickup_date	Affiliated_base_num	locationID
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
...	...	...	...	...
99995	False	False	False	False
99996	False	False	False	False
99997	False	False	False	False
99998	False	False	False	False
99999	False	False	False	False

99946 rows × 4 columns

```
In [19]: uber.isnull().sum()

Out[19]: Dispatching_base_num       0
         Pickup_date                0
         Affiliated_base_num     1116
         locationID                 0
         dtype: int64
```

```
In [20]: type(uber['Pickup_date'][0])

Out[20]: str
```

```
In [21]: uber['Pickup_date'] = pd.to_datetime(uber['Pickup_date'])
```

```
In [22]: uber['Pickup_date'].dtypes

Out[22]: dtype('<M8[ns]')
```

```
In [23]: uber['Pickup_date'][0]

Out[23]: Timestamp('2015-05-02 21:43:00')
```

```
In [24]: uber.dtypes

Out[24]: Dispatching_base_num            object
         Pickup_date             datetime64[ns]
         Affiliated_base_num             object
         locationID                       int64
         dtype: object
```

```
In [ ]:
```

```
In [ ]:
```

```
In [25]: uber['Pickup_date'].dt.month_name()
```

```
Out[25]: 0            May
         1        January
         2          March
         3          April
         4          March
                    ...
         99995      April
         99996      March
         99997      March
         99998        May
         99999       June
         Name: Pickup_date, Length: 99946, dtype: object
```

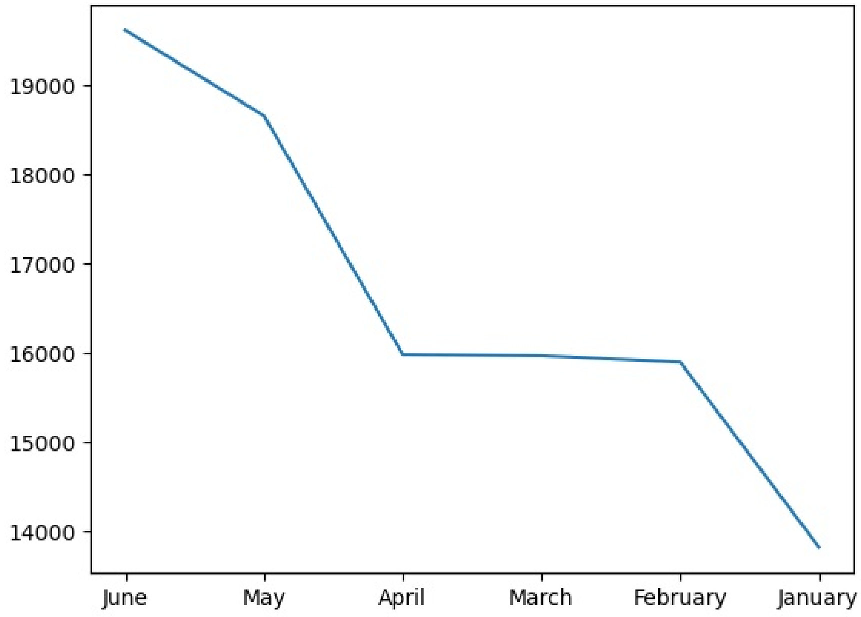
```
In [26]: uber['Month'] = uber['Pickup_date'].dt.month_name()
```

```
In [27]: uber['Month'].value_counts()
```

```
Out[27]: June        19620
         May         18660
         April       15982
         March       15969
         February    15896
         January     13819
         Name: Month, dtype: int64
```

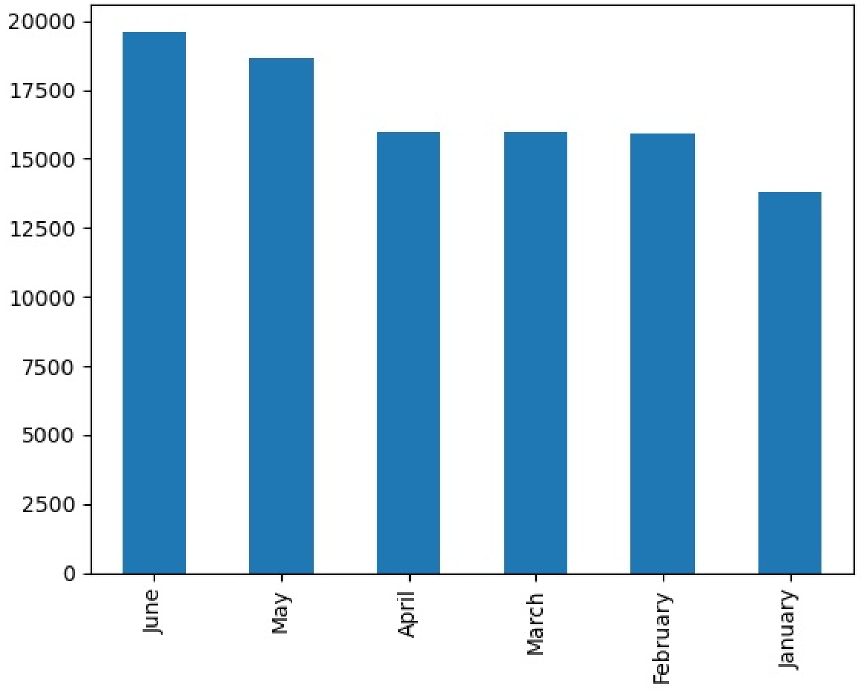
```
In [28]: uber['Month'].value_counts().plot()
```

```
Out[28]: <Axes: >
```



```
In [29]: uber['Month'].value_counts().plot(kind = 'bar')
```

```
Out[29]: <Axes: >
```



In [30]:

```
uber['Day_name'] = uber['Pickup_date'].dt.day_name()
```

In [31]:

```
uber['Day_name']
```

Out[31]:

```
0         Saturday
1          Tuesday
2         Thursday
3           Friday
4           Monday
           ...    
99995       Monday
99996       Friday
99997     Thursday
99998     Saturday
99999    Wednesday
Name: Day_name, Length: 99946, dtype: object
```

In [32]:

```
uber['Day'] = uber['Pickup_date'].dt.day
uber['Hour'] = uber['Pickup_date'].dt.hour
uber['Minute'] = uber['Pickup_date'].dt.minute
```

In [33]:

```
uber['Day']
```

Out[33]:

```
0         2
1        20
2        19
3        10
4        23
         ..
99995    13
99996     6
99997    19
99998     2
99999    24
Name: Day, Length: 99946, dtype: int64
```

In [34]:

```
uber.head(5)
```

Out[34]:

	Dispatching_base_num	Pickup_date	Affiliated_base_num	locationID	Month	Day_name	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
4	B02764	2015-03-23 07:03:00	B00111	140	March	Monday	23	7	3

In [35]:

```
pivot = pd.crosstab(uber['Month'], uber['Day_name'])
pivot
```

Out[35]:

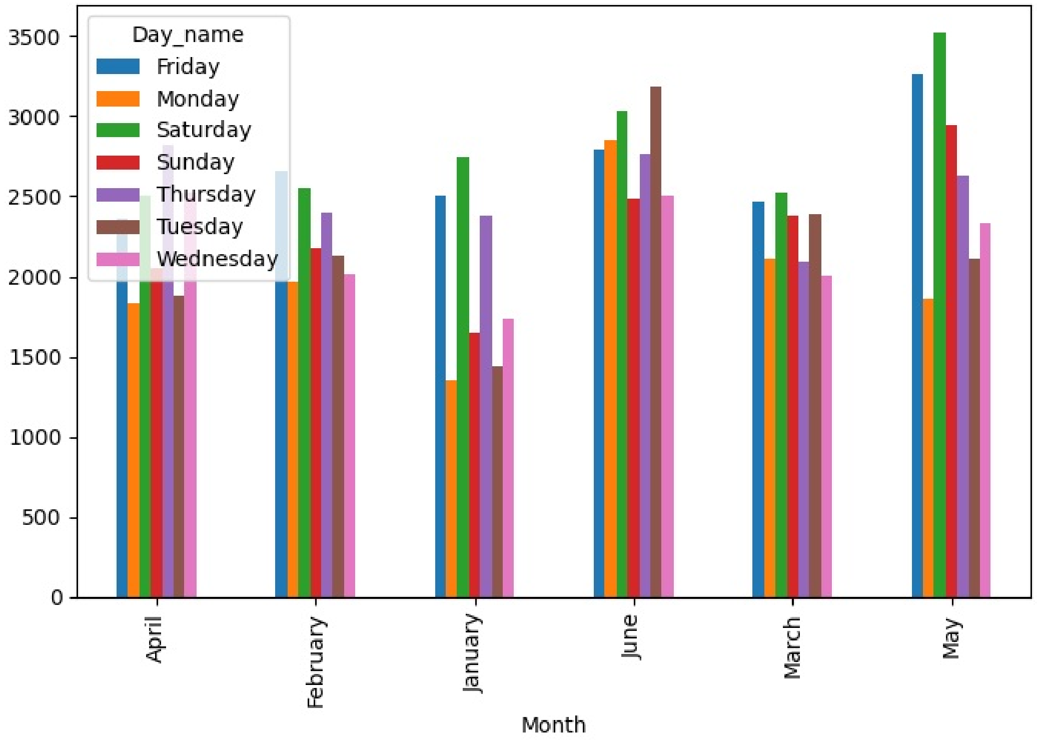
Day_name	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
Month							
April	2365	1833	2508	2052	2823	1880	2521
February	2655	1970	2550	2183	2396	2129	2013
January	2508	1353	2745	1651	2378	1444	1740
June	2793	2848	3037	2485	2767	3187	2503
March	2465	2115	2522	2379	2093	2388	2007
May	3262	1865	3519	2944	2627	2115	2328

In [36]:

```
pivot.plot(kind = 'bar', figsize=(8,5))
```

Out[36]:

```
<Axes: xlabel='Month'>
```



In [ ]:

```
In [37]: summary = uber.groupby(['Day_name', 'Hour'], as_index=False).size()
```

```
In [38]: summary
```

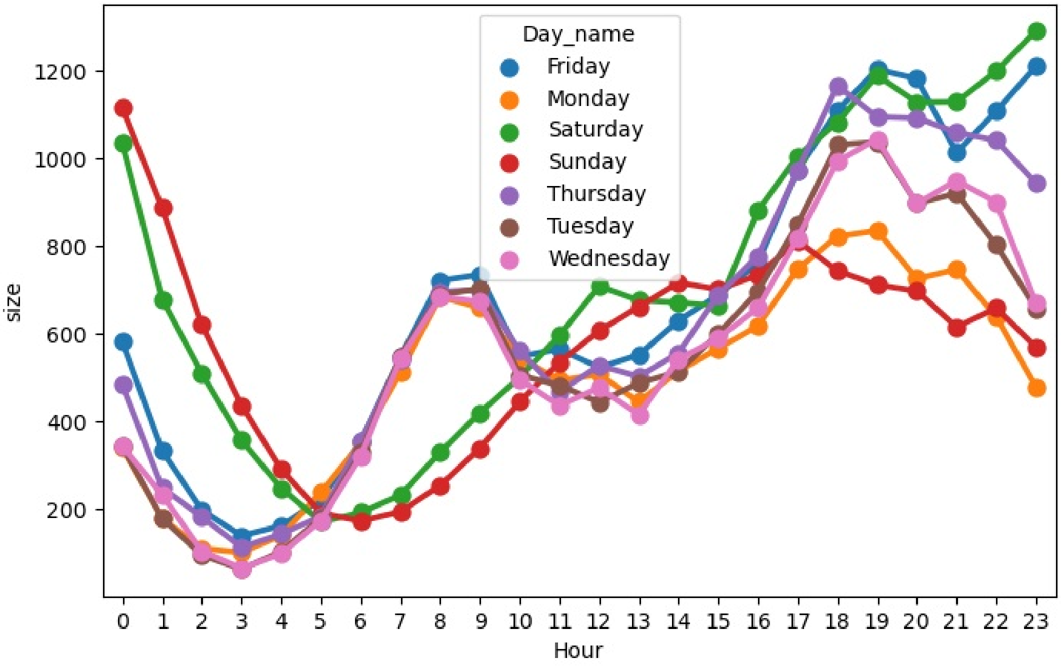
Out[38]:

	Day_name	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 [39]: plt.figure(figsize=(8,5))
         sns.pointplot(x='Hour', y='size', hue='Day_name', data=summary)
```

```
Out[39]: <Axes: xlabel='Hour', ylabel='size'>
```



```
In [ ]:
```

```
In [40]: uber.columns
```

```
Out[40]: Index(['Dispatching_base_num', 'Pickup_date', 'Affiliated_base_num',
                'locationID', 'Month', 'Day_name', 'Day', 'Hour', 'Minute'],
               dtype='object')
```

```
In [8]: uber_foil = pd.read_csv(r"C:\Users\jopau\Documents\Courses\Python Data Analysis\Uber Analysis\Datasets\Uber-Jan
```

```
In [5]: uber_foil.shape
```

```
Out[5]: (354, 4)
```

```
In [6]: uber_foil.head(5)
```

Out[6]:

	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 [7]: !pip install chart_studio
        !pip install plotly
```

```
Requirement already satisfied: chart_studio in c:\users\jopau\anaconda3\lib\site-packages (1.1.0)
Requirement already satisfied: plotly in c:\users\jopau\anaconda3\lib\site-packages (from chart_studio) (5.9.0)
Requirement already satisfied: requests in c:\users\jopau\anaconda3\lib\site-packages (from chart_studio) (2.29
.0)
Requirement already satisfied: retrying>=1.3.3 in c:\users\jopau\anaconda3\lib\site-packages (from chart_studio
) (1.3.4)
Requirement already satisfied: six in c:\users\jopau\appdata\roaming\python\python311\site-packages (from chart
_studio) (1.16.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\jopau\anaconda3\lib\site-packages (from plotly->char
t_studio) (8.2.2)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\jopau\anaconda3\lib\site-packages (from req
uests->chart_studio) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\jopau\anaconda3\lib\site-packages (from requests->chart
_studio) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\jopau\anaconda3\lib\site-packages (from reques
ts->chart_studio) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\jopau\anaconda3\lib\site-packages (from requests-
>chart_studio) (2023.11.17)
^C
```

```
In [1]: 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 [3]:

```
init_notebook_mode(connected=True)
```

In [9]:

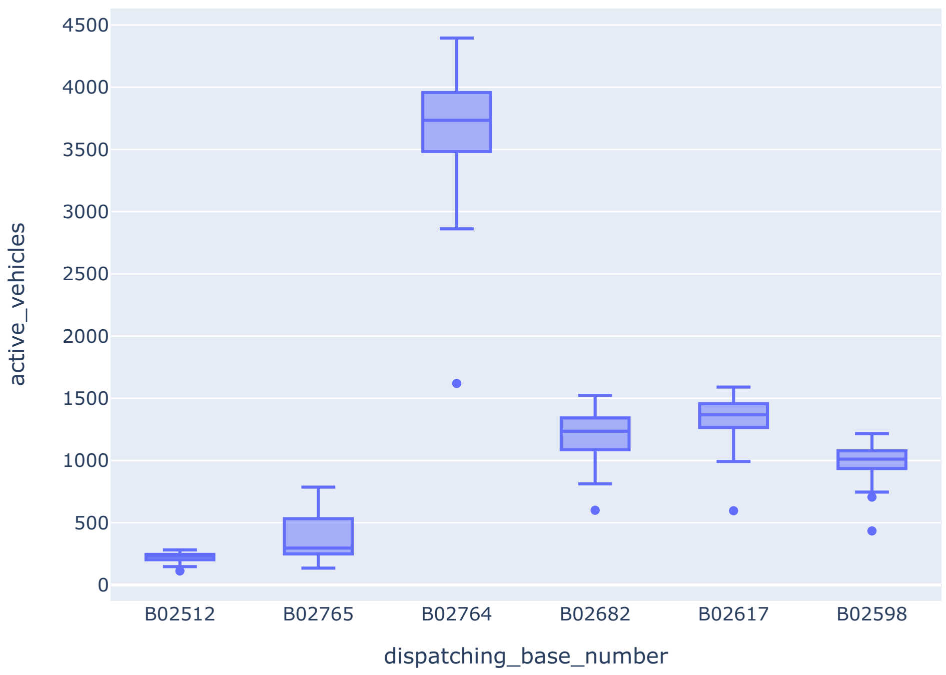
```
uber_foil.columns
```

Out[9]:

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

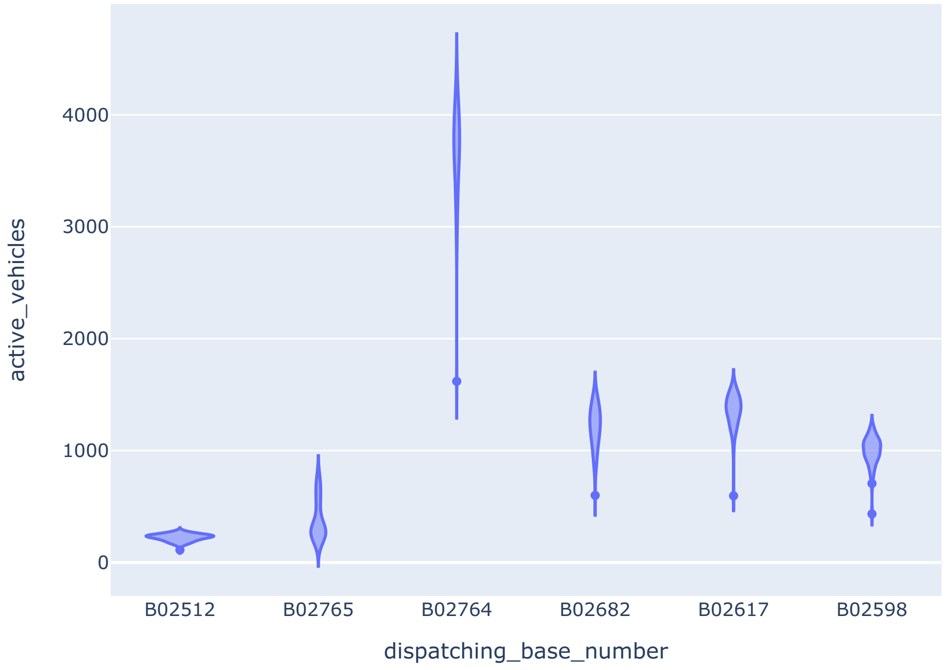
In [11]:

```
px.box(x='dispatching_base_number' , y='active_vehicles' , data_frame=uber_foil)
```



In [12]:

```
px.violin(x='dispatching_base_number' , y='active_vehicles' , data_frame=uber_foil)
```



In [ ]:

In [14]:

```
os.listdir(r"C:\Users\jopau\Documents\Courses\Python Data Analysis\Uber Analysis\Datasets")
```

```
Out[14]: ['other-American_B01362.csv',
          'other-Carmel_B00256.csv',
          'other-Dial7_B00887.csv',
          'other-Diplo_B01196.csv',
          'other-Federal_02216.csv',
          'other-FHV-services_jan-aug-2015.csv',
          'other-Firstclass_B01536.csv',
          'other-Highclass_B01717.csv',
          'other-Lyft_B02510.csv',
          'other-Prestige_B01338.csv',
          'other-Skyline_B00111.csv',
          'Uber-Jan-Feb-FOIL.csv',
          'uber-raw-data-apr14.csv',
          'uber-raw-data-aug14.csv',
          'uber-raw-data-janjune-15.csv',
          'uber-raw-data-janjune-15_sample.csv',
          'uber-raw-data-jul14.csv',
          'uber-raw-data-jun14.csv',
          'uber-raw-data-may14.csv',
          'uber-raw-data-sep14.csv']
```

```
In [16]: files = os.listdir(r"C:\Users\jopau\Documents\Courses\Python Data Analysis\Uber Analysis\Datasets")[-8:]
```

```
In [18]: files
```

```
Out[18]: ['uber-raw-data-apr14.csv',
          'uber-raw-data-aug14.csv',
          'uber-raw-data-janjune-15.csv',
          'uber-raw-data-janjune-15_sample.csv',
          'uber-raw-data-jul14.csv',
          'uber-raw-data-jun14.csv',
          'uber-raw-data-may14.csv',
          'uber-raw-data-sep14.csv']
```

```
In [ ]:
```

```
In [13]: final = pd.DataFrame()
```

```
In [14]: path = r"C:\Users\jopau\Documents\Courses\Python Data Analysis\Uber Analysis\Datasets"
```

```
In [18]: for file in files:
             current_df = pd.read_csv(path+'/'+file)
             final = pd.concat([current_df, final])
```

```
In [19]: final.shape
```

```
Out[19]: (37809612, 8)
```

```
In [20]: final.duplicated().sum()
```

```
Out[20]: 19985612
```

```
In [21]: final.drop_duplicates(inplace = True)
```

```
In [22]: final.shape
```

```
Out[22]: (17824000, 8)
```

```
In [23]: final.head(5)
```

Out[23]:

	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
3	9/1/2014 0:06:00	40.7450	-73.9889	B02512	NaN	NaN	NaN	NaN
4	9/1/2014 0:11:00	40.8145	-73.9444	B02512	NaN	NaN	NaN	NaN

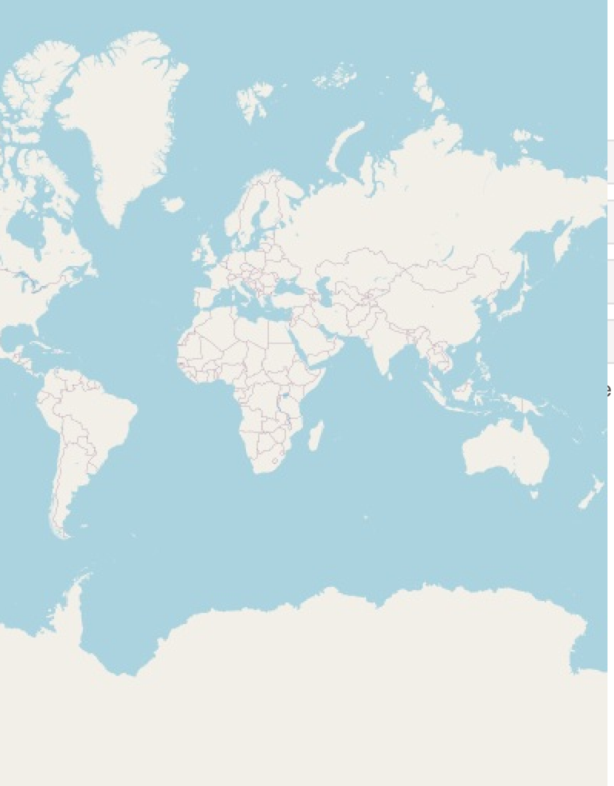
```
In [ ]:
```

```
In [24]: uber_rush = final.groupby(['Lat', 'Lon'], as_index=False).size()
```

```
In [25]: uber_rush.head(5)
```

Out[25]:

	Lat	Lon	size
0	39.6569	-74.2258	1
1	39.6686	-74.1607	1



e -> Trust Notebook

In [29]:
```
from folium.plugins import HeatMap
```

In [30]:
```
HeatMap(uber_rush).add_to(basemap)
```

Out[30]:
```
<folium.plugins.heat_map.HeatMap at 0x26fc76ad310>
```

In [31]:
```
basemap
```

Out[31]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [ ]:

```

```

In [32]:

```
final.head(5)
```

Out[32]:

	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
3	9/1/2014 0:06:00	40.7450	-73.9889	B02512	NaN	NaN	NaN	NaN
4	9/1/2014 0:11:00	40.8145	-73.9444	B02512	NaN	NaN	NaN	NaN

In [33]:

```
final.dtypes
```

Out[33]:

```
Date/Time                 object
Lat                      float64
Lon                      float64
Base                      object
Dispatching_base_num      object
Pickup_date               object
Affiliated_base_num       object
locationID               float64
dtype: object
```

In [34]:

```
final['Date/Time'][0]
```

Out[34]:

```
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 [35]:

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

In [36]:

```
final['Date/Time'].dtype
```

Out[36]:

```
dtype('<M8[ns]')
```

In [37]:

```
final['day'] = final['Date/Time'].dt.day
```

In [38]:

```
final['hour'] = final['Date/Time'].dt.hour
```

In [39]:

```
final.head(5)
```

```
Out[39]:
              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
4   2014-09-01 00:11:00  40.8145  -73.9444  B02512                    NaN          NaN                  NaN         NaN  1.0   0.0
```

```
In [40]: pivot = final.groupby(['day','hour']).size().unstack()
         pivot
```

```
Out[40]:
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   22.0   23.0
day
 1.0  3178  1944  1256  1308  1429  2126  3664  5380  5292  4617  ...   6933   7910   8633   9511   8604   8001   7315   7803   6268   4050
 2.0  2435  1569  1087  1414  1876  2812  4920  6544  6310  4712  ...   6904   8449  10109  11100  11123   9474   8759   8357   6998   5160
 3.0  3354  2142  1407  1467  1550  2387  4241  5663  5386  4657  ...   7226   8850  10314  10491  11239   9599   9026   8531   7142   4686
 4.0  2897  1688  1199  1424  1696  2581  4592  6029  5704  4744  ...   7158   8515   9492  10357  10259   9097   8358   8649   7706   5130
 5.0  2733  1541  1030  1253  1617  2900  4814  6261  6469  5530  ...   6955   8312   9609  10699  10170   9430   9354   9610   8853   6518
 6.0  4537  2864  1864  1555  1551  2162  3642  4766  4942  4401  ...   7235   8612   9444   9929   9263   8405   8117   8567   7852   5946
 7.0  3645  2296  1507  1597  1763  2422  4102  5575  5376  4639  ...   7276   8474  10393  11013  10573   9472   8691   8525   7194   4801
 8.0  2830  1646  1123  1483  1889  3224  5431  7361  7357  5703  ...   7240   8775   9851  10673   9687   8796   8604   8367   6795   4256
 9.0  2657  1724  1222  1480  1871  3168  5802  7592  7519  5895  ...   7877   9220  10270  11910  11449   9804   8909   8665   7499   5203
10.0  3296  2126  1464  1434  1591  2594  4664  6046  6158  5072  ...   7612   9578  11045  11875  10934   9613   9687   9240   7766   5496
11.0  3036  1665  1095  1424  1842  2520  4954  6876  6871  5396  ...   7503   8920  10125  10898  10361   9327   8824   8730   7771   5360
12.0  3227  2147  1393  1362  1757  2710  4576  6250  6231  5177  ...   7743   9390  10734  11713  12216  10393   9965  10310   9992   7945
13.0  5408  3509  2262  1832  1705  2327  4196  5685  6060  5631  ...   8200   9264  10534  11826  11450   9921   8705   8423   7363   5936
14.0  3748  2349  1605  1656  1756  2629  4257  5781  5520  4824  ...   6963   8192   9511  10115   9553   9146   9182   8589   6891   4460
15.0  2497  1515  1087  1381  1862  2980  5050  6837  6729  5201  ...   7633   8505  10285  11959  11728  11032  10509   9105   7153   4480
16.0  2547  1585  1119  1395  1818  2966  5558  7517  7495  5958  ...   7597   9290  10804  11773  10855  10924  10142  10374   8094   5380
17.0  3155  2048  1500  1488  1897  2741  4562  6315  5882  4934  ...   7472   8997  10323  11236  11089   9919   9935   9823   8362   5699
18.0  3390  2135  1332  1626  1892  2959  4688  6618  6451  5377  ...   7534   9040  10274  10692  10338   9551   9310   9285   8015   5492
19.0  3217  2188  1604  1675  1810  2639  4733  6159  6014  5006  ...   7374   8898   9893  10741  10429   9701  10051  10049   9090   6666
20.0  4475  3190  2100  1858  1618  2143  3584  4900  5083  4765  ...   7462   8630   9448  10046   9272   8592   8614   8703   7787   5907
21.0  4294  3194  1972  1727  1926  2615  4185  5727  5529  4707  ...   7064   8127   9483   9817   9291   8317   8107   8245   7362   5231
22.0  2787  1637  1175  1468  1934  3151  5204  6872  6850  5198  ...   7337   9148  10574  10962   9884   8980   8772   8430   6784   4530
23.0  2546  1580  1136  1429  1957  3132  5204  6890  6436  5177  ...   7575   9309   9980  10341  10823  11347  11447  10347   8637   5577
24.0  3200  2055  1438  1493  1798  2754  4484  6013  5913  5146  ...   7083   8706  10366  10786   9772   9080   9213   8831   7480   4456
25.0  2405  1499  1072  1439  1943  2973  5356  7627  7078  5994  ...   7298   8732   9922  10504  10673   9048   8751   9508   8522   6605
26.0  3810  3065  2046  1806  1730  2337  3776  5172  5071  4808  ...   7269   8815   9885  10697  10867  10122   9820  10441   9486   7593
27.0  5196  3635  2352  2055  1723  2336  3539  4937  5053  4771  ...   7519   8803   9793   9838   9228   8267   7908   8507   7720   6046
28.0  4123  2646  1843  1802  1883  2793  4290  5715  5671  5206  ...   7341   8584   9671   9975   9132   8255   8309   7949   6411   4461
29.0  2678  1827  1409  1678  1948  3056  5213  6852  6695  5481  ...   7630   9249  10105  11113  10411   9301   9270   9114   6992   4323
30.0  2401  1510  1112  1403  1841  3216  5757  7596  7611  6064  ...   8396  10243  11554  12126  12561  11024  10836  10042   8275   4723
31.0  2174  1394  1087   919   773   997  1561  2169  2410  2525  ...   4104   5099   5386   5308   5350   4898   4819   5064   5164   3961

31 rows × 24 columns
```

```
In [41]: pivot.style.background_gradient()
```

Out[41]:

hour	0.000000	1.000000	2.000000	3.000000	4.000000	5.000000	6.000000	7.000000	8.000000	9.000000	10.000000	11.000000	12.0000
day													
1.000000	3178	1944	1256	1308	1429	2126	3664	5380	5292	4617	4607	4729	49
2.000000	2435	1569	1087	1414	1876	2812	4920	6544	6310	4712	4797	4975	51
3.000000	3354	2142	1407	1467	1550	2387	4241	5663	5386	4657	4788	5065	53
4.000000	2897	1688	1199	1424	1696	2581	4592	6029	5704	4744	4743	4975	51
5.000000	2733	1541	1030	1253	1617	2900	4814	6261	6469	5530	5141	5011	50
6.000000	4537	2864	1864	1555	1551	2162	3642	4766	4942	4401	4801	5174	54
7.000000	3645	2296	1507	1597	1763	2422	4102	5575	5376	4639	4905	5166	53
8.000000	2830	1646	1123	1483	1889	3224	5431	7361	7357	5703	5288	5350	54
9.000000	2657	1724	1222	1480	1871	3168	5802	7592	7519	5895	5406	5443	54
10.000000	3296	2126	1464	1434	1591	2594	4664	6046	6158	5072	4976	5415	55
11.000000	3036	1665	1095	1424	1842	2520	4954	6876	6871	5396	5215	5423	55
12.000000	3227	2147	1393	1362	1757	2710	4576	6250	6231	5177	5157	5319	55
13.000000	5408	3509	2262	1832	1705	2327	4196	5685	6060	5631	5442	5720	59
14.000000	3748	2349	1605	1656	1756	2629	4257	5781	5520	4824	4911	5118	51
15.000000	2497	1515	1087	1381	1862	2980	5050	6837	6729	5201	5347	5517	55
16.000000	2547	1585	1119	1395	1818	2966	5558	7517	7495	5958	5626	5480	55
17.000000	3155	2048	1500	1488	1897	2741	4562	6315	5882	4934	5004	5306	56
18.000000	3390	2135	1332	1626	1892	2959	4688	6618	6451	5377	5150	5487	54
19.000000	3217	2188	1604	1675	1810	2639	4733	6159	6014	5006	5092	5240	55
20.000000	4475	3190	2100	1858	1618	2143	3584	4900	5083	4765	5135	5650	57
21.000000	4294	3194	1972	1727	1926	2615	4185	5727	5529	4707	4911	5212	54
22.000000	2787	1637	1175	1468	1934	3151	5204	6872	6850	5198	5277	5352	55
23.000000	2546	1580	1136	1429	1957	3132	5204	6890	6436	5177	5066	5304	55
24.000000	3200	2055	1438	1493	1798	2754	4484	6013	5913	5146	4947	5311	52
25.000000	2405	1499	1072	1439	1943	2973	5356	7627	7078	5994	5432	5504	56
26.000000	3810	3065	2046	1806	1730	2337	3776	5172	5071	4808	5061	5179	53
27.000000	5196	3635	2352	2055	1723	2336	3539	4937	5053	4771	5198	5732	58
28.000000	4123	2646	1843	1802	1883	2793	4290	5715	5671	5206	5247	5500	54
29.000000	2678	1827	1409	1678	1948	3056	5213	6852	6695	5481	5234	5163	52
30.000000	2401	1510	1112	1403	1841	3216	5757	7596	7611	6064	5987	6090	64
31.000000	2174	1394	1087	919	773	997	1561	2169	2410	2525	2564	2777	29

In [42]:

```
def gen_pivot_tabel(df, col1, col2):
    pivote = df.groupby([col1, col2]).size().unstack()
    return pivot.style.background_gradient()
```

In [43]:

```
final.columns
```

Out[43]:

```
Index(['Date/Time', 'Lat', 'Lon', 'Base', 'Dispatching_base_num',
       'Pickup_date', 'Affiliated_base_num', 'locationID', 'day', 'hour'],
      dtype='object')
```

In [44]:

```
gen_pivot_tabel(final, 'day', 'hour')
```

Out[44]:

hour	0.000000	1.000000	2.000000	3.000000	4.000000	5.000000	6.000000	7.000000	8.000000	9.000000	10.000000	11.000000	12.0000
day													
1.000000	3178	1944	1256	1308	1429	2126	3664	5380	5292	4617	4607	4729	49
2.000000	2435	1569	1087	1414	1876	2812	4920	6544	6310	4712	4797	4975	51
3.000000	3354	2142	1407	1467	1550	2387	4241	5663	5386	4657	4788	5065	53
4.000000	2897	1688	1199	1424	1696	2581	4592	6029	5704	4744	4743	4975	51
5.000000	2733	1541	1030	1253	1617	2900	4814	6261	6469	5530	5141	5011	50
6.000000	4537	2864	1864	1555	1551	2162	3642	4766	4942	4401	4801	5174	54
7.000000	3645	2296	1507	1597	1763	2422	4102	5575	5376	4639	4905	5166	53
8.000000	2830	1646	1123	1483	1889	3224	5431	7361	7357	5703	5288	5350	54
9.000000	2657	1724	1222	1480	1871	3168	5802	7592	7519	5895	5406	5443	54
10.000000	3296	2126	1464	1434	1591	2594	4664	6046	6158	5072	4976	5415	55
11.000000	3036	1665	1095	1424	1842	2520	4954	6876	6871	5396	5215	5423	55
12.000000	3227	2147	1393	1362	1757	2710	4576	6250	6231	5177	5157	5319	55
13.000000	5408	3509	2262	1832	1705	2327	4196	5685	6060	5631	5442	5720	59
14.000000	3748	2349	1605	1656	1756	2629	4257	5781	5520	4824	4911	5118	51
15.000000	2497	1515	1087	1381	1862	2980	5050	6837	6729	5201	5347	5517	55
16.000000	2547	1585	1119	1395	1818	2966	5558	7517	7495	5958	5626	5480	55
17.000000	3155	2048	1500	1488	1897	2741	4562	6315	5882	4934	5004	5306	56
18.000000	3390	2135	1332	1626	1892	2959	4688	6618	6451	5377	5150	5487	54
19.000000	3217	2188	1604	1675	1810	2639	4733	6159	6014	5006	5092	5240	55
20.000000	4475	3190	2100	1858	1618	2143	3584	4900	5083	4765	5135	5650	57
21.000000	4294	3194	1972	1727	1926	2615	4185	5727	5529	4707	4911	5212	54
22.000000	2787	1637	1175	1468	1934	3151	5204	6872	6850	5198	5277	5352	55
23.000000	2546	1580	1136	1429	1957	3132	5204	6890	6436	5177	5066	5304	55
24.000000	3200	2055	1438	1493	1798	2754	4484	6013	5913	5146	4947	5311	52
25.000000	2405	1499	1072	1439	1943	2973	5356	7627	7078	5994	5432	5504	56
26.000000	3810	3065	2046	1806	1730	2337	3776	5172	5071	4808	5061	5179	53
27.000000	5196	3635	2352	2055	1723	2336	3539	4937	5053	4771	5198	5732	58
28.000000	4123	2646	1843	1802	1883	2793	4290	5715	5671	5206	5247	5500	54
29.000000	2678	1827	1409	1678	1948	3056	5213	6852	6695	5481	5234	5163	52
30.000000	2401	1510	1112	1403	1841	3216	5757	7596	7611	6064	5987	6090	64
31.000000	2174	1394	1087	919	773	997	1561	2169	2410	2525	2564	2777	29

In [ ]:

In [ ]:

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js