Uber New York Pickups by Patrick BENIE

April 7, 2021

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

[11]:
```
# Load & collect entire data set
files=os.listdir(r'/Users/patrickslearningprogams/Desktop/Python Projects/Uber␣
 ↪Trip Deploy/uber-pickups-in-new-york-city')
files
```

[11]:
```
['other-Lyft_B02510.csv',
 'other-FHV-services_jan-aug-2015.csv',
 'other-Firstclass_B01536.csv',
 'other-Skyline_B00111.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']
```

[13]:
```
files.remove('other-Lyft_B02510.csv')
files.remove('other-FHV-services_jan-aug-2015.csv')
files.remove('other-Firstclass_B01536.csv')
files.remove('other-Skyline_B00111.csv')
files.remove('uber-raw-data-janjune-15.csv')
files.remove('other-American_B01362.csv')
```

```
files.remove('Uber-Jan-Feb-FOIL.csv')
files.remove('other-Highclass_B01717.csv')
files.remove('other-Federal_02216.csv')
files.remove('other-Carmel_B00256.csv')
files.remove('other-Diplo_B01196.csv')
files.remove('other-Dial7_B00887.csv')
files.remove('other-Prestige_B01338.csv')
```

[14]:
```
files
```

[14]:
```
['uber-raw-data-apr14.csv',
 'uber-raw-data-aug14.csv',
 'uber-raw-data-sep14.csv',
 'uber-raw-data-jul14.csv',
 'uber-raw-data-jun14.csv',
 'uber-raw-data-may14.csv']
```

[ ]:
```
# Concatenate the 6 remaining csv files into one big data frame
```

[15]:
```
path=r'/Users/patrickslearningprogams/Desktop/Python Projects/Uber Trip Deploy/
 ↪uber-pickups-in-new-york-city'

#blank dataframe
final=pd.DataFrame()

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

[16]:
```
# data on which we will perfom the data analysis
final.shape
```

[16]:
```
(4534327, 4)
```

[ ]:
```

```

[ ]:
```
# Data preparation for Analysis
```

[17]:
```
df=final.copy()
```

[18]:
```
df.head()
# lat: latitude of Uber pickup / lon: longitude of Uber pickup / Base: Base␣
 ↪company code affiliated with Uber pickups
```

[18]:

	Date/Time	Lat	Lon	Base
0	5/1/2014 0:02:00	40.7521	-73.9914	B02512
1	5/1/2014 0:06:00	40.6965	-73.9715	B02512

```
2  5/1/2014 0:15:00  40.7464 -73.9838  B02512
3  5/1/2014 0:17:00  40.7463 -74.0011  B02512
4  5/1/2014 0:17:00  40.7594 -73.9734  B02512
```

[20]:
```
df.dtypes
```

[20]:
```
Date/Time     object
Lat          float64
Lon          float64
Base          object
dtype: object
```

[21]:
```
# we have to convert Date/Time into some timestamp format
df['Date/Time']=pd.to_datetime(df['Date/Time'], format="%m/%d/%Y %H:%M:%S")
```

[22]:
```
df.dtypes
```

[22]:
```
Date/Time    datetime64[ns]
Lat                 float64
Lon                 float64
Base                 object
dtype: object
```

[23]:
```
df.head()
```

[23]:
```
            Date/Time      Lat      Lon    Base
0 2014-05-01 00:02:00  40.7521 -73.9914  B02512
1 2014-05-01 00:06:00  40.6965 -73.9715  B02512
2 2014-05-01 00:15:00  40.7464 -73.9838  B02512
3 2014-05-01 00:17:00  40.7463 -74.0011  B02512
4 2014-05-01 00:17:00  40.7594 -73.9734  B02512
```

[24]:
```
df['weekday']=df['Date/Time'].dt.day_name()
df['day']=df['Date/Time'].dt.day
df['minute']=df['Date/Time'].dt.minute
df['month']=df['Date/Time'].dt.month
df['hour']=df['Date/Time'].dt.hour
```

[25]:
```
df.dtypes
```

[25]:
```
Date/Time    datetime64[ns]
Lat                 float64
Lon                 float64
Base                 object
weekday              object
day                   int64
minute                int64
```

```
month                int64
hour                 int64
dtype: object
```

[26]:
```
df.head()
```

[26]:
```
            Date/Time      Lat      Lon    Base   weekday  day  minute  month  \
0 2014-05-01 00:02:00  40.7521 -73.9914  B02512  Thursday    1       2      5
1 2014-05-01 00:06:00  40.6965 -73.9715  B02512  Thursday    1       6      5
2 2014-05-01 00:15:00  40.7464 -73.9838  B02512  Thursday    1      15      5
3 2014-05-01 00:17:00  40.7463 -74.0011  B02512  Thursday    1      17      5
4 2014-05-01 00:17:00  40.7594 -73.9734  B02512  Thursday    1      17      5

   hour
0     0
1     0
2     0
3     0
4     0
```

[27]:
```
df['Base'].unique()
```

[27]:
```
array(['B02512', 'B02598', 'B02617', 'B02682', 'B02764'], dtype=object)
```

[28]:
```
df['day'].unique()
```

[28]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31])
```

[29]:
```
df['weekday'].unique()
```

[29]:
```
array(['Thursday', 'Friday', 'Saturday', 'Sunday', 'Monday', 'Tuesday',
       'Wednesday'], dtype=object)
```

[31]:
```
df['weekday'].value_counts()
```

[31]:
```
Thursday     755145
Friday       741139
Wednesday    696488
Tuesday      663789
Saturday     646114
Monday       541472
Sunday       490180
Name: weekday, dtype: int64
```

[ ]:
```

```

[ ]:
```
# Analysis of journey by week-days
```

[32]:
```
!pip install plotly
```

```
Collecting plotly
  Downloading plotly-4.14.3-py2.py3-none-any.whl (13.2 MB)
     |                | 13.2 MB 498 kB/s eta 0:00:01
Requirement already satisfied: six in
/Applications/anaconda3/lib/python3.8/site-packages (from plotly) (1.15.0)
Collecting retrying>=1.3.3
  Downloading retrying-1.3.3.tar.gz (10 kB)
Building wheels for collected packages: retrying
  Building wheel for retrying (setup.py) … done
  Created wheel for retrying: filename=retrying-1.3.3-py3-none-any.whl
size=11429
sha256=dfc7e46f10e0b74ac870cacb4cc1489f9e189f463a9626e5cec7d4f01f33f306
  Stored in directory: /Users/patrickslearningprogams/Library/Caches/pip/wheels/
c4/a7/48/0a434133f6d56e878ca511c0e6c38326907c0792f67b476e56
Successfully built retrying
Installing collected packages: retrying, plotly
Successfully installed plotly-4.14.3 retrying-1.3.3
```

[33]:
```
import plotly.express as px
```

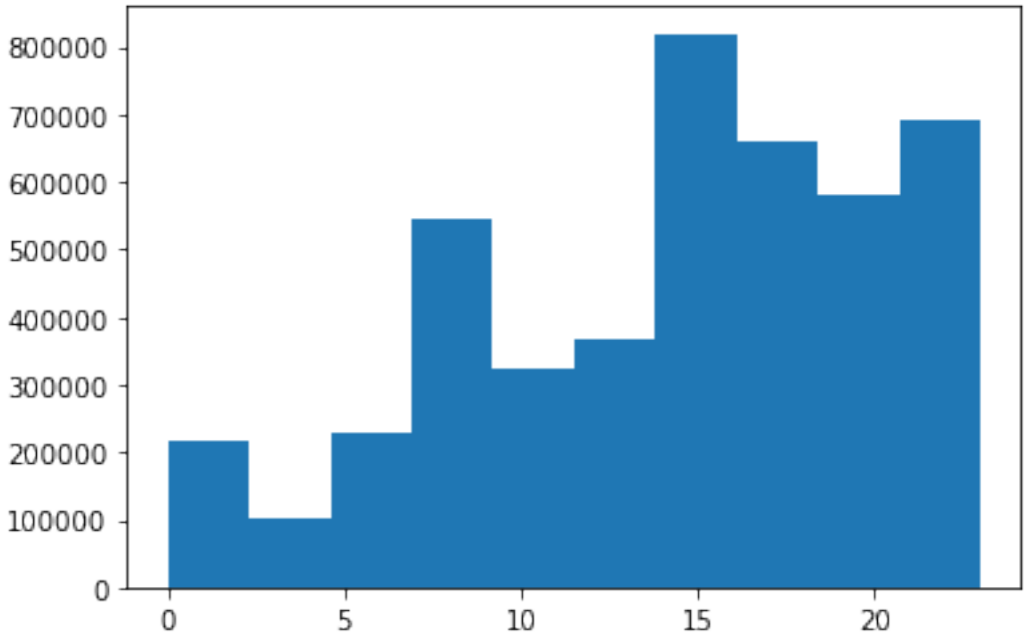
[34]:
```
px.bar(x=df['weekday'].value_counts().index,
       y=df['weekday'].value_counts().values
       )
```

[ ]:
```
# From the chart above we can conclude that Thursdays are the busiest days with␣
↪the highest sales
```

[ ]:
```
# Analysis of journey by hour
```

[36]:
```
plt.hist(df['hour'])
# It peaks during evening time when people are logging off from work
```

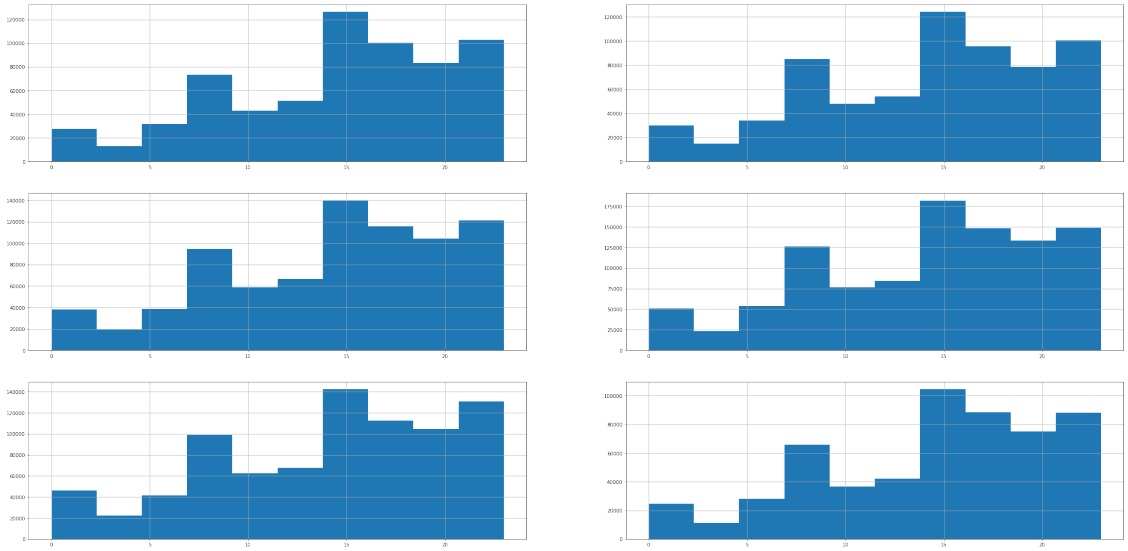
[36]:
```
(array([216928., 103517., 227152., 543565., 324851., 366329., 819491.,
        660869., 579117., 692508.]),
 array([ 0. ,  2.3,  4.6,  6.9,  9.2, 11.5, 13.8, 16.1, 18.4, 20.7, 23. ]),
 <BarContainer object of 10 artists>)
```



[37]:
```
# analysis of journey by hour for each month in the dataframe
for i,month in enumerate(df['month'].unique()):
    print(month)
```

```
5
6
7
9
8
4
```

[38]:
```
plt.figure(figsize=(40,20))
for i,month in enumerate(df['month'].unique()):
    plt.subplot(3,2,i+1)
    df[df['month']==month]['hour'].hist()
```



[ ]:

[ ]:
```
# Analysis of which month as max rides
```

[40]:
```
!pip install chart_studio
```

```
Collecting chart_studio
  Downloading chart_studio-1.1.0-py3-none-any.whl (64 kB)
     |                  | 64 kB 522 kB/s eta 0:00:01
Requirement already satisfied: plotly in
/Applications/anaconda3/lib/python3.8/site-packages (from chart_studio) (4.14.3)
Requirement already satisfied: requests in
/Applications/anaconda3/lib/python3.8/site-packages (from chart_studio) (2.24.0)
Requirement already satisfied: six in
/Applications/anaconda3/lib/python3.8/site-packages (from chart_studio) (1.15.0)
Requirement already satisfied: retrying>=1.3.3 in
/Applications/anaconda3/lib/python3.8/site-packages (from chart_studio) (1.3.3)
Requirement already satisfied: chardet<4,>=3.0.2 in
/Applications/anaconda3/lib/python3.8/site-packages (from
requests->chart_studio) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
/Applications/anaconda3/lib/python3.8/site-packages (from
requests->chart_studio) (2020.6.20)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/Applications/anaconda3/lib/python3.8/site-packages (from
requests->chart_studio) (1.25.11)
Requirement already satisfied: idna<3,>=2.5 in
/Applications/anaconda3/lib/python3.8/site-packages (from
requests->chart_studio) (2.10)
```

```
Installing collected packages: chart-studio
Successfully installed chart-studio-1.1.0
```

[41]:
```
import chart_studio.plotly as py
import plotly.graph_objs as go
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
```

[42]:
```
trace1 = go.Bar(
        x = df.groupby('month')['hour'].sum().index,
        y = df.groupby('month')['hour'].sum(),
        name= 'Priority')
iplot([trace1])
```

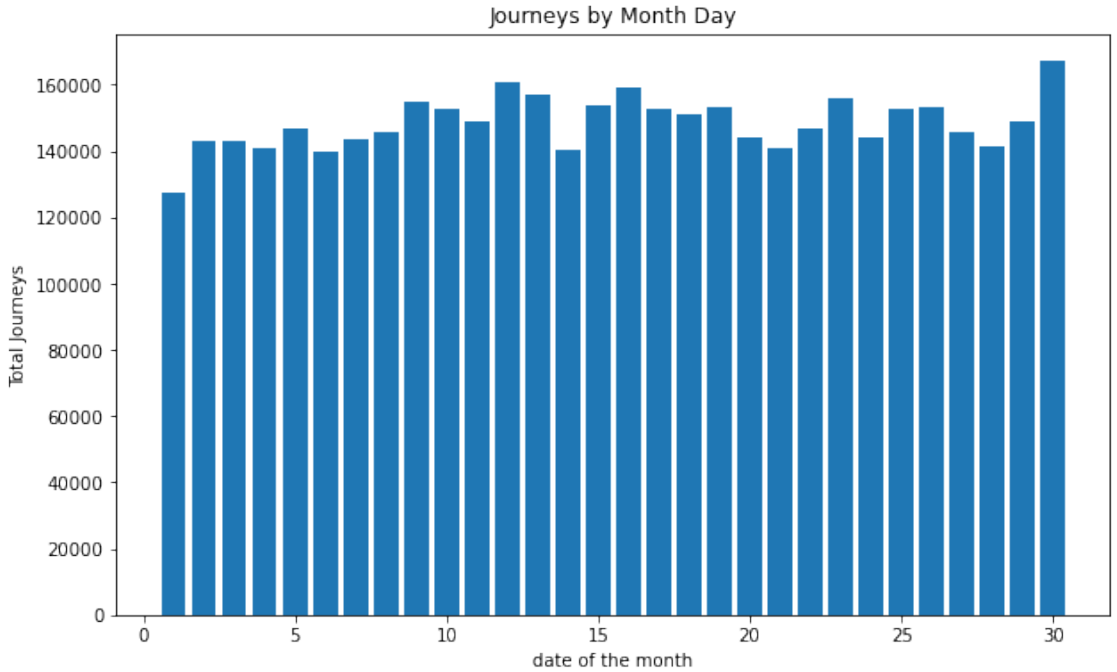
[ ]:
```
# this means that september has the highest rush
```

[ ]:
```

```

[ ]:
```
# Analysis of journey by day
```

[43]:
```
plt.figure(figsize=(10,6))
plt.hist(df['day'], bins=30, rwidth=.8, range=(0.5, 30.5))
plt.xlabel('date of the month')
plt.ylabel('Total Journeys')
plt.title('Journeys by Month Day')
```

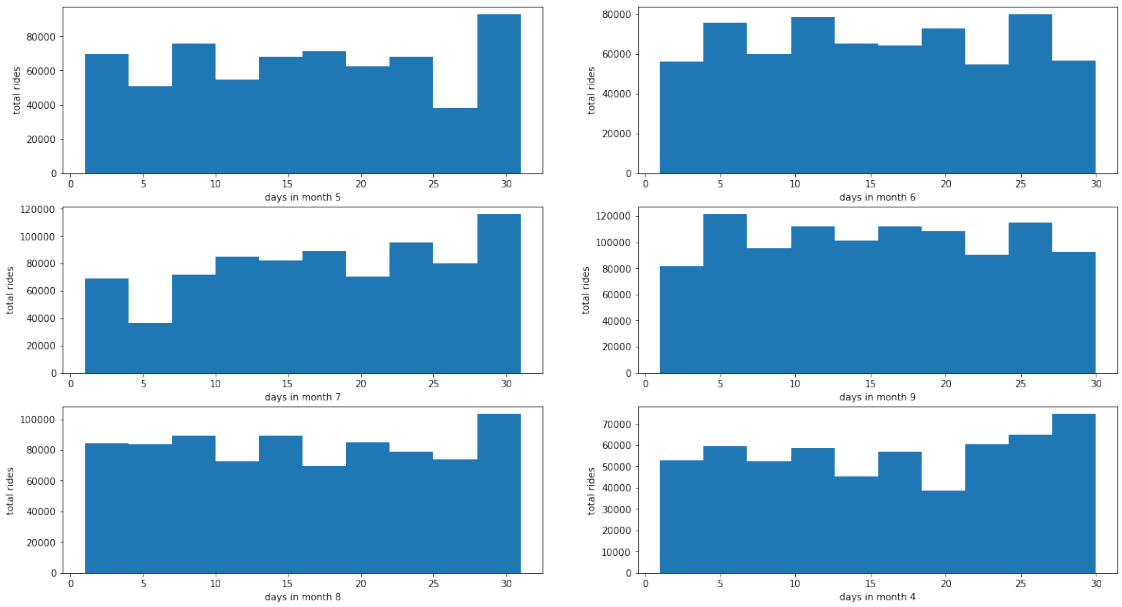
[43]:
```
Text(0.5, 1.0, 'Journeys by Month Day')
```



[ ]:

[ ]:
```
# Analysis of Total rides monthly wise
```

[50]:
```
plt.figure(figsize=(20,11))
for i,month in enumerate(df['month'].unique(),1):
    plt.subplot(3,2,i)
    df_out=df[df['month']==month]
    plt.hist(df_out['day'])
    plt.xlabel('days in month {}'.format(month))
    plt.ylabel('total rides')
```

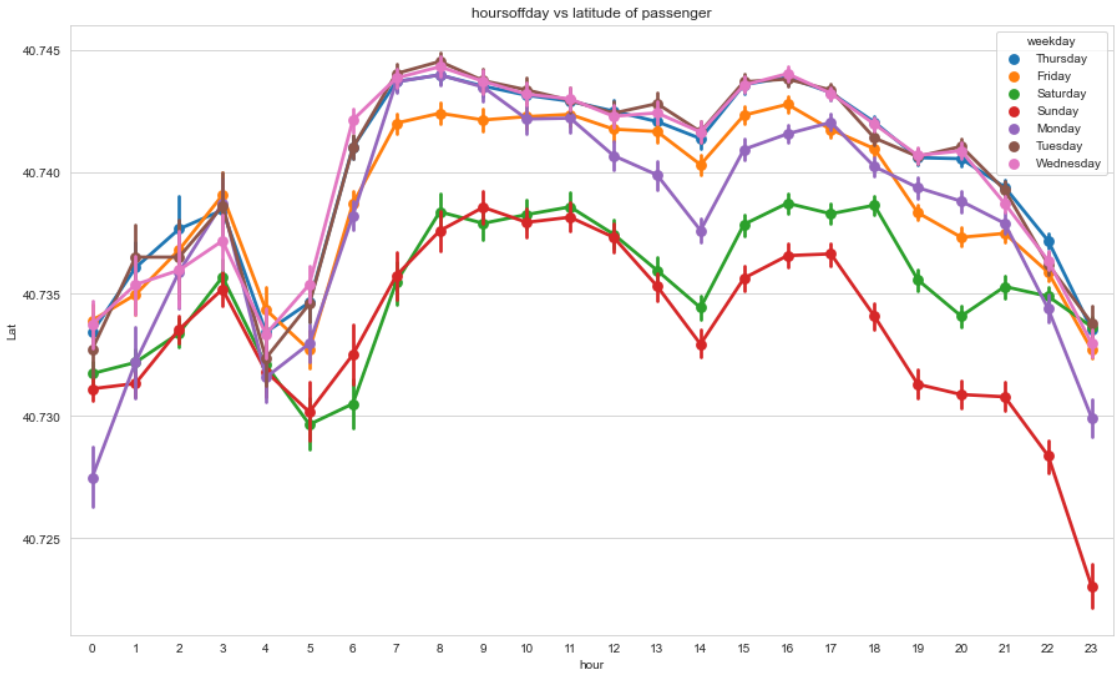


[ ]:

[ ]:
```
# Ananlysis of rush in hour
```

[56]:
```
plt.figure(figsize=(15,9))
sns.set_style(style='whitegrid')
ax=sns.pointplot(x="hour",y="Lat",data=df,hue='weekday')
ax.set_title('hoursoffday vs latitude of passenger')
```

[56]:
```
Text(0.5, 1.0, 'hoursoffday vs latitude of passenger')
```



[ ]:

[ ]:
```
# Analysis of which base number gets popular by months name
```

[57]:
```
df.head()
```

[57]:
```
             Date/Time      Lat      Lon    Base   weekday  day  minute  month  \
0 2014-05-01 00:02:00  40.7521 -73.9914  B02512  Thursday    1       2      5
1 2014-05-01 00:06:00  40.6965 -73.9715  B02512  Thursday    1       6      5
2 2014-05-01 00:15:00  40.7464 -73.9838  B02512  Thursday    1      15      5
3 2014-05-01 00:17:00  40.7463 -74.0011  B02512  Thursday    1      17      5
4 2014-05-01 00:17:00  40.7594 -73.9734  B02512  Thursday    1      17      5

   hour
0     0
1     0
2     0
3     0
4     0
```

[58]:
```
df.groupby(['Base','month'])['Date/Time'].count()
```

[58]:
```
Base    month
B02512  4          35536
        5          36765
```

```
         6          32509
         7          35021
         8          31472
         9          34370
B02598   4         183263
         5         260549
         6         242975
         7         245597
         8         220129
         9         240600
B02617   4         108001
         5         122734
         6         184460
         7         310160
         8         355803
         9         377695
B02682   4         227808
         5         222883
         6         194926
         7         196754
         8         173280
         9         197138
B02764   4           9908
         5           9504
         6           8974
         7           8589
         8          48591
         9         178333
Name: Date/Time, dtype: int64
```

[59]:
```
# create dataframe
base=df.groupby(['Base','month'])['Date/Time'].count().reset_index()
base
```

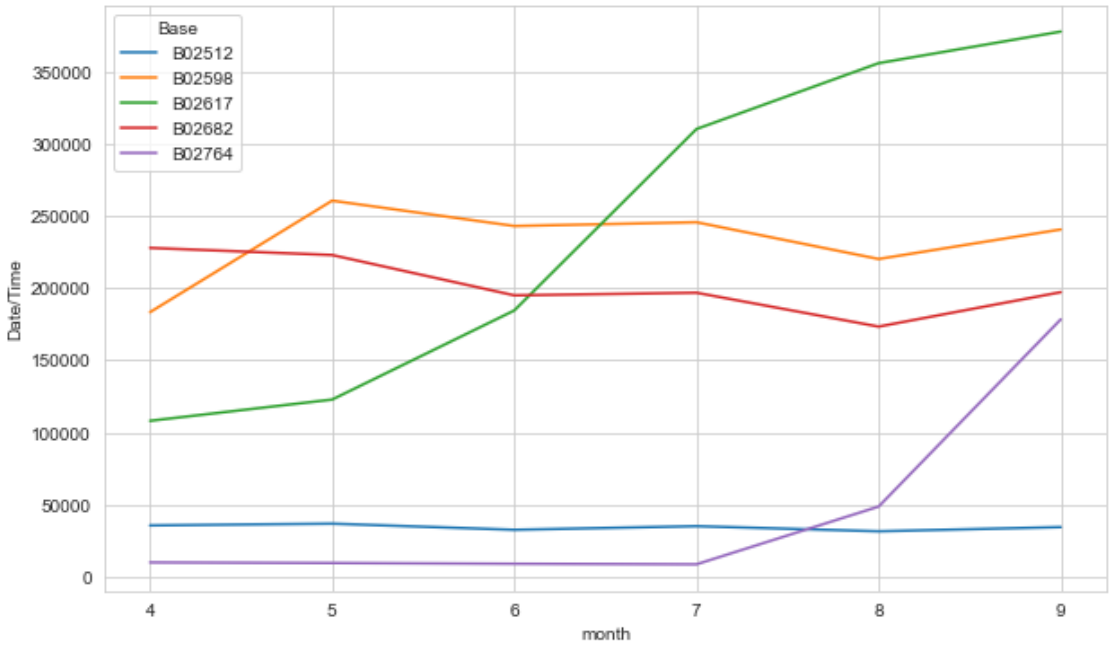
[59]:

	Base	month	Date/Time
0	B02512	4	35536
1	B02512	5	36765
2	B02512	6	32509
3	B02512	7	35021
4	B02512	8	31472
5	B02512	9	34370
6	B02598	4	183263
7	B02598	5	260549
8	B02598	6	242975
9	B02598	7	245597
10	B02598	8	220129
11	B02598	9	240600

12	B02617	4	108001
13	B02617	5	122734
14	B02617	6	184460
15	B02617	7	310160
16	B02617	8	355803
17	B02617	9	377695
18	B02682	4	227808
19	B02682	5	222883
20	B02682	6	194926
21	B02682	7	196754
22	B02682	8	173280
23	B02682	9	197138
24	B02764	4	9908
25	B02764	5	9504
26	B02764	6	8974
27	B02764	7	8589
28	B02764	8	48591
29	B02764	9	178333

[60]:
```
plt.figure(figsize=(10,6))
sns.lineplot(x='month',y='Date/Time',hue='Base',data=base)
```

[60]: <AxesSubplot:xlabel='month', ylabel='Date/Time'>



[ ]:

[ ]:
```
# Perform cross analysis
# Visualize heat maps by 'Hour and weekday','Hour and day', 'Month and day',␣
↪'Month and weekday'
```

[61]:
```
def count_rows(rows):
    return len(rows)
```

[62]:
```
# Hour and weekday
by_cross = df.groupby(['weekday','hour']).apply(count_rows)
by_cross
```

[62]:
```
weekday    hour
Friday     0        13716
           1         8163
           2         5350
           3         6930
           4         8806
                    …
Wednesday  19       47017
           20       47772
           21       44553
           22       32868
           23       18146
Length: 168, dtype: int64
```

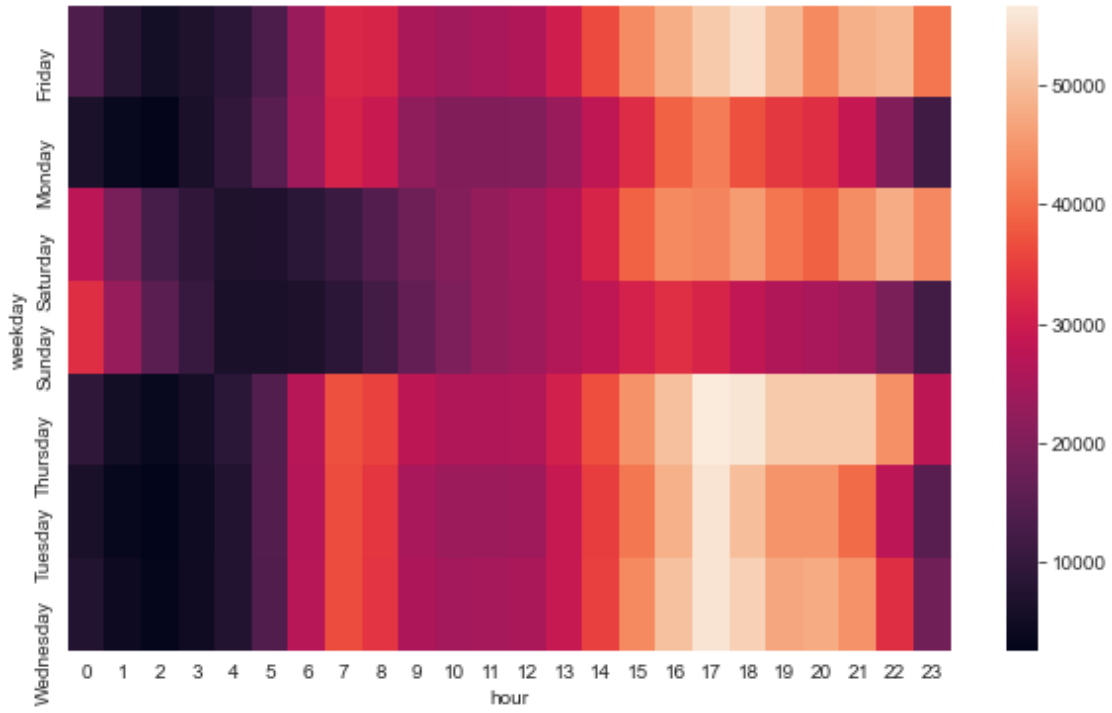
[63]:
```
# unstack() is used to convert dataframes in pivot tables
pivot=by_cross.unstack()
pivot
```

[63]:
```
hour          0      1      2      3     4      5      6      7      8   \
weekday
Friday     13716   8163   5350   6930  8806  13450  23412  32061  31509
Monday      6436   3737   2938   6232  9640  15032  23746  31159  29265
Saturday   27633  19189  12710   9542  6846   7084   8579  11014  14411
Sunday     32877  23015  15436  10597  6374   6169   6596   8728  12128
Thursday    9293   5290   3719   5637  8505  14169  27065  37038  35431
Tuesday     6237   3509   2571   4494  7548  14241  26872  36599  33934
Wednesday   7644   4324   3141   4855  7511  13794  26943  36495  33826

hour          9   …     14     15     16     17     18     19     20     21  \
weekday           …
Friday     25230  …  36206  43673  48169  51961  54762  49595  43542  48323
Monday     22197  …  28157  32744  38770  42023  37000  34159  32849  28925
Saturday   17669  …  31418  38769  43512  42844  45883  41098  38714  43826
Sunday     16401  …  28151  31112  33038  31521  28291  25948  25076  23967
Thursday   27812  …  36699  44442  50560  56704  55825  51907  51990  51953
Tuesday    25023  …  34846  41338  48667  55500  50186  44789  44661  39913
```

```
Wednesday  25635  …  35148  43388  50684  55637  52732  47017  47772  44553

hour          22     23
weekday
Friday      49409  41260
Monday      20158  11811
Saturday    47951  43174
Sunday      19566  12166
Thursday    44194  27764
Tuesday     27712  14869
Wednesday   32868  18146

[7 rows x 24 columns]
```

[64]:
```
# creating heatmap
plt.figure(figsize=(10,6))
sns.heatmap(pivot, annot=False)
```

[64]:
```
<AxesSubplot:xlabel='hour', ylabel='weekday'>
```

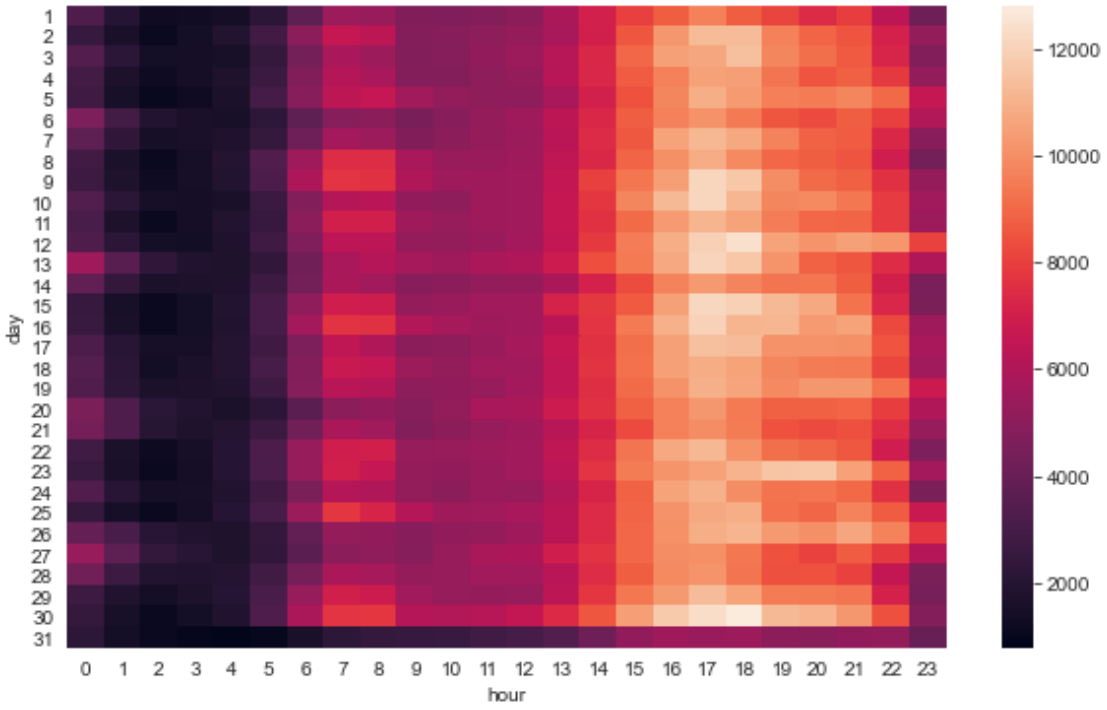


[ ]:

[65]:
```
# Heatmap Function
def heatmap(col1,col2):
    by_cross = df.groupby([col1,col2]).apply(lambda x:len(x))
    pivot=by_cross.unstack()
    plt.figure(figsize=(10,6))
    return sns.heatmap(pivot,annot=False)
```

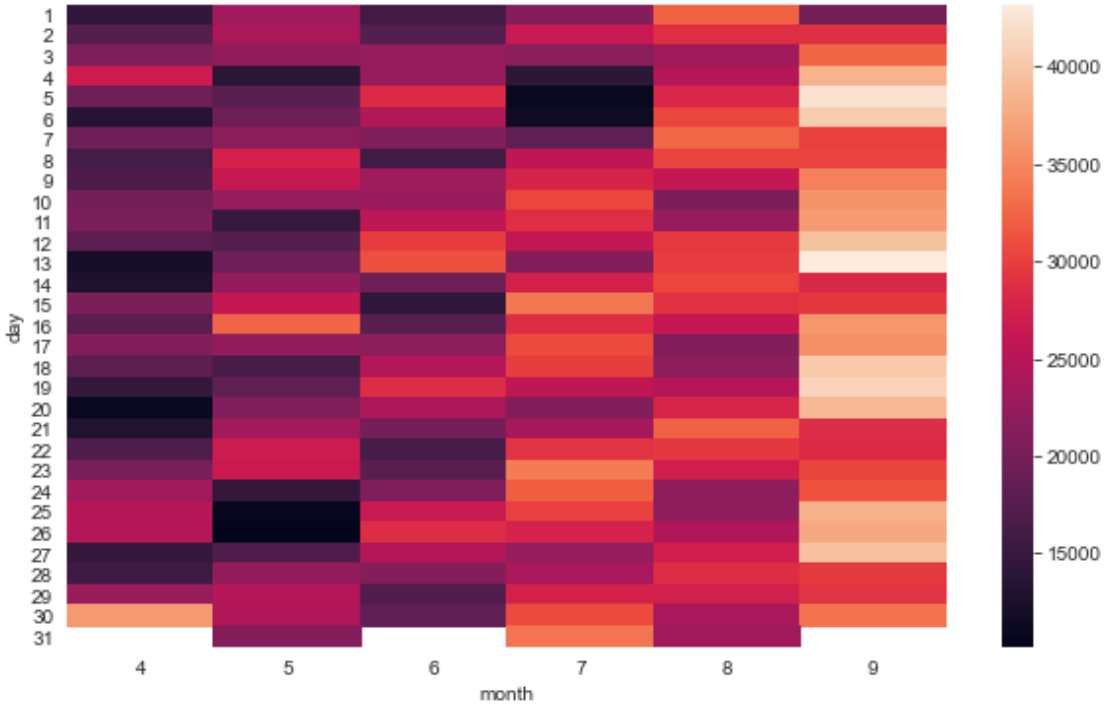
[69]:
```
# Hour and day
heatmap('day', 'hour')
```

[69]:
```
<AxesSubplot:xlabel='hour', ylabel='day'>
```



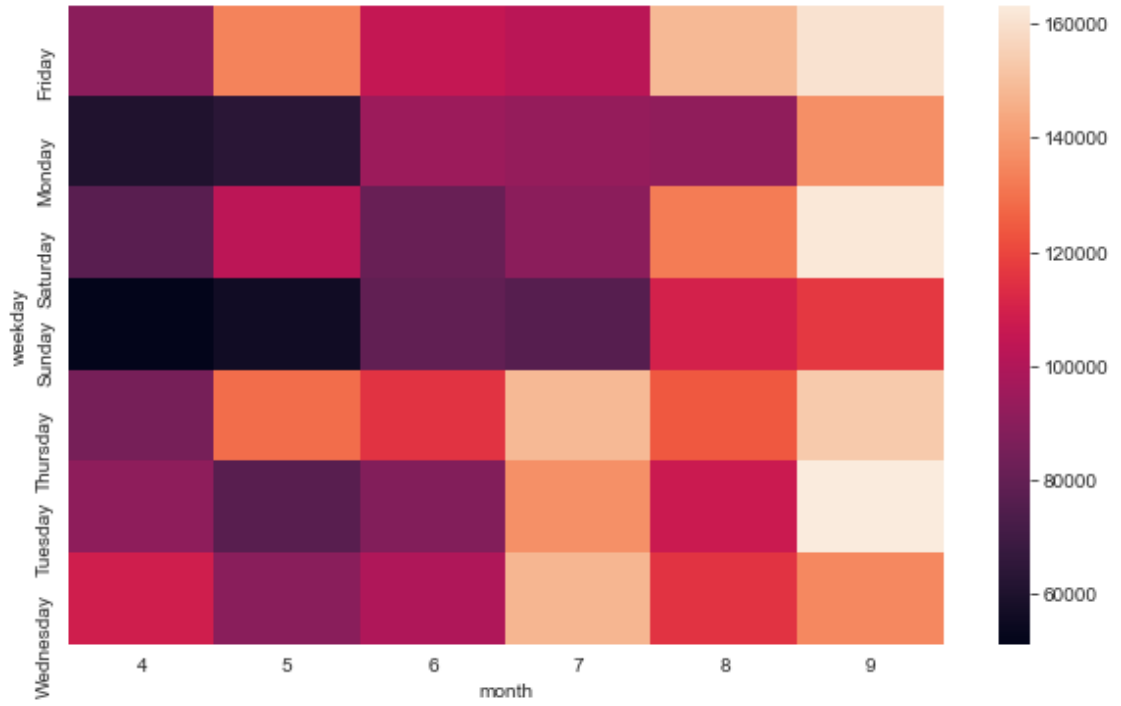
[70]:
```
# Month and day
heatmap('day','month')
```

[70]:
```
<AxesSubplot:xlabel='month', ylabel='day'>
```



[71]:
```
# Month and weekday
heatmap('weekday','month')
```

[71]: <AxesSubplot:xlabel='month', ylabel='weekday'>



[ ]:

[ ]:

[ ]:
```
# Analysis of location data points
```

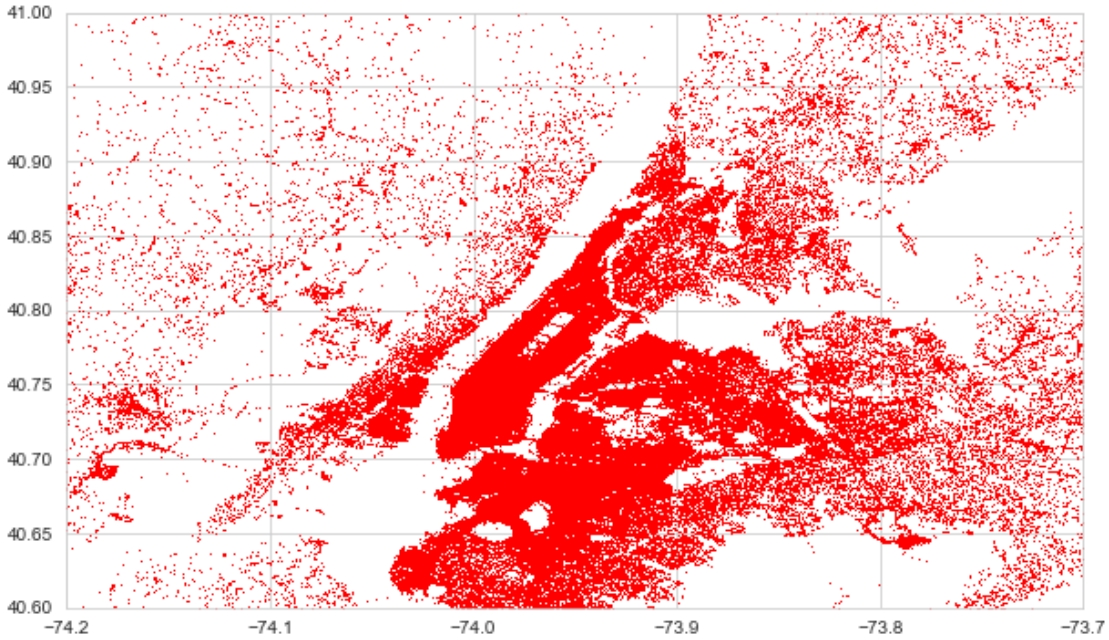
[72]:
```
df.head()
```

[72]:
```
             Date/Time      Lat      Lon    Base   weekday  day  minute  month  \
0 2014-05-01 00:02:00  40.7521 -73.9914  B02512  Thursday    1       2      5
1 2014-05-01 00:06:00  40.6965 -73.9715  B02512  Thursday    1       6      5
2 2014-05-01 00:15:00  40.7464 -73.9838  B02512  Thursday    1      15      5
3 2014-05-01 00:17:00  40.7463 -74.0011  B02512  Thursday    1      17      5
4 2014-05-01 00:17:00  40.7594 -73.9734  B02512  Thursday    1      17      5

   hour
0     0
1     0
2     0
3     0
4     0
```

[73]:
```
plt.figure(figsize=(10,6))
```

```
plt.plot(df['Lon'], df['Lat'],'r+', ms=0.5)
plt.xlim(-74.2, -73.7)
plt.ylim(40.6,41)
```

[73]: (40.6, 41.0)



[ ]:
```
#We can see a number of hot spots here. Midtown Manhattan is clearly a huge␣
↪bright spot.
#& these are made from Midtown to Lower Manhattan.
#Followed by Upper Manhattan and the Heights of Brooklyn
```

[ ]:
```

```

[ ]:
```
# Spacial analysis on Uber requests
```

[74]:
```
df_out = df[df['weekday']=='Sunday']
```

[75]:
```
df_out.shape
```

[75]: (490180, 9)

[76]:
```
df_out.head()
```

[76]:

	Date/Time	Lat	Lon	Base	weekday	day	minute	\
4122	2014-05-04 00:00:00	40.7340	-73.9986	B02512	Sunday	4	0	
4123	2014-05-04 00:01:00	40.7266	-73.9916	B02512	Sunday	4	1	

```
4124 2014-05-04 00:03:00  40.7740 -73.9633  B02512  Sunday    4       3
4125 2014-05-04 00:03:00  40.7698 -73.9923  B02512  Sunday    4       3
4126 2014-05-04 00:03:00  40.7142 -73.9896  B02512  Sunday    4       3

      month  hour
4122      5     0
4123      5     0
4124      5     0
4125      5     0
4126      5     0
```

[ ]:

[77]:
```
#Group data on the basis of Lat and Long
df_out.groupby(['Lat','Lon'])['weekday'].count()
```

[77]:
```
Lat      Lon
39.9374  -74.0722    1
39.9378  -74.0721    1
39.9384  -74.0742    1
39.9385  -74.0734    1
39.9415  -74.0736    1
                    ..
41.3141  -74.1249    1
41.3180  -74.1298    1
41.3195  -73.6905    1
41.3197  -73.6903    1
42.1166  -72.0666    1
Name: weekday, Length: 209230, dtype: int64
```

[78]:
```
#convert into dataframe (rush)
rush=df_out.groupby(['Lat','Lon'])['weekday'].count().reset_index()
rush
```

[78]:
```
            Lat      Lon  weekday
0       39.9374 -74.0722        1
1       39.9378 -74.0721        1
2       39.9384 -74.0742        1
3       39.9385 -74.0734        1
4       39.9415 -74.0736        1
…         …       …       …
209225  41.3141 -74.1249        1
209226  41.3180 -74.1298        1
209227  41.3195 -73.6905        1
209228  41.3197 -73.6903        1
209229  42.1166 -72.0666        1
```

```
[209230 rows x 3 columns]
```

[80]:
```
rush.columns=['Lat','Lon','No of Trips']
rush
```

[80]:
```
             Lat      Lon  No of Trips
0        39.9374 -74.0722            1
1        39.9378 -74.0721            1
2        39.9384 -74.0742            1
3        39.9385 -74.0734            1
4        39.9415 -74.0736            1
…          …       …           …
209225   41.3141 -74.1249            1
209226   41.3180 -74.1298            1
209227   41.3195 -73.6905            1
209228   41.3197 -73.6903            1
209229   42.1166 -72.0666            1

[209230 rows x 3 columns]
```

[81]:
```
!pip install folium
```

```
Collecting folium
  Downloading folium-0.12.1-py2.py3-none-any.whl (94 kB)
     |                  | 94 kB 399 kB/s eta 0:00:01
Requirement already satisfied: jinja2>=2.9 in
/Applications/anaconda3/lib/python3.8/site-packages (from folium) (2.11.2)
Requirement already satisfied: numpy in
/Applications/anaconda3/lib/python3.8/site-packages (from folium) (1.19.2)
Collecting branca>=0.3.0
  Downloading branca-0.4.2-py3-none-any.whl (24 kB)
Requirement already satisfied: requests in
/Applications/anaconda3/lib/python3.8/site-packages (from folium) (2.24.0)
Requirement already satisfied: MarkupSafe>=0.23 in
/Applications/anaconda3/lib/python3.8/site-packages (from jinja2>=2.9->folium)
(1.1.1)
Requirement already satisfied: certifi>=2017.4.17 in
/Applications/anaconda3/lib/python3.8/site-packages (from requests->folium)
(2020.6.20)
Requirement already satisfied: chardet<4,>=3.0.2 in
/Applications/anaconda3/lib/python3.8/site-packages (from requests->folium)
(3.0.4)
Requirement already satisfied: idna<3,>=2.5 in
/Applications/anaconda3/lib/python3.8/site-packages (from requests->folium)
(2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/Applications/anaconda3/lib/python3.8/site-packages (from requests->folium)
```

```
(1.25.11)
Installing collected packages: branca, folium
Successfully installed branca-0.4.2 folium-0.12.1
```

[82]:
```
from folium.plugins import HeatMap
```

[83]:
```
import folium
from folium.plugins import HeatMap
basemap=folium.Map()
```

[84]:
```
HeatMap(df_out.groupby(['Lat','Lon'])['weekday'].count().
 ↪reset_index(),zoom=20,radius=15).add_to(basemap)
basemap
```

[84]:
```
<folium.folium.Map at 0x7fba0530dd90>
```

[ ]:

[ ]:
```
# Automate analysis
```

[85]:
```
def plot(df,day):
    df_out=df[df['weekday']==day]
    df_out.groupby(['Lat','Lon'])['weekday'].count().reset_index()
    HeatMap(df_out.groupby(['Lat','Lon'])['weekday'].count().
 ↪reset_index(),zoom=20,radius=15).add_to(basemap)
    return basemap
```

[87]:
```
plot(df,'Saturday')
```

[87]:
```
<folium.folium.Map at 0x7fba0530dd90>
```

[ ]:

[ ]:

[ ]:
```
# analysis of Uber pickups in each month
# Data preparation
```

[88]:
```
uber_15=pd.read_csv(r'/Users/patrickslearningprogams/Desktop/Python Projects/
 ↪Uber Trip Deploy/uber-pickups-in-new-york-city/uber-raw-data-janjune-15.csv')
```

[89]:
```
uber_15.head()
```

[89]:

	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
2	B02617	2015-05-17 09:47:00	B02617	100

```
3                B02617  2015-05-17 09:47:00              B02774          80
4                B02617  2015-05-17 09:47:00              B02617          90
```

[90]:
```
uber_15.dtypes
```

[90]:
```
Dispatching_base_num    object
Pickup_date             object
Affiliated_base_num     object
locationID               int64
dtype: object
```

[91]:
```
uber_15['Pickup_date'] =  pd.to_datetime(uber_15['Pickup_date'],␣
 ↪format='%Y-%m-%d %H:%M:%S')
```

[92]:
```
uber_15.dtypes
```

[92]:
```
Dispatching_base_num            object
Pickup_date             datetime64[ns]
Affiliated_base_num             object
locationID                       int64
dtype: object
```

[93]:
```
uber_15['weekday']=uber_15['Pickup_date'].dt.day_name()
uber_15['day']=uber_15['Pickup_date'].dt.day
uber_15['minute']=uber_15['Pickup_date'].dt.minute
uber_15['month']=uber_15['Pickup_date'].dt.month
uber_15['hour']=uber_15['Pickup_date'].dt.hour
```

[94]:
```
uber_15.head()
```

[94]:
```
  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
2               B02617 2015-05-17 09:47:00              B02617         100
3               B02617 2015-05-17 09:47:00              B02774          80
4               B02617 2015-05-17 09:47:00              B02617          90

  weekday  day  minute  month  hour
0  Sunday   17      47      5     9
1  Sunday   17      47      5     9
2  Sunday   17      47      5     9
3  Sunday   17      47      5     9
4  Sunday   17      47      5     9
```

[95]:
```
px.bar(x=uber_15['month'].value_counts().index,
           y=uber_15['month'].value_counts().values)
```

[ ]:
```
# We can see that the number of Uber pickup has been steadily increasing␣
↪throughout the first half of 2015 in NYC
```

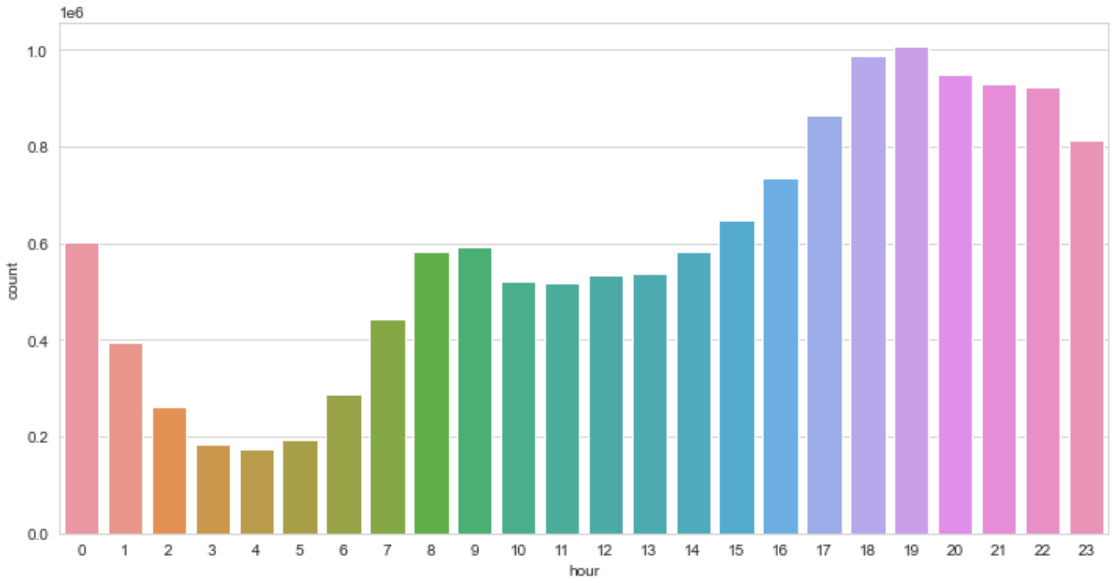
[ ]:
```

```

[ ]:
```
# Analysis of rush in NYC
```

[97]:
```
plt.figure(figsize=(12,6))
sns.countplot(uber_15['hour'])
```

```
/Applications/anaconda3/lib/python3.8/site-packages/seaborn/_decorators.py:36:
FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```

[97]:
```
<AxesSubplot:xlabel='hour', ylabel='count'>
```



[ ]:
```
# In depth analysis of Rush in NYC Day & Hour wise
```

[98]:
```
# Group data by weekday and hour
uber_15.groupby(['weekday', 'hour'])['Pickup_date'].count()
```

[98]:
```
weekday    hour
Friday     0       85939
           1       46616
```

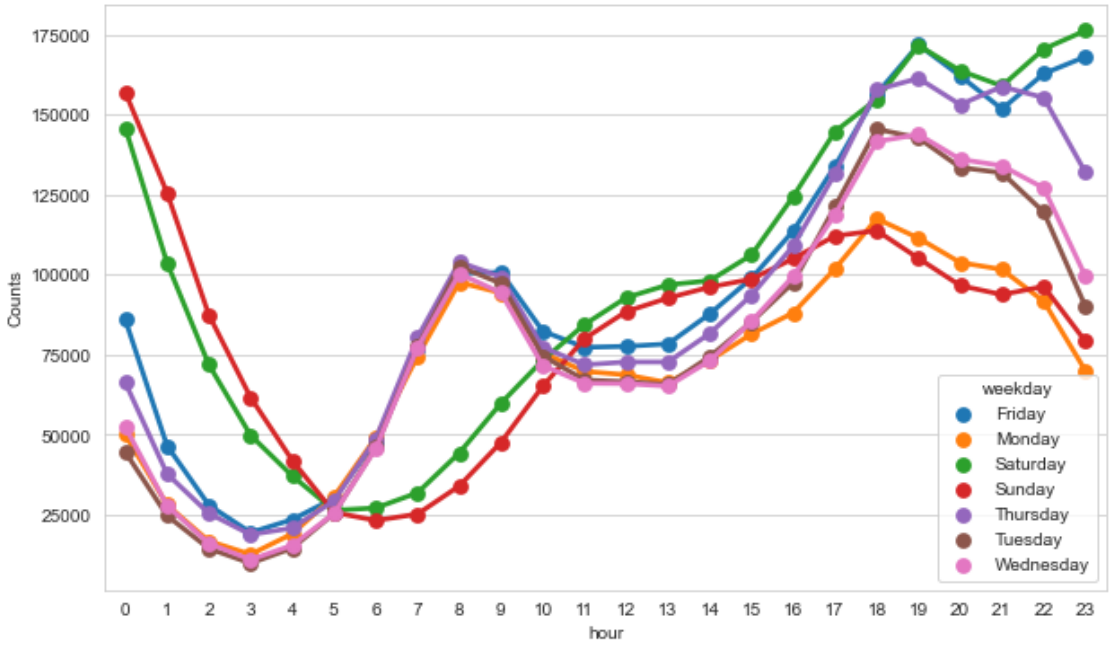
```
                2         28102
                3         19518
                4         23575
                          …
Wednesday       19       143751
                20       136003
                21       133993
                22       127026
                23        99490
Name: Pickup_date, Length: 168, dtype: int64
```

[99]:
```
#convert into dataframe (summary)
summary=uber_15.groupby(['weekday', 'hour'])['Pickup_date'].count().
 ↪reset_index()
summary=summary.rename(columns = {'Pickup_date':'Counts'})
summary
```

[99]:
```
        weekday  hour  Counts
0        Friday     0   85939
1        Friday     1   46616
2        Friday     2   28102
3        Friday     3   19518
4        Friday     4   23575
..          …   …      …
163  Wednesday    19  143751
164  Wednesday    20  136003
165  Wednesday    21  133993
166  Wednesday    22  127026
167  Wednesday    23   99490

[168 rows x 3 columns]
```

[100]:
```
plt.figure(figsize=(10,6))
sns.pointplot(x="hour", y="Counts", hue="weekday", data=summary)
```

[100]:
```
<AxesSubplot:xlabel='hour', ylabel='Counts'>
```



[ ]:

[ ]:

[101]:
```
uber_foil=pd.read_csv(r'/Users/patrickslearningprogams/Desktop/Python Projects/
↪Uber Trip Deploy/uber-pickups-in-new-york-city/Uber-Jan-Feb-FOIL.csv')
```

[102]:
```
uber_foil.head()
```

[102]:

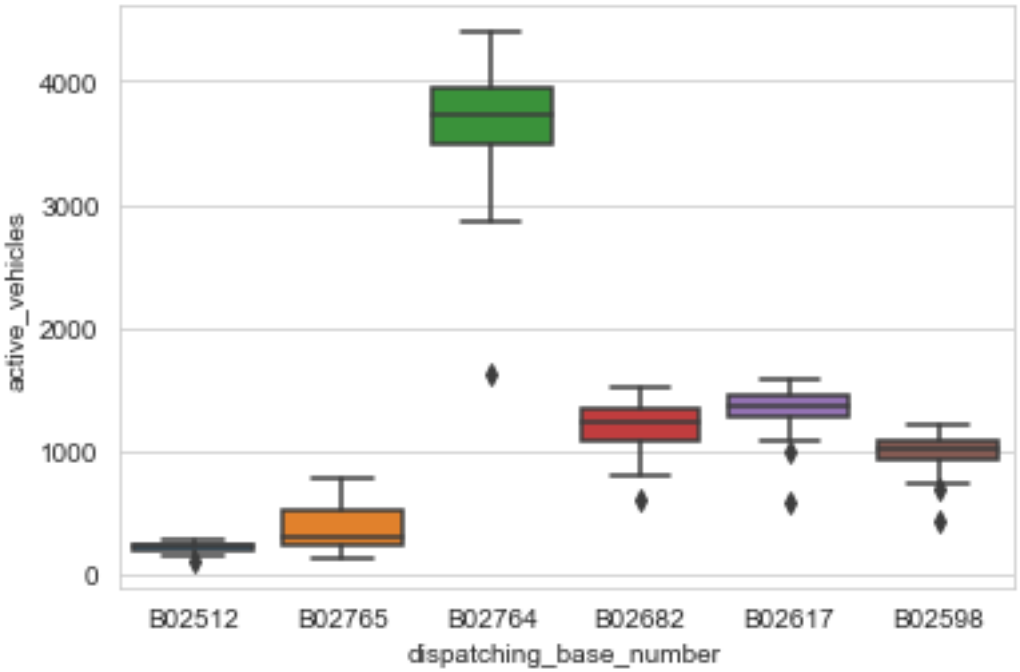
	dispatching_base_number	date	active_vehicles	trips
0	B02512	1/1/2015	190	1132
1	B02765	1/1/2015	225	1765
2	B02764	1/1/2015	3427	29421
3	B02682	1/1/2015	945	7679
4	B02617	1/1/2015	1228	9537

[103]:
```
# Analysis of which base number has the most active vehicles
uber_foil['dispatching_base_number'].unique()
```

[103]:
```
array(['B02512', 'B02765', 'B02764', 'B02682', 'B02617', 'B02598'],
      dtype=object)
```

[104]:
```
sns.boxplot(x = 'dispatching_base_number', y = 'active_vehicles', data =␣
↪uber_foil)
```

[104]:
```
<AxesSubplot:xlabel='dispatching_base_number', ylabel='active_vehicles'>
```

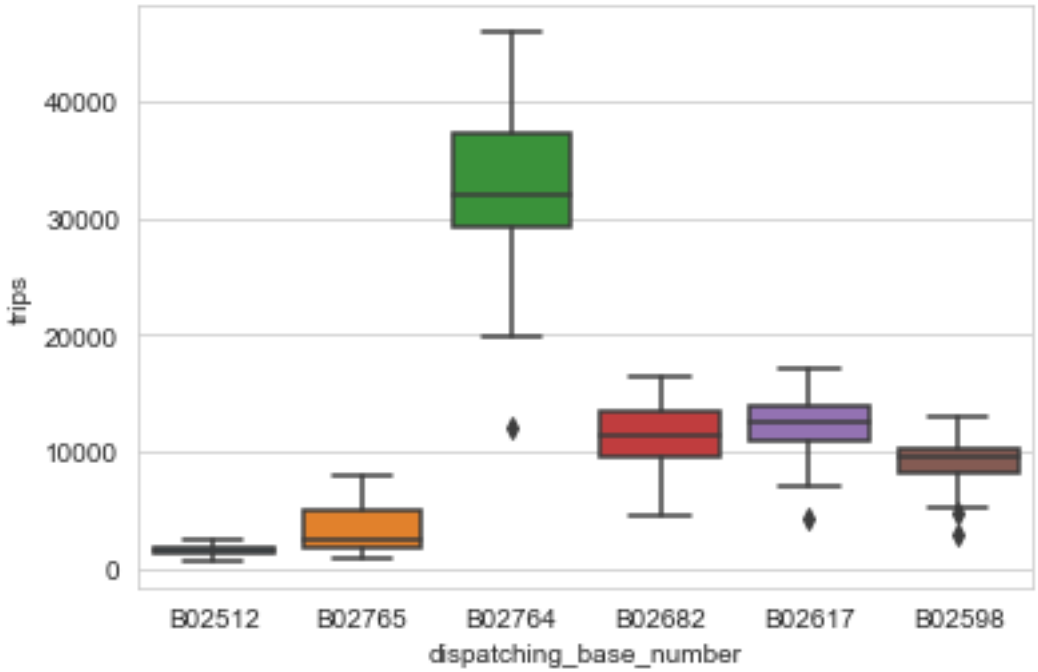


[ ]:
```
#seems to have more number of Active Vehicles in B02764
```

[ ]:
```
#Analysis of which base number has the most trips
```

[105]:
```
sns.boxplot(x = 'dispatching_base_number', y = 'trips', data = uber_foil)
```

[105]:
```
<AxesSubplot:xlabel='dispatching_base_number', ylabel='trips'>
```



[ ]:
```
#seems to have more number of trips in B02764
```

[ ]:
```
```

[106]:
```
# Finding the ratio of trips/active_vehicles
uber_foil['trips/vehicle'] = uber_foil['trips']/uber_foil['active_vehicles']
```

[107]:
```
uber_foil.head()
```

[107]:

	dispatching_base_number	date	active_vehicles	trips	trips/vehicle
0	B02512	1/1/2015	190	1132	5.957895
1	B02765	1/1/2015	225	1765	7.844444
2	B02764	1/1/2015	3427	29421	8.585060
3	B02682	1/1/2015	945	7679	8.125926
4	B02617	1/1/2015	1228	9537	7.766287

[108]:
```
uber_foil.set_index('date')
```

[108]:

date	dispatching_base_number	active_vehicles	trips	trips/vehicle
1/1/2015	B02512	190	1132	5.957895
1/1/2015	B02765	225	1765	7.844444
1/1/2015	B02764	3427	29421	8.585060
1/1/2015	B02682	945	7679	8.125926
1/1/2015	B02617	1228	9537	7.766287

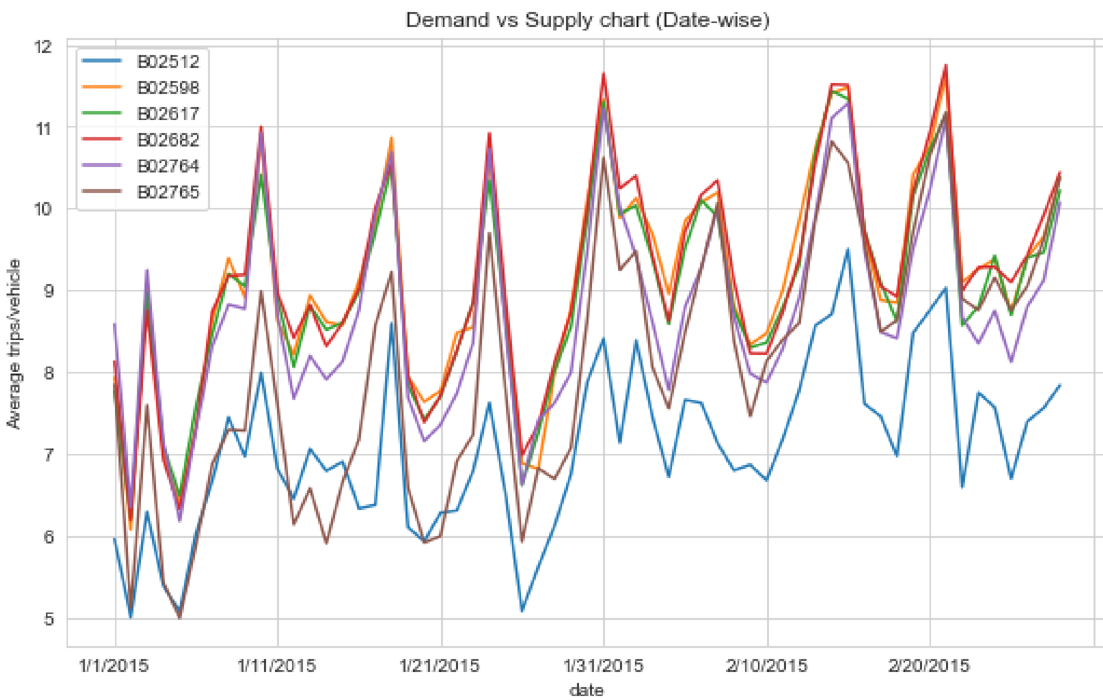
```
…                                          …                    …     …              …
2/28/2015                                  B02764            3952  39812      10.073887
2/28/2015                                  B02617            1372  14022      10.220117
2/28/2015                                  B02682            1386  14472      10.441558
2/28/2015                                  B02512             230   1803       7.839130
2/28/2015                                  B02765             747   7753      10.378849

[354 rows x 4 columns]
```

[ ]:

[ ]:
```
# how Average trips/vehicle inc/decreases with dates with each of base uber
```

[109]:
```
plt.figure(figsize=(10,6))
uber_foil.set_index('date').groupby(['dispatching_base_number'])['trips/
 ↪vehicle'].plot()
plt.ylabel('Average trips/vehicle')
plt.title('Demand vs Supply chart (Date-wise)')
plt.legend()
```

[109]:
```
<matplotlib.legend.Legend at 0x7fb97fc5adf0>
```



[ ]: