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

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

In [10]:
```
files=os.listdir(r'C:\Users\Admin\Desktop\New folder')[-7:]
```

In [12]:
```
files.remove('uber-raw-data-janjune-15.csv')
```

In [13]:
```
files
```

Out[13]:
```
['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 [25]:
```
path = r'C:\Users\Admin\Desktop\New folder'

#blank dataframe
final=pd.DataFrame()

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

In [26]:
```
final
```

Out[26]:

	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
...	...	...	...	...
564511	4/30/2014 23:22:00	40.7640	-73.9744	B02764
564512	4/30/2014 23:26:00	40.7629	-73.9672	B02764
564513	4/30/2014 23:31:00	40.7443	-73.9889	B02764
564514	4/30/2014 23:32:00	40.6756	-73.9405	B02764
564515	4/30/2014 23:48:00	40.6880	-73.9608	B02764

4534327 rows × 4 columns

In [27]:
```
final.shape
```

Out[27]:
```
(4534327, 4)
```

In [28]:
```
df=final.copy()
```

In [30]:
```
df.head()
```

Out[30]:

	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 [33]:
```
df.dtypes
```

Out[33]:
```
Date/Time     object
Lat          float64
Lon          float64
Base          object
dtype: object
```

In [39]:
```
df['Date/Time'] = pd.to_datetime(df['Date/Time'], format="%m/%d/%y %H:&M
```

In [38]:
```
df['Date/Time'] = pd.to_datetime(df['Date/Time'], format="%m/%d/%Y %H:%M
```

In [40]:
```
df.dtypes
```

Out[40]:
```
Date/Time    datetime64[ns]
Lat                 float64
Lon                 float64
Base                 object
dtype: object
```

In [41]:
```
df['Weekdays'] = df['Date/Time'].dt.day_name()
```

In [42]:
```
df['Day'] = df['Date/Time'].dt.day
df['Hour'] = df['Date/Time'].dt.hour
```

In [44]:
```
df['Minute']= df['Date/Time'].dt.minute
df['months'] = df['Date/Time'].dt.month
```

In [45]:
```
df.head()
```

Out[45]:

	Date/Time	Lat	Lon	Base	Weekdays	Day	Hours	Minute	months

0	2014-09-01 00:01:00	40.2201	-74.0021	B02512	Monday	1	0	1	9
1	2014-09-01 00:01:00	40.7500	-74.0027	B02512	Monday	1	0	1	9
2	2014-09-01 00:03:00	40.7559	-73.9864	B02512	Monday	1	0	3	9
3	2014-09-01 00:06:00	40.7450	-73.9889	B02512	Monday	1	0	6	9
4	2014-09-01 00:11:00	40.8145	-73.9444	B02512	Monday	1	0	11	9

In [46]:
```
df.size
```

Out[46]:
```
40808943
```

In [49]:
```
df.iloc[(27)]
```

Out[49]:
```
Date/Time    2014-09-01 03:18:00
Lat                      40.7529
Lon                      -74.004
Base                      B02512
Weekdays                  Monday
Day                            1
Hours                          3
Minute                        18
months                         9
Name: 27, dtype: object
```

In [51]:
```
pip install plotly
```

```
Collecting plotlyNote: you may need to restart the kernel to use updated
packages.
  Downloading plotly-5.3.0-py2.py3-none-any.whl (22.9 MB)
Collecting tenacity>=6.2.0
  Downloading tenacity-8.0.1-py3-none-any.whl (24 kB)
Requirement already satisfied: six in c:\users\admin\anaconda3\lib\site-p
ackages (from plotly) (1.15.0)
Installing collected packages: tenacity, plotly
Successfully installed plotly-5.3.0 tenacity-8.0.1
```

In [52]:
```
import plotly.express as px
```

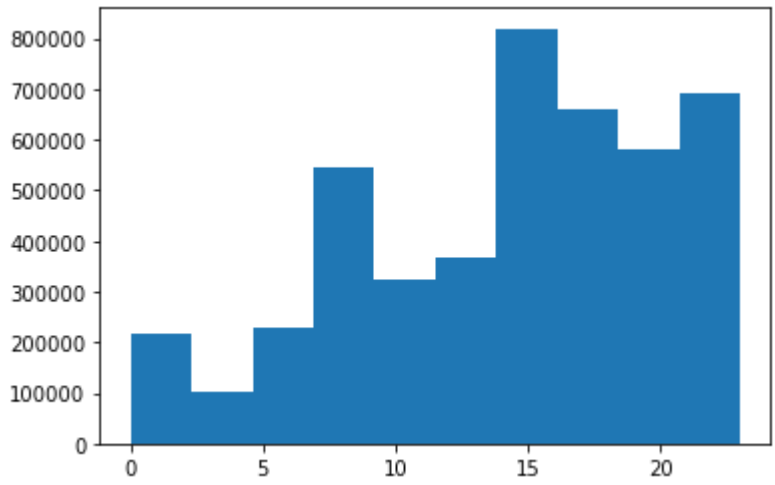
In [59]:
```
df['Weekdays'].value_counts().index
```

Out[59]:
```
Index(['Thursday', 'Friday', 'Wednesday', 'Tuesday', 'Saturday', 'Monda
y',
       'Sunday'],
      dtype='object')
```

In [60]:
```
px.bar(x=df['Weekdays'].value_counts().index,
       y=df['Weekdays'].value_counts())
```

In [68]:
```
plt.hist(df['Hours'])
```

Out[68]:
```
(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>)
```



In [72]:
```
df['months'].unique()
```

Out[72]:
```
array([9, 5, 6, 7, 8, 4], dtype=int64)
```

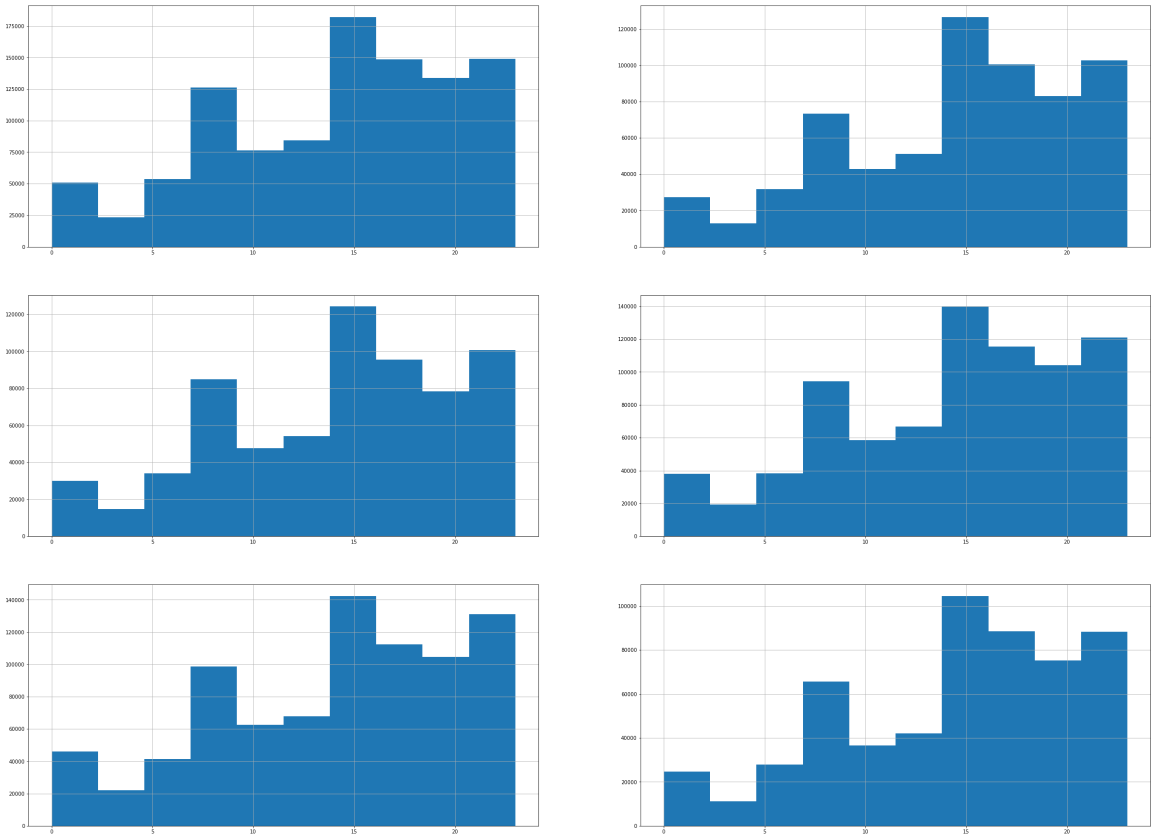
In [74]:
```
for i,month in enumerate(df['months'].unique()):
```

```
    print(i)
    print(month)
```

```
0
9
1
5
2
6
3
7
4
8
5
4
```

In [77]:

```
plt.figure(figsize=(40,30))
for i,month in enumerate(df['months'].unique()):
    plt.subplot(3,2,i+1)
    df[df['months']==month]['Hours'].hist()
```



In [89]:

```
import plotly.graph_objs as go
from plotly.offline import download_plotlyjs, init_notebook_mode, plot,
```

In [92]:

```
df.groupby('months')['Hours'].count()
```

Out[92]:

```
months
4     564516
5     652435
6     663844
7     796121
8     829275
9    1028136
Name: Hours, dtype: int64
```

In [ ]:

```
```

In [99]:

```
trace1 = go.Bar(
x= df.groupby('months')['Hours'].count().index,
    y= df.groupby('months')['Hours'].count(),
    name = 'Priority '
)
```

In [101…

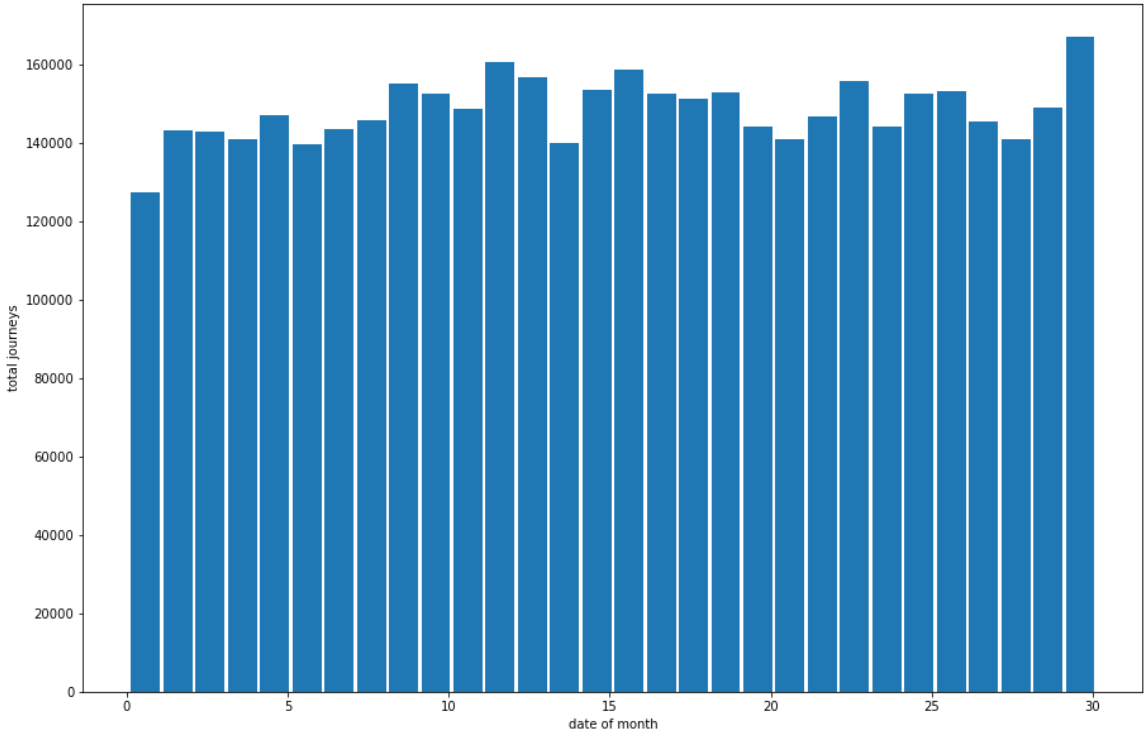
```
iplot([trace1])
```

In [140…

```
plt.figure(figsize=(15,10))
plt.hist(df['Day'],bins=30, rwidth=0.9,range=(0.1,30.1))
plt.xlabel('date of month')
plt.ylabel('total journeys')
```

Out[140…

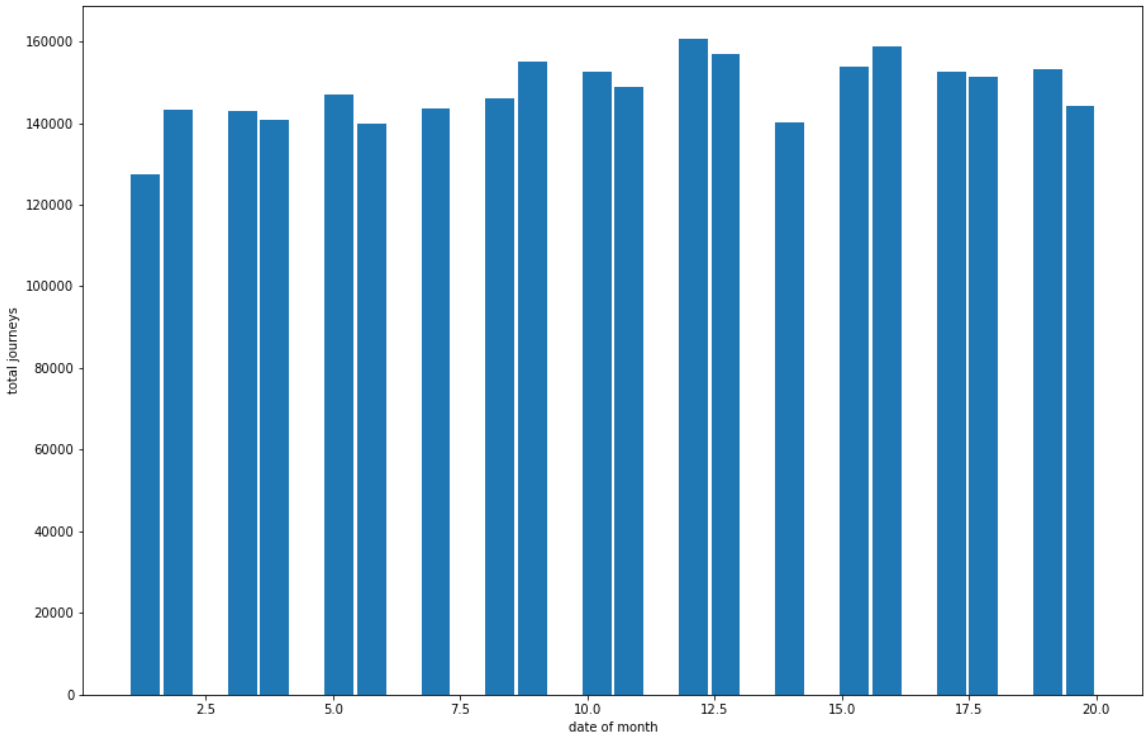
```
Text(0, 0.5, 'total journeys')
```



In [144…

```
plt.figure(figsize=(15,10))
plt.hist(df['Day'],bins=30, rwidth=0.9,range=(1,20))
plt.xlabel('date of month')
plt.ylabel('total journeys')
```

Out[144… Text(0, 0.5, 'total journeys')



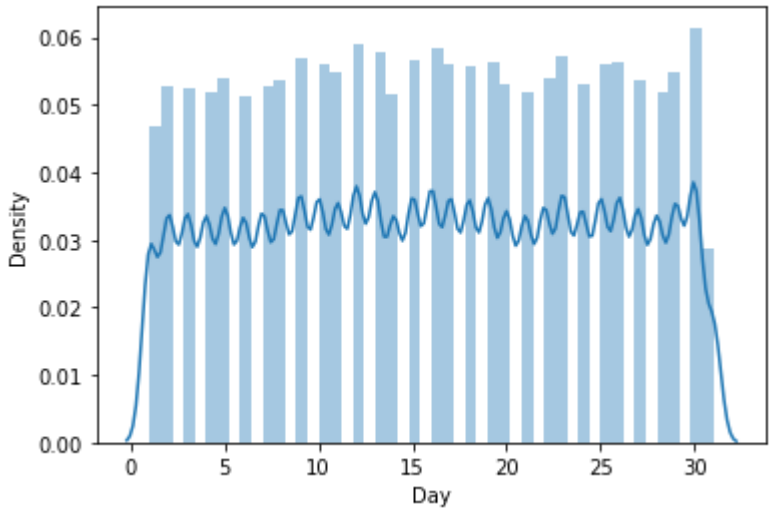
In [121…

```
sns.distplot(df['Day'])
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\distributions.py:2557:
FutureWarning:

`distplot` is a deprecated function and will be removed in a future versi
```

```
on. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
```

Out[121…
```
<AxesSubplot:xlabel='Day', ylabel='Density'>
```



In [126…
```
df['months'].value_counts()
```

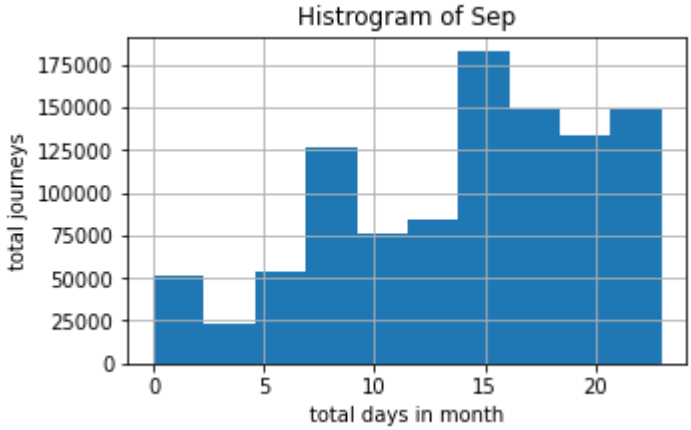
Out[126…
```
9    1028136
8     829275
7     796121
6     663844
5     652435
4     564516
Name: months, dtype: int64
```

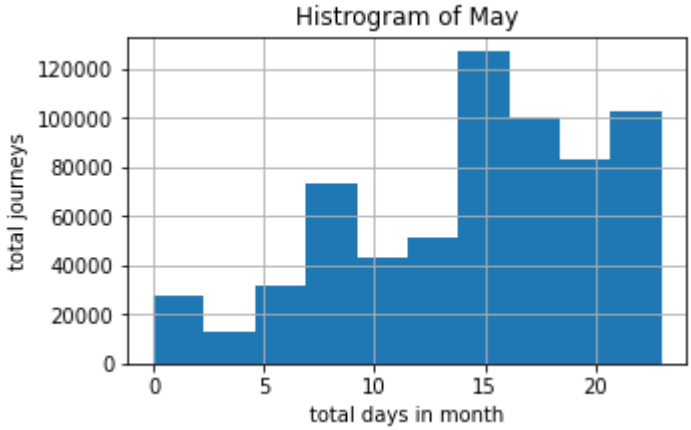
In [130…
```
df.groupby('months')['Day'].count()
```

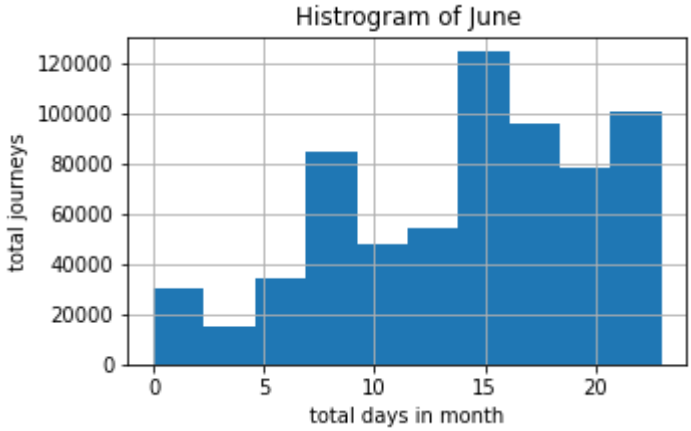
Out[130…
```
months
4     564516
5     652435
6     663844
7     796121
8     829275
9    1028136
Name: Day, dtype: int64
```

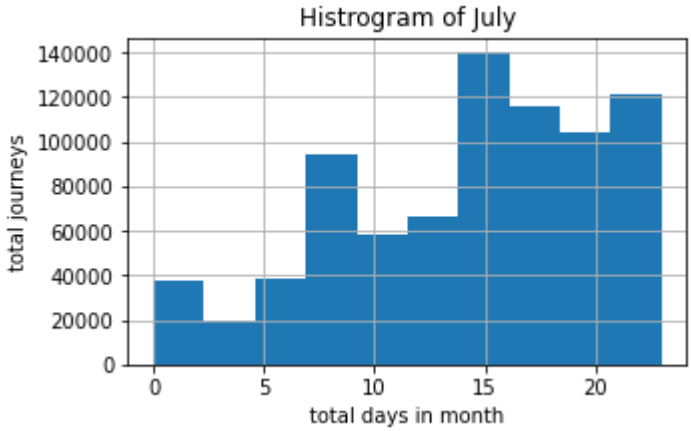
In [141…
```
plt.figure(figsize=(10,8))

months={9:'Sep',5:'May',6:'June',7:'July',8:'August',4:'April'}
for i in df['months'].unique():
    plt.figure(figsize=(5,3))
    df[df['months']==i]['Hours'].hist()
    plt.title('Histrogram of {}'.format(months[i]))
    plt.xlabel('total days in month')
    plt.ylabel('total journeys')
```

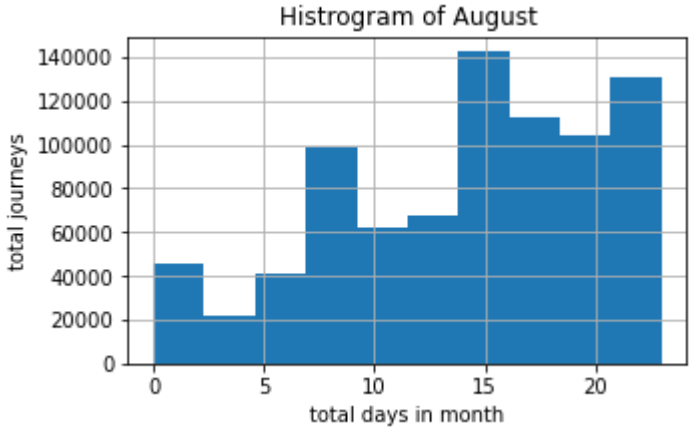
```
<Figure size 720x576 with 0 Axes>
```

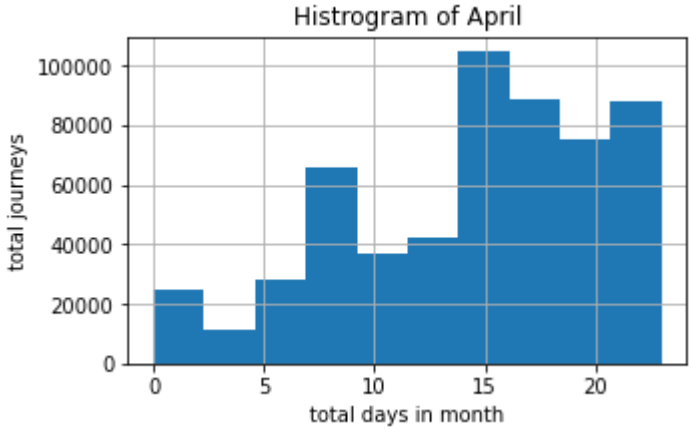

Histrogram of Sep
total journeys
175000
150000
125000
100000
75000
50000
25000
0
0
5
10
15
20
total days in month


Histrogram of May
total journeys
120000
100000
80000
60000
40000
20000
0
0
5
10
15
20
total days in month


Histrogram of June
total journeys
120000
100000
80000
60000
40000
20000
0
0
5
10
15
20
total days in month


Histrogram of July
total journeys
140000
120000
100000
80000
60000
40000
20000
0
0
5
10
15
20
total days in month

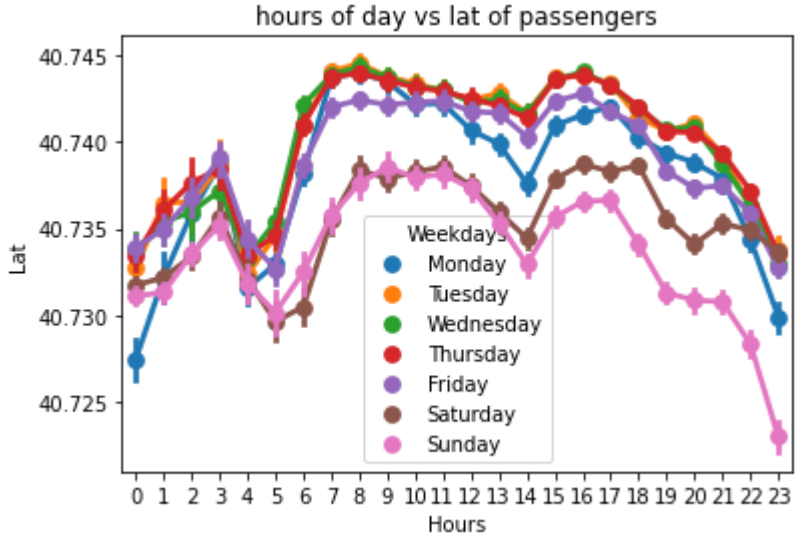




In [174…

```
ax= sns.pointplot(x='Hours',y='Lat',data=df,hue='Weekdays')
ax.set_title('hours of day vs lat of passengers')
```

Out[174… `Text(0.5, 1.0, 'hours of day vs lat of passengers')`



In [146…

```
df
```

Out[146…

	Date/Time	Lat	Lon	Base	Weekdays	Day	Hours	Minute	months
0	2014-09-01 00:01:00	40.2201	-74.0021	B02512	Monday	1	0	1	9
1	2014-09-01	40.7500	-74.0027	B02512	Monday	1	0	1	9

	00:01:00								
2	2014-09-01 00:03:00	40.7559	-73.9864	B02512	Monday	1	0	3	9
3	2014-09-01 00:06:00	40.7450	-73.9889	B02512	Monday	1	0	6	9
4	2014-09-01 00:11:00	40.8145	-73.9444	B02512	Monday	1	0	11	9
...	...	...	...	...	...	...	...	...	...
564511	2014-04-30 23:22:00	40.7640	-73.9744	B02764	Wednesday	30	23	22	4
564512	2014-04-30 23:26:00	40.7629	-73.9672	B02764	Wednesday	30	23	26	4
564513	2014-04-30 23:31:00	40.7443	-73.9889	B02764	Wednesday	30	23	31	4
564514	2014-04-30 23:32:00	40.6756	-73.9405	B02764	Wednesday	30	23	32	4
564515	2014-04-30 23:48:00	40.6880	-73.9608	B02764	Wednesday	30	23	48	4

4534327 rows × 9 columns

In [147…
```
df['Base'].head()
```

Out[147…
```
0    B02512
1    B02512
2    B02512
3    B02512
4    B02512
Name: Base, dtype: object
```

In [152…
```
base = df.groupby(['Base','months'])['Date/Time'].count().reset_index()
```

In [153…
```
base
```

Out[153…

	Base	months	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

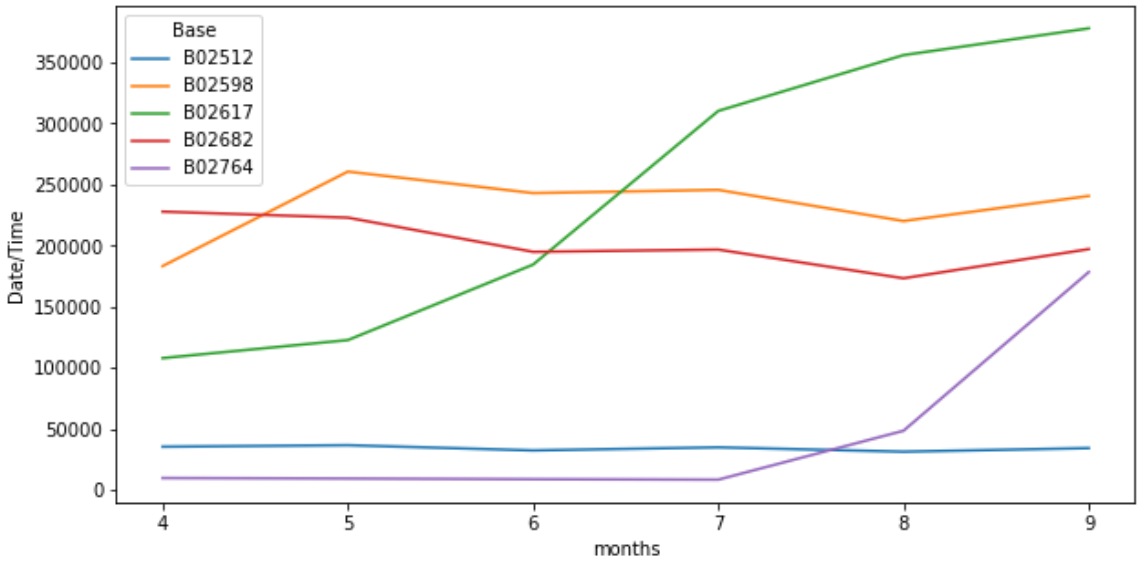
8	B02598	6	242975
9	B02598	7	245597
10	B02598	8	220129
11	B02598	9	240600
12	B02617	4	108001
13	B02617	5	122734
14	B02617	6	184460
15	B02617	7	310160
16	B02617	8	355803
17	B02617	9	377695
18	B02682	4	227808
19	B02682	5	222883
20	B02682	6	194926
21	B02682	7	196754
22	B02682	8	173280
23	B02682	9	197138
24	B02764	4	9908
25	B02764	5	9504
26	B02764	6	8974
27	B02764	7	8589
28	B02764	8	48591
29	B02764	9	178333

In [176…

```
plt.figure(figsize=(10,5))
sns.lineplot(x='months', y= 'Date/Time', hue = 'Base', data= base)
```

Out[176…

```
<AxesSubplot:xlabel='months', ylabel='Date/Time'>
```



In [170…

```
def count_rows(rows):
```

```
    return len(rows)
```

In [172…

```
hm= df.groupby(['Weekdays','Hours']).apply(count_rows)
hm
```

Out[172…

```
Weekdays   Hours
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
```

In [177…

```
pivot = hm.unstack()
```

In [178…

```
pivot
```

Out[178…

Hours	0	1	2	3	4	5	6	7	8	9	...
Weekdays											
Friday	13716	8163	5350	6930	8806	13450	23412	32061	31509	25230	... 36
Monday	6436	3737	2938	6232	9640	15032	23746	31159	29265	22197	... 28
Saturday	27633	19189	12710	9542	6846	7084	8579	11014	14411	17669	... 31
Sunday	32877	23015	15436	10597	6374	6169	6596	8728	12128	16401	... 28
Thursday	9293	5290	3719	5637	8505	14169	27065	37038	35431	27812	... 36
Tuesday	6237	3509	2571	4494	7548	14241	26872	36599	33934	25023	... 34
Wednesday	7644	4324	3141	4855	7511	13794	26943	36495	33826	25635	... 35

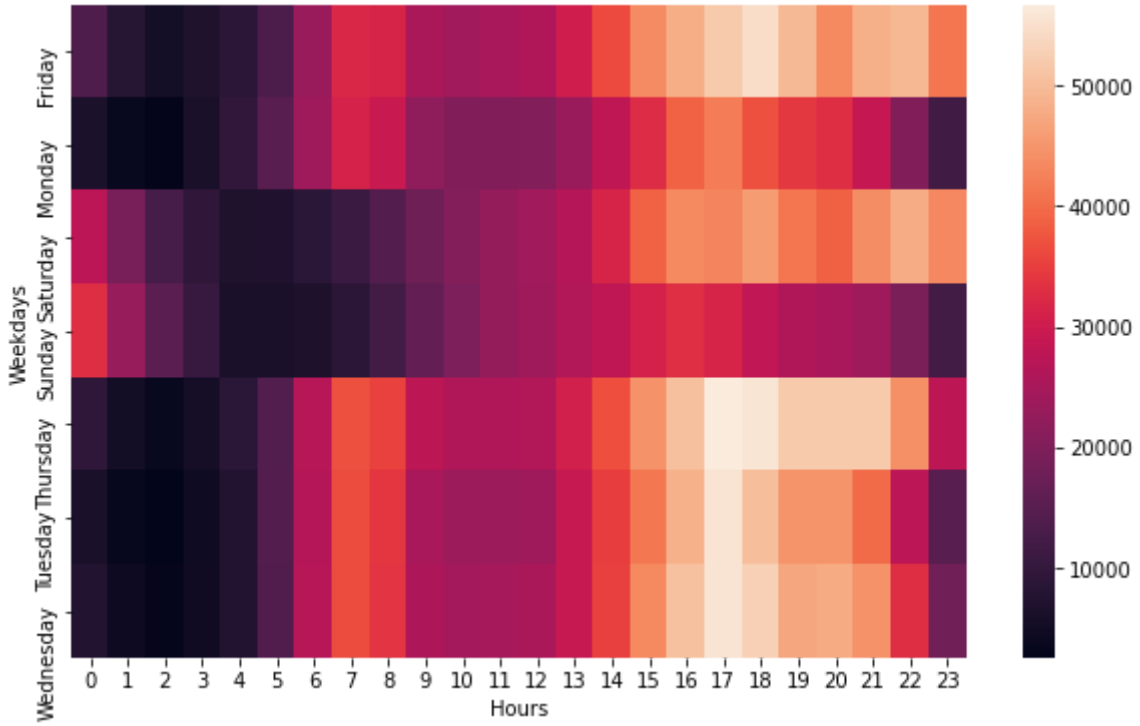
7 rows × 24 columns

In [180…

```
plt.figure(figsize=(10,6))
sns.heatmap(pivot, annot=False)
```

Out[180…

```
<AxesSubplot:xlabel='Hours', ylabel='Weekdays'>
```



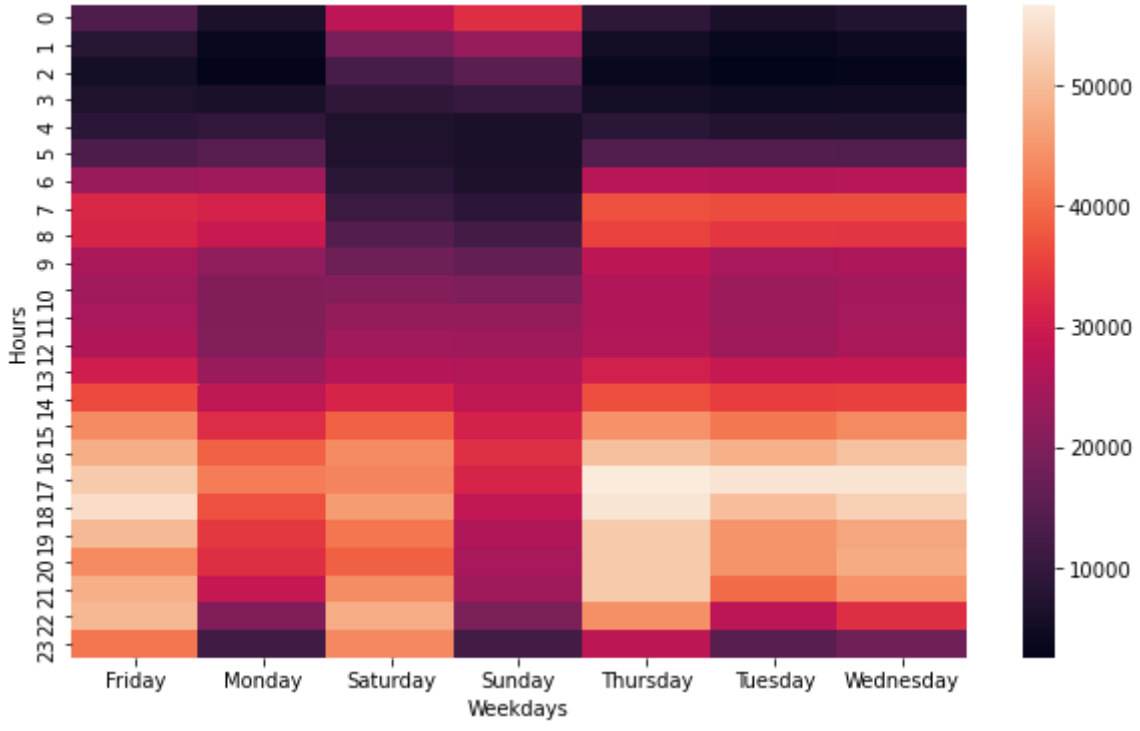
In [190…

```
def heatmap(col1,col2):
    hm= df.groupby([col1,col2]).apply(count_rows)
    pivot = hm.unstack()
    plt.figure(figsize=(10,6))
    return sns.heatmap(pivot)
```

In [191…

```
heatmap('Hours','Weekdays')
```

Out[191… <AxesSubplot:xlabel='Weekdays', ylabel='Hours'>



In [186…

```
df.head()
```

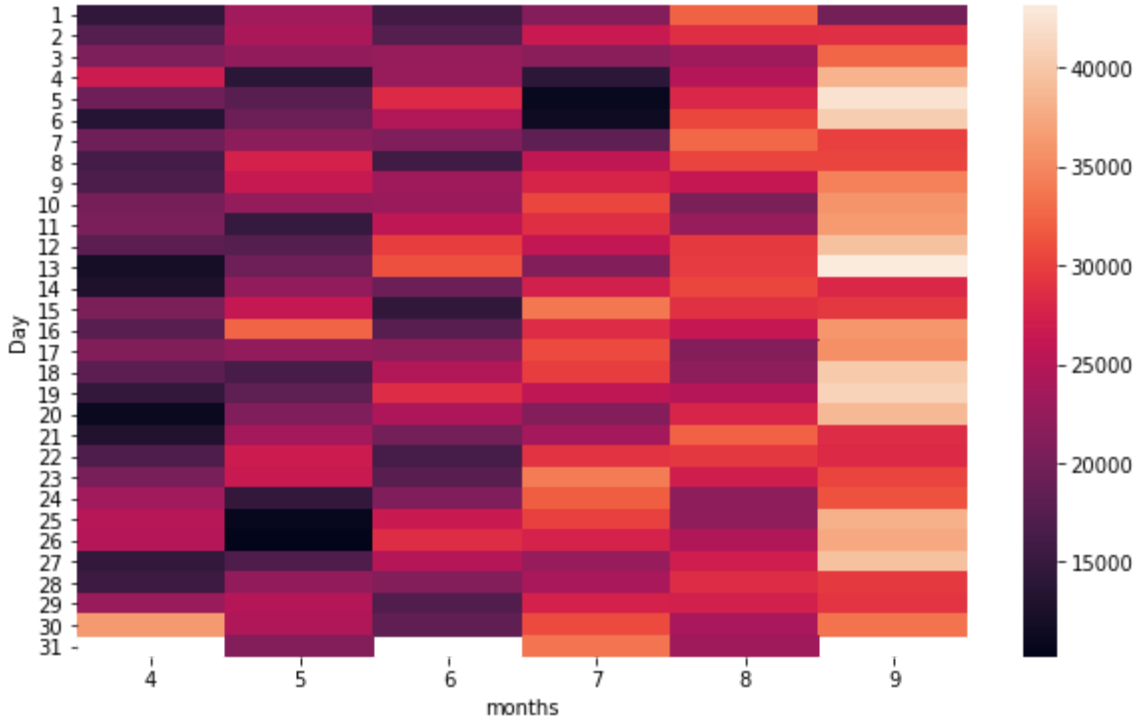
Out[186…

	Date/Time	Lat	Lon	Base	Weekdays	Day	Hours	Minute	months
0	2014-09-01 00:01:00	40.2201	-74.0021	B02512	Monday	1	0	1	9
1	2014-09-01 00:01:00	40.7500	-74.0027	B02512	Monday	1	0	1	9
2	2014-09-01 00:03:00	40.7559	-73.9864	B02512	Monday	1	0	3	9
3	2014-09-01 00:06:00	40.7450	-73.9889	B02512	Monday	1	0	6	9
4	2014-09-01 00:11:00	40.8145	-73.9444	B02512	Monday	1	0	11	9

In [194…

```
heatmap('Day','months')
```

Out[194… <AxesSubplot:xlabel='months', ylabel='Day'>



In [195…

```
df.head()
```

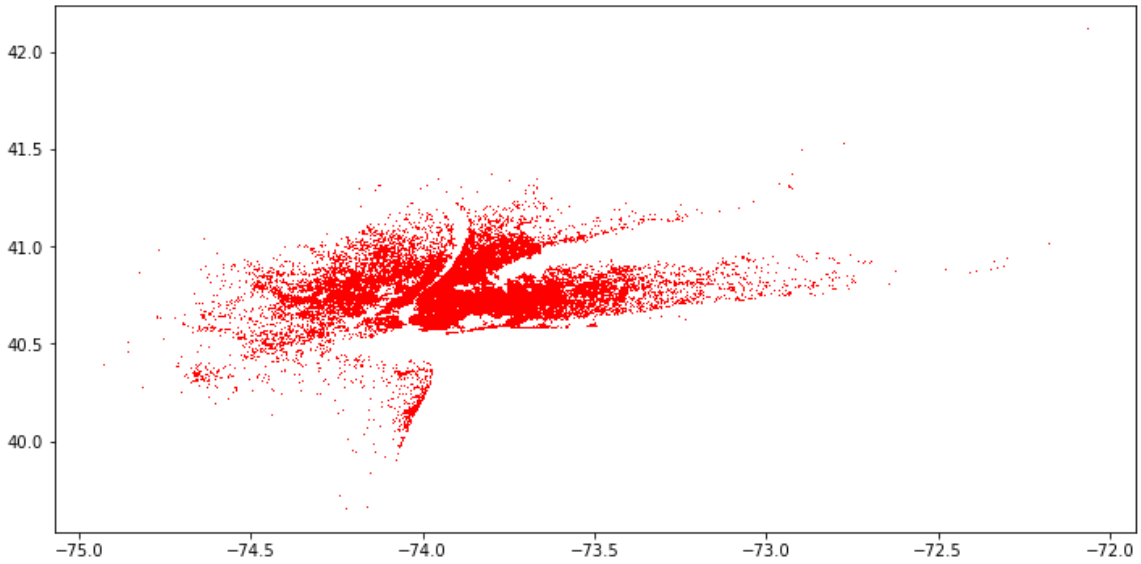
Out[195…

	Date/Time	Lat	Lon	Base	Weekdays	Day	Hours	Minute	months
0	2014-09-01 00:01:00	40.2201	-74.0021	B02512	Monday	1	0	1	9
1	2014-09-01 00:01:00	40.7500	-74.0027	B02512	Monday	1	0	1	9
2	2014-09-01 00:03:00	40.7559	-73.9864	B02512	Monday	1	0	3	9
3	2014-09-01 00:06:00	40.7450	-73.9889	B02512	Monday	1	0	6	9
4	2014-09-01 00:11:00	40.8145	-73.9444	B02512	Monday	1	0	11	9

In [204…

```
plt.figure(figsize=(12,6))
plt.plot(df['Lon'], df['Lat'],'r+', ms=0.5)
plt.xlim(-74.2, -73.7)
plt.ylim(40.6,41)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-204-84254a5631e4> in <module>
      1 plt.figure(figsize=(12,6))
      2 plt.plot(df['Lon'], df['Lat'],'r+', ms=0.5)
----> 3 plt.xlim(-74.2, -73.7)
      4 plt.ylim(40.6,41)

TypeError: 'tuple' object is not callable
```



In [208…

```
df_out = df[df['Weekdays']=='Sunday']
df_out
```

Out[208…

	Date/Time	Lat	Lon	Base	Weekdays	Day	Hours	Minute	months
8011	2014-09-07 00:00:00	40.7341	-74.0005	B02512	Sunday	7	0	0	9
8012	2014-09-07 00:00:00	40.7344	-73.9900	B02512	Sunday	7	0	0	9
8013	2014-09-07 00:00:00	40.7806	-73.9582	B02512	Sunday	7	0	0	9
8014	2014-09-07 00:01:00	40.7293	-73.9859	B02512	Sunday	7	0	1	9
8015	2014-09-07 00:01:00	40.7713	-74.0133	B02512	Sunday	7	0	1	9
...	...	...	...	...	...	...	...	...	...
563273	2014-04-27 22:59:00	40.7741	-73.8725	B02764	Sunday	27	22	59	4
563274	2014-04-	40.7782	-73.9625	B02764	Sunday	27	23	23	4

	27 23:23:00								
563275	2014-04-27 23:33:00	40.6449	-73.7823	B02764	Sunday	27	23	33	4
563276	2014-04-27 23:35:00	40.7278	-73.9822	B02764	Sunday	27	23	35	4
563277	2014-04-27 23:41:00	40.6879	-74.1813	B02764	Sunday	27	23	41	4

490180 rows × 9 columns

In [214…
```
rush = df.groupby(['Lat','Lon'])['Weekdays'].count().reset_index()
```

In [215…
```
!pip install folium
```

```
Collecting folium
  Downloading folium-0.12.1-py2.py3-none-any.whl (94 kB)
Requirement already satisfied: numpy in c:\users\admin\anaconda3\lib\site
-packages (from folium) (1.20.1)
Requirement already satisfied: requests in c:\users\admin\anaconda3\lib\s
ite-packages (from folium) (2.25.1)
Requirement already satisfied: jinja2>=2.9 in c:\users\admin\anaconda3\li
b\site-packages (from folium) (2.11.3)
Collecting branca>=0.3.0
  Downloading branca-0.4.2-py3-none-any.whl (24 kB)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\admin\anacond
a3\lib\site-packages (from jinja2>=2.9->folium) (1.1.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\admin\an
aconda3\lib\site-packages (from requests->folium) (1.26.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\admin\anaco
nda3\lib\site-packages (from requests->folium) (2020.12.5)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\admin\anacon
da3\lib\site-packages (from requests->folium) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in c:\users\admin\anaconda3\l
ib\site-packages (from requests->folium) (2.10)
Installing collected packages: branca, folium
Successfully installed branca-0.4.2 folium-0.12.1
```

In [217…
```
from folium.plugins import HeatMap
```

In [218…
```
import folium
from folium.plugins import HeatMap
basemap=folium.Map()
```

In [241…
```
folium.Map()
```

Out[241… Make this Notebook Trusted to load map: File -> Trust Notebook

In [244…

```
def plot(df,day):
    df_out=df[df['Weekdays']==day]
    df_out.groupby(['Lat','Lon'])['Weekdays'].count().reset_index()
    HeatMap(df_out.groupby(['Lat','Lon'])['Weekdays'].count().reset_inde
    return basemap
```

In [245…

```
plot(df,'Sunday')
```

Out[245… Make this Notebook Trusted to load map: File -> Trust Notebook

In [263…

```
from pathlib import Path
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [283…
```
files=os.listdir(r'C:\Users\Admin\Desktop\New folder')[-5:15]
```

In [284…
```
files
```

Out[284…
```
['uber-raw-data-janjune-15 - Copy.csv']
```

In [ ]:
```

```

In [296…
```
files_new=os.listdir(r'C:\Users\Admin\Downloads\New folder')[-5:15]
```

In [297…
```
files_new
```

Out[297…
```
['uber-raw-data-janjune-15 - Copy.csv']
```

In [304…
```
path = r'C:\Users\Admin\Downloads\New folder'

#blank dataframe
finals=pd.DataFrame()

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

In [302…
```
finals.head()
```

Out[302…

	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

In [305…
```
finals.shape
```

Out[305…
```
(14270479, 4)
```

In [306…
```
finals['Pickup_date'].min()
```

Out[306…
```
'2015-01-01 00:00:05'
```

In [307…
```
finals['Pickup_date'] = pd.to_datetime(finals['Pickup_date'], format = '
```

In [308…
```
finals.dtypes
```

Out[308…
```
Dispatching_base_num            object
Pickup_date                datetime64[ns]
Affiliated_base_num             object
locationID                       int64
dtype: object
```

In [313…
```
finals['weekday']= finals['Pickup_date'].dt.day_name()
finals['day']= finals['Pickup_date'].dt.day
finals['month'] = finals['Pickup_date'].dt.month
finals['hour'] = finals['Pickup_date'].dt.hour
finals['minute'] = finals['Pickup_date'].dt.minute
```

In [314…
```
finals.head()
```

Out[314…

	Dispatching_base_num	Pickup_date	Affiliated_base_num	locationID	weekday	day	mon
0	B02617	2015-05-17 09:47:00	B02617	141	Sunday	17	
1	B02617	2015-05-17 09:47:00	B02617	65	Sunday	17	
2	B02617	2015-05-17 09:47:00	B02617	100	Sunday	17	
3	B02617	2015-05-17 09:47:00	B02774	80	Sunday	17	
4	B02617	2015-05-17 09:47:00	B02617	90	Sunday	17	

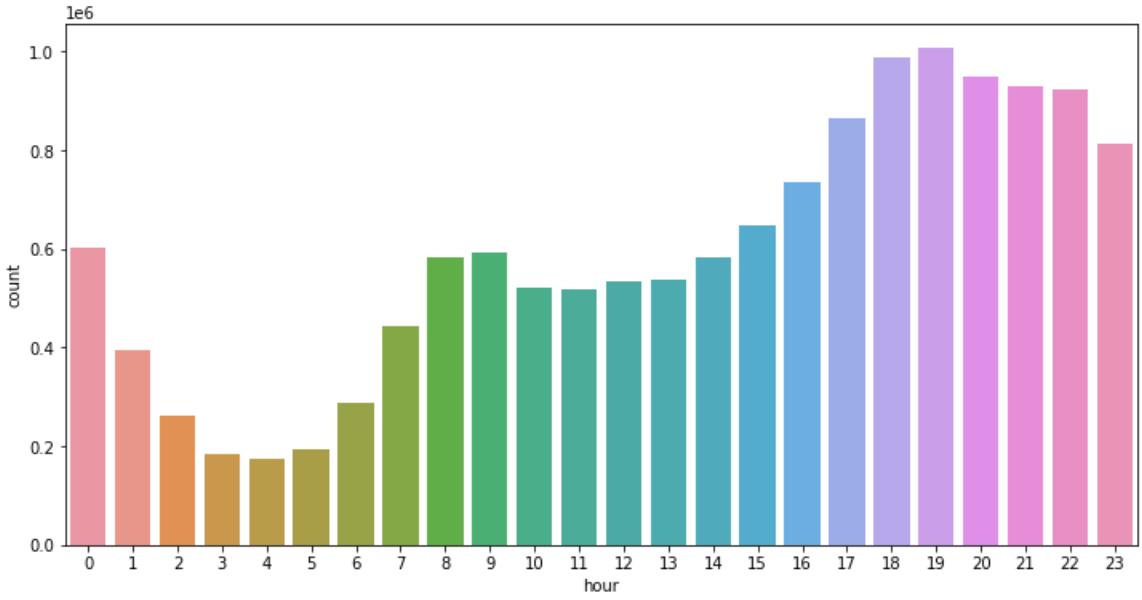
In [324…
```
finals['month'].value_counts().index
```

Out[324…
```
Int64Index([6, 5, 4, 2, 3, 1], dtype='int64')
```

In [332…
```
plt.figure(figsize=(12,6))
sns.countplot(finals['hour'])
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fut
ureWarning:

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

Out[332…
```
<AxesSubplot:xlabel='hour', ylabel='count'>
```



In [335…
```
summary=finals.groupby(['weekday','hour'])['Pickup_date'].count()
```

In [361…
```
summary=finals.groupby(['weekday','hour'])['Pickup_date'].count().reset_
```

In [362…
```
summary.head()
```

Out[362…

	weekday	hour	Pickup_date
0	Friday	0	85939
1	Friday	1	46616
2	Friday	2	28102
3	Friday	3	19518
4	Friday	4	23575

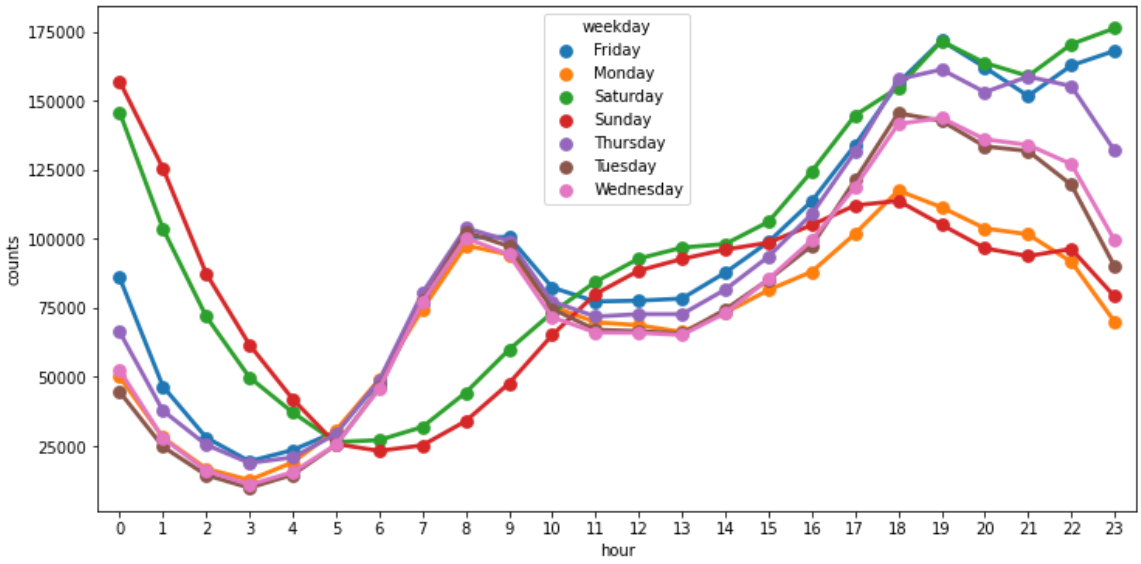
In [365…
```
summary.columns=('weekday','hour','counts')
summary.head()
```

Out[365…

	weekday	hour	counts
0	Friday	0	85939
1	Friday	1	46616
2	Friday	2	28102
3	Friday	3	19518
4	Friday	4	23575

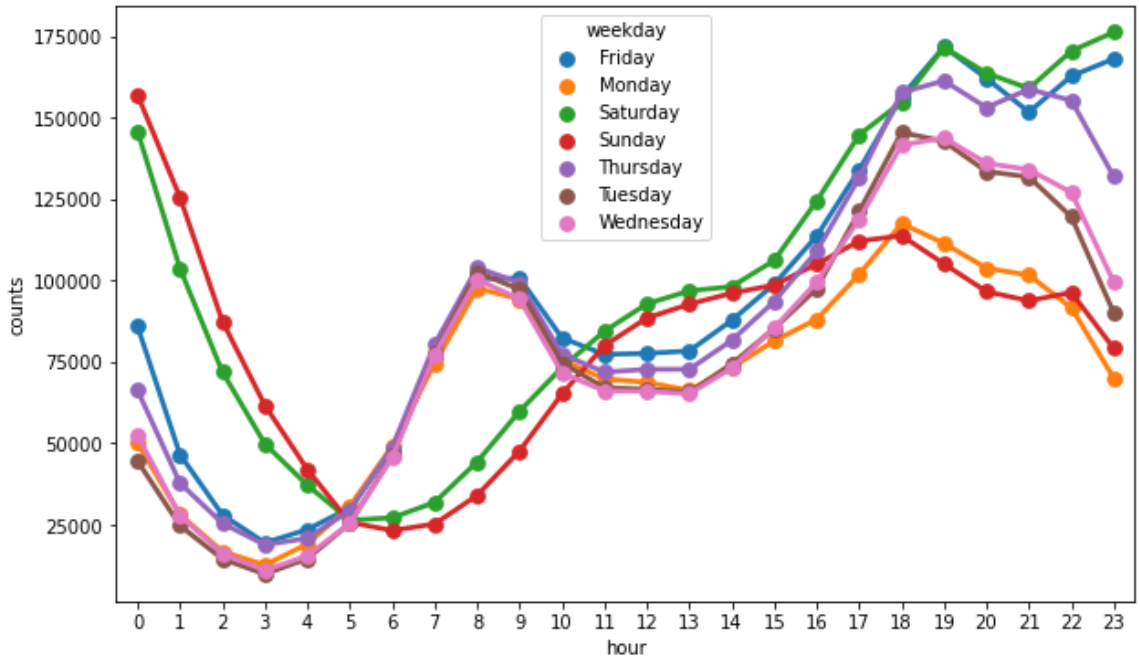
In [366…
```
plt.figure(figsize=(12,6))
sns.pointplot(x='hour',y='counts',hue='weekday',data=summary)
```

Out[366… `<AxesSubplot:xlabel='hour', ylabel='counts'>`



In [369…
```
plt.figure(figsize=(10,6))
sns.pointplot(x="hour", y="counts", hue="weekday", data=summary)
```

Out[369…
```
<AxesSubplot:xlabel='hour', ylabel='counts'>
```



In [373…
```
from pathlib import Path
```

In [376…
```
import pandas as pd
```

In [ ]:
```

```

In [380…
```
files_new=os.listdir(r'C:\Users\Admin\Downloads\New folder')[-8:12]
```

In [383…
```
files_new
```

Out[383… ['Uber-Jan-Feb-FOIL.csv']

In [386…
```
path = r'C:\Users\Admin\Downloads\New folder'

#blank dataframe
uber_foil=pd.DataFrame()

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

In [ ]:

In [387…
```
uber_foil.head()
```

Out[387…

	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 [388…
```
uber_foil.shape
```

Out[388… (354, 4)

In [389…
```
uber_foil
```

Out[389…

	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
...	...	...	...	...
349	B02764	2/28/2015	3952	39812
350	B02617	2/28/2015	1372	14022
351	B02682	2/28/2015	1386	14472
352	B02512	2/28/2015	230	1803
353	B02765	2/28/2015	747	7753

354 rows × 4 columns

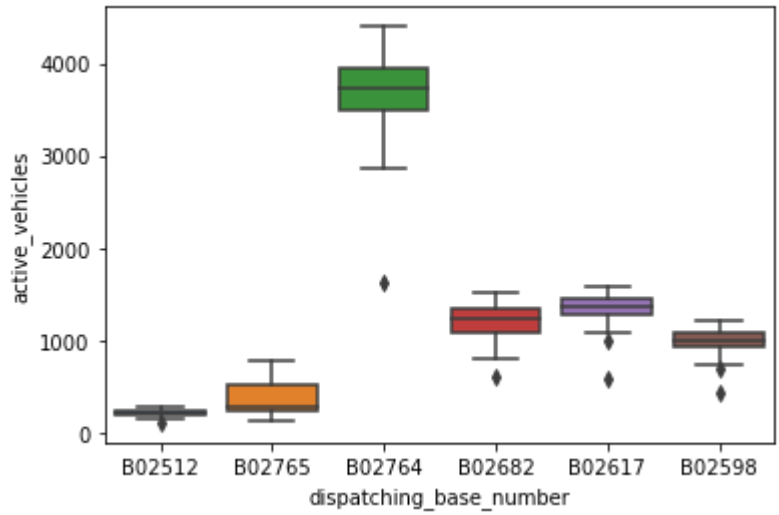
In [ ]:

In [390…
```
uber_foil['dispatching_base_number'].unique()
```

Out[390…
```
array(['B02512', 'B02765', 'B02764', 'B02682', 'B02617', 'B02598'],
      dtype=object)
```

In [394…
```
sns.boxplot(x='dispatching_base_number', y= 'active_vehicles', data= ube
```

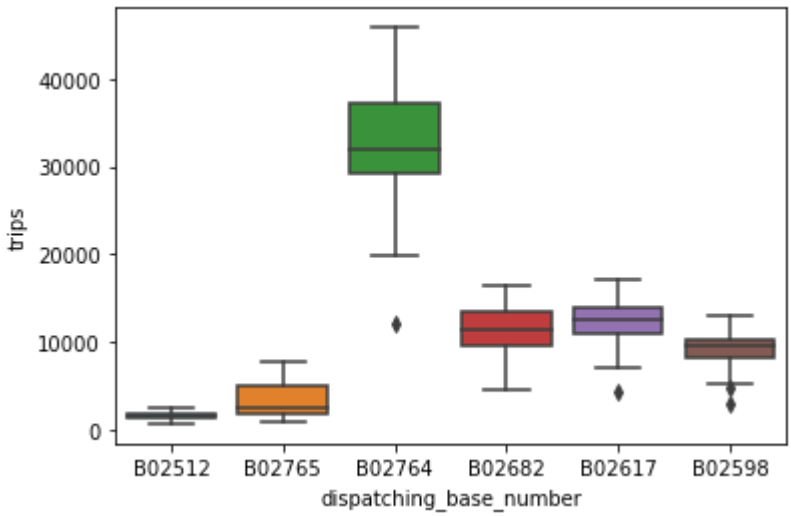
Out[394…
```
<AxesSubplot:xlabel='dispatching_base_number', ylabel='active_vehicles'>
```



In [ ]:
```
```

In [395…
```
sns.boxplot(x = 'dispatching_base_number', y = 'trips', data = uber_foil
```

Out[395…
```
<AxesSubplot:xlabel='dispatching_base_number', ylabel='trips'>
```



In [396…
```
uber_foil['trips/vehicle']= uber_foil['trips']/uber_foil['active_vehicle
uber_foil.head()
```

Out[396…

	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

In [399…

```
uber_foil.set_index('date')
```

Out[399…

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 × 4 columns

In [ ]:

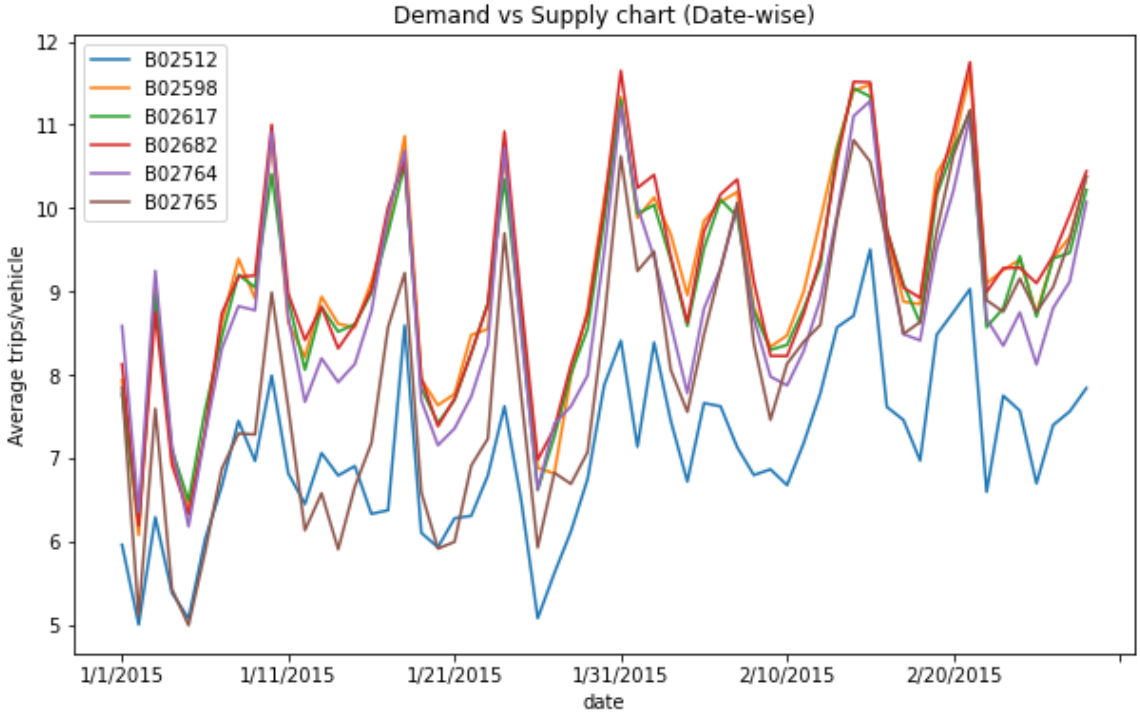
```
```

In [402…

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

Out[402…

```
<matplotlib.legend.Legend at 0x1cc0b3da490>
```


Demand vs Supply chart (Date-wise)
B02512
B02598
B02617
B02682
B02764
B02765
Average trips/vehicle
12
11
10
9
8
7
6
5
1/1/2015
1/11/2015
1/21/2015
1/31/2015
2/10/2015
2/20/2015
date

In [ ]: