

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

To read/get the CSV file The 'r' is important to avoid error in the below code utf is not important Save it in something u like

```
uber_23=pd.read_csv(r'C:\Users\Aakasha\Desktop\uber-pickups-in-new-york-city-20220814T121250Z-001\uber-pickups-in-new-york-city/uber-raw-data-janjune-15.csv',encoding='utf-8')
```

uber\_23

	Dispatching_base_num	Pickup_date	Affiliated_base_num
\			
0	B02617	2015-05-17 09:47:00	B02617
1	B02617	2015-05-17 09:47:00	B02617
2	B02617	2015-05-17 09:47:00	B02617
3	B02617	2015-05-17 09:47:00	B02774
4	B02617	2015-05-17 09:47:00	B02617
...	...	...	...
14270474	B02765	2015-05-08 15:43:00	B02765
14270475	B02765	2015-05-08 15:43:00	B02765
14270476	B02765	2015-05-08 15:43:00	B02765
14270477	B02765	2015-05-08 15:44:00	B01899
14270478	B02765	2015-05-08 15:44:00	B02682

	locationID
0	141
1	65
2	100
3	80
4	90
...	...
14270474	186
14270475	263
14270476	90
14270477	45

14270478                      144

[14270479 rows x 4 columns]

1D is series that is Array 2D is Data that is Table 3D is Panel mostly used

To see only 1st 2 rows

```
uber_23.head(2)
```

	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			

To count the No.of duplicates

```
uber_23.duplicated().sum()
```

898225

To remove the Duplicate Rows

```
uber_23.drop_duplicates(inplace=True)
```

Take a look at 'inplace' in Drop\_duplicates

```
uber_23.shape
```

(13372254, 4)

Problem: Which month has max uber pickups Ans:

1. We have to extract the month from the pickup\_date column in the above table
2. Or we have change the data type of the pickup\_date from object to below given

```
uber_23['Pickup_date']=pd.to_datetime(uber_23['Pickup_date'],format='%Y-%m-%d %H:%M:%S')
```

```
uber_23['Pickup_date'].dtype
```

```
dt=datetime/daytime
```

```
month_count=uber_23['Pickup_date'].dt.month
```

```
month_count.value_counts()
```

To Represent in graph

```
month_count.value_counts().plot(kind='bar')
```

Now, We have to find the No.of Trips in Month and Week Days

```
uber_23['month']=uber_23['Pickup_date'].dt.month
uber_23['week_day']=uber_23['Pickup_date'].dt.weekday
uber_23['day_name']=uber_23['Pickup_date'].dt.day_name
uber_23['hour']=uber_23['Pickup_date'].dt.hour
uber_23['minute']=uber_23['Pickup_date'].dt.minute
```

```
uber_23.head(2)
```

```
uber_23.groupby(['month','week_day']).size()
```

month	week_day	
1	0	190606
	1	196574
	2	245650
	3	330319
	4	339285
	5	386049
	6	230487
2	0	274948
	1	287260
	2	286387
	3	335603
	4	373550
	5	368311
	6	296130
3	0	269931
	1	320634
	2	256767
	3	277026
	4	309631
	5	314785
	6	313865
4	0	238429
	1	250632
	2	338015
	3	372522
	4	315002
	5	324545
	6	273560
5	0	255501
	1	290004
	2	316045
	3	337607
	4	430134
	5	464298
	6	390391
6	0	375312
	1	405500
	2	328141
	3	357782
	4	371225

```
5          399377
6          334434
dtype: int64
```

To convert the upper one into DATA FRAME using as\_index= False

To see the unique month

```
x=uber_23.groupby(['month','week_day'],as_index=False).size()
x.head(2)
```

	month	week_day	size
0	1	0	190606
1	1	1	196574

To change the month number to text:-

```
x['month'].unique()
array([1, 2, 3, 4, 5, 6], dtype=int64)
x.head(5)
```

	month	week_day	size
0	1	0	190606
1	1	1	196574
2	1	2	245650
3	1	3	330319
4	1	4	339285

```
x_dict={1:'JAN',2:'FEB',3:'MAR',4:'APR',5:'MAY',6:'JUNE'}
```

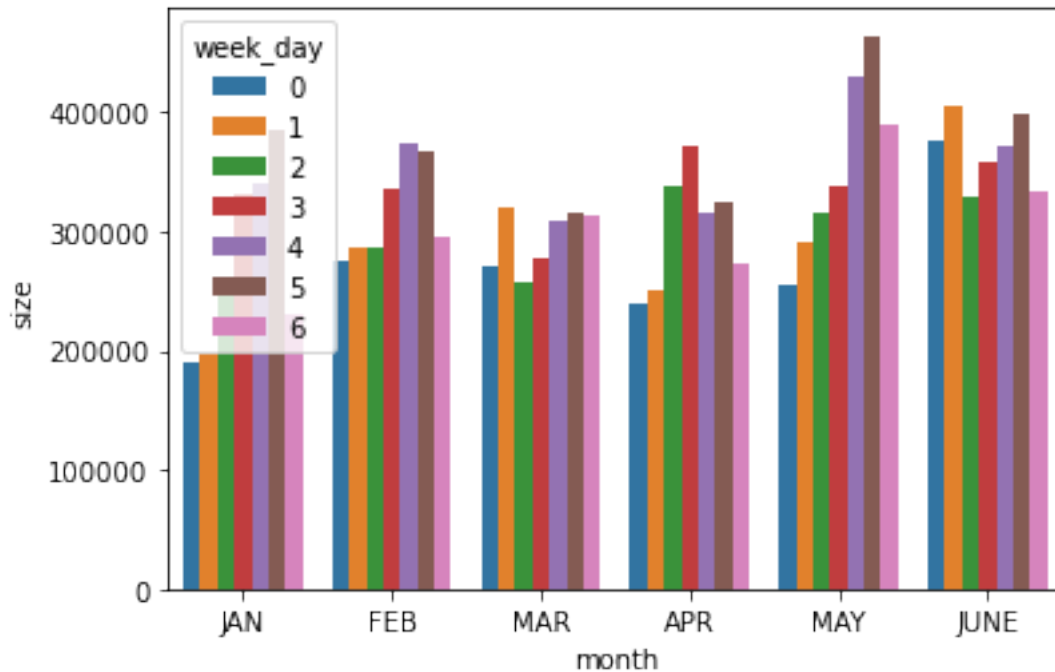
```
x['month']=x['month'].map(x_dict)
x.head(4)
```

	month	week_day	size
0	JAN	0	190606
1	JAN	1	196574
2	JAN	2	245650
3	JAN	3	330319

To plot this Data

```
sns.barplot(x='month',y='size',hue='week_day',data=x)
plt.figure(figsize=(12,8))
```

<Figure size 864x576 with 0 Axes>



<Figure size 864x576 with 0 Axes>

PROBLEM:-

To find the rush of the New York city:- No.of rides per hour

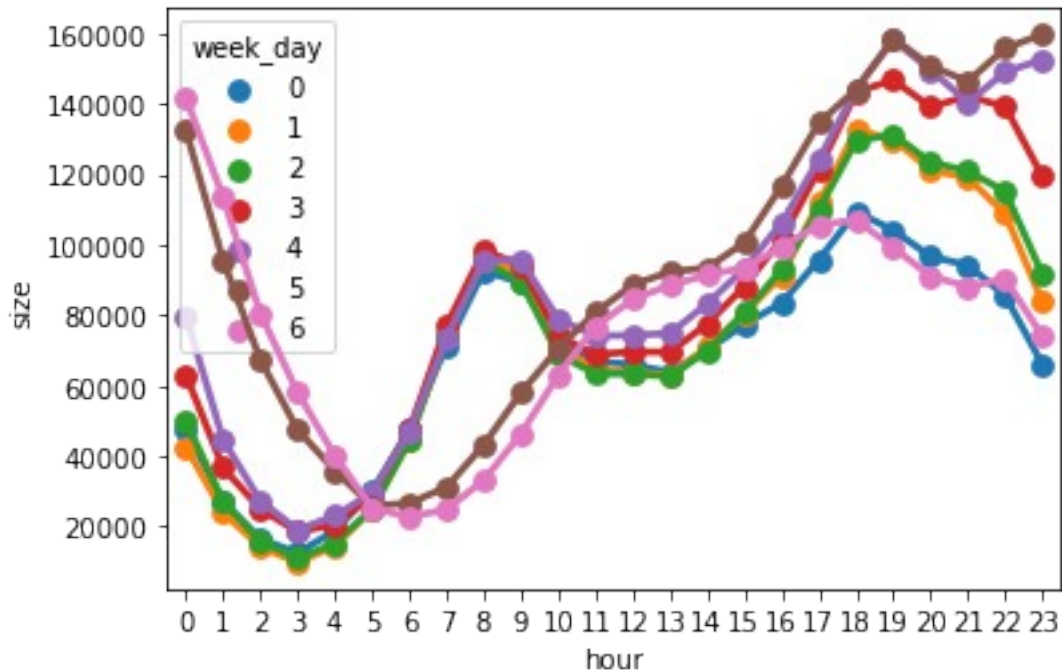
```
y=uber_23.groupby(['week_day','hour'],as_index=False).size()
y.head(3)
```

	week_day	hour	size
0	0	0	47608
1	0	1	27093
2	0	2	16394

If we use Bar Graph, The chart is messy and full of disturbance So, we use POINT PLOT

```
sns.pointplot(x='hour',y='size',hue='week_day',data=y)
plt.figure(figsize=(12,8))
```

<Figure size 864x576 with 0 Axes>



<Figure size 864x576 with 0 Axes>

NOW, OBSERVE THE FIGURE CAREFULLY AND SAY, AT WHAT STAGES THE TRIPS ARE HIGH AND LOW DURING WHICH THE DAYS ACCORDING TO THE HOURS FROM 1-24

NOW,

WE NEED TO FIND, WHICH BASE NUMBER HAS THE MOST NUMBER OF ACTIVE VEHICLES...

SO, WE NEED BRING THE NEW FILE...

```
uber_foil=pd.read_csv(r'C:\Users\Aakasha\Desktop\uber-pickups-in-new-york-city-20220626T112836Z-001\uber-pickups-in-new-york-city\Uber-Jan-Feb-FOIL.csv')
```

```
uber_foil.head(3)
```

	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

NOW, BY SEEING WE CAN EASILY SAY THE MAX ACTIVE\_NUMBER. SO, WE WILL EXPAND THIS NOW,

WE NEED TO FIND THE DISTRIBUTION OF THE BASE NUMBER THROUGH GRAPHS

WE NEED TO RUN BELOW CODES FOR FEW INSTALLATION

```
!pip install chart_studio
!pip install plotly
```

```
Requirement already satisfied: chart_studio in c:\python folder\
anaconda aakash\lib\site-packages (1.1.0)
Requirement already satisfied: requests in c:\python folder\anaconda
aakash\lib\site-packages (from chart_studio) (2.25.1)
Requirement already satisfied: six in c:\python folder\anaconda
aakash\lib\site-packages (from chart_studio) (1.15.0)
Requirement already satisfied: plotly in c:\python folder\anaconda
aakash\lib\site-packages (from chart_studio) (5.10.0)
Requirement already satisfied: retrying>=1.3.3 in c:\python folder\
anaconda aakash\lib\site-packages (from chart_studio) (1.3.3)
Requirement already satisfied: tenacity>=6.2.0 in c:\python folder\
anaconda aakash\lib\site-packages (from plotly->chart_studio) (8.0.1)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\python folder\
anaconda aakash\lib\site-packages (from requests->chart_studio)
(4.0.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\python
folder\anaconda aakash\lib\site-packages (from requests->chart_studio)
(1.26.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\python folder\
anaconda aakash\lib\site-packages (from requests->chart_studio)
(2020.12.5)
Requirement already satisfied: idna<3,>=2.5 in c:\python folder\
anaconda aakash\lib\site-packages (from requests->chart_studio) (2.10)
Requirement already satisfied: plotly in c:\python folder\anaconda
aakash\lib\site-packages (5.10.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\python folder\
anaconda aakash\lib\site-packages (from plotly) (8.0.1)
```

```
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)
```

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

```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"alignmentgroup":"True","hovertemplate":"dispatching_base_number=%
{x}<br>active_vehicles=%{y}<extra></extra>","legendgroup":"","marker":
{"color":"#636efa"},"name":"","notched":false,"offsetgroup":"","orient
ation":"v","showlegend":false,"type":"box","x":
["B02512","B02765","B02764","B02682","B02617","B02598","B02598","B0261
7","B02512","B02682","B02765","B02764","B02765","B02617","B02598","B02
682","B02512","B02764","B02512","B02682","B02598","B02765","B02617","B
02764","B02512","B02682","B02617","B02764","B02598","B02765","B02764",
"B02682","B02617","B02765","B02512","B02598","B02617","B02682","B02764
```

[illegible]



```
,211,1367,1223,245,183,3300,226,829,1046,1203,1150,860,230,3012,1084,1
97,600,135,596,434,112,1619,3692,1235,286,1356,1011,235,1474,3959,1316
,295,250,1082,256,1501,1384,4124,322,1106,309,225,1394,1321,3947,1027,
961,1214,193,289,1355,3740,1217,1152,275,939,3270,227,299,991,257,3674
,1350,1269,3856,309,244,1311,1393,1072,1524,1418,1179,264,355,4093,152
6,385,1181,261,4170,1414,1031,211,1383,1300,3849,345,3422,313,923,1256
,1136,176,1312,1241,976,3543,228,388,3700,233,1364,422,1281,1029,1450,
3849,255,1115,450,1396,1532,269,1468,536,1181,4137,1590,1523,4395,599,
281,1216,4129,236,1111,583,1486,1428,1261,3651,1293,521,210,1003,934,2
07,1214,3524,1164,508,3826,241,1314,1378,1066,578,1078,1314,1394,586,3
842,228,1127,250,1428,4110,663,1452,4384,1574,1186,1497,736,272,1044,1
374,685,1443,238,3981,199,1248,1220,3478,909,566,966,1332,3734,1262,66
5,238,3965,247,1061,1346,1456,698,246,1076,706,1395,1473,3934,1134,153
9,1465,243,745,4101,786,1551,1114,272,4253,1510,994,3952,1372,1386,230
,747],"y0":" ","yaxis":"y"}],"layout":{"boxmode":"group","legend":
{"tracegroupgap":0},"margin":{"t":60},"template":{"data":{"bar":
[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":"","type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":"","colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
```

```

{"fillmode":"overlay","size":10,"solidity":0.2}},{"type":"histogram"}]],
"histogram2d":[{"colorbar":{"linewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]]],"type":"histogram2d"}]],"histogram2dcontour":
[{"colorbar":{"linewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]]],"type":"histogram2dcontour"}]],"mesh3d":[{"colorbar":
{"linewidth":0,"ticks":"","type":"mesh3d"}]],"parcoords":[{"line":
{"colorbar":{"linewidth":0,"ticks":"","type":"parcoords"}]],"pie":
[{"automargin":true,"type":"pie"}]],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}]],"scatter3d":[{"line":{"colorbar":{"linewidth":0,"ticks":"","type":"scatter3d"}]],"scattercarpet":
[{"marker":{"colorbar":
{"linewidth":0,"ticks":"","type":"scattercarpet"}]],"scattergeo":
[{"marker":{"colorbar":
{"linewidth":0,"ticks":"","type":"scattergeo"}]],"scattergl":
[{"marker":{"colorbar":
{"linewidth":0,"ticks":"","type":"scattergl"}]],"scattermapbox":
[{"marker":{"colorbar":
{"linewidth":0,"ticks":"","type":"scattermapbox"}]],"scatterpolar":
[{"marker":{"colorbar":
{"linewidth":0,"ticks":"","type":"scatterpolar"}]],"scatterpolargl":
[{"marker":{"colorbar":
{"linewidth":0,"ticks":"","type":"scatterpolargl"}]],"scatterternary":
[{"marker":{"colorbar":
{"linewidth":0,"ticks":"","type":"scatterternary"}]],"surface":
[{"colorbar":{"linewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]]],"type":"surface"}]],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"},"type":"table"}]],"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers":
"strict","coloraxis":{"colorbar":
{"linewidth":0,"ticks":"","colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],

```

```

[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fb341"],[0.9,"#4d9221"],[1,"#276419"]], "sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]], "sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]], "colorway":
["#636efa", "#EF553B", "#00cc96", "#ab63fa", "#FFA15A", "#19d3f3", "#FF6692",
"#B6E880", "#FF97FF", "#FECB52"], "font": {"color": "#2a3f5f"}, "geo":
{"bgcolor": "white", "lakecolor": "white", "landcolor": "#E5ECF6", "showlake
s": true, "showland": true, "subunitcolor": "white"}, "hoverlabel":
{"align": "left"}, "hovermode": "closest", "mapbox":
{"style": "light"}, "paper_bgcolor": "white", "plot_bgcolor": "#E5ECF6", "po
lar": {"angularaxis":
{"gridcolor": "white", "linecolor": "white", "ticks": ""}, "bgcolor": "#E5ECF
6", "radialaxis":
{"gridcolor": "white", "linecolor": "white", "ticks": ""}}, "scene":
{"xaxis":
{"backgroundcolor": "#E5ECF6", "gridcolor": "white", "gridwidth": 2, "lineco
lor": "white", "showbackground": true, "ticks": "", "zerolinecolor": "white"},
"yaxis":
{"backgroundcolor": "#E5ECF6", "gridcolor": "white", "gridwidth": 2, "lineco
lor": "white", "showbackground": true, "ticks": "", "zerolinecolor": "white"},
"zaxis":
{"backgroundcolor": "#E5ECF6", "gridcolor": "white", "gridwidth": 2, "lineco
lor": "white", "showbackground": true, "ticks": "", "zerolinecolor": "white"}
}, "shapedefaults": {"line": {"color": "#2a3f5f"}}, "ternary": {"aaxis":
{"gridcolor": "white", "linecolor": "white", "ticks": ""}, "baxis":
{"gridcolor": "white", "linecolor": "white", "ticks": ""}, "bgcolor": "#E5ECF
6", "caxis":
{"gridcolor": "white", "linecolor": "white", "ticks": ""}}, "title":
{"x": 5.0e-2, "xaxis":
{"automargin": true, "gridcolor": "white", "linecolor": "white", "ticks": "",
"title":
{"standoff": 15}, "zerolinecolor": "white", "zerolinewidth": 2}, "yaxis":
{"automargin": true, "gridcolor": "white", "linecolor": "white", "ticks": "",
"title":
{"standoff": 15}, "zerolinecolor": "white", "zerolinewidth": 2}}}, "xaxis":
{"anchor": "y", "domain": [0, 1], "title":
{"text": "dispatching_base_number"}}, "yaxis": {"anchor": "x", "domain":
[0, 1], "title": {"text": "active_vehicles"}}}]

```

VIOLIN CHART IS ALSO USEFUL FOR THE DISTRIBUTION

```

px.violin(x='dispatching_base_number', y='active_vehicles', data_frame=u
ber_foil)

```

```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":  
[{"alignmentgroup":"True","box":  
{  
"visible":false,"hvertemplate":"dispatching_base_number=%  
{x}<br>active_vehicles=%{y}<extra></extra>","legendgroup":"","marker":  
{  
"color":"#636efa"},"name":"","offsetgroup":"","orientation":"v","scal  
egroup":"True","showlegend":false,"type":"violin","x":  
["B02512","B02765","B02764","B02682","B02617","B02598","B02598","B0261  
7","B02512","B02682","B02765","B02764","B02765","B02617","B02598","B02  
682","B02512","B02764","B02512","B02682","B02598","B02765","B02617","B  
02764","B02512","B02682","B02617","B02764","B02598","B02765","B02764",  
"B02682","B02617","B02765","B02512","B02598","B02617","B02682","B02764  
","B02765","B02512","B02598","B02765","B02598","B02512","B02682","B027  
64","B02617","B02617","B02512","B02764","B02682","B02598","B02765","B0  
2682","B02617","B02598","B02512","B02764","B02765","B02765","B02598","  
B02682","B02764","B02617","B02512","B02764","B02765","B02512","B02598"  
,"B02682","B02617","B02765","B02598","B02617","B02512","B02764","B0268  
2","B02764","B02765","B02598","B02512","B02617","B02682","B02512","B02  
682","B02617","B02765","B02764","B02598","B02617","B02765","B02764","B  
02682","B02512","B02598","B02598","B02512","B02682","B02765","B02617",  
"B02764","B02512","B02598","B02765","B02764","B02682","B02617","B02682  
","B02617","B02765","B02764","B02512","B02598","B02598","B02682","B025  
12","B02764","B02765","B02617","B02764","B02512","B02682","B02598","B0  
2765","B02617","B02617","B02764","B02512","B02598","B02682","B02765","  
B02598","B02512","B02765","B02764","B02617","B02682","B02598","B02764"  
,"B02512","B02617","B02682","B02765","B02512","B02764","B02765","B0259  
8","B02682","B02617","B02617","B02598","B02765","B02764","B02682","B02  
512","B02682","B02765","B02617","B02598","B02512","B02764","B02764","B  
02682","B02765","B02617","B02598","B02512","B02617","B02764","B02682",  
"B02765","B02512","B02598","B02512","B02617","B02682","B02764","B02765  
","B02598","B02765","B02512","B02617","B02682","B02764","B02598","B025  
98","B02682","B02512","B02765","B02617","B02764","B02617","B02682","B0  
2765","B02598","B02764","B02512","B02765","B02598","B02512","B02764","  
B02617","B02682","B02764","B02765","B02512","B02682","B02617","B02598"  
,"B02617","B02682","B02598","B02512","B02765","B02764","B02617","B0276  
5","B02598","B02512","B02764","B02682","B02598","B02512","B02617","B02  
682","B02764","B02765","B02764","B02765","B02598","B02617","B02682","B  
02512","B02617","B02682","B02598","B02764","B02512","B02765","B02764",  
"B02512","B02617","B02765","B02682","B02598","B02617","B02764","B02512  
","B02598","B02765","B02682","B02617","B02512","B02682","B02765","B025  
98","B02764","B02617","B02682","B02764","B02765","B02512","B02598","B0  
2764","B02512","B02598","B02765","B02617","B02682","B02682","B02764","  
B02617","B02765","B02512","B02598","B02598","B02512","B02617","B02764"  
,"B02682","B02765","B02764","B02512","B02682","B02617","B02598","B0276  
5","B02598","B02682","B02617","B02765","B02764","B02512","B02598","B02  
512","B02682","B02764","B02765","B02617","B02764","B02617","B02598","B  
02682","B02765","B02512","B02598","B02682","B02765","B02617","B02512",  
"B02764","B02512","B02617","B02682","B02764","B02598","B02765","B02598  
","B02617","B02764","B02682","B02765","B02512","B02764","B02512","B025  
98","B02682","B02617","B02765","B02512","B02598","B02765","B02682","B0  
2617","B02764","B02598","B02617","B02682","B02512","B02765","B02764",""
```

```

B02765", "B02617", "B02598", "B02512", "B02764", "B02682", "B02598", "B02764"
, "B02617", "B02682", "B02512", "B02765" ], "x0": " ", "xaxis": "x", "y":
[190,225,3427,945,1228,870,785,1137,175,890,196,3147,201,1188,818,915,
173,3215,147,812,746,183,1088,2862,194,951,1218,3387,907,227,3473,1022
,1336,234,218,933,1363,1039,3603,248,217,974,262,1070,238,1135,3831,14
63,1455,224,3820,1140,1070,280,1057,1331,949,206,3558,245,220,832,943,
3186,1228,162,3499,279,217,964,1082,1323,258,975,1342,234,3658,1092,37
36,271,1030,233,1405,1174,237,1208,1457,270,3840,1068,1445,290,3975,12
50,234,1079,974,201,1137,252,1306,3657,177,869,248,3290,1056,1223,883,
992,238,2958,168,706,944,1151,221,3654,272,1350,3718,242,1228,1035,296
,1429,1471,3889,246,1071,1295,295,1093,246,299,4040,1482,1330,945,3652
,211,1367,1223,245,183,3300,226,829,1046,1203,1150,860,230,3012,1084,1
97,600,135,596,434,112,1619,3692,1235,286,1356,1011,235,1474,3959,1316
,295,250,1082,256,1501,1384,4124,322,1106,309,225,1394,1321,3947,1027,
961,1214,193,289,1355,3740,1217,1152,275,939,3270,227,299,991,257,3674
,1350,1269,3856,309,244,1311,1393,1072,1524,1418,1179,264,355,4093,152
6,385,1181,261,4170,1414,1031,211,1383,1300,3849,345,3422,313,923,1256
,1136,176,1312,1241,976,3543,228,388,3700,233,1364,422,1281,1029,1450,
3849,255,1115,450,1396,1532,269,1468,536,1181,4137,1590,1523,4395,599,
281,1216,4129,236,1111,583,1486,1428,1261,3651,1293,521,210,1003,934,2
07,1214,3524,1164,508,3826,241,1314,1378,1066,578,1078,1314,1394,586,3
842,228,1127,250,1428,4110,663,1452,4384,1574,1186,1497,736,272,1044,1
374,685,1443,238,3981,199,1248,1220,3478,909,566,966,1332,3734,1262,66
5,238,3965,247,1061,1346,1456,698,246,1076,706,1395,1473,3934,1134,153
9,1465,243,745,4101,786,1551,1114,272,4253,1510,994,3952,1372,1386,230
,747], "y0": " ", "yaxis": "y"}], "layout": {"legend":
{"tracegroupgap":0}, "margin":{"t":60}, "template":{"data":{"bar":
[{"error_x":{"color":"#2a3f5f"}, "error_y":
{"color":"#2a3f5f"}, "marker":{"line":
{"color":"#E5ECF6", "width":0.5}, "pattern":
{"fillmode":"overlay", "size":10, "solidity":0.2}}, "type":"bar"}], "barpo
lar":[{"marker":{"line":{"color":"#E5ECF6", "width":0.5}, "pattern":
{"fillmode":"overlay", "size":10, "solidity":0.2}}, "type":"barpolar"}], "
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f", "gridcolor":"white", "linecolor":"white", "min
orgridcolor":"white", "startlinecolor":"#2a3f5f"}, "baxis":
{"endlinecolor":"#2a3f5f", "gridcolor":"white", "linecolor":"white", "min
orgridcolor":"white", "startlinecolor":"#2a3f5f"}, "type":"carpet"}], "ch
oropleth":[{"colorbar":
{"outlinewidth":0, "ticks":""}, "type":"choropleth"}], "contour":
[{"colorbar":{"outlinewidth":0, "ticks":""}, "colorscale":
[[0, "#0d0887"], [0.11111111111111111, "#46039f"],
[0.22222222222222222, "#7201a8"], [0.33333333333333333, "#9c179e"],
[0.44444444444444444, "#bd3786"], [0.55555555555555556, "#d8576b"],
[0.66666666666666666, "#ed7953"], [0.77777777777777778, "#fb9f3a"],
[0.88888888888888888, "#fdca26"],
[1, "#f0f921"]], "type":"contour"}], "contourcarpet":[{"colorbar":
{"outlinewidth":0, "ticks":""}, "type":"contourcarpet"}], "heatmap":
[{"colorbar":{"outlinewidth":0, "ticks":""}, "colorscale":
[[0, "#0d0887"], [0.11111111111111111, "#46039f"],

```



```

[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]], "type": "heatmap"}], "heatmapgl": [{"colorbar":
{"linewidth":0,"ticks":"","colorscale": [[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]], "type": "heatmapgl"}], "histogram": [{"marker": {"pattern":
{"fillmode": "overlay", "size": 10, "solidity": 0.2}}, "type": "histogram"}],
"histogram2d": [{"colorbar": {"linewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]], "type": "histogram2d"}], "histogram2dcontour":
[{"colorbar": {"linewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]], "type": "histogram2dcontour"}], "mesh3d": [{"colorbar":
{"linewidth":0,"ticks":"","type": "mesh3d"}], "parcoords": [{"line":
{"colorbar": {"linewidth":0,"ticks":"","type": "parcoords"}}, "pie":
[{"automargin": true, "type": "pie"}], "scatter": [{"fillpattern":
{"fillmode": "overlay", "size": 10, "solidity": 0.2}, "type": "scatter"}], "sc
atter3d": [{"line": {"colorbar": {"linewidth":0,"ticks":"","marker":
{"colorbar":
{"linewidth":0,"ticks":"","type": "scatter3d"}}, "scattercarpet":
[{"marker": {"colorbar":
{"linewidth":0,"ticks":"","type": "scattercarpet"}}, "scattergeo":
[{"marker": {"colorbar":
{"linewidth":0,"ticks":"","type": "scattergeo"}}, "scattergl":
[{"marker": {"colorbar":
{"linewidth":0,"ticks":"","type": "scattergl"}}, "scattermapbox":
[{"marker": {"colorbar":
{"linewidth":0,"ticks":"","type": "scattermapbox"}}, "scatterpolar"
: [{"marker": {"colorbar":
{"linewidth":0,"ticks":"","type": "scatterpolar"}}, "scatterpolargl
": [{"marker": {"colorbar":
{"linewidth":0,"ticks":"","type": "scatterpolargl"}}, "scatterterna
ry": [{"marker": {"colorbar":
{"linewidth":0,"ticks":"","type": "scatterternary"}}, "surface":
[{"colorbar": {"linewidth":0,"ticks":"","colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],

```

```

[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]], "type": "surface"}], "table": [{"cells": {"fill":
{"color": "#EBF0F8"}, "line": {"color": "white"}}, "header": {"fill":
{"color": "#C8D4E3"}, "line":
{"color": "white"}}, "type": "table"}]], "layout": {"annotationdefaults":
{"arrowcolor": "#2a3f5f", "arrowhead": 0, "arrowwidth": 1}, "autotypenumbers
": "strict", "coloraxis": {"colorbar":
{"linewidth": 0, "ticks": ""}}, "colorscale": {"diverging":
[[0, "#8e0152"], [0.1, "#c51b7d"], [0.2, "#de77ae"], [0.3, "#f1b6da"],
[0.4, "#fde0ef"], [0.5, "#f7f7f7"], [0.6, "#e6f5d0"], [0.7, "#b8e186"],
[0.8, "#7fb3c1"], [0.9, "#4d9221"], [1, "#276419"]]], "sequential":
[[0, "#0d0887"], [0.1111111111111111, "#46039f"],
[0.2222222222222222, "#7201a8"], [0.3333333333333333, "#9c179e"],
[0.4444444444444444, "#bd3786"], [0.5555555555555556, "#d8576b"],
[0.6666666666666666, "#ed7953"], [0.7777777777777778, "#fb9f3a"],
[0.8888888888888888, "#fdca26"], [1, "#f0f921"]]], "sequentialminus":
[[0, "#0d0887"], [0.1111111111111111, "#46039f"],
[0.2222222222222222, "#7201a8"], [0.3333333333333333, "#9c179e"],
[0.4444444444444444, "#bd3786"], [0.5555555555555556, "#d8576b"],
[0.6666666666666666, "#ed7953"], [0.7777777777777778, "#fb9f3a"],
[0.8888888888888888, "#fdca26"], [1, "#f0f921"]]]}, "colorway":
["#636efa", "#EF553B", "#00cc96", "#ab63fa", "#FFA15A", "#19d3f3", "#FF6692",
"#B6E880", "#FF97FF", "#FECB52"], "font": {"color": "#2a3f5f"}, "geo":
{"bgcolor": "white", "lakecolor": "white", "landcolor": "#E5ECF6", "showlake
s": true, "showland": true, "subunitcolor": "white"}, "hoverlabel":
{"align": "left"}, "hovermode": "closest", "mapbox":
{"style": "light"}, "paper_bgcolor": "white", "plot_bgcolor": "#E5ECF6", "po
lar": {"angularaxis":
{"gridcolor": "white", "linecolor": "white", "ticks": ""}, "bgcolor": "#E5ECF
6", "radialaxis":
{"gridcolor": "white", "linecolor": "white", "ticks": ""}}, "scene":
{"xaxis":
{"backgroundcolor": "#E5ECF6", "gridcolor": "white", "gridwidth": 2, "lineco
lor": "white", "showbackground": true, "ticks": "", "zerolinecolor": "white"},
"yaxis":
{"backgroundcolor": "#E5ECF6", "gridcolor": "white", "gridwidth": 2, "lineco
lor": "white", "showbackground": true, "ticks": "", "zerolinecolor": "white"},
"zaxis":
{"backgroundcolor": "#E5ECF6", "gridcolor": "white", "gridwidth": 2, "lineco
lor": "white", "showbackground": true, "ticks": "", "zerolinecolor": "white"}
}, "shapedefaults": {"line": {"color": "#2a3f5f"}}, "ternary": {"aaxis":
{"gridcolor": "white", "linecolor": "white", "ticks": ""}, "baxis":
{"gridcolor": "white", "linecolor": "white", "ticks": ""}, "bgcolor": "#E5ECF
6", "caxis":
{"gridcolor": "white", "linecolor": "white", "ticks": ""}}, "title":
{"x": 5.0e-2}, "xaxis":
{"automargin": true, "gridcolor": "white", "linecolor": "white", "ticks": "",
"title":

```

```
{
  "standoff":15,
  "zerolinecolor":"white",
  "zerolinewidth":2,
  "yaxis":{
    "automargin":true,
    "gridcolor":"white",
    "linecolor":"white",
    "ticks":"","title":
    {
      "standoff":15,
      "zerolinecolor":"white",
      "zerolinewidth":2
    }
  },
  "violinmode":"group",
  "xaxis":{
    "anchor":"y",
    "domain":[0,1],
    "title":{
      "text":"dispatching_base_number"
    }
  },
  "yaxis":{
    "anchor":"x",
    "domain":[0,1],
    "title":{
      "text":"active_vehicles"
    }
  }
}
```

NOW,

WE HAVE TO COLLECT THE DATA & MAKE IT READY FOR THE DATA ANALYSIS

```
import os
```

```
os.listdir(r'C:\Users\Aakasha\Desktop\uber-pickups-in-new-york-city-20220626T112836Z-001\uber-pickups-in-new-york-city')
```

```
[
  '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-jul14.csv',
  'uber-raw-data-jun14.csv',
  'uber-raw-data-may14.csv',
  'uber-raw-data-sep14.csv'
]
```

NOW, WE HAVE CONSIDER ONLY THE FILES OF UBER\_RAW-DATA- [APR-SEP]. THAT IS THE LAST SEVEN EXCEPT THE JANJUNE-15.CSV FILE

```
files=os.listdir(r'C:\Users\Aakasha\Desktop\uber-pickups-in-new-york-city-20220626T112836Z-001\uber-pickups-in-new-york-city')[-7:]
```

```
files
```

```
[
  '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'
]
```



NOW, WE HAVE REMOVE THE UBER-.....JANJUNE-15.CSV FILE

```
files.remove('uber-raw-data-janjune-15.csv')
```

```
files
```

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

NOW, WE GOT THE REQUIRED FILES

```
path=r'C:\Users\Aakash\Desktop\uber-pickups-in-new-york-city-  
20220626T112836Z-001\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])
```

```
final.shape
```

```
(4534327, 4)
```

```
final.head(2)
```

	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

NOW, WE HAVE ANALYSE i.e., CLEAN THE DATA

1.DUPLICATE VALUES

2.MISSING VALUE

3.WRONG DATA TYPE

TO KNOW NO.OF DUPLICATES:-

```
final.duplicated().sum()
```

```
82581
```

TO DELETE/CLEAR THE DUPLICATES:-

IF, INPLACE= TRUE, THEN DATASET CHANGE OCCURS.

IF, INPLACE= FALSE, NO CHANGE OCCURS IN THE DATASET

```
final.drop_duplicates(inplace=True)
```

```
final.shape
```

(4451746, 4)

WE CAN OBSERVE THE DECREASE IN THE NO.OF ROWS

NOW,

PROB:-

AT WHAT LOCATIONS OF NEW YORK CITY WE ARE GETTING RUSH

```
final.head()
```

	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

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

```
rush_uber
```

	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 x 3 columns]

HOW TO SEE THE WORLD MAP!!

```
!pip install folium
```

```
Requirement already satisfied: folium in c:\python folder\anaconda
aakash\lib\site-packages (0.12.1.post1)
Requirement already satisfied: requests in c:\python folder\anaconda
aakash\lib\site-packages (from folium) (2.25.1)
Requirement already satisfied: jinja2>=2.9 in c:\python folder\
anaconda aakash\lib\site-packages (from folium) (2.11.3)
Requirement already satisfied: numpy in c:\python folder\anaconda
aakash\lib\site-packages (from folium) (1.20.1)
Requirement already satisfied: branca>=0.3.0 in c:\python folder\
anaconda aakash\lib\site-packages (from folium) (0.5.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\python folder\
```

```

anaconda aakash\lib\site-packages (from jinja2>=2.9->folium) (1.1.1)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\python folder\
anaconda aakash\lib\site-packages (from requests->folium) (4.0.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\python
folder\anaconda aakash\lib\site-packages (from requests->folium)
(1.26.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\python folder\
anaconda aakash\lib\site-packages (from requests->folium) (2020.12.5)
Requirement already satisfied: idna<3,>=2.5 in c:\python folder\
anaconda aakash\lib\site-packages (from requests->folium) (2.10)

```

```
import folium
```

```
basemap=folium.Map()
```

```
basemap
```

```
<folium.folium.Map at 0x1d81cbf9a90>
```

```
from folium.plugins import HeatMap
```

```
HeatMap(rush_uber).add_to(basemap)
```

```
<folium.plugins.heat_map.HeatMap at 0x1d81d1809d0>
```

```
basemap
```

```
<folium.folium.Map at 0x1d81cbf9a90>
```

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

```
final
```

	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
4	2014-09-01 00:11:00	40.8145	-73.9444	B02512
...	...	...	...	...
564511	2014-04-30 23:22:00	40.7640	-73.9744	B02764
564512	2014-04-30 23:26:00	40.7629	-73.9672	B02764
564513	2014-04-30 23:31:00	40.7443	-73.9889	B02764
564514	2014-04-30 23:32:00	40.6756	-73.9405	B02764
564515	2014-04-30 23:48:00	40.6880	-73.9608	B02764

```
[4451746 rows x 4 columns]
```

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

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

final

	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
4	2014-09-01 00:11:00	40.8145	-73.9444	B02512
...	...	...	...	...
564511	2014-04-30 23:22:00	40.7640	-73.9744	B02764
564512	2014-04-30 23:26:00	40.7629	-73.9672	B02764
564513	2014-04-30 23:31:00	40.7443	-73.9889	B02764
564514	2014-04-30 23:32:00	40.6756	-73.9405	B02764
564515	2014-04-30 23:48:00	40.6880	-73.9608	B02764

[4451746 rows x 4 columns]

PROB:-

EXAMINE THE RUSH ON HOUR AND WEEKDAY(PAIR WISE ANALYSIS)

WE CAN DO IT BY MATRIX METHOD, D D D H | | H | | H | |

WE NEED TO EXTRACT THE MONTH, WEEKDAY

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

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

```
final.head()
```

	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
4	2014-09-01 00:11:00	40.8145	-73.9444	B02512	1	0

```
final.groupby(['weekday','hour']).size()
```

weekday	hour	
1	0	3178
	1	1944
	2	1256
	3	1308
	4	1429
	...	...
31	19	4898
	20	4819
	21	5064

```
22      5164
23      3961
Length: 744, dtype: int64
```

```
type(final.groupby(['weekday', 'hour']).size())
```

```
pandas.core.series.Series
```

UNSTACK() WILL CHANGE THE DATATYPE FROM SERIES TO DATAFRAMES

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

```
pivot
```

hour	0	1	2	3	4	5	6	7	8
9 ... \									
weekday									
..									
1	3178	1944	1256	1308	1429	2126	3664	5380	5292
4617 ...									
2	2435	1569	1087	1414	1876	2812	4920	6544	6310
4712 ...									
3	3354	2142	1407	1467	1550	2387	4241	5663	5386
4657 ...									
4	2897	1688	1199	1424	1696	2581	4592	6029	5704
4744 ...									
5	2733	1541	1030	1253	1617	2900	4814	6261	6469
5530 ...									
6	4537	2864	1864	1555	1551	2162	3642	4766	4942
4401 ...									
7	3645	2296	1507	1597	1763	2422	4102	5575	5376
4639 ...									
8	2830	1646	1123	1483	1889	3224	5431	7361	7357
5703 ...									
9	2657	1724	1222	1480	1871	3168	5802	7592	7519
5895 ...									
10	3296	2126	1464	1434	1591	2594	4664	6046	6158
5072 ...									
11	3036	1665	1095	1424	1842	2520	4954	6876	6871
5396 ...									
12	3227	2147	1393	1362	1757	2710	4576	6250	6231
5177 ...									
13	5408	3509	2262	1832	1705	2327	4196	5685	6060
5631 ...									
14	3748	2349	1605	1656	1756	2629	4257	5781	5520
4824 ...									
15	2497	1515	1087	1381	1862	2980	5050	6837	6729
5201 ...									
16	2547	1585	1119	1395	1818	2966	5558	7517	7495
5958 ...									
17	3155	2048	1500	1488	1897	2741	4562	6315	5882



9	7877	9220	10270	11910	11449	9804	8909	8665	7499
5203									
10	7612	9578	11045	11875	10934	9613	9687	9240	7766
5496									
11	7503	8920	10125	10898	10361	9327	8824	8730	7771
5360									
12	7743	9390	10734	11713	12216	10393	9965	10310	9992
7945									
13	8200	9264	10534	11826	11450	9921	8705	8423	7363
5936									
14	6963	8192	9511	10115	9553	9146	9182	8589	6891
4460									
15	7633	8505	10285	11959	11728	11032	10509	9105	7153
4480									
16	7597	9290	10804	11773	10855	10924	10142	10374	8094
5380									
17	7472	8997	10323	11236	11089	9919	9935	9823	8362
5699									
18	7534	9040	10274	10692	10338	9551	9310	9285	8015
5492									
19	7374	8898	9893	10741	10429	9701	10051	10049	9090
6666									
20	7462	8630	9448	10046	9272	8592	8614	8703	7787
5907									
21	7064	8127	9483	9817	9291	8317	8107	8245	7362
5231									
22	7337	9148	10574	10962	9884	8980	8772	8430	6784
4530									
23	7575	9309	9980	10341	10823	11347	11447	10347	8637
5577									
24	7083	8706	10366	10786	9772	9080	9213	8831	7480
4456									
25	7298	8732	9922	10504	10673	9048	8751	9508	8522
6605									
26	7269	8815	9885	10697	10867	10122	9820	10441	9486
7593									
27	7519	8803	9793	9838	9228	8267	7908	8507	7720
6046									
28	7341	8584	9671	9975	9132	8255	8309	7949	6411
4461									
29	7630	9249	10105	11113	10411	9301	9270	9114	6992
4323									
30	8396	10243	11554	12126	12561	11024	10836	10042	8275
4723									
31	4104	5099	5386	5308	5350	4898	4819	5064	5164
3961									

[31 rows x 24 columns]

NOW, WE ACTUALLY NEED THE MAX OF THE UBER RUSH ON (HOUR,WEEKDAY)

```
pivot.style.background_gradient()
```

```
<pandas.io.formats.style.Styler at 0x1d82bf470d0>
```