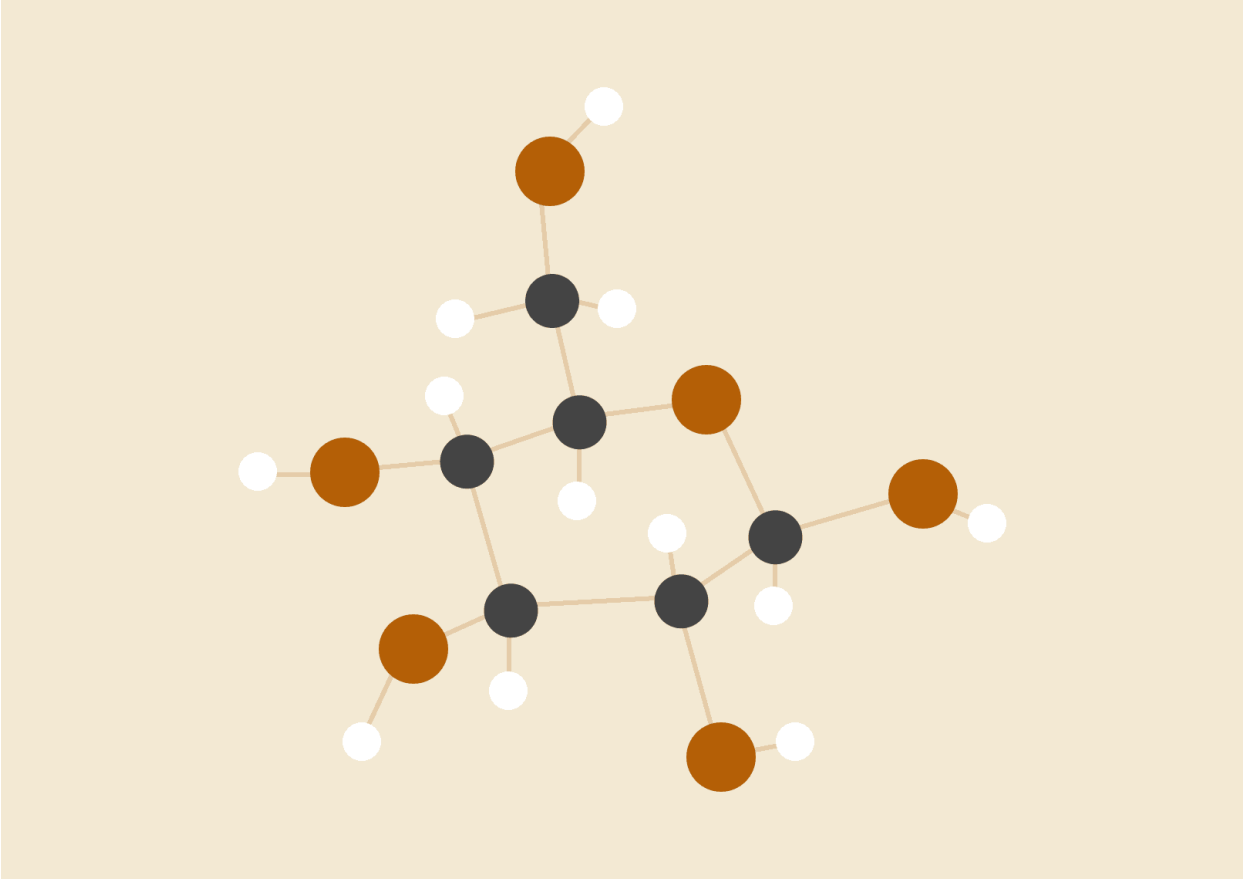
PDS - REPORT

Uber Trip density predictor



K.Navdeep | K.Sai Bhargava | K.Varun | Yash gupta

S20190010099 | S20190010083 | S20190020222 | S20190010197

29-04-2022

ABSTRACT

In recent years transportation has taken a huge change of form, the business of transportation as a service has taken a huge turn. With an increase in demand for quick and easy transportation, companies like Uber have achieved it by various strategies. To meet this heavy need with only limited resources, companies have to use their resources (vehicles) strategically. Over many years, thera are logs of data how cabs are being hired at different locations and time. We have taken the case of New York city. Our main idea is to firstly forecast how the demand for uber fluctuates access to new york with respect to time. With this we can manage our resources accordingly to meet the needs. This helps in improving customer experience because with enough resources, there is no waiting time for the customer. Hence we are forecasting the demand of uber with respect to time in different sub - companies of uber

PROCEDURE

- Data set description
- EDA over the dataset
- Visualize the traffic at uber
- Looking into models from the dataset
- Predict the trend of uber with time

DATA SET DESCRIPTION

The data set given consists of 6 csv files which contains the data of uber for months of April to september. The data set consists of

1. date/time - the date and time of uber pickup
2. Lat - latitude of the uber pickup
3. Lon - longitude of the uber pickup
4. Base - The sub company of uber.

As we are looking at the data collectively we can concatenate the dataset

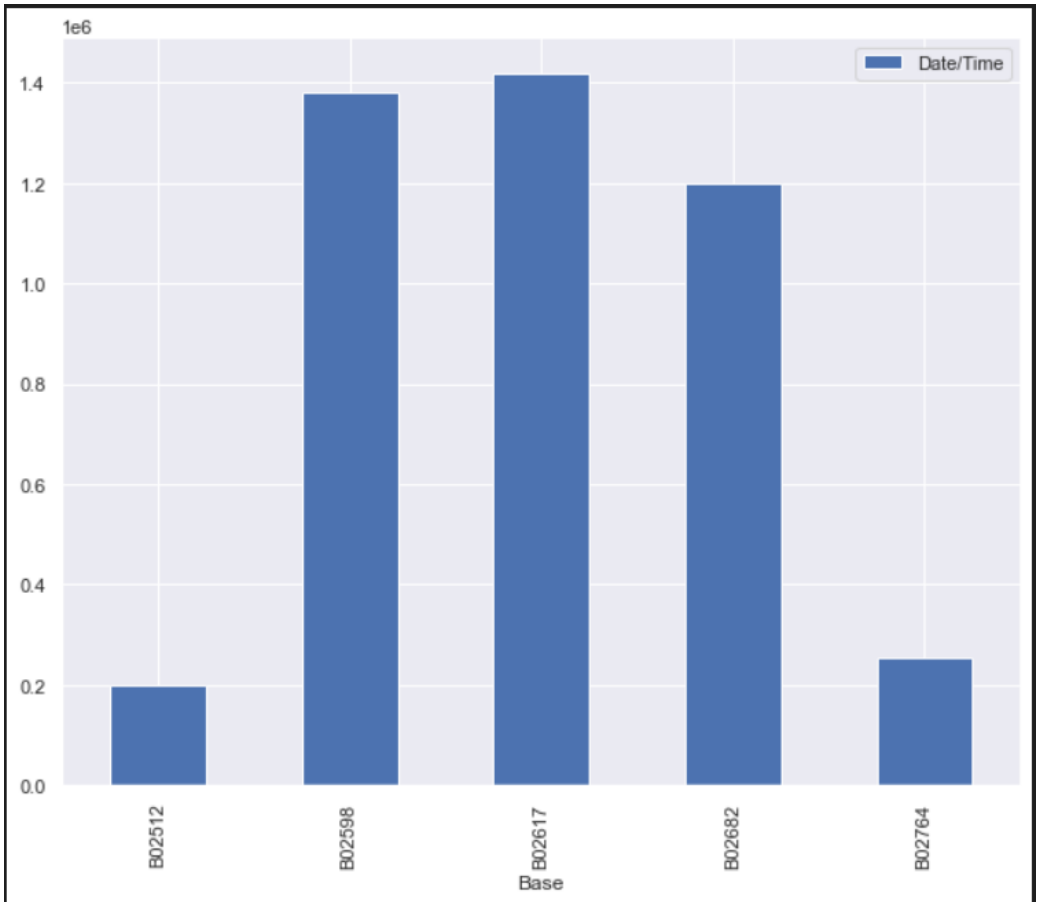
	Date/Time	Lat	Lon	Base
0	4/1/2014 0:11:00	40.7690	-73.9549	B02512
1	4/1/2014 0:17:00	40.7267	-74.0345	B02512
2	4/1/2014 0:21:00	40.7316	-73.9873	B02512
3	4/1/2014 0:28:00	40.7588	-73.9776	B02512
4	4/1/2014 0:33:00	40.7594	-73.9722	B02512
...	...	...	...	...
1028131	9/30/2014 22:57:00	40.7668	-73.9845	B02764
1028132	9/30/2014 22:57:00	40.6911	-74.1773	B02764
1028133	9/30/2014 22:58:00	40.8519	-73.9319	B02764
1028134	9/30/2014 22:58:00	40.7081	-74.0066	B02764
1028135	9/30/2014 22:58:00	40.7140	-73.9496	B02764

For individual month we can extract it based on the date/time column

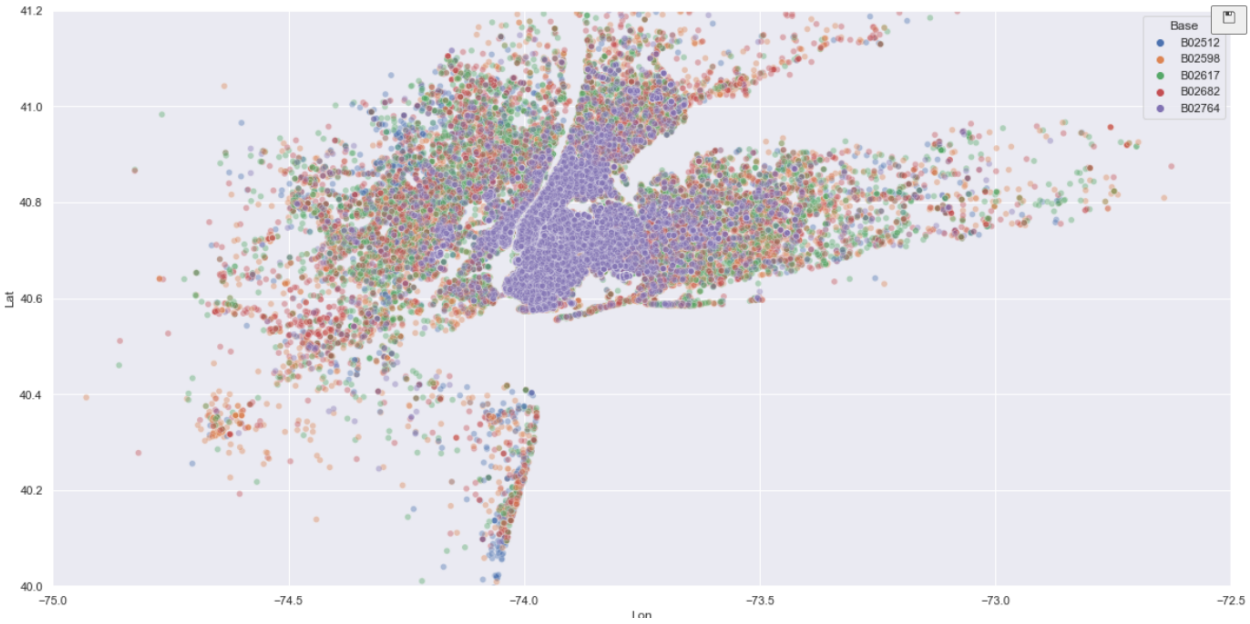
DATA PRE-PROCESSING

1. The first data pre processing step is concatenating the 6 months datasets into one dataframe as discussed previously
2. Then we convert the data/time feature from object type to datatime64
3. We check for null values in the dataset. In our dataset we have no null values, so we can proceed forward.
4. We check for duplicate entries in our dataset, because duplicate entries can make the data biased in particular cases. We observe that we have 82,581 duplicate entries which is a significant number with respect to the entire dataset size which is 4,534,327 entries. So we drop the duplicate entries.

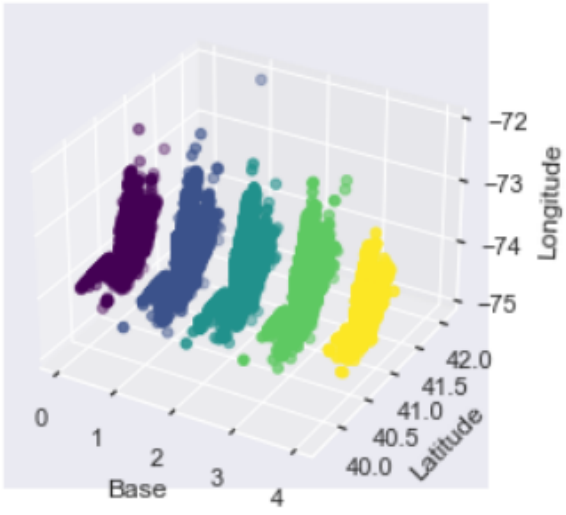
EXPLORATORY DATA ANALYSIS



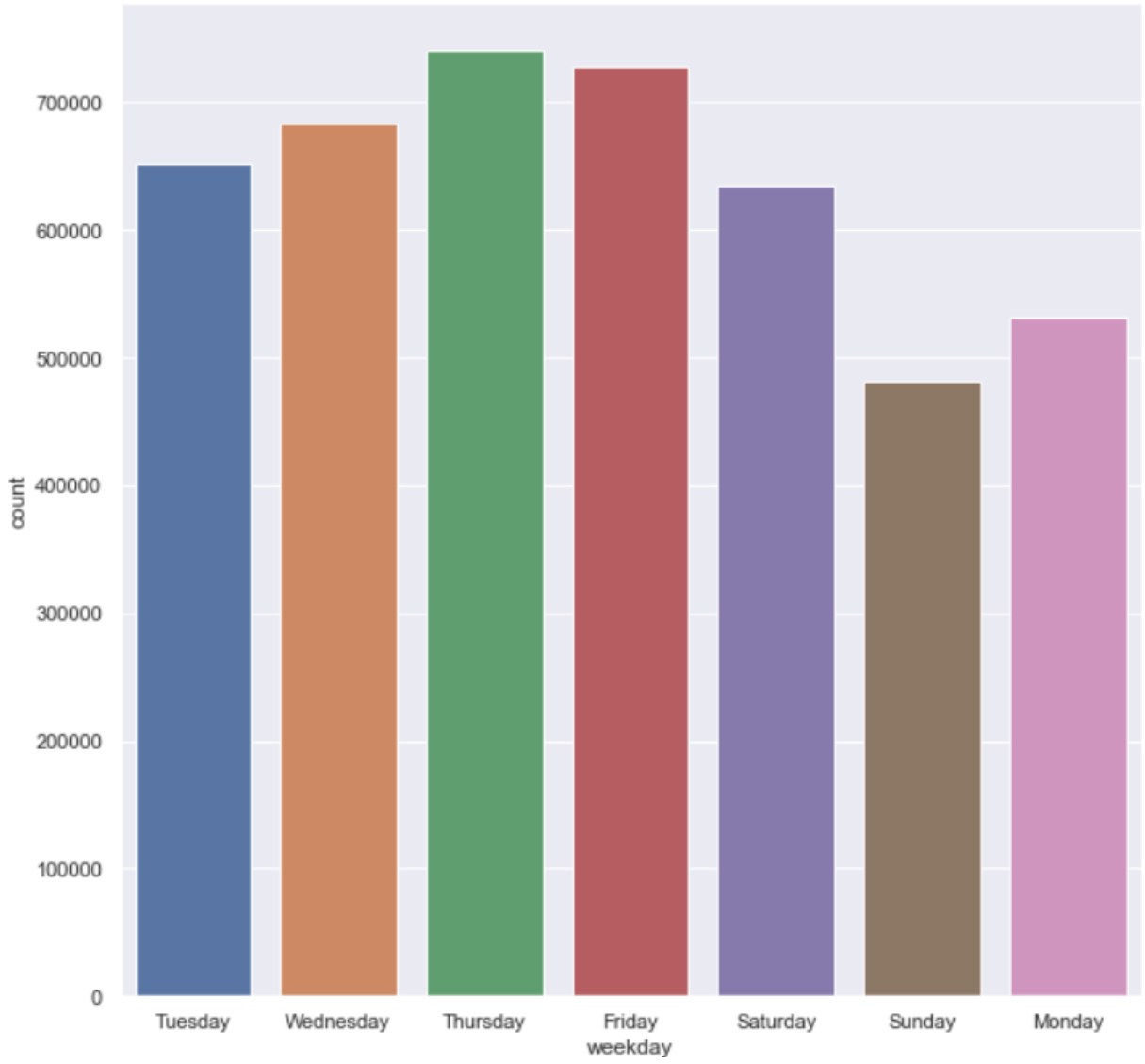
The above plot is the bar plot of number of ubers under each base (ie. sub company) for all the months together in millions. So we can observe that only 3 of the 5 bases contribute to the major part of the pickups and play a major role compared to the other two.



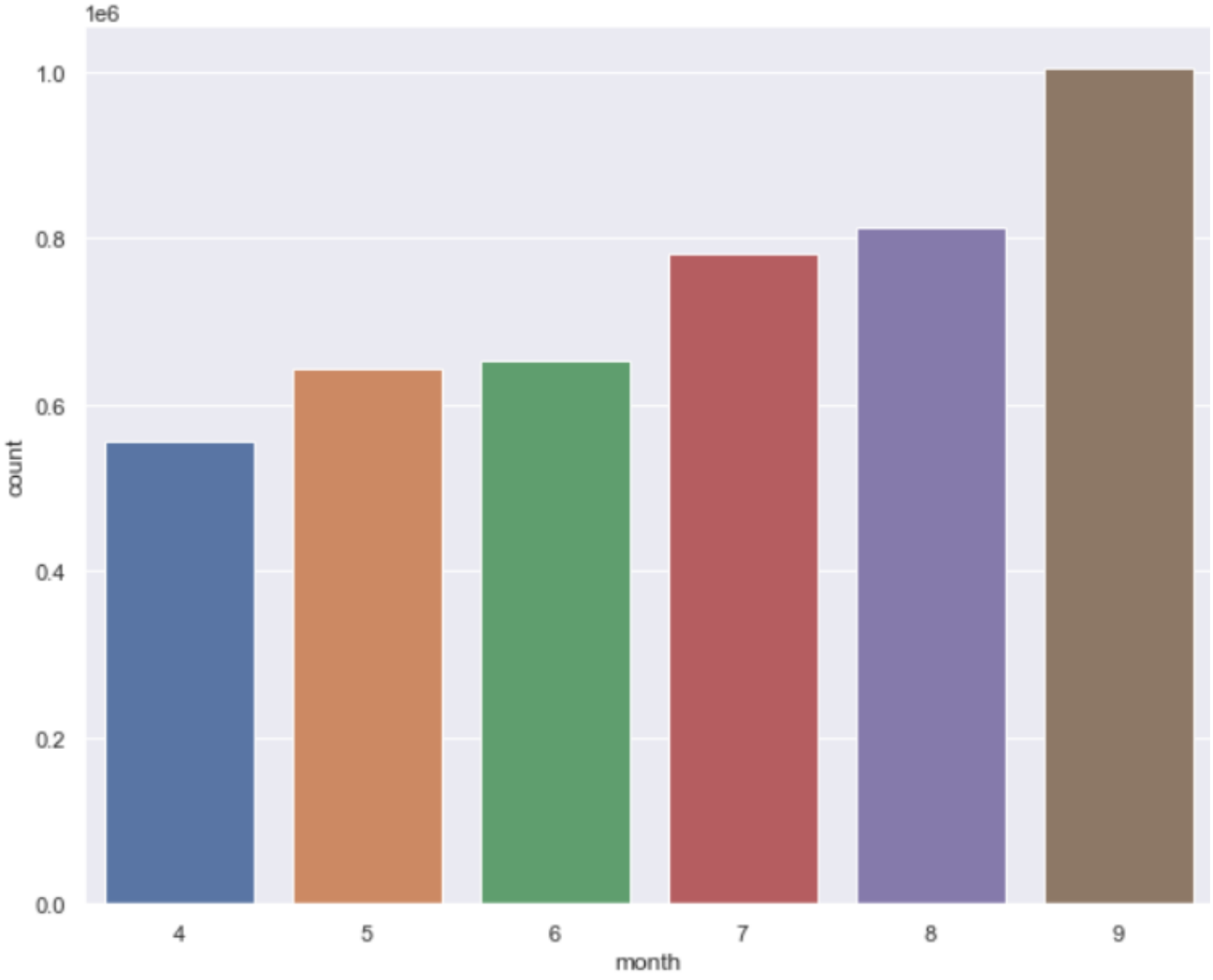
The above plot gives us the visual representation of how the pickups are scattered across New york. We can observe that we cannot distinguish the base of pickup from a particular position so it does not provide any valuable information regarding the position of pickup. Since the pickups are clustered closely and overlapping with reference to the base company.



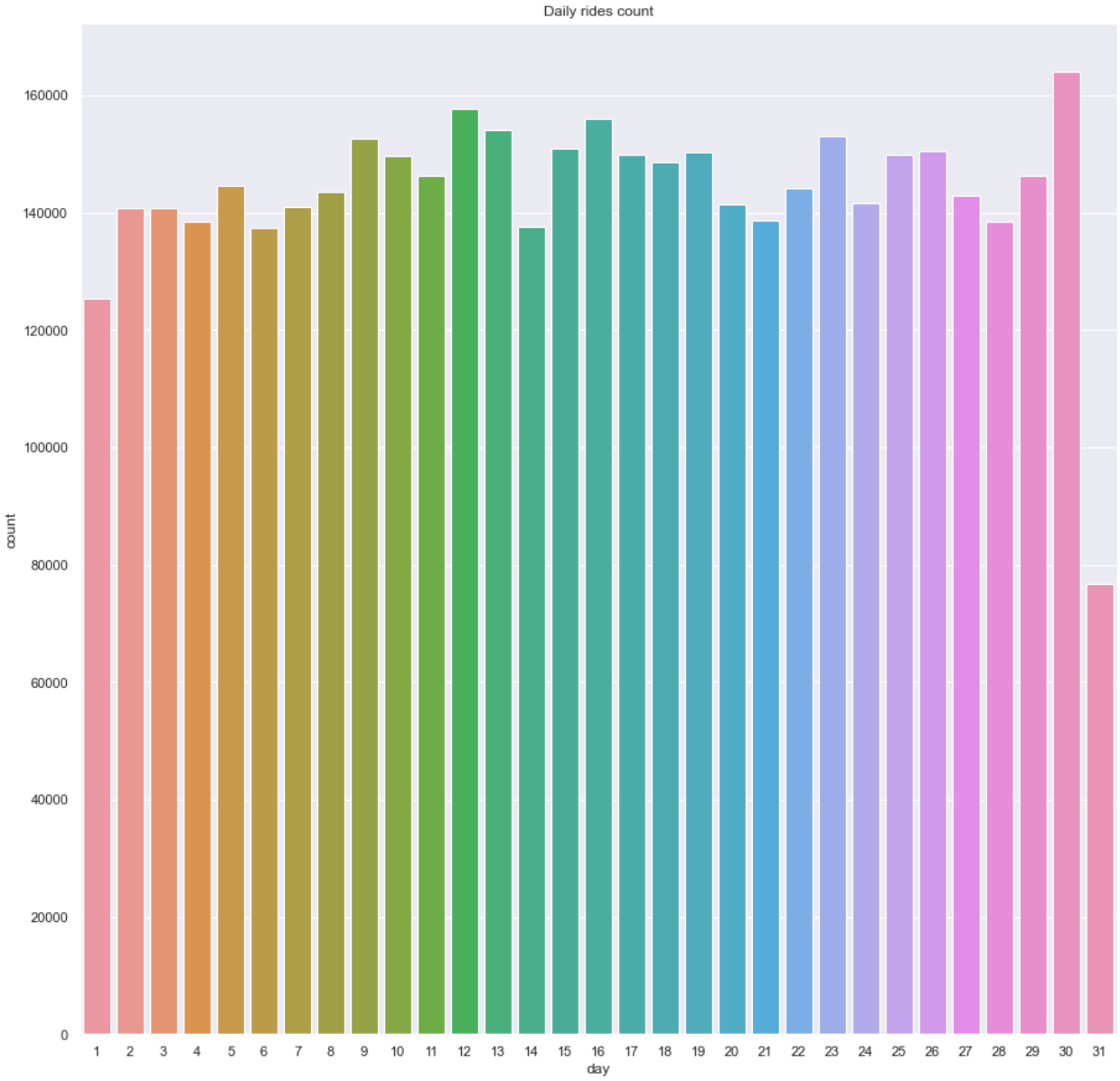
This 3d plot supports our conclusions previously mentioned. Notice the distributions of pickups of each base with respect to latitude and longitude, they are around the same center with some randomness at the tails or outside regions.



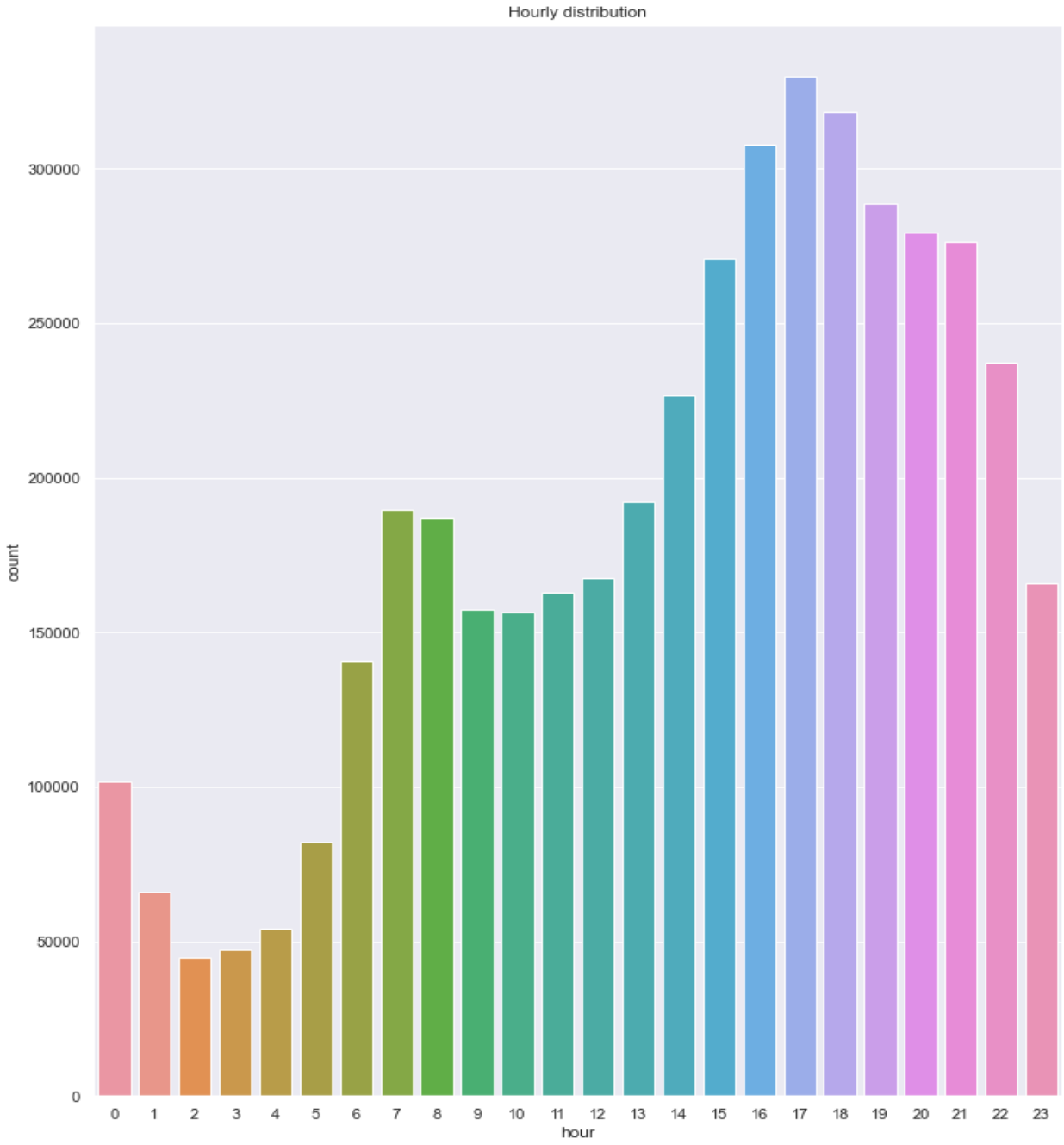
The above plot shows the count of uber pickups on each day of the week. We can observe the sudden drop in values during weekends which might be due to the reason that most of the organizations are off on sunday, and the trend in weekdays is quite constant from monday to friday with slight increase from monday to thursday and thursday as the most busy day of the week and friday slightly falling back of thursday.



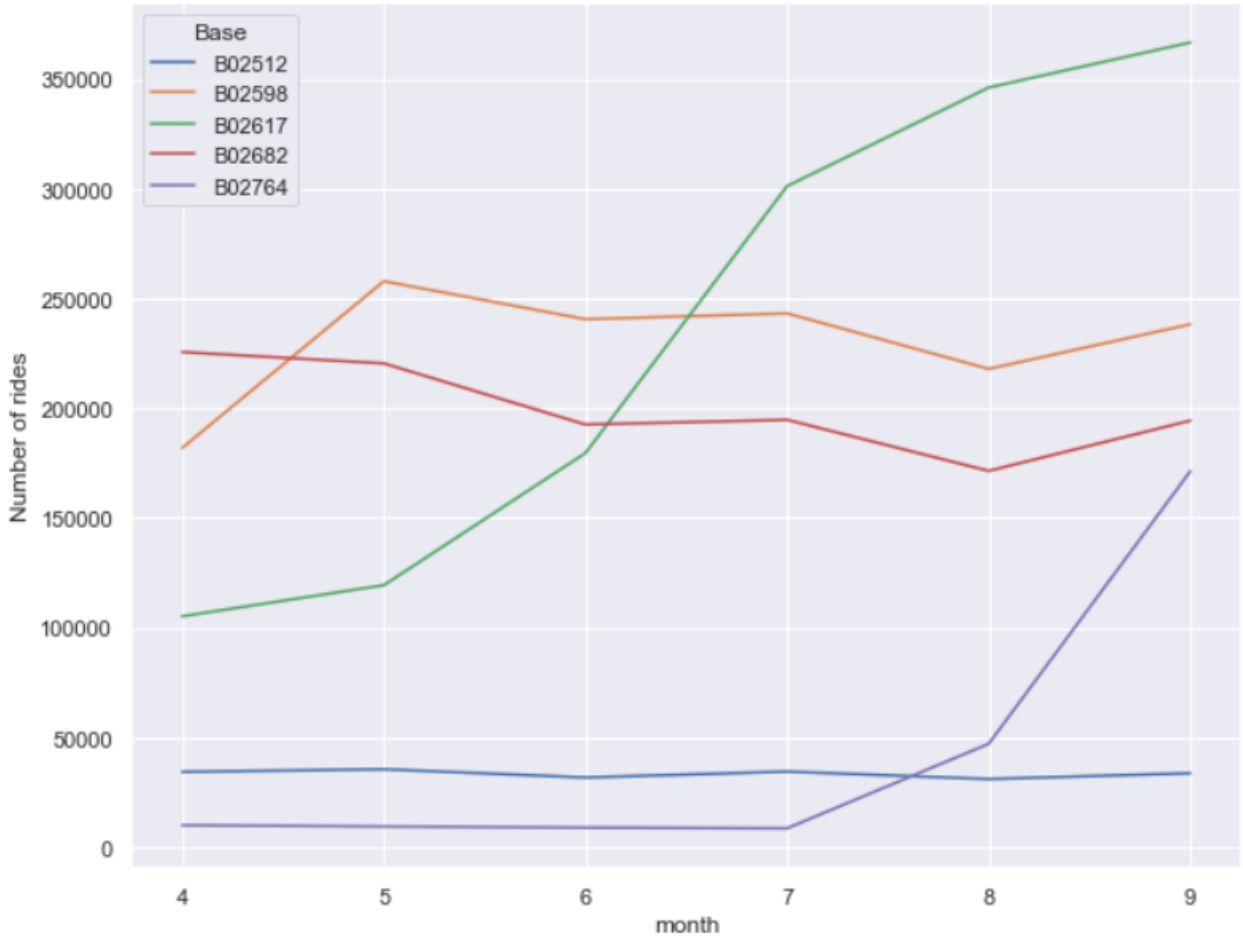
This plot shows how the pickups increase or vary based on the month. We observe a significant growth month by month in the number of pickups from April to september. Over 5 months the number of pickups doubled from 0.4 million to 0.8 million and there is a huge 0.2 million increase the next month. One take-away from this graph is that we observe a 0.2 million increase in pickups every 2 months which are from months 4-5, 6-7, 8-9.



This graph shows the daily demand in uber counts, notice the sudden fall in pickups count on 31st, this is since only few or half of the months have the day 31 in a month. If we look into it closely with 1 as the first day, we can see that the pickups show the weekly trend we discussed before which is constant increase from monday to thursday and decline from friday to sunday.

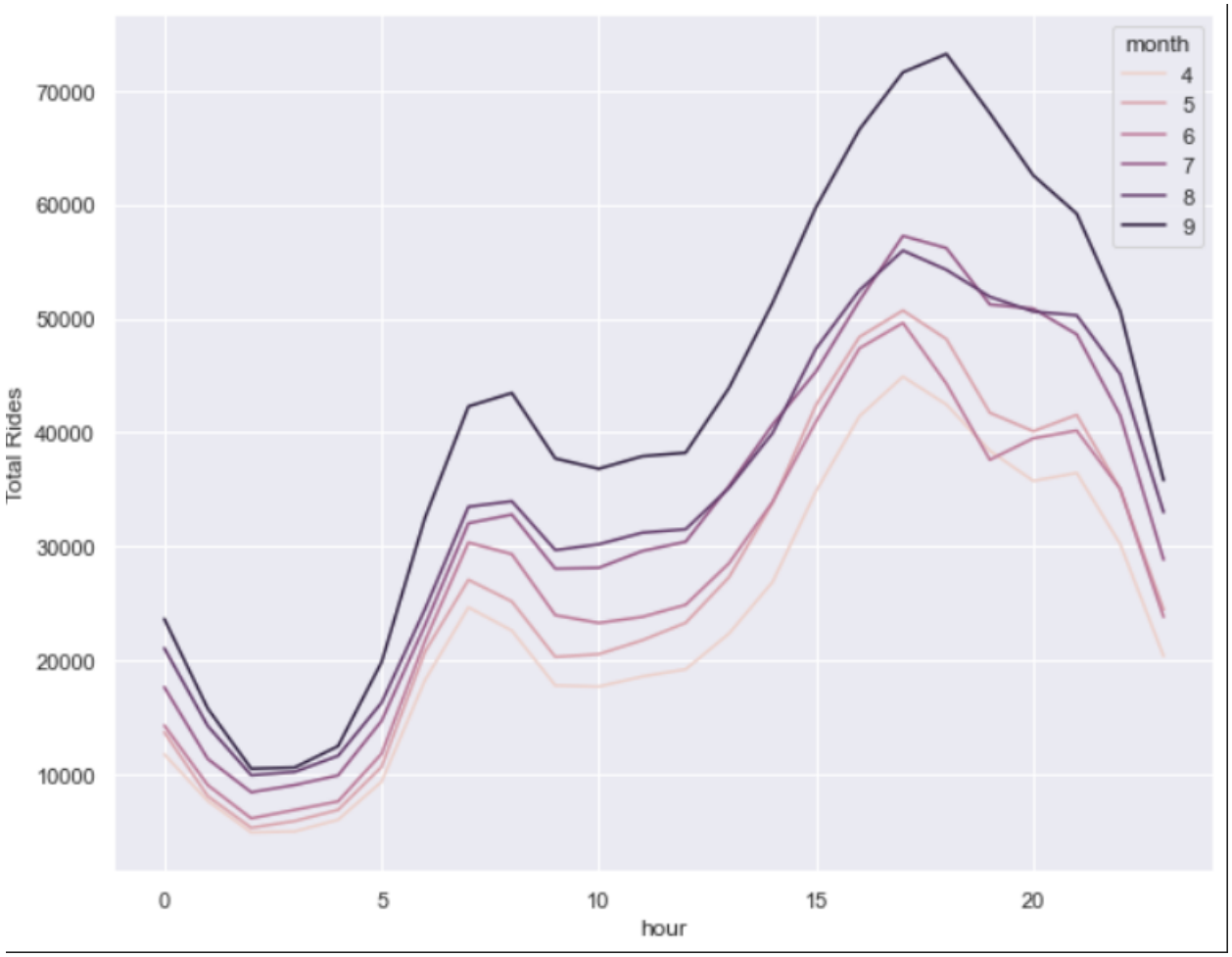


This is the hourly graph in a day. We can see that the busiest hours of the day are the evenings with the peak showing at 5 PM. Another peak is at 7-9 AM. From this we can say that these spikes are caused by the working class or employees who use uber for transport, this is the reason why we see a spike at the starting and ending work hours 7-9 AM and 4-6 PM.

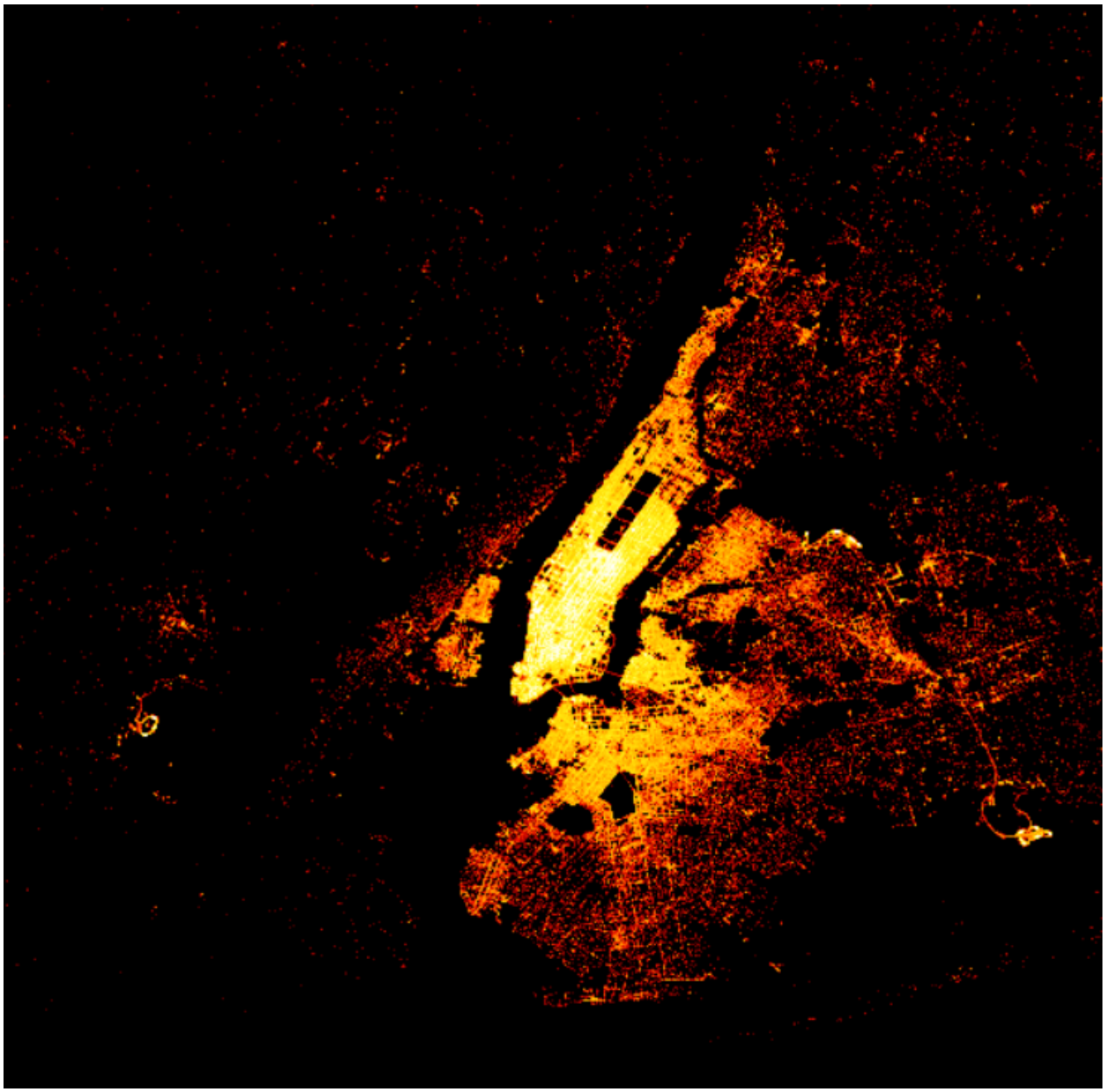


This graph shows the pickup counts from each base over the months. We can see that the green ie. B02617 base shows a huge increase in count from month to month and is drastically multiplying. Similarly the violet ie. B02764 is low and constant from April to June and has a spike in the month of July and keeps multiplying till September . Rest of the bases have a constant trend and show no significant spike but the bse B02512 has a constant low trend with a very less average count. The rest two have a constant trend with the mean around 225000.

The above graph shows the trend of pickups over multiple months on different days. They follow the same trend of weeks we discussed previously and one thing we can take-away is that the 16th of april june and September are zero. Which could be due to multiple reasons like, it could be a public holiday or the bases or uber has taken those days off for various reasons.



The above is the continuous graph showing the trend in pick counts with the hour in each month, and it shows the same trend as discussed. It has two peaks with constant increase in count. One peak at 7-9 AM and other at 4-6 PM. This graph gives the collective data of the monthly and hourly graphs together, showing how the mean of each month is increasing and the hourly trend.

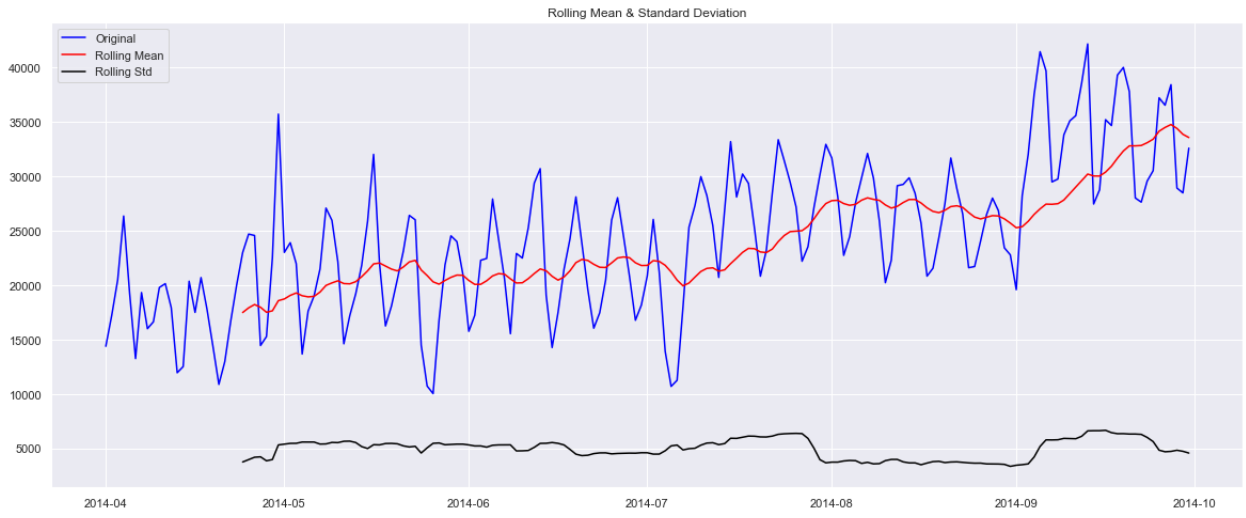


This is the heat map of how the pickups are scattered throughout New york. We can see that most of the pickups are heavily located near manhattan where most of the companies and employees are located. The places like Staten Island (bottom left) and Bronx (top right) are very less dense although they are around manhattan. Whereas there is a significant amount of pickups near brooklyn and queens since they include most of the living space in new york and are heavily populated.

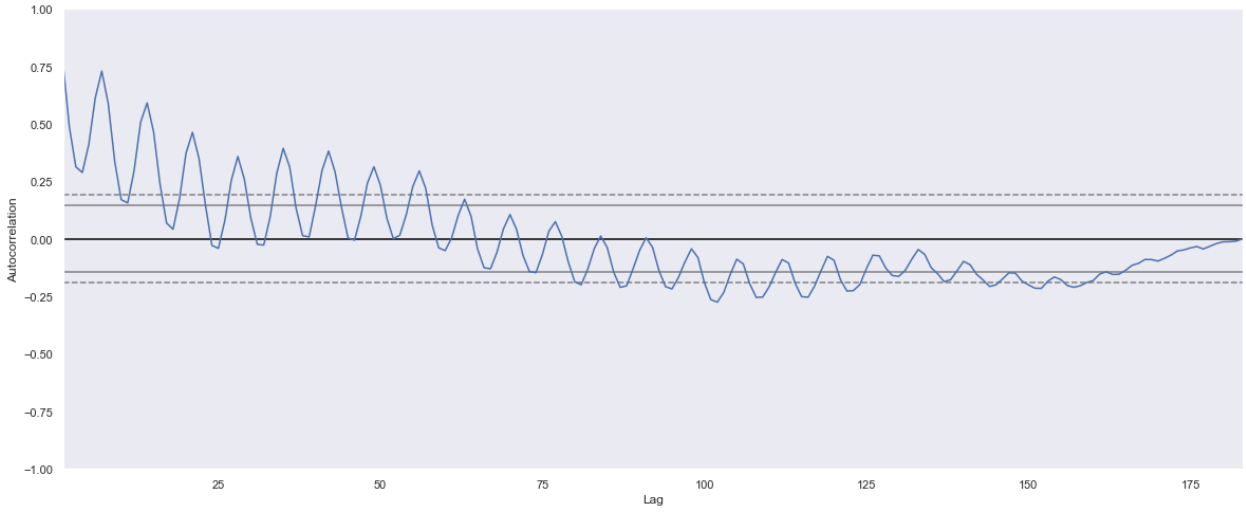
BUILDING THE MODEL AND OPTIMIZING

From the insights that we have taken from exploring the data, we have concluded that with this data, we can predict the trend of number of pickups or frequency of pickups across New York at a given time.

Since the data shows trends in pickups with respect to time, we can use time series to forecast the frequency of pickups. We have taken the train and test data ration as 4:1 i.e. 80% training data and 20% test data. So by training our model on data from April to 3rd week of august, we predict from last week of august to end of september.



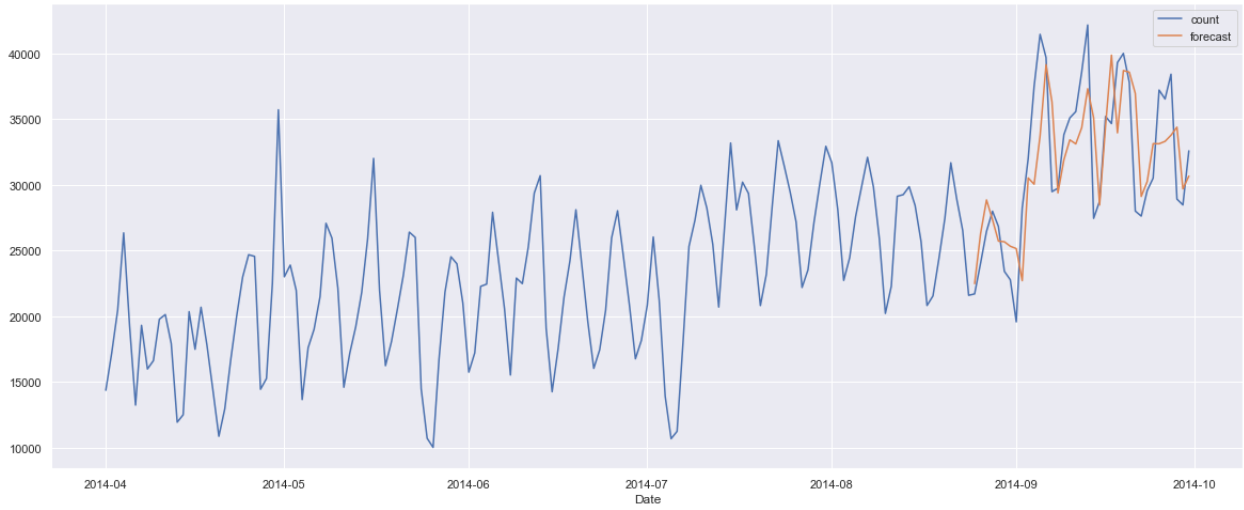
The blue line shows the original data and red with rolling mean and black with rolling standard deviation. We can see that the mean constantly increases from april to september and the standard deviation of data from april to september is quite constant.

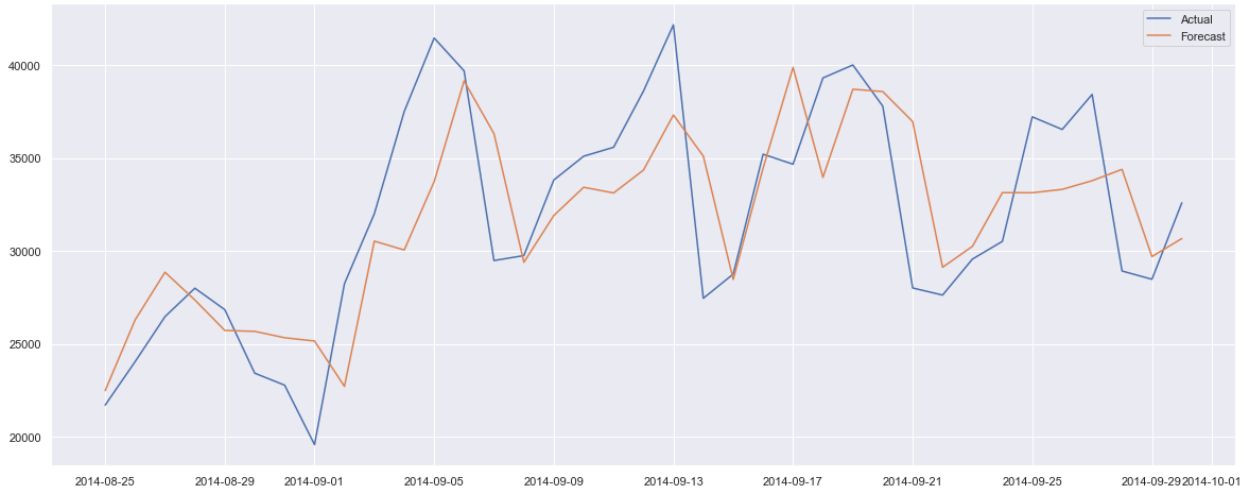


The above is the autocorrelation map of the time series data. We can see that the local maximas slowly dip into the significance range i.e. 0.25 and the local maximas continue to remain in the region, so we can say that the model is stationary and we can perform time series models over the dataset. We fit our model into a arima model with (p, d, q) (1,1,1) and the SARIMAX results are

```
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.6710      0.094      7.130      0.000       0.487       0.855
ma.L1         -0.9395      0.062    -15.103      0.000      -1.061      -0.818
ar.S.L24      -0.4591      0.071     -6.428      0.000      -0.599      -0.319
ma.S.L24      -0.9958      0.116     -8.571      0.000      -1.223      -0.768
sigma2      1.453e+07   8.04e-09   1.81e+15      0.000    1.45e+07    1.45e+07
==============================================================================
Ljung-Box (L1) (Q):                   2.72   Jarque-Bera (JB):                 1.41
Prob(Q):                              0.10   Prob(JB):                         0.50
Heteroskedasticity (H):               0.72   Skew:                            -0.22
Prob(H) (two-sided):                  0.24   Kurtosis:                         2.85
==============================================================================
```

This shows the coefficients and variance of the arima model.





The above graphs show the actual and our predicted value, the second graph gives us a closer look at the model. Visually we can conclude that our model is working very well in predicting the trends in the data and magnitude of spikes. Let us check if we can improve the performance of our model .

Using the pmd arima library we set the max and min limits of p, d, q values and find the best fitting model by simply checking all permutations. We use the adf-test to find the optimal D value for differencing the data. We set the max frequencies of p and q as 3 since it is most suggested and suits our model.

```
Performing stepwise search to minimize aic
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=inf, Time=0.54 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=3587.645, Time=0.01 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=3586.091, Time=0.02 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=3584.067, Time=0.03 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=3585.735, Time=0.01 sec
 ARIMA(0,1,2)(0,0,0)[0] intercept   : AIC=inf, Time=0.19 sec
 ARIMA(1,1,2)(0,0,0)[0] intercept   : AIC=inf, Time=0.49 sec
 ARIMA(0,1,1)(0,0,0)[0]             : AIC=3582.094, Time=0.02 sec
 ARIMA(1,1,1)(0,0,0)[0]             : AIC=3565.112, Time=0.05 sec
 ARIMA(1,1,0)(0,0,0)[0]             : AIC=3584.146, Time=0.02 sec
 ARIMA(2,1,1)(0,0,0)[0]             : AIC=3532.102, Time=0.11 sec
 ARIMA(2,1,0)(0,0,0)[0]             : AIC=3574.826, Time=0.03 sec
 ARIMA(3,1,1)(0,0,0)[0]             : AIC=3526.720, Time=0.10 sec
 ARIMA(3,1,0)(0,0,0)[0]             : AIC=3563.595, Time=0.04 sec
 ARIMA(3,1,2)(0,0,0)[0]             : AIC=3481.167, Time=0.32 sec
 ARIMA(2,1,2)(0,0,0)[0]             : AIC=3489.487, Time=0.32 sec
 ARIMA(3,1,3)(0,0,0)[0]             : AIC=3493.212, Time=0.45 sec
 ARIMA(2,1,3)(0,0,0)[0]             : AIC=3470.268, Time=0.53 sec
 ARIMA(1,1,3)(0,0,0)[0]             : AIC=3542.734, Time=0.11 sec
 ARIMA(1,1,2)(0,0,0)[0]             : AIC=3555.489, Time=0.09 sec
 ARIMA(2,1,3)(0,0,0)[0] intercept   : AIC=3474.285, Time=0.48 sec

Best model:  ARIMA(2,1,3)(0,0,0)[0]
Total fit time: 3.961 seconds
```

The above is the results of pmd arima, we get that ARIMA(2,1,3) is the best fit so let us see.

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	1.2492	0.030	41.728	0.000	1.190	1.308
ar.L2	-0.9815	0.025	-38.955	0.000	-1.031	-0.932
ma.L1	-1.5820	0.112	-14.104	0.000	-1.802	-1.362
ma.L2	1.3020	0.179	7.260	0.000	0.950	1.653
ma.L3	-0.3244	0.141	-2.308	0.021	-0.600	-0.049
ar.S.L24	0.0659	0.171	0.386	0.700	-0.269	0.401
ma.S.L24	-0.9910	0.174	-5.703	0.000	-1.332	-0.650
sigma2	1.905e+07	9.45e-09	2.02e+15	0.000	1.91e+07	1.91e+07

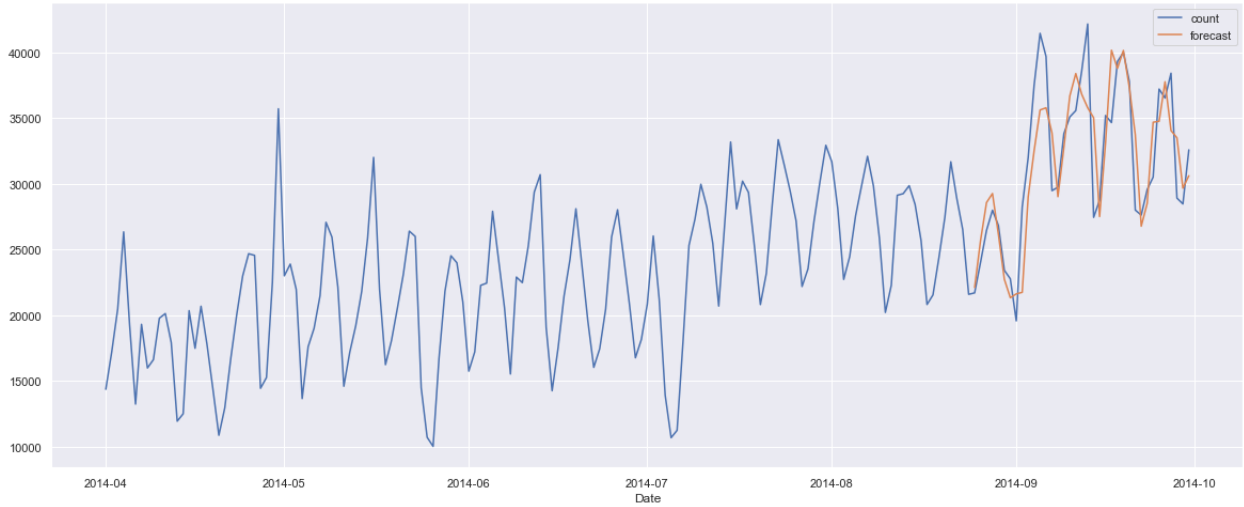
Here we can see the coefficients and std error values of the new ARIM model

We now compare these both with mathematical metrics mape, mpe,corr, minmax. Let us see what each metric tells us.

- The Mean Absolute Percentage Error (MAPE) is one of the most commonly used KPIs to measure forecast accuracy. MAPE is the sum of the individual absolute errors divided by the demand (each period separately). It is the average of the percentage errors.

ARIMA MODEL	BASE MODEL	OPTIMIZED
MAPE	0.1027	0.0839
MPE	0.0112	0.0028
CORR	0.7341	0.8281
MINMAX	0.0934	0.0783

RESULTS AND CONCLUSION



Above is the time series model along with actual and predicted data differentiated with color. From the graph and previously mentioned metrics with a mape of 0.084 we conclude that our model is performing good on the data.

We can conclude that the given data is a time series data and it follows a constant trend in ups and downs because the majority of pickups are from employees or by daily workers who repeatedly follow the same pattern for the entire year and year after year. So the ARIMA model is best suited to forecast the frequency in pickups at a given time.