Locale.ai_final

April 17, 2019

1 Locale.ai Interview Task

XRides in Bangalore has approached Locale to help use its location data to make better decisions. The challenge was to increse the utilisation of the cabs of XRides as the demand keeps on fluctuating during different time of day and and according to different areas. The idea was to identify the areas and at what time the demands are high so as to implement a geo surge strategy to increase the prices in those areas in order to meet the demand

The dataset given was the data of cabs with 43K(43431) instances and 19 attributes with different datatypes . The data was first cleaned , the NULL values for required attributes was changed by the mean of all the values rather than removing the instance so as to reduce the data loss .

```
In [1]: # importing the libraries :

        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import time
        %matplotlib inline
        import seaborn as sns
        from collections import Counter

        # ignoring the warnings :

        import warnings
        warnings.filterwarnings("ignore")
```

```
In [2]: # importing the dataset :

        df=pd.read_csv("data.csv")
        df.head()
        df.shape
```

```
Out[2]: (43431, 19)
```

```
In [3]: # changing the null values with mean of all the values :

        from sklearn.preprocessing import Imputer
        imputer=Imputer(missing_values="NaN",strategy="mean",axis=0)
        imputer=imputer.fit(df[["from_lat","from_long","to_lat","to_long"]])
        df[["from_lat","from_long","to_lat","to_long"]]=imputer.transform(df[["from_lat","from_
```

```
In [4]: df.isnull().sum()
```

```
Out[4]: id                         0
        user_id                    0
        vehicle_model_id           0
        package_id             35881
        travel_type_id             0
        from_area_id              88
        to_area_id              9138
        from_city_id           27086
        to_city_id             41843
        from_date                  0
        to_date                17890
        online_booking             0
        mobile_site_booking        0
        booking_created            0
        from_lat                   0
        from_long                  0
        to_lat                     0
        to_long                    0
        Car_Cancellation           0
        dtype: int64
```

```
In [5]: # changing the data type of attributes :

        df[['from_date','to_date','booking_created']] = \
                df[['from_date','to_date','booking_created']].apply(pd.to_datetime)
```

```
In [6]: # adding the extra coloumns :

        df['from_date_hour']=df['from_date'].apply(lambda x: x.hour)
        df['from_day']=df['from_date'].apply(lambda x: x.dayofweek)
        df['from_month']=df['from_date'].apply(lambda x: x.month)
        df.head()
```

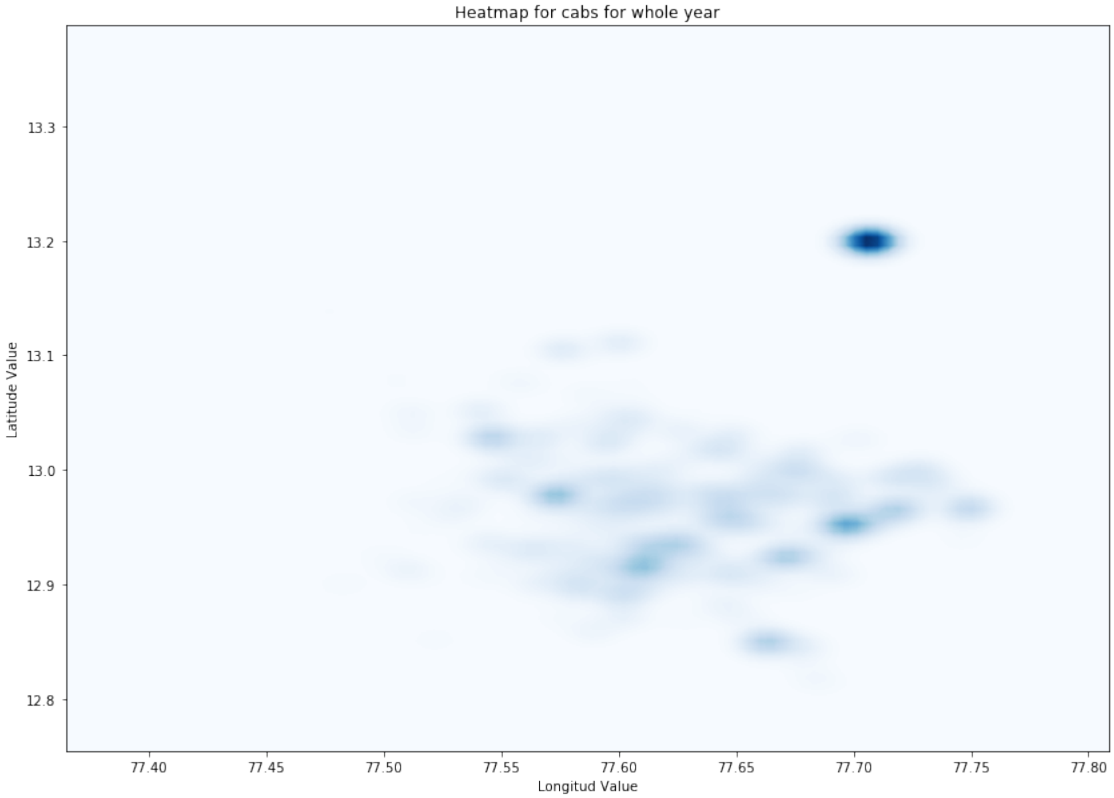
```
Out[6]:        id  user_id  vehicle_model_id  package_id  travel_type_id  \
        0  132512    22177                28         NaN               2
        1  132513    21413                12         NaN               2
        2  132514    22178                12         NaN               2
        3  132515    13034                12         NaN               2
        4  132517    22180                12         NaN               2

           from_area_id  to_area_id  from_city_id  to_city_id           from_date  \
        0          83.0       448.0           NaN         NaN 2013-01-01 02:00:00
        1        1010.0       540.0           NaN         NaN 2013-01-01 09:00:00
        2        1301.0      1034.0           NaN         NaN 2013-01-01 03:30:00
```

```
3          768.0       398.0           NaN         NaN 2013-01-01 05:45:00
4         1365.0       849.0           NaN         NaN 2013-01-01 09:00:00

      ...     mobile_site_booking     booking_created   from_lat  from_long  \
0     ...                       0 2013-01-01 01:39:00  12.924150  77.672290
1     ...                       0 2013-01-01 02:25:00  12.966910  77.749350
2     ...                       0 2013-01-01 03:08:00  12.937222  77.626915
3     ...                       0 2013-01-01 04:39:00  12.989990  77.553320
4     ...                       0 2013-01-01 07:53:00  12.845653  77.677925

      to_lat    to_long  Car_Cancellation  from_date_hour  from_day  \
0  12.927320  77.635750                 0               2         1
1  12.927680  77.626640                 0               9         1
2  13.047926  77.597766                 0               3         1
3  12.971430  77.639140                 0               5         1
4  12.954340  77.600720                 0               9         1

   from_month
0           1
1           1
2           1
3           1
4           1

[5 rows x 22 columns]
```

1.0.1 Heatmap for cabs for whole year :

As I plotted the heatmap of the cabs with the given data for the whole year , I saw that there is a point (approx. location 77.7 , 13.2) that is in great demand everytime. The point is also so far from the major city area , so I infered that this point must be the airport area (which i checked with the google maps and found the same), which has high demand for cabs throught the year irrespective of the time or the day of the week . So for further analysis I removed the points from this area for a better analysis of the data .

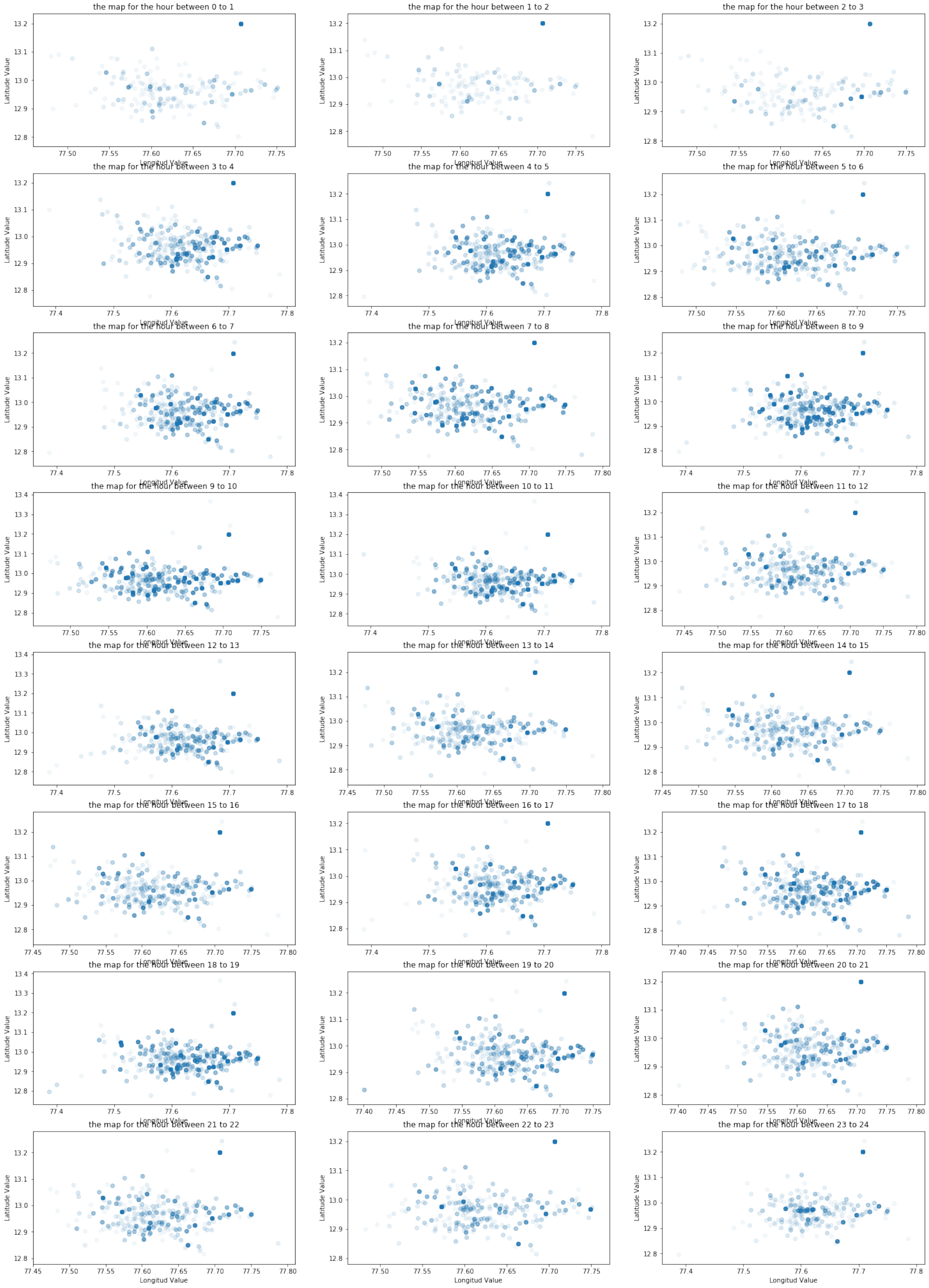
```
In [57]: plt.figure(figsize=(14,10))
         sns.kdeplot(df['from_long'],df['from_lat'],shade=True,cmap="Blues",n_levels=100)
         plt.title("Heatmap for cabs for whole year")
         plt.xlabel("Longitud Value")
         plt.ylabel("Latitude Value")
         plt.show()
```



1.0.2 Scatter Plots during the differnt hour of the day :

As we know that the demand of the cabs varies with the hours of the day , so as to check the same I have ploted the scatter plots on hourly basis . The alpha value taken is very less so as the points which coincide which each other gets a darker marker , to show the demand is high for those areas .

```
In [56]: ##plotting the data with hours of the day (0-23) :

         temp_hour=list()
         for i in range(0,24):
             temp_hour.append(df[df['from_date_hour']==i])

         plt.figure(figsize=(25, 36))
         for i in range(0,24):
             plt.subplot(8, 3, i+1)
             plt.scatter(temp_hour[i]['from_long'],temp_hour[i]['from_lat'],alpha=0.05)
             plt.title("the map for the hour between {} to {}".format(i,i+1))
             plt.xlabel("Longitud Value")
             plt.ylabel("Latitude Value")
         plt.show()
```



the map for the hour between 0 to 1
the map for the hour between 1 to 2
the map for the hour between 2 to 3
the map for the hour between 3 to 4
the map for the hour between 4 to 5
the map for the hour between 5 to 6
the map for the hour between 6 to 7
the map for the hour between 7 to 8
the map for the hour between 8 to 9
the map for the hour between 9 to 10
the map for the hour between 10 to 11
the map for the hour between 11 to 12
the map for the hour between 12 to 13
the map for the hour between 13 to 14
the map for the hour between 14 to 15
the map for the hour between 15 to 16
the map for the hour between 16 to 17
the map for the hour between 17 to 18
the map for the hour between 18 to 19
the map for the hour between 19 to 20
the map for the hour between 20 to 21
the map for the hour between 21 to 22
the map for the hour between 22 to 23
the map for the hour between 23 to 24
Latitude Value
Longitud Value
13.4
13.3
13.2
13.1
13.0
12.9
12.8
77.4
77.5
77.6
77.7
77.8
77.40
77.45
77.50
77.55
77.60
77.65
77.70
77.75
77.80

1.0.3 Interpretation :

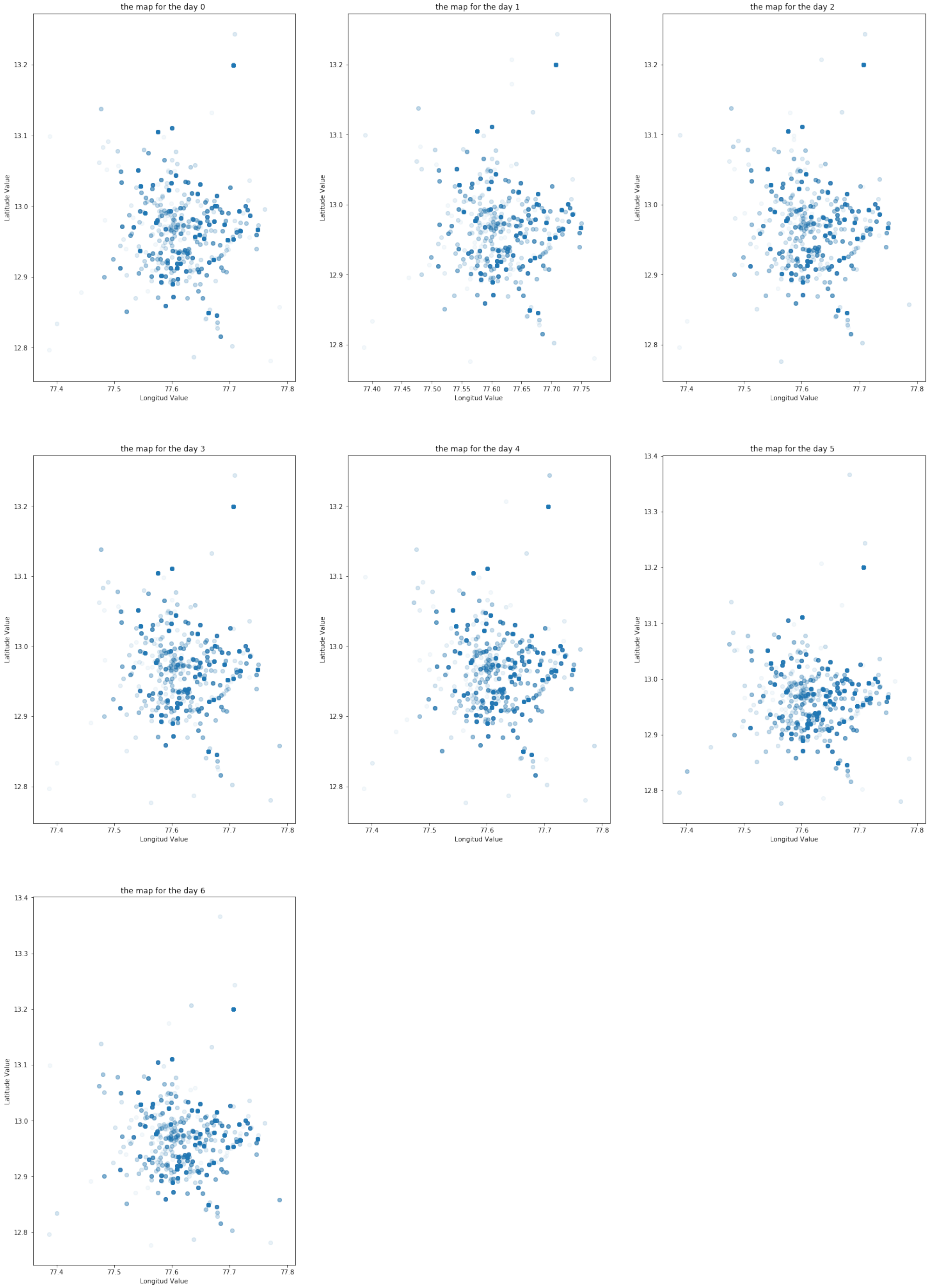
From the above plotted 24 hourly graphs we can interpret the following :

- The demand of cabs is very less after midnight like from 12 to 4
- The demand starts increasing after 4 and reach to its peak from 8 to 10 (office hours)
    - espcially in the areas (77.55 , 12.9) to (77.70 , 13.1)
    - areas like banashankari and whitefelid (used google maps to find approx lacation)
    - That may be the areas where major offices are located .
- The demand again decreses from its peak and shifts towards different parts of the city in the day hours .
- The demaand again take an increase in the evening hours in the office areas .
- The demand then decreses towards the end of the day .

1.0.4 Scatter Plots during the differnt days of the week :

As we know that the demand of the cabs varies with the days of the week , like during the weekdays the demand is high in office areas and during the weekend the demnad is high in the areas with restraunts cafes and pubs . So as to check the same I have ploted the scatter plots for the days of the week . The alpha value taken is very less so as the points which coincide which each other gets a darker marker , to show the demand is high for those areas

```
In [55]: #plotting the data with days of the week (0-6) :

         temp_day=list()
         for i in range(0,7):
             temp_day.append(df[df['from_day']==i])

         plt.figure(figsize=(25,36))
         for i in range(0,7):
             plt.subplot(3,3,i+1)
             plt.scatter(temp_day[i]['from_long'],temp_day[i]['from_lat'],alpha=0.05)
             plt.title("the map for the day {} ".format(i))
             plt.xlabel("Longitud Value")
             plt.ylabel("Latitude Value")
```


the map for the day 0
the map for the day 1
the map for the day 2
the map for the day 3
the map for the day 4
the map for the day 5
the map for the day 6
Latitude Value
Longitud Value
12.8
12.9
13.0
13.1
13.2
13.3
13.4
77.4
77.5
77.6
77.7
77.8
77.40
77.45
77.50
77.55
77.60
77.65
77.70
77.75

1.0.5 Interpretation :

From the above plotted graph we can interpret the following :

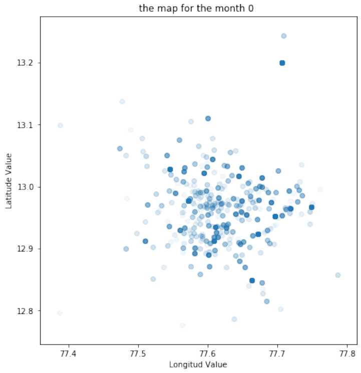
- The demand of the cabs increases in the areas (77.65 , 12.9) to (77.77 , 13.05) on weekdays
  - These are the office areas of the city
- The demand of the cabs increses in the ares (77.55 , 12.9) to (77.65 , 13.0) on weekends
  - These are areas like MG road , church street , koromomgala (used google maps to find approx locations)
  - These may be the areas with restraunts cafes and pubs , where people go to chill on weekends

1.0.6 Scatter plot during different months of the year :

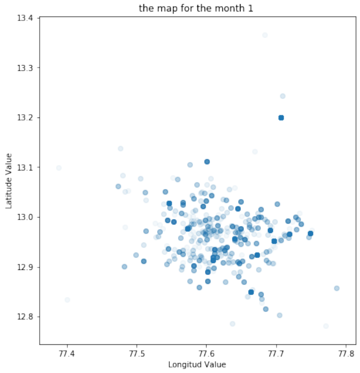
As we know the the months of may , june and july are months of vacation . So the cabs demand should increase in these months . Also the months like the month in which diwali is there or in the month of december during christmas vacation the demand should increase .

```
In [54]: # plotting data with months (0-11) :

         temp_month=list()
         for i in range(1,13):
             temp_month.append(df[df['from_month']==i])

         plt.figure(figsize=(25,36))
         for i in range(0,12):
             plt.subplot(4,3,i+1)
             plt.scatter(temp_month[i]['from_long'],temp_month[i]['from_lat'],alpha=0.05,cmap='
             plt.title("the map for the month {} ".format(i))
             plt.xlabel("Longitud Value")
             plt.ylabel("Latitude Value")
```

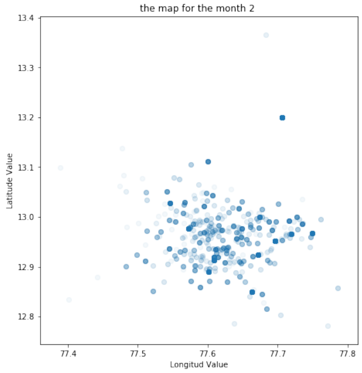
the map for the month 0



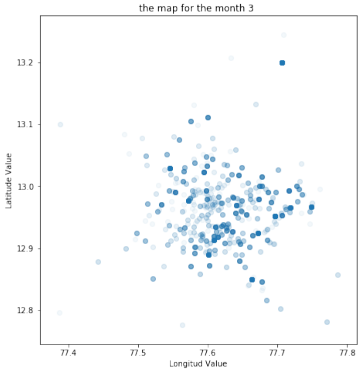
the map for the month 1



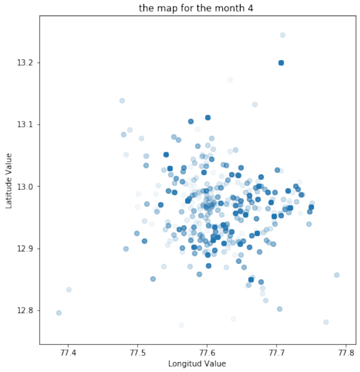
the map for the month 2



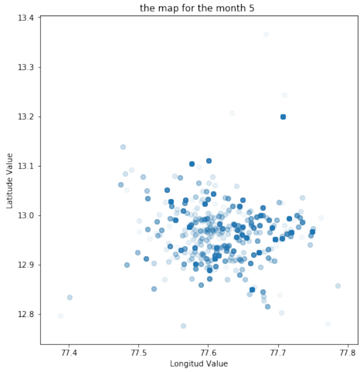
the map for the month 3



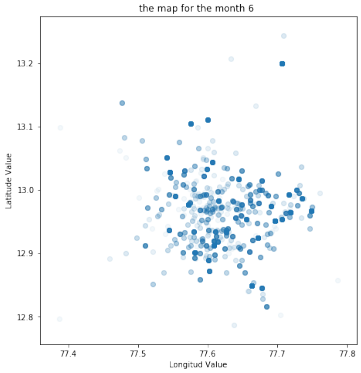
the map for the month 4



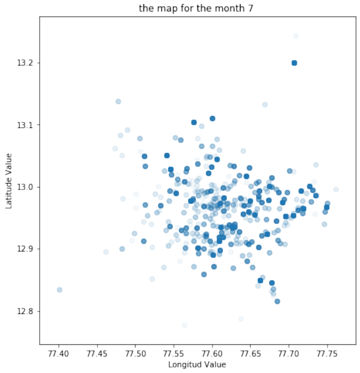
the map for the month 5



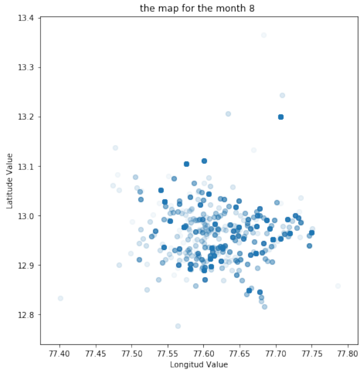
the map for the month 6



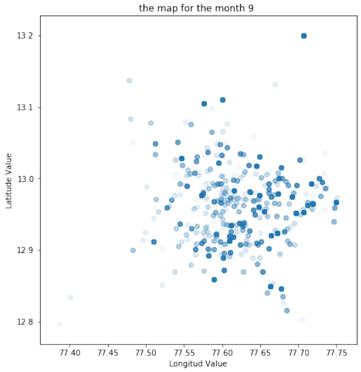
the map for the month 7



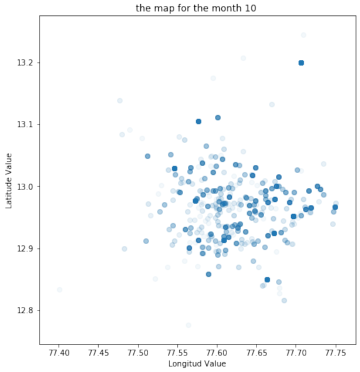
the map for the month 8



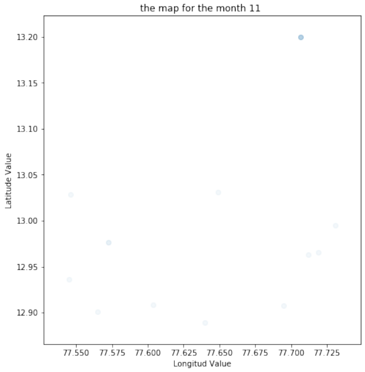
the map for the month 9



the map for the month 10



the map for the month 11

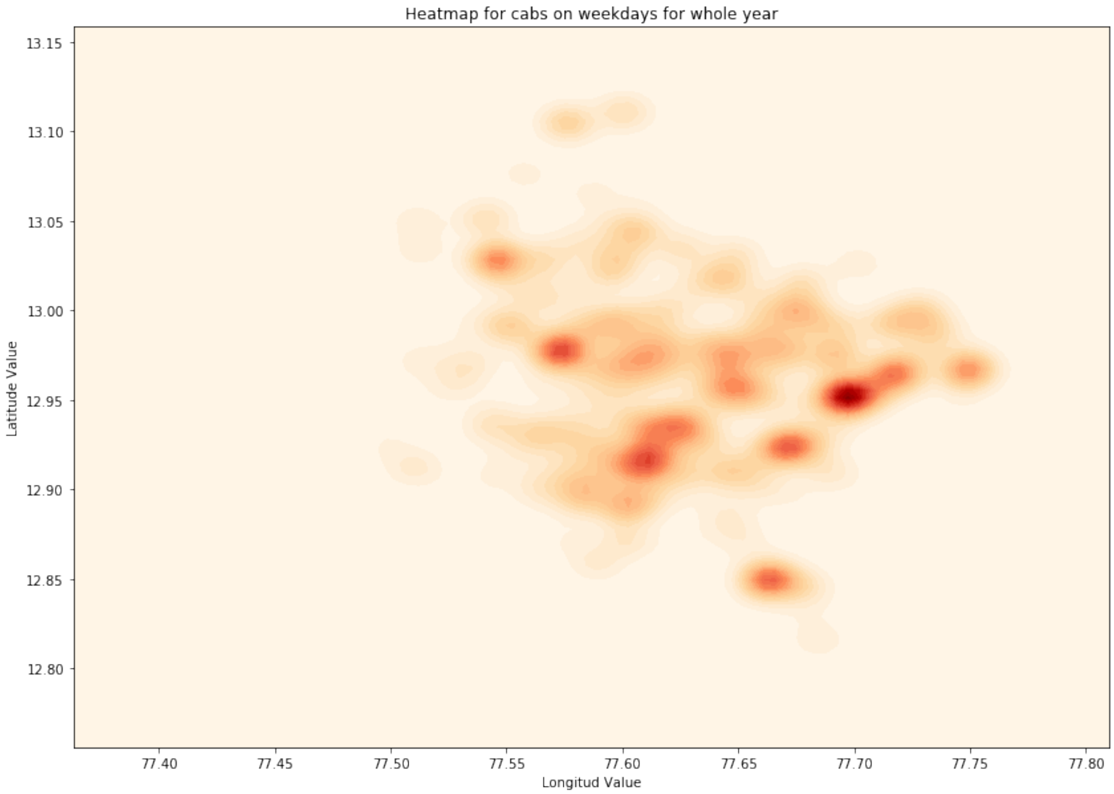


1.0.7 Interpretation :

- I interpreted that there is a slight increase in the demand of cabs for the month of june and july , thats due to the vactaions during that time .

```
In [13]: # data after removing the points around airport :

         datat=df[df['from_lat']<13.15]
```

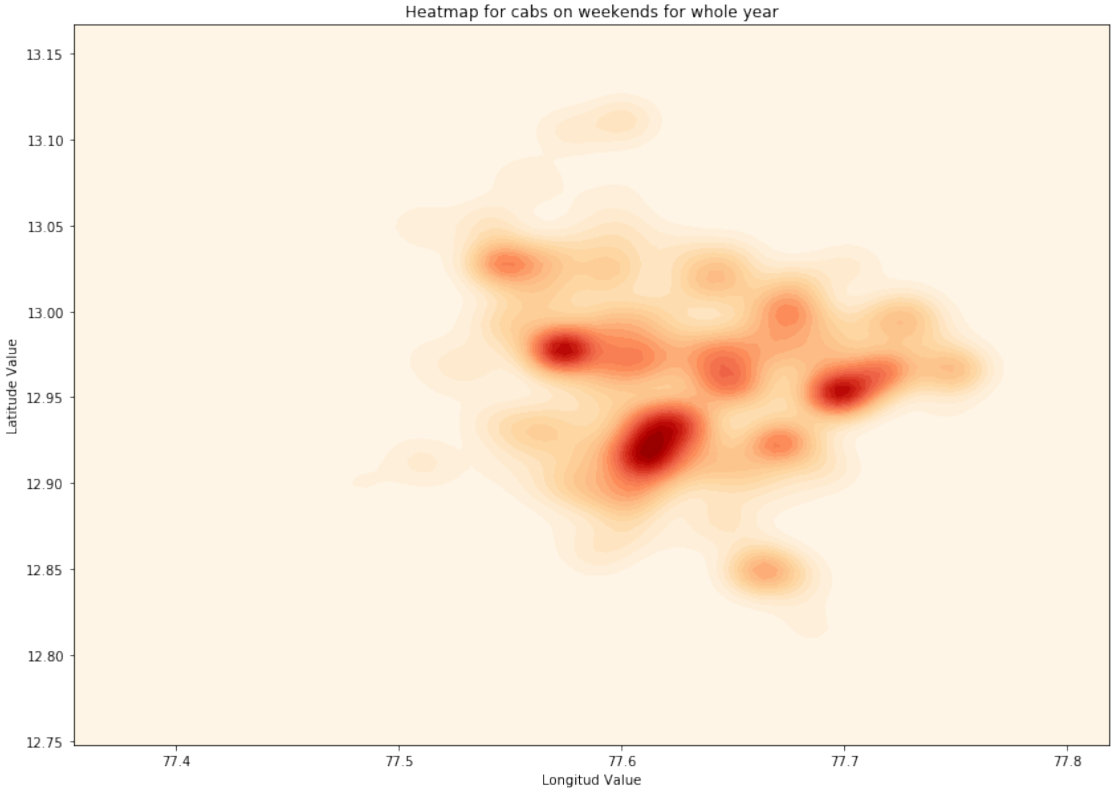
```
In [53]: # weekdays of the whole year :

         plt.figure(figsize=(14,10) )
         dtd=datat[datat['from_day']<5]
         sns.kdeplot(dtd['from_long'],dtd['from_lat'],shade=True,cmap="OrRd",n_levels=25)
         plt.title("Heatmap for cabs on weekdays for whole year")
         plt.xlabel("Longitud Value")
         plt.ylabel("Latitude Value")
         plt.show()
```



1.0.8 Interpretation :

The plot above is a heat map of all the cabs showing for the weekdays of the year. I intepretedd that there are sevral hotspots during the day time , the hotspots are basically the office areas like whitefeild

```
In [52]: # weekends of the whole year :

         plt.figure(figsize=(14,10))
         dtwe=datat[datat['from_day']>5]
         sns.kdeplot(dtwe['from_long'],dtwe['from_lat'],shade=True,cmap="OrRd",n_levels=25)
         plt.title("Heatmap for cabs on weekends for whole year")
         plt.xlabel("Longitud Value")
         plt.ylabel("Latitude Value")
         plt.show()
```

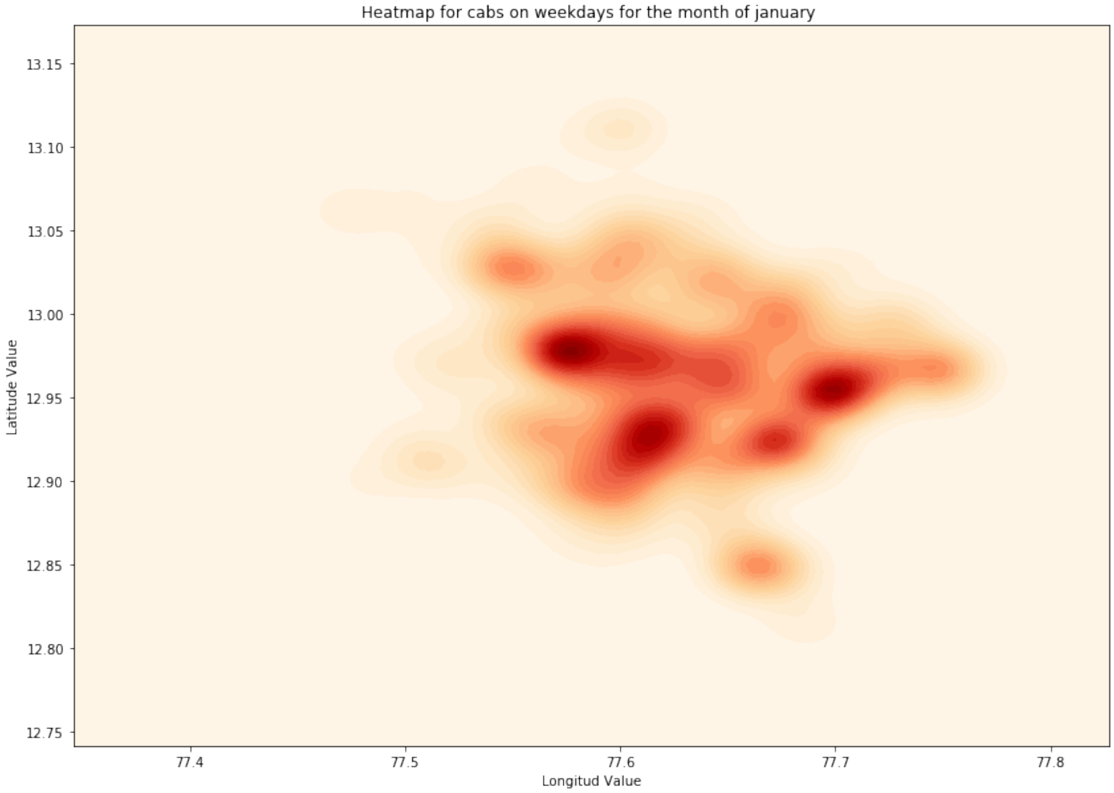


1.0.9 Interpretation :

The plot above is a heat map of all the cabs showing for the weekends of the year. I intepretedd that there are sevral hotspots during the weekend , the hotspots are basically the areas with restraunts, cafes and pubs like MG road and church street .

```
In [51]: # data for weekdays plot for the month of january :

         # variable to change the month of the year :(change the value(1-12) to change the mon
         MONTH=1

         plt.figure(figsize=(14,10))
```

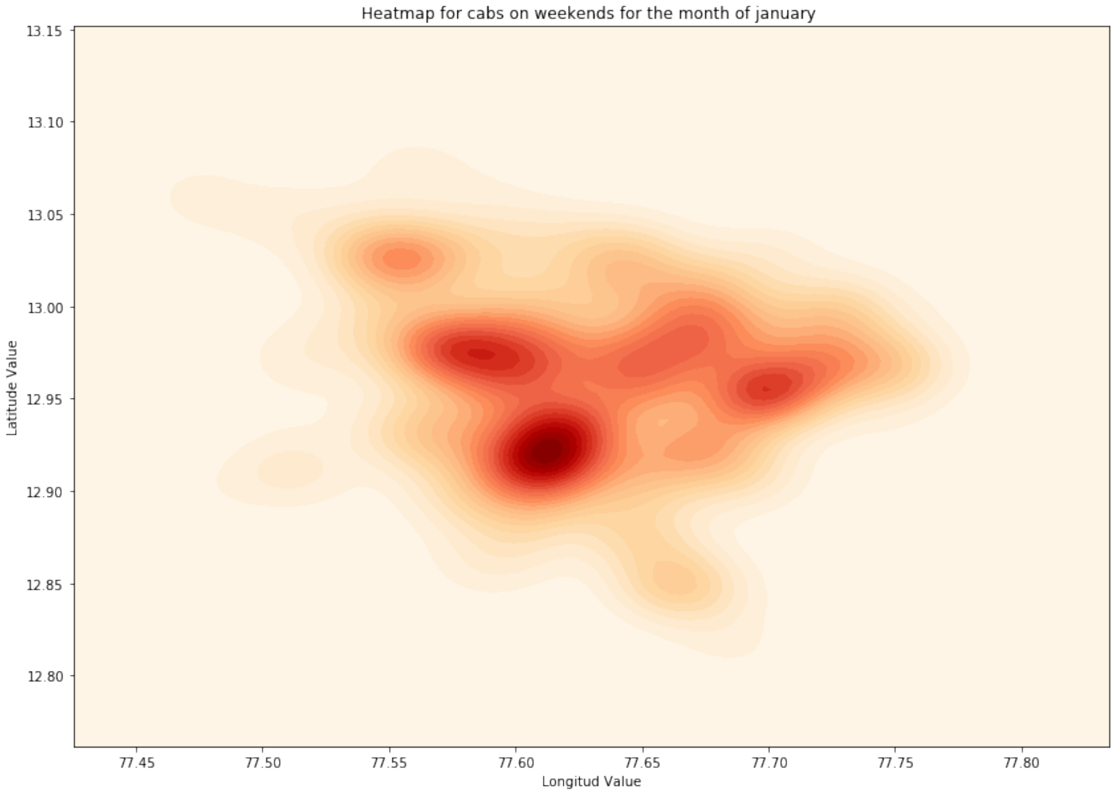
```
dt=datat[(datat['from_day']<5)&(datat['from_month']==MONTH)]
sns.kdeplot(dt['from_long'],dt['from_lat'],shade=True,cmap="OrRd",n_levels=25)
plt.title("Heatmap for cabs on weekdays for the month of january ")
plt.xlabel("Longitud Value")
plt.ylabel("Latitude Value")
plt.show()
```



1.0.10 Interpretation :

To check the hotspots more clearly I plotted the heatmap for the weekday of a single month. I interpreted that on weekdays the office areas are the most in demand

```
In [50]: # data for weekend plots for the month of january :

         # variable to change the month of the year :(change the value(1-12) to change the mon
         MONTH=1

         plt.figure(figsize=(14,10))
         dtw=datat[(datat['from_day']>=5) & (datat['from_month']==1)]
         sns.kdeplot(dtw['from_long'],dtw['from_lat'],shade=True,cmap="OrRd",n_levels=25)
         plt.title("Heatmap for cabs on weekends for the month of january ")
         plt.xlabel("Longitud Value")
         plt.ylabel("Latitude Value")
         plt.show()
```



1.0.11 Interpretation :

To check the hotspots more clearly I plotted the heatmap for the weekend of a single month. I interpreted that on weekdays the areas with cafes and restraunts are the most in demand

1.0.12 Type of travel :

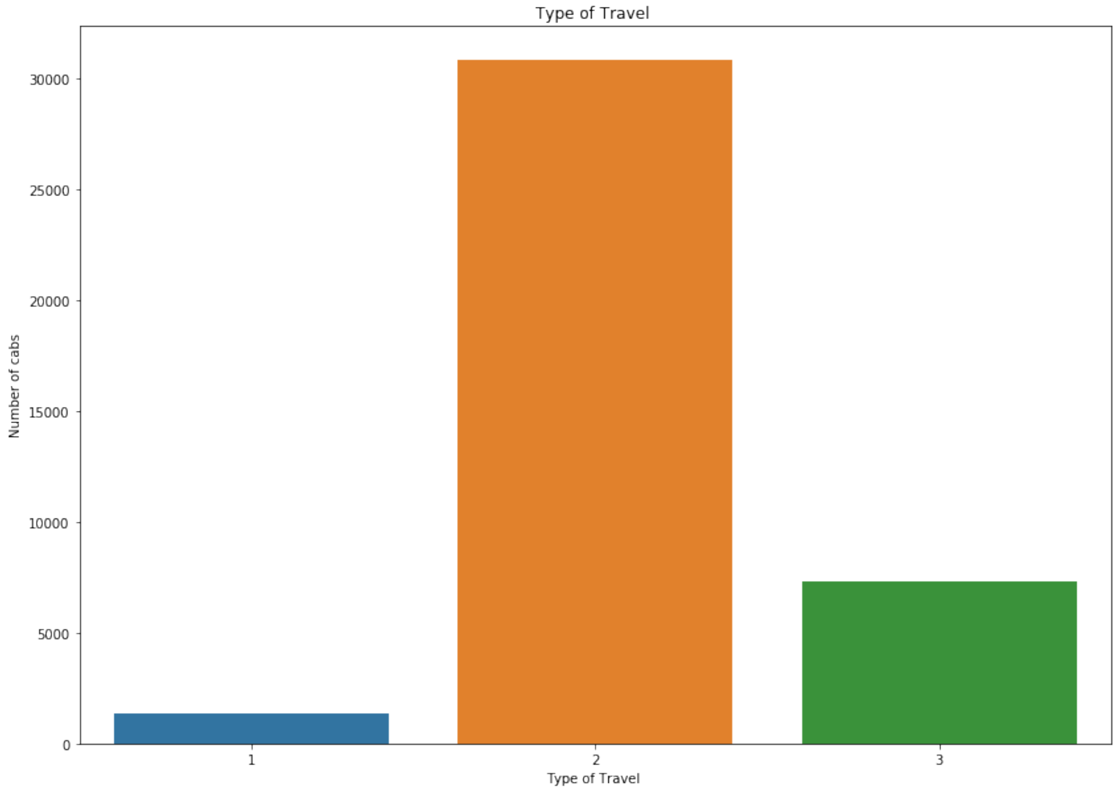
```
In [18]: #counter for travel types :

         Counter(datat['travel_type_id'])

Out[18]: Counter({2: 30849, 1: 1331, 3: 7318})
```

Travel type Id gives us the type of travel viz.

1. long distance
2. point to point
3. hourly rental

I used a counter to count the no of trips of each types and plotted a bar-chart.

```
In [60]: # barplot showing travel type :

         plt.figure(figsize=(14,10))
         sns.countplot(datat['travel_type_id'])
         plt.title("Type of Travel ")
         plt.xlabel("Type of Travel ")
         plt.ylabel("Number of cabs")
         plt.show()
```



1.0.13 Interpretation :

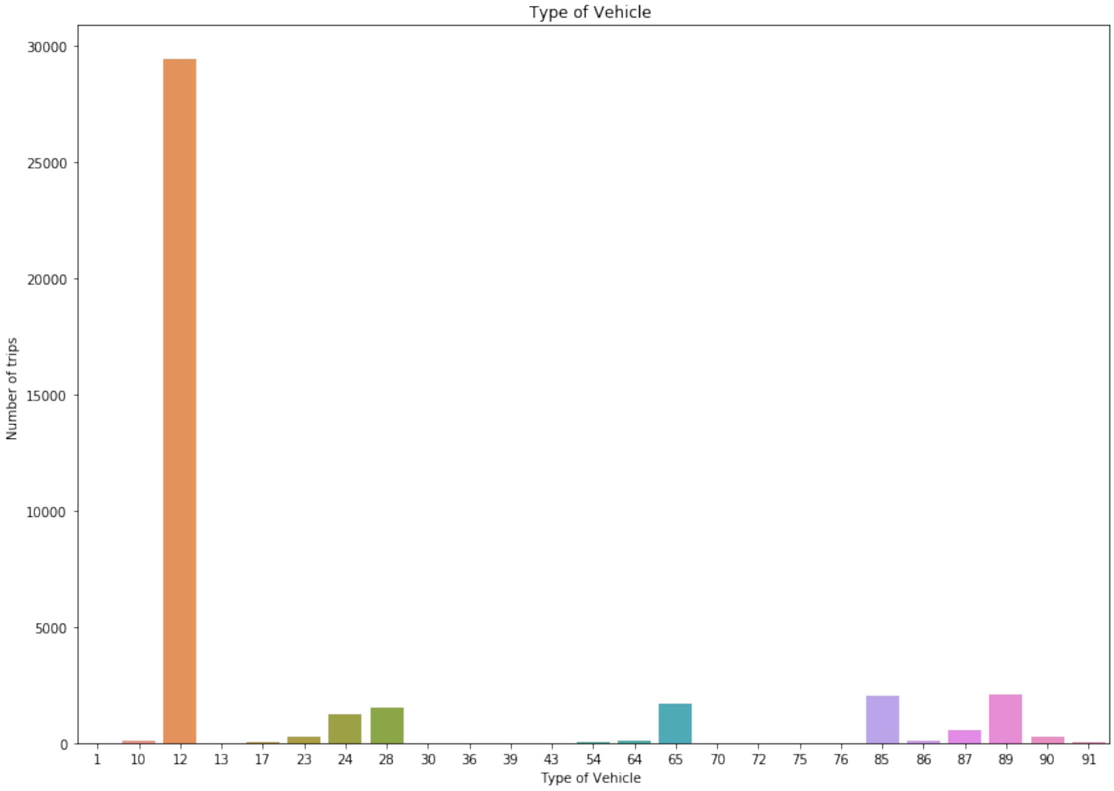
The above bar graph shows the distribution on the basis of travel type . I interpreted that the most prefered travel type is point to point and the least preffered travel type is long distance . While some of the trips were also there when cabs were booked on hourly rental.

1.0.14 Vehicle type :

Vehicle model ID gives us the type of vehicle used fro the trip . The types are not mentioned in the data description but these may be like 1. hatchback 2. sedan 3. premium sedan 4. SUV etc

```
In [62]: # bar plot showing vehicle type used for rides :
```

```
plt.figure(figsize=(14,10))
sns.countplot(datat['vehicle_model_id'])
plt.title("Type of Vehicle ")
plt.xlabel("Type of Vehicle ")
plt.ylabel("Number of trips ")
plt.show()
```



1.0.15 Interpretation :

From the bar chart above I interpreted that most common used vehicle type is type 12 . That must be a hatchback or a carpool

1.0.16 Number of rides per user :

I used a counter for counting the number of trips a user has taken . The trips varies from 471 being the maximum and 1 being the minimum .

```
In [42]: # counter for number of rides per user id  :

         print(Counter(datat['user_id']))
```

1.0.17 Number of users and number of rides :

A counter to calculate the number of person with the number of rides . There is a lot of variation in the data . ike most of the people has taken the ride once or twice or thrice after that the number of rides decreases.

```
In [22]: #counter for number of users with rides :

         no_of_rides = list(Counter(datat['user_id']).values())
         print(Counter(no_of_rides))
```

```
Counter({1: 15026, 2: 2844, 3: 1057, 4: 535, 5: 331, 6: 186, 7: 138, 8: 93, 9: 72, 10: 54, 11:
```

1.0.18 Mode of booking :

A counter to calculate the mode of booking so as to infer that what mode of booking do people generally use for booking . I infered that online booking is more popular than mobie site booking.

```
In [23]: # counter for the mode of booking :

         print(Counter(datat['mobile_site_booking']))
         print(Counter(datat['online_booking']))
```

```
Counter({0: 37906, 1: 1592})
Counter({0: 26050, 1: 13448})
```

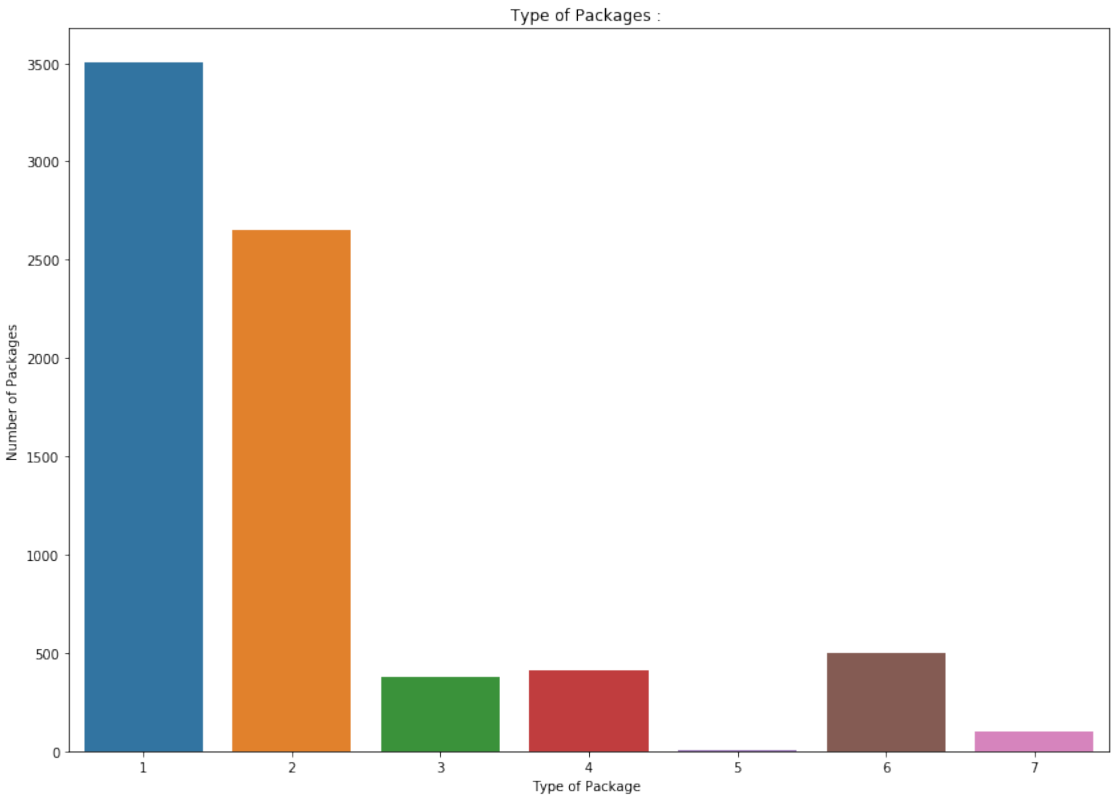
1.0.19 Package type :

A counter to count the package type using package ID : a bar chart is plotted to show the distribution

```
In [24]: # package type :

         datap=pd.read_csv("data.csv",dtype=object)
         package_data=datap.dropna(axis=0,subset=['package_id'])
         print(Counter(package_data['package_id']))
```

```
Counter({'1': 3503, '2': 2651, '6': 502, '4': 412, '3': 375, '7': 101, '5': 6})
```

```
In [63]: # barplot for type of package :

         plt.figure(figsize=(14,10))
         sns.countplot(package_data['package_id'])
         plt.title("Type of Packages : ")
         plt.xlabel("Type of Package ")
         plt.ylabel("Number of Packages")
         plt.show()
```



The above barchart shows the distribution of package type being transported and I infered that mostly the package of type 1 and type 2 are transported

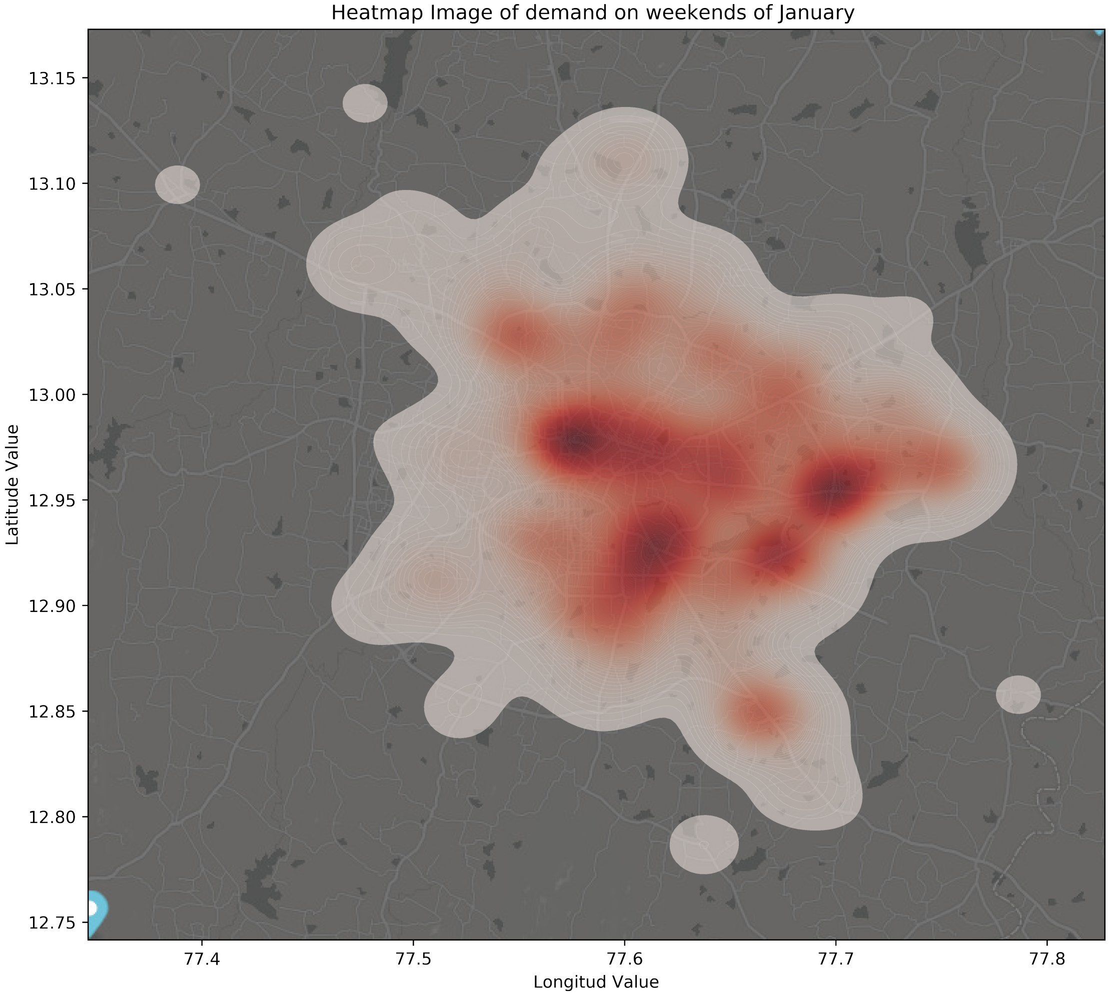
1.0.20 Plotting on a map image :

For further details and to make the visualisation good I used mapbox , took the coordinates and take a screenshot of the map between the coordinates . Then I used that image as a background to plot th heat map .

```
In [49]: import matplotlib.image as mpimg
         img = mpimg.imread('./dark.jpeg')

         plt.figure(figsize=(14,10),dpi=600)
         sns.kdeplot(dt['from_long'],dt['from_lat'],
                     gridsize=200,shade_lowest=False,
                     shade=True,cmap='Reds',n_levels=100,alpha=0.5)

         ax=plt.gca()
         xlims=ax.get_xlim()
         ylims=ax.get_ylim()
         print(xlims,ylims)
         plt.imshow(img,zorder=0,extent=[*xlims,*ylims],alpha=0.75)
         plt.title("Heatmap Image of demand on weekends of January ")
```

```
    plt.xlabel("Longitud Value")
    plt.ylabel("Latitude Value")
    plt.show()
```

```
(77.34602521692977, 77.82732478307022) (12.741570158172719, 13.172909841827282)
```



1.0.21 Geo - Surge Strategy :

The Geo-surge strategy should be based on

- Demand of the cabs in that particular area .
- The availability of cabs in that particular area .
- The location i.e where the pickup location is and where the drop location is to see wheather the pickup area has high demand or whether we will get any other customer from the drop area .

- The time of the booking i.e the time when the demand is high the price should go high and vice versa .
- Also at night time the prices should be kept high, as there is no other option to travel so the customer will book a cab anyways.

For eg ,

- If I am in the office areas on weekdays and I want a cab from office to way back home , I am in the area of great demand and also the time is the office leaving hours so the prices can be increased accordingly like 1.5 times or 1.8 times and as I am tierd and I want to reach home fast I will book a cab at a higher price also .
- If I am in the cafe or pub areas on weekends and I am booking a cab late in night for going back home ,the prices can be increased accordingly to 2 times or 2.5 times and as the time is late and I have to reach home I will anyways book a cab at a higher price .

The long term and short term effects of surge-pricing : In the short term, surge pricing substantially affects the rate of demand, while long-term use could be the key to retaining or losing customers

1.0.22 Strategy to reduce cancellation :

Why basically the cancellation occurs :
The main reason why cancellation occurs are -

- The high demand and not enough supply for those demands
- The rival comapany is providing the cab in lesser time

To reduce these cancellation what we can do is , we predicted the area of higher demands at a given time of the day . So we can increase the supply of cabs in those areas . Also a prompt can be given to the drivers that are currently free in some other areas nearby to go to the higly demanding areas so as to meet the demand.

Also at the time of high demand what human mentality is that we book a cab which is coming for us in lesser time rather than to see how much more price it is taking , so an effective surge pricing can also be implemented so as to increase the revnue of the company

Also this will help the company to reduce its cost on fuel ,the customer will get cabs in the required area or else he will book a cab in some far area and then the cab has to travel to the required area for the customer .