UBER DATA ANALYSIS

Location: Bangalore Quarter-3

Objective: -

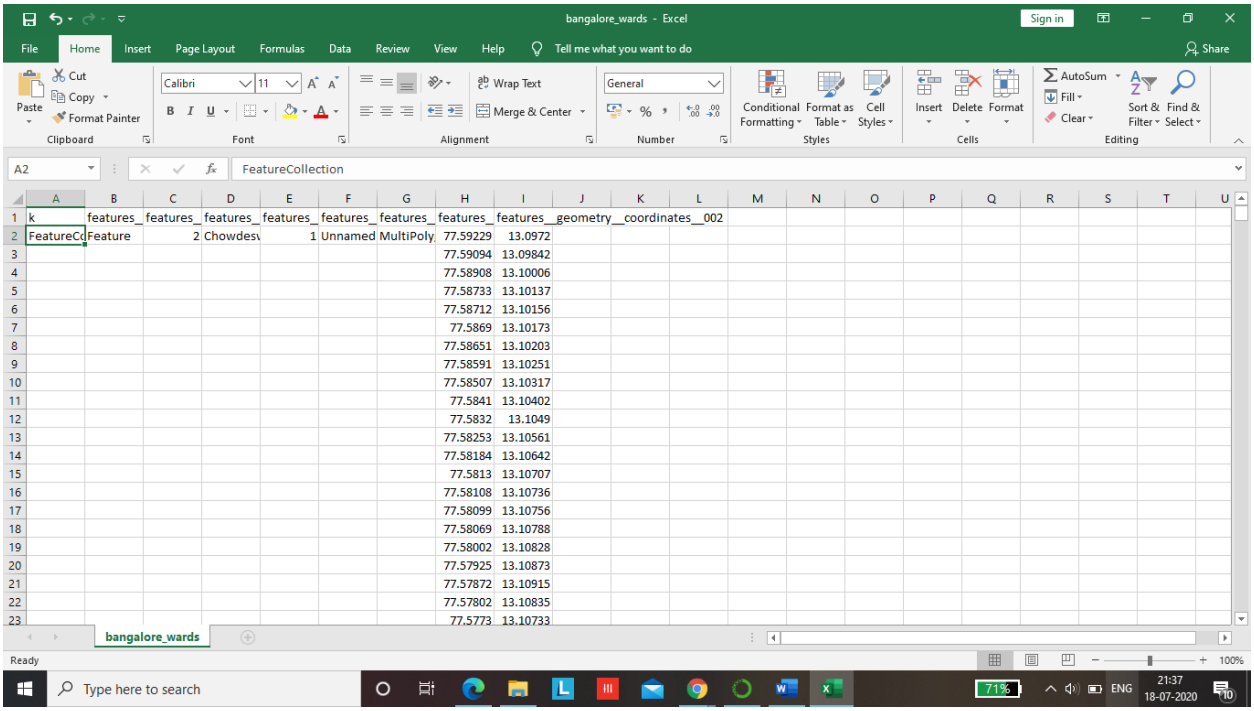
1. Given a place in the city, find all places within 30 minutes of driving distance
2. Given an origin and a destination, find the best and worst day of the week for commuting.

Data Resources: -

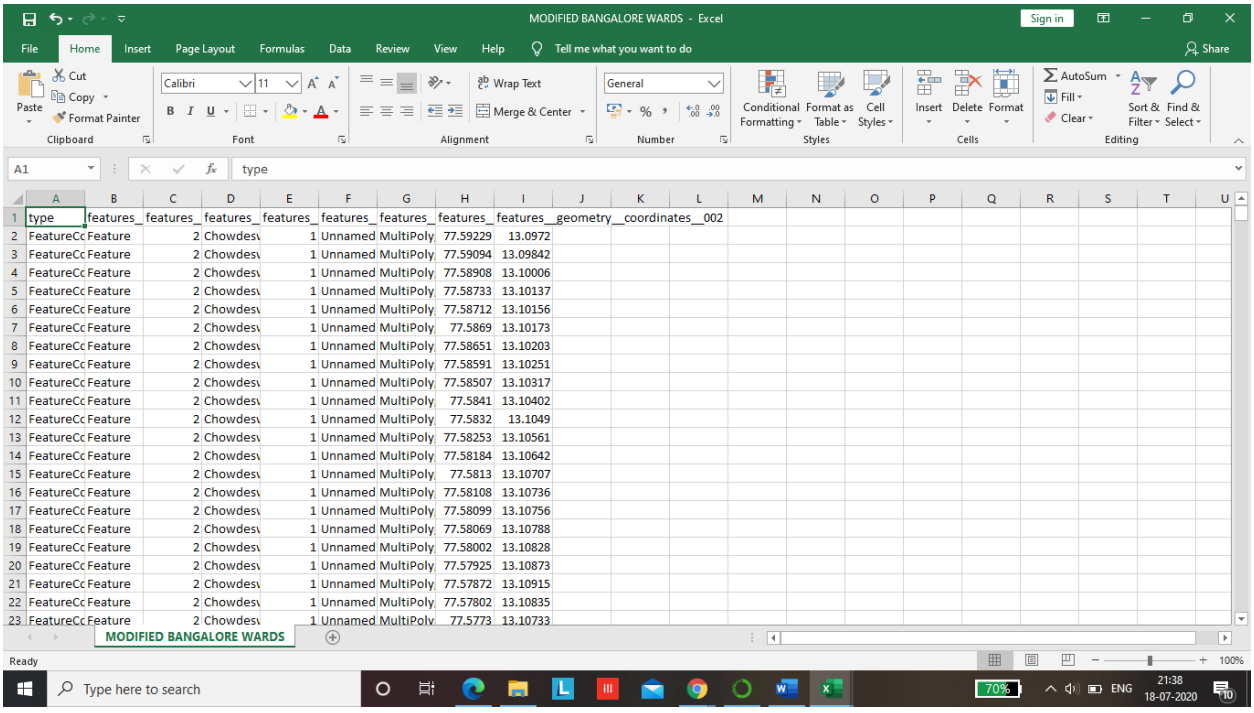
1. Data was exclusively downloaded from Ubermovement.com for they city of Bangalore, Quarter-3. The *Travel Times* Uber Movement data for Bangalore.
2. Weekly, Hourly, Weekday, Weekend, Monthly, Wards data was downloaded to study and analyse the commute behaviour of UBER in the city of Bangalore.

Converting Raw Data.

1. The Wards data received has void columns and empty rows till we get the value for the next feature.
2. Working with this data, would fetch us very poor visualisations. So, we first convert the json file to csv file online.
3. Following this we add alternate blank columns in between the data columns and equate a formula such that if length of the right column is greater than the right column, so the right one gets pasted else the left one.
4. Following this, step we fill our entire empty void datasets as shown in the below figure and save the new dataset as modified dataset.



The data before filling in the spaces.



The modified data wards Bangalore.

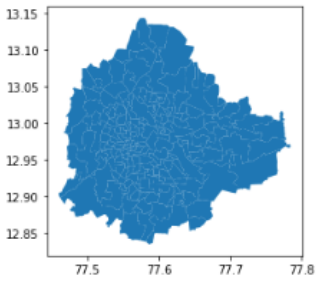
Plotting the ward dataset in Bangalore Quarter-3.

Out[6]:

	WARD_NO	WARD_NAME	MOVEMENT_ID	DISPLAY_NAME	geometry
0	2	Chowdeswari Ward	1	Unnamed Road, Bengaluru	MULTIPOLYGON (((77.59229 13.09720, 77.59094 13...
1	3	Atturu	2	9th Cross Bhel Layout, Adityanagar, Vidyaranya...	MULTIPOLYGON (((77.56862 13.12705, 77.57064 13...
2	4	Yelahanka Satellite Town	3	15th A Cross Road, Yelahanka Satellite Town, Y...	MULTIPOLYGON (((77.59094 13.09842, 77.59229 13...
3	51	Vijnanapura	4	SP Naidu Layout 4th Cross Street, SP Naidu Lay...	MULTIPOLYGON (((77.67683 13.01147, 77.67695 13...
4	53	Basavanapura	5	Medahalli Kadugodi Road, Bharathi Nagar, Krish...	MULTIPOLYGON (((77.72899 13.02081, 77.72994 13...

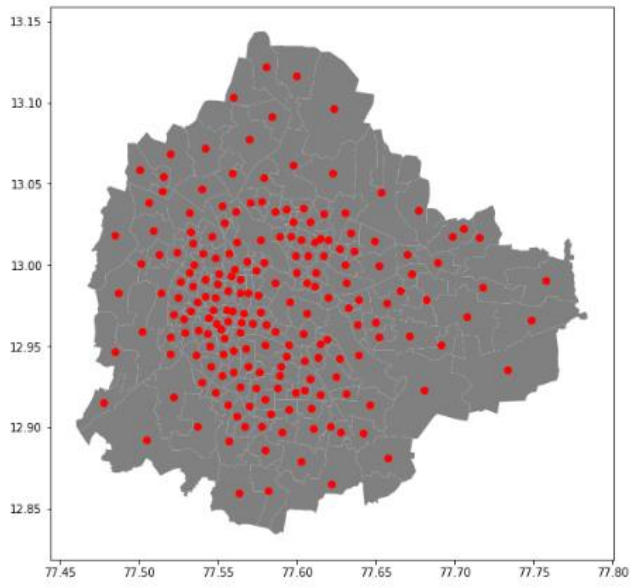
```
In [7]: bglr2.plot()
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb20b8270f0>
```



Further Plotting the ward dataset with its centroid gives us a wholistic view of the Bangalore Quarter-3.

```
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb20b44f9b0>
```



DATA DICTIONARY

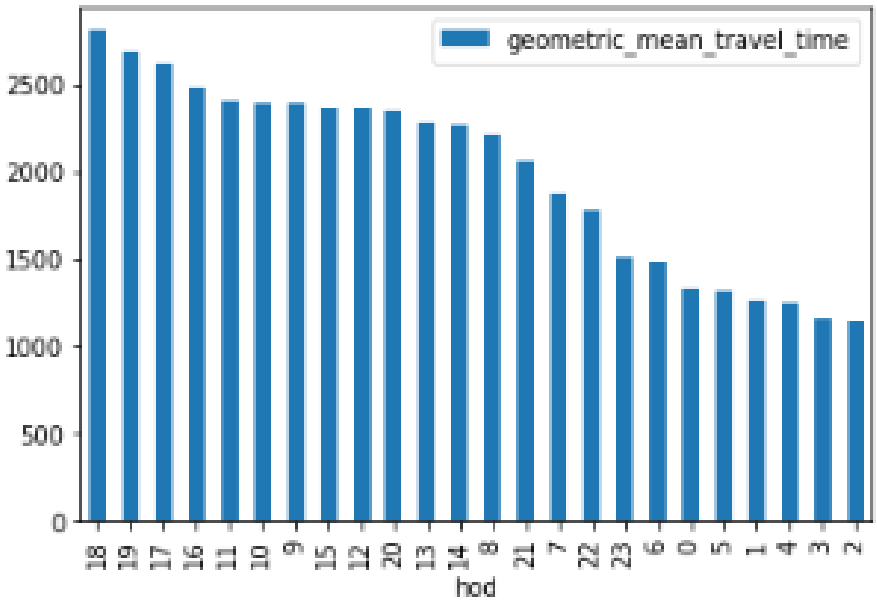
```
[13]: bglr1=gpd.read_file('/content/bangalore-wards-2019-3-All-HourlyAggregate.csv')
      bglr1.head()
```

t[13]:

	sourceid	dstid	hod	mean_travel_time	standard_deviation_travel_time	geometric_mean_travel_time	geometric_standard_deviation_travel_time	geometry
0	88	33	12	2022.22	665.53	1944.1	1.31	None
1	163	16	14	3159.21	680.03	3095.19	1.22	None
2	160	46	14	3943.34	581.21	3900.2	1.16	None
3	162	26	14	3193.64	615.47	3143.88	1.19	None
4	84	73	12	801.05	324.42	766.42	1.31	None

1. **Sourceid- The sourceid stands as a notation for the starting ward when the trip begins.**
2. **Destid-The destid stands as a notation for the ending ward where the passenger is dropped.**
3. **Hod- Acronym of Hour of the Day. This tells us the hour of the day.**
4. **Mean_travel_time- Aggregate Travel Time to Commute (in Seconds)[MAX]**
5. **Standard_deviation_travel_time-The amount by which the aggregate time travel may be deviated in seconds.**
6. **Geometric_mean_travel_time-The Central Time travel to commute{in seconds)[MIN]**
7. **Geometric_standard_deviation_travel_time-The amount by which the Central Time travel to commute deviates in seconds.**
8. Geometry-It simply indicates the geometry of the polygon/ward.

We're likely to get an estimate of how the delay, the travel times of commutes vary at different hours of the day through the various visualizations underneath with the objective of optimizing our travels, with an explicit inclusion of extension for a better obtained Uber Movement with suggestive plans for an even more passenger friendly travel.



The Geometric mean travel time varies with different hours of the day.

The mean travel time varies with different hours of the day.

Furthermore, we bin the dataset 'Hour Of the Day' into segments for better understanding of commute behavior with the different hours of the day.

```
In [71]: bglr1['hod']=pd.cut(bglr1.hod,
                             bins=[0,4,8,12,16,24],
                             labels=['Latenight','Early_morning','morning','afternoon','evening'])
```

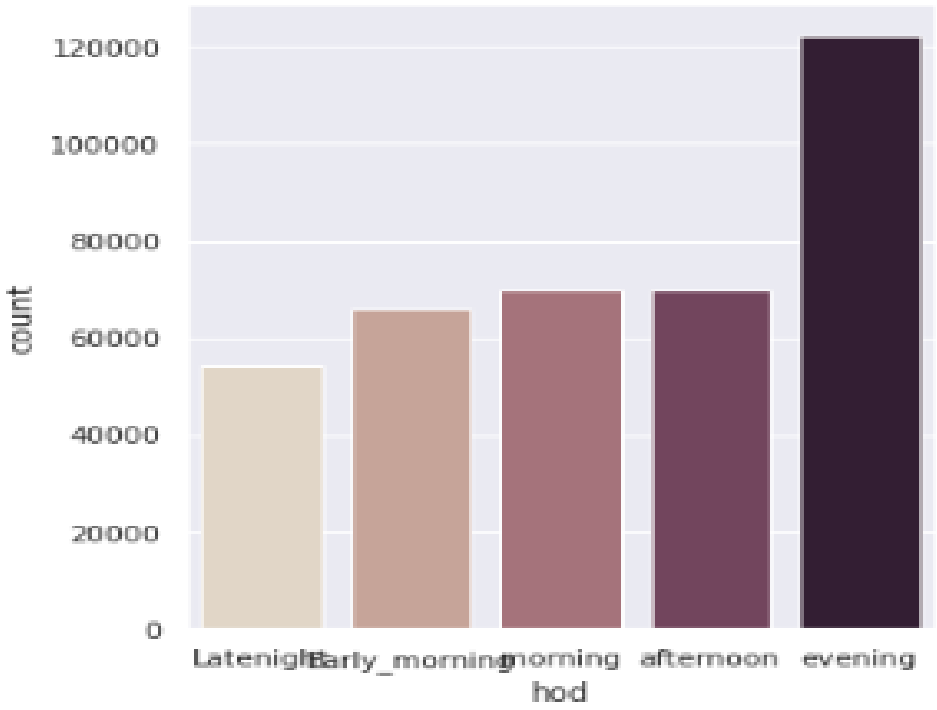
Incorporating the binning of dataset,gives the HOD column a categorical division from numeric.

In [69]:
```
bglr1
```

Out[69]:

	sourceid	dstid	hod	mean_travel_time	standard_deviation_travel_time	geometric_mean_travel_time	geometric_standard_deviation_travel_time	geo
0	88.0	33.0	morning	2022.22	665.53	1944.10	1.31	
1	163.0	16.0	afternoon	3159.21	680.03	3095.19	1.22	
2	160.0	46.0	afternoon	3943.34	581.21	3900.20	1.16	
3	162.0	26.0	afternoon	3193.64	615.47	3143.88	1.19	
4	84.0	73.0	morning	801.05	324.42	766.42	1.31	
...	...	...	...	...	...	...	...	
398337	55.0	51.0	afternoon	1351.52	388.79	1309.87	1.27	
398338	59.0	11.0	afternoon	1334.32	411.91	1277.79	1.34	
398339	46.0	159.0	evening	4468.16	822.11	4394.74	1.20	
398340	58.0	175.0	afternoon	3637.46	786.0	3552.02	1.25	
398341	56.0	195.0	afternoon	2310.67		0.00	0.00	

398342 rows x 8 columns

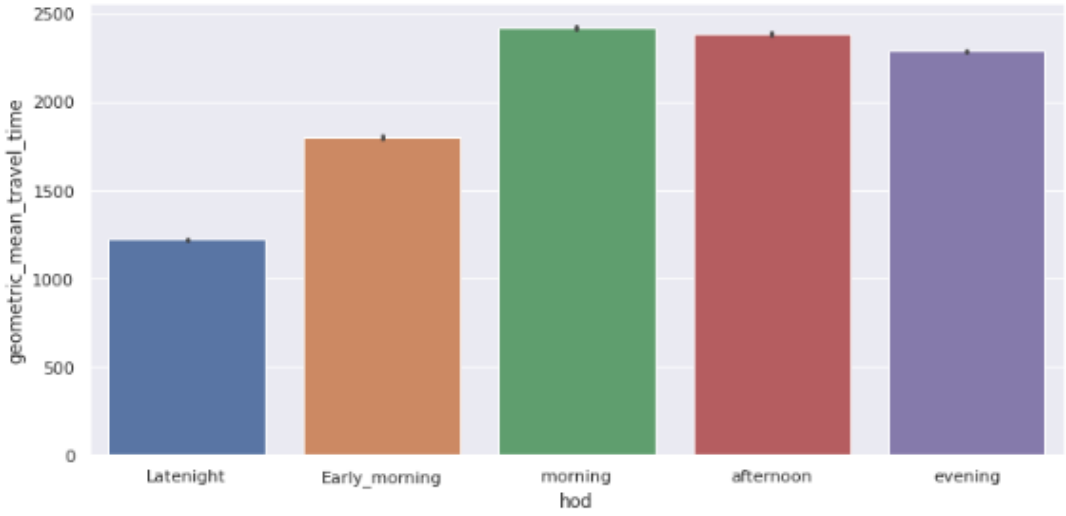


The above bar plot gives us an idea about the order of maximum commutes at different binned hours of the day.

For instance, Evening is the time when maximum Uber movements occur for office-goers,school children, workers for they return to their respective homes'.

The morning schedule is further divided into early morning, morning, afternoon which certainly distributes the commute movements by various professionals or children dependent on the population that's dependency on Uber for commuting.

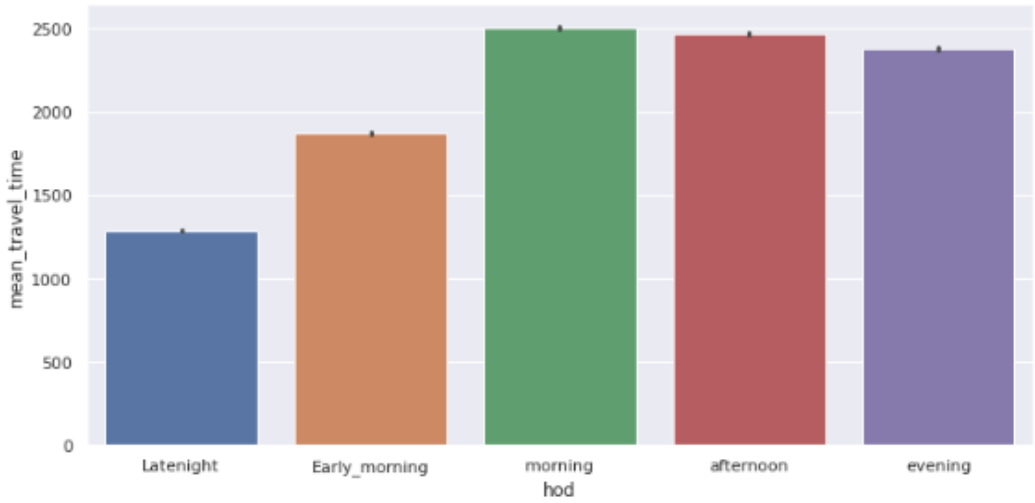
Late Night with the lowest count is obvious for we commute at night in case of catching hold of a train, or late night workers, or attending to an event or a party.



The Geometric Mean Travel Time with different hours of the day is highest for morning, for in the morning, commute is high and there's atraffic for it happens to be the hour of rush and commutes takes highest time for this hour travel.

Following this is the afternoon , evening where office-goers, school children return back.

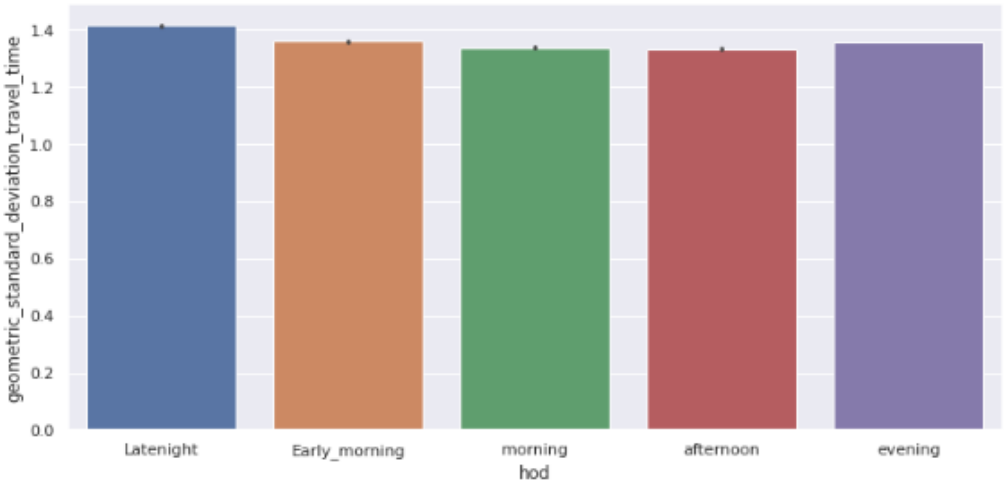
Least is for Late-night for Uber Movement operates for firstly a constraint timing at night with limited acceptance of commutes. late night travel arises in cases of emergency, party or an event, or professionals working overnight shift.



The Mean Travel Time with different hours of the day is highest for morning, for in the morning, commute is high and there's a traffic for it happens to be the hour of rush and commutes takes highest time for this hour travel.

Following this is the afternoon , evening where office-goers, school children return back.

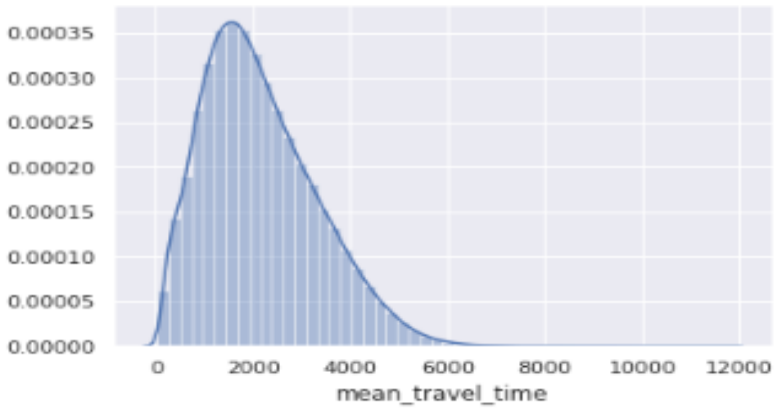
Least is for Late-night for Uber Movement operates for firstly a constraint timing at night with limited acceptance of commutes. Late-night travel arises in cases of emergency, party or an event, or professionals working overnight shift.

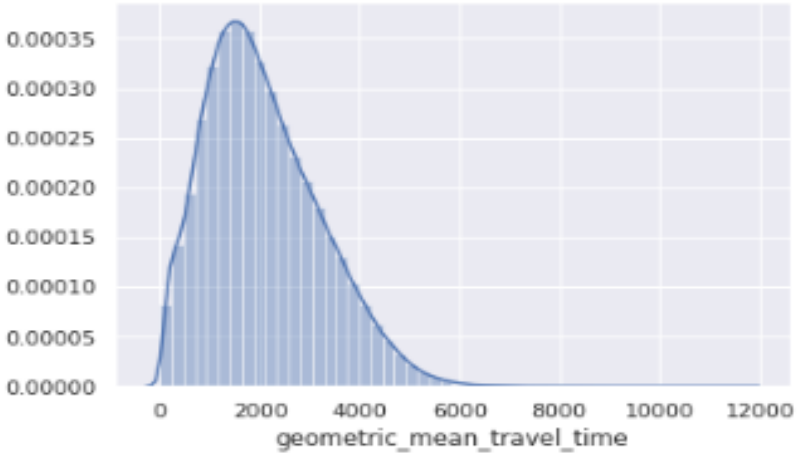


The deviation time is least for morning, following I the afternoon, early morning, evening and then the late night.

Morning, Early Morning, Afternoon are times when the highways, roads, lanes, bridges open.

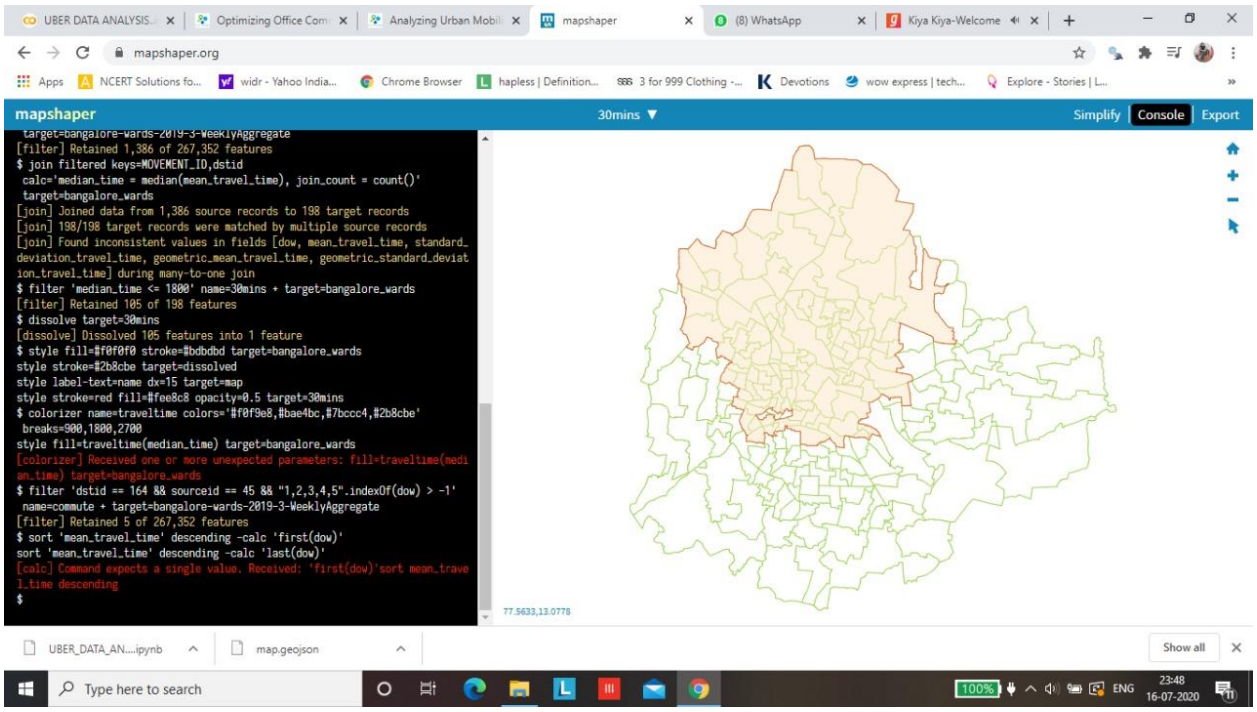
Whereas as the evening approaches, and the late night, the bridges, the highways, the roads have a restricted movement. This results to greater deviation to the standard time for the driver opts for longer routes.

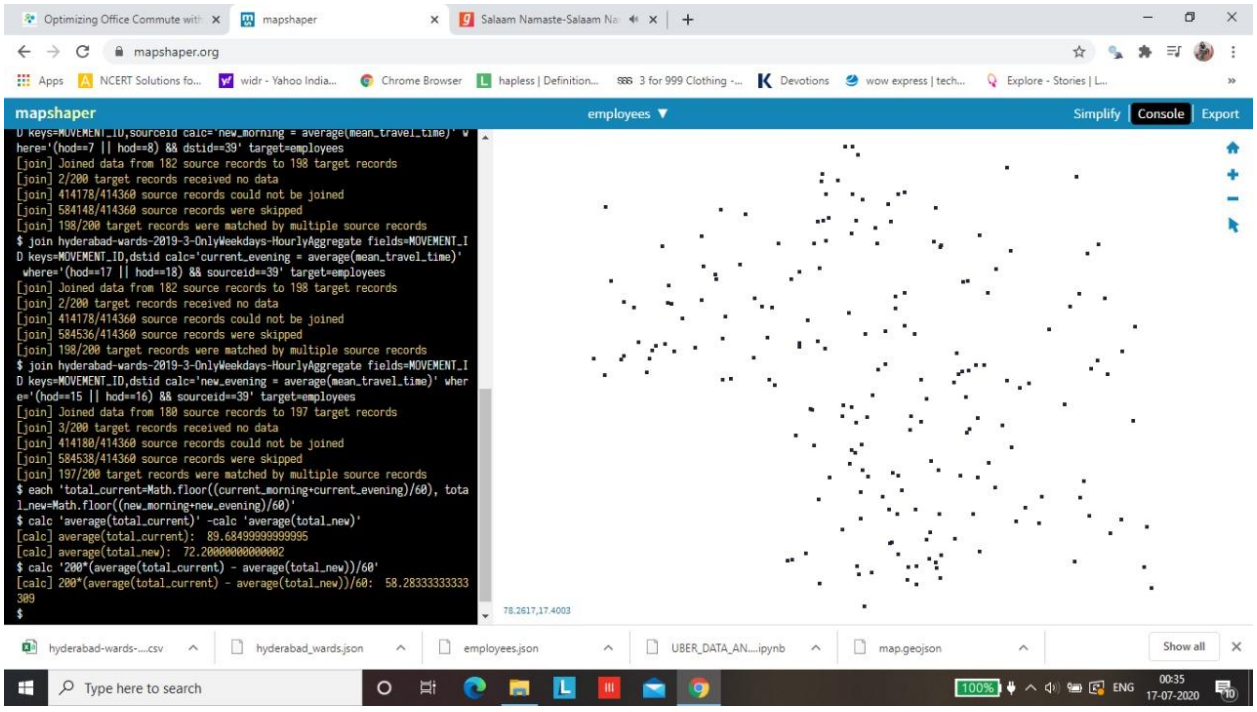




This depicts the variation in the data distribution of the mean_travel-time and the geometric_mean_travel_time .

MAPSHAPER.


mapshaper
30mins
Simplify
Console
Export
mapshaper.org
target=bangalore-wards-2019-3-WeeklyAggregate
[filter] Retained 1,386 of 267,352 features
$ join filtered keys=MOVEMENT_ID,dstid
calc='median_time = median(mean_travel_time), join_count = count()'
target=bangalore_wards
[join] Joined data from 1,386 source records to 198 target records
[join] 198/198 target records were matched by multiple source records
[join] Found inconsistent values in fields [dow, mean_travel_time, standard_deviation_travel_time, geometric_mean_travel_time, geometric_standard_deviation_travel_time] during many-to-one join
$ filter 'median_time <= 1800' name=30mins + target=bangalore_wards
[filter] Retained 105 of 198 features
$ dissolve target=30mins
[dissolve] Dissolved 105 features into 1 feature
$ style fill=#f0f0f0 stroke=#bdbdbd target=bangalore_wards
style stroke=#2b8cbe target=dissolved
style label-text=name dx=15 target=map
style stroke=red fill=#fee8c8 opacity=0.5 target=30mins
$ colorizer name=traveltime colors='#f0f9e8,#bae4bc,#7bccc4,#2b8cbe'
breaks=900,1800,2700
style fill=traveltime(median_time) target=bangalore_wards
[colorizer] Received one or more unexpected parameters: fill=traveltime(median_time) target=bangalore_wards
$ filter 'dstid == 164 && sourceid == 45 && "1,2,3,4,5".indexOf(dow) > -1'
name=commute + target=bangalore-wards-2019-3-WeeklyAggregate
[filter] Retained 5 of 267,352 features
$ sort 'mean_travel_time' descending -calc 'first(dow)'
sort 'mean_travel_time' descending -calc 'last(dow)'
[calc] Command expects a single value. Received: 'first(dow)'sort mean_travel_time descending
$
77.5633,13.0778
UBER_DATA_AN....ipynb
map.geojson
Show all
Type here to search
23:48
16-07-2020


mapshaper
employees
Simplify
Console
Export
mapshaper.org
here='(hod==7 || hod==8) && dstid==39' target=employees
[join] Joined data from 182 source records to 198 target records
[join] 2/200 target records received no data
[join] 414178/414360 source records could not be joined
[join] 584148/414360 source records were skipped
[join] 198/200 target records were matched by multiple source records
$ join hyderabad-wards-2019-3-OnlyWeekdays-HourlyAggregate fields=MOVEMENT_ID keys=MOVEMENT_ID,dstid calc='current_evening = average(mean_travel_time)'
where='(hod==17 || hod==18) && sourceid==39' target=employees
[join] Joined data from 182 source records to 198 target records
[join] 2/200 target records received no data
[join] 414178/414360 source records could not be joined
[join] 584536/414360 source records were skipped
[join] 198/200 target records were matched by multiple source records
$ join hyderabad-wards-2019-3-OnlyWeekdays-HourlyAggregate fields=MOVEMENT_ID keys=MOVEMENT_ID,dstid calc='new_evening = average(mean_travel_time)' where='(hod==15 || hod==16) && sourceid==39' target=employees
[join] Joined data from 180 source records to 197 target records
[join] 3/200 target records received no data
[join] 414180/414360 source records could not be joined
[join] 584538/414360 source records were skipped
[join] 197/200 target records were matched by multiple source records
$ each 'total_current=Math.floor((current_morning+current_evening)/60), total_new=Math.floor((new_morning+new_evening)/60)'
$ calc 'average(total_current)' -calc 'average(total_new)'
[calc] average(total_current): 89.68499999999995
[calc] average(total_new): 72.20000000000002
$ calc '200*(average(total_current) - average(total_new))/60'
[calc] 200*(average(total_current) - average(total_new))/60: 58.28333333333309
$
78.2617,17.4003
hyderabad-wards-....csv
hyderabad_wards.json
employees.json
UBER_DATA_AN....ipynb
map.geojson
Show all
Type here to search
00:35
17-07-2020