About Delhivery:

- Delhivery is the largest and fastest-growing fully integrated player in India by revenue in Fiscal 2021. They aim to build the operating system for commerce, through a combination of world-class infrastructure, logistics operations of the highest quality, and cutting-edge engineering and technology capabilities.

- The Data team builds intelligence and capabilities using this data that helps them to widen the gap between the quality, efficiency, and profitability of their business versus their competitors.

Business Problem Statement:

- The company wants to understand and process the data coming out of data engineering pipelines:

  • Clean, sanitize and manipulate data to get useful features out of raw fields
  • Make sense out of the raw data and help the data science team to build forecasting models on it.

Column Profiling:

- data : tells whether the data is testing or training data

- trip_creation_time : Timestamp of trip creation
- route_schedule_uuid : Unique Id for a particular route schedule
- route_type : Transportation type
  - FTL – Full Truck Load: FTL shipments get to the destination sooner, as the truck is making no other pickups or drop-offs along the way
  - Carting: Handling system consisting of small vehicles (carts)
- trip_uuid : Unique ID given to a particular trip (A trip may include different source and destination centers)
- source_center : Source ID of trip origin
- source_name : Source Name of trip origin
- destination_cente : Destination ID
- destination_name : Destination Name
- od_start_time : Trip start time
- od_end_time : Trip end time
- start_scan_to_end_scan : Time taken to deliver from source to destination
- is_cutoff : Unknown field
- cutoff_factor : Unknown field
- cutoff_timestamp : Unknown field
- actual_distance_to_destination : Distance in Kms between source and destination warehouse
- actual_time : Actual time taken to complete the delivery (Cumulative)
- osrm_time : An open-source routing engine time calculator which computes the shortest path between points in a given map (Includes usual traffic, distance through major and minor roads) and gives the time (Cumulative)
- osrm_distance : An open-source routing engine which computes the shortest path between points in a given map (Includes usual traffic, distance through major and minor roads) (Cumulative)
- factor : Unknown field
- segment_actual_time : This is a segment time. Time taken by the subset of the package delivery
- segment_osrm_time : This is the OSRM segment time. Time taken by the subset of the package delivery
- segment_osrm_distance : This is the OSRM distance. Distance covered by subset of the package delivery
- segment_factor : Unknown field

Understanding the structure

Columns	*Unique values per column*
data	2
trip_creation_time	14817
route_schedule_uuid	1504
route_type	2
trip_uuid	14817
source_center	1508
source_name	1498
destination_center	1481
destination_name	1468
od_start_time	26369
od_end_time	26369
start_scan_to_end_scan	1915
is_cutoff	2
cutoff_factor	501
cutoff_timestamp	93180
actual_distance_to_destination	144515
actual_time	3182
osrm_time	1531
osrm_distance	138046
factor	45641
segment_actual_time	747
segment_osrm_time	214
segment_osrm_distance	113799
segment_factor	5675
trip_creation_day	7
trip_creation_month	2
trip_creation_year	1

dtype: int64

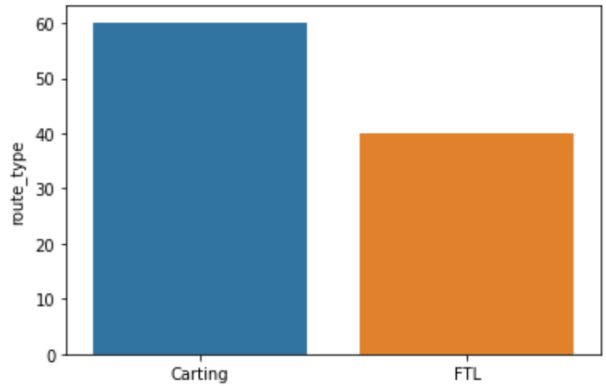
we have **14817 different trips happened between source to destinations.**

total **1504 delivery routes** we have.

1508 unique source centers
1481 unique destination centers

There are two different kind of routes are there :

Route type	% of routes
Carting	60.120132 %
FTL	39.879868 %



From 14817 total different trips , we have

8908 (60%) of the trip-routes are Carting , which consists of small vehicles and

5909 (40%) of total trip-routes are FTL : which are Full Truck Load get to the destination sooner. as no other pickups or drop offs along the way .

OBSERVATIONS IN RAW DATA:

- Each trip is taking stops between mentioned source and destination centers(warehouses).
- od-end-time and od-start-time are the time when the that particular trip was ended and started.
- start-scan-to-end-scan is the time duration of trips are being scanned when start and end.
- start-scan-to-end-scan time is given cumulative. which is not given per trip segments.
- trip cut off False, shows the record of trip when trip changes from one warehouse to another. between source to destination.
- Actual-time given is the time to complete the entire delivery from source to destination (given cumulatively)
- OSRM -time is an open-source routing engine time calculator which computes the shortest path between points in a given map and gives the time and OSRM distance gives the shortest distance (given cumulatively)
- Actual-distance-to-destination is the actual distance between warehouses, given cumulative during the trip.
- every time cutoff is False, distance count starts from beginning.
- Segment actual time, is the actual time taken between two stops in between trips. given per each segment (taken between subset of package delivery)
- segment OSRM time is the OSRM segment time, taken between subset of package delivery

After feature engineering Checking Data information and Null Values presence :

data	0
trip_creation_time	0
route_schedule_uuid	0
route_type	0
trip_uuid	0
source_center	0
source_name	293
destination_center	0
destination_name	261
od_start_time	0
od_end_time	0
start_scan_to_end_scan	0
is_cutoff	0
cutoff_factor	0
cutoff_timestamp	0
actual_distance_to_destination	0
actual_time	0
osrm_time	0
osrm_distance	0
factor	0
segment_actual_time	0
segment_osrm_time	0
segment_osrm_distance	0
segment_factor	0
trip_creation_day	0
trip_creation_month	0
trip_creation_year	0
source_city	293
source_state	293
destination_city	261
destination_state	261
source_place	2400
destination_place	2702
source_pincode	0
destination_pincode	0
time_taken_btwn_odstart_and_od_end	0

Hypothesis Tests for time durations and distance related features:

Analysing TimeTaken Between OdStart and OdEnd time & StartScanToEndScan :

H0: Mean of time taken betweenn trip end ans start time = Mean of start and end scan time

Ha: Mean of time taken betweenn trip end ans start time != Mean of start and end scan time

```
stats.ks_2samp(time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"]
               ,start_scan_to_end_scan["start_scan_to_end_scan"])
```

```
KstestResult(statistic=0.004184382803536474, pvalue=0.9994337058695081)
```

```
for i in range(5):
    print(stats.ttest_ind((time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"].sample(3000))
                ,(start_scan_to_end_scan["start_scan_to_end_scan"].sample(3000))))
```

```
Ttest_indResult(statistic=-0.9716508785225971, pvalue=0.33126344290575926)
Ttest_indResult(statistic=0.5567970071788092, pvalue=0.5776869027555587)
Ttest_indResult(statistic=0.9443024700723182, pvalue=0.34505313959905415)
Ttest_indResult(statistic=-0.1332016137806006, pvalue=0.8940384030818919)
Ttest_indResult(statistic=-1.0779783124344828, pvalue=0.28108678529475184)
```

from Kolmogorov–Smirnov test , p-value is 0.9943 , from which we can conclude tht both the distributions

(time_taken_btwn_odstart_and_od_end and start_scan_to_end_scan) are closly similar.

from 2 sample t-test ,we can also conclude that Average time_taken_btwn_odstart_and_od_end for population is also equal to Average start_scan_to_end_scan for population.

Analysing Actual Time taken to complete the delivery & start-scan-end-scan

H0: Mean of start and end scan time <= Mean of Actual time taken to complete delivery

Ha: Mean of start and end scan time > Mean of Actual time taken to complete delivery

```
stats.ks_2samp(actual_time["actual_time"],start_scan_to_end_scan["start_scan_to_end_scan"])
```

```
KstestResult(statistic=0.27387460349598436, pvalue=0.0)
```

```
for i in range(7):
    print(stats.ttest_ind((actual_time["actual_time"].sample(3000))
                ,(start_scan_to_end_scan["start_scan_to_end_scan"].sample(3000)),alternative="less"))
```

```
Ttest_indResult(statistic=-11.63433183388071, pvalue=2.9583265159466375e-31)
Ttest_indResult(statistic=-10.646343810381081, pvalue=1.553084944843656e-26)
Ttest_indResult(statistic=-10.156542708781046, pvalue=2.422564888869492e-24)
Ttest_indResult(statistic=-10.114166980280935, pvalue=3.7103301862926956e-24)
Ttest_indResult(statistic=-11.503124522312334, pvalue=1.3201426804392581e-30)
Ttest_indResult(statistic=-11.480381777463917, pvalue=1.7080817943699753e-30)
Ttest_indResult(statistic=-11.828941944307852, pvalue=3.124724654614158e-32)
```

from KS test for actual-time and start_scan_to_end_scan distributions are not same.

from ttest of population average actual_time is less than population average start_scan_to_end_scan.

Analyzing Actual Time & Time Taken between start and end trip time.

H0: Mean of Actual time taken to complete delivery = Mean of time taken between trip end and start time

Ha: Mean of Actual time taken to complete delivery != Mean of time taken betweenn trip end and start time

```
stats.ks_2samp(actual_time["actual_time"],time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"])
```

```
KstestResult(statistic=0.2765067152594992, pvalue=0.0)
```

```
for i in range(5):
    print(stats.ttest_ind((actual_time["actual_time"].sample(1000))
                ,(time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"].sample(1000))))
```

```
Ttest_indResult(statistic=-6.339363378930524, pvalue=2.844207890862028e-10)
Ttest_indResult(statistic=-5.287428254125523, pvalue=1.3760850460629462e-07)
Ttest_indResult(statistic=-5.426527295043085, pvalue=6.444232760235031e-08)
Ttest_indResult(statistic=-7.008804877370162, pvalue=3.274269234731331e-12)
Ttest_indResult(statistic=-7.274518124194945, pvalue=4.969630991558775e-13)
```

from above kstest of distribution and two sample ttest ,

we can conclude that population mean Actual time taken to complete delivery and population mean time_taken_btwn_od_start_and_od_end are also not same.

Analyzing Actual Time taken to complete delivery from source to destination hub & OSRM measured time :

H0: Mean of OSRM time >= Mean of Actual time taken to complete delivery

Ha: Mean of OSRM time < Mean of Actual time taken to complete delivery

```
stats.ks_2samp(actual_time["actual_time"],
               osrm_time["osrm_time"])
```

```
KstestResult(statistic=0.2945265573327934, pvalue=0.0)
```

```
for i in range(5):
    print(stats.ttest_ind(actual_time["actual_time"].sample(5000),
               osrm_time["osrm_time"].sample(5000),alternative='greater'))
```

```
Ttest_indResult(statistic=22.275852062067113, pvalue=1.2586832394425356e-107)
Ttest_indResult(statistic=22.840308069504303, pvalue=6.723368237976783e-113)
Ttest_indResult(statistic=21.655547845863687, pvalue=5.708722419472675e-102)
Ttest_indResult(statistic=22.748352118857557, pvalue=4.94980866328736e-112)
Ttest_indResult(statistic=22.70043491079571, pvalue=1.3967821910213581e-111)
```

from two sample ttest can conclude , that population mean actual time taken to complete delivery from source to warehouse and orsm estimate mean time for population are not same.

actual time is higher than the osrm estimated time for delivery

Analyzing Actual Time taken to complete delivery from source to destination hub & Segment Actual Time :

H0: Actual time = segment actual time

Ha: Actual time != segment actual time

```
for i in range(7):
    print(stats.ttest_ind((actual_time["actual_time"].sample(3000)),
                (segment_actual_time["segment_actual_time"].sample(3000))))
```

```
Ttest_indResult(statistic=0.668982430355294, pvalue=0.5035324325305446)
Ttest_indResult(statistic=1.0928793187629657, pvalue=0.27449068140309024)
Ttest_indResult(statistic=0.18976302834376332, pvalue=0.8495012497910923)
Ttest_indResult(statistic=1.8787326099267305, pvalue=0.06032947096429208)
Ttest_indResult(statistic=0.261938449594607, pvalue=0.7933778903832364)
Ttest_indResult(statistic=-1.8172223579091173, pvalue=0.06923295050680427)
Ttest_indResult(statistic=-0.17973873333446588, pvalue=0.8573637553487372)
```

from two sample ttest , we can conclude that
Population average for
Actual Time taken to complete delivery trip and segment actual time are same.

Analysing osrm Time & segment-osrm-time :

H0: segment actual time <= OSRM time

Ha: segment actual time > OSRM time

```
for i in range(7):
    print(stats.ttest_ind((osrm_time["osrm_time"].sample(3000)),
                (segment_osrm_time["segment_osrm_time"].sample(3000)),alternative ="less"))
```

```
Ttest_indResult(statistic=-3.9043900186502776, pvalue=4.774987601124152e-05)
Ttest_indResult(statistic=-2.3492890821791597, pvalue=0.009420778139131399)
Ttest_indResult(statistic=-3.8650902297510834, pvalue=5.611034049972779e-05)
Ttest_indResult(statistic=-2.9894378987333314, pvalue=0.0014031259058058675)
Ttest_indResult(statistic=-1.3435499156387603, pvalue=0.08957240009955342)
Ttest_indResult(statistic=-2.197740666984369, pvalue=0.014002838853677181)
Ttest_indResult(statistic=-2.7891500916110057, pvalue=0.0026506620667620814)
```

from ttest , we can conclude that
average of osrm Time & segment-osrm-time for population is not same.
Population Mean osrm time is less than Population Mean segment osrm time.

Analysing Distances measures :

Analysing and Visulizing OSRM Estimated distance and Segment-osrm-distance :

H0 : Segment OSRM distance <= OSRM distance

Ha : Segment OSRM distance > OSRM distance

```
stats.ks_2samp(osrm_distance["osrm_distance"],segment_osrm_distance["segment_osrm_distance"])
```

```
KstestResult(statistic=0.03948167645272321, pvalue=1.8042208791084262e-10)
```

```
for i in range(7):
    print(stats.ttest_ind(osrm_distance["osrm_distance"].sample(5000),
                segment_osrm_distance["segment_osrm_distance"].sample(5000),alternative="less"))
```

```
Ttest_indResult(statistic=-2.491000734355803, pvalue=0.006377239874874842)
Ttest_indResult(statistic=-2.102001693118751, pvalue=0.017789038400820426)
Ttest_indResult(statistic=-3.3966080254519477, pvalue=0.00034246196944600184)
Ttest_indResult(statistic=-2.5542752505894315, pvalue=0.005327801097742841)
Ttest_indResult(statistic=-1.0712919316998133, pvalue=0.14203202292688147)
Ttest_indResult(statistic=-0.6319798477969442, pvalue=0.26370724791908506)
Ttest_indResult(statistic=-2.8196040126585413, pvalue=0.002408876652137734)
```

from KS test , we can conclude the distributions of segment OSRM distance and OSRM distance are not same!

from two sample one sided t-test, we can conclude: Average of OSRM distance for population is less than average of segment OSRM distance

Analyzing and Visualizing OSRM Estimated distance and Actual Distance between source and destination warehouse :

H0 : Mean OSRM distance <= Mean Actual distance

Ha : Mean OSRM distance > Mean Actual distance

```
stats.ks_2samp(osrm_distance["osrm_distance"],actual_distance_to_destination["actual_distance_to_destination"])
```

```
KstestResult(statistic=0.11837753931295136, pvalue=6.578385372142345e-91)
```

```
for i in range(5):
    print(stats.ttest_ind(osrm_distance["osrm_distance"].sample(5000),
                actual_distance_to_destination["actual_distance_to_destination"].sample(5000),alternative="greater"))
```

```
Ttest_indResult(statistic=5.986800487278355, pvalue=1.1068469174541993e-09)
Ttest_indResult(statistic=6.733721409029123, pvalue=8.723285355072898e-12)
Ttest_indResult(statistic=7.414041580271733, pvalue=6.622254242522077e-14)
Ttest_indResult(statistic=5.3925866490117045, pvalue=3.5516787577043784e-08)
Ttest_indResult(statistic=5.791549079071094, pvalue=3.5922124805689545e-09)
```

From left sided t-test , we can conclude

for population OSRM estimated distance is higher than the actual distance from source to destination warehouse.

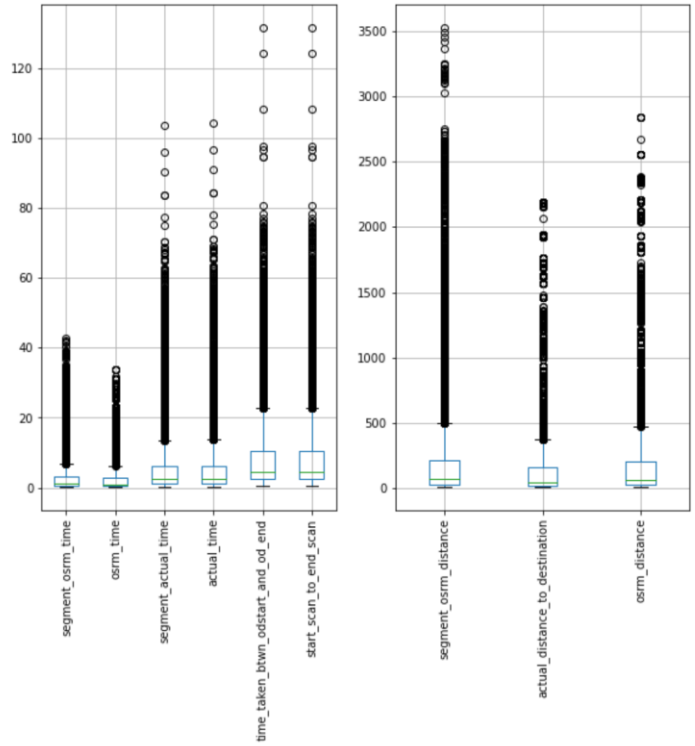
After Aggregating and Merging All the numerical Fields as per Trip-ID and Route-ID:

Data uploaded here in Github link :

https://github.com/28101991SUNNY/BusinessCase_Data_Exploration-/tree/main/Delhivery

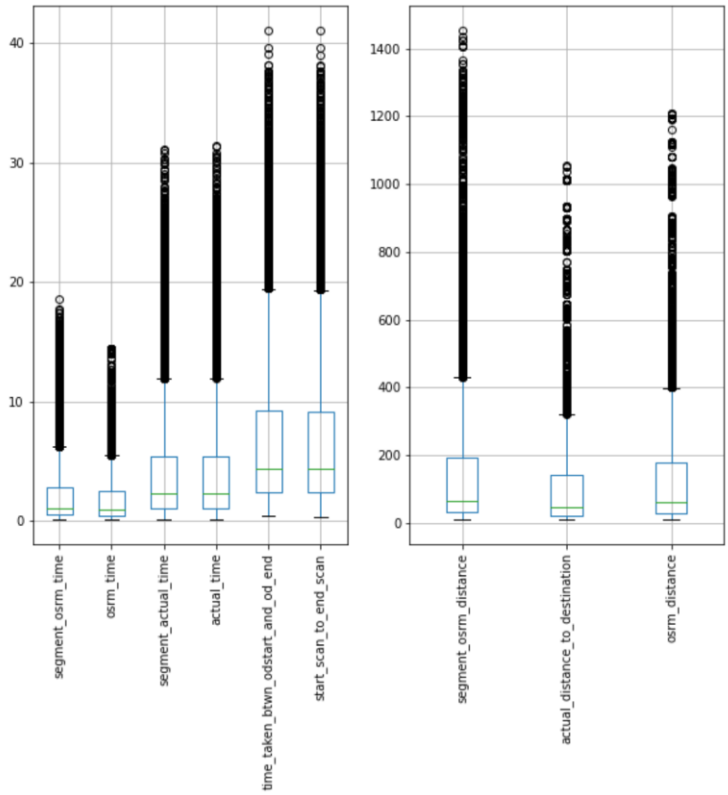
Treating Outliers :

- checking distributions using boxplot for all numerical figures :



After treating outliers using z-score :

Removing all the figures beyond +/- 3 standard deviation :



Exploratory Data Analysis: (getting some insights from preprocessed data:)

Busiest Route Analysis :

Number of Trips between cities , sorted highest to lowest

Top 20 source and destination cities wihc have high freqency of trips in between .

	source_city_state	destination_city_state	trip_uuid
0	Bengaluru Karnataka	Bengaluru Karnataka	1369
1	Bhiwandi Maharashtra	Mumbai Maharashtra	512
2	Mumbai Maharashtra	Mumbai Maharashtra	361
3	Hyderabad Telangana	Hyderabad Telangana	308
4	Mumbai Maharashtra	Bhiwandi Maharashtra	282
5	Delhi Delhi	Gurgaon Haryana	248
6	Gurgaon Haryana	Delhi Delhi	237
7	Mumbai Hub Maharashtra	Mumbai Maharashtra	227
8	Chennai Tamil Nadu	Chennai Tamil Nadu	205
9	MAA Tamil Nadu	Chennai Tamil Nadu	204
10	Chennai Tamil Nadu	MAA Tamil Nadu	141
11	Bengaluru Karnataka	HBR Karnataka	133
12	Ahmedabad Gujarat	Ahmedabad Gujarat	131
13	Pune Maharashtra	PNQ Maharashtra	122
14	Jaipur Rajasthan	Jaipur Rajasthan	111
15	Delhi Delhi	Delhi Delhi	109
16	Pune Maharashtra	Bhiwandi Maharashtra	107
17	Pune Maharashtra	Pune Maharashtra	101
18	Chandigarh Chandigarh	Chandigarh Punjab	100
19	Kolkata West Bengal	CCU West Bengal	96
20	Gurgaon Haryana	Sonipat Haryana	92
21	Sonipat Haryana	Gurgaon Haryana	86
22	Chandigarh Punjab	Chandigarh Chandigarh	84
23	HBR Karnataka	Bengaluru Karnataka	79
24	Bengaluru Karnataka	BLR Karnataka	78

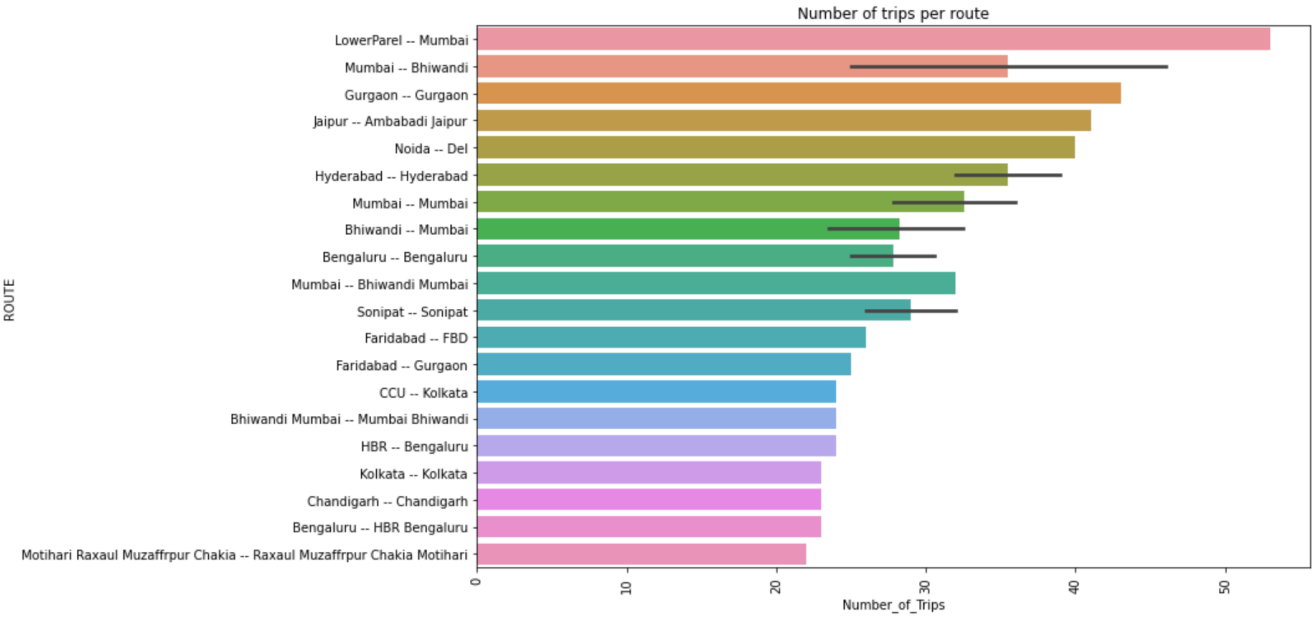
	source_city_state	destination_city_state	trip_uuid
1	Bhiwandi Maharashtra	Mumbai Maharashtra	512
4	Mumbai Maharashtra	Bhiwandi Maharashtra	282
5	Delhi Delhi	Gurgaon Haryana	248
6	Gurgaon Haryana	Delhi Delhi	237
7	Mumbai Hub Maharashtra	Mumbai Maharashtra	227
9	MAA Tamil Nadu	Chennai Tamil Nadu	204
10	Chennai Tamil Nadu	MAA Tamil Nadu	141
11	Bengaluru Karnataka	HBR Karnataka	133
13	Pune Maharashtra	PNQ Maharashtra	122
16	Pune Maharashtra	Bhiwandi Maharashtra	107
18	Chandigarh Chandigarh	Chandigarh Punjab	100
19	Kolkata West Bengal	CCU West Bengal	96
20	Gurgaon Haryana	Sonipat Haryana	92
21	Sonipat Haryana	Gurgaon Haryana	86
22	Chandigarh Punjab	Chandigarh Chandigarh	84
23	HBR Karnataka	Bengaluru Karnataka	79
24	Bengaluru Karnataka	BLR Karnataka	78
26	Del Delhi	Gurgaon Haryana	76
27	Bhiwandi Maharashtra	Pune Maharashtra	72
28	Ludhiana Punjab	Chandigarh Punjab	71
30	Chandigarh Punjab	Gurgaon Haryana	66
31	Gurgaon Haryana	Bengaluru Karnataka	66
32	LowerParel Maharashtra	Mumbai Maharashtra	65
34	Mumbai Hub Maharashtra	Bhiwandi Maharashtra	63
35	PNQ Maharashtra	Pune Maharashtra	62

From above table, we can observe that Mumbai Maharashtra ,Delhi , Gurgaon(Haryana),Bengaluru Karnataka ,Hyderabad Telangana, Chennai Tamil Nadu, Ahmedabad-Gujarat, Pune Maharashtra, Chandigarh Chandigarh and Kolkata West Bengal are some cities have higest amount of trips happening states with in the city

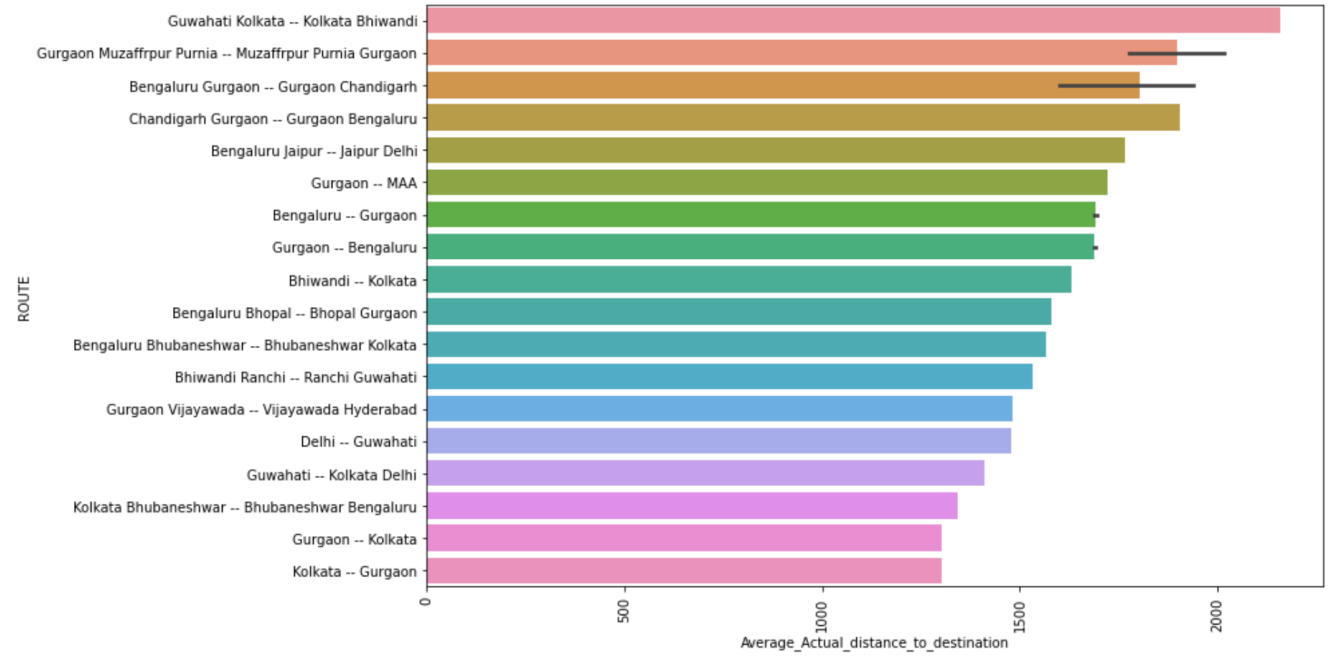
If we talk about , not having equal source and destination states , source and destination cities having higest number of trips in between are : Delhi TO Gurgao , Gurgaon TO Bengaluru , Bhiwandi/Mumbai TO Pune Maharashtra , Sonipat TO Gurgaon,Haryana
It is also been observed that lots of deliveries are happening to airpots like : Chennai to MAA chennai international Airport , Pune to Pune Airport (PNQ),Kolkata to CCU West Bengal Kolkata International Airport , Bengluru to BLR-Bengaluru International Airport etc.

Below chart tells

Top Routes having Maximum Number of Trips between/within the source and destinations .



Top Routes as per distance between the source and destinations.



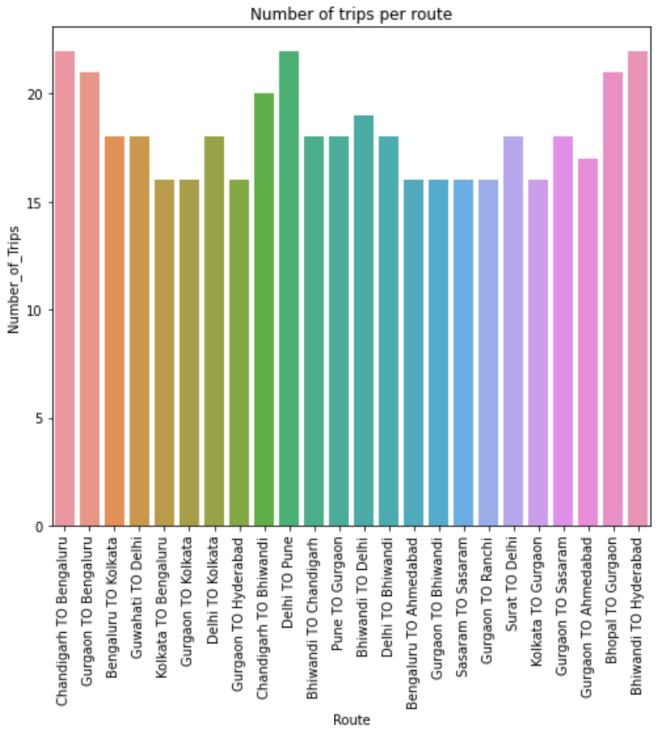
From above Bar chart , and table , we can observe that higest trips are happening is with in the particular cities.

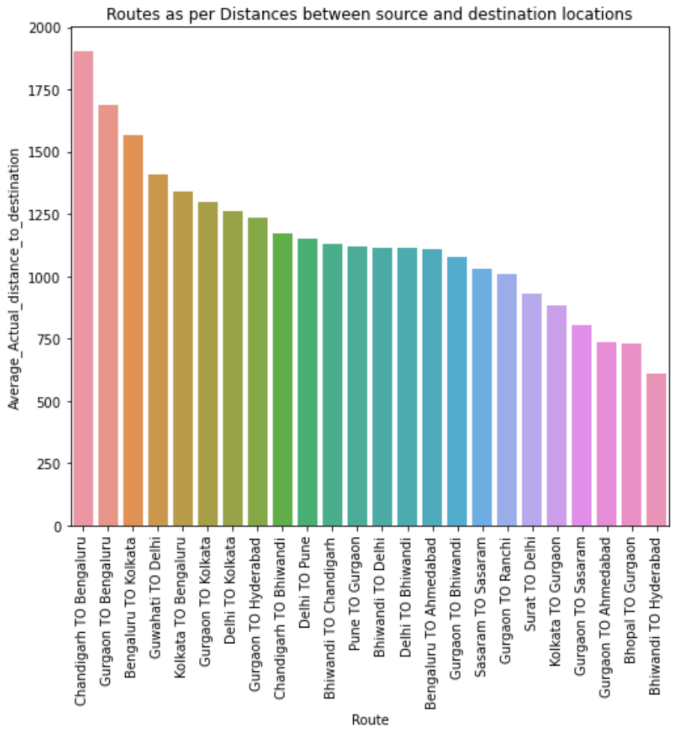
in terms of average distance between destinations , we can observe Guwahati to Mumbai , Benglore to Chandigarh ,Benglore to Delhi , Benglore to Gurgaon are the longest routes .

Busiest and Longest Routes:

source_cities	destination_cities	Number_of_Trips	Average_Actual_distance_to_destination
Chandigarh Gurgaon	Gurgaon Bengaluru	22.0	1905.766051
Gurgaon	Bengaluru	21.0	1689.873158
Gurgaon	Bengaluru	21.0	1689.791894
Bengaluru Bhubaneshwar	Bhubaneshwar Kolkata	18.0	1567.577507
Guwahati	Kolkata Delhi	18.0	1411.208424
Kolkata Bhubaneshwar	Bhubaneshwar Bengaluru	16.0	1342.143081
Gurgaon	Kolkata	16.0	1300.572161
Delhi Gurgaon	Gurgaon Kolkata	18.0	1263.113211
Gurgaon	Hyderabad	16.0	1236.572072
Chandigarh Gurgaon	Gurgaon Bhiwandi	20.0	1170.817927
Delhi Gurgaon	Gurgaon Pune	22.0	1151.514940
Bhiwandi Sonipat	Sonipat Chandigarh	18.0	1129.609705
Pune	Gurgaon	18.0	1120.729446
Bhiwandi	Delhi	19.0	1114.214670
Delhi	Bhiwandi	18.0	1114.182197
Bengaluru Kolhapur Surat	Kolhapur Surat Ahmedabad	16.0	1110.015339
Gurgaon	Bhiwandi	16.0	1078.076312
Sasaram Kanpur Kolkata Dhanbad	Kanpur Gurgaon Dhanbad Sasaram	16.0	1028.024726
Gurgaon	Ranchi	16.0	1010.953223
Surat	Delhi	18.0	931.980821
Kolkata Ranchi	Ranchi Gurgaon	16.0	881.621264
Gurgaon	Sasaram	18.0	804.210670
Gurgaon	Ahmedabad	17.0	735.550450
Bhopal Kanpur Auraiya Etawah	Kanpur Auraiya Etawah Gurgaon	21.0	731.634456
Bhiwandi	Hyderabad	22.0	607.514619

Above Table shows the souce to destination city routes having largest numbers of trip happening having large distnaces :





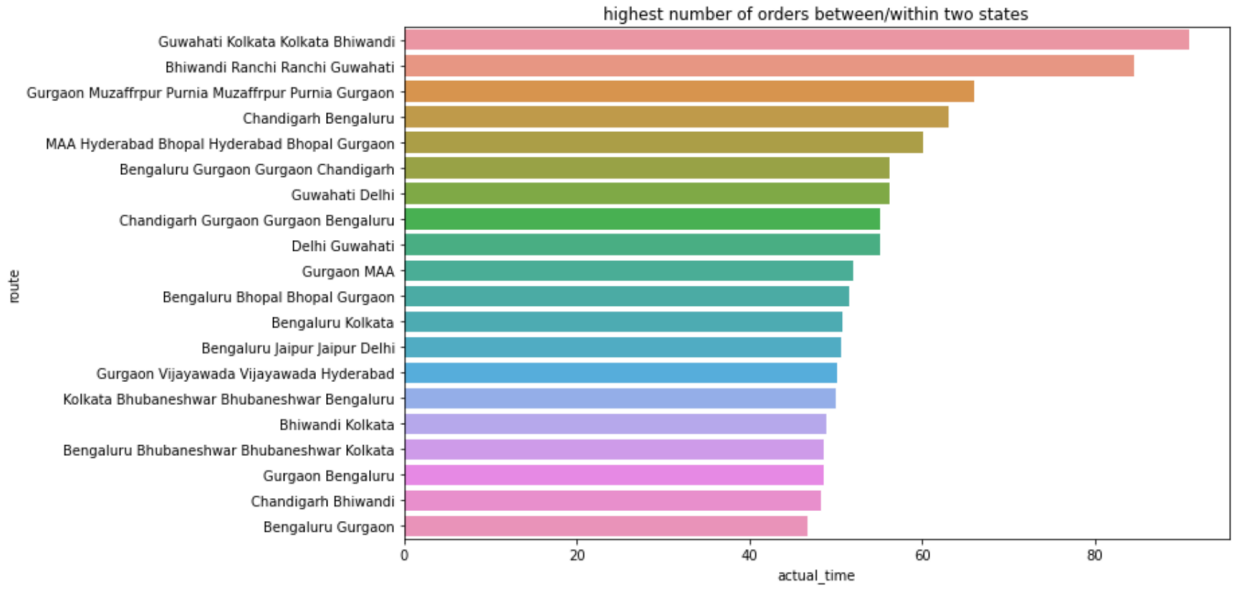
Top 25 Routes which passes through Maximum Numbers of Cities :

SouceToDestination_city	Number_of_Trips	Average_Actual_distance_to_destination	#source_cities	#destination_cities
Guwahati TO LakhimpurN	14.0	281.596486	13	11
Jaipur TO Tarnau	20.0	351.611796	10	10
Guwahati TO Tura	12.0	332.602225	10	10
Mangalore TO Udupi	9.0	195.257193	9	9
Ajmer TO Raipur	20.0	178.737233	9	8
Mainpuri TO Tilhar	12.0	207.247057	8	8
Hassan TO Koppa	21.0	200.497832	7	7
Shrirampur TO Sangamner	20.0	204.509529	7	7
Musiri TO Tiruchi	19.0	219.845121	7	7
Bijnor TO Bijnor	17.0	209.400685	7	7
Dausa TO Lalsot	17.0	232.408310	7	7
Tinusukia TO Dibrugarh	16.0	111.098543	7	7
Pondicherry TO Pondicherry	12.0	230.253602	7	7
Mysore TO Mysore	12.0	154.324190	7	7
Golaghat TO Guwahati	11.0	258.546587	7	7
Varanasi TO Varanasi	8.0	82.545019	7	7
Vijayawada TO Suryapet	8.0	407.029391	7	7
Hyderabad TO Miryalguda	7.0	420.603709	7	7
Srikakulam TO Bobbili	22.0	154.495283	6	6
Pukhrayan TO Kanpur	22.0	139.834945	6	6
Dhule TO Shirpur	22.0	150.016233	6	6
Madhupur TO Madhupur	21.0	252.072259	6	6
Kamareddy TO Kamareddy	21.0	177.923330	6	6
Noida TO Khurja	21.0	208.714043	6	6
Junagadh TO Veraval	19.0	179.538596	6	6

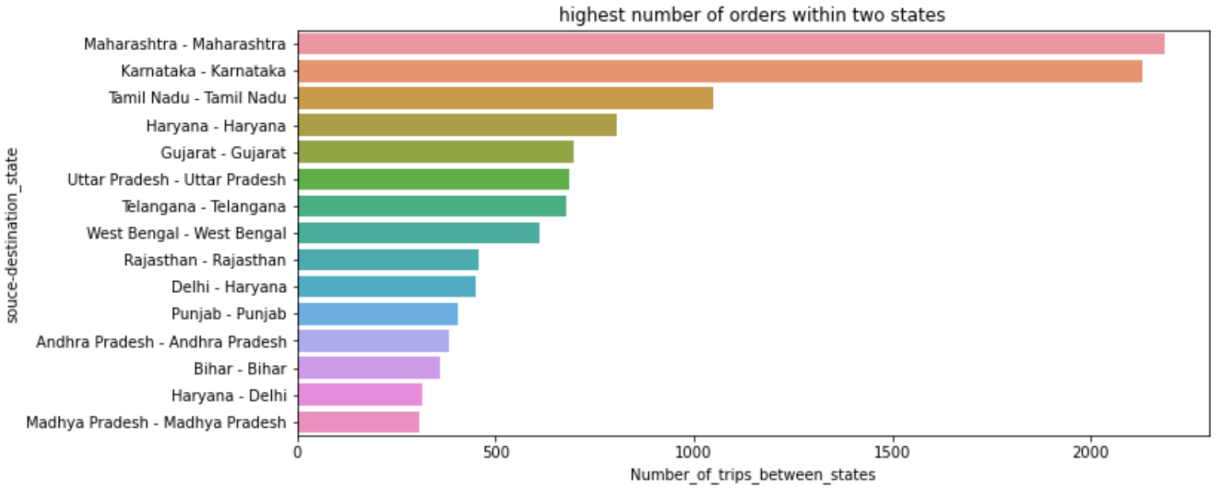
From above table shows the routes which covered multiple cities in between source and destination.

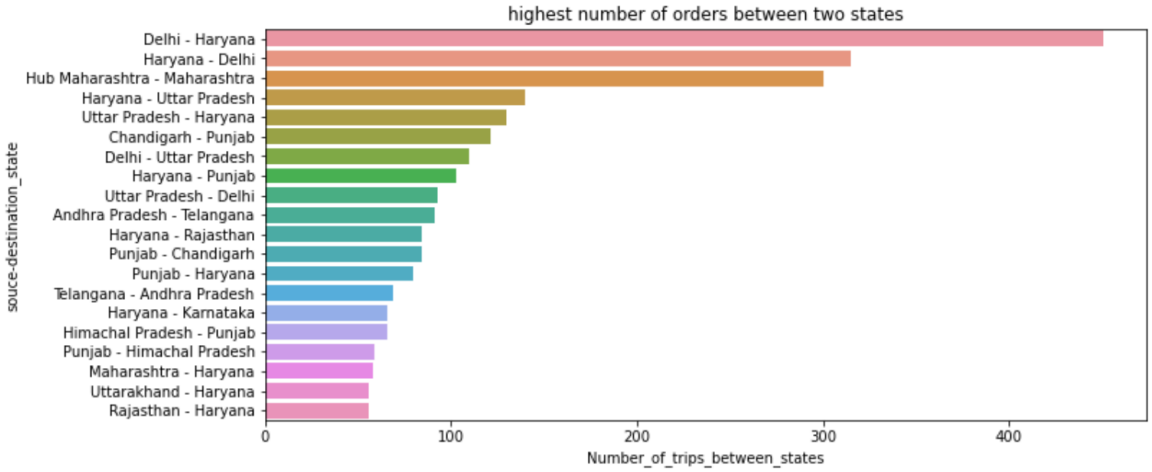
Most covered cities routes are : Guwahati TO LakhimpurN , Jaipur TO Tarnau , Guwahati TO Tura , Mangalore TO Udupi , Ajmer TO Raipur , Mainpuri TO Tilhar . which passes through more than 8 cities.

Top 20 Longest Route as per : Average actual time taken:



Highest number of Trips happening between/within two states:

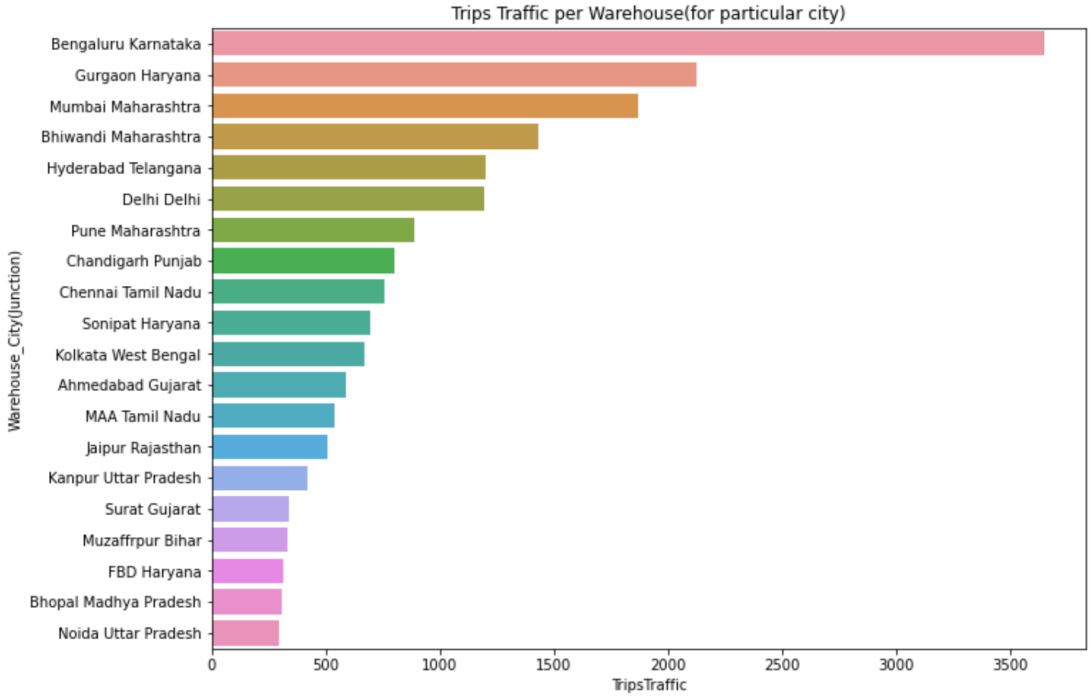




From above charts ,

Delhi to Haryana is the busiest route, having more than 400 trips in between. Some of such busy routes are Haryana to Uttar Pradesh , Chandigarh to Punjab , Delhi to Uttar Pradesh .
Within the state , Maharashtra , Karnataka, Tamil Nadu are some states having above 1000 trips.

Top 20 warehouses with heavy traffic:



From above chart are some warehouse having Maximum traffic and hence busiest junctions.
'Bengaluru Karnataka', 'Gurgaon Haryana', 'Mumbai Maharashtra', 'Bhiwandi Maharashtra', 'Hyderabad Telangana', 'Delhi Delhi', 'Pune Maharashtra', 'Chandigarh Punjab', 'Chennai Tamil Nadu', 'Sonipat Haryana', Kolkata West Bengal', 'Ahmedabad Gujarat', 'MAA Tamil Nadu', 'Jaipur Rajasthan', 'Kanpur Uttar Pradesh', 'Surat Gujarat', 'Muzaffrpur Bihar', 'FBD Haryana', 'Bhopal Madhya Pradesh', 'Noida Uttar Pradesh'

Inferences and Recommendations :

Insights and Observations :

- 14817 different trips happened between source to destinations during 2018 , September and October.
- 1504 delivery routes on which trips are happenig.
- we have 1508 unique source centers and 1481 unique destination centers

- From 14817 total different trips , we have 8908 (60%) of the trip-routes are Carting , which consists of small vehicles and 5909 (40%) of total trip-routes are FTL : which are Full Truck Load get to the destination sooner. as no other pickups or drop offs along the way .

- *Hypothesis tests Results : (In.[52] to In.[89])*

  from 2 sample t-test ,we can also conclude that
  - Average time_taken_btwn_odstart_and_od_end for population is equal to Average start_scan_to_end_scan for population.
  - population average actual_time is less than population average start_scan_to_end_scan.
  - population mean Actual time taken to complete delivery and population mean time_taken_btwn_od_start_and_od_end are also not same.
  - Mean of actual time is higher than Mean of the OSRM estimated time for delivery
  - Population average for Actual Time taken to complete delivery trip and segment actual time are same.
  - Average of OSRM Time & segment-osrm-time for population is not same.
  - Population Mean osrm time is less than Population Mean segment osrm time.
  - Average of OSRM distance for population is less than average of segment OSRM distance
  - population OSRM estimated distance is higher than the actual distance from source to destination warehouse.

- *From Exploratory Data Analysis (Cells In.[154] to In.[172])*

- we can observe that Mumbai Maharashtra ,Delhi , Gurgaon(Haryana),Bengaluru Karnataka ,Hyderabad Telangana, Chennai Tamil Nadu, Ahmedabad-Gujarat, Pune Maharashtra, Chandigarh Chandigarh and Kolkata West Bengal are some cities have higest amount of trips happening states with in the city.
- If we talk about , not having equal source and destination states , source and destination cities having higest number of trips in between are : Delhi TO Gurgao , Gurgaon TO Bengaluru , Bhiwandi/Mumbai TO Pune Maharashtra , Sonipat TO Gurgaon,Haryana
- It is also been observed that lots of deliveries are happening to airpots like : Chennai to MAA chennai international Airport , Pune to Pune Airport (PNQ),Kolkata to CCU West Bengal Kolkata International Airport , Bengluru to BLR-Bengaluru International Airport etc.

- From Bar charts , and calculated tables in analysis , we can observe that higest trips are happening is with in the particular cities, in terms of average distance between destinations , we can observe Guwahati to Mumbai , Benglore to Chandigarh ,Benglore to Delhi , Benglore to Gurgaon are the longest routes.

- *the souce to destination city routes having largest numbers of trip happening having large distnaces :In[160-163] :*
  - Guwahati TO Bhiwandi, Bengaluru TO Chandigarh, Bengaluru TO Delhi,Gurgaon TO MAA Chennai Airport,Bhiwandi TO Kolkata, Bengaluru TO Kolkata, Gurgaon TO Hyderabad, Gurgaon TO Kolkata

- *the routes which covered multiple cities in between source and destination :*
  - Most covered cities routes are : Guwahati TO LakhimpurN , Jaipur TO Tarnau , Guwahati TO Tura , Mangalore TO Udupi , Ajmer TO Raipur , Mainpuri TO Tilhar . which passes through more than 8 cities.

- *Routes which are busiest from source to destinations and states in which highest activities are noticed :*
  - Delhi to Haryana is the busiest route, having more than 400 trips in between. Some of such busy routes are Haryana to Uttar Pradesh , Chandigarh to Punjab , Delhi to Uttar Pradesh .

    - Within the state , Maharashtra , Karnataka, Tamil Nadu, Haryana, Telangana, Gujarat , West Benglore and Uttar Pradesh are some states having above 1000 trips. (In.[173])

- *From above chart(In.[172]) are some warehouse having Maximum traffic and hence busiest junctions.*
    - Bengaluru Karnataka, Gurgaon Haryana, Mumbai Maharashtra, Hyderabad Telangana, Delhi, Pune Maharashtra, Chandigarh Punjab, Chennai Tamil Nadu, Sonipat Haryana, Kolkata West Bengal, Ahmedabad Gujarat, MAA Tamil Nadu, Jaipur Rajasthan, Kanpur Uttar Pradesh, Surat Gujarat, Muzaffrpur Bihar, FBD Haryana, Bhopal Madhya Pradesh, Noida Uttar Pradesh.

Recommendations :

- As per analysis, It is recommended to use Carting (small vehicles) for delivery with in the city in order to reduce the delivery time, and Heavy trucks for long distance trips or heavy load. based on this , we can optimize the delivery time as well as increase the revenue as per requirements.
- Incresing the connectivity in tier 2 and tier 3 cities along with profession tie-ups with several e-commerce giants can increase the revenue as well as the reputation on connectivity across borders.
- We can work on optimizing the scanning time on both ends which is start scanning time and end scanning time so that the delivery time can be equated to the OSRM estimated delivery time.

Thank you .